User's Manual

RZ/T2L Group

User's Manual: Hardware

RZ/T Series for Real-Time Control
RZ Family
64-Bit & 32-Bit Arm®-Based High-End MPUs

Renesas Confidential

All information contained in these materials, including products and product specifications, represents information on the product at the time of publication and is subject to change by Renesas Electronics Corp. without notice. Please review the latest information published by Renesas Electronics Corp. through various means, including the Renesas Electronics Corp. website (http://www.renesas.com).

Renesas Electronics
www.renesas.com

Rev.1.20 Oct 2023

Notice

1. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation or any other use of the circuits, software, and information in the design of your product or system. Renesas Electronics disclaims any and all liability for any losses and damages incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics hereby expressly disclaims any warranties against and liability for infringement or any other claims involving patents, copyrights, or other intellectual property rights of third parties, by or arising from the use of Renesas Electronics products or technical information described in this document, including but not limited to, the product data, drawings, charts, programs, algorithms, and application examples.
3. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
4. You shall be responsible for determining what licenses are required from any third parties, and obtaining such licenses for the lawful import, export, manufacture, sales, utilization, distribution or other disposal of any products incorporating Renesas Electronics products, if required.
5. You shall not alter, modify, copy, or reverse engineer any Renesas Electronics product, whether in whole or in part. Renesas Electronics disclaims any and all liability for any losses or damages incurred by you or third parties arising from such alteration, modification, copying or reverse engineering.
6. Renesas Electronics products are classified according to the following two quality grades: “Standard” and “High Quality”. The intended applications for each Renesas Electronics product depends on the product’s quality grade, as indicated below.
   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; industrial robots; etc.
   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control (traffic lights); large-scale communication equipment; key financial terminal systems; safety control equipment; etc.
   Unless expressly designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not intended or authorized for use in products or systems that may pose a direct threat to human life or bodily injury (artificial life support devices or systems; surgical implantations; etc.), or may cause serious property damage (space system; undersea repeaters; nuclear power control systems; aircraft control systems; key plant systems; military equipment; etc.). Renesas Electronics disclaims any and all liability for any damages or losses incurred by you or any third parties arising from the use of any Renesas Electronics product that is inconsistent with any Renesas Electronics data sheet, user’s manual or other Renesas Electronics document.
7. No semiconductor product is absolutely secure. Notwithstanding any security measures or features that may be implemented in Renesas Electronics hardware or software products, Renesas Electronics shall have absolutely no liability arising out of any vulnerability or security breach, including but not limited to any unauthorized access to or use of a Renesas Electronics product or a system that uses a Renesas Electronics product. RENESAS ELECTRONICS DOES NOT WARRANT OR GUARANTEE THAT RENESAS ELECTRONICS PRODUCTS, OR ANY SYSTEMS CREATED USING RENESAS ELECTRONICS PRODUCTS WILL BE INVULNERABLE OR FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (“Vulnerability Issues”). RENESAS ELECTRONICS DISCLAIMS ANY AND ALL RESPONSIBILITY OR LIABILITY ARISING FROM OR RELATED TO ANY VULNERABILITY ISSUES. FURTHERMORE, TO THE EXTENT PERMITTED BY APPLICABLE LAW, RENESAS ELECTRONICS DISCLAIMS ANY AND ALL WARRANTIES, EXPRESS OR IMPLIED, WITH RESPECT TO THIS DOCUMENT AND ANY RELATED OR ACCOMPANYING SOFTWARE OR HARDWARE, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OF MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE.
8. When using Renesas Electronics products, refer to the latest product information (data sheets, user’s manuals, application notes, “General Notes for Handling and Using Semiconductor Devices” in the reliability handbook, etc.), and ensure that usage conditions are within the ranges specified by Renesas Electronics with respect to maximum ratings, operating power supply voltage range, heat dissipation characteristics, installation, etc. Renesas Electronics disclaims any and all liability for any malfunctions, failure or accident arising out of the use of Renesas Electronics products outside of such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of Renesas Electronics products, semiconductor products have specific characteristics, such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Unless designated as a high reliability product or a product for harsh environments in a Renesas Electronics data sheet or other Renesas Electronics document, Renesas Electronics products are not subject to radiation resistance design. You are responsible for implementing safety measures to guard against the possibility of bodily injury, injury or damage caused by fire, and/or danger to the public in the event of a failure or malfunction of Renesas Electronics products, such as safety design for hardware and software, including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult and impractical, you are responsible for evaluating the safety of the final products or systems manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. You are responsible for carefully and sufficiently investigating applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive, and using Renesas Electronics products in compliance with all these applicable laws and regulations. Renesas Electronics disclaims any and all liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. Renesas Electronics products and technologies shall not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations. You shall comply with any applicable export control laws and regulations promulgated and administered by the governments of any countries asserting jurisdiction over the parties or transactions.
12. It is the responsibility of the buyer or distributor of Renesas Electronics products, or any other party who distributes, disposes of, or otherwise sells or transfers the product to a third party, to notify such third party in advance of the contents and conditions set forth in this document.
13. This document shall not be reprinted, reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
14. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products.

(Note1) “Renesas Electronics” as used in this document means Renesas Electronics Corporation and also includes its directly or indirectly controlled subsidiaries.
(Note2) “Renesas Electronics product(s)” means any product developed or manufactured by or for Renesas Electronics.

(Rev.5.0-1 October 2020)

Corporate Headquarters

TOYOSU FORESIA, 3-2-24 Toyosu,
Koto-ku, Tokyo 135-0061, Japan
www.renesas.com

Contact Information

For further information on a product, technology, the most up-to-date version of a document, or your nearest sales office, please visit:
www.renesas.com/contact/

Trademarks

Renesas and the Renesas logo are trademarks of Renesas Electronics Corporation. All trademarks and registered trademarks are the property of their respective owners.

© 2023 Renesas Electronics Corporation. All rights reserved.

General Precautions in the Handling of Microprocessing Unit and Microcontroller Unit Products

The following usage notes are applicable to all Microprocessing unit and Microcontroller unit products from Renesas. For detailed usage notes on the products covered by this document, refer to the relevant sections of the document as well as any technical updates that have been issued for the products.

1. Precaution against Electrostatic Discharge (ESD)
   A strong electrical field, when exposed to a CMOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop the generation of static electricity as much as possible, and quickly dissipate it when it occurs. Environmental control must be adequate. When it is dry, a humidifier should be used. This is recommended to avoid using insulators that can easily build up static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work benches and floors must be grounded. The operator must also be grounded using a wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions must be taken for printed circuit boards with mounted semiconductor devices.
2. Processing at power-on
   The state of the product is undefined at the time when power is supplied. The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the time when power is supplied. In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the time when power is supplied until the reset process is completed. In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the time when power is supplied until the power reaches the level at which resetting is specified.
3. Input of signal during power-off state
   Do not input signals or an I/O pull-up power supply while the device is powered off. The current injection that results from input of such a signal or I/O pull-up power supply may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements. Follow the guideline for input signal during power-off state as described in your product documentation.
4. Handling of unused pins
   Handle unused pins in accordance with the directions given under handling of unused pins in the manual. The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of the LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible.
5. Clock signals
   After applying a reset, only release the reset line after the operating clock signal becomes stable. When switching the clock signal during program execution, wait until the target clock signal is stabilized. When the clock signal is generated with an external resonator or from an external oscillator during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Additionally, when switching to a clock signal produced with an external resonator or by an external oscillator while program execution is in progress, wait until the target clock signal is stable.
6. Voltage application waveform at input pin
   Waveform distortion due to input noise or a reflected wave may cause malfunction. If the input of the CMOS device stays in the area between V_{IL} (Max.) and V_{IH} (Min.) due to noise, for example, the device may malfunction. Take care to prevent chattering noise from entering the device when the input level is fixed, and also in the transition period when the input level passes through the area between V_{IL} (Max.) and V_{IH} (Min.).
7. Prohibition of access to reserved addresses
   Access to reserved addresses is prohibited. The reserved addresses are provided for possible future expansion of functions. Do not access these addresses as the correct operation of the LSI is not guaranteed.
8. Differences between products
   Before changing from one product to another, for example to a product with a different part number, confirm that the change will not lead to problems. The characteristics of a microprocessing unit or microcontroller unit products in the same group but having a different part number might differ in terms of internal memory capacity, layout pattern, and other factors, which can affect the ranges of electrical characteristics, such as characteristic values, operating margins, immunity to noise, and amount of radiated noise. When changing to a product with a different part number, implement a system-evaluation test for the given product.

Preface

1. About this document

This manual is generally organized into an overview of the product, descriptions of the CPU, system control functions, peripheral functions, electrical characteristics, and usage notes. This manual describes the product specification of the microprocessors (MPUs) superset. Depending on your product, some pins, registers, or functions might not exist. Address space that store unavailable registers are reserved.

2. Audience

This manual is written for system designers who are designing and programming applications using the Renesas MPU. The user is expected to have basic knowledge of electrical circuits, logic circuits, and the MPU.

3. Renesas Publications

Renesas provides the following documents. Before using any of these documents, visit www.renesas.com for the most up-to-date version of the document.

Component	Document Type	Description
Hardware	Datasheet	Features, overview, and electrical characteristics
	User's Manual: Hardware	Specifications such as pin assignments, memory maps, peripheral functions, electrical characteristics, timing diagrams, and operation descriptions
	Application Notes	Technical notes, board design guidelines, and software migration information
	Technical Update (TU)	Preliminary reports on product specifications such as restriction and errata
Software	User's Manual: Software	API reference and programming information
	Application Notes	Project files, guidelines for software programming, and application examples to develop embedded software applications
Tools & Kits, Solutions	User's Manual: Development Tools	User's manuals and quick start guides for developing embedded software applications with Software Packages, Development Kits, Starter Kits, Promotion Kits, Product Examples, and Application Examples
	Quick Start Guide	
	Application Notes	Project files, guidelines for software programming, and application examples to develop embedded software applications

4. Numbering Notation

The following numbering notation is used throughout this manual:

Example	Description
011b	Binary number. For example, the binary equivalent of the number 3 is 011b.
0x1F	Hexadecimal number. For example, the hexadecimal equivalent of the number 31 is described 0x1F. In some cases, a hexadecimal number is shown with the suffix "h".
1234	Decimal number. A decimal number is followed by this symbol only when the possibility of confusion exists. Decimal numbers are generally shown without a suffix.

5. Typographic Notation

The following typographic notation is used throughout this manual:

Example	Description
AAA.BBB.CCC	Periods separated a function module symbol (AAA), register symbol (BBB), and bit field symbol (CCC).
AAA.BBB	A period separated a function module symbol (AAA) and register symbol (BBB).
BBB.DDD	A period separated a register symbol (BBB) and bit field symbol (DDD).
EEE[3:0]	Numbers in brackets expresses a bit number. For example, EEE[3:0] occupies bits 3 to 0.

6. Unit and Unit Prefix

The following units and unit prefixes are sometimes misleading. Those unit prefixes are described throughout this manual with the following meaning:

Symbol	Name	Description
b	Binary Digit	Single 0 or 1
B	Byte	This unit is generally used for memory specification and address space.
k	kilo-	$1000 = 10^3$. k is also used to denote 1024 (2^{10}) but this unit prefix is used to denote 1000 (10^3) throughout this manual.
K	Kilo-	$1024 = 2^{10}$. This unit prefix is used to denote 1024 (2^{10}) not 1000 (10^3) throughout this manual.

7. Special Terms

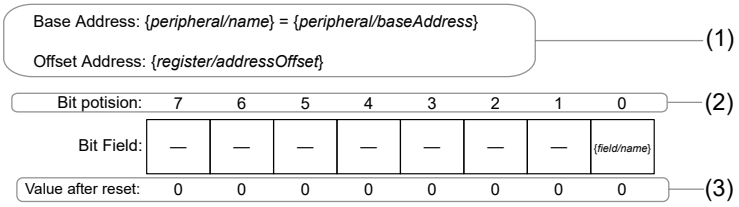
The following terms have special meanings.

Term	Description
NC	Not connected pin. This pin should be left floating unless specified otherwise.
Hi-Z	High impedance.
x	Don't care or undefined.

8. Register Description

Each register description includes both a register diagram that shows the bit assignments and a register bit table that describes the content of each bit. The example of symbols used in these tables are described in the sections that follow. The following is an example of a register description and associated bit field definition.

XX.X.X {*register/name*} : {*register/description*}



(4) (5) (6)

Bit	Symbol	Function	R/W
0	{*field/name*}	{*field/description*} 0: {*enumeratedValue/description*} 1: {*enumeratedValue/description*}	{*access*}
7:1	—	These bits are read as 0. The write value should be 0.	R/W

(1) Function module symbol, register symbol, and address assignment

Function module symbol, {*pheripheral/name*}, register symbol, {*register/name*}, and address assignment of this register are generally expressed. Base Address and Offset Address mean {*regiser/name*} : {*register/description*} of {*peripheral/name*} is assigned to address {*peripheral/baseAddress*} + {*register/addressOffset*}.

(2) Bit number

This number indicates the bit number. This bits are shown in order from bits 31 to 0 for 32-bit register, from bits 15 to 0 for 16-bit register, and from bits 7 to 0 for 8-bit register.

(3) Value after reset

This symbol or number indicate the value of each bit after a hard reset. The value is shown in binary unless specified otherwise.

- 0: Indicates that the value is 0 after a reset.
- 1: Indicates that the value is 1 after a reset.
- x: Indicates that the value is undefined after a reset.

(4) Symbol

{*filed/name*} indicates the short name of bit field. Reserved bit is expressed with a —.

(5) Function

Function indicates the full name of the bit field, {*field/description*}, and enumerated values.

(6) R/W

The R/W column indicates access type whether the bit field is readable or writable.

- R/W: The bit field is readable and writable.
- R: The bit field is readable only. Writing to this bit field has no effect.
- W: The bit field is writable only. The read value is the same as after a reset unless specified otherwise.

9. Abbreviations

Abbreviations used in this document are shown in the following table.

Abbreviation	Description
AES	Advanced Encryption Standard
AHB	Advanced High-performance Bus
AHB-AP	AHB Access Port
APB	Advanced Peripheral Bus
ATB	Advanced Trace Bus
AXI	Advanced eXtensible Interface
BCD	Binary Coded Decimal
BSDL	Boundary Scan Description Language
CTI	Cross Trigger Interface
DAP	Coresight Debug Access Port
DES	Data Encryption Standard
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
ETM	Embedded Trace Macrocell
JTAG	Joint Test Action Group
LSB	Least Significant Bit
MSB	Most Significant Bit
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
RSA	Rivest Shamir Adleman
S/H	Sample and Hold
SHA	Secure Hash Algorithm
SWD	Serial Wire Debug
TRNG	True Random Number Generator
UART	Universal Asynchronous Receiver/Transmitter

10. Proprietary Notice

All text, graphics, photographs, trademarks, logos, artwork and computer code, collectively known as content, contained in this document is owned, controlled or licensed by or to Renesas, and is protected by trade dress, copyright, patent and trademark laws, and other intellectual property rights and unfair competition laws. Except as expressly provided herein, no part of this document or content may be copied, reproduced, republished, posted, publicly displayed, encoded, translated, transmitted or distributed in any other medium for publication or distribution or for any commercial enterprise, without prior written consent from Renesas.

Arm® and Cortex® are registered trademarks of Arm Limited. CoreSight™ is a trademark of Arm Limited.

Magic Packet™ is a trademark of Advanced Micro Devices, Inc.

Other brands and names mentioned in this document may be the trademarks or registered trademarks of their respective holders.

11. Feedback on the product

If you have any comments or suggestions about this product, go to Contact Us.

Contents

Features **53**

1. Overview **54**

1.1 Outline of Specifications 54

1.2 Product Lineup 62

1.3 Block Diagram 63

1.4 Pin Functions 64

1.5 FBGA 196 Pin Assignments 70

2. CPU **78**

2.1 Overview 78

2.2 Configuration Information 78

2.3 Restrictions on the CPU 79

2.4 Register Descriptions 79

2.4.1 WAIT : ATCM Wait State Register 79

2.4.2 CNTCR : Global System Counter Control Register 79

2.4.3 CNTSR : Global System Counter Status Register 80

2.4.4 CNTCVL : Global System Counter Current Value Lower Register 80

2.4.5 CNTCVU : Global System Counter Current Value Upper Register 81

2.4.6 CNTFID0 : Global System Counter Base Frequency ID Register 81

2.5 Usage Notes 81

2.5.1 Global System Counter (TSG-CNT) 81

2.5.2 Caution of Interrupt Service Routine 81

3. Operating Modes **82**

3.1 Overview 82

3.2 Types and Selection of Operating Modes 82

3.3 Hardware Used in Individual Operating Modes 83

3.4 Register Descriptions 84

3.4.1 MD_MON : Operating Mode Monitor Register 84

3.5 Operating Mode Descriptions 85

3.5.1 Boot Function 85

3.5.2 Parameters for the Loader 86

3.5.3 Loader Program 90

3.5.4 xSPI Boot Mode (Serial Flash) 90

3.5.5 16-Bit Bus Boot Modes (NOR Flash Memory) 93

3.5.6 Host Interface Boot Mode 94

3.5.7 SCI (UART) Boot Mode 100

3.5.8 USB Boot Mode 100

3.5.9 Motorola S-record Format for SCI/USB Boot 101

3.5.10 SCI Boot Procedure 102
3.5.11 USB Boot Procedure 103
3.5.12 SCI/USB Boot Disable 105
3.5.13 Usage Note 105
4. Address Space 106
4.1 Address Space 106
5. Reset 110
5.1 Overview 110
5.2 Register Descriptions 111
5.2.1 RSTSR0 : Reset Status Register 0 111
5.2.2 SWRSYS : System Software Reset Register 112
5.2.3 SWRCPU0 : CPU0 Software Reset Register 112
5.2.4 MRCTLA : Module Reset Control Register A 113
5.2.5 MRCTLE : Module Reset Control Register E 114
5.2.6 MRCTLI : Module Reset Control Register I 115
5.2.7 MRCTLJ : Module Reset Control Register J 115
5.2.8 MRCTLK : Module Reset Control Register K 116
5.3 Operation 116
5.3.1 RES# Pin Reset 116
5.3.2 System Software Reset 117
5.3.3 CPU0 Software Reset 117
5.3.4 Error Reset 117
5.3.5 Determination of Reset Generation Source 117
5.3.6 Reset Output Pin (RSTOUT#) 118
5.3.7 Noise Cancellation for Reset Input 118
5.4 Usage Note 118
5.4.1 Notes on Module Reset Control Register Operation 118
5.4.2 Connection of Reset Output Pin (RSTOUT#) 118
6. Clock Generation Circuit 119
6.1 Overview 119
6.2 Register Descriptions 123
6.2.1 SCKCR : System Clock Control Register 123
6.2.2 SCKCR2 : System Clock Control Register 2 125
6.2.3 PLL0MON : PLL0 Monitor Register 126
6.2.4 PLL1MON : PLL1 Monitor Register 127
6.2.5 PLL1EN : PLL1 Enable Register 127
6.2.6 LOCOCR : Low-Speed On-Chip Oscillator Control Register 128
6.2.7 HIZCTRLEN : High-Impedance Control Enable Register 128
6.3 Selecting Input to Main Clock Oscillator 129

6.3.1 Connecting a Crystal Resonator 129
6.3.2 External Clock Input 130
6.4 PLL Circuit 130
6.5 Low-Speed On-Chip Oscillator (LOCO) 130
6.6 Oscillation Abnormality Detection Function for the Main Clock 130
6.6.1 Main Clock Oscillation Abnormality Detection and Operation after Detection 130
6.6.2 Main Clock Oscillation Abnormality Detection Interrupt 131
6.7 PLL Oscillation Abnormality Detection Function 131
6.8 Low-Speed On-Chip Oscillator Oscillation Abnormality Detection Function 131
6.9 Internal Clock 131
6.10 Reference Clock to the External Ethernet PHY 133
6.11 Usage Notes 134
6.11.1 Notes on Clock Generation Circuit 134
6.11.2 Notes on Resonator 134
6.11.3 Notes on Designing the Board 134

7. Clock Monitor Circuit (CLMA) 136
7.1 Overview 136
7.2 Register Descriptions 137
7.2.1 CTL0 : CLMA Control Register 0 137
7.2.2 CMPL : CLMA Compare Register L 138
7.2.3 CMPH : CLMA Compare Register H 138
7.2.4 PCMD : CLMA Command Register 139
7.2.5 PROTSR : CLMA Protection Status Register 139
7.3 Operation 139
7.3.1 CLMAm Operation 139
7.3.2 Detecting Error Clock Frequency 140
7.3.3 Detecting Error Clock Frequency 142
7.4 Notes on Using CLMAm 142

8. Low-Power Consumption Function 143
8.1 Overview 143
8.2 Register Descriptions 144
8.2.1 MSTPCRA : Module Stop Control Register A 144
8.2.2 MSTPCRB : Module Stop Control Register B 145
8.2.3 MSTPCRC : Module Stop Control Register C 146
8.2.4 MSTPCRD : Module Stop Control Register D 147
8.2.5 MSTPCRE : Module Stop Control Register E 148
8.2.6 MSTPCRF : Module Stop Control Register F 148
8.2.7 MSTPCRG : Module Stop Control Register G 149
8.2.8 MSTPCRI : Module Stop Control Register I 150

8.2.9 MSTPCRJ : Module Stop Control Register J 150
8.2.10 MSTPCRK : Module Stop Control Register K 151
8.2.11 MSTPCRL : Module Stop Control Register L 151
8.3 Operation 152
8.3.1 Module-Stop Function 152
8.3.2 Cortex-R52 Standby Mode 154
8.4 Usage Notes 154
8.4.1 I/O Port State 154
8.4.2 On-Chip Peripheral Module Interrupts in Module-Stop State 154
8.4.3 USB Low Power Consumption 154
8.4.4 Low Power Consumption for Ethernet-Related Functions 154
8.4.5 Low Power Consumption for HIPERFACE DSL and EnDat 2.2 154
8.4.6 Write Protection Function 155
9. Debugging Interface 156
9.1 Overview 156
9.2 Register Descriptions 158
9.2.1 OIRn : OCD Authentication ID Code Register n (n = 0 to 3) 158
9.2.2 OSR : OCD Status Register 159
9.3 Operation 159
9.3.1 JTAG Interface 159
9.3.2 SWD Interface 159
9.3.3 Trace Port Interface 160
9.3.4 Reset Configuration and the Method of Connecting with the Emulator 161
9.3.5 JTAG Pin Treatment When No Emulator Is Connected 162
9.3.6 Noise Reduction of the TRST# Pin 162
9.3.7 Available Trace Functions 162
9.3.8 Debug Authentication 162
9.3.9 Debug Authentication Procedure 163
9.4 Usage Notes 164
9.4.1 Access to the Main Bus 164
9.4.2 Authentication Level 164
9.4.3 Debug Reset 164
10. Register Write Protection Function 165
10.1 Overview 165
10.2 Register Descriptions 165
10.2.1 PRCRN : Non_Safety Area Protect Register 165
10.2.2 PRCRS : Safety Area Protect Register 166
11. Interrupt Controller (ICU) 168
11.1 Overview 168

11.2 Register Descriptions 169
11.2.1 Interrupt Control Unit Register 0 (Safety Region) 169
11.2.2 Interrupt Control Unit Register 1 (Non-Safety Region) 215
11.3 Operations 221
11.3.1 Event Handling 221
11.3.2 Event Table 221
11.3.3 Error Events 239
11.3.4 Noise Filter 247
12. Internal Buses 249
12.1 Overview 249
12.2 Register Descriptions 250
12.2.1 STADDm : Master MPU Safety Region Start Address Register m (m = 0 to 7) 250
12.2.2 ENDADDm : Master MPU Safety Region End Address Register m (m = 0 to 7) 251
12.2.3 ERRINF : Master MPU Error Information Register for AHB type 251
12.2.4 ERRINF_x : Master MPU Error Information Register for AXI type (x = R, W) 252
12.3 Operation 252
12.3.1 Bus Connection 252
12.3.2 Master MPU 253
12.3.3 Interrupt 254
12.4 Usage Notes 255
12.4.1 Release of Register Write Protection 255
12.4.2 Initial Setting of Master MPU 255
12.4.3 New MPU Error Detection While Preserving Valid Error Information 255
12.4.4 TCM Access via AXIS Interface of Cortex-R52 255
13. Bus State Controller 257
13.1 Overview 257
13.2 Area Overview 259
13.2.1 Address Map 259
13.2.2 Data Bus Width and Related Pin Setting for Each Area Depending on Boot Mode 259
13.3 Register Descriptions 260
13.3.1 CSnBCR : CSn Space Bus Control Register (n = 0, 2, 3, 5) 260
13.3.2 CSn Space Wait Control Register (CSnWCR) (n = 0, 2, 3, 5) 262
13.3.3 SDCR : SDRAM Control Register 271
13.3.4 RTCSR : Refresh Timer Control/Status Register 272
13.3.5 RTCNT : Refresh Timer Counter 273
13.3.6 RTCOR : Refresh Time Constant Register 274
13.3.7 TOSCORn : Timeout Cycle Constant Register n (n = 0, 2, 3, 5) 274
13.3.8 TOSTR : Timeout Status Register 275
13.3.9 TOENR : Timeout Enable Register 276

13.4 Operation 276
13.4.1 Access Size and Data Alignment 276
13.4.2 SRAM Interface 277
13.4.3 Access Wait Control 281
13.4.4 CSn# Assert Period Expansion 283
13.4.5 MPX-I/O Interface 284
13.4.6 SDRAM Interface 288
13.4.7 Burst ROM (Clocked Asynchronous) Interface 313
13.4.8 SRAM Interface with Byte Selection 315
13.4.9 Burst ROM (Clocked Synchronous) Interface 319
13.4.10 Wait between Access Cycles 320
13.4.11 Others 322

14. DMA Controller (DMAC) 324
14.1 Overview 324
14.2 Register Descriptions 325
14.2.1 NmSA_n : Nextm Source Address Register n (m = 0, 1) (n = 0 to 15) 325
14.2.2 NmDA_n : Nextm Destination Address Register n (m = 0, 1) (n = 0 to 15) 325
14.2.3 NmTB_n : Nextm Transaction Byte Register n (m = 0, 1) (n = 0 to 15) 325
14.2.4 CRSA_n : Current Source Address Register n (n = 0 to 15) 326
14.2.5 CRDA_n : Current Destination Address Register n (n = 0 to 15) 326
14.2.6 CRTB_n : Current Transaction Byte Register n (n = 0 to 15) 327
14.2.7 CHSTAT_n : Channel Status Register n (n = 0 to 15) 327
14.2.8 CHEXT_n : Channel Extension Register n (n = 0 to 15) 331
14.2.9 DMAC0_RSSELi : DMAC Unit 0 Resource Select Register i (i = 0 to 5) 331
14.2.10 DMAC1_RSSELi : DMAC Unit 1 Resource Select Register i (i = 0 to 5) 332
14.2.11 CMNCR : Common Control Register 333
14.2.12 CHCTRL_n : Channel Control Register n (n = 0 to 15) 334
14.2.13 CHCFG_n : Channel Configuration Register n (n = 0 to 15) 335
14.2.14 CHITVL_n : Channel Interval Register n (n = 0 to 15) 338
14.2.15 NXLA_n : Next Link Address Register n (n = 0 to 15) 338
14.2.16 CRLA_n : Current Link Address Register n (n = 0 to 15) 339
14.2.17 DCTRL_x : DMA Control Register x (x = A, B) 339
14.2.18 DSTAT_EN_x : DMA Status EN Register x (x = A, B) 340
14.2.19 DSTAT_ER_x : DMA Status ER Register x (x = A, B) 340
14.2.20 DSTAT_END_x : DMA Status END Register x (x = A, B) 341
14.2.21 DSTAT_SUS_x : DMA Status SUS Register x (x = A, B) 342
14.3 Operation 342
14.3.1 DMA Mode 342
14.3.2 Transfer Mode 353
14.3.3 DMA Channel Priority Control 354

14.3.4 DMA Transfer Request 357
14.3.5 DMA Acknowledge Output/DMA Transaction Completion Output Function 368
14.3.6 DMA Transaction Completion Signal Output 370
14.3.7 Interval Count Function 371
14.3.8 Differences in Operation According to the Transfer Data Size 372
14.3.9 DMA Transfer Status 373
14.3.10 Suspending a Transfer 374
14.3.11 Aborting a Transfer 375
14.4 Interrupts 378
14.4.1 Interrupt Sources 378
14.4.2 DMA Transfer Completion Interrupts 379
14.4.3 DMA Error Interrupt 379
14.5 DMA Setting Examples 379
14.5.1 Setting Example 1 (Register Hardware Request) 380
14.5.2 Setting Example 2 (Register Mode Software Request) 381
14.5.3 Setting Example 3 (Register Mode Continuous Execution) 383
14.5.4 Setting Example 4 (Link Mode) 385
14.5.5 Next Register Continuous Execution Settings 388
14.6 Usage Notes 390
14.6.1 When the DACK Signal is Divided and Output 390
15. Event Link Controller (ELC) 392
15.1 Overview 392
15.2 Register Descriptions 392
15.2.1 ELC_SSELn : ELC Event Source Select Register n (n = 0 to 18) 393
15.2.2 ELOPA : Event Link Option Setting Register A 399
15.2.3 ELOPB : Event Link Option Setting Register B 400
15.2.4 ELC_PGRn : ELC Port Group Setting Register n (n = 1, 2) 400
15.2.5 ELC_PGCn : ELC Port Group Control Register n (n = 1, 2) 401
15.2.6 ELC_PDBFn : ELC Port Buffer Register n (n = 1, 2) 401
15.2.7 ELC_PELn : ELC Port Setting Register n (n = 0 to 3) 402
15.2.8 ELC_DPTC : ELC Edge Detection Control Register 402
15.2.9 ELC_ELSR2 : ELC Port Event Control Register 403
15.3 Operation 403
15.3.1 Event Linkage 403
15.3.2 Operation of MTU When Event is Input 405
15.3.3 Operation of GPT When Event is Input 405
15.3.4 Operation of A/D Converter When Event is Input 405
15.3.5 I/O Port Operation on Event Input and Event Generation 405
15.3.6 Example of Procedure for Linking Events 408
15.4 Usage Notes 409

15.4.1 Setting Clocks 409

16. I/O Ports 410

16.1 Overview 410

16.2 I/O Port Configuration 410

16.3 Register Descriptions 433

16.3.1 Pm : Port m Register (m = 00 to 24) 433

16.3.2 PMm : Port m Mode Register (m = 00 to 24) 434

16.3.3 PMCm : Port m Mode Control Register (m = 00 to 24) 434

16.3.4 PFCm : Port m Function Control Register (m = 00 to 24) 435

16.3.5 PINm : Port m Input Register (m = 00 to 24) 435

16.3.6 DRCTLm : I/O Buffer m Function Switching Register (m = 00 to 24) 436

16.3.7 RSELPm : Port m Region Select Register (m = 00 to 24) 438

16.4 Multiplexed Pin Configurations 438

16.4.1 Port 00 Function Assignment 438

16.4.2 Port 01 Function Assignment 439

16.4.3 Port 02 Function Assignment 439

16.4.4 Port 04 Function Assignment 440

16.4.5 Port 05 Function Assignment 440

16.4.6 Port 06 Function Assignment 440

16.4.7 Port 07 Function Assignment 441

16.4.8 Port 08 Function Assignment 441

16.4.9 Port 09 Function Assignment 442

16.4.10 Port 10 Function Assignment 442

16.4.11 Port 13 Function Assignment 443

16.4.12 Port 14 Function Assignment 443

16.4.13 Port 15 Function Assignment 444

16.4.14 Port 16 Function Assignment 444

16.4.15 Port 17 Function Assignment 444

16.4.16 Port 18 Function Assignment 445

16.4.17 Port 20 Function Assignment 445

16.4.18 Port 21 Function Assignment 446

16.4.19 Port 22 Function Assignment 446

16.4.20 Port 23 Function Assignment 447

16.4.21 Port 24 Function Assignment 447

16.5 Handling of Unused Pins 447

16.6 Usage Notes 448

16.6.1 Release of Register Write Protection 448

16.6.2 Examples of Port Setting 448

16.6.3 Notes on Port Register Setting 450

16.6.4 Pin Assignment of each Boot Mode 450

17. Multi-Function Timer Pulse Unit 3 (MTU3) 453
17.1 Overview 453
17.2 Register Descriptions 460
17.2.1 TCR : Timer Control Register 460
17.2.2 TCRx : Timer Control Register x (x = U, V, W) 462
17.2.3 TCR2 — Timer Control Registers 462
17.2.4 TMDR1 : Timer Mode Register 1 468
17.2.5 TMDR2x : Timer Mode Register 2x (x = A, B) 470
17.2.6 TMDR3 : Timer Mode Register 3 470
17.2.7 TIOR — Timer I/O Control Registers 472
17.2.8 TCNTCMPCLR : Timer Compare Match Clear Register 490
17.2.9 TIER — Timer Interrupt Enable Registers 491
17.2.10 TSR : Timer Status Register 494
17.2.11 TBTM : Timer Buffer Operation Transfer Mode Register 496
17.2.12 TICCR : Timer Input Capture Control Register 497
17.2.13 TSYCR : Timer Synchronous Clear Register 497
17.2.14 TCNT — Timer Counters 498
17.2.15 TCNTLW : Timer Longword Counter 499
17.2.16 TGR — Timer General Registers 499
17.2.17 TGRxLW : Timer Longword General Register x (x = A, B) 501
17.2.18 TSTR — Timer Start Registers 502
17.2.19 TSYRA : Timer Synchronous Register A 503
17.2.20 TSYRB : Timer Synchronous Register B 504
17.2.21 TCSYSTR : Timer Counter Synchronous Start Register 505
17.2.22 TRWERx : Timer Read/Write Enable Register x (x = A, B) 506
17.2.23 TOERA : Timer Output Master Enable Register A 507
17.2.24 TOERB : Timer Output Master Enable Register B 508
17.2.25 TOCR1x : Timer Output Control Register 1x (x = A, B) 508
17.2.26 TOCR2x : Timer Output Control Register 2x (x = A, B) 510
17.2.27 TOLBRx : Timer Output Level Buffer Register x (x = A, B) 512
17.2.28 TGCRA : Timer Gate Control Register A 513
17.2.29 TCNTSx : Timer Subcounter x (x = A, B) 515
17.2.30 TCDRx : Timer Cycle Data Register x (x = A, B) 515
17.2.31 TCBRx : Timer Cycle Buffer Register x (x = A, B) 515
17.2.32 TDDRx : Timer Dead Time Data Register x (x = A, B) 516
17.2.33 TDERx : Timer Dead Time Enable Register x (x = A, B) 516
17.2.34 TBTERx : Timer Buffer Transfer Set Register x (x = A, B) 516
17.2.35 TWCRx : Timer Waveform Control Register x (x = A, B) 517
17.2.36 NFCR — Noise Filter Control Registers 518
17.2.37 TADCR — Timer A/D Converter Start Request Control Registers 522

17.2.38 TADCORx : Timer A/D Converter Start Request Cycle Set Register x (x = A, B) 525
17.2.39 TADCOBRx : Timer A/D Converter Start Request Cycle Set Buffer Register x (x = A, B) .. 526
17.2.40 TITMRx : Timer Interrupt Skipping Mode Register x (x = A, B) 526
17.2.41 TITCR1A : Timer Interrupt Skipping Set Register 1A 527
17.2.42 TITCR1B : Timer Interrupt Skipping Set Register 1B 527
17.2.43 TITCNT1A : Timer Interrupt Skipping Counter 1A 528
17.2.44 TITCNT1B : Timer Interrupt Skipping Counter 1B 529
17.2.45 TITCR2A : Timer Interrupt Skipping Set Register 2A 530
17.2.46 TITCR2B : Timer Interrupt Skipping Set Register 2B 530
17.2.47 TITCNT2A : Timer Interrupt Skipping Counter 2A 531
17.2.48 TITCNT2B : Timer Interrupt Skipping Counter 2B 531
17.2.49 Bus Master Interface 532
17.3 Operations 532
17.3.1 Basic Functions 532
17.3.2 Synchronous Operation 536
17.3.3 Buffer Operation 538
17.3.4 Cascaded Operation 541
17.3.5 PWM Modes 545
17.3.6 Phase Counting Mode 549
17.3.7 Reset-Synchronized PWM Mode 561
17.3.8 Complementary PWM Mode 564
17.3.9 A/D Converter Start Request Delaying Function 599
17.3.10 Synchronous Operation of MTU0 to MTU4, MTU6, and MTU7 605
17.3.11 External Pulse Width Measurement 608
17.3.12 Dead Time Compensation 609
17.3.13 TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode 611
17.3.14 Noise Filter Function 611
17.4 Interrupt Sources 612
17.4.1 Interrupt Sources and Priorities 612
17.4.2 DMAC Trigger Sources 614
17.4.3 A/D Converter Trigger Sources 614
17.5 Operation Timing 615
17.5.1 Input/Output Timing 615
17.5.2 Interrupt Signal Timing 621
17.6 Usage Notes 623
17.6.1 Module Stop Function Setting 623
17.6.2 Input Clock Restrictions 623
17.6.3 Note on Period Setting 624
17.6.4 Contention between TCNT Write and Clear Operations 624

17.6.5 Contention between TCNT Write and Increment Operations 624
17.6.6 Contention between TGR Write Operation and Compare Match 625
17.6.7 Contention between Buffer Register Write Operation and Compare Match 625
17.6.8 Contention between Buffer Register Write and TCNT Clear Operations 626
17.6.9 Contention between TGR Read Operation and Input Capture 626
17.6.10 Contention between TGR Write Operation and Input Capture 627
17.6.11 Contention between Buffer Register Write Operation and Input Capture 627
17.6.12 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation 628
17.6.13 Counter Value When Count Operation is Stopped in Complementary PWM Mode 629
17.6.14 Buffer Operation Setting in Complementary PWM Mode 630
17.6.15 Buffer Operation and Compare Match in Reset-Synchronized PWM Mode 630
17.6.16 Overflow in Reset-Synchronized PWM Mode 631
17.6.17 Contention between Overflow/Underflow and Counter Clearing 631
17.6.18 Contention between TCNT Write Operation and Overflow/Underflow 632
17.6.19 Note on Transition from Normal Mode or PWM Mode 1 to Reset-Synchronized PWM Mode 632
17.6.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode 633
17.6.21 Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection 633
17.6.22 Interrupt Skipping Function 2 633
17.6.23 Notes When Complementary PWM Mode Output Protection Function is Not Used 633
17.6.24 Notes Regarding Timer Counter (MTU5.TCNT) and Timer General Register (MTU5.TGR) 633
17.6.25 Notes to Prevent Malfunctions in Synchronous Clearing for Complementary PWM Mode 634
17.6.26 Notes on Timer Mode Register Setting for ELC Event Input 635
17.6.27 Continuous Output of Interrupt Signal in Response to a Compare Match 635
17.7 MTU Output Pin Initialization 636
17.7.1 Operating Modes 636
17.7.2 Operation in Case of Re-Setting Due to Error during Operation 636
17.7.3 Overview of Initialization Procedures and Mode Transitions in Case of Error During Operation 637
17.8 Operations Linked by the ELC 655
17.8.1 Event Signal Output to the ELC 655
17.8.2 MTU Operations in Response to Receiving Event Signals from the ELC 655
17.8.3 Usage Notes on MTU Operation by Event Signal Reception from the ELC 655

18. Port Output Enable 3 (POE3) 657

18.1 Overview 657
18.2 Register Descriptions 659
18.2.1 ICSR1 : Input Level Control/Status Register 1 659
18.2.2 ICSR2 : Input Level Control/Status Register 2 660
18.2.3 ICSR3 : Input Level Control/Status Register 3 661

18.2.4 ICSR4 : Input Level Control/Status Register 4 662
18.2.5 ICSR5 : Input Level Control/Status Register 5 663
18.2.6 ICSR6 : Input Level Control/Status Register 6 664
18.2.7 ICSR7 : Input Level Control/Status Register 7 665
18.2.8 OCSR1 : Output Level Control/Status Register 1 666
18.2.9 OCSR2 : Output Level Control/Status Register 2 667
18.2.10 ALR1 : Active Level Setting Register 1 668
18.2.11 SPOER : Software Port Output Enable Register 669
18.2.12 POECR1 : Port Output Enable Control Register 1 670
18.2.13 POECR2 : Port Output Enable Control Register 2 671
18.2.14 POECR4 : Port Output Enable Control Register 4 672
18.2.15 POECR5 : Port Output Enable Control Register 5 675
18.2.16 M0SELR1 : MTU0 Pin Select Register 1 676
18.2.17 M0SELR2 : MTU0 Pin Select Register 2 677
18.2.18 M3SELR : MTU3 Pin Select Register 677
18.2.19 M4SELR1 : MTU4 Pin Select Register 1 678
18.2.20 M4SELR2 : MTU4 Pin Select Register 2 678
18.2.21 M6SELR : MTU6 Pin Select Register 679
18.2.22 M7SELR1 : MTU7 Pin Select Register 1 680
18.2.23 M7SELR2 : MTU7 Pin Select Register 2 680
18.3 Operation 681
18.3.1 MTU3 Pin Selection 684
18.3.2 Input-Level Detection Operation 684
18.3.3 Output-Level Compare Operation 685
18.3.4 High-Impedance Control Using Registers 686
18.3.5 High-Impedance Control through Detection of Stopped Oscillation and Oscillation Abnormality 686
18.3.6 Additional Functions for High-Impedance Control 686
18.3.7 Recover from High-Impedance State 686
18.4 POE3 Setting Procedure 686
18.5 Interrupts 687
18.6 Usage Notes 687
18.6.1 High-Impedance Control when the MTU is not Selected 687
19. General PWM Timer (GPT) 688
19.1 Overview 688
19.2 Register Descriptions 692
19.2.1 GTWP : General PWM Timer Write-Protection Register 692
19.2.2 GTSTR : General PWM Timer Software Start Register 692
19.2.3 GTSTP : General PWM Timer Software Stop Register 694
19.2.4 GTCLR : General PWM Timer Software Clear Register 695

19.2.5 GTSSR : General PWM Timer Start Source Select Register 696
19.2.6 GTPSR : General PWM Timer Stop Source Select Register 698
19.2.7 GTCSR : General PWM Timer Clear Source Select Register 699
19.2.8 GTUPSR : General PWM Timer Count-Up Source Select Register 701
19.2.9 GTDNSR : General PWM Timer Count-Down Source Select Register 703
19.2.10 GTICASR : General PWM Timer Input Capture Source Select Register A 704
19.2.11 GTICBSR : General PWM Timer Input Capture Source Select Register B 707
19.2.12 GTCR : General PWM Timer Control Register 709
19.2.13 GTUDDTYC : General PWM Timer Count Direction and Duty Setting Register 710
19.2.14 GTIOR : General PWM Timer I/O Control Register 711
19.2.15 GTINTAD : General PWM Timer Interrupt Output Setting Register 714
19.2.16 GTST : General PWM Timer Status Register 715
19.2.17 GTBER : General PWM Timer Buffer Enable Register 716
19.2.18 GTITC : General PWM Timer Interrupt and A/D Converter Start Request Skipping Setting Register 718
19.2.19 GTCNT : General PWM Timer Counter 719
19.2.20 GTCCRm : General PWM Timer Compare Capture Register m (m = A to F) 720
19.2.21 GTPR : General PWM Timer Cycle Setting Register 720
19.2.22 GTPBR : General PWM Timer Cycle Setting Buffer Register 720
19.2.23 GTPDBR : General PWM Timer Cycle Setting Double-Buffer Register 721
19.2.24 GTADTRm : A/D Converter Start Request Timing Register m (m = A, B) 721
19.2.25 GTADTBRm : A/D Converter Start Request Timing Buffer Register m (m = A, B) 721
19.2.26 GTADTDBRm : A/D Converter Start Request Timing Double-Buffer Register m (m = A, B) 722
19.2.27 GTDTCR : General PWM Timer Dead Time Control Register 722
19.2.28 GTDVm : General PWM Timer Dead Time Value Register m (m = U, D) 723
19.2.29 GTDBm : General PWM Timer Dead Time Value Buffer Register m (m = U, D) 723
19.2.30 GTSOS : General PWM Timer Output Protection Function Status Register 724
19.2.31 GTSOTR : General PWM Timer Output Protection Function Temporary Release Register 724
19.2.32 GTADSMR : General PWM Timer A/D Conversion Start Request Signal Monitoring Register 725
19.2.33 GTEITC : General PWM Timer Extended Interrupt Skipping Counter Control Register 726
19.2.34 GTEITLI1 : General PWM Timer Extended Interrupt Skipping Setting Register 1 727
19.2.35 GTEITLI2 : General PWM Timer Extended Interrupt Skipping Setting Register 2 728
19.2.36 GTEITLB : General PWM Timer Extended Buffer Transfer Skipping Setting Register 729
19.2.37 GTSECSR : General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register 731
19.2.38 GTSECR : General PWM Timer Operation Enable Bit Simultaneous Control Register 732
19.2.39 GTSWSR : General PWM Timer Switch Source Select Register 733
19.2.40 GTSWOS : General PWM Timer Switch Offset Setting Register 734
19.3 Operation 734

19.3.1 Basic Operation ... 734
19.3.2 Buffer Operation ... 742
19.3.3 PWM Output Operating Mode ... 752
19.3.4 Automatic Dead Time Setting Function ... 762
19.3.5 Count Direction Changing Function ... 766
19.3.6 Function of Output Duty 0% and 100% ... 767
19.3.7 Hardware Count Start/Count Stop and Clear Operation ... 769
19.3.8 Hardware Count Switch Operation ... 774
19.3.9 Synchronous Operation ... 782
19.3.10 PWM Output Operation Examples ... 785
19.3.11 Phase Counting Mode ... 791
19.4 Interrupt Sources ... 799
19.4.1 Interrupt Sources ... 799
19.4.2 Interrupt and A/D Conversion Start Request Skipping Function ... 802
19.5 A/D Converter Start Request ... 807
19.6 Operations Linked by the ELC ... 809
19.6.1 Event Signal Output to the ELC ... 809
19.6.2 Operations in Response to Receiving Event Signals from the ELC ... 810
19.7 Noise Filter Function ... 810
19.8 Protection Function ... 810
19.8.1 Write-Protection for Registers ... 810
19.8.2 Disabling of Buffer Operation ... 811
19.8.3 GTIOCnm Pin Output Negate Control (n = 0 to 17, m = A, B) ... 816
19.8.4 GTIOCnm Protection Function for GTIOCnm Pin Output (n = 0 to 17, m = A, B) ... 817
19.9 Initialization Method of Output Pins ... 824
19.9.1 Pin Settings after Reset ... 824
19.9.2 Pin Initialization Due to Error during Operation ... 825
19.10 Usage Notes ... 825
19.10.1 Module-Stop Function Setting ... 825
19.10.2 Settings of the GTCCRn Register During Compare Match Operation (n = A to F) ... 825
19.10.3 Setting Range of the GTCNT Counter ... 827
19.10.4 The GTCNT Counter Start/Stop ... 827
19.10.5 Order of Priority in Events ... 827

20. Port Output Enable for GPT (POEG) ... 829

20.1 Overview ... 829
20.2 Register Descriptions ... 834
20.2.1 POEG0Gn : POEG0 Group n Setting Register (n = A to D) ... 834
20.2.2 POEGmGn : POEGm Group n Setting Register (m = 1, 2) (n = A to D) ... 836
20.3 Operation ... 837
20.3.1 Output-Disable Control Operation ... 837

20.3.2 Output-Disable Requests from the GPT 838
20.3.3 Output-Disable Control Using Detection of Stopped Oscillation 838
20.3.4 Output-Disable Control Using DSMIF Error Detection 838
20.3.5 Output-Disable Control Using Registers 838
20.3.6 Release from Output-Disable 838

21. Trigonometric Function Unit (TFU) 840
21.1 Overview 840
21.1.1 Precautions on Use of the Trigonometric Function Unit 841
21.2 Register Descriptions 841
21.2.1 FXSCIOC : Fixed-point Sine Cosine Input/Output Setting Register 841
21.2.2 FXATIOC : Fixed-point Arctangent hypot_k Input/Output Setting Register 842
21.2.3 TRGSTS : Trigonometric Status Register 842
21.2.4 FPSCDT0 : Floating-point Sine Cosine Data Register 0 843
21.2.5 FPSCDT1 : Floating-point Sine Cosine Data Register 1 844
21.2.6 FPATDT0 : Floating-point Arctangent hypot_k Data Register 0 844
21.2.7 FPATDT1 : Floating-point Arctangent hypot_k Data Register 1 845
21.2.8 FXSCDT0 : Fixed-point Sine Cosine Data Register 0 845
21.2.9 FXSCDT1 : Fixed-point Sine Cosine Data Register 1 846
21.2.10 FXATDT0 : Fixed-point Arctangent hypot_k Data Register 0 846
21.2.11 FXATDT1 : Fixed-point Arctangent hypot_k Data Register 1 847
21.2.12 DTSR0 : Data Save Restore Register 0 847
21.2.13 DTSR1 : Data Save Restore Register 1 848
21.3 Operation 848
21.3.1 Arithmetic Processing 848
21.3.2 Input and Output Value Formats 848
21.3.3 Angle Unit 849
21.3.4 Relationship Between Input and Output Values for Sincos Operation 850
21.3.5 Relationship Between Input and Output Values for Atan Operation 850
21.3.6 Relationship Between Input and Output Values for hypot_k Operation 851
21.3.7 Procedure for Trigonometric Function Operation 851
21.3.8 Usage of Interrupt 853

22. Compare Match Timer (CMT) 855
22.1 Overview 855
22.2 Register Descriptions 856
22.2.1 CMSTRn : Compare Match Timer Start Register (n = 0, 2, 4) 856
22.2.2 CMCR : Compare Match Timer Control Register 856
22.2.3 CMCNT : Compare Match Timer Counter Register 857
22.2.4 CMCOR : Compare Match Timer Constant Register 857
22.3 Operation 857

22.3.1 Periodic Count Operation 857
22.3.2 CMCNT Count Timing 858
22.4 Interrupts 858
22.4.1 Interrupt Sources 858
22.4.2 Timing of Compare Match Interrupt Generation 859
22.5 Usage Notes 859
22.5.1 Setting the Module Stop Function 859
22.5.2 Conflict between Write and Compare-Match Processes of CMCNT 859
22.5.3 Conflict between Write and Count-Up Processes of CMCNT 860
23. Compare Match Timer W (CMTW) 861
23.1 Overview 861
23.2 Register Descriptions 862
23.2.1 CMWSTR : Timer Start Register 862
23.2.2 CMWCR : Timer Control Register 862
23.2.3 CMWIOR : Timer I/O Control Register 864
23.2.4 CMWCNT : Timer Counter 865
23.2.5 CMWCOR : Compare Match Constant Register 865
23.2.6 CMWICRn : Input Capture Registers (n = 0, 1) 866
23.2.7 CMWOCRn : Output Compare Registers (n = 0, 1) 866
23.3 Operation 866
23.3.1 Period Counting Operation 866
23.3.2 Compare Match Function 867
23.3.3 Output Compare Function 868
23.3.4 Input Capture Function 869
23.3.5 Counter Size 870
23.3.6 Count Timing Based on CMWCNT 870
23.3.7 Output Compare Output Timing 870
23.3.8 Input Capture Signal Timing 871
23.4 Interrupts 871
23.4.1 CMTW Interrupt Sources and DMAC Transfer Requests 871
23.4.2 Timing of Compare Match Interrupt Generation 872
23.4.3 Timing of Output Compare Interrupt Generation 872
23.4.4 Timing of Input Capture Interrupt Generation 873
23.5 Usage Notes 873
23.5.1 Module-Stop Function 873
23.5.2 Contention Between CMWCNT Counter Writing and Compare Match 873
23.5.3 Contention Between CMWCNT Counter Writing and Incrementing or Clearing 874
23.5.4 Contention Between CMWCOR Register Writing and Compare Match 874
23.5.5 Contention Between CMWOCRn Register Writing and Compare Match 875
23.5.6 Contention Between CMWCNT Counter Reading and Incrementing or Clearing 875

23.5.7 Contention Between CMWICRn Register Reading and Input Capture 876

24. Watchdog Timer (WDT) 878

24.1 Overview 878

24.2 Register Descriptions 879

24.2.1 WDTRR : WDT Refresh Register 879

24.2.2 WDTCR : WDT Control Register 879

24.2.3 WDTSR : WDT Status Register 882

24.2.4 WDTRCR : WDT Reset Control Register 883

24.3 Operation 884

24.3.1 Count Operation in Start Mode 884

24.3.2 Control over Writing to the WDTCR and WDTRCR Registers 885

24.3.3 Refresh Operation 885

24.3.4 Status Flag 886

25. Real Time Clock (RTC) 887

25.1 Overview 887

25.2 Register Descriptions 888

25.2.1 RTCA0CTL0 : RTC Control Register 0 888

25.2.2 RTCA0CTL1 : RTC Control Register 1 889

25.2.3 RTCA0CTL2 : RTC Control Register 2 889

25.2.4 RTCA0SUBC : RTC Sub Count Register 890

25.2.5 RTCA0SRBU : RTC Sub Count Register Read Buffer 891

25.2.6 RTCA0SEC : RTC Sec Count Buffer Register 891

25.2.7 RTCA0MIN : RTC Min Count Buffer Register 892

25.2.8 RTCA0HOUR : RTC Hour Count Buffer Register 892

25.2.9 RTCA0WEEK : RTC Week Count Buffer Register 893

25.2.10 RTCA0DAY : RTC Day Count Buffer Register 893

25.2.11 RTCA0MONTH : RTC Month Count Buffer Register 893

25.2.12 RTCA0YEAR : RTC Year Count Buffer Register 894

25.2.13 RTCA0TIME : RTC Time Set Register 894

25.2.14 RTCA0CAL : RTC Calendar Set Register 894

25.2.15 RTCA0SCMP : RTC Sub Count Compare Register 895

25.2.16 RTCA0ALM : RTC Alarm Min Set Register 895

25.2.17 RTCA0ALH : RTC Alarm Hour Set Register 896

25.2.18 RTCA0ALW : RTC Alarm Week Set Register 896

25.2.19 RTCA0SECC : RTC Second Count Register 897

25.2.20 RTCA0MINC : RTC Minute Count Register 897

25.2.21 RTCA0HOURC : RTC Hour Count Register 897

25.2.22 RTCA0WEEKC : RTC Week Count Register 898

25.2.23 RTCA0DAYC : RTC Day Count Register 898

25.2.24 RTCA0MONC : RTC Month Count Register 898
25.2.25 RTCA0YEARC : RTC Year Count Register 899
25.2.26 RTCA0TIMEC : RTC Time Count Register 899
25.2.27 RTCA0CALC : RTC Calendar Count Register 899
25.3 Operation 900
25.3.1 Programming RTC 900
26. Ethernet Subsystem 909
26.1 Overview 909
26.2 Register Descriptions 911
26.2.1 PRCMD : Ethernet Protect Register 911
26.2.2 MODCTRL : Mode Control Register 912
26.2.3 PTPMCTRL : PTP Mode Control Register 913
26.2.4 PHYLNK : Ethernet PHY Link Mode Register 913
26.2.5 CONVCTRLn : RGMII/RMII Converter n Control Register (n = 0 to 2) 914
26.2.6 CONVRST : RGMII/RMII Converter Reset Control Register 915
26.3 Operation 915
26.3.1 Ethernet Initializing Operation 915
27. Ethernet MAC (GMAC) 917
27.1 Overview 917
27.2 Register Map 919
27.2.1 GMAC register map 919
27.3 Register Description 922
27.3.1 MAC_Configuration : MAC Configuration Register 922
27.3.2 MAC_Frame_Filter : MAC Frame Filter Register 925
27.3.3 GMII_Address : GMII Address Register 927
27.3.4 GMII_Data : GMII Data Register 928
27.3.5 Flow_Control : Flow Control Register 929
27.3.6 VLAN_Tag : VLAN Tag Register 930
27.3.7 Version : Version Register 931
27.3.8 Debug : Debug Register 931
27.3.9 Remote_Wake_Up_Frame_Filter : Remote Wake-Up Frame Filter Register 932
27.3.10 PMT_Control_Status : PMT Control and Status Register 933
27.3.11 LPI_Control_Status : LPI Control and Status Register 934
27.3.12 LPI_Timers_Control : LPI Timers Control Register 935
27.3.13 Interrupt_Status : Interrupt Status Register 935
27.3.14 Interrupt_Mask : Interrupt Mask Register 936
27.3.15 MAR0_H : MAC Address 0 High Register 937
27.3.16 MAR0_L : MAC Address 0 Low Register 937
27.3.17 MARn_H : MAC ADDRESS High Register (n = 1 to 17) 937

27.3.18 MARn_L : MAC ADDRESS Low Register (n = 1 to 17) 938
27.3.19 WDog_Timeout : Watchdog Timeout Register 938
27.3.20 MMC_Control : MMC Control Register 939
27.3.21 MMC_Receive_Interrupt : MMC Receive Interrupt Register 940
27.3.22 MMC_Transmit_Interrupt : MMC Transmit Interrupt Register 942
27.3.23 MMC_Receive_Interrupt_Mask : MMC Receive Interrupt Mask Register 944
27.3.24 MMC_Transmit_Interrupt_Mask : MMC Transmit Interrupt Mask Register 946
27.3.25 Tx_Octet_Count_Good_Bad : Transmit Octet Count for Good and Bad Frames 947
27.3.26 Tx_Frame_Count_Good_Bad : Transmit Frame Count for Good and Bad Frames 948
27.3.27 Tx_Broadcast_Frames_Good : Transmit Frame Count for Good Broadcast Frames 948
27.3.28 Tx_Multicast_Frames_Good : Transmit Frame Count for Good Multicast Frames 948
27.3.29 Tx_64Octets_Frames_Good_Bad : Transmit Octet Count for Good and Bad 64 Byte Frames 949
27.3.30 Tx_65To127Octets_Frames_Good_Bad : Transmit Octet Count for Good and Bad 65 to 127 Bytes Frames 949
27.3.31 Tx_128To255Octets_Frames_Good_Bad : Transmit Octet Count for Good and Bad 128 to 255 Bytes Frames 949
27.3.32 Tx_256To511Octets_Frames_Good_Bad : Transmit Octet Count for Good and Bad 256 to 511 Bytes Frames 950
27.3.33 Tx_512To1023Octets_Frames_Good_Bad : Transmit Octet Count for Good and Bad 512 to 1023 Bytes Frames 950
27.3.34 Tx_1024ToMaxOctets_Frames_Good_Bad : Transmit Octet Count for Good and Bad 1024 to Maxsize Bytes Frames 950
27.3.35 Tx_Unicast_Frames_Good_Bad : Transmit Frame Count for Good and Bad Unicast Frames 951
27.3.36 Tx_Multicast_Frames_Good_Bad : Transmit Frame Count for Good and Bad Multicast Frames 951
27.3.37 Tx_Broadcast_Frames_Good_Bad : Transmit Frame Count for Good and Bad Broadcast Frames 951
27.3.38 Tx_Underflow_Error_Frames : Transmit Frame Count for Underflow Error Frames 951
27.3.39 Tx_Single_Collision_Good_Frames : Transmit Frame Count for Frames Transmitted after Single Collision 952
27.3.40 Tx_Multiple_Collision_Good_Frames : Transmit Frame Count for Frames Transmitted after Multiple Collision 952
27.3.41 Tx_Deferred_Frames : Transmit Frame Count for Deferred Frames 952
27.3.42 Tx_Late_Collision_Frames : Transmit Frame Count for Late Collision Error Frames 953
27.3.43 Tx_Excessive_Collision_Frames : Transmit Frame Count for Excessive Collision Error Frames 953
27.3.44 Tx_Carrier_Error_Frames : Transmit Frame Count for Carrier Sense Error Frames 953
27.3.45 Tx_Octet_Count_Good : Transmit Octet Count for Good Frames 954
27.3.46 Tx_Frame_Count_Good : Transmit Frame Count for Good Frames 954
27.3.47 Tx_Excessive_Deferral_Error : Transmit Frame Count for Excessive Deferral Error Frames 954
27.3.48 Tx_Pause_Frames : Transmit Frame Count for Good PAUSE Frames 954

27.3.49 Tx_VLAN_Frames_Good : Transmit Frame Count for Good VLAN Frames 955
27.3.50 Tx_OSize_Frames_Good : Transmit Frame Count for Good Oversize Frames 955
27.3.51 Rx_Frames_Count_Good_Bad : Receive Frame Count for Good and Bad Frames 955
27.3.52 Rx_Octet_Count_Good_Bad : Receive Octet Count for Good and Bad Frames..... 956
27.3.53 Rx_Octet_Count_Good : Receive Octet Count for Good Frames 956
27.3.54 Rx_Broadcast_Frames_Good : Receive Frame Count for Good Broadcast Frames 956
27.3.55 Rx_Multicast_Frames_Good : Receive Frame Count for Good Multicast Frames 956
27.3.56 Rx_CRC_Error_Frames : Receive Frame Count for CRC Error Frames..... 957
27.3.57 Rx_Alignment_Error_Frames : Receive Frame Count for Alignment Error Frames 957
27.3.58 Rx_Runt_Error_Frames : Receive Frame Count for Runt Error Frames 957
27.3.59 Rx_Jabber_Error_Frames : Receive Frame Count for Jabber Error Frames 957
27.3.60 Rx_Undersize_Frames_Good : Receive Frame Count for Undersize Frames 958
27.3.61 Rx_Oversize_Frames_Good : Receive Frame Count for Oversize Frames 958
27.3.62 Rx_64Octets_Frames_Good_Bad : Receive Frame Count for Good and Bad 64 Byte Frames 958
27.3.63 Rx_65To127Octets_Frames_Good_Bad : Receive Frame Count for Good and Bad 65 to 127 Bytes Frames 959
27.3.64 Rx_128To255Octets_Frames_Good_Bad : Receive Frame Count for Good and Bad 128 to 255 Bytes Frames 959
27.3.65 Rx_256To511Octets_Frames_Good_Bad : Receive Frame Count for Good and Bad 256 to 511 Bytes Frames 959
27.3.66 Rx_512To1023Octets_Frames_Good_Bad : Receive Frame Count for Good and Bad 512 to 1,023 Bytes Frames 960
27.3.67 Rx_1024ToMaxOctets_Frames_Good_Bad : Receive Frame Count for Good and Bad 1,024 to Maxsize Bytes Frames 960
27.3.68 Rx_Unicast_Frames_Good : Receive Frame Count for Good Unicast Frames..... 960
27.3.69 Rx_Length_Error_Frames : Receive Frame Count for Length Error Frames 960
27.3.70 Rx_Out_Of_Range_Type_Frames : Receive Frame Count for Out of Range Frames..... 961
27.3.71 Rx_Pause_Frames : Receive Frame Count for PAUSE Frames 961
27.3.72 Rx_FIFO_Overflow_Frames : Receive Frame Count for FIFO Overflow Frames 961
27.3.73 Rx_VLAN_Frames_Good_Bad : Receive Frame Count for Good and Bad VLAN Frames 962
27.3.74 Rx_Watchdog_Error_Frames : Receive Frame Count for Watchdog Error Frames 962
27.3.75 Rx_Receive_Error_Frames : Receive Frame Count for Receive Error Frames 962
27.3.76 Rx_Control_Frames_Good : Receive Frame Count for Good Control Frames 962
27.3.77 HASH_TABLE_REGn : Hash Table Register n (n = 0 to 7) 963
27.3.78 VLAN_Hash_Table_Reg : VLAN Hash Table Register 963
27.3.79 Timestamp_Control : Timestamp Control Register 963
27.3.80 Sub_Second_Increment : Sub-Second Increment Register 965
27.3.81 System_Time_Seconds : System Time - Seconds Register..... 965
27.3.82 System_Time_Nanoseconds : System Time - Nanoseconds Register..... 966
27.3.83 System_Time_Seconds_Update : System Time - Seconds Update Register..... 966
27.3.84 System_Time_Nanoseconds_Update : System Time - Nanoseconds Update Register..... 966

27.3.85 Timestamp_Addend : Timestamp Addend Register 967
27.3.86 Target_Time_Seconds : Target Time Seconds Register 967
27.3.87 Target_Time_Nanoseconds : Target Time Nanoseconds Register 967
27.3.88 Timestamp_Status : Timestamp Status Register 968
27.3.89 Auxiliary_Timestamp_Nanoseconds : Auxiliary Timestamp – Nanoseconds Register 969
27.3.90 Auxiliary_Timestamp_Seconds : Auxiliary Timestamp - Seconds Register 969
27.3.91 Bus_Mode : Bus Mode Register 969
27.3.92 Transmit_Poll_Demand : Transmit Poll Demand Register 971
27.3.93 Receive_Poll_Demand : Receive Poll Demand Register 971
27.3.94 Receive_Descriptor_List_Address : Receive Descriptor List Address Register 971
27.3.95 Transmit_Descriptor_List_Address : Transmit Descriptor List Address Register 972
27.3.96 Status : Status Register 972
27.3.97 Operation_Mode : Operation Mode Register 975
27.3.98 Interrupt_Enable : Interrupt Enable Register 977
27.3.99 Missed_Frame_And_Buffer_Overflow_Counter : Missed Frame and Buffer Overflow Counter Register 979
27.3.100 Receive_Interrupt_Watchdog_Timer : Receive Interrupt Watchdog Timer Register 979
27.3.101 AXI_Bus_Mode : AXI Bus Mode Register 980
27.3.102 AXI_Status : AXI Status Register 981
27.3.103 Current_Host_Transmit_Descriptor : Current Host Transmit Descriptor Register 981
27.3.104 Current_Host_Receive_Descriptor : Current Host Receive Descriptor Register 981
27.3.105 Current_Host_Transmit_Buffer_Address : Current Host Transmit Buffer Address Register 982
27.3.106 Current_Host_Receive_Buffer_Address : Current Host Receive Buffer Address Register 982
27.3.107 HW_Feature : HW Feature Register 982
27.3.108 GMACTRGSEL : GMAC PTP Trigger Select Register 983
27.3.109 PTP Timer Pulse Control Registers 984
27.4 Operation 987
27.4.1 Initializing 987
27.4.2 Initializing DMA 987
27.4.3 Initializing GMAC 988
27.4.4 Performing Normal Receive and Transmit Operation 988
27.4.5 Stopping and Starting Transmission 989
27.4.6 Programming Guidelines for GMII Link Transitions 989
27.4.7 Programming Guidelines for IEEE 1588 Timestamping 990
27.4.8 System Time Register Module 990
27.4.9 Programming Guidelines for Energy Efficient Ethernet 992
27.4.10 Enhanced Descriptors 993
27.4.11 PTP Pulse Generator 1002
27.5 Usage Notes 1003
27.5.1 Configurations 1003

27.5.2 Restriction 1003
27.5.3 MAC Management Counters 1004
27.5.4 MMC Registers 1005
27.5.5 Power Management Block 1005
27.5.6 DMA Register 1007
27.5.7 MAC Register 1009
27.5.8 IEEE 1588 Timestamp Registers 1012
27.5.9 Known Issue and Workaround 1014

28. EtherCAT Slave Controller (ESC) 1015
28.1 Overview 1015
28.2 Functional Overview 1016
28.3 Description of Registers 1018
28.3.1 ECATOFFADR : EtherCAT PHY Offset Address Setting Register 1018
28.3.2 ECATOPMOD : EtherCAT Operation Mode Register 1019
28.3.3 ECATDBGC : EtherCAT Debug Control Register 1019
28.3.4 ECATTRGSEL : EtherCAT DC Latch Trigger Select Register 1020
28.4 ESC Information Register 1020
28.4.1 TYPE : Type Register 1020
28.4.2 REVISION : Revision Register 1021
28.4.3 BUILD : Build Register 1021
28.4.4 FMMU_NUM : FMMU Supported Register 1021
28.4.5 SYNC_MANAGER : SyncManager Supported Register 1022
28.4.6 RAM_SIZE : RAM Size Register 1022
28.4.7 PORT_DESC : Port Descriptor Register 1022
28.4.8 FEATURE : ESC Features Supported Register 1023
28.5 Station Address Registers 1024
28.5.1 STATION_ADR : Configured Station Address Register 1024
28.5.2 STATION_ALIAS : Configured Station Alias Register 1024
28.6 Write Protection Registers 1024
28.6.1 WR_REG_ENABLE : Write Register Enable Register 1024
28.6.2 WR_REG_PROTECT : Write Register Protection Register 1025
28.6.3 ESC_WR_ENABLE : ESC Write Enable Register 1025
28.6.4 ESC_WR_PROTECT : ESC Write Protection Register 1026
28.7 Data Link Layer Registers 1026
28.7.1 ESC_RESET_ECAT_W : ESC Reset ECAT Register for write 1026
28.7.2 ESC_RESET_ECAT_R : ESC Reset ECAT Register for read 1026
28.7.3 ESC_RESET_PDI_W : ESC Reset PDI Register for write 1027
28.7.4 ESC_RESET_PDI_R : ESC Reset PDI Register for read 1027
28.7.5 ESC_DL_CONTROL : ESC DL Control Register 1028
28.7.6 PHYSICAL_RW_OFFSET : Physical Read/Write Offset Register 1029

28.7.7 ESC_DL_STATUS : ESC DL Status Register 1029
28.8 Application Layer Registers 1031
28.8.1 AL_CONTROL : AL Control Register 1031
28.8.2 AL_STATUS : AL Status Register 1031
28.8.3 AL_STATUS_CODE : AL Status Code Register 1032
28.8.4 RUN_LED_OVERRIDE : RUN LED Override Register 1032
28.8.5 ERR_LED_OVERRIDE : ERR LED Override Register 1033
28.9 PDI Registers 1033
28.9.1 PDI_CONTROL : PDI Control Register 1033
28.9.2 ESC_CONFIG : ESC Configuration Register 1034
28.9.3 PDI_CONFIG : PDI Configuration Register 1034
28.9.4 SYNC_LATCH_CONFIG : SYNC/LATCH PDI Configuration Register 1035
28.9.5 EXT_PDI_CONFIG : Extended PDI Configuration Register 1035
28.10 Interrupt Registers 1036
28.10.1 ECAT_EVENT_MASK : ECAT Event Mask Register 1036
28.10.2 AL_EVENT_MASK : AL Event Mask Register 1036
28.10.3 ECAT_EVENT_REQ : ECAT Event Request Register 1037
28.10.4 AL_EVENT_REQ : AL Event Request Register 1038
28.11 Error Counter Registers 1039
28.11.1 RX_ERR_COUNTn : RX Error Counter n Register (n = 0 to 2) 1039
28.11.2 FWD_RX_ERR_COUNTn : Forwarded RX Error Counter n Register (n = 0 to 2) 1039
28.11.3 ECAT_PROC_ERR_COUNT : ECAT Processing Unit Error Counter Register 1040
28.11.4 PDI_ERR_COUNT : PDI Error Counter Register 1040
28.11.5 LOST_LINK_COUNTn : Lost Link Counter n Register (n = 0 to 2) 1040
28.12 Watchdog Registers 1041
28.12.1 WD_DIVIDE : Watchdog Divider Register 1041
28.12.2 WDT_PDI : Watchdog Time PDI Register 1041
28.12.3 WDT_DATA : Watchdog Time Process Data Register 1042
28.12.4 WDS_DATA : Watchdog Status Process Data Register 1042
28.12.5 WDC_DATA : Watchdog Counter Process Data Register 1042
28.12.6 WDC_PDI : Watchdog Counter PDI Register 1043
28.13 SII EEPROM Interface Registers 1043
28.13.1 EEP_CONF : EEPROM Configuration Register 1043
28.13.2 EEP_STATE : EEPROM PDI Access State Register 1044
28.13.3 EEP_CONT_STAT : EEPROM Control/Status Register 1044
28.13.4 EEP_ADR : EEPROM Address Register 1045
28.13.5 EEP_DATA : EEPROM Data Register 1046
28.14 MII Management Interface Registers 1046
28.14.1 MII_CONT_STAT : MII Management Control/Status Register 1046
28.14.2 PHY_ADR : PHY Address Register 1047

28.14.3 PHY_REG_ADR : PHY Register Address Register 1048
28.14.4 PHY_DATA : PHY Data Register 1048
28.14.5 MII_ECAT_ACS_STAT : MII Management ECAT Access State Register 1048
28.14.6 MII_PDI_ACS_STAT : MII Management PDI Access State Register 1049
28.15 FMMU Registers 1049
28.15.1 FMMUn_L_START_ADR : FMMU Logical Start Address n Register (n = 0 to 7) 1049
28.15.2 FMMUn_LEN : FMMU Length n Register (n = 0 to 7) 1049
28.15.3 FMMUn_L_START_BIT : FMMU Logical Start Bit n Register (n = 0 to 7) 1050
28.15.4 FMMUn_L_STOP_BIT : FMMU Logical Stop Bit n Register (n = 0 to 7) 1050
28.15.5 FMMUn_P_START_ADR : FMMU Physical Start Address n Register (n = 0 to 7) 1050
28.15.6 FMMUn_P_START_BIT : FMMU Physical Start Bit n Register (n = 0 to 7) 1051
28.15.7 FMMUn_TYPE : FMMU Type n Register (n = 0 to 7) 1051
28.15.8 FMMUn_ACT : FMMU Activate n Register (n = 0 to 7) 1052
28.16 SyncManager Registers 1052
28.16.1 SMn_P_START_ADR : SyncManager Physical Start Address n Register (n = 0 to 7) 1052
28.16.2 SMn_LEN : SyncManager Length n Register (n = 0 to 7) 1052
28.16.3 SMn_CONTROL : SyncManager Control n Register (n = 0 to 7) 1053
28.16.4 SMn_STATUS : SyncManager Status n Register (n = 0 to 7) 1053
28.16.5 SMn_ACT : SyncManager Activate n Register (n = 0 to 7) 1054
28.16.6 SMn_PDI_CONT : SyncManager PDI Control n Register (n = 0 to 7) 1055
28.17 Distributed Clock Registers 1055
28.17.1 DC Receive Time Registers 1055
28.17.2 DC Time Loop Control Unit Registers 1056
28.17.3 Cyclic Unit Control Registers 1060
28.17.4 SYNC Output Unit Registers 1061
28.17.5 Latch Input Unit Registers 1065
28.17.6 SyncManager Event Time Registers 1069
28.18 Other Registers 1070
28.18.1 PRODUCT_ID : Product ID Register 1070
28.18.2 VENDOR_ID : Vendor ID Register 1070
28.18.3 USER_RAM : User RAM 1070
28.18.4 DATA_RAM : Process Data RAM 1071
28.18.5 Setting the Module-Stop Function 1072
28.19 Initial Settings 1072
28.20 Configuration of the Reset Circuit 1072
28.21 Usage Notes 1073

29. USB 2.0 Host Module (USBHC) 1074

29.1 Overview 1074
29.1.1 Precautions on the Use of USB Host Controller 1074
29.2 Register Descriptions 1075

29.2.1 OHCI Operational Registers 1075
29.2.2 EHCI Operational Registers 1093
29.2.3 AHB Bridge Registers 1107
29.2.4 Core Defined Registers 1111
29.2.5 UCOM Registers 1112
29.3 Clock Lines 1116
29.3.1 Externally Supplied Clocks 1116
29.4 Interrupts 1116
29.4.1 Interrupt Control Registers 1116
29.4.2 U2H_BIND_INT 1117
29.4.3 Time Required to Clear Interrupt Signals 1118
29.5 Overcurrent Control and VBUS Control 1118
29.5.1 Overcurrent Control 1118
29.5.2 VBUS Control 1119
29.5.3 Initial Settings of PPON1 1120
29.5.4 Procedure for USB_VBUSEN Control for Overcurrent Detection 1121
29.5.5 Procedure for USB_VBUSEN Setting 1121
29.6 Initial Setting Sequence 1122
29.6.1 Sample of Initial Settings 1122
29.7 OTG Control 1123
29.7.1 Port Power (VBUS) Control and Connection with External Power-Supply IC 1123
29.7.2 OTG System Configuration Example 1123
29.7.3 OTG Initial Setting Procedure 1124
29.7.4 A-Device/B-Device Recognition Procedure 1125
29.7.5 Disconnect Detection Procedure 1126
29.7.6 HNP Control Procedure (Conforming to OTG Revision 2.0) 1127
29.7.7 SRP Control Procedure (Conforming to OTG Revision 2.0) 1135

30. USB 2.0 HS Function Module (USBf) 1138

30.1 Overview 1138
30.1.1 Identification of the USB Transfer Speed 1139
30.1.2 Bus Interface 1139
30.1.3 USB Event 1139
30.1.4 USB Data Transfer 1139
30.1.5 DMA Transfer 1140
30.2 Register Descriptions 1140
30.2.1 System Configuration Control 1140
30.2.2 USB Signal Control 1143
30.2.3 Test Mode 1144
30.2.4 FIFO Ports 1145
30.2.5 Enabling Interrupts 1152

30.2.6 SOF Control Register 1154
30.2.7 Interrupt Status 1155
30.2.8 Frame Number Registers 1162
30.2.9 USB Address 1163
30.2.10 USB Request Registers 1163
30.2.11 DCP Configuration 1165
30.2.12 Pipe Configuration Register 1168
30.2.13 Pipe Control Registers 1176
30.2.14 Transaction Counters 1182
30.2.15 Next Register Set 1184
30.2.16 Current Register Set 1185
30.2.17 Channel Register Set 1187
30.2.18 Link Register Set 1198
30.2.19 Skip Register Set 1198
30.2.20 DMA Register Set 1202
30.3 Operation 1205
30.3.1 System Controls and Oscillation Controls 1205
30.3.2 Resets 1206
30.3.3 USB Data Bus Resistor Controller 1206
30.3.4 Notes on Stopping Clocks 1207
30.4 Interrupt Functions 1207
30.4.1 Overview of Interrupt Functions 1207
30.4.2 Device State Transition Interrupt 1209
30.4.3 Control Transfer Stage Transition Interrupt 1210
30.5 Pipe Control 1211
30.5.1 Maximum Packet Size Setting 1212
30.5.2 Response PID 1213
30.5.3 Pipe Control Register Switching Procedures 1213
30.5.4 Data PID Sequence Bit 1214
30.6 FIFO Buffers 1214
30.6.1 FIFO Buffer Allocation 1214
30.6.2 Clearing FIFO Buffers 1215
30.7 FIFO Port Function 1215
30.7.1 FIFO Port Selection 1216
30.7.2 DxFIFO Automatic Clear Mode (DxFIFO Port Read Direction) 1217
30.7.3 BRDY Interrupt Timing Selection 1217
30.8 Control Transfer (DCP) 1218
30.8.1 Setup Stage 1218
30.8.2 Data Stage 1218
30.8.3 Status Stage 1218

30.8.4 Control Transfer Automatic Response 1219
30.9 Bulk Transfer (Pipes 1 to 5) 1219
30.9.1 NYET Handshake Control 1219
30.10 Interrupt Transfer (Pipes 6 to 9) 1219
30.11 Isochronous Transfer (Pipes 1 and 2) 1220
30.11.1 Isochronous Transfer Error Detection 1220
30.11.2 DATA-PID 1220
30.11.3 Interval Counter 1221
30.11.4 Isochronous Transfer Transmission Data Setup 1222
30.11.5 Isochronous Transfer Transmission Buffer Flush 1223
30.12 SOF Interpolation 1224
30.13 Link Power Management Processing 1224
30.13.1 Descriptor 1225
30.13.2 Basic Processing 1225
30.13.3 HIRD Value Negotiation 1225
30.14 DMA Mode 1226
30.14.1 Register Mode/Link Mode 1226
30.15 DMA Transfer 1240
30.15.1 Transfer Mode 1240
30.15.2 DMA Channel Priority Control 1240
30.15.3 Forced Ejection Request 1242
30.15.4 DMA Transfer Completion Interrupt (USB_FDMAn) 1242
30.15.5 DMA Error Interrupt (USB_FDMAERR) 1243
30.15.6 Interval Count Function 1243
30.15.7 Differences in Operation According to the Transfer Data Size 1244
30.15.8 Transfer Status 1246
30.16 Access Type 1248
30.16.1 AHB Master DMA Transfer Combinations 1248

31. Serial Communications Interface (SCI) 1251

31.1 Overview 1251
31.2 Register Descriptions 1254
31.2.1 RSR : Receive Shift Register 1255
31.2.2 RDR : Receive Data Register 1255
31.2.3 TDR : Transmit Data Register 1256
31.2.4 TSR : Transmit Shift Register 1257
31.2.5 CCR0 : Common Control Register 0 1257
31.2.6 CCR1 : Common Control Register 1 1260
31.2.7 CCR2 : Common Control Register 2 1263
31.2.8 CCR3 : Common Control Register 3 1269
31.2.9 CCR4 : Common Control Register 4 1273

31.2.10 ICR : Simple I2C Control Register 1275
31.2.11 FCR : FIFO Control Register 1277
31.2.12 DCR : Driver Control Register 1279
31.2.13 CSR : Common Status Register 1280
31.2.14 ISR : Simple I2C Status Register 1285
31.2.15 FRSR : FIFO Receive Status Register 1286
31.2.16 FTSR : FIFO Transmit Status Register 1287
31.2.17 CFCLR : Common Flag Clear Register 1288
31.2.18 ICFCLR : Simple I2C Flag Clear Register 1289
31.2.19 FFCLR : FIFO Flag Clear Register 1289
31.3 Asynchronous Mode 1289
31.3.1 Serial Data Transfer Format 1290
31.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode 1291
31.3.3 Clock 1292
31.3.4 Double-Speed Operation and frequency of 6 times the bit rate 1293
31.3.5 CTS and RTS Function 1293
31.3.6 Address Match (receive data match detection) Function 1293
31.3.7 SCI Initialization (Asynchronous Mode) 1296
31.3.8 Serial Data Transmission (Asynchronous Mode) 1299
31.3.9 Serial Data Reception (Asynchronous Mode) 1305
31.3.10 The function of adjust receive sampling timing (Asynchronous Mode) 1311
31.3.11 The function of adjust transmit timing (Asynchronous Mode) 1315
31.4 Multi-Processor Communication Function 1319
31.4.1 Multi-Processor Serial Data Transmission 1320
31.4.2 Multi-Processor Serial Data Reception 1323
31.5 Smart Card Interface Mode 1329
31.5.1 Sample Connection 1329
31.5.2 Data Format (Except in Block Transfer Mode) 1330
31.5.3 Block Transfer Mode 1332
31.5.4 Receive Data Sampling Timing and Reception Margin 1332
31.5.5 Initialization of SCI (Smart Card Interface Mode) 1333
31.5.6 Serial Data Transmission (Except in Block Transfer Mode) 1335
31.5.7 Serial Data Reception (Except in Block Transfer Mode) 1337
31.5.8 Clock Output Control 1339
31.6 Simple I2C mode 1340
31.6.1 Generation of Start, Restart, and Stop Conditions 1341
31.6.2 Clock Synchronization 1342
31.6.3 SDA Output Delay 1343
31.6.4 SCI Initialization (Simple I2C mode) 1344
31.6.5 Operation in Master Transmission (Simple I2C mode) 1345

31.6.6 Master Reception (Simple I2C mode) 1349
31.7 Clock Synchronous Mode 1352
31.7.1 Clock 1353
31.7.2 CTS and RTS Functions 1353
31.7.3 Initializing the SCI (Clock Synchronous Mode) 1354
31.7.4 Serial Data Transmission (Clock Synchronous Mode) 1355
31.7.5 Serial Data Reception (Clock Synchronous Mode) 1359
31.7.6 Serial Data Transmission and Reception (Clock Synchronous Mode) 1363
31.7.7 Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used 1365
31.8 Simple-SPI Mode 1366
31.8.1 Pin Status in Master and Slave Mode 1367
31.8.2 SS Function in Master Mode 1367
31.8.3 SS Function in Slave Mode 1368
31.8.4 Relationship between Clock and Transmit/Receive Data 1368
31.8.5 SCI Initialization in Simple-SPI Mode 1369
31.8.6 Serial Data Transmission and Reception in Simple-SPI Mode 1369
31.8.7 Reception Sampling Timing Adjustment Function in Simple-SPI Mode with Internal Clock Used 1370
31.9 Bit Rate Modulation Function 1370
31.10 Noise Cancellation Function 1370
31.11 RS-485 Driver Control Function 1371
31.12 Loopback Function 1372
31.13 Half-Duplex communication Function 1373
31.14 Synchronizer Bypass Function 1373
31.15 Interrupt signal 1374
31.15.1 Buffer Operations for TXI and RXI interrupts 1375
31.15.2 Interrupt in Asynchronous Mode, Clock Synchronous Mode, and Simple-SPI Mode 1375
31.15.3 Interrupt in Smart Card Interface Mode 1376
31.15.4 Interrupts in Simple-I2C Mode 1377
31.16 Usage Notes 1378
31.16.1 SCI Operations during Low Power Consumption State 1378
31.16.2 Break Detection and Processing 1382
31.16.3 Mark State and Production of Breaks 1383
31.16.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode) 1383
31.16.5 Writing Data to TDR (Transmit FIFO) 1383
31.16.6 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode) 1383
31.16.7 Notes on Starting Transfer 1384
31.16.8 Limitations on Simple SPI Mode 1385

31.16.9 Notes on Transmit Enable bit (TE) 1385
31.16.10 Notes on RS-485 Driver Control function 1386
31.16.11 Notes on Loopback function 1386
31.16.12 Notes Regarding Register Access When Operation Clock (PCLKSCIn) Is Slower Than Bus Clock (PCLKM) 1386
31.16.13 Notes on interrupting operation 1386
31.16.14 Notes on communication mode transition 1386

32. I2C Bus Interface (IIC) 1387

32.1 Overview 1387
32.2 Register Descriptions 1390
32.2.1 ICCR1 : I2C Bus Control Register 1 1390
32.2.2 ICCR2 : I2C Bus Control Register 2 1392
32.2.3 ICMR1 : I2C Bus Mode Register 1 1395
32.2.4 ICMR2 : I2C Bus Mode Register 2 1395
32.2.5 ICMR3 : I2C Bus Mode Register 3 1397
32.2.6 ICFER : I2C Bus Function Enable Register 1399
32.2.7 ICSER : I2C Bus Status Enable Register 1400
32.2.8 ICIER : I2C Bus Interrupt Enable Register 1401
32.2.9 ICSR1 : I2C Bus Status Register 1 1402
32.2.10 ICSR2 : I2C Bus Status Register 2 1405
32.2.11 SARLy : Slave Address Register L y (y = 0 to 2) 1408
32.2.12 SARUy : Slave Address Register U y (y = 0 to 2) 1408
32.2.13 ICBRL : I2C Bus Bit Rate Low-Level Register 1409
32.2.14 ICBRH : I2C Bus Bit Rate High-Level Register 1410
32.2.15 ICDRT : I2C Bus Transmit Data Register 1411
32.2.16 ICDRR : I2C Bus Receive Data Register 1411
32.2.17 ICDRS : I2C Bus Shift Register 1412
32.3 Operation 1412
32.3.1 Communication Data Format 1412
32.3.2 Initial Settings 1413
32.3.3 Master Transmit Operation 1414
32.3.4 Master Receive Operation 1418
32.3.5 Slave Transmit Operation 1423
32.3.6 Slave Receive Operation 1425
32.4 SCL Synchronization Circuit 1427
32.5 SDA Output Delay Function 1428
32.6 Digital Noise Filter Circuits 1429
32.7 Address Match Detection 1430
32.7.1 Slave-Address Match Detection 1430

32.7.2 Detection of General Call Address ... 1432
32.7.3 Device ID Address Detection ... 1433
32.7.4 Host Address Detection ... 1434
32.8 Automatic Low-Hold Function for SCL ... 1435
32.8.1 Function to Prevent Wrong Transmission of Transmit Data ... 1435
32.8.2 NACK Reception Transfer Suspension Function ... 1436
32.8.3 Function to Prevent Failure to Receive Data ... 1437
32.9 Arbitration-Lost Detection Functions ... 1438
32.9.1 Master Arbitration-Lost Detection (MALE Bit) ... 1438
32.9.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit) ... 1440
32.9.3 Slave Arbitration-Lost Detection (SALE Bit) ... 1441
32.10 Start, Restart, and Stop Condition Issuing Function ... 1442
32.10.1 Issuing a Start Condition ... 1442
32.10.2 Issuing a Restart Condition ... 1442
32.10.3 Issuing a Stop Condition ... 1444
32.11 Bus Hanging ... 1444
32.11.1 Timeout Function ... 1444
32.11.2 Extra SCL Clock Cycle Output Function ... 1445
32.11.3 IIC Reset and Internal Reset ... 1446
32.12 SMBus Operation ... 1446
32.12.1 SMBus Timeout Measurement ... 1447
32.12.2 Packet Error Code (PEC) ... 1448
32.12.3 SMBus Host Notification Protocol (Notify ARP Master Command) ... 1448
32.13 Interrupt Sources ... 1448
32.14 Resets, Registers, and Function States When Issuing Each Condition ... 1449
32.15 Usage Notes ... 1450
32.15.1 Settings for the Module-Stop Function ... 1450

33. CAN-FD Interface (CANFD) ... 1451

33.1 Overview ... 1451
33.2 Register Descriptions ... 1454
33.2.1 CFDCnNCFG : Channel n Nominal Bit Rate Configuration Register (n = 0, 1) ... 1454
33.2.2 CFDCnCTR : Channel n Control Register (n = 0, 1) ... 1455
33.2.3 CFDCnSTS : Channel n Status Register (n = 0, 1) ... 1459
33.2.4 CFDCnERFL : Channel n Error Flag Register (n = 0, 1) ... 1462
33.2.5 CFDCnDCFG : Channel n Data Bit Rate Configuration Register (n = 0, 1) ... 1467
33.2.6 CFDCnFDCFG : Channel n CAN-FD Configuration Register (n = 0, 1) ... 1468
33.2.7 CFDCnFDCTR : Channel n CAN-FD Control Register (n = 0, 1) ... 1472
33.2.8 CFDCnFDSTS : Channel n CAN-FD Status Register (n = 0, 1) ... 1473
33.2.9 CFDCnFDCRC : Channel n CAN-FD CRC Register (n = 0, 1) ... 1475
33.2.10 CFDGIPV : Global IP Version Register ... 1475

33.2.11 CFDGCFG : Global Configuration Register 1476
33.2.12 CFDGCTR : Global Control Register 1478
33.2.13 CFDGSTS : Global Status Register 1480
33.2.14 CFDGERFL : Global Error Flag Register 1481
33.2.15 CFDGTINTSTS0 : Global TX Interrupt Status Register 0 1483
33.2.16 CFDGTSC : Global Timestamp Counter Register 1486
33.2.17 CFDGAFLECTR : Global Acceptance Filter List Entry Control Register 1486
33.2.18 CFDGAFLCFG0 : Global Acceptance Filter List Configuration Register 0 1487
33.2.19 CFDGAFLIDn : Global Acceptance Filter List ID Register n (n = 1 to 16) 1488
33.2.20 CFDGAFLMn : Global Acceptance Filter List Mask Register n (n = 1 to 16) 1489
33.2.21 CFDGAFLP0n : Global Acceptance Filter List Pointer 0 Register n (n = 1 to 16) 1490
33.2.22 CFDGAFLP1n : Global Acceptance Filter List Pointer 1 Register n (n = 1 to 16) 1492
33.2.23 CFDRMNB : RX Message Buffer Number Register 1494
33.2.24 CFDRMND0 : RX Message Buffer New Data Register 0 1495
33.2.25 CFDRFCCn : RX FIFO Configuration/Control Register n (n = 0 to 7) 1496
33.2.26 CFDRFSTSn : RX FIFO Status Register n (n = 0 to 7) 1498
33.2.27 CFDRFPCTRn : RX FIFO Pointer Control Register n (n = 0 to 7) 1499
33.2.28 CFDCFCCn : Common FIFO Configuration/Control Register n (n = 0 to 5) 1500
33.2.29 CFDCFCCEn : Common FIFO Configuration/Control Enhancement Register n (n = 0 to 5) 1503
33.2.30 CFDCFSTSn : Common FIFO Status Register n (n = 0 to 5) 1504
33.2.31 CFDCFPCTRn : Common FIFO Pointer Control Register n (n = 0 to 5) 1508
33.2.32 CFDFESTS : FIFO Empty Status Register 1509
33.2.33 CFDFFSTS : FIFO Full Status Register 1509
33.2.34 CFDFMSTS : FIFO Message Lost Status Register 1510
33.2.35 CFDRFISTS : RX FIFO Interrupt Flag Status Register 1510
33.2.36 CFDCFRISTS : Common FIFO RX Interrupt Flag Status Register 1511
33.2.37 CFDCFTISTS : Common FIFO TX Interrupt Flag Status Register 1512
33.2.38 CFDFFFSTS : FIFO FDC Full Status Register 1512
33.2.39 CFDCFMOWSTS : Common FIFO Message Overwrite Status Register 1513
33.2.40 CFDCFOFRISTS : Common FIFO One Frame RX Interrupt Flag Status Register 1513
33.2.41 CFDCFOFTISTS : Common FIFO One Frame TX Interrupt Flag Status Register 1514
33.2.42 CFDCDTCT : DMA Transfer Control Register 1514
33.2.43 CFDCDTSTS : DMA Transfer Status Register 1515
33.2.44 CFDCDTTCT : DMA TX Transfer Control Register 1517
33.2.45 CFDCDTTSTS : DMA TX Transfer Status Register 1518
33.2.46 CFDGRINTSTSn : Global RX Interrupt Status Register n (n = 0, 1) 1519
33.2.47 CFDTMCn : TX Message Buffer Control Register n (n = 0 to 127) 1521
33.2.48 CFDTMSTSn : TX Message Buffer Status Register n (n = 0 to 127) 1522
33.2.49 CFDTMTRSTSf : TX Message Buffer Transmission Request Status Register f (f = 0 to 3) 1523

33.2.50 CFDTMTARSTSf : TX Message Buffer Transmission Abort Request Status Register f (f = 0 to 3) 1524
33.2.51 CFDTMTCSTSf : TX Message Buffer Transmission Completion Status Register f (f = 0 to 3) 1525
33.2.52 CFDTMTASTSf : TX Message Buffer Transmission Abort Status Register f (f = 0 to 3) 1526
33.2.53 CFDTMIECf : TX Message Buffer Transmission Interrupt Enable Register f (f = 0 to 3) 1527
33.2.54 CFDTXQCC0n : TX Queue Configuration/Control Register 0n (n = 0, 1) 1528
33.2.55 CFDTXQCC1n : TX Queue Configuration/Control Register 1n (n = 0, 1) 1530
33.2.56 CFDTXQCC2n : TX Queue Configuration/Control Register 2n (n = 0, 1) 1533
33.2.57 CFDTXQCC3n : TX Queue Configuration/Control Register 3n (n = 0, 1) 1535
33.2.58 CFDTXQSTS0n : TX Queue Status Register 0n (n = 0, 1) 1537
33.2.59 CFDTXQSTS1n : TX Queue Status Register 1n (n = 0, 1) 1540
33.2.60 CFDTXQSTS2n : TX Queue Status Register 2n (n = 0, 1) 1543
33.2.61 CFDTXQSTS3n : TX Queue Status Register 3n (n = 0, 1) 1545
33.2.62 CFDTXQPCTR0n : TX Queue Pointer Control Register 0n (n = 0, 1) 1547
33.2.63 CFDTXQPCTR1n : TX Queue Pointer Control Register 1n (n = 0, 1) 1548
33.2.64 CFDTXQPCTR2n : TX Queue Pointer Control Register 2n (n = 0, 1) 1548
33.2.65 CFDTXQPCTR3n : TX Queue Pointer Control Register 3n (n = 0, 1) 1549
33.2.66 CFDTXQESTS : TX Queue Empty Status Register 1549
33.2.67 CFDTXQFISTS : TX Queue Full Interrupt Status Register 1550
33.2.68 CFDTXQMSTS : TX Queue Message Lost Status Register 1551
33.2.69 CFDTXQISTS : TX Queue Interrupt Status Register 1551
33.2.70 CFDTXQOFTISTS : TX Queue One Frame TX Interrupt Status Register 1552
33.2.71 CFDTXQOFRISTS : TX Queue One Frame RX Interrupt Status Register 1553
33.2.72 CFDTXQFSTS : TX Queue Full Status Register 1554
33.2.73 CFDTHLCCn : TX History List Configuration/Control Register n (n = 0, 1) 1554
33.2.74 CFDTHLSTSn : TX History List Status Register n (n = 0, 1) 1556
33.2.75 CFDTHLACC0n : Channel n TX History List Access Register 0 (n = 0, 1) 1557
33.2.76 CFDTHLACC1n : Channel n TX History List Access Register 1 (n = 0, 1) 1558
33.2.77 CFDTHLPCTRn : TX History List Pointer Control Register n (n = 0, 1) 1559
33.2.78 CFDGRSTC : Global Reset Control Register 1559
33.2.79 CFDGFCMC : Global Flexible CAN Mode Configuration Register 1560
33.2.80 CFDGFTBAC : Global Flexible Transmission Buffer Assignment Configuration Register 1560
33.2.81 CFDGTSTCFG : Global Test Configuration Register 1561
33.2.82 CFDGTSTCTR : Global Test Control Register 1562
33.2.83 CFDGFDCFG : Global FD Configuration Register 1563
33.2.84 CFDGLOCKK : Global Lock Key Register 1563
33.2.85 CFDRPGACCn : RAM Test Page Access Register n (n = 0 to 63) 1564
33.2.86 CFDCnBLCT : Channel n Bus Load Control Register (n = 0, 1) 1564
33.2.87 CFDCnBLSTS : Channel n Bus Load Status Register (n = 0, 1) 1565
33.2.88 Identifier Bits Alignment 1565

33.2.89 Message Buffer Component Structure 1566
33.3 Operation 1580
33.3.1 Overview 1580
33.3.2 Global Modes 1581
33.3.3 Channel Modes 1587
33.3.4 Global Mode and Channel Mode Transition Interactions 1592
33.4 Initialization 1594
33.4.1 Initialization of CAN Clock, Bit Timing and Baud Rate 1594
33.4.2 CAN Module Configuration after Hardware Reset 1601
33.5 Acceptance Filtering Function using Global Acceptance Filter List (AFL) 1603
33.5.1 Overview 1603
33.5.2 Allocation of AFL Entries to each CAN Channel 1604
33.5.3 AFL Entry Description 1604
33.5.4 Entering Entries in the AFL 1606
33.5.5 Loopback Modes 1609
33.5.6 IDE Masking 1609
33.6 FIFO Buffers and Normal Message Buffer Configuration 1610
33.6.1 Normal RX Message Buffers 1611
33.6.2 FIFO Buffer 1611
33.7 Interrupt and DMA 1616
33.7.1 Interrupts 1616
33.7.2 DMA Transfer 1619
33.8 Reception and Transmission 1622
33.8.1 Reception 1622
33.8.2 Transmission 1628
33.9 Test Mode 1648
33.9.1 Channel Specific Test Modes 1648
33.9.2 Global Test Modes 1651
33.10 Bus Traffic Measurement 1656
33.10.1 How to count the CAN Bus Idle Time 1656
33.10.2 Operations and Measurement Procedure 1656
33.11 Flexible CAN Mode 1657
33.12 Flexible Transmission Buffer Assignment 1658
34. Serial Peripheral Interface (SPI) 1663
34.1 Overview 1663
34.1.1 Features 1663
34.2 Register Descriptions 1666
34.2.1 SPDR/SPDR_HA/SPDR_BY : SPI Data Register 1666
34.2.2 SPCKD : SPI Clock Delay Register 1668
34.2.3 SSLND : SPI Slave Select Negation Delay Register 1669

34.2.4 SPND : SPI Next-Access Delay Register 1670
34.2.5 MRCKD : SPI ClocK Digital control Register for Master Receive 1670
34.2.6 SPCR : SPI Control Register 1671
34.2.7 SPCRRM : SPI Control Register for Master Receive only 1676
34.2.8 SPDRCR : SPI Control Register for Received Data Ready Detection 1677
34.2.9 SPPCR : SPI Pin Control Register 1678
34.2.10 SPCR2 : SPI Control Register 2 1679
34.2.11 SSLP : SPI Slave Select Polarity Register 1679
34.2.12 SPBR : SPI Bit Rate Register 1680
34.2.13 SPSCR : SPI Sequence Control Register 1681
34.2.14 SPCMDm : SPI Command Register m (m = 0 to 7) 1682
34.2.15 SPDCR : SPI Data Control Register 1684
34.2.16 SPDCR2 : SPI Data Control Register 2 1686
34.2.17 SPSSR : SPI Sequence Status Register 1686
34.2.18 SPSR : SPI Status Register 1687
34.2.19 SPTFSR : SPI Transfer FIFO Status Register 1693
34.2.20 SPRFSR : SPI Receive FIFO Status Register 1693
34.2.21 SPPSR : SPI Poling Register 1694
34.2.22 SPSRC : SPI Status Clear Register 1694
34.2.23 SPFCR : SPI FIFO Clear Register 1695
34.3 Operation 1695
34.3.1 Overview of SPI Operations 1695
34.3.2 SPI Pin Control 1696
34.3.3 SPI System Configuration Examples 1697
34.3.4 Data Format 1703
34.3.5 Transfer Format (Frame Format) 1716
34.3.6 Communications Operating Mode 1718
34.3.7 Transmission Buffer Empty/Receive Buffer Full Interrupt 1721
34.3.8 Idle Interrupt 1722
34.3.9 Communication End Interrupt 1724
34.3.10 Error Detection 1733
34.3.11 Received Data Ready Detection 1739
34.3.12 SPI Initialization 1739
34.3.13 SPI Operation 1740
34.3.14 Clock Synchronous Operation 1759
34.3.15 Loopback Mode 1765
34.3.16 Self-Diagnosis of Parity Function 1766
34.3.17 Interrupt Requests 1767
34.4 Usage Notes 1768
34.4.1 Notes on Communication Start 1768

34.4.2 Notes on the Module-Stop Function ... 1768
34.4.3 Notes on SPRF and SPTEF Flags ... 1768
34.4.4 Notes on Receive Sampling Function ... 1768
34.4.5 Notes on Burst Transfer in Master Mode ... 1768
34.4.6 Notes on TI-SSP in Slave Mode ... 1768
34.4.7 Notes at SPE = 1 ... 1769

35. Expanded Serial Peripheral Interface (xSPI) ... 1771
35.1 Overview ... 1771
35.2 Register Descriptions ... 1772
35.2.1 xSPI Configuration Registers ... 1772
35.2.2 xSPI Control Registers ... 1778
35.2.3 xSPI Status Registers ... 1792
35.2.4 xSPI Interrupt Registers ... 1794
35.3 Operation ... 1798
35.3.1 xSPI Bus ... 1798
35.3.2 Manual-command ... 1808
35.3.3 Memory-mapping ... 1809
35.3.4 Pattern Control ... 1814
35.3.5 Integrity Checking ... 1816
35.3.6 Interrupts ... 1816
35.3.7 Flows of Operations ... 1817
35.3.8 Usage Notes ... 1824

36. CRC Operation Unit (CRC) ... 1825
36.1 Overview ... 1825
36.2 Register Descriptions ... 1826
36.2.1 CRCCR0 : CRC Control Register 0 ... 1826
36.2.2 CRCDIR/CRCDIR_BY : CRC Data Input Register ... 1826
36.2.3 CRCDOR/CRCDOR_HA/CRCDOR_BY : CRC Data Output Register ... 1827
36.3 Operation ... 1827
36.4 Usage Notes ... 1830
36.4.1 Module-Stop Function Setting ... 1830
36.4.2 Note on Transmission ... 1830

37. Boundary Scan ... 1832
37.1 Overview ... 1832
37.2 Register Descriptions ... 1833
37.2.1 JTIR : Instruction Register ... 1833
37.2.2 JTIDR : ID Code Register ... 1834
37.2.3 JTBPR : Bypass Register ... 1834
37.2.4 JTBSR : Boundary Scan Register ... 1834

37.3 Operation 1834
37.3.1 TAP Controller 1834
37.3.2 List of Commands 1836
37.4 Usage Notes 1837
38. ΔΣ Interface (DSMIF) 1838
38.1 Overview 1838
38.2 Register Descriptions 1841
38.2.1 DSMIF Common Control Registers 1841
38.2.2 DSMIF Channel Control Registers 1859
38.3 Operation 1875
38.3.1 Start Flow 1875
38.3.2 Normal Processing Flow 1876
38.3.3 Stop Flow 1877
38.3.4 Error Flow 1879
38.3.5 Clock Specification 1880
38.3.6 Channel Start and Channel Stop 1881
38.3.7 Channel Synchronization 1881
38.3.8 Data Input Control 1881
38.3.9 Sinc Filter 1881
38.3.10 Available Settings of Sinc Filter 1882
38.3.11 Current Measurement Data Processing 1885
38.3.12 Short-circuit Detection 1885
38.3.13 Overcurrent Detection 1885
38.3.14 Overcurrent Sum Error Detection 1887
38.3.15 Settling Time 1888
38.3.16 Common Interrupt 1888
39. 12-Bit A/D converter (ADC12) 1890
39.1 Overview 1890
39.2 Register Descriptions 1894
39.2.1 ADCSR : A/D Control Register 1894
39.2.2 ADANSA0 : A/D Channel Select Register A0 1895
39.2.3 ADADS0 : A/D-Converted Value Addition/Average Function Channel Select Register 0 1895
39.2.4 ADADC : A/D-Converted Value Addition/Average Count Select Register 1896
39.2.5 ADCER : A/D Control Extended Register 1896
39.2.6 ADSTRGR : A/D Conversion Start Trigger Select Register 1897
39.2.7 ADANSB0 : A/D Channel Select Register B0 1899
39.2.8 ADDBLDR : A/D Data Duplication Register 1899
39.2.9 ADDRn : A/D Data Register n (n = 0 to 3) 1900
39.2.10 ADSHCR : A/D Sample and Hold Control Register 1900

39.2.11 ADELCCR : A/D Event Link Control Register 1901
39.2.12 ADGSPCR : A/D Group Scan Priority Control Register 1901
39.2.13 ADDBLDRA : A/D Data Duplication Register A 1902
39.2.14 ADDBLDRB : A/D Data Duplication Register B 1902
39.2.15 ADWINMON : A/D Compare Function Window A/B Status Monitoring Register 1903
39.2.16 ADCMPCR : A/D Compare Function Control Register 1903
39.2.17 ADCMPANSR0 : A/D Compare Function Window A Channel Select Register 0 1904
39.2.18 ADCMPLR0 : A/D Compare Function Window A Comparison Condition Setting Register 0 1904
39.2.19 ADCMPDR0 : A/D Comparison Function Window A Lower Level Setting Register 1905
39.2.20 ADCMPDR1 : AD Comparison Function Window A Upper Level Setting Register 1905
39.2.21 ADCMPSR0 : A/D Comparison Function Window A Channel Status Register 0 1905
39.2.22 ADWINLLB : A/D Compare Function Window B Lower-Side Level Setting Register 1906
39.2.23 ADWINULB : A/D Compare Function Window B Upper-Side Level Setting Register 1906
39.2.24 ADCMPBSR : A/D Compare Function Window B Status Register 1906
39.2.25 ADCMPBNSR : A/D Compare Function Window B Channel Select Register 1907
39.2.26 ADANSC0 : A/D Channel Select Register C0 1907
39.2.27 ADGCTRGR : A/D Group C Trigger Select Register 1908
39.2.28 ADSSTRn : A/D Sampling State Register n (n = 0 to 3) 1909
39.3 Operation 1909
39.3.1 Scanning Operation 1909
39.3.2 Single Scan Mode 1910
39.3.3 Continuous Scan Mode 1913
39.3.4 Group Scan Mode 1915
39.3.5 Compare Function (Window A, Window B) 1921
39.3.6 A/D Data Register Automatic Clearing Function 1924
39.3.7 A/D-converted Value Addition/Average Function 1925
39.3.8 Starting A/D Conversion with Asynchronous Trigger 1925
39.3.9 Starting A/D Conversion with Synchronous Trigger from Peripheral Module 1925
39.3.10 Event Link Function 1925
39.4 Usage Notes 1926
39.4.1 Notes on Reading Data Registers 1926
39.4.2 Notes on Stopping A/D Conversion 1926
39.4.3 A/D conversion forced stop and operation timing at starts 1928
39.4.4 Notes on Scan End Interrupt Processing 1928
39.4.5 Module-Stop Function Setting 1928
39.4.6 Notes on Entering Low Power Consumption States 1928
40. Temperature Sensor Unit (TSU) 1929
40.1 Overview 1929
40.2 Register Descriptions 1929

40.2.1 TSUSM : Sensor Mode Register 1929
40.2.2 TSUST : Sensor Trigger Register 1930
40.2.3 TSUSCS : Sensor Configuration Setting Register 1930
40.2.4 TSUSAD : Sensor ADC Data Register 1930
40.2.5 TSUSS : Sensor Status Register 1931
40.3 Operation 1931
40.3.1 Startup Sequence 1931
40.3.2 Stop Sequence 1931
40.3.3 A/D Start and Stop Timing 1932
40.3.4 A/D Output Data Store Timing 1932
40.3.5 A/D Conversion Status (CONV) 1932
40.3.6 Using the Temperature Sensor 1933
41. Data Operation Circuit (DOC) 1935
41.1 Overview 1935
41.2 Register Descriptions 1936
41.2.1 DOCR : DOC Control Register 1936
41.2.2 DODIR : DOC Data Input Register 1936
41.2.3 DODSR : DOC Data Setting Register 1937
41.3 Operation 1937
41.3.1 Data Comparison Mode 1937
41.3.2 Data Addition Mode 1937
41.3.3 Data Subtraction Mode 1938
41.4 Interrupt Requests 1939
41.5 Usage Note 1939
41.5.1 Module Stop Function Setting 1939
42. System SRAM (SYSRAM) 1940
42.1 Overview 1940
42.2 Register Descriptions 1941
42.2.1 SYSRAM_CTRLn : System SRAM Control Register n (n = 0, 1) 1941
42.2.2 Wn_EC710CTL : ECC Control Register n (n = 0 to 3) 1942
42.2.3 Wn_EC710TMC : ECC Test Mode Control Register n (n = 0 to 3) 1944
42.2.4 Wn_EC710TRC : ECC Redundancy Bit Data Control Test Register n (n = 0 to 3) 1945
42.2.5 Wn_EC710TED : ECC Encode/Decode Input/Output Switchover Test Register n (n = 0 to 3) 1946
42.2.6 Wn_EC710EADm : ECC Error Address m Register n (n = 0 to 3, m = 0 to 7) 1946
42.3 Operation 1947
42.3.1 Initialization 1947
42.3.2 Access Cycle to System SRAM with ECC Function Enabled 1947
42.3.3 ECC Error Address Calculation Method 1947
42.3.4 Notes on ECC Error Address Register 1947

42.3.5 Notes on ECC Error Detection 1947
42.3.6 Test Function (Error Injection) 1949
42.3.7 Interrupt 1950
42.4 Usage Notes 1950
42.4.1 Release of Register Write Protection 1950
42.4.2 Note on Setting the Register 1950
42.4.3 Access in Bit Size Units 1950
43. One-Time Programmable Memory (OTP) 1951
43.1 Overview 1951
43.2 Register Descriptions 1951
43.2.1 OTPPWR : OTP Power Control Register 1951
43.2.2 OTPSTR : OTP Access Status Register 1952
43.2.3 OTPSTAWR : OTP Write Command Register 1953
43.2.4 OTPADRWR : OTP Write Address Register 1953
43.2.5 OTPDATAWR : OTP Write Data Register 1954
43.2.6 OTPADRRD : OTP Read Address Register 1954
43.2.7 OTPDATARD : OTP Read Data Register 1954
43.3 OTP Memory Map 1955
43.3.1 Unique ID (UID) 1955
43.3.2 JTAG Authentication Mode (AUTHMODEJ) 1956
43.3.3 JTAG Authentication Level 1 Plaintext ID (JID1P) 1956
43.3.4 JTAG Authentication Level 2 Plaintext ID (JID2P) 1956
43.3.5 SCI/USB Boot Mode Disable (SUDIS) 1957
43.3.6 Anti-rollback Counter (ARBC) 1957
43.3.7 Part Number (PN) 1958
43.3.8 Product Version (VER) 1958
43.3.9 SHOSTIF Boot Mode Setting (SHBMS) 1958
43.3.10 User Area (USERAREA) 1959
43.4 Operation 1959
43.4.1 Write procedure 1959
43.4.2 Read procedure 1960
44. Serial Host Interface (SHOSTIF) 1961
44.1 Overview 1961
44.2 Register Descriptions 1962
44.2.1 CTRLR0 : Control Register 0 1962
44.2.2 ENR : Enable Register 1963
44.2.3 RXFBTR : Receive FIFO Burst Threshold Register 1963
44.2.4 TXFTLR : Transmit FIFO Threshold Level Register 1964
44.2.5 RXFTLR : Receive FIFO Threshold Level Register 1964

44.2.6 SR : Status Register 1965
44.2.7 IMR : Interrupt Mask Register 1965
44.2.8 ISR : Interrupt Status Register 1966
44.2.9 RISR : Raw Interrupt Status Register 1967
44.2.10 TXUICR : Transmit FIFO Underflow Interrupt Clear Register 1968
44.2.11 RXOICR : Receive FIFO Overflow Interrupt Clear Register 1968
44.2.12 SPIMECR : SPI Master Interrupt Clear Register 1969
44.2.13 AHBECR : AHB Error Clear Register 1969
44.2.14 ICR : Interrupt Clear Register 1970
44.2.15 SHCFG : SHOSTIF Configuration Register 1970
44.3 Operations 1971
44.3.1 Instruction Phase 1971
44.3.2 Address Phase 1972
44.3.3 Data Phase 1973
44.3.4 Bridge Operation 1973
44.3.5 Write Operation 1973
44.3.6 Read Operation 1975
44.3.7 External SPI Master Expectations 1979
44.3.8 AHB Master Interface 1980
44.3.9 Interrupt 1980
44.4 Usage Notes 1982
45. Mailbox and Semaphore (MBXSEM) 1983
45.1 Overview 1983
45.2 Register Descriptions 1984
45.2.1 SEMn : Semaphore Register n (n = 0 to 7) 1984
45.2.2 SEMRCEN : Semaphore Read Clear Enable Register 1985
45.2.3 MBXH2Cn : Host to CR52 Mailbox Register n (n = 0 to 3) 1985
45.2.4 MBXISETH2C : Host to CR52 Mailbox Interrupt Set Register 1986
45.2.5 MBXICLRH2C : Host to CR52 Mailbox Interrupt Clear Register 1987
45.2.6 MBXC2Hn : CR52 to Host Mailbox Register n (n = 0 to 3) 1987
45.2.7 MBXISETC2H : CR52 to Host Mailbox Interrupt Set Register 1988
45.2.8 MBXICLRC2H : CR52 to Host Mailbox Interrupt Clear Register 1989
45.3 Operation 1989
45.3.1 Semaphore 1989
45.3.2 Mailbox 1990
45.3.3 Interrupt 1990
46. Encoder Interface Subsystem 1991
46.1 Overview 1991
46.2 Register Descriptions 1993

46.2.1 ENCODER_CFG : Encoder IF Configuration Register 1993
46.2.2 BISS_STAT : BiSS-C Status Register 1994
46.2.3 HDSL_STAT : HIPERFACE DSL Status Register 1995
46.2.4 ENDAT_CFG : EnDat Configuration Register 1996
46.2.5 AFMT_STAT : A-Format Status Register 1997
46.2.6 HDSLn_OS_D : HIPERFACE_DSL (SSn) Online Status (Drive) Register (n = 0, 1) 1998
46.2.7 HDSL0_OS_1 : HIPERFACE_DSL (SS0) Online Status (Safe1) Register 1999
46.2.8 HDSL0_OS_2 : HIPERFACE_DSL (SS0) Online Status (Safe2) Register 2000
46.3 Usage Notes 2001
46.3.1 Release of Register Write Protection 2001
47. EnDat 2.2 (ENDAT) 2002
47.1 Overview 2002
47.2 Register Descriptions 2004
47.2.1 SEND : Send Register 2004
47.2.2 Receive Registers 2006
47.2.3 Configuration Registers 2008
47.2.4 STATR : Status Register 2011
47.2.5 INTMR : Interrupt Mask Register 2014
47.2.6 SWSTRBR : SW Strobe Register 2015
47.2.7 IDR : ID Register 2015
47.3 Operation 2015
47.3.1 Control Sequence 2015
47.3.2 EnDat Data Transfer 2016
47.3.3 Interrupt 2019
47.3.4 Timer for Sampling Rate 2019
47.3.5 Watchdog Timer 2020
48. BiSS-C (BISS) 2021
48.1 Overview 2021
48.2 Register Descriptions 2022
48.2.1 SCDATALn : Sensor Data L (n = 1 to 8) 2022
48.2.2 SCDATAHn : Sensor Data H (n = 1 to 8) 2023
48.2.3 RDATAn : Register Data n (n = 1 to 16) 2023
48.2.4 CNFSLVn : Configuration Slave n (n = 1 to 8) 2023
48.2.5 REGACC : Register Access 2024
48.2.6 CTRLCOMM : Control Communication 2025
48.2.7 MSTRCNF : Master Configuration 2027
48.2.8 CHCNF : Channel Configuration 2028
48.2.9 STINF : Status Information 2028
48.2.10 INSTREG : Instruction Register 2030

48.2.11 STINF2 : Status Information 2 2031
48.3 Operation 2032
48.3.1 BiSS Function 2032
48.3.2 BiSS-C 2033
48.3.3 Sensor Data 2034
48.3.4 Control Frame 2036
48.3.5 Data Acquisition 2037
48.4 Usage Notes 2038
48.4.1 The value to be set to SCDLEN in the CNFSVn register 2038
48.4.2 The storage of received CRC when SCDLEN = 33 bits or more 2038
49. HIPERFACE DSL (HDSL) 2039
49.1 Overview 2039
49.2 Register Descriptions 2041
49.2.1 Drive Interface Register 2041
49.2.2 Safe Channel 1 Interface Registers 2058
49.2.3 Safe Channel 2 Interface Registers 2065
49.3 Operation 2069
49.3.1 System Start 2069
49.3.2 Quality Monitoring 2070
49.3.3 Interrupt 2071
49.3.4 Fast Position 2072
50. A-Format (AFMT) 2074
50.1 Overview 2074
50.2 Register Descriptions 2075
50.2.1 COMMAND : Command Code Encoder Address Register 2075
50.2.2 ROMDATA : EEPROM Address and Write Data Register 2077
50.2.3 ID : ID Code Register 2077
50.2.4 TRGSEL : Trigger Select Register 2078
50.2.5 TXTRG : Transmission Trigger Register 2078
50.2.6 BRSEL : Baud Rate Setting Register 2078
50.2.7 ENCnRXDATAm : ENCn Receive Data m Register (n = 1 to 8, m = 0 to 7) 2079
50.3 Operation 2082
50.3.1 Basic Operation 2082
50.3.2 Command Configuration 2085
50.3.3 Interrupts 2085
50.3.4 Individual Transmission Mode 2085
50.3.5 Multiple transmission Mode 2086
50.4 Usage Note 2086
50.4.1 CDF20 Command Code 2086

51. Encoder Divided-Output (ENCOUT) Module 2087
51.1 Overview 2087
51.2 Register Descriptions 2089
51.2.1 CTL : Control Register 2089
51.2.2 STR : Start Register 2089
51.2.3 POSMAX_PERIOD : Maximum Position and Carrier Period Register 2090
51.2.4 OUTCNT : Output Count Register 2090
51.2.5 POSCNT : Position Count Register 2091
51.3 Operation 2091
51.3.1 Phase A and Phase B Output Format 2092
51.3.2 Phase Z Output Format 2092
51.3.3 Carrier Period Generation by Events from ELC 2092
51.3.4 PERIOD Setting Value 1 2092
51.3.5 PERIOD Setting Value 2 2093
51.3.6 Uniformization of Waveform Output Accumulation Error 2094
51.3.7 Setting Procedure 2095
51.3.8 Setting and Output Examples 2098

52. Electrical Characteristics 2100
52.1 Absolute Maximum Ratings 2100
52.2 Power Supply 2101
52.3 Power On/Off Sequence 2101
52.4 DC Characteristics 2103
52.5 AC Characteristics 2106
52.5.1 Clock Timing 2107
52.5.2 Reset, Interrtup, and Mode Timing 2109
52.5.3 Bus Timing 2111
52.5.4 DMAC Timing 2138
52.5.5 On-Chip Peripheral Module Timing 2139
52.6 USB Characteristics 2168
52.7 A/D Conversion Characteristics 2170
52.8 Temperature Sensor Characteristics 2171
52.9 Debug Interface Timing 2171

Appendix 1. Peripheral I/O Registers 2174
1.1 Peripheral Base Addresses 2174
1.2 Register Descriptions 2177

Appendix 2. Package Dimensions 2220

Revision History 2221

200/400/800 MHz, Single Arm® Cortex®-R52 on-chip FPU and NEON™, 1.0 MB of on-chip SRAM, Ethernet MAC, EtherCAT, USB 2.0 high-speed, CAN/CANFD, various communications interfaces such as an xSPI and ΔΣ interface, encoder interfaces, and security functions

Features

On-chip 32-bit Arm Cortex-R52 processor

- High-speed realtime control with operating frequency of 200/400/800 MHz
- On-chip Single 32-bit Arm Cortex-R52 (revision r1p2)
- Tightly coupled memory (TCM) with ECC
  - 512 KB/64 KB
- Instruction cache/data cache with ECC
  - 16 KB per cache
- High-speed interrupt
- The FPU supports addition, subtraction, multiplication, division, multiply-and-accumulate, and square-root operations at single-precision and double-precision.
- The NEON, Advanced SIMD, supports integer or single precision results.
- Harvard architecture with 8-stage pipeline
- Supports the memory protection unit (MPU)
- Arm CoreSight architecture, includes support for debugging through JTAG and SWD interfaces.

Low power consumption

- Standby mode and module stop function

On-chip SRAM

- 1.0 MB of the on-chip SRAM with ECC
- 150/200 MHz

Data transfer

- DMAC: 16 channels × 2 units

Event link controller

- Module operations can be started by event signals rather than by interrupt handlers.
- Linked operation of modules is available even while the CPU is in the standby state.

Reset and power supply voltage control

- Four reset sources including a pin reset

Clock functions

- External clock/oscillator input frequency: 25 MHz
- CPU clock frequency: 200/400/800 MHz or 150/300/600 MHz
- System clock frequency: 200 MHz or 150 MHz
- Low-speed on-chip oscillator (LOCO): 240 kHz

Safety functions

- Register write protection, input clock oscillation stop detection and CRC
- Master Memory Protection Unit (MPU)

Security functions (optional)

- Boot mode with security through encryption
- JTAG authentication
- Cryptologic accelerator
- TRNG

Encoder interfaces

- Up to 2 channels
- EnDat 2.2, BiSS-C, A-format, and HIPERFACE DSL-compliant interfaces
- Frequency-divided output from an encoder

Various communications interfaces

- Ethernet
  - EtherCAT slave Controller: 3 ports
  - Ethernet MAC: 1 port
- USB 2.0 high-speed host/functions: 1 channel
- CAN/CANFD (compliant with ISO11898-1): 2 channels
- SCI with 16-byte transmission and reception FIFOs: 6 channels
- I2C bus interface: 3 channel for transfer at up to 400 kbps
- SPI: 4 channels
- xSPI: 2 channels

External host interfaces

- Serial host interface (SHOSTIF)

External address space

- Buses for high-speed data transfer at up to 100 MHz
- Support for up to 4 CS areas
- 8- or 16-bit bus space is selectable per area

Up to 35 extended-function timers

- 16-bit × 8 + 32-bit MTU3 (9 channels), 32-bit GPT (18 channels): Input capture, output compare, PWM waveform output
- 16-bit CMT (6 channels), 32-bit CMTW (2 channels)

ΔΣ interface

- Up to 6 ΔΣ modulators are connectable externally.

Trigonometric function unit

- Simultaneous calculation of sine and cosine
- Simultaneous calculation of arctangent and hypot_k

12-bit A/D converter

- 12 bits × 2 unit (4 channels for unit 0 and 1)

Temperature sensor for measuring temperature within the chip

General-purpose I/O ports

- Input pull-up/pull-down
- The locations of input/output functions for peripheral modules are selectable from among multiple pins.

Operating temperature range

- Tj = -40 to +125°C

1. Overview

1.1 Outline of Specifications

The MPU is a high-performance ASSP that has Arm Cortex®-R52 processor with Floating-Point Unit (FPU) and NEON™. It incorporates integrated peripheral functions necessary for system configuration.

Table 1.1 CPU

Feature	Functional description
Arm® Cortex®-R52	• Single processor • Operating frequency – 200 MHz/400 MHz/800 MHz (in case of 200 MHz system clock) – 150 MHz/300 MHz/600 MHz (in case of 150 MHz system clock) • 32-bit CPU Cortex-R52 designed by Arm (core revision r1p2) • Address space: 4 GB • Instruction cache – 16 KB (with ECC) • Data cache – 16 KB (with ECC) • Tightly coupled memory (TCM) – ATCM: 512 KB (with ECC) 1 wait (0 wait selectable if less than 400 MHz) – BTCM: 64 KB (with ECC) 0 wait – CTCM: 0 KB (with ECC) • Instruction set: Arm v8-R architecture, so support includes Thumb® and Thumb-2 • Data arrangement – Instructions: Little endian – Data: Little endian • 2-stage memory protection unit (MPU)
FPU	• Supports addition, subtraction, multiplication, division, multiply-and-accumulate, and square-root operations at single- and double-precision. • Registers 64-bit single-word registers: 64 bits × 32 (can be used as 16 double-word registers: 128 bits × 16)
NEON	The Advanced SIMD supporting integer or single precision results

Table 1.2 Memory

Feature	Functional description
On-chip system SRAM with ECC	• Capacity: Up to 1.0 MB (512 KB × 2 units) (with ECC) • Operating frequency: 150 MHz/200 MHz • SEC-DED (single error correction/double error detection) Error injection supported
One-time programmable memory	• Overwrite protection • Redundancy support • ECC support • Available information – Unique ID – Authentication settings – Trimming data – Boot mode setting – User area

Table 1.3 System (1 of 2)

Feature	Functional description
Operating modes	The operating mode can be selected from the following seven boot modes: • xSPI0 boot mode (CS0 × 1 boot Serial Flash) • xSPI0 boot mode (CS0 × 8 boot Serial Flash) • 16-bit bus boot mode (CS0 NOR Flash) • xSPI1 boot mode (CS0 × 1 boot Serial Flash) • SHOSTIF boot mode • SCI boot mode • USB boot mode

Table 1.3 System (2 of 2)

Feature	Functional description
Clock generation circuit	• The input clock can be selected from an external clock or external resonator. • Detection of input clock oscillation stopping • The following clocks are generated: – CPU0 clock: System clock ×1, ×2, or ×4 – System clock: 150 or 200 MHz – High-speed peripheral module clock: 150 or 200 MHz – Middle-speed peripheral module clock: 75 or 100 MHz – Low-speed peripheral module clock: 37.5 or 50 MHz – ADC clock in the 12-bit A/D converter: 18.75 or 25 MHz – External bus clock: 100 MHz (max.) – Low-speed on-chip oscillator: 240 kHz (fixed)
Reset	RES# pin reset, software reset, error reset, CPU0 software reset
Low-power consumption function	• Standby mode (Cortex-R52) • Module stop function
Interrupt controller (ICU)	• Connect an interrupt to the GIC (Generic Interrupt Controller) for Cortex-R52 CPU0 • Connect an activating trigger to the DMAC and ELC • Peripheral function interrupts: 448 sources • External interrupts: 16 sources (IRQ0 to IRQ15 pins) • Software interrupts: 8 sources • Non-maskable interrupts: 1 sources • 16 levels specifiable for the order of priority in the GIC
Bus state controller (BSC)	• The external address space is divided into four areas (CS0, CS2, CS3, and CS5) for management. • The following features are configurable for each area independently: Bus size (8 or 16 bits): Available sizes depend on the area. Number of access wait cycles (different wait cycles can be specified for read and write access cycles in some areas). Idle wait cycle insertion (between same area access cycles or different area access cycles). Specifying the memory to be connected to each area enables direct connection to SRAM, SRAM with byte selection, SDRAM, and burst ROM (clocked synchronous or asynchronous). The address/data multiplexed I/O (MPX) interface is also available. • Outputs a chip select signal (CS0# to CS5#) according to the target area (CS assert or negate timing can be selected by software). • Connectable memory type for each area CS0: SRAM, burst ROM CS2: SRAM CS2 + CS3: SRAM, SDRAM (CS2 only for SDRAM is not supported) CS3: SRAM, SDRAM CS5: SRAM, MPX-IO • SDRAM refresh Auto refresh or self-refresh mode selectable • SDRAM burst access

Table 1.4 Direct memory access

Feature	Functional description
Direct memory access controller (DMAC)	• 2 unit (16 channels each unit) • Transfer modes: Single transfer mode and block transfer mode • Transfer size - Unit 0: 1/2/4/8/16/32/64 bytes - Unit 1: 1/2/4/8/16/32 bytes • Activation sources: Software trigger, external DMA requests (DREQ), external interrupts, and interrupt requests from peripheral functions

Table 1.5 I/O Ports

Feature	Functional description
General-purpose I/O ports	• 196-pin FBGA – I/O pins: 110 – Input pins: 1 – Pull-up/pull-down resistors: 111 • The locations of input/output functions are selectable from among multiple pins.

Table 1.6 Event link

Feature	Functional description
Event link controller (ELC)	● Up to 213 event signals can be interlinked with the operation of modules. ● In particular, the operation of timer modules can be started by input event signals. ● Event-linked operation of signals of ports 16 and port 18 is to be possible.

Table 1.7 Timers (1 of 2)

Feature	Functional description
Multi-function timer pulse unit 3 (MTU3)	● 9 channels (16 bits × 8 channels, 32 bits × 1 channel) ● Maximum of 28 pulse-input/output and 3 pulse-input possible ● Select from among 10, 11, 12, or 14 counter-input clock signals for each channel (with maximum operating frequency of 200 MHz) ● Input capture function ● 39 output compare/input capture registers ● Counter clear operation (synchronous clearing by compare match/input capture) ● Simultaneous writing to multiple timer counters (TCNT) ● Simultaneous register input/output by synchronous counter operation ● Buffered operation ● Support for cascade-connected operation ● Automatic transfer of register data ● Pulse output mode Toggle/PWM/complementary PWM/reset-synchronized PWM ● Complementary PWM output mode - Outputs non-overlapping waveforms for controlling 3-phase inverters - Automatic specification of dead times - PWM duty cycle: Selectable as any value from 0% to 100% - Delay can be applied to requests for A/D conversion. - Non-generation of interrupt requests at peak or trough values of counters can be selected. - Double buffer configuration ● Reset synchronous PWM mode Six phases of positive and negative PWM waveforms can be output with desired duty cycles. ● Phase-counting mode: 16-bit mode (channels 1 and 2), 32-bit mode (channels 1 and 2 in cascade connection) ● Counter functionality for dead-time compensation ● Generation of triggers for A/D converter conversion ● A/D converter start triggers can be skipped ● Digital noise filter function for signals on the input capture and external counter clock pins ● Event linking by the ELC

Table 1.7 Timers (2 of 2)

Feature	Functional description
General PWM timer (GPT)	• 32 bits × 18 channels • Counting up or down (saw-wave), counting up and down (triangle-wave) selectable for all channels • Select from among four counter-input clock signals for each channel (with maximum operating frequency of 400 MHz for LLPP) • 2 input/output pins per channel • 2 output compare/input capture registers per channel • For the 2 output compare/input capture registers of each channel, 4 registers are provided as buffer registers and are capable of operating as comparison registers when buffering is not in use. • In output compare operation, buffer switching can be at peaks or troughs, enabling the generation of laterally asymmetrically PWM waveforms. • Registers for setting up frame intervals on each channel (with capability for generating interrupts on overflow or underflow) • Synchronizable operation of the several counters • Modes of synchronized operation (synchronized, or displaced by desired times for phase shifting) • Generation of dead times in PWM operation • Through combination of 3 counters, generation of automatic 3-phase PWM waveforms incorporating dead times • Starting, clearing, stopping, switching, up/down counters, and input capture in response to external or internal triggers • Starting, clearing, stopping, switching, up/down counters, and input capture in response to input level comparison • Internal trigger sources: software and compare-match • Generation of triggers for A/D converter conversion • Digital noise filter function for signals on the input capture and external trigger pins • Event linking by the ELC • Function of output duty 0% and 100% is selectable from troughs or crests/troughs. (all units)
Compare match timer (CMT)	• (16 bits × 2 channels) × 3 units • Select from among four counter-input clock signals for each channel
Compare match timer W (CMTW)	• (32 bits × 1 channel) × 2 units • Compare-match, input-capture input, and output-comparison output are available. • Select from among four counter-input clock signals for each channel • Interrupt requests can be output in response to compare-match, input-capture, and output-comparison events.
Watchdog timer (WDT)	• 14 bits × 1 channel • Select from among six counter-input clock signals for each channel
Port output enable 3 (POE3)	• Control of the high-impedance state of the MTU3 waveform output pins • 5 pins for input from signal sources: POE0#, POE4#, POE8#, POE10#, POE11# • Initiation on detection of short-circuited outputs (detection of simultaneous PWM output to the active level) • Initiation by input clock oscillation-stoppage detection, PLL oscillation anomaly detection, two types of DSMIF error detection, or software • Additional programming of output control target pins is enabled
Port output enable for GPT (POEG)	• Controlling the output disable for GPT waveform output • Initiation by input level detection of GTETRG pins • Initiation by output disable request from GPT • Initiation by detection of oscillation stop, two types of DSMIF error detection (unit in LLPP only), or by software
Real time clock (RTC)	• A 100 year calendar from 2000 to 2099 • BCD code display • Clock source is division of main oscillator. • Automatic adjustment function for leap years

Table 1.8 Communication interfaces (1 of 2)

Feature	Functional description
Ethernet MAC (GMAC)*2	• 1 port • IEEE802.3 • IEEE1588-2008 • IEEE802.3-az-2010 for EEE • Support for 10/100/1000 Mbps data transfer • Full duplex and half duplex are supported • Programmable frame length to support both standard and jumbo frames up to 16 KB • 17 MAC address registers for the address filter block • Variety of flexible addresses filtering modes are supported • Advanced IEEE 1588-2002 & 2008 Ethernet frame time-stamping supported • MII/RMII/RGMII interface is supported by RMII/RGMII converter • Timer module
EtherCAT slave controller (ESC)*1 *2	• 1 channel (3 ports) • EtherCAT Slave Controller IP core (made by Beckhoff Automation GmbH) implemented • MII interface is supported. MII/RMII interface is supported by RGMII converter in Ethernet Subsystem.
USB 2.0 HS host/function module	• 1 port • Compliance with the USB 2.0 specification • OTG support • Transfer rate High speed (480 Mbps), full speed (12 Mbps), low speed (1.5 Mbps, host only) • Communications buffer – Incorporates 1 KB of RAM for host mode – Incorporates 8 KB of RAM for function mode • DMAC (2 channels) incorporated
Serial communication interface (SCI)	• 6 channels • 5 communication mode – Asynchronous interfaces – 8-bit clock synchronous interface – Simple I2C (master-only) – Simple SPI – Smart card interface • Clock source is select from among four internal clock signals • Bit rate specifiable with the on-chip baud rate generator • Full-duplex and half-duplex communication • Data length: 7 to 9 bits (Asynchronous mode) • Bit rate modulation • Double speed mode (Asynchronous, Clock Synchronous, Simple SPI mode) • RS-485 driver control function (Asynchronous mode) • Loopback function to enable self-diagnosis (Asynchronous, Clock synchronous mode)
I^2C bus interface (IIC)	• 3 channels • Communication formats: I2C bus format or SMBus format • Master or slave mode selectable • Supports the multi-master • Maximum transfer rate: 400 kbps (Standard mode and Fast mode)
CAN-FD module (CANFD)*3	• 2 channels • Comply with CAN-FD ISO 11898-1 (2015) • Communication speed – Classical CAN mode: 1 Mbps – CAN FD mode: Nominal bit rate: max. 1 Mbps Data bit rate: max. 8 Mbps • Total 192 message buffers (in case frame size is 76 bytes) – Individual buffers: 64 for TX – Shared buffers: 128 for TX and RX including FIFO • Selectable ID type with 11-bit Standard and 18-bit Extended • Selectable Frame type: Data Frame and Remote Frame • Up to 256 receive rules

Table 1.8 Communication interfaces (2 of 2)

Feature	Functional description
Serial peripheral interface (SPI)	• 4 channels • SPI transfer facility Using the MOSI (master out slave in), MISO (master in slave out), SSL (slave select), and RSPCK (SPI clock) signals enables serial transfer through SPI operation (four lines) or clock-synchronous operation (three lines) capable of handling serial transfer as a master or slave • Data formats – Switching between MSB first and LSB first – Transfer bit length selectable to 4 - 32 bits – 32 bits × 4-stage FIFO transmit and receive buffers – Up to four frames can be transmitted or received in a single transfer operation (with each frame having up to 32 bits) • RSPCK can be stopped automatically with the reception buffer full for master reception
Expanded serial peripheral interface (xSPI)	• 2 channels • Comply with JESD251 • Multiple slave up to 2 slaves • Protocol mode: 1/4/8pin with SDR/DDR 1S-1S-1S, 4S-4D-4D, 8D-8D-8D • Support OctaFlash, OctaRAM, HyperFlash and HyperRAM • Protocol mode: 2/4pin with SDR compatible with QSPI 1S-2S-2S, 2S-2S-2S 1S-4S-4S, 4S-4S-4S • Configurable address length • Configurable initial access latency cycle • Support XiP mode • Support up to 256 MB address space • Prefetch function for burst-read with low latency • Outstanding buffer for burst-write with high throughput • Manual command configurable up to 4 commands • Output clock/input strobe port timing shift • Automatic command after released reset: up to 4 commands • 1.8 V/3.3 V selectable for xSPI0 and 3.3 V fix for xSPI1

Note 1. EtherCAT is a registered trademark of Beckhoff Automation GmbH, Germany.
Note 2. The product has a device which do not support GMAC and ESC.
Note 3. The product have a device with Classical CAN mode only.

Table 1.9 Analog

Feature	Functional description
12-bit A/D converter (ADC12)	• 12 bits × 2 units (unit 0: 4 channels, unit 1: 4 channels) • 12-bit resolution • Conversion time 0.84 µs per channel • Operating mode Scan mode (single scan mode, continuous scan mode, or 3 group scan mode) Group priority control • Sample-and-hold function Common sample-and-hold circuit included In addition, channel-dedicated sample-and-hold function (3 channels: in both unit 0 and unit 1) included • Sampling variable Sampling time can be set up for each channel • Double trigger mode (A/D conversion data duplicated) • Three ways to start A/D conversion Software trigger, timer (MTU3, ELC) trigger, external trigger • Event linking by the ELC
Temperature sensor unit (TSU)	• 1 channel • Relative precision: ±1°C (typ)

Table 1.10 Hardware accelerator for industrial interfaces

Feature	Functional description
ΔΣ interface (DSMIF)	• 3 channels × 2 units • Selectable 2 inputs (U/V) or 3 inputs (U/V/W) • Up to 6 ΔΣ modulators are externally connectable • Sinc filter can be selected as first, second or third order • Direct error connection to POE3 and POEG • Enhancement of current error detection
Trigonometric function unit (TFU)	Calculation of sine, cosine, arctangent, hypot_k ($\sqrt{x^2+y^2}/k$) • Simultaneous calculation of sine and cosine • Simultaneous calculation of arctangent and hypot_k
Encoder interfaces	• EnDat 2.2 (2 units) • BiSS-C (2 units) • A-format (2 units) • HIPERFACE DSL (2 units) • ENCOUT (1 unit)

Table 1.11 Safety

Feature	Functional description
Memory protection unit (MPU)	• Cortex-R52 MPU Two stages MPUs (EL2 and EL1) 24 regions each MPU • Master MPU Memory protection for masters except Cortex-R52 (DMAC, USB, Ethernet MAC, CoreSight, SHOSTIF)
Register write protection function	Protects important registers from being overwritten for in case a program runs out of control.
CRC calculator (CRC)	• 2 channels • CRC code generation for arbitrary amounts of data in 8-, 16-, or 32-bit units • Select any of four generating polynomials: – $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$ (32-Ethernet) – $X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$ (CRC-32C) – $X^{16} + X^{15} + X^2 + 1$ (CRC-16) – $X^{16} + X^{12} + X^5 + 1$ (CRC-CCITT) – $X^8 + X^2 + X + 1$ (CRC-8)
Clock monitor circuit (CLMA)	• Monitors the abnormal output clock frequency from the input clock (main clock oscillator), PLL circuit, or low-speed on-chip oscillator. • Input clock oscillation stop detection: Available
Data operation circuit (DOC)	The function to compare, add, or subtract 16-bit data
Isolated peripherals	• Safety dedicated peripherals are available: – GPT: 4 ch – SCI: 1 ch – IIC: 1 ch – SPI: 1 ch – CRC: 1 unit – RTC: 1 unit – GPIO: Sharable with normal GPIO – On-chip system SRAM with ECC • They are mapped independently from normal peripherals so that access protection can be done by EL2 MPU.

Table 1.12 Security

Feature	Functional description
Security[*1]	• Secure boot • JTAG authentication • Cryptographic accelerators – Symmetric Cipher: AES 128/192/256 bits with CBC/ECB/CTR/GCM/XTS – Asymmetric Cipher: ECC 256 bits, RSA 1024/2048/3072 bits, RSAES-OAEP – Hash: SHA-1, SHA-2 – Message authentication: HMAC, CMAC, GMAC – Signature algorithms: ECDSA with NIST P-256, RSASSA-PSS, RSASSA-PKCS1 • TRNG

Note 1. For details, contact our sales representative.

Table 1.13 Debug

Feature	Functional description
Debugging interface	• CoreSight architecture designed by Arm • Debugging function by the JTAG/SWD interface, and trace function by the trace port interface

Table 1.14 External host interface

Feature	Functional description
Serial host interface (SHOSTIF)	• Serial communication is possible in slave mode. • Supported interface – Motorola Serial Peripheral Interface (4-wire SPI) – Enhanced SPI Modes with Dual, Quad, or Octal SPI • Serial clock polarity switching • Serial clock phase switching • Single Data Transfer • Data size is up to 32 bits × 64 burst
Mailbox and semaphore (MBXSEM)	• Eight semaphores • Four 32-bit mailboxes for both external host CPU to Cortex-R52 and Cortex-R52 to external host CPU • Interrupts can be generated and cleared from both external host CPU and Cortex-R52

Table 1.15 Others

Feature	Functional description
Power supply voltage	VDD = 1.1 V (Core) VCC18 = 1.8 V (PLL, USB, ADC, TSU) VCC33 = 3.3 V (I/O, USB) VCC1833 = 1.8 V (RGMII, xSPI) or 3.3 V (RMII/MII, xSPI[*1])
Operating temperature	Tj = −40 to +125°C
Packages	196 pin FBGA 12 × 12 mm, 0.8-mm pitch

Note 1. Maximum xSPI clock frequency is 75 MHz at 3.3 V.

1.2 Product Lineup

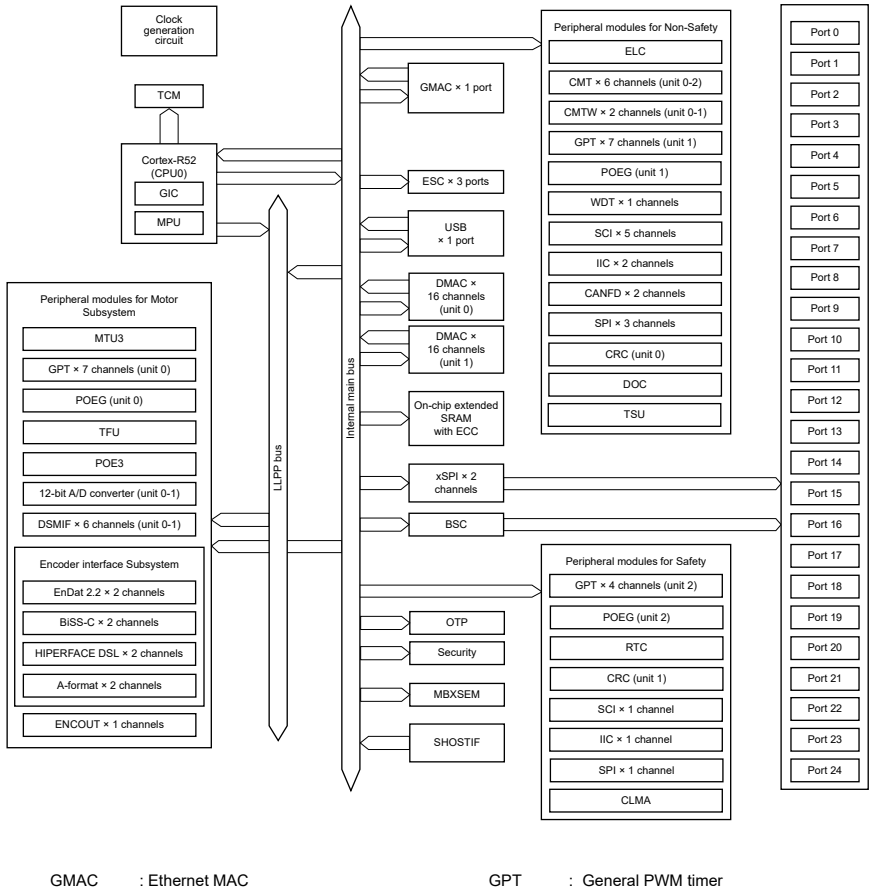
Table 1.16 shows a product lineup.

Table 1.16 Product lineup

Part Number	Package	CPU	System SRAM Capacity	CAN	EtherCAT	Security
R9A07G074M08GBG	PLBG0196GC-A	Single Cortex-R52	1.0 MB	CAN-FD	Available	Available
R9A07G074M05GBG	PLBG0196GC-A	Single Cortex-R52	1.0 MB	Classical CAN	Not available	Available
R9A07G074M04GBG	PLBG0196GC-A	Single Cortex-R52	1.0 MB	CAN-FD	Available	Not available
R9A07G074M01GBG	PLBG0196GC-A	Single Cortex-R52	1.0 MB	Classical CAN	Not available	Not available

1.3 Block Diagram

Figure 1.1 shows a block diagram.



GMAC : Ethernet MAC
ETHSW : Ethernet switch
ESC : EtherCAT slave controller
DMAC : DMA controller
BSC : Bus state controller
xSPI : Expanded serial peripheral interface
WDT : Watchdog timer
SCI : Serial communication interface
SPI : Serial peripheral interface
USB : USB 2.0 HS host/function module
GIC : Generic interrupt controller
MPU : Memory protection unit
ELC : Event link controller
MTU3 : Multi-function timer pulse unit 3
SHOSTIF : Serial host interface
MBXSEM : Mailbox and Semaphore

GPT : General PWM timer
CMT : Compare match timer
CMTW : Compare match timer W
POE3 : Port output enable 3
POEG : Port output enable for GPT
RTC : Real time clock
IIC : I^2C bus interface
CANFD : CANFD module
DSMIF : ΔΣ interface
TFU : Trigonometric function unit
CRC : CRC (cyclic redundancy check) calculator
CLMA : Clock monitor circuit
DOC : Data operation circuit
TSU : Temperature sensor unit
OTP : One-time programmable memory

Figure 1.1 Block diagram

1.4 Pin Functions

Table 1.17 lists the pin functions.

Table 1.17 Pin functions (1 of 7)

Classification	Pin name	I/O	Description
Power supply	VDD	Input	Power supply pin. Connect this pin to the system power supply.
	VSS	Input	Ground pin. Connect this pin to the system power supply (0 V).
	VCC1833_2 VCC1833_3	Input	Power supply pin for each I/O domains. (1.8 V or 3.3 V)
	VCC33	Input	Power supply pin for I/O pins.
	VCC18_PLL0, VCC18_PLL1	Input	Power supply pins for the on-chip PLL oscillator
	AVCC18_TSU	Input	Power supply pin for the temperature sensor unit
Clock	XTAL	Output	Connected to a crystal resonator. When external clock signal is used, EXTAL pin should be driven low. XTAL pin should never be driven or loaded by anything other than crystal oscillator. The voltage level of EXTAL must not exceed core VDD (1.16 V).
	EXTAL	Input	
	EXTCLKIN	Input	Inputs the external clock. When a crystal resonator is connected, it should be driven low.
	CKIO	Output	Outputs the external bus clock for external devices.
	ETH0_REFCLK	Output	Outputs 25 MHz clock for EtherPHY 0
	ETH1_REFCLK	Output	Outputs 25 MHz clock for EtherPHY 1
	ETH2_REFCLK	Output	Outputs 25 MHz clock for EtherPHY 2
	RMII0_REFCLK	Output	Outputs 50 MHz clock for RMII0
	RMII1_REFCLK	Output	Outputs 50 MHz clock for RMII1
	RMII2_REFCLK	Output	Outputs 50 MHz clock for RMII2
Operating mode control	MDX	Input	This signal should be driven low.
	MD0 to MD2	Input	Input the operating mode select signal. The signal level on these pins must not be changed during operation mode transition on release from the reset state.
	MDV2, MDV3	Input	Input the operating voltage select signal. The signal level on these pins must not be changed during operation mode transition on release from the reset state.
	MDW	Input	Input the ATCM wait cycle select signal. The signal level on this pin must not be changed during operation mode transition on release from the reset state.
	MDD	Input	Input the enabling JTAG authentication by hash signal. The signal level on this pin must not be changed during operation mode transition on release from the reset state.
System control	RES#	Input	Inputs the reset signal. This MPU enters the reset state when this signal goes low.
	BSCANP	Input	Inputs the boundary scan enable signal. Boundary scan is enabled when this pin goes high. When boundary scan is not used, this pin should be driven low.
	RSTOUT#	Output	Outputs the reset signal externally

Table 1.17 Pin functions (2 of 7)

Classification	Pin name	I/O	Description
Debugging interface	TRST#	Input	Test reset pin for the on-chip emulator
	TMS	I/O	Test mode select pin for the on-chip emulator Functions as the SWDIO pin in serial wire debug (SWD) mode
	TDI	Input	Test data input pin for the on-chip emulator
	TDO	Output	Test data output pin for the on-chip emulator
	TCK	Input	Test clock pin for the on-chip emulator Functions as the SWCLK pin in serial wire debug (SWD) mode
	TRACECLK	Output	Outputs the clock for synchronization with trace data
	TRACECTL	Output	Outputs the enable signal for trace control
	TRACEDATA0 to TRACEDATA7	Output	Output trace data
Bus state controller (BSC)	A25 to A0	Output	Output the address
	D15 to D0	I/O	Input and output the data
	CS0#, CS2#, CS3#, CS5#	Output	Output the chip select signal for the external memory or device.
	RD#	Output	Outputs the strobe signal which indicates a read is in progress.
	RD/WR#	Output	Outputs the strobe signal which indicates a read or write access
	BS#	Output	Outputs the status signal which indicates the start of the bus cycle
	AH#	Output	Outputs the address hold signal for the device that uses the multiplexed I/O interface
	WAIT#	Input	Inputs the external wait control signal which inserts a wait cycle into the bus cycle
	WE0#	Output	Outputs the write strobe signal to D7 to D0
	WE1#	Output	Outputs the write strobe signal to D15 to D8
	DQMLL	Output	Outputs the data mask enable signal to D7 to D0 when SDRAM is connected
	DQMLU	Output	Outputs the data mask enable signal to D15 to D8 when SDRAM is connected
	RAS#	Output	Outputs the row-address strobe signal to the SDRAM. This pin should be connected to the RAS# pin on the SDRAM.
	CAS#	Output	Outputs the column-address strobe signal to the SDRAM. This pin should be connected to the CAS# pin on the SDRAM.
	CKE	Output	Outputs the clock enable signal to the SDRAM. This pin should be connected to the CKE pin on the SDRAM.
Direct memory access controller (DMAC)	DREQ	Input	Inputs the DMA transfer request signal from the external device
	DACK	Output	Outputs the acknowledge signal which indicates acceptance of the DMA transfer request from the external device
	TEND	Output	Outputs the DMA transfer end signal
Interrupt	NMI	Input	Inputs the non-maskable interrupt request signal
	IRQ0 to IRQ15	Input	Input the external interrupt request signal

Table 1.17 Pin functions (3 of 7)

Classification	Pin name	I/O	Description
Multi-function timer pulse unit 3 (MTU3)	MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D	I/O	TGRA0 to TGRD0 input capture input, output compare output, and PWM output pins
	MTIOC1A, MTIOC1B	I/O	TGRA1 and TGRB1 input capture input, output compare output, and PWM output pins
	MTIOC2A, MTIOC2B	I/O	TGRA2 and TGRB2 input capture input, output compare output, and PWM output pins
	MTIOC3A, MTIOC3B, MTIOC3C, MTIOC3D	I/O	TGRA3 to TGRD3 input capture input, output compare output, and PWM output pins
	MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D	I/O	TGRA4 to TGRD4 input capture input, output compare output, and PWM output pins
	MTIC5U, MTIC5V, MTIC5W	Input	TGRU5, TGRV5, and TGRW5 input capture input and dead time compensation input pins
	MTIOC6A, MTIOC6B, MTIOC6C, MTIOC6D	I/O	TGRA6 to TGRD6 input capture input/output compare output/PWM output pins
	MTIOC7A, MTIOC7B, MTIOC7C, MTIOC7D	I/O	TGRA7 to TGRD7 input capture input/output compare output/PWM output pins
	MTIOC8A, MTIOC8B, MTIOC8C, MTIOC8D	I/O	TGRA8 to TGRD8 input capture input/output compare output/PWM output pins
	MTCLKA, MTCLKB, MTCLKC, MTCLKD	Input	External clock input pins for MTU3
Port output enable 3 (POE3)	POE0#, POE4#, POE8#, POE10#, POE11#	Input	Input the request signal to place the MTU3 in the high-impedance state
General PWM timer (GPT)/ Port output enable for GPT (POEG)	GTETRGA, GTETRGB, GTETRGC, GTETRGD	Input	External trigger input and output-disable request input pins
	GTETRGSA, GTETRGSB	Input	External trigger input and output-disable request input pins (SAFETY)
	GTIOC0A to GTIOC17A, GTIOC0B to GTIOC17B	I/O	Input capture input/output compare output/PWM output pins
	GTADSML0, GTADSML1, GTADSMP0, GTADSMP1	Output	Output pins for monitoring A/D conversion start requests
Compare match timer W (CMTW)	CMTW0_TIC0, CMTW0_TIC1, CMTW1_TIC0, CMTW1_TIC1	Input	CMTW input capture input pins
	CMTW0_TOC0, CMTW0_TOC1, CMTW1_TOC0, CMTW1_TOC1	Output	CMTW output compare output pins
Real time clock (RTC)	RTCAT1HZ	Output	RTC 1 Hz output pin

Table 1.17 Pin functions (4 of 7)

Classification	Pin name	I/O	Description
Serial communication interface (SCI)	SCK0 to SCK5	I/O	Clock I/O pins (clock synchronous mode/simple SPI mode/smart card mode)
	RXD0 to RXD5	Input	Input the receive data (asynchronous mode/clock synchronous mode/smart card mode)
	TXD0 to TXD5	Output	Output the transmit data (asynchronous mode/clock synchronous mode/smart card mode)
	CTS0# to CTS5#	Input	Input the start of transmission (asynchronous mode/ clock synchronous mode) active-low
	RTS0# to RTS5#	Output	Output the reception (asynchronous mode/clock synchronous mode) active-low
	SCL0 to SCL5	I/O	Input/output the I2C clocks (simple I2C mode)
	SDA0 to SDA5	I/O	Input/output the I2C data (simple I2C mode)
	MISO0 to MISO5	I/O	Input/output the data for slave transmission (simple SPI mode)
	MOSI0 to MOSI5	I/O	Input/output the data for master transmission (simple SPI mode)
	SS0# to SS5#	Input	Chip-select input pins (simple SPI mode) active-low
	DE0 to DE5	Output	Driver enable output pins (asynchronous mode)
I2C bus interface (IIC)	IIC_SCL0 to IIC_SCL2	I/O	Clock I/O pins
	IIC_SDA0 to IIC_SDA2	I/O	Data I/O pins
Ethernet (RGMII is supported for port-2 only.)	ETH0_TXCLK to ETH2_TXCLK	I/O	TX clock input pins (MII mode) TX clock output pins (RGMII mode)
	ETH0_TXD0 to ETH2_TXD0	Output	TX data 0 pins (RGMII, RMII, and MII modes)
	ETH0_TXD1 to ETH2_TXD1	Output	TX data 1 pins (RGMII, RMII, and MII modes)
	ETH0_TXD2 to ETH2_TXD2	Output	TX data 2 pins (RGMII and MII modes)
	ETH0_TXD3 to ETH2_TXD3	Output	TX data 3 pins (RGMII and MII modes)
	ETH0_TXEN to ETH2_TXEN	Output	TX data enable pins (RMII and MII modes) TX data enable/TX data error (TX_CTL) pins (RGMII mode)
	ETH2_TXER	Output	TX data error pins (MII mode)
	ETH0_RXCLK to ETH2_RXCLK	Input	RX clock pins (RGMII, RMII, and MII modes)
	ETH0_RXD0 to ETH2_RXD0	Input	RX data 0 pins (RGMII, RMII, and MII modes)
	ETH0_RXD1 to ETH2_RXD1	Input	RX data 1 pins (RGMII, RMII, and MII modes)
	ETH0_RXD2 to ETH2_RXD2	Input	RX data 2 pins (RGMII and MII modes)
	ETH0_RXD3 to ETH2_RXD3	Input	RX data 3 pins (RGMII and MII modes)
	ETH0_RXDV to ETH2_RXDV	Input	RX data valid pins (MII mode) Carrier sense/RX data valid (CRS_DV) pins (RMII mode) RX data valid/RX error (RX_CTL) pins (RGMII mode)
	ETH0_RXER to ETH2_RXER	Input	RX data error pins (RMII and MII modes)
	ETH2_CRS	Input	Carrier sense pins (MII mode)
	ETH2_COL	Input	Collision detection pins (MII mode)

Table 1.17 Pin functions (5 of 7)

Classification	Pin name	I/O	Description
Ethernet MAC (GMAC)	GMAC_PTPTRG0	Input	PTP timer trigger external input 0
	GMAC_PTPTRG1	Input	PTP timer trigger external input 1
	GMAC_PTPOUT0 to GMAC_PTPOUT3	Output	PTP timer pulse output
	GMAC_MDC	Output	Management data clock output pin
	GMAC_MDIO	I/O	Management data I/O pin
EtherCAT slave controller (ESC)	ESC_LEDRUN	Output	Outputs the EtherCAT RUN LED signal
	ESC_IRQ	Output	Outputs the EtherCAT IRQ signal
	ESC_LEDSTER	Output	Outputs the EtherCAT Dual-color state LED signal
	ESC_LEDERR	Output	Outputs the EtherCAT error LED signal
	ESC_LINKACT0 to ESC_LINKACT2	Output	Output the EtherCAT link/activity LED signal
	ESC_SYNC0, ESC_SYNC1	Output	Output the EtherCAT SYNC signal
	ESC_LATCH0, ESC_LATCH1	Input	Input the EtherCAT LATCH signal
	ESC_RESETOUT#	Output	Output the EtherCAT reset signal
	ESC_I2CCLK	Output	Outputs the EtherCAT EEPROM I2C clock signal
	ESC_I2CDATA	I/O	Inputs and outputs the EtherCAT EEPROM I2C data signal
	ESC_PHYLINK0 to ESC_PHYLINK2	Input	Inputs the EtherCAT PHY link status signal.
	ESC_MDC	Output	Management data clock output pin
	ESC_MDIO	I/O	Management data I/O pin
USB 2.0 host/function module	VCC33_USB	Input	Power supply input pin for USB
	VCC18_USB	Input	Power supply input pin for USB
	VSS_USB	Input	Ground input pins for USB
	AVCC18_USB	Input	Analog power supply input pin for USB
	USB_RREF	Input	Reference current input pin for USB. Connect this pin to the VSS_USB pin with 1.8 kΩ (±1%).
	USB_DP	I/O	USB bus D+ data I/O pin
	USB_DM	I/O	USB bus D- data I/O pin
	USB_VBUSEN	Output	Outputs the VBUS power enable signal for USB
	USB_OVRCUR	Input	Inputs the overcurrent signal for USB
	USB_VBUSIN	Input	USB cable connection/disconnection detection input pin
	USB_EXICEN	Output	OTG power supply IC control pin
	USB_OTGID	Input	OTG ID pin
CANFD module (CANFD)	CANRX0, CANRX1	Input	Receive data input pins
	CANTX0, CANTX1	Output	Transmit data output pins
	CANRXDP0, CANRXDP1	Output	Receive data phase output pins
	CANTXDP0, CANTXDP1	Output	Transmit data phase output pins

Table 1.17 Pin functions (6 of 7)

Classification	Pin name	I/O	Description
Serial peripheral interface (SPI)	SPI_RSPCK0 to SPI_RSPCK3	I/O	Clock I/O pins
	SPI_MOSI0 to SPI_MOSI3	I/O	Master transmit data I/O pins
	SPI_MISO0 to SPI_MISO3	I/O	Slave transmit data I/O pins
	SPI_SSL00 to SPI_SSL30	I/O	Slave select signal I/O pins
	SPI_SSL01 to SPI_SSL31, SPI_SSL02 to SPI_SSL32, SPI_SSL03 to SPI_SSL33	Output	Slave select signal output pins
Expanded serial peripheral interface (xSPI)	XSPI0_CKP, XSPI1_CKP, XSPI0_CKN	Output	Clock output pins
	XSPI0_CS0#, XSPI0_CS1#, XSPI1_CS0#	Output	Chip select output pins
	XSPI0_DS, XSPI1_DS	I/O	Read data strobe/write data mask input/output pin
	XSPI0_IO0 to XSPI0_IO7, XSPI1_IO0 to XSPI1_IO7	I/O	Data0 to Data7 input/output pins
	XSPI0_RESET0#, XSPI0_RESET1#	Output	Master reset status output pins
	XSPI0_RSTO0#, XSPI0_RSTO1#	Input	Slave reset status input pins
	XSPI0_INT0#, XSPI0_INT1#	Input	Interrupt input pins
	XSPI0_ECS0#, XSPI0_ECS1#	Input	Error correction status input pins
	XSPI0_WP0#, XSPI0_WP1#	Output	Write protect output pins
ΔΣ interface (DSMIF)	MCLK0 to MCLK5	I/O	Clock I/O pins
	MDAT0 to MDAT5	Input	Data input pins
12-bit A/D converter (ADC12)	AN000 to AN003, AN100 to AN103	Input	Analog input pins for the A/D converter
	ADTRG0#, ADTRG1#	Input	External trigger input pins for the start of A/D conversion
Analog power supply	VCC18_ADC0, VCC18_ADC1	Input	Analog power supply input pin for the 12-bit A/D converter. Connect this pin to the 1.8 V power supply if the 12-bit A/D converter is not to be used.
	VREFH0, VREFH1	Input	Reference voltage input pin for the 12-bit A/D converter. Connect this pin to the 1.8 V power supply if the 12-bit A/D converter is not to be used.
I/O ports	P00_0 to P24_2	I/O	General-purpose input/output pins
Serial host interface (SHOSTIF)	HSPI_CK	Input	Clock input pin
	HSPI_CS#	Input	Chip select input pin
	HSPI_IO0 to HSPI_IO7	I/O	Data0 to Data7 input/output pins
	HSPI_INT#	Output	Interrupt output pin
Mailbox	MBX_HINT#	Output	Mailbox (Cortex-R52 to Host) interrupt output pin
Encoder I/F common	ENCIFCK0, ENCIFCK1	Output	Encoder I/F clock output pins
	ENCIFOE0, ENCIFOE1	Output	Encoder I/F data output enable pins
	ENCIFDO0, ENCIFDO1	Output	Encoder I/F data output pins
	ENDIFDI0, ENDIFDI1	Input	Encoder I/F data input pins
EnDat 2.2 (ENDAT)	DUEI0, DUEI1	Output	EnDat 2.2 Data transfer
	TST_OUT0, TST_OUT1	Output	EnDat 2.2 Data input after internal synchronization
	SI0#, SI1#	Output	EnDat 2.2 Start pulse

Table 1.17 Pin functions (7 of 7)

Classification	Pin name	I/O	Description
Hiperface DSL (HDSL)	HSDL0_LINK, HSDL1_LINK	Output	HDSL LINK
	HDSL0_SMPL, HDSL1_SMPL	Output	HDSL Test signal line sampler
	HDSL0_CLK1, HDSL1_CLK1	Input	HDSL SPI clock safe 1
	HDSL0_SEL1, HDSL1_SEL1	Input	HDSL SPI selection safe 1
	HDSL0_MISO1, HDSL1_MISO1	Output	HDSL SPI data output safe 1
	HDSL0_MOSI1, HDSL1_MOSI1	Input	HDSL SPI data input safe 1
	HDSL0_CLK2, HDSL1_CLK2	Input	HDSL SPI clock safe 2
	HDSL0_SEL2, HDSL1_SEL2	Input	HDSL SPI selection safe 2
	HDSL0_MISO2, HDSL1_MISO2	Output	HDSL SPI data output safe 2
	HDSL0_MOSI2, HDSL1_MOSI2	Input	HDSL SPI data input safe 2
ENCOUT	POUTA	Output	ENCOUT A-phase output pin
	POUTB	Output	ENCOUT B-phase output pin
	POUTZ	Output	ENCOUT C-phase output pin

1.5 FBGA 196 Pin Assignments

Figure 1.2 shows the pin arrangement. Table 1.18 shows the lists of pins and the pin functions.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	
A	VSS	P01_3	P00_3	P24_1	P22_3	P22_2	P21_6	P21_1	P20_4	VSS	AN100	AN002	AN000	VSS	A
B	P01_7	P01_5	P01_2	P00_1	P23_7	P21_7	P21_5	P21_2	VSS	AN103	AN101	AN001	VSS	P18_6	B
C	P02_1	P01_6	P01_1	P00_2	P24_2	P22_0	P21_3	P20_3	VSS	AN102	AN003	AVCC18_TSU	P18_5	P18_4	C
D	P02_3	P02_0	P01_0	P00_6	P24_0	P22_1	P21_4	VCC33	VREFH1	VREFH0	VSS	P18_2	P18_3	P18_1	D
E	P02_5	P02_2	P02_4	P01_4	P00_0	VCC1833_2	VDD	VDD	VCC18_ADC1	VCC18_ADC0	VSS	P17_4	P17_7	P18_0	E
F	P04_1	P02_7	P02_6	TRST#	VCC33	VDD	VSS	VSS	VDD	VCC33	P17_0	P17_6	P17_3	P17_5	F
G	P06_0	P05_6	P05_5	BSCANP	VDD	VSS	VSS	VSS	VSS	VDD	P16_1	P16_0	P16_2	P16_3	G
H	P06_4	P06_2	P06_1	P05_7	VDD	VSS	VSS	VSS	VSS	VDD	P15_6	P15_5	P15_4	P15_7	H
J	P06_7	P06_5	P06_6	P06_3	VCC33	VDD	VSS	VSS	VDD	VCC1833_3	P15_3	P15_0	P15_1	P15_2	J
K	P07_0	P07_2	P07_3	P07_1	P09_0	VDD	VDD	VDD	VCC33	VCC1833_3	P14_2	P14_7	P14_4	P14_6	K
L	VSS	P08_5	P08_4	P09_4	P10_1	VCC33	P07_4	VDD	P13_2	P13_3	P13_7	P13_5	P14_3	P14_5	L
M	P08_7	P08_6	P09_1	P09_7	P10_2	VSS	MDX	VCC18_PLL1	VSS	VCC33_USB	VSS_USB	P13_4	P13_6	P14_1	M
N	P09_2	P09_3	P09_5	P10_0	RES#	EXT CLKIN	VSS	VCC18_PLL0	VSS_USB	VSS_USB	AVCC18_USB	VSS_USB	VSS_USB	P14_0	N
P	VSS	P09_6	P10_3	P10_4	VSS	EXTAL	XTAL	VSS	USB_DM	USB_DP	VCC18_USB	USB_RREF	VSS_USB	VSS	P
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	

Figure 1.2 Pin arrangement (196-pin FBGA) (top view)

Table 1.18 List of pins and pin functions (196-pin FBGA) (1 of 8)

Pin number	I/O power domain	Power supply clock system control	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI)	DSMIF, Mailbox	Interrupt	ADC12	Host interface (SHOST)	Encoder I/F
A1	—	VSS	—	—	—	—	—	—	—	—	—

Table 1.18 List of pins and pin functions (196-pin FBGA) (2 of 8)

Pin number	I/O power domain	Power supply clock system control	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI)	DSMIF, Mailbox	Interrupt	ADC12	Host interface (SHOST)	Encoder I/F
A2	VCC1833_2	—	P01_3	AH#	MTIOC4D / GTIOC3B	ETH2_TXD2	—	IRQ2	—	—	POUTZ
A3	VCC1833_2	—	P00_3	RD/WR#	MTIC5W	SS2# / CTS2# / RTS2# / ETH2_REFCLK / RMII2_REFCLK	—	IRQ1	—	—	SI1# / HDSL1_CLK1
A4	VCC1833_2	—	P24_1	D13 / CAS#	MTIOC0C / POE8# / GTETRGC	ETH2_RXCLK	MCLK5	—	—	—	—
A5	—	—	P22_3	D10	MTIOC8D / GTETRGSB	RXD5 / SCL5 / MISO5 / USB_VBUSEN / SPI_SSL00	—	—	—	—	ENCIFDI1
A6	—	TRACECLK	P22_2	D9	MTIOC8C / GTETRGSA	SPI_SSL12	MCLK1	IRQ4	—	—	ENCIFDO1
A7	—	TRACEDATA5	P21_6	D5 / TEND	MTIOC7B / GTIOC16B	CTS0# / ESC_LINKACT0	MDAT2	IRQ9	—	HSPI_IO4	—
A8	—	TRACEDATA0	P21_1	D0	MTIOC6A / GTIOC14A / CMTW0_TIC0	SCK5 / IIC_SCL1 / ESC_SYNC0 / ESC_SYNC1 / SPI_SSL20	MCLK0	—	—	HSPI_INT#	—
A9	—	MDV3	P20_4	—	—	GMAC_PTPOUT0 / ESC_LINKACT1	—	—	—	—	—
A10	—	VSS	—	—	—	—	—	—	—	—	—
A11	—	—	—	—	—	—	—	—	AN100	—	—
A12	—	—	—	—	—	—	—	—	AN002	—	—
A13	—	—	—	—	—	—	—	—	AN000	—	—
A14	—	VSS	—	—	—	—	—	—	—	—	—
B1	—	TRACEDATA1	P01_7	A19	MTIOC1B / GTIOC9B	SCK1 / ETH2_TXER / CANRX0 / SPI_RSPCK3	—	—	ADTRG0#	—	ENCIFOE0
B2	VCC1833_2	—	P01_5	WE0# / DQMLL / CS0#	MTIOC4A / GTIOC2A	ETH2_TXD0 / SPI_RSPCK1	—	—	—	—	TST_OUT0 / HDSL0_SMPL
B3	VCC1833_2	—	P01_2	CS2#	MTIOC4B / GTIOC2B	ETH2_TXD3	—	IRQ2	—	—	POUTB
B4	VCC1833_2	—	P00_1	A13	MTIC5U	RXD2 / SCL2 / MISO2 / ETH2_RXDV	MCLK0	IRQ0	—	—	—
B5	VCC1833_2	—	P23_7	D11 / BS	MTIOC0A / GTETRGA	SCK1 / ETH2_RXD0	MCLK4	—	—	—	—
B6	—	TRACEDATA6	P21_7	D6 / DREQ	MTIOC7C / GTIOC17A	DE0	MCLK3	IRQ10	—	HSPI_IO5	—
B7	—	TRACEDATA4	P21_5	D4	MTIOC7A / GTIOC16A / CMTW1_TOC1	CTS5# / SPI_MISO0	MCLK2	IRQ6	ADTRG1#	HSPI_IO3	ENCIFDI0
B8	—	TRACEDATA1	P21_2	D1	MTIOC6B / GTIOC14B / GTIOC15A / CMTW0_TIC1	RXD5 / SCL5 / MISO5 / IIC_SDA1 / ESC_SYNC0 / ESC_SYNC1 / SPI_MISO2	MDAT0	—	—	HSPI_CS#	—
B9	—	VSS	—	—	—	—	—	—	—	—	—
B10	—	—	—	—	—	—	—	—	AN103	—	—
B11	—	—	—	—	—	—	—	—	AN101	—	—
B12	—	—	—	—	—	—	—	—	AN001	—	—
B13	—	VSS	—	—	—	—	—	—	—	—	—
B14	—	TRACECLK	P18_6	A15	MTIC5W	SCK4 / DE4 / IIC_SCL2 / GMAC_PTPOUT2 / SPI_MISO2 / XSPI1_IO7	—	IRQ11	ADTRG0#	HSPI_CK	SI0# / HDSL0_CLK1
C1	—	MDW	P02_1	A17	—	DE1 / GMAC_PTPOUT1 / ESC_SYNC0 / ESC_SYNC1	—	—	—	—	HDSL0_MISO1

Table 1.18 List of pins and pin functions (196-pin FBGA) (3 of 8)

Pin number	I/O power domain	Power supply clock system control	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI)	DSMIF, Mailbox	Interrupt	ADC12	Host interface (SHOST)	Encoder I/F
C2	—	TRACEDATA0	P01_6	A20	MTIOC1A / MTIOC3A / GTIOC0A / GTIOC9A	CTS1# / GMAC_PTPTRG1 / ESC_LATCH1 / ESC_LATCH0 / ESC_PHYLINK0 / CANTXDP1	—	—	—	—	SI0# / HDSL0_CLK1
C3	VCC1833_2	—	P01_1	CKE	MTIOC3D / GTIOC1B / GTETRGC	DE2 / GMAC_MDC / ESC_MDC / ESC_PHYLINK2	MDAT1	—	—	—	POUTA
C4	VCC1833_2	—	P00_2	RD#	MTIC5V	TXD2 / SDA2 / MOSI2 / ETH2_TXEN / USB_OVRCUR	—	—	—	—	TST_OUT1 / HDSL0_MOSI2
C5	VCC1833_2	—	P24_2	D14 / RAS#	MTIOC0D / GTETRGD	TXD1 / SDA1 / MOSI1 / ETH2_RXD2	MDAT5	—	—	—	—
C6	—	TRACEDATA7	P22_0	D7	MTIOC7D / GTIOC17B	DE5	MDAT3	IRQ15	—	HSPI_IO6	—
C7	—	TRACEDATA2	P21_3	D2	MTIOC6C / GTIOC14B / GTIOC15A	TXD5 / SDA5 / MOSI5 / ESC_LEDRUN / ESC_LEDSTER / SPI_SSL33	MCLK1	NMI	—	HSPI_IO2	—
C8	—	MDV2	P20_3	—	—	GMAC_PTPOUT1 / ESC_LEDERR / CANTX1	—	—	—	—	—
C9	—	VSS	—	—	—	—	—	—	—	—	—
C10	—	—	—	—	—	—	—	—	AN102	—	—
C11	—	—	—	—	—	—	—	—	AN003	—	—
C12	—	AVCC18_TSU	—	—	—	—	—	—	—	—	—
C13	—	TRACECTL	P18_5	RAS#	MTIC5V	RXD4 / SCL4 / MISO4 / ETH2_COL / CANRX0 / SPI_MOSI2 / XSPI1_IO6	—	—	—	HSPI_IO0	TST_OUT0 / HDSL0_SMPL
C14	—	—	P18_4	CAS#	MTIC5U	TXD4 / SDA4 / MOSI4 / ETH2_RXER / CANTX0 / SPI_RSPCK2 / XSPI1_IO5	—	IRQ1	—	HSPI_IO1	DUEI0 / HDSL0_LINK
D1	—	—	P02_3	A15 / AH#	MTIOC2B / POE11# / GTIOC10B	SS1# / CTS1# / RTS1# / ETH2_COL / CANRX1 / SPI_SSL30	—	IRQ15	—	—	ENCIFDO0
D2	—	TRACEDATA2	P02_0	A18	MTIOC3C / GTIOC0B / GTADSML0	RXD1 / SCL1 / MISO1 / ETH2_CRS / USB_OTGID / CANTX1 / SPI_MISO3	—	IRQ4	—	—	ENCIFCK0
D3	VCC1833_2	—	P01_0	CAS#	MTIOC3C / MTIOC4C / GTIOC0B / GTIOC3A	CTS2# / GMAC_MDIO / ESC_MDIO	MCLK1	IRQ13	—	—	—
D4	VCC1833_2	—	P00_6	CS5#	MTIOC3A / MTIOC3B / GTIOC0A / GTIOC1A	ETH2_TXCLK	MDAT0	—	—	—	—
D5	VCC1833_2	—	P24_0	D12 / CKE / DREQ	MTIOC0B / GTETRGB	RXD1 / SCL1 / MISO1 / ETH2_RXD1	MDAT4	—	—	—	—
D6	—	TRACECTL	P22_1	D8	POE4# / GTETRGB	SS4# / CTS4# / RTS4# / ESC_LINKACT2 / SPI_MOSI0	—	IRQ13	—	HSPI_IO7	—
D7	—	TRACEDATA3	P21_4	D3	MTIOC6D / GTIOC15B	SS5# / CTS5# / RTS5# / GMAC_PTPOUT1 / ESC_SYNC0 / ESC_SYNC1 / SPI_SSL02	MDAT1 / MBX_HINT#	—	—	—	ENCIFDO0
D8	—	VCC33	—	—	—	—	—	—	—	—	—

Table 1.18 List of pins and pin functions (196-pin FBGA) (4 of 8)

Pin number	I/O power domain	Power supply clock system control	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI)	DSMIF, Mailbox	Interrupt	ADC12	Host interface (SHOST)	Encoder I/F
D9	—	VREFH1	—	—	—	—	—	—	—	—	—
D10	—	VREFH0	—	—	—	—	—	—	—	—	—
D11	—	VSS	—	—	—	—	—	—	—	—	—
D12	—	—	P18_2	BS	MTIOC4B / MTIOC4D / GTIOC2B / GTIOC3B	SCK0 / IIC_SDA2 / GMAC_PTPOUT3 / XSPI1_CS0#	—	—	—	—	—
D13	—	—	P18_3	CKE	MTIOC4B / MTIOC4D / GTIOC2B / GTIOC3B / CMTW1_TIC1	DE3 / ETH2_CRS / CANRXDP1 / XSPI1_IO4	—	IRQ0	—	—	—
D14	—	—	P18_1	WE1# / DQMLU	MTIOC3D / GTIOC1B	SS3# / CTS3# / RTS3#	—	IRQ10	ADTRG1#	—	—
E1	—	TDI	P02_5	WE1# / DQMLU	—	SCK5 / SPI_SSL31	—	—	—	—	—
E2	—	—	P02_2	A16	MTIOC2A / POE10# / GTIOC10A / RTCAT1HZ	TXD1 / SDA1 / MOSI1 / CANTX0 / SPI_MOSI3	—	IRQ14	—	—	ENCIFDI0
E3	—	TDO	P02_4	WE0# / DQMLL	—	DE1 / SPI_SSL33	—	—	—	—	—
E4	VCC1833_2	—	P01_4	WE1# / DQMLU	POE0#	ETH2_TXD1	—	IRQ3	—	—	DUEI0 / HDSL0_LINK
E5	VCC1833_2	—	P00_0	D15	—	SCK2 / DE2 / ETH2_RXD3	—	—	—	—	DUEI1 / HDSL0_SEL1
E6	—	VCC1833_2	—	—	—	—	—	—	—	—	—
E7	—	VDD	—	—	—	—	—	—	—	—	—
E8	—	VDD	—	—	—	—	—	—	—	—	—
E9	—	VCC18_ADC1	—	—	—	—	—	—	—	—	—
E10	—	VCC18_ADC0	—	—	—	—	—	—	—	—	—
E11	—	VSS	—	—	—	—	—	—	—	—	—
E12	—	TRACECLK	P17_4	DACK	MTIOC3C / GTIOC0A / GTETRGB	CTS3# / ETH2_TXER / ESC_PHYLINK2 / SPI_SSL32 / SPI_RSPCK0 / XSPI1_IO3	—	—	—	—	HDSL1_CLK2
E13	—	—	P17_7	RD# / DACK	MTIOC4A / MTIOC4C / GTIOC2A / GTIOC3A	RXD3 / SCL3 / MISO3 / XSPI1_CKP	—	—	—	—	—
E14	—	—	P18_0	WE0# / DQMLL	MTIOC4A / MTIOC4C / GTIOC2A / GTIOC3A	TXD3 / SDA3 / MOSI3 / XSPI1_IO0	—	—	—	—	—
F1	—	—	P04_1	CKIO	—	TXD3 / SDA3 / MOSI3 / IIC_SDA2 / SPI_MOSI0	—	—	—	—	—
F2	—	TCK	P02_7	—	—	TXD5 / SDA5 / MOSI	—	—	—	—	—
F3	—	TMS	P02_6	—	—	RXD5 / SCL5 / MISO5	—	—	—	—	—
F4	—	TRST#	—	—	—	—	—	—	—	—	—
F5	—	VCC33	—	—	—	—	—	—	—	—	—
F6	—	VDD	—	—	—	—	—	—	—	—	—
F7	—	VSS	—	—	—	—	—	—	—	—	—
F8	—	VSS	—	—	—	—	—	—	—	—	—
F9	—	VDD	—	—	—	—	—	—	—	—	—
F10	—	VCC33	—	—	—	—	—	—	—	—	—
F11	—	MDD	P17_0	—	—	SS0# / CTS0# / RTS0# / ESC_IRQ / SPI_SSL01 / XSPI1_IO1	—	—	—	—	HDSL1_MISO2

Table 1.18 List of pins and pin functions (196-pin FBGA) (5 of 8)

Pin number	I/O power domain	Power supply clock system control	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI)	DSMIF, Mailbox	Interrupt	ADC12	Host interface (SHOST)	Encoder I/F
F12	—	—	P17_6	RD/WR#	MTIOC3B / GTIOC1A	SCK3 / XSPI1_DS	—	—	—	—	—
F13	—	TRACECTL	P17_3	DREQ	POE0# / GTETRGA	CANRX1 / SPI_SSL31 / XSPI1_IO2	—	—	ADTRG1#	—	—
F14	—	RSTOUT#	P17_5	TEND	MTIOC3A / GTIOC0B / GTETRGC	USB_OVRCUR	—	—	—	—	—
G1	—	—	P06_0	A12	GTIOC16A / CMTW1_TOC0	SS4# / CTS4# / RTS4# / ETH1_TXD3 / CANRX1 / SPI_SSL23	—	—	—	—	HDSL0_MOSI1
G2	—	—	P05_6	A13	GTIOC14A / CMTW1_TIC0	ETH1_RXER / SPI_SSL22	—	IRQ12	—	—	HDSL1_MOSI2
G3	—	—	P05_5	A14	GTIOC14B / CMTW0_TOC1	ESC_PHYLINK1 / SPI_RSPCK2	—	—	—	—	HDSL1_LINK
G4	—	BSCANP	—	—	—	—	—	—	—	—	—
G5	—	VDD	—	—	—	—	—	—	—	—	—
G6	—	VSS	—	—	—	—	—	—	—	—	—
G7	—	VSS	—	—	—	—	—	—	—	—	—
G8	—	VSS	—	—	—	—	—	—	—	—	—
G9	—	VSS	—	—	—	—	—	—	—	—	—
G10	—	VDD	—	—	—	—	—	—	—	—	—
G11	VCC1833_3	—	P16_1	CS2#	CMTW0_TOC1	RXD0 / SCL0 / MISO0 / SPI_MISO3 / XSPI0_RESET0#	MDAT3	—	ADTRG0#	—	ENCIFOE0 / HDSL1_SEL1
G12	VCC1833_3	—	P16_0	—	—	TXD0 / SDA0 / MOSI0 / ETH2_REFCLK / SPI_MOSI3 / XSPI0_CS1#	MCLK3	—	—	—	ENCIFCK0 / HDSL0_MOSI2
G13	VCC1833_3	—	P16_2	—	—	CTS0# / USB_EXICEN / SPI_RSPCK3 / SPI_SSL03 / XSPI0_RESET1#	—	NMI	—	—	ENCIFCK1 / HDSL1_MISO1
G14	VCC1833_3	—	P16_3	CS3#	GTADSMP1	SCK0 / ETH2_RXER / SPI_SSL30 / XSPI0_RSTO0#		IRQ7	—	—	ENCIFOE1 / HDSL1_MOSI1
H1	—	—	P06_4	A7	GTIOC11A	ETH1_TXCLK / SPI_MOSI1	—	—	—	—	HDSL0_SEL2
H2	—	MD1	P06_2	A9	—	ETH1_TXD1 / CANRXDP1	—	—	—	—	—
H3	—	—	P06_1	A10	GTIOC16B	CTS4# / ETH1_REFCLK / RMII1_REFCLK / CANTX1 / SPI_SSL22	—	—	—	—	HDSL0_CLK2
H4	—	MD2	P05_7	A11	CMTW1_TOC1	ETH1_TXD2 / SPI_SSL23	—	—	—	—	—
H5	—	VDD	—	—	—	—	—	—	—	—	—
H6	—	VSS	—	—	—	—	—	—	—	—	—
H7	—	VSS	—	—	—	—	—	—	—	—	—
H8	—	VSS	—	—	—	—	—	—	—	—	—
H9	—	VSS	—	—	—	—	—	—	—	—	—
H10	—	VDD	—	—	—	—	—	—	—	—	—
H11	VCC1833_3	—	P15_6	D14	—	SPI_SSL12 / XSPI0_IO7	MDAT2	—	—	—	—
H12	VCC1833_3	—	P15_5	D13	—	XSPI0_IO6	MCLK2	IRQ7	—	—	—
H13	VCC1833_3	—	P15_4	D12	MTIOC8D	XSPI0_IO5	MDAT1	IRQ3	—	—	—
H14	VCC1833_3	—	P15_7	TEND	—	CTS5# / SPI_SSL13 / XSPI0_CS0#	—	—	—	—	—

Table 1.18 List of pins and pin functions (196-pin FBGA) (6 of 8)

Pin number	I/O power domain	Power supply clock system control	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI)	DSMIF, Mailbox	Interrupt	ADC12	Host interface (SHOST)	Encoder I/F
J1	—	—	P06_7	A4	GTIOC12B	ETH1_RXD1 / SPI_SSL11	—	—	—	—	DUEI1 / HDSL1_LINK
J2	—	—	P06_5	A6	GTIOC11B	ETH1_TXEN / SPI_MISO1	—	—	—	—	HDSL0_MISO2
J3	—	—	P06_6	A5	GTIOC12A	ETH1_RXD0 / SPI_SSL10	—	—	—	—	HDSL1_SMPL
J4	—	MD0	P06_3	A8	—	DE4 / ETH1_TXD0 / CANTXDP1	—	—	—	—	—
J5	—	VCC33	—	—	—	—	—	—	—	—	—
J6	—	VDD	—	—	—	—	—	—	—	—	—
J7	—	VSS	—	—	—	—	—	—	—	—	—
J8	—	VSS	—	—	—	—	—	—	—	—	—
J9	—	VDD	—	—	—	—	—	—	—	—	—
J10	—	VCC1833_3	—	—	—	—	—	—	—	—	—
J11	VCC1833_3	—	P15_3	D11	MTIOC8C	XSPI0_IO4	MCLK1	NMI	—	—	—
J12	VCC1833_3	—	P15_0	A23 / CKE	—	RXD5 / SCL5 / MISO5 / SPI_MOSI1 / XSPI0_IO1	—	—	—	—	—
J13	VCC1833_3	—	P15_1	A24 / CAS#	MTIOC0C	TXD5 / SDA5 / MOSI5 / SPI_SSL10 / XSPI0_IO2	—	—	—	—	—
J14	VCC1833_3	—	P15_2	A25 / RAS#	MTIOC0D	SS5# / CTS5# / RTS5# / SPI_SSL11 / XSPI0_IO3	—	—	—	—	—
K1	—	—	P07_0	A3	GTIOC13A	ETH1_RXD2	—	—	—	—	TST_OUT1 / HDSL1_SMPL
K2	—	—	P07_2	A1	GTIOC17A	ETH1_RXDV	—	—	—	—	HDSL1_SEL1
K3	—	—	P07_3	A0	GTIOC17B	ETH1_RXCLK / SPI_SSL00	—	—	—	—	HDSL1_MISO1
K4	—	—	P07_1	A2	GTIOC13B	ETH1_RXD3	—	—	—	—	SI1# / HDSL1_CLK1
K5	—	—	P09_0	CS0#	MTIOC4A / MTIOC7A / GTIOC6A	RXD3 / SCL3 / MISO3 / GMAC_MDIO / ESC_MDIO	—	—	—	—	—
K6	—	VDD	—	—	—	—	—	—	—	—	—
K7	—	VDD	—	—	—	—	—	—	—	—	—
K8	—	VDD	—	—	—	—	—	—	—	—	—
K9	—	VCC33	—	—	—	—	—	—	—	—	—
K10	—	VCC1833_3	—	—	—	—	—	—	—	—	—
K11	VCC1833_3	—	P14_2	—	MTIOC8B / GTIOC8B	ETH2_CRS / SPI_SSL10 / XSPI0_ECS0#	—	IRQ6	—	—	POUTA / HDSL0_CLK2
K12	VCC1833_3	—	P14_7	A22 / BS	—	SCK5 / SPI_MISO1 / XSPI0_IO0	—	—	—	—	—
K13	VCC1833_3	—	P14_4	BS	MTIOC0B	ESC_IRQ / SPI_SSL13 / XSPI0_DS	—	—	—	—	POUTZ / HDSL0_MISO2
K14	VCC1833_3	—	P14_6	A21	—	XSPI0_CKP	—	—	—	—	—
L1	—	VSS	—	—	—	—	—	—	—	—	—
L2	—	—	P08_5	—	MTIOC3C / MTIOC6B / GTIOC5A	RXD2 / SCL2 / MISO2 / ETH0_RXDV	MCLK2	IRQ5	—	—	—
L3	—	—	P08_4	CS5#	MTIOC3A / MTIOC6A / GTIOC4A	SCK2 / ETH0_RXD3 / CANTXDP1 / SPI_SSL32	—	IRQ14	—	—	HDSL1_MOSI1
L4	—	—	P09_4	—	GTADSMP0	TXD4 / SDA4 / MOSI4 / ETH0_TXD2 / SPI_SSL21	—	—	—	—	HDSL1_MISO2

Table 1.18 List of pins and pin functions (196-pin FBGA) (7 of 8)

Pin number	I/O power domain	Power supply clock system control	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI)	DSMIF, Mailbox	Interrupt	ADC12	Host interface (SHOST)	Encoder I/F
L5	—	—	P10_1	—	POE10#	CTS3# / ETH0_RXD0 / SPI_RSPCK1	—	IRQ10	—	—	ENCIFDI1
L6	—	VCC33	—	—	—	—	—	—	—	—	—
L7	—	—	P07_4	—	—	USB_VBUSIN	—	IRQ1	ADTRG0#	—	HDSL1_SEL2
L8	—	VDD	—	—	—	—	—	—	—	—	—
L9	—	TRACEDATA6	P13_2	D9	MTIOC0A / GTIOC10A / POE8#	SS1# / CTS1# / RTS1# / IIC_SCL0 / GMAC_PTPOUT2 / ESC_I2CCLK / SPI_MISO0	MCLK4	IRQ5	—	—	—
L10	—	TRACEDATA7	P13_3	D8 / RD#	MTIOC0B / MTIOC0C / GTIOC10B / CMTW1_TOC0	CTS1# / IIC_SDA0 / GMAC_PTPOUT3 / ESC_I2CDATA / SPI_RSPCK0	MDAT4	—	—	—	—
L11	VCC1833_3	—	P13_7	—	MTCLKC	GMAC_PTPTRG1 / ESC_LATCH0 / ESC_LATCH1 / SPI_MOSI1 / XSPI0_ECS1#	MBX_HINT#	—	—	—	HDSL0_SEL1
L12	VCC1833_3	—	P13_5	—	MTCLKA	IIC_SCL2 / GMAC_PTPTRG0 / ESC_LATCH0 / ESC_LATCH1 / SPI_RSPCK1 / XSPI0_WP1#	—	—	—	—	—
L13	VCC1833_3	—	P14_3	—	MTIOC0A	ETH2_COL / SPI_SSL11 / XSPI0_RSTO1#	—	—	—	—	POUTB / HDSL0_SEL2
L14	VCC1833_3	—	P14_5	CS3#	POE8#	XSPI0_CKN	—	IRQ15	—	HSPI_INT#	—
M1	—	—	P08_7	WAIT#	MTIOC3D / MTIOC6D / GTIOC5B	TXD2 / SDA2 / MOSI2 / GMAC_MDC / ESC_MDC / SPI_SSL13	MDAT2	IRQ8	—	—	—
M2	—	—	P08_6	—	MTIOC3B / MTIOC6C / GTIOC4B / CMTW1_TIC1	SCK3 / ETH0_RXCLK	—	IRQ9	—	—	—
M3	—	—	P09_1	—	MTIOC4C / MTIOC7B / GTIOC7A / GTETRGSA	DE3 / ETH0_REFCLK/ RMII0_REFCLK / GMAC_PTPOUT0 / ESC_SYNC0 / ESC_SYNC1 SPI_SSL10	—	—	—	—	—
M4	—	—	P09_7	DACK	GTIOC15B	RXD4 / SCL4 / MISO4 / ETH0_TXCLK / USB_OVRCUR / CANTXDP0 / SPI_SSL00	—	IRQ12	—	—	ENCIFOE1
M5	—	—	P10_2	—	MTIC5U	TXD0 / SDA0 / MOSI0 / ETH0_RXD1	—	—	—	—	—
M6	—	VSS	—	—	—	—	—	—	—	—	—
M7	—	MDX	—	—	—	—	—	—	—	—	—
M8	—	VCC18_PLL1	—	—	—	—	—	—	—	—	—
M9	—	VSS	—	—	—	—	—	—	—	—	—
M10	—	VCC33_USB	—	—	—	—	—	—	—	—	—
M11	—	VSS_USB	—	—	—	—	—	—	—	—	—
M12	—	—	P13_4	A0 / WAIT#	MTIOC0D / GTIOC8B	ESC_RESETOUT# / SPI_SSL12	—	—	—	—	—
M13	VCC1833_3	—	P13_6	—	MTCLKB	GMAC_PTPOUT0 / ESC_SYNC0 / ESC_SYNC1 / XSPI0_WP0#	—	—	—	—	—

Table 1.18 List of pins and pin functions (196-pin FBGA) (8 of 8)

Pin number	I/O power domain	Power supply clock system control	I/O port	Bus, DMAC	Timer (MTU3, POE3, GPT / POEG, CMTW, RTC)	Communication (SCI, IIC, GMAC, ESC, USB, CANFD, SPI, xSPI)	DSMIF, Mailbox	Interrupt	ADC12	Host interface (SHOST)	Encoder I/F
M14	VCC1833_3	—	P14_1	—	MTIOC8A / GTIOC8A	GMAC_PTPTRG1 / ESC_LATCH0 / ESC_LATCH1 / SPI_MISO1 / XSPI0_INT1#	—	—	—	—	HDSL0_MOSI1
N1	—	—	P09_2	RAS# / DACK	MTIOC4B / MTIOC7C / GTIOC6B / RTCAT1HZ	ETH0_RXER	—	IRQ0	—	—	HDSL1_CLK2
N2	—	—	P09_3	CS3#	MTIOC4D / MTIOC7D / GTIOC7B / GTETRGSB / CMTW0_TOC0	SS5# / CTS5# / RTS5# / ETH0_TXD3 / USB_VBUSEN / CANTXDP0	MCLK3	IRQ12	—	—	HDSL1_SEL2
N3	—	—	P09_5	DREQ	GTADSML1 / CMTW0_TOC0	DE5 / IIC_SCL1 / ETH0_TXD1 / USB_VBUSEN / CANRX0	—	IRQ14	—	—	HDSL1_MOSI2
N4	—	—	P10_0	—	POE11# / CMTW0_TIC0	SCK4 / IIC_SDA1 / ETH0_TXEN / USB_EXICEN / CANTX0	—	IRQ15	—	—	ENCIFDO1
N5	—	RES#	—	—	—	—	—	—	—	—	—
N6	—	EXTCLKIN	—	—	—	—	—	—	—	—	—
N7	—	VSS	—	—	—	—	—	—	—	—	—
N8	—	VCC18_PLL0	—	—	—	—	—	—	—	—	—
N9	—	VSS_USB	—	—	—	—	—	—	—	—	—
N10	—	VSS_USB	—	—	—	—	—	—	—	—	—
N11	—	AVCC18_USB	—	—	—	—	—	—	—	—	—
N12	—	VSS_USB	—	—	—	—	—	—	—	—	—
N13	—	VSS_USB	—	—	—	—	—	—	—	—	—
N14	VCC1833_3	—	P14_0	—	MTCLKD	GMAC_PTPOUT1 / ESC_SYNC0 / ESC_SYNC1 / XSPI0_INT0#	—	—	—	—	HDSL0_MISO1
P1	—	VSS	—	—	—	—	—	—	—	—	—
P2	—	—	P09_6	—	GTIOC15A / CMTW0_TIC1	CTS5# / ETH0_TXD0 / USB_EXICEN / CANRXDP0	MDAT3	IRQ13	—	—	ENCIFCK1
P3	—	—	P10_3	—	MTIC5V / RTCAT1HZ	RXD0 / SCL0 / MISO0 / ETH0_RXD2	—	IRQ8	—	—	—
P4	—	—	P10_4	D15	MTIC5W	SCK0 / ESC_PHYLINK0 / SPI_SSL01	MBX_HINT#	IRQ11	—	—	—
P5	—	VSS	—	—	—	—	—	—	—	—	—
P6	—	EXTAL	—	—	—	—	—	—	—	—	—
P7	—	XTAL	—	—	—	—	—	—	—	—	—
P8	—	VSS	—	—	—	—	—	—	—	—	—
P9	—	USB_DM	—	—	—	—	—	—	—	—	—
P10	—	USB_DP	—	—	—	—	—	—	—	—	—
P11	—	VCC18_USB	—	—	—	—	—	—	—	—	—
P12	—	USB_RREF	—	—	—	—	—	—	—	—	—
P13	—	VSS_USB	—	—	—	—	—	—	—	—	—
P14	—	VSS	—	—	—	—	—	—	—	—	—

2. CPU

2.1 Overview

The LSI includes one Cortex®-R52 CPU. The revision of module is r1p3-00rel0.

Table 2.1 shows the specifications of the CPU.

Table 2.1 CPU specifications

Item		Description
Cortex-R52	Minimum instruction execution time	One clock per instruction
	Address space	4 GB
	Instruction cache size	16 KB (with ECC)
	Data cache size	16 KB (with ECC)
	Tightly coupled memory (TCM) size	ATCM: 512 KB (with ECC) BTCM: 64 KB (with ECC)
	Instruction set	Armv8-R architecture supporting Thumb®/Thumb-2
	Data arrangement	Instruction: Little-endian Data: Little-endian
	Memory protection	2-stage memory protection unit (MPU)
	FPU	• Supports addition, subtraction, multiplication, division, product-sum operation, and square-root operation in single-precision and double-precision • 64-bit single word register: 32 registers • Can also be used as sixteen 128-bit double word registers
	NEON	• The advanced SIMD supporting integer or single precision results

For details, see the following Arm documents:

- Arm® Architecture Reference Manual Armv8, for Armv8-A architecture profile
- Arm® Architecture Reference Manual Supplement Armv8, for the Armv8-R architecture profile
- Arm® Cortex®-R52 Processor Technical Reference Manual

2.2 Configuration Information

Table 2.2 lists the configuration options for the Cortex-R52 CPU of this LSI.

Table 2.2 Configuration options (1 of 2)

Feature	Setting value
Number of cores	1
Lock-step	No redundant logic included
EL1-controlled MPU	24
EL2-controlled MPU	24
AXI bus protection	No bus protection included
RAM protection	Included for all RAMs
Number of interrupts (SPIs) into the interrupt controller	448
AXIS ID bits	8
Advanced SIMD and floating-point capabilities for each core	Supported
Size of each of the three TCMs on each core	ATCM: 512 KB BTCM: 64 KB CTCM: 0 KB

Table 2.2 Configuration options (2 of 2)

Feature	Setting value
TCM wait states for each core	ATCM: 0 wait state (less than or equal to 400 MHz only) or 1 wait state[*1] BTCM: 0 wait states
Instruction cache size for each core	16 KB
Data cache size for each core	16 KB
External device interfaces to GIC	0
Flash ECC scheme	Not available
Vector table base address out of reset (after running boot ROM)	0x0000_0000

Note 1. It depends on the MDW pin setting at release of reset. If more than 400 MHz operation is required, 1 wait state is mandatory.

2.3 Restrictions on the CPU

For details on restrictions of the Cortex-R52 CPU implemented in this LSI, see the information provided on the Arm website.

2.4 Register Descriptions

2.4.1 WAIT : ATCM Wait State Register

Base address: TCMAW = 0x8128_1900

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WAIT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x[*1]

Bit	Symbol	Function	R/W
0	WAIT	ATCM Wait state 0: 0-wait state (MDW = 0) 1: 1-wait state (MDW = 1)	R
31:1	—	These bits are read as 0.	R

Note 1. It depends on the state of MDW pin latched at release of RES# pin on power on.

The WAIT is a register that indicates ATCM access wait. It is set by the state of the MDW pin latched at release of RES# pin on power on.

2.4.2 CNTCR : Global System Counter Control Register

Base address: GSC = 0xC006_0000

Offset address: 0x00

Bit position:	31	…	1	0
Bit field:	—	…	HDBG	EN
Value after reset:	0	…	0	0

Bit	Symbol	Function	R/W
0	EN	Counter Enable 0: The counter is disabled and not incrementing 1: The counter is enabled and is incrementing	R/W
1	HDBG	Halt on Debug HLTDBG is internal signal and it is driven high when CTI trigger output[4] (TSG-CNT HALTREQ) is asserted. It is driven low when CTI trigger output[5] (TSG-CNT RESTARTREQ) is asserted. 0: Do not halt on debug, HLTDBG signal into the counter has no effect 1: Halt on debug, when HLTDBG is driven high, the count value is held static	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The CNTCR register controls the global system counter (TSG-CNT) increments.

2.4.3 CNTSR : Global System Counter Status Register

Base address: GSC = 0xC006_0000

Offset address: 0x04

Bit position:	31																												1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DBGH	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0.	R
1	DBGH	Debug Halted 0: Not halt on debug 1: Halt on debug	R
31:2	—	These bits are read as 0.	R

The CNTSR register indicates the global system counter (TSG-CNT) status.

2.4.4 CNTCVL : Global System Counter Current Value Lower Register

Base address: GSC = 0xC006_0000

Offset address: 0x08

Bit position:	31																													0
Bit field:	CNTCVL_L_32[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CNTCVL_L_32[31:0]	Current value of the counter, lower 32 bits To change the current value, write the lower 32 bits of the new value to this register before writing the upper 32 bits to CNTCVU. The counter value is not changed until the CNTCVU register is written to.	R/W

The CNTCVL register reads or writes the lower 32 bits of the current global system counter (TSG-CNT) value. Before writing to this register, clear the CNTCR.EN bit.

2.4.5 CNTCVU : Global System Counter Current Value Upper Register

Base address: GSC = 0xC006_0000

Offset address: 0x0C

Bit position:	31																											0
Bit field:	CNTCVU_U_32[31:0]																											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CNTCVU_U_32[31:0]	Current value of the counter, upper 32 bits To change the current value, write the lower 32 bits of the new value to CNTCVL before writing the upper 32 bits to this register. The 64-bit value is updated with the value from both writes when this register is written to.	R/W

The CNTCVU register reads or writes the upper 32 bits of the current global system counter (TSG-CNT) value. Before writing to this register, clear the CNTCR.EN bit.

2.4.6 CNTFID0 : Global System Counter Base Frequency ID Register

Base address: GSC = 0xC006_0000

Offset address: 0x20

Bit position:	31																											0
Bit field:	FREQ[31:0]																											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FREQ[31:0]	Frequency in number of ticks per second Clock for generic timer counter is the main oscillator clock (25 MHz). Write 0x017D7840 for 25 MHz clock.	R/W

The CNTFID0 register specifies the clock frequency of the global system counter (TSG-CNT) in ticks per second.

2.5 Usage Notes

2.5.1 Global System Counter (TSG-CNT)

The global system counter (TSG-CNT) is a 64-bit counter that provides system counter value to generic timers of Cortex-R52 CPU0. Before using generic timer, the global system counter related registers must be set. Clock for global system counter is the main oscillator clock (25 MHz). Therefore, the Global System Counter Base Frequency ID register (CNTFID0) must be set to 0x017D7840.

TSG-CNT is implemented in CoreSight. For details, see section 9, Debugging Interface.

2.5.2 Caution of Interrupt Service Routine

In case of level interrupt, it is needed to clear interrupt factor in the interrupt service routine (ISR) generally. It may take some time to complete it due to write data buffering in the CPU and internal bus. If interrupt handler exits before completion of clearing interrupt factor at the peripheral, interrupt handler starts again. And if clearing interrupt factor is completed before GIC register is read in the interrupt handler, INTID of 1023 (0x3FF) may be read since no pending interrupt is available at this moment. To avoid this situation, make sure that interrupt factor is cleared definitely by reading related register before exiting ISR.

3. Operating Modes

3.1 Overview

The LSI is intended for booting up from an external flash memory. Seven boot modes are available for the types of flash memory that supports seven different operating modes. In each of the boot modes, the user program stored in the corresponding external flash memory is booted up and runs. Secure boot mode (in which a user program is protected by encryption) can also be selected for the products that support security functions.

3.2 Types and Selection of Operating Modes

One of the seven types of operating modes can be selected, depending on the method of connecting to an external flash memory and device. An operating mode is selected based on the input levels of the mode setting pins (MD2, MD1, and MD0) at the time pin reset (except software reset) is released.

Table 3.1 describes the relationship between the input levels of the mode setting pins (MD2, MD1, and MD0) at the time reset is released and the selected operating mode. Value of the pins (MD2 to MD0) are latched to the registers when reset is released. For details on individual operating modes, see section 3.5. Operating Mode Descriptions.

Table 3.1 Selection of operating mode for each combination of levels of mode setting pins (MD2, MD1, and MD0)

Mode setting pins			Operating mode
MD2	MD1	MD0	
Low	Low	Low	xSPI0 boot mode (x1 boot serial flash) Boots a program from x1 boot serial flash memory connected to the xSPI0 CS0 space. Supporting voltage (3.3 V or 1.8 V)
Low	Low	High	xSPI0 boot mode (x8 boot serial flash) Boots a program from x8 boot serial flash memory such as HyperFlash memory connected to the xSPI0 CS0 space. Supporting voltage (3.3 V or 1.8 V)
Low	High	Low	16-bit bus boot mode (NOR flash) Boots a program from a NOR flash memory (bus width: 16 bits) connected to the CS0 space.
Low	High	High	Serial host interface boot mode Boots a program downloaded from an external host CPU connected to SHOSTIF.
High	Low	Low	xSPI1 boot mode (x1 boot serial flash) Boots a program from x1 boot serial flash memory connected to the xSPI1 CS0 space.
High	Low	High	SCI (UART) boot mode Boots a program from host PC through UART communication connected to the SCI0. For flash writer use.
High	High	Low	USB boot mode Boots a program from host PC through USB. For flash writer use.
High	High	High	Reserved (setting prohibited)

Additionally, operating voltage of I/O domain 2 and 3 are selected based on the input levels of the voltage setting pins (MDV3 and MDV2). Table 3.2 shows the selection of operating voltage of I/O domain 2 and 3. Value of the pins (MDV3 and MDV2) are latched to the registers when reset is released.

Table 3.2 Selection of operating voltage of I/O domain 2 and 3 (MDV3 and MDV2)

MDVn	Operation voltage of I/O domain n
Low	1.8 V
High	3.3 V

Table 3.3 Selection of operating voltage of I/O domain 2 and 3 (MDV3 and MDV2) (1 of 2)

I/O Domain	MDVn value	Operating voltage of I/O domain
	196 FBGA	196 FBGA
ETH0 (MDV0)*1	High	3.3 V
ETH1 (MDV1)*1	High	3.3 V

Table 3.3 Selection of operating voltage of I/O domain 2 and 3 (MDV3 and MDV2) (2 of 2)

I/O Domain	MDVn value	Operating voltage of I/O domain
	196 FBGA	196 FBGA
ETH2 (MDV2)	Low/High	1.8 V or 3.3 V
XSPI0 (MDV3)	Low/High	1.8 V or 3.3 V
XSPI1 (MDV4)*1	High	3.3 V

Note 1. MDV4, MDV1, and MDV0 pins are not available. I/O voltage is fixed to 3.3 V.

Wait cycle for ATCM is selectable from 0 wait or 1 wait only during reset. So ATCM wait cycle setting pin (MDW) is used to capture the value. The value of this pin is latched to the register when reset is released as well as MDn and MDVn.

Mode setting pins and voltage setting pins can be shared with other peripheral function pins as strap option. The I/O voltage of this pin is 3.3 V. The value of this pin is latched to the register when reset is released.

Table 3.4 Selection of ATCM wait cycle setting pin (MDW)

MDW	ATCM wait cycle
0	0 wait Valid for CPU operating frequency equal to or less than 400 MHz.
1	1 wait

JTAG authentication by hash is handled by boot ROM. Boot ROM must identify whether JTAG is connected at boot sequence. JTAG authentication by hash enabling pin (MDD) is used to notify the CPU of whether JTAG is connected securely. The value of this pin is latched to the register when reset is released as well as other mode setting pins. This pin is valid for products that support security functions.

Table 3.5 Selection of JTAG authentication by Hash

MDD	JTAG mode
Low	Normal mode JTAG authentication by Hash is disabled
High	JTAG authentication by Hash mode

Mode setting pins and voltage setting pins can be shared with other peripheral function pins as strap option. The I/O voltage of these pins is 3.3 V.

3.3 Hardware Used in Individual Operating Modes

Table 3.6 describes hardware used in individual operating modes. Pins to be used indicates pins required to execute each operating mode. At boot, the initial values defined for each operating mode are automatically set to these pins by hardware.

Table 3.6 Hardware used in individual operating modes (1 of 2)

Operating mode	Peripheral module	Pins to be used
xSPI boot mode (x1 boot Serial flash)	Expanded serial peripheral interface (xSPI0)	XSPI0_CKP XSPI0_CS0# XSPI0_IO0, XSPI0_IO1
xSPI boot mode (x8 boot Serial flash)	Expanded serial peripheral interface (xSPI0)	XSPI0_CKP, XSPI0_CKN XSPI0_CS0#, XSPI0_DS XSPI0_IO0 to XSPI0_IO7 XSPI0_RESET0#
16-bit bus boot mode (NOR flash)	Bus state controller (BSC)	A20 to A1, D15 to D0 CS0#, RD#
Serial host interface boot mode	Serial host interface (SHOSTIF)	HSPI_CK, HSPI_CS#, HSPI_IO[n:0], MBX_HINT# (Number of HSPI_IO (n) is dependent on OTP setting (SHBMS).)
xSPI boot mode (x1 boot Serial flash)	Expanded serial peripheral interface (xSPI1)	XSPI1_CKP, XSPI1_CS0#, XSPI1_IO0, XSPI1_IO1

Table 3.6 Hardware used in individual operating modes (2 of 2)

Operating mode	Peripheral module	Pins to be used
SCI (UART) boot mode	Serial communication interface (SCI)	TXD0, RXD0
USB boot mode	USB 2.0 HS function module	USB_DP USB_DM

Note: For relationship between pin function and I/O port on boot mode, see section 16.6.4. Pin Assignment of each Boot Mode.

3.4 Register Descriptions

3.4.1 MD_MON : Operating Mode Monitor Register

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x1100

Bit position:	31											20	19	18	17	16		14	13	12							4				0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	MDV4MON	MDV3MON	MDV2MON	MDV1MON	MDV0MON	—	MD2MON	MD1MON	MD0MON	—	—	—	—	—	—	—	MDWMON	—	—	—	MDDMON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	*1	*1	*1	*1	*1	0	*1	*1	*1	0	0	0	0	0	0	0	*1	0	0	0	*1

Bit	Symbol	Function	R/W
0	MDDMON	MDD status flag This bit holds a value that indicates the level on the external MDD pin latched when reset is released.	R
3:1	—	These bits are read as 0.	R
4	MDWMON	MDW status flag This bit holds a value that indicates the level on the external MDW pin latched when reset is released.	R
11:5	—	These bits are read as 0.	R
12	MD0MON	MD0 pin status flag This bit holds a value that indicates the level on the external MD0 pin latched when reset is released.	R
13	MD1MON	MD1 pin status flag This bit holds a value that indicates the level on the external MD1 pin latched when reset is released.	R
14	MD2MON	MD2 pin status flag This bit holds a value that indicates the level on the external MD2 pin latched when reset is released.	R
15	—	This bit is read as 0.	R
16	MDV0MON	MDV0 status flag (ETH0 domain) This bit holds a value that indicates the level on the external MDV0 pin latched when reset is released. This bit is reserved and read as 1.	R
17	MDV1MON	MDV1 status flag (ETH1 domain) This bit holds a value that indicates the level on the external MDV1 pin latched when reset is released. This bit is reserved and read as 1.	R
18	MDV2MON	MDV2 status flag (ETH2 domain) This bit holds a value that indicates the level on the external MDV2 pin latched when reset is released.	R
19	MDV3MON	MDV3 status flag (xSPI0 domain) This bit holds a value that indicates the level on the external MDV3 pin latched when reset is released.	R

Bit	Symbol	Function	R/W
20	MDV4MON	MDV4 status flag (xSPI1 domain) This bit holds a value that indicates the level on the external MDV4 pin latched when reset is released. This bit is reserved and read as 1.	R
31:21	—	These bits are read as 0.	R

Note 1. The value depends on whether the signal is being externally pulled up or down, and the value at the time of latching on deassertion of the reset signal.

The MD_MON register is for checking the mode (externally pulled up or down) of the MDn, MDVn, and MDD pins by the CPU.

This register latches the values at the same time as deassertion of the signal on the external reset pin (RES#). Afterwards, it holds the values until the signal is asserted again. For reset sources and the timing of release from the reset state, see section 5, Reset.

3.5 Operating Mode Descriptions

3.5.1 Boot Function

After reset is released on this LSI, the boot function executes the boot processing on CPU0 as described below. The boot processing can extract a loader program that was stored in an external memory in advance by the user, to the internal tightly coupled memory (TCM) area, and hand over the processing to the loader program at the start address of that program.

(1) Setting the expanded serial peripheral interface or bus controller (xSPI or BSC) specified by the mode setting pins (MD2, MD1, and MD0)

(2) Loading parameters for the loader from an external memory, and checking CHECK_SUM value

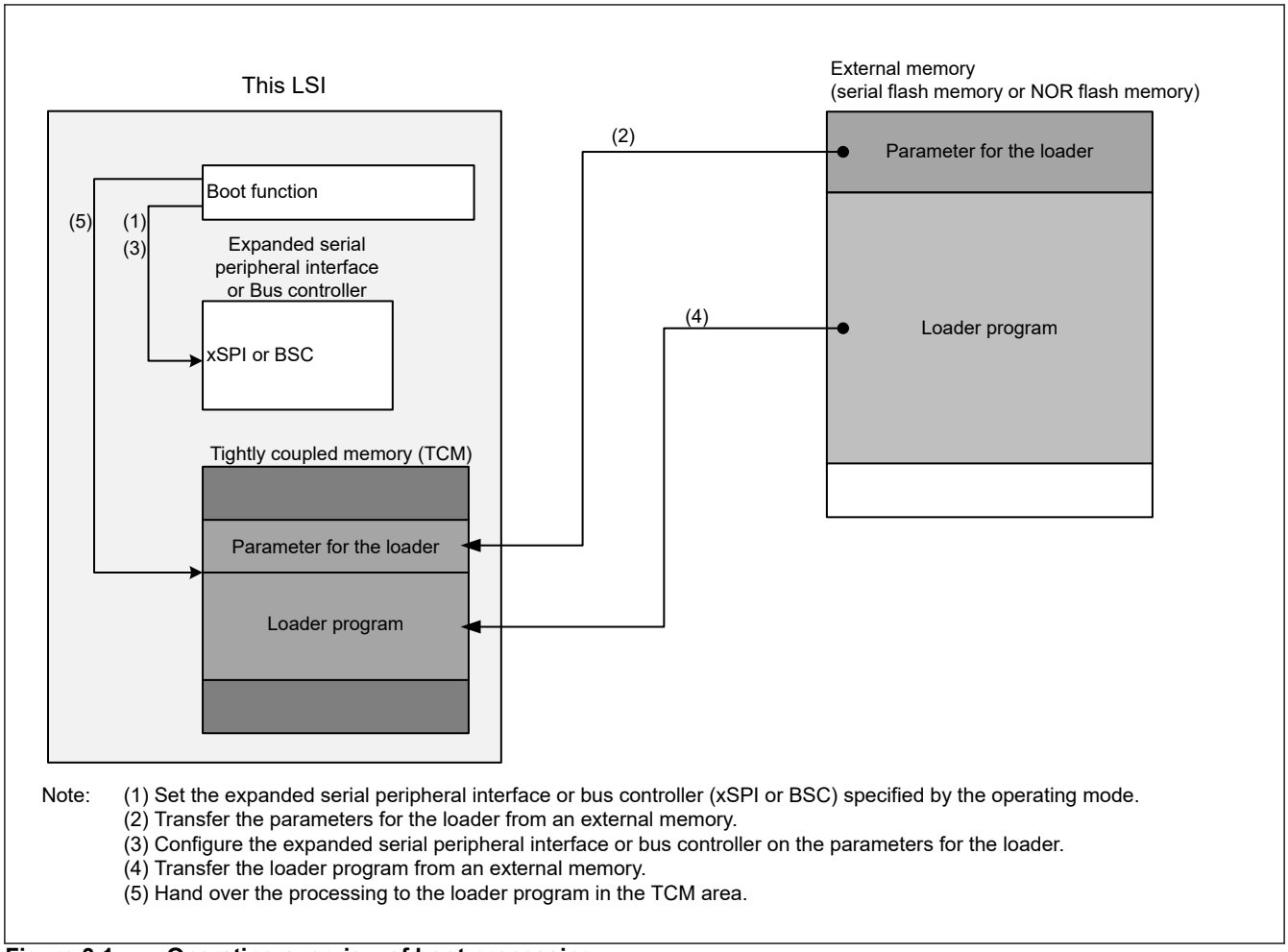
(3) Setting for speeding up the expanded serial peripheral interface or bus controller (xSPI or BSC) by using parameters for the loader

(4) Loading the loader program from an external memory

(5) Branching off to the start address of the loader program extracted to the tightly coupled memory (TCM)

Parameters for the loader can have configuration information that is suited for the user system, such as loader program information, cache setting for speeding up the boot processing, and expanded serial peripheral interface or bus controller (xSPI or BSC) settings. Parameters for the loader must be stored in an external memory in advance by the user.

Figure 3.1 shows the operating overview of boot processing.



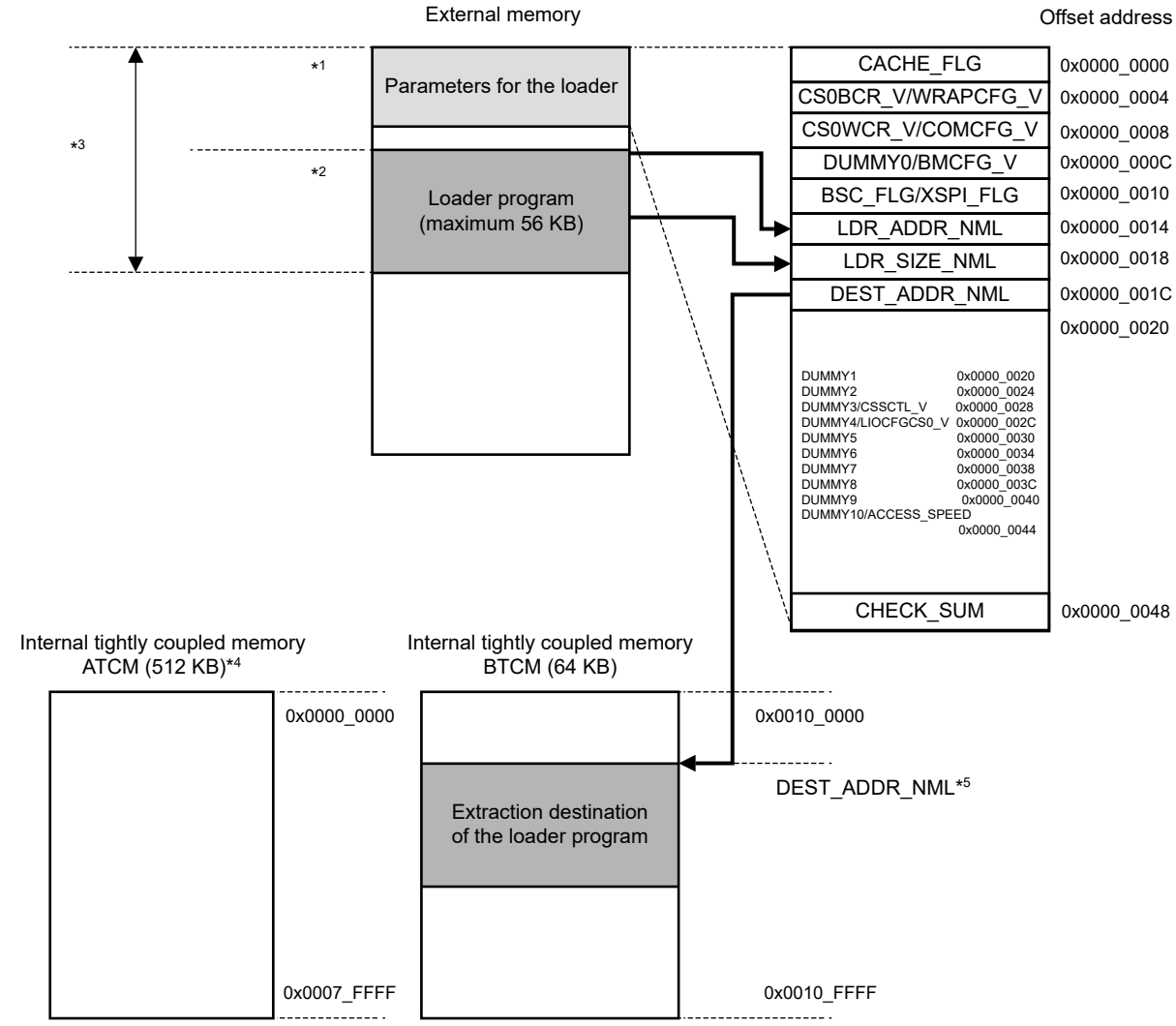
Note: (1) Set the expanded serial peripheral interface or bus controller (xSPI or BSC) specified by the operating mode.
(2) Transfer the parameters for the loader from an external memory.
(3) Configure the expanded serial peripheral interface or bus controller on the parameters for the loader.
(4) Transfer the loader program from an external memory.
(5) Hand over the processing to the loader program in the TCM area.

Figure 3.1 Operating overview of boot processing

3.5.2 Parameters for the Loader

The parameters for the loader are setting parameters for boot processing, which are loaded from an external memory during boot processing and used by the boot function. The parameters for the loader specify information, such as the cache settings during boot processing in individual operating modes, setting of the expanded serial peripheral interface or bus controller (xSPI or BSC) used for communication with an external memory, and the size of the loader program.

Figure 3.2 shows memory assignment of the loader program and parameters for the loader.



Note 1. The parameters for the loader must be stored in the following address range:
Serial flash boot mode: xSPI0 or xSPI1 space (0x6000_0000 and higher or 0x6800_0000 and higher)
Parallel flash boot mode: External address space (CS0) (0x7000_0000 and higher)
Note 2. The loader program can be stored in any address (LDR_ADDR_NML). However, the program size (LDR_SIZE_NML) must be multiples of 512 bytes but no more than 56 KB.
Note 3. In 16-bit bus boot mode, use addresses in the range from 0x7000_0000 to 0x701F_FFFF because the address output pins (A1 to A20) of the bus state controller (BSC) is used in this mode.
Note 4. The ATCM area is assumed to be used for storing user application programs.
Note 5. Specify the extraction destination area of the loader program so that the area falls in the range from 0x0010_2000 to 0x0010_FFFF.

Figure 3.2 Memory assignment of the loader program and parameters for the loader

Table 3.7 describes parameter information for the loader in 16-bit bus boot mode and Table 3.8 describes parameter information for the loader in xSPI boot mode.

Table 3.7 Parameter information for the loader in 16-bit bus boot mode (1 of 2)

Offset address	Parameter name	Description
0x0000_0000	CACHE_FLG	Selects whether to enable the I1 cache and D1 cache of Cortex-R52 during boot processing (for speeding up). 0x0000_0001: Enables the I1 and D1 caches. Setting values other than above: Disables the I1 and D1 caches.

Table 3.7 Parameter information for the loader in 16-bit bus boot mode (2 of 2)

Offset address	Parameter name	Description
0x0000_0004	CS0BCR_V	Setting value of the CS0 Space Bus Control Register (CS0BCR) This parameter value is set to the CS0BCR register during the setting for speeding up of BSC in (3) of section 3.5.1. Boot Function.*1
0x0000_0008	CS0WCR_V	Setting value of the CS0 Space Wait Control Register (CS0WCR) This parameter value is set to the CS0WCR register during the setting for speeding up of BSC in (3) of section 3.5.1. Boot Function.*1
0x0000_000C	DUMMY0	Option (not used in this mode)
0x0000_0010	BSC_FLG	Selects whether to change the BSC setting back to the initial value after the boot processing finishes. 0x2236_0679: Changes the BSC setting value back to the initial value after the boot processing finishes. Setting values other than above: Retains the BSC setting value used in the boot processing.*2
0x0000_0014	LDR_ADDR_NML	Sets the start address of the loader program stored in the external memory.*3
0x0000_0018	LDR_SIZE_NML	Specifies the size of the loader program. Note that the program size must be multiples of 512 bytes but no more than 56 KB.*3
0x0000_001C	DEST_ADDR_NML	Specifies the start address of the tightly coupled memory (BTCM) that is used as the extraction destination of the loader program. Specify the extraction destination area of the loader program so that the area falls in the range from 0x0010_2000 to 0x0010_FFFF.
0x0000_0020	DUMMY1	Option (not used in this mode)
0x0000_0024	DUMMY2	Option (not used in this mode)
0x0000_0028	DUMMY3	Option (not used in this mode)
0x0000_002C	DUMMY4	Option (not used in this mode)
0x0000_0030	DUMMY5	Option (not used in this mode)
0x0000_0034	DUMMY6	Option (not used in this mode)
0x0000_0038	DUMMY7	Option (not used in this mode)
0x0000_003C	DUMMY8	Option (not used in this mode)
0x0000_0040	DUMMY9	Option (not used in this mode)
0x0000_0044	DUMMY10	Option (not used in this mode)
0x0000_0048	CHECK_SUM	CHECK_SUM value for the parameters for the loader This parameter specifies the sum of the upper-order 16 bits and the lower-order 16 bits of the parameters (in unsigned long (32-bit) format) in the range of the offset addresses 0x0000 to 0x0044.*4

Note 1. For details on the CS0BCR and CS0WCR registers, see section 13, Bus State Controller.
Note 2. For details on the setting status of individual peripheral modules after the boot processing finishes, see section 3.5.5.1. Operation Settings in 16-Bit Bus Boot Modes.
Note 3. The loader program must be in the range from 0x7000_004C to 0x701F_FFFF in the external address space (CS0).
Note 4. An example for calculating CHECK_SUM is as follows:

If CS0BCR_V = 0x36DB_0C00, CS0WCR_V = 0x0000_0340, LDR_ADDR_NML = 0x7000_004C, LDR_SIZE_NML = 0x0000_6000, DEST_ADDR_NML = 0x0080_2000, and others = 0x0000_0000

CHECK_SUM is calculated as below (0x0000 is omitted in the formula):

CHECK_SUM = 0x36DB + 0x0C00 + 0x0340 + 0x7000 + 0x004C + 0x6000 + 0x0080 + 0x2000 = 0x0001_36E7

Table 3.8 Parameter information for the loader in xSPI boot mode (1 of 3)

Offset address	Parameter name	Description
0x0000_0000	CACHE_FLG	Selects whether to enable the I1 cache and D1 cache of Cortex-R52 at boot processing (for speeding up). 0x0000_0001: Enables the I1 and D1 caches. Setting values other than above: Disables the I1 and D1 caches.

Table 3.8 Parameter information for the loader in xSPI boot mode (2 of 3)

Offset address	Parameter name	Description
0x0000_0004	WRAPCFG_V	Setting value of the xSPI Wrapper Configuration register (WRAPCFG). This parameter value is set to the DSSFTCS0[4:0] bits in the WRAPCFG register during the setting for speeding up of xSPI in (3) of section 3.5.1. Boot Function *1. The bit 13 of WRAPCFG_V is used as the flag. In case of WRAPCFG_V[13] = 0: This parameter value is ignored and kept the reset value. In case of WRAPCFG_V[13] = 1: This parameter value is valid and set to the WRAPCFG register. The DSSFTCS1[4:0] bits in the WRAPCFG register remain the reset value.
0x0000_0008	COMCFG_V	Setting value of the xSPI Common Configuration register (COMCFG). This parameter value is set to the COMCFG register during the setting for speeding up of xSPI in (3) of section 3.5.1. Boot Function *1.
0x0000_000C	BMCFG_V	Setting value of the xSPI Bridge Map Configuration register (BMCFG). This parameter value is set to the PREEN bit in the BMCFG register during the setting for speeding up of xSPI in (3) of section 3.5.1. Boot Function *1. The other bits in the BMCFG register remain the reset value.
0x0000_0010	xSPI_FLG	Selects whether to change the xSPI setting back to the initial value after the boot processing finishes. 0x2236_0679: Changes the xSPI setting value back to the initial value after the boot processing finishes. Setting values other than above: Retains the xSPI setting value used during boot processing.*2
0x0000_0014	LDR_ADDR_NML	Sets the start address of the loader program stored in the external memory.
0x0000_0018	LDR_SIZE_NML	Specifies the size of the loader program. Note that the program size must be multiples of 512 bytes but no more than 56 KB*3.
0x0000_001C	DEST_ADDR_NML	Specifies the start address of the tightly coupled memory (BTCM) that is used as the extraction destination of the loader program. Specify the extraction destination area of the loader program so that the area falls in the range from 0x0010_2000 to 0x0010_FFFF.
0x0000_0020	DUMMY1	Option (not used in this mode)
0x0000_0024	DUMMY2	Option (not used in this mode)
0x0000_0028	CSSCTL_V	Setting value of the xSPI CS Size Control register (CSSCTL). This parameter value is set to the CS0SIZE[5:0] bits in the CSSCTL register during the setting for speeding up of xSPI in (3) of section 3.5.1. Boot Function*1. The CS1SIZE[5:0] bits in the CSSCTL register remain the reset value.
0x0000_002C	LIOCFGCS0_V	Setting value of the xSPI link I/O Configuration Register CS0 (LIOCFGCS0). This parameter value is set to the LIOCFGCS0 register during the setting for speeding up of xSPI in (3) of section 3.5.1. Boot Function*1. This parameter is effective on only bit 31 to bit 16 of LIOCFGCS0 register. Bit 11 to bit 0 are fixed each boot mode (x1 or x8). The bit 14 of LIOCFGCS0_V is used as the flag. In case of LIOCFGCS0_V[14] = 0: This parameter value is ignored and kept the reset value. In case of LIOCFGCS0_V[14] = 1: This parameter value is valid and set to the LIOCFGCS0 register.
0x0000_0030	DUMMY5	Option (not used in this mode)
0x0000_0034	DUMMY6	Option (not used in this mode)
0x0000_0038	DUMMY7	Option (not used in this mode)
0x0000_003C	DUMMY8	Option (not used in this mode)
0x0000_0040	DUMMY9	Option (not used in this mode)

Table 3.8 Parameter information for the loader in xSPI boot mode (3 of 3)

Offset address	Parameter name	Description
0x0000_0044	ACCESS_SPEED	This parameter is used to adjust access speed for the external memory. For xSPI boot mode using xSPI0, this value is set to FSELXSPI0[2:0] bits and DIVSELXSPI0 bit of SCKCR register. For example, Bit 6: DIVSELXSPI0 = 0 The base clock to generate serial clock is 800 MHz. Bits [2:0]: FSELXSPI0[2:0] = 100b. The clock frequency is 50.0 MHz. Other bit: Is not used to set SCKCR register. For xSPI boot mode using xSPI1, set bit 6 and bits [2:0] of this parameter as well as xSPI0 to set FSELXSPI1[2:0] bits and DIVSELXSPI1 bit of SCKCR register. Bit 14 and bits [10:8] of this parameter are ignored.
0x0000_0048	CHECK_SUM	CHECK_SUM value of the parameters for the loader This parameter specifies the sum of the upper-order 16 bits and the lower-order 16 bits of the parameters (in unsigned long (32-bit) format) in the range of the offset addresses 0x0000 to 0x0044.

Note 1. For details on the WRAPCFG, COMCFG, MBCFG, CMCFG0CS0, CMCFG1CS0, and LIOCFGCSn registers, see section 35, Expanded Serial Peripheral Interface (xSPI) (n = 0, 1).
Note 2. For details about the setting status of the individual peripheral modules after the boot processing finishes, see section 3.5.7.1. Operation Setting in SCI Boot Mode.
Note 3. The loader program must be in the range from 0x6000_004C to 0x60FF_FE00 (xSPI0) or 0x6800_004C to 0x68FF_FE00 (xSPI1) in the external address space.

3.5.3 Loader Program

The loader program is a user program that is transferred from an external memory to the internal tightly coupled memory (TCM) by the boot function and starts its processing after the boot processing finishes. The loader program can execute such processing that suits the user system, for example, extracting a user application program from an external memory to the internal TCM area and executing it at high speed.

Set the loader program so that the following conditions are satisfied:

Program size (LDR_SIZE_NML)	Multiples of 512 bytes but no more than 56 KB
Storage address in the external memory (LDR_ADDR_NMI) in 16-bit boot mode	Address range from A1 to A20 for 16-bit Bus boot mode
Storage address in the external memory (LDR_ADDR_NML) in xSPI0 boot mode	Address range from 0x6000_004C to 0x60FF_FE00
Storage address in the external memory (LDR_ADDR_NML) in xSPI1 boot mode	Address range from 0x6800_004C to 0x68FF_FE00

These setting values must be stored in an external memory as parameters for the loader. For details, see section 3.5.2. Parameters for the Loader.

3.5.4 xSPI Boot Mode (Serial Flash)

3.5.4.1 x1 Boot Serial Flash

In xSPI boot mode with x1 boot serial flash, this LSI boots a program from an external serial flash memory connected to the expanded Serial Peripheral Interface (xSPI0 or xSPI1) space CS0 with single, dual, quad, or octal data signals. In this mode, used protocol is 1S-1S-1S regardless of number of data signals of the flash. The xSPI controller is set to the mode of reading the external address space, and the XSPI0_CKP, XSPI0_CS0#, and XSPI0_IO[1:0] pins are enabled for MD[2:0] = 000b. XSPI1_CKP, XSPI1_CS0#, and XSPI1_IO[1:0] pins are enabled for MD[2:0] = 111b.

After the reset is released, the LSI executes the boot processing on CPU0. The program stored in a serial flash memory connected to the expanded SPI space is extracted to the internal tightly coupled memory (TCM), and then the processing is executed.

Figure 3.3 shows the connection diagram of this LSI with a serial flash memory.

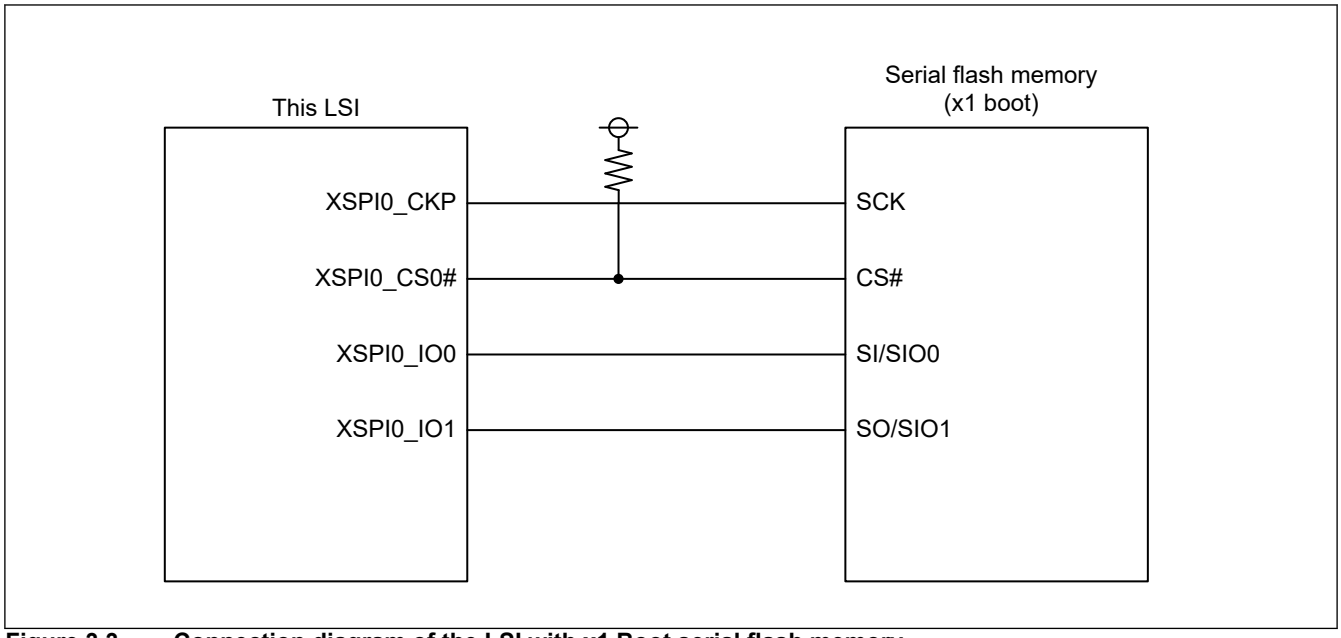


Figure 3.3 Connection diagram of the LSI with x1 Boot serial flash memory

3.5.4.2 x8 Boot Serial Flash

In xSPI boot mode with x8 boot serial flash, this LSI boots a program from an external x8 serial flash memory such as HyperFlash and xSPI 8D-8D-8D profile 2.0 compliant device connected to the expanded Serial Peripheral Interface (xSPI0) space CS0 with octal data signals. In this mode, the xSPI0 controller is set to the mode of reading the external address space, and the XSPI0_CKP, XSPI0_CKN, XSPI0_CS0#, XSPI0_IO[7:0], XSPI0_DS, and XSPI0_RESET0# pins are enabled.

After the reset is released, the LSI executes the boot processing on CPU0. The program stored in a serial flash memory connected to the expanded SPI space is extracted to the internal tightly coupled memory (TCM), and then the processing is executed.

Figure 3.4 shows the connection diagram of this LSI with a serial flash memory.

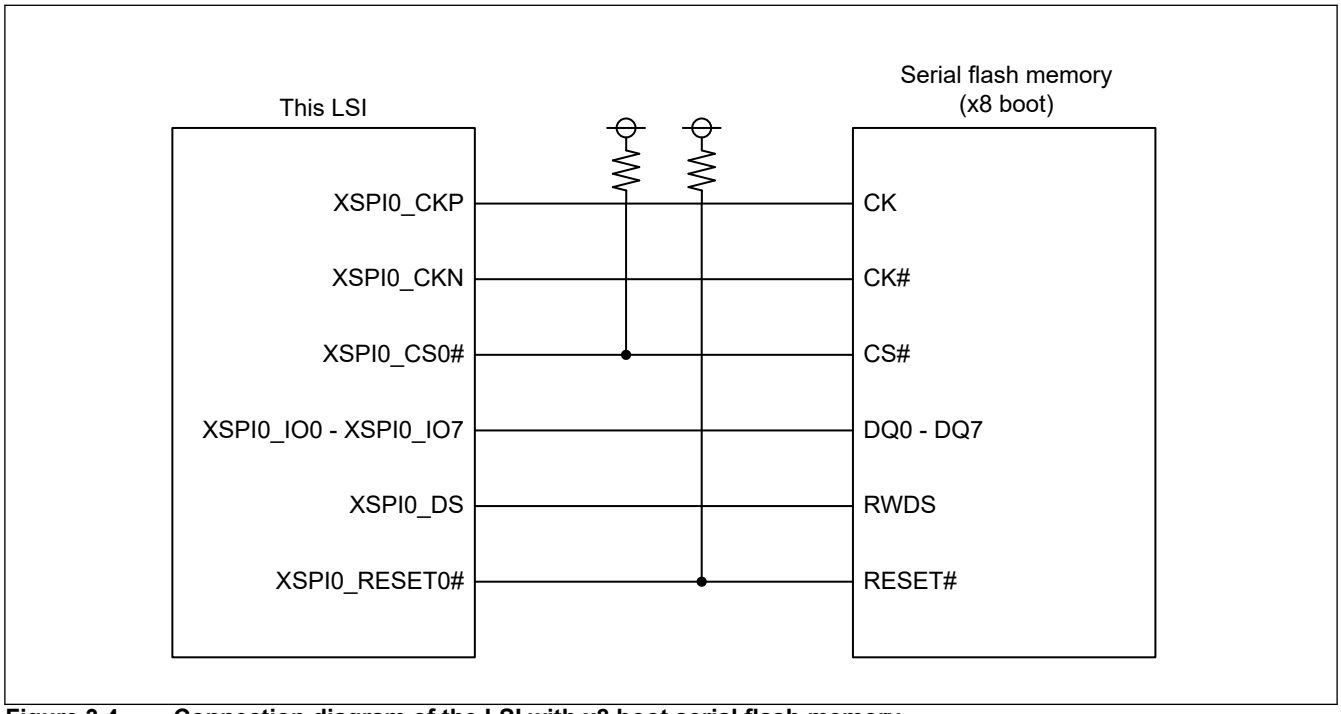


Figure 3.4 Connection diagram of the LSI with x8 boot serial flash memory

3.5.4.3 Operation Settings in xSPI Boot Mode

Immediately after the boot processing starts in xSPI boot mode after the reset is released, the LSI operates with the following initial setting values and executes processing until it transfers parameters for the loader.

Items	x1 boot	x8 boot
CPU clock (CPU0CLK)	200 MHz	
xSPI bit rate (XSPI_CLKn)	12.5 MHz*1	
Supported command	Read (0x03)	Read (CA[47:45] = 101b)
Command output	1 byte	6 bytes
Address output	3 bytes	
Dummy cycle (Read latency cycle)	0	10 Latency cycle increments from CA[23:16]
Initial data width	1 bit	8 bits
Protocol	1S-1S-1S	8D-8D-8D profile 2.0
Driving edge	Falling edge of XSPIn_CKP	Both edges of XSPI0_CKP
Sampling edge	Falling edge of XSPIn_CKP	Both edges of XSPI0_DS with shift*2
Flash reset	Software (0x66 + 0x99)	Hardware by XSPI0_RESET0#

Note 1. Serial clock frequency is configurable in parameter.
Note 2. DS shift size is configurable in parameter.

Table 3.9 describes the setting values of the related peripheral modules and registers at the time the xSPI boot mode finishes.

Table 3.9 Setting register values of the related peripheral modules when xSPI boot mode finishes (1 of 2)

Peripheral module	Register	When xSPI is initialized (xSPI_FLG = 0x22360679)		When xSPI is not initialized (xSPI_FLG ≠ 0x22360679)	
		x1	x8	x1	x8
Low power consumption	MSTPCRA	xSPI0: 0x00001F21 xSPI1: 0x00001F11	xSPI0: 0x00001F21	xSPI0: 0x00001F21 xSPI1: 0x00001F11	xSPI0: 0x00001F21
xSPI	WRAPCFG	0x00000000 (initial value)	0x00000000 (initial value)	Setting value of WRAPCFG_V (only DSSFTCS0[4:0] bits) If flag bit is not set, 0x00000000.	Setting value of WRAPCFG_V (only DSSFTCS0[4:0] bits) If flag bit is not set, 0x00001F00.
	COMCFG	0x00000000 (initial value)	0x00000000 (initial value)	Setting value of COMCFG_V	Setting value of COMCFG_V
	BMCFG	0x00000000 (initial value)	0x00000000 (initial value)	Setting value of BMCFG_V (only PREEN bit)	Setting value of BMCFG_V (only PREEN bit)
	CMCFG0CS0	0x00000008	0x0000000E	0x00000008	0x0000000E
	CMCFG1CS0	0x00000300	0x000AA000	0x00000300	0x000AA000
	LIOCFGCS0	Setting value of LIOCFGCS0_V (bit 31 to bit 16) 0x0000 (bit 15 to bit 0) If flag bit is not set, 0x00070000.	Setting value of LIOCFGCS0_V (bit 31 to bit 16) 0x07FF (bit 15 to bit 0) If flag bit is not set, 0x000707FF.	Setting value of LIOCFGCS0_V (bit 31 to bit 16) 0x0000 (bit 15 to bit 0) If flag bit is not set, 0x00070000.	Setting value of LIOCFGCS0_V (bit 31 to bit 16) 0x07FF (bit 15 to bit 0) If flag bit is not set, 0x000707FF.
	BMCTL0	0x00000001	0x00000001	0x00000001	0x00000001
	CSSCTL	0x00000000 (initial value)	0x00000000 (initial value)	Setting value of CSSCTL_V (only CS0SIZE[5:0] bits)	Setting value of CSSCTL_V (only CS0SIZE[5:0] bits)

Table 3.9 Setting register values of the related peripheral modules when xSPI boot mode finishes (2 of 2)

Peripheral module	Register	When xSPI is initialized (xSPI_FLG = 0x22360679)		When xSPI is not initialized (xSPI_FLG ≠ 0x22360679)	
		x1	x8	x1	x8
Clock	SCKCR	0x00720606 (initial value)	0x00720606 (initial value)	Setting value of ACCESS_SPEED (only bit 6, bit 2 to bit 0 for xSPI0 and bit 14, bit 10 to bit 8 for xSPI1)	Setting value of ACCESS_SPEED (only bit 6, bit 2 to bit 0)
I/O ports*1	PMCm	xSPI0: 0xC0 (m = 14), 0x81 (m = 15) xSPI1: 0xA1 (m = 17), 0x05 (m = 18)	xSPI0: 0xF0 (m = 14), 0xFF (m = 15), 0x02 (m = 16)	xSPI0: 0xC0 (m = 14), 0x81 (m = 15) xSPI1: 0xA1 (m = 17), 0x05 (m = 18)	xSPI0: 0xF0 (m = 14), 0xFF (m = 15), 0x02 (m = 16)
	PFCm	xSPI0: 0x00000000 (m = 14, 15) xSPI1: 0x70000002 (m = 17) 0x00000508 (m = 18)	xSPI0: 0x00000000 (m = 14 to 16)	xSPI0: 0x00000000 (m = 14, 15) xSPI1: 0x70000002 (m = 17) 0x00000508 (m = 18)	xSPI0: 0x00000000 (m = 14 to 16)
	DRCTLm	xSPI0: (3.3 V) 0x32320000_00000000 (m = 14) 0x32000000_00000032 (m = 15) (1.8 V) 0x22220000_00000000 (m = 14) 0x22000000_00000022 (m = 15) xSPI1: 0x32000000_00000032 (m = 17) 0x00000000_00320032 (m = 18)	xSPI0: (3.3 V) 0x32323232_00000000 (m = 14) 0x32323232_32323232 (m = 15) 0x00000000_00000000 (m = 16) (1.8 V) 0x22222222_00000000 (m = 14) 0x22222222_22222222 (m = 15) 0x00000000_00000000 (m = 16)	xSPI0: (3.3 V) 0x32320000_00000000 (m = 14) 0x32000000_00000032 (m = 15) (1.8 V) 0x22220000_00000000 (m = 14) 0x22000000_00000022 (m = 15) xSPI1: 0x32000000_00000032 (m = 17) 0x00000000_00320032 (m = 18)	xSPI0: (3.3 V) 0x32323232_00000000 (m = 14) 0x32323232_32323232 (m = 15) 0x00000000_00000000 (m = 16) (1.8 V) 0x22222222_00000000 (m = 14) 0x22222222_22222222 (m = 15) 0x00000000_00000000 (m = 16)

Note 1. See section 16.6.4.2. Pin Assignment for xSPI Boot Mode in detail.

3.5.5 16-Bit Bus Boot Modes (NOR Flash Memory)

In 16-bit bus boot mode, this LSI boots a program from an external NOR flash memory connected to the external address space (CS0). The bus width is fixed to 16 bits.

In 16-bit Bus boot mode, the A20 to A1, D15 to D0, CS0#, and RD# of the bus state controller are enabled.

After the reset is released, the LSI executes the boot processing. The loader program stored in a NOR flash memory connected to the CS0 space is extracted to the internal tightly coupled memory (TCM), and then the processing is executed.

Figure 3.5 shows the connection diagrams of the LSI with NOR flash memory.

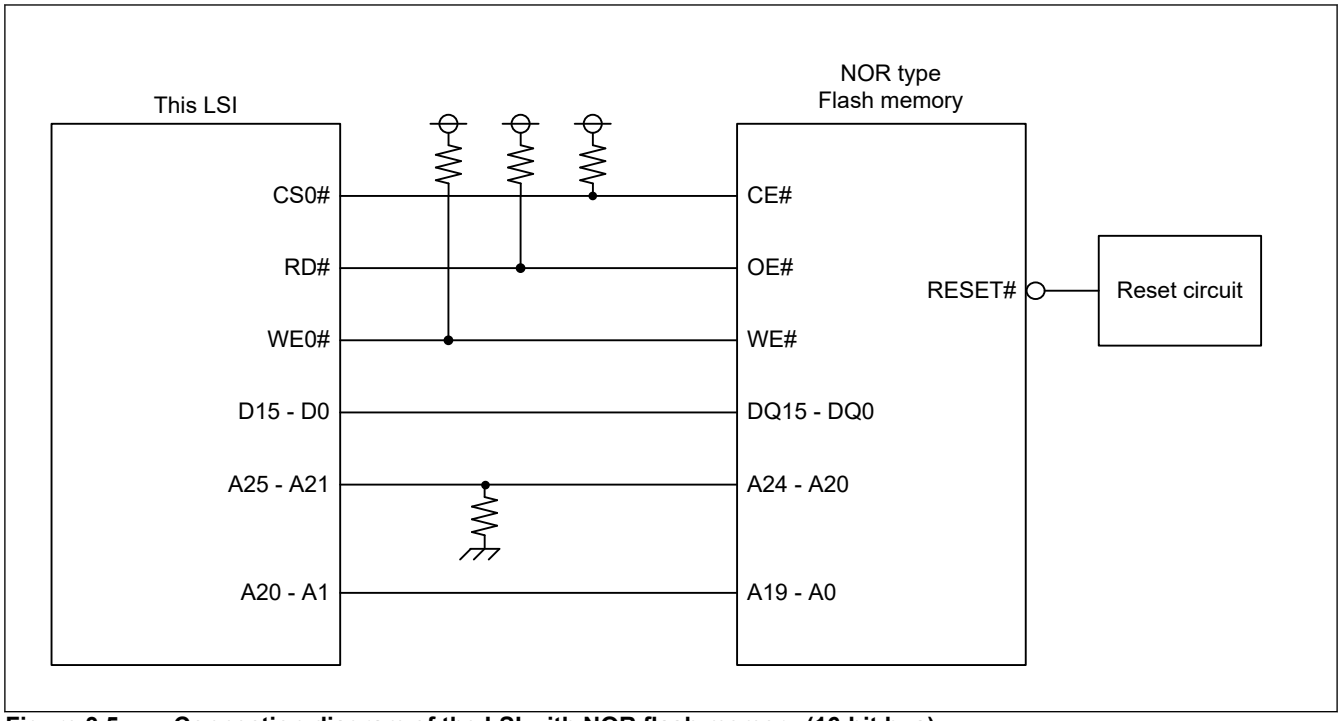


Figure 3.5 Connection diagram of the LSI with NOR flash memory (16-bit bus)

3.5.5.1 Operation Settings in 16-Bit Bus Boot Modes

After the reset is released, if the boot processing starts in 16-bit bus boot mode, the following setting values are used:

- CPU clock (CPU0CLK): 200 MHz
- External bus clock (CKIO): 50 MHz

After the parameters for the loader are loaded, the settings for the I1 and D1 caches of Cortex-R52 and for the CS0BCR and CS0WCR registers are performed based on the values of the CACHE_FLG, CS0BCR_V, and CS0WCR_V parameters so that the processing can be sped up.

Table 3.10 describes the setting values of the related peripheral modules and registers at the time when 16-bit bus boot mode finishes.

Table 3.10 Setting register values of the related peripheral modules when bus boot mode finishes

Peripheral module	Register	When BSC is initialized (BSC_FLG = 0x22360679)	When BSC is not initialized (BSC_FLG ≠ 0x22360679)
Low power consumption	MSTPCRA	0x00001F30	0x00001F30
BSC	CS0BCR	0x36DB0C00 (initial value)	Setting value of CS0BCR_V
	CS0WCR	0x00000500 (initial value)	Setting value of CS0WCR_V
I/O ports	PMCm	Only the corresponding bits are 1*1	Only the corresponding bits are 1*1
	PFCm	Only the corresponding bits are 0000b, 0001b, 0010b, 0011b, or 1000b*1	Only the corresponding bits are 0000b, 0001b, 0010b, 0011b, or 1000b*1
	DRCTLm	Only the corresponding DRV[1:0] bits are 01b if IO Type is Type A.	Only the corresponding DRV[1:0] bits are 01b if IO Type is Type A.

Note 1. The bits corresponding to A1 to A20 pins, D0 to D15 pins, RD# pin, and CS0# pin in 16-bit bus boot mode. See section 16.6.4.1. Pin Assignment for BSC Boot Mode.

3.5.6 Host Interface Boot Mode

In host interface boot mode, loader program and user program are transferred from external host CPU as master device to the RZ/T2L as slave device via host interface. Then Cortex-R52 on this LSI executes those programs.

This LSI configures the host interface based on the operation mode setting stored in OTP after releasing reset first. After receiving the reception ready mailbox interrupt from this LSI, external host CPU can transfer parameter and loader program to the System SRAM. After external host CPU informs the completion of the transfer via mailbox, Cortex-R52 transfers the loader programs from System SRAM to BTCM. Then loader program on BTCM is executed.

Loader program should inform the status of the boot and further requests to the external host CPU via mailbox and its interrupt.

3.5.6.1 Boot Sequence

Figure 3.6 is the boot sequence of host interface boot mode.

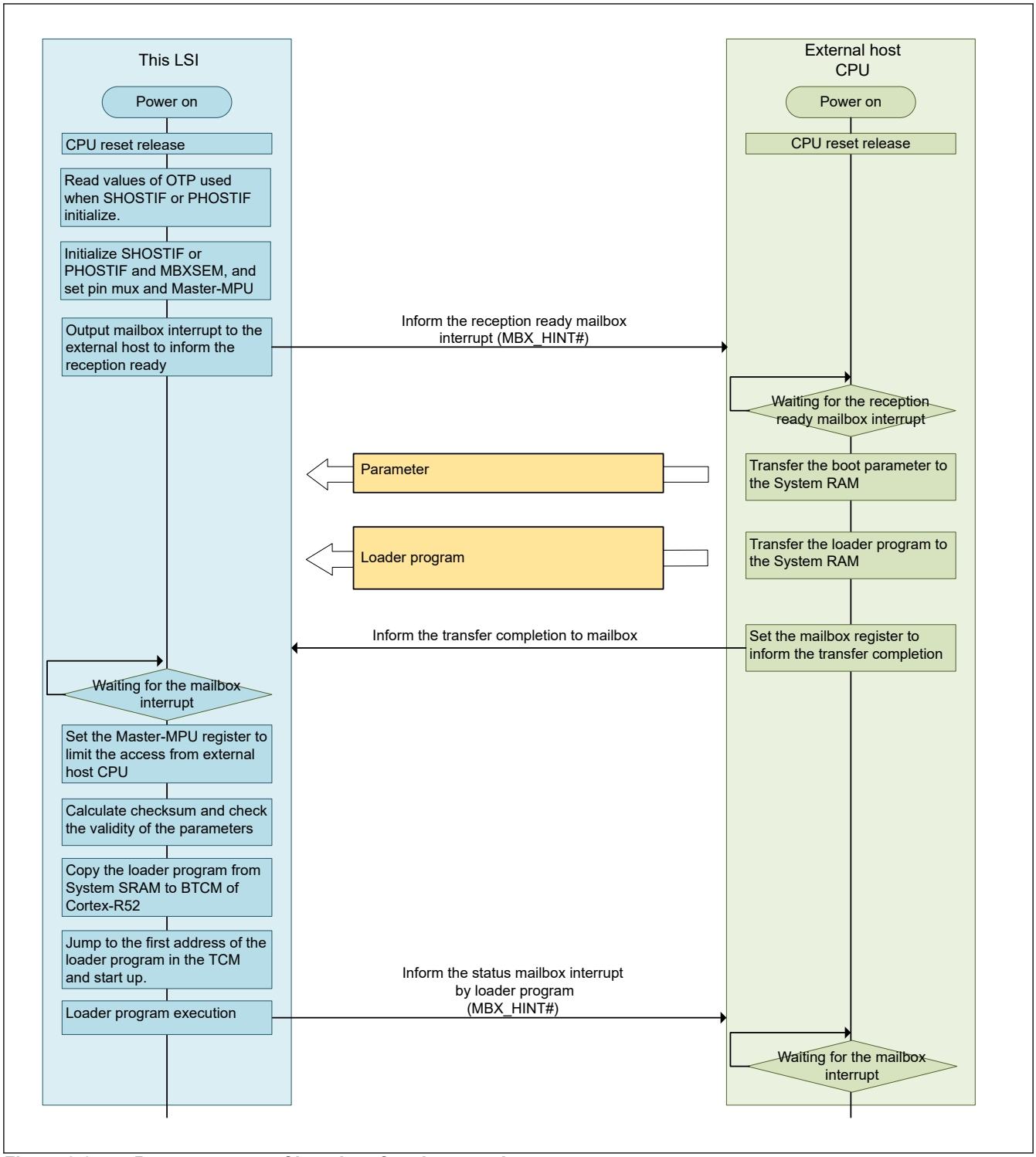


Figure 3.6 Boot sequence of host interface boot mode

3.5.6.2 Parameter and Loader Program

Parameter for the loader program is transferred from external host CPU to the System SRAM to set the size and address of the loader program. Transferred address of parameter is fixed to 0x1000_0000.

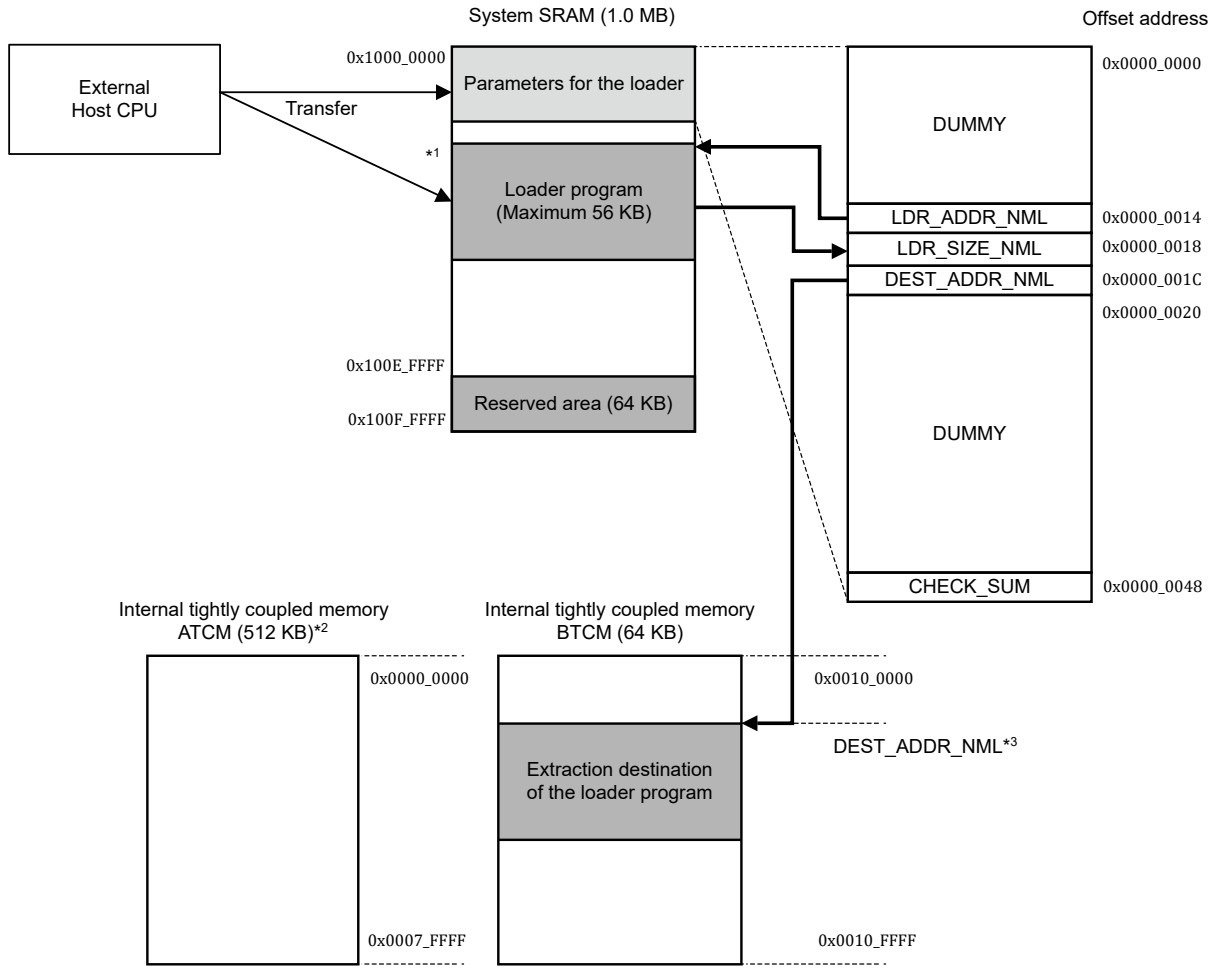
The loader program is a user program that is transferred from external host CPU to the System SRAM, and starts its processing after the boot processing finishes. Set the loader program so that the following conditions are satisfied:

- Program size (LDR_SIZE_NML)

  Multiples of 512 bytes and no more than 56 KB

- Storage address in the System SRAM (LDR_ADDR_NMI)
  Address range from 0x1000_004C to 0x100E_FFFF

Figure 3.7 shows memory assignment of the loader program and parameters for the loader.



Note 1. The loader program should be transferred in any address (LDR_ADDR_NML) in System SRAM from external host CPU. Note that, however, the program size (LDR_SIZE_NML) must be multiples of 512 bytes and no more than 56 KB.
Note 2. The ATCM area is assumed to be used for storing user application programs.
Note 3. Specify the extraction destination area of the loader program so that the area falls in the range from 0x0010_2000 to 0x0010_FFFF.

Figure 3.7 Memory assignment of the loader program and parameters for the loader

Table 3.11 describes parameter information for the loader in host interface boot mode.

Table 3.11 Parameter information for the loader in host interface boot mode (1 of 2)

Offset address	Parameter name	Description
0x0000_0000	DUMMY	Option (not used in this mode)
0x0000_0004	DUMMY	Option (not used in this mode)
0x0000_0008	DUMMY	Option (not used in this mode)
0x0000_ 000C	DUMMY	Option (not used in this mode)
0x0000_ 0010	DUMMY	Option (not used in this mode)
0x0000_ 0014	LDR_ADDR_NML	Sets the start address of the loader program stored in the System SRAM. Allowable address range is from 0x1000_004C to 0x100E_FFFF.

Table 3.11 Parameter information for the loader in host interface boot mode (2 of 2)

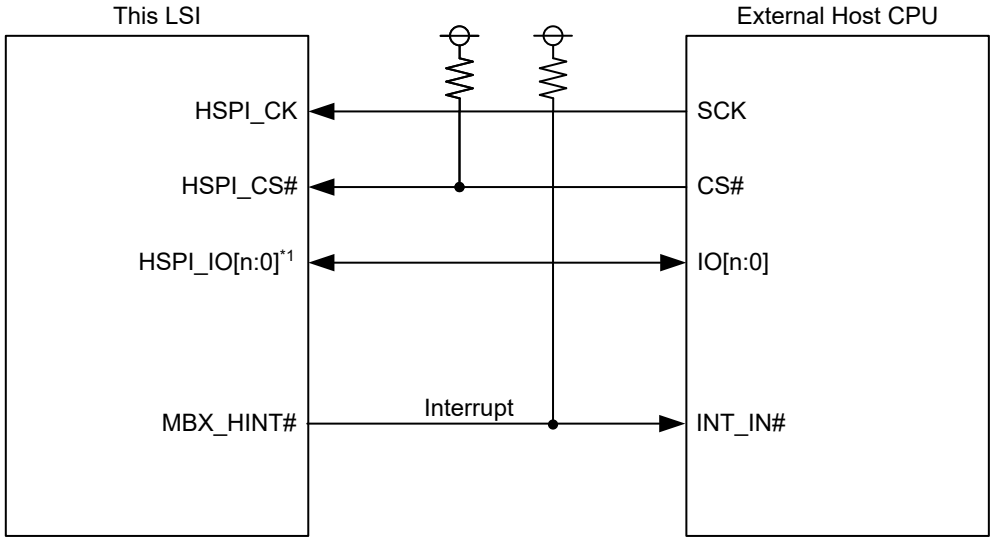
Offset address	Parameter name	Description
0x0000_ 0018	LDR_SIZE_NML	Specifies the size of the loader program. Note that the program size must be multiples of 512 bytes and no more than 56 KB.
0x0000_ 001C	DEST_ADDR_NML	Specifies the start address of the tightly coupled memory (BTCM) that is used as the extraction destination of the loader program. Specify the extraction destination area of the loader program so that the area falls in the range from 0x0010_2000 to 0x0010_FFFF.
0x0000_ 0020	DUMMY	Option (not used in this mode)
0x0000_ 0024	DUMMY	Option (not used in this mode)
0x0000_ 0028	DUMMY	Option (not used in this mode)
0x0000_ 002C	DUMMY	Option (not used in this mode)
0x0000_ 0030	DUMMY	Option (not used in this mode)
0x0000_ 0034	DUMMY	Option (not used in this mode)
0x0000_ 0038	DUMMY	Option (not used in this mode)
0x0000_ 003C	DUMMY	Option (not used in this mode)
0x0000_ 0040	DUMMY	Option (not used in this mode)
0x0000_ 0044	DUMMY	Option (not used in this mode)
0x0000_ 0048	CHECK_SUM	Checksum value for the parameters for the loader This parameter specifies the sum of the higher-order 16 bits and the lower-order 16 bits of the parameters (in unsigned long (32-bit) format) in the range of the offset addresses 0x0000 to 0x0044.*1

Note 1. An example for calculating CHECK_SUM is given below.
If LDR_ADDR_NML = 0x1000_004C, LDR_SIZE_NML = 0x0000_6000, DEST_ADDR_NML= 0x0010_2000, and others = 0x0000_0000,
CHECK_SUM is calculated as below ((0x0000_) is omitted in the formula):
CHECK_SUM = (0x1000) + (0x004C) + (0x6000) + (0x0010) + (0x2000) = (0x0000_905C)

3.5.6.3 Serial Host Interface Boot Mode

In serial host interface boot mode, this LSI is connected to external host CPU via serial host interface (SHOSTIF). Parameter and the loader program are transferred from external host CPU to this LSI via SHOSTIF, and this LSI boots that program.

Figure 3.8 shows the connection diagram of this LSI with external host CPU via serial host interface. At booting, x1, x2, x4, or x8 connection is available according to the SHOSTIF Boot Mode Setting (SHBMS) in OTP. SHBMS includes other boot mode setting as well as width of IO as shown in Table 3.12. During boot processing, values are set to the corresponding registers. To enable to change the setting, 4 areas (OTP address: 0x1B5 to 0x1B8) are available. If VALID bit of either address is 1, setting of the valid address is applied. If VALID bits of multiple addresses are 1, setting of the highest address in the valid addresses is applied. If no VALID bit in 4 areas, all the configuration bits are set to 0.



Note 1. Used I/Os are dependent on setting of the SPIMODE[1:0].

Figure 3.8 Connection diagram of this LSI with external host CPU (serial host interface)

Table 3.12 SHBMS contents

Bit	Bit name	Description
0	SCPH	Clock Phase Specifies value of SCPH bit in CTRLR0 register.
1	SCPOL	Clock Polarity Specifies value of SCPOL bit in CTRLR0 register.
2	—	Reserved
4:3	SPIMODE[1:0]	Width of IOs Specifies value of SPIMODE[1:0] bits in SHCFG register.
5	BYTESWAP	Byteswap mode Specifies value of BYTESWAP bit in SHCFG register.
6	ADDRESSING	Addressing mode (8-bit or 32-bit) Specifies value of ADDRESSING bit in SHCFG register.
14:7	—	Reserved
15	VALID	Valid flag to show the setting is valid. Writes 1 to indicate setting values of this area is valid.

Table 3.13 describes the setting values of the individual peripheral modules and registers at the time serial host interface boot mode finishes.

Table 3.13 Setting values of the individual peripheral modules and registers at the time serial host interface boot mode finishes (1 of 2)

Peripheral module	Register	Setting value at the time the boot processing finishes
Low power consumption	MSTPCRI	0x0000 0000
Module Reset	MRCTLI	0x0000 0000
SHOSTIF	CTRLR0	0x0000 0X00 X: Corresponding bits to the bit 0 and/or bit 1 in the SHBMS set 1b are 1b.
	ENR	0x0000 0001
	SHCFG	0x0000 000X X: Corresponding bits to either or multiples of bit 3 to bit 6 in the SHBMS set 1b are 1b.

Table 3.13 Setting values of the individual peripheral modules and registers at the time serial host interface boot mode finishes (2 of 2)

Peripheral module	Register	Setting value at the time the boot processing finishes
I/O ports	PMCm	Only the corresponding bits are 1b.
	PFCm	Only the corresponding bits are 0011b, 0100b, 0110b to 1001b, or 1100b.
	DRCTLm	Only the corresponding DRVn[1:0] bits are 10b and SRn bits are 1b.

3.5.7 SCI (UART) Boot Mode

The SCI (UART) Boot mode supports download and executes the loader program using SCI0 (UART) interface, which can be directly executed in internal memory. When the LSI transmits the start message to the PC by serial communication, the PC starts to transmit the loader program and it is transferred to the internal tightly coupled memory (TCM). After completion of the transmission, the processing of the loader program is executed.

Figure 3.9 shows the connection diagram of this LSI with a PC.

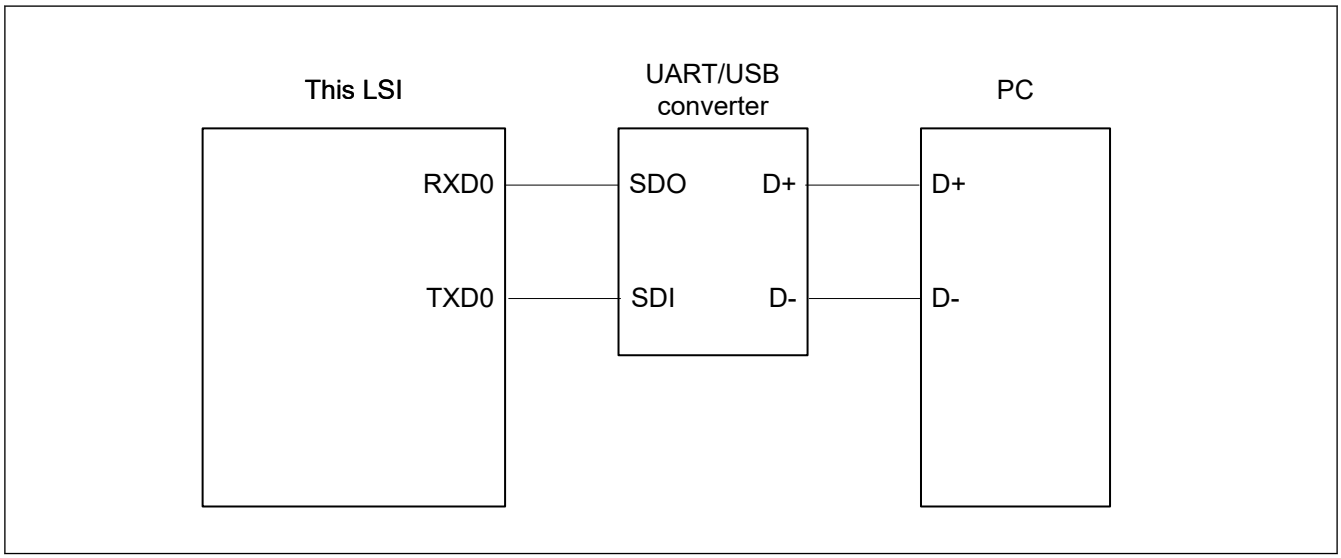


Figure 3.9 Connection diagram of this LSI with a PC

3.5.7.1 Operation Setting in SCI Boot Mode

After the reset is released, if the boot processing starts in SCI boot mode, the following setting values are used:

- CPU clock (CPU0CLK): 200 MHz
- SCI mode: Asynchronous mode
- Start bit: 1 bit
- Parity bit: None
- Data: 8 bits
- Stop bit: 1 bit
- Baud rate: 115200 bps

3.5.8 USB Boot Mode

The USB Boot mode supports download and executes the loader program using the USB 2.0 interface, which can be directly executed in internal memory. When the LSI receives the Open request from PC, the LSI transmits the start message to the PC. The PC starts to transmit the loader program and it is transferred to the internal tightly coupled memory (TCM). After completion of the transmission, the processing of the loader program is executed.

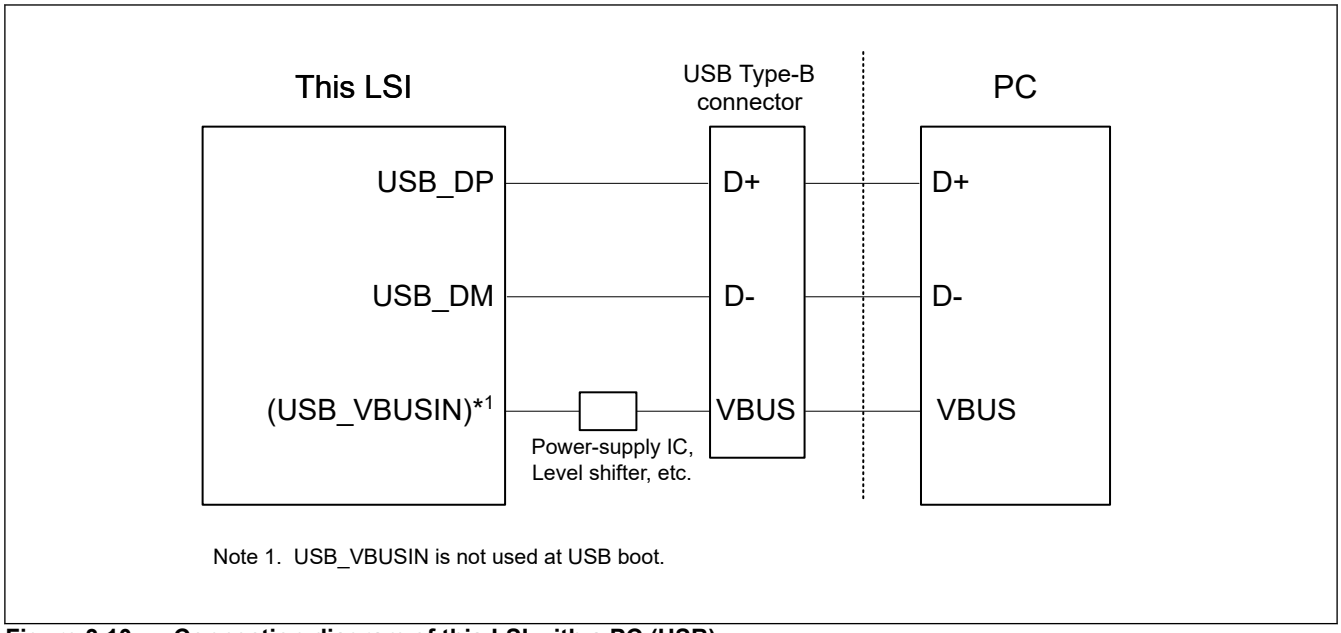


Figure 3.10 Connection diagram of this LSI with a PC (USB)

3.5.8.1 Operation Setting in USB Boot Mode

After the reset is released, if the boot processing starts in USB boot mode, the following setting values are used:

- CPU clock (CPU0CLK): 200 MHz
- USB mode: High speed, function mode
- USB function: USB Communication Device Class (CDC)

Note: After USB cable is connected with the LSI, USB cable should not be disconnected until the loader program is executed.

3.5.9 Motorola S-record Format for SCI/USB Boot

SCI/USB boot uses S0, S3, and S7 record type only. The record field is shown in Figure 3.11.

S-record Format

S-Type	Byte Count	Address	Data	Check sum
S0	2 byte	2 byte	2 byte × n (n = 1, 2, …)	2 byte

S-Type	Byte Count	Address	Data	Check sum
S3	2 byte	4 byte	2 byte × n (n = 1, 2, …)	2 byte

S-Type	Byte Count	Address	Check sum
S7	2 byte	4 byte	2 byte

Figure 3.11 S-record format for SCI/USB boot

3.5.10 SCI Boot Procedure

This boot procedure assumes using terminal emulator.

Table 3.14 SCI boot procedure

Steps	What to do
1	PC will receive "SCI Download mode (Normal SCI boot)" message after power-on-reset is released and LSI initialization is done on SCI boot mode.
2	The LSI waits to receive "HDR NM" data of Motorola S-record format, S0 record type, from the PC.
3	After received the data, the LSI will send out the following message to the PC. "--- Load Program to BTCM ----------"
4	The LSI can receive a loader program by Motorola S-record format, S3 record type, from the PC. The received data will be changed to binary data and forwarded to internal memory.
5	When SCI boot program receives Motorola S-record format, S7 record type, that is a notification from the PC to terminate the transfer.
6	The LSI will send out the following message to the PC. "--- Start Boot Program on BTCM ----------"
7	An end process is executed and the loader program is transferred to BTCM.

When an error happens, an error code is returned and the boot program is aborted. The error condition is below.

- SCI communication error
- Receive Motorola S-record format except S0, S3, and S7 record
- Check sum error of S-record format
- Addressing: Outside of BTCM area
- Loader program size: Outside of BTCM capacity

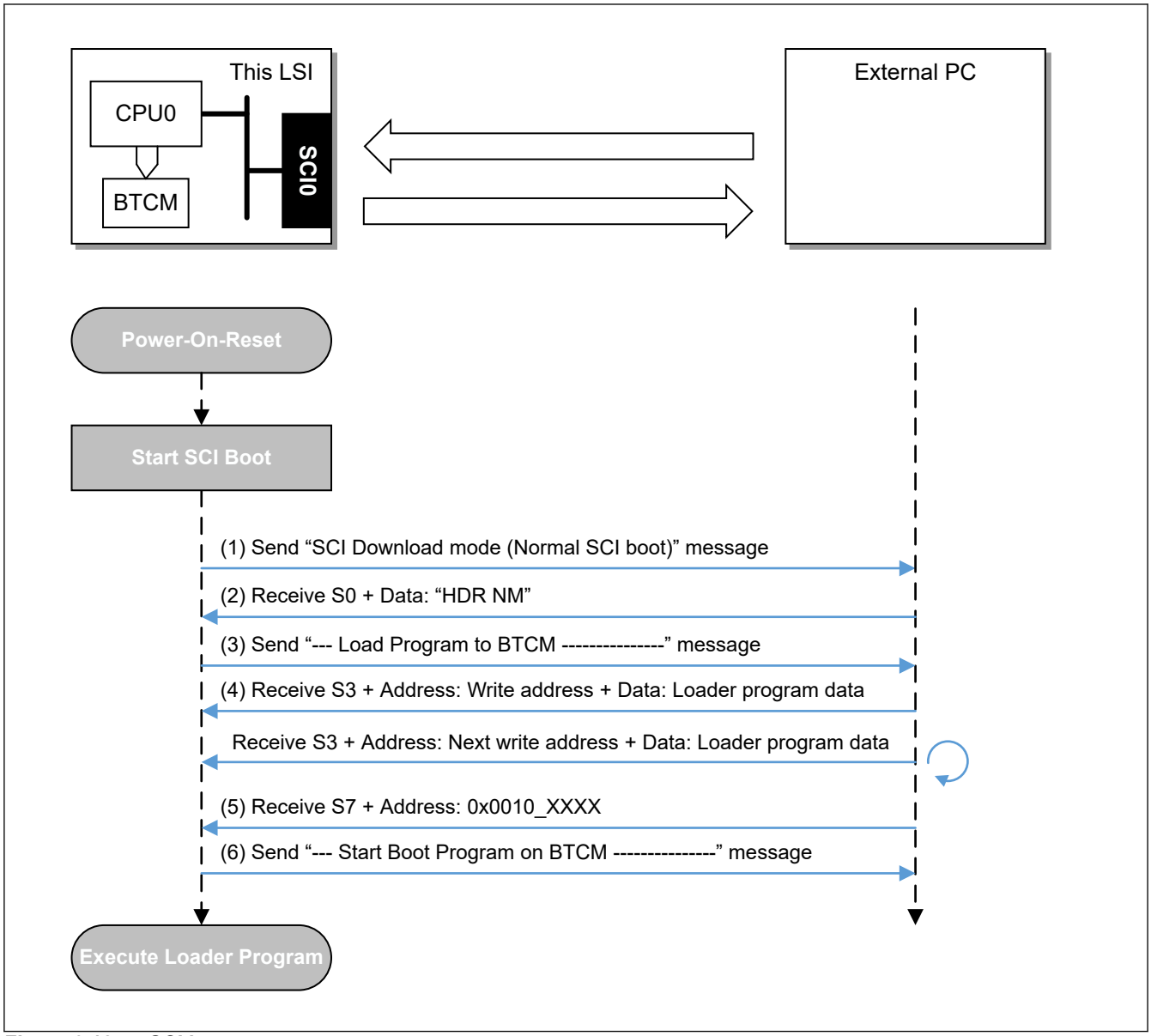


Figure 3.12 SCI boot sequence

3.5.11 USB Boot Procedure

This boot procedure assumes using terminal emulator.

Table 3.15 USB boot procedure (1 of 2)

Steps	What to do
1	After power-on-reset is released, LSI initialization and USB enumeration are performed on USB boot mode. Then, the LSI waits to receive "OPEN" data of Motorola S-record format, S0 record type, from the PC.
2	PC will receive "USB Download mode (Normal USB boot)" message.
3	The LSI waits to receive "HDR NM" data of Motorola S-record format, S0 record type, from the PC.
4	After received the data, the LSI will send out the following message to the PC. "--- Load Program to BTCM ----------"
5	The LSI can receive a loader program by Motorola S-record format, S3 record type, from the PC. The received data will be changed to binary data and forwarded to internal memory.
6	When USB boot program receives Motorola S-record format, S7 record type, that is a notification from the PC to terminate the transfer.

Table 3.15 USB boot procedure (2 of 2)

Steps	What to do
7	The LSI will send out the following message to the PC. "--- Start Boot Program on BTCM ----------"
8	An end process is executed and the loader program is transferred to BTCM.

When an error happens, an error code is returned and the boot program is aborted. The error condition is below.

- USB communication error
- Receive Motorola S-record format except S0, S3, and S7 record
- Check sum error of S-record format
- Addressing: Outside of BTCM area
- Loader program size: Outside of BTCM capacity

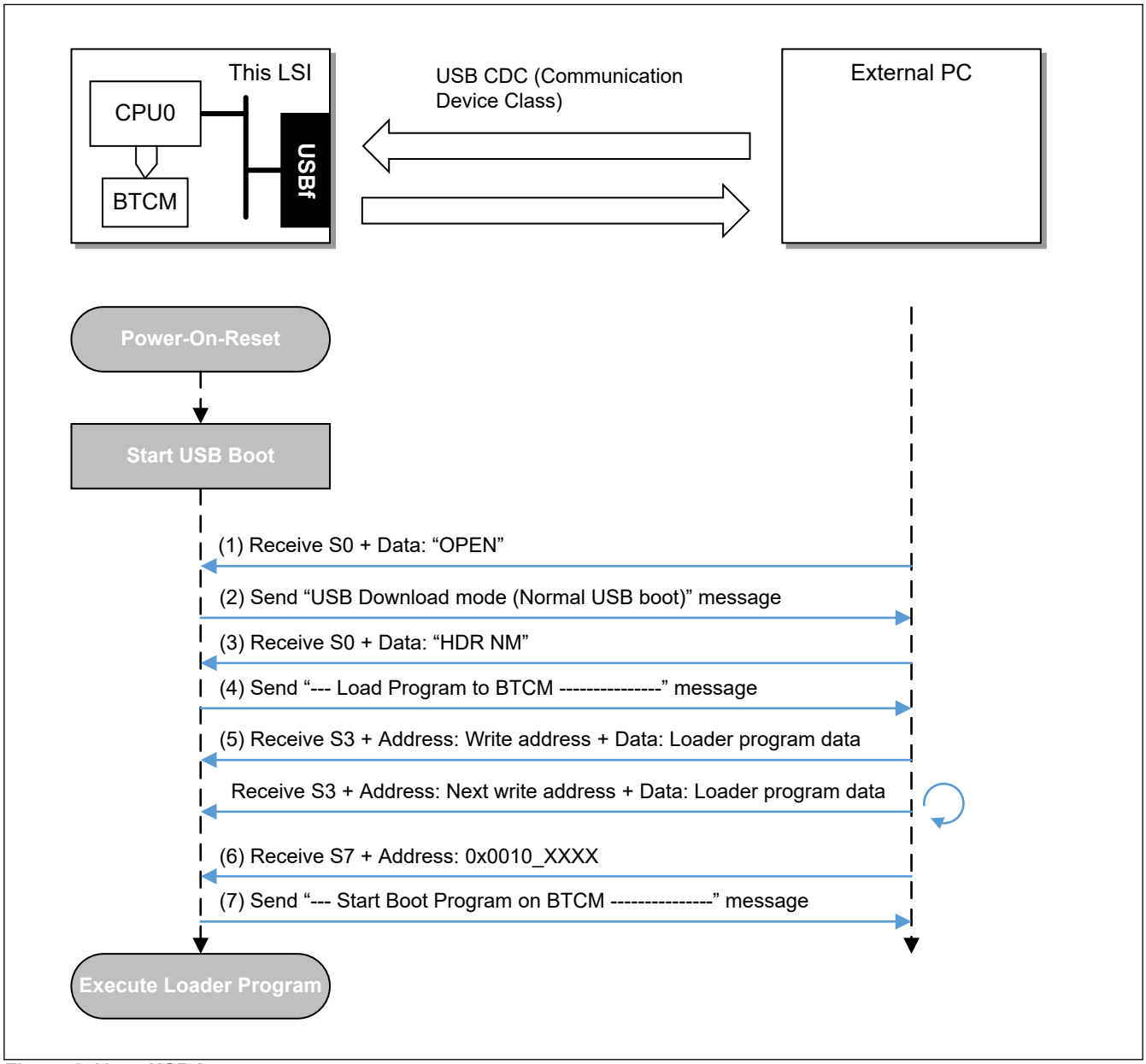


Figure 3.13 USB boot sequence

3.5.12 SCI/USB Boot Disable

The SCI/USB boot can be disabled by setting SCI/USB Boot Mode Disable area (SUDIS) in OTP.

If disable is selected, this LSI prohibits connection by SCI/USB boot mode permanently and boot fails.

3.5.13 Usage Note

3.5.13.1 Important Notice for MDVn Pin Setting

The LSI supports 1.8 V or 3.3 V operation voltage of I/O domain for BGA package. Operation voltage of I/O domain is selected as described in section 3.2. Types and Selection of Operating Modes.

MDVn setting and supply voltage for VCC1833_n must be matched. Otherwise it may cause malfunctions or permanent damage to the device. Table 3.16 shows a list of MDVn pins.

When 1.8 V operation voltage is selected for an I/O domain, 3.3 V should not be applied to the I/O domain because the I/O domain can operate at 1.8 V only. If 3.3 V is applied to the domain, the device may be permanently damaged. Table 3.3 shows 1.8 V or 3.3 V selectable I/O domains.

Table 3.16 List of MDVn pins

I/O voltage setting pins	Power supply	Pin number	Pin name
		196 FPGA	
MDV2	VCC1833_2	C8	P20_3 / GMAC_PTPOUT1 / MDV2 / ESC_LEDERR / CANTX1
MDV3	VCC1833_3	B3	P20_4 / GMAC_PTPOUT0 / MDV3 / ESC_LINKACT1

Note: Appropriate MDVn input level setting is required until operation mode setting is done. An operating mode is set according to the input level of the mode setting pins (MD[2:0], MDV[3:2], and MDD) when RES# or TRST# is set to high from the pin reset state (RES# and TRST# pins are both low). See section 3.2. Types and Selection of Operating Modes and section 5, Reset. For this reason, MDVn pins can be used as other functions. However, do not pull-up or pull-down MDVn to correct voltage level externally. Also, do not let MDVn pins be floated.

3.5.13.2 Bus Conflict on 16-Bit and 32-Bit Bus Boot Modes

If a reset such as #RES pin reset, system software reset, and error reset occur when the bus is shared with NOR Flash and SDRAM, the boot operation may not work due to bus conflict. In such cases, the following either measures should be taken.

- The reset should be asserted after SDRAM state is changed to no data output like SDRAM power-down mode.
- A circuit is implemented to turn the SDRAM power on/off when a reset occurs.

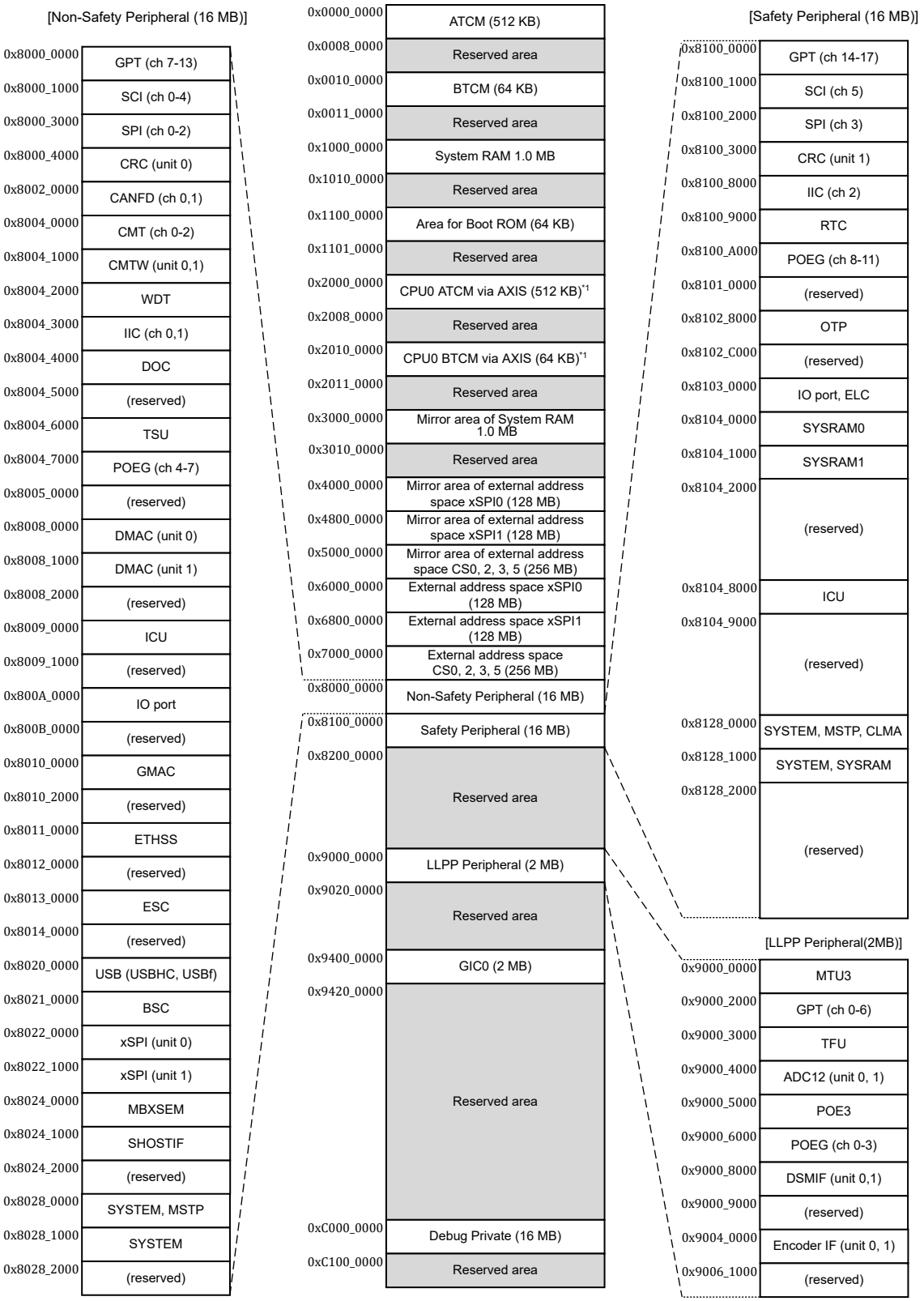
3.5.13.3 Serial Flash Reset on xSPI Boot Modes (x1 Boot Serial Flash)

If a reset occurs when serial flash configuration is changed other than 1S-1S-1S, the boot operation may not work. In this case, external H/W reset for serial flash should be required.

4. Address Space

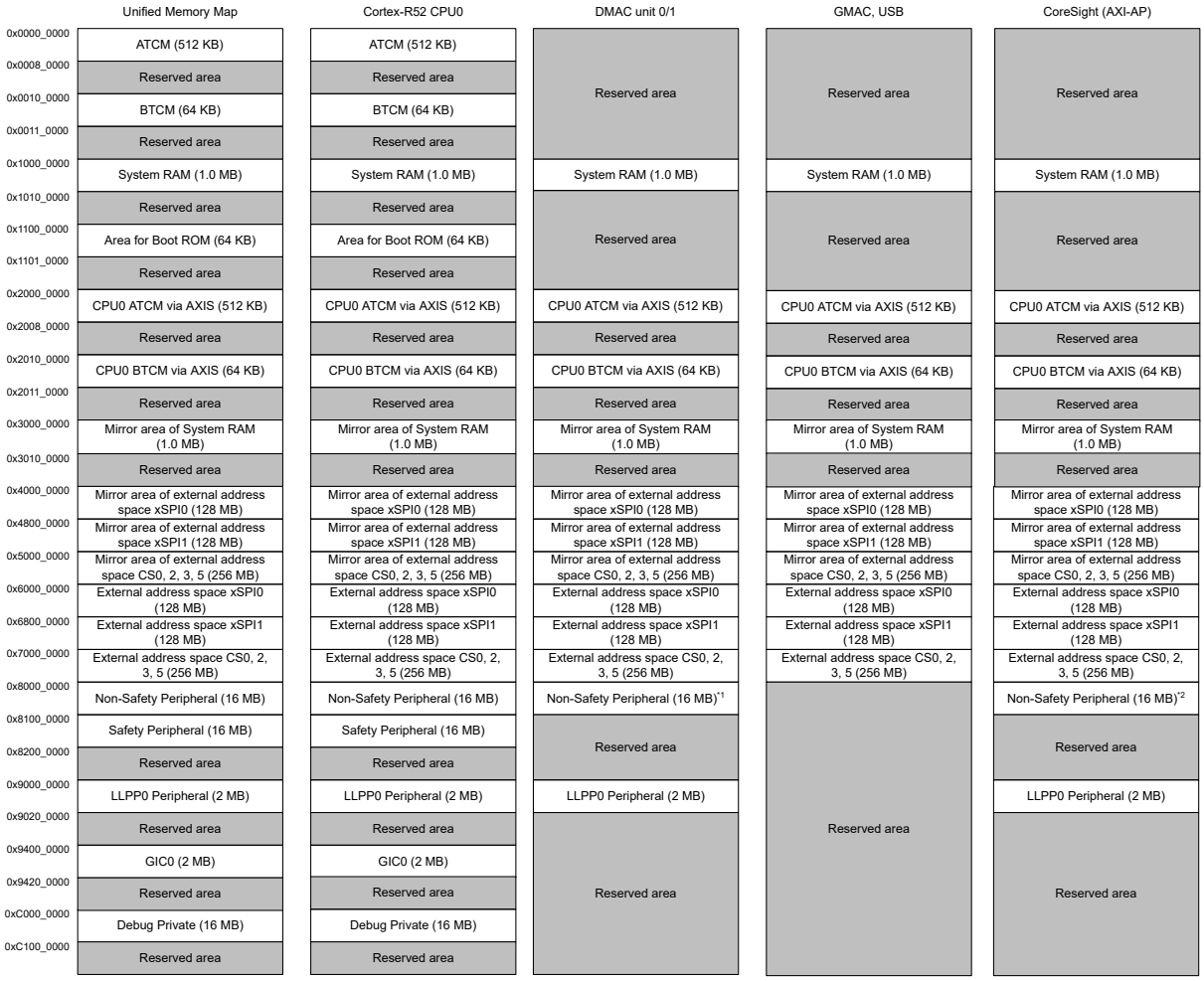
4.1 Address Space

The LSI has a 4-GB address space ranging from 0x0000_0000 to 0xFFFF_FFFF. That is, up to 4 GB of program and data areas can be accessed linearly. Figure 4.1, Figure 4.2, and Figure 4.3 show the memory maps of the bus masters. Accessible areas will differ depending on the operating mode and the states of control bits.



Note: Reserved area should not be accessed.
Note 1. Address to access the tightly coupled memory for CPU0 from bus master through AXIS interface.

Figure 4.1 Unified memory map



Note 1. DMAC, ICU, IO port, and SYSTEM registers are not accessible.
Note 2. IO Port registers are not accessible.

Figure 4.2 Memory map each bus master

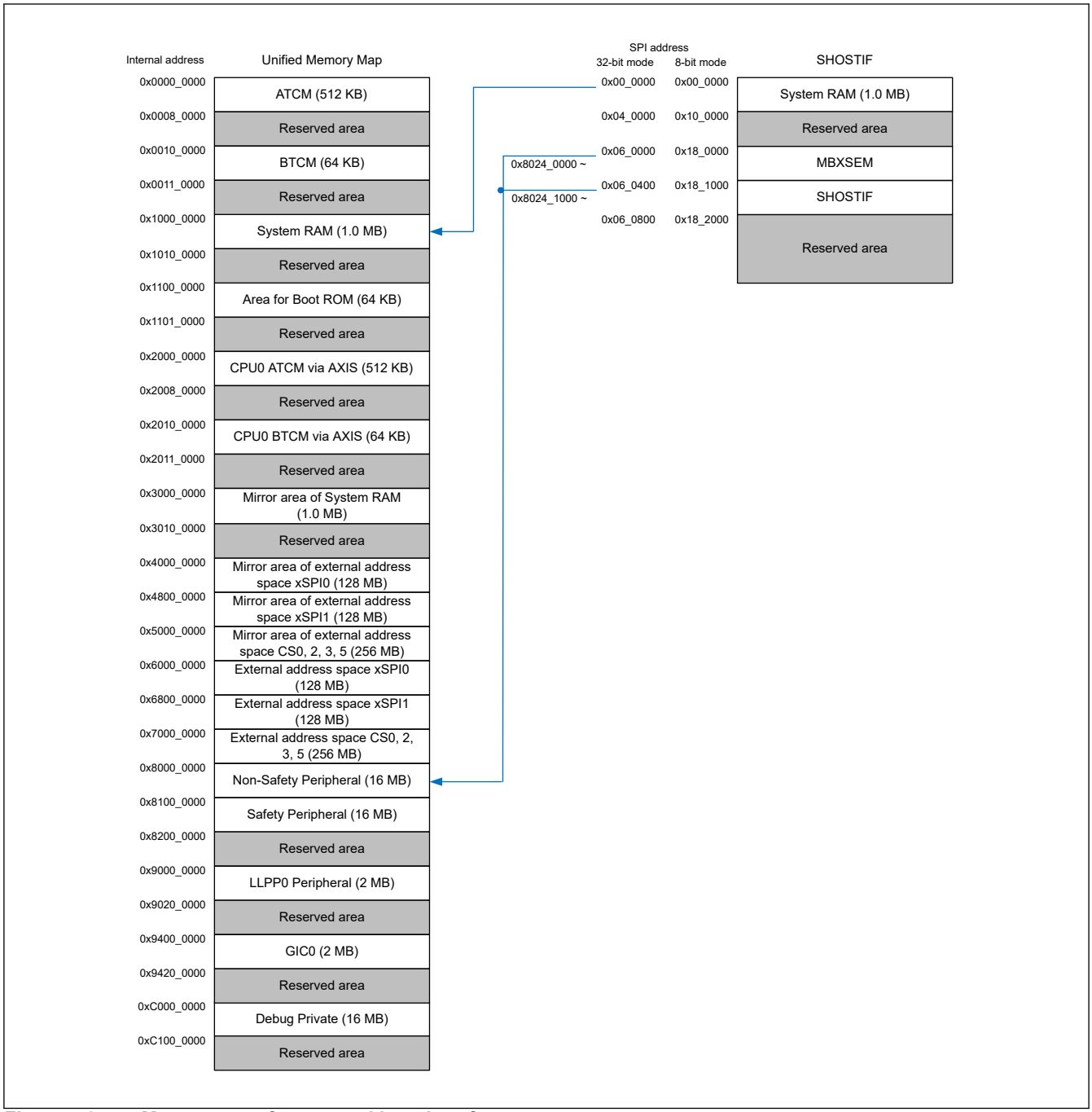


Figure 4.3 Memory map for external host interface

5. Reset

5.1 Overview

Available reset types are RES# pin reset, system software reset, CPU0 software reset, and error reset. Table 5.1 lists the reset names and sources.

Table 5.1 Reset names and sources

Reset name	Source
RES# pin reset	The low level is applied to the RES# pin
System software reset	SWRSYS register setting
CPU0 software reset	SWRCPU0 register setting
Error reset	Reset request from the ICU

The internal states and pins are initialized by reset.

Table 5.2 lists the reset targets to be initialized for each reset type. For details on reset control during debugging, see section 9.3.4. Reset Configuration and the Method of Connecting with the Emulator.

Table 5.2 Targets to be initialized for each reset type (✓: to be initialized, —: no change)

Reset target	Reset source			
	RES# pin reset	System software reset	CPU0 software reset	Error reset
RES# Pin Reset flag (RSTSR0.TRF)	—	✓	—	✓
System Software Reset Detect flag (RSTSR0.SWRSF)	✓	—	—	✓
CPU0 Software Reset Detect flag (RSTSR0.SWR0F)	✓	✓	—	✓
Error Reset Detect flag (RSTSR0.ERRF) Error Event Capture Status registers	✓	✓	—	—
Pin state	✓	✓	—	✓
Operating mode	✓*1	—*2	—	—*2
CPU0	✓	✓	✓	✓
WDT0 for CPU0	✓	✓	✓	✓
System SRAM, TCM, cache	✓	✓	—	✓
Peripheral modules, registers and internal states other than listed	✓	✓	—	✓
RSTOUT# pin output	✓(low)*3	✓(low)*3	—	✓(low)*3

Note 1. An operating mode is selected according to the input level of the mode setting pin (MD2, MD1, and MD0) when the pin reset state (RES# and TRST# pins are both low) is released. For details on the operating mode, see section 3.2. Types and Selection of Operating Modes.

Note 2. The operating mode is not initialized in response to this type of reset and the LSI is in the operating mode that was selected following the previous release from the reset state applied by the low level on the RES#. For details, see section 3, Operating Modes.

Note 3. For the low output period, see section 5.3.6. Reset Output Pin (RSTOUT#).

Table 5.3 lists the input and output pins related to the reset.

Table 5.3 Input and output pins related to reset

Pin name	I/O	Function
RES#	Input	Reset pin. Use this pin to reset the entire LSI except the debugging circuit and the TAP (Test Access Port). Since the power-on reset circuit is not incorporated, a reset circuit must be implemented outside this LSI. For an example of configuration of the external reset circuit, see section 9.3.4. Reset Configuration and the Method of Connecting with the Emulator.
TRST#	Input	Test reset pin. Use this pin to reset the TAP. If you design a board which enables an emulator, set the TRST# pin to low level during the same period as the RES# pin at the time of power-on. The TRST# pin should also be controllable independently. When unused, this pin must be set to low level or connected to the same signal as the RES# pin. For details, see section 9.3.4. Reset Configuration and the Method of Connecting with the Emulator.
RSTOUT#	Output	Reset output pin. This pin outputs low-level signal on occurrence of a reset. For details, see section 5.3.6. Reset Output Pin (RSTOUT#). This pin can be used for resetting the external device.

Note: For details on resetting the debugging circuit, see section 9, Debugging Interface.

5.2 Register Descriptions

The Reset Status Register 0 contains bits assigned to respective reset types to indicate reset generation sources. Registers RSTSR0, SWRSYS, SWRCPU0, MRCTLA, MRCTLE, MRCTLI, MRCTLJ, MRCTLK, and MRCTLL are protected by the register write protection function. To write these registers, clear the write protection bit [1] in the Protect Register (PRCRN for RSTSR0 and MRCTLE, RPCRS for SWRSYS and SWRCPU0). For details on the register write protection, see section 10, Register Write Protection Function.

5.2.1 RSTSR0 : Reset Status Register 0

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x200

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	SWR0F	SWRSF	ERRF	TRF	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	x*1	x*1	x*1	x*1	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	TRF	RES# Pin Reset Detect Flag This bit indicates the detection of RES# pin reset. 0: RES# pin reset not detected 1: RES# pin reset detected	R/W*2
2	ERRF	Error Reset Detect Flag This bit indicates the detection of error reset. 0: Error reset not detected 1: Error reset detected	R/W*2
3	SWRSF	System Software Reset Detect Flag This bit indicates the detection of system software reset. 0: System software reset not detected 1: System software reset detected	R/W*2
4	SWR0F	CPU0 Software Reset Detect Flag This bit indicates the detection of CPU0 software reset. 0: CPU0 software reset not detected 1: CPU0 software reset detected	R/W*2
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The value after reset depends on the reset source. The following table lists the sources for the setting (1) and clearing (0) of each bit.

Bit	Setting conditions (0 to 1)	Clearing conditions (1 to 0)
TRF	When the low level is input to the RES# pin.	• When system software reset or error reset occurs • When 0x00000000 is written to RSTSR0 after RSTSR0 is read
ERRF	When an error predefined as a reset source by ICU setting occurs	• When the low level is input to the RES# pin or system software reset occurs • When 0x00000000 is written to RSTSR0 after RSTSR0 is read
SWRSF	When system software reset occurs	• When the low level is input to the RES# pin or error reset occurs • When 0x00000000 is written to RSTSR0 after RSTSR0 is read
SWR0F	When CPU0 software reset occurs	• When the low level is input to the RES# pin, system software reset or error reset occurs • When 0x00000000 is written to RSTSR0 after RSTSR0 is read

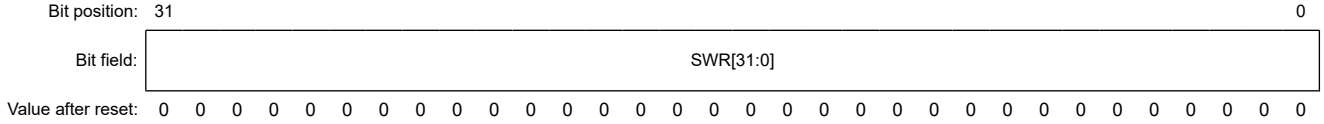
Note 2. Only 0 can be written to clear the flag.

The RSTSR0 register specifies reset generation sources.

5.2.2 SWRSYS : System Software Reset Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x210



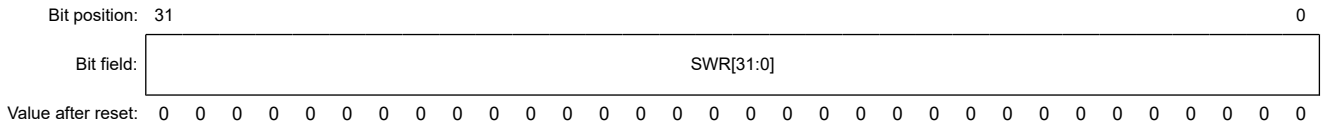
Bit	Symbol	Function	R/W
31:0	SWR[31:0]	System Software Reset 1. When 0x4321A501 is written, a system software reset occurs. After about 200 µs (Typ.) from the reset occurrence, it automatically releases the reset, starts the system in the selected boot mode in the operating mode pins, and the program is restarted. 2. Other than the above written value, nothing is changed.	W

The SWRSYS register controls system software reset.

5.2.3 SWRCPU0 : CPU0 Software Reset Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x220



Bit	Symbol	Function	R/W
31:0	SWR[31:0]	CPU0 Software Reset 1. When 0x4321A501 is written, CPU0 software reset occurs. CPU0 and WDT0 for CPU0 are in reset state. Reset is not released automatically. The RES# pin reset required to release reset. 2. When 0x4321A502 is written, CPU0 software reset occurs. CPU0 and WDT0 for CPU0 are in reset state. After the below time passes from reset occurrence, reset is released automatically. ● SCKCR2.DIVSELSUB = 0: around 1.28 µs ● SCKCR2.DIVSELSUB = 1: around 1.70 µs Prohibit to write 0x00000000 to this register before above time passes. 3. Other than the above written value, nothing is changed. These bits are read as 0x00000001 during reset, and as 0x00000000 when reset is released. Be careful that after writing 0x4321A501 or 0x4321A502 to this register, transfer CPU0 to WFI state manually. After confirming CPU0 is in WFI state by hardware, CPU0 and WDT0 for CPU0 are reset. It is prohibited to write 0x4321A501 after writing 0x4321A502 and 0x4321A502 after writing 0x4321A501.	R/W

The SWRCPU0 register controls the CPU0 software reset.

5.2.4 MRCTLA : Module Reset Control Register A

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x240

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	MRCTLA05	MRCTLA04	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	MRCTLA04	xSPI Unit 0 Reset Control This bit controls the reset of the xSPI unit 0. 0: xSPI Unit 0 is released from reset 1: xSPI Unit 0 is in the reset state	R/W
5	MRCTLA05	xSPI Unit 1 Reset Control This bit controls the reset of xSPI unit 1. 0: xSPI Unit 1 is released from reset 1: xSPI Unit 1 is in the reset state	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

The MRCTLA register controls the reset of xSPI. This register is available in Non Safety region.

5.2.5 MRCTLE : Module Reset Control Register E

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x250

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	MRCTLE06	MRCTLE05	MRCTLE04	MRCTLE03	—	MRCTLE01	MRCTLE00
Value after reset:	0	0	0	0	0	0	0	0	0	1	1	1	1	0	1	1

Bit	Symbol	Function	R/W
0	MRCTLE00	GMAC (PCLKH clock domain) Reset Control This bit controls the reset of GMAC (PCLKH clock domain). 0: GMAC (PCLKH clock domain) is released from reset 1: GMAC (PCLKH clock domain) is in the reset state	R/W
1	MRCTLE01	GMAC (PCLKM clock domain) Reset Control This bit controls the reset of GMAC (PCLKM clock domain). 0: GMAC (PCLKM clock domain) is released from reset 1: GMAC (PCLKM clock domain) is in the reset state	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	MRCTLE03	ESC (Bus clock domain) Reset Control This bit controls the reset of ESC (Bus clock domain). 0: ESC (Bus clock domain) is released from reset 1: ESC (Bus clock domain) is in the reset state	R/W
4	MRCTLE04	ESC (IP clock domain) Reset Control This bit controls the reset of ESC (IP clock domain). 0: ESC (IP clock domain) is released from reset 1: ESC (IP clock domain) is in the reset state	R/W
5	MRCTLE05	Ethernet Subsystem Register Reset Control This bit controls the reset of Ethernet Subsystem register. 0: Ethernet Subsystem register is released from reset 1: Ethernet Subsystem register is in the reset state	R/W
6	MRCTLE06	MII Converter Reset Control This bit controls the reset of MII converter. 0: MII converter is released from reset 1: MII converter is in the reset state	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

The MRCTLE register controls the reset of peripheral modules related to Ethernet.

5.2.6 MRCTLI : Module Reset Control Register I

Base address: SYSC_S = 0x8128_0000

Offset address: 0x0260

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MRCTLI03	MRCTLI02	MRCTLI01	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	MRCTLI01	SHOSTIF (Master bus clock domain) Reset Control 0: Reset released 1: Reset state	R/W
2	MRCTLI02	SHOSTIF (Slave bus clock domain) Reset Control 0: Reset released 1: Reset state	R/W
3	MRCTLI03	SHOSTIF (IP clock domain) Reset Control 0: Reset released 1: Reset state	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

The MRCTLI register controls reset of each IP. This register is available in Safety region.

5.2.7 MRCTLJ : Module Reset Control Register J

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x264

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MRCTLJ03	MRCTLJ02	MRCTLJ01	MRCTLJ00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MRCTLJ00	AFMT0 Reset Control This bit controls the reset of AFMT0. 0: AFMT0 is released from reset 1: AFMT0 is in the reset state	R/W
1	MRCTLJ01	HDSL0 Reset Control This bit controls the reset of HDSL0. 0: HDSL0 is released from reset 1: HDSL0 is in the reset state	R/W

Bit	Symbol	Function	R/W
2	MRCTLJ02	BISS0 Reset Control This bit controls the reset of BISS0. 0: BISS0 is released from reset 1: BISS0 is in the reset state	R/W
3	MRCTLJ03	ENDAT0 Reset Control This bit controls the reset of ENDAT0. 0: ENDAT0 is released from reset 1: ENDAT0 is in the reset state	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

The MRCTLJ register controls the reset of Encoder Interface unit 0. This register is available in Non Safety region.

5.2.8 MRCTLK : Module Reset Control Register K

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x268

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MRCTLK03	MRCTLK02	MRCTLK01	MRCTLK00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MRCTLK00	AFMT1 Reset Control This bit controls the reset of AFMT1. 0: AFMT1 is released from reset 1: AFMT1 is in the reset state	R/W
1	MRCTLK01	HDSL1 Reset Control This bit controls the reset of HDSL1. 0: HDSL1 is released from reset 1: HDSL1 is in the reset state	R/W
2	MRCTLK02	BISS1 Reset Control This bit controls the reset of BISS1. 0: BISS1 is released from reset 1: BISS1 is in the reset state	R/W
3	MRCTLK03	ENDAT1 Reset Control This bit controls the reset of ENDAT1. 0: ENDAT1 is released from reset 1: ENDAT1 is in the reset state	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

The MRCTLK register controls the reset of Encoder Interface unit 1. This register is available in Non Safety region.

5.3 Operation

5.3.1 RES# Pin Reset

This reset occurs when a signal arrives at the RES# pin from the externally connected reset circuit. When the signal becomes low at the RES# pin, all on-going processes are aborted and the LSI enters the reset state. To successfully reset the LSI, the RES# pin should be held at the low level for the specified time after power-on. For details on the reset configuration, see section 9.3.4. Reset Configuration and the Method of Connecting with the Emulator.

After the reset is released, CPU0 starts boot processing. When a RES# pin reset occurs, the RSTSR0.TRF flag is set to 1.

5.3.2 System Software Reset

System software reset occurs when 0x4321A501 is written to the SWRSYS register. When system software reset occurs, all on-going processes are aborted and the LSI enters the reset state. After the reset is released, CPU0 starts boot processing.

When system software reset occurs, the RSTSR0.SWRSF flag is set to 1.

5.3.3 CPU0 Software Reset

CPU0 software reset applies to CPU0 and WDT0 for CPU0. It occurs when 0x4321A501 or 0x4321A502 is written to the SWRCPU0 register and CPU0 is transferred to WFI state by WFI instruction. Do not write the SWRCPU0 register again until CPU0 is transferred to WFI state and SWRCPU0 register is changed to 0x00000001.

When 0x4321A501 is written, reset is not released automatically. To release the reset state of CPU0 software reset, the RES# pin reset is required.

When 0x4321A502 is written, reset is released automatically. Do not write 0x00000000 to the SWRCPU0 register.

5.3.4 Error Reset

This reset is generated in response to a reset request from ICU. ICU receives critical errors, such as oscillation stop detection, from individual modules in the LSI, and generates reset requests corresponding to respective errors. For details on the ICU operation, see section 11, Interrupt Controller (ICU).

When error reset occurs, all on-going processes are aborted and the LSI enters the reset state. After the reset is released, CPU0 starts boot processing. When the error reset occurs, the RSTSR0.ERRF flag is set to 1.

5.3.5 Determination of Reset Generation Source

Reading the RSTSR0 register determines which reset source was used for reset execution. Figure 5.1 shows an example of the flow to identify a reset generation source.

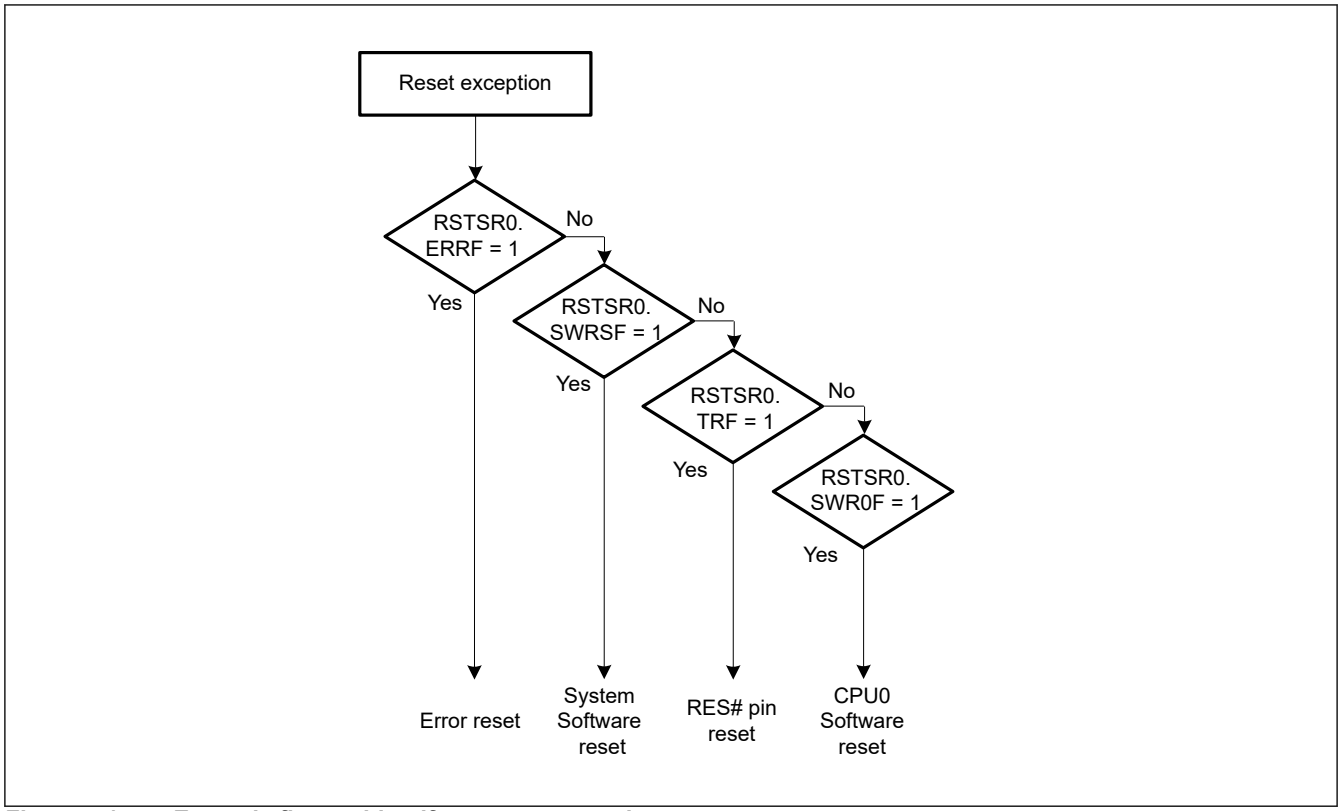


Figure 5.1 Example flow to identify a reset generation source

5.3.6 Reset Output Pin (RSTOUT#)

The reset output pin (RSTOUT#) outputs the low level on occurrence of a reset when the input level of the RES# pin is low. It also outputs the low level on occurrence of a system software reset or an error reset. If the RES# pin remains low for a specified time period and then changes to high, the reset output remains low for 200 µs (Typ.) and then changes to high. Similarly, the reset output remains low for 200 µs (Typ.) following a system software reset or an error reset, and then changes to high.

5.3.7 Noise Cancellation for Reset Input

Noise cancellation using analog delay is applied to the RES# and TRST# pins. This can eliminate noises within 100 ns (Min.).

5.4 Usage Note

5.4.1 Notes on Module Reset Control Register Operation

1. To secure processing after release from a module reset, dummy read the same register at least three times after writing to initiate release from the module reset, and only then proceed with the subsequent processing.
2. When module is reset once and released again, make sure that the target bit of module reset control register is set to 1 by reading the register before releasing from a module reset. Then release from a module reset.
3. When controlling the module resetting of modules that can initiate DMA transfer or that otherwise have operations related to those of other blocks, confirm that DMA transfer initiated by such modules and other points of interlinked operation has been completed before releasing modules from or placing modules in the module reset state. For set products in which such target modules are not used, design of the control of the module reset is on the assumption that the modules are initially stopped. Therefore, the absence of unexpected operations in states other than the above is not guaranteed. For example, operation after releasing modules from or placing modules in the module reset state while a module is running is not guaranteed.

5.4.2 Connection of Reset Output Pin (RSTOUT#)

The low level is output for a specified period of time on RSTOUT# after deassertion of the signal on the RES# pin. This means that RSTOUT# should not be directly connected as a reset signal for a flash memory to be used in booting the LSI. Otherwise, release of the LSI itself from the reset state may precede that for the flash memory.

However, using RSTOUT# as the reset signal for external device is still possible as long as the timing requirement imposed by the output on the RSTOUT# pin is satisfied.

RSTOUT# is shared function for port P17_5 but it is set by default after RES# pin reset, system software reset, and error reset. Even if function other than RSTOUT# is assigned, low level output from RSTOUT# is always effective when RES# pin reset, system software reset, or error reset occurs.

For details, see section 5.3.6. Reset Output Pin (RSTOUT#).

6. Clock Generation Circuit

6.1 Overview

The LSI incorporates a Clock Generation Circuit (CGC).

Table 6.1 and Table 6.2 show the specifications of the clock generation circuit. Figure 6.1 shows the block diagram and Table 6.3 shows the pin configuration of the clock generation circuit.

Table 6.1 Specifications of Clock Generation Circuit

Item	Description
Main clock oscillator	Resonator frequency: 25 MHz
	External clock input frequency: 25 MHz
	Connectable resonator or additional circuit: Ceramic resonator, crystal resonator Connection pin: EXTAL, XTAL
	External clock input instead of resonator Connection pin: EXTCLKIN
	Oscillation stop detection function: When the CLMA3 detects an abnormality in oscillation of the main clock, the system clock source is switched to LOCO and PLL output clocks are switched to LOCO or free-running clock of PLL, and MTU3 and GPT pins are placed in the high-impedance state.
PLL0 circuit (for main systems that include for example, a CPU, buses)	Input clock source: Main clock oscillator
	Input frequency: 25 MHz
	Frequency multiplication ratio: 96 multiplications
	Output clock frequency of the PLL0 circuit: 2400 MHz
	Oscillation abnormality detection function: When the CLMA0 detects an abnormality in oscillation of the PLL0, the PLL0 output clock is switched to the main clock, and MTU3 and GPT pins are placed in the high-impedance state.
PLL1 circuit (for clocks used in Ethernet)	Input clock source: Main clock oscillator
	Input frequency: 25 MHz
	Frequency multiplication ratio: 40 multiplications
	Output clock frequency of the PLL1 circuit: 1000 MHz
	Oscillation abnormality detection function: When the CLMA1 detects an abnormality in oscillation of the PLL1, the PLL1 output clock is switched to the main clock, and MTU3 and GPT pins are placed in the high-impedance state.
Low-speed on-chip oscillator (LOCO)	Oscillation frequency: 240 kHz
	Oscillation abnormality detection function: The CLMA2 can detect an abnormality in oscillation of the LOCO.
External clock input for JTAG (TCK)	Input frequency: 50 MHz (max.)

Table 6.2 Specifications of Clock Generation Circuit (internal clock) (1 of 3)

Parameter	Clock source	Supplied to	Frequency	Reset value
CPU0 clock (CPU0CLK)	Frequency-dividing clock for PLL0	CPU0 (Cortex-R52)	200 MHz or 150 MHz (ICLK × 1) 400 MHz or 300 MHz (ICLK × 2) 800 MHz or 600 MHz (ICLK × 4)	200 MHz
System clock (ICLK)	Frequency-dividing clock for PLL0	DMAC, ICU, CPU AXIM, BSC	200 MHz or 150 MHz	200 MHz

Table 6.2 Specifications of Clock Generation Circuit (internal clock) (2 of 3)

Parameter	Clock source	Supplied to	Frequency	Reset value
Peripheral module clock H (PCLKH)	Frequency-dividing clock for PLL0	Peripheral module (MTU3, GPT (LLPP, bus), TFU, DSMIF, POE3, POEG (LLPP), ADC12 (bus), xSPI (bus), GMAC, SHOSTIF), System SRAM	200 MHz or 150 MHz (ICLK × 1)	200 MHz
Peripheral module clock M (PCLKM)	Frequency-dividing clock for PLL0	Peripheral module (CRC, SCI (bus), SPI (bus), GPT (Non Safety, Safety), CANFD, GMAC, SHIP, OTP, Network Subsystem Reg, ESC (bus), USB (bus), GPIO, CLMA)	100 MHz or 75 MHz (ICLK/2)	100 MHz
Peripheral module clock L (PCLKL)	Frequency-dividing clock for PLL0	Peripheral module (POEG (Non Safety, Safety), IIC, DOC, CMT, CMTW, WDT, RTC (bus), TSU)	50 MHz or 37.5 MHz (ICLK/4)	50 MHz
Peripheral module clock for ADC12 (PCLKADC)	Frequency-dividing clock for PLL0	ADC12 (conversion)	25 MHz or 18.75 MHz (ICLK/8)	25 MHz
Peripheral module clock for GPT in LLPP (PCLKGPTL)	Frequency-dividing clock for PLL0	GPT (LLPP, operation)	400 MHz or 300 MHz (ICLK × 2)	400 MHz
Peripheral module clock for SCIn (PCLKSCIn) (n = 0 to 5)	Frequency-dividing clock for PLL0	SCIn (operation)	96 MHz or 75 MHz (Asynchronous mode) 100 MHz or 75 MHz (ICLK/2)	75 MHz (Asynchronous mode)
Peripheral module clock for SPIn (PCLKSPIn) (n = 0 to 3)	Frequency-dividing clock for PLL0	SPIn (operation)	96 MHz or 75 MHz (Asynchronous mode) 100 MHz or 75 MHz (ICLK/2)	75 MHz (Asynchronous mode)
Peripheral module clock for CANFD (PCLKCAN)	Frequency-dividing clock for PLL0	CANFD (serial)	40 MHz 80 MHz	40 MHz
Peripheral module clock for SHOSTIF (PCLKSHOST)	Frequency-dividing clock for PLL0	SHOSTIF (operation)	400 MHz	400 MHz
External bus clock (BSC_CLK, CKIO)	Frequency-dividing clock for PLL0	External bus	100.0 MHz or 75.0 MHz (ICLK/2) 66.7 MHz or 50.0 MHz (ICLK/3) 50.0 MHz or 37.5 MHz (ICLK/4) 40.0 MHz or 30.0 MHz (ICLK/5) 33.3 MHz or 25.0 MHz (ICLK/6) 28.6 MHz or 21.4 MHz (ICLK/7) 25.0 MHz or 18.75 MHz (ICLK/8)*1	50 MHz
Peripheral module clock for RTC (PCLKRTC)	Frequency-dividing clock for main clock	RTC (operation)	195.3 kHz	195.3 kHz
USB clock (USB_CLK48)	Frequency-dividing clock for PLL0	USB PHY	48 MHz	48 MHz
xSPIn serial clock (XSPI_CLKn) (n = 0, 1)	Frequency-dividing clock for PLL0	xSPIn (operation)	133.3, 100, 75, 50, 37.5, 25, 12.5 MHz*2	12.5 MHz
Ethernet clock B (ETCLKB)	Frequency-dividing clock for PLL1	GMAC, Converter	125 MHz	125 MHz
Ethernet clock C (ETCLKC)	Frequency-dividing clock for PLL1	ESC	100 MHz	100 MHz
Ethernet clock D (ETCLKD)	Frequency-dividing clock for PLL1	MII Converter	50 MHz	50 MHz

Table 6.2 Specifications of Clock Generation Circuit (internal clock) (3 of 3)

Parameter	Clock source	Supplied to	Frequency	Reset value
Ethernet clock E (ETCLKE)	Frequency-dividing clock for PLL1	ESC, MII Converter	25 MHz	25 MHz
CLMAn sampling clock (CLMAMCLKA) (n = 0, 1)	Main clock divided by 2	CLMA0, CLMA1	12.5 MHz	12.5 MHz
CLMA2 sampling clock (CLMAMCLKB)	Main clock divided by 256	CLMA2	97.7 kHz	97.7 kHz
CLMA3 sampling clock (CLMALCLKA)	LOCO output clock	CLMA3	240 kHz	240 kHz
CLMA0 monitor clock (CLMAPLCLK0)	PLL0 clock divided by 32	CLMA0	75 MHz	75 MHz
CLMA1 monitor clock (CLMAPLCLK1)	PLL1 clock divided by 16	CLMA1	62.5 MHz	62.5 MHz
CLMA2 monitor clock (CLMALCLK)	LOCO output clock	CLMA2	240 kHz	240 kHz
CLMA3 monitor clock (CLMAMCLK)	Main clock divided by 2	CLMA3	12.5 MHz	12.5 MHz
JTAG clock (JTAGTCK)	TCK	JTAG	Up to 50 MHz	—
Trace interface clock (TCLK)	Frequency-dividing clock for PLL0	CoreSight TPIU	100 MHz	100 MHz
Serial Host I/F clock (SHIFCLK)	HSPI_CLK	SHOSTIF	Up to 40 MHz	—
EnDat clock (PCLKENDAT)	Frequency-dividing clock for PLL0	ENDAT	100 MHz	100 MHz
BiSS-C clock (PCLKBISS)	Frequency-dividing clock for PLL0	BISS	20 MHz	20 MHz
HIPERFACE DSL clock (PCLKDSL)	Frequency-dividing clock for PLL0	HDSL	75 MHz	75 MHz
A-format clock (PCLKAFMT)	Frequency-dividing clock for PLL0	AFMT	80 MHz	80 MHz
ENCOUT clock (PCLKENCO)	Frequency-dividing clock for PLL0	ENCOUT	20 MHz	20 MHz
Ethernet PHY reference clock (ETHn_REFCLK) (n = 0 to 2)	Frequency-dividing clock for PLL1	— (External Ethernet PHY)	25 MHz	25 MHz
Ethernet PHY reference clock (RMIIn_REFCLK) (n = 0 to 2)	Frequency-dividing clock for PLL1	— (External Ethernet PHY)	50 MHz	50 MHz

Note 1. Max operating frequency depends on operating conditions (C, $T_j$min). For details, see section 52.5.3. Bus Timing.
Note 2. Max operating frequency depends on operating conditions (Voltage, SDR/DDR). For details, see section 52.5.5.12. xSPI Timing.

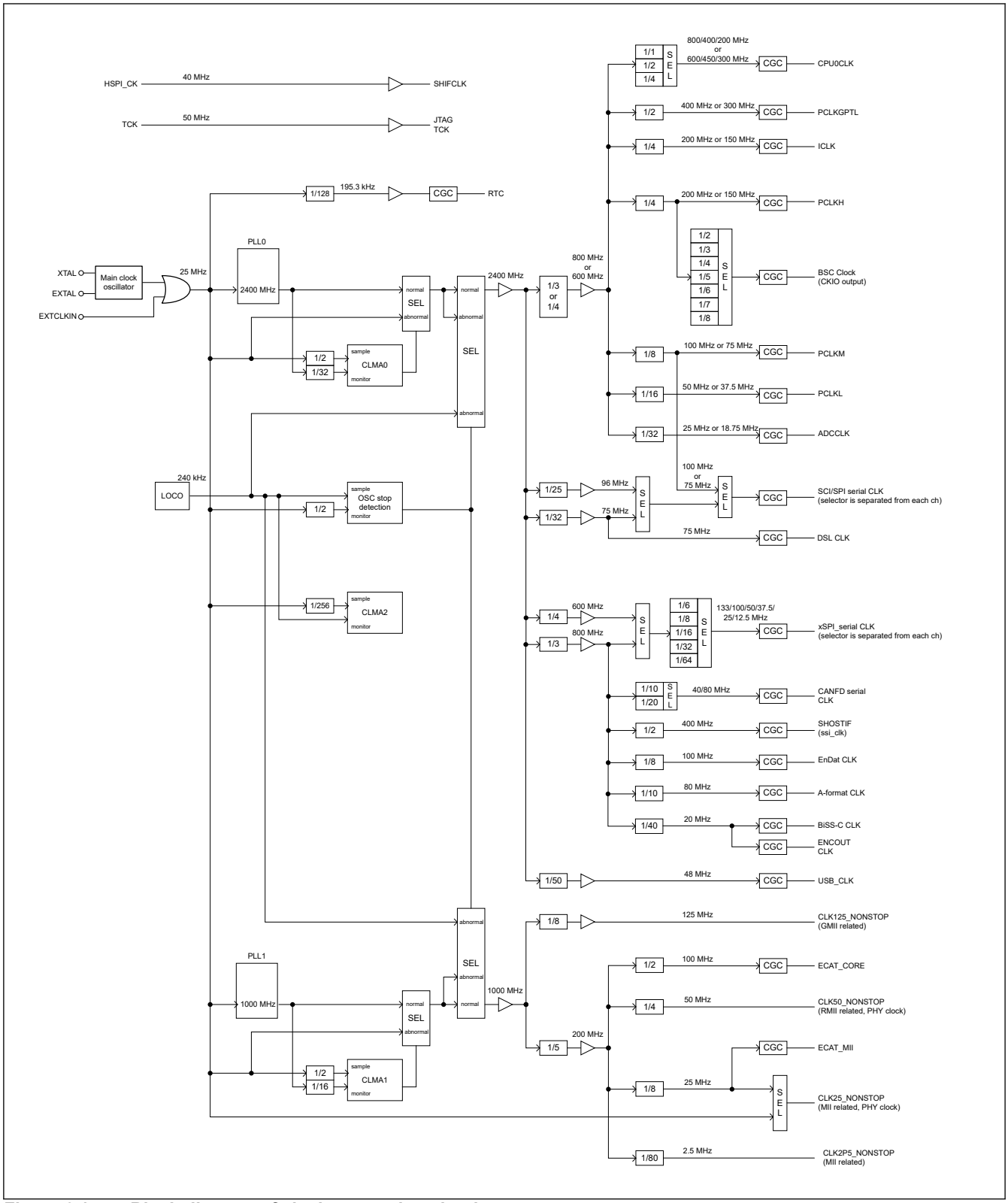


Figure 6.1 Block diagram of clock generation circuit

Table 6.3 Pin configuration of clock generation circuit (1 of 2)

Pin name	I/O	Description
XTAL	Output	These pins are used to connect a crystal resonator. When using an external clock signal, EXTAL pin must be driven low. XTAL pin must never be driven or loaded by anything other than crystal resonator. For details, see section 6.3.2. External Clock Input.
EXTAL	Input	

Table 6.3 Pin configuration of clock generation circuit (2 of 2)

Pin name	I/O	Description
EXTCLKIN	Input	This pin is used to input an external clock signal. When using a crystal resonator, this pin must be driven low.
TCK	Input	This pin is used to input the clock for the JTAG.
CKIO	Output	This pin is used to supply the external bus clock (CKIO) to external devices.
ETH0_REFCLK	Output	This pin is used to supply 25 MHz reference clock to the external Ethernet PHY.
ETH1_REFCLK	Output	This pin is used to supply 25 MHz reference clock to the external Ethernet PHY.
ETH2_REFCLK	Output	This pin is used to supply 25 MHz reference clock to the external Ethernet PHY.
RMII0_REFCLK	Output	This pin is used to supply 50 MHz reference clock to the external Ethernet PHY.
RMII1_REFCLK	Output	This pin is used to supply 50 MHz reference clock to the external Ethernet PHY.
RMII2_REFCLK	Output	This pin is used to supply 50 MHz reference clock to the external Ethernet PHY.

6.2 Register Descriptions

The registers related to the clock generation circuit can be write-protected. To write to the registers, specify bit 0 or 1 of the Protect Register (PRCRN and PRCRS) to cancel the write protection. For details, see section 10, Register Write Protection Function.

6.2.1 SCKCR : System Clock Control Register

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SCI4A SYNC SEL	SCI3A SYNC SEL	SCI2A SYNC SEL	SCI1A SYNC SEL	SCI0A SYNC SEL	SPI2A SYNC SEL	SPI1A SYNC SEL	SPI0A SYNC SEL	—	CLMA SEL	PHYS EL	FSEL CANF D	—	CKIO[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	DIVSE LXSPI 1	—	—	—	FSELXSPI1[2:0]			—	DIVSE LXSPI 0	—	—	—	FSELXSPI0[2:0]		
Value after reset:	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1	0

Bit	Symbol	Function	R/W
2:0	FSELXSPI0[2:0]	Set the frequency of the clock provided to xSPI Unit 0 in combination with bit 6 (DIVSELXSPI0). The combination is shown below. Both DIVSELXSPI0 and FSELXPI0 can be changed at the same time. Be careful that the set frequency is output after the intermediate frequency is output. Max operating frequency depends on operating conditions (Voltage, SDR/DDR). For details, see section 52.5.5.12. xSPI Timing.	R/W

FSELXSPI0[2:0]	DIVSELXSPI0 = 0	DIVSELXSPI0 = 1
000b	Setting prohibited	
001b	Setting prohibited	
010b	133.3 MHz	Setting prohibited
011b	100.0 MHz	75.0 MHz
100b	50.0 MHz	37.5 MHz
101b	25.0 MHz	Setting prohibited
110b	12.5 MHz	Setting prohibited
111b	Setting prohibited	

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>5:3</td><td>—</td><td>These bits are read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>6</td><td>DIVSELXSPI0</td><td>Select the base clock to generate serial clock for xSPI Unit 0
0: 800 MHz
1: 600 MHz</td><td>R/W</td></tr>
<tr><td>7</td><td>—</td><td>This bit is read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>10:8</td><td>FSELXSPI1[2:0]</td><td>Set the frequency of the clock provided to xSPI Unit 1 in combination with bit 14 (DIVSELXSPI1).
Same as bits [2:0] for xSPI Unit 0.</td><td>R/W</td></tr>
<tr><td>13:11</td><td>—</td><td>These bits are read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>14</td><td>DIVSELXSPI1</td><td>Select the base clock to generate serial clock for xSPI Unit 1
0: 800 MHz
1: 600 MHz</td><td>R/W</td></tr>
<tr><td>15</td><td>—</td><td>This bit is read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>18:16</td><td>CKIO[2:0]</td><td>Set the frequency of the external bus clock (CKIO) and the clock supplied to BSC in combination with the DIVSELSUB in the SCKCR2 register. The combination is shown below.
Max operating frequency depends on operating conditions (C, $T_j$min). For details, see section 52.5.3. Bus Timing.
<table>
<tr><th>CKIO[2:0]</th><th>SCKCR2.DIVSELSUB = 0</th><th>SCKCR2.DIVSELSUB = 1</th></tr>
<tr><td>000b</td><td>100 MHz</td><td>75 MHz</td></tr>
<tr><td>001b</td><td>66.7 MHz</td><td>50.0 MHz</td></tr>
<tr><td>010b</td><td>50.0 MHz</td><td>37.5 MHz</td></tr>
<tr><td>011b</td><td>40.0 MHz</td><td>30.0 MHz</td></tr>
<tr><td>100b</td><td>33.3 MHz</td><td>25.0 MHz</td></tr>
<tr><td>101b</td><td>28.6 MHz</td><td>21.4 MHz</td></tr>
<tr><td>110b</td><td>25.0 MHz</td><td>18.75 MHz</td></tr>
<tr><td>111b</td><td colspan="2">Setting prohibited</td></tr>
</table></td><td>R/W</td></tr>
<tr><td>19</td><td>—</td><td>This bit is read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>20</td><td>FSELCANFD</td><td>Select the frequency of the clock supplied to CANFD
0: 80.0 MHz
1: 40.0 MHz</td><td>R/W</td></tr>
<tr><td>21</td><td>PHYSEL</td><td>Select the Ethernet PHY reference clock output (ETHn_REFCLK, n = 0 to 2)
0: PLL1 divider clock
1: Main clock oscillator</td><td>R/W</td></tr>
<tr><td>22</td><td>CLMASEL</td><td>Select alternative clock when main clock abnormal oscillation is detected in CLMA3
0: LOCO
1: PLL</td><td>R/W</td></tr>
<tr><td>23</td><td>—</td><td>This bit is read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>24</td><td>SPI0ASYNCSEL</td><td>Select clock frequency when asynchronous serial clock is selected in SPI0
0: 75.0 MHz
1: 96.0 MHz</td><td>R/W</td></tr>
<tr><td>25</td><td>SPI1ASYNCSEL</td><td>Select clock frequency when asynchronous serial clock is selected in SPI1
0: 75.0 MHz
1: 96.0 MHz</td><td>R/W</td></tr>
<tr><td>26</td><td>SPI2ASYNCSEL</td><td>Select clock frequency when asynchronous serial clock is selected in SPI2
0: 75.0 MHz
1: 96.0 MHz</td><td>R/W</td></tr>
<tr><td>27</td><td>SCI0ASYNCSEL</td><td>Select clock frequency when asynchronous serial clock is selected in SCI0
0: 75.0 MHz
1: 96.0 MHz</td><td>R/W</td></tr>
</table>

Bit	Symbol	Function	R/W
28	SCI1ASYNCSEL	Select clock frequency when asynchronous serial clock is selected in SCI1 0: 75.0 MHz 1: 96.0 MHz	R/W
29	SCI2ASYNCSEL	Select clock frequency when asynchronous serial clock is selected in SCI2 0: 75.0 MHz 1: 96.0 MHz	R/W
30	SCI3ASYNCSEL	Select clock frequency when asynchronous serial clock is selected in SCI3 0: 75.0 MHz 1: 96.0 MHz	R/W
31	SCI4ASYNCSEL	Select clock frequency when asynchronous serial clock is selected in SCI4 0: 75.0 MHz 1: 96.0 MHz	R/W

The SCKCR register is used to select the frequency and source for each of the xSPI interface clock (XSPI_CLKn, n = 0, 1), external bus clock (CKIO), CANFD clock (PCLKCAN), Ethernet PHY reference clock (ETHn_REFCLK, n = 0 to 2), and alternative clock when the main clock abnormal oscillation is detected in CLMA3.

6.2.2 SCKCR2 : System Clock Control Register 2

Base address: SYSC_S = 0x8128_0000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	SCI5ASYNCSEL	SPI3ASYNCSEL	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	DIVSELSUB	—	—	—	FSELCPU0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

Bit	Symbol	Function	R/W
1:0	FSELCPU0[1:0]	Set the frequency of the clock provided to Coretex-R52 CPU0 in combination with bit 5 (DIVSELSUB). The combination is shown below. Both DIVSELSUB and FSELCPU0 can be changed at the same time. Be careful that the set frequency is output after the intermediate frequency is output.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	—	This bit is read as 1. The write value should be 1.	R/W

FSELCPU0[1:0]	DIVSELSUB = 0	DIVSELSUB = 1
00b	800.0 MHz	600.0 MHz
01b	400.0 MHz	300.0 MHz
10b	200.0 MHz	150.0 MHz
11b	Setting prohibited	

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>5</td><td>DIVSELSUB</td><td>Select the base clock frequency for peripheral module.
<table>
<tr><th>Clock name</th><th>DIVSELSUB = 0</th><th>DIVSELSUB = 1</th></tr>
<tr><td>ICLK</td><td>200.0 MHz</td><td>150.0 MHz</td></tr>
<tr><td>PCLKH</td><td>200.0 MHz</td><td>150.0 MHz</td></tr>
<tr><td>PCLKM</td><td>100.0 MHz</td><td>75.0 MHz</td></tr>
<tr><td>PCLKL</td><td>50.0 MHz</td><td>37.5 MHz</td></tr>
<tr><td>PCLKADC</td><td>25.0 MHz</td><td>18.75 MHz</td></tr>
<tr><td>PCLKGPTL</td><td>400.0 MHz</td><td>300.0 MHz</td></tr>
</table></td><td>R/W</td></tr>
<tr><td>23:6</td><td>—</td><td>These bits are read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>24</td><td>SPI3ASYNCSEL</td><td>Select clock frequency when asynchronous serial clock is selected in SPI3
0: 75.0 MHz
1: 96.0 MHz</td><td>R/W</td></tr>
<tr><td>25</td><td>SCI5ASYNCSEL</td><td>Select clock frequency when asynchronous serial clock is selected in SCI5
0: 75.0 MHz
1: 96.0 MHz</td><td>R/W</td></tr>
<tr><td>31:26</td><td>—</td><td>These bits are read as 0. The write value should be 0.</td><td>R/W</td></tr>
</table>

The SCKCR2 register is used to select the frequency of the clock supplied to the CPU0.

6.2.3 PLL0MON : PLL0 Monitor Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL0MON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>0</td><td>PLL0MON</td><td>PLL0 Lock State Monitor
This bit monitors whether the PLL0 is locked or not.
0: PLL0 is not locked
1: PLL0 is locked</td><td>R</td></tr>
<tr><td>31:1</td><td>—</td><td>These bits are read as 0.</td><td>R</td></tr>
</table>

The PLL0MON register is used to monitor the locking state of PLL0.

6.2.4 PLL1MON : PLL1 Monitor Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL1MON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*1

Bit	Symbol	Function	R/W
0	PLL1MON	PLL1 Lock State Monitor This bit monitors whether the PLL1 is locked or not. 0: PLL1 is not locked 1: PLL1 is locked	R
31:1	—	These bits are read as 0.	R

Note 1. Depends on package. QFP package: 0, BGA package: 1.

The PLL1MON register is used to monitor the locking state of PLL1.

6.2.5 PLL1EN : PLL1 Enable Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x50

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PLL1EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	PLL1EN	PLL1 Enable 0: PLL1 is in standby state 1: PLL1 is in normal state	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The PLL1EN register is used to set PLL1 to the standby mode. If PLL1 is not used, set this register to 0.

6.2.6 LOCOCR : Low-Speed On-Chip Oscillator Control Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x70

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LCSTP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LCSTP	LOCO Stop Specify whether to run or stop the low-speed on-chip oscillator (LOCO). 0: Stop 1: Run	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: 1. When setting the LOCOCR to make the LOCO run again after it has been stopped, start using the LOCO clock after the LOCO oscillation stabilization time (t_{LOCOWT}) has elapsed.

2. Processing to stop oscillation after the bit has been set to stop the LOCO also requires a certain amount of time. Therefore, the following restrictions apply to making the LOCO run and stopping it:
   - When setting the bit to make the LOCO run again after it has been stopped, ensure that the period over which the LOCO has been stopped is at least the LOCO oscillation stabilization time (t_{LOCOWT}).
   - To set the bit to stop the LOCO, ensure that the LOCO oscillation is stable at the time.

3. LOCO may generate glitch when it is stopped after operation. Therefore, stop CLMA2 and CLMA3 which use LOCO clock, and transit to the module stop state by MSTPCRG register before LOCO is stopped.

The LOCOCR register is used to control operation of a low-speed on-chip oscillator (LOCO).

6.2.7 HIZCTRLEN : High-Impedance Control Enable Register

Base address: SYSC_S = 0x8128_0000

Offset address: 0x80

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	CLMA1MASK	CLMA0MASK	CLMA3MASK
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	CLMA3MASK	CLMA3 error mask to POE3 and POEG 0: CLMA3 error is not transferred to POE3 and POEG. 1: CLMA3 error is transferred to POE3 and POEG.	R/W
1	CLMA0MASK	CLMA0 error mask to POE3 and POEG 0: CLMA0 error is not transferred to POE3 and POEG. 1: CLMA0 error is transferred to POE3 and POEG.	R/W
2	CLMA1MASK	CLMA1 error mask to POE3 and POEG 0: CLMA1 error is not transferred to POE3 and POEG. 1: CLMA1 error is transferred to POE3 and POEG.	R/W

Bit	Symbol	Function	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

The HIZCTRLEN register is used to transfer CLMA error signals to POE3 and POEG.

6.3 Selecting Input to Main Clock Oscillator

The clock signal is supplied to the main clock oscillator by connecting a resonator or by the input of an external clock signal to EXTCLKIN. The clock input mode is set based on the status of the XTAL, EXTAL, and EXTCLKIN pins. For details about the settings, see Table 6.4.

Table 6.4 Clock input mode selection

Clock input mode	XTAL	EXTAL	EXTCLKIN
Resonator connection	Resonator connected	Resonator connected	VSS via a resistor
External clock input	Open	VSS via a resistor	Clock input

6.3.1 Connecting a Crystal Resonator

Figure 6.2 shows an example of connecting a crystal resonator. Make sure to connect EXTCLKIN to VSS via a resistor.

A damping resistor Rd should be added, if necessary. Because the resistor values vary depending on the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer.

When connecting a resonator to supply the clock, the frequency of the resonator should be in the frequency of the resonator for the main clock oscillator described in Table 6.5.

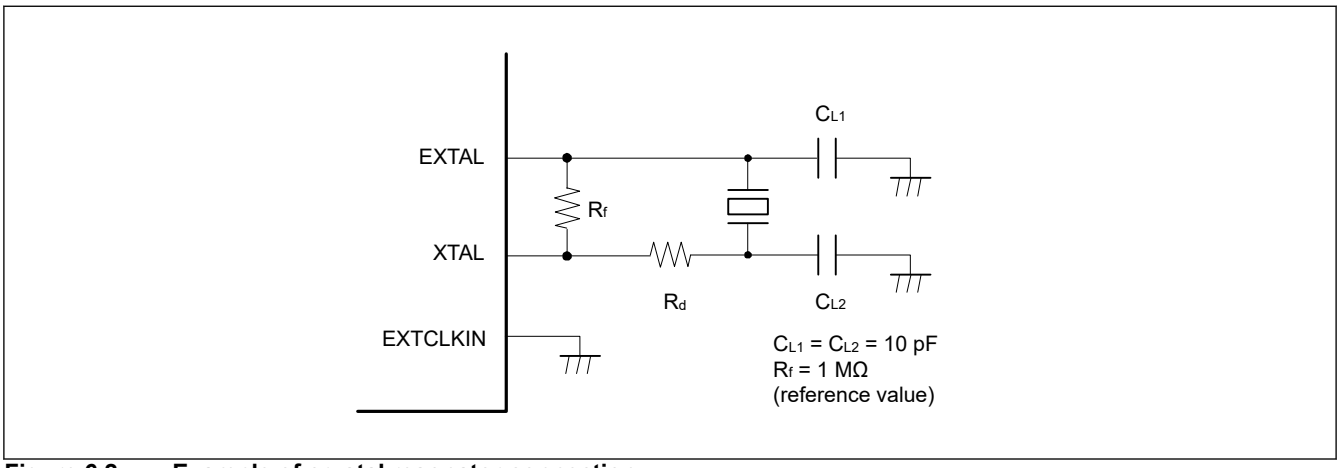


Figure 6.2 Example of crystal resonator connection

Table 6.5 Damping resistance (reference values)

Frequency (MHz)	25
Rd (Ω)	0

Figure 6.3 shows an equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in Table 6.6.

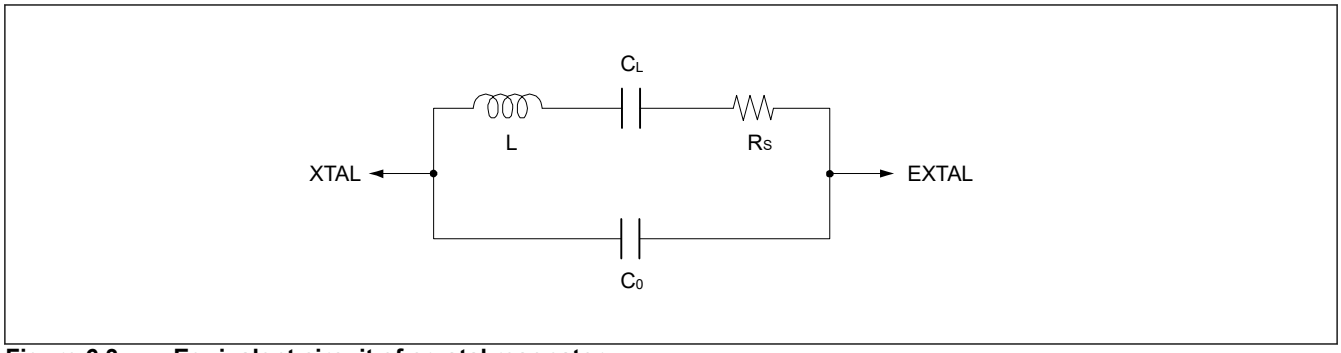


Figure 6.3 Equivalent circuit of crystal resonator

Table 6.6 Crystal resonator characteristics (reference values)

Frequency (MHz)	25
RS_{max} (Ω)	150 Ω

6.3.2 External Clock Input

Figure 6.4 shows an example connection of external clock input. Connect EXTAL to VSS via a resistor and leave XTAL open.

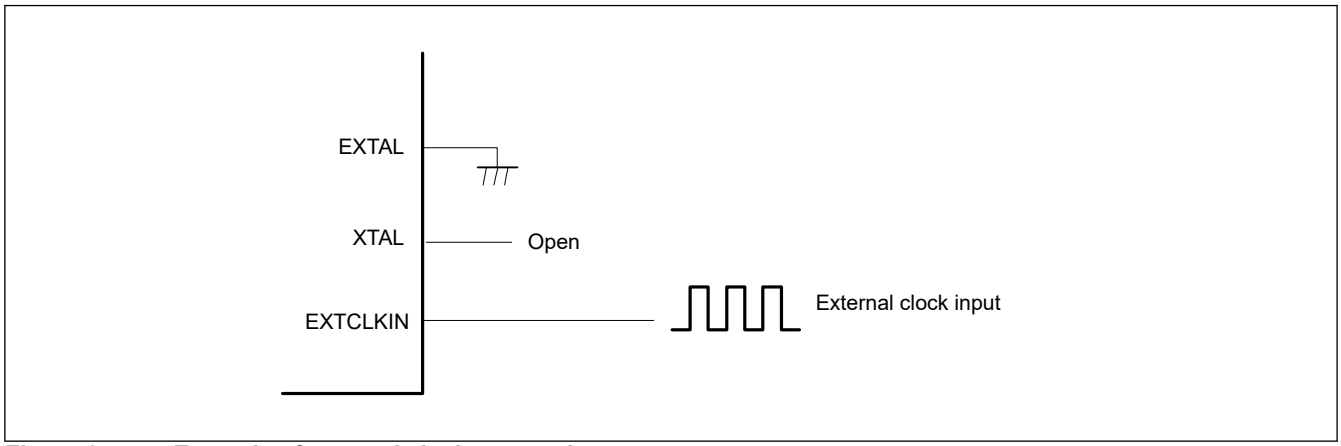


Figure 6.4 Example of external clock connection

6.4 PLL Circuit

The PLL circuit has a function to multiply the frequency from the oscillator. The LSI incorporates two PLL circuits (PLL0 and PLL1).

6.5 Low-Speed On-Chip Oscillator (LOCO)

The LSI incorporates low-speed on-chip oscillator (LOCO). It is used as sampling clock of CLMA and alternative clock of main clock and PLL clock when oscillation abnormality is detected.

After a reset is released, the LOCO is disabled. To enable this function, set the LOCO stop bit in the Low-Speed On-Chip Oscillator Control Register (LOCOCR.LCSTP) to 1.

6.6 Oscillation Abnormality Detection Function for the Main Clock

6.6.1 Main Clock Oscillation Abnormality Detection and Operation after Detection

The LSI incorporates a clock monitor circuit (CLMA3) that enables the detection of abnormal oscillation by the main clock. After a reset is released, the oscillation abnormality detection function is disabled. To enable this function, set the clock monitor enable bit in CLMA Control Register 0 (CLMA3.CTL0.CLME) to 1. For details, see section 7, Clock Monitor Circuit (CLMA).

An oscillation abnormality interrupt can be generated when an oscillation abnormality is detected. Low-speed on-chip oscillator (LOCO) clock or PLL free run clock is supplied as the system clock source instead of the main clock, PLL0 clock, and PLL1 clock. LOCO clock or PLL free run clock is selectable by the select alternative clock bit in the System Clock Control Register (SCKCR.CLMASEL). When PLL free run clock is selected, clock frequency is not guaranteed.

In addition, the MTU3 and GPT output can be forcedly driven to the high-impedance on the detection. For details, see:

- section 17, Multi-Function Timer Pulse Unit 3 (MTU3)
- section 18, Port Output Enable 3 (POE3)
- section 19, General PWM Timer (GPT)
- section 20, Port Output Enable for GPT (POEG)

Note: If CLMA3 detects abnormal oscillation, CLMA0/CLMA1 also generate abnormal oscillation deteced interrupt when both CLMA3 and CLMA0/CLMA1 are enabled.

6.6.2 Main Clock Oscillation Abnormality Detection Interrupt

When abnormal oscillation is detected, an abnormal oscillation detected interrupt is generated by the CLMA3 circuit, and it is transfered to the Port Output Enable 3 (POE3) block and Port Output Enable for GPT (POEG) if transmission is enabled in HIZCTRLEN register. On accepting the notification that the oscillation is to be stopped, the POE3 sets the OSTST high-impedance flag in the Input Level Control/Status Register 6 (ICSR6.OSTSTF) to 1 and the POEG also sets the Oscillation Stop Detection flags (POEGGn.OSTPF (n = A to D)) in the POEG Group n Setting Registers to 1.

The abnormal oscillation detected interrupt is connected to the Interrupt Controller Unit (ICU) as one of the error events of peripheral error interrupt so that error interrupt output and error reset are usable. When error interrupt output is used, the interrupt by all error events is disabled in the initial state after release from the reset state. To use oscillation stop detection interrupt (error event), release mask of the event in PRIERR_RSTMSK0 register when the ICU detects an error event. For details, see section 11, Interrupt Controller (ICU).

Note: If an abnormality in oscillation of the main clock occurs, oscillation of the PLL0 clock source also becomes abnormal, which may result in incorrect operation of the CPU. The POE3 and POEG set the high-impedance flag regardless of the state of the CPU, and the MTU3 and GPT outputs are driven to the high-impedance state.

6.7 PLL Oscillation Abnormality Detection Function

The PLL oscillation abnormality detection function detects an abnormality in oscillation of the PLL0 or PLL1 by using the Clock Monitor Circuits (CLMA0 and CLMA1) to monitor the frequency. It then supplies the main clock instead of the PLL0 or PLL1 clock. For details about the CLMA, see section 7, Clock Monitor Circuit (CLMA).

When abnormal oscillation is detected, an abnormal oscillation detected interrupt is generated by the CLMA0 or CLMA1 circuit, and it is transferred to POE3 and POEG if transmission is enabled in HIZCTRLEN register. On accepting the notification that the oscillation is to be stopped, the POE3 sets the OSTST high-impedance flag in the Input Level Control/Status Register 6 (ICSR6.OSTSTF) to 1 and the POEG also sets the Oscillation Stop Detection flags (POEGGn.OSTPF (n = A to D)) in the POEG Group n Setting Registers to 1.

The abnormal oscillation detected interrupt is connected to the Interrupt Controller Unit (ICU) as one of the error events of peripheral error interrupt so that error interrupt output and error reset are usable. When error interrupt output is used, the interrupt by all error events is disabled in the initial state after release from the reset state. To use oscillation stop detection interrupt (error event), release mask of the event in PRIERR_RSTMSK0 register when the ICU detects an error event. For details, see section 11, Interrupt Controller (ICU) .

6.8 Low-Speed On-Chip Oscillator Oscillation Abnormality Detection Function

The clock monitor circuit (CLMA2) can be used to detect an abnormality in oscillation of the low-speed on-chip oscillator. For details about the CLMA, see section 7, Clock Monitor Circuit (CLMA).

6.9 Internal Clock

Clock sources of internal clock signals are the main clock, LOCO clock, PLL0 clock, PLL1 clock, and the external clock for JTAG. Multiple internal clocks listed in Table 6.7 are produced from these sources.

Table 6.7 Overview of Clock Generation Circuit Specifications (Internal Clock) (1 of 2)

Clock name	Description
CPU0 clock (CPU0CLK)	The CPU0 clock (CPU0CLK) is used as the operating clock of the Cortex-R52 CPU0. The CPU0 operating frequency is specified by combination of the SCKCR2.FSELCPU0 and DIVSELSUB bits. Frequency of CPU0 is selectable from ×1, ×2, or ×4 of ICLK frequency.
System clock (ICLK)	The system clock (ICLK) is a clock supplied to, for example DMAC, interrupt controller, CPU bus interface, BSC. The ICLK is specified by the SCKCR2.DIVSELSUB bit.
Peripheral module clock H (PCLKH)	The peripheral module clock H (PCLKH) is used as the operating clock and/or bus clock for high-speed peripheral modules such as MTU3, GPT connected to LLPP bus, TFU, DSMIF, POE3, POEG (LLPP), ADC12 (bus), GMAC, xSPI (bus), SHOSTIF, and System SRAM. The PCLKH frequency is specified by the SCKCR2.DIVSELSUB bit. Frequency of PCLKH is the same as ICLK.
Peripheral module clock M (PCLKM)	The peripheral module clock M (PCLKM) is used as the operating clock and/or bus clock for middle-speed peripheral modules such as CRC, SCI (bus), SPI (bus), GPT connected to non-safety and safety regions, CANFD, GMAC (bus), SHIP, OTP, Network Reg, ESC (bus), USB (bus), GPIO, and CLMA. The PCLKM frequency is specified by the SCKCR2.DIVSELSUB bit. Frequency of PCLKM is the same as 1/2 of ICLK.
Peripheral module clock L (PCLKL)	The peripheral module clock L (PCLKL) is used as the operating clock and/or bus clock for low-speed peripheral modules such as POEG (Non Safety, Safety), IIC, DOC, CMT, CMTW, WDT, RTC (bus), and TSU. The PCLKL frequency is specified by the SCKCR2.DIVSELSUB bit. Frequency of PCLKL is the same as 1/4 of ICLK.
Peripheral module clock for ADC12 (PCLKADC)	The peripheral module clock for ADC12 (PCLKADC) is used as the operating clock for ADC12 unit 0 and 1. The PCLKADC frequency is specified by the SCKCR2.DIVSELSUB bit. Frequency of PCLKADC is the same as 1/8 of ICLK.
Peripheral module clock for GPT in LLPP (PCLKGPTL)	The peripheral module clock for GPT in LLPP (PCLKGPTL) is used as the operational clock for GPT in LLPP. The PCLKGPTL frequency is specified by the SCKCR2.DIVSELSUB bit. Frequency of PCLKGPTL is the same as ×2 of ICLK.
Peripheral module clock for SCIn (PCLKSCIn) (n = 0 to 5)	The peripheral module clock for SCIn (PCLKSCIn) is used as the operating clock for SCIn. The PCLKSCIn frequency is specified by combination of the SCKCR.SCInASYNCSEL (n = 0 to 4) or SCKCR2.SCInASYNCSEL (n = 5) bit and the SCIn.CCR3.BPEN bit. SCIn.CCR3.BPEN bit specifies which clock is used, synchronous clock (1) or asynchronous clock (0). When the bit is set to 1 (synchronous clock), the frequency of PCLKSCIn is the same as PCLKM. When the bit is set to 0 (asynchronous clock), the frequency of PCKLSCIn is the clock frequency selected by SCKCR.SCInASYNCSEL (n = 0 to 4) or SCKCR2.SCInASYNCSEL (n = 5) bit. It is selectable from 75 MHz or 96 MHz. Setting is independent for each channel.
Peripheral module clock for SPIn (PCLKSPIn) (n = 0 to 3)	The peripheral module clock for SPIn (PCLKSPIn) is used as the operating clock for SPIn. The PCLKSPIn frequency is specified by combination of the SCKCR.SPInASYNCSEL (n = 0 to 2) or SCKCR2.SPInASYNCSEL (n = 3) bit and the SPIn.SPCR.BPEN bit. SPIn.SPCR.BPEN bit specifies which clock is used, synchronous clock (1) or asynchronous clock (0). When the bit is set to 1 (synchronous clock), the frequency of PCLKSPIn is the same as PCLKM. When the bit is set to 0 (asynchronous clock), the frequency of PCKLSPIn is the clock frequency selected by SCKCR.SPInASYNCSEL (n = 0 to 2) or SCKCR2.SPInASYNCSEL (n = 3) bit. It is selectable from 75 MHz or 96 MHz. Setting is independent for each channel.
Peripheral module clock for CANFD (PCLKCAN)	The peripheral module clock for CANFD (PCLKCAN) is used as the serial clock for CANFD. The PCLKCAN frequency is specified by the SCKCR.FSELCANFD bit. It is selectable from 40 MHz or 80 MHz.
External bus clock (BSC_CLK, CKIO)	The external bus clock (CKIO) is used as an operating clock. The CKIO operating frequency is selectable from ICLK/2, ICLK/3, ICLK/4, ICLK/5, ICLK/6, ICLK/7, or ICLK/8 by combination of the SCKCR.CKIO and SCKCR2.DIVSELSUB bits.
Peripheral module clock for RTC (PCLKRTC)	The peripheral module clock for RTC (PCLKRTC) is used as the operational clock for RTC. The PCLKRTC frequency is fixed to 195.3 kHz, and cannot be user-specified.
USB clock (USB_CLK48)	The USB clock is used as the reference clock for USB PHY layer. The USB clock frequency is fixed to 48 MHz, and cannot be user-specified.
xSPIn serial clock (XSPI_CLKn) (n = 0, 1)	The xSPI serial clock for xSPIn (XSPI_CLKn) is used as the serial clock for xSPIn. Clock frequency is selectable by combination of the SCKCR.FSELXSPIn and DIVSELXSPIn bits.
Ethernet clock B (ETCLKB)	The Ethernet clock B (ETCLKB) is a clock for GMAC and MII converter. The ETCLKB frequency is fixed to 125 MHz, and cannot be user-specified.

Table 6.7 Overview of Clock Generation Circuit Specifications (Internal Clock) (2 of 2)

Clock name	Description
Ethernet clock C (ETCLKC)	The Ethernet clock C (ETCLKC) is a clock for ESC. The ETCLKC frequency is fixed to 100 MHz, and cannot be user-specified.
Ethernet clock D (ETCLKD)	The Ethernet clock D (ETCLKD) is a clock for MII Converter. The ETCLKC frequency is fixed to 50 MHz, and cannot be user-specified.
Ethernet clock E (ETCLKE)	The Ethernet clock E (ETCLKE) is a clock for ESC and MII converter. The ETCLKE frequency is fixed to 25 MHz, and cannot be user-specified.
CLMAn sampling clock (CLMAMCLKA) (n = 0, 1)	The CLMAn (n = 0, 1) sampling clocks (CLMAMCLKA) are for use with CLMA0 and CLMA1. The CLMAMCLKA frequency is fixed to 12.5 MHz, and cannot be user-specified.
CLMA2 sampling clock (CLMAMCLKB)	The CLMA2 sampling clock (CLMAMCLKB) is for use with CLMA2. The CLMAMCLKB frequency is fixed to 97.7 kHz, and cannot be user-specified.
CLMA3 sampling clock (CLMALCLKA)	The CLMA3 sampling clock (CLMALCLKA) is for use with CLMA3. The CLMALCLKA frequency is fixed to 240 kHz, and cannot be user-specified.
CLMA0 monitor clock (CLMAPLCLK0)	The CLMA0 monitor clock (CLMAPLCLK0) is the monitoring clock for use with CLMA0. The CLMAPLCLK0 frequency is fixed to 75 MHz, and cannot be user-specified.
CLMA1 monitor clock (CLMAPLCLK1)	The CLMA1 monitor clock (CLMAPLCLK1) is the monitoring clock for use with CLMA1. The CLMAPLCLK1 frequency is fixed to 62.5 MHz, and cannot be user-specified.
CLMA2 monitor clock (CLMALCLK)	The CLMA2 monitor clock (CLMAPLCLK) is the monitoring clock for use with CLMA2. The CLMALCLK frequency is fixed to 240 kHz, and cannot be user-specified.
CLMA3 monitor clock (CLMAMCLK)	The CLMA3 monitor clock (CLMAMCLK) is the monitoring clock for use with CLMA3. The CLMAMCLK frequency is fixed to 12.5 MHz, and cannot be user-specified.
JTAG clock (JTAGTCK)	The JTAG clock (JTAGTCK) is the operating clock for JTAG. JTAGTCK is generated by the external clock for JTAG (TCK).
Trace interface clock (TCLK)	The trace interface clock (TCLK) is used as the operating clock for the trace interface within CoreSight architecture. The TCLK frequency is fixed to 100 MHz, and cannot be user-specified. This clock frequency-divided by 2 is output from the LSI as the CPU trace clock for the on-chip debugger (TRACECLK).
Serial Host I/F clock (SHIFCLK)	The Serial Host I/F clock (SHIFCLK) is used as the interface clock for SHOSTIF. The SHIFCLK frequency is up to 40 MHz.
EnDat clock (PCLKENDAT)	The EnDat clock (PCLKENDAT) is used as the operating clock for EnDat. The PCLKENDAT is fixed to 100 MHz.
BiSS-C clock (PCLKBISS)	The BiSS-C clock (PCLKBISS) is used as the operating clock for BiSS-C. The PCLKBISS is fixed to 20 MHz.
HIPERFACE DSL clock (PCLKDSL)	The HIPERFACE DSL clock (PCLKDSL) is used as the operating clock for HIPERFACE. The PCLKDSL is fixed to 75 MHz.
A-format clock (PCLKAFMT)	The A-format clock (PCLKAFMT) is used as the operating clock for A-format. The PCLKAFMT is fixed to 80 MHz.
ENCOUT clock (PCLKENCO)	The ENCOUT clock (PCLKENCO) is used as the operating clock for ENCOUT. The PCLKENCO is fixed to 20 MHz.

6.10 Reference Clock to the External Ethernet PHY

The LSI provides reference clocks to the external Ethernet PHY.

Table 6.8 List of reference clock to the external Ethernet PHY

Clock name	Description
ETHn_REFCLK (n = 0 to 2)	25 MHz reference clock to the external Ethernet PHY. Clock source is selectable from main clock or frequency-dividing clock for PLL1 by the SCKCR.PHYSEL bit.
RMIIn_REFCLK (n = 0 to 2)	50 MHz reference clock to the external Ethernet PHY for RMII.

6.11 Usage Notes

6.11.1 Notes on Clock Generation Circuit

1. The operating frequency of xSPI interface, the external bus clock (CKIO), and CANFD serial clock that are supplied to the modules based on the SCKCR and SCKCR2 register settings varies before and after the frequency is changed. Do not change the clock frequency during access via the external interface. To start access via the external interface after the clock frequency is changed, confirm that the change to the frequency has been completed before you start access via the interface.
2. Do not change the clock frequency of the PCLKSPIn or PCLKSCIn while the SPIn (channels 0 to 3) or the SCIn (channels 0 to 5) is operating. To cancel the module-stop state after the clock frequency is changed, confirm that the change to the frequency is completed before you cancel the module-stop state.
3. In order to secure the processing after the clock frequency is changed, modify the pertinent clock control register to change the frequency, and then perform a dummy read of the register at least three times. Then perform the subsequent processing.
4. In order to secure processing after release from the module-stop state, dummy read the same register at least three times after writing to initiate release from the module-stop state, and only then proceed with the subsequent processing.
5. Do not change the base clock frequency for peripheral module by the DIVSELSUB bit in SCKSCR2 register during operation.
6. When controlling the module-stop state of the modules that initiate DMA transfer or that otherwise operate in relation with other blocks, confirm that DMAC transfer initiated by such modules and other points of interlinked operation are completed before releasing modules from or placing modules in the module-stop state. For set products in which such target modules are not used, design of the control of the module-stop state is on the assumption that the clocks are initially stopped. Therefore, the absence of unexpected operations in states other than the specified is not guaranteed. For example, operation after releasing modules from or placing modules in the module-stop state while a module is running is not guaranteed. For details on the operation and stopping of individual modules, see the section on processing by the corresponding block.
7. When CPU clock frequency is changed from 200 MHz to 800 MHz, change 400 MHz once and then change to 800 MHz.
   Example code:

```
volatile unsigned long dummy; // Declared volatile to prevent optimization being applied
R_SYSC_S->SCKCR2.FSELCPU0 = 01; // Step 1: Setting to frequency of CPU0 (400 MHz)
dummy = R_SYSC_S->SCKCR2.FSELCPU0; // Step 2: Dummy-read the SCKCR2 register.
dummy = R_SYSC_S->SCKCR2.FSELCPU0; // Step 3: Dummy-read the SCKCR2 register at least 3 times.
dummy = R_SYSC_S->SCKCR2.FSELCPU0;
R_SYSC_S->SCKCR2.FSELCPU0 = 00; // Step 4: Setting to frequency of CPU0 (800 MHz)
dummy = R_SYSC_S->SCKCR2.FSELCPU0; // Step 2: Dummy-read the SCKCR2 register.
dummy = R_SYSC_S->SCKCR2.FSELCPU0; // Step 3: Dummy-read the SCKCR2 register at least 3 times.
dummy = R_SYSC_S->SCKCR2.FSELCPU0;
```

6.11.2 Notes on Resonator

Since various resonator characteristics relate closely to the user's board design, adequate evaluation is required on the user side before use, referencing the resonator connection example shown in this section. The circuit constants for the resonator depend on the resonator to be used and the stray capacitance of the mounting circuit. Therefore, the circuit constants should be determined in full consultation with the resonator manufacturer. The voltage to be applied between the resonator pins must be within the absolute maximum rating.

6.11.3 Notes on Designing the Board

When using a crystal resonator, place the resonator and its load capacitors as close to the XTAL and EXTAL pins as possible. Other signal lines should be routed away from the oscillation circuit as shown in Figure 6.5. This prevents electromagnetic induction from interfering with correct oscillation.

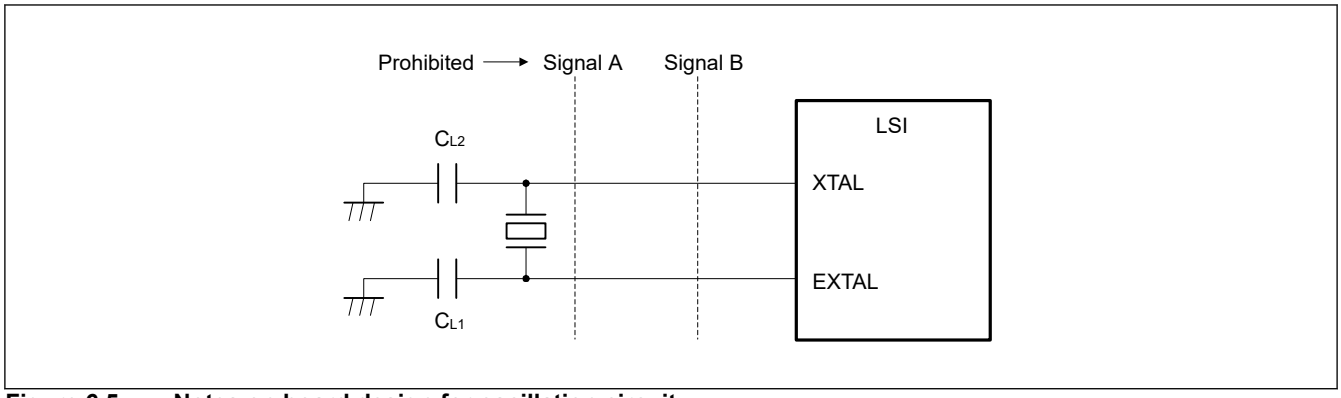


Figure 6.5 Notes on board design for oscillation circuit

7. Clock Monitor Circuit (CLMA)

7.1 Overview

A clock monitor circuit, CLMAm (m = 0 to 3), can detect a frequency error in PLL0 output, PLL1 output, low-speed on-chip oscillator (LOCO) output, or main clock. During 16 cycles of the sampling clock, it counts the rising edges of the monitoring clock (frequency-dividing clock from PLL0, PLL1, LOCO, or main clock), and then compares the counter value with the compare register.

When CLMAm detects an error, error events are generated and sent to the interrupt controller (ICU).

Table 7.1 lists the specifications of CLMAm and Figure 7.1 shows a block diagram.

Table 7.1 CLMAm (m = 0 to 3) specifications

Parameter	Description
Monitoring clock	The following monitoring clock frequency errors can be detected: ● PLL0 output clock divided by 32 (CLMAPLCLK0, which is to be supplied to CLMA0): 75 MHz ● PLL1 output clock divided by 16 (CLMAPLCLK1, which is to be supplied to CLMA1): 62.5 MHz ● Low-speed on-chip oscillator (LOCO) output clock (CLMALCLK, which is to be supplied to CLMA2): 240 kHz. ● Clock which equals to the main clock frequency divided by 2 (CLMAMCLK, which is to be supplied to CLMA3): 12.5 MHz
Sampling clock	The following clock frequency errors are monitored as the sampling clock: ● Clock which equals to the main clock frequency divided by 2 (CLMAMCLKA, which is to be supplied to CLMA0): 12.5 MHz ● Clock which equals to the main clock frequency divided by 2 (CLMAMCLKA, which is to be supplied to CLMA1): 12.5 MHz ● Clock which equals to the main clock frequency divided by 256 (CLMAMCLKB, which is to be supplied to CLMA2): 97.7 kHz. ● Clock which equals to LOCO frequency (CLMALCLKA, which is to be supplied to CLMA3): 240 kHz
Error event	When CLMAm detects a frequency error, error events are generated and sent to the interrupt controller (ICU). ● CLMA0 oscillator-stopped detection error ● CLMA1 oscillator-stopped detection error ● CLMA2 oscillator-stopped detection error ● CLMA3 oscillator-stopped detection error

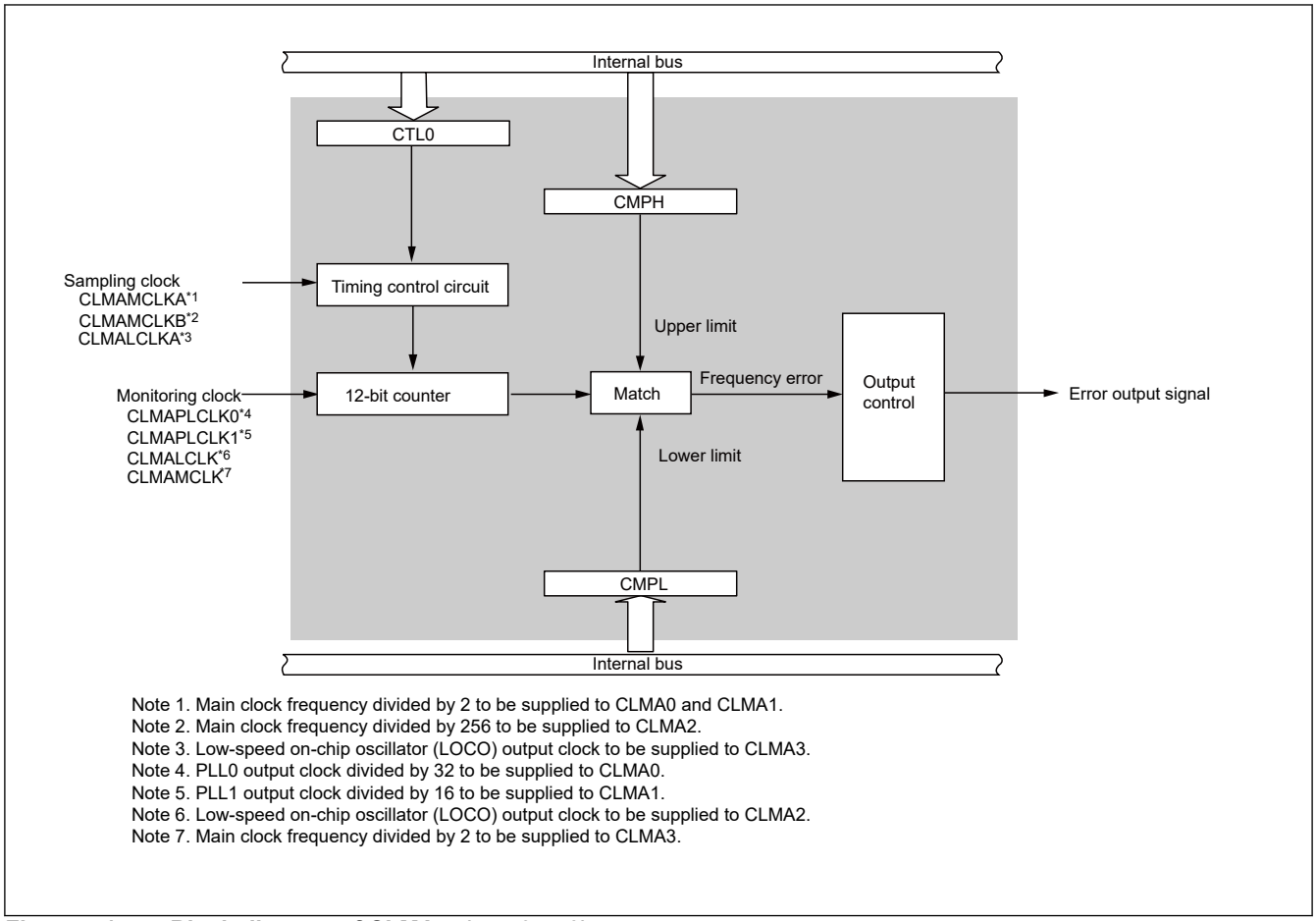


Figure 7.1 Block diagram of CLMAm (m = 0 to 3)

7.2 Register Descriptions

7.2.1 CTL0 : CLMA Control Register 0

Base address: CLMAm = 0x8128_0800 + 0x0020 × m (m = 0 to 3)

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CLME
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLME	Clock Monitor m Enable (m = 0 to 3) Enables or disables operation of clock monitor circuit CLMAm. 0: Disables CLMAm operation 1: Enables CLMAm operation*1	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Once the CLME bit is set to 1, it cannot be cleared by any operation other than reset (except CPU0 software reset).

The CTL0 registers control operation of clock monitor circuit CLMAm. Writing to these registers is protected by a specific command sequence. For details, see (1)Enabling operations.

7.2.2 CMPL : CLMA Compare Register L

Base address: CLMAm = 0x8128_0800 + 0x0020 × m (m = 0 to 3)

Offset address: 0x08

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CMPL[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
11:0	CMPL[11:0]	Clock Monitor m Compare L (m = 0 to 3) Specify the lower-limit threshold for the frequency domain.*1 ● For details, see (2)Method of calculating threshold values, the CMPL and CMPH registers. Recommended value: $f_{CLMATMON}$ (min) / $f_{CLMATSMP}$ (max) × 16 - 1 $f_{CLMATMON}$: Monitoring clock frequency $f_{CLMATSMP}$: Sampling clock frequency ● Minimum value: 0x0001	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. To set the CMPL registers, the following conditions must be met:
1. 1 ≤ CMPL
2. CMPL + 3 ≤ CMPH

The CMPL registers set the lower limit for comparing frequency domains. Values can be written to these registers when CTL0.CLME = 0. When CTL0.CLME = 1, writing to these registers has no effect.

7.2.3 CMPH : CLMA Compare Register H

Base address: CLMAm = 0x8128_0800 + 0x0020 × m (m = 0 to 3)

Offset address: 0x0C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CMPH[11:0]											
Value after reset:	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
11:0	CMPH[11:0]	Clock Monitor m Compare H (m = 0 to 3) Specify the upper-limit threshold for the frequency domain.*1 ● For details, see (2)Method of calculating threshold values, the CMPL and CMPH registers. Recommended value: $f_{CLMATMON}$ (max) / $f_{CLMATSMP}$ (min) × 16 + 1 $f_{CLMATMON}$: Monitoring clock frequency $f_{CLMATSMP}$: Sampling clock frequency ● Minimum value: CMPL + 0x0003	R/W
15:12	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. To set the CMPH registers, the following conditions must be met:
- 1 ≤ CMPL
- CMPL + 3 ≤ CMPH

The CMPH registers set the upper limit for comparing frequency domains. Values can be written to these registers when CTL0.CLME = 0. When CTL0.CLME = 1, writing to these registers has no effect.

7.2.4 PCMD : CLMA Command Register

Base address: CLMAm = 0x8128_0800 + 0x0020 × m (m = 0 to 3)

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:								
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	n/a	CLMAm Protect Key Code (m = 0 to 3) Write a specific command sequence.	W

The PCMD registers control writing to the protected registers.

For details, see (1)Enabling operations.

7.2.5 PROTSR : CLMA Protection Status Register

Base address: CLMAm = 0x8128_0800 + 0x0020 × m (m = 0 to 3)

Offset address: 0x14

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	PRERR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRERR	CLMAm Error (m = 0 to 3) 0: No protection error occurred 1: A protection error occurred	R
7:1	—	These bits are read as 0.	R

The PROTSR registers indicate whether writing to the protected register is performed correctly. If writing is not performed correctly, a protection error occurs and the PROTSR.PRERR bit is set to 1.

For details, see (1)Enabling operations.

7.3 Operation

7.3.1 CLMAm Operation

(1) Enabling operations

Setting the CTL0.CLME bit to 1 starts monitoring the monitoring clock (PLL0 output divided by 32, PLL1 output divided by 16, LOCO output, and main clock divided by 2) by using clock monitor circuit CLMAm (m = 0 to 3) output. To write 1 in the CTL0.CLME bit, use the following command sequence:

1. Write 0xA5 to the PCMD register.
2. Writing to CTL0 is performed with the following sequence:
   (a) Write the target setting value (0x01).
   (b) Write the reversed value of the target (0xFE).
   (c) Write the target value (0x01) again.
3. Read CTL0.
   When the value of CTL0 shows 0x01, the operation of CLMAm is enabled. If it shows another value, check the value of PROTSR register.

When PROTSR = 0x01, the command sequence is not performed correctly. Execute the sequence again from step 1 to perform writing.

(2) Stopping operations

When a sampling clock stops due to register operation, the corresponding clock monitor circuit (CLMAm) is disabled automatically. After that, when the monitoring clock starts oscillating again and stabilizes, CLMAm resumes operation.

Table 7.2 Monitoring clock and sampling clock each CLMAm

Parameter		CLMA0	CLMA1	CLMA2	CLMA3
Detection target	Clock	PLL0 output clock	PLL1 output clock	LOCO	Main clock
	Frequency	2400 MHz	1000 MHz	240 kHz	25 MHz
Monitoring clock	Clock	PLL0 output clock divided by 32	PLL1 output clock divided by 16	LOCO	Main clock divided by 2
	Frequency	75.0 MHz	62.5 MHz	240 kHz	12.5 MHz
Sampling clock	Clock	Main clock divided by 2	Main clock divided by 2	Main clock divided by 256	LOCO
	Frequency	12.5 MHz	12.5 MHz	97.7 kHz	240 kHz

7.3.2 Detecting Error Clock Frequency

(1) Detection Method

1. During 16 cycles of the sampling clock (main clock frequency dividing clock and LOCO), CLMAm counts the rising edge of the monitoring clock (PLL0 output divided by 32, PLL1 output divided by 16, LOCO output, and main clock divided by 2), and compares the counter value with the set threshold (m = 0 to 3).
   - CMPL.CMPL[11:0] set the lower-limit threshold of the frequency domain.
   - CMPH.CMPH[11:0] set the upper-limit threshold of the frequency domain.
2. If the monitoring clock stops or shows a value lower than the expected frequency, the counter value shows a value lower than the setting of CMPL.CMPL[11:0].
3. If the monitoring clock shows a value higher than the expected frequency, the counter value shows a value higher than the setting of CMPH.CMPH[11:0].

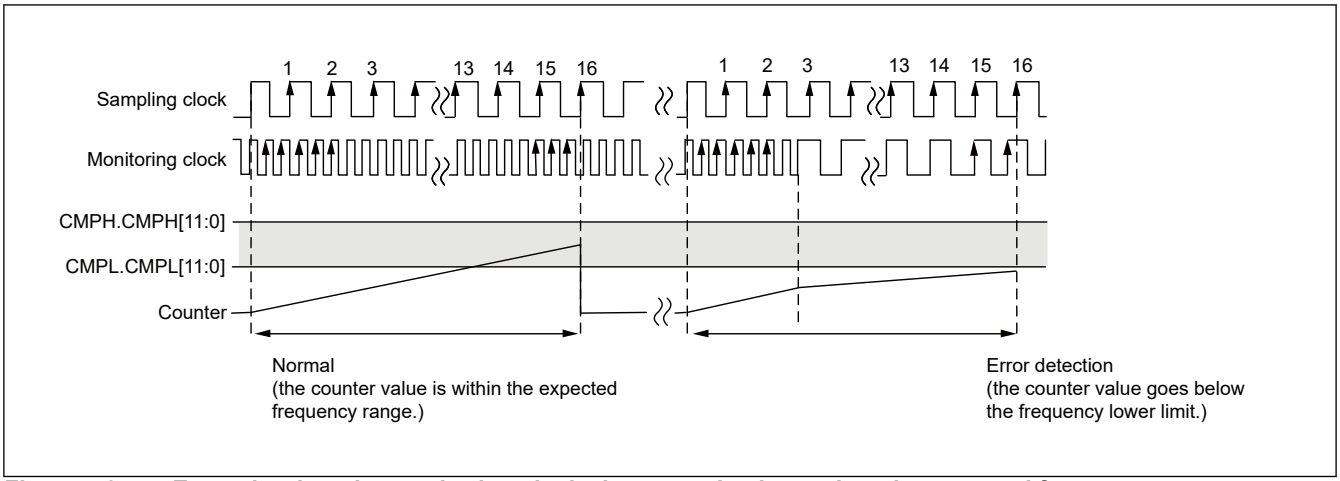


Figure 7.2 Example when the monitoring clock shows a value lower than the expected frequency

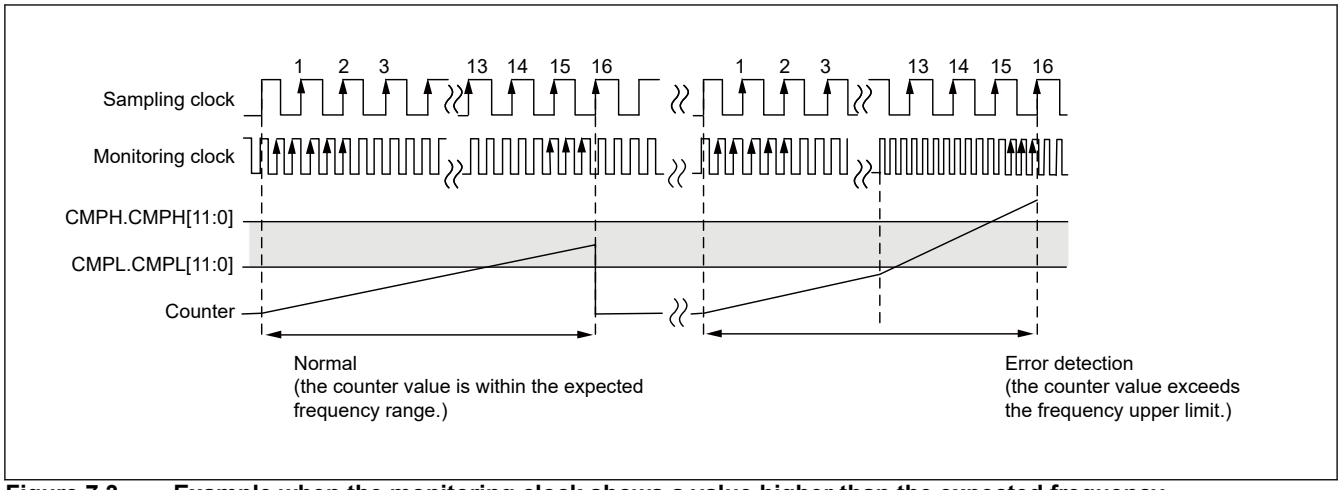


Figure 7.3 Example when the monitoring clock shows a value higher than the expected frequency

Note: No error is detected if the monitoring clock frequency changes within a sampling interval, and it eventually stays within a valid counter value. A monitoring clock error is detected after a sampling interval (16 cycles of the sampling clock).

(2) Method of calculating threshold values, the CMPL and CMPH registers

For compare registers CMPL and CMPH, set the minimum and maximum values of the number of cycles (the number of rising edges) for the monitoring clock which is assumed to be normal within 16 cycles of the sampling clock (main clock frequency dividing clock and LOCO).

$f_{CLMATSMP}$ indicates the sampling clock frequency, $f_{CLMATMON}$ indicates the monitoring clock frequency, and N indicates the number of cycles (rising edges) of the monitoring clock, which is expected within 16 cycles of the sampling clock.

$$N = \frac{f_{\mathrm{CLMATMON}}}{f_{\mathrm{CLMATSMP}}} \times 16$$

Consider the allowable frequency deviations for the monitoring clock and the sampling clock, use the following formula to calculate the threshold:

$$\text{Lower-limit threshold} = N_{\min} = \frac{f_{\mathrm{CLMATMON(min)}}}{f_{\mathrm{CLMATSMP(max)}}} \times 16 - 1$$

$$\text{Upper-limit threshold} = N_{\max} = \frac{f_{\mathrm{CLMATMON(max)}}}{f_{\mathrm{CLMATSMP(min)}}} \times 16 + 1$$

Note: Set the threshold within the following range:
CMPL ≥ 0x0001
CMPH ≥ CMPL + 0x0003

(3) Example: For CLMA0

For example, when the sampling clock equals to the main clock frequency divided by 2, $f_{CLMATSMP}$ = 12.5 MHz (±5%), or the monitoring clock equals to PLL0 output divided by 32, $f_{CLMATMON}$ = 75 MHz (±5%), the recommended threshold is calculated as follows:

$$N_{\min} = \frac{f_{\mathrm{CLMATMON(min)}}}{f_{\mathrm{CLMATSMP(max)}}} = \frac{71.25}{13.125} \times 16 - 1 = 85.86$$

$$\mathrm{CMPL} = 86 = 0\mathrm{x}0056$$

$$N_{\max} = \frac{f_{\mathrm{CLMATMON(max)}}}{f_{\mathrm{CLMATSMP(min)}}} = \frac{78.75}{11.875} \times 16 + 1 = 107.11$$

$$\mathrm{CMPH} = 107 = 0\mathrm{x}006\mathrm{B}$$

7.3.3 Detecting Error Clock Frequency

When the monitoring clock frequency (PLL0 output divided by 32, PLL1 output divided by 16, LOCO output, and main clock divided by 2) is higher than the upper-limit threshold or lower than the lower-limit threshold, any of the following signals is sent to the interrupt controller (ICU) as error event:

- CLMA0 oscillator-stopped detection error
- CLMA1 oscillator-stopped detection error
- CLMA2 oscillator-stopped detection error
- CLMA3 oscillator-stopped detection error

Error signal can be cleared by RES# pin reset, System software reset, or Error reset only.

7.4 Notes on Using CLMAm

- Do not use a clock for which CLMAm detected an error. If you use the clock, operation of the device is not guaranteed.
- No multiple clocks but only single monitoring clock error is assumed at the same time.
- Abnormality of the main clock oscillator may cause not only CLMA3 error but also other CLMA errors because it is also used as sampling clock. Abnormality of the LOCO may cause not only CMA2 but also other CLMA errors.
- In case of higher frequency than expectation, the product may not work correctly due to outside of the operation range.
- Glitch may be generated when switching to the alternative clock in case main oscillator clock frequency is lower than one-eight of the LOCO frequency.
- The clock is switched to the alternative clock when the clock abnormality is detected as below table.

CLMAm error				**Alternative clock**			
CLMA3 (MOSC) error	**CLMA2 (LOCO) error**	**CLMA0 (PLL0) error**	**CLMA1 (PLL1) error**	**MOSC clock**	**PLL0 output clock**	**PLL1 output clock**	**Description**
0	0	0	0	MOSC	PLL0	PLL1	Normal operation
1	0	X	X	LOCO	LOCO or PLL0	LOCO or PLL1	MOSC error*1 *2 *3
0	1	0	0	MOSC	PLL0	PLL1	LOCO error
1	1	X	X	LOCO	LOCO or PLL0	LOCO or PLL1	Either MOSC or LOCO error*1 *2 *3
0	0	1	0	MOSC	MOSC	PLL1	PLL0 error
0	0	0	1	MOSC	PLL0	MOSC	PLL1 error
Other than the above				Indefinite			Indefinite

Note: MOSC is main clock oscillator. It is assumed that all CLMAm are enabled.
Note 1. Set LOCOCR.LCSTP to 1 when the clock is switched to LOCO in case of detecting abnormality since LOCO is stopped at the reset value. Be careful that if the reset is asserted after switching to the LOCO, this LSI may not restart because LOCO is disabled at the reset value in case clock abnormality originates from the hard error.
Note 2. PLL output clocks are selected by SCKCR.CLMASEL.
Note 3. When PLL is selected as alternative clock, PLL output clock is free run and the clock frequency is not guaranteed.

8. Low-Power Consumption Function

8.1 Overview

This LSI has several functions for reducing power consumption, including the Cortex®-R52 standby function and the module-stop function that stops functions independently for each peripheral module. Power consumption using clock control such as CKIO output control is also possible.

Table 8.1 lists the specifications of low-power consumption functions. Table 8.2 describes how to stop each peripheral module and exit the stop state.

Table 8.1 Specifications of low-power consumption function

Parameter	Specifications
Low-power consumption mode	Standby mode (Cortex-R52)
Module-stop function	Functions can be stopped independently for each peripheral module
CKIO output control function	CKIO clock output or stop (held at the low level) can be selected

Table 8.2 Stopping peripheral modules and exiting module-stop state (1 of 2)

Module	How to stop operation and exit the stop state	Initial state*1
Cortex-R52	Standby mode condition: Execution of the Wait For Interrupt (WFI) instruction Condition for release from the Standby mode: interrupt	Operating
Internal bus	Always operating	Operating
Tightly coupled memory (ATCM or BTCM)	Operate only during access	Operate only during access
System SRAM (SYSRAM)	Operate only during access	Operate only during access
Interrupt controller unit (ICU)	Operation is always enabled	Operating
Bus state controller (BSC)	Set the control register to enter or exit the module-stop state	Stop*2
Direct memory access controller (DMAC)	Operation is always enabled	Operating
General-purpose I/O ports	Operation is always enabled	Operating
Event link controller (ELC)	Operation is always enabled	Operating
Multi-function timer pulse unit 3 (MTU3)	Set the control register to enter or exit the module-stop state	Stop
General PWM timer (GPT)	Set the control register to enter or exit the module-stop state	Stop
Compare match timer (CMT)	Set the control register to enter or exit the module-stop state	Stop
Compare match timer W (CMTW)	Set the control register to enter or exit the module-stop state	Stop
Watchdog timer (WDT)	Operation is always enabled	Operating
Port output enable 3 (POE3)	Operation is always enabled	Operating
Port output enable for GPT (POEG)	Operation is always enabled	Operating
Real time clock (RTC)	Set the control register to enter or exit the module-stop state	Stop
Ethernet MAC (GMAC)	Set the control register to stop the module or exit the module-stop state	Stop
EtherCAT slave controller (ESC)	Set the control register to stop the module or exit the module-stop state	Stop
USB 2.0 HS host/function module	Set the control register to stop the module or exit the module-stop state	Stop*2
Serial communication interface (SCI)	Set the control register to enter or exit the module-stop state	Stop*2

Table 8.2 Stopping peripheral modules and exiting module-stop state (2 of 2)

Module	How to stop operation and exit the stop state	Initial state*1
I2C bus interface (IIC)	Set the control register to enter or exit the module-stop state	Stop
CANFD interface (CANFD)	Set the control register to enter or exit the module-stop state	Stop
Serial peripheral interface (SPI)	Set the control register to enter or exit the module-stop state	Stop
Expanded serial peripheral interface (xSPI)	Set the control register to enter or exit the module-stop state	Stop*2
ΔΣ interface (DSMIF)	Set the control register to enter or exit the module-stop state	Stop
Trigonometric function unit (TFU)	Set the control register to enter or exit the module-stop state	Stop
12-bit A/D converter (ADC12)	Set the control register to enter or exit the module-stop state	Stop
Temperature sensor unit (TSU)	Set the control register to enter or exit the module-stop state	Stop
CRC operation unit (CRC)	Set the control register to enter or exit the module-stop state	Stop
Clock monitor circuit (CLMA)	Set the control register to enter or exit the module-stop state	Stop
Data operation circuit (DOC)	Set the control register to enter or exit the module-stop state	Stop
Serial host interface (SHOSTIF)	Set the control register to enter or exit the module-stop state	Stop*2
EnDat 2.2 (ENDAT)	Set the control register to enter or exit the module-stop state	Stop
BiSS-C (BISS)	Set the control register to enter or exit the module-stop state	Stop
HIPERFACE DSL (HDSL)	Set the control register to enter or exit the module-stop state	Stop
A-format (AFMT)	Set the control register to enter or exit the module-stop state	Stop
ENCOUT	Set the control register to enter or exit the module-stop state	Stop
Debugging interface	Set the control register to stop the module or exit the module-stop state	Operating

Note 1. Each module is returned to the initial state by means of the RES# pin reset, error reset, or system software reset.
Note 2. The state of the module is Operating if its boot mode is selected.

8.2 Register Descriptions

The registers are applicable to the register write protection function. To write these registers, clear write protection for the target register by setting bit [1] of the Protect Register (PRCRN and PRCRS). For details, see section 10, Register Write Protection Function.

8.2.1 MSTPCRA : Module Stop Control Register A

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x0300

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	MSTPCRA12	MSTPCRA11	MSTPCRA10	MSTPCRA09	MSTPCRA08	—	—	MSTPCRA05	MSTPCRA04	—	—	—	MSTPCRA00
Value after reset:	0	0	0	1	1	1	1	1*1	0	0	1*1	1*1	0	0	0	1*1

Bit	Symbol	Function	R/W
0	MSTPCRA00	BSC Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
4	MSTPCRA04	xSPI Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
5	MSTPCRA05	xSPI Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	MSTPCRA08	SCI Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
9	MSTPCRA09	SCI Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
10	MSTPCRA10	SCI Unit 2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
11	MSTPCRA11	SCI Unit 3 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
12	MSTPCRA12	SCI Unit 4 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31:13	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The initial value changes depending on the operation mode setting.

The MSTPCRA register controls the module-stop state. This register is available in Non Safety region.

8.2.2 MSTPCRB : Module Stop Control Register B

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x0304

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	MSTPCRB06	MSTPCRB05	MSTPCRB04	—	—	MSTPCRB01	MSTPCRB00
Value after reset:	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1

Bit	Symbol	Function	R/W
0	MSTPCRB00	IIC Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
1	MSTPCRB01	IIC Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	MSTPCRB04	SPI Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W

Bit	Symbol	Function	R/W
5	MSTPCRB05	SPI Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
6	MSTPCRB06	SPI Unit 2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

The MSTPCRB register controls the module-stop state. This register is available in Non Safety region.

8.2.3 MSTPCRC : Module Stop Control Register C

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x308

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	MSTPCRC07	MSTPCRC06	MSTPCRC05	—	—	MSTPCRC02	MSTPCRC01	MSTPCRC00
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	1

Bit	Symbol	Function	R/W
0	MSTPCRC00	MTU3 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
1	MSTPCRC01	GPT Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
2	MSTPCRC02	GPT Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	MSTPCRC05	TFU Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
6	MSTPCRC06	ADC12 Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
7	MSTPCRC07	ADC12 Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

The MSTPCRC register controls the module-stop state. This register is available in Non Safety region.

8.2.4 MSTPCRD : Module Stop Control Register D

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x30C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	MSTPCRD11	MSTPCRD10	MSTPCRD09	MSTPCRD08	MSTPCRD07	MSTPCRD06	MSTPCRD05	MSTPCRD04	MSTPCRD03	MSTPCRD02	MSTPCRD01	MSTPCRD00
Value after reset:	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	MSTPCRD00	DSMIF Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
1	MSTPCRD01	DSMIF Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
2	MSTPCRD02	CMT Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
3	MSTPCRD03	CMT Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
4	MSTPCRD04	CMT Unit 2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
5	MSTPCRD05	CMTW Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
6	MSTPCRD06	CMTW Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
7	MSTPCRD07	TSU Module Stop 0: Release from the module-stop state 1: Transition to the module stop state is made	R/W
8	MSTPCRD08	DOC Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
9	MSTPCRD09	CRC Unit 0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
10	MSTPCRD10	CANFD Module Stop 0: Release from the module-stop state 1: Transition to the module stop state is made	R/W
11	MSTPCRD11	CKIO Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The MSTPCRD register controls the module-stop state. This register is available in Non Safety region.

8.2.5 MSTPCRE : Module Stop Control Register E

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x310

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	MSTPCRE08	—	—	—	—	MSTPCRE03	MSTPCRE02	—	MSTPCRE00
Value after reset:	0	0	0	0	0	0	0	1*1	0	0	0	0	1	1	0	1

Bit	Symbol	Function	R/W
0	MSTPCRE00	GMAC Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	MSTPCRE02	ESC Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
3	MSTPCRE03	Ethernet Subsystem Register Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	MSTPCRE08	USB Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The initial value changes depending on the operation mode setting.

The MSTPCRE register controls the module-stop state. This register is available in Non Safety region.

8.2.6 MSTPCRF : Module Stop Control Register F

Base address: SYSC_S = 0x8128_0000

Offset address: 0x314

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTPCRF00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MSTPCRF00	Trace Clock for Debugging Interface Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The MSTPCRF register controls the module-stop state. This register is available in Safety region.

8.2.7 MSTPCRG : Module Stop Control Register G

Base address: SYSC_S = 0x8128_0000

Offset address: 0x318

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	MSTPCRG11	MSTPCRG10	MSTPCRG09	MSTPCRG08	—	—	MSTPCRG05	MSTPCRG04	MSTPCRG03	MSTPCRG02	MSTPCRG01	MSTPCRG00
Value after reset:	0	0	0	0	1	1	1	1	0	0	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	MSTPCRG00	SCI Unit 5 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
1	MSTPCRG01	IIC Unit 2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
2	MSTPCRG02	SPI Unit 3 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
3	MSTPCRG03	GPT Unit 2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
4	MSTPCRG04	CRC Unit 1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
5	MSTPCRG05	RTC Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	MSTPCRG08	CLMA3 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
9	MSTPCRG09	CLMA0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
10	MSTPCRG10	CLMA1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
11	MSTPCRG11	CLMA2 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The MSTPCRG register controls the module-stop state. This register is available in Safety region.

8.2.8 MSTPCRI : Module Stop Control Register I

Base address: SYSC_S = 0x8128_0000

Offset address: 0x320

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTPCRI01	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1*1	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	MSTPCRI01	SHOSTIF Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made.	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The initial value changes depending on the operation mode setting.

The MSTPCRI register controls the module stop state of each IP. This register is available in Safety region.

8.2.9 MSTPCRJ : Module Stop Control Register J

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x324

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MSTPCRJ03	MSTPCRJ02	MSTPCRJ01	MSTPCRJ00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bit	Symbol	Function	R/W
0	MSTPCRJ00	AFMT0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
1	MSTPCRJ01	HDSL0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
2	MSTPCRJ02	BISS0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
3	MSTPCRJ03	ENDAT0 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

The MSTPCRJ register controls the module stop state of Encoder Interface unit 0. This register is available in Safety region.

8.2.10 MSTPCRK : Module Stop Control Register K

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x328

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MSTPCRK03	MSTPCRK02	MSTPCRK01	MSTPCRK00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bit	Symbol	Function	R/W
0	MSTPCRK00	AFMT1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
1	MSTPCRK01	HDSL1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
2	MSTPCRK02	BISS1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
3	MSTPCRK03	ENDAT1 Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

The MSTPCRK register controls the module stop state of Encoder Interface unit 1. This register is available in Safety region.

8.2.11 MSTPCRL : Module Stop Control Register L

Base address: SYSC_NS = 0x8028_0000

Offset address: 0x32C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTPCRL00
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	MSTPCRL00	ENCOUT Module Stop 0: Release from the module-stop state 1: Transition to the module-stop state is made	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The MSTPCRL register controls the module stop state of ENCOUT. This register is available in Safety region.

8.3 Operation

8.3.1 Module-Stop Function

The module-stop function can stop operation of each module of on-chip peripheral functions. When the MSTPCRmi bit (m = A to G and L, i = 31 to 00) in the MSTPCRA to MSTPCRG and MSTPCRL registers is set to 1, the specified module stops operating and enters the module-stop state. Clearing the MSTPCRmi bit (m = A to G and L, i = 31 to 00) to 0 leads to release from the module-stop state. Follow the procedure given below when releasing any of the peripheral modules listed in Table 8.3 from the module-stop state. This procedure is not required for peripheral modules not listed in Table 8.3. However, release from the module-stop state should be in accord with the procedure for initialization described in the section for the given peripheral module.

(1) Procedure

1. Clear the corresponding bit in the relevant Module Stop Control Register (MSTPCRm, m = A to G and L) then immediately dummy-read the MSTPCRm register once.
2. Dummy-read any register of the peripheral module which is being released from the module-stop state at least 5 times except RTC. For RTC, wait for 20 µs from release of module stop until clock is supplied. After that, all registers of that peripheral module can be accessible.

Note: The MPU having set the access control attribute for the peripheral I/O register region to 'Device nGnRnE' or 'Device nGnRE' is a prerequisite for the procedure.

(2) Example code

```
volatile unsigned long dummy;           // Declared volatile to prevent optimization being applied

R_SYSC_NS->MSTPCRD_b.MSTPCRD06 = 0;     // Setting to release CMTW unit 1 from the module-stop state
dummy = R_SYSC_NS->MSTPCRD;             // Step 1: Dummy-read the MSTPCRm register.

dummy = R_CMTW1->CMWCOR;                // Step 2: Dummy-read any register of CMTW unit 1 at least 5 times.
dummy = R_CMTW1->CMWCOR;
dummy = R_CMTW1->CMWCOR;
dummy = R_CMTW1->CMWCOR;
dummy = R_CMTW1->CMWCOR;
R_CMTW1->CMWIOR = 0x0001U;              // The first setting for CMTW unit 1 (value is an example)
```

For details about the initial state after a release from the reset state, see Table 8.2.

Note: Directly after a module is set to the module-stop state, writing to the control register of that module might still be possible.

Table 8.3 Peripheral functions requiring the above procedure for release from the module-stop state (1 of 3)

Peripheral functions	Corresponding Module-Stop Control Register
BSC	MSTPCRA register, MSTPCRA00 bit
xSPI Unit 0	MSTPCRA register, MSTPCRA04 bit
xSPI Unit 1	MSTPCRA register, MSTPCRA05 bit
SCI Unit 0	MSTPCRA register, MSTPCRA08 bit
SCI Unit 1	MSTPCRA register, MSTPCRA09 bit
SCI Unit 2	MSTPCRA register, MSTPCRA10 bit
SCI Unit 3	MSTPCRA register, MSTPCRA11 bit
SCI Unit 4	MSTPCRA register, MSTPCRA12 bit
IIC Unit 0	MSTPCRB register, MSTPCRB00 bit
IIC Unit 1	MSTPCRB register, MSTPCRB01 bit
SPI Unit 0	MSTPCRB register, MSTPCRB04 bit
SPI Unit 1	MSTPCRB register, MSTPCRB05 bit

Table 8.3 Peripheral functions requiring the above procedure for release from the module-stop state (2 of 3)

Peripheral functions	Corresponding Module-Stop Control Register
SPI Unit 2	MSTPCRB register, MSTPCRB06 bit
MTU3	MSTPCRC register, MSTPCRC00 bit
GPT Unit 0	MSTPCRC register, MSTPCRC01 bit
GPT Unit 1	MSTPCRC register, MSTPCRC02 bit
TFU	MSTPCRC register, MSTPCRC05 bit
ADC12 Unit 0	MSTPCRC register, MSTPCRC06 bit
ADC12 Unit 1	MSTPCRC register, MSTPCRC07 bit
DSMIF Unit 0	MSTPCRD register, MSTPCRD00 bit
DSMIF Unit 1	MSTPCRD register, MSTPCRD01 bit
CMT Unit 0	MSTPCRD register, MSTPCRD02 bit
CMT Unit 1	MSTPCRD register, MSTPCRD03 bit
CMT Unit 2	MSTPCRD register, MSTPCRD04 bit
CMTW Unit 0	MSTPCRD register, MSTPCRD05 bit
CMTW Unit 1	MSTPCRD register, MSTPCRD06 bit
TSU	MSTPCRD register, MSTPCRD07 bit
DOC	MSTPCRD register, MSTPCRD08 bit
CRC Unit 0	MSTPCRD register, MSTPCRD09 bit
CANFD	MSTPCRD register, MSTPCRD10 bit
CKIO	MSTPCRD register, MSTPCRD11 bit
GMAC	MSTPCRE register, MSTPCRE00 bit
ESC	MSTPCRE register, MSTPCRE02 bit
Ethernet Subsystem register	MSTPCRE register, MSTPCRE03 bit
USB	MSTPCRE register, MSTPCRE08 bit
SCI Unit 5	MSTPCRG register, MSTPCRG00 bit
IIC Unit 2	MSTPCRG register, MSTPCRG01 bit
SPI Unit 3	MSTPCRG register, MSTPCRG02 bit
GPT Unit 2	MSTPCRG register, MSTPCRG03 bit
CRC Unit1	MSTPCRG register, MSTPCRG04 bit
RTC	MSTPCRG register, MSTPCRG05 bit
CLMA3	MSTPCRG register, MSTPCRG08 bit
CLMA0	MSTPCRG register, MSTPCRG09 bit
CLMA1	MSTPCRG register, MSTPCRG10 bit
CLMA2	MSTPCRG register, MSTPCRG11 bit
SHOSTIF	MSTPCRI register, MSTPCRI01 bit
AFMT0	MSTPCRJ register, MSTPCRJ00 bit
HDSL0	MSTPCRJ register, MSTPCRJ01 bit
BISS0	MSTPCRJ register, MSTPCRJ02 bit
ENDAT0	MSTPCRJ register, MSTPCRJ03 bit
AFMT1	MSTPCRK register, MSTPCRK00 bit
HDSL1	MSTPCRK register, MSTPCRK01 bit
BISS1	MSTPCRK register, MSTPCRK02 bit

Table 8.3 Peripheral functions requiring the above procedure for release from the module-stop state (3 of 3)

Peripheral functions	Corresponding Module-Stop Control Register
ENDAT1	MSTPCRK register, MSTPCRK03 bit
ENCOUT	MSTPCRL register, MSTPCRL00 bit

8.3.2 Cortex-R52 Standby Mode

8.3.2.1 Transition to Cortex-R52 Standby Mode

The Cortex-R52 enters Standby mode by execution of a WFI instruction. In Standby mode, the Cortex-R52 stops operating, thus reducing power consumption. For details, see the technical reference manual provided by Arm.

8.3.2.2 Release from Cortex-R52 Standby Mode

Release from the Cortex-R52 Standby mode is initiated by any interrupt, the RES# pin reset, an error reset, system software reset, or CPU software reset.

(1) Release triggered by an interrupt signal

Generation of an interrupt in a CPU triggers release from Standby mode or Sleep mode of the CPU and the interrupt exception handling starts. Release is triggered by a non-maskable interrupt or a maskable interrupt of which interrupt request is permitted by using the Interrupt Enable register.

(2) Release by a reset

After the RES# pin reset, error reset, system software reset, or CPU software reset is cleared, the Cortex-R52 CPU0 starts the reset exception handling. For details about resets, see section 5, Reset.

8.4 Usage Notes

8.4.1 I/O Port State

To reduce I/O power consumption, pin processing based on I/O control is required. For details, see section 16, I/O Ports.

8.4.2 On-Chip Peripheral Module Interrupts in Module-Stop State

Peripheral modules cannot interrupt in a module-stop state. Therefore, if the module-stop state is set during interrupt processing of the module or during DMA transfer by the DMAC, a CPU interrupt source or a DMAC startup source cannot be cleared. For this reason, disable the relevant interrupts before setting the module-stop state.

8.4.3 USB Low Power Consumption

For details about how to clear the USB module-stop state and how to enter the USB in low-power consumption mode, see section 29, USB 2.0 Host Module (USBHC) and section 30, USB 2.0 HS Function Module (USBf).

8.4.4 Low Power Consumption for Ethernet-Related Functions

Ethernet-related functions are stopped and reset in the initial state. To use the functions, specify the MSTPCRE.MSTPCRE00 to MSTPCRE03 bits for release from the module-stop state. After the dummy read, specify the MRCTLE.MRCTLE00 to MRCTLE06 bits for release from reset state.

Note: After release from the module-stop state, the module-stop state cannot be set again. Operation is not guaranteed after release from the module-stop state is initiated the second time. After the modules are returned to the initial state (stop state) by a reset, they can be released from the module-stop state again.

8.4.5 Low Power Consumption for HIPERFACE DSL and EnDat 2.2

Encoder interface functions, EnDat 2.2, BiSS-C, HIPERFACE DSL, and A-Format, are stopped and reset in the initial state. To use the functions, specify the MSTPCRJ.MSTPCRJ00 to MSTPCRJ03 bits and MSTPCRK.MSTPCRK00 to MSTPCRJ03 bits for release from the module-stop state. After the dummy read, specify the MRCTLJ.MRCTLJ00 to MRCTLJ03 bits and MRCTLK.MRCTLK00 to MRCTLK03 bits for release from reset state.

Note: After release from the module-stop state, the reset state has to be set when the module-stop state is set. Operation is not guaranteed after release from the module-stop state is initiated the second time without reset state. After the modules are returned to the initial state (stop state) by a reset, they can be released from the module-stop state again.

8.4.6 Write Protection Function

The Module Stop Control registers (MSTPCRA to MSTPCRG and MSTPCRI to MSTPCRL) can be write-protected. To write to the MSTPCRA to MSTPCRG and MSTPCRI to MSTPCRL registers, specify bit 1 of the Protect Register (PRCRN for MSTPCRA to MSTPCRE, and PRCRS for MSTPCRF, MSTPCRG, MSTPCRI, MSTRCRJ, MSTRCRK, and MSTRCRL) to unlock the write protection. For details, see section 10, Register Write Protection Function.

9. Debugging Interface

The LSI has an internal debugging interface that adopts an architecture in which Cortex-R52 are integrated by CoreSight. This LSI supports debugging functions, such as downloading, running, and breaking a program, and trace function, which outputs execution history of programs.

9.1 Overview

The LSI supports JTAG and SWD interfaces for debugging and trace port as interface for trace.

The LSI has TAP controllers for boundary scan and for CoreSight debugging, which can be selected by the input level of the BSCANP pin. To use the debugging function, set the input level of the BSCANP pin to low. For details on boundary scan, see section 37, Boundary Scan.

Table 9.1 lists the specifications of CoreSight and Figure 9.1 shows a block diagram of CoreSight. For details on CoreSight, see the *Arm Technical Reference Manual*.

Table 9.1 CoreSight specifications

Parameter	Description
Debugging function	• JTAG interface • SWD (Serial Wire Debug) interface
Trace function	• Trace port interface – 8 bit × 100 Mbps (50 MHz DDR) trace data pin output – Embedded Trace FIFO (ETF) 16 KB

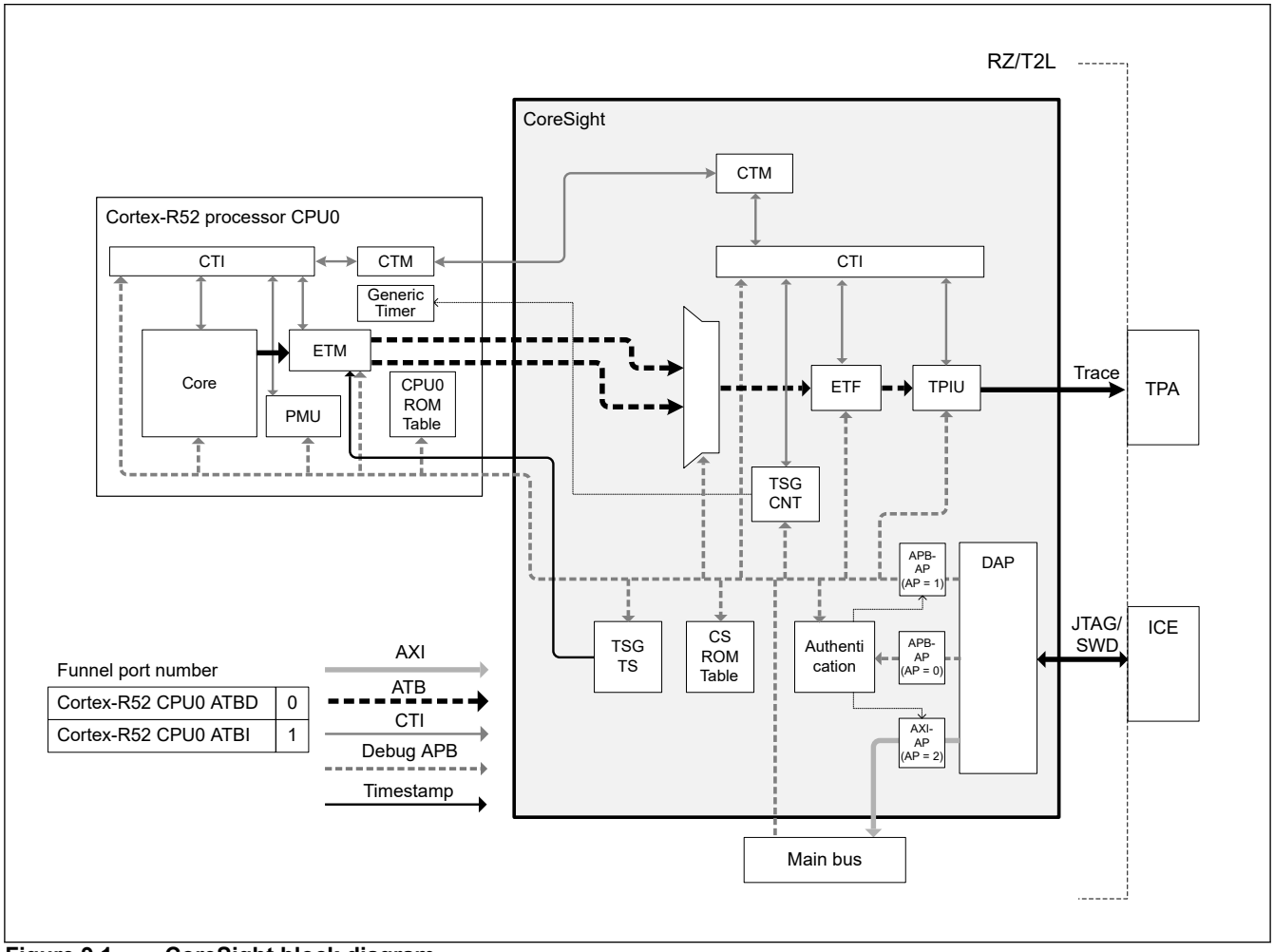


Figure 9.1 CoreSight block diagram

Table 9.2 lists the input and output for the CTI trigger of CoreSight.

Table 9.2 CTI trigger input and output (CoreSight)

CTI input	Description	CTI output	Description
0	ETF ACQCOMP	0	ETF FLUSHIN
1	ETF FULL	1	ETF TRIGIN
2	—	2	TPIU FLUSHIN
3	—	3	TPIU TRIGIN
4	—	4	TSGCNT HALTREQ
5	—	5	TSGCNT RESTARTREQ
6	—	6	—
7	—	7	—

Table 9.3 lists the input and output of the CTI trigger for Cortex-R52 CPU0.

Table 9.3 CTI trigger input and output (Cortex-R52 CPU0)

CTI input	Description	CTI output	Description
0	Cross halt trigger event	0	Requests the processor to enter debug state
1	PMU interrupt	1	Requests the processor to exit debug state
2	—	2	CTI interrupt
3	—	3	—
4	ETM trace unit external output 0	4	ETM trace unit external input 0
5	ETM trace unit external output 1	5	ETM trace unit external input 1
6	ETM trace unit external output 2	6	ETM trace unit external input 2
7	ETM trace unit external output 3	7	ETM trace unit external input 3

Table 9.4 lists the address map of CoreSight (Debug-APB).

Table 9.4 CoreSight address map (Debug-APB)

Cortex-R52 CPU0 view	Debugger view (AP = 1)	Module
0xC000_0000 to 0xC000_0FFF	0x8000_0000 to 0x8000_0FFF	CoreSight/ROM Table
0xC001_0000 to 0xC001_0FFF	0x8001_0000 to 0x8001_0FFF	CoreSight/TMC-ETF
0xC002_0000 to 0xC002_0FFF	0x8002_0000 to 0x8002_0FFF	CoreSight/CTI
0xC003_0000 to 0xC003_0FFF	0x8003_0000 to 0x8003_0FFF	CoreSight/Funnel
0xC004_0000 to 0xC004_0FFF	0x8004_0000 to 0x8004_0FFF	CoreSight/TPIU
0xC005_0000 to 0xC005_0FFF	0x8005_0000 to 0x8005_0FFF	CoreSight/TSG-TS
0xC006_0000 to 0xC006_0FFF	0x8006_0000 to 0x8006_0FFF	CoreSight/TSG-CNT
0xC040_0000 to 0xC040_0FFF	0x8040_0000 to 0x8040_0FFF	Cortex-R52 CPU0 ROM Table
0xC041_0000 to 0xC041_0FFF	0x8041_0000 to 0x8041_0FFF	Cortex-R52 CPU0 Debug
0xC042_0000 to 0xC042_0FFF	0x8042_0000 to 0x8042_0FFF	Cortex-R52 CPU0 CTI
0xC043_0000 to 0xC043_0FFF	0x8043_0000 to 0x8043_0FFF	Cortex-R52 CPU0 PMU
0xC044_0000 to 0xC044_0FFF	0x8044_0000 to 0x8044_0FFF	Cortex-R52 CPU0 ETM

Note: No listed areas between 0xC000_0000 and 0xC0FF_FFFF for Cortex-R52 CPU0 view and between 0x8000_0000 and 0x80FF_FFFF for debugger view are reserved. Do not access those areas.

Table 9.5 lists the input/output pins of the debugging interface.

Table 9.5 Configuration of pins for the debugging interface

Name	Pin name	I/O	Functions
Test clock	TCK	Input	Data is serially supplied from the Test Data Input (TDI) pin to this module, synchronized with this clock, and output from the Test Data Output (TDO) pin. In SWD mode, this pin functions as the SWCLK pin. This pin can also be used as a general-purpose port. The initial function is TCK.
Test mode select	TMS	Input/ Output	Changing the level of this signal, by synchronizing with TCK, determines the status of the TAP (Test Access Port) control circuit. The protocol conforms to the JTAG standard (IEEE Std.1149.1). In SWD mode, this pin functions as the SWDIO pin. This pin can also be used as a general-purpose port. The initial function is TMS.
Test reset	TRST#	Input	This pin accepts input asynchronously with TCK. When the level of this pin is low, TAP (Test Access Port) is reset. When the levels of the TRST# pin and RES# pin are both low, TAP and the debugging circuit are reset.
Test data input	TDI	Input	Changing the level of this pin, by synchronizing with TCK, sends data to this module. This pin can also be used as a general-purpose port. The initial function is TDI.
Test data output	TDO	Output	Reading the level of this pin, by synchronizing with TCK, reads data from this module. This pin can also be used as a general-purpose port. The initial function is TDO.
Trace clock output	TRACECLK	Output	This pin is an output pin of the clock used for synchronizing trace data.
Trace enable output	TRACECTL	Output	This pin is an output pin of the enable signal for trace control.
Trace data output	TRACEDATA7 to TRACEDATA0	Output	These pins are output pins of trace data.
Boundary scan setting	BSCANP	Input	Input the high level when boundary scan test is performed. Input low when debugging is performed by CoreSight. For details on boundary scan, see section 37, Boundary Scan.

Note: When you design a board on which an emulator can be used, set the TRST# pin to low while the RES# pin is asserted at power up, and configure the board so that control is available by the TRST# pin only. When this pin is not used, fix it to low, or connect it so that the same signal as that on the RES# pin is to be input. For details, see section 9.3.4. Reset Configuration and the Method of Connecting with the Emulator.

9.2 Register Descriptions

9.2.1 OIRn : OCD Authentication ID Code Register n (n = 0 to 3)

Address: 0x0008_0000 + 0x04 × n (the address is the debugger view (AP = 0))

Bit position:	31																												0
Bit field:	IDn[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	IDn[31:0]	128-bit Authentication ID n = 0: Authentication ID[31:0] n = 1: Authentication ID[63:32] n = 2: Authentication ID[95:64] n = 3: Authentication ID[127:96]	W*1

Note 1. Once all four registers are written, further write is blocked.

The OIRn registers are used to write 128-bit authentication ID from the external debugger to access debug system.

9.2.2 OSR : OCD Status Register

Address: 0x0008_0010 (the address is the debugger view (AP = 0))

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	WFLG	RESULT[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RESULT[1:0]	Authentication result 0 0: No authentication 0 1: Passed Authentication level 1. Both EL2 and EL1/EL0 accesses are permitted. 1 0: Passed Authentication level 2. Only EL1/EL0 access is permitted. 1 1: Failed or access permanently prohibited	R
2	WFLG	All OIRn registers written flag 0: Not written 1: Written	R
31:3	—	These bits are read as 0.	R

The OSR register indicates authentication result.

9.3 Operation

9.3.1 JTAG Interface

The JTAG interface uses five signals (TCK, TMS, TDO, TDI, and TRST#) to communicate with the host machine (PC) through the emulator. Figure 9.2 shows an example connection, which includes the RES# pin connection.

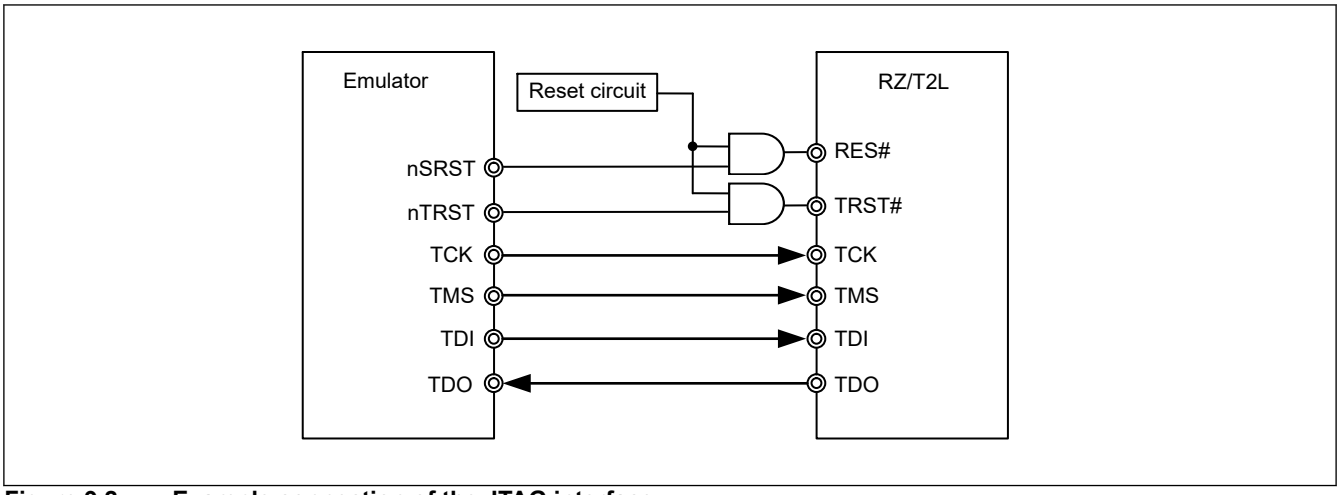


Figure 9.2 Example connection of the JTAG interface

When the JTAG interface is not used, the TCK, TMS, TDI, and TDO pins can be used as general-purpose ports. When you use these pins as general-purpose ports, perform pin settings by referring to section 16, I/O Ports.

9.3.2 SWD Interface

The SWD (Serial Wire Debug) interface uses two signals (SWCLK (TCK) and SWDIO (TMS)) to communicate with the host machine (PC) through the emulator. Figure 9.3 shows an example connection, which includes the RES# pin connection.

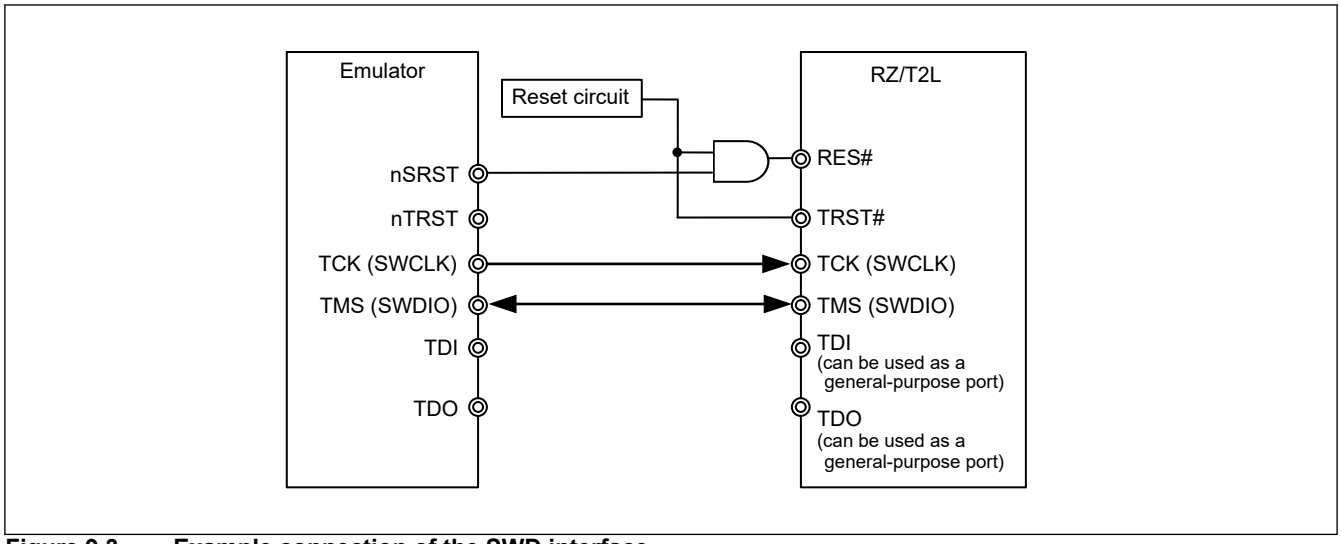


Figure 9.3 Example connection of the SWD interface

When the SWD interface is used for debugging, the TDI and TDO pins can be used as general-purpose ports. When using these pins as general-purpose ports, perform pin settings by referring to section 16, I/O Ports.

Note: In the initial status of the LSI, the debugging interface is in JTAG mode. If you use the TDI and TDO pins as general-purpose ports and the emulator connection for debugging, switch the mode to SWD (Serial Wire Debug) mode using the control from the debugger, and then start debugging.

9.3.3 Trace Port Interface

The trace port interface uses ten signals (TRACECLK, TRACECTL, and TRACEDATA7 to TRACEDATA0) to output trace information. Information about branch instructions of the executed program (obtained by the ETM (Embedded Trace Macrocell) trace) is output from the trace port interface. After the debugger complements the information, you can know the branch source and destination at the time a branch occurred. For details on trace information, see the manual of each emulator vendor.

Only DDR clocking mode is supported for the synchronization relationship between the TRACECLK pin and TRACEDATA pin.

The maximum number of available TRACEDATA pins is 8. If the number of TRACEDATA pins is smaller than 8, the pins with smaller numbers are used (from TRACEDATA0). Set whether to connect the TRACECTL pin to TPA (Trace Port Analyzer), according to the specifications of the trace data transfer format of the TPA.

As the output frequency of the TRACECLK pin, 50 MHz (obtained by dividing the trace I/F clock (TCLK) by 2) can be set. For details, see section 6, Clock Generation Circuit.

In the initial status, different functions are assigned to the TRACECLK, TRACEDATA0 to TRACEDATA7, and TRACECTL pins. Perform pin settings by referring to section 16, I/O Ports.

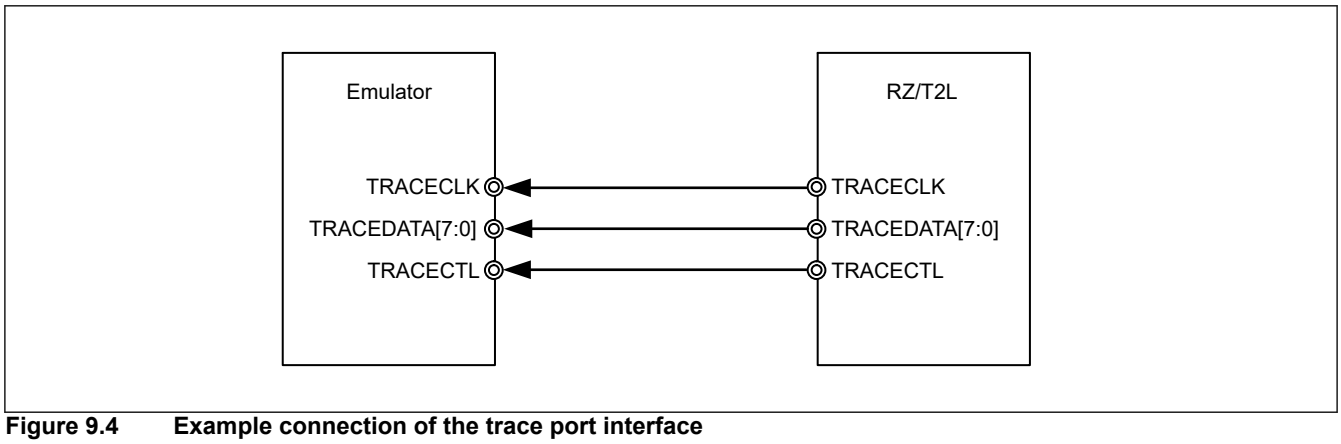


Figure 9.4 Example connection of the trace port interface

9.3.4 Reset Configuration and the Method of Connecting with the Emulator

When designing a board on which the emulator can be used, set the TRST# pin to low while the RES# pin is asserted at power up. Also, configure the board so that control is available by the TRST# pin only.

When debugging is performed, if the initial level of the RES# and TRST# pins are both low, the CPU and debugging section become the reset status. Then, setting the TRST# pin to high while the RES# pin is kept to low enables the TAP access to set SWJ-DP registers. After RES# pin is set to high, it is possible to access OCD registers (OIRn and OSR) in AP = 0. Debugging setting is possible after on-chip boot code runs if JTAG authentication passes, or no authentication is needed.

When the emulator is not connected, fix the TRST# pin to low, or let the signal be the same as that on the RES# pin input to the TRST# pin.

Note: When debugging is performed by CoreSight, input low to the BSCANP pin to disable the boundary scan function.

9.3.4.1 Example Connection of the Emulator That Cannot Drive the nTRST Output to High

Figure 9.5 shows an example of connection circuit when an emulator that cannot drive the nTRST output to high is used. The TRST# pin is pulled up and is asserted to low by the emulator.

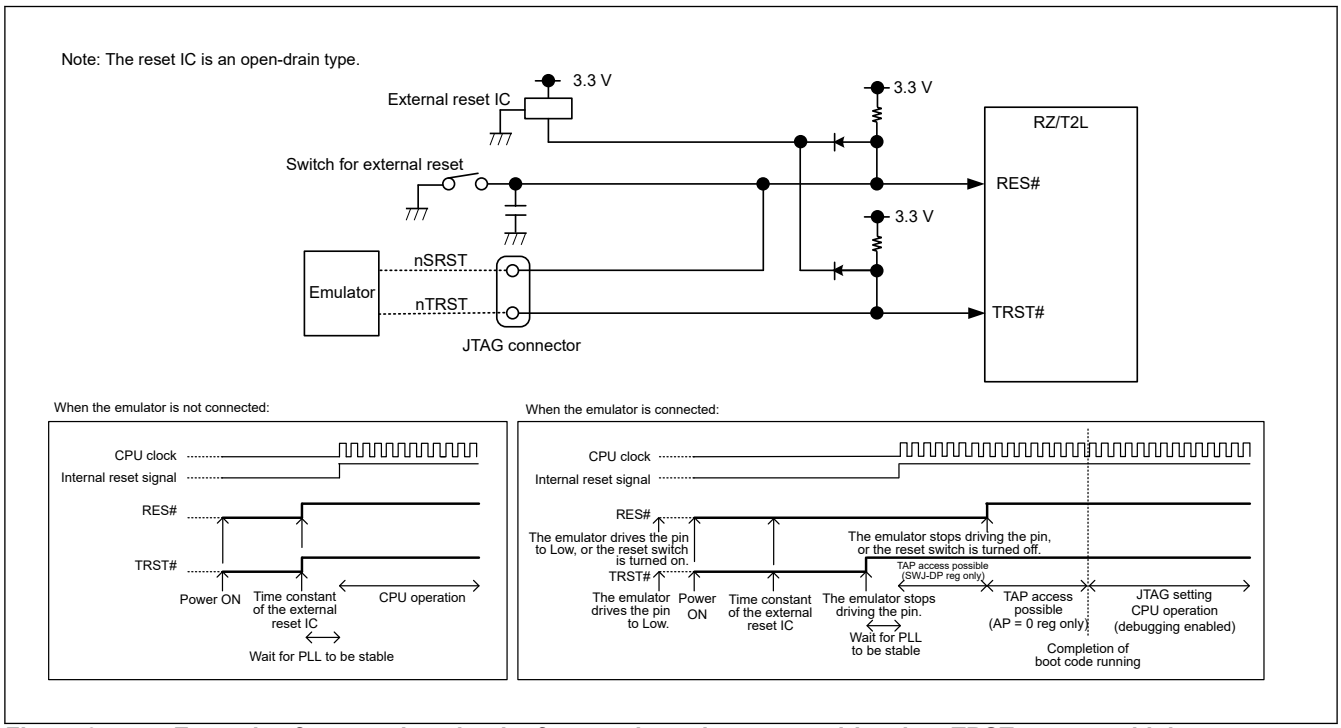


Figure 9.5 Example of connection circuit of an emulator that cannot drive the nTRST output to high

9.3.4.2 Example Connection of the Emulator That Can Drive the nTRST Output to High

Figure 9.6 shows an example of connection circuit when an emulator that can drive the nTRST output to high is used. The TRST# pin (high or low) is controlled by the emulator.

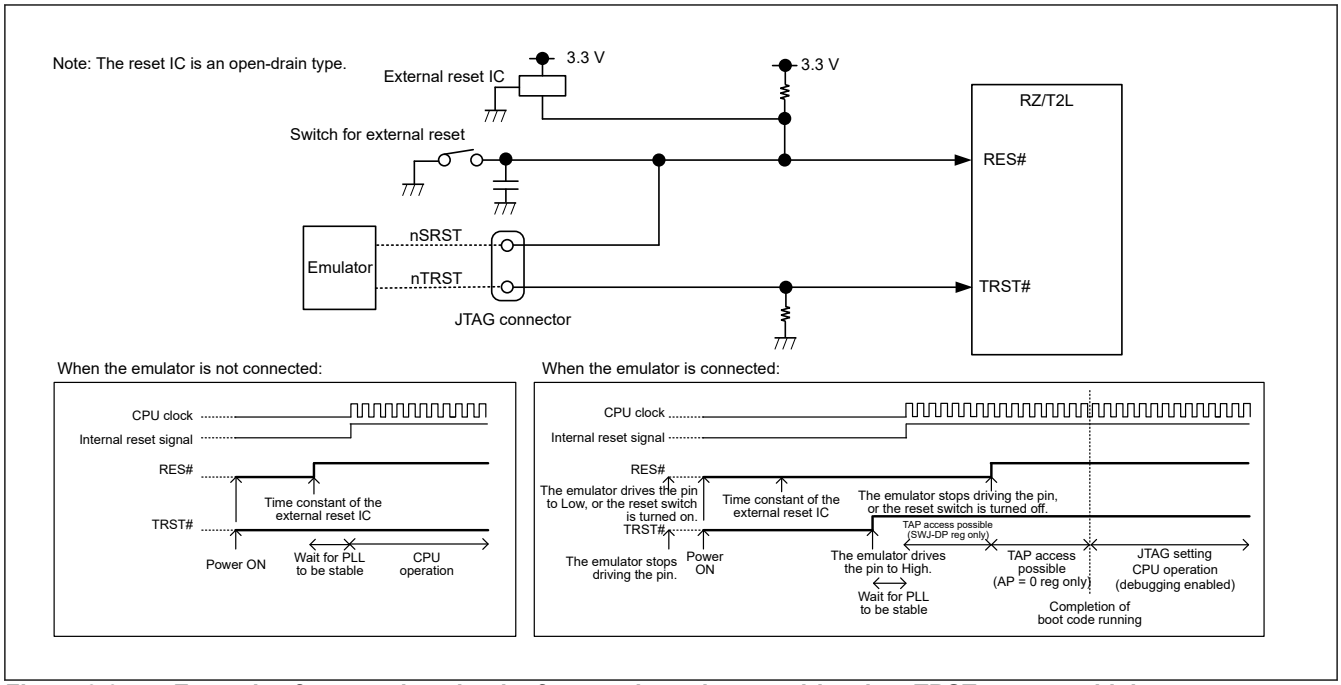


Figure 9.6 Example of connection circuit of an emulator that can drive the nTRST output to high

9.3.5 JTAG Pin Treatment When No Emulator Is Connected

If no emulator is connected, pin treatment is required according to Table 9.6.

Table 9.6 JTAG pin treatment when no emulator is connected

Pin name	Treatment
TCK	Pull down the pin (except when the port is used as a general-purpose port)
TMS	Pull up the pin (except when the port is used as a general-purpose port)
TDI	Pull up the pin (except when the port is used as a general-purpose port)
TDO	Open the pin (except when the port is used as a general-purpose port)
TRST#	Pull down the pin, or let the signal be the same as that on the RES# pin input

9.3.6 Noise Reduction of the TRST# Pin

Analog delay noise reduction is performed for the TRST# pin. This measure against noise can remove noise that is within 100 ns at maximum.

9.3.7 Available Trace Functions

Table 9.7 lists the trace functions that are available through the respective debugging ports (trace port interface and SWD or JTAG).

Table 9.7 Available trace functions

CPU core	Debugging port	Trace functions
Cortex-R52	Trace port interface	Full instruction and data tracing through the ETM of the Cortex-R52. ETF is programmed to be in Circular Buffer mode or Hardware FIFO mode.
	SWD/JTAG	The same information as for the trace port interface can be acquired through the ETF which is programmed to be in Software FIFO mode.

9.3.8 Debug Authentication

Debug authentication mechanism checks access permission for debugging and LSI resources. To obtain full debug functionality, a pass result of the authentication mechanism is required if authentication is required.

Authentication level is selectable from four states by setting AUTHMODEJ area of OTP (One Time Programmable memory):

- No authentication
- Authentication level 1 (both EL2 and EL1/EL0 access permission)
- Authentication level 2 (only EL1/EL0 access permission)
- Permanent prohibition

If authentication level 1 or 2 is selected, 128-bit authentication ID must be written to OCD Authentication ID Code Register n (OIRn) (n = 0 to 3) from debugger to compare with the ID which is already stored in OTP. Once all OIRn registers are written, authentication automatically starts and the authentication result is available in OCD Status Register (OSR). If authentication failed, then no debug connection is permitted. Reset (RES# pin reset, system software reset, or error reset) is required to retry debug authentication. If "No Authentication" is selected, RESULT[1:0] bits in OSR register indicate 01b which means both EL2 and EL1/EL0 accesses are permitted.

9.3.9 Debug Authentication Procedure

Figure 9.7 shows the debug authentication procedure.

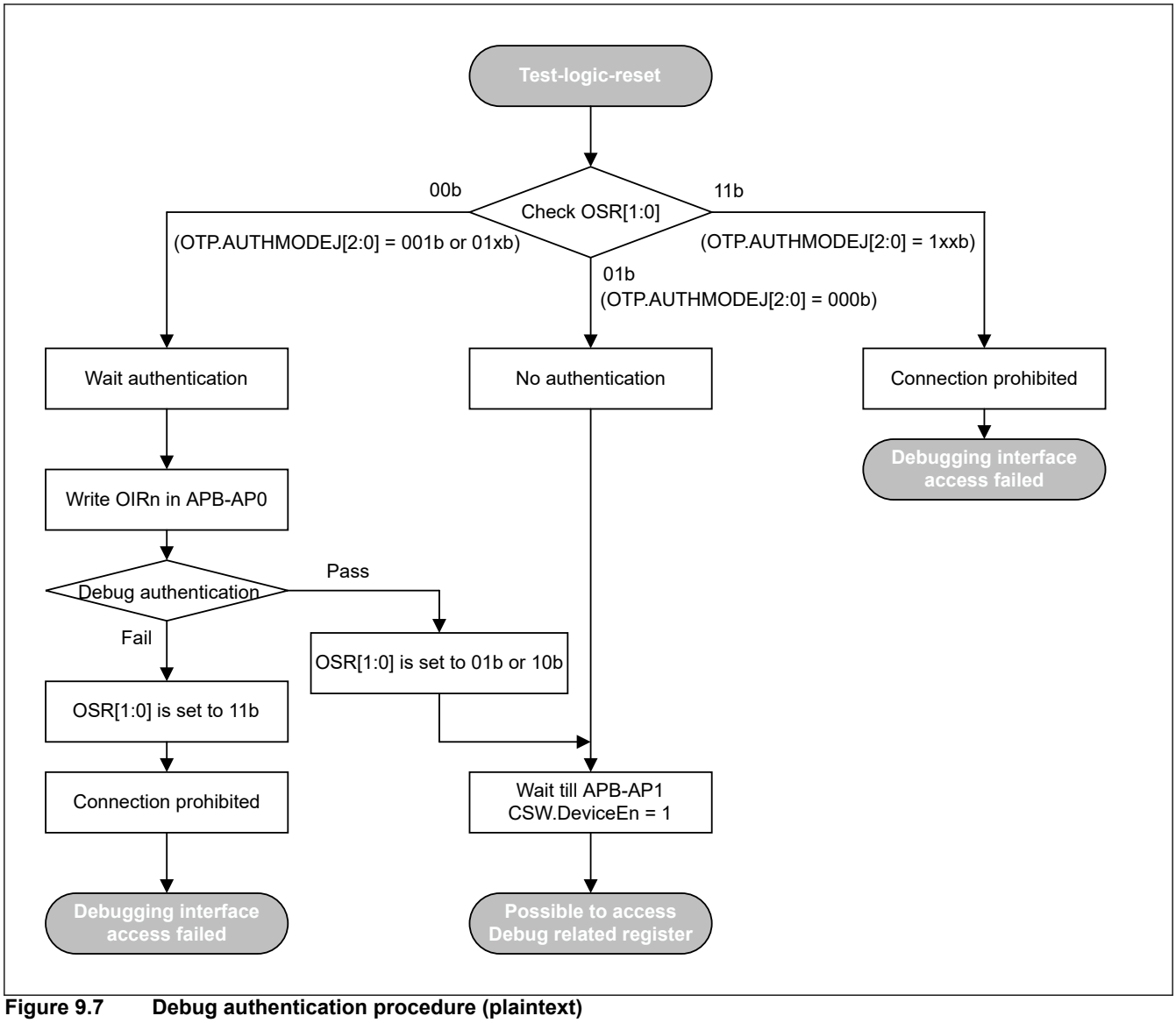


Figure 9.7 Debug authentication procedure (plaintext)

9.4 Usage Notes

9.4.1 Access to the Main Bus

Access to the main bus by the DAP is through either the Cortex-R52 CPU0 or AXI-AP. When access is to be through the AXI-AP, the setting of M-MPU for AXI-AP must be changed to access internal resources.

9.4.2 Authentication Level

Authentication level 2 is available only when debugger can access by EL1/EL0. In case using debugger which can access by EL2 only, do not use authentication level 2.

9.4.3 Debug Reset

The reset request from debugger is prohibited during boot operation.

10. Register Write Protection Function

10.1 Overview

The register write protection function protects important registers from being overwritten in case a program runs out of control. The registers to be protected are set with the protect register (PRCRN and PRCRS).

Table 10.1 and Table 10.2 show the relationship between the PRCRN/PRCRS register bits and the registers to be protected.

Table 10.1 Relationship between PRCRN register bits and registers to be protected

PRCRN register	Registers to be protected
PRC0 bit	• Registers related to the clock generation circuit: SCKCR
PRC1 bit	• Registers related to the low-power consumption functions: MSTPCRA, MSTPCRB, MSTPCRC, MSTPCRD, MSTPCRE, MSTPCRJ, MSTPCRK, MSTPCRL • Reset-related registers: RSTSR0, MRCTLA, MRCTLE, MRCTLJ, MRCTLK
PRC2 bit	• GPIO-related registers: PMC00 to PMC24, PFC00 to PFC24, DRCTL00 to DRCTL24, ELC_PGR1, ELC_PGR2, ELC_PGC1, ELC_PGC2, ELC_PDBF1, ELC_PDBF2, ELC_PEL0 to ELC_PEL3, ELC_DPTC, ELC_ELSR2
PRC3 bit	• Encoder interface subsystem registers: ENCODER_CFG, ENDAT_CFG

Table 10.2 Relationship between PRCRS register bits and registers to be protected

PRCRS register	Registers to be protected
PRC0 bit	• Registers related to the clock generation circuit: SCKCR2, LOCOCR
PRC1 bit	• Registers related to the clock generation circuit: PLL1EN • Registers related to the low-power consumption functions: MSTPCRF, MSTPCRG • Reset related registers: SWRSYS, SWRCPU0, MRCTLI
PRC2 bit	• GPIO-related registers: PMC00 to PMC24, PFC00 to PFC24, DRCTL00 to DRCTL24, RSELP00 to RSELP24, ELC_PGR1, ELC_PGR2, ELC_PGC1, ELC_PGC2, ELC_PDBF1, ELC_PDBF2, ELC_PEL0 to ELC_PEL3, ELC_DPTC, ELC_ELSR2
PRC3 bit	• System control registers: STADD0 to STADD7, ENDADD0 to ENDADD7, ERRINF, ERRINF_R, ERRINF_W, SYSRAM_CTRL0 to SYSRAM_CTRL1

10.2 Register Descriptions

10.2.1 PRCRN : Non_Safety Area Protect Register

Base address: RWP_NS = 0x8028_1A10

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PRKEY[7:0]								—	—	—	—	PRC3	PRC2	PRC1	PRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRC0	Protect 0 Enables writing to the registers related to the clock generation circuit. 0: Write disabled 1: Write enabled	R/W
1	PRC1	Protect 1 Enables writing to the registers related to low power consumption and reset. 0: Write disabled 1: Write enabled	R/W
2	PRC2	Protect 2 Enables writing to the registers related to GPIO settings. 0: Write disabled 1: Write enabled	R/W
3	PRC3	Protect 3 Enables writing to the registers related to Encoder interface subsystem configuration. 0: Write disabled 1: Write enabled	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	PRC Key Code These bits control permission and prohibition of writing to the PRCRN register. To modify the PRCRN register, write 0xA5 to the PRKEY[7:0] bits. Any other value is ignored if written to the PRCRN register.*1	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Write a value to the PRKEY[7:0] bits at the same time as other bits. When setting PRC0 to 1 and, PRC1 and PRC2 to 0, for example, write 0x0000A501 to the PRCRN register.

PRCi bit (Protect i) (i = 0 to 3)

Setting the PRCi bit to 1 or 0 enables or disables writing to the corresponding registers to be protected, respectively.

The PRCRN register controls writes to the protected registers for non-safety area. The non-safety area is as follows:

- 0x800A_0000 to 0x800A_0FFF: address area for GPIO-related registers
- 0x8028_0000 to 0x8028_1FFF: address area for clock generation, low power consumption, reset, and Encoder interface subsystem configuration registers

10.2.2 PRCRS : Safety Area Protect Register

Base address: RWP_S = 0x8128_1A00

Offset address: 0x00

Bit position:	31																15								8				3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PRKEY[7:0]								—	—	—	—	PRC3	PRC2	PRC1	PRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRC0	Protect 0 Enables writes to the registers related to the clock generation circuit. 0: Write disabled 1: Write enabled	R/W
1	PRC1	Protect 1 Enables writes to the registers related to low power consumption and reset. 0: Write disabled 1: Write enabled	R/W
2	PRC2	Protect 2 Enables writes to the registers related to GPIO settings. 0: Write disabled 1: Write enabled	R/W

Bit	Symbol	Function	R/W
3	PRC3	Protect 3 Enables writes to the system control registers. 0: Write disabled 1: Write enabled	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	PRC Key Code These bits control permission and prohibition of writing to the PRCRS register. To modify the PRCRS register, write 0xA5 to the PRKEY[7:0] bits. Any other value is ignored if written to the PRCRS register.*1	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Write a value to the PRKEY[7:0] bits at the same time as other bits. When setting PRC0 to 1 and PRC1, PRC2, and PRC3 to 0, for example, write 0x0000A501 to the PRCRS register.

PRCi bit (Protect i) (i = 0 to 3)

Setting the PRCi bit to 1 or 0 enables or disables writing to the corresponding registers to be protected, respectively.

The PRCRS register controls writes to the protected registers for safety area. The safety area is as follows:

- 0x8103_0000 to 0x8103_0FFF: address area for GPIO-related registers
- 0x8128_0000 to 0x8128_1FFF: address area for system control, clock generation, low power consumption, and reset registers

11. Interrupt Controller (ICU)

11.1 Overview

The interrupt controller accepts interrupt request from peripheral module, external pins include the external DMA request pins. An interrupt accepted by the interrupt controller is set either as an interrupt to the CPU (Cortex-R52 CPU0) or as an activating trigger signal for the DMAC and ELC.

Table 11.1 lists the specifications of the ICU, and Figure 11.1 shows a block diagram of the interrupt controller.

Table 11.1 ICU specifications

Item		Description
Interrupts	Interrupt contact destinations	• Cortex-R52 CPU0 • Two DMAC units • ELC
	Peripheral function interrupts	Interrupts from peripheral modules Interrupt detection: Edge detection/level detection
	External pin interrupts	Interrupts from pins IRQ0 to IRQ15 Number of sources: 16 IRQ0 to IRQ13: for NONSAFETY IRQ14 to IRQ15: for SAFETY Interrupt detection: Low level/falling edge/rising edge/rising and falling edges. One of these detection methods can be set for each source. Digital noise filter function: Supported
	Software interrupts	Software interrupts Number of sources: 8 INTCPU0 to INTCPU5: for NONSAFETY INTCPU6 to INTCPU7: for SAFETY
	Error events	• Error events from CPU and peripherals are captured and merged to two interrupts with mask. Initial state is all masked. (no interrupt generated) • Hardware reset can be generated by error events.
	DMAC control	Interrupts can be assigned as DMA activation sources.
	External DMA request	External DMA request pins (DREQ) can output DACK, TEND when DMA transfer to external bus. Number of sources: 1 Digital noise filter function: Supported
Non maskable interrupts	NMI pin interrupts	Interrupts from the NMI pin Interrupt detection: Falling edge/rising edge Digital noise filter function: Supported The NMI pin can be allocated as SEI (system error interrupt).
Restoration from the sleep status		Restoration due to non-maskable interrupt and all unmasked interrupt sources

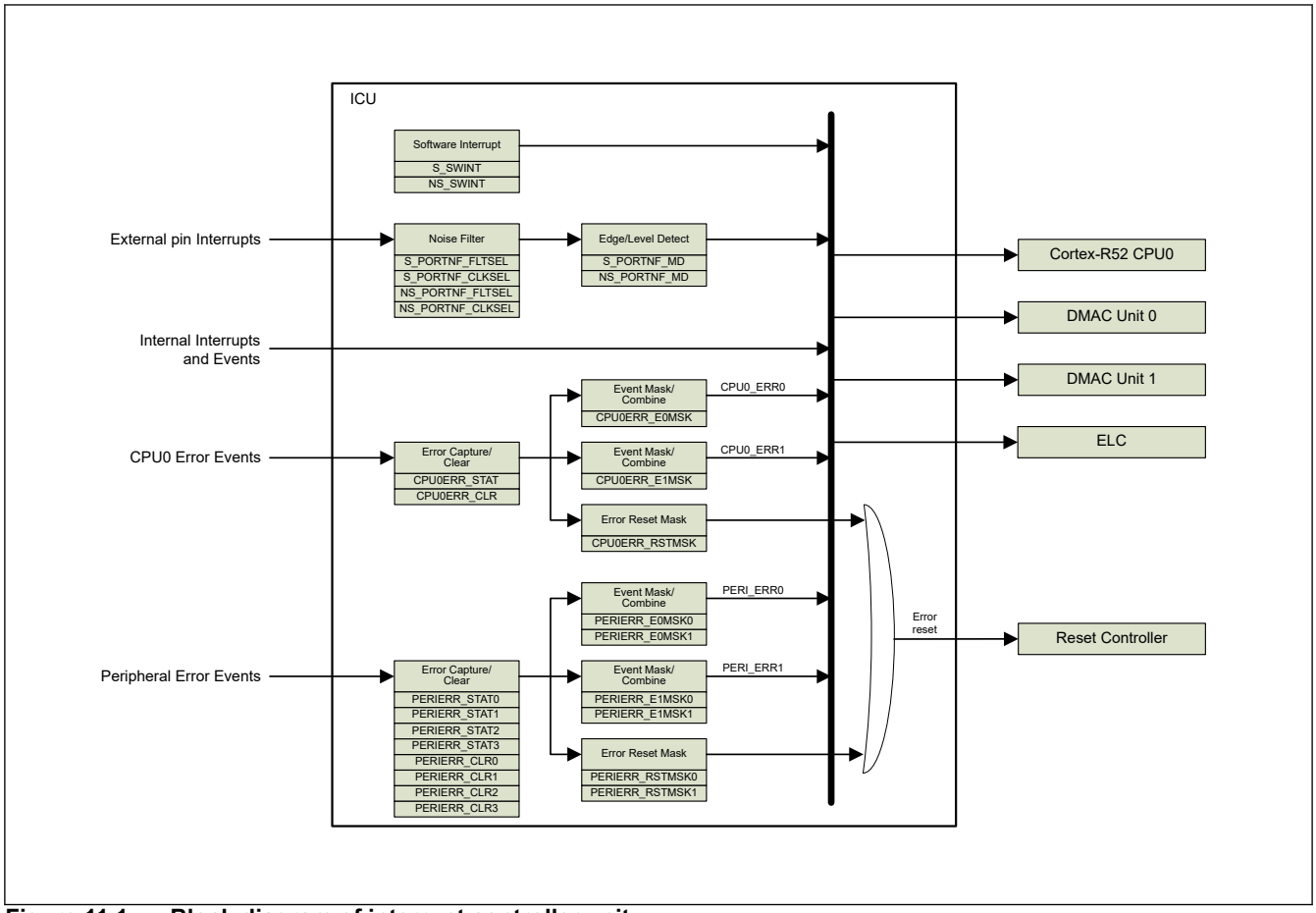


Figure 11.1 Block diagram of interrupt controller unit

11.2 Register Descriptions

This section describes details of the register functions.

11.2.1 Interrupt Control Unit Register 0 (Safety Region)

11.2.1.1 S_SWINT : Software Interrupt Register for Safety Register

Base address: ICU = 0x8104_8000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	IC7	IC6
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IC6	Software Interrupt register Occur software interrupt INTCPU6 when this value is set to 1. This bit is read as 0.	W

Bit	Symbol	Function	R/W
1	IC7	Software Interrupt register Occur software interrupt INTCPU7 when this value is set to 1. This bit is read as 0.	W
31:2	—	The write value should be 0.	W

11.2.1.2 S_PORTNF_FLTSEL : Interrupt Noise Filter Enable Register for Safety Register

Base address: ICU = 0x8104_8000

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	FLTNMI	FLT15	FLT14
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FLT14	Noise filter enable for IRQ14 0: Disable 1: Enable	R/W
1	FLT15	Noise filter enable for IRQ15 0: Disable 1: Enable	R/W
2	FLTNMI	Noise filter enable for NMI 0: Disable 1: Enable	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

11.2.1.3 S_PORTNF_CLKSEL : Interrupt Noise Filter Setting Register for Safety Register

Base address: ICU = 0x8104_8000

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	CKSELNMI[1:0]		CKSEL15[1:0]		CKSEL14[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CKSEL14[1:0]	Select noise filter sampling frequency dividend rate for IRQ14. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W

Bit	Symbol	Function	R/W
3:2	CKSEL15[1:0]	Select noise filter sampling frequency dividend rate for IRQ15. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
5:4	CKSELNMI[1:0]	Select noise filter sampling frequency dividend rate for NMI. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

11.2.1.4 S_PORTNF_MD : Interrupt Edge Detection Setting Register for Safety Register

Base address: ICU = 0x8104_8000

Offset address: 0x00C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	MDNMI[1:0]		MD15[1:0]		MD14[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1

Bit	Symbol	Function	R/W
1:0	MD14[1:0]	Select detection mode for IRQ14 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
3:2	MD15[1:0]	Select detection mode for IRQ15 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
5:4	MDNMI[1:0]	Select detection mode for NMI 0 0: Invert 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

11.2.1.5 CPU0ERR_STAT : CPU0 Error Event Status Register

Base address: ICU = 0x8104_8000

Offset address: 0x060

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	ER_ST25	ER_ST24	ER_ST23	ER_ST22	ER_ST21	ER_ST20	ER_ST19	ER_ST18	ER_ST17	ER_ST16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ER_ST15	ER_ST14	ER_ST13	ER_ST12	ER_ST11	ER_ST10	ER_ST9	ER_ST8	ER_ST7	ER_ST6	ER_ST5	ER_ST4	ER_ST3	ER_ST2	ER_ST1	ER_ST0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_ST0	Indicate captured error status for CPU0_ERREVENT0 0: No error 1: Error occurred	R
1	ER_ST1	Indicate captured error status for CPU0_ERREVENT1 0: No error 1: Error occurred	R
2	ER_ST2	Indicate captured error status for CPU0_ERREVENT2 0: No error 1: Error occurred	R
3	ER_ST3	Indicate captured error status for CPU0_ERREVENT3 0: No error 1: Error occurred	R
4	ER_ST4	Indicate captured error status for CPU0_ERREVENT4 0: No error 1: Error occurred	R
5	ER_ST5	Indicate captured error status for CPU0_ERREVENT5 0: No error 1: Error occurred	R
6	ER_ST6	Indicate captured error status for CPU0_ERREVENT6 0: No error 1: Error occurred	R
7	ER_ST7	Indicate captured error status for CPU0_ERREVENT7 0: No error 1: Error occurred	R
8	ER_ST8	Indicate captured error status for CPU0_ERREVENT8 0: No error 1: Error occurred	R
9	ER_ST9	Indicate captured error status for CPU0_ERREVENT9 0: No error 1: Error occurred	R
10	ER_ST10	Indicate captured error status for CPU0_ERREVENT10 0: No error 1: Error occurred	R
11	ER_ST11	Indicate captured error status for CPU0_ERREVENT11 0: No error 1: Error occurred	R
12	ER_ST12	Indicate captured error status for CPU0_ERREVENT12 0: No error 1: Error occurred	R

Bit	Symbol	Function	R/W
13	ER_ST13	Indicate captured error status for CPU0_ERREVENT13 0: No error 1: Error occurred	R
14	ER_ST14	Indicate captured error status for CPU0_ERREVENT14 0: No error 1: Error occurred	R
15	ER_ST15	Indicate captured error status for CPU0_ERREVENT15 0: No error 1: Error occurred	R
16	ER_ST16	Indicate captured error status for CPU0_ERREVENT16 0: No error 1: Error occurred	R
17	ER_ST17	Indicate captured error status for CPU0_ERREVENT17 0: No error 1: Error occurred	R
18	ER_ST18	Indicate captured error status for CPU0_ERREVENT18 0: No error 1: Error occurred	R
19	ER_ST19	Indicate captured error status for CPU0_ERREVENT19 0: No error 1: Error occurred	R
20	ER_ST20	Indicate captured error status for CPU0_ERREVENT20 0: No error 1: Error occurred	R
21	ER_ST21	Indicate captured error status for CPU0_ERREVENT21 0: No error 1: Error occurred	R
22	ER_ST22	Indicate captured error status for CPU0_ERREVENT22 0: No error 1: Error occurred	R
23	ER_ST23	Indicate captured error status for CPU0_ERREVENT23 0: No error 1: Error occurred	R
24	ER_ST24	Indicate captured error status for CPU0_ERREVENT24 0: No error 1: Error occurred	R
25	ER_ST25	Indicate captured error status for CPU0_ERREVENT25 0: No error 1: Error occurred	R
31:26	—	These bits are read as 0.	R

11.2.1.6 PERIERR_STAT0 : Peripheral Error Event Status Register 0

Base address: ICU = 0x8104_8000

Offset address: 0x068

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ER_ST31	ER_ST30	ER_ST29	ER_ST28	ER_ST27	ER_ST26	ER_ST25	ER_ST24	ER_ST23	ER_ST22	ER_ST21	ER_ST20	ER_ST19	ER_ST18	ER_ST17	ER_ST16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ER_ST15	ER_ST14	ER_ST13	ER_ST12	ER_ST11	ER_ST10	ER_ST9	—	ER_ST7	ER_ST6	ER_ST5	ER_ST4	ER_ST3	ER_ST2	ER_ST1	ER_ST0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_ST0	Indicate captured error status for CLMA3_INT 0: No error 1: Error occurred	R
1	ER_ST1	Indicate captured error status for CLMA0_INT 0: No error 1: Error occurred	R
2	ER_ST2	Indicate captured error status for CLMA1_INT 0: No error 1: Error occurred	R
3	ER_ST3	Indicate captured error status for CLMA2_INT 0: No error 1: Error occurred	R
4	ER_ST4	Indicate captured error status for BSC_WTO 0: No error 1: Error occurred	R
5	ER_ST5	Indicate captured error status for DMAC0_ERR 0: No error 1: Error occurred	R
6	ER_ST6	Indicate captured error status for DMAC1_ERR 0: No error 1: Error occurred	R
7	ER_ST7	Indicate captured error status for WDT_NMIUNDF0 0: No error 1: Error occurred	R
8	—	This bit is read as 0.	R
9	ER_ST9	Indicate captured error status for USB_FDMAERR 0: No error 1: Error occurred	R
10	ER_ST10	Indicate captured error status for DSMIF0_LTCSE 0: No error 1: Error occurred	R
11	ER_ST11	Indicate captured error status for DSMIF0_UTCSE 0: No error 1: Error occurred	R
12	ER_ST12	Indicate captured error status for DSMIF0_OVC0ELn (n = 0 to 2) 0: No error 1: Error occurred	R
13	ER_ST13	Indicate captured error status for DSMIF0_OVC1ELn (n = 0 to 2) 0: No error 1: Error occurred	R
14	ER_ST14	Indicate captured error status for DSMIF0_OVC2ELn (n = 0 to 2) 0: No error 1: Error occurred	R
15	ER_ST15	Indicate captured error status for DSMIF0_OVC0EHn (n = 0 to 2) 0: No error 1: Error occurred	R
16	ER_ST16	Indicate captured error status for DSMIF0_OVC1EHn (n = 0 to 2) 0: No error 1: Error occurred	R
17	ER_ST17	Indicate captured error status for DSMIF0_OVC2EHn (n = 0 to 2) 0: No error 1: Error occurred	R
18	ER_ST18	Indicate captured error status for DSMIF0_SCDE0 0: No error 1: Error occurred	R

Bit	Symbol	Function	R/W
19	ER_ST19	Indicate captured error status for DSMIF0_SCDE1 0: No error 1: Error occurred	R
20	ER_ST20	Indicate captured error status for DSMIF0_SCDE2 0: No error 1: Error occurred	R
21	ER_ST21	Indicate captured error status for DSMIF1_LTCSE 0: No error 1: Error occurred	R
22	ER_ST22	Indicate captured error status for DSMIF1_UTCSE 0: No error 1: Error occurred	R
23	ER_ST23	Indicate captured error status for DSMIF1_OVC0ELn (n = 0 to 2) 0: No error 1: Error occurred	R
24	ER_ST24	Indicate captured error status for DSMIF1_OVC1ELn (n = 0 to 2) 0: No error 1: Error occurred	R
25	ER_ST25	Indicate captured error status for DSMIF1_OVC2ELn (n = 0 to 2) 0: No error 1: Error occurred	R
26	ER_ST26	Indicate captured error status for DSMIF1_OVC0EHn (n = 0 to 2) 0: No error 1: Error occurred	R
27	ER_ST27	Indicate captured error status for DSMIF1_OVC1EHn (n = 0 to 2) 0: No error 1: Error occurred	R
28	ER_ST28	Indicate captured error status for DSMIF1_OVC2EHn (n = 0 to 2) 0: No error 1: Error occurred	R
29	ER_ST29	Indicate captured error status for DSMIF1_SCDE0 0: No error 1: Error occurred	R
30	ER_ST30	Indicate captured error status for DSMIF1_SCDE1 0: No error 1: Error occurred	R
31	ER_ST31	Indicate captured error status for DSMIF1_SCDE2 0: No error 1: Error occurred	R

11.2.1.7 PERIERR_STAT1 : Peripheral Error Event Status Register 1

Base address: ICU = 0x8104_8000

Offset address: 0x06C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ER_ST31	ER_ST30	ER_ST29	ER_ST28	ER_ST27	—	—	ER_ST24	ER_ST23	ER_ST22	ER_ST21	ER_ST20	ER_ST19	ER_ST18	ER_ST17	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ER_ST15	—	ER_ST13	—	—	—	—	—	—	ER_ST6	ER_ST5	ER_ST4	ER_ST3	ER_ST2	ER_ST1	ER_ST0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_ST0	Indicate captured error status for DOC_DOPCI 0: No error 1: Error occurred	R
1	ER_ST1	Indicate captured error status for SRAM0_IE1 0: No error 1: Error occurred	R
2	ER_ST2	Indicate captured error status for SRAM0_IE2 0: No error 1: Error occurred	R
3	ER_ST3	Indicate captured error status for SRAM0_OVF 0: No error 1: Error occurred	R
4	ER_ST4	Indicate captured error status for SRAM1_IE1 0: No error 1: Error occurred	R
5	ER_ST5	Indicate captured error status for SRAM1_IE2 0: No error 1: Error occurred	R
6	ER_ST6	Indicate captured error status for SRAM1_OVF 0: No error 1: Error occurred	R
12:7	—	These bits are read as 0.	R
13	ER_ST13	Indicate captured error status for MAIN_BUS_ERRINT 0: No error 1: Error occurred	R
14	—	This bit is read as 0.	R
15	ER_ST15	Indicate captured error status for MPU_SHOSTIF 0: No error 1: Error occurred	R
16	—	This bit is read as 0.	R
17	ER_ST17	Indicate captured error status for MPU_DMACR0 0: No error 1: Error occurred	R
18	ER_ST18	Indicate captured error status for MPU_DMACW0 0: No error 1: Error occurred	R
19	ER_ST19	Indicate captured error status for MPU_DMACR1 0: No error 1: Error occurred	R
20	ER_ST20	Indicate captured error status for MPU_DMACW1 0: No error 1: Error occurred	R
21	ER_ST21	Indicate captured error status for MPU_GMACR 0: No error 1: Error occurred	R
22	ER_ST22	Indicate captured error status for MPU_GMACW 0: No error 1: Error occurred	R
23	ER_ST23	Indicate captured error status for MPU_USBH 0: No error 1: Error occurred	R
24	ER_ST24	Indicate captured error status for MPU_USBF 0: No error 1: Error occurred	R

Bit	Symbol	Function	R/W
26:25	—	These bits are read as 0.	R
27	ER_ST27	Indicate captured error status for MPU_DBGR 0: No error 1: Error occurred	R
28	ER_ST28	Indicate captured error status for MPU_DBGW 0: No error 1: Error occurred	R
29	ER_ST29	Indicate capture error status for DSMIF0_OVC1EL0 0: No error 1: Error occurred	R
30	ER_ST30	Indicate capture error status for DSMIF0_OVC1EH0 0: No error 1: Error occurred	R
31	ER_ST31	Indicate capture error status for DSMIF0_OVC2EL0 0: No error 1: Error occurred	R

11.2.1.8 CPU0ERR_CLR : CPU0 Error Event Status Clear Register

Base address: ICU = 0x8104_8000

Offset address: 0x070

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	ER_CL25	ER_CL24	ER_CL23	ER_CL22	ER_CL21	ER_CL20	ER_CL19	ER_CL18	ER_CL17	ER_CL16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ER_CL15	ER_CL14	ER_CL13	ER_CL12	ER_CL11	ER_CL10	ER_CL9	ER_CL8	ER_CL7	ER_CL6	ER_CL5	ER_CL4	ER_CL3	ER_CL2	ER_CL1	ER_CL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_CL0	Clear captured error status for CPU0ERR_STAT register by writing 1	W
1	ER_CL1	Clear captured error status for CPU0ERR_STAT register by writing 1	W
2	ER_CL2	Clear captured error status for CPU0ERR_STAT register by writing 1	W
3	ER_CL3	Clear captured error status for CPU0ERR_STAT register by writing 1	W
4	ER_CL4	Clear captured error status for CPU0ERR_STAT register by writing 1	W
5	ER_CL5	Clear captured error status for CPU0ERR_STAT register by writing 1	W
6	ER_CL6	Clear captured error status for CPU0ERR_STAT register by writing 1	W
7	ER_CL7	Clear captured error status for CPU0ERR_STAT register by writing 1	W
8	ER_CL8	Clear captured error status for CPU0ERR_STAT register by writing 1	W
9	ER_CL9	Clear captured error status for CPU0ERR_STAT register by writing 1	W
10	ER_CL10	Clear captured error status for CPU0ERR_STAT register by writing 1	W
11	ER_CL11	Clear captured error status for CPU0ERR_STAT register by writing 1	W
12	ER_CL12	Clear captured error status for CPU0ERR_STAT register by writing 1	W
13	ER_CL13	Clear captured error status for CPU0ERR_STAT register by writing 1	W
14	ER_CL14	Clear captured error status for CPU0ERR_STAT register by writing 1	W
15	ER_CL15	Clear captured error status for CPU0ERR_STAT register by writing 1	W
16	ER_CL16	Clear captured error status for CPU0ERR_STAT register by writing 1	W

Bit	Symbol	Function	R/W
17	ER_CL17	Clear captured error status for CPU0ERR_STAT register by writing 1	W
18	ER_CL18	Clear captured error status for CPU0ERR_STAT register by writing 1	W
19	ER_CL19	Clear captured error status for CPU0ERR_STAT register by writing 1	W
20	ER_CL20	Clear captured error status for CPU0ERR_STAT register by writing 1	W
21	ER_CL21	Clear captured error status for CPU0ERR_STAT register by writing 1	W
22	ER_CL22	Clear captured error status for CPU0ERR_STAT register by writing 1	W
23	ER_CL23	Clear captured error status for CPU0ERR_STAT register by writing 1	W
24	ER_CL24	Clear captured error status for CPU0ERR_STAT register by writing 1	W
25	ER_CL25	Clear captured error status for CPU0ERR_STAT register by writing 1	W
31:26	—	The write value should be 0.	W

11.2.1.9 PERIERR_CLR0 : Peripheral Error Event Status Clear Register 0

Base address: ICU = 0x8104_8000

Offset address: 0x078

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ER_CL31	ER_CL30	ER_CL29	ER_CL28	ER_CL27	ER_CL26	ER_CL25	ER_CL24	ER_CL23	ER_CL22	ER_CL21	ER_CL20	ER_CL19	ER_CL18	ER_CL17	ER_CL16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ER_CL15	ER_CL14	ER_CL13	ER_CL12	ER_CL11	ER_CL10	ER_CL9	—	ER_CL7	ER_CL6	ER_CL5	ER_CL4	ER_CL3	ER_CL2	ER_CL1	ER_CL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_CL0	Clear captured error status for PERIERR_STAT0 register by writing 1	W
1	ER_CL1	Clear captured error status for PERIERR_STAT0 register by writing 1	W
2	ER_CL2	Clear captured error status for PERIERR_STAT0 register by writing 1	W
3	ER_CL3	Clear captured error status for PERIERR_STAT0 register by writing 1	W
4	ER_CL4	Clear captured error status for PERIERR_STAT0 register by writing 1	W
5	ER_CL5	Clear captured error status for PERIERR_STAT0 register by writing 1	W
6	ER_CL6	Clear captured error status for PERIERR_STAT0 register by writing 1	W
7	ER_CL7	Clear captured error status for PERIERR_STAT0 register by writing 1	W
8	—	The write value should be 0.	W
9	ER_CL9	Clear captured error status for PERIERR_STAT0 register by writing 1	W
10	ER_CL10	Clear captured error status for PERIERR_STAT0 register by writing 1	W
11	ER_CL11	Clear captured error status for PERIERR_STAT0 register by writing 1	W
12	ER_CL12	Clear captured error status for PERIERR_STAT0 register by writing 1	W
13	ER_CL13	Clear captured error status for PERIERR_STAT0 register by writing 1	W
14	ER_CL14	Clear captured error status for PERIERR_STAT0 register by writing 1	W
15	ER_CL15	Clear captured error status for PERIERR_STAT0 register by writing 1	W
16	ER_CL16	Clear captured error status for PERIERR_STAT0 register by writing 1	W
17	ER_CL17	Clear captured error status for PERIERR_STAT0 register by writing 1	W
18	ER_CL18	Clear captured error status for PERIERR_STAT0 register by writing 1	W

Bit	Symbol	Function	R/W
19	ER_CL19	Clear captured error status for PERIERR_STAT0 register by writing 1	W
20	ER_CL20	Clear captured error status for PERIERR_STAT0 register by writing 1	W
21	ER_CL21	Clear captured error status for PERIERR_STAT0 register by writing 1	W
22	ER_CL22	Clear captured error status for PERIERR_STAT0 register by writing 1	W
23	ER_CL23	Clear captured error status for PERIERR_STAT0 register by writing 1	W
24	ER_CL24	Clear captured error status for PERIERR_STAT0 register by writing 1	W
25	ER_CL25	Clear captured error status for PERIERR_STAT0 register by writing 1	W
26	ER_CL26	Clear captured error status for PERIERR_STAT0 register by writing 1	W
27	ER_CL27	Clear captured error status for PERIERR_STAT0 register by writing 1	W
28	ER_CL28	Clear captured error status for PERIERR_STAT0 register by writing 1	W
29	ER_CL29	Clear captured error status for PERIERR_STAT0 register by writing 1	W
30	ER_CL30	Clear captured error status for PERIERR_STAT0 register by writing 1	W
31	ER_CL31	Clear captured error status for PERIERR_STAT0 register by writing 1	W

11.2.1.10 PERIERR_CLR1 : Peripheral Error Event Status Clear Register 1

Base address: ICU = 0x8104_8000

Offset address: 0x07C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ER_CL31	ER_CL30	ER_CL29	ER_CL28	ER_CL27	—	—	ER_CL24	ER_CL23	ER_CL22	ER_CL21	ER_CL20	ER_CL19	ER_CL18	ER_CL17	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ER_CL15	—	ER_CL13	—	—	—	—	—	—	ER_CL6	ER_CL5	ER_CL4	ER_CL3	ER_CL2	ER_CL1	ER_CL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_CL0	Clear captured error status for PERIERR_STAT1 register by writing 1	W
1	ER_CL1	Clear captured error status for PERIERR_STAT1 register by writing 1	W
2	ER_CL2	Clear captured error status for PERIERR_STAT1 register by writing 1	W
3	ER_CL3	Clear captured error status for PERIERR_STAT1 register by writing 1	W
4	ER_CL4	Clear captured error status for PERIERR_STAT1 register by writing 1	W
5	ER_CL5	Clear captured error status for PERIERR_STAT1 register by writing 1	W
6	ER_CL6	Clear captured error status for PERIERR_STAT1 register by writing 1	W
12:7	—	The write value should be 0.	W
13	ER_CL13	Clear captured error status for PERIERR_STAT1 register by writing 1	W
14	—	The write value should be 0.	W
15	ER_CL15	Clear captured error status for PERIERR_STAT1 register by writing 1	W
16	—	The write value should be 0.	W
17	ER_CL17	Clear captured error status for PERIERR_STAT1 register by writing 1	W
18	ER_CL18	Clear captured error status for PERIERR_STAT1 register by writing 1	W
19	ER_CL19	Clear captured error status for PERIERR_STAT1 register by writing 1	W
20	ER_CL20	Clear captured error status for PERIERR_STAT1 register by writing 1	W

Bit	Symbol	Function	R/W
21	ER_CL21	Clear captured error status for PERIERR_STAT1 register by writing 1	W
22	ER_CL22	Clear captured error status for PERIERR_STAT1 register by writing 1	W
23	ER_CL23	Clear captured error status for PERIERR_STAT1 register by writing 1	W
24	ER_CL24	Clear captured error status for PERIERR_STAT1 register by writing 1	W
26:25	—	The write value should be 0.	W
27	ER_CL27	Clear captured error status for PERIERR_STAT1 register by writing 1	W
28	ER_CL28	Clear captured error status for PERIERR_STAT1 register by writing 1	W
29	ER_CL29	Clear captured error status for PERIERR_STAT1 register by writing 1	W
30	ER_CL30	Clear captured error status for PERIERR_STAT1 register by writing 1	W
31	ER_CL31	Clear captured error status for PERIERR_STAT1 register by writing 1	W

11.2.1.11 CPU0ERR_RSTMSK : CPU0 Error Event Reset Mask Register

Base address: ICU = 0x8104_8000

Offset address: 0x080

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	RS_MK25	RS_MK24	RS_MK23	RS_MK22	RS_MK21	RS_MK20	RS_MK19	RS_MK18	RS_MK17	RS_MK16
Value after reset:	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RS_MK15	RS_MK14	RS_MK13	RS_MK12	RS_MK11	RS_MK10	RS_MK9	RS_MK8	RS_MK7	RS_MK6	RS_MK5	RS_MK4	RS_MK3	RS_MK2	RS_MK1	RS_MK0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	RS_MK0	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
1	RS_MK1	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
2	RS_MK2	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
3	RS_MK3	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
4	RS_MK4	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
5	RS_MK5	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
6	RS_MK6	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
7	RS_MK7	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
8	RS_MK8	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
9	RS_MK9	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
10	RS_MK10	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
11	RS_MK11	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
12	RS_MK12	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
13	RS_MK13	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
14	RS_MK14	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
15	RS_MK15	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
16	RS_MK16	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
17	RS_MK17	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
18	RS_MK18	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
19	RS_MK19	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
20	RS_MK20	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
21	RS_MK21	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
22	RS_MK22	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
23	RS_MK23	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
24	RS_MK24	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
25	RS_MK25	Mask captured error status as a reset event for CPU0ERR_STAT 0: Release 1: Mask	R/W
26	—	This bit is read as 1. The write value should be 1.	R/W

Bit	Symbol	Function	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

11.2.1.12 PERIERR_RSTMSK0 : Peripheral Error Event Reset Mask Register 0

Base address: ICU = 0x8104_8000

Offset address: 0x088

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RS_MK31	RS_MK30	RS_MK29	RS_MK28	RS_MK27	RS_MK26	RS_MK25	RS_MK24	RS_MK23	RS_MK22	RS_MK21	RS_MK20	RS_MK19	RS_MK18	RS_MK17	RS_MK16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RS_MK15	RS_MK14	RS_MK13	RS_MK12	RS_MK11	RS_MK10	RS_MK9	—	RS_MK7	RS_MK6	RS_MK5	RS_MK4	RS_MK3	RS_MK2	RS_MK1	RS_MK0
Value after reset:	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	RS_MK0	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
1	RS_MK1	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
2	RS_MK2	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
3	RS_MK3	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
4	RS_MK4	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
5	RS_MK5	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
6	RS_MK6	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
7	RS_MK7	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
8	—	This bit is read as 0. The write value should be 0.	R/W
9	RS_MK9	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
10	RS_MK10	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
11	RS_MK11	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
12	RS_MK12	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
13	RS_MK13	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
14	RS_MK14	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
15	RS_MK15	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
16	RS_MK16	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
17	RS_MK17	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
18	RS_MK18	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
19	RS_MK19	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
20	RS_MK20	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
21	RS_MK21	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
22	RS_MK22	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
23	RS_MK23	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
24	RS_MK24	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
25	RS_MK25	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
26	RS_MK26	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
27	RS_MK27	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
28	RS_MK28	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
29	RS_MK29	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W
30	RS_MK30	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
31	RS_MK31	Mask captured error status as a reset event for PERIERR_STAT0 0: Release 1: Mask	R/W

11.2.1.13 PERIERR_RSTMSK1 : Peripheral Error Event Reset Mask Register 1

Base address: ICU = 0x8104_8000

Offset address: 0x08C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RS_MK31	RS_MK30	RS_MK29	RS_MK28	RS_MK27	—	—	RS_MK24	RS_MK23	RS_MK22	RS_MK21	RS_MK20	RS_MK19	RS_MK18	RS_MK17	—
Value after reset:	1	1	1	1	1	0	0	1	1	1	1	1	1	1	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RS_MK15	—	RS_MK13	—	—	—	—	—	—	RS_MK6	RS_MK5	RS_MK4	RS_MK3	RS_MK2	RS_MK1	RS_MK0
Value after reset:	1	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	RS_MK0	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W
1	RS_MK1	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W
2	RS_MK2	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W
3	RS_MK3	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W
4	RS_MK4	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W
5	RS_MK5	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W
6	RS_MK6	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W
12:7	—	These bits are read as 0. The write value should be 0.	R/W
13	RS_MK13	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W
14	—	This bit is read as 0. The write value should be 0.	R/W
15	RS_MK15	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W
16	—	This bit is read as 0. The write value should be 0.	R/W
17	RS_MK17	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
18	RS_MK18	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W
19	RS_MK19	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W
20	RS_MK20	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W
21	RS_MK21	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W
22	RS_MK22	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W
23	RS_MK23	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W
24	RS_MK24	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W
26:25	—	These bits are read as 0. The write value should be 0.	R/W
27	RS_MK27	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W
28	RS_MK28	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W
29	RS_MK29	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W
30	RS_MK30	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W
31	RS_MK31	Mask captured error status as a reset event for PERIERR_STAT1 0: Release 1: Mask	R/W

11.2.1.14 CPU0ERR_E0MSK : CPU0 E0 Error Event Mask Register

Base address: ICU = 0x8104_8000

Offset address: 0x090

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	E0_MK25	E0_MK24	E0_MK23	E0_MK22	E0_MK21	E0_MK20	E0_MK19	E0_MK18	E0_MK17	E0_MK16
Value after reset:	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	E0_MK15	E0_MK14	E0_MK13	E0_MK12	E0_MK11	E0_MK10	E0_MK9	E0_MK8	E0_MK7	E0_MK6	E0_MK5	E0_MK4	E0_MK3	E0_MK2	E0_MK1	E0_MK0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E0_MK0	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
1	E0_MK1	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
2	E0_MK2	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
3	E0_MK3	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
4	E0_MK4	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
5	E0_MK5	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
6	E0_MK6	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
7	E0_MK7	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
8	E0_MK8	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
9	E0_MK9	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
10	E0_MK10	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
11	E0_MK11	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
12	E0_MK12	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
13	E0_MK13	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
14	E0_MK14	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
15	E0_MK15	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
16	E0_MK16	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
17	E0_MK17	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
18	E0_MK18	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
19	E0_MK19	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
20	E0_MK20	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
21	E0_MK21	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
22	E0_MK22	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
23	E0_MK23	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
24	E0_MK24	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
25	E0_MK25	Mask captured error status as a CPU0_ERR0 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
26	—	This bit is read as 1. The write value should be 1.	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

11.2.1.15 PERIERR_E0MSK0 : Peripheral E0 Error Event Mask Register 0

Base address: ICU = 0x8104_8000

Offset address: 0x098

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	E0_MK31	E0_MK30	E0_MK29	E0_MK28	E0_MK27	E0_MK26	E0_MK25	E0_MK24	E0_MK23	E0_MK22	E0_MK21	E0_MK20	E0_MK19	E0_MK18	E0_MK17	E0_MK16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	E0_MK15	E0_MK14	E0_MK13	E0_MK12	E0_MK11	E0_MK10	E0_MK9	—	E0_MK7	E0_MK6	E0_MK5	E0_MK4	E0_MK3	E0_MK2	E0_MK1	E0_MK0
Value after reset:	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E0_MK0	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
1	E0_MK1	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
2	E0_MK2	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
3	E0_MK3	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
4	E0_MK4	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
5	E0_MK5	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
6	E0_MK6	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
7	E0_MK7	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
8	—	This bit is read as 0. The write value should be 0.	R/W
9	E0_MK9	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
10	E0_MK10	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
11	E0_MK11	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
12	E0_MK12	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
13	E0_MK13	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
14	E0_MK14	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
15	E0_MK15	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
16	E0_MK16	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
17	E0_MK17	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
18	E0_MK18	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
19	E0_MK19	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
20	E0_MK20	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
21	E0_MK21	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
22	E0_MK22	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
23	E0_MK23	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
24	E0_MK24	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
25	E0_MK25	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
26	E0_MK26	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
27	E0_MK27	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
28	E0_MK28	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
29	E0_MK29	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
30	E0_MK30	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W
31	E0_MK31	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT0 0: Release 1: Mask	R/W

11.2.1.16 PERIERR_E0MSK1 : Peripheral E0 Error Event Mask Register 1

Base address: ICU = 0x8104_8000

Offset address: 0x09C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	E0_MK31	E0_MK30	E0_MK29	E0_MK28	E0_MK27	—	—	E0_MK24	E0_MK23	E0_MK22	E0_MK21	E0_MK20	E0_MK19	E0_MK18	E0_MK17	—
Value after reset:	0	0	0	1	1	0	0	1	1	1	1	1	1	1	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	E0_MK15	—	E0_MK13	—	—	—	—	—	—	E0_MK6	E0_MK5	E0_MK4	E0_MK3	E0_MK2	E0_MK1	E0_MK0
Value after reset:	1	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E0_MK0	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W
1	E0_MK1	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W
2	E0_MK2	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
3	E0_MK3	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W
4	E0_MK4	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W
5	E0_MK5	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W
6	E0_MK6	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W
12:7	—	These bits are read as 0. The write value should be 0.	R/W
13	E0_MK13	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W
14	—	This bit is read as 0. The write value should be 0.	R/W
15	E0_MK15	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W
16	—	This bit is read as 0. The write value should be 0.	R/W
17	E0_MK17	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W
18	E0_MK18	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W
19	E0_MK19	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W
20	E0_MK20	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W
21	E0_MK21	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W
22	E0_MK22	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W
23	E0_MK23	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W
24	E0_MK24	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W
26:25	—	These bits are read as 0. The write value should be 0.	R/W
27	E0_MK27	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W
28	E0_MK28	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
29	E0_MK29	Mask capture error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W
30	E0_MK30	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W
31	E0_MK31	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT1 0: Release 1: Mask	R/W

11.2.1.17 CPU0ERR_E1MSK : CPU0 E1 Error Event Mask Register

Base address: ICU = 0x8104_8000

Offset address: 0x100

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	E1_MK25	E1_MK24	E1_MK23	E1_MK22	E1_MK21	E1_MK20	E1_MK19	E1_MK18	E1_MK17	E1_MK16
Value after reset:	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	E1_MK15	E1_MK14	E1_MK13	E1_MK12	E1_MK11	E1_MK10	E1_MK9	E1_MK8	E1_MK7	E1_MK6	E1_MK5	E1_MK4	E1_MK3	E1_MK2	E1_MK1	E1_MK0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E1_MK0	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
1	E1_MK1	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
2	E1_MK2	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
3	E1_MK3	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
4	E1_MK4	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
5	E1_MK5	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
6	E1_MK6	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
7	E1_MK7	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
8	E1_MK8	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
9	E1_MK9	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
10	E1_MK10	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
11	E1_MK11	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
12	E1_MK12	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
13	E1_MK13	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
14	E1_MK14	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
15	E1_MK15	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
16	E1_MK16	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
17	E1_MK17	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
18	E1_MK18	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
19	E1_MK19	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
20	E1_MK20	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
21	E1_MK21	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
22	E1_MK22	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
23	E1_MK23	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
24	E1_MK24	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
25	E1_MK25	Mask captured error status as a CPU0_ERR1 event for CPU0ERR_STAT 0: Release 1: Mask	R/W
26	—	This bit is read as 1. The write value should be 1.	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

11.2.1.18 PERIERR_E1MSK0 : Peripheral E1 Error Event Mask Register 0

Base address: ICU = 0x8104_8000

Offset address: 0x108

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	E1_MK31	E1_MK30	E1_MK29	E1_MK28	E1_MK27	E1_MK26	E1_MK25	E1_MK24	E1_MK23	E1_MK22	E1_MK21	E1_MK20	E1_MK19	E1_MK18	E1_MK17	E1_MK16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	E1_MK15	E1_MK14	E1_MK13	E1_MK12	E1_MK11	E1_MK10	E1_MK9	—	E1_MK7	E1_MK6	E1_MK5	E1_MK4	E1_MK3	E1_MK2	E1_MK1	E1_MK0
Value after reset:	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E1_MK0	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
1	E1_MK1	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
2	E1_MK2	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
3	E1_MK3	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
4	E1_MK4	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
5	E1_MK5	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
6	E1_MK6	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
7	E1_MK7	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
8	—	This bit is read as 0. The write value should be 0.	R/W
9	E1_MK9	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
10	E1_MK10	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
11	E1_MK11	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
12	E1_MK12	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
13	E1_MK13	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
14	E1_MK14	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
15	E1_MK15	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
16	E1_MK16	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
17	E1_MK17	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
18	E1_MK18	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
19	E1_MK19	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
20	E1_MK20	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
21	E1_MK21	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
22	E1_MK22	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
23	E1_MK23	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
24	E1_MK24	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
25	E1_MK25	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
26	E1_MK26	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
27	E1_MK27	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
28	E1_MK28	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
29	E1_MK29	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
30	E1_MK30	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W
31	E1_MK31	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT0 0: Release 1: Mask	R/W

11.2.1.19 PERIERR_E1MSK1 : Peripheral E1 Error Event Mask Register 1

Base address: ICU = 0x8104_8000

Offset address: 0x10C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	E1_MK31	E1_MK30	E1_MK29	E1_MK28	E1_MK27	—	—	E1_MK24	E1_MK23	E1_MK22	E1_MK21	E1_MK20	E1_MK19	E1_MK18	E1_MK17	—
Value after reset:	0	0	0	1	1	0	0	1	1	1	1	1	1	1	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	E1_MK15	—	E1_MK13	—	—	—	—	—	—	E1_MK6	E1_MK5	E1_MK4	E1_MK3	E1_MK2	E1_MK1	E1_MK0
Value after reset:	1	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E1_MK0	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W
1	E1_MK1	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W
2	E1_MK2	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W
3	E1_MK3	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W
4	E1_MK4	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W
5	E1_MK5	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W
6	E1_MK6	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W
12:7	—	These bits are read as 0. The write value should be 0.	R/W
13	E1_MK13	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W
14	—	This bit is read as 0. The write value should be 0.	R/W
15	E1_MK15	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W
16	—	This bit is read as 0. The write value should be 0.	R/W
17	E1_MK17	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W
18	E1_MK18	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W
19	E1_MK19	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
20	E1_MK20	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W
21	E1_MK21	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W
22	E1_MK22	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W
23	E1_MK23	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W
24	E1_MK24	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W
26:25	—	These bits are read as 0. The write value should be 0.	R/W
27	E1_MK27	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W
28	E1_MK28	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W
29	E1_MK29	Mask capture error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W
30	E1_MK30	Mask capture error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W
31	E1_MK31	Mask capture error status as a PERI_ERR1 event for PERIERR_STAT1 0: Release 1: Mask	R/W

11.2.1.20 PERIERR_STAT2 : Peripheral Error Event Status Register 2

Base address: ICU = 0x8104_8000

Offset address: 0x120

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ER_ST31	ER_ST30	ER_ST29	ER_ST28	ER_ST27	ER_ST26	ER_ST25	ER_ST24	ER_ST23	ER_ST22	ER_ST21	ER_ST20	ER_ST19	ER_ST18	ER_ST17	ER_ST16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ER_ST15	ER_ST14	ER_ST13	ER_ST12	ER_ST11	ER_ST10	ER_ST9	ER_ST8	ER_ST7	ER_ST6	ER_ST5	ER_ST4	ER_ST3	ER_ST2	ER_ST1	ER_ST0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_ST0	Indicate capture error status for DSMIF0_OVC2EH0 0: No error 1: Error occurred	R
1	ER_ST1	Indicate capture error status for DSMIF0_OVC1EL1 0: No error 1: Error occurred	R

Bit	Symbol	Function	R/W
2	ER_ST2	Indicate capture error status for DSMIF0_OVC1EH1 0: No error 1: Error occurred	R
3	ER_ST3	Indicate capture error status for DSMIF0_OVC2EL1 0: No error 1: Error occurred	R
4	ER_ST4	Indicate capture error status for DSMIF0_OVC2EH1 0: No error 1: Error occurred	R
5	ER_ST5	Indicate capture error status for DSMIF0_OVC1EL2 0: No error 1: Error occurred	R
6	ER_ST6	Indicate capture error status for DSMIF0_OVC1EH2 0: No error 1: Error occurred	R
7	ER_ST7	Indicate capture error status for DSMIF0_OVC2EL2 0: No error 1: Error occurred	R
8	ER_ST8	Indicate capture error status for DSMIF0_OVC2EH2 0: No error 1: Error occurred	R
9	ER_ST9	Indicate capture error status for DSMIF0_OVC0WN0 0: No error 1: Error occurred	R
10	ER_ST10	Indicate capture error status for DSMIF0_OVC1WN0 0: No error 1: Error occurred	R
11	ER_ST11	Indicate capture error status for DSMIF0_OVC2WN0 0: No error 1: Error occurred	R
12	ER_ST12	Indicate capture error status for DSMIF0_OVC3WN0 0: No error 1: Error occurred	R
13	ER_ST13	Indicate capture error status for DSMIF0_OVC0WN1 0: No error 1: Error occurred	R
14	ER_ST14	Indicate capture error status for DSMIF0_OVC1WN1 0: No error 1: Error occurred	R
15	ER_ST15	Indicate capture error status for DSMIF0_OVC2WN1 0: No error 1: Error occurred	R
16	ER_ST16	Indicate capture error status for DSMIF0_OVC3WN1 0: No error 1: Error occurred	R
17	ER_ST17	Indicate capture error status for DSMIF0_OVC0WN2 0: No error 1: Error occurred	R
18	ER_ST18	Indicate capture error status for DSMIF0_OVC1WN2 0: No error 1: Error occurred	R
19	ER_ST19	Indicate capture error status for DSMIF0_OVC2WN2 0: No error 1: Error occurred	R

Bit	Symbol	Function	R/W
20	ER_ST20	Indicate capture error status for DSMIF0_OVC3WN2 0: No error 1: Error occurred	R
21	ER_ST21	Indicate capture error status for DSMIF1_OVC1EL0 0: No error 1: Error occurred	R
22	ER_ST22	Indicate capture error status for DSMIF1_OVC1EH0 0: No error 1: Error occurred	R
23	ER_ST23	Indicate capture error status for DSMIF1_OVC2EL0 0: No error 1: Error occurred	R
24	ER_ST24	Indicate capture error status for DSMIF1_OVC2EH0 0: No error 1: Error occurred	R
25	ER_ST25	Indicate capture error status for DSMIF1_OVC1EL1 0: No error 1: Error occurred	R
26	ER_ST26	Indicate capture error status for DSMIF1_OVC1EH1 0: No error 1: Error occurred	R
27	ER_ST27	Indicate capture error status for DSMIF1_OVC2EL1 0: No error 1: Error occurred	R
28	ER_ST28	Indicate capture error status for DSMIF1_OVC2EH1 0: No error 1: Error occurred	R
29	ER_ST29	Indicate capture error status for DSMIF1_OVC1EL2 0: No error 1: Error occurred	R
30	ER_ST30	Indicate capture error status for DSMIF1_OVC1EH2 0: No error 1: Error occurred	R
31	ER_ST31	Indicate capture error status for DSMIF1_OVC2EL2 0: No error 1: Error occurred	R

11.2.1.21 PERIERR_STAT3 : Peripheral Error Event Status Register 3

Base address: ICU = 0x8104_8000

Offset address: 0x124

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	ER_ST30	ER_ST29	ER_ST28	ER_ST27	ER_ST26	ER_ST25	ER_ST24	ER_ST23	ER_ST22	ER_ST21	ER_ST20	ER_ST19	ER_ST18	ER_ST17	ER_ST16
Value after reset:	0	0	0	1	0	0	1	1	1	1	1	0	0	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ER_ST15	ER_ST14	ER_ST13	ER_ST12	ER_ST11	ER_ST10	ER_ST9	ER_ST8	ER_ST7	ER_ST6	ER_ST5	ER_ST4	ER_ST3	ER_ST2	ER_ST1	ER_ST0
Value after reset:	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_ST0	Indicate capture error status for DSMIF1_OVC2EH2 0: No error 1: Error occurred	R
1	ER_ST1	Indicate capture error status for DSMIF1_OVC0WN0 0: No error 1: Error occurred	R
2	ER_ST2	Indicate capture error status for DSMIF1_OVC1WN0 0: No error 1: Error occurred	R
3	ER_ST3	Indicate capture error status for DSMIF1_OVC2WN0 0: No error 1: Error occurred	R
4	ER_ST4	Indicate capture error status for DSMIF1_OVC3WN0 0: No error 1: Error occurred	R
5	ER_ST5	Indicate capture error status for DSMIF1_OVC0WN1 0: No error 1: Error occurred	R
6	ER_ST6	Indicate capture error status for DSMIF1_OVC1WN1 0: No error 1: Error occurred	R
7	ER_ST7	Indicate capture error status for DSMIF1_OVC2WN1 0: No error 1: Error occurred	R
8	ER_ST8	Indicate capture error status for DSMIF1_OVC3WN1 0: No error 1: Error occurred	R
9	ER_ST9	Indicate capture error status for DSMIF1_OVC0WN2 0: No error 1: Error occurred	R
10	ER_ST10	Indicate capture error status for DSMIF1_OVC1WN2 0: No error 1: Error occurred	R
11	ER_ST11	Indicate capture error status for DSMIF1_OVC2WN2 0: No error 1: Error occurred	R
12	ER_ST12	Indicate capture error status for DSMIF1_OVC3WN2 0: No error 1: Error occurred	R
13	ER_ST13	Indicate capture error status for BISS0_NER 0: No error 1: Error occurred	R
14	ER_ST14	Indicate capture error status for BISS1_NER 0: No error 1: Error occurred	R
15	ER_ST15	Indicate capture error status for HDSL0_estimator_on 0: No error 1: Error occurred	R
16	ER_ST16	Indicate capture error status for HDSL0_safe_channel_err 0: No error 1: Error occurred	R
17	ER_ST17	Indicate capture error status for HDSL0_safe_pos_err 0: No error 1: Error occurred	R

Bit	Symbol	Function	R/W
18	ER_ST18	Indicate capture error status for HDSL0_acceleration_err 0: No error 1: Error occurred	R
19	ER_ST19	Indicate capture error status for HDSL0_acc_thr_err 0: No error 1: Error occurred	R
20	ER_ST20	Indicate capture error status for HDSL0_encoding_err 0: No error 1: Error occurred	R
21	ER_ST21	Indicate capture error status for HDSL0_dev_thr_err 0: No error 1: Error occurred	R
22	ER_ST22	Indicate capture error status for HDSL1_estimator_on 0: No error 1: Error occurred	R
23	ER_ST23	Indicate capture error status for HDSL1_safe_channel_err 0: No error 1: Error occurred	R
24	ER_ST24	Indicate capture error status for HDSL1_safe_pos_err 0: No error 1: Error occurred	R
25	ER_ST25	Indicate capture error status for HDSL1_acceleration_err 0: No error 1: Error occurred	R
26	ER_ST26	Indicate capture error status for HDSL1_acc_thr_err 0: No error 1: Error occurred	R
27	ER_ST27	Indicate capture error status for HDSL1_encoding_err 0: No error 1: Error occurred	R
28	ER_ST28	Indicate capture error status for HDSL1_dev_thr_err 0: No error 1: Error occurred	R
29	ER_ST29	Indicate capture error status for AFMT0_TMOUT 0: No error 1: Error occurred	R
30	ER_ST30	Indicate capture error status for AFMT1_TMOUT 0: No error 1: Error occurred	R
31	—	This bit is read as 0.	R

11.2.1.22 PERIERR_CLR2 : Peripheral Error Event Status Clear Register 2

Base address: ICU = 0x8104_8000

Offset address: 0x130

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ER_CL31	ER_CL30	ER_CL29	ER_CL28	ER_CL27	ER_CL26	ER_CL25	ER_CL24	ER_CL23	ER_CL22	ER_CL21	ER_CL20	ER_CL19	ER_CL18	ER_CL17	ER_CL16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ER_CL15	ER_CL14	ER_CL13	ER_CL12	ER_CL11	ER_CL10	ER_CL9	ER_CL8	ER_CL7	ER_CL6	ER_CL5	ER_CL4	ER_CL3	ER_CL2	ER_CL1	ER_CL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_CL0	Clear captured error status for PERIERR_STAT2 register by writing 1	W
1	ER_CL1	Clear captured error status for PERIERR_STAT2 register by writing 1	W
2	ER_CL2	Clear captured error status for PERIERR_STAT2 register by writing 1	W
3	ER_CL3	Clear captured error status for PERIERR_STAT2 register by writing 1	W
4	ER_CL4	Clear captured error status for PERIERR_STAT2 register by writing 1	W
5	ER_CL5	Clear captured error status for PERIERR_STAT2 register by writing 1	W
6	ER_CL6	Clear captured error status for PERIERR_STAT2 register by writing 1	W
7	ER_CL7	Clear captured error status for PERIERR_STAT2 register by writing 1	W
8	ER_CL8	Clear captured error status for PERIERR_STAT2 register by writing 1	W
9	ER_CL9	Clear captured error status for PERIERR_STAT2 register by writing 1	W
10	ER_CL10	Clear captured error status for PERIERR_STAT2 register by writing 1	W
11	ER_CL11	Clear captured error status for PERIERR_STAT2 register by writing 1	W
12	ER_CL12	Clear captured error status for PERIERR_STAT2 register by writing 1	W
13	ER_CL13	Clear captured error status for PERIERR_STAT2 register by writing 1	W
14	ER_CL14	Clear captured error status for PERIERR_STAT2 register by writing 1	W
15	ER_CL15	Clear captured error status for PERIERR_STAT2 register by writing 1	W
16	ER_CL16	Clear captured error status for PERIERR_STAT2 register by writing 1	W
17	ER_CL17	Clear captured error status for PERIERR_STAT2 register by writing 1	W
18	ER_CL18	Clear captured error status for PERIERR_STAT2 register by writing 1	W
19	ER_CL19	Clear captured error status for PERIERR_STAT2 register by writing 1	W
20	ER_CL20	Clear captured error status for PERIERR_STAT2 register by writing 1	W
21	ER_CL21	Clear captured error status for PERIERR_STAT2 register by writing 1	W
22	ER_CL22	Clear captured error status for PERIERR_STAT2 register by writing 1	W
23	ER_CL23	Clear captured error status for PERIERR_STAT2 register by writing 1	W
24	ER_CL24	Clear captured error status for PERIERR_STAT2 register by writing 1	W
25	ER_CL25	Clear captured error status for PERIERR_STAT2 register by writing 1	W
26	ER_CL26	Clear captured error status for PERIERR_STAT2 register by writing 1	W
27	ER_CL27	Clear captured error status for PERIERR_STAT2 register by writing 1	W
28	ER_CL28	Clear captured error status for PERIERR_STAT2 register by writing 1	W
29	ER_CL29	Clear captured error status for PERIERR_STAT2 register by writing 1	W
30	ER_CL30	Clear captured error status for PERIERR_STAT2 register by writing 1	W
31	ER_CL31	Clear captured error status for PERIERR_STAT2 register by writing 1	W

11.2.1.23 PERIERR_CLR3 : Peripheral Error Event Status Clear Register 3

Base address: ICU = 0x8104_8000

Offset address: 0x134

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	ER_CL30	ER_CL29	ER_CL28	ER_CL27	ER_CL26	ER_CL25	ER_CL24	ER_CL23	ER_CL22	ER_CL21	ER_CL20	ER_CL19	ER_CL18	ER_CL17	ER_CL16
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ER_CL15	ER_CL14	ER_CL13	ER_CL12	ER_CL11	ER_CL10	ER_CL9	ER_CL8	ER_CL7	ER_CL6	ER_CL5	ER_CL4	ER_CL3	ER_CL2	ER_CL1	ER_CL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER_CL0	Clear captured error status for PERIERR_STAT3 register by writing 1	W
1	ER_CL1	Clear captured error status for PERIERR_STAT3 register by writing 1	W
2	ER_CL2	Clear captured error status for PERIERR_STAT3 register by writing 1	W
3	ER_CL3	Clear captured error status for PERIERR_STAT3 register by writing 1	W
4	ER_CL4	Clear captured error status for PERIERR_STAT3 register by writing 1	W
5	ER_CL5	Clear captured error status for PERIERR_STAT3 register by writing 1	W
6	ER_CL6	Clear captured error status for PERIERR_STAT3 register by writing 1	W
7	ER_CL7	Clear captured error status for PERIERR_STAT3 register by writing 1	W
8	ER_CL8	Clear captured error status for PERIERR_STAT3 register by writing 1	W
9	ER_CL9	Clear captured error status for PERIERR_STAT3 register by writing 1	W
10	ER_CL10	Clear captured error status for PERIERR_STAT3 register by writing 1	W
11	ER_CL11	Clear captured error status for PERIERR_STAT3 register by writing 1	W
12	ER_CL12	Clear captured error status for PERIERR_STAT3 register by writing 1	W
13	ER_CL13	Clear captured error status for PERIERR_STAT3 register by writing 1	W
14	ER_CL14	Clear captured error status for PERIERR_STAT3 register by writing 1	W
15	ER_CL15	Clear captured error status for PERIERR_STAT3 register by writing 1	W
16	ER_CL16	Clear captured error status for PERIERR_STAT3 register by writing 1	W
17	ER_CL17	Clear captured error status for PERIERR_STAT3 register by writing 1	W
18	ER_CL18	Clear captured error status for PERIERR_STAT3 register by writing 1	W
19	ER_CL19	Clear captured error status for PERIERR_STAT3 register by writing 1	W
20	ER_CL20	Clear captured error status for PERIERR_STAT3 register by writing 1	W
21	ER_CL21	Clear captured error status for PERIERR_STAT3 register by writing 1	W
22	ER_CL22	Clear captured error status for PERIERR_STAT3 register by writing 1	W
23	ER_CL23	Clear captured error status for PERIERR_STAT3 register by writing 1	W
24	ER_CL24	Clear captured error status for PERIERR_STAT3 register by writing 1	W
25	ER_CL25	Clear captured error status for PERIERR_STAT3 register by writing 1	W
26	ER_CL26	Clear captured error status for PERIERR_STAT3 register by writing 1	W
27	ER_CL27	Clear captured error status for PERIERR_STAT3 register by writing 1	W
28	ER_CL28	Clear captured error status for PERIERR_STAT3 register by writing 1	W
29	ER_CL29	Clear captured error status for PERIERR_STAT3 register by writing 1	W
30	ER_CL30	Clear captured error status for PERIERR_STAT3 register by writing 1	W

Bit	Symbol	Function	R/W
31	—	The write value should be 0.	W

11.2.1.24 PERIERR_RSTMSK2 : Peripheral Error Event Reset Mask Register 2

Base address: ICU = 0x8104_8000

Offset address: 0x140

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RS_MK31	RS_MK30	RS_MK29	RS_MK28	RS_MK27	RS_MK26	RS_MK25	RS_MK24	RS_MK23	RS_MK22	RS_MK21	RS_MK20	RS_MK19	RS_MK18	RS_MK17	RS_MK16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RS_MK15	RS_MK14	RS_MK13	RS_MK12	RS_MK11	RS_MK10	RS_MK9	RS_MK8	RS_MK7	RS_MK6	RS_MK5	RS_MK4	RS_MK3	RS_MK2	RS_MK1	RS_MK0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	RS_MK0	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
1	RS_MK1	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
2	RS_MK2	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
3	RS_MK3	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
4	RS_MK4	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
5	RS_MK5	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
6	RS_MK6	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
7	RS_MK7	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
8	RS_MK8	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
9	RS_MK9	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
10	RS_MK10	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
11	RS_MK11	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
12	RS_MK12	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
13	RS_MK13	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
14	RS_MK14	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
15	RS_MK15	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
16	RS_MK16	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
17	RS_MK17	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
18	RS_MK18	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
19	RS_MK19	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
20	RS_MK20	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
21	RS_MK21	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
22	RS_MK22	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
23	RS_MK23	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
24	RS_MK24	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
25	RS_MK25	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
26	RS_MK26	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
27	RS_MK27	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
28	RS_MK28	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
29	RS_MK29	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
30	RS_MK30	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W
31	RS_MK31	Mask captured error status as a reset event for PERIERR_STAT2 0: Release 1: Mask	R/W

11.2.1.25 PERIERR_RSTMSK3 : Peripheral Error Event Reset Mask Register 3

Base address: ICU = 0x8104_8000

Offset address: 0x144

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	RS_MK30	RS_MK29	RS_MK28	RS_MK27	RS_MK26	RS_MK25	RS_MK24	RS_MK23	RS_MK22	RS_MK21	RS_MK20	RS_MK19	RS_MK18	RS_MK17	RS_MK16
Value after reset:	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RS_MK15	RS_MK14	RS_MK13	RS_MK12	RS_MK11	RS_MK10	RS_MK9	RS_MK8	RS_MK7	RS_MK6	RS_MK5	RS_MK4	RS_MK3	RS_MK2	RS_MK1	RS_MK0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	RS_MK0	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
1	RS_MK1	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
2	RS_MK2	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
3	RS_MK3	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
4	RS_MK4	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
5	RS_MK5	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
6	RS_MK6	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
7	RS_MK7	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
8	RS_MK8	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
9	RS_MK9	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
10	RS_MK10	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
11	RS_MK11	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
12	RS_MK12	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
13	RS_MK13	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
14	RS_MK14	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
15	RS_MK15	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
16	RS_MK16	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
17	RS_MK17	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
18	RS_MK18	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
19	RS_MK19	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
20	RS_MK20	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
21	RS_MK21	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
22	RS_MK22	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
23	RS_MK23	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
24	RS_MK24	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
25	RS_MK25	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
26	RS_MK26	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
27	RS_MK27	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
28	RS_MK28	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
29	RS_MK29	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
30	RS_MK30	Mask captured error status as a reset event for PERIERR_STAT3 0: Release 1: Mask	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

11.2.1.26 PERIERR_E0MSK2 : Peripheral E0 Error Event Mask Register 2

Base address: ICU = 0x8104_8000

Offset address: 0x150

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	E0_MK31	E0_MK30	E0_MK29	E0_MK28	E0_MK27	E0_MK26	E0_MK25	E0_MK24	E0_MK23	E0_MK22	E0_MK21	E0_MK20	E0_MK19	E0_MK18	E0_MK17	E0_MK16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	E0_MK15	E0_MK14	E0_MK13	E0_MK12	E0_MK11	E0_MK10	E0_MK9	E0_MK8	E0_MK7	E0_MK6	E0_MK5	E0_MK4	E0_MK3	E0_MK2	E0_MK1	E0_MK0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E0_MK0	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
1	E0_MK1	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
2	E0_MK2	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
3	E0_MK3	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
4	E0_MK4	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
5	E0_MK5	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
6	E0_MK6	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
7	E0_MK7	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
8	E0_MK8	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
9	E0_MK9	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
10	E0_MK10	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
11	E0_MK11	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
12	E0_MK12	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
13	E0_MK13	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
14	E0_MK14	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
15	E0_MK15	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
16	E0_MK16	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
17	E0_MK17	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
18	E0_MK18	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
19	E0_MK19	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
20	E0_MK20	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
21	E0_MK21	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
22	E0_MK22	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
23	E0_MK23	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
24	E0_MK24	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
25	E0_MK25	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
26	E0_MK26	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
27	E0_MK27	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
28	E0_MK28	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
29	E0_MK29	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
30	E0_MK30	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W
31	E0_MK31	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT2 0: Release 1: Mask	R/W

11.2.1.27 PERIERR_E0MSK3 : Peripheral E0 Error Event Mask Register 3

Base address: ICU = 0x8104_8000

Offset address: 0x154

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	E0_MK30	E0_MK29	E0_MK28	E0_MK27	E0_MK26	E0_MK25	E0_MK24	E0_MK23	E0_MK22	E0_MK21	E0_MK20	E0_MK19	E0_MK18	E0_MK17	E0_MK16
Value after reset:	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	E0_MK15	E0_MK14	E0_MK13	E0_MK12	E0_MK11	E0_MK10	E0_MK9	E0_MK8	E0_MK7	E0_MK6	E0_MK5	E0_MK4	E0_MK3	E0_MK2	E0_MK1	E0_MK0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E0_MK0	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
1	E0_MK1	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
2	E0_MK2	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
3	E0_MK3	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
4	E0_MK4	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
5	E0_MK5	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
6	E0_MK6	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
7	E0_MK7	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
8	E0_MK8	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
9	E0_MK9	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
10	E0_MK10	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
11	E0_MK11	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
12	E0_MK12	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
13	E0_MK13	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
14	E0_MK14	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
15	E0_MK15	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
16	E0_MK16	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
17	E0_MK17	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
18	E0_MK18	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
19	E0_MK19	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
20	E0_MK20	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
21	E0_MK21	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
22	E0_MK22	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
23	E0_MK23	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
24	E0_MK24	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
25	E0_MK25	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
26	E0_MK26	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
27	E0_MK27	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
28	E0_MK28	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
29	E0_MK29	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
30	E0_MK30	Mask captured error status as a PERI_ERR0 event for PERIERR_STAT3 0: Release 1: Mask	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

11.2.1.28 PERIERR_E1MSK2 : Peripheral E1 Error Event Mask Register 2

Base address: ICU = 0x8104_8000

Offset address: 0x1C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	E1_MK31	E1_MK30	E1_MK29	E1_MK28	E1_MK27	E1_MK26	E1_MK25	E1_MK24	E1_MK23	E1_MK22	E1_MK21	E1_MK20	E1_MK19	E1_MK18	E1_MK17	E1_MK16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	E1_MK15	E1_MK14	E1_MK13	E1_MK12	E1_MK11	E1_MK10	E1_MK9	E1_MK8	E1_MK7	E1_MK6	E1_MK5	E1_MK4	E1_MK3	E1_MK2	E1_MK1	E1_MK0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E1_MK0	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
1	E1_MK1	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
2	E1_MK2	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
3	E1_MK3	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
4	E1_MK4	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
5	E1_MK5	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
6	E1_MK6	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
7	E1_MK7	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
8	E1_MK8	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
9	E1_MK9	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
10	E1_MK10	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
11	E1_MK11	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
12	E1_MK12	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
13	E1_MK13	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
14	E1_MK14	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
15	E1_MK15	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
16	E1_MK16	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
17	E1_MK17	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
18	E1_MK18	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
19	E1_MK19	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
20	E1_MK20	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
21	E1_MK21	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
22	E1_MK22	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
23	E1_MK23	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
24	E1_MK24	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
25	E1_MK25	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
26	E1_MK26	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
27	E1_MK27	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
28	E1_MK28	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
29	E1_MK29	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
30	E1_MK30	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W
31	E1_MK31	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT2 0: Release 1: Mask	R/W

11.2.1.29 PERIERR_E1MSK3 : Peripheral E1 Error Event Mask Register 3

Base address: ICU = 0x8104_8000

Offset address: 0x1C4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	E1_MK30	E1_MK29	E1_MK28	E1_MK27	E1_MK26	E1_MK25	E1_MK24	E1_MK23	E1_MK22	E1_MK21	E1_MK20	E1_MK19	E1_MK18	E1_MK17	E1_MK16
Value after reset:	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	E1_MK15	E1_MK14	E1_MK13	E1_MK12	E1_MK11	E1_MK10	E1_MK9	E1_MK8	E1_MK7	E1_MK6	E1_MK5	E1_MK4	E1_MK3	E1_MK2	E1_MK1	E1_MK0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	E1_MK0	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
1	E1_MK1	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
2	E1_MK2	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
3	E1_MK3	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
4	E1_MK4	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
5	E1_MK5	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
6	E1_MK6	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
7	E1_MK7	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
8	E1_MK8	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
9	E1_MK9	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
10	E1_MK10	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
11	E1_MK11	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
12	E1_MK12	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
13	E1_MK13	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
14	E1_MK14	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
15	E1_MK15	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
16	E1_MK16	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
17	E1_MK17	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
18	E1_MK18	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
19	E1_MK19	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
20	E1_MK20	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
21	E1_MK21	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W

Bit	Symbol	Function	R/W
22	E1_MK22	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
23	E1_MK23	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
24	E1_MK24	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
25	E1_MK25	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
26	E1_MK26	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
27	E1_MK27	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
28	E1_MK28	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
29	E1_MK29	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
30	E1_MK30	Mask captured error status as a PERI_ERR1 event for PERIERR_STAT3 0: Release 1: Mask	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

11.2.2 Interrupt Control Unit Register 1 (Non-Safety Region)

11.2.2.1 NS_SWINT : Software Interrupt Register

Base address: ICU_NS = 0x8009_0000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	IC5	IC4	IC3	IC2	IC1	IC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IC0	Software Interrupt register Occur software interrupt INTCPU0 when this value is set to 1. This bit is read as 0.	W
1	IC1	Software Interrupt register Occur software interrupt INTCPU1 when this value is set to 1. This bit is read as 0.	W

Bit	Symbol	Function	R/W
2	IC2	Software Interrupt register Occur software interrupt INTCPU2 when this value is set to 1. This bit is read as 0.	W
3	IC3	Software Interrupt register Occur software interrupt INTCPU3 when this value is set to 1. This bit is read as 0.	W
4	IC4	Software Interrupt register Occur software interrupt INTCPU4 when this value is set to 1. This bit is read as 0.	W
5	IC5	Software Interrupt register Occur software interrupt INTCPU5 when this value is set to 1. This bit is read as 0.	W
31:6	—	The write value should be 0.	W

11.2.2.2 NS_PORTNF_FLTSEL : Interrupt Noise Filter Enable Register

Base address: ICU_NS = 0x8009_0000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	FLTDRQ	FLT13	FLT12	FLT11	FLT10	FLT9	FLT8	FLT7	FLT6	FLT5	FLT4	FLT3	FLT2	FLT1	FLT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FLT0	Noise filter enable for IRQ0 0: Disable 1: Enable	R/W
1	FLT1	Noise filter enable for IRQ1 0: Disable 1: Enable	R/W
2	FLT2	Noise filter enable for IRQ2 0: Disable 1: Enable	R/W
3	FLT3	Noise filter enable for IRQ3 0: Disable 1: Enable	R/W
4	FLT4	Noise filter enable for IRQ4 0: Disable 1: Enable	R/W
5	FLT5	Noise filter enable for IRQ5 0: Disable 1: Enable	R/W
6	FLT6	Noise filter enable for IRQ6 0: Disable 1: Enable	R/W
7	FLT7	Noise filter enable for IRQ7 0: Disable 1: Enable	R/W

Bit	Symbol	Function	R/W
8	FLT8	Noise filter enable for IRQ8 0: Disable 1: Enable	R/W
9	FLT9	Noise filter enable for IRQ9 0: Disable 1: Enable	R/W
10	FLT10	Noise filter enable for IRQ10 0: Disable 1: Enable	R/W
11	FLT11	Noise filter enable for IRQ11 0: Disable 1: Enable	R/W
12	FLT12	Noise filter enable for IRQ12 0: Disable 1: Enable	R/W
13	FLT13	Noise filter enable for IRQ13 0: Disable 1: Enable	R/W
14	FLTDRQ	Noise filter enable for External DMA request (DREQ) 0: Disable 1: Enable	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

11.2.2.3 NS_PORTNF_CLKSEL : Interrupt Noise Filter Setting Register

Base address: ICU_NS = 0x8009_0000

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	CKSELDREQ[1:0]		CKSEL13[1:0]		CKSEL12[1:0]		CKSEL11[1:0]		CKSEL10[1:0]		CKSEL9[1:0]		CKSEL8[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CKSEL7[1:0]		CKSEL6[1:0]		CKSEL5[1:0]		CKSEL4[1:0]		CKSEL3[1:0]		CKSEL2[1:0]		CKSEL1[1:0]		CKSEL0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CKSEL0[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ0. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
3:2	CKSEL1[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ1. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W

Bit	Symbol	Function	R/W
5:4	CKSEL2[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ2. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
7:6	CKSEL3[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ3. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
9:8	CKSEL4[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ4. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
11:10	CKSEL5[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ5. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
13:12	CKSEL6[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ6. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
15:14	CKSEL7[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ7. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
17:16	CKSEL8[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ8. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
19:18	CKSEL9[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ9. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W

Bit	Symbol	Function	R/W
21:20	CKSEL10[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ10. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
23:22	CKSEL11[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ11. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
25:24	CKSEL12[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ12. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
27:26	CKSEL13[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for IRQ13. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
29:28	CKSELDREQ[1:0]	Noise filter sampling clock selector Select noise filter sampling frequency dividend rate for DREQ of DMAC. Sampling clock is generated by divided PCLKM. 0 0: Divided by 1 0 1: Divided by 8 1 0: Divided by 32 1 1: Divided by 64	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

11.2.2.4 NS_PORTNF_MD : Interrupt Edge Detection Setting Register

Base address: ICU_NS = 0x8009_0000

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	MDDRQ[1:0]		MD13[1:0]		MD12[1:0]		MD11[1:0]		MD10[1:0]		MD9[1:0]		MD8[1:0]	
Value after reset:	0	0	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MD7[1:0]		MD6[1:0]		MD5[1:0]		MD4[1:0]		MD3[1:0]		MD2[1:0]		MD1[1:0]		MD0[1:0]	
Value after reset:	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

Bit	Symbol	Function	R/W
1:0	MD0[1:0]	Select detection mode for IRQ0 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W

Bit	Symbol	Function	R/W
3:2	MD1[1:0]	Select detection mode for IRQ1 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
5:4	MD2[1:0]	Select detection mode for IRQ2 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
7:6	MD3[1:0]	Select detection mode for IRQ3 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
9:8	MD4[1:0]	Select detection mode for IRQ4 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
11:10	MD5[1:0]	Select detection mode for IRQ5 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
13:12	MD6[1:0]	Select detection mode for IRQ6 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
15:14	MD7[1:0]	Select detection mode for IRQ7 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
17:16	MD8[1:0]	Select detection mode for IRQ8 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
19:18	MD9[1:0]	Select detection mode for IRQ9 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
21:20	MD10[1:0]	Select detection mode for IRQ10 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
23:22	MD11[1:0]	Select detection mode for IRQ11 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W

Bit	Symbol	Function	R/W
25:24	MD12[1:0]	Select detection mode for IRQ12 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
27:26	MD13[1:0]	Select detection mode for IRQ13 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
29:28	MDDRQ[1:0]	Select detection mode for DREQ of DMAC 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Falling and rising edges	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

11.3 Operations

11.3.1 Event Handling

Events and interrupts from internal modules and external pins include the external DMA request pins are connected to the CPU (Cortex-R52 CPU0) and/or the DMAC and ELC as an activating trigger signal. Table 11.2 shows event table of this device. The checked event in the CPU0 GIC Request (SPI)*1, DMAC Activation Request, and ELC Source columns indicates that there is a connection to the modules. Event number is common to SPI number of the GIC, DMACn_RSSELi register (n = 0, 1 and i = 0 to 5), and ELC_SSELn register (n = 0 to 18).

Note 1. GIC request number shows Shared Peripheral interrupts (SPIs) number of the interrupt interface for the Cortex-R52 processor. Generic Interrupt Controller (GIC) handles these interrupts.

Regarding to GIC, please refer to the Arm® Cortex®-R52 Processor Technical Reference Manual.

11.3.2 Event Table

Table 11.2 shows event table of this device.

Table 11.2 Event table (1 of 17)

Module	UNIT /CH	Name	Interrupt sources	Interrupt type	Event number	CPU0 GIC request (SPI)	DMAC activation request	ELC source
ICU	—	INTCPU0	Software interrupt 0	Edge	0 (0x000)	✓	✓	✓
		INTCPU1	Software interrupt 1	Edge	1 (0x001)	✓	✓	✓
		INTCPU2	Software interrupt 2	Edge	2 (0x002)	✓	✓	✓
		INTCPU3	Software interrupt 3	Edge	3 (0x003)	✓	✓	✓
		INTCPU4	Software interrupt 4	Edge	4 (0x004)	✓	✓	✓
		INTCPU5	Software interrupt 5	Edge	5 (0x005)	✓	✓	✓
		INTCPU6	Software interrupt 6	Edge	392 (0x188)	✓	—	—
		INTCPU7	Software interrupt 7	Edge	393 (0x189)	✓	—	—
		NMI	Non-maskable interrupt	Level/Edge	—	SEI/VSEI*1	—	—
		IRQ0	External pin interrupt 0	Level/Edge	6 (0x006)	✓	✓	—
		IRQ1	External pin interrupt 1	Level/Edge	7 (0x007)	✓	✓	—
		IRQ2	External pin interrupt 2	Level/Edge	8 (0x008)	✓	✓	—
		IRQ3	External pin interrupt 3	Level/Edge	9 (0x009)	✓	✓	—
		IRQ4	External pin interrupt 4	Level/Edge	10 (0x00A)	✓	✓	—
		IRQ5	External pin interrupt 5	Level/Edge	11 (0x00B)	✓	✓	—
		IRQ6	External pin interrupt 6	Level/Edge	12 (0x00C)	✓	✓	—
		IRQ7	External pin interrupt 7	Level/Edge	13 (0x00D)	✓	✓	—
		IRQ8	External pin interrupt 8	Level/Edge	14 (0x00E)	✓	✓	—
		IRQ9	External pin interrupt 9	Level/Edge	15 (0x00F)	✓	✓	—
		IRQ10	External pin interrupt 10	Level/Edge	16 (0x010)	✓	✓	—
		IRQ11	External pin interrupt 11	Level/Edge	17 (0x011)	✓	✓	—
		IRQ12	External pin interrupt 12	Level/Edge	18 (0x012)	✓	✓	—
		IRQ13	External pin interrupt 13	Level/Edge	19 (0x013)	✓	✓	—
		IRQ14	External pin interrupt 14	Level/Edge	394 (0x18A)	✓	—	—
		IRQ15	External pin interrupt 15	Level/Edge	395 (0x18B)	✓	—	—
		DREQ	External DMA request	Level/Edge	448 (0x1C0)	—	✓	—
BSC	CH0	BSC_CMI	Refresh compare match interrupt	Level	20 (0x014)	✓	—	—
DMAC	UNIT0	DMAC0_INT0	DMAC0 transfer completion 0	Level	21 (0x015)	✓	—	✓
		DMAC0_INT1	DMAC0 transfer completion 1	Level	22 (0x016)	✓	—	✓
		DMAC0_INT2	DMAC0 transfer completion 2	Level	23 (0x017)	✓	—	✓
		DMAC0_INT3	DMAC0 transfer completion 3	Level	24 (0x018)	✓	—	✓
		DMAC0_INT4	DMAC0 transfer completion 4	Level	25 (0x019)	✓	—	✓
		DMAC0_INT5	DMAC0 transfer completion 5	Level	26 (0x01A)	✓	—	✓
		DMAC0_INT6	DMAC0 transfer completion 6	Level	27 (0x01B)	✓	—	✓
		DMAC0_INT7	DMAC0 transfer completion 7	Level	28 (0x01C)	✓	—	✓
		DMAC0_INT8	DMAC0 transfer completion 8	Level	29 (0x01D)	✓	—	—
		DMAC0_INT9	DMAC0 transfer completion 9	Level	30 (0x01E)	✓	—	—
		DMAC0_INT10	DMAC0 transfer completion 10	Level	31 (0x01F)	✓	—	—
		DMAC0_INT11	DMAC0 transfer completion 11	Level	32 (0x020)	✓	—	—
		DMAC0_INT12	DMAC0 transfer completion 12	Level	33 (0x021)	✓	—	—
		DMAC0_INT13	DMAC0 transfer completion 13	Level	34 (0x022)	✓	—	—
		DMAC0_INT14	DMAC0 transfer completion 14	Level	35 (0x023)	✓	—	—
		DMAC0_INT15	DMAC0 transfer completion 15	Level	36 (0x024)	✓	—	—

Table 11.2 Event table (2 of 17)

Module	UNIT /CH	Name	Interrupt sources	Interrupt type	Event number	CPU0 GIC request (SPI)	DMAC activation request	ELC source
DMAC	UNIT1	DMAC1_INT0	DMAC1 transfer completion 0	Level	37 (0x025)	✓	—	✓
		DMAC1_INT1	DMAC1 transfer completion 1	Level	38 (0x026)	✓	—	✓
		DMAC1_INT2	DMAC1 transfer completion 2	Level	39 (0x027)	✓	—	✓
		DMAC1_INT3	DMAC1 transfer completion 3	Level	40 (0x028)	✓	—	✓
		DMAC1_INT4	DMAC1 transfer completion 4	Level	41 (0x029)	✓	—	✓
		DMAC1_INT5	DMAC1 transfer completion 5	Level	42 (0x02A)	✓	—	✓
		DMAC1_INT6	DMAC1 transfer completion 6	Level	43 (0x02B)	✓	—	✓
		DMAC1_INT7	DMAC1 transfer completion 7	Level	44 (0x02C)	✓	—	✓
		DMAC1_INT8	DMAC1 transfer completion 8	Level	45 (0x02D)	✓	—	—
		DMAC1_INT9	DMAC1 transfer completion 9	Level	46 (0x02E)	✓	—	—
		DMAC1_INT10	DMAC1 transfer completion 10	Level	47 (0x02F)	✓	—	—
		DMAC1_INT11	DMAC1 transfer completion 11	Level	48 (0x030)	✓	—	—
		DMAC1_INT12	DMAC1 transfer completion 12	Level	49 (0x031)	✓	—	—
		DMAC1_INT13	DMAC1 transfer completion 13	Level	50 (0x032)	✓	—	—
		DMAC1_INT14	DMAC1 transfer completion 14	Level	51 (0x033)	✓	—	—
		DMAC1_INT15	DMAC1 transfer completion 15	Level	52 (0x034)	✓	—	—
I/O	—	IOPORT_GROUP1	Input edge detection of input port group 1	Edge	467 (0x1D3)	—	—	✓
		IOPORT_GROUP2	Input edge detection of input port group 2	Edge	468 (0x1D4)	—	—	✓
		IOPORT_SINGLE0	Input edge detection of single input port 0	Edge	469 (0x1D5)	—	—	✓
		IOPORT_SINGLE1	Input edge detection of single input port 1	Edge	470 (0x1D6)	—	—	✓
		IOPORT_SINGLE2	Input edge detection of single input port 2	Edge	471 (0x1D7)	—	—	✓
		IOPORT_SINGLE3	Input edge detection of single input port 3	Edge	472 (0x1D8)	—	—	✓
CMT	CH0	CMT0_CMI	CMT0 Compare match	Edge	53 (0x035)	✓	✓	—
	CH1	CMT1_CMI	CMT1 Compare match	Edge	54 (0x036)	✓	✓	—
	CH2	CMT2_CMI	CMT2 Compare match	Edge	55 (0x037)	✓	✓	—
	CH3	CMT3_CMI	CMT3 Compare match	Edge	56 (0x038)	✓	✓	—
	CH4	CMT4_CMI	CMT4 Compare match	Edge	57 (0x039)	✓	✓	—
	CH5	CMT5_CMI	CMT5 Compare match	Edge	58 (0x03A)	✓	✓	—
CMTW	CH0	CMTW0_CMWI	CMTW0 Compare match	Edge	59 (0x03B)	✓	✓	—
		CMTW0_IC0I	CMTW0 Input capture of register 0	Edge	60 (0x03C)	✓	✓	—
		CMTW0_IC1I	CMTW0 Input capture of register 1	Edge	61 (0x03D)	✓	✓	—
		CMTW0_OC0I	CMTW0 Output compare of register 0	Edge	62 (0x03E)	✓	✓	—
		CMTW0_OC1I	CMTW0 Output compare of register 1	Edge	63 (0x03F)	✓	✓	—

Table 11.2 Event table (3 of 17)

Module	UNIT /CH	Name	Interrupt sources	Interrupt type	Event number	CPU0 GIC request (SPI)	DMAC activation request	ELC source
CMTW	CH1	CMTW1_CMWI	CMTW1 Compare match	Edge	64 (0x040)	✓	✓	—
		CMTW1_IC0I	CMTW1 Input capture of register 0	Edge	65 (0x041)	✓	✓	—
		CMTW1_IC1I	CMTW1 Input capture of register 1	Edge	66 (0x042)	✓	✓	—
		CMTW1_OC0I	CMTW1 Output compare of register 0	Edge	67 (0x043)	✓	✓	—
		CMTW1_OC1I	CMTW1 Output compare of register 1	Edge	68 (0x044)	✓	✓	—
MTU3	CH0	TGIA0	MTU0.TGRA input capture/ compare match	Edge	69 (0x045)	✓	✓	✓
		TGIB0	MTU0.TGRB input capture/ compare match	Edge	70 (0x046)	✓	✓	✓
		TGIC0	MTU0.TGRC input capture/ compare match	Edge	71 (0x047)	✓	✓	✓
		TGID0	MTU0.TGRD input capture/ compare match	Edge	72 (0x048)	✓	✓	✓
		TCIV0	MTU0.TCNT overflow	Edge	73 (0x049)	✓	—	✓
		TGIE0	MTU0.TGRE compare match	Edge	74 (0x04A)	✓	—	✓
		TGIF0	MTU0.TGRF compare match	Edge	75 (0x04B)	✓	—	✓
MTU3	CH1	TGIA1	MTU1.TGRA input capture/ compare match	Edge	76 (0x04C)	✓	✓	—
		TGIB1	MTU1.TGRB input capture/ compare match	Edge	77 (0x04D)	✓	✓	—
		TCIV1	MTU1.TCNT overflow	Edge	78 (0x04E)	✓	—	—
		TCIU1	MTU1.TCNT underflow	Edge	79 (0x04F)	✓	—	—
MTU3	CH2	TGIA2	MTU2.TGRA input capture/ compare match	Edge	80 (0x050)	✓	✓	—
		TGIB2	MTU2.TGRB input capture/ compare match	Edge	81 (0x051)	✓	✓	—
		TCIV2	MTU2.TCNT overflow	Edge	82 (0x052)	✓	—	—
		TCIU2	MTU2.TCNT underflow	Edge	83 (0x053)	✓	—	—
MTU3	CH3	TGIA3	MTU3.TGRA input capture/ compare match	Edge	84 (0x054)	✓	✓	✓
		TGIB3	MTU3.TGRB input capture/ compare match	Edge	85 (0x055)	✓	✓	✓
		TGIC3	MTU3.TGRC input capture/ compare match	Edge	86 (0x056)	✓	✓	✓
		TGID3	MTU3.TGRD input capture/ compare match	Edge	87 (0x057)	✓	✓	✓
		TCIV3	MTU3.TCNT overflow	Edge	88 (0x058)	✓	—	✓
MTU3	CH4	TGIA4	MTU4.TGRA input capture/ compare match	Edge	89 (0x059)	✓	✓	✓
		TGIB4	MTU4.TGRB input capture/ compare match	Edge	90 (0x05A)	✓	✓	✓
		TGIC4	MTU4.TGRC input capture/ compare match	Edge	91 (0x05B)	✓	✓	✓
		TGID4	MTU4.TGRD input capture/ compare match	Edge	92 (0x05C)	✓	✓	✓
		TCIV4	MTU4.TCNT overflow/underflow	Edge	93 (0x05D)	✓	✓	—
		TCIV4_OF	MTU4.TCNT overflow	Edge	463 (0x1CF)	—	—	✓
		TCIV4_UF	MTU4.TCNT underflow	Edge	464 (0x1D0)	—	—	✓

Table 11.2 Event table (4 of 17)

Module	UNIT /CH	Name	Interrupt sources	Interrupt type	Event number	CPU0 GIC request (SPI)	DMAC activation request	ELC source
MTU3	CH5	TGIU5	MTU5.TGRU input capture/ compare match	Edge	94 (0x05E)	✓	✓	—
		TGIV5	MTU5.TGRV input capture/ compare match	Edge	95 (0x05F)	✓	✓	—
		TGIW5	MTU5.TGRW input capture/ compare match	Edge	96 (0x060)	✓	✓	—
MTU3	CH6	TGIA6	MTU6.TGRA input capture/ compare match	Edge	97 (0x061)	✓	✓	✓
		TGIB6	MTU6.TGRB input capture/ compare match	Edge	98 (0x062)	✓	✓	✓
		TGIC6	MTU6.TGRC input capture/ compare match	Edge	99 (0x063)	✓	✓	✓
		TGID6	MTU6.TGRD input capture/ compare match	Edge	100 (0x064)	✓	✓	✓
		TCIV6	MTU6.TCNT overflow	Edge	101 (0x065)	✓	—	✓
MTU3	CH7	TGIA7	MTU7.TGRA input capture/ compare match	Edge	102 (0x066)	✓	✓	✓
		TGIB7	MTU7.TGRB input capture/ compare match	Edge	103 (0x067)	✓	✓	✓
		TGIC7	MTU7.TGRC input capture/ compare match	Edge	104 (0x068)	✓	✓	✓
		TGID7	MTU7.TGRD input capture/ compare match	Edge	105 (0x069)	✓	✓	✓
		TCIV7	MTU7.TCNT overflow/underflow	Edge	106 (0x06A)	✓	✓	—
		TCIV7_OF	MTU7.TCNT overflow	Edge	465 (0x1D1)	—	—	✓
		TCIV7_UF	MTU7.TCNT underflow	Edge	466 (0x1D2)	—	—	✓
MTU3	CH8	TGIA8	MTU8.TGRA input capture/ compare match	Edge	107 (0x06B)	✓	✓	—
		TGIB8	MTU8.TGRB input capture/ compare match	Edge	108 (0x06C)	✓	✓	—
		TGIC8	MTU8.TGRC input capture/ compare match	Edge	109 (0x06D)	✓	✓	—
		TGID8	MTU8.TGRD input capture/ compare match	Edge	110 (0x06E)	✓	✓	—
		TCIV8	MTU8.TCNT overflow	Edge	111 (0x06F)	✓	—	—
POE3	—	OEI1	Output enable interrupt 1	Level	112 (0x070)	✓	—	—
		OEI2	Output enable interrupt 2	Level	113 (0x071)	✓	—	—
		OEI3	Output enable interrupt 3	Level	114 (0x072)	✓	—	—
		OEI4	Output enable interrupt 4	Level	115 (0x073)	✓	—	—

Table 11.2 Event table (5 of 17)

Module	UNIT /CH	Name	Interrupt sources	Interrupt type	Event number	CPU0 GIC request (SPI)	DMAC activation request	ELC source
GPT (LLPP)	CH0	GPT0_CCMPA	GPT0 GTCCRA input capture/ compare match	Edge	116 (0x074)	✓	✓	✓
		GPT0_CCMPB	GPT0 GTCCRB input capture/ compare match	Edge	117 (0x075)	✓	✓	✓
		GPT0_CMPC	GPT0 GTCCRC compare match	Edge	118 (0x076)	✓	✓	✓
		GPT0_CMPD	GPT0 GTCCRD compare match	Edge	119 (0x077)	✓	✓	✓
		GPT0_CMPE	GPT0 GTCCRE compare match	Edge	120 (0x078)	✓	✓	✓
		GPT0_CMPF	GPT0 GTCCRF compare match	Edge	121 (0x079)	✓	✓	✓
		GPT0_OVF	GPT0 GTCNT overflow (GTPR compare match)	Edge	122 (0x07A)	✓	✓	✓
		GPT0_UDF	GPT0 GTCNT underflow	Edge	123 (0x07B)	✓	✓	✓
		GPT0_DTE	GPT0 Dead time error	Edge	124 (0x07C)	✓	—	—
		GPT0_ADTRGA	GPT0 GTADTRA compare match (A/D Converter Start Request A)	Edge	473 (0x1D9)	—	—	✓
		GPT0_ADTRGB	GPT0 GTADTRB compare match (A/D Converter Start Request B)	Edge	474 (0x1DA)	—	—	✓
GPT (LLPP)	CH1	GPT1_CCMPA	GPT1 GTCCRA input capture/ compare match	Edge	125 (0x07D)	✓	✓	✓
		GPT1_CCMPB	GPT1 GTCCRB input capture/ compare match	Edge	126 (0x07E)	✓	✓	✓
		GPT1_CMPC	GPT1 GTCCRC compare match	Edge	127 (0x07F)	✓	✓	✓
		GPT1_CMPD	GPT1 GTCCRD compare match	Edge	128 (0x080)	✓	✓	✓
		GPT1_CMPE	GPT1 GTCCRE compare match	Edge	129 (0x081)	✓	✓	✓
		GPT1_CMPF	GPT1 GTCCRF compare match	Edge	130 (0x082)	✓	✓	✓
		GPT1_OVF	GPT1 GTCNT overflow (GTPR compare match)	Edge	131 (0x083)	✓	✓	✓
		GPT1_UDF	GPT1 GTCNT underflow	Edge	132 (0x084)	✓	✓	✓
		GPT1_DTE	GPT1 Dead time error	Edge	133 (0x085)	✓	—	—
		GPT1_ADTRGA	GPT1 GTADTRA compare match (A/D Converter Start Request A)	Edge	475 (0x1DB)	—	—	✓
		GPT1_ADTRGB	GPT1 GTADTRB compare match (A/D Converter Start Request B)	Edge	476 (0x1DC)	—	—	✓

Table 11.2 Event table (6 of 17)

Module	UNIT /CH	Name	Interrupt sources	Interrupt type	Event number	CPU0 GIC request (SPI)	DMAC activation request	ELC source
GPT (LLPP)	CH2	GPT2_CCMPA	GPT2 GTCCRA input capture/ compare match	Edge	134 (0x086)	✓	✓	✓
		GPT2_CCMPB	GPT2 GTCCRB input capture/ compare match	Edge	135 (0x087)	✓	✓	✓
		GPT2_CMPC	GPT2 GTCCRC compare match	Edge	136 (0x088)	✓	✓	✓
		GPT2_CMPD	GPT2 GTCCRD compare match	Edge	137 (0x089)	✓	✓	✓
		GPT2_CMPE	GPT2 GTCCRE compare match	Edge	138 (0x08A)	✓	✓	✓
		GPT2_CMPF	GPT2 GTCCRF compare match	Edge	139 (0x08B)	✓	✓	✓
		GPT2_OVF	GPT2 GTCNT overflow (GTPR compare match)	Edge	140 (0x08C)	✓	✓	✓
		GPT2_UDF	GPT2 GTCNT underflow	Edge	141 (0x08D)	✓	✓	✓
		GPT2_DTE	GPT2 Dead time error	Edge	142 (0x08E)	✓	—	—
		GPT2_ADTRGA	GPT2 GTADTRA compare match (A/D Converter Start Request A)	Edge	477 (0x1DD)	—	—	✓
		GPT2_ADTRGB	GPT2 GTADTRB compare match (A/D Converter Start Request B)	Edge	478 (0x1DE)	—	—	✓
GPT (LLPP)	CH3	GPT3_CCMPA	GPT3 GTCCRA input capture/ compare match	Edge	143 (0x08F)	✓	✓	✓
		GPT3_CCMPB	GPT3 GTCCRB input capture/ compare match	Edge	144 (0x090)	✓	✓	✓
		GPT3_CMPC	GPT3 GTCCRC compare match	Edge	145 (0x091)	✓	✓	✓
		GPT3_CMPD	GPT3 GTCCRD compare match	Edge	146 (0x092)	✓	✓	✓
		GPT3_CMPE	GPT3 GTCCRE compare match	Edge	147 (0x093)	✓	✓	✓
		GPT3_CMPF	GPT3 GTCCRF compare match	Edge	148 (0x094)	✓	✓	✓
		GPT3_OVF	GPT3 GTCNT overflow (GTPR compare match)	Edge	149 (0x095)	✓	✓	✓
		GPT3_UDF	GPT3 GTCNT underflow	Edge	150 (0x096)	✓	✓	✓
		GPT3_DTE	GPT3 Dead time error	Edge	151 (0x097)	✓	—	—
		GPT3_ADTRGA	GPT3 GTADTRA compare match (A/D Converter Start Request A)	Edge	479 (0x1DF)	—	—	✓
		GPT3_ADTRGB	GPT3 GTADTRB compare match (A/D Converter Start Request B)	Edge	480 (0x1E0)	—	—	✓

Table 11.2 Event table (7 of 17)

Module	UNIT /CH	Name	Interrupt sources	Interrupt type	Event number	CPU0 GIC request (SPI)	DMAC activation request	ELC source
GPT (LLPP)	CH4	GPT4_CCMPA	GPT4 GTCCRA input capture/ compare match	Edge	152 (0x098)	✓	✓	✓
		GPT4_CCMPB	GPT4 GTCCRB input capture/ compare match	Edge	153 (0x099)	✓	✓	✓
		GPT4_CMPC	GPT4 GTCCRC compare match	Edge	154 (0x09A)	✓	✓	✓
		GPT4_CMPD	GPT4 GTCCRD compare match	Edge	155 (0x09B)	✓	✓	✓
		GPT4_CMPE	GPT4 GTCCRE compare match	Edge	156 (0x09C)	✓	✓	✓
		GPT4_CMPF	GPT4 GTCCRF compare match	Edge	157 (0x09D)	✓	✓	✓
		GPT4_OVF	GPT4 GTCNT overflow (GTPR compare match)	Edge	158 (0x09E)	✓	✓	✓
		GPT4_UDF	GPT4 GTCNT underflow	Edge	159 (0x09F)	✓	✓	✓
		GPT4_DTE	GPT4 Dead time error	Edge	160 (0x0A0)	✓	—	—
		GPT4_ADTRGA	GPT4 GTADTRA compare match (A/D Converter Start Request A)	Edge	481 (0x1E1)	—	—	✓
		GPT4_ADTRGB	GPT4 GTADTRB compare match (A/D Converter Start Request B)	Edge	482 (0x1E2)	—	—	✓
GPT (LLPP)	CH5	GPT5_CCMPA	GPT5 GTCCRA input capture/ compare match	Edge	161 (0x0A1)	✓	✓	✓
		GPT5_CCMPB	GPT5 GTCCRB input capture/ compare match	Edge	162 (0x0A2)	✓	✓	✓
		GPT5_CMPC	GPT5 GTCCRC compare match	Edge	163 (0x0A3)	✓	✓	✓
		GPT5_CMPD	GPT5 GTCCRD compare match	Edge	164 (0x0A4)	✓	✓	✓
		GPT5_CMPE	GPT5 GTCCRE compare match	Edge	165 (0x0A5)	✓	✓	✓
		GPT5_CMPF	GPT5 GTCCRF compare match	Edge	166 (0x0A6)	✓	✓	✓
		GPT5_OVF	GPT5 GTCNT overflow (GTPR compare match)	Edge	167 (0x0A7)	✓	✓	✓
		GPT5_UDF	GPT5 GTCNT underflow	Edge	168 (0x0A8)	✓	✓	✓
		GPT5_DTE	GPT5 Dead time error	Edge	169 (0x0A9)	✓	—	—
		GPT5_ADTRGA	GPT5 GTADTRA compare match (A/D Converter Start Request A)	Edge	483 (0x1E3)	—	—	✓
		GPT5_ADTRGB	GPT5 GTADTRB compare match (A/D Converter Start Request B)	Edge	484 (0x1E4)	—	—	✓

Table 11.2 Event table (8 of 17)

Module	UNIT /CH	Name	Interrupt sources	Interrupt type	Event number	CPU0 GIC request (SPI)	DMAC activation request	ELC source
GPT (LLPP)	CH6	GPT6_CCMPA	GPT6 GTCCRA input capture/ compare match	Edge	170 (0x0AA)	✓	✓	✓
		GPT6_CCMPB	GPT6 GTCCRB input capture/ compare match	Edge	171 (0x0AB)	✓	✓	✓
		GPT6_CMPC	GPT6 GTCCRC compare match	Edge	172 (0x0AC)	✓	✓	✓
		GPT6_CMPD	GPT6 GTCCRD compare match	Edge	173 (0x0AD)	✓	✓	✓
		GPT6_CMPE	GPT6 GTCCRE compare match	Edge	174 (0x0AE)	✓	✓	✓
		GPT6_CMPF	GPT6 GTCCRF compare match	Edge	175 (0x0AF)	✓	✓	✓
		GPT6_OVF	GPT6 GTCNT overflow (GTPR compare match)	Edge	176 (0x0B0)	✓	✓	✓
		GPT6_UDF	GPT6 GTCNT underflow	Edge	177 (0x0B1)	✓	✓	✓
		GPT6_DTE	GPT6 Dead time error	Edge	178 (0x0B2)	✓	—	—
		GPT6_ADTRGA	GPT6 GTADTRA compare match (A/D Converter Start Request A)	Edge	485 (0x1E5)	—	—	✓
		GPT6_ADTRGB	GPT6 GTADTRB compare match (A/D Converter Start Request B)	Edge	486 (0x1E6)	—	—	✓
GPT (NONSAFETY)	CH7	GPT7_CCMPA	GPT7 GTCCRA input capture/ compare match	Edge	179 (0x0B3)	✓	✓	✓
		GPT7_CCMPB	GPT7 GTCCRB input capture/ compare match	Edge	180 (0x0B4)	✓	✓	✓
		GPT7_CMPC	GPT7 GTCCRC compare match	Edge	181 (0x0B5)	✓	✓	✓
		GPT7_CMPD	GPT7 GTCCRD compare match	Edge	182 (0x0B6)	✓	✓	✓
		GPT7_CMPE	GPT7 GTCCRE compare match	Edge	183 (0x0B7)	✓	✓	✓
		GPT7_CMPF	GPT7 GTCCRF compare match	Edge	184 (0x0B8)	✓	✓	✓
		GPT7_OVF	GPT7 GTCNT overflow (GTPR compare match)	Edge	185 (0x0B9)	✓	✓	✓
		GPT7_UDF	GPT7 GTCNT underflow	Edge	186 (0x0BA)	✓	✓	✓
		GPT7_DTE	GPT7 Dead time error	Edge	187 (0x0BB)	✓	—	—
		GPT7_ADTRGA	GPT7 GTADTRA compare match (A/D Converter Start Request A)	Edge	487 (0x1E7)	—	—	✓
		GPT7_ADTRGB	GPT7 GTADTRB compare match (A/D Converter Start Request B)	Edge	488 (0x1E8)	—	—	✓

Table 11.2 Event table (9 of 17)

Module	UNIT /CH	Name	Interrupt sources	Interrupt type	Event number	CPU0 GIC request (SPI)	DMAC activation request	ELC source
GPT (NONSAFETY)	CH8	GPT8_CCMPA	GPT8 GTCCRA input capture/ compare match	Edge	188 (0x0BC)	✓	✓	✓
		GPT8_CCMPB	GPT8 GTCCRB input capture/ compare match	Edge	189 (0x0BD)	✓	✓	✓
		GPT8_CMPC	GPT8 GTCCRC compare match	Edge	190 (0x0BE)	✓	✓	✓
		GPT8_CMPD	GPT8 GTCCRD compare match	Edge	191 (0x0BF)	✓	✓	✓
		GPT8_CMPE	GPT8 GTCCRE compare match	Edge	192 (0x0C0)	✓	✓	✓
		GPT8_CMPF	GPT8 GTCCRF compare match	Edge	193 (0x0C1)	✓	✓	✓
		GPT8_OVF	GPT8 GTCNT overflow (GTPR compare match)	Edge	194 (0x0C2)	✓	✓	✓
		GPT8_UDF	GPT8 GTCNT underflow	Edge	195 (0x0C3)	✓	✓	✓
		GPT8_DTE	GPT8 Dead time error	Edge	196 (0x0C4)	✓	—	—
		GPT8_ADTRGA	GPT8 GTADTRA compare match (A/D Converter Start Request A)	Edge	489 (0x1E9)	—	—	✓
		GPT8_ADTRGB	GPT8 GTADTRB compare match (A/D Converter Start Request B)	Edge	490 (0x1EA)	—	—	✓
GPT (NONSAFETY)	CH9	GPT9_CCMPA	GPT9 GTCCRA input capture/ compare match	Edge	197 (0x0C5)	✓	✓	✓
		GPT9_CCMPB	GPT9 GTCCRB input capture/ compare match	Edge	198 (0x0C6)	✓	✓	✓
		GPT9_CMPC	GPT9 GTCCRC compare match	Edge	199 (0x0C7)	✓	✓	✓
		GPT9_CMPD	GPT9 GTCCRD compare match	Edge	200 (0x0C8)	✓	✓	✓
		GPT9_CMPE	GPT9 GTCCRE compare match	Edge	201 (0x0C9)	✓	✓	✓
		GPT9_CMPF	GPT9 GTCCRF compare match	Edge	202 (0x0CA)	✓	✓	✓
		GPT9_OVF	GPT9 GTCNT overflow (GTPR compare match)	Edge	203 (0x0CB)	✓	✓	✓
		GPT9_UDF	GPT9 GTCNT underflow	Edge	204 (0x0CC)	✓	✓	✓
		GPT9_DTE	GPT9 Dead time error	Edge	205 (0x0CD)	✓	—	—
		GPT9_ADTRGA	GPT9 GTADTRA compare match (A/D Converter Start Request A)	Edge	491 (0x1EB)	—	—	✓
		GPT9_ADTRGB	GPT9 GTADTRB compare match (A/D Converter Start Request B)	Edge	492 (0x1EC)	—	—	✓

Table 11.2 Event table (10 of 17)

Module	UNIT /CH	Name	Interrupt sources	Interrupt type	Event number	CPU0 GIC request (SPI)	DMAC activation request	ELC source
GPT (NONSAFETY)	CH10	GPT10_CCMPA	GPT10 GTCCRA input capture/ compare match	Edge	206 (0x0CE)	✓	✓	✓
		GPT10_CCMPB	GPT10 GTCCRB input capture/ compare match	Edge	207 (0x0CF)	✓	✓	✓
		GPT10_CMPC	GPT10 GTCCRC compare match	Edge	208 (0x0D0)	✓	✓	✓
		GPT10_CMPD	GPT10 GTCCRD compare match	Edge	209 (0x0D1)	✓	✓	✓
		GPT10_CMPE	GPT10 GTCCRE compare match	Edge	210 (0x0D2)	✓	✓	✓
		GPT10_CMPF	GPT10 GTCCRF compare match	Edge	211 (0x0D3)	✓	✓	✓
		GPT10_OVF	GPT10 GTCNT overflow (GTPR compare match)	Edge	212 (0x0D4)	✓	✓	✓
		GPT10_UDF	GPT10 GTCNT underflow	Edge	213 (0x0D5)	✓	✓	✓
		GPT10_DTE	GPT10 Dead time error	Edge	214 (0x0D6)	✓	—	—
		GPT10_ADTRGA	GPT10 GTADTRA compare match (A/D Converter Start Request A)	Edge	493 (0x1ED)	—	—	✓
		GPT10_ADTRGB	GPT10 GTADTRB compare match (A/D Converter Start Request B)	Edge	494 (0x1EE)	—	—	✓
GPT (NONSAFETY)	CH11	GPT11_CCMPA	GPT11 GTCCRA input capture/ compare match	Edge	215 (0x0D7)	✓	✓	✓
		GPT11_CCMPB	GPT11 GTCCRB input capture/ compare match	Edge	216 (0x0D8)	✓	✓	✓
		GPT11_CMPC	GPT11 GTCCRC compare match	Edge	217 (0x0D9)	✓	✓	✓
		GPT11_CMPD	GPT11 GTCCRD compare match	Edge	218 (0x0DA)	✓	✓	✓
		GPT11_CMPE	GPT11 GTCCRE compare match	Edge	219 (0x0DB)	✓	✓	✓
		GPT11_CMPF	GPT11 GTCCRF compare match	Edge	220 (0x0DC)	✓	✓	✓
		GPT11_OVF	GPT11 GTCNT overflow (GTPR compare match)	Edge	221 (0x0DD)	✓	✓	✓
		GPT11_UDF	GPT11 GTCNT underflow	Edge	222 (0x0DE)	✓	✓	✓
		GPT11_DTE	GPT11 Dead time error	Edge	223 (0x0DF)	✓	—	—
		GPT11_ADTRGA	GPT11 GTADTRA compare match (A/D Converter Start Request A)	Edge	495 (0x1EF)	—	—	✓
		GPT11_ADTRGB	GPT11 GTADTRB compare match (A/D Converter Start Request B)	Edge	496 (0x1F0)	—	—	✓

Table 11.2 Event table (11 of 17)

Module	UNIT /CH	Name	Interrupt sources	Interrupt type	Event number	CPU0 GIC request (SPI)	DMAC activation request	ELC source
GPT (NONSAFETY)	CH12	GPT12_CCMPA	GPT12 GTCCRA input capture/ compare match	Edge	224 (0x0E0)	✓	✓	✓
		GPT12_CCMPB	GPT12 GTCCRB input capture/ compare match	Edge	225 (0x0E1)	✓	✓	✓
		GPT12_CMPC	GPT12 GTCCRC compare match	Edge	226 (0x0E2)	✓	✓	✓
		GPT12_CMPD	GPT12 GTCCRD compare match	Edge	227 (0x0E3)	✓	✓	✓
		GPT12_CMPE	GPT12 GTCCRE compare match	Edge	228 (0x0E4)	✓	✓	✓
		GPT12_CMPF	GPT12 GTCCRF compare match	Edge	229 (0x0E5)	✓	✓	✓
		GPT12_OVF	GPT12 GTCNT overflow (GTPR compare match)	Edge	230 (0x0E6)	✓	✓	✓
		GPT12_UDF	GPT12 GTCNT underflow	Edge	231 (0x0E7)	✓	✓	✓
		GPT12_DTE	GPT12 Dead time error	Edge	232 (0x0E8)	✓	—	—
		GPT12_ADTRGA	GPT12 GTADTRA compare match (A/D Converter Start Request A)	Edge	497 (0x1F1)	—	—	✓
		GPT12_ADTRGB	GPT12 GTADTRB compare match (A/D Converter Start Request B)	Edge	498 (0x1F2)	—	—	✓
GPT (NONSAFETY)	CH13	GPT13_CCMPA	GPT13 GTCCRA input capture/ compare match	Edge	233 (0x0E9)	✓	✓	✓
		GPT13_CCMPB	GPT13 GTCCRB input capture/ compare match	Edge	234 (0x0EA)	✓	✓	✓
		GPT13_CMPC	GPT13 GTCCRC compare match	Edge	235 (0x0EB)	✓	✓	✓
		GPT13_CMPD	GPT13 GTCCRD compare match	Edge	236 (0x0EC)	✓	✓	✓
		GPT13_CMPE	GPT13 GTCCRE compare match	Edge	237 (0x0ED)	✓	✓	✓
		GPT13_CMPF	GPT13 GTCCRF compare match	Edge	238 (0x0EE)	✓	✓	✓
		GPT13_OVF	GPT13 GTCNT overflow (GTPR compare match)	Edge	239 (0x0EF)	✓	✓	✓
		GPT13_UDF	GPT13 GTCNT underflow	Edge	240 (0x0F0)	✓	✓	✓
		GPT13_DTE	GPT13 Dead time error	Edge	241 (0x0F1)	✓	—	—
		GPT13_ADTRGA	GPT13 GTADTRA compare match (A/D Converter Start Request A)	Edge	499 (0x1F3)	—	—	✓
		GPT13_ADTRGB	GPT13 GTADTRB compare match (A/D Converter Start Request B)	Edge	500 (0x1F4)	—	—	✓

Table 11.2 Event table (12 of 17)

Module	UNIT /CH	Name	Interrupt sources	Interrupt type	Event number	CPU0 GIC request (SPI)	DMAC activation request	ELC source
GPT (SAFETY)	CH14	GPT14_CCMPA	GPT14 GTCCRA input capture/ compare match	Edge	396 (0x18C)	✓	—	—
		GPT14_CCMPB	GPT14 GTCCRB input capture/ compare match	Edge	397 (0x18D)	✓	—	—
		GPT14_CMPC	GPT14 GTCCRC compare match	Edge	398 (0x18E)	✓	—	—
		GPT14_CMPD	GPT14 GTCCRD compare match	Edge	399 (0x18F)	✓	—	—
		GPT14_CMPE	GPT14 GTCCRE compare match	Edge	400 (0x190)	✓	—	—
		GPT14_CMPF	GPT14 GTCCRF compare match	Edge	401 (0x191)	✓	—	—
		GPT14_OVF	GPT14 GTCNT overflow (GTPR compare match)	Edge	402 (0x192)	✓	—	—
		GPT14_UDF	GPT14 GTCNT underflow	Edge	403 (0x193)	✓	—	—
GPT (SAFETY)	CH15	GPT15_CCMPA	GPT15 GTCCRA input capture/ compare match	Edge	404 (0x194)	✓	—	—
		GPT15_CCMPB	GPT15 GTCCRB input capture/ compare match	Edge	405 (0x195)	✓	—	—
		GPT15_CMPC	GPT15 GTCCRC compare match	Edge	406 (0x196)	✓	—	—
		GPT15_CMPD	GPT15 GTCCRD compare match	Edge	407 (0x197)	✓	—	—
		GPT15_CMPE	GPT15 GTCCRE compare match	Edge	408 (0x198)	✓	—	—
		GPT15_CMPF	GPT15 GTCCRF compare match	Edge	409 (0x199)	✓	—	—
		GPT15_OVF	GPT15 GTCNT overflow (GTPR compare match)	Edge	410 (0x19A)	✓	—	—
		GPT15_UDF	GPT15 GTCNT underflow	Edge	411 (0x19B)	✓	—	—
GPT (SAFETY)	CH16	GPT16_CCMPA	GPT16 GTCCRA input capture/ compare match	Edge	412 (0x19C)	✓	—	—
		GPT16_CCMPB	GPT16 GTCCRB input capture/ compare match	Edge	413 (0x19D)	✓	—	—
		GPT16_CMPC	GPT16 GTCCRC compare match	Edge	414 (0x19E)	✓	—	—
		GPT16_CMPD	GPT16 GTCCRD compare match	Edge	415 (0x19F)	✓	—	—
		GPT16_CMPE	GPT16 GTCCRE compare match	Edge	416 (0x1A0)	✓	—	—
		GPT16_CMPF	GPT16 GTCCRF compare match	Edge	417 (0x1A1)	✓	—	—
		GPT16_OVF	GPT16 GTCNT overflow (GTPR compare match)	Edge	418 (0x1A2)	✓	—	—
		GPT16_UDF	GPT16 GTCNT underflow	Edge	419 (0x1A3)	✓	—	—

Table 11.2 Event table (13 of 17)

Module	UNIT /CH	Name	Interrupt sources	Interrupt type	Event number	CPU0 GIC request (SPI)	DMAC activation request	ELC source
GPT (SAFETY)	CH17	GPT17_CCMPA	GPT17 GTCCRA input capture/ compare match	Edge	420 (0x1A4)	✓	—	—
		GPT17_CCMPB	GPT17 GTCCRB input capture/ compare match	Edge	421 (0x1A5)	✓	—	—
		GPT17_CMPC	GPT17 GTCCRC compare match	Edge	422 (0x1A6)	✓	—	—
		GPT17_CMPD	GPT17 GTCCRD compare match	Edge	423 (0x1A7)	✓	—	—
		GPT17_CMPE	GPT17 GTCCRE compare match	Edge	424 (0x1A8)	✓	—	—
		GPT17_CMPF	GPT17 GTCCRF compare match	Edge	425 (0x1A9)	✓	—	—
		GPT17_OVF	GPT17 GTCNT overflow (GTPR compare match)	Edge	426 (0x1AA)	✓	—	—
		GPT17_UDF	GPT17 GTCNT underflow	Edge	427 (0x1AB)	✓	—	—
POEG (LLPP)	UNIT0	POEG0_GROUP0	POEG group A interrupt for channels in LLPP	Level	242 (0x0F2)	✓	—	—
		POEG0_GROUP1	POEG group B interrupt for channels in LLPP	Level	243 (0x0F3)	✓	—	—
		POEG0_GROUP2	POEG group C interrupt for channels in LLPP	Level	244 (0x0F4)	✓	—	—
		POEG0_GROUP3	POEG group D interrupt for channels in LLPP	Level	245 (0x0F5)	✓	—	—
POEG (NONSAFETY)	UNIT1	POEG1_GROUP0	POEG group A interrupt for channels in NONSAFETY	Level	246 (0x0F6)	✓	—	—
		POEG1_GROUP1	POEG group B interrupt for channels in NONSAFETY	Level	247 (0x0F7)	✓	—	—
		POEG1_GROUP2	POEG group C interrupt for channels in NONSAFETY	Level	248 (0x0F8)	✓	—	—
		POEG1_GROUP3	POEG group D interrupt for channels in NONSAFETY	Level	249 (0x0F9)	✓	—	—
POEG (SAFETY)	UNIT2	POEG2_GROUP0	POEG group A interrupt for channels in SAFETY	Level	428 (0x1AC)	✓	—	—
		POEG2_GROUP1	POEG group B interrupt for channels in SAFETY	Level	429 (0x1AD)	✓	—	—
		POEG2_GROUP2	POEG group C interrupt for channels in SAFETY	Level	430 (0x1AE)	✓	—	—
		POEG2_GROUP3	POEG group D interrupt for channels in SAFETY	Level	431 (0x1AF)	✓	—	—
RTC	UNIT0	RTC_ALM	Alarm interrupt	Edge	432 (0x1B0)	✓	—	—
		RTC_1S	1 second interrupt	Edge	433 (0x1B1)	✓	—	—
		RTC_PRD	Fixed interval interrupt	Edge	434 (0x1B2)	✓	—	—
GMAC	UNIT0	GMAC_LPI	GMAC1 energy efficient	Level	250 (0x0FA)	✓	—	—
		GMAC_PMT	GMAC1 power management	Level	251 (0x0FB)	✓	—	—
		GMAC_SBD	GMAC1 general interrupt	Level	252 (0x0FC)	✓	✓	—
		GMAC_PTPOUT0	GMAC PTP timer pulse output 0	Level/Edge	269 (0x10D)	✓	✓	✓
		GMAC_PTPOUT1	GMAC PTP timer pulse output 1	Level/Edge	270 (0x10E)	✓	✓	✓
		GMAC_PTPOUT2	GMAC PTP timer pulse output 2	Level/Edge	271 (0x10F)	✓	✓	✓
		GMAC_PTPOUT3	GMAC PTP timer pulse output 3	Level/Edge	272 (0x110)	✓	✓	✓
ENDAT	UNIT0	ENDAT0_INT1	ENDAT0 interrupt output 1	Level	255 (0x0FF)	✓	✓	—
	UNIT1	ENDAT1_INT1	ENDAT1 interrupt output 1	Level	258 (0x102)	✓	✓	—

Table 11.2 Event table (14 of 17)

Module	UNIT /CH	Name	Interrupt sources	Interrupt type	Event number	CPU0 GIC request (SPI)	DMAC activation request	ELC source
BISS	UNIT0	BISS0_EOT	BISS0 End of transmission interrupt	Edge	261 (0x105)	✓	✓	—
	UNIT1	BISS1_EOT	BISS1 End of transmission interrupt	Edge	262 (0x106)	✓	✓	—
HDSL	UNIT0	HDSL0_INT	HDSL0 Interrupt output	Level	263 (0x107)	✓	✓	—
		HDSL0_INTS	HDSL0 Interrupt output (safety related)	Level	264 (0x108)	✓	✓	—
		HDSL0_FPR	HDSL0 Fast position value ready interrupt	Edge	273 (0x111)	✓	✓	—
		HDSL0_SP	HDSL0 Safe position received interrupt	Edge	275 (0x113)	✓	✓	✓
	UNIT1	HDSL1_INT	HDSL1 Interrupt output	Level	265 (0x109)	✓	✓	—
		HDSL1_INTS	HDSL1 Interrupt output (safety related)	Level	266 (0x10A)	✓	✓	—
		HDSL1_FPR	HDSL1 Fast position value ready interrupt	Edge	274 (0x112)	✓	✓	—
		HDSL1_SP	HDSL1 Safe position received interrupt	Edge	276 (0x114)	✓	✓	✓
AFMT	UNIT0	AFMT0_EOF	AFMT0 communication completion	Level	267 (0x10B)	✓	✓	—
	UNIT1	AFMT1_EOF	AFMT1 communication completion	Level	268 (0x10C)	✓	✓	—
ESC	UNIT0	ESC_SYNC0	EtherCAT Sync0 interrupt	Level/Edge	277 (0x115)	✓	✓	✓
		ESC_SYNC1	EtherCAT Sync1 interrupt	Level/Edge	278 (0x116)	✓	✓	✓
		ESC_CAT	EtherCAT interrupt	Level	279 (0x117)	✓	✓	—
		ESC_SOF	EtherCAT SOF interrupt	Edge	280 (0x118)	✓	✓	—
		ESC_EOF	EtherCAT EOF interrupt	Edge	281 (0x119)	✓	✓	—
		ESC_WDT	EtherCAT WDT interrupt	Edge	282 (0x11A)	✓	—	—
		ESC_RST	EtherCAT RESET interrupt	Level	283 (0x11B)	✓	—	—
USBHC	UNIT0	USB_HI	USB (Host) interrupt	Level	284 (0x11C)	✓	✓	—
USBf	UNIT0	USB_FI	USB (Function) interrupt	Level/Edge	285 (0x11D)	✓	✓	—
		USB_FDMA0	USB (Function) DMA 0 transmit completion	Level/Edge	286 (0x11E)	✓	—	—
		USB_FDMA1	USB (Function) DMA 1 transmit completion	Level/Edge	287 (0x11F)	✓	—	—
SCI (NONSAFETY)	CH0	SCI0_ERI	SCI0 Receive error	Level	288 (0x120)	✓	—	—
		SCI0_RXI	SCI0 Receive data full	Edge	289 (0x121)	✓	✓	—
		SCI0_TXI	SCI0 Transmit data empty	Edge	290 (0x122)	✓	✓	—
		SCI0_TEI	SCI0 Transmit end	Level	291 (0x123)	✓	—	—
SCI (NONSAFETY)	CH1	SCI1_ERI	SCI1 Receive error	Level	292 (0x124)	✓	—	—
		SCI1_RXI	SCI1 Receive data full	Edge	293 (0x125)	✓	✓	—
		SCI1_TXI	SCI1 Transmit data empty	Edge	294 (0x126)	✓	✓	—
		SCI1_TEI	SCI1 Transmit end	Level	295 (0x127)	✓	—	—
SCI (NONSAFETY)	CH2	SCI2_ERI	SCI2 Receive error	Level	296 (0x128)	✓	—	—
		SCI2_RXI	SCI2 Receive data full	Edge	297 (0x129)	✓	✓	—
		SCI2_TXI	SCI2 Transmit data empty	Edge	298 (0x12A)	✓	✓	—
		SCI2_TEI	SCI2 Transmit end	Level	299 (0x12B)	✓	—	—

Table 11.2 Event table (15 of 17)

Module	UNIT /CH	Name	Interrupt sources	Interrupt type	Event number	CPU0 GIC request (SPI)	DMAC activation request	ELC source
SCI (NONSAFETY)	CH3	SCI3_ERI	SCI3 Receive error	Level	300 (0x12C)	✓	—	—
		SCI3_RXI	SCI3 Receive data full	Edge	301 (0x12D)	✓	✓	—
		SCI3_TXI	SCI3 Transmit data empty	Edge	302 (0x12E)	✓	✓	—
		SCI3_TEI	SCI3 Transmit end	Level	303 (0x12F)	✓	—	—
SCI (NONSAFETY)	CH4	SCI4_ERI	SCI4 Receive error	Level	304 (0x130)	✓	—	—
		SCI4_RXI	SCI4 Receive data full	Edge	305 (0x131)	✓	✓	—
		SCI4_TXI	SCI4 Transmit data empty	Edge	306 (0x132)	✓	✓	—
		SCI4_TEI	SCI4 Transmit end	Level	307 (0x133)	✓	—	—
SCI (SAFETY)	CH5	SCI5_ERI	SCI5 Receive error	Level	435 (0x1B3)	✓	—	—
		SCI5_RXI	SCI5 Receive data full	Edge	436 (0x1B4)	✓	—	—
		SCI5_TXI	SCI5 Transmit data empty	Edge	437 (0x1B5)	✓	—	—
		SCI5_TEI	SCI5 Transmit end	Level	438 (0x1B6)	✓	—	—
IIC (NONSAFETY)	CH0	IIC0_EEI	IIC0 Transfer error or event generation	Level	308 (0x134)	✓	—	—
		IIC0_RXI	IIC0 Receive data full	Edge	309 (0x135)	✓	✓	—
		IIC0_TXI	IIC0 Transmit data empty	Edge	310 (0x136)	✓	✓	—
		IIC0_TEI	IIC0 Transmit end	Level	311 (0x137)	✓	—	—
IIC (NONSAFETY)	CH1	IIC1_EEI	IIC1 Transfer error or event generation	Level	312 (0x138)	✓	—	—
		IIC1_RXI	IIC1 Receive data full	Edge	313 (0x139)	✓	✓	—
		IIC1_TXI	IIC1 Transmit data empty	Edge	314 (0x13A)	✓	✓	—
		IIC1_TEI	IIC1 Transmit end	Level	315 (0x13B)	✓	—	—
IIC (SAFETY)	CH2	IIC2_EEI	IIC2 Transfer error or event generation	Level	439 (0x1B7)	✓	—	—
		IIC2_RXI	IIC2 Receive data full	Edge	440 (0x1B8)	✓	—	—
		IIC2_TXI	IIC2 Transmit data empty	Edge	441 (0x1B9)	✓	—	—
		IIC2_TEI	IIC2 Transmit end	Level	442 (0x1BA)	✓	—	—
CANFD	COM	CAN_RXF	CANFD RX FIFO interrupt	Level	316 (0x13C)	✓	—	—
		CAN_GLERR	CANFD Global error interrupt	Level	317 (0x13D)	✓	—	—
		CAN_RF_DMAREQ0	CANFD RX FIFO 0 DMA request	Edge	449 (0x1C1)	—	✓	—
		CAN_RF_DMAREQ1	CANFD RX FIFO 1 DMA request	Edge	450 (0x1C2)	—	✓	—
		CAN_RF_DMAREQ2	CANFD RX FIFO 2 DMA request	Edge	451 (0x1C3)	—	✓	—
		CAN_RF_DMAREQ3	CANFD RX FIFO 3 DMA request	Edge	452 (0x1C4)	—	✓	—
		CAN_RF_DMAREQ4	CANFD RX FIFO 4 DMA request	Edge	453 (0x1C5)	—	✓	—
		CAN_RF_DMAREQ5	CANFD RX FIFO 5 DMA request	Edge	454 (0x1C6)	—	✓	—
		CAN_RF_DMAREQ6	CANFD RX FIFO 6 DMA request	Edge	455 (0x1C7)	—	✓	—
		CAN_RF_DMAREQ7	CANFD RX FIFO 7 DMA request	Edge	456 (0x1C8)	—	✓	—

Table 11.2 Event table (16 of 17)

Module	UNIT /CH	Name	Interrupt sources	Interrupt type	Event number	CPU0 GIC request (SPI)	DMAC activation request	ELC source
CANFD	CH0	CAN0_TX	CANFD0 Channel TX interrupt	Level	318 (0x13E)	✓	—	—
		CAN0_CHERR	CANFD0 Channel CAN error interrupt	Level	319 (0x13F)	✓	—	—
		CAN0_COMFRX	CANFD0 Common RX FIFO or TXQ interrupt	Level	320 (0x140)	✓	—	—
		CAN0_CF_DMAREQ	CANFD0 First common FIFO DMA request	Edge	457 (0x1C9)	—	✓	—
CANFD	CH1	CAN1_TX	CANFD1 Channel TX interrupt	Level	321 (0x141)	✓	—	—
		CAN1_CHERR	CANFD1 Channel CAN error interrupt	Level	322 (0x142)	✓	—	—
		CAN1_COMFRX	CANFD1 Common RX FIFO or TXQ interrupt	Level	323 (0x143)	✓	—	—
		CAN1_CF_DMAREQ	CANFD1 First common FIFO DMA request	Edge	458 (0x1CA)	—	✓	—
SPI (NONSAFTY)	CH0	SPI0_SPRI	SPI0 Reception buffer full	Edge	324 (0x144)	✓	✓	—
		SPI0_SPTI	SPI0 Transmit buffer empty	Edge	325 (0x145)	✓	✓	—
		SPI0_SPII	SPI0 SPI idle	Level	326 (0x146)	✓	—	—
		SPI0_SPEI	SPI0 errors (mode fault, underrun, overrun, parity error, and received data ready)	Level	327 (0x147)	✓	—	—
		SPI0_SPCEND	SPI0 Communication complete	Edge	328 (0x148)	✓	—	—
SPI (NONSAFTY)	CH1	SPI1_SPRI	SPI1 Reception buffer full	Edge	329 (0x149)	✓	✓	—
		SPI1_SPTI	SPI1 Transmit buffer empty	Edge	330 (0x14A)	✓	✓	—
		SPI1_SPII	SPI1 SPI idle	Level	331 (0x14B)	✓	—	—
		SPI1_SPEI	SPI1 errors (mode fault, underrun, overrun, parity error, and received data ready)	Level	332 (0x14C)	✓	—	—
		SPI1_SPCEND	SPI1 Communication complete	Edge	333 (0x14D)	✓	—	—
SPI (NONSAFTY)	CH2	SPI2_SPRI	SPI2 Reception buffer full	Edge	334 (0x14E)	✓	✓	—
		SPI2_SPTI	SPI2 Transmit buffer empty	Edge	335 (0x14F)	✓	✓	—
		SPI2_SPII	SPI2 SPI idle	Level	336 (0x150)	✓	—	—
		SPI2_SPEI	SPI2 errors (mode fault, underrun, overrun, parity error, and received data ready)	Level	337 (0x151)	✓	—	—
		SPI2_SPCEND	SPI2 Communication complete	Edge	338 (0x152)	✓	—	—
SPI (SAFETY)	CH3	SPI3_SPRI	SPI3 Reception buffer full	Edge	443 (0x1BB)	✓	—	—
		SPI3_SPTI	SPI3 Transmit buffer empty	Edge	444 (0x1BC)	✓	—	—
		SPI3_SPII	SPI3 SPI idle	Level	445 (0x1BD)	✓	—	—
		SPI3_SPEI	SPI3 errors (mode fault, underrun, overrun, parity error, and received data ready)	Level	446 (0x1BE)	✓	—	—
		SPI3_SPCEND	SPI3 Communication complete	Edge	447 (0x1BF)	✓	—	—
xSPI	UNIT0	XSPI0_INT	xSPI0 Interrupt	Level	339 (0x153)	✓	—	—
		XSPI0_INTERR	xSPI0 Error interrupt	Level	340 (0x154)	✓	—	—
	UNIT1	XSPI1_INT	xSPI1 Interrupt	Level	341 (0x155)	✓	—	—
		XSPI1_INTERR	xSPI1 Error interrupt	Level	342 (0x156)	✓	—	—

Table 11.2 Event table (17 of 17)

Module	UNIT /CH	Name	Interrupt sources	Interrupt type	Event number	CPU0 GIC request (SPI)	DMAC activation request	ELC source
DSMIF	UNIT0	DSMIF0_CDRUI	DSMIF0 current data register update (ORed ch0 to ch2)	Edge	343 (0x157)	✓	—	—
		DSMIF0_CDUPDC	DSMIF0 current data register update common	Edge	376 (0x178)	✓	✓	—
		DSMIF0_CDACUPDC	DSMIF0 current capture data register A update common	Edge	377 (0x179)	✓	✓	—
		DSMIF0_CDBCUPDC	DSMIF0 current capture data register B update common	Edge	378 (0x17A)	✓	✓	—
	UNIT1	DSMIF1_CDRUI	DSMIF1 current data register update (ORed ch3 to ch5)	Edge	344 (0x158)	✓	—	—
		DSMIF1_CDUPDC	DSMIF1 current data register update common	Edge	379 (0x17B)	✓	✓	—
		DSMIF1_CDACUPDC	DSMIF1 current capture data register A update common	Edge	380 (0x17C)	✓	✓	—
		DSMIF1_CDBCUPDC	DSMIF1 current capture data register B update common	Edge	381 (0x17D)	✓	✓	—
ADC12	UNIT0	ADC0_ADI	ADC0 A/D scan end interrupt	Edge	345 (0x159)	✓	✓	✓
		ADC0_GBADI	ADC0 A/D scan end interrupt for Group B	Edge	346 (0x15A)	✓	✓	—
		ADC0_GCADI	ADC0 A/D scan end interrupt for Group C	Edge	347 (0x15B)	✓	✓	—
		ADC0_CMPAI	ADC0 Window A compare match	Level	348 (0x15C)	✓	—	—
		ADC0_CMPBI	ADC0 Window B compare match	Level	349 (0x15D)	✓	—	—
		ADC0_WCMPM	ADC0 compare match	Edge	459 (0x1CB)	—	✓	✓
		ADC0_WCMPUM	ADC0 compare mismatch.	Edge	460 (0x1CC)	—	✓	✓
ADC12	UNIT1	ADC1_ADI	ADC1 A/D scan end interrupt	Edge	350 (0x15E)	✓	✓	✓
		ADC1_GBADI	ADC1 A/D scan end interrupt for Group B	Edge	351 (0x15F)	✓	✓	—
		ADC1_GCADI	ADC1 A/D scan end interrupt for Group C	Edge	352 (0x160)	✓	✓	—
		ADC1_CMPAI	ADC1 Window A compare match	Level	353 (0x161)	✓	—	—
		ADC1_CMPBI	ADC1 Window B compare match	Level	354 (0x162)	✓	—	—
		ADC1_WCMPM	ADC1 compare match	Edge	461 (0x1CD)	—	✓	✓
		ADC1_WCMPUM	ADC1 compare mismatch.	Edge	462 (0x1CE)	—	✓	✓
MBXSEM	UNIT0	MBX_INT0	Mailbox (Host CPU to Cortex-R52) interrupt 0	Level	372 (0x174)	✓	✓	✓
		MBX_INT1	Mailbox (Host CPU to Cortex-R52) interrupt 1	Level	373 (0x175)	✓	✓	✓
		MBX_INT2	Mailbox (Host CPU to Cortex-R52) interrupt 2	Level	374 (0x176)	✓	✓	✓
		MBX_INT3	Mailbox (Host CPU to Cortex-R52) interrupt 3	Level	375 (0x177)	✓	✓	✓
SHOSTIF	UNIT0	SHOST_INT	SHOSTIF interrupt	Level	390 (0x186)	✓	—	—
Cortex-R52	CPU0	CPU0_ERR0	Cortex-R52 CPU0 error event 0	Edge	384 (0x180)	✓	—	—
		CPU0_ERR1	Cortex-R52 CPU0 error event 1	Edge	385 (0x181)	✓	—	—
Peripherals	—	PERI_ERR0	Peripherals error event 0	Edge	388 (0x184)	✓	—	—
		PERI_ERR1	Peripherals error event 1	Edge	389 (0x185)	✓	—	—

Note 1. NMI is connected to GIC Physical system error interrupt (SEI) and Virtual system error interrupt (VSEI) of CPU0.

11.3.3 Error Events

The ICU handles error events from modules. CPU0ERR_STAT, PERIERR_STAT0, PERIERR_STAT1, PERIERR_STAT2, PERIERR_STAT3 registers show each error status. The ICU captures error events and generates combined interrupt signal CPU0_ERR0, PERI_ERR0 and CPU0_ERR1, PERI_ERR1.

Error events can be disabled by CPU0ERR_E0MSK, PERIERR_E0MSK0, PERIERR_E0MSK1, PERIERR_E0MSK2, PERIERR_E0MSK3 registers and CPU0ERR_E1MSK, PERIERR_E1MSK0, PERIERR_E1MSK1, PERIERR_E1MSK2, PERIERR_E1MSK3 registers.

Additionally, error events can cause the error reset. Error events that cause error reset are maskable in CPU0ERR_RSTMSK, PERIERR_RSTMSK0, PERIERR_RSTMSK1, PERIERR_RSTMSK2, and PERIERR_RSTMSK3.

All error events are masked as initial values. *ERR_STAT register is not subject to initialization by Error reset. Since it is not initialized, Error will occur again if the Error reset mask is released. If it returns to normal operation after Error reset, it is necessary to confirm error status and clear error flag after restarting.

Table 11.3 shows the relation among error control registers and combined interrupts. Table 11.4 shows error events.

Table 11.3 Relation among error control registers and combined interrupts

Module	Error Event Status register	Error Event Status Clear register	Error Event Mask register	Combined interrupt
Cortex-R52 CPU0	CPU0ERR_STAT	CPU0ERR_CLR	CPU0ERR_E0MSK	CPU0_ERR0
			CPU0ERR_E1MSK	CPU0_ERR1
			CPU0ERR_RSTMSK	Error reset
Peripheral modules	PERIERR_STAT0, PERIERR_STAT1, PERIERR_STAT2, PERIERR_STAT3	PERIERR_CLR0, PERIERR_CLR1, PERIERR_CLR2, PERIERR_CLR3	PERIERR_E0MSK0	PERI_ERR0
			PERIERR_E0MSK1	
			PERIERR_E0MSK2	
			PERIERR_E0MSK3	
			PERIERR_E1MSK0	PERI_ERR1
			PERIERR_E1MSK1	
			PERIERR_E1MSK2	
			PERIERR_E1MSK3	
			PERIERR_RSTMSK0	Error reset
			PERIERR_RSTMSK1	
			PERIERR_RSTMSK2	
			PERIERR_RSTMSK3	

Table 11.4 Error event table (1 of 9)

Module	UNIT/ CH	Name	Interrupt sources	Bit of PERIERR _xxx0*1	Bit of PERIERR _xxx1*2	Bit of PERIERR _xxx2*3	Bit of PERIERR _xxx3*4	Bit of CPU0ERR _xxx*5
CLMA	UNIT3	CLMA3_INT	Main OSC stop detection	0	—	—	—	—
	UNIT0	CLMA0_INT	CLMA0 error detection	1	—	—	—	—
	UNIT1	CLMA1_INT	CLMA1 error detection	2	—	—	—	—
	UNIT2	CLMA2_INT	CLMA2 error detection	3	—	—	—	—
BSC	CH0	BSC_WTO	External wait timeout detection interrupt	4	—	—	—	—
DMAC	UNIT0	DMAC0_ERR	DMAC0 transfer error	5	—	—	—	—
	UNIT1	DMAC1_ERR	DMAC1 transfer error	6	—	—	—	—
WDT	CH0	WDT_NMIUNDF0	Down-counter underflow/ refresh error for CPU0	7	—	—	—	—
USB	CH0	USB_FDMAERR	USB (Function) DMA error	9	—	—	—	—

Table 11.4 Error event table (2 of 9)

Module	UNIT/ CH	Name	Interrupt sources	Bit of PERIERR _xxx0*1	Bit of PERIERR _xxx1*2	Bit of PERIERR _xxx2*3	Bit of PERIERR _xxx3*4	Bit of CPU0ERR _xxx*5
DSMIF	UNIT0	DSMIF0_LTCSE	DSMIF0 lower threshold current sum error	10	—	—	—	—
		DSMIF0_UTCSE	DSMIF0 upper threshold current sum error	11	—	—	—	—
		DSMIF0_OVC0EL0	DSMIF0 lower threshold overcurrent detection 0 CH0	12	—	—	—	—
		DSMIF0_OVC0EL1	DSMIF0 lower threshold overcurrent detection 0 CH1	13	—	—	—	—
		DSMIF0_OVC0EL2	DSMIF0 lower threshold overcurrent detection 0 CH2	14	—	—	—	—
		DSMIF0_OVC0EH0	DSMIF0 upper threshold overcurrent detection 0 CH0	15	—	—	—	—
		DSMIF0_OVC0EH1	DSMIF0 upper threshold overcurrent detection 0 CH1	16	—	—	—	—
		DSMIF0_OVC0EH2	DSMIF0 upper threshold overcurrent detection 0 CH2	17	—	—	—	—
		DSMIF0_SCDE0	DSMIF0 Short-circuit detection 0	18	—	—	—	—
		DSMIF0_SCDE1	DSMIF0 Short-circuit detection 1	19	—	—	—	—
		DSMIF0_SCDE2	DSMIF0 Short-circuit detection 2	20	—	—	—	—
		DSMIF0_OVC1EL0	DSMIF0 lower threshold overcurrent detection 1 CH0	—	29	—	—	—
		DSMIF0_OVC1EH0	DSMIF0 upper threshold overcurrent detection 1 CH0	—	30	—	—	—
		DSMIF0_OVC2EL0	DSMIF0 lower threshold overcurrent detection 2 CH0	—	31	—	—	—
		DSMIF0_OVC2EH0	DSMIF0 upper threshold overcurrent detection 2 CH0	—	—	0	—	—
		DSMIF0_OVC1EL1	DSMIF0 lower threshold overcurrent detection 1 CH1	—	—	1	—	—
		DSMIF0_OVC1EH1	DSMIF0 upper threshold overcurrent detection 1 CH1	—	—	2	—	—
		DSMIF0_OVC2EL1	DSMIF0 lower threshold overcurrent detection 2 CH1	—	—	3	—	—
		DSMIF0_OVC2EH1	DSMIF0 upper threshold overcurrent detection 2 CH1	—	—	4	—	—

Table 11.4 Error event table (3 of 9)

Module	UNIT/ CH	Name	Interrupt sources	Bit of PERIERR _xxx0*1	Bit of PERIERR _xxx1*2	Bit of PERIERR _xxx2*3	Bit of PERIERR _xxx3*4	Bit of CPU0ERR _xxx*5
DSMIF	UNIT0	DSMIF0_OVC1EL2	DSMIF0 lower threshold overcurrent detection 1 CH2	—	—	5	—	—
		DSMIF0_OVC1EH2	DSMIF0 upper threshold overcurrent detection 1 CH2	—	—	6	—	—
		DSMIF0_OVC2EL2	DSMIF0 lower threshold overcurrent detection 2 CH2	—	—	7	—	—
		DSMIF0_OVC2EH2	DSMIF0 upper threshold overcurrent detection 2 CH2	—	—	8	—	—
		DSMIF0_OVC0WN0	DSMIF0 Overcurrent Detection Window Notification 0 CH0	—	—	9	—	—
		DSMIF0_OVC1WN0	DSMIF0 Overcurrent Detection Window Notification 1 CH0	—	—	10	—	—
		DSMIF0_OVC2WN0	DSMIF0 Overcurrent Detection Window Notification 2 CH0	—	—	11	—	—
		DSMIF0_OVC3WN0	DSMIF0 Overcurrent Detection Window Notification 3 CH0	—	—	12	—	—
		DSMIF0_OVC0WN1	DSMIF0 Overcurrent Detection Window Notification 0 CH1	—	—	13	—	—
		DSMIF0_OVC1WN1	DSMIF0 Overcurrent Detection Window Notification 1 CH1	—	—	14	—	—
		DSMIF0_OVC2WN1	DSMIF0 Overcurrent Detection Window Notification 2 CH1	—	—	15	—	—
		DSMIF0_OVC3WN1	DSMIF0 Overcurrent Detection Window Notification 3 CH1	—	—	16	—	—
		DSMIF0_OVC0WN2	DSMIF0 Overcurrent Detection Window Notification 0 CH2	—	—	17	—	—
		DSMIF0_OVC1WN2	DSMIF0 Overcurrent Detection Window Notification 1 CH2	—	—	18	—	—
		DSMIF0_OVC2WN2	DSMIF0 Overcurrent Detection Window Notification 2 CH2	—	—	19	—	—
		DSMIF0_OVC3WN2	DSMIF0 Overcurrent Detection Window Notification 3 CH2	—	—	20	—	—

Table 11.4 Error event table (4 of 9)

Module	UNIT/ CH	Name	Interrupt sources	Bit of PERIERR _xxx0*1	Bit of PERIERR _xxx1*2	Bit of PERIERR _xxx2*3	Bit of PERIERR _xxx3*4	Bit of CPU0ERR _xxx*5
DSMIF	UNIT1	DSMIF1_OVC1EL0	DSMIF1 lower threshold overcurrent detection 1 CH3	—	—	21	—	—
		DSMIF1_OVC1EH0	DSMIF1 upper threshold overcurrent detection 1 CH3	—	—	22	—	—
		DSMIF1_OVC2EL0	DSMIF1 lower threshold overcurrent detection 2 CH3	—	—	23	—	—
		DSMIF1_OVC2EH0	DSMIF1 upper threshold overcurrent detection 2 CH3	—	—	24	—	—
		DSMIF1_OVC1EL1	DSMIF1 lower threshold overcurrent detection 1 CH4	—	—	25	—	—
		DSMIF1_OVC1EH1	DSMIF1 upper threshold overcurrent detection 1 CH4	—	—	26	—	—
		DSMIF1_OVC2EL1	DSMIF1 lower threshold overcurrent detection 2 CH4	—	—	27	—	—
		DSMIF1_OVC2EH1	DSMIF1 upper threshold overcurrent detection 2 CH4	—	—	28	—	—
		DSMIF1_OVC1EL2	DSMIF1 lower threshold overcurrent detection 1 CH5	—	—	29	—	—
		DSMIF1_OVC1EH2	DSMIF1 upper threshold overcurrent detection 1 CH5	—	—	30	—	—
		DSMIF1_OVC2EL2	DSMIF1 lower threshold overcurrent detection 2 CH5	—	—	31	—	—
		DSMIF1_LTCSE	DSMIF1 lower threshold current sum error	21	—	—	—	—
		DSMIF1_UTCSE	DSMIF1 upper threshold current sum error	22	—	—	—	—
		DSMIF1_OVC0EL0	DSMIF1 lower threshold overcurrent detection 0 CH3	23	—	—	—	—
		DSMIF1_OVC0EL1	DSMIF1 lower threshold overcurrent detection 0 CH4	24	—	—	—	—
		DSMIF1_OVC0EL2	DSMIF1 lower threshold overcurrent detection 0 CH5	25	—	—	—	—
		DSMIF1_OVC0EH0	DSMIF1 upper threshold overcurrent detection 0 CH3	26	—	—	—	—
		DSMIF1_OVC0EH1	DSMIF1 upper threshold overcurrent detection 0 CH4	27	—	—	—	—
		DSMIF1_OVC0EH2	DSMIF1 upper threshold overcurrent detection 0 CH5	28	—	—	—	—

Table 11.4 Error event table (5 of 9)

Module	UNIT/ CH	Name	Interrupt sources	Bit of PERIERR _xxx0*1	Bit of PERIERR _xxx1*2	Bit of PERIERR _xxx2*3	Bit of PERIERR _xxx3*4	Bit of CPU0ERR _xxx*5
DSMIF	UNIT1	DSMIF1_SCDE0	DSMIF1 Short-circuit detection 0	29	—	—	—	—
		DSMIF1_SCDE1	DSMIF1 Short-circuit detection 1	30	—	—	—	—
		DSMIF1_SCDE2	DSMIF1 Short-circuit detection 2	31	—	—	—	—
		DSMIF1_OVC2EH2	DSMIF1 upper threshold overcurrent detection 2 CH5	—	—	—	0	—
		DSMIF1_OVC0WN0	DSMIF1 Overcurrent Detection Window Notification 0 CH3	—	—	—	1	—
		DSMIF1_OVC1WN0	DSMIF1 Overcurrent Detection Window Notification 1 CH3	—	—	—	2	—
		DSMIF1_OVC2WN0	DSMIF1 Overcurrent Detection Window Notification 2 CH3	—	—	—	3	—
		DSMIF1_OVC3WN0	DSMIF1 Overcurrent Detection Window Notification 3 CH3	—	—	—	4	—
		DSMIF1_OVC0WN1	DSMIF1 Overcurrent Detection Window Notification 0 CH4	—	—	—	5	—
		DSMIF1_OVC1WN1	DSMIF1 Overcurrent Detection Window Notification 1 CH4	—	—	—	6	—
		DSMIF1_OVC2WN1	DSMIF1 Overcurrent Detection Window Notification 2 CH4	—	—	—	7	—
		DSMIF1_OVC3WN1	DSMIF1 Overcurrent Detection Window Notification 3 CH4	—	—	—	8	—
		DSMIF1_OVC0WN2	DSMIF1 Overcurrent Detection Window Notification 0 CH5	—	—	—	9	—
		DSMIF1_OVC1WN2	DSMIF1 Overcurrent Detection Window Notification 1 CH5	—	—	—	10	—
		DSMIF1_OVC2WN2	DSMIF1 Overcurrent Detection Window Notification 2 CH5	—	—	—	11	—
		DSMIF1_OVC3WN2	DSMIF1 Overcurrent Detection Window Notification 3 CH5	—	—	—	12	—
BISS	UNIT0	BISS0_NER	BISS0 Error interrupt	—	—	—	13	—
	UNIT1	BISS1_NER	BISS1 Error interrupt	—	—	—	14	—

Table 11.4 Error event table (6 of 9)

Module	UNIT/ CH	Name	Interrupt sources	Bit of PERIERR _xxx0*1	Bit of PERIERR _xxx1*2	Bit of PERIERR _xxx2*3	Bit of PERIERR _xxx3*4	Bit of CPU0ERR _xxx*5
HDSL	UNIT0	HDSL0_estimator_on	HDSL0 Position Estimator activated	—	—	—	15	—
		HDSL0_safe_channel_err	HDSL0 Transmission error in safe channel 1	—	—	—	16	—
		HDSL0_safe_pos_err	HDSL0 Safe position not valid	—	—	—	17	—
		HDSL0_acceleration_err	HDSL0 Fast channel / position error	—	—	—	18	—
		HDSL0_acc_thr_err	HDSL0 Fast channel / position threshold error	—	—	—	19	—
		HDSL0_encoding_err	HDSL0 DSL message encoding error	—	—	—	20	—
		HDSL0_dev_thr_err	HDSL0 Estimator deviation threshold reached	—	—	—	21	—
	UNIT1	HDSL1_estimator_on	HDSL1 Position Estimator activated	—	—	—	22	—
		HDSL1_safe_channel_err	HDSL1 Transmission error in safe channel 1	—	—	—	23	—
		HDSL1_safe_pos_err	HDSL1 Safe position not valid	—	—	—	24	—
		HDSL1_acceleration_err	HDSL1 Fast channel / position error	—	—	—	25	—
		HDSL1_acc_thr_err	HDSL1 Fast channel / position threshold error	—	—	—	26	—
		HDSL1_encoding_err	HDSL1 DSL message encoding error	—	—	—	27	—
		HDSL1_dev_thr_err	HDSL1 Estimator deviation threshold reached	—	—	—	28	—
AFMT	UNIT0	AFMT0_TMOUT	AFMT0 Time out error	—	—	—	29	—
	UNIT1	AFMT1_TMOUT	AFMT1 Time out error	—	—	—	30	—
DOC	CH0	DOC_DOPCI	DOC interrupt	—	0	—	—	—
SYSRAM	UNIT0	SRAM0_IE1	System SRAM0 ECC 1 bit error	—	1	—	—	—
		SRAM0_IE2	System SRAM0 ECC 2 bits error	—	2	—	—	—
		SRAM0_OVF	System SRAM0 ECC error address capture overflow	—	3	—	—	—
	UNIT1	SRAM1_IE1	System SRAM1 ECC 1 bit error	—	4	—	—	—
		SRAM1_IE2	System SRAM1 ECC 2 bits error	—	5	—	—	—
		SRAM1_OVF	System SRAM1 ECC error address capture overflow	—	6	—	—	—
BUS	—	BUSERR	Bus error	—	13	—	—	—

Table 11.4 Error event table (7 of 9)

Module	UNIT/ CH	Name	Interrupt sources	Bit of PERIERR _xxx0*1	Bit of PERIERR _xxx1*2	Bit of PERIERR _xxx2*3	Bit of PERIERR _xxx3*4	Bit of CPU0ERR _xxx*5
MPU	UNIT7	MPU_SHOST	Master MPU for Serial Host I/F error	—	15	—	—	—
	UNIT0	MPU_DMACR0	Master MPU Read channel for DMAC0 error	—	17	—	—	—
		MPU_DMACW0	Master MPU Write channel for DMAC0 error	—	18	—	—	—
	UNIT1	MPU_DMACR1	Master MPU Read channel for DMAC1 error	—	19	—	—	—
		MPU_DMACW1	Master MPU Write channel for DMAC1 error	—	20	—	—	—
	UNIT2	MPU_GMACR	Master MPU Read channel for GMAC error	—	21	—	—	—
		MPU_GMACW	Master MPU Write channel for GMAC error	—	22	—	—	—
	UNIT3	MPU_USBH	Master MPU for USB Host error	—	23	—	—	—
	UNIT4	MPU_USBF	Master MPU for USB Function error	—	24	—	—	—
	UNIT6	MPU_DBGR	Master MPU Read channel for Debug error	—	27	—	—	—
		MPU_DBGW	Master MPU Write channel for Debug error	—	28	—	—	—
Cortex-R52	CPU0	CPU0_ERREVENT0	Correctable memory error detected from L1 instruction cache, L1 data cache, ATCM, BTCM, CTCM, or flash. The PMU mnemonic is KITE_CORE_ERR_MEM.	—	—	—	—	0
		CPU0_ERREVENT1	Fatal memory error occurred from ATCM, BTCM, CTCM, or flash. The PMU mnemonic is KITE_FAT_ERR_MEM.	—	—	—	—	1
		CPU0_ERREVENT2	Correctable data payload bus error occurred from AXIM or LLPP. The PMU mnemonic is KITE_BUS_COR_DATA.	—	—	—	—	2
		CPU0_ERREVENT3	Fatal data payload bus error occurred from AXIM or LLPP.	—	—	—	—	3
		CPU0_ERREVENT4	Fatal READY signal bus error occurred from AXIM, Flash, or LLPP.	—	—	—	—	4
		CPU0_ERREVENT5	Fatal non-protocol bus error occurred from AXIM, Flash, or LLPP. These are payload errors. The PMU mnemonic is KITE_BUS_FAT_ANY.	—	—	—	—	5

Table 11.4 Error event table (8 of 9)

Module	UNIT/ CH	Name	Interrupt sources	Bit of PERIERR _xxx0*1	Bit of PERIERR _xxx1*2	Bit of PERIERR _xxx2*3	Bit of PERIERR _xxx3*4	Bit of CPU0ERR _xxx*5
Cortex-R52	CPU0	CPU0_ERREVENT6	Fatal protocol bus error occurred from AXIM, Flash, or LLPP. These are LEN, ID, LAST, VALID and READY, and interconnect protection errors. The mnemonic is KITE_BUS_PROTOCOL_ ANY.	—	—	—	—	6
		CPU0_ERREVENT7	Correctable L1 instruction cache data or L1 instruction cache tag memory error.	—	—	—	—	7
		CPU0_ERREVENT8	Correctable L1 data cache data or L1 data cache tag memory error.	—	—	—	—	8
		CPU0_ERREVENT9	Correctable ATCM, BTCM, or CTCM memory error.	—	—	—	—	9
		CPU0_ERREVENT10	Fatal ATCM, BTCM, or CTCM memory error.	—	—	—	—	10
		CPU0_ERREVENT11	Correctable flash memory error.	—	—	—	—	11
		CPU0_ERREVENT12	Fatal flash memory error.	—	—	—	—	12
		CPU0_ERREVENT13	Timeout for main AXIM bus.	—	—	—	—	13
		CPU0_ERREVENT14	Timeout for Flash bus.	—	—	—	—	14
		CPU0_ERREVENT15	Timeout for LLPP bus.	—	—	—	—	15
		CPU0_ERREVENT16	Correctable memory error occurred L1 instruction cache, L1 data cache, ATCM, BTCM, CTCM, or flash, but was not recorded in the Error Record Registers (ERRs), because they were full or because it was overwritten by a fatal error.	—	—	—	—	16
		CPU0_ERREVENT17	Fatal memory error occurred L1 instruction cache, L1 data cache, ATCM, BTCM, CTCM, or flash, but was not recorded in the ERRs, because they were full.	—	—	—	—	17
		CPU0_ERREVENT18	Interrupts masked in Hyp mode for too long. The IMP_INTMONR watchdog is triggered.	—	—	—	—	18
		CPU0_ERREVENT19	A memory or bus fatal error was synchronously detected in Hyp mode.	—	—	—	—	19
		CPU0_ERREVENT20	An abort exception was taken because an EL2-controlled MPU fault generated an abort.	—	—	—	—	20

Table 11.4 Error event table (9 of 9)

Module	UNIT/ CH	Name	Interrupt sources	Bit of PERIERR _xxx0*1	Bit of PERIERR _xxx1*2	Bit of PERIERR _xxx2*3	Bit of PERIERR _xxx3*4	Bit of CPU0ERR _xxx*5
Cortex-R52	CPU0	CPU0_ERREVENT21	An Abort exception was taken because an EL1-controlled MPU fault generated an abort.	—	—	—	—	21
		CPU0_ERREVENT22	An Undefined exception was taken due to any cause.	—	—	—	—	22
		CPU0_ERREVENT23	A memory access marked in EL1-controlled MPU as Device was flushed because it was marked as Normal in EL2-controlled MPU.	—	—	—	—	23
		CPU0_ERREVENT24	Processor livelock because of hard errors or exception at exception vector. An exception of the same type and vector offset was consecutively taken 20 times.	—	—	—	—	24
		CPU0_ERREVENT25	Software running in EL2 unlocks TESTR1. TESTR1 is for testing purposes only.	—	—	—	—	25

Note 1. PERIERR_STAT0, PERIERR_CLR0, PERIERR_E0MSK0, PERIERR_E1MSK0, and PERIERR_RSTMSK0
Note 2. PERIERR_STAT1, PERIERR_CLR1, PERIERR_E0MSK1, PERIERR_E1MSK1, and PERIERR_RSTMSK1
Note 3. PERIERR_STAT2, PERIERR_CLR2, PERIERR_E0MSK2, PERIERR_E1MSK2, and PERIERR_RSTMSK2
Note 4. PERIERR_STAT3, PERIERR_CLR3, PERIERR_E0MSK3, PERIERR_E1MSK3, and PERIERR_RSTMSK3
Note 5. CPU0ERR_STAT, CPU0ERR_CLR, CPU0ERR_E0MSK, CPU0ERR_E1MSK, and CPU0ERR_RSTMSK

11.3.4 Noise Filter

Noise filter can be used for external interrupt signals. Noise filter has the following functions:

- Noise filter function with sampling cycles of 64 divisions, 32 divisions, 8 divisions, and 1 division for PCLKM
- Inversion/rising/falling/both edge detection method selection function

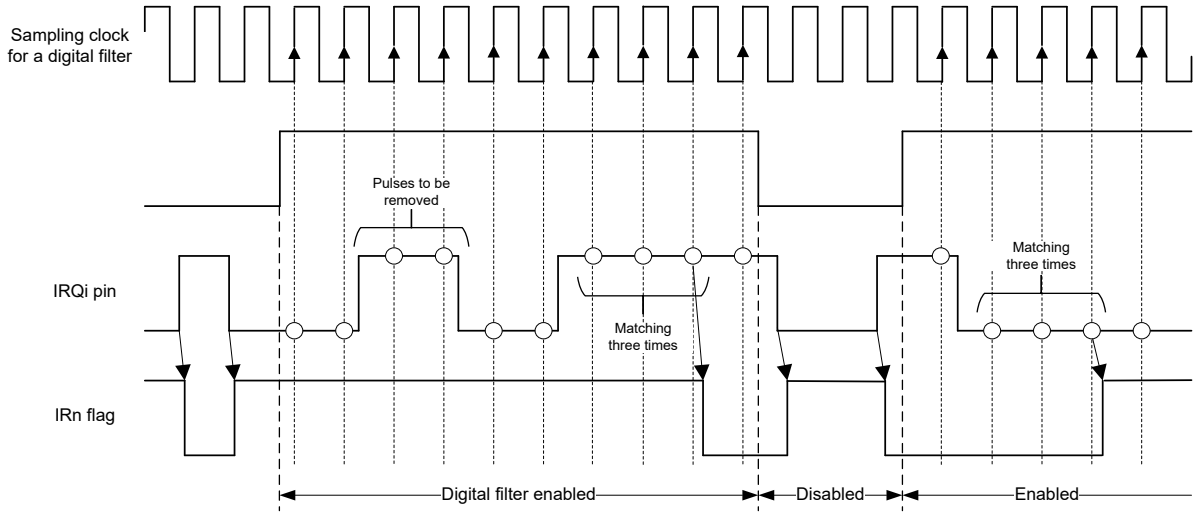
Noise filters are desirable for all signals as initial value. Set S_PORTNF_FLTSEL, NS_PORTNF_FLTSEL when using noise filter.

The digital noise filter function is provided for the external interrupt request IRQi pins (i = 0 to 15), NMI pin interrupts.

The digital noise filter samples input signals at the filter sampling clock (PCLKM) and pulsed with levels that only match once or twice are removed.

To use the digital noise filter for the IRQi pins (IRQ0 to IRQ15), set the sampling clock cycle (PCLKM, PCLKM/8, PLCKM/32, PCLKM/64) with the S(NS)_PORTNF_CLKSEL.CLSELx[1:0] bits, and set the S(NS)_PORTNF_FLTSEL.FLTx bits to 1.

Figure 11.2 shows an example of digital noise filter operation.



Operation example for S(NS)_PORTNF_MD.MDx[1:0] = 00b (operation in the Low level)

Figure 11.2 Digital noise filter operation example

12. Internal Buses

12.1 Overview

The LSI contains internal main bus, low latency peripheral port bus, and multiple peripheral buses. Master MPUs are implemented for bus masters except the Cortex-R52 so that access to the bus slaves can be controlled by the Cortex-R52. Additionally, physical paths to the bus slaves are separated among some peripherals. It allows CPU access to avoid access conflict from bus masters except the Cortex-R52 physically. Table 12.1 lists the specifications of internal buses and Figure 12.1 shows the internal bus configuration.

Table 12.1 Internal buses specifications

Internal bus type		Description
Internal main bus	• Bus masters Cortex-R52 (CPU0), DMAC (unit 0, unit 1), GMAC, USBHC, USBf, CoreSight, SHOSTIF	• Operates in synchronization with ICLK • Bus protocol: AMBA AXI • Priority order decision: Round-robin
Low latency peripheral port (LLPP) bus	• Bus masters Cortex-R52 (CPU0), internal main bus (DMAC (unit 0, unit 1), CoreSight (AXI-AP))	• Operates in synchronization with ICLK • Bus protocol: AMBA AXI • Priority order decision: Fixed priority (Cortex-R52 > internal main bus)
Peripheral bus 0	MTU3, GPT (ch0 to ch6), TFU, DSMIF, POE3, POEG (unit 0), ADC12 (unit 0, unit1), Encoder Subsytem (ENDAT, BISS, HDSL, AFMT) (unit 0, unit 1), ENCOUT	• LLPP peripheral (LLPP) • Operates in synchronization with PCLKH
Peripheral bus 1	GPT (ch7 to ch13), SCI (ch0 to ch4), SPI (ch0 to ch2), CRC (unit 0), CANFD	• Non-Safety peripheral (NONSAFETY) • Operates in synchronization with PCLKM
Peripheral bus 2	CMT, CMTW, WDT, IIC (ch0, ch1), DOC, TSU, POEG (unit 1)	• Non-Safety peripheral (NONSAFETY) • Operates in synchronization with PCLKL
Peripheral bus 3	GPT (ch14 to ch17), SCI (ch5), SPI (ch3), CRC (unit 1)	• Safety peripheral (SAFETY) • Operates in synchronization with PCLKM
Peripheral bus 4	IIC (ch2), RTC, POEG (unit 2)	• Safety peripheral (SAFETY) • Operates in synchronization with PCLKL
Peripheral bus 5	DMAC	• Non-Safety peripheral (NONSAFETY) • Operates in synchronization with PCLKH
Peripheral bus 6	GMAC, ESC, Ethernet Subsystem registers, USBHC, USBf	• Non-Safety peripheral (NONSAFETY) • Operates in synchronization with PCLKM
Cortex-R52 CPU0 AXIS	Cortex-R52 CPU0 ATCM, BTCM	• Operates in synchronization with ICLK
BSC	• External memory space • Related registers	• Operates in synchronization with PCLKM
xSPI	• External memory space • Related registers	• Operates in synchronization with PCLKH
System SRAM	• System SRAM memory space • Related registers	• Operates in synchronization with PCLKH
System Control	ICU, ELC, clock controller, reset controller, master MPU	• Operates in synchronization with PCLKM
HOSTIF Control	MBXSEM, SHOSTIF registers	• Operates in synchronization with PCLKH
IO port		• Operates in synchronization with PCLKM
OTP		• Operates in synchronization with PCLKM
Security		• Operates in synchronization with PCLKM
CoreSight		• Operates in synchronization with PCLKH

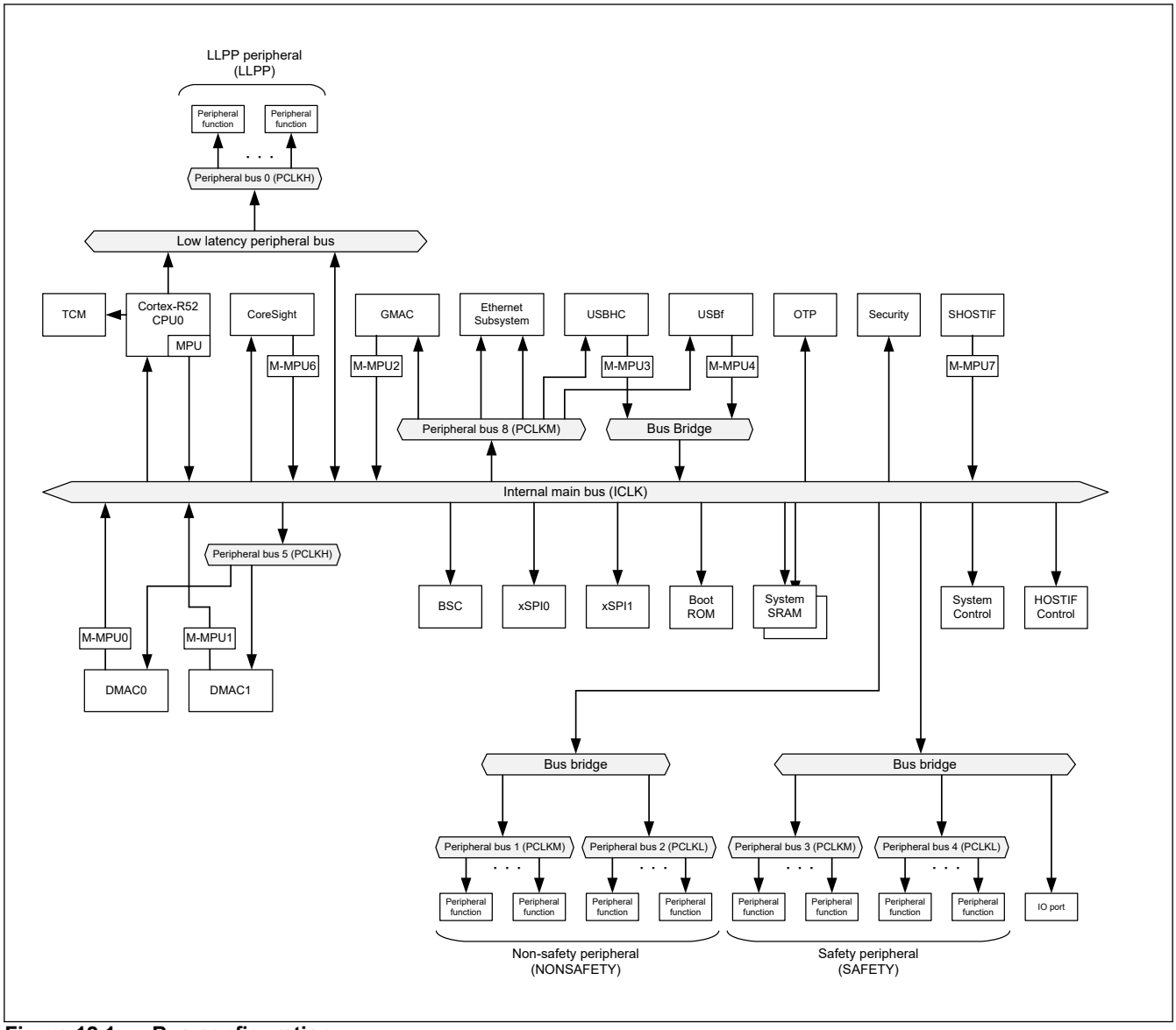


Figure 12.1 Bus configuration

12.2 Register Descriptions

12.2.1 STADDm : Master MPU Safety Region Start Address Register m (m = 0 to 7)

Base address: MPUi = 0x8128_1100 + 0x0100 × i (i = 0 to 4 and 6)
MPU7 = 0x8128_1C00

Offset address: 0x00 + 0x10 × m

Bit position:	31																					10								1	0
Bit field:	STADDR[21:0]																					—	—	—	—	—	—	—	—	WRPR	RDPR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*1	*1

Bit	Symbol	Function	R/W
0	RDPR	Enable read protection for region m of master MPU 0: Disable read protection. Read access to the safety region is permitted. 1: Enable read protection. Read access to the safety region is protected and error is generated.	R/W

Bit	Symbol	Function	R/W
1	WRPR	Enable write protection for region m of master MPU 0: Disable write protection. Write access to the safety region is permitted. 1: Enable write protection. Write access to the safety region is protected and error is generated.	R/W
9:2	—	These bits are read as 0. The write value should be 0.	R/W
31:10	STADDR[21:0]	Start address for MPU region Set upper 22 bits of the 32-bit start address of the safety region to be protected access. Lower 10 bits of the 32-bit start address is set 0x000 automatically. For AXI type MPU (i = 0 to 2, and 6), STADDR[1:0] must be set 00b.	R/W

Note 1. Initial value depends on the number of m. The value is 1 for m = 0 and 0 for the others.

The STADDm register is a 32-bit register that controls the start address of the safety region. Access from the master module to the safety region which is specified is protected by the master MPU. Up to 7 regions (m = 0 to 7) can be defined for each master MPU. Boundary and granularity of the start address to be set is 1 KB for AHB type MPU (i = 3, 4, and 7) and 4 KB for AXI type MPU (i = 0 to 2, and 6).

12.2.2 ENDADDm : Master MPU Safety Region End Address Register m (m = 0 to 7)

Base address: MPUi = 0x8128_1100 + 0x0100 × i (i = 0 to 4 and 6)
MPU7 = 0x8128_1C00

Offset address: 0x04 + 0x10 × m

Bit position:	31																			10										0
Bit field:	ENDADDR[21:0]																			—	—	—	—	—	—	—	—	—	—	
Value after reset:	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	—	These bits are read as 0. The write value should be 0.	R/W
31:10	ENDADDR[21:0]	End address for MPU region Set upper 22 bits of the 32-bit end address of the safety region to be protected access. Lower 10 bits of the 32-bit end address is set 0x3FF automatically. For AXI type MPU (i = 0 to 2, and 6), ENDADDR[1:0] must be set 11b.	R/W

Note 1. Initial value depends on the number of m. The value is 1 for m = 0 and 0 for the others.

The ENDADDm register is a 32-bit register that controls the end address of the safety region. Access from the master module to the safety region which is specified is protected by the master MPU. Up to 7 regions (m = 0 to 7) can be defined for each master MPU. Boundary and granularity of the end address to be set is 1 KB for AHB type MPU (i = 3, 4, and 7) and 4 KB for AXI type MPU (i = 0 to 2, and 6).

12.2.3 ERRINF : Master MPU Error Information Register for AHB type

Base address: MPUj = 0x8128_1100 + 0x0100 × j (j = 3, 4)
MPU7 = 0x8128_1C00

Offset address: 0x80

Bit position:	31																												2	1	0
Bit field:	ERRADDR[29:0]																												RW	VALID	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VALID	Validity of Access Error Information Cleared to 0 when 0 is written. 0: Error information is invalid 1: Error information is valid	R/W

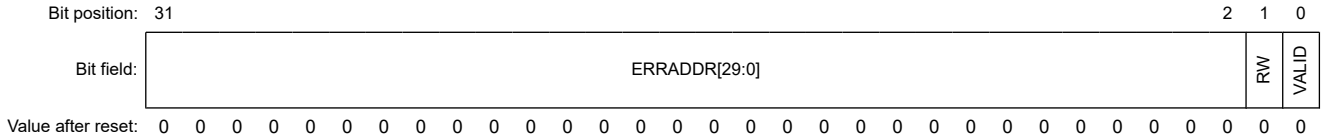
Bit	Symbol	Function	R/W
1	RW	Access error type 0: Read access 1: Write access	R
31:2	ERRADDR[29:0]	Access Error Address Indicates upper 30 bits of the 32-bit address. Lower 2 bits are masked.	R

The ERRINF is a 32-bit register to store the safety region access error information by master module. AHB type MPUj (j = 3, 4, and 7) provides one error information for both read and write.

12.2.4 ERRINF_x : Master MPU Error Information Register for AXI type (x = R, W)

Base address: MPUi = 0x8128_1100 + 0x0100 × i (i = 0 to 2, and 6)

Offset address: 0x80 (ERRINF_R)
0x84 (ERRINF_W)



Bit	Symbol	Function	R/W
0	VALID	Validity of access error information Cleared to 0 when 0 is written. 0: Error information is invalid 1: Error information is valid	R/W
1	RW	Access error type 0: Read access 1: Write access	R
31:2	ERRADDR[29:0]	Access error address Indicates upper 30 bits of the 32-bit address. Lower 2 bits are masked.	R

The ERRINF_x register is a 32-bit register to store the safety region access error information by master module. AXI type MPUi (i = 0 to 2, and 6) provides error information for each read and write channel separately. ERRINF_R is for read channel and ERRINF_W is for write channel.

12.3 Operation

12.3.1 Bus Connection

The internal main bus and low-latency peripheral bus of this LSI both have a multiple-layer structure. If bus masters request to access different bus slaves respectively, multiple accesses are processed in parallel. If bus masters request to access the same bus slave, priority order decision is performed and accesses are processed sequentially according to the priority order. Arbitration scheme for internal main bus is round-robin and for low-latency peripheral bus is fixed priority. Access to slaves on low-latency peripheral bus from the Cortex-R52 CPU0 is prioritized compared to the access from the bus masters through internal main bus.

Table 12.2 lists the connection between the bus masters and the bus slaves.

Table 12.2 Connection between bus masters and bus slaves (A: Accessible, —: Inaccessible) (1 of 2)

Bus slave	Bus master							
	Cortex-R52 CPU0	SHOSTIF	DMAC unit 0	DMAC unit 1	GMAC	USBHC	USBf	CoreSight
Peripheral bus 0 (LLPP)	A	—	A*1	A*1	—	—	—	A*1
Peripheral bus 1 (NONSAFETY)	A	—	A	A	—	—	—	A

Table 12.2 Connection between bus masters and bus slaves (A: Accessible, —: Inaccessible) (2 of 2)

Bus slave	Bus master							
	Cortex-R52 CPU0	SHOSTIF	DMAC unit 0	DMAC unit 1	GMAC	USBHC	USBf	CoreSight
Peripheral bus 2 (NONSAFETY)	A	—	A	A	—	—	—	A
Peripheral bus 3 (SAFETY)	A	—	—	—	—	—	—	—
Peripheral bus 4 (SAFETY)	A	—	—	—	—	—	—	—
Peripheral bus 5 (DMAC)	A	—	—	—	—	—	—	A
Peripheral bus 6 (Ethernet Subsystem, USBHC, USBf)	A	—	A*5	A*5	—	—	—	A
Cortex-R52 CPU0 AXIS (ATCM, BTCM)	A	—	A	A	A	A	A	A
BSC	A	—	A*4	A*4	A*4	A*4	A*4	A
xSPI	A	—	A*2	A*2	A*2	A*2	A*2	A
System SRAM	A	A*3	A*3	A*3	A*3	A*3	A*3	A*3
System Control	A	—	—	—	—	—	—	A
HOSTIF Control	A	A	—	—	—	—	—	—
IO port	A	—	—	—	—	—	—	—
OTP	A	—	—	—	—	—	—	—
Security	A	—	A	A	—	—	—	—
CoreSight	A	—	—	—	—	—	—	—

Note 1. Access through internal main bus is available. Priority order for these bus masters is round-robin.
Note 2. External memory space only. Control registers related to xSPI are not accessible.
Note 3. Memory space only. Control registers related to system SRAM are not accessible.
Note 4. External memory space only. Control registers related to BSC are not accessible.
Note 5. Process data RAM on ESC only.

12.3.2 Master MPU

Master MPU (MPUi, (i = 0 to 4 and 6 to 7)) is used to prevent master modules except the Cortex-R52 from accessing the safety region that is specified by STADDm and ENDADDm registers. When the master MPU detects the access to the safety region, access is terminated with an error response from the bus. Up to 7 regions can be protected for each master module. Boundary of the safety region is set with 1 KB granularity for AHB type MPU and 4 KB for AXI type MPU.

Master MPU stores error address and access attribution (read or write) to the Master MPU Error Information registers (ERRINF, ERRINF_R, ERRINF_W) when access error is detected. Interrupt is also generated to inform the CPU of the error occurrence.

Master MPU registers (STADDm, ENDADDm, ERRINF, and ERRINF_x) can be controlled by Cortex-R52 CPU0.

Table 12.3 lists the correspondence between MPUi and the master module. AXI type master MPU monitors access address and stores error address to the registers for read and write channels separately. On the other hand, AHB type master MPU monitors access address and stores error address to the same register for commonly read and write operation.

Table 12.3 Correspondence between master module and error information registers (1 of 2)

Master module	Type	Error information registers	
		Base address symbol	Register symbol
DMAC Unit 0	AXI	MPU0	ERRINF_R, ERRINF_W
DMAC Unit 1	AXI	MPU1	ERRINF_R, ERRINF_W
GMAC	AXI	MPU2	ERRINF_R, ERRINF_W
USB Host	AHB	MPU3	ERRINF
USB Function	AHB	MPU4	ERRINF

Table 12.3 Correspondence between master module and error information registers (2 of 2)

Master module	Type	Error information registers	
		Base address symbol	Register symbol
CoreSight (AXI-AP)	AXI	MPU6	ERRINF_R, ERRINF_W
SHOSTIF	AHB	MPU7	ERRINF

Safety region that is specified can be overlapped. If different attributions (read protection, write protection, or read/write protection) are set to the same area, combined protection is applied. Figure 12.2 shows an example of the safety region and attribution by the master MPU.

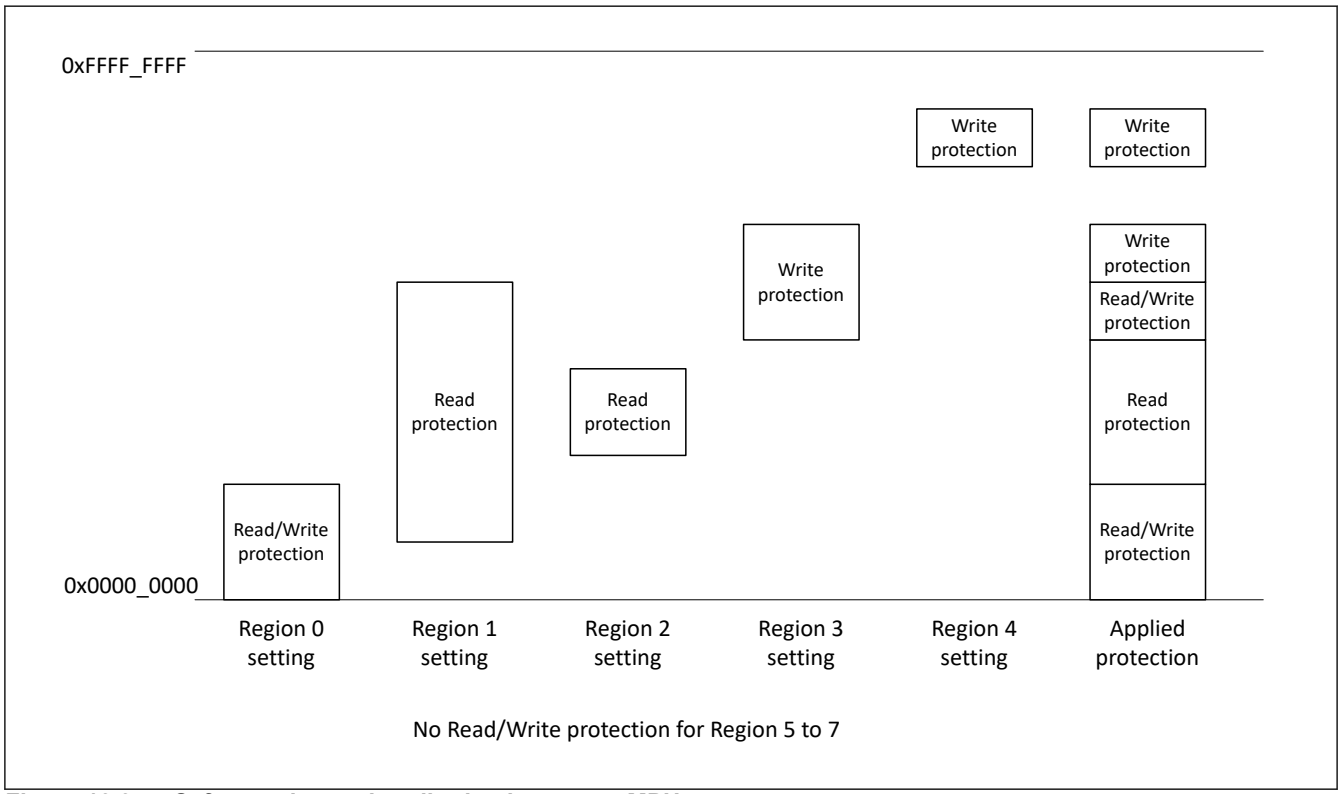


Figure 12.2 Safety region and attribution by master MPU

12.3.3 Interrupt

When a bus error or an MPU error occurs by reading or writing to the safety region, interrupt can be output. Table 12.4 lists the internal bus and master MPU interrupt sources. These interrupts are combined with peripheral error interrupts (PERI_ERR0 and PERI_ERR1) as error events with other error interrupts.

Table 12.4 Internal bus and master MPU interrupt sources (1 of 2)

Name	Interrupt sources	CPU0 GIC request	DMAC activation
BUS_ERRINT	Internal bus error	Possible*1	Not possible
MPU_DMACR0	Master MPU read channel for DMAC0 error	Possible*1	Not possible
MPU_DMACW0	Master MPU write channel for DMAC0 error	Possible*1	Not possible
MPU_DMACR1	Master MPU read channel for DMAC1 error	Possible*1	Not possible
MPU_DMACW1	Master MPU write channel for DMAC1 error	Possible*1	Not possible
MPU_GMACR	Master MPU read channel for GMAC error	Possible*1	Not possible
MPU_GMACW	Master MPU write channel for GMAC error	Possible*1	Not possible
MPU_USBH	Master MPU for USB host error	Possible*1	Not possible

Table 12.4 Internal bus and master MPU interrupt sources (2 of 2)

Name	Interrupt sources	CPU0 GIC request	DMAC activation
MPU_USBF	Master MPU for USB function error	Possible*1	Not possible
MPU_DBGR	Master MPU read channel for CoreSight error	Possible*1	Not possible
MPU_DBGW	Master MPU write channel for CoreSight error	Possible*1	Not possible
MPU_SHOSTIF	Master MPU for SHOSTIF error	Possible*1	Not possible

Note 1. These are not directly connected to CPU GIC but through peripheral error interrupts (PERI_ERR0 and PERI_ERR1).

12.4 Usage Notes

12.4.1 Release of Register Write Protection

The Master MPU registers are write protected. To release write protection, follow the steps below.

1. Write 0x0000A508 to PRCRS register (release write protection using PRCS3 bit).
2. Write value to the STADDm, ENDADDm, ERRINF, and ERRINF_x registers.
3. Write 0x0000A500 to PRCRS register (set write protection with PRCS3 bit).

12.4.2 Initial Setting of Master MPU

The initial setting of region 0 specifies all the area of memory map with read/write protection for every master MPU. This means the master modules cannot access any of the bus slaves. The Master MPU register must be changed from Cortex-R52 CPU0 before operating master modules.

12.4.3 New MPU Error Detection While Preserving Valid Error Information

When master MPU detects new error access while preserving valid error information (VALID bit = 1 in Master MPU error information registers), the error information is not updated to the new one unless the VALID bit is cleared to 0 by writing 0. Nevertheless, access to the safety region is protected. Interrupt is also not regenerated while the VALID bit is 1.

12.4.4 TCM Access via AXIS Interface of Cortex-R52

Bus masters can access to TCMs of Cortex-R52 CPU0 via AXIS interface in Cortex-R52 CPU0. AXIS has some limitations for data size and address alignment. Table 12.5 shows the limitation of TCM access via AXIS interface each bus master and Table 12.6 shows register setting to access TCM under limitation.

Table 12.5 Limitation of TCM access via AXIS interface (1 of 2)

Bus master	Description
CPU0 via AXIM	Instruction fetch is not possible. Data access Read ● Possible if region attribution is set to Device and LD instruction (8, 16, or 32 bits) is used to access. Write ● Possible if region attribution is set to Device and ST instruction (8, 16, or 32 bits) is used to access.
DMAC0	Descriptor access is not possible. Data access Read ● Possible if transfer size is 8, 16, 32, 64, 128, or 512 bits. Write ● Possible if transfer size is 8, 16, 32, 64, or 128 bits. ● Possible if transfer size is 512 bits and access address is 64-byte aligned. 256-bits transfer is not possible.
DMAC1	Descriptor access is not possible. Data access Read and Write ● Possible if transfer size is 8, 16, 32, 64, or 128 bits. 256-bit transfer is not possible.

Table 12.5 Limitation of TCM access via AXIS interface (2 of 2)

Bus master	Description
GMAC	Read and Write ● Possible if transfer is 32-bit ×1 beat and access address is 4-byte aligned. ● Possible if transfer is 32-bit ×4 beat and access address is 16-byte aligned. Other combination transfer is not possible.
USBHC USBf	Descriptor access is not possible. Data access Read and Write ● Possible if transfer size is 128 bits and access addresses is 16-byte aligned. Other combination transfer is not possible.
CoreSight (AXI-AP)	No limitation in the supported access.

Table 12.6 Register setting to access TCM from each bus master

Bus master	Register	Bit setting
DMAC0	CHCFG_n	DDS = 0x0 (8-bit), 0x1 (16-bit), 0x2 (32-bit), 0x3 (64-bit), 0x4 (128-bit), or 0x6 (512-bit) SDS = 0x0 (8-bit), 0x1 (16-bit), 0x2 (32-bit), 0x3 (64-bit), 0x4 (128-bit), or 0x6 (512-bit)
	CHEXT_n	DCA = 0x2 if CHCFG_n.DDS = 0x4 or 0x6 and destination is TCM SCA = 0x2 if CHCFG_n.SDS = 0x4 or 0x6 and source is TCM
DMAC1	CHCFG_n	DDS = 0x0 (8-bit), 0x1 (16-bit), 0x2 (32-bit), 0x3 (64-bit), or 0x4 (128-bit) SDS = 0x0 (8-bit), 0x1 (16-bit), 0x2 (32-bit), 0x3 (64-bit), or 0x4 (128-bit)
	CHEXT_n	DCA = 0x2 if CHCFG_n.DDS = 0x4 and destination is TCM SCA = 0x2 if CHCFG_n.SDS = 0x4 and source is TCM
GMAC	Bus_Mode	AAL = 0x1 USP = 0 RPBL = any FB = 0x1 PBL = 0x4
	AXI_Bus_Mode	BLEN16 = 0 BLEN8 = 0 BLEN4 = 0x1
USBHC	AHB_BUS_CTR	PROT_TYPE = 1xxxb ALIGN_ADDRESS = 11b
USBf	CHCFG_n	DDS = 0x4 (128-bit) SDS = 0x4 (128-bit)
	CHEXT_n	DPR = 1xxxb SPR = 1xxxb

13. Bus State Controller

13.1 Overview

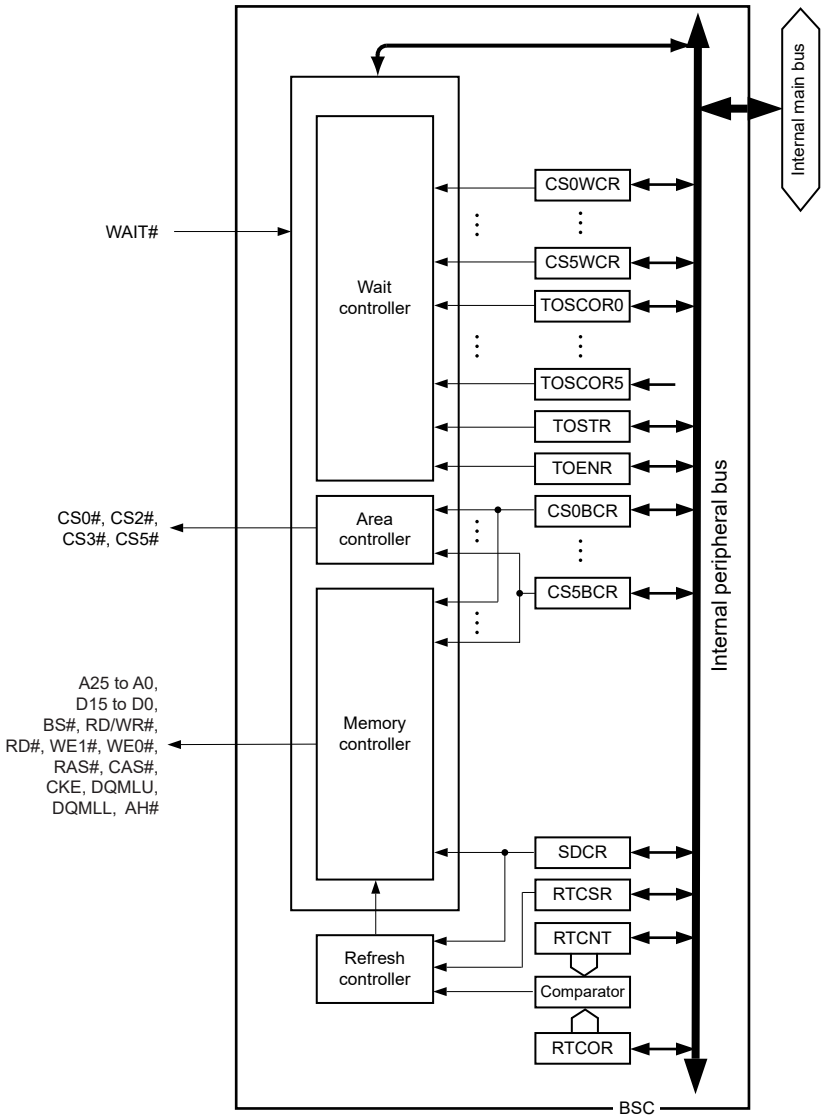
The bus state controller outputs control signals for various types of memory and external devices that are connected to the external address space. The functions of this module enable this LSI to connect directly with SRAM, SDRAM, other memory storage devices, and external devices.

Table 13.1 Bus state controller specifications

Item	Description
External address space	• A maximum of 64 MB for each of areas CS0, CS2, CS3, and CS5 • Can specify the following for each CS space: SRAM interface, SRAM interface with byte selection, burst ROM (clocked synchronous or asynchronous), MPX-I/O, and SDRAM memory. – Data bus width (8 or 16 bits) Insertion of wait cycles for each read access and write access. • Can set independent idle cycles for the following five cases: Read-write (in same space/different spaces), read-read (in same space/different spaces), and the first cycle is a write access.
Various interfaces	• SRAM interface Supports the interface that can directly connect to the SRAM.
	• Burst ROM interface (clocked asynchronous) High-speed access to the ROM that has the page mode function.
	• MPX-I/O interface Can directly connect to a peripheral LSI that needs an address/data multiplexing.
	• SDRAM interface – Can set the SDRAM in up to two areas[*1] Multiplex output for row address/column address. Efficient access by single read/single write. – High-speed access in bank-active mode Supports an auto-refresh and self-refresh. Supports a power-down mode. – Issues MRS and EMRS commands.
	• SRAM interface with byte selection Can connect directly to an SRAM with byte selection.
	• Burst ROM interface (clocked synchronous) Can connect directly to a burst ROM of the clocked synchronous type.
Refresh function	• Supports the auto-refresh and self-refresh functions • Specifies the refresh interval using the refresh counter and clock selection • Can execute concentrated refresh by specifying the refresh counts (1, 2, 4, 6, or 8) Usage as interval timer for refresh counter. • Generates an interrupt request at compare match
External WAIT# pin	• Detection of long wait state for access by the signal on the external WAIT# pin • A timeout detection condition is specifiable per CS space • Once timeout is detected, the external WAIT function is disabled and an external wait timeout detection interrupt is issued

Note 1. When connecting one SDRAM, set up the CS3 space for SDRAM or CS2 space and CS3 space for the SDRAM. CS2 space only is not supported.

Figure 13.1 is a block diagram of the bus state controller.



Note: CSnWCR: CSn Space Wait Control Register (n = 0, 2, 3, 5)
CSnBCR: CSn Space Bus Control Register (n = 0, 2, 3, 5)
SDCR: SDRAM Control Register
RTCSR: Refresh Timer Control/Status Register
RTCNT: Refresh Timer Counter
RTCOR: Refresh Time Constant Register
TOSCORn: Timeout Cycle Constant Register (n = 0, 2, 3, 5)
TOSTR: Timeout Status Register
TOENR: Timeout Enable Register

Figure 13.1 Block diagram of bus state controller

Table 13.2 lists the input/output pins of the bus state controller.

Table 13.2 Input/output pins of the bus state controller (1 of 2)

Name	I/O	Function
A25 to A0	Output	Address output pins
D15 to D0	I/O	Data input/output pins
BS#	Output	Status signal output pin that indicates the start of the bus cycle

Table 13.2 Input/output pins of the bus state controller (2 of 2)

Name	I/O	Function
CS0#, CS2#, CS3#, CS5#	Output	Chip select signal output pins
RD/WR#	Output	Strobe signal output pin that indicates a read or write access Connect this pin to WE# pins when SDRAM or SRAM with byte selection is connected.
RD#	Output	Strobe signal output pin that indicates a read signal (read data output enable signal)
AH#	Output	Functions as the address hold signal output pin when the MPX-I/O interface is used.
WE1#/DQMLU	Output	Write strobe signal output pin for D15 to D8 Connect this pin to the byte select pin when SRAM with byte selection is connected. Functions as the data mask enable signal output pin for D15 to D8 when SDRAM is connected.
WE0#/DQMLL	Output	Write strobe signal output pin for D7 to D0 Connect this pin to the byte select pin when SRAM with byte selection is connected. Functions as the data mask enable signal output pin for D7 to D0 when SDRAM is connected.
RAS#	Output	Row address strobe signal output pin for SDRAM. Connect this pin to RAS# pin of the SDRAM.
CAS#	Output	Column address strobe signal output pin for SDRAM. Connect this pin to CAS# pin of the SDRAM.
CKE	Output	Clock enable signal output pin for SDRAM. Connect this pin to CKE pin of the SDRAM.
WAIT#	Input	External wait control signal input pin used to insert a wait state into a bus cycle

13.2 Area Overview

13.2.1 Address Map

The kind of memory to be connected to the external address space and the data bus width are specified in each of CS0, CS2, CS3, and CS5 spaces. The address map for the external address space is listed below.

Table 13.3 Address map

Internal address	Space	Memory to be connected
0x5000_0000 to 0x53FF_FFFF	CS0 mirror	SRAM interface, SRAM with byte selection, burst ROM (asynchronous or synchronous)
0x5400_0000 to 0x57FF_FFFF	CS2 mirror	SRAM interface, SRAM with byte selection, SDRAM*1
0x5800_0000 to 0x5BFF_FFFF	CS3 mirror	SRAM interface, SRAM with byte selection, SDRAM*1
0x5C00_0000 to 0x5FFF_FFFF	CS5 mirror	SRAM interface, SRAM with byte selection, MPX-I/O
0x7000_0000 to 0x73FF_FFFF	CS0	SRAM interface, SRAM with byte selection, burst ROM (asynchronous or synchronous)
0x7400_0000 to 0x77FF_FFFF	CS2	SRAM interface, SRAM with byte selection, SDRAM
0x7800_0000 to 0x7BFF_FFFF	CS3	SRAM interface, SRAM with byte selection, SDRAM*1
0x7C00_0000 to 0x7FFF_FFFF	CS5	SRAM interface, SRAM with byte selection, MPX-I/O

Note 1. When connecting one SDRAM, set up the CS3 space for SDRAM or CS2 space and CS3 space for the SDRAM. CS2 space only is not supported.

13.2.2 Data Bus Width and Related Pin Setting for Each Area Depending on Boot Mode

The initial state of data bus width and settings of the pins related to this module depends on boot mode. For boot mode, see section 3, Operating Modes.

In 16-bit bus boot, the bus width of area 0 is automatically set to 16 bits, because this LSI is started up by the program stored in the ROM connected to area 0. For areas 2, 3, and 5, the bus width and the memory to be connected can be changed by the program. Immediately after a power-on reset in these modes, some of the address and data-bus signals and the CS0# and RD# signals are automatically selected by default as the functions of the corresponding pins, since these signals are required to read ROM data from area 0. With the exception of these pins, the general purpose pin function is selected by

default, and other required pin functions must be specified by software. Read access to area 0 is only permitted before the pin settings are completed.

In other boot modes, the state of areas 0, 2, 3, and 5 can be changed from the initial state by software, because the LSI does not use the ROM connected to area 0. Pin functions related to this module need to be set by software. Do not access external address spaces before the pin settings are completed.

Table 13.4 shows the initial state for individual boot modes and areas.

The sample access waveforms shown in this section include the pins such as BS#, RD/WR#, and WEn#. They are the waveforms when individual pin functions are enabled by the multi-function pin controller. For example, when 8-bit bus width is used in 16-bit bus boot mode, setting for pin A0 is needed.

For details on pin function settings, see section 16, I/O Ports.

Table 13.4 Initial states for individual boot modes and areas

Boot mode	Parameter	Area 0	Areas 2, 3, and 5
16-bit bus boot	Data bus width	Fixed to 16 bits. Not changeable.	16 bits. Can be changed by program.
	Settings of pins related to this module	Pins A20 to A1, D15 to D0, CS0#, and RD# are set automatically. Other pins need to be set by program.	
Others	Data bus width	16 bits. Can be changed by program.	
	Settings of pins related to this module	The initial value is the general I/O function. For external bus access, all the necessary pins need to be set by program.	

Note:
- In 16-bit bus boot, if an external memory that uses higher-order address lines than A21 is connected, the circuit board must include pull-down resistors for those address lines.
- The data-bus width may be limited by the type of memory in use. For details, see section 13.3.1. CSnBCR : CSn Space Bus Control Register (n = 0, 2, 3, 5).

13.3 Register Descriptions

13.3.1 CSnBCR : CSn Space Bus Control Register (n = 0, 2, 3, 5)

Base address: BSC = 0x8021_0000

Offset address: 0x004 + 0x4 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	IWW[2:0]			IWRWD[2:0]			IWRWS[2:0]			IWRRD[2:0]			IWRRS[2:0]		
Value after reset:	0	0	1	1	0	1	1	0	1	1	0	1	1	0	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	TYPE[2:0]			—	BSZ[1:0]		—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
10:9	BSZ[1:0]	Data Bus Width Specification Specify the data bus width of CS spaces. ● If area 5 is specified as MPX-I/O, setting these bits to 11b selects 8-bit or 16-bit bus width according to the address settings of the SZSEL bit in CS5WCR. Setting these bits to 01b or 10b selects the fixed bus width of 8-bit or 16-bit, respectively. ● In 16-bit bus boot mode, 10b is automatically written to the BSZ[1:0] bits of the CS0BCR register. ● If area 2 or area 3 is specified as SDRAM space, the bus width should be fixed to 16 bits. ● If area 0 is specified as clocked synchronous burst ROM space,the bus width should be fixed to 16 bits. 0 0: Setting prohibited 0 1: 8-bit size 1 0: 16-bit size 1 1: For MPX-I/O, selects bus width by address.	R/W
11	—	This bit is read as 1. The write value should be 1.	R/W
14:12	TYPE[2:0]	Memory Connected to a Space Specify the type of memory connected to a CS space. For details for memory type in each area, see Table 13.3. ● When connecting the burst ROM to the CS0 space in 16-bit bus boot, change the CS0WCR register to the settings according to the burst ROM, and then set TYPE[2:0] to the burst ROM setting. In other boot modes, memory access should be performed after setting CS0BCR and CS0WCR. ● Selecting SDRAM requires setting of I/O Buffer m Function Switching Register (DRCTLm) at the same time. For details, see section 16.3.6. DRCTLm : I/O Buffer m Function Switching Register (m = 00 to 24). 0 0 0: Normal space 0 0 1: Burst ROM (clock asynchronous) 0 1 0: MPX-I/O 0 1 1: SRAM with byte selection 1 0 0: SDRAM 1 0 1: Setting prohibited 1 1 0: Setting prohibited 1 1 1: Burst ROM (clock synchronous)	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
18:16	IWRRS[2:0]	Idle State Insertion between Read-Read Cycles in the Same CS Space Specify the number of idle cycles to be inserted after the access to an external memory that is connected to the CS space. The target cycle is a read-read cycle of which continuous access cycles are for the same CS space. 0 0 0: No idle cycle inserted 0 0 1: 1 idle cycle inserted 0 1 0: 2 idle cycles inserted 0 1 1: 4 idle cycles inserted 1 0 0: 6 idle cycles inserted 1 0 1: 8 idle cycles inserted 1 1 0: 10 idle cycles inserted 1 1 1: 12 idle cycles inserted	R/W
21:19	IWRRD[2:0]	Idle State Insertion between Read-Read Cycles in Different CS Spaces Specify the number of idle cycles to be inserted after the access to an external memory that is connected to the CS space. The target cycle is a read-read cycle of which continuous access cycles switch between different CS spaces. 0 0 0: No idle cycle inserted 0 0 1: 1 idle cycle inserted 0 1 0: 2 idle cycles inserted 0 1 1: 4 idle cycles inserted 1 0 0: 6 idle cycles inserted 1 0 1: 8 idle cycles inserted 1 1 0: 10 idle cycles inserted 1 1 1: 12 idle cycles inserted	R/W

Bit	Symbol	Function	R/W
24:22	IWRWS[2:0]	Idle State Insertion between Read-Write Cycles in the Same CS Space Specify the number of idle cycles to be inserted after the access to an external memory that is connected to the CS space. The target cycle is a read-write cycle of which continuous access cycles are for the same CS space. 0 0 0: No idle cycle inserted 0 0 1: 1 idle cycle inserted 0 1 0: 2 idle cycles inserted 0 1 1: 4 idle cycles inserted 1 0 0: 6 idle cycles inserted 1 0 1: 8 idle cycles inserted 1 1 0: 10 idle cycles inserted 1 1 1: 12 idle cycles inserted	R/W
27:25	IWRWD[2:0]	Idle State Insertion between Read-Write Cycles in Different CS Spaces Specify the number of idle cycles to be inserted after the access to an external memory that is connected to the CS space. The target access cycle is a read-write one in which continuous access cycles switch between different CS spaces. 0 0 0: No idle cycle inserted 0 0 1: 1 idle cycle inserted 0 1 0: 2 idle cycles inserted 0 1 1: 4 idle cycles inserted 1 0 0: 6 idle cycles inserted 1 0 1: 8 idle cycles inserted 1 1 0: 10 idle cycles inserted 1 1 1: 12 idle cycles inserted	R/W
30:28	IWW[2:0]	Idle Cycles between Write-Read Cycles and Write-Write Cycles These bits specify the number of idle cycles to be inserted after the access to an external memory that is connected to the CS space. The target access cycles are the write-read cycle and write-write cycle. 0 0 0: No idle cycle inserted 0 0 1: 1 idle cycle inserted 0 1 0: 2 idle cycles inserted 0 1 1: 4 idle cycles inserted 1 0 0: 6 idle cycles inserted 1 0 1: 8 idle cycles inserted 1 1 0: 10 idle cycles inserted 1 1 1: 12 idle cycles inserted	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The CSnBCR register specifies the memory connected to each CS space, the number of idle cycles between bus cycles, and the bus width.

Do not access external memory for the corresponding area until CSnBCR initial setting and pin setting that are required for accessing the external memory are completed.

Idle cycles may be inserted even when they are not specified. For details, see section 13.4.10. Wait between Access Cycles.

13.3.2 CSn Space Wait Control Register (CSnWCR) (n = 0, 2, 3, 5)

The CSnWCR register specifies various wait cycles for external memory access. The bit configuration of this register varies as shown below according to the memory type (TYPE2 to TYPE0) specified by the CSn space bus control register (CSnBCR). Specify CSnWCR before accessing the target area. Specify CSnBCR first, then specify CSnWCR.

13.3.2.1 CS0WCR_0 : CS0 Space Wait Control Register for Normal Space, SRAM with Byte Selection

Base address: BSC = 0x8021_0000

Offset address: 0x028

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	BAS	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SW[1:0]		WR[3:0]				WM	—	—	—	—	HW[1:0]	
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	HW[1:0]	Delay States from RD#, WEn# Negation to Address, CS0# Negation Specify the number of delay states from RD# and WEn# inactive to address and CS0# inactive. 0 0: 0.5 cycles 0 1: 1.5 cycles 1 0: 2.5 cycles 1 1: 3.5 cycles	R/W
5:2	—	These bits are read as 0. The write value should be 0.	R/W
6	WM	External Wait Mask Specification Specifies whether the external wait input is valid. The specification by this bit is valid even when the number of access wait cycle is 0. 0: External wait input is valid 1: External wait input is ignored	R/W
10:7	WR[3:0]	Number of Access Waits Specify the number of wait insertions that are necessary for read/write access. 0x0: No wait insertion 0x1: 1 cycle 0x2: 2 cycles 0x3: 3 cycles 0x4: 4 cycles 0x5: 5 cycles 0x6: 6 cycles 0x7: 8 cycles 0x8: 10 cycles 0x9: 12 cycles 0xA: 14 cycles 0xB: 18 cycles 0xC: 24 cycles Others: Setting prohibited	R/W
12:11	SW[1:0]	Number of Delay Cycles from Address, CSn# Assertion to RD#, WEn# Assertion Specify the number of delay cycle states from address and CS0# active to RD# and WEn# active. 0 0: 0.5 cycles 0 1: 1.5 cycles 1 0: 2.5 cycles 1 1: 3.5 cycles	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
17:16	—*1	Set these bits to 0 when the interfaces for SRAM or for SRAM with byte selection are used.	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
20	BAS*2	SRAM with Byte Selection Byte Access Select Specifies the WE# and RD/WR# signal timing when the SRAM interface with byte selection is used. 0: Activates the WEn# signal at the read/write timing and activates the RD/WR# signal during the write access cycle. 1: Activates the WEn# signal during the read/write access cycle and activates the RD/WR# signal at the write timing.	R/W
21	—*1	Set this bit to 0 when the interfaces for SRAM or for SRAM with byte selection are used.	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. In 16-bit bus boot mode, to connect the burst ROM to the CS0 space and switch to burst ROM interface after activation, set the TYPE[2:0] bits in CS0BCR after setting the burst number by the bits 20 and 21 and the burst wait cycle number by the bits 16 and 17. (For details on individual bits, see section 13.3.2.4. CS0WCR_1 : CS0 Space Wait Control Register for Burst ROM with Clocked Asynchronous and section 13.3.2.7. CS0WCR_2 : CS0 Space Wait Control Register for Burst ROM with Clocked Synchronous.) Do not write 1 to the reserved bits other than above bits.

Note 2. When BAS = 0, the write access timing of the SRAM interface with byte selection is the same as that of the SRAM interface. In write operation, data is written to the memory when the byte select pins (WEn# (n = 0, 1)) are active. In read operation, the active timing of the WEn# (n = 0, 1) pins are different from that of the SRAM interface and the byte selection signal is output from the WEn# (n = 0, 1) pins. When BAS = 1, the active timing of the WEn# (n = 0, 1) and RD/WR# pins varies. In write operation, data is written to the memory when the write enable pin (RD/WR#) is active. Set HW[1:0] for the write data hold time to the RD/WR# inactive timing to ensure the hold time. For details of access timing, see section 13.4.8. SRAM Interface with Byte Selection.

13.3.2.2 CSnWCR_0 : CSn Space Wait Control Register for Normal Space, SRAM with Byte Selection (n = 2, 3)

Base address: BSC = 0x8021_0000

Offset address: 0x030 + 0x4 × (n - 2)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	BAS	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	WR[3:0]				WM	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	WM	External Wait Mask Specification Specifies whether the external wait input is valid. The specification by this bit is valid even when the number of access waits is 0. 0: External wait input is valid 1: External wait input is ignored	R/W

Bit	Symbol	Function	R/W
10:7	WR[3:0]	Number of Access Waits Specify the number of waits that are necessary for read/write access. 0x0: No wait insertion 0x1: 1 cycle 0x2: 2 cycles 0x3: 3 cycles 0x4: 4 cycles 0x5: 5 cycles 0x6: 6 cycles 0x7: 8 cycles 0x8: 10 cycles 0x9: 12 cycles 0xA: 14 cycles 0xB: 18 cycles 0xC: 24 cycles Others: Setting prohibited	R/W
19:11	—	These bits are read as 0. The write value should be 0.	R/W
20	BAS	SRAM with Byte Selection Byte Access Select Specifies the WEn# and RD/WR# signal timing when the SRAM interface with byte selection is used. 0: Activates the WEn# signal at the read/write timing and activates the RD/WR# signal during the write access cycle. 1: Activates the WEn# signal during the read/write access cycle and activates the RD/WR# signal at the write timing.	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

13.3.2.3 CS5WCR : CS5 Space Wait Control Register for Normal Space, SRAM with Byte Selection, and MPX-I/O

Base address: BSC = 0x8021_0000

Offset address: 0x03C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	SZSEL	MPXW SBAS	—	WW[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SW[1:0]		WR[3:0]				WM	—	—	—	—	HW[1:0]	
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	HW[1:0]	Delay Cycles from RD#, WEn# to Address, CS5# Specify the number of delay states from RD# and WEn# inactive to address and CS5# inactive when area 5 is specified as the SRAM interface or the SRAM interface with byte selection. When area 5 is specified as the MPX-I/O interface, these bits specify the number of delay states from RD# and WEn# inactive to CS5# inactive. 0 0: 0.5 states 0 1: 1.5 states 1 0: 2.5 states 1 1: 3.5 states	R/W
5:2	—	These bits are read as 0. The write value should be 0.	R/W
6	WM	External Wait Mask Specification Specifies whether the external wait input is valid. The specification by this bit is valid even when the number of access wait is 0. 0: External wait input is valid 1: External wait input is ignored	R/W

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>10:7</td><td>WR[3:0]</td><td>Number of Read Access Waits
Specify the number of waits that are necessary for read access.
0x0: No wait insertion
0x1: 1 wait
0x2: 2 waits
0x3: 3 waits
0x4: 4 waits
0x5: 5 waits
0x6: 6 waits
0x7: 8 waits
0x8: 10 waits
0x9: 12 waits
0xA: 14 waits
0xB: 18 waits
0xC: 24 waits
Others: Setting prohibited</td><td>R/W</td></tr>
<tr><td>12:11</td><td>SW[1:0]</td><td>Number of Delay Cycles from Address, CS5# Assertion to RD#, WEn# Assertion
Specify the number of delay states from address and CS5# active to RD# and WEn# active when area 5 is specified as the SRAM interface or the SRAM interface with byte selection. When area 5 is specified as the MPX-I/O interface, these bits specify the number of delay states from the end of the address cycle (Ta3) to RD# and WEn# active.
0 0: 0.5 states
0 1: 1.5 states
1 0: 2.5 states
1 1: 3.5 states</td><td>R/W</td></tr>
<tr><td>15:13</td><td>—</td><td>These bits are read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>18:16</td><td>WW[2:0]</td><td>Number of Write Access Waits
Specify the number of wait insertions that are necessary for write access.
0 0 0: The same as WR[3:0] setting (number of read access waits)
0 0 1: No wait insertion
0 1 0: 1 wait
0 1 1: 2 waits
1 0 0: 3 waits
1 0 1: 4 waits
1 1 0: 5 waits
1 1 1: 6 waits</td><td>R/W</td></tr>
<tr><td>19</td><td>—</td><td>This bit is read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>20</td><td>MPXWSBAS</td><td>MPX-I/O Interface Address Cycle Wait and SRAM with Byte Selection Byte Access Select
CS5BCR.TYPE[2:0] = 010b specifies the address cycle wait for MPX-I/O interface. This bit (MPXW) setting is valid only when area 5 is specified as MPX-I/O.
CS5BCR.TYPE[2:0] = 011b specifies the WEn# and RD/WR# signal timing when the SRAM interface with byte selection is used. This bit (BAS) setting is valid only when area 5 is specified as SRAM with byte selection.
0: CS5BCR.TYPE[2:0] = 010b: Inserts no wait cycle
CS5BCR.TYPE[2:0] = 011b: Activates the WEn# signal at the read/write timing and activates the RD/WR# signal during the write access cycle.
1: CS5BCR.TYPE[2:0] = 010b: Inserts 1 wait cycle
CS5BCR.TYPE[2:0] = 011b: Activates the WEn# signal during the read/write access cycle and activates the RD/WR# signal at the write timing.</td><td>R/W</td></tr>
<tr><td>21</td><td>SZSEL</td><td>MPX-I/O Interface Bus Width Specification
Specifies an address to select the bus width when the BSZ[1:0] of the CS5BCR register are specified as 11b.
This bit is valid only when area 5 is specified as MPX-I/O.
The relationship between the SZSEL bit and bus width selected by A14 or A21 are summarized in the following table.
<table>
<tr><th>SZSEL</th><th>A14</th><th>A21</th><th>Bus Width</th></tr>
<tr><td rowspan="2">0</td><td>0</td><td>Not affected</td><td>8 bits</td></tr>
<tr><td>1</td><td>Not affected</td><td>16 bits</td></tr>
<tr><td rowspan="2">1</td><td>Not affected</td><td>0</td><td>8 bits</td></tr>
<tr><td>Not affected</td><td>1</td><td>16 bits</td></tr>
</table></td><td>R/W</td></tr>
</table>

Bit	Symbol	Function	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

13.3.2.4 CS0WCR_1 : CS0 Space Wait Control Register for Burst ROM with Clocked Asynchronous

Base address: BSC = 0x8021_0000

Offset address: 0x028

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	BST[1:0]		—	—	BW[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	W[3:0]				WM	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	WM	External Wait Mask Specification Specifies whether the external wait input is valid. The specification by this bit is valid even when the number of access waits is 0. 0: External wait input is valid 1: External wait input is ignored	R/W
10:7	W[3:0]	Number of Access Waits Specify the number of waits to be inserted in the first access cycle. 0x0: No wait insertion 0x1: 1 wait 0x2: 2 waits 0x3: 3 waits 0x4: 4 waits 0x5: 5 waits 0x6: 6 waits 0x7: 8 waits 0x8: 10 waits 0x9: 12 waits 0xA: 14 waits 0xB: 18 waits 0xC: 24 waits Others: Setting prohibited	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
17:16	BW[1:0]	Number of Waits during Burst Access Specify the number of waits to be inserted between the second or subsequent access cycles in burst access. 0 0: No wait insertion 0 1: 1 wait 1 0: 2 waits 1 1: 3 waits	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>21:20</td><td>BST[1:0]</td><td>Burst Count Specification
Specify the burst count for 16-byte or more access. Do not set BST[1:0] to 11b. For details, see Table 13.16.
<table>
<tr><th>Bus width</th><th>BST[1:0]</th><th>Burst count (16-byte access)</th></tr>
<tr><td rowspan="2">8 bits</td><td>00b</td><td>16 burst × 1 time</td></tr>
<tr><td>01b</td><td>4 burst × 4 times</td></tr>
<tr><td rowspan="3">16 bits</td><td>00b</td><td>8 burst × 1 time</td></tr>
<tr><td>01b</td><td>2 burst × 4 times</td></tr>
<tr><td>10b</td><td>4-4 or 2-4-2 burst</td></tr>
</table></td><td>R/W</td></tr>
<tr><td>31:22</td><td>—</td><td>These bits are read as 0. The write value should be 0.</td><td>R/W</td></tr>
</table>

13.3.2.5 CS2WCR_1 : CS2 Space Wait Control Register for SDRAM

Base address: BSC = 0x8021_0000

Offset address: 0x030

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	A2CL[1:0]		—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
8:7	A2CL[1:0]	CAS Latency for Area 2 Specify the CAS latency for area 2. 0 0: 1 0 1: 2 1 0: 3 1 1: 4	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	—	This bit is read as 1. The write value should be 1.	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: If only one area is connected to the SDRAM, specify area 3. In this case, specify area 2 as normal space or SRAM with byte selection.

13.3.2.6 CS3WCR_1 : CS3 Space Wait Control Register for SDRAM

Base address: BSC = 0x8021_0000

Offset address: 0x034

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	WTRP[1:0]		—	WTRCD[1:0]		—	A3CL[1:0]		—	—	TRWL[1:0]		—	WTRC[1:0]	
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	WTRC[1:0]*1	Number of Idle States from REF Command/Self-Refresh Release to ACTV/REF/MRS Command Specify the number of minimum idle cycles in the periods shown below. ● From the issuance of the REF command until the issuance of the ACTV/REF/MRS command ● From releasing self-refresh until the issuance of the ACTV/REF/MRS command The setting for areas 2 and 3 is common. 0 0: 2 states 0 1: 3 states 1 0: 5 states 1 1: 8 states	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
4:3	TRWL[1:0]*1	Number of Auto-Precharge Startup Wait Cycles Specify the number of minimum auto-precharge startup wait cycles as shown below. ● State number from the issuance of the WRITA command by this LSI until the activation of auto-precharge in the SDRAM. Equivalent to the state number from the issuance of the WRITA command until the issuance of the ACTV command. Confirm that how many states are required between the WRITA command receive in the SDRAM and the auto-precharge activation, referring to each SDRAM data sheet. And set the state number so as not to exceed the state number specified by this bit. ● State number from the issuance of the WRIT command until the issuance of the PRE command. This is the case when accessing another row address in the same bank in bank active mode. The setting for areas 2 and 3 is common. 0 0: No wait insertion 0 1: 1 state 1 0: 2 states 1 1: 3 states	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
8:7	A3CL[1:0]	CAS Latency for Area 3 Specify the CAS latency for area 3. 0 0: 1 0 1: 2 1 0: 3 1 1: 4	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
11:10	WTRCD[1:0]*1	Number of Waits between ACTV Command and READ(A)/WRIT(A) Command Specify the minimum number of waits from issuing the ACTV command to issuing the READ(A)/WRIT(A) command. The setting for areas 2 and 3 is common. 0 0: No wait insertion 0 1: 1 state 1 0: 2 states 1 1: 3 states	R/W
12	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
14:13	WTRP[1:0]*1	Number of Auto-Precharge Completion Wait States Specify the number of minimum precharge completion wait states as shown below. ● From the start of auto-precharge to issuing of ACTV command for the same bank ● From issuing of the PRE/PALL command to issuing of the ACTV command for the same bank ● Till entering the power-down mode or deep power-down mode ● From the issuing of PALL command to issuing REF command in auto refresh mode ● From the issuing of PALL command to issuing SELF command in self refresh mode The setting for areas 2 and 3 is common. 0 0: No wait insertion 0 1: 1 state 1 0: 2 states 1 1: 3 states	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. If both areas 2 and 3 are specified as SDRAM, WTRP[1:0], WTRCD[1:0], TRWL[1:0], and WTRC[1:0] bit settings are used in both areas in common.
If only one area is connected to the SDRAM, specify area 3. In this case, specify area 2 as SRAM interface or SRAM with byte selection.

13.3.2.7 CS0WCR_2 : CS0 Space Wait Control Register for Burst ROM with Clocked Synchronous

Base address: BSC = 0x8021_0000

Offset address: 0x028

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BW[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	W[3:0]				WM	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	WM	External Wait Mask Specification Specifies whether the external wait input is valid. The specification by this bit is valid even when the number of access waits is 0. 0: External wait input is valid 1: External wait input is ignored	R/W
10:7	W[3:0]	Number of Access Waits Specify the number of waits to be inserted in the first access cycle. 0x0: No wait insertion 0x1: 1 wait 0x2: 2 waits 0x3: 3 waits 0x4: 4 waits 0x5: 5 waits 0x6: 6 waits 0x7: 8 waits 0x8: 10 waits 0x9: 12 waits 0xA: 14 waits 0xB: 18 waits 0xC: 24 waits Others: Setting prohibited	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
17:16	BW[1:0]	Number of Burst Wait Cycles Specify the number of waits to be inserted between the second or subsequent access cycles in burst access. 0 0: No wait insertion 0 1: 1 wait 1 0: 2 waits 1 1: 3 waits	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

13.3.3 SDCR : SDRAM Control Register

Base address: BSC = 0x8021_0000

Offset address: 0x04C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	A2ROW[1:0]		—	A2COL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	DEEP	—	RFSH	RMODE	PDOWN	BACTV	—	—	—	A3ROW[1:0]		—	A3COL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	A3COL[1:0]	Number of Bits of Column Address for Area 3 Specify the number of bits of the column address for area 3. 0 0: 8 bits 0 1: 9 bits 1 0: 10 bits 1 1: Setting prohibited	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
4:3	A3ROW[1:0]	Number of Bits of Row Address for Area 3 Specify the number of bits of the row address for area 3. 0 0: 11 bits 0 1: 12 bits 1 0: 13 bits 1 1: Setting prohibited	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	BACTV	Bank Active Mode Specifies whether to access in auto-precharge mode (using READA and WRITA commands) or in bank active mode (using READ and WRIT commands). Bank active mode can be set only for area 3. When both areas 2 and 3 are set to SDRAM, specify the auto-precharge mode. 0: Auto-precharge mode (using READA and WRITA commands) 1: Bank active mode (using READ and WRIT commands)	R/W
9	PDOWN	Power-Down Mode Specifies whether the SDRAM will enter the power-down mode after the access to the SDRAM. With this bit being set to 1, after the SDRAM is accessed, the CKE signal is driven low and the SDRAM enters the power-down mode. 0: The SDRAM does not enter the power-down mode after being accessed. 1: The SDRAM enters the power-down mode after being accessed.	R/W

Bit	Symbol	Function	R/W
10	RMODE	Refresh Mode Specifies whether to perform auto-refresh or self-refresh when the RFSH bit is 1. When the RFSH bit is 1 and this bit is 1, self-refresh starts immediately. When the RFSH bit is 1 and this bit is 0, auto-refresh starts according to the contents that are set in registers RTCSR, RTCNT, and RTCOR. 0: Auto-refresh is performed 1: Self-refresh is performed	R/W
11	RFSH	Refresh Control Specifies the refresh control for the SDRAM. 0: No refresh 1: Refresh	R/W
12	—	This bit is read as 0. The write value should be 0.	R/W
13	DEEP	Deep Power-Down Mode This bit is valid for low-power SDRAM. If the RFSH or RMODE bit is set to 1 while this bit is set to 1, the deep power-down entry command is issued and the low-power SDRAM enters the deep power-down mode. 0: Self-refresh mode 1: Deep power-down mode	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
17:16	A2COL[1:0]	Number of Bits of Column Address for Area 2 Specify the number of bits of column address for area 2. 0 0: 8 bits 0 1: 9 bits 1 0: 10 bits 1 1: Setting prohibited	R/W
18	—	This bit is read as 0. The write value should be 0.	R/W
20:19	A2ROW[1:0]	Number of Bits of Row Address for Area 2 Specify the number of bits of row address of the SDRAM connected to area 2. 0 0: 11 bits 0 1: 12 bits 1 0: 13 bits 1 1: Setting prohibited	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The SDCR register specifies the method to refresh and access SDRAM, and the types of SDRAMs to be connected.

13.3.4 RTCSR : Refresh Timer Control/Status Register

Base address: BSC = 0x8021_0000

Offset address: 0x050

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CWP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	CMF	CMIE	CKS[2:0]			RRC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	RRC[2:0]	Refresh Count Specify the number of continuous refresh cycles, when the refresh request occurs after the coincidence of the values of the refresh timer counter (RTCNT) and the refresh time constant register (RTCOR). These bits can make the period of occurrence of refresh long. 0 0 0: 1 time 0 0 1: 2 times 0 1 0: 4 times 0 1 1: 6 times 1 0 0: 8 times Others: Setting prohibited	R/W
5:3	CKS[2:0]	Clock Select Select the clock input to count-up the refresh timer counter (RTCNT). 0 0 0: Stop the counting-up 0 0 1: CKIO/4 0 1 0: CKIO/16 0 1 1: CKIO/64 1 0 0: CKIO/256 1 0 1: CKIO/1024 1 1 0: CKIO/2048 1 1 1: CKIO/4096	R/W
6	CMIE	Compare Match Interrupt Enable Enables or disables CMF interrupt requests when the CMF bit in RTCSR is set to 1. 0: Disables CMF interrupt requests. 1: Enables CMF interrupt requests.	R/W
7	CMF	Compare Match Flag Indicates that a compare match occurs between the refresh timer counter (RTCNT) and refresh time constant register (RTCOR). This bit is set or cleared in the following conditions. 0: Clearing condition: When 0 is written in CMF after reading out RTCSR during CMF = 1. 1: Setting condition: When the condition RTCNT = RTCOR is satisfied.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
31:16	CWP[15:0]	Cancel Write Protection To modify the bits [7:0], write the value 0xA55A to the CWP[15:0] bits simultaneously, as a 32-bit unit.	W

The RTCSR register specifies various items about refresh for SDRAM.

The phase of the clock for incrementing the count in the refresh timer counter (RTCNT) is adjusted only by a power-on reset. Note that there is an error in the time until the compare match flag is set for the first time after the timer is started with the CKS[2:0] bits being set to a value other than 000b.

To modify the bits [7:0], write the value 0xA55A to the CWP[15:0] bits simultaneously, as a 32-bit unit.

13.3.5 RTCNT : Refresh Timer Counter

Base address: BSC = 0x8021_0000

Offset address: 0x054

Bit position:	31															16							7							0
Bit field:	CWP[15:0]															—	—	—	—	—	—	—	—	RFSHTC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RFSHTC[7:0]	Refresh Timer Counter	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
31:16	CWP[15:0]	Cancel Write Protection To modify the RFSHTC[7:0] bits, write the value 0xA55A to the CWP[15:0] bits simultaneously.	W

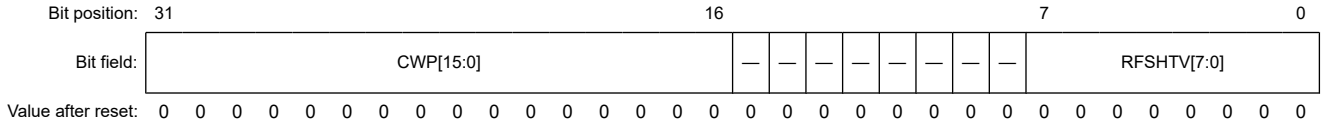
The RTCNT register has the refresh timer counter of 8-bit width that increments using the clock selected by the RTCSR.CKS[2:0] bits. When the RFSHTC[7:0] bits match the RTCOR.RFSHTV[7:0] bits, the RTCSR.CMF bit is set to 1 and RTCNT is automatically set to 0.

To modify the RFSHTC[7:0] bits, write the value 0xA55A to the CWP[15:0] bits simultaneously, as a 32-bit unit.

13.3.6 RTCOR : Refresh Time Constant Register

Base address: BSC = 0x8021_0000

Offset address: 0x058



Bit	Symbol	Function	R/W
7:0	RFSHTV[7:0]	Refresh time value	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
31:16	CWP[15:0]	Cancel Write Protection To modify the RFSHTV[7:0] bits, write the value 0xA55A to the CWP[15:0] bits simultaneously.	W

The RTCOR register sets the count value for issuing a memory refresh request. When the RFSHTC[7:0] bits match the RTCNT.RFSHTV[7:0] bits, the CMF bit in RTCSR is set to 1 and RTCNT is automatically set to 0.

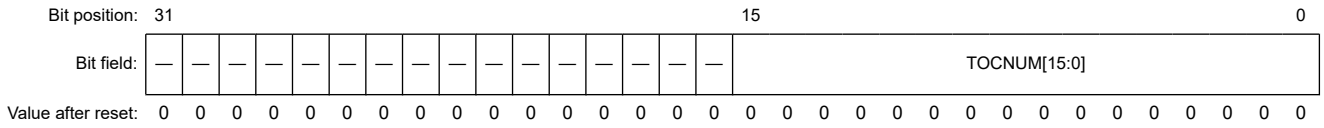
When the RFSH bit in SDCR is 1, a memory refresh request is issued by this matching signal. This request is maintained until the refresh operation is performed. If the request is not processed when the next matching occurs, the previous request is ignored. When the CMIE bit in RTCSR is set to 1, an interrupt request (BSC_CMI) is issued by this matching signal. The request continues to be output until the CMF bit in RTCSR is cleared. Clearing the CMF bit only affects the interrupt request and does not clear the refresh request. Therefore, a combination of refresh request and interval timer interrupt can be specified so that the number of refresh requests is counted using the timer interrupts while refresh is performed periodically.

To modify the RFSHTV[7:0] bits, write the value 0xA55A to the CWP[15:0] bits simultaneously, as a 32-bit unit.

13.3.7 TOSCORn : Timeout Cycle Constant Register n (n = 0, 2, 3, 5)

Base address: BSC = 0x8021_0000

Offset address: 0x060 + 0x4 × n



Bit	Symbol	Function	R/W
15:0	TOCNUM[15:0]	Timeout Cycle Number When the number of external wait cycles matches the setting value of these bits, the external wait timeout detection interrupt occurs and the wait state is forcibly interrupted to end the cycle of access. 0x0000 65536 (number of external wait cycles) : 0x0001 1 : ⋮ 0xFFFF 65535 :	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The TOSCORn register sets the timeout to forcibly interrupt the wait state and end the access cycle of the CSn space. This register is enabled when the WM bit of the CSn Space Wait Control Register (CSnWCR) is 0 and the corresponding bit of the Timeout Enable Register (TOENR) is 1. When the number of wait cycles due to the signal on the external wait input pin matches the setting of TOSCORn, the wait state is forcibly interrupted (external wait input is disabled) to end the cycle of access, the Timeout Status flag for the corresponding space in the Timeout Status Register (TOSTR) is set, and an external wait timeout detection interrupt request from the external WAIT# pin is generated. The external wait timeout detection interrupt request is retained until the corresponding bit in the TOENR register is set to 0, or 0 is written to the Timeout Status flag for the corresponding space. Note that timeout detection is enabled even while the Timeout Status flag for the corresponding space in the TOSTR register is 1, and the wait state is forcibly interrupted (external wait input is disabled) to end the cycle of access in response to a further timeout.

13.3.8 TOSTR : Timeout Status Register

Base address: BSC = 0x8021_0000

Offset address: 0x080

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	CS5T OSTF	—	CS3T OSTF	CS2T OSTF	—	CS0T OSTF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CS0TOSTF	CS0 Space Timeout Status Flag Status flag that indicates that the number of cycles of waiting due to the input on the external wait pin during access to the CS0 space has matched the setting of the CS0 space timeout cycle constant register (TOSCOR0). This bit is set or cleared in the following conditions. 0: Clearing condition When 0 is written in CS0TOSTF. 1: Setting condition When the WM bit in the CS0 space wait control register (CS0WCR) is 0 and the CS0TOEN bit in the timeout enable register (TOENR) is 1, the number of cycles of waiting due to the input on the external wait pin during access to the CS0 space has matched the setting of the TOSCOR0 register.	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	CS2TOSTF	CS2 Space Timeout Status Flag Status flag that indicates that the number of cycles of waiting due to the input on the external wait pin during access to the CS2 space has matched the setting of the CS2 space timeout cycle constant register (TOSCOR2). For the condition to set or clear this bit, see the description of CS0TOSTF.	R/W
3	CS3TOSTF	CS3 Space Timeout Status Flag Status flag that indicates that the number of cycles of waiting due to the input on the external wait pin during access to the CS3 space has matched the setting of the CS3 space timeout cycle constant register (TOSCOR3). For the condition to set or clear this bit, see the description of CS0TOSTF.	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	CS5TOSTF	CS5 Space Timeout Status Flag Status flag that indicates that the number of cycles of waiting due to the input on the external wait pin during access to the CS5 space has matched the setting of the CS5 space timeout cycle constant register (TOSCOR5). For the condition to set or clear this bit, see the description of CS0TOSTF.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

The TOSTR register is used for status of the external wait input. When the WM bit in the CSn space wait control register (CSnWCR) is 0 and the corresponding bit in the timeout enable register (TOENR) is 1 and the number of waits in response

to the signal on the external wait input matches the setting of TOSCORn, the timeout status flag for the corresponding space is set and an external wait timeout detection interrupt request is generated. The only writable value for the timeout status flags is 0, which clears the flag. Writing 1 to a flag is ignored.

13.3.9 TOENR : Timeout Enable Register

Base address: BSC = 0x8021_0000

Offset address: 0x084

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	CS5TOEN	—	CS3TOEN	CS2TOEN	—	CS0TOEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CS0TOEN	CS0 Space Timeout Detection Enable Specifies enabling or disabling the detection of timeout for waiting in the CS0 space. 0: The timeout detection is disabled. 1: The timeout detection is enabled.	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	CS2TOEN	CS2 Space Timeout Detection Enable Specifies enabling or disabling the detection of timeout for waiting in the CS2 space. 0: The timeout detection is disabled. 1: The timeout detection is enabled.	R/W
3	CS3TOEN	CS3 Space Timeout Detection Enable Specifies enabling or disabling the detection of timeout for waiting in the CS3 space. 0: The timeout detection is disabled. 1: The timeout detection is enabled.	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	CS5TOEN	CS5 Space Timeout Detection Enable Specifies enabling or disabling the detection of timeout for waiting in the CS5 space. 0: The timeout detection is disabled. 1: The timeout detection is enabled.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

The TOENR register is used to enable or disable timeout detection function for each space.

13.4 Operation

13.4.1 Access Size and Data Alignment

This LSI supports little endian, in which the least significant byte (LSB) is that in the direction of the 0th address. Data bus width can be selected from 8 bits and 16 bits for the normal memory and SRAM with byte selection. Data bus width is fixed to 16 bits for SDRAM. For MPX-I/O, the data bus width is fixed to either 8 or 16 bits, or made selectable as 8 bits or 16 bits by one of the address lines.

Data bus width varies depending on boot mode. For details, see section 13.2.2. Data Bus Width and Related Pin Setting for Each Area Depending on Boot Mode.

Data alignment is performed in accordance with the data bus width selected for the device. This also means that four read operations are required to read 32-bit data from a byte-width device. In this LSI, data alignment and conversion of data length is performed automatically between the respective interfaces.

Table 13.5 and Table 13.6 show the relationship between device data width and access unit.

Table 13.5 16-bit external device access and data alignment in little endian

Operation		Data bus		Strobe signals	
		D15 to D8	D7 to D0	WE1#, DQMLU	WE0#, DQMLL
8-bit access at address 0		—	Data 7 to 0	—	Active
8-bit access at address 1		Data 7 to 0	—	Active	—
8-bit access at address 2		—	Data 7 to 0	—	Active
8-bit access at address 3		Data 7 to 0	—	Active	—
16-bit access at address 0		Data 15 to 8	Data 7 to 0	Active	Active
16-bit access at address 2		Data 15 to 8	Data 7 to 0	Active	Active
32-bit access at address 0	1st access at address 0	Data 15 to 8	Data 7 to 0	Active	Active
	2nd access at address 2	Data 31 to 24	Data 23 to 16	Active	Active

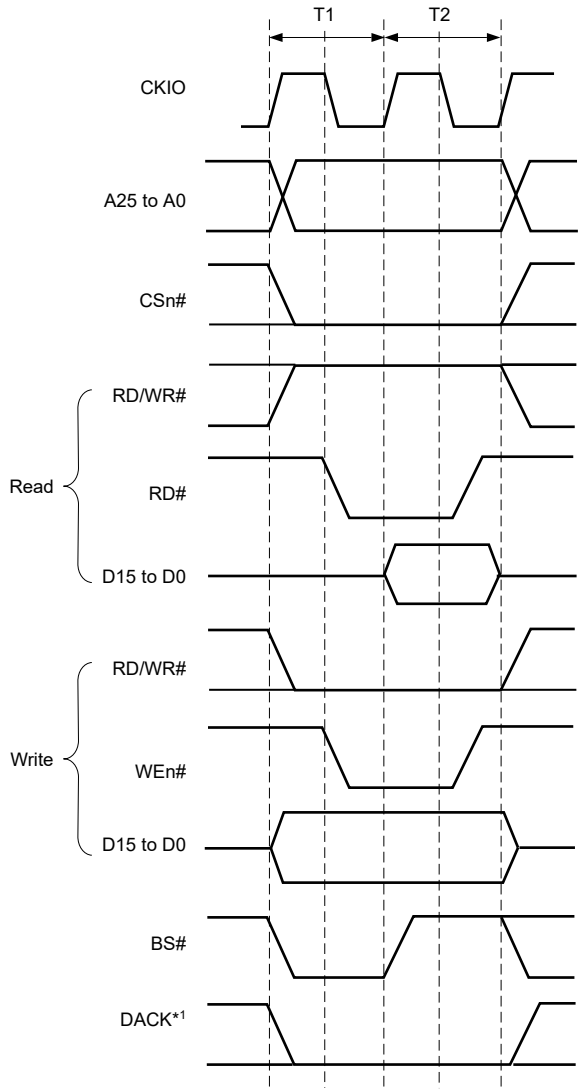
Table 13.6 8-bit external device access and data alignment in little endian

Operation		Data bus		Strobe signals	
		D15 to D8	D7 to D0	WE1#, DQMLU	WE0#, DQMLL
8-bit access at address 0		—	Data 7 to 0	—	Active
8-bit access at address 1		—	Data 7 to 0	—	Active
8-bit access at address 2		—	Data 7 to 0	—	Active
8-bit access at address 3		—	Data 7 to 0	—	Active
16-bit access at address 0	1st access at address 0	—	Data 7 to 0	—	Active
	2nd access at address 1	—	Data 15 to 8	—	Active
16-bit access at address 2	1st access at address 0	—	Data 7 to 0	—	Active
	2nd access at address 1	—	Data 15 to 8	—	Active
32-bit access at address 0	1st access at address 0	—	Data 7 to 0	—	Active
	2nd access at address 1	—	Data 15 to 8	—	Active
	3rd access at address 2	—	Data 23 to 16	—	Active
	4th access at address 3	—	Data 31 to 24	—	Active

13.4.2 SRAM Interface

(1) Basic Timing

For access to an SRAM interface, this LSI uses strobe signal output in consideration of the fact that mainly static RAM will be directly connected. When using SRAM with a byte-selection pin, see section 13.4.8. SRAM Interface with Byte Selection. Figure 13.2 shows the basic timings of SRAM interface access. A no-wait normal access is completed in two cycles. The BS# signal is activated for one state to indicate the start of a bus cycle.



Note 1. The waveform for DACKn is when active-low is specified.

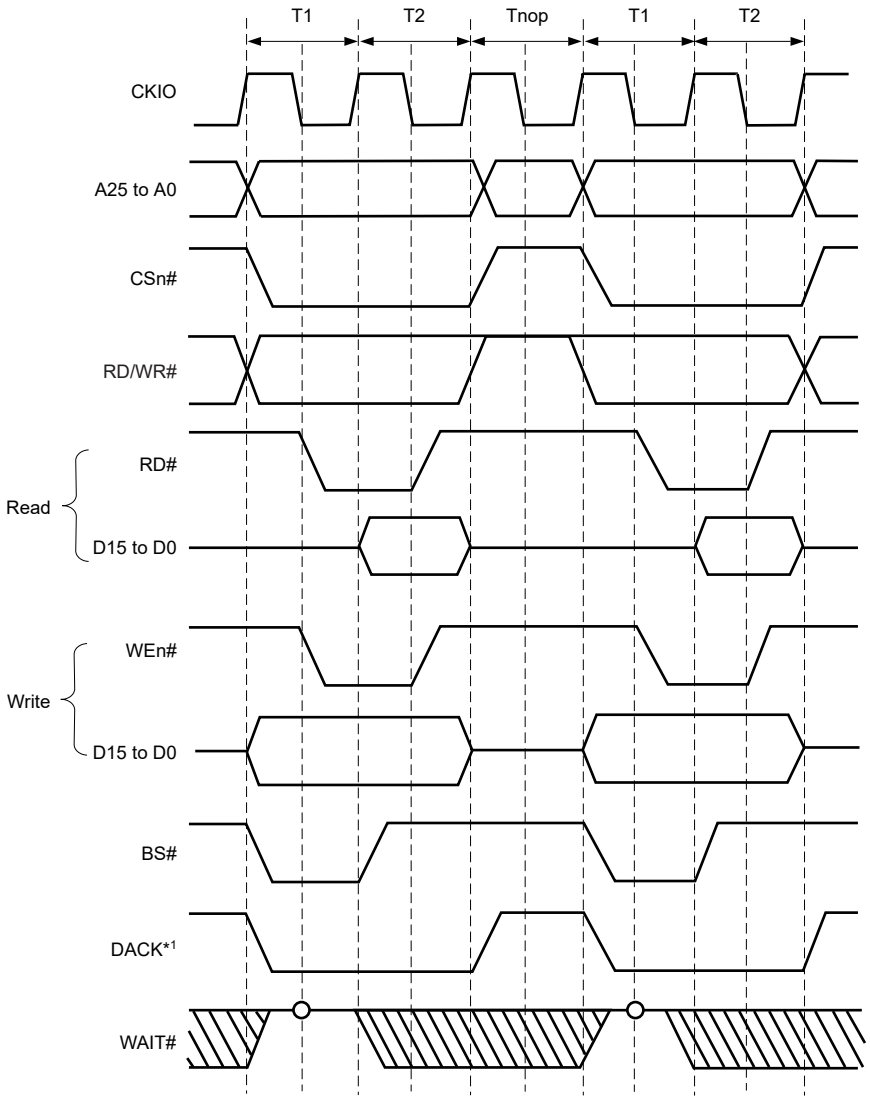
Figure 13.2 SRAM interface basic access timing (access wait 0)

There is no access size specification when reading. The correct access start address is output in the least significant bit of the address, but since there is no access size specification, 16 bits are always read in case of a 16-bit device. When writing, only the WEn# signal for the byte to be written is activated.

It is necessary to output the data that has been read using RD# when a buffer is established in the data bus. The RD/WR# signal is in a read state (high output) when no access has been carried out. Therefore, care must be taken when controlling the external data buffer with this signal, to avoid output collision.

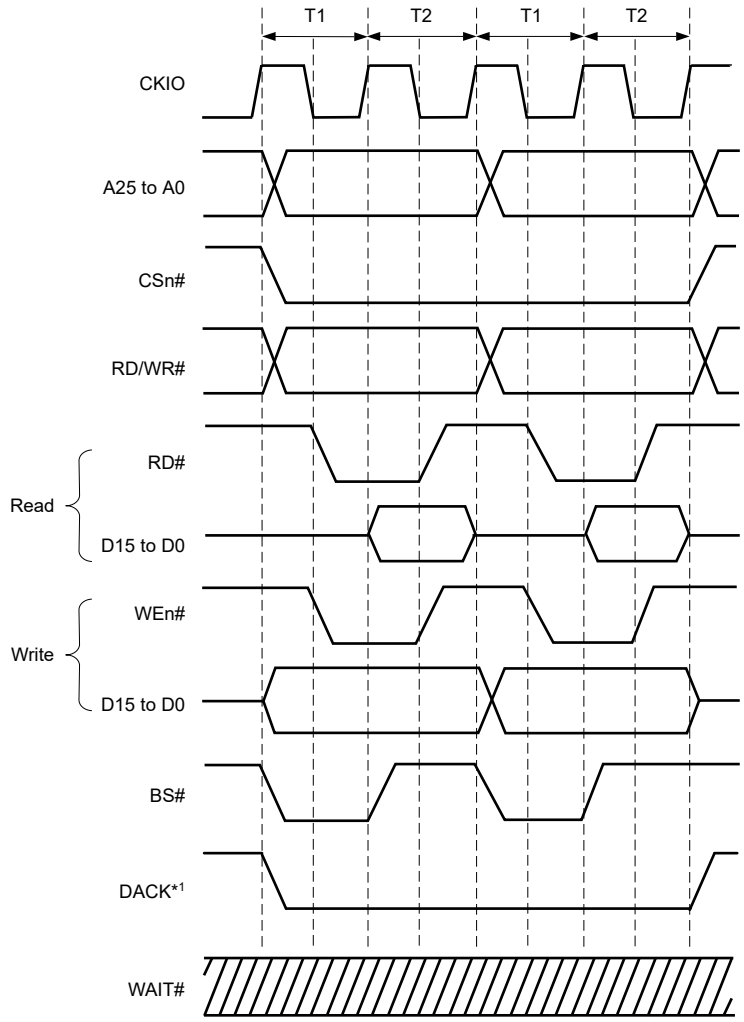
Figure 13.3 and Figure 13.4 show the basic timings in continuous access to the SRAM interface. If the WM bit in CSnWCR is cleared to 0, a Tnop cycle is inserted after the CSn space access to sample the external wait (Figure 13.3).

If the WM bit in CSnWCR is set to 1, external waits are ignored and no Tnop cycle is inserted (Figure 13.4).



Note 1. The waveform for DACKn is when active-low is specified.

Figure 13.3 Continuous access to the SRAM interface (1) (bus width = 16 bits, 32-bit access, CSnWCR.WM bit = 0 (access wait = 0, cycle wait = 0))



Note 1. The waveform for DACKn is when active-low is specified.

Figure 13.4 Continuous access to the SRAM interface (2) (bus width = 16 bits, 32-bit access, CSnWCR.WM bit = 1 (access wait = 0, cycle wait = 0))

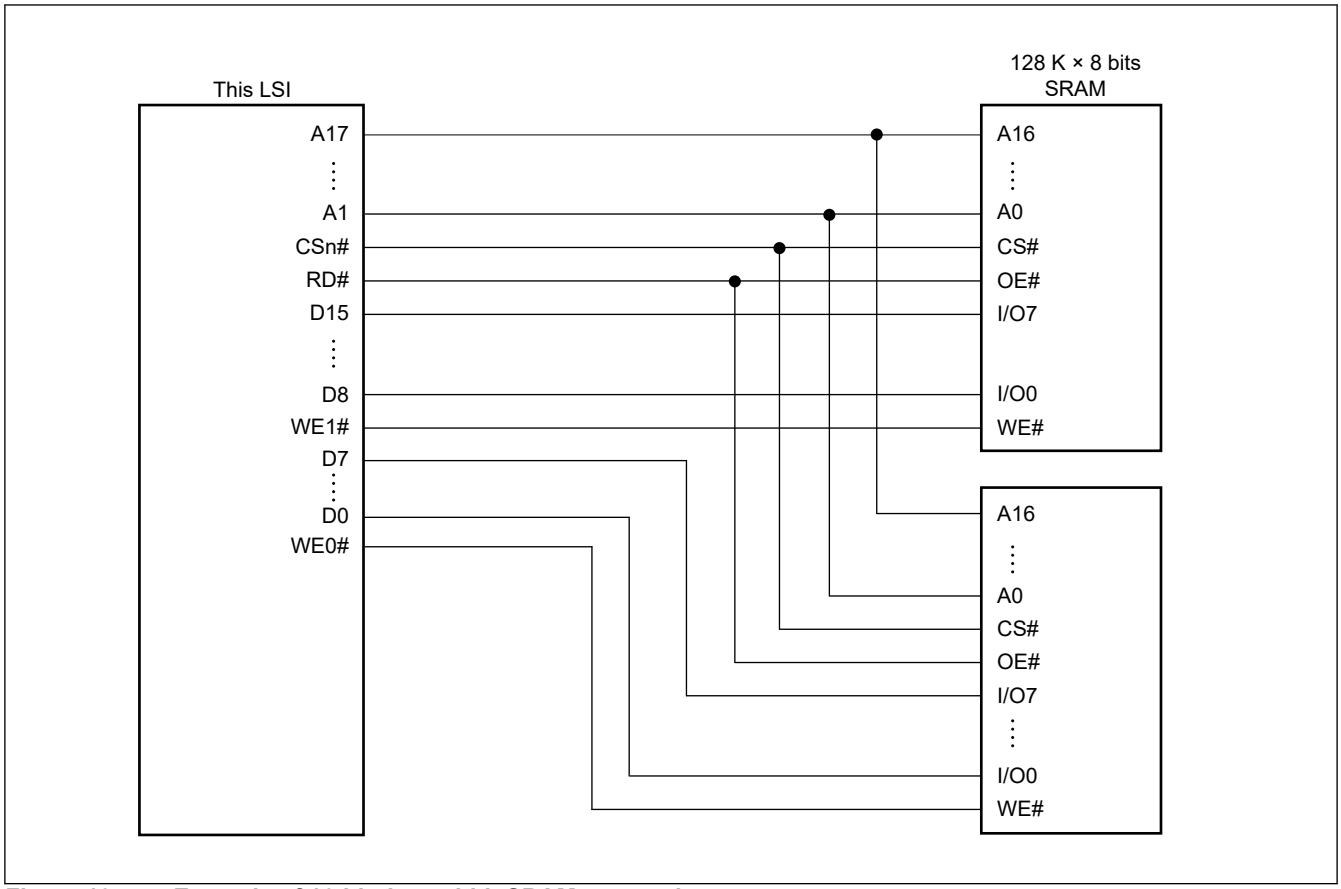


Figure 13.5 Example of 16-bit data-width SRAM connection

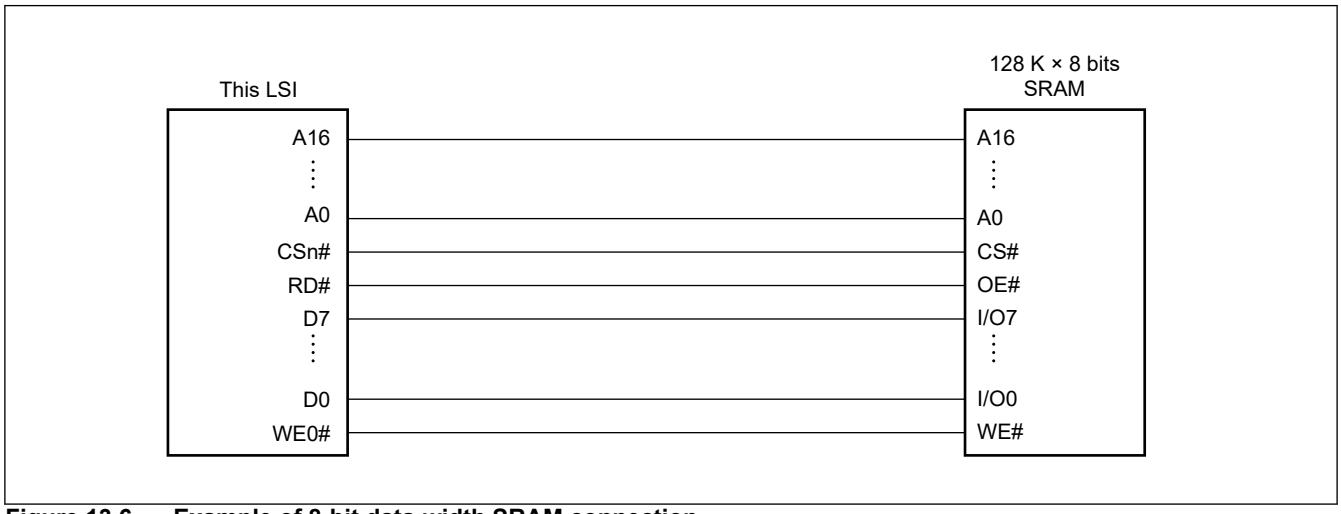
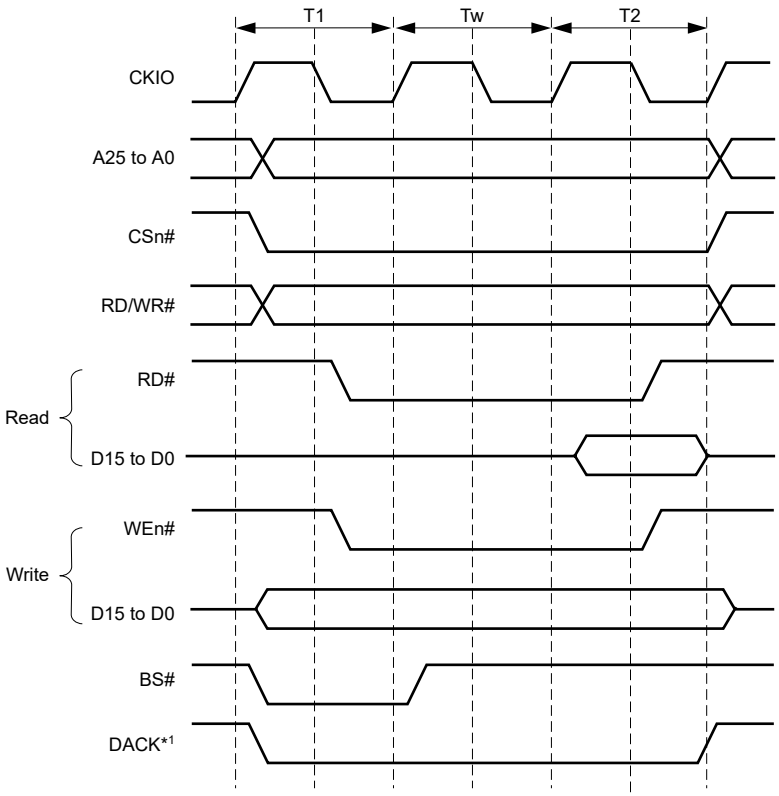


Figure 13.6 Example of 8-bit data-width SRAM connection

13.4.3 Access Wait Control

Wait insertion on an SRAM interface access can be controlled by the settings of bits WR3 to WR0 in CSnWCR. It is possible for area 5 to insert waits independently in read access and in write access. Areas 0, 2, and 3 have common access wait for read cycle and write cycle. The specified number of Tw states are inserted as waits in an SRAM interface access shown in Figure 13.7.

Example: Insertion of 1 cycle of waiting by a software wait
(WR[3:0] = 0001b, WW[2:0] = 000b)

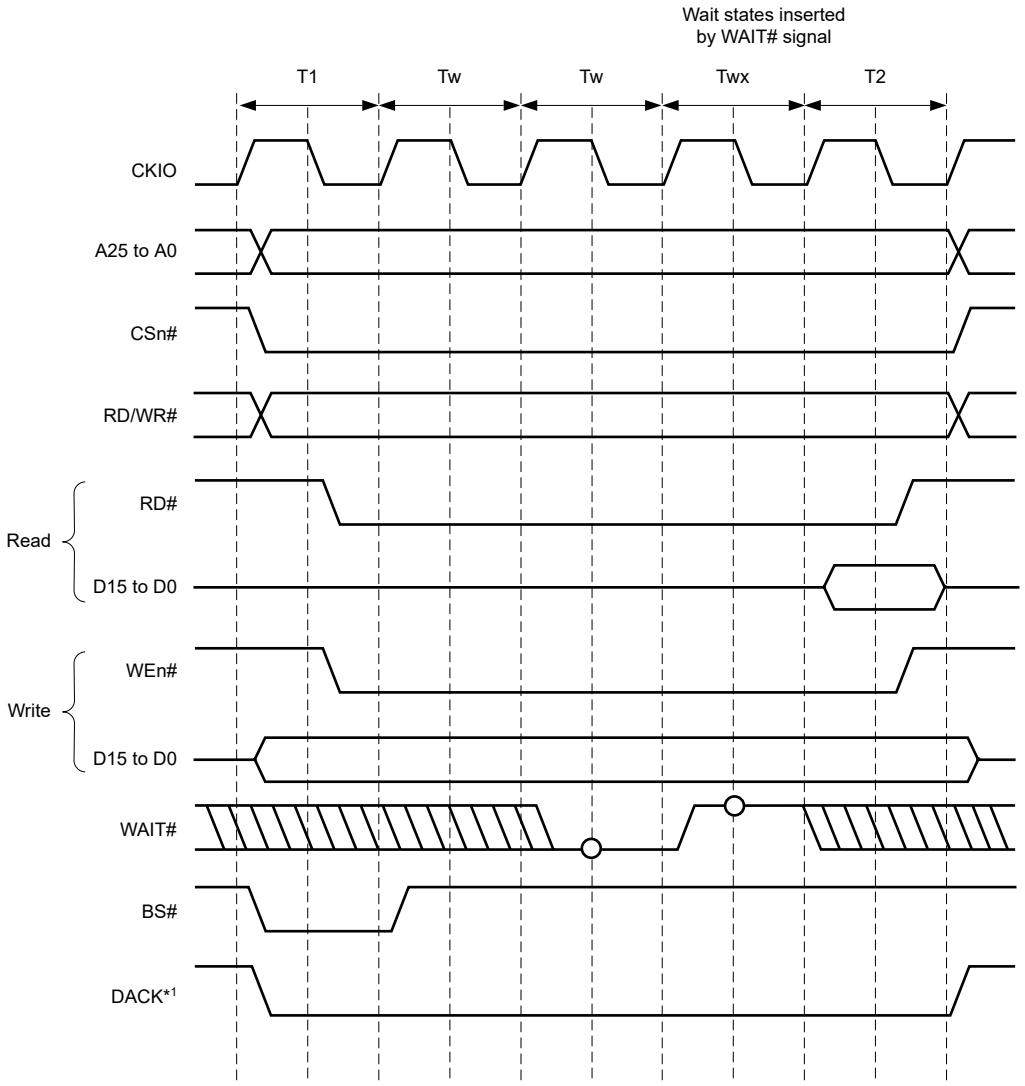


Note 1. The waveform for DACKn is when active-low is specified.

Figure 13.7 Wait timing for SRAM interface access (software wait only)

When the WM bit in the CSnWCR register is cleared to 0, the signal on the external WAIT# pin is also sampled. WAIT# pin sampling is shown in Figure 13.8. Two waits are specified as software waits. The WAIT# signal is sampled on the falling edge of the CKIO signal at the transition from the T1 or Tw cycle to the T2 cycle.

Example: Insertion of 2 cycles of waiting by a software wait and 1 cycle of waiting by the WAIT# signal (WR[3:0] = 0010b, WW[2:0] = 000b, WM = 0)



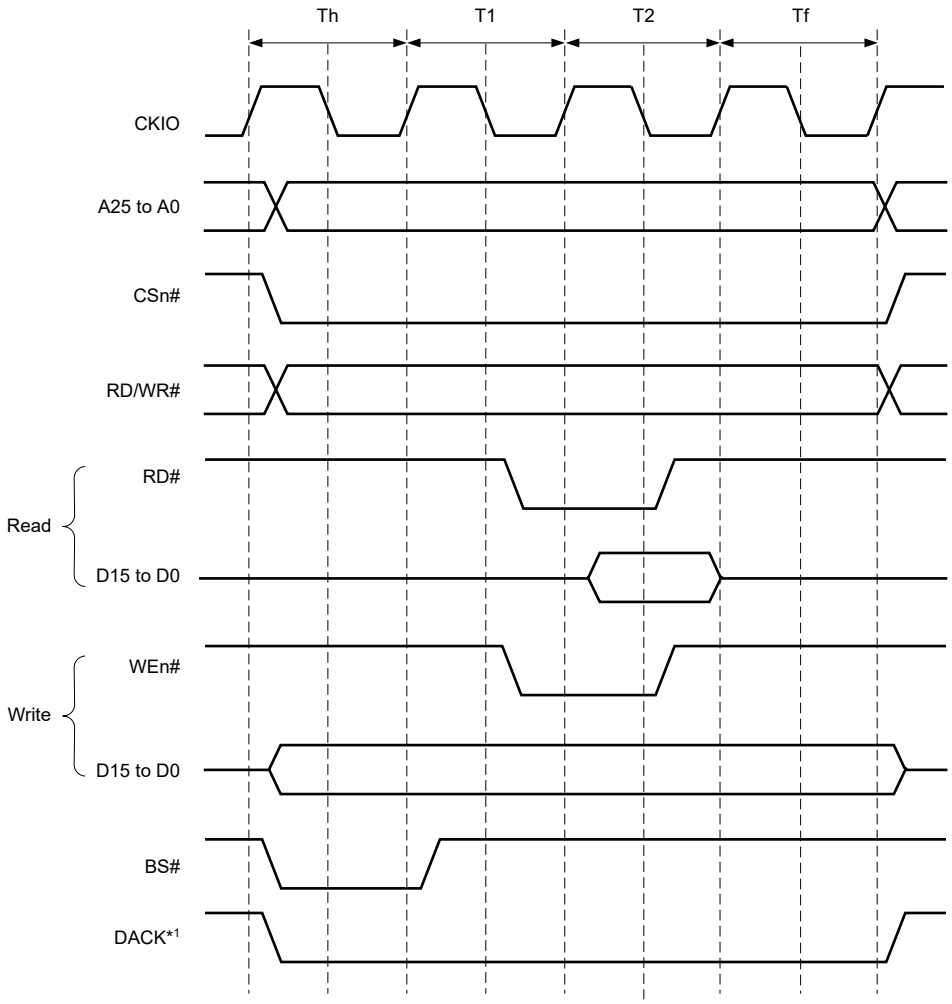
Note 1. The waveform for DACKn is when active-low is specified.

Figure 13.8 Wait cycle timing for SRAM interface access (wait cycle insertion using WAIT# signal)

13.4.4 CSn# Assert Period Expansion

The number of states from CSn# active to RD#, WEn# active can be specified by setting bits SW1 and SW0 in CSnWCR. The number of states from the inactive state of the RD# and WEn# signals to the inactive state of the CSn# signal can be specified by setting bits HW1 and HW0. Therefore, a flexible interface to an external device can be obtained. Figure 13.9 shows an example. A Th state and a Tf state are added before and after an ordinary cycle, respectively. In these states, the RD# and WEn# signals are not activated, while other signals are activated. The data output is prolonged to the Tf cycle, and this prolongation is useful for devices with slow writing operations.

Example: SW[1:0] = 01b (delay of 1.5 cycles) and HW[1:0] = 01b (delay of 1.5 cycles)



Note 1. The waveform for DACKn is when active-low is specified.

Figure 13.9 Active period expansion for the CSn signal

13.4.5 MPX-I/O Interface

Access timing for the MPX space is shown below. In the MPX space, CS5#, AH#, RD#, and WEn# signals control the accessing. The basic access for the MPX space consists of 2 cycles of address output followed by an access to an SRAM interface. The bus width for the address output cycle or the data input/output cycle is fixed to 8 bits or 16 bits.

Alternatively, it can be 8 bits or 16 bits depending on the address to be accessed.

Output of the addresses D15 to D0 or D7 to D0 is performed from state Ta2 to state Ta3. Because state Ta1 has a high-impedance state, collisions of addresses and data can be avoided without inserting idle cycles, even in continuous access cycles. Address output is increased to 3 cycles by setting the MPXW bit in the CS5WCR register to 1.

The RD/WR# signal is output at the same time as the CS5# signal; it is high in the read cycle and low in the write cycle.

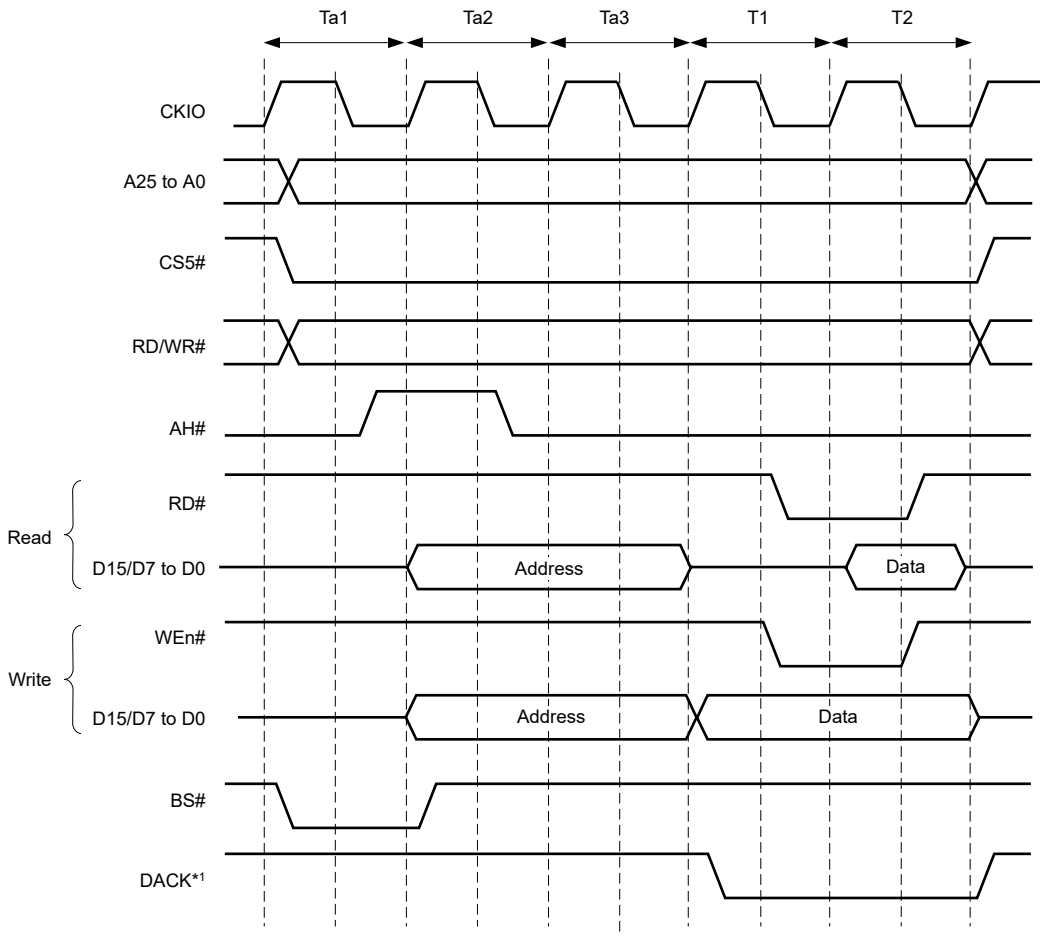
The data cycle is the same as that in an SRAM interface access.

Delay cycles specified in the SW[1:0] bits of the CS5WCR register are inserted between the Ta3 state and the T1 state.

Delay cycles specified in the HW[1:0] bits are added after the T2 state.

Timing charts are shown in Figure 13.10 to Figure 13.13.

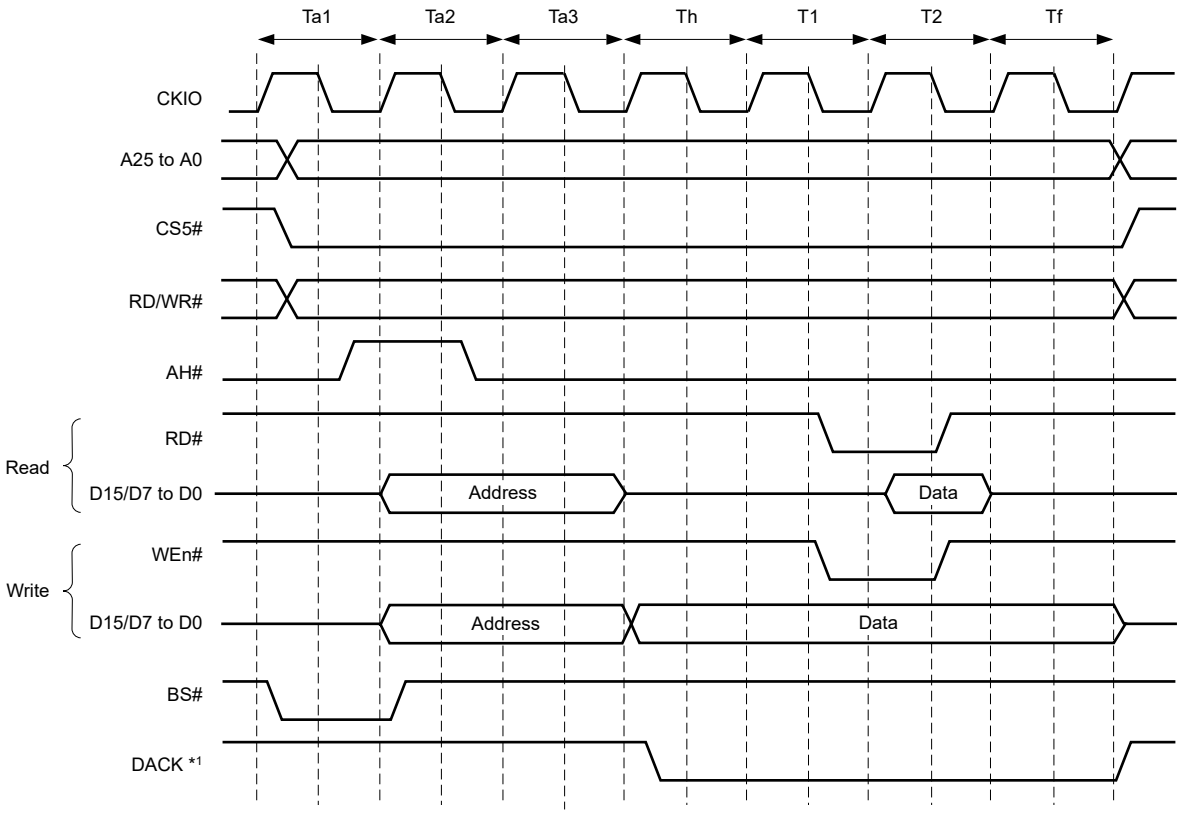
Example: MPXW = 0 (no insertion of wait cycles)



Note 1. The waveform for DACKn is when active-low is specified.

Figure 13.10 Access timing for MPX space (1) (address cycle no wait, data cycle no wait)

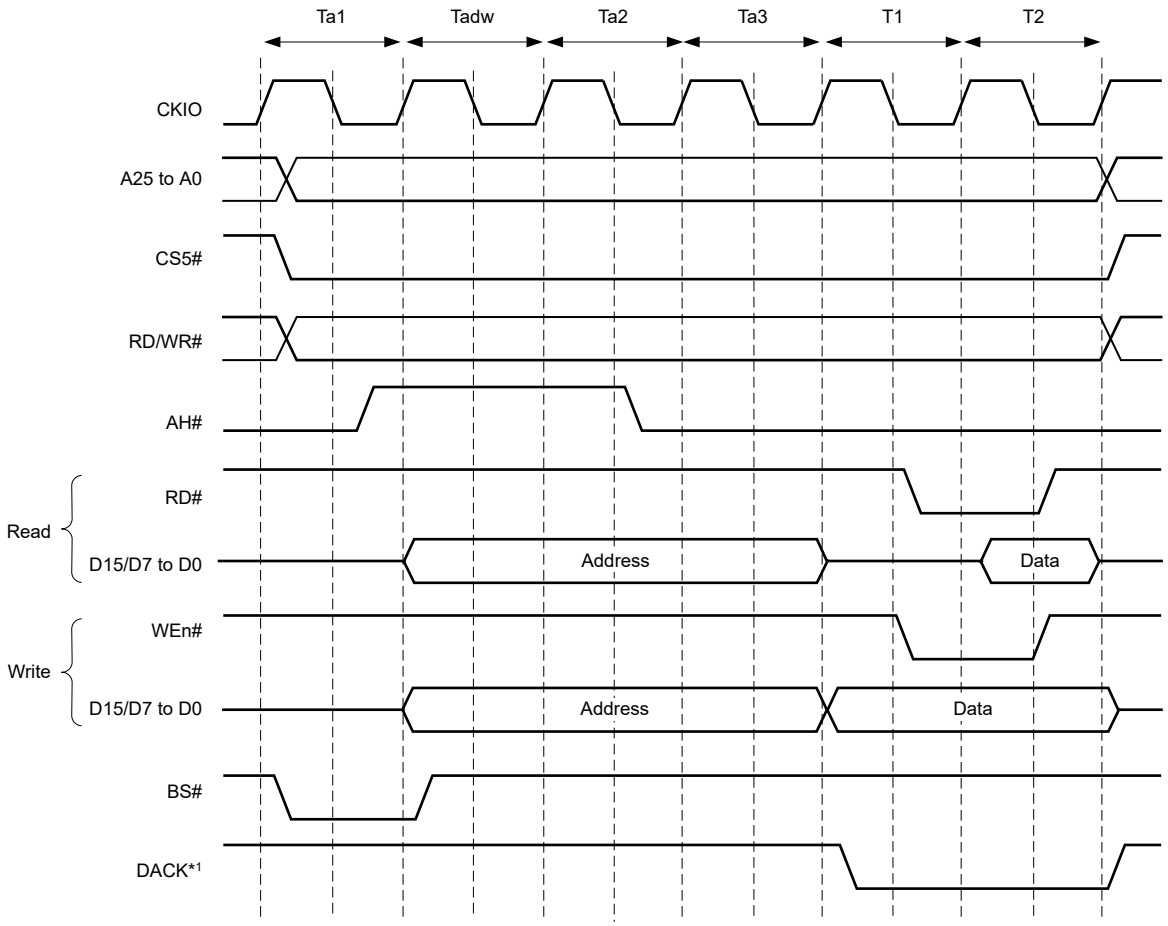
Example: MPXW = 0 (no insertion of wait cycles), SW[1:0] = 01b (delay of 1.5 cycles), HW[1:0] = 01b (delay of 1.5 cycles)



Note 1. The waveform for DACKn is when active-low is specified.

Figure 13.11 Access timing for MPX space (2) (address cycle no wait, 1.5 aassertion delay states, data cycle no wait, 1.5 delay states)

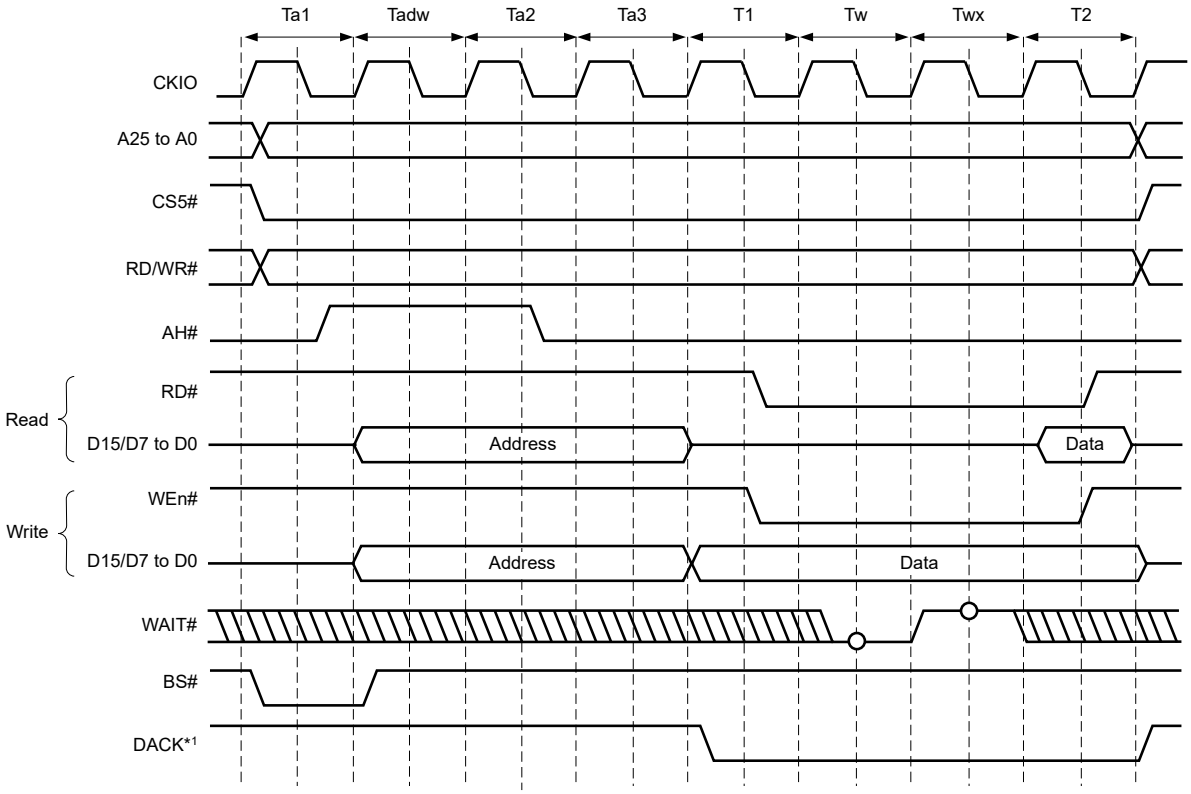
Example: MPXW = 1 (1 cycle of waiting)



Note 1. The waveform for DACKn is when active-low is specified.

Figure 13.12 Access timing for MPX space (3) (address cycle wait 1, data cycle no wait)

Example: MPXW = 1 (1 cycle of waiting), WR[3:0] = 0001b, WW[2:0] = 000b, WM = 0



Note 1. The waveform for DACKn is when active-low is specified.

Figure 13.13 Access timing for MPX space (4) (address cycle access wait 1, data cycle wait 1, external wait 1)

13.4.6 SDRAM Interface

(1) SDRAM Direct Connection

The SDRAM that can be connected to this LSI is a product that has 11/12/13 bits of row address, 8/9/10 bits of column address, 4 or less banks, and uses the A10 pin for setting precharge mode in read and write command cycles.

The control signals for direct connection of SDRAM are RAS#, CAS#, RD/WR#, DQMLU, DQMLL, CKE, CS2#, and CS3#. All the signals other than CS2# and CS3# are common to all areas, and signals other than CKE are valid only when CS2# or CS3# is asserted. SDRAM can be connected to up to 2 spaces. The data bus width of the area that is connected to SDRAM is 16 bits .

Commands for SDRAM can be specified by RAS#, CAS#, RD/WR#, and specific address signals. These commands supports:

- NOP
- Auto-refresh (REF)
- Self-refresh (SELF)
- All banks pre-charge (PALL)
- Specified bank pre-charge (PRE)
- Bank active (ACTV)
- Read (READ)
- Read with pre-charge (READA)

- Write (WRIT)
- Write with pre-charge (WRITA)
- Set mode register (MRS, EMRS)

The byte to be accessed is specified by DQMLU and DQMLL. Reading or writing is performed for a byte whose corresponding DQMxx is low. For details on the relationship between DQMxx (xx = LU, LL) and the byte to be accessed, see section 13.4.1. Access Size and Data Alignment.

Figure 13.14 shows example of the connection of the SDRAM with this LSI.

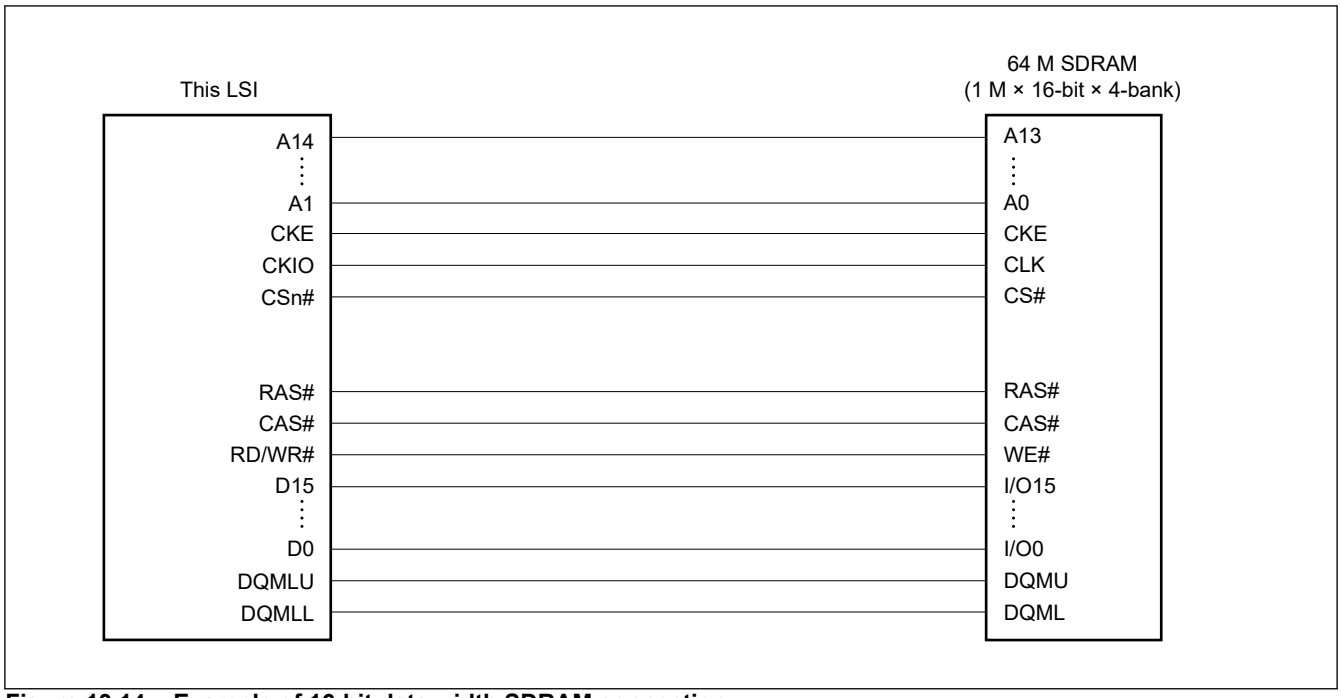


Figure 13.14 Example of 16-bit data width SDRAM connection

(2) Address Multiplexing

An address multiplexing is specified so that SDRAM can be connected without external multiplexing circuitry according to the setting of bits BSZ[1:0] in CSnBCR and bits A2ROW[1:0], A2COL[1:0], A3ROW[1:0], and A3COL[1:0] in SDCR. Table 13.7 to Table 13.12 show the relationship between the settings of bits BSZ[1:0], A2ROW[1:0], A2COL[1:0], A3ROW[1:0], and A3COL[1:0] and the bits output at the address pins. Do not specify those bits in the manner other than this table, otherwise the operation of this LSI is not guaranteed. A25 to A18 are not multiplexed and the original values of address are always output at these pins.

When the data bus width is 16 bits (BSZ1 and BSZ0 = 10b), A0 of SDRAM specifies a 16-bit address. Therefore, connect this A0 pin of SDRAM to the A1 pin of this LSI; the A1 pin of SDRAM to the A2 pin of the LSI, and so on.

Table 13.7 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and address multiplex output (1-1)

Setting				
BSZ[1:0]	**A2/3ROW[1:0]**	**A2/3COL[1:0]**		
10 (16 bits)	**00 (11 bits)**	**00 (8 bits)**		
Output pin of this LSI	**Row address output cycle**	**Column address output cycle**	**SDRAM pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22	A14		
A13	A21	A21		
A12	A20*2	A20*2	A11	Specifies bank
A11	A19	L/H*1	A10/AP	Specifies address/ precharge
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
16-Mbit product (512 Kwords × 16 bits × 2 banks, column 8 bits product): 1				

Note 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
Note 2. Bank address specification

Table 13.8 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and address multiplex output (1-2) (1 of 2)

Setting				
BSZ[1:0]	**A2/3ROW[1:0]**	**A2/3COL[1:0]**		
10 (16 bits)	**01 (12 bits)**	**00 (8 bits)**		
Output pin of this LSI	**Row address output cycle**	**Column address output cycle**	**SDRAM pin**	**Function**
A17	A25	A17		Unused
A16	A24	A16		
A15	A23	A15		
A14	A22*2	A22*2	A13 (BA1)	Specifies bank
A13	A21*2	A21*2	A12 (BA0)	
A12	A20	A12	A11	Address
A11	A19	L/H*1	A10/AP	Specifies address/ precharge

Table 13.8 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and address multiplex output (1-2) (2 of 2)

Setting				
BSZ[1:0]	**A2/3ROW[1:0]**	**A2/3COL[1:0]**		
10 (16 bits)	**01 (12 bits)**	**00 (8 bits)**		
Output pin of this LSI	**Row address output cycle**	**Column address output cycle**	**SDRAM pin**	**Function**
A10	A18	A10	A9	Address
A9	A17	A9	A8	
A8	A16	A8	A7	
A7	A15	A7	A6	
A6	A14	A6	A5	
A5	A13	A5	A4	
A4	A12	A4	A3	
A3	A11	A3	A2	
A2	A10	A2	A1	
A1	A9	A1	A0	
A0	A8	A0		Unused
Example of connected memory				
64-Mbit product (1 Mwords × 16 bits × 4 banks, column 8 bits product): 1				

Note 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
Note 2. Bank address specification

Table 13.9 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and address multiplex output (2-1) (1 of 2)

Setting				
BSZ[1:0]	**A2/3ROW[1:0]**	**A2/3COL[1:0]**		
10 (16 bits)	**01 (12 bits)**	**01 (9 bits)**		
Output pin of this LSI	**Row address output cycle**	**Column address output cycle**	**SDRAM pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24	A15		
A14	A23*2	A23*2	A13 (BA1)	Specifies bank
A13	A22*2	A22*2	A12 (BA0)	
A12	A21	A12	A11	Address
A11	A20	L/H*1	A10/AP	Specifies address/ precharge

Table 13.9 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and address multiplex output (2-1) (2 of 2)

Setting				
BSZ[1:0]	**A2/3ROW[1:0]**	**A2/3COL[1:0]**		
10 (16 bits)	**01 (12 bits)**	**01 (9 bits)**		
Output pin of this LSI	**Row address output cycle**	**Column address output cycle**	**SDRAM pin**	**Function**
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused
Example of connected memory				
128-Mbit product (2 Mwords × 16 bits × 4 banks, column 9 bits product): 1				

Note 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
Note 2. Bank address specification

Table 13.10 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and address multiplex output (2-2) (1 of 2)

Setting				
BSZ[1:0]	**A2/3ROW[1:0]**	**A2/3COL[1:0]**		
10 (16 bits)	**01 (12 bits)**	**10 (10 bits)**		
Output pin of this LSI	**Row address output cycle**	**Column address output cycle**	**SDRAM pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25	A15		
A14	A24*2	A24*2	A13 (BA1)	Specifies bank
A13	A23*2	A23*2	A12 (BA0)	
A12	A22	A12	A11	Address
A11	A21	L/H*1	A10/AP	Specifies address/ precharge

Table 13.10 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and address multiplex output (2-2) (2 of 2)

Setting				
BSZ[1:0]	**A2/3ROW[1:0]**	**A2/3COL[1:0]**		
10 (16 bits)	**01 (12 bits)**	**10 (10 bits)**		
Output pin of this LSI	**Row address output cycle**	**Column address output cycle**	**SDRAM pin**	**Function**
A10	A20	A10	A9	Address
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unused
Example of connected memory				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 10 bits product): 1				

Note 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
Note 2. Bank address specification

Table 13.11 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and address multiplex output (3-1) (1 of 2)

Setting				
BSZ[1:0]	**A2/3ROW[1:0]**	**A2/3COL[1:0]**		
10 (16 bits)	**10 (13 bits)**	**01 (9 bits)**		
Output pin of this LSI	**Row address output cycle**	**Column address output cycle**	**SDRAM pin**	**Function**
A17	A26	A17		Unused
A16	A25	A16		
A15	A24*2	A24*2	A14 (BA1)	Specifies bank
A14	A23*2	A23*2	A13 (BA0)	
A13	A22	A13	A12	Address
A12	A21	A12	A11	
A11	A20	L/H*1	A10/AP	Specifies address/ precharge

Table 13.11 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and address multiplex output (3-1) (2 of 2)

Setting				
BSZ[1:0]	**A2/3ROW[1:0]**	**A2/3COL[1:0]**		
10 (16 bits)	**10 (13 bits)**	**01 (9 bits)**		
Output pin of this LSI	**Row address output cycle**	**Column address output cycle**	**SDRAM pin**	**Function**
A10	A19	A10	A9	Address
A9	A18	A9	A8	
A8	A17	A8	A7	
A7	A16	A7	A6	
A6	A15	A6	A5	
A5	A14	A5	A4	
A4	A13	A4	A3	
A3	A12	A3	A2	
A2	A11	A2	A1	
A1	A10	A1	A0	
A0	A9	A0		Unused
Example of connected memory				
256-Mbit product (4 Mwords × 16 bits × 4 banks, column 9 bits product): 1				

Note 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
Note 2. Bank address specification

Table 13.12 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and address multiplex output (3-2) (1 of 2)

Setting				
BSZ[1:0]	**A2/3ROW[1:0]**	**A2/3COL[1:0]**		
10 (16 bits)	**10 (13 bits)**	**10 (10 bits)**		
Output pin of this LSI	**Row sddress output cycle**	**Column address output cycle**	**SDRAM pin**	**Function**
A17	A27	A17		Unused
A16	A26	A16		
A15	A25*2	A25*2	A14 (BA1)	Specifies bank
A14	A24*2	A24*2	A13 (BA0)	
A13	A23	A13	A12	Address
A12	A22	A12	A11	
A11	A21	L/H*1	A10/AP	Specifies address/ precharge

Table 13.12 Relationship between BSZ[1:0], A2/3ROW[1:0], A2/3COL[1:0], and address multiplex output (3-2) (2 of 2)

Setting				
BSZ[1:0]	A2/3ROW[1:0]	A2/3COL[1:0]		
10 (16 bits)	10 (13 bits)	10 (10 bits)		
Output pin of this LSI	Row sddress output cycle	Column address output cycle	SDRAM pin	Function
A10	A20	A10	A9	Address
A9	A19	A9	A8	
A8	A18	A8	A7	
A7	A17	A7	A6	
A6	A16	A6	A5	
A5	A15	A5	A4	
A4	A14	A4	A3	
A3	A13	A3	A2	
A2	A12	A2	A1	
A1	A11	A1	A0	
A0	A10	A0		Unused
Example of connected memory				
512-Mbit product (8 Mwords × 16 bits × 4 banks, column 10 bits product): 1				

Note 1. L/H is a bit used in the command specification; it is fixed at L or H according to the access mode.
Note 2. Bank address specification

(3) Burst Read

A burst read occurs in the following cases with this LSI.

- Access size in reading is larger than data bus width.
- 16-, 32-, or 64-byte transfer

This LSI always accesses the SDRAM with burst length 1. For example, read access of burst length 1 is performed consecutively 8 times to read 16-byte continuous data from the SDRAM that is connected to a 16-bit data bus. This access is called the burst read with the burst number 8. Table 13.13 shows the relationship between the access size and the number of bursts.

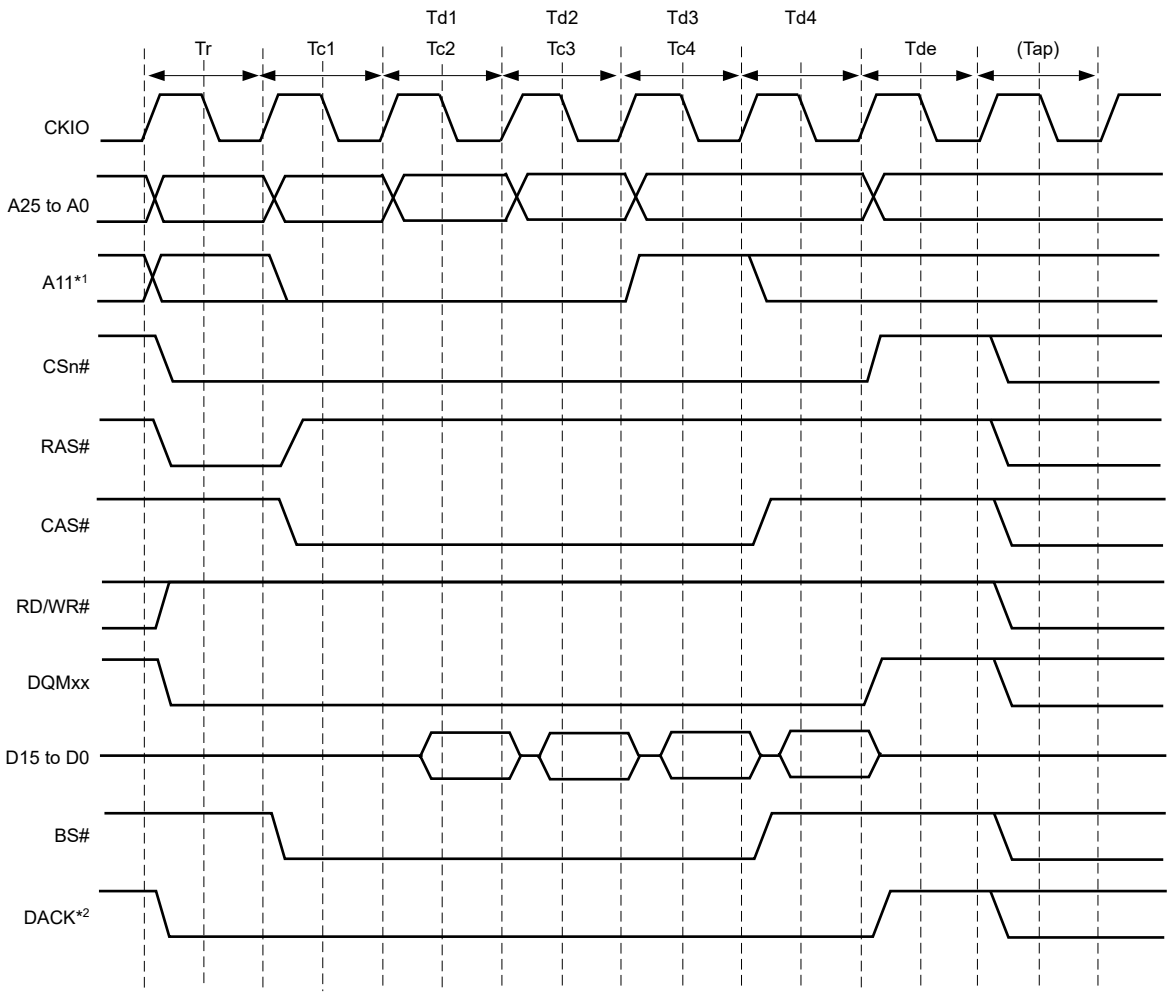
Table 13.13 Relationship between access size and number of bursts

Bus width	Access size	Number of bursts
16 bits	8 bits	1
	16 bits	1
	32 bits	2
	16 bytes	8
	32 bytes	16
	64 bytes	32

Figure 13.15 and Figure 13.16 show timing charts in burst read. In burst read, an ACTV command is output in the Tr cycle, the READ command is issued in the Tc1, Tc2, and Tc3 cycles, the READA command is issued in the Tc4 cycle, and the read data is received at the rising edge of the external clock (CKIO) in the Td1 to Td4 cycles. The Tap cycle is used to wait for the completion of an auto-precharge induced by the READA command in the SDRAM. In the Tap cycle, a new command will not be issued to the same bank. However, access to another CS space or another bank in the same SDRAM space is enabled. The number of Tap cycles is specified by the WTRP[1:0] bits in CS3WCR.

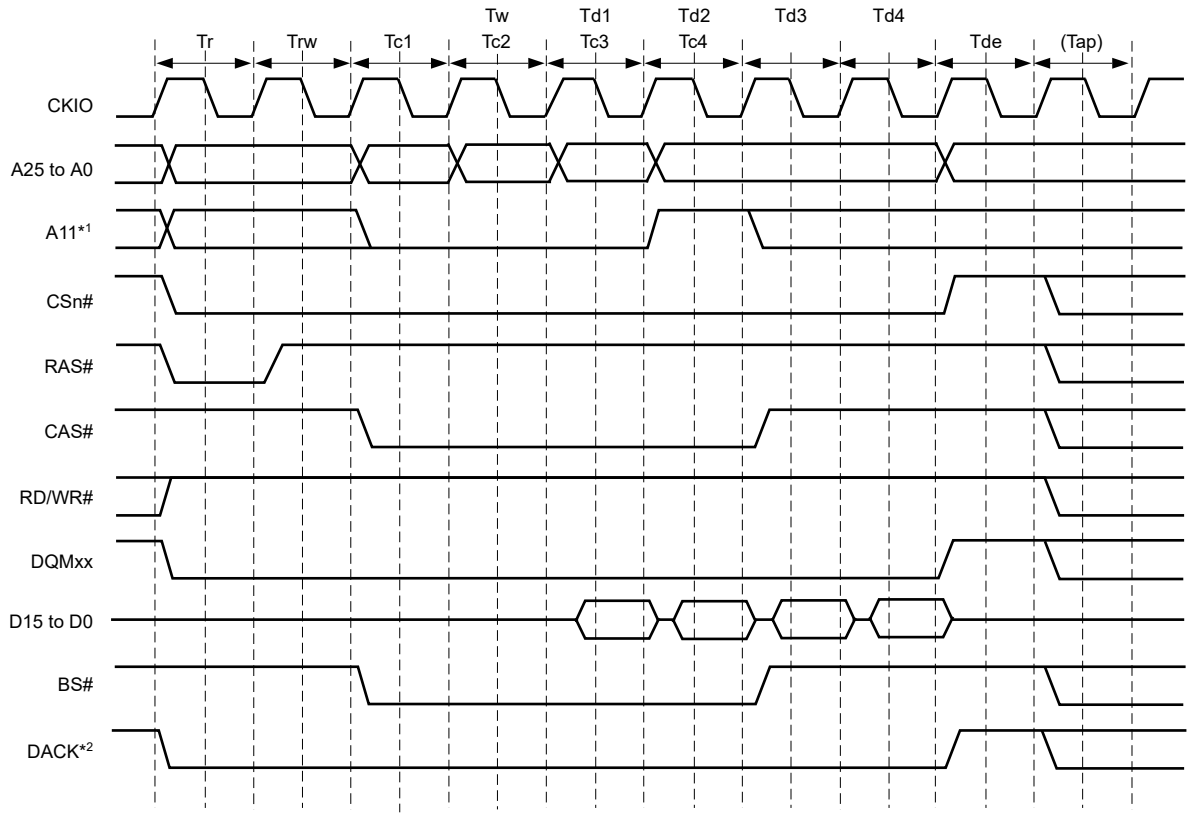
In this LSI, wait cycles can be inserted by specifying each bit in CS3WCR to connect the SDRAM in variable frequencies. Figure 13.16 shows an example in which wait cycles are inserted. The number of states from the Tr state where the ACTV command is output to the Tc1 state where the READ command is output can be specified using the WTRCD[1:0] bits in the CS3WCR register. If the WTRCD[1:0] bits specify one wait or more, a Trw state where the NOP command is issued is inserted between the Tr state and Tc1 state. The number of states from the Tc1 state where the READ command is output to the Td1 state where the read data is latched can be specified for the CS2 and CS3 spaces independently, using the A2CL1 and A2CL0 bits in the CS2WCR register or the A3CL[1:0] bits in the CS3WCR register. The number of states from Tc1 to Td1 corresponds to the SDRAM CAS latency. The CAS latency for the SDRAM is normally defined as up to three. However, the CAS latency in this LSI can be specified as 1 to 4. This CAS latency can be achieved by connecting a latch circuit between this LSI and the SDRAM.

A Tde state is an idle state required to transfer the read data into this LSI and occurs once for every burst read or every single read.



Note 1. Address pin to be connected to pin A10 of SDRAM.
Note 2. The waveform for DACKn is when active-low is specified.

Figure 13.15 Burst read basic timing (CAS latency 1, auto-precharge)



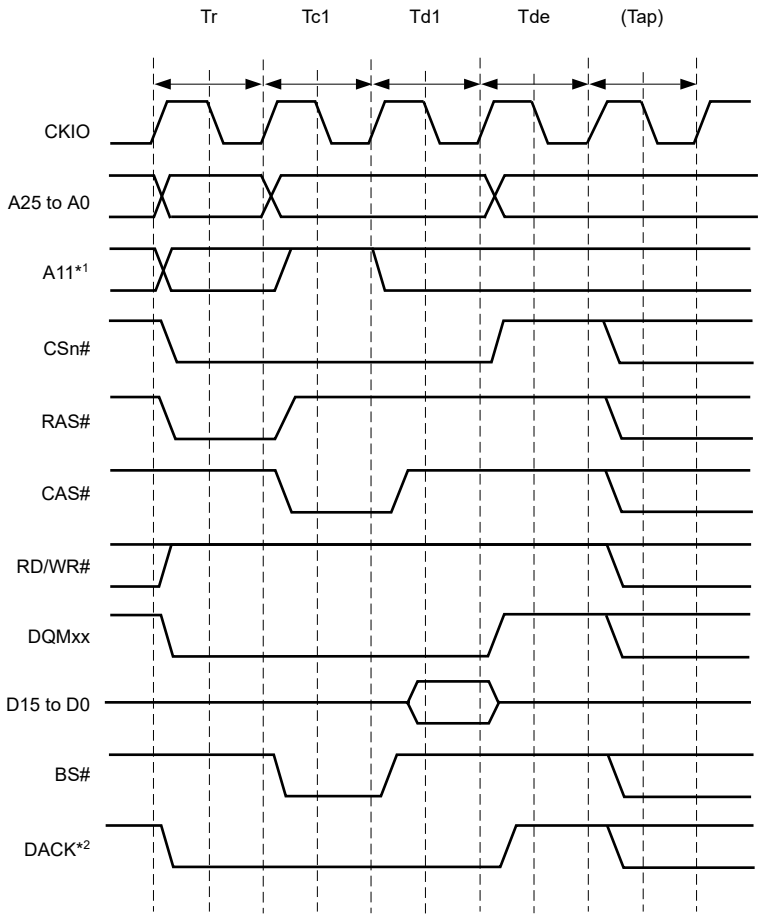
Note 1. Address pin to be connected to pin A10 of SDRAM.
Note 2. The waveform for DACKn is when active-low is specified.

Figure 13.16 Burst read wait specification timing (CAS latency 2, WTRCD[1:0] = 1 state, auto-precharge)

(4) Single Read

A read access ends in one cycle when the data bus width is larger than or equal to the access size. A read access that ends in one cycle is called single read.

Figure 13.17 shows the single read basic timing.



Note 1. Address pin to be connected to pin A10 of SDRAM.
Note 2. The waveform for DACKn is when active-low is specified.

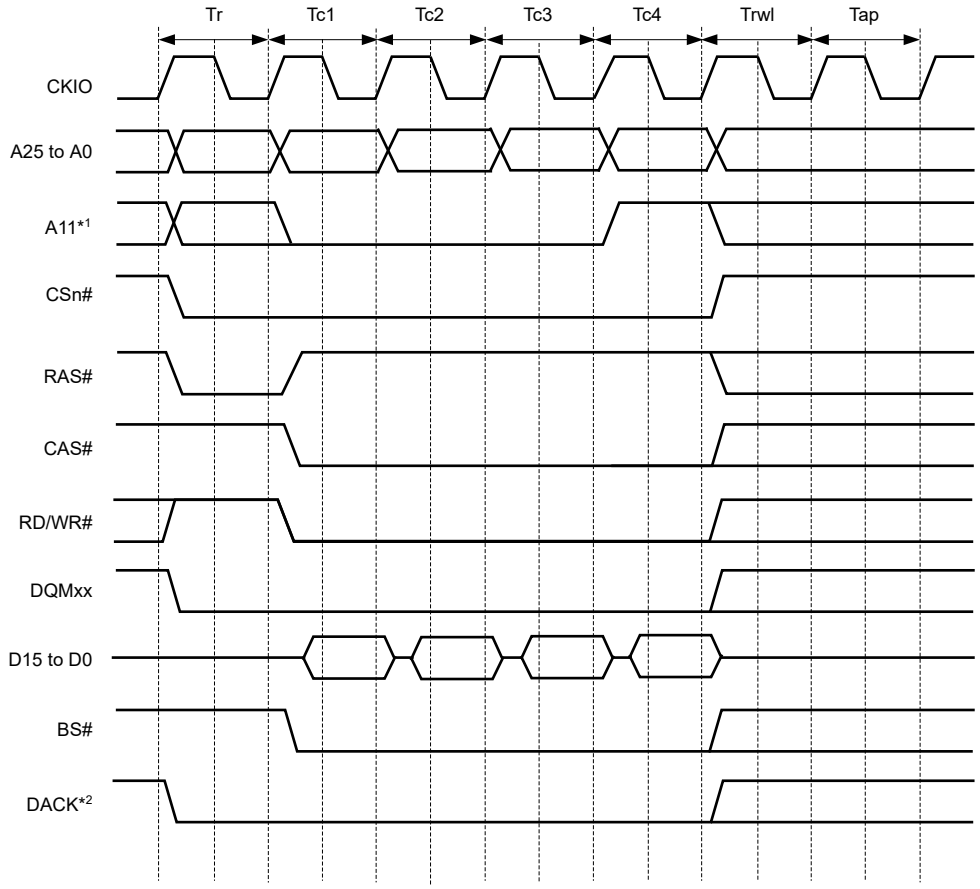
Figure 13.17 Basic timing for single read (CAS latency 1, auto-precharge)

(5) Burst Write

A burst write occurs in the following cases in this LSI.

- Access size in writing is larger than data bus width.
- 16-, 32-, or 64-byte transfer

This LSI always accesses SDRAM with burst length 1. For example, write access of burst length 1 is performed continuously 8 times to write 16-byte continuous data to the SDRAM that is connected to a 16-bit data bus. This access is called burst write with the burst number 8. The relationship between the access size and the number of bursts is shown in Table 13.13. Figure 13.18 shows a timing chart for burst writes. In burst write, an ACTV command is output in the Tr state, the WRIT command is issued in the Tc1, Tc2, and Tc3 states, and the WRITA command is issued to execute an auto-precharge in the Tc4 state. In the write cycle, the write data is output simultaneously with the write command. After the write command with the auto-precharge is output, the Trw1 state that waits for the auto-precharge initiation is followed by the Tap state that waits for completion of the auto-precharge induced by the WRITA command in the SDRAM. Between the Trwl and the Tap states, a new command will not be issued to the same bank. However, access to another CS space or another bank in the same SDRAM space is enabled. The number of Trw1 states is specified by the TRWL[1:0] bits in the CS3WCR register. The number of Tap states is specified by the WTRP[1:0] bits in the CS3WCR register.

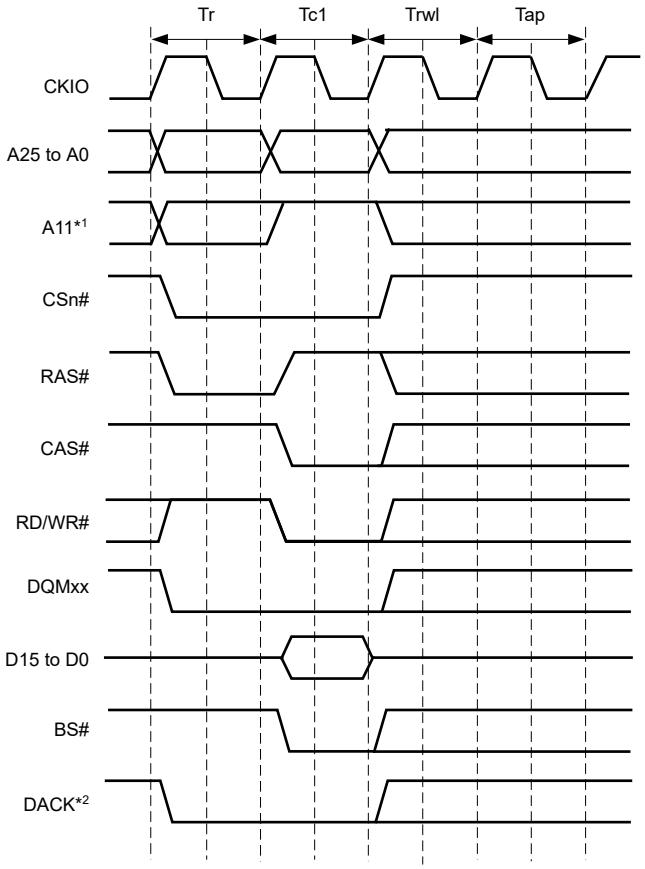


Note 1. Address pin to be connected to pin A10 of SDRAM.
Note 2. The waveform for DACKn is when active-low is specified.

Figure 13.18 Basic timing for burst write (auto-precharge)

(6) Single Write

A write access ends in one cycle when the data bus width is larger than or equal to the access size. A write access that ends in one cycle is called single write. Figure 13.19 shows the basic timing of single-write operation.



Note 1. Address pin to be connected to pin A10 of SDRAM.
Note 2. The waveform for DACKn is when active-low is specified.

Figure 13.19 Single write basic timing (auto-precharge)

(7) Bank Active

The SDRAM bank function can be used to support high-speed access to the same row address. When the BACTV bit in SDCR is 1, access is performed using commands without auto-precharge (READ or WRIT). This function is called bank-active function. This function is valid only for area 3. When area 3 is set to bank-active mode, area 2 should be set to normal space or SRAM with byte selection. When areas 2 and 3 are both set to SDRAM, auto precharge mode must be set.

When the bank-active function is used, precharging is not performed when the access ends. When accessing the same row address in the same bank, it is possible to issue the READ or WRIT command immediately, without issuing an ACTV command. As SDRAM is internally divided into several banks, it is possible to activate one row address in each bank. If the next access is to a different row address, a PRE command is first issued to precharge the relevant bank, then when precharging is completed, the access is performed by issuing an ACTV command followed by a READ or WRIT command. If this is followed by an access to a different row address, the access time will be longer because of the precharging performed after the access request is received. The number of states between issuance of the PRE command and the ACTV command is determined by the WTRP[1:0] bits in CS3WCR.

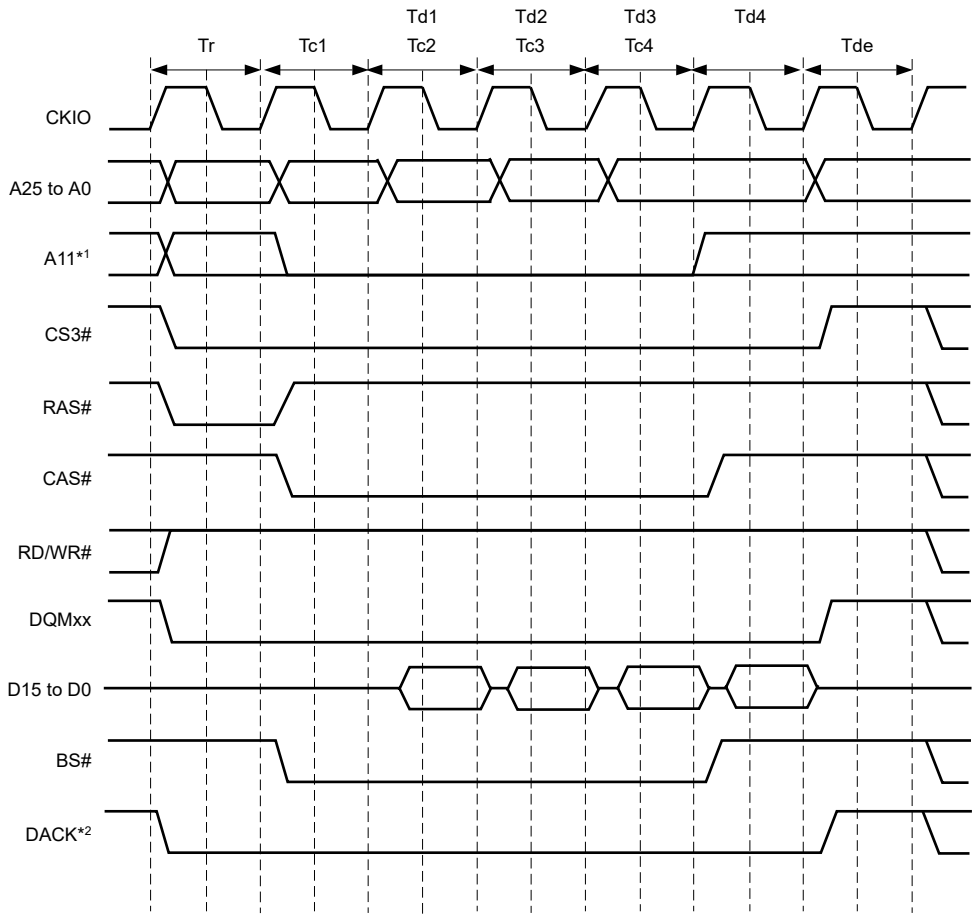
In a write, when an auto-precharge is performed, a command cannot be issued to the same bank for a period of Trwl + Tap states after issuance of the WRITA command. When bank active mode is used, READ or WRIT commands can be issued successively if the row address is the same. The number of cycles can thus be reduced by Trwl + Tap states for each write.

There is a limit on tRAS, the time for placing each bank in the active state. If there is no guarantee that there will not be a cache hit and another row address will be accessed within the period in which this value is maintained by program execution, it is necessary to set auto-refresh and set the refresh cycle to no more than the maximum value of tRAS. A burst read cycle without auto-precharge is shown in Figure 13.20, a burst read cycle for the same row address in Figure 13.21, and a burst read cycle for different row addresses in Figure 13.22. Similarly, a single write cycle without auto-precharge is

shown in Figure 13.23, a single write cycle for the same row address in Figure 13.24, and a single write cycle for different row addresses in Figure 13.25.

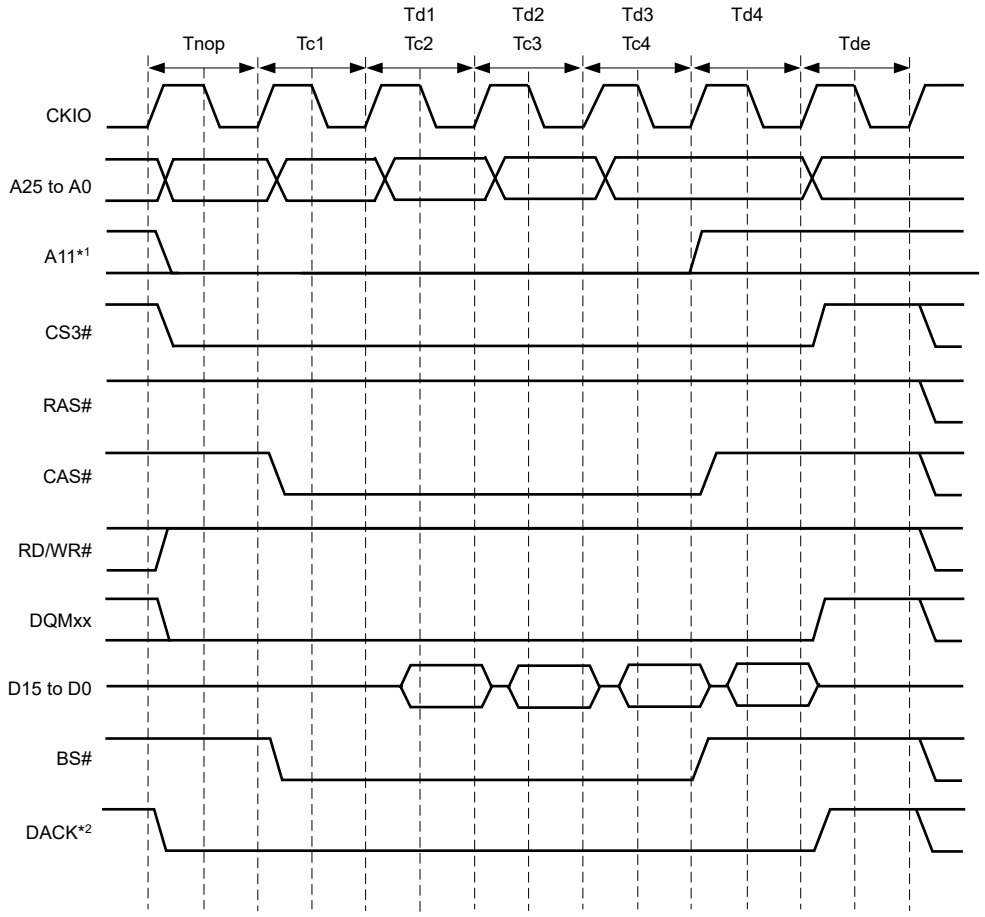
In Figure 13.21, a Tnop state in which no operation is performed is inserted before the Tc state that issues the READ command. The Tnop state is inserted to maintain latency 2 for the DQMxx signal that specifies the read byte in the data read from the SDRAM. If the CAS latency is specified as two or more, the Tnop state is not inserted because latency 2 can be maintained even if the DQMxx signal is activated after the Tc state.

When bank active mode is set, if only access cycles to the respective banks in the area 3 space are considered, as long as access cycles to the same row address continue, the operation starts with the cycle in Figure 13.20 or Figure 13.23, followed by repetition of the cycle in Figure 13.21 or Figure 13.24. An access to a different area during this time has no effect. If there is an access to a different row address in the bank active state, the bus cycle in Figure 13.22 or Figure 13.25 is executed instead of that in Figure 13.21 or Figure 13.24. In bank active mode, too, all banks become inactive after a refresh cycle.



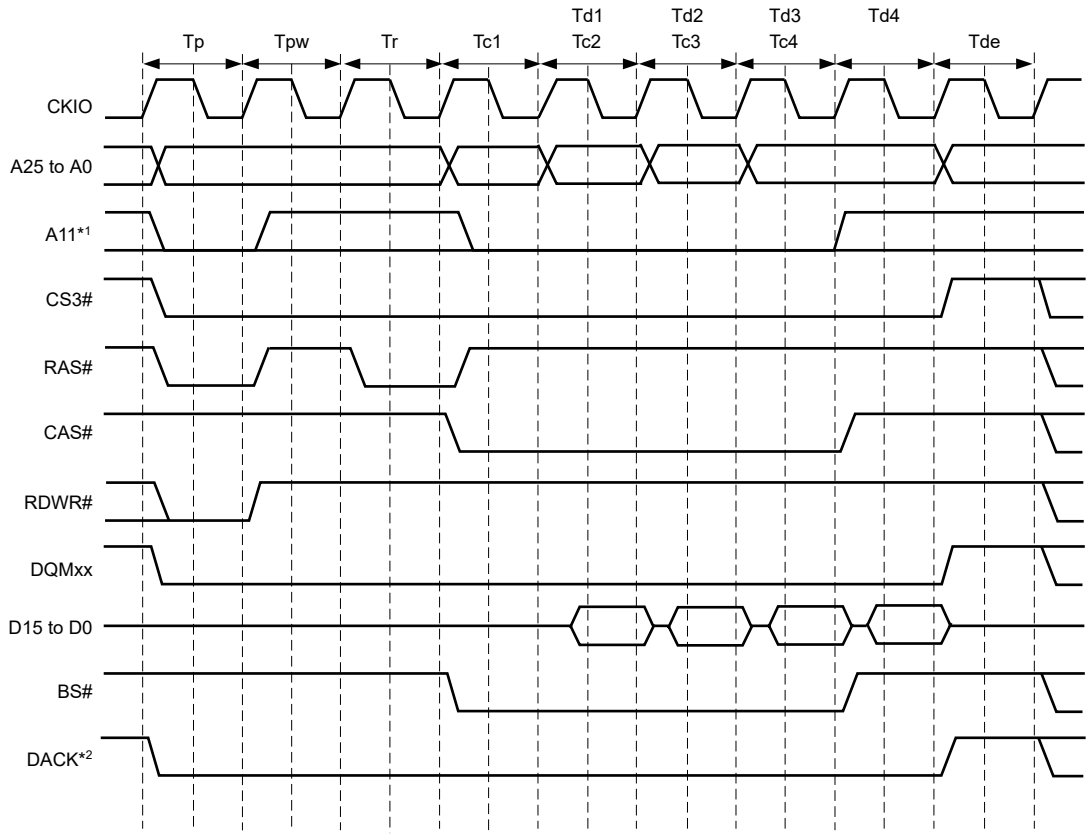
Note 1. Address pin to be connected to pin A10 of SDRAM.
Note 2. The waveform for DACKn is when active-low is specified.

Figure 13.20 Burst read timing (bank active, different bank, CAS latency 1)



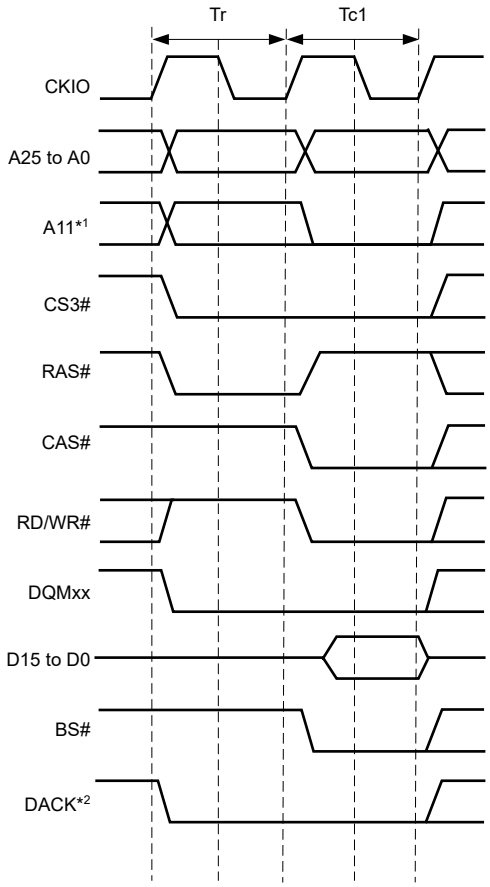
Note 1. Address pin to be connected to pin A10 of SDRAM.
Note 2. The waveform for DACKn is when active-low is specified.

Figure 13.21 Burst read timing (bank active, same row addresses in the same bank, CAS latency 1)



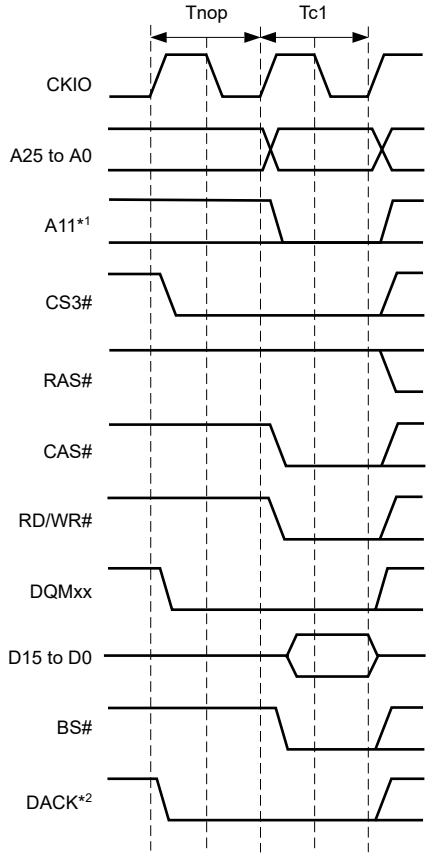
Note 1. Address pin to be connected to pin A10 of SDRAM.
Note 2. The waveform for DACKn is when active-low is specified.

Figure 13.22 Burst read timing (bank active, different row addresses in the same bank, CAS latency 1)



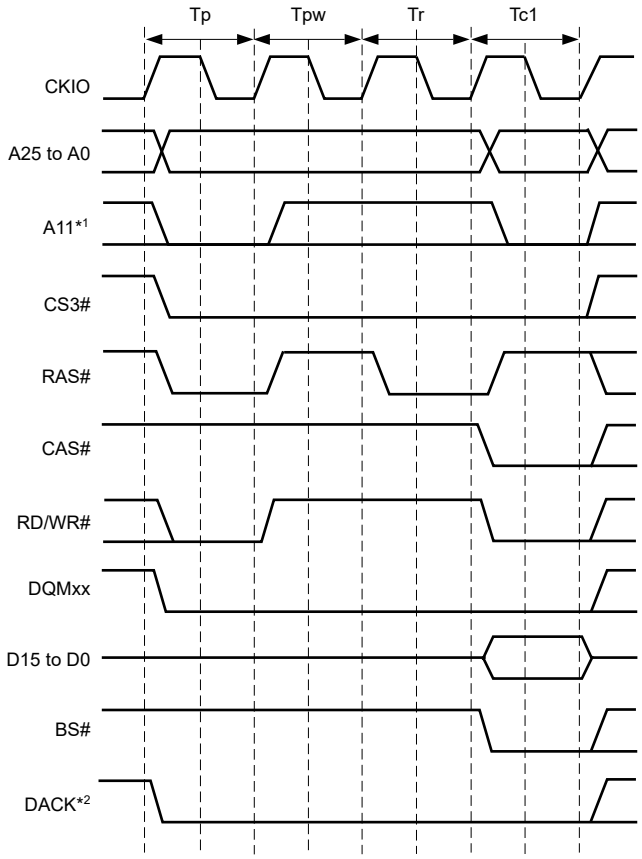
Note 1. Address pin to be connected to pin A10 of SDRAM.
Note 2. The waveform for DACKn is when active-low is specified.

Figure 13.23 Single write timing (bank active, different bank)



Note 1. Address pin to be connected to pin A10 of SDRAM.
Note 2. The waveform for DACKn is when active-low is specified.

Figure 13.24 Single write timing (bank active, same row addresses in the same bank)



Note 1. Address pin to be connected to pin A10 of SDRAM.
Note 2. The waveform for DACKn is when active-low is specified.

Figure 13.25 Single write timing (bank active, different row addresses in the same bank)

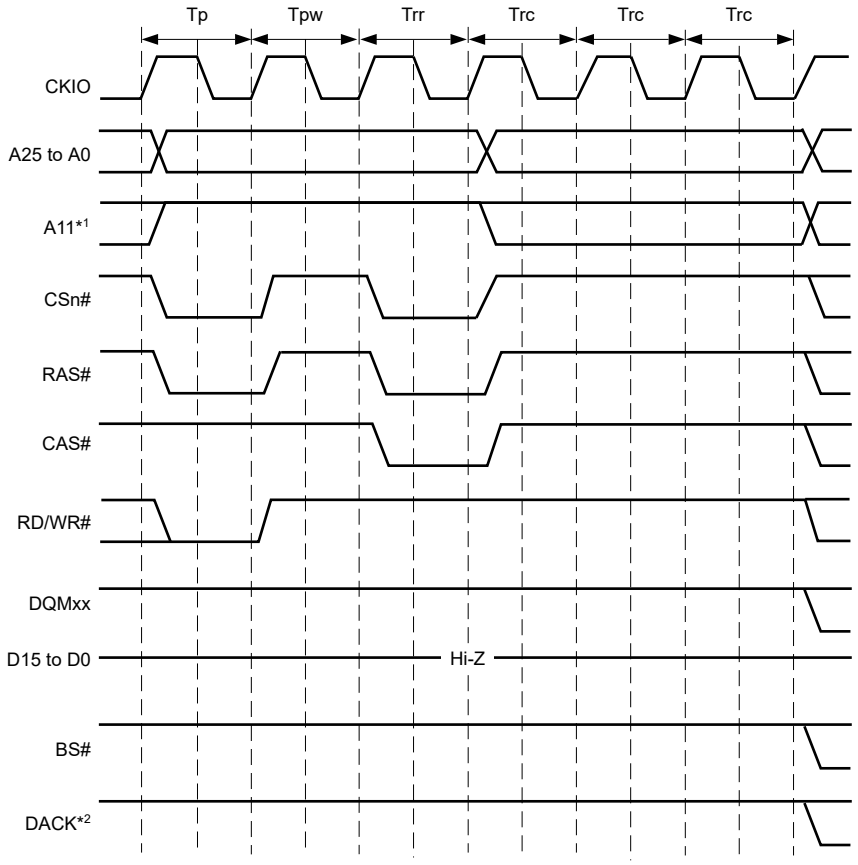
(8) Refreshing

This module has a function for controlling SDRAM refreshing. Auto-refreshing can be performed by clearing the RMODE bit to 0 and setting the RFSH bit to 1 in SDCR. A continuous refreshing can be performed by setting the RRC2 to RRC0 bits in the RTCSR register. If SDRAM is not accessed for a long period, self-refresh mode, in which the power consumption for data retention is low, can be activated by setting both the RMODE bit and the RFSH bit to 1.

(a) Auto-refreshing

Refreshing is performed at intervals determined by the input clock selected by bits CKS2 to CKS0 in RTCSR, and the value set by in RTCOR. The value of bits CKS2 to CKS0 in RTCOR should be set so as to satisfy the refresh interval stipulation for the SDRAM used. First make the settings for RTCOR, RTCNT, and the RMODE and RFSH bits in SDCR, and then make the CKS2 to CKS0 and RRC2 to RRC0 settings. When the clock is selected by bits CKS2 to CKS0, RTCNT starts counting up from the value at that time. The RTCNT value is constantly compared with the RTCOR value, and if the two values are the same, a refresh request is generated and an auto-refresh is performed for the number of times specified by the RRC2 to RRC0. At the same time, RTCNT is cleared to zero and the count-up is restarted.

Figure 13.26 shows the auto-refresh cycle timing. After starting the auto refreshing, PALL command is issued in the Tp state to make all the banks to pre-charged state from active state when some bank is being pre-charged. Then REF command is issued in the Trr state after inserting idle states of which number is specified by the WTRP[1:0] bits in CS3WCR. A new command is not issued for the duration of the number of states specified by the WTRC[1:0] bits in CS3WCR after the Trr state. The WTRC[1:0] bits must be set so as to satisfy the SDRAM refreshing cycle time stipulation (tRC). An idle state is inserted between the Tp state and Trr state when the setting value of the WTRP[1:0] bits in CS3WCR is longer than or equal to 1 state.



Note 1. Address pin to be connected to pin A10 of SDRAM.
Note 2. The waveform for DACKn is when active-low is specified.

Figure 13.26 Auto-refresh timing

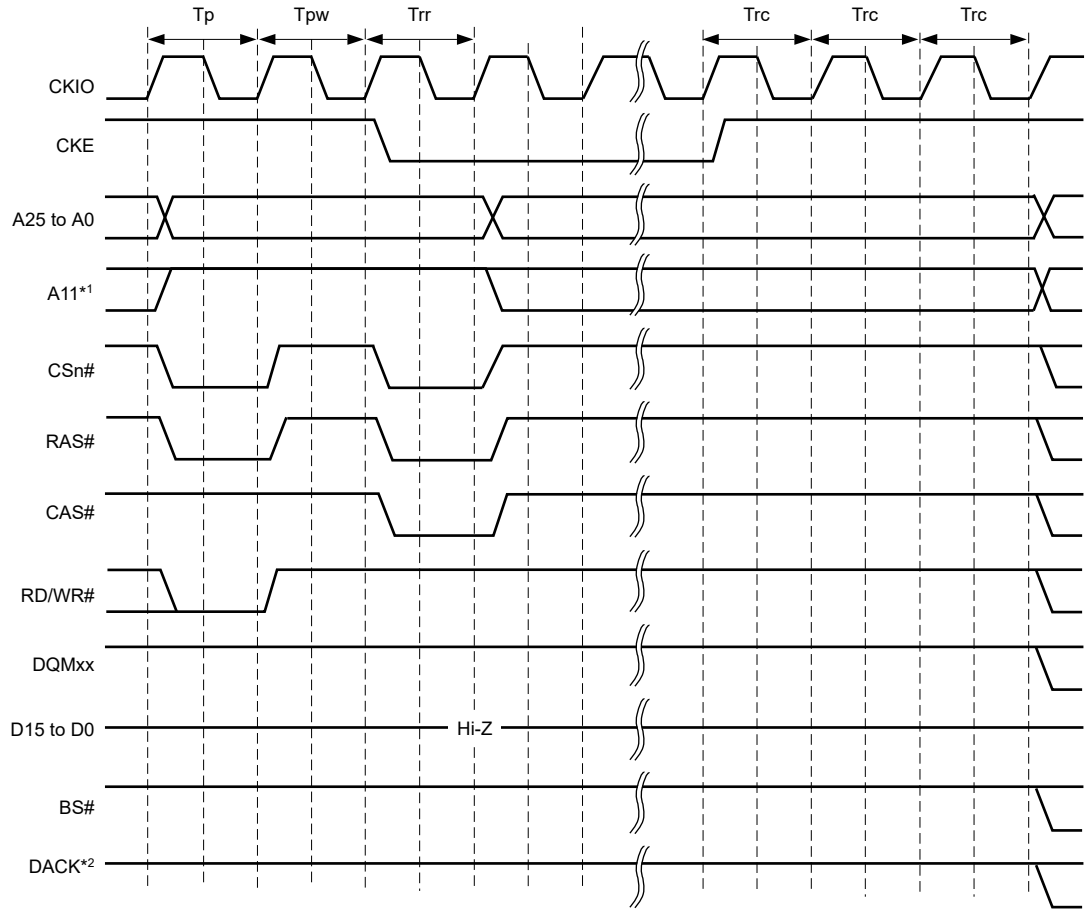
(b) Self-refreshing

Self-refresh mode is a kind of standby mode, in which the refresh timing and refresh addresses are generated within the SDRAM. This LSI enters self-refreshing by setting both the RMODE bit and the RFSH bit in SDCR to 1. After starting the self-refreshing, PALL command is issued in Tp state after the completion of the pre-charging bank. A SELF command is then issued after inserting idle states of which number is specified by the WTRP[1:0] bits in CS3WSR. SDRAM cannot be accessed while in the self-refresh state. Clearing the RMODE bit to 0 clears the self-refresh mode. After self-refresh mode has been cleared, command issuance is disabled for the number of states specified by the WTRC[1:0] bits in CS3WCR.

Self-refresh timing is shown in Figure 13.27. Settings must be made so that self-refresh clearing and data retention are performed correctly, and auto-refreshing is performed at the correct intervals. When self-refreshing is activated from the state in which auto-refreshing is set, auto-refreshing is restarted if the RFSH bit is set to 1 and the RMODE bit is cleared to 0 when self-refresh mode is cleared. If the transition from clearing of self-refresh mode to the start of auto-refreshing takes time, this time should be taken into consideration when setting the initial value of RTCNT. Making the RTCNT value 1 less than the RTCOR value will enable refreshing to be started immediately.

After self-refreshing has been set, the self-refresh state continues even if the chip standby state is entered using this LSI standby function, and is maintained even after recovery from standby mode due to an interrupt.

In case of a power-on reset, the bus state controller's registers are initialized, and therefore the self-refresh state is cleared.



Note 1. Address pin to be connected to pin A10 of SDRAM.
Note 2. The waveform for DACKn is when active-low is specified.

Figure 13.27 Self-refresh timing

(9) Relationship between Refresh Requests and Bus Cycles

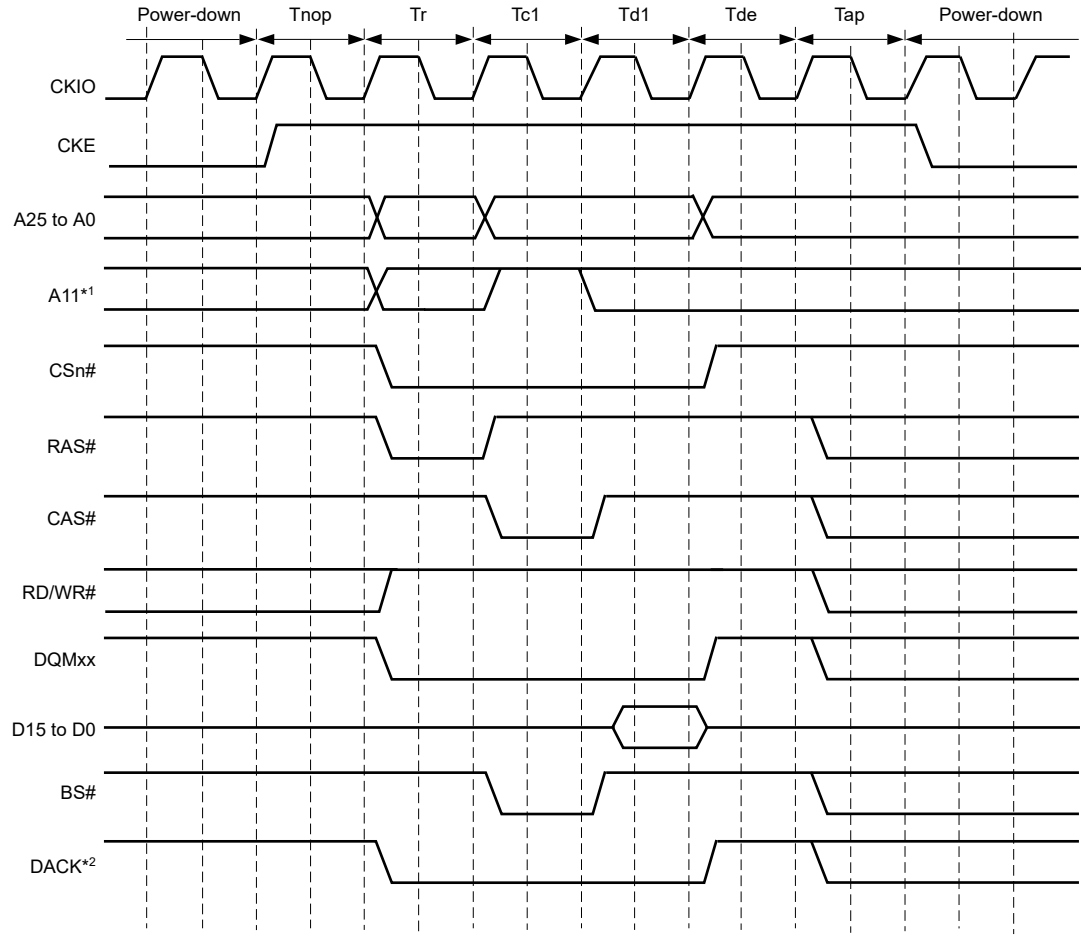
If a refresh request occurs during bus cycle execution, the refresh cycle must wait for the bus cycle to be completed.

If a new refresh request occurs while waiting for the previous refresh request, the previous refresh request is deleted. To refresh correctly, a bus cycle longer than the refresh interval must be prevented from occurring.

(10) Power-Down Mode

If the PDOWN bit in SDCR is set to 1 and the CKE signal is set to the low level after the access to the SDRAM, the SDRAM can enter the power-down mode. This power-down mode can effectively lower the power consumption in the non-access cycle. However, please note that if an access occurs in power-down mode, a state of overhead occurs because a state is needed to activate the CKE in order to cancel the power-down mode.

Figure 13.28 shows the access timing in power-down mode.



Note 1. Address pin to be connected to pin A10 of SDRAM.
Note 2. The waveform for DACKn is when active-low is specified.

Figure 13.28 Power-down mode access timing

(11) Power-On Sequence

In order to use SDRAM, mode setting must first be made for SDRAM after the pose interval specified for the SDRAM to be used after powering on. The pose interval should be obtained by a power-on reset generating circuit or software.

To perform SDRAM initialization correctly, the registers of this module must first be set, followed by a write to the SDRAM mode register. In SDRAM mode register setting, the address signal value at that time is latched by a combination of the CSn#, RAS#, CAS#, and RD/WR# signals. If the value to be set is X, the bus state controller provides for value X to be written to the SDRAM mode register by performing a 16-bit write to address 0x8021_1000 + X for area 2 SDRAM and to address 0x8021_2000 + X for area 3 SDRAM. In this operation, the data is ignored, but the mode write is performed as a byte-size access. To set burst read/single write or burst read/burst write (CAS latency 2 to 3, wrap type = sequential, and burst length 1) supported by this LSI, arbitrary data is written in 16 bits to the access addresses shown in Table 13.14. In this time, 0 is output at the external address pins of A12 or later.

Table 13.14 Access address in SDRAM mode register write

Setting for Area 2
Burst read/single write (burst length 1):

Data bus width	CAS latency	Access address	External address pin
16 bits	2	0x8021_1440	0x0000440
	3	0x8021_1460	0x0000460

Burst read/burst write (burst length 1):

Data bus width	CAS latency	Access address	External address pin
16 bits	2	0x8021_1040	0x0000040
	3	0x8021_1060	0x0000060

Setting for Area 3
Burst read/single write (burst length 1):

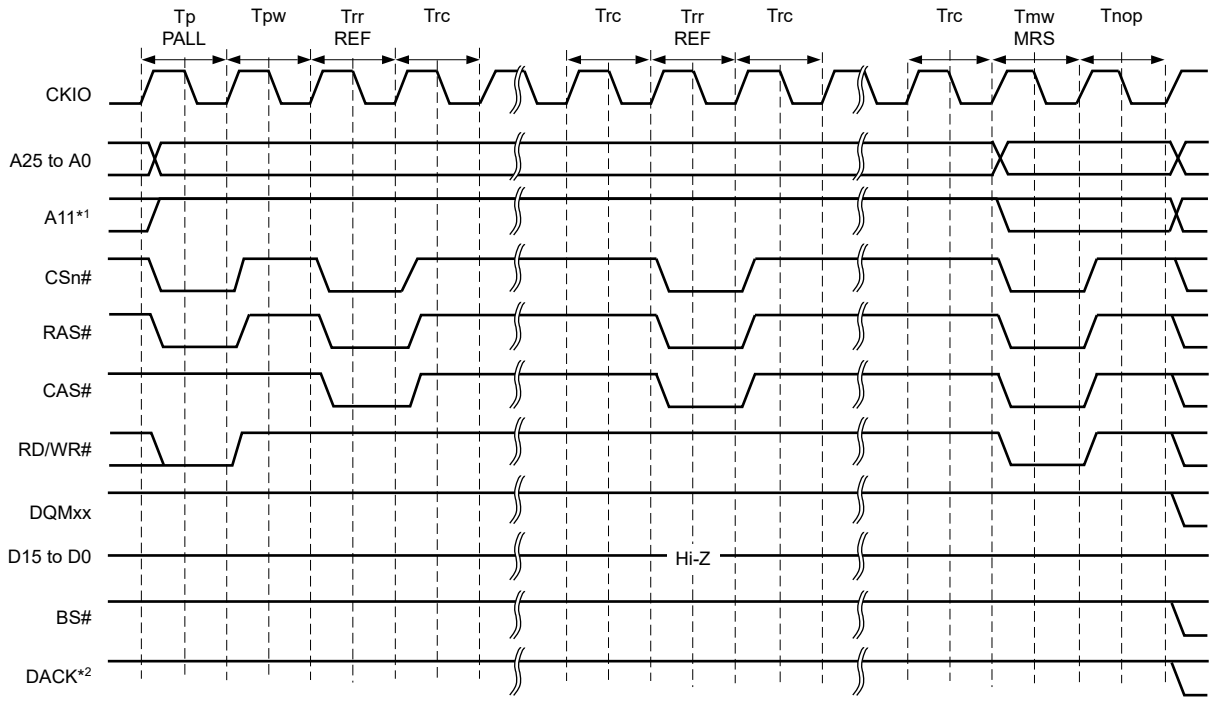
Data bus width	CAS latency	Access address	External address pin
16 bits	2	0x8021_2440	0x0000440
	3	0x8021_2460	0x0000460

Burst read/burst write (burst length 1):

Data bus width	CAS latency	Access address	External address pin
16 bits	2	0x8021_2040	0x0000040
	3	0x8021_2060	0x0000060

Mode register setting timing is shown in Figure 13.29. A PALL command (all bank pre-charge command) is firstly issued. A REF command (auto refresh command) is then issued 8 times. An MRS command (mode register set command) is finally issued. Idle states, of which number is specified by the WTRP[1:0] bits in CS3WCR, are inserted between the PALL and the first REF. Idle states, of which number is specified by the WTRC[1:0] bits in CS3WCR, are inserted between REF and REF, and between the 8th REF and MRS. One or more idle states are inserted between the MRS and a command to be issued next.

It is necessary to keep idle time of certain cycles for SDRAM before issuing PALL command after power-on. Refer to the manual of the SDRAM for the idle time to be needed. When the pulse width of the reset signal is longer than the idle time, mode register setting can be started immediately after the reset, but care should be taken when the pulse width of the reset signal is shorter than the idle time.



Note 1. Address pin to be connected to pin A10 of SDRAM.
Note 2. The waveform for DACKn is when active-low is specified.

Figure 13.29 SDRAM mode set timing (based on JEDEC)

(12) Low-Power SDRAM

The low-power SDRAM can be accessed using the same protocol as the normal SDRAM.

The differences between the low-power SDRAM and normal SDRAM are that partial refresh takes place that puts only a part of the SDRAM in the self-refresh state during the self-refresh function, and that power consumption is low during refresh under user conditions such as the operating temperature. The partial refresh is effective in systems in which the data in a work area other than the specific area can be lost without severe repercussions. For details, please see the Data Sheet for the low-power SDRAM to be used.

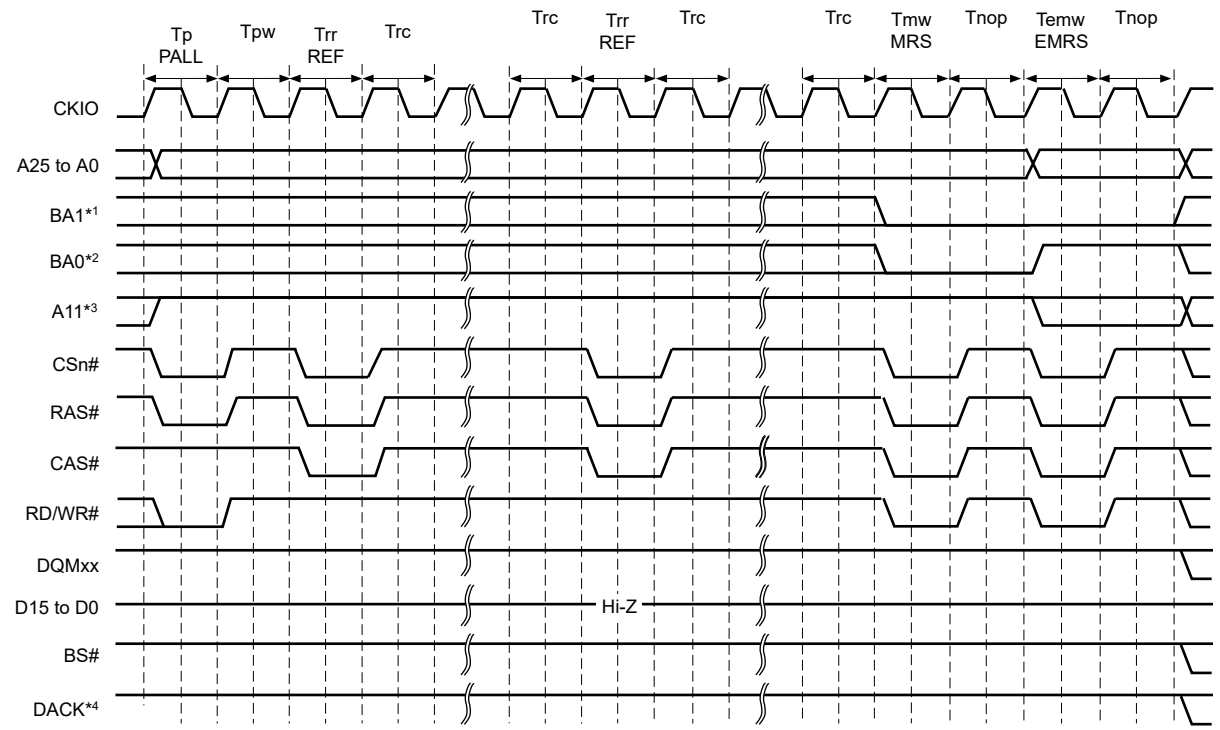
The low-power SDRAM supports the extended mode register in addition to the mode registers as the normal SDRAM.

This LSI supports issuing of the extended mode register set command (EMRS).

The EMRS command is issued according to the conditions specified in table below. For example, if data 0x0YYY_YYYY is written to address 0x8021_2XX0 in 32 bits, the commands are issued to the CS3 space in the following sequence: PALL -> REF × 8 -> MRS -> EMRS. In this case, the MRS and EMRS issue addresses are 0x0000XX0 and 0xYYYYYYY, respectively. If data 0x1YYY_YYYY is written to address 0x8021_2XX0 in 32 bits, the commands are issued to the CS3 space in the following sequence: PALL -> MRS -> EMRS.

Table 13.15 Output addresses when EMRS command is issued

Command to be issued	Access address	Access data	Write access size	MRS command issue address	EMRS command issue address
CS2 MRS	0x8021_1XX0	0x****_****	16 bits	0x0000XX0	0x-------
CS3 MRS	0x8021_2XX0	0x****_****	16 bits	0x0000XX0	0x-------
CS2 MRS + EMRS (with refresh)	0x8021_1XX0	0x0YYY_YYYY	32 bits	0x0000XX0	0xYYYYYYY
CS3 MRS + EMRS (with refresh)	0x8021_2XX0	0x0YYY_YYYY	32 bits	0x0000XX0	0xYYYYYYY
CS2 MRS + EMRS (without refresh)	0x8021_1XX0	0x1YYY_YYYY	32 bits	0x0000XX0	0xYYYYYYY
CS3 MRS + EMRS (without refresh)	0x8021_2XX0	0x1YYY_YYYY	32 bits	0x0000XX0	0xYYYYYYY

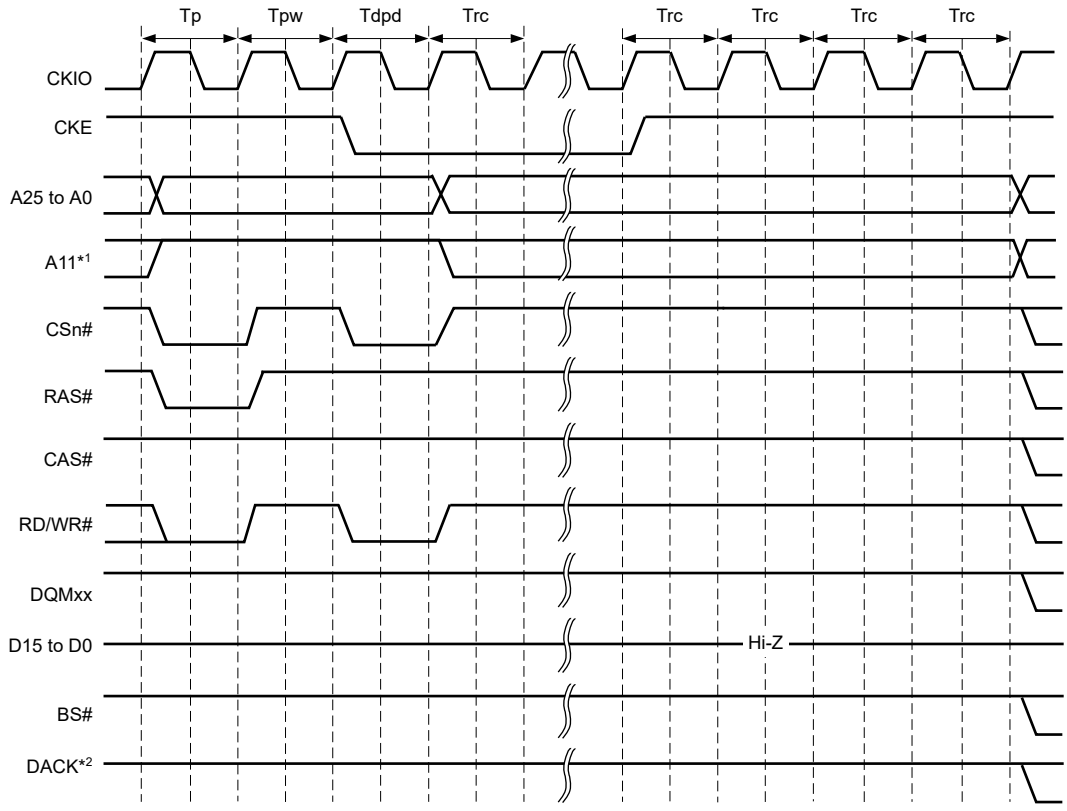


Note 1. Address pin to be connected to pin BA1 of SDRAM.
Note 2. Address pin to be connected to pin BA0 of SDRAM.
Note 3. Address pin to be connected to pin A10 of SDRAM.
Note 4. The waveform for DACKn is when active-low is specified.

Figure 13.30 EMRS command issue timing

- Deep power-down mode
  The low-power SDRAM supports the deep power-down mode as a low-power consumption mode. In the partial self-refresh function, self-refresh is performed on a specific area. In the deep power-down mode, self-refresh will not be performed on any memory area. This mode is effective in systems where all of the system memory areas are used as work areas.

If the RMODE bit in the SDCR is set to 1 while the DEEP and RFSH bits in the SDCR register are set to 1, the lowpower SDRAM enters the deep power-down mode. If the RMODE bit is cleared to 0, the CKE signal is pulled high to cancel the deep power-down mode. Before executing an access after returning from the deep power-down mode, the power-up sequence must be re-executed.



Note 1. Address pin to be connected to pin A10 of SDRAM.
Note 2. The waveform for DACKn is when active-low is specified

Figure 13.31 Deep power-down mode transition timing

13.4.7 Burst ROM (Clocked Asynchronous) Interface

The burst ROM (clocked asynchronous) interface is used to access a memory with a high-speed read function using a method of address switching called the burst mode or page mode. In a burst ROM (clocked asynchronous) interface, basically, the same access as the SRAM interface is performed, but the 2nd and subsequent access cycles are performed only by changing the address, without inactivating the RD# signal at the end of the 1st cycle. In the 2nd and subsequent access cycles, addresses are changed at the falling edge of the CKIO signal.

For the 1st access cycle, the number of waits specified by the W3 to W0 bits in the CS0WCR_1 register is inserted. For the 2nd and subsequent access cycles, the number of wait cycles specified by the BW1 and BW0 bits in the CS0WCR_1 register is inserted.

In the access to the burst ROM (clocked asynchronous), the BS# signal is activated only to the first access cycle. An external wait input is valid only to the first access cycle.

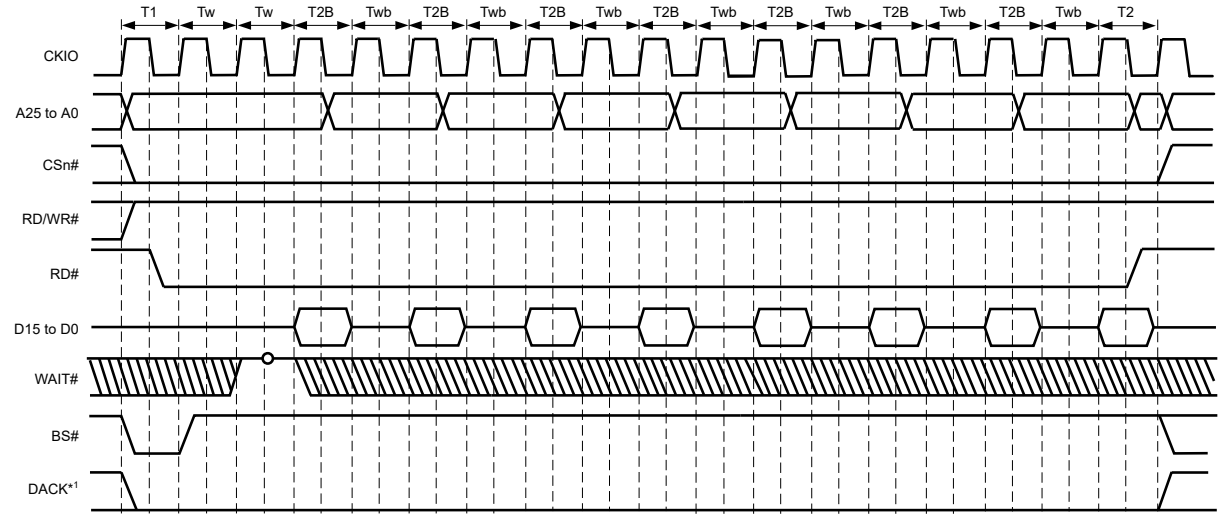
In the single access or write access that does not perform the burst operation in the burst ROM (clocked asynchronous) interface, access timing is the same as an SRAM interface.

Table 13.16 lists a relationship between bus width, access size, and the number of bursts. Figure 13.32 shows a timing chart.

Table 13.16 Relationship between bus width, access size, and number of bursts

Bus width	Access size	CSnWCR.BST[1:0] bits	Number of bursts	Access count
8 bits	8 bits	Not affected	1	1
	16 bits	Not affected	2	1
	32 bits	Not affected	4	1
	16 bytes	00	16	1
		01	4	4
	32 bytes	00	16	2
		01	4	8
	64 bytes	00	16	4
		01	4	16
16 bits	8 bits	Not affected	1	1
	16 bits	Not affected	1	1
	32 bits	Not affected	2	1
	16 bytes	00	8	1
		01	2	4
		10*1	4	2
			2, 4, 2	3
	32 bytes	00	8	2
		01	2	8
		10*1	4	4
			2, 4, 2	6
	64 bytes	00	8	4
		01	2	16
		10*1	4	8
			2, 4, 2	12

Note 1. When the bus width is 16 bits, the access size is 16 bits or more, and the BST[1:0] bits in CS0WCR_1 are 10b, the number of bursts and access count depends on the access start address. At address 0x***0 or 0x***8, 4-4 burst access is performed (burst numbers: 4, 4 for a total of 2 rounds of access). At address 0x***4 or 0x***C, 2-4-2 burst access is performed (burst numbers: 2, 4, 2 for a total of 3 rounds of access).



Note 1. The waveform for DACKn is when active-low is specified.

Figure 13.32 Burst ROM access timing (clocked asynchronous) (bus width = 16 bits, 16-byte transfer (number of burst 8), wait cycles inserted in first access = 2, wait cycles inserted in second and subsequent access cycles = 1)

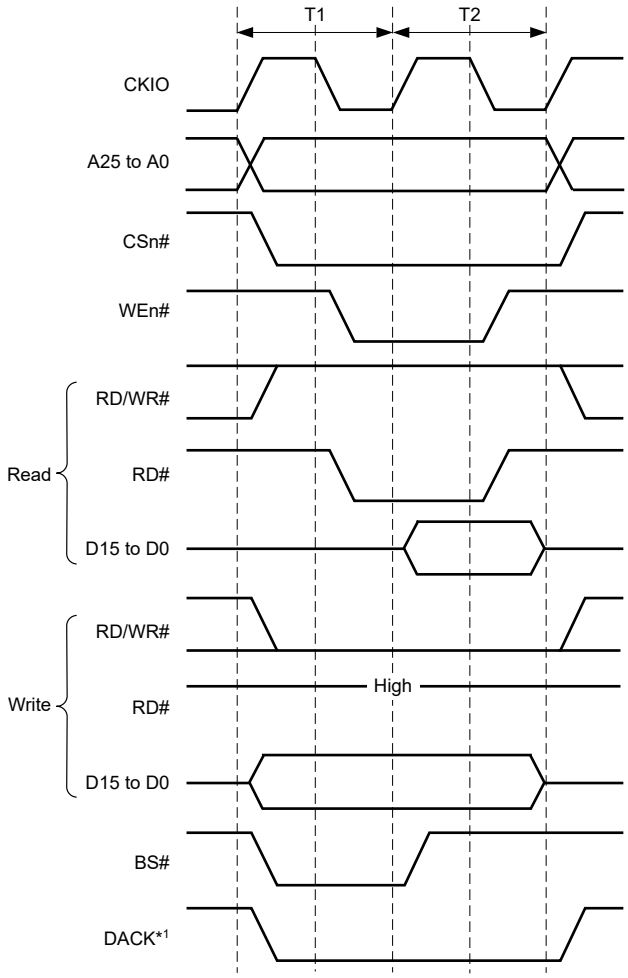
13.4.8 SRAM Interface with Byte Selection

The SRAM interface with byte selection is a memory interface that outputs the byte selection signal (WEn#) in both read and write bus cycles. This interface has 16-bit data pins and accesses SRAMs having upper and lower byte selection pins, such as UB# and LB#.

When the BAS bit in CSnWCR is cleared to 0 (value after reset), the write access timing of the SRAM interface with byte selection is the same as that for the SRAM interface. While in read access of a byte-selection SRAM interface, the byte-selection signal is output from the WEn# pin, which is different from that for the SRAM interface. The basic access timing is shown in Figure 13.33. In write access, data is written to the memory according to the timing of the byte-selection pin (WEn#). For details, please see the Data Sheet for the corresponding memory.

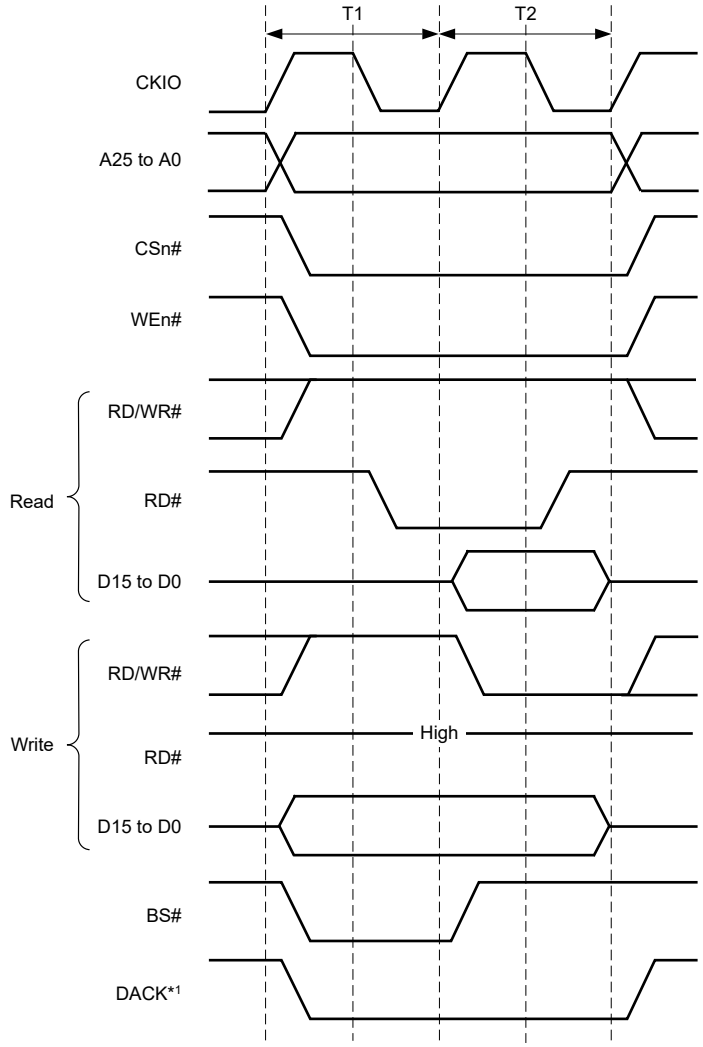
If the BAS bit in CSnWCR is set to 1, the WEn# pin and RD/WR# pin timings change. Figure 13.34 shows the basic access timing. In write access, data is written to the memory according to the timing of the write enable pin (RD/WR#).

The data hold of write data at the timing of RD/WR# inactive must be acquired by setting the HW1 and HW0 bits in the CSnWCR register. Figure 13.35 shows the access timing when a software wait is specified.



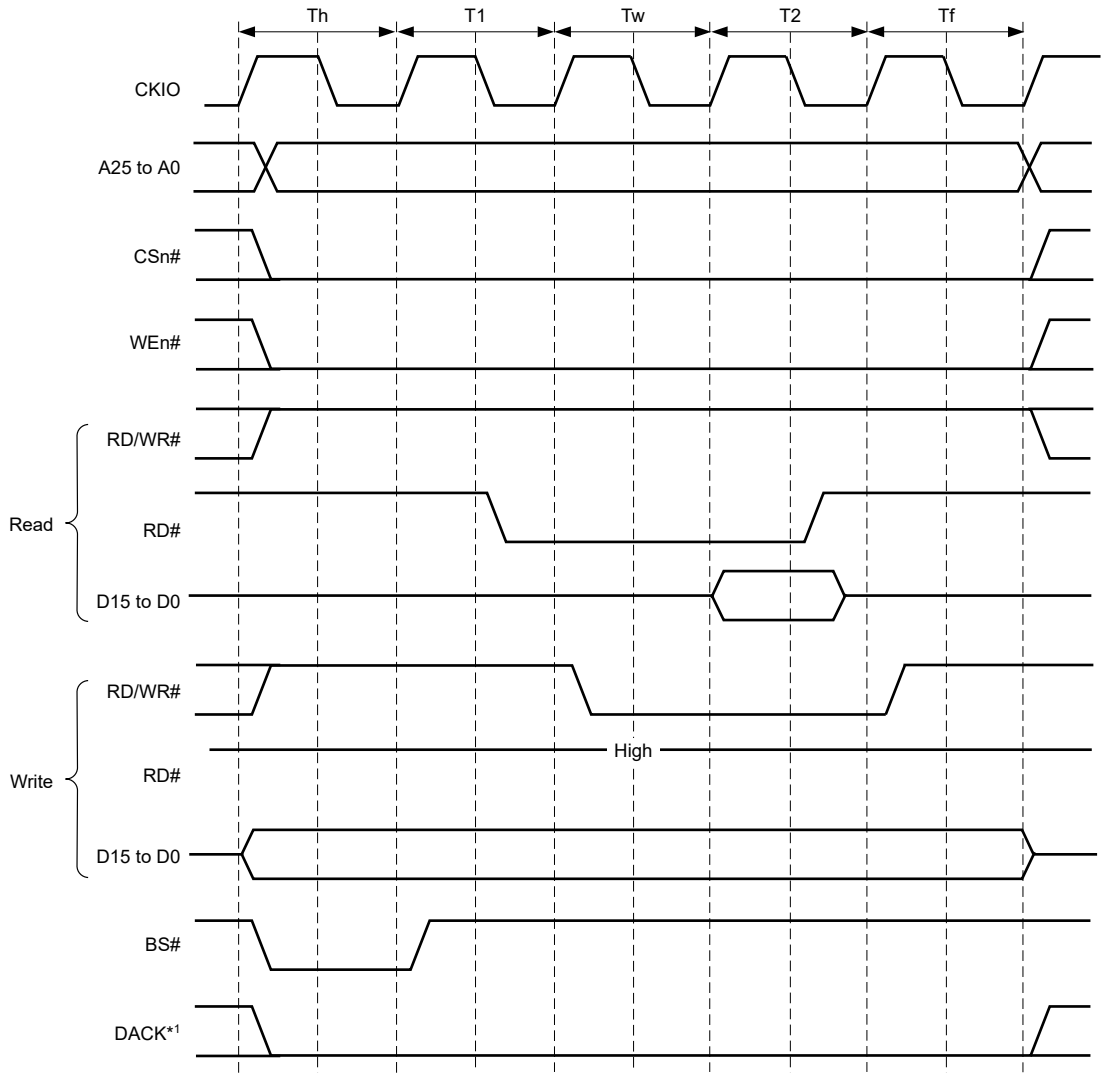
Note 1. The waveform for DACKn is when active-low is specified.

Figure 13.33 Basic access timing for SRAM with byte selection (BAS = 0)



Note 1. The waveform for DACKn is when active-low is specified.

Figure 13.34 Basic access timing for SRAM with byte selection (BAS = 1)



Note 1. The waveform for DACKn is when active-low is specified.

Figure 13.35 Wait timing for SRAM with byte selection (BAS = 1) (SW[1:0] = 01b, WR[3:0] = 0001b, HW[1:0] = 01b)

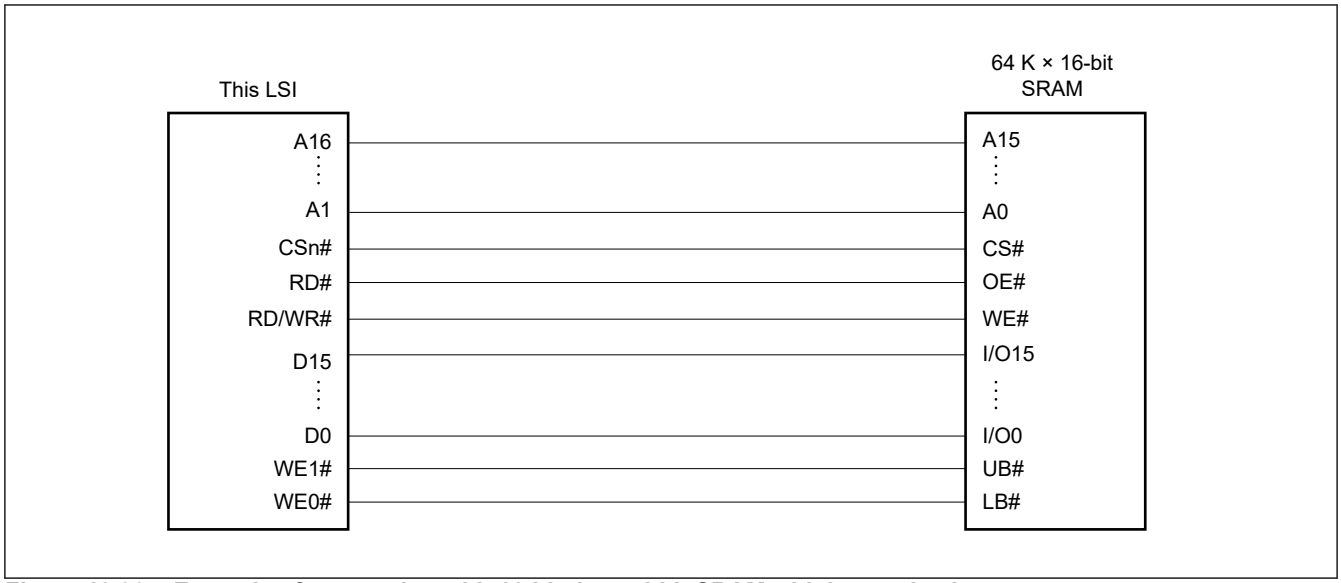


Figure 13.36 Example of connection with 16-bit data-width SRAM with byte selection

13.4.9 Burst ROM (Clocked Synchronous) Interface

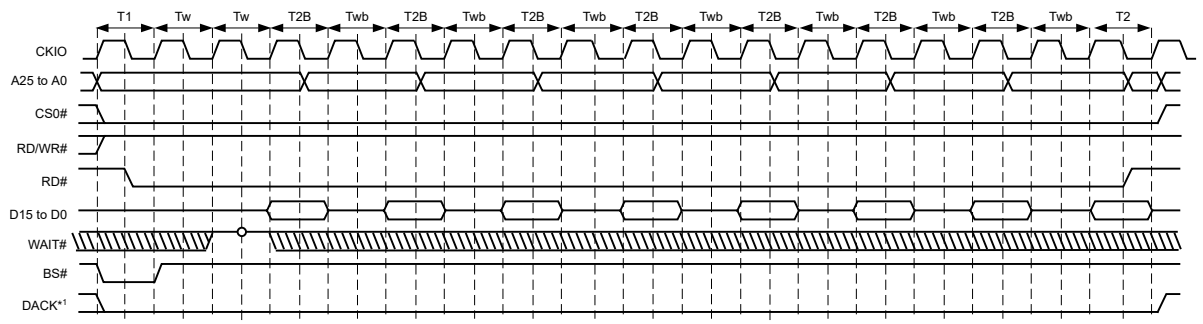
The burst ROM (clocked synchronous) interface is supported to access a ROM with a synchronous burst function at high speed. The burst ROM interface accesses the burst ROM in the same way as an SRAM interface. This interface is valid only for area 0.

In the first access cycle, wait cycles are inserted. In this case, the number of wait cycles to be inserted is specified by the W3 to W0 bits in CS0WCR. In the second and subsequent cycles, the number of wait cycles to be inserted is specified by the BW1 and BW0 bits in CS0WCR.

While the burst ROM (clocked synchronous) is accessed, the BS# signal is activated only for the first access cycle and an external wait input is also valid for the first access cycle.

When the bus width is 16 bits, the burst length must be specified as 8. The burst ROM interface does not support the 8-bit bus width for the burst ROM.

The burst ROM interface performs burst operations for all read access. For example, in a 32-bit access over a 16-bit bus, valid 16-bit data is read two times and invalid 16-bit data is read six times. These invalid data read cycles increase the memory access time and degrade the program execution speed and DMA transfer speed. To prevent this problem, it is recommended using a read in a 16-byte or more access size. The burst ROM interface performs write access in the same way as SRAM interface access.



Note 1. The waveform for DACKn is when active-low is specified.

Figure 13.37 Burst ROM access timing (clocked synchronous) (burst length = 8, wait cycles inserted in first access = 2, wait cycles inserted in second and subsequent access cycles = 1)

13.4.10 Wait between Access Cycles

As the operating frequency of LSIs becomes higher, the off-operation of the output data buffer for devices with slow access speed often collides with the data output for the next cycle. As a result of these collisions, the reliability of the device might be lowered or malfunctions might occur. Data collisions can be avoided by inserting idle (wait) cycles between continuous access cycles.

The number of idle states between access cycles can be set by the WM bit in CSnWCR, bits IWW2 to IWW0, IWRWD2 to IWRWD0, IWRWS2 to IWRWS0, IWRRD2 to IWRRD0, and IWRRS2 to IWRRS 0 in CSnBCR. The conditions for setting the idle cycles between access cycles are shown below.

1. Continuous access cycles are write-read or write-write
2. Continuous access cycles are read-write for different spaces
3. Continuous access cycles are read-write for the same space
4. Continuous access cycles are read-read for different spaces
5. Continuous access cycles are read-read for the same space

For the specification of the number of idle cycles between access cycles described above, see the description of each register.

Besides the idle states between access cycles specified by the registers, idle cycles must be inserted to interface with the internal bus or to obtain the minimum pulse width for a multiplexed pin (WEn#). The following gives detailed information about the idle cycles and describes how to estimate the number of idle cycles.

The number of idle cycles on the external bus from the inactive state of the CSn# signal to the active state of CSn# is described below.

There are seven conditions that determine the number of idle cycles on the external bus as shown in Table 13.17. The effects of these conditions are shown in Figure 13.38.

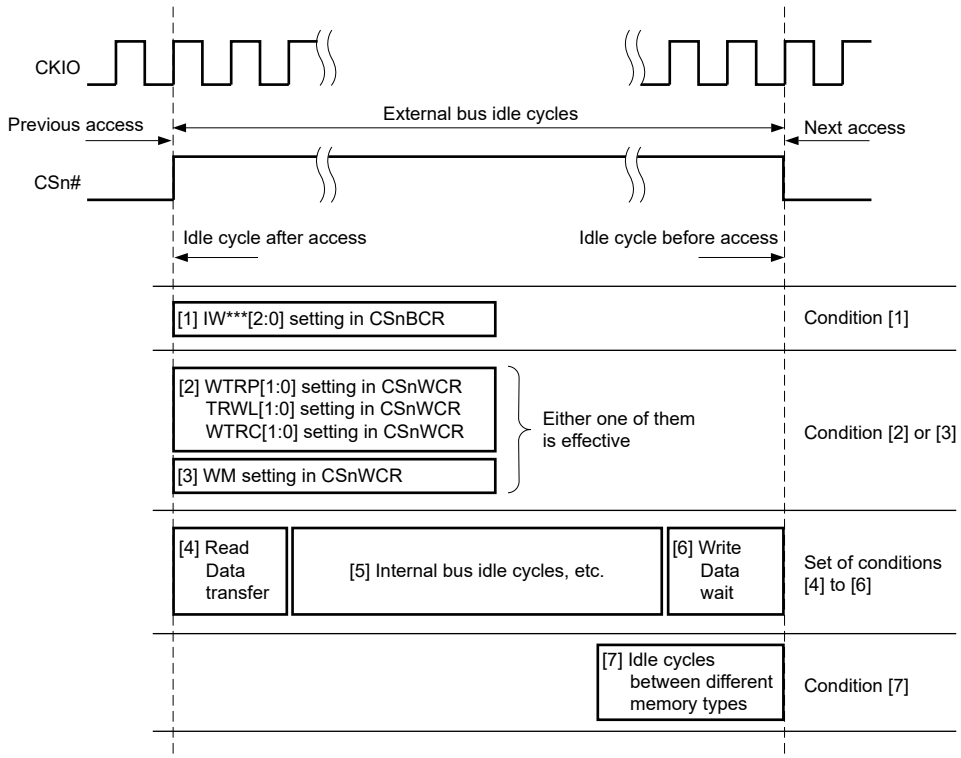
Table 13.17 Conditions for determining number of idle cycles (1 of 2)

No.	Condition	Description	Range	Note
[1]	IW***[2:0] in CSnBCR	These bits specify the number of idle cycles for access. The number of idle cycles can be specified independently for each combination of the previous and next cycles. For example, in the case where reading CS2 space followed by reading other CS space, the bits IWRRD[2:0] in CS2BCR should be set to 100b to specify six or more idle cycles. This condition is effective only for access states other than single address transfer and generates idle states after the access is completed.	0 to 12	Do not set 0 for the number of idle cycles between memory types which are not allowed to be accessed successively.
[2]	SDRAM-related bits in CSnWCR	These bits specify precharge completion and startup wait states and idle states between commands for SDRAM access. This condition is effective only for SDRAM access and generates idle states after the access is completed.	0 to 3	Specify these bits in accordance with the specification of the target SDRAM.
[3]	WM in CSnWCR	This bit enables or disables external WAIT# pin input for the memory types other than SDRAM. When this bit is cleared to 0 (external WAIT# enabled), one idle state is inserted to check the external WAIT# pin input after the access is completed. When this bit is set to 1 (disabled), no idle state is generated.	0 or 1	—
[4]	Read data transfer cycle	One idle state is inserted after a read access is completed. This idle state is not generated for the first or middle cycles in divided access cycles. This is neither generated when the HW[1:0] bits in CSnWCR are not 00b.	0 or 1	One idle state is always inserted during a read cycle with SDRAM.

Table 13.17 Conditions for determining number of idle cycles (2 of 2)

No.	Condition	Description	Range	Note
[5]	Internal bus idle cycles, etc.	External bus access requests from the CPU or the direct memory access controller (DMA) and their results are passed through the internal bus. The external bus enters idle state during internal bus idle cycles or while a bus other than the external bus is being accessed. This condition is not effective for divided access cycles, which are generated by the bus state controller when the access size is larger than the external data bus width.	0 or larger	The number of internal bus idle cycles may not become 0 depending on the CPU: internal bus: CKIO.
[6]	Write data wait cycles	During write access, a write cycle is executed on the external bus only after the write data becomes ready. This write data wait period generates idle cycles before the write cycle. Note that when the previous cycle is a write cycle and the internal bus idle cycles are shorter than the previous write cycle, write data can be prepared in parallel with the previous write cycle and therefore, no idle cycle is generated (write buffer effect).	0 or 1	For write → write or write → read access cycles, successive access cycles without idle cycles may be available due to the write buffer effect described in the left column. If successive access cycles without idle cycles are not allowed, specify the minimum number of idle cycles between access cycles through the CSnBCR register.
[7]	Idle cycles between different memory types	To ensure the minimum pulse width on the multi-use pins, idle cycles may be inserted before access after memories are switched. For some memory types, idle cycles are inserted even when memory types are not switched.	0 to 2	The number of idle cycles depends on the target memory types. See Table 13.18.

In the above conditions, a total of four conditions, that is, condition [1], condition [2] or [3] (either one is effective), a set of conditions [4] to [6] (these are generated successively, and therefore the sum of them should be taken as one set of idle cycles), and condition [7] are generated at the same time. The maximum number of idle cycles among these four conditions becomes the number of idle cycles on the external bus. To ensure the minimum idle cycles, be sure to make register settings for condition [1].



Note: A total of four conditions (condition [1], condition [2] or [3], a set of conditions [4] to [6], and condition [7]) generate idle cycle at the same time. Accordingly, the maximum number of cycles among these four conditions become the number of idle cycles.

Figure 13.38 Idle cycle conditions

Table 13.18 Number of idle cycles inserted between access cycles to different memory types

Previous cycle	Next Cycle						
	SRAM	Burst ROM (asynchronous)	MPX-I/O	Byte SRAM (BAS = 0)	Byte SRAM (BAS = 1)	SDRAM	Burst ROM (synchronous)
SRAM	0	0	1	0	0/1*1	0/1*1	0
Burst ROM (asynchronous)	0	0	1	0	0/1*1	0/1*1	0
MPX-I/O	1	1	0	1	1	1	1
Byte SRAM (BAS = 0)	0	0	1	0	0/1*1	0/1*1	0
Byte SRAM (BAS = 1)	0/1*1	0/1*1	1/2*1	0/1*1	0	0	0/1*1
SDRAM	1	1	2	1	0	0	1
Burst ROM (synchronous)	0	0	1	0	1	1	0

Note 1. The number of idle cycles depends on the setting of the CSnWCR.HW[1:0] bits in the previous cycle. When HW[1:0] ≠ 00b, the number of idle cycles is the value on the left; when HM[1:0] = 00b, the number of idle cycles is the value on the right. If the CSnWCR.HW[1:0] bits were for a CSn space that does not exist in the previous cycle, the number of idle cycles is the value on the right.

13.4.11 Others

(1) Reset

This module can be initialized completely only at reset of the entire chip (pin reset, software reset, and error reset).

When reset of the entire chip occurs, all signals are inactivated and data output buffers are turned off regardless of the bus cycle state after the internal reset is synchronized with the internal clock. All control registers are initialized. In the module stop state, control registers of the bus state controller are not initialized.

(2) Caution on Write Buffer

Since the bus state controller incorporates a one-stage write buffer, it can execute an access via the internal bus before the previous external bus cycle is completed in a write cycle. If the on-chip module is read or written after the external low-speed memory is written, the on-chip module can be accessed before the completion of the external low-speed memory write cycle.

In read cycles, the CPU is placed in the wait state until read operation has been completed. To continue the process after the data write to the device has been completed, perform a dummy read to the same address to check for completion of the write before the next process to be executed.

The write buffer of the bus state controller functions in the same way for an access by a bus master other than the CPU such as the direct memory access controller (DMA). Accordingly, to perform DMA transfers, the next read cycle is initiated before the previous write cycle is completed. Note, however, that if both the DMA source and destination addresses exist in external memory space, the next read cycle will not be initiated until the previous write cycle is completed.

Changing the registers in this module while the write buffer is operating may disrupt correct write access. Therefore, do not change the registers in this module immediately after a write access. If this change becomes necessary, do it after executing a dummy read of the write data.

14. DMA Controller (DMAC)

14.1 Overview

The LSI contains Direct Memory Access Controller (DMAC) consisting of two units, DMAC0 and DMAC1.

DMAC transfers data without using the CPU. When a transfer is requested, DMAC transfers data stored at the transfer source address to the transfer destination address.

Table 14.1 lists the specifications of DMAC.

Table 14.1 DMAC specifications

Parameter		Description	
		DMAC0	DMAC1
Number of channels		16 channels	16 channels
Address space		4 GB	
DMAC activation source		• External request (DREQ) • External interrupts (IRQ0 to IRQ13) • On-chip peripheral module requests/software requests*1	
Channel priority		• Priority among channels: Fixed priority levels within the group of channels 0 to 7 and within the group of channels 8 to 15, respectively, or round-robin scheduling for all channels. • Priority between channel groups: Round-robin scheduling between the group of channels 0 to 7 and the group of channels 8 to 15.	
Transfer data unit		8, 16, 32, 64, 128, 256, and 512 bits	8, 16, 32, 64, 128, and 256 bits
Maximum transfer size		2^{32} - 1 bytes	
Transfer mode	Single transfer	Performs DMA transfer for each DMA transfer request.	
	Block transfer	Performs DMA transfer of the specified transfer size for a DMAC activation request.	
DMA mode	Register mode	• DMA transfer setting value: Control register value within the DMA controller • DMA transfer from the source to the destination specified by a register.	
	Link mode	• DMA transfer setting value: Descriptors in the internal RAM or an external memory • Various DMA transfers specified by descriptors can be performed (responsiveness: register mode > link mode).	
Interval function		The DMA transfer interval can be specified (bus occupation ratio can be adjusted).	
Suspending function		Current DMA transfer can be paused.	
Buffer flush function		When DMAC is stopped forcibly, data in the buffer can be flushed.	
Interrupt request		Each channel has the following interrupt requests: • Transfer completion: Indicates completion of transfer of the specified size; each channel has this source. • Transfer error: Indicates a bus error; total of two sources, one for unit 0 and the other for unit 1.	

Note 1. A software request is output as a source of an on-chip peripheral module request from the interrupt controller. For how to set a software request, see section 11, Interrupt Controller (ICU).

Table 14.2 lists the input/output pins of the DMAC.

Table 14.2 Pin configuration of DMAC

Pin name	I/O	Description
DREQ	Input	DMA transfer request input from an external device to DMAC
DACK	Output	Output of DMAC transfer request acceptance from DMAC to an external device
TEND	Output	Output of transfer completion from DMAC to an external device

Note: For details on active levels of DACK and TEND, see section 14.2.11. CMNCR : Common Control Register.

14.2 Register Descriptions

14.2.1 NmSA_n : Nextm Source Address Register n (m = 0, 1) (n = 0 to 15)

Base address: DMACi = 0x8008_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x000 + 0x40 × n (m = 0) (n = 0 to 7)
0x400 + 0x40 × (n - 8) (m = 0) (n = 8 to 15)
0x00C + 0x40 × n (m = 1) (n = 0 to 7)
0x40C + 0x40 × (n - 8) (m = 1) (n = 8 to 15)

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Source Address Sets the start address of the DMA transfer source.	R/W

Note: During link mode transfer, descriptor read data is set to the N0SA_n register automatically.

The N0SA_n and N1SA_n registers set the DMA transfer source address of DMA channel n (n = 0 to 15). The N0SA_n register is for Next0 Register Set, and the N1SA_n register is for Next1 Register Set.

14.2.2 NmDA_n : Nextm Destination Address Register n (m = 0, 1) (n = 0 to 15)

Base address: DMACi = 0x8008_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x004 + 0x40 × n (m = 0) (n = 0 to 7)
0x404 + 0x40 × (n - 8) (m = 0) (n = 8 to 15)
0x010 + 0x40 × n (m = 1) (n = 0 to 7)
0x410 + 0x40 × (n - 8) (m = 1) (n = 8 to 15)

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Destination Address Sets the start address of the DMA transfer destination.	R/W

Note: During link mode transfer, descriptor read data is set to the N0DA_n register automatically.

The N0DA_n and N1DA_n registers set the DMA transfer destination address of DMA channel n (n = 0 to 15). The N0DA_n register is for Next0 Register Set, and the N1DA_n register is for the Next1 Register Set.

14.2.3 NmTB_n : Nextm Transaction Byte Register n (m = 0, 1) (n = 0 to 15)

Base address: DMACi = 0x8008_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x008 + 0x40 × n (m = 0) (n = 0 to 7)
0x408 + 0x40 × (n - 8) (m = 0) (n = 8 to 15)
0x014 + 0x40 × n (m = 1) (n = 0 to 7)
0x414 + 0x40 × (n - 8) (m = 1) (n = 8 to 15)

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Transaction Byte Sets the total number of transfer bytes. Do not start DMA transfer when 0 is set.	R/W

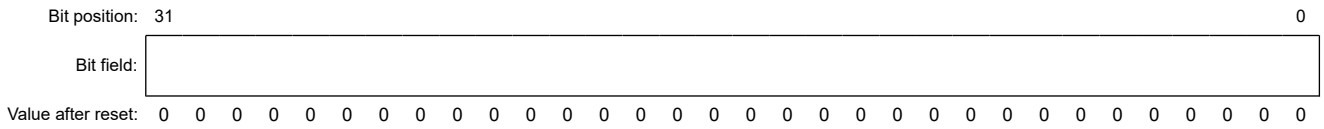
Note: During link mode transfer, descriptor read data is set to the N0TB_n register automatically.

The N0TB_n and N1TB_n registers set the total number of transfer bytes of DMA channel n (n = 0 to 15). The N0TB_n register is for Next0 Register Set, and the N1TB_n register is for Next1 Register Set.

14.2.4 CRSA_n : Current Source Address Register n (n = 0 to 15)

Base address: DMACi = 0x8008_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x018 + 0x40 × n (n = 0 to 7)
0x418 + 0x40 × (n - 8) (n = 8 to 15)



Bit	Symbol	Function	R/W
31:0	n/a	Current Source Address Displays the read address of the next DMA transfer.	R

The CRSA_n register is a register that displays the DMA transfer source address of DMA channel n (n = 0 to 15). During DMA transfer, the value is incremented automatically (fixed when CHCFG_n.SAD = 1).

The value after a reset is loaded from the following registers:

- In register mode:
  Loads the transfer source address from the Next0/1 register.
- In link mode:
  Loads the transfer source address from the descriptor read data (the hardware inputs the descriptor read data to the N0SA_n register automatically, and loads it to the CRSA_n register when transfer starts).

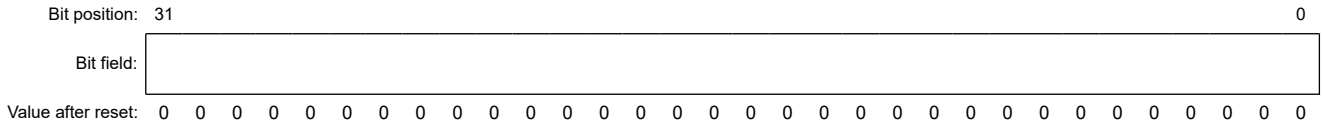
The value is incremented when reading data from the transfer source completes.

Read this register after DMA stops (when CHSTAT_n.EN = 0). Handle the value during DMA operation as a reference value.

14.2.5 CRDA_n : Current Destination Address Register n (n = 0 to 15)

Base address: DMACi = 0x8008_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x01C + 0x40 × n (n = 0 to 7)
0x41C + 0x40 × (n - 8) (n = 8 to 15)



Bit	Symbol	Function	R/W
31:0	n/a	Current Destination Address Displays the write address of the next DMA transfer.	R

The CRDA_n register displays the DMA transfer destination address of DMA channel n (n = 0 to 15).

During DMA transfer, the value is incremented automatically (fixed when CHCFG_n.DAD = 1).

The value after a reset is loaded from the following registers:

- In register mode:

Loads the transfer destination address from the Next0/1 register.

- In link mode:
  Loads the transfer destination address from the descriptor read data (the hardware inputs the descriptor read data to the N0DA_n register automatically, and loads it to the CRDA_n register when transfer starts).

The value is incremented when writing data to the transfer destination completes.

Read this register after DMA stops (when CHSTAT_n.EN = 0). Handle the value during DMA operation as a reference value.

14.2.6 CRTB_n : Current Transaction Byte Register n (n = 0 to 15)

Base address: DMACi = 0x8008_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x020 + 0x40 × n (n = 0 to 7)
0x420 + 0x40 × (n - 8) (n = 8 to 15)

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Current Transaction Byte Displays the remaining number of transfer bytes that is currently performed.	R

The CRTB_n register displays the total number of transfer bytes of DMA channel n (n = 0 to 15). The value is cleared to 0 when transfer completes.

During DMA transfer, the value is decremented automatically.

The value after a reset is loaded from the following registers:

- In register mode:
  Loads the number of transfer bytes from the Next0/1 register.
- In link mode:
  Loads the number of transfer bytes from descriptor read data (the hardware inputs the descriptor read data to the N0TB_n register automatically and loads it to the CRTB_n register when transfer starts).

The value is decremented when writing data to the transfer destination completes.

Read this register after DMA stops (when CHSTAT_n.EN = 0). Handle the value during DMA operation as a reference value.

14.2.7 CHSTAT_n : Channel Status Register n (n = 0 to 15)

Base address: DMACi = 0x8008_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x024 + 0x40 × n (n = 0 to 7)
0x424 + 0x40 × (n - 8) (n = 8 to 15)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	INTM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	MODE	DER	DW	DL	SR	TC	END	ER	SUS	TACT	RQST	EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EN	DMA Activation Enable Displays the operation status of DMA channel n. [Setting condition] ● Writing 1 to the SETEN bit in the CHCTRL_n register [Clearing conditions] When any of the following conditions is met: ● Writing 1 to the CLREN bit in the CHCTRL_n register ● Writing 1 to the SWRST bit in the CHCTRL_n register ● When a bus error is reported during transfer ● When all DMA transfers complete in register mode (transfers complete when the REN bit in the CHCFG_n register = 0) ● In link mode, when DMA transfer (write back when WBD = 0) of the last descriptor (LE = 1) completes ● When reading the descriptor in link mode stopped (when LV = 0) 0: Operation disabled 1: Operation enabled	R
1	RQST	DMA Transfer Request This bit indicates that the transfer request is accepted. [Setting condition] ● The transfer request is accepted. [Clearing conditions] When any of the following conditions is met: ● Writing 1 to the SWRST bit in the CHCTRL_n register ● Writing 1 to the STG bit in the CHCTRL_n register ● Writing 1 to the CLRRQ bit in the CHCTRL_n register ● In single transfer (the TM bit in the CHCFG_n register = 0) mode, the transfer specified in the REQD bit in the CHCFG_n register is performed ● When all DMA transfers complete in register mode (transfer completes when the REN bit in the CHCFG_n register = 0) ● In link mode, DMA transfer of the last descriptor (LE = 1) completes ● In link mode, reading the descriptor is disabled (when LV = 0) ● In link mode, the DEM bit in the CHCFG_n register = 0, and DMA transfer completes ● When a bus error is reported to the master interface 0: DMA transfer request not accepted 1: DMA transfer request accepted	R
2	TACT	DMAC Operating Status This bit indicates that DMAC is running. This bit is used to ensure the channel stops completely. For details, see section 14.3.9. DMA Transfer Status. [Setting condition] ● Writing 1 to the SETEN bit in the CHCTRL_n register (reading the descriptor starts, or the DMA request is being serviced). [Clearing condition] ● When the internal state is idling (the EN bit in the CHSTAT_n register is cleared to 0, and all DMA transfer completes). 0: DMA in Channel_n stops 1: DMA in Channel_n is running	R
3	SUS	Suspend This bit indicates that the channel is suspended. For details, see section 14.3.10. Suspending a Transfer. [Setting condition] ● Writing 1 to the SETSUS bit in the CHCTRL_n register during DMA transfer of Channel_n, and the internal state is changed to the suspended state [Clearing conditions] When any of the following conditions is met: ● Writing 1 to the CLRSUS bit in the CHCTRL_n register ● Writing 1 to the CLREN bit in the CHCTRL_n register ● Condition for clearing the EN bit in the CHSTAT_n register 0: Channel_n not suspended 1: Channel_n suspended	R

Bit	Symbol	Function	R/W
4	ER	DMA Error This bit indicates that a DMA error interrupt is generated as a result of the bus error during DMA transfer. [Setting condition] ● A bus error is reported to the bus cycle [Clearing condition] ● Writing 1 to the SWRST bit in the CHCTRL_n register 0: No bus error occurred 1: A bus error occurred	R
5	END	DMA Transfer Completion Interrupt This bit indicates that DMA transfer completes, and a DMA interrupt is generated. [Setting conditions] When any of the following conditions is met: ● When the TC bit setting condition is met while DEM bit in the CHCFG_n register = 0 ● When the descriptor is read in link mode, LV of header = 0, and DIM in the header register = 0 [Clearing conditions] When any of the following conditions is met: ● Writing 1 to the CLREND bit in the CHCTRL_n register ● Writing 1 to the SWRST bit in the CHCTRL_n register 0: DMA transfer not completed 1: DMA transfer completed	R
6	TC	DMA Transfer Completion (total number of data bytes for transaction) This bit indicates that DMA transfer is completed. [Setting conditions] When any of the following conditions is met: ● When a transfer for the total number of data bytes for transaction set in the CRTB_n register is completed in register mode ● When the header bit WBD = 1 in link mode, and a transfer for the total number of data bytes for transaction set in the CRTB_n register is completed ● When the header bit WBD = 0 in link mode, and write-back to the descriptor is completed [Clearing conditions] When any of the following conditions is met: ● Writing 1 to the CLRTC bit in the CHCTRL_n register ● Writing 1 to the SWRST bit in the CHCTRL_n register 0: DMA transfer not completed 1: DMA transfer completed	R
7	SR	Next Register Select In register mode, this bit indicates the selected register set. [Setting condition] ● Setting 1 to the RSEL bit in the CHCFG_n register [Clearing condition] ● Clearing the RSEL bit in the CHCFG_n register to 0 0: Next0 register set 1: Next1 register set	R
8	DL	Descriptor Load This bit indicates that the descriptor is being read. In addition, if a bus error is reported while the descriptor is being read, 1 is retained. [Setting condition] ● Reading the descriptor in link mode is started [Clearing conditions] When any of the following conditions is met: ● Reading the descriptor in link mode completes with the OK response ● Writing 1 to the SWRST bit in the CHCTRL_n register (if 1 is retained for the bus error, it can be cleared to 0 only by the SWRST bit) 0: The descriptor is not being read 1: When ER = 0 The descriptor is being read in link mode When ER = 1 A bus error occurred while the descriptor is being read in link mode	R

Bit	Symbol	Function	R/W
9	DW	Descriptor Write Back This bit indicates that the descriptor is being written back. In addition, if a bus error is reported while the descriptor is written back, 1 is retained. [Setting condition] ● Writing back of header in link mode is started [Clearing conditions] ● Writing back of header in link mode completes with the OK response ● Writing 1 to the SWRST bit in the CHCTRL_n register (if 1 is retained due to the bus error, it can be cleared to 0 only by the SWRST bit) 0: Header is not written back in link mode 1: The ER bit in the CHSTAT_n register = 0 Header is written back in link mode The ER bit in the CHSTAT_n register = 1 A bus error occurred when header is written back in link mode	R
10	DER	Descriptor Error This bit indicates that the read descriptor is invalid (LV = 0) (does not depend on the value of the DIM bit in the header register). [Setting condition] ● In link mode, the LV of the read descriptor is 0 [Clearing conditions] When the following condition is met: ● Writing 1 to the SWRST bit in the CHCTRL_n register 0: No descriptor error occurred 1: Descriptor error occurred	R
11	MODE	DMA Mode This bit indicates DMA mode and displays the setting value of the DMS bit in the CHCFG_n register. 0: Register mode 1: Link mode	R
15:12	—	These bits are read as 0.	R
16	INTM	DMA Transfer Completion Interrupt Request Mask This bit displays the temporary mask status of DMA interrupt output. [Setting condition] ● Writing 1 to the SETINTM bit in the CHCTRL_n register [Clearing conditions] When any of the following conditions is met: ● Writing 1 to the CLRINTM bit in the CHCTRL_n register ● Writing 1 to the SWRST bit in the CHCTRL_n register 0: The temporary mask status is cleared 1: Temporary mask status	R
31:17	—	These bits are read as 0.	R

Note:
- Handle the transfer as invalid when CHSTAT_n.ER is set to 1.
- To suspend DMA transfer, mask or clear the transfer request, or clear CHSTAT_n.EN. See section 14.3.11. Aborting a Transfer for the procedure).
- To request a transfer by software, make sure that the previously requested DMA transfer operation completes (by checking the current register), set the STG bit in the Channel Control Register n (CHCTRL_n), then activate DMA.
- When DMA transfer for the same channel is requested by both hardware and software, the DMA transfer request cannot be identified for which. Therefore, set the DMA transfer request on your system to either hardware or software.

The CHSTAT_n register displays the status of DMA channel n (n = 0 to 15).

14.2.8 CHEXT_n : Channel Extension Register n (n = 0 to 15)

Base address: DMACi = 0x8008_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x034 + 0x40 × n (n = 0 to 7)
0x434 + 0x40 × (n - 8) (n = 8 to 15)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DCA[3:0]				—	DPR[2:0]			SCA[3:0]				—	SPR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SPR[2:0]	Source PROT These bits set the value to be output to the PROT[2:0] pins during DMA read transfer.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
7:4	SCA[3:0]	Source CACHE These bits set the value to be output to the CACHE[3:0] pins during DMA read transfer.	R/W
10:8	DPR[2:0]	Destination PROT These bits set the value to be output to the PROT[2:0] pins during DMA write transfer.	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
15:12	DCA[3:0]	Destination CACHE These bits set the value to be output to the CACHE[3:0] pins during DMA write transfer.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The CHEXT_n register is an extension register for DMA channel n.

14.2.9 DMAC0_RSSELi : DMAC Unit 0 Resource Select Register i (i = 0 to 5)

Base address: DMA = 0x8009_0060

Offset address: 0x0C + 0x04 × i

Bit position:	31			28								20	18									10	8								0
Bit field:	—	—	—	REQ_SELC[8:0]								—	REQ_SELB[8:0]									—	REQ_SELA[8:0]								
Value after reset:	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
8:0	REQ_SELA[8:0]	DMA Resource Select for Channel n Select the trigger source of the DMA channel. 0x1FF: No resource assignment 0x000: ⋮ 0x1FE: DMA trigger source is assigned (event number)	R/W
9	—	This bit is read as 1. The write value should be 1.	R/W
18:10	REQ_SELB[8:0]	DMA Resource Select for Channel n + 1 Select the trigger source of the DMA channel. When i = 5, the REQ_SELB[8:0] bits are invalid. 0x1FF: No resource assignment 0x000: ⋮ 0x1FE: DMA trigger source is assigned (event number)	R/W

Bit	Symbol	Function	R/W
19	—	This bit is read as 1. The write value should be 1.	R/W
28:20	REQ_SELC[8:0]	DMA Resource Select for Channel n + 2 Select the trigger source of the DMA channel. When i = 5, the REQ_SELC[8:0] bits are invalid. 0x1FF: No resource assignment 0x000: ⋮ 0x1FE: DMA trigger source is assigned (event number)	R/W
29	—	This bit is read as 1. The write value should be 1.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note:
- Relationship between register number and channel number is n = i × 3.
- The [30:10] bits in the DMAC0_RSSEL5 register are read as 0. The write value should be 0.

The DMAC0_RSSELi register selects the activation trigger source for DMAC unit 0. For details on numbers that are selected by this source selection, see the event numbers listed in Table 14.19.

In addition, do not set the same source for multiple DMAC0_RSSELi and DMAC1_RSSELi registers. If the same source is set, the operation of this LSI cannot be guaranteed.

14.2.10 DMAC1_RSSELi : DMAC Unit 1 Resource Select Register i (i = 0 to 5)

Base address: DMA = 0x8009_0060

Offset address: 0x24 + 0x04 × i

Bit position:	31			28									20	18									10	8								0
Bit field:	—	—	—	REQ_SELC[8:0]									—	REQ_SELB[8:0]									—	REQ_SELA[8:0]								
Value after reset:	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
8:0	REQ_SELA[8:0]	DMA Resource Select for Channel n Select the trigger source of the DMA channel. 0x1FF: No resource assignment 0x000: ⋮ 0x1FE: DMA trigger source is assigned (event number)	R/W
9	—	This bit is read as 1. The write value should be 1.	R/W
18:10	REQ_SELB[8:0]	DMA Resource Select for Channel n + 1 Select the trigger source of the DMA channel. When i = 5, the REQ_SELB[8:0] bits are invalid. 0x1FF: No resource assignment 0x000: ⋮ 0x1FE: DMA trigger source is assigned (event number)	R/W
19	—	This bit is read as 1. The write value should be 1.	R/W
28:20	REQ_SELC[8:0]	DMA Resource Select for Channel n + 2 Select the trigger source of the DMA channel. When i = 5, the REQ_SELC[8:0] bits are invalid. 0x1FF: No resource assignment 0x000: ⋮ 0x1FE: DMA trigger source is assigned (event number)	R/W
29	—	This bit is read as 1. The write value should be 1.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note:
- Relationship between register number and channel number is n = i × 3.
- The [30:10] bits in the DMAC1_RSSEL5 register are read as 0. The write value should be 0.

The DMAC1_RSSELi register selects the activation trigger source for DMAC unit 1. For details on numbers that are selected by this source selection, see the event numbers listed in Table 14.19.

In addition, do not set the same source for multiple DMAC0_RSSELi and DMAC1_RSSELi registers. If the same source is set, the operation of this LSI cannot be guaranteed.

14.2.11 CMNCR : Common Control Register

Base address: BSC = 0x8021_0000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	TL	—	—	—	AL	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	DPRTY[1:0]		—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	1	0	0	0	0	0	0	0	1	1	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
4:3	—	These bits are read as 1. The write value should be 1.	R/W
8:5	—	These bits are read as 0. The write value should be 0.	R/W
10:9	DPRTY[1:0]	DMA Burst Transfer Priority 0 x: Accept a refresh request during DMA burst transfer 1 0: Do not accept a refresh request during DMA burst transfer 1 1: Setting prohibited	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	—	This bit is read as 1. The write value should be 1.	R/W
23:13	—	These bits are read as 0. The write value should be 0.	R/W
24	AL	Acknowledge Level Specify the active level of the DACK signal 0: Active-low output from DACK 1: Active-high output from DACK	R/W
27:25	—	These bits are read as 0. The write value should be 0.	R/W
28	TL	Transfer End Level Specify the active level of the TEND signal 0: Active-low output from TEND 1: Active-high output from TEND	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

The CMNCR register is a 32-bit register that controls the common items from each area.

14.2.12 CHCTRL_n : Channel Control Register n (n = 0 to 15)

Base address: DMACi = 0x8008_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x028 + 0x40 × n (n = 0 to 7)
0x428 + 0x40 × (n - 8) (n = 8 to 15)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRINTM	SETINTM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CLRSUS	SETSUS	—	CLRTC	CLREND	CLRRQ	SWRST	STG	CLREN	SETEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SETEN	DMA Activation Enable Enable DMA transfer of DMA channel n. When this bit is set with the SWRST bit, clearing by the SWRST bit takes precedence, and a transfer does not start. This bit is always read as 0. To reset the DMA register, stop the ongoing DMA transfer by setting the CLREN bit and then set the SETEN bit. 0: Operation is not affected 1: DMA transfer is enabled (the EN bit in the CHSTAT_n register is set)	R/W
1	CLREN	DMA Activation Enable Clear Clear the EN bit in the CHSTAT_n register (for details, see section 14.3.11. Aborting a Transfer). This bit is always read as 0. 0: Operation is not affected 1: DMA transfer is disabled (the EN bit in the CHSTAT_n register is cleared)	R/W
2	STG	Software Trigger Set a transfer request by software. When this bit is set at the same time as the SWRST bit is being set, clearing by the SWRST bit takes precedence. This bit is always read as 0. 0: Operation is not affected 1: A transfer request is set by software (the RQST bit in the CHSTAT_n register is set)	R/W
3	SWRST	Software Reset Clear each bit in the CHSTAT_n register (for details on the bit to be cleared, see the descriptions of the applicable bit). Set this bit to 0 when the EN bit and the TACT bit are cleared to 0. This bit is always read as 0. 0: Operation is not affected 1: Clear each bit in the CHSTAT_n register	R/W
4	CLRRQ	DMA Transfer Request Clear Clear the RQST bit in the CHSTAT_n register to 0. This bit is always read as 0. 0: Operation is not affected 1: Clear the RQST bit in the CHSTAT_n register	R/W
5	CLREND	END Clear Clear the END bit in the CHSTAT_n register to 0. This bit is always read as 0. 0: Operation is not affected 1: Clear the END bit	R/W
6	CLRTC	TC Clear Clear the TC bit in the CHSTAT_n register to 0. This bit is always read as 0. 0: Operation is not affected 1: Clear the TC bit in the CHSTAT_n register	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
8	SETSUS	Suspend Request If 1 is set to this bit when the EN bit in the CHSTAT_n register is 1, the current DMA transfer is suspended. This bit is always read as 0. 0: Operation is not affected 1: Suspends the current DMA transfer	R/W

Bit	Symbol	Function	R/W
9	CLRSUS	Suspend Clear If 1 is set to this bit when the SUS bit in the CHSTAT_n register is 1, the temporary suspend status is cleared. This bit is always read as 0. 0: Operation is not affected 1: Clear the temporary suspend status of the current DMA transfer	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	SETINTM	DMA Transfer Completion Interrupt Request Mask Mask DMA transfer completion interrupt output temporarily. Additionally, the INTM bit in the CHSTATn register is set to 1. This bit is always read as 0. 0: Operation is not affected 1: Mask a DMA transfer completion interrupt temporarily	R/W
17	CLRINTM	DMA Transfer Completion Interrupt Request Mask Clear Clear the mask status of DMA transfer completion interrupt output. Additionally, the INTM bit in the CHSTATn register is cleared to 0. If the mask state is cleared when the LVINT bit in the DCTRL register = 1 and the END bit in the CHSTAT_n register = 1, DMA transfer completion interrupt is deactivated. It is not activated when LVINT = 0. This bit is always read as 0. 0: Operation is not affected 1: Clear the mask state set with the SETINTM bit	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The CHCTRL_n register controls DMA transfer of DMA channel n (n = 0 to 15).

14.2.13 CHCFG_n : Channel Configuration Register n (n = 0 to 15)

Base address: DMACi = 0x8008_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x02C + 0x40 × n (n = 0 to 7)
0x42C + 0x40 × (n - 8) (n = 8 to 15)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DMS	REN	RSW	RSEL	SBE	—	TCM	DEM	—	TM	DAD	SAD	DDS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SDS[3:0]				—	AM[2:0]			—	LVL	HIEN	LOEN	REQD	SEL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SEL[2:0]	Pin Select Set channels of DMAC. Set the following values so that the channels of CHCFG_n (n = 0 to 15) and the channel set by SEL become the same. For example, set 001b to the SEL bits in CHCFG_1. 0 0 0: Channel 0/8 0 0 1: Channel 1/9 0 1 0: Channel 2/10 0 1 1: Channel 3/11 1 0 0: Channel 4/12 1 0 1: Channel 5/13 1 1 0: Channel 6/14 1 1 1: Channel 7/15	R/W

Bit	Symbol	Function	R/W
3	REQD	DMA Activation Request Source Select Specifies the DMA transfer request input source (internal modules or external devices that require DACK). The timing when DACK becomes active is determined by the setting of this bit. 0: Requested by a transfer source module. DACK output is activated when it is read (a value after a reset). 1: Requested by a transfer destination module. DACK output is activated when it is written.	R/W
4	LOEN	L Detection Enable Specifies the detection method of the DMA request signal. When LVL = 0 LOEN = 1: If DMA transfer request input detects a falling edge, it is regarded as requested. LOEN = 0: If the DMA transfer request input falls, the request is not recognized (a value after a reset). When LVL = 1 LOEN = 1: If DMA transfer request input detects a low level, it is regarded as requested. LOEN = 0: If the DMA transfer request input is a low level, the request is not recognized (a value after reset).	R/W
5	HIEN	H Detection Enable Specifies the detection method of the DMA request signal. When LVL = 0 HIEN = 1: If DMA transfer request input detects a rising edge, it is regarded as requested. HIEN = 0: If the DMA transfer request input rises, the request is not recognized (a value after reset). When LVL = 1 HIEN = 1: If DMA transfer request input detects a high level, it is regarded as requested. HIEN = 0: If the DMA transfer request input is a high level, the request is not recognized (a value after reset).	R/W
6	LVL	Level Detection Enable Specifies the detection method of the DMA request signal. 0: Detects the edge (a value after reset) 1: Detects the level	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	AM[2:0]	ACK Mode These bits specify the DACK output mode. For the conditions of outputting the DACK and TEND signals, see section 14.3.5. DMA Acknowledge Output/DMA Transaction Completion Output Function. 0 0 0: Setting is prohibited (a value after reset). At the initial setup, set a value other than 000b. 0 0 1: Level mode (active until DMA transfer request input becomes inactive) 0 1 x: Bus cycle mode (active between DMA transfer cycles (active if internal resources are the target)) 1 x x: Mask DACK output	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
15:12	SDS[3:0]	Source Data Size Set the size of data in the transfer source to be transferred at a time. 0x0: 8 bits (a value after reset) 0x1: 16 bits 0x2: 32 bits 0x3: 64 bits 0x4: 128 bits*1 0x5: 256 bits*2 0x6: 512 bits (can be set only for DMAC0)*1 Others: Setting prohibited	R/W

Bit	Symbol	Function	R/W
19:16	DDS[3:0]	Destination Data Size Set the size of data in the transfer destination to be transferred at a time. 0x0: 8 bits (a value after reset) 0x1: 16 bits 0x2: 32 bits 0x3: 64 bits 0x4: 128 bits*1 0x5: 256 bits*2 0x6: 512 bits (can be set only for DMAC0)*1 Others: Setting prohibited	R/W
20	SAD	Source Address Count Direction Set the count direction of the transfer source address of DMA channel n. When the transfer source address is beat-unaligned, specify SAD = 0 (increment). When the transfer source address is aligned with a boundary of the size specified by the SDS[3:0] bits, SAD can be set to 0 or 1. 0: Increment (a value after reset) 1: Fixed	R/W
21	DAD	Destination Address Count Direction Set the count direction of the transfer destination address of DMA channel n. When the transfer destination address is beat-unaligned, specify DAD = 0 (increment). When the transfer destination address is aligned with a boundary of the size specified by the DDS[3:0] bits, DAD can be set to 0 or 1. 0: Increment (a value after reset) 1: Fixed	R/W
22	TM	Transfer Mode Set DMA transfer mode. 0: Single transfer mode (a value after reset) 1: Block transfer mode	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
24	DEM	DMA Transfer Completion Interrupt Mask Mask DMA transfer completion interrupt detection. If this bit is set to 1 when DMA transfer completion interrupt is output, DMA transfer completion interrupt is not activated. At this time, the DEM bit is automatically cleared to 0. [Clearing condition] ● When DEM = 1 and DMA transfer completes 0: Does not mask (a value after reset) 1: Mask	R/W
25	TCM	TEND Mask Mask the TEND output. If this bit is 1 at the TEND output timing, TEND is not asserted. At this time, the TCM bit is automatically cleared to 0. Use this bit to control DMA transfers by software. [Clearing condition] ● When TCM = 1 and DMA transfer completes 0: Does not mask (a value after reset) 1: Mask	R/W
26	—	This bit is read as 0. The write value should be 0.	R/W
27	SBE	Buffer Flush Enable If the EN bit in the CHSTAT_n register is cleared to 0 during DMA transfer, select whether to stop flushing (writing) data that is already read and stored in the buffer. Only when REQD = 0, flush mode can be used. 0: Stop transfer without flushing data in the buffer (a value after reset) 1: Stop transfer after flushing data in the buffer	R/W
28	RSEL	Next Register Select Select the Next register set to be executed next. This bit is valid in register mode only. When RSW = 1, the value is reversed automatically (0 is reversed to 1, 1 is reversed to 0) after DMA transfer completes. [Transition condition] ● When RSW = 1 and DMA transfer completes 0: Execute Next0 Register Set (a value after reset) 1: Execute Next1 Register Set	R/W

Bit	Symbol	Function	R/W
29	RSW	RSEL Reverse When DMA transfer completes, the RSEL bit is automatically reversed (0 is reversed to 1, 1 is reversed to 0). This bit is valid in register mode only. 0: Does not reverse the RSEL bit when DMA transfer completes (a value after reset) 1: Reverse the RSEL bit when DMA transfer completes	R/W
30	REN	Register Set Enable When DMA transfer completes, performs DMA transfer of the Next register set selected by the RSEL bit accordingly. This bit is valid in register mode only. [Setting condition] ● Writing 1 to this bit [Clearing conditions] When any of the following conditions is met: ● Writing 0 to this bit ● When REN = 1 and DMA transfer completes 0: Does not perform DMA transfer accordingly 1: Performs DMA transfer accordingly	R/W
31	DMS	DMA Mode Select Set DMA mode. 0: Register mode (a value after reset). 1: Link mode	R/W

Note 1. The SCA[3:0] bits in the CHEXT_n register should be set to 0x2 if source is TCM. The DCA[3:0] bits in the CHEXT_n register should be set to 0x2 if destination is TCM.
Note 2. This setting is prohibited when DMAC accesses TCM.

The CHCFG_n register controls DMA transfer of DMA channel n (n = 0 to 15).

Specify the detection method depending on the DMA transfer source to be used. For details on the DMA request signals, see section 14.3.4.1. Specifying Detection Operation of DMA Transfer Requests for Each Source.

14.2.14 CHITVL_n : Channel Interval Register n (n = 0 to 15)

Base address: DMACi = 0x8008_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x030 + 0x40 × n (n = 0 to 7)
0x430 + 0x40 × (n - 8) (n = 8 to 15)

Bit position:	31															15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ITVL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ITVL[15:0]	Interval These bits set the DMA transfer internal.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The CHITVL_n register sets the DMA transfer interval of DMA channel n (n = 0 to 15).

For details, see section 14.3.7. Interval Count Function.

14.2.15 NXLA_n : Next Link Address Register n (n = 0 to 15)

Base address: DMACi = 0x8008_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x038 + 0x40 × n (n = 0 to 7)
0x438 + 0x40 × (n - 8) (n = 8 to 15)

Bit position:	31																													0
Bit field:																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

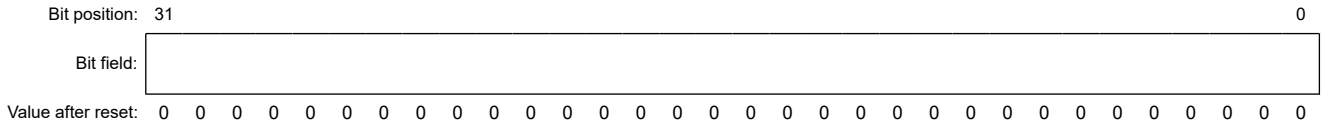
Bit	Symbol	Function	R/W
31:0	n/a	Next Link Address Sets the address of the link destination. Because the 2 lower-order bits are fixed to 0, only word-aligned addresses can be set.	R/W

The NXLA_n register sets the link address of DMA channel n (n = 0 to 15).

14.2.16 CRLA_n : Current Link Address Register n (n = 0 to 15)

Base address: DMACi = 0x8008_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x03C + 0x40 × n (n = 0 to 7)
0x43C + 0x40 × (n - 8) (n = 8 to 15)



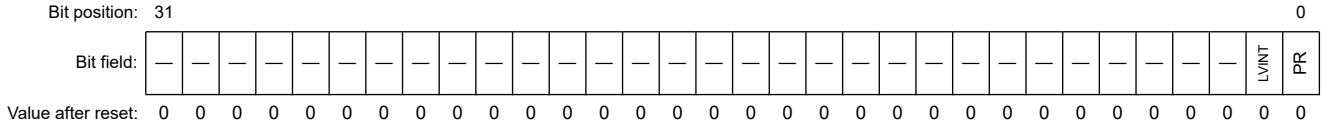
Bit	Symbol	Function	R/W
31:0	n/a	Current Link Address Displays the address of the descriptor that is being executed.	R

The CRLA_n register sets the link address of DMA channel n (n = 0 to 15).

14.2.17 DCTRL_x : DMA Control Register x (x = A, B)

Base address: DMACi = 0x8008_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x300 (DCTRL_A)
0x700 (DCTRL_B)



Bit	Symbol	Function	R/W
0	PR	Priority Control Select Sets the transfer priority control mode (see section 14.3.3. DMA Channel Priority Control. 0: Fixed priority mode 1: Round-robin mode	R/W
1	LVINT	DMA Interrupt Output Select Sets the interrupt output mode. This bit must be set to 1 when the DMA interrupt is enabled. 0: Pulse output 1: Level output	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The DCTRL_x register sets the arbitration between channels in all channels (DCTRL_A = channels 0 to 7, DCTRL_B = channels 8 to 15).

14.2.18 DSTAT_EN_x : DMA Status EN Register x (x = A, B)

Base address: DMACi = 0x8008_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x310 (DSTAT_EN_A)
0x710 (DSTAT_EN_B)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	EN0715	EN0614	EN0513	EN0412	EN0311	EN0210	EN0109	EN0008
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EN0008	Channel 0/8 EN Displays the status for the EN bit of DMA channel 0/8.	R
1	EN0109	Channel 1/9 EN Displays the status for the EN bit of DMA channel 1/9.	R
2	EN0210	Channel 2/10 EN Displays the status for the EN bit of DMA channel 2/10.	R
3	EN0311	Channel 3/11 EN Displays the status for the EN bit of DMA channel 3/11.	R
4	EN0412	Channel 4/12 EN Displays the status for the EN bit of DMA channel 4/12.	R
5	EN0513	Channel 5/13 EN Displays the status for the EN bit of DMA channel 5/13.	R
6	EN0614	Channel 6/14 EN Displays the status for the EN bit of DMA channel 6/14.	R
7	EN0715	Channel 7/15 EN Displays the status for the EN bit of DMA channel 7/15.	R
31:8	—	These bits are read as 0.	R

The DSTAT_EN_x register displays the status for the EN bit of all channels (DSTAT_EN_A = channels 0 to 7, DSTAT_EN_B = channels 8 to 15).

Writing data to this register does not change the value of each bit.

14.2.19 DSTAT_ER_x : DMA Status ER Register x (x = A, B)

Base address: DMACi = 0x8008_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x314 (DSTAT_ER_A)
0x714 (DSTAT_ER_B)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	ER0715	ER0614	ER0513	ER0412	ER0311	ER0210	ER0109	ER0008
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER0008	Channel 0/8 ER Displays the status for the ER bit of DMA channel 0/8.	R
1	ER0109	Channel 1/9 ER Displays the status for the ER bit of DMA channel 1/9.	R
2	ER0210	Channel 2/10 ER Displays the status for the ER bit of DMA channel 2/10.	R
3	ER0311	Channel 3/11 ER Displays the status for the ER bit of DMA channel 3/11.	R
4	ER0412	Channel 4/12 ER Displays the status for the ER bit of DMA channel 4/12.	R
5	ER0513	Channel 5/13 ER Displays the status for the ER bit of DMA channel 5/13.	R
6	ER0614	Channel 6/14 ER Displays the status for the ER bit of DMA channel 6/14.	R
7	ER0715	Channel 7/15 ER Displays the status for the ER bit of DMA channel 7/15.	R
31:8	—	These bits are read as 0.	R

The DSTAT_ER_x register displays the status for the ER bit of all channels (DSTAT_ER_A = channels 0 to 7, DSTAT_ER_B = channels 8 to 15).

Writing data to this register does not change the value of each bit.

14.2.20 DSTAT_END_x : DMA Status END Register x (x = A, B)

Base address: DMACi = 0x8008_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x318 (DSTAT_END_A)
0x718 (DSTAT_END_B)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	END0 715	END0 614	END0 513	END0 412	END0 311	END0 210	END0 109	END0 008
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	END0008	Channel 0/8 END Displays the status for the END bit of DMA channel 0/8.	R
1	END0109	Channel 1/9 END Displays the status for the END bit of DMA channel 1/9.	R
2	END0210	Channel 2/10 END Displays the status for the END bit of DMA channel 2/10.	R
3	END0311	Channel 3/11 END Displays the status for the END bit of DMA channel 3/11.	R
4	END0412	Channel 4/12 END Displays the status for the END bit of DMA channel 4/12.	R
5	END0513	Channel 5/13 END Displays the status for the END bit of DMA channel 5/13.	R
6	END0614	Channel 6/14 END Displays the status for the END bit of DMA channel 6/14.	R

Bit	Symbol	Function	R/W
7	END0715	Channel 7/15 END Displays the status for the END bit of DMA channel 7/15.	R
31:8	—	These bits are read as 0.	R

The DSTAT_END_x register displays the status for the END bit of all channels (DSTAT_END_A = channels 0 to 7, DSTAT_END_B = channels 8 to 15).

Writing data to this register does not change the value of each bit.

14.2.21 DSTAT_SUS_x : DMA Status SUS Register x (x = A, B)

Base address: DMACi = 0x8008_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x320 (DSTAT_SUS_A)
0x720 (DSTAT_SUS_B)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SUS07 15	SUS06 14	SUS05 13	SUS04 12	SUS03 11	SUS02 10	SUS01 09	SUS00 08
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SUS0008	Channel 0/8 SUS Displays the status for the SUS bit of DMA channel 0/8.	R
1	SUS0109	Channel 1/9 SUS Displays the status for the SUS bit of DMA channel 1/9.	R
2	SUS0210	Channel 2/10 SUS Displays the status for the SUS bit of DMA channel 2/10.	R
3	SUS0311	Channel 3/11 SUS Displays the status for the SUS bit of DMA channel 3/11.	R
4	SUS0412	Channel 4/12 SUS Displays the status for the SUS bit of DMA channel 4/12.	R
5	SUS0513	Channel 5/13 SUS Displays the status for the SUS bit of DMA channel 5/13.	R
6	SUS0614	Channel 6/14 SUS Displays the status for the SUS bit of DMA channel 6/14.	R
7	SUS0715	Channel 7/15 SUS Displays the status for the SUS bit of DMA channel 7/15.	R
31:8	—	These bits are read as 0.	R

The DSTAT_SUS_x register displays the status for the SUS bit of all channels (DSTAT_SUS_A = channels 0 to 7, DSTAT_SUS_B = channels 8 to 15).

Writing data to this register does not change the value of each bit.

14.3 Operation

14.3.1 DMA Mode

With the DMS bit in the CHCFG_n register, DMA mode can be switched between register mode and link mode.

Table 14.3 DMA mode setting

DMS (CHCFG_n)	Description	Applications
0	Register mode	With the values set for the next register set, performs a DMA transfer.
1	Link mode	Accesses the descriptor area, and performs a DMA transfer with the value set for the descriptor. Repeats descriptor read and DMA transfer unless you set the descriptor or use the control register to stop them.

14.3.1.1 Register Mode

In register mode, you can perform a DMA transfer with the value set in the internal register. You can set two sets (Next0 Register Set and Next1 Register Set) of the transfer source addresses, transfer destination addresses, and numbers of transfer bytes. You can select the next register set to perform a transfer, or transfer two next register sets successively.

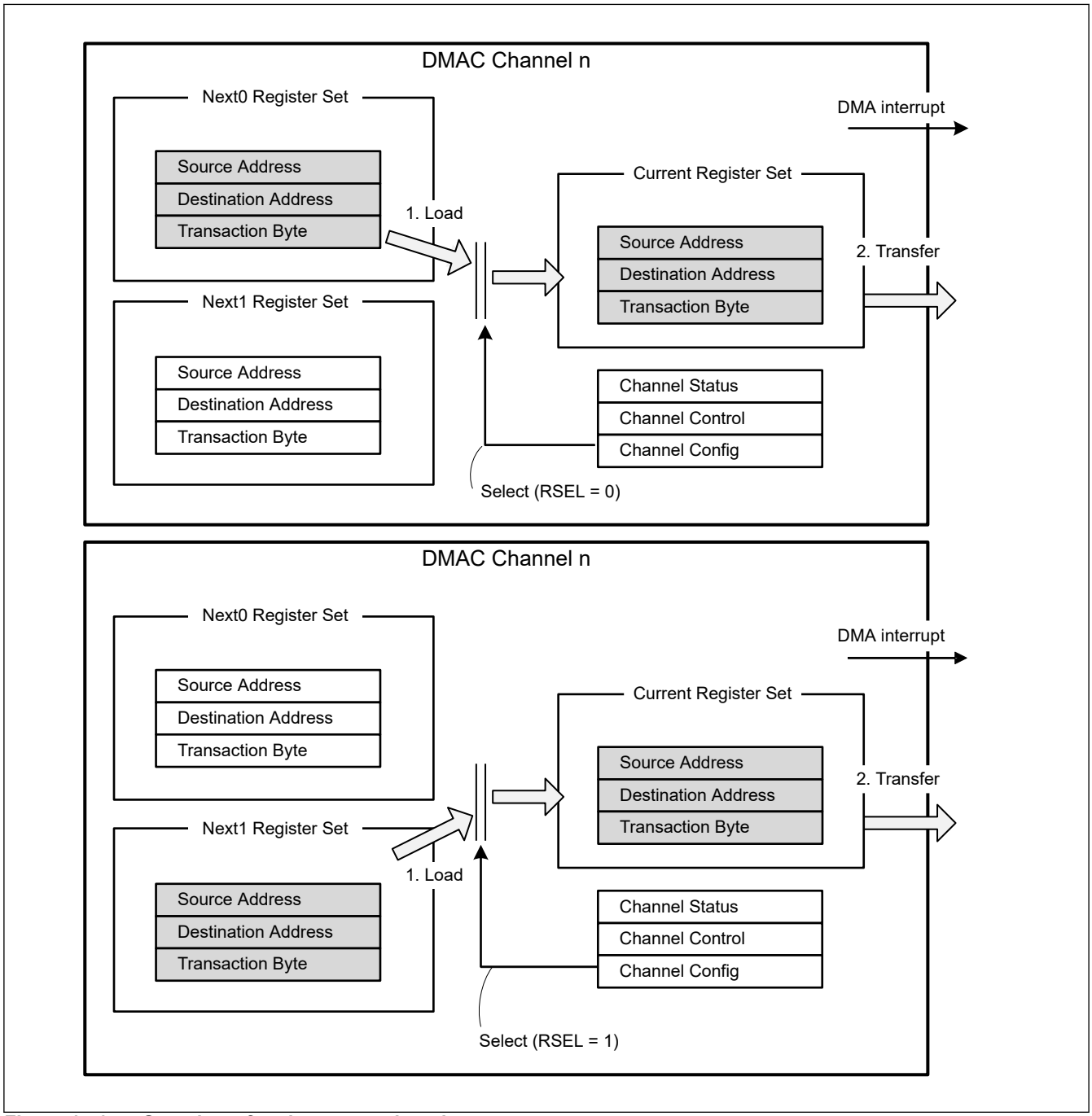


Figure 14.1 Overview of register normal mode

Figure 14.1 shows operations when Next0 Register Set is executed above and when Next1 Register Set is executed below.

(1) Operation Flow in Register Mode

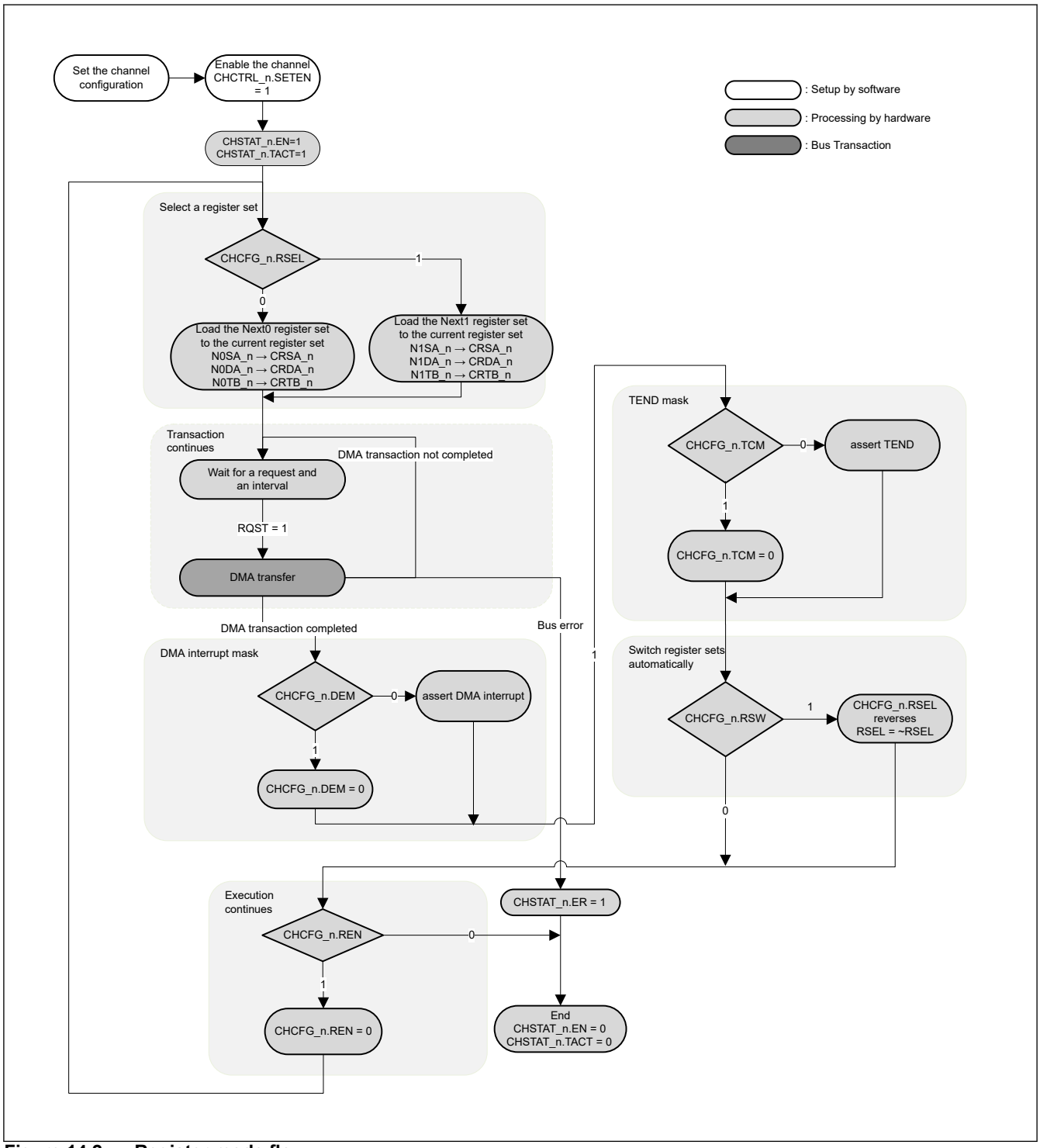


Figure 14.2 Register mode flow

(2) Register Mode Flow Description

1. Channel setting
   Set Next0 or Next1 Register Set (the transfer destination address, the transfer source address, and the total number of transfer bytes).
   In addition, use setting registers CHCTRL_n or CHCFG_n for each channel to set the detection method of the DMA transfer request, the output mode of the DACK/TEND signal, and the amount of data for a transfer.

2. Selecting the register set
   When 1 is written to the SETEN bit in the CHCTRL_n register, the EN bit and the TACT bit in the CHSTAT_n register are set to 1, and the setting value of the next register set selected by the RSEL bit in the CHCFG_n register is loaded to the current register set.
3. DMA transfer
   According to the set value, a DMA transfer is performed. For details on the transfer, see from section 14.3.2. Transfer Mode to section 14.3.11. Aborting a Transfer.
4. DMA transfer completion mask
   According to the value set for the DEM bit in the CHCFG_n register, the DMA transfer completion interrupt is masked. When DEM = 1, the DMA transfer completion interrupt is masked. Immediately after the DMA transfer completion interrupt conditions are satisfied, the DEM bit is cleared to 0 automatically.
5. TEND mask
   According to the values set for the TCM bit in the CHCFG_n register, the TEND is masked. When TCM = 1, TEND is masked. Immediately, the DMA transfer completes and the TCM bit is cleared to 0 automatically.
6. Switching register sets automatically
   According to the value set for the RSW bit in the CHCFG_n register, the register set is switched to the other next register set.
7. Successive execution
   According to the value set for the REN bit in the CHCFG_n register, DMA transfers are performed successively. When REN = 0, the EN bit and the TACT bit in the CHSTAT_n register are cleared to 0, and DMAC stops operation. When REN = 1, the DMA transfer continues. Immediately after the conditions for performing DMA transfer again are satisfied with the REN bit, the REN bit is cleared to 0 automatically.

(3) Setting the Register Mode

Select the register set to execute.

Table 14.4 Register mode settings

DMS (CHCFG_n)	RSEL (CHCFG_n)	Description
0	0	Executes Next0 Register Set
	1	Executes Next1 Register Set

DMA transfer completion interrupts can be masked.

Table 14.5 DMA transfer completion interrupt mask settings

DEM (CHCFG_n)	Description
0	When a DMA transfer completes, the DMA transfer completion interrupt is generated
1	Even if a DMA transfer completes, no DMA transfer completion interrupt is generated. When a DMA transfer completes, the DEM bit is cleared to 0 automatically.

TEND output can be masked.

Table 14.6 TEND output mask settings

TCM (CHCFG_n)	Description
0	When a DMA transfer completes, the TEND is asserted
1	Even if a DMA transfer completes, no TEND is asserted. When a DMA transfer completes, the TCM bit is cleared to 0 automatically.

After a DMA transfer, another DMA transfer can be performed successively.

Table 14.7 Settings for automatic execution of register sets

REN (CHCFG_n)	Operation	Remarks
0	When a DMA transfer of the register set for the RSEL bit completes, the EN bit is cleared to 0, and the DMA operation ends	Set this value if you want to perform a DMA transfer once
1	After a DMA transfer completes, a DMA transfer is performed for the register set that was selected to be performed successively. When the successive transfer is performed, the REN bit is cleared to 0.	Set this register if you want to perform register sets successively

When a DMA transfer completes, the register set can be switched to the next register set.

Table 14.8 Settings for switching register sets automatically

RSW (CHCFG_n)	Operation	Remarks
0	When a DMA transfer completes, register sets are not switched	Set this value when you want to use one register set only
1	When a DMA transfer completes, the RSEL bit is reversed automatically, and the other register set is selected	Set this value if you want to switch register sets

(4) Register Mode Setting Examples

Using the Next0 register set only.

Table 14.9 Register mode setting example 1

DMS (CHCFG_n)	RSEL (CHCFG_n)	DEM (CHCFG_n)	TCM (CHCFG_n)	RSW (CHCFG_n)	REN (CHCFG_n)
0 (Register mode)	0 (Next0)	0 (Do not mask)	0 (Do not mask)	0 (Do not switch)	0 (Continuous execution not in progress)

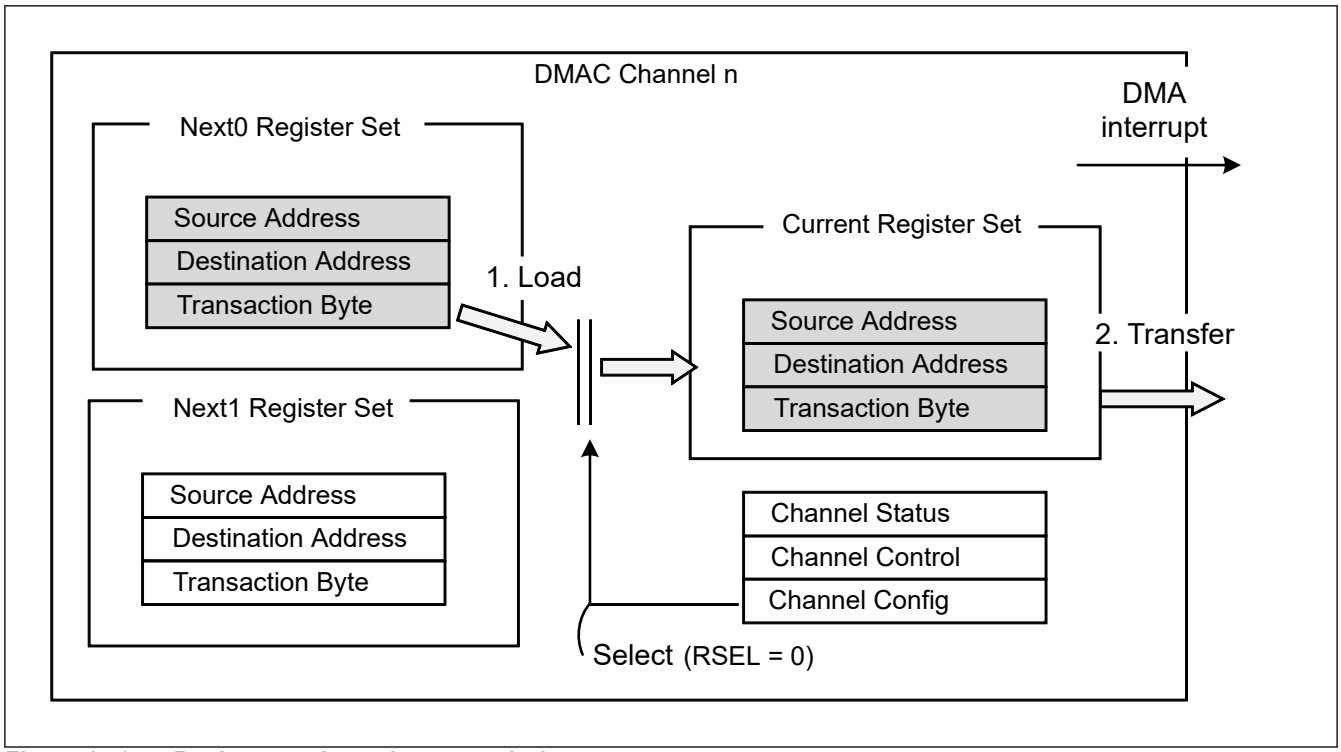


Figure 14.3 Register mode setting example 1

1. Writing 1 to the SETEN bit in the CHCTRL_n register sets 1 to the EN bit in the CHSTAT_n register, and loads Next0 Register Set to the current register set.
2. According to the values set for the current register set and the channel register set, a DMA transfer is performed.

3. Because the DEM bit in the CHCFG_n register is 0, a DMA transfer completion interrupt is generated when a DMA transfer completes.
4. Because the TCM bit in the CHCFG_n register is 0, the TEND is asserted when DMA transfer completes.
5. Because the REN bit in the CHCFG_n register is 0, the EN bit in the CHSTAT_n register is cleared to 0, and the operation ends.

Using two register sets successively.

Table 14.10 Register mode setting example 2

DMS (CHCFG_n)	RSEL (CHCFG_n)	DEM (CHCFG_n)	TCM (CHCFG_n)	RSW (CHCFG_n)	REN (CHCFG_n)
0 (Register mode)	0 (Next0)	1 (Mask)	0 (Do not mask)	1 (Switch)	1 (Continuous execution not in progress)

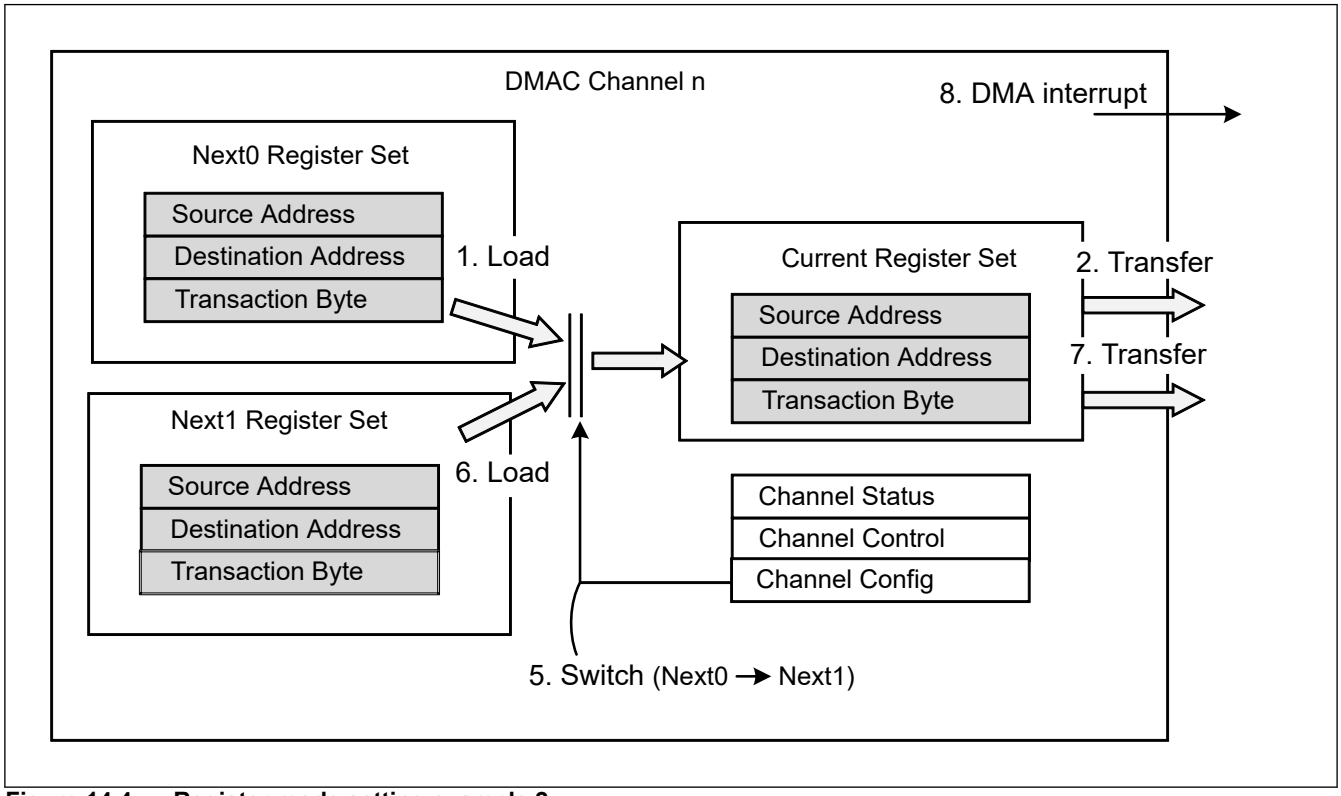


Figure 14.4 Register mode setting example 2

1. Writing 1 to the SETEN bit in the CHCTRL_n register sets 1 to the EN bit in the CHSTAT_n register, and loads Next0 Register Set to the current register set.
2. According to the values set for the current register set and the channel register set, a DMA transfer is performed.
3. Because the DEM bit in the CHCFG_n register is 1, when the DMA transfer completes, no DMA transfer completion interrupt is generated. In addition, the DEM bit is cleared to 0 automatically.
4. Because the REN bit in the CHCFG_n register is 1, DMA transfers are performed successively. In addition, the REN bit is cleared to 0.
5. Because the RSW bit in the CHCFG_n register is 1, the register set to be executed next is switched (RSEL = 0 → 1).
6. Loads Next1 Register Set to Current Register Set.
7. According to the values for Current Register Set and Channel Register Set, DMA transfers are performed.
8. Because the DEM bit in the CHCFG_n register is 0, a DMA transfer completion interrupt is generated when a DMA transfer completes.
9. Because the TCM bit in the CHCFG_n register is 0, the TEND is asserted when DMA transfer completes.

10. Because the REN bit in the CHCFG_n register is 0, the EN bit in the CHSTAT_n register is cleared to 0 automatically.

14.3.1.2 Link Mode

In link mode, a DMA transfer is performed by reading a descriptor in the memory area outside the DMAC as the setting value. Within DMAC, each channel has the next link address (NXLA_n) register and the current link address (CRLA_n) register. Each of them is used to set the address of the descriptor to be executed next and to display the descriptor address of the current DMA transfer respectively.

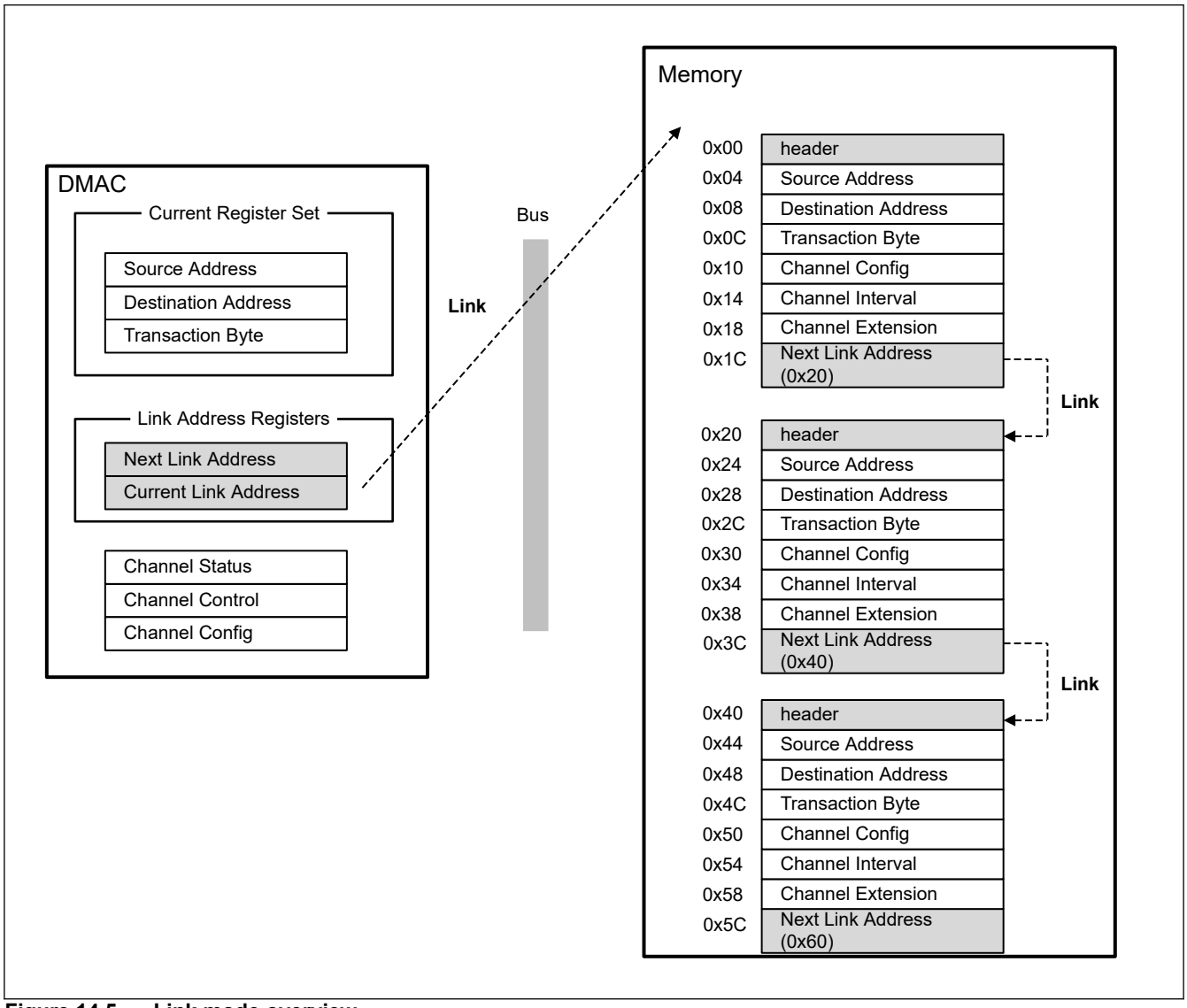


Figure 14.5 Link mode overview

(1) Operation Flows in Link Mode

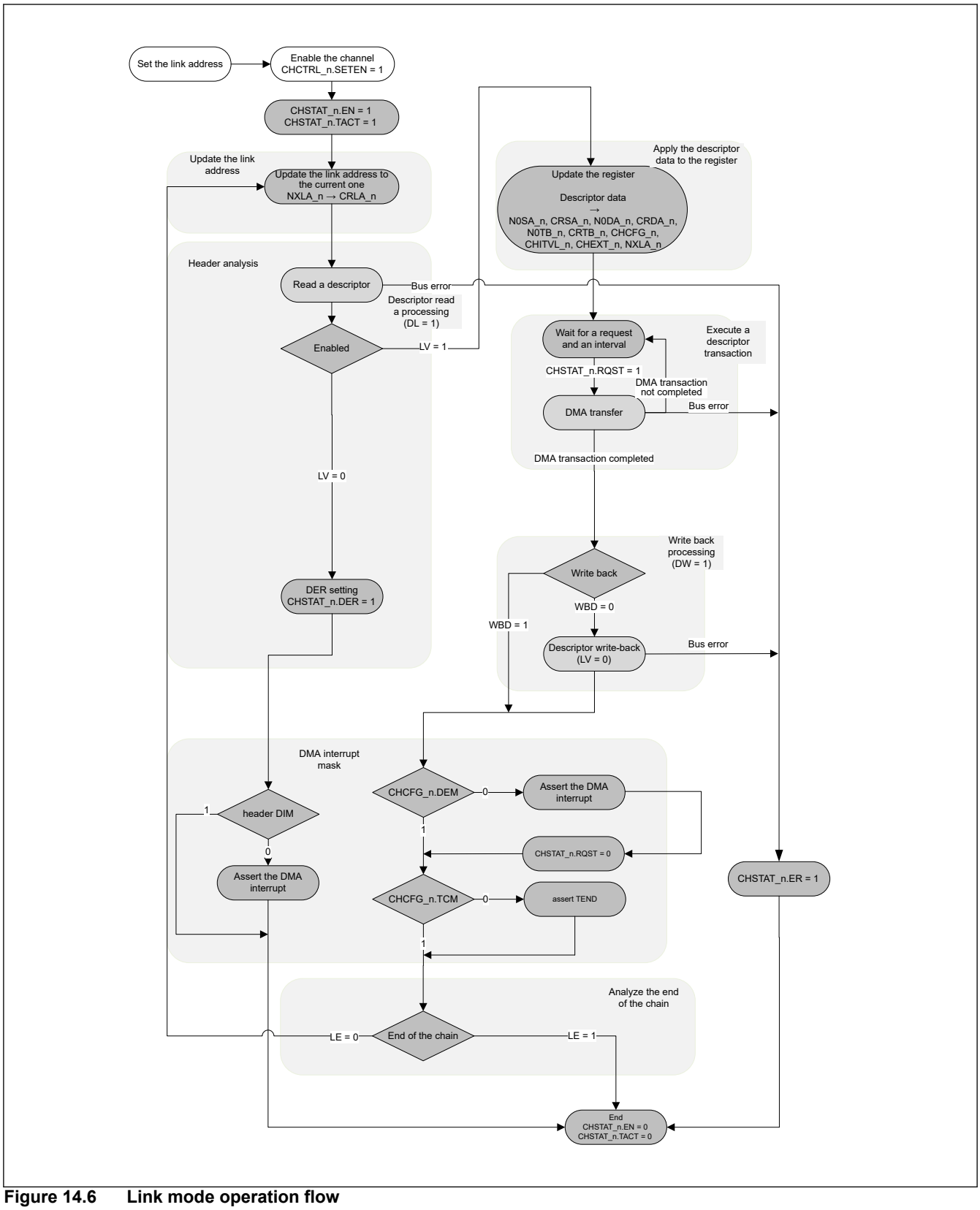


Figure 14.6 Link mode operation flow

(2) Description of Link Mode Operation Flow

1. Channel setting
   Set the start address of the link destination in the NXLA_n register.

2. Updating the link address
   Writing 1 to the SETEN bit in the CHCTRL_n register sets 1 to the EN bit and the TACT bit in the CHSTAT_n register, and loads the link address set for the NXLA_n register to the CRLA_n register.
3. Descriptor read and header judgment
   DMAC starts reading a descriptor and checks the header contents. When LV = 0, the descriptor is discarded and the DER bit in the CHSTAT_n register = 1 indicates the end state (the EN bit in the CHSTAT_n register = 0). At that time, if the DIM bit in the header register is 0, a DMA transfer completion interrupt is generated.
4. Descriptor settings
   When LV = 1, the read descriptor data is loaded to the current register set and the channel register set. In addition, the next link destination is loaded to the NXLA_n register.
5. DMA transfer
   According to the set value, a DMA transfer is performed. For details on the transfer, see from section 14.3.2. Transfer Mode to section 14.3.11. Aborting a Transfer.
6. Writing back of header
   When WBD of header = 0, DMAC writes LV = 0 back to the header area.
7. DMA interrupt mask
   When the DEM bit in the CHCFG_n register is 0, a DMA transfer completion interrupt is generated. When DMA transfer is performed using multi descriptors, DMA request should be issued again and RQST is set to 1 to continue the DMA transfer because RQST is set to 0 when DEM = 0.
8. TEND mask
   When the TCM bit in the CHCGF_n register is 0, the TEND is generated.
9. Link end judgment
   When LE of header = 1, the EN bit and the TACT bit in the CHSTAT_n register are cleared to 0, and DMAC stops operation. When LE = 0, the current register set is updated, and the next descriptor read is started.

(3) Register Settings

When using link mode, set the DMS bit in the CHCFG_n register to 1.

Table 14.11 Link mode settings

DMS (CHCFG_n)	Description
1	Operates in link mode. This bit cannot be changed by using a descriptor.

As registers that indicate the link destination, the next link address (NXLA_n) register and the current link address (CRLA_n) register are available.

To start link mode, set the link destination in the NXLA_n register.

The NXLA_n register is updated to the next link after a descriptor is read. In addition, the CRLA_n register indicates the link address of the currently executed descriptor.

Table 14.12 Link address register set

Register	Description
NXLA_n	Sets and displays the next link destination. Before starting link mode, set the address of the link destination for this register.
CRLA_n	Displays the currently executed link destination. This register is read-only.

(4) Descriptor Settings

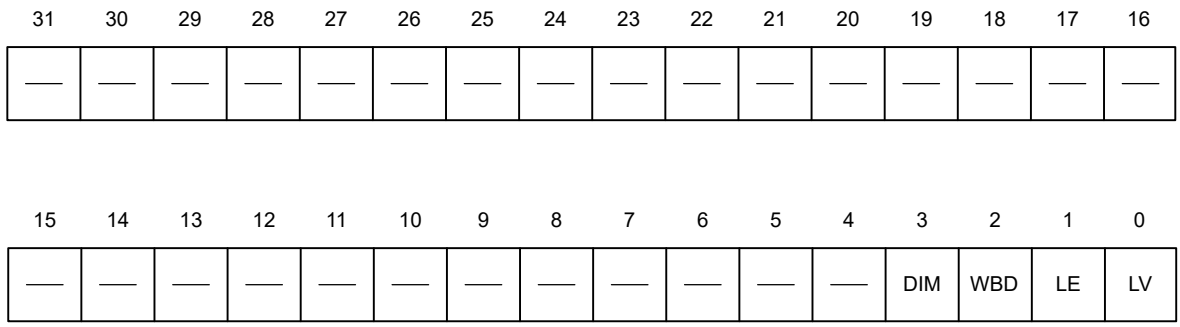
A descriptor should be deployed in the link address as shown in Table 14.13. DMAC interprets data obtained through descriptor read in order.

Table 14.13 Descriptor placement

Address (Link Address + N)*1							
+0x1C	**+0x18**	**+0x14**	**+0x10**	**+0x0C**	**+0x08**	**+0x04**	**+0x00**
Next link address	Extension	Interval	Config*2	Transaction byte	Destination address	Source address	Header

Note 1. Link address should be aligned to 32-bit boundary.
Note 2. Register mode setting is prohibited.

As shown in Figure 14.7, header indicates descriptor states. This area is read by DMAC before starting a DMA transfer in link mode. In addition, after the DMA transfer, the transfer status is written back by DMAC.



When reading header, DMA performs a write back for this 4 bytes area after DMA transfer and when the descriptor information is completed. Write back data is a header read data except LV, and LV = 0.

Figure 14.7 Header area

Table 14.14 Header area

Bit position	Bit name	Description
0	LV	Link Valid Indicates that this descriptor is enabled. When WBD = 0, DMAC writes 0 after the DMA transfer written in the descriptor. Set 1 when setting header. 0: The descriptor is disabled 1: The descriptor is enabled
1	LE	Link End Indicates that the link ends during DMA transfer of this descriptor. To indicate the end of the link, set this bit to 1. 0: The link continues 1: The link ends
2	WBD	Write Back Disable Masks write back execution of the LV bit. When this bit is 1, DMAC does not perform write-back operation. 0: The LV bit is written back to 0 1: The LV bit is not written back
3	DIM	Descriptor Interrupt Mask When LV = 0 at loading of header, specifies whether DMA transfer completion interrupt mask or not. 0: DMA transfer completion interrupt is asserted 1: DMA transfer completion interrupt is masked
31:4	—	A reserved area. Set 0.

Data in descriptors except header has the same specifications as an on-chip register. However, the DMS bit in the CHCFG_n register cannot be changed by a descriptor. For details on the on-chip register specifications, see section 14.2. Register Descriptions.

For descriptor setting examples, see section 14.5.4. Setting Example 4 (Link Mode).

Figure 14.8 provides an overview of the descriptor area and the DMA transfer area to which DMAC accesses.

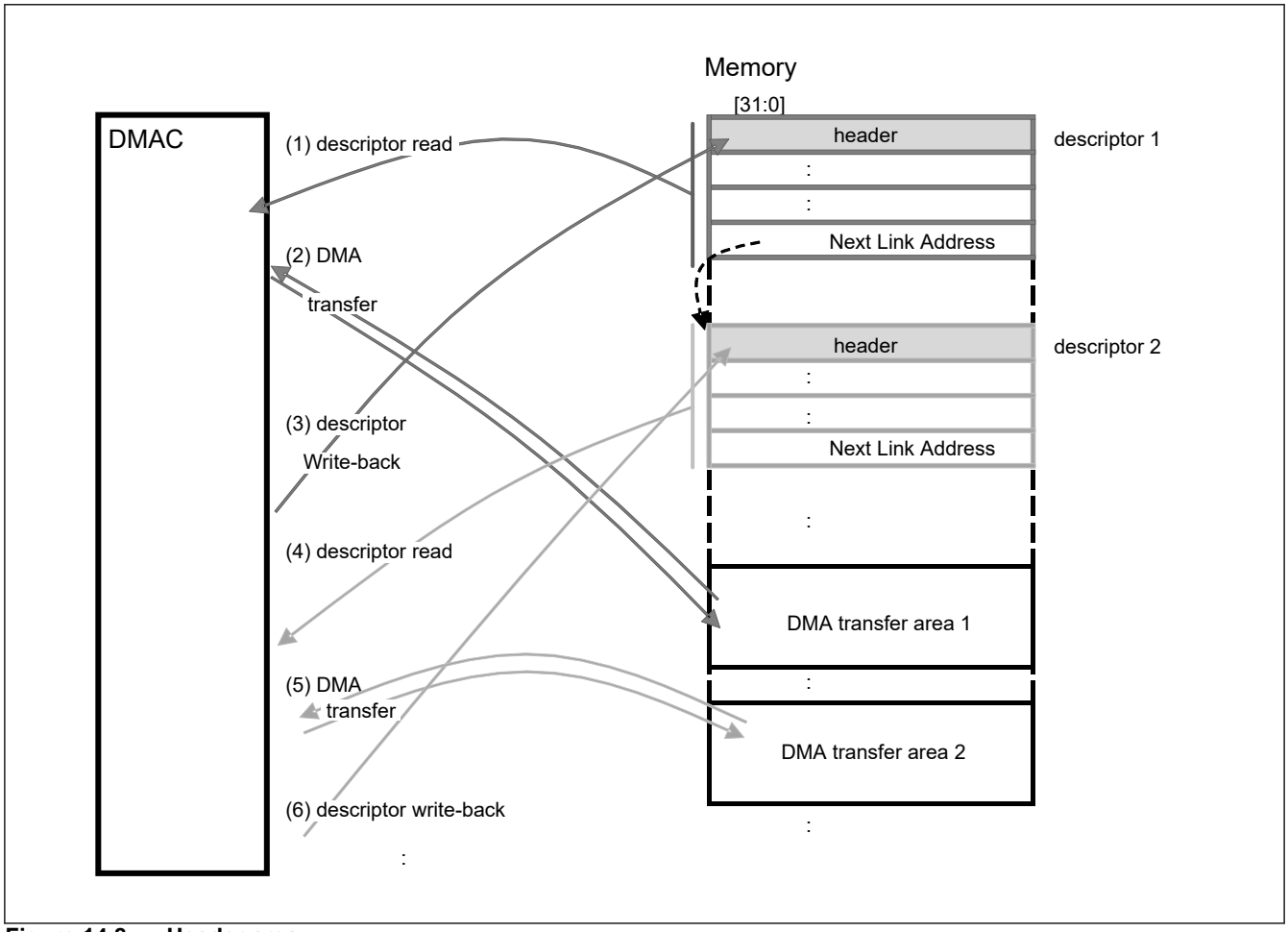


Figure 14.8 Header area

1. Descriptor read
   Loads the value set for the on-chip NXLA_n register to the CRLA_n register, and then reads the descriptor from the memory space (descriptor 1) indicated by the CRLA_n register.
2. DMA transfer
   When the LV bit in header is 1, performs a DMA transfer according to the descriptor information.
3. Descriptor write-back
   After the DMA transfer for the set number of bytes, if the WBD bit in header is 0, the LV bit writes 0 and other fields writes the value read in step 1 as data back to header [31:1] of descriptor 1 in bytes.
4. Descriptor read
   If the LE bit in header of the descriptor which was read previously (step 1) is 0, reads the next descriptor from the address (descriptor 2) indicated by the next link address in the descriptor.
5. DMA transfer
   When the LV bit in header is 1, performs a DMA transfer according to the descriptor information.
6. Descriptor write-back
   After the DMA transfer for the set number of bytes, if the WBD bit in header is 0, the LV bit writes 0 and other fields writes the value read in step 4 as data back to header [31:1] of descriptor 2 in bytes.

Hereafter, repeats steps (4) through (6).

When LE of header is 1 and WBD = 0, DMA transfer proceeds in accord with the descriptor settings, and 0 is written back to the LV bit of the header, after which operation ends.

When LE of header is 1 and WBD = 1, performs a DMA transfer with the setting, and ends the operation (no write-back is performed).

When LV of header is 0 the operation stops (no DMA transfer is performed).

(5) Link Configuration Examples

In link mode, descriptors can be configured as shown in Figure 14.9.

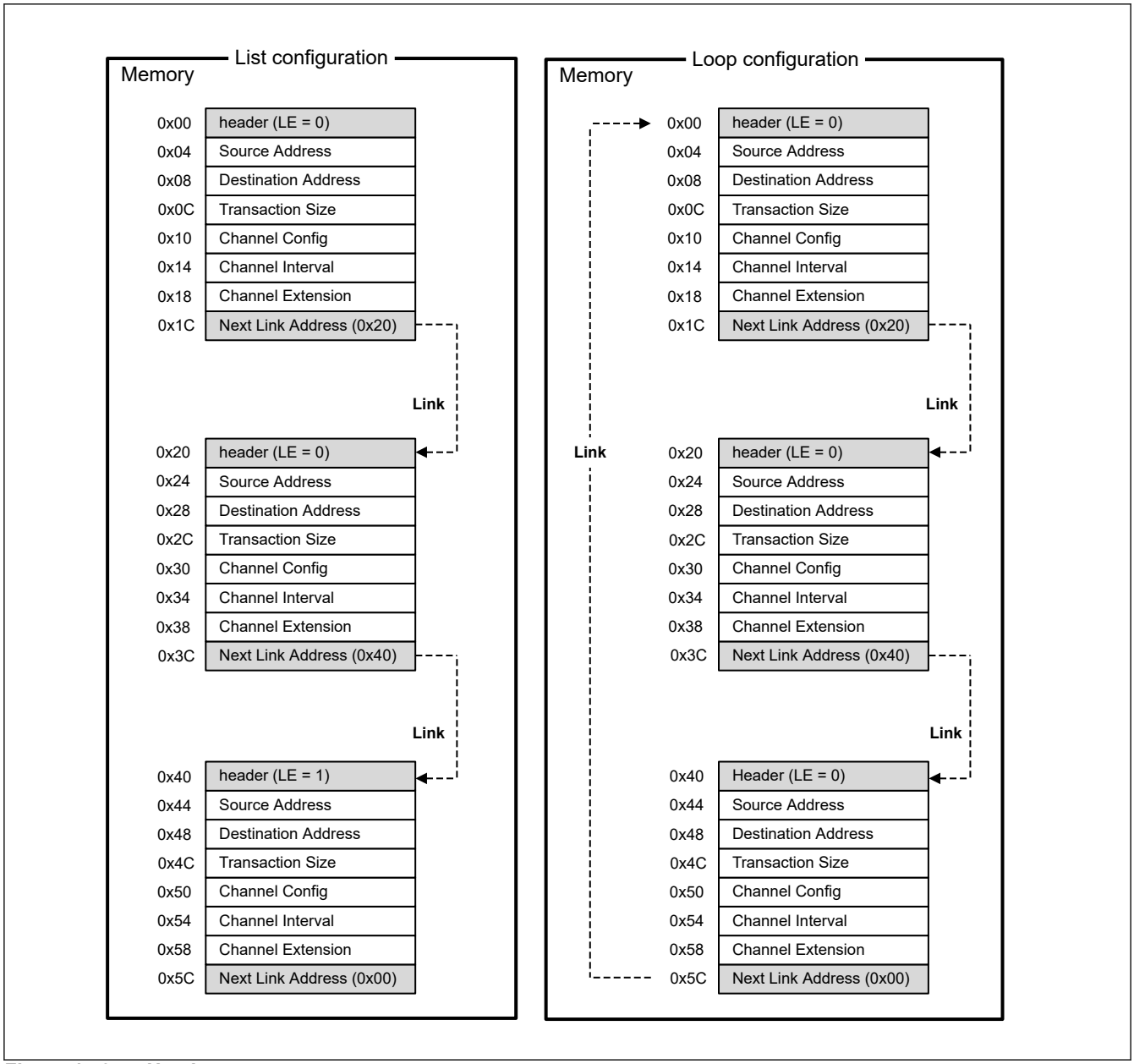


Figure 14.9 Header area

Setting 1 to the LE bit in header of the last descriptor ends the link.

Setting the link destination of the last descriptor to the address of the previous descriptor configures the descriptors in a loop. To end the loop, change the LE bit of header to 1 before DMAC reads a descriptor, or follow the procedure to pause a transfer.

14.3.2 Transfer Mode

Single transfer mode and block transfer mode are supported. When selecting mode, use the TM bit in the CHCFG_n register for each channel.

Table 14.15 Basic transfer settings

Transfer mode	TM (CHCFG_n)	Description
Single transfer	0	For one DMA request, performs a DMA transfer
Block transfer	1	For one DMA request, performs transfers until a DMA transfer completes

14.3.2.1 Single Transfer Mode

When a DMA transfer request is accepted, a DMA transfer is performed on the side (transfer source or transfer destination) indicated by the REQD bit in the CHCFG_n register to activate DACK. Each time a transfer request is accepted, transfer proceeds. This operation is repeated until it reaches the transfer size loaded from the N0TB_n or N1TB_n register to the CRTB_n register. Arbitration between channels is performed for each DMA transfer.

The DACK output timing differs depending on the setting for the REQD bit in the CHCFG_n register or settings for the transfer size (DDS[2:0] and SDS[2:0] in the CHCFG_n register) bit settings. For details, see section 14.3.8. Differences in Operation According to the Transfer Data Size.

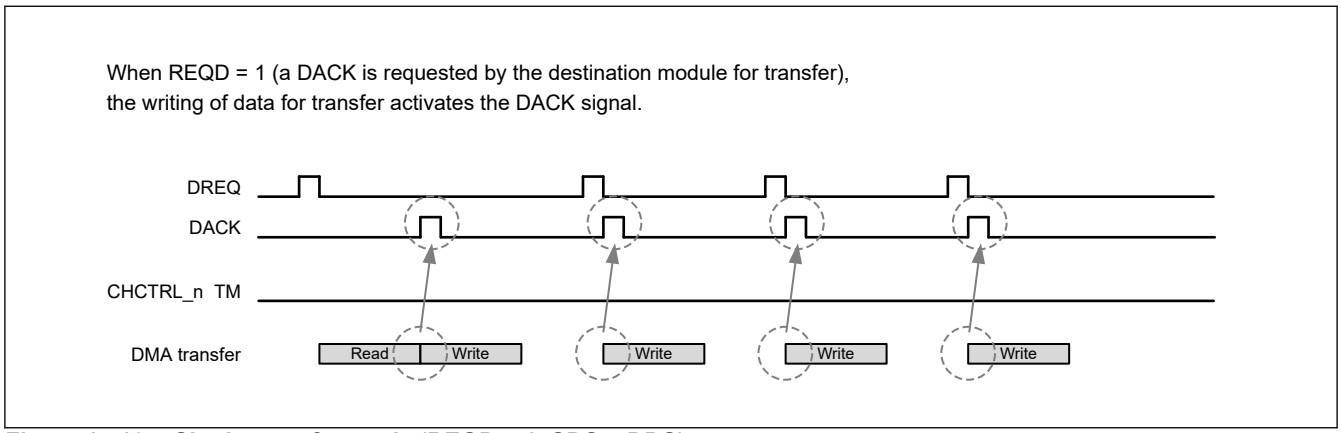


Figure 14.10 Single transfer mode (REQD = 1, SDS > DDS)

14.3.2.2 Block Transfer Mode

Once a DMA transfer request is accepted, the transfer continues until transfer of the number of bytes loaded from the N0TB_n or N1TB_n register to the DMA transfer byte register (CRTB_n register) is completed (After DMA transfer completion, arbitration between channels is performed for each DMA transfer).

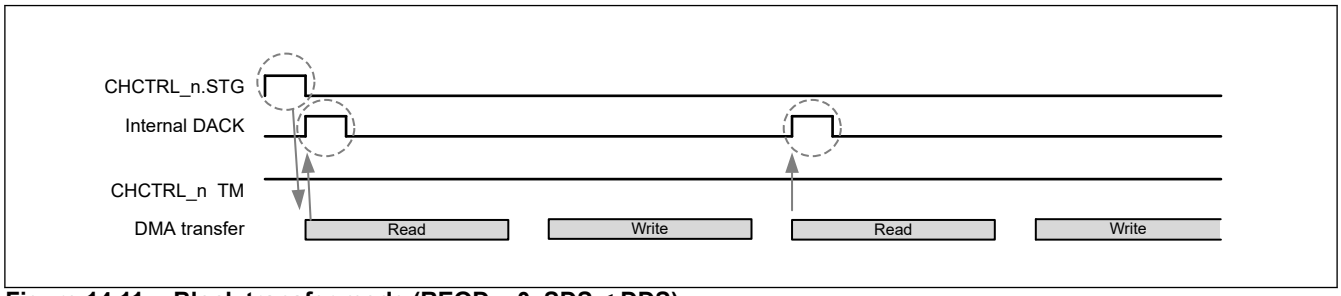


Figure 14.11 Block transfer mode (REQD = 0, SDS < DDS)

14.3.3 DMA Channel Priority Control

As an arbitration method between channels, fixed priority mode and round-robin mode are supported. To select the mode, use the PR bit in the DCTRL register. When the PR bit is 0, fixed priority mode is selected. When the PR bit is set to 1, round-robin mode is selected.

Table 14.16 Priority control settings

Transfer mode	PR (DCTRL)	Description	Applications
Fixed Priority	0	Channels 0 to 7, channels 8 to 15 are fixed priority mode. CH0 (CH8) > CH1 (CH9) > CH2 (CH10) > CH3 (CH11) > CH4 (CH12) > CH5 (CH13) > CH6 (CH14) > CH7 (CH15)	Select this mode if channels have priority
Round-Robin	1	Controls requests in round-robin mode	Select this mode if you want to execute requests equally

14.3.3.1 Fixed Priority Mode

In fixed priority mode, the priority levels are fixed within the group of channels 0 to 7 and within the group of channels 8 to 15. In addition, the priority levels between the group of channels 0 to 7 and the group of channels 8 to 15 are in round-robin mode.

The priority levels immediately after a reset and transfer through DMA channel 0 are shown in Figure 14.12.

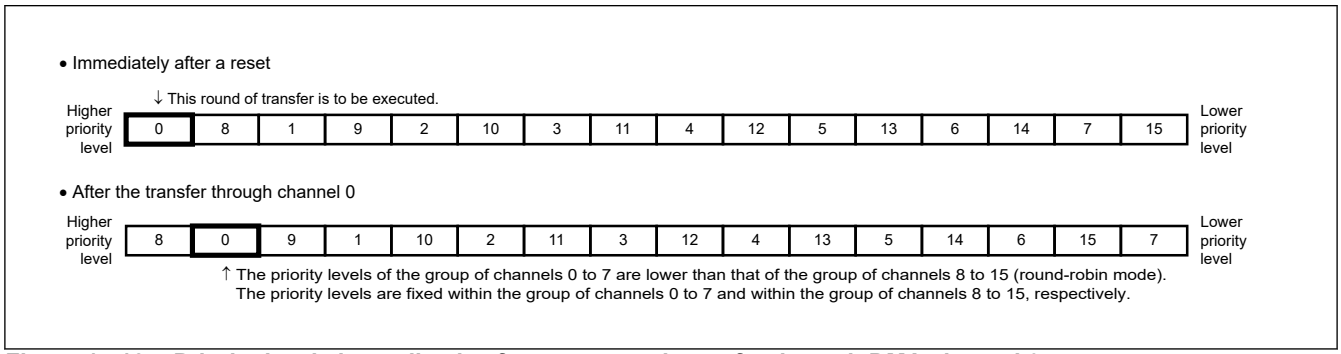
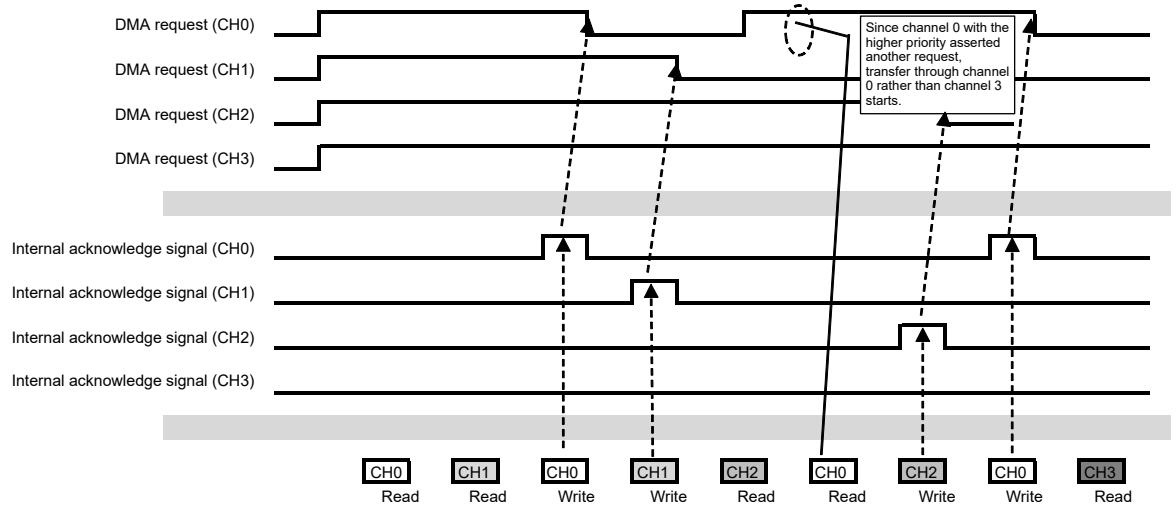


Figure 14.12 Priority levels immediately after a reset and transfer through DMA channel 0

If DMA transfer requests are generated on multiple channels simultaneously, a DMA transfer request with a smaller channel number takes precedence.

Figure 14.13 provides an example of when a DMA transfer is performed in fixed priority mode if another DMA transfer with higher priority is requested.

Example: DMA requests are issued through four channels at the same time.



Note: Channel 0 is handled with the highest priority, but the transfer with the next highest priority is performed because bus arbitration is performed after data has been read from the transfer source of channel 0. Read operation of another channel may interrupt the operation between reading data from and writing data to the same channel.

Figure 14.13 Fixed priority mode (number of channels = 4 and REQD = 1)

14.3.3.2 Round-Robin Mode

In round-robin mode, priority is changed every time a transfer of a channel is accepted so that the lowest priority is given to the channel in which the last transfer is performed.

The priority levels immediately after a reset and transfer through DMA channel 2 are shown in Figure 14.14.

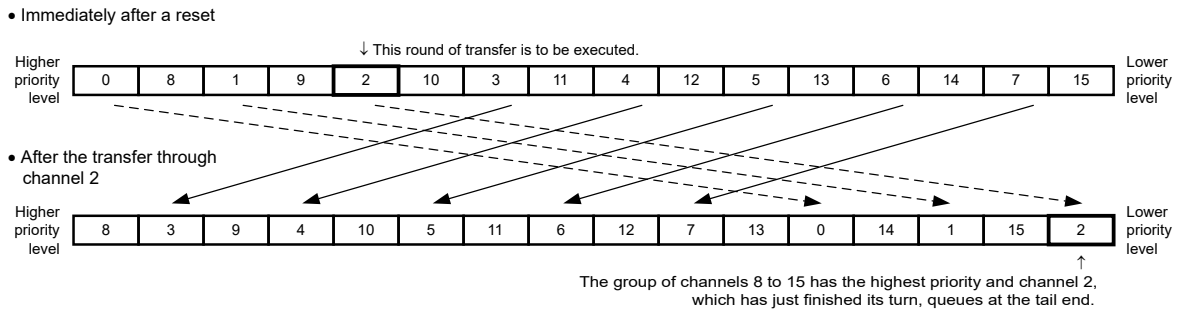
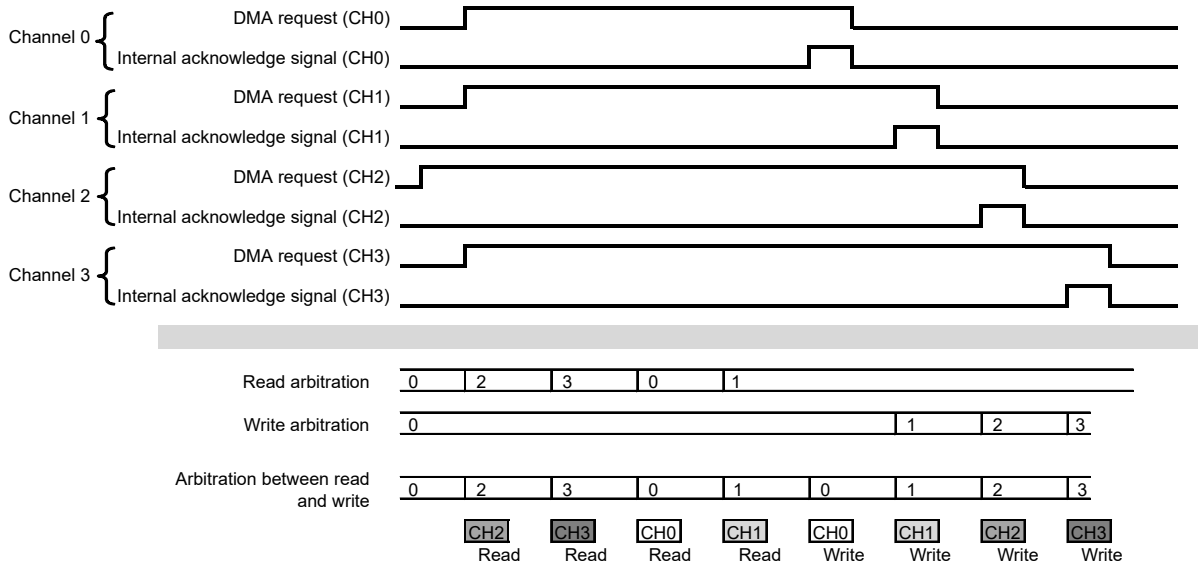


Figure 14.14 Priority levels immediately after a reset and transfer through DMA channel 2

Figure 14.15 provides an example of a DMA transfer in round-robin mode.



Note: Within DMAC, arbitration between read channels and arbitration between write channels are performed. As a result of those arbitrations, further arbitration is performed, and a bus access is issued.

Figure 14.15 Round-robin mode (number of channels = 4 and REQD = 1)

14.3.4 DMA Transfer Request

DMA activation requests have four types including software requests, external requests, on-chip peripheral module requests, and external interrupts.

Select the transfer request source for an external request, on-chip peripheral module request, external interrupt, and software request with the DMACm_RSSELn register (m = 0 or 1, n = 0 to 5).

For details on the DMACm_RSSELn register (m = 0 or 1, n = 0 to 5), see section 14.2.9. DMAC0_RSSELi : DMAC Unit 0 Resource Select Register i (i = 0 to 5) and section 14.2.10. DMAC1_RSSELi : DMAC Unit 1 Resource Select Register i (i = 0 to 5).

14.3.4.1 Specifying Detection Operation of DMA Transfer Requests for Each Source

Detection methods for DMA transfer requests of external requests, on-chip peripheral module requests, external interrupts, and software requests might be specified according to sources.

For each DMA transfer source, set the LVL, LEN, and HEN bits in the CHCFG_n register according to Table 14.17, Table 14.18, and Table 14.19.

For details on edge detection operation, see Figure 14.16 and Figure 14.17. For information about level detection operation, see Figure 14.18 and Figure Figure 14.19.

Table 14.17 Detection operation specification for each source of DMA transfer requests

DMA transfer request source	Detection operation specification of DMA transfer requests	DMA acknowledge signal specification
External request	Detects the rising edge. Detects the falling edge. Detects the high level. Detects the low level.	Depends on specifications of the DMA transfer request source
On-chip peripheral module request	Depends on specifications of the DMA transfer request source. See Table 14.19.	Bus cycle mode
External interrupt	Detects the rising edge. Detects the high level.	Bus cycle mode
Software request	Detects the rising edge	Bus cycle mode

Table 14.18 Method of detecting DMA transfer request signals

Mode	LVL (CHCFG_n)	HIEN (CHCFG_n)	LOEN (CHCFG_n)	Description
Edge detection	0	0	0	Disables detection
			1	Detects the falling edge
		1	0	Detects the rising edge
			1	Detects the falling and rising edges
Level detection	1	0	0	Disables detection
			1	Detects the low level
		1	0	Detects the high level
			1	DMA transfer starts regardless of the request signal level when the SETEN bit in the CHCTRL_n is 1. In this setting, DMA acknowledge must be set to other than level mode. If level mode is set, dead lock will occur.

Table 14.19 DMA transfer request detection operation setting table (1 of 9)

DMA transfer request source	DMA transfer source	Transfer source	Transfer destination	REQ_SEL n[8:0]	CHCFG_n						
					TM	AM[2:0]*2	LVL	HIEN	LOEN	REQD*2	SEL[2:0]
External DMA Request	DREQ	Arbitrary	Arbitrary	0x1C0	0/1	001/ 010/ 100	0/1	10/01		0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
Software Interrupt	Software interrupt 0	Arbitrary	Arbitrary	0x000	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	Software interrupt 1			0x001							
	Software interrupt 2			0x002							
	Software interrupt 3			0x003							
	Software interrupt 4			0x004							
	Software interrupt 5			0x005							
External Interrupt	IRQ0	Arbitrary	Arbitrary	0x006	0/1	010	0/1*1	1*1	0*1	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	IRQ1			0x007							
	IRQ2			0x008							
	IRQ3			0x009							
	IRQ4			0x00A							
	IRQ5			0x00B							
	IRQ6			0x00C							
	IRQ7			0x00D							
	IRQ8			0x00E							
	IRQ9			0x00F							
	IRQ10			0x010							
	IRQ11			0x011							
	IRQ12			0x012							
	IRQ13			0x013							

Table 14.19 DMA transfer request detection operation setting table (2 of 9)

DMA transfer request source	DMA transfer source	Transfer source	Transfer destination	REQ_SELn[8:0]	CHCFG_n						
					TM	AM[2:0]*2	LVL	HIEN	LOEN	REQD*2	SEL[2:0]
CMT Unit 0	Compare match 0	Arbitrary	Arbitrary	0x035	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	Compare match 1			0x036							
CMT Unit 1	Compare match 0	Arbitrary	Arbitrary	0x037	0/1	010	0	1	0	0/1	
	Compare match 1			0x038							
CMT Unit 2	Compare match 0	Arbitrary	Arbitrary	0x039	0/1	010	0	1	0	0/1	
	Compare match 1			0x03A							
CMTW Unit 0	Compare match	Arbitrary	Arbitrary	0x03B	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	Input capture 0			0x03C							
	Input capture 1			0x03D							
	Output compare 0			0x03E							
	Output compare 1			0x03F							
CMTW Unit 1	Compare match	Arbitrary	Arbitrary	0x040	0/1	010	0	1	0	0/1	
	Input capture 0			0x041							
	Input capture 1			0x042							
	Output compare 0			0x043							
	Output compare 1			0x044							

Table 14.19 DMA transfer request detection operation setting table (3 of 9)

DMA transfer request source	DMA transfer source	Transfer source	Transfer destination	REQ_SELn[8:0]	CHCFG_n						
					TM	AM[2:0]*2	LVL	HIEN	LOEN	REQD*2	SEL[2:0]
MTU3	TGIA0	Arbitrary	Arbitrary	0x045	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	TGIB0			0x046							
	TGIC0			0x047							
	TGID0			0x048							
	TGIA1			0x04C							
	TGIB1			0x04D							
	TGIA2			0x050							
	TGIB2			0x051							
	TGIA3			0x054							
	TGIB3			0x055							
	TGIC3			0x056							
	TGID3			0x057							
	TGIA4			0x059							
	TGIB4			0x05A							
	TGIC4			0x05B							
	TGID4			0x05C							
	TCIV4			0x05D							
	TGIU5			0x05E							
	TGIV5			0x05F							
	TGIW5			0x060							
	TGIA6			0x061							
	TGIB6			0x062							
	TGIC6			0x063							
	TGID6*3			0x064							
	TGIA7			0x066							
	TGIB7			0x067							
	TGIC7			0x068							
	TGID7			0x069							
	TCIV7			0x06A							
	TGIA8			0x06B							
	TGIB8			0x06C							
	TGIC8			0x06D							
	TGID8			0x06E							
GPT Unit 0	CCMPA	Arbitrary	Arbitrary	0x074	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	CCMPB			0x075							
	CMPC			0x076							
	CMPD			0x077							
	CMPE			0x078							
	CPMF			0x079							
	OVF			0x07A							
	UDF			0x07B							

Table 14.19 DMA transfer request detection operation setting table (4 of 9)

DMA transfer request source	DMA transfer source	Transfer source	Transfer destination	REQ_SELn[8:0]	CHCFG_n						
					TM	AM[2:0]*2	LVL	HIEN	LOEN	REQD*2	SEL[2:0]
GPT Unit 1	CCMPA	Arbitrary	Arbitrary	0x07D	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	CCMPB			0x07E							
	CMPC			0x07F							
	CMPD			0x080							
	CMPE			0x081							
	CPMF			0x082							
	OVF			0x083							
	UDF			0x084							
GPT Unit 2	CCMPA	Arbitrary	Arbitrary	0x086	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	CCMPB			0x087							
	CMPC			0x088							
	CMPD			0x089							
	CMPE			0x08A							
	CPMF			0x08B							
	OVF			0x08C							
	UDF			0x08D							
GPT Unit 3	CCMPA	Arbitrary	Arbitrary	0x08F	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	CCMPB			0x090							
	CMPC			0x091							
	CMPD			0x092							
	CMPE			0x093							
	CPMF			0x094							
	OVF			0x095							
	UDF			0x096							
GPT Unit 4	CCMPA	Arbitrary	Arbitrary	0x098	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	CCMPB			0x099							
	CMPC			0x09A							
	CMPD			0x09B							
	CMPE			0x09C							
	CPMF			0x09D							
	OVF			0x09E							
	UDF			0x09F							
GPT Unit 5	CCMPA	Arbitrary	Arbitrary	0x0A1	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	CCMPB			0x0A2							
	CMPC			0x0A3							
	CMPD			0x0A4							
	CMPE			0x0A5							
	CPMF			0x0A6							
	OVF			0x0A7							
	UDF			0x0A8							

Table 14.19 DMA transfer request detection operation setting table (5 of 9)

DMA transfer request source	DMA transfer source	Transfer source	Transfer destination	REQ_SELn[8:0]	CHCFG_n						
					TM	AM[2:0]*2	LVL	HIEN	LOEN	REQD*2	SEL[2:0]
GPT Unit 6	CCMPA	Arbitrary	Arbitrary	0x0AA	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	CCMPB			0x0AB							
	CMPC			0x0AC							
	CMPD			0x0AD							
	CMPE			0x0AE							
	CPMF			0x0AF							
	OVF			0x0B0							
	UDF			0x0B1							
GPT Unit 7	CCMPA	Arbitrary	Arbitrary	0x0B3	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	CCMPB			0x0B4							
	CMPC			0x0B5							
	CMPD			0x0B6							
	CMPE			0x0B7							
	CPMF			0x0B8							
	OVF			0x0B9							
	UDF			0x0BA							
GPT Unit 8	CCMPA	Arbitrary	Arbitrary	0x0BC	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	CCMPB			0x0BD							
	CMPC			0x0BE							
	CMPD			0x0BF							
	CMPE			0x0C0							
	CPMF			0x0C1							
	OVF			0x0C2							
	UDF			0x0C3							
GPT Unit 9	CCMPA	Arbitrary	Arbitrary	0x0C5	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	CCMPB			0x0C6							
	CMPC			0x0C7							
	CMPD			0x0C8							
	CMPE			0x0C9							
	CPMF			0x0CA							
	OVF			0x0CB							
	UDF			0x0CC							
GPT Unit 10	CCMPA	Arbitrary	Arbitrary	0x0CE	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	CCMPB			0x0CF							
	CMPC			0x0D0							
	CMPD			0x0D1							
	CMPE			0x0D2							
	CPMF			0x0D3							
	OVF			0x0D4							
	UDF			0x0D5							

Table 14.19 DMA transfer request detection operation setting table (6 of 9)

DMA transfer request source	DMA transfer source	Transfer source	Transfer destination	REQ_SELn[8:0]	CHCFG_n						
					TM	AM[2:0]*2	LVL	HIEN	LOEN	REQD*2	SEL[2:0]
GPT Unit 11	CCMPA	Arbitrary	Arbitrary	0x0D7	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	CCMPB			0x0D8							
	CMPC			0x0D9							
	CMPD			0x0DA							
	CMPE			0x0DB							
	CPMF			0x0DC							
	OVF			0x0DD							
	UDF			0x0DE							
GPT Unit 12	CCMPA	Arbitrary	Arbitrary	0x0E0	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	CCMPB			0x0E1							
	CMPC			0x0E2							
	CMPD			0x0E3							
	CMPE			0x0E4							
	CPMF			0x0E5							
	OVF			0x0E6							
	UDF			0x0E7							
GPT Unit 13	CCMPA	Arbitrary	Arbitrary	0x0E9	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	CCMPB			0x0EA							
	CMPC			0x0EB							
	CMPD			0x0EC							
	CMPE			0x0ED							
	CPMF			0x0EE							
	OVF			0x0EF							
	UDF			0x0F0							
GMAC	SBD	Arbitrary	Arbitrary	0x0FC	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
ENDAT Unit 0	Interrupt output 1	Arbitrary	Arbitrary	0x0FF	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
ENDAT Unit 1	Interrupt output 1	Arbitrary	Arbitrary	0x102	0/1	010	0	1	0	0/1	
BISS Unit 0	End of transmission interrupt	Arbitrary	Arbitrary	0x105	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
BISS Unit 1	End of transmission interrupt	Arbitrary	Arbitrary	0x106	0/1	010	0	1	0	0/1	

Table 14.19 DMA transfer request detection operation setting table (7 of 9)

DMA transfer request source	DMA transfer source	Transfer source	Transfer destination	REQ_SELn[8:0]	CHCFG_n						
					TM	AM[2:0]*2	LVL	HIEN	LOEN	REQD*2	SEL[2:0]
HDSL Unit 0	Interrupt output	Arbitrary	Arbitrary	0x107	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	Interrupt output (safety related)			0x108							
HDSL Unit 1	Interrupt output	Arbitrary	Arbitrary	0x109	0/1	010	0	1	0	0/1	
	Interrupt output (safety related)			0x10A							
AFMT Unit 0	Communication completion	Arbitrary	Arbitrary	0x10B	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
AFMT Unit 1	Communication completion	Arbitrary	Arbitrary	0x10C	0/1	010	0	1	0	0/1	
GMAC	PTP timer pulse output 0	Arbitrary	Arbitrary	0x10D	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	PTP timer pulse output 1			0x10E							
	PTP timer pulse output 2			0x10F							
	PTP timer pulse output 3			0x110							
HDSL Unit 0	Fast position value ready interrupt	Arbitrary	Arbitrary	0x111	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
HDSL Unit 1	Fast position value ready interrupt	Arbitrary	Arbitrary	0x112	0/1	010	0	1	0	0/1	
HDSL Unit 0	Safe position received interrupt	Arbitrary	Arbitrary	0x113	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
HDSL Unit 1	Safe position received interrupt	Arbitrary	Arbitrary	0x114	0/1	010	0	1	0	0/1	
EtherCAT slave	Sync0 interrupt	Arbitrary	Arbitrary	0x115	0/1	010	0	0/1	0/1	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	Sync1 interrupt			0x116							
	EtherCAT interrupt	Arbitrary	Arbitrary	0x117	0/1	010	0	1	0	0/1	
	SOF interrupt			0x118							
	EOF interrupt			0x119							
USB	Host interrupt	Arbitrary	Arbitrary	0x11C	0/1	010	0/1	1	0	0/1	
	Function interrupt			0x11D							

Table 14.19 DMA transfer request detection operation setting table (8 of 9)

DMA transfer request source	DMA transfer source	Transfer source	Transfer destination	REQ_SELn[8:0]	CHCFG_n						
					TM	AM[2:0]*2	LVL	HIEN	LOEN	REQD*2	SEL[2:0]
SCI Unit 0	Receive data full	RDR	Arbitrary	0x121	0	010	0	1	0	0	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	Transmit data empty	Arbitrary	TDR	0x122	0	010	0	1	0	1	
SCI Unit 1	Receive data full	RDR	Arbitrary	0x125	0	010	0	1	0	0	
	Transmit data empty	Arbitrary	TDR	0x126	0	010	0	1	0	1	
SCI Unit 2	Receive data full	RDR	Arbitrary	0x129	0	010	0	1	0	0	
	Transmit data empty	Arbitrary	TDR	0x12A	0	010	0	1	0	1	
SCI Unit 3	Receive data full	RDR	Arbitrary	0x12D	0	010	0	1	0	0	
	Transmit data empty	Arbitrary	TDR	0x12E	0	010	0	1	0	1	
SCI Unit 4	Receive data full	RDR	Arbitrary	0x131	0	010	0	1	0	0	
	Transmit data empty	Arbitrary	TDR	0x132	0	010	0	1	0	1	
IIC Unit 0	Receive data full	ICDRR	Arbitrary	0x135	0	010	0	1	0	0	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	Transmit data empty	Arbitrary	ICDRT	0x136	0	010	0	1	0	1	
IIC Unit 1	Receive data full	ICDRR	Arbitrary	0x139	0	010	0	1	0	0	
	Transmit data empty	Arbitrary	ICDRT	0x13A	0	010	0	1	0	1	
SPI Unit 0	Receive data full	SPDR	Arbitrary	0x144	0	010	0	1	0	0	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	Transmit data empty	Arbitrary	SPDR	0x145	0	010	0	1	0	1	
SPI Unit 1	Receive data full	SPDR	Arbitrary	0x149	0	010	0	1	0	0	
	Transmit data empty	Arbitrary	SPDR	0x14A	0	010	0	1	0	1	
SPI Unit 2	Receive data full	SPDR	Arbitrary	0x14E	0	010	0	1	0	0	
	Transmit data empty	Arbitrary	SPDR	0x14F	0	010	0	1	0	1	
ADC12 Unit 0	Scan end interrupt	ADDRm	Arbitrary	0x159	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	Scan end interrupt for Group B			0x15A							
	Scan end interrupt for Group C			0x15B							
ADC12 Unit 1	Scan end interrupt	ADDRm	Arbitrary	0x15E	0/1	010	0	1	0	0/1	
	Scan end interrupt for Group B			0x15F							
	Scan end interrupt for Group C			0x160							
CANFD	RX FIFO 0 DMA request	CFDRFDF00	Arbitrary	0x1C1	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	RX FIFO 1 DMA request	CFDRFDF01	Arbitrary	0x1C2							
	RX FIFO 2 DMA request	CFDRFDF02	Arbitrary	0x1C3							
	RX FIFO 3 DMA request	CFDRFDF03	Arbitrary	0x1C4							
	RX FIFO 4 DMA request	CFDRFDF04	Arbitrary	0x1C5							
	RX FIFO 5 DMA request	CFDRFDF05	Arbitrary	0x1C6							
	RX FIFO 6 DMA request	CFDRFDF06	Arbitrary	0x1C7							
	RX FIFO 7 DMA request	CFDRFDF07	Arbitrary	0x1C8							
	CH0 first common FIFO DMA request	CFDCFDF00	Arbitrary	0x1C9							
	CH1 first common FIFO DMA request	CFDCFDF03	Arbitrary	0x1CA							

Table 14.19 DMA transfer request detection operation setting table (9 of 9)

DMA transfer request source	DMA transfer source	Transfer source	Transfer destination	REQ_SELn[8:0]	CHCFG_n						
					TM	AM[2:0]*2	LVL	HIEN	LOEN	REQD*2	SEL[2:0]
Mailbox	Mailbox (Host CPU to Cortex-R52) interrupt 0	Arbitrary	Arbitrary	0x174	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	Mailbox (Host CPU to Cortex-R52) interrupt 1			0x175							
	Mailbox (Host CPU to Cortex-R52) interrupt 2			0x176							
	Mailbox (Host CPU to Cortex-R52) interrupt 3			0x177							
DSMIF Unit 0	Current data register update common	DSCDRCHn	Arbitrary	0x178	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	Current capture data register A update common	DSCCDRACHn		0x179							
	Current capture data register B update common	DSCCDRBCHn		0x17A							
DSMIF Unit 1	Current data register update common	DSCDRCHn	Arbitrary	0x17B	0/1	010	0	1	0	0/1	
	Current capture data register A update common	DSCCDRACHn		0x17C							
	Current capture data register B update common	DSCCDRBCHn		0x17D							
ADC12 Unit 0	Compare match	ADDRm	Arbitrary	0x1CB	0/1	010	0	1	0	0/1	DMAC0/1 ch0/8: 0x0 ch1/9: 0x1 ch2/10: 0x2 ch3/11: 0x3 ch4/12: 0x4 ch5/13: 0x5 ch6/14: 0x6 ch7/15: 0x7
	Compare mismatch			0x1CC							
ADC12 Unit 1	Compare match	ADDRm	Arbitrary	0x1CD	0/1	010	0	1	0	0/1	
	Compare mismatch			0x1CE							

Note 1. Set the LVL, HIEN, and LOEN bits of the external interrupts (IRQ0 to IRQ13) as follows:

For the setting of the NS_PORTNF_MD register, see section 11.2.2.4. NS_PORTNF_MD : Interrupt Edge Detection Setting Register.

LVL: Set according to the edge/level setting of the NS_PORTNF_MD register.

HIEN: Set to 1 regardless of the detection level of the NS_PORTNF_MD register.

LOEN: Set to 0 regardless of the detection level of the NS_PORTNF_MD register.

Note 2. If the DACK/TEND signal is not used, specify any setting.

Note 3. When an interrupt occurred at the same time as DMA transfer request, DMA transfer completion interrupt does not occur. DMA transfer completion should be confirmed by END bit in CHSTAT_n register.

Remarks: CHCFG_n register setting values

TM bit
- 0: Single transfer
- 1: Block transfer

AM[2:0] bits
- 001b: DACK level output
- 010b: DACK bus cycle output
- 100b: Masks DACK output

LVL bit
- 0: Detects the edge of a DMA request
- 1: Detects the level of a DMA request

REQD bit
- 0: DACK output during read operation
- 1: DACK output during write operation

Note: Instead of an interrupt source from peripheral modules, transfer completion sources of DMAC channels selected by the DMAC source select register are connected to the vector number selected by the DMAC Source Select Register.

14.3.4.2 Edge Detection

Setting the LVL bit in the CHCFG_n register to 0 detects the edge.

Setting the HIEN bit in the CHCFG_n register to 1 detects the rising edge, and setting the LOEN bit in the CHCFG_n register to 1 detects the falling edge.

For DMA requests by the DREQ signal, request the next DMA transfer after the DACK signal is activated.

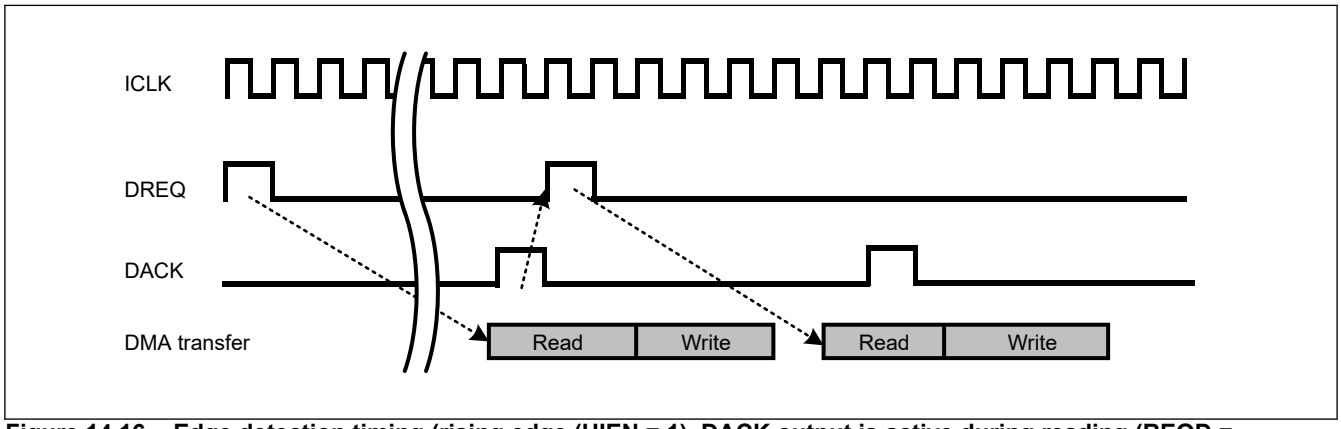


Figure 14.16 Edge detection timing (rising edge (HIEN = 1), DACK output is active during reading (REQD = 0))

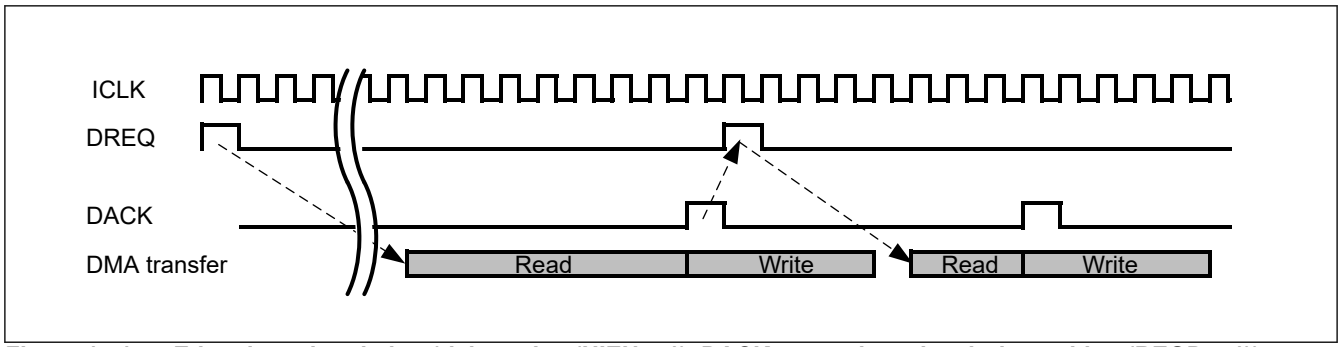


Figure 14.17 Edge detection timing (rising edge (HIEN = 1), DACK output is active during writing (REQD = 1))

14.3.4.3 Level Detection

Setting the LVL bit in the CHCFG_n register to 1 detects the level.

If a DMA transfer request is active (according to HIEN or LOEN settings) for two consecutive clocks (CLK) or more, it is recognized as a DMA request.

If the DACK signal output is set to level mode, the DACK signal output remains at the active level until the DMA transfer request is deactivated.

If you want to request the next DMA transfer, request it after the DACK signal is deactivated.

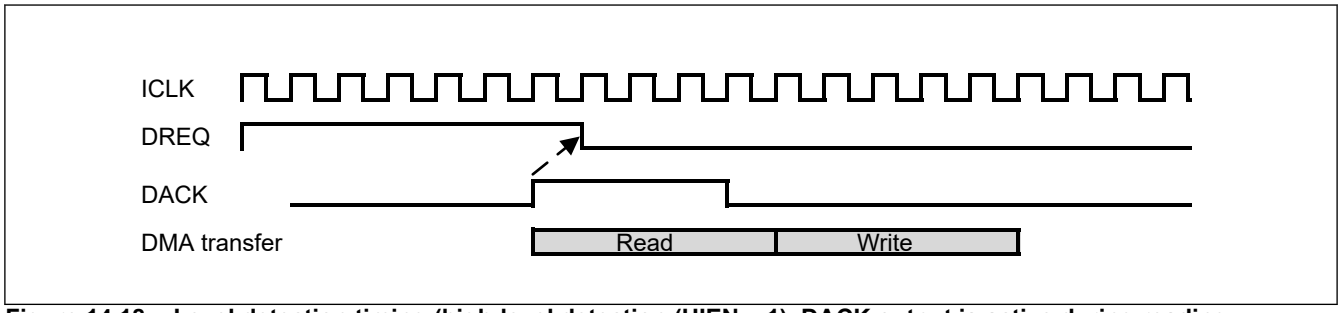


Figure 14.18 Level detection timing (high-level detection (HIEN = 1), DACK output is active during reading (REQD = 0))

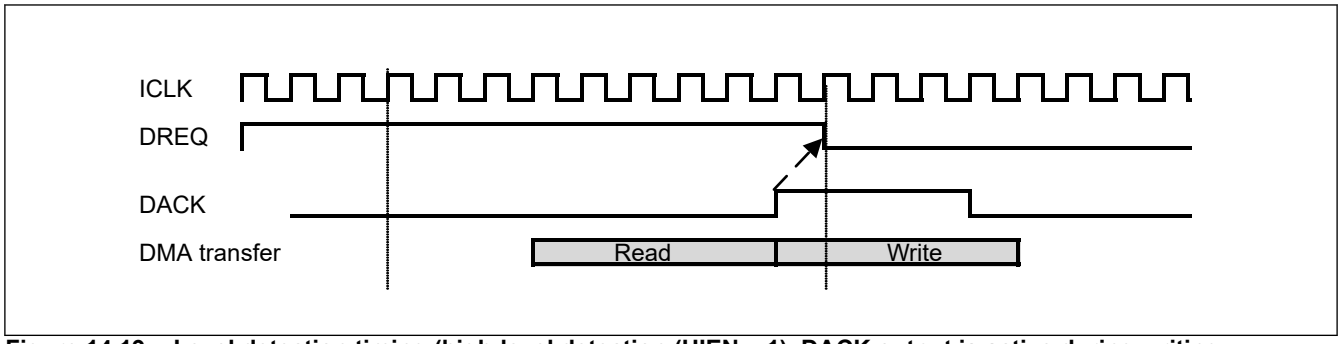


Figure 14.19 Level detection timing (high-level detection (HIEN = 1), DACK output is active during writing (REQD = 1))

14.3.5 DMA Acknowledge Output/DMA Transaction Completion Output Function

The DMA acknowledge signal (DACK) is output as a reception response signal for a DMA transfer request.

The DMA transfer completion signal (TEND) is output when the DACK signal for the last DMA transfer is output.

If the DREQ signal is used for a DMA transfer request, use DACK as the DMA acknowledge signal, and use TEND as the DMA transaction completion output signal.

Set the output mode of the DACK signals with the AM[2:0] bits in the Channel Configuration register (CHCFG_n). Level output and bus cycle output are supported.

The outputs of the DACK and TEND signals on the DACK and TEND pins are active only when the conditions given below are met. To use the DACK and TEND pins, specify appropriate settings for the Port Function Control register (PFC) and the Port Mode Control register (PMC) to use those pins as peripheral functions. For details on pin settings, see section 16, I/O Ports.

The conditions:

- The DREQ pin is set as the source for the DMA transfer request
- The source and destination of the DMA transfer are set as follows:
  - DMA is set to be transferred from an external address space (CSn) to any destination while the CHCFG_n.REQD bit is set to 0 (requested from the source of transfer)
  - DMA is set to be transferred to an external address space (CSn) from any source while the CHCFG_n.REQD bit is set to 1 (requested from the destination of transfer)

The outputs on the pins are not active for the cases other than above. However, the internal signals become active according to the setting of the AM[2:0] bits.

Transfer requested from	Source for DMA transfer	Transferred from	Transferred to	CHCFG_n		External pins used for outputting an active signal	
				AM[2:0]	REQD	DACK Signal	TEND Signal
Request from outside	DREQ	External bus	Anywhere	001/010	0	DACK	TEND
		Anywhere	External bus		1	DACK	TEND
		Internal	Anywhere		0	—	—
		Anywhere	Internal bus		1	—	—
Other than above	Other than above	Anywhere	Anywhere	Any value	Any value	—	—

Note: —: The outputs on the pin are always inactive. Inactive signal level is obtained by inverting the active level set by the bits which specify the acknowledge levels and transfer-end levels.
External bus refers to the CS0 to CS5 spaces and internal bus refers to the address spaces other than the external bus.

14.3.5.1 DMA Acknowledge Signal Output

For the DMA acknowledge signal and DMA transaction completion signal, output mode might be specified according to the source.

For each DMA transfer request source, specify the correct operation with the AM[2:0] bits of the Channel Configuration register (CHCFGn) according to Table 14.17, Table 14.19, and Table 14.20.

For details on level output operation, see Figure 14.20 and Figure 14.21.

For details on bus cycle output operation, see Figure 14.22 and Figure 14.23.

Table 14.20 DACK/TEND pin output settings

Mode	AM[2] (CHCFG_n)	AM[1:0] (CHCFG_n)	REQD (CHCFG_n)	Applications
Level	0	01b	0 (activated during read operation)	In level mode, enable DACK. The DACK output remains active until the DMA transfer request is deactivated. The TEND signal is activated only when the last DMA transfer is performed.
			1 (activated during write operation)	
Bus cycle	0	10b or 11b	0 (activated during read operation)	The DACK signal remains active during a bus cycle. Use this mode if you want to activate the DACK signal until a bus cycle ends.
			1 (activated during write operation)	
Mask	1	Any value	Any value	Set DACK output to inactive (fixed). Use this mode if you do not want to notify the connection destination of DACK output.

14.3.5.2 Level Output

By setting the AM field in the CHCFG_n register to 001b, the mode is switched to level output. DACK output remains active until the DMA transfer request is deactivated.

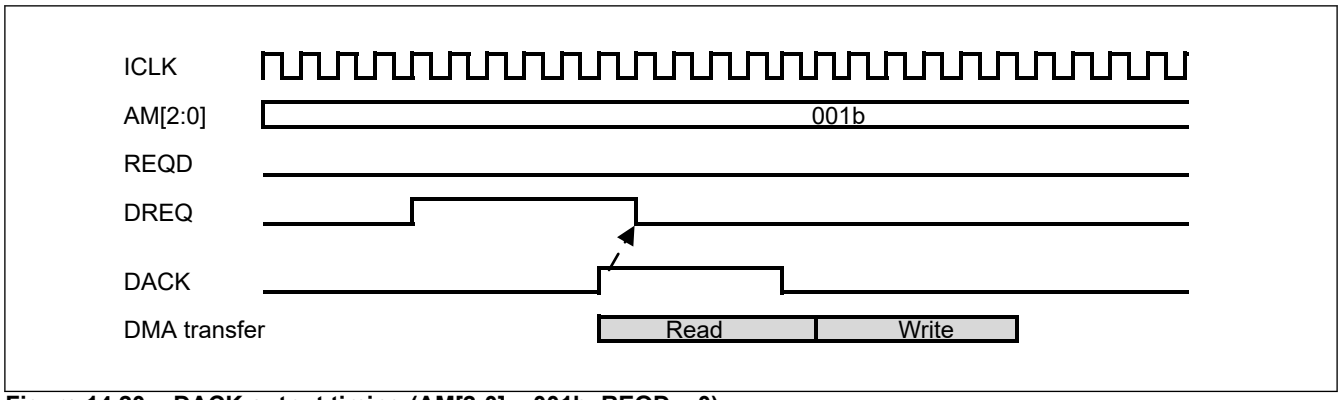


Figure 14.20 DACK output timing (AM[2:0] = 001b, REQD = 0)

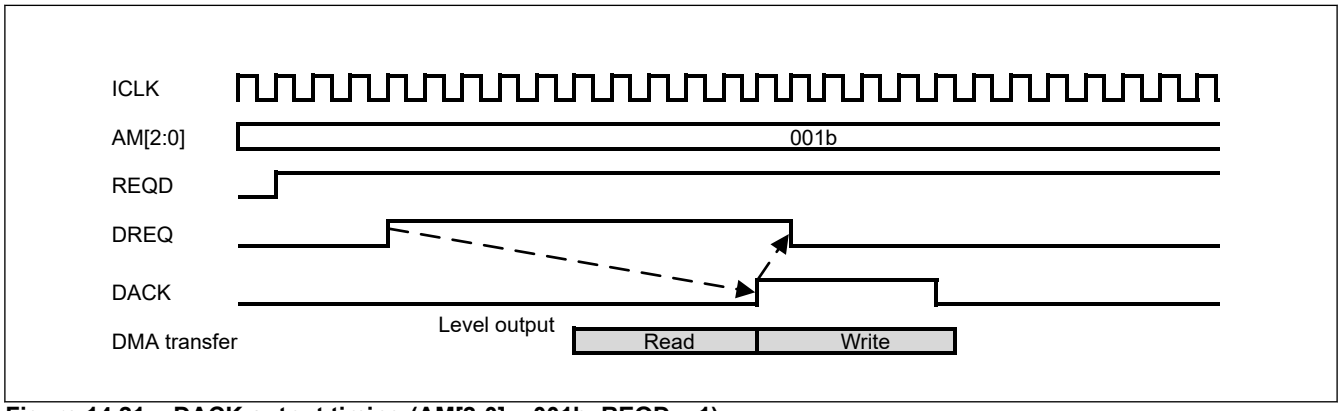


Figure 14.21 DACK output timing (AM[2:0] = 001b, REQD = 1)

14.3.5.3 Bus Cycle Output

By setting the AM field in the CHCFG_n register to 010b, the mode switches to bus cycle output. During a bus cycle, the DACK output remains active.

While the DACK signal is active, the DREQ signal is masked in the CPU. Therefore, even if the DREQ signal is set to level detection, it is not necessary to deactivate the signal on the output side.

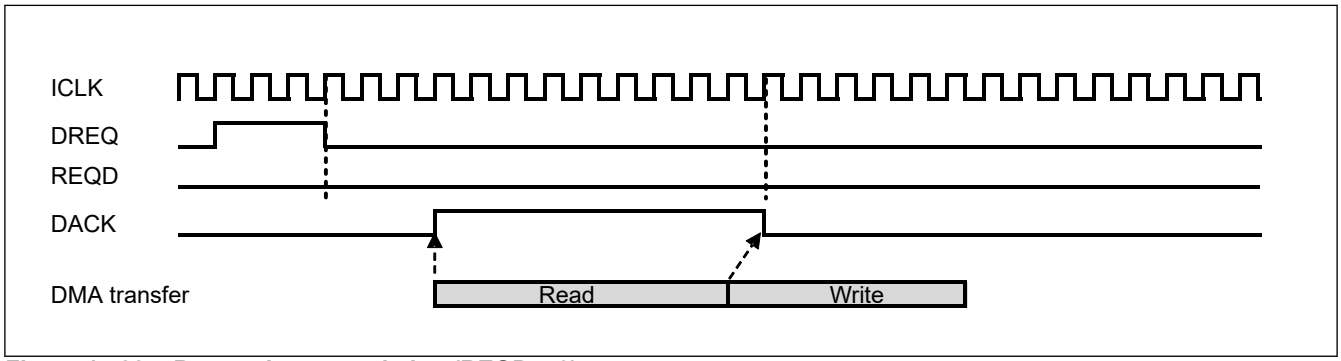


Figure 14.22 Bus cycle output timing (REQD = 0)

- When the REQD bit in the CHCFG_n register = 0 (activated during read operation), the DACK output remains active for the period from when a read request is output on the bus until a cycle elapses after reading the last data.
- If DMA transfer request input is detected by level, DMA transfer request input is deactivated until the next cycle begins after a bus cycle ends.

The following signals trigger the rising and falling of the DACK output:

Rising: Starts a transfer.

Falling: Ends a transfer.

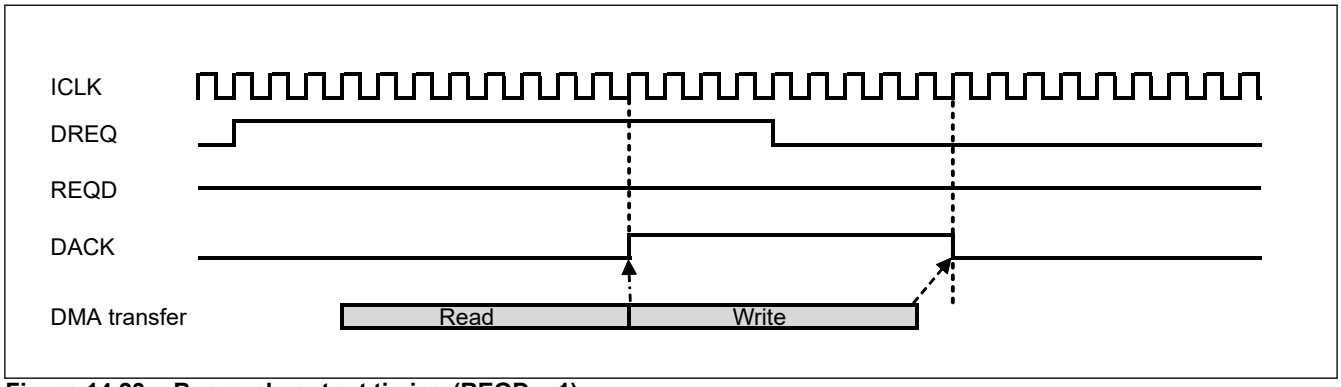


Figure 14.23 Bus cycle output timing (REQD = 1)

- When the REQD bit in the CHCFG_n register = 1 (activated during write operation), the DACK signal remains active for the period from when a write request is output until a clock (ICLK) elapses after responding to the last data.
- If DMA transfer request input is detected by level, DMA transfer request input is deactivated until the next cycle begins after a bus cycle ends.

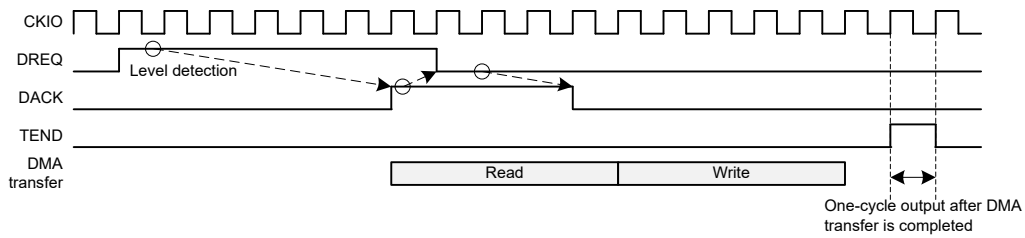
The following signals trigger the rising and falling of the DACK output:

Rising: Starts a transfer.

Falling: Ends a transfer.

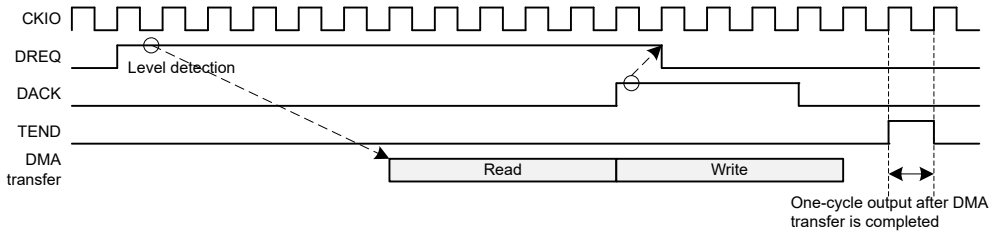
14.3.6 DMA Transaction Completion Signal Output

Figure 14.24 and Figure 14.25 show external DMA interface operation examples. One cycle of the TEND signal is output after the DMA transfer is completed.



Note:
- DMA transfer request: High level detection, request from transfer source (CHCFG_n.REQD = 0)
- DMA acknowledge output: Level mode (CHCFG_n.AM[2:0] = 001b)
- When source and destination are specified to external bus, the TEND signal might be output before write cycle completion.

Figure 14.24 External DMA interface operation example (REQD = 0: read)



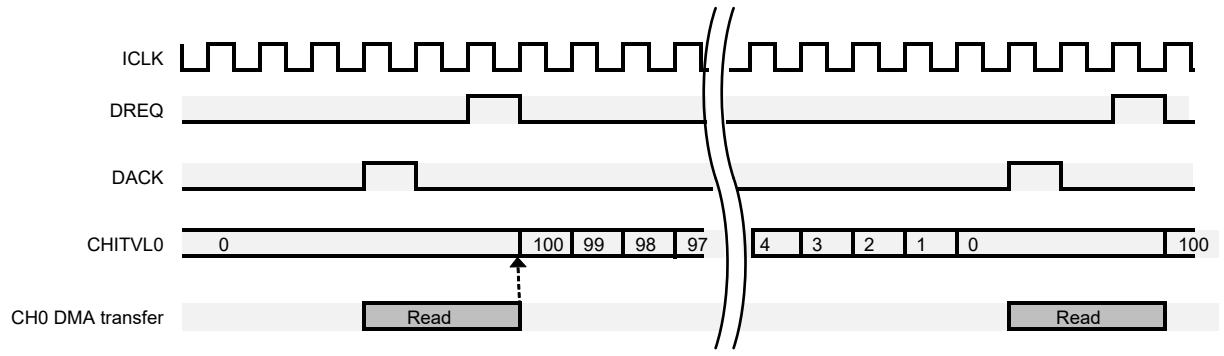
Note:
- DMA transfer request: High level detection, request from transfer source (CHCFG_n.REQD = 1)
- DMA acknowledge output: Level mode (CHCFG_n.AM[2:0] = 001b)

Figure 14.25 External DMA interface operation example (REQD = 1: write)

14.3.7 Interval Count Function

By the setting for the ITVL field in the CHITVL_n register, the execution interval of DMA transfers can be adjusted. This function is used to avoid occupation of a bus by DMAC. Until the count value becomes 0, no DMA transfer for the next DMA request is performed.

Figure 14.26 provides an operation example.



Note: In this example, when REQD = 0 (a transfer request issued by the source module) and SDS < DDS (data size at the destination is greater than that at the source), a preset number of intervals (100) are inserted between several rounds of read access after a transfer request.

Figure 14.26 Interval count (REQD = 0, SDS < DDS)

An interval is inserted after a transfer specified by the REQD bit in the CHCFG_n register. Figure 14.27 shows the relationship between the setting values for the REQD, SDS, and DDS bits in the CHCFG_n register and the cycles to which intervals are applied.

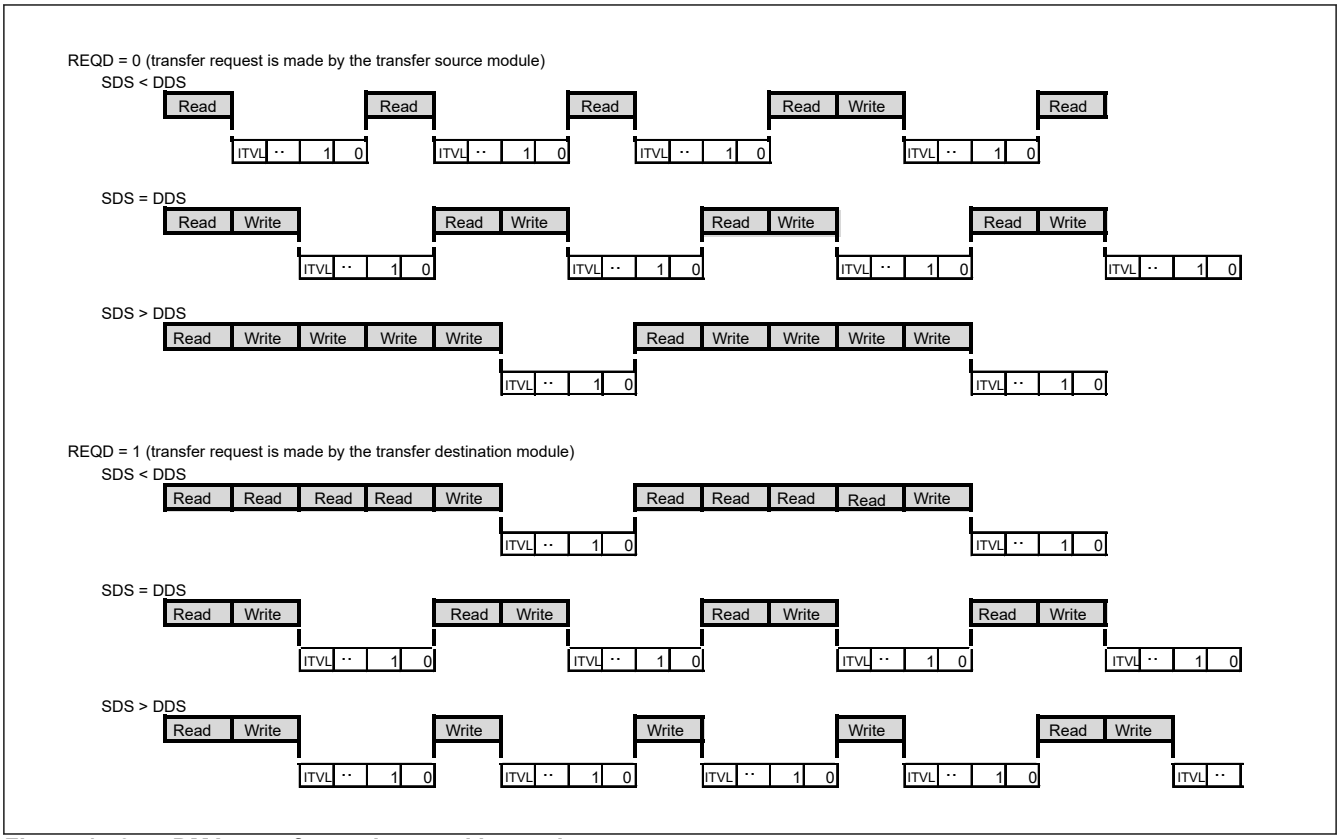


Figure 14.27 DMA transfer settings and interval counts

14.3.8 Differences in Operation According to the Transfer Data Size

14.3.8.1 When the Transfer Data Size on the Transfer Source is Small

When the transfer data size at the destination is large, reading from the source proceeds several times, and this is followed by writing to the destination.

Figure 14.28 shows the timing diagram when the transfer source is 8 bits, and the transfer destination is 32 bits (SDS = 0 and DDS = 2 in the CHCFG_n register) for rising edge detection.

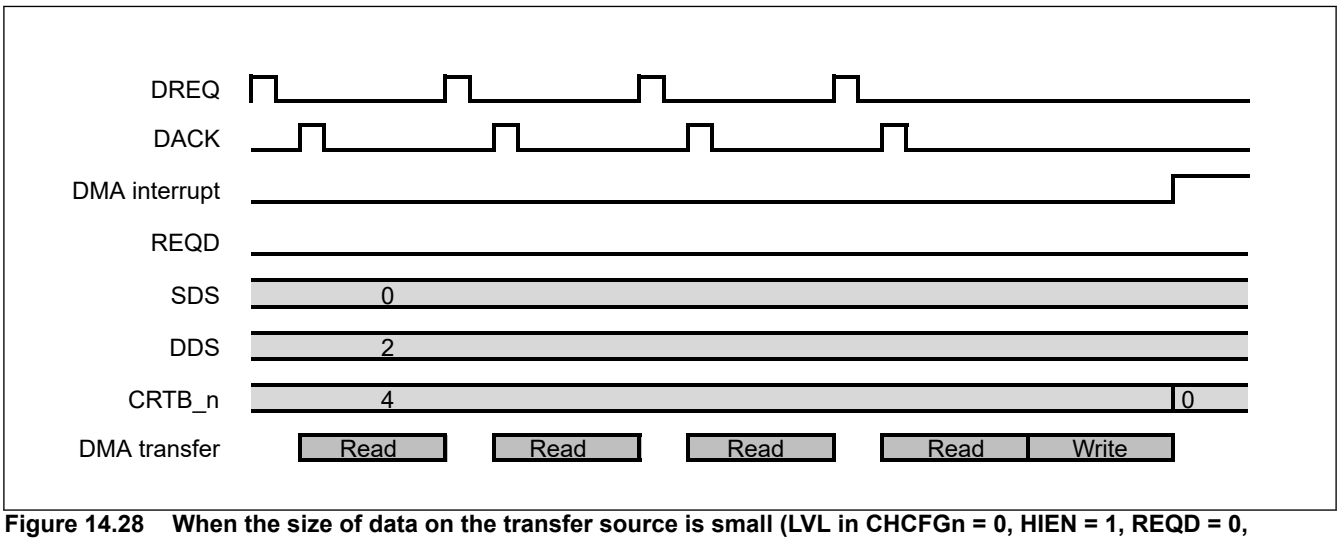


Figure 14.28 When the size of data on the transfer source is small (LVL in CHCFGn = 0, HIEN = 1, REQD = 0, SDS < DDS)

14.3.8.2 When the Transfer Data Size on the Transfer Destination is Small

When the transfer data size on the transfer source is large, after a single read operation, write operation to the transfer destination is performed a few times. Figure 14.29 shows the timing diagram when the transfer source is 64 bits, and the transfer destination is 16 bits (SDS = 3 and DDS = 1 in the CHCFG_n register) for rising edge detection.

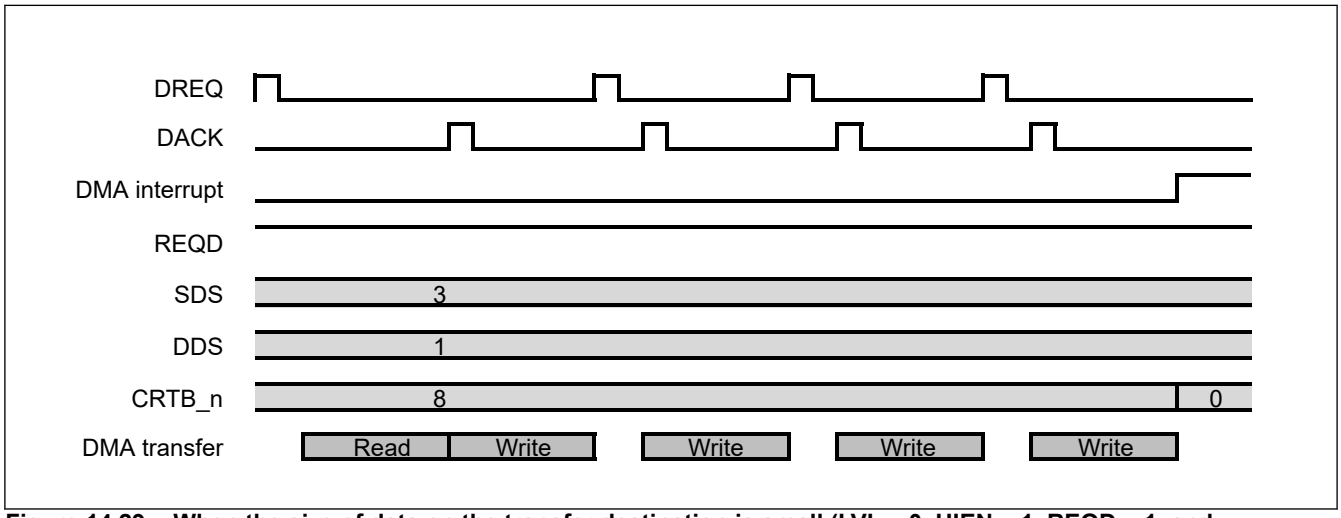


Figure 14.29 When the size of data on the transfer destination is small (LVL = 0, HIEN = 1, REQD = 1, and SDS > DDS in CHCFG_n)

14.3.8.3 When the Size of Transfer Data on the Transfer Source and on the Transfer Destination is the Same

Every time a DMA transfer request is detected, read operation is performed on the transfer source, and write operation is performed on the transfer destination.

Figure 14.30 shows the timing diagram when the transfer source and the transfer destination are 8 bits (SDS = 0 and DDS = 0 in the CHCFG_n register) for rising edge detection.



Figure 14.30 When the sizes of data on the transfer source and the transfer destination are the same (LVL = 0, HIEN = 1, REQD = 0, and SDS = DDS in CHCFG_n)

14.3.9 DMA Transfer Status

The CHSTAT_n register indicates the DMA transfer status of each channel.

The TACT bit in the CHSTAT_n register indicates that DMA operation is being performed on channel n. Writing 1 to the SETEN bit in the CHCTRL_n register sets 1. The TACT bit remains 1 while accessing a descriptor, or waiting for a DMA request.

The TACT bit is cleared when the EN bit in the CHSTAT_n register is cleared (for details on clear conditions, see section 14.2.7. CHSTAT_n : Channel Status Register n (n = 0 to 15)), and DMA transfers for the set number of times are finished.

The TACT bit is not cleared even when a DMA transfer finishes, but the EN bit is not cleared (when the REN bit in the CHCFG_n register = 1 in register mode or the next descriptor access is performed in link mode).

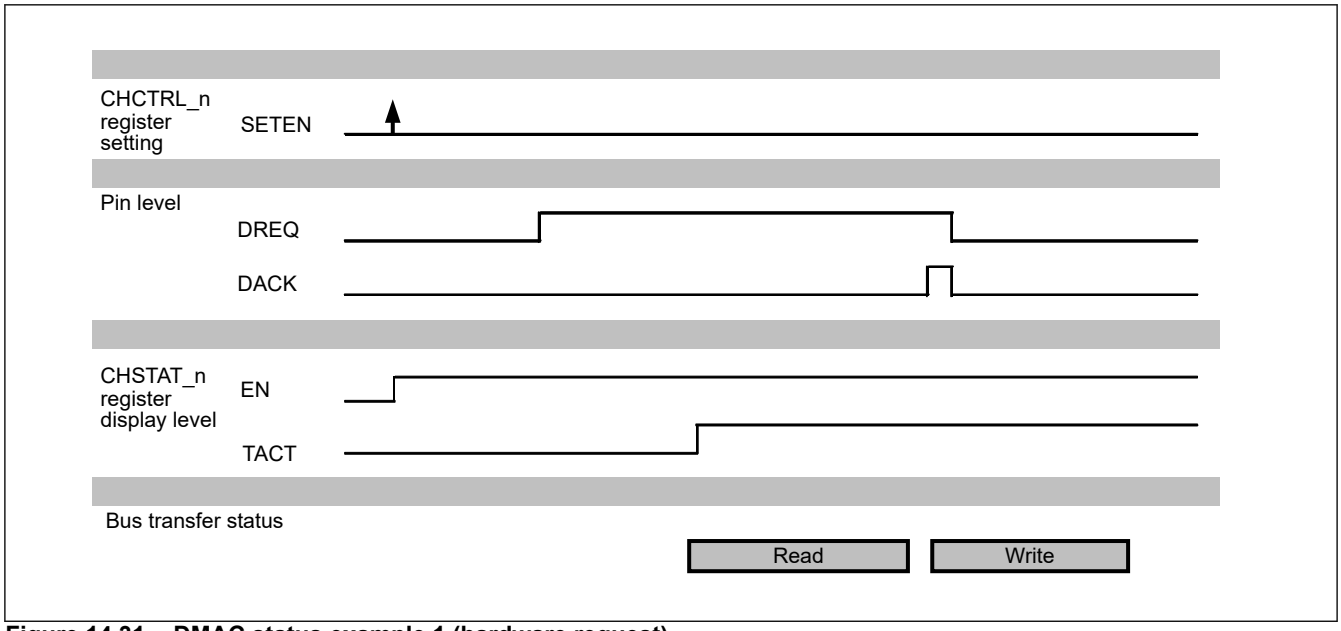


Figure 14.31 DMAC status example 1 (hardware request)

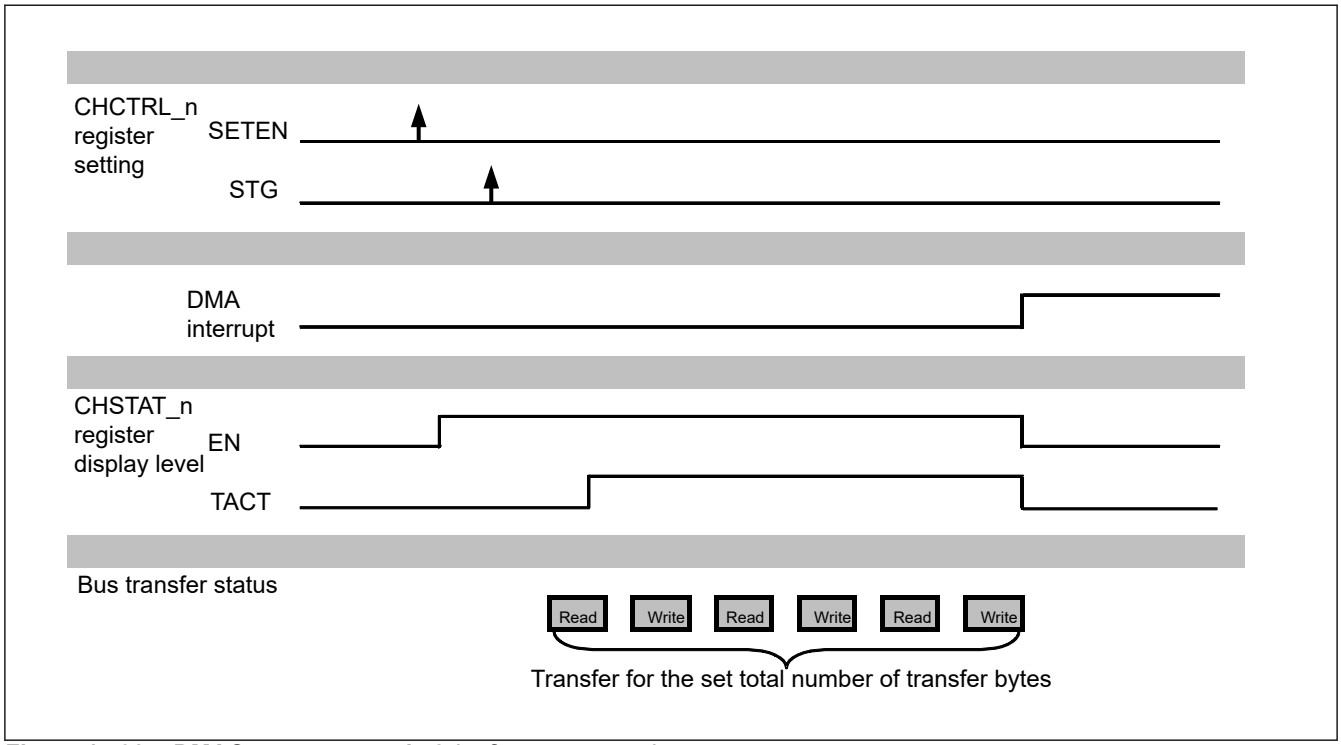


Figure 14.32 DMAC status example 2 (software request)

14.3.10 Suspending a Transfer

You can suspend a DMA transfer at the SETSUS bit in the CHCTRL_n register. At that time, if there is an already running bus cycle, waits for the cycle to end, then suspends the transfer. Writing 1 to the CLRSUS bit in the CHCTRL_n register resumes from the suspended state.

To check if the transfer is suspended, set the SETSUS bit in the CHCTRL_n register, and then make sure that the SUS bit in the CHSTAT_n register or the SUS bit in the DSTAT_SUS register on the applicable channel is set to 1.

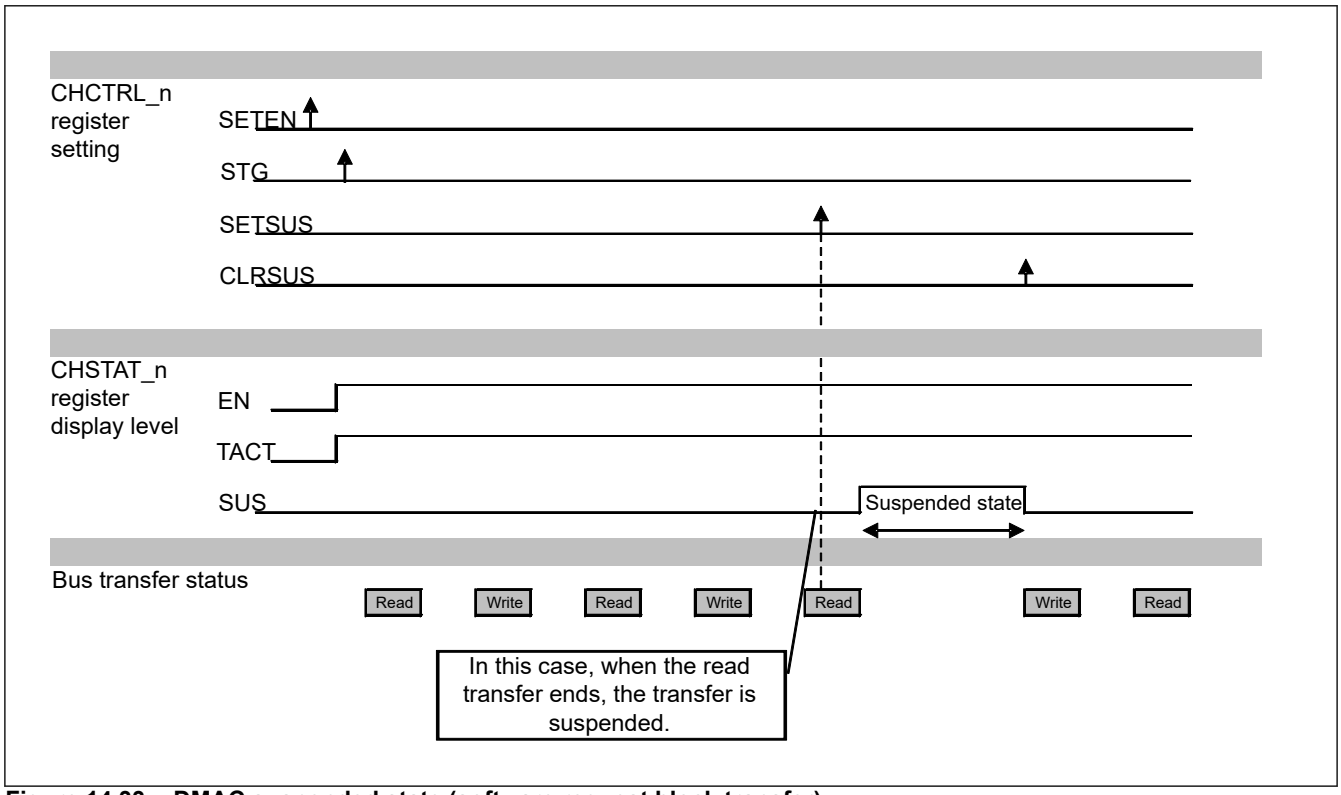


Figure 14.33 DMAC suspended state (software request block transfer)

14.3.11 Aborting a Transfer

During a DMA transfer, if you write 1 to the CLREN bit in the CHCTRL_n register, you can abort the DMA transfer of the channel. As processing after aborting the transfer, you can use the SBE bit in the CHCFG_n register to determine whether to flush data remaining in a buffer when a transfer is suspended. By default, SBE = 0 (do not flush data) is selected.

When this mode (flush data) is activated, if a transfer that is being performed when the CLREN bit in the CHCTRL_n register = 1 is aborted, data remaining in the buffer of DMAC is flushed, and the operation stops.

14.3.11.1 Aborting a Transfer (No Buffer Flush: SBE = 0)

During a DMA transfer, if you write 1 to the CLREN bit in the CHCTRL_n register, you can abort and then stop the DMA transfer. The timing to stop the transfer depends on the value set for the REQD bit. After the transfer stops, write 1 to the SWRST bit in the CHCTRL_n register, and clear the contents within the DMAC before setting the next transfer.

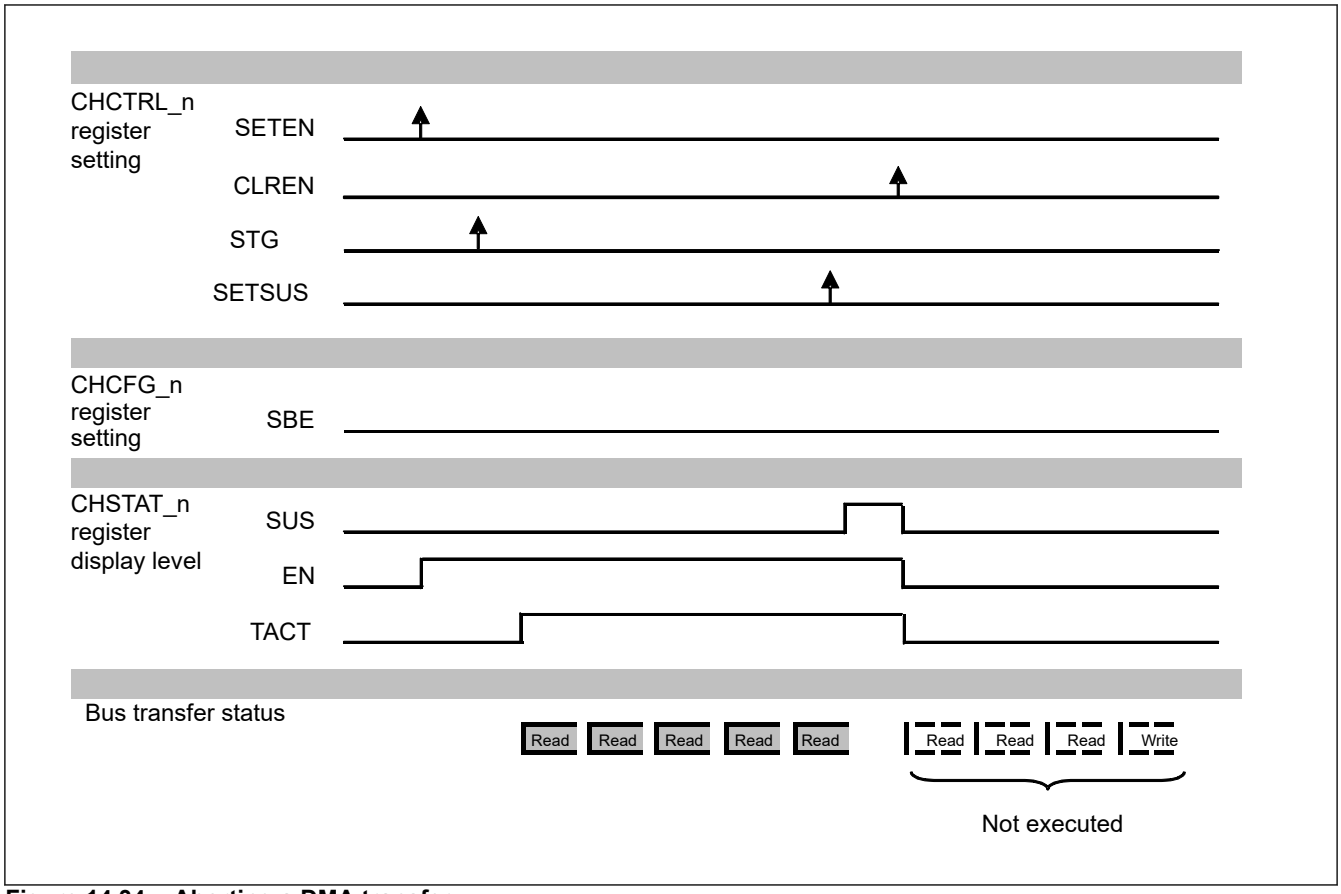


Figure 14.34 Aborting a DMA transfer

- When the TACT bit in the CHSTAT_n register is cleared, you can confirm that the channel stops completely.
- If a DMA transfer is aborted, no DMA transfer completion interrupt is generated.
- When the REQD bit in the CHCFG_n register = 0, the transfer stops when the next read operation completes. If there is data that can be written in the buffer, the transfer stops after the data is written.
- When the REQD bit in the CHCFG_n register = 1, the transfer stops when the next write operation completes.

14.3.11.2 Aborting a Transfer (Buffer Flush: SBE = 1)

During a DMA transfer, if you write 1 to the CLREN bit in the CHCTRL_n register, you can abort the DMA transfer.

When the REQD bit in the CHCFG_n register = 0, already read data is flushed (written), and then the DMA transfer stops. When REQD = 1, flush mode cannot be used.

After the transfer stops, set the SWRST bit in the CHCTRL_n register, and clear the contents within the DMAC before setting the next transfer.

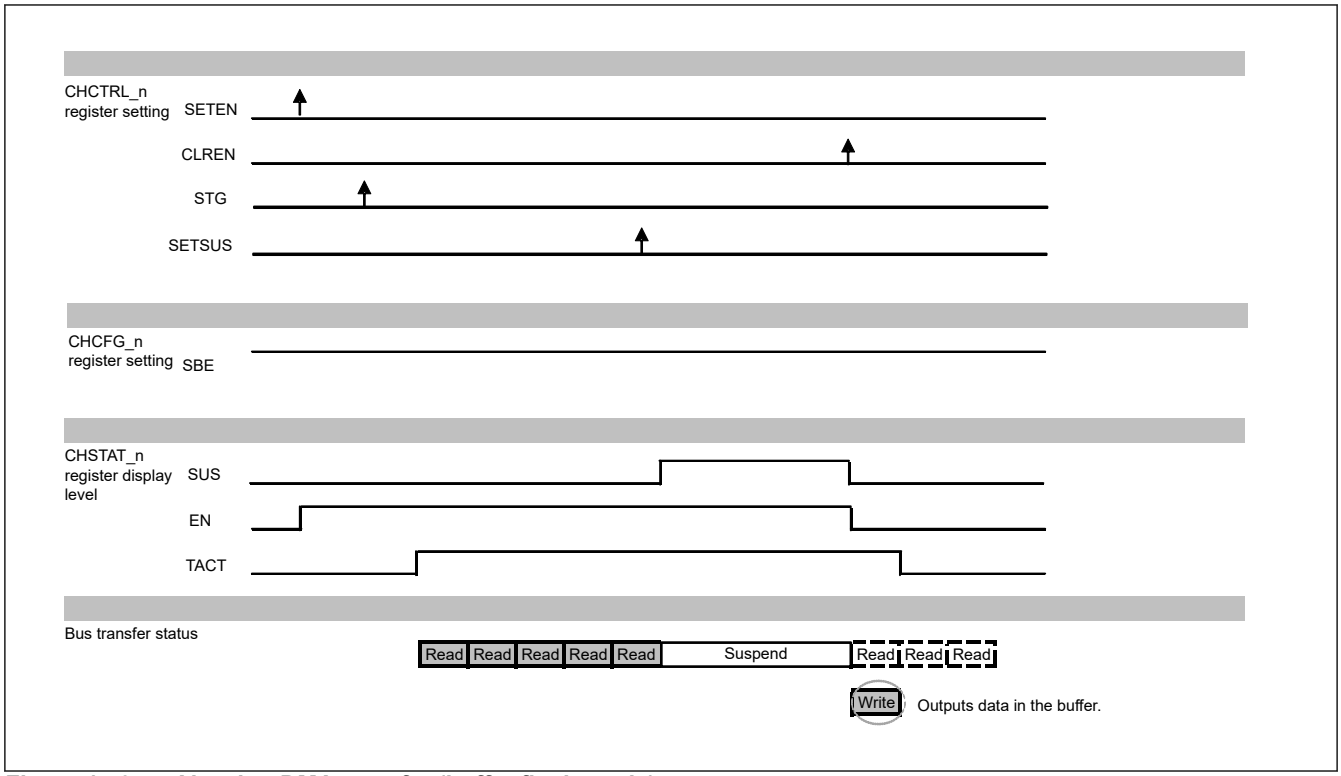


Figure 14.35 Aborting DMA transfer (buffer flush mode)

- Figure 14.35 shows an example when 1 is written to the CLREN bit in the CHCTRL_n register during the fifth read transfer in flush mode (the SBE bit in the CHCFG_n register = 1), and the transfer is aborted. It shows how read data is written, and the DMA transfer stops.
- When the TACT bit in the CHSTAT_n register is cleared to 0, you can confirm that the channel completely stops.

14.3.11.3 Checking if the Channel Stops

When 1 is written to the CLREN bit in the CHCTRL_n register, and the EN bit in the CHSTAT_n register is cleared to 0, if a transfer is already performed on a bus, DMAC cannot stop immediately. To check if DMA completely stops, make sure that the EN bit is cleared to 0, and the TACT bit in the CHSTAT_n register is cleared to 0.

14.3.11.4 Procedure for Aborting a Transfer

The following steps show the procedure for stopping a transfer:

1. Write 1 to the SETSUS bit in the CHCTRL_n register.
2. Perform polling until the SUS bit in the CHCTRL_n register is set to 1. If the EN bit = 0, go to step 6 because DMAC transfer is suspended.
3. Write 1 to the CLREN bit in the CHCTRL_n register.
4. When the SBE bit in the CHCFG_n register = 0, the transfer stops according to value for the REQD bit in the CHCFG_n register. At that time, if SBE = 1, flush mode is used.
5. By reading the CHSTAT_n register, make sure the TACT bit is cleared to 0. When TACT = 0, DMA stops completely. When TACT = 1, perform polling until the TACT bit is cleared to 0.
6. After a transfer is aborted, to perform the next DMA transfer, you must set the SWRST (software reset) bit in the CHCTRL_n register immediately before the next transfer starts.

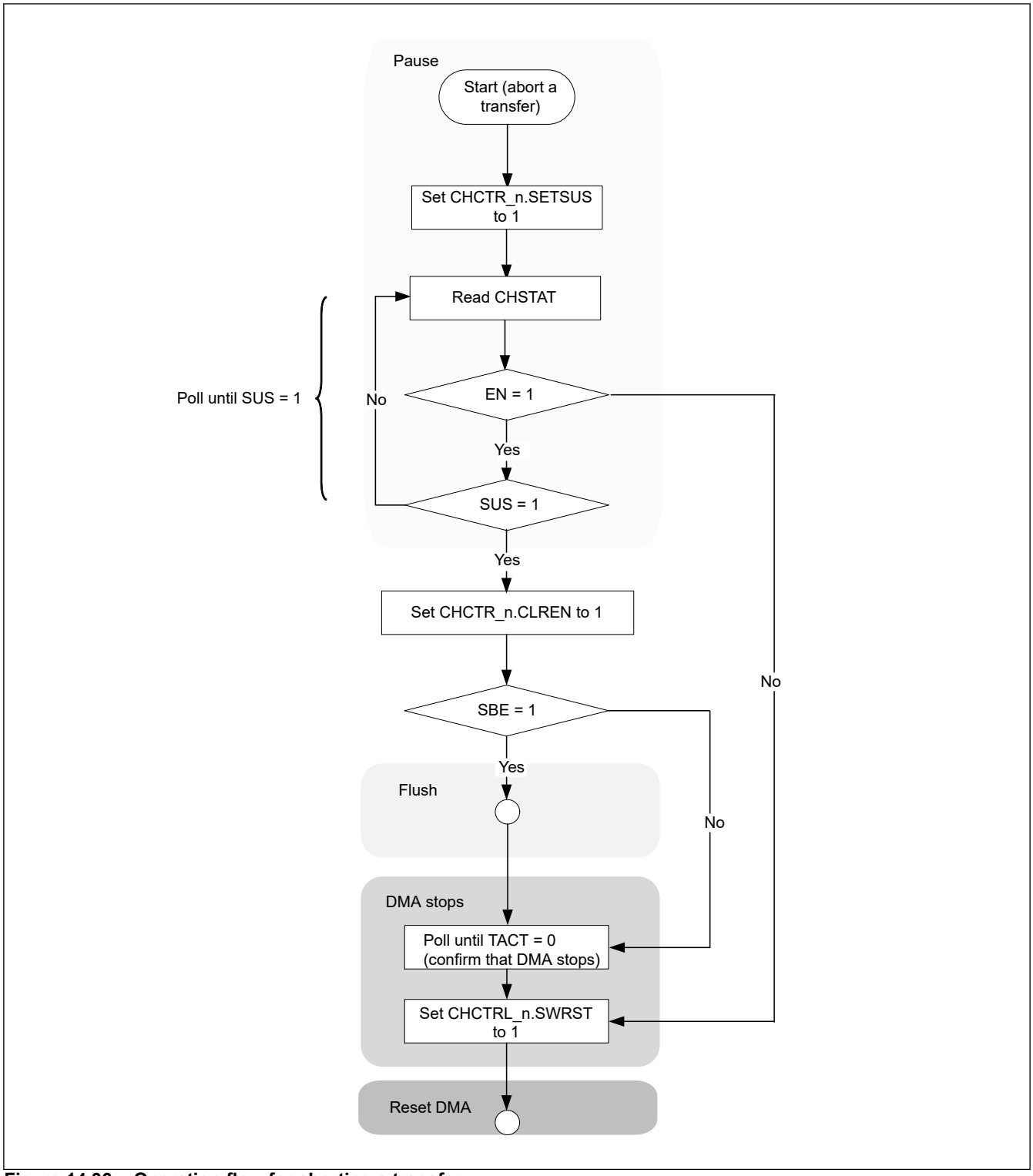


Figure 14.36 Operation flow for aborting a transfer

14.4 Interrupts

14.4.1 Interrupt Sources

DMAC has two types of interrupt sources, DMA transfer completion interrupts and DMA error interrupts for each channel. Table 14.21 shows the relationship among interrupt sources, enable bits, and status flags.

Table 14.21 Interrupt sources of DMAC

Interrupt source		Interrupt Enable bit	Interrupt Status flag	Output condition
DMA transfer completion interrupt	DMA transfer completion	CHCFG_n.DEM	CHSTAT_n.END	When a transfer for the total number of transfer bytes loaded to the CRTB_n register completes (after a write back if write back is performed in link mode)
	Descriptor invalid	DIM in header register		In link mode, when the header of the LV of the read descriptor = 0
DMA error interrupt		— (mask disabled)	CHSTAT_n.ER	When a bus error occurs during a DMA transfer and descriptor access

14.4.2 DMA Transfer Completion Interrupts

A DMA transfer completion interrupt is an interrupt request signal indicating that the DMA transfer completes. Each bit of the DMA transfer completion interrupt corresponds to each channel.

When the transfer of the total number of bytes loaded to the CRTB_n register is completed, the END bit in the CHSTAT_n register is set to 1. At that time, if the DEM bit in the CHCFG_n register = 0, a DMA transfer completion interrupt is generated (n = 0 to 15). To perform write back in link mode, an interrupt is generated after the write back.

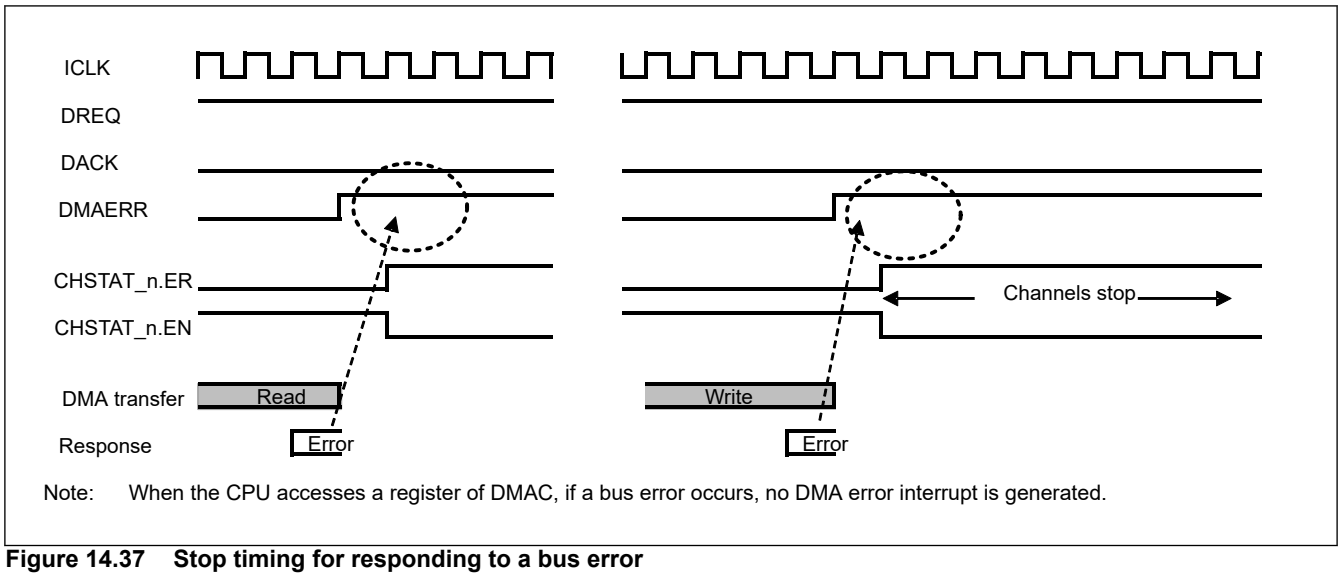
In addition, when link mode and header of the read descriptor is LV = 0, the DER bit in the CHSTAT_n register is set to 1. At that time, if the DIM bit in the header register = 0, a DMA transfer completion interrupt is generated.

14.4.3 DMA Error Interrupt

If a bus error occurs during a DMA transfer or descriptor access, this module determines that an error occurred, and stops the transfer. When a bus error occurs, the EN bit in the CHSTAT_n register for channel n where a transfer is performed is cleared to 0, and the ER bit is set to 1 (n = 0 to 15). In addition, a DMA error interrupt is generated. DMA error interrupts cannot be masked.

Data for a series of error transfers cannot be guaranteed. You must perform the transfer from the beginning by using the following procedure:

1. Set the SWRST bit in the CHCTRL_n register to 1.
2. Set each register again.



Note: When the CPU accesses a register of DMAC, if a bus error occurs, no DMA error interrupt is generated.

Figure 14.37 Stop timing for responding to a bus error

14.5 DMA Setting Examples

This section provides examples of DMA transfers. Table 14.22 lists the transfer conditions for the setting examples described in this section.

Table 14.22 List of transfer conditions for the DMA transfer setting examples

Setting example	DMA mode	Transfer mode	Transfer request
Setting example 1	Register mode	Single transfer mode	Hardware
Setting example 2	Register mode	Block transfer mode	Software
Setting example 3	Register mode (continuous execution)	Block transfer mode	Software
Setting example 4	Link mode	Block transfer mode	Software

14.5.1 Setting Example 1 (Register Hardware Request)

This section provides setting examples of DMA transfers that use hardware requests in register mode.

Table 14.23 DMA transfer setting example 1

Parameter		Description
Channel to use		DMAC0 channel 3
Priority control		Fixed priority
DMA mode		Register mode
Transfer mode		Single transfer mode
Register set to use		Next0 register set
Transfer source	Start address	0x6000_0000
	Address direction	Increment
	Data size	32 bits
Transfer destination	Start address	0x1000_0000
	Address direction	Increment
	Data size	32 bits
Number of DMA transfer bytes		64 bytes
DMA transfer request		Detects the rising edge by hardware (DREQ)
Selection of the side which makes a DMA transfer request		Requested from the transfer-side module
DACK/TEND signal		Outputs by level during read operation
DMA transfer completion interrupt mask		None

Setting Example 1

N0SA = 0x6000_0000 (transfer source address)

N0DA = 0x1000_0000 (transfer destination address)

N0TB = 0x0000_0040 (number of transfer bytes)

CHCFG = 0x0002_2123 (configuration)

CHITVL = 0x0000_0000 (interval)

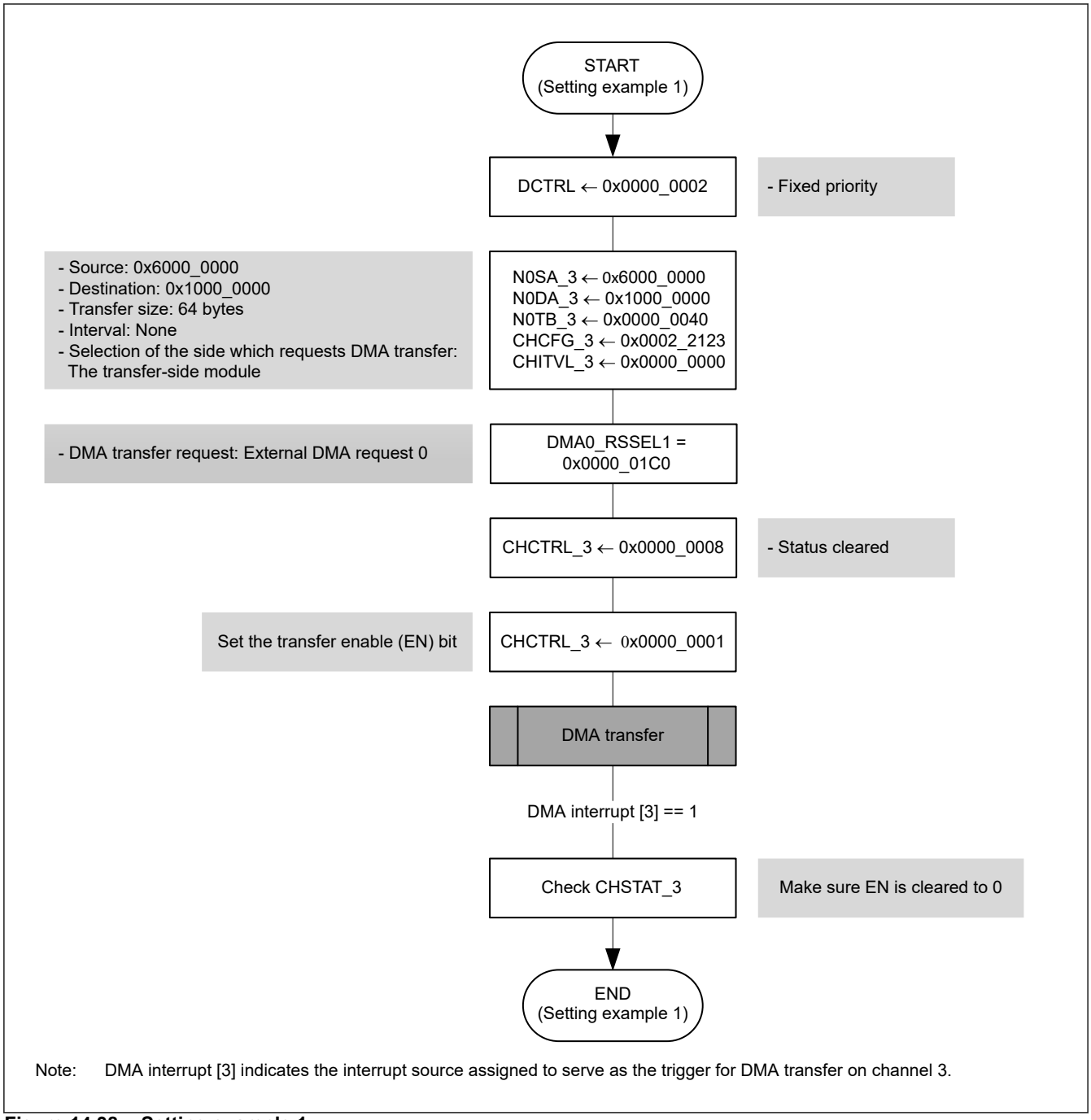


Figure 14.38 Setting example 1

14.5.2 Setting Example 2 (Register Mode Software Request)

This section provides a setting example of DMA transfers that use software requests in register mode.

Table 14.24 DMA transfer setting example 2 (1 of 2)

Parameter	Description
Channel to use	DMAC0 channel 2
Priority control	Round-robin
DMA mode	Register mode
Transfer mode	Block transfer mode
Register set to use	Next1 register set

Table 14.24 DMA transfer setting example 2 (2 of 2)

Parameter		Description
Transfer source	Start address	0x1000_0000
	Address direction	Increment
	Data size	8 bits
Transfer destination	Start address	0x1000_1000
	Address direction	Increment
	Data size	256 bits
Number of DMA transfer bytes		128 bytes
DMA transfer request		Software request
DACK/TEND signal		Mask
DMA transfer completion interrupt mask		None

Setting Example 2

DCTRL = 0x0000_0003 (DMA setting)

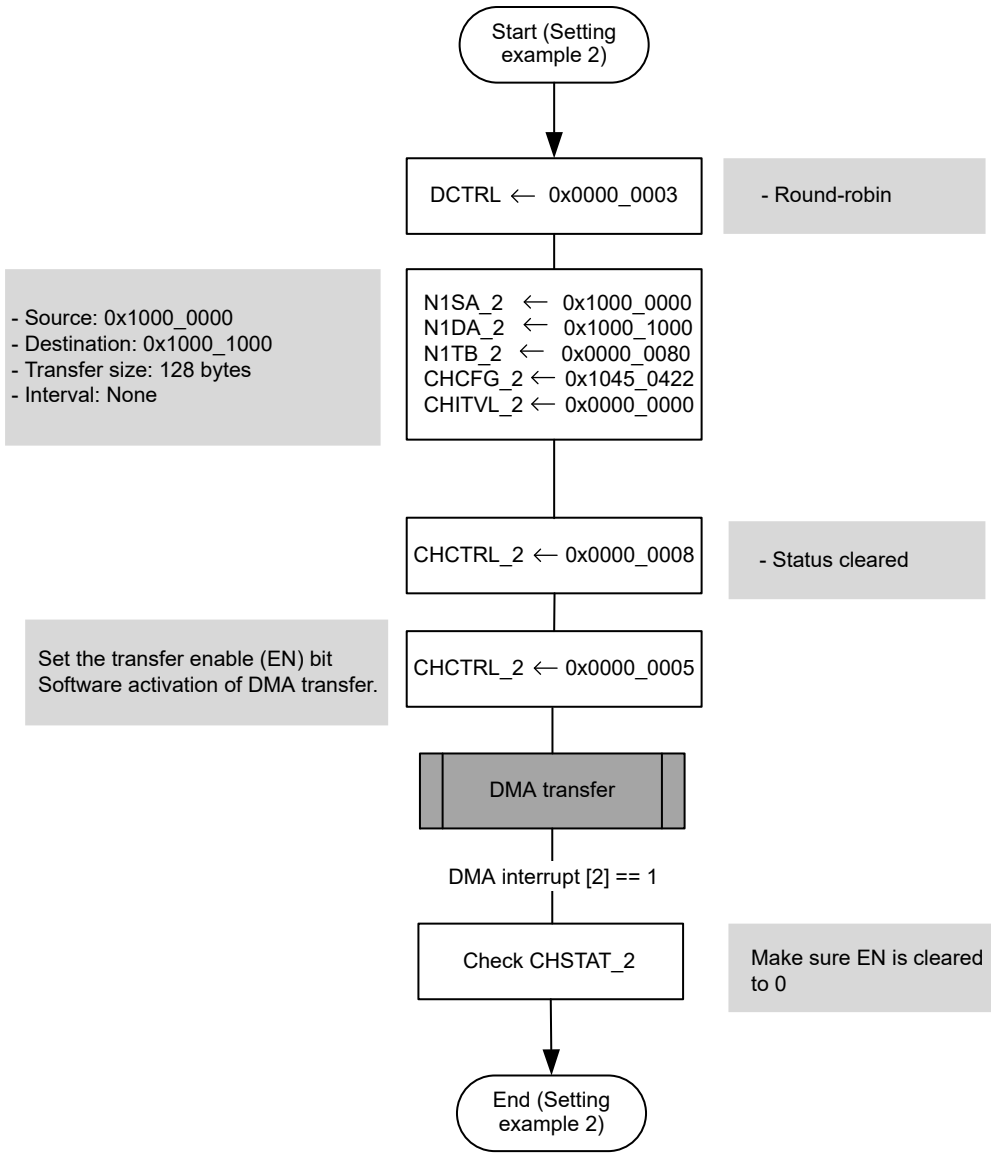
N1SA = 0x1000_0000 (transfer source address)

N1DA = 0x1000_1000 (transfer destination address)

N1TB = 0x0000_0080 (number of transfer bytes)

CHCFG = 0x1045_0422 (configuration)

CHITVL = 0x0000_0000 (interval)



Note: DMA interrupt [2] indicates the interrupt source assigned to serve as the trigger for DMA transfer on channel 2.

Figure 14.39 Setting example 2

14.5.3 Setting Example 3 (Register Mode Continuous Execution)

This section provides a setting example of DMA transfers that use Next0/1 Register Sets in series in register mode.

Table 14.25 DMA transfer setting example 3 (1 of 2)

Parameter	Description
Channel to use	DMAC0 channel 1
Priority control	Round-robin
DMA mode	Register mode
Transfer mode	Block transfer mode
Register set to use	Next0 register set → Next1 register set in series

Table 14.25 DMA transfer setting example 3 (2 of 2)

Parameter		Description
Transfer source (Next0)	Start address	0x1000_4000
	Address direction	Fixed
	Data size	32 bits
	Number of DMA transfer bytes	512 bytes
Transfer destination (Next0)	Start address	0x1000_5000
	Address direction	Fixed
	Data size	512 bits
	Number of DMA transfer bytes	512 bytes
Transfer source (Next1)	Start address	0x1000_6000
	Address direction	Fixed
	Data size	32 bits
	Number of DMA transfer bytes	2048 bytes
Transfer destination (Next1)	Start address	0x1000_7000
	Address direction	Fixed
	Data size	512 bits
	Number of DMA transfer bytes	2048 bytes
DMA transfer request		Software request
DACK/TEND signal		Mask
DMA transfer completion interrupt mask		Masks when Next0 completes

Setting Example 3

DCTRL = 0x0000_0003 (DMA setting)

N0SA = 0x1000_4000 (transfer source address)

N0DA = 0x1000_5000 (transfer destination address)

N0TB = 0x0000_0200 (number of transfer bytes)

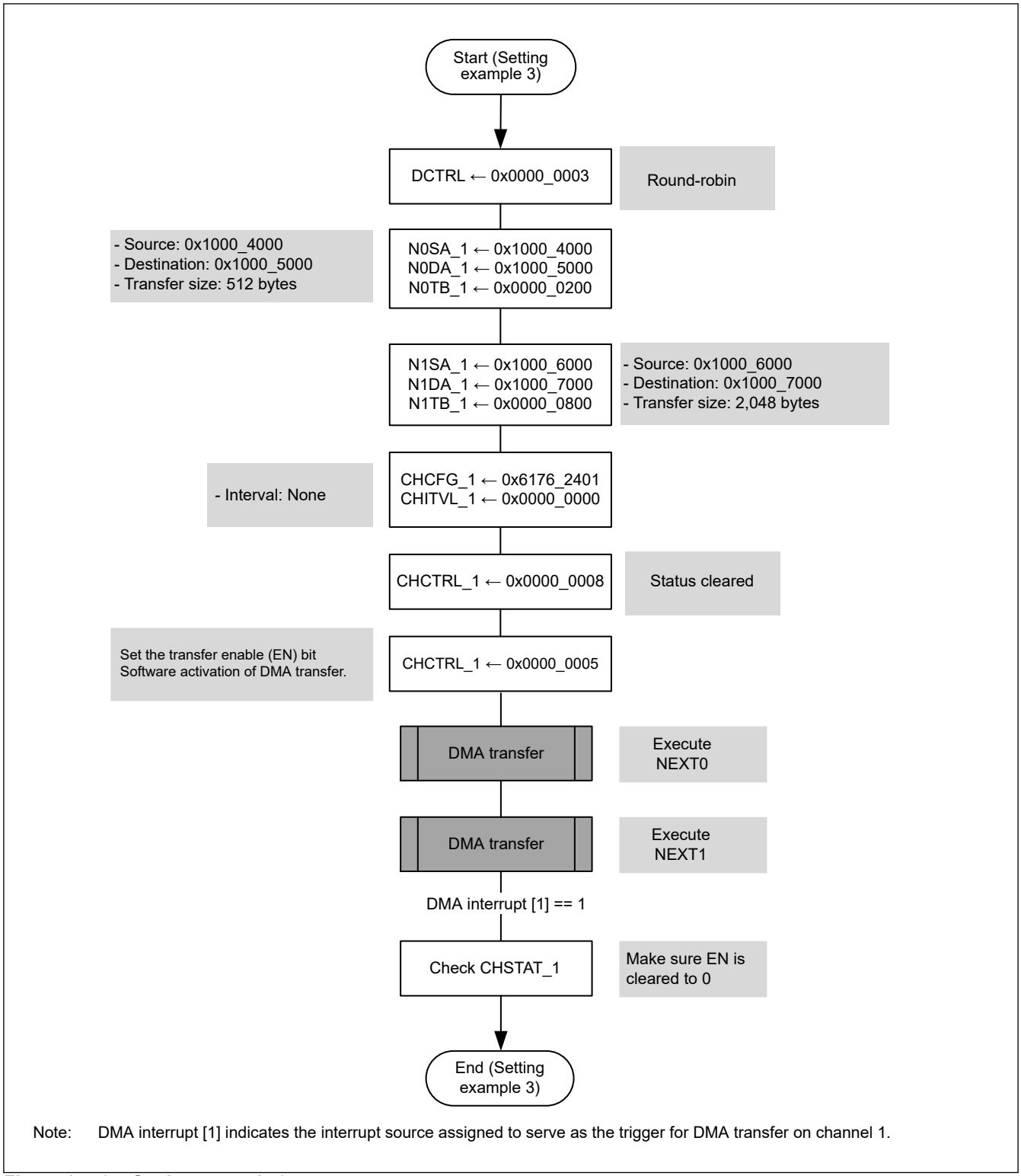
N1SA = 0x1000_6000 (transfer source address)

N1DA = 0x1000_7000 (transfer destination address)

N1TB = 0x0000_0800 (number of transfer bytes)

CHCFG = 0x6176_2401 (configuration)

CHITVL = 0x0000_0000 (interval)



Note: DMA interrupt [1] indicates the interrupt source assigned to serve as the trigger for DMA transfer on channel 1.

Figure 14.40 Setting example 3

14.5.4 Setting Example 4 (Link Mode)

This section provides a setting example when a DMA transfer is performed in link mode.

Table 14.26 DMA transfer setting example 4 (1 of 2)

Parameter	Description
Channel to use	DMAC0 channel 0

Table 14.26 DMA transfer setting example 4 (2 of 2)

Parameter	Description
Priority control	Round-robin
DMA mode	Link mode
Transfer mode	Block transfer mode
Descriptor start address	0x1080_0000

Table 14.27 DMA transfer setting example 4 (descriptor 1)

Parameter		Description
Descriptor start address		0x1080_0000
Next descriptor start address		0x1080_1000
Transfer mode		Block transfer mode
Transfer source	Start address	0x1000_0000
	Address direction	Increment
	Data size	32 bits
Transfer destination	Start address	0x1000_1000
	Address direction	Increment
	Data size	32 bits
Number of DMA transfer bytes		2048 bytes
DMA transfer request		Software request
DACK/TEND signal		Mask
DMA transfer completion interrupt output mask		Masks when a DMA transfer on descriptor 1 completes
Descriptor header	DMA interrupt when LV =1	Issued (DIM = 0)
	LV write back	Done (WBD = 0)
	Next link address	Available (LE = 0)
	Descriptor valid	Valid (LV = 1)

Table 14.28 DMA transfer setting example 4 (descriptor 2) (1 of 2)

Parameter		Description
Descriptor start address		0x1080_1000
Next descriptor start address		0x1080_2000
Transfer mode		Block transfer mode
Transfer source	Start address	0x1000_2000
	Address direction	Increment
	Data size	256 bits
Transfer destination	Start address	0x1000_3000
	Address direction	Increment
	Data size	256 bits
Number of DMA transfer bytes		1024 bytes
DMA transfer request		Software request
DACK/TEND signal		Mask
DMA transfer completion interrupt output mask		Masks when a DMA transfer on descriptor 2 completes

Table 14.28 DMA transfer setting example 4 (descriptor 2) (2 of 2)

Parameter		Description
Descriptor header	DMA interrupt when LV =1	Issued (DIM = 0)
	LV write back	Done (WBD = 0)
	Next link address	Available (LE = 0)
	Descriptor valid	Valid (LV = 1)

Table 14.29 DMA transfer setting example 4 (descriptor 3)

Parameter		Description
Descriptor start address		0x1080_2000
Next descriptor start address		—
Transfer mode		Block transfer mode
Transfer source	Start address	0x1000_4000
	Address direction	Increment
	Data size	512 bits
Transfer destination	Start address	0x1000_6000
	Address direction	Increment
	Data size	512 bits
Number of DMA transfer bytes		4096 bytes
DMA transfer request		Software request
DACK/TEND signal		Mask
DMA transfer completion interrupt output mask		Do not mask
Descriptor header	DMA interrupt when LV =1	Issued (DIM = 0)
	LV write back	Done (WBD = 0)
	Next link address	Not available (LE = 1)
	Descriptor valid	Valid (LV = 1)

Setting Example 4

DCTRL = 0x0000_0003 (DMA setting)

NXLA = 0x1080_0000 (descriptor start address)

CHCFG = 0x8000_0000 (configuration)

Table 14.30 Descriptor settings

Parameter	Descriptor 1	Descriptor 2	Descriptor 3
Header	0x0000_0001	0x0000_0001	0x0000_0003
SA (Source Address)	0x1000_0000	0x1000_2000	0x1000_4000
DA (Destination Address)	0x1000_1000	0x1000_3000	0x1000_6000
TB (Transaction Bytes)	0x0000_0800	0x0000_0400	0x0000_1000
CFG (Configuration)	0x8142_2220	0x8145_5220	0x8046_6220
ITVL (Interval)	0x0000_0000	0x0000_0000	0x0000_0000
EXT (Extension)	0x0000_0000	0x0000_0000	0x0000_0000
NXLA (Next Link Address)	0x1080_1000	0x1080_2000	0x0000_0000

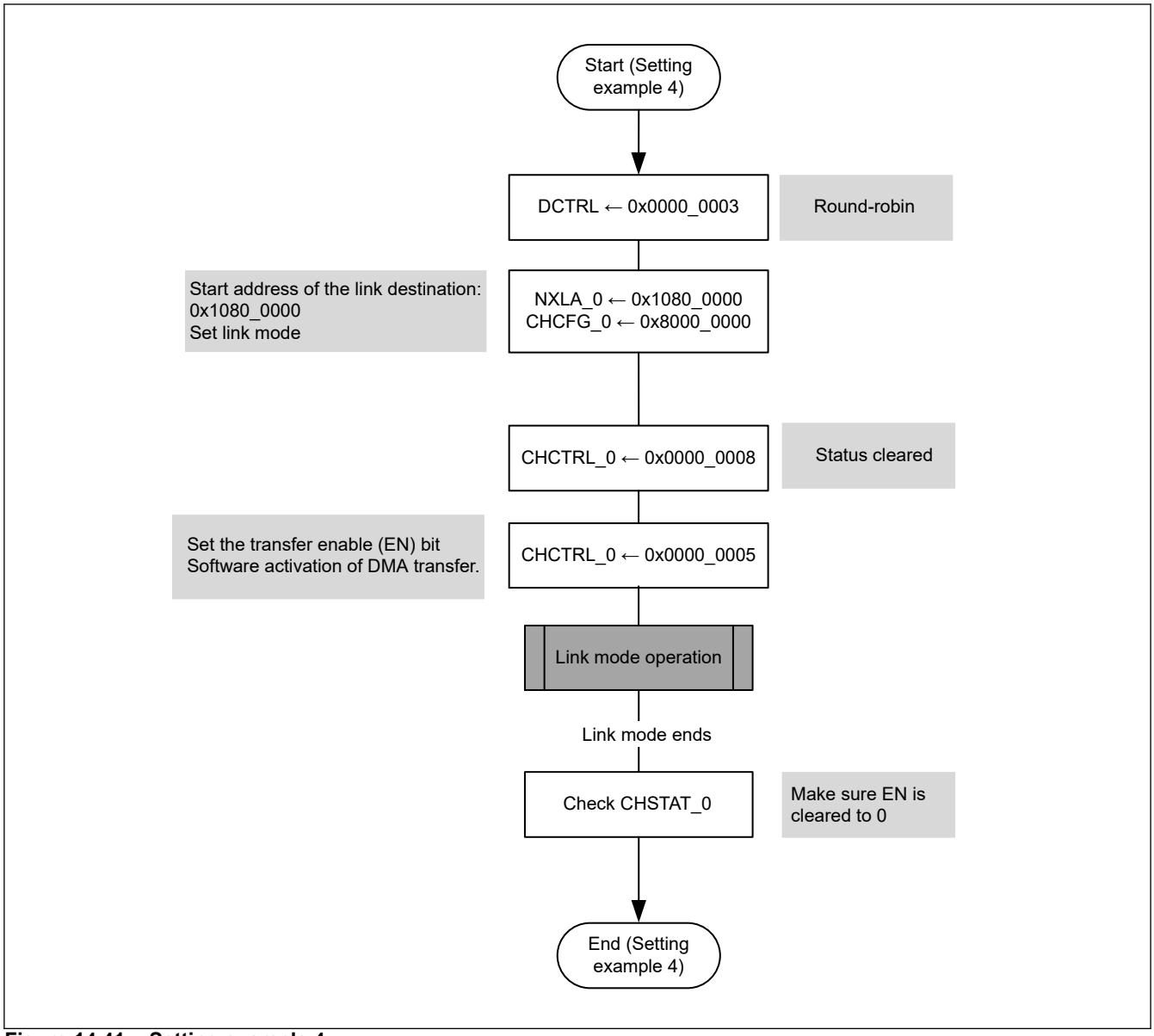


Figure 14.41 Setting example 4

14.5.5 Next Register Continuous Execution Settings

Figure 14.42 shows the flow when using two Next register sets to continue DMA transfers in register mode.

While performing a DMA transfer of a Next register, set the other Next register, and perform the DMA transfers in series.

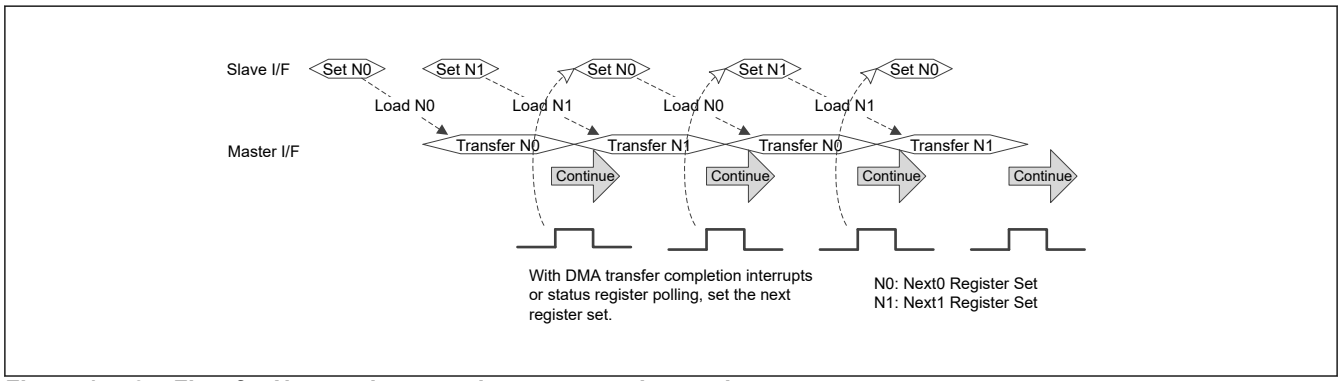


Figure 14.42 Flow for Next register continuous execution setting

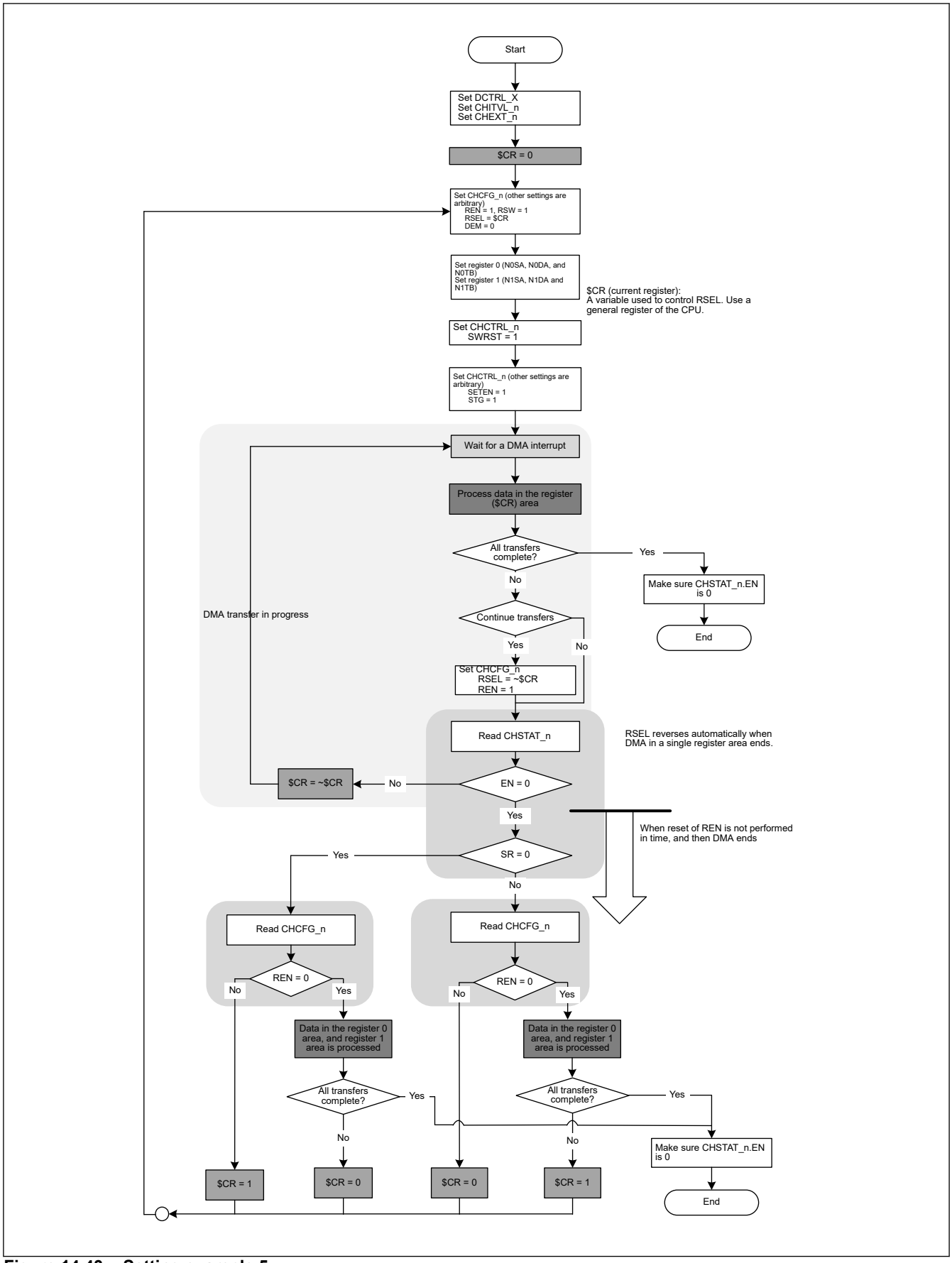


Figure 14.43 Setting example 5

(1) Supplementary information

Save the register sets 0 (N0SA_n, N0DA_n, and N0TB_n registers) and 1 (N1SA_n, N1DA_n, and N1TB_n registers) in a general register (call this register value $CR for convenience).

Every time a DMA transfer of a register set completes (a DMA transfer completion interrupt is generated), the REN bit in the CHCFG_n register is cleared to 0 automatically. To perform transfers in series, write 1 to the REN bit in the CHCFG_n register.

In this mode, two Next registers are executed in series. However, if the REN bit is not set before a DMA transfer completes (before the next DMA transfer completion interrupt is generated), the continuous execution stops. In this case, by reading the SR and EN bits in the CHSTAT_n register, and the REN bit in the CHCFG_n register, you can check how far the transfer has performed. To resume the transfer, perform the procedure described in Figure 14.43.

14.6 Usage Notes

This section provides notes on the DMA Controller module.

- When a transfer of which source and destination are the same, or partially shared area is performed, consistency of data cannot be guaranteed. Therefore, do not perform a transfer of which source and destination address areas overlap.
- For hardware activation, if REQD = 1 (the transfer destination issues a hardware request), SBE = 1 (flush mode) then the forced ejection function cannot be used. The transfer is deactivated by hardware.

14.6.1 When the DACK Signal is Divided and Output

When 4 bytes or more of data is transferred to an 8-bit or 16-bit external device, or 2 bytes or more of data is transferred to an 8-bit external device, a DMA transfer unit is divided into multiple bus cycles. The TEND signal outputs after the last DMA access completion.

When a DMA transfer is divided into multiple bus cycles and the CS signal is deactivated between bus cycles, DACK output is divided in the same way as the CS signal.

Figure 14.44 provides an example of DACK/TEND.

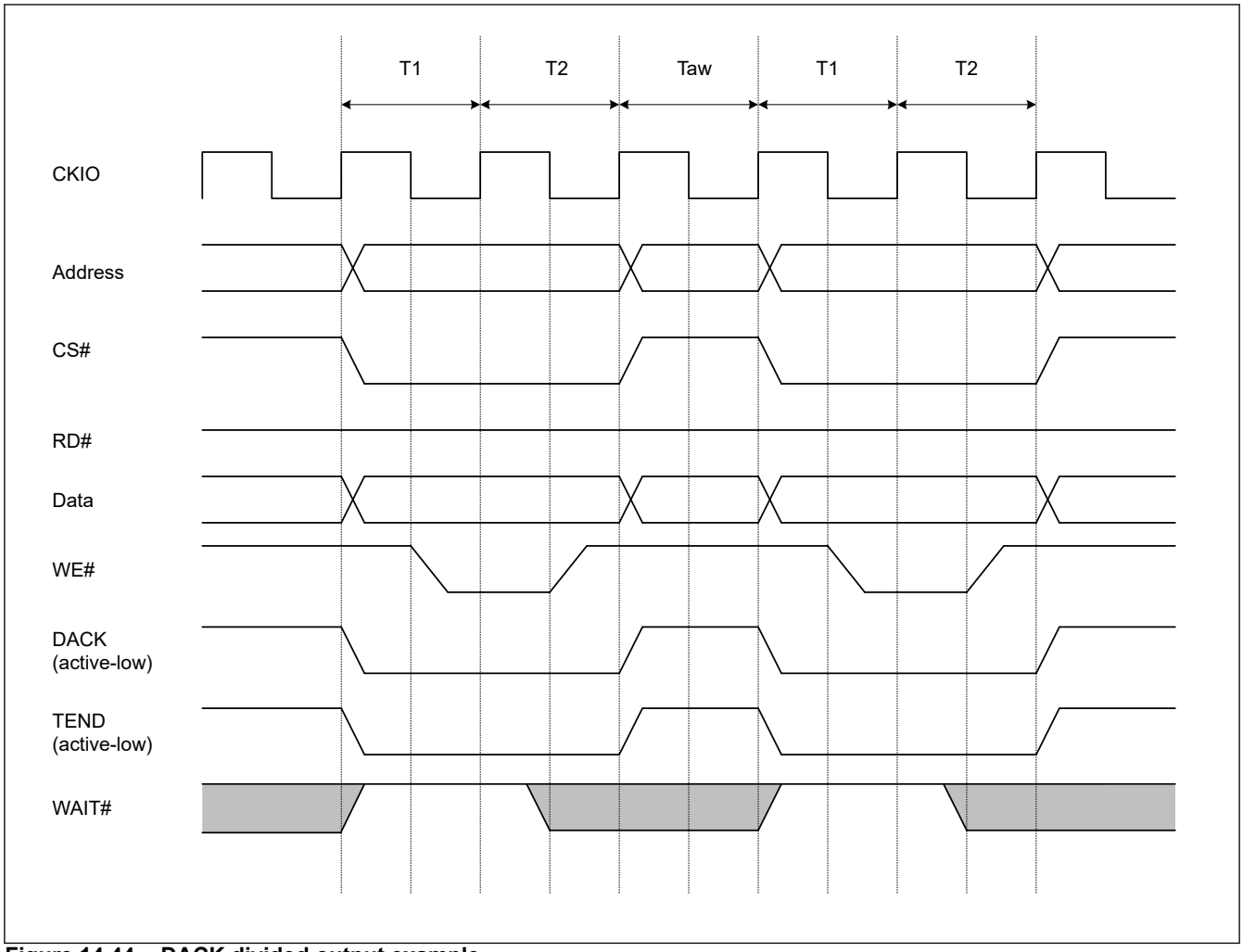


Figure 14.44 DACK divided output example

15. Event Link Controller (ELC)

15.1 Overview

The event link controller (ELC) connects (links) events generated by various peripheral modules to other modules. Event link allows direct cooperation among modules without CPU intervention.

Table 15.1 lists the specifications of the ELC and Figure 15.1 shows a block diagram.

Table 15.1 ELC specifications

Parameter	Specifications
Event link function	• 214 types of event signals can be directly connected to modules. • The operation of timer modules can be selected when an event is input to the timer module. • Event link operation is possible for Port 16 and Port 18. Single port*1: An event link can be set for a single bit specified in a port. Port group*1: An event link can be set for a group of single bits specified within eight I/O ports.

Note 1. The single port and port group specified as the input generate an event according to the change in the connected signal value.

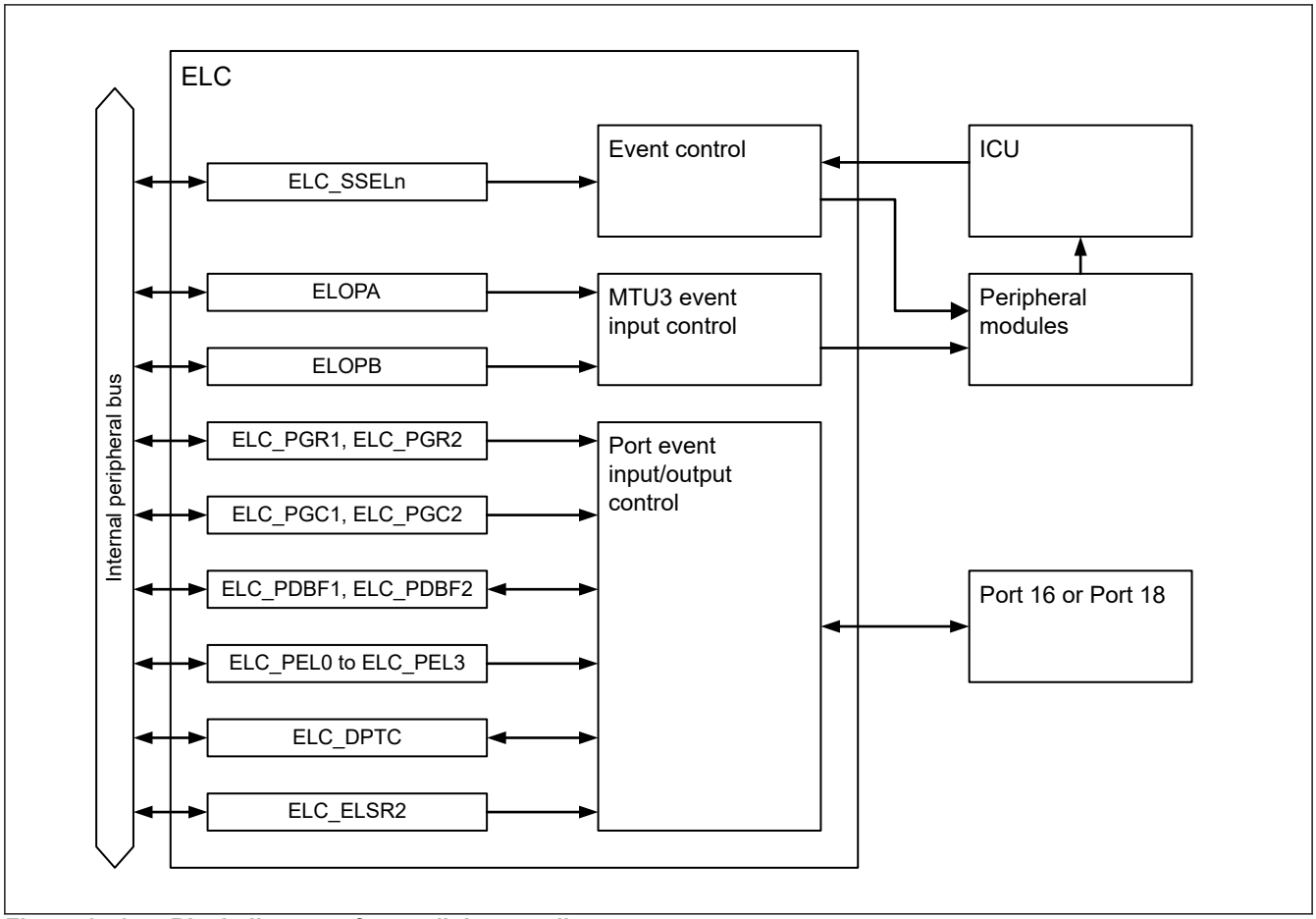


Figure 15.1 Block diagram of event link controller

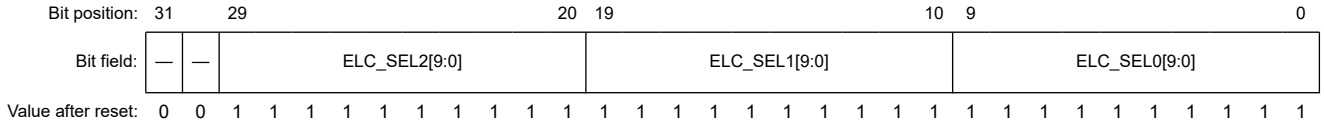
15.2 Register Descriptions

This section describes details of the register functions.

15.2.1 ELC_SSELn : ELC Event Source Select Register n (n = 0 to 18)

Base address: ELC = 0x8009_0010

Offset address: 0x00 + 0x04 × n



Bit	Symbol	Function	R/W
9:0	ELC_SEL0[9:0]	Set the number for ELC event source to be linked to the ELC destination. Event source is selectable from Table 15.2. If ELC function is not used, set 0x3FF (Disable ELC function). Setting of the number not listed in Table 15.2 is prohibited. See Table 15.3 for correspondence between register/bit assignment and ELC destination.	R/W
19:10	ELC_SEL1[9:0]		R/W
29:20	ELC_SEL2[9:0]		R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

The ELC_SSELn register specifies an event source to be linked to for each destination peripheral module. Table 15.2 lists the correspondence between the event source number set in the ELC_SSELn register and the event source description. Table 15.3 lists the correspondence between the ELC_SSELn register/bits and the destination peripheral modules.

Table 15.2 ELC Event Source (1 of 6)

Event Source No.	Peripheral module	Event source description
0x000	ICU	Software interrupt 0
0x001		Software interrupt 1
0x002		Software interrupt 2
0x003		Software interrupt 3
0x004		Software interrupt 4
0x005		Software interrupt 5
0x015	DMAC0	DMAC0 transfer completion ch0
0x016		DMAC0 transfer completion ch1
0x017		DMAC0 transfer completion ch2
0x018		DMAC0 transfer completion ch3
0x019		DMAC0 transfer completion ch4
0x01A		DMAC0 transfer completion ch5
0x01B		DMAC0 transfer completion ch6
0x01C		DMAC0 transfer completion ch7
0x025	DMAC1	DMAC1 transfer completion ch0
0x026		DMAC1 transfer completion ch1
0x027		DMAC1 transfer completion ch2
0x028		DMAC1 transfer completion ch3
0x029		DMAC1 transfer completion ch4
0x02A		DMAC1 transfer completion ch5
0x02B		DMAC1 transfer completion ch6
0x02C		DMAC1 transfer completion ch7

Table 15.2 ELC Event Source (2 of 6)

Event Source No.	Peripheral module	Event source description
0x045	MTU0	MTU0.TGRA input capture/compare match
0x046		MTU0.TGRB input capture/compare match
0x047		MTU0.TGRC input capture/compare match
0x048		MTU0.TGRD input capture/compare match
0x049		MTU0.TCNT overflow
0x04A		MTU0.TGRE compare match
0x04B		MTU0.TGRF compare match
0x054	MTU3	MTU3.TGRA input capture/compare match
0x055		MTU3.TGRB input capture/compare match
0x056		MTU3.TGRC input capture/compare match
0x057		MTU3.TGRD input capture/compare match
0x058		MTU3.TCNT overflow
0x059	MTU4	MTU4.TGRA input capture/compare match
0x05A		MTU4.TGRB input capture/compare match
0x05B		MTU4.TGRC input capture/compare match
0x05C		MTU4.TGRD input capture/compare match
0x061	MTU6	MTU6.TGRA input capture/compare match
0x062		MTU6.TGRB input capture/compare match
0x063		MTU6.TGRC input capture/compare match
0x064		MTU6.TGRD input capture/compare match
0x065		MTU6.TCNT overflow
0x066	MTU7	MTU7.TGRA input capture/compare match
0x067		MTU7.TGRB input capture/compare match
0x068		MTU7.TGRC input capture/compare match
0x069		MTU7.TGRD input capture/compare match
0x074	GPT0 (LLPP)	GPT0 GTCCRA input capture/compare match
0x075		GPT0 GTCCRB input capture/compare match
0x076		GPT0 GTCCRC compare match
0x077		GPT0 GTCCRD compare match
0x078		GPT0 GTCCRE compare match
0x079		GPT0 GTCCRF compare match
0x07A		GPT0 GTCNT overflow (GTPR compare match)
0x07B		GPT0 GTCNT underflow
0x07D	GPT1 (LLPP)	GPT1 GTCCRA input capture/compare match
0x07E		GPT1 GTCCRB input capture/compare match
0x07F		GPT1 GTCCRC compare match
0x080		GPT1 GTCCRD compare match
0x081		GPT1 GTCCRE compare match
0x082		GPT1 GTCCRF compare match
0x083		GPT1 GTCNT overflow (GTPR compare match)
0x084		GPT1 GTCNT underflow

Table 15.2 ELC Event Source (3 of 6)

Event Source No.	Peripheral module	Event source description
0x086	GPT2 (LLPP)	GPT2 GTCCRA input capture/compare match
0x087		GPT2 GTCCRB input capture/compare match
0x088		GPT2 GTCCRC compare match
0x089		GPT2 GTCCRD compare match
0x08A		GPT2 GTCCRE compare match
0x08B		GPT2 GTCCRF compare match
0x08C		GPT2 GTCNT overflow (GTPR compare match)
0x08D		GPT2 GTCNT underflow
0x08F	GPT3 (LLPP)	GPT3 GTCCRA input capture/compare match
0x090		GPT3 GTCCRB input capture/compare match
0x091		GPT3 GTCCRC compare match
0x092		GPT3 GTCCRD compare match
0x093		GPT3 GTCCRE compare match
0x094		GPT3 GTCCRF compare match
0x095		GPT3 GTCNT overflow (GTPR compare match)
0x096		GPT3 GTCNT underflow
0x098	GPT4 (LLPP)	GPT4 GTCCRA input capture/compare match
0x099		GPT4 GTCCRB input capture/compare match
0x09A		GPT4 GTCCRC compare match
0x09B		GPT4 GTCCRD compare match
0x09C		GPT4 GTCCRE compare match
0x09D		GPT4 GTCCRF compare match
0x09E		GPT4 GTCNT overflow (GTPR compare match)
0x09F		GPT4 GTCNT underflow
0x0A1	GPT5 (LLPP)	GPT5 GTCCRA input capture/compare match
0x0A2		GPT5 GTCCRB input capture/compare match
0x0A3		GPT5 GTCCRC compare match
0x0A4		GPT5 GTCCRD compare match
0x0A5		GPT5 GTCCRE compare match
0x0A6		GPT5 GTCCRF compare match
0x0A7		GPT5 GTCNT overflow (GTPR compare match)
0x0A8		GPT5 GTCNT underflow
0x0AA	GPT6 (LLPP)	GPT6 GTCCRA input capture/compare match
0x0AB		GPT6 GTCCRB input capture/compare match
0x0AC		GPT6 GTCCRC compare match
0x0AD		GPT6 GTCCRD compare match
0x0AE		GPT6 GTCCRE compare match
0x0AF		GPT6 GTCCRF compare match
0x0B0		GPT6 GTCNT overflow (GTPR compare match)
0x0B1		GPT6 GTCNT underflow

Table 15.2 ELC Event Source (4 of 6)

Event Source No.	Peripheral module	Event source description
0x0B3	GPT7 (NONSAFETY)	GPT7 GTCCRA input capture/compare match
0x0B4		GPT7 GTCCRB input capture/compare match
0x0B5		GPT7 GTCCRC compare match
0x0B6		GPT7 GTCCRD compare match
0x0B7		GPT7 GTCCRE compare match
0x0B8		GPT7 GTCCRF compare match
0x0B9		GPT7 GTCNT overflow (GTPR compare match)
0x0BA		GPT7 GTCNT underflow
0x0BC	GPT8 (NONSAFETY)	GPT8 GTCCRA input capture/compare match
0x0BD		GPT8 GTCCRB input capture/compare match
0x0BE		GPT8 GTCCRC compare match
0x0BF		GPT8 GTCCRD compare match
0x0C0		GPT8 GTCCRE compare match
0x0C1		GPT8 GTCCRF compare match
0x0C2		GPT8 GTCNT overflow (GTPR compare match)
0x0C3		GPT8 GTCNT underflow
0x0C5	GPT9 (NONSAFETY)	GPT9 GTCCRA input capture/compare match
0x0C6		GPT9 GTCCRB input capture/compare match
0x0C7		GPT9 GTCCRC compare match
0x0C8		GPT9 GTCCRD compare match
0x0C9		GPT9 GTCCRE compare match
0x0CA		GPT9 GTCCRF compare match
0x0CB		GPT9 GTCNT overflow (GTPR compare match)
0x0CC		GPT9 GTCNT underflow
0x0CE	GPT10 (NONSAFETY)	GPT10 GTCCRA input capture/compare match
0x0CF		GPT10 GTCCRB input capture/compare match
0x0D0		GPT10 GTCCRC compare match
0x0D1		GPT10 GTCCRD compare match
0x0D2		GPT10 GTCCRE compare match
0x0D3		GPT10 GTCCRF compare match
0x0D4		GPT10 GTCNT overflow (GTPR compare match)
0x0D5		GPT10 GTCNT underflow
0x0D7	GPT11 (NONSAFETY)	GPT11 GTCCRA input capture/compare match
0x0D8		GPT11 GTCCRB input capture/compare match
0x0D9		GPT11 GTCCRC compare match
0x0DA		GPT11 GTCCRD compare match
0x0DB		GPT11 GTCCRE compare match
0x0DC		GPT11 GTCCRF compare match
0x0DD		GPT11 GTCNT overflow (GTPR compare match)
0x0DE		GPT11 GTCNT underflow

Table 15.2 ELC Event Source (5 of 6)

Event Source No.	Peripheral module	Event source description
0x0E0	GPT12 (NONSAFETY)	GPT12 GTCCRA input capture/compare match
0x0E1		GPT12 GTCCRB input capture/compare match
0x0E2		GPT12 GTCCRC compare match
0x0E3		GPT12 GTCCRD compare match
0x0E4		GPT12 GTCCRE compare match
0x0E5		GPT12 GTCCRF compare match
0x0E6		GPT12 GTCNT overflow (GTPR compare match)
0x0E7		GPT12 GTCNT underflow
0x0E9	GPT13 (NONSAFETY)	GPT13 GTCCRA input capture/compare match
0x0EA		GPT13 GTCCRB input capture/compare match
0x0EB		GPT13 GTCCRC compare match
0x0EC		GPT13 GTCCRD compare match
0x0ED		GPT13 GTCCRE compare match
0x0EE		GPT13 GTCCRF compare match
0x0EF		GPT13 GTCNT overflow (GTPR compare match)
0x0F0		GPT13 GTCNT underflow
0x10D	GMAC	GMAC PTP timer pulse output 0
0x10E		GMAC PTP timer pulse output 1
0x10F		GMAC PTP timer pulse output 2
0x110		GMAC PTP timer pulse output 3
0x113	HDSL	HDSL0 Safe position received interrupt
0x114		HDSL1 Safe position received interrupt
0x115	ESC	EtherCAT Sync0 interrupt
0x116		EtherCAT Sync1 interrupt
0x159	ADC120	ADC0 A/D scan end interrupt
0x15E	ADC121	ADC1 A/D scan end interrupt
0x174	MBXSEM	Mailbox (Host to Cortex-R52 interrupt 0)
0x175		Mailbox (Host to Cortex-R52 interrupt 1)
0x176		Mailbox (Host to Cortex-R52 interrupt 2)
0x177		Mailbox (Host to Cortex-R52 interrupt 3)
0x1CB	ADC120	ADC0 compare match
0x1CC		ADC0 compare mismatch
0x1CD	ADC121	ADC1 compare match
0x1CE		ADC1 compare mismatch
0x1CF	MTU4	MTU4.TCNT overflow
0x1D0		MTU4.TCNT underflow
0x1D1	MTU7	MTU7.TCNT overflow
0x1D2		MTU7.TCNT underflow
0x1D3	I/O ports	Input edge detection of input port group 1
0x1D4		Input edge detection of input port group 2
0x1D5		Input edge detection of single input port 0
0x1D6		Input edge detection of single input port 1
0x1D7		Input edge detection of single input port 2
0x1D8		Input edge detection of single input port 3

Table 15.2 ELC Event Source (6 of 6)

Event Source No.	Peripheral module	Event source description
0x1D9	GPT0 (LLPP)	GPT0 GTADTRA compare match (A/D Converter Start Request A)
0x1DA		GPT0 GTADTRB compare match (A/D Converter Start Request B)
0x1DB	GPT1 (LLPP)	GPT1 GTADTRA compare match (A/D Converter Start Request A)
0x1DC		GPT1 GTADTRB compare match (A/D Converter Start Request B)
0x1DD	GPT2 (LLPP)	GPT2 GTADTRA compare match (A/D Converter Start Request A)
0x1DE		GPT2 GTADTRB compare match (A/D Converter Start Request B)
0x1DF	GPT3 (LLPP)	GPT3 GTADTRA compare match (A/D Converter Start Request A)
0x1E0		GPT3 GTADTRB compare match (A/D Converter Start Request B)
0x1E1	GPT4 (LLPP)	GPT4 GTADTRA compare match (A/D Converter Start Request A)
0x1E2		GPT4 GTADTRB compare match (A/D Converter Start Request B)
0x1E3	GPT5 (LLPP)	GPT5 GTADTRA compare match (A/D Converter Start Request A)
0x1E4		GPT5 GTADTRB compare match (A/D Converter Start Request B)
0x1E5	GPT6 (LLPP)	GPT6 GTADTRA compare match (A/D Converter Start Request A)
0x1E6		GPT6 GTADTRB compare match (A/D Converter Start Request B)
0x1E7	GPT7 (NONSAFETY)	GPT7 GTADTRA compare match (A/D Converter Start Request A)
0x1E8		GPT7 GTADTRB compare match (A/D Converter Start Request B)
0x1E9	GPT8 (NONSAFETY)	GPT8 GTADTRA compare match (A/D Converter Start Request A)
0x1EA		GPT8 GTADTRB compare match (A/D Converter Start Request B)
0x1EB	GPT9 (NONSAFETY)	GPT9 GTADTRA compare match (A/D Converter Start Request A)
0x1EC		GPT9 GTADTRB compare match (A/D Converter Start Request B)
0x1ED	GPT10 (NONSAFETY)	GPT10 GTADTRA compare match (A/D Converter Start Request A)
0x1EE		GPT10 GTADTRB compare match (A/D Converter Start Request B)
0x1EF	GPT11 (NONSAFETY)	GPT11 GTADTRA compare match (A/D Converter Start Request A)
0x1F0		GPT11 GTADTRB compare match (A/D Converter Start Request B)
0x1F1	GPT12 (NONSAFETY)	GPT12 GTADTRA compare match (A/D Converter Start Request A)
0x1F2		GPT12 GTADTRB compare match (A/D Converter Start Request B)
0x1F3	GPT13 (NONSAFETY)	GPT13 GTADTRA compare match (A/D Converter Start Request A)
0x1F4		GPT13 GTADTRB compare match (A/D Converter Start Request B)
0x3FF	—	Disable ELC function

Table 15.3 Correspondence between the ELC_SSELn register and ELC destination (1 of 2)

Register Name	Bit Name	ELC destination	Register Name	Bit Name	ELC destination
ELC_SSEL0	ELC_SEL0	MTU0	ELC_SSEL10	ELC_SEL0	DSMIF0 CDCNT_INT_TRG1
	ELC_SEL1	MTU3		ELC_SEL1	DSMIF0 CDCNT_INT_TRG2
	ELC_SEL2	MTU4		ELC_SEL2	DSMIF1 CAP_TRG0
ELC_SSEL1	ELC_SEL0	LLPP GPT event A	ELC_SSEL11	ELC_SEL0	DSMIF1 CAP_TRG1
	ELC_SEL1	LLPP GPT event B		ELC_SEL1	DSMIF1 CAP_TRG2
	ELC_SEL2	LLPP GPT event C		ELC_SEL2	DSMIF1 CAP_TRG3
ELC_SSEL2	ELC_SEL0	LLPP GPT event D	ELC_SSEL12	ELC_SEL0	DSMIF1 CAP_TRG4
	ELC_SEL1	LLPP GPT event E		ELC_SEL1	DSMIF1 CAP_TRG5
	ELC_SEL2	LLPP GPT event F		ELC_SEL2	DSMIF1 CDCNT_INT_TRG0
ELC_SSEL3	ELC_SEL0	LLPP GPT event G	ELC_SSEL13	ELC_SEL0	DSMIF1 CDCNT_INT_TRG1
	ELC_SEL1	LLPP GPT event H		ELC_SEL1	DSMIF1 CDCNT_INT_TRG2
	ELC_SEL2	NONSAFTY GPT event A		ELC_SEL2	ENCIF TRG0

Table 15.3 Correspondence between the ELC_SSELn register and ELC destination (2 of 2)

Register Name	Bit Name	ELC destination	Register Name	Bit Name	ELC destination
ELC_SSEL4	ELC_SEL0	NONSAFTY GPT event B	ELC_SSEL14	ELC_SEL0	ENCIF TRG1
	ELC_SEL1	NONSAFTY GPT event C		ELC_SEL1	ESC LATCH0
	ELC_SEL2	NONSAFTY GPT event D		ELC_SEL2	ESC LATCH1
ELC_SSEL5	ELC_SEL0	NONSAFTY GPT event E	ELC_SSEL15	ELC_SEL0	GMAC PTP timer capture 0
	ELC_SEL1	NONSAFTY GPT event F		ELC_SEL1	GMAC PTP timer capture 1
	ELC_SEL2	NONSAFTY GPT event G		ELC_SEL2	Output port group 1 (Port 16)
ELC_SSEL6	ELC_SEL0	NONSAFTY GPT event H	ELC_SSEL16	ELC_SEL0	Output port group 2 (Port 18)
	ELC_SEL1	ADC0 A		ELC_SEL1	Input port group 1 (Port 16)
	ELC_SEL2	ADC0 B		ELC_SEL2	Input port group 2 (Port 18)
ELC_SSEL7	ELC_SEL0	ADC1 A	ELC_SSEL17	ELC_SEL0	Single port 0
	ELC_SEL1	ADC1 B		ELC_SEL1	Single port 1
	ELC_SEL2	DSMIF0 CAP_TRG0		ELC_SEL2	Single port 2
ELC_SSEL8	ELC_SEL0	DSMIF0 CAP_TRG1	ELC_SSEL18	ELC_SEL0	Single port 3
	ELC_SEL1	DSMIF0 CAP_TRG2		ELC_SEL1	ENCOUT trigger
	ELC_SEL2	DSMIF0 CAP_TRG3		ELC_SEL2	Reserved
ELC_SSEL9	ELC_SEL0	DSMIF0 CAP_TRG4	—		
	ELC_SEL1	DSMIF0 CAP_TRG5			
	ELC_SEL2	DSMIF0 CDCNT_INT_TRG0			

15.2.2 ELOPA : Event Link Option Setting Register A

Base address: ELO = 0x8028_1200

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	MTU3MD[1:0]		—	—	—	—	MTU0MD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
1:0	MTU0MD[1:0]	MTU0 Operation Select 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture (The MTU0.TCNT value is captured into MTU0.TGRA.) 1 1: Event is disabled.	R/W
5:2	—	These bits are read as 1. The write value should be 1.	R/W
7:6	MTU3MD[1:0]	MTU3 Operation Select 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture (The MTU3.TCNT value is captured into MTU3.TGRA.) 1 1: Event is disabled.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

The ELOPA register determines the operation of MTU0 and MTU3 when an event is input. The event setting should be disabled when the ELC function is not used.

15.2.3 ELOPB : Event Link Option Setting Register B

Base address: ELO = 0x8028_1200

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MTU4MD[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
1:0	MTU4MD[1:0]	MTU4 Operation Select 0 0: Counting is started. 0 1: Counting is restarted. 1 0: Input capture (The MTU4.TCNT value is captured into MTU4.TGRA.) 1 1: Event is disabled.	R/W
7:2	—	These bits are read as 1. The write value should be 1.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

The ELOPB register determines the operation of MTU4 when an event is input. The event setting should be disabled when the ELC function is not used.

15.2.4 ELC_PGRn : ELC Port Group Setting Register n (n = 1, 2)

Base address: PORT_NSR = 0x800A_0000
PORT_SR = 0x8103_0000

Offset address: 0xE00 + 0x01 × (n − 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PG7	PG6	PG5	PG4	PG3	PG2	PG1	PG0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PG7 to PG0	Port Group Setting 0: The port bit is not specified as a member of the same group. 1: The port bit is specified as a member of the same group.	R/W

The ELC_PGRn register specifies a group for I/O port bits. This register specifies each port bit in the same eight I/O ports as the member of a group. One to eight port bits can be specified as the members of the same group as required.

Table 15.4 Registers related to port groups and corresponding port numbers

Port Number	Port Group Setting Register (ELC_PGRn)	Port Group Control Register (ELC_PGCn)	Port Buffer Register (ELC_PDBFn)
Port 16	ELC_PGR1 register	ELC_PGC1 register	ELC_PDBF1 register
Port 18	ELC_PGR2 register	ELC_PGC2 register	ELC_PDBF2 register

15.2.5 ELC_PGCn : ELC Port Group Control Register n (n = 1, 2)

Base address: PORT_NSR = 0x800A_0000
PORT_SR = 0x8103_0000

Offset address: 0xE02 + 0x01 × (n − 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	PGCO[2:0]			—	PGCOVE	PGCI[1:0]	
Value after reset:	1	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
1:0	PGCI[1:0]	Event Output Edge Select 0 0: Event is generated on detection of the rising edge of the external input signal. 0 1: Event is generated on detection of the falling edge of the external input signal. 1 x: Event is generated on detection of both the rising and falling edges of the external input signal.	R/W
2	PGCOVE	PDBF Overwrite 0: Overwriting ELC_PDBFn register is disabled. 1: Overwriting ELC_PDBFn register is enabled.	R/W
3	—	This bit is read as 1. The write value should be 1.	R/W
6:4	PGCO[2:0]	Port Group Operation Select 0 0 0: 0 is output when the event is input. 0 0 1: 1 is output when the event is input. 0 1 0: The toggled (inverted) value is output when the event is input. The target Port m Mode register should be set to 11b: Output (Output data is input to input buffer) 0 1 1: The buffer value is output when the event is input. 1 x x: Setting prohibited	R/W
7	—	This bit is read as 1. The write value should be 1.	R/W

For the output port group, the ELC_PGCn register specifies the form of outputting the signal externally via the port when the event signal is input. For the input port group, this register enables/disables overwriting of the ELC_PDBFn register and specifies the conditions of event generation (edge of the externally input signal).

15.2.6 ELC_PDBFn : ELC Port Buffer Register n (n = 1, 2)

Base address: PORT_NSR = 0x800A_0000
PORT_SR = 0x8103_0000

Offset address: 0xE04 + 0x04 × (n − 1)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PB7 to PB0	Port Buffer n ● In case Port is output Data is transferred from ELC_PDBFn register to Pm register when an event is input. ● In case Port is input External input data is transferred to ELC_PDBFn register when an event is input. Write access to the bit specified as a member of the input port group is invalid.	R/W

The ELC_PDBFn register is an 8-bit readable/writable register used in combination with the ELC_PGRn register.

15.2.7 ELC_PELn : ELC Port Setting Register n (n = 0 to 3)

Base address: PORT_NSR = 0x800A_0000
PORT_SR = 0x8103_0000

Offset address: 0xE0C+ 0x01 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	PSM[1:0]		PSP[1:0]		PSB[2:0]		
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	PSB[2:0]	Bit Number Specification A bit number in eight I/O ports is specified.	R/W
4:3	PSP[1:0]	Port Number Specification 0 0: Setting prohibited 0 1: Port 16 (corresponding to ELC_PGR1) 1 0: Port 18 (corresponding to ELC_PGR2) 1 1: Setting prohibited	R/W
6:5	PSM[1:0]	Event Link Specification For the output port, data to be output from the port is specified. For the input port, the edge on which the event is to be output is specified. 0 0: For the output port, 0 is output when the event is input. For the input port, the event is output on detection of the rising edge. 0 1: For the output port, 1 is output when the event is input. For the input port, the event is output on detection of the falling edge. 1 x: For the output port, the toggled (inverted) value is output when the event is input. For the input port, the event is output on detection of both the rising and falling edges. The target Port m Mode register should be set to 11b: Output (Output data is input to input buffer)	R/W
7	—	This bit is read as 1. The write value should be 1.	R/W

The ELC_PELn register specifies the single port to which an event is to be linked, the port operation on the event signal input, and the conditions of event generation. In this LSI, a total of 4 bits in Port 16 or Port 18 can be specified as single ports.

15.2.8 ELC_DPTC : ELC Edge Detection Control Register

Base address: PORT_NSR = 0x800A_0000
PORT_SR = 0x8103_0000

Offset address: 0xE10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	PTC3	PTC2	PTC1	PTC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	PTC3 to PTC0	Single Input Port n Edge Detection 0: Disable edge detection for Single Input Port n 1: Enable edge detection for Single Input Port n	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

The ELC_DPTC register specifies input edge detection function for single input port linked to the event.

15.2.9 ELC_ELSR2 : ELC Port Event Control Register

Base address: PORT_NSR = 0x800A_0000
PORT_SR = 0x8103_0000

Offset address: 0xE11

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PES3	PES2	PES1	PES0	PEG2	PEG1	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
3:2	PEG2 to PEG1	ELC Port Buffer Register (ELC_PDBFn) write access control. When set to 1, writing to the ELC_PDBFn register via Internal peripheral bus is disabled, preventing overwriting. 0: When not use event link function of input port group n 1: When use event link function of input port group n	R/W
7:4	PES3 to PES0	Single Port n Event Link Function Enable 0: Disable Single Port event link function 1: Enable Single Port event link function	R/W

The ELC_ELSR2 register specifies input port function linked to ELC. When enabled in this register, corresponding ELC_SSELn should be enabled.

15.3 Operation

15.3.1 Event Linkage

ELC connects events generated by various source peripheral modules to other destination modules. It allows direct interlinked operation among source and destination modules without using software.

When source events are specified in the ELC_SSELn registers, the corresponding destination peripheral modules can be operated at generation of the specified events. A single destination peripheral module can link only with a single event source. Set the ELC_SSELn register after completing the initialization of the destination peripheral module to operate by an event. Table 15.5 lists the operations of destination peripheral modules when an event signal is input. When ELC is not used in the destination peripheral module, set 0x3FF to the corresponding ELC_SSELn register.

Table 15.5 ELC destination and operation (1 of 3)

No	ELC destination	Operation
0	MTU0	The following operations can be selected by setting the ELOPA and ELOPB registers: ● Starts counting when an event signal is input ● Restarts counting when an event signal is input ● Performs input-capture operation when an event signal is input
1	MTU3	
2	MTU4	
3	LLPP GPT event A	The following operations can be selected by setting GPT registers in LLPP: ● Starts counting when an event signal is input ● Stops counting when an event signal is input ● Clears counter when an event signal is input ● Increments counter when an event signal is input ● Decrements counter when an event signal is input ● Performs input-capture operation when an event signal is input
4	LLPP GPT event B	
5	LLPP GPT event C	
6	LLPP GPT event D	
7	LLPP GPT event E	
8	LLPP GPT event F	
9	LLPP GPT event G	
10	LLPP GPT event H	

Table 15.5 ELC destination and operation (2 of 3)

No	ELC destination	Operation
11	NONSAFTY GPT event A	The following operations can be selected by setting GPT registers in NONSAFETY: ● Starts counting when an event signal is input ● Stops counting when an event signal is input ● Clears counter when an event signal is input ● Increments counter when an event signal is input ● Decrements counter when an event signal is input ● Performs input-capture operation when an event signal is input
12	NONSAFTY GPT event B	
13	NONSAFTY GPT event C	
14	NONSAFTY GPT event D	
15	NONSAFTY GPT event E	
16	NONSAFTY GPT event F	
17	NONSAFTY GPT event G	
18	NONSAFTY GPT event H	
19	ADC0 A	Starts A/D conversion when an event signal is input
20	ADC0 B	
21	ADC1 A	
22	ADC1 B	
23	DSMIF0 CAP_TRG0	Current Data capture trigger
24	DSMIF0 CAP_TRG1	
25	DSMIF0 CAP_TRG2	
26	DSMIF0 CAP_TRG3	
27	DSMIF0 CAP_TRG4	
28	DSMIF0 CAP_TRG5	
29	DSMIF0 CDCNT_INT_TRG0	Decimation dividing counter initialization trigger
30	DSMIF0 CDCNT_INT_TRG1	
31	DSMIF0 CDCNT_INT_TRG2	
32	DSMIF1 CAP_TRG0	Current Data capture trigger
33	DSMIF1 CAP_TRG1	
34	DSMIF1 CAP_TRG2	
35	DSMIF1 CAP_TRG3	
36	DSMIF1 CAP_TRG4	
37	DSMIF1 CAP_TRG5	
38	DSMIF1 CDCNT_INT_TRG0	Decimation dividing counter initialization trigger
39	DSMIF1 CDCNT_INT_TRG1	
40	DSMIF1 CDCNT_INT_TRG2	
41	ENCIF TRG0	Encoder interface subsystem trigger 0
42	ENCIF TRG1	Encoder interface subsystem trigger 1
43	ESC LATCH0	Capture current value of EtherCAT Slave Controller DC
44	ESC LATCH1	
45	GMAC PTP timer capture 0	Capture current value of PTP timer
46	GMAC PTP timer capture 1	
47	Output port group 1 (Port 16)	● Change the Pm value to the value specified in the ELC_PGCn register ● Transfer ELC_PDBFn value to the Pm register
48	Output port group 2 (Port 18)	
49	Input port group 1 (Port 16)	Transfers the signal value of the external pin to the ELC_PDBFn register
50	Input port group 2 (Port 18)	

Table 15.5 ELC destination and operation (3 of 3)

No	ELC destination	Operation
51	Single port 0	• In case single port is output Changes the Pm value to the value specified in the ELC_PELn register • In case single port is input Generates an event when edge detection on the single input port
52	Single port 1	
53	Single port 2	
54	Single port 3	
55	ENCOUT trigger	ENCOUT trigger

15.3.2 Operation of MTU When Event is Input

The operations are performed depending on the ELOPA and ELOPB registers when an event is input.

15.3.2.1 Count Start Operation

When an event is input, the timer starts counting, which sets the count start bit*1 in each timer control register to 1. An event that is input while the count start bit is 1 is invalid.

Note 1. Refer to the register descriptions on starting the timer in the relevant Peripheral Timer Module section.

15.3.2.2 Count Restart Operation

When an event is input, the timer counter is initialized. Since the count start bit*1 in each timer control register is retained, counting is restarted when an event signal is input while the count start bit is 1.

Note 1. Refer to the register descriptions on starting the timer in the relevant Peripheral Timer Module section.

15.3.2.3 Input Capture Operation

When an event is input, the timer performs input-capture operation.

15.3.3 Operation of GPT When Event is Input

Eight event signals of each GPT in LLPP and NONSAFETY are connected to every channel of each GPT in LLPP and NONSAFETY as GPT event sources A to H. To set GPT operation when inputting an event signal, enable any of the event sources A to H by a corresponding bit in registers listed in Table 15.6.

Table 15.6 Operations when event is input and corresponding source select registers

Operations on event	Register symbol	Register name
Count start	GTSSR	General PWM Timer Start Source Select Register
Count stop	GTPSR	General PWM Timer Stop Source Select Register
Counter clear	GTCSR	General PWM Timer Clear Source Select Register
Up-count	GTUPSR	General PWM Timer Count-Up Source Select Register
Down-count	GTDNSR	General PWM Timer Count-Down Source Select Register
Input capture A	GTICASR	General PWM Timer Input Capture Source Select Register A
Input capture B	GTICBSR	General PWM Timer Input Capture Source Select Register B

15.3.4 Operation of A/D Converter When Event is Input

The A/D converter starts A/D conversion when the ADCSR.ADST bit*1 is set to 1.

Note 1. For details, see section 39, 12-Bit A/D converter (ADC12).

15.3.5 I/O Port Operation on Event Input and Event Generation

The I/O port operation to be performed on event input to the port and the operation causing the port to generate an event can be set.

15.3.5.1 Single Ports and Port Groups

There are two event link modes:

- Event link to single ports
- Event link to port groups

In the former mode, events can be connected to eight I/O ports. In the latter mode, events can be connected to port groups consisting of any two or more bits in the same eight I/O ports.

A single port can be set by specifying any bit in the I/O port*1 to which an event can be connected using the ELC_PELn register. A port group can be set by specifying any one or more bits in the I/O port*1 to which an event can be connected using the ELC_PGRn register. One input port group and one output port group can be set in the same I/O port.

If the I/O port bit is specified as both a single port and a member of a port group, both functions are enabled when the relevant port is input, whereas only the port group function is enabled when the relevant port is output.

Set the PMm register to select the direction of the I/O ports.

Note 1. Port 16 and port 18

15.3.5.2 Single Input Port Operation on Event Generation

A single input port which is specified by the PMm register generates an event when the signal value of the external pin connected to the relevant port changes. The event generation condition is specified using the ELC_PELn register. An example of event linking operation by a single input port is shown as [1] in Figure 15.2.

15.3.5.3 Single Output Ports Operation on Event Input

When an event is input to a single output port which is specified by the PMm register, the signal of the external pin connected to the relevant port changes according to the settings of the ELC_PELn register. This changes the signal value of the external pin connected to the relevant port. An example of event linking operation by a single output port is shown as [2] in Figure 15.2.

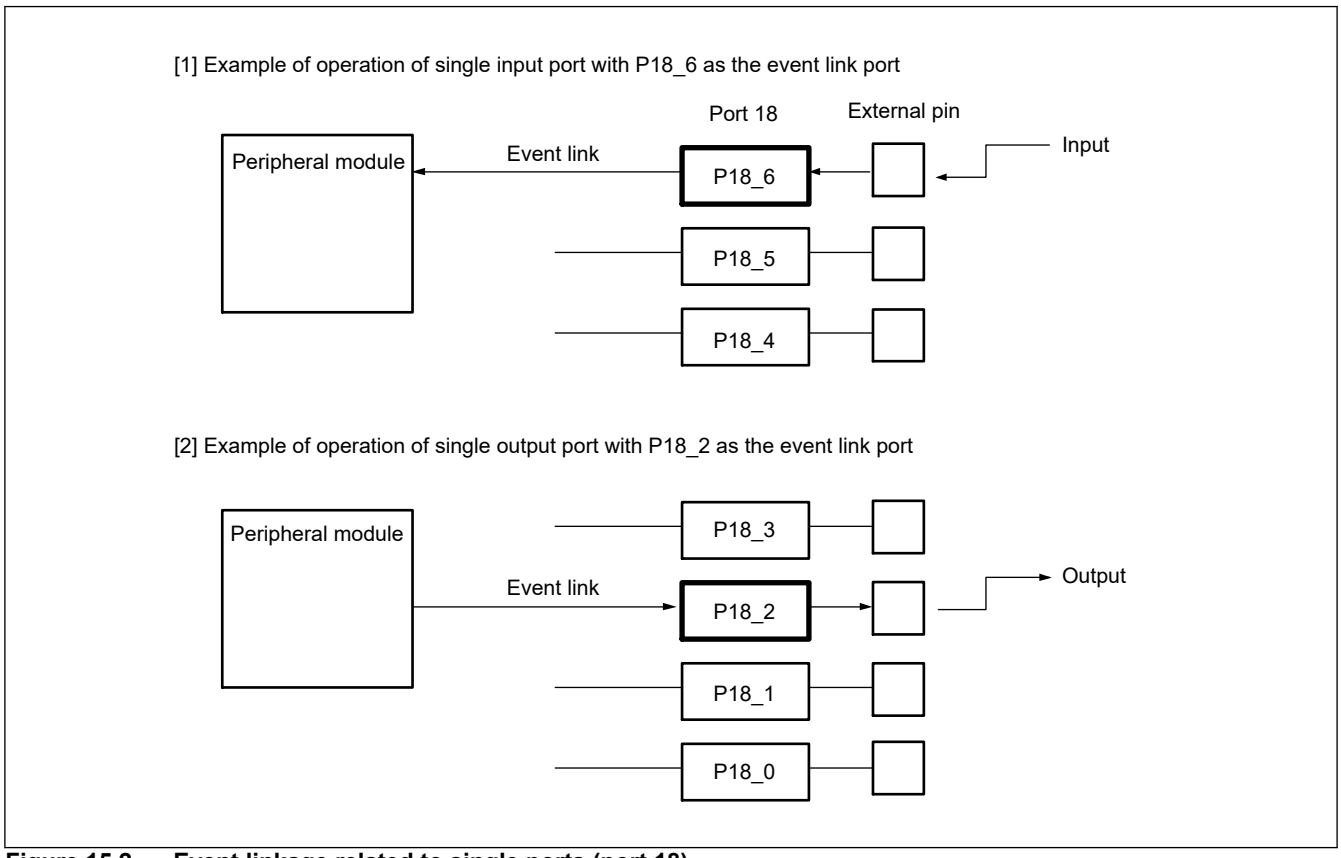


Figure 15.2 Event linkage related to single ports (port 18)

15.3.5.4 Input Port Group Operation on Event Generation

An input port group which is specified by the PMm register generates an event when the signal value of any one of the external pins connected to the relevant port group changes. The event generation condition is specified using the PGCIn bit of the ELC_PGCn register.

15.3.5.5 Input Port Group Operation on Event Input

When an event is input to an input port group, the signal value of the external pin on event input is transferred to the ELC_PDBFn register. In this case, only the values of the bits specified as members of the input port group are transferred. If another event is input to the input port group in this state, operations are performed depending on the ELC_PGCn.PGCOVE bit setting as described below. An example of operation is shown in Figure 15.3.

- ELC_PGCn.PGCOVE = 0 (overwriting is disabled)
  If the value that was transferred to the ELC_PDBFn register on the previous event input has already been read by the CPU, the signal value of the external pin is transferred to the ELC_PDBFn register. If not read, the signal value of the external pin is not transferred and the input event is invalid.
- ELC_PGCn.PGCOVE = 1 (overwriting is enabled)
  When another event is input to an input port group, the signal value of the external pin is transferred to the ELC_PDBFn register.

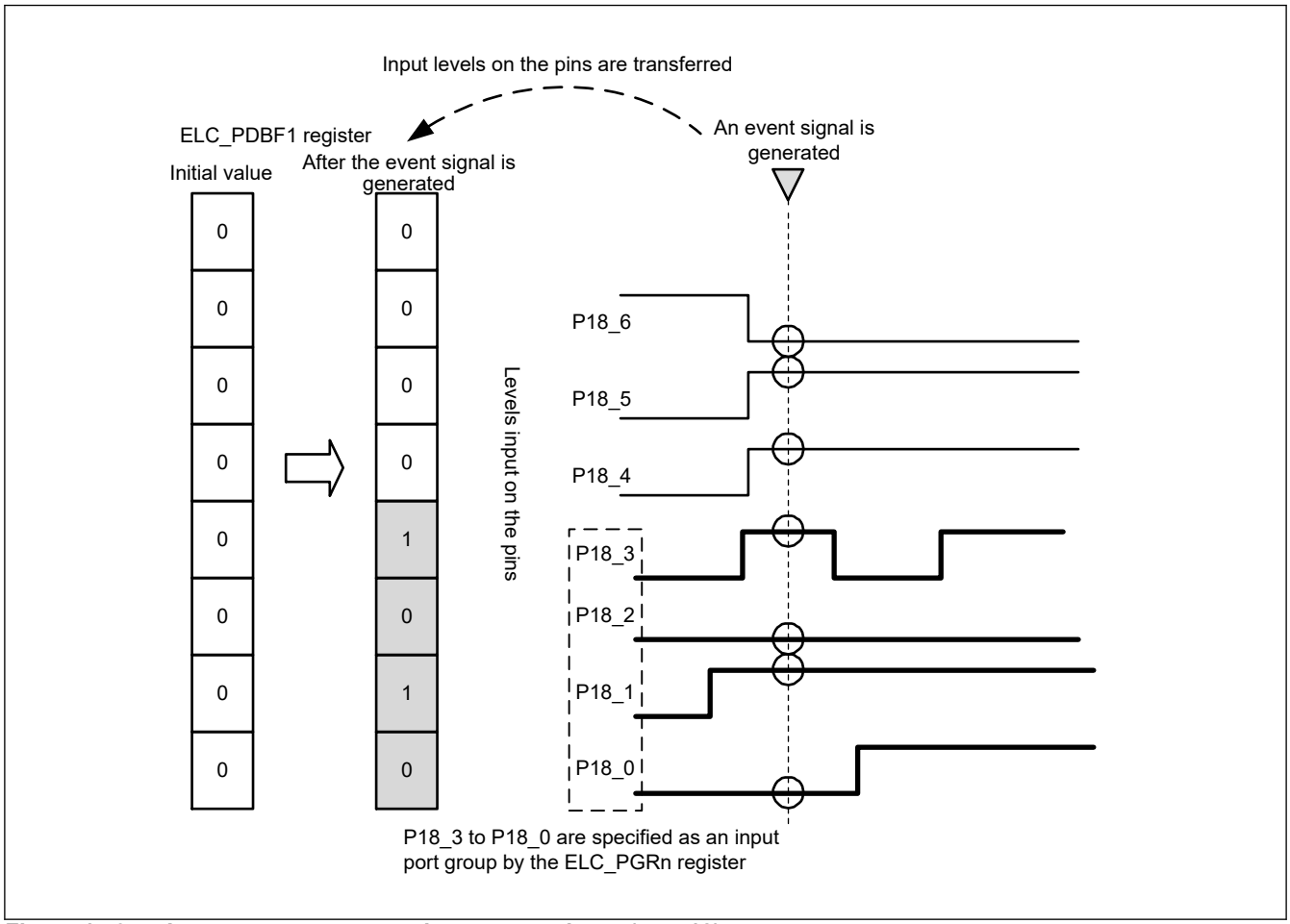


Figure 15.3 Input port group operation on event input (port 18)

15.3.5.6 Output Port Group Operation on Event Input

If an event is input to an output port group while the ELC_PGCn.PGCOn bit is 000b, 001b, or 010b, the Pm register value is changed to the value which was specified in the ELC_PGCn register.

If an event is input to an output port group while the ELC_PGCn.PGCOn bit is 011b, the ELC_PDBFn value is transferred to the Pm register of the port which was specified in the ELC_PGRn register. Example of operation of the output port group on an event input (when ELC_PGCn.PGCOn = 011b) is shown in Figure 15.4.

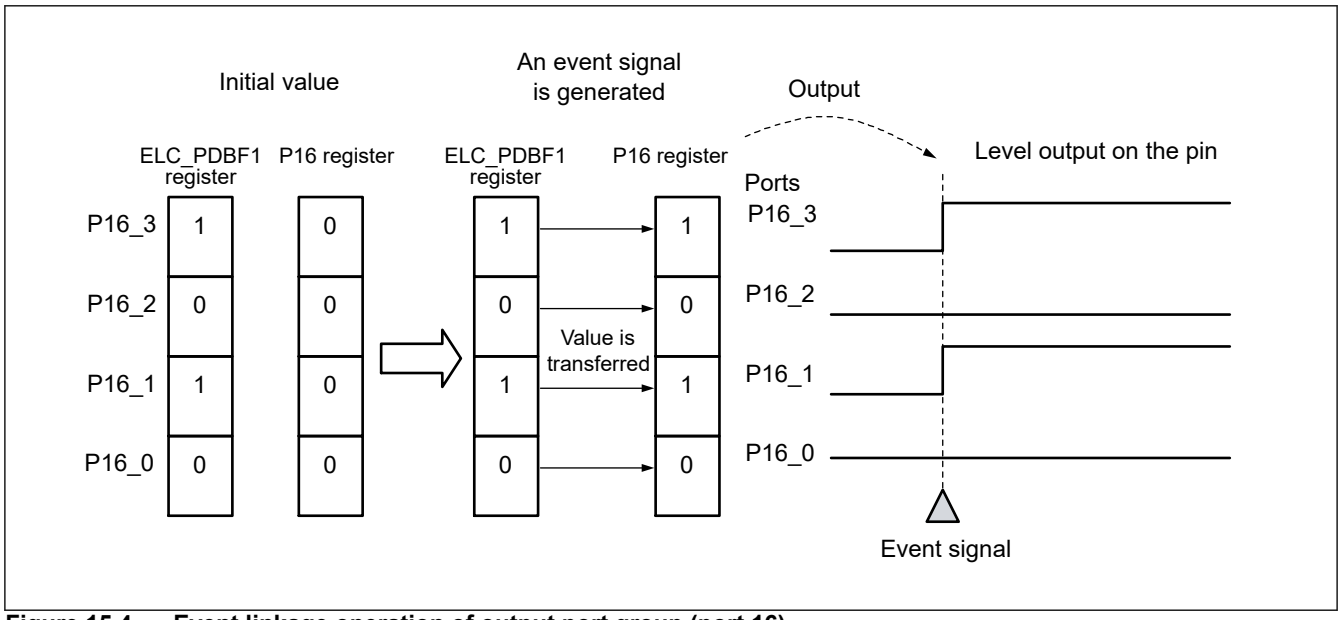


Figure 15.4 Event linkage operation of output port group (port 16)

15.3.5.7 Restrictions on Writing to Pm and ELC_PDBFn Registers by a CPU

For event linkage through the I/O ports, following restrictions apply when a CPU writes the Pm and ELC_PDBFn registers.

- If port bits are specified as members of the input port group, write access to the relevant bits in the ELC_PDBFn register is invalid.
- If port bits are specified as members of the output port group, write access to the relevant bits in the Pm register is invalid.
- If a port bit is specified as a single output port and the event linkage is set (by the ELC_SSELn register) for the port, write access to the relevant bit in the Pm register is invalid.

15.3.6 Example of Procedure for Linking Events

The following procedure describes the steps for linking events.

1. Set the operation of the module to which an event is to be linked.
2. If events are linked to ports, set the registers associated with the ports as follows:
   - I/O port setting
     Pm: Set the initial values of the output ports.
     PMm: Set the I/O direction of the ports.
   - ELC setting
     ELC_PGRn: If ports are used as a port group, set the ports (in bit units) to be grouped.
     ELC_PGCn: Set the operation of the port group.
     ELC_PELn: If ports are used as single ports, set the ports, the operation of the ports when an event is input, and the condition when an event is generated.
     ELC_DPTC: If pots are used as single input port edge detection, set the ports.
     ELC_ELSR2: If ports are used as input port groups, set the PEGn bits. If ports are used as single ports, set the PESn bits.
3. For the ELC_SSELn register associated with the module to which an event signal is to be linked, set the number of the event signal.
4. If events are to be linked to timer modules, set the ELOPA and ELOPB registers corresponding to the timers as required.

5. Set the operation of the module from which an event is output, and activate the module. This allows the event output from the module to start the module to which an event is linked as preset.
6. To stop event linkage of independent modules, set 0x3FF to ELC_SELx bits in the ELC_SSEL register corresponding to the modules.

Note: Setting the ELC_PDBFn register
- Output port groups
  Set the ELC_PGCn register before setting the ELC_PDBFn register. The value of the ELC_PGCn register can be modified if this precedes an event trigger which causes the value of the Pm register to change.
- Input port groups
  Setting the ELC_PDBFn is not required. However, since the value after a reset is 0x00, if the ELC_PDBFn register is used to confirm changes from H to L due to an event input, set the ELC_PDBFn bit for the pin for which you want to confirm this to be 1.

15.4 Usage Notes

15.4.1 Setting Clocks

To link events, it is necessary for the ELC and the related modules to be enabled. The modules cannot operate if the related modules are in the module-stop state or if the low power consumption mode causes the modules to stop (all-module stop mode).

16. I/O Ports

16.1 Overview

The I/O port pins operate as general I/O port pins, I/O pins for peripheral modules, or interrupt input pins. All pins are set to non-use immediately after a reset (Hi-Z input protection) excluding P02_4 to P02_7 and P17_5 which are shared with JTAG pins and reset output pin. The pin functions are switched by register settings. The I/O ports and peripheral modules for each pin are specified in the associated registers.

The configuration of the I/O ports differs depending on the package.

Table 16.1 shows the specifications of I/O ports.

Table 16.1 Specification of I/O ports

Port	Package	
	196-pin FBGA	Number of pins
Port 00	P00_0 to P00_3, P00_6	5
Port 01	P01_0 to P01_7	8
Port 02	P02_0 to P02_7	8
Port 03	—	0
Port 04	P04_1	1
Port 05	P05_5 to P05_7	3
Port 06	P06_0 to P06_7	8
Port 07	P07_0 to P07_4	5
Port 08	P08_4 to P08_7	4
Port 09	P09_0 to P09_7	8
Port 10	P10_0 to P10_4	5
Port 11	—	0
Port 12	—	0
Port 13	P13_2 to P13_7	6
Port 14	P14_0 to P14_7	8
Port 15	P15_0 to P15_7	8
Port 16	P16_0 to P16_3	4
Port 17	P17_0, P17_3 to P17_7	6
Port 18	P18_0 to P18_6	7
Port 19	—	0
Port 20	P20_3, P20_4	2
Port 21	P21_1 to P21_7	7
Port 22	P22_0 to P22_3	4
Port 23	P23_7	1
Port 24	P24_0 to P24_2	3
Total of pins		111

16.2 I/O Port Configuration

Figure 16.1 shows a connection diagram for the I/O port registers.

This figure shows the logic for reference, not the actual circuit.

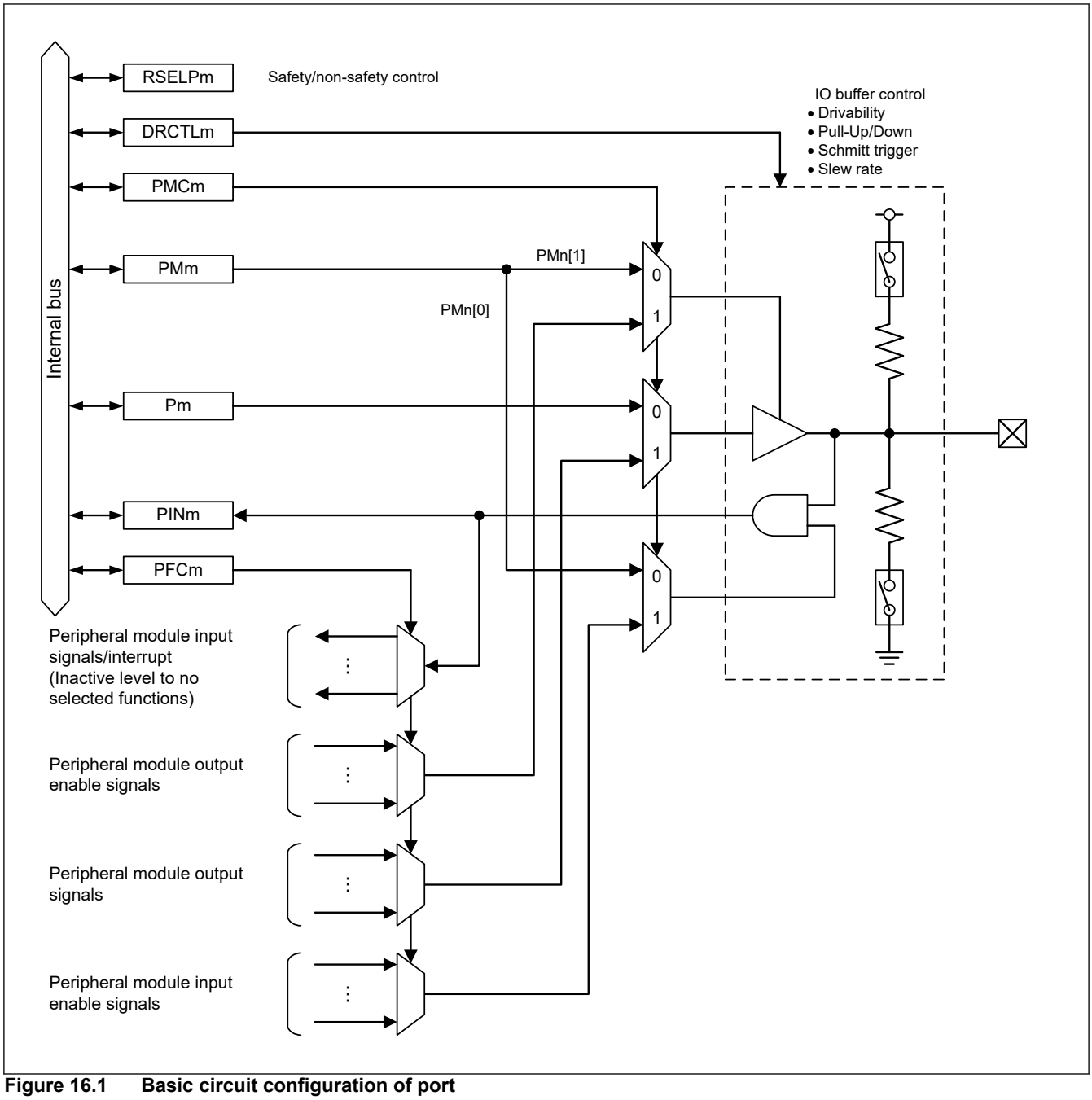


Figure 16.1 Basic circuit configuration of port

The pin assignments for each module/function are shown below. Please see section 16.4. Multiplexed Pin Configurations for detail of pin multiplex mapping.

Table 16.2 List of multiplexed pin configurations (1 of 22)

Module/Function	Channel	Pin function	Allocation port
Clock	—	CKIO	P04_1
		ETH0_REFCLK	P09_1
		ETH1_REFCLK	P06_1
		ETH2_REFCLK	P00_3
			P16_0
		RMII0_REFCLK	P09_1
		RMII1_REFCLK	P06_1
		RMII2_REFCLK	P00_3
Operating mode	—	MD0	P06_3
		MD1	P06_2
		MD2	P05_7
		MDD	P17_0
		MDV2	P20_3
		MDV3	P20_4
		MDW	P02_1
System control	—	RSTOUT#	P17_5
Debugging interface	—	TCK	P02_7
		TDI	P02_5
		TDO	P02_4
		TMS	P02_6
		TRACECLK	P17_4
			P18_6
			P22_2
		TRACECTL	P17_3
			P18_5
			P22_1
		TRACEDATA0	P01_6
			P21_1
		TRACEDATA1	P01_7
			P21_2
		TRACEDATA2	P02_0
			P21_3
		TRACEDATA3	P21_4
		TRACEDATA4	P21_5
		TRACEDATA5	P21_6
		TRACEDATA6	P13_2
			P21_7
		TRACEDATA7	P13_3
			P22_0

Table 16.2 List of multiplexed pin configurations (2 of 22)

Module/Function	Channel	Pin function	Allocation port
Bus State Controller	BSC	A0	P07_3
			P13_4
		A1	P07_2
		A2	P07_1
		A3	P07_0
		A4	P06_7
		A5	P06_6
		A6	P06_5
		A7	P06_4
		A8	P06_3
		A9	P06_2
		A10	P06_1
		A11	P05_7
		A12	P06_0
		A13	P00_1
			P05_6
			P13_2
		A14	P05_5
		A15	P02_3
			P18_6
		A16	P02_2
		A17	P02_1
		A18	P02_0
		A19	P01_7
		A20	P01_6
		A21	P14_6
		A22	P14_7
		A23	P15_0
		A24	P15_1
		A25	P15_2
		BS#	P14_4
			P14_7
			P18_2
			P23_7
		AH#	P01_3
			P02_3
		CAS#	P01_0
			P15_1
			P18_4
			P24_1

Table 16.2 List of multiplexed pin configurations (3 of 22)

Module/Function	Channel	Pin function	Allocation port
Bus State Controller	BSC	CKE	P01_1
			P15_0
			P18_3
			P24_0
		CS0#	P01_5
			P09_0
		CS2#	P01_2
			P16_1
		CS3#	P09_3
			P14_5
			P16_3
		CS5#	P00_6
			P08_4
		D0	P21_1
		D1	P21_2
		D2	P21_3
		D3	P21_4
		D4	P21_5
		D5	P21_6
		D6	P21_7
		D7	P22_0
		D8	P13_3
			P22_1
		D9	P13_2
			P22_2
		D10	P22_3
		D11	P15_3
			P23_7
		D12	P15_4
			P24_0
		D13	P15_5
			P24_1
		D14	P15_6
			P24_2
		D15	P00_0
			P10_4
		RAS#	P09_2
			P15_2
			P18_5
			P24_2

Table 16.2 List of multiplexed pin configurations (4 of 22)

Module/Function	Channel	Pin function	Allocation port
Bus State Controller	BSC	RD#	P00_2
			P13_3
			P17_7
		RD/WR#	P00_3
			P17_6
		WAIT#	P08_7
			P13_4
		WE0#/DQMLL	P01_5
			P02_4
			P18_0
		WE1#/DQMLU	P01_4
			P02_5
			P18_1
DMA Controller	DMAC	DACK	P09_2
			P09_7
			P17_4
			P17_7
		DREQ	P09_5
			P17_3
			P21_7
			P24_0
		TEND	P15_7
			P17_5
			P21_6
Interrupt	NMI	NMI	P15_3
			P16_2
			P21_3
	IRQ0	IRQ0	P00_1
			P09_2
			P18_3
	IRQ1	IRQ1	P00_3
			P07_4
			P18_4
	IRQ2	IRQ2	P01_2
			P01_3
	IRQ3	IRQ3	P01_4
			P15_4
	IRQ4	IRQ4	P02_0
			P22_2
	IRQ5	IRQ5	P08_5
			P13_2

Table 16.2 List of multiplexed pin configurations (5 of 22)

Module/Function	Channel	Pin function	Allocation port
Interrupt	IRQ6	IRQ6	P14_2
			P21_5
	IRQ7	IRQ7	P15_5
			P16_3
	IRQ8	IRQ8	P08_7
			P10_3
	IRQ9	IRQ9	P08_6
			P21_6
	IRQ10	IRQ10	P10_1
			P18_1
			P21_7
	IRQ11	IRQ11	P10_4
			P18_6
	IRQ12	IRQ12	P05_6
			P09_3
			P09_7
	IRQ13	IRQ13	P01_0
			P09_6
			P22_1
	IRQ14	IRQ14	P02_2
			P08_4
			P09_5
	IRQ15	IRQ15	P02_3
			P10_0
			P14_5
			P22_0
Multi-Function Timer Pulse Unit	MTU0	MTIOC0A	P13_2
			P14_3
			P23_7
		MTIOC0B	P13_3
			P14_4
			P24_0
		MTIOC0C	P13_3
			P15_1
			P24_1
		MTIOC0D	P13_4
			P15_2
			P24_2
	MTU1	MTIOC1A	P01_6
		MTIOC1B	P01_7

Table 16.2 List of multiplexed pin configurations (6 of 22)

Module/Function	Channel	Pin function	Allocation port
Multi-Function Timer Pulse Unit	MTU2	MTIOC2A	P02_2
		MTIOC2B	P02_3
	MTU3	MTIOC3A	P00_6
			P01_6
			P08_4
			P17_5
		MTIOC3B	P00_6
			P08_6
			P17_6
		MTIOC3C	P01_0
			P02_0
			P08_5
			P17_4
		MTIOC3D	P01_1
			P08_7
			P18_1
	MTU4	MTIOC4A	P01_5
			P09_0
			P17_7
			P18_0
		MTIOC4B	P01_2
			P09_2
			P18_2
			P18_3
		MTIOC4C	P01_0
			P09_1
			P17_7
			P18_0
		MTIOC4D	P01_3
			P09_3
			P18_2
			P18_3
	MTU5	MTIC5U	P00_1
			P10_2
			P18_4
		MTIC5V	P00_2
			P10_3
			P18_5
		MTIC5W	P00_3
			P10_4
			P18_6

Table 16.2 List of multiplexed pin configurations (7 of 22)

Module/Function	Channel	Pin function	Allocation port
Multi-Function Timer Pulse Unit	MTU6	MTIOC6A	P08_4
			P21_1
		MTIOC6B	P08_5
			P21_2
		MTIOC6C	P08_6
			P21_3
		MTIOC6D	P08_7
			P21_4
	MTU7	MTIOC7A	P09_0
			P21_5
		MTIOC7B	P09_1
			P21_6
		MTIOC7C	P09_2
			P21_7
		MTIOC7D	P09_3
			P22_0
	MTU8	MTIOC8A	P14_1
		MTIOC8B	P14_2
		MTIOC8C	P15_3
			P22_2
		MTIOC8D	P15_4
			P22_3
	MTU	MTCLKA	P13_5
		MTCLKB	P13_6
		MTCLKC	P13_7
		MTCLKD	P14_0
Port Output Enable 3	POE3	POE0#	P01_4
			P17_3
		POE4#	P22_1
		POE8#	P13_2
			P14_5
			P24_1
		POE10#	P02_2
			P10_1
		POE11#	P02_3
			P10_0
General PWM Timer	GPT	GTADSML0	P02_0
		GTADSML1	P09_5
		GTADSMP0	P09_4
		GTADSMP1	P16_3

Table 16.2 List of multiplexed pin configurations (8 of 22)

Module/Function	Channel	Pin function	Allocation port
General PWM Timer	GPT0	GTIOC0A	P00_6
			P01_6
			P17_4
		GTIOC0B	P01_0
			P02_0
			P17_5
	GPT1	GTIOC1A	P00_6
			P17_6
		GTIOC1B	P01_1
			P18_1
	GPT2	GTIOC2A	P01_5
			P17_7
			P18_0
		GTIOC2B	P01_2
			P18_2
			P18_3
	GPT3	GTIOC3A	P01_0
			P17_7
			P18_0
		GTIOC3B	P01_3
			P18_2
			P18_3
	GPT4	GTIOC4A	P08_4
		GTIOC4B	P08_6
	GPT5	GTIOC5A	P08_5
		GTIOC5B	P08_7
	GPT6	GTIOC6A	P09_0
		GTIOC6B	P09_2
	GPT7	GTIOC7A	P09_1
		GTIOC7B	P09_3
	GPT8	GTIOC8A	P14_1
		GTIOC8B	P13_4
			P14_2
	GPT9	GTIOC9A	P01_6
		GTIOC9B	P01_7
	GPT10	GTIOC10A	P02_2
			P13_2
		GTIOC10B	P02_3
			P13_3
	GPT11	GTIOC11A	P06_4
		GTIOC11B	P06_5

Table 16.2 List of multiplexed pin configurations (9 of 22)

Module/Function	Channel	Pin function	Allocation port
General PWM Timer	GPT12	GTIOC12A	P06_6
		GTIOC12B	P06_7
	GPT13	GTIOC13A	P07_0
		GTIOC13B	P07_1
	GPT14	GTIOC14A	P05_6
			P21_1
		GTIOC14B	P05_5
			P21_2
			P21_3
	GPT15	GTIOC15A	P09_6
			P21_2
			P21_3
		GTIOC15B	P09_7
			P21_4
	GPT16	GTIOC16A	P06_0
			P21_5
		GTIOC16B	P06_1
			P21_6
	GPT17	GTIOC17A	P07_2
			P21_7
		GTIOC17B	P07_3
			P22_0
Port Output Enable for GPT	POEG	GTETRGA	P17_3
			P23_7
		GTETRGB	P17_4
			P22_1
			P24_0
		GTETRGC	P01_1
			P17_5
			P24_1
		GTETRGD	P24_2
		GTETRGSA	P09_1
			P22_2
		GTETRGSB	P09_3
			P22_3

Table 16.2 List of multiplexed pin configurations (10 of 22)

Module/Function	Channel	Pin function	Allocation port
Compare Match Timer W	CMTW0	CMTW0_TIC0	P10_0
			P21_1
		CMTW0_TIC1	P09_6
			P21_2
		CMTW0_TOC0	P09_3
			P09_5
		CMTW0_TOC1	P05_5
			P16_1
	CMTW1	CMTW1_TIC0	P05_6
		CMTW1_TIC1	P08_6
			P18_3
		CMTW1_TOC0	P06_0
			P13_3
		CMTW1_TOC1	P05_7
			P21_5
Real Time Clock	RTC	RTCAT1HZ	P02_2
			P09_2
			P10_3
Serial Communications Interface	SCI0	CTS0#	P16_2
			P21_6
		DE0	P21_7
		RXD0/SCL0/MISO0	P10_3
			P16_1
		SCK0	P10_4
			P16_3
			P18_2
		SS0#/CTS0#/RTS0#	P17_0
		TXD0/SDA0/MOSI0	P10_2
			P16_0
	SCI1	CTS1#	P01_6
			P13_3
		DE1	P02_1
			P02_4
		RXD1/SCL1/MISO1	P02_0
			P24_0
		SCK1	P01_7
			P23_7
		SS1#/CTS1#/RTS1#	P02_3
			P13_2
		TXD1/SDA1/MOSI1	P02_2
			P24_2

Table 16.2 List of multiplexed pin configurations (11 of 22)

Module/Function	Channel	Pin function	Allocation port
Serial Communications Interface	SCI2	CTS2#	P01_0
		DE2	P00_0
			P01_1
		RXD2/SCL2/MISO2	P00_1
			P08_5
		SCK2	P00_0
			P08_4
		SS2#/CTS2#/RTS2#	P00_3
		TXD2/SDA2/MOSI2	P00_2
			P08_7
	SCI3	CTS3#	P10_1
			P17_4
		DE3	P09_1
			P18_3
		RXD3/SCL3/MISO3	P09_0
			P17_7
		SCK3	P08_6
			P17_6
		SS3#/CTS3#/RTS3#	P18_1
		TXD3/SDA3/MOSI3	P04_1
			P18_0
	SCI4	CTS4#	P06_1
		DE4	P06_3
			P18_6
		RXD4/SCL4/MISO4	P09_7
			P18_5
		SCK4	P10_0
			P18_6
		SS4#/CTS4#/RTS4#	P06_0
			P22_1
		TXD4/SDA4/MOSI4	P09_4
			P18_4

Table 16.2 List of multiplexed pin configurations (12 of 22)

Module/Function	Channel	Pin function	Allocation port
Serial Communications Interface	SCI5	CTS5#	P09_6
			P15_7
			P21_5
		DE5	P09_5
			P22_0
		RXD5/SCL5/MISO5	P02_6
			P15_0
			P21_2
			P22_3
		SCK5	P02_5
			P14_7
			P21_1
		SS5#/CTS5#/RTS5#	P09_3
			P15_2
			P21_4
		TXD5/SDA5/MOSI5	P02_7
			P15_1
			P21_3
I2C Bus Interface	IIC0	IIC_SCL0	P13_2
		IIC_SDA0	P13_3
	IIC1	IIC_SCL1	P09_5
			P21_1
		IIC_SDA1	P10_0
			P21_2
	IIC2	IIC_SCL2	P13_5
			P18_6
		IIC_SDA2	P04_1
			P18_2
Ethernet Subsystem	ETH0	ETH0_RXCLK	P08_6
		ETH0_RXD0	P10_1
		ETH0_RXD1	P10_2
		ETH0_RXD2	P10_3
		ETH0_RXD3	P08_4
		ETH0_RXDV	P08_5
		ETH0_RXER	P09_2
		ETH0_TXCLK	P09_7
		ETH0_TXD0	P09_6
		ETH0_TXD1	P09_5
		ETH0_TXD2	P09_4
		ETH0_TXD3	P09_3
		ETH0_TXEN	P10_0

Table 16.2 List of multiplexed pin configurations (13 of 22)

Module/Function	Channel	Pin function	Allocation port
Ethernet Subsystem	ETH1	ETH1_RXCLK	P07_3
		ETH1_RXD0	P06_6
		ETH1_RXD1	P06_7
		ETH1_RXD2	P07_0
		ETH1_RXD3	P07_1
		ETH1_RXDV	P07_2
		ETH1_RXER	P05_6
		ETH1_TXCLK	P06_4
		ETH1_TXD0	P06_3
		ETH1_TXD1	P06_2
		ETH1_TXD2	P05_7
		ETH1_TXD3	P06_0
		ETH1_TXEN	P06_5
	ETH2	ETH2_COL	P02_3
			P14_3
			P18_5
		ETH2_CRS	P02_0
			P14_2
			P18_3
		ETH2_RXCLK	P24_1
		ETH2_RXD0	P23_7
		ETH2_RXD1	P24_0
		ETH2_RXD2	P24_2
		ETH2_RXD3	P00_0
		ETH2_RXDV	P00_1
		ETH2_RXER	P16_3
			P18_4
		ETH2_TXCLK	P00_6
		ETH2_TXD0	P01_5
		ETH2_TXD1	P01_4
		ETH2_TXD2	P01_3
		ETH2_TXD3	P01_2
		ETH2_TXEN	P00_2
		ETH2_TXER	P01_7
			P17_4

Table 16.2 List of multiplexed pin configurations (14 of 22)

Module/Function	Channel	Pin function	Allocation port
Ethernet MAC	GMAC	GMAC_MDC	P01_1
			P08_7
		GMAC_MDIO	P01_0
			P09_0
		GMAC_PTPTRG0	P13_5
		GMAC_PTPTRG1	P01_6
			P13_7
			P14_1
		GMAC_PTPOUT0	P09_1
			P13_6
			P20_4
		GMAC_PTPOUT1	P02_1
			P14_0
			P20_3
			P21_4
		GMAC_PTPOUT2	P13_2
			P18_6
		GMAC_PTPOUT3	P13_3
			P18_2
EtherCAT slave controller	ESC	ESC_I2CCLK	P13_2
		ESC_I2CDATA	P13_3
		ESC_IRQ	P14_4
			P17_0
		ESC_LATCH0	P01_6
			P13_5
			P13_7
			P14_1
		ESC_LATCH1	P01_6
			P13_7
			P13_5
			P14_1
		ESC_LEDERR	P20_3
		ESC_LEDRUN	P21_3
		ESC_LEDSTER	P21_3
		ESC_LINKACT0	P21_6
		ESC_LINKACT1	P20_4
		ESC_LINKACT2	P22_1
		ESC_MDC	P01_1
			P08_7
		ESC_MDIO	P01_0
			P09_0

Table 16.2 List of multiplexed pin configurations (15 of 22)

Module/Function	Channel	Pin function	Allocation port
EtherCAT slave controller	ESC	ESC_PHYLINK0	P01_6
			P10_4
		ESC_PHYLINK1	P05_5
		ESC_PHYLINK2	P01_1
			P17_4
		ESC_RESETOUT#	P13_4
		ESC_SYNC0	P09_1
			P02_1
			P13_6
			P14_0
			P21_1
			P21_2
			P21_4
		ESC_SYNC1	P09_1
			P02_1
			P13_6
			P14_0
			P21_1
			P21_2
			P21_4
USB 2.0 HS host/function module	USB	USB_EXICEN	P09_6
			P10_0
			P16_2
		USB_OTGID	P02_0
		USB_OVRCUR	P00_2
			P09_7
			P17_5
		USB_VBUSEN	P09_3
			P09_5
			P22_3
		USB_VBUSIN	P07_4
CAN-FD interface	CAN0	CANRX0	P01_7
			P09_5
			P18_5
		CANRXDP0	P09_6
		CANTX0	P02_2
			P10_0
			P18_4
		CANTXDP0	P09_3
			P09_7

Table 16.2 List of multiplexed pin configurations (16 of 22)

Module/Function	Channel	Pin function	Allocation port
CAN-FD interface	CAN1	CANRX1	P02_3
			P06_0
			P17_3
		CANRXDP1	P06_2
			P18_3
		CANTX1	P02_0
			P06_1
			P20_3
		CANTXDP1	P01_6
			P06_3
			P08_4
Serial Peripheral Interface	SPI0	SPI_MISO0	P13_2
			P21_5
		SPI_MOSI0	P04_1
			P22_1
		SPI_RSPCK0	P13_3
			P17_4
		SPI_SSL00	P07_3
			P09_7
			P22_3
		SPI_SSL01	P10_4
			P17_0
		SPI_SSL02	P21_4
		SPI_SSL03	P16_2
	SPI1	SPI_MISO1	P06_5
			P14_1
			P14_7
		SPI_MOSI1	P06_4
			P13_7
			P15_0
		SPI_RSPCK1	P01_5
			P10_1
			P13_5
		SPI_SSL10	P06_6
			P09_1
			P14_2
			P15_1
		SPI_SSL11	P06_7
			P14_3
			P15_2

Table 16.2 List of multiplexed pin configurations (17 of 22)

Module/Function	Channel	Pin function	Allocation port
Serial Peripheral Interface	SPI1	SPI_SSL12	P13_4
			P15_6
			P22_2
		SPI_SSL13	P08_7
			P14_4
			P15_7
	SPI2	SPI_MISO2	P18_6
			P21_2
		SPI_MOSI2	P18_5
		SPI_RSPCK2	P05_5
			P18_4
		SPI_SSL20	P21_1
		SPI_SSL21	P09_4
		SPI_SSL22	P05_6
			P06_1
		SPI_SSL23	P05_7
			P06_0
	SPI3	SPI_MISO3	P02_0
			P16_1
		SPI_MOSI3	P02_2
			P16_0
		SPI_RSPCK3	P01_7
			P16_2
		SPI_SSL30	P02_3
			P16_3
		SPI_SSL31	P02_5
			P17_3
		SPI_SSL32	P08_4
			P17_4
		SPI_SSL33	P02_4
			P21_3

Table 16.2 List of multiplexed pin configurations (18 of 22)

Module/Function	Channel	Pin function	Allocation port
expanded Serial Peripheral Interface	XSPI0	XSPI0_CKN	P14_5
		XSPI0_CKP	P14_6
		XSPI0_CS0#	P15_7
		XSPI0_CS1#	P16_0
		XSPI0_DS	P14_4
		XSPI0_ECS0#	P14_2
		XSPI0_ECS1#	P13_7
		XSPI0_INT0#	P14_0
		XSPI0_INT1#	P14_1
		XSPI0_IO0	P14_7
		XSPI0_IO1	P15_0
		XSPI0_IO2	P15_1
		XSPI0_IO3	P15_2
		XSPI0_IO4	P15_3
		XSPI0_IO5	P15_4
		XSPI0_IO6	P15_5
		XSPI0_IO7	P15_6
		XSPI0_RESET0#	P16_1
		XSPI0_RESET1#	P16_2
		XSPI0_RSTO0#	P16_3
		XSPI0_RSTO1#	P14_3
		XSPI0_WP0#	P13_6
		XSPI0_WP1#	P13_5
	XSPI1	XSPI1_CKP	P17_7
		XSPI1_CS0#	P18_2
		XSPI1_DS	P17_6
		XSPI1_IO0	P18_0
		XSPI1_IO1	P17_0
		XSPI1_IO2	P17_3
		XSPI1_IO3	P17_4
		XSPI1_IO4	P18_3
		XSPI1_IO5	P18_4
		XSPI1_IO6	P18_5
		XSPI1_IO7	P18_6

Table 16.2 List of multiplexed pin configurations (19 of 22)

Module/Function	Channel	Pin function	Allocation port
Delta-sigma Interface	DSMIF0	MCLK0	P00_1
			P21_1
		MDAT0	P00_6
			P21_2
		MCLK1	P01_0
			P15_3
			P21_3
			P22_2
		MDAT1	P01_1
			P15_4
			P21_4
		MCLK2	P08_5
			P15_5
			P21_5
		MDAT2	P08_7
			P15_6
			P21_6
	DSMIF1	MCLK3	P09_3
			P16_0
			P21_7
		MDAT3	P09_6
			P16_1
			P22_0
		MCLK4	P13_2
			P23_7
		MDAT4	P13_3
			P24_0
		MCLK5	P24_1
		MDAT5	P24_2
12-Bit A/D Converter	ADC120	ADTRG0#	P01_7
			P07_4
			P16_1
			P18_6
	ADC121	ADTRG1#	P17_3
			P18_1
			P21_5

Table 16.2 List of multiplexed pin configurations (20 of 22)

Module/Function	Channel	Pin function	Allocation port
Serial host interface	SHOSTIF	HSPI_CK	P18_6
		HSPI_CS#	P21_2
		HSPI_IO0	P18_5
		HSPI_IO1	P18_4
		HSPI_IO2	P21_3
		HSPI_IO3	P21_5
		HSPI_IO4	P21_6
		HSPI_IO5	P21_7
		HSPI_IO6	P22_0
		HSPI_IO7	P22_1
		HSPI_INT#	P14_5
			P21_1
Mailbox	—	MBX_HINT#	P10_4
			P13_7
			P21_4
Encoder interface subsystem	ENCIF0	ENCIFCK0	P02_0
			P16_0
		ENCIFOE0	P01_7
			P16_1
		ENCIFDO0	P02_3
			P21_4
		ENCIFDI0	P02_2
			P21_5
	ENCIF1	ENCIFCK1	P09_6
			P16_2
		ENCIFOE1	P09_7
			P16_3
		ENCIFDO1	P10_0
			P22_2
		ENCIFDI1	P10_1
			P22_3

Table 16.2 List of multiplexed pin configurations (21 of 22)

Module/Function	Channel	Pin function	Allocation port
EnDat 2.2	ENDAT0	DUEI0	P01_4
			P18_4
		TST_OUT0	P01_5
			P18_5
		SI0#	P01_6
			P18_6
	ENDAT1	DUEI1	P00_0
			P06_7
		TST_OUT1	P00_2
			P07_0
		SI1#	P00_3
			P07_1
HIPERFACE DSL	HDSL0	HDSL0_LINK	P01_4
			P18_4
		HDSL0_SMPL	P01_5
			P18_5
		HDSL0_CLK1	P01_6
			P18_6
		HDSL0_SEL1	P00_0
			P13_7
		HDSL0_MISO1	P02_1
			P14_0
		HDSL0_MOSI1	P06_0
			P14_1
		HDSL0_CLK2	P06_1
			P14_2
		HDSL0_SEL2	P06_4
			P14_3
		HDSL0_MISO2	P06_5
			P14_4
		HDSL0_MOSI2	P00_2
			P16_0

Table 16.2 List of multiplexed pin configurations (22 of 22)

Module/Function	Channel	Pin function	Allocation port
HIPERFACE DSL	HDSL1	HDSL1_LINK	P05_5
			P06_7
		HDSL1_SMPL	P06_6
			P07_0
		HDSL1_CLK1	P00_3
			P07_1
		HDSL1_SEL1	P07_2
			P16_1
		HDSL1_MISO1	P07_3
			P16_2
		HDSL1_MOSI1	P08_4
			P16_3
		HDSL1_CLK2	P09_2
			P17_4
		HDSL1_SEL2	P07_4
			P09_3
		HDSL1_MISO2	P09_4
			P17_0
		HDSL1_MOSI2	P05_6
			P09_5
ENCOUT	—	POUTA	P01_1
			P14_2
		POUTB	P01_2
			P14_3
		POUTZ	P01_3
			P14_4

16.3 Register Descriptions

16.3.1 Pm : Port m Register (m = 00 to 24)

Base address: PORT_NSR = 0x800A_0000
PORT_SR = 0x8103_0000

Offset address: 0x000 + 0x001 × m

Bit position:	7	6	5	4	3	2	1	0
Bit field:	POUT_7	POUT_6	POUT_5	POUT_4	POUT_3	POUT_2	POUT_1	POUT_0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W[*1]
7:0	POUT_0 to POUT_7	Pm_n Output Data Store (n: bit position) 0: Low output 1: High output	R/W

Note: Available pins depend on port group. For details, see Table 16.1. P07_4 is not available because of input only.
Note 1. If access address is not region selected by RSELPm register, the bit is read only.

The Pm register holds the data to be output from the pins used for general I/O.

16.3.2 PMm : Port m Mode Register (m = 00 to 24)

Base address: PORT_NSR = 0x800A_0000
PORT_SR = 0x8103_0000

Offset address: 0x200 + 0x002 × m

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PM7[1:0]		PM6[1:0]		PM5[1:0]		PM4[1:0]		PM3[1:0]		PM2[1:0]		PM1[1:0]		PM0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W*1
1:0	PM0[1:0]	Pm_0 I/O Select 0 0: Hi-Z 0 1: Input 1 0: Output (Output data is not input to input buffer) 1 1: Output (Output data is input to input buffer)	R/W
3:2	PM1[1:0]	Pm_1 I/O Select This function is the same as PM0[1:0].	R/W
5:4	PM2[1:0]	Pm_2 I/O Select This function is the same as PM0[1:0].	R/W
7:6	PM3[1:0]	Pm_3 I/O Select This function is the same as PM0[1:0].	R/W
9:8	PM4[1:0]	Pm_4 I/O Select This function is the same as PM0[1:0].	R/W
11:10	PM5[1:0]	Pm_5 I/O Select This function is the same as PM0[1:0].	R/W
13:12	PM6[1:0]	Pm_6 I/O Select This function is the same as PM0[1:0].	R/W
15:14	PM7[1:0]	Pm_7 I/O Select This function is the same as PM0[1:0].	R/W

Note: Available pins depend on port group. For details, see Table 16.1. P07_4 supports only Input (01b).
Note: For the port shared with the following signals, initial value is 01b; input.
MDD (P17_0), MDV3 (P20_4), MDV2 (P20_3), MDW (P02_1), MD2 (P05_7), MD1 (P06_2), MD0 (P06_3)
Note 1. If access address is not region selected by RSELPm register, the bit is read only.

The PMm register is used to select I/O function for the pins used for general I/O.

16.3.3 PMCm : Port m Mode Control Register (m = 00 to 24)

Base address: PORT_NSR = 0x800A_0000
PORT_SR = 0x8103_0000

Offset address: 0x400 + 0x001 × m

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PMC7	PMC6	PMC5	PMC4	PMC3	PMC2	PMC1	PMC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W*1
7:0	PMC0 to PMC7	Pm_n Pin Mode Control (n: bit position) 0: Uses the pin as a general I/O pin. 1: Uses the pin as an I/O port for peripheral functions.	R/W

Note: Available pins depend on port group. For details, see Table 16.1.
Note: For the port shared with the following signals, initial value is 1: peripheral function.
RSTOUT# (P17_5), TDO (P02_4), TDI (P02_5), TMS (P02_6), TCK (P02_7)

Note 1. If access address is not region selected by RSELPm register, the bit is read only.

The PMCm register is used to select the function of the pins of the port.

16.3.4 PFCm : Port m Function Control Register (m = 00 to 24)

Base address: PORT_NSR = 0x800A_0000
PORT_SR = 0x8103_0000

Offset address: 0x600 + 0x004 × m

Bit position:	31 28	27 24	23 20	19 16	15 12	11 8	7 4	3 0
Bit field:	PFC7[3:0]	PFC6[3:0]	PFC5[3:0]	PFC4[3:0]	PFC3[3:0]	PFC2[3:0]	PFC1[3:0]	PFC0[3:0]
Value after reset:	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 0 0

Bit	Symbol	Function	R/W*1
3:0	PFC0[3:0]	Pm_0 Pin function Select Select one function from maximum 16 shared functions. For details of pin multiplex mapping, see section 16.4. Multiplexed Pin Configurations.	R/W
7:4	PFC1[3:0]	Pm_1 Pin function Select This function is the same as PFC0[3:0].	R/W
11:8	PFC2[3:0]	Pm_2 Pin function Select This function is the same as PFC0[3:0].	R/W
15:12	PFC3[3:0]	Pm_3 Pin function Select This function is the same as PFC0[3:0].	R/W
19:16	PFC4[3:0]	Pm_4 Pin function Select This function is the same as PFC0[3:0].	R/W
23:20	PFC5[3:0]	Pm_5 Pin function Select This function is the same as PFC0[3:0].	R/W
27:24	PFC6[3:0]	Pm_6 Pin function Select This function is the same as PFC0[3:0].	R/W
31:28	PFC7[3:0]	Pm_7 Pin function Select This function is the same as PFC0[3:0].	R/W

Note: Available pins depend on port group. For details, see Table 16.1.
Note: For the port shared with the following signal, initial value is 1.
TDI (P02_5)
Note 1. If access address is not region selected by RSELPm register, the bit is read only.

The PFCm register is used to select peripheral function for the pins.

16.3.5 PINm : Port m Input Register (m = 00 to 24)

Base address: PORT_NSR = 0x800A_0000
PORT_SR = 0x8103_0000

Offset address: 0x800 + 0x001 × m

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PIN7	PIN6	PIN5	PIN4	PIN3	PIN2	PIN1	PIN0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PIN0 to PIN7	Pm_n Pin Input (n: bit position) 0: Low input 1: High input	R

Note: Available pins depend on port group. For details, see Table 16.1.

The PINm register reflects the states of the individual input port pins.

16.3.6 DRCTLm : I/O Buffer m Function Switching Register (m = 00 to 24)

Base address: PORT_NSR = 0x800A_0000
PORT_SR = 0x8103_0000

Offset address: 0xA00 + 0x008 × m

Bit position:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48
Bit field:	—	—	SR7	SMT7	PUD7[1:0]		DRV7[1:0]		—	—	SR6	SMT6	PUD6[1:0]		DRV6[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bit field:	—	—	SR5	SMT5	PUD5[1:0]		DRV5[1:0]		—	—	SR4	SMT4	PUD4[1:0]		DRV4[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	SR3	SMT3	PUD3[1:0]		DRV3[1:0]		—	—	SR2	SMT2	PUD2[1:0]		DRV2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	SR1	SMT1	PUD1[1:0]		DRV1[1:0]		—	—	SR0	SMT0	PUD0[1:0]		DRV0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W*1
1:0	DRV0[1:0]	Pm_0 Driving Ability Control 0 0: Low 0 1: Middle 1 0: High 1 1: Ultra High	R/W
3:2	PUD0[1:0]	Pm_0 Pull-Up/Down Control 0 0: No pull-up/down 0 1: Pull-up 1 0: Pull-down 1 1: Setting prohibited	R/W
4	SMT0	Pm_0 Schmitt Trigger Control 0: Disable Schmitt Trigger input 1: Enable Schmitt Trigger input	R/W
5	SR0	Pm_0 Slew Rate Control 0: Slow 1: Fast	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
9:8	DRV1[1:0]	Pm_1 Driving Ability Control This function is the same as DRV0[1:0].	R/W
11:10	PUD1[1:0]	Pm_1 Pull-Up/Down Control This function is the same as PUD0[1:0].	R/W
12	SMT1	Pm_1 Schmitt Trigger Control This function is the same as SMT0.	R/W
13	SR1	Pm_1 Slew Rate Control This function is the same as SR0.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
17:16	DRV2[1:0]	Pm_2 Driving Ability Control This function is same as DRV0[1:0].	R/W
19:18	PUD2[1:0]	Pm_2 Pull-Up/Down Control This function is same as PUD0[1:0].	R/W

Bit	Symbol	Function	R/W*1
20	SMT2	Pm_2 Schmitt Trigger Control This function is the same as SMT0.	R/W
21	SR2	Pm_2 Slew Rate Control This function is the same as SR0.	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
25:24	DRV3[1:0]	Pm_3 Driving Ability Control This function is same as DRV0[1:0].	R/W
27:26	PUD3[1:0]	Pm_3 Pull-Up/Down Control This function is same as PUD0[1:0].	R/W
28	SMT3	Pm_3 Schmitt Trigger Control This function is the same as SMT0.	R/W
29	SR3	Pm_3 Slew Rate Control This function is the same as SR0.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W
33:32	DRV4[1:0]	Pm_4 Driving Ability Control This function is same as DRV0[1:0].	R/W
35:34	PUD4[1:0]	Pm_4 Pull-Up/Down Control This function is same as PUD0[1:0].	R/W
36	SMT4	Pm_4 Schmitt Trigger Control This function is the same as SMT0.	R/W
37	SR4	Pm_4 Slew Rate Control This function is the same as SR0.	R/W
39:38	—	These bits are read as 0. The write value should be 0.	R/W
41:40	DRV5[1:0]	Pm_5 Driving Ability Control This function is same as DRV0[1:0].	R/W
43:42	PUD5[1:0]	Pm_5 Pull-Up/Down Control This function is same as PUD0[1:0].	R/W
44	SMT5	Pm_5 Schmitt Trigger Control This function is the same as SMT0.	R/W
45	SR5	Pm_5 Slew Rate Control This function is the same as SR0.	R/W
47:46	—	These bits are read as 0. The write value should be 0.	R/W
49:48	DRV6[1:0]	Pm_6 Driving Ability Control This function is same as DRV0[1:0].	R/W
51:50	PUD6[1:0]	Pm_6 Pull-Up/Down Control This function is same as PUD0[1:0].	R/W
52	SMT6	Pm_6 Schmitt Trigger Control This function is the same as SMT0.	R/W
53	SR6	Pm_6 Slew Rate Control This function is the same as SR0.	R/W
55:54	—	These bits are read as 0. The write value should be 0.	R/W
57:56	DRV7[1:0]	Pm_7 Driving Ability Control This function is same as DRV0[1:0].	R/W
59:58	PUD7[1:0]	Pm_7 Pull-Up/Down Control This function is same as PUD0[1:0].	R/W
60	SMT7	Pm_7 Schmitt Trigger Control This function is the same as SMT0.	R/W
61	SR7	Pm_7 Slew Rate Control This function is the same as SR0.	R/W
63:62	—	These bits are read as 0. The write value should be 0.	R/W

Note: Available pins depend on port group. For details, see Table 16.1. P07_4 supports only Schmitt Trigger Control. Other ports have all features.

Note: For the port shared with the following signals, initial value of SRx and DRVx is different.
TDO (P02_4) : SR4 = 1, DRV4[1:0] = 10b
TMS (P02_6) : SR6 = 1, DRV6[1:0] = 10b
Note 1. If access address is not region selected by RSELPm register, the bit is read only.

The DRCTLm register is used to control function of I/O buffers. This register is valid even if the pin is used as either general I/O pin or an I/O port for peripheral functions in PMCm register.

16.3.7 RSELPm : Port m Region Select Register (m = 00 to 24)

Base address: PTADR = 0x8103_0C00

Offset address: 0x000 + 0x001 × m

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RS7	RS6	RS5	RS4	RS3	RS2	RS1	RS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RS0 to RS7	Pm_n pin I/O port registers Region Select (n = bit position) 0: Safety region (Address is 0x8103_0xxx) 1: Non safety region (Address is 0x800A_0xxx)	R/W

Note: Available pins depend on port group. For details, see Table 16.1.

The RSELPm register is used to specify which is the attribution of the I/O port, Safety or Non safety. I/O ports registers except this register are defined two address, Safety region and Non safety region. When Safety region is selected in this register, access to Non safety region in I/O ports registers is limited to read-only though access to Safety region is allowed R/W. On the other hand, when Non safety region is selected in this register, access to Safety region in I/O ports registers is limited to read-only though access to Non safety region is allowed R/W. (Except PINm register due to read-only originally.)

16.4 Multiplexed Pin Configurations

Select the function number with PFCn bit of Port m Function Control Register (PFCm: m = 00 to 24).

One of three types of IO buffers, Type A, B, or C is used each port. Depending on the IO type, recommended DRCTLm register setting may be different each peripheral function. For details, see Table 52.14.

Each port belongs to one voltage domain. Voltage domain VCC1833_n (n = 2, 3) are selectable from 1.8 V or 3.3 V according to the selected peripheral function. When using Ethernet with RGMII or 1.8 V xSPI, supply 1.8 V to the corresponding voltage domain (VCC1833_n). In other cases, supply 3.3 V to the corresponding voltage domain (VCC1833_n). Note that 1.8 V operation is not supported for other peripheral function than Ethernet with RGMII and xSPI.

16.4.1 Port 00 Function Assignment

Table 16.3 Port 00 function assignment (1 of 2)

Function number (PFCn)	Port	P00_0	P00_1	P00_2	P00_3	P00_6
	IO Type	Type B	Type B	Type B	Type A	Type B
	Voltage domain	VCC1833_2	VCC1833_2	VCC1833_2	VCC1833_2	VCC1833_2
0		ETH2_RXD3	IRQ0	ETH2_TXEN	IRQ1	ETH2_TXCLK
1		—	ETH2_RXDV	—	ETH2_REFCLK	CS5#
2		D15	—	RD#	RMII2_REFCLK	MTIOC3B
3		SCK2	A13	MTIC5V	—	GTIOC1A
4		DE2	MTIC5U	TXD2/SDA2/MOSI2	RD/WR#	MDAT0
5		—	RXD2/SCL2/MISO2	USB_OVRCUR	MTIC5W	MTIOC3A
6		DUEI1	MCLK0	TST_OUT1	SS2#/CTS2#/RTS2#	GTIOC0A
7		HDSL0_SEL1	—	HDSL0_MOSI2	SI1#	—
8		—	—	—	HDSL1_CLK1	—

Table 16.3 Port 00 function assignment (2 of 2)

Function number (PFCn)	Port	P00_0	P00_1	P00_2	P00_3	P00_6
	IO Type	Type B	Type B	Type B	Type A	Type B
	Voltage domain	VCC1833_2	VCC1833_2	VCC1833_2	VCC1833_2	VCC1833_2
9 to 15		—	—	—	—	—

16.4.2 Port 01 Function Assignment

Table 16.4 Port 01 function assignment

Function number (PFCn)	Port	P01_0	P01_1	P01_2	P01_3	P01_4	P01_5	P01_6	P01_7
	IO Type	Type A	Type A	Type B	Type B	Type B	Type B	Type A	Type A
	Voltage domain	VCC1833_2	VCC1833_2	VCC1833_2	VCC1833_2	VCC1833_2	VCC1833_2	VCC33	VCC33
0		GMAC_MDIO	GMAC_MDC	IRQ2	ETH2_TXD2	IRQ3	ETH2_TXD0	GMAC_PTPTRG1	ETH2_TXER
1		—	—	ETH2_TXD3	AH#	ETH2_TXD1	WE0#/DQMLL	TRACEDATA0	TRACEDATA1
2		CAS#	CKE	CS2#	MTIOC4D	WE1#/DQMLU	CS0#	A20	A19
3		ESC_MDIO	ESC_MDC	MTIOC4B	GTIOC3B	POE0#	MTIOC4A	ESC_LATCH1	MTIOC1B
4		MTIOC4C	MTIOC3D	GTIOC2B	IRQ2	DUEI0	GTIOC2A	ESC_LATCH0	GTIOC9B
5		GTIOC3A	GTIOC1B	POUTB	POUTZ	HDSL0_LINK	SPI_RSPCK1	MTIOC1A	ADTRG0#
6		CTS2#	DE2	—	—	—	TST_OUT0	GTIOC9A	SCK1
7		MCLK1	MDAT1	—	—	—	HDSL0_SMPL	CTS1#	SPI_RSPCK3
8		MTIOC3C	ESC_PHYLINK2	—	—	—	—	CANTXDP1	CANRX0
9		GTIOC0B	GTETRGC	—	—	—	—	ESC_PHYLINK0	ENCIFOE0
10		IRQ13	POUTA	—	—	—	—	—	—
11		—	—	—	—	—	—	MTIOC3A	—
12		—	—	—	—	—	—	GTIOC0A	—
13		—	—	—	—	—	—	SI0#	—
14		—	—	—	—	—	—	HDSL0_CLK1	—
15		—	—	—	—	—	—	—	—

16.4.3 Port 02 Function Assignment

Table 16.5 Port 02 function assignment (1 of 2)

Function number (PFCn)	Port	P02_0	P02_1	P02_2	P02_3	P02_4	P02_5	P02_6	P02_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33
0		IRQ4	GMAC_PTPOUT1	IRQ14	IRQ15	TDO	—	TMS	TCK
1		ETH2_CRS	A17	—	ETH2_COL	WE0#/DQMLL	TDI	RXD5/SCL5/MISO5	TXD5/SDA5/MOSI5
2		TRACEDATA2	ESC_SYNC1	A16	A15	DE1	WE1#/DQMLU	—	—
3		A18	ESC_SYNC0	MTIOC2A	AH#	SPI_SSL33	SCK5	—	—
4		GTADSML0	DE1	GTIOC10A	MTIOC2B	—	SPI_SSL31	—	—
5		RXD1/SCL1/MISO1	—	POE10#	GTIOC10B	—	—	—	—
6		SPI_MISO3	HDSL0_MISO1	TXD1/SDA1/MOSI1	POE11#	—	—	—	—
7		CANTX1	—	SPI_MOSI3	SS1#/CTS1#/RTS1#	—	—	—	—

Table 16.5 Port 02 function assignment (2 of 2)

Function number (PFCn)	Port	P02_0	P02_1	P02_2	P02_3	P02_4	P02_5	P02_6	P02_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33
8		USB_OTGID	—	CANTX0	SPI_SSL30	—	—	—	—
9		ENCIFCK0	—	ENCIFDI0	CANRX1	—	—	—	—
10		—	—	RTCAT1HZ	ENCIFDO0	—	—	—	—
11		MTIOC3C	—	—	—	—	—	—	—
12		GTIOC0B	—	—	—	—	—	—	—
13 to 15		—	—	—	—	—	—	—	—

16.4.4 Port 04 Function Assignment

Table 16.6 Port 04 function assignment

Function number (PFCn)	Port	P04_1
	IO Type	Type A
	Voltage domain	VCC33
0		CKIO
1		TXD3/SDA3/MOSI3
2		SPI_MOSI0
3		IIC_SDA2
4		—
5 to 15		—

16.4.5 Port 05 Function Assignment

Table 16.7 Port 05 function assignment

Function number (PFCn)	Port	P05_5	P05_6	P05_7
	IO Type	Type A	Type A	Type B
	Voltage domain	VCC33	VCC33	VCC33
0		—	IRQ12	ETH1_TXD2
1		A14	ETH1_RXER	A11
2		ESC_PHYLINK1	A13	—
3		GTIOC14B	GTIOC14A	CMTW1_TOC1
4		CMTW0_TOC1	CMTW1_TIC0	—
5		SPI_RSPCK2	SPI_SSL22	SPI_SSL23
6		HDSL1_LINK	HDSL1_MOSI2	—
7 to 15		—	—	—

16.4.6 Port 06 Function Assignment

Table 16.8 Port 06 function assignment (1 of 2)

Function number (PFCn)	Port	P06_0	P06_1	P06_2	P06_3	P06_4	P06_5	P06_6	P06_7
	IO Type	Type B	Type A	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33
0		ETH1_TXD3	ETH1_REFCLK	ETH1_TXD1	ETH1_TXD0	ETH1_TXCLK	ETH1_TXEN	ETH1_RXD0	ETH1_RXD1
1		A12	RMII1_REFCLK	A9	A8	A7	A6	A5	A4

Table 16.8 Port 06 function assignment (2 of 2)

Function number (PFCn)	Port	P06_0	P06_1	P06_2	P06_3	P06_4	P06_5	P06_6	P06_7
	IO Type	Type B	Type A	Type B	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33
2		GTIOC16A	A10	—	—	GTIOC11A	GTIOC11B	GTIOC12A	GTIOC12B
3		CMTW1_TOC0	GTIOC16B	CANRXDP1	—	SPI_MOSI1	SPI_MISO1	SPI_SSL10	SPI_SSL11
4		SS4#/CTS4#/ RTS4#	CTS4#	—	DE4	HDSL0_SEL2	HDSL0_MISO2	HDSL1_SMPL	DUEI1
5		SPI_SSL23	SPI_SSL22	—	—	—	—	—	HDSL1_LINK
6		CANRX1	CANTX1	—	CANTXDP1	—	—	—	—
7		HDSL0_MOSI1	HDSL0_CLK2	—	—	—	—	—	—
8 to 15		—	—	—	—	—	—	—	—

16.4.7 Port 07 Function Assignment

Table 16.9 Port 07 function assignment

Function number (PFCn)	Port	P07_0	P07_1	P07_2	P07_3	P07_4
	IO Type	Type B	Type B	Type B	Type B	Type C
	Voltage domain	VCC33	VCC33	VCC33	VCC33	VCC33
0		ETH1_RXD2	ETH1_RXD3	ETH1_RXDV	ETH1_RXCLK	IRQ1
1		A3	A2	A1	A0	ADTRG0#
2		GTIOC13A	GTIOC13B	GTIOC17A	GTIOC17B	USB_VBUSIN
3		TST_OUT1	SI1#	HDSL1_SEL1	SPI_SSL00	HDSL1_SEL2
4		HDSL1_SMPL	HDSL1_CLK1	—	HDSL1_MISO1	—
5 to 15		—	—	—	—	—

16.4.8 Port 08 Function Assignment

Table 16.10 Port 08 function assignment

Function number (PFCn)	Port	P08_4	P08_5	P08_6	P08_7
	IO Type	Type B	Type B	Type B	Type A
	Voltage domain	VCC33	VCC33	VCC33	VCC33
0		ETH0_RXD3	ETH0_RXDV	ETH0_RXCLK	GMAC_MDC
1		CS5#	MTIOC6B	MTIOC6C	—
2		MTIOC6A	MTIOC3C	MTIOC3B	WAIT#
3		MTIOC3A	GTIOC5A	GTIOC4B	ESC_MDC
4		GTIOC4A	RXD2/SCL2/MISO2	CMTW1_TIC1	MTIOC6D
5		SCK2	MCLK2	SCK3	MTIOC3D
6		SPI_SSL32	IRQ5	IRQ9	GTIOC5B
7		CANTXDP1	—	—	TXD2/SDA2/MOSI2
8		IRQ14	—	—	SPI_SSL13
9		HDSL1_MOSI1	—	—	MDAT2
10		—	—	—	IRQ8
11 to 15		—	—	—	—

16.4.9 Port 09 Function Assignment

Table 16.11 Port 09 function assignment

Function number (PFCn)	Port	P09_0	P09_1	P09_2	P09_3	P09_4	P09_5	P09_6	P09_7
	IO Type	Type A	Type A	Type A	Type B	Type B	Type B	Type B	Type B
	Voltage domain	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33
0		GMAC_MDIO	ETH0_REFCLK	IRQ0	ETH0_TXD3	ETH0_TXD2	ETH0_TXD1	ETH0_TXD0	ETH0_TXCLK
1		—	RMII0_REFCLK	ETH0_RXER	MTIOC7D	GTADSMP0	GTADSML1	GTIOC15A	GTIOC15B
2		CS0#	MTIOC7B	RAS#	MTIOC4D	TXD4/SDA4/MOSI4	CMTW0_TOC0	CMTW0_TIC1	RXD4/SCL4/MISO4
3		ESC_MDIO	MTIOC4C	MTIOC7C	GTETRGSB	SPI_SSL21	DE5	CTS5#	SPI_SSL00
4		MTIOC7A	GTETRGSA	MTIOC4B	GTIOC7B	HDSL1_MISO2	IIC_SCL1	CANRXDP0	CANTXDP0
5		MTIOC4A	GTIOC7A	GTIOC6B	CMTW0_TOC0	—	CANRX0	USB_EXICEN	DACK
6		GTIOC6A	DE3	DACK	SS5#/CTS5#/RTS5#	—	DREQ	MDAT3	USB_OVRCUR
7		RXD3/SCL3/MISO3	SPI_SSL10	RTCAT1HZ	CANTXDP0	—	USB_VBUSEN	ENCIFCK1	ENCIFOE1
8		—	GMAC_PTPOUT0	HDSL1_CLK2	USB_VBUSEN	—	IRQ14	IRQ13	IRQ12
9		—	ESC_SYNC0	—	MCLK3	—	HDSL1_MOSI2	—	—
10		—	ESC_SYNC1	—	CS3#	—	—	—	—
11		—	—	—	IRQ12	—	—	—	—
12		—	—	—	HDSL1_SEL2	—	—	—	—
13 to 15		—	—	—	—	—	—	—	—

16.4.10 Port 10 Function Assignment

Table 16.12 Port 10 function assignment

Function number (PFCn)	Port	P10_0	P10_1	P10_2	P10_3	P10_4
	IO Type	Type B	Type B	Type B	Type B	Type A
	Voltage domain	VCC33	VCC33	VCC33	VCC33	VCC33
0		ETH0_TXEN	ETH0_RXD0	ETH0_RXD1	ETH0_RXD2	IRQ11
1		POE11#	POE10#	TXD0/SDA0/MOSI0	RTCAT1HZ	—
2		CMTW0_TIC0	CTS3#	MTIC5U	RXD0/SCL0/MISO0	D15
3		SCK4	SPI_RSPCK1	—	MTIC5V	ESC_PHYLINK0
4		IIC_SDA1	ENCIFDI1	—	IRQ8	SPI_SSL01
5		CANTX0	IRQ10	—	—	SCK0
6		USB_EXICEN	—	—	—	MBX_HINT#
7		ENCIFDO1	—	—	—	MTIC5W
8		IRQ15	—	—	—	—
9 to 15		—	—	—	—	—

16.4.11 Port 13 Function Assignment

Table 16.13 Port 13 function assignment

Function number (PFCn)	Port	P13_2	P13_3	P13_4	P13_5	P13_6	P13_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VCC33	VCC33	VCC33	VCC1833_3	VCC1833_3	VCC1833_3
0		IRQ5	—	—	XSPI0_WP1#	XSPI0_WP0#	XSPI0_ECS1#
1		—	GMAC_PTPOUT3	ESC_RESETOUT#	GMAC_PTPTRG0	GMAC_PTPOUT0	GMAC_PTPTRG1
2		GMAC_PTPOUT2	TRACEDATA7	MTIOC0D	ESC_LATCH0	ESC_SYNC0	ESC_LATCH1
3		TRACEDATA6	D8	GTIOC8B	ESC_LATCH1	ESC_SYNC1	ESC_LATCH0
4		D9	ESC_I2CDATA	—	MTCLKA	MTCLKB	MTCLKC
5		ESC_I2CCLK	MTIOC0C	A0	SPI_RSPCK1	—	MBX_HINT#
6		MTIOC0A	MTIOC0B	WAIT#	IIC_SCL2	—	SPI_MOSI1
7		GTIOC10A	GTIOC10B	SPI_SSL12	—	—	HDSL0_SEL1
8		POE8#	CMTW1_TOC0	—	—	—	—
9		SS1#/CTS1#/RTS1#	CTS1#	—	—	—	—
10		SPI_MISO0	SPI_RSPCK0	—	—	—	—
11		IIC_SCL0	IIC_SDA0	—	—	—	—
12		MCLK4	MDAT4	—	—	—	—
13		—	—	—	—	—	—
14		A13	RD#	—	—	—	—
15		—	—	—	—	—	—

16.4.12 Port 14 Function Assignment

Table 16.14 Port 14 function assignment

Function number (PFCn)	Port	P14_0	P14_1	P14_2	P14_3	P14_4	P14_5	P14_6	P14_7
	IO Type	Type A	Type A	Type A	Type A	Type B	Type B	Type B	Type B
	Voltage domain	VCC1833_3	VCC1833_3	VCC1833_3	VCC1833_3	VCC1833_3	VCC1833_3	VCC1833_3	VCC1833_3
0		XSPI0_INT0#	XSPI0_INT1#	IRQ6	XSPI0_RSTO1#	XSPI0_DS	XSPI0_CKN	XSPI0_CKP	XSPI0_IO0
1		GMAC_PTPOUT1	—	XSPI0_ECS0#	—	BS#	CS3#	A21	A22
2		ESC_SYNC1	—	—	—	ESC_IRQ	POE8#	—	SCK5
3		ESC_SYNC0	MTIOC8A	—	—	MTIOC0B	HSPI_INT#	—	SPI_MISO1
4		MTCLKD	GTIOC8A	MTIOC8B	MTIOC0A	—	IRQ15	—	BS#
5		HDSL0_MISO1	—	GTIOC8B	—	SPI_SSL13	—	—	—
6		—	GMAC_PTPTRG1	—	ETH2_COL	POUTZ	—	—	—
7		—	ESC_LATCH0	ETH2_CRS	—	HDSL0_MISO2	—	—	—
8		—	ESC_LATCH1	—	SPI_SSL11	—	—	—	—
9		—	—	SPI_SSL10	POUTB	—	—	—	—
10		—	SPI_MISO1	POUTA	HDSL0_SEL2	—	—	—	—
11		—	HDSL0_MOSI1	HDSL0_CLK2	—	—	—	—	—
12 to 15		—	—	—	—	—	—	—	—

16.4.13 Port 15 Function Assignment

Table 16.15 Port 15 function assignment

Function number (PFCn)	Port	P15_0	P15_1	P15_2	P15_3	P15_4	P15_5	P15_6	P15_7
	IO Type	Type B	Type B	Type B	Type B	Type B	Type B	Type B	Type A
	Voltage domain	VCC1833_3	VCC1833_3	VCC1833_3	VCC1833_3	VCC1833_3	VCC1833_3	VCC1833_3	VCC1833_3
0		XSPI0_IO1	XSPI0_IO2	XSPI0_IO3	XSPI0_IO4	XSPI0_IO5	XSPI0_IO6	XSPI0_IO7	XSPI0_CS0#
1		A23	A24	A25	MTIOC8C	MTIOC8D	MCLK2	SPI_SSL12	CTS5#
2		RXD5/SCL5/ MISO5	MTIOC0C	MTIOC0D	MCLK1	MDAT1	D13	MDAT2	SPI_SSL13
3		SPI_MOSI1	TXD5/SDA5/ MOSI5	SS5#/CTS5#/ RTS5#	D11	D12	IRQ7	D14	TEND
4		CKE	SPI_SSL10	SPI_SSL11	NMI	IRQ3	—	—	—
5		—	CAS#	RAS#	—	—	—	—	—
6 to 15		—	—	—	—	—	—	—	—

16.4.14 Port 16 Function Assignment

Table 16.16 Port 16 function assignment

Function number (PFCn)	Port	P16_0	P16_1	P16_2	P16_3
	IO Type	Type A	Type A	Type A	Type A
	Voltage domain	VCC1833_3	VCC1833_3	VCC1833_3	VCC1833_3
0		XSPI0_CS1#	XSPI0_RESET0#	NMI	IRQ7
1		—	CMTW0_TOC1	XSPI0_RESET1#	XSPI0_RSTO0#
2		TXD0/SDA0/MOSI0	ADTRG0#	CTS0#	—
3		SPI_MOSI3	RXD0/SCL0/MISO0	SPI_RSPCK3	GTADSMP1
4		MCLK3	SPI_MISO3	USB_EXICEN	SCK0
5		ENCIFCK0	MDAT3	ENCIFCK1	SPI_SSL30
6		ETH2_REFCLK	ENCIFOE0	—	ENCIFOE1
7		—	CS2#	—	ETH2_RXER
8		HDSL0_MOSI2	—	SPI_SSL03	CS3#
9		—	HDSL1_SEL1	HDSL1_MISO1	—
10		—	—	—	HDSL1_MOSI1
11 to 15		—	—	—	—

16.4.15 Port 17 Function Assignment

Table 16.17 Port 17 function assignment (1 of 2)

Function number (PFCn)	Port	P17_0	P17_3	P17_4	P17_5	P17_6	P17_7
	IO Type	Type B	Type B	Type B	Type A	Type B	Type B
	Voltage domain	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33
0		ESC_IRQ	TRACECTL	TRACECLK	RSTOUT#	MTIOC3B	MTIOC4A
1		SS0#/CTS0#/RTS0#	GTETRGA	MTIOC3C	MTIOC3A	GTIOC1A	MTIOC4C
2		XSPI1_IO1	POE0#	GTETRGB	GTETRGC	SCK3	GTIOC2A
3		SPI_SSL01	ADTRG1#	GTIOC0A	GTIOC0B	—	GTIOC3A
4		HDSL1_MISO2	SPI_SSL31	CTS3#	TEND	XSPI1_DS	RXD3/SCL3/ MISO3
5		—	DREQ	SPI_SSL32	USB_OVRCUR	RD/WR#	DACK

Table 16.17 Port 17 function assignment (2 of 2)

Function number (PFCn)	Port	P17_0	P17_3	P17_4	P17_5	P17_6	P17_7
	IO Type	Type B	Type B	Type B	Type A	Type B	Type B
	Voltage domain	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33
6		—	—	—	—	—	—
7		—	XSPI1_IO2	XSPI1_IO3	—	—	XSPI1_CKP
8		—	CANRX1	DACK	—	—	RD#
9		—	—	SPI_RSPCK0	—	—	—
10		—	—	ETH2_TXER	—	—	—
11		—	—	ESC_PHYLINK2	—	—	—
12		—	—	HDSL1_CLK2	—	—	—
13 to 15		—	—	—	—	—	—

16.4.16 Port 18 Function Assignment

Table 16.18 Port 18 function assignment

Function number (PFCn)	Port	P18_0	P18_1	P18_2	P18_3	P18_4	P18_5	P18_6
	IO Type	Type B	Type A	Type A	Type B	Type B	Type B	Type B
	Voltage domain	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33
0		MTIOC4C	IRQ10	MTIOC4B	IRQ0	IRQ1	TRACECTL	IRQ11
1		MTIOC4A	MTIOC3D	MTIOC4D	MTIOC4D	MTIC5U	MTIC5V	TRACECLK
2		GTIOC3A	GTIOC1B	GTIOC2B	MTIOC4B	TXD4/SDA4/ MOSI4	RXD4/SCL4/ MISO4	MTIC5W
3		GTIOC2A	ADTRG1#	GTIOC3B	GTIOC3B	SPI_RSPCK2	SPI_MOSI2	ADTRG0#
4		TXD3/SDA3/ MOSI3	SS3#/CTS3#/ RTS3#	—	GTIOC2B	—	—	SCK4
5		WE0#/DQMLL	—	XSPI1_CS0#	CMTW1_TIC1	XSPI1_IO5	XSPI1_IO6	SPI_MISO2
6		—	WE1#/DQMLU	GMAC_PTPOUT3	CANRXDP1	ETH2_RXER	ETH2_COL	IIC_SCL2
7		—	—	BS#	—	CAS#	RAS#	—
8		XSPI1_IO0	—	SCK0	XSPI1_IO4	CANTX0	CANRX0	XSPI1_IO7
9		—	—	IIC_SDA2	ETH2_CRS	HSPI_IO1	HSPI_IO0	GMAC_PTPOUT2
10		—	—	—	CKE	DUEI0	TST_OUT0	DE4
11		—	—	—	DE3	HDSL0_LINK	HDSL0_SMPL	A15
12		—	—	—	—	—	—	HSPI_CK
13		—	—	—	—	—	—	SI0#
14		—	—	—	—	—	—	HDSL0_CLK1
15		—	—	—	—	—	—	—

16.4.17 Port 20 Function Assignment

Table 16.19 Port 20 function assignment (1 of 2)

Function number (PFCn)	Port	P20_3	P20_4
	IO Type	Type A	Type A
	Voltage domain	VCC33	VCC33
0		—	—
1		ESC_LEDERR	ESC_LINKACT1
2		GMAC_PTPOUT1	GMAC_PTPOUT0
3		CANTX1	—

Table 16.19 Port 20 function assignment (2 of 2)

Function number (PFCn)	Port	P20_3	P20_4
	IO Type	Type A	Type A
	Voltage domain	VCC33	VCC33
4 to 15		—	—

16.4.18 Port 21 Function Assignment

Table 16.20 Port 21 function assignment

Function number (PFCn)	Port	P21_1	P21_2	P21_3	P21_4	P21_5	P21_6	P21_7
	IO Type	Type A	Type A	Type A	Type A	Type A	Type A	Type A
	Voltage domain	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33	VCC33
0		TRACEDATA0	TRACEDATA1	TRACEDATA2	TRACEDATA3	IRQ6	IRQ9	IRQ10
1		D0	D1	D2	D3	TRACEDATA4	TRACEDATA5	TRACEDATA6
2		MTIOC6A	MTIOC6B	MTIOC6C	MTIOC6D	D4	D5	D6
3		GTIOC14A	GTIOC14B	GTIOC15A	GTIOC15B	MTIOC7A	MTIOC7B	MTIOC7C
4		CMTW0_TIC0	CMTW0_TIC1	TXD5/SDA5/ MOSI5	SS5#/CTS5#/ RTS5#	GTIOC16A	GTIOC16B	GTIOC17A
5		SCK5	RXD5/SCL5/ MISO5	SPI_SSL33	SPI_SSL02	CMTW1_TOC1	CTS0#	DE0
6		SPI_SSL20	SPI_MISO2	MCLK1	MDAT1	ADTRG1#	TEND	DREQ
7		IIC_SCL1	IIC_SDA1	—	ENCIFDO0	CTS5#	MDAT2	MCLK3
8		MCLK0	MDAT0	NMI	GMAC_PTPOUT1	SPI_MISO0	—	—
9		—	—	—	ESC_SYNC0	MCLK2	ESC_LINKACT0	HSPI_IO5
10		ESC_SYNC0	ESC_SYNC0	GTIOC14B	ESC_SYNC1	ENCIFDI0	HSPI_IO4	—
11		ESC_SYNC1	ESC_SYNC1	ESC_LEDRUN	—	—	—	—
12		HSPI_INT#	—	ESC_LEDSTER	MBX_HINT#	HSPI_IO3	—	—
13		—	GTIOC15A	HSPI_IO2	—	—	—	—
14		—	HSPI_CS#	—	—	—	—	—
15		—	—	—	—	—	—	—

16.4.19 Port 22 Function Assignment

Table 16.21 Port 22 function assignment

Function number (PFCn)	Port	P22_0	P22_1	P22_2	P22_3
	IO Type	Type A	Type A	Type A	Type A
	Voltage domain	VCC33	VCC33	VCC33	VCC33
0		IRQ15	TRACECTL	IRQ4	D10
1		TRACEDATA7	D8	TRACECLK	MTIOC8D
2		D7	ESC_LINKACT2	D9	GTETRGSB
3		MTIOC7D	POE4#	MTIOC8C	ENCIFDI1
4		GTIOC17B	SS4#/CTS4#/RTS4#	GTETRGSA	RXD5/SCL5/MISO5
5		DE5	—	SPI_SSL12	—
6		MDAT3	GTETRGB	ENCIFDO1	SPI_SSL00
7		—	SPI_MOSI0	—	USB_VBUSEN
8		HSPI_IO6	HSPI_IO7	MCLK1	—
9		—	IRQ13	—	—
10 to 15		—	—	—	—

16.4.20 Port 23 Function Assignment

Table 16.22 Port 23 function assignment

Function number (PFCn)	Port	P23_7
	IO Type	Type B
	Voltage domain	VCC1833_2
0		ETH2_RXD0
1		—
2		D11
3		BS#
4		MTIOC0A
5		GTETRGA
6		SCK1
7		MCLK4
8		—
9		—
10 to 15		—

16.4.21 Port 24 Function Assignment

Table 16.23 Port 24 function assignment

Function number (PFCn)	Port	P24_0	P24_1	P24_2
	IO Type	Type B	Type B	Type B
	Voltage domain	VCC1833_2	VCC1833_2	VCC1833_2
0		ETH2_RXD1	ETH2_RXCLK	ETH2_RXD2
1		—	—	—
2		D12	D13	D14
3		CKE	CAS#	RAS#
4		MTIOC0B	MTIOC0C	MTIOC0D
5		GTETRGB	GTETRGC	GTETRGD
6		RXD1/SCL1/MISO1	POE8#	TXD1/SDA1/MOSI1
7		DREQ	MCLK5	MDAT5
8		MDAT4	—	—
9		—	—	—
10		—	—	—
11 to 15		—	—	—

16.5 Handling of Unused Pins

Details on the handling of unused pins are given in Table 16.24.

Table 16.24 Handling of unused pins (1 of 2)

Pin name	Handling
AN000 to AN007, AN100 to AN115	Keep these pins open.
VREFH0	Connect this pin to VCC18_ADC0.
VREFH1	Connect this pin to VCC18_ADC1.
XTAL	Keep this pin open when external clock signal is used.
EXTAL	Connect this pin to VSS via a resistor (pulling down) when external clock signal is used.

Table 16.24 Handling of unused pins (2 of 2)

Pin name	Handling
EXTCLKIN	Connect this pin to VSS via a resistor (pulling down) when a crystal resonator is connected.
TRST#	Connect these pins to VSS via a resistor (pulling down), or input the same signal as that on the RES# pin.
TCK (P02_7)	Connect this pin to VSS via a resistor (pulling down).
TMS (P02_6)	Connect this pin to VCC33 via a resistor (pulling up).
TDI (P02_5)	Connect this pin to VCC33 via a resistor (pulling up).
TDO (P02_4)	Keep this pin open
RSTOUT# (P17_5)	Keep this pin open
MDX	— (Always connect this pin to VSS via a resister (pull down).)
BSCANP	— (Use this as a boundary scan enable pin.)
USB_DP, USB_DM	Connect these pins to GND via a resistor of 10 kΩ or keep these pins open with setting the DIRPD bit in the USBCTR register to 1.
USB_RREF	Keep this pin open.
Other Pins*1	Keep these pins open, connect them to VCC33 via a resistor (pulling up), or connect them to VSS via a resistor (pulling down).

Note 1. When handling them as unused pins, set the corresponding bits of Port m Mode Register (PMm: m = 00 to 24) to "Non-use (Hi-Z input protection)" which is the value after reset release. For P07_4 pin, set to "input" and connect it to VCC33 via a resister (pulling up), or connect it to VSS via a resister (pulling down).

16.6 Usage Notes

16.6.1 Release of Register Write Protection

PMCm, PFCm, DRCTLm, and RSELPm registers are write protected. To release write protection, follow the steps below.

In case of access to 0x8103_0xxx and RSELPm = 0 or in case of access to RSELPm,

1. Write 0x0000A504 to PRCRS register (Release write protection by PRC2 bit)
2. Write value to PMCm, PFCm, DRCTLm, or RSELPm registers
3. Write 0x0000A500 to PRCRS register (Set write protection by PRC2 bit)

In case of access to 0x800A_0xxx and RSELPm = 1,

1. Write 0x0000A504 to PRCRN register (Release write protection by PRC2 bit)
2. Write value to PMCm, PFCm, or DRCTLm registers
3. Write 0x0000A500 to PRCRN register (Set write protection by PRC2 bit)

16.6.2 Examples of Port Setting

The registers to be controlled are shown in Table 16.25.

Table 16.25 Port register configuration (1 of 2)

Register name		Bit name	Description
GPIO control register	Pm	POUT_n	Port register
	PMm	PMn	Port mode register
	PINm	PINn	Port input register
Multiplexed function select register	PMCm	PMCn	Port mode control register
	PFCm	PFCn	Port function control register

Table 16.25 Port register configuration (2 of 2)

Register name		Bit name	Description
I/O Buffer function switching register	DRCTLm	DRVn	Port driving ability control
		PUDn	Port Pull-Up/Down control
		SMTn	Port Schmitt trigger control
		SRn	Port slew rate control

Note: m: 00 to 24, n: 0 to 7

Table 16.26 is overview of the mode register settings for the pin function.

Table 16.26 Pin function configuration overview

Mode	PMCm.PMCn	PMm.PMn	Pm.POUT_n	PINm.PINn	I/O
General I/O input	0	01	x	Input value	I
General I/O output	0	1x	Output value	(Input value)	O
I/O port for peripheral functions*1	1	x	—	—	I/O

Note: m: 00 to 24, n: 0 to 7
Note 1. When PMCm.PMCn = 1, select pin function by PFCm.PFCn.

Figure 16.2 is an example flow of setting the Port Pm_n.

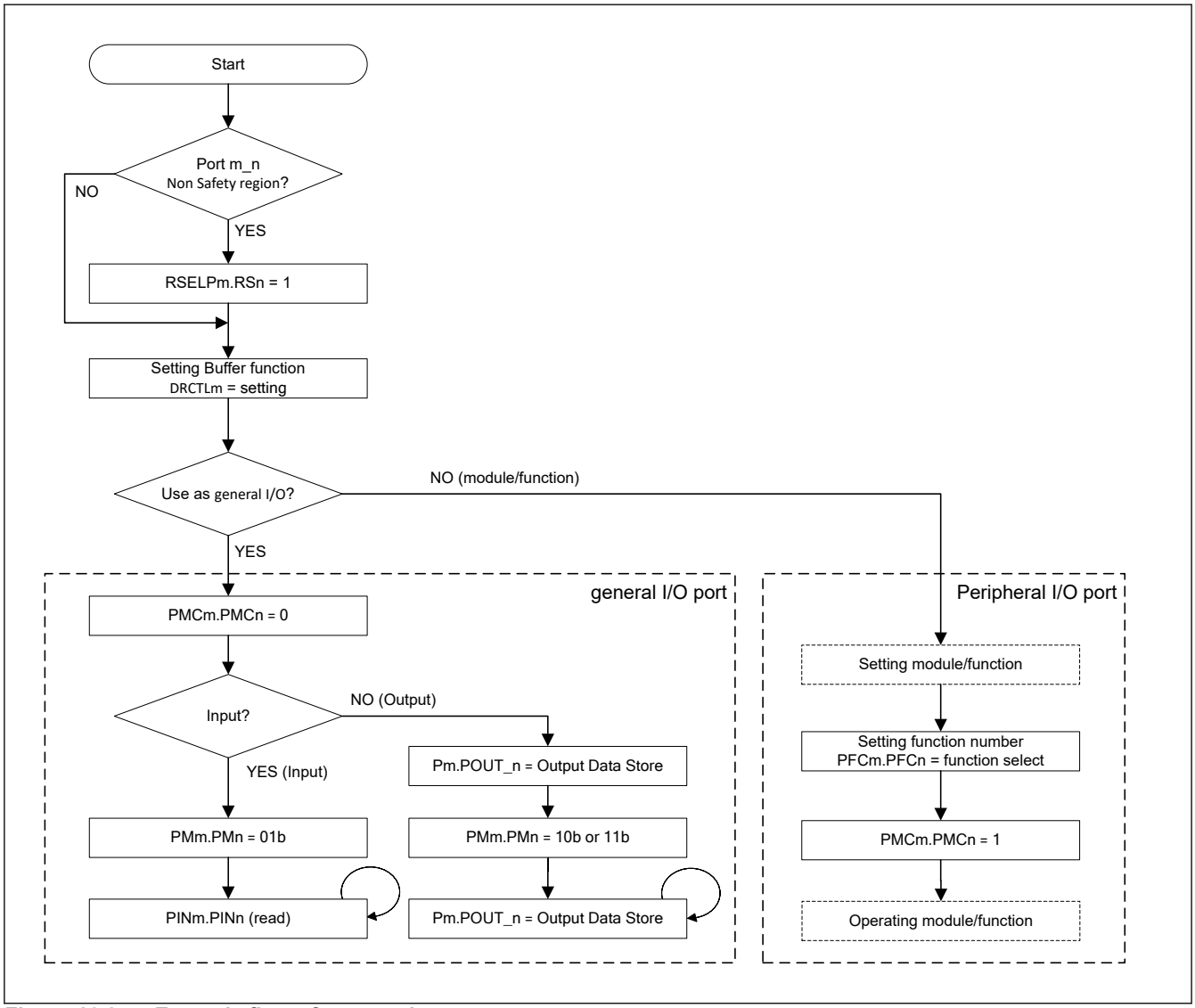


Figure 16.2 Example flow of port setting

16.6.3 Notes on Port Register Setting

1. If the multiplexed function is switched, a spike pulse may be generated in the signal of the on-chip peripheral function multiplexed with port due to the pin state immediately before the switching or other reasons. Take general measures shown below against spike pulses by software.
   - Switch the multiplexed function while the on-chip function is not working.
   - Clear the interrupt request flags, and then release the mask for pins multiplexed with interrupt signals.
   - Determine the output value, and then change the mode.
2. Only the allowed functions should be specified for the PFCm register. If a value that is not allowed for the register is specified, correct operation is not guaranteed.
3. Do not assign a single function to multiple pins through the PFCm registers.

16.6.4 Pin Assignment of each Boot Mode

Functional pin is shared with multiple ports for flexibility of pin assignment. However, pins to be used at booting are uniquely determined each boot mode and each package. These pin assignments are automatically set, and they can't be changed. PMCm and PFCm registers related to function pins at selected boot mode are changed from the initial values and stay after boot processing. If BSC_FLG at BSC boot mode or xSPI_FLG at xSPI boot mode is set to 0x2236_0679 in the boot parameter, these registers are changed back to the initial value after boot processing.

16.6.4.1 Pin Assignment for BSC Boot Mode

Table 16.27 Pins to be used for BSC boot mode (1 of 2)

Function pin	Pin assignment
A1	P07_2
A2	P07_1
A3	P07_0
A4	P06_7
A5	P06_6
A6	P06_5
A7	P06_4
A8	P06_3
A9	P06_2
A10	P06_1
A11	P05_7
A12	P06_0
A13	P05_6
A14	P05_5
A15	P02_3
A16	P02_2
A17	P02_1
A18	P02_0
A19	P01_7
A20	P01_6
D0	P21_1
D1	P21_2
D2	P21_3
D3	P21_4

Table 16.27 Pins to be used for BSC boot mode (2 of 2)

Function pin	Pin assignment
D4	P21_5
D5	P21_6
D6	P21_7
D7	P22_0
D8	P22_1
D9	P22_2
D10	P22_3
D11	P15_3
D12	P15_4
D13	P15_5
D14	P15_6
D15	P10_4
CS0#	P09_0
RD#	P17_7

16.6.4.2 Pin Assignment for xSPI Boot Mode

Table 16.28 Pins to be used for xSPI0 boot mode

Function pin	Pin assignment
XSPI0_DS	P14_4
XSPI0_CKP	P14_6
XSPI0_CKN	P14_5
XSPI0_IO0	P14_7
XSPI0_IO1	P15_0
XSPI0_IO2	P15_1
XSPI0_IO3	P15_2
XSPI0_IO4	P15_3
XSPI0_IO5	P15_4
XSPI0_IO6	P15_5
XSPI0_IO7	P15_6
XSPI0_CS0#	P15_7
XSPI0_RESET0#	P16_1

Table 16.29 Pins to be used for xSPI1 boot mode

Function pin	Pin assignment
XSPI1_CKP	P17_7
XSPI1_IO0	P18_0
XSPI1_IO1	P17_0
XSPI1_CS0#	P18_2

16.6.4.3 Pin Assignment for SCI Boot Mode

Table 16.30 Pins to be used for SCI boot mode

Function pin	Pin assignment
TXD0	P16_0
RXD0	P16_1

16.6.4.4 Pin Assignment for Serial Host Interface Boot Mode

Table 16.31 Pins to be used for serial host interface boot mode

Function pin	Pin assignment
HSPI_CK	P18_6
HSPI_IO0	P18_5
HSPI_IO1	P18_4
HSPI_IO2	P21_3*1
HSPI_IO3	P21_5*1
HSPI_IO4	P21_6*2
HSPI_IO5	P21_7*2
HSPI_IO6	P22_0*2
HSPI_IO7	P22_1*2
HSPI_CS#	P21_2
MBX_HINT#	P21_4

Note 1. Activated when SPIMODE[1:0] bits of SHBMS area in OTP are 10b or 11b.
Note 2. Activated when SPIMODE[1:0] bits of SHBMS area in OTP are 11b.

17. Multi-Function Timer Pulse Unit 3 (MTU3)

17.1 Overview

This LSI has an on-chip multi-function timer pulse unit 3 (MTU3), consisting of eight 16-bit timer channels and one 32-bit timer channel.

Table 17.1 shows the specifications of the MTU and Table 17.2 lists the functions of the MTU. Figure 17.1 and Figure 17.2 show block diagrams of the MTU.

Table 17.1 MTU specifications

Item	Description
Pulse input/output	28 lines maximum
Pulse input	3 lines
Count clock	11 clocks for each channel (14 clocks for MTU0, 12 clocks for MTU1 and MTU2, 10 clocks for MTU5)
Operating frequency	PCLKH (150 or 200 MHz)
Available operations	[MTU0 to MTU4, MTU6, MTU7, MTU8] ● Waveform output on compare match ● Input capture function (noise filter setting available) ● Counter-clearing operation ● Simultaneous writing to multiple timer counters (TCNT) (excluding MTU8) ● Simultaneous clearing on compare match or input capture (excluding MTU8) ● Simultaneous input and output to registers in synchronization with counter operations (excluding MTU8) ● Up to 12-phase PWM output in combination with synchronous operation (excluding MTU8)
	[MTU0, MTU3, MTU4, MTU6, MTU7, MTU8] ● Buffer operation specifiable
	[MTU1, MTU2] ● Phase counting mode can be specified independently ● 32-bit phase counting mode can be specified for interlocked operation of MTU1 and MTU2 (when TMDR3.LWA = 1) ● Cascade connection operation available
	[MTU3, MTU4, MTU6, MTU7] ● Through interlocked operation of MTU3/4 and MTU6/7, the positive and negative signals in six phases (12 phases in total) can be output in complementary PWM and reset-synchronized PWM operation. ● In complementary PWM mode, transfer of values from buffer registers to temporary registers on crests and troughs of the timer-counter values or writing to the buffer registers (MTU4.TGRD and MTU7.TGRD) ● Double-buffering selectable in complementary PWM mode
	[MTU3, MTU4] ● Through interlocking with MTU0, a mode for driving AC synchronous motors (brushless DC motors) by using complementary PWM output and reset PWM output is settable and allows the selection of two types of waveform output (chopping or level)
	[MTU5] ● Capable of operation as a dead-time compensation counter
	[MTU0/MTU5, MTU1, MTU2, MTU8] ● 32-bit phase counting mode specifiable by combining MTU1 and MTU2 and through interlocked operation with MTU0/MTU5 and MTU8
Interrupt skipping function	● In complementary PWM mode, interrupts on crests and troughs of counter values and triggers to start conversion by the A/D converter can be skipped
Interrupt sources	43 sources
Buffer operation	Automatic transfer of register data (transfer from the buffer register to the timer register)
Trigger generation	A/D converter start triggers can be generated
	A/D converter start request delaying function enables A/D converter to be started with any desired timing and to be synchronized with PWM output
Low power consumption function	Module stop mode can be set

Table 17.2 MTU functions (1 of 3)

✓: Possible —: Not possible

Parameter		MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4	MTU5	MTU6	MTU7	MTU8
Count clock*1		PCLKH/1 PCLKH/2 PCLKH/4 PCLKH/8 PCLKH/16 PCLKH/32 PCLKH/64 PCLKH/256 PCLKH/1024 MTCLKA MTCLKB MTCLKC MTCLKD MTIOC1A	PCLKH/1 PCLKH/2 PCLKH/4 PCLKH/8 PCLKH/16 PCLKH/32 PCLKH/64 PCLKH/256 PCLKH/1024 MTCLKA MTCLKB	PCLKH/1 PCLKH/2 PCLKH/4 PCLKH/8 PCLKH/16 PCLKH/32 PCLKH/64 PCLKH/256 PCLKH/1024 MTCLKA MTCLKB MTCLKC	MTCLKA MTCLKB MTCLKC MTCLKD	PCLKH/1 PCLKH/2 PCLKH/4 PCLKH/8 PCLKH/16 PCLKH/32 PCLKH/64 PCLKH/256 PCLKH/1024 MTCLKA MTCLKB	PCLKH/1 PCLKH/2 PCLKH/4 PCLKH/8 PCLKH/16 PCLKH/32 PCLKH/64 PCLKH/256 PCLKH/1024 MTCLKA MTCLKB	PCLKH/1 PCLKH/2 PCLKH/4 PCLKH/8 PCLKH/16 PCLKH/32 PCLKH/64 PCLKH/256 PCLKH/1024 MTIOC1A	PCLKH/1 PCLKH/2 PCLKH/4 PCLKH/8 PCLKH/16 PCLKH/32 PCLKH/64 PCLKH/256 PCLKH/1024 MTCLKA MTCLKB	PCLKH/1 PCLKH/2 PCLKH/4 PCLKH/8 PCLKH/16 PCLKH/32 PCLKH/64 PCLKH/256 PCLKH/1024 MTCLKA MTCLKB	PCLKH/1 PCLKH/2 PCLKH/4 PCLKH/8 PCLKH/16 PCLKH/32 PCLKH/64 PCLKH/256 PCLKH/1024 MTCLKA MTCLKB
General registers (TGR)		TGRA TGRB TGRE	TGRA TGRB TGRALW TGRBLW	TGRA TGRB	TGRALW TGRBLW	TGRA TGRB	TGRA TGRB	TGRU TGRV TGRW	TGRA TGRB	TGRA TGRB	TGRA TGRB
General registers/ buffer registers		TGRC TGRD TGRF	—	—	—	TGRC TGRD TGRE	TGRC TGRD TGRE TGRF	—	TGRC TGRD TGRE	TGRC TGRD TGRE TGRF	TGRC TGRD
I/O pins		MTIOC0A MTIOC0B MTIOC0C MTIOC0D	MTIOC1A MTIOC1B	MTIOC2A MTIOC2B	MTIOC1A MTIOC1B	MTIOC3A MTIOC3B MTIOC3C MTIOC3D	MTIOC4A MTIOC4B MTIOC4C MTIOC4D	MTIC5U MTIC5V MTIC5W	MTIOC6A MTIOC6B MTIOC6C MTIOC6D	MTIOC7A MTIOC7B MTIOC7C MTIOC7D	MTIOC8A MTIOC8B MTIOC8C MTIOC8D
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGRALW/ TGRBLW compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	✓	✓	✓	—	✓	✓	—	✓	✓	✓
	1 output	✓	✓	✓	—	✓	✓	—	✓	✓	✓
	Toggle output	✓	✓	✓	—	✓	✓	—	✓	✓	✓
Input capture function		✓	✓	✓	✓*1	✓	✓	✓	✓	✓	✓*2
Synchronous operation		✓	✓	✓	—	✓	✓	—	✓	✓	—
PWM mode 1		✓	✓	✓	—	✓	✓	—	✓	✓	—
PWM mode 2		✓	✓	✓	—	—	—	—	—	—	—
Complementary PWM mode		—	—	—	—	✓	✓	—	✓	✓	—
Reset-synchronized PWM mode		—	—	—	—	✓	✓	—	✓	✓	—
AC synchronous motor drive mode		✓	—	—	—	✓	✓	—	—	—	—
Phase counting mode		—	✓	✓	✓	—	—	—	—	—	—
Buffer operation		✓	—	—	—	✓	✓	—	✓	✓	✓
Dead time compensation counter function		—	—	—	—	—	—	✓	—	—	—
DMAC activation		TGRm compare match or input capture (m = A to D)	TGRm compare match or input capture (m = A, B)	TGRm compare match or input capture (m = A, B)	TGRALW/ TGRBLW input capture	TGRm compare match or input capture (m = A to D)	TGRm compare match or input capture (m = A to D) and TCNT overflow/ underflow (only in complemen -tary PWM mode)	TGRm compare match or input capture (m = U, V, W)	TGRm compare match or input capture (m = A to D)	TGRm compare match or input capture (m = A to D) or TCNT overflow/ underflow (only in complemen tary PWM mode)	TGRm compare match or input capture (m = A to D)

Table 17.2 MTU functions (2 of 3)

✓: Possible —: Not possible

Parameter	MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4	MTU5	MTU6	MTU7	MTU8
A/D converter start trigger	TGRA compare match or input capture TGRE compare match	TGRA compare match or input capture	TGRA compare match or input capture	TGRALW input capture	TGRA compare match or input capture	TGRA compare match or input capture, or TCNT underflow (trough) in complementary PWM mode	—	TGRA compare match or input capture	TGRA compare match or input capture, or TCNT underflow (trough) in complementary PWM mode	—
Interrupt sources	Seven sources	Four sources	Four sources	Four sources	Five sources	Five sources	Three sources	Five sources	Five sources	Five sources
	Overflow	Overflow	Overflow	Overflow	Overflow	Overflow or underflow (only in complementary PWM mode)	—	Overflow	Overflow or underflow (only in complementary PWM mode)	Overflow
	—	Underflow	Underflow	Underflow	—		—	—	—	—
	Compare match or input capture 0A	Compare match or input capture 1A	Compare match or input capture 2A	Input capture 1A	Compare match or input capture 3A	Compare match or input capture 4A	Compare match or input capture 5U	Compare match or input capture 6A	Compare match or input capture 7A	Compare match or input capture 8A
	Compare match or input capture 0B	Compare match or input capture 1B	Compare match or input capture 2B	Input capture 1B	Compare match or input capture 3B	Compare match or input capture 4B	Compare match or input capture 5V	Compare match or input capture 6B	Compare match or input capture 7B	Compare match or input capture 8B
	Compare match or input capture 0C	—	—	—	Compare match or input capture 3C	Compare match or input capture 4C	Compare match or input capture 5W	Compare match or input capture 6C	Compare match or input capture 7C	Compare match or input capture 8C
	Compare match or input capture 0D	—	—	—	Compare match or input capture 3D	Compare match or input capture 4D	—	Compare match or input capture 6D	Compare match or input capture 7D	Compare match or input capture 8D
	Compare match 0E	—	—	—	—	—	—	—	—	—
	Compare match 0F	—	—	—	—	—	—	—	—	—
Event link function (output)	Seven sources	—	—	—	Five sources	Six sources	—	Five sources	Six sources	—
	Overflow				Overflow	Overflow		Overflow	Overflow	
	—				—	Underflow		—	Underflow	
	Compare match 0A				Compare match 3A	Compare match 4A		Compare match 6A	Compare match 7A	
	Compare match 0B				Compare match 3B	Compare match 4B		Compare match 6B	Compare match 7B	
	Compare match 0C				Compare match 3C	Compare match 4C		Compare match 6C	Compare match 7C	
	Compare match 0D				Compare match 3D	Compare match 4D		Compare match 6D	Compare match 7D	
	Compare match 0E				—	—		—	—	
	Compare match 0F				—	—		—	—	
Event link function (input)	Start counting	—	—	—	Start counting	Start counting	—	—	—	—
	Input capture (to be captured in the TGRA)				Input capture (to be captured in the TGRA)	Input capture (to be captured in the TGRA)				
	Restart counting (clear counter)				Restart counting (clear counter)	Restart counting (clear counter)				

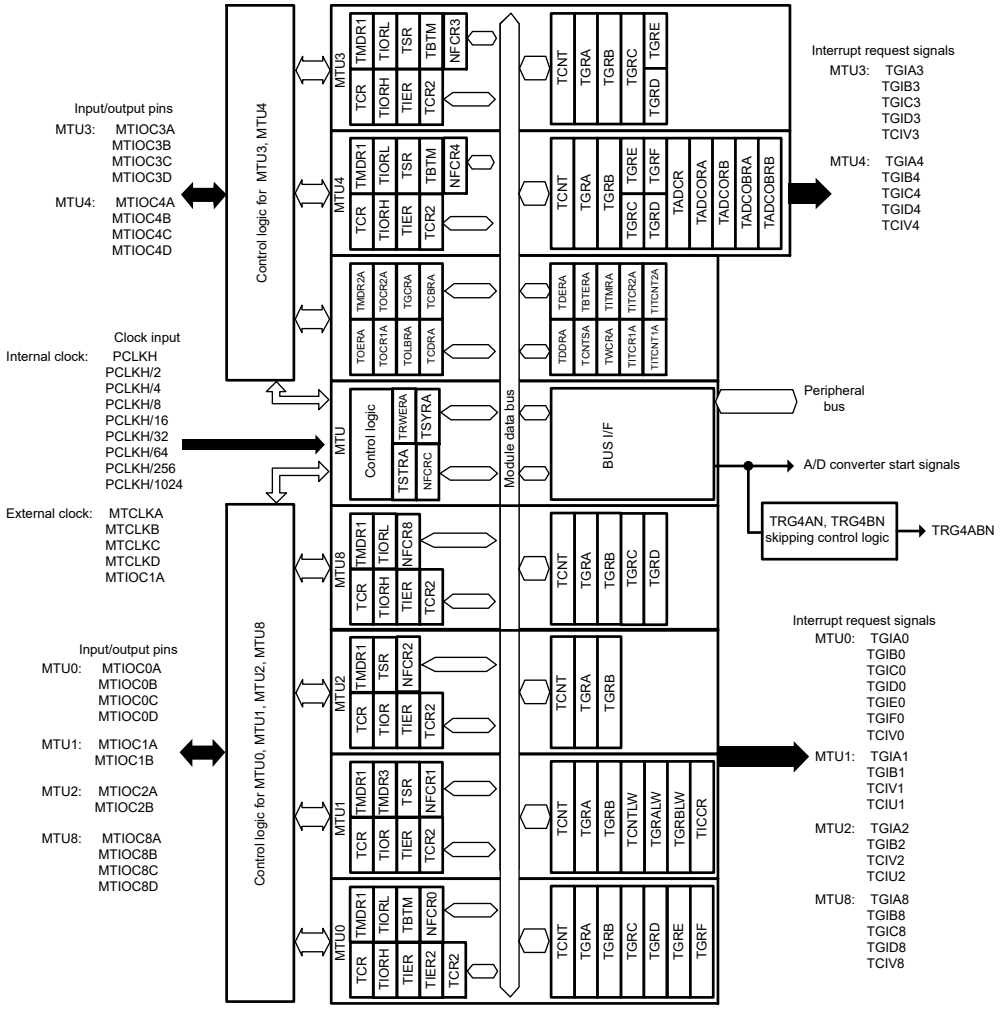
Table 17.2 MTU functions (3 of 3)

✓: Possible —: Not possible

Parameter	MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4	MTU5	MTU6	MTU7	MTU8
A/D converter start request delaying function	—	—	—	—	—	A/D converter start request at a match between TADCORA and TCNT or A/D converter start request at a match between TADCORB and TCNT	—	—	A/D converter start request at a match between TADCORA and TCNT or A/D converter start request at a match between TADCORB and TCNT	—
Interrupt skipping 1	—	—	—	—	Skips TGRA compare match interrupts	Skips TCIV interrupts	—	Skips TGRA compare match interrupts	Skips TCIV interrupts	—
Interrupt skipping 2	—	—	—	—	—	Skipping in compare count between TADCORA and TCNT, and TADCORB and TCNT	—	—	Skipping in compare count between TADCORA and TCNT, and TADCORB and TCNT	—
Module stop function	Module stop setting by the MSTPCRC.MSTPCRC00 bit									

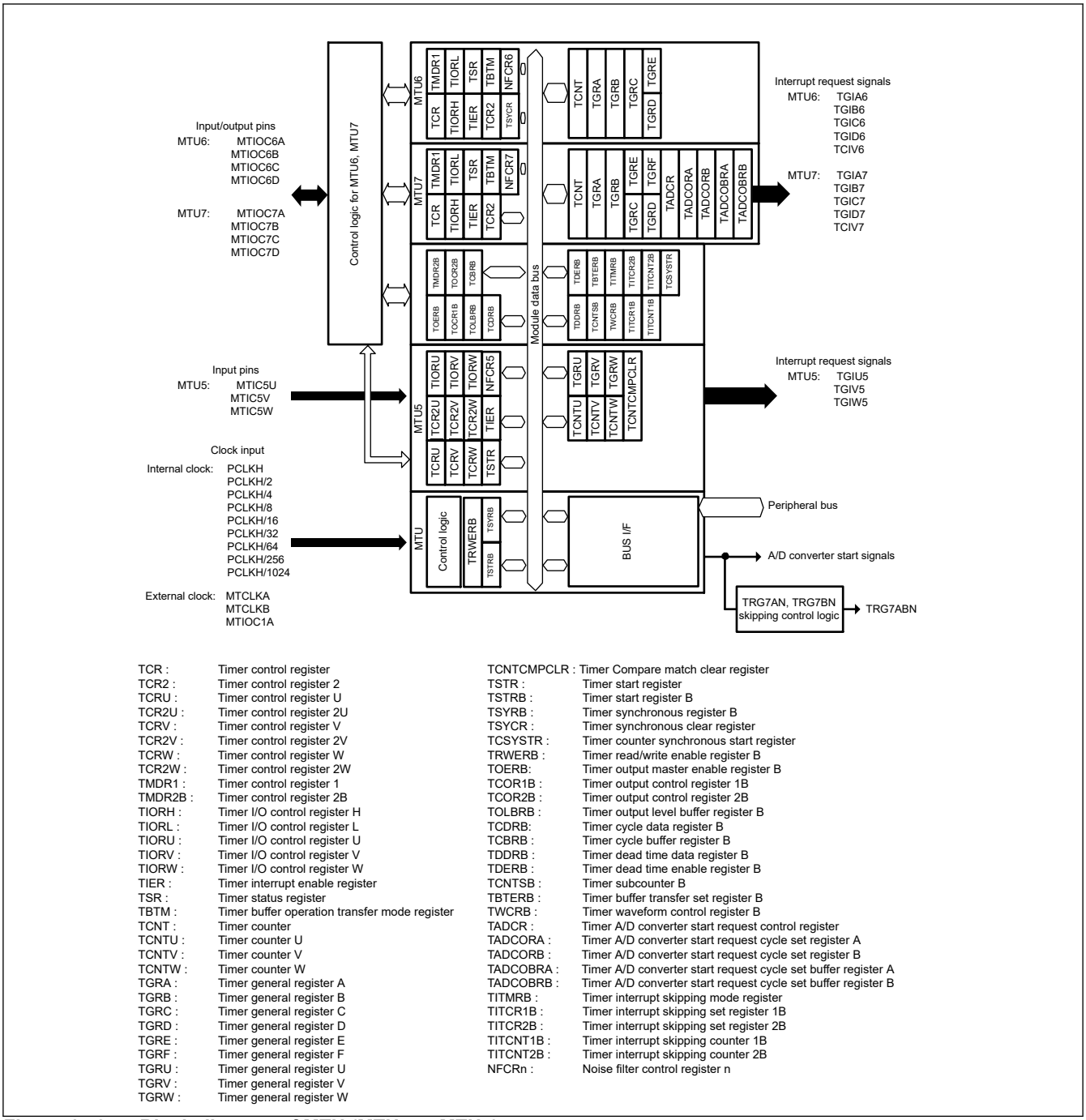
Note 1. When LWA is 1, the TGRALW capture source can be selected from either of the following an input from MTIOC1A or MTU0.TRGA compare match/input capture event. The TGRBLW capture source can be selected from any of the following: an input from MTIOC1B, MTU0.TRGC compare match/input capture event, or MTU8.TGRC compare match event.

Note 2. Capture in MTU8 is supported only in normal mode.



TCR :	Timer control register	TRWERA :	Timer read/write enable register
TCR2 :	Timer control register 2	TOERA :	Timer output master enable register A
TMDR1 :	Timer mode register 1	TOCR1A :	Timer output control register 1A
TMDR2A :	Timer mode register 2A	TOCR2A :	Timer output control register 2A
TMDR3 :	Timer mode register 3	TOLBRA :	Timer output level buffer register A
TIOR :	Timer I/O control register	TGCRA :	Timer gate control register A
TIORH :	Timer I/O control register H	TCNTSA :	Timer subcounter A
TIORL :	Timer I/O control register L	TCDRA :	Timer cycle data register A
TIER :	Timer interrupt enable register	TCBRA :	Timer cycle buffer register A
TIER2 :	Timer interrupt enable register 2	TDDRA :	Timer dead time data register A
TSR :	Timer status register	TDERA :	Timer dead time enable register A
TBTM :	Timer buffer operation transfer mode register	TBTERA :	Timer buffer transfer set register A
TCNT :	Timer counter	TWCRA :	Timer waveform control register A
TCNTLW :	Timer longword counter	TICCR :	Timer input capture control register
TGRA :	Timer general register A	TADCR :	Timer A/D converter start request control register
TGRALW :	Timer longword general register A	TADCORA :	Timer A/D converter start request cycle set register A
TGRB :	Timer general register B	TADCORB :	Timer A/D converter start request cycle set register B
TGRBLW :	Timer longword general register B	TADCOBRA :	Timer A/D converter start request cycle set buffer register A
TGRC :	Timer general register C	TADCOBRB :	Timer A/D converter start request cycle set buffer register B
TGRD :	Timer general register D	TITMRA :	Timer interrupt skipping mode register A
TGRE :	Timer general register E	TITCR1A :	Timer interrupt skipping set register 1A
TGRF :	Timer general register F	TITCR2A :	Timer interrupt skipping set register 2A
TSTRA :	Timer start register A	TITCNT1A :	Timer interrupt skipping counter 1A
TSYRA :	Timer synchronous register A	TITCNT2A :	Timer interrupt skipping counter 2A
TCSYSTR :	Timer counter synchronous start register	NFCRn :	Noise filter control register n

Figure 17.1 Block diagram of MTU (MTU0 to MTU4, MTU8)



TCR : Timer control register
TCR2 : Timer control register 2
TCRU : Timer control register U
TCR2U : Timer control register 2U
TCRV : Timer control register V
TCR2V : Timer control register 2V
TCRW : Timer control register W
TCR2W : Timer control register 2W
TMDR1 : Timer control register 1
TMDR2B : Timer control register 2B
TIORH : Timer I/O control register H
TIORL : Timer I/O control register L
TIORU : Timer I/O control register U
TIORV : Timer I/O control register V
TIORW : Timer I/O control register W
TIER : Timer interrupt enable register
TSR : Timer status register
TBTM : Timer buffer operation transfer mode register
TCNT : Timer counter
TCNTU : Timer counter U
TCNTV : Timer counter V
TCNTW : Timer counter W
TGRA : Timer general register A
TGRB : Timer general register B
TGRC : Timer general register C
TGRD : Timer general register D
TGRE : Timer general register E
TGRF : Timer general register F
TGRU : Timer general register U
TGRV : Timer general register V
TGRW : Timer general register W

TCNTCMPCLR : Timer Compare match clear register
TSTR : Timer start register
TSTRB : Timer start register B
TSYRB : Timer synchronous register B
TSYCR : Timer synchronous clear register
TCSYSTR : Timer counter synchronous start register
TRWERB : Timer read/write enable register B
TOERB: Timer output master enable register B
TCOR1B : Timer output control register 1B
TCOR2B : Timer output control register 2B
TOLBRB : Timer output level buffer register B
TCDRB: Timer cycle data register B
TCBRB : Timer cycle buffer register B
TDDRB : Timer dead time data register B
TDERB : Timer dead time enable register B
TCNTSB : Timer subcounter B
TBTERB : Timer buffer transfer set register B
TWCRB : Timer waveform control register B
TADCR : Timer A/D converter start request control register
TADCORA : Timer A/D converter start request cycle set register A
TADCORB : Timer A/D converter start request cycle set register B
TADCOBRA : Timer A/D converter start request cycle set buffer register A
TADCOBRB : Timer A/D converter start request cycle set buffer register B
TITMRB : Timer interrupt skipping mode register
TITCR1B : Timer interrupt skipping set register 1B
TITCR2B : Timer interrupt skipping set register 2B
TITCNT1B : Timer interrupt skipping counter 1B
TITCNT2B : Timer interrupt skipping counter 2B
NFCRn : Noise filter control register n

Figure 17.2 Block diagram of MTU (MTU5 to MTU7)

Table 17.3 shows the configuration of pins for the MTU.

Table 17.3 Pin configuration of the MTU (1 of 2)

Channel	Pin name	I/O	Function
MTU	MTCLKA	Input	External clock A input pin (MTU1 and MTU2 phase counting mode A phase input)
	MTCLKB	Input	External clock B input pin (MTU1 and MTU2 phase counting mode B phase input)
	MTCLKC	Input	External clock C input pin (MTU2 phase counting mode A phase input)
	MTCLKD	Input	External clock D input pin (MTU2 phase counting mode B phase input)

Table 17.3 Pin configuration of the MTU (2 of 2)

Channel	Pin name	I/O	Function
MTUm (m = 0, 3, 4, 6, 7)	MTIOCmA	I/O	MTUm.TGRA input capture input/output compare output/PWM output pin
	MTIOCmB	I/O	MTUm.TGRB input capture input/output compare output/PWM output pin
	MTIOCmC	I/O	MTUm.TGRC input capture input/output compare output/PWM output pin
	MTIOCmD	I/O	MTUm.TGRD input capture input/output compare output/PWM output pin
MTUn (n = 1, 2)	MTIOCnA	I/O	MTUn.TGRA input capture input/output compare output/PWM output pin
	MTIOCnB	I/O	MTUn.TGRB input capture input/output compare output/PWM output pin
MTU5	MTIC5U	Input	MTU5.TGRU input capture input/external pulse input pin
	MTIC5V	Input	MTU5.TGRV input capture input/external pulse input pin
	MTIC5W	Input	MTU5.TGRW input capture input/external pulse input pin
MTU8	MTIOC8A	I/O	MTU8.TGRA input capture input/output compare output pin
	MTIOC8B	I/O	MTU8.TGRB input capture input/output compare output pin
	MTIOC8C	I/O	MTU8.TGRC input capture input/output compare output pin
	MTIOC8D	I/O	MTU8.TGRD input capture input/output compare output pin

Table 17.4 MTU interrupt sources (1 of 2)

Channel	Name	Interrupt sources	CPU0 GIC request	DMAC activation
MTU0	TGIA0	MTU0.TGRA input capture/compare match	Possible	Possible
	TGIB0	MTU0.TGRB input capture/compare match	Possible	Possible
	TGIC0	MTU0.TGRC input capture/compare match	Possible	Possible
	TGID0	MTU0.TGRD input capture/compare match	Possible	Possible
	TCIV0	MTU0.TCNT overflow	Possible	Not possible
	TGIE0	MTU0.TGRE compare match	Possible	Not possible
	TGIF0	MTU0.TGRF compare match	Possible	Not possible
MTU1	TGIA1	MTU1.TGRA input capture/compare match	Possible	Possible
	TGIB1	MTU1.TGRB input capture/compare match	Possible	Possible
	TCIV1	MTU1.TCNT overflow	Possible	Not possible
	TCIU1	MTU1.TCNT underflow	Possible	Not possible
MTU2	TGIA2	MTU2.TGRA input capture/compare match	Possible	Possible
	TGIB2	MTU2.TGRB input capture/compare match	Possible	Possible
	TCIV2	MTU2.TCNT overflow	Possible	Not possible
	TCIU2	MTU2.TCNT underflow	Possible	Not possible
MTU3	TGIA3	MTU3.TGRA input capture/compare match	Possible	Possible
	TGIB3	MTU3.TGRB input capture/compare match	Possible	Possible
	TGIC3	MTU3.TGRC input capture/compare match	Possible	Possible
	TGID3	MTU3.TGRD input capture/compare match	Possible	Possible
	TCIV3	MTU3.TCNT overflow	Possible	Not possible
MTU4	TGIA4	MTU4.TGRA input capture/compare match	Possible	Possible
	TGIB4	MTU4.TGRB input capture/compare match	Possible	Possible
	TGIC4	MTU4.TGRC input capture/compare match	Possible	Possible
	TGID4	MTU4.TGRD input capture/compare match	Possible	Possible
	TCIV4	MTU4.TCNT overflow/underflow*1	Possible	Possible

Table 17.4 MTU interrupt sources (2 of 2)

Channel	Name	Interrupt sources	CPU0 GIC request	DMAC activation
MTU5	TGIU5	MTU5.TGRU input capture/compare match	Possible	Possible
	TGIV5	MTU5.TGRV input capture/compare match	Possible	Possible
	TGIW5	MTU5.TGRW input capture/compare match	Possible	Possible
MTU6	TGIA6	MTU6.TGRA input capture/compare match	Possible	Possible
	TGIB6	MTU6.TGRB input capture/compare match	Possible	Possible
	TGIC6	MTU6.TGRC input capture/compare match	Possible	Possible
	TGID6	MTU6.TGRD input capture/compare match	Possible	Possible
	TCIV6	MTU6.TCNT overflow	Possible	Not possible
MTU7	TGIA7	MTU7.TGRA input capture/compare match	Possible	Possible
	TGIB7	MTU7.TGRB input capture/compare match	Possible	Possible
	TGIC7	MTU7.TGRC input capture/compare match	Possible	Possible
	TGID7	MTU7.TGRD input capture/compare match	Possible	Possible
	TCIV7	MTU7.TCNT overflow/underflow*1	Possible	Possible
MTU8	TGIA8	MTU8.TGRA input capture/compare match	Possible	Possible
	TGIB8	MTU8.TGRB input capture/compare match	Possible	Possible
	TGIC8	MTU8.TGRC input capture/compare match	Possible	Possible
	TGID8	MTU8.TGRD input capture/compare match	Possible	Possible
	TCIV8	MTU8.TCNT overflow	Possible	Not possible

Note 1. Only in complementary PWM mode.

17.2 Register Descriptions

17.2.1 TCR : Timer Control Register

Base address: MTU0 = 0x9000_1300, MTU1 = 0x9000_1380, MTU2 = 0x9000_1400, MTU3 = 0x9000_1100
MTU4 = 0x9000_1200, MTU6 = 0x9000_1900, MTU7 = 0x9000_1A00, MTU8 = 0x9000_1600

Offset address: 0x000 (MTU0, MTU1, MTU2, MTU8), 0x100 (MTU3, MTU6), 0x001 (MTU4, MTU7)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CCLR[2:0]			CKEG[1:0]		TPSC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TPSC[2:0]	Time Prescaler Select See section 17.2.3. TCR2 — Timer Control Registers.	R/W
4:3	CKEG[1:0]	Clock Edge Select 0 0: Count at rising edge 0 1: Count at falling edge 1 x: Count at both edges	R/W
7:5	CCLR[2:0]	Counter Clear Source Select See Table 17.5 and Table 17.6.	R/W

The TCR register controls the TCNT operation for each channel in combination with the TCR2 register. TCR values should be specified only while TCNT operation is stopped.

TPSC[2:0] bits (Time Prescaler Select)

These bits select the TCNT counter clock. The count clock source can be selected independently for each channel. For details, see section 17.2.3. TCR2 — Timer Control Registers.

CKEG[1:0] bits (Clock Edge Select)

The CKEG[1:0] bits select the input clock edge, including the MTIOC1A pin. When the input clock is counted at both edges, the input clock period is halved (e.g. PCLKH/4 at both edges = PCLKH/2 at rising edge). If phase counting mode is used on MTU1 and MTU2, the setting of these bits is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the input clock is PCLKH/2 or slower. When PCLKH/1 or the overflow/underflow in another channel is selected for the input clock, a value can be written to these bits but counter operation compiles with the initial value.

CCLR[2:0] bits (Counter Clear Source Select)

These bits select the TCNT counter clearing source. See Table 17.5 and Table 17.6 for details.

Table 17.5 CCLR[2:0] (MTU0, MTU3, MTU4, MTU6, MTU7, MTU8)

Channel	Bit 7 CCLR[2]	Bit 6 CCLR[1]	Bit 5 CCLR[0]	Description
MTU0 MTU3 MTU4 MTU6 MTU7 MTU8	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture
	0	1	0	TCNT cleared by TGRB compare match/input capture
	0	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/synchronous operation*1
	1	0	0	TCNT clearing disabled
	1	0	1	TCNT cleared by TGRC compare match/input capture*2
	1	1	0	TCNT cleared by TGRD compare match/input capture*2
	1	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/synchronous operation*1

Note 1. Synchronous operation is selected by setting the TSYRA.SYNC or TSYRB.SYNC bit to 1 except for MTU8.
Note 2. When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority and compare match/input capture does not occur.

Table 17.6 CCLR[2:0] (MTU1 and MTU2)

Channel	Bit 7 Reserved*2	Bit 6 CCLR[1]	Bit 5 CCLR[0]	Description
MTU1 MTU2	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture
	0	1	0	TCNT cleared by TGRB compare match/input capture
	0	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/synchronous operation*1

Note 1. Synchronous operation is selected by setting the TSYRA.SYNC and TSYRB.SYNC bits to 1.
Note 2. Bit 7 is reserved in MTU1 and MTU2. It is read as 0. The write value is ignored.

17.2.2 TCRx : Timer Control Register x (x = U, V, W)

Base address: MTU5 = 0x9000_1C00

Offset address: 0x084 (TCRU)
0x094 (TCRV)
0x0A4 (TCRW)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TPSC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	TPSC[1:0]	Time Prescaler Select See Table 17.12.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

TPSC[1:0] bits (Time Prescaler Select)

The TPSC[1:0] bits select the TCNT counter clock. For details, see Table 17.12.

17.2.3 TCR2 — Timer Control Registers

17.2.3.1 MTU0.TCR2 : Timer Control Register 2

Base address: MTU0 = 0x9000_1300

Offset address: 0x28

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TPSC2[2:0]	Time Prescaler Select See Table 17.7.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The TCR2 register controls the TCNT operation for each channel in combination with the TCR register. TCR2 values should be specified only while TCNT operation is stopped.

TPSC2[2:0] bits (Time Prescaler Select)

The TPSC2[2:0] bits select the TCNT count clock source. The count clock source can be selected independently for each channel. For details, see Table 17.7.

Table 17.7 TCR2.TPSC2[2:0], TCR.TPSC[2:0] (MTU0)

Channel	TCR2 register			TCR register			Description
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	
MTU0	0	0	0	0	0	0	Internal clock: counts on PCLKH/1
	0	0	0	0	0	1	Internal clock: counts on PCLKH/4
	0	0	0	0	1	0	Internal clock: counts on PCLKH/16
	0	0	0	0	1	1	Internal clock: counts on PCLKH/64
	0	0	0	1	0	0	External clock: counts on MTCLKA pin input
	0	0	0	1	0	1	External clock: counts on MTCLKB pin input
	0	0	0	1	1	0	External clock: counts on MTCLKC pin input
	0	0	0	1	1	1	External clock: counts on MTCLKD pin input
	0	0	1	x	x	x	Internal clock: counts on PCLKH/2
	0	1	0	x	x	x	Internal clock: counts on PCLKH/8
	0	1	1	x	x	x	Internal clock: counts on PCLKH/32
	1	0	0	x	x	x	Internal clock: counts on PCLKH/256
	1	0	1	x	x	x	Internal clock: counts on PCLKH/1024
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	External clock: counts on MTIOC1A pin input

17.2.3.2 MTU1.TCR2 : Timer Control Register 2

Base address: MTU1 = 0x9000_1380

Offset address: 0x14

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	PCB[1:0]		TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TPSC2[2:0]	Time Prescaler Select See Table 17.8.	R/W
4:3	PCB[1:0]	Functional Expansion Control for Phase Counting Modes 2, 3, and 5 Controls expansion of functionality to modes 2, 3, and 5 of phase-counting mode	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The TCR2 register controls the TCNT operation for each channel in combination with the TCR register. TCR2 values should be specified only while TCNT operation is stopped.

TPSC2[2:0] bits (Time Prescaler Select)

The TPSC2[2:0] bits select the TCNT count clock source. The count clock source can be selected independently for each channel. For details, see Table 17.8.

PCB[1:0] bits (Functional Expansion Control for Phase Counting Modes 2, 3, and 5)

The PCB[1:0] bits control extended functions for phase counting modes 2, 3, and 5 in MTU1 and MTU2. For details, see section 17.3.6. Phase Counting Mode.

Table 17.8 TCR2.TPSC2[2:0], TCR.TPSC[2:0] (MTU1)

Channel	TCR2 register			TCR register			Description
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	
MTU1	0	0	0	0	0	0	Internal clock: counts on PCLKH/1
	0	0	0	0	0	1	Internal clock: counts on PCLKH/4
	0	0	0	0	1	0	Internal clock: counts on PCLKH/16
	0	0	0	0	1	1	Internal clock: counts on PCLKH/64
	0	0	0	1	0	0	External clock: counts on MTCLKA pin input
	0	0	0	1	0	1	External clock: counts on MTCLKB pin input
	0	0	0	1	1	0	Internal clock: counts on PCLKH/256
	0	0	0	1	1	1	Overflow/underflow of MTU2.TCNT
	0	0	1	x	x	x	Internal clock: counts on PCLKH/2
	0	1	0	x	x	x	Internal clock: counts on PCLKH/8
	0	1	1	x	x	x	Internal clock: counts on PCLKH/32
	1	0	0	x	x	x	Internal clock: counts on PCLKH/1024
	1	0	1	x	x	x	Setting prohibited
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	Setting prohibited

Note: This setting has no effect when MTU1 is in phase counting mode.

17.2.3.3 MTU2.TCR2 : Timer Control Register 2

Base address: MTU2 = 0x9000_1400

Offset address: 0x0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	PCB[1:0]		TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TPSC2[2:0]	Time Prescaler Select See Table 17.9.	R/W
4:3	PCB[1:0]	Functional Expansion Control for Phase Counting Modes 2, 3, and 5 Controls expansion of functionality to modes 2, 3, and 5 of phase-counting mode.	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The TCR2 register controls the TCNT operation for each channel in combination with the TCR register. TCR2 values should be specified only while TCNT operation is stopped.

TPSC2[2:0] bits (Time Prescaler Select)

The TPSC2[2:0] bits select the TCNT count clock source. The count clock source can be selected independently for each channel. See Table 17.9 for details.

PCB[1:0] bits (Functional Expansion Control for Phase Counting Modes 2, 3, and 5)

The PCB[1:0] bits control extended functions for phase counting modes 2, 3, and 5 in MTU1 and MTU2. See section 17.3.6. Phase Counting Mode for details.

Table 17.9 TCR2.TPSC2[2:0], TCR.TPSC[2:0] (MTU2)

Channel	TCR2 register			TCR register			Description
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	
MTU2	0	0	0	0	0	0	Internal clock: counts on PCLKH/1
	0	0	0	0	0	1	Internal clock: counts on PCLKH/4
	0	0	0	0	1	0	Internal clock: counts on PCLKH/16
	0	0	0	0	1	1	Internal clock: counts on PCLKH/64
	0	0	0	1	0	0	External clock: counts on MTCLKA pin input
	0	0	0	1	0	1	External clock: counts on MTCLKB pin input
	0	0	0	1	1	0	External clock: counts on MTCLKC pin input
	0	0	0	1	1	1	Internal clock: counts on PCLKH/1024
	0	0	1	x	x	x	Internal clock: counts on PCLKH/2
	0	1	0	x	x	x	Internal clock: counts on PCLKH/8
	0	1	1	x	x	x	Internal clock: counts on PCLKH/32
	1	0	0	x	x	x	Internal clock: counts on PCLKH/256
	1	0	1	x	x	x	Setting prohibited
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	Setting prohibited

Note: This setting has no effect when the MTU2 is in phase counting mode.

17.2.3.4 MTUm.TCR2 : Timer Control Register 2 (m = 3, 4, 6, 7)

Base address: MTU3 = 0x9000_1100, MTU4 = 0x9000_1200, MTU6 = 0x9000_1900, MTU7 = 0x9000_1A00

Offset address: 0x4D (MTU4, MTU7)
0x14C (MTU3, MTU6)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TPSC2[2:0]	Time Prescaler Select See Table 17.10.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The TCR2 register controls the TCNT operation for each channel in combination with the TCR register. TCR2 values should be specified only while TCNT operation is stopped.

TPSC2[2:0] bits (Time Prescaler Select)

The TPSC2[2:0] bits select the TCNT count clock source. The count clock source can be selected independently for each channel. See Table 17.10 for details.

Table 17.10 TCR2.TPSC2[2:0], TCR.TPSC[2:0] (MTU3, MTU4, MTU6, MTU7)

Channel	TCR2 register			TCR register			Description
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	
MTU3 MTU4 MTU6 MTU7	0	0	0	0	0	0	Internal clock: counts on PCLKH/1
	0	0	0	0	0	1	Internal clock: counts on PCLKH/4
	0	0	0	0	1	0	Internal clock: counts on PCLKH/16
	0	0	0	0	1	1	Internal clock: counts on PCLKH/64
	0	0	0	1	0	0	Internal clock: counts on PCLKH/256
	0	0	0	1	0	1	Internal clock: counts on PCLKH/1024
	0	0	0	1	1	0	External clock: counts on MTCLKA pin input
	0	0	0	1	1	1	External clock: counts on MTCLKB pin input
	0	0	1	x	x	x	Internal clock: counts on PCLKH/2
	0	1	0	x	x	x	Internal clock: counts on PCLKH/8
	0	1	1	x	x	x	Internal clock: counts on PCLKH/32
	1	0	0	x	x	x	Setting prohibited
	1	0	1	x	x	x	Setting prohibited
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	Setting prohibited

17.2.3.5 MTU8.TCR2 : Timer Control Register 2

Base address: MTU8 = 0x9000_1600

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TPSC2[2:0]	Time Prescaler Select See Table 17.10.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The TCR2 register controls the TCNT operation for each channel in combination with the TCR register. TCR2 values should be specified only while TCNT operation is stopped.

TPSC2[2:0] bits (Time Prescaler Select)

The TPSC2[2:0] bits select the TCNT count clock source. The count clock source can be selected independently for each channel. See Table 17.10 for details.

Table 17.11 TCR2.TPSC2[2:0], TCR.TPSC[2:0] (MTU8)

Channel	TCR2 register			TCR register			Description
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	
MTU8	0	0	0	0	0	0	Internal clock: counts on PCLKH/1
	0	0	0	0	0	1	Internal clock: counts on PCLKH/4
	0	0	0	0	1	0	Internal clock: counts on PCLKH/16
	0	0	0	0	1	1	Internal clock: counts on PCLKH/64
	0	0	0	1	0	0	Internal clock: counts on PCLKH/256
	0	0	0	1	0	1	Internal clock: counts on PCLKH/1024
	0	0	0	1	1	0	External clock: counts on MTCLKA pin input
	0	0	0	1	1	1	External clock: counts on MTCLKB pin input
	0	0	1	x	x	x	Internal clock: counts on PCLKH/2
	0	1	0	x	x	x	Internal clock: counts on PCLKH/8
	0	1	1	x	x	x	Internal clock: counts on PCLKH/32
	1	0	0	x	x	x	Setting prohibited
	1	0	1	x	x	x	Setting prohibited
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	Setting prohibited

17.2.3.6 MTU5.TCR2x : Timer Control Register 2x (x = U, V, W)

Base address: MTU5 = 0x9000_1C00

Offset address: 0x85 (TCR2U)
0x95 (TCR2V)
0xA5 (TCR2W)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	CKEG[1:0]		TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TPSC2[2:0]	Time Prescaler Select See Table 17.12.	R/W
4:3	CKEG[1:0]	Clock Edge Select 0 0: Counts at the rising edge 0 1: Counts at the falling edge 1 x: Counts at both edges	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

TPSC2[2:0] bits (Time Prescaler Select)

The TPSC2[2:0] bits select the TCNT count clock source. See Table 17.12 for details.

CKEG[1:0] bits (Clock Edge Select)

The CKEG[1:0] bits select the edge of the count clock signal input from the MTIOC1A pin.

Table 17.12 TPSC[1:0], CKEG[1:0], TPSC2[2:0] (MTU5)

Channel	TCR2 register					TCR register		Description
	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Bit 1	Bit 0	
	CKEG[1]	CKEG[0]	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[1]	TPSC[0]	
MTU5	x	x	0	0	0	0	0	Internal clock: counts on PCLKH/1
	x	x	0	0	0	0	1	Internal clock: counts on PCLKH/4
	x	x	0	0	0	1	0	Internal clock: counts on PCLKH/16
	x	x	0	0	0	1	1	Internal clock: counts on PCLKH/64
	x	x	0	0	1	x	x	Internal clock: counts on PCLKH/2
	x	x	0	1	0	x	x	Internal clock: counts on PCLKH/8
	x	x	0	1	1	x	x	Internal clock: counts on PCLKH/32
	x	x	1	0	0	x	x	Internal clock: counts on PCLKH/256
	x	x	1	0	1	x	x	Internal clock: counts on PCLKH/1024
	x	x	1	1	0	x	x	Setting prohibited
	0	0	1	1	1	x	x	Counts at the rising edge on MTIOC1A pin input
	0	1	1	1	1	x	x	Counts at the falling edge on MTIOC1A pin input
	1	x	1	1	1	x	x	Counts at the both edges on MTIOC1A pin input

Note: Bits 7 to 2 in the TCR register are reserved for MTU5. These bits are read as 0. The write value should be 0.

17.2.4 TMDR1 : Timer Mode Register 1

Base address: MTU0 = 0x9000_1300

Offset address: 0x001

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	BFE	BFB	BFA	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Base address: MTU1 = 0x9000_1380, MTU2 = 0x9000_1400

Offset address: 0x001

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Base address: MTU3 = 0x9000_1100, MTU4 = 0x9000_1200, MTU6 = 0x9000_1900, MTU7 = 0x9000_1A00
MTU8 = 0x9000_1600

Offset address: 0x102 (MTU3, MTU6), 0x003 (MTU4, MTU7), 0x001 (MTU8)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	BFB	BFA	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	MD[3:0]	Mode Select These bits specify the timer operating mode. See Table 17.13 for details.	R/W

Bit	Symbol	Function	R/W
4	BFA	Buffer Operation A 0: TGRA and TGRC operate normally 1: TGRA and TGRC used together for buffer operation	R/W
5	BFB	Buffer Operation B 0: TGRB and TGRD operate normally 1: TGRB and TGRD used together for buffer operation	R/W
6	BFE	Buffer Operation E 0: MTU0.TGRE and MTU0.TGRF operate normally 1: MTU0.TGRE and MTU0.TGRF used together for buffer operation	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

The TMDR1 register specifies the operating mode of each channel. The MTU has a total of eight TMDR1 registers, one each for MTU0 to MTU4, MTU6, MTU7, and MTU8. TMDR1 register values should be specified only while TCNT operation is stopped.

MD[3:0] bits (Mode Select)

Table 17.13 Operating mode setting by MD[3:0] bits (MTU0 to MTU4 and MTU6 to MTU8)

MD[3]	MD[2]	MD[1]	MD[0]	Description	MTU0	MTU1	MTU2	MTU3	MTU4	MTU6	MTU7	MTU8
0	0	0	0	Normal mode	✓	✓	✓	✓	✓	✓	✓	✓
0	0	0	1	Setting prohibited								
0	0	1	0	PWM mode 1	✓	✓	✓	✓	✓	✓	✓	
0	0	1	1	PWM mode 2	✓	✓	✓					
0	1	0	0	Phase counting mode 1		✓	✓					
0	1	0	1	Phase counting mode 2		✓	✓					
0	1	1	0	Phase counting mode 3		✓	✓					
0	1	1	1	Phase counting mode 4		✓	✓					
1	0	0	0	Reset-synchronized PWM mode*1				✓		✓		
1	0	0	1	Phase counting mode 5		✓	✓					
1	0	1	0	Setting prohibited								
1	0	1	1	Setting prohibited								
1	1	0	0	Setting prohibited								
1	1	0	1	Complementary PWM mode 1 (transfer at crest)*1				✓		✓		
1	1	1	0	Complementary PWM mode 2 (transfer at trough)*1				✓		✓		
1	1	1	1	Complementary PWM mode 3 (transfer at crest and trough)*1				✓		✓		

Note: Only set the corresponding operating mode listed above for each channel.
Note 1. Reset-synchronized PWM mode and complementary PWM mode can only be set for MTU3 and MTU6. When MTU3 or MTU6 is set to reset-synchronized PWM mode or complementary PWM mode, the MTU4 or MTU7 settings become ineffective and automatically conform to the MTU3 or MTU6 setting, respectively. MTU4 and MTU7 should be set to the initial values (normal mode).

BFA bit (Buffer Operation A)

The BFA bit specifies whether to operate TGRA in the normal way or to use TGRA and TGRC together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare does not take place in a mode other than complementary PWM mode, but TGRC compare match takes place in complementary PWM mode. If MTU4 compare match takes place in the Tb interval of complementary PWM mode, clear the TGIEC bit in the timer interrupt enable register (MTU4.TIER) to 0.

In reset synchronized PWM mode, buffer operation in MTU3 and MTU4 (MTU6 and MTU7) depends on the settings in the BFA bit of MTU3.TMDR1. The BFA bit of MTU4.TMDR1 should be set to 0. In complementary PWM mode, buffer

operation in MTU3 and MTU4 (or MTU6 and MTU7) depends on the BFA bit setting in MTU3.TMDR1 (MTU6.TMDR1). Set the BFA bit in MTU4.TMDR1 (MTU7.TMDR1) to 0.

In MTU1 and MTU2, which have no TGRC, this bit is reserved. It is read as 0. The write value should be 0. For details about the Tb interval of complementary PWM mode, see Figure 17.50.

BFB bit (Buffer Operation B)

The BFB bit specifies whether to operate TGRB in the normal way or to use TGRB and TGRD together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare does not usually take place, but TGRD compare match takes place in complementary PWM mode. If MTU4 compare match takes place in the Tb interval of complementary PWM mode, clear the TGIED bit in the timer interrupt enable register (MTU4.TIER) to 0.

In reset synchronized PWM mode, buffer operation in MTU3 and MTU4 (MTU6 and MTU7) depends on the settings in the BFB bit of MTU3.TMDR1. The BFB bit of MTU4.TMDR1 should be set to 0. In complementary PWM mode, buffer operation in MTU3 and MTU4 (or MTU6 and MTU7) depends on the BFB bit setting in MTU3.TMDR1 (MTU6.TMDR1). Set the BFB bit in MTU4.TMDR1 (MTU7.TMDR1) to 0.

In MTU1 and MTU2, which have no TGRD, this bit is reserved. It is read as 0. The write value should be 0. For details about the Tb interval of complementary PWM mode, see Figure 17.50.

BFE bit (Buffer Operation E)

The BFE bit specifies whether to operate MTU0.TGRE and MTU0.TGRF in the normal way or to use them together for buffer operation. TGRF compare match takes place even when TGRF is used as a buffer register.

In MTU1 to MTU4, MTU6, MTU7, and MTU8, this bit is reserved. It is read as 0. The write value should be 0.

17.2.5 TMDR2x : Timer Mode Register 2x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x270 (TMDR2A)
0xA70 (TMDR2B)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	DRS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DRS	Double Buffer Select 0: Double buffer function is disabled 1: Double buffer function is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

TMDR2A and TMDR2B specify the double buffer function in complementary PWM mode 3 (transfer at the crest and trough of the counter value). The MTU has two TMDR2 registers, and one each for MTU3 (TMDR2A) and MTU6 (TMDR2B). TMDR2A and TMDR2B values should be specified only while TCNT operation is stopped.

DRS bit (Double Buffer Select)

This bit enables or disables the double buffer function in complementary PWM mode.

17.2.6 TMDR3 : Timer Mode Register 3

Base address: MTU1 = 0x9000_1380

Offset address: 0x011

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PHCK SEL	LWA
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
0	LWA	MTU1/MTU2 Combination Longword Access Control 0: 16-bit access is enabled. 1: 32-bit access is enabled.	R/W
1	PHCKSEL	External Input Phase Clock Select Selects the external clock pin for phase counting mode. 0: MTCLKA and MTCLKB are selected for the external phase clock. 1: MTCLKC and MTCLKD are selected for the external phase clock.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

The TMDR3 register controls access in longword units to the combination of MTU1 and MTU2, and there is only one register, which is in MTU1. Such access can proceed when the LWA bit is set in the ways listed in Table 17.14.

LWA bit (MTU1/MTU2 Combination Longword Access Control)

This bit selects a 32-bit access in combination of MTU1 and MTU2. Set this bit only in 32-bit phase counting mode.

When LWA is set to 0, the MTU1.TCNTLW, MTU1.TGRALW, and MTU1.TGRBLW registers cannot be accessed. These bits are read as 0x00000000.

When LWA is set to 1, the MTU1.TCNT, MTU2.TCNT, MTU1.TGRA, MTU2.TGRA, MTU1.TGRB, and MTU2.TGRB registers cannot be accessed. These bits are read as 0x0000.

The settings of the control registers (TCR, TCR2, TIOR, and TMDR1) in MTU1 take priority because MTU1 and MTU2 operates together while LWA = 1.

In this case, the settings of the control registers in MTU2 are ignored. Furthermore, MTU2 input capture and compare match are disabled, which in turn disables any linked operation with the ELC.

The cascaded connection of MTU1 and MTU2 with the LWA bit set to 1 can only be used in phase counting mode, but not in normal mode, PWM mode 1, or PWM mode 2. Select phase counting mode when LWA = 1.

Initialize the counters and general registers in MTU1 and MTU2 prior to rewriting to the LWA bit.

PHCKSEL bit (External Input Phase Clock Select)

When the MTU1 and MTU2 registers are combined for 32-bit phase counting mode or MTU2 phase counting mode, this bit selects the external input clock pin on which A- or B-phase signal is to be input. See Table 17.61 for details.

Table 17.14 Setting and combination of the TMDR3 register

Register	TMDR3.LWA = 0		TMDR3.LWA = 1	
	Symbol	Access mode	Symbol	Access mode
Counter in MTU1*1	MTU1.TCNT	Word	MTU1.TCNTLW	Longword
Counter in MTU2	MTU2.TCNT	Word		
General register A in MTU1	MTU1.TGRA	Word	MTU1.TGRALW	Longword
General register A in MTU2	MTU2.TGRA	Word		
General register B in MTU1	MTU1.TGRB	Word	MTU1.TGRBLW	Longword
General register B in MTU2	MTU2.TGRB	Word		

Note 1. When the LWA bit is set to 1, setting the count clock for MTU1 as MTU2.TCNT overflow/underflow is not required.

17.2.7 TIOR — Timer I/O Control Registers

17.2.7.1 MTUm.TIOR : Timer I/O Control Register (m = 1, 2)

Base address: MTU1 = 0x9000_1380
MTU2 = 0x9000_1400

Offset address: 0x002

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOB[3:0]				IOA[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOA[3:0]	I/O Control A*1 For MTU1, see Table 17.15. For MTU2, see Table 17.17.	R/W
7:4	IOB[3:0]	I/O Control B*1 For MTU1, see Table 17.16. For MTU2, see Table 17.18.	R/W

Note 1. When the value of IOn[3:0] (n = A, B) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIOR register controls the TGR register. The TIOR register should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIOR is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIOR is affected by the TMDR1 setting.

Table 17.15 IOA[3:0] bits enumerated value and function for MTU1

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA register function	MTIOC1A pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Input capture at generation of MTU0.TGRA compare match/ input capture.

Table 17.16 IOB[3:0] bits enumerated value and function for MTU1

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB register function	MTIOC1B pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	0	0		Input capture at occurrence of compare match or input capture in the MTU0.TGRC register
1	1	1	x		Input capture at occurrence of compare match or input capture in the MTU8.TGRC register

Table 17.17 IOA[3:0] bits enumerated value and function for MTU2

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA register function	MTIOC2A pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Table 17.18 IOB[3:0] bits enumerated value and function for MTU2 (1 of 2)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB register function	MTIOC2B pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.

Table 17.18 IOB[3:0] bits enumerated value and function for MTU2 (2 of 2)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB register function	MTIOC2B pin function
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

17.2.7.2 MTU0.TIORH : Timer I/O Control Register H

Base address: MTU0 = 0x9000_1300

Offset address: 0x002

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOB[3:0]				IOA[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOA[3:0]	I/O Control A*1 See Table 17.19.	R/W
7:4	IOB[3:0]	I/O Control B*1 See Table 17.20.	R/W

Note 1. When the value of IOn[3:0] (n = A, B) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORH register controls the TGR register. The TIORH register should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORH is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORH is affected by the TMDR1 setting.

Table 17.19 IOA[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA register function	MTIOC0A pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	0	0		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).*1
1	1	1	x		Input capture on generation of compare match with MTU8.TGRC

Note 1. When PCLKH/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKH/1 as the count clock for MTU1.

Table 17.20 IOB[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB register function	MTIOC0B pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).*1

Note 1. When PCLKH/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKH/1 as the count clock for MTU1.

17.2.7.3 MTU3.TIORH : Timer I/O Control Register H

Base address: MTU3 = 0x9000_1100

Offset address: 0x104

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOB[3:0]				IOA[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOA[3:0]	I/O Control A*1 See Table 17.21.	R/W
7:4	IOB[3:0]	I/O Control B*1 See Table 17.22.	R/W

Note 1. When the value of IOn[3:0] (n = A, B) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORH register controls the TGR register. The TIORH register should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORH is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORH is affected by the TMDR1 setting.

Table 17.21 IOA[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA register function	MTIOC3A pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Table 17.22 IOB[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB register function	MTIOC3B pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

17.2.7.4 MTU4.TIORH : Timer I/O Control Register H

Base address: MTU4 = 0x9000_1200

Offset address: 0x006

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOB[3:0]				IOA[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOA[3:0]	I/O Control A*1 See Table 17.23.	R/W
7:4	IOB[3:0]	I/O Control B*1 See Table 17.24.	R/W

Note 1. When the value of IOn[3:0] (n = A, B) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORH register controls the TGR register. The TIORH register should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORH is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORH is affected by the TMDR1 setting.

Table 17.23 IOA[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA register function	MTIOC4A pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

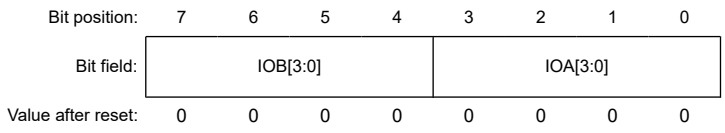
Table 17.24 IOB[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB register function	MTIOC4B pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

17.2.7.5 MTU6.TIORH : Timer I/O Control Register H

Base address: MTU6 = 0x9000_1900

Offset address: 0x104



Bit	Symbol	Function	R/W
3:0	IOA[3:0]	I/O Control A*1 See Table 17.25.	R/W
7:4	IOB[3:0]	I/O Control B*1 See Table 17.26.	R/W

Note 1. When the value of IOn[3:0] (n = A, B) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORH register controls the TGR register. The TIORH register should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORH is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORH is affected by the TMDR1 setting.

Table 17.25 IOA[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA register function	MTIOC6A pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Table 17.26 IOB[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB register function	MTIOC6B pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

17.2.7.6 MTU7.TIORH : Timer I/O Control Register H

Base address: MTU7 = 0x9000_1A00

Offset address: 0x006

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOB[3:0]				IOA[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOA[3:0]	I/O Control A*1 See Table 17.27.	R/W
7:4	IOB[3:0]	I/O Control B*1 See Table 17.28.	R/W

Note 1. When the value of IOn[3:0] (n = A, B) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORH register controls the TGR register. The TIORH register should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORH is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORH is affected by the TMDR1 setting.

Table 17.27 IOA[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA register function	MTIOC7A pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Table 17.28 IOB[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB register function	MTIOC7B pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

17.2.7.7 MTU8.TIORH : Timer I/O Control Register H

Base address: MTU8 = 0x9000_1600

Offset address: 0x002

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOB[3:0]				IOA[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOA[3:0]	I/O Control A*1 See Table 17.29.	R/W
7:4	IOB[3:0]	I/O Control B*1 See Table 17.30.	R/W

Note 1. When the value of IOn[3:0] (n = A, B) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORH register controls the TGR register. The TIORH register should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORH is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORH is affected by the TMDR1 setting.

Table 17.29 IOA[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA register function	MTIOC8A pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Table 17.30 IOB[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB register function	MTIOC8B pin function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).

17.2.7.8 MTU0.TIORL : Timer I/O Control Register L

Base address: MTU0 = 0x9000_1300

Offset address: 0x003

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOD[3:0]				IOC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOC[3:0]	I/O Control A*1 See Table 17.31.	R/W
7:4	IOD[3:0]	I/O Control B*1 See Table 17.32.	R/W

Note 1. When the value of IOn[3:0] (n = C, D) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORL register controls the TGR register and should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORL is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORL is affected by the TMDR1 setting.

Table 17.31 IOC[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC register function	MTIOC0C pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register*1	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).*2

Note 1. When the MTU0.TMDR1.BFA bit is set to 1 and MTU0.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.
Note 2. When PCLKH/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKH/1 as the count clock for MTU1.

Table 17.32 IOD[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD register function	MTIOC0D pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	0	0	0	Input capture register*1	Input capture at rising edge.
1	0	0	1		Input capture at falling edge.
1	0	1	x		Input capture at both edges.
1	1	x	x		Capture input source is the clock source for counting in MTU1. Input capture on counting up or down by MTU1.TCNT (LWA = 0) or MTU1.TCNTLW (LWA = 1).*2

Note 1. When the MTU0.TMDR1.BFB bit is set to 1 and MTU0.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.
Note 2. When PCLKH/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKH/1 as the count clock for MTU1.

17.2.7.9 MTU3.TIORL : Timer I/O Control Register L

Base address: MTU3 = 0x9000_1100

Offset address: 0x105

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOD[3:0]				IOC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOC[3:0]	I/O Control A*1 See Table 17.33.	R/W
7:4	IOD[3:0]	I/O Control B*1 See Table 17.34.	R/W

Note 1. When the value of IOn[3:0] (n = C, D) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORL register controls the TGR register and should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORL is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORL is affected by the TMDR1 setting.

Table 17.33 IOC[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC register function	MTIOC3C pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Note 1. When the MTU3.TMDR1.BFA bit is set to 1 and MTU3.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 17.34 IOD[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD register function	MTIOC3D pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Note 1. When the MTU3.TMDR1.BFB bit is set to 1 and MTU3.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

17.2.7.10 MTU4.TIORL : Timer I/O Control Register L

Base address: MTU4 = 0x9000_1200

Offset address: 0x007

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOD[3:0]				IOC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOC[3:0]	I/O Control A*1 See Table 17.35.	R/W
7:4	IOD[3:0]	I/O Control B*1 See Table 17.36.	R/W

Note 1. When the value of IOn[3:0] (n = C, D) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORL register controls the TGR register and should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORL is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORL is affected by the TMDR1 setting.

Table 17.35 IOC[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC register function	MTIOC4C pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Note 1. When the MTU4.TMDR1.BFB bit is set to 1 and MTU4.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 17.36 IOD[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD register function	MTIOC4D pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Note 1. When the MTU4.TMDR1.BFB bit is set to 1 and MTU4.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

17.2.7.11 MTU6.TIORL : Timer I/O Control Register L

Base address: MTU6 = 0x9000_1900

Offset address: 0x105

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOD[3:0]				IOC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOC[3:0]	I/O Control A*1 See Table 17.37.	R/W

Bit	Symbol	Function	R/W
7:4	IOD[3:0]	I/O Control B*1 See Table 17.38.	R/W

Note 1. When the value of IOn[3:0] (n = C, D) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORL register controls the TGR register and should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORL is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORL is affected by the TMDR1 setting.

Table 17.37 IOC[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC register function	MTIOC6C pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Note 1. When the MTU6.TMDR1.BFA bit is set to 1 and MTU6.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 17.38 IOD[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD register function	MTIOC6D pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Note 1. When the MTU6.TMDR1.BFB bit is set to 1 and MTU6.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

17.2.7.12 MTU7.TIORL : Timer I/O Control Register L

Base address: MTU7 = 0x9000_1A00

Offset address: 0x007

Bit position:	7	6	5	4	3	2	1	0
Bit field:	IOD[3:0]				IOC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	IOC[3:0]	I/O Control A*1 See Table 17.39.	R/W
7:4	IOD[3:0]	I/O Control B*1 See Table 17.40.	R/W

Note 1. When the value of IOn[3:0] (n = C, D) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORL register controls the TGR register and should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORL is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORL is affected by the TMDR1 setting.

Table 17.39 IOC[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC register function	MTIOC7C pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Note 1. When the MTU7.TMDR1.BFA bit is set to 1 and MTU7.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 17.40 IOD[3:0] bits enumerated value and function

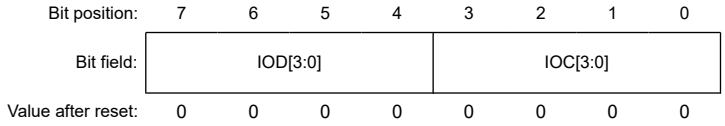
Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD register function	MTIOC7D pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Note 1. When the MTU7.TMDR1.BFB bit is set to 1 and MTU7.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

17.2.7.13 MTU8.TIORL : Timer I/O Control Register L

Base address: MTU8 = 0x9000_1600

Offset address: 0x003



Bit	Symbol	Function	R/W
3:0	IOC[3:0]	I/O Control A*1 See Table 17.41.	R/W
7:4	IOD[3:0]	I/O Control B*1 See Table 17.42.	R/W

Note 1. When the value of IOn[3:0] (n = C, D) is changed to the output-prohibited state (0000b or 0100b) during low, high, or toggle output at compare match, this register is in Hi-Z.

The TIORL register controls the TGR register and should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORL is valid when the counter is stopped (the CSTn bit in TSTRA and the CSTn bit in TSTRB are set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORL is affected by the TMDR1 setting.

Table 17.41 IOC[3:0] bits enumerated value and function

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRC register function	MTIOC8C pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Note 1. When the MTU8.TMDR1.BFA bit is set to 1 and MTU8.TGRC register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 17.42 IOD[3:0] bits enumerated value and function

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD[3]	IOD[2]	IOD[1]	IOD[0]	TGRD register function	MTIOC8D pin function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is low. Low output at compare match.
0	0	1	0		Initial output is low. High output at compare match.
0	0	1	1		Initial output is low. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is high. Low output at compare match.
0	1	1	0		Initial output is high. High output at compare match.
0	1	1	1		Initial output is high. Toggle output at compare match.
1	x	0	0	Input capture register*1	Input capture at rising edge.
1	x	0	1		Input capture at falling edge.
1	x	1	x		Input capture at both edges.

Note 1. When the MTU8.TMDR1.BFB bit is set to 1 and MTU8.TGRD register is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

17.2.7.14 MTU5.TIORx : Timer I/O Control Register x (x = U, V, W)

Base address: MTU5 = 0x9000_1C00

Offset address: 0x086 (TIORU)
0x096 (TIORV)
0x0A6 (TIORW)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	IOC[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	IOC[4:0]	I/O Control C For details, see Table 17.43.	R/W

Bit	Symbol	Function	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The TIORx register controls the TGR register and should be set when the TMDR1 register setting is normal mode, PWM mode, or phase counting mode. The initial output specified by TIORx is valid when the counter is stopped (the CST5 bit in TSTR is set to 0). In PWM mode 2, the output when the counter is cleared to 0 is specified.

Note: TIORx is affected by the TMDR1 setting.

Table 17.43 IOC[4:0] bits enumerated value and function for MTU5

Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC[4]	IOC[3]	IOC[2]	IOC[1]	IOC[0]	TGRU, TGRV, TGRW register function	MTIC5U, MTIC5V, MTIC5W pin function
0	0	0	0	0	Output compare register	No function
0	0	0	0	1		Setting prohibited
0	0	0	1	x		Setting prohibited
0	0	1	x	x		Setting prohibited
0	1	x	x	x		Setting prohibited
1	0	0	0	0	Input capture register	Setting prohibited
1	0	0	0	1		Input capture at rising edge.
1	0	0	1	0		Input capture at falling edge.
1	0	0	1	1		Input capture at both edges.
1	0	1	x	x		Input capture on generation of compare match with MTU8.TGRC
1	1	0	0	0		Setting prohibited
1	1	0	0	1		Measurement of low pulse width of external input signal. Capture at trough in complementary PWM mode.
1	1	0	1	0		Measurement of low pulse width of external input signal. Capture at crest of complementary PWM mode.
1	1	0	1	1		Measurement of low pulse width of external input signal. Capture at crest and trough of complementary PWM mode.
1	1	1	0	0		Setting prohibited
1	1	1	0	1		Measurement of high pulse width of external input signal. Capture at trough in complementary PWM mode.
1	1	1	1	0		Measurement of high pulse width of external input signal. Capture at crest of complementary PWM mode.
1	1	1	1	1		Measurement of high pulse width of external input signal. Capture at crest and trough of complementary PWM mode.

17.2.8 TCNTCMPCLR : Timer Compare Match Clear Register

Base address: MTU5 = 0x9000_1C00

Offset address: 0x0B6

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	CMPCLR5U	CMPCLR5V	CMPCLR5W
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMPCLR5W	TCNT Compare Clear 5W 0: Disables MTU5.TCNTW to be cleared to 0x0000 at MTU5.TCNTW and MTU5.TGRW compare match or input capture 1: Enables MTU5.TCNTW to be cleared to 0x0000 at MTU5.TCNTW and MTU5.TGRW compare match or input capture	R/W
1	CMPCLR5V	TCNT Compare Clear 5V 0: Disables MTU5.TCNTV to be cleared to 0x0000 at MTU5.TCNTV and MTU5.TGRV compare match or input capture 1: Enables MTU5.TCNTV to be cleared to 0x0000 at MTU5.TCNTV and MTU5.TGRV compare match or input capture	R/W
2	CMPCLR5U	TCNT Compare Clear 5U 0: Disables MTU5.TCNTU to be cleared to 0x0000 at MTU5.TCNTU and MTU5.TGRU compare match or input capture 1: Enables MTU5.TCNTU to be cleared to 0x0000 at MTU5.TCNTU and MTU5.TGRU compare match or input capture	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

TCNTCMPCLR specifies requests to clear MTU5.TCNTU, MTU5.TCNTV, and MTU5.TCNTW. The MTU has one TCNTCMPCLR (on MTU5).

17.2.9 TIER — Timer Interrupt Enable Registers

17.2.9.1 MTUm.TIER : Timer Interrupt Enable Register (m = 0 to 4, 6 to 8)

Base address: MTU1 = 0x9000_1380, MTU2 = 0x9000_1400

Offset address: 0x004 (MTU1, MTU2)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

Base address: MTU0 = 0x9000_1300, MTU3 = 0x9000_1100, MTU6 = 0x9000_1900

Offset address: 0x004 (MTU0), 0x108 (MTU3, MTU6)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

Base address: MTU4 = 0x9000_1200, MTU7 = 0x9000_1A00

Offset address: 0x009 (MTU4, MTU7)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

Base address: MTU8 = 0x9000_1600

Offset address: 0x004 (MTU8)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TGIEA	TGR Interrupt Enable A 0: Interrupt requests (TGIA) disabled 1: Interrupt requests (TGIA) enabled	R/W
1	TGIEB	TGR Interrupt Enable B 0: Interrupt requests (TGIB) disabled 1: Interrupt requests (TGIB) enabled	R/W
2	TGIEC	TGR Interrupt Enable C 0: Interrupt requests (TGIC) disabled 1: Interrupt requests (TGIC) enabled	R/W
3	TGIED	TGR Interrupt Enable D 0: Interrupt requests (TGID) disabled 1: Interrupt requests (TGID) enabled	R/W
4	TCIEV	Overflow Interrupt Enable 0: Interrupt requests (TCIV) disabled 1: Interrupt requests (TCIV) enabled	R/W
5	TCIEU	Underflow Interrupt Enable 0: Interrupt requests (TCIU) disabled 1: Interrupt requests (TCIU) enabled	R/W
6	TTGE2	A/D Converter Start Request Enable 2 0: A/D converter start request generation by MTUn.TCNT underflow (trough) disabled 1: A/D converter start request generation by MTUn.TCNT underflow (trough) enabled	R/W
7	TTGE	A/D Converter Start Request Enable 0: A/D converter start request generation disabled 1: A/D converter start request generation enabled	R/W

Note: n = 4 or 7

The TIER register enables or disables interrupt requests from each channel. The MTU has a total of ten TIER registers, two for MTU0 and one each for MTU1 to MTU8.

TGIEA and TGIEB bits (TGR Interrupt Enable A and TGR Interrupt Enable B)

Each bit enables or disables interrupt requests (TGIn) (n = A, B).

TGIEC and TGIED bits (TGR Interrupt Enable C and TGR Interrupt Enable D)

Each bit enables or disables interrupt requests (TGIn) (n = C, D).

In MTU1 and MTU2, these bits are reserved. They are read as 0. The write value should be 0.

TCIEV bit (Overflow Interrupt Enable)

This bit enables or disables interrupt requests (TCIV).

TCIEU bit (Underflow Interrupt Enable)

This bit enables or disables interrupt requests (TCIU).

In MTU0, MTU3, MTU4, MTU6, MTU7, and MTU8, this bit is reserved. It is read as 0. The write value should be 0.

TTGE2 bit (A/D Converter Start Request Enable 2)

This bit enables or disables generation of A/D converter start requests by MTUn.TCNT underflow (trough) in complementary PWM mode (n = 4 or 7).

In MTU0 to MTU3, MTU6, and MTU8, this bit is reserved. It is read as 0. The write value should be 0.

TTGE bit (A/D Converter Start Request Enable)

This bit enables or disables generation of A/D converter start requests by TGRA input capture/compare match.

MTU8 is a reserved bit. It is read as 0. The write value should be 0.

17.2.9.2 MTU0.TIER2 : Timer Interrupt Enable Register 2

Base address: MTU0 = 0x9000_1300

Offset address: 0x024

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TTGE2	—	—	—	—	—	TGIEF	TGIEE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TGIEE	TGR Interrupt Enable E 0: Interrupt requests (TGIE) disabled 1: Interrupt requests (TGIE) enabled	R/W
1	TGIEF	TGR Interrupt Enable F 0: Interrupt requests (TGIF) disabled 1: Interrupt requests (TGIF) enabled	R/W
6:2	—	The read values are undefined. The write value should be 0.	R/W
7	TTGE2	A/D Converter Start Request Enable 2 0: A/D converter start request generation by compare match between MTU0.TCNT and MTU0.TGRE disabled 1: A/D converter start request generation by compare match between MTU0.TCNT and MTU0.TGRE enabled	R/W

TGIEE bit (TGR Interrupt Enable E)

The TGIEE bit enables or disables interrupt requests by compare match between MTU0.TCNT and MTU0.TGRE.

TGIEF bit (TGR Interrupt Enable F)

The TGIEF bit enables or disables interrupt requests by compare match between MTU0.TCNT and MTU0.TGRF.

TTGE2 bit (A/D Converter Start Request Enable 2)

The TTGE2 bit enables or disables A/D converter start requests by compare match between MTU0.TCNT and MTU0.TGRE.

17.2.9.3 MTU5.TIER : Timer Interrupt Enable Register

Base address: MTU5 = 0x9000_1C00

Offset address: 0x0B2

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	TGIE5W	TGR Interrupt Enable 5W 0: Interrupt requests TGIW5 disabled 1: Interrupt requests TGIW5 enabled	R/W
1	TGIE5V	TGR Interrupt Enable 5V 0: Interrupt requests TGIV5 disabled 1: Interrupt requests TGIV5 enabled	R/W
2	TGIE5U	TGR Interrupt Enable 5U 0: Interrupt requests TGIU5 disabled 1: Interrupt requests TGIU5 enabled	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

TGIE5x bits (TGR Interrupt Enable 5x)

Each bit enables or disables interrupt requests (TGIx5) (x = U, V, W).

17.2.10 TSR : Timer Status Register

Base address: MTU1 = 0x9000_1380, MTU2 = 0x9000_1400
MTU3 = 0x9000_1100, MTU4 = 0x9000_1200, MTU6 = 0x9000_1900, MTU7 = 0x9000_1A00

Offset address: 0x005 (MTU1, MTU2), 0x02D (MTU4, MTU7), 0x12C (MTU3, MTU6)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TGFA	Input Capture/Output Compare Flag A 0: TGRA input capture or compare match has not occurred 1: TGRA input capture or compare match has occurred	R/W*1
1	TGFB	Input Capture/Output Compare Flag B 0: TGRB input capture or compare match has not occurred 1: TGRB input capture or compare match has occurred	R/W*1
2	TGFC	Input Capture/Output Compare Flag C 0: TGRC input capture or compare match has not occurred 1: TGRC input capture or compare match has occurred	R/W*1
3	TGFD	Input Capture/Output Compare Flag D 0: TGRD input capture or compare match has not occurred 1: TGRD input capture or compare match has occurred	R/W*1
4	TCFV	Overflow flag 0: TCNT overflow has not occurred 1: TCNT overflow has occurred	R/W*1
5	TCFU	Underflow flag 0: TCNT underflow has not occurred when MTU1 and MTU2 are in phase counting mode 1: TCNT underflow has occurred when MTU1 and MTU2 are in phase counting mode	R/W*1
6	—	This bit is read as 1. The write value should be 1.	R/W
7	TCFD	Count Direction Flag 0: TCNT counts down 1: TCNT counts up	R

Note 1. Only writing 0 to this bit after reading 1 from this bit in order to clear the flag is possible. After 1 is read from this bit, if the next flag is set before 0 is written to this bit, writing 0 to this bit does not clear the flag. In this case, read 1 again from this bit, and then write 0 to this bit.

TSR indicates the states of each of the channels. The MTU has a total of six TSR registers, one each for MTU1 to MTU4, MTU6, and MTU7.

TGFA flag (Input Capture/Output Compare Flag A)

The TGFA flag indicates generation of TGRA input capture or compare match. Only writing 0 to this bit in order to clear the flag is possible.

[Clearing condition]

- When TGFA = 1, the TGFA flag is read, and then 0 is written to the TGFA flag.

[Setting conditions]

- TCNT = TGRA when the TGRA register functions as an output compare register.
- The value of TCNT is transferred to TGRA due to the input capture signal when the TGRA register functions as an input capture register.

TGFB flag (Input Capture/Output Compare Flag B)

The TGFB flag indicates generation of TGRB input capture or compare match. Only writing 0 to this bit in order to clear the flag is possible.

[Clearing condition]

- When TGFB = 1, the TGFB flag is read, and then 0 is written to the TGFB flag.

[Setting conditions]

- TCNT = TGRB when the TGRB register functions as an output compare register.
- The value of TCNT is transferred to the TGRB register due to the input capture signal when the TGRB register functions as an input capture register.

TGFC flag (Input Capture/Output Compare Flag C)

The TGFC flag indicates generation of TGRC input capture or compare match for MTU3, MTU4, MTU6, or MTU7. Only writing 0 to this bit in order to clear the flag is possible.

[Clearing condition]

- When TGFC = 1, the TGFC flag is read, and then 0 is written to the TGFC flag.

[Setting conditions]

- TCNT = TGRC when the TGRC register functions as an output compare register.
- The value of TCNT is transferred to the TGRC register due to the input capture signal when the TGRC register functions as an input capture register.

This bit is reserved for MTU1 and MTU2. This bit is read as 0. The write value should be 0.

TGFD flag (Input Capture/Output Compare Flag D)

The TGFD flag indicates generation of TGRD input capture or compare match for MTU3, MTU4, MTU6, or MTU7. Only writing 0 to this bit in order to clear the flag is possible.

[Clearing condition]

- When TGFD = 1, the TGFD flag is read, and then 0 is written to the TGFD flag.

[Setting conditions]

- TCNT = TGRD when the TGRD register functions as the output compare register.
- The value of TCNT is transferred to the TGRD register due to the input capture signal when the TGRD register functions as an input capture register.

This bit is reserved for MTU1 and MTU2. This bit is read as 0. The write value should be 0.

TCFV flag (Overflow flag)

The TCFV flag indicates generation of TCNT overflow. Only writing 0 to this bit in order to clear the flag is possible.

[Clearing condition]

- When TCFV = 1, the TCFV flag is read, and then 0 is written to the TCFV flag.

[Setting conditions]

- The value of TCNT overflows (changed from 0xFFFF to 0x0000).

For MTU4 or MTU7, the TCFV flag is also set to 1 when the value of TCNT of MTU4 or MTU7 underflows (changed from 0x0001 to 0x0000) in complementary PWM mode.

TCFU flag (Underflow flag)

The TCFU flag indicates generation of TCNT underflow when MTU1 and MTU2 are in phase counting mode. Only writing 0 to this bit in order to clear the flag is possible.

[Clearing condition]

- When TCFU = 1, TCFU is read, and then 0 is written to TCFU.

[Setting conditions]

- The value of TCNT underflows (changed from 0x0000 to 0xFFFF).

This bit is reserved for MTU3, MTU4, MTU6, and MTU7. This bit is read as 0. The write value should be 0.

TCFD flag (Count Direction Flag)

The TCFD flag indicates the direction in which TCNT is counting in MTU1 to MTU4, MTU6, and MTU7.

17.2.11 TBTM : Timer Buffer Operation Transfer Mode Register

Base address: MTU0 = 0x9000_1300

Offset address: 0x026

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TTSE	TTSB	TTSA
Value after reset:	0	0	0	0	0	0	0	0

Base address: MTU3 = 0x9000_1100, MTU4 = 0x9000_1200, MTU6 = 0x9000_1900, MTU7 = 0x9000_1A00

Offset address: 0x138 (MTU3, MTU6), 0x039 (MTU4, MTU7)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TTSB	TTSA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TTSA	Timing Select A 0: When compare match A occurs in each channel, data is transferred from TGRC to TGRA 1: When TCNT is cleared in each channel, data is transferred from TGRC to TGRA	R/W
1	TTSB	Timing Select B 0: When compare match B occurs in each channel, data is transferred from TGRD to TGRB 1: When TCNT is cleared in each channel, data is transferred from TGRD to TGRB	R/W
2	TTSE	Timing Select E 0: When compare match E occurs in MTU0, data is transferred from MTU0.TGRF to MTU0.TGRE 1: When MTU0.TCNT is cleared in MTU0, data is transferred from MTU0.TGRF to MTU0.TGRE	R/W
7:37:2	—	These bits are read as 0. The write value should be 0.	R/W

TBTM specifies the timing for transferring data from the buffer register to the timer general register in PWM mode. The MTU has a total of five TBTM registers, one each for MTU0, MTU3, MTU4, MTU6, and MTU7.

TTSA bit (Timing Select A)

This bit specifies the timing for transferring data from TGRC to TGRA in each channel when they are used together for buffer operation. When a channel is not set to PWM mode, do not set the TTSA bit in the channel to 1.

TTSB bit (Timing Select B)

This bit specifies the timing for transferring data from TGRD to TGRB in each channel when they are used together for buffer operation. When a channel is not set to PWM mode, do not set the TTSB bit in the channel to 1.

TTSE bit (Timing Select E)

This bit specifies the timing for transferring data from MTU0.TGRF to MTU0.TGRE when they are used together for buffer operation.

In MTU3, MTU4, MTU6, and MTU7, this bit is reserved. It is read as 0 and the write value should be 0. When a channel is not set to PWM mode, do not set the TTSE bit in the channel to 1.

17.2.12 TICCR : Timer Input Capture Control Register

Base address: MTU1 = 0x9000_1380

Offset address: 0x010

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	I2BE	I2AE	I1BE	I1AE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	I1AE	Input Capture Enable 0: Does not include the MTIOC1A pin in the MTU2.TGRA input capture conditions 1: Includes the MTIOC1A pin in the MTU2.TGRA input capture conditions	R/W
1	I1BE	Input Capture Enable 0: Does not include the MTIOC1B pin in the MTU2.TGRB input capture conditions 1: Includes the MTIOC1B pin in the MTU2.TGRB input capture conditions	R/W
2	I2AE	Input Capture Enable 0: Does not include the MTIOC2A pin in the MTU1.TGRA input capture conditions 1: Includes the MTIOC2A pin in the MTU1.TGRA input capture conditions	R/W
3	I2BE	Input Capture Enable 0: Does not include the MTIOC2B pin in the MTU1.TGRB input capture conditions 1: Includes the MTIOC2B pin in the MTU1.TGRB input capture conditions	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

TICCR specifies input capture conditions when MTU1.TCNT and MTU2.TCNT are cascaded. The MTU has one TICCR for MTU1.

17.2.13 TSYCR : Timer Synchronous Clear Register

Base address: MTU6 = 0x9000_1900

Offset address: 0x150

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CE0A	CE0B	CE0C	CE0D	CE1A	CE1B	CE2A	CE2B
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CE2B	Clear Enable 2B 0: Disables counter clearing by the MTU2.TGIB2 interrupt generation timing. 1: Enables counter clearing by the MTU2.TGIB2 interrupt generation timing.	R/W
1	CE2A	Clear Enable 2A 0: Disables counter clearing by the MTU2.TGIA2 interrupt generation timing.*1 1: Enables counter clearing by the MTU2.TGIA2 interrupt generation timing.*1	R/W
2	CE1B	Clear Enable 1B 0: Disables counter clearing by the MTU1.TGIB1 interrupt generation timing.*1 1: Enables counter clearing by the MTU1.TGIB1 interrupt generation timing.*1	R/W

Bit	Symbol	Function	R/W
3	CE1A	Clear Enable 1A 0: Disables counter clearing by the MTU1.TGIA1 interrupt generation timing.*1 1: Enables counter clearing by the MTU1.TGIA1 interrupt generation timing.*1	R/W
4	CE0D	Clear Enable 0D 0: Disables counter clearing by the MTU0.TGID0 interrupt generation timing.*1 1: Enables counter clearing by the MTU0.TGID0 interrupt generation timing.*1	R/W
5	CE0C	Clear Enable 0C 0: Disables counter clearing by the MTU0.TGIC0 interrupt generation timing.*1 1: Enables counter clearing by the MTU0.TGIC0 interrupt generation timing.*1	R/W
6	CE0B	Clear Enable 0B 0: Disables counter clearing by the MTU0.TGIB0 interrupt generation timing.*1 1: Enables counter clearing by the MTU0.TGIB0 interrupt generation timing.*1	R/W
7	CE0A	Clear Enable 0A 0: Disables counter clearing by the MTU0.TGIA0 interrupt generation timing.*1 1: Enables counter clearing by the MTU0.TGIA0 interrupt generation timing.*1	R/W

Note 1. This does not depend on the MTUn.TIERn.TGIEm bit setting. (n = 0, 1, 2; m = A, B, C, D)

TSYCR specifies synchronous clear conditions for MTU6.TCNT and MTU7.TCNT. The MTU has one TSYCR for MTU6.

17.2.14 TCNT — Timer Counters

17.2.14.1 MTUn.TCNT : Timer Counter (n = 0 to 4, 6, 7)

Base address: MTU0 = 0x9000_1300, MTU1 = 0x9000_1380, MTU2 = 0x9000_1400, MTU3 = 0x9000_1100
MTU4 = 0x9000_1200, MTU6 = 0x9000_1900, MTU7 = 0x9000_1A00

Offset address: 0x006 (MTU0, MTU1, MTU2), 0x110 (MTU3, MTU6), 0x012 (MTU4, MTU7)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15	n/a	Timer Counter	R/W

Note: The TCNT counters should be accessed in 16-bit units.

17.2.14.2 MTU5.TCNTx : Timer Counter x (x = U, V, W)

Base address: MTU5 = 0x9000_1C00

Offset address: 0x080 (TCNTU)
0x090 (TCNTV)
0x0A0 (TCNTW)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

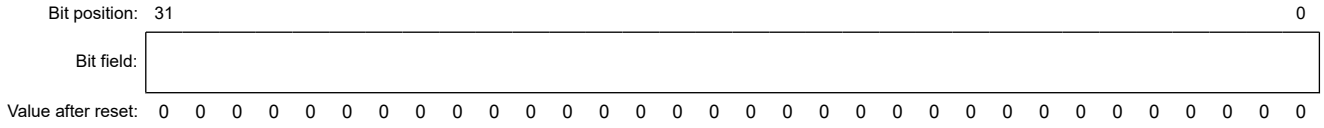
Bit	Symbol	Function	R/W
15	n/a	Timer Counter	R/W

Note: The TCNT counters should be accessed in 16-bit units.

17.2.14.3 MTU8.TCNT : Timer Counter

Base address: MTU8 = 0x9000_1600

Offset address: 0x008



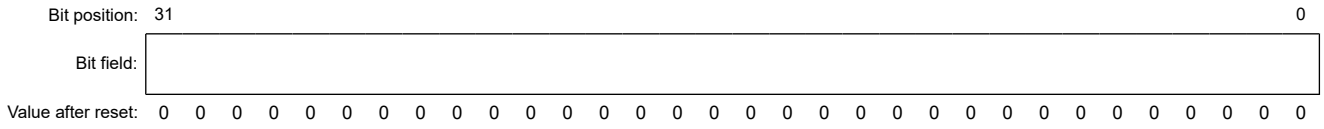
Bit	Symbol	Function	R/W
31:0	n/a	Timer Counter	R/W

Note: The TCNT counters should be accessed in 32-bit units.

17.2.15 TCNTLW : Timer Longword Counter

Base address: MTU1 = 0x9000_1380

Offset address: 0x020



Bit	Symbol	Function	R/W
31:0	n/a	The TCNTLW counter is a 32-bit readable/writable counter. Only one counter of this type is provided, and is formed by combining MTU1.TCNT and MTU2.TCNT. Such operation is only effective when TMDR3.LWA = 1. The TCNTLW counter is initialized to 0x00000000 by a reset. See section 17.2.6. TMDR3 : Timer Mode Register 3 for details. This register can only be used in 32-bit phase counting mode.	R/W

17.2.16 TGR — Timer General Registers

17.2.16.1 MTU0.TGRx : Timer General Register x (x = A, B, C, D, E, F)

Base address: MTU0 = 0x9000_1300

Offset address: 0x008 (TGRA), 0x00A (TGRB), 0x00C (TGRC), 0x00E (TGRD), 0x020 (TGRE), 0x022 (TGRF)



Bit	Symbol	Function	R/W
15:0	n/a	The TGRA, TGRB, TGRC, and TGRD registers function as either output compare or input capture registers. The TGRC and TGRD registers can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD. TGRE and TGRF function as compare registers. When the MTU0.TCNT count matches the TGRE value, an A/D converter start request can be issued. The TGRF register can also be designated for operation as a buffer register. TGR buffer register combination is TGRE and TGRF.	R/W

Note: TGR should be accessed in 16-bit units.

17.2.16.2 MTU1/2.TGRx : Timer General Register x (x = A, B)

Base address: MTU1 = 0x9000_1380, MTU2 = 0x9000_1400

Offset address: 0x008 (TGRA), 0x00A (TGRB)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	n/a	The TGRA and TGRB registers function as either output compare or input capture registers. The TGRA and TGRB registers are read as 0x0000 when TMDR3.LWA = 1. For details, see section 17.2.6. TMDR3 : Timer Mode Register 3.	R/W

Note: TGR should be accessed in 16-bit units.

17.2.16.3 MTU3/6.TGRx : Timer General Register x (x = A, B, C, D, E)

Base address: MTU3 = 0x9000_1100, MTU6 = 0x9000_1900

Offset address: 0x118 (TGRA), 0x11A (TGRB), 0x124 (TGRC), 0x126 (TGRD), 0x172 (TGRE)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	n/a	The TGRA, TGRB, TGRC, and TGRD registers function as either output compare or input capture registers. The TGRC and TGRD registers for MTU3 and MTU6 can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD.	R/W

Note: TGR should be accessed in 16-bit units.

17.2.16.4 MTU4/7.TGRx : Timer General Register x (x = A, B, C, D, E, F)

Base address: MTU4 = 0x9000_1200, MTU7 = 0x9000_1A00

Offset address: 0x01C (TGRA), 0x01E (TGRB), 0x028 (TGRC), 0x02A (TGRD), 0x074 (TGRE), 0x076 (TGRF)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	n/a	The TGRA, TGRB, TGRC, and TGRD registers function as either output compare or input capture registers. The TGRC and TGRD registers for MTU4 and MTU7 can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD.	R/W

Note: TGR should be accessed in 16-bit units.

17.2.16.5 MTU8.TGRx : Timer General Register x (x = A, B, C, D)

Base address: MTU8 = 0x9000_1600

Offset address: 0x00C (TGRA), 0x010 (TGRB), 0x014 (TGRC), 0x018 (TGRD)

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The TGRA, TGRB, TGRC, and TGRD registers function as either output compare or input capture registers. The TGRC and TGRD registers can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD.	R/W

Note: TGR should be accessed in 32-bit units.

17.2.16.6 MTU5.TGRx : Timer General Register x (x = U, V, W)

Base address: MTU5 = 0x9000_1C00

Offset address: 0x082 (TGRU), 0x092 (TGRV), 0x0A2 (TGRW)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	n/a	MTU5.TGRU, MTU5.TGRV, and MTU5.TGRW function as compare match, input capture, or external pulse width measurement registers.	R/W

Note: TGR should be accessed in 16-bit units.

17.2.17 TGRxLW : Timer Longword General Register x (x = A, B)

Base address: MTU1 = 0x9000_1380

Offset address: 0x024 (TGRALW)
0x028 (TGRBLW)

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The TGRnLW register is a 32-bit readable/writable register. Two general registers of this type are provided, and are formed by combining MTU1.TGRn and MTU2.TGRn. Such operation is only effective when TMDR3.LWA = 1. See section 17.2.6. TMDR3 : Timer Mode Register 3 for details. The TGRnLW register functions as an output compare or input capture register when TMDR3.LWA = 1. This register can only be used in 32-bit phase counting mode.	R/W

17.2.18 TSTR — Timer Start Registers

17.2.18.1 TSTRA : Timer Start Register A

Base address: MTU = 0x9000_1000

Offset address: 0x280

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CST4	CST3	—	—	CST8	CST2	CST1	CST0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CST0	Counter Start 0 0: MTU0.TCNT counting is stopped 1: MTU0.TCNT performs count operation	R/W
1	CST1	Counter Start 1 0: MTU1.TCNT counting is stopped 1: MTU1.TCNT performs count operation	R/W
2	CST2	Counter Start 2 0: MTU2.TCNT counting is stopped 1: MTU2.TCNT performs count operation	R/W
3	CST8	Counter Start 8 0: MTU8.TCNT counting is stopped 1: MTU8.TCNT performs count operation	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	CST3	Counter Start 3 0: MTU3.TCNT counting is stopped 1: MTU3.TCNT performs count operation	R/W
7	CST4	Counter Start 4 0: MTU4.TCNT counting is stopped 1: MTU4.TCNT performs count operation	R/W

Note: When 1 is written to a bit in TCSYSTR, the corresponding bit in TSTRA is also set to 1 automatically.

CSTn bits (Counter Start n (n = 0 to 4, 8))

Each bit starts or stops TCNT in the corresponding channel.

If 0 is written to the CSTn bit during operation with the MTIOC pin designated for output, the counter stops. At this time the MTIOC pin outputs the initial output level set in the TOCR1A or TOCR2A register in complementary PWM mode or reset-synchronized PWM mode, but the output compare signal level from the MTIOC pin is retained in the other modes. If TIOR is written to while the CSTn bit is 0, the pin output level will be changed to the specified initial output value.

17.2.18.2 TSTRB : Timer Start Register B

Base address: MTU = 0x9000_1000

Offset address: 0xA80

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CST7	CST6	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
6	CST6	Counter Start 6 0: MTU6.TCNT counting is stopped 1: MTU6.TCNT performs count operation	R/W
7	CST7	Counter Start 7 0: MTU7.TCNT counting is stopped 1: MTU7.TCNT performs count operation	R/W

Note: When 1 is written to a bit in TCSYSTR, the corresponding bit in TSTRB is also set to 1 automatically.

CSTn bits (Counter Start n (n = 6, 7))

Each bit starts or stops TCNT in the corresponding channel.

If 0 is written to the CSTn bit during operation with the MTIOC pin designated for output, the counter stops. At this time the MTIOC pin outputs the initial output level set in the TOCR1B or TOCR2B register in complementary PWM mode or reset-synchronized PWM mode, but the output compare signal level from the MTIOC pin is retained in the other modes. If TIOR is written to while the CSTn bit is 0, the pin output level will be changed to the specified initial output value.

17.2.18.3 MTU5.TSTR : Timer Start Register

Base address: MTU5 = 0x9000_1C00

Offset address: 0x0B4

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	CSTU5	CSTV5	CSTW5
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CSTW5	Counter Start W5 0: MTU5.TCNTW counting is stopped 1: MTU5.TCNTW performs count operation	R/W
1	CSTV5	Counter Start V5 0: MTU5.TCNTV counting is stopped 1: MTU5.TCNTV performs count operation	R/W
2	CSTU5	Counter Start U5 0: MTU5.TCNTU counting is stopped 1: MTU5.TCNTU performs count operation	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

17.2.19 TSYRA : Timer Synchronous Register A

Base address: MTU = 0x9000_1000

Offset address: 0x281

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SYNC0	Timer Synchronous Operation 0 0: MTU0.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU0.TCNT performs synchronous operation (TCNT synchronous presetting/ synchronous clearing is enabled).	R/W

Bit	Symbol	Function	R/W
1	SYNC1	Timer Synchronous Operation 1 0: MTU1.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU1.TCNT performs synchronous operation (TCNT synchronous presetting/ synchronous clearing is enabled).	R/W
2	SYNC2	Timer Synchronous Operation 2 0: MTU2.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU2.TCNT performs synchronous operation (TCNT synchronous presetting/ synchronous clearing is enabled).	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	SYNC3	Timer Synchronous Operation 3 0: MTU3.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU3.TCNT performs synchronous operation (TCNT synchronous presetting/ synchronous clearing is enabled).	R/W
7	SYNC4	Timer Synchronous Operation 4 0: MTU4.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU4.TCNT performs synchronous operation (TCNT synchronous presetting/ synchronous clearing is enabled).	R/W

TSYRA selects independent operation or synchronous operation of TCNT in MTU0 to MTU4. A channel performs synchronous operation when the corresponding bit in TSYRA is set to 1.

SYNCn bits (Timer Synchronous Operation n)

Each bit selects whether operation is independent of or synchronized with other channels.

When synchronous operation is selected, the TCNT synchronous presetting of multiple channels and synchronous clearing by counter clearing on another channel are possible. To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of the TCR.CCLR[2:0] bits.

17.2.20 TSYRB : Timer Synchronous Register B

Base address: MTU = 0x9000_1000

Offset address: 0xA81

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SYNC 7	SYNC 6	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	SYNC6	Timer Synchronous Operation 6 0: MTU6.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU6.TCNT performs synchronous operation (TCNT synchronous presetting/ synchronous clearing is enabled).	R/W
7	SYNC7	Timer Synchronous Operation 7 0: MTU7.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU7.TCNT performs synchronous operation (TCNT synchronous presetting/ synchronous clearing is enabled).	R/W

TSYRB selects independent operation or synchronous operation of TCNT in MTU6 and MTU7. A channel performs synchronous operation when the corresponding bit in TSYRB is set to 1.

SYNCn bits (Timer Synchronous Operation n)

Each bit selects whether operation is independent of or synchronized with other channels.

When synchronous operation is selected, the TCNT synchronous presetting of multiple channels and synchronous clearing by counter clearing on another channel are possible. To set synchronous operation, the SYNC bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNC bit, the TCNT clearing source must also be set by means of the TCR.CCLR[2:0] bits.

17.2.21 TCSYSTR : Timer Counter Synchronous Start Register

Base address: MTU = 0x9000_1000

Offset address: 0x282

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SCH0	SCH1	SCH2	SCH3	SCH4	—	SCH6	SCH7
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SCH7	Synchronous Start 7 0: Does not specify synchronous start for MTU7.TCNT 1: Specifies synchronous start for MTU7.TCNT	R/W*1
1	SCH6	Synchronous Start 6 0: Does not specify synchronous start for MTU6.TCNT 1: Specifies synchronous start for MTU6.TCNT	R/W*1
2	—	This bit is read as 0. The write value should be 0.	R/W
3	SCH4	Synchronous Start 4 0: Does not specify synchronous start for MTU4.TCNT 1: Specifies synchronous start for MTU4.TCNT	R/W*1
4	SCH3	Synchronous Start 3 0: Does not specify synchronous start for MTU3.TCNT 1: Specifies synchronous start for MTU3.TCNT	R/W*1
5	SCH2	Synchronous Start 2 0: Does not specify synchronous start for MTU2.TCNT 1: Specifies synchronous start for MTU2.TCNT	R/W*1
6	SCH1	Synchronous Start 1 0: Does not specify synchronous start for MTU1.TCNT 1: Specifies synchronous start for MTU1.TCNT	R/W*1
7	SCH0	Synchronous Start 0 0: Does not specify synchronous start for MTU0.TCNT 1: Specifies synchronous start for MTU0.TCNT	R/W*1

Note 1. Only 1 can be written to this bit, and doing so sets the register. TCSYSTR is automatically cleared to 0 after 1 is written to.

TCSYSTR specifies synchronous start of the counters.

SCH7 bit (Synchronous Start 7)

This bit controls synchronous start of MTU7.TCNT.

[Clearing condition]

- When 1 is set to the TSTRB.CST7 bit while SCH7 = 1

SCH6 bit (Synchronous Start 6)

This bit controls synchronous start of MTU6.TCNT.

[Clearing condition]

- When 1 is set to the TSTRB.CST6 bit while SCH6 = 1

SCH4 bit (Synchronous Start 4)

This bit controls synchronous start of MTU4.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST4 bit while SCH4 = 1

SCH3 bit (Synchronous Start 3)

This bit controls synchronous start of MTU3.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST3 bit while SCH3 = 1

SCH2 bit (Synchronous Start 2)

This bit controls synchronous start of MTU2.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST2 bit while SCH2 = 1

SCH1 bit (Synchronous Start 1)

This bit controls synchronous start of MTU1.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST1 bit while SCH1 = 1

SCH0 bit (Synchronous Start 0)

This bit controls synchronous start of MTU0.TCNT.

[Clearing condition]

- When 1 is set to the TSTRA.CST0 bit while SCH0 = 1

17.2.22 TRWERx : Timer Read/Write Enable Register x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x284 (TRWERA)
0xA84 (TRWERB)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	RWE
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	RWE	Read/Write Enable 0: Read/write access to the registers is disabled 1: Read/write access to the registers is enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

TRWERA enables or disables access to the registers and counters that have write-protection capability against accidental modification in MTU3 and MTU4.

TRWERB enables or disables access to the registers and counters that have write-protection capability against accidental modification in MTU6 and MTU7.

RWE bit (Read/Write Enable)

This bit enables or disables access to the registers that have write-protection capability against accidental modification.

[Clearing condition]

- When 0 is written to the RWE bit after reading RWE = 1

Note:
- **Registers and Counters having Write-Protection Capability against Accidental Modification (TRWERA)**
  24 registers: MTUn.TCR, MTUn.TCR2, MTUn.TMDR1, MTUn.TIORH, MTUn.TIORL, MTUn.TIER, MTUn.TGRA, MTUn.TGRB, MTU.TOERA, MTU.TOCR1A, MTU.TOCR2A, MTU.TGCRA, MTU.TCDRA, MTU.TDDRA, and MTUn.TCNT (n = 3, 4)
- **Registers and Counters having Write-Protection Capability against Accidental Modification (TRWERB)**
  23 registers: MTUn.TCR, MTUn.TCR2, MTUn.TMDR1, MTUn.TIORH, MTUn.TIORL, MTUn.TIER, MTUn.TGRA, MTUn.TGRB, MTUn.MTU.TOERB, MTU.TOCR1B, MTU.TOCR2B, MTU.TCDRB, MTU.TDDRB, and MTUn.TCNT (n = 6, 7)

17.2.23 TOERA : Timer Output Master Enable Register A

Base address: MTU = 0x9000_1000

Offset address: 0x20A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OE3B	Master Enable MTIOC3B 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
1	OE4A	Master Enable MTIOC4A 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
2	OE4B	Master Enable MTIOC4B 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
3	OE3D	Master Enable MTIOC3D 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
4	OE4C	Master Enable MTIOC4C 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
5	OE4D	Master Enable MTIOC4D 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
7:6	—	These bits are read as 1. The write value should be 1.	R/W

Note 1. The non-active level depends on the setting of the timer output control register 1 or 2 (TOCR 1/2). See section 17.2.25. TOCR1x : Timer Output Control Register 1x (x = A, B) and section 17.2.26. TOCR2x : Timer Output Control Register 2x (x = A, B) for details. When MTU output is in a mode other than complementary PWM mode or reset-synchronized PWM mode, set the relevant bit to 1. If the setting is 0, the non-active level will be output due to this setting of the timer output control register.

TOERA enables or disables output settings for output pins MTIOC4D, MTIOC4C, MTIOC3D, MTIOC4B, MTIOC4A, and MTIOC3B.

These pins do not output correctly if the bits in the TOERA register have not been set. In MTU3 and MTU4, set TOERA prior to setting TIOR.

Set MTU.TOERA after setting the CST3 and CST4 bits in MTU.TSTRA to 0 (see Figure 17.44 and Figure 17.48).

17.2.24 TOERB : Timer Output Master Enable Register B

Base address: MTU = 0x9000_1000

Offset address: 0xA0A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	OE7D	OE7C	OE6D	OE7B	OE7A	OE6B
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OE6B	Master Enable MTIOC6B 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
1	OE7A	Master Enable MTIOC7A 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
2	OE7B	Master Enable MTIOC7B 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
3	OE6D	Master Enable MTIOC6D 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
4	OE7C	Master Enable MTIOC7C 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
5	OE7D	Master Enable MTIOC7D 0: MTU output is disabled (non-active level)*1 1: MTU output is enabled	R/W
7:6	—	These bits are read as 1. The write value should be 1.	R/W

Note 1. The non-active level depends on the setting of the timer output control register 1 or 2 (TOCR 1/2). See section 17.2.25. TOCR1x : Timer Output Control Register 1x (x = A, B) and section 17.2.26. TOCR2x : Timer Output Control Register 2x (x = A, B) for details. When MTU output is in a mode other than complementary PWM mode or reset-synchronized PWM mode, set the relevant bit to 1. If the setting is 0, the non-active level will be output due to this setting of the timer output control register.

TOERB enables or disables output settings for output pins MTIOC7D, MTIOC7C, MTIOC6D, MTIOC7B, MTIOC7A, and MTIOC6B.

These pins do not output correctly if the bits in the TOERB register have not been set. In MTU6 and MTU7, set TOERB prior to setting TIOR.

Set MTU.TOERB after setting the CST6 and CST7 bits in MTU.TSTRB to 0 (see Figure 17.44 and Figure 17.48).

17.2.25 TOCR1x : Timer Output Control Register 1x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x20E (TOCR1A)
0xA0E (TOCR1B)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OLSP	Output Level Select P*1 *3 See Table 17.44.	R/W

Bit	Symbol	Function	R/W
1	OLSN	Output Level Select N*1 *3 See Table 17.45.	R/W
2	TOCS	TOC Select 0: TOCR1j setting is selected (j = A, B). 1: TOCR2j setting is selected (j = A, B).	R/W
3	TOCL	TOC Register Write Protection*2 *4 0: Write access to the TOCS, OLSN, and OLSP bits is enabled. 1: Write access to the TOCS, OLSN, and OLSP bits is disabled.	R/W
5:4	—	These bits are read as 0. The write value should be 0.	R/W
6	PSYE	PWM Synchronous Output Enable 0: Toggle output is disabled. 1: Toggle output is enabled.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. Setting the TOCR1j.TOCS bit to 0 makes this bit setting valid.
Note 2. Setting the TOCR1j.TOCL bit to 1 prevents accidental modification when the CPU goes out of control.
Note 3. If dead-time is not generated, the negative-phase output is the exact inverse of the positive-phase output. In this case, only the OLSP bit is valid.
Note 4. This bit can be set to 1 only once after a reset. After 1 is written, 0 cannot be written to the bit.

TOCR1A and TOCR1B enable or disable PWM-synchronized toggle output in complementary PWM mode and reset synchronized PWM mode, and control inversion of PWM output level.

OLSP bit (Output Level Select P)

This bit selects the positive-phase output level in reset-synchronized PWM mode and complementary PWM mode. The initial output is selected while the counter is stopped.

OLSN bit (Output Level Select N)

This bit selects the negative-phase output level in reset-synchronized PWM mode and complementary PWM mode. The initial output is selected while the counter is stopped.

TOCS bit (TOC Select)

This bit selects either the TOCR1j or TOCR2j (j = A, B) setting to be used for the output level in complementary PWM mode and reset-synchronized PWM mode.

TOCL bit (TOC Register Write Protection)

This bit enables or disables write access to the TOCS, OLSN, and OLSP bits in TOCR1j (j = A, B).

PSYE bit (PWM Synchronous Output Enable)

This bit enables or disables toggle output synchronized with the PWM period.

Table 17.44 Output level select function

Bit 0	Function			
			Compare match output	
OLSP	Initial output	Active level	Up-counting	Down-counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Table 17.45 Output level select function

Bit 1	Function			
			Compare match output	
OLSN	Initial output	Active level	Up-counting	Down-counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Figure 17.3 shows an example of output in complementary PWM mode (one phase) when OLSN = 1 and OLSP = 1.

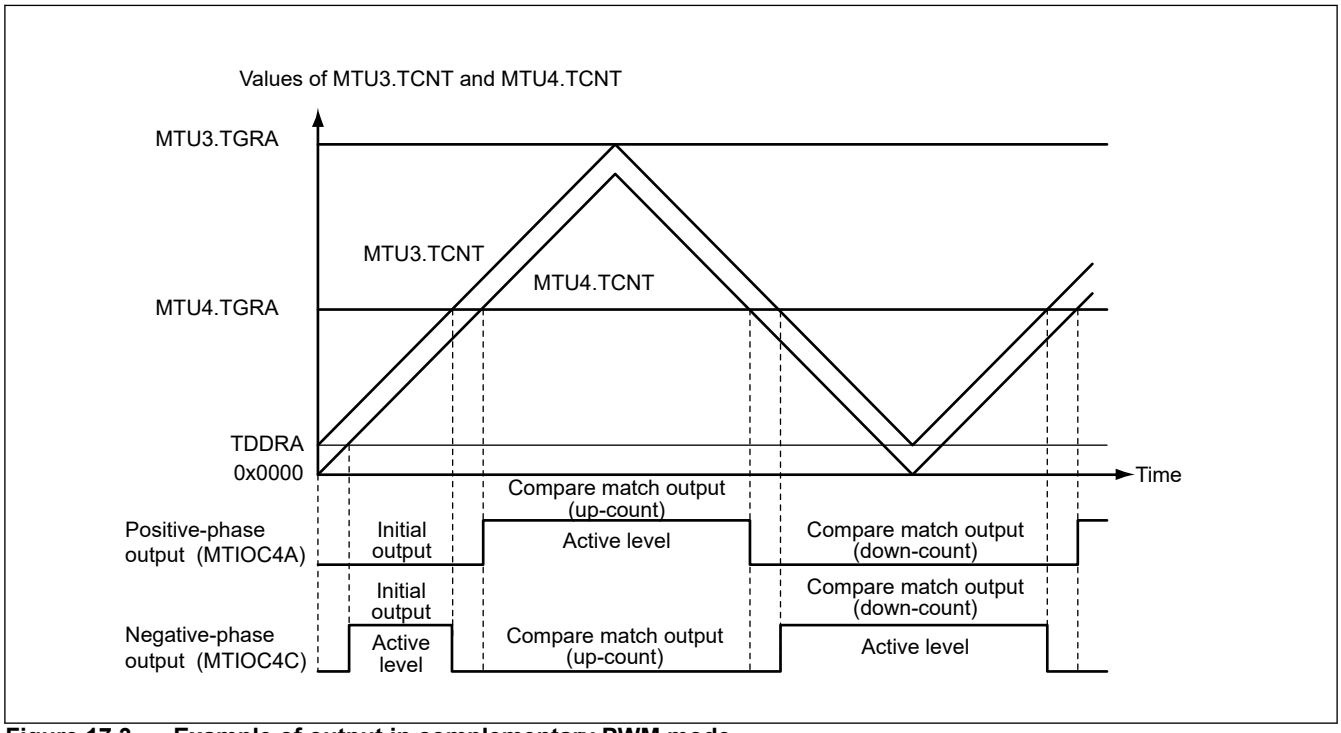


Figure 17.3 Example of output in complementary PWM mode

17.2.26 TOCR2x : Timer Output Control Register 2x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x20F (TOCR2A)
0xA0F (TOCR2B)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OLS1P	Output Level Select 1P*1 *2 This bit selects the output level on MTIOC3B or MTIOC6B in reset-synchronized PWM mode and complementary PWM mode. See Table 17.46.	R/W
1	OLS1N	Output Level Select 1N*1 *2 This bit selects the output level on MTIOC3D or MTIOC6D in reset-synchronized PWM mode and complementary PWM mode. See Table 17.47.	R/W
2	OLS2P	Output Level Select 2P*1 *2 This bit selects the output level on MTIOC4A or MTIOC7A in reset-synchronized PWM mode and complementary PWM mode. See Table 17.48.	R/W
3	OLS2N	Output Level Select 2N*1 *2 This bit selects the output level on MTIOC4C or MTIOC7C in reset-synchronized PWM mode and complementary PWM mode. See Table 17.49.	R/W

Bit	Symbol	Function	R/W
4	OLS3P	Output Level Select 3P*1 *2 This bit selects the output level on MTIOC4B or MTIOC7B in reset-synchronized PWM mode and complementary PWM mode. See Table 17.50.	R/W
5	OLS3N	Output Level Select 3N*1 *2 This bit selects the output level on MTIOC4D or MTIOC7D in reset-synchronized PWM mode and complementary PWM mode. See Table 17.51.	R/W
7:6	BF[1:0]	TOLBR Buffer Transfer Timing Select These bits select the timing for transferring data from TOLBRj to TOCR2j. See Table 17.52 for details.	R/W

Note 1. Setting the TOCR1j.TOCS bit to 1 makes this bit setting valid (j = A, B).
Note 2. If dead-time is not generated, the negative-phase output is the exact inverse of the positive-phase output. In this case, only the OLSiP bits are valid (i = 1 to 3).

TOCR2A and TOCR2B control inversion of PWM output level in complementary PWM mode and reset-synchronized PWM mode.

The initial output is selected while the counter is stopped.

Table 17.46 MTIOCmB output level select function

Bit 0	Function			
			Compare match output	
OLS1P	Initial output	Active level	Up-counting	Down-counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Note: m = 3, 6

Table 17.47 MTIOCmD output level select function

Bit 1	Function			
			Compare match output	
OLS1N	Initial output	Active level	Up-counting	Down-counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: m = 3, 6
Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 17.48 MTIOCmA output level select function

Bit 2	Function			
			Compare match output	
OLS2P	Initial output	Active level	Up-counting	Down-counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Note: m = 4, 7

Table 17.49 MTIOCmC output level select function

Bit 3	Function			
			Compare match output	
OLS2N	Initial output	Active level	Up-counting	Down-counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: m = 4, 7

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 17.50 MTIOCmB output level select function

Bit 4	Function			
			Compare match output	
OLS3P	Initial output	Active level	Up-counting	Down-counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

Note: m = 4, 7

Table 17.51 MTIOCmD output level select function

Bit 5	Function			
			Compare match output	
OLS3N	Initial output	Active level	Up-counting	Down-counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

Note: m = 4, 7
Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 17.52 Setting of TOCR2j.BF[1:0] bits

Bit 7	Bit 6	Description	
BF[1]	BF[0]	Complementary PWM mode	Reset-synchronized PWM mode
0	0	Does not transfer data from the buffer register (TOLBRj) to TOCR2j.	Does not transfer data from the buffer register (TOLBRj) to TOCR2j.
0	1	Transfers data from the buffer register (TOLBRj) to TOCR2j at the crest of the MTUn.TCNT count.	Transfers data from the buffer register (TOLBRj) to TOCR2j when MTUm.TCNT or MTUn.TCNT is cleared.
1	0	Transfers data from the buffer register (TOLBRj) to TOCR2j at the trough of the MTUn.TCNT count.	Setting prohibited
1	1	Transfers data from the buffer register (TOLBRj) to TOCR2j at the crest and trough of the MTUn.TCNT count.	Setting prohibited

Note: n = 4, 7; m = 3, 6; j = A, B

17.2.27 TOLBRx : Timer Output Level Buffer Register x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x236 (TOLBRA)
0xA36 (TOLBRB)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OLS1P	Output Level Select 1P Specify the buffer value to be transferred to the OLS1P bit in TOCR2A and TOCR2B.	R/W
1	OLS1N	Output Level Select 1N Specify the buffer value to be transferred to the OLS1N bit in TOCR2A and TOCR2B.	R/W
2	OLS2P	Output Level Select 2P Specify the buffer value to be transferred to the OLS2P bit in TOCR2A and TOCR2B.	R/W

Bit	Symbol	Function	R/W
3	OLS2N	Output Level Select 2N Specify the buffer value to be transferred to the OLS2N bit in TOCR2A and TOCR2B.	R/W
4	OLS3P	Output Level Select 3P Specify the buffer value to be transferred to the OLS3P bit in TOCR2A and TOCR2B.	R/W
5	OLS3N	Output Level Select 3N Specify the buffer value to be transferred to the OLS3N bit in TOCR2A and TOCR2B.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

TOLBRA and TOLBRB are buffer registers for TOCR2A and TOCR2B and specify the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Figure 17.4 shows an example of the PWM output level setting procedure in buffer operation.

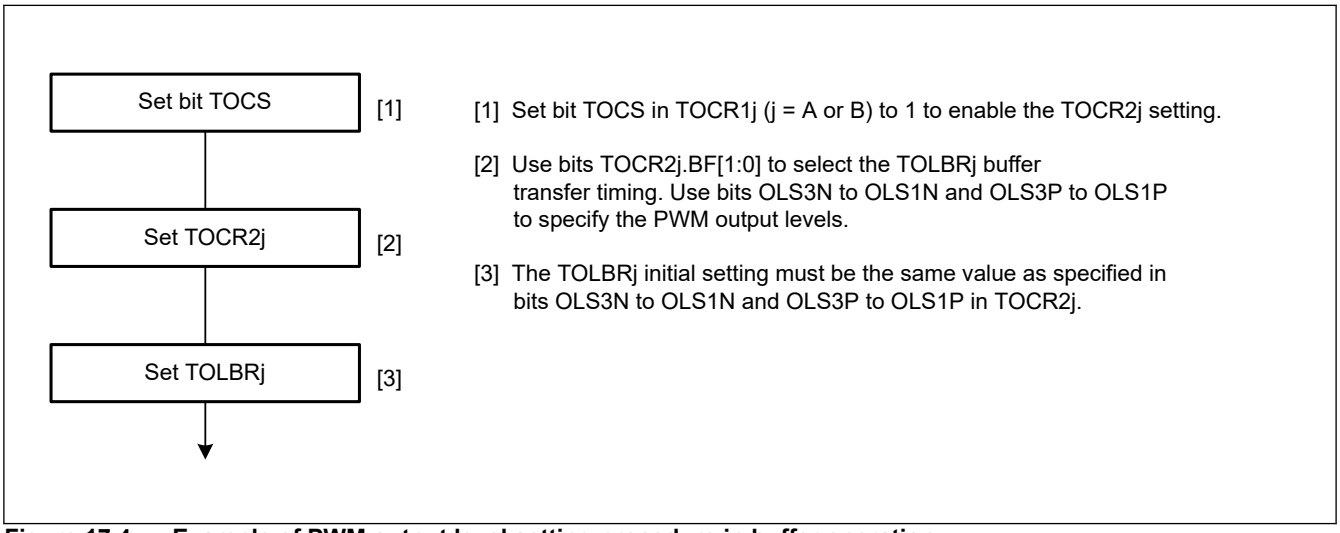


Figure 17.4 Example of PWM output level setting procedure in buffer operation

17.2.28 TGCRA : Timer Gate Control Register A

Base address: MTU = 0x9000_1000

Offset address: 0x20D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	BDC	N	P	FB	WF	VF	UF
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	UF	Output Phase Switch These bits turn on or off the positive-phase/negative-phase output. The setting of these bits is valid only when FB = 1. In this case, the setting of bit 0 to bit 2 is used instead of the external input. See Table 17.53.	R/W
1	VF	Output Phase Switch See Table 17.53.	R/W
2	WF	Output Phase Switch See Table 17.53.	R/W
3	FB	External Feedback Signal Enable 0: Output is switched by external input (input sources are TGRA, TGRB, and TGRC input capture signals in MTU0) 1: Output is switched by software (UF, VF, and WF settings)	R/W

Bit	Symbol	Function	R/W
4	P	Positive-Phase Output (P) Control 0: Level output 1: Reset-synchronized PWM or complementary PWM output	R/W
5	N	Negative-Phase Output (N) Control 0: Level output 1: Reset-synchronized PWM or complementary PWM output	R/W
6	BDC	Brushless DC Motor 0: Ordinary output 1: Functions of this register are made effective	R/W
7	—	This bit is read as 1. The write value should be 1.	R/W

TGCRA controls the output waveform necessary for brushless DC motor control in reset-synchronized PWM mode and complementary PWM mode. TGCRA register settings are ineffective for anything other than complementary PWM mode and reset-synchronized PWM mode.

UF, VF, and WF bits (Output Phase Switch)

The setting of these bits is valid only when FB = 1. In this case, the setting of bit 0 to bit 2 is used instead of the external input. For details, see Table 17.53.

Table 17.53 Output level select function

Bit 2	Bit 1	Bit 0	Function					
			MTIOC3B	MTIOC4A	MTIOC4B	MTIOC3D	MTIOC4C	MTIOC4D
WF	VF	UF	U phase	V phase	W phase	U phase	V phase	W phase
0	0	0	OFF	OFF	OFF	OFF	OFF	OFF
0	0	1	ON	OFF	OFF	OFF	OFF	ON
0	1	0	OFF	ON	OFF	ON	OFF	OFF
0	1	1	OFF	ON	OFF	OFF	OFF	ON
1	0	0	OFF	OFF	ON	OFF	ON	OFF
1	0	1	ON	OFF	OFF	OFF	ON	OFF
1	1	0	OFF	OFF	ON	ON	OFF	OFF
1	1	1	OFF	OFF	OFF	OFF	OFF	OFF

FB bit (External Feedback Signal Enable)

This bit selects whether the positive-/negative-phase output is switched automatically with the TGRA, TGRB, and TGRC input capture signals in MTU0 or by writing 0 or 1 to bits 2 to 0 in TGCRA. When the TGCRA.FB bit is 0, output of MTU3 and MTU4 can be switched with the TGRA, TGRB, and TGRC input capture signals in MTU0.

P bit (Positive-Phase Output (P) Control)

This bit selects the level output or the reset-synchronized PWM/complementary PWM output for the positive-phase output pins (MTIOC3B, MTIOC4A, and MTIOC4B pins).

N bit (Negative-Phase Output (N) Control)

This bit selects the level output or the reset-synchronized PWM/complementary PWM output for the negative-phase output pins (MTIOC3D, MTIOC4C, and MTIOC4D pins).

BDC bit (Brushless DC Motor)

This bit selects whether to make the functions of TGCRA effective or ineffective.

17.2.29 TCNTSx : Timer Subcounter x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x220 (TCNTSA)
0xA20 (TCNTSB)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	n/a	TCNTSA and TCNTSB are 16-bit read-only counters used only in complementary PWM mode.	R

Note: TCNTSA and TCNTSB should be accessed in 16-bit units.

17.2.30 TCDRx : Timer Cycle Data Register x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x214 (TCDRA)
0xA14 (TCDRB)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	n/a	TCDRA and TCDRB are 16-bit readable/writable registers used only in complementary PWM mode. Set half the PWM carrier period as the TCDRA and TCDRB values. The TCDRA and TCDRB registers are constantly compared with the TCNTSA and TCNTSB counters in complementary PWM mode, respectively. When a match occurs, the TCNTSA and TCNTSB counters switch the count direction (down-count to up-count).	R/W

Note: TCDRA and TCDRB should be accessed in 16-bit units.

17.2.31 TCBRx : Timer Cycle Buffer Register x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x222 (TCBRA)
0xA22 (TCBRB)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	n/a	TCBRA and TCBRB are 16-bit readable/writable registers, used only in complementary PWM mode, that function as buffer registers for TCDRA and TCDRB. The TCBRA and TCBRB values are transferred to TCDRA and TCDRB with the transfer timing set in TMDR1.	R/W

Note: TCBRA and TCBRB should be accessed in 16-bit units.

17.2.32 TDDRx : Timer Dead Time Data Register x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x216 (TDDRA)
0xA16 (TDDRB)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	n/a	TDDRA and TDDRB are 16-bit readable/writable registers, used only in complementary PWM mode, that specify the MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) counter offset value. In complementary PWM mode, when the MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT) counters are cleared and then restarted, the TDDRA (TDDRB) value is loaded into the MTU3.TCNT (MTU6.TCNT) counter and the count operation starts.	R/W

Note: TDDRA and TDDRB should be accessed in 16-bit units.

17.2.33 TDERx : Timer Dead Time Enable Register x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x234 (TDERA)
0xA34 (TDERB)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TDER
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	TDER	Dead Time Enable 0: No dead time is generated 1: Dead time is generated*1	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. TDDRA and TDDRB should be set to 1 or a larger value.

TDERA and TDERB control dead time generation in complementary PWM mode. The MTU has one TDER each for MTU3 and MTU6. TDERA and TDERB should be modified only while TCNT stops.

TDER bit (Dead Time Enable)

This bit specifies whether to generate dead time.

[Clearing condition]

- When 0 is written to TDER after reading TDER = 1

17.2.34 TBTERx : Timer Buffer Transfer Set Register x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x232 (TBTERA)
0xA32 (TBTERB)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	BTE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	BTE[1:0]	Buffer Transfer Disable and Interrupt Skipping Link Setting These bits enable or disable transfer from the buffer registers*1 used in complementary PWM mode to the temporary registers, and specify whether to link the transfer with interrupt skipping function 1. For details, see Table 17.54.	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Applicable buffer registers (TBTERA):
MTU3.TGRC, MTU3.TGRD, MTU4.TGRC, MTU4.TGRD, and TCBRA
Applicable buffer registers (TBTERB):
MTU6.TGRC, MTU6.TGRD, MTU7.TGRC, MTU7.TGRD, and TCBRB

TBTERA and TBTERB enable or disable transfer from the buffer registers used in complementary PWM mode to the temporary registers, and specify whether to link the transfer with interrupt skipping 1 operation.

Table 17.54 Setting of TBTERA.BTE[1:0] bits and TBTERB.BTE[1:0] bits

Bit 1	Bit 0	
BTE[1]	BTE[0]	Description
0	0	Enables transfer from the buffer registers to the temporary registers*1 and does not link the transfer with interrupt skipping function 1.
0	1	Disables transfer from the buffer registers to the temporary registers.
1	0	Links transfer from the buffer registers to the temporary registers with interrupt skipping function 1.*2
1	1	Setting prohibited

Note 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR1. For details, see section 17.3.8. Complementary PWM Mode.
Note 2. When interrupt skipping is disabled the T3AEN and T4VEN (T6AEN and T7VEN) bits are set to 0 in the timer interrupt skipping set register (TITCR1A (TITCR1B)) or the skipping count set bits (T3ACOR and T4VCOR (T6ACOR and T7VCOR)) in TITCR1A (TITCR1B) are set to 0), be sure to disable link of buffer transfer with interrupt skipping (set the BTE1 bit in the timer buffer transfer set register (TBTERA (TBTERB)) to 0). If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed.

17.2.35 TWCRx : Timer Waveform Control Register x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x260 (TWCRA)
0xA60 (TWCRB)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CCE	—	—	—	—	—	SCC	WRE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WRE*3	Waveform Retain Enable 0: Initial values specified in TOCR1A and TOCR2A (TOCR1B and TOCR2B) are output 1: Initial output is inhibited	R/W
1	SCC*1*3	Synchronous Clearing Control (Only valid in TWCRB) 0: Clearing of MTU6.TCNT and MTU7.TCNT in response to synchronous clearing for MTU0, MTU1, MTU2 to MTU6, and MTU7 is enabled. 1: Clearing of MTU6.TCNT and MTU7.TCNT in response to synchronous clearing for MTU0, MTU1, MTU2 to MTU6, and MTU7 is disabled.	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	CCE*2	Compare Match Clear Enable 0: Counters are not cleared at MTU3.TGRA (MTU6.TGRA) compare match 1: Counters are cleared at MTU3.TGRA (MTU6.TGRA) compare match	R/W

Note 1. This bit is only valid in register TWCRB and is a reserved bit in register TWCRA.
Note 2. Do not set to 1 when complementary PWM mode 1 is not selected.

Note 3. Do not set to 1 when complementary PWM mode is not selected.

TWCRA and TWCRB control the output waveform when synchronous counter clearing occurs in MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) in complementary PWM mode and specifies whether to clear the counters at MTU3.TGRA (MTU6.TGRA) compare match.

The CCE bit and WRE bit in TWCRA and TWCRB should be modified only while TCNT stops.

WRE bit (Waveform Retain Enable)

This bit selects the waveform output when synchronous counter clearing occurs in complementary PWM mode.

The initial output is inhibited with this function only when synchronous clearing occurs within the Tb interval at the trough in complementary PWM mode. When synchronous clearing occurs outside this interval, the initial values specified in TOCR1A and TOCR2A are output regardless of the WRE bit setting. The initial values specified in TOCR1A and TOCR2A are also output when synchronous clearing occurs in the Tb interval at the trough immediately after MTU3.TCNT and MTU4.TCNT start operation.

For the Tb interval at the trough in complementary PWM mode, see Figure 17.50.

[Setting condition]

- When 1 is written to the WRE bit after reading WRE = 0

SCC bit (Synchronous Clearing Control)

The setting of this bit selects whether MTU6.TCNT and MTU7.TCNT are or are not cleared when counter-synchronous clearing is generated for MTU0, MTU1, MTU2 to MTU6, and MTU7 in complementary PWM mode.

Make the complementary PWM mode settings for MTU6 and MTU7 when this function is in use. When writing a new value to the SCC bit while the counter is operating, do so in such a way that the values of the CCE and WRE bits are not changed.

Synchronous clearing from the MTU module only becomes disabled due to the setting of the SCC bit when synchronous clearing is generated outside the Tb interval in the trough. If synchronous clearing is generated within the Tb interval in the trough including immediately after the value at which MTU6.TCNT and MTU7.TCNT start, MTU6.TCNT and MTU7.TCNT are cleared.

Regarding the Tb interval in the trough in complementary PWM mode, see Figure 17.50.

[Setting condition]

- Writing of 1 to the SCC bit after reading SCC = 0

The corresponding bit in register TWCRA is reserved and is read as 0. When writing to TWCRA, write 0 to this bit.

CCE bit (Compare Match Clear Enable)

This bit specifies whether to clear counters at MTU3.TGRA (MTU6.TGRA) compare match in complementary PWM mode.

[Setting condition]

- When 1 is written to the CCE bit after reading CCE = 0

17.2.36 NFCR — Noise Filter Control Registers

17.2.36.1 NFCRn : Noise Filter Control Register n (n = 0, 3, 4, 6 to 8)

Base address: MTU_NF = 0x9000_1290

Offset address: 0x000 + 0x01 × n (n = 0 to 4, 8)
0x803 + 0x01 × (n - 6) (n = 6, 7)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	NFCS[1:0]		NFDEN	NFCEN	NFBEN	NFAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NFAEN	Noise Filter A Enable 0: The noise filter for the MTIOCnA pin is disabled 1: The noise filter for the MTIOCnA pin is enabled.	R/W
1	NFBEN	Noise Filter B Enable 0: The noise filter for the MTIOCnB pin is disabled 1: The noise filter for the MTIOCnB pin is enabled	R/W
2	NFCEN	Noise Filter C Enable 0: The noise filter for the MTIOCnC pin is disabled 1: The noise filter for the MTIOCnC pin is enabled	R/W
3	NFDEN	Noise Filter D Enable 0: The noise filter for the MTIOCnD pin is disabled 1: The noise filter for the MTIOCnD pin is enabled	R/W
5:4	NFCS[1:0]	Noise Filter Clock Select 0 0: PCLKH/1 0 1: PCLKH/8 1 0: PCLKH/32 1 1: Clock source for counting	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

The NFCRn register controls noise filtering for the input pins of MTUn.

NFAEN bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the MTIOCnA pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR1.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFBEN bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the MTIOCnB pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR1.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFCEN bit (Noise Filter C Enable)

This bit disables or enables the noise filter for input from the MTIOCnC pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR1.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFDEN bit (Noise Filter D Enable)

This bit disables or enables the noise filter for input from the MTIOCnD pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR1.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFCS[1:0] bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. When setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval before setting the input-capture function. When the NFCS[1:0] bits are set to 11b, i.e. selecting the external clock as the source to drive counting, wait for two cycles of the external clock before setting the input capture function.

17.2.36.2 NFCRm : Noise Filter Control Register m (m = 1, 2)

Base address: MTU_NF = 0x9000_1290

Offset address: 0x001 (NFCR1), 0x002 (NFCR2)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	NFCS[1:0]		—	—	NFBEN	NFAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NFAEN	Noise Filter A Enable 0: The noise filter for the MTIOCmA pin is disabled 1: The noise filter for the MTIOCmA pin is enabled.	R/W
1	NFBEN	Noise Filter B Enable 0: The noise filter for the MTIOCmB pin is disabled 1: The noise filter for the MTIOCmB pin is enabled	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
5:4	NFCS[1:0]	Noise Filter Clock Select 0 0: PCLKH/1 0 1: PCLKH/8 1 0: PCLKH/32 1 1: Clock source for counting	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

The NFCRm register (m = 1, 2) controls noise filtering for the input pins of MTUm.

NFAEN bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the MTIOCmA pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR1.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFBEN bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the MTIOCmB pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, select the output compare function for the relevant pin in the timer I/O control register or set the TMDR1.MD[3:0] bits to a value other than that for normal mode (0000b) before doing so.

NFCS[1:0] bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. When setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval before setting the input-capture function. When the NFCS[1:0] bits are set to 11b, i.e. selecting the external clock as the source to drive counting, wait for two cycles of the external clock before setting the input capture function.

17.2.36.3 NFCRC : Noise Filter Control Register C

Base address: MTU_NF = 0x9000_1290

Offset address: 0x009

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	NFCS[1:0]		NFDEN	NFCEN	NFBEN	NFAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NFAEN	Noise Filter A Enable 0: The noise filter for the MTCLKA pin is disabled 1: The noise filter for the MTCLKA pin is enabled.	R/W
1	NFBEN	Noise Filter B Enable 0: The noise filter for the MTCLKB pin is disabled 1: The noise filter for the MTCLKB pin is enabled	R/W
2	NFCEN	Noise Filter C Enable 0: The noise filter for the MTCLKC pin is disabled 1: The noise filter for the MTCLKC pin is enabled	R/W
3	NFDEN	Noise Filter D Enable 0: The noise filter for the MTCLKD pin is disabled 1: The noise filter for the MTCLKD pin is enabled	R/W
5:4	NFCS[1:0]	Noise Filter Clock Select 0 0: PCLKH/1 0 1: PCLKH/2 1 0: PCLKH/8 1 1: PCLKH/32	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

The NFCRC register controls noise filtering for the external clock input pins of the MTU.

NFAEN bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the MTCLKA pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFBEN bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the MTCLKB pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFCEN bit (Noise Filter C Enable)

This bit disables or enables the noise filter for input from the MTCLKC pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFDEN bit (Noise Filter D Enable)

This bit disables or enables the noise filter for input from the MTCLKD pin. Since changing the value of the bit may lead to the internal generation of an unexpected edge, do so after stopping the internal counter.

NFCS[1:0] bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. After setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval to set the counting start.

17.2.36.4 NFCR5 : Noise Filter Control Register 5

Base address: MTU_NF = 0x9000_1290

Offset address: 0x805

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	NFCS[1:0]		—	NFWEN	NFVEN	NFUEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	NFUEN	Noise Filter U Enable 0: The noise filter for the MTIC5U pin is disabled 1: The noise filter for the MTIC5U pin is enabled	R/W

Bit	Symbol	Function	R/W
1	NFVEN	Noise Filter V Enable 0: The noise filter for the MTIC5V pin is disabled 1: The noise filter for the MTIC5V pin is enabled	R/W
2	NFWEN	Noise Filter W Enable 0: The noise filter for the MTIC5W pin is disabled 1: The noise filter for the MTIC5W pin is enabled	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
5:4	NFCS[1:0]	Noise Filter Clock Select 0 0: PCLKH/1 0 1: PCLKH/8 1 0: PCLKH/32 1 1: Clock source for counting	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

The NFCR5 register controls noise filtering for the input pins of MTU5.

NFUEN bit (Noise Filter U Enable)

This bit disables or enables the noise filter for input from the MTIC5U pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

NFVEN bit (Noise Filter V Enable)

This bit disables or enables the noise filter for input from the MTIC5V pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

NFWEN bit (Noise Filter W Enable)

This bit disables or enables the noise filter for input from the MTIC5W pin. Since unexpected edges may be internally generated when the value of this bit is changed, select the output compare function for the relevant pin in the timer I/O control register.

NFCS[1:0] bits (Noise Filter Clock Select)

These bits set the sampling interval for the noise filters. When setting the NFCS[1:0] bits, wait for two cycles of the selected sampling interval before setting the input-capture function.

17.2.37 TADCR — Timer A/D Converter Start Request Control Registers

17.2.37.1 MTU4.TADCR : Timer A/D Converter Start Request Control Register

Base address: MTU4 = 0x9000_1200

Offset address: 0x040

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BF[1:0]		—	—	—	—	—	—	UT4AE	DT4AE	UT4BE	DT4BE	ITA3AE	ITA4VE	ITB3AE	ITB4VE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ITB4VE	TCIV4 Interrupt Skipping Link Enable*1 *2 *3 0: A/D converter start request signal TRG4BN and TCIV4 interrupt skipping 1 are not linked. 1: A/D converter start request signal TRG4BN and TCIV4 interrupt skipping 1 are linked.	R/W

Bit	Symbol	Function	R/W
1	ITB3AE	TGIA3 Interrupt Skipping Link Enable*1 *2 *3 0: A/D converter start request signal TRG4BN and TGIA3 interrupt skipping 1 are not linked. 1: A/D converter start request signal TRG4BN and TGIA3 interrupt skipping 1 are linked.	R/W
2	ITA4VE	TCIV4 Interrupt Skipping Link Enable*1 *2 *3 0: A/D converter start request signal TRG4AN and TCIV4 interrupt skipping 1 are not linked. 1: A/D converter start request signal TRG4AN and TCIV4 interrupt skipping 1 are linked.	R/W
3	ITA3AE	TGIA3 Interrupt Skipping Link Enable*1 *2 *3 0: A/D converter start request signal TRG4AN and TGIA3 interrupt skipping 1 are not linked. 1: A/D converter start request signal TRG4AN and TGIA3 interrupt skipping 1 are linked.	R/W
4	DT4BE	Down-Count TRG4BN Enable*3 0: A/D converter start requests (TRG4BN) disabled during MTU4.TCNT down-count operation. 1: A/D converter start requests (TRG4BN) enabled during MTU4.TCNT down-count operation.	R/W
5	UT4BE	Up-Count TRG4BN Enable 0: A/D converter start requests (TRG4BN) disabled during MTU4.TCNT up-count operation. 1: A/D converter start requests (TRG4BN) enabled during MTU4.TCNT up-count operation.	R/W
6	DT4AE	Down-Count TRG4AN Enable*3 0: A/D converter start requests (TRG4AN) disabled during MTU4.TCNT down-count operation. 1: A/D converter start requests (TRG4AN) enabled during MTU4.TCNT down-count operation.	R/W
7	UT4AE	Up-Count TRG4AN Enable 0: A/D converter start requests (TRG4AN) disabled during MTU4.TCNT up-count operation. 1: A/D converter start requests (TRG4AN) enabled during MTU4.TCNT up-count operation.	R/W
13:8	—	These bits are read as 0. The write value should be 0.	R/W
15:14	BF[1:0]	MTU4.TADCOBRA/TADCOBRB Transfer Timing Select These bits specify the transfer timing from MTU4.TADCOBRA and MTU4.TADCOBRB to MTU4.TADCORA and MTU4.TADCORB. For detail, see Table 17.55.	R/W

Note: MTU4.TADCR should be accessed in 16-bit units.
Note 1. When interrupt skipping is disabled (the T3AEN and T4VEN bits in TITCR1A are cleared to 0 or the T3ACOR and T4VCOR bits in TITCR1A are cleared to 0), do not link A/D converter start requests with interrupt skipping function 1 (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in MTU4.TADCR to 0).
Note 2. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests will not be issued.
Note 3. Do not set to 1 when complementary PWM mode is not selected.

TADCR enables or disables A/D converter start requests and specifies whether to link A/D converter start requests with interrupt skipping function. The MTU has one TADCR each for MTU4 and MTU7.

Table 17.55 Setting of transfer timing by TADCR.BF[1:0] bits (MTU4) (1 of 2)

BF[1]	BF[0]	Description
0	0	Does not transfer data from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).
0	1	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) when the crest of the MTU4.TCNT count is reached or the MTU4.TGRD is written to in complementary PWM mode, when a compare match occurs between MTU3.TCNT and MTU3.TGRA in reset-synchronized PWM mode, or when a compare match occurs between MTU4.TCNT and MTU4.TGRA in PWM mode 1 or normal operation mode.

Table 17.55 Setting of transfer timing by TADCR.BF[1:0] bits (MTU4) (2 of 2)

BF[1]	BF[0]	Description
1	0	Transfers data from the cycle set buffer register to the cycle set register at the trough of the MTU4.TCNT count. These settings are prohibited when complementary PWM mode is not selected.
1	1	Transfers data from the cycle set buffer register to the cycle set register at the crest and trough of the MTU4.TCNT count. These settings are prohibited when complementary PWM mode is not selected.

17.2.37.2 MTU7.TADCR : Timer A/D Converter Start Request Control Register

Base address: MTU7 = 0x9000_1A00

Offset address: 0x040

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BF[1:0]		—	—	—	—	—	—	UT7AE	DT7AE	UT7BE	DT7BE	ITA6AE	ITA7VE	ITB6AE	ITB7VE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ITB7VE	TCIV7 Interrupt Skipping Link Enable*1 *2 *3 0: A/D converter start request signal TRG7BN and TCIV7 interrupt skipping 1 are not linked. 1: A/D converter start request signal TRG7BN and TCIV7 interrupt skipping 1 are linked.	R/W
1	ITB6AE	TGIA6 Interrupt Skipping Link Enable*1 *2 *3 0: A/D converter start request signal TRG7BN and TGIA6 interrupt skipping 1 are not linked. 1: A/D converter start request signal TRG7BN and TGIA6 interrupt skipping 1 are linked.	R/W
2	ITA7VE	TCIV7 Interrupt Skipping Link Enable*1 *2 *3 0: A/D converter start request signal TRG7AN and TCIV7 interrupt skipping 1 are not linked. 1: A/D converter start request signal TRG7AN and TCIV7 interrupt skipping 1 are linked.	R/W
3	ITA6AE	TGIA6 Interrupt Skipping Link Enable*1 *2 *3 0: A/D converter start request signal TRG7AN and TGIA6 interrupt skipping 1 are not linked. 1: A/D converter start request signal TRG7AN and TGIA6 interrupt skipping 1 are linked.	R/W
4	DT7BE	Down-Count TRG7BN Enable*3 0: A/D converter start requests (TRG7BN) disabled during MTU7.TCNT down-count operation. 1: A/D converter start requests (TRG7BN) enabled during MTU7.TCNT down-count operation.	R/W
5	UT7BE	Up-Count TRG7BN Enable 0: A/D converter start requests (TRG7BN) disabled during MTU7.TCNT up-count operation. 1: A/D converter start requests (TRG7BN) enabled during MTU7.TCNT up-count operation.	R/W
6	DT7AE	Down-Count TRG7AN Enable*3 0: A/D converter start requests (TRG7AN) disabled during MTU7.TCNT down-count operation. 1: A/D converter start requests (TRG7AN) enabled during MTU7.TCNT down-count operation.	R/W

Bit	Symbol	Function	R/W
7	UT7AE	Up-Count TRG7AN Enable 0: A/D converter start requests (TRG7AN) disabled during MTU7.TCNT up-count operation. 1: A/D converter start requests (TRG7AN) enabled during MTU7.TCNT up-count operation.	R/W
13:8	—	These bits are read as 0. The write value should be 0.	R/W
15:14	BF[1:0]	MTU7.TADCOBRA/TADCOBRB Transfer Timing Select These bits specify the transfer timing from MTU7.TADCOBRA and MTU7.TADCOBRB to MTU7.TADCORA and MTU7.TADCORB. For detail, see Table 17.56.	R/W

Note: MTU7.TADCR should be accessed in 16-bit units.
Note 1. When interrupt skipping is disabled (the T6AEN and T7VEN bits in TITCR1B are cleared to 0 or the T6ACOR and T7VCOR bits in TITCR1B are cleared to 0), do not link A/D converter start requests with interrupt skipping function 1 (clear the ITA6AE, ITA7VE, ITB6AE, and ITB7VE bits in MTU7.TADCR to 0).
Note 2. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests are not issued.
Note 3. Do not set to 1 when complementary PWM mode is not selected.

TADCR enables or disables A/D converter start requests and specifies whether to link A/D converter start requests with interrupt skipping function.

Table 17.56 Setting of transfer timing by TADCR.BF[1:0] bits (MTU7)

BF[1]	BF[0]	Description
0	0	Does not transfer data from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB).
0	1	Data is transferred from the cycle set buffer register (MTU7.TADCOBRA, MTU7.TADCOBRB) to the cycle set register (MTU7.TADCORA, MTU7.TADCORB) when the crest of the MTU7.TCNT count is reached or the MTU7.TGRD is written to in complementary PWM mode, when a compare match occurs between MTU6.TCNT and MTU6.TGRA in reset-synchronized PWM mode, or when a compare match occurs between MTU7.TCNT and MTU7.TGRA in PWM mode 1 or normal operation mode.
1	0	Transfers data from the cycle set buffer register to the cycle set register at the trough of the MTU7.TCNT count. These settings are prohibited when complementary PWM mode is not selected.
1	1	Transfers data from the cycle set buffer register to the cycle set register at the crest and trough of the MTU7.TCNT count. These settings are prohibited when complementary PWM mode is not selected.

17.2.38 TADCORx : Timer A/D Converter Start Request Cycle Set Register x (x = A, B)

Base address: MTU4 = 0x9000_1200, MTU7 = 0x9000_1A00

Offset address: 0x044 (TADCORA)
0x046 (TADCORB)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	n/a	TADCORA and TADCORB are 16-bit readable/writable registers that issue a corresponding A/D converter start request when the MTUn.TCNT (n = 4, 7) count reaches the value in TADCORA or TADCORB.	R/W

Note: The TADCORA and TADCORB registers should be accessed in 16-bit units.

When the A/D converter start request delaying function linked with skipping function 1 (for details, see section 17.3.9.4. A/D Converter Start Request Delaying Function Linked with Interrupt Skipping Function 1) is used, the value of this register should be 0x0002 to TCDRA setting − 2 in MTU4 and 0x0002 to TCDRB setting − 2 in MTU7.

When interrupt skipping function 2 is used and the difference between the TADCORA value and the TADCORB value is small, the skipping count may not be counted correctly and the A/D converter start request may not be generated with the expected timing in some cases. The TADCORA and TADCORB values should satisfy the following conditions.

1. When skipping function 2 is specified with the skipping count set to 0
   - The difference between the TADCORA and TADCORB values should be equal to or greater than 4.
   - The TADCORA compare interval should be equal to or greater than 4 PCLKH cycles (the TADCORA update value should be the previous value + 4 or greater, or previous value − 4 or smaller).
   - The TADCORB compare interval should be equal to or greater than 4 PCLKH cycles (the TADCORB update value should be the previous value + 4 or greater, or previous value − 4 or smaller).
2. When skipping function 2 is specified with the skipping count set to 1 or greater
   - The difference between the TADCORA and TADCORB values should be equal to or greater than 2.
   - The TADCORB compare interval should be equal to or greater than 2 PCLKH cycles (the TADCORB update value should be the previous value + 2 or greater, or previous value − 2 or smaller)

17.2.39 TADCOBRx : Timer A/D Converter Start Request Cycle Set Buffer Register x (x = A, B)

Base address: MTU4 = 0x9000_1200, MTU7 = 0x9000_1A00

Offset address: 0x048 (TADCOBRA)
0x04A (TADCOBRB)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	n/a	TADCOBRA and TADCOBRB are 16-bit readable/writable registers whose values are transferred to TADCORA and TADCORB, respectively, when the crest or trough of the MTUn.TCNT count is reached.	R/W

Note: TADCOBRA and TADCOBRB should be accessed in 16-bit units.

17.2.40 TITMRx : Timer Interrupt Skipping Mode Register x (x = A, B)

Base address: MTU = 0x9000_1000

Offset address: 0x23A (TITMRA)
0xA3A (TITMRB)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TITM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TITM	Interrupt Skipping Function Select Selects one of the two types of interrupt skipping functions. 0: Selects interrupt skipping function 1 Setting the TITCR1A or TITCR1B register enables interrupt skipping function 1. 1: Selects interrupt skipping function 2 Setting the TITCR2A or TITCR2B register enables interrupt skipping function 2.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

TITMRA and TITMRB are used to select either of two skipping functions.

17.2.41 TITCR1A : Timer Interrupt Skipping Set Register 1A

Base address: MTU = 0x9000_1000

Offset address: 0x230

Bit position:	7	6	5	4	3	2	1	0
Bit field:	T3AEN	T3ACOR[2:0]			T4VEN	T4VCOR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	T4VCOR[2:0]	TCIV4 Interrupt Skipping Count Setting These bits specify the TCIV4 interrupt skipping count within the range from 0 to 7.*1 0 0 0: Does not skip TCIV4 interrupts. 0 0 1: Sets the TCIV4 interrupt skipping count to 1. 0 1 0: Sets the TCIV4 interrupt skipping count to 2. 0 1 1: Sets the TCIV4 interrupt skipping count to 3. 1 0 0: Sets the TCIV4 interrupt skipping count to 4. 1 0 1: Sets the TCIV4 interrupt skipping count to 5. 1 1 0: Sets the TCIV4 interrupt skipping count to 6. 1 1 1: Sets the TCIV4 interrupt skipping count to 7.	R/W
3	T4VEN	TCIV4 Interrupt Skipping Enable 0: Disabled 1: Enabled	R/W
6:4	T3ACOR[2:0]	TGIA3 Interrupt Skipping Count Setting These bits specify the TGIA3 interrupt skipping count within the range from 0 to 7.*1 0 0 0: Does not skip TGIA3 interrupts. 0 0 1: Sets the TGIA3 interrupt skipping count to 1. 0 1 0: Sets the TGIA3 interrupt skipping count to 2. 0 1 1: Sets the TGIA3 interrupt skipping count to 3. 1 0 0: Sets the TGIA3 interrupt skipping count to 4. 1 0 1: Sets the TGIA3 interrupt skipping count to 5. 1 1 0: Sets the TGIA3 interrupt skipping count to 6. 1 1 1: Sets the TGIA3 interrupt skipping count to 7.	R/W
7	T3AEN	TGIA3 Interrupt Skipping Enable 0: Disabled 1: Enabled	R/W

Note 1. When 0 is specified for the interrupt skipping count, no interrupt skipping will be performed. Before changing the interrupt skipping count, be sure to set the TITCR1A.T3AEN and TITCR1A.T4VEN bits to 0 to clear the skipping counter (TITCNT1A).

The TITCR1A register enables or disables interrupt skipping and specifies the interrupt skipping count. This setting is valid only while the TITMRA.TITM bit is set to 0; when the TITMRA.TITM bit is set to 1, the setting in the TITCR1A register is cleared.

17.2.42 TITCR1B : Timer Interrupt Skipping Set Register 1B

Base address: MTU = 0x9000_1000

Offset address: 0xA30

Bit position:	7	6	5	4	3	2	1	0
Bit field:	T6AEN	T6ACOR[2:0]			T7VEN	T7VCOR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	T7VCOR[2:0]	TCIV7 Interrupt Skipping Count Setting These bits specify the TCIV7 interrupt skipping count within the range from 0 to 7.*1 0 0 0: Does not skip TCIV7 interrupts. 0 0 1: Sets the TCIV7 interrupt skipping count to 1. 0 1 0: Sets the TCIV7 interrupt skipping count to 2. 0 1 1: Sets the TCIV7 interrupt skipping count to 3. 1 0 0: Sets the TCIV7 interrupt skipping count to 4. 1 0 1: Sets the TCIV7 interrupt skipping count to 5. 1 1 0: Sets the TCIV7 interrupt skipping count to 6. 1 1 1: Sets the TCIV7 interrupt skipping count to 7.	R/W
3	T7VEN	TCIV7 Interrupt Skipping Enable 0: Disabled 1: Enabled	R/W
6:4	T6ACOR[2:0]	TGIA6 Interrupt Skipping Count Setting These bits specify the TGIA6 interrupt skipping count within the range from 0 to 7.*1 0 0 0: Does not skip TGIA6 interrupts. 0 0 1: Sets the TGIA6 interrupt skipping count to 1. 0 1 0: Sets the TGIA6 interrupt skipping count to 2. 0 1 1: Sets the TGIA6 interrupt skipping count to 3. 1 0 0: Sets the TGIA6 interrupt skipping count to 4. 1 0 1: Sets the TGIA6 interrupt skipping count to 5. 1 1 0: Sets the TGIA6 interrupt skipping count to 6. 1 1 1: Sets the TGIA6 interrupt skipping count to 7.	R/W
7	T6AEN	TGIA6 Interrupt Skipping Enable 0: Disabled 1: Enabled	R/W

Note 1. When 0 is specified for the interrupt skipping count, no interrupt skipping will be performed.
Before changing the interrupt skipping count, be sure to set the TITCR1B.T6AEN and TITCR1B.T7VEN bits to 0 to clear the skipping counter (TITCNT1B).

The TITCR1B register enables or disables interrupt skipping and specifies the interrupt skipping count. This setting is valid only while the TITMRB.TITM bit is set to 0; when the TITMRB.TITM bit is set to 1, the setting in the TITCR1B register is cleared.

17.2.43 TITCNT1A : Timer Interrupt Skipping Counter 1A

Base address: MTU = 0x9000_1000

Offset address: 0x231

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	T3ACNT[2:0]			—	T4VCNT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	T4VCNT[2:0]	TCIV4 Interrupt Counter While the T4VEN bit in TITCR1A is set to 1, the count in these bits is incremented every time a TCIV4 interrupt occurs.	R
3	—	This bit is read as 0.	R
6:4	T3ACNT[2:0]	TGIA3 Interrupt Counter While the T3AEN bit in TITCR1A is set to 1, the count in these bits is incremented every time a TGIA3 interrupt occurs.	R
7	—	This bit is read as 0.	R

Note: To clear the TITCNT1A, set the TITCR1A.T3AEN and TITCR1A.T4VEN bits to 0.

TITCNT1A is 8-bit readable/writable counters. TITCNT1A retains their values even after stopping the count operation of MTU3.TCNT and MTU4.TCNT.

T4VCNT[2:0] bits (TCIV4 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRA is 1
- When the T4VEN bit in TITCR1A is set to 0
- When the T4VCOR[2:0] bits in TITCR1A are set to 000b
- When the T4VCNT[2:0] bits in TITCNT1A match the T4VCOR[2:0] bits in TITCR1A

T3ACNT[2:0] bits (TGIA3 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRA is 1
- When the T3AEN bit in TITCR1A is set to 0
- When the T3ACOR[2:0] bits in TITCR1A are set to 000b
- When the T3ACNT[2:0] bits in TITCNT1A match the T3ACOR[2:0] bits in TITCR1A

17.2.44 TITCNT1B : Timer Interrupt Skipping Counter 1B

Base address: MTU = 0x9000_1000

Offset address: 0xA31

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	T6ACNT[2:0]			—	T7VCNT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	T7VCNT[2:0]	TCIV7 Interrupt Counter While the T7VEN bit in TITCR1B is set to 1, the count in these bits is incremented every time a TCIV7 interrupt occurs.	R
3	—	This bit is read as 0.	R
6:4	T6ACNT[2:0]	TGIA6 Interrupt Counter While the T6AEN bit in TITCR1B is set to 1, the count in these bits is incremented every time a TGIA6 interrupt occurs.	R
7	—	This bit is read as 0.	R

Note: To clear the TITCNT1B, set the TITCR1B.T6AEN and TITCR1B.T7VEN bits to 0.

TITCNT1B is 8-bit readable/writable counters. TITCNT1B retains their values even after stopping the count operation of MTU6.TCNT and MTU7.TCNT.

T7VCNT[2:0] bits (TCIV7 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRB is 1
- When the T7VEN bit in TITCR1B is set to 0
- When the T7VCOR[2:0] bits in TITCR1B are set to 000b
- When the T7VCNT[2:0] bits in TITCNT1B match the T7VCOR[2:0] bits in TITCR1B

T6ACNT[2:0] bits (TGIA6 Interrupt Counter)

[Clearing conditions]

- When the TITM bit in TITMRB is 1
- When the T6AEN bit in TITCR1B is set to 0
- When the T6ACOR[2:0] bits in TITCR1B are set to 000b

- When the T6ACNT[2:0] bits in TITCNT1B match the T6ACOR[2:0] bits in TITCR1B

17.2.45 TITCR2A : Timer Interrupt Skipping Set Register 2A

Base address: MTU = 0x9000_1000

Offset address: 0x23B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TRG4COR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TRG4COR[2:0]	TRG4AN/TRG4BN Interrupt Skipping Count Setting These bits specify the TRG4AN/TRG4BN interrupt skipping count within the range from 0 to 7. 0 0 0: Does not skip TRG4AN and TRG4BN interrupts. 0 0 1: Sets the TRG4AN and TRG4BN interrupt skipping count to 1. 0 1 0: Sets the TRG4AN and TRG4BN interrupt skipping count to 2. 0 1 1: Sets the TRG4AN and TRG4BN interrupt skipping count to 3. 1 0 0: Sets the TRG4AN and TRG4BN interrupt skipping count to 4. 1 0 1: Sets the TRG4AN and TRG4BN interrupt skipping count to 5. 1 1 0: Sets the TRG4AN and TRG4BN interrupt skipping count to 6. 1 1 1: Sets the TRG4AN and TRG4BN interrupt skipping count to 7.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

TITCR2A specifies the interrupt skipping count for TRG4AN and TRG4BN.

This setting is valid only while TITMRA is set to 1.

17.2.46 TITCR2B : Timer Interrupt Skipping Set Register 2B

Base address: MTU = 0x9000_1000

Offset address: 0xA3B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TRG7COR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TRG7COR[2:0]	TRG7AN/TRG7BN Interrupt Skipping Count Setting These bits specify the TRG7AN/TRG7BN interrupt skipping count within the range from 0 to 7. 0 0 0: Does not skip TRG7AN and TRG7BN interrupts. 0 0 1: Sets the TRG7AN and TRG7BN interrupt skipping count to 1. 0 1 0: Sets the TRG7AN and TRG7BN interrupt skipping count to 2. 0 1 1: Sets the TRG7AN and TRG7BN interrupt skipping count to 3. 1 0 0: Sets the TRG7AN and TRG7BN interrupt skipping count to 4. 1 0 1: Sets the TRG7AN and TRG7BN interrupt skipping count to 5. 1 1 0: Sets the TRG7AN and TRG7BN interrupt skipping count to 6. 1 1 1: Sets the TRG7AN and TRG7BN interrupt skipping count to 7.	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

TITCR2B specifies the interrupt skipping count for TRG7AN and TRG7BN.

This setting is valid only while TITMRB is set to 1.

17.2.47 TITCNT2A : Timer Interrupt Skipping Counter 2A

Base address: MTU = 0x9000_1000

Offset address: 0x23C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TRG4CNT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TRG4CNT[2:0]	TRG4AN/TRG4BN Interrupt Counter These bits start counting from the value set in TRG4COR[2:0] and the count decrements every time TRG4AN or TRG4BN is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid.	R
7:3	—	These bits are read as 0.	R

TITCNT2A starts counting from the values set in the TRG4COR[2:0] bits and the count decrements every time TRG4AN or TRG4BN is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid.

TRG4CNT[2:0] bits (TRG4AN/TRG4BN Interrupt Counter)

These bits start counting from the value set in the TRG4COR[2:0] bits and the count decrements every time a TRG4AN or TRG4BN interrupt is generated. When the count reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid.

[Clearing conditions]

- When the TITM bit in TITMRA is 0
- When the TRG4COR[2:0] bits in TITCR2A are set to 000b
- When the count of TRG4AN and TRG4BN occurrence matches the TRG4COR[2:0] value in TITCR2A

17.2.48 TITCNT2B : Timer Interrupt Skipping Counter 2B

Base address: MTU = 0x9000_1000

Offset address: 0xA3C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TRG7CNT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TRG7CNT[2:0]	TRG7AN/TRG7BN Interrupt Counter These bits start counting from the value set in TRG7COR[2:0] and the count decrements every time TRG7AN or TRG7BN is generated. When the count reaches 0 and is reloaded, the TRG7AN and TRG7BN interrupts become valid.	R
7:3	—	These bits are read as 0.	R

TITCNT2B starts counting from the values set in the TRG7COR[2:0] bits and the count decrements every time TRG7AN or TRG7BN is generated. When the count reaches 0 and is reloaded, the TRG7AN and TRG7BN interrupts become valid.

TRG7CNT[2:0] bits (TRG7AN/TRG7BN Interrupt Counter)

These bits start counting from the value set in the TRG7COR[2:0] bits and the count decrements every time a TRG7AN or TRG7BN interrupt is generated. When the count reaches 0 and is reloaded, the TRG7AN and TRG7BN interrupts become valid.

[Clearing conditions]

- When the TITM bit in TITMRB is 0

- When the TRG7COR[2:0] bits in TITCR2B are set to 000b
- When the count of TRG7AN and TRG7BN occurrence matches the TRG7COR[2:0] value in TITCR2B

17.2.49 Bus Master Interface

The timer counter (MTU8.TCNT), general registers (MTU8.TGRn) for MTU8, and MTU1.TCNTLW, MTU1.TGRALW, and MTU1.TGRBLW registers when TMDR3.LWA = 1 are 32-bit registers. A 32-bit data bus to the bus master enables 32-bit read/write access. 8- and 16-bit read/write are not allowed. Always access these registers in 32-bit units.

The timer counters (MTU0.TCNT to MTU7.TCNT) excluding MTU8, general registers (MTU0.TGRn to MTU7.TGRn), timer subcounters (TCNTSA and TCNTSB), timer cycle buffer registers (TCBRA and TCBRB), timer dead time data registers (TDDRA and TDDRB), timer cycle data registers (TCDRA and TCDRB), timer A/D converter start request control registers (MTU4.TADCR and MTU7.TADCR), timer A/D converter start request cycle set registers (MTU4.TADCORA, MTU4.TADCORB, MTU7.TADCORA, and MTU7.TADCORB), and timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA, MTU4.TADCOBRB, MTU7.TADCOBRA, and MTU7.TADCOBRB) are 16-bit registers. A 16-bit data bus to the bus master enables 16-bit read/write access. 8-bit read/write is not allowed. Always access the registers in 16-bit units.

All registers other than the above registers are 8-bit registers. A 16-bit data bus to the CPU enables 16-bit read/write access. These registers can also be accessed in 8-bit units.

17.3 Operations

17.3.1 Basic Functions

Each channel has TCNT and TGR. TCNT performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting.

Each TGR can be used as an input capture register or an output compare register.

17.3.1.1 Counter Operation

When one of bits CST0 to CST4 and CST8 in the TSTRA register, bits CST6 and CST7 in the TSTRB register, and bits CSTU5, CSTV5, and CSTW5 in the MTU5.TSTR register is set to 1, TCNT for the corresponding channel begins counting. TCNT can operate as a free-running counter, periodic counter, for example.

(1) Example of Count Operation Setting Procedure

Figure 17.5 shows an example of the count operation setting procedure.

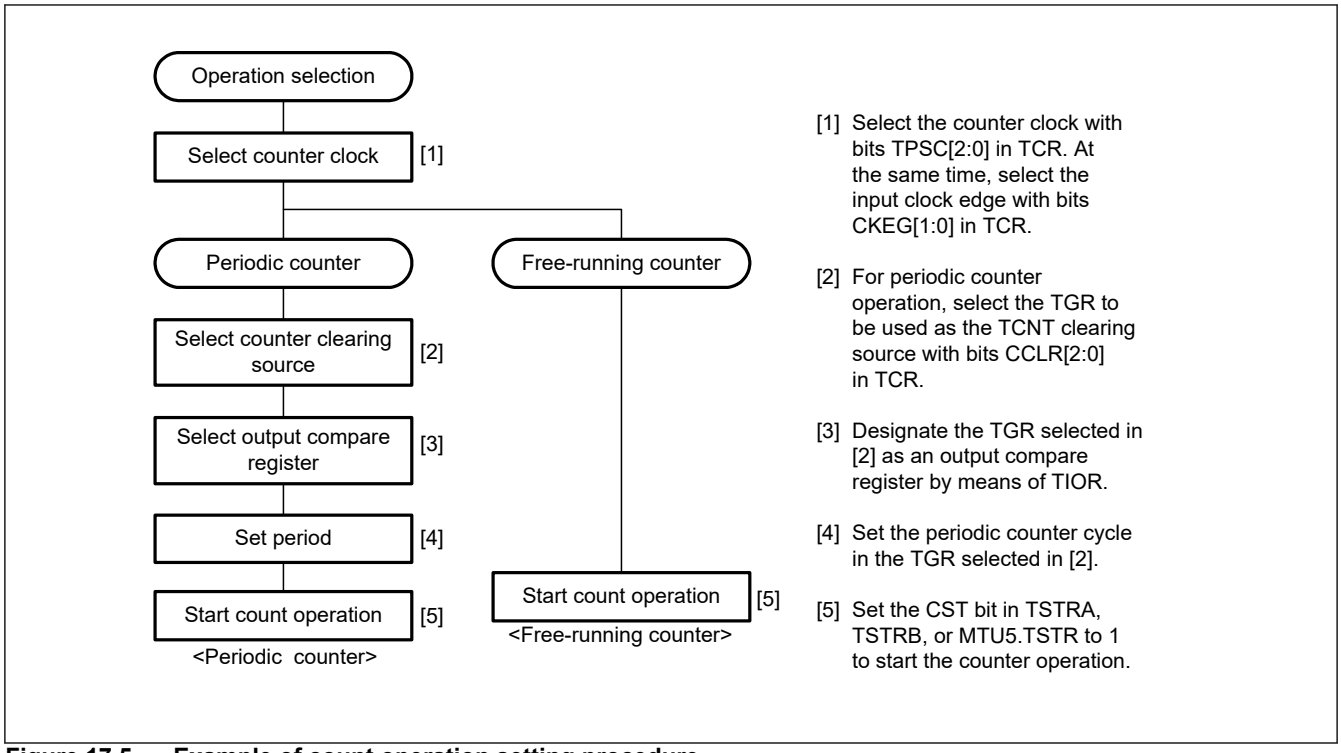


Figure 17.5 Example of count operation setting procedure

(2) Free-Running Count Operation and Periodic Count Operation

Immediately after a reset, the MTU's TCNT counters are all designated as free-running counters. When the relevant bit in TSTRA, TSTRB, or MTU5.TSTR is set to 1, the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from 0xFFFF to 0x0000), if the corresponding TIER.TCIEV bit is 1, the MTU requests an interrupt. After an overflow, TCNT starts counting up again from 0x0000.

Figure 17.6 shows free-running counter operation.

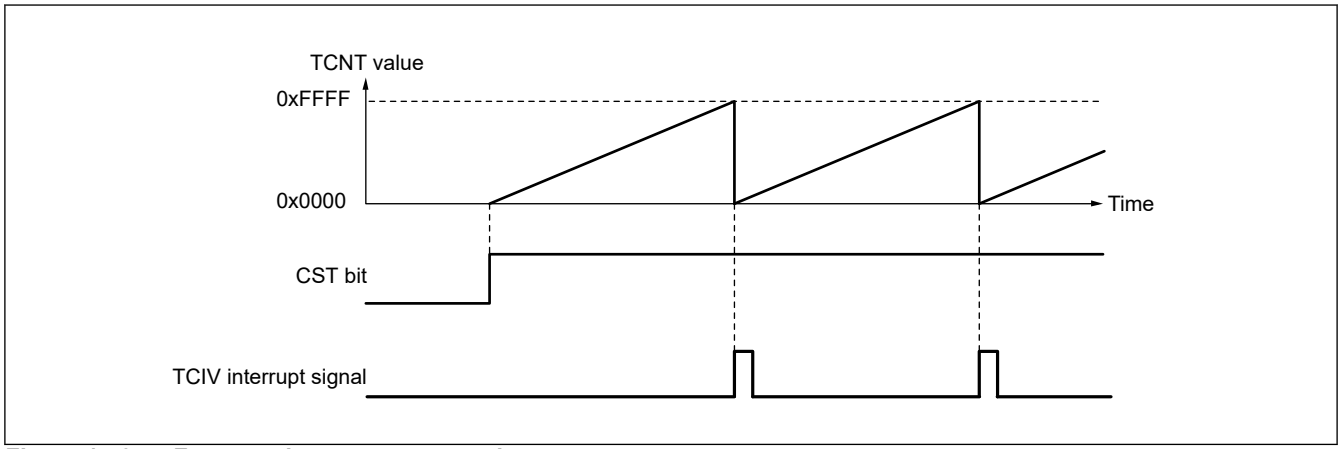


Figure 17.6 Free-running counter operation

When compare match is selected as the TCNT clearing source, TCNT for the relevant channel performs periodic count operation. TGR for setting the cycle is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR[2:0] in TCR. After the settings have been made, TCNT starts up-count operation as a periodic counter when the corresponding bit in TSTRA, TSTRB or MTU5.TSTR is set to 1. When the count matches the value in TGR, TCNT is cleared to 0x0000.

If the value of the corresponding TIER.TGIE bit is 1 at this point, the MTU requests an interrupt. After a compare match, TCNT starts counting up again from 0x0000.

Figure 17.7 shows periodic counter operation.

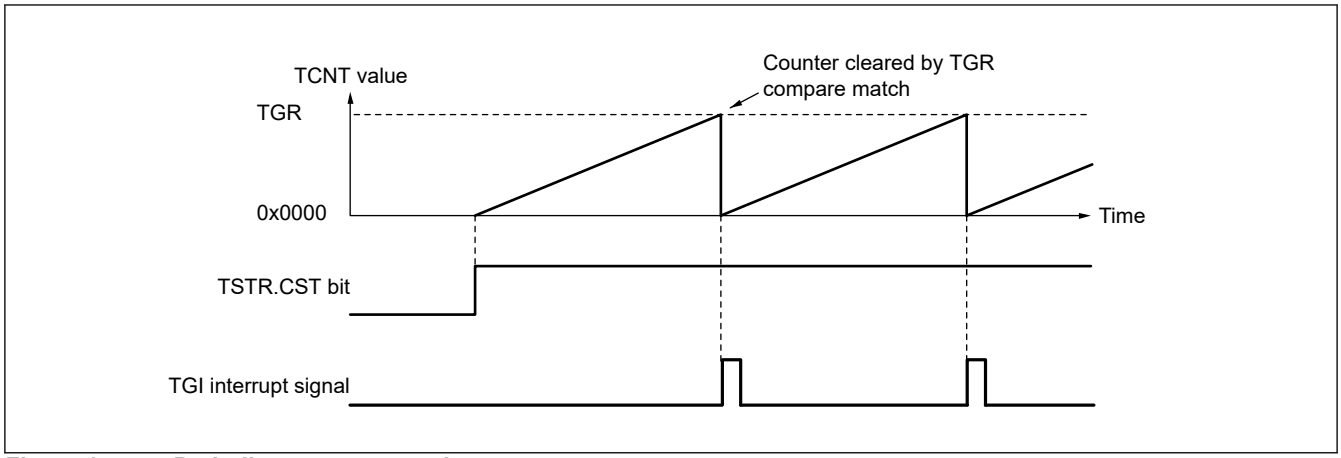


Figure 17.7 Periodic counter operation

17.3.1.2 Waveform Output by Compare Match

The MTU can output low or high or toggles output from the corresponding output pin using compare match. Note that MTU5 cannot output compare-match signals.

(1) Example of Procedure for Setting Waveform Output by Compare Match

Figure 17.8 shows an example of the procedure for setting waveform output by compare match.

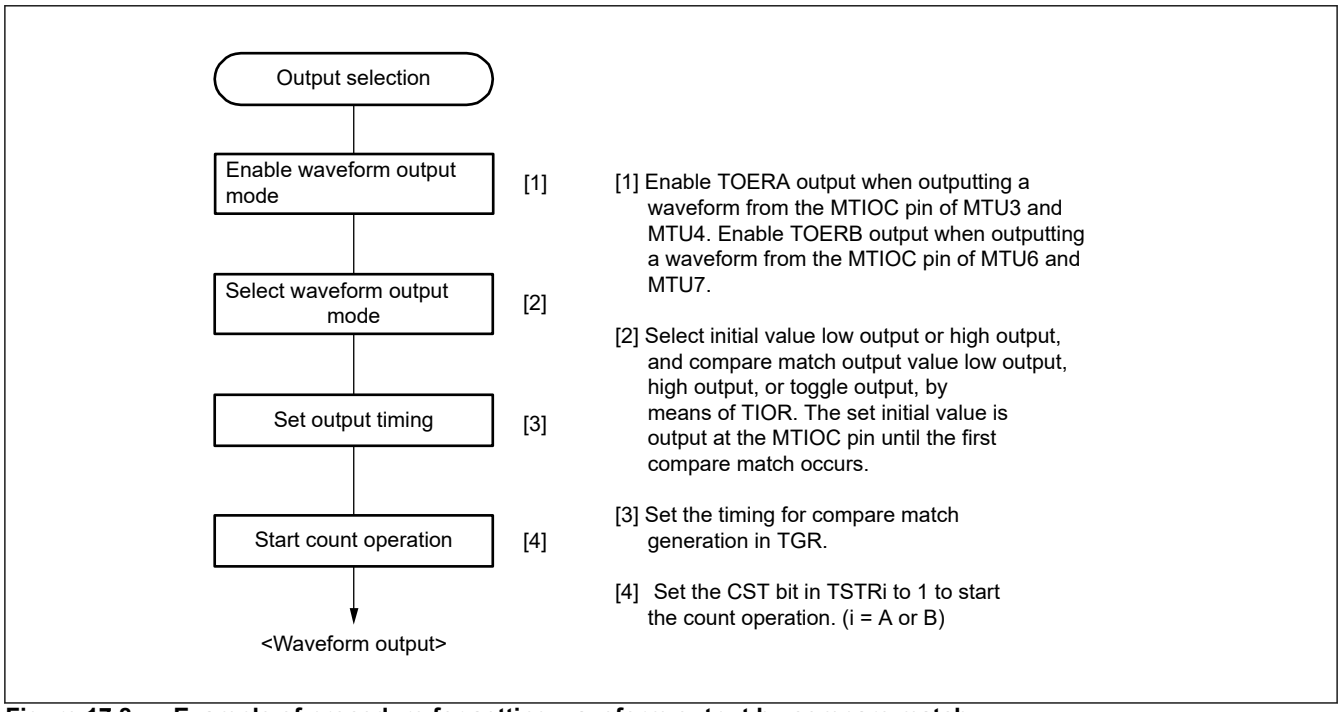


Figure 17.8 Example of procedure for setting waveform output by compare match

(2) Examples of Waveform Output Operation

Figure 17.9 shows an example of low output and high output.

In this example, TCNT has been designated as a free-running counter, and settings have been made so that high is output by compare match A and low is output by compare match B. When the pin level is the same as the specified level, the pin level does not change.

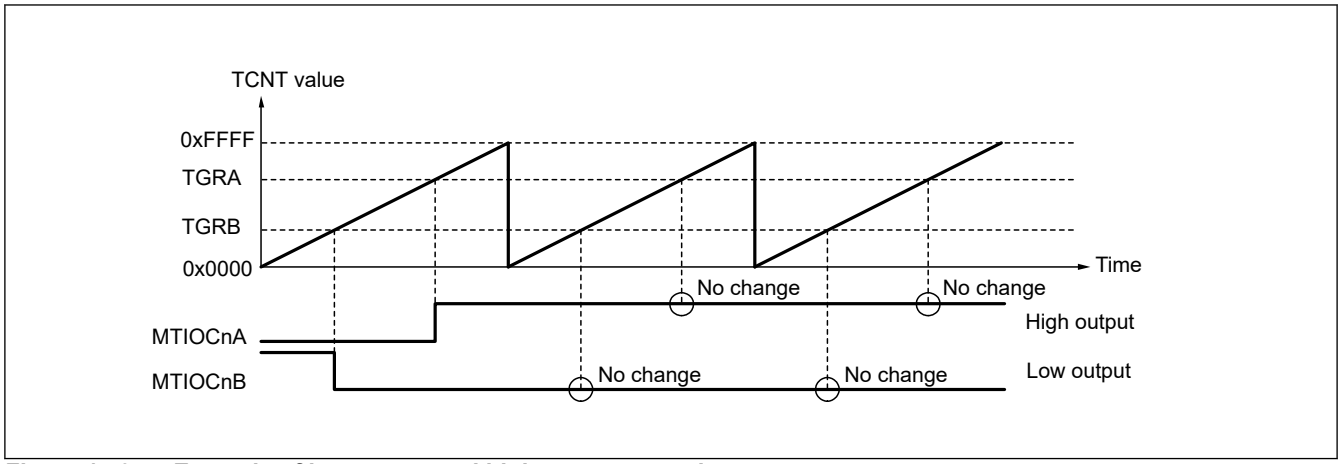


Figure 17.9 Example of low output and high output operation

Figure 17.10 shows an example of toggle output.

In this example, TCNT has been designated as a periodic counter (with counter clearing on compare match B), and settings have been made so that the output is toggled by both compare match A and compare match B.

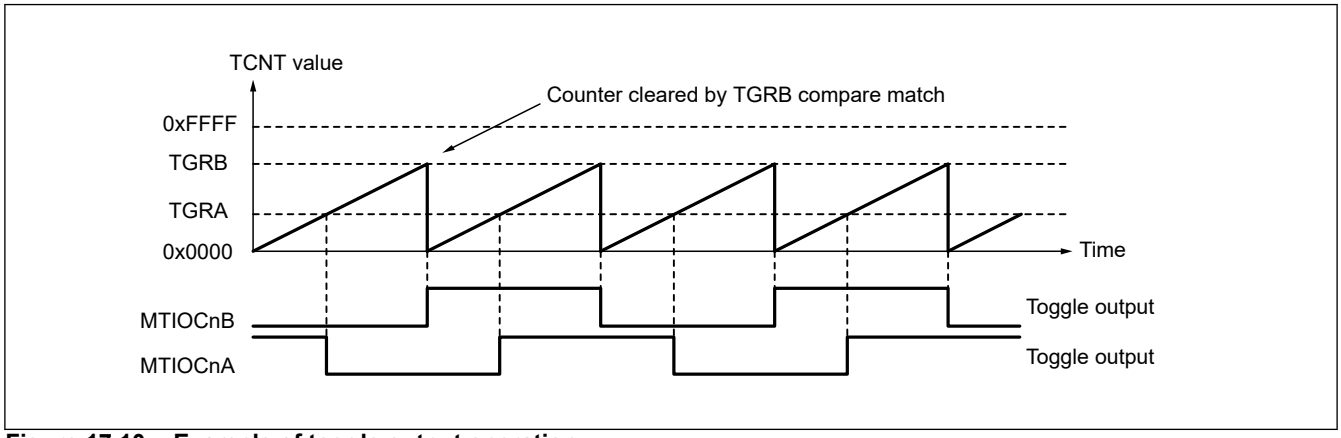


Figure 17.10 Example of toggle output operation

17.3.1.3 Input Capture Function

The TCNT value can be transferred to TGR on detection of an edge input on the MTIOCnm (n = 0 to 4, 6, 7, or 8, m = A to D), MTIC5U, MTIC5V, or MTIC5W pin.

The rising edge, falling edge, or both edges can be selected as the detection edge. For MTU0 and MTU1, a count clock of another channel or compare match signal can also be specified as the input capture source.

Note: When a count input clock of another channel is used as the input capture input for MTU0 and MTU1, PCLKH/1 should not be selected as the count input clock used for input capture input. Input capture is not generated if PCLKH/1 is selected.

(1) Example of Input Capture Operation Setting Procedure

Figure 17.11 shows an example of the input capture operation setting procedure.

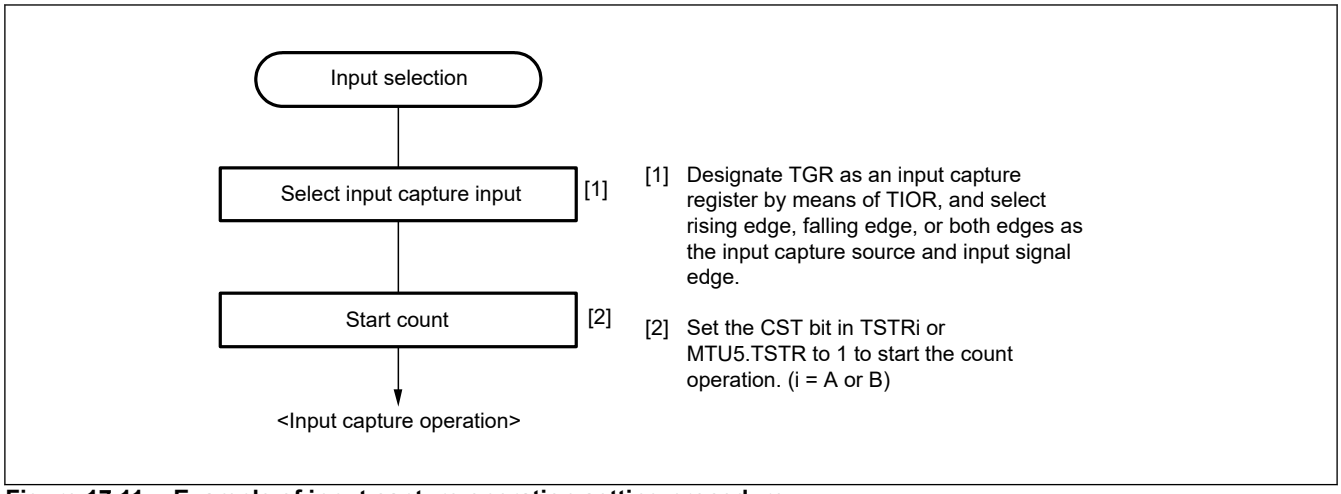


Figure 17.11 Example of input capture operation setting procedure

(2) Example of Input Capture Operation

Figure 17.12 shows an example of input capture operation.

In this example, both rising and falling edges have been selected as the MTIOCnA pin input capture input edge, the falling edge has been selected as the MTIOCnB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT. (n = 0 to 4, 6, 7, 8)

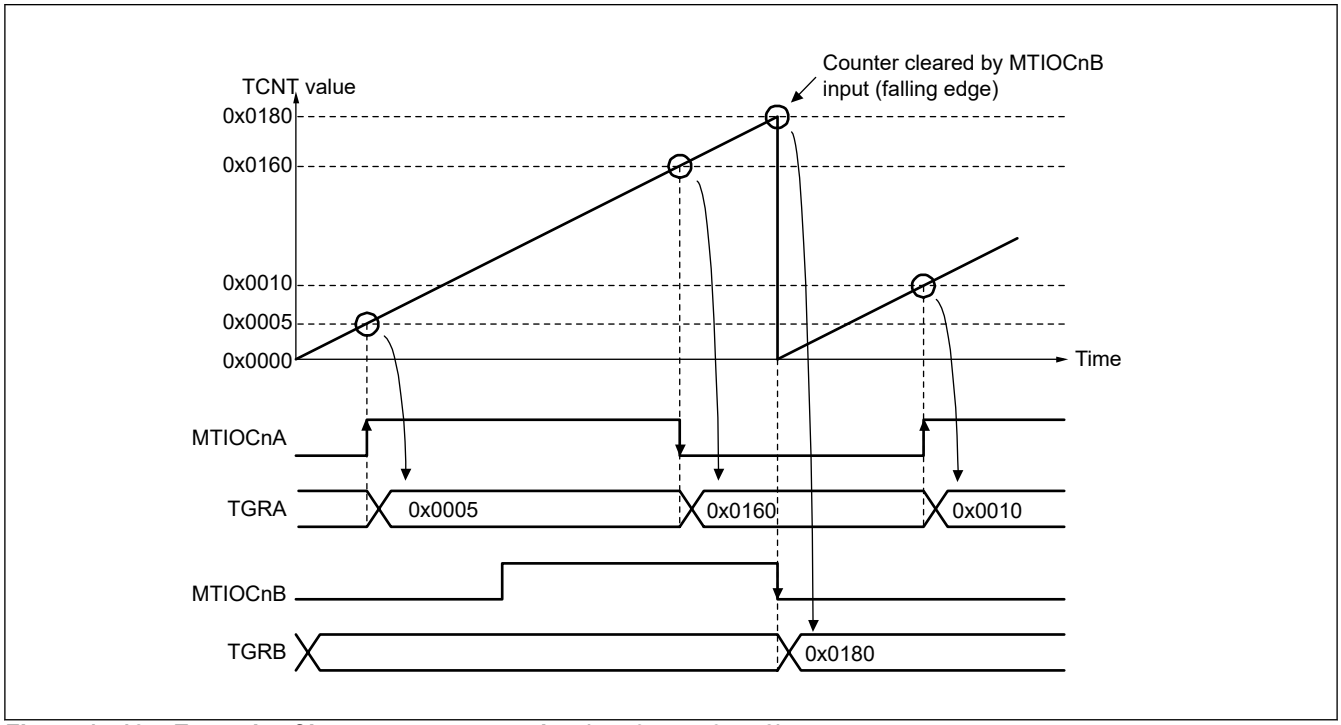


Figure 17.12 Example of input capture operation (n = 0 to 4, 6, 7, 8)

17.3.2 Synchronous Operation

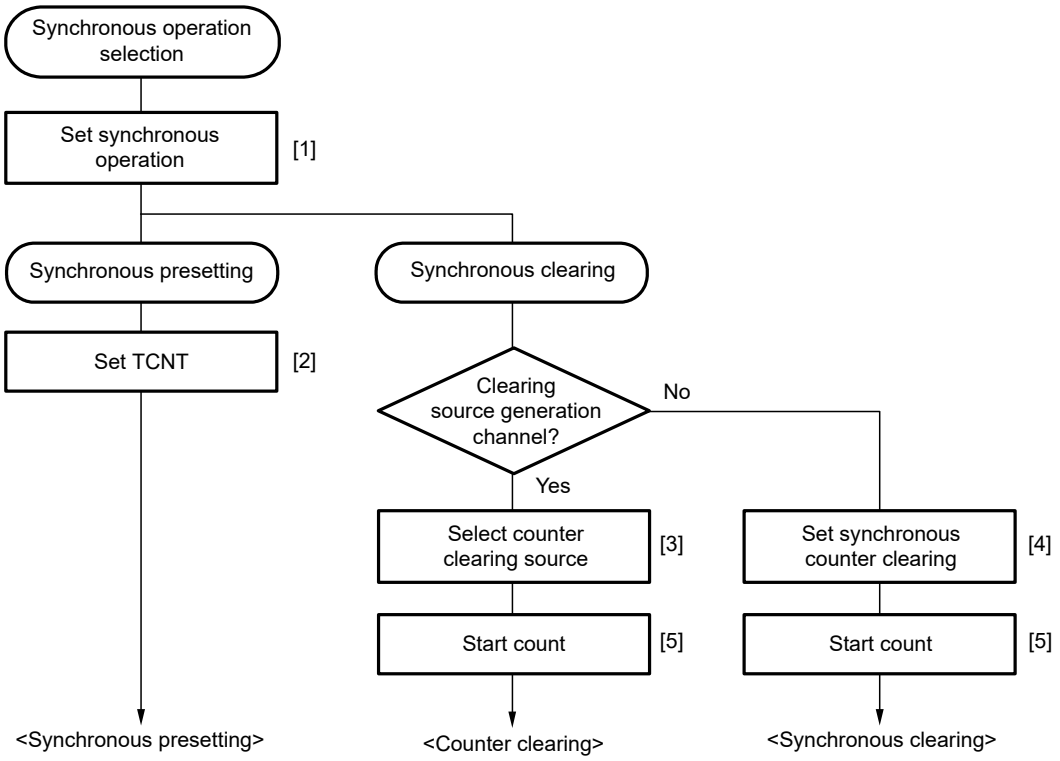
In synchronous operation, the values in multiple TCNT counters can be modified simultaneously (synchronous presetting). In addition, multiple TCNT counters can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TCR.

Synchronous operation increases the number of TGR registers assigned to a single time base.

MTU0 to MTU4, MTU6, and MTU7 can all be designated for synchronous operation. MTU5 and MTU8 cannot be used for synchronous operation.

17.3.2.1 Example of Synchronous Operation Setting Procedure

Figure 17.13 shows an example of the synchronous operation setting procedure.



[1] Set to 1 the SYNC bits in TSYRj corresponding to the channels to be designated for synchronous operation. (j = A or B)
[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.
[3] Use bits CCLR[2:0] in TCR to specify TCNT clearing by input capture/output compare, etc.
[4] Use bits CCLR[2:0] in TCR to designate synchronous clearing for the counter clearing source.
[5] Set to 1 the CST bits in TSTRj for the relevant channels, to start the count operation. (j = A or B)

Figure 17.13 Example of synchronous operation setting procedure

17.3.2.2 Example of Synchronous Operation

Figure 17.14 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for MTU0 to MTU2, MTU0.TGRB compare match has been set as the counter clearing source in MTU0, and synchronous clearing has been set for the counter clearing source in MTU1 and MTU2.

Three-phase PWM waveforms are output from pins MTIOC0A, MTIOC1A, and MTIOC2A. At this time, synchronous presetting and synchronous clearing by MTU0.TGRB compare match are performed for the TCNT counters in MTU0 to MTU2, and the data set in MTU0.TGRB is used as the PWM cycle.

For details of PWM modes, see section 17.3.5. PWM Modes.

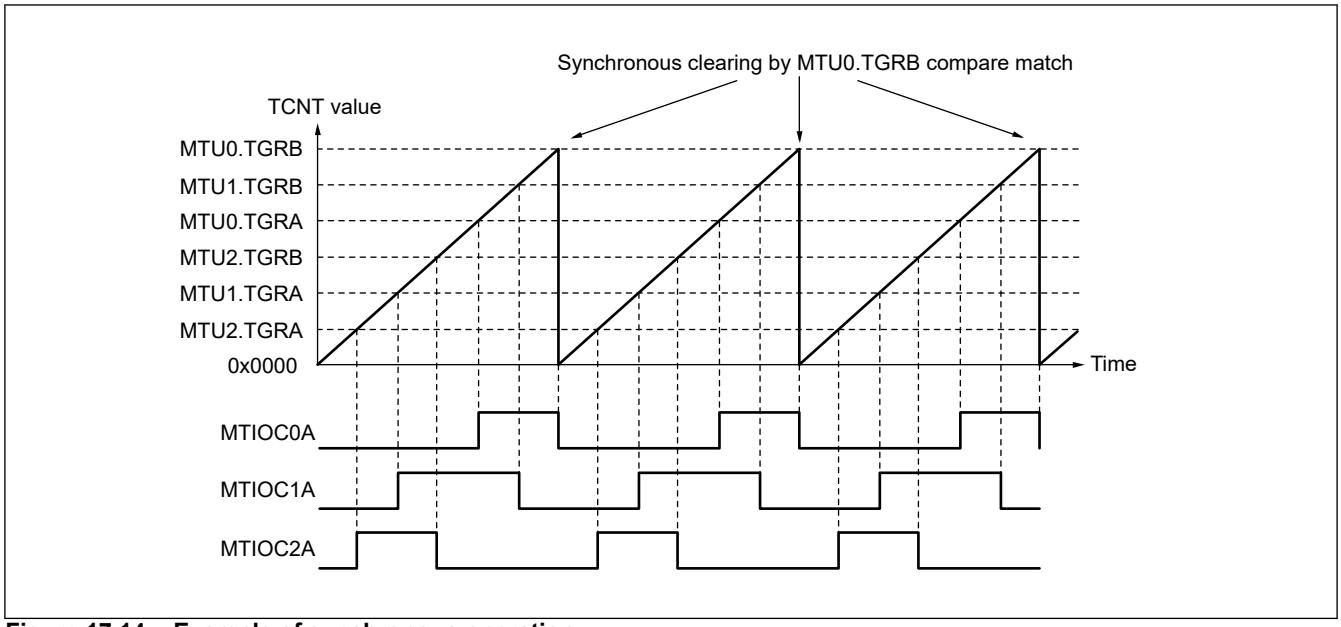


Figure 17.14 Example of synchronous operation

17.3.3 Buffer Operation

Buffer operation, provided for MTU0, MTU3, MTU4, MTU6, MTU7, and MTU8, enables TGRC and TGRD to be used as buffer registers. In MTU0, TGRF can also be used as a buffer register.

Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register.

Note: MTU0.TGRE cannot be designated as an input capture register and can only operate as a compare match register.

Table 17.57 shows the register combinations used in buffer operation.

Table 17.57 Register combinations in buffer operation

Channel	Timer general register	Buffer register
MTU0	TGRA	TGRC
	TGRB	TGRD
	TGRE	TGRF
MTU3	TGRA	TGRC
	TGRB	TGRD
MTU4	TGRA	TGRC
	TGRB	TGRD
MTU6	TGRA	TGRC
	TGRB	TGRD
MTU7	TGRA	TGRC
	TGRB	TGRD
MTU8	TGRA	TGRC
	TGRB	TGRD

(1) When TGR is an output compare register

When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.

This operation is illustrated in Figure 17.15.

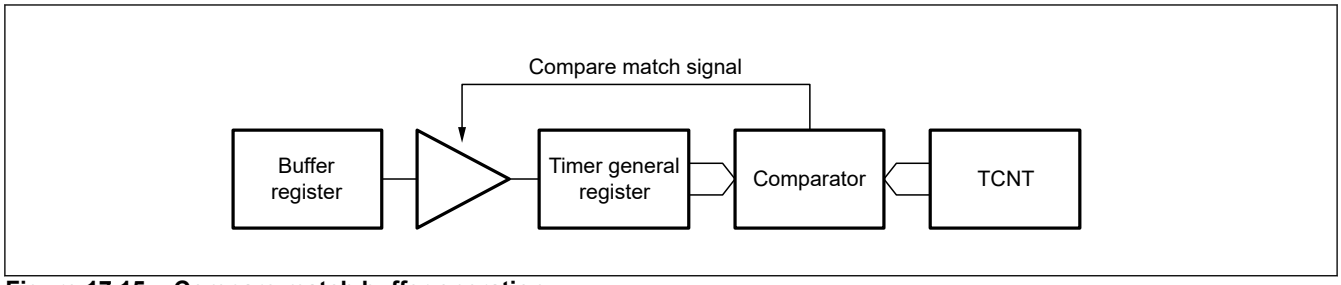


Figure 17.15 Compare match buffer operation

(2) When TGR is an input capture register

When an input capture occurs, the value in TCNT is transferred to TGR and the value previously held in TGR is transferred to the buffer register.

This operation is illustrated in Figure 17.16.

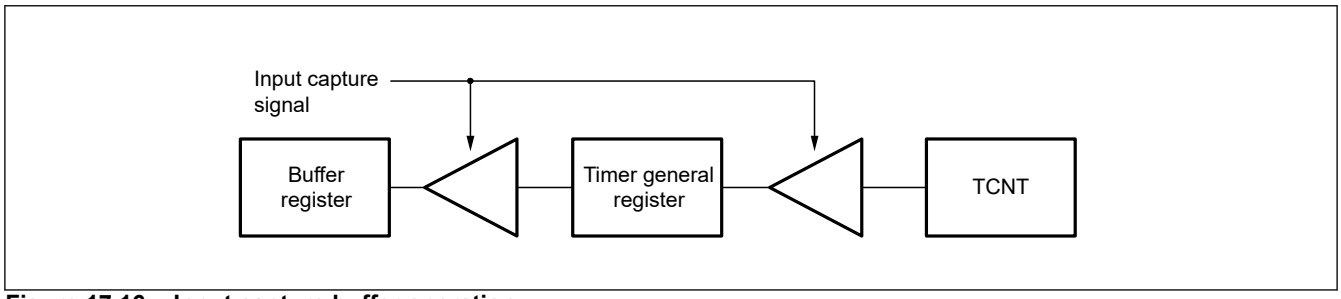


Figure 17.16 Input capture buffer operation

17.3.3.1 Example of Buffer Operation Setting Procedure

Figure 17.17 shows an example of the buffer operation setting procedure.

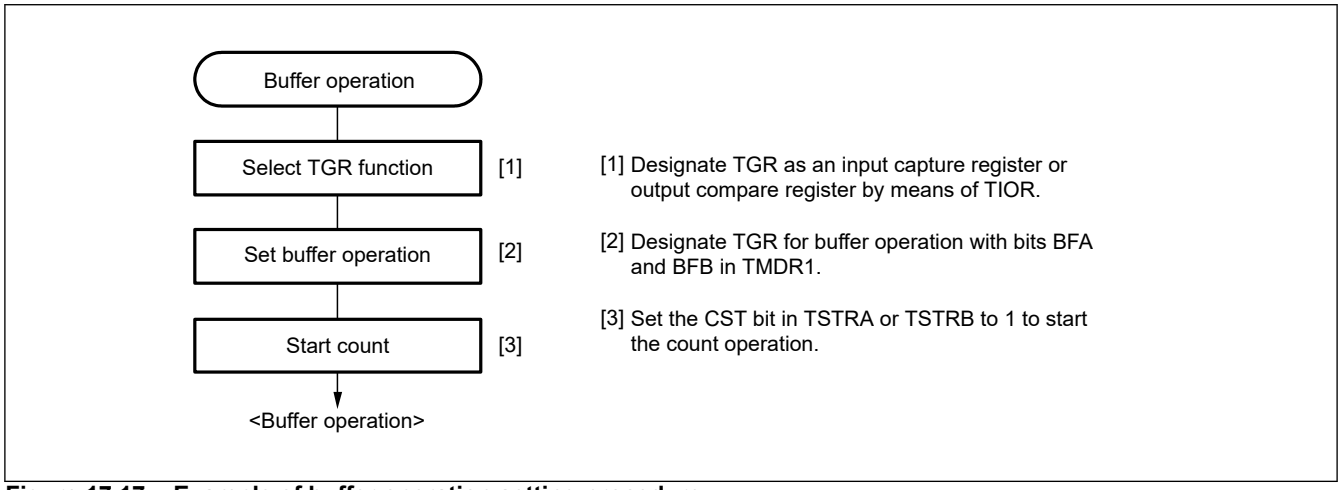


Figure 17.17 Example of buffer operation setting procedure

17.3.3.2 Examples of Buffer Operation

(1) When TGR is an Output Compare Register

Figure 17.18 shows an operation example in which PWM mode 1 has been designated for MTU0, and TGRC has been designated as a buffer register for TGRA. The settings used in this example are TCNT clearing by compare match B, high output at compare match A, and low output at compare match B. In this example, the TTSA bit in TBTM is set to 0.

As buffer operation has been set, when compare match A occurs, the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs.

For details of PWM modes, see section 17.3.5. PWM Modes.

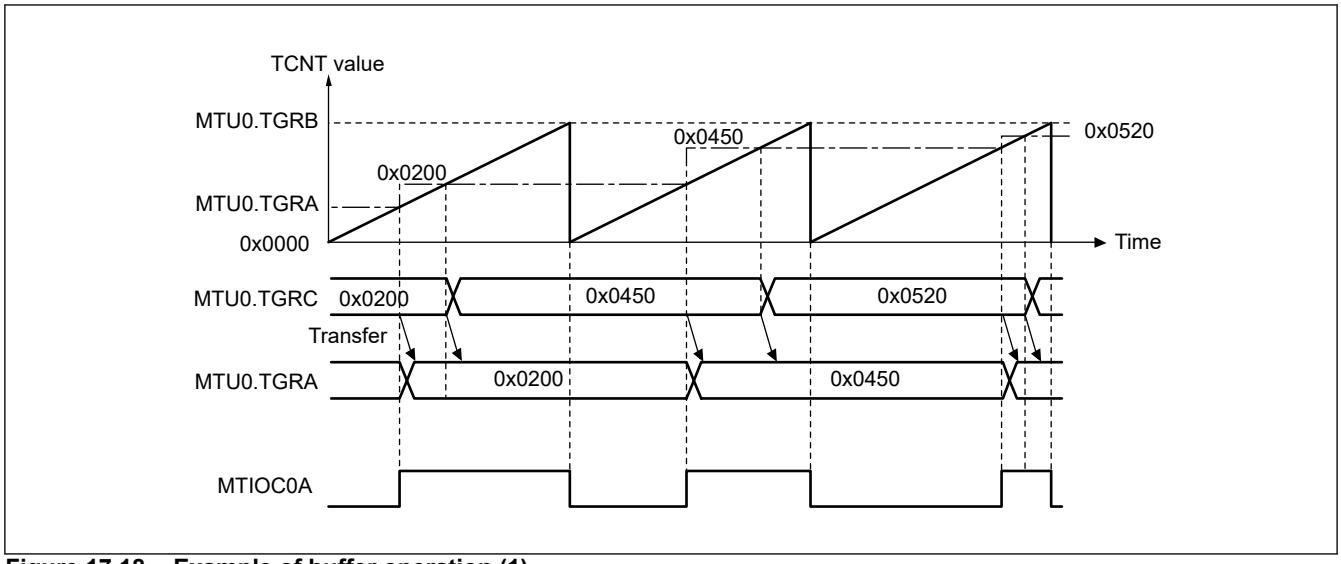


Figure 17.18 Example of buffer operation (1)

(2) When TGR is an Input Capture Register

Figure 17.19 shows an operation example in which TGRA has been designated as an input capture register, and TGRC has been designated as a buffer register for TGRA.

Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the MTIOCnA pin input capture input edge. (n = 0 to 4, 6, 7, 8)

As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

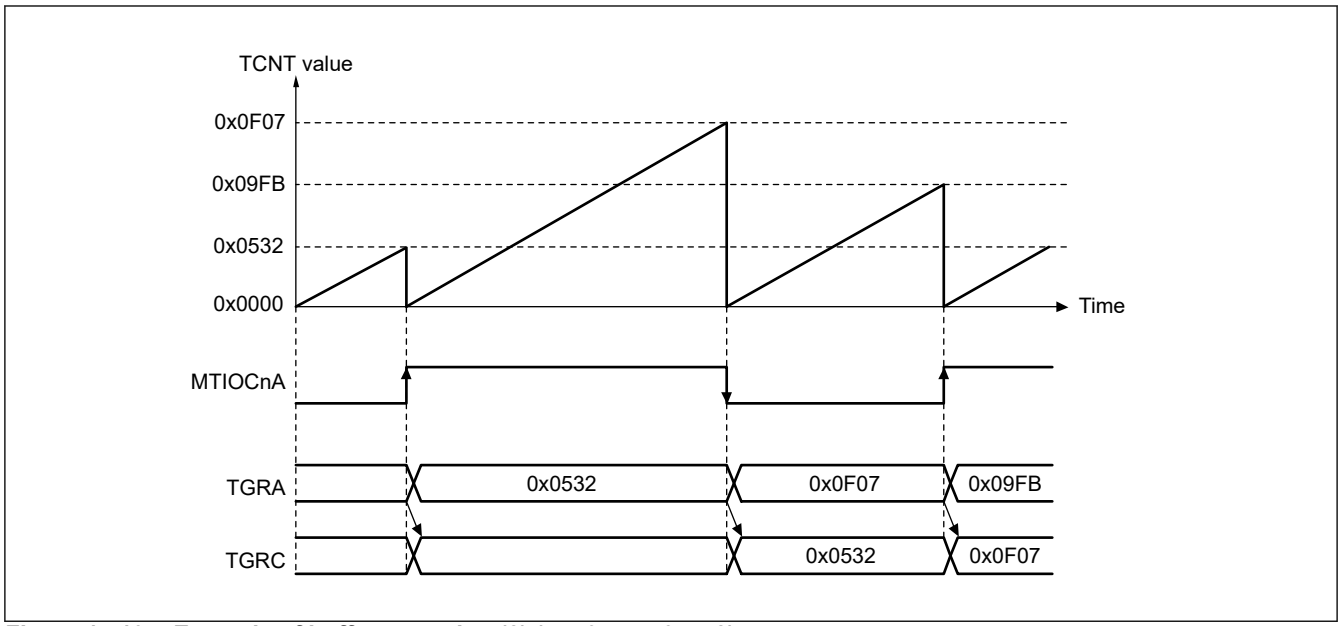


Figure 17.19 Example of buffer operation (2) (n = 0 to 4, 6, 7, 8)

17.3.3.3 Selecting Timing for Transfer from Buffer Registers to Timer General Registers in PWM Mode

The timing for transfer from buffer registers to timer general registers can be selected in PWM mode 1 or 2 for MTU0 or in PWM mode 1 for MTU3, MTU4, MTU6, and MTU7 by setting the buffer operation transfer mode registers (MTUn.TBTM (n = 0, 3, 4, 6, 7)). Either compare match (value after reset) or TCNT clearing can be selected for the transfer timing. TCNT clearing as transfer timing is one of the following cases.

- When TCNT overflows (0xFFFF to 0x0000)

- When 0x0000 is written to TCNT during counting
- When TCNT is set to 0x0000 under the condition specified in the CCLR[2:0] bits in TCR

Note: TBTM must be modified only while TCNT stops.

Figure 17.20 shows an operation example in which PWM mode 1 is designated for MTU0 and buffer operation is designated for MTU0.TGRA and MTU0.TGRC. The settings used in this example are MTU0.TCNT clearing by compare match B, high output at compare match A, and low output at compare match B. The TTSA bit in MTU0.TBTM is set to 1.

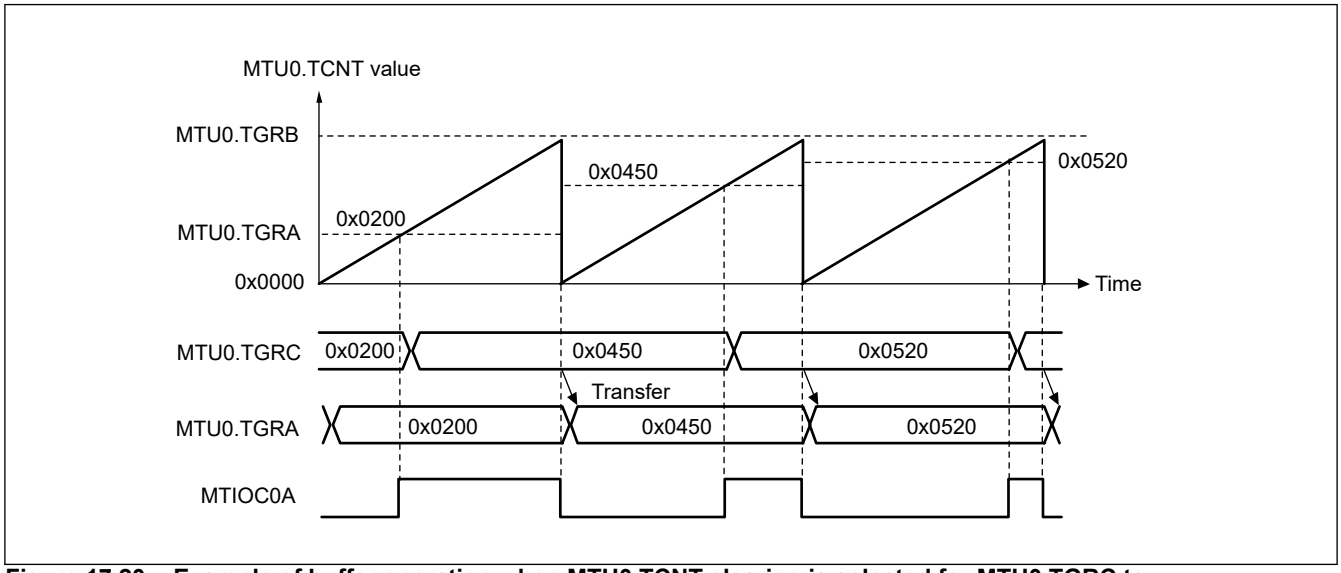


Figure 17.20 Example of buffer operation when MTU0.TCNT clearing is selected for MTU0.TGRC to MTU0.TGRA transfer timing

17.3.4 Cascaded Operation

In cascaded operation, two 16-bit counters in different channels are used together as a 32-bit counter.

There are two functions for connecting MTU1 and MTU2 to use as a 32-bit counter: cascade connection to be set when the MTU1.TMDR3.LWA bit is 0, and cascade connection 32-bit phase counting mode to be set when the MTU1.TMDR3.LWA bit is 1. For details on cascade connection 32-bit phase counting mode, see section 17.3.6.2. Cascade Connection 32-Bit Phase Counting Mode. This section describes the cascade connection function to be set when the MTU1.TMDR3.LWA bit is 0.

This function operates when the MTU1.TMDR3.LWA bit is set to 0 and the MTU1.TCR.TPSC[2:0] bits are set so that MTU1.TCNT counts at an overflow/underflow of MTU2.TCNT. Underflow occurs only when the MTU2 to which the lower 16 bits allocated is in phase counting mode.

Table 17.58 shows the register combinations used in cascaded operation.

When phase counting mode is set for MTU1, the count clock setting is invalid and the counters operate independently in phase counting mode.

Table 17.58 Cascaded combinations

Combination	Upper 16 bits	Lower 16 bits
MTU1 and MTU2	MTU1.TCNT	MTU2.TCNT

For simultaneous input capture of MTU1.TCNT and MTU2.TCNT during cascaded operation, additional input capture input pins can be specified by the input capture control register (TICCR). The input-capture condition is of edges in the signal produced by taking the logical OR of the input level on the main input pin and the input level on the added input pin. Accordingly, if either is at the high level, a change in the level of the other will not produce an edge for detection.

For details, see section 17.3.4.4. Cascaded Operation Example (c). For input capture in cascade connection, see section 17.6.21. Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection.

Table 17.59 shows the TICCR setting and input capture input pins.

Table 17.59 TICCR setting and input capture input pins

Target input capture	TICCR setting	Input capture input pin
Input capture from MTU1.TCNT to MTU1.TGRA	I2AE bit = 0 (Initial value)	MTIOC1A
	I2AE bit = 1	MTIOC1A, MTIOC2A
Input capture from MTU1.TCNT to MTU1.TGRB	I2BE bit = 0 (Initial value)	MTIOC1B
	I2BE bit = 1	MTIOC1B, MTIOC2B
Input capture from MTU2.TCNT to MTU2.TGRA	I1AE bit = 0 (Initial value)	MTIOC2A
	I1AE bit = 1	MTIOC2A, MTIOC1A
Input capture from MTU2.TCNT to MTU2.TGRB	I1BE bit = 0 (Initial value)	MTIOC2B
	I1BE bit = 1	MTIOC2B, MTIOC1B

17.3.4.1 Example of Cascaded Operation Setting Procedure

Figure 17.21 shows an example of the cascaded operation setting procedure.

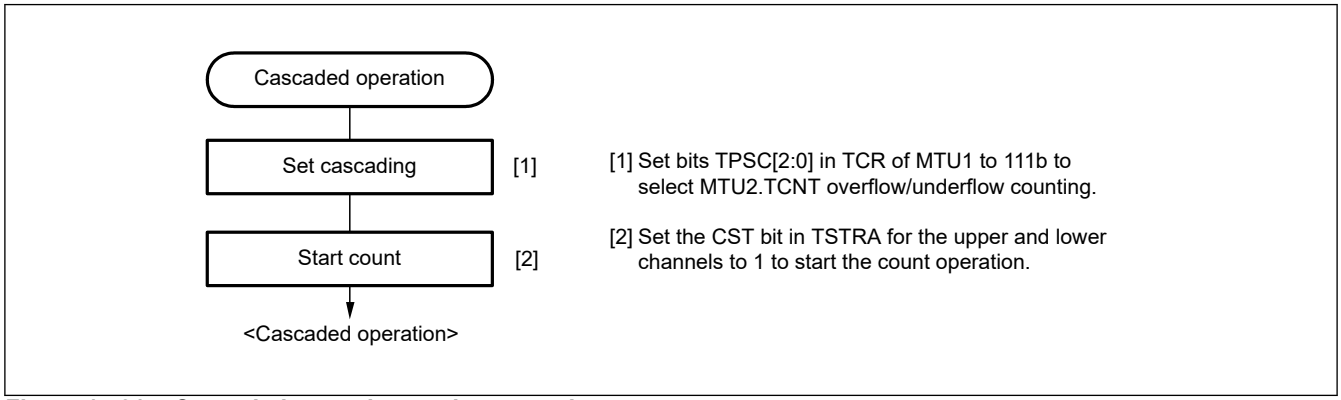


Figure 17.21 Cascaded operation setting procedure

17.3.4.2 Cascaded Operation Example (a)

Figure 17.22 shows the operation when MTU1.TCNT is set for counting at MTU2.TCNT overflow/underflow and MTU2 is set for phase counting mode while MTU1.TCNT and MTU2.TCNT are cascaded.

MTU1.TCNT is incremented by MTU2.TCNT overflow and decremented by MTU2.TCNT underflow.

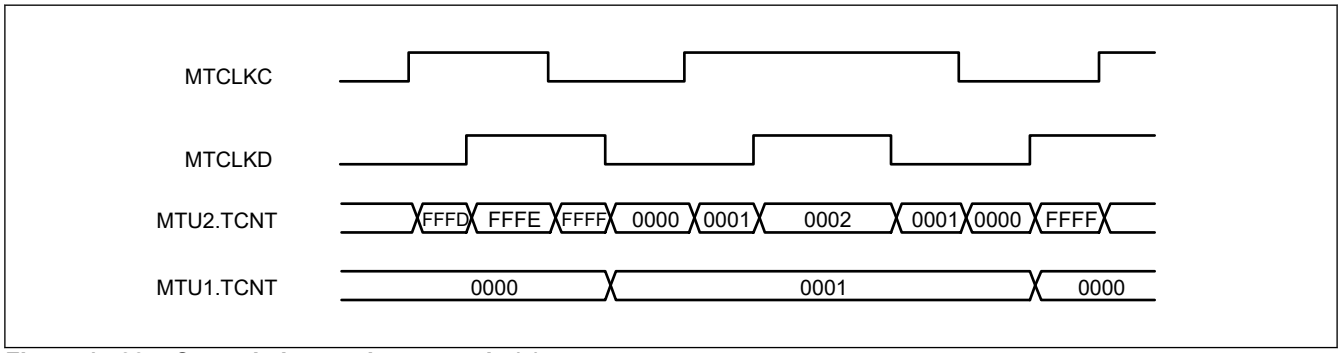


Figure 17.22 Cascaded operation example (a)

17.3.4.3 Cascaded Operation Example (b)

Figure 17.23 shows the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the I2AE bit in TICCR has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the MTU1.TIOR.IOA[3:0] bits have selected the MTIOC1A rising edge for the input capture timing while the MTU2.TIOR.IOA[3:0] bits have selected the MTIOC2A rising edge for the input capture timing.

Under these conditions, the rising edge of both MTIOC1A and MTIOC2A is used for the MTU1.TGRA input capture condition. For the MTU2.TGRA input capture condition, the MTIOC2A rising edge is used.

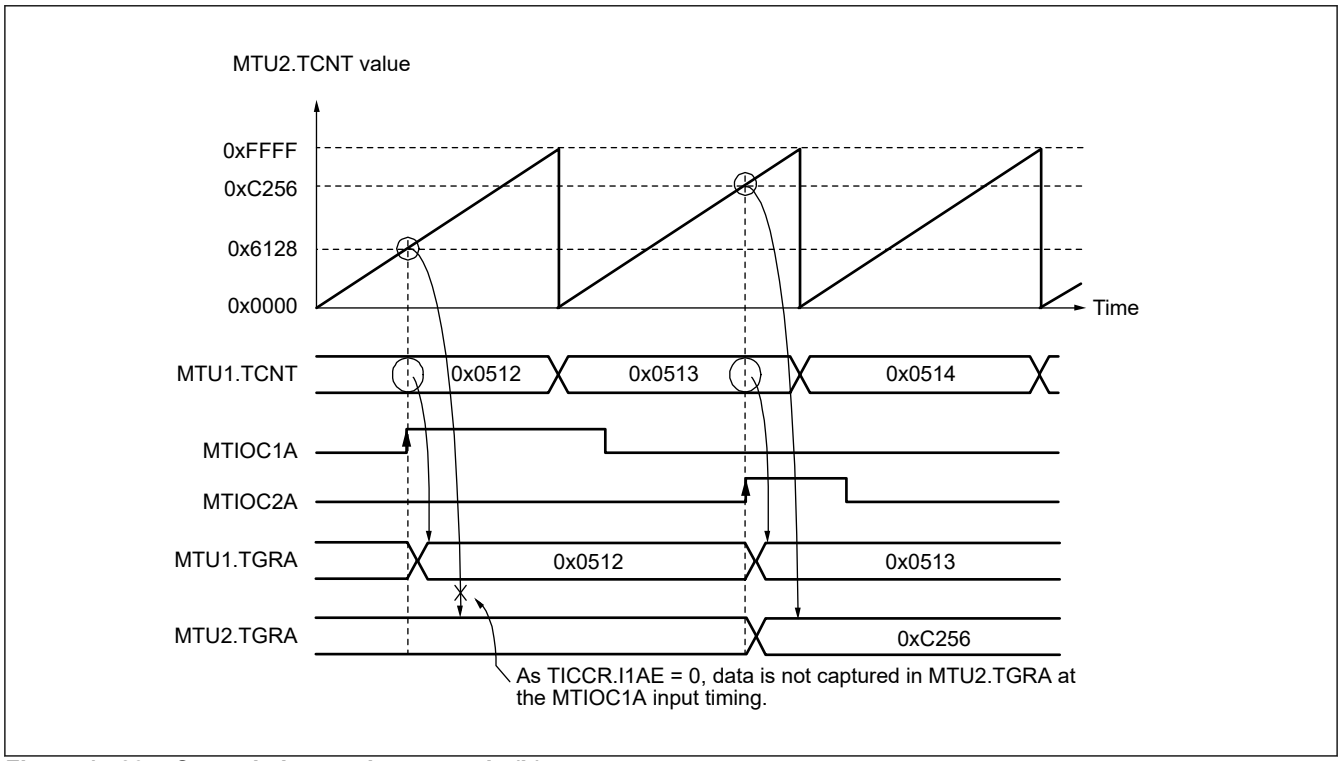
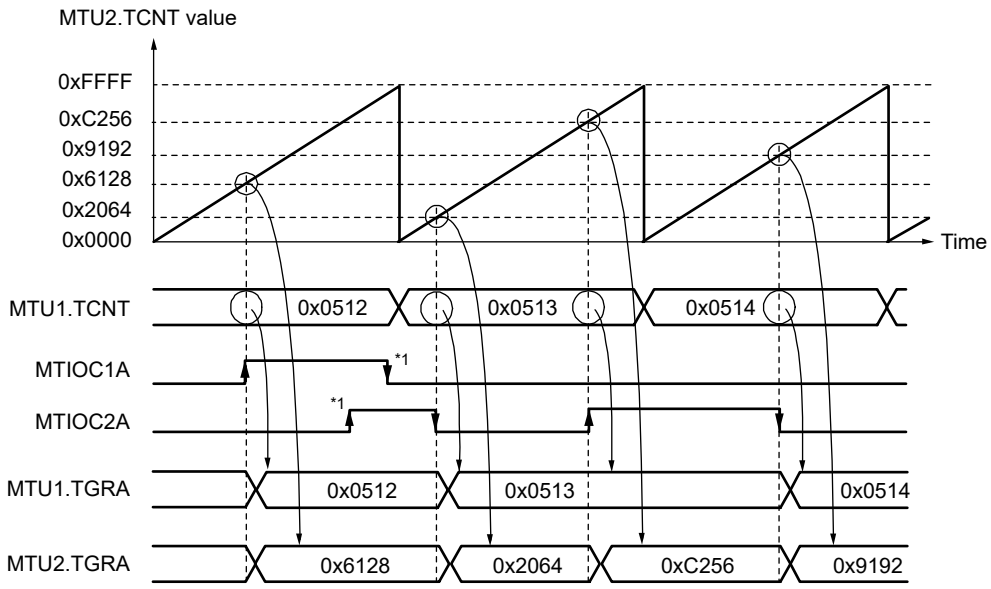


Figure 17.23 Cascaded operation example (b)

17.3.4.4 Cascaded Operation Example (c)

Figure 17.24 shows the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the TICCR.I2AE and I1AE bits have been set to 1 to include the MTIOC2A and MTIOC1A pins in the MTU1.TGRA and MTU2.TGRA input capture conditions, respectively. In this example, the IOA[3:0] bits in both MTU1.TIOR and MTU2.TIOR have selected both the rising and falling edges for the input capture timing. Under these conditions, the ORed result of MTIOC1A and MTIOC2A input is used for the MTU1.TGRA and MTU2.TGRA input capture conditions.



Note 1. When either of the input signals is driven high, the selected edge of the other cannot act as the input capture condition.

Figure 17.24 Cascaded operation example (c)

17.3.4.5 Cascaded Operation Example (d)

Figure 17.25 shows the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the TICCR.I2AE bit has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the IOA[3:0] bits in MTU1.TIOR have selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing while the IOA[3:0] bits in MTU2.TIOR have selected the MTIOC2A rising edge for the input capture timing.

Under these conditions, as MTU1.TIOR has selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing, the MTIOC2A edge is not used for MTU1.TGRA input capture condition although the I2AE bit in TICCR has been set to 1.

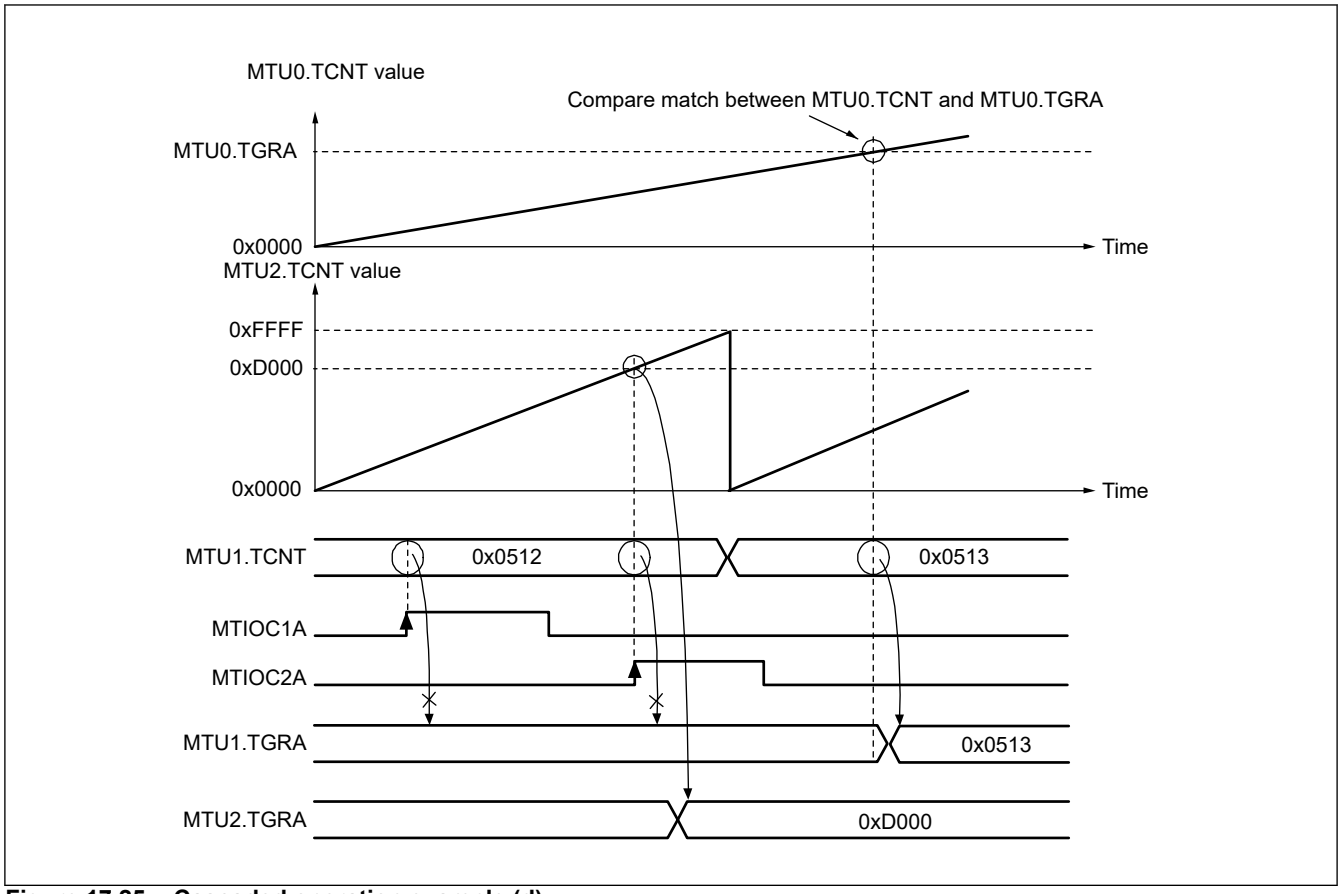


Figure 17.25 Cascaded operation example (d)

17.3.5 PWM Modes

PWM modes are provided to output PWM waveforms from the external pins. The output level can be selected as low, high, or toggle output in response to a compare match of each TGR.

PWM waveforms in the range of 0% to 100% duty cycle can be output according to the TGR settings.

By designating TGR compare match as the counter clearing source, the PWM period can be specified in that register.

Every channel except MTU5 and MTU8 can be set to PWM mode independently. Channels set to PWM mode can perform synchronous operation with each other or other channels set to any other mode.

There are two PWM modes as described below.

(1) PWM Mode 1

PWM waveforms are output from the MTIOCnA and MTIOCnC pins by pairing TGRA with TGRB and TGRC with TGRD. The levels specified by the TIOR.IOA[3:0] and IOC[3:0] bits are output from the MTIOCnA and MTIOCnC pins at compare matches A and C, and the level specified by the TIOR.IOB[3:0] and IOD[3:0] bits are output at compare matches B and D (n = 0 to 4, 6, 7). The initial output value is set in TGRA or TGRC. If the values set in paired TGRs are identical, the output value does not change even when a compare match occurs.

In PWM mode 1, PWM waveforms in up to 12 phases can be output.

(2) PWM Mode 2

PWM waveform output is generated using one TGR as the period register and the others as duty registers. The level specified in TIOR is output at compare matches. Upon counter clearing by a period register compare match, the initial value set in TIOR is output from each pin. If the values set in the period and duty registers are identical, the output value does not change even when a compare match occurs.

Up to eight phases of PWM waveforms can be output by combining synchronous clearing of channels that cannot be set to PWM mode 2 as synchronous operation.

The correspondence between PWM output pins and registers is shown in Table 17.60.

Table 17.60 PWM output registers and output pins

Channel	Register	Output pins	
		PWM mode 1	PWM mode 2
MTU0	TGRA	MTIOC0A	MTIOC0A
	TGRB		MTIOC0B
	TGRC	MTIOC0C	MTIOC0C
	TGRD		MTIOC0D
MTU1	TGRA	MTIOC1A	MTIOC1A
	TGRB		MTIOC1B
MTU2	TGRA	MTIOC2A	MTIOC2A
	TGRB		MTIOC2B
MTU3	TGRA	MTIOC3A	Setting prohibited
	TGRB		
	TGRC	MTIOC3C	
	TGRD		
MTU4	TGRA	MTIOC4A	
	TGRB		
	TGRC	MTIOC4C	
	TGRD		
MTU6	TGRA	MTIOC6A	
	TGRB		
	TGRC	MTIOC6C	
	TGRD		
MTU7	TGRA	MTIOC7A	
	TGRB		
	TGRC	MTIOC7C	
	TGRD		

Note: In PWM mode 2, PWM waveform output is not possible for the TGR register in which the PWM period is set.

17.3.5.1 Example of PWM Mode Setting Procedure

Figure 17.26 shows an example of the PWM mode setting procedure.

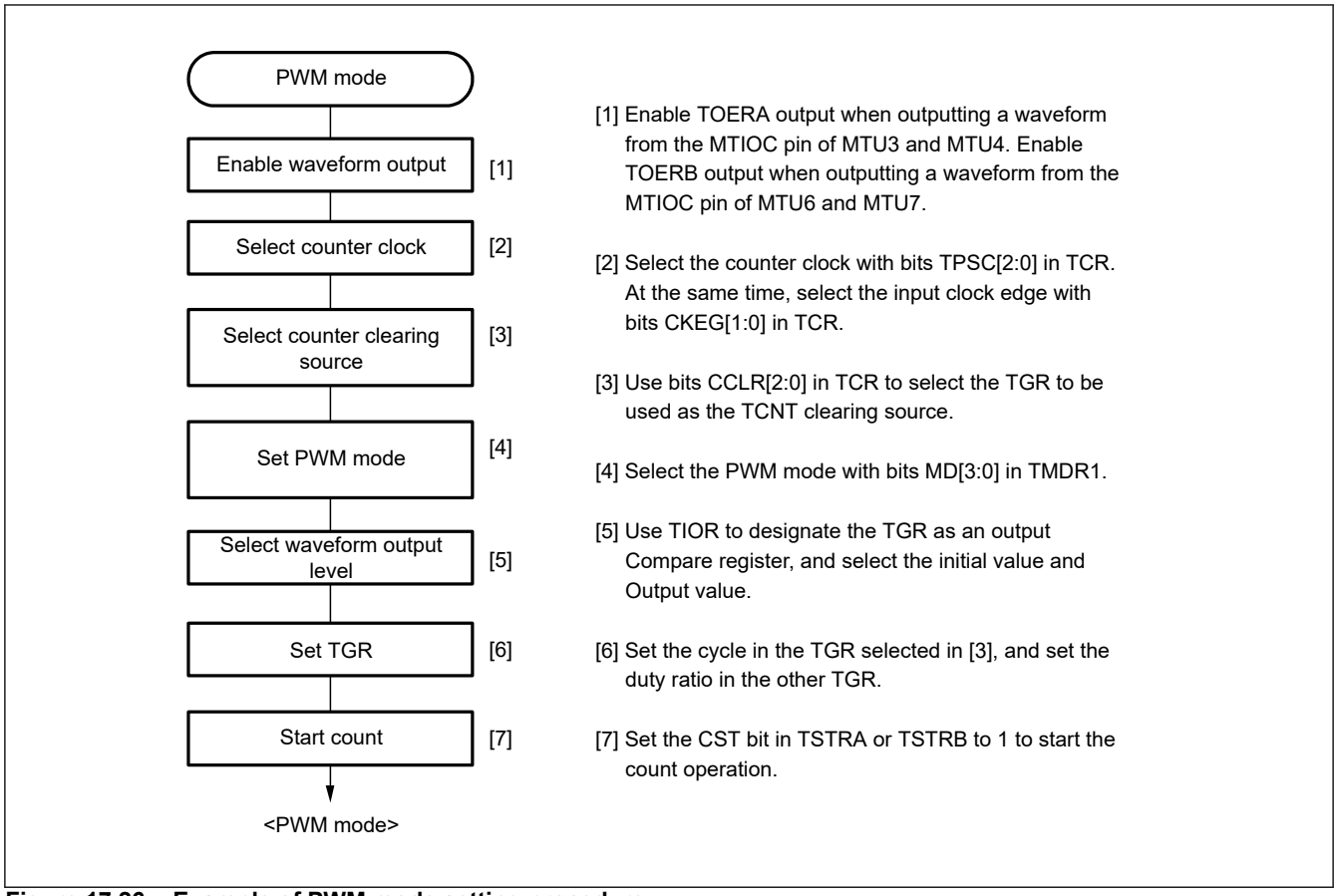


Figure 17.26 Example of PWM mode setting procedure

17.3.5.2 Examples of PWM Mode Operation

Figure 17.27 shows an example of operation in PWM mode 1.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set as the initial output value and output value for TGRA, and 1 is set as the output value for TGRB.

In this case, the value set in TGRA is used as the period, and the value set in TGRB is used as the duty ratio.

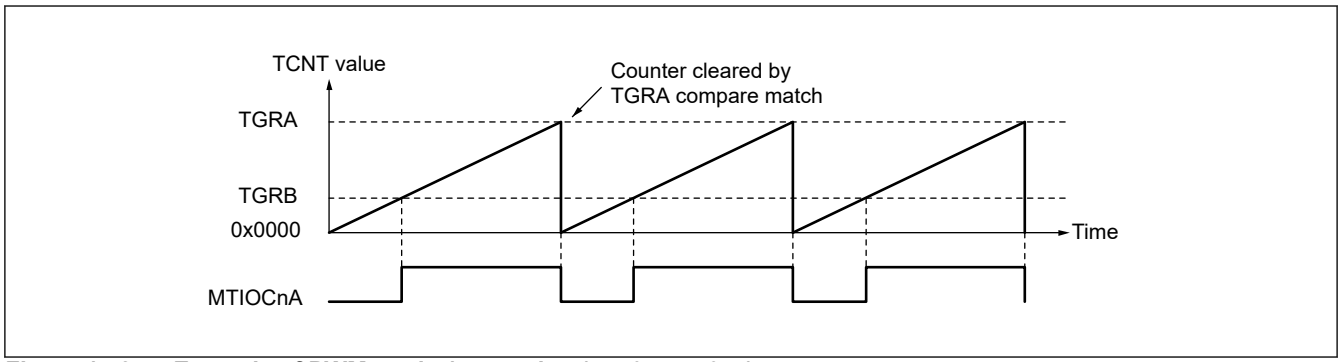


Figure 17.27 Example of PWM mode 1 operation (n = 0 to 4, 6, 7)

Figure 17.28 shows an example of operation in PWM mode 2.

In this example, synchronous operation is designated for MTU0 and MTU1, MTU1.TGRB compare match is set as the TCNT clearing source, and low is set as the initial output value and high as the output value for the other TGR registers (MTU0.TGRA to MTU0.TGRD and MTU1.TGRA), outputting 5-phase PWM waveforms.

In this case, the value set in MTU1.TGRB is used as the period, and the values set in the other TGRs are used as the duty ratio.

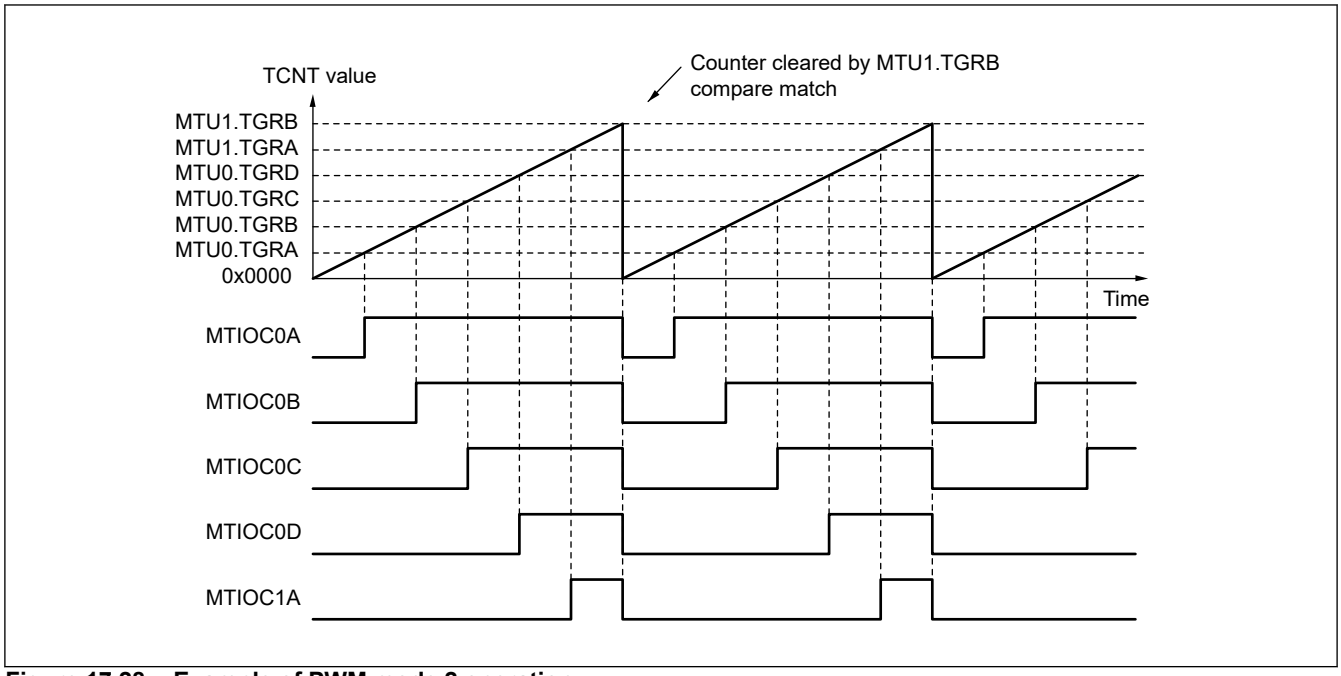


Figure 17.28 Example of PWM mode 2 operation

Figure 17.29 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode 1.

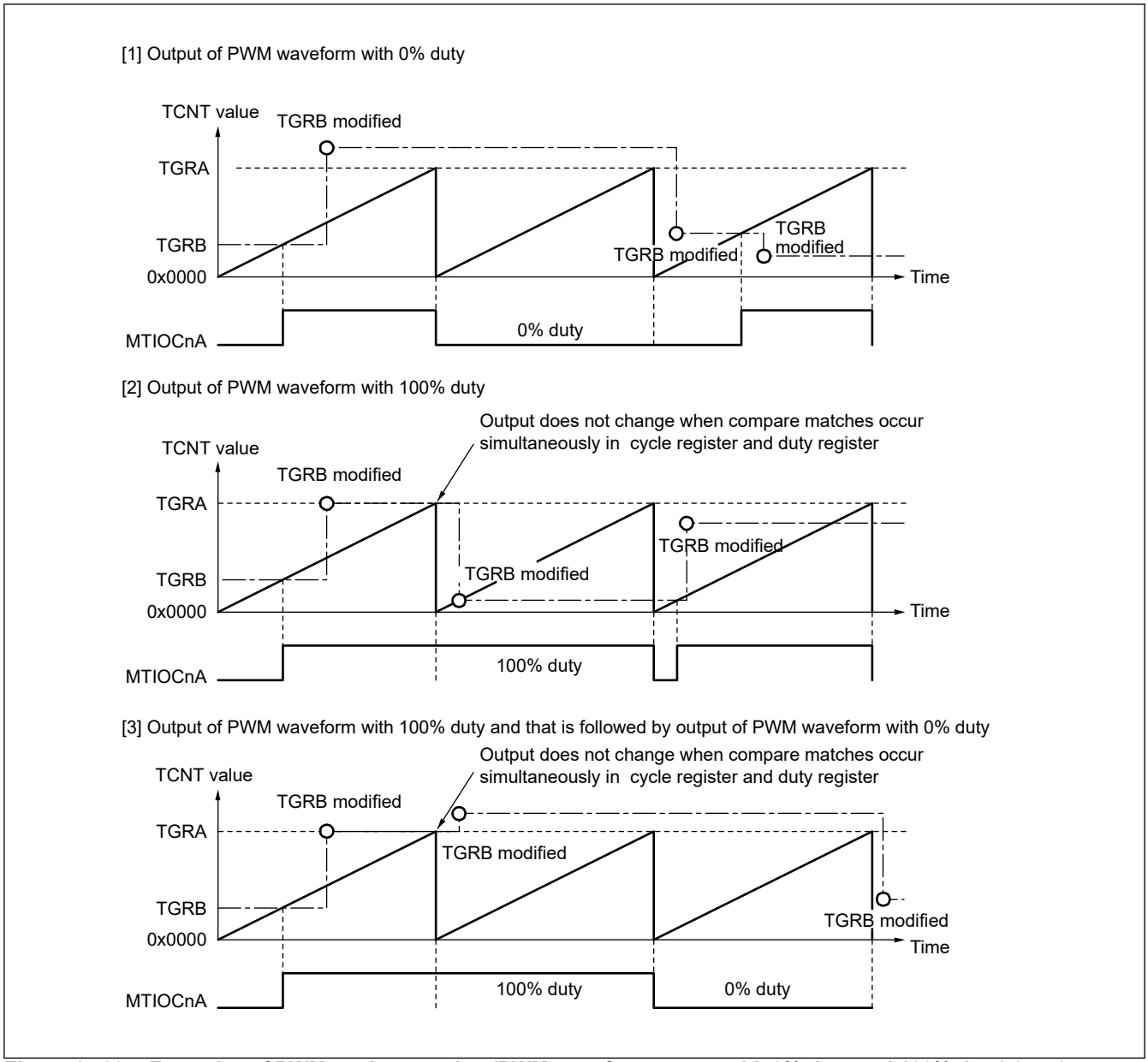


Figure 17.29 Examples of PWM mode operation (PWM waveform output with 0% duty and 100% duty) (n = 0 to 4, 6, 7)

17.3.6 Phase Counting Mode

There are two phase counting modes: 16-bit phase counting mode in which MTU1 and MTU2 operate independently, and cascade connection 32-bit phase counting mode in which MTU1 and MTU2 are cascaded.

In phase counting mode, the phase difference between two external input clocks is detected and the corresponding TCNT is incremented or decremented.

Two external clock input pins for each phase counting mode are not affected by the settings of TCR.TPSC[2:0], TCR2.TPSC2[2:0], and CKEG[1:0]. The two external clock input pins used in 16-bit phase counting mode and cascade connection 32-bit phase counting mode of MTU2 can be selected by MTU1.TMDR3.PHCKSEL. In a phase counting mode other than 16-bit phase counting mode and cascade connection 32-bit phase counting mode of MTU2, MTCLKA and MTCLKB are selected for A-phase and B-phase, respectively. In phase counting mode, the external clock pins MTCLKA, MTCLKB, MTCLKC, and MTCLKD are used for two-phase encoder pulse input.

Table 17.61 lists the external clock input pins to be connected in each phase counting mode.

Table 17.61 Clock input pins in phase counting mode

Phase counting mode	TMDR3.PHCKSEL bit	External clock input pins	
		A-phase	B-phase
MTU1 16-bit phase counting mode	x (Don't care)	MTCLKA	MTCLKB
MTU2 16-bit phase counting mode	0	MTCLKA	MTCLKB
	1 (initial value)	MTCLKC	MTCLKD
Cascade connection 32-bit phase counting mode	0	MTCLKA	MTCLKB
	1 (initial value)	MTCLKC	MTCLKD

17.3.6.1 16-Bit Phase Counting Mode

When the MTU1.TMDR3.LWA is 0, 16-bit phase counting mode can be set individually for MTU1 and MTU2.

In 16-bit phase counting mode, the phase difference between two external input clocks is detected and the 16-bit counter TCNT of the corresponding channel is incremented or decremented.

When 16-bit phase counting mode is specified, an external clock is selected as the count clock and TCNT operates as an up-counter/down-counter regardless of the setting of TCR.TPSC[2:0], TCR2.TPSC2[2:0], and CKEG[1:0]. However, the functions of TCR.CCLR[1:0], TIOR, TIER, and TGR are enabled, and input capture/compare match and interrupt functions can be used.

These external input pins can be used for two-phase encoder pulse input.

When an overflow occurs while TCNT is counting up and the corresponding TIER.TCIEV bit is 1, a TCIV interrupt is generated. When an underflow occurs while TCNT is counting down and the corresponding TIER.TCIEU bit is 1, a TCIU interrupt is generated.

The TSR.TCFD flag is the count direction flag. Read the TCFD flag to check whether TCNT is counting up and down.

17.3.6.1.1 Example of 16-Bit Phase Counting Mode Setting Procedure

Figure 17.30 shows an example of the 16-bit phase counting mode setting procedure.

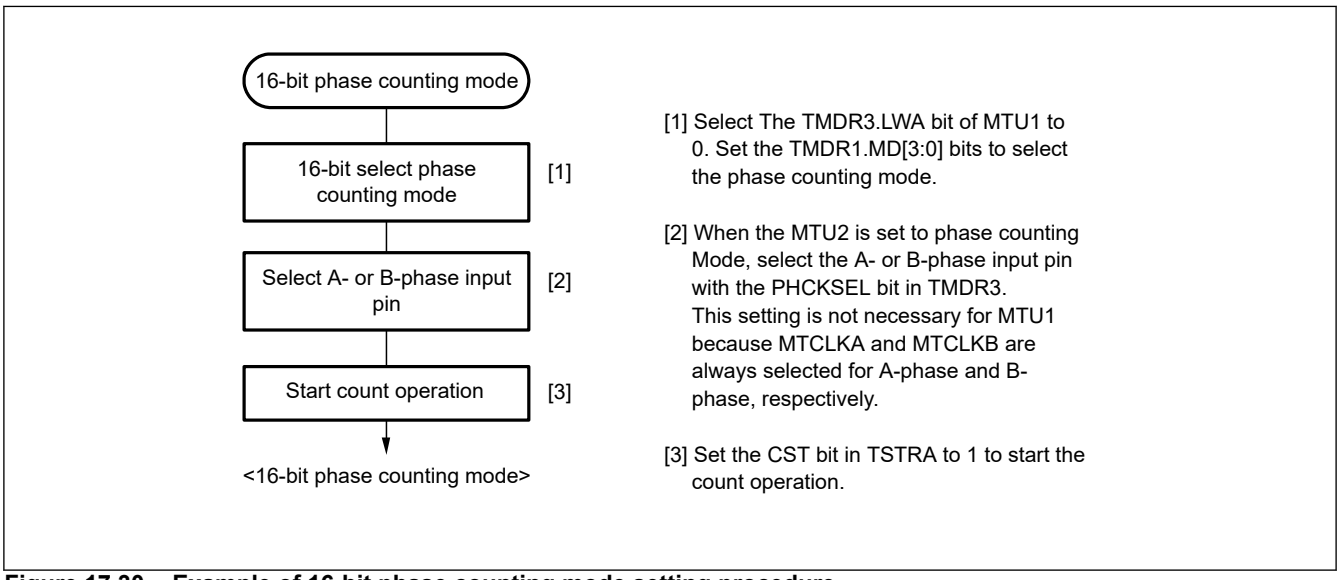


Figure 17.30 Example of 16-bit phase counting mode setting procedure

17.3.6.1.2 Examples of 16-Bit Phase Counting Mode Operation

In phase counting mode, TCNT is incremented or decremented according to the phase difference between two external clocks. There are five modes according to the count conditions. Each mode operates under the condition PHCKSEL = 1, which means the phase clock for MTU1 is input from MTCLKA or MTCLKB and that for MTU2 is input from MTCLKC or MTCLKD.

(1) Phase Counting Mode 1

Figure 17.31 shows an example of operation in phase counting mode 1, and Table 17.62 summarizes the TCNT upcounting and down-counting conditions.

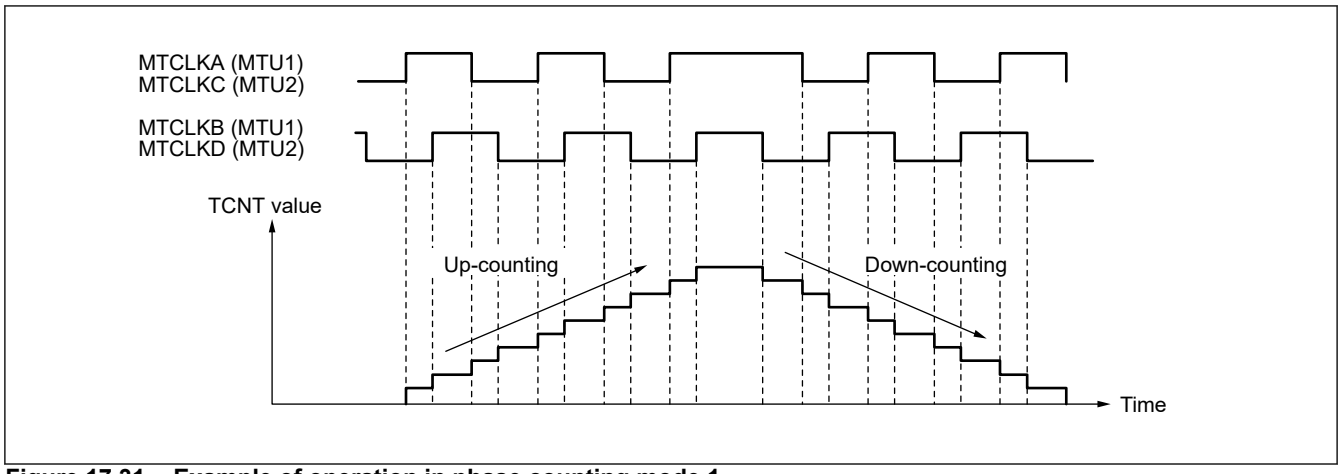


Figure 17.31 Example of operation in phase counting mode 1

Table 17.62 Up-counting and down-counting conditions in phase counting mode 1

: Rising edge

: Falling edge

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High	↑	Up-counting
Low	↓	
↑	Low	
↓	High	
High	↓	Down-counting
Low	↑	
↑	High	
↓	Low	

(2) Phase Counting Mode 2

Figure 17.32 to Figure 17.34 show the examples of operation in phase counting mode 2 and Table 17.63 summarizes the TCNT up-counting and down-counting conditions.

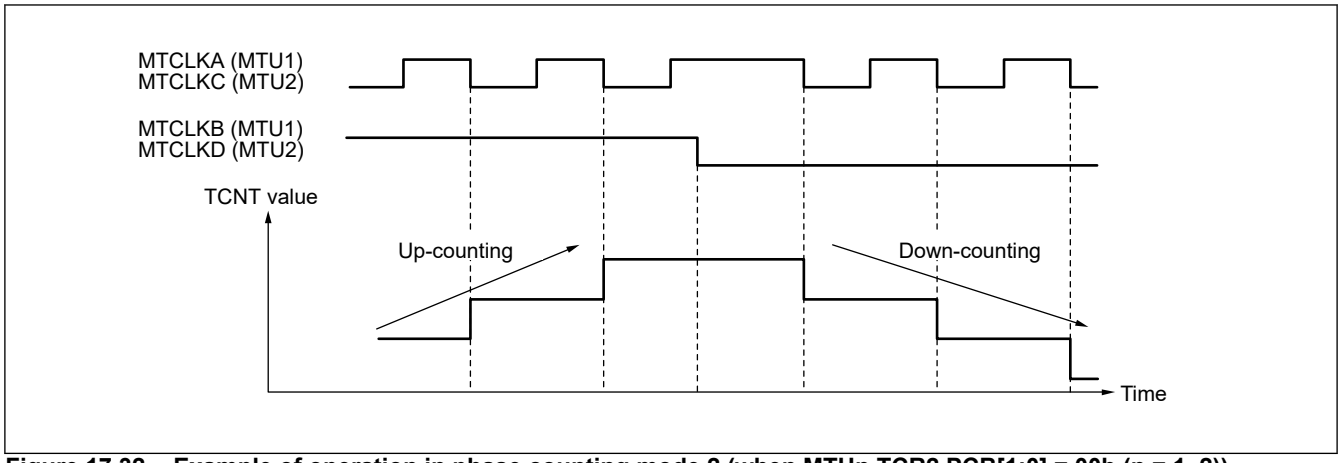


Figure 17.32 Example of operation in phase counting mode 2 (when MTUn.TCR2.PCB[1:0] = 00b (n = 1, 2))

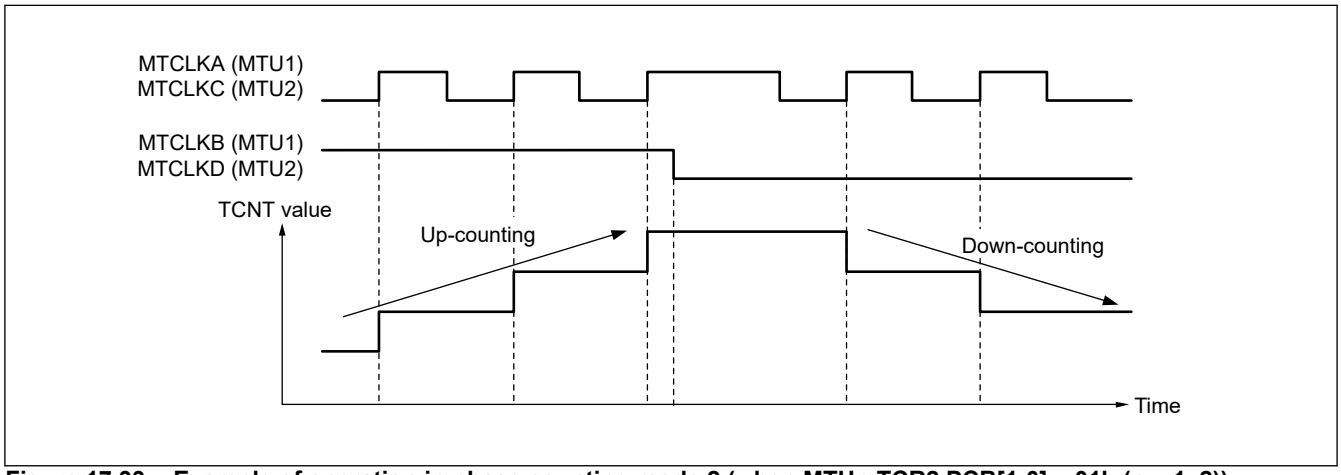


Figure 17.33 Example of operation in phase counting mode 2 (when MTUn.TCR2.PCB[1:0] = 01b (n = 1, 2))

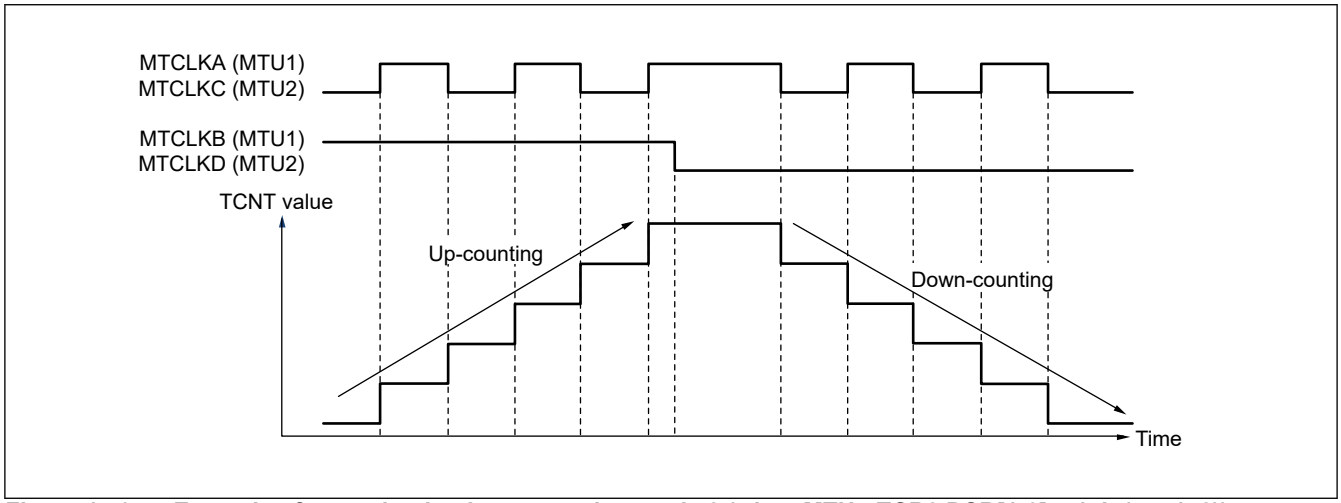


Figure 17.34 Example of operation in phase counting mode 2 (when MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 17.63 Up-counting and down-counting conditions in phase counting mode 2

: Rising edge

: Falling edge

PCB[1:0]	MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
00b	High		Not counted (Don't care)
	Low		
		Low	
		High	Up-counting
	High		Not counted (Don't care)
	Low		
		High	
		Low	Down-counting
01b	High		Not counted (Don't care)
	Low		
		Low	Down-counting
		High	Not counted (Don't care)
	High		
	Low		
		High	Up-counting
		Low	Not counted (Don't care)
1xb	High		Not counted (Don't care)
	Low		
		Low	Down-counting
		High	Up-counting
	High		Not counted (Don't care)
	Low		
		High	Up-counting
		Low	Down-counting

(3) Phase Counting Mode 3

Figure 17.35 to Figure 17.37 show the examples of operation in phase counting mode 3 and Table 17.64 summarizes the TCNT up-counting and down-counting conditions.

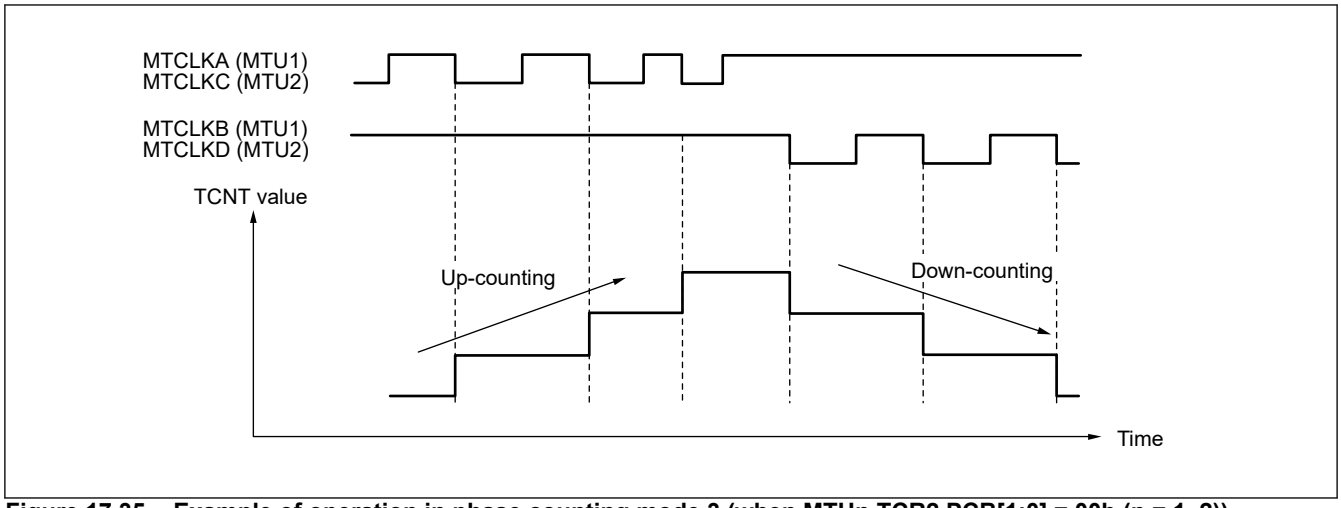


Figure 17.35 Example of operation in phase counting mode 3 (when MTUn.TCR2.PCB[1:0] = 00b (n = 1, 2))

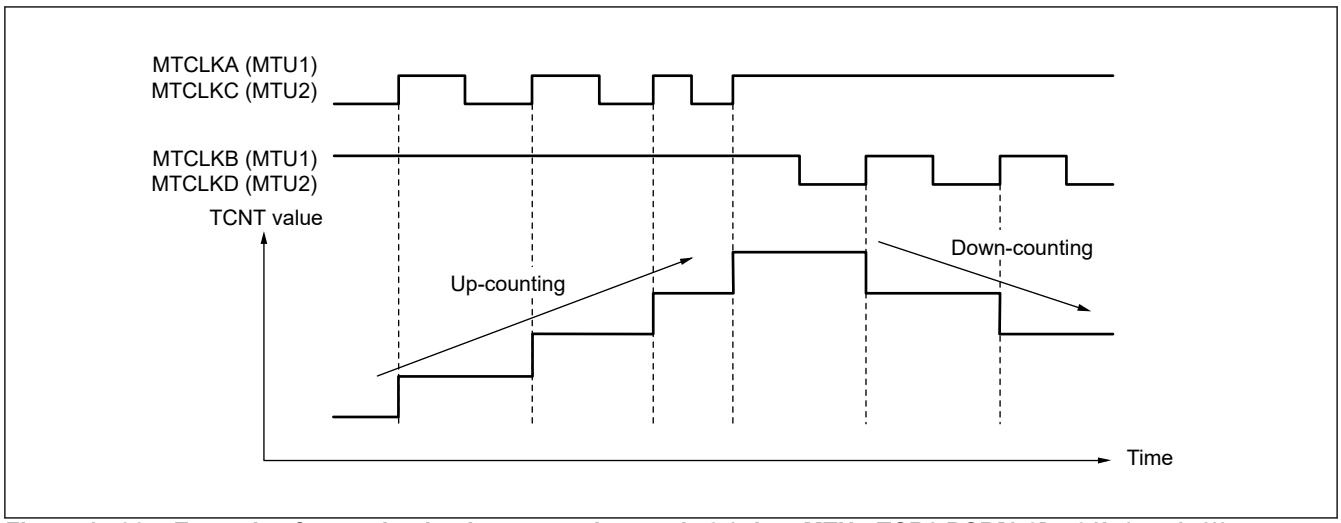


Figure 17.36 Example of operation in phase counting mode 3 (when MTUn.TCR2.PCB[1:0] = 01b (n = 1, 2))

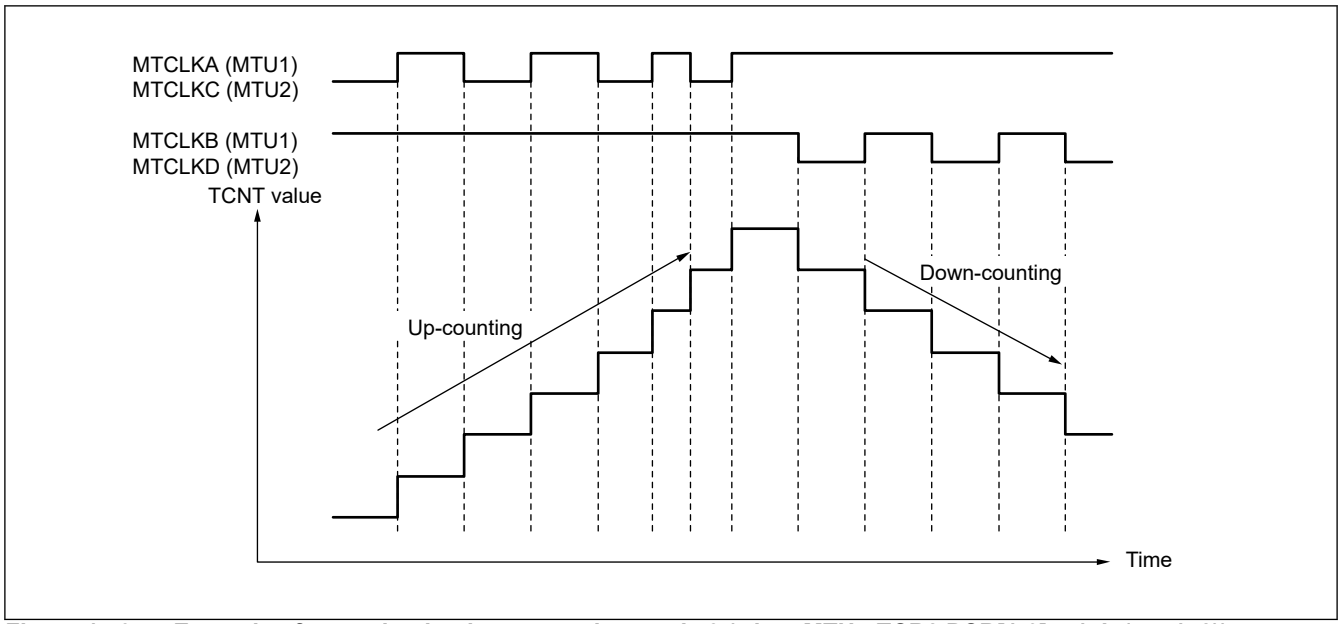


Figure 17.37 Example of operation in phase counting mode 3 (when MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 17.64 Up-counting and down-counting conditions in phase counting mode 3

: Rising edge

: Falling edge

PCB[1:0]	MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
00b	High	↑	Not counted (Don't care)
	Low	↓	
	↑	Low	
	↓	High	Up-counting
	High	↓	Down-counting
	Low	↑	Not counted (Don't care)
	↑	High	
	↓	Low	
01b	High	↑	Down-counting
	Low	↓	Not counted (Don't care)
	↑	Low	
	↓	High	
	High	↓	
	Low	↑	
	↑	High	Up-counting
	↓	Low	Not counted (Don't care)
1xb	High	↑	Down-counting
	Low	↓	Not counted (Don't care)
	↑	Low	
	↓	High	Up-counting
	High	↓	Down-counting
	Low	↑	Not counted (Don't care)
	↑	High	Up-counting
	↓	Low	Not counted (Don't care)

(4) Phase Counting Mode 4

Figure 17.38 shows an example of operation in phase counting mode 4, and Table 17.65 summarizes the TCNT up-counting and down-counting conditions.

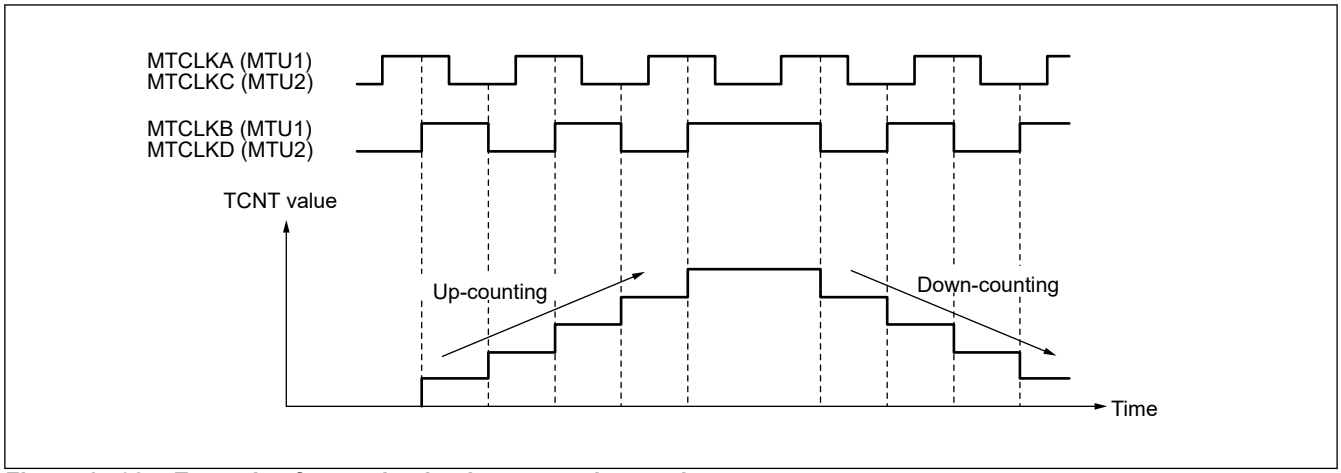


Figure 17.38 Example of operation in phase counting mode 4

Table 17.65 Up-counting and down-counting conditions in phase counting mode 4

: Rising edge

: Falling edge

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High	↑	Up-counting
Low	↓	
↑	Low	Not counted (Don't care)
↓	High	
High	↓	Down-counting
Low	↑	
↑	High	Not counted (Don't care)
↓	Low	

(5) Phase Counting Mode 5

Figure 17.39 and Figure 17.40 show the examples of operation in phase counting mode 5 and Table 17.66 summarizes the TCNT up-counting and down-counting conditions.

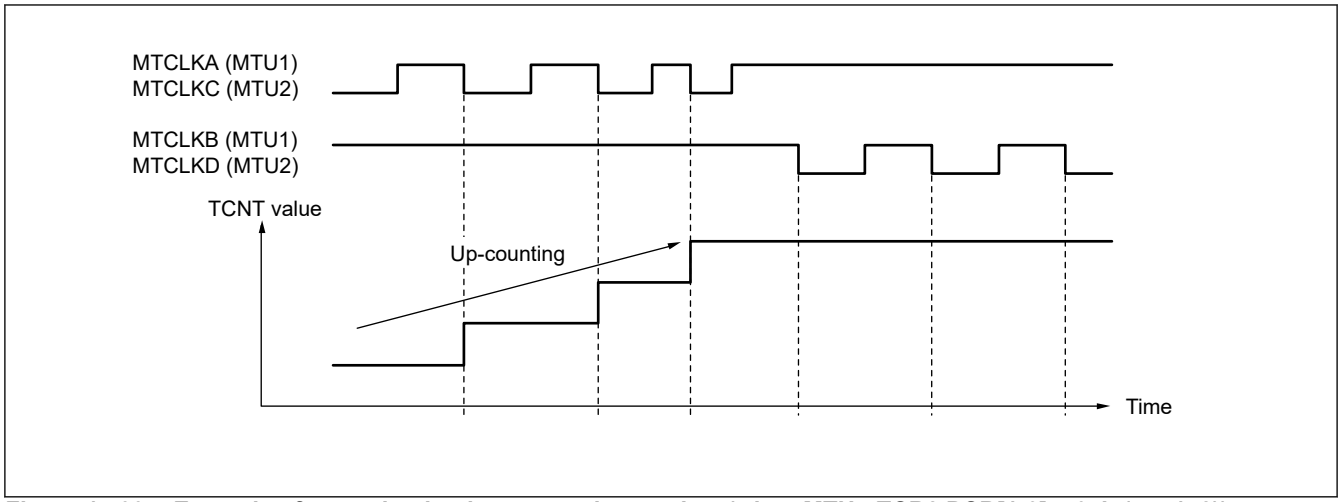


Figure 17.39 Example of operation in phase counting mode 5 (when MTUn.TCR2.PCB[1:0] = 0xb (n = 1, 2))

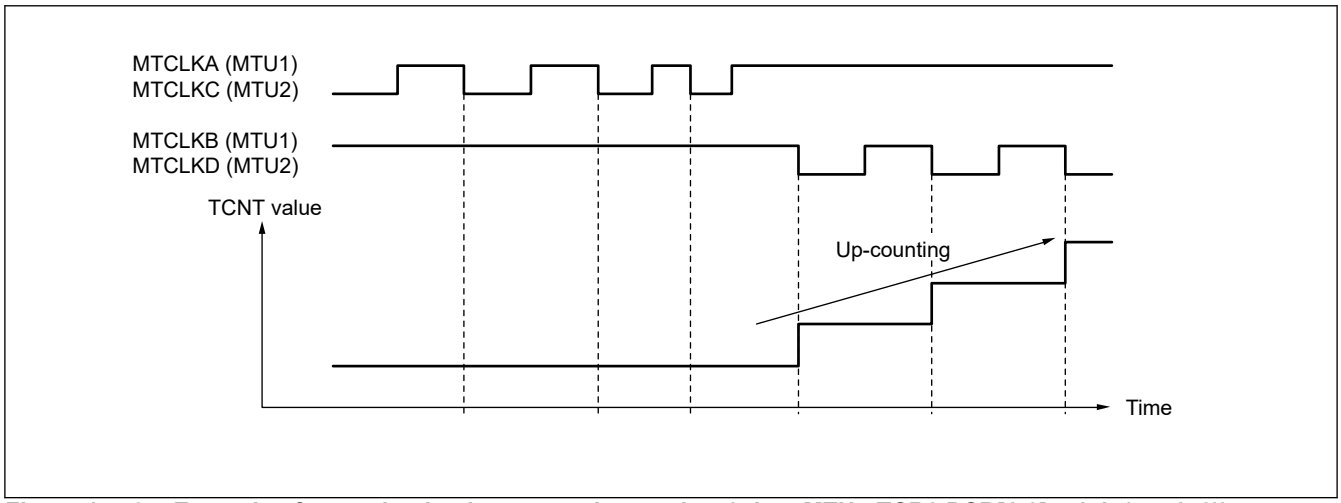


Figure 17.40 Example of operation in phase counting mode 5 (when MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 17.66 Up-counting and down-counting conditions in phase counting mode 5 (1 of 2)

: Rising edge

: Falling edge

PCB[1:0]	MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
0xb	High		Not counted (Don't care)
	Low		
		Low	
		High	Up-counting
	High		Not counted (Don't care)
	Low		
		High	
		Low	Up-counting

Table 17.66 Up-counting and down-counting conditions in phase counting mode 5 (2 of 2)

: Rising edge

: Falling edge

PCB[1:0]	MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
1xb	High		Not counted (Don't care)
	Low		Up-counting
		Low	Not counted (Don't care)
		High	
	High		Up-counting
	Low		Not counted (Don't care)
		High	
		Low	

17.3.6.1.3 16-Bit Phase Counting Mode Application Example

Figure 17.41 shows an example in which MTU1 is in phase counting mode, and MTU1 is coupled with MTU0 to input 2-phase encoder pulses of a servo motor in order to detect position or speed.

MTU1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to MTCLKA and MTCLKB.

In MTU0, MTU0.TGRC compare match is specified as the TCNT clearing source and MTU0.TGRA and MTU0.TGRC are used for the compare match function and are set with the speed control period and position control period.

MTU0.TGRB is used for input capture, with MTU0.TGRB and MTU0.TGRD operating in buffer mode. The MTU1 count clock is designated as the MTU0.TGRB input capture source, and the widths of 2-phase encoder 4-multiplication pulses are detected.

MTU1.TGRA and MTU1.TGRB for MTU1 are designated for the input capture function and MTU0.TGRA and MTU0.TGRC compare matches in MTU0 are selected as the input capture sources to store the up/down-counter values for the control periods.

This procedure enables the accurate detection of position and speed.

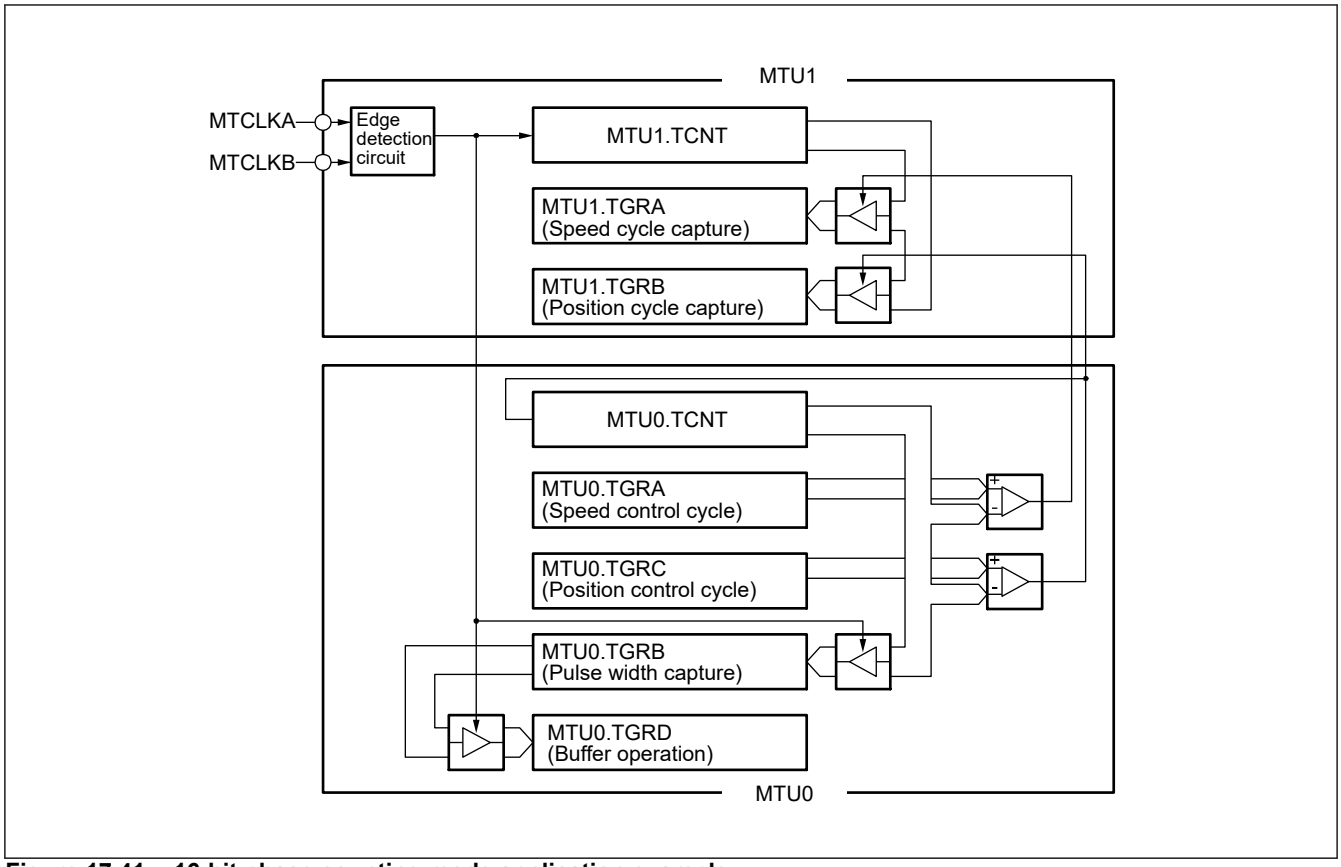


Figure 17.41 16-bit phase counting mode application example

17.3.6.2 Cascade Connection 32-Bit Phase Counting Mode

When MTU1 is set to phase counting mode by setting MTU1.TMDR3.LWA = 1, MTU1 and MTU2 are connected to operate in cascade connection 32-bit phase counting mode as shown in Figure 17.42. When this mode is used, the TCR, TCR2, TIOR, TIER, TGR, and TSR registers are controlled by MTU1 and the settings of MTU2 are disabled. See Figure 17.43 for the procedure for setting cascade connection 32-bit phase counting mode.

In this mode, three-phase (A, B, and Z) signals can be input. As an encoder pulse signal, the external input phase clocks MTCLKA and MTCLKB or MTCLKC and MTCLKD can be selected for A-phase and B-phase, and MTIOC1A can be selected for Z-phase, respectively. See Table 17.61 for selecting external clock input of A-phase and B-phase. A counter event is generated using an A-phase or B-phase pulse and counted by the 32-bit counter MTU1.TCNTLW.

An input capture can also be generated using a Z-phase signal; thus angular velocity can be measured using the captured value in the general register.

Furthermore, MTU8 can be used as a channel for measuring a control period interval, and a compare match signal can be output at a control period interval to the MTU1 and MTU2, which operate in cascade connection 32-bit phase counting mode. That is, a compare match signal of MTU8 is used as a capture signal of MTU1 and MTU2, and the number of A-phase and B-phase pulses for a control period can be measured.

When MTU0 or MTU5 is specified as the channel for measuring a Z-phase signal pulse, the compare match signal of the MTU8.TGRC register can be output as a capture signal or clear signal to MTU0 or MTU5, thus the Z-phase count at a control period interval can be measured.

In addition, a counter event signal of combined MTU1 and MTU2 can be used as a capture signal of the MTU8.TGRD register, and measurement can be performed including the intervals of A, B, or both phase pulses. In this case, the MTU8.TGRD register should be set to buffer operation.

See section 17.3.4. Cascaded Operation, for details on the cascade connection function for connecting MTU1 and MTU2 in a mode other than cascade connection 32-bit phase counting mode.

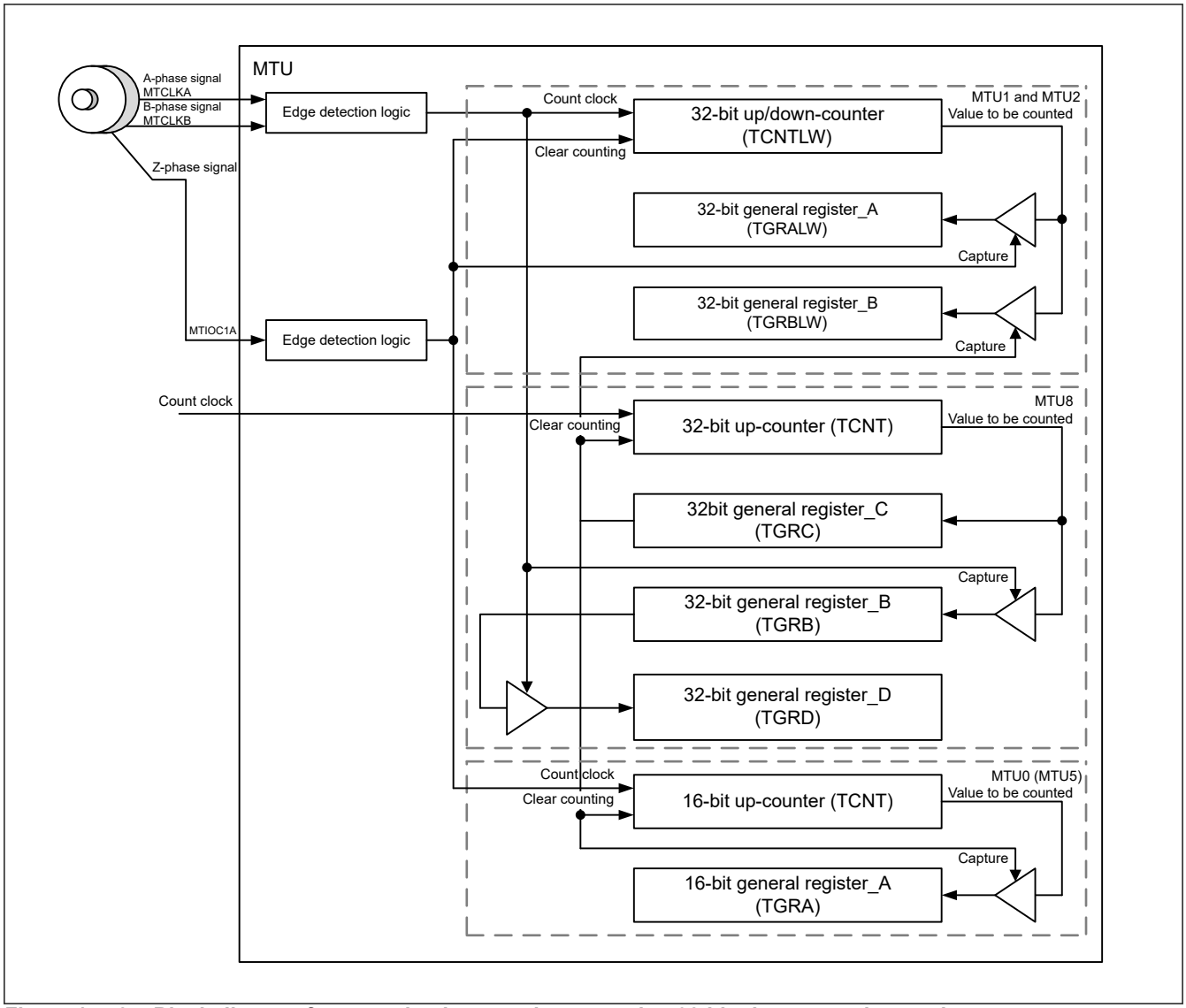
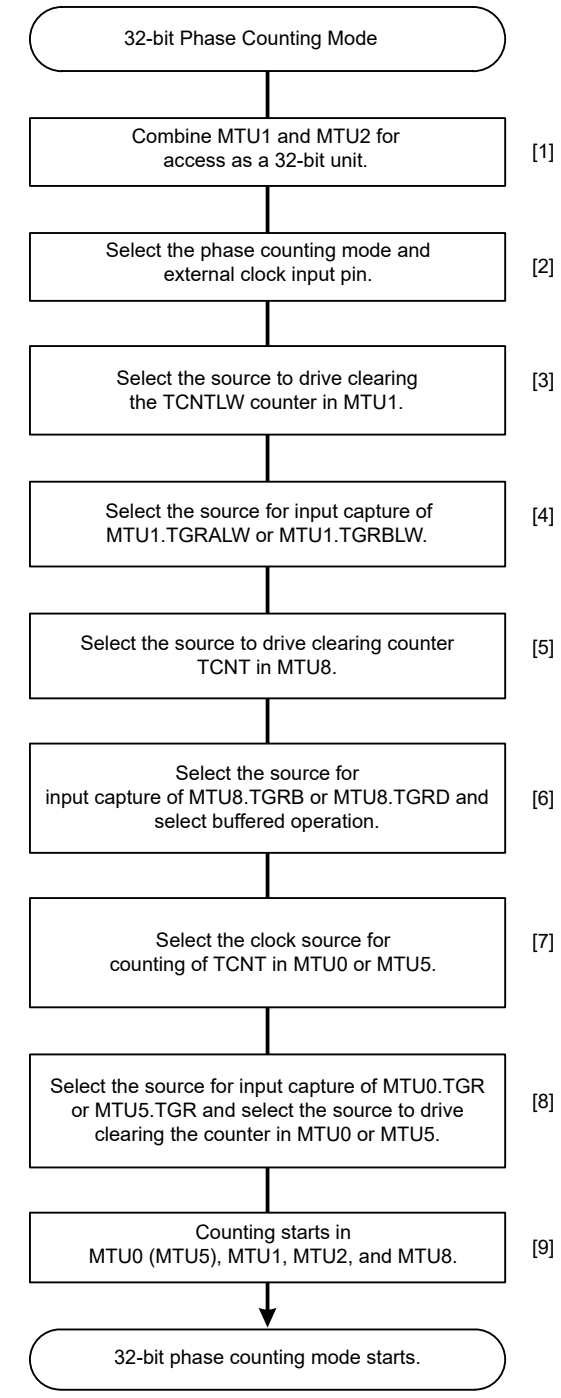


Figure 17.42 Block diagram for operation in cascade connection 32-bit phase counting mode

17.3.6.2.1 Example of Setting Cascade Connection 32-Bit Phase Counting Mode

Figure 17.43 shows an example of the procedure for setting cascade connection 32-bit phase counting mode.



[1] Set the LWA bit in MTU1.TMDR3 to 1 and combine MTU1 and MTU2 for access as a 32-bit unit.

[2] Select phase counting mode with the MD[3:0] bits in MTU1.TMDR1 and select the external clock input pin with the PHCKSEL bits in MTU1.TMDR3.

[3] Set the CCLR[2:0] bits in MTU1.TCR to 001b to select input capture of MTU1.TGRALW as the source of the signal to drive clearing counter TCNTLW in MTU1.

[4] Set the IOA[3:0] bits in MTU1.TIOR to select the effective edge (rising, falling or both) of the signal produced by input capture of TGRALW in MTU1. After that, set the IOB[3:0] bits in MTU1.TIOR to 1110b to select the compare match signal of MTU8.TGRC as the source to drive input capture of MTU1.TGRBLW.

[5] Set the CCLR[2:0] bits in MTU8.TCR to 101b to select the compare match signal of MTU8.TGRC as the source to drive clearing counter TCNT in MTU8.

[6] Select the source for input capture of MTU8.TGRB as the clock source for counting of TCNTLW in MTU1 by setting the IOB[3:0] bits in MTU8.TIORH to 1110b. At the same time, select buffered operation for TGRB and TGRD by setting the BFB bit in MTU8.TMDR1 to 1.

[7] Set the TPSC2[2:0] bits in MTU0.TCR2 to 111b to select the MTIOC1A pin as the clock source for counting of TCNT in MTU0.
(When using TCNT in MTU5, set the TPSC2[2:0] bits in MTU5.TCR2 to 111b).

[8] Select the compare match signal of MTU8.TGRC as the source to drive input capture of MTU0.TGRA by setting the IOA[3:0] bits in MTU0.TIORH to 1110b.
(When using TCNT in MTU5, set the TIORU, TIORV, or TIORW bit in MTU5 is set to 10100b).
After that, select input capture of MTU0.TGRA as the source to drive clearing counter TCNT in MTU0 by setting the CCLR[2:0] bits in MTU0.TCR to 001b.

[9] Start counting in MTU0, MTU1, MTU2, and MTU8 by setting the TSTRA bit to 00001111b.
(When using TCNT in MTU5, set the counter corresponding to TSTR in MTU5 to 1).

Note: When the LWA bit in MTU1.TMDR3 is set to 1, the TCR, TCR2, TMDR1, and TIOR registers in MTU1 are effective in controlling cascaded operation. In this case, the settings of these registers in MTU2 are ignored.

Figure 17.43 Procedure for setting cascade connection 32-bit phase counting mode

17.3.7 Reset-Synchronized PWM Mode

In the reset-synchronized PWM mode, six phases of positive and negative PWM waveforms (12 phases in total) that share a common wave transition point can be output by combining MTU3 and MTU4 and MTU6 and MTU7.

When set for reset-synchronized PWM mode, the MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, and MTIOC7D pins function as PWM output pins and timer counters 3 and 6 (MTU3.TCNT and MTU6.TCNT) functions as an up-counter.

Table 17.67 shows the PWM output pins used. Table 17.68 shows the settings of the registers.

Table 17.67 Output pins for reset-synchronized PWM mode

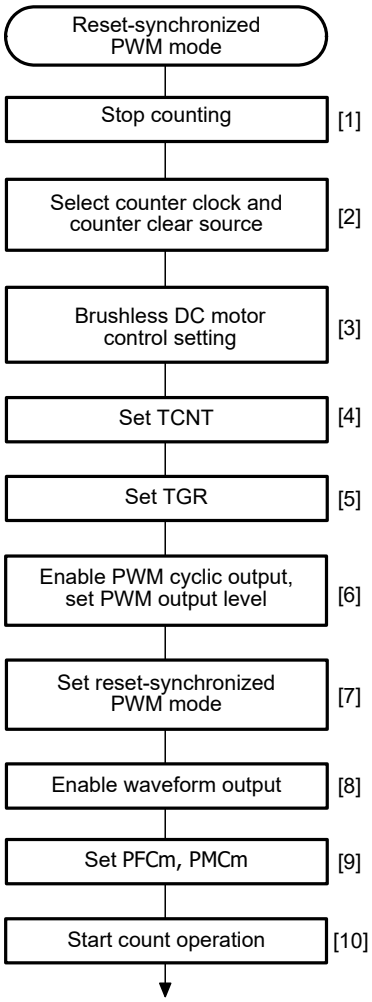
Channel	Output pin	Description
MTU3	MTIOC3B	PWM output pin 1
	MTIOC3D	PWM output pin 1' (negative-phase waveform of PWM output 1)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (negative-phase waveform of PWM output 2)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (negative-phase waveform of PWM output 3)
MTU6	MTIOC6B	PWM output pin 4
	MTIOC6D	PWM output pin 4' (negative-phase waveform of PWM output 4)
MTU7	MTIOC7A	PWM output pin 5
	MTIOC7C	PWM output pin 5' (negative-phase waveform of PWM output 5)
	MTIOC7B	PWM output pin 6
	MTIOC7D	PWM output pin 6' (negative-phase waveform of PWM output 6)

Table 17.68 Register settings for reset-synchronized PWM mode

Register	Setting
MTU3.TCNT	Initial setting (0x0000)
MTU4.TCNT	Initial setting (0x0000)
MTU3.TGRA	Set the count period for MTU3.TCNT
MTU3.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC3B and MTIOC3D pins
MTU4.TGRA	Set the transition point of the PWM waveform to be output from the MTIOC4A and MTIOC4C pins
MTU4.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC4B and MTIOC4D pins
MTU6.TCNT	Initial setting (0x0000)
MTU7.TCNT	Initial setting (0x0000)
MTU6.TGRA	Set the count period for MTU6.TCNT
MTU6.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC6B and MTIOC6D pins
MTU7.TGRA	Set the transition point of the PWM waveform to be output from the MTIOC7A and MTIOC7C pins
MTU7.TGRB	Set the transition point of the PWM waveform to be output from the MTIOC7B and MTIOC7D pins

17.3.7.1 Example of Procedure for Setting Reset-Synchronized PWM Mode

Figure 17.44 shows an example of procedure for setting the reset-synchronized PWM mode.



[1] Clear the CST3 and CST4 bits in TSTRA to 0 to stop the TCNT operation. Specify the reset-synchronized PWM mode while MTU3.TCNT and MTU4.TCNT are stopped.

[2] Set bits TPSC[2:0] and CKEG[1:0] in MTU3.TCR to select the counter clock and clock edge for MTU3. Set bits CCLR[2:0] to select TGRA compare match as a counter clear source.

[3] When performing brushless DC motor control, set bit BDC in the timer gate control register (TGCRA) and set the feedback signal input source and output chopping or gate signal direct output.

[4] Set MTU3.TCNT and MTU4.TCNT to 0x0000.

[5] MTU3.TGRA is the cycle register. Set the waveform cycle value in MTU3.TGRA. Set the transition timing of the PWM output waveforms in MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB. Set times within the compare match range of MTU3.TCNT.

X = MTU3.TGRA (X: Set value)

[6] Enable or disable toggle output synchronized with the PMW cycle using bit PSYE in TOCR1A, and set the PWM output level with bits OLSP and OLSN. When specifying the PWM output level by using TOLBRA as a buffer for TOCR2A, see Figure 17.4.

[7] Set bits MD[3:0] in MTU3.TMDR1 to 1000b to select the reset-synchronized PWM mode. In addition, specify the buffer operation of the MTU3.TGRA (BFA = 1) and MTU3.TGRB (BFB = 1) when necessary. Set the MTU4.TMDR1 to the initial value (normal operation; BFA = 0 and BFB = 0).

[8] Set the enabling/disabling of the PWM waveform output pin in TOERA.

[9] Set the Port Function Control register (PFCm) and the Port Mode Control registers (PMCm) of the I/O ports.

[10] Set the CST3 bit in the TSTRA to 1 to start the count operation.

Note: The output waveform starts to toggle at the point of MTU3.TCNT = MTU3.TGRA = X by setting X = MTU3.TGRA for example, period = duty ratio.

Figure 17.44 Procedure for selecting reset-synchronized PWM mode

17.3.7.2 Example of Reset-Synchronized PWM Mode Operation

Figure 17.45 shows an example of operation in the reset-synchronized PWM mode.

MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) operate as up-counters. The counters are cleared when a compare match occurs between MTU3.TCNT (MTU6.TCNT) and MTU3.TGRA (MTU6.TGRA), and then begin incrementing from 0x0000. The output from the PWM pins toggles every time a compare match occurs in MTU3.TGRB (MTU6.TGRB), MTU4.TGRA (MTU7.TGRA), and MTU4.TGRB (MTU7.TGRB) and the counters are cleared.

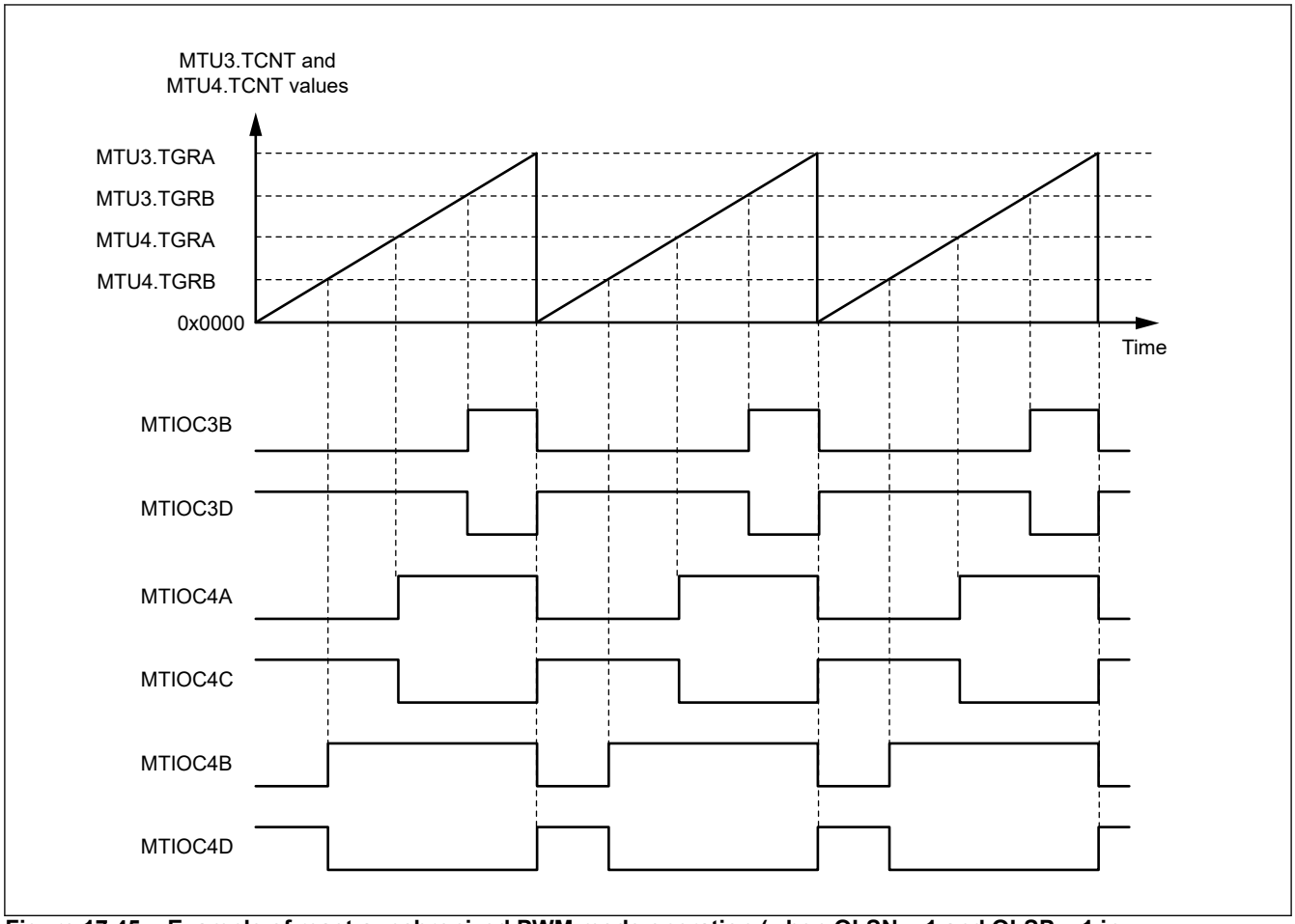


Figure 17.45 Example of reset-synchronized PWM mode operation (when OLSN = 1 and OLSP = 1 in MTU3.TOCR1A and MTU4.TOCR1A)

17.3.8 Complementary PWM Mode

In complementary PWM mode, dead time can be set for PWM waveforms to be output. The dead time is the period during which the upper and lower arm transistors are set to the inactive level in order to prevent short-circuiting of the arms.

Six positive-phase and six negative-phase PWM waveforms (12 phases in total) with dead time can be output by combining MTU3/ MTU4 and MTU6/MTU7. PWM waveforms without dead time can also be output.

In complementary PWM mode, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC7C, and MTIOC7D pins function as PWM output pins, and the MTIOC3A and MTIOC6A pins can be set for toggle output synchronized with the PWM period.

MTU3.TCNT, MTU4.TCNT, MTU6.TCNT, and MTU7.TCNT function as up/down-counters.

Table 17.69 shows the PWM output pins used. Table 17.70 shows the settings of the registers used.

A function to directly cut off the PWM output by using an external signal is supported as a port function.

Table 17.69 Output pins for complementary PWM mode (1 of 2)

Channel	Output pin	Description
MTU3	MTIOC3A	Toggle output synchronized with PWM period (or I/O port)
	MTIOC3B	PWM output pin 1
	MTIOC3C	I/O port*1
	MTIOC3D	PWM output pin 1' (non-overlapping negative-phase waveform of PWM output 1 — PWM output without non-overlapping interval is also available)

Table 17.69 Output pins for complementary PWM mode (2 of 2)

Channel	Output pin	Description
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (non-overlapping negative-phase waveform of PWM output 2 — PWM output without non-overlapping interval is also available)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (non-overlapping negative-phase waveform of PWM output 3 — PWM output without non-overlapping interval is also available)
MTU6	MTIOC6A	Toggle output synchronized with PWM period (or I/O port)
	MTIOC6B	PWM output pin 4
	MTIOC6C	I/O port*1
	MTIOC6D	PWM output pin 4' (non-overlapping negative-phase waveform of PWM output 4 — PWM output without non-overlapping interval is also available)
MTU7	MTIOC7A	PWM output pin 5
	MTIOC7C	PWM output pin 5' (non-overlapping negative-phase waveform of PWM output 5 — PWM output without non-overlapping interval is also available)
	MTIOC7B	PWM output pin 6
	MTIOC7D	PWM output pin 6' (non-overlapping negative-phase waveform of PWM output 6 — PWM output without non-overlapping interval is also available)

Note 1. Avoid setting the MTIOC3C and MTIOC6C pins as timer I/O pins in complementary PWM mode.

Table 17.70 Register settings for complementary PWM mode (1 of 3)

Channel	Counter/ register	Description	Read/write from CPU
MTU3	TCNT	Starts up-counting from the value set in the dead time register	Maskable by TRWERA setting*1
	TGRA	Set MTU3.TCNT upper limit value (1/2 carrier period + dead time)	Maskable by TRWERA setting*1
	TGRB	PWM output 1 compare register	Maskable by TRWERA setting*1
	TGRC	MTU3.TGRA buffer register	Readable/writable
	TGRD	PWM output 1/MTU3.TGRB buffer register	Readable/writable
	TGRE	MTU3.TGRB buffer register B (when double buffer function is used)	Readable/writable
MTU4	TCNT	Starts up-counting after being initialized to 0x0000	Maskable by TRWERA setting*1
	TGRA	PWM output 2 compare register	Maskable by TRWERA setting*1
	TGRB	PWM output 3 compare register	Maskable by TRWERA setting*1
	TGRC	PWM output 2/MTU4.TGRA buffer register	Readable/writable
	TGRD	PWM output 3/MTU4.TGRB buffer register	Readable/writable
	TGRE	MTU4.TGRA buffer register B (when double buffer function is used)	Readable/writable
	TGRF	MTU4.TGRB buffer register B (when double buffer function is used)	Readable/writable

Table 17.70 Register settings for complementary PWM mode (2 of 3)

Channel	Counter/ register	Description	Read/write from CPU
MTU6	TCNT	Starts up-counting from the value set in the dead time register	Maskable by TRWERB setting*2
	TGRA	Set MTU6.TCNT upper limit value (1/2 carrier period + dead time)	Maskable by TRWERB setting*2
	TGRB	PWM output 4 compare register	Maskable by TRWERB setting*2
	TGRC	MTU6.TGRA buffer register	Readable/writable
	TGRD	PWM output 4/MTU6.TGRB buffer register	Readable/writable
	TGRE	MTU6.TGRB buffer register B (when double buffer function is used)	Readable/writable
MTU7	TCNT	Starts up-counting after being initialized to 0x0000	Maskable by TRWERB setting*2
	TGRA	PWM output 5 compare register	Maskable by TRWERB setting*2
	TGRB	PWM output 6 compare register	Maskable by TRWERB setting*2
	TGRC	PWM output 5/MTU7.TGRA buffer register	Readable/writable
	TGRD	PWM output 6/MTU7.TGRB buffer register	Readable/writable
	TGRE	MTU7.TGRA buffer register B (when double buffer function is used)	Readable/writable
	TGRF	MTU7.TGRB buffer register B (when double buffer function is used)	Readable/writable
Timer dead time data register A (TDDRA)		Set MTU4.TCNT and MTU3.TCNT offset value (dead time value)	Maskable by TRWERA setting*1
Timer dead time data register B (TDDRB)		Set MTU7.TCNT and MTU6.TCNT offset value (dead time value)	Maskable by TRWERB setting*2
Timer cycle data register A (TCDRA)		Set MTU4.TCNT upper limit value (1/2 carrier period)	Maskable by TRWERA setting*1
Timer cycle data register B (TCDRB)		Set MTU7.TCNT upper limit value (1/2 carrier period)	Maskable by TRWERB setting*2
Timer cycle buffer register A (TCBRA)		TCDRA buffer register	Readable/writable
Timer cycle buffer register B (TCBRB)		TCDRB buffer register	Readable/writable
Timer subcounter A (TCNTSA)		Subcounter A for dead time generation	Read-only
Timer subcounter B (TCNTSB)		Subcounter B for dead time generation	Read-only
Temporary register 1A (TEMP1A)		PWM output 1/MTU3.TGRB temporary register A	Not readable/writable
Temporary register 1B (TEMP1B)		PWM output 1/MTU3.TGRB temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 2A (TEMP2A)		PWM output 2/MTU4.TGRA temporary register A	Not readable/writable
Temporary register 2B (TEMP2B)		PWM output 2/MTU4.TGRA temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 3A (TEMP3A)		PWM output 3/MTU4.TGRB temporary register A	Not readable/writable
Temporary register 3B (TEMP3B)		PWM output 3/MTU4.TGRB temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 4A (TEMP4A)		PWM output 4/MTU6.TGRB temporary register A	Not readable/writable

Table 17.70 Register settings for complementary PWM mode (3 of 3)

Channel	Counter/ register	Description	Read/write from CPU
Temporary register 4B (TEMP4B)		PWM output 4/MTU6.TGRB temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 5A (TEMP5A)		PWM output 5/MTU7.TGRA temporary register A	Not readable/writable
Temporary register 5B (TEMP5B)		PWM output 5/MTU7.TGRA temporary register B (when double buffer function is used)	Not readable/writable
Temporary register 6A (TEMP6A)		PWM output 6/MTU7.TGRB temporary register A	Not readable/writable
Temporary register 6B (TEMP6B)		PWM output 6/MTU7.TGRB temporary register B (when double buffer function is used)	Not readable/writable

Note 1. Access can be enabled or disabled according to the setting in TRWERA (timer read/write enable register A).
Note 2. Access can be enabled or disabled according to the setting in TRWERB (timer read/write enable register B).

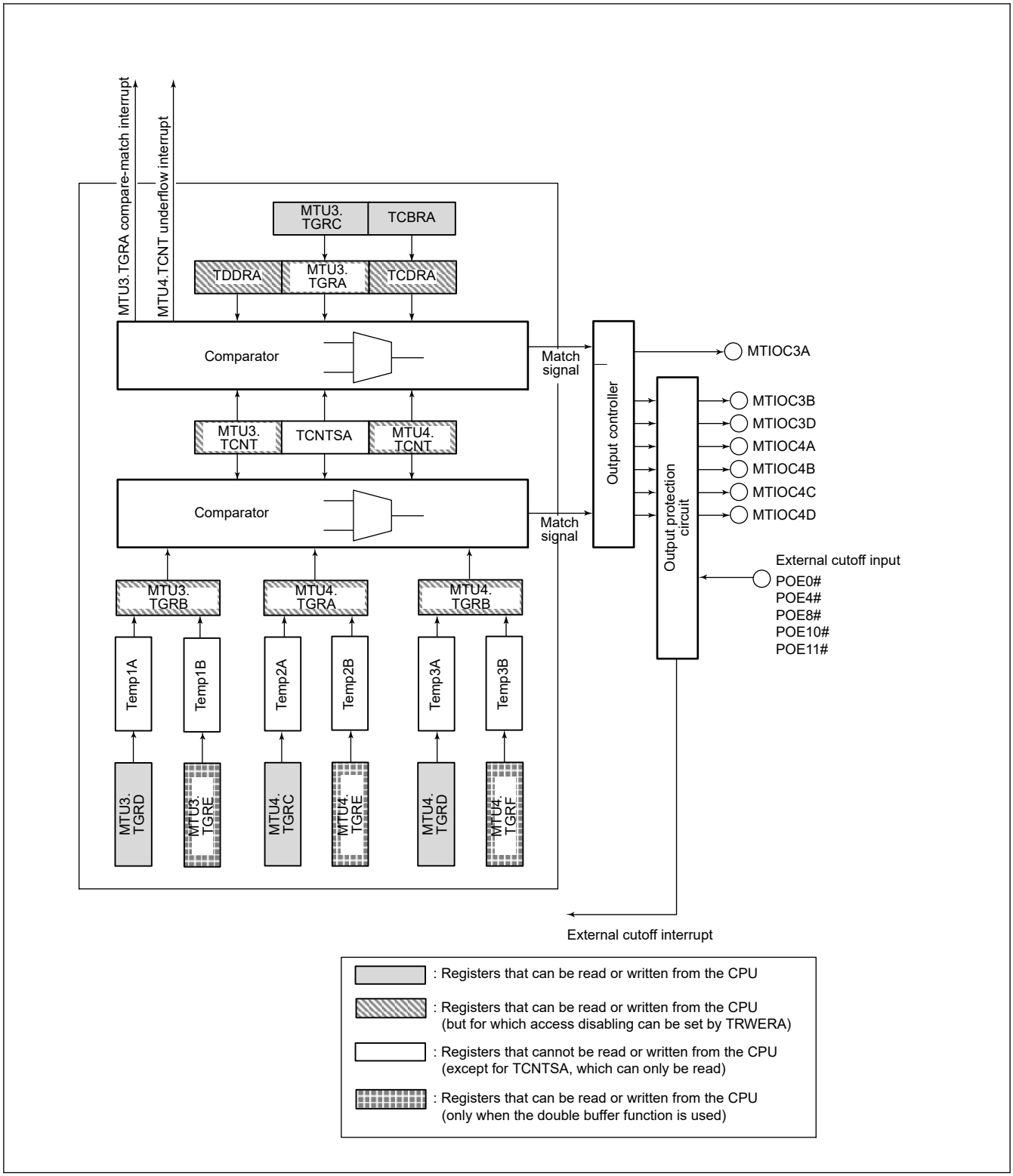


Figure 17.46 Block diagram of MTU3 and MTU4 in complementary PWM mode

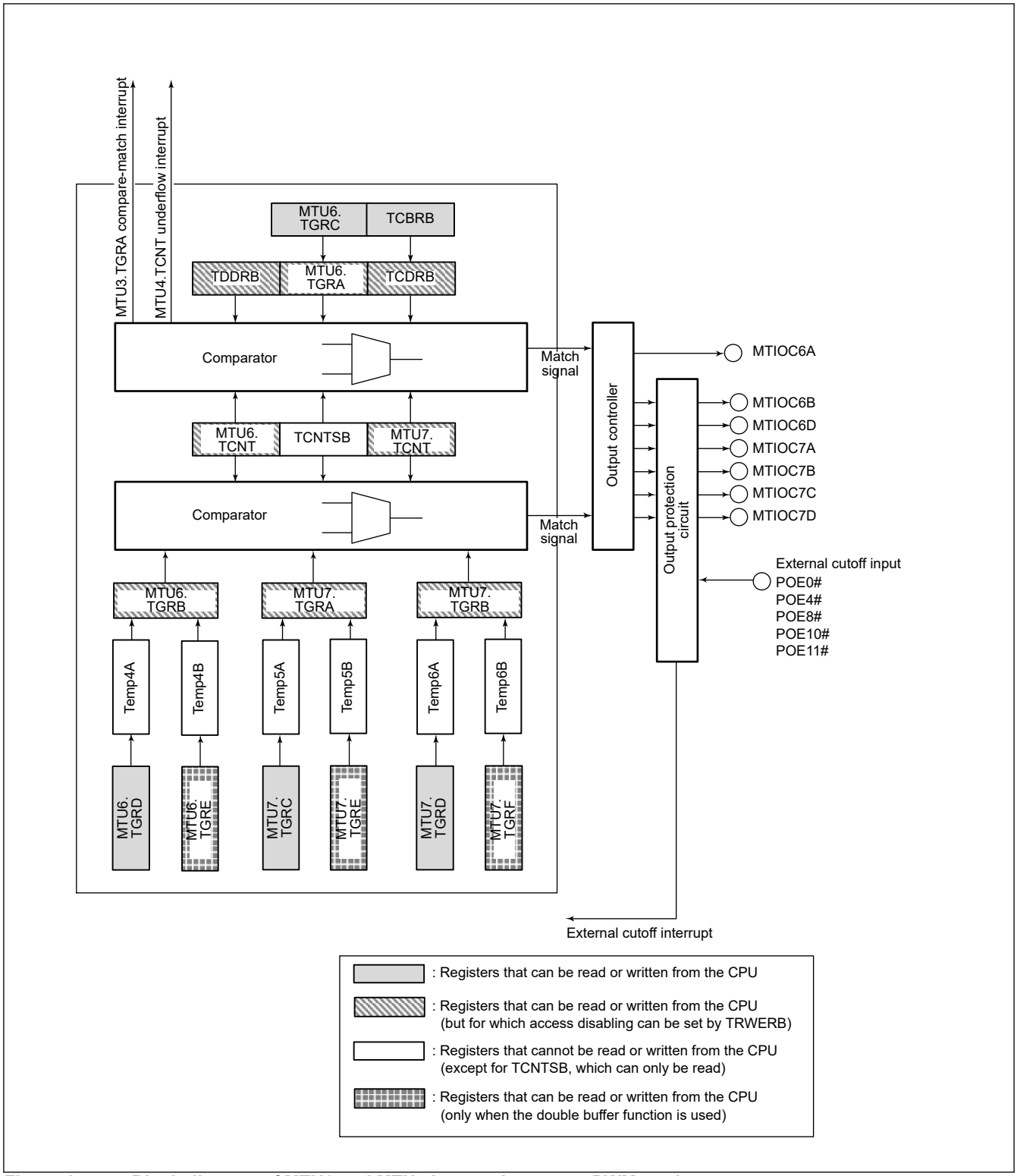
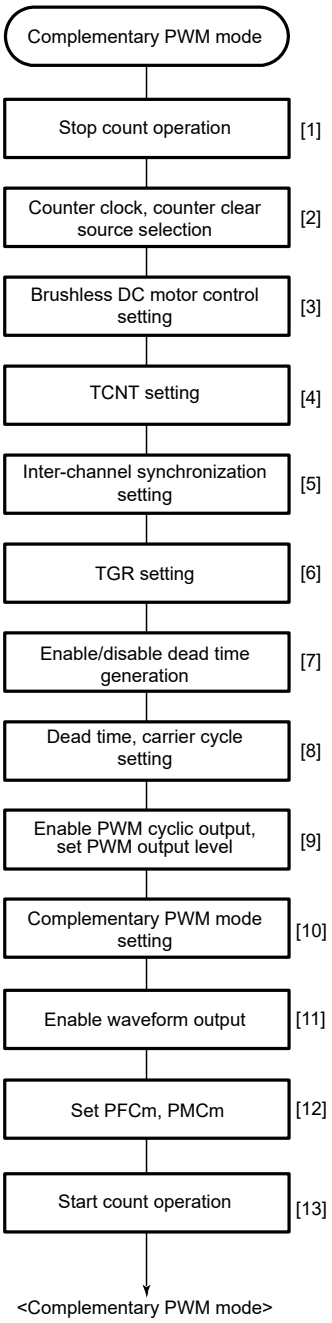


Figure 17.47 Block diagram of MTU6 and MTU7 in complementary PWM mode

17.3.8.1 Example of Complementary PWM Mode Setting Procedure

Figure 17.48 shows an example of the complementary PWM mode setting procedure.



[1] Clear bits CST3 and CST4 in the Timer Start register (TSTRA) to 0 to stop timer counter (TCNT) operation. Specify complementary PWM mode while MTU3.TCNT and MTU4.TCNT are stopped.

[2] Set the same counter clock and clock edge for MTU3 and MTU4 with bits TPSC[2:0] and bits CKEG[1:0] in the Timer Control Register (TCR). Use bits CCLR[2:0] to set synchronous clearing only when restarting by synchronous clearing with another channel during complementary PWM mode operation.

[3] When performing brushless DC motor control, set bit BDC in the Timer Gate Control register (TGCRA) and set the feedback signal input source and output chopping or gate signal direct output.

[4] Set the dead time in MTU3.TCNT. Set MTU4.TCNT to 0x0000.

[5] Set only when restarting by synchronous clearing with another channel during complementary PWM mode operation. In this case, synchronize the channel generating the synchronous clear with MTU3 and MTU4 using the Timer Synchronous register (TSYRA).

[6] Set the output PWM duty cycles in the Duty registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) and Buffer registers (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD). Be sure to set the same value in each pair of Duty register and Buffer register.
Only when using the double buffer function, set the output PWM duty value - 1 in the Buffer registers (MTU3.TGRE, MTU4.TGRE, MTU4.TGRF).

[7] This setting is necessary only when no dead time should be generated. Make appropriate settings in the Timer Dead Time Enable register (TDERA) so that no dead time is generated.

[8] Set the dead time in the Timer Dead Time register (TDDRA), 1/2 the carrier cycle in the Timer Cycle Data register (TCDRA) and Timer Cycle Buffer register (TCBRA), and 1/2 the carrier cycle plus the dead time in MTU3.
TGRA and MTU3.TGRC. When no dead time generation is selected, set 1 in TDDRA and 1/2 the carrier cycle + 1 in MTU3.TGRA and MTU3.TGRC.

[9] Enable or disable toggle output synchronized with the PWM cycle using bit PSYE in the timer output control register 1 (TOCR1A), and set the PWM output level with bits OLSP and OLSN. When specifying the PWM output level by using TOLBRA as a buffer for TOCR2A, see Figure 17.4.

[10] Select complementary PWM mode in the Timer Mode register 1 (MTU3.TMDR1). In addition, specify the buffer operation of the MTU3.TGRA (BFA = 1) and MTU3.TGRB (BFB = 1) when necessary. Enable the double buffer operation with the TMDR2 register only when the double buffer function is used.

[11] Set enabling/disabling of PWM waveform output pin output in the Timer Output Master Enable register (TOERA).

[12] Set the Port Function Control register (PFCm) and Port Mode Control registers (PMCm) of the I/O ports.

[13] Set bits CST3 and CST4 in TSTRA to 1 simultaneously to start the count operation.

Figure 17.48 Example of complementary PWM mode setting procedure

17.3.8.2 Outline of Complementary PWM Mode Operation

In complementary PWM mode, six phases (three positive and three negative) PWM waveforms can be output. Figure 17.49 shows counter operation in complementary PWM mode (MTU3 and MTU4), and Figure 17.50 shows an example of operation in complementary PWM mode.

(1) Counter Operation

In complementary PWM mode, three counters—MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) —in each unit perform up-/down-count operations.

MTU3.TCNT (MTU6.TCNT) is automatically initialized to the value set in TDDRA (TDDRB) when complementary PWM mode is selected and the CST bit in TSTRA (TSTRB) is 0. When the CST bit is set to 1, MTU3.TCNT (MTU6.TCNT)

counts up to the value set in MTU3.TGRA (MTU6.TGRA), then switches to down-counting when it matches MTU3.TGRA (MTU6.TGRA). When the MTU4.TCNT (MTU7.TCNT) value matches 0x0000, MTU3.TCNT (MTU6.TCNT) switches to up-counting, and the operation is repeated in this way.

MTU4.TCNT (MTU7.TCNT) should be initialized to 0x0000 after a reset. When the CST4 bit is set to 1, MTU4.TCNT (MTU7.TCNT) counts up in synchronization with MTU3.TCNT (MTU6.TCNT), and switches to down-counting when MTU3.TCNT (MTU6.TCNT) matches MTU3.TGRA (MTU6.TGRA). On reaching 0x0000, MTU4.TCNT (MTU7.TCNT) switches to up-counting, and the operation is repeated in this way.

MTU4.TCNT (MTU7.TCNT) should be initialized to 0x0000 after a reset. When the CST4 bit is set to 1, MTU4.TCNT (MTU7.TCNT) counts up in synchronization with MTU3.TCNT (MTU6.TCNT), and switches to down-counting when MTU3.TCNT (MTU6.TCNT) matches MTU3.TGRA (MTU6.TGRA).

On reaching 0x0000, MTU4.TCNT (MTU7.TCNT) switches to up-counting, and the operation is repeated in this way. TCNTSA (TCNTSB) is a read-only counter. It does not need to be initialized after a reset.

In counting up by MTU3.TCNT and MTU4.TCNT (or MTU6.TCNT and MTU7.TCNT), MTU3.TCNT (or MTU6.TCNT) starts counting up when it matches TCDRA (or TCDRB) and switches to counting down when it matches MTU3.TGRA (or MTU6.TGRA). Furthermore, when MTU4.TCNT (or MTU7.TCNT) matches TDDRA (or TDDRB), TCNTSA (or TCNTSB) is set to the value in MTU3.TGRA (or MTU6.TGRA) and counting is stopped.

When MTU4.TCNT (MTU7.TCNT) matches TDDRA (TDDRB) during down-counting of MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT), TCNTSA (TCNTSB) starts up-counting, and when MTU4.TCNT (MTU7.TCNT) matches 0x0000, the operation switches to down-counting. When MTU3.TCNT (MTU6.TCNT) matches TCDRA (TCDRB), TCNTSA (TCNTSB) becomes 0x0000 and stops counting.

TCNTSA (TCNTSB) is compared with the compare register and temporary register, in which the PWM duty is specified, only during the count operation.

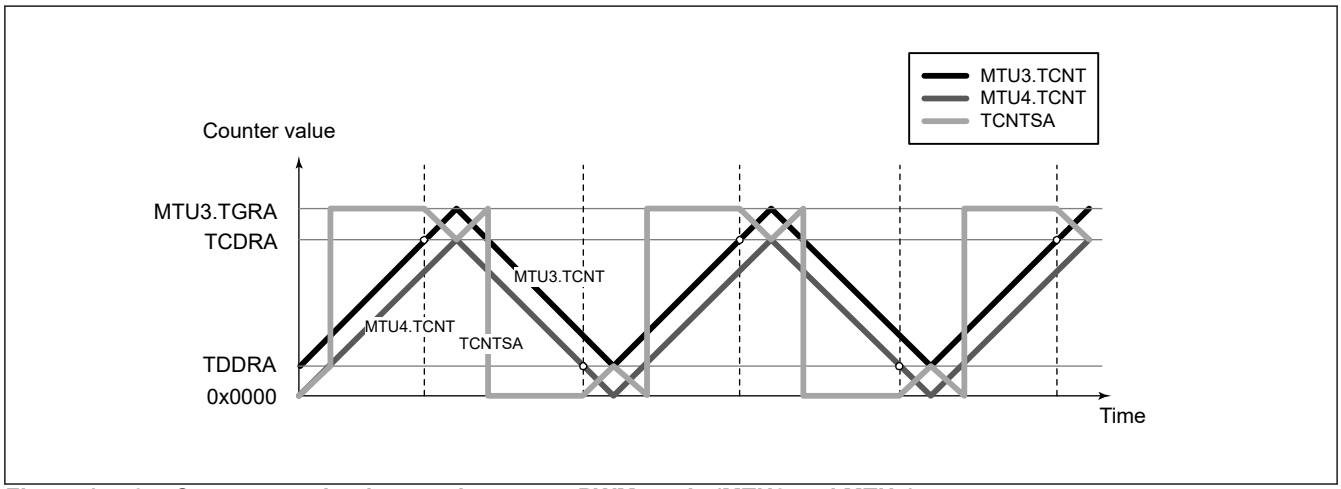


Figure 17.49 Count operation in complementary PWM mode (MTU3 and MTU4)

(2) Register Operation

In complementary PWM mode, nine registers (compare registers, buffer registers, and temporary registers) are used for each unit. Figure 17.50 shows an example of operation in complementary PWM mode (MTU3 and MTU4).

MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB (MTU6.TGRB, MTU7.TGRA, and MTU7.TGRB) are compared with the counters to generate PWM waveforms. When these registers match the counter, the value set in the OLSN and OLSP bits in the timer output control registers (TOCR1A and TOCR1B) is output from the PWM output pin. MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD (MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD) are buffer registers for these compare registers.

When the double buffer function is used, MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF (MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF) are also used as buffer registers B. For details of double buffer operation, see (19)Double Buffer Function in Complementary PWM Mode.

Data in a compare register can be changed by writing new data to the corresponding buffer register. The buffer registers can be read or written at any time.

When modifying data in a buffer register, write to MTU4.TGRD (MTU7.TGRD) last and enable data transfer from the buffer register to a temporary register. At this time, transfer from the TCBRA (TCBRB) register and MTU3.TGRC (MTU6.TGRC) register, which operate as buffer registers for the timer period registers, to temporary registers is also enabled. Data is transferred to all five temporary registers at the same time.

When transfer is enabled in the Ta interval, data written to a buffer register is transferred to the temporary register. The data is not transferred to the temporary register in the Tb1 and Tb2 intervals. Data enabled for transfer in this interval is transferred to the temporary register at the end of this interval.

The value transferred to a temporary register is transferred to the compare register when TCNTSA (TCNTSB) for which the Tb interval ends matches MTU3.TGRA (MTU6.TGRA) while TCNTSA (TCNTSB) is counting up, or 0x0000 while counting down.

The timing for transfer from the temporary register to the compare register can be selected with bits MD[3:0] in the timer mode register 1 (TMDR1). Figure 17.50 shows an example in which the trough is selected for the transfer timing.

In the Tb interval in which data is not transferred to the temporary register (Tb1 in Figure 17.50), the temporary register has the same function as the compare register and is compared with the counter. In this interval, therefore, there are two compare match registers for one output phase; the compare register contains the pre-change data and the temporary register contains new data. In this interval, three counters MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) and two registers (compare register and temporary register) are compared, and PWM output is controlled accordingly.

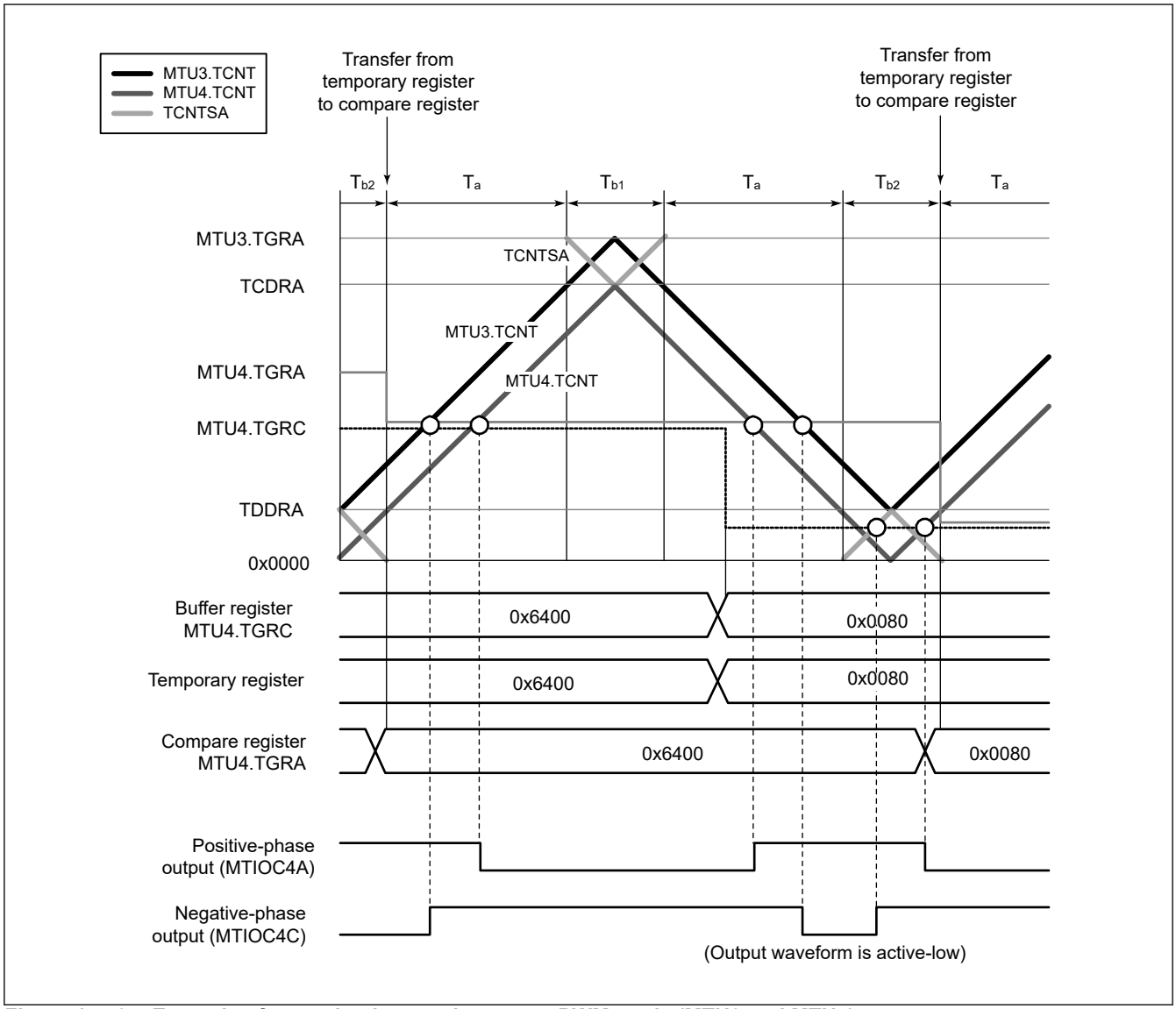


Figure 17.50 Example of operation in complementary PWM mode (MTU3 and MTU4)

(3) Initial Setting

In complementary PWM mode, there are nine registers that require initial setting. In addition, there is a register that specifies whether to generate dead time (it should be used only when dead time generation should be disabled).

Before setting complementary PWM mode with MTU3.TMDR1.MD[3:0] (MTU6.TMDR1.MD[3:0]) bits, initial values should be set in the following registers.

MTU3.TGRC (MTU6.TGRC) operates as the buffer register for MTU3.TGRA (MTU6.TGRA), and should be set with 1/2 the PWM carrier period + dead time Td. The timer cycle buffer register (TCBRA or TCBRB) operates as the buffer register for the timer cycle data register (TCDRA or TCDRB), and should be set with 1/2 the PWM carrier period. Set dead time Td in the timer dead time data register (TDDRA or TDDRB).

When dead time is not needed, the TDER bit in the timer dead time enable register (TDERA or TDERB) should be set to 0, MTU3.TGRC and MTU3.TGRA (MTU6.TGRC and MTU6.TGRA) should be set to 1/2 the PWM carrier period + 1, and TDDRA (TDDRB) should be set to 1.

Set the respective initial PWM duty values in three buffer registers A (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD (MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD)).

Set the respective (initial PWM duty − 1) values in three buffer registers B (MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF (MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF)) only when the double buffer function is used.

The values set in the five buffer registers excluding TDDRA (TDDRB) are transferred to the corresponding compare registers as soon as complementary PWM mode is set.

Set MTU4.TCNT (MTU7.TCNT) to 0x0000 before setting complementary PWM mode.

Table 17.71 Registers and counters requiring initial setting

Register and counter	Setting
TOCR1A, TOCR2A, TOCR1B, TOCR2B	PWM output level
MTU3.TGRC MTU6.TGRC	1/2 PWM carrier period + dead time Td (1/2 PWM carrier period + 1 when dead time generation is disabled by TDERA or TDERB)
TDDRA, TDDRB	Dead time Td (1 when dead time generation is disabled by TDERA or TDERB)
TCBRA, TCBRB	1/2 PWM carrier period
MTU3.TGRD, MTU4.TGRC, MTU4.TGRD MTU6.TGRD, MTU7.TGRC, MTU7.TGRD	Initial PWM duty ratio value for each phase
MTU3.TGRE, MTU4.TGRE, MTU4.TGRF MTU6.TGRE, MTU7.TGRE, MTU7.TGRF	Initial PWM duty ratio − 1 value for each phase (only when double buffer function is used)
MTU4.TCNT MTU7.TCNT	0x0000

Note: The value set in MTU3.TGRC (MTU6.TGRC) should be the sum of 1/2 the PWM carrier period set in TCBRA (TCBRB) and dead time Td set in TDDRA (TDDRB). When dead time generation is disabled by TDERA (TDERB), TGRC should be set to 1/2 the PWM carrier period + 1.

(4) PWM Output Level Setting

In complementary PWM mode, the PWM output level is set with bits OLSN and OLSP in timer output control register 1 (TOCR1A or TOCR1B) or bits OLS1P to OLS3P and OLS1N to OLS3N in timer output control register 2 (TOCR2A or TOCR2B).

The output level can be set for each of the three positive phases and three negative phases of 6-phase output.

Complementary PWM mode should be cleared before setting or changing output levels.

(5) Dead Time Setting

In complementary PWM mode, dead time can be set for PWM output.

The dead time is set in the timer dead time data register (TDDRA or TDDRB). The value set in TDDRA (TDDRB) is used as the MTU3.TCNT (MTU6.TCNT) counter start value and creates a non-overlapping interval between MTU3.TCNT (MTU6.TCNT) and MTU4.TCNT (MTU7.TCNT). Complementary PWM mode should be cleared before changing the contents of TDDRA (TDDRB).

(6) Dead Time Suppressing

Dead time generation is suppressed by setting the TDER bit in the timer dead time enable register (TDERA or TDERB) to 0. TDERA (TDERB) can be set to 0 only when 0 is written to it after reading TDER = 1.

MTU3.TGRA and MTU3.TGRC (MTU6.TGRA and MTU6.TGRC) should be set to 1/2 PWM carrier period + 1 and the timer dead time data register (TDDRA or TDDRB) should be set to 1.

By the above settings, PWM waveforms without dead time can be obtained. Figure 17.51 shows an example of operation without dead time (MTU3 and MTU4).

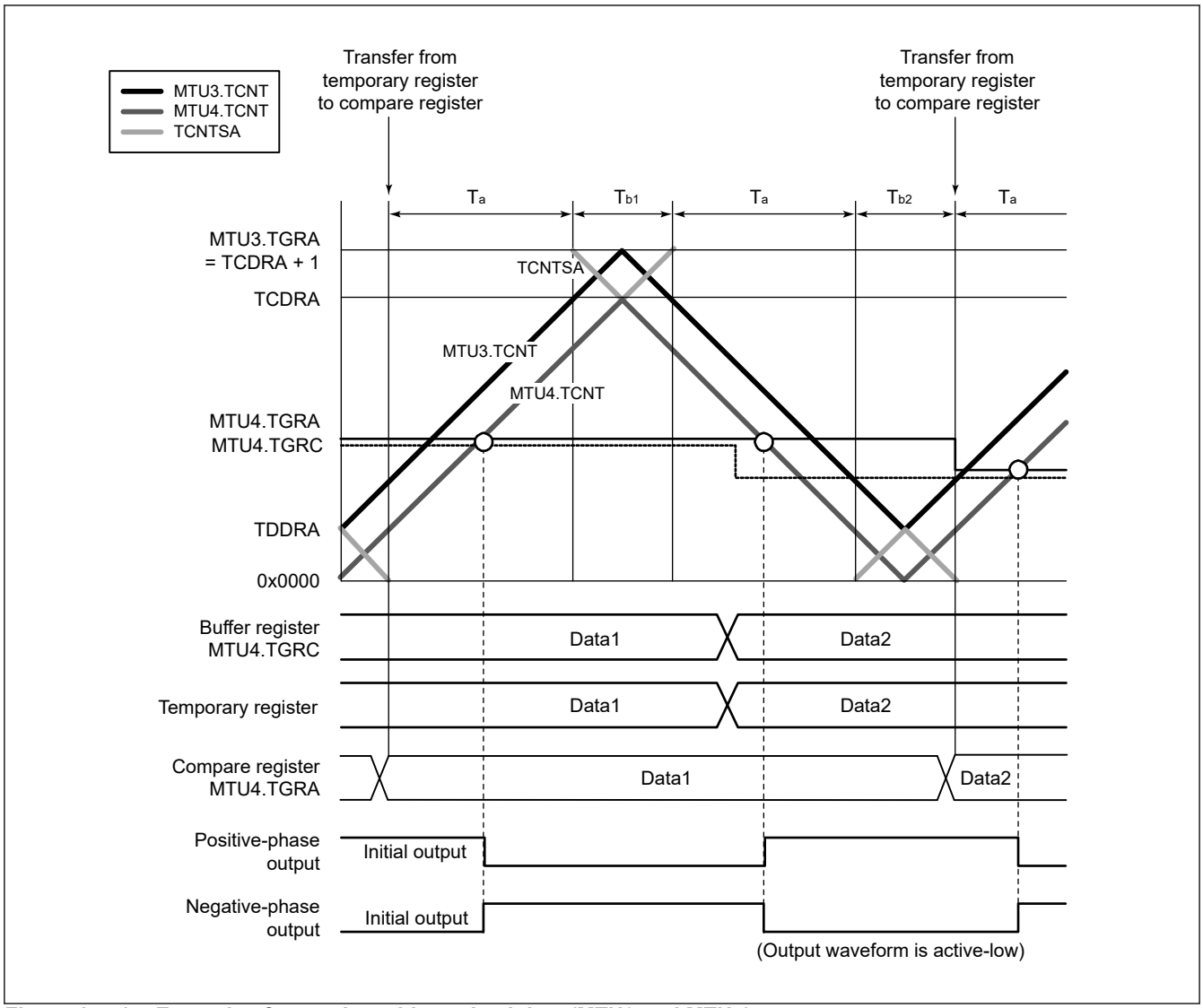


Figure 17.51 Example of operation without dead time (MTU3 and MTU4)

(7) PWM Cycle Setting

In complementary PWM mode, the PWM period is set in two registers—MTU3.TGRA (MTU6.TGRA), in which the MTU3.TCNT (MTU6.TCNT) upper limit value is set, and TCDRA (TCDRB), in which the MTU4.TCNT (MTU7.TCNT) upper limit value is set. The settings should be made so as to achieve the following relationship between these two registers:

With dead time: MTU3.TGRA (MTU6.TGRA) setting = TCDRA (TCDRB) setting + TDDRA (TDDRB) setting

Without dead time: MTU3.TGRA (MTU6.TGRA) setting = TCDRA (TCDRB) setting + 1

In addition, the settings should be made so as to achieve the following relationship between the TCDRA (TCDRB) register and the TDDRA (TDDRB) register:

TCDRA (TCDRB) setting > TDDRA (TDDRB) setting × 2 + 2

The MTU3.TGRA and TCDRA (MTU6.TGRA and TCDRB) settings are made by setting values in buffer registers MTU3.TGRC and TCBRA (MTU6.TGRC and TCBRB). The values set in MTU3.TGRC and TCBRA (MTU6.TGRC and TCBRB) are transferred simultaneously to MTU3.TGRA and TCDRA (MTU6.TGRA and TCDRB) with the transfer timing selected with bits MD[3:0] in the timer mode register 1 (TMDR1).

The new PWM period is reflected from the next cycle when data is updated at the crest, or from the current cycle when updated in the trough. Figure 17.52 shows the operation when the PWM period is updated at the crest.

See the following section, (8)Register Data Updating, for the method of updating the data in each buffer register.

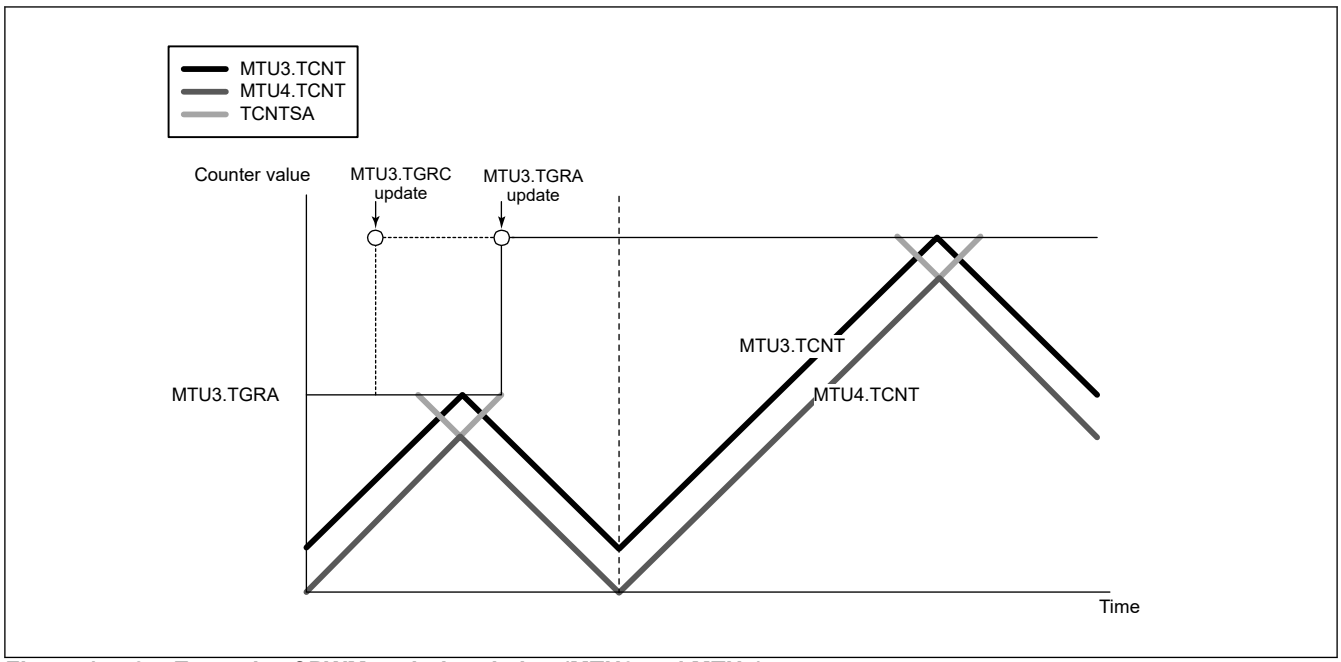


Figure 17.52 Example of PWM period updating (MTU3 and MTU4)

(8) Register Data Updating

In complementary PWM mode, the buffer register is used to update the data in a compare register. The update data can be written to the buffer register at any time. There are five registers (PWM duty and carrier cycle registers) that have buffer registers and can be updated during operation.

There is a temporary register between each of these registers and its buffer register. While subcounter TCNTSA (TCNTSB) is not counting, if buffer register data is updated, the temporary register value also changes. Data is not transferred from buffer registers to temporary registers while TCNTSA (TCNTSB) is counting; in this case, the value written to a buffer register is transferred after TCNTSA (TCNTSB) halts.

The temporary register value is transferred to the compare register at the data update timing set with bits MD[3:0] in the timer mode register 1 (TMDR1). Figure 17.53 shows an example of data updating in complementary PWM mode (MTU3 and MTU4). This example shows the mode in which data is updated at both the counter crest and trough.

When updating buffer register data, be sure to write to MTU4.TGRD (MTU7.TGRD) at the end of the update. Data is transferred from buffer registers to the temporary registers simultaneously for all five registers after the write to MTU4.TGRD (MTU7.TGRD).

Even when not updating all five registers or when not updating the MTU4.TGRD (MTU7.TGRD) data, be sure to write to MTU4.TGRD (MTU7.TGRD) after writing data to the registers to be updated. In this case, the data written to MTU4.TGRD (MTU7.TGRD) should be the same as the data prior to the write operation.

See (19)Double Buffer Function in Complementary PWM Mode, for data updating when the double buffer function is used.

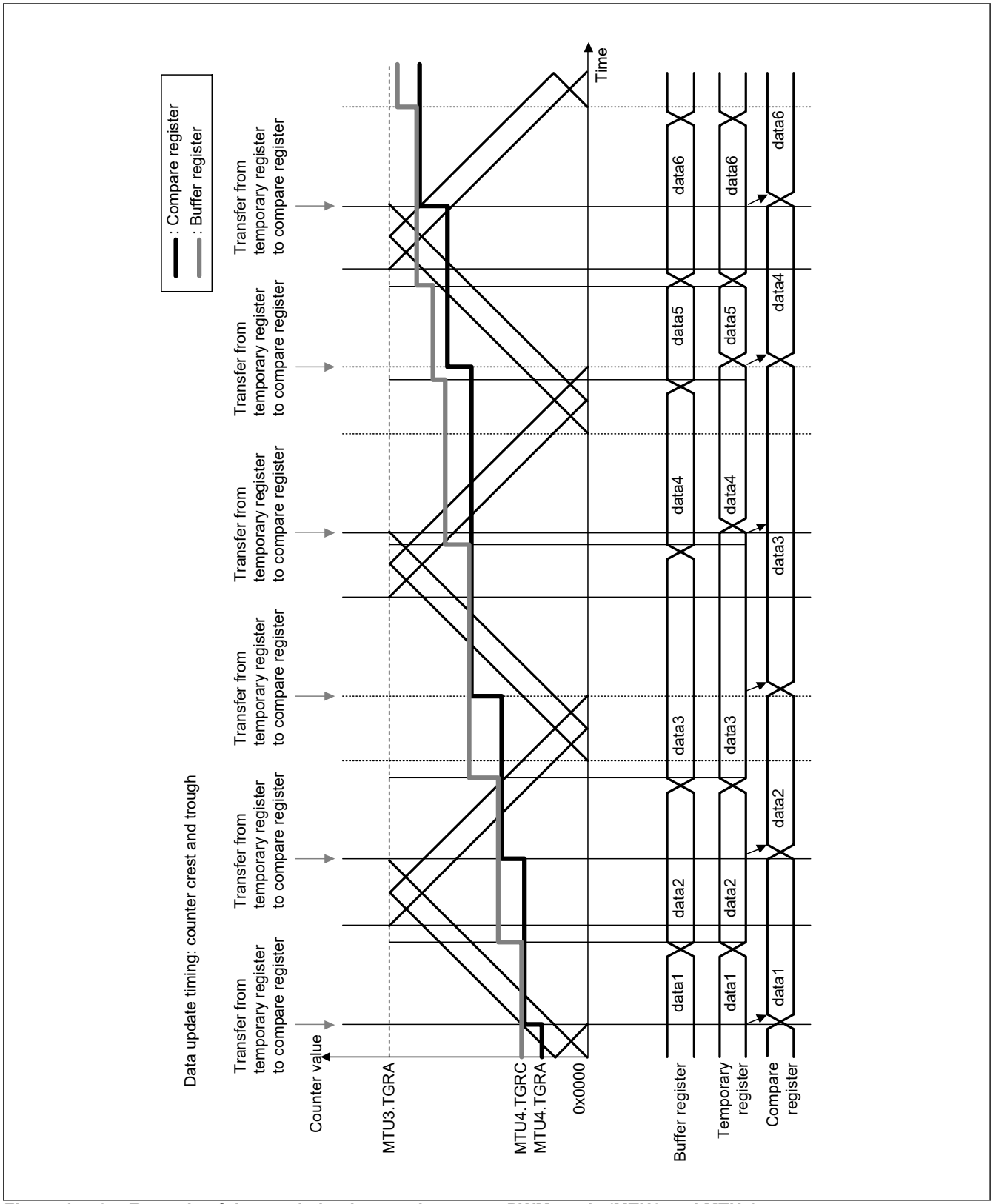


Figure 17.53 Example of data updating in complementary PWM mode (MTU3 and MTU4)

(9) Initial Output in Complementary PWM Mode

In complementary PWM mode, the initial output is determined by the setting of bits OLSN and OLSP in timer output control register 1 (TOCR1A or TOCR1B) or bits OLS1N to OLS3N and OLS1P to OLS3P in timer output control register 2 (TOCR2A or TOCR2B).

This initial output is the non-active level of the PWM pulse and continues from when complementary PWM mode is set with the timer mode register 1 (TMDR1) until MTU4.TCNT (MTU7.TCNT) exceeds the value set in the Timer dead time data register (TDDRA or TDDRB). Figure 17.54 shows an example of the initial output in complementary PWM mode.

An example of the waveform when the initial PWM duty ratio value is smaller than the TDDRA value is shown in Figure 17.55.

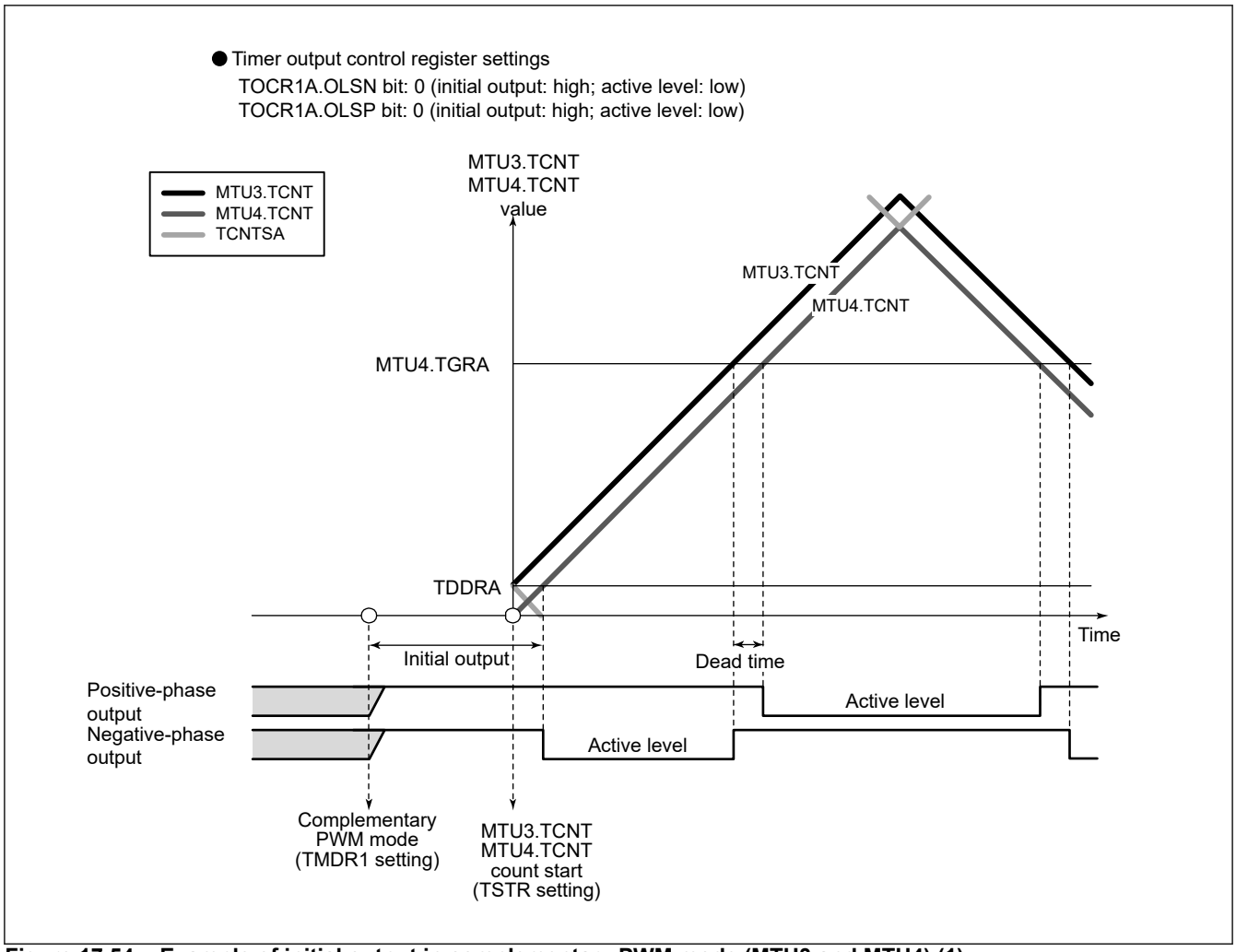


Figure 17.54 Example of initial output in complementary PWM mode (MTU3 and MTU4) (1)

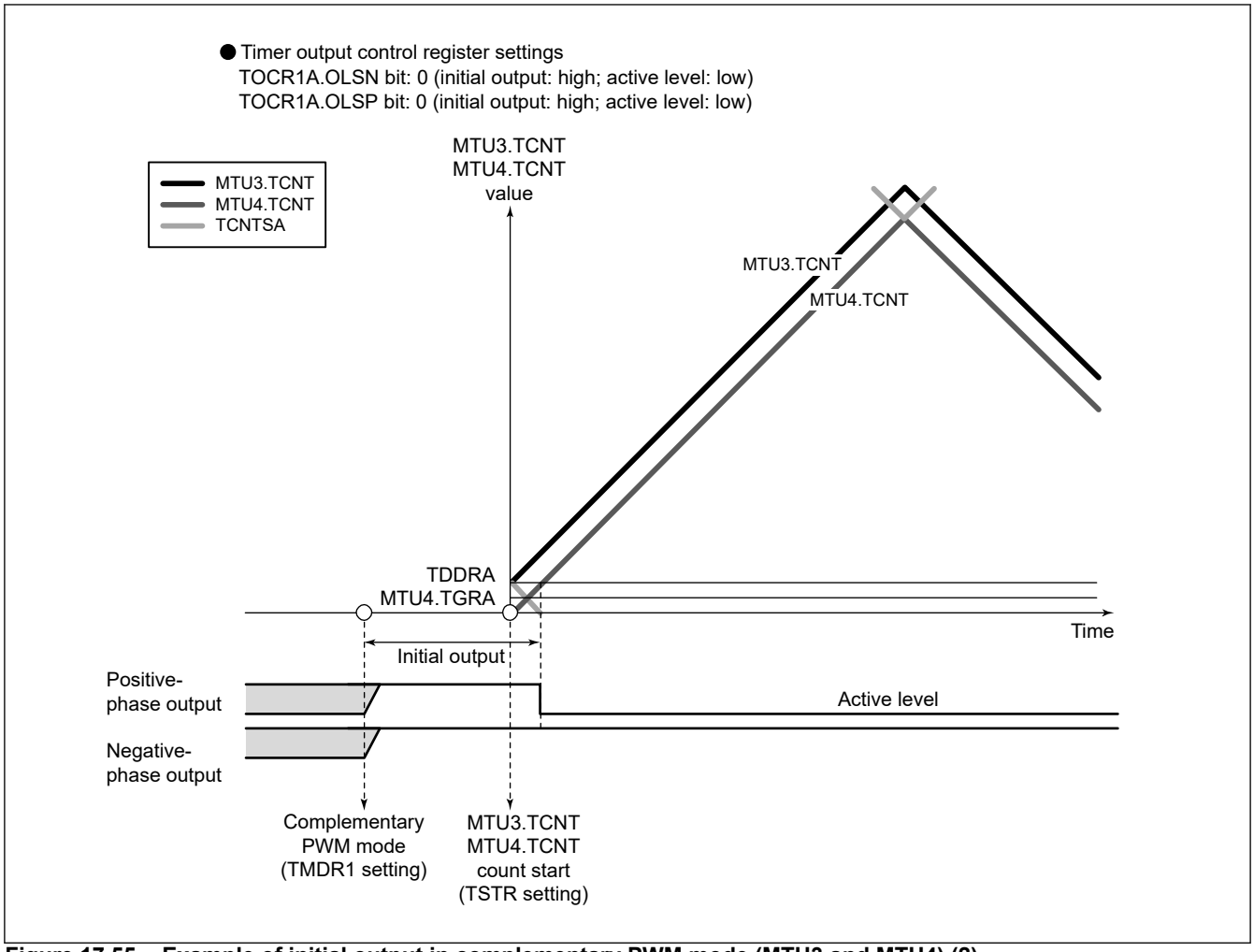


Figure 17.55 Example of initial output in complementary PWM mode (MTU3 and MTU4) (2)

(10) Method for Generating PWM Output in Complementary PWM Mode

In complementary PWM mode, six waveforms of three-phase PWM waveforms can be output. Dead time can be set for PWM waveforms to be output.

A PWM waveform is generated by output of the level selected in the Timer Output Control register in the event of a compare match between a counter and a compare register. While TCNTSA (TCNTSB) is counting, the compare register and temporary register values are simultaneously compared to create consecutive PWM pulses from 0 to 100% duty ratio. The relative timing of turn-on and turn-off compare match occurrence may vary, but the compare match that turns off each phase takes precedence to secure the dead time and ensure that the positive-phase and negative-phase turn-on times do not overlap. Figure 17.56 to Figure 17.58 show examples of waveform generation in complementary PWM mode.

The positive-phase and negative-phase turn-off timing is generated by a compare match with the counter indicated by a solid line, and the turn-on timing is generated by a compare match with the counter indicated by a dotted line, which operates with a delay equal to the dead time behind the counter indicated by a solid line. In the T1 period, compare match a that turns off the negative phase has the highest priority, and compare matches before a are ignored. In the T2 period, compare match c that turns off the positive phase has the highest priority, and compare matches before c are ignored.

In most cases, compare matches occur in the order a → b → c → d (or c → d → a' → b') as shown in Figure 17.56.

If compare matches deviate from the a → b → c → d order, since the time for which the negative phase is off is shorter than twice the dead time, the positive phase is not turned on. If compare matches deviate from the c → d → a' → b' order, since the time for which the positive phase is off is shorter than twice the dead time, the negative phase is not turned on.

As shown in Figure 17.57, if compare match c follows compare match a before compare match b, compare match b is ignored and the negative phase is turned on by compare match d. This is because turning off the positive phase has priority due to the occurrence of compare match c (positive-phase off timing) before compare match b (positive-phase on timing) (consequently, the waveform does not change because the positive phase goes from off to off).

Similarly, in the example in Figure 17.58, compare match a' with new data in the temporary register occurs before compare-match c, but until compare-match c, which turns off the positive phase, other compare matches are ignored. As a result, the negative phase is not turned on.

Thus, in complementary PWM mode, compare matches at turn-off timings take precedence, and turn-on timing compare matches that occur before a turn-off timing compare match are ignored.

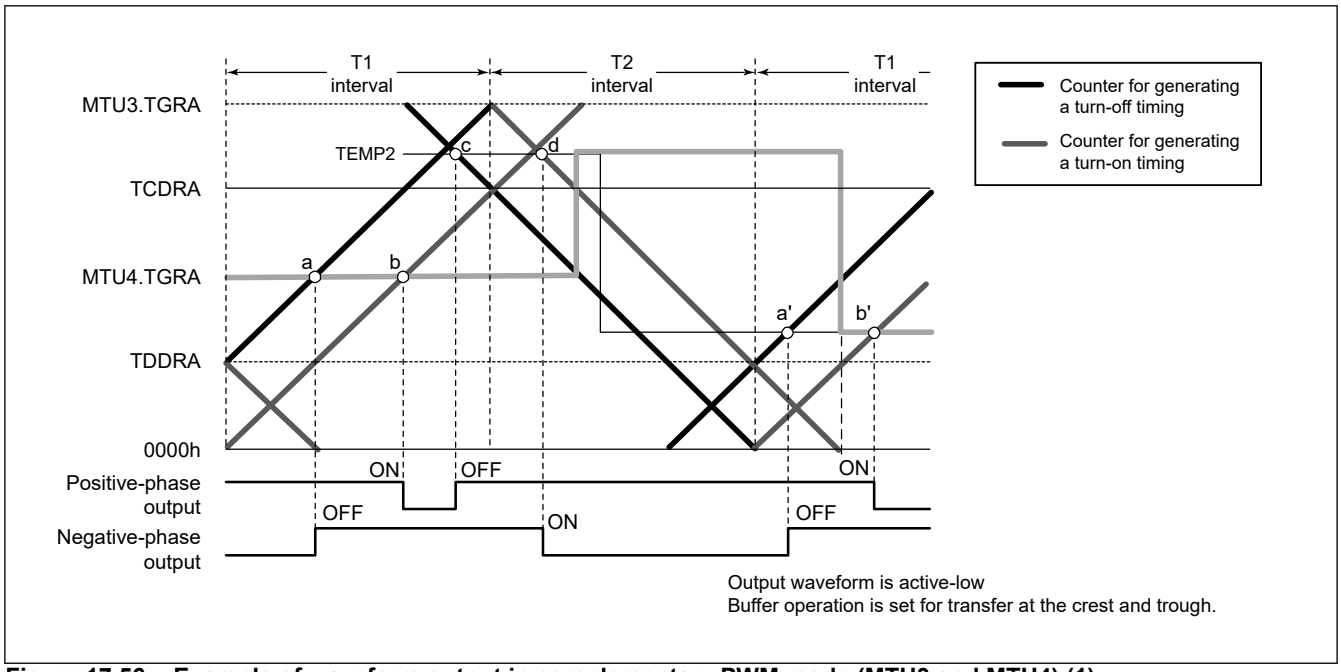


Figure 17.56 Example of waveform output in complementary PWM mode (MTU3 and MTU4) (1)

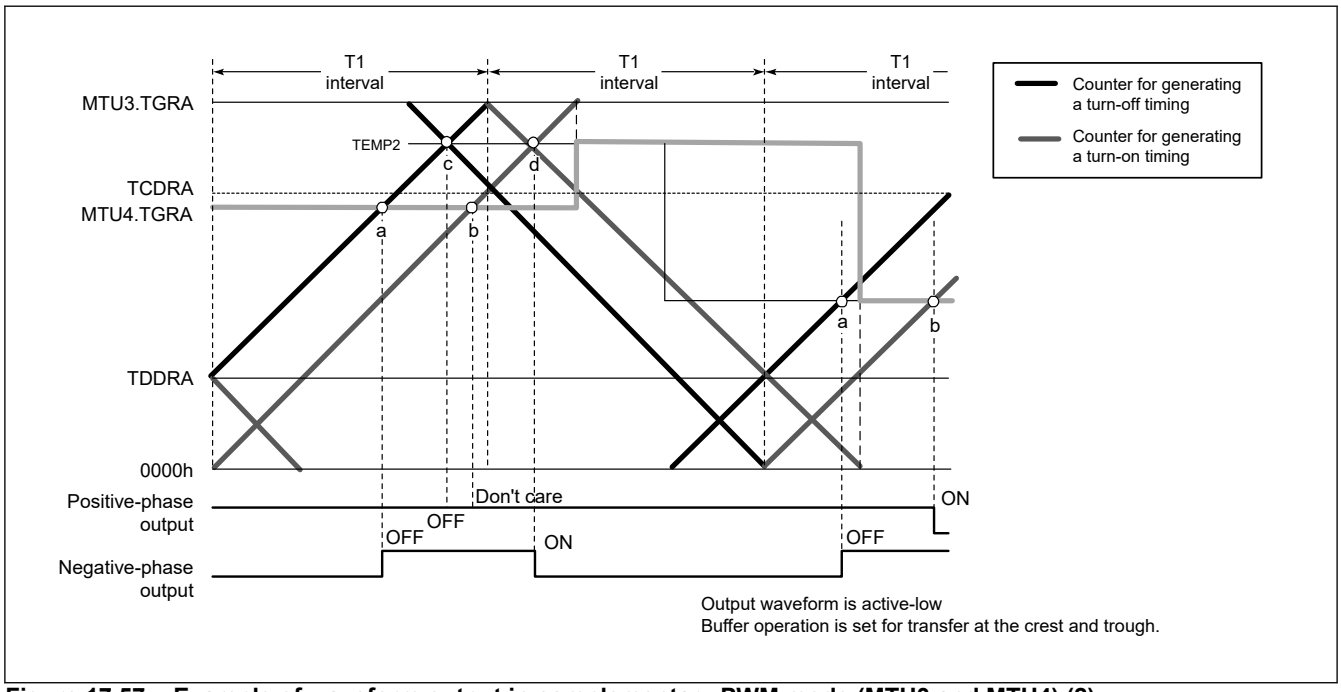


Figure 17.57 Example of waveform output in complementary PWM mode (MTU3 and MTU4) (2)

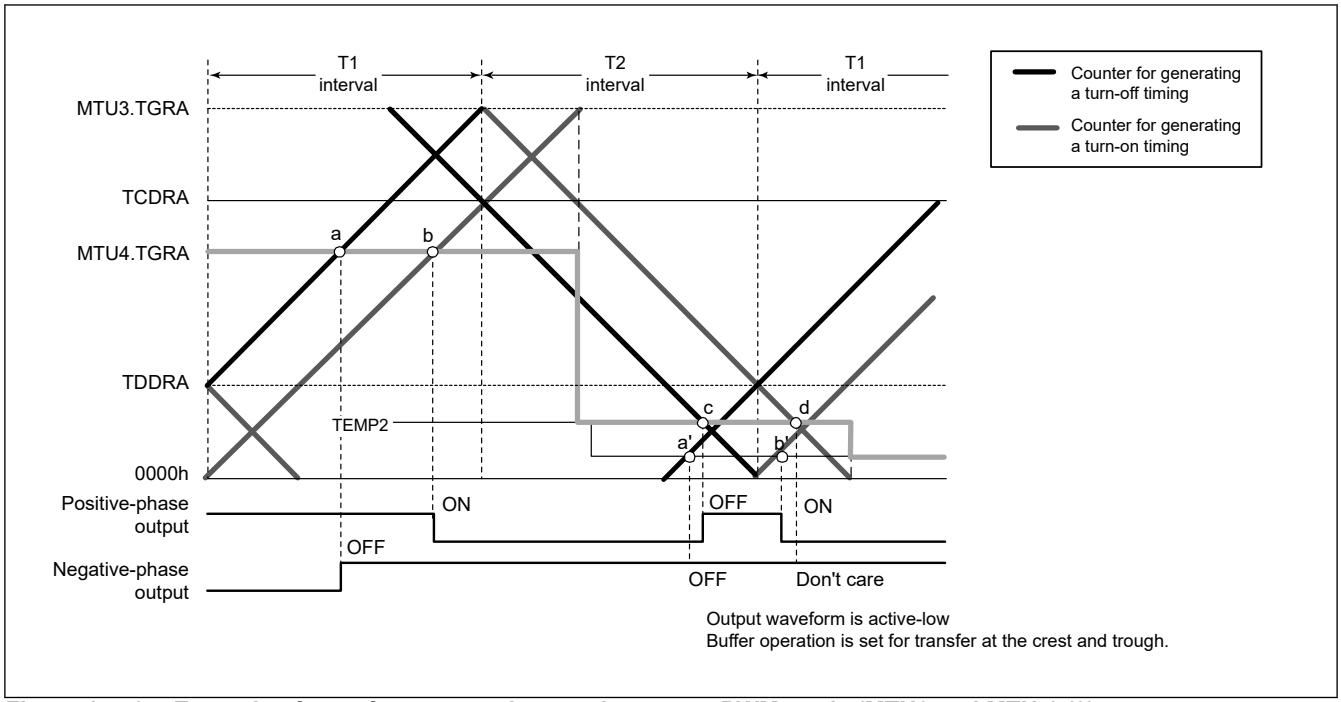


Figure 17.58 Example of waveform output in complementary PWM mode (MTU3 and MTU4) (3)

(11) 0% and 100% Duty Ratio Output in Complementary PWM Mode

In complementary PWM mode, 0% and 100% duty PWM waveforms can be output as required. Figure 17.59 to Figure 17.63 show output examples.

A 100% duty waveform is output when the compare register value is set to 0x0000. The waveform in this case has a positive phase with a 100% on-state. A 0% duty waveform is output when the compare register value is set to the same value as MTU3.TGRA (MTU6.TGRA). The waveform in this case has a positive phase with a 100% off-state.

Turn-on and turn-off compare matches occur simultaneously, but if a turn-on compare match and turn-off compare match for the same phase occur simultaneously, both compare matches are ignored and the waveform does not change.

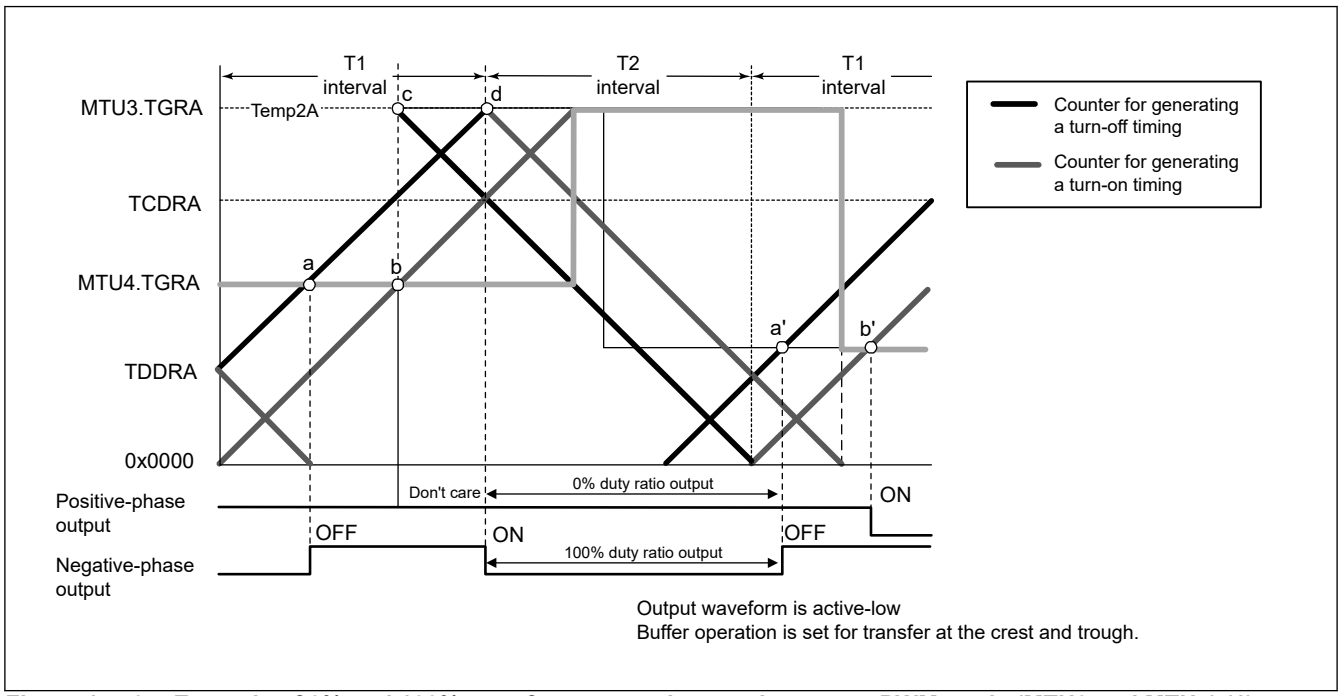


Figure 17.59 Example of 0% and 100% waveform output in complementary PWM mode (MTU3 and MTU4) (1)

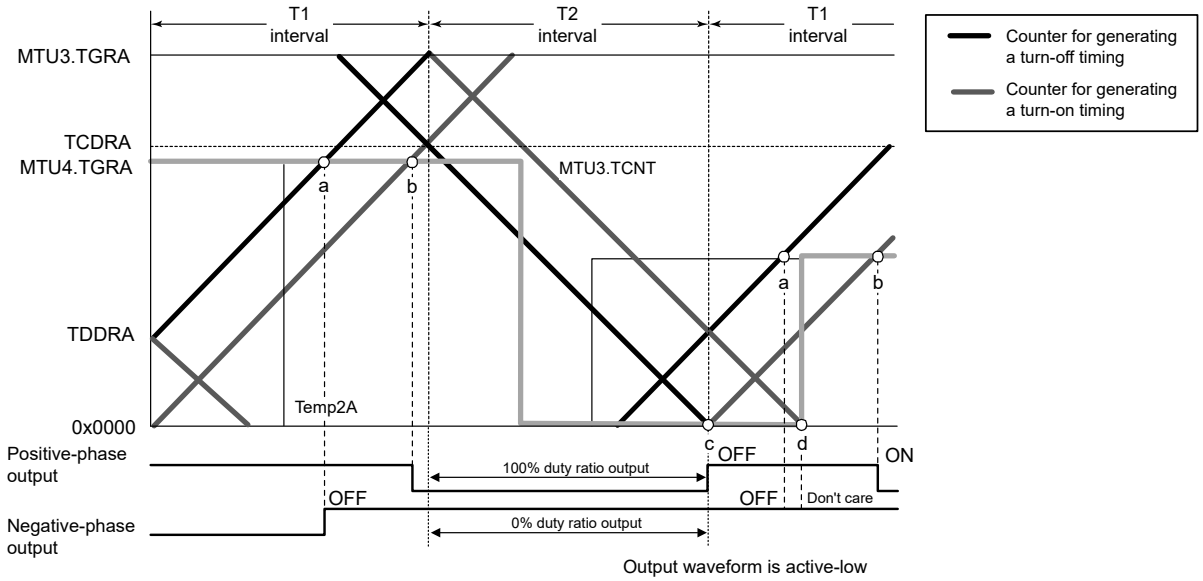


Figure 17.60 Example of 0% and 100% waveform output in complementary PWM mode (MTU3 and MTU4) (2)

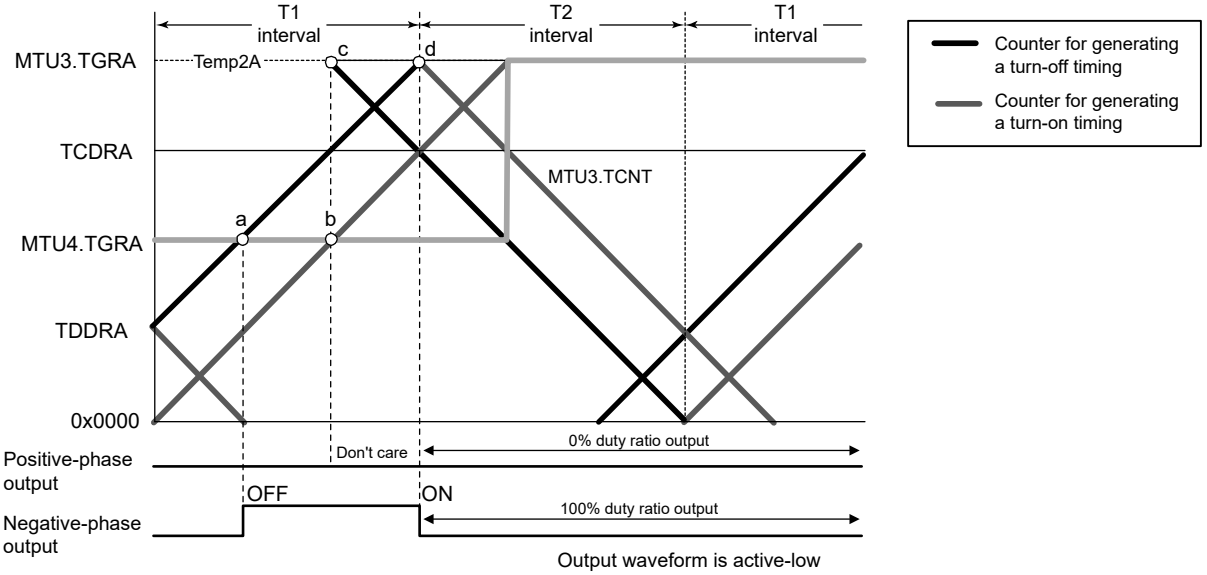


Figure 17.61 Example of 0% and 100% waveform output in complementary PWM mode (MTU3 and MTU4) (3)

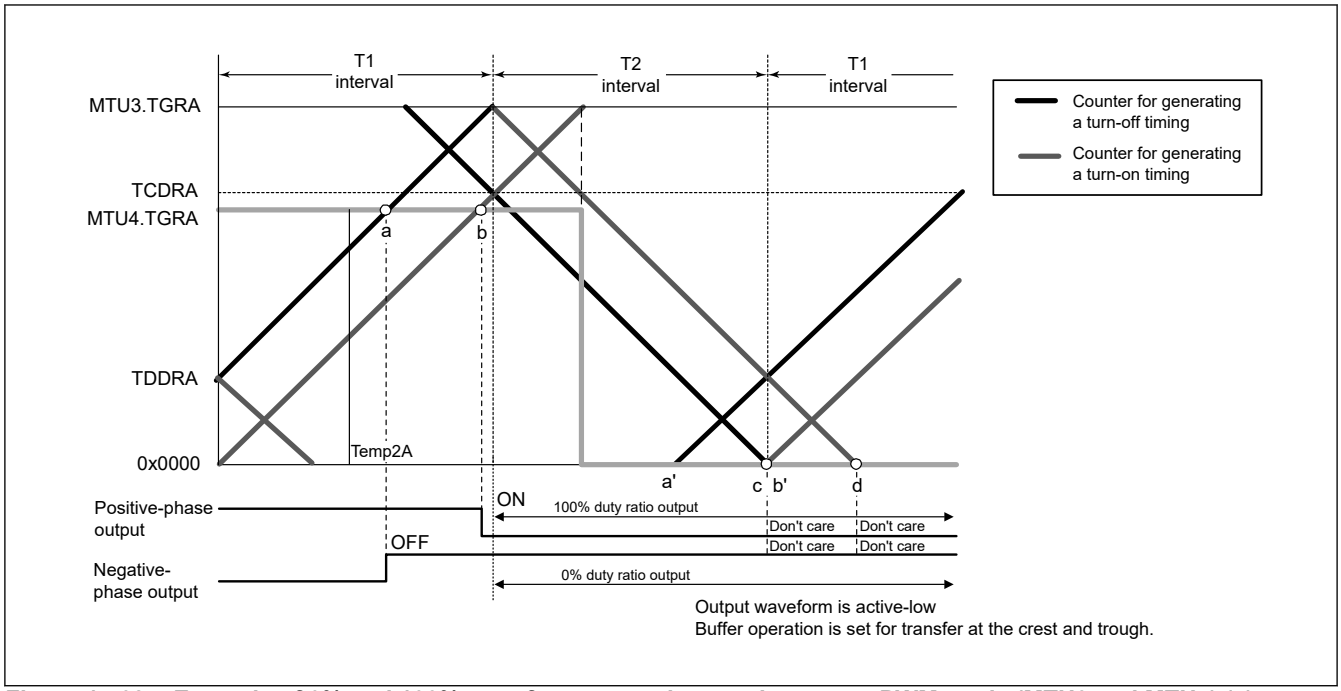


Figure 17.62 Example of 0% and 100% waveform output in complementary PWM mode (MTU3 and MTU4) (4)

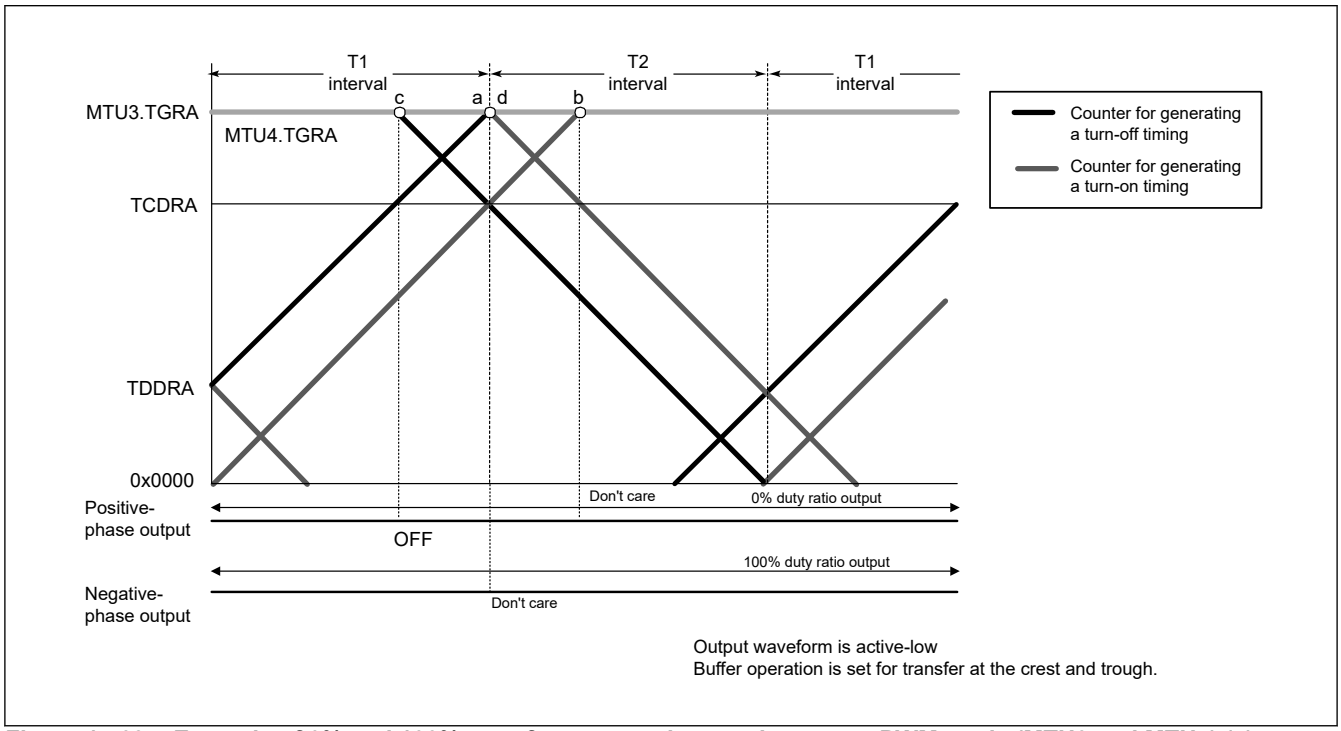


Figure 17.63 Example of 0% and 100% waveform output in complementary PWM mode (MTU3 and MTU4) (5)

(12) Toggle Output Synchronized with PWM Period

In complementary PWM mode, toggle output from the PWM output pin in synchronization with the PWM carrier period can be generated by setting the PSYE bit in the TOCR1A (TOCR1B) register to 1. An example of a toggle output waveform is shown in Figure 17.64.

This output is toggled by a compare match between MTU3.TCNT and MTU3.TGRA (MTU6.TCNT and MTU6.TGRA) and a compare match between MTU4.TCNT (MTU7.TCNT) and 0x0000.

The MTIOC3A (MTIOC6A) pin is assigned for this toggle output. The initial output is high-level.

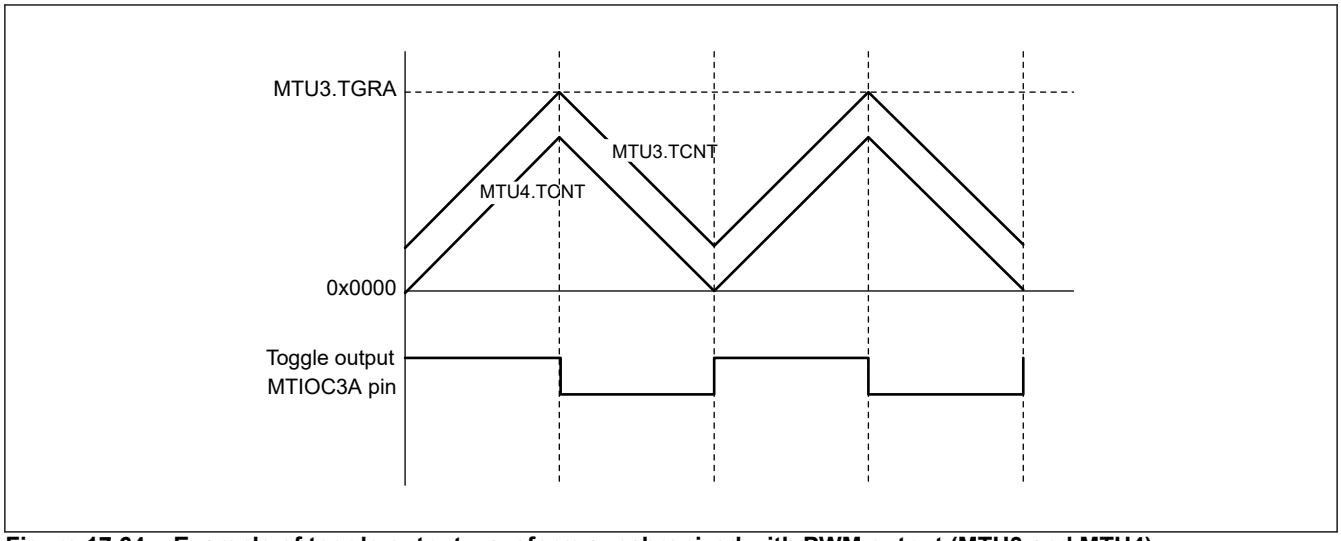


Figure 17.64 Example of toggle output waveform synchronized with PWM output (MTU3 and MTU4)

(13) Counter Clearing by Another Channel

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) can be cleared by another channel when a mode for synchronization with another channel is specified through the timer synchronous register (TSYRA or TSYRB)) and synchronous clearing is selected with bits CCLR[2:0] in the timer control register (TCR).

Figure 17.65 shows an example of this operation.

Use of this function enables a counter to be cleared and restarted through an external signal.

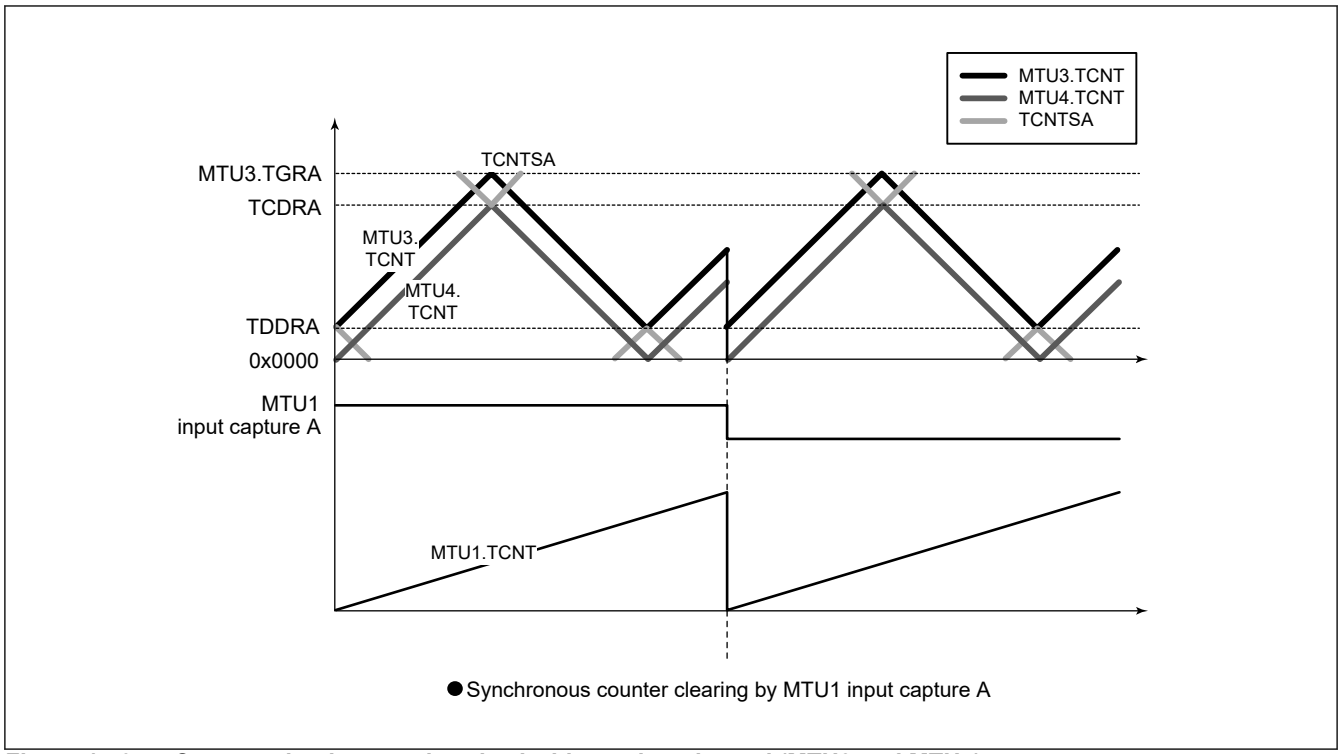


Figure 17.65 Counter clearing synchronized with another channel (MTU3 and MTU4)

(14) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the WRE bit in TWCRA (TWCRB) to 1 suppresses initial output when synchronous counter clearing occurs in the Tb interval at the trough in complementary PWM mode and controls abrupt change in duty cycle at synchronous counter clearing.

Initial output suppression through the WRE bit = 1 is applicable only when synchronous clearing occurs in the Tb interval at the trough as indicated by (10) or (11) in Figure 17.66. When synchronous clearing occurs outside that interval, the initial value specified by the OLSN and OLSP bits in TOCR1A (TOCR1B) is output. Even in the Tb interval at the trough, if synchronous clearing occurs in the initial value output period (indicated by (1) in Figure 17.66) immediately after the counters start operation, initial value output is not suppressed.

This function can be used in both channel combinations of MTU3 and MTU4, and MTU6 and MTU7. In MTU3 and MTU4, synchronous clearing generated in MTU0 to MTU2 can cause counter clearing; in MTU6 and MTU7, compare match or input capture in any of MTU0 to MTU2 can cause counter clearing.

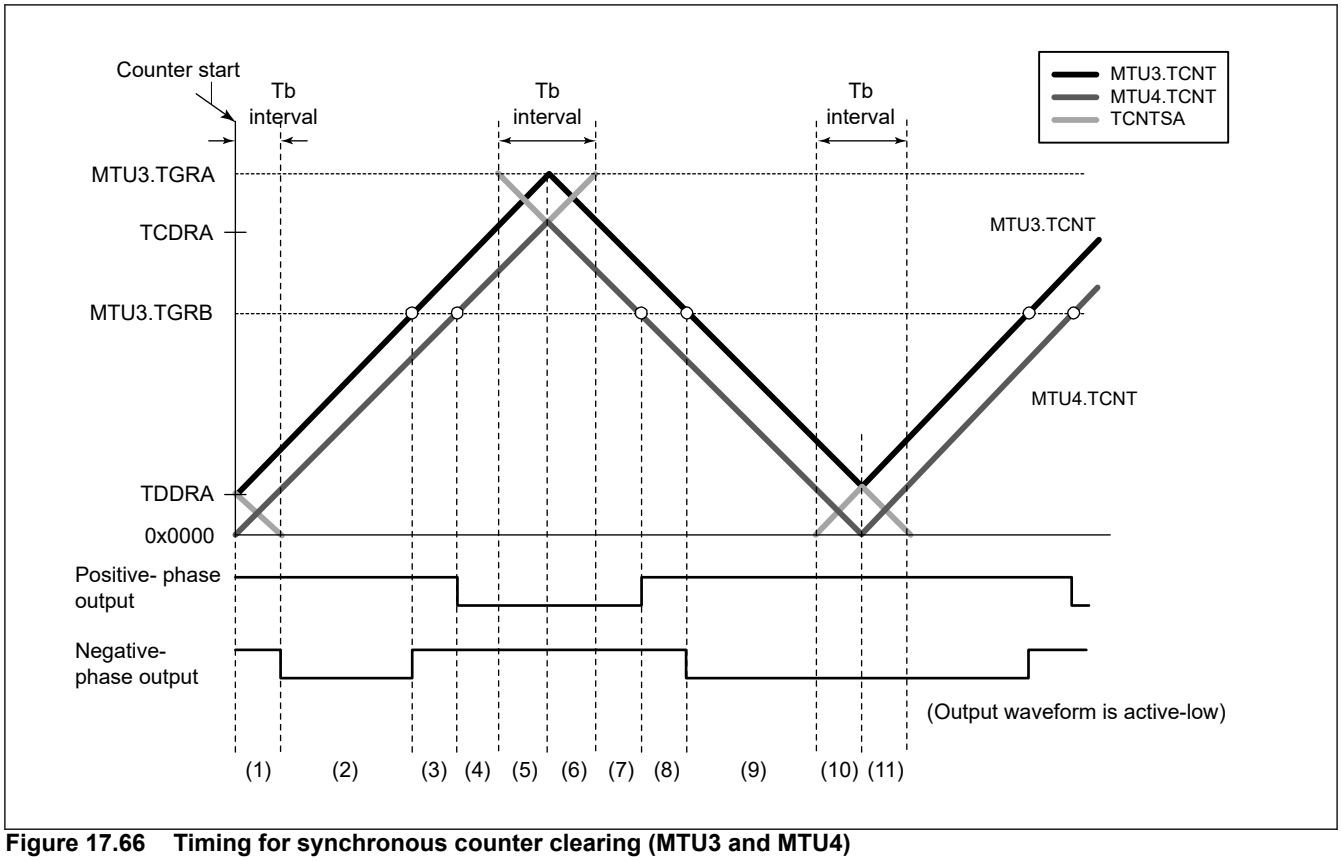


Figure 17.66 Timing for synchronous counter clearing (MTU3 and MTU4)

Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

An example of the procedure for setting output waveform control at synchronous counter clearing in complementary PWM mode is shown in Figure 17.67.

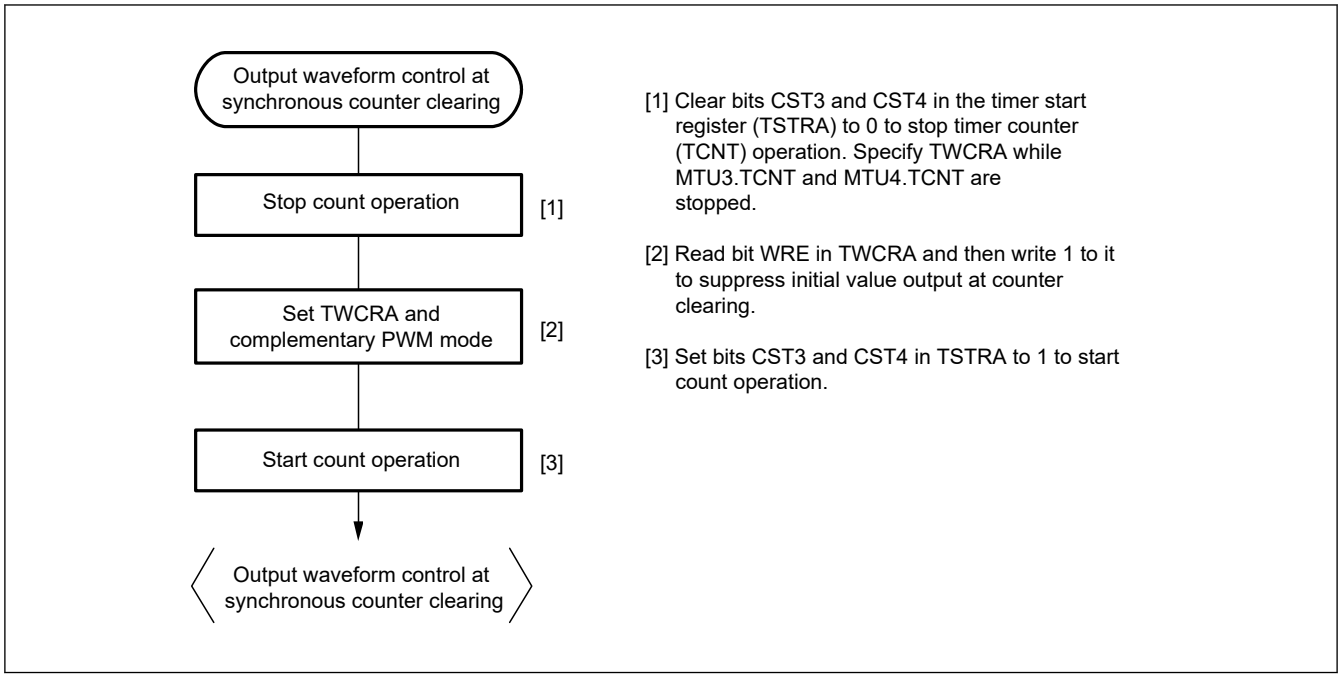
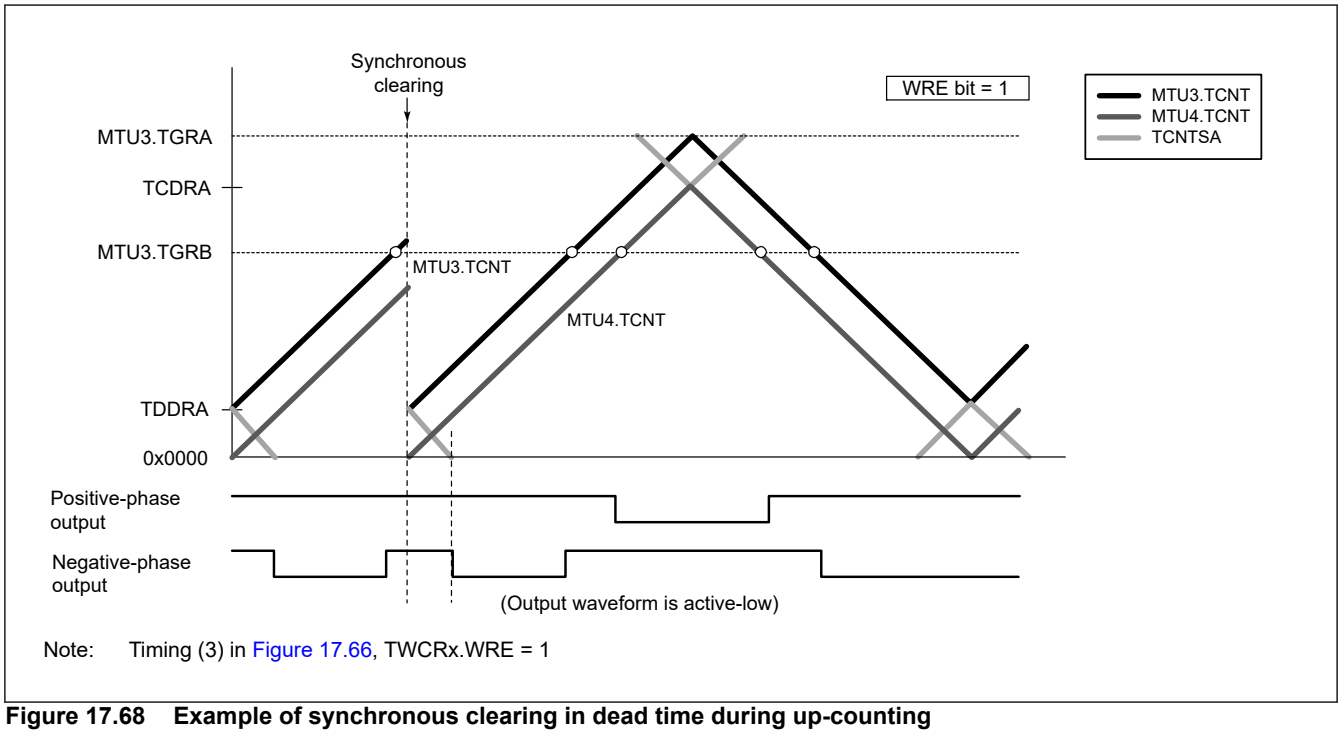


Figure 17.67 Example of procedure for setting output waveform control at synchronous counter clearing in complementary PWM mode (MTU3 and MTU4)

Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

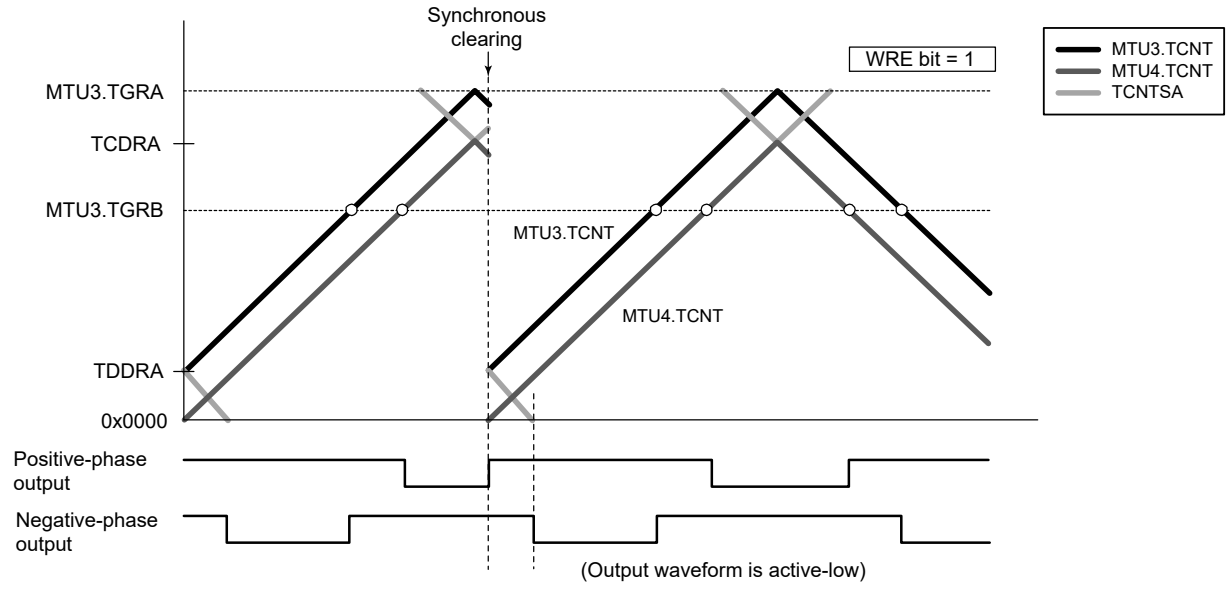
Figure 17.68 to Figure 17.71 show examples of output waveform control in which MTU3 and MTU4 operate in complementary PWM mode and synchronous counter clearing is generated while the WRE bit in TWCRA is set to 1. In the examples shown in Figure 17.68 to Figure 17.71, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 17.66, respectively.

In MTU6 and MTU7, these examples are equivalent to the cases when MTU6 and MTU7 operate in complementary PWM mode and synchronous counter clearing is generated while the SCC bit is set to 0 and the WRE bit is set to 1 in TWCRB.



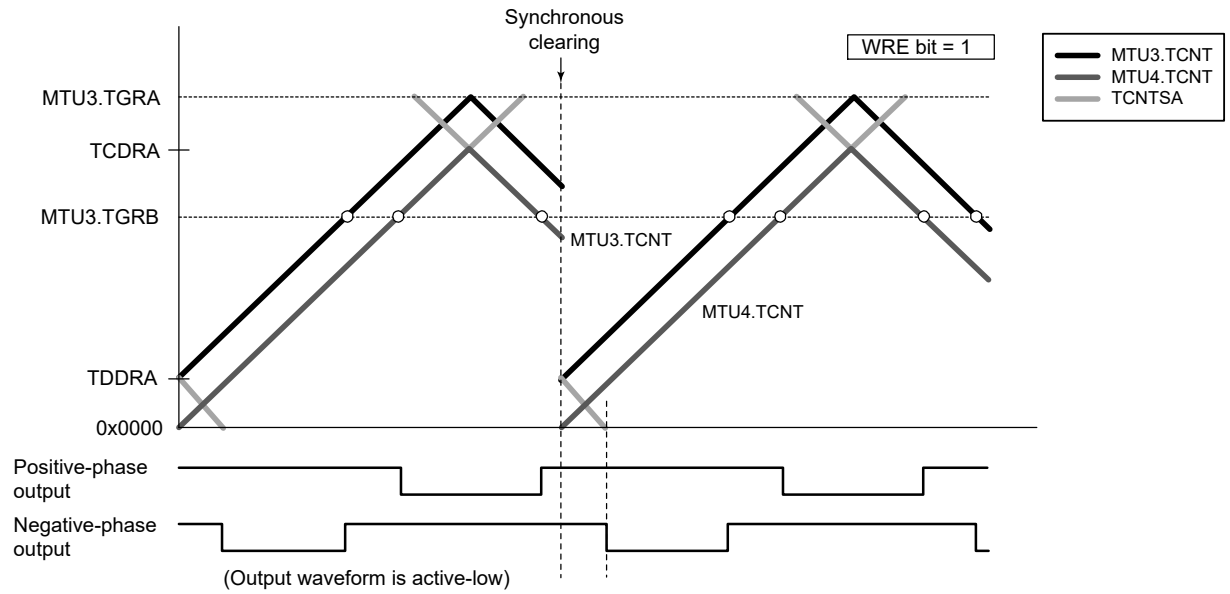
Note: Timing (3) in Figure 17.66, TWCRx.WRE = 1

Figure 17.68 Example of synchronous clearing in dead time during up-counting



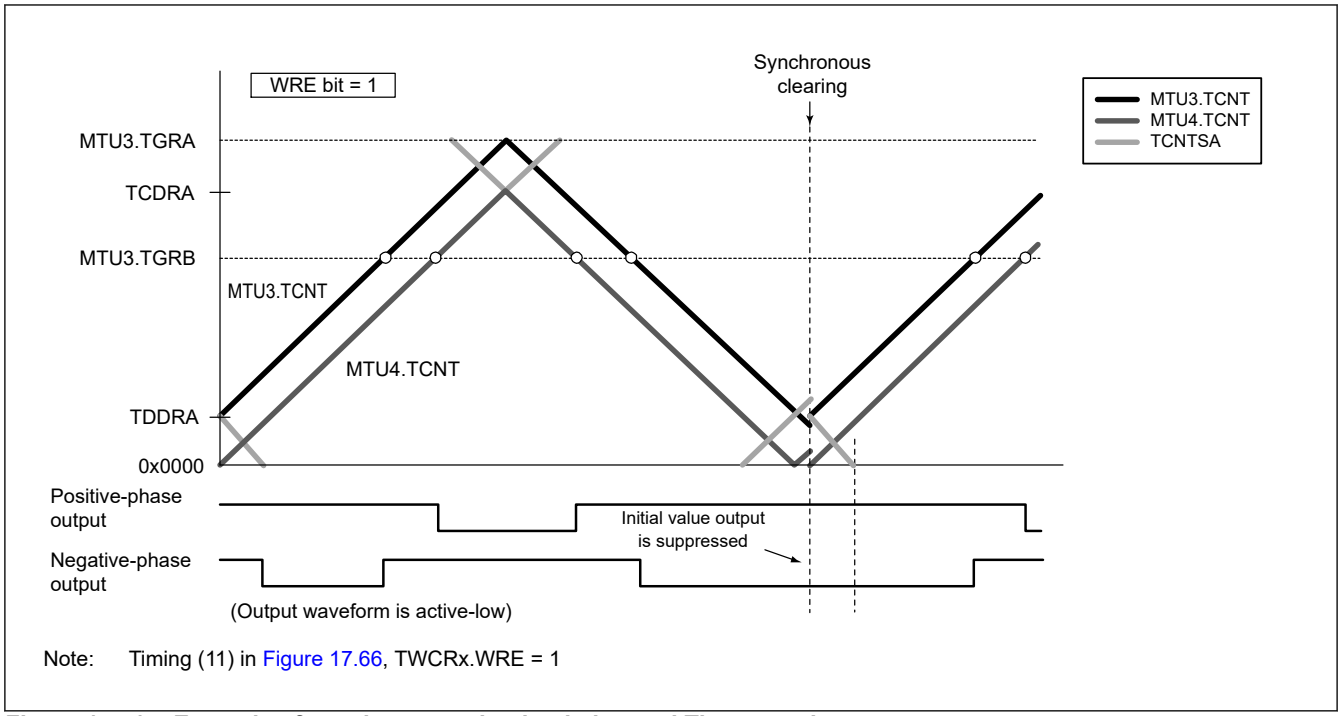
Note: Timing (6) in Figure 17.66, TWCRx.WRE = 1

Figure 17.69 Example of synchronous clearing in interval Tb at crest



Note: Timing (8) in Figure 17.66, TWCRx.WRE = 1

Figure 17.70 Example of synchronous clearing in dead time during down-counting



Note: Timing (11) in Figure 17.66, TWCRx.WRE = 1

Figure 17.71 Example of synchronous clearing in interval Tb at trough

(15) Suppressing Synchronous Counter Clearing by MTU0 to MTU2 in MTU6 and MTU7

In MTU6 and MTU7, setting the SCC bit in TWCRB to 1 suppresses synchronous counter clearing caused by MTU0 to MTU2.

Synchronous counter clearing caused by MTU0 to MTU2 is suppressed only within the interval shown in Figure 17.72. When using this function, MTU6 and MTU7 should be set to complementary PWM mode.

For details of synchronous clearing caused by MTU0 to MTU2, see section 17.3.10.2. Synchronous Counter Clearing for MTU6 and MTU7.

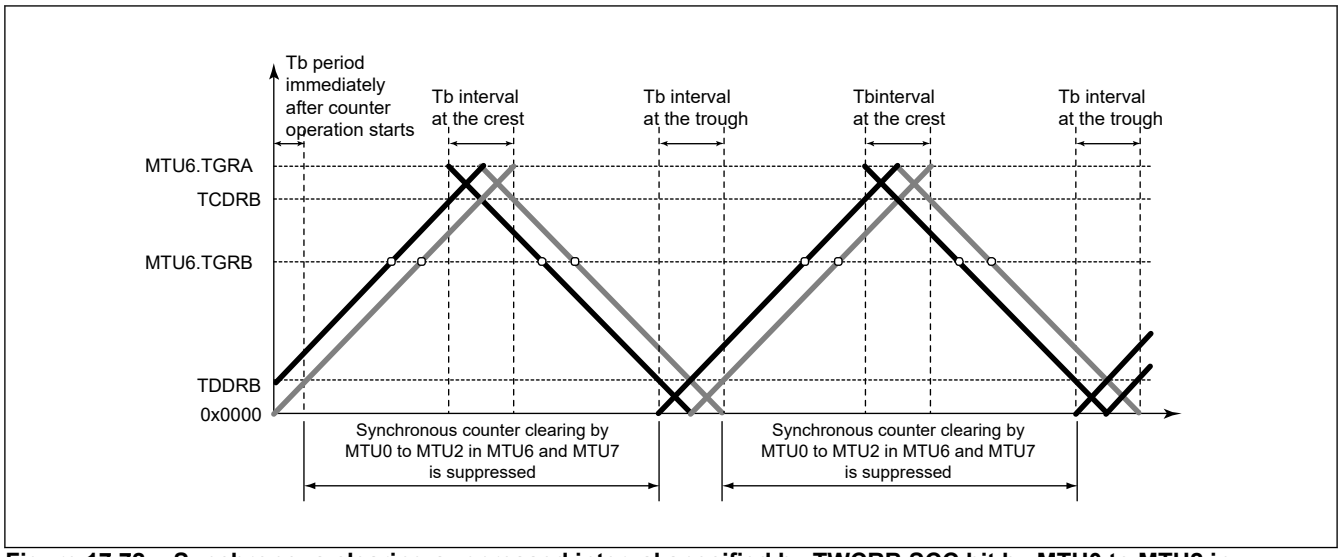
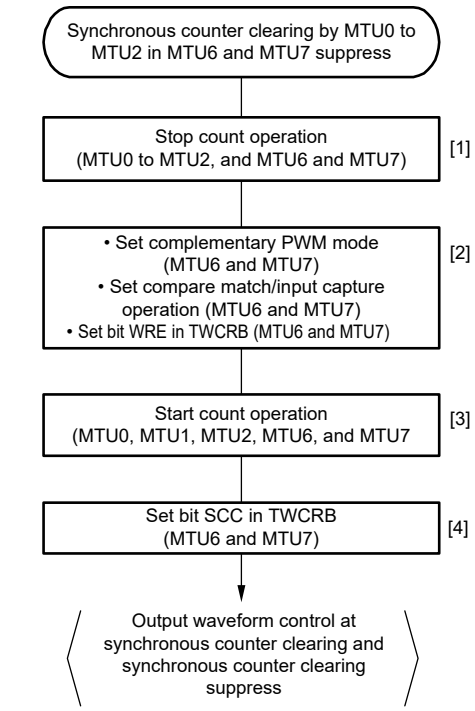


Figure 17.72 Synchronous clearing-suppressed interval specified by TWCRB.SCC bit by MTU0 to MTU2 in MTU6 and MTU7

Example of Procedure for Suppressing Synchronous Counter Clearing by MTU0 to MTU2 in MTU6 and MTU7

An example of the procedure for suppressing synchronous counter clearing by MTU0 to MTU2 in MTU6 and MTU7 is shown in Figure 17.73.



[1] Clear bits CST of the timer start register (TSTRB) in MTU6 and MTU7 to 0, and stop count operation. Clear bits CST of TSTRA in MTU0, MTU1, and MTU2 to 0, and stop count operation.

[2] Set the complementary PWM mode in MTU6 and MTU7 and compare match/input capture operation in MTU0, MTU1, and MTU2.
When bit WRE in TWCRA or TWCRB should be set, make appropriate setting here.

[3] Set bits CST6 and CST7 in MTU6 and MTU7 to 1 to start count operation.
For synchronous counter clearing by MTU0 to MTU2 in MTU6 and MTU7, set bits CST of TSTRA in MTU0, MTU1, and MTU2 to 1 to start count operation in any one of MTU0.TCNT to MTU2.TCNT.

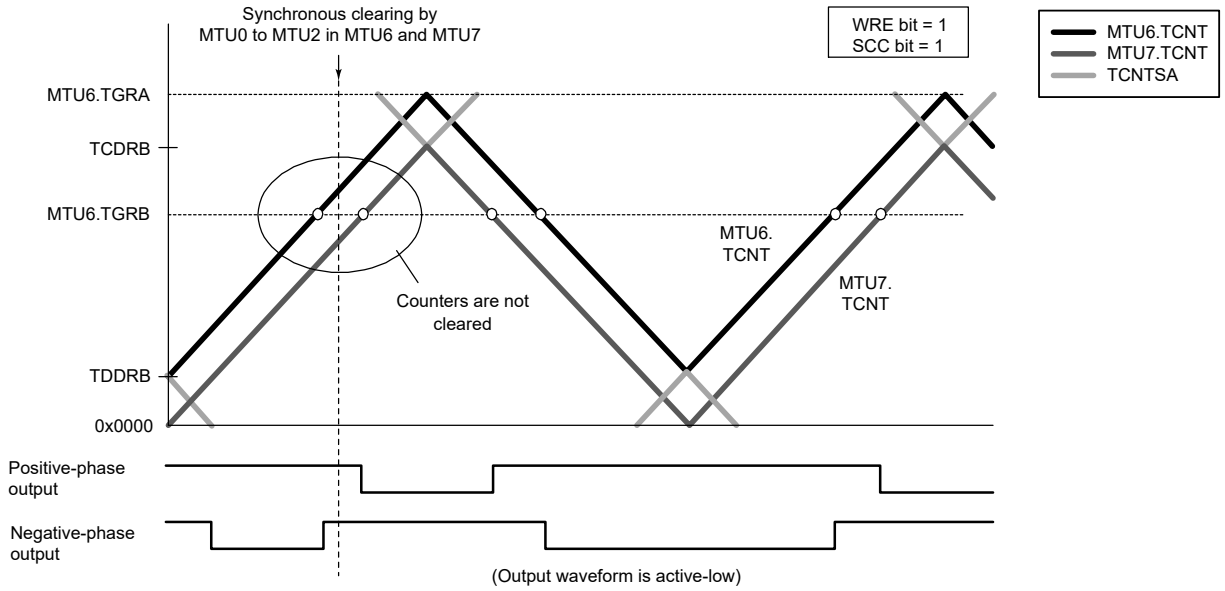
[4] Read TWCRB and then set bit SCC in TWCRB to 1 to suppress synchronous counter clearing by MTU0 to MTU2 in MTU6 and MTU7*1.
Here, do not modify the CCE and WRE bit values in TWCRB of MTU6 and MTU7. Synchronous counter clearing by MTU0 to MTU2 in MTU6 and MTU7 is suppressed in the synchronous counter clearing-suppressed periods.

Note 1. The TWCRB.SCC bit value can be modified during counter operation.
However, if a synchronous clearing occurs when bit TWCRB.SCC is modified from 0 to 1, the synchronous clearing may not be suppressed.
If a synchronous clearing occurs when bit TWCRB.SCC is modified from 1 to 0, the synchronous clearing may be suppressed.

Figure 17.73 Example of procedure for suppressing synchronous counter clearing by MTU0 to MTU2 in MTU6 and MTU7

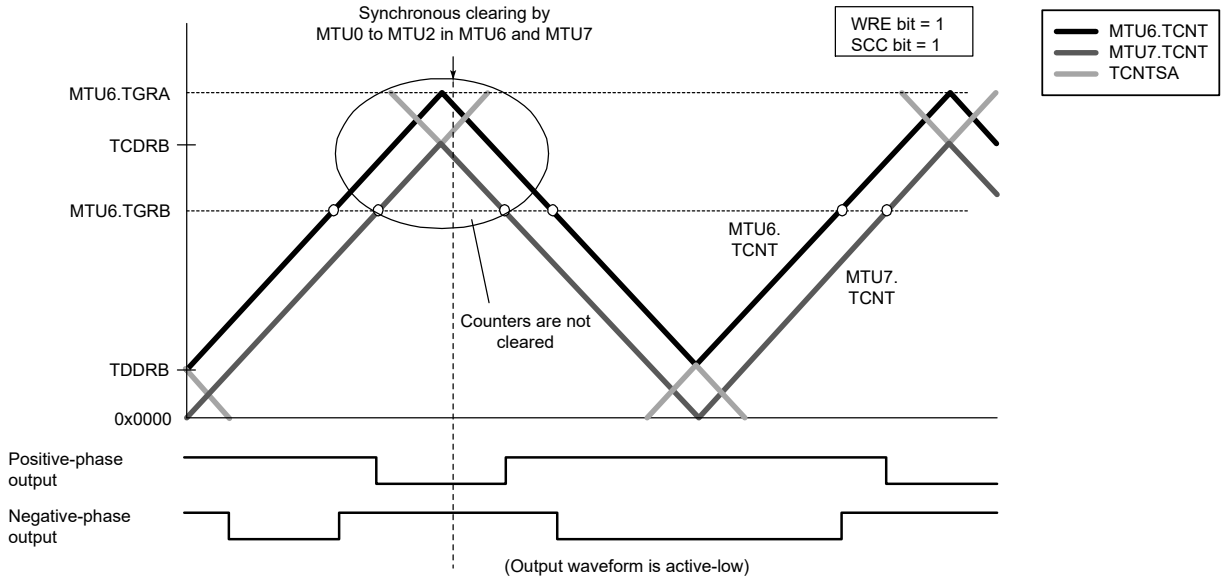
Examples of Suppression of Synchronous Counter Clearing by MTU0 to MTU2 in MTU6 and MTU7

Figure 17.74 to Figure 17.77 show examples of operation in which MTU6 and MTU7 operate in complementary PWM mode and synchronous counter clearing by MTU0 to MTU2 in MTU6 and MTU7 is suppressed by setting the SCC bit in TWCRB in MTU6 and MTU7 to 1. In the examples shown in Figure 17.74 to Figure 17.77, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 17.66, respectively. In these examples, the WRE bit in TWCRB is set to 1.



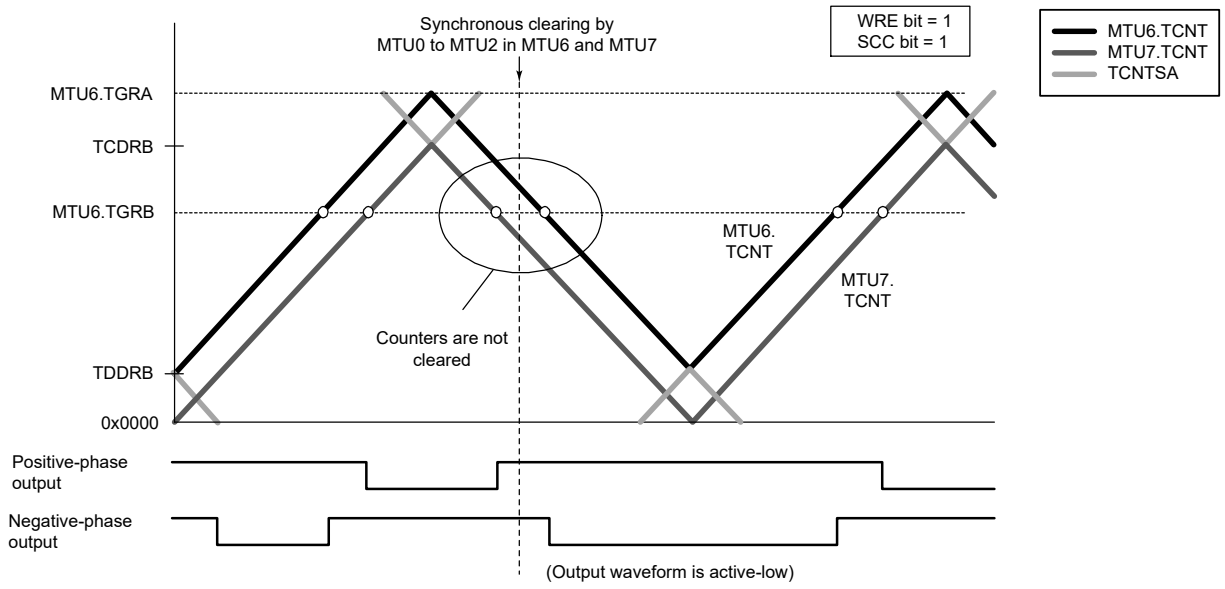
Note: Timing (3) in Figure 17.66, TWCRB.WRE = 1 and TWCRB.SCC = 1 for MTU6 and MTU7

Figure 17.74 Example of synchronous clearing in dead time during up-counting



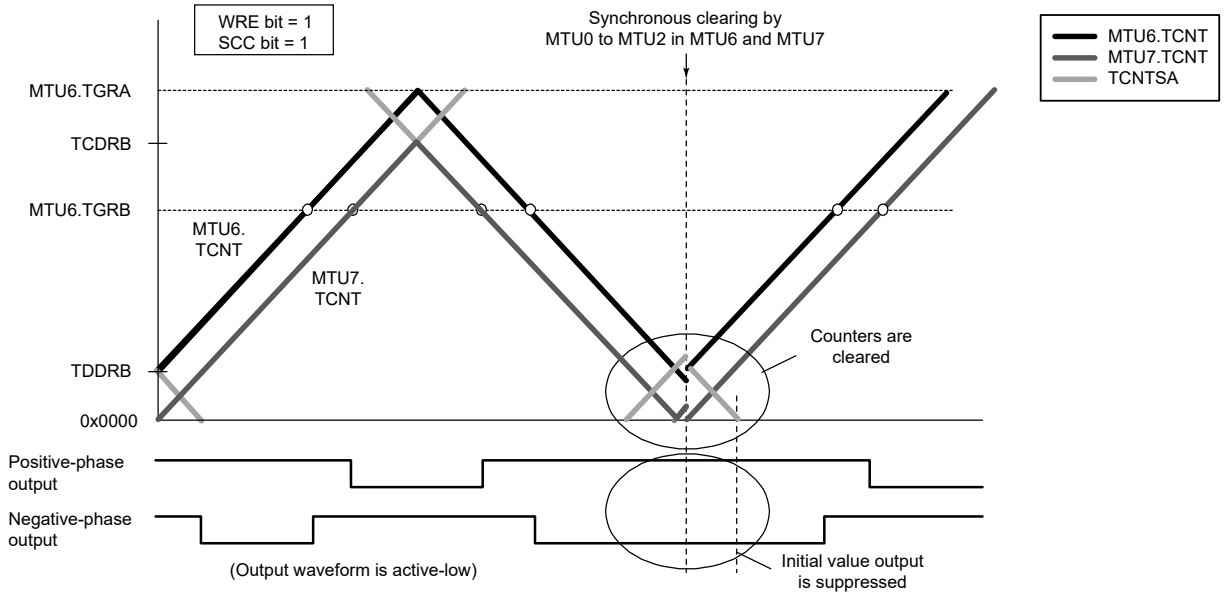
Note: Timing (6) in Figure 17.66, TWCRB.WRE = 1 and TWCRB.SCC = 1 for MTU6 and MTU7

Figure 17.75 Example of synchronous clearing in interval Tb at crest



Note: Timing (8) in Figure 17.66, TWCRB.WRE = 1 and TWCRB.SCC = 1 for MTU6 and MTU7

Figure 17.76 Example of synchronous clearing in dead time during down-counting



Note: Timing (11) in Figure 17.66, TWCRB.WRE = 1 and TWCRB.SCC = 1 for MTU6 and MTU7

Figure 17.77 Example of synchronous clearing in interval Tb at trough

(16) Counter Clearing by MTU3.TGRA (MTU6.TGRA) Compare Match

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTSA (MTU6.TCNT, MTU7.TCNT, and TCNTSB) can be cleared by MTU3.TGRA (MTU6.TGRA) compare match when the TWCRA.CCE (TWCRB.CCE) bit is set.

Figure 17.78 shows an operation example.

Note:
- Use this function only in complementary PWM mode 1 (transfer at crest).
- Do not specify synchronous clearing by another channel (do not set 1 in the SYNC0 to SYNC4, or SYNC6 and SYNC7 bits in the timer synchronous register (TSYRA or TSYRB) and the CE0A to CE0D, CE1A, and CE1B, or CE2A and CE2B bits in TSYCR).

- Do not set the PWM duty value to 0x0000.
- Do not set the PSYE bit in timer output control register 1 (TOCR1A or TOCR1B) to 1.

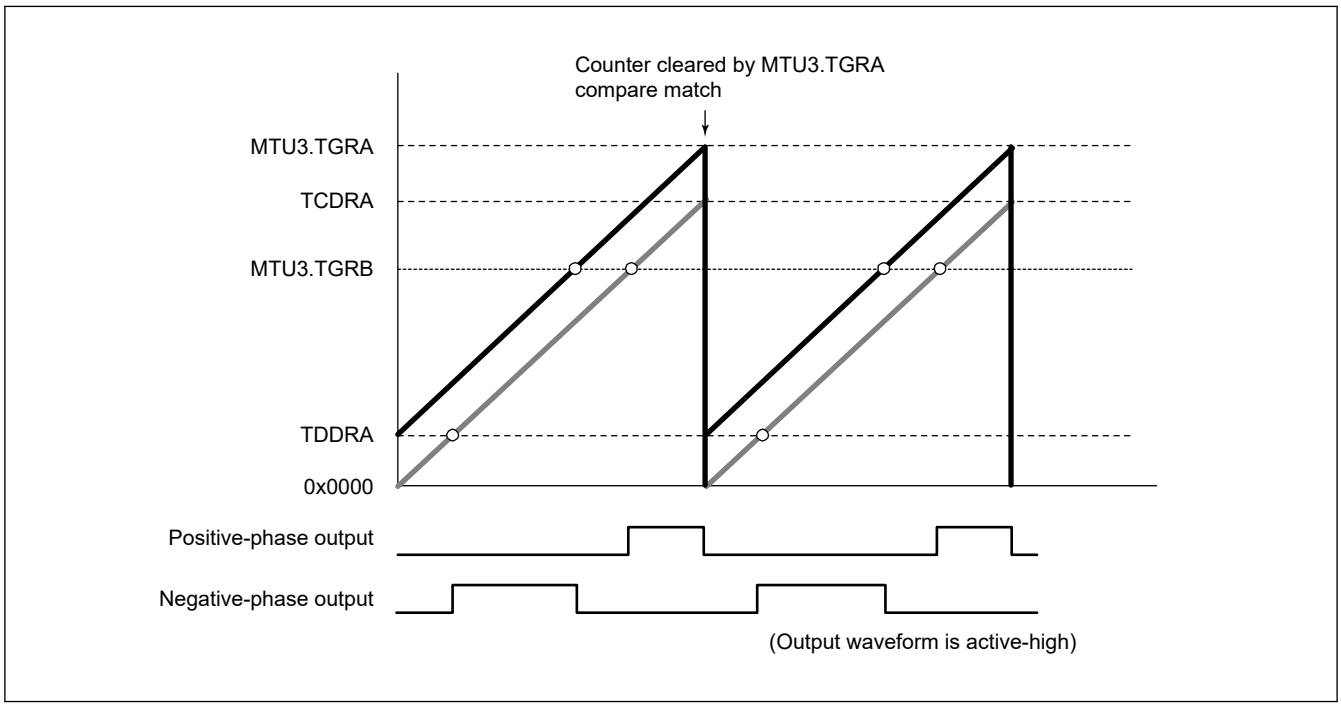


Figure 17.78 Example of counter clearing operation by MTU3.TGRA compare match

(17) Example of Waveform Output for Driving AC Synchronous Motor (Brushless DC Motor)

In complementary PWM mode, a brushless DC motor can easily be controlled using the TGCRA register. Figure 17.79 to Figure 17.82 show examples of brushless DC motor driving waveforms created using TGCRA.

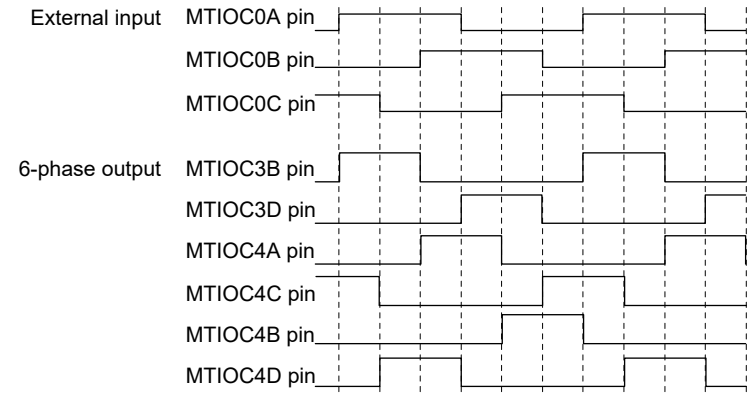
To switch the output phases for a 3-phase brushless DC motor by means of external signals detected with a Hall element, etc., set the TGCRA.FB bit to 0. In this case, the external signals indicating the magnetic pole position should be input to timer input pins MTIOC0A, MTIOC0B, and MTIOC0C in MTU0 (specify appropriate settings with the I/O port registers of the I/O ports). When an edge is detected at pin MTIOC0A, MTIOC0B, or MTIOC0C, the output on/off state is switched automatically.

When the TGCRA.FB bit is 1, the output on/off state is switched when the UF, VF, or WF bit in TGCRA is set to 0 or 1.

The driving waveforms are output from the 6-phase output pins for complementary PWM mode.

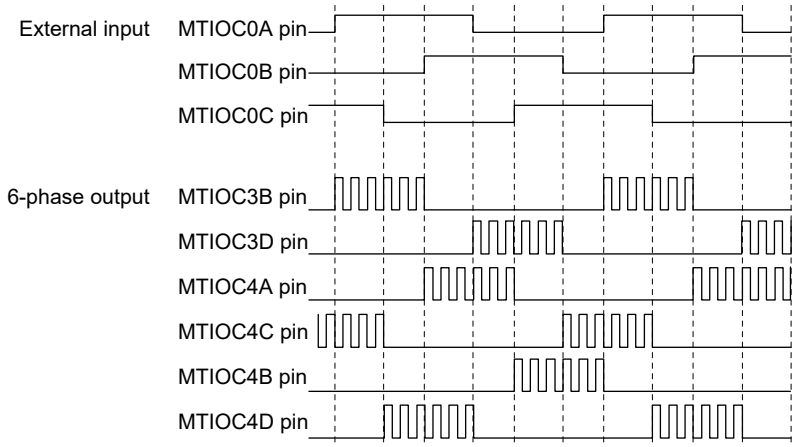
With this 6-phase output, while the output is turned on, chopping output is available through complementary PWM mode output function by setting the N bit or P bit in TGCRA to 1. When the N bit or P bit is 0, the level output is selected.

The active level of the 6-phase output (on output level) can be set with the TOCR1A.OLSN and TOCR1A.OLSP bits regardless of the setting of the N and P bits.



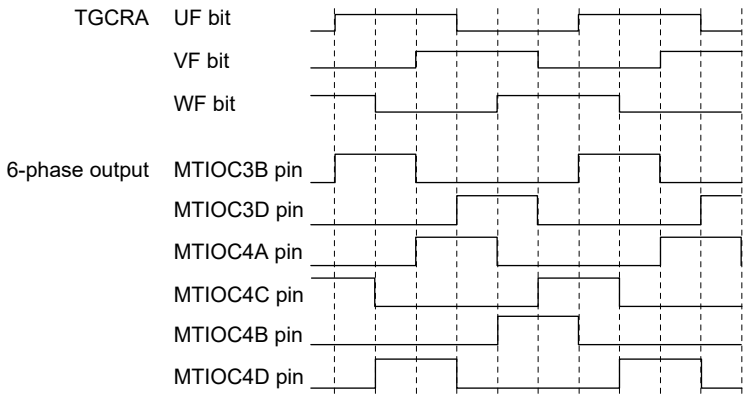
■ When TGCRA.BDC = 1, TGCRA.N = 0, TGCRA.P = 0, TGCRA.FB = 0, and output active level = high

Figure 17.79 Example of output phase switching by external input (1)



■ When TGCRA.BDC = 1, TGCRA.N = 1, TGCRA.P = 1, TGCRA.FB = 0, and output active level = high

Figure 17.80 Example of output phase switching by external input (2)



■ When TGCRA.BDC = 1, TGCRA.N = 0, TGCRA.P = 0, TGCRA.FB = 1, and output active level = high

Figure 17.81 Example of output phase switching through UF, VF, and WF bit settings (1)

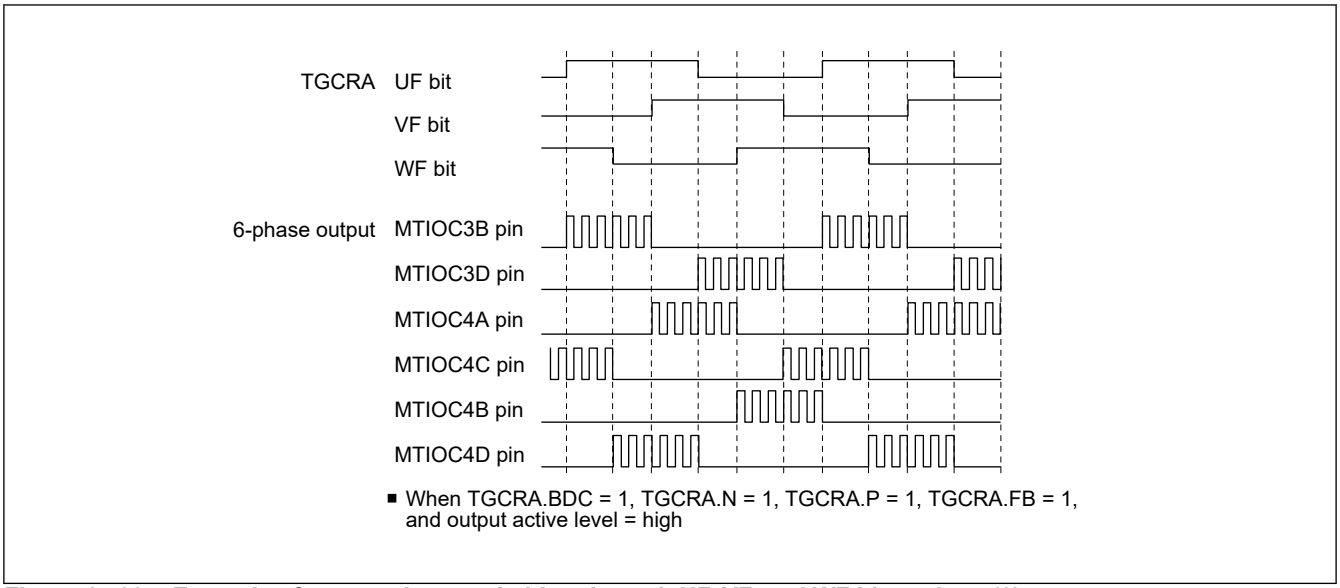


Figure 17.82 Example of output phase switching through UF, VF, and WF bit settings (2)

(18) A/D Converter Start Request Setting

In complementary PWM mode, an A/D converter start request can be issued using MTU3.TGRA (MTU6.TGRA) compare match, MTU4.TCNT (MTU7.TCNT) underflow (trough), or compare match on a channel other than MTU3 and MTU4 (MTU6 and MTU7).

When start requests using MTU3.TGRA (MTU6.TGRA) compare match are specified, A/D conversion can be started at the crest of the MTU3.TCNT (MTU6.TCNT) count.

A/D converter start requests can be specified by setting the TIER.TTGE bit to 1. To issue an A/D converter start request at an MTU4.TCNT (MTU7.TCNT) underflow (trough), set the MTU4.TIER.TTGE2 (MTU7.TIER.TTGE2) bit to 1.

(19) Double Buffer Function in Complementary PWM Mode

In complementary PWM mode 3 (transfer at the crest and trough), the PWM output setting resolution can be improved from ±2 to ±1 by setting the TMDR2A.DRS (TMDR2B.DRS) bit to 1.

When setting buffer registers A (MTU3.TGRD, MTU4.TGRC, MTU4.TGRD, MTU6.TGRD, MTU7.TGRC, and MTU7.TGRD), set also buffer registers B (MTU3.TGRE, MTU4.TGRE, MTU4.TGRF, MTU6.TGRE, MTU7.TGRE, and MTU7.TGRF) at the same time. Each buffer register B should be set to the buffer register A value or (buffer register A value − 1). For details of the setting procedure, see section 17.3.8.1. Example of Complementary PWM Mode Setting Procedure.

Note: When a buffer register B is set to the buffer register A value, symmetric PWM waveforms are output. When a buffer register B is set to (buffer register A value − 1), asymmetric PWM waveforms are output.

Figure 17.83 shows an example of double buffer operation.

Each register data is transferred as follows.

- After MTU4.TGRD or MTU7.TGRD (buffer A) is written to, data is transferred from MTU4.TGRD or MTU7.TGRD (buffer A) to Temp3A or Temp6A (temporary A) and from MTU4.TGRF or MTU7.TGRF (buffer B) to Temp3B or Temp6B (temporary B).
- With timing (1) in the figure, data is transferred from Temp3A or Temp6A (temporary A) to MTU4.TGRB or MTU7.TGRB (compare).
- With timing (2) in the figure, data is transferred from Temp3B or Temp6B (temporary B) to MTU4.TGRB or MTU7.TGRB (compare).

In the crest interval (Tb interval at crest), the compare register and temporary register A are valid; in the trough interval (Tb interval at trough), the compare register and temporary register B are valid.

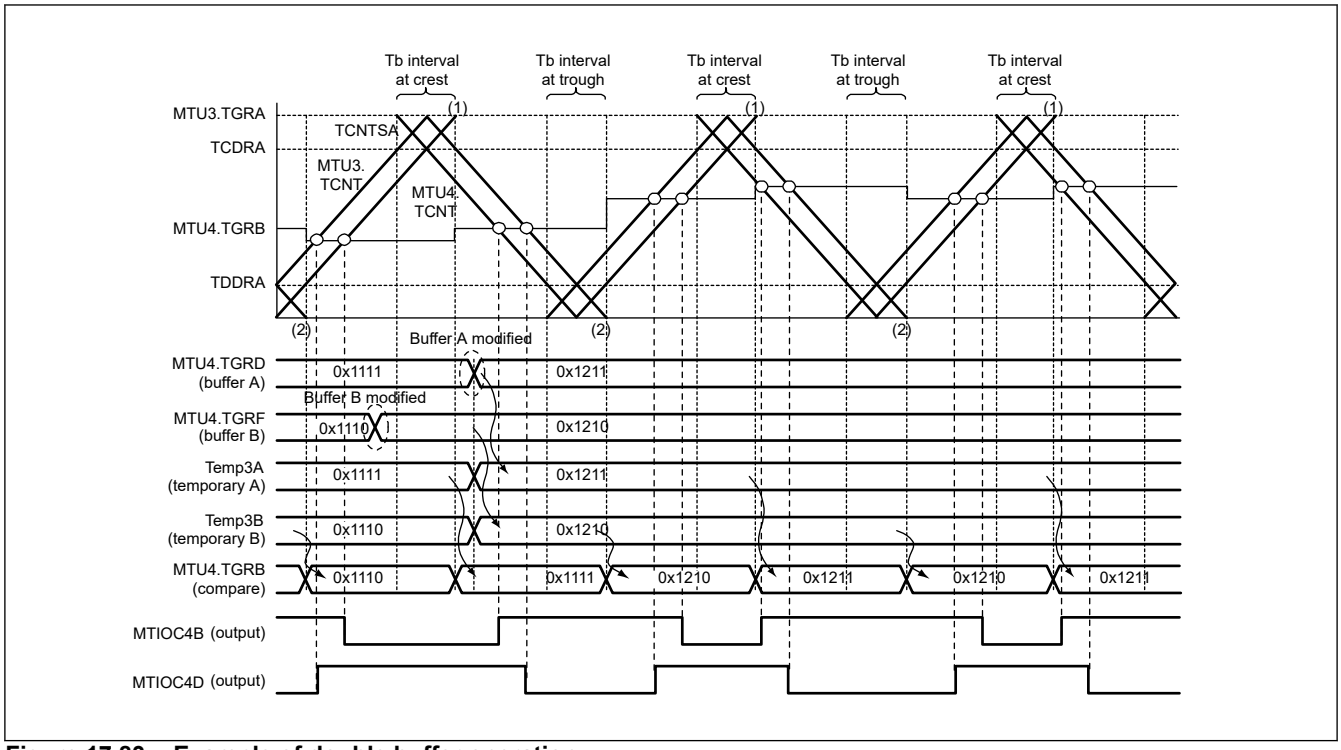


Figure 17.83 Example of double buffer operation

Figure 17.84 shows an example when the buffer write value is smaller than the TDDRA value, and Figure 17.85 shows an example when the write value is greater than TCDRA.

In the crest interval, the output is controlled according to the compare match with the compare register or temporary register A; in the trough interval, the output is controlled according to the compare match with the compare register or temporary register B.

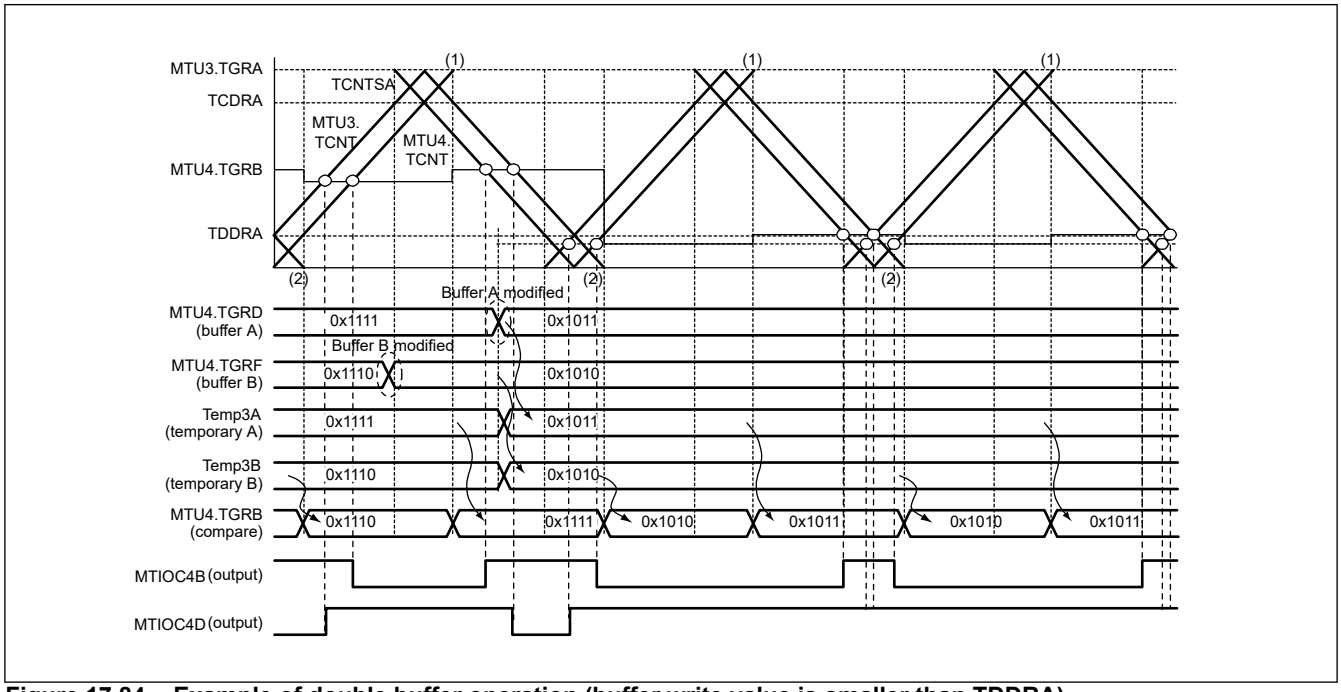


Figure 17.84 Example of double buffer operation (buffer write value is smaller than TDDRA)

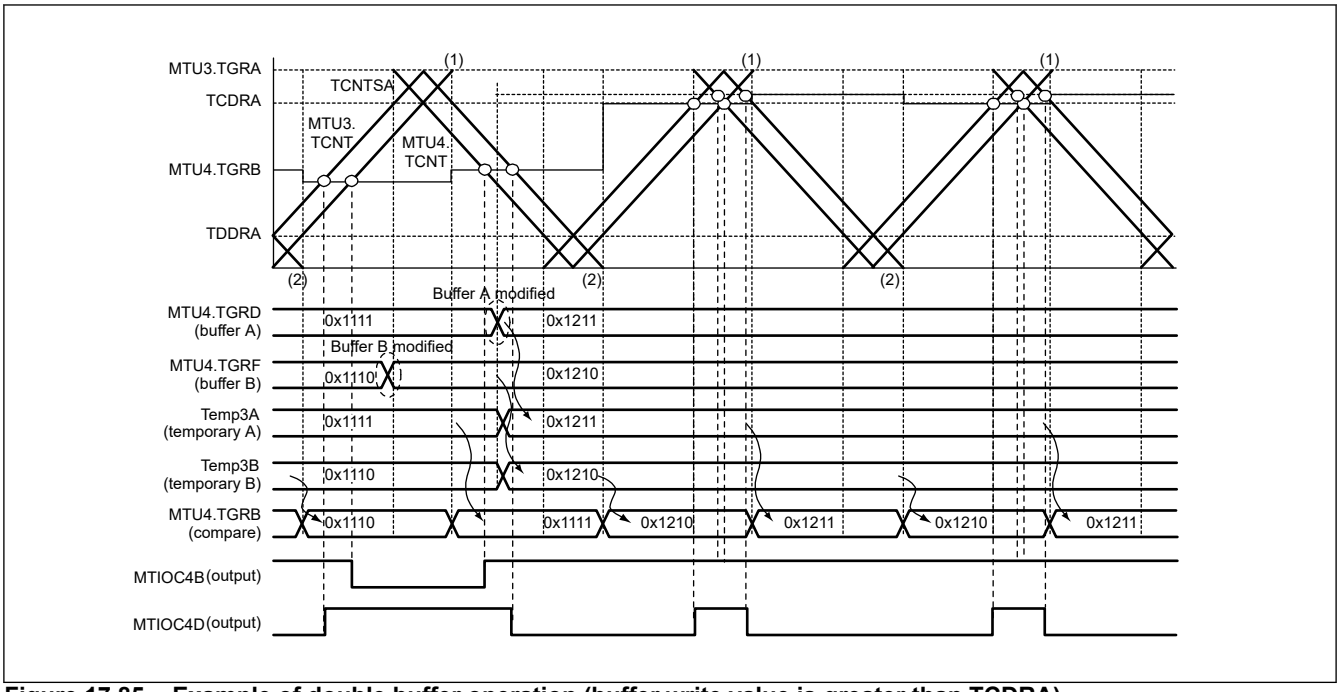


Figure 17.85 Example of double buffer operation (buffer write value is greater than TCDRA)

17.3.8.3 Interrupt Skipping Function 1 in Complementary PWM Mode

Interrupts TGIA3 (TGIA6) (at the crest) and TCIV4 (TCIV7) (at the trough) in MTU3 and MTU4 (MTU6 and MTU7) can be skipped up to seven times by making settings in the TITCR1A (TITCR1B) register.

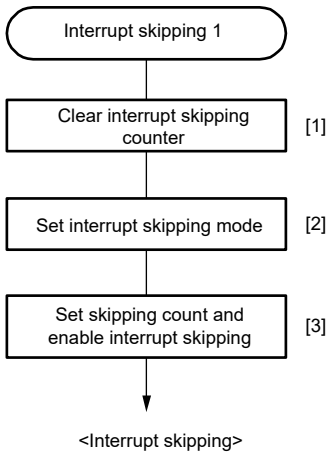
Transfers from a buffer register to a temporary register or a compare register can be skipped in coordination with interrupt skipping by making settings in the TBTERA (TBTERB) register. For the linkage with buffer registers, see description (3)Buffer Transfer Control Linked with Interrupt Skipping, below.

A/D converter start requests generated by the A/D converter start request delaying function can also be skipped in coordination with interrupt skipping by making settings in the MTU4.TADCR (MTU7.TADCR) register. For the linkage with the A/D converter start request delaying function, see section 17.3.9. A/D Converter Start Request Delaying Function.

The TITCR1A (TITCR1B) register should be set while interrupt skipping function 1 is selected by setting the TITM bit in the timer interrupt skipping mode register (TITMRA or TITMRB) to 0, TGIA3 (TGIA6) interrupt requests are disabled by setting the MTU3.TIER (MTU6.TIER) register, TCIV4 (TCIV7) interrupt requests are disabled by setting the MTU4.TIER (MTU7.TIER) register, and a compare match is not generated. Before changing the skipping count, be sure to set the T3AEN (T6AEN) and T4VEN (T7VEN) bits to 0 to clear the skipping counter.

(1) Example of Interrupt Skipping Function 1 Setting Procedure

Figure 17.86 shows an example of the interrupt skipping function 1 setting procedure. Figure 17.87 shows the periods during which interrupt skipping count can be changed.



[1] Set bits T3AEN and T4VEN in the timer interrupt skipping set register 1 (TITCR1A) to 0 to clear the skipping counter.

[2] Set bit TITM in the timer interrupt skipping mode register (TITMRA) to 0 to select interrupt skipping function 1.

[3] Specify the interrupt skipping count within the range from 0 to 7 times in bits T3ACOR2 to T3ACOR0 and T4VCOR2 to T4VCOR0 in TITCR1A, and enable interrupt skipping through bits T3AEN and T4VEN.

Note: The setting of TITCR1A must be done while the TGIA3 and TCIV4 interrupt requests are disabled by the settings of MTU3.TIER and MTU4.TIER under the conditions in which a compare match is not generated. Before changing the skipping count, be sure to set the T3AEN and T4VEN bits to 0 to clear the skipping counter.

Figure 17.86 Example of interrupt skipping function 1 setting procedure

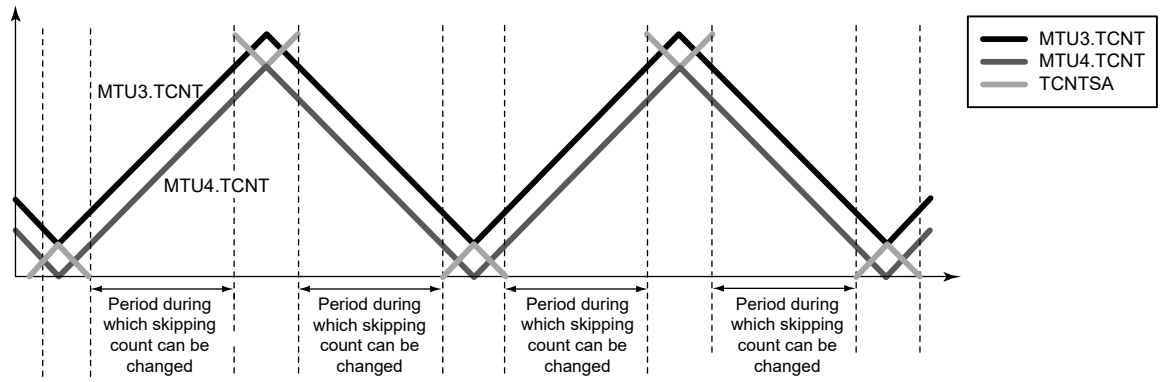


Figure 17.87 Periods during which interrupt skipping count can be changed

(2) Example of Interrupt Skipping Function 1

Figure 17.88 shows an example of TGIA3 (TGIA6) interrupt skipping in which the interrupt skipping count is set to three by the T3ACOR (T6ACOR) bits and the T3AEN (T6AEN) bit is set to 1 in the TITCR1A (TITCR1B) register.

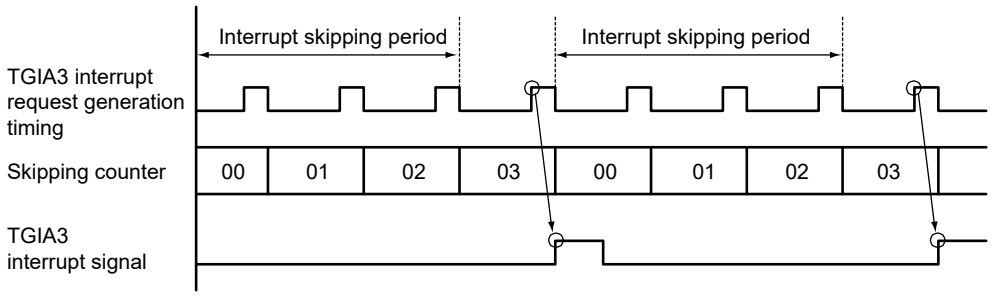


Figure 17.88 Example of interrupt skipping function 1

(3) Buffer Transfer Control Linked with Interrupt Skipping

In complementary PWM mode, whether to transfer data from a buffer register to a temporary register and whether to link the transfer with interrupt skipping can be specified with the BTE[1:0] bits in the TBTERA (TBTERB) register.

Figure 17.89 shows an example of operation when buffer transfer is disabled (BTE[1:0] = 01b). While this setting is valid, data is not transferred from the buffer register to the temporary register.

Figure 17.90 shows an example of operation when buffer transfer is linked with interrupt skipping (BTE[1:0] = 10b).

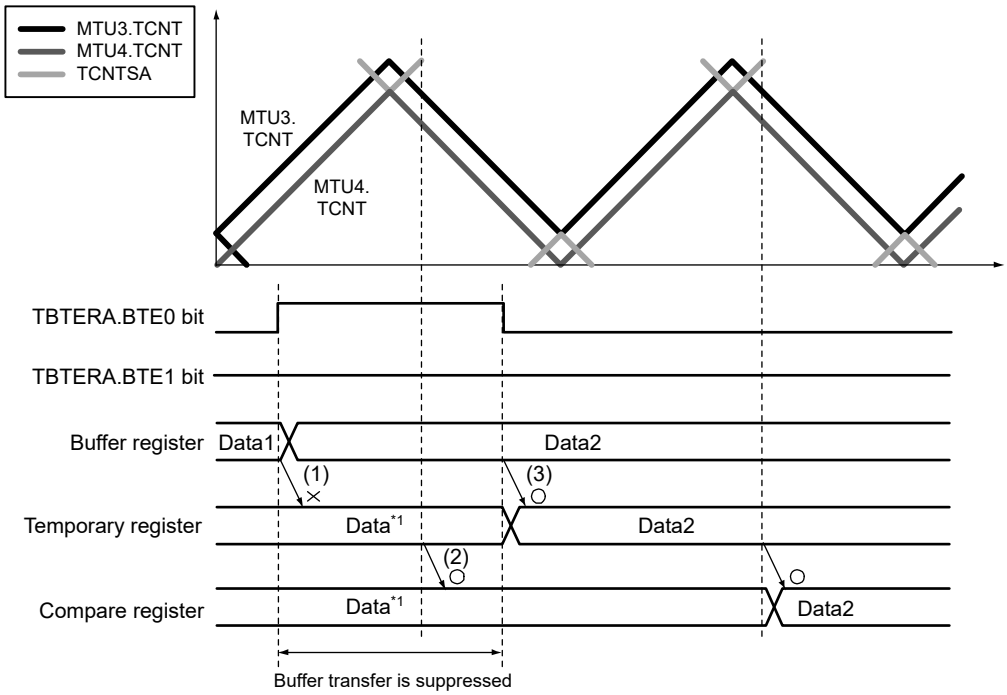
While this setting is valid, data is not transferred from the buffer register outside the buffer transfer-enabled period.

Note that the buffer transfer-enabled period depends on the T3AEN (T6AEN) and T4VEN (T7VEN) bit settings in the TITCR1A (TITCR1B) register. Figure 17.91 shows the relationship between the T3AEN (T6AEN) and T4VEN (T7VEN) bit settings in TITCR1A (TITCR1B) and buffer transfer-enabled period.

Note: This function must be used in combination with interrupt skipping function 1.

When interrupt skipping is disabled (the T3AEN and T4VEN (T6AEN and T7VEN) bits in the timer interrupt skipping set register 1 (TITCR1A or TITCR1B) are set to 0 or the skipping count set bits (T3ACOR and T4VCOR (T6ACOR and T7VCOR)) in TITCR1A (TITCR1B) are set to 0), make sure that buffer transfer is not linked with interrupt skipping (set the BTE1 bit in TBTERA or TBTERB to 0).

If buffer transfer is linked with interrupt skipping while interrupt skipping is disabled, buffer transfer is never performed.

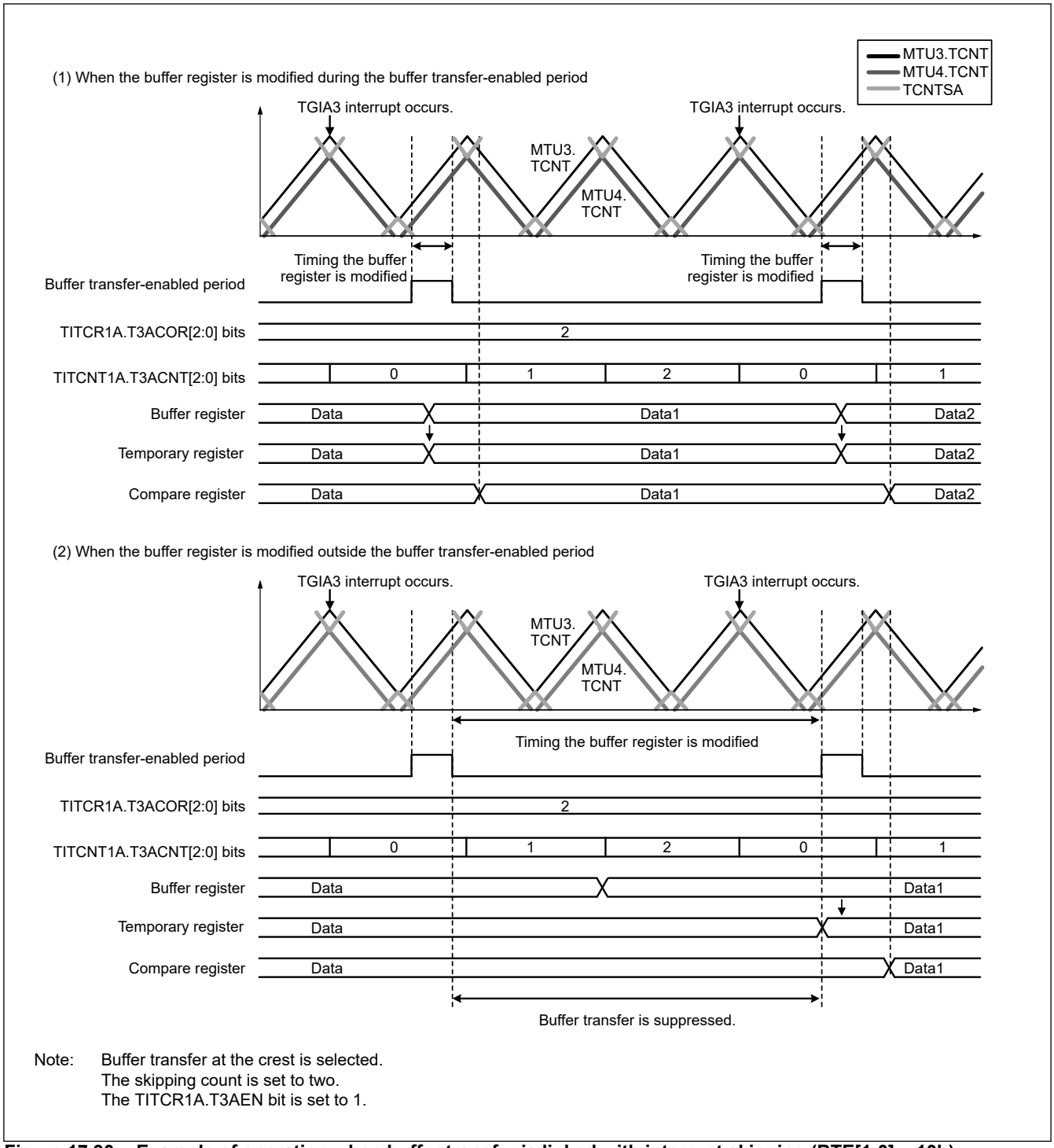


[Legend]

(1) No data is transferred from the buffer register to the temporary register in the buffer transfer-disabled period (bits BTE[1:0] in TBTERA are set to 01b).
(2) Data is transferred from the temporary register to the compare register even in the buffer transfer-disabled period.
(3) After buffer transfer is enabled, data is transferred from the buffer register to the temporary register.

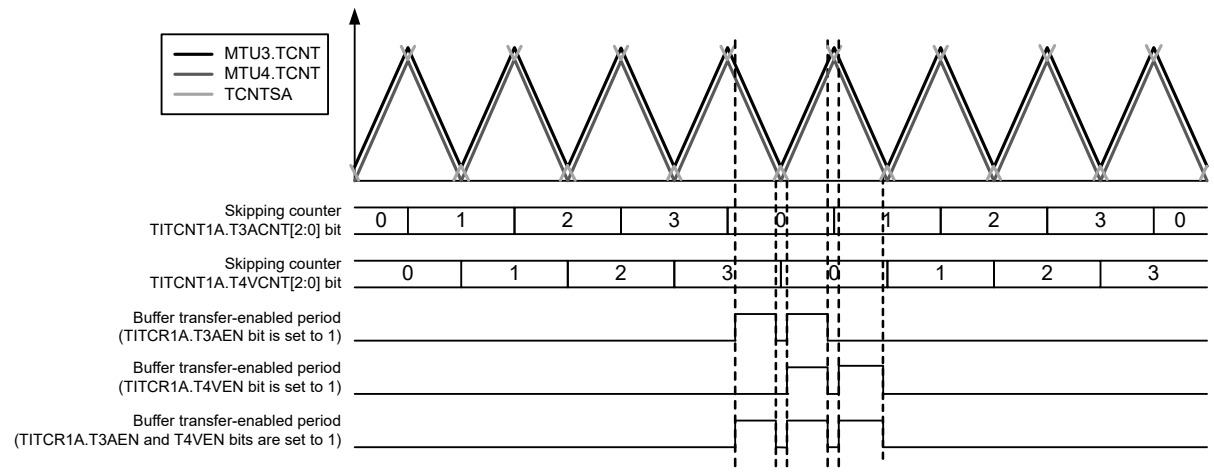
Note 1. When buffer transfer at the crest is selected.

Figure 17.89 Example of operation when buffer transfer is disabled (BTE[1:0] = 01b)



Note: Buffer transfer at the crest is selected.
The skipping count is set to two.
The TITCR1A.T3AEN bit is set to 1.

Figure 17.90 Example of operation when buffer transfer is linked with interrupt skipping (BTE[1:0] = 10b)



Note: The skipping count is set to three.
Buffer transfer at the crest and trough is selected.

Figure 17.91 Relationship between bits T3AEN and T4VEN in TITCR1A and buffer transfer-enabled period

17.3.8.4 Complementary PWM Mode Output Protection Functions

The MTU provides the following protection functions for complementary PWM mode output.

(1) Register and Counter Miswrite Prevention Function

With the exception of the buffer registers, which can be modified, access by the CPU can be enabled or disabled for the mode registers, control registers, compare registers, and counters used in complementary PWM mode by means of the RWE bit in the timer read/write enable register (TRWERA or TRWERB). The applicable registers are some of the registers in MTU3, MTU4, MTU6, and MTU7 shown below:

47 registers in total

MTU3.TCR, MTU4.TCR, MTU3.TCR2, MTU4.TCR2, MTU3.TMDR1, MTU4.TMDR1, MTU3.TIORH, MTU4.TIORH, MTU3.TIORL, MTU4.TIORL, MTU3.TIER, MTU4.TIER, MTU3.TCNT, MTU4.TCNT, MTU3.TGRA, MTU4.TGRA, MTU3.TGRB, MTU4.TGRB, MTU.TOEERA, MTU.TOCR1A, MTU.TOCR2A, MTU.TGCRA, MTU.TCDRA, MTU.TDDRA, MTU6.TCR, MTU7.TCR, MTU6.TCR2, MTU7.TCR2, MTU6.TMDR1, MTU7.TMDR1, MTU6.TIORH, MTU7.TIORH, MTU6.TIORL, MTU7.TIORL, MTU6.TIER, MTU7.TIER, MTU6.TCNT, MTU7.TCNT, MTU6.TGRA, MTU7.TGRA, MTU6.TGRB, MTU7.TGRB, MTU.TOERB, MTU.TOCR1B, MTU.TOCR2B, MTU.TCDRB, and MTU.TDDRB

This function can disable CPU access to the mode registers, control registers, and counters to prevent miswriting due to CPU runaway. In the access-disabled state, the applicable registers are read as undefined and writing to these registers is ignored.

(2) Halting of PWM Output by External Signal

The 6-phase PWM output pins can be set to the high-impedance state automatically by inputting specified external signals.

For details, see section 18, Port Output Enable 3 (POE3).

17.3.9 A/D Converter Start Request Delaying Function

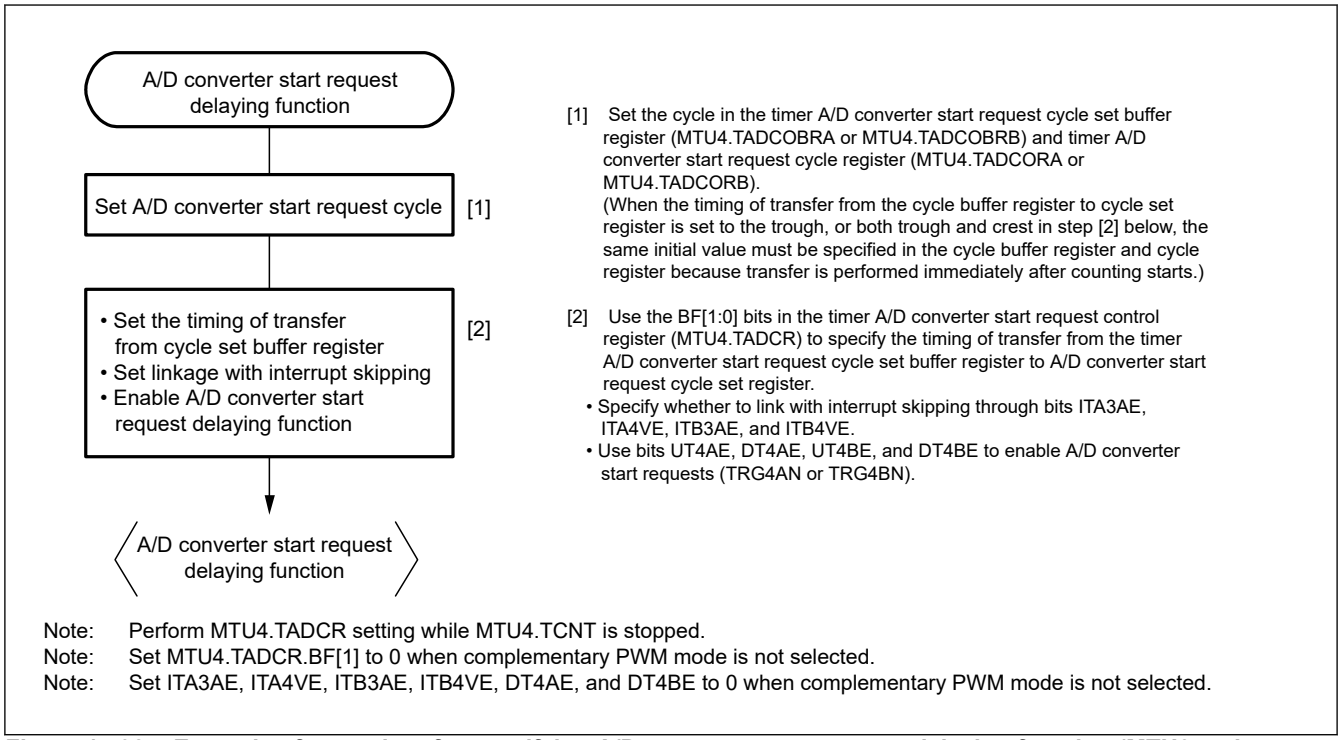
A/D converter start requests can be issued in MTU4 or MTU7 by making settings in the timer A/D converter start request control register (MTU4.TADCR or MTU7.TADCR), timer A/D converter start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB, or MTU7.TADCORA and MTU7.TADCORB), and timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB, or MTU7.TADCOBRA and MTU7.TADCOBRB).

The A/D converter start request delaying function compares MTU4.TCNT with MTU4.TADCORA or MTU4.TADCORB (MTU7.TCNT with MTU7.TADCORA or MTU7.TADCORB), and when their values match, the function issues a respective A/D converter start request (TRG4AN or TRG4BN (TRG7AN or TRG7BN)).

A/D converter start requests (TRG4AN and TRG4BN (TRG7AN and TRG7BN)) can be skipped in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in MTU4.TADCR (the ITA6AE, ITA7VE, ITB6AE, and ITB7VE bits in MTU7.TADCR).

17.3.9.1 Example of Procedure for Specifying A/D Converter Start Request Delaying Function

Figure 17.92 shows an example of procedure for specifying the A/D converter start request delaying function.

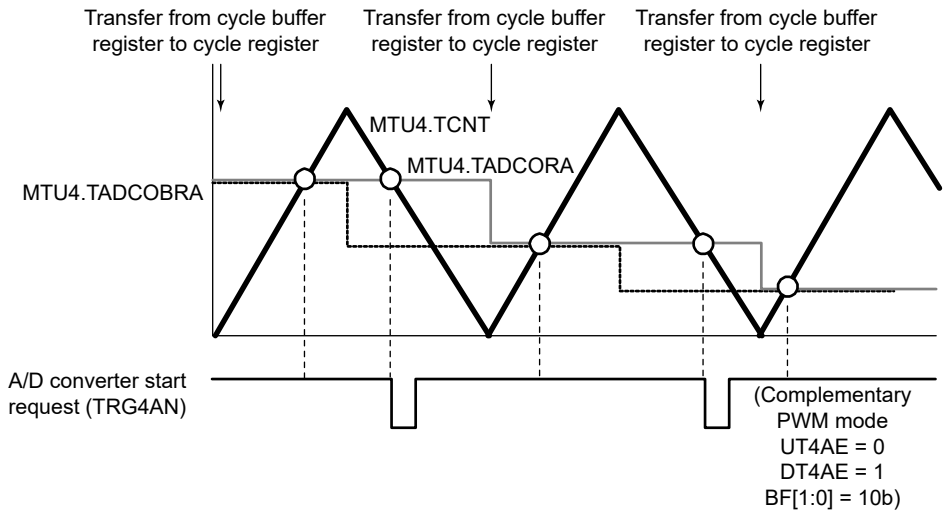


Note: Perform MTU4.TADCR setting while MTU4.TCNT is stopped.
Note: Set MTU4.TADCR.BF[1] to 0 when complementary PWM mode is not selected.
Note: Set ITA3AE, ITA4VE, ITB3AE, ITB4VE, DT4AE, and DT4BE to 0 when complementary PWM mode is not selected.

Figure 17.92 Example of procedure for specifying A/D converter start request delaying function (MTU3 and MTU4)

17.3.9.2 Basic Example of A/D Converter Start Request Delaying Function Operation

Figure 17.93 shows a basic example of A/D converter start request signal (TRG4AN (TRG7AN)) operation when the trough of MTU4.TCNT (MTU7.TCNT) is specified for the buffer transfer timing and an A/D converter start request signal is output during MTU4.TCNT (MTU7.TCNT) down-counting.



Note: Period in which A/D converter start requests are enabled (UT4AE = 1): 0 ≤ MTU4.TCNT ≤ TCDR − 1
Period in which A/D converter start requests are enabled (DT4AE = 1): TCDR ≥ MTU4.TCNT ≥ 1
UT4AE, DT4AE, BF[1:0]: Bits in TADCR

Figure 17.93 Basic example of A/D converter start request signal (TRG4AN) operation

17.3.9.3 Buffer Transfer

The data in the timer A/D converter start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB, or MTU7.TADCORA and MTU7.TADCORB) is updated by writing data to the timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB, or MTU7.TADCOBRA and MTU7.TADCOBRB). Data is transferred from the buffer registers to the respective cycle set registers at the timing selected with the BF[1:0] bits in the MTU4.TADCR (MTU7.TADCR) register.

In complementary PWM mode, data is also transferred from the timer A/D converter start request cycle set buffer registers to the timer A/D converter start request cycle set registers when MTU4.TGRD (MTU7.TGRD) register is updated.

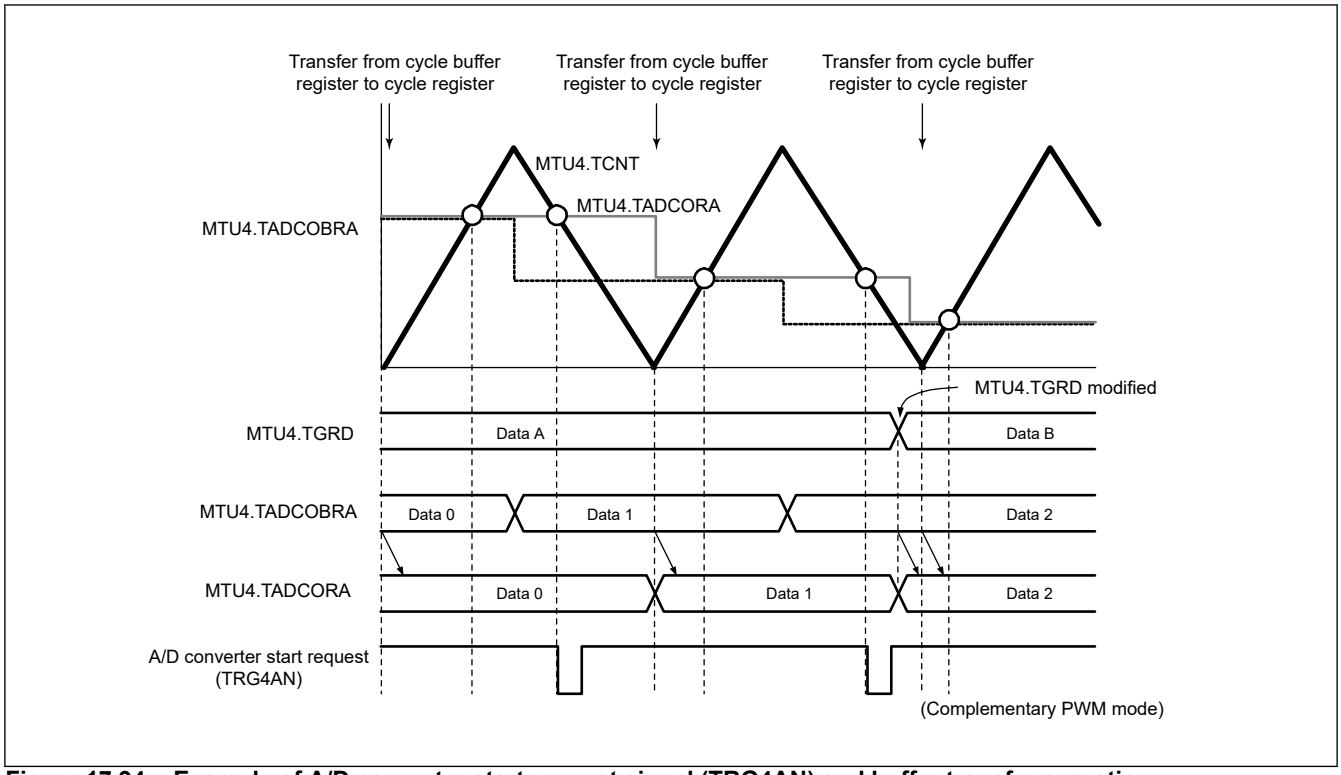


Figure 17.94 Example of A/D converter start request signal (TRG4AN) and buffer transfer operation

17.3.9.4 A/D Converter Start Request Delaying Function Linked with Interrupt Skipping Function 1

A/D converter start requests (TRG4AN and TRG4BN (TRG7AN and TRG7BN)) can be issued in coordination with interrupt skipping 1 by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE (ITA6AE, ITA7VE, ITB6AE, and ITB7VE) bits in the MTU4.TADCR (MTU7.TADCR) register.

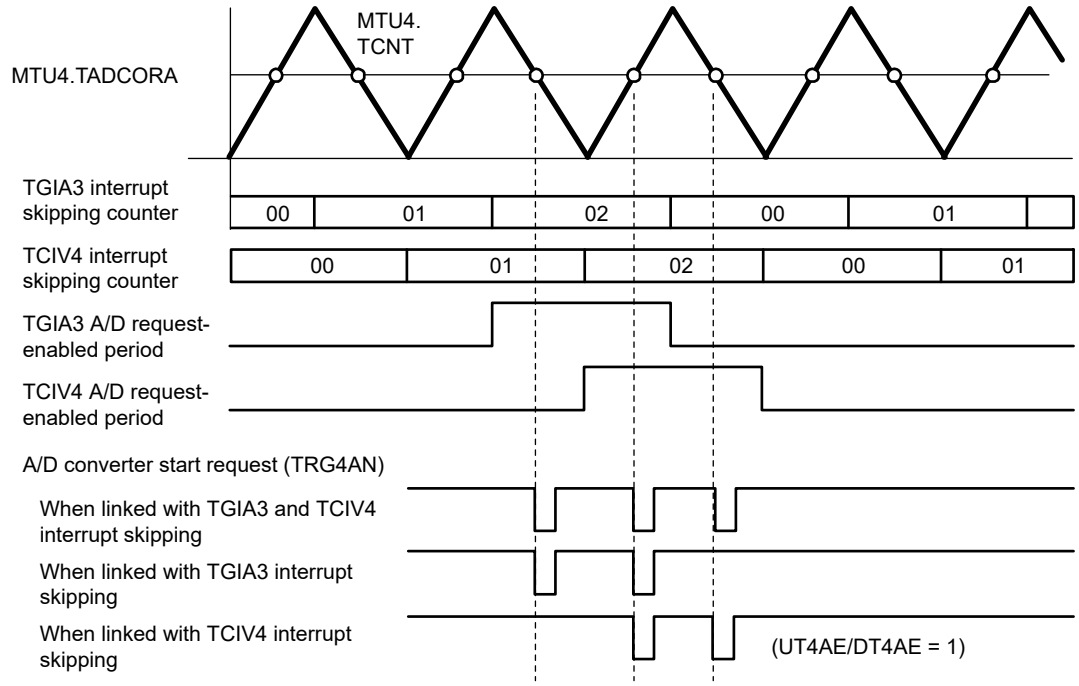
Figure 17.95 shows an example of A/D converter start request signal (TRG4AN (TRG7AN)) operation when TRG4AN (TRG7AN) output is enabled during MTU4.TCNT (MTU7.TCNT) up-counting and down-counting and A/D converter start requests are linked with interrupt skipping 1.

Figure 17.96 shows another example of A/D converter start request signal (TRG4AN (TRG7AN)) operation when TRG4AN (TRG7AN) output is enabled during MTU4.TCNT (MTU7.TCNT) up-counting and A/D converter start requests are linked with interrupt skipping 1.

Note: This function should be used in combination with interrupt skipping 1.

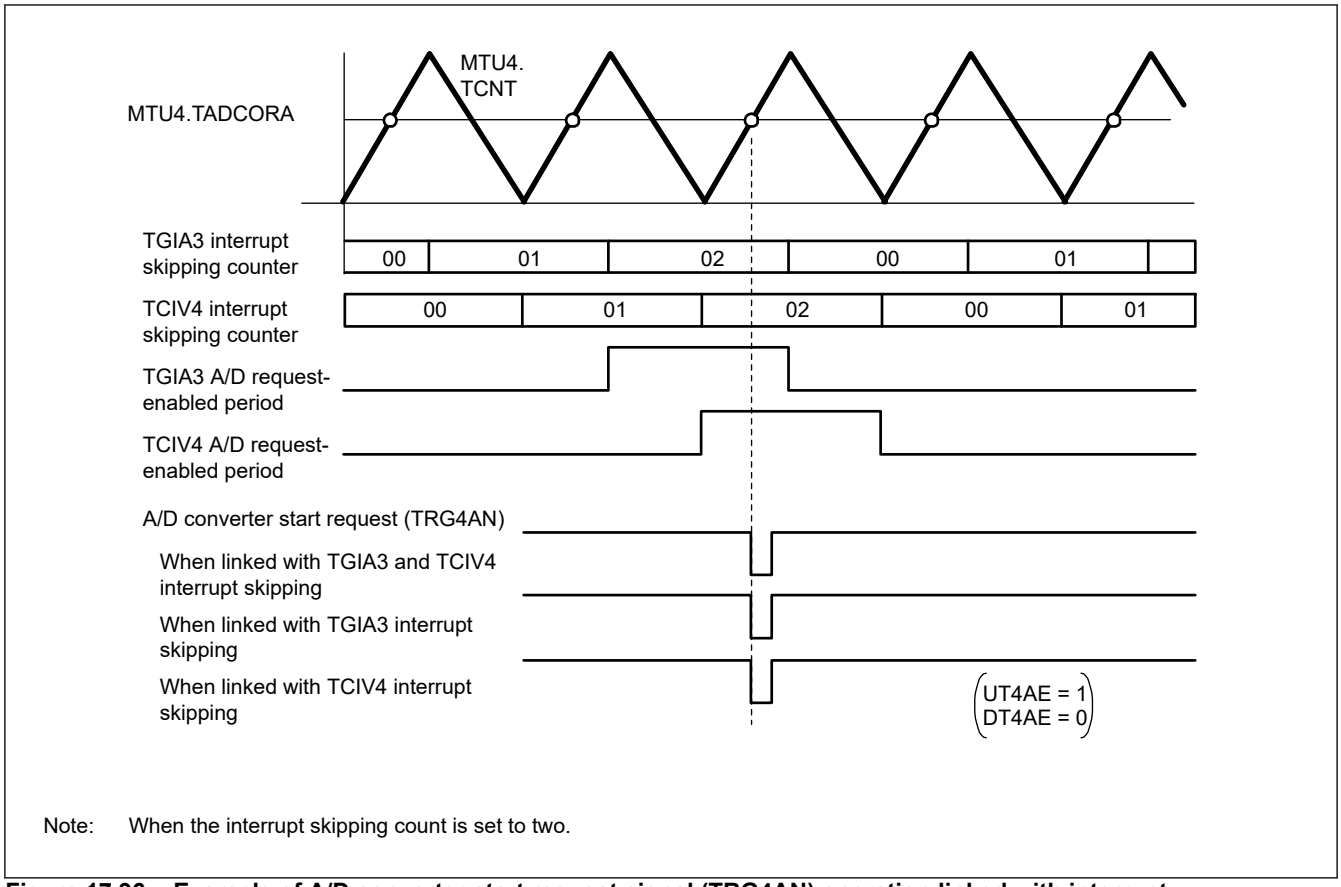
When interrupt skipping is disabled (the T3AEN and T4VEN (T6AEN and T7VEN) bits in the timer interrupt skipping set register (TITCR1A (TITCR1B)) are set to 0 or the skipping count set bits (T3ACOR and T4VCOR (T6ACOR and T7VCOR)) in TITCR1A (TITCR1B) are set to 0), make sure that A/D converter start requests are not linked with interrupt skipping 1 (set the ITA3AE, ITA4VE, ITB3AE, and ITB4VE (ITA6AE, ITA7VE, ITB6AE, and ITB7VE) bits in the timer A/D converter start request control register (MTU4.TADCR (MTU7.TADCR)) to 0).

When this function is used, MTU4.TADCORA and MTU4.TADCORB (MTU7.TADCORA and MTU7.TADCORB) should be set with the value ranging 0x0002 to the value set in TCDRA minus 2 (value set in TCDRB minus 2).



Note: When the interrupt skipping count is set to two.

Figure 17.95 Example of A/D converter start request signal (TRG4AN) operation linked with interrupt skipping function 1 (UT4AE and DT4AE = 1)



Note: When the interrupt skipping count is set to two.

Figure 17.96 Example of A/D converter start request signal (TRG4AN) operation linked with interrupt skipping function 1 (UT4AE = 1, DT4AE = 0)

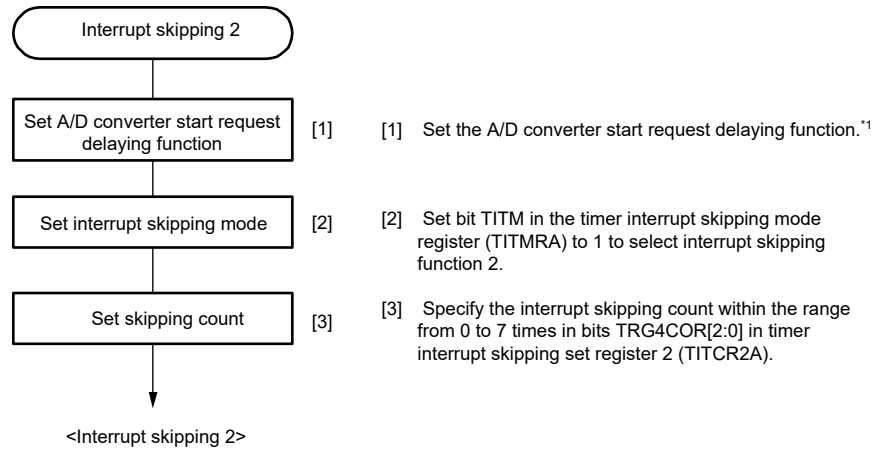
17.3.9.5 A/D Converter Start Request Delaying Function Linked with Interrupt Skipping Function 2

By setting the TITM bit to 1 in the TITMRA (TITMRB) register, the counter starts down-counting from the value (0 to 7) set in the TRG4COR[2:0] (TRG7COR[2:0]) bits in TITCR2A (TITCR2B) register every time an A/D converter start trigger (TRG4AN or TRG4BN (TRG7AN or TRG7BN)) is generated. When the counter value reaches 0 and is reloaded, the TRG4AN and TRG4BN (TRG7AN and TRG7BN) interrupts become valid and an A/D converter start request signal (TRG4ABN (TRG7ABN)) is output.

This function is valid only when the A/D converter request delaying function is enabled.

(1) Example of Procedure for Setting Interrupt Skipping Function 2

Figure 17.97 shows an example of procedure for setting interrupt skipping function 2.



Note 1. For details, see section 17.3.9.1. Example of Procedure for Specifying A/D Converter Start Request Delaying Function.

Figure 17.97 Example of procedure for setting interrupt skipping function 2

(2) Example of Interrupt Skipping Function 2 Operation

Figure 17.98 shows an example of interrupt skipping function 2 operation.

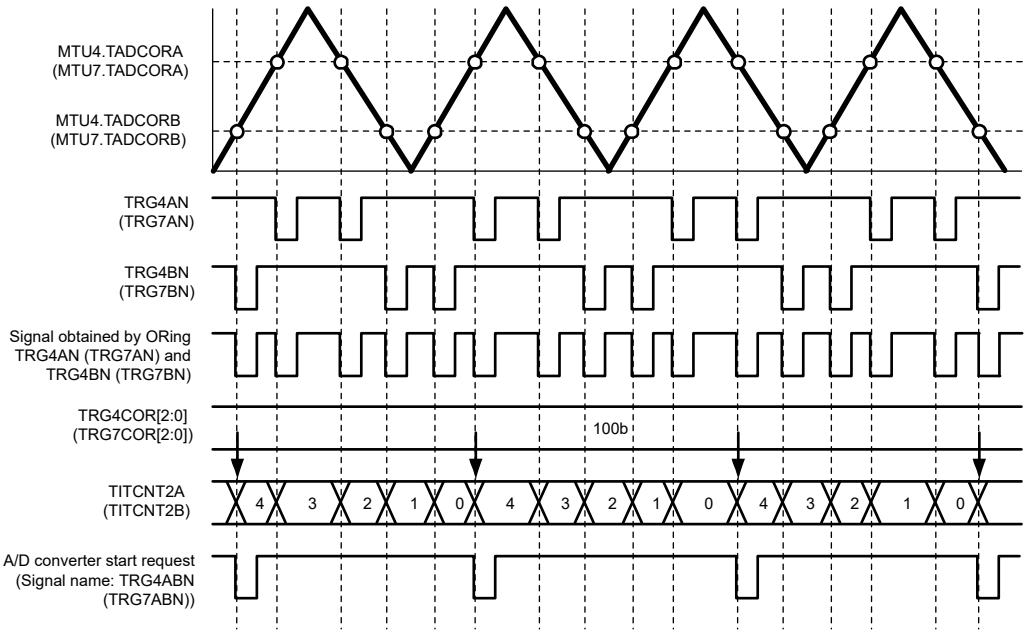


Figure 17.98 Example of interrupt skipping function 2 operation (skipping count is set to four)

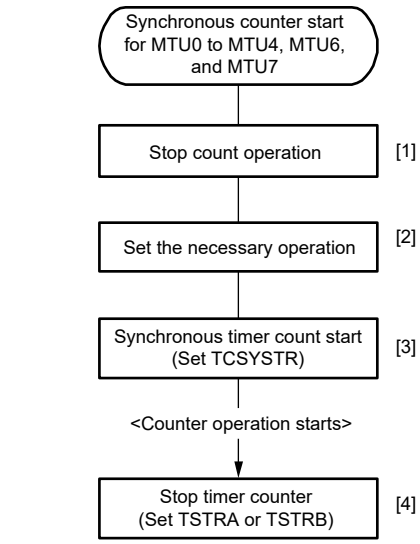
17.3.10 Synchronous Operation of MTU0 to MTU4, MTU6, and MTU7

17.3.10.1 Synchronous Counter Start for MTU0 to MTU4, MTU6, and MTU7

The counters in MTU0 to MTU4, MTU6, and MTU7 can be started synchronously by making the TCSYSTR settings.

(1) Example of Procedure for Setting Synchronous Counter Start for MTU0 to MTU4, MTU6, and MTU7

Figure 17.99 shows an example of procedure for setting synchronous counter start for MTU0 to MTU4, MTU6, and MTU7.



[1] Set TSTRA and TSTRB to stop the counters used for synchronous start operation.

[2] Specify the necessary operation with appropriate registers such as TCR, TMDR1, and TMDR2.

[3] In TCSYSTR, set the bits corresponding to the counters to be started synchronously to 1 at the same time. When TCSYSTR is set, TSTRA and TSTRB for the target timer counters are automatically set appropriately, the counters start synchronously, and TCSYSTR is automatically cleared.

[4] To stop a timer counter, set the bit in TSTRA or TSTRB that corresponds to the counter to 0.

Note:
- Even if a bit in TCSYSTR corresponding to an operating counter is set to 0, the counter will not stop.
- To start counters synchronously in reset-synchronized PWM mode or complementary PWM mode using MTU3 and MTU4 or MTU6 and MTU7, set the bits in TCSYSTR that correspond to MTU3, MTU4, MTU6, and MTU7 to 1 at the same time.

Figure 17.99 Example of procedure for setting synchronous counter start for MTU0 to MTU4, MTU6, and MTU7

(2) Examples of Synchronous Counter Start Operation

Figure 17.100 shows an examples of synchronous counter start operation for MTU0 to MTU4, MTU6, and MTU7.

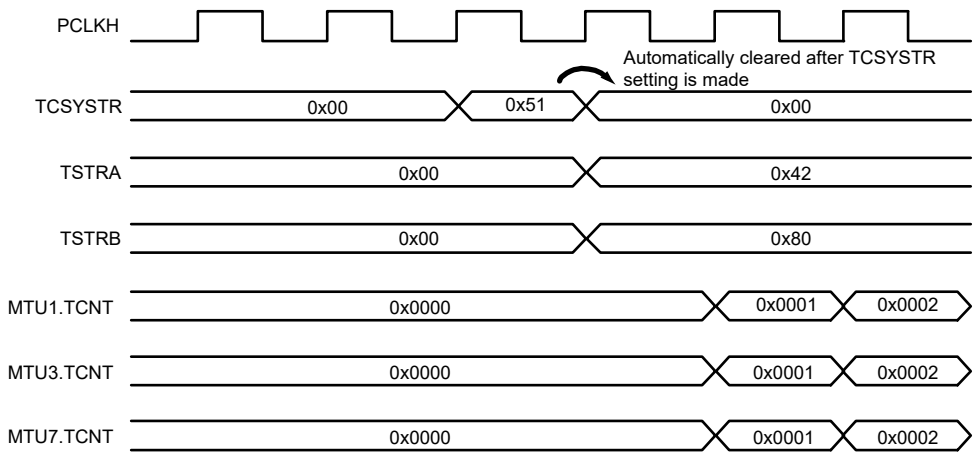


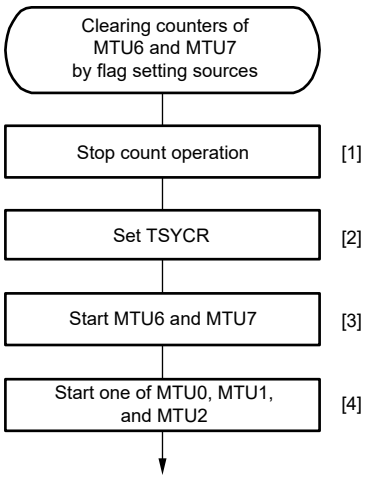
Figure 17.100 Examples of synchronous counter start operation for MTU0 to MTU4, MTU6, and MTU7

17.3.10.2 Synchronous Counter Clearing for MTU6 and MTU7

The counters in MTU6 and MTU7 can be cleared by the TGImn interrupt generation timing (m = A to D; n = 0 to 2) through the TSYCR setting.

(1) Example of Procedure for Specifying Synchronous Counter Clearing for MTU6 and MTU7

Figure 17.101 shows an example of procedure for specifying synchronous counter clearing for MTU6 and MTU7 by flag setting sources.



[1] Set TSTRA and TSTRB to stop the counters used for this function.

[2] Use TSYCR to specify the source to be used for the MTU6.TCNT and MTU7.TCNT counter clearing.

[3] Start MTU6.TCNT or MTU7.TCNT.

[4] Start MTU0.TCNT, MTU1.TCNT, or MTU2.TCNT.

Note: The TSYCR setting is ignored while the counter is stopped. The setting becomes valid after MTU6.TCNT or MTU7.TCNT is started.

Figure 17.101 Example of procedure for specifying synchronous counter clearing for MTU6 and MTU7

(2) Examples of Synchronous Counter Clearing for MTU6 and MTU7

Figure 17.102 and Figure 17.103 show examples of synchronous counter clearing for MTU6 and MTU7 by flag setting sources.

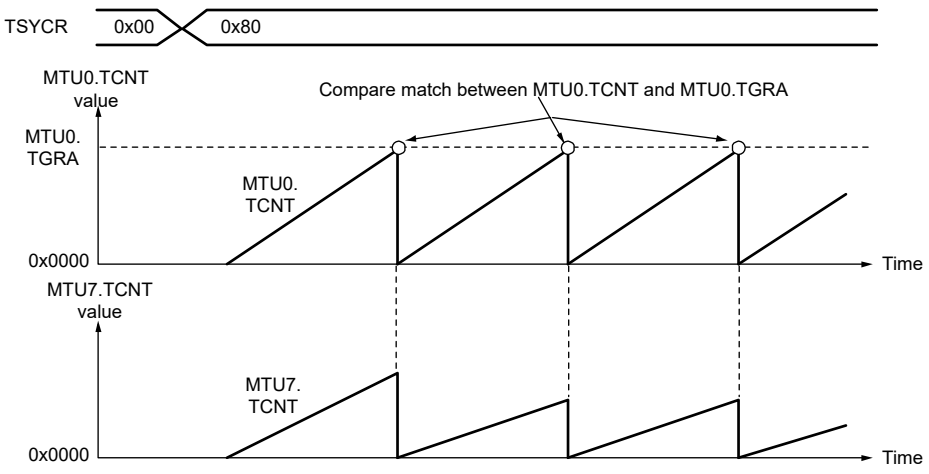


Figure 17.102 Example of synchronous counter clearing for MTU6 and MTU7 (1)

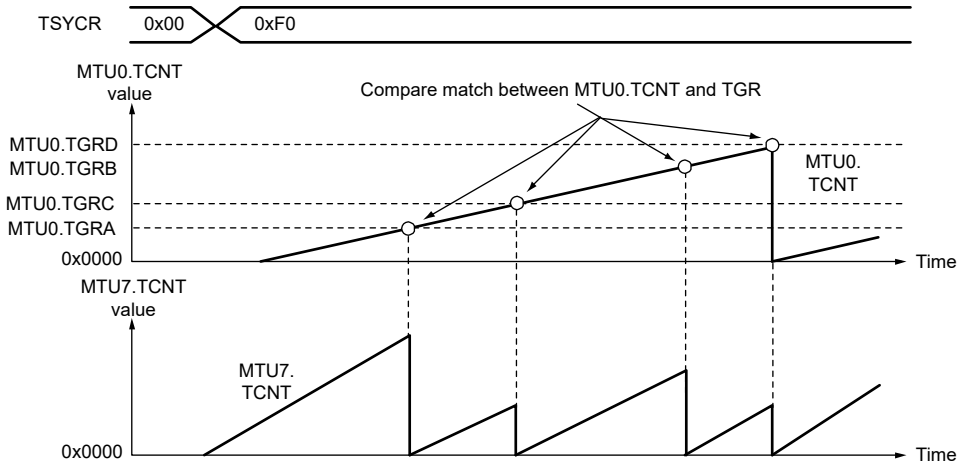


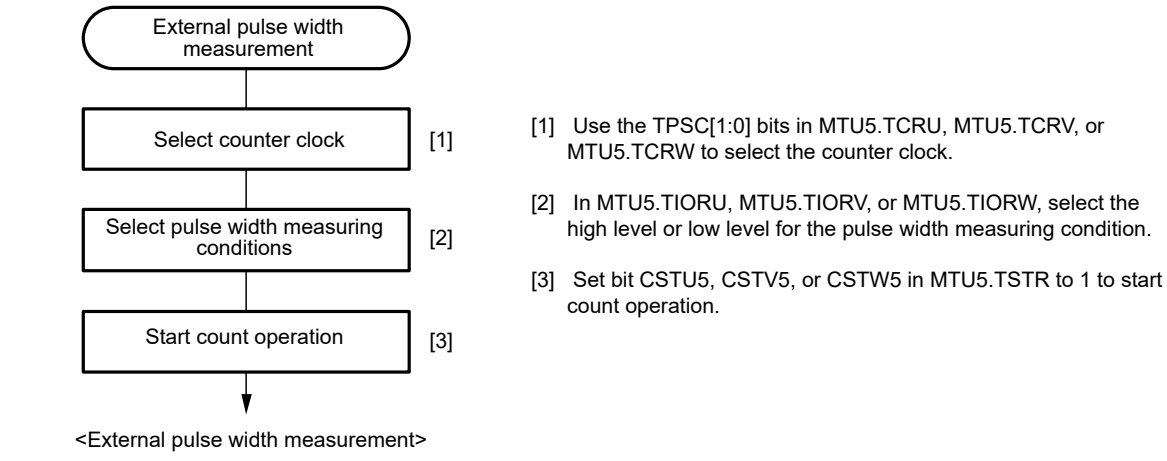
Figure 17.103 Example of synchronous counter clearing for MTU6 and MTU7 (2)

17.3.11 External Pulse Width Measurement

The pulse widths of up to three external input lines can be measured in MTU5.

When the IOC[4:0] bits in MTU5.TIORU, MTU5.TIORV, MTU5.TIORW are set for pulse width measurement, the pulse width of the signal input to the MTIC5U, MTIC5V, and MTIC5W pins are measured. TCNTU, TCNTV, and TCNTW count up while the level specified by the IOC[4:0] bits is input.

Figure 17.104 shows an example of setting external pulse width measurement and Figure 17.105 an example of external pulse width measurement.



Note:
- Set bit CMPCLR5U, CMPCLR5V, and CMPCLR5W in TCNTCMPCLR to 0.
- Set bit TGIE5U, TGIE5V, and TGIE5W in MTU5.TIER to 0.
- The value in TCNT is not captured in TGR.

Figure 17.104 Example of external pulse width measurement setting procedure

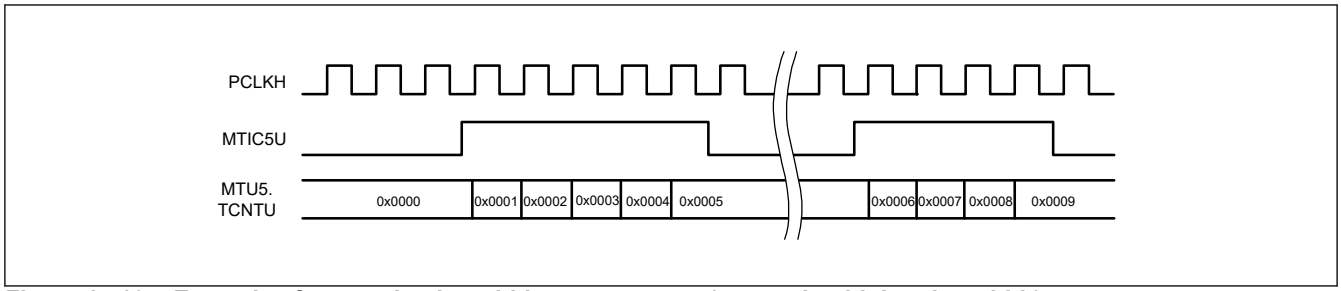


Figure 17.105 Example of external pulse width measurement (measuring high pulse width)

17.3.12 Dead Time Compensation

A dead time delay (a propagation delay of the inverter output from the complementary PWM output) can be compensated by combining MTU5 with MTU6 and MTU7. Figure 17.106 shows an example of the motor control circuit compensating a dead time delay by combining MTU5 with MTU6 and MTU7. A dead time for the PWM output waveform during complementary PWM operation using MTU6 and MTU7 can be compensated by adjusting a duty ratio set in a compare register for the PWM output after measuring a delay of the inverter output from the complementary PWM output by an external pulse measurement function for MTU5 (Figure 17.107). Figure 17.108 shows the procedure for setting dead time compensation using MTU5 to MTU7. For details on MTU5 operation at this time, see section 17.3.13. TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode.

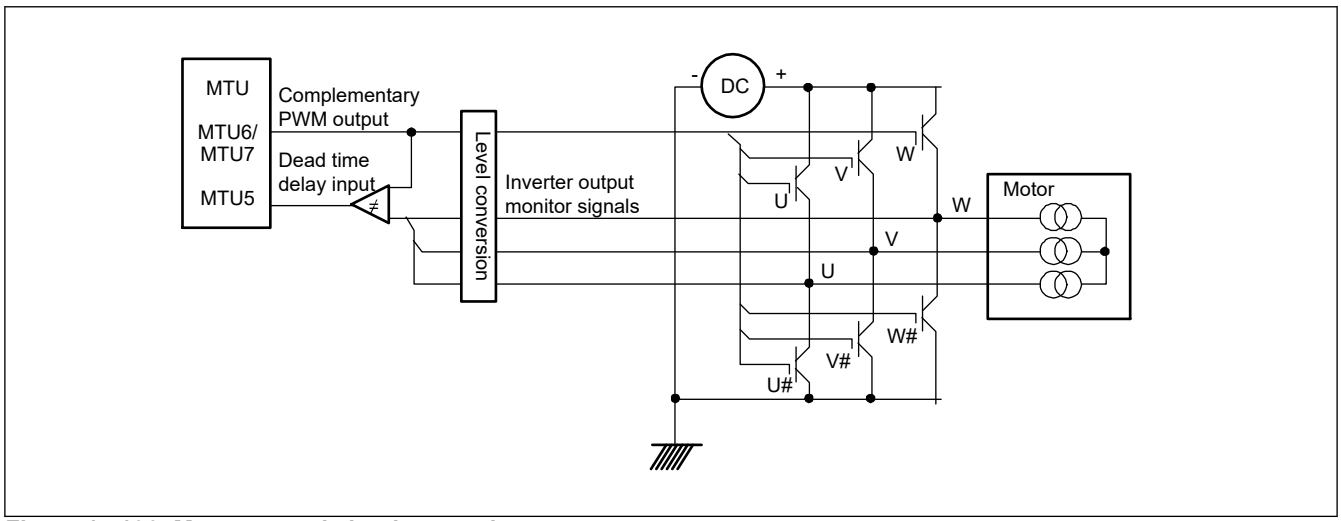


Figure 17.106 Motor control circuit example

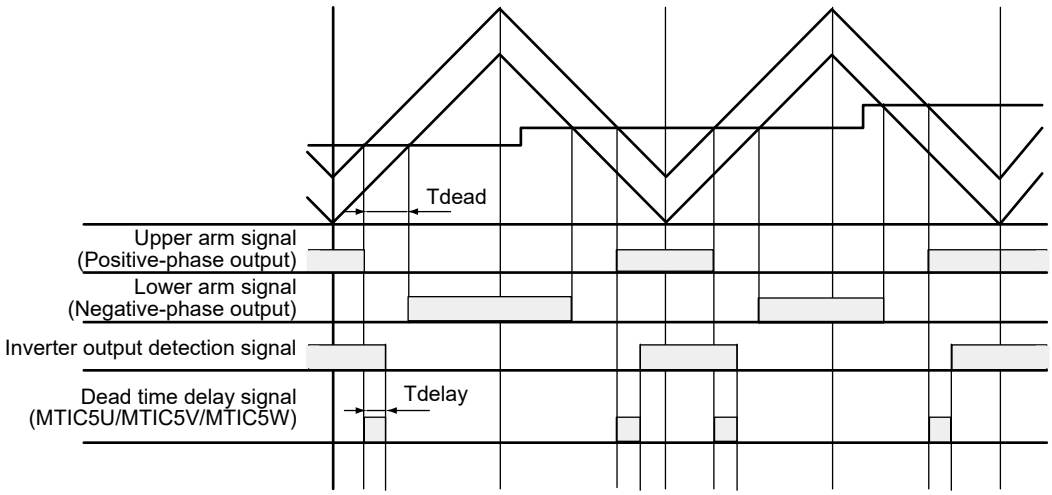
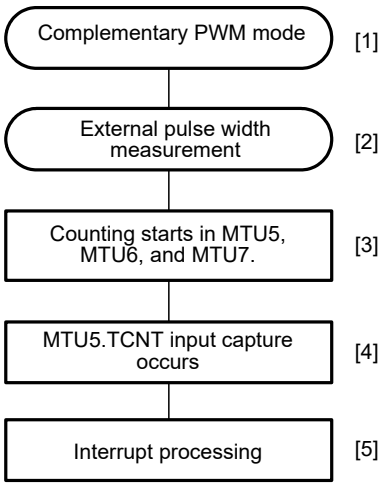


Figure 17.107 Delay in dead time in complementary PWM operation

17.3.12.1 Example of Dead Time Compensation Setting Procedure

Figure 17.108 shows an example of dead time compensation setting procedure by using three counters in MTU5.



[1] Set MTU6 and MTU7 to complementary PWM mode. For details, refer to section 17.3.8, Complementary PWM Mode.

[2] Specify the external pulse width measurement function for the target TIORx (x: U, V, W) in MTU5. For details, refer to section 17.3.11, External Pulse Width Measurement.

[3] Set bits CST6 and CST7 in TSTRB and bits CSTU5, CSTV5, and CSTW5 in MTU5.TSTR to 1 to start count operation.

[4] When the capture condition specified in TIORx is satisfied, the MTU5.TCNT value is captured in MTU5.TGRx (x: U, V, W). To prevent the generation of the MTU5.TGIm5 interrupt (m = U, V, W) request under the capture condition specified in TIORx, do not enable interrupt requests in MTU5.TIER.

[5] For U-phase dead time compensation, when an interrupt is generated at the crest (TGIA6) or trough (TCIV7) in complementary PWM mode, read the MTU5.TGRU value, calculate the difference in time between MTU6.TGRB and write the corrected value to MTU6.TGRD in the interrupt processing. For the V phase and W phase, read the MTU5.TGRV and MTU5.TGRW values and write the corrected values to MTU7.TGRC and TGRD, respectively, in the same way as for U-phase compensation. The MTU5.TCNT value should be cleared through the TCNTCMPCLR setting or by software.

Note: The I/O port registers of the I/O ports should be set up in advance.
As the interrupt flag is set under the capture condition specified in TIORx, do not enable interrupt requests in MTU5.TIER.

Figure 17.108 Example of dead time compensation setting procedure

17.3.13 TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode

The MTU5 external pulse width measurement function allows to transfer the value in TCNTU, TCNTV, and TCNTW to TGRU, TGRV, and TGRW at the crest, or trough, or crest and trough when MTU6 and MTU7 operate in complementary PWM mode. The transfer timing is set in TIORU, TIORV, and TIORW. When the CMPCLR5U, CMPCLR5V, and CMPCLR5W bits in the TCNTCMPCLR register are set to 1, TCNTU, TCNTV, and TCNTW become 0x0000 at the transfer timing for TGRU, TGRV, and TGRW.

Figure 17.109 shows an operation example in which MTU5.TCNTU is used as a free-running counter without being cleared, and the value is captured in TGR at the crest and trough in complementary PWM mode.

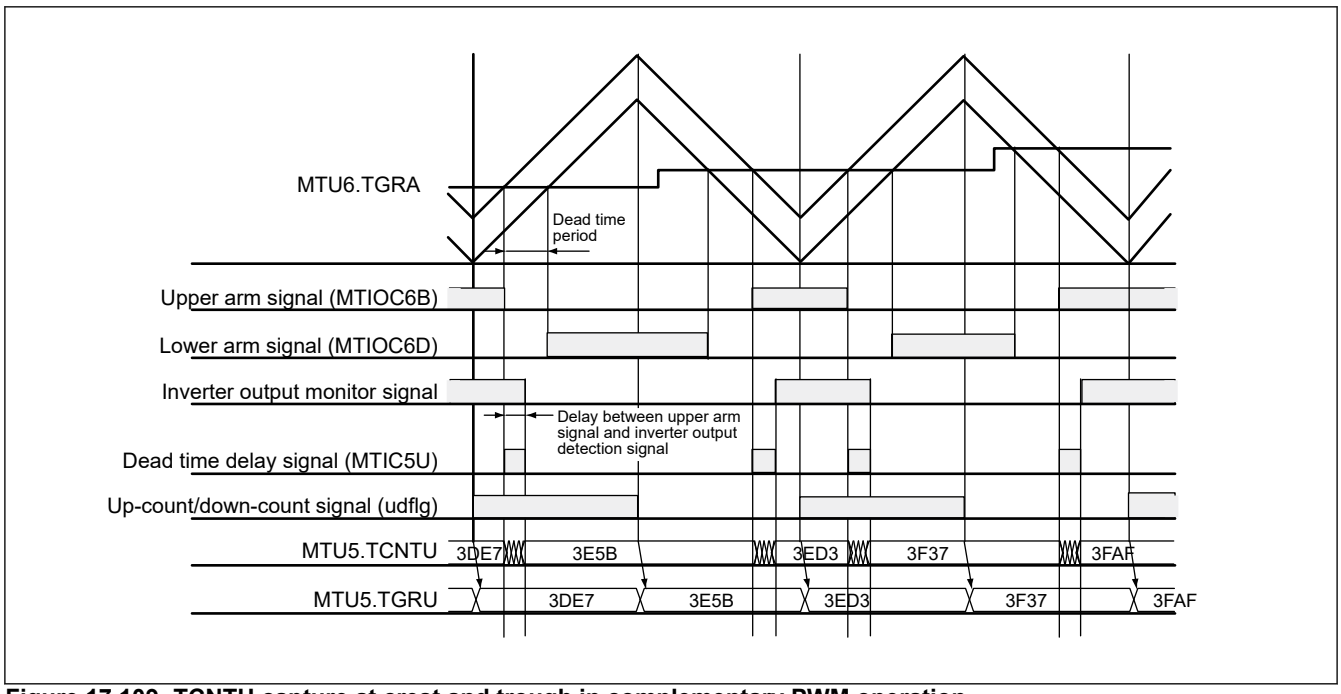


Figure 17.109 TCNTU capture at crest and trough in complementary PWM operation

17.3.14 Noise Filter Function

The input capture input pins and external pulse input pins have a noise filter function.

Set the NFCRn register (n = 0 to 8, C) to enable or disable the noise filter function and set the sampling clock. The noise filter for each pin can be enabled or disabled individually, and the sampling clock can be set for each channel.

Figure 17.110 shows the timing of noise filtering.

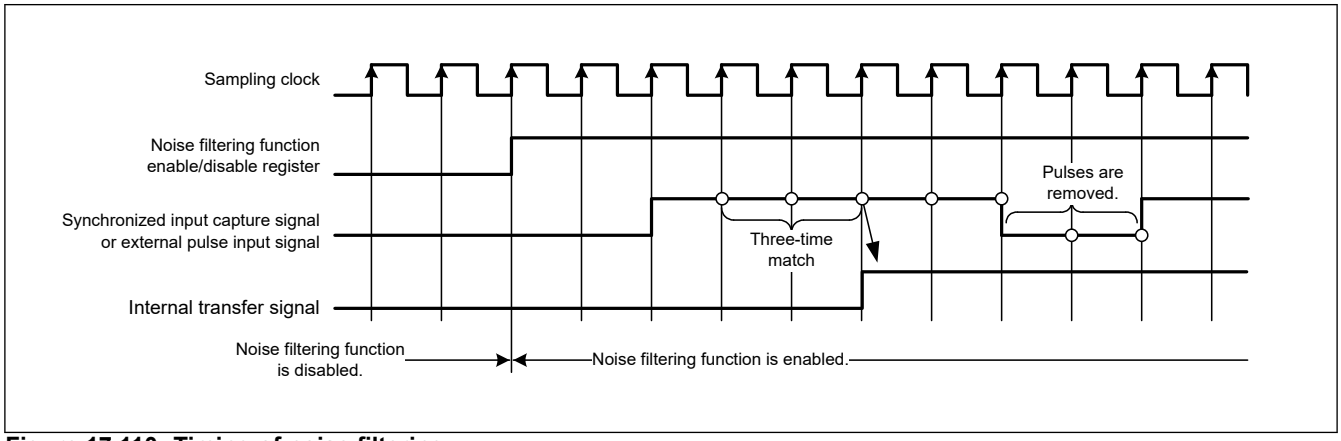


Figure 17.110 Timing of noise filtering

17.4 Interrupt Sources

17.4.1 Interrupt Sources and Priorities

There are three kinds of MTU interrupt sources:

- TGR input capture/compare match
- TCNT overflow
- TCNT underflow

Each interrupt source has its own enable/disable bit, allowing the generation of interrupt request signals to be enabled or disabled individually.

When an interrupt source is generated, if the corresponding enable/disable bit in TIER is set to 1, an interrupt is requested.

Relative channel priorities can be changed by the interrupt controller. For details, see section 11, Interrupt Controller (ICU). Table 17.72 lists the MTU interrupt sources.

Table 17.72 MTU interrupt sources

Channel	Name	Interrupt source	DMAC activation	Priority
MTU0	TGIA0	ch0 input capture/compare match A interrupt	Possible	High
	TGIB0	ch0 input capture/compare match B interrupt	Possible	↑
	TGIC0	ch0 input capture/compare match C interrupt	Possible	↑
	TGID0	ch0 input capture/compare match D interrupt	Possible	↑
	TCIV0	ch0 overflow interrupt	Not possible	↑
	TGIE0	ch0 compare match E interrupt	Not possible	↑
	TGIF0	ch0 compare match F interrupt	Not possible	↑
MTU1	TGIA1	ch1 input capture/compare match A interrupt	Possible	↑
	TGIB1	ch1 input capture/compare match B interrupt	Possible	↑
	TCIV1	ch1 overflow interrupt	Not possible	↑
	TCIU1	ch1 underflow interrupt	Not possible	↑
MTU2	TGIA2	ch2 input capture/compare match A interrupt	Possible	↑
	TGIB2	ch2 input capture/compare match B interrupt	Possible	↑
	TCIV2	ch2 overflow interrupt	Not possible	↑
	TCIU2	ch2 underflow interrupt	Not possible	↑
MTU3	TGIA3	ch3 input capture/compare match A interrupt	Possible	↑
	TGIB3	ch3 input capture/compare match B interrupt	Possible	↑
	TGIC3	ch3 input capture/compare match C interrupt	Possible	↑
	TGID3	ch3 input capture/compare match D interrupt	Possible	↑
	TCIV3	ch3 overflow interrupt	Not possible	↑
MTU4	TGIA4	ch4 input capture/compare match A interrupt	Possible	↑
	TGIB4	ch4 input capture/compare match B interrupt	Possible	↑
	TGIC4	ch4 input capture/compare match C interrupt	Possible	↑
	TGID4	ch4 input capture/compare match D interrupt	Possible	↑
	TCIV4	ch4 overflow/underflow interrupt*1	Possible	↑
MTU5	TGIU5	ch5 input capture/compare match U interrupt	Possible	↑
	TGIV5	ch5 input capture/compare match V interrupt	Possible	↑
	TGIW5	ch5 input capture/compare match W interrupt	Possible	↑
MTU6	TGIA6	ch6 compare match A interrupt	Possible	↑
	TGIB6	ch6 compare match B interrupt	Possible	↑
	TGIC6	ch6 compare match C interrupt	Possible	↑
	TGID6	ch6 compare match D interrupt	Possible	↑
	TCIV6	ch6 overflow interrupt	Not possible	↑
MTU7	TGIA7	ch7 compare match A interrupt	Possible	↑
	TGIB7	ch7 compare match B interrupt	Possible	↑
	TGIC7	ch7 compare match C interrupt	Possible	↑
	TGID7	ch7 compare match D interrupt	Possible	↑
	TCIV7	ch7 overflow/underflow interrupt*1	Possible	↑
MTU8	TGIA8	ch8 compare match A interrupt	Possible	↑
	TGIB8	ch8 compare match B interrupt	Possible	↑
	TGIC8	ch8 compare match C interrupt	Possible	↑
	TGID8	ch8 compare match D interrupt	Possible	↑
	TCIV8	ch8 overflow interrupt	Not possible	Low

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.
Note 1. Only in complementary PWM mode.

(1) Input Capture/Compare Match Interrupt

If the TIER.TGIE bit is set to 1 when a TGR input capture/compare match occurs on a channel, an interrupt is requested. The MTU has 33 input capture/compare match interrupts (six for MTU0, four each for MTU3, MTU4, MTU6, MTU7, and MTU8, two each for MTU1 and MTU2, and three for MTU5).

(2) Overflow Interrupt

If the TIER.TCIEV bit is set to 1 when a TCNT overflow occurs on a channel, an interrupt is requested. The MTU has eight overflow interrupts (one for each channel).

(3) Underflow Interrupt

If the TIER.TCIEU bit is set to 1 when a TCNT underflow occurs on a channel, an interrupt is requested. The MTU has two underflow interrupts (one each for MTU1 and MTU2).

17.4.2 DMAC Trigger Sources

The DMAC can be triggered by the TGR input capture/compare match interrupt in each channel and the overflow interrupt in MTU4 and MTU7. For details, see section 14, DMA Controller (DMAC).

The MTU provides a total of 33 input capture/compare match interrupts and overflow interrupts that can be used as DMAC trigger sources: four each for MTU0, MTU3, MTU6, and MTU8, two each for MTU1 and MTU2, five each for MTU4 and MTU7, and three for MTU5.

17.4.3 A/D Converter Trigger Sources

The A/D converter can be triggered by one of the following three methods in the MTU. Table 17.73 shows the relationship between interrupt sources and A/D converter start request signals.

(1) A/D Conversion Start by MTUn.TGRA (n = 0 to 4, 6, 7) Input Capture/Compare Match or at Trough of MTUm.TCNT (m = 4 or 7) in Complementary PWM Mode

The A/D converter can be triggered by the occurrence of a TGRA input capture/compare match in channels 0 to 4, 6, and 7. In addition, if complementary PWM operation is performed while the TTGE2 bit in MTU4.TIER is set to 1, the A/D converter can be triggered at the trough of MTUm.TCNT count (MTUm.TCNT = 0x0000) (m = 4 or 7).

A/D converter start request signal TRGAnN is issued to the A/D converter under either of the following conditions (n = 0 to 4, 6, 7).

- When a TGRA input capture/compare match occurs on a channel while the TIER.TTGE bit is set to 1
- When the MTUm.TCNT count reaches the trough (MTUm.TCNT = 0x0000) during complementary PWM operation while the TTGE2 bit in MTUm.TIER is set to 1

When either condition is satisfied, if A/D converter start signal TRGAnN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(2) A/D Conversion Start by Compare Match between MTU0.TCNT and MTU0.TGRE

A/D converter start request signal TRG0N is issued to the A/D converter when a compare match occurs between MTU0.TCNT and MTU0.TGRE.

When a compare match occurs between MTU0.TCNT and MTU0.TGRE while the TTGE2 bit in MTU0.TIER2 is set to 1, A/D converter start request TRG0N is issued to the A/D converter. If A/D converter start signal TRG0N from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

(3) A/D Conversion Start by A/D Converter Start Request Delaying Function (MTU4 and MTU7)

The A/D converter can be triggered by generating A/D converter start request signal TRG4AN or TRG4BN (TRG7AN or TRG7BN) when the MTU4.TCNT (MTU7.TCNT) count matches the MTU4.TADCORA or MTU4.TADCORB (MTU7.TADCORA or MTU7.TADCORB) value if the UT4AE, DT4AE, UT4BE, or DT4BE (UT7AE, DT7AE, UT7BE, or DT7BE) bit in the A/D converter start request control register (MTU4.TADCR (MTU7.TADCR)) is set to 1. For details, see section 17.3.9. A/D Converter Start Request Delaying Function.

A/D conversion will start when TRG4AN (TRG7AN) is generated if A/D converter start signal TRG4AN (TRG7AN) from the MTU is selected as the trigger in the A/D converter, when TRG4BN (TRG7BN) is generated if TRG4BN (TRG7BN) from the MTU is selected as the trigger in the A/D converter, or when TRG4ABN (TRG7ABN) is generated if TRG4ABN (TRG7ABN) from the MTU is selected as the trigger in the A/D converter.

Table 17.73 Interrupt sources and A/D converter start request signals

Target registers	Interrupt source	A/D converter start request signal
MTU0.TGRA and MTU0.TCNT	Input capture/compare match	TRGA0N
MTU1.TGRA and MTU1.TCNT		TRGA1N
MTU2.TGRA and MTU2.TCNT		TRGA2N
MTU3.TGRA and MTU3.TCNT		TRGA3N
MTU4.TGRA and MTU4.TCNT*1		TRGA4N
MTU4.TCNT	MTU4.TCNT trough in complementary PWM mode	
MTU6.TGRA and MTU6.TCNT	Input capture/compare match	TRGA6N
MTU7.TGRA and MTU7.TCNT*1		TRGA7N
MTU7.TCNT	MTU7.TCNT trough in complementary PWM mode	TRGA7N
MTU0.TGRE and MTU0.TCNT	Compare match	TRG0N
MTU4.TADCORA and MTU4.TCNT		TRG4AN
MTU4.TADCORB and MTU4.TCNT		TRG4BN
MTU7.TADCORA and MTU7.TCNT		TRG7AN
MTU7.TADCORB and MTU7.TCNT		TRG7BN
MTU4.TADCORA and MTU4.TCNT, MTU4.TADCORB and MTU4.TCNT	Compare match (interrupt skipping function 2)	TRG4ABN
MTU7.TADCORA and MTU7.TCNT, MTU7.TADCORB and MTU7.TCNT		TRG7ABN

Note 1. Since PWM waveforms are generated in complementary PWM mode, MTU4.TGRA (MTU7.TGRA) compare match not only with MTU4.TCNT (MTU7.TCNT) but also with MTU3.TCNT (MTU6.TCNT) and TCNTSA (TCNTSB) is detected. Accordingly, when compare match with MTU3.TCNT (MTU6.TCNT) and TCNTSA (TCNTSB) occurs, TRGA4N (TRGA7N) is also generated. When MTU3 and MTU 4 (MTU 6 and MTU7) are made to operate in complementary PWM mode for generating an A/D converter start request, use the A/D converter start request by compare match between MTU4.TCNT (MTU7.TCNT) and MTU4.TADCORA/TADCORB (MTU7.TADCORA/TADCORB).

17.5 Operation Timing

17.5.1 Input/Output Timing

17.5.1.1 TCNT Count Timing

Figure 17.111 and Figure 17.112 show the TCNT count timing in internal clock operation, Figure 17.113 shows the TCNT count timing in external clock operation (normal mode), and Figure 17.114 shows the TCNT count timing in external clock operation (phase counting mode).

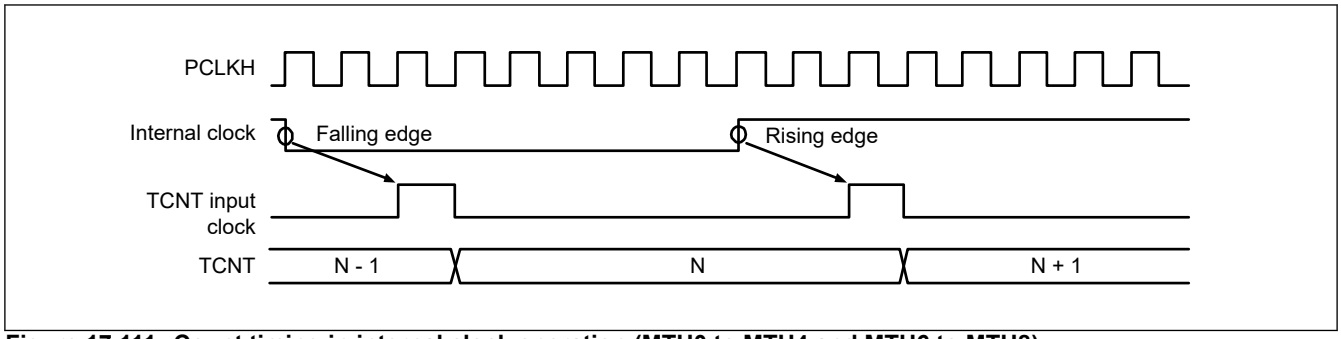


Figure 17.111 Count timing in internal clock operation (MTU0 to MTU4 and MTU6 to MTU8)

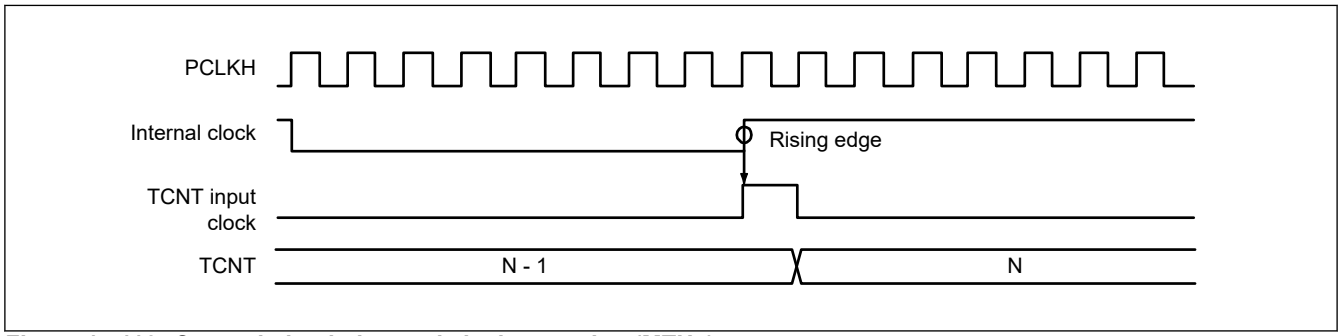


Figure 17.112 Count timing in internal clock operation (MTU5)

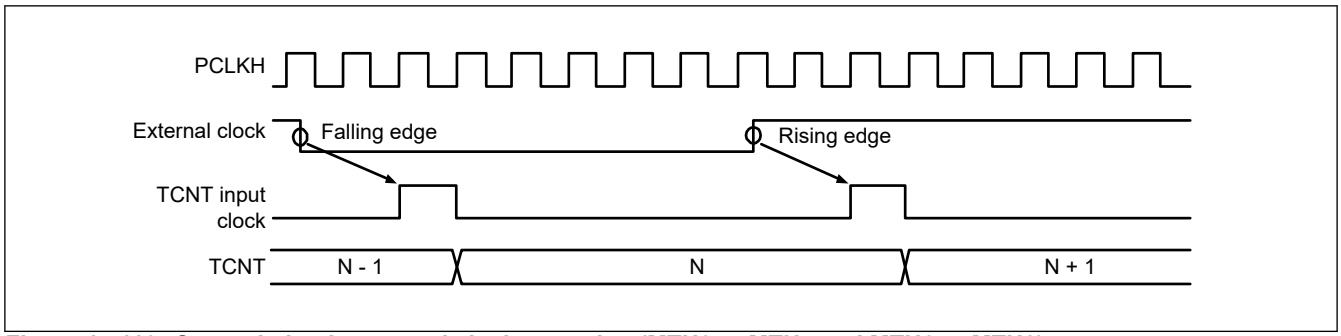


Figure 17.113 Count timing in external clock operation (MTU0 to MTU4 and MTU6 to MTU8)

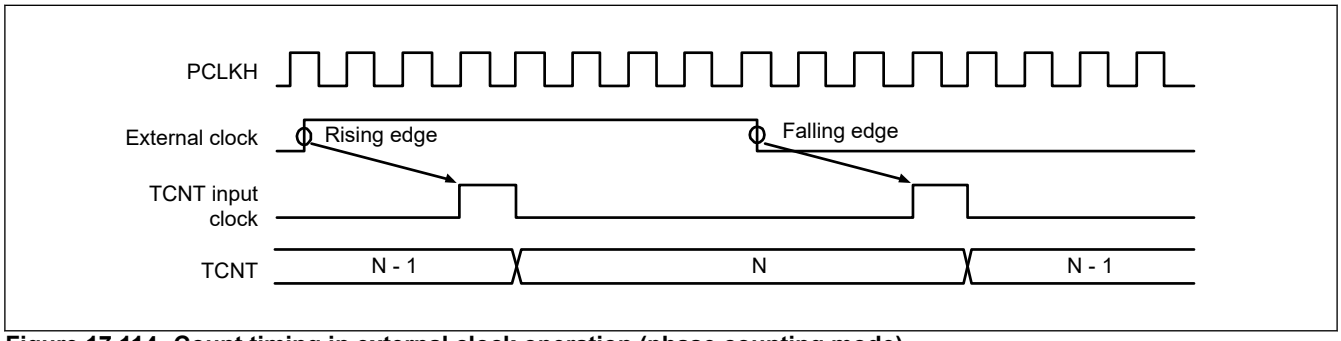


Figure 17.114 Count timing in external clock operation (phase counting mode)

17.5.1.2 Output Compare Output Timing

A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the value set in TIOR is output from MTIOCnm pin (n = 0 to 4, 6, 7, 8; m = A to D). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 17.115 shows the output compare output timing (normal mode or PWM mode) and Figure 17.116 shows the output compare output timing (complementary PWM mode or reset-synchronized PWM mode).

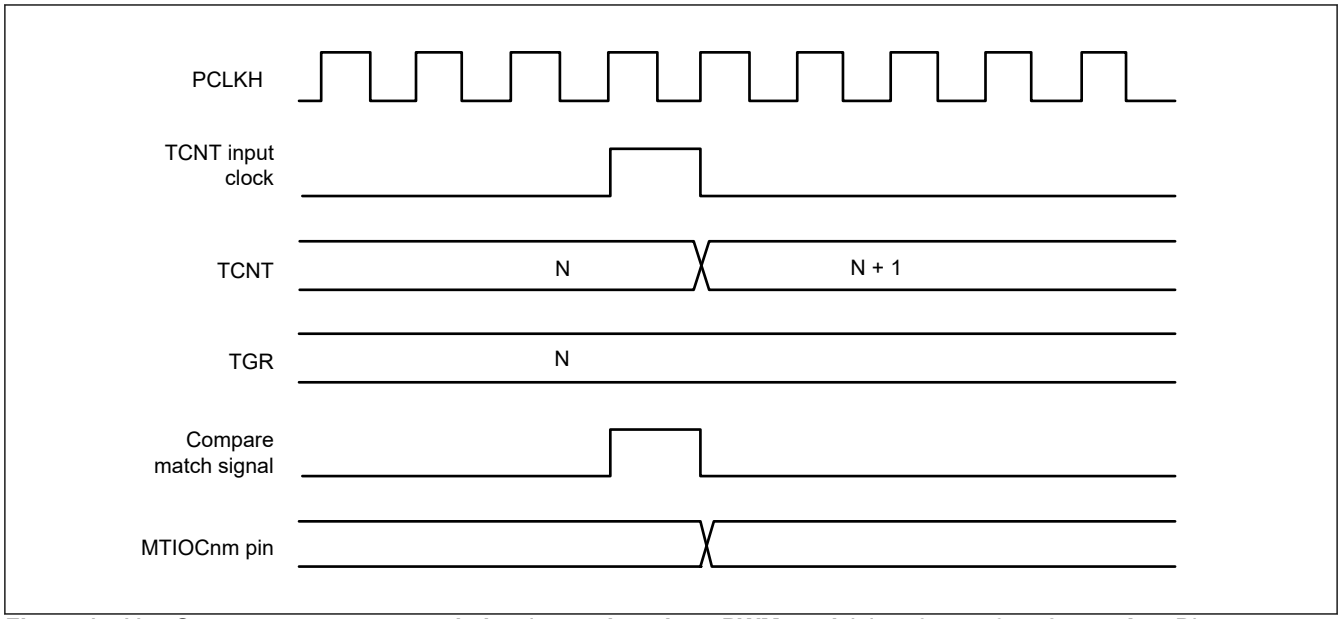


Figure 17.115 Output compare output timing (normal mode or PWM mode) (n = 0 to 4, 6, 7, 8; m = A to D)

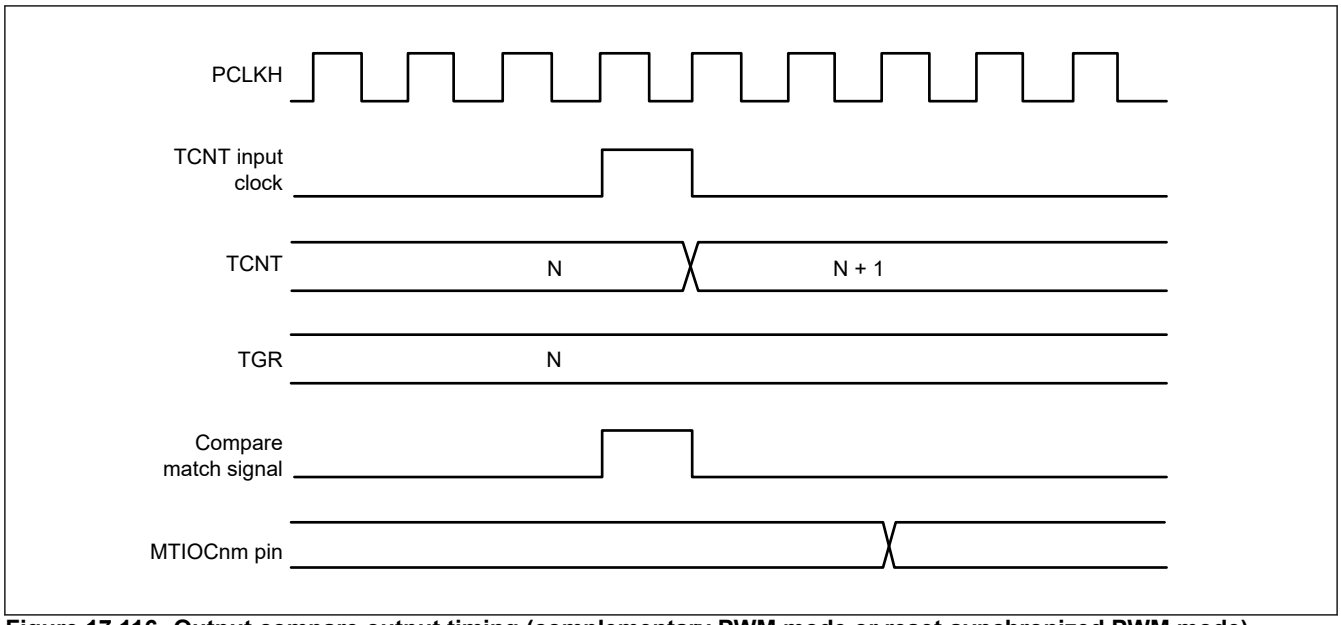


Figure 17.116 Output compare output timing (complementary PWM mode or reset-synchronized PWM mode) (n = 3, 4, 6, 7; m = A to D)

17.5.1.3 Input Capture Signal Timing

Figure 17.117 shows the input capture signal timing.

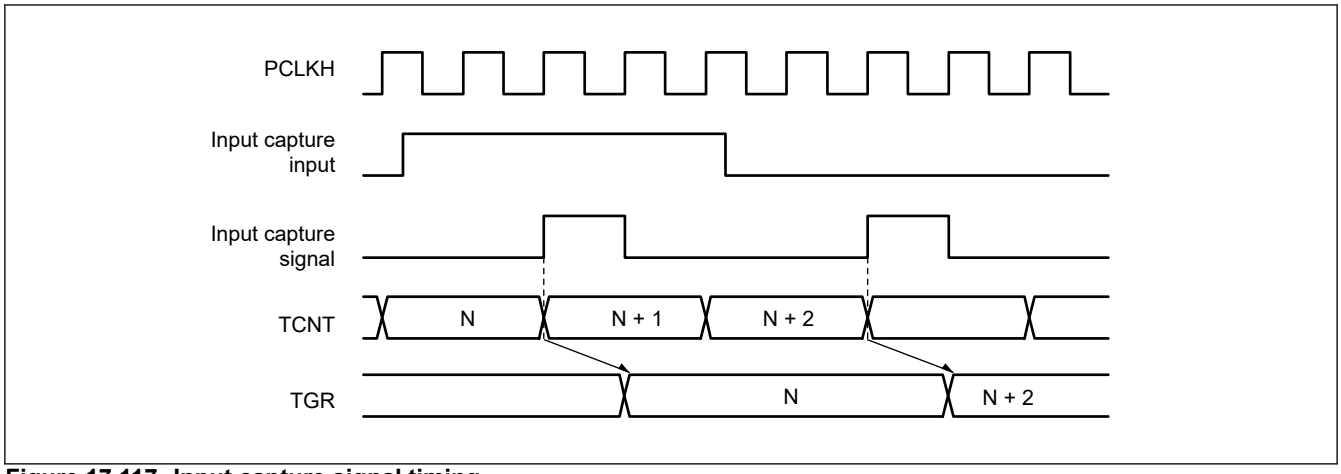


Figure 17.117 Input capture signal timing

17.5.1.4 Timing for Counter Clearing by Compare Match/Input Capture

Figure 17.118 and Figure 17.119 show the timing when counter clearing on compare match is specified, and Figure 17.120 shows the timing when counter clearing on input capture is specified.

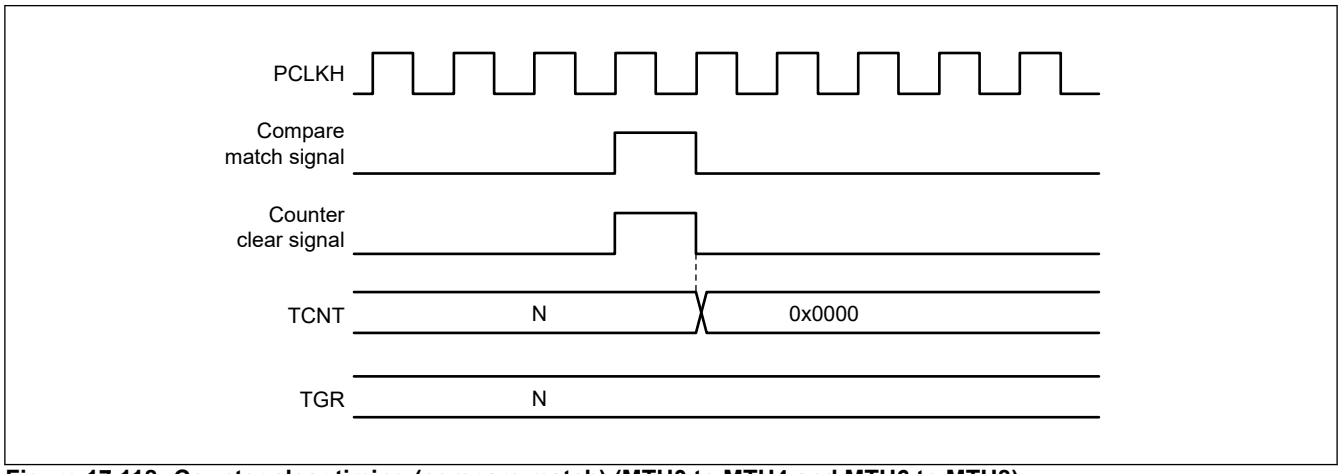


Figure 17.118 Counter clear timing (compare match) (MTU0 to MTU4 and MTU6 to MTU8)

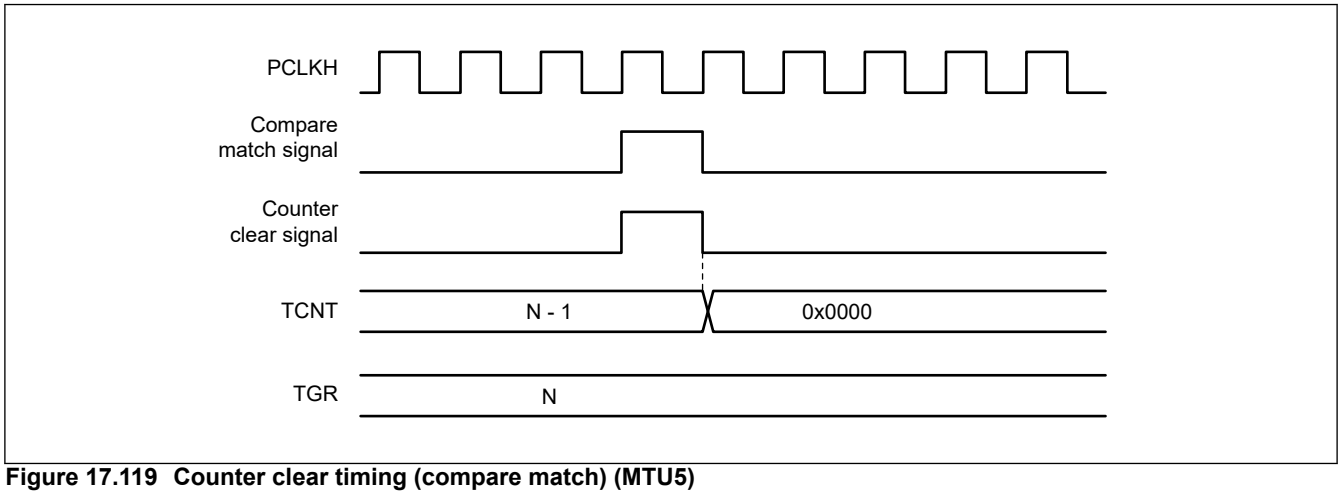


Figure 17.119 Counter clear timing (compare match) (MTU5)

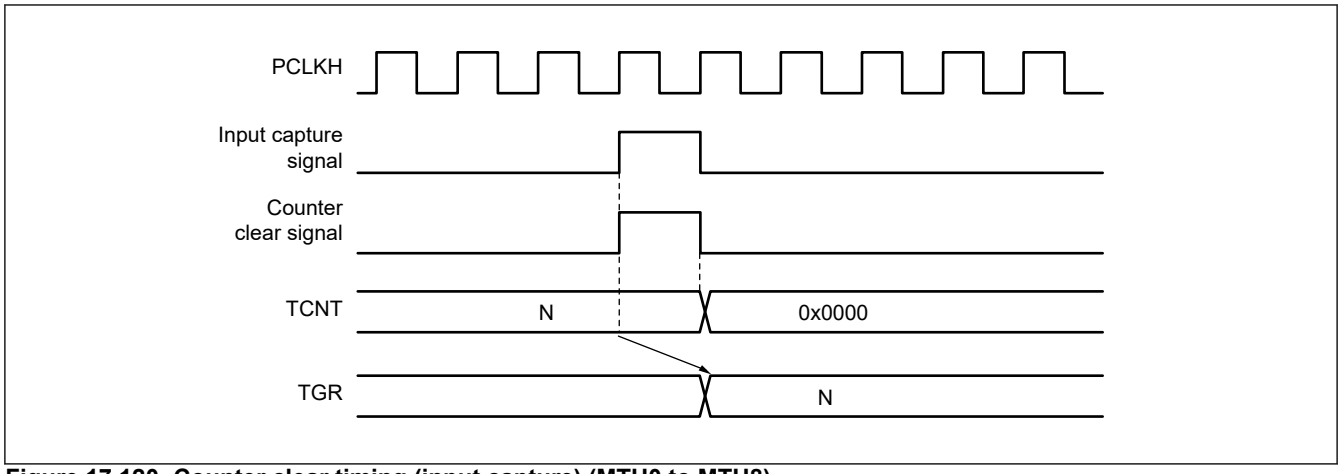


Figure 17.120 Counter clear timing (input capture) (MTU0 to MTU8)

17.5.1.5 Buffer Operation Timing

Figure 17.121 to Figure 17.123 show the timing in buffer operation.

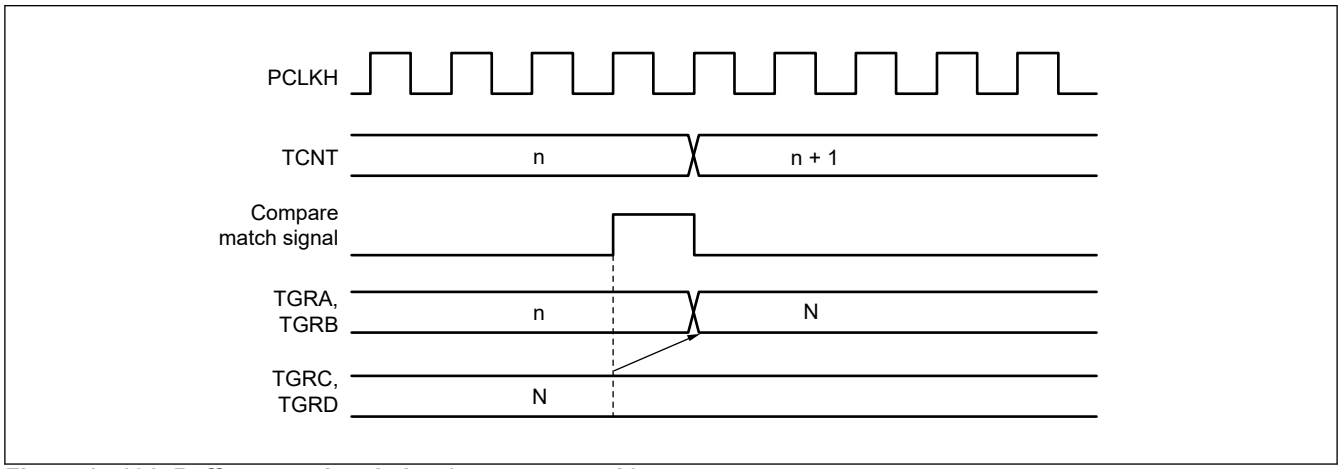


Figure 17.121 Buffer operation timing (compare match)

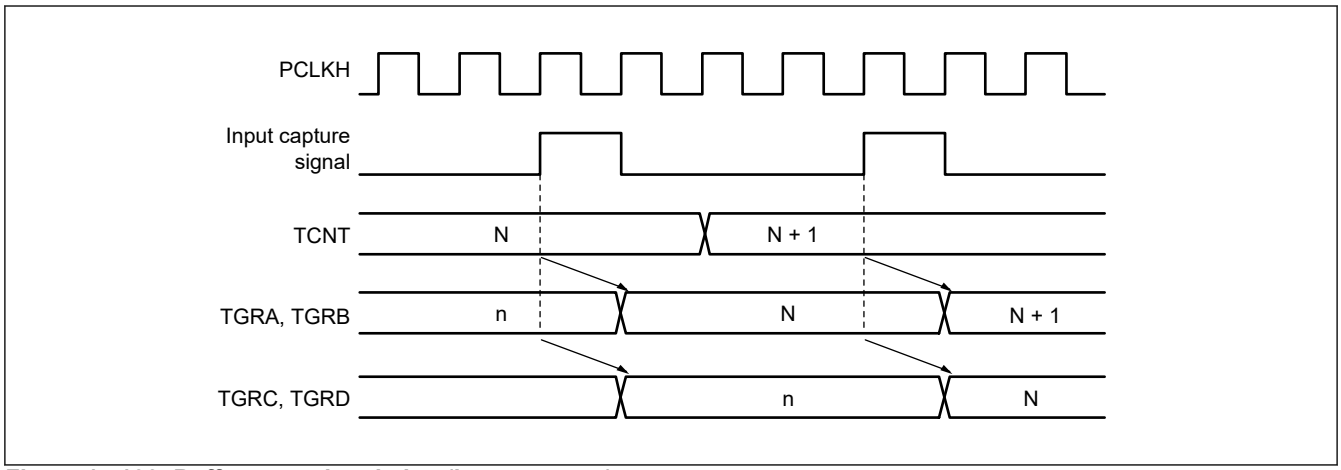


Figure 17.122 Buffer operation timing (input capture)

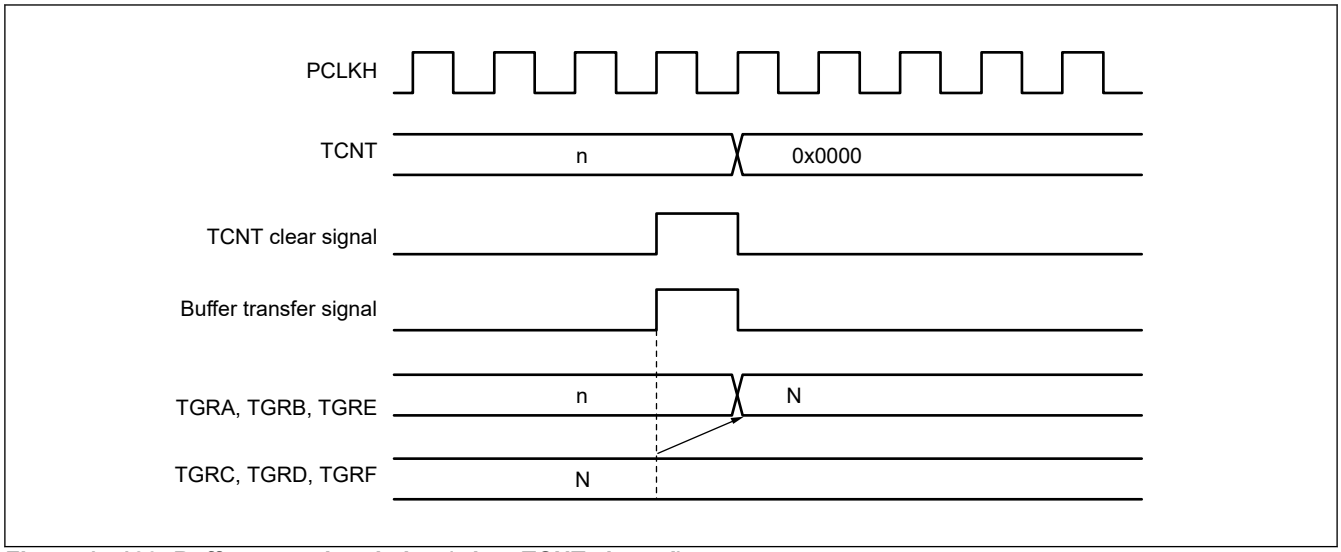


Figure 17.123 Buffer operation timing (when TCNT cleared)

17.5.1.6 Buffer Transfer Timing (Complementary PWM Mode)

Figure 17.124 to Figure 17.126 show the buffer transfer timing in complementary PWM mode.

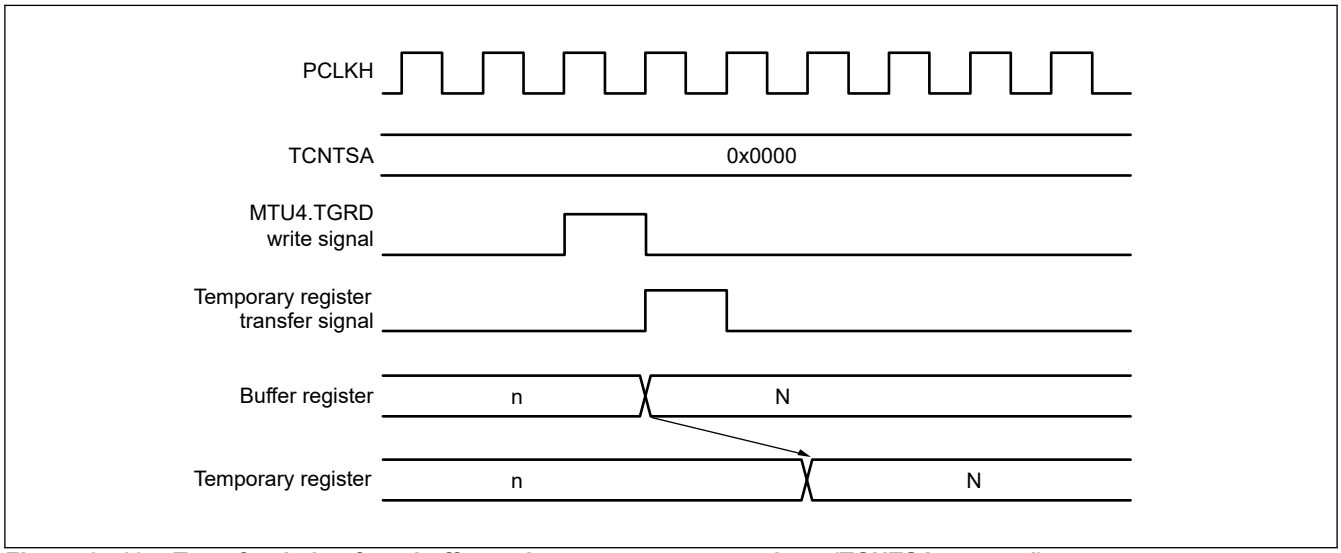


Figure 17.124 Transfer timing from buffer register to temporary register (TCNTSA stopped)

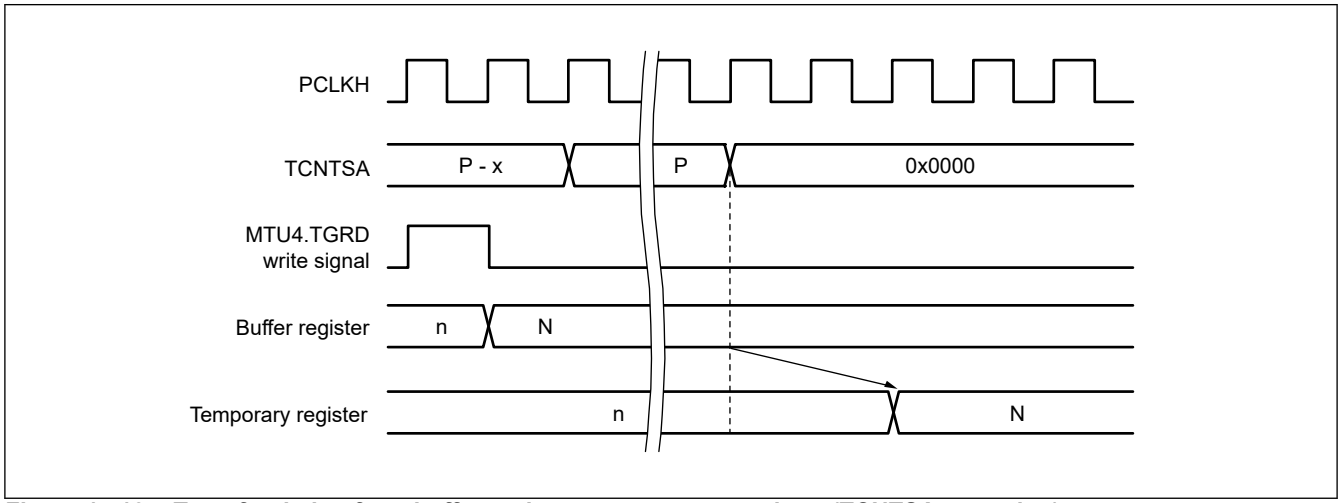


Figure 17.125 Transfer timing from buffer register to temporary register (TCNTSA operating)

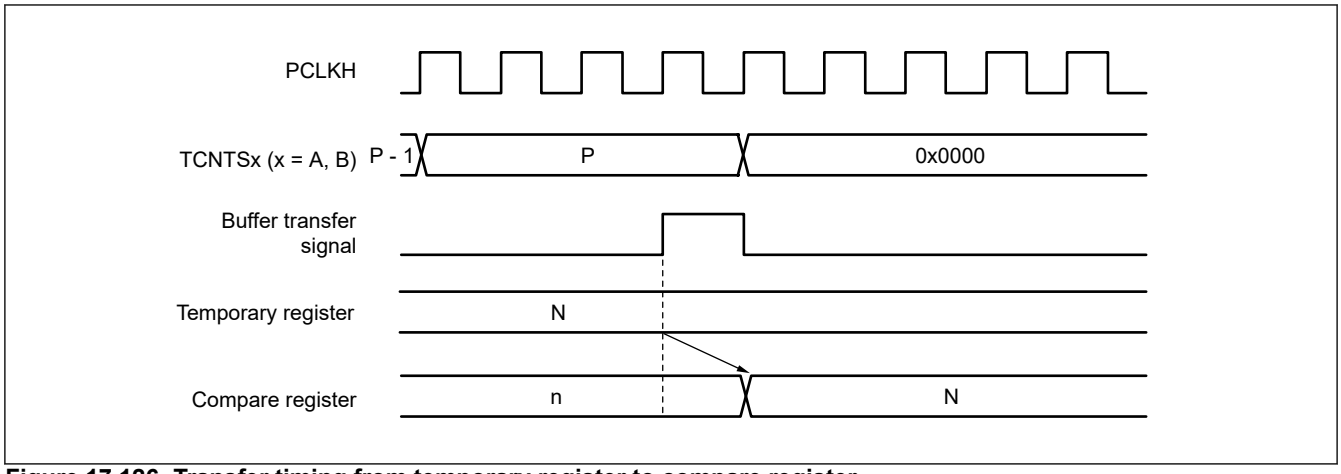


Figure 17.126 Transfer timing from temporary register to compare register

17.5.2 Interrupt Signal Timing

17.5.2.1 TGI Interrupt Timing by Compare Match

Figure 17.127 and Figure 17.128 show the TGI interrupt request signal timing when a compare match occurs.

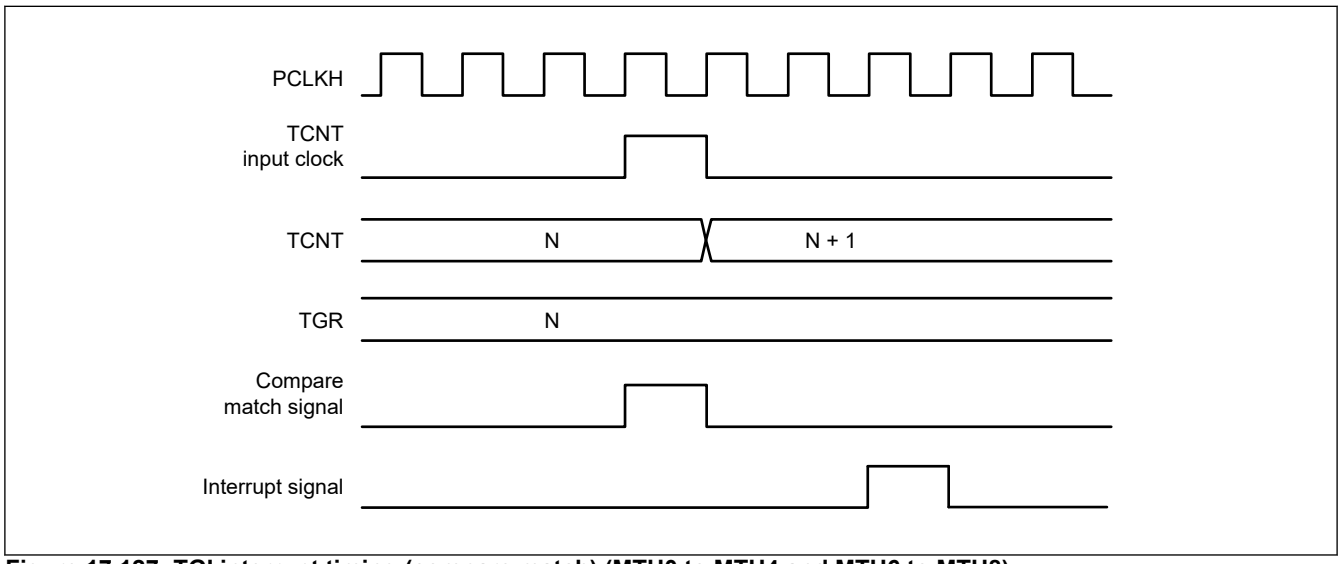


Figure 17.127 TGI interrupt timing (compare match) (MTU0 to MTU4 and MTU6 to MTU8)

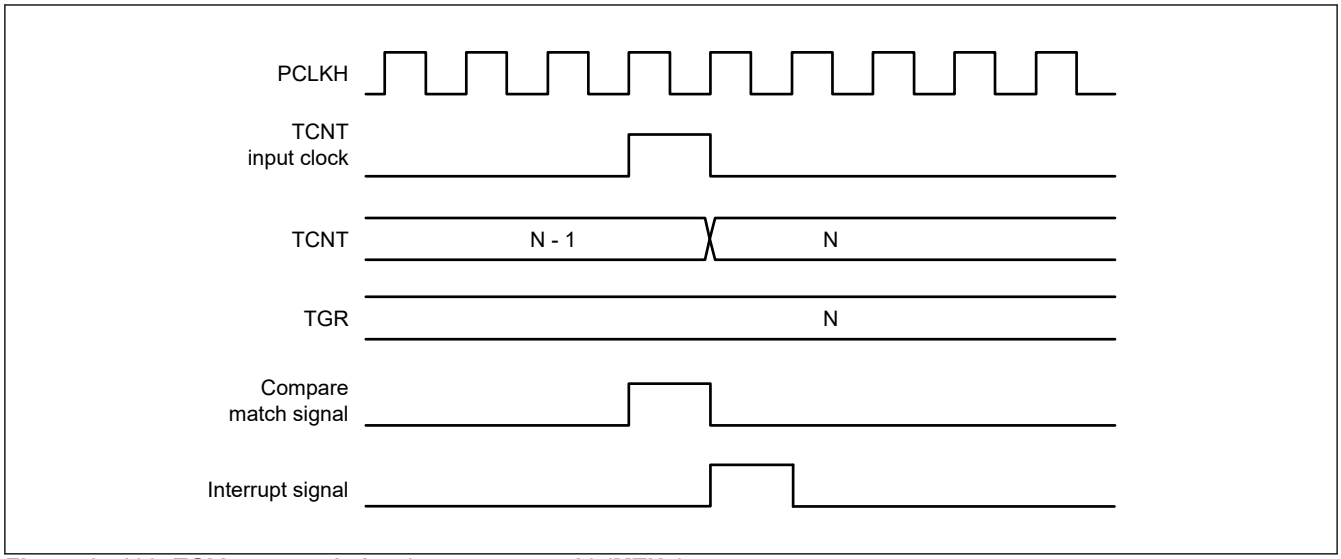


Figure 17.128 TGI interrupt timing (compare match) (MTU5)

17.5.2.2 TGI Interrupt Timing by Input Capture

Figure 17.129 and Figure 17.130 show the TGI interrupt request signal timing when an input capture occurs.

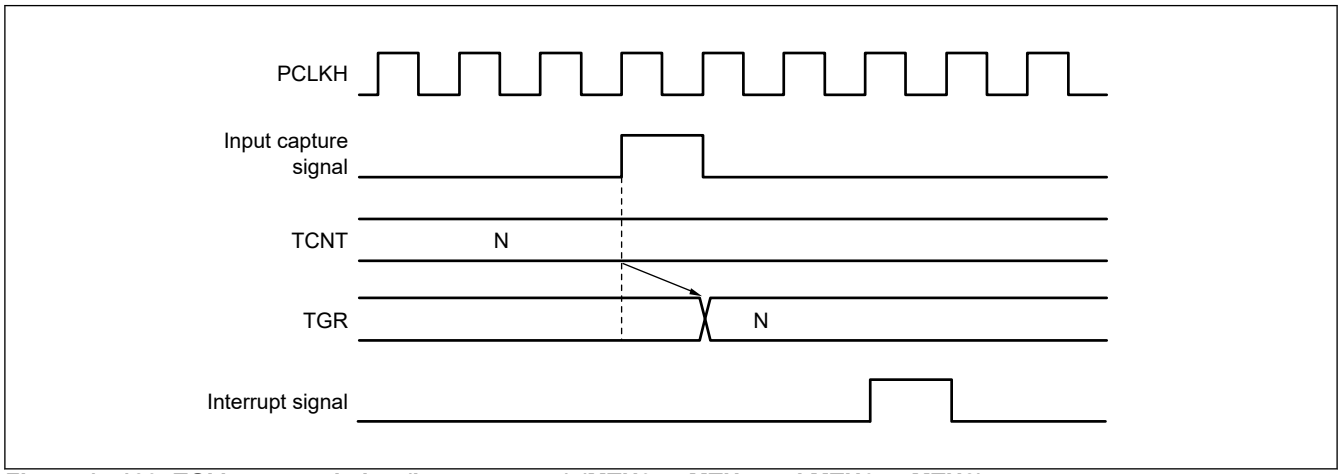


Figure 17.129 TGI interrupt timing (input capture) (MTU0 to MTU4 and MTU6 to MTU8)

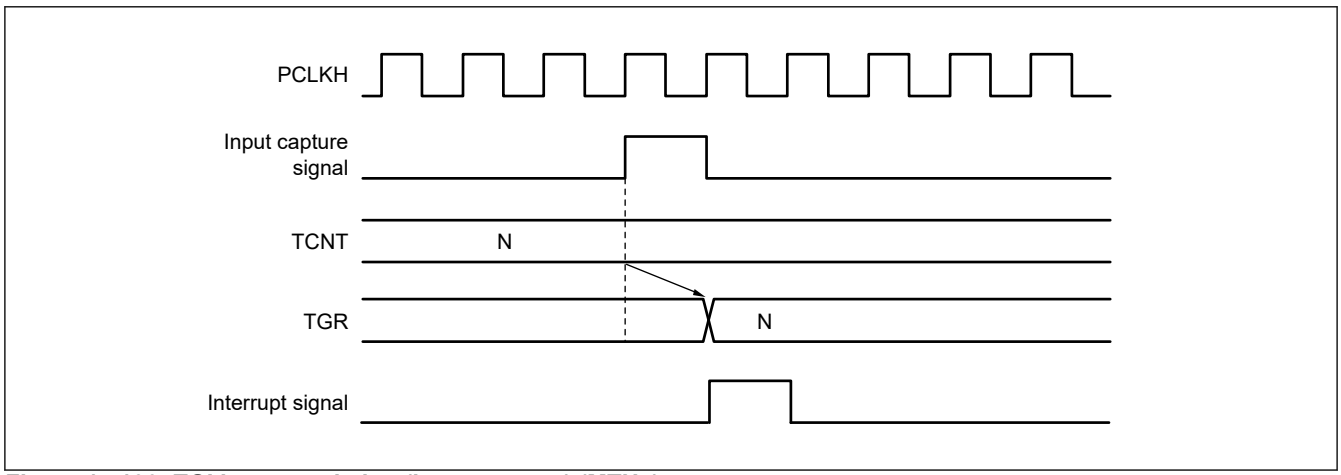


Figure 17.130 TGI interrupt timing (input capture) (MTU5)

17.5.2.3 TCIV and TCIU Interrupt Timing on Overflow/Underflow

Figure 17.131 shows the TCIV interrupt request signal timing when an overflow is generated and Figure 17.132 shows the TCIU interrupt request signal timing when an underflow is generated.

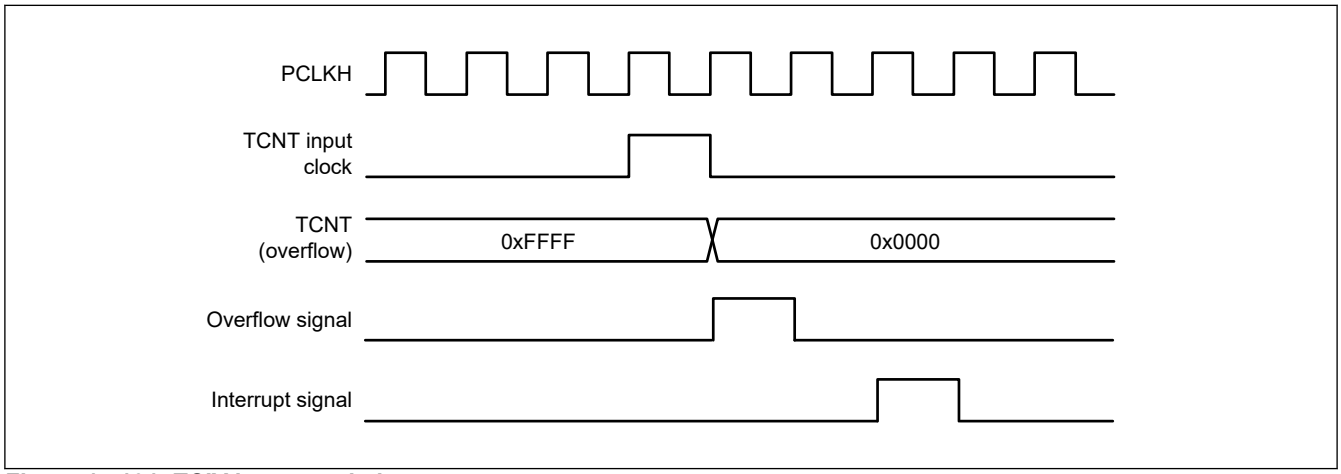


Figure 17.131 TCIV interrupt timing

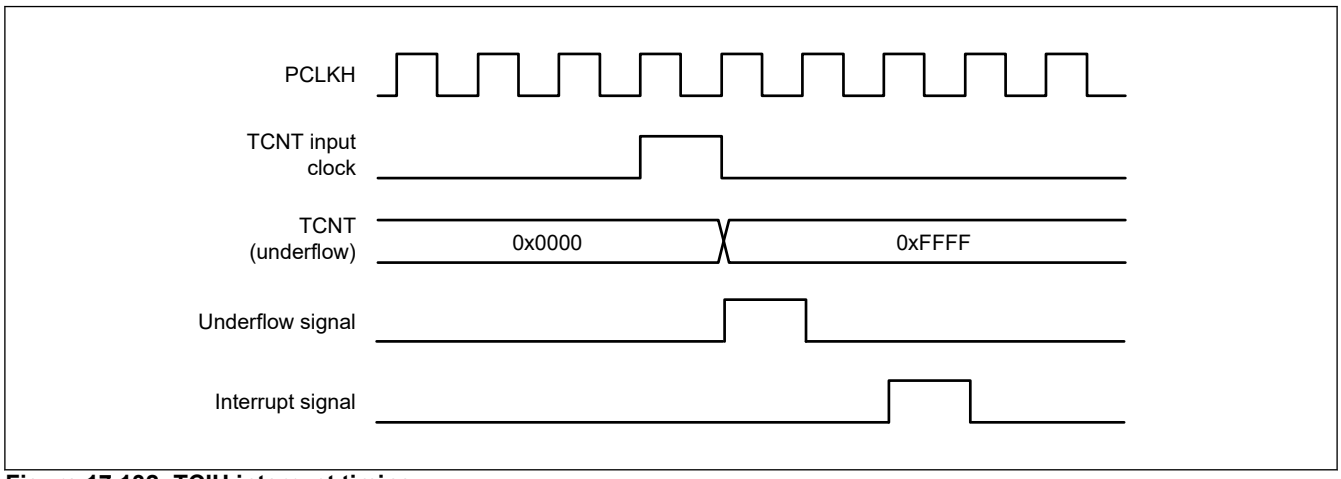


Figure 17.132 TCIU interrupt timing

17.6 Usage Notes

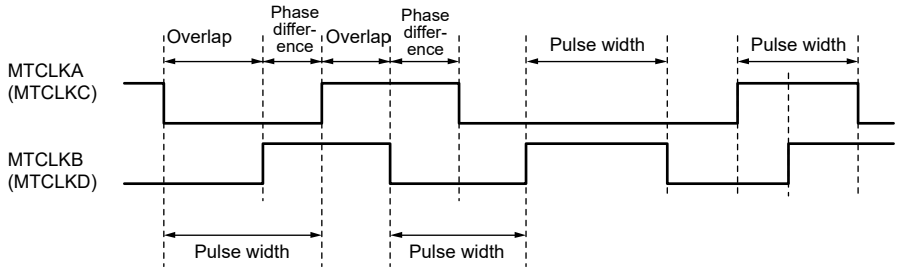
17.6.1 Module Stop Function Setting

MTU operation can be disabled or enabled using the module stop control register. MTU operation is stopped with the initial setting. Register access is enabled by releasing the module clock stop state. For details, see section 8, Low-Power Consumption Function.

17.6.2 Input Clock Restrictions

The input clock pulse width must be at least three PCLKH cycles for single-edge detection, and at least five PCLKH cycles for both-edge detection. The MTU will not operate properly at narrower pulse widths.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least three PCLKH cycles, and the pulse width must be at least five PCLKH cycles. Figure 17.133 shows the input clock conditions in phase counting mode.



Note: Phase difference and overlap: 3 PCLKH cycles or more.
Pulse width: 5 PCLKH cycles or more.

Figure 17.133 Phase difference, overlap, and pulse width in phase counting mode

17.6.3 Note on Period Setting

When counter clearing on compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which TCNT updates the matched count value). Consequently, the actual counter frequency is given by the following formula:

- MTU0 to MTU4 and MTU6 to MTU8
  $f = \frac{CNTCLK}{N+1}$
- MTU5
  $f = \frac{CNTCLK}{N}$

f: Counter frequency

CNTCLK: The counter clock frequency set by TCR.TPSC[2:0] and TCR2.TPSC2[2:0]

N: TGR setting

17.6.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in the TCNT write cycle, TCNT clearing takes precedence and TCNT write operation is not performed.

Figure 17.134 shows the timing in this case.

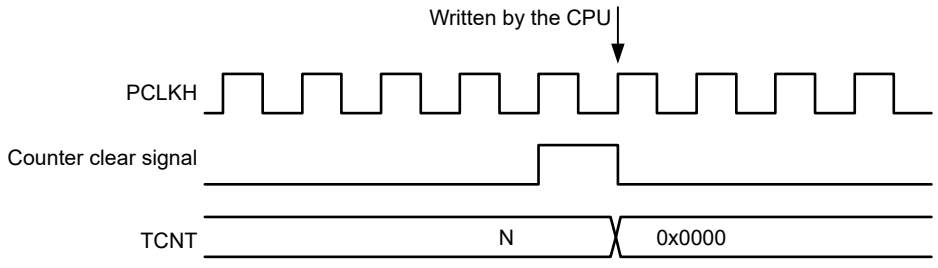


Figure 17.134 Contention between TCNT write and clear operations

17.6.5 Contention between TCNT Write and Increment Operations

If incrementing occurs in the TCNT write cycle, TCNT write operation takes precedence and TCNT is not incremented.

Figure 17.135 shows the timing in this case.

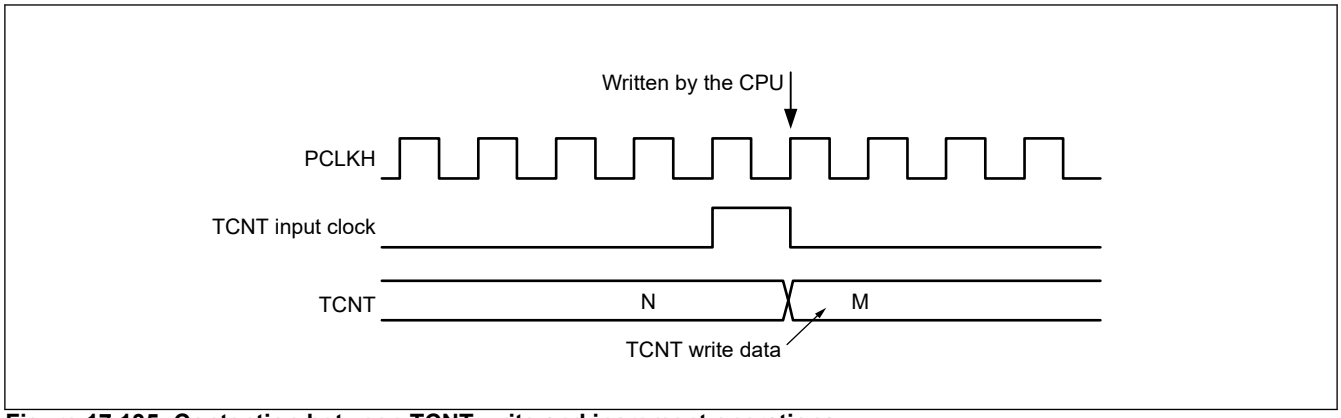


Figure 17.135 Contention between TCNT write and increment operations

17.6.6 Contention between TGR Write Operation and Compare Match

If a compare match occurs in a TGR write cycle, TGR write operation is executed and the compare match signal is also generated.

Figure 17.136 shows the timing in this case.

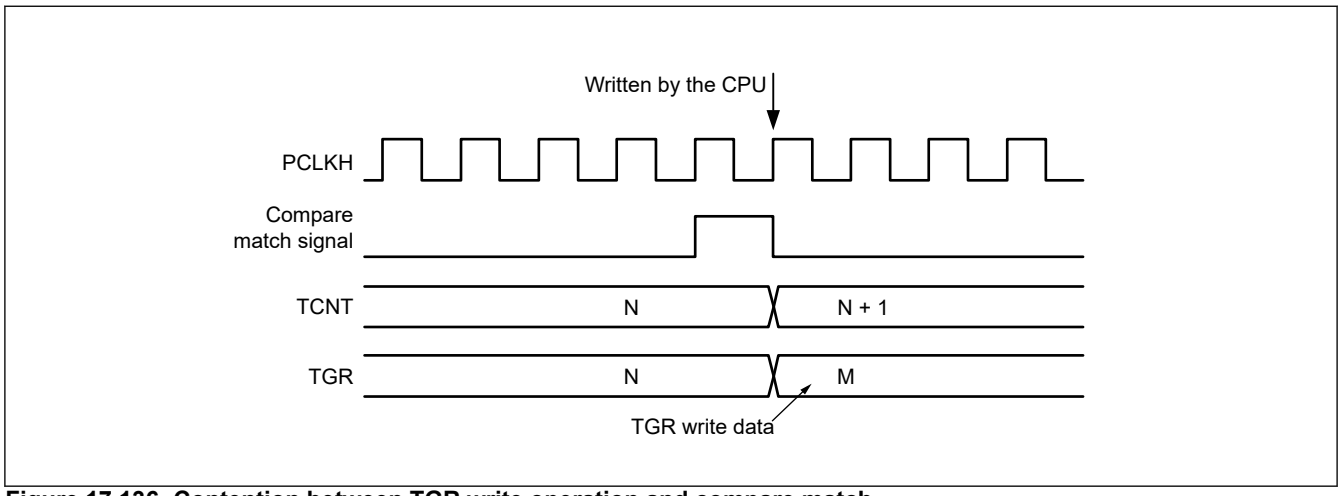


Figure 17.136 Contention between TGR write operation and compare match

17.6.7 Contention between Buffer Register Write Operation and Compare Match

If a compare match occurs in a TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.

Figure 17.137 shows the timing in this case.

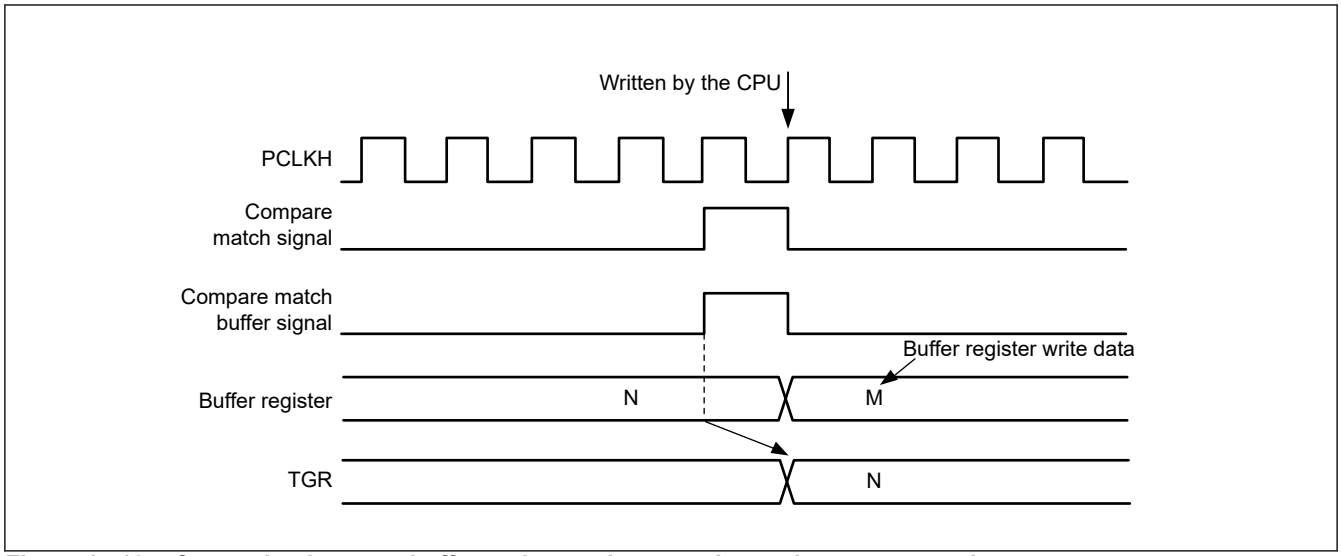


Figure 17.137 Contention between buffer register write operation and compare match

17.6.8 Contention between Buffer Register Write and TCNT Clear Operations

When the buffer transfer timing is set at the TCNT clear timing by the timer buffer operation transfer mode register (TBTM), if TCNT clearing occurs in the TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.

Figure 17.138 shows the timing in this case.

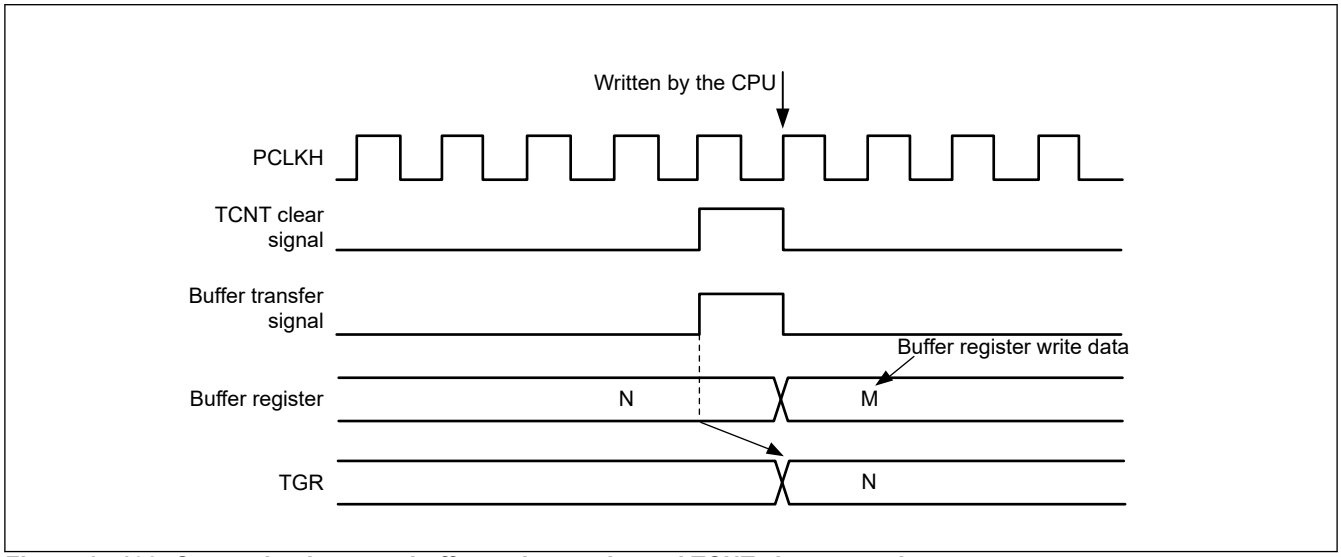


Figure 17.138 Contention between buffer register write and TCNT clear operations

17.6.9 Contention between TGR Read Operation and Input Capture

If an input capture signal is generated in the TGR read cycle, the data before input capture transfer is read.

Figure 17.139 shows the timing in this case.

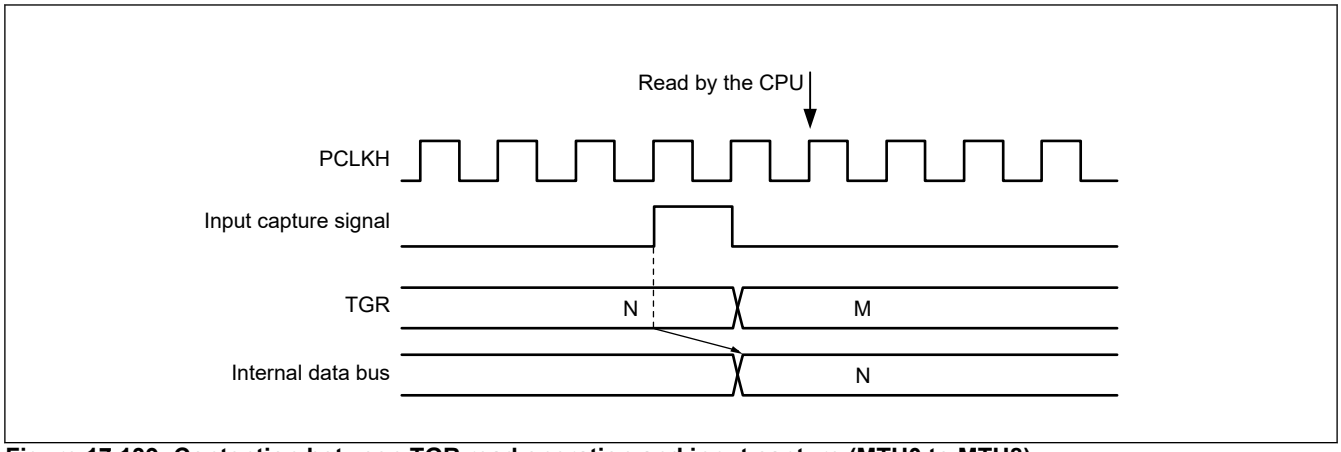


Figure 17.139 Contention between TGR read operation and input capture (MTU0 to MTU8)

17.6.10 Contention between TGR Write Operation and Input Capture

If an input capture signal is generated in a TGR write cycle, the input capture operation takes precedence and the TGR write operation is not performed in MTU0 to MTU4 and MTU6 to MTU8. In MTU5, the TGR write operation is performed and the input capture signal is generated.

Figure 17.140 and Figure 17.141 show the timing in this case.

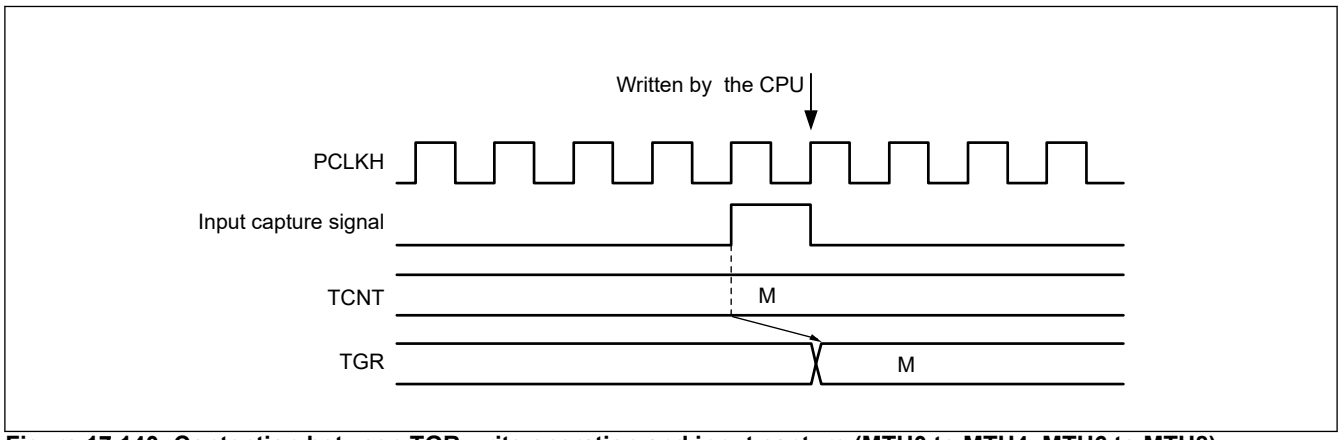


Figure 17.140 Contention between TGR write operation and input capture (MTU0 to MTU4, MTU6 to MTU8)

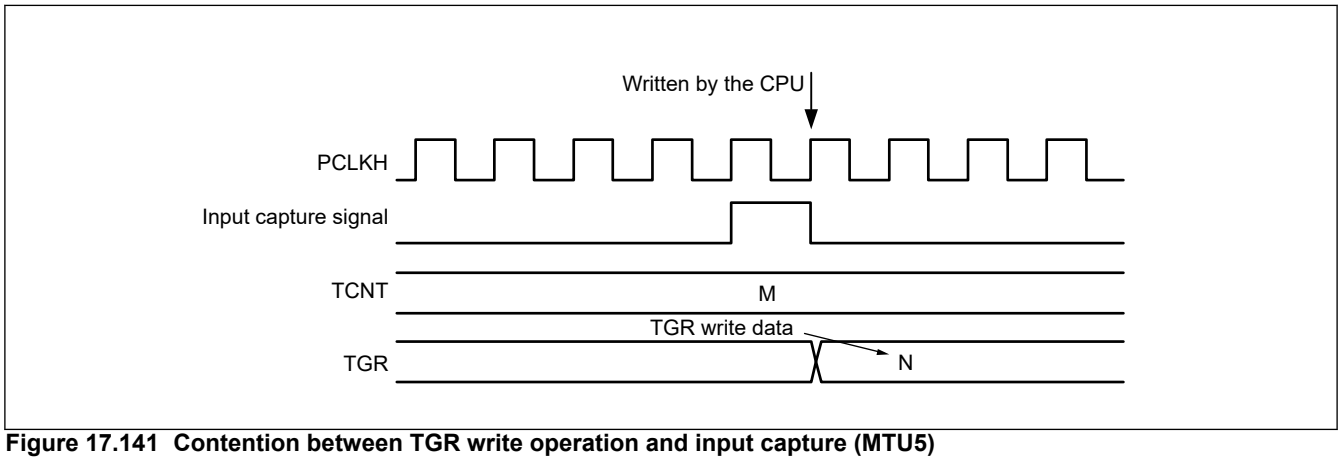


Figure 17.141 Contention between TGR write operation and input capture (MTU5)

17.6.11 Contention between Buffer Register Write Operation and Input Capture

If an input capture signal is generated in a buffer register write cycle, the buffer operation takes precedence and the buffer register write operation is not performed.

Figure 17.142 shows the timing in this case.

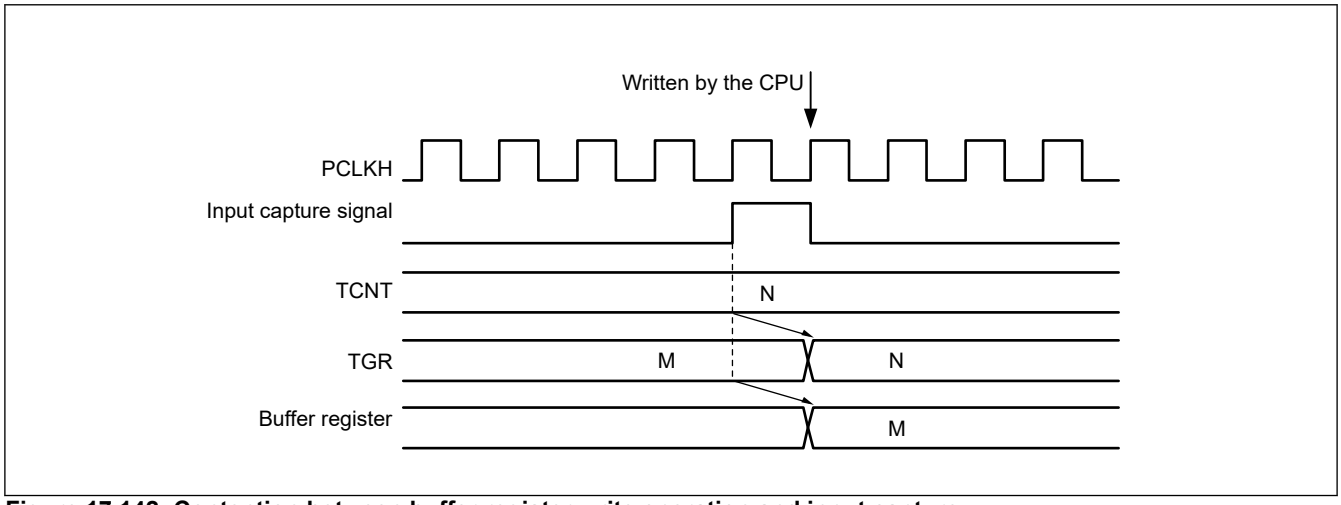


Figure 17.142 Contention between buffer register write operation and input capture

17.6.12 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

With timer counters MTU1.TCNT and MTU2.TCNT in a cascade, when a contention occurs between MTU1.TCNT counting (an MTU2.TCNT overflow/underflow) and the MTU2.TCNT write operation, the MTU2.TCNT write operation is performed and the MTU1.TCNT count signal is disabled. In this case, if MTU1.TGRA works as a compare match register and there is a match between the MTU1.TGRA and MTU1.TCNT values, a compare match signal is issued.

Furthermore, when the MTU1.TCNT count clock is selected as the input capture source of MTU0, MTU0.TGRA to MTU0.TGRD work in input capture mode. In addition, when the MTU0.TGRC compare match/input capture is selected as the input capture source of MTU1.TGRB, MTU1.TGRB works in input capture mode.

Figure 17.143 shows the timing in this case.

When setting the TCNT clearing function in cascaded operation, be sure to synchronize MTU1 and MTU2.

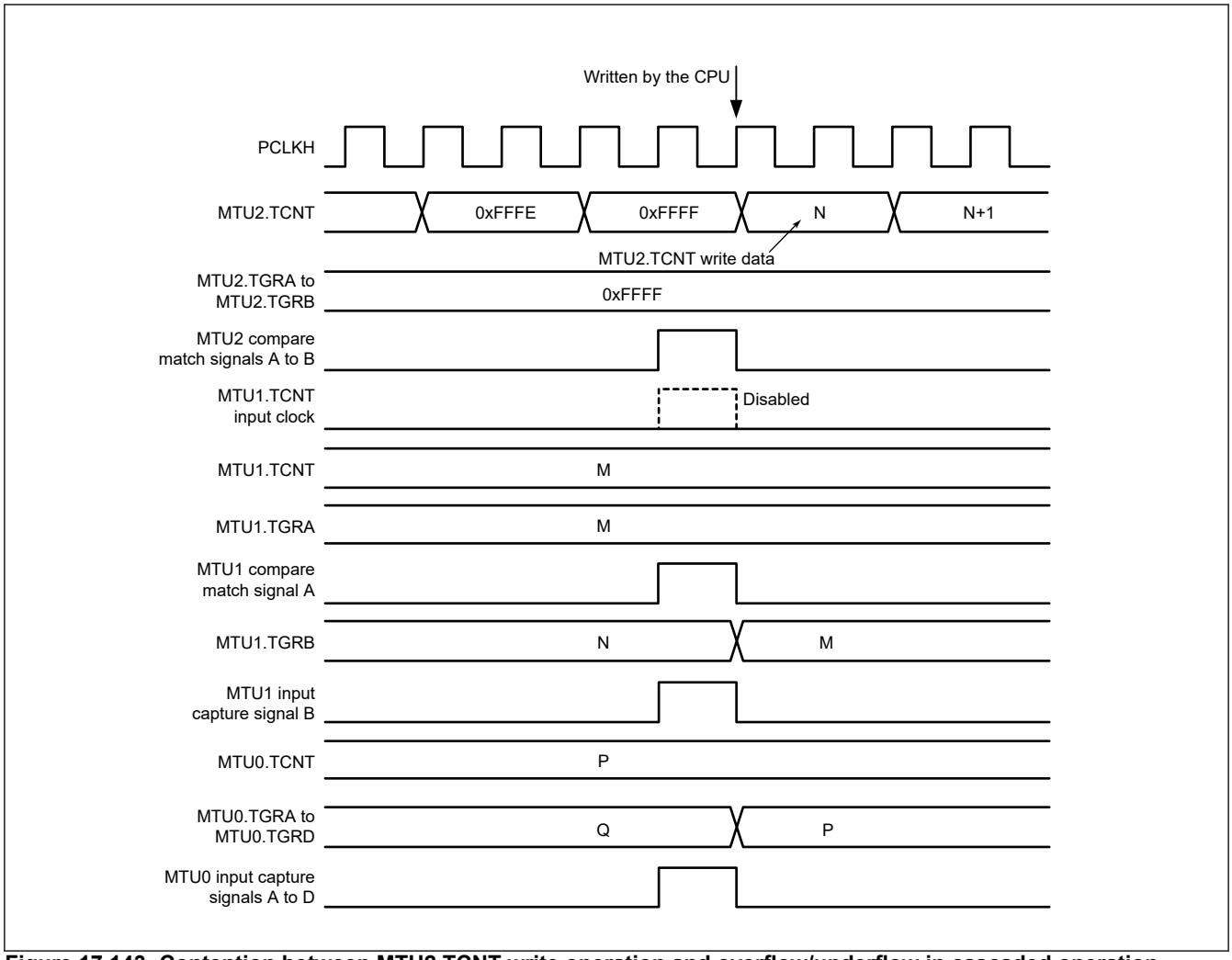


Figure 17.143 Contention between MTU2.TCNT write operation and overflow/underflow in cascaded operation

17.6.13 Counter Value When Count Operation is Stopped in Complementary PWM Mode

When counting operation in MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) is stopped in complementary PWM mode, the MTU3.TCNT (MTU6.TCNT) value is set to the timer dead time data register (TDDRA (TDDRB)) value and MTU4.TCNT (MTU7.TCNT) is set to 0x0000.

When operation is restarted in complementary PWM mode, counting begins automatically from the initial setting state.

Figure 17.144 shows this operation.

When counting begins in another operating mode, be sure to make initial settings in MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT).

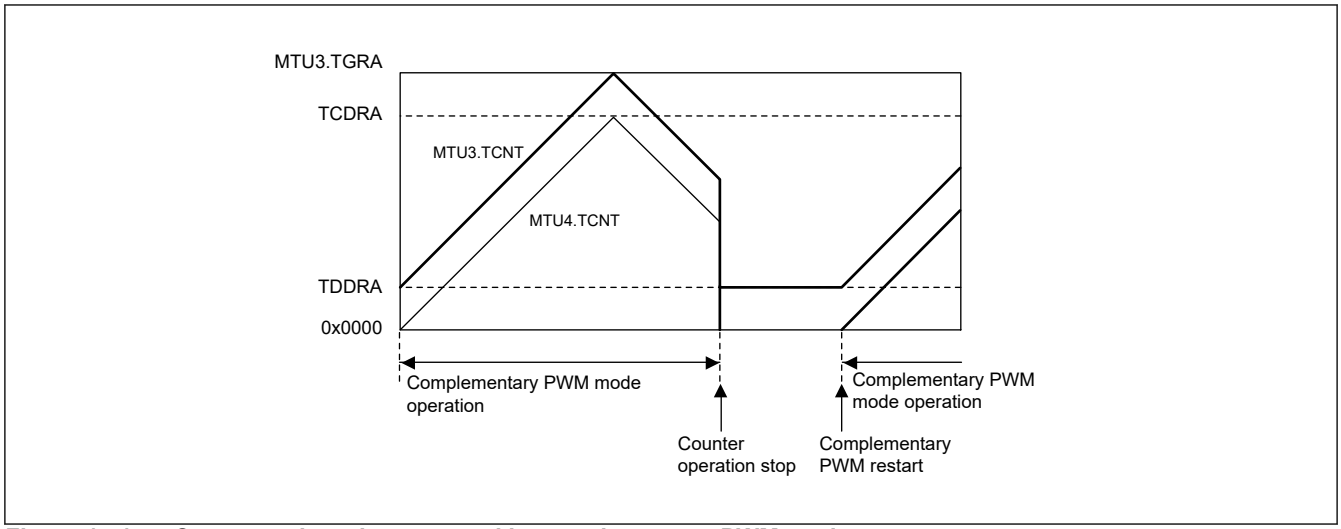


Figure 17.144 Counter value when stopped in complementary PWM mode

17.6.14 Buffer Operation Setting in Complementary PWM Mode

When modifying the PWM period set register (MTU3.TGRA or MTU6.TGRA), timer cycle data register (TCDRA or TCDRB), and duty set registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB (MTU6.TGRB, MTU7.TGRA, and MTU7.TGRB)) in complementary PWM mode, be sure to use buffer operation. In addition, set the BFA and BFB bits in MTU4.TMDR1 (MTU7.TMDR1) to 0. The MTIOC4C (MTIOC7C) pin cannot output waveforms if the BFA bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1. Likewise, the MTIOC4D (MTIOC7D) pin cannot output waveforms if the BFB bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1.

In complementary PWM mode, buffer operation in MTU3 and MTU4 (or MTU6 and MTU7) depends on the settings in bits BFA and BFB of MTU3.TMDR1 (MTU6.TMDR1). When the BFA bit in MTU3.TMDR1 (MTU6.TMDR1) is set to 1, MTU3.TGRC (MTU6.TGRC) functions as a buffer register for MTU3.TGRA (MTU6.TGRA). At the same time, MTU4.TGRC (MTU7.TGRC) functions as a buffer register for MTU4.TGRA (MTU7.TGRA), and TCBRA (TCBRB) functions as a buffer register for TCDRA (TCDRB).

17.6.15 Buffer Operation and Compare Match in Reset-Synchronized PWM Mode

When setting buffer operation in reset-synchronized PWM mode, set the BFA and BFB bits in MTU4.TMDR1 (MTU7.TMDR1) to 0. The MTIOC4C (MTIOC7C) pin cannot output waveforms if the BFA bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1. Likewise, the MTIOC4D (MTIOC7D) pin cannot output waveforms if the BFB bit in MTU4.TMDR1 (MTU7.TMDR1) is set to 1.

In reset-synchronized PWM mode, buffer operation in MTU3 and MTU4 (or MTU6 and MTU7) depends on the settings in the BFA and BFB bits of MTU3.TMDR1 (MTU6.TMDR1). For example, if the BFA bit in MTU3.TMDR1 (MTU6.TMDR1) is set to 1, MTU3.TGRC (MTU6.TGRC) functions as a buffer register for MTU3.TGRA (MTU6.TGRA). At the same time, MTU4.TGRC (MTU7.TGRC) functions as a buffer register for MTU4.TGRA (MTU7.TGRA).

While the MTU3.TGRC (MTU6.TGRC) and MTU3.TGRD (MTU6.TGRD) are operating as buffer registers, a TGImn interrupt (m = C, D; n = 3, 4 or 6, 7) is not generated.

Figure 17.145 shows an example of MTU3.TGR (MTU6.TGR), MTU4.TGR (MTU7.TGR), MTIOC3 (MTIOC6), and MTIOC4 (MTIOC7) operation with the BFA and BFB bits in MTU3.TMDR1 (MTU6.TMDR1) set to 1 and the BFA and BFB bits in MTU4.TMDR1 (MTU7.TMDR1) set to 0.

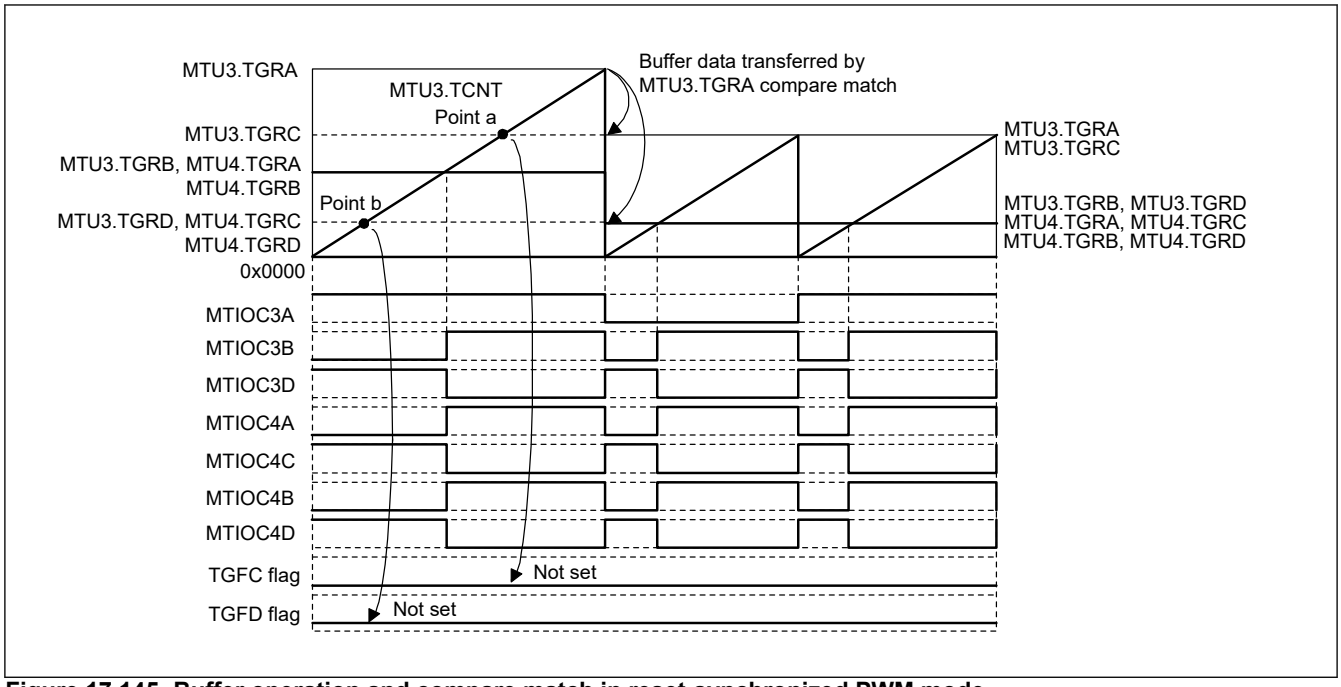


Figure 17.145 Buffer operation and compare match in reset-synchronized PWM mode

17.6.16 Overflow in Reset-Synchronized PWM Mode

After reset-synchronized PWM mode is selected, MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) start counting when the CST3 (CST6) bit of TSTRA (TSTRB) is set to 1. In this state, the MTU4.TCNT (MTU7.TCNT) count clock source and count edge are determined by the MTU3.TCR (MTU6.TCR) setting.

In reset-synchronized PWM mode, with period register MTU3.TGRA (MTU6.TGRA) set to 0xFFFF and the MTU3.TGRA (MTU6.TGRA) compare match selected as the counter clearing source, MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) count up to 0xFFFF, then a compare match occurs with MTU3.TGRA (MTU6.TGRA), and MTU3.TCNT and MTU4.TCNT (MTU6.TCNT and MTU7.TCNT) are both cleared. In this case, a TCIVn interrupt (n = 3, 4 or 6, 7) is not generated.

Figure 17.146 shows an example of operation in reset-synchronized PWM mode with period register MTU3.TGRA (MTU6.TGRA) set to 0xFFFF and the MTU3.TGRA (MTU6.TGRA) compare match specified for the counter clearing source.

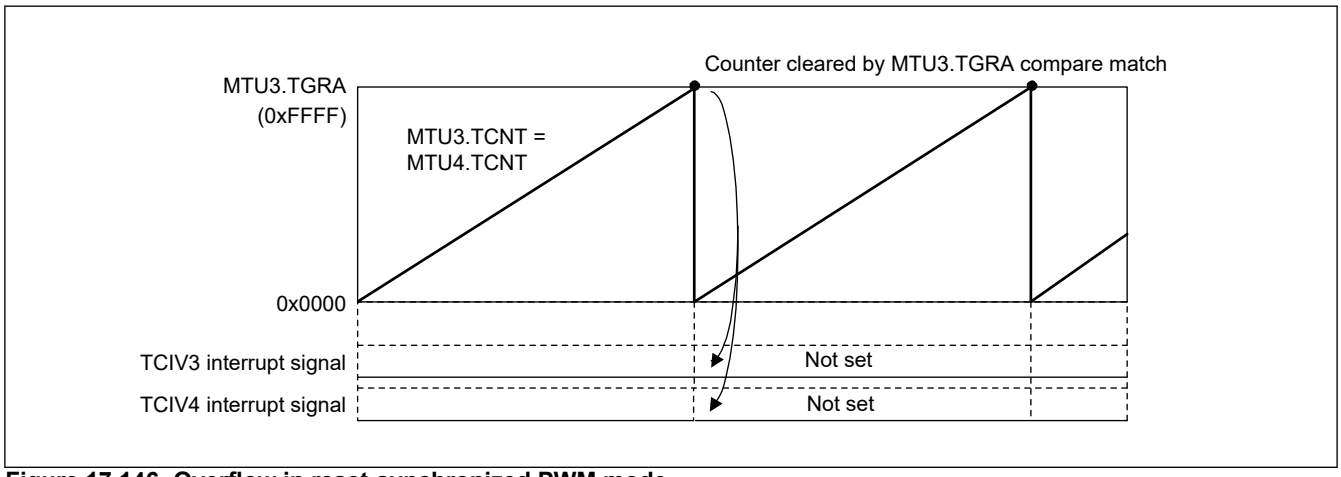


Figure 17.146 Overflow in reset-synchronized PWM mode

17.6.17 Contention between Overflow/Underflow and Counter Clearing

If an overflow/underflow and counter clearing occur simultaneously, neither a TCIVn interrupt (n = 0 to 4, 6 to 8) nor a TCIUn interrupt (n = 1, 2) is generated and TCNT clearing takes precedence.

Figure 17.147 shows the operation timing when a TGR compare match is specified as the clearing source and TGR is set to 0xFFFF.

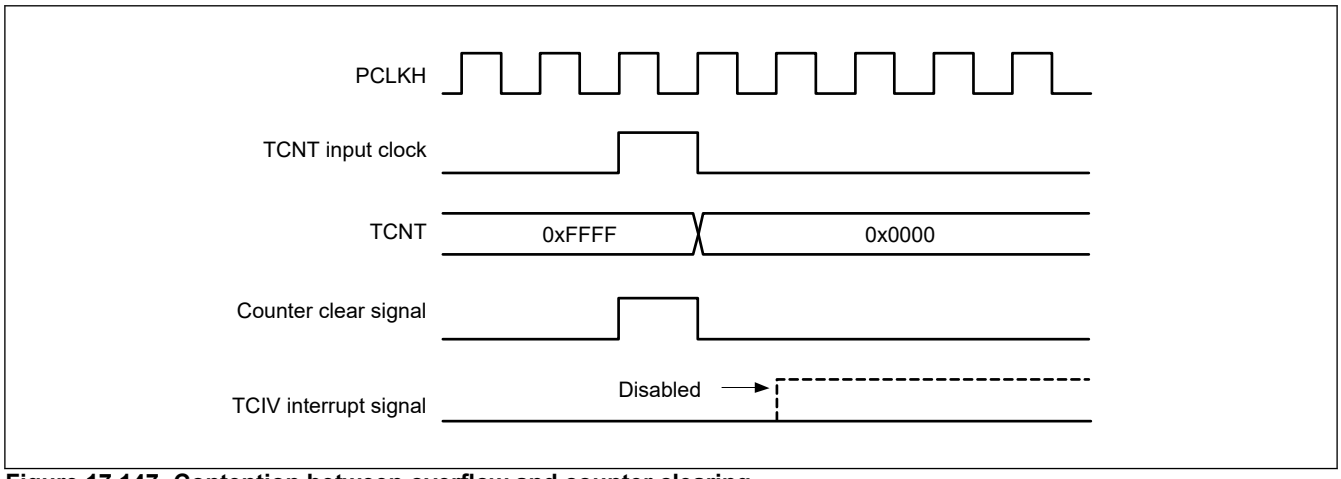


Figure 17.147 Contention between overflow and counter clearing

17.6.18 Contention between TCNT Write Operation and Overflow/Underflow

If TCNT counts up or down in a TCNT write cycle and an overflow or an underflow occurs, the TCNT write operation takes precedence. A TCIVn interrupt (n = 0 to 4, 6 to 8) nor a TCIUn interrupt (n = 1, 2) is not generated.

Figure 17.148 shows the operation timing when there is contention between TCNT write operation and overflow.

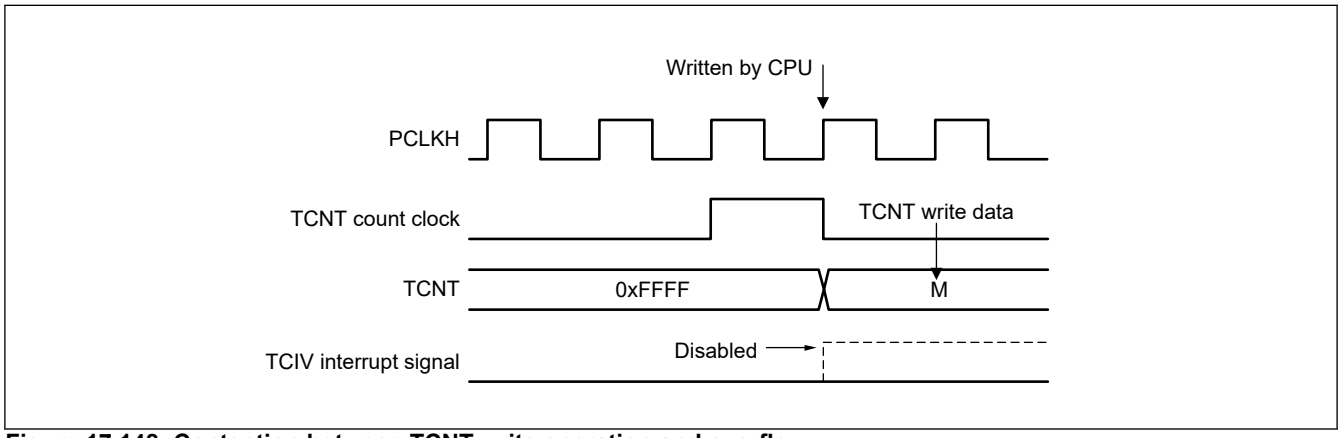


Figure 17.148 Contention between TCNT write operation and overflow

17.6.19 Note on Transition from Normal Mode or PWM Mode 1 to Reset-Synchronized PWM Mode

When making a transition from normal operation or PWM mode 1 to reset-synchronized PWM mode in MTU3 and MTU4 (or MTU6 and MTU7), if the counter is stopped while the output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, and MTIOC7D) are held at a high level and then operation is started after a transition to reset-synchronized PWM mode, the initial pin output will not be correct.

When making a transition from normal mode to reset-synchronized PWM mode, write 0x11 to MTU3.TIORH, MTU3.TIORL, MTU4.TIORH, and MTU4.TIORL (MTU6.TIORH, MTU6.TIORL, MTU7.TIORH, and MTU7.TIORL) to initialize the output pin state to a low level, then set the registers to the initial value (0x00) before making the mode transition.

When making a transition from PWM mode 1 to reset-synchronized PWM mode, switch to normal mode, initialize the output pin state to a low level, and then set the registers to the initial value (0x00) before making the transition to reset-synchronized PWM mode.

17.6.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode

When MTU3 and MTU4 (or MTU6 and MTU7) are in complementary PWM mode or reset-synchronized PWM mode, the PWM waveform output level is determined by the OLSP and OLSN bits in the timer output control register (TOCR1A or TOCR1B). In complementary PWM mode or reset-synchronized PWM mode, TIOR should be set to 0x00.

The output level in negative phase when the TDERA.TDER (TDERB.TDER) bit is set to 0 in complementary PWM mode (the dead time is not generated) does not depend on the setting of the TOCR1A.OLSN (TOCR1B.OLSN) bit. It is equivalent to the inverted level of positive phase output based on the setting of the TOCR1A.OLSP (TOCR1B.OLSP) bit.

17.6.21 Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection

When timer counters 1 and 2 (MTU1.TCNT and MTU2.TCNT) operate as a 32-bit counter in cascade connection, the cascaded counter value cannot be captured successfully in some cases even if input-capture input is simultaneously done to MTIOC1A and MTIOC2A or to MTIOC1B and MTIOC2B. This is because the input timing of MTIOC1A and MTIOC2A or to MTIOC1B and MTIOC2B may not be the same when external input-capture signals input into MTU1.TCNT and MTU2.TCNT are taken in synchronization with the internal clock.

For example, MTU1.TCNT (the counter for upper 16 bits) does not capture the count-up value by an overflow from MTU2.TCNT (the counter for lower 16 bits) but captures the count value before the up-counting. In this case, the values of MTU1.TCNT = 0xFFF1 and MTU2.TCNT = 0x0000 should be transferred to MTU1.TGRA and MTU2.TGRA or to MTU1.TGRB and MTU2.TGRB, but the values of MTU1.TCNT = 0xFFF0 and MTU2.TCNT = 0x0000 are erroneously transferred.

The MTU has a new function that allows simultaneous capture of MTU1.TCNT and MTU2.TCNT with a single input capture input as the trigger. This function can be used to read the 32-bit counter such that MTU1.TCNT and MTU2.TCNT are captured at the same time. For details, see section 17.2.12. TICCR : Timer Input Capture Control Register.

17.6.22 Interrupt Skipping Function 2

When interrupt skipping function 2 is in use and the difference between the values in MTU4.TADCORA and MTU4.TADCORB is small, correct counting of the number skipped may not be possible, in which case requests for A/D conversion will not be generated with the expected timing. The conditions listed below thus apply to these settings.

For MTU6 and MTU7, the same conditions apply to the settings of MTU7.TADCORA and MTU7.TADCORB.

1. When the number skipped is zero for skipping function 2
   - The difference between the values in MTU4.TADCORA and MTU4.TADCORB must be at least four.
   - The interval of comparison for MTU4.TADCORA must be at least four cycles of PCLKH (the updated value of MTU4.TADCORA is set to the previous value plus or minus at least four).
   - The interval of comparison for MTU4.TADCORB must be at least four cycles of PCLKH (the updated value of MTU4.TADCORB is set to the previous value plus or minus at least four).
2. When the number skipped is one or more for skipping function 2
   - The difference between the values in MTU4.TADCORA and MTU4.TADCORB must be at least two.
   - The interval of comparison for MTU4.TADCORB must be at least two cycles of PCLKH (the updated value of MTU4.TADCORB is set to the previous value plus or minus at least two).

17.6.23 Notes When Complementary PWM Mode Output Protection Function is Not Used

The complementary PWM mode output protection function is initially enabled. If it is not used, write 0x00 in the POE3.POECR2 register.

17.6.24 Notes Regarding Timer Counter (MTU5.TCNT) and Timer General Register (MTU5.TGR)

Do not set an MTU5.TGRj (j = U, V, W) bit to the value of the corresponding MTU5.TCNTj (j = U, V, W) plus one while counting by the MTU5.TCNTj (j = U, V, W) register is stopped. If an MTU5.TGRj (j = U, V, W) bit is set to the value of the

corresponding MTU5.TCNTj (j = U, V, W) plus one while counting by the MTU5.TCNTj (j = U, V, W) is stopped, a compare-match will be generated even though counting is stopped.

In this case, if the compare match enable bit (MTU5.TIER.TGIE5j (j = U, V, W) bit is set to 1 (enabling interrupts), a compare-match interrupt will also be generated. If the value of the timer compare match clear register is 1 (enabled), the timer is automatically cleared to 0x0000 when the compare-match is generated, regardless of whether interrupts from the MTU5.TCNTj (j = U, V, W) are enabled or disabled.

17.6.25 Notes to Prevent Malfunctions in Synchronous Clearing for Complementary PWM Mode

If control of the output waveform is enabled (TWCRA.WRE bit = 1 or TWCRB.WRE bit = 1) at the time of synchronous counter clearing in complementary PWM mode, satisfaction of either condition 1 or 2 below has the following effects.

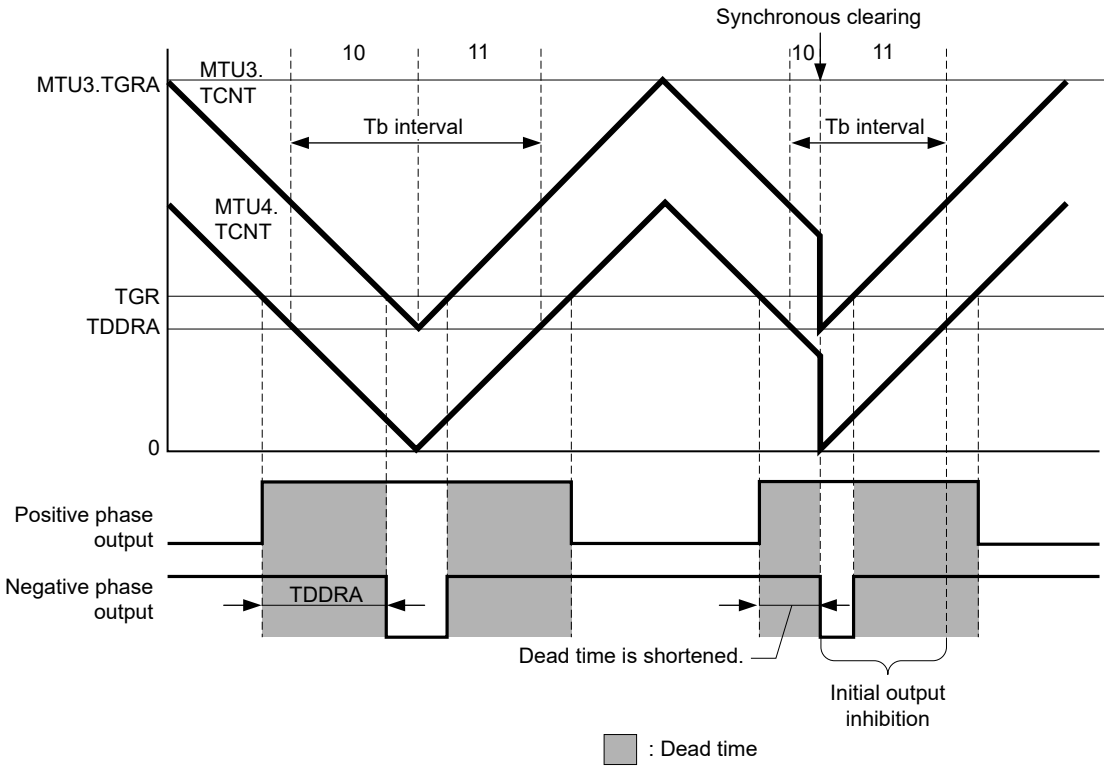
- Dead time on the PWM output pins is shortened (or disappears).
- The active level is output on the negative phase PWM output pins beyond the period for active-level output.

Condition 1: In portion (10) of the initial output inhibition period in Figure 17.149, synchronous clearing occurs within the dead-time period for PWM output.

Condition 2: In portions (10) and (11) of the initial output inhibition period in Figure 17.150, synchronous clearing occurs when any condition from among MTU3.TGRB (MTU6.TGRB) ≤ TDDRA, MTU4.TGRA (MTU7.TGRA) ≤ TDDRA, or MTU4.TGRB (MTU7.TGRB) ≤ TDDRA is satisfied.

The following method avoids the above phenomena.

Ensure that synchronous clearing proceeds with the value of each comparison register (MTU3.TGRB (MTU6.TGRB), MTU4.TGRA (MTU7.TGRA), and MTU4.TGRB (MTU7.TGRB)) set to at least double the value of the TDDRA register (TDDRB register).



Note: PWM output is active-low.

Figure 17.149 Example of synchronous clearing (when condition 1 applies)

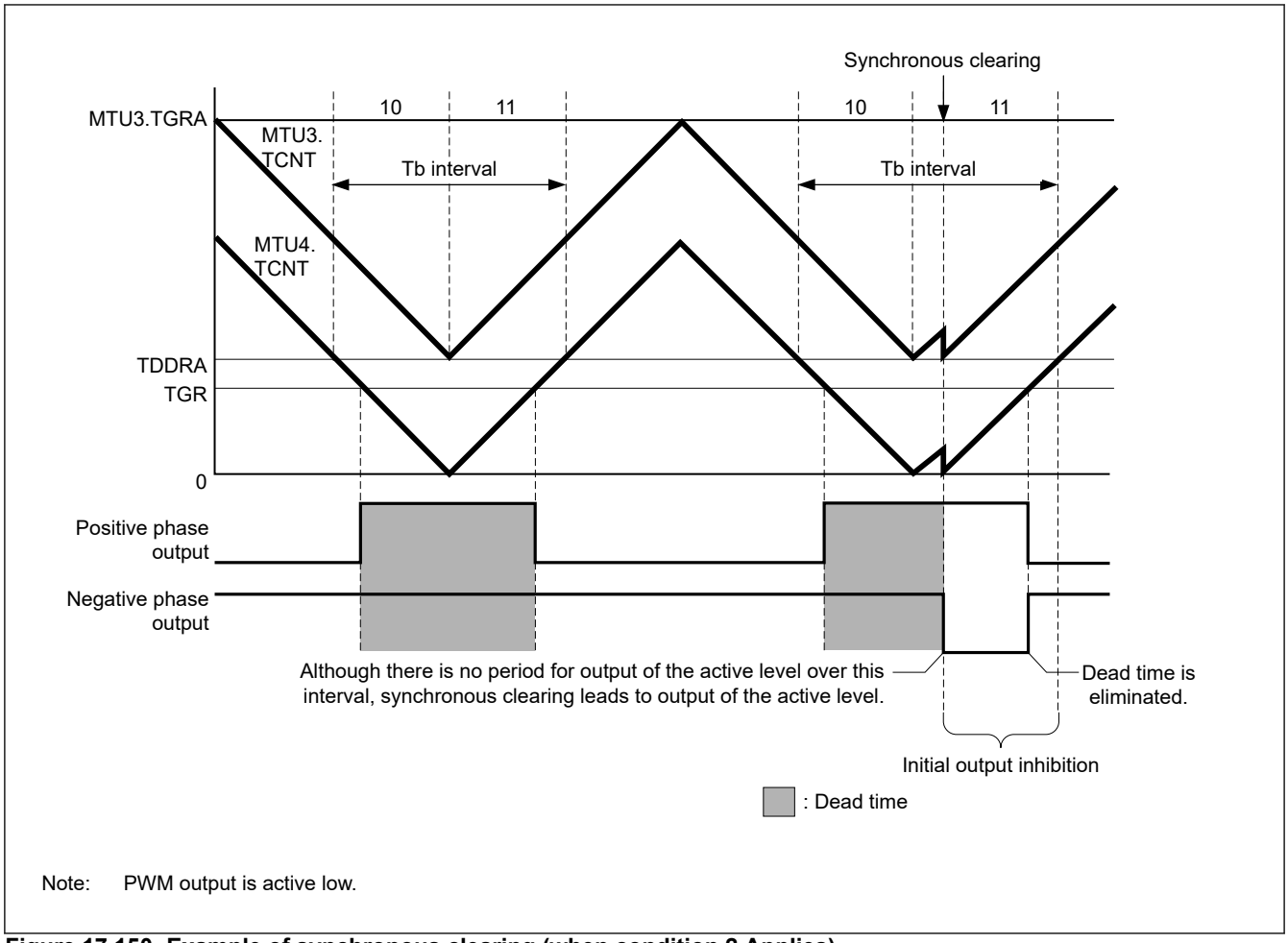


Figure 17.150 Example of synchronous clearing (when condition 2 Applies)

17.6.26 Notes on Timer Mode Register Setting for ELC Event Input

When MTU is used in ELC operation, set the timer mode register (TMDR) of the relevant channel to its initial value (0x00).

17.6.27 Continuous Output of Interrupt Signal in Response to a Compare Match

When the TGR register is set to 0x0000, the PCLKH/1 clock is set as the counter clock, and compare match is set as the trigger for clearing of the counter clock, the value of the TCNT counter remains 0x0000, and the interrupt signal is output continuously (i.e. its level is flat) rather than output over a single cycle. Consequently, interrupts are not detected in response to second and subsequent compare matches.

Figure 17.151 shows the timing for continuous output of the interrupt signal in response to a compare match.

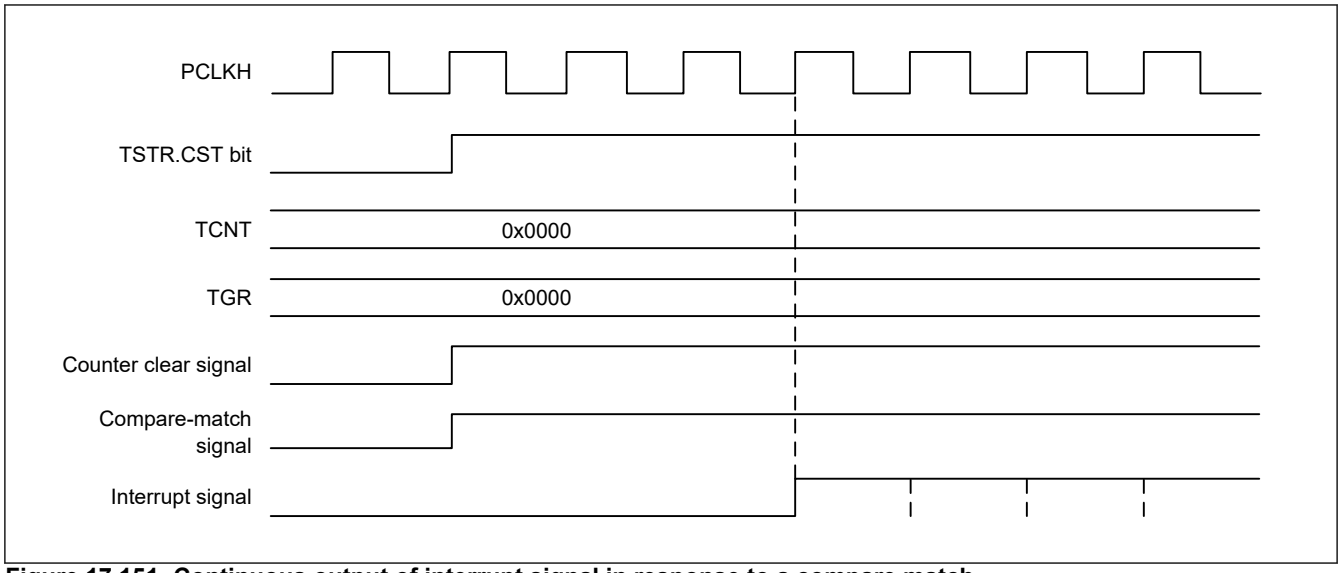


Figure 17.151 Continuous output of interrupt signal in response to a compare match

17.7 MTU Output Pin Initialization

17.7.1 Operating Modes

The MTU has the following six operating modes. Waveforms can be output in any of these modes.

- Normal mode (MTU0 to MTU4 and MTU6 to MTU8)
- PWM mode 1 (MTU0 to MTU4, MTU6, and MTU7)
- PWM mode 2 (MTU0 to MTU2)
- Phase counting modes 1 to 5 (MTU1 and MTU2)
- Complementary PWM mode (MTU3, MTU4, MTU6, and MTU7)
- Reset-synchronized PWM mode (MTU3, MTU4, MTU6, and MTU7)

This section describes how to initialize the MTU output pins in each of these modes.

17.7.2 Operation in Case of Re-Setting Due to Error during Operation

If an error occurs during MTU operation, MTU output should be cut off by the system. The output can be cut off by allowing non-active level output from the pins by setting the pins as general output ports using the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports. MTU output can be disabled through TIOR settings. Complementary PWM output (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D, MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC7C, and MTIOC7D) should be specified through TOERA and TOERB settings. For PWM output pins, output can also be cut by hardware, using port output enable 3 (POE3). The pin initialization procedures for re-setting due to an error during operation and the procedures for restarting in a different mode after re-setting are described below.

The MTU has six operating modes, as stated above. There are thus 36 mode transition combinations, but some transitions are not available with certain channel and mode combinations. Available mode transition combinations are shown in Table 17.74.

Table 17.74 Mode transition combinations (1 of 2)

	Normal	PWM1	PWM2	PCM	CPWM	RPWM
Normal	(1)	(2)	(3)	(4)	(5)	(6)
PWM1	(7)	(8)	(9)	(10)	(11)	(12)
PWM2	(13)	(14)	(15)	(16)	Not available	Not available
PCM	(17)	(18)	(19)	(20)	Not available	Not available

Table 17.74 Mode transition combinations (2 of 2)

	Normal	PWM1	PWM2	PCM	CPWM	RPWM
CPWM	(21)	(22)	Not available	Not available	(23) (24)	(25)
RPWM	(26)	(27)	Not available	Not available	(28)	(29)

Note: Normal: Normal mode
PWM1: PWM mode 1
PWM2: PWM mode 2
PCM: Phase counting modes 1 to 5
CPWM: Complementary PWM mode
RPWM: Reset-synchronized PWM mode

17.7.3 Overview of Initialization Procedures and Mode Transitions in Case of Error During Operation

- When making a transition to a mode (Normal, PWM1, PWM2, or PCM) in which the pin output level is selected by the timer I/O control register (TIOR) setting, initialize the pins by means of TIOR setting.
- In PWM mode 1, waveforms are not output to the MTIOCnB and MTIOCnD (n = 3, 4, 6, 7) pins. If no other module outputs signals through these pins, they enter high-impedance state. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.
- In PWM mode 2, waveforms are not output to the cycle register pins. If no other module outputs signals through these pins, they enter high-impedance state. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.
- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, waveforms are not output to the corresponding pins. If no other module outputs signals through these pins, they enter high-impedance state. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, waveforms are not output to the corresponding pins. If no other module outputs signals through these pins, they enter high-impedance state. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.
- When making a transition to a mode (CPWM or RPWM) in which the pin output level is selected by the Timer Output Control register (TOCR1A, TOCR2A, TOCR1B, or TOCR2B) setting, temporarily disable output in MTU3 and MTU4 (or MTU6 and MTU7) with the Timer Output Master Enable register (TOERA or TOERB). At this time, if no other module outputs signals through these pins, they enter high-impedance state. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports. Switch to normal mode, perform initialization with TIOR, restore TIOR to its initial value, then operate the MTU in accordance with the mode setting procedure (TOCR1A setting, TOCR2A setting, TMDR1 setting, and TOERA setting (TOCR1B setting, TOCR2B setting, TMDR1 setting, and TOERB setting)).

Note: Channel number is substituted for "n" indicated in this section, unless otherwise stated.

Pin initialization procedures are described below for the numbered combinations in Table 17.74. The active level is assumed to be low.

17.7.3.1 Operation When Error Occurs in Normal Mode and Operation is Restarted in Normal Mode

Figure 17.152 shows a case in which an error occurs in normal mode and operation is restarted in normal mode after resetting.

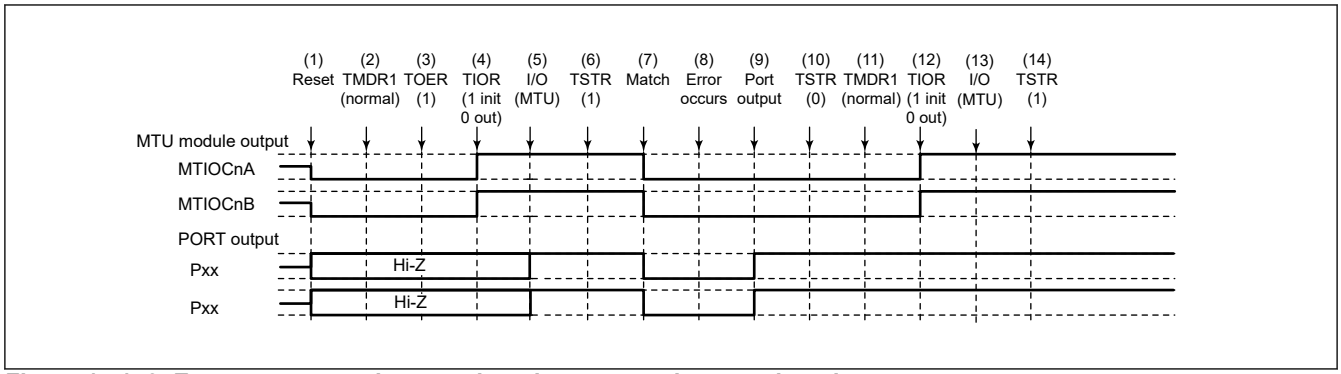


Figure 17.152 Error occurrence in normal mode, recovery in normal mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.

(2) After a reset, the TMDR1 setting is for normal mode.

(3) For MTU3 and MTU4 (MTU6 and MTU7), enable output with the TOERA (TOERB) register before initializing the pins with the TIOR register.

(4) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)

(5) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(6) Start count operation by setting the TSTRA (TSTRB) register.

(7) Output goes low on compare match occurrence.

(8) An error occurs.

(9) Allow non-active level output by setting the pins as general output ports using the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.

(10) Stop count operation by setting the TSTRA (TSTRB) register.

(11) This step is not necessary when restarting in normal mode.

(12) Initialize the pins with the TIOR register.

(13) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA (TSTRB) register.

17.7.3.2 Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM Mode 1

Figure 17.153 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

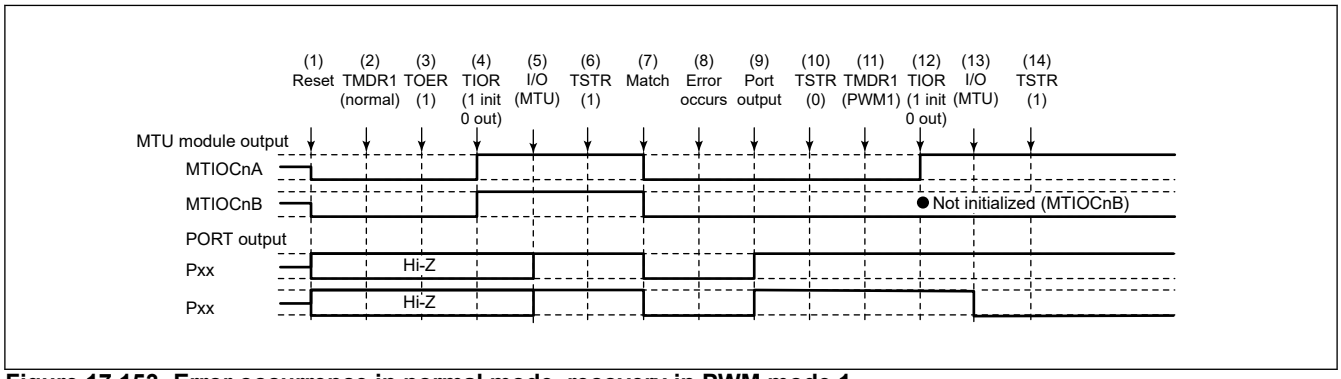


Figure 17.153 Error occurrence in normal mode, recovery in PWM mode 1

(1) to (10) are the same as in Figure 17.152.

(11) Set PWM mode 1.

(12) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.)

(13) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA (TSTRB) register.

17.7.3.3 Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM mode 2

Figure 17.154 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

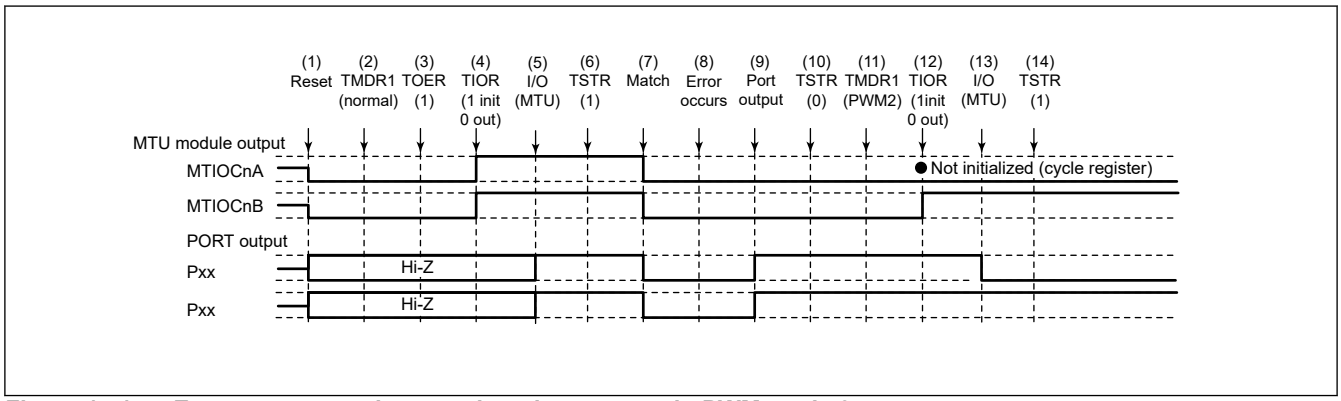


Figure 17.154 Error occurrence in normal mode, recovery in PWM mode 2

(1) to (10) are the same as in Figure 17.152.

(11) Set PWM mode 2.

(12) Initialize the pins with the TIOR register. (In PWM mode 2, a waveform is not output to the pin that corresponds to the TGR register used as a period register. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.)

(13) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA register.

Note: PWM mode 2 can only be selected for MTU0 to MTU2, and therefore the TOERA register setting is not necessary.

17.7.3.4 Operation When Error Occurs in Normal Mode and Operation is Restarted in Phase Counting Mode

Figure 17.155 shows a case in which an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

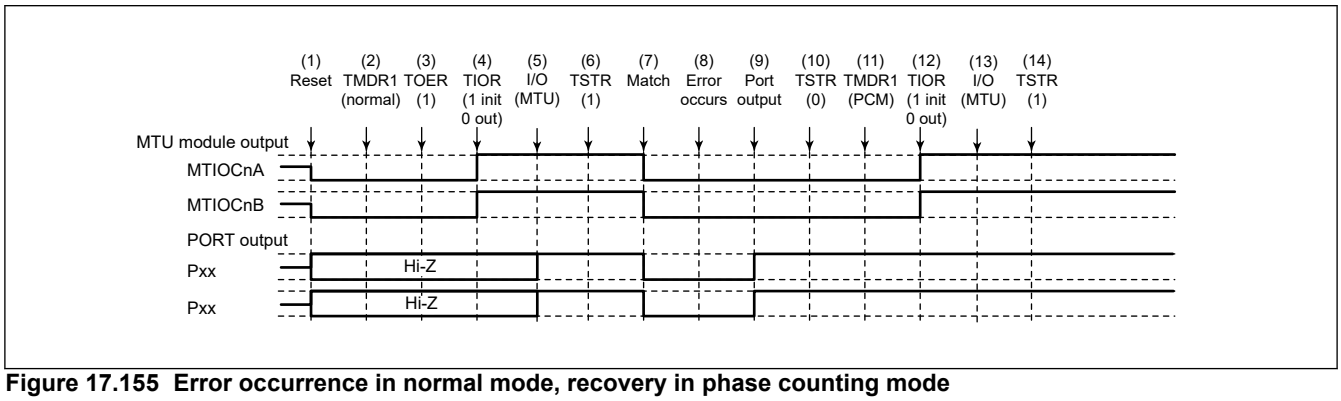


Figure 17.155 Error occurrence in normal mode, recovery in phase counting mode

(1) to (10) are the same as in Figure 17.152.

(11) Set the phase counting mode.

(12) Initialize the pins with the TIOR register.

(13) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA register.

Note: The phase counting mode can only be selected for MTU1 and MTU2, and therefore the TOERA register setting is not necessary.

17.7.3.5 Operation When Error Occurs in Normal Mode and Operation is Restarted in Complementary PWM Mode

Figure 17.156 shows a case in which an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

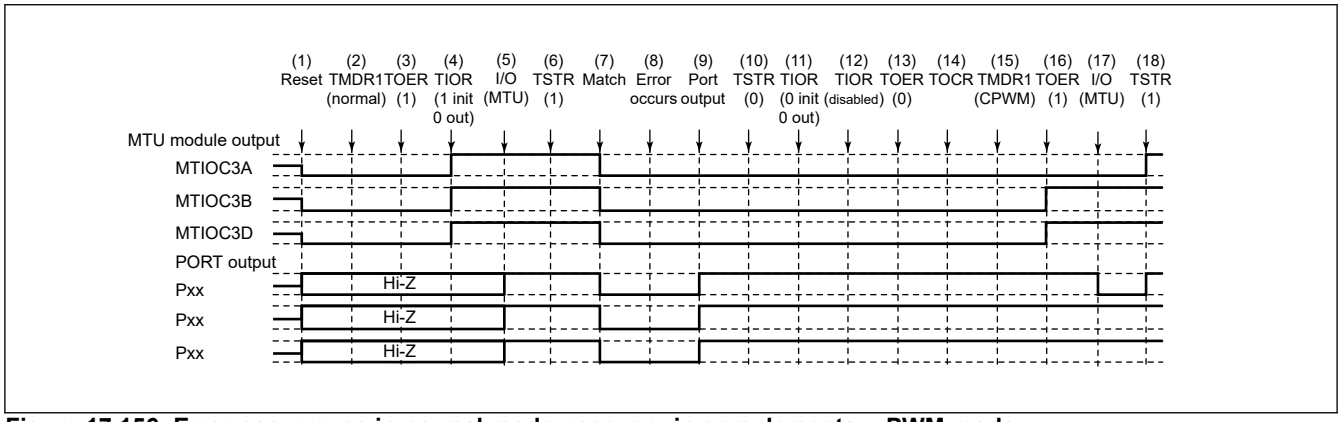


Figure 17.156 Error occurrence in normal mode, recovery in complementary PWM mode

(1) to (10) are the same as in Figure 17.152.

(11) Initialize the normal mode waveform generation block with the TIOR register.

(12) Disable operation of the normal mode waveform generation block with the TIOR register.

(13) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(14) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(15) Set complementary PWM mode.

(16) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(17) Set MTU output using the I/O port registers corresponding to the I/O ports.

(18) Restart operation by setting the TSTRA (TSTRB) register.

17.7.3.6 Operation When Error Occurs in Normal Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 17.157 shows a case in which an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.

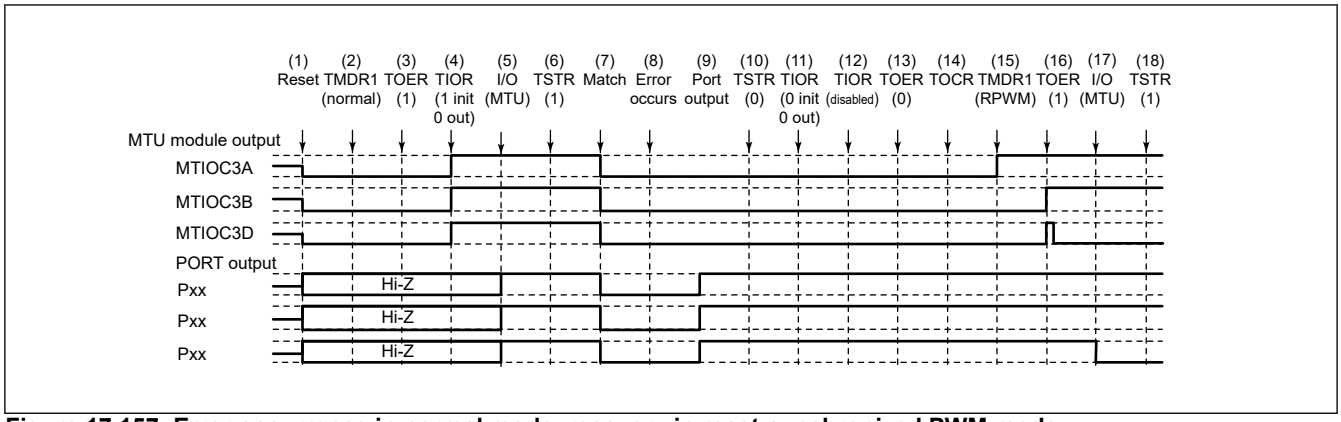


Figure 17.157 Error occurrence in normal mode, recovery in reset-synchronized PWM mode

(1) to (13) are the same as in Figure 17.156.

(14) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(15) Set reset-synchronized PWM mode.

(16) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(17) Set MTU output using the I/O port registers corresponding to the I/O ports.

(18) Restart operation by setting the TSTRA (TSTRB) register.

17.7.3.7 Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Normal Mode

Figure 17.158 shows a case in which an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

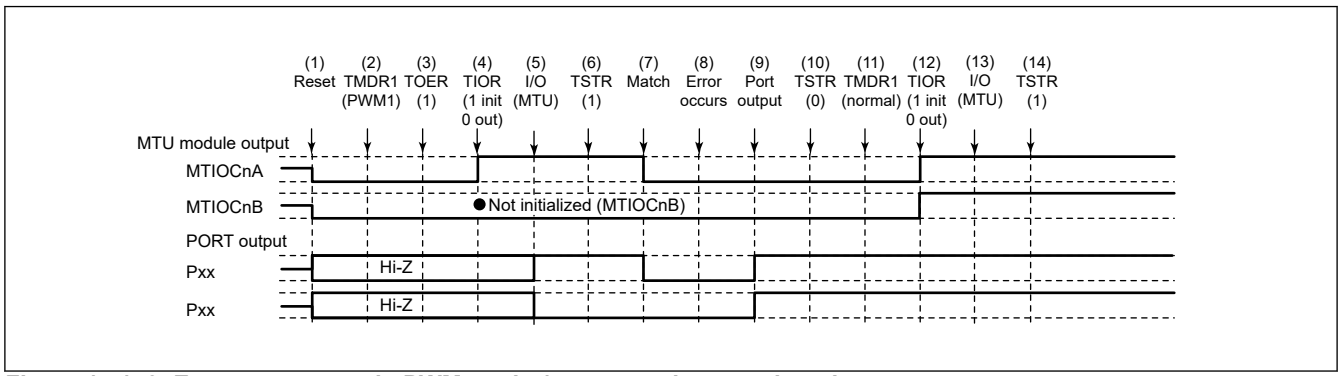


Figure 17.158 Error occurrence in PWM mode 1, recovery in normal mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.

(2) Set PWM mode 1.

(3) For MTU3 and MTU4 (MTU6 and MTU7), enable output with the TOERA (TOERB) register before initializing the pins with the TIOR register.

(4) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence. In PWM mode 1, the MTIOCnB side is not initialized.)

(5) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(6) Start count operation by setting the TSTRA (TSTRB) register.

(7) Output goes low on compare match occurrence.

(8) An error occurs.

(9) Allow non-active level output by setting the pins as general output ports using the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.

(10) Stop count operation by setting the TSTRA (TSTRB) register.

(11) Set normal mode.

(12) Initialize the pins with the TIOR register.

(13) Set MTU output using the I/O port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA (TSTRB) register.

17.7.3.8 Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 1

Figure 17.159 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

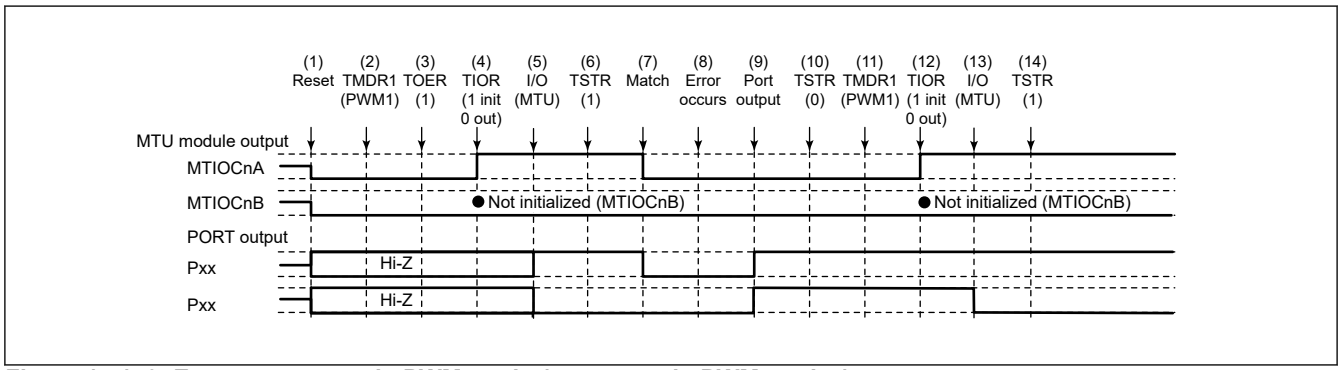


Figure 17.159 Error occurrence in PWM mode 1, recovery in PWM mode 1

(1) to (10) are the same as in Figure 17.158.

(11) This step is not necessary when restarting in PWM mode 1.

(12) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.)

(13) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA (TSTRB) register.

17.7.3.9 Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 2

Figure 17.160 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

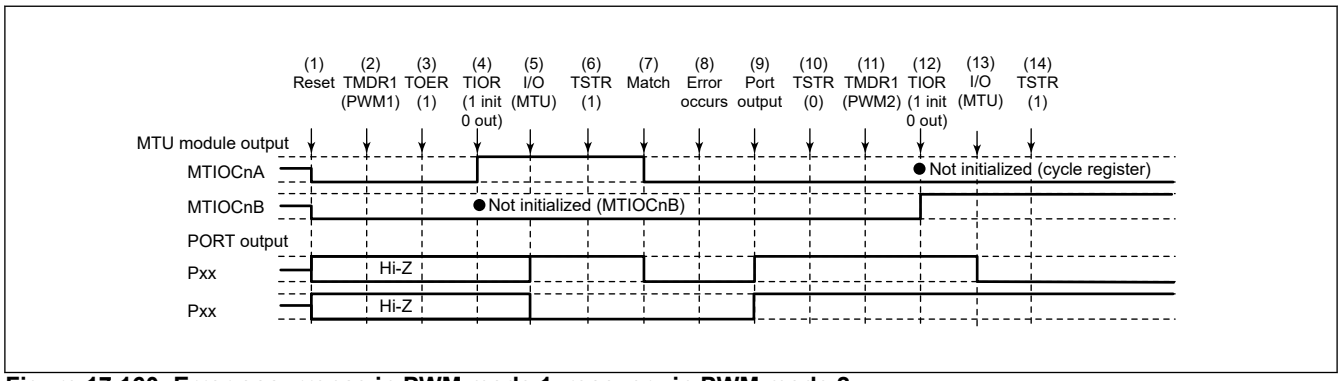


Figure 17.160 Error occurrence in PWM mode 1, recovery in PWM mode 2

(1) to (10) are the same as in Figure 17.158.

(11) Set PWM mode 2.

(12) Initialize the pins with the TIOR register. (In PWM mode 2, a waveform is not output to the pin that corresponds to the TGR register used as a period register. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.)

(13) Set MTU output using the I/O port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA register.

Note: PWM mode 2 can only be selected for MTU0 to MTU2, and therefore the TOERA register setting is not necessary.

17.7.3.10 Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Phase Counting Mode

Figure 17.161 shows a case in which an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

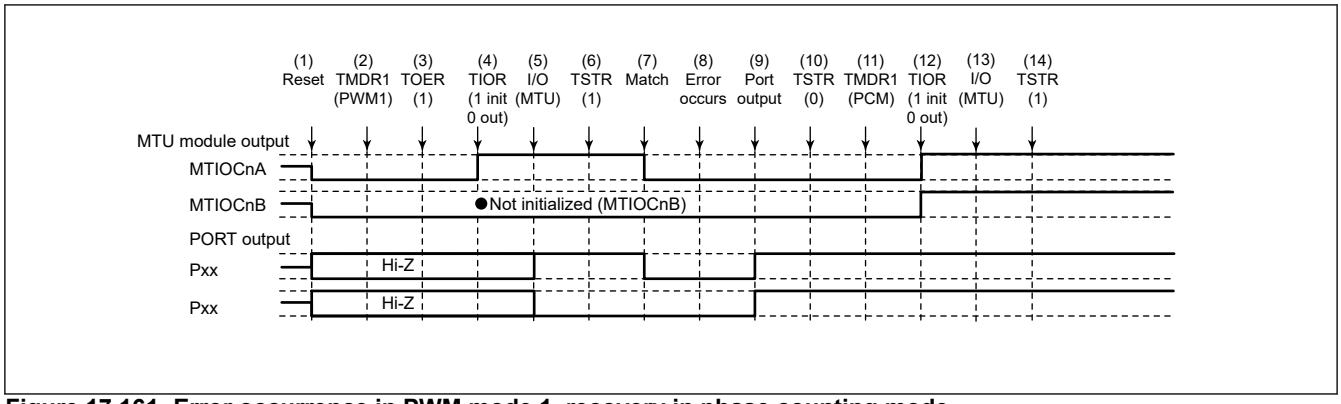


Figure 17.161 Error occurrence in PWM mode 1, recovery in phase counting mode

(1) to (10) are the same as in Figure 17.158.

(11) Set the phase counting mode.

(12) Initialize the pins with the TIOR register.

(13) Set MTU output using the I/O port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA register.

Note: The phase counting mode can only be selected for MTU1 and MTU2, and therefore the TOERA register setting is not necessary.

17.7.3.11 Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Complementary PWM Mode

Figure 17.162 shows a case in which an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

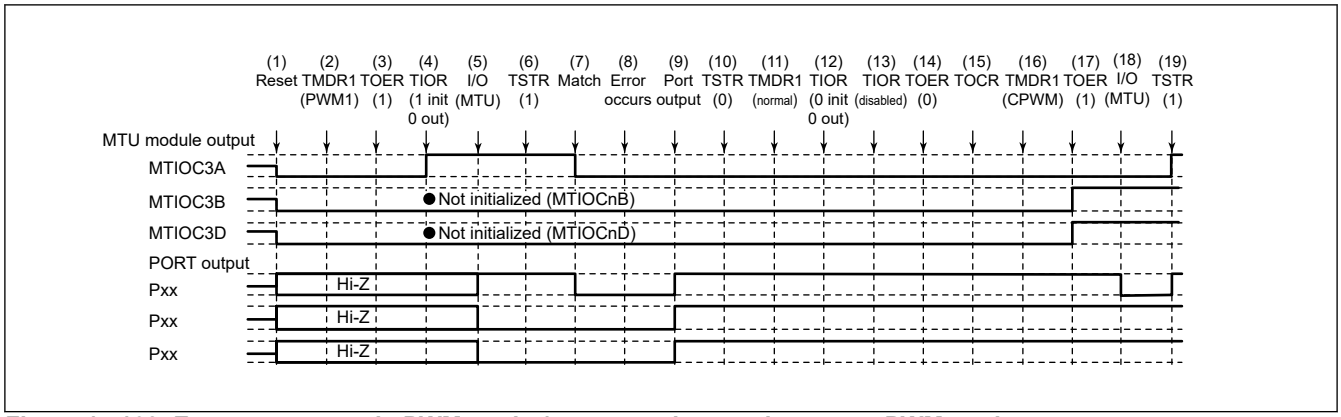


Figure 17.162 Error occurrence in PWM mode 1, recovery in complementary PWM mode

(1) to (10) are the same as in Figure 17.158.

(11) Set normal mode to initialize the normal mode waveform generation block.

(12) Initialize the PWM mode 1 waveform generation block with the TIOR register.

(13) Disable operation of the PWM mode 1 waveform generation block with the TIOR register.

(14) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(15) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TSTRB).

(16) Set complementary PWM mode.

(17) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(18) Set MTU output using the I/O port registers corresponding to the I/O ports.

(19) Restart operation by setting the TSTRA (TSTRB) register.

17.7.3.12 Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 17.163 shows a case in which an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

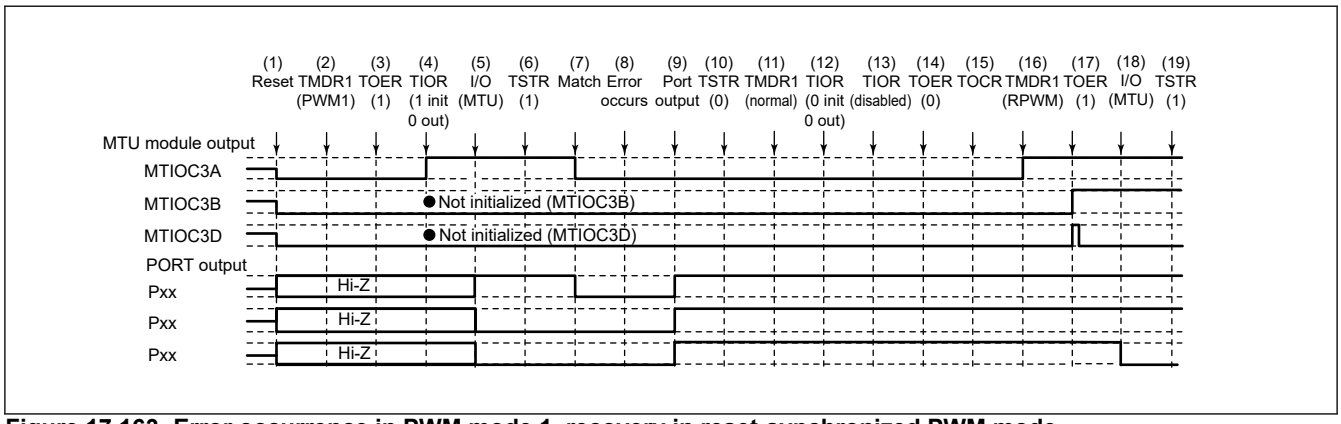


Figure 17.163 Error occurrence in PWM mode 1, recovery in reset-synchronized PWM mode

(1) to (14) are the same as in Figure 17.162.

(15) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(16) Set reset-synchronized PWM mode.

(17) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(18) Set MTU output using the I/O port registers corresponding to the I/O ports.

(19) Restart operation by setting the TSTRA (TSTRB) register.

17.7.3.13 Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Normal Mode

Figure 17.164 shows a case in which an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

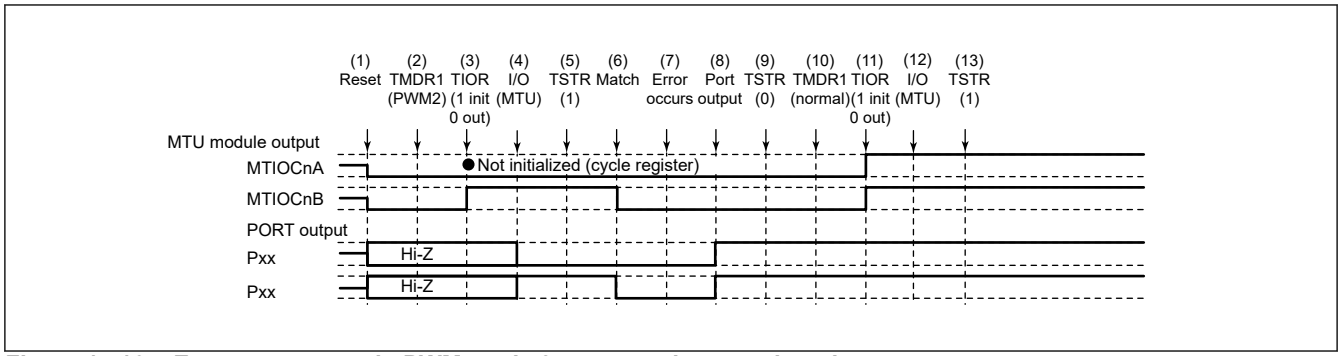


Figure 17.164 Error occurrence in PWM mode 2, recovery in normal mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.

(2) Set PWM mode 2.

(3) Initialize the pins with TIOR. (In the example, the initial output is a high level, and a low level is output on compare match occurrence. In PWM mode 2, the cycle register pins are not initialized. In the example, MTIOCnA is the cycle register.)

(4) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(5) Start count operation by setting the TSTRA (TSTRB) register.

(6) Output goes low on compare match occurrence.

(7) An error occurs.

(8) Allow non-active level output by setting the pins as general output ports using the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.

(9) Stop count operation by setting the TSTRA (TSTRB) register.

(10) Set normal mode.

(11) Initialize the pins with the TIOR register.

(12) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(13) Restart operation by setting the TSTRA (TSTRB) register.

17.7.3.14 Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 1

Figure 17.165 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

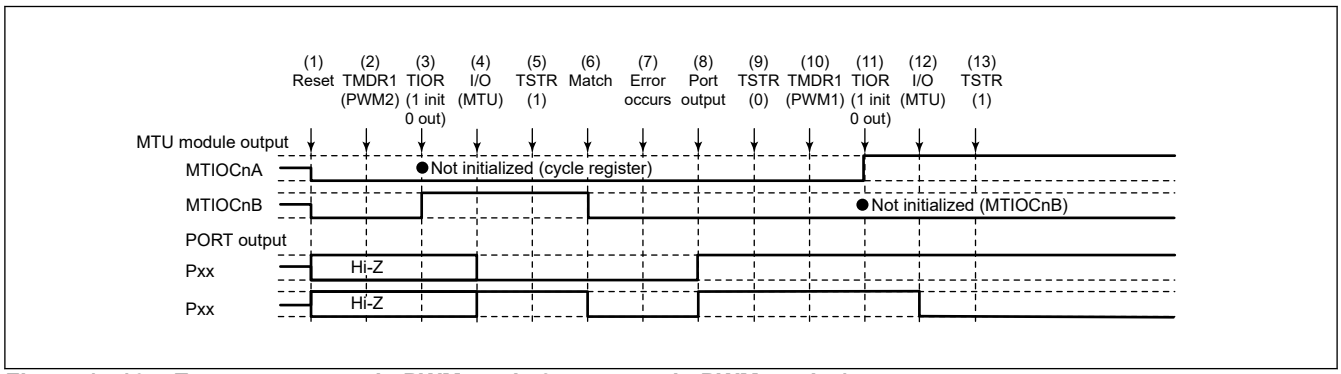


Figure 17.165 Error occurrence in PWM mode 2, recovery in PWM mode 1

(1) to (9) are the same as in Figure 17.164.

(10) Set PWM mode 1.

(11) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.)

(12) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(13) Restart operation by setting the TSTRA (TSTRB) register.

17.7.3.15 Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 2

Figure 17.166 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

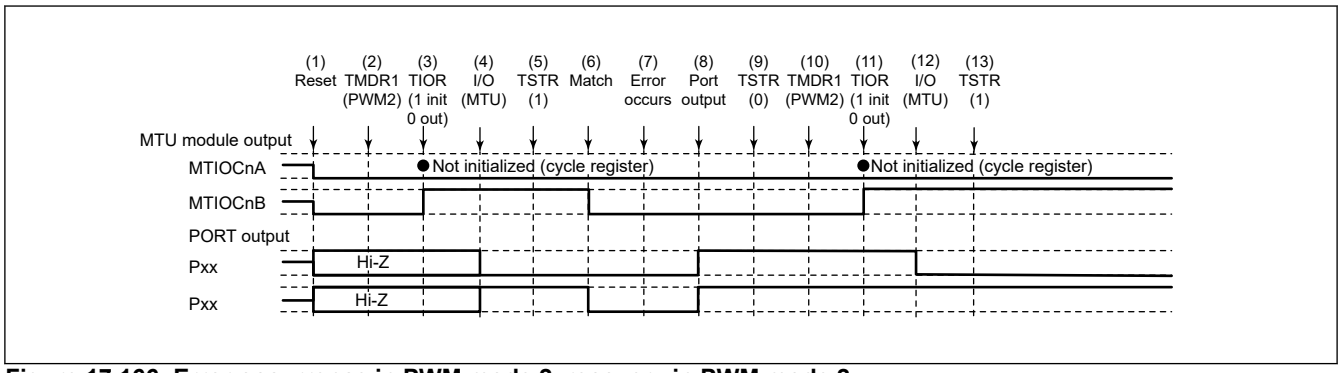


Figure 17.166 Error occurrence in PWM mode 2, recovery in PWM mode 2

(1) to (9) are the same as in Figure 17.164.

(10) This step is not necessary when restarting in PWM mode 2.

(11) Initialize the pins with TIOR. (In PWM mode 2, waveforms are not output to the cycle register pins. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.)

(12) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(13) Restart operation by setting the TSTRA register.

17.7.3.16 Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Phase Counting Mode

Figure 17.167 shows a case in which an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

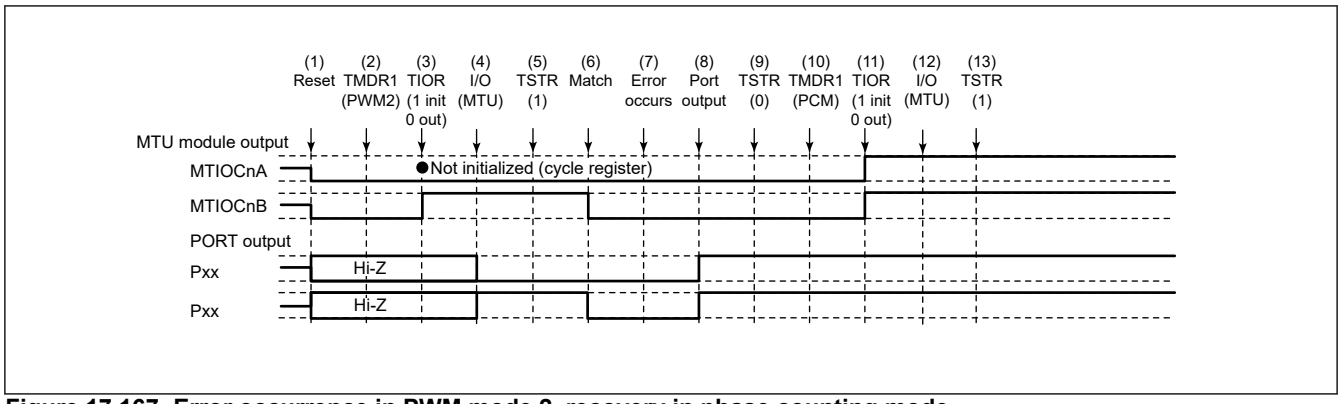


Figure 17.167 Error occurrence in PWM mode 2, recovery in phase counting mode

(1) to (9) are the same as in Figure 17.164.

(10) Set the phase counting mode.

(11) Initialize the pins with the TIOR register.

(12) Set MTU output using the I/O port registers corresponding to the I/O ports.

(13) Restart operation by setting the TSTRA register.

17.7.3.17 Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Normal Mode

Figure 17.168 shows a case in which an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

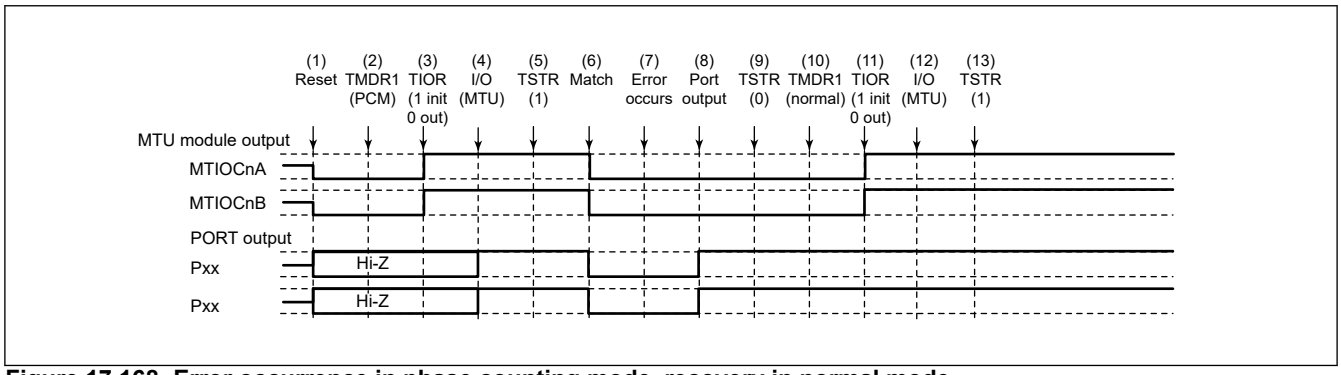


Figure 17.168 Error occurrence in phase counting mode, recovery in normal mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.

(2) Set phase counting mode.

(3) Initialize the pins with the TIOR register. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)

(4) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(5) Start count operation by setting the TSTRA (TSTRB) register.

(6) Output goes low on compare match occurrence.

(7) An error occurs.

(8) Allow non-active level output by setting the pins as general output ports using the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.

(9) Stop count operation by setting the TSTRA (TSTRB) register.

(10) Set normal mode.

(11) Initialize the pins with the TIOR register.

(12) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(13) Restart operation by setting the TSTRA (TSTRB) register.

17.7.3.18 Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 1

Figure 17.169 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

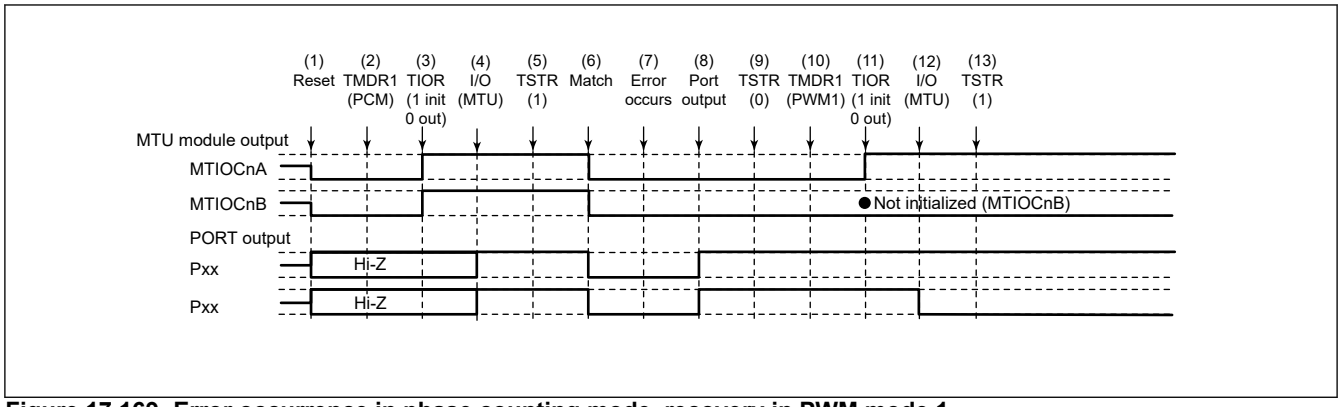


Figure 17.169 Error occurrence in phase counting mode, recovery in PWM mode 1

(1) to (9) are the same as in Figure 17.168.

(10) Set PWM mode 1.

(11) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.)

(12) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(13) Restart operation by setting the TSTRA (TSTRB) register.

17.7.3.19 Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 2

Figure 17.170 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

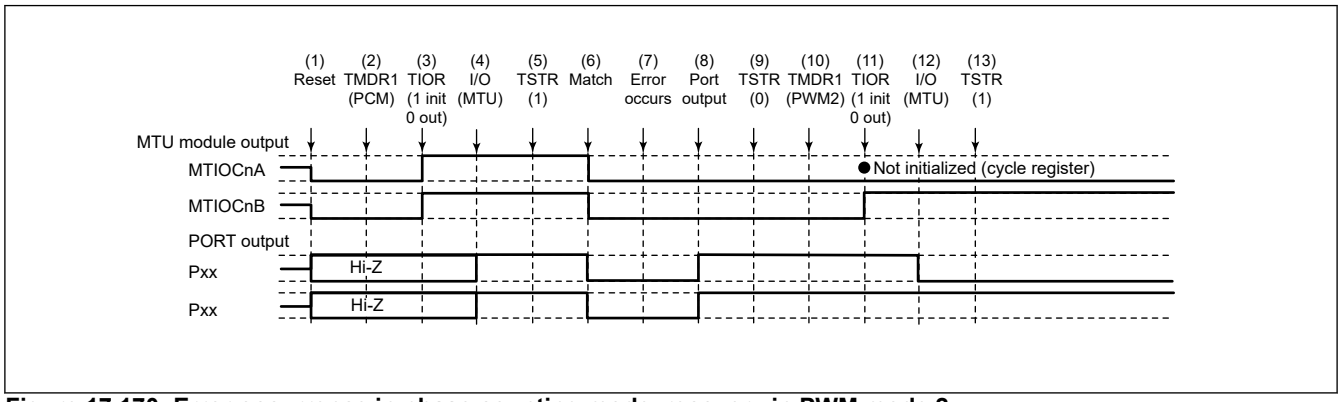


Figure 17.170 Error occurrence in phase counting mode, recovery in PWM mode 2

(1) to (9) are the same as in Figure 17.168.

(10) Set PWM mode 2.

(11) Initialize the pins with TIOR. (In PWM mode 2, waveforms are not output to the cycle register pins. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.)

(12) Set MTU output using the I/O port registers corresponding to the I/O ports.

(13) Restart operation by setting the TSTRA register.

17.7.3.20 Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Phase Counting Mode

Figure 17.171 shows a case in which an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

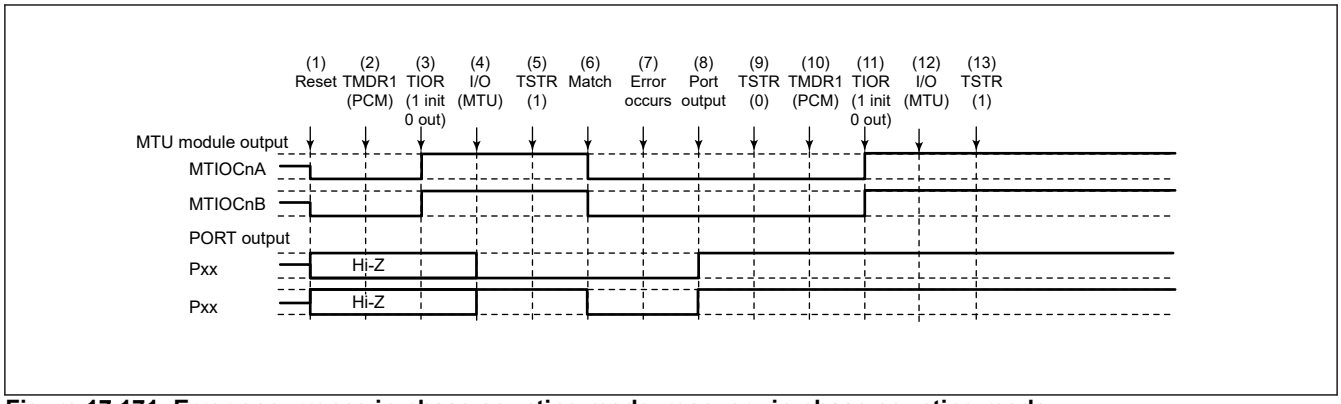


Figure 17.171 Error occurrence in phase counting mode, recovery in phase counting mode

(1) to (9) are the same as in Figure 17.168.

(10) This step is not necessary when restarting in phase counting mode.

(11) Initialize the pins with the TIOR register.

(12) Set MTU output using the I/O port registers corresponding to the I/O ports.

(13) Restart operation by setting the TSTRA register.

17.7.3.21 Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Normal Mode

Figure 17.172 shows a case in which an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

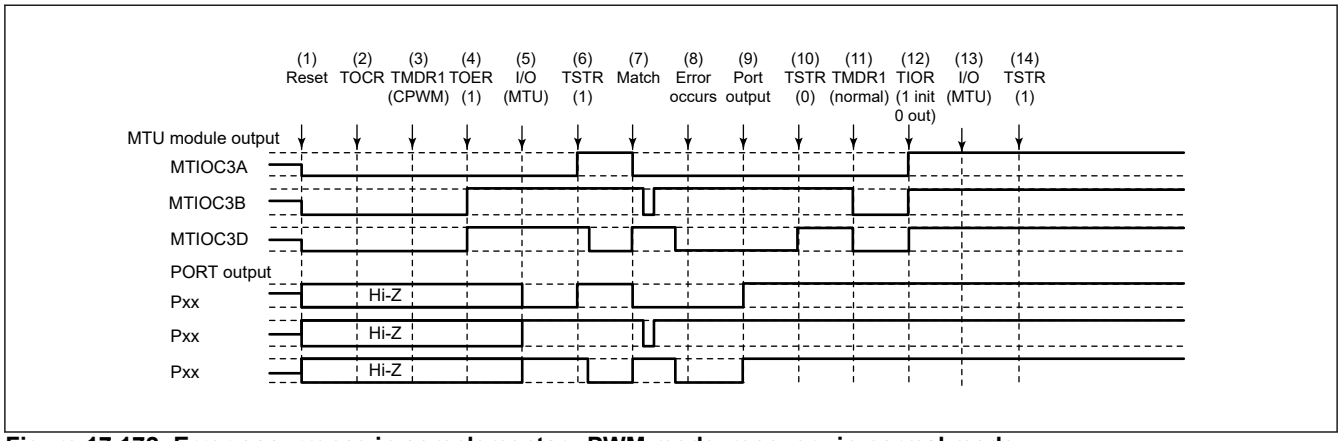


Figure 17.172 Error occurrence in complementary PWM mode, recovery in normal mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.

(2) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(3) Set complementary PWM mode.

(4) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(5) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(6) Start count operation by setting the TSTRA (TSTRB) register.

(7) The complementary PWM waveform is output on compare match occurrence.

(8) An error occurs.

(9) Allow non-active level output by setting the pins as general output ports using the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.

(10) Stop count operation by setting the TSTRA (TSTRB) register. (MTU output becomes the initial complementary PWM output value).

(11) Set normal mode (MTU output goes low).

(12) Initialize the pins with the TIOR register.

(13) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA (TSTRB) register.

17.7.3.22 Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in PWM Mode 1

Figure 17.173 shows a case in which an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

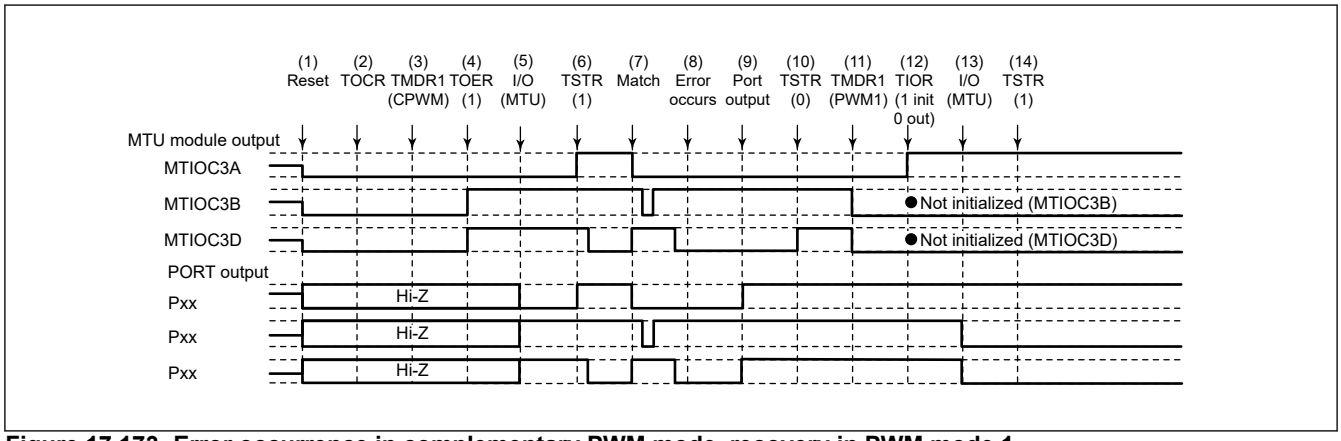


Figure 17.173 Error occurrence in complementary PWM mode, recovery in PWM mode 1

(1) to (10) are the same as in Figure 17.172.

(11) Set PWM mode 1 (MTU output goes low).

(12) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.)

(13) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA (TSTRB) register.

17.7.3.23 Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 17.174 shows a case in which an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (when operation is restarted using the period and duty settings at the time of stopping the counter).

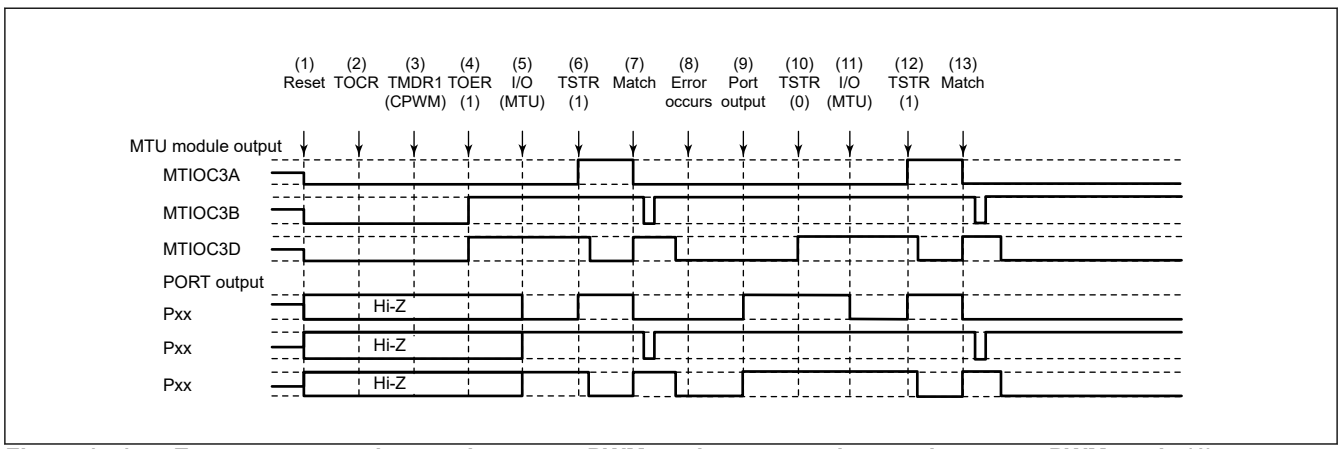


Figure 17.174 Error occurrence in complementary PWM mode, recovery in complementary PWM mode (1)

(1) to (10) are the same as in Figure 17.172.

(11) Set MTU output using the I/O port registers corresponding to the I/O ports.

(12) Restart operation by setting the TSTRA (TSTRB) register.

(13) The complementary PWM waveform is output on compare match occurrence.

17.7.3.24 Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode with New Settings

Figure 17.175 shows a case in which an error occurs in complementary PWM mode and operation is restarted in complementary PWM mode after re-setting (operation is restarted using new period and duty ratio settings).

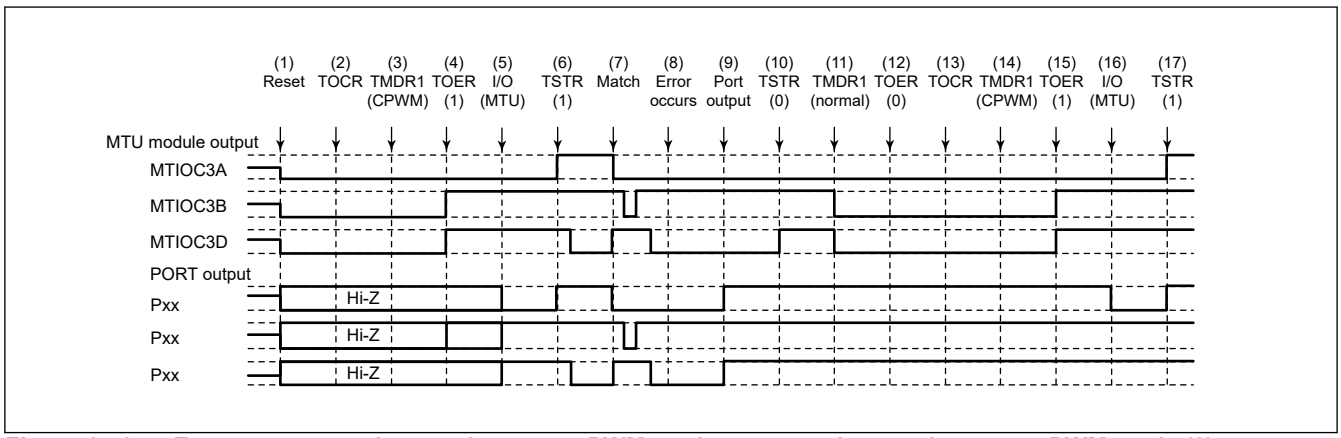


Figure 17.175 Error occurrence in complementary PWM mode, recovery in complementary PWM mode (2)

(1) to (10) are the same as in Figure 17.172.

(11) Set normal mode and make new settings (MTU output goes low).

(12) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(13) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(14) Set complementary PWM mode.

(15) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(16) Set MTU output using the I/O port registers corresponding to the I/O ports.

(17) Restart operation by setting the TSTRA (TSTRB) register.

17.7.3.25 Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 17.176 shows a case in which an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

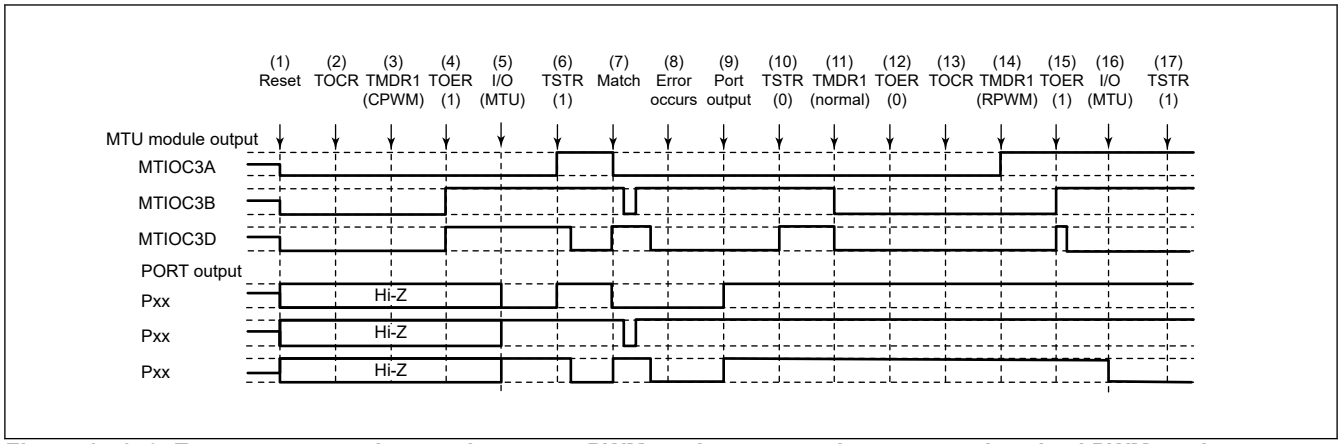


Figure 17.176 Error occurrence in complementary PWM mode, recovery in reset-synchronized PWM mode

(1) to (10) are the same as in Figure 17.172.

(11) Set normal mode (MTU output goes low).

(12) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(13) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(14) Set reset-synchronized PWM mode.

(15) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(16) Set MTU output using the I/O port registers corresponding to the I/O ports.

(17) Restart operation by setting the TSTRA (TSTRB) register.

17.7.3.26 Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Normal Mode

Figure 17.177 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

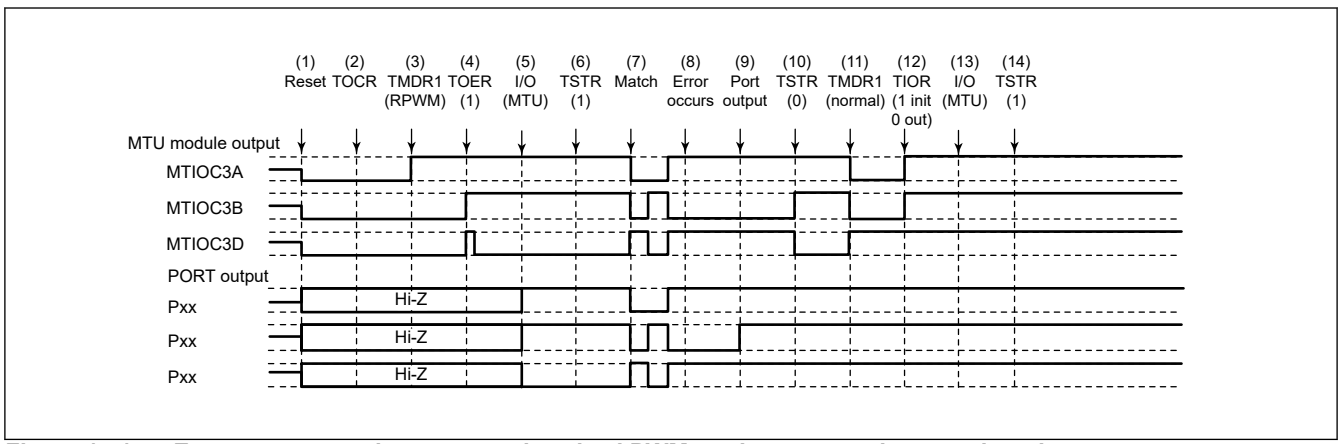


Figure 17.177 Error occurrence in reset-synchronized PWM mode, recovery in normal mode

(1) After a reset, the MTU output goes low and the ports enter high-impedance state.

(2) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(3) Set reset-synchronized PWM mode.

(4) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(5) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(6) Start count operation by setting the TSTRA (TSTRB) register.

(7) The reset-synchronized PWM waveform is output on compare match occurrence.

(8) An error occurs.

(9) Allow non-active level output by setting the pins as general output ports using the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.

(10) Stop count operation by setting the TSTRA (TSTRB) register. (MTU output becomes the initial reset-synchronized PWM output value.)

(11) Set normal mode (positive-phase MTU output goes low, and negative-phase output goes high).

(12) Initialize the pins with the TIOR register.

(13) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA (TSTRB) register.

17.7.3.27 Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in PWM Mode 1

Figure 17.178 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in PWM mode 1 after re-setting.

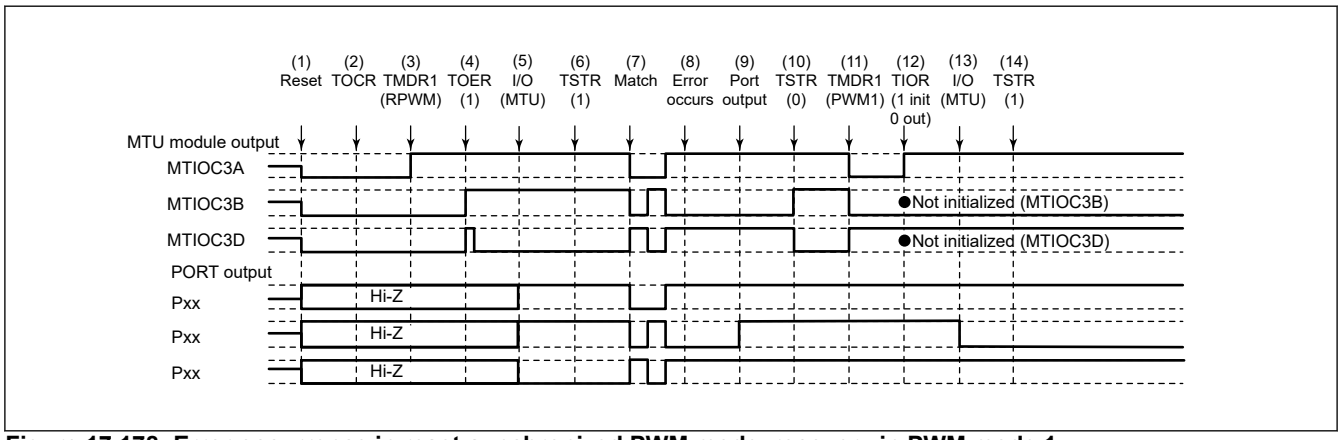


Figure 17.178 Error occurrence in reset-synchronized PWM mode, recovery in PWM mode 1

(1) to (10) are the same as in Figure 17.177.

(11) Set PWM mode 1 (positive-phase MTU output goes low, and negative-phase output goes high).

(12) Initialize the pins with the TIOR register. (In PWM mode 1, waveforms are not output to the MTIOCnB (MTIOCnD) pins. To output a specified level, make necessary settings for general output ports in the port mode control register (PMCm), port mode register (PMm), and port register (Pm) of the I/O ports.)

(13) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA (TSTRB) register.

17.7.3.28 Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 17.179 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in complementary PWM mode after re-setting.

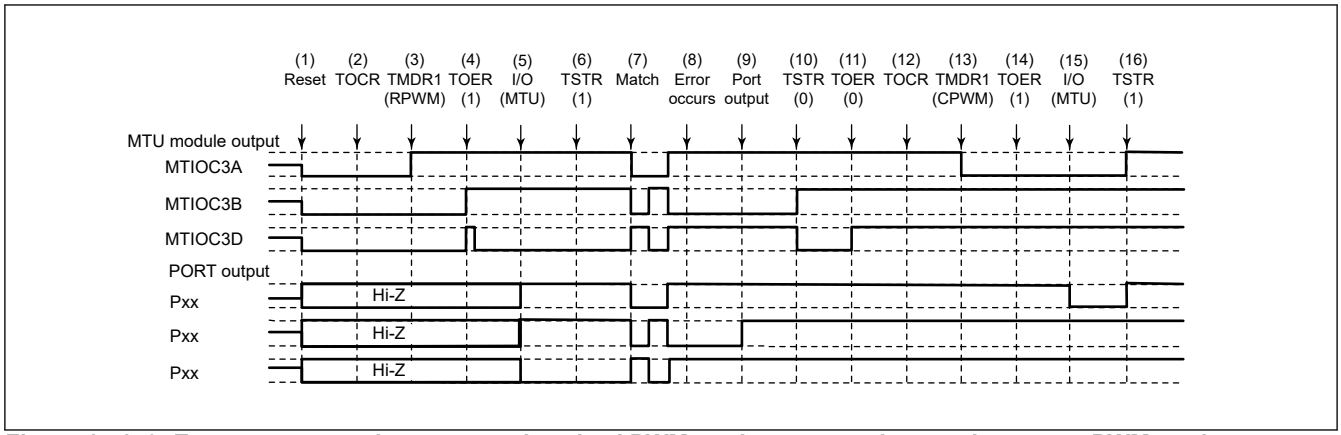


Figure 17.179 Error occurrence in reset-synchronized PWM mode, recovery in complementary PWM mode

(1) to (10) are the same as in Figure 17.177.

(11) Disable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(12) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A (TOCR1B and TOCR2B).

(13) Set complementary PWM mode (MTU cyclic output pin goes low).

(14) Enable output in MTU3 and MTU4 (MTU6 and MTU7) with the TOERA (TOERB) register.

(15) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(16) Restart operation by setting the TSTRA (TSTRB) register.

17.7.3.29 Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 17.180 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

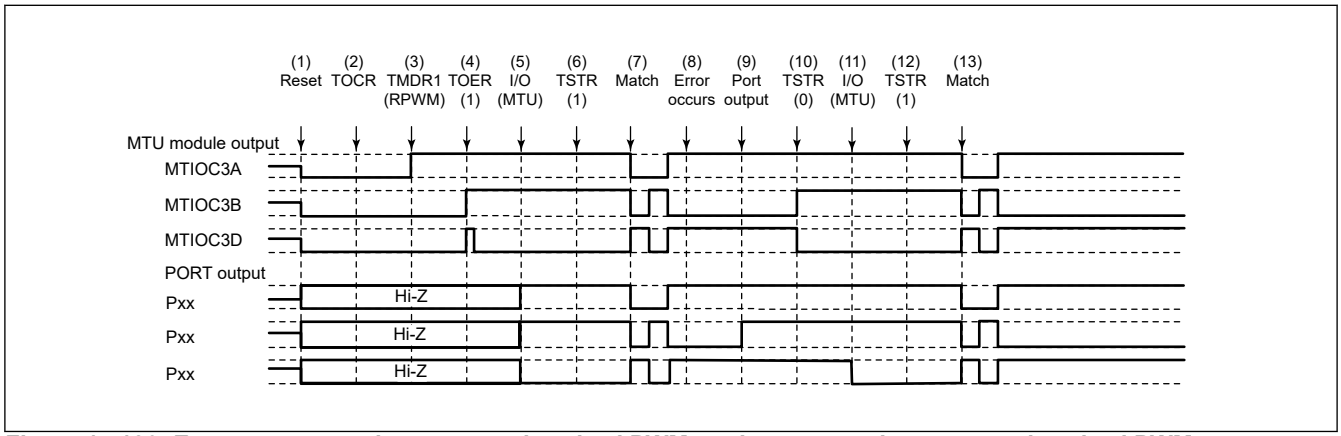


Figure 17.180 Error occurrence in reset-synchronized PWM mode, recovery in reset-synchronized PWM mode

(1) to (10) are the same as in Figure 17.177.

(11) Set MTU output using the I/O Port registers corresponding to the I/O ports.

(12) Restart operation by setting the TSTRA (TSTRB) register.

(13) The reset-synchronized PWM waveform is output on compare match occurrence.

17.8 Operations Linked by the ELC

17.8.1 Event Signal Output to the ELC

The MTU is capable of operation linked with another module set in advance when its interrupt request signal is used as an event signal by the event link controller (ELC).

The MTU outputs the event signal regardless of the setting of the corresponding interrupt request enable bit.

17.8.2 MTU Operations in Response to Receiving Event Signals from the ELC

The MTU can perform the following operations in response to the event set in advance in the ELC_SSEL0 register of the event link controller (ELC).

(1) Start Counting Operation

Select "counting is started" as the operation of the MTU by setting the ELOPA or ELOPB register in the ELC. The ELOPA register controls the operation of MTU0 and MTU3 and the ELOPB register controls the operation of MTU4. When the event specified in the ELC_SSEL0 register occurs, the CSTn bit in the TSTRA register shown in Table 17.75 is set to 1, and the MTU counter starts.

However, when the specified event is generated while the CSTn bit in the TSTRA register has already been set to 1, the event has no effect. Table 17.75 lists the TSTRA register bits used for each channel.

Table 17.75 Counter start bit set by the ELC

Channel no.	Counter start bit
MTU0	TSTRA.CST0 bit
MTU3	TSTRA.CST3 bit
MTU4	TSTRA.CST4 bit

(2) Input Capture Operation

Select "input capture" as the operation of the MTU by setting the ELOPA or ELOPB register in the ELC. The ELOPA register controls the operation of MTU0 and MTU3 and the ELOPB register controls the operation of MTU4. When the event specified in the ELC_SSEL0 register occurs, the value of the TCNT register is captured in the TGR register. When using input capture in response to an event, the corresponding bit of the TIOR register in the MTU should be set for input capture and the CSTn bit of TSTRA register should be set to 1 to start counting by the counter.

In this case, the TIOCnA pin (input capture pin) input has no effect.

Table 17.76 lists the timer general register and I/O control bit used for each channel in input capture operations in response to the ELC.

Table 17.76 Timer general register and I/O control bit used in the Input capture operation

Channel no.	Timer general register	I/O control bit
MTU0	MTU0.TGRA	MTU0.TIORH.IOA[3:0] bits
MTU3	MTU3.TGRA	MTU3.TIORH.IOA[3:0] bits
MTU4	MTU4.TGRA	MTU4.TIORH.IOA[3:0] bits

(3) Restart Counting (Clear Counter) Operation

Select "counting is restarted" as the operation of the MTU by setting the ELOPA or ELOPB register in the ELC. The ELOPA register controls the operation of MTU0 and MTU3 and the ELOPB register controls the operation of MTU4. When the event specified in the ELC_SSEL0 register occurs, the TCNT register is cleared. If the corresponding CSTn bit in the TSTRA register is set to 1, counting continues. For the CSTn bits in the TSTRA register, see Table 17.75.

17.8.3 Usage Notes on MTU Operation by Event Signal Reception from the ELC

The following notes on usage apply when the MTU is used in event link operation.

(1) Start Counting

If the event specified in the ELC_SSEL0 register occurs during a cycle of writing to a CSTn bit in the TSTRA register, writing to the CSTn bit in the TSTRA register does not proceed because setting of the bit to 1 due to the event takes priority.

(2) Restart Counting (Clear Counter)

If the event specified in the ELC_SSEL0 register occurs during a cycle of writing to the TCNT counter, writing to the TCNT counter does not proceed because clearing of the counter due to the event takes priority.

In addition, for MTU3 and MTU4 in complementary PWM mode, do not use the counter restarting by the ELC.

18. Port Output Enable 3 (POE3)

18.1 Overview

The LSI incorporates a port output enable 3 (POE3) module that can be used to place output pins for the MTU3 in the high-impedance state under various conditions. Every output signal is put in the high-impedance state when the output is disabled.

Table 18.1 lists the specifications of the POE3 and Figure 18.1 shows a block diagram of the POE3.

Table 18.1 POE3 specifications

Parameter	Description
Target pins to be placed in the high-impedance state	• MTU3 output pins MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D) MTU3 pins (MTIOC3B, MTIOC3D) MTU4 pins (MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D) MTU6 pins (MTIOC6B, MTIOC6D) MTU7 pins (MTIOC7A, MTIOC7B, MTIOC7C, MTIOC7D)
Conditions for the high-impedance state	• Setting pins as inputs: setting the POE0#, POE4#, POE8#, POE10#, and POE11# pins as inputs (falling edge or low-level sampling) • Short-circuits between output pins: a match (short circuit) between the output signal levels at the active level over one or more cycle on the following combination of pins: – MTU complementary PWM output pins 1. MTIOC3B and MTIOC3D 2. MTIOC4A and MTIOC4C 3. MTIOC4B and MTIOC4D 4. MTIOC6B and MTIOC6D 5. MTIOC7A and MTIOC7C 6. MTIOC7B and MTIOC7D • SPOER register setting being made • Detection that the clock generation circuit had stopped oscillating or the oscillation is abnormal (CLMA0: PLL0, CLMA1: PLL1, CLMA3: main clock) • DSMIF error interrupt
Function	• Each of the POE0#, POE4#, POE8#, POE10#, and POE11# input pins can be set for falling edge, PCLKH/8 × 16, PCLKH/16 × 16, or PCLKH/128 × 16 low-level sampling. • The outputs of the target pins can be in the high-impedance state by falling-edge or low-level sampling of the POE0#, POE4#, POE8#, POE10#, or POE11# pin. • The outputs of the target pins can be in the high-impedance state when oscillation stop is detected by the oscillation stop detection function of the clock generator. • The outputs of the target pins can be in the high-impedance state when DSMIF error occurs. • The MTU complementary PWM outputs can be in the high-impedance state when output levels of the MTU complementary PWM output pins are compared and simultaneous active-level output continues for one cycle or more. • The outputs of the target pins can be in the high-impedance state by modifying the settings of the POE3 registers. • Interrupts can be generated by input-level sampling or output-level comparison results.

The POE3 has input-level detection circuits, pin selection circuits, output-level comparison circuits, and a high-impedance request/interrupt request generating circuit as shown in Figure 18.1.

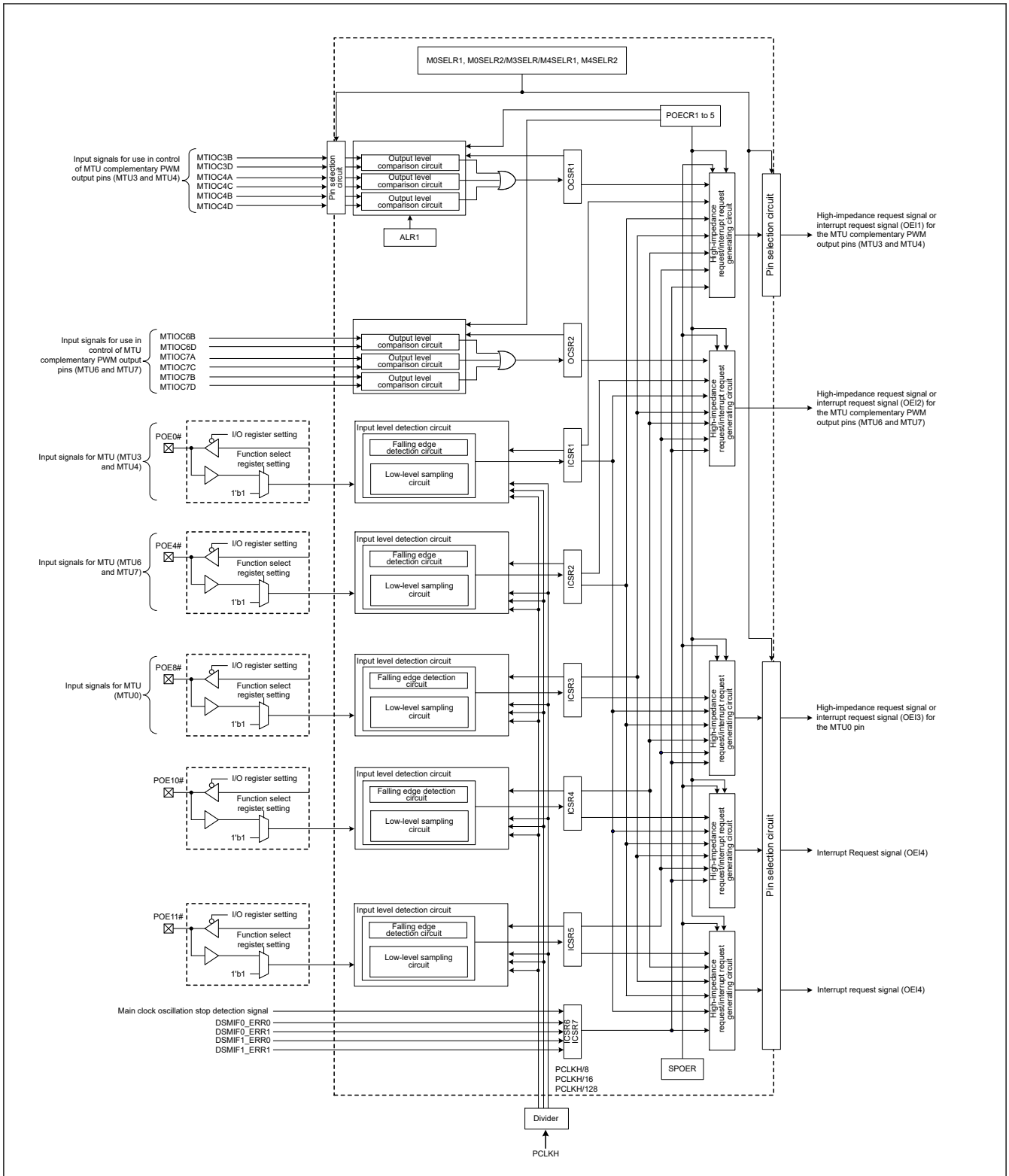


Figure 18.1 POE3 block diagram

Table 18.2 shows I/O pins to be used by the POE3.

Table 18.2 POE3 I/O pins

Pin name	I/O	Description
POE0#	Input	Input pin for the request signal to place the MTU3 and MTU4 pins for MTU complementary PWM output in high-impedance state. In accord with register settings, this pin is also capable of placing the MTU0, MTU6, and MTU7 pins in the high-impedance state.
POE4#	Input	Input pin for the request signal to place the MTU6 and MTU7 pins for MTU complementary PWM output in high-impedance state. In accord with register settings, this pin is also capable of placing the MTU0, MTU3, and MTU4 pins in the high-impedance state.
POE8#	Input	Input pin for the request signal to place the pins for MTU0 in high-impedance state. In accord with register settings, this pin is also capable of placing the MTU3 and MTU4 pins or MTU6 and MTU7 pins for MTU complementary PWM output.
POE10#	Input	In accord with register settings, this pin is capable of placing the any target pins in high-impedance state.
POE11#	Input	In accord with register settings, this pin is capable of placing the any target pins in high-impedance state.

Table 18.3 shows output-level comparisons with pin combinations.

Table 18.3 Pin combinations

Pin combination	I/O	Description
MTIOC3B and MTIOC3D	Output	The MTU3 and MTU4 pins for MTU complementary PWM output are placed in high-impedance state when both pins of a pair simultaneously output the active level*1 for one or more cycles of the peripheral clock (PCLKH). Pin combinations for output comparison and high-impedance control can be selected by registers of POE3.
MTIOC4A and MTIOC4C	Output	
MTIOC4B and MTIOC4D	Output	
MTIOC6B and MTIOC6D	Output	The MTU6 and MTU7 pins for MTU complementary PWM output are placed in high-impedance state when both pins of a pair simultaneously output the active level*2 for one or more cycles of the peripheral clock (PCLKH). Pin combinations for output comparison and high-impedance control can be selected by registers of POE3.
MTIOC7A and MTIOC7C	Output	
MTIOC7B and MTIOC7D	Output	

Note 1. The low level is output when the OLSP bit in TOCR1A of MTUn is 0 with the TOCS bit in TOCR1A of MTUn cleared to 0, or the high level is output when the OLSP bit is 1. Otherwise, the low level is output when the OLS3N, OLS3P, OLS2N, OLS2P, OLS1N, and OLS1P bits in TOCR2A of MTUn are 0 with the TOCS bit in TOCR1A of MTUn set to 1, or the high level is output when these bits are 1.

Note 2. The low level is output when the OLSP bit in TOCR1B of MTUn is 0 with the TOCS bit in TOCR1B of MTUn cleared to 0, or the high level is output when the OLSP bit is 1. Otherwise, the low level is output when the OLS3N, OLS3P, OLS2N, OLS2P, OLS1N, and OLS1P bits in TOCR2B of MTUn are 0 with the TOCS bit in TOCR1B of MTUn set to 1, or the high level is output when these bits are 1.

18.2 Register Descriptions

The POE3 registers are initialized by a reset.

18.2.1 ICSR1 : Input Level Control/Status Register 1

Base address: POE3 = 0x9000_5000

Offset address: 0x00

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	POE0F	—	—	—	PIE1	—	—	—	—	—	—	POE0M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	POE0M[1:0]	POE0 Mode Select 0 0: Accepts a request on the falling edge of POE0# pin input 0 1: Accepts a request when POE0# pin input has been sampled 16 times at PCLKH/8 clock pulses and all are low level 1 0: Accepts a request when POE0# pin input has been sampled 16 times at PCLKH/16 clock pulses and all are low level 1 1: Accepts a request when POE0# pin input has been sampled 16 times at PCLKH/128 clock pulses and all are low level	R/W*1

Bit	Symbol	Function	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	PIE1	Port Interrupt Enable 1 0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W
12	POE0F	POE0 Flag 0: Indicates that a high-impedance request has not been input to the POE0# pin 1: Indicates that a high-impedance request has been input to the POE0# pin	R/W*2
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed method.

The ICSR1 register selects the input mode for the POE0# pin, controls the enable/disable of interrupts, and indicates status.

POE0M[1:0] bits (POE0 Mode Select)

These bits select the input mode of the POE0# pin.

PIE1 bit (Port Interrupt Enable 1)

This bit enables or disables interrupt requests when the POE0F flag in ICSR1 is set to 1.

POE0F flag (POE0 Flag)

This flag indicates that a high-impedance request has been input to the POE0# pin.

[Setting condition]

- When the input set by the POE0M[1:0] bits occurs at the POE0# pin

[Clearing condition]

- By writing 0 to the POE0F flag after reading POE0F = 1

18.2.2 ICSR2 : Input Level Control/Status Register 2

Base address: POE3 = 0x9000_5000

Offset address: 0x04

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	POE4F	—	—	—	PIE2	—	—	—	—	—	—	POE4M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	POE4M[1:0]	POE4 Mode Select 0 0: Accepts a request on the falling edge of POE4# pin input 0 1: Accepts a request when POE4# pin input has been sampled 16 times at PCLKH/8 clock pulses and all are low level 1 0: Accepts a request when POE4# pin input has been sampled 16 times at PCLKH/16 clock pulses and all are low level 1 1: Accepts a request when POE4# pin input has been sampled 16 times at PCLKH/128 clock pulses and all are low level	R/W*1
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	PIE2	Port Interrupt Enable 2 0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
11:9	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
12	POE4F	POE4 Flag 0: Indicates that a high-impedance request has not been input to the POE4# pin 1: Indicates that a high-impedance request has been input to the POE4# pin	R/W*2
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed method.

The ICSR2 register selects the input mode for the POE4# pin, controls the enable/disable of interrupts, and indicates status.

POE4M[1:0] bits (POE4 Mode Select)

These bits select the input mode of the POE4# pin.

PIE2 bit (Port Interrupt Enable 2)

This bit enables or disables interrupt requests when the POE4F flag in ICSR2 is set to 1.

POE4F flag (POE4 Flag)

This flag indicates that a high-impedance request has been input to the POE4# pin.

[Setting condition]

- When the input set by the POE4M[1:0] bits occurs at the POE4# pin

[Clearing condition]

- By writing 0 to the POE4F flag after reading POE4F = 1

18.2.3 ICSR3 : Input Level Control/Status Register 3

Base address: POE3 = 0x9000_5000

Offset address: 0x08

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	POE8F	—	—	POE8E	PIE3	—	—	—	—	—	—	POE8M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	POE8M[1:0]	POE8 Mode Select 0 0: Accepts a request on the falling edge of POE8# pin input 0 1: Accepts a request when POE8# pin input has been sampled 16 times at PCLKH/8 clock pulses and all are low level 1 0: Accepts a request when POE8# pin input has been sampled 16 times at PCLKH/16 clock pulses and all are low level 1 1: Accepts a request when POE8# pin input has been sampled 16 times at PCLKH/128 clock pulses and all are low level	R/W*1
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	PIE3	Port Interrupt Enable 3 0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
9	POE8E	POE8 High-Impedance Enable 0: Does not place the pin in the high-impedance state 1: Places the pin in the high-impedance state	R/W*1
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	POE8F	POE8 Flag 0: Indicates that a high-impedance request has not been input to the POE8# pin. 1: Indicates that a high-impedance request has been input to the POE8# pin.	R/W*2
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed method.

The ICSR3 register selects the input mode for the POE8# pin, controls the enable/disable of interrupts, and indicates status.

POE8M[1:0] bits (POE8 Mode Select)

These bits select the input mode of the POE8# pin.

PIE3 bit (Port Interrupt Enable 3)

This bit enables or disables interrupt requests when the POE8F flag in ICSR3 is set to 1.

POE8E bit (POE8 High-Impedance Enable)

This bit specifies whether to place the corresponding pin in the high-impedance state when the POE8F flag is set to 1.

POE8F flag (POE8 Flag)

This flag indicates that a high-impedance request has been input to the POE8# pin.

[Setting condition]

- When the input set by the POE8M[1:0] bits occurs at the POE8# pin

[Clearing condition]

- By writing 0 to the POE8F flag after reading POE8F = 1

18.2.4 ICSR4 : Input Level Control/Status Register 4

Base address: POE3 = 0x9000_5000

Offset address: 0x16

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	POE10F	—	—	POE10E	PIE4	—	—	—	—	—	—	POE10M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	POE10M[1:0]	POE10 Mode Select 0 0: Accepts a request on the falling edge of POE10# pin input 0 1: Accepts a request when POE10# pin input has been sampled 16 times at PCLKH/8 clock pulses and all are low level 1 0: Accepts a request when POE10# pin input has been sampled 16 times at PCLKH/16 clock pulses and all are low level 1 1: Accepts a request when POE10# pin input has been sampled 16 times at PCLKH/128 clock pulses and all are low level	R/W*1
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	PIE4	Port Interrupt Enable 4 0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
9	POE10E	POE10 High-Impedance Enable 0: Does not place the pin in the high-impedance state 1: Places the pin in the high-impedance state	R/W*1
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	POE10F	POE10 Flag 0: Indicates that a high-impedance request has not been input to the POE10# pin 1: Indicates that a high-impedance request has been input to the POE10# pin	R/W*2
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed method.

The ICSR4 register selects the input mode for the POE10# pin, controls the enable/disable of interrupts, and indicates status.

POE10M[1:0] bits (POE10 Mode Select)

These bits select the input mode of the POE10# pin.

PIE4 bit (Port Interrupt Enable 4)

This bit enables or disables interrupt requests when the POE10F flag in ICSR4 is set to 1.

POE10E bit (POE10 High-Impedance Enable)

This bit specifies whether to place the corresponding pin in the high-impedance state when the POE10F flag is set to 1.

POE10F flag (POE10 Flag)

This flag indicates that a high-impedance request has been input to the POE10# pin.

[Setting condition]

- When the input set by the POE10M[1:0] bits occurs at the POE10# pin

[Clearing condition]

- By writing 0 to the POE10F flag after reading POE10F = 1

18.2.5 ICSR5 : Input Level Control/Status Register 5

Base address: POE3 = 0x9000_5000

Offset address: 0x18

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	POE11F	—	—	POE11E	PIE5	—	—	—	—	—	—	POE11M[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	POE11M[1:0]	POE11 Mode Select 0 0: Accepts a request on the falling edge of POE11# pin input 0 1: Accepts a request when POE11# pin input has been sampled 16 times at PCLKH/8 clock pulses and all are low level 1 0: Accepts a request when POE11# pin input has been sampled 16 times at PCLKH/16 clock pulses and all are low level 1 1: Accepts a request when POE11# pin input has been sampled 16 times at PCLKH/128 clock pulses and all are low level	R/W*1
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	PIE5	Port Interrupt Enable 5 0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
9	POE11E	POE11 High-Impedance Enable 0: Does not place the pin in the high-impedance state 1: Places the pin in the high-impedance state	R/W*1
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	POE11F	POE11 Flag 0: Indicates that a high-impedance request has not been input to the POE11# pin 1: Indicates that a high-impedance request has been input to the POE11# pin	R/W*2
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: For the ICSR5 register, set the values the same as those in the ICSR4 register.
Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed method.

The ICSR5 register selects the pin input mode of the POE11#, controls the enable/disable of interrupts, and indicates status.

POE11M[1:0] bits (POE11 Mode Select)

These bits select the input mode of the POE11# pin.

PIE5 bit (Port Interrupt Enable 5)

This bit enables or disables interrupt requests when the POE11F flag in ICSR5 is set to 1.

POE11E bit (POE11 High-Impedance Enable)

This bit specifies whether to place the corresponding pin in the high-impedance state when the POE11F flag is set to 1. The interrupt output is multiplexed with OEI4.

POE11F flag (POE11 Flag)

This flag indicates that a high-impedance request has been input to the POE11# pin.

[Setting condition]

- When the input set by the POE11M[1:0] bits occurs at the POE11# pin

[Clearing condition]

- By writing 0 to the POE11F flag after reading POE11F = 1

18.2.6 ICSR6 : Input Level Control/Status Register 6

Base address: POE3 = 0x9000_5000

Offset address: 0x1C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	OSTSTF	—	—	OSTSTE	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	—	These bits are read as 0. The write value should be 0.	R/W
9	OSTSTE	Oscillation Stop High-Impedance Enable 0: Does not place the MTU complementary PWM output pins and MTU0 pins in high-impedance state 1: Places the MTU complementary PWM output pins and MTU0 pins in high-impedance state	R/W*1
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	OSTSTF	Oscillation Stop High-Impedance Flag 0: Indicates that a high-impedance request or oscillation abnormality has not been generated. 1: Indicates that a high-impedance request by oscillation stop or oscillation abnormality has been generated.	R/W*2
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed method.

The ICSR6 register controls the high-impedance for stopped oscillation or oscillation abnormality and indicates the status.

OSTSTE bit (Oscillation Stop High-Impedance Enable)

This bit enables or disables the MTU complementary PWM output pins and MTU0 pins to be placed in the high-impedance state when stopped oscillation or oscillation abnormality of the PLL is detected.

OSTSTF flag (Oscillation Stop High-Impedance Flag)

This flag indicates that a stopped oscillation high-impedance request has been generated.

When stopped oscillation is detected, this flag is set to 1. When clearing this flag, write 0 while the detection signal for stopped oscillation or oscillation abnormality of the PLL is negated. Writing 0 to this flag while the detection signal for

stopped oscillation or oscillation abnormality is asserted does not clear this flag to 0. After clearing this flag, confirm that the flag has actually been modified to 0.

[Setting condition]

- When stopped oscillation is detected

[Clearing condition]

- By writing 0 to the OSTSTF flag after reading OSTSTF = 1

18.2.7 ICSR7 : Input Level Control/Status Register 7

Base address: POE3 = 0x9000_5000

Offset address: 0x1E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	D1ER R0ST	D0ER R0ST	D1ER R1ST	D0ER R1ST	—	—	—	D1ER R0IE	D0ER R0IE	D1ER R1IE	D0ER R1IE	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	D0ERR1IE	DSMIF0 Error 1 Interrupt enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
5	D1ERR1IE	DSMIF1 Error 1 Interrupt enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
6	D0ERR0IE	DSMIF0 Error 0 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
7	D1ERR0IE	DSMIF1 Error 0 Interrupt Enable 0: Interrupt is disabled 1: Interrupt is enabled	R/W
10:8	—	These bits are read as 0. The write value should be 0.	R/W
11	D0ERR1ST	DSMIF0 Error 1 Status 0: Indicates that DSMIF0 error 1 has not been generated 1: Indicates that DSMIF0 error 1 has been generated	R
12	D1ERR1ST	DSMIF1 Error 1 Status 0: Indicates that DSMIF1 error 1 has not been generated 1: Indicates that DSMIF1 error 1 has been generated	R
13	D0ERR0ST	DSMIF0 Error 0 Status 0: Indicates that DSMIF0 error 0 has not been generated 1: Indicates that DSMIF0 error 0 has been generated	R
14	D1ERR0ST	DSMIF1 Error 0 Status 0: Indicates that DSMIF1 error 0 has not been generated 1: Indicates that DSMIF1 error 0 has been generated	R
15	—	This bit is read as 0. The write value should be 0.	R/W

The ICSR7 register controls the high-impedance for DSMIF error and indicates the status.

D0ERR1IE bit (DSMIF0 Error 1 Interrupt enable)

This bit enables or disables the interrupt request when DSMIF0 error 1 interrupt is detected. The interrupt output is multiplexed with OEI4.

D1ERR1IE bit (DSMIF1 Error 1 Interrupt enable)

This bit enables or disables the interrupt request when DSMIF1 error 1 interrupt is detected. The interrupt output is multiplexed with OEI4.

D0ERR0IE bit (DSMIF0 Error 0 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF0 error 0 interrupt is detected. The interrupt output is multiplexed with OEI4.

D1ERR0IE bit (DSMIF1 Error 0 Interrupt Enable)

This bit enables or disables the interrupt request when DSMIF1 error 0 interrupt is detected. The interrupt output is multiplexed with OEI4.

D0ERR1ST bit (DSMIF0 Error 1 Status)

This bit indicates that DSMIF0 error 1 interrupt request has been generated.

When DSMIF0 error 1 is detected, this bit is set to 1.

D1ERR1ST bit (DSMIF1 Error 1 Status)

This bit indicates that DSMIF1 error 1 interrupt request has been generated.

When DSMIF1 error 1 is detected, this bit is set to 1.

D0ERR0ST bit (DSMIF0 Error 0 Status)

This bit indicates that a DSMIF0 error 0 interrupt request has been generated.

When DSMIF0 error 0 is detected, this bit is set to 1.

D1ERR0ST bit (DSMIF1 Error 0 Status)

This bit indicates that a DSMIF1 error 0 interrupt request has been generated.

When DSMIF1 error 0 is detected, this bit is set to 1.

18.2.8 OCSR1 : Output Level Control/Status Register 1

Base address: POE3 = 0x9000_5000

Offset address: 0x02

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OSF1	—	—	—	—	—	OCE1	OIE1	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	OIE1	Output Short Circuit Interrupt Enable 1 0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
9	OCE1	Output Short Circuit High-Impedance Enable 1 0: Does not place the pins in the high-impedance state 1: Places the pins in the high-impedance state	R/W*1
14:10	—	These bits are read as 0. The write value should be 0.	R/W
15	OSF1	Output Short Circuit Flag 1 0: Indicates that outputs have not simultaneously become an active level 1: Indicates that outputs have simultaneously become an active level	R/W*2

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed method.

The OCSR1 register controls the enable or disable of output-level comparison and interrupts and indicates status.

OIE1 bit (Output Short Circuit Interrupt Enable 1)

This bit enables or disables interrupt requests when the OSF1 flag is set to 1.

OCE1 bit (Output Short Circuit High-Impedance Enable 1)

This bit specifies whether to place the pins in high-impedance state when the OSF1 bit in OCSR1 is set to 1.

OSF1 flag (Output Short Circuit Flag 1)

This flag indicates that any one of the three pairs of 2-phase MTU3 and MTU4 pins for MTU complementary PWM output to be compared has simultaneously become an active level.

[Setting condition]

- When any one of the three pairs of 2-phase outputs has simultaneously become an active level

[Clearing condition]

- By writing 0 to the OSF1 flag after reading OSF1 = 1

18.2.9 OCSR2 : Output Level Control/Status Register 2

Base address: POE3 = 0x9000_5000

Offset address: 0x06

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OSF2	—	—	—	—	—	OCE2	OIE2	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	OIE2	Output Short Circuit Interrupt Enable 2 0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
9	OCE2	Output Short Circuit High-Impedance Enable 2 0: Does not place the pins in the high-impedance state 1: Places the pins in the high-impedance state	R/W*1
14:10	—	These bits are read as 0. The write value should be 0.	R/W
15	OSF2	Output Short Circuit Flag 2 0: Indicates that outputs have not simultaneously become an active level 1: Indicates that outputs have simultaneously become an active level	R/W*2

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed method.

The OCSR2 register controls the enable or disable of output-level comparison and interrupts and indicates status.

OIE2 bit (Output Short Circuit Interrupt Enable 2)

This bit enables or disables interrupt requests when the OSF2 flag in OCSR2 is set to 1.

OCE2 bit (Output Short Circuit High-Impedance Enable 2)

This bit specifies whether to place the pins in the high-impedance state when the OSF2 flag in OCSR2 is set to 1.

OSF2 flag (Output Short Circuit Flag 2)

This flag indicates that any one of the three pairs of 2-phase MTU6 and MTU7 pins for MTU complementary PWM output to be compared has simultaneously become an active level.

[Setting condition]

- When any one of the three pairs of 2-phase outputs has simultaneously become an active level

[Clearing condition]

- By writing 0 to the OSF2 flag after reading OSF2 = 1

18.2.10 ALR1 : Active Level Setting Register 1

Base address: POE3 = 0x9000_5000

Offset address: 0x1A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	OLSEN	—	OLSG2B	OLSG2A	OLSG1B	OLSG1A	OLSG0B	OLSG0A
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OLSG0A	MTIOC3B Pin Active Level Setting 0: Active-low 1: Active-high	R/W*1
1	OLSG0B	MTIOC3D Pin Active Level Setting 0: Active-low 1: Active-high	R/W*1
2	OLSG1A	MTIOC4A Pin Active Level Setting 0: Active-low 1: Active-high	R/W*1
3	OLSG1B	MTIOC4C Pin Active Level Setting 0: Active-low 1: Active-high	R/W*1
4	OLSG2A	MTIOC4B Pin Active Level Setting 0: Active-low 1: Active-high	R/W*1
5	OLSG2B	MTIOC4D Pin Active Level Setting 0: Active low 1: Active high	R/W*1
6	—	This bit is read as 0. The write value should be 0.	R/W
7	OLSEN	Active Level Setting Enable 0: Disabled 1: Enabled	R/W*1
15:8	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

The ALR1 register specifies the active levels of the MTU outputs selected by the I/O Ports, as the active levels for detection of short circuits of those outputs reflected in OCSR1.

OLSG0A bit (MTIOC3B Pin Active Level Setting)

This bit sets the active level of the MTIOC3B output. Specifically, setting the OLSG0A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSG0B bit (MTIOC3D Pin Active Level Setting)

This bit sets the active level of the MTIOC3D output. Specifically, setting the OLSG0B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSG1A bit (MTIOC4A Pin Active Level Setting)

This bit sets the active level of the MTIOC4A output. Specifically, setting the OLSG1A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSG1B bit (MTIOC4C Pin Active Level Setting)

This bit sets the active level of the MTIOC4C output. Specifically, setting the OLSG1B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSG2A bit (MTIOC4B Pin Active Level Setting)

This bit sets the active level of the MTIOC4B output. Specifically, setting the OLSG2A bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSG2B bit (MTIOC4D Pin Active Level Setting)

This bit sets the active level of the MTIOC4D output. Specifically, setting the OLSG2B bit to 0 sets the low level and to 1 sets the high level as the active level for detection of short circuits.

OLSEN bit (Active Level Setting Enable)

This bit enables or disables the active-level settings in the OLSGnm bits (n = 0 to 2; m = A, B). Clearing the OLSEN bit to 0 disables the OLSGnm bits, in which case the active levels of the MTU output are determined by the MTU.TOCR1A and MTU.TOCR2A registers. Setting the OLSEN bit to 1 enables the OLSGnm bits, in which case the active levels of the MTU output are as selected by the OLSGnm bits in this register.

18.2.11 SPOER : Software Port Output Enable Register

Base address: POE3 = 0x9000_5000

Offset address: 0x0A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	MTUCH0HIZ	MTUCH67HIZ	MTUCH34HIZ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MTUCH34HIZ	MTU3 and MTU4 Output High-Impedance Enable 0: Does not place the pins in the high-impedance state 1: Places the pins in the high-impedance state	R/W
1	MTUCH67HIZ	MTU6 and MTU7 Output High-Impedance Enable 0: Does not place the pins in the high-impedance state 1: Places the pins in the high-impedance state	R/W
2	MTUCH0HIZ	MTU0 Pin High-Impedance Enable 0: Does not place the pins in the high-impedance state 1: Places the pins in the high-impedance state	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The SPOER register controls the high-impedance state of the pins.

MTUCH34HIZ bit (MTU3 and MTU4 Output High-Impedance Enable)

This bit specifies whether to place the MTU complementary PWM output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D) in high-impedance state.

[Setting condition]

- By writing 1 to the MTUCH34HIZ bit

[Clearing conditions]

- Reset
- By writing 0 to the MTUCH34HIZ bit after reading MTUCH34HIZ = 1.
  To write 0 to this bit, be sure to read 1 first and then write 0.

MTUCH67HIZ bit (MTU6 and MTU7 Output High-Impedance Enable)

This bit specifies whether to place the MTU complementary PWM output pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7B, MTIOC7C, and MTIOC7D) in the high-impedance state.

[Setting condition]

- By writing 1 to the MTUCH67HIZ bit

[Clearing conditions]

- Reset
- By writing 0 to the MTUCH67HIZ bit after reading MTUCH67HIZ = 1.
  To write 0 to this bit, be sure to read 1 first and then write 0.

MTUCH0HIZ bit (MTU0 Pin High-Impedance Enable)

This bit specifies whether to place the MTU0 pins in the high-impedance state.

[Setting condition]

- By writing 1 to the MTUCH0HIZ bit

[Clearing conditions]

- Reset
- By writing 0 to the MTUCH0HIZ bit after reading MTUCH0HIZ = 1.
  To write 0 to this bit, be sure to read 1 first and then write 0.

18.2.12 POECR1 : Port Output Enable Control Register 1

Base address: POE3 = 0x9000_5000

Offset address: 0x0B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	MTU0 DZE	MTU0 CZE	MTU0 BZE	MTU0 AZE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MTU0AZE	MTIOC0A High-Impedance Enable 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state	R/W*1
1	MTU0BZE	MTIOC0B High-Impedance Enable 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state	R/W*1
2	MTU0CZE	MTIOC0C High-Impedance Enable 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state	R/W*1
3	MTU0DZE	MTIOC0D High-Impedance Enable 0: Does not place the pin in high-impedance state 1: Places the pin in high-impedance state	R/W*1
7:4	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

The POECR1 register controls high-impedance state of the MTU0 pins.

MTU0AZE bit (MTIOC0A High-Impedance Enable)

This bit specifies whether to place the MTIOC0A output for the MTU0 pin in the high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, and ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally

specified in the POECR5 register, the ICSRn.POEmF flag (n = 1, 2, 4, 5; m = 0, 4, 10, 11), ICSR7.DmERRnST (m = 0, 1; n = 0, 1), is set to 1.

MTU0BZE bit (MTIOC0B High-Impedance Enable)

This bit specifies whether to place the MTIOC0B output for the MTU0 pin in the high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, and ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR5 register, the ICSRn.POEmF flag (n = 1, 2, 4, 5; m = 0, 4, 10, 11), ICSR7.DmERRnST (m = 0, 1; n = 0, 1), is set to 1.

MTU0CZE bit (MTIOC0C High-Impedance Enable)

This bit specifies whether to place the MTIOC0C output for the MTU0 pin in the high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, and ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR5 register, the ICSRn.POEmF flag (n = 1, 2, 4, 5; m = 0, 4, 10, 11), ICSR7.DmERRnST (m = 0, 1; n = 0, 1), is set to 1.

MTU0DZE bit (MTIOC0D High-Impedance Enable)

This bit specifies whether to place the MTIOC0D output for the MTU0 pin in the high-impedance state when any of the ICSR3.POE8F flag, SPOER.MTUCH0HIZ bit, and ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR5 register, the ICSRn.POEmF flag (n = 1, 2, 4, 5; m = 0, 4, 10, 11), ICSR7.DmERRnST (m = 0, 1; n = 0, 1), is set to 1.

18.2.13 POECR2 : Port Output Enable Control Register 2

Base address: POE3 = 0x9000_5000

Offset address: 0x0C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	MTU3 BDZE	MTU4 ACZE	MTU4 BDZE	—	—	—	—	—	MTU6 BDZE	MTU7 ACZE	MTU7 BDZE
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

Bit	Symbol	Function	R/W
0	MTU7BDZE	MTIOC7B/MTIOC7D High-Impedance Enable 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state	R/W*1
1	MTU7ACZE	MTIOC7A/MTIOC7C High-Impedance Enable 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state	R/W*1
2	MTU6BDZE	MTIOC6B/MTIOC6D High-Impedance Enable 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state	R/W*1
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	MTU4BDZE	MTIOC4B/MTIOC4D High-Impedance Enable 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state	R/W*1
9	MTU4ACZE	MTIOC4A/MTIOC4C High-Impedance Enable 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state	R/W*1
10	MTU3BDZE	MTIOC3B/MTIOC3D High-Impedance Enable 0: Does not place the pins in high-impedance state 1: Places the pins in high-impedance state	R/W*1
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

The POECR2 register controls high-impedance state of the MTU complementary PWM output pins (MTU3, MTU4, MTU6, and MTU7 pins).

MTU7BDZE bit (MTIOC7B/MTIOC7D High-Impedance Enable)

This bit specifies whether to place the MTIOC7B output and MTIOC7D output for the MTU7 pin in the high-impedance state when any one of the OCSR2.OSF2 flag, ICSR2.POE4F flag, SPOER.MTUCH67HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 1, 3 to 5; m = 0, 8, 10, 11), ICSR7.DmERRnST (m = 0, 1; n = 0, 1), is set to 1.

MTU7ACZE bit (MTIOC7A/MTIOC7C High-Impedance Enable)

This bit specifies whether to place the MTIOC7A output and MTIOC7C output for the MTU7 pin in the high-impedance state when any one of the OCSR2.OSF2 flag, ICSR2.POE4F flag, SPOER.MTUCH67HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 1, 3 to 5; m = 0, 8, 10, 11), ICSR7.DmERRnST (m = 0, 1; n = 0, 1), is set to 1.

MTU6BDZE bit (MTIOC6B/MTIOC6D High-Impedance Enable)

This bit specifies whether to place the MTIOC6B output and MTIOC6D output for the MTU6 pin in the high-impedance state when any one of the OCSR2.OSF2 flag, ICSR2.POE4F flag, SPOER.MTUCH67HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 1, 3 to 5; m = 0, 8, 10, 11), ICSR7.DmERRnST (m = 0, 1; n = 0, 1), is set to 1.

MTU4BDZE bit (MTIOC4B/MTIOC4D High-Impedance Enable)

This bit specifies whether to place the MTIOC4B output and MTIOC4D output for the MTU4 pin in the high-impedance state when any one of the OCSR1.OSF1 flag, ICSR1.POE0F flag, SPOER.MTUCH34HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 2 to 5; m = 4, 8, 10, 11), ICSR7.DmERRnST (m = 0, 1; n = 0, 1), is set to 1.

MTU4ACZE bit (MTIOC4A/MTIOC4C High-Impedance Enable)

This bit specifies whether to place the MTIOC4A output and MTIOC4C output for the MTU4 pin in the high-impedance state when any one of the OCSR1.OSF1 flag, ICSR1.POE0F flag, SPOER.MTUCH34HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 2 to 5; m = 4, 8, 10, 11), ICSR7.DmERRnST (m = 0, 1; n = 0, 1), is set to 1.

MTU3BDZE bit (MTIOC3B/MTIOC3D High-Impedance Enable)

This bit specifies whether to place the MTIOC3B output and MTIOC3D output for the MTU3 pin in the high-impedance state when any one of the OCSR1.OSF1 flag, ICSR1.POE0F flag, SPOER.MTUCH34HIZ bit, ICSR6.OSTSTF flag (when the OSTSTE bit is 1), or, as additionally specified in the POECR4 register, the ICSRn.POEmF flag (n = 2 to 5; m = 4, 8, 10, 11), ICSR7.DmERRnST (m = 0, 1; n = 0, 1), is set to 1.

18.2.14 POECR4 : Port Output Enable Control Register 4

Base address: POE3 = 0x9000_5000

Offset address: 0x10

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	D1E0A DDMT 67ZE	D0E0A DDMT 67ZE	IC5AD DMT6 7ZE	IC4AD DMT6 7ZE	IC3AD DMT6 7ZE	D1E1A DDMT 67ZE	IC1AD DMT6 7ZE	D0E1A DDMT 67ZE	D1E0A DDMT 34ZE	D0E0A DDMT 34ZE	IC5AD DMT3 4ZE	IC4AD DMT3 4ZE	IC3AD DMT3 4ZE	IC2AD DMT3 4ZE	D1E1A DDMT 34ZE	D0E1A DDMT 34ZE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	D0E1ADDMT34ZE	MTU3 and MTU4 High-Impedance D0ERR1ST Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
1	D1E1ADDMT34ZE	MTU3 and MTU4 High-Impedance D1ERR1ST Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
2	IC2ADDMT34ZE	MTU3 and MTU4 High-Impedance POE4F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1

Bit	Symbol	Function	R/W
3	IC3ADDMT34ZE	MTU3 and MTU4 High-Impedance POE8F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
4	IC4ADDMT34ZE	MTU3 and MTU4 High-Impedance POE10F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
5	IC5ADDMT34ZE	MTU3 and MTU4 High-Impedance POE11F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
6	D0E0ADDMT34ZE	MTU3 and MTU4 High-Impedance D0ERR0ST Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
7	D1E0ADDMT34ZE	MTU3 and MTU4 High-Impedance D1ERR0ST Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
8	D0E1ADDMT67ZE	MTU6 and MTU7 High-Impedance D0ERR1ST Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
9	IC1ADDMT67ZE	MTU6 and MTU7 High-Impedance POE0F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
10	D1E1ADDMT67ZE	MTU6 and MTU7 High-Impedance D1ERR1ST Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
11	IC3ADDMT67ZE	MTU6 and MTU7 High-Impedance POE8F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
12	IC4ADDMT67ZE	MTU6 and MTU7 High-Impedance POE10F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
13	IC5ADDMT67ZE	MTU6 and MTU7 High-Impedance POE11F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
14	D0E0ADDMT67ZE	MTU6 and MTU7 High-Impedance D0ERR0ST Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
15	D1E0ADDMT67ZE	MTU6 and MTU7 High-Impedance D1ERR0ST Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

The POECR4 register is used to extend the control conditions of the high-impedance state for the MTU complementary PWM output pins (MTU3, MTU4, MTU6, and MTU7).

For details about the targets and conditions of high-impedance control, see Figure 18.2.

D0E1ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D0ERR1ST Add)

Adds the ICSR7.D0ERR1ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D1E1ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D1ERR1ST Add)

Adds the ICSR7.D1ERR1ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

IC2ADDMT34ZE bit (MTU3 and MTU4 High-Impedance POE4F Add)

Adds the ICSR2.POE4F flag (POE4#) to the high-impedance control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

IC3ADDMT34ZE bit (MTU3 and MTU4 High-Impedance POE8F Add)

Adds the ICSR3.POE8F flag (POE8#) to the high-impedance control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

IC4ADDMT34ZE bit (MTU3 and MTU4 High-Impedance POE10F Add)

Adds the ICSR4.POE10F flag (POE10#) to the high-impedance control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

IC5ADDMT34ZE bit (MTU3 and MTU4 High-Impedance POE11F Add)

Adds the ICSR5.POE11F flag (POE11#) to the high-impedance control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D0E0ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D0ERR0ST Add)

Adds the ICSR7.D0ERR0ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D1E0ADDMT34ZE bit (MTU3 and MTU4 High-Impedance D1ERR0ST Add)

Adds the ICSR7.D1ERR0ST to the high-impedance control conditions for MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

D0E1ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D0ERR1ST Add)

Adds the ICSR7.D0ERR1ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

IC1ADDMT67ZE bit (MTU6 and MTU7 High-Impedance POE0F Add)

Adds the ICSR1.POE0F flag (POE0#) to the high-impedance control conditions for the MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D1E1ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D1ERR1ST Add)

Adds the ICSR7.D1ERR1ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

IC3ADDMT67ZE bit (MTU6 and MTU7 High-Impedance POE8F Add)

Adds the ICSR3.POE8F flag (POE8#) to the high-impedance control conditions for the MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

IC4ADDMT67ZE bit (MTU6 and MTU7 High-Impedance POE10F Add)

Adds the ICSR4.POE10F flag (POE10#) to the high-impedance control conditions for the MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

IC5ADDMT67ZE bit (MTU6 and MTU7 High-Impedance POE11F Add)

Adds the ICSR5.POE11F flag (POE11#) to the high-impedance control conditions for the MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D0E0ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D0ERR0ST Add)

Adds the ICSR7.D0ERR0ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

D1E0ADDMT67ZE bit (MTU6 and MTU7 High-Impedance D1ERR0ST Add)

Adds the ICSR7.D1ERR0ST to the high-impedance control conditions for MTU6 and MTU7 pins (MTIOC6B, MTIOC6D, MTIOC7A, MTIOC7C, MTIOC7B, MTIOC7D).

18.2.15 POECR5 : Port Output Enable Control Register 5

Base address: POE3 = 0x9000_5000

Offset address: 0x12

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	D1E0ADDMT0ZE	D0E0ADDMT0ZE	IC5ADDMT0ZE	IC4ADDMT0ZE	D1E1ADDMT0ZE	IC2ADDMT0ZE	IC1ADDMT0ZE	D0E1ADDMT0ZE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	D0E1ADDMT0ZE	MTU0 High-Impedance D0ERR1ST Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
1	IC1ADDMT0ZE	MTU0 High-Impedance POE0F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
2	IC2ADDMT0ZE	MTU0 High-Impedance POE4F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
3	D1E1ADDMT0ZE	MTU0 High-Impedance D1ERR1ST Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
4	IC4ADDMT0ZE	MTU0 High-Impedance POE10F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
5	IC5ADDMT0ZE	MTU0 High-Impedance POE11F Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
6	D0E0ADDMT0ZE	MTU0 High-Impedance D0ERR0ST Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
7	D1E0ADDMT0ZE	MTU0 High-Impedance D1ERR0ST Add 0: Does not add the flag to the high-impedance control conditions 1: Adds the flag to the high-impedance control conditions	R/W*1
15:8	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset. After the bit is modified once, writing this bit does not modify its value until the register is reset.

The POECR5 register is used to extend the control conditions to place the output of the MTU0 pins in the high-impedance state.

D0E1ADDMT0ZE bit (MTU0 High-Impedance D0ERR1ST Add)

Adds the ICSR7.D0ERR1ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC1ADDMT0ZE bit (MTU0 High-Impedance POE0F Add)

Adds the ICSR1.POE0F flag (POE0#) to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC2ADDMT0ZE bit (MTU0 High-Impedance POE4F Add)

Adds the ICSR2.POE4F flag (POE4#) to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

D1E1ADDMT0ZE bit (MTU0 High-Impedance D1ERR1ST Add)

Adds the ICSR7.D1ERR1ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC4ADDMT0ZE bit (MTU0 High-Impedance POE10F Add)

Adds the ICSR4.POE10F flag (POE10#) to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC5ADDMT0ZE bit (MTU0 High-Impedance POE11F Add)

Adds the ICSR5.POE11F flag (POE11#) to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

D0E0ADDMT0ZE bit (MTU0 High-Impedance D0ERR0ST Add)

Adds the ICSR7.D0ERR0ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

D1E0ADDMT0ZE bit (MTU0 High-Impedance D1ERR0ST Add)

Adds the ICSR7.D1ERR0ST to the high-impedance control conditions for MTU0 pins (MTIOC0A, MTIOC0C, MTIOC0B, MTIOC0D).

18.2.16 M0SELR1 : MTU0 Pin Select Register 1

Base address: POE3 = 0x9000_5000

Offset address: 0x24

Bit position:	7	6	5	4	3	2	1	0
Bit field:	M0BSEL[3:0]				M0ASEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	M0ASEL[3:0]	MTU0-A (MTIOC0A) Pin Select 0x0: Controls the high-impedance state of P13_2 when it is in use as the MTIOC0A pin 0x1: Controls the high-impedance state of P14_3 when it is in use as the MTIOC0A pin 0x2: Controls the high-impedance state of P23_7 when it is in use as the MTIOC0A pin Others: Setting prohibited.	R/W*1
7:4	M0BSEL[3:0]	MTU0-B (MTIOC0B) Pin Select 0x0: Setting prohibited. 0x1: Controls the high-impedance state of P14_4 when it is in use as the MTIOC0B pin 0x2: Controls the high-impedance state of P24_0 when it is in use as the MTIOC0B pin 0x3: Controls the high-impedance state of P13_3 when it is in use as the MTIOC0B pin Others: Setting prohibited.	R/W*1

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.

The M0SELR1 register is an 8-bit readable/writable register that selects the MTU0-A/B pins as targets for high-impedance control.

M0ASEL[3:0] bits (MTU0-A (MTIOC0A) Pin Select)

These bits select the target MTIOC0A pin for high-impedance control.

M0BSEL[3:0] bits (MTU0-B (MTIOC0B) Pin Select)

These bits select the target MTIOC0B pin for high-impedance control.

18.2.17 M0SELR2 : MTU0 Pin Select Register 2

Base address: POE3 = 0x9000_5000

Offset address: 0x25

Bit position:	7	6	5	4	3	2	1	0
Bit field:	M0DSEL[3:0]				M0CSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	M0CSEL[3:0]	MTU0-C (MTIOC0C) Pin Select 0x0: Controls the high-impedance state of P13_3 when it is in use as the MTIOC0C pin 0x1: Controls the high-impedance state of P15_1 when it is in use as the MTIOC0C pin 0x2: Controls the high-impedance state of P24_1 when it is in use as the MTIOC0C pin Others: Setting prohibited.	R/W*1
7:4	M0DSEL[3:0]	MTU0-D (MTIOC0D) Pin Select 0x0: Controls the high-impedance state of P13_4 when it is in use as the MTIOC0D pin 0x1: Controls the high-impedance state of P15_2 when it is in use as the MTIOC0D pin 0x2: Controls the high-impedance state of P24_2 when it is in use as the MTIOC0D pin Others: Setting prohibited.	R/W*1

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.

The M0SELR2 register is an 8-bit readable/writable register that selects the MTU0-C/D pins as targets for high-impedance control.

M0CSEL[3:0] bits (MTU0-C (MTIOC0C) Pin Select)

These bits select the target MTIOC0C pin for high-impedance control.

M0DSEL[3:0] bits (MTU0-D (MTIOC0D) Pin Select)

These bits select the target MTIOC0D pin for high-impedance control.

18.2.18 M3SELR : MTU3 Pin Select Register

Base address: POE3 = 0x9000_5000

Offset address: 0x26

Bit position:	7	6	5	4	3	2	1	0
Bit field:	M3DSEL[3:0]				M3BSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	M3BSEL[3:0]	MTU3-B (MTIOC3B) Pin Select 0x0: Controls the high-impedance state of P17_6 when it is in use as the MTIOC3B pin 0x1: Controls the high-impedance state of P00_6 when it is in use as the MTIOC3B pin 0x2: Controls the high-impedance state of P08_6 when it is in use as the MTIOC3B pin Others: Setting prohibited.	R/W*1
7:4	M3DSEL[3:0]	MTU3-D (MTIOC3D) Pin Select 0x0: Controls the high-impedance state of P18_1 when it is in use as the MTIOC3D pin 0x1: Controls the high-impedance state of P01_1 when it is in use as the MTIOC3D pin 0x2: Controls the high-impedance state of P08_7 when it is in use as the MTIOC3D pin Others: Setting prohibited.	R/W*1

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.

The M3SELR register is an 8-bit readable/writable register that selects the MTU3-B/D pins as targets for high-impedance control.

M3BSEL[3:0] bits (MTU3-B (MTIOC3B) Pin Select)

These bits select the target MTIOC3B pin for high-impedance control.

M3DSEL[3:0] bits (MTU3-D (MTIOC3D) Pin Select)

These bits select the target MTIOC3D pin for high-impedance control.

18.2.19 M4SELR1 : MTU4 Pin Select Register 1

Base address: POE3 = 0x9000_5000

Offset address: 0x27

Bit position:	7	6	5	4	3	2	1	0
Bit field:	M4CSEL[3:0]				M4ASEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	M4ASEL[3:0]	MTU4-A (MTIOC4A) Pin Select 0x0: Controls the high-impedance state of P17_7 when it is in use as the MTIOC4A pin 0x1: Controls the high-impedance state of P01_5 when it is in use as the MTIOC4A pin 0x2: Controls the high-impedance state of P09_0 when it is in use as the MTIOC4A pin 0x3: Controls the high-impedance state of P18_0 when it is in use as the MTIOC4A pin Others: Setting prohibited.	R/W*1
7:4	M4CSEL[3:0]	MTU4-C (MTIOC4C) Pin Select 0x0: Controls the high-impedance state of P18_0 when it is in use as the MTIOC4C pin 0x1: Controls the high-impedance state of P01_0 when it is in use as the MTIOC4C pin 0x2: Controls the high-impedance state of P09_1 when it is in use as the MTIOC4C pin 0x3: Controls the high-impedance state of P17_7 when it is in use as the MTIOC4C pin Others: Setting prohibited.	R/W*1

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.

The M4SELR1 register is an 8-bit readable/writable register that selects the MTU4-A/C pins as targets for high-impedance control.

M4ASEL[3:0] bits (MTU4-A (MTIOC4A) Pin Select)

These bits select the target MTIOC4A pin for high-impedance control.

M4CSEL[3:0] bits (MTU4-C (MTIOC4C) Pin Select)

These bits select the target MTIOC4C pin for high-impedance control.

18.2.20 M4SELR2 : MTU4 Pin Select Register 2

Base address: POE3 = 0x9000_5000

Offset address: 0x28

Bit position:	7	6	5	4	3	2	1	0
Bit field:	M4DSEL[3:0]				M4BSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	M4BSEL[3:0]	MTU4-B (MTIOC4B) Pin Select 0x0: Controls the high-impedance state of P18_2 when it is in use as the MTIOC4B pin 0x1: Controls the high-impedance state of P01_2 when it is in use as the MTIOC4B pin 0x2: Controls the high-impedance state of P09_2 when it is in use as the MTIOC4B pin 0x3: Controls the high-impedance state of P18_3 when it is in use as the MTIOC4B pin Others: Setting prohibited.	R/W*1

Bit	Symbol	Function	R/W
7:4	M4DSEL[3:0]	MTU4-D (MTIOC4D) Pin Select 0x0: Controls the high-impedance state of P18_3 when it is in use as the MTIOC4D pin 0x1: Controls the high-impedance state of P01_3 when it is in use as the MTIOC4D pin 0x2: Controls the high-impedance state of P09_3 when it is in use as the MTIOC4D pin 0x3: Controls the high-impedance state of P18_2 when it is in use as the MTIOC4D pin Others: Setting prohibited.	R/W*1

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.

The M4SELR2 register is an 8-bit readable/writable register that selects the MTU4-B/D pins as targets for high-impedance control.

M4BSEL[3:0] bits (MTU4-B (MTIOC4B) Pin Select)

These bits select the target MTIOC4B pin for high-impedance control.

M4DSEL[3:0] bits (MTU4-D (MTIOC4D) Pin Select)

These bits select the target MTIOC4D pin for high-impedance control.

18.2.21 M6SELR : MTU6 Pin Select Register

Base address: POE3 = 0x9000_5000

Offset address: 0x2A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	M6DSEL[3:0]				M6BSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	M6BSEL[3:0]	MTU6-B (MTIOC6B) Pin Select 0x0: Setting prohibited. 0x1: Controls the high-impedance state of P21_2 when it is in use as the MTIOC6B pin 0x2: Controls the high-impedance state of P08_5 when it is in use as the MTIOC6B pin Others: Setting prohibited.	R/W*1
7:4	M6DSEL[3:0]	MTU6-D (MTIOC6D) Pin Select 0x0: Setting prohibited. 0x1: Controls the high-impedance state of P21_4 when it is in use as the MTIOC6D pin 0x2: Controls the high-impedance state of P08_7 when it is in use as the MTIOC6D pin Others: Setting prohibited.	R/W*1

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.

The M6SELR register is an 8-bit readable/writable register that selects the MTU6-B/D pins as targets for high-impedance control.

M6BSEL[3:0] bits (MTU6-B (MTIOC6B) Pin Select)

These bits select the target MTIOC6B pin for high-impedance control.

M6DSEL[3:0] bits (MTU6-D (MTIOC6D) Pin Select)

These bits select the target MTIOC6D pin for high-impedance control.

18.2.22 M7SELR1 : MTU7 Pin Select Register 1

Base address: POE3 = 0x9000_5000

Offset address: 0x2B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	M7CSEL[3:0]				M7ASEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	M7ASEL[3:0]	MTU7-A (MTIOC7A) Pin Select 0x0: Setting prohibited. 0x1: Controls the high-impedance state of P21_5 when it is in use as the MTIOC7A pin 0x2: Controls the high-impedance state of P09_0 when it is in use as the MTIOC7A pin Others: Setting prohibited.	R/W*1
7:4	M7CSEL[3:0]	MTU7-C (MTIOC7C) Pin Select 0x0: Setting prohibited. 0x1: Controls the high-impedance state of P21_7 when it is in use as the MTIOC7C pin 0x2: Controls the high-impedance state of P09_2 when it is in use as the MTIOC7C pin Others: Setting prohibited.	R/W*1

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.

The M7SELR1 register is an 8-bit readable/writable register that selects the MTU7-A/C pins as targets for high-impedance control.

M7ASEL[3:0] bits (MTU7-A (MTIOC7A) Pin Select)

The M7ASEL[3:0] bits select the target MTIOC7A pin for high-impedance control.

M7CSEL[3:0] bits (MTU7-C (MTIOC7C) Pin Select)

The M7CSEL[3:0] bits select the target MTIOC7C pin for high-impedance control.

18.2.23 M7SELR2 : MTU7 Pin Select Register 2

Base address: POE3 = 0x9000_5000

Offset address: 0x2C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	M7DSEL[3:0]				M7BSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	M7BSEL[3:0]	MTU7-B (MTIOC7B) Pin Select 0x0: Setting prohibited. 0x1: Controls the high-impedance state of P21_6 when it is in use as the MTIOC7B pin 0x2: Controls the high-impedance state of P09_1 when it is in use as the MTIOC7B pin Others: Setting prohibited.	R/W*1
7:4	M7DSEL[3:0]	MTU7-D (MTIOC7D) Pin Select 0x0: Setting prohibited. 0x1: Controls the high-impedance state of P22_0 when it is in use as the MTIOC7D pin 0x2: Controls the high-impedance state of P09_3 when it is in use as the MTIOC7D pin Others: Setting prohibited.	R/W*1

Note 1. Can be modified only once after a reset. After the bits are modified once, writing these bits does not modify their values until the register is reset.

The M7SELR2 register is an 8-bit readable/writable register that selects the MTU7-B/D pins as targets for high-impedance control.

M7BSEL[3:0] bits (MTU7-B (MTIOC7B) Pin Select)

The M7BSEL[3:0] bits select the target MTIOC7B pin for high-impedance control.

M7DSEL[3:0] bits (MTU7-D (MTIOC7D) Pin Select)

The M7DSEL[3:0] bits select the target MTIOC7D pin for high-impedance control.

18.3 Operation

Table 18.4 shows the target pins for high-impedance control and conditions to place the pins in high-impedance state.

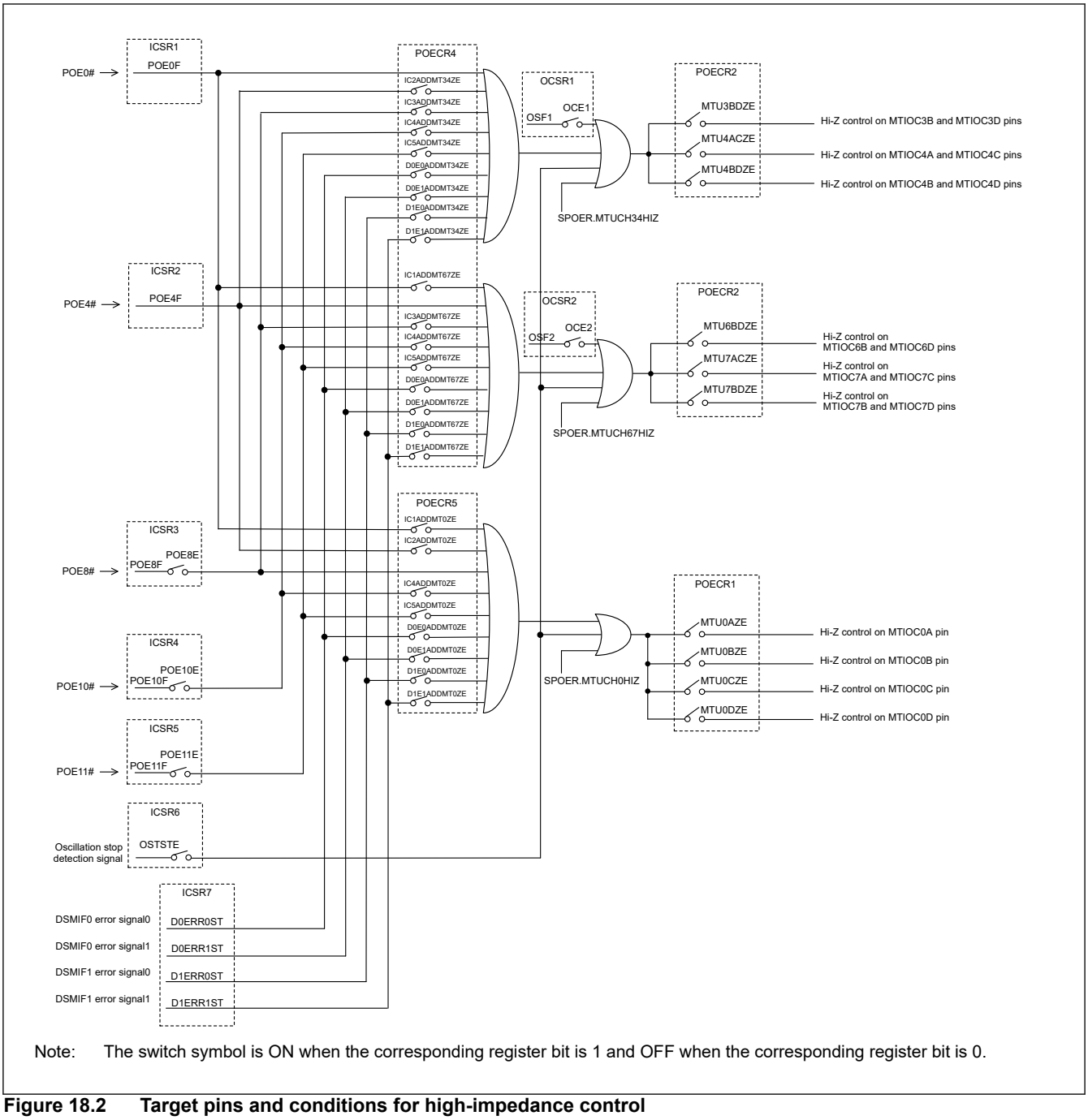
Table 18.4 Target pins and conditions for high-impedance control (1 of 2)

Pins	Conditions	Detailed conditions
MTU3 pins (MTIOC3B and MTIOC3D)	• Operation for detection of the POE0# input level • Operation for comparison of the output levels on the MTIOC3B and MTIOC3D pins • SPOER setting • Additional conditions of the POECR4 • Detection of stopped oscillation or oscillation abnormality • DSMIF error interrupt	MTU3BDZE • ((POE0F) + (OSF1 • OCE1) + (MTUCH34HIZ) + (IC2ADDMT34ZE • POE4F) + (IC3ADDMT34ZE • POE8E • POE8F) + (IC4ADDMT34ZE • POE10E • POE10F) + (IC5ADDMT34ZE • POE11E • POE11F) + (OSTSTE • OSTSTF) + (D0E1ADDMT34ZE•D0ERR1ST) + (D1E1ADDMT34ZE•D1ERR1ST) + (D0E0ADDMT34ZE•D0ERR0ST) + (D1E0ADDMT34ZE•D1ERR0ST))
MTU4 pins (MTIOC4A and MTIOC4C)	• Operation for detection of the POE0# input level • Operation for comparison of the output levels on the MTIOC4A and MTIOC4C pins • SPOER setting • Additional conditions of the POECR4 • Detection of stopped oscillation or oscillation abnormality • DSMIF error interrupt	MTU4ACZE • ((POE0F) + (OSF1 • OCE1) + (MTUCH34HIZ) + (IC2ADDMT34ZE • POE4F) + (IC3ADDMT34ZE • POE8E • POE8F) + (IC4ADDMT34ZE • POE10E • POE10F) + (IC5ADDMT34ZE • POE11E • POE11F) + (OSTSTE • OSTSTF) + (D0E1ADDMT34ZE•D0ERR1ST) + (D1E1ADDMT34ZE•D1ERR1ST) + (D0E0ADDMT34ZE•D0ERR0ST) + (D1E0ADDMT34ZE•D1ERR0ST))
MTU4 pins (MTIOC4B and MTIOC4D)	• Operation for detection of the POE0# input level • Operation for comparison of the output levels on the MTIOC4B and MTIOC4D pins • SPOER setting • Additional conditions of the POECR4 • Detection of stopped oscillation or oscillation abnormality • DSMIF error interrupt	MTU4BDZE • ((POE0F) + (OSF1 • OCE1) + (MTUCH34HIZ) + (IC2ADDMT34ZE • POE4F) + (IC3ADDMT34ZE • POE8E • POE8F) + (IC4ADDMT34ZE • POE10E • POE10F) + (IC5ADDMT34ZE • POE11E • POE11F) + (OSTSTE • OSTSTF) + (D0E1ADDMT34ZE•D0ERR1ST) + (D1E1ADDMT34ZE•D1ERR1ST) + (D0E0ADDMT34ZE•D0ERR0ST) + (D1E0ADDMT34ZE•D1ERR0ST))
MTU6 pins (MTIOC6B and MTIOC6D)	• Operation for detection of the POE4# input level • Operation for comparison of the output levels on the MTIOC6B and MTIOC6D pins • SPOER setting • Additional conditions of the POECR4 • Detection of stopped oscillation or oscillation abnormality • DSMIF error interrupt	MTU6BDZE • ((POE4F) + (OSF2 • OCE2) + (MTUCH67HIZ) + (IC1ADDMT67ZE • POE0F) + (IC3ADDMT67ZE • POE8E • POE8F) + (IC4ADDMT67ZE • POE10E • POE10F) + (IC5ADDMT67ZE • POE11E • POE11F) + (OSTSTE • OSTSTF) + (D0E1ADDMT67ZE•D0ERR1ST) + (D1E1ADDMT67ZE•D1ERR1ST) + (D0E0ADDMT67ZE•D0ERR0ST) + (D1E0ADDMT67ZE•D1ERR0ST))
MTU7 pins (MTIOC7A and MTIOC7C)	• Operation for detection of the POE4# input level • Operation for comparison of the output levels on the MTIOC7A and MTIOC7C pins • SPOER setting • Additional conditions of the POECR4 • Detection of stopped oscillation or oscillation abnormality • DSMIF error interrupt	MTU7ACZE • ((POE4F) + (OSF2 • OCE2) + (MTUCH67HIZ) + (IC1ADDMT67ZE • POE0F) + (IC3ADDMT67ZE • POE8E • POE8F) + (IC4ADDMT67ZE • POE10E • POE10F) + (IC5ADDMT67ZE • POE11E • POE11F) + (OSTSTE • OSTSTF) + (D0E1ADDMT67ZE•D0ERR1ST) + (D1E1ADDMT67ZE•D1ERR1ST) + (D0E0ADDMT67ZE•D0ERR0ST) + (D1E0ADDMT67ZE•D1ERR0ST))

Table 18.4 Target pins and conditions for high-impedance control (2 of 2)

Pins	Conditions	Detailed conditions
MTU7 pins (MTIOC7B and MTIOC7D)	• Operation for detection of the POE4# input level • Operation for comparison of the output levels on the MTIOC7B and MTIOC7D pins • SPOER setting • Additional conditions of the POECR4 • Detection of stopped oscillation or oscillation abnormality • DSMIF error interrupt	MTU7BDZE • ((POE4F) + (OSF2 • OCE2) + (MTUCH67HIZ) + (IC1ADDMT67ZE • POE4F) + (IC3ADDMT67ZE • POE8E • POE8F) + (IC4ADDMT67ZE • POE10E • POE10F) + (IC5ADDMT67ZE • POE11E • POE11F) + (OSTSTE • OSTSTF) + (D0E1ADDMT67ZE•D0ERR1ST) + (D1E1ADDMT67ZE•D1ERR1ST) + (D0E0ADDMT67ZE•D0ERR0ST) + (D1E0ADDMT67ZE•D1ERR0ST))
MTU0 pin (MTIOC0A)	• Operation for detection of the POE8# input level • SPOER setting • Additional conditions of the POECR5 • Detection of stopped oscillation or oscillation abnormality*1 • DSMIF error interrupt	MTU0AZE • ((POE8F • POE8E) + (MTUCH0HIZ) + (IC1ADDMT0ZE • POE0F) + (IC2ADDMT0ZE • POE4F) + (IC4ADDMT0ZE • POE10E • POE10F) + (IC5ADDMT0ZE • POE11E • POE11F) + (OSTSTE • OSTSTF) + (D0E1ADDMT0ZE•D0ERR1ST) + (D1E1ADDMT0ZE•D1ERR1ST) + (D0E0ADDMT0ZE•D0ERR0ST) + (D1E0ADDMT0ZE•D1ERR0ST))
MTU0 pin (MTIOC0B)	• Operation for detection of the POE8# input level • SPOER setting • Additional conditions of the POECR5 • Detection of stopped oscillation or oscillation abnormality*1 • DSMIF error interrupt	MTU0BZE • ((POE8F • POE8E) + (MTUCH0HIZ) + (IC1ADDMT0ZE • POE0F) + (IC2ADDMT0ZE • POE4F) + (IC4ADDMT0ZE • POE10E • POE10F) + (IC5ADDMT0ZE • POE11E • POE11F) + (OSTSTE • OSTSTF) + (D0E1ADDMT0ZE•D0ERR1ST) + (D1E1ADDMT0ZE•D1ERR1ST) + (D0E0ADDMT0ZE•D0ERR0ST) + (D1E0ADDMT0ZE•D1ERR0ST))
MTU0 pin (MTIOC0C)	• Operation for detection of the POE8# input level • SPOER setting • Additional conditions of the POECR5 • Detection of stopped oscillation or oscillation abnormality*1 • DSMIF error interrupt	MTU0CZE • ((POE8F • POE8E) + (MTUCH0HIZ) + (IC1ADDMT0ZE • POE0F) + (IC2ADDMT0ZE • POE4F) + (IC4ADDMT0ZE • POE10E • POE10F) + (IC5ADDMT0ZE • POE11E • POE11F) + (OSTSTE • OSTSTF) + (D0E1ADDMT0ZE•D0ERR1ST) + (D1E1ADDMT0ZE•D1ERR1ST) + (D0E0ADDMT0ZE•D0ERR0ST) + (D1E0ADDMT0ZE•D1ERR0ST))
MTU0 pin (MTIOC0D)	• Operation for detection of the POE8# input level • SPOER setting • Additional conditions of the POECR5 • Detection of stopped oscillation or oscillation abnormality*1 • DSMIF error interrupt	MTU0DZE • ((POE8F • POE8E) + (MTUCH0HIZ) + (IC1ADDMT0ZE • POE0F) + (IC2ADDMT0ZE • POE4F) + (IC4ADDMT0ZE • POE10E • POE10F) + (IC5ADDMT0ZE • POE11E • POE11F) + (OSTSTE • OSTSTF) + (D0E1ADDMT0ZE•D0ERR1ST) + (D1E1ADDMT0ZE•D1ERR1ST) + (D0E0ADDMT0ZE•D0ERR0ST) + (D1E0ADDMT0ZE•D1ERR0ST))

Note 1. The clock monitor circuit (CLMA3) can detect the oscillation abnormality.



Note: The switch symbol is ON when the corresponding register bit is 1 and OFF when the corresponding register bit is 0.

Figure 18.2 Target pins and conditions for high-impedance control

High-impedance requests to individual pins can be controlled by the settings in the POECR1 to POECR2 registers.

The following input pins can be added to high-impedance control conditions by the settings in the POECR4 to POECR5 registers:

- Input pins other than the POE0# pin for the MTU3 and MTU4 pins
- Input pins other than the POE4# pin for the MTU6 and MTU7 pins
- Input pins other than the POE8# pin for the MTU0 pins

For example, setting the IC2ADDMT34ZE bit in POECR4 to 1 outputs high-impedance requests to the MTU3 and MTU4 pins even when the POE4# input is detected.

High-impedance requests due to the detection of POE8# input, POE10# input, POE11# input, stopped oscillation, or oscillation abnormality can be controlled by the settings in the ICSR3 to ICSR6 registers. The ICSR1 and ICSR2 registers do not control enabling or disabling of output of high-impedance requests.

High-impedance requests to the MTU3, MTU4, MTU6, and MTU7 pins as the results of output level comparison can be controlled by the settings in the OCSR1 and OCSR2 registers.

18.3.1 MTU3 Pin Selection

In RZ/T2, the pin functions for MTU are respectively assigned to multiple sets of port pins. The target pins for high-impedance control can be selected by the pin select register in POE3 (M0SELR1, M0SELR2, M3SELR, M4SELR1, M4SELR2, M6SELR, M7SELR1, or M7SELR7 register). Table 18.5 shows the correspondence between MTU pins and select registers.

Note: Settings for pins to be used as MTU must be separately made in the registers of the I/O Ports. Take care so that there are no differences between the pins selected in the POE3 registers and the pins selected in the I/O port registers. For details of I/O port registers, see section 16, I/O Ports.

Table 18.5 Correspondence between MTU

MTU pin functions	Corresponding ports	Select registers
MTIOC0A	P13_2, P14_3, P23_7	M0SELR1
MTIOC0B	P14_4, P24_0, P13_3	
MTIOC0C	P13_3, P15_1, P24_1	M0SELR2
MTIOC0D	P13_4, P15_2, P24_2	
MTIOC3B	P17_6, P00_6, P08_6	M3SELR
MTIOC3D	P18_1, P01_1, P08_7	
MTIOC4A	P17_7, P01_5, P09_0, P18_0	M4SELR1
MTIOC4C	P18_0, P01_0, P09_1, P17_7	
MTIOC4B	P18_2, P01_2, P09_2, P18_3	M4SELR2
MTIOC4D	P18_3, P01_3, P09_3, P18_2	
MTIOC6B	P21_2, P08_5	M6SELR
MTIOC6D	P21_4, P08_7	
MTIOC7A	P21_5, P09_0	M7SELR1
MTIOC7C	P21_7, P09_2	
MTIOC7B	P21_6, P09_1	M7SELR2
MTIOC7D	P22_0, P09_3	

18.3.2 Input-Level Detection Operation

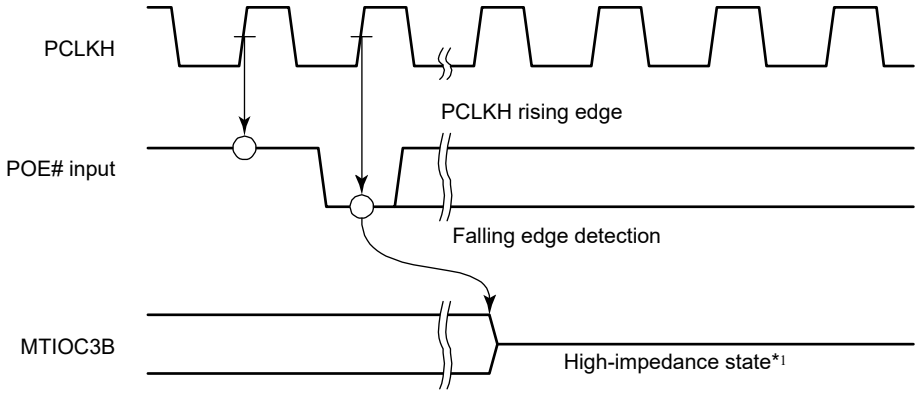
If the input conditions set by ICSR1 to ICSR5 occur on the POE0#, POE4#, POE8#, POE10#, and POE11# pins, the MTU3 and MTU4 pins for the MTU complementary PWM output, and MTU0 pins are placed in high-impedance state.

Note: These pins are still placed in the high-impedance state even when the MTU3 function is not selected for the pins on which they are multiplexed.

(1) Falling Edge Detection

When a change from a high to low level is input to the POE0#, POE4#, POE8#, POE10#, and POE11# pins, the pin functions which are multiplexed with the pins for the MTU complementary PWM output, and MTU0 pins are placed in high-impedance state.

Figure 18.3 shows a sample timing after the level changes in input to the POE0#, POE4#, POE8#, POE10#, and POE11# pins until the respective pins enter the high-impedance state.



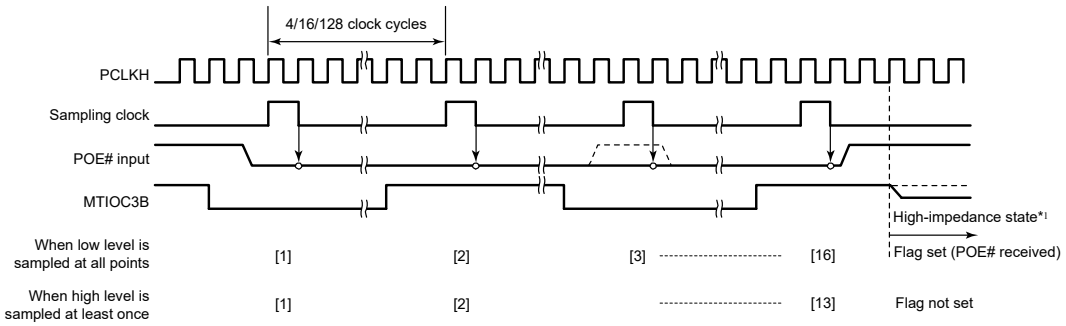
Note 1. Other pins also enter the high-impedance state at the same timing.

Figure 18.3 Falling edge detection

(2) Low-Level Detection

Figure 18.4 shows the low-level detection operation. When the POE# input has been sampled 16 times at the clock pulses selected by ICSR1 to ICSR5 and all are low level, the request is accepted. If even one high level is detected during this interval, the request is not accepted.

The timing when pins for the MTU complementary PWM output, and MTU0 pins enter the high-impedance state after the sampling clock is input is the same in both falling-edge detection and in low-level detection.



Note 1. Other pins also enter the high-impedance state at the same timing.

Figure 18.4 Low-level detection operation

18.3.3 Output-Level Compare Operation

Figure 18.5 shows an example of the output-level compare operation for the combination of MTIOC3B and MTIOC3D. The operation is the same for the other pin combinations.

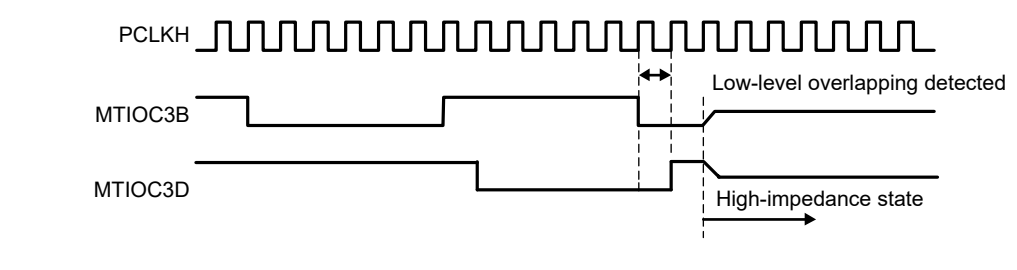


Figure 18.5 Output-level compare operation

18.3.4 High-Impedance Control Using Registers

The high-impedance state of the MTU pins (MTU0, MTU3, MTU4, MTU6, and MTU7) can be directly controlled by using the Software Port Output Enable Register (SPOER).

For instance, setting the MTUCH34HIZ bit in SPOER to 1 places the MTU3 and MTU4 pins specified by the Port Output Enable Control Register 2 (POECR2) in the high-impedance state.

The high-impedance state of other pins can also be controlled by setting the appropriate bits in SPOER.

18.3.5 High-Impedance Control through Detection of Stopped Oscillation and Oscillation Abnormality

When the oscillation-stop detection circuit in the clock pulse generator detects the stopped oscillation or the clock monitor circuit detects the oscillation abnormality for PLL0, PLL1, and main clock, with the OSTSTE bit in ICSR6 set, the MTU3 and MTU4 or MTU6 and MTU7 pins if specified in the Port Output Enable Control Register 2 (POECR2), the MTU0 pins if specified in the Port Output Enable Control Register 1 (POECR1) can be placed in the high-impedance state.

18.3.6 Additional Functions for High-Impedance Control

Settings in the Port Output Enable Control Registers 4 to 5 (POECR4 to POECR5) can add additional high-impedance control conditions for the MTU complementary PWM output, MTU0 pins.

For instance, the following settings can be added as high-impedance control conditions for the MTU3 and MTU4 pins:

- Setting the POECR4.IC2ADDMT34ZE bit to 1 adds the input-level detection by the POE4# pin
- Setting the POECR4.IC3ADDMT34ZE bit to 1 and adds the input-level detection by the POE8# pin
- Setting the POECR4.IC4ADDMT34ZE bit to 1 and adds the input-level detection by the POE10# pin
- Setting the POECR4.IC5ADDMT34ZE bit to 1 and adds the input-level detection by the POE11# pin

The high-impedance control of other pins can also be controlled by setting the appropriate bits in the POECR4 and POECR5 registers.

18.3.7 Recover from High-Impedance State

The outputs which have been in the high-impedance state due to input-level detection can be recovered from the state either by returning them to their initial state with a reset, or by clearing all of the ICSR1.POE0F, ICSR2.POE4F, ICSR3.POE8F, ICSR4.POE10F, and ICSR5.POE11F flags. However, when low-level sampling is selected with the ICSR1.POE0M[1:0], ICSR2.POE4M[1:0], ICSR3.POE8M[1:0], ICSR4.POE10M[1:0], and ICSR5.POE11M[1:0] bits, just writing 0 to a flag is ignored (the flag is not set to 0). Flag can be cleared by writing 0 to it only after a high level is input to the POE0#, POE4#, POE8#, POE10#, and POE11# pins and is detected.

The outputs which have been in the high-impedance state due to output-level detection can be recovered from the state either by returning them to their initial state with a reset, or by setting the OCSR1.OSF1 flag or the OCSR2.OSF2 flag to 0. However, just writing 0 to a flag is ignored (the flag is not set to 0). Flag can be cleared by writing 0 to it only after setting the inactive level to be output from the pin. In the MTU, the inactive level (initial output level) can be output by stopping the count operation.

The outputs which have been in the high-impedance state due to SPOER register setting can be recovered from the state either by returning them to their initial state with reset, or by setting the SPOER.MTUCH34HIZ bit, SPOER.MTUCH67HIZ bit or SPOER.MTUCH0HIZ bit to 0.

The outputs which have been in the high-impedance state due to oscillation stop detection for PLL0, PLL1, and main clock can be recovered from the state by returning them to their initial state with a reset.

The outputs which have been in the high-impedance state due to DSMIF error detection can be recovered from the state either by returning them to their initial state with reset, or by clearing DSMIF error status flags.

18.4 POE3 Setting Procedure

The procedure for setting POE3 is described in the section that follows. It shows an example of high-impedance control in response to comparison of the output levels on the MTU3 pins (MTIOC3B/MTIOC3D). In the procedure, P17_6 is selected as the MTIOC3B pin and P18_1 is selected as the MTIOC3D pin.

Set the MTU3 pins (P17_6 and P18_1) for the peripheral function by the PMCm and PFCm registers of the I/O Ports before making the following settings:

1. Set PMC17.PMC6 and PMC18.PMC1 bit to 1.
   Set the PMC17 and PMC18 registers of the I/O Ports to select P17_6 and P18_1 for the peripheral function pins.
2. Set PFC17.PFC6[3:0] bits to 0x0 and PFC18.PFC1[3:0] bits to 0x1.
   Set the PFC17 and PFC18 registers of the I/O Ports to select P17_6 for the MTIOC3B pin and P18_1 for the MTIOC3D pin.
3. Set the M3SELR.M3BSEL[3:0] bits to 0x0 and M3SELR.M3DSEL[3:0] bits to 0x0.
   Set the M3SELR register to control the high-impedance state of P17_6 for MTIOC3B and P18_1 for MTIOC3D.
4. Set the ALR1.OLSG0A and OLSG0B bits to 0 and the OLSEN bit to 1 to set MTIOC3B and MTIOC3D as active-low.
5. Set the POECR2.MTU3BDZE bit to 1 to enable high-impedance control on the MTIOC3B and MTIOC3D pins.
6. Specify the setting to operate MTU3.

After the above settings, if MTIOC3B (P17_6) and MTIOC3D (P18_1) are simultaneously at the low level, then the pins are placed in high-impedance state and the OCSR1.OSF1 bit is set.

For the method of releasing high-impedance, see section 18.3.7. Recover from High-Impedance State.

18.5 Interrupts

The POE3 issues a request to generate an interrupt when the specified condition is satisfied during input-level detection or output-level comparison. Table 18.6 shows the interrupt sources and their conditions.

Table 18.6 Interrupt sources and conditions

Name	Interrupt source	Interrupt flag	Condition
OEI1	Output enable interrupt 1	POE0F and OSF1	PIE1 • POE0F + OIE1 • OSF1
OEI2	Output enable interrupt 2	POE4F and OSF2	PIE2 • POE4F + OIE2 • OSF2
OEI3	Output enable interrupt 3	POE8F	PIE3 • POE8F
OEI4	Output enable interrupt 4	POE10F, POE11F, D0ERR0ST, D1ERR0ST, D0ERR1ST, and D1ERR1ST	PIE4 • POE10F + PIE5 • POE11F + D0ERR0IE • D0ERR0ST + D1ERR0IE • D1ERR0ST + D0ERR1IE • D0ERR1ST + D1ERR1IE • D1ERR1ST

18.6 Usage Notes

18.6.1 High-Impedance Control when the MTU is not Selected

If high-impedance control for a pin having a multiplexed MTU pin function on a register POECR1 to POECR2 is enabled and the high-impedance condition is satisfied, the pin is placed in the high-impedance state even if the MTU function is not selected for the pin on which it is multiplexed.

To avoid unintended high-impedance states, ensure that there are no differences between the settings for MTU pin selection in the PMCm and PFCm registers of the I/O Ports and for MTU pin selection in the pin select register of the POE.

19. General PWM Timer (GPT)

19.1 Overview

The LSI has a General PWM Timer (GPT) that is a 32-bit timer with 18 channels. These 18 channels are divided into three groups (LLPP, NONSAFETY, and SAFETY). Also, each group is independent of each other. PWM waveforms can be generated by controlling the up-counter, down-counter, or the up- and down-counter. The GPT can also be used as a general-purpose timer.

Table 19.1 lists the specifications for the GPT and Figure 19.1 shows a block diagram of the GPT. Table 19.2 lists the I/O pins and Table 19.3 lists the GPT interrupt sources.

Table 19.1 GPT specifications

Parameter	Description
Number of channels	• 18 channels • 7 channels in LLPP (GPT0 to GPT6) • 7 channels in NONSAFETY (GPT7 to GPT13) • 4 channels in SAFTY (GPT14 to GPT17)
Functions	• 32 bits • Up-counting or down-counting (saw waves) or up/down-counting (triangle waves) for each counter • Clock sources independently selectable for each channel • 2 I/O pins per channel • 2 output compare/input capture registers per channel • For the two output compare/input capture registers of each channel, four registers are provided as buffer registers and are capable of operating as comparison registers when buffering is not in use • In output compare operation, buffer switching can be at crests or troughs, enabling the generation of laterally asymmetric PWM waveforms • Registers for setting up frame cycles in each channel with capability for generating interrupts at overflow or underflow • Generation of dead times in PWM operation • Count direction switching is available in the triangle-wave PWM modes • Synchronous starting, stopping, clearing, and switching counters for arbitrary channels within group (LLPP, NONSAFETY, SAFETY) • Starting, stopping, clearing, switching, and capturing up/down counters in response to a maximum of eight ELC events for channels in LLPP and NONSAFETY • Starting, stopping, and clearing up/down counters in response to input level comparison • Starting, stopping, clearing, switching, and capturing up/down counters in response to a maximum of four external triggers • Output pin disable function by dead time error and detected short circuits between output pins • A/D converter start trigger can be generated • Compare match A to F event and overflow/underflow event can be output to the ELC for channels in LLPP and NONSAFETY • Enables the noise filter for input capture • Bus clock: PCLKH in LLPP, PCLKM in NONSAFETY, and PCLKM in SAFETY • Core clock: PCLKGPTL in LLPP, PCLKM in NONSAFETY, and PCLKM in SAFETY
Prescaler	• 9 dividing clocks are output • Selectable from any of core clock, core clock/2, core clock/4, core clock/8, core clock/16, core clock/32, core clock/64, core clock/256, core clock/1024
Interrupts	9 sources: • GTCCRA compare match/input capture (GPTn_CCMPA) • GTCCRB compare match/input capture (GPTn_CCMPB) • GTCCRC compare match (GPTn_CMPC) • GTCCRD compare match (GPTn_CMPD) • GTCCRE compare match (GPTn_CMPE) • GTCCRF compare match (GPTn_CMPF) • GTADTRA compare match (GPTn_ADTRGA) → only ELC for channels in LLPP and NONSAFETY • GTADTRB compare match (GPTn_ADTRGB) → only ELC for channels in LLPP and NONSAFETY • GTCNT overflow (GTPR compare match) (GPTn_OVF) • GTCNT underflow (GPTn_UDF) • Dead time error (GPTn_DTE) → only for channels in LLPP and NONSAFETY
Module-stop function	Module-stop state can be set to reduce power consumption

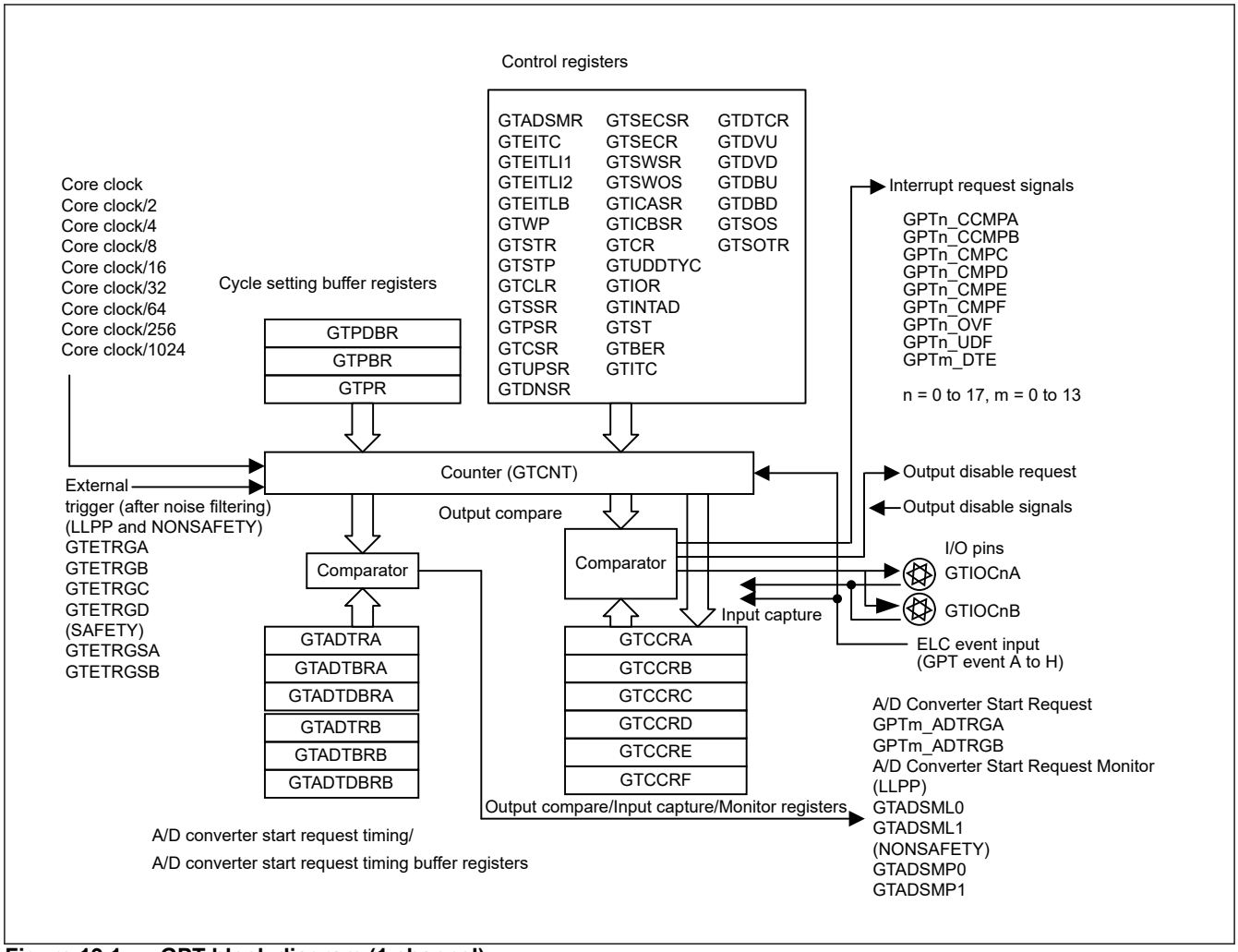


Figure 19.1 GPT block diagram (1 channel)

Table 19.2 GPT I/O pins (1 of 3)

Channel	Pin name	I/O	Function
Shared (LLPP and NONSAFETY)	GTETRGA	Input	External trigger input pin A (after noise filtering) for channels in LLPP and NONSAFETY
	GTETRGB	Input	External trigger input pin B (after noise filtering) for channels in LLPP and NONSAFETY
	GTETRGC	Input	External trigger input pin C (after noise filtering) for channels in LLPP and NONSAFETY
	GTETRGD	Input	External trigger input pin D (after noise filtering) for channels in LLPP and NONSAFETY
Shared (SAFETY)	GTETRGSA	Input	External trigger input pin A (after noise filtering) for channels in SAFETY
	GTETRGSB	Input	External trigger input pin B (after noise filtering) for channels in SAFETY
Shared (LLPP)	GTADSML0	Output	A/D conversion start request monitor 0 output pin for channels in LLPP
	GTADSML1	Output	A/D conversion start request monitor 1 output pin for channels in LLPP
Shared (NONSAFETY)	GTADSMP0	Output	A/D conversion start request monitor 0 output pin for channels in NONSAFETY
	GTADSMP1	Output	A/D conversion start request monitor 1 output pin for channels in NONSAFETY

Table 19.2 GPT I/O pins (2 of 3)

Channel	Pin name	I/O	Function
GPT0 (LLPP)	GTIOC0A	I/O	GPT0 GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC0B	I/O	GPT0 GTCCRB register input capture input/output compare output/PWM output pin
GPT1 (LLPP)	GTIOC1A	I/O	GPT1 GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC1B	I/O	GPT1 GTCCRB register input capture input/output compare output/PWM output pin
GPT2 (LLPP)	GTIOC2A	I/O	GPT2 GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC2B	I/O	GPT2 GTCCRB register input capture input/output compare output/PWM output pin
GPT3 (LLPP)	GTIOC3A	I/O	GPT3 GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC3B	I/O	GPT3 GTCCRB register input capture input/output compare output/PWM output pin
GPT4 (LLPP)	GTIOC4A	I/O	GPT4 GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC4B	I/O	GPT4 GTCCRB register input capture input/output compare output/PWM output pin
GPT5 (LLPP)	GTIOC5A	I/O	GPT5 GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC5B	I/O	GPT5 GTCCRB register input capture input/output compare output/PWM output pin
GPT6 (LLPP)	GTIOC6A	I/O	GPT6 GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC6B	I/O	GPT6 GTCCRB register input capture input/output compare output/PWM output pin
GPT7 (NONSAFETY)	GTIOC7A	I/O	GPT7 GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC7B	I/O	GPT7 GTCCRB register input capture input/output compare output/PWM output pin
GPT8 (NONSAFETY)	GTIOC8A	I/O	GPT8 GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC8B	I/O	GPT8 GTCCRB register input capture input/output compare output/PWM output pin
GPT9 (NONSAFETY)	GTIOC9A	I/O	GPT9 GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC9B	I/O	GPT9 GTCCRB register input capture input/output compare output/PWM output pin
GPT10 (NONSAFETY)	GTIOC10A	I/O	GPT10 GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC10B	I/O	GPT10 GTCCRB register input capture input/output compare output/PWM output pin
GPT11 (NONSAFETY)	GTIOC11A	I/O	GPT11 GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC11B	I/O	GPT11 GTCCRB register input capture input/output compare output/PWM output pin
GPT12 (NONSAFETY)	GTIOC12A	I/O	GPT12 GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC12B	I/O	GPT12 GTCCRB register input capture input/output compare output/PWM output pin

Table 19.2 GPT I/O pins (3 of 3)

Channel	Pin name	I/O	Function
GPT13 (NONSAFETY)	GTIOC13A	I/O	GPT13 GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC13B	I/O	GPT13 GTCCRB register input capture input/output compare output/PWM output pin
GPT14 (SAFETY)	GTIOC14A	I/O	GPT14 GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC14B	I/O	GPT14 GTCCRB register input capture input/output compare output/PWM output pin
GPT15 (SAFETY)	GTIOC15A	I/O	GPT15 GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC15B	I/O	GPT15 GTCCRB register input capture input/output compare output/PWM output pin
GPT16 (SAFETY)	GTIOC16A	I/O	GPT16 GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC16B	I/O	GPT16 GTCCRB register input capture input/output compare output/PWM output pin
GPT17 (SAFETY)	GTIOC17A	I/O	GPT17 GTCCRA register input capture input/output compare output/PWM output pin
	GTIOC17B	I/O	GPT17 GTCCRB register input capture input/output compare output/PWM output pin

Note: External triggers are available through POEG.

Table 19.3 GPT interrupt sources

Name	Interrupt sources	CPU0 GIC request	DMAC activation
GPTn_CCMPA	GTCCRA input capture / compare match	Possible	Possible for n = 0 to 13 Not Possible for n = 14 to 17
GPTn_CCMPB	GTCCRB input capture / compare match	Possible	Possible for n = 0 to 13 Not Possible for n = 14 to 17
GPTn_CMPC	GTCCRC compare match	Possible	Possible for n = 0 to 13 Not Possible for n = 14 to 17
GPTn_CMPD	GTCCRD compare match	Possible	Possible for n = 0 to 13 Not Possible for n = 14 to 17
GPTn_CMPE	GTCCRE compare match	Possible	Possible for n = 0 to 13 Not Possible for n = 14 to 17
GPTn_CMPF	GTCCRF compare match	Possible	Possible for n = 0 to 13 Not Possible for n = 14 to 17
GPTn_OVF	GTCNT overflow (GTPR compare match)	Possible	Possible for n = 0 to 13 Not Possible for n = 14 to 17
GPTn_UDF	GTCNT underflow	Possible	Possible for n = 0 to 13 Not Possible for n = 14 to 17
GPTm_DTE	Dead time error	Possible	Not possible
GPTm_ADTRGA*1	GTADTRA compare match (A/D Converter Start Request A)	Not possible	Not possible
GPTm_ADTRGB*1	GTADTRB compare match (A/D Converter Start Request B)	Not possible	Not possible

Note: n = 0 to 17, m = 0 to 13.
Note 1. A/D converter start requests are only used for ELC.

19.2 Register Descriptions

19.2.1 GTWP : General PWM Timer Write-Protection Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PRKEY[7:0]								—	—	—	CMNWP	CLRWP	STPWP	STRWP	WP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WP	Register Write Disabled 0: Write to the register is enabled 1: Write to the register is disabled	R/W
1	STRWP	GTSTR.CSTRT Bit Write Disabled 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
2	STPWP	GTSTP.CSTOP Bit Write Disabled 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
3	CLRWP	GTCLR.CCLR Bit Write Disabled 0: Write to the bit is enabled 1: Write to the bit is disabled	R/W
4	CMNWP	Common Register Write Disabled 0: Write to the register is enabled 1: Write to the register is disabled	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
15:8	PRKEY[7:0]	GTWP Key Code To modify the WP, STRWP, STPWP, CLRWP, and CMNWP bits, write 0xA5.	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

For detail, see section 19.8.1. Write-Protection for Registers.

19.2.2 GTSTR : General PWM Timer Software Start Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	CSTRT6	CSTRT5	CSTRT4	CSTRT3	CSTRT2	CSTRT1	CSTRT0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CSTRT0	Channel 0 Count Start 0: (Reading) The counter is stopped (Writing) No effect 1: (Reading) The counter is running (Writing) The counter starts running	R/W
1	CSTRT1	Channel 1 Count Start 0: (Reading) The counter is stopped (Writing) No effect 1: (Reading) The counter is running (Writing) The counter starts running	R/W
2	CSTRT2	Channel 2 Count Start 0: (Reading) The counter is stopped (Writing) No effect 1: (Reading) The counter is running (Writing) The counter starts running	R/W
3	CSTRT3	Channel 3 Count Start 0: (Reading) The counter is stopped (Writing) No effect 1: (Reading) The counter is running (Writing) The counter starts running	R/W
4	CSTRT4	Channel 4 Count Start 0: (Reading) The counter is stopped (Writing) No effect 1: (Reading) The counter is running (Writing) The counter starts running	R/W
5	CSTRT5	Channel 5 Count Start 0: (Reading) The counter is stopped (Writing) No effect 1: (Reading) The counter is running (Writing) The counter starts running	R/W
6	CSTRT6	Channel 6 Count Start 0: (Reading) The counter is stopped (Writing) No effect 1: (Reading) The counter is running (Writing) The counter starts running	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

The GTSTR register starts the GTCNT counter operation of each channel in the same domain group.

General PWM Timers are grouped by each of the following domains:

- LLPP
- NONSAFETY
- SAFETY

Bit 0 to bit 6 bits are available for the GTSTR register of LLPP and NONSAFETY domain group, and bit 0 to bit 3 are available for the GTSTR register of SAFTY domain group.

A bit position of the GTSTR register indicates a channel number assigned from 0th for each domain group. For example GPT7 is assigned bit 0 in the GTSTR register of NONSAFETY domain. The GTSTR register of each channel is a common register, and writing 1 to the GTSTR register in any channel and updating it can start operation of the GTCNT counter in all the channels related to the position of the bit written with 1. A change in counter operation and the register value is not generated by writing 0 to the bit.

19.2.3 GTSTP : General PWM Timer Software Stop Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	CSTOP6	CSTOP5	CSTOP4	CSTOP3	CSTOP2	CSTOP1	CSTOP0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	CSTOP0	Channel 0 Count Stop 0: (Reading) The counter is operating (Writing) No effect 1: (Reading) The counter is stopped (Writing) The counter is stopped	R/W
1	CSTOP1	Channel 1 Count Stop 0: (Reading) The counter is operating (Writing) No effect 1: (Reading) The counter is stopped (Writing) The counter is stopped	R/W
2	CSTOP2	Channel 2 Count Stop 0: (Reading) The counter is operating (Writing) No effect 1: (Reading) The counter is stopped (Writing) The counter is stopped	R/W
3	CSTOP3	Channel 3 Count Stop 0: (Reading) The counter is operating (Writing) No effect 1: (Reading) The counter is stopped (Writing) The counter is stopped	R/W
4	CSTOP4	Channel 4 Count Stop 0: (Reading) The counter is operating (Writing) No effect 1: (Reading) The counter is stopped (Writing) The counter is stopped	R/W
5	CSTOP5	Channel 5 Count Stop 0: (Reading) The counter is operating (Writing) No effect 1: (Reading) The counter is stopped (Writing) The counter is stopped	R/W
6	CSTOP6	Channel 6 Count Stop 0: (Reading) The counter is operating (Writing) No effect 1: (Reading) The counter is stopped (Writing) The counter is stopped	R/W
31:7	—	These bits are read as 1. The write value should be 1.	R/W

The GTSTP register stops the GTCNT counter operation of each channel in the same domain group.

General PWM Timers are grouped by each of the following domains:

- LLPP
- NONSAFETY

- SAFETY

Bit 0 to bit 6 are available for the GTSTP register of LLPP and NONSAFETY domain group, and bit 0 to bit 3 are available for the GTSTP register of SAFTY domain group.

A bit position of the GTSTP register indicates a channel number assigned from 0th for each domain group. For example GPT7 is assigned bit 0 in the GTSTP register of NONSAFETY domain. The GTSTP register of each channel is a common register, and writing 1 to the GTSTP register in any channel and updating it can stop operation of the GTCNT counter in all the channels related to the position of the bit written with 1. A change in counter operation and the register value is not generated by writing 0 to the bit.

19.2.4 GTCLR : General PWM Timer Software Clear Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	CCLR6	CCLR5	CCLR4	CCLR3	CCLR2	CCLR1	CCLR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CCLR0	Channel 0 Count Clear 0: The counter is not cleared 1: The counter is cleared	W
1	CCLR1	Channel 1 Count Clear 0: The counter is not cleared 1: The counter is cleared	W
2	CCLR2	Channel 2 Count Clear 0: The counter is not cleared 1: The counter is cleared	W
3	CCLR3	Channel 3 Count Clear 0: The counter is not cleared 1: The counter is cleared	W
4	CCLR4	Channel 4 Count Clear 0: The counter is not cleared 1: The counter is cleared	W
5	CCLR5	Channel 5 Count Clear 0: The counter is not cleared 1: The counter is cleared	W
6	CCLR6	Channel 6 Count Clear 0: The counter is not cleared 1: The counter is cleared	W
31:7	—	The write value should be 0.	W

The GTCLR register is a write-only register that sets the clearing of the GTCNT counter of each channel in the same domain group. A bit position of the GTCLR register indicates a channel number assigned from 0th for each domain group. For example GPT7 is assigned bit 0 in the GTCLR register of NONSAFETY domain. The GTCLR register of each channel is a common register, and writing 1 to the GTCLR register in any channel and updating it can clear the GTCNT counter in all the channels related to the position of the bit written with 1. A change in counter operation and the register value is not generated by writing 0 to the bit.

General PWM Timers are grouped by each domain, LLPP, NONSAFETY and SAFETY. Bit 0 to bit 6 are available for the GTCLR register of LLPP and NONSAFETY domain group, and bit 0 to bit 3 are available for the GTCLR register of SAFTY domain group.

19.2.5 GTSSR : General PWM Timer Start Source Select Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CSTRT	—	—	—	—	—	—	—	SSELCH	SSELCG	SSELCF	SSELCE	SSELCD	SSELCC	SSELCB	SSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SSCBFAHL[1:0]		SSCBRAHL[1:0]		SSCAFBHL[1:0]		SSCARBHL[1:0]		SSGTRGDFR[1:0]		SSGTRGCFR[1:0]		SSGTRGBFR[1:0]		SSGTRGAFR[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SSGTRGAFR[1:0]	GTETRGA Signal Edge Select (GTETRGSA Signal for SAFETY) 0 0: The GTETRGA signal is not used as a trigger to start counting 0 1: The counter starts at a rising edge of the GTETRGA signal 1 0: The counter starts at a falling edge of the GTETRGA signal 1 1: The counter starts at both edges of the GTETRGA signal	R/W
3:2	SSGTRGBFR[1:0]	GTETRGB Signal Edge Select (GTETRGSB Signal for SAFETY) 0 0: The GTETRGB signal is not used as a trigger to start counting 0 1: The counter starts at a rising edge of the GTETRGB signal 1 0: The counter starts at a falling edge of the GTETRGB signal 1 1: The counter starts at both edges of the GTETRGB signal	R/W
5:4	SSGTRGCFR[1:0]	GTETRGC Signal Edge Select (GTETRGSC Signal for SAFETY) 0 0: The GTETRGC signal is not used as a trigger to start counting 0 1: The counter starts at a rising edge of the GTETRGC signal 1 0: The counter starts at a falling edge of the GTETRGC signal 1 1: The counter starts at both edges of the GTETRGC signal	R/W
7:6	SSGTRGDFR[1:0]	GTETRGD Signal Edge Select (GTETRGSD Signal for SAFETY) 0 0: The GTETRGD signal is not used as a trigger to start counting 0 1: The counter starts at a rising edge of the GTETRGD signal 1 0: The counter starts at a falling edge of the GTETRGD signal 1 1: The counter starts at both edges of the GTETRGD signal	R/W
9:8	SSCARBHL[1:0]	GTIOCnA Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to start counting 0 1: The counter starts at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low 1 0: The counter starts at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high 1 1: The counter starts at a rising edge of the GTIOCnA signal	R/W
11:10	SSCAFBHL[1:0]	GTIOCnA Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to start counting 0 1: The counter starts at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low 1 0: The counter starts at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high 1 1: The counter starts at a falling edge of the GTIOCnA signal	R/W

Bit	Symbol	Function	R/W
13:12	SSCBRAHL[1:0]	GTIOCnB Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to start counting 0 1: The counter starts at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low 1 0: The counter starts at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high 1 1: The counter starts at a rising edge of the GTIOCnB signal	R/W
15:14	SSCBFAHL[1:0]	GTIOCnB Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to start counting 0 1: The counter starts at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low 1 0: The counter starts at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high 1 1: The counter starts at a falling edge of the GTIOCnB signal	R/W
16	SSELCA	Event Source Count Start Enable 0: Disables count start by the LLPP GPT event A / NONSAFTY GPT event A input 1: Enables count start by the LLPP GPT event A / NONSAFTY GPT event A input	R/W
17	SSELCB	Event Source Count Start Enable 0: Disables count start by the LLPP GPT event B / NONSAFTY GPT event B input 1: Enables count start by the LLPP GPT event B / NONSAFTY GPT event B input	R/W
18	SSELCC	Event Source Count Start Enable 0: Disables count start by the LLPP GPT event C / NONSAFTY GPT event C input 1: Enables count start by the LLPP GPT event C / NONSAFTY GPT event C input	R/W
19	SSELCD	Event Source Count Start Enable 0: Disables count start by the LLPP GPT event D / NONSAFTY GPT event D input 1: Enables count start by the LLPP GPT event D / NONSAFTY GPT event D input	R/W
20	SSELCE	Event Source Count Start Enable 0: Disables count start by the LLPP GPT event E / NONSAFTY GPT event E input 1: Enables count start by the LLPP GPT event E / NONSAFTY GPT event E input	R/W
21	SSELCF	Event Source Count Start Enable 0: Disables count start by the LLPP GPT event F / NONSAFTY GPT event F input 1: Enables count start by the LLPP GPT event F / NONSAFTY GPT event F input	R/W
22	SSELCG	Event Source Count Start Enable 0: Disables count start by the LLPP GPT event G / NONSAFTY GPT event G input 1: Enables count start by the LLPP GPT event G / NONSAFTY GPT event G input	R/W
23	SSELCH	Event Source Count Start Enable 0: Disables count start by the LLPP GPT event H / NONSAFTY GPT event H input 1: Enables count start by the LLPP GPT event H / NONSAFTY GPT event H input	R/W
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	CSTRT	Software Source Count Start Enable 0: Disables count start by the GTSTR register 1: Enables count start by the GTSTR register	R/W

19.2.6 GTPSR : General PWM Timer Stop Source Select Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CSTOP	—	—	—	—	—	—	—	PSELCH	PSELCG	PSELCF	PSELCE	PSELCD	PSELCC	PSELCB	PSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PSCBFAHL[1:0]		PSCBRAHL[1:0]		PSCAFBHL[1:0]		PSCARBHL[1:0]		PSGTRGDFR[1:0]		PSGTRGCFR[1:0]		PSGTRGBFR[1:0]		PSGTRGAFR[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PSGTRGAFR[1:0]	GTETRGA Signal Edge Select (GTETRGSA Signal for SAFETY) 0 0: The GTETRGA signal is not used as a trigger to stop counting 0 1: The counter stops at a rising edge of the GTETRGA signal 1 0: The counter stops at a falling edge of the GTETRGA signal 1 1: The counter stops at both edges of the GTETRGA signal	R/W
3:2	PSGTRGBFR[1:0]	GTETRGB Signal Edge Select (GTETRGSB Signal for SAFETY) 0 0: The GTETRGB signal is not used as a trigger to stop counting 0 1: The counter stops at a rising edge of the GTETRGB signal 1 0: The counter stops at a falling edge of the GTETRGB signal 1 1: The counter stops at both edges of the GTETRGB signal	R/W
5:4	PSGTRGCFR[1:0]	GTETRGC Signal Edge Select (GTETRGSC Signal for SAFETY) 0 0: The GTETRGC signal is not used as a trigger to stop counting 0 1: The counter stops at a rising edge of the GTETRGC signal 1 0: The counter stops at a falling edge of the GTETRGC signal 1 1: The counter stops at both edges of the GTETRGC signal	R/W
7:6	PSGTRGDFR[1:0]	GTETRGD Signal Edge Select (GTETRGSD Signal for SAFETY) 0 0: The GTETRGD signal is not used as a trigger to stop counting 0 1: The counter stops at a rising edge of the GTETRGD signal 1 0: The counter stops at a falling edge of the GTETRGD signal 1 1: The counter stops at both edges of the GTETRGD signal	R/W
9:8	PSCARBHL[1:0]	GTIOCnA Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to stop counting 0 1: The counter stops at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low 1 0: The counter stops at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high 1 1: The counter stops at a rising edge of the GTIOCnA signal	R/W
11:10	PSCAFBHL[1:0]	GTIOCnA Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to stop counting 0 1: The counter stops at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low 1 0: The counter stops at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high 1 1: The counter stops at a falling edge of the GTIOCnA signal	R/W
13:12	PSCBRAHL[1:0]	GTIOCnB Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to stop counting 0 1: The counter stops at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low 1 0: The counter stops at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high 1 1: The counter stops at a rising edge of the GTIOCnB signal	R/W

Bit	Symbol	Function	R/W
15:14	PSCBFAHL[1:0]	GTIOCnB Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to stop counting 0 1: The counter stops at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low 1 0: The counter stops at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high 1 1: The counter stops at a falling edge of the GTIOCnB signal	R/W
16	PSELCA	Event Source Count Stop Enable 0: Disables count stop by the LLPP GPT event A / NONSAFTY GPT event A input 1: Enables count stop by the LLPP GPT event A / NONSAFTY GPT event A input	R/W
17	PSELCB	Event Source Count Stop Enable 0: Disables count stop by the LLPP GPT event B / NONSAFTY GPT event B input 1: Enables count stop by the LLPP GPT event B / NONSAFTY GPT event B input	R/W
18	PSELCC	Event Source Count Stop Enable 0: Disables count stop by the LLPP GPT event C / NONSAFTY GPT event C input 1: Enables count stop by the LLPP GPT event C / NONSAFTY GPT event C input	R/W
19	PSELCD	Event Source Count Stop Enable 0: Disables count stop by the LLPP GPT event D / NONSAFTY GPT event D input 1: Enables count stop by the LLPP GPT event D / NONSAFTY GPT event D input	R/W
20	PSELCE	Event Source Count Stop Enable 0: Disables count stop by the LLPP GPT event E / NONSAFTY GPT event E input 1: Enables count stop by the LLPP GPT event E / NONSAFTY GPT event E input	R/W
21	PSELCF	Event Source Count Stop Enable 0: Disables count stop by the LLPP GPT event F / NONSAFTY GPT event F input 1: Enables count stop by the LLPP GPT event F / NONSAFTY GPT event F input	R/W
22	PSELCG	Event Source Count Stop Enable 0: Disables count stop by the LLPP GPT event G / NONSAFTY GPT event G input 1: Enables count stop by the LLPP GPT event G / NONSAFTY GPT event G input	R/W
23	PSELCH	Event Source Count Stop Enable 0: Disables count stop by the LLPP GPT event H / NONSAFTY GPT event H input 1: Enables count stop by the LLPP GPT event H / NONSAFTY GPT event H input	R/W
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	CSTOP	Software Source Count Stop Enable 0: Disables count stop by the GTSTP register 1: Enables count stop by the GTSTP register	R/W

19.2.7 GTCSR : General PWM Timer Clear Source Select Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CCLR	—	—	—	—	—	—	—	CSELCH	CSELCG	CSELCF	CSELCE	CSELCD	CSELCC	CSELCB	CSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CSCBFAHL[1:0]		CSCBRAHL[1:0]		CSCAFBHL[1:0]		CSCARBHL[1:0]		CSGTRGDFR[1:0]		CSGTRGCFR[1:0]		CSGTRGBFR[1:0]		CSGTRGAFR[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CSGTRGAFR[1:0]	GTETRGA Signal Edge Select (GTETRGSA Signal for SAFETY) 0 0: The GTETRGA signal is not used as a trigger to clear the counter 0 1: The counter is cleared at a rising edge of the GTETRGA signal 1 0: The counter is cleared at a falling edge of the GTETRGA signal 1 1: The counter is cleared at both edges of the GTETRGA signal	R/W
3:2	CSGTRGBFR[1:0]	GTETRGB Signal Edge Select (GTETRGSB Signal for SAFETY) 0 0: The GTETRGB signal is not used as a trigger to clear the counter 0 1: The counter is cleared at a rising edge of the GTETRGB signal 1 0: The counter is cleared at a falling edge of the GTETRGB signal 1 1: The counter is cleared at both edges of the GTETRGB signal	R/W
5:4	CSGTRGCFR[1:0]	GTETRGC Signal Edge Select (GTETRGSC Signal for SAFETY) 0 0: The GTETRGC signal is not used as a trigger to clear the counter 0 1: The counter is cleared at a rising edge of the GTETRGC signal 1 0: The counter is cleared at a falling edge of the GTETRGC signal 1 1: The counter is cleared at both edges of the GTETRGC signal	R/W
7:6	CSGTRGDFR[1:0]	GTETRGD Signal Edge Select (GTETRGSD Signal for SAFETY) 0 0: The GTETRGD signal is not used as a trigger to clear the counter 0 1: The counter is cleared at a rising edge of the GTETRGD signal 1 0: The counter is cleared at a falling edge of the GTETRGD signal 1 1: The counter is cleared at both edges of the GTETRGD signal	R/W
9:8	CSCARBHL[1:0]	GTIOCnA Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to clear the counter 0 1: The counter is cleared at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low 1 0: The counter is cleared at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high 1 1: The counter is cleared at a rising edge of the GTIOCnA signal	R/W
11:10	CSCAFBHL[1:0]	GTIOCnA Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to clear the counter 0 1: The counter is cleared at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low 1 0: The counter is cleared at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high 1 1: The counter is cleared at a falling edge of the GTIOCnA signal	R/W
13:12	CSCBRAHL[1:0]	GTIOCnB Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to clear the counter 0 1: The counter is cleared at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low 1 0: The counter is cleared at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high 1 1: The counter is cleared at a rising edge of the GTIOCnB signal	R/W
15:14	CSCBFAHL[1:0]	GTIOCnB Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to clear the counter 0 1: The counter is cleared at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low 1 0: The counter is cleared at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high 1 1: The counter is cleared at a falling edge of the GTIOCnB signal	R/W
16	CSELCA	Event Source Count Clear Enable 0: Disables counter clear by the LLPP GPT event A / NONSAFTY GPT event A input 1: Enables counter clear by the LLPP GPT event A / NONSAFTY GPT event A input	R/W
17	CSELCB	Event Source Count Clear Enable 0: Disables counter clear by the LLPP GPT event B / NONSAFTY GPT event B input 1: Enables counter clear by the LLPP GPT event B / NONSAFTY GPT event B input	R/W
18	CSELCC	Event Source Count Clear Enable 0: Disables counter clear by the LLPP GPT event C / NONSAFTY GPT event C input 1: Enables counter clear by the LLPP GPT event C / NONSAFTY GPT event C input	R/W

Bit	Symbol	Function	R/W
19	CSELCD	Event Source Count Clear Enable 0: Disables counter clear by the LLPP GPT event D / NONSAFTY GPT event D input 1: Enables counter clear by the LLPP GPT event D / NONSAFTY GPT event D input	R/W
20	CSELCE	Event Source Count Clear Enable 0: Disables counter clear by the LLPP GPT event E / NONSAFTY GPT event E input 1: Enables counter clear by the LLPP GPT event E / NONSAFTY GPT event E input	R/W
21	CSELCF	Event Source Count Clear Enable 0: Disables counter clear by the LLPP GPT event F / NONSAFTY GPT event F input 1: Enables counter clear by the LLPP GPT event F / NONSAFTY GPT event F input	R/W
22	CSELCG	Event Source Count Clear Enable 0: Disables counter clear by the LLPP GPT event G / NONSAFTY GPT event G input 1: Enables counter clear by the LLPP GPT event G / NONSAFTY GPT event G input	R/W
23	CSELCH	Event Source Count Clear Enable 0: Disables counter clear by the LLPP GPT event H / NONSAFTY GPT event H input 1: Enables counter clear by the LLPP GPT event H / NONSAFTY GPT event H input	R/W
30:24	—	These bits are read as 0. The write value should be 0.	R/W
31	CCLR	Software Source Count Clear Enable 0: Disables counter clear by the GTCLR register 1: Enables counter clear by the GTCLR register	R/W

19.2.8 GTUPSR : General PWM Timer Count-Up Source Select Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x1C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	USELCH	USELCG	USELCF	USELCE	USELCD	USELCC	USELCB	USELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	USCBFAHL[1:0]		USCBRAHL[1:0]		USCAFBHL[1:0]		USCARBHL[1:0]		USGTRGDFR[1:0]		USGTRGCFR[1:0]		USGTRGBFR[1:0]		USGTRGAFR[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	USGTRGAFR[1:0]	GTETRGA Signal Edge Select (GTETRGSA Signal for SAFETY) 0 0: The GTETRGA signal is not used as a trigger to increment the counter 0 1: The counter is incremented at a rising edge of the GTETRGA signal 1 0: The counter is incremented at a falling edge of the GTETRGA signal 1 1: The counter is incremented at both edges of the GTETRGA signal	R/W
3:2	USGTRGBFR[1:0]	GTETRGB Signal Edge Select (GTETRGSB Signal for SAFETY) 0 0: The GTETRGB signal is not used as a trigger to increment the counter 0 1: The counter is incremented at a rising edge of the GTETRGB signal 1 0: The counter is incremented at a falling edge of the GTETRGB signal 1 1: The counter is incremented at both edges of the GTETRGB signal	R/W
5:4	USGTRGCFR[1:0]	GTETRGC Signal Edge Select (GTETRGCSC Signal for SAFETY) 0 0: The GTETRGC signal is not used as a trigger to increment the counter 0 1: The counter is incremented at a rising edge of the GTETRGC signal 1 0: The counter is incremented at a falling edge of the GTETRGC signal 1 1: The counter is incremented at both edges of the GTETRGC signal	R/W

Bit	Symbol	Function	R/W
7:6	USGTRGDFR[1:0]	GTETRGD Signal Edge Select (GTETRGSD Signal for SAFETY) 0 0: The GTETRGD signal is not used as a trigger to increment the counter 0 1: The counter is incremented at a rising edge of the GTETRGD signal 1 0: The counter is incremented at a falling edge of the GTETRGD signal 1 1: The counter is incremented at both edges of the GTETRGD signal	R/W
9:8	USCARBHL[1:0]	GTIOCnA Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to increment the counter 0 1: The counter is incremented at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low 1 0: The counter is incremented at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high 1 1: The counter is incremented at a rising edge of the GTIOCnA signal	R/W
11:10	USCAFBHL[1:0]	GTIOCnA Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to increment the counter 0 1: The counter is incremented at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low 1 0: The counter is incremented at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high 1 1: The counter is incremented at a falling edge of the GTIOCnA signal	R/W
13:12	USCBRAHL[1:0]	GTIOCnB Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to increment the counter 0 1: The counter is incremented at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low 1 0: The counter is incremented at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high 1 1: The counter is incremented at a rising edge of the GTIOCnB signal	R/W
15:14	USCBFAHL[1:0]	GTIOCnB Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to increment the counter 0 1: The counter is incremented at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low 1 0: The counter is incremented at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high 1 1: The counter is incremented at a falling edge of the GTIOCnB signal	R/W
16	USELCA	Event Source Count-Up Enable 0: Disables count-up by the LLPP GPT event A / NONSAFTY GPT event A input 1: Enables count-up by the LLPP GPT event A / NONSAFTY GPT event A input	R/W
17	USELCB	Event Source Count-Up Enable 0: Disables count-up by the LLPP GPT event B / NONSAFTY GPT event B input 1: Enables count-up by the LLPP GPT event B / NONSAFTY GPT event B input	R/W
18	USELCC	Event Source Count-Up Enable 0: Disables count-up by the LLPP GPT event C / NONSAFTY GPT event C input 1: Enables count-up by the LLPP GPT event C / NONSAFTY GPT event C input	R/W
19	USELCD	Event Source Count-Up Enable 0: Disables count-up by the LLPP GPT event D / NONSAFTY GPT event D input 1: Enables count-up by the LLPP GPT event D / NONSAFTY GPT event D input	R/W
20	USELCE	Event Source Count-Up Enable 0: Disables count-up by the LLPP GPT event E / NONSAFTY GPT event E input 1: Enables count-up by the LLPP GPT event E / NONSAFTY GPT event E input	R/W
21	USELCF	Event Source Count-Up Enable 0: Disables count-up by the LLPP GPT event F / NONSAFTY GPT event F input 1: Enables count-up by the LLPP GPT event F / NONSAFTY GPT event F input	R/W
22	USELCG	Event Source Count-Up Enable 0: Disables count-up by the LLPP GPT event G / NONSAFTY GPT event G input 1: Enables count-up by the LLPP GPT event G / NONSAFTY GPT event G input	R/W

Bit	Symbol	Function	R/W
23	USELCH	Event Source Count-Up Enable 0: Disables count-up by the LLPP GPT event H / NONSAFTY GPT event H input 1: Enables count-up by the LLPP GPT event H / NONSAFTY GPT event H input	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

19.2.9 GTDNSR : General PWM Timer Count-Down Source Select Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DSELCH	DSELCG	DSELCF	DSELCE	DSELCD	DSELCC	DSELCB	DSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DSCBFAHL[1:0]		DSCBRAHL[1:0]		DSCAFBHL[1:0]		DSCARBHL[1:0]		DSGTRGDFR[1:0]		DSGTRGCFR[1:0]		DSGTRGBFR[1:0]		DSGTRGAFR[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	DSGTRGAFR[1:0]	GTETRGA Signal Edge Select (GTETRGSA Signal for SAFETY) 0 0: The GTETRGA signal is not used as a trigger to decrement the counter 0 1: The counter is decremented at a rising edge of the GTETRGA signal 1 0: The counter is decremented at a falling edge of the GTETRGA signal 1 1: The counter is decremented at both edges of the GTETRGA signal	R/W
3:2	DSGTRGBFR[1:0]	GTETRGB Signal Edge Select (GTETRGSB Signal for SAFETY) 0 0: The GTETRGB signal is not used as a trigger to decrement the counter 0 1: The counter is decremented at a rising edge of the GTETRGB signal 1 0: The counter is decremented at a falling edge of the GTETRGB signal 1 1: The counter is decremented at both edges of the GTETRGB signal	R/W
5:4	DSGTRGCFR[1:0]	GTETRGC Signal Edge Select (GTETRGSC Signal for SAFETY) 0 0: The GTETRGC signal is not used as a trigger to decrement the counter 0 1: The counter is decremented at a rising edge of the GTETRGC signal 1 0: The counter is decremented at a falling edge of the GTETRGC signal 1 1: The counter is decremented at both edges of the GTETRGC signal	R/W
7:6	DSGTRGDFR[1:0]	GTETRGD Signal Edge Select (GTETRGSD Signal for SAFETY) 0 0: The GTETRGD signal is not used as a trigger to decrement the counter 0 1: The counter is decremented at a rising edge of the GTETRGD signal 1 0: The counter is decremented at a falling edge of the GTETRGD signal 1 1: The counter is decremented at both edges of the GTETRGD signal	R/W
9:8	DSCARBHL[1:0]	GTIOCnA Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to decrement the counter 0 1: The counter is decremented at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low 1 0: The counter is decremented at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high 1 1: The counter is decremented at a rising edge of the GTIOCnA signal	R/W
11:10	DSCAFBHL[1:0]	GTIOCnA Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to decrement the counter 0 1: The counter is decremented at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low 1 0: The counter is decremented at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high 1 1: The counter is decremented at a falling edge of the GTIOCnA signal	R/W

Bit	Symbol	Function	R/W
13:12	DSCBRAHL[1:0]	GTIOCnB Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to decrement the counter 0 1: The counter is decremented at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low 1 0: The counter is decremented at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high 1 1: The counter is decremented at a rising edge of the GTIOCnB signal	R/W
15:14	DSCBFAHL[1:0]	GTIOCnB Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to decrement the counter 0 1: The counter is decremented at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low 1 0: The counter is decremented at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high 1 1: The counter is decremented at a falling edge of the GTIOCnB signal	R/W
16	DSELCA	Event Source Count-Down Enable 0: Disables count-down by the LLPP GPT event A / NONSAFTY GPT event A input 1: Enables count-down by the LLPP GPT event A / NONSAFTY GPT event A input	R/W
17	DSELCB	Event Source Count-Down Enable 0: Disables count-down by the LLPP GPT event B / NONSAFTY GPT event B input 1: Enables count-down by the LLPP GPT event B / NONSAFTY GPT event B input	R/W
18	DSELCC	Event Source Count-Down Enable 0: Disables count-down by the LLPP GPT event C / NONSAFTY GPT event C input 1: Enables count-down by the LLPP GPT event C / NONSAFTY GPT event C input	R/W
19	DSELCD	Event Source Count-Down Enable 0: Disables count-down by the LLPP GPT event D / NONSAFTY GPT event D input 1: Enables count-down by the LLPP GPT event D / NONSAFTY GPT event D input	R/W
20	DSELCE	Event Source Count-Down Enable 0: Disables count-down by the LLPP GPT event E / NONSAFTY GPT event E input 1: Enables count-down by the LLPP GPT event E / NONSAFTY GPT event E input	R/W
21	DSELCF	Event Source Count-Down Enable 0: Disables count-down by the LLPP GPT event F / NONSAFTY GPT event F input 1: Enables count-down by the LLPP GPT event F / NONSAFTY GPT event F input	R/W
22	DSELCG	Event Source Count-Down Enable 0: Disables count-down by the LLPP GPT event G / NONSAFTY GPT event G input 1: Enables count-down by the LLPP GPT event G / NONSAFTY GPT event G input	R/W
23	DSELCH	Event Source Count-Down Enable 0: Disables count-down by the LLPP GPT event H / NONSAFTY GPT event H input 1: Enables count-down by the LLPP GPT event H / NONSAFTY GPT event H input	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

19.2.10 GTICASR : General PWM Timer Input Capture Source Select Register A

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	ASELCH	ASELCG	ASELCF	ASELCE	ASELCD	ASELCC	ASELCB	ASELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ASCBFAHL[1:0]		ASCBRAHL[1:0]		ASCAFBHL[1:0]		ASCARBHL[1:0]		ASGTRGDFR[1:0]		ASGTRGCFR[1:0]		ASGTRGBFR[1:0]		ASGTRGAFR[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ASGTRGAFR[1:0]	GTETRGA Signal Edge Select (GTETRGSA Signal for SAFETY) 0 0: The GTETRGA signal is not used as a trigger to capture the counter value in the GTCCRA register 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTETRGA signal 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTETRGA signal 1 1: The counter value is captured in the GTCCRA register at both edges of the GTETRGA signal	R/W
3:2	ASGTRGBFR[1:0]	GTETRGB Signal Edge Select (GTETRGSB Signal for SAFETY) 0 0: The GTETRGB signal is not used as a trigger to capture the counter value in the GTCCRA register 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTETRGB signal 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTETRGB signal 1 1: The counter value is captured in the GTCCRA register at both edges of the GTETRGB signal	R/W
5:4	ASGTRGCFR[1:0]	GTETRGC Signal Edge Select (GTETRGSC Signal for SAFETY) 0 0: The GTETRGC signal is not used as a trigger to capture the counter value in the GTCCRA register 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTETRGC signal 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTETRGC signal 1 1: The counter value is captured in the GTCCRA register at both edges of the GTETRGC signal	R/W
7:6	ASGTRGDFR[1:0]	GTETRGD Signal Edge Select (GTETRGSD Signal for SAFETY) 0 0: The GTETRGD signal is not used as a trigger to capture the counter value in the GTCCRA register 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTETRGD signal 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTETRGD signal 1 1: The counter value is captured in the GTCCRA register at both edges of the GTETRGD signal	R/W
9:8	ASCARBHL[1:0]	GTIOCnA Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to capture the counter value in the GTCCRA register 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low 1 0: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high 1 1: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnA signal	R/W
11:10	ASCAFBHL[1:0]	GTIOCnA Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to capture the counter value in the GTCCRA register 0 1: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high 1 1: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnA signal	R/W

Bit	Symbol	Function	R/W
13:12	ASCBRAHL[1:0]	GTIOCnB Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to capture the counter value in the GTCCRA register 0 1: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low 1 0: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high 1 1: The counter value is captured in the GTCCRA register at a rising edge of the GTIOCnB signal	R/W
15:14	ASCBFAHL[1:0]	GTIOCnB Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to capture the counter value in the GTCCRA register 0 1: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low 1 0: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high 1 1: The counter value is captured in the GTCCRA register at a falling edge of the GTIOCnB signal	R/W
16	ASELCA	Event Source GTCCRA Input Capture Enable 0: Disables GTCCRA input capture by the LLPP GPT event A / NONSAFTY GPT event A input 1: Enables GTCCRA input capture by the LLPP GPT event A / NONSAFTY GPT event A input	R/W
17	ASELCB	Event Source GTCCRA Input Capture Enable 0: Disables GTCCRA input capture by the LLPP GPT event B / NONSAFTY GPT event B input 1: Enables GTCCRA input capture by the LLPP GPT event B / NONSAFTY GPT event B input	R/W
18	ASELCC	Event Source GTCCRA Input Capture Enable 0: Disables GTCCRA input capture by the LLPP GPT event C / NONSAFTY GPT event C input 1: Enables GTCCRA input capture by the LLPP GPT event C / NONSAFTY GPT event C input	R/W
19	ASELCD	Event Source GTCCRA Input Capture Enable 0: Disables GTCCRA input capture by the LLPP GPT event D / NONSAFTY GPT event D input 1: Enables GTCCRA input capture by the LLPP GPT event D / NONSAFTY GPT event D input	R/W
20	ASELCE	Event Source GTCCRA Input Capture Enable 0: Disables GTCCRA input capture by the LLPP GPT event E / NONSAFTY GPT event E input 1: Enables GTCCRA input capture by the LLPP GPT event E / NONSAFTY GPT event E input	R/W
21	ASELCF	Event Source GTCCRA Input Capture Enable 0: Disables GTCCRA input capture by the LLPP GPT event F / NONSAFTY GPT event F input 1: Enables GTCCRA input capture by the LLPP GPT event F / NONSAFTY GPT event F input	R/W
22	ASELCG	Event Source GTCCRA Input Capture Enable 0: Disables GTCCRA input capture by the LLPP GPT event G / NONSAFTY GPT event G input 1: Enables GTCCRA input capture by the LLPP GPT event G / NONSAFTY GPT event G input	R/W
23	ASELCH	Event Source GTCCRA Input Capture Enable 0: Disables GTCCRA input capture by the LLPP GPT event H / NONSAFTY GPT event H input 1: Enables GTCCRA input capture by the LLPP GPT event H / NONSAFTY GPT event H input	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

19.2.11 GTICBSR : General PWM Timer Input Capture Source Select Register B

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x28

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	BSELCH	BSELCG	BSELCF	BSELCE	BSELCD	BSELCC	BSELCB	BSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSCBFAHL[1:0]		BSCBRAHL[1:0]		BSCAFBHL[1:0]		BSCARBHL[1:0]		BSGTRGDFR[1:0]		BSGTRGCFR[1:0]		BSGTRGBFR[1:0]		BSGTRGAFR[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	BSGTRGAFR[1:0]	GTETRGA Signal Edge Select (GTETRGSA Signal for SAFETY) 0 0: The GTETRGA signal is not used as a trigger to capture the counter value in the GTCCRB register 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTETRGA signal 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTETRGA signal 1 1: The counter value is captured in the GTCCRB register at both edges of the GTETRGA signal	R/W
3:2	BSGTRGBFR[1:0]	GTETRGB Signal Edge Select (GTETRGSB Signal for SAFETY) 0 0: The GTETRGB signal is not used as a trigger to capture the counter value in the GTCCRB register 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTETRGB signal 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTETRGB signal 1 1: The counter value is captured in the GTCCRB register at both edges of the GTETRGB signal	R/W
5:4	BSGTRGCFR[1:0]	GTETRGC Signal Edge Select (GTETRGSC Signal for SAFETY) 0 0: The GTETRGC signal is not used as a trigger to capture the counter value in the GTCCRB register 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTETRGC signal 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTETRGC signal 1 1: The counter value is captured in the GTCCRB register at both edges of the GTETRGC signal	R/W
7:6	BSGTRGDFR[1:0]	GTETRGD Signal Edge Select (GTETRGSD Signal for SAFETY) 0 0: The GTETRGD signal is not used as a trigger to capture the counter value in the GTCCRB register 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTETRGD signal 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTETRGD signal 1 1: The counter value is captured in the GTCCRB register at both edges of the GTETRGD signal	R/W
9:8	BSCARBHL[1:0]	GTIOCnA Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCnA signal is not used as a trigger to capture the counter value in the GTCCRB register 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven low 1 0: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnA signal while the GTIOCnB pin is driven high 1 1: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnA signal	R/W

Bit	Symbol	Function	R/W
11:10	BSCAFBHL[1:0]	GTIOCnA Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCnA signal is not used as a trigger to capture the counter value in the GTCCRB register 0 1: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven low 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnA signal while the GTIOCnB pin is driven high 1 1: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnA signal	R/W
13:12	BSCBRAHL[1:0]	GTIOCnB Signal Rising Edge Applying Condition Select 0 0: Rising edge of the GTIOCnB signal is not used as a trigger to capture the counter value in the GTCCRB register 0 1: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven low 1 0: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnB signal while the GTIOCnA pin is driven high 1 1: The counter value is captured in the GTCCRB register at a rising edge of the GTIOCnB signal	R/W
15:14	BSCBFAHL[1:0]	GTIOCnB Signal Falling Edge Applying Condition Select 0 0: Falling edge of the GTIOCnB signal is not used as a trigger to capture the counter value in the GTCCRB register 0 1: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven low 1 0: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnB signal while the GTIOCnA pin is driven high 1 1: The counter value is captured in the GTCCRB register at a falling edge of the GTIOCnB signal	R/W
16	BSELCA	Event Source GTCCRB Input Capture Enable 0: Disables GTCCRB input capture by the LLPP GPT event A / NONSAFTY GPT event A input 1: Enables GTCCRB input capture by the LLPP GPT event A / NONSAFTY GPT event A input	R/W
17	BSELCB	Event Source GTCCRB Input Capture Enable 0: Disables GTCCRB input capture by the LLPP GPT event B / NONSAFTY GPT event B input 1: Enables GTCCRB input capture by the LLPP GPT event B / NONSAFTY GPT event B input	R/W
18	BSELCC	Event Source GTCCRB Input Capture Enable 0: Disables GTCCRB input capture by the LLPP GPT event C / NONSAFTY GPT event C input 1: Enables GTCCRB input capture by the LLPP GPT event C / NONSAFTY GPT event C input	R/W
19	BSELCD	Event Source GTCCRB Input Capture Enable 0: Disables GTCCRB input capture by the LLPP GPT event D / NONSAFTY GPT event D input 1: Enables GTCCRB input capture by the LLPP GPT event D / NONSAFTY GPT event D input	R/W
20	BSELCE	Event Source GTCCRB Input Capture Enable 0: Disables GTCCRB input capture by the LLPP GPT event E / NONSAFTY GPT event E input 1: Enables GTCCRB input capture by the LLPP GPT event E / NONSAFTY GPT event E input	R/W
21	BSELCF	Event Source GTCCRB Input Capture Enable 0: Disables GTCCRB input capture by the LLPP GPT event F / NONSAFTY GPT event F input 1: Enables GTCCRB input capture by the LLPP GPT event F / NONSAFTY GPT event F input	R/W
22	BSELCG	Event Source GTCCRB Input Capture Enable 0: Disables GTCCRB input capture by the LLPP GPT event G / NONSAFTY GPT event G input 1: Enables GTCCRB input capture by the LLPP GPT event G / NONSAFTY GPT event G input	R/W

Bit	Symbol	Function	R/W
23	BSELCH	Event Source GTCCRB Input Capture Enable 0: Disables GTCCRB input capture by the LLPP GPT event H / NONSAFTY GPT event H input 1: Enables GTCCRB input capture by the LLPP GPT event H / NONSAFTY GPT event H input	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

19.2.12 GTCR : General PWM Timer Control Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x2C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SWMD[2:0]			—	—	TPCS[3:0]				—	—	—	—	MD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ICDS	—	—	—	—	—	—	—	CST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CST	Count Start The start/stop of the counter can also be set from other registers (GTSTR, GTSSR, GTSTP, GTPSR) and the counter status is reflected. 0: Count operation is stopped 1: Count operation is started	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	ICDS	Input Capture Operation Select at Count Stop 0: Input capture is operated during count stop 1: Input capture is not operated during count stop	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
18:16	MD[2:0]	Mode Select 0 0 0: Sawtooth-wave PWM mode (single buffer or double buffer possible) 0 0 1: Sawtooth-wave one-shot pulse mode (fixed buffer operation) 0 1 0: Setting prohibited 0 1 1: Setting prohibited 1 0 0: Triangle-wave PWM mode 1 (32-bit transfer at trough) (single buffer or double buffer possible) 1 0 1: Triangle-wave PWM mode 2 (32-bit transfer at crest and trough) (single buffer or double buffer possible) 1 1 0: Triangle-wave PWM mode 3 (64-bit transfer at trough) (fixed buffer operation) 1 1 1: Setting prohibited	R/W
22:19	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
26:23	TPCS[3:0]	Timer Prescaler Select 0x0: Core Clock/1 0x1: Core Clock/2 0x2: Core Clock/4 0x3: Core Clock/8 0x4: Core Clock/16 0x5: Core Clock/32 0x6: Core Clock/64 0x7: Setting prohibited 0x8: Core Clock/256 0x9: Setting prohibited 0xA: Core Clock/1024 0xB: Setting prohibited 0xC: GTETRGA, GTETRGSA for SAFETY group (using the POEG) 0xD: GTETRGB, GTETRGSB for SAFETY group (using the POEG) 0xE: GTETRGC (using the POEG) only LLPP and NONSAFTY group 0xF: GTETRGD (using the POEG) only LLPP and NONSAFTY group	R/W
28:27	—	These bits are read as 0. The write value should be 0.	R/W
31:29	SWMD[2:0]	Switch Mode Select 0 0 0: Hardware switch operation is disabled 0 0 1: Switch to upcount with offset reload 0 1 0: Switch to downcount with offset reload 0 1 1: Setting prohibited 1 0 0: Setting prohibited 1 0 1: Switch to upcount without offset reload 1 1 0: Switch to downcount without offset reload 1 1 1: Setting prohibited	R/W

19.2.13 GTUDDTYC : General PWM Timer Count Direction and Duty Setting Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	OABDTYT	OBDTYR	OBDTYF	OBDTY[1:0]		—	—	—	—	OADTYR	OADTYF	OADTY[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	UDF	UD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	UD	Count Direction Setting 0: GTCNT counts down 1: GTCNT counts up	R/W
1	UDF	Forcible Count Direction Setting 0: Count direction is not forcibly set 1: Count direction is forcibly set	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
17:16	OADTY[1:0]	GTIOCnA Pin Output Duty Setting 0 x: Compare matches determine the duty cycle of the output on the GTIOCnA pin 1 0: The duty cycle of the output on the GTIOCnA pin is 0% 1 1: The duty cycle of the output on the GTIOCnA pin is 100%	R/W

Bit	Symbol	Function	R/W
18	OADTYF	GTIOCnA Pin Output Duty Forced Setting 0: Duty of the GTIOCnA pin output is not forcibly set 1: Duty of the GTIOCnA pin output is forcibly set	R/W
19	OADTYR	Output after Release of GTIOCnA Pin Output 0%/100% Duty Cycle Settings 0: The function selected by the GTIOR.GTIOA[3:2] bits is applied to the output value when the duty cycle is set after release from the 0 or 100% duty-cycle setting 1: The function selected by the GTIOR.GTIOA[3:2] bits is applied to the compare match output value which is masked after release from the 0 or 100% duty-cycle setting	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
25:24	OBDTY[1:0]	GTIOCnB Pin Output Duty Setting 0 x: Compare matches determine the duty cycle of the output on the GTIOCnB pin 1 0: The duty cycle of the output on the GTIOCnB pin is 0% 1 1: The duty cycle of the output on the GTIOCnB pin is 100%	R/W
26	OBDTYF	GTIOCnB Pin Output Duty Forced Setting 0: Duty of the GTIOCnB pin output is not forcibly set 1: Duty of the GTIOCnB pin output is forcibly set	R/W
27	OBDTYR	Output after Release of GTIOCnB Pin Output 0%/100% Duty Cycle Settings 0: The function selected by the GTIOR.GTIOB[3:2] bits is applied to the output value when the duty cycle is set after release from the 0 or 100% duty-cycle setting 1: The function selected by the GTIOR.GTIOB[3:2] bits is applied to the compare match output value which is masked after release from the 0 or 100% duty-cycle setting	R/W
28	OABDTYT	GTUDDTYC.OADTY[1:0] and GTUDDTYC.OBDTY[1:0] reflect timing setting in the triangle-wave mode Setting this bit to 1 is prohibited except in triangle-wave mode. 0: Reflect GTUDDTYC.OADTY[1:0] and GTUDDTYC.OBDTY[1:0] setting at trough of triangle-wave 1: Reflect GTUDDTYC.OADTY[1:0] and GTUDDTYC.OBDTY[1:0] setting at trough and crest of triangle-wave	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

19.2.14 GTIOR : General PWM Timer I/O Control Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x34

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFCSB[1:0]		NFBEN	—	—	OBDF[1:0]		OBE	OBHLD	OBDFLT	—	GTIOB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	NFCSA[1:0]		NFAEN	—	—	OADF[1:0]		OAE	OAHLD	OADFLT	—	GTIOA[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	GTIOA[4:0]	GTIOCnA Pin Function Select See Table 19.4.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	OADFLT	GTIOCnA Pin Output Value Setting at the Count Stop 0: The GTIOCnA pin outputs low when counting is stopped 1: The GTIOCnA pin outputs high when counting is stopped	R/W

Bit	Symbol	Function	R/W
7	OAHLD	GTIOCnA Pin Output Retention at the Start/Stop Count 0: The GTIOCnA pin output level at start/stop of counting depends on the register setting 1: The GTIOCnA pin output level is retained at start/stop of counting	R/W
8	OAE	GTIOCnA Pin Output Enable 0: Pin output is disabled 1: Pin output is enabled	R/W
10:9	OADF[1:0]	GTIOCnA Pin Negate Value Setting 0 0: None of the following sources is specified 0 1: GTIOCnA pin is placed in the Hi-Z state in response to control for output negation 1 0: GTIOCnA pin is set to 0 in response to control for output negation 1 1: GTIOCnA pin is set to 1 in response to control for output negation	R/W
12:11	—	These bits are read as 0. The write value should be 0.	R/W
13	NFAEN	GTIOCnA Pin Input Noise Filter Enable 0: The noise filter for GTIOCnA pin input is disabled 1: The noise filter for GTIOCnA pin input is enabled	R/W
15:14	NFCSA[1:0]	GTIOCnA Pin Input Noise Filter Sampling Clock Select 0 0: Core clock 0 1: Core clock/4 1 0: Core clock/16 1 1: Core clock/64	R/W
20:16	GTIOB[4:0]	GTIOCnB Pin Function Select See Table 19.4.	R/W
21	—	This bit is read as 0. The write value should be 0.	R/W
22	OBDFLT	GTIOCnB Pin Output Value Setting at the Count Stop 0: The GTIOCnB pin outputs low when counting is stopped 1: The GTIOCnB pin outputs high when counting is stopped	R/W
23	OBHLD	GTIOCnB Pin Output Retention at the Start/Stop Count 0: The GTIOCnB pin output level at start/stop of counting depends on the register setting 1: The GTIOCnB pin output level is retained at start/stop of counting	R/W
24	OBE	GTIOCnB Pin Output Enable 0: Pin output is disabled 1: Pin output is enabled	R/W
26:25	OBDF[1:0]	GTIOCnB Pin Negate Value Setting 0 0: None of the following sources is specified 0 1: GTIOCnB pin is placed in the Hi-Z state in response to control for output negation 1 0: GTIOCnB pin is set to 0 in response to control for output negation 1 1: GTIOCnB pin is set to 1 in response to control for output negation	R/W
28:27	—	These bits are read as 0. The write value should be 0.	R/W
29	NFBEN	GTIOCnB Pin Input Noise Filter Enable 0: The noise filter for GTIOCnB pin input is disabled 1: The noise filter for GTIOCnB pin input is enabled	R/W
31:30	NFCSB[1:0]	GTIOCnB Pin Input Noise Filter Sampling Clock Select 0 0: Core clock 0 1: Core clock/4 1 0: Core clock/16 1 1: Core clock/64	R/W

Table 19.4 Settings of GTIOA[4:0] bits, GTIOB[4:0] bits

GTIOA/GTIOB bits					Function		
[4]	[3]	[2]	[1]	[0]	bit [4]	bits [3:2]	bits [1:0]
0	0	0	0	0	Initial output is low	Output retained at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
0	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	0	1	0			High output at GTCCRA/GTCCRB compare match
0	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	0	1	0	0		Low output at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
0	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	0	1	1	0			High output at GTCCRA/GTCCRB compare match
0	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	0	0	0		High output at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
0	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	0	1	0			High output at GTCCRA/GTCCRB compare match
0	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
0	1	1	0	0		Output toggled at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
0	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
0	1	1	1	0			High output at GTCCRA/GTCCRB compare match
0	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	0	0	0	0	Initial output is high	Output retained at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
1	0	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	0	1	0			High output at GTCCRA/GTCCRB compare match
1	0	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	0	1	0	0		Low output at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
1	0	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	0	1	1	0			High output at GTCCRA/GTCCRB compare match
1	0	1	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	0	0	0		High output at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
1	1	0	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	0	1	0			High output at GTCCRA/GTCCRB compare match
1	1	0	1	1			Output toggled at GTCCRA/GTCCRB compare match
1	1	1	0	0		Output toggled at the end of a cycle	Output retained at GTCCRA/GTCCRB compare match
1	1	1	0	1			Low output at GTCCRA/GTCCRB compare match
1	1	1	1	0			High output at GTCCRA/GTCCRB compare match
1	1	1	1	1			Output toggled at GTCCRA/GTCCRB compare match

Note:
- In sawtooth-wave mode, "end of a cycle" refers to an overflow (the value of the GTCNT counter changing from that of the GTPR register to 0x00000000 in up-counting), an underflow (the value of the GTCNT counter changing from 0x00000000 to that of the GTPR register in down-counting), or counter clearing. It refers to a trough in triangle-wave mode (the value of the GTCNT counter changing from 0x00000000 to 0x00000001).
- When the timing of an end of a cycle and the timing of a GTCCRA/GTCCRB register compare match are the same in a compare-match operation, the bit 3 and bit 2 settings are given priority in sawtooth-wave PWM mode, and the bit 1 and bit 0 settings are given priority in any other mode.
- During the event count operation (when at least one bit among the bits for the GTUPSR and GTDNSR registers is set to 1), bit 3 and bit 2 settings become invalid.

19.2.15 GTINTAD : General PWM Timer Interrupt Output Setting Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x38

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	GRPABL	GRPABH	GRPDTE	—	—	GRP[1:0]		—	—	—	—	ADTRBDEN	ADTRBUEN	ADTRADEN	ADTRAUEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	GTINTPR[1:0]		GTINTF	GTINTE	GTINTD	GTINTC	GTINTB	GTINTA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	GTINTA	GTCCRA Register Compare Match / Input Capture Interrupt Enable 0: Interrupt request is disabled 1: Interrupt request is enabled	R/W
1	GTINTB	GTCCRB Register Compare Match / Input Capture Interrupt Enable 0: Interrupt request is disabled 1: Interrupt request is enabled	R/W
2	GTINTC	GTCCRC Register Compare Match Interrupt Enable 0: Interrupt request is disabled 1: Interrupt request is enabled	R/W
3	GTINTD	GTCCRD Register Compare Match Interrupt Enable 0: Interrupt request is disabled 1: Interrupt request is enabled	R/W
4	GTINTE	GTCCRE Register Compare Match Interrupt Enable 0: Interrupt request is disabled 1: Interrupt request is enabled	R/W
5	GTINTF	GTCCRF Register Compare Match Interrupt Enable 0: Interrupt request is disabled 1: Interrupt request is enabled	R/W
7:6	GTINTPR[1:0]	GTPR Register Compare Match Interrupt Enable 0 0: Interrupt request is disabled 0 1: In sawtooth-wave mode, interrupt requests are enabled at overflows. In triangle-wave mode, interrupt requests are enabled at crests 1 0: In sawtooth-wave mode, interrupt requests are enabled at underflows. In triangle-wave mode, interrupt requests are enabled at troughs 1 1: In sawtooth-wave mode, interrupt requests are enabled at both overflows and underflows. In triangle-wave mode, interrupt requests are enabled at both crests and troughs	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
16	ADTRAUEN	GTADTRA Register Compare Match (Up-Counting) A/D Converter Start Request Enable 0: A/D converter start request is disabled 1: A/D converter start request is enabled	R/W
17	ADTRADEN	GTADTRA Register Compare Match (Down-Counting) A/D Converter Start Request Enable 0: A/D converter start request is disabled 1: A/D converter start request is enabled	R/W
18	ADTRBUEN	GTADTRB Register Compare Match (Up-Counting) A/D Converter Start Request Enable 0: A/D converter start request is disabled 1: A/D converter start request is enabled	R/W
19	ADTRBDEN	GTADTRB Register Compare Match (Down-Counting) A/D Converter Start Request Enable 0: A/D converter start request is disabled 1: A/D converter start request is enabled	R/W

Bit	Symbol	Function	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
25:24	GRP[1:0]	Output Stop Group Select Select the group to detect disabling of output (dead-time error or simultaneous driving of outputs to the high or low level) to POEG and to request of disabling of output from POEG. 0 0: Group A is selected 0 1: Group B is selected 1 0: Group C is selected 1 1: Group D is selected	R/W
27:26	—	These bits are read as 0. The write value should be 0.	R/W
28	GRPDTE	Dead Time Error Output Stop Detection Enable 0: Dead time error output stop detection is disabled 1: Dead time error output stop detection is enabled	R/W
29	GRPABH	Simultaneous High Output Stop Detection Enable (GTIOCnA pin and GTIOCnB output) 0: Simultaneous high output stop detection is disabled 1: Simultaneous high output stop detection is enabled	R/W
30	GRPABL	Simultaneous Low Output Stop Detection Enable (GTIOCnA pin and GTIOCnB output) 0: Simultaneous low output stop detection is disabled 1: Simultaneous low output stop detection is enabled	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

19.2.16 GTST : General PWM Timer Status Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x3C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	OABLF	OABHF	DTEF	—	—	—	ODF	—	—	—	—	ADTRBDF	ADTRBUF	ADTRADF	ADTRAUF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TUCF	—	—	—	—	ITCNT[2:0]			—	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
10:8	ITCNT[2:0]	GPTn_OVF/GPTn_UDF Interrupt Skipping Count Counter Counter for counting the number of times a timer interrupt has been skipped. The counter is incremented by 1 every time the interrupt source which is selected in GTITC.IVTC[1:0] is generated. Cleared when the interrupt skip count matches the specified count (GTITC.IVTT[2:0]) or when the count operation has stopped.	R
14:11	—	These bits are read as 0. The write value should be 0.	R/W
15	TUCF	Count Direction Flag 0: The GTCNT counter counts downward 1: The GTCNT counter counts upward	R
16	ADTRAUF	GTADTRA Register Compare Match (Up-Counting) A/D Converter Start Request Flag 0: No GTADTRA register compare match has occurred in up-counting 1: A GTADTRA register compare match has occurred in up-counting	R/W*1

Bit	Symbol	Function	R/W
17	ADTRADF	GTADTRA Register Compare Match (Down-Counting) A/D Converter Start Request Flag 0: No GTADTRA register compare match has occurred in down-counting 1: A GTADTRA register compare match has occurred in down-counting	R/W*1
18	ADTRBUF	GTADTRB Register Compare Match (Up-Counting) A/D Converter Start Request Flag 0: No GTADTRB register compare match has occurred in up-counting 1: A GTADTRB register compare match has occurred in up-counting	R/W*1
19	ADTRBDF	GTADTRB Register Compare Match (Down-Counting) A/D Converter Start Request Flag 0: No GTADTRB register compare match has occurred in down-counting 1: A GTADTRB register compare match has occurred in down-counting	R/W*1
23:20	—	These bits are read as 0. The write value should be 0.	R/W
24	ODF	Output Stop Request Flag This flag monitors the output stop request for the group selected by GTINTAD.GRP[1:0] bits. 0: No output stop request has occurred 1: An output stop request has occurred	R
27:25	—	These bits are read as 0. The write value should be 0.	R/W
28	DTEF	Dead Time Error Flag This flag returns to 0 when the timer output toggle point after the automatic addition of dead time is back within the period. 0: No dead time error has occurred 1: A dead time error has occurred	R
29	OABHF	Simultaneous High Output Flag When one of pins changes to 0, the flag changes to 0. 0: No simultaneous generation of high output both for the GTIOCA and GTIOCB pins has occurred 1: A simultaneous generation of high output both for the GTIOCA and GTIOCB pins has occurred	R
30	OABLF	Simultaneous Low Output Flag When one of pins changes to 1, the flag changes to 0. 0: No simultaneous generation of low output both for the GTIOCA and GTIOCB pins has occurred 1: A simultaneous generation of low output both for the GTIOCA and GTIOCB pins has occurred	R
31	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to clear the flag. When clearing the ADTRAUF, ADTRADF, ADTRBUF, or ADTRBDF flag, be sure to write 0 only to the target flag or flags for clearing and to write 1 to the other flags not for clearing.

19.2.17 GTBER : General PWM Timer Buffer Enable Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x40

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	ADTDB	ADTTB[1:0]		—	ADTDA	ADTTA[1:0]		—	CCRSWT	PR[1:0]		CCRB[1:0]		CCRA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	DBRTECB	—	DBRTECA	—	—	—	—	BD3	BD2	BD1	BD0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BD0	GTCCRA/GTCCRB Registers Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W

Bit	Symbol	Function	R/W
1	BD1	GTPR Register Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
2	BD2	GTADTRA/GTADTRB Registers Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
3	BD3	GTDVU/GTDVD Registers Buffer Operation Disable 0: Buffer operation is enabled 1: Buffer operation is disabled	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	DBRTECA	GTCCRA Register Double Buffer Repeat Operation Enable 0: GTCCRA register double buffer repeat operation is disabled 1: GTCCRA register double buffer repeat operation is enabled	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
10	DBRTECB	GTCCRB Register Double Buffer Repeat Operation Enable 0: GTCCRB register double buffer repeat operation is disabled 1: GTCCRB register double buffer repeat operation is enabled	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
17:16	CCRA[1:0]	GTCCRA Register Buffer Operation 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTCCRA register ↔ GTCCRC register) 1 x: Double buffer operation (GTCCRA register ↔ GTCCRC register ↔ GTCCRD register)	R/W
19:18	CCRB[1:0]	GTCCRB Register Buffer Operation 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTCCRB register ↔ GTCCRE register) 1 x: Double buffer operation (GTCCRB register ↔ GTCCRE register ↔ GTCCRF register)	R/W
21:20	PR[1:0]	GTPR Register Buffer Operation 0 0: Buffer operation is not performed 0 1: Single buffer operation (GTPBR register → GTPR register) 1 x: Double buffer operation (GTPDBR register → GTPBR register → GTPR register)	R/W
22	CCRSWT	GTCCRA and GTCCRB Registers Forcible Buffer Operation Writing 1 to this bit forcibly performs buffer transfer of GTCCRA and GTCCRB registers. This bit automatically returns to 0 after the writing of 1. This bit is read as 0.	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
25:24	ADTTA[1:0]	GTADTRA Register Buffer Transfer Timing Select 0 0: No transfer 0 1: (Triangle waves) Transfer at crest (Sawtooth waves) Transfer at underflow (during down-counting), overflow (during up-counting), or counter clearing 1 0: (Triangle waves) Transfer at trough (Sawtooth waves) Transfer at underflow (during down-counting), overflow (during up-counting), or counter clearing 1 1: (Triangle waves) Transfer at both crest and trough (Sawtooth waves) Transfer at underflow (during down-counting), overflow (during up-counting), or counter clearing	R/W
26	ADTDA	GTADTRA Register Double Buffer Operation 0: Single buffer operation (GTADTBRA register → GTADTRA register) 1: Double buffer operation (GTADTDBRA register → GTADTBRA register → GTADTRA register)	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
29:28	ADTTB[1:0]	GTADTRB Register Buffer Transfer Timing Select 0 0: No transfer 0 1: (Triangle waves) Transfer at crest (Sawtooth waves) Transfer at underflow (during down-counting), overflow (during up-counting), or counter clearing 1 0: (Triangle waves) Transfer at trough (Sawtooth waves) Transfer at underflow (during down-counting), overflow (during up-counting), or counter clearing 1 1: (Triangle waves) Transfer at both crest and trough (Sawtooth waves) Transfer at underflow (during down-counting), overflow (during up-counting), or counter clearing	R/W
30	ADTDB	GTADTRB Register Double Buffer Operation 0: Single buffer operation (GTADTBRB register → GTADTRB register) 1: Double buffer operation (GTADTDBRB register → GTADTBRB register → GTADTRB register)	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

Note: The buffer operation mode is fixed in sawtooth-wave one-shot pulse mode or triangle-wave PWM mode 3 (64-bit transfer at trough).

19.2.18 GTITC : General PWM Timer Interrupt and A/D Converter Start Request Skipping Setting Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x44

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ADTBL	—	ADTAL	—	IVTT[2:0]			IVTC[1:0]		ITLF	ITLE	ITLD	ITLC	ITLB	ITLA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ITLA	GTCCRA Register Compare Match / Input Capture Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function	R/W
1	ITLB	GTCCRB Register Compare Match / Input Capture Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function	R/W
2	ITLC	GTCCRC Register Compare Match Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function	R/W
3	ITLD	GTCCRD Register Compare Match Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function	R/W
4	ITLE	GTCCRE Register Compare Match Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function	R/W
5	ITLF	GTCCRF Register Compare Match Interrupt Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function	R/W

Bit	Symbol	Function	R/W
7:6	IVTC[1:0]	GPTn_OVF/GPTn_UDF Interrupt Skipping Function Select 0 0: Skipping is not performed 0 1: Both overflow and underflow for sawtooth waves and crest for triangle waves are counted and skipped 1 0: Both overflow and underflow for sawtooth waves and trough for triangle waves are counted and skipped 1 1: Both overflow and underflow for sawtooth waves and both crest and trough for triangle waves are counted and skipped	R/W
10:8	IVTT[2:0]	GPTn_OVF/GPTn_UDF Interrupt Skipping Count Select 0 0 0: Skipping is not performed 0 0 1: Skipping count of 1 0 1 0: Skipping count of 2 0 1 1: Skipping count of 3 1 0 0: Skipping count of 4 1 0 1: Skipping count of 5 1 1 0: Skipping count of 6 1 1 1: Skipping count of 7	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	ADTAL	GTADTRA Register A/D Converter Start Request Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W
14	ADTBL	GTADTRB Register A/D Converter Start Request Link 0: Not linked with GPTn_OVF/GPTn_UDF interrupt skipping function 1: Linked with GPTn_OVF/GPTn_UDF interrupt skipping function	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

19.2.19 GTCNT : General PWM Timer Counter

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x48

Bit position:	31																													0
Bit field:																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	The GTCNT counter is a 32-bit readable/writable counter. There is one GTCNT counter for each channel. Writing is available only when counting is stopped, and thus writing is disabled while counting is in progress (when CST = 1). Access in 8-bit or 16-bit units to the GTCNT counter is prohibited, and it should be accessed in 32-bit units. Set the range of the GTCNT counter within the range of 0 ≤ GTCNT counter ≤ GTPR register.	R/W

19.2.20 GTCCRm : General PWM Timer Compare Capture Register m (m = A to F)

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x4C (GTCCRA), 0x50 (GTCCRB), 0x54 (GTCCRC)
0x58 (GTCCRE), 0x5C (GTCCRD), 0x60 (GTCCRF)

Bit position:	31																													0
Bit field:																														
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The GTCCRm register is a 32-bit readable/writable register. Access in 8-bit or 16-bit units to the GTCCRm register is prohibited, and it should be accessed in 32-bit units. The GTCCRA and GTCCRB registers are used for both output compare and input capture. The GTCCRC and GTCCRE registers are compare match registers, and can also function as buffer registers for the GTCCRA and GTCCRB registers. The GTCCRD and GTCCRF registers are compare match registers, and can also function as buffer registers for the GTCCRC and GTCCRE registers (double buffer registers for the GTCCRA and GTCCRB registers).	R/W

19.2.21 GTPR : General PWM Timer Cycle Setting Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x64

Bit position:	31																													0
Bit field:																														
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The GTPR register is a 32-bit readable/writable register that sets the maximum count value of the GTCNT counter. Access in 8-bit or 16-bit units to the GTPR register is prohibited, and it should be accessed in 32-bit units. In sawtooth-wave mode, the value of (GTPR register + 1) is the count period. In triangle-wave mode, the value of (GTPR register value × 2) is the count period.	R/W

19.2.22 GTPBR : General PWM Timer Cycle Setting Buffer Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x68

Bit position:	31																													0
Bit field:																														
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The GTPBR register is a 32-bit readable/writable register that functions as a buffer register for the GTPR register. Access in 8-bit or 16-bit units to the GTPBR register is prohibited, and it should be accessed in 32-bit units.	R/W

19.2.23 GTPDBR : General PWM Timer Cycle Setting Double-Buffer Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x6C

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The GTPDBR register is a 32-bit readable/writable register that functions as a buffer register for the GTPBR register (a double buffer register for the GTPR register). Access in 8-bit or 16-bit units to the GTPDBR register is prohibited, and it should be accessed in 32-bit units.	R/W

19.2.24 GTADTRm : A/D Converter Start Request Timing Register m (m = A, B)

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)

Offset address: 0x70 (GTADTRA)
0x7C (GTADTRB)

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The GTADTRm register is a 32-bit readable/writable register that sets the timing of A/D converter start request generation. Access in 8-bit or 16-bit units to the GTADTRm register is prohibited, and it should be accessed in 32-bit units. When the GTADTRm register value matches the GTCNT counter value, an A/D converter start request is generated.	R/W

SAFTY has not A/D Converter Start request.

19.2.25 GTADTBRm : A/D Converter Start Request Timing Buffer Register m (m = A, B)

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)

Offset address: 0x74 (GTADTBRA)
0x80 (GTADTBRB)

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The GTADTBRm register is a 32-bit readable/writable register that functions as buffer register for the GTADTRm register. Access in 8-bit or 16-bit units to the GTADTBRm register is prohibited, and it should be accessed in 32-bit units.	R/W

SAFTY has not A/D Converter Start request.

19.2.26 GTADTDBRm : A/D Converter Start Request Timing Double-Buffer Register m (m = A, B)

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)

Offset address: 0x78 (GTADTDBRA)
0x84 (GTADTDBRB)

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The GTADTDBRm register is a 32-bit readable/writable register that functions as a buffer register for the GTADTBRm register (double buffer register for the GTADTRm register). Access in 8-bit or 16-bit units to the GTADTDBRm register is prohibited, and it should be accessed in 32-bit units.	R/W

SAFTY has not A/D Converter Start request.

19.2.27 GTDTCR : General PWM Timer Dead Time Control Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x88

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	TDFER	—	—	TDBDE	TDBUE	—	—	—	TDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TDE	Negative-Phase Waveform Setting 0: GTCCRB register is set without using GTDVU and GTDVD registers 1: GTDVU and GTDVD registers are used to set the compare match value for negative-phase waveform with dead time automatically in GTCCRB register	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	TDBUE	GTDVU Register Buffer Operation Enable 0: GTDVU register buffer operation is disabled 1: GTDVU register buffer operation is enabled	R/W
5	TDBDE	GTDVD Register Buffer Operation Enable 0: GTDVD register buffer operation is disabled 1: GTDVD register buffer operation is enabled	R/W

Bit	Symbol	Function	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	TDFER	GTDVD Register Setting 0: GTDVU and GTDVD registers are set separately 1: The value written to GTDVU register is automatically set to GTDVD register	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The GTDTCR register enables automatic setting of a compare match value for negative-phase waveform with dead time. This register is invalid during the event count operation.

19.2.28 GTDVm : General PWM Timer Dead Time Value Register m (m = U, D)

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x8C (GTDVU)
0x90 (GTDVD)

Bit position:	31																													0
Bit field:																														
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The GTDVm register is a 32-bit readable/writable register that sets the dead time for generating PWM waveforms with dead time. Access in 8-bit or 16-bit units to the GTDVm register is prohibited, and it should be accessed in 32-bit units. The GTDVU register is used for up-counting. The GTDVD register is used for down-counting. The setting which the value for the GTDVm register is equal to or larger than the value for the GTPR register is prohibited. In addition, when using the automatic dead time setting function, do not set a value that makes a change point of the waveform exceeding the count period. The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. When using the GTDVm register, writing to the GTCCRB register is prohibited. When setting this register to 0, waveforms without dead time are output. When the GTDTCR.TDFER bit is 1, the writing to the GTDVD register has no effect. At this time, when the GTDVD register is read, the value for the GTDVU register is read.	R/W

19.2.29 GTDBm : General PWM Timer Dead Time Value Buffer Register m (m = U, D)

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x94 (GTDBU)
0x98 (GTDBD)

Bit position:	31																													0
Bit field:																														
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The GTDBm register is a 32-bit readable/writable register that functions as a buffer register for the GTDVm register. Access in 8-bit or 16-bit units to the GTDBm register is prohibited, and it should be accessed in 32-bit units.	R/W

19.2.30 GTSOS : General PWM Timer Output Protection Function Status Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0x9C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOS[1:0]	
Value after reset:	0	0	0	0	0	0	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SOS[1:0]	Output Protection Function Status 0 0: Normal operation 0 1: Protected state (GTCCRA = 0 is set during transfer at trough or crest) 1 0: Protected state (GTCCRA ≥ GTPR is set during transfer at trough) 1 1: Protected state (GTCCRA ≥ GTPR is set during transfer at crest)	R
31:2	—	These bits are read as 0.	R

The GTSOS register is a status register that indicates the status of the output protection function. The output protection function is enabled only when the dead time is automatically set (the GTDTCR.TDE bit is 1) in triangle-wave mode.

19.2.31 GTSOTR : General PWM Timer Output Protection Function Temporary Release Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0xA0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOTR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SOTR	Output Protection Function Temporary Release When the output is protected, the GTIOCnB pin output is temporarily released from the protected state. The protected state can be released only for the case of GTSOS.SOS[1:0] bits are 10b (protected state in which GTCCRA ≥ GTPR has occurred during transfer at trough). 0: Protected state is not released 1: Protected state is released	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

19.2.32 GTADSMR : General PWM Timer A/D Conversion Start Request Signal Monitoring Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)

Offset address: 0xA4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	ADSMEN1	—	—	—	—	—	—	ADSMS1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ADSMEN0	—	—	—	—	—	—	ADSMS0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ADSMS0[1:0]	A/D Conversion Start Request Signal Monitor 0 Selection 0 0: A/D conversion start request signal generated by the GTADTRA register during up-counting 0 1: A/D conversion start request signal generated by the GTADTRA register during down-counting 1 0: A/D conversion start request signal generated by the GTADTRB register during up-counting 1 1: A/D conversion start request signal generated by the GTADTRB register during down-counting	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	ADSMEN0	A/D Conversion Start Request Signal Monitor 0 Output Enabling 0: Output of A/D conversion start request signal monitor 0 is disabled 1: Output of A/D conversion start request signal monitor 0 is enabled	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
17:16	ADSMS1[1:0]	A/D Conversion Start Request Signal Monitor 1 Selection 0 0: A/D conversion start request signal generated by the GTADTRA register during up-counting 0 1: A/D conversion start request signal generated by the GTADTRA register during down-counting 1 0: A/D conversion start request signal generated by the GTADTRB register during up-counting 1 1: A/D conversion start request signal generated by the GTADTRB register during down-counting	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
24	ADSMEN1	A/D Conversion Start Request Signal Monitor 1 Output Enabling 0: Output of A/D conversion start request signal monitor 1 is disabled 1: Output of A/D conversion start request signal monitor 1 is enabled	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

SAFTY has not A/D conversion Start Request.

The GTADSMR register is used to control monitors for the A/D conversion start request signal that is synchronized with a frame period.

When the output of the same A/D converter start request signal monitoring output is enabled for multiple channels, ORed signals will be output.

19.2.33 GTEITC : General PWM Timer Extended Interrupt Skipping Counter Control Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0xA8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EITCNT2[3:0]				EITCNT2IV[3:0]				EIVTT2[3:0]				—	—	EIVTC2[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EITCNT1[3:0]				—	—	—	—	EIVTT1[3:0]				—	—	EIVTC1[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	EIVTC1[1:0]	Extended Interrupt Skipping Counter 1 Count Source Select 0 0: Not counted (not skipped) 0 1: Counting both at overflow or underflow in sawtooth-wave mode and counting crests in triangle-wave mode 1 0: Counting both at overflow or underflow in sawtooth-wave mode and counting troughs in triangle-wave mode 1 1: Counting both at overflow or underflow in sawtooth-wave mode and counting both crests and troughs in triangle-wave mode	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	EIVTT1[3:0]	Extended Interrupt Skipping 1 Skipping Count Setting Skipping count for the extended interrupt skipping 1	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
15:12	EITCNT1[3:0]	Extended Interrupt Skipping Counter 1 The counting is incremented by 1 every time a count source selected by the EIVTC1[1:0] bits is generated. When the count source is generated while the EIVTT1[3:0] bits match, these bits are cleared.	R
17:16	EIVTC2[1:0]	Extended Interrupt Skipping Counter 2 Count Source Select 0 0: Not counted (not skipped) 0 1: Counting both at overflow or underflow in sawtooth-wave mode and counting crests in triangle-wave mode 1 0: Counting both at overflow or underflow in sawtooth-wave mode and counting troughs in triangle-wave mode 1 1: Counting both at overflow or underflow in sawtooth-wave mode and counting both crests and troughs in triangle-wave mode	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
23:20	EIVTT2[3:0]	Extended Interrupt Skipping 2 Skipping Count Setting	R/W
27:24	EITCNT2IV[3:0]	Extended Interrupt Skipping Counter 2 Initial Value The EITCNT2IV[3:0] bits are only writable when a value other than 00b is written to the EIVTC2[1:0] bits that are 00b.	R/W
31:28	EITCNT2[3:0]	Extended Interrupt Skipping Counter 2 The counting is incremented by 1 every time a count source selected by the EIVTC2[1:0] bits is generated. When the count source is generated while the EIVTT2[3:0] bits match, these bits are cleared.	R

19.2.34 GTEITLI1 : General PWM Timer Extended Interrupt Skipping Setting Register 1

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0xAC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	EITLU[2:0]			—	EITLV[2:0]			—	EITLF[2:0]			—	EITLE[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	EITLD[2:0]			—	EITLC[2:0]			—	EITLB[2:0]			—	EITLA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	EITLA[2:0]	GTCCRA Register Compare Match / Input Capture Interrupt Extended Skipping Function Select See Table 19.5.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	EITLB[2:0]	GTCCRB Register Compare Match / Input Capture Interrupt Extended Skipping Function Select See Table 19.5.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	EITLC[2:0]	GTCCRC Register Compare Match / Input Capture Interrupt Extended Skipping Function Select See Table 19.5.	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
14:12	EITLD[2:0]	GTCCRD Register Compare Match / Input Capture Interrupt Extended Skipping Function Select See Table 19.5.	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
18:16	EITLE[2:0]	GTCCRE Register Compare Match / Input Capture Interrupt Extended Skipping Function Select See Table 19.5.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	EITLF[2:0]	GTCCRF Register Compare Match / Input Capture Interrupt Extended Skipping Function Select See Table 19.5.	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
26:24	EITLV[2:0]	Overflow Interrupt Extended Skipping Function Select See Table 19.5.	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
30:28	EITLU[2:0]	Underflow Interrupt Extended Skipping Function Select See Table 19.5.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The GTEITLI1 register sets the extended skipping function for interrupts such as compare match/input capture, overflow, and underflow.

Setting only with this register does not perform skipping. The GTEITC register should be set so that a corresponding extended interrupt skipping counter operates the counting.

The setting is operated independently from the interrupt skipping by the GTITC register.

The setting is invalid during the event count operation.

Table 19.5 Setting the function select for the GTEITLI1 register

EITLy[2:0] bits	Function
000b	Do not perform an extended interrupt skipping
001b	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 is other than 0. An interrupt is output in the period of EITCNT1[3:0] bits = 0000b.
010b	Skip an interrupt in the period when the value for the extended interrupt skipping counter 2 is other than 0. An interrupt is output in the period of EITCNT2[3:0] bits = 0000b.
011b	Skip an interrupt in the period the value for the extended interrupt skipping counter 1 or 2 is other than 0. An interrupt is output in the period of EITCNT1[3:0] bits = 0000b and EITCNT2[3:0] bits = 0000b.
100b	Do not set this value
101b	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count. An interrupt is output in the period of EITCNT1[3:0] bits = EIVTT1[3:0] bits.
110b	Skip an interrupt in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count. An interrupt is output in the period of EITCNT2[3:0] bits = EIVTT2[3:0] bits.
111b	Skip an interrupt in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count. An interrupt is output in the period of EITCNT1[3:0] bits = EIVTT1[3:0] bits and EITCNT2[3:0] bits = EIVTT2[3:0] bits.

Note:
- y = A, B, C, D, E, F, V, U; k = 1, 2
- When the intended skipping counter is set as not to count (EIVTCk[1:0] bits = 00b or EIVTTk[3:0] bits = 0000b), skipping is not performed.
- When the EITLy[2:0] bits are set to 011b or 111b, or when one of the skipping counters is set as not to count, skipping is not performed.

19.2.35 GTEITLI2 : General PWM Timer Extended Interrupt Skipping Setting Register 2

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0xB0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	EADTBL[2:0]			—	EADTAL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	EADTAL[2:0]	GTADTRA A/D Converter Start Request Extended Skipping Function Select See Table 19.6.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	EADTBL[2:0]	GTADTRB A/D Converter Start Request Extended Skipping Function Select See Table 19.6.	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

The GTEITLI2 register sets the extended skipping function for A/D converter start requests.

Setting only with this register does not perform skipping. The GTEITC register should be set so that a corresponding extended interrupt skipping counter operates the counting.

The setting is operated independently from the interrupt skipping by the GTITC register.

The setting is invalid during the event count operation.

Table 19.6 Setting the function select for the GTEITLI2 register

EADTmL[2:0] bits	Function
000b	Do not perform an extended interrupt skipping
001b	Skip an A/D converter start request in the period when the value for the extended interrupt skipping counter 1 is other than 0. An A/D converter start request is output in the period of EITCNT1[3:0] bits = 0000b.
010b	Skip an A/D converter start request in the period when the value for the extended interrupt skipping counter 2 is other than 0. An A/D converter start request is output in the period of EITCNT2[3:0] bits = 0000b.
011b	Skip an A/D converter start request in the period when the value for the extended interrupt skipping counter 1 or 2 is other than 0. An A/D converter start request is output in the period of EITCNT1[3:0] bits = 0000b and EITCNT2[3:0] bits = 0000b.
100b	Do not set this value
101b	Skip an A/D converter start request in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count. An A/D converter start request is output in the period of EITCNT1[3:0] bits = EIVTT1[3:0] bits.
110b	Skip an A/D converter start request in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count. An A/D converter start request is output in the period of EITCNT2[3:0] bits = EIVTT2[3:0] bits.
111b	Skip an A/D converter start request in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count. An A/D converter start request is output in the period of EITCNT1[3:0] bits = EIVTT1[3:0] bits and EITCNT2[3:0] bits = EIVTT2[3:0] bits.

Note:
- m = A, B; k = 1, 2
- When the intended skipping counter is set as not to count (EIVTCk[1:0] bits = 00b or EIVTTk[3:0] bits = 0000b), skipping is not performed.
- When the EADTmL[2:0] bits are set to 011b or 111b, or when one of the skipping counters is set as not to count, skipping is not performed.

19.2.36 GTEITLB : General PWM Timer Extended Buffer Transfer Skipping Setting Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0xB4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	EBTLDVD[2:0]			—	EBTLDVU[2:0]			—	EBTLADB[2:0]			—	EBTLADA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	EBTLPR[2:0]			—	EBTLCB[2:0]			—	EBTLCA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	EBTLCA[2:0]	GTCCRA Register Buffer Transfer Extended Skipping Function Select See Table 19.7.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	EBTLCB[2:0]	GTCCRB Register Buffer Transfer Extended Skipping Function Select See Table 19.7.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	EBTLPR[2:0]	GTPR Register Buffer Transfer Extended Skipping Function Select See Table 19.7.	R/W

Bit	Symbol	Function	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
18:16	EBTLADA[2:0]	GTADTRA Register Buffer Transfer Extended Skipping Function Select See Table 19.7.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
22:20	EBTLADB[2:0]	GTADTRB Register Buffer Transfer Extended Skipping Function Select See Table 19.7.	R/W
23	—	This bit is read as 0. The write value should be 0.	R/W
26:24	EBTLDVU[2:0]	GTDVU Register Buffer Transfer Extended Skipping Function Select See Table 19.7.	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
30:28	EBTLDVD[2:0]	GTDVD Register Buffer Transfer Extended Skipping Function Select See Table 19.7.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The GTEITLB register sets the extended skipping function for buffer transfers.

Setting only with this register does not perform skipping. The GTEITC register should be set so that a corresponding extended interrupt skipping counter operates the counting.

The setting is operated independently from the interrupt skipping by the GTITC register.

The setting is invalid during the event count operation.

Table 19.7 Setting the function select for the GTEITLB register

EBTLx[2:0] bits	Function
000b	Do not perform an extended interrupt skipping
001b	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 is other than 0. A buffer transfer is output in the period of EITCNT1[3:0] bits = 0000b.
010b	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 2 is other than 0. A buffer transfer is output in the period of EITCNT2[3:0] bits = 0000b.
011b	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 or 2 is other than 0. A buffer transfer is output in the period of EITCNT1[3:0] bits = 0000b and EITCNT2[3:0] bits = 0000b.
100b	Do not set this value
101b	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 is other than the skipping count. A buffer transfer is output in the period of EITCNT1[3:0] bits = EIVTT1[3:0] bits.
110b	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 2 is other than the skipping count. A buffer transfer is output in the period of EITCNT2[3:0] bits = EIVTT2[3:0] bits.
111b	Skip a buffer transfer in the period when the value for the extended interrupt skipping counter 1 or 2 is other than the skipping count. A buffer transfer is output in the period of EITCNT1[3:0] bits = EIVTT1[3:0] bits and EITCNT2[3:0] bits = EIVTT2[3:0] bits.

Note:
- x = CA, CB, PR, ADA, ADB, DVU, DVD; k = 1, 2
- When the intended skipping counter is set as not to count (EIVTCk[1:0] bits = 00b or EIVTTk[3:0] bits = 0000b), skipping is not performed.
- When the EBTLx[2:0] bits are set to 011b or 111b, or when one of the skipping counters is set as not to count, skipping is not performed.

19.2.37 GTSECSR : General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0xD0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	SECSEL6	SECSEL5	SECSEL4	SECSEL3	SECSEL2	SECSEL1	SECSEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SECSEL0	Channel 0 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
1	SECSEL1	Channel 1 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
2	SECSEL2	Channel 2 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
3	SECSEL3	Channel 3 Operation Enable Bit Simultaneous Control Channel Select 0: Disable simultaneous control 1: Enable simultaneous control	R/W
4	SECSEL4	Channel 4 Operation Enable Bit Simultaneous Control Channel Select This bit is available for LLPP/NONSAFETY region. 0: Disable simultaneous control 1: Enable simultaneous control	R/W
5	SECSEL5	Channel 5 Operation Enable Bit Simultaneous Control Channel Select This bit is available for LLPP/NONSAFETY region. 0: Disable simultaneous control 1: Enable simultaneous control	R/W
6	SECSEL6	Channel 6 Operation Enable Bit Simultaneous Control Channel Select This bit is available for LLPP/NONSAFETY region. 0: Disable simultaneous control 1: Enable simultaneous control	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

The GTSECSR register selects an intended channel for updating an operation enable bit by the GTSECR register.

A bit position for the GTSECSR register indicates a channel number. The GTSECSR register of each channel is a common register, and writing 1 to a bit in the GTSECSR register in any channel and updating it changes a channel, related to the position of the bit written with 1 by the GTSECSR register, to be simultaneously controlled of the operation enable bit by the GTSECR register.

19.2.38 GTSECR : General PWM Timer Operation Enable Bit Simultaneous Control Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0xD4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	SBDDD	SBDAD	SBDPD	SBDCD	—	—	—	—	SBDDE	SBDAE	SBDPE	SBDCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SBDCE	GTCCR Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTCCR buffer operations 1: Enable GTCCR register buffer operations simultaneously	R/W
1	SBDPE	GTPR Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTPR buffer operations 1: Enable GTPR register buffer operations simultaneously	R/W
2	SBDAE	GTADTR Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTADTR buffer operations 1: Enable GTADTR register buffer operations simultaneously	R/W
3	SBDDE	GTDV Register Buffer Operation Simultaneous Enable 0: Disable simultaneous enabling GTDV buffer operations 1: Enable GTDV register buffer operations simultaneously	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	SBDCD	GTCCR Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTCCR buffer operations 1: Disable GTCCR register buffer operations simultaneously	R/W
9	SBDPD	GTPR Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTPR buffer operations 1: Disable GTPR register buffer operations simultaneously	R/W
10	SBDAD	GTADTR Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTADTR buffer operations 1: Disable GTADTR register buffer operations simultaneously	R/W
11	SBDDD	GTDV Register Buffer Operation Simultaneous Disable 0: Disable simultaneous disabling GTDV buffer operations 1: Disable GTDV register buffer operations simultaneously	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The GTSECR register simultaneously updates the value for operation enable bits of a channel set by the GTSECSR register.

Writing 1 to a bit in the GTSECR register in any channel and updating it updates an operation enable bit for all channels, related to the position of the bit written with 1 by the all GTSECSR registers. It is reflected in the GTBER.BDn (n = 0 to 3) bits of simultaneous channels. Setting enable and disable bits for the same operation enable bit to 1 in the GTSECR is prohibited.

A bit written to 1 is automatically cleared. When the GTSECR is read, 0 is read.

Access in 8-bit or 16-bit units to the GTSECR register is prohibited, and it should be accessed in 32-bit units.

19.2.39 GTSWSR : General PWM Timer Switch Source Select Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0xD8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	CSELCH	WSELCG	WSELCF	WSELCE	WSELCD	WSELCC	WSELCB	WSELCA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	WSGTRGD[1:0]		WSGTRGC[1:0]		WSGTRGB[1:0]		WSGTRGA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	WSGTRGA[1:0]	GTETRGA Signal Edge Select to Switch Counter (GTETRGSA Signal for SAFTY) 0 0: The GTETRGA signal is not used as a trigger to switch the counter 0 1: The counter is switched at a rising edge of the GTETRGA signal 1 0: The counter is switched at a falling edge of the GTETRGA signal 1 1: The counter is switched at both edges of the GTETRGA signal	R/W
3:2	WSGTRGB[1:0]	GTETRGB Signal Edge Select to Switch Counter (GTETRGSB Signal for SAFTY) 0 0: The GTETRGB signal is not used as a trigger to switch the counter 0 1: The counter is switched at a rising edge of the GTETRGB signal 1 0: The counter is switched at a falling edge of the GTETRGB signal 1 1: The counter is switched at both edges of the GTETRGB signal	R/W
5:4	WSGTRGC[1:0]	GTETRGC Signal Edge Select to Switch Counter (GTETRGSC Signal for SAFTY) 0 0: The GTETRGC signal is not used as a trigger to switch the counter 0 1: The counter is switched at a rising edge of the GTETRGC signal 1 0: The counter is switched at a falling edge of the GTETRGC signal 1 1: The counter is switched at both edges of the GTETRGC signal	R/W
7:6	WSGTRGD[1:0]	GTETRGD Signal Edge Select to Switch Counter (GTETRGSD Signal for SAFTY) 0 0: The GTETRGD signal is not used as a trigger to switch the counter 0 1: The counter is switched at a rising edge of the GTETRGD signal 1 0: The counter is switched at a falling edge of the GTETRGD signal 1 1: The counter is switched at both edges of the GTETRGD signal	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
16	WSELCA	Event Source Counter Switch Enable 0: Disables count switch by the LLPP GPT event A / NONSAFTY GPT event A input 1: Enables count switch by the LLPP GPT event A / NONSAFTY GPT event A input	R/W
17	WSELCB	Event Source Counter Switch Enable 0: Disables count switch by the LLPP GPT event B / NONSAFTY GPT event B input 1: Enables count switch by the LLPP GPT event B / NONSAFTY GPT event B input	R/W
18	WSELCC	Event Source Counter Switch Enable 0: Disables count switch by the LLPP GPT event C / NONSAFTY GPT event C input 1: Enables count switch by the LLPP GPT event C / NONSAFTY GPT event C input	R/W
19	WSELCD	Event Source Counter Switch Enable 0: Disables count switch by the LLPP GPT event D / NONSAFTY GPT event D input 1: Enables count switch by the LLPP GPT event D / NONSAFTY GPT event D input	R/W
20	WSELCE	Event Source Counter Switch Enable 0: Disables count switch by the LLPP GPT event E / NONSAFTY GPT event E input 1: Enables count switch by the LLPP GPT event E / NONSAFTY GPT event E input	R/W
21	WSELCF	Event Source Counter Switch Enable 0: Disables count switch by the LLPP GPT event F / NONSAFTY GPT event F input 1: Enables count switch by the LLPP GPT event F / NONSAFTY GPT event F input	R/W

Bit	Symbol	Function	R/W
22	WSELCG	Event Source Counter Switch Enable 0: Disables count switch by the LLPP GPT event G / NONSAFTY GPT event G input 1: Enables count switch by the LLPP GPT event G / NONSAFTY GPT event G input	R/W
23	CSELCH	Event Source Counter Switch Enable 0: Disables count switch by the LLPP GPT event H / NONSAFTY GPT event H input 1: Enables count switch by the LLPP GPT event H / NONSAFTY GPT event H input	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The GTSWSR register sets a counter switch source for the GTCNT counter. Count switch can be executed whether the counter is running (GTCR.CST = 1) or stopped (GTCR.CST = 0). Inputs from the GTETRGA, GTETRGB, GTETRGC, and GTETRGD, (GTETRGSA and GTETRGSB for SAFETY) pins are input to the GPT through the POEG. Set the polarity of these signals with the POEG.

19.2.40 GTSWOS : General PWM Timer Switch Offset Setting Register

Base address: GPTi = 0x9000_2000 + 0x0100 × i (i = 0 to 6)
GPTj = 0x8000_0000 + 0x0100 × (j - 7) (j = 7 to 13)
GPTk = 0x8100_0000 + 0x0100 × (k - 14) (k = 14 to 17)

Offset address: 0xDC

Bit position:	31																													0
Bit field:																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The GTSWOS register is a readable/writable register that sets the offset value of GTCNT counter which is reloaded by count switch operation. Access in 8-bit or 16-bit units to the GTSWOS register is prohibited, and it should be accessed in 32-bit units. The GTSWOS register is writable even while the GTCNT counter operation. If the GTSWOS register writing and the event input of count switch occur at the same time, the GTSWOS register setting before writing is reloaded to GTCNT counter. Write the GTSWOS register earlier than the event input of count switch to ensure that the written GTSWOS register value is reloaded by the count switch operation.

19.3 Operation

19.3.1 Basic Operation

The timer in each channel performs periodic count operation by count clock or hardware source. The GTCNT counter performs up-counting, or down-counting. The timer period is controlled by the GTPR register. When the GTCNT counter value matches the values for GTCCRA or GTCCRB register, the GTIOCnA or GTIOCnB pin outputs (n = 0 to 17) can be changed respectively. Also, the GTCCRA or GTCCRB register can be used as the input capture register by hardware source. The GTCCRC and GTCCRD registers can be performed as buffer registers for GTCCRA register, and the GTCCRE and GTCCRF registers can be performed as buffer registers for GTCCRB register.

19.3.1.1 Counter Operation

(1) Count Start/Count Stop

The counter in each channel starts counting operation when the corresponding GTCR.CST bit is set to 1, and stops counting when the bit is set to 0.

CST bit value can be changed by the following sources:

- Writing to the GTCR register
- Writing 1 to the bit corresponding to the channel number for the GTSTR register while the GTSSR.CSTRT bit is 1
- Writing 1 to the bit corresponding to the channel number for the GTSTP register while the GTPSR.CSTOP bit is 1
- Hardware source specified by the GTSSR register
- Hardware source specified by the GTPSR register

(2) Periodic Count Operation (In Up-Counting by the Count Clock)

The counter in each channel starts up-counting when the corresponding CST bit is set to 1 while the GTUPSR register and the GTDNSR register are 0x00000000. When the value of the GTCNT counter changes from that of the GTPR register to 0x00000000 (an overflow) while the GTINTAD.GTINTPR[0] bit is 1, a GTPn_OVF interrupt request is also issued. After GTCNT counter overflows, up-counting is resumed from 0x00000000.

Figure 19.2 shows an example of periodic count operation in up-counting by count clock.

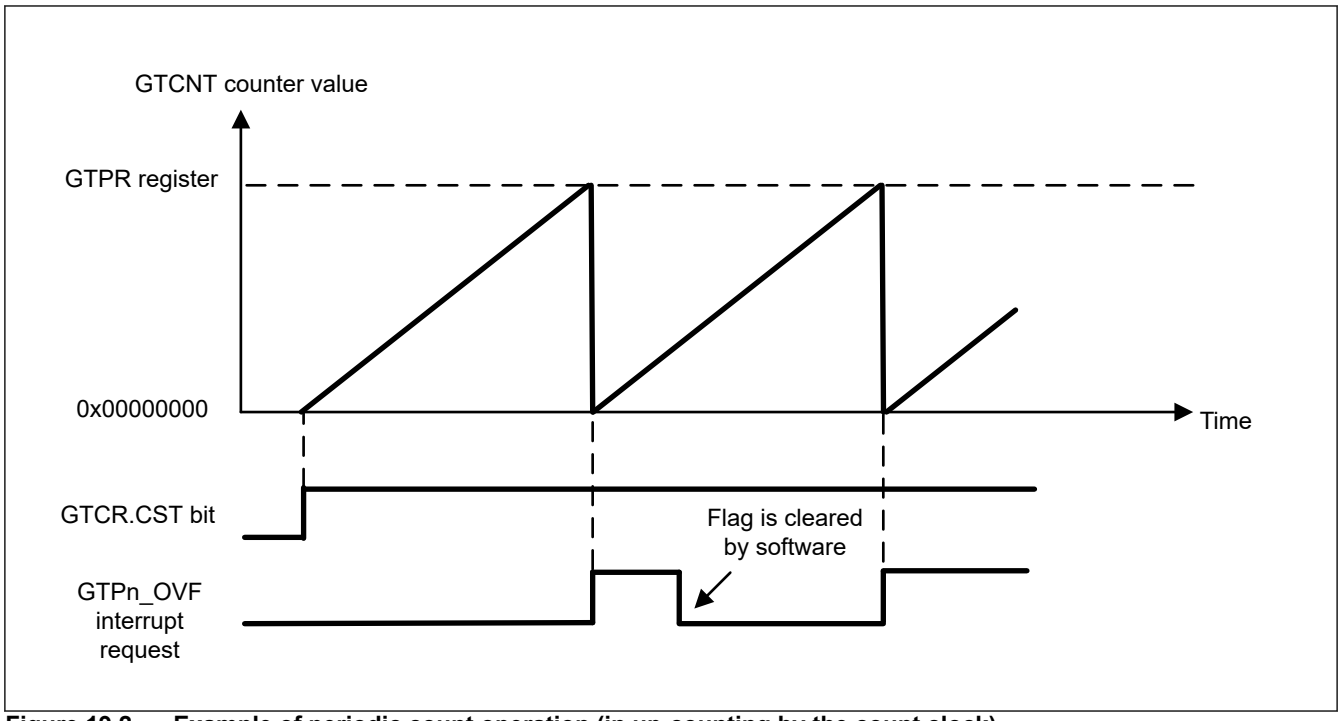


Figure 19.2 Example of periodic count operation (in up-counting by the count clock)

The following example shows the setting for periodic count operation in up-counting by count clock.

1. Set operating mode
   Set the operating mode with GTCR.MD[2:0] bits. In Figure 19.2, 000b (sawtooth-wave PWM mode) is set.
2. Set count direction
   Select the count direction with the GTUDDTYC register. In Figure 19.2, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter. In Figure 19.2, 0x00000000 is set.
6. Start count operation
   Set the GTCR.CST bit to 1 to start count operation.

(3) Periodic Count Operation (In Down-Counting by the Count Clock)

The counter in each channel can start down-counting when the corresponding GTUDDTYC.UD bit is set in a state of the GTUPSR register and the GTDNSR register as 0x00000000. When the value of the GTCNT counter changes from 0x00000000 to that of the GTPR register (an underflow) while the GTINTAD.GTINTPR[1] bit is 1, a GTPn_UDF interrupt request is also issued. After GTCNT counter underflows, down-counting is resumed from GTPR register value.

Figure 19.3 shows an example of periodic count operation in down-counting by count clock.

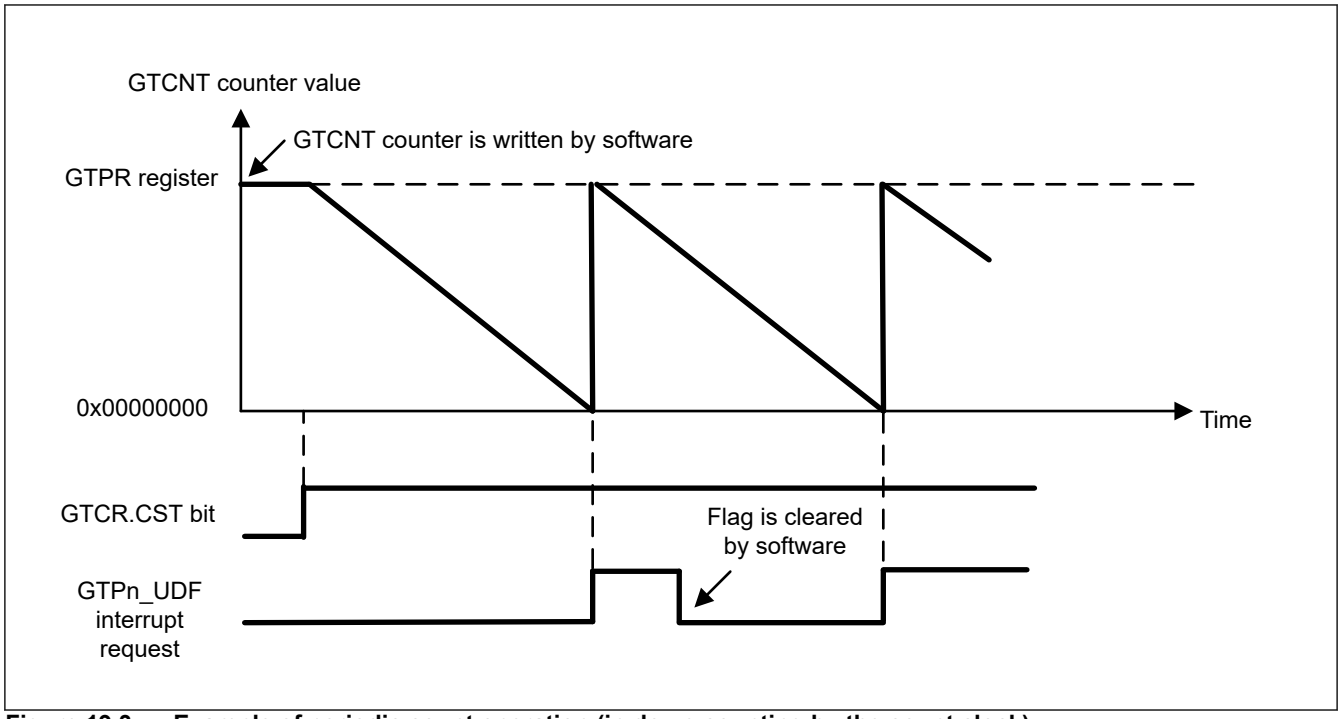


Figure 19.3 Example of periodic count operation (in down-counting by the count clock)

The following example shows the setting for periodic count operation in down-counting by count clock.

1. Set operating mode
   Set the operating mode with GTCR.MD[2:0] bits. In Figure 19.3, 000b (sawtooth-wave PWM mode) is set.
2. Set count direction
   Select the count direction with the GTUDDTYC register. In Figure 19.3, the lower 2 bits of the GTUDDTYC register is set to 10b and then to 00b for down-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter. In Figure 19.3, the GTPR register value is set.
6. Start count operation
   Set the GTCR.CST bit to 1 to start count operation.

(4) Event Count Operation (In Up-Counting by Hardware Source)

The counter in each channel can start up-counting by hardware source by setting the GTUPSR register. When the GTUPSR register is set, the count clock selected by the GTCR.TPCS[3:0] bits and the count direction set by the GTUDDTYC.UD bit are invalid. If the hardware source in up-counting and the hardware source in down-counting are generated at the same time, the GTCNT counter value does not change.

Operation due to overflow in up-counting by hardware source is the same as the periodic count operation by the count clock.

To perform up-counting by hardware source, start count operation by setting the GTCR.CST bit to 1. At the start, the count operation is synchronized with the count clock selected by GTCR.TPCS[3:0] bits, set the CST bit to 1, and wait until the first cycle of the count clock elapses to start the up-counting operation. To start up-counting after 1 PCLKGPTL (LLPP)/ PCLKM (NONSAFETY, SAFETY) from setting the CST bit to 1, set the GTCR.TPCS[3:0] bits to 0000b.

Figure 19.4 shows an example of up-counting operation by hardware source.

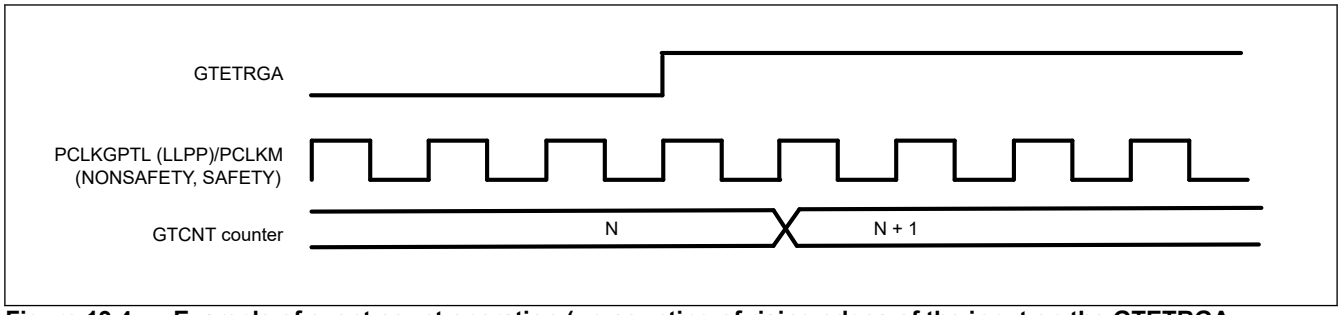


Figure 19.4 Example of event count operation (up-counting of rising edges of the input on the GTETRGA pin)

The following example shows the setting for event count operation in up-counting by a hardware source.

1. Set count-up source
   Select the count-up source with the GTUPSR register.
2. Period setting
   Set a period in the GTPR register.
3. Set initial value for counter
   Set the initial value in the GTCNT counter.
4. Start count operation
   Set the GTCR.CST bit to 1 to start count operation.

(5) Event Count Operation (In Down-Counting by Hardware Source)

The counter in each channel can start down-counting by hardware source by setting the GTDNSR register. When the GTDNSR register is set, the count clock selected by the GTCR.TPCS[3:0] bits and the count direction set by the GTUDDTYC.UD bit are invalid. If the hardware source in up-counting and the hardware source in down-counting are generated at the same time, the GTCNT counter value does not change.

Operation due to underflow in down-counting by hardware source is the same as the periodic count operation by the count clock.

To perform down-counting by hardware source, start count operation by setting the GTCR.CST bit to 1. At the start, the count operation is synchronized with the count clock selected by GTCR.TPCS[3:0] bits, set the CST bit to 1, and wait until the first cycle of the count clock elapses to start down-counting operation. To start down-counting after 1 PCLKGPTL (LLPP)/PCLKM (NONSAFETY, SAFETY) from setting the CST bit to 1, set the GTCR.TPCS[3:0] bits to 0000b.

Figure 19.5 shows an example of down-counting operation by hardware source.

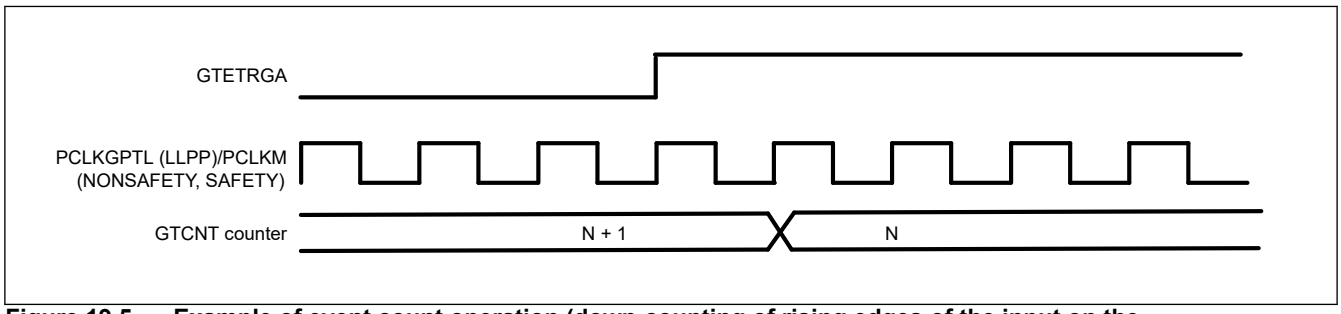


Figure 19.5 Example of event count operation (down-counting of rising edges of the input on the GTETRGA pin)

The following example shows the setting for event count operation in down-counting by hardware source.

1. Set count-down source
   Select the count-down source with the GTDNSR register.
2. Period setting
   Set a period in the GTPR register.
3. Set initial value for counter
   Set the initial value in the GTCNT counter.

4. Start count operation
   Set the GTCR.CST bit to 1 to start count operation.

(6) Counter Clearing Operation

The counter in each channel can be cleared by the following sources:

- Writing to the GTCNT counter
- Writing 1 to the bit corresponding to the channel number for the GTCLR register while the GTCSR.CCLR bit is 1
- Hardware source specified by the GTCSR register

Write access during counting (when CST = 1) is disabled.

GTCLR register can be written to and cleared by hardware source whether the count operation is performed (GTCR.CST bit = 1) or is stopped (CST bit = 0). When the count direction flag is set as decrement (GTST.TUCF flag = 0) in sawtooth-wave mode selected with GTCR.MD[2:0] bits, the GTCNT counter value for writing to the GTCLR register and for clearing by hardware source becomes the GTPR register value whether the counter is in operation or is stopped. The GTCNT counter value becomes 0x00000000 for other settings.

When the event count operation is set (at least one of the bits for the GTUPSR or the GTDNSR register is set as 1), writing to the GTCLR register and clearing by hardware source are performed right after the clear source is generated (the operation is based on PCLKGPTL (LLPP)/PCLKM (NONSAFETY, SAFETY)). For other settings, clearing operation is synchronized with the count clock selected with the GTCR.TPCS[3:0] bits.

19.3.1.2 Waveform Output by Compare Match Counter Operation

Compare match refers to when the GTCNT counter value matches the GTCCRA or GTCCRB register value. The output of the corresponding GTIOCnA or GTIOCnB pin (n = 0 to 17) can be set to be driven low, high, or toggled in synchronization with the counter clock (including in the case of event counting) after the match.

In addition, the GTIOCnA or GTIOCnB pin output can be driven low, high, or toggled at the "end of the cycle" which is determined by the GTPR register. The end of the cycle is as follows:

- For sawtooth waves in up-counting: When the GTCNT counter value changes from the GTPR register value to 0x00000000 (overflow)
- For sawtooth waves in down-counting: When the GTCNT counter value changes from 0x00000000 to the GTPR register value (underflow)
- For sawtooth waves with the GTCNT counter clear
- For triangle waves: When the GTCNT counter value changes from 0x00000000 to 0x00000001 (trough)

(1) Low Output and High Output

Figure 19.6 shows an example of low output and high output operation by a compare match of the GTCNT counter and GTCCRA register, and of the GTCNT counter and GTCCRB register.

In this example, the GPT0.GTCNT counter performs up-counting, and settings are made so that high is output from the GTIOC0A pin by a GPT0.GTCCRA register compare match, and low is output from the GTIOC0B pin by a GPT0.GTCCRB register compare match. The pin level does not change when the specified level and pin level match.

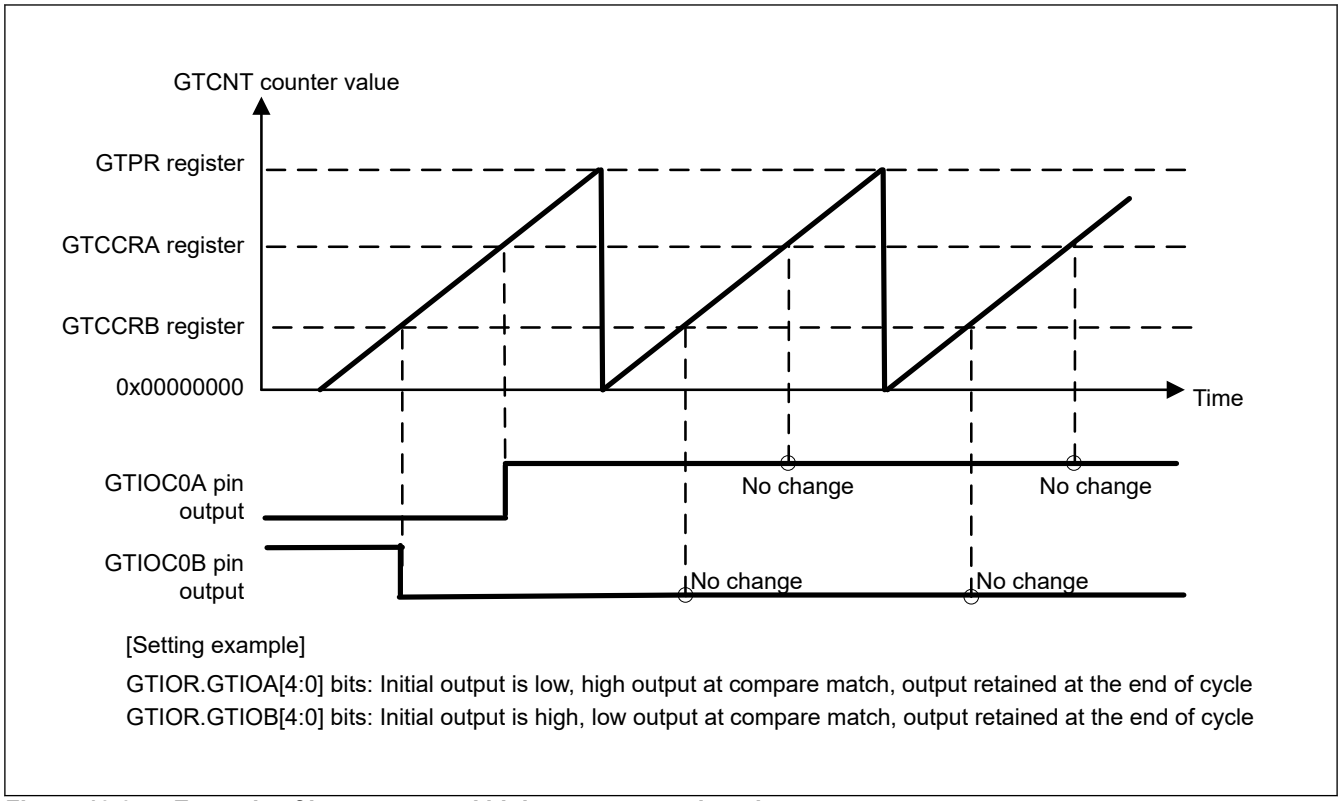


Figure 19.6 Example of low output and high output operation pin

The following example shows the setting for low/high output operation.

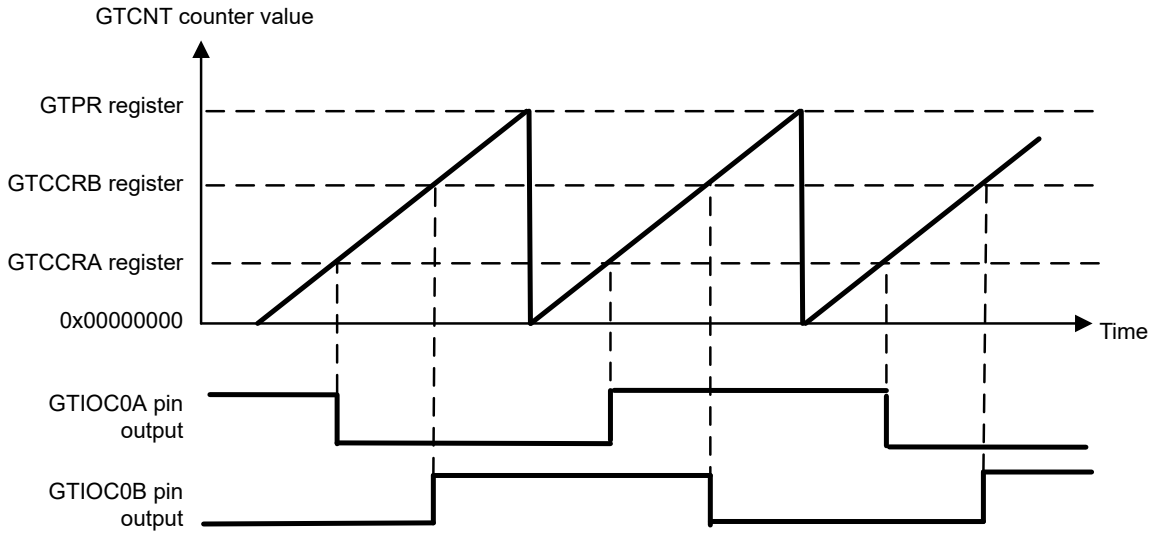
1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits. In Figure 19.6, 000b (sawtooth-wave PWM mode) is set.
2. Set count direction
   Select the count direction (up or down) with the GTUDDTYC register.
   In Figure 19.6, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
6. Set GTIOCnm pin function
   Set the GTIOCnm pin function with GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register.
   In Figure 19.6, GTIOA[4:0] bits = 00010b and GTIOB[4:0] bits = 10001b.
7. Enable GTIOCnm pin output
   Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8. Set compare match value
   Set compare match values in the GTCCRA and GTCCRB registers.
9. Set count operation
   Set the GTCR.CST bit to 1 to start count operation.

(2) Toggled Output

Figure 19.7 and Figure 19.8 show examples of toggle output operations by compare match between the GPT0.GTCNT counter and the GTCCRA and GTCCRB registers.

In Figure 19.7, the GPT0.GTCNT counter performs up-counting. Settings are made so that the GTIOC0A pin output by a GPT0.GTCCRA register compare match and GTIOC0B pin output by a GPT0.GTCCRB register compare match are toggled.

In Figure 19.8, the GPT0.GTCNT counter performs up-counting. Settings are made so that the GTIOC0A pin output is toggled by a compare match of GPT0.GTCCRA register and the GTIOC0B pin output is toggled at the end of the cycle.

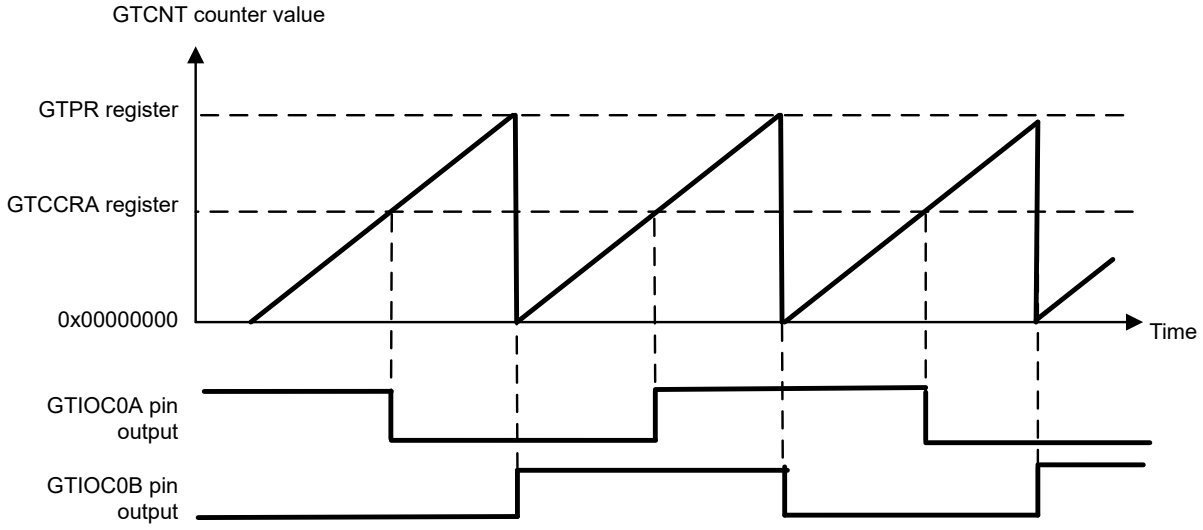


[Setting example]

GTIOR.GTIOA[4:0] bits: Initial output is high, output toggled at compare match, output retained at the end of cycle

GTIOR.GTIOB[4:0] bits: Initial output is low, output toggled at compare match, output retained at the end of cycle

Figure 19.7 Example of toggled output operation (1)



[Setting example]

GTIOR.GTIOA[4:0] bits: Initial output is high, output toggled at compare match, output retained at the end of cycle

GTIOR.GTIOB[4:0] bits: Initial output is low, output retained at compare match, output toggled at the end of cycle

Figure 19.8 Example of toggled output operation (2)

The following example shows the setting for toggled output operation.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 19.7 and Figure 19.8, 000b (sawtooth-wave PWM mode) is set.
2. Set count direction
   Select the count direction (up or down) with the GTUDDTYC register.
   In Figure 19.7 and Figure 19.8, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
6. Set GTIOCnm pin function
   Set the GTIOCnm pin function with GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register.
   In Figure 19.7, GTIOA[4:0] bits = 10011b and GTIOB[4:0] bits = 00011b.
   In Figure 19.8, GTIOA[4:0] bits = 10011b and GTIOB[4:0] bits = 01100b.
7. Enable GTIOCnm pin output
   Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8. Set compare match value
   Set compare match values in the GTCCRA and GTCCRB registers.
9. Set count operation
   Set the GTCR.CST bit to 1 to start count operation.

19.3.1.3 Input Capture Function

Through detecting hardware sources selected by the GTICASR and GTICBSR registers, the GTCNT counter values can be transferred to the GTCCRA and GTCCRB registers, respectively.

Figure 19.9 shows an example of input capture function operation.

In this example, the GPT0.GTCNT counter performs up-counting by count clock, and settings are made so that an input capture is performed at both edges of the GTIOC0A pin input in the GTCCRA register and at the rising edge of the GTIOC0B pin input in the GTCCRB register.

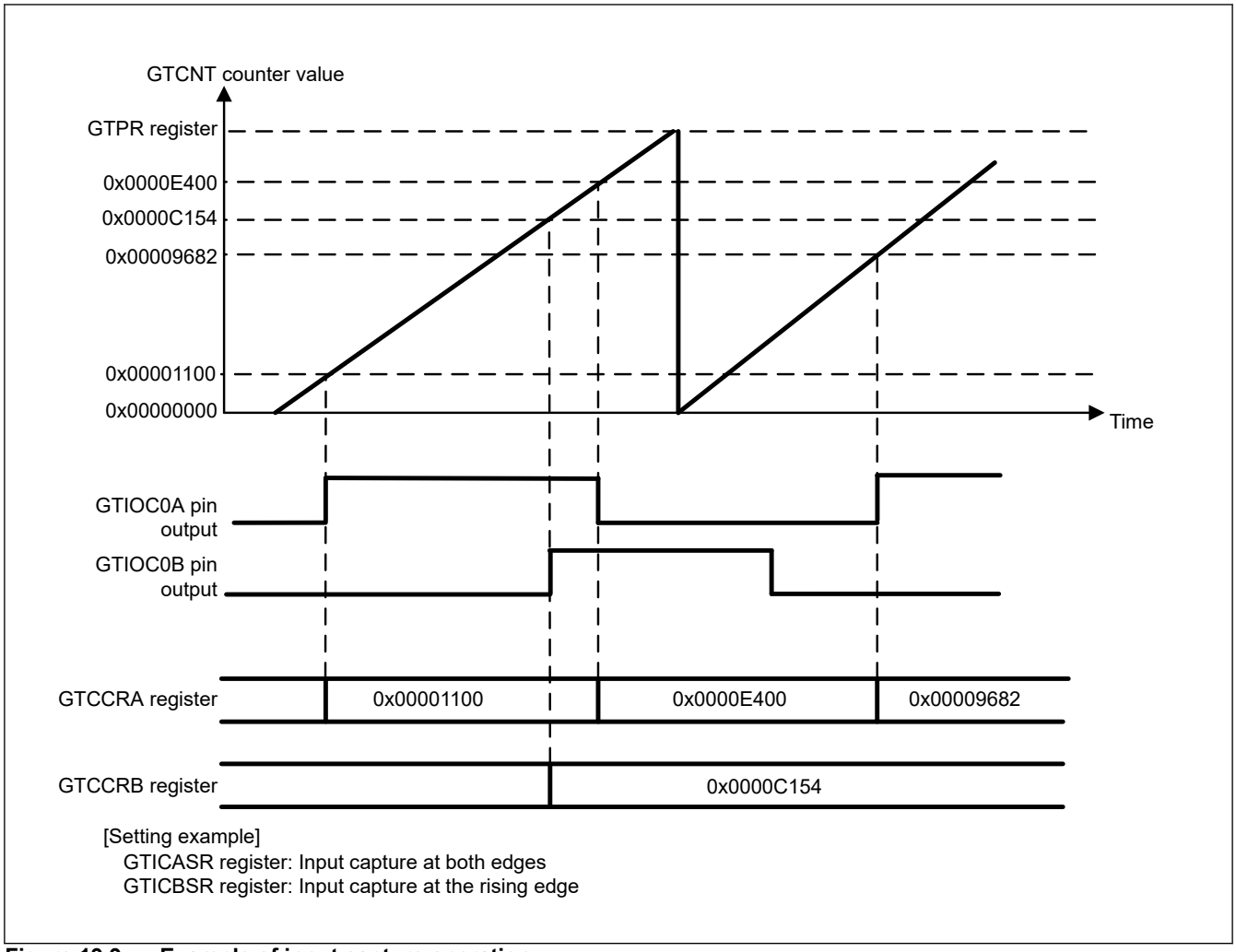


Figure 19.9 Example of input capture operation

The following example shows the setting for input capture operation.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 19.9, 000b (sawtooth-wave PWM mode) is set.
2. Set count direction
   Select the count direction (up or down) with the GTUDDTYC register. In Figure 19.9, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
6. Select input capture source
   Select the input capture sources with the GTICASR and GTICBSR registers.
   In Figure 19.9, GTICASR = 0x00000F00 and GTICBSR = 0x00003000.
7. Set count operation
   Set the GTCR.CST bit to 1 to start count operation.

19.3.2 Buffer Operation

The following buffer operations can be set with the GTBER register:

- With the GTPR, GTPBR, and GTPDBR registers used together
- With the GTCCRA, GTCCRC, and GTCCRD registers used together
- With the GTCCRB, GTCCRE, and GTCCRF registers used together
- With the GTADTRA, GTADTBRA, and GTADTDBRA registers used together
- With the GTADTRB, GTADTBRB, and GTADTDBRB registers used together

The following buffer operations can be set with the GTDTCR register:

- With the GTDVU and GTDBU registers used together
- With the GTDVD and GTDBD registers used together

19.3.2.1 GTPR Register Buffer Operation

The GTPBR register can function as a buffer register for the GTPR register, and the GTPDBR register can function as a buffer register for the GTPBR register (double buffer register for the GTPR register).

The buffer transfer is performed at an overflow (in up-counting) or underflow (in down-counting) in sawtooth-wave mode or in event counting, or at a trough in triangle-wave mode.

When in sawtooth-wave mode and in event counting, the buffer transfer is performed even when the counter clearing occurs during the count is in operation.

- Hardware source clearing (clear source selected with bits [23:0] of GTCSR register)
- Clearing by software (when the GTCLR.CCLRn bit is written to 1 while GTCSR.CCLR bit is set to 1) (n = 0 to 17)

To set the GTPR register to function as double buffer, set the GTBER.PR[1:0] bits to 10b or 11b. For single buffer operation, set 01b. For not to function as buffer, set 00b.

Figure 19.10 to Figure 19.12 show examples of the GTPR register buffer operation.

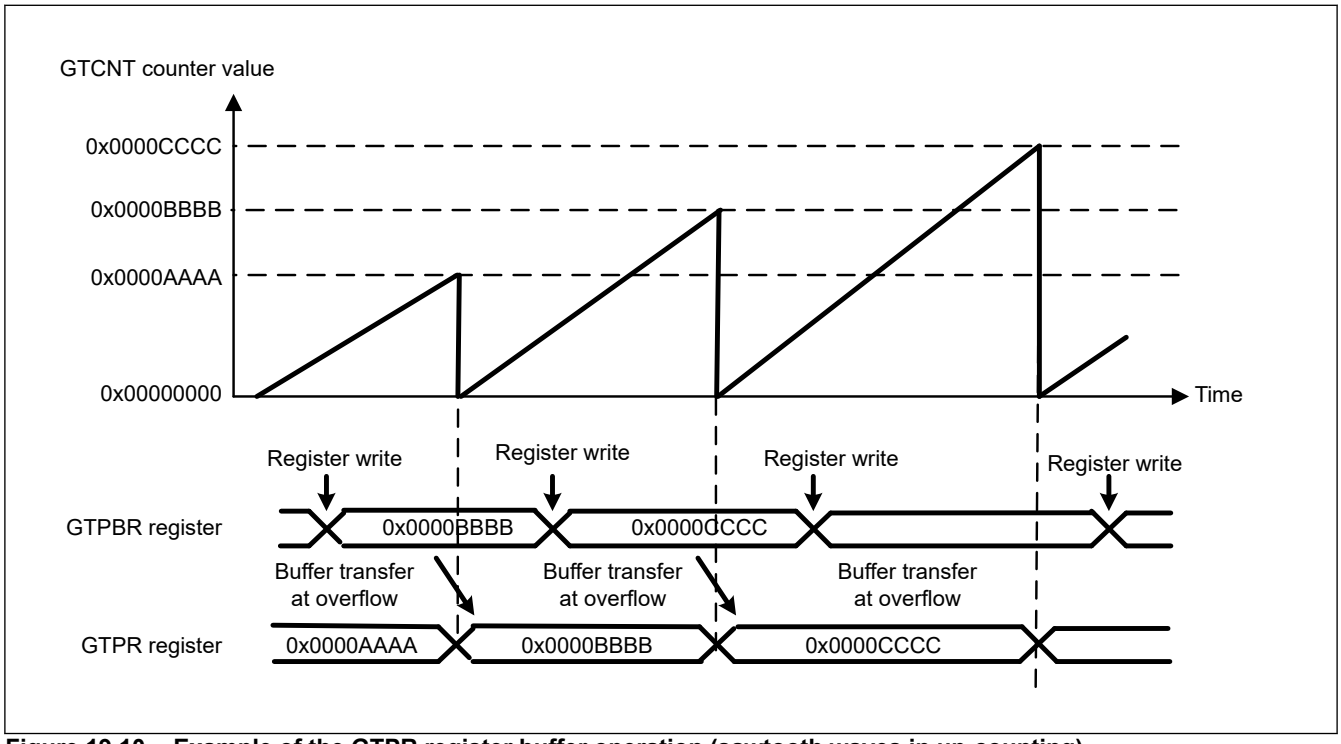


Figure 19.10 Example of the GTPR register buffer operation (sawtooth waves in up-counting)

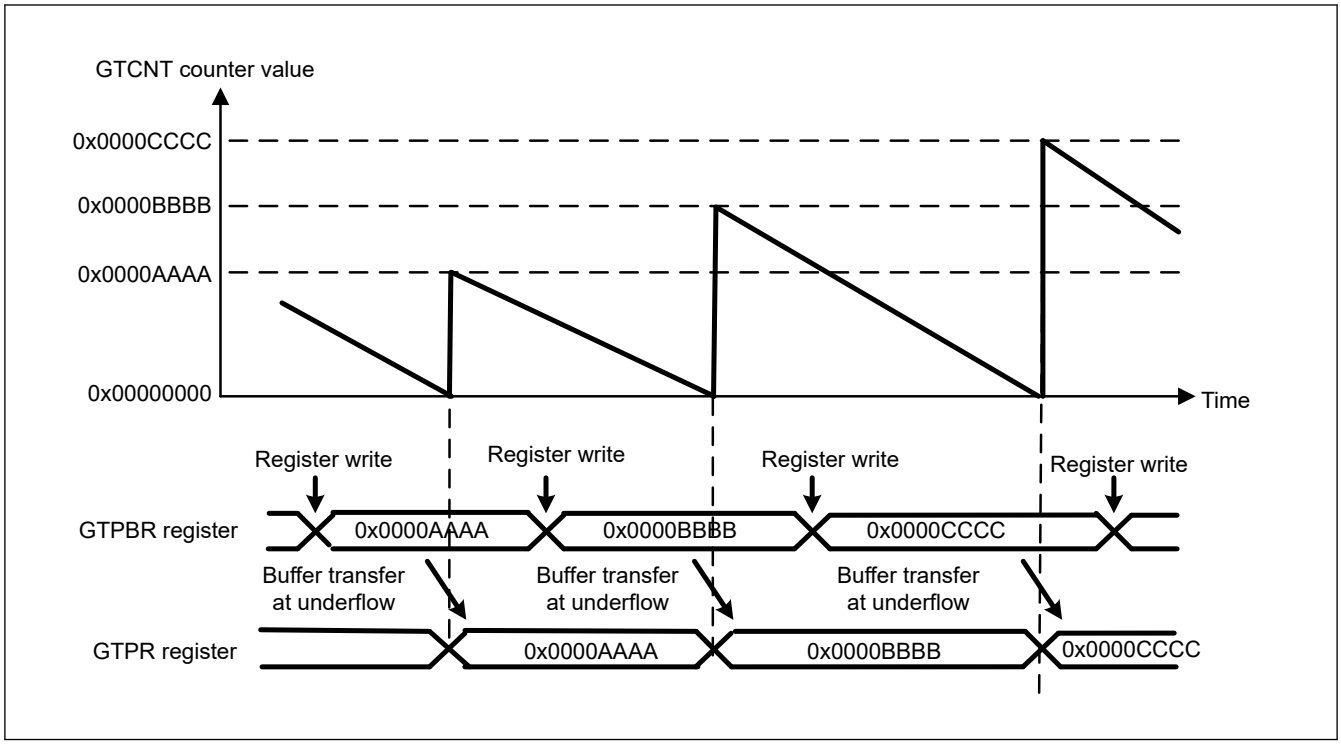


Figure 19.11 Example of the GTPR register buffer operation (sawtooth waves in down-counting)

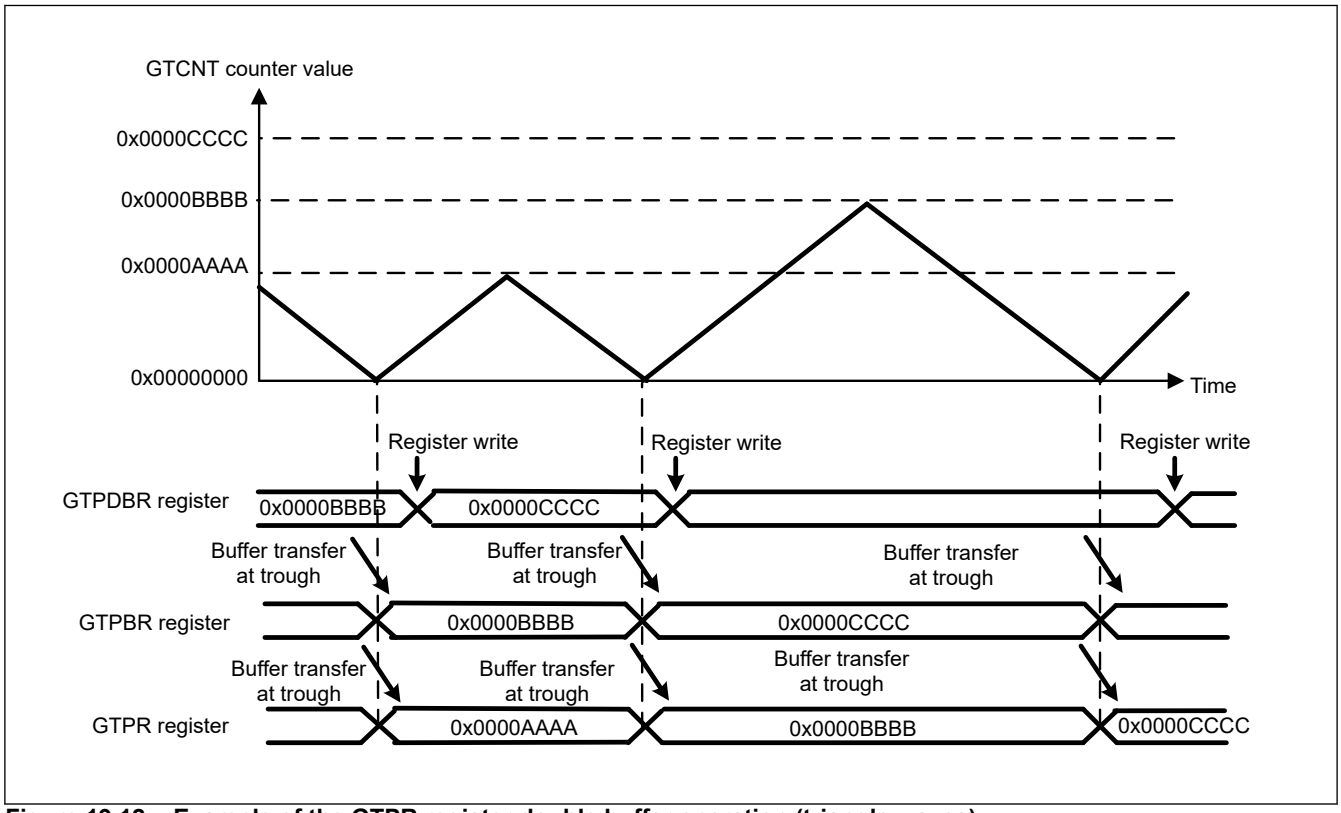


Figure 19.12 Example of the GTPR register double buffer operation (triangle waves)

The following example shows the setting for the GTPR register buffer operation.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 19.10 and Figure 19.11, 000b (sawtooth-wave PWM mode) is set, and in Figure 19.12, 100b (triangle-wave PWM mode 1) is set.
2. Set count direction

Select the count direction (up or down) with the GTUDDTYC register. In Figure 19.10, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting. In Figure 19.11, the lower 2 bits of the GTUDDTYC register is set to 10b and then to 10b for down-counting.

3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
6. Set buffer operation
   Set buffer operation with the GTBER.PR[1:0] bits.
   In Figure 19.10 and Figure 19.11, PR[1:0] bits = 01b. In Figure 19.12, PR[1:0] bits = 1xb.
7. Set buffer value
   For buffer operation, set a period value for the next cycle from the current cycle in the GTPBR register. For double buffer operation, also set a period value for the cycle after the next cycle in the GTPDBR register.
8. Set count operation
   Set the GTCR.CST bit to 1 to start count operation.
9. Set buffer value by cycle
   For buffer operation, set a period value for the next cycle from the current in the GTPBR register. For double buffer operation, set a period value for the cycle after the next cycle in the GTPDBR register.

19.3.2.2 Buffer Operation for the GTCCRA and GTCCRB Registers

The GTCCRC register can function as the GTCCRA buffer register and the GTCCRD register can function as the GTCCRC buffer register (double buffer register for the GTCCRA register). Similarly, the GTCCRE register can function as the GTCCRB buffer register and the GTCCRF register can function as the GTCCRE buffer register (double buffer register for the GTCCRB register).

To set the GTCCRA or GTCCRB register to function as a double buffer, set the GTBER.CCRA[1:0] or GTBER.CCRB[1:0] bits to 10b or 11b. For single buffer operation, set 01b. Not to function as buffer, set 00b.

The following describes buffer operation during output compare and input capture operation.

(1) When the GTCCRA or GTCCRB Register Functions as Output Compare Register

Buffer transfer is performed for the following three cases:

- Buffer transfer at an overflow (in up-counting) and underflow (in down-counting)
  Buffer transfer is performed at an overflow (in up-counting) or an underflow (in down-counting) in sawtooth-wave mode and in event counting, and at a trough (triangle-wave PWM mode 1) or at a crest/trough (triangle-wave PWM mode 2) in triangle-wave mode.
- Buffer transfer by counter clearing
  In sawtooth-wave mode and in event counting, the buffer transfer by similar counter clearing sources in counting operation and in section 19.3.2.1. GTPR Register Buffer Operation, is performed in the same way at an overflow (in up-counting) or an underflow (in down-counting).
  In triangle-wave mode, buffer transfer by counter clearing is not performed.
- Forcible buffer transfer
  In both sawtooth-wave mode and triangle-wave mode, buffer transfer of the GTCCRA and GTCCRB registers is forcibly performed by writing 1 to the GTBER.CCRSWT bit while the counting is stopped.
  In sawtooth-wave one-shot pulse mode and triangle-wave PWM mode 3, buffer transfer from the GTCCRD register to temporary register A and from the GTCCRF register to temporary register B is also performed using the forcible buffer operation.

When the setting of the GTBER.DBRTECm (m = A, B) bit is 1 in sawtooth-wave one-shot pulse mode or triangle-wave PWM mode 3, transfer from the intermediate buffers to the GTCCRm (m = A, B) registers is repeated on a cyclic basis even while buffer transfer is disabled by the setting of the GTBER.BD0 bit or buffer transfer extended skipping function

(function for repeated double-buffer operation while buffer transfer is disabled). For details, see section 19.8.2.2. Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer.

Figure 19.13 to Figure 19.15 show examples of the GTCCRA and GTCCRB registers buffer operations.

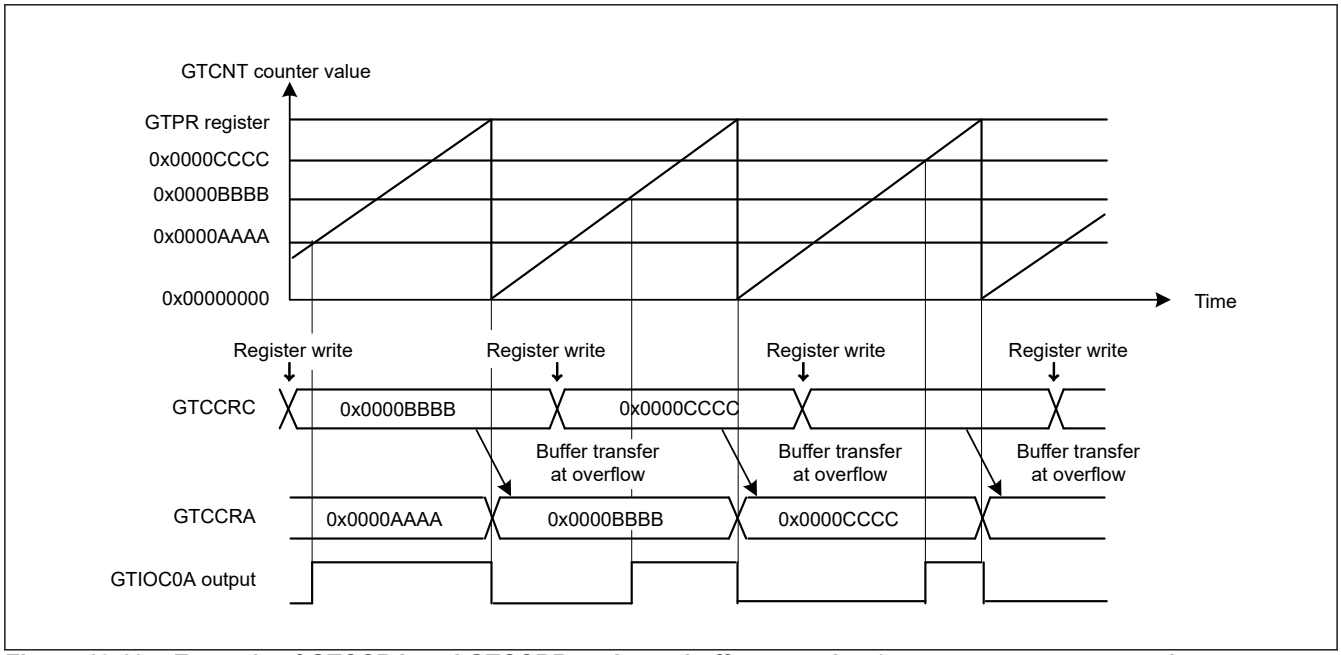


Figure 19.13 Example of GTCCRA and GTCCRB registers buffer operation (output compare, sawtooth waves in up-counting, high output at GTCCRA register compare match, low output at the end of the cycle)

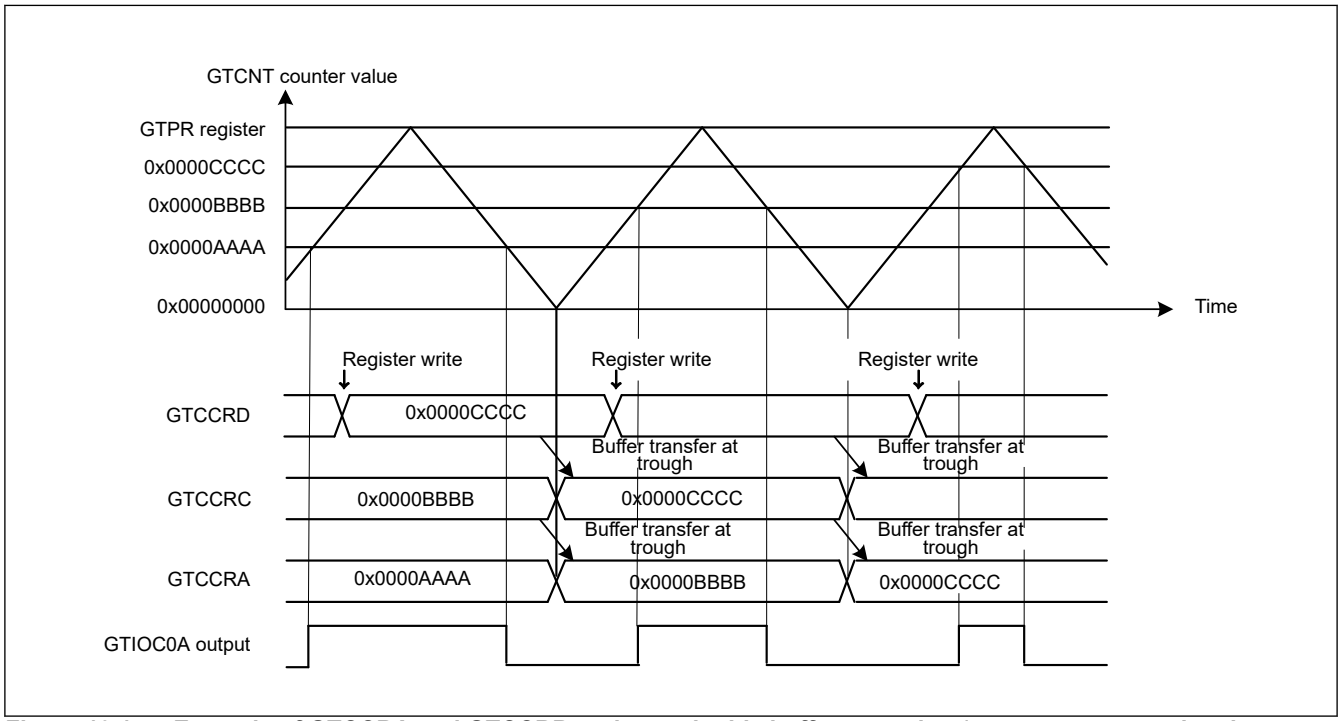


Figure 19.14 Example of GTCCRA and GTCCRB registers double buffer operation (output compare, triangle waves, buffer operation at trough, output toggled at GTCCRA register compare match, output retained at the end of the cycle)

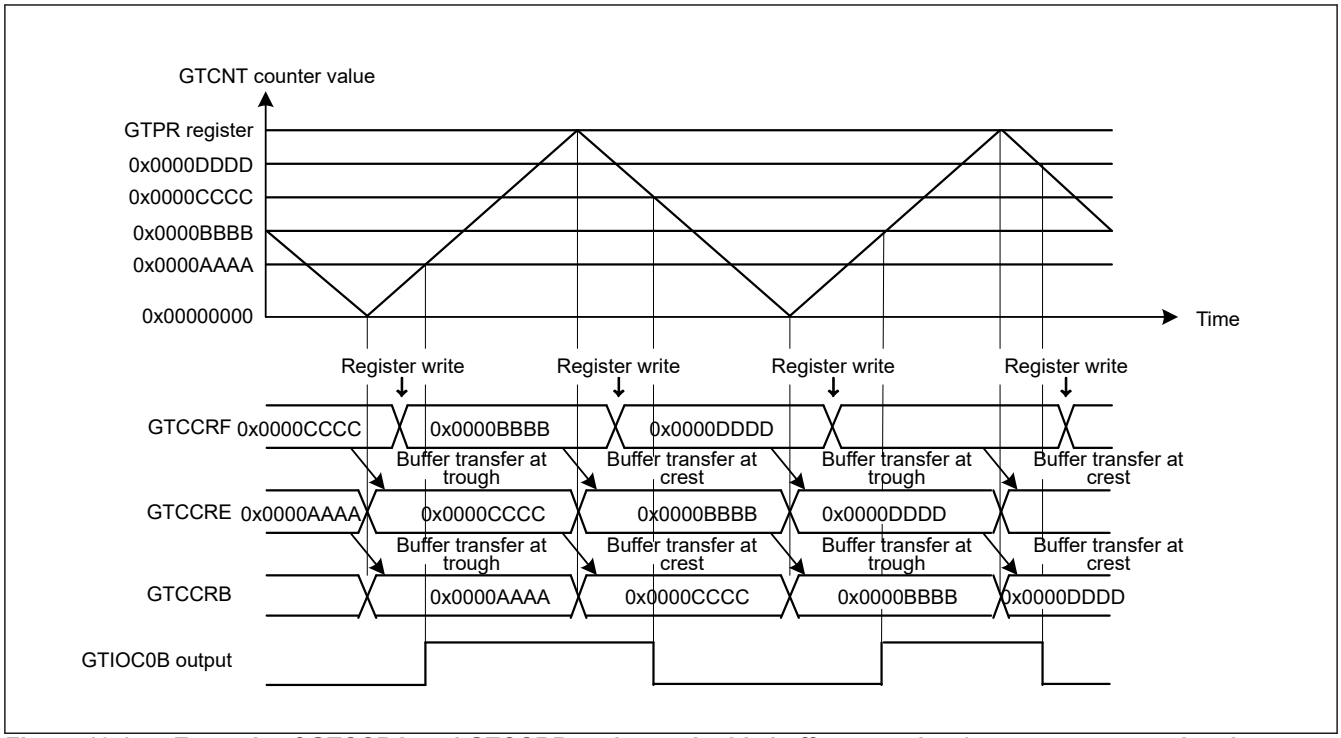


Figure 19.15 Example of GTCCRA and GTCCRB registers double buffer operation (output compare, triangle waves, buffer operation at both troughs and crests, output toggled at GTCCRB register compare match, output retained at the end of the cycle)

The following example shows setting for buffer operation of the GTCCRA and GTCCRB registers for output compare.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 19.13, 000b (sawtooth-wave PWM mode) is set, in Figure 19.14, 100b (triangle-wave PWM mode 1) is set, and in Figure 19.15, 101b (triangle-wave PWM mode 2) is set.
2. Set count direction
   Select the count direction (up or down) with the GTUDDTYC register.
   In Figure 19.13, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
6. Set GTIOCnm pin function
   Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 19.13, GTIOA[4:0] bits = 00110b, in Figure 19.14, GTIOA[4:0] bits = 00011b, and in Figure 19.15, GTIOB[4:0] bits = 00011b.
7. Enable GTIOCnm pin output
   Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8. Set buffer operation
   Set buffer operation with the GTBER.CCRA[1:0] bits and CCRB[1:0] bits.
   In Figure 19.13, CCRA[1:0] bits = 01b, in Figure 19.14, CCRA[1:0] bits = 1xb, and in Figure 19.15, CCRB[1:0] bits = 1xb.
9. Set compare match value
   Set the GTIOCnA pin changing point in the GTCCRA register and GTIOCnB pin changing point in the GTCCRB register.
10. Set buffer value

For buffer operation, set the GTIOCnA pin and GTIOCnB pin changing points in one cycle after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA pin and GTIOCnB pin changing points in two cycles after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRD and GTCCRF registers, respectively.

11. Set count operation
    Set the GTCR.CST bit to 1 to start count operation.
12. Set buffer value by cycle
    For buffer operation, set the GTIOCnA pin and GTIOCnB pin changing points in one cycle after the current cycle (in sawtooth-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA pin and GTIOCnB pin changing points in two cycles after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTCCRD and GTCCRF registers, respectively.

(2) When the GTCCRA or GTCCRB Register Functions as Input Capture Register

The timing of transfer to the buffers is when an input capture is generated. When an input capture is generated, the GTCNT counter value is transferred to GTCCRA and GTCCRB registers and the stored GTCCRA and GTCCRB registers values are transferred to buffer registers. Buffer transfer by counter clearing is not performed at input capture.

Figure 19.16 and Figure 19.17 show examples of buffer operation of the GTCCRA and GTCCRB registers.

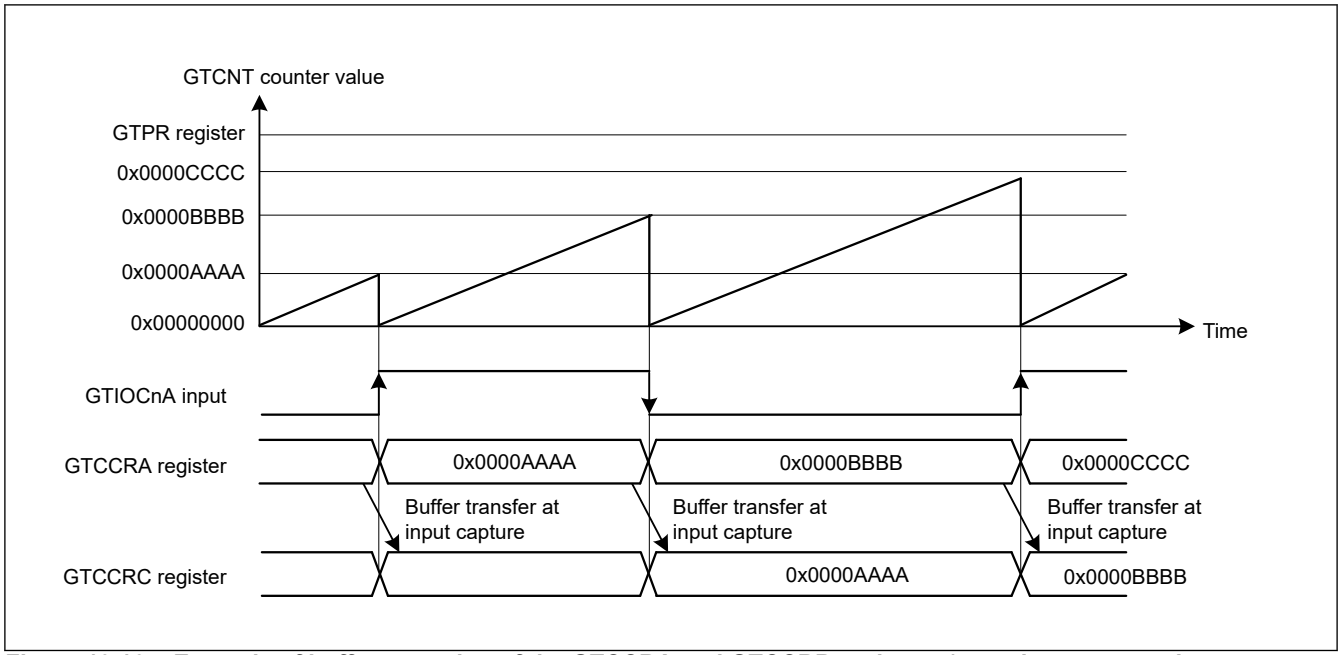


Figure 19.16 Example of buffer operation of the GTCCRA and GTCCRB registers (counting up to produce a sawtooth wave, and clearing the GTCNT counter and input capture on both edges of the input on the GTIOCnA pin) (n = 0 to 17)

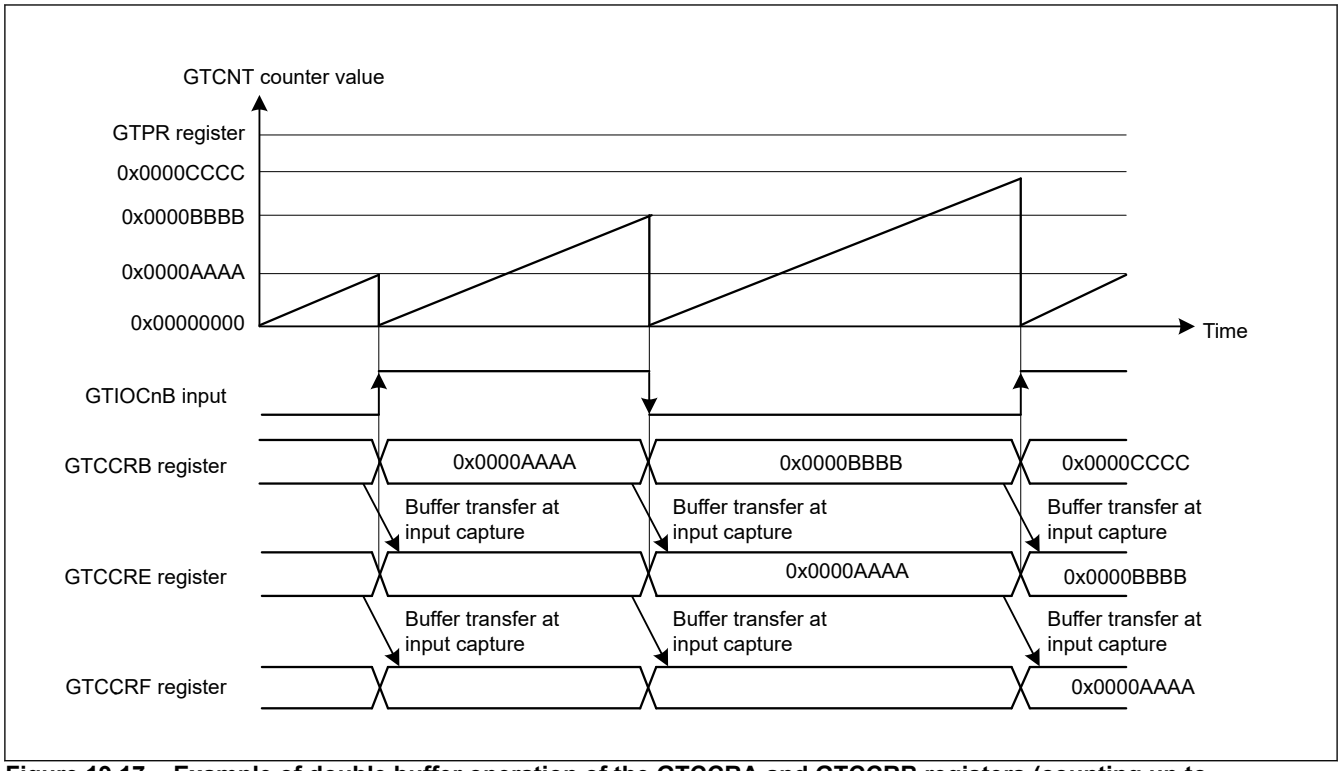


Figure 19.17 Example of double buffer operation of the GTCCRA and GTCCRB registers (counting up to produce a sawtooth wave, and clearing the GTCNT counter and input capture on both edges of the input on the GTIOCnB pin) (n = 0 to 17)

The following example shows setting for buffer operation of the GTCCRA and GTCCRB registers for input capture.

1. Set operating mode and counter clear source
   Set the operating mode with GTCR.MD[2:0] and counter clear source with the GTCSR register. In Figure 19.16, MD[2:0] = 000b (sawtooth-wave PWM mode) and GTCSR register = 0x00000F00 and in Figure 19.17, MD[2:0] bits = 000b (sawtooth-wave PWM mode) and GTCSR register = 0x0000F000.
2. Select count direction
   Select the count direction (up or down) with the GTUDDTYC register.
   In Figure 19.16, the lower 2 bits of GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
6. Set input capture source
   Select input capture sources with the GTICASR and GTICBSR registers. In Figure 19.16, GTICASR register = 0x00000F00 and in Figure 19.17, GTICBSR register = 0x0000F000.
7. Set buffer operation
   Set buffer operation with the GTBER.CCRA[1:0] bits and CCRB[1:0] bits.
   In Figure 19.16, CCRA[1:0] bits = 01b and in Figure 19.17, CCRB[1:0] bits = 1xb.
8. Set count operation
   Set the GTCR.CST bit to 1 to start count operation.

19.3.2.3 Buffer Operation for the GTADTRA and GTADTRB Registers

The GTADTBRA register can function as the GTADTRA buffer register and the GTADTDBRA register can function as the GTADTBRA buffer register (double buffer register for the GTADTRA register). Similarly, the GTADTBRB register can

function as the GTADTRB buffer register and the GTADTDBRB register can function as the GTADTBRB buffer register (double buffer register for the GTADTRB register).

To set the GTADTRA or GTADTRB register to function as a double buffer, set the GTBER.ADTDA or ADTDB bit to 1. For single buffer operation, set 0. Not to function as buffer, set the GTBER.ADTTA[1:0] or ADTTB[1:0] bits to 00b. The buffer transfer timing can be set with the ADTTA[1:0] and ADTTB[1:0] bits to an overflow (in up-counting) or an underflow (in down-counting) in sawtooth-wave mode, with ADTTA[1:0] and ADTTB[1:0] bits to 01b for a crest, to 10b for a trough, or to 11b for both crest and trough in triangle-wave mode.

In sawtooth-wave mode, when the ADTTA[1:0] and ADTTB[1:0] bits are set to value other than 00b and in count operation, the buffer transfer, by similar counter clearing sources in section 19.3.2.1. GTPR Register Buffer Operation, is performed in the same way at an overflow (in up-counting) or an underflow (in down-counting).

Figure 19.18 to Figure 19.20 show examples of buffer operation of the GTADTRA and GTADTRB registers.

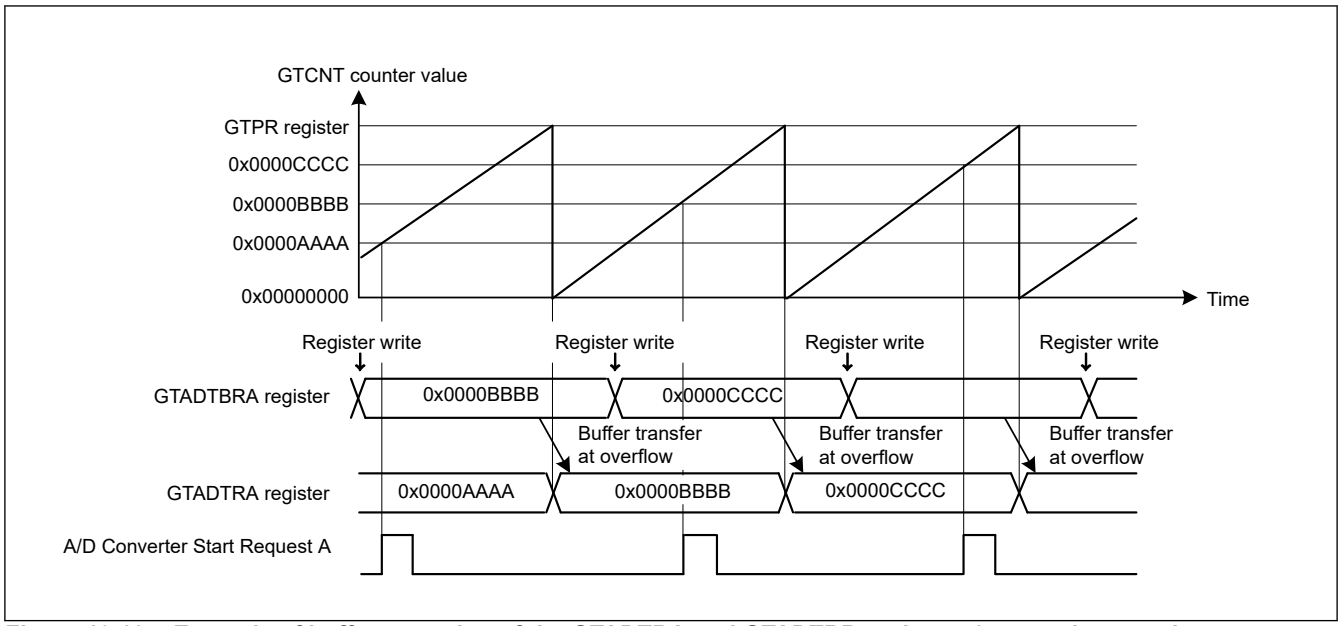


Figure 19.18 Example of buffer operation of the GTADTRA and GTADTRB registers (sawtooth waves in up-counting, A/D converter start request generated by up-counting)

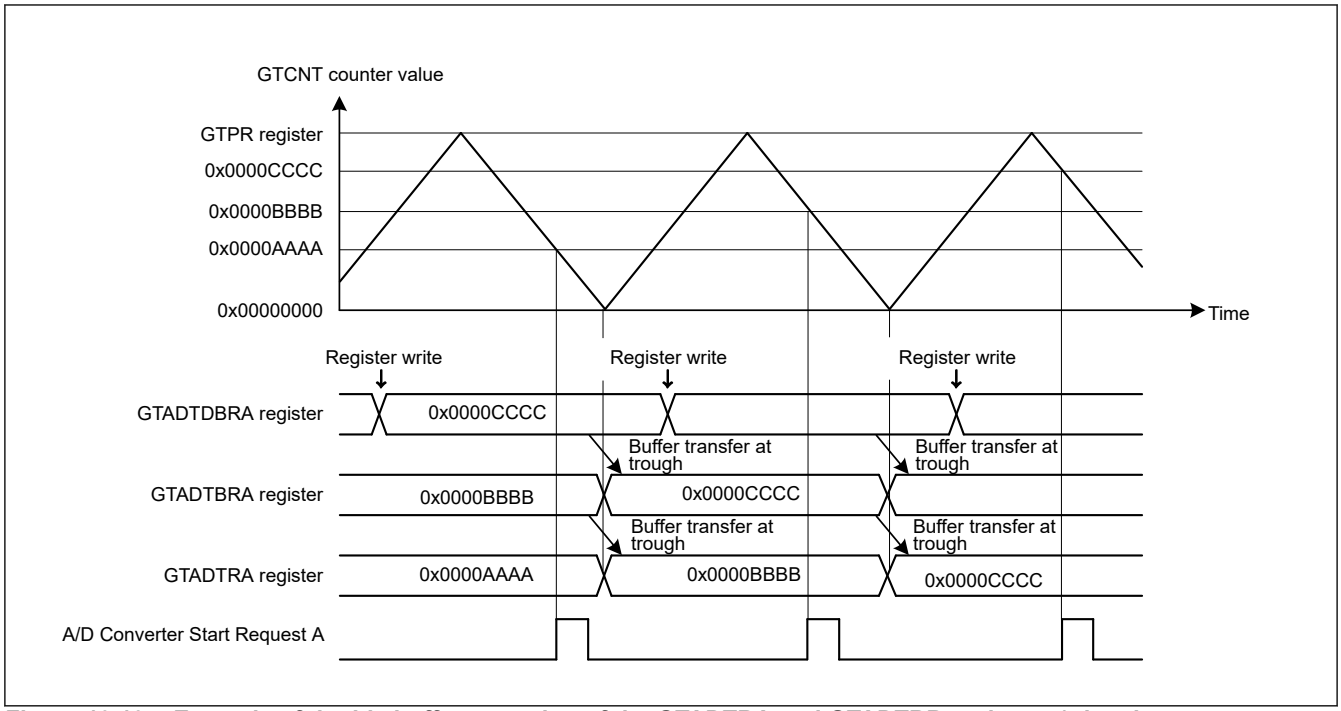


Figure 19.19 Example of double buffer operation of the GTADTRA and GTADTRB registers (triangle waves, buffer transfer at troughs, A/D converter start request generated by down-counting)

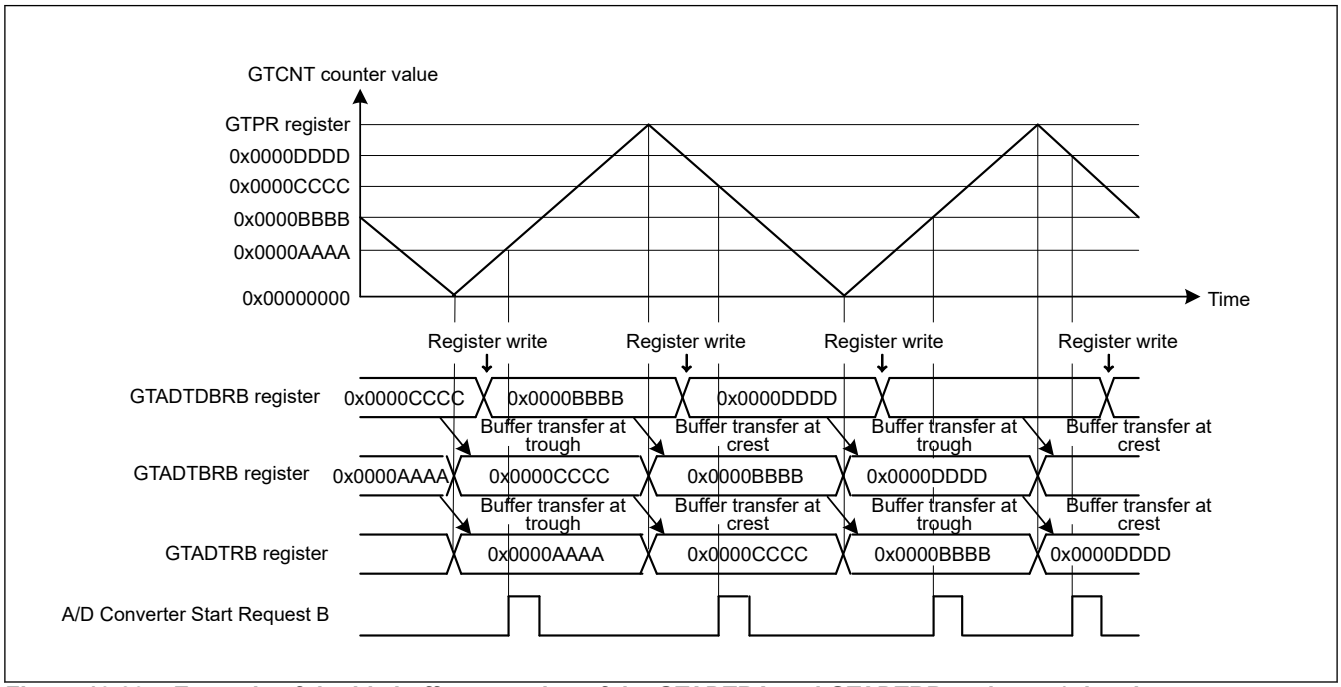


Figure 19.20 Example of double buffer operation of the GTADTRA and GTADTRB registers (triangle waves, buffer transfer at both troughs and crests, A/D converter start request generated by both up-counting and down-counting)

The following example shows setting for buffer operation of the GTADTRA and GTADTRB registers.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 19.18, 000b (sawtooth-wave PWM mode) is set, and in Figure 19.19 and Figure 19.20, 100b, 101b, or 110b (triangle-wave PWM mode) is set.
2. Set count direction
   Select the count direction (up or down) with the GTUDDTYC register.
   In Figure 19.18, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.

3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
6. Set buffer operation
   Set buffer operation with the ADTTA[1:0], ADTTB[1:0], ADTDA, and ADTDB bits in the GTBER register. In Figure 19.18, ADTTA[1:0] bits = 01b, 10b, or 11b and ADTDA bit = 0. In Figure 19.19, ADTTA[1:0] bits = 10b and ADTDA bit = 1, and in Figure 19.20, ADTTB[1:0] bits = 11b and ADTDB bit = 1.
7. Set compare match value
   Set the A/D converter start request point in the GTADTRA and GTADTRB registers.
8. Set buffer value
   For buffer operation, set the A/D converter start request point in one cycle after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, also set the A/D converter start request points in two cycle after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.
9. Enable A/D converter start request
   Set the enable A/D converter start request with the ADTRAUEN, ADTRADEN, ADTRBUEN, and ADTRBDEN bits in the GTINTAD register. In Figure 19.18, ADTRAUEN bit = 1, in Figure 19.19, ADTRADEN bit = 1, and in Figure 19.20, ADTRBUEN bit = 1 and ADTRBDEN bit = 1 .
10. Set count operation
    Set the GTCR.CST bit to 1 to start count operation.
11. Set buffer value for each cycle
    For buffer operation, set the A/D converter start request point in one cycle after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers. For double buffer operation, set the A/D converter start request points in two cycle after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers.

19.3.3 PWM Output Operating Mode

The GPT can output PWM waveforms to the GTIOCnA pin or GTIOCnB pin (n = 0 to 17) by a compare match between the GTCNT counter and the GTCCRA or GTCCRB register.

By setting the GTDTCR, GTDVU, and GTDVD registers, the compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

(1) Sawtooth-Wave PWM Mode

In sawtooth-wave PWM mode, the GTCNT counter performs sawtooth-wave (half-wave) operation by setting the period in the GTPR register and a PWM waveform is output to the GTIOCnA or GTIOCnB pin when a GTCCRA or GTCCRB register compare match occurs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR register setting.

Figure 19.21 shows an example of sawtooth-wave PWM mode operation.

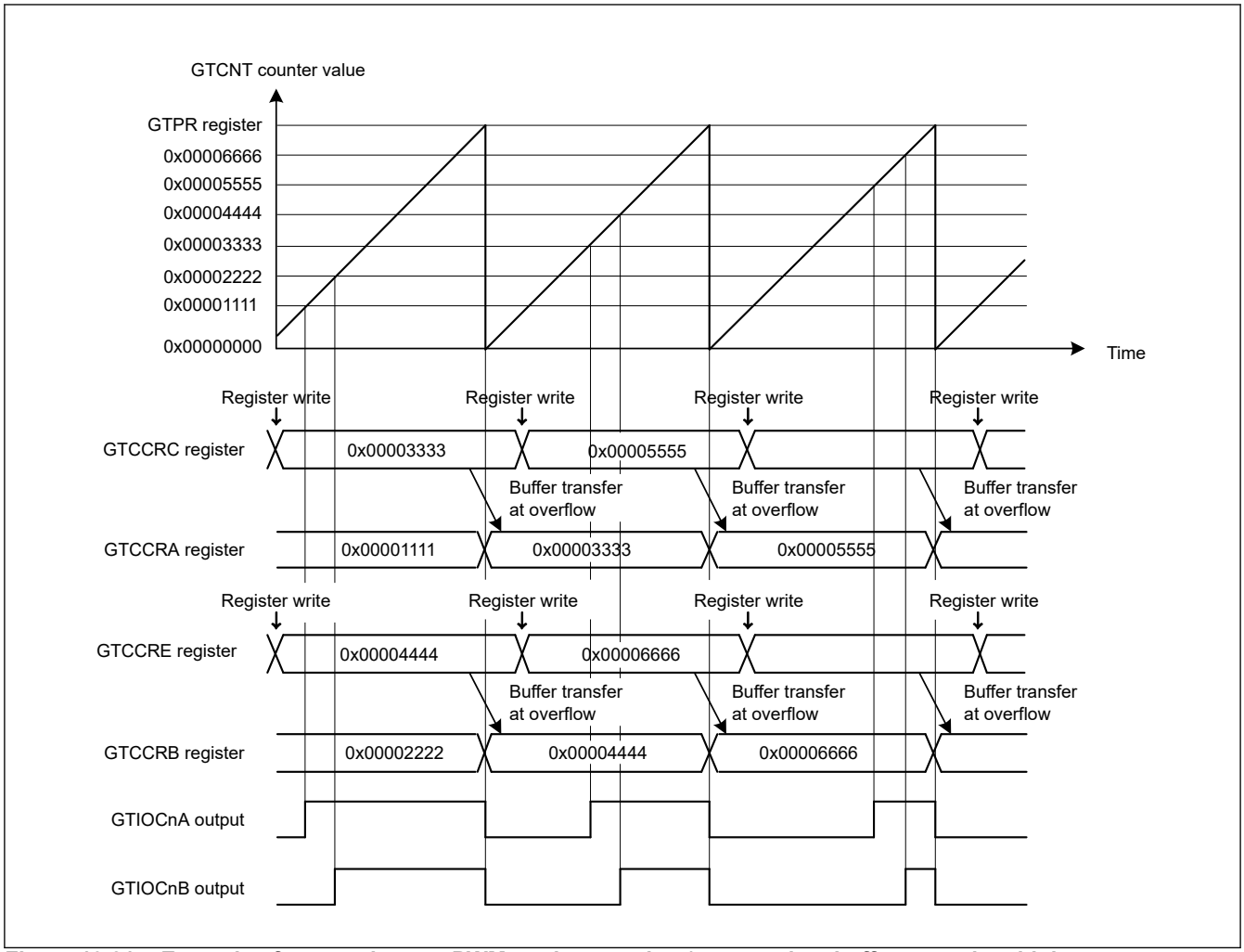


Figure 19.21 Example of sawtooth-wave PWM mode operation (up-counting, buffer operation, high output at GTCCRA/GTCCRB register compare match, low output at the end of the cycle) (n = 0 to 17)

The following example shows setting for sawtooth-wave PWM mode.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 19.21, 000b (sawtooth-wave PWM mode) is set.
2. Set count direction
   Select the count direction (up or down) with the GTUDDTYC register.
   In Figure 19.21, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
6. Set GTIOCnm pin function
   Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 19.21, GTIOA[4:0] bits = 00110b and GTIOB[4:0] bits = 00110b.
7. Enable GTIOCnm pin output
   Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8. Set buffer operation
   Set buffer operation with the CCRA[1:0] and CCRB[1:0] bits in the GTBER register.
   In Figure 19.21, CCRA[1:0] bits = 01b and CCRB[1:0] bits = 01b.

9. Set compare match value
   Set the GTIOCnA pin changing point in the GTCCRA register and the GTIOCnB pin changing point in the GTCCRB register.
10. Set buffer value
    For buffer operation, set the GTIOCnA and the GTIOCnB pin changing points in one cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA pin and GTIOCnB pin changing points in two cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.
11. Set count operation
    Set the GTCR.CST bit to 1 to start count operation.
12. Set buffer value for each cycle
    For buffer operation, set the GTIOCnA pin and the GTIOCnB pin changing point in one cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively.
    For double buffer operation, set the GTIOCnA pin and GTIOCnB pin changing points in two cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.

Note: n = 0 to 17, m = A, B

(2) Sawtooth-Wave One-Shot Pulse Mode

The sawtooth-wave one-shot pulse mode is a mode in which the period is set in the GTPR register, the GTCNT counter performs sawtooth-wave (half-wave) operation and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 17) at a compare match of the GTCCRA or GTCCRB register with buffer operation fixed.

Buffer operation in sawtooth-wave one-shot pulse mode is different from the usual buffer operation. Buffer transfer is performed from the following registers:

- GTCCRC register to GTCCRA register at the end of the cycle
- GTCCRE register to GTCCRB register at the end of the cycle
- GTCCRD register to temporary register A at the end of the cycle
- GTCCRF register to temporary register B at the end of the cycle
- Temporary register A to GTCCRA register at a GTCCRA register compare match
- Temporary register B to GTCCRB register at a GTCCRB register compare match

The pin output value can be selected from low output, high output, or toggle output separately for a compare match and the end of the cycle according to the GTIOR register setting.

The temporary register A and the temporary register B can perform forcible buffer transfer from the GTCCRD register to the temporary register A and from the GTCCRF register to the temporary register B respectively by writing 1 to the GTBER.CCRSW bit while the count is stopped.

By setting the GTDTCR, GTDVU, and GTDVD registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

When the GTBER.DBRTECm (m = A, B) bit is set to 1, transfer from an intermediate buffer to the GTCCRm register is repeated on a cyclic basis by using the temporary register x (x = C, E) and temporary register m which operate as intermediate buffers for the GTCCRx and GTCCRm registers, respectively, even while buffer transfer is disabled (repeated double buffer operation function during disabling of buffer transfer). For details, see section 19.8.2.2. Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer.

Figure 19.22 shows an example of sawtooth-wave one-shot pulse mode operation.

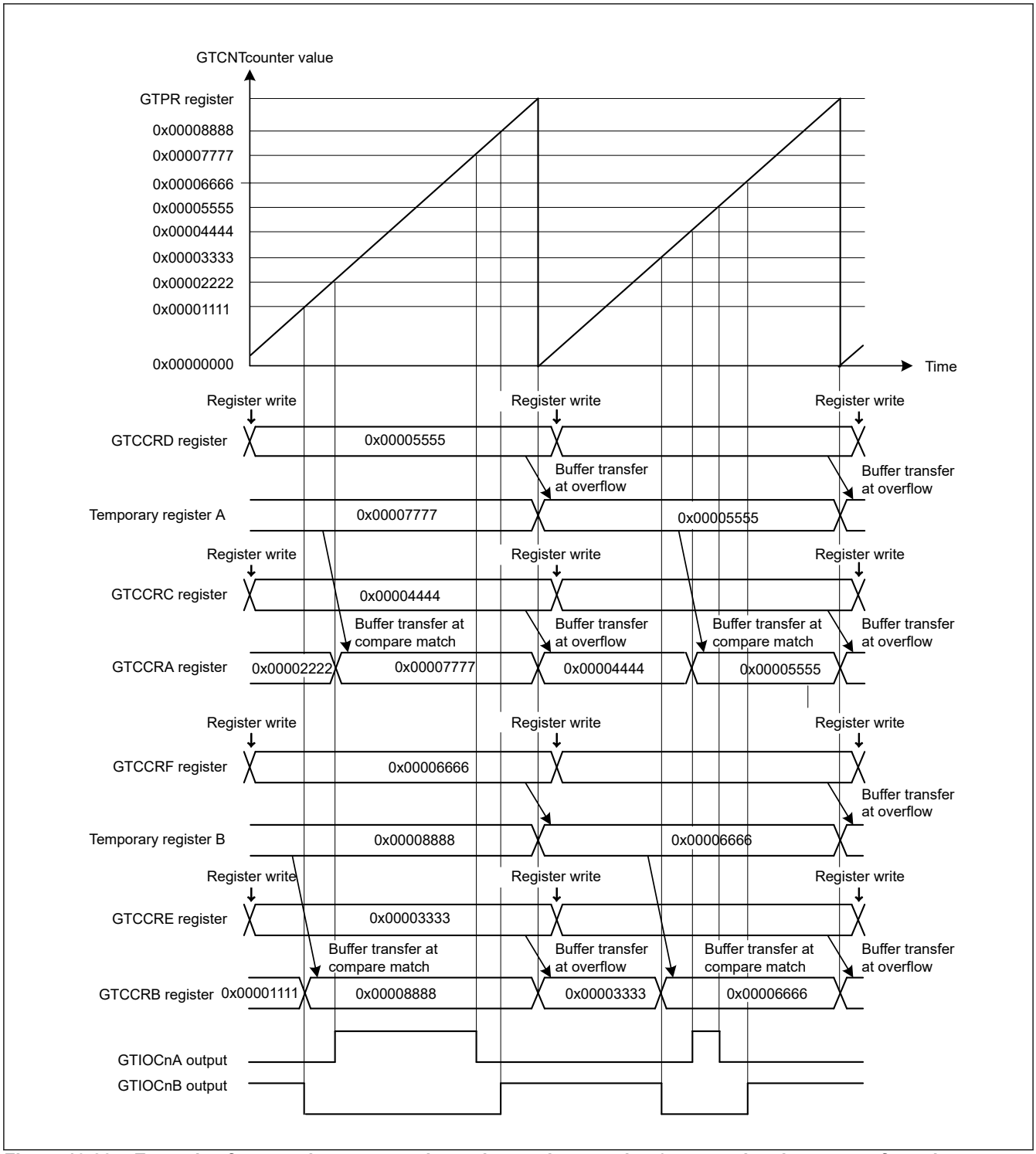


Figure 19.22 Example of sawtooth-wave one-shot pulse mode operation (up-counting, low output from the GTIOCnA pin and high output from the GTIOCnB pin at initial output, output toggled at GTCCRA/GTCCRB register compare match, output retained at the end of the cycle)

The following example shows setting for sawtooth-wave one-shot PWM mode.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits. In Figure 19.22, 001b (sawtooth-wave PWM one-shot mode) is set.
2. Set count direction
   Select the count direction (up or down) with the GTUDDTYC register.
   In Figure 19.22, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.

3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
6. Set GTIOCnm pin function
   Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register.
   In Figure 19.22, GTIOA[4:0] bits = 00011b and GTIOB[4:0] bits = 10011b.
7. Enable GTIOCnm pin output
   Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8. Set buffer value
   Set the GTIOCnA pin changing point immediately after the count start in the GTCCRC and GTCCRD registers and the GTIOCnB pin changing point in the GTCCRE and GTCCRF registers.
9. Set forcible buffer transfer
   Set the GTBER.CCRSWT bit to 1 to forcibly transfer buffer register data.
10. Set buffer value
    For buffer operation, set the GTIOCnA pin changing point for the next cycle from the current cycle to the GTCCRC and GRCCRD registers, and set the GTIOCnB pin changing point for the next cycle from the current cycle to the GTCCRE and GTCCRF registers.
11. Set count operation
    Set the GTCR.CST bit to 1 to start count operation.
12. Set buffer value by cycle
    For buffer operation, set the GTIOCnA pin changing point for next cycle from the current cycle to the GTCCRC and GTCCRD registers and set the GTIOCnB pin changing point for the next cycle from the current cycle to the GTCCRE and GTCCRF registers.

Note: n = 0 to 17, m = A, B

(3) Triangle-Wave PWM Mode 1 (32-Bit Transfer at Trough)

The triangle-wave PWM mode 1 is a mode in which the period is set in the GTPR register, the GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 17) when a GTCCRA or GTCCRB register compare match occurs.

Buffer transfer is performed at the trough. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR register setting.

By setting the GTDTCR, GTDVU, and GTDVD registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

Figure 19.23 shows an example of triangle-wave PWM mode 1 operation.

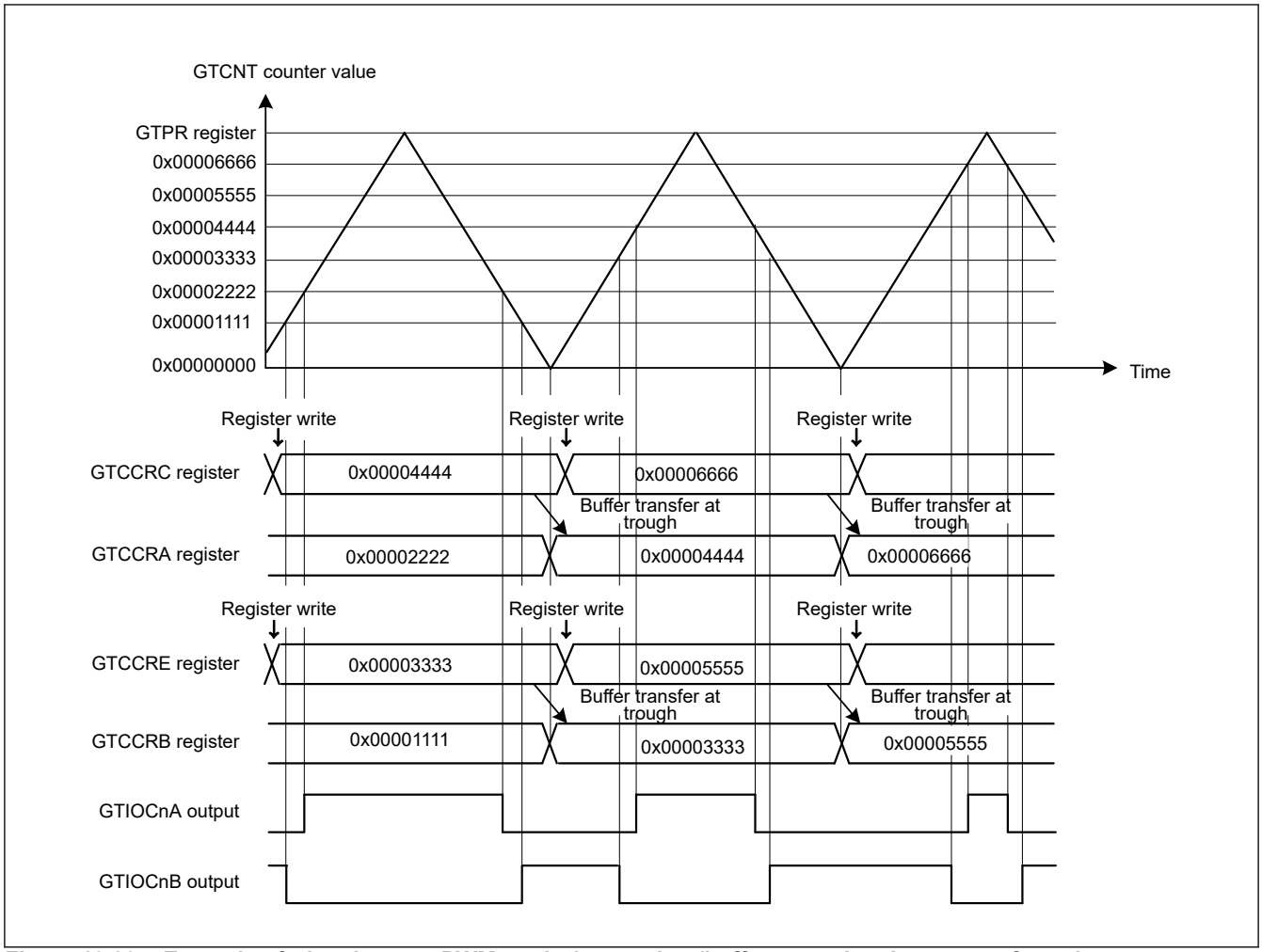


Figure 19.23 Example of triangle-wave PWM mode 1 operation (buffer operation, low output from the GTIOCnA pin and high output from the GTIOCnB pin at initial output, output toggled at GTCCRA/GTCCRB register compare match, output retained at the end of the cycle) (n = 0 to 17)

The following example shows setting for triangle-wave PWM mode 1.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 19.23, 100b (triangle-wave PWM mode 1) is set.
2. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
3. Period setting
   Set a period in the GTPR register.
4. Set initial value for counter
   Set the initial value in the GTCNT counter.
5. Set GTIOCnm pin function
   Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 19.23, GTIOA[4:0] bits = 00011b and GTIOB[4:0] bits = 10011b.
6. Enable GTIOCnm pin output
   Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
7. Set buffer operation
   Set buffer operation with CCRA[1:0] and CCRB[1:0] bits in the GTBER register.
   In Figure 19.23, CCRA[1:0] bits = 01b and CCRB[1:0] bits = 01b.
8. Set compare match value

Set the GTIOCnA pin changing point in the GTCCRA register and GTIOCnB pin changing point in the GTCCRB register.

9. Set buffer value
   For buffer operation, set the GTIOCnA pin and GTIOCnB pin changing points in one cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively. For double buffer operation, also set the GTIOCnA pin and GTIOCnB pin changing points in two cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.
10. Set count operation
    Set the GTCR.CST bit to 1 to start count operation.
11. Set buffer value by cycle
    For buffer operation, set the GTIOCnA pin and GTIOCnB pin changing points in one cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively.
    For double buffer operation, set the GTIOCnA pin and GTIOCnB pin changing points in two cycles after the current cycle in the GTCCRD and GTCCRF registers, respectively.

Note: n = 0 to 17, m = A, B

(4) Triangle-Wave PWM Mode 2 (32-Bit Transfer at Crest and Trough)

Similarly to triangle-wave PWM mode 1, in triangle-wave PWM mode 2, the period is set in the GTPR register, the GTCNT counter performs triangle-wave (full-wave) operation, and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 17) when a GTCCRA or GTCCRB register compare match occurs. The buffer transfer is performed at both crests and troughs. The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR register setting.

By setting the GTDTCR, GTDVU, and GTDVD registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

Figure 19.24 shows an example of triangle-wave PWM mode 2 operation.

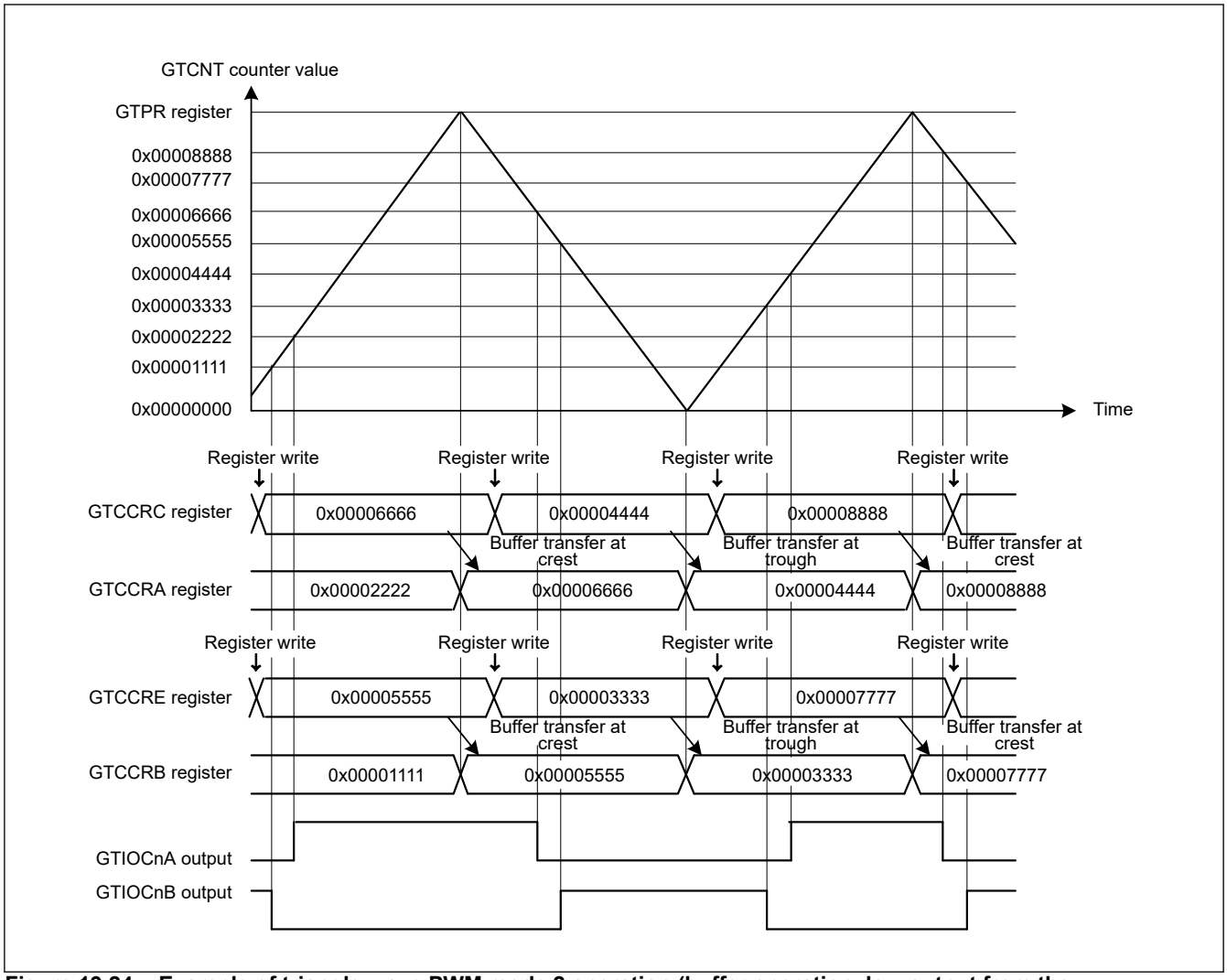


Figure 19.24 Example of triangle-wave PWM mode 2 operation (buffer operation, low output from the GTIOCnA pin and high output from the GTIOCnB pin at initial output, output toggled at GTCCRA/GTCCRB register compare match, output retained at the end of the cycle) (n = 0 to 17)

The following example shows setting for triangle-wave PWM mode 2.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 19.24, 101b (triangle-wave PWM mode 2) is set.
2. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
3. Period setting
   Set a period in the GTPR register.
4. Set initial value for counter
   Set the initial value in the GTCNT counter.
5. Set GTIOCnm pin function
   Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 19.24, GTIOA[4:0] bits = 00011b and GTIOB[4:0] bits = 10011b.
6. Enable GTIOCnm pin output
   Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
7. Set buffer operation
   Set buffer operation with CCRA[1:0] and CCRB[1:0] bits in the GTBER register. In Figure 19.24, CCRA[1:0] bits = 01b and CCRB[1:0] bits = 01b.

8. Set compare match value
   Set the GTIOCnA pin changing point in the GTCCRA register and GTIOCnB pin changing point in the GTCCRB register.
9. Set buffer value
   For buffer operation, set the GTIOCnA pin and GTIOCnB pin changing points in half cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively.
   For double buffer operation, also set the GTIOCnA pin and GTIOCnB pin changing points in one cycle after the current cycle in the GTCCRD and GTCCRF registers, respectively.
10. Set count operation
    Set the GTCR.CST bit to 1 to start count operation.
11. Set buffer value by half cycle
    For buffer operation, set the GTIOCnA pin and GTIOCnB pin changing points in half cycle after the current cycle in the GTCCRC and GTCCRE registers, respectively.
    For double buffer operation, also set the GTIOCnA pin and GTIOCnB pin changing points in one cycle after the current cycle in the GTCCRD and GTCCRF registers, respectively.

Note: n = 0 to 17, m = A, B

(5) Triangle-Wave PWM Mode 3 (64-Bit Transfer at Trough)

The triangle-wave PWM mode 3 is a mode in which the period is set in the GTPR register, the GTCNT counter performs triangle-wave (full-wave) operation and a PWM waveform is output to the GTIOCnA or GTIOCnB pin (n = 0 to 17) at a compare match of the GTCCRA or GTCCRB register with buffer operation fixed. Buffer operation in triangle-wave PWM mode 3 is different from the usual buffer operation. Buffer transfer is performed from the following registers:

- GTCCRC register to GTCCRA register at the trough
- GTCCRE register to GTCCRB register at the trough
- GTCCRD register to temporary register A at the trough
- GTCCRF register to temporary register B at the trough
- Temporary register A to GTCCRA register at the crest
- Temporary register B to GTCCRB register at the crest

The pin output value can be selected from low output, high output, or toggle output separately for a compare match and for the end of the cycle according to the GTIOR register setting.

By setting the GTDTCR, GTDVU, and GTDVD registers, a compare match value for a negative-phase waveform with dead time can automatically be set to the GTCCRB register.

When the GTBER.DBRTECm (m = A, B) bit is set to 1, transfer from an intermediate buffer to the GTCCRm (m = A, B) register is repeated on a cyclic basis using the temporary register x (x = C, E) and temporary register m (m = A, B) which operate as intermediate buffers for the GTCCRx (x = C, E) and GTCCRm registers, respectively, even while buffer transfer is disabled (repeated double buffer operation function during disabling of buffer transfer). For details, see section 19.8.2.2. Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer.

Figure 19.25 shows an example of triangle-wave PWM mode 3 operation.

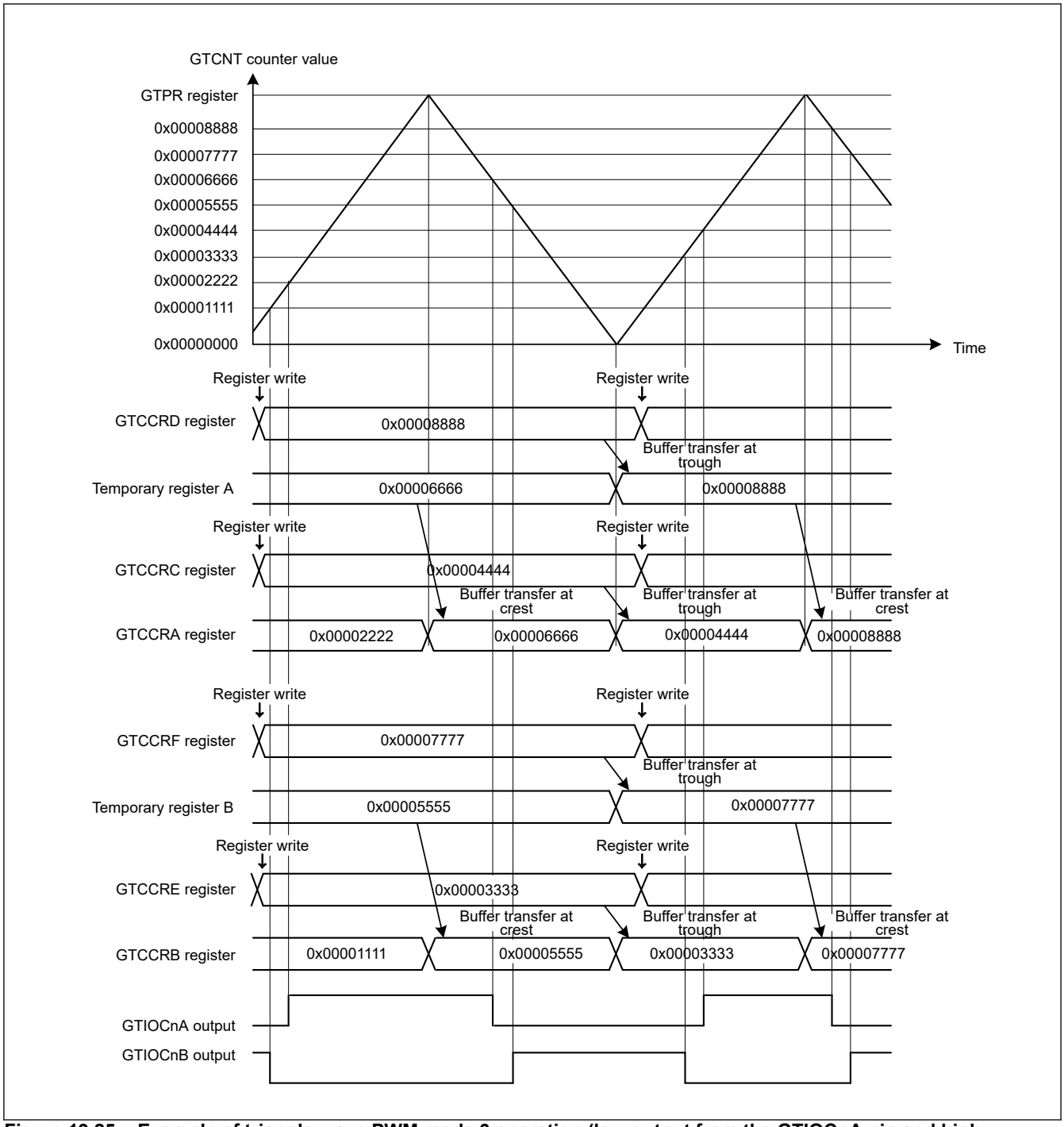


Figure 19.25 Example of triangle-wave PWM mode 3 operation (low output from the GTIOCnA pin and high output from the GTIOCnB pin at initial output, output toggled at GTCCRA/GTCCRB register compare match, output retained at the end of the cycle) (n = 0 to 17)

The following example shows setting for triangle-wave PWM mode 3.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 19.25, 110b (triangle-wave PWM mode 3) is set.
2. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
3. Period setting
   Set a period in the GTPR register.
4. Set initial value for counter

Set the initial value in the GTCNT counter.

5. Set GTIOCnm pin function
   Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 19.25, GTIOA[4:0] bits = 00011b and GTIOB[4:0] bits = 10011b.
6. Enable GTIOCnm pin output
   Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
7. Set buffer value
   Set the GTIOCnA pin changing point immediately after the count start in the GTCCRC and GTCCRD registers and the GTIOCnB pin changing point in the GTCCRE and GTCCRF registers.
8. Set forcible buffer transfer
   Set the GTBER.CCRSWT bit to 1 to forcibly transfer buffer register data.
9. Set buffer value
   For buffer operation, set the GTIOCnA pin changing point for the next cycle from the current cycle to the GTCCRC and GRCCRD registers, and set the GTIOCnB pin changing point for the next cycle from the current cycle to the GTCCRE and GTCCRF registers.
10. Set count operation
    Set the GTCR.CST bit to 1 to start count operation.
11. Set buffer value by cycle
    For buffer operation, set the GTIOCnA pin changing point for next cycle from the current cycle to the GTCCRC and GTCCRD registers and set the GTIOCnB pin changing point for the next cycle from the current cycle to the GTCCRE and GTCCRF registers.

Note: n = 0 to 17, m = A, B

19.3.4 Automatic Dead Time Setting Function

By setting the GTDTCR register, a compare match value for a negative waveform with dead time obtained by a compare match value for a positive waveform (the GTCCRA register value) and specified dead time values (the GTDVU and GTDVD registers values) can automatically be set to the GTCCRB register.

The automatic dead time setting function can be used in sawtooth-wave one-shot pulse mode and all the triangle PWM modes.

Dead time can be separately set for the first half and second half of a waveform. Dead time for the changing point in the first half of a negative waveform is set in the GTDVU register and that in the second half is set in the GTDVD register. The same dead time can also be set for the first and second halves by setting the GTDTCR.TDFER bit to 1.

The GTDBU register can be used as a buffer register of the GTDVU register, and the GTDBD register can be used as a buffer register of the GTDVD register. Buffer transfer is performed at the end of the cycle (in sawtooth-wave mode: either of an overflow of the GTCNT counter (up-counting), an underflow (down-counting), or the GTCNT counter clearing in triangle-wave mode: a trough).

The change point of the negative-phase waveform, which is automatically calculated, is obtained by reading the GTCCRB register. Writing to the GTCCRB register is prohibited when the automatic dead time setting function is used.

Do not set the dead-time that makes the change point of the waveform exceeding the count period. When a dead-time setting which would generate a dead-time error is made, adjust the change points of the positive- and negative-phase waveforms to generate waveforms with secured dead-time as shown in Table 19.8. The adjusted change point of the negative-phase waveform is automatically set in the GTCCRB register. An internal signal is used to determine the change point of the positive-phase waveform, thus the value of the GTCCRA register is not updated by the adjusted value.

In sawtooth-wave one-shot pulse mode, if the order of the change point becomes inconsistent by adjustment of the waveform change point due to occurrence of dead time errors, or if the change point exceeds the count period even after the adjustment, the complementary relation between the positive- and negative-phases cannot be guaranteed.

In triangle-wave PWM mode, if dead-time exceeds the count period by setting 0x00000000 or a value greater than or equal to the setting value of the GTPR register is set in the GTCCRA register, output change is controlled by the output protection function (see section 19.8.4. GTIOCnm Protection Function for GTIOCnm Pin Output (n = 0 to 17, m = A, B)). When GTCCRA register is greater than or equal to [GTPR register + GTDVm (m = U, D) register], [GTPR register − 1] is set in the GTCCRB register as the upper limit.

Automatic setting for a dead time value to the GTCCRB register is performed at the next count clock after the register value for calculating the automatic setting value is updated. In triangle-wave mode, it can also be done at the next count clock from the current crest.

Table 19.8 Adjustment of the waveform change point when a dead-time error occurs

Mode	Count direction	Period	Condition for dead-time error	Change point of the positive-phase waveform after adjustment	Change point of the negative-phase waveform after adjustment
Sawtooth-wave one-shot pulse mode	Up-counting	First half	GTCCRA − GTDVU < 0	GTDVU	0
		Second half	GTCCRA + GTDVD > GTPR	GTPR − GTDVD	GTPR
	Down-counting	First half	GTCCRA + GTDVU > GTPR	GTPR − GTDVU	GTPR
		Second half	GTCCRA − GTDVD < 0	GTDVD	0
Triangle-wave PWM mode 1/2/3	Up-counting	(First half)	GTCCRA − GTDVU ≤ 0	GTDVU + 1	1
	Down-counting	(Second half)	GTCCRA − GTDVD < 0	GTDVD	0

Figure 19.26 to Figure 19.28 show examples of automatic dead time setting function operation.

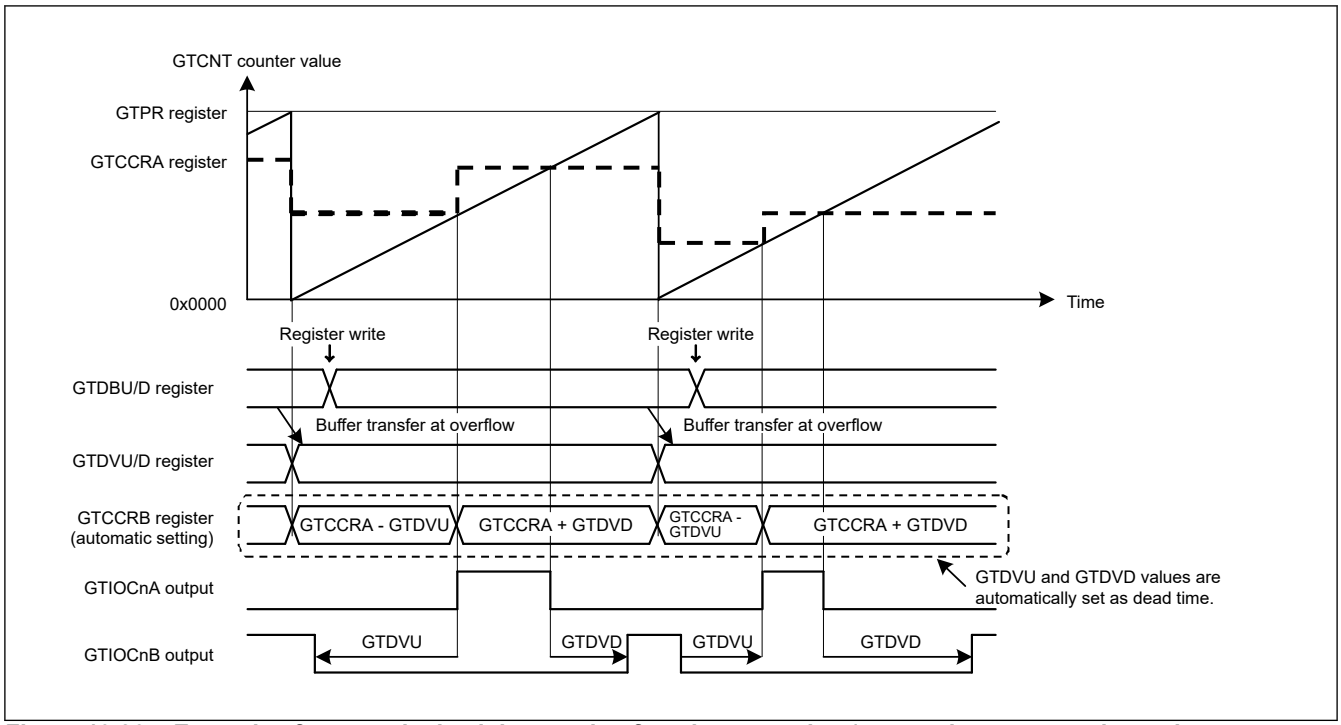


Figure 19.26 Example of automatic dead time setting function operation (sawtooth-wave one-shot pulse mode, up-counting, GTDVU and GTDVD registers set to buffer operation, active-level: high) (n = 0 to 17)

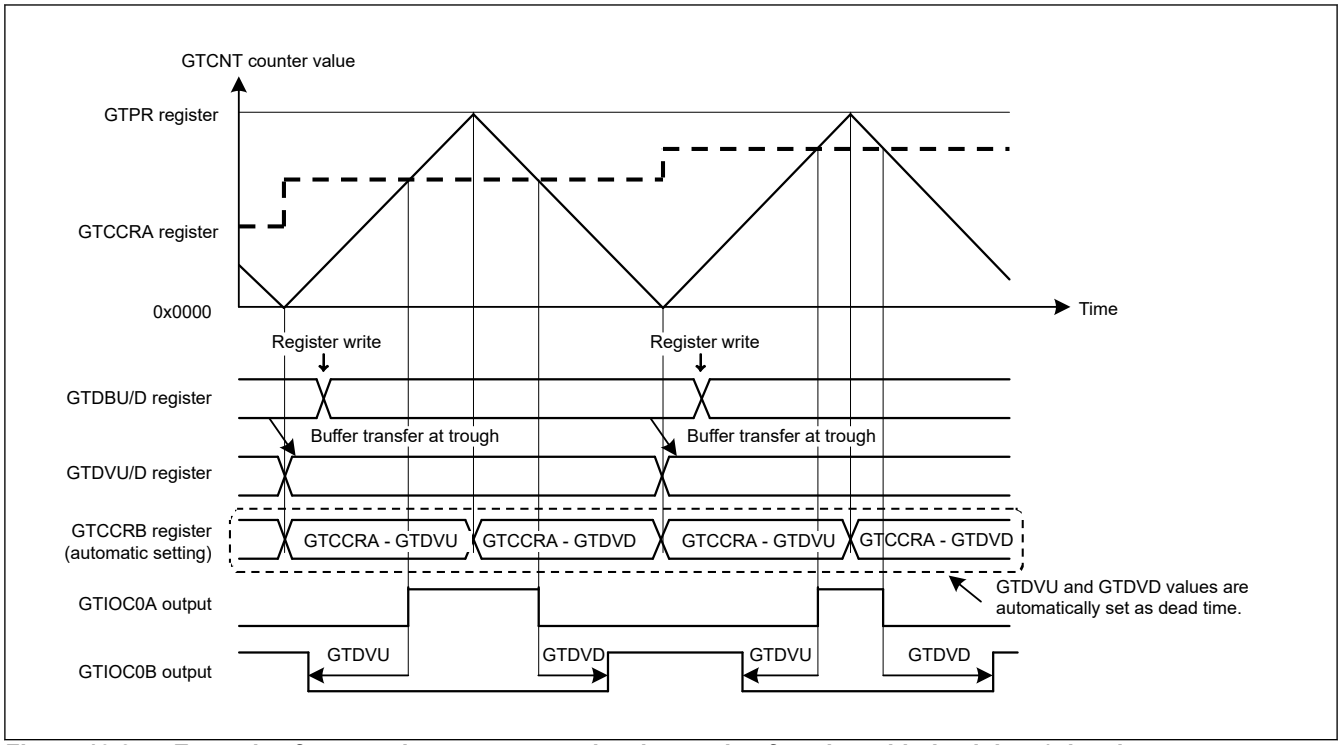


Figure 19.27 Example of automatic compare-match value setting function with dead time (triangle-wave PWM mode 1, GTDVU and GTDVD registers set to buffer operation, active-level: high)

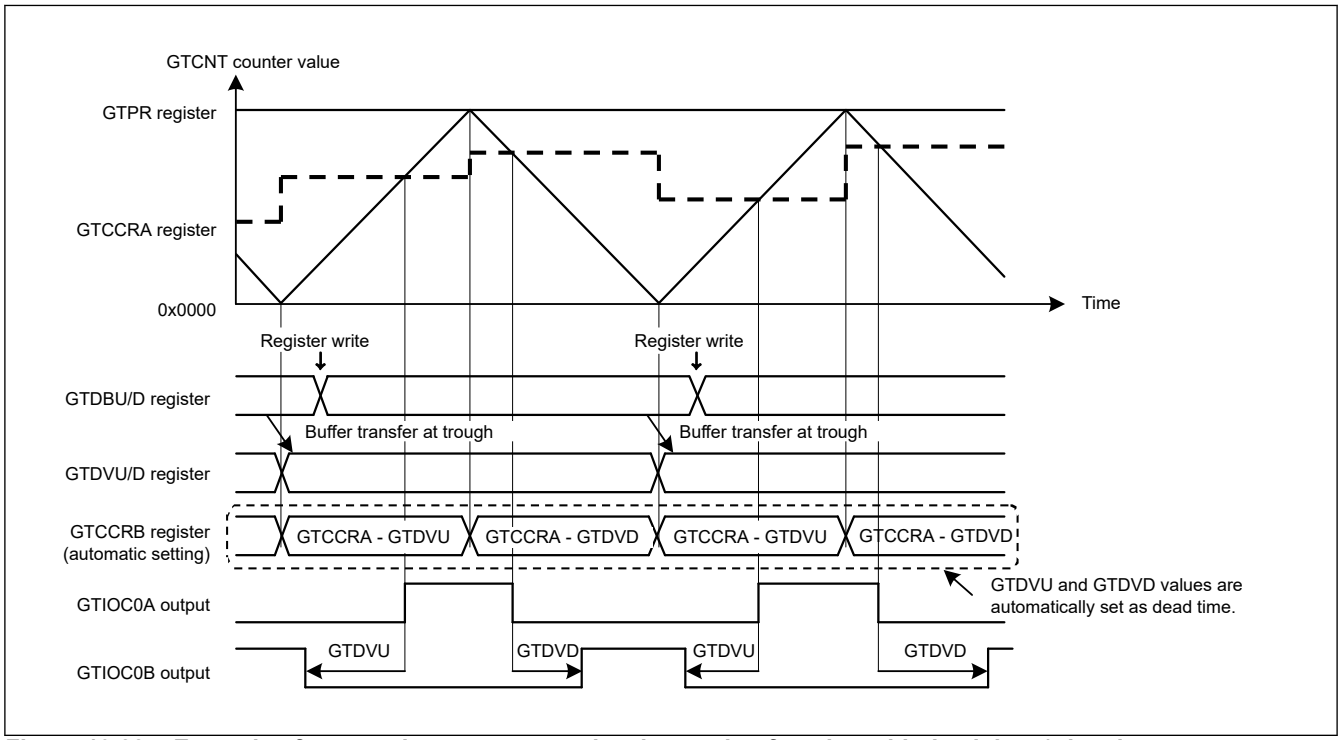


Figure 19.28 Example of automatic compare-match value setting function with dead time (triangle-wave PWM mode 2 or 3, GTDVU and GTDVD registers set to buffer operation, active-level: high)

The following example shows setting for automatic dead time setting function (sawtooth-wave one-shot pulse mode, triangle-wave PWM mode 3).

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 19.26, 001b (sawtooth-wave one-shot pulse mode) is set. In Figure 19.28, 110b (triangle-wave PWM mode 3) is set.

2. Select count direction
   Select the count direction (up or down) with the GTUDDTYC register. In Figure 19.26, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
6. Set GTIOCnm pin function
   Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register. In Figure 19.26 and Figure 19.28, GTIOA[4:0] bits = 00011b and GTIOB[4:0] bits = 10011b.
7. Enable GTIOCnm pin output
   Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
8. Set buffer value for compare match
   Set the GTIOCnA pin changing point immediately after the count start in the GTCCRC and GTCCRD registers.
9. Set forcible buffer transfer for compare match
   Set the GTBER.CCRSWT bit to 1 to transfer buffer register data forcibly to the GTCCRA register.
10. Set buffer value for compare match
    Set the GTIOCnA pin changing point in one cycle after the count start in the GTCCRC and GTCCRD registers.
11. Set automatic dead time setting function
    Set GTDTCR.TDE to 1 to enable the automatic dead time setting function.
12. Set buffer operation for dead time setting
    Set buffer operation with the TDBUE and TDBDE bits in the GTDTCR register.
13. Set dead time value
    Set the first half dead time value in the GTDVU register and the second half dead time in the GTDVD register. When the GTDVU register is set with the GTDTCR.TDFER bit set to 1, the same value is also set to the GTDVD register, the same dead time value can be set for the first and second halves.
14. Set buffer value for dead time
    For buffer operation, set the first half dead time in one cycle after the current cycle in the GTDBU register and the second half dead time in the GTDBD register.
15. Set count operation
    Set the GTCR.CST bit to 1 to start count operation.
16. Set buffer value by cycle
    Set the GTIOCnA pin changing point for the next cycle from the current cycle to the GTCCRC and GTCCRD registers. When the dead time register is used for buffer operation, set the dead time value in the first half of the next cycle from the current cycle to the GTDBU register and the dead time value in the second half to the GTDBD register.

Note: n = 0 to 17, m = A, B

The following example shows setting for automatic dead time setting function (triangle-wave PWM mode 1 or 2).

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 19.27, 100b (triangle-wave PWM mode 1) is set.
   In Figure 19.28, 101b (triangle-wave PWM mode 2) is set.
2. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
3. Period setting
   Set a period in the GTPR register.
4. Set initial value for counter
   Set the initial value in the GTCNT counter.

5. Set GTIOCnm pin function
   Set the GTIOCnm pin function with the GTIOA[4:0] and GTIOB[4:0] bits in the GTIOR register.
   In Figure 19.27 and Figure 19.28, GTIOA[4:0] bits = 00011b and GTIOB[4:0] bits = 10011b.
6. Enable GTIOCnm pin output
   Set to enable the GTIOCnm pin output with the OAE and OBE bits in the GTIOR register.
7. Set buffer operation for compare match
   Set buffer operation with the GTBER.CCRA[1:0] bits.
8. Set the value for compare match
   Set the GTIOCnA pin changing point to the GTCCRA register.
9. Set buffer value for compare match
   For buffer operation, set the GTIOCnA pin changing point in one cycle after the current cycle (in triangle-wave PWM mode 1) or half cycle after the current cycle (in triangle-wave PWM mode 2) in the GTCCRC register. For double buffer operation, also set the GTIOCnA pin changing point in two cycles after the current cycle (in triangle-wave PWM mode 1) or one cycle after the current cycle (in triangle-wave PWM mode 2) in the GTCCRD register.
10. Set automatic dead time setting function
    Set GTDTCR.TDE to 1 to enable the automatic dead time setting function.
11. Set buffer operation for dead time setting
    Set buffer operation with the TDBUE and TDBDE bits in the GTDTCR register.
12. Set dead time value
    Set the first half dead time value in the GTDVU register and the second half dead time in the GTDVD register. When the GTDVU register is set with the GTDTCR.TDFER bit set to 1, the same value is also set to the GTDVD register, the same dead time value can be set for the first and second halves.
13. Set buffer value for dead time
    For buffer operation, set the first half dead time in one cycle after the current cycle in the GTDBU register and the second half dead time in the GTDBD register.
14. Set count operation
    Set the GTCR.CST bit to 1 to start count operation.
15. Set buffer value by cycle
    For buffer operation of compare match register, set the GTIOCnA pin changing point in one cycle after the current cycle (in triangle-wave PWM mode 1) or half cycle after the current cycle (in triangle-wave PWM mode 2) in the GTCCRC register. For double buffered operation of compare match register, set the GTICCnA pin changing point in two cycles after the current cycle (in triangle-wave PWM mode 1) or one cycle after the current cycle (in triangle-wave PWM mode 2) in the GTCCRD register.
    In the same way, set the dead time value in the first half of the cycle after the current cycle in the GTDBU register and the dead time in the second half in the GTDBD register.

Note: n = 0 to 17, m = A, B

19.3.5 Count Direction Changing Function

The count direction of the GTCNT counter can be changed by modifying the GTUDDTYC.UD bit.

In sawtooth-wave mode, if the UD bit is modified during count operation, the count direction is changed at an overflow (when modified during up-counting) or an underflow (when modified during down-counting). If the UD bit is modified while count operation is stopped and the GTUDDTYC.UDF bit is 0, the UD bit modification is not reflected at the start of counting and the count direction is changed at an overflow or an underflow. If the UDF bit is set to 1 while count operation is stopped, the UD bit value at that time is reflected at the start of counting.

In triangle-wave mode, changing the value of the GTUDDTYC.UD bit during counting does not switch the direction of counting. Likewise, changing the value of the GTUDDTYC.UD bit while the GTUDDTYC.UDF bit is 0 and counting is stopped does not actually update the value of the bit. If 1 is written to the GTUDDTYC.UDF bit while counting is stopped, the value of the GTUDDTYC.UD bit at that time is reflected from the time counting is started.

When counting direction is switched during count operation in sawtooth-wave mode, the value of GTPR register after the start of up-counting is reflected in the count period in up-counting operation, and the value of GTPR register after the start of down-counting is reflected in the count period in down-counting operation.

Figure 19.29 shows an example of the count direction changing function operation.

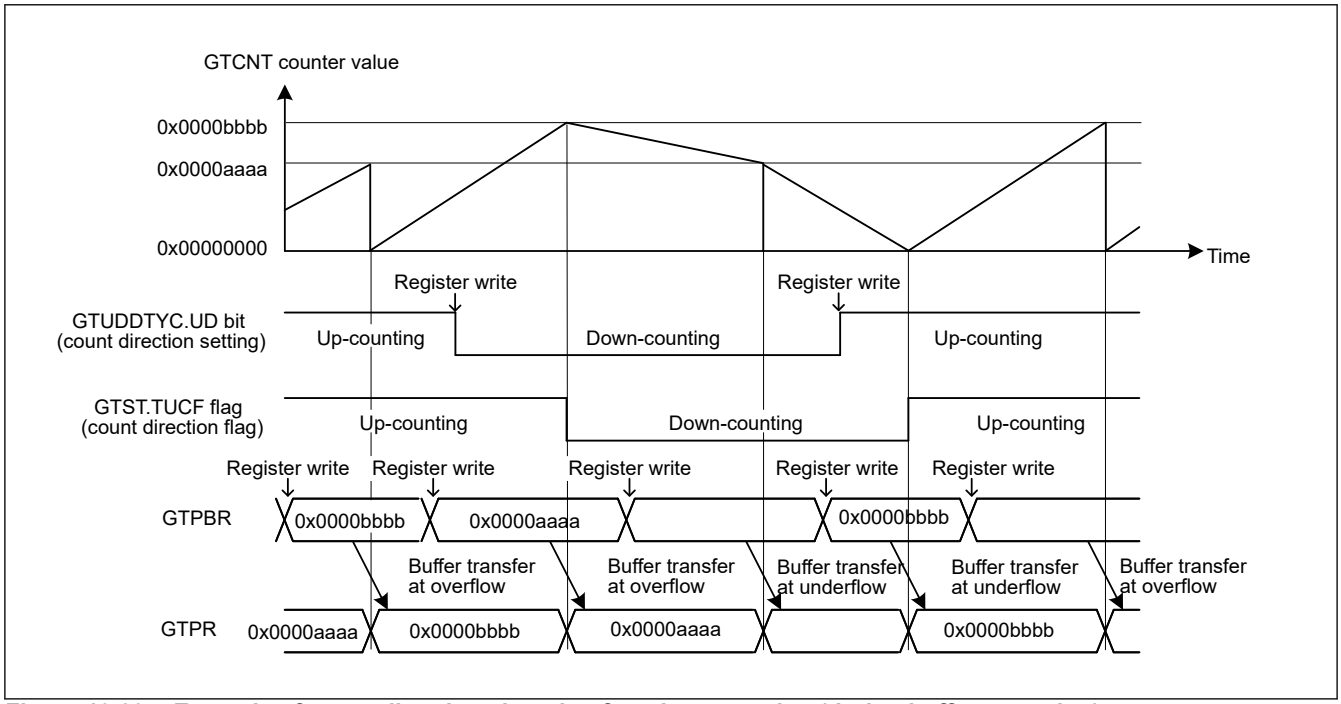


Figure 19.29 Example of count direction changing function operation (during buffer operation)

19.3.6 Function of Output Duty 0% and 100%

Changing the value of the GTUDDTYC.OADTY[1:0] bits and GTUDDTYC.OBDTY[1:0] bits specifies the output duty setting on the GTIOCnA and GTIOCnB pins to 0% or 100% (n = 0 to 17).

In sawtooth-wave mode, if the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed during the count operation, the output duty-cycle setting is reflected at an overflow (when changed during up-counting) or an underflow (when changed during down-counting). When the GTUDDTYC.OADTYF bit or GTUDDTYC.OBDTYF bit is 0 while count operation is stopped and the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed, the output duty-cycle setting changed at the start of counting is not reflected, but the output duty-cycle setting changed at an overflow or underflow is reflected. When the OADTYF bit or OBDTYF bit is 1 while count operation is stopped and the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed, the value of the OADTY[1:0] bits or OBDTY[1:0] bits at that time is reflected at the start of counting.

In triangle-wave mode, if the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed during the count operation, in the case of OABDTYT bit = 0, the output duty-cycle setting is reflected at trough of triangle-wave and in the case of OABDTYT bit = 1, the output duty-cycle setting is reflected at trough and crest of triangle-wave.

When the OADTYF bit or OBDTYF bit is 0 while count operation is stopped and the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed, the output duty-cycle setting changed at the start of counting is not reflected, but the output duty-cycle setting changed at trough is reflected. When the OADTYF bit or OBDTYF bit is 1 while count operation is stopped and the value of the OADTY[1:0] bits or OBDTY[1:0] bits is changed, the value of the OADTY[1:0] bits or OBDTY[1:0] bits at that time is reflected at the start of counting.

During the operation under the settings of duty cycle for 0% or 100%, compare match operation inside the GPT continues to perform interrupt outputs and buffer operations.

When the output duty setting is changed from 0% or 100% due to a compare match, the GTIOR.GTIOA[3:2] bits and the GTUDDTYC.OADTYR bits determine the level output on the GTIOCnA pin and GTIOR.GTIOB[3:2] bits and GTUDDTYC.OBDTYR bits determine the level output on the GTIOCnB pin at the end of the cycle. When the GTIOA[3:2] and GTIOB[3:2] bits are set as 01b, output is driven low at the end of the cycle, and when set as 10b, output is driven high at the end of the cycle.

The output is retained at the end of the cycles when the GTIOA[3:2] or GTIOB[3:2] bits are 00b, and toggled at the end of the cycles when the GTIOA[3:2] or GTIOB[3:2] bits are 11b. A value to be output is selectable in the OADTYR or OBDTYR bit when the corresponding GTIOA[3:2] or GTIOB[3:2] bits are either 00b or 11b.

In triangle-wave mode and OABDTYT bit = 1, output duty-cycle setting is reflected at trough and crest of triangle-wave.

When masking by a duty cycle 0% or 100% release at crest of triangle-wave, the level output on the GTIOCnA pin and GTIOCnB pin is determined by value at compare match output.

Table 19.9 lists the output values at the end of the cycle when the output setting is changed from duty 0% or 100% to compare match.

Table 19.9 Output value after release of duty 0% or 100%

GTIOR.GTIOm[3:2]	Value at compare match output at the end of the cycle when masking by a duty cycle 0% or 100%	GTUDDTYC.OmDTYR bit at duty 0% setting		GTUDDTYC.OmDTYR bit at duty 100% setting	
		0	1	0	1
00b (output retained at the end of the cycle)	0	0	0	1	0
	1	0	1	1	1
01b (low output at the end of the cycle)	—	0	0	0	0
10b (high output at the end of the cycle)	—	1	1	1	1
11b (toggle output at the end of the cycle)	0	1	1	0	1
	1	1	0	0	0

Note: m = A, B

Figure 19.30 shows an example of operation of the output of the duty cycle 0% or 100%.

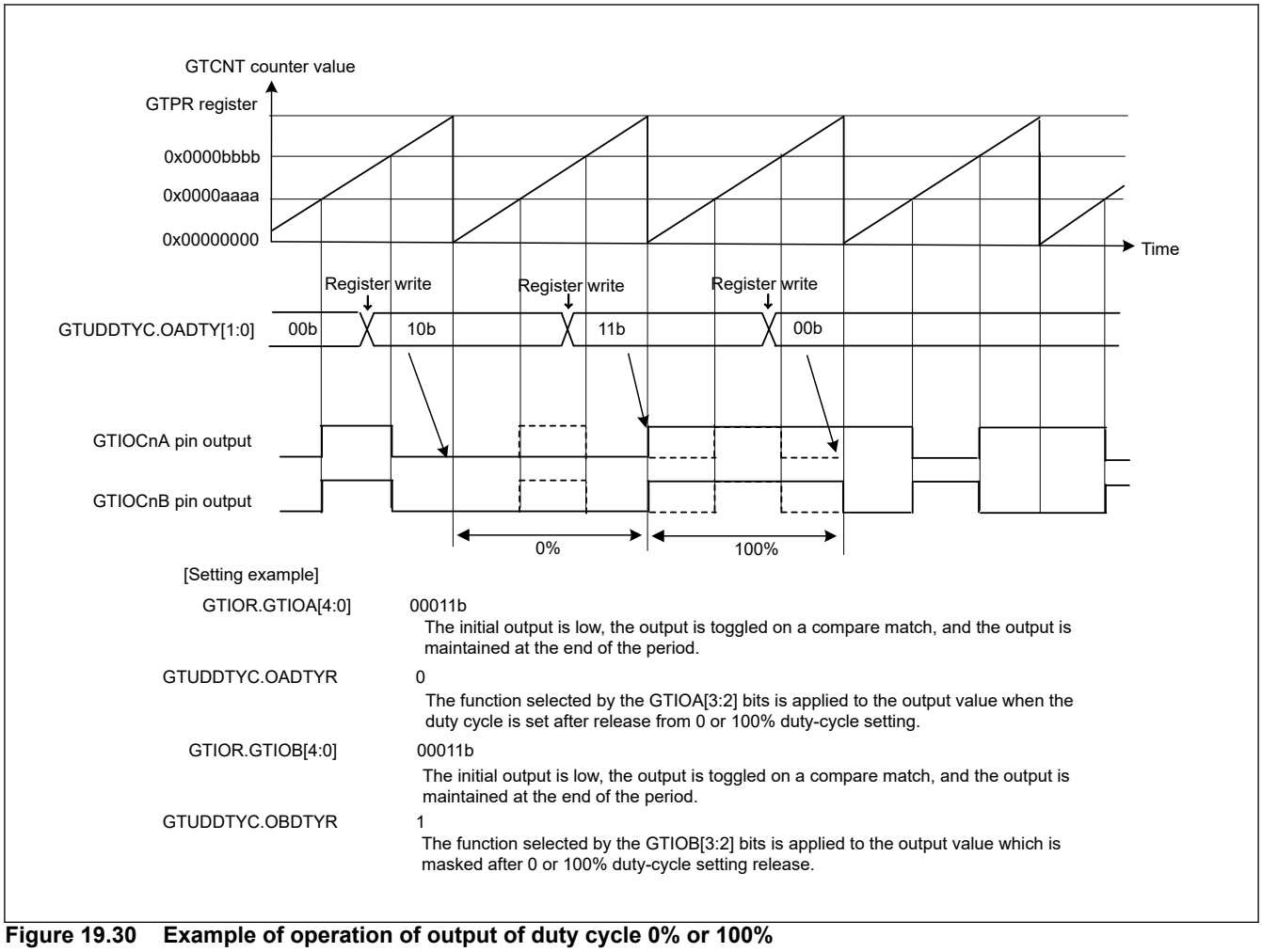


Figure 19.30 Example of operation of output of duty cycle 0% or 100%

19.3.7 Hardware Count Start/Count Stop and Clear Operation

The GTCNT counter can be started, stopped, or cleared by hardware sources. There are 3 types of hardware sources, including external trigger input, ELC event input, and GTIOCnm pin input (n = 0 to 17, m = A, B).

19.3.7.1 Hardware Start Operation

The GTCNT counter can be started by a hardware source. Select a hardware source to start counting using the GTSSR register, and enable to start counting.

Figure 19.31 shows an example of count start operation by a hardware source.

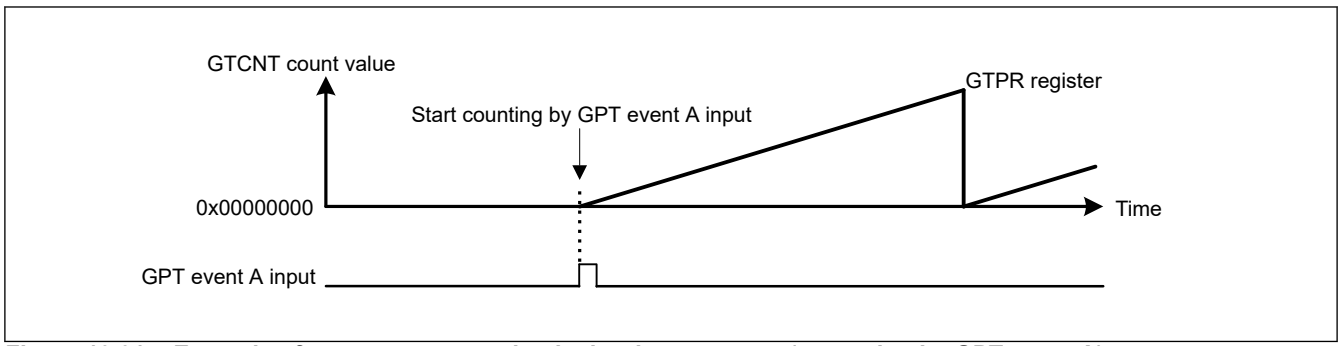


Figure 19.31 Example of count start operation by hardware source (at starting by GPT event A)

The following example shows setting for count start operation by hardware source.

1. Set operating mode

Set the operating mode with the GTCR.MD[2:0] bits.
In Figure 19.31, 000b (sawtooth-wave PWM mode 1) is set.

2. Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 19.31, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
Set a period in the GTPR register.
5. Set initial value for counter
Set the initial value in the GTCNT counter.
In Figure 19.31, 0x00000000 is set.
6. Set hardware count start
Select a hardware source to start count operation with the GTSSR register.
In Figure 19.31, GTSSR.SSELCA bit = 1.
7. Start hardware source operation
Set operation of the hardware source selected by the GTSSR register and start counting.
In Figure 19.31, the GPT event A input operation is set.

19.3.7.2 Hardware Stop Operation

The GTCNT counter can be stopped by a hardware source. Select a hardware source to stop counting with the GTPSR register, and enable to stop counting.

Figure 19.32 shows an example of count stop operation by a hardware source. In this example, the count operation is stopped by the GPT event A input, and is restarted by the GPT event B input.

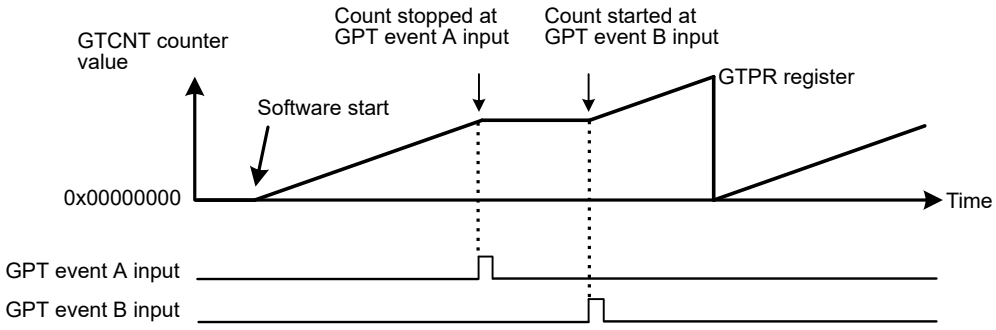


Figure 19.32 Example of count stop operation by hardware source (started by software, stopped at GPT event A input, and restarted at GPT event B input)

The following example shows setting for count stop operation by hardware source.

1. Set operating mode
Set the operating mode with the GTCR.MD[2:0] bits.
In Figure 19.32, 000b (sawtooth-wave PWM mode) is set.
2. Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 19.32, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
Set a period in the GTPR register.
5. Set initial value for counter

Set the initial value in the GTCNT counter.
In Figure 19.32, 0x00000000 is set.

6. Set hardware count start
Select a hardware source for starting count operation with the GTSSR register.
In Figure 19.32, GTSSR.SSELCB bit = 1.

7. Set hardware count stop
Select a hardware source for starting count operation with the GTPSR register.
In Figure 19.32, GTPSR.PSELCA bit = 1.

8. Start hardware source operation
Set operation of the hardware source selected by the GTSSR or GTPSR register then start and stop counting.
In Figure 19.32, the GPT event A and GPT event B input operations are set.

Figure 19.33 shows an example of count start/stop operation by a hardware source.

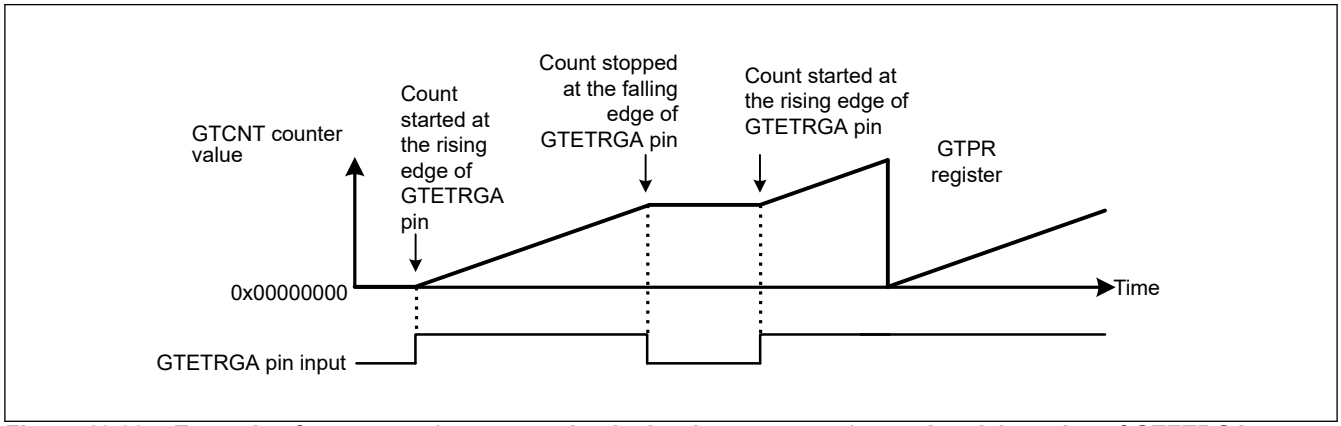


Figure 19.33 Example of count start/stop operation by hardware source (started at rising edge of GTETRGA pin input, stopped at falling edge of GTETRGA pin input)

The following example shows setting for count start/stop operation by hardware source.

1. Set operating mode
Set the operating mode with the GTCR.MD[2:0] bits.
In Figure 19.33, 000b (sawtooth-wave PWM mode) is set.

2. Set count direction
Select the count direction (up or down) with the GTUDDTYC register.
In Figure 19.33, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.

3. Select count clock
Select the count clock with GTCR.TPCS[3:0] bits.

4. Period setting
Set a period in the GTPR register.

5. Set initial value for counter
Set the initial value in the GTCNT counter.
In Figure 19.33, 0x00000000 is set.

6. Set hardware count start
Select a hardware source for starting count operation with the GTSSR register.
In Figure 19.33, GTSSR.SSGTRGAFR[1:0] bits = 01b.

7. Set hardware count stop
Select a hardware source for stopping count operation with the GTPSR register.
In Figure 19.33, GTPSR.PSGTRGAFR[1:0] bits = 10b.

8. Start hardware source operation
Set operation of the hardware source selected by the GTSSR or GTPSR register then start and stop counting.
In Figure 19.33, the GTETRGA pin operation is set.

19.3.7.3 Hardware Clear Operation

The GTCNT counter can be cleared by a hardware source. Select a hardware source to clear the counter with the GTCSR register, and enable to clear the counter.

Note: The GTPn_OVF/GTPn_UDF interrupt (overflow/underflow interrupt) is not generated when the GTCNT counter is cleared by a hardware source or by software.

Figure 19.34 and Figure 19.35 show examples of the GTCNT counter clearing operation by a hardware source.

In this example, the GTCNT counter is started at the GPT event A input, and is stopped/cleared at the GPT event B input.

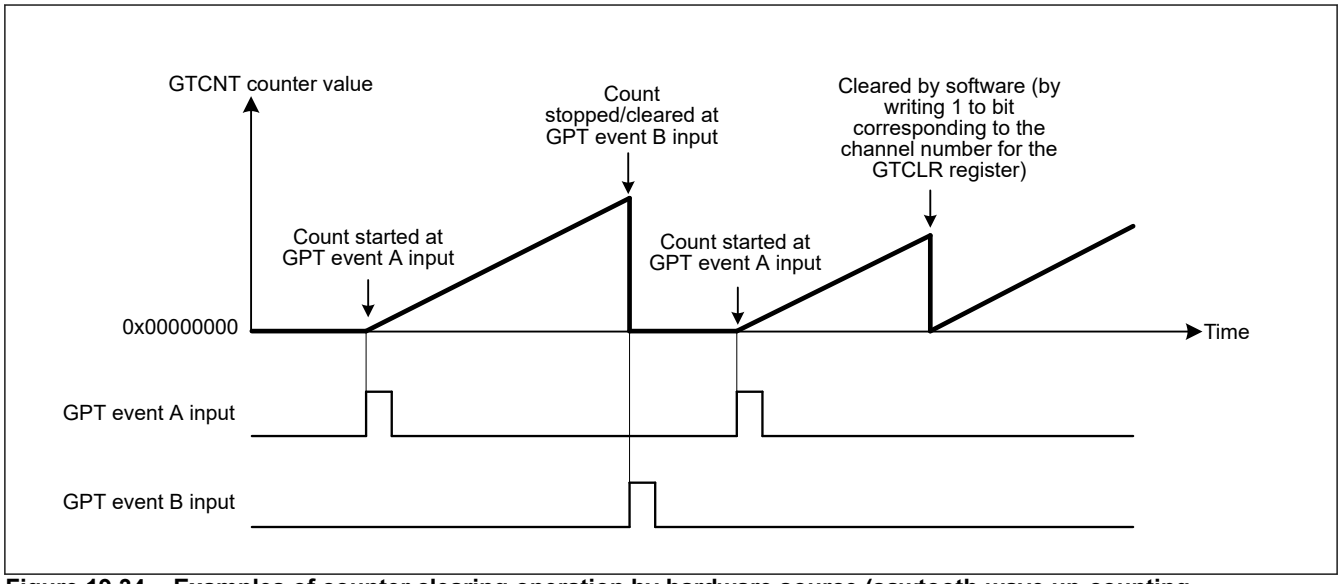


Figure 19.34 Examples of counter clearing operation by hardware source (sawtooth-wave up-counting, started at GPT event A input, and count stopped/counter cleared at GPT event B input)

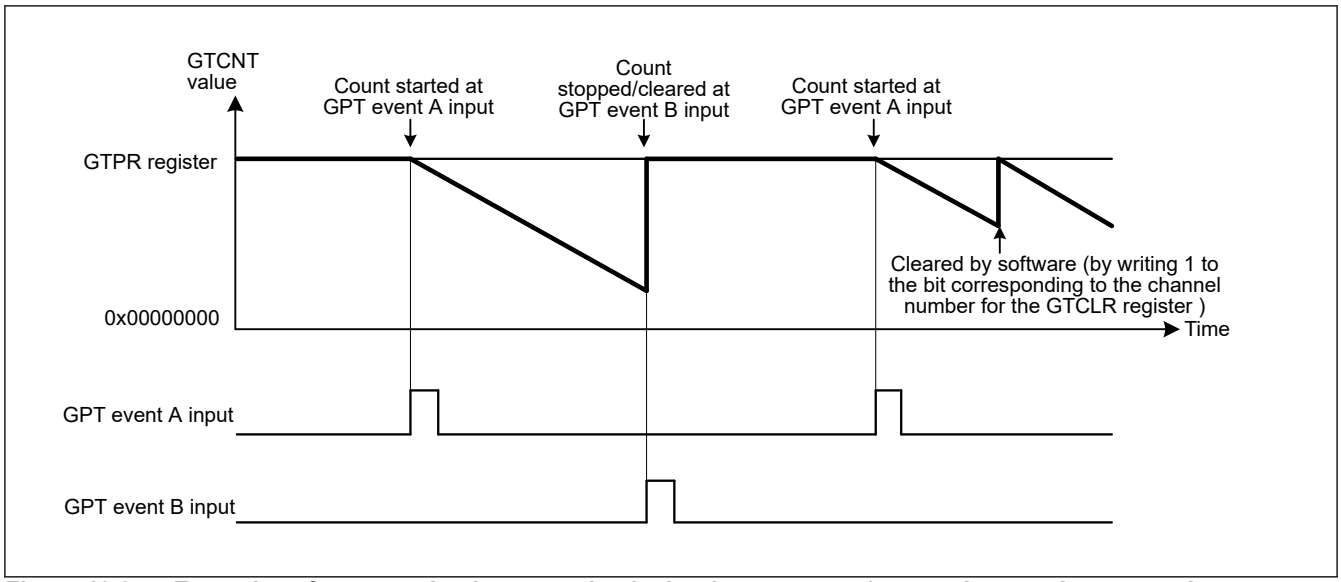


Figure 19.35 Examples of counter clearing operation by hardware source (sawtooth-wave down-counting, started at GPT event A input, and count stopped/counter cleared at GPT event B input)

The following example shows setting for counter clearing operation by hardware source.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits.
   In Figure 19.34 and Figure 19.35, 000b (sawtooth-wave PWM mode 1) is set.
2. Set count direction

Select the count direction (up or down) with the GTUDDTYC register.
In Figure 19.34, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
In Figure 19.35, the lower 2 bits of the GTUDDTYC register is set to 10b and then to 00b for down-counting.

3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
4. Period setting
   Set a period in the GTPR register.
5. Set initial value for counter
   Set the initial value in the GTCNT counter.
   In Figure 19.34, 0x00000000 is set.
   In Figure 19.35, the GTPR register value is set.
6. Set hardware count start
   Select a hardware source for starting count operation with the GTSSR register and wait for count start by the hardware source.
   In Figure 19.34 and Figure 19.35, GTSSR.SSELCA bit = 1.
7. Set hardware count stop
   Select a hardware source for stopping count operation with the GTPSR register and wait for count stop by the hardware source.
   In Figure 19.34 and Figure 19.35, GTPSR.PSELCB bit = 1.
8. Set hardware count clear
   Select a hardware source to clear the counter with the GTCSR register and wait for count clearing by the hardware source.
   In Figure 19.34 and Figure 19.35, GTPSR.CSELCB bit = 1.
9. Start hardware source operation
   Set operation of the hardware source selected by the GTSSR, GTPSR, or GTCSR register then start, stop, or clear counting.
   In Figure 19.34 and Figure 19.35, the GPT event A and B inputs operations are set.

The GTPn_OVF/GTPn_UDF interrupt (overflow/underflow interrupt) does not occur when the counter is cleared by a hardware source. In the same way, the GTPn_OVF/GTPn_UDF interrupt does not occur when the counter is cleared by software.

Figure 19.36 shows the relationship between counter clearing by a hardware source and the GTPn_OVF interrupt.

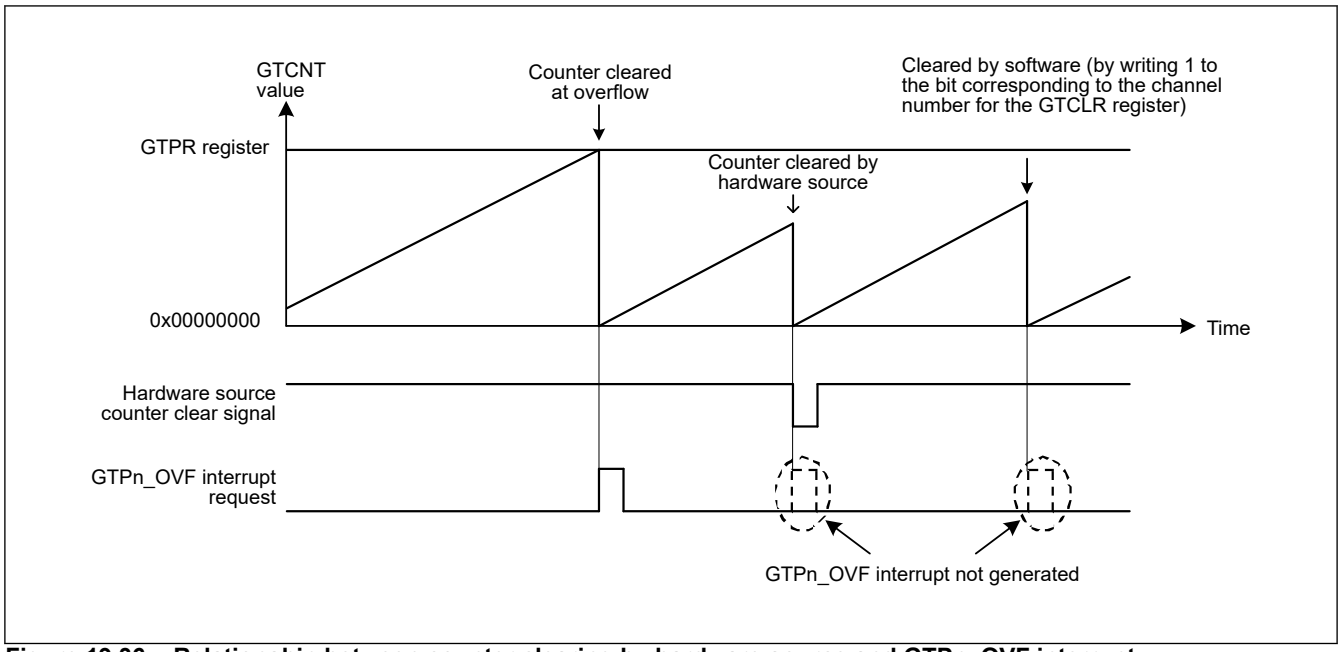


Figure 19.36 Relationship between counter clearing by hardware source and GTPn_OVF interrupt

19.3.8 Hardware Count Switch Operation

In the triangle-wave PWM mode, the GTCNT counter can be switched by a hardware source. Select the count switch operation with the GTCR.SWMD[2:0] bits and select a hardware source to switch the counter with the GTSWSR register and enable to switch the counter.

When the offset reload is enabled by the GTCR.SWMD[2:0] bits, the offset value of the GTCNT counter reloaded by count switch is set by the GTSWOS register in the range of 0 to GTPR.

If the count switch and counter clear occur at the same time, the counter clear has priority.

Figure 19.37 to Figure 19.40 show examples of the GTCNT count switch operation by a hardware source.

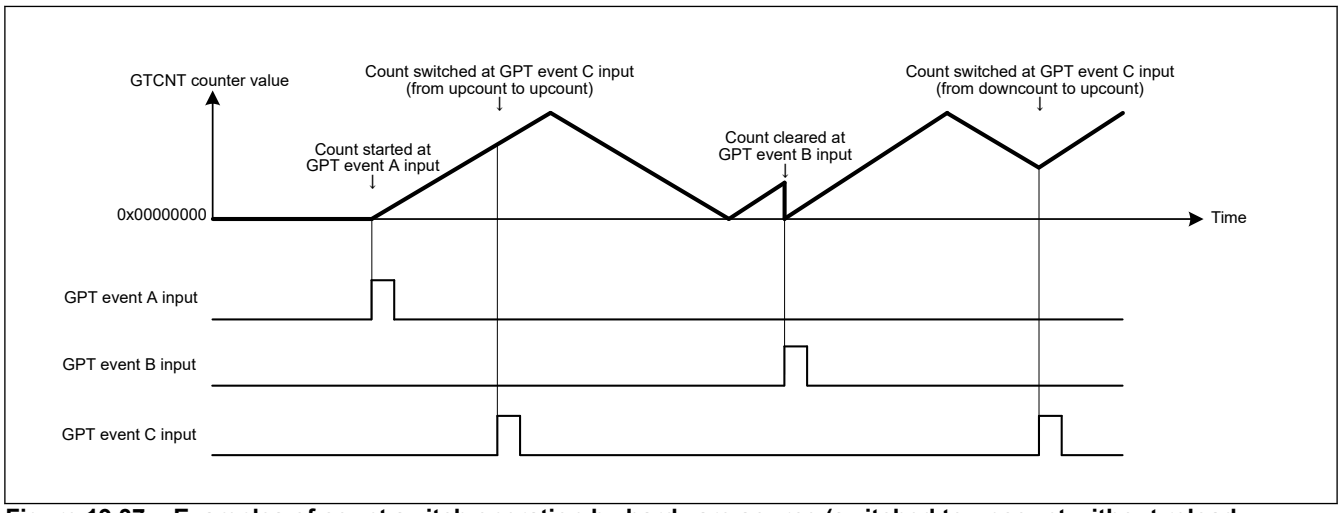


Figure 19.37 Examples of count switch operation by hardware source (switched to upcount without reload at GPT event C input)

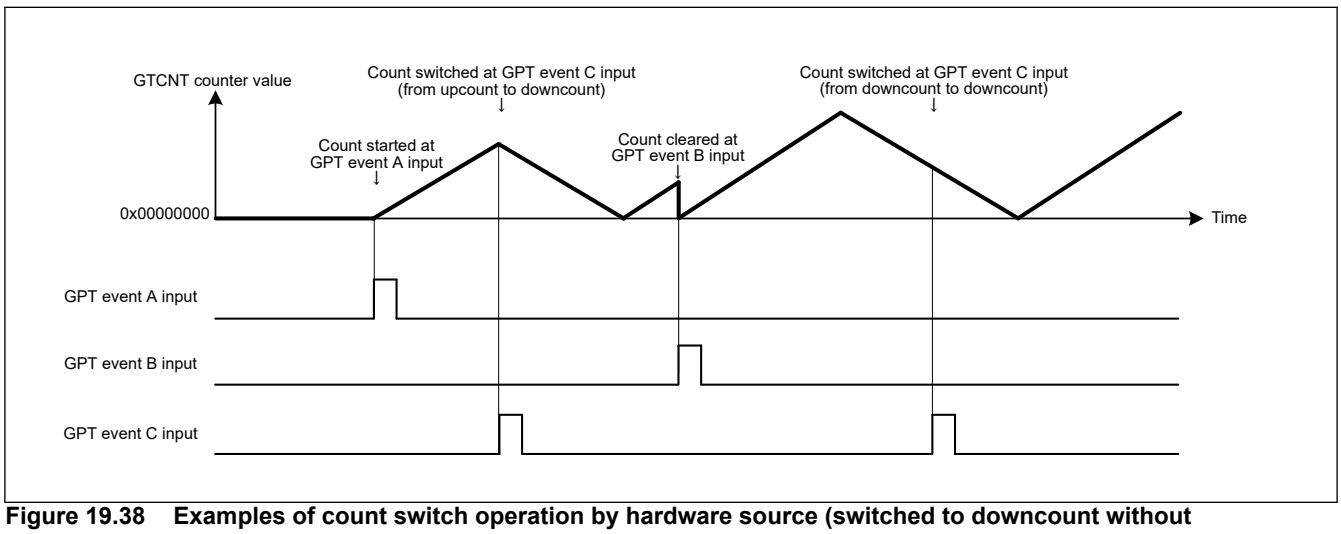


Figure 19.38 Examples of count switch operation by hardware source (switched to downcount without reload at GPT event C input)

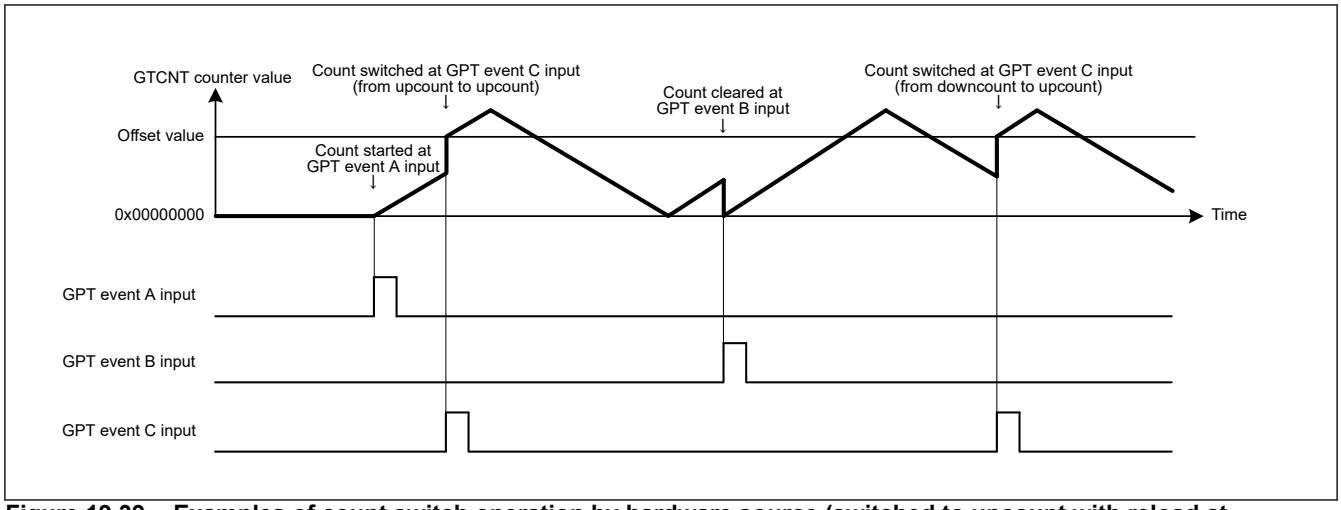


Figure 19.39 Examples of count switch operation by hardware source (switched to upcount with reload at GPT event C input)

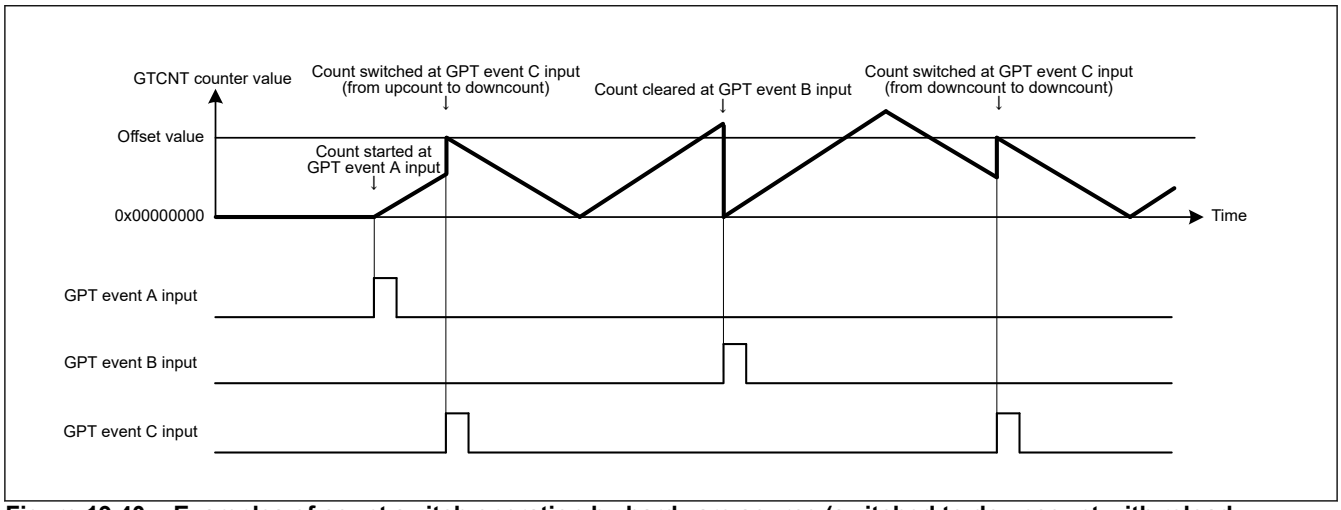


Figure 19.40 Examples of count switch operation by hardware source (switched to downcount with reload at GPT event C input)

The following procedure shows the setting example. In this example, the GTCNT counter started at the GTP event A input, is stopped/cleared at the GTP event B input, and is switched at the GTP event C input.

1. Set operating mode
   Set the operating mode to the triangle-wave PWM mode with the GTCR.MD[2:0] bits.
   In Figure 19.37, Figure 19.38, Figure 19.39, and Figure 19.40, 100b (triangle-wave PWM mode 1) is set.
2. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits.
3. Period setting
   Set a period in the GTPR register.
4. Set initial value for counter
   Set the initial value in the GTCNT counter.
   In Figure 19.37, Figure 19.38, Figure 19.39, and Figure 19.40, 0x00000000 is set.
5. Set switch mode
   Set the switch mode with the GTCR.SWMD[2:0] bits.
   In Figure 19.37, 101b is set, in Figure 19.38, 110b is set, in Figure 19.39, 001b is set, in Figure 19.40, 010b is set.
6. Set hardware count start
   Select a hardware source for starting count operation with the GTSSR register and wait for count start by the hardware source.
   In Figure 19.37, Figure 19.38, Figure 19.39, and Figure 19.40, GTSSR.SSELCA bit = 1.

7. Set hardware count stop
   Select a hardware source for stopping count operation with the GTPSR register and wait for count stop by the hardware source.
8. Set hardware count clear
   Select a hardware source to clear the counter with the GTCSR register and wait for counter clearing by the hardware source.
   In Figure 19.37, Figure 19.38, Figure 19.39, and Figure 19.40, GTCSR.CSELCB bit = 1.
9. Set hardware count switch
   Select a hardware source to switch the counter with the GTSWSR register and wait for count switch by the hardware source.
   In Figure 19.37, Figure 19.38, Figure 19.39, and Figure 19.40, GTSWSR.WSELCC bit = 1.
10. Set hardware source operation
    Set operation of the hardware source selected with the GTSSR register, GTPSR register, GTCSR register, or GTSWSR register and start, stop, clear, or switch the counter.
    In Figure 19.37, Figure 19.38, Figure 19.39, and Figure 19.40, the GPT event A, GPT event B, and GPT event C input operations are set.

The compare match operation is valid even when the count switch and the compare match occur at the same time. Similarly, even when the count switch occurs at the crest (GTCNT counter matches GTPR register value during upcounting) or trough (GTCNT counter is 0 during downcounting) of triangle-waves, overflow or underflow interrupt is output and the buffer transfer of crest or trough functions normally.

If the count switch is performed at the crest or trough without the offset reload, the crest or trough occurs and then the count direction changes to downcount or upcount regardless of the count switch setting. Also, if the count switch is performed at the count clear state (GTCNT counter value is 0 during upcount) without the offset reload, the count direction continues upcount regardless of the count switch setting.

If the switch mode is set to Switch to upcount with offset reload with the SWMD[2:0] bits and the offset is set to 0 with the GTSWOS register, the count switch functions the same as the counter clear and the GTCNT counter does not operate as the trough. Also, if the switch mode is set to Switch to downcount with offset reload and the offset is set to the GTPR value, the GTCNT counter does not operate as the crest. On the other hand, if the switch mode is set to Switch to upcount with offset reload and the offset is set to the GTPR value, the GTCNT counter operates as the crest, if the switch mode is set to Switch to downcount with offset reload and the offset is set to 0, the GTCNT counter operates as the trough, and in these cases, the interrupt outputs and the buffer transfers of the crest and the trough functions normally.

If the crest or the trough due to the count switch occurs at the timing when the GTCNT counter value is the crest or the trough, the buffer transfer and the interrupt output at the crest or trough will occur twice in a row.

Table 19.10 Triangle-waves condition after the count switch with offset reload

Count direction of switch mode (GTCR.SWMD[2:0])	Offset (GTSWOS)	The triangle-waves condition after the count switch			
		GTCNT	GTST.TUCF	Triangle-waves condition	Buffer transfer/ Interrupt output
Upcount	0	GTSWOS register value	1	Counter clear (start of upcount)	Nothing
	1 to GTPR - 1			Upcount	Nothing
	GTPR			Crest (end of upcount)	Buffer transfer available (only Triangle-Wave PWM Mode 2)/ GTPn_OVF Output
Downcount	0		0	Trough (end of downcount)	Buffer transfer available/ GTPn_UDF Output
	1 to GTPR - 1			Downcount	Nothing
	GTPR			Start of downcount	Nothing

Figure 19.41 and Figure 19.42 show examples of the buffer operation due to the count switch when the offset is set to 0 and GTPR. Figure 19.43 shows example of the buffer operation due to the count switch when the offset is set other than the crest and trough.

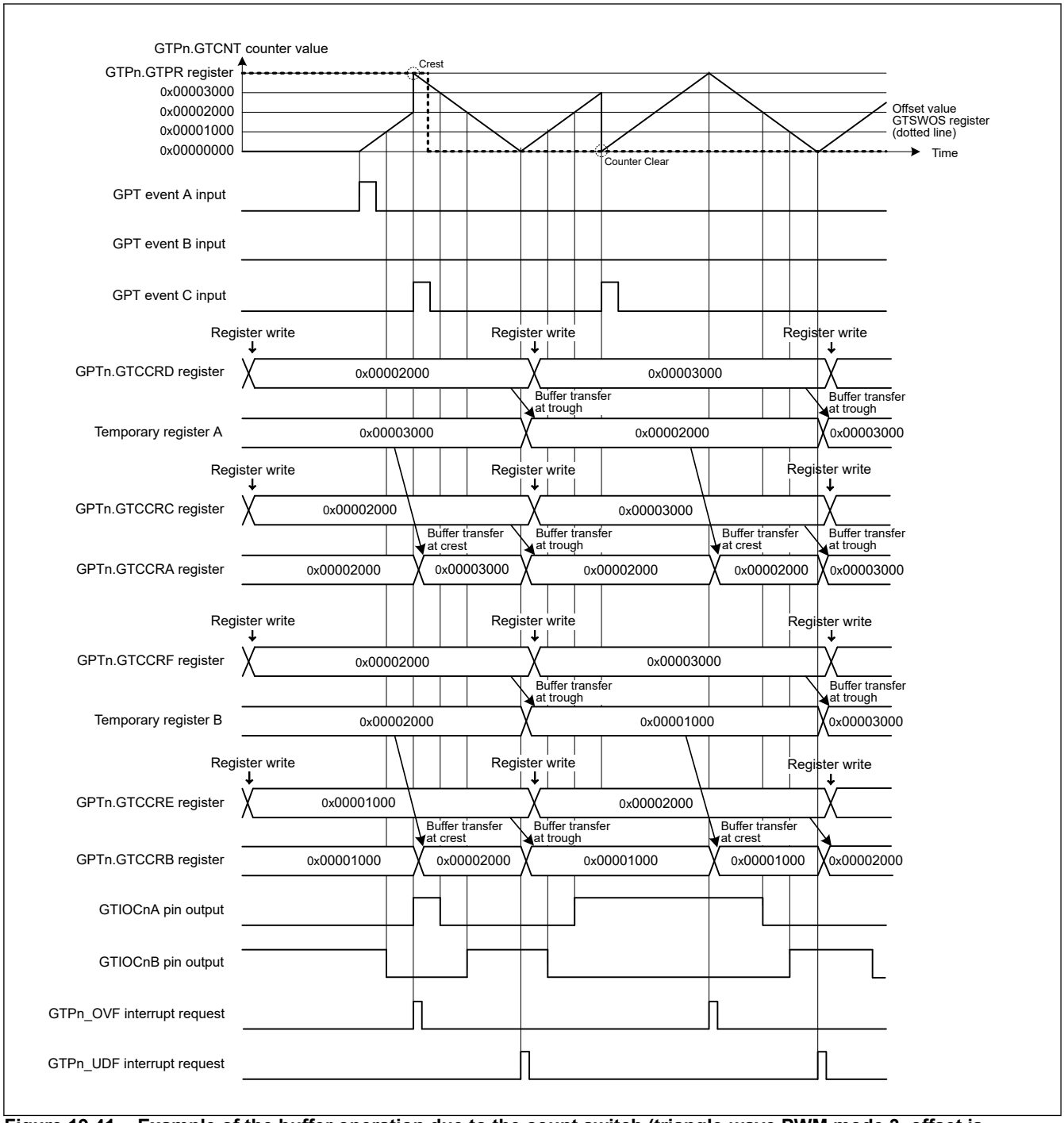


Figure 19.41 Example of the buffer operation due to the count switch (triangle-wave PWM mode 3, offset is set to 0 and GTPR, and switch to upcount)

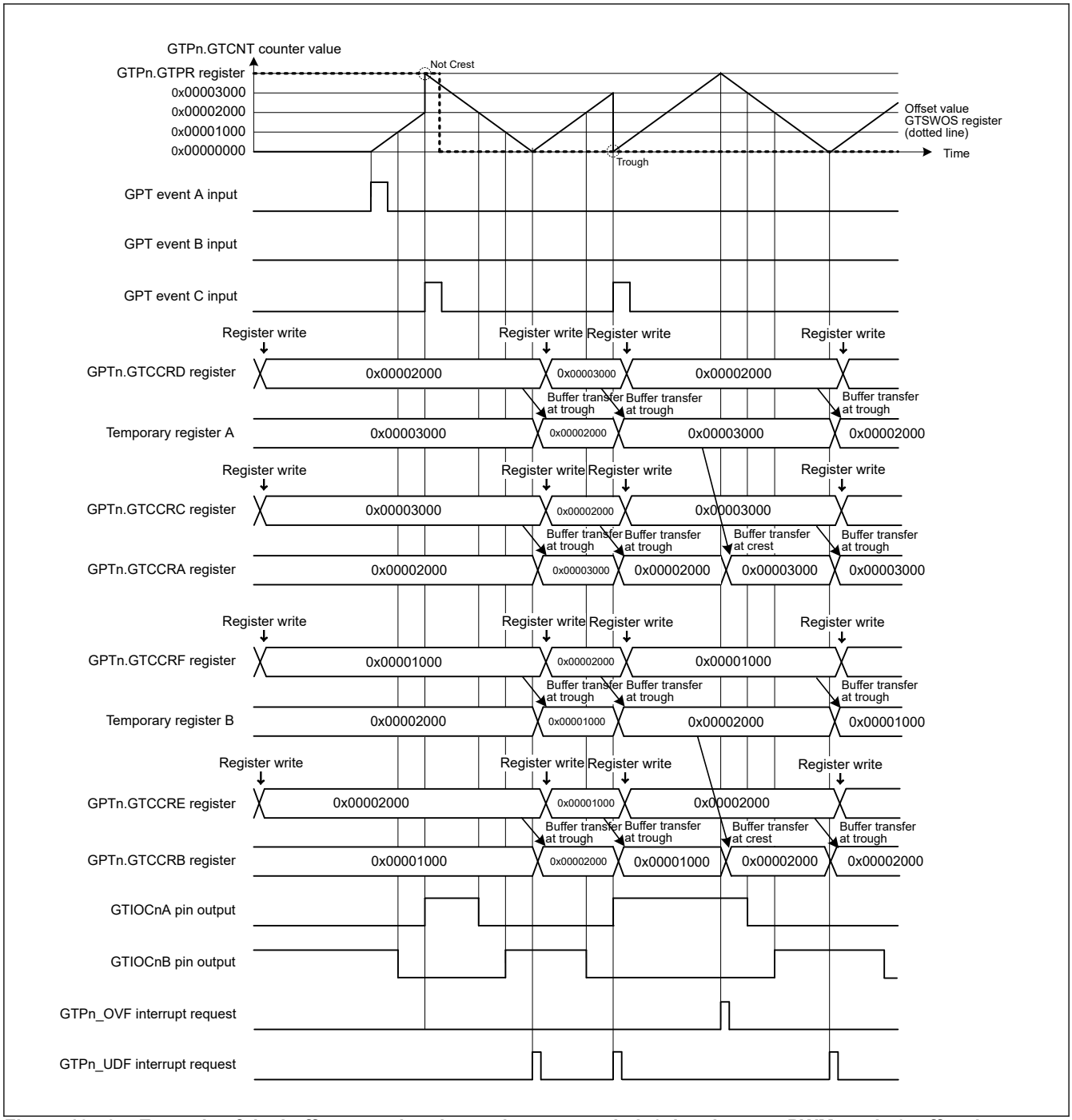


Figure 19.42 Example of the buffer operation due to the count switch (triangle-wave PWM mode 3, offset is set to 0 and GTPR, and switch to downcount)

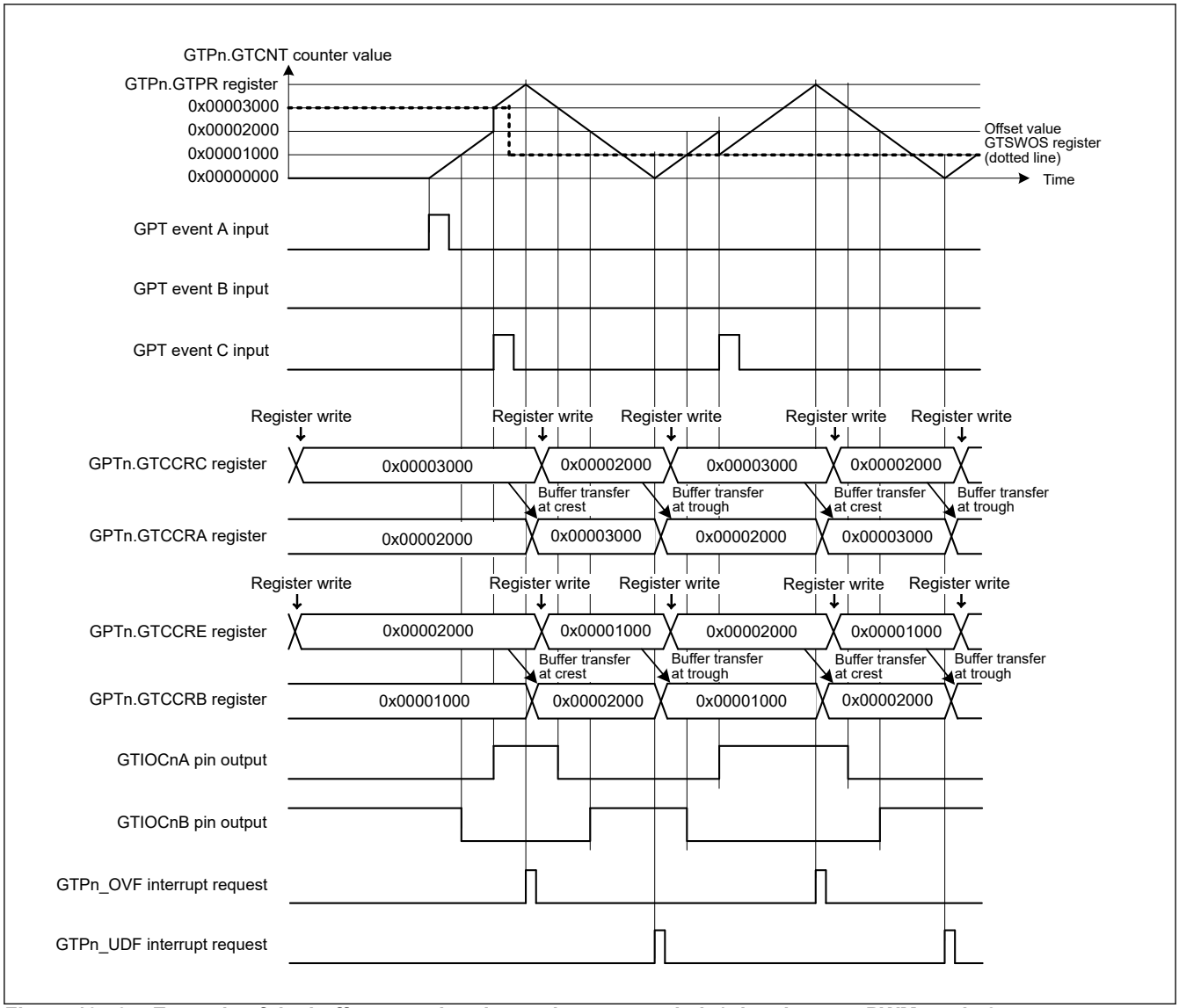


Figure 19.43 Example of the buffer operation due to the count switch (triangle-wave PWM mode 2, no crest and trough occurrence due to the count switch)

When the counter value skips the compare match value by the count switch, no compare match occurs and the GTIOCnm pin output does not change. Figure 19.44 shows example of the operation when the compare match is skipped by switch operation.

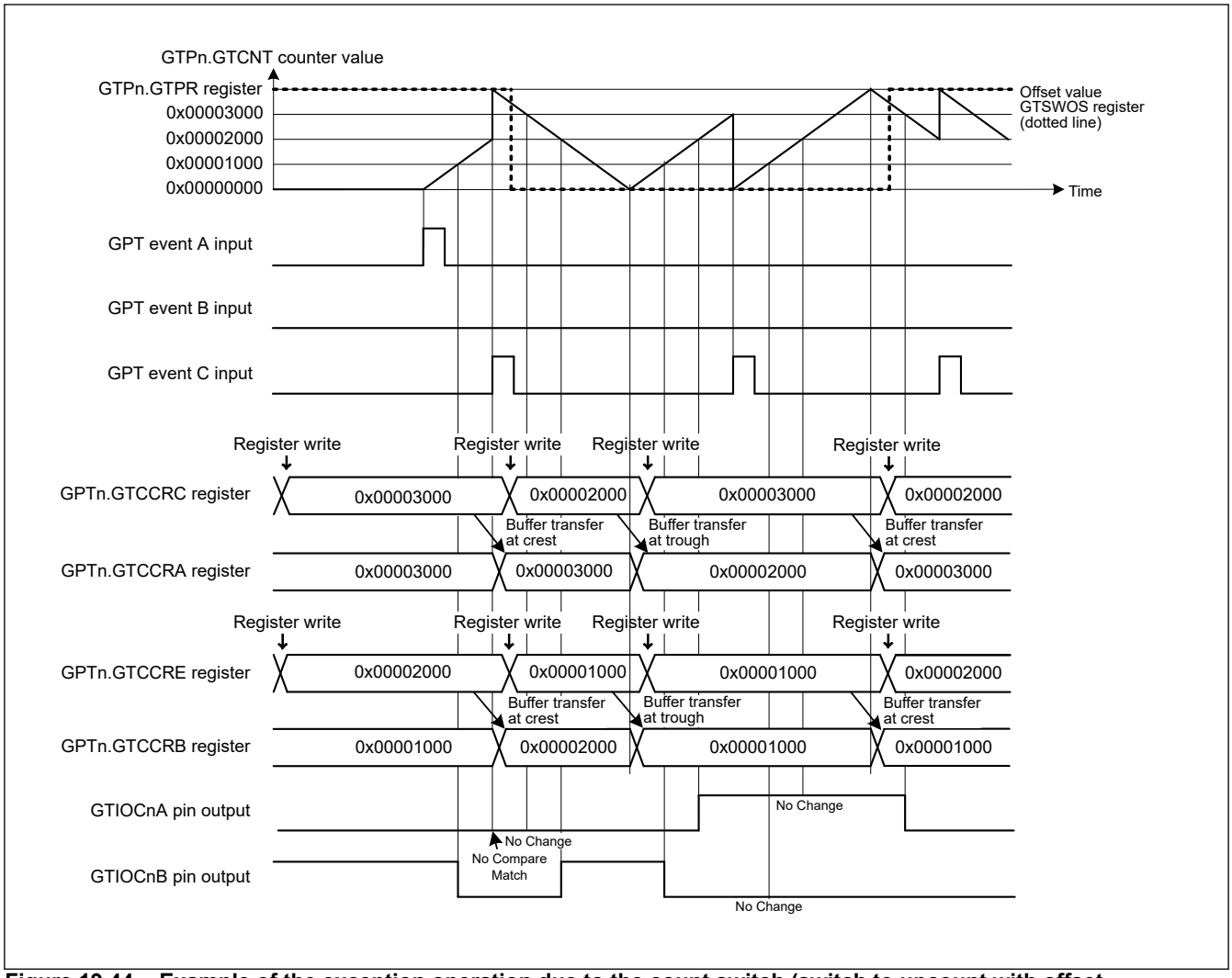


Figure 19.44 Example of the exception operation due to the count switch (switch to upcount with offset reload by GPT event C)

Figure 19.45 and Figure 19.46 show examples of the timing of operations to switch the counter in response to a rising edge of the input on the GTETRGA pin when a clock signal produced by frequency-dividing the Core clock signal is used as the counter clock for the GTCNT counter.

The GTCNT counter is switched when counting is in progress after the GPT has detected the internal switching signal.

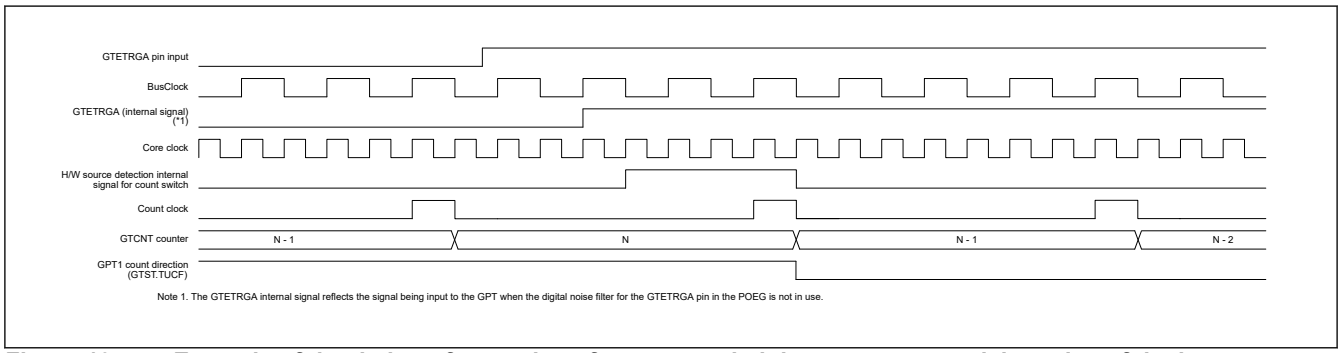


Figure 19.45 Example of the timing of operations for count switch in response to a rising edge of the input on the GTETRGA pin (switch to downcount without offset reload during the counting)

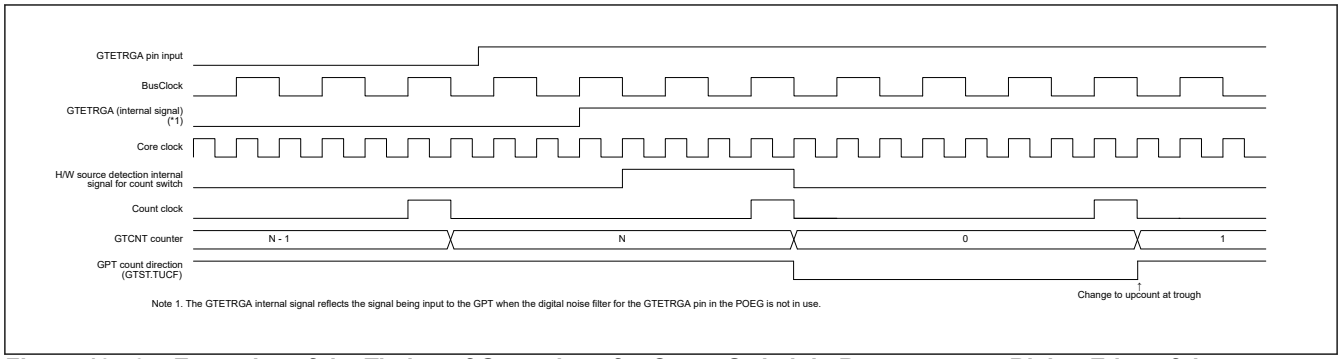


Figure 19.46 Examples of the Timing of Operations for Count Switch in Response to a Rising Edge of the Input on the GTETRGA Pin (switch to downcount with offset reload during the counting and offset value is 0)

Figure 19.47 and Figure 19.48 show examples of the timing of count switch operations in response to the input of an event signal from the GPT event A when counting is triggered by a hardware source.

This is an example of operations to switch the GPT1.GTCNT counter in response to a signal. An event signal is output to the ELC after matches in comparison with the GPT0.GTCCRA register. This is selected as a trigger for output to the GPT by the ELC as event source A.

The GPT0 compare match A signal is synchronized with Core clock. The ELC receives the signal in synchronization with BusClock, and outputs the GPT event source A signal after 1 cycle of BusClock has elapsed. The GTCNT counter is switched in synchronization with Core clock after the GPT has detected the internal switching signal.

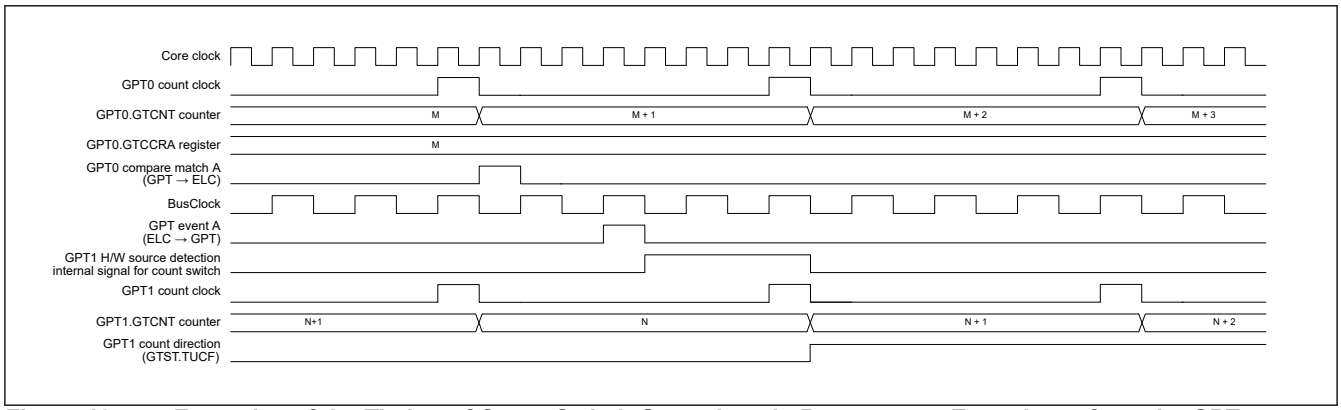


Figure 19.47 Examples of the Timing of Count Switch Operations in Response to Event Input from the GPT event A (switch to upcount without offset reload during the counting)

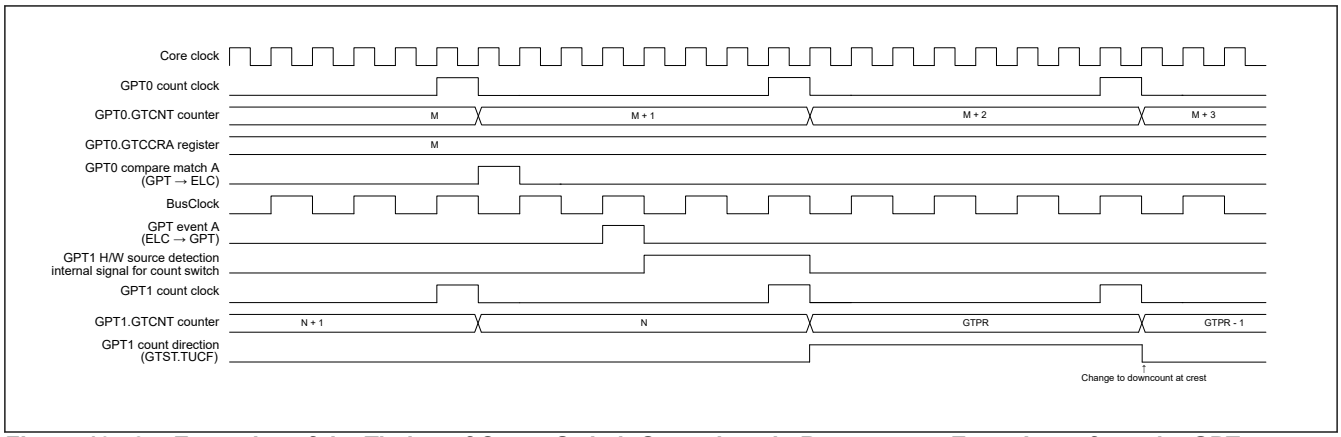


Figure 19.48 Examples of the Timing of Count Switch Operations in Response to Event Input from the GPT event A (switch to upcount with offset reload during the counting and offset value is GTPR)

19.3.9 Synchronous Operation

Synchronous operation on channels such as synchronous start/stop/clear/switch operation can be performed.

19.3.9.1 Synchronous Operation by Software

The count operations for respective channels can be started, stopped and cleared simultaneously by setting multiple bits of the GTSTR, GTSTP, and GTCLR registers to 1 at the same time.

Count start with phase differences among channels is possible by setting the GTCNT counter value before counting starts and then by setting simultaneously multiple bits in the GTSTR register to 1.

The Figure 19.49 shows an example of simultaneous start/stop/clear of four channels by software, and Figure 19.50 shows an example of phase shift start among four channels by software.

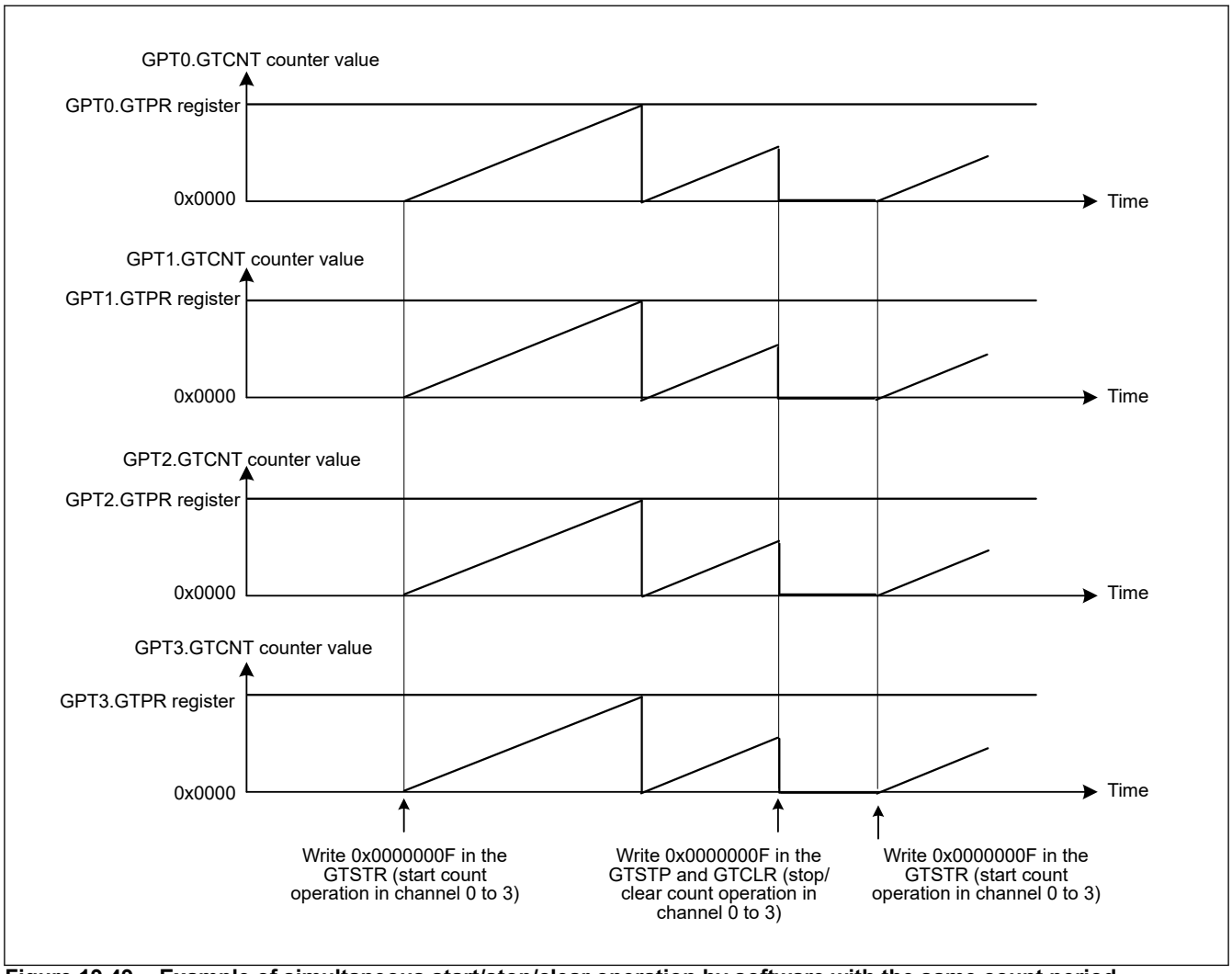


Figure 19.49 Example of simultaneous start/stop/clear operation by software with the same count period (GTPR register value)

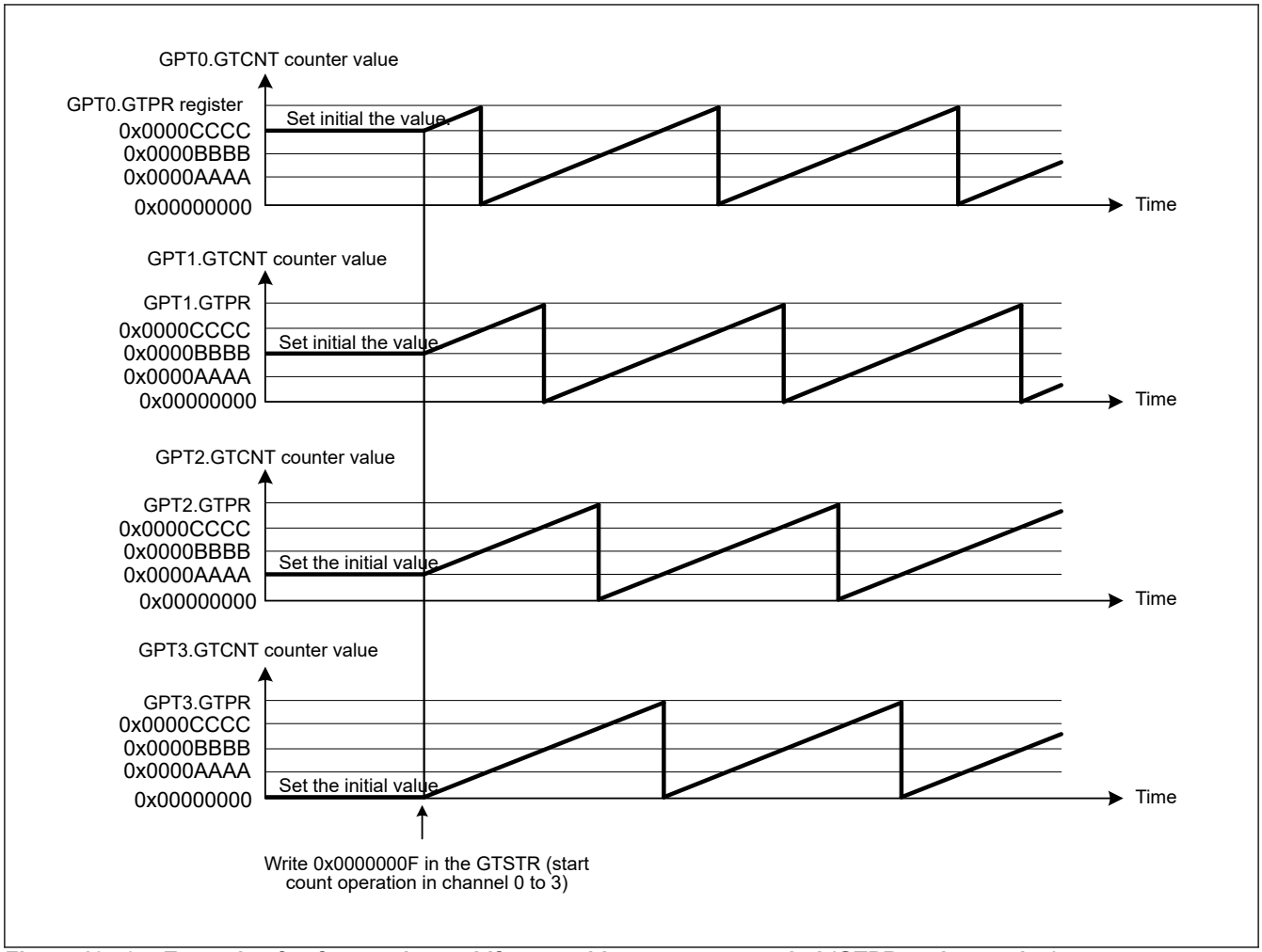


Figure 19.50 Example of software phase shift start with same count period (GTPR register value)

19.3.9.2 Synchronous Operation by Hardware Source

The count operations for respective channels can be started/stopped/cleared/switched simultaneously by hardware sources such as external trigger input and ELC event input.

Synchronous operation by the GTIOCnA and GTIOCnB pin inputs (n = 0 to 17) can be performed by setting the ELC event with input capture as a hardware source.

Figure 19.51 shows an example of simultaneous start/stop/clear/switch operation of four channels by hardware source.

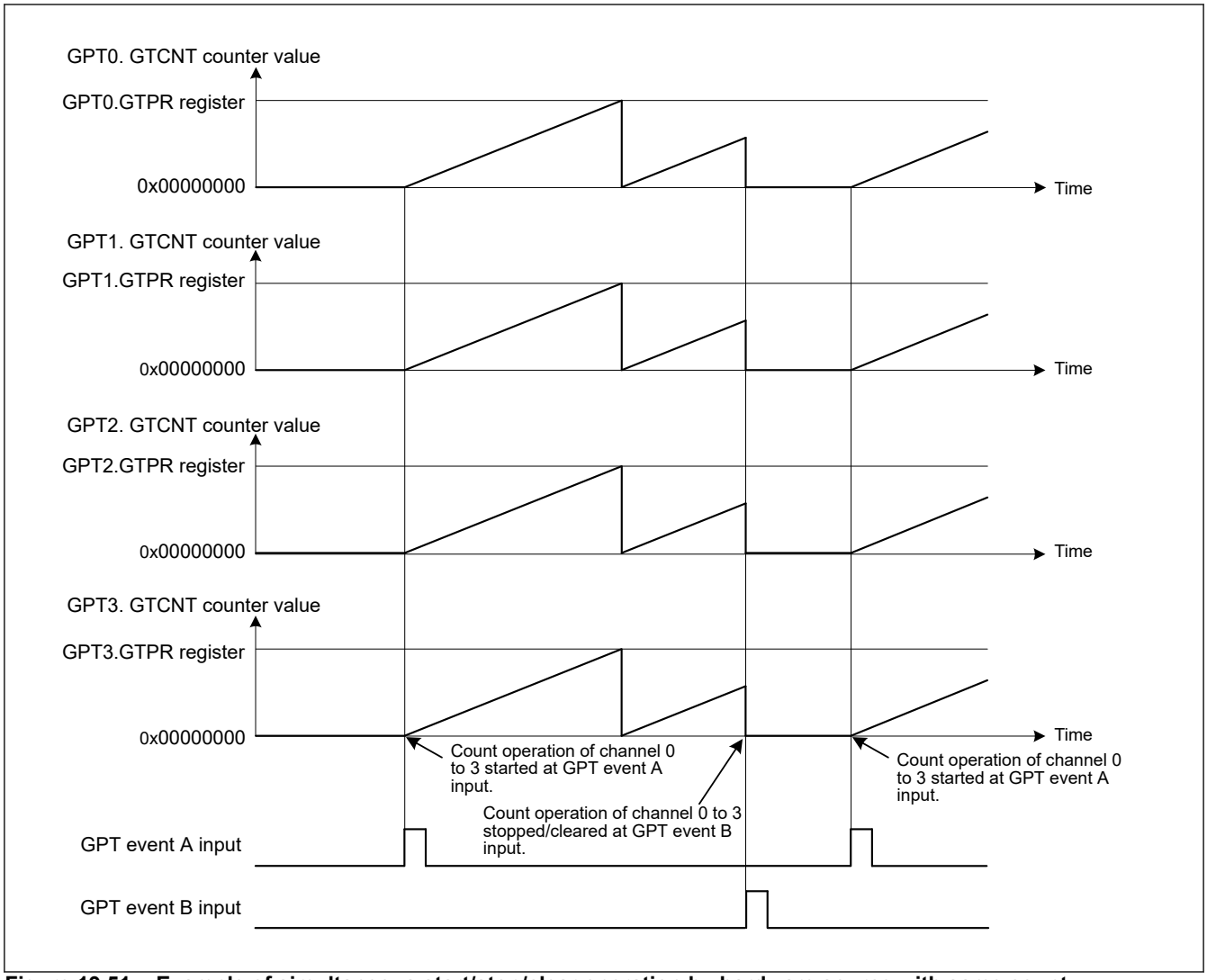


Figure 19.51 Example of simultaneous start/stop/clear operation by hardware source with same count period (GTPR register value)

The following example shows setting for simultaneous start by hardware source.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits of the corresponding channel.
   In Figure 19.51, 000b (sawtooth-wave PWM mode) is set.
2. Set count direction
   Select the count direction (up or down) with the GTUDDTYC register.
   In Figure 19.51, the lower 2 bits of the GTUDDTYC register is set to 11b and then to 01b for up-counting.
3. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits of the corresponding channel.
4. Period setting
   Set a period in the GTPR register of the corresponding channel.
5. Set initial value for counter
   Set the initial value in the GTCNT counter of the corresponding channel.
   In Figure 19.51, 0x00000000 is set.
6. Set hardware count start
   Select a hardware source to start the count operation with the GTSSR register and wait for count start by the hardware source.
   In Figure 19.51, GTSSR.SSELCA bit = 1.
7. Set hardware count stop

Select a hardware source to stop count operation with the GTPSR register and wait for count stop by the hardware source.
In Figure 19.51, GTPSR.PSELCB bit = 1.

8. Set hardware count clearing
Select a hardware source to clear the counter with the GTCSR register and wait for counter clear by the hardware source.
In Figure 19.51, GTCSR.CSELCB bit = 1.

9. Start hardware source operation
Set operation of the hardware source selected by the GTSSR, GTPSR, and GTCSR registers and start counting, stop counting, or clearing the counter.
In Figure 19.51, the GPT event A and B inputs are set.

19.3.10 PWM Output Operation Examples

(1) Synchronous PWM Output

The GPT can output two phases of linked PWM waveforms for one channel or 14 phases of linked PWM waveforms for a maximum of seven channels by one GPT or 36 phases of linked PWM waveforms for a maximum of 18 channels by plural GPT by synchronizing operation of the channels.

Figure 19.52 shows an example in which four channels perform synchronous operation in sawtooth-wave PWM mode and eight phases of PWM waveforms are output. The GTIOCnA pin (n = 0 to 3) is set so that it outputs low as the initial output, high at a GTCCRA register compare match, and low at the end of the cycle. The GTIOCnB pin is set so that it outputs low as the initial output, high at a GTCCRB register compare match, and low at the end of the cycle.

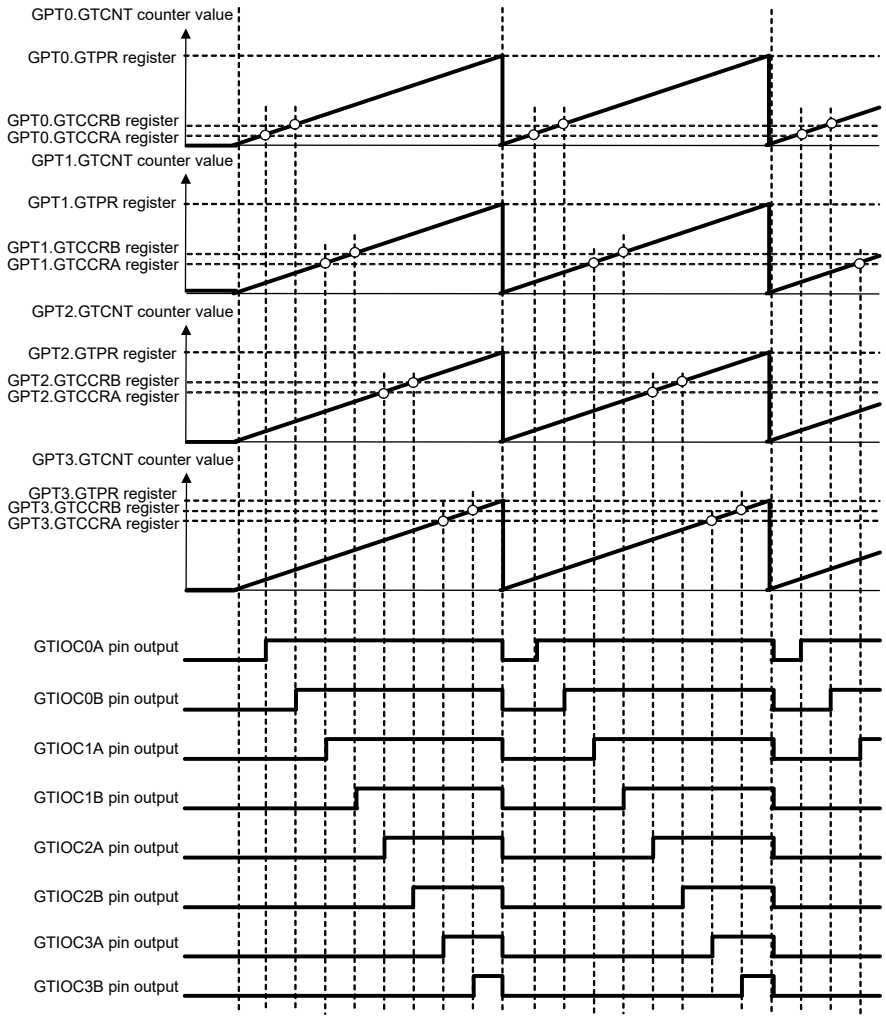


Figure 19.52 Example of synchronous PWM output

(2) 3-Phase Sawtooth-Wave Complementary PWM Output

Figure 19.53 shows an example in which three channels perform synchronous operation in sawtooth-wave PWM mode and 3-phase complementary PWM waveforms are output. The GTIOCnA pin (n = 0 to 2) is set so that it outputs low as the initial output, high at a GTCCRA register compare match, and low at the end of the cycle. The GTIOCnB pin is set so that it outputs high as the initial output, low at a GTCCRB register compare match, and high at the end of the cycle.

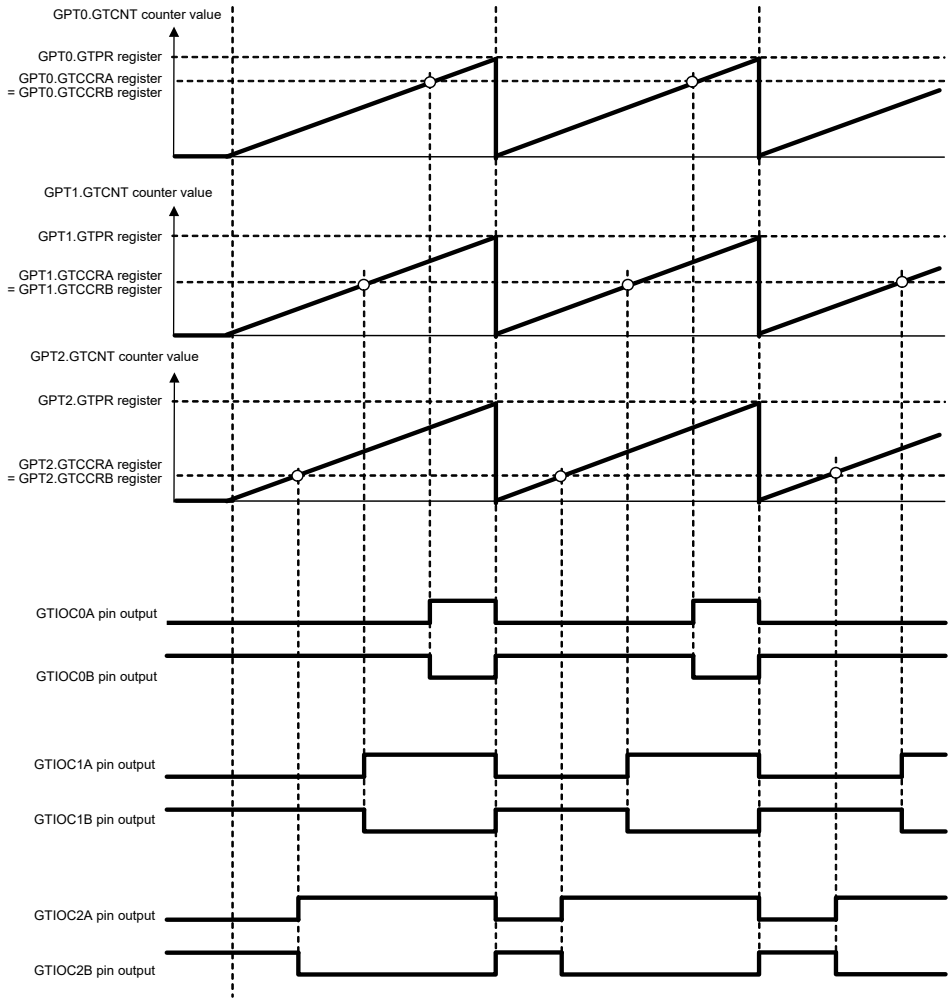


Figure 19.53 Example of 3-phase sawtooth-wave complementary PWM output

(3) 3-phase sawtooth-wave complementary PWM output with automatic dead time setting

Figure 19.54 shows an example in which three channels perform synchronous operation in sawtooth-wave one-shot pulse mode with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA pin (n = 0 to 2) is set so that it outputs low as the initial output, toggles the output at a GTCCRA register compare match, and retains the output at the end of the cycle. The GTIOCnB pin is set so that it outputs high as the initial output, toggles the output at a GTCCRB register compare match, and retains the output at the end of the cycle.

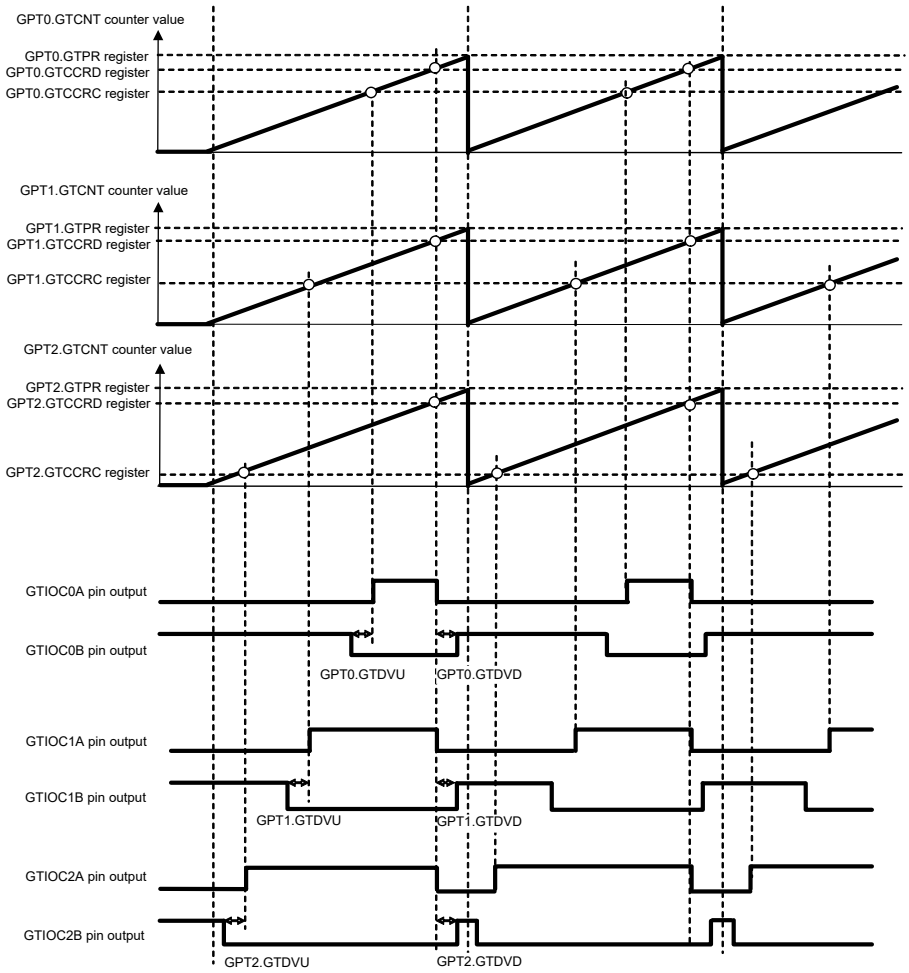


Figure 19.54 Example of 3-phase sawtooth-wave complementary PWM output with automatic dead time setting

(4) 3-Phase Triangle-Wave Complementary PWM Output

Figure 19.55 shows an example in which three channels perform synchronous operation in triangle-wave PWM mode 1 and 3-phase complementary PWM waveforms are output. The GTIOCnA pin (n = 0 to 2) is set so that it outputs low as the initial output, toggles the output at a GTCCRA register compare match, and retains the output at the end of the cycle. The GTIOCnB pin is set so that it outputs high as the initial output, toggles the output at a GTCCRB register compare match, and retains the output at the end of the cycle.

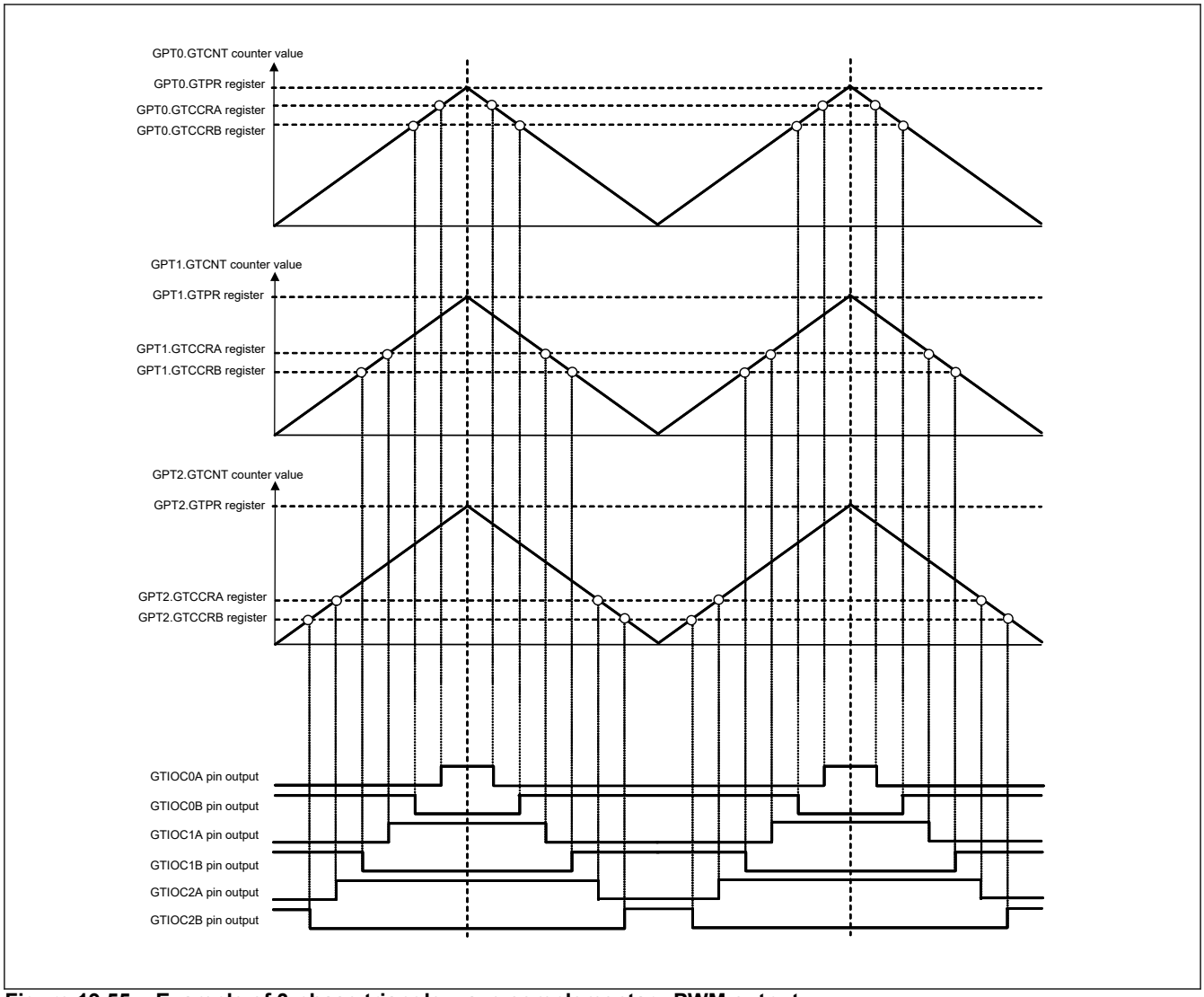


Figure 19.55 Example of 3-phase triangle-wave complementary PWM output

(5) 3-Phase Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

Figure 19.56 shows an example in which three channels perform synchronous operation in triangle-wave PWM mode 1 with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA pin (n = 0 to 2) is set so that it outputs low as the initial output, toggles the output at a GTCCRA register compare match, and retains the output at the end of the cycle. The GTIOCnB pin is set so that it outputs high as the initial output, toggles the output at a GTCCRB register compare match, and retains the output at the end of the cycle.

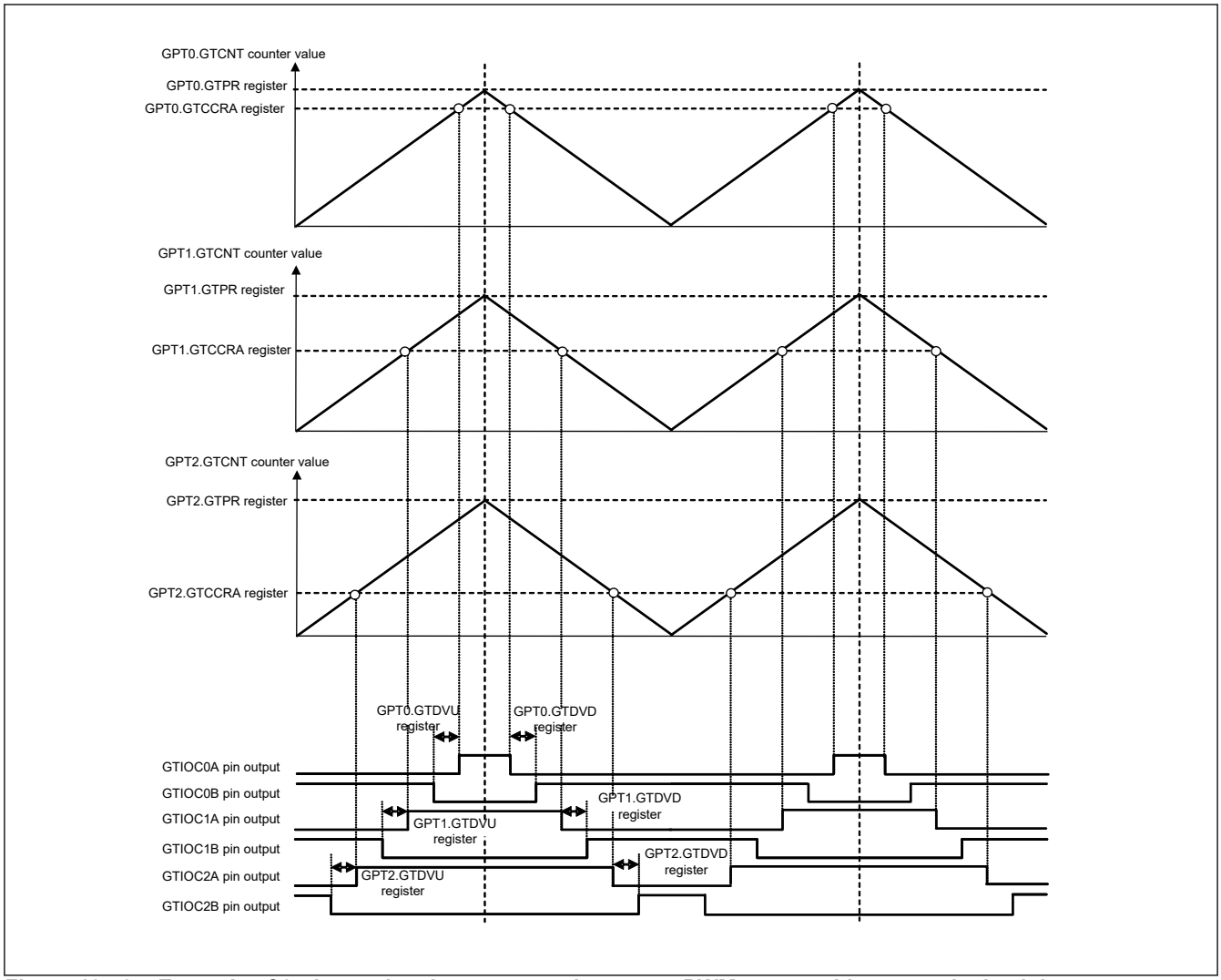


Figure 19.56 Example of 3-phase triangle-wave complementary PWM output with automatic dead time setting

(6) 3-Phase Asymmetric Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

Figure 19.57 shows an example in which three channels perform synchronous operation in triangle-wave PWM mode 3 with automatic dead time setting and 3-phase complementary PWM waveforms are output. The GTIOCnA pin (n = 0 to 2) is set so that it outputs low as the initial output, toggles the output at a GTCCRA register compare match, and retains the output at the end of the cycle. The GTIOCnB pin is set so that it outputs high as the initial output, toggles the output at a GTCCRB register compare match, and retains the output at the end of the cycle.

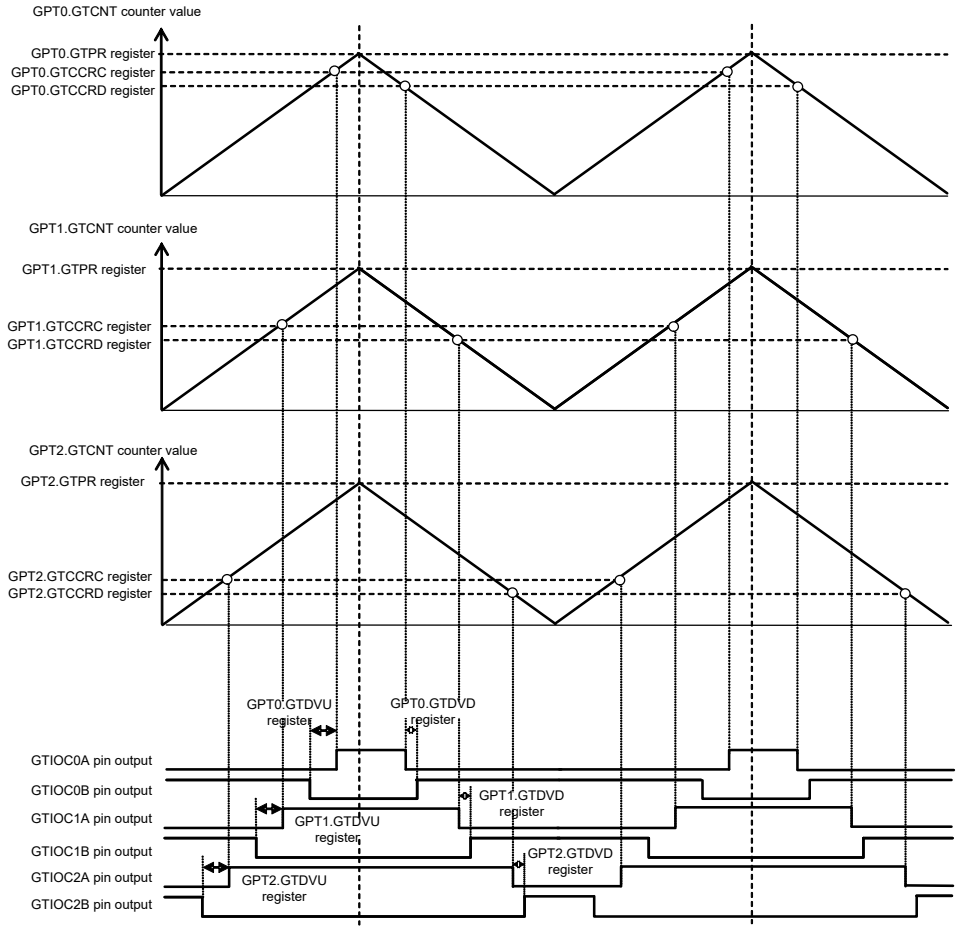


Figure 19.57 Example of 3-Phase Asymmetric Triangle-Wave Complementary PWM Output with Automatic Dead Time Setting

19.3.11 Phase Counting Mode

The GTCNT counter can be up-counting or down-counting by detecting the phase difference between the GTIOCnA pin input (n = 0 to 17) and the GTIOCnB pin input. Detection of the phase difference can be performed by setting the relationship between the edge and the level of the GTIOCnA pin input and the GTIOCnB pin input using the GTUPSR and GTDNSR registers for input combination. See section 19.3.1.1. Counter Operation.

Figure 19.58 to Figure 19.67 show the examples of setting procedure for the phase counting mode 1 to 5, and Table 19.11 to Table 19.20 show the setting of the GTUPSR and GTDNSR registers including up-counting and down-counting conditions.

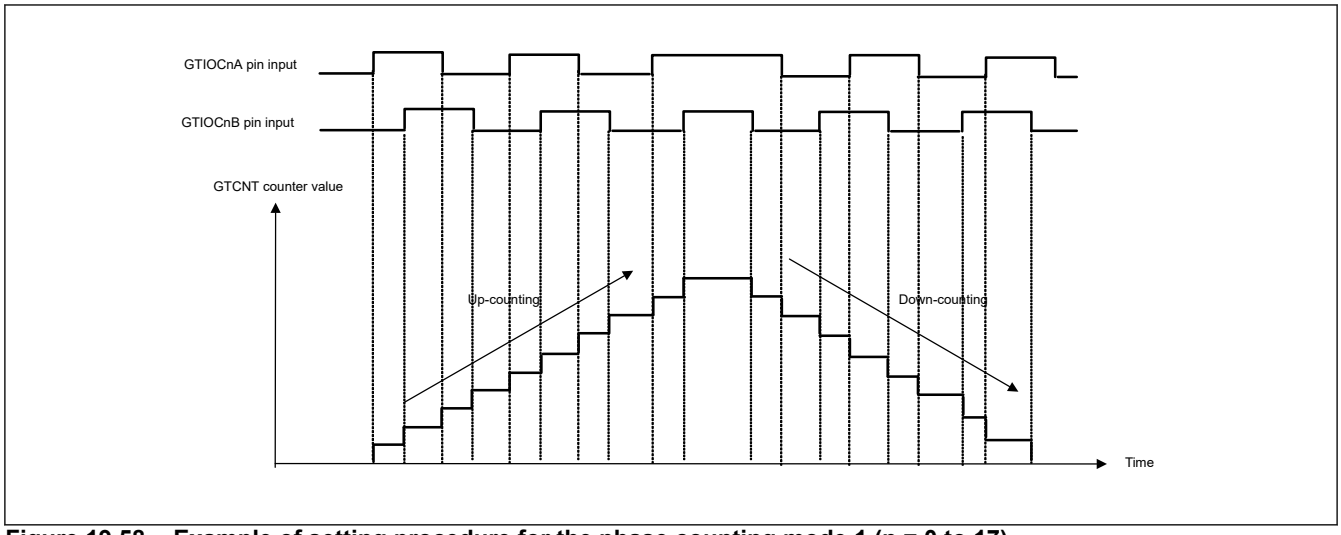


Figure 19.58 Example of setting procedure for the phase counting mode 1 (n = 0 to 17)

Table 19.11 Up-counting and down-counting conditions for phase counting mode 1

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Up-counting	GTUPSR = 0x00006900 GTDNSR = 0x00009600
Low	↓		
↑	Low		
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High		
↓	Low		

Note: n = 0 to 17
↑ : Rising edge
↓ : Falling edge

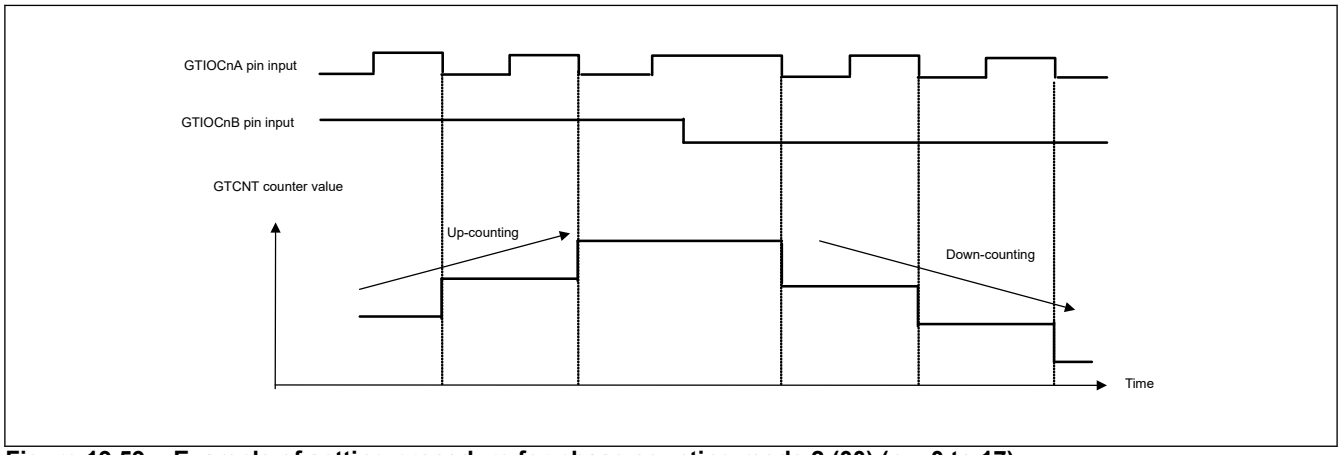


Figure 19.59 Example of setting procedure for phase counting mode 2 (00) (n = 0 to 17)

Table 19.12 Up-counting and down-counting conditions for phase counting mode 2 (00)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0x00000800 GTDNSR = 0x00000400
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High		
↓	Low	Down-counting	

Note: n = 0 to 17
↑ : Rising edge
↓ : Falling edge

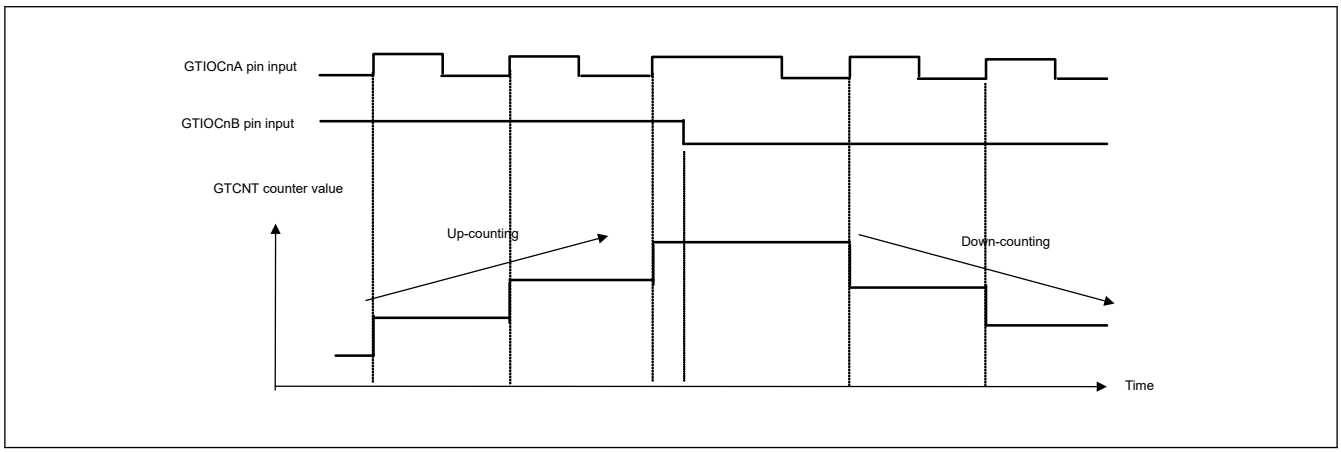


Figure 19.60 Example of setting procedure for the phase counting mode 2 (01) (n = 0 to 17)

Table 19.13 Up-counting and down-counting conditions for phase counting mode 2 (01)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0x00000200 GTDNSR = 0x00000100
Low	↓		
↑	Low	Down-counting	
↓	High	Don't care	
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Don't care	

Note: n = 0 to 17
↑ : Rising edge
↓ : Falling edge

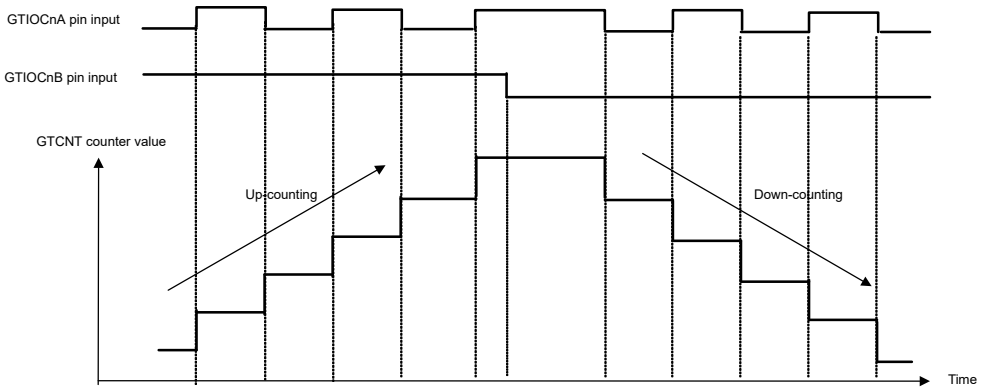


Figure 19.61 Example of setting procedure for the phase counting mode 2 (1x) (n = 0 to 17)

Table 19.14 Up-counting and down-counting conditions for phase counting mode 2 (1x)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0x00000A00 GTDNSR = 0x00000500
Low	↓		
↑	Low	Down-counting	
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High	Up-counting	
↓	Low	Down-counting	

Note: n = 0 to 17
↑ : Rising edge
↓ : Falling edge

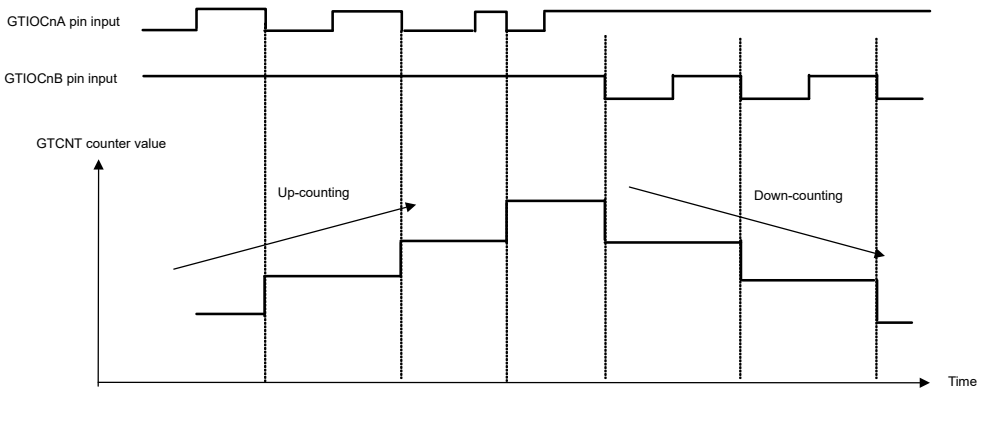


Figure 19.62 Example of setting procedure for the phase counting mode 3 (00) (n = 0 to 17)

Table 19.15 Up-counting and down-counting conditions for the phase counting mode 3 (00)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0x00000800 GTDNSR = 0x00008000
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Down-counting	
Low	↑	Don't care	
↑	High		
↓	Low		

Note: n = 0 to 17
↑ : Rising edge
↓ : Falling edge

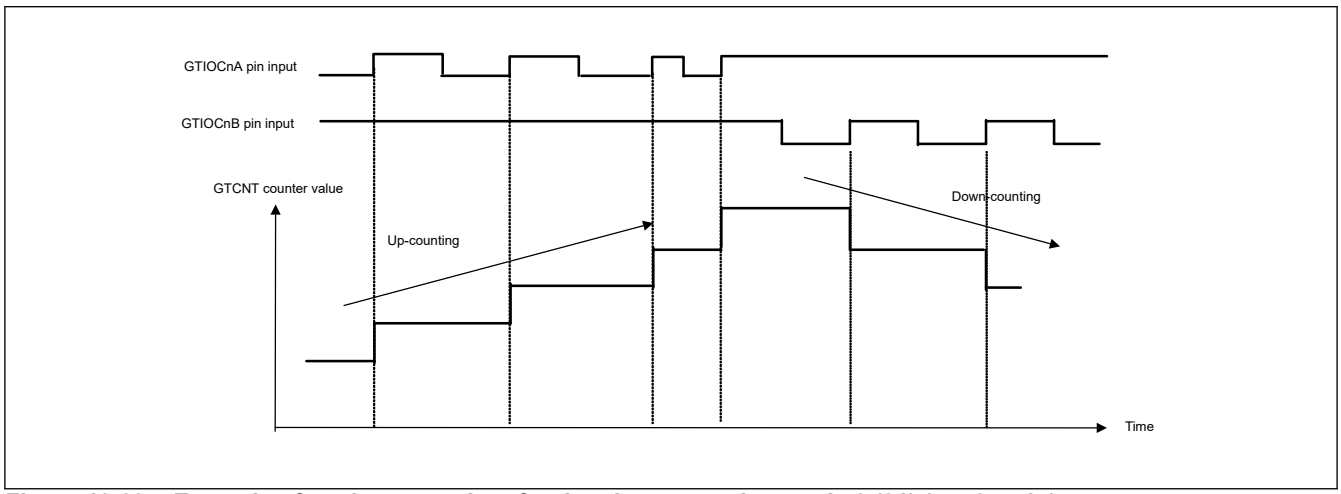


Figure 19.63 Example of setting procedure for the phase counting mode 3 (01) (n = 0 to 17)

Table 19.16 Up-counting and down-counting conditions for the phase counting mode 3 (01)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Down-counting	GTUPSR = 0x00000200 GTDNSR = 0x00002000
Low	↓	Don't care	
↑	Low		
↓	High		
High	↓		
Low	↑		
↑	High	Up-counting	
↓	Low	Don't care	

Note: n = 0 to 17
↑ : Rising edge

: Falling edge

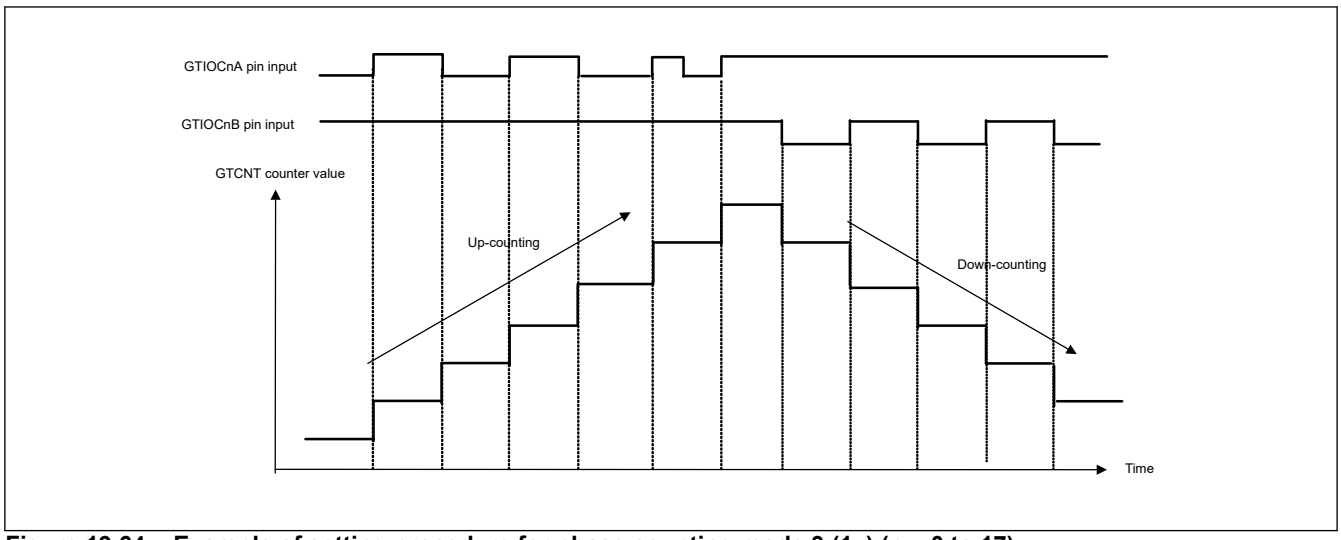


Figure 19.64 Example of setting procedure for phase counting mode 3 (1x) (n = 0 to 17)

Table 19.17 Up-counting and down-counting conditions for phase counting mode 3 (1x)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Down-counting	GTUPSR = 0x00000A00 GTDNSR = 0x0000A000
Low	↓	Don't care	
↑	Low		
↓	High	Up-counting	
High	↓	Down-counting	
Low	↑	Don't care	
↑	High	Up-counting	
↓	Low	Don't care	

Note: n = 0 to 17

: Rising edge

: Falling edge

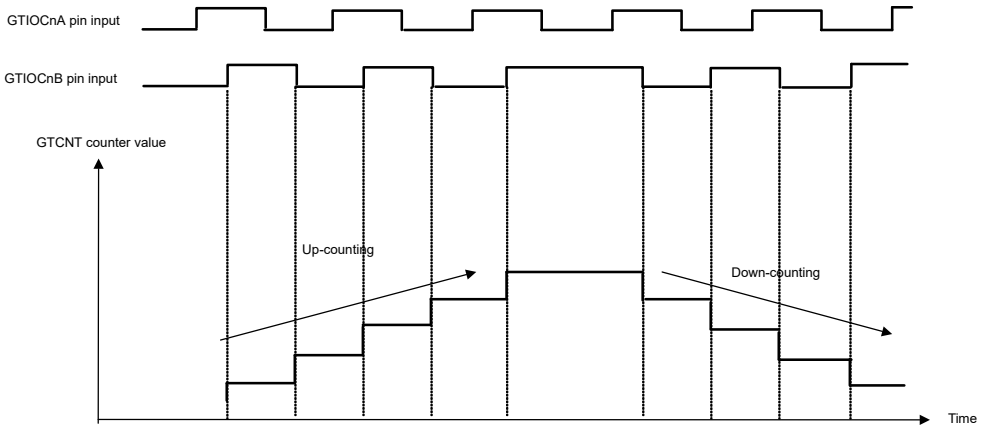


Figure 19.65 Example of setting procedure for phase counting mode 4 (n = 0 to 17)

Table 19.18 Up-counting and down-counting conditions for phase counting mode 4

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Up-counting	GTUPSR = 0x00006000 GTDNSR = 0x00009000
Low	↓		
↑	Low	Don't care	
↓	High		
High	↓	Down-counting	
Low	↑		
↑	High	Don't care	
↓	Low		

Note: n = 0 to 17
↑ : Rising edge
↓ : Falling edge

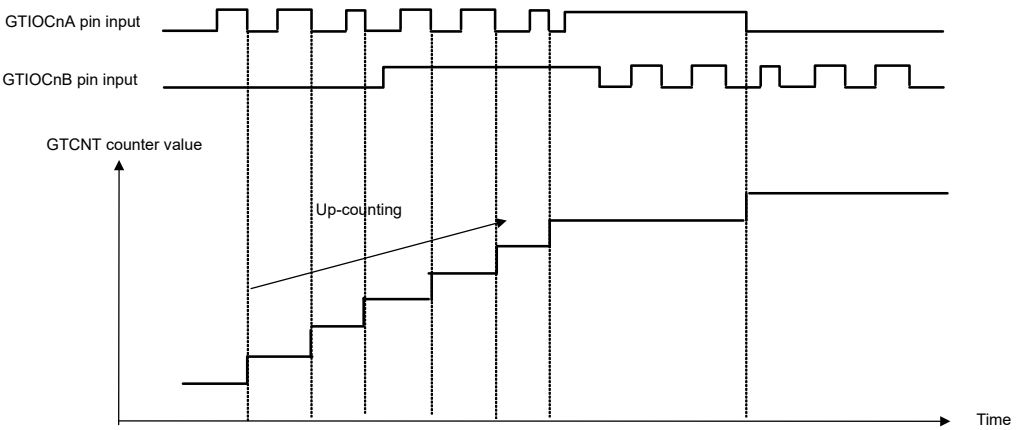


Figure 19.66 Example of setting procedure for the phase counting mode 5 (0x) (n = 0 to 17)

Table 19.19 Up-counting and down-counting conditions for phase counting mode 5 (0x)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0x00000C00 GTDNSR = 0x00000000
Low	↓		
↑	Low		
↓	High	Up-counting	
High	↓	Don't care	
Low	↑		
↑	High		
↓	Low	Up-counting	

Note: n = 0 to 17
↑ : Rising edge
↓ : Falling edge

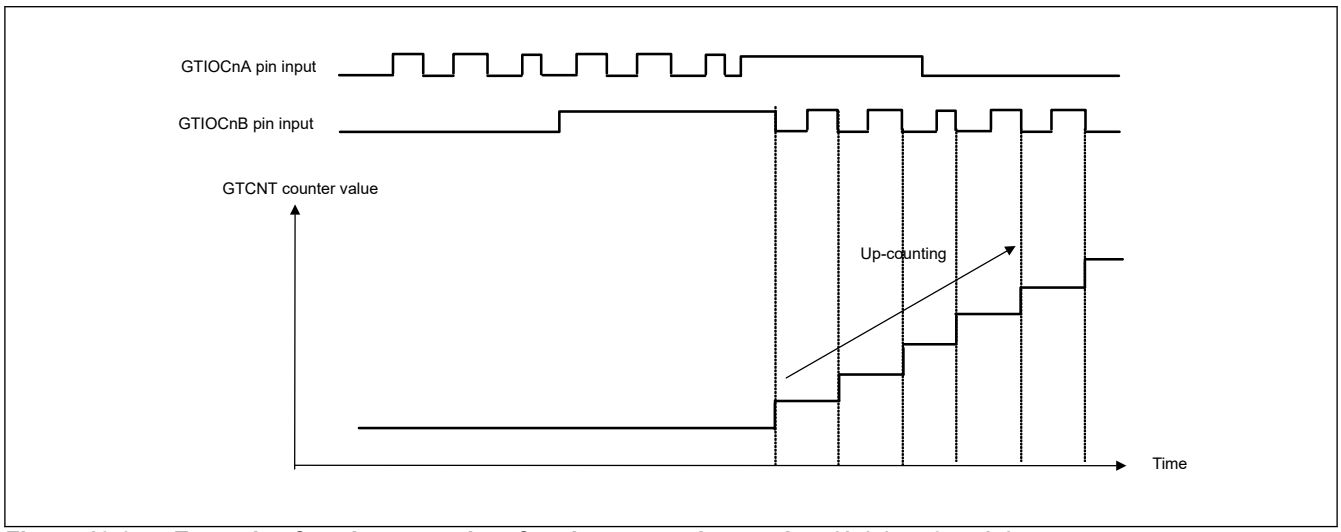
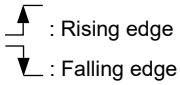


Figure 19.67 Example of setting procedure for phase counting mode 5 (1x) (n = 0 to 17)

Table 19.20 Up-counting and down-counting conditions for phase counting mode 5 (1x)

GTIOCnA pin input	GTIOCnB pin input	Operation	Setting of register
High	↑	Don't care	GTUPSR = 0x0000C000 GTDNSR = 0x00000000
Low	↓	Up-counting	
↑	Low	Don't care	
↓	High		
High	↓	Up-counting	
Low	↑	Don't care	
↑	High		
↓	Low		

Note: n = 0 to 17



19.4 Interrupt Sources

19.4.1 Interrupt Sources

There are four kinds of interrupt sources:

- GTCCRm register (m = A to F) input capture / compare match
- GTADTR register compare match (only used for ELC)
- GTCNT counter overflow (GTPR register compare match) / underflow
- Dead time error

Each interrupt source has its own status flag and control bit for generating interrupt request signals, where the generation of interrupt request signals can be enabled or disabled individually.

When an interrupt source condition is satisfied, the corresponding status flag in GTST is set to 1 and an interrupt request is generated if the corresponding interrupt request enable or disable bit in the GTINTAD register is 1.

The corresponding status flag in the GTST register is writable to clear. If flag set and flag clear occur at the same time, flag clear has priority over flag set. Both dead time error (DTEF flag) and those flags are automatically updated by internal state.

For details, see section 11, Interrupt Controller (ICU). Table 19.21 shows a list of the GPT interrupt sources.

Table 19.21 Interrupt sources

Channel	Name	Interrupt source	Interrupt flag
GPTn	GPTn_CCMPA	GPTn.GTCCRA register input capture / compare match	—
	GPTn_CCMPB	GPTn.GTCCRB register input capture / compare match	—
	GPTn_CMPC	GPTn.GTCCRC register compare match	—
	GPTn_CMPD	GPTn.GTCCRD register compare match	—
	GPTn_CMPE	GPTn.GTCCRE register compare match	—
	GPTn_CMPF	GPTn.GTCCRF register compare match	—
	GPTm_DTE	GPTm dead time error	GPTm.GTST.DTEF
	GPTm_ADTRGA	GPTm.GTADTRA register compare match	GPTm.GTST.ADTRAUF GPTm.GTST.ADTRADF
	GPTm_ADTRGB	GPTm.GTADTRB register compare match	GPTm.GTST.ADTRBUF GPTm.GTST.ADTRBDF
	GPTn_OVF	GPTn.GTCNT counter overflow (GPTn.GTPR register compare match)	—
	GPTn_UDF	GPTn.GTCNT counter underflow	—

Note: n = 0 to 17, m = 0 to 13

(1) GPTn_CCMPA Interrupt (n = 0 to 17)

When the GTINTAD.GTINTA bit is 1, an interrupt request is generated under the following conditions:

- When the GTCCRA register functions as a compare match register, the GTCNT counter value matches with the GTCCRA register
- When the GTCCRA register functions as an input capture register, the input capture signal causes transfer of the GTCNT counter value to the GTCCRA register

(2) GPTn_CCMPB Interrupt (n = 0 to 17)

When the GTINTAD.GTINTB bit is 1, an interrupt request is generated under the following conditions:

- When the GTCCRB register functions as a compare match register, the GTCNT counter value matches with the GTCCRB register

- When the GTCCRB register functions as an input capture register, the input capture signal causes transfer of the GTCNT counter value to the GTCCRB register

(3) GPTn_CMPC Interrupt (n = 0 to 17)

When the GTINTAD.GTINTC bit is 1, an interrupt request is generated under the following condition:

- When the GTCCRC register functions as a compare match register, the GTCNT counter value matches with the GTCCRC register

A compare match is not performed and thus interrupt is not requested under the following conditions:

- GTCR.MD[2:0] bits = 001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] bits = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] bits = 01b, 10b, 11b (buffer operation with the GTCCRC register)

(4) GPTn_CMPD Interrupt (n = 0 to 17)

When the GTINTAD.GTINTD bit is 1, an interrupt request is generated under the following condition:

- When the GTCCRD register functions as a compare match register, the GTCNT counter value matches with the GTCCRD register

A compare match is not performed and thus interrupt is not requested under the following conditions:

- GTCR.MD[2:0] bits = 001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] bits = 110b (triangle-wave PWM mode 3)
- GTBER.CCRA[1:0] bits = 10b, 11b (buffer operation with the GTCCRD register)

(5) GPTn_CMPE Interrupt (n = 0 to 17)

When the GTINTAD.GTINTE bit is 1, an interrupt request is generated under the following condition:

- When the GTCCRE register functions as a compare match register, the GTCNT counter value matches with the GTCCRE register.

A compare match is not performed and thus interrupt is not requested under the following conditions:

- GTCR.MD[2:0] bits = 001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] bits = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] bits = 01b, 10b, 11b (buffer operation with the GTCCRE register)

(6) GPTn_CMPF Interrupt (n = 0 to 17)

When the GTINTAD.GTINTF bit is 1, an interrupt request is generated under the following condition:

- When the GTCCRF register functions as a compare match register, the GTCNT counter value matches with the GTCCRF register.

A compare match is not performed and thus interrupt is not requested under the following conditions:

- GTCR.MD[2:0] bits = 001b (sawtooth-wave one-shot pulse mode)
- GTCR.MD[2:0] bits = 110b (triangle-wave PWM mode 3)
- GTBER.CCRB[1:0] bits = 10b, 11b (buffer operation with the GTCCRF register)

(7) GPTm_ADTRGA Interrupt (m = 0 to 13)

When the GTCNT counter value matches with the GTADTRA register, an interrupt request is generated under the following conditions:

- In up-counting, the interrupt enable bit (ADTRAUEN) in the GTINTAD register is 1
- In down-counting, the interrupt enable bit (ADTRADEN) in the GTINTAD register is 1

In event count operation, this interrupt request is not generated.

(8) GPTm_ADTRGB Interrupt (m = 0 to 13)

When the GTCNT counter value matches with the GTADTRB register, an interrupt request is generated under the following conditions:

- In up-counting, the interrupt enable bit (ADTRBUEN) in the GTINTAD register is 1
- In down-counting, the interrupt enable bit (ADTRBDEN) in the GTINTAD register is 1

In event count operation, this interrupt request is not generated.

(9) GPTn_OVF Interrupt (n = 0 to 17)

When the GTINTAD.GTINTPR[0] bit is 1, an interrupt request is generated under the following conditions:

- In sawtooth-wave mode, interrupt requests are enabled at overflows (GTCNT counter value changes from GTPR register value to 0 during up-counting)
- In triangle-wave mode, interrupt requests are enabled at crests (GTCNT counter value changes from GTPR register value to GTPR register value minus 1)
- In count operation by a hardware source, interrupt requests are enabled at overflows (GTCNT counter value changes from GTPR register value to 0 during up-counting)

(10) GPTn_UDF Interrupt (n = 0 to 17)

When the GTINTAD.GTINTPR[1] bit is 1, an interrupt request is generated under the following conditions:

- In sawtooth-wave mode, interrupt requests are enabled at underflows (GTCNT counter value changes from 0 to GTPR register value during down-counting)
- In triangle-wave mode, interrupt requests are enabled at troughs (GTCNT counter value changes from 0 to 1)
- In counting by a hardware source, interrupt requests are enabled at underflows (GTCNT counter value changes from 0 to GTPR register value during down-counting)

(11) GPTm_DTE Interrupt (m = 0 to 13)

When automatic dead time setting is made, the GTST.DTEF flag becomes 1 when the timer output toggle point with dead time added exceeds the count period. If GTINTAD.GRPDTE bit is 1 at this time, a dead time error interrupt request is generated.

In addition, when the timer output toggle point with dead time added is back within the count period, the GTST.DTEF flag changes from 1 to 0.

Table 19.22 Relationship between interrupt signals and interrupt enable bits (1 of 2)

Interrupt signal	Interrupt enable bit	Interrupt status flag
GPTn_CCMPA	GPTn.GTINTAD.GTINTA bit	—
GPTn_CCMPB	GPTn.GTINTAD.GTINTB bit	—
GPTn_CMPC	GPTn.GTINTAD.GTINTC bit	—
GPTn_CMPD	GPTn.GTINTAD.GTINTD bit	—
GPTn_CMPE	GPTn.GTINTAD.GTINTE bit	—
GPTn_CMPF	GPTn.GTINTAD.GTINTF bit	—
GPTm_DTE	GPTm.GTINTAD.GRPDTE bit	GPTm.GTST.DTEF
GPTm_ADTRGA	GPTm.GTINTAD.ADTRAUEN bit	GPTm.GTST.ADTRAUF bit
	GPTm.GTINTAD.ADTRADEN bit	GPTm.GTST.ADTRADF bit
GPTm_ADTRGB	GPTm.GTINTAD.ADTRBUEN bit	GPTm.GTST.ADTRBUF bit
	GPTm.GTINTAD.ADTRBDEN bit	GPTm.GTST.ADTRBDF bit

Table 19.22 Relationship between interrupt signals and interrupt enable bits (2 of 2)

Interrupt signal	Interrupt enable bit	Interrupt status flag
GPTn_OVF	GPTn.GTINTAD.GTINTPR[1:0] bits	—
GPTn_UDF		—

Note: n = 0 to 17, m = 0 to 13

19.4.2 Interrupt and A/D Conversion Start Request Skipping Function

19.4.2.1 Interrupt Skipping Function by GTITC Register

By setting the GTITC register, the GTCNT counter overflow (GTPR register compare match) interrupt (GTPn_OVF) and underflow interrupt (GTPn_UDF) can be skipped. Other interrupts and A/D converter start request signals can be skipped in coordination with the GTPn_OVF/GTPn_UDF skipping function. However, the dead time error interrupts cannot be linked with the GTPn_OVF/GTPn_UDF skipping function.

The interrupt skipping function is related only to the GTITC register setting, and is not related to setting of the GTINTAD register interrupt enable bit.

When both troughs and crests are counted and skipped in triangle-wave mode, if the number of times of skipping is odd, GTPn_OVF/GTPn_UDF interrupt requests cannot be generated at troughs only or at crests only depending on the skipping counter start timing. Therefore, to count both troughs and crests and generate the GTPn_OVF/GTPn_UDF interrupts at troughs only or crests only in triangle-wave mode, the number of times of skipping should be even.

Similarly, in sawtooth-wave mode, when both overflows and underflows are counted and skipped with the count direction changed, GTPn_OVF/GTPn_UDF interrupt requests cannot be generated at overflows only or at underflows only.

Therefore, to count both overflows and underflows with the count direction changed and generate the GTPn_OVF/ GTPn_UDF interrupts at overflows only or underflows only in sawtooth-wave mode, the skipping state should be carefully checked before using.

When changing the skipping count, be sure to release the skipping count setting (GTITC.IVTC[1:0] bits = 00b).

Figure 19.68 to Figure 19.73 show examples of skipping function operation.

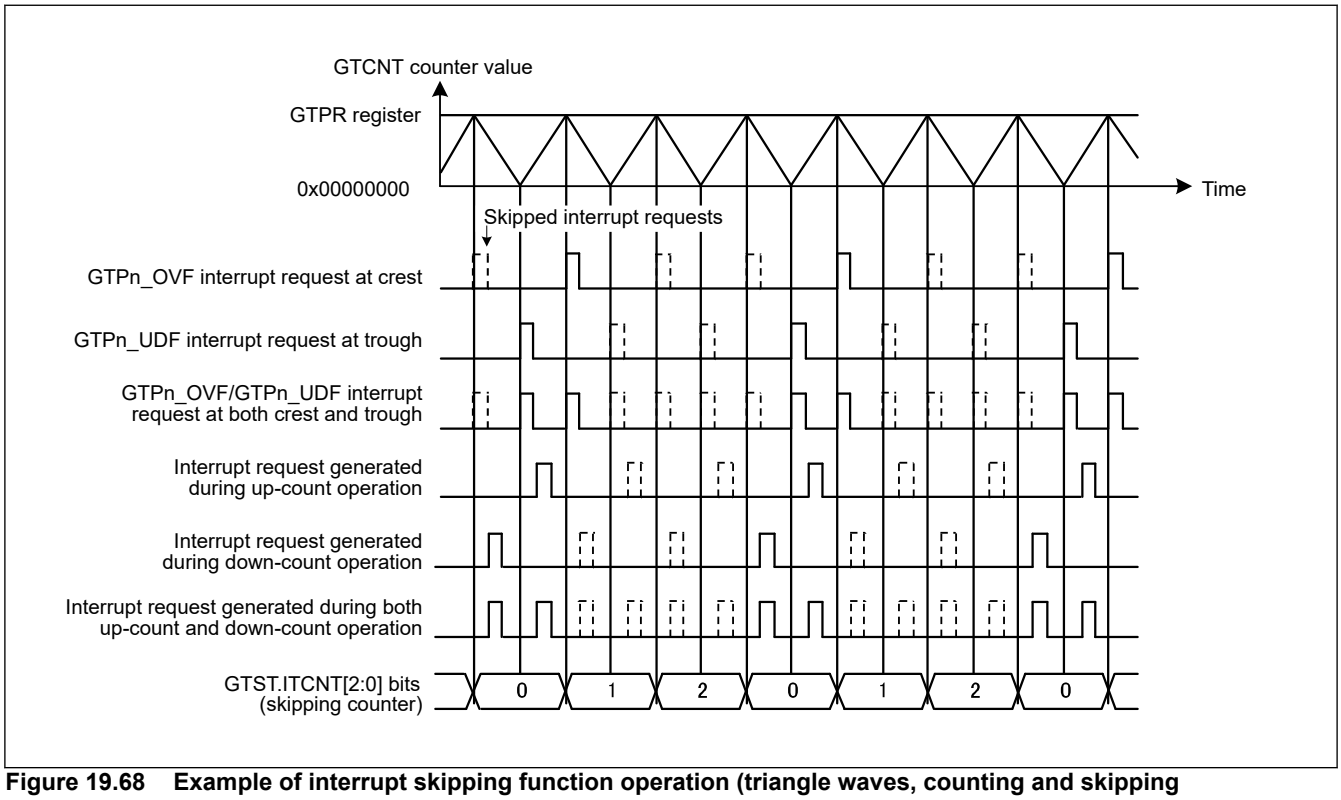


Figure 19.68 Example of interrupt skipping function operation (triangle waves, counting and skipping crests, skipping count = 2)

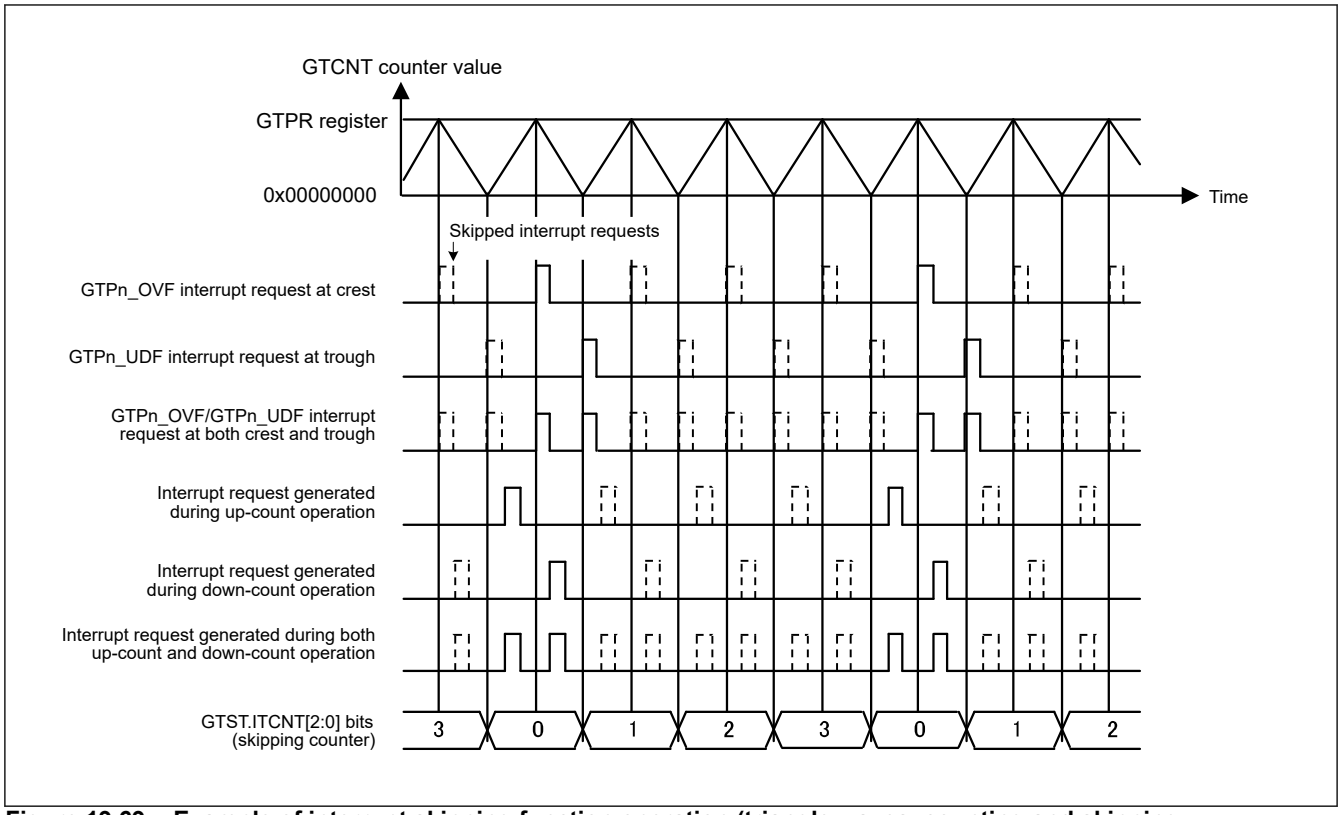


Figure 19.69 Example of interrupt skipping function operation (triangle waves, counting and skipping troughs, skipping count = 3)

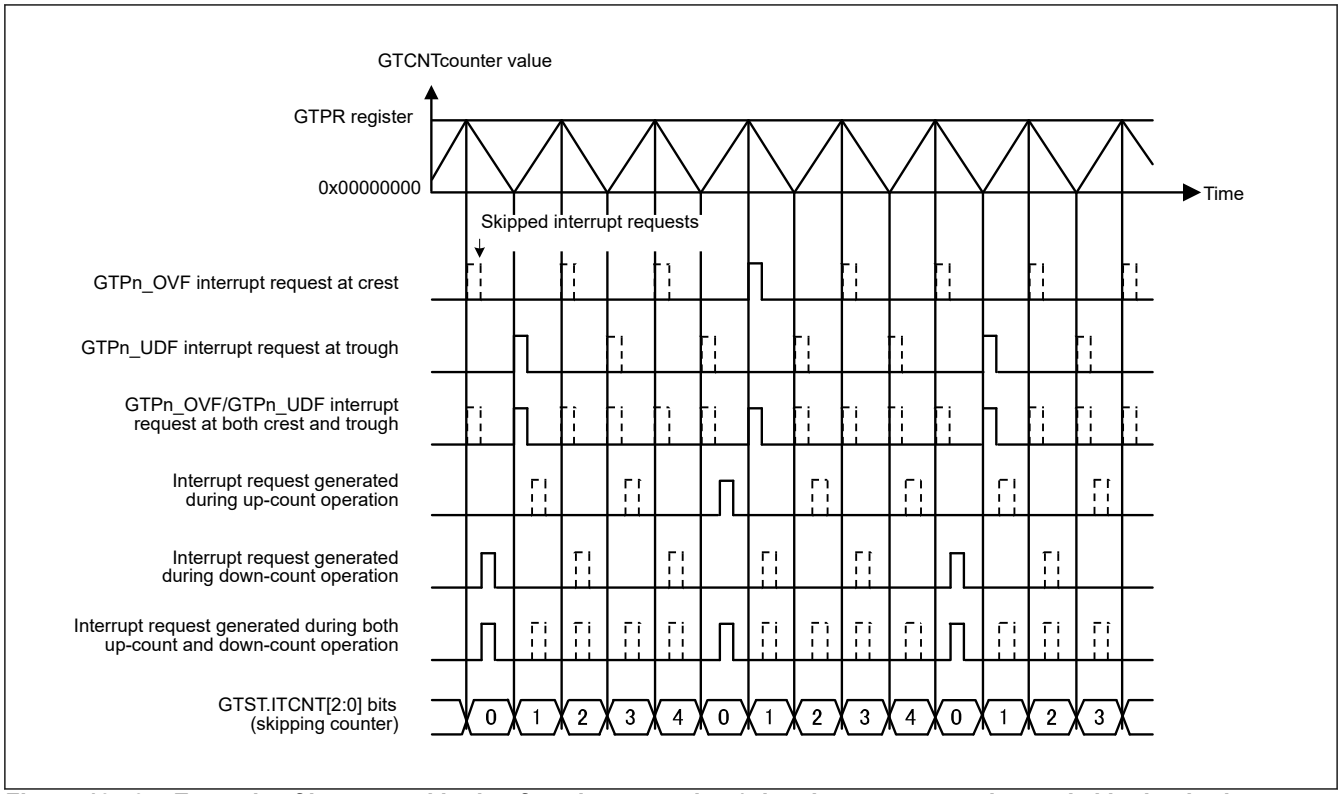


Figure 19.70 Example of interrupt skipping function operation (triangle waves, counting and skipping both troughs and crests, skipping count = 4)

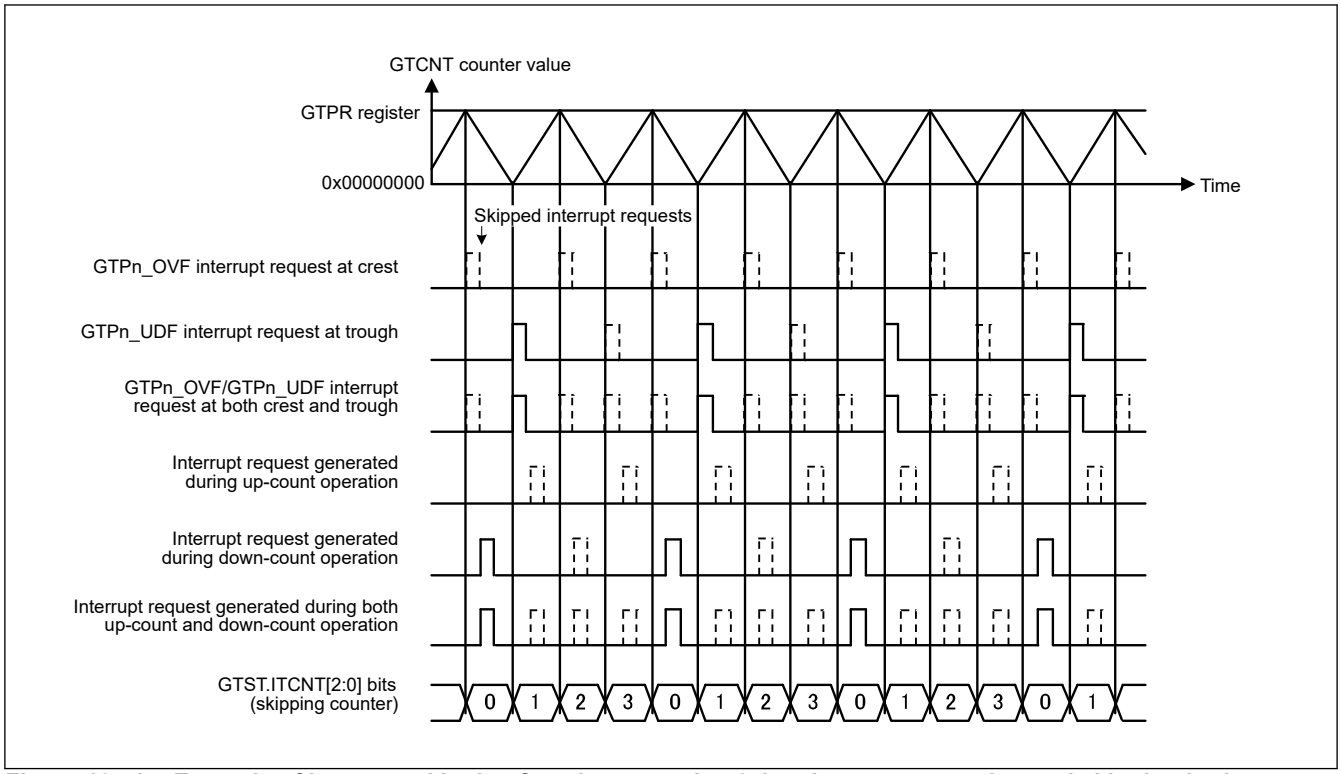


Figure 19.71 Example of interrupt skipping function operation (triangle waves, counting and skipping both troughs and crests, skipping count = 3, skipping started at up-counting)

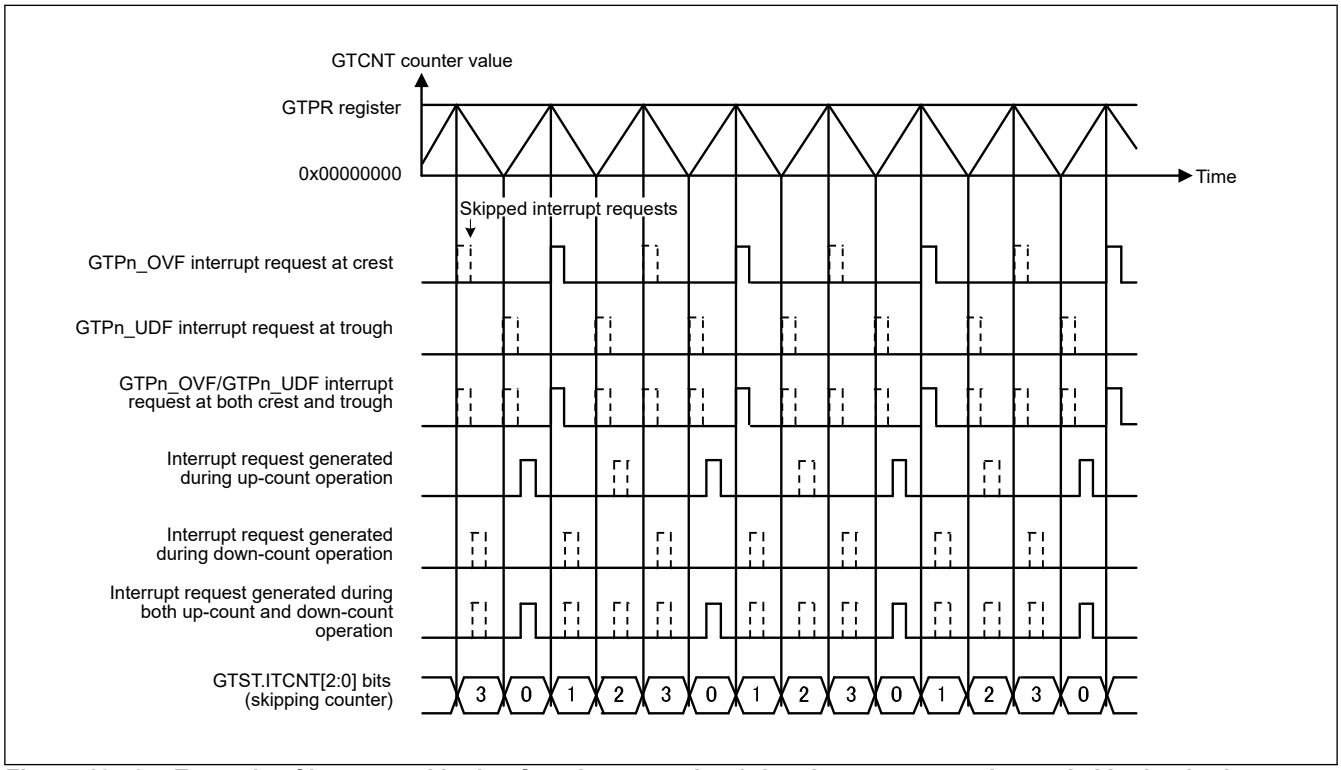


Figure 19.72 Example of interrupt skipping function operation (triangle waves, counting and skipping both troughs and crests, skipping count = 3, skipping started at down-counting)

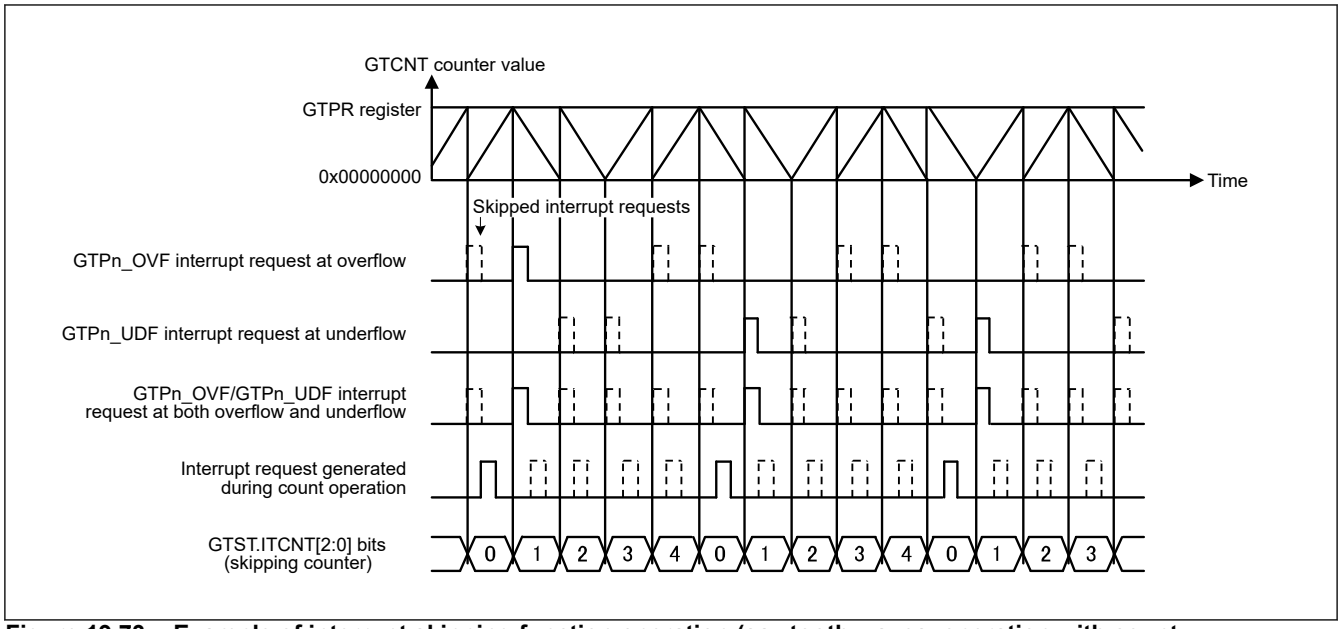


Figure 19.73 Example of interrupt skipping function operation (sawtooth waves, operation with count direction changed, counting and skipping both overflows and underflows, skipping count = 4)

19.4.2.2 Extended Interrupt Skipping Function

Overflow/underflow interrupt, compare match/input capture interrupt, A/D converter start request, and buffer transfer can be skipped by counting the GTCNT counter overflow or underflow based on settings of the GTEITC, GTEITLI1, GTEITLI2, and GTEITLB registers. A dead time error interrupt cannot be skipped.

Overflow/underflow interrupt, compare match/input capture interrupt, A/D converter start request, and buffer transfer can individually be set for skipping and the skipping period using the GTEITLI1, GTEITLI2, and GTEITLB registers.

The skipping period is set, related to the operation of the two independent extended interrupt skipping counters (EITCNT1[3:0] and EITCNT2[3:0] bits in the GTEITC registers), as a period that either of the counter value for such skipping counters is other than 0 or other than the skipping count. The skipping period can also be set as a period that both skipping counters are set as other than 0 or other than the skipping count for the counter value.

Figure 19.74 shows the counter operation for interrupt skipping by the GTITC register and extended interrupt skipping. The counter operation for extended interrupt skipping is set by the GTEITC register.

The EITCNT1[3:0] bits count the count source (in Figure 19.74, a crest is selected) selected by the extended interrupt counter 1 count source select bits (GTEITC.EIVTC1[1:0]) when the initial value is set for 0, and repeat the count operation which returns to 0 when the skipping count (in Figure 19.74, the count is 2) is set by the extended interrupt skipping 1 skipping count setting bits (EIVTT1[3:0]) is achieved.

The EITCNT2[3:0] bits can set the initial value, count the count source (in Figure 19.74, a trough is selected) selected by the extended interrupt counter 2 count source select bits (EIVTC2[1:0]), and repeat the count operation which returns to 0 when the skipping count (in Figure 19.74, the count is 2) is set by the EIVTT2[3:0] bits is achieved. The initial value is set when the GTEITC register is written by the access of upper 16 bits or 32 bits and when the written value to the EIVTC2[1:0] bits is other than 00b, while the setting for the extended interrupt skipping counter 2 is not in counting (EIVTC2[1:0] bits are 00b).

When the initial value is set, the written value to the extended interrupt skipping counter 2 initial value bits (EITCNT2IV[3:0]) is set as the initial value for the EITCNT2[3:0] bits.

The extended interrupt skipping counter starts up-counting at the first count clock after the setting is modified from not counting to counting.

The interrupt skipping count counter (GTST.ITCNT[2:0] bits) by the GTITC register is reset to 000b when the GTCNT counter operation is stopped: however, the EITCNT1[3:0] and EITCNT2[3:0] bits which are extended interrupt skipping functions retain the value even after the GTCNT counter operation is stopped, and the counting can be resumed from the value before the counter is stopped. When values for the EITCNT1[3:0] and EITCNT2[3:0] bits are to be reset (0000b), set the EIVTC1[1:0] and EIVTC2[1:0] bits to the setting for not counting (not skipping) (00b).

When the skipping count is to be changed, change the count after stopping the skipping counter operation (set either of the EIVTC1[1:0] bits or the EIVTC2[1:0] bits to 00b).

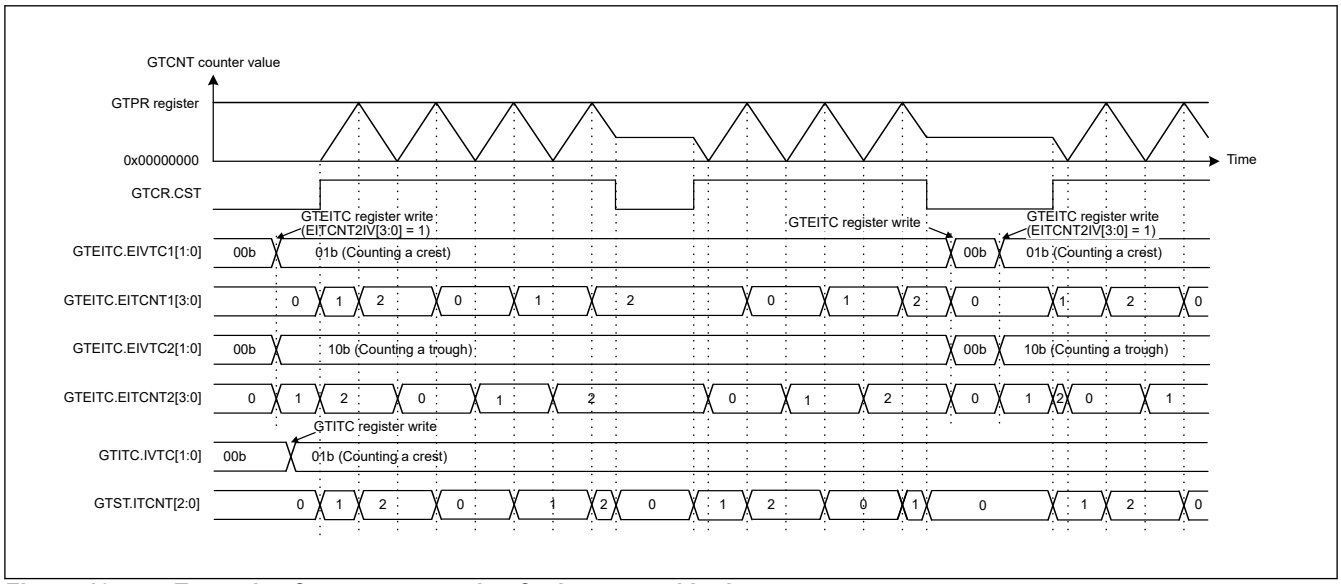


Figure 19.74 Example of counter operation for interrupt skipping

Interrupt skipping by the GTEITL1 register and A/D conversion start request skipping by the GTEITLI2 register can be performed simultaneously with interrupt skipping by the GTITC register. A skipping period at this time is represented by ORed skipping periods of respective registers.

Figure 19.75 shows corresponding interrupt skipping operations by different registers are performed simultaneously.

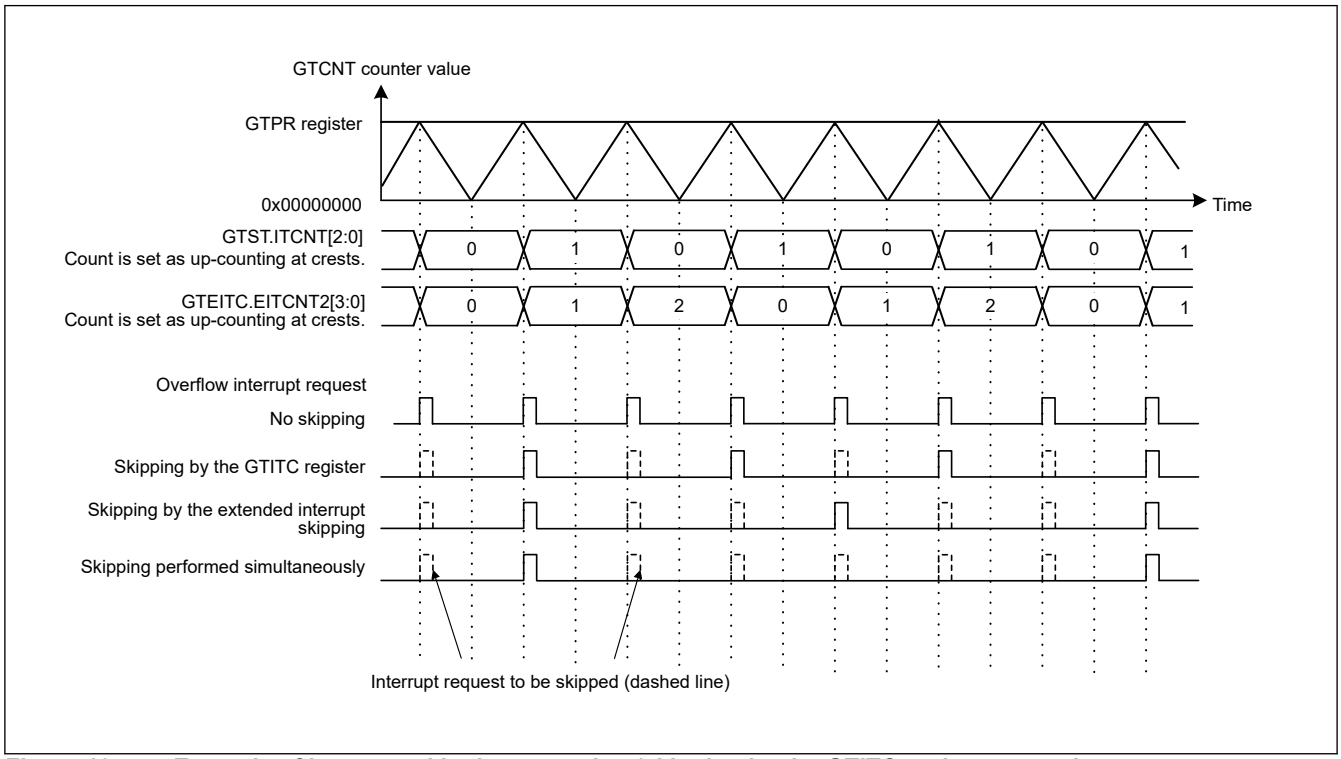


Figure 19.75 Example of interrupt skipping operation (skipping by the GTITC register: counting crests, extended interrupt skipping: EIVTC1[1:0] bits = 00b, EIVTC2[1:0] bits = 01b, EITLV[2:0] bits = 010b)

Skipping of ELC event source outputs corresponding to interrupts for which skipping can be set up in the GTEITLI1 register is only based on the settings of the GTITC register and the extended interrupt skipping register, and has no

connection with the setting of the interrupt enable bit in the GTINTAD register. The interrupt enable bit is used only for the interrupt signal output after skipping.

When A/D converter start request skipping which can be set by the GTEITLI2 register is performed, the ELC event output depend on the A/D converter start request enable bit in the GTINTAD register. All the operations by A/D converter start request which are set as disable by the GTINTAD register is not performed.

A buffer transfer skipping by the GTEITLB register is performed in all of buffer operation which is enabled in the GTBER and GTDTCR registers, or all buffer operations performed by sawtooth-wave one-shot pulse mode or triangle-wave PWM mode 3.

An interrupt skipping and a buffer transfer skipping are operated individually. An interrupt output without performing a buffer transferred performing a buffer transfer without an interrupt output can also be operated.

The following example shows the setting for the extended interrupt skipping.

The extended interrupt skipping counter 2 initial value is set by the written value to the ETICNT2IV[3:0] bits which is applied to change the setting from not counting the extended interrupt skipping counter 2 count source (GTEITC.EIVTC2[1:0] = 00b) to counting (EIVTC2[1:0] = other than 00b). Writing to the extended interrupt skipping counter 2 initial value bits (ETICNT2IV[3:0]) is performed only when the setting of the above-mentioned extended interrupt skipping counter 2 initial value is written.

1. Set GTCNT counter operation, set buffer operation, set compare match value
2. Set the extended interrupt skipping function
   Select the skipping counter used for skipping and the skipping period with the skipping function select bit for the interrupt, for the A/D converter start request, and for a buffer transfer, all of which are to be skipped, in the GTEITLI1, GTEITLI2, and GTEITLB registers, respectively.
3. Set the extended skipping counter
   With the GTEITC register, set a count source of the skipping counter used for skipping, skipping count, and the skipping counter 2 initial value as the following order.
   Set the EIVTCk[1:0] bits (k = 1, 2) as value other than 00b, and set the EIVTTk[3:0] bits as value other than 0000b.
   When the skipping counter 2 is used, change the EIVTC2[1:0] bits from 00b to a value other than 00b as well as set the EITCNT2IV[3:0] bits to the skipping counter 2 initial value.
4. Start count operation
   Set the GTCR.CST bit to 1 to start count operation.
5. Set buffer value by cycle

19.5 A/D Converter Start Request

An A/D converter start request can be issued at a compare match between the GTCNT counter and the GTADTRA or GTADTRB register, up-counting only, down-counting only, or both up-counting and down-counting can be specified by setting the GTINTAD register.

During event count operation, the A/D converter start request cannot be generated.

The A/D converter start request is not output directly to the A/D converter but output the event signal to the ELC.

The GTADTRA and GTADTRB registers each has two buffer registers. Buffer operation with the GTADTRA register used together with the GTADTBRA and GTADTDBRA registers, and buffer operation with the GTADTRB register used together with the GTADTBRB and GTADTDBRB registers can be performed.

Figure 19.76 shows an example of A/D converter start request operation.

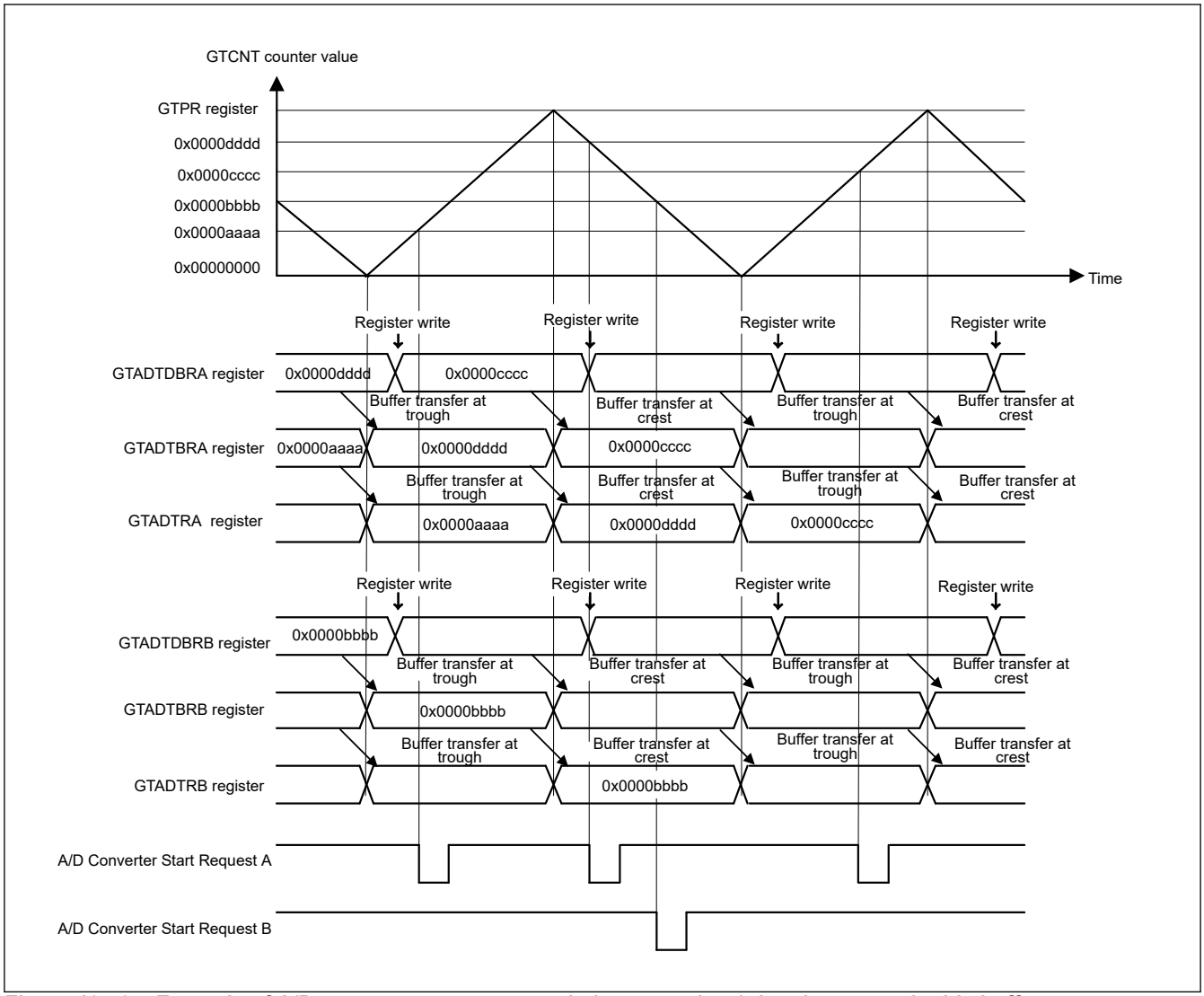


Figure 19.76 Example of A/D converter start request timing operation (triangle waves, double buffer operation, buffer transfer at both troughs and crests, A/D converter start request by GTADTRA register at both up-counting and down-counting, A/D converter start request by GTADTRB register at down-counting)

The following example shows setting for A/D converter start request operation.

1. Set operating mode
   Set the operating mode with the GTCR.MD[2:0] bits of the corresponding channel.
   In Figure 19.76, 100b, 101b, or 110b (triangle-wave PWM mode) is set.
2. Select count clock
   Select the count clock with GTCR.TPCS[3:0] bits of corresponding channel.
3. Period setting
   Set a period in the GTPR register of corresponding channel.
4. Set initial value for counter
   Set the initial value in the GTCNT counter of the corresponding channel.
5. Set buffer operation
   Set buffer operation with the ADTTA[1:0], ADTTB[1:0], ADTDA, and ADTDB bits in the GTBER register.
   In Figure 19.76, ADTTA[1:0] bits = 11b, ADTTB[1:0] bits = 11b, ADTDA bit = 1, and ADTDB bit = 1.
6. Set compare match value
   Set the A/D converter start request point in the GTADTRA and GTADTRB registers.
7. Set buffer value

For buffer operation, set the A/D converter start request point in one cycle after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers, respectively.
For double buffer operation, set the A/D converter start request point in two cycles after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers, respectively.

8. Set A/D converter start request for monitoring
   Select the A/D converter start request signal to be monitored with the ADSMS0[1:0] and ADSMS1[1:0] bits in the GTADSMR register from the GTADSM0 and GTADSM1 pins and enable output of the A/D converter start request signal being monitored to the ADSMEN0 and ADSMEN1 bits in the GTADSMR register.
9. Enable A/D converter start request
   Set to enable A/D converter start request with the ADTRAUEN, ADTRADEN, ADTRBUEN, and ADTRBDEN bits in the GTINTAD register.
   In Figure 19.76, ADTRAUEN bit = 1, ADTRADEN bit = 1, ADTRBUEN bit = 0, and ADTRBDEN bit = 1.
10. Start count operation
    Set the GTCR.CSR bit to 1 to start count operation.
11. Set buffer value by cycle
    For buffer operation, set the A/D converter start request point in one cycle after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or half cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTBRA and GTADTBRB registers, respectively.
    For double buffer operation, also set the A/D converter start request point in two cycles after the current cycle (in sawtooth-wave mode or triangle-wave mode with buffer transfer at trough or crest) or one cycle after the current cycle (in triangle-wave mode with buffer transfer at both trough and crest) in the GTADTDBRA and GTADTDBRB registers, respectively.

19.6 Operations Linked by the ELC

19.6.1 Event Signal Output to the ELC

GPT is capable of operation linked with another module set in advance when its interrupt request signal is used as an event signal by the event link controller (ELC).

The event signal can be output regardless of the setting of the corresponding interrupt request enable bits except the A/D converter start request.

The A/D converter start request during up-counting/down-counting can be enabled/disabled individually with the A/D converter start request enable bit to output both interrupts and events output to ELC.

The corresponding channels in LLPP and NONSAFETY respectively have 10 event signals to the ELC in the GPT, and they are shown below. A dead time error interrupt does not have a corresponding event signal.

- Generation of compare match A interrupt
- Generation of compare match B interrupt
- Generation of compare match C interrupt
- Generation of compare match D interrupt
- Generation of compare match E interrupt
- Generation of compare match F interrupt
- Generation of overflow interrupt
- Generation of underflow interrupt
- Generation of A/D converter start request A (Only used for ELC)
- Generation of A/D converter start request B (Only used for ELC)

19.6.2 Operations in Response to Receiving Event Signals from the ELC

The GPT can perform the following operations by the signals for sources A to H which are output from the ELC. Each signal is provided to all channels. Select an event source for the desired operation by a relevant source select register in a channel:

- Operation of count start/count stop/counter clear
- Operation of switch and up-counting/down-counting
- Operation of input capture A and B

See operation by hardware sources in section 19.3. Operation for respective operations.

19.7 Noise Filter Function

Each pin for use in input capture to the GPT is equipped with a noise filter. The noise filter samples input signals at the sampling clock (Core clock) and removes the pulses of which length is less than 3 sampling cycles.

The noise filter functionality includes enabling and disabling of the noise filter for each pin and setting of the sampling clock for each channel.

Figure 19.77 shows the timing of noise filtering.

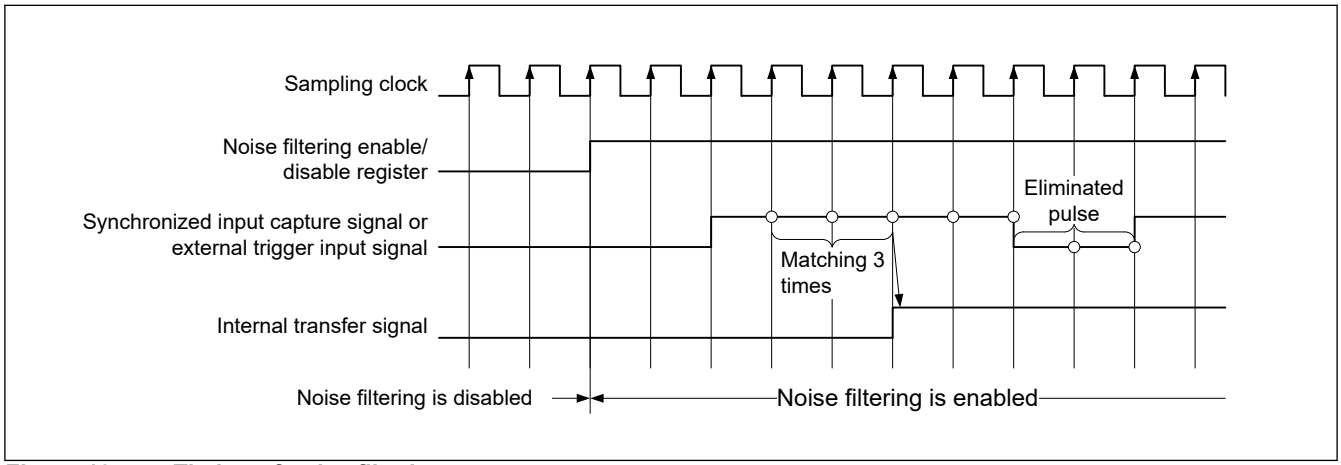


Figure 19.77 Timing of noise filtering

19.8 Protection Function

19.8.1 Write-Protection for Registers

To avoid incorrect writing to registers, write access to registers can be enabled or disabled per channel by setting the GTWP.WP bit for the given channel.

The WP bits enable or disable writing to the following registers:

GTSSR, GTPSR, GTCSR, GTUPSR, GTDNSR, GTICASR, GTICBSR, GTCR, GTUDDTYC, GTIOR, GTINTAD, GTST, GTBER, GTITC, GTCNT, GTCCRA, GTCCRB, GTCCRC, GTCCRD, GTCCRE, GTCCRF, GTPR, GTPBR, GTPDBR, GTADTRA, GTADTBRA, GTADTDBRA, GTADTRB, GTADTBRB, GTADTDBRB, GTDTCR, GTDVU, GTDVD, GTDBU, GTDBD, GTSOS, GTSOTR, GTADSMR, GTEITC, GTEITLI1, GTEITLI2, GTEITLB, GTSWSR, GTSWOS.

Every bit in registers GTSTR, GTSTP, and GTCLR, which can update the corresponding registers in other channels and can be updated by any of the corresponding registers in other channels conversely, can be protected by setting the GTWP.STRWP, STPWP, and CLRWP bits, respectively, per channel.

Likewise, writing to the GTSECSR and GTSECR registers, which can control all channel in group by writing to the GTSECSR and GTSECR registers of a given channel, can be enabled or disabled by the setting of the GTWP.CMNWP bit.

Protection by the GTWP register only targets writing operation by the CPU. Update to the register generated in related to the CPU writing is not protected.

19.8.2 Disabling of Buffer Operation

If the timing of buffer register write is too late for the buffer transfer timing, buffer operation can be suspended by setting the BD0, BD1, BD2, and BD3 bits in the GTBER register.

Specifically, buffer transfer can be temporarily disabled, even though a buffer transfer condition is generated during buffer register write, by setting the BD0, BD1, BD2, and BD3 bits to 1 (buffer operation disabled) before buffer register write, and setting the bits to 0 (buffer operation enabled) after completion of writing to all the buffer registers.

The BD0, BD1, BD2, and BD3 bits can be set on channel basis by writing directly to the GTBER register or can be set simultaneously by setting the GTSECR register for multiple channels which were set by the GTSECSR register. Figure 19.78 shows an example of operation for disabling buffer operation by writing to the GTBER register.

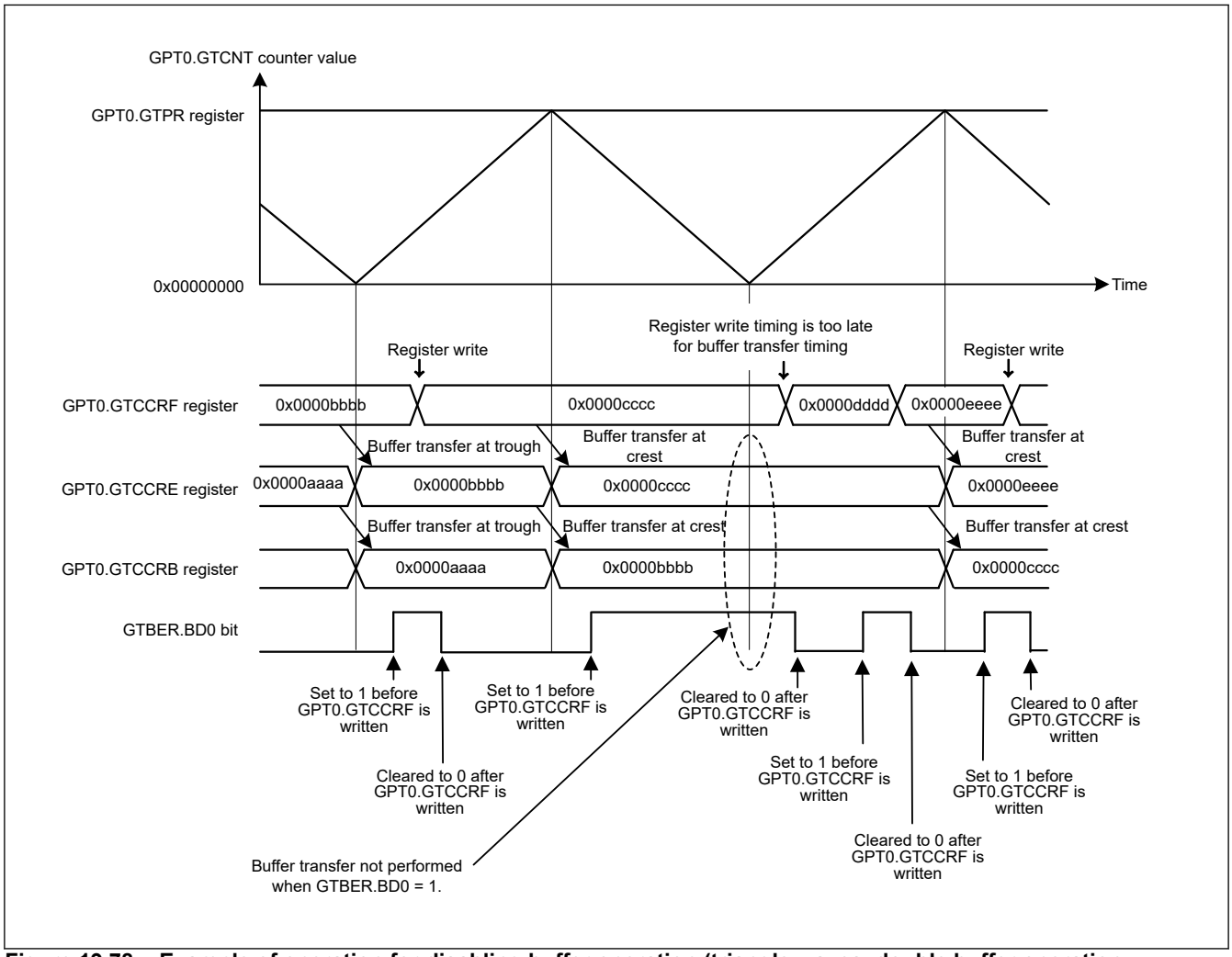


Figure 19.78 Example of operation for disabling buffer operation (triangle waves, double buffer operation, buffer transfer at both troughs and crests)

19.8.2.1 Simultaneous Control of Buffer Operations of Multiple Channels

The GTBER.BDn bit can be set by writing directly to the GTBER register per channel or by making settings in the GTSECR register for multiple channels that are already set in the GTSECSR register.

Use the following procedure to simultaneously set the GTBER.BDn bit of multiple channels.

1. Select the channels to simultaneously set the GTSECSR register
   Set the GTSECSR register so that the values at the bit positions for the corresponding channels for simultaneously setting of the GTBER.BDn bits become 1. All GTSECSR registers can be updated by writing to the GTSECSR register of any channel.
2. Simultaneously set the GTBER.BDn bits by updating the GTSECR register

In the GTSECR register, set the operation of the GTBER.BDn bit (enabling or disabling of buffer operation) that are to be simultaneously set. Writing to the GTSECR register from any channel updates the GTBER.BDn bit in all channels corresponding to the bits set as 1 in the GTSECSR register, in accordance with the value of the GTSECR register.

Figure 19.79 shows an example of simultaneously controlling the enabling or disabling of buffer operation for multiple channels.

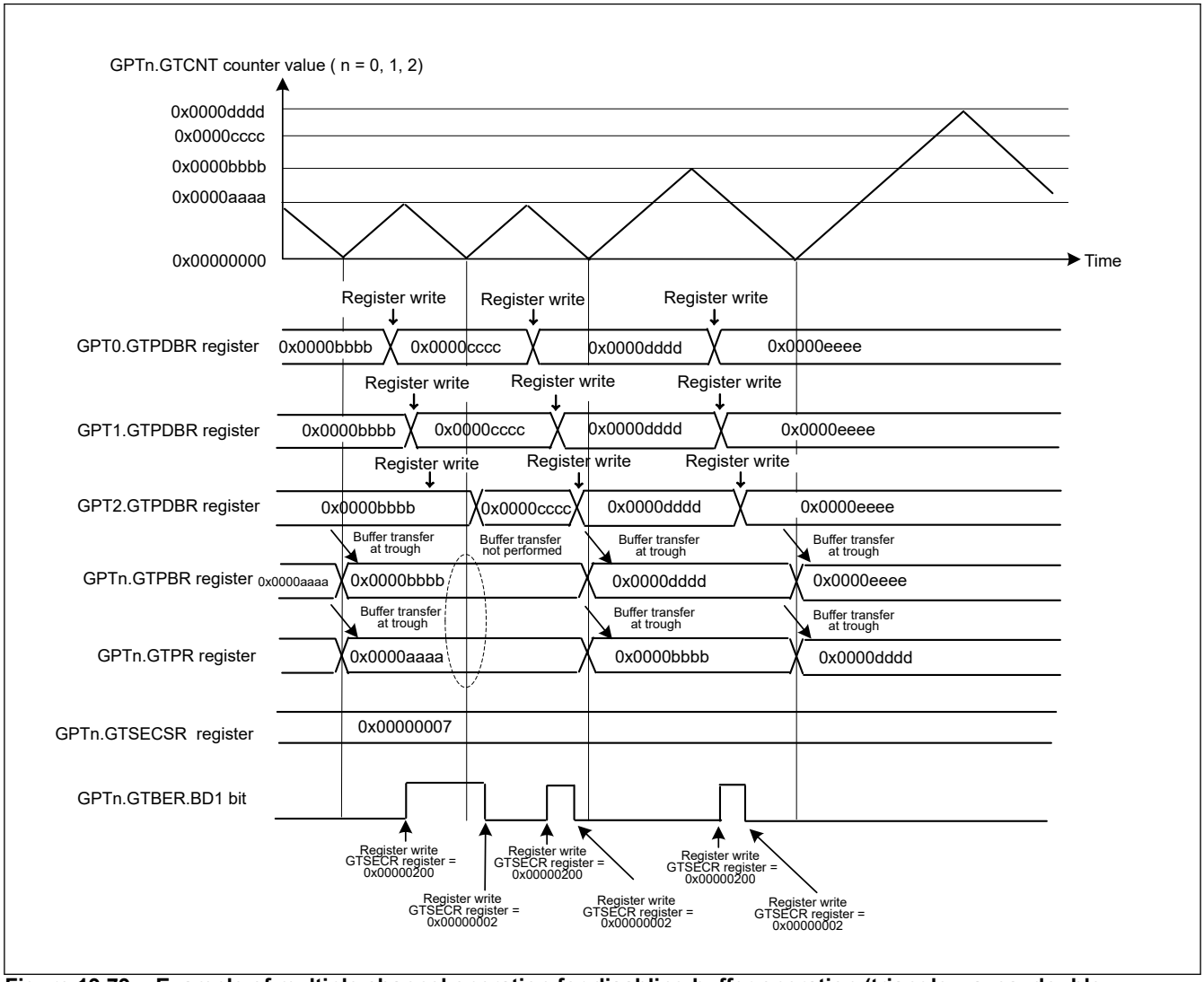


Figure 19.79 Example of multiple channel operation for disabling buffer operation (triangle waves, double buffer operation)

19.8.2.2 Repeated Double-Buffered Operation When Disabling GTCCR Buffer Transfer

When a GTBER.DBRTECm (m = A, B) bit is set to 1 in sawtooth-wave one-shot pulse mode or triangle-wave PWM mode 3, transfer from the intermediate buffer to the GTCCRm (m = A, B) register is repeated on a cyclic basis even while buffer transfer is disabled by the setting of the GTBER.BD0 bit or by the buffer transfer extended skipping function.

(1) In Sawtooth-Wave One-Shot Pulse Mode

In sawtooth-wave one-shot pulse mode, the compare match value for the first half of a waveform is stored in temporary register x (x = C, E) as the intermediate buffer for the GTCCRx (x = C, E) register and the compare match value for the second half of a waveform is stored in temporary register m (m = A, B) as the intermediate buffer for the GTCCRy (y = D, F) register, respectively, for compare match values during repeated buffer operation, and the given values are alternately transferred to the GTCCRm (m = A, B) registers.

Table 19.23 lists the types of buffer transfer of the GTCCR register during counting in sawtooth-wave one-shot pulse mode.

While counting is stopped, the setting of the value in the temporary register is transferred through forcible buffer transfer.

In forcible buffer transfer, the values of the GTCCRy (y = D, F) registers are transferred to temporary registers m (m = A, B), and the values of the GTCCRx (x = C, E) are transferred to temporary registers x (x = C, E) when the setting of the corresponding GTBER.DBRTECm (m = A, B) bit is 1, respectively.

When the setting of the GTBER.DBRTECm (m = A, B) bit is 1, values written by the CPU to the GTCCRm (m = A, B) registers are reflected as the values of temporary registers x (x = C, E).

Table 19.23 GTCCR buffer transfer operation in sawtooth-wave one-shot pulse mode during GTCNT counting

GTBER.DBRTECm	Buffer transfer	Timing of transfer				
		GTCCRx ↓ GTCCRm	GTCCRx ↓ Temporary register x	Temporary register x ↓ GTCCRm	GTCCRy ↓ Temporary register m	Temporary register m ↓ GTCCRm
0	Transfer enabled period	Overflow or underflow	No transfer	No transfer	Overflow or underflow	GTCCRm compare match
	Transfer disabled period	No transfer	No transfer	No transfer	No transfer	No transfer
1	Transfer enabled period	Overflow or underflow	Overflow or underflow	Not Transferred	Overflow or underflow	GTCCRm compare match
	Transfer disabled period	No transfer	No transfer	Overflow or underflow	No transfer	GTCCRm compare match

Note: m = A, B; x = C, E; y = D, F

Figure 19.80 shows the operation of generating transfer-disabled period by extended buffer transfer skipping as an example of repeated double buffer operations when the GTCCR buffer transfer is disabled in sawtooth-wave one-shot pulse mode.

Figure 19.81 shows the operation of generating transfer-disabled period by updating the GTBER.BD0 bit as an example of repeated double-buffer operations when the GTCCR buffer transfer is disabled in sawtooth-wave one-shot pulse mode.

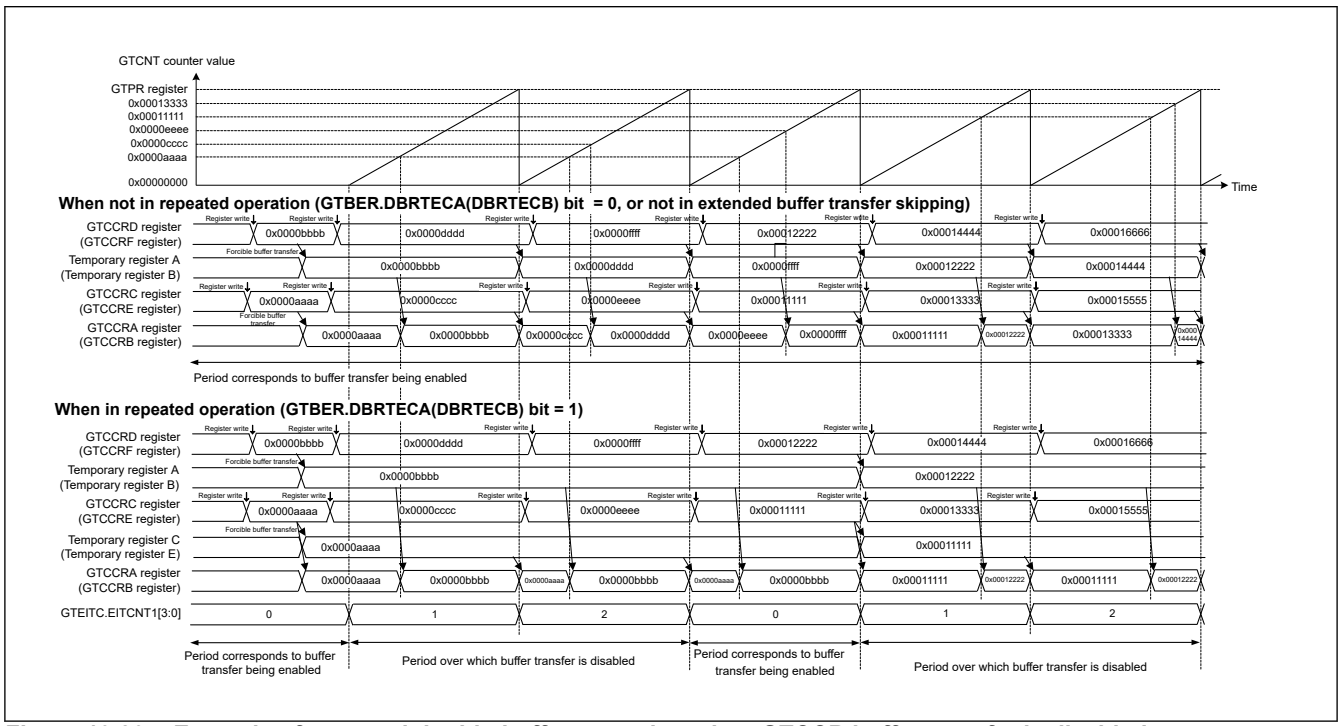


Figure 19.80 Example of repeated double-buffer operation when GTCCR buffer transfer is disabled (sawtooth-wave one-shot pulse mode, using extended buffer transfer skipping, GTBER.BD0 is constantly 0)

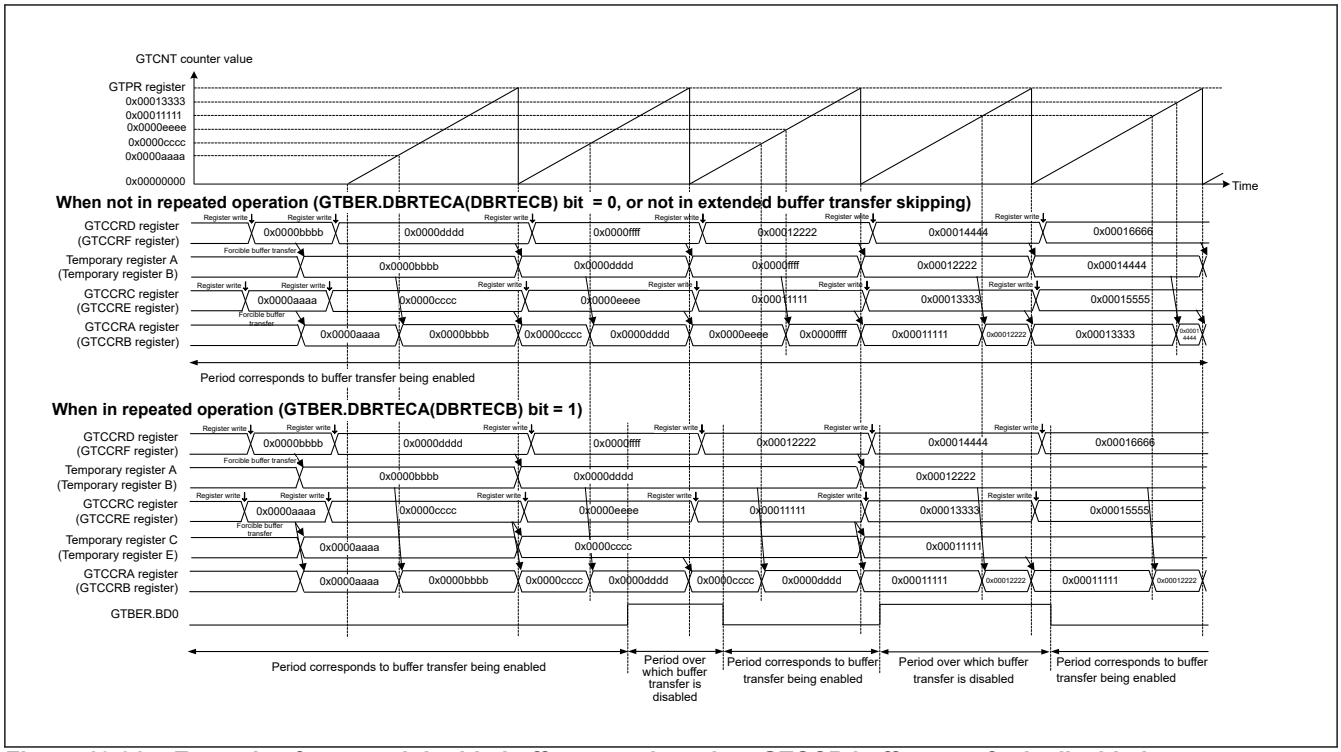


Figure 19.81 Example of repeated double-buffer operation when GTCCR buffer transfer is disabled (sawtooth-wave one-shot pulse mode, updating the GTBER.BD0 bit)

(2) In Triangle-Wave PWM Mode 3

In triangle-wave PWM mode 3, the compare match value for the first half of a waveform is stored in the temporary register x (x = C, E) as the intermediate buffer for the GTCCRx (x = C, E) register and the compare match value for the second half of a waveform is stored in temporary register m (m = A, B) as the intermediate buffer for the GTCCRy (y = D, F) register, respectively, for compare match values during repeated buffer operation, and the given values are alternately transferred to the GTCCRm (m = A, B) register.

Table 19.24 lists the types of buffer transfer of the GTCCR register during counting operation in triangle-wave PWM mode 3.

While counting is stopped, the setting of the value in the temporary register is transferred through forcible buffer transfer. In forcible buffer transfer, the values of the GTCCRy (y = D, F) registers are transferred to temporary registers m (m = A, B), and the values of the GTCCRx (x = C, E) are transferred to temporary registers x (x = C, E), when the setting of the corresponding GTBER.DBRTECm (m = A, B) bit is 1, respectively.

When the setting of the GTBER.DBRTECm (m = A, B) bit is set to 1, values written by the CPU to the GTCCRm (m = A, B) registers are reflected as the values of temporary registers x (x = C, E).

Table 19.24 GTCCR buffer transfer operation in triangle-wave PWM mode 3 during GTCNT counting

GTBER.DBRTECm	Buffer transfer	Timing of transfer				
		GTCCRx ↓ GTCCRm	GTCCRx ↓ temporary register x	temporary register x ↓ GTCCRm	GTCCRy ↓ temporary register m	temporary register m ↓ GTCCRm
0	Transfer enabled period	Trough	No transfer	No transfer	Trough	Crest
	Transfer disabled period	No transfer	No transfer	No transfer	No transfer	No transfer
1	Transfer enabled period	Trough	Trough	No transfer	Trough	Crest
	Transfer disabled period	No transfer	No transfer	Trough	No transfer	Crest

Note: m = A, B; x = C, E; y = D, F

Figure 19.82 shows the operation of generating transfer-disabled period by extended buffer transfer skipping as an example of repeated double buffer operations when the GTCCR buffer transfer is disabled in triangle-wave PWM mode 3.

Figure 19.83 shows the operation of generating transfer-disabled period by updating the GTBER.BD0 bit as an example of repeated double-buffer operations when the GTCCR buffer transfer is disabled in triangle-wave PWM mode 3.

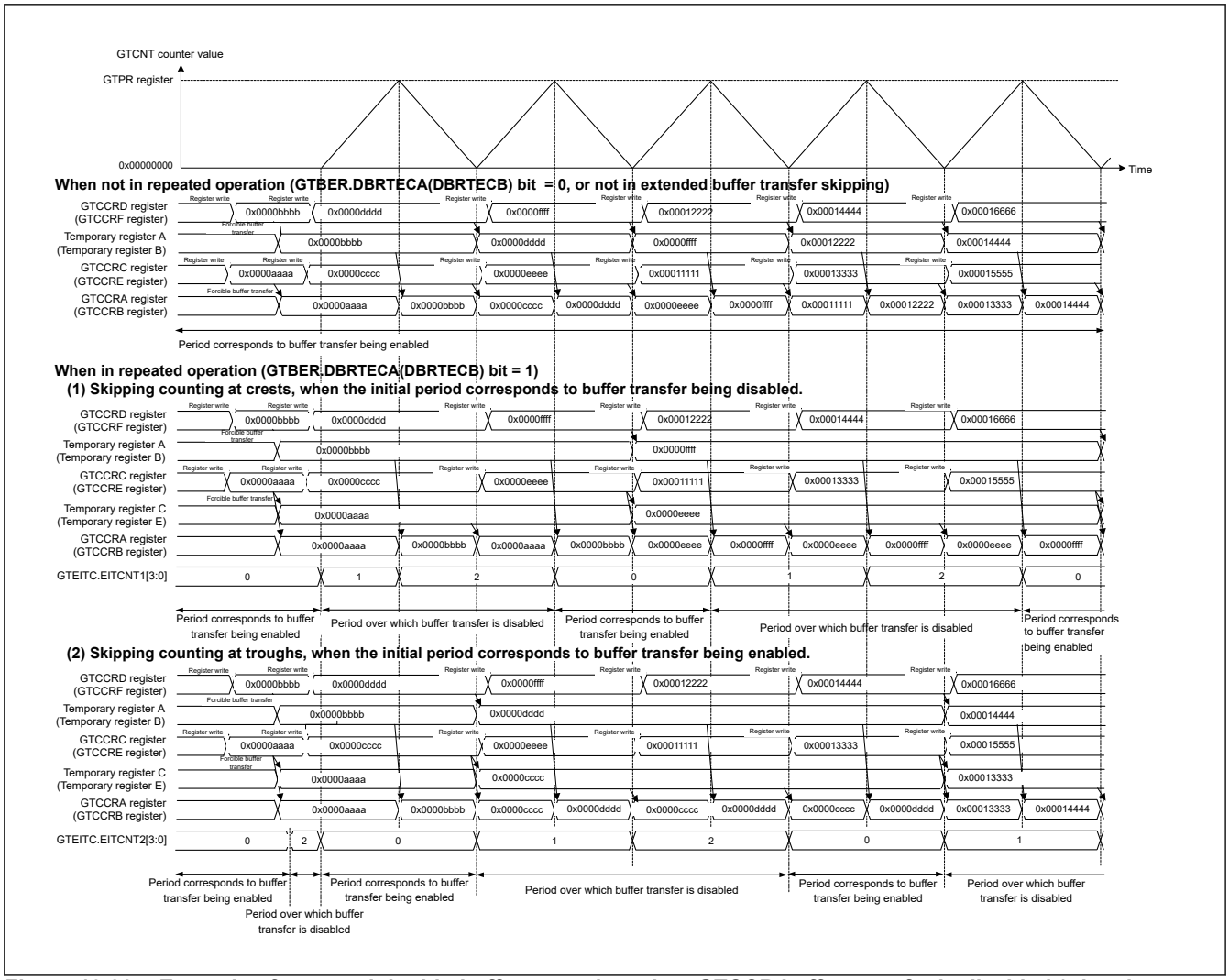


Figure 19.82 Example of repeated double-buffer operation when GTCCR buffer transfer is disabled (triangle-wave PWM mode 3, using extended buffer transfer skipping, GTBER.BD0 is constantly 0)

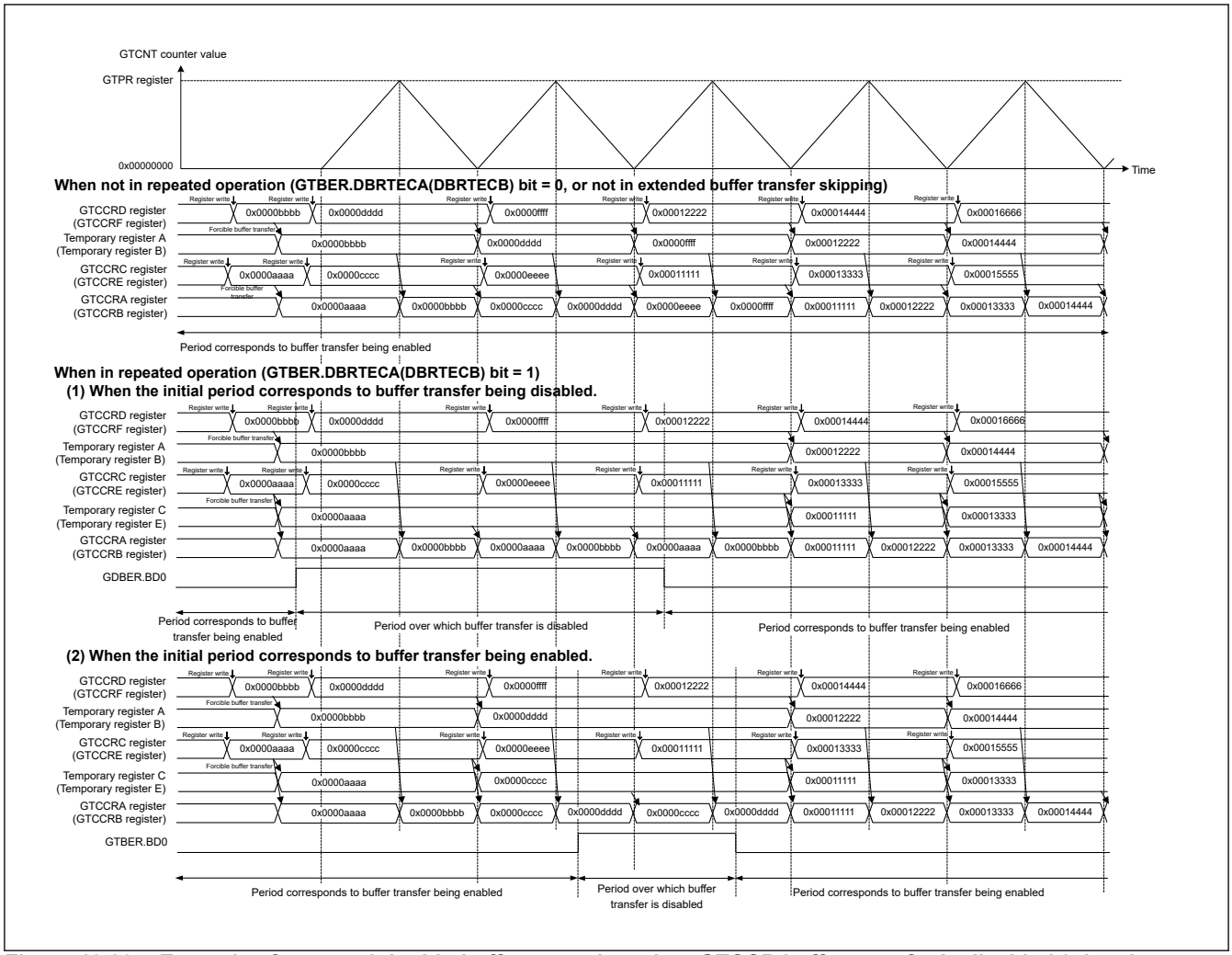


Figure 19.83 Example of repeated double-buffer operation when GTCCR buffer transfer is disabled (triangle-wave PWM mode 3, updating the GTBER.BD0 bit)

19.8.3 GTIOCnm Pin Output Negate Control (n = 0 to 17, m = A, B)

For protection from system failure, the output negate control that changes the GTIOCnm pin output forcibly by the output stop request from the POEG is provided.

Output protection is required when a dead-time error or the same output level on the GTIOCnA and GTIOCnB pins is detected. At that time, the detection of disabling of output is output to the group of the POEG set in the GTINTAD.GRP[1:0] bits based on the setting of the output disable detection enable bits, such as the GRPDTE, GRPABH, and GRPABL bits in the GTINTAD register. The POEG takes the logical OR of detection of the types as described and other forms of detection of the disabling of output, and outputs requests for the disabling of output to the GPT.

By setting the GTINTAD.GRP[1:0] bits, one output stop request can be selected out of four output stop requests which are input from the POEG as output stop request signal common in the GTIOCnA and GTIOCnB pins. Selected output stop request can be checked by reading the GTST.ODF flag.

The states of outputs at the time of control for their negation can be set with the GTIOR.OADF[1:0] bits for the GTIOCnA pin and with the GTIOR.OBDF[1:0] bits for the GTIOCnB pin.

Transition to the output negate condition by generating the output stop request from the POEG is performed asynchronously, while a release from the output negate condition by making the output stop request no longer satisfied is performed at the end of the cycle. It is after 3 PCLKGPTL (LLPP)/PCLKM (NONSAFETY, SAFETY) at shortest when the output negate condition is released after the output stop request is no longer satisfied. To reliably control output negation, clear the POEG flag for which the condition of the request to disable the output is no longer satisfied after 4 PCLKGPTL (LLPP)/PCLKM (NONSAFETY, SAFETY) cycles.

To release the output stop condition during event count operation or without waiting the end of the cycle, set the OADF[1:0] bits to 00b for the GTIOCnA pin, and set the OBDF[1:0] bits to 00b for the GTIOCnB pin. Figure 19.84 shows an example of the output negate control operation for the GTIOCnm pin output.

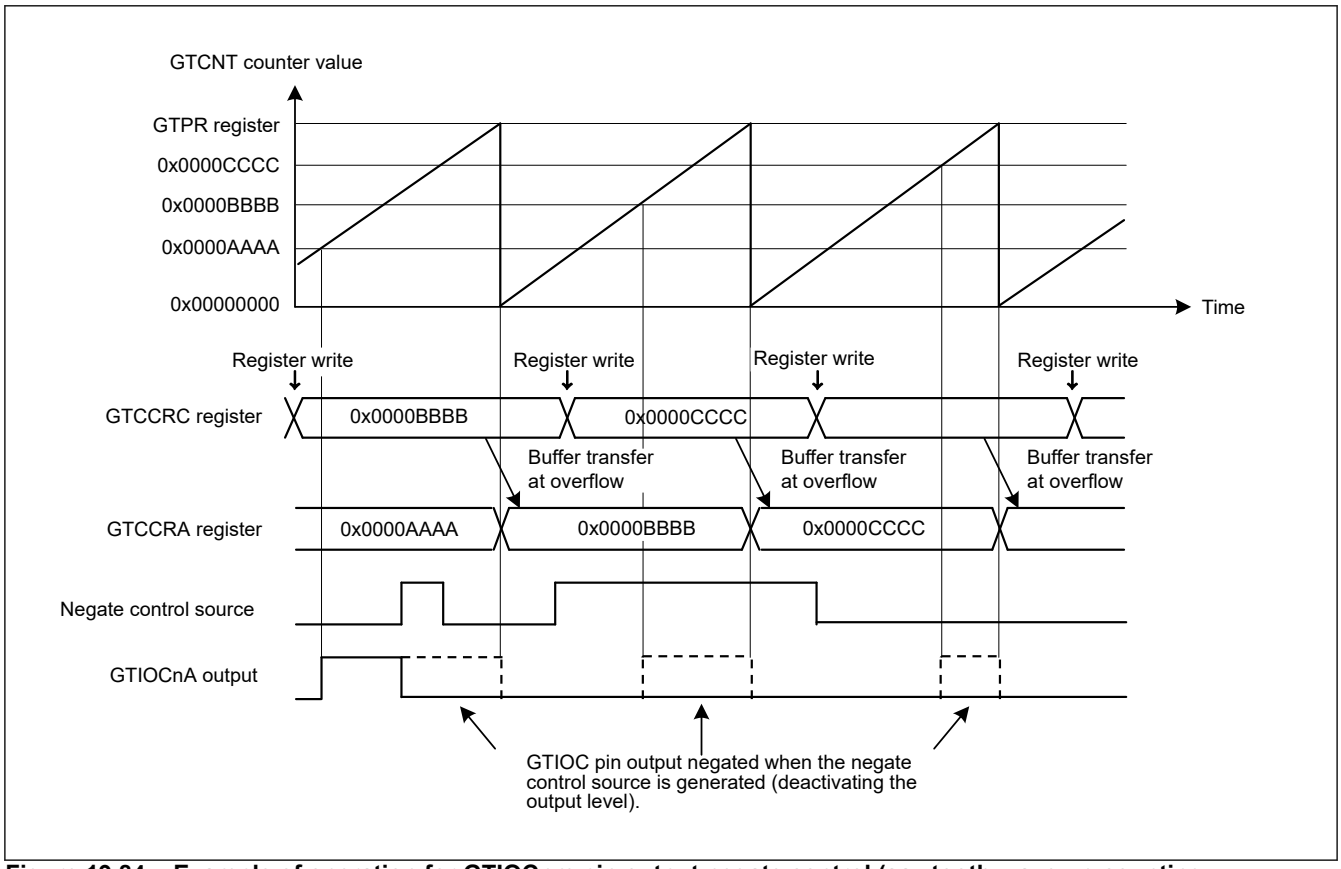


Figure 19.84 Example of operation for GTIOCnm pin output negate control (sawtooth-wave up-counting, buffer operation, active-level: high output at GTCCRA register compare match, low output at the end of the cycle, and low output at output negate) (n = 0 to 17, m = A, B)

19.8.4 GTIOCnm Protection Function for GTIOCnm Pin Output (n = 0 to 17, m = A, B)

To prepare for a case when an incorrect value (0x00000000 or a value greater than or equal to the GTPR register value) is set in the GTCCRA register, the output protection function for the GTIOCnm pin output (disabling function) is activated when the automatic dead time is set (GTDTCR.TDE bit = 1) in triangle-wave PWM mode.

The status of the output protection function can be read from the GTSOS.SOS[1:0] bits.

Figure 19.85 shows the output protection function state transition.

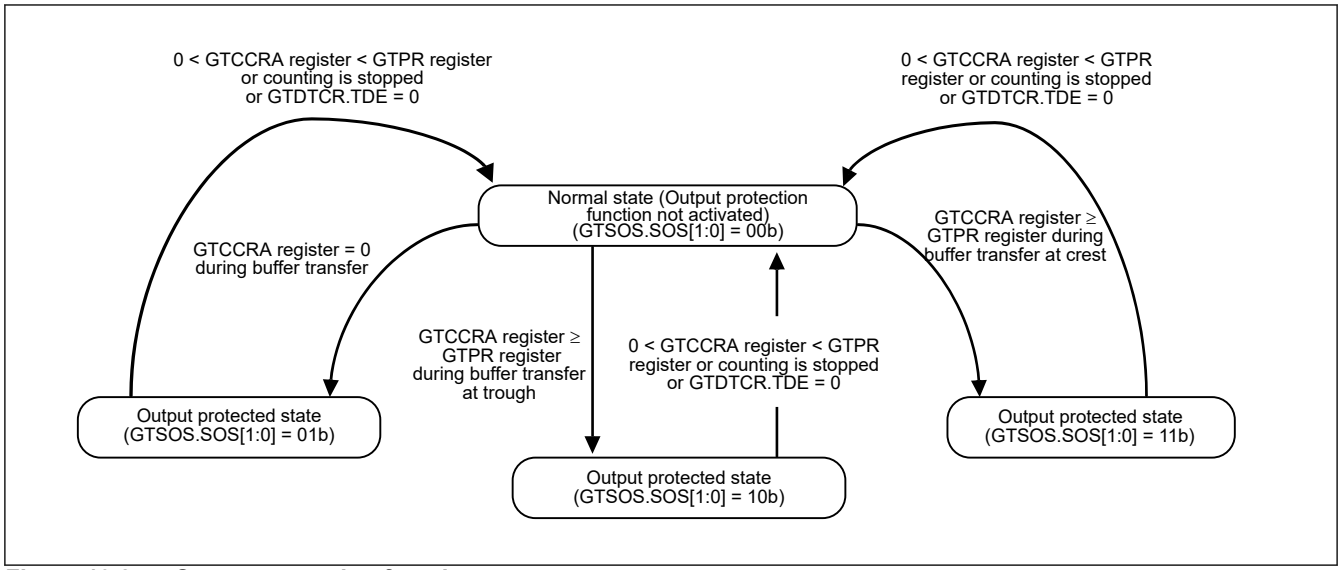
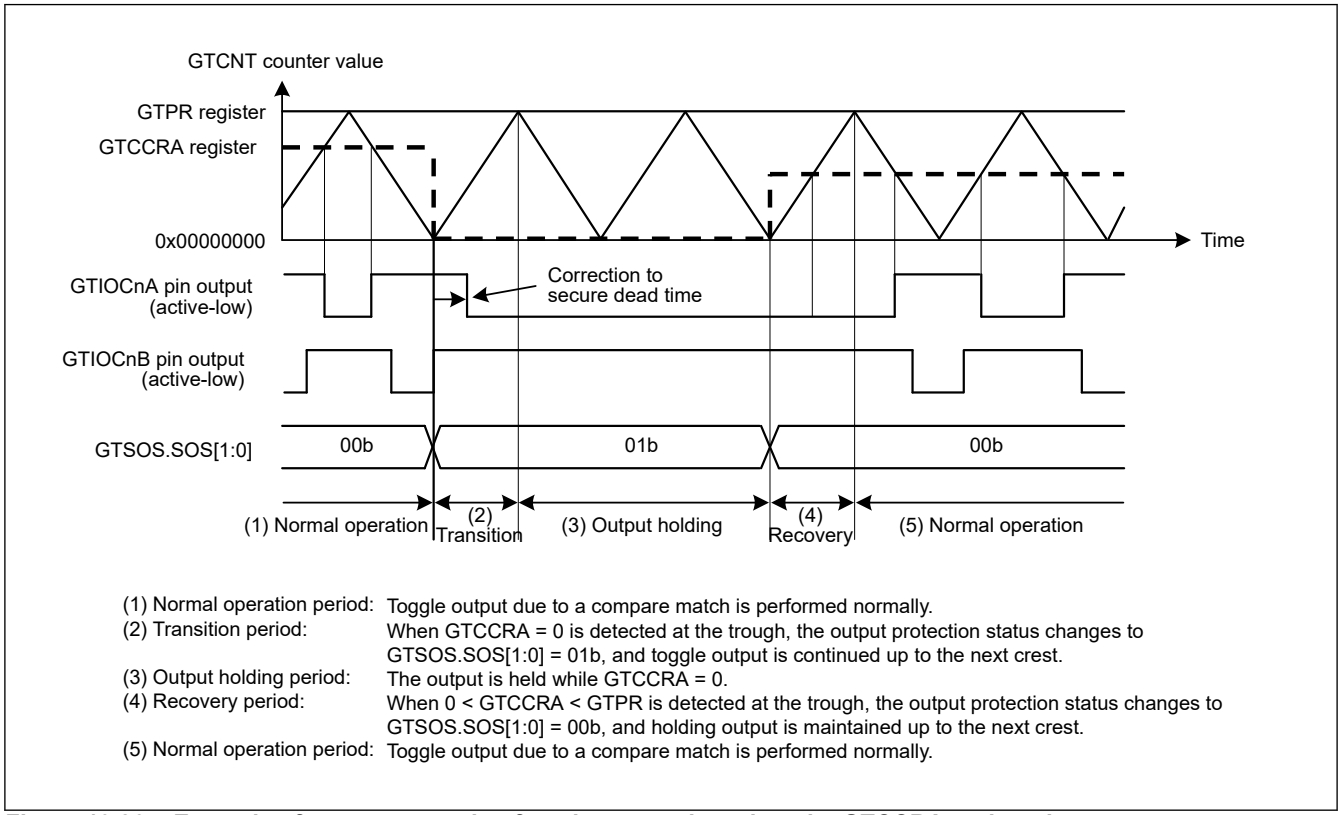


Figure 19.85 Output protection function

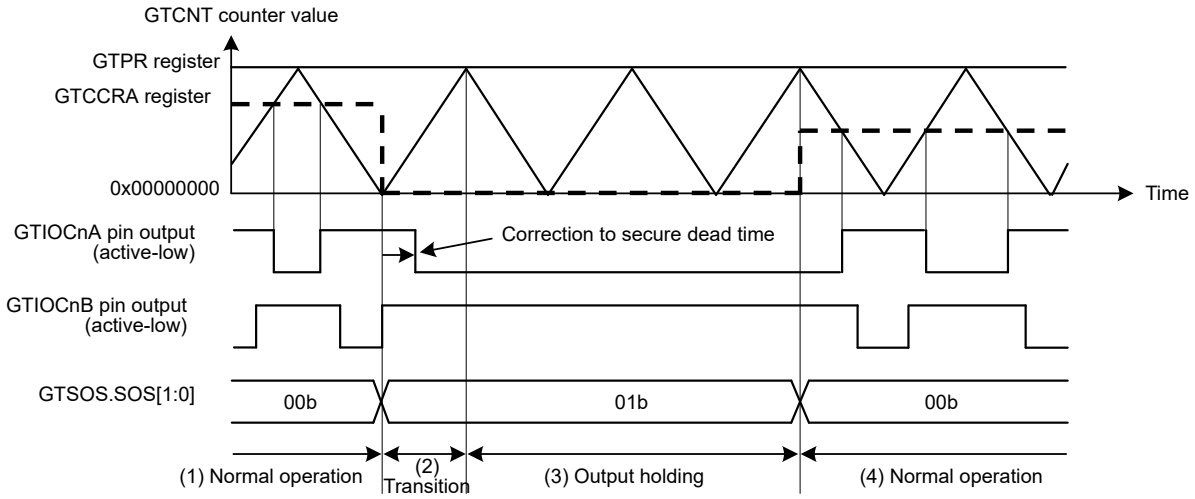
(1) Output Protection Function when the GTCCRA Register is Set to 0x00000000 during Buffer Transfer

Figure 19.86 and Figure 19.87 show examples of output protection function operation when the GTCCRA register is set to 0x00000000 during buffer transfer at troughs, and Figure 19.88 and Figure 19.89 show examples of output protection function operation when the GTCCRA register is set to 0x00000000 during buffer transfer at crests.



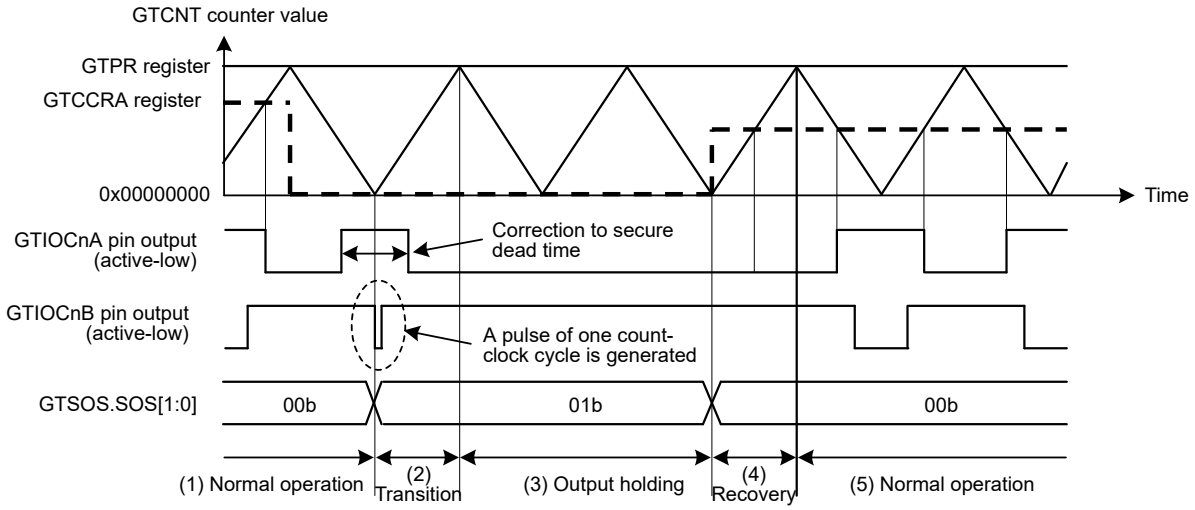
(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Transition period: When GTCCRA = 0 is detected at the trough, the output protection status changes to GTSOS.SOS[1:0] = 01b, and toggle output is continued up to the next crest.
(3) Output holding period: The output is held while GTCCRA = 0.
(4) Recovery period: When 0 < GTCCRA < GTPR is detected at the trough, the output protection status changes to GTSOS.SOS[1:0] = 00b, and holding output is maintained up to the next crest.
(5) Normal operation period: Toggle output due to a compare match is performed normally.

Figure 19.86 Example of output protection function operation when the GTCCRA register is set to 0x00000000 during buffer transfer at troughs (restored to 0 < GTCCRA register < GTPR register during buffer transfer at troughs, active-level: low) (n = 0 to 17)



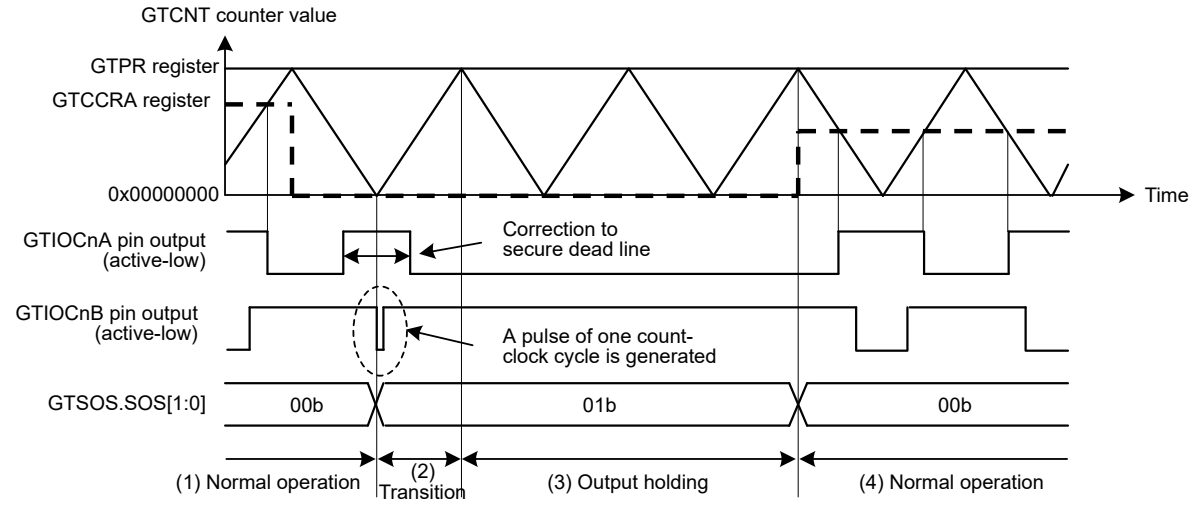
(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Transition period: When GTCCRA = 0 is detected at the trough, the output protection status changes to GTSOS.SOS[1:0] = 01b, and toggle output is continued up to the next crest.
(3) Output holding period: The output is held while GTCCRA = 0.
(4) Normal operation period: When 0 < GTCCRA < GTPR is detected at the crest, the output protection status changes to GTSOS.SOS[1:0] = 00b, and toggle output due to a compare match is performed normally.

Figure 19.87 Example of output protection function operation when the GTCCRA register is set to 0x00000000 during buffer transfer at troughs (restored to 0 < GTCCRA register < GTPR register during buffer transfer at crests, active-level: low) (n = 0 to 17)



(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Transition period: When GTCCRA = 0 is detected at the trough, the output protection status changes to GTSOS.SOS[1:0] = 01b, and toggle output is continued up to the next crest.
(3) Output holding period: The output is held while GTCCRA = 0.
(4) Recovery period: When 0 < GTCCRA < GTPR is detected at the trough, the output protection status changes to GTSOS.SOS[1:0] = 00b, and holding output is maintained up to the next crest.
(5) Normal operation period: Toggle output due to a compare match is performed normally.

Figure 19.88 Example of output protection function operation when the GTCCRA register is set to 0x00000000 during buffer transfer at crests (restored to 0 < GTCCRA register < GTPR register during buffer transfer at troughs, active-level: low) (n = 0 to 17)

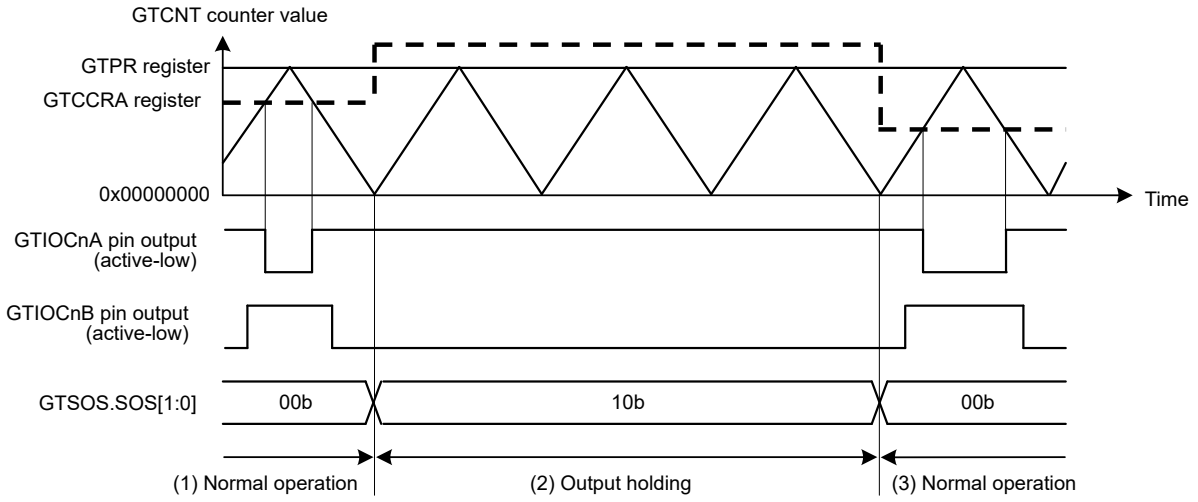


(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Transition period: When GTCCRA = 0 is detected at the trough, the output protection status changes to GTSOS.SOS[1:0] = 01b, and toggle output is continued up to the next crest.
(3) Output holding period: The output is held while GTCCRA = 0.
(4) Recovery period: When 0 < GTCCRA < GTPR is detected at the crest, the output protection status changes to GTSOS.SOS[1:0] = 00b, and toggle output due to a compare match is performed normally.

Figure 19.89 Example of output protection function operation when the GTCCRA register is set to 0x00000000 during buffer transfer at crests (restored to 0 < GTCCRA register < GTPR register during buffer transfer at crests, active-level: low) (n = 0 to 17)

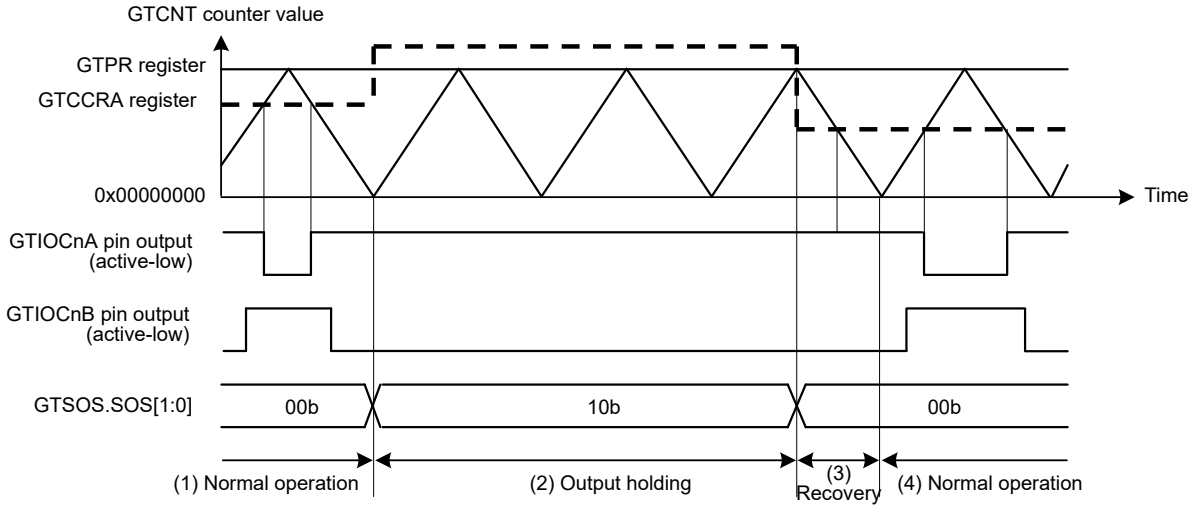
(2) Output Protection Function when GTCCRA Register ≥ GTPR Register is Set during Buffer Transfer at Troughs

Figure 19.90 and Figure 19.91 show examples of output protection function operation when GTCCRA register ≥ GTPR register is set during buffer transfer at troughs.



(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Output holding period: When GTCCRA ≥ GTPR is detected at the trough, the output protection status changes to GTSOS[1:0] = 10b, and the output is held while GTCCRA ≥ GTPR.
(3) Normal operation period: When 0 < GTCCRA < GTPR is detected at the trough, the output protection status changes to GTSOS[1:0] = 00b, ant toggle output due to a compare match is performed normally.

Figure 19.90 Example of output protection function operation when GTCCRA register ≥ GTPR register is set during buffer transfer at troughs (restored to 0 < GTCCRA register < GTPR register during buffer transfer at crests, active-level: low) (n = 0 to 17)

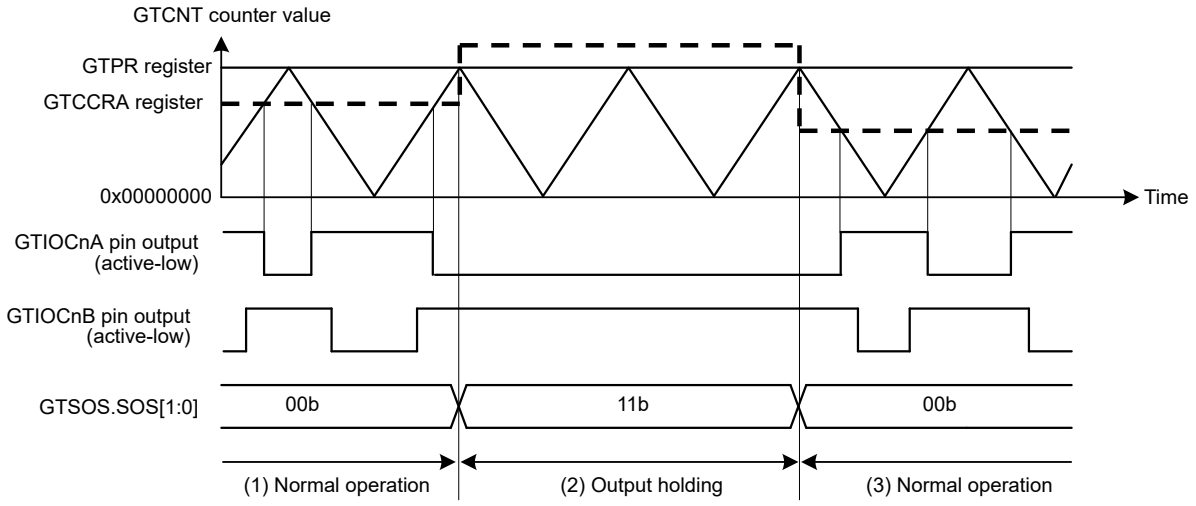


(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Output holding period: When GTCCRA ≥ GTPR is detected at the trough, the output protection status changes to GTSOS[1:0] = 10b, and the output is held while GTCCRA ≥ GTPR.
(3) Recovery period: When 0 < GTCCRA < GTPR is detected at the crest, the output protection status changes to GTSOS.SOS[1:0] = 00b, and holding output is maintained up to the next trough.
(4) Normal operation period: When 0 < GTCCRA register < GTPR register is detected in the crest, the output protection status changes to GTSOS[1:0] = 00b, and toggle output due to a compare match is performed normally.

Figure 19.91 Example of output protection function operation when GTCCRA register ≥ GTPR register is set during buffer transfer at troughs (restored to 0 < GTCCRA register < GTPR register during buffer transfer at troughs, active-level: low) (n = 0 to 17)

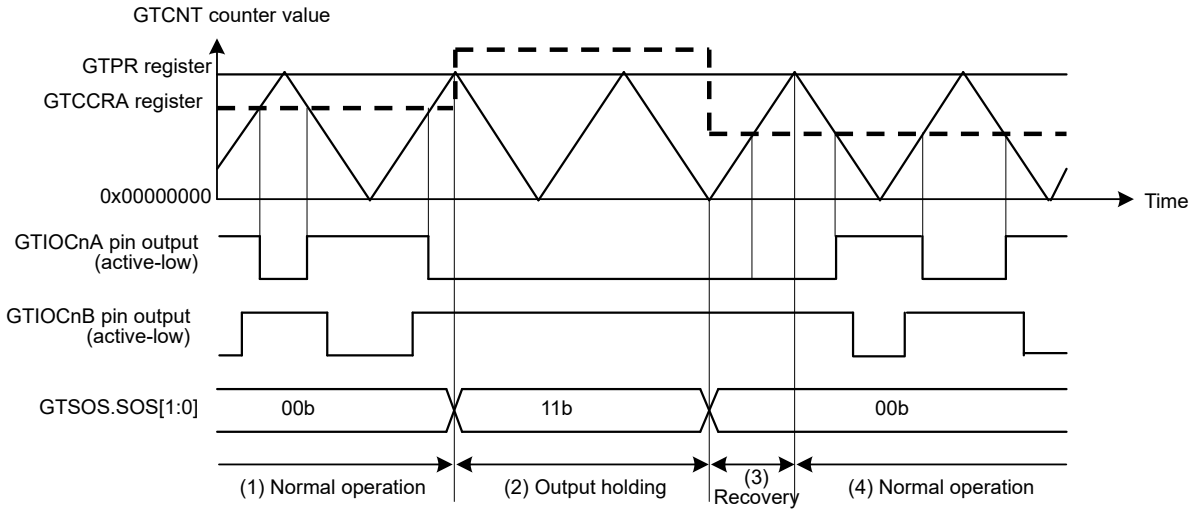
(3) Output Protection Function when GTCCRA Register ≥ GTPR Register is Set during Buffer Transfer at Crests

Figure 19.92 and Figure 19.93 show examples of output protection function operation when GTCCRA register ≥ GTPR register is set during buffer transfer at crests.



(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Output holding period: When GTCCRA ≥ GTPR is detected at the crest, the output protection status changes to GTSOS[1:0] = 11b, and the output is held while GTCCRA ≥ GTPR.
(3) Normal operation period: Toggle output due to a compare match is performed normally.

Figure 19.92 Example of output protection function operation when GTCCRA register ≥ GTPR register is set during buffer transfer at crests (restored to 0 < GTCCRA register < GTPR register during buffer transfer at crests, active-level: low) (n = 0 to 17)



(1) Normal operation period: Toggle output due to a compare match is performed normally.
(2) Output holding period: When GTCCRA ≥ GTPR Is detected at the crest, the output protection status changes to GTSOS[1:0] = 11b, and the output is held while GTCCRA ≥ GTPR.
(3) Recovery period: When 0 < GTCCRA < GTPR is detected at the trough, the output protection status changes to GTSOS.SOS[1:0] = 00b, and holding output is maintained up to the next crest.
(4) Normal operation period: Toggle output due to a compare match is performed normally.

Figure 19.93 Example of output protection function operation when GTCCRA register ≥ GTPR register is set during buffer transfer at crests (restored to 0 < GTCCRA register < GTPR register during buffer transfer at troughs, active-level: low) (n = 0 to 17)

(4) Restricted Specification of Output Protection Function

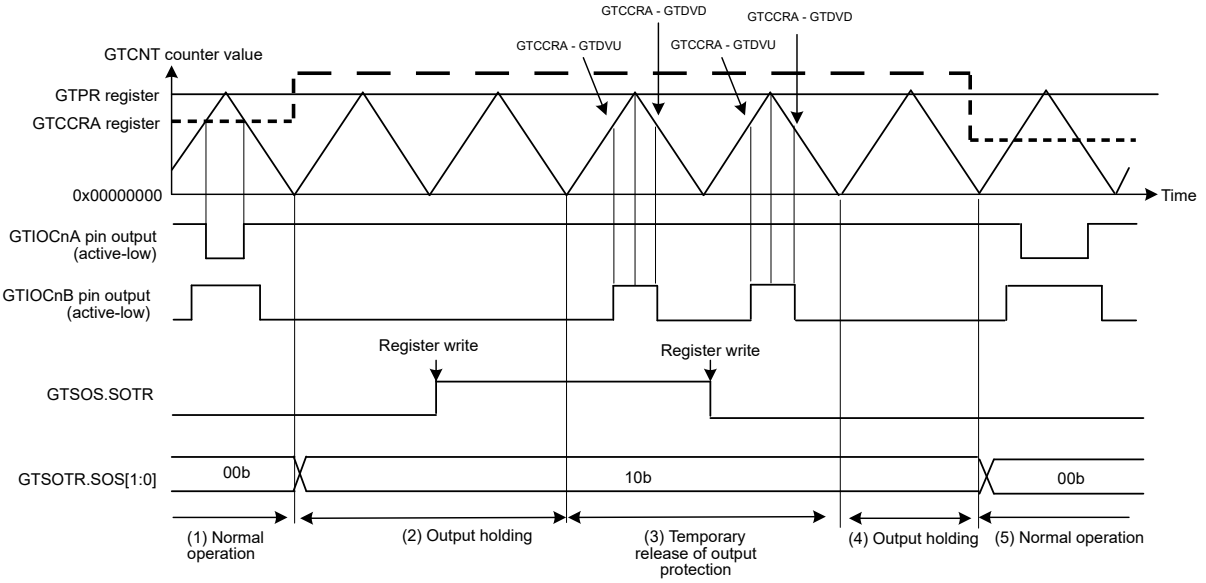
Even if an incorrect value (0x00000000 or a value greater than or equal to the GTPR register value) is set in the GTCCRA register during count operation, the output protection functions in a specific way such that one of the positive- and negative-phase outputs becomes non-active. However, the output protection does not operate normally when the GTCCRA register value at the start of count operation is greater than 0x00000000 and less than the setting value of the GTPR register.

(5) Temporary Release of Output Protection Function

When the GTSOS.SOS[1:0] bits = 10b (protected state in which GTCCRA register ≥ GTPR register occurred during transfer at trough), the protected state of the GTIOCnB pin output can be temporarily released by setting the GTSOTR.SOTR bit to 1. The SOS[1:0] bits retain 10b even if the output protection function is released.

When the SOTR bit is set to 0, the GTIOCnB pin output protection can be restarted.

Figure 19.94 shows an example of the operation of temporary release of output protection when the setting of the GTCCRA register ≥ GTPR register during buffer transfer at troughs.



(1) Normal operation period: Toggle output due to compare match is performed normally.

(2) Output holding period: When GTCCRA register ≥ GTPR register is detected in the trough, the output protection status changes to GTSOS[1:0] = 10b, and the output is held while GTCCRA register ≥ GTPR register. Even when the GTSOTR.SOTR bit is set to 1 during the output holding period, the GTIOCnB pin output is held until the first trough after the setting.

(3) Temporary release of output protection period: After GTSOTR.SOTR bit is set to 1, the GTIOCnB pin is performed normally from the first trough after the setting. Even when the GTSOTR.SOTR bit is set to 0 during the output protection releasing period, the GTIOCnB pin output is held until the first trough after the setting.

(4) Output holding period: The output is held until 0 < GTCCRA register < GTPR register is detected in the trough.

(5) Normal operation period: When 0 < GTCCRA register < GTPR register is detected in the trough, the output protection status changes to GTSOS[1:0] = 00b, and toggle output due to a compare match is performed normally.

Figure 19.94 Example of the operation of temporary release of output protection when the setting of the GTCCRA register ≥ GTPR register during buffer transfer at troughs (restored to 0 < GTCCRA register < GTPR register during buffer transfer at troughs, active-level: low) (n = 0 to 17)

19.9 Initialization Method of Output Pins

19.9.1 Pin Settings after Reset

The GPT registers are initialized at a reset. Initialize the GPT for output to the external pins by making the port mode settings and setting the OAE and OBE bits in the GTIOR register, and then start the counter counting.

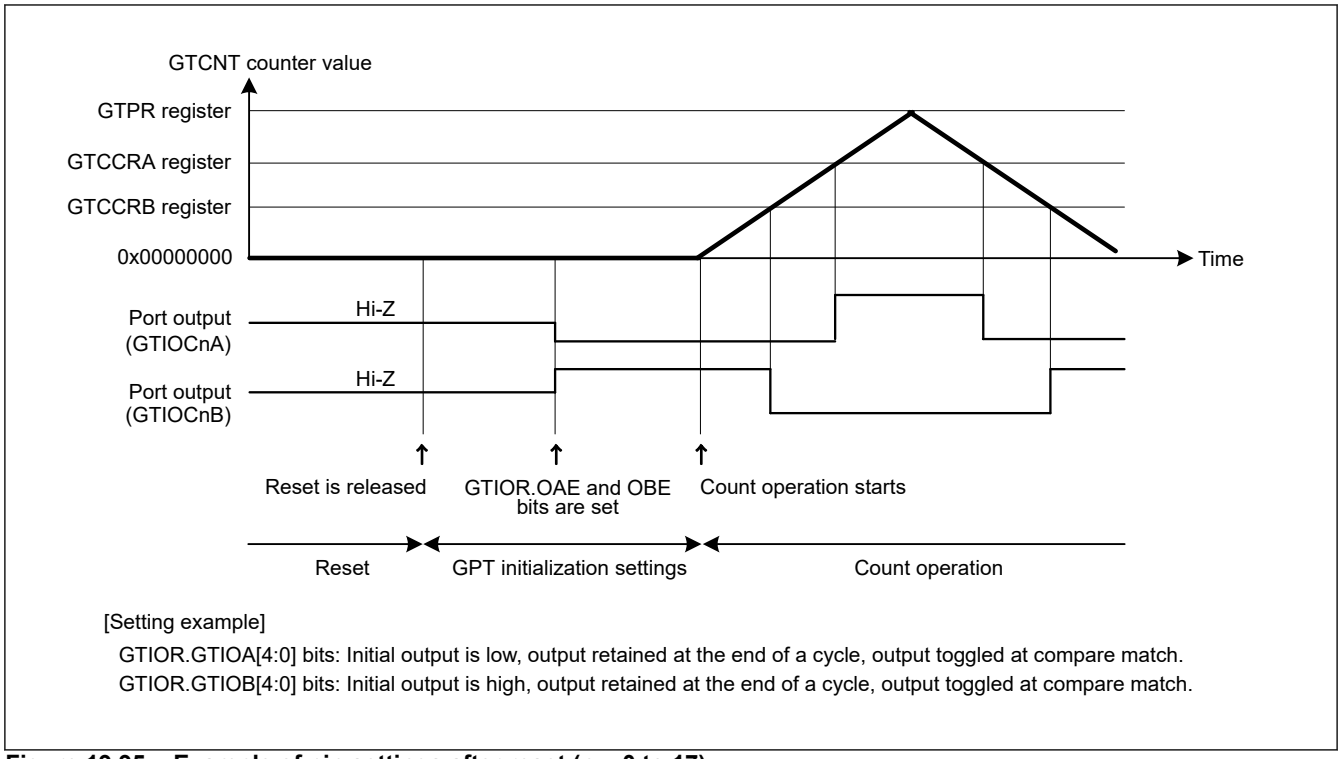


Figure 19.95 Example of pin settings after reset (n = 0 to 17)

19.9.2 Pin Initialization Due to Error during Operation

If an error occurs during GPT operation, the following four types of pin processing can be performed before pin initialization.

1. Set OAHLD and OBHLD bits in the GTIOR register to 1 and retain the outputs at count stop.
2. Set OAHLD and OBHLD bits to 0, specify arbitrary output values at OADFLT and OBDFLT bits in the GTIOR register, and output the arbitrary values at count stop.
3. Set the pin to output an arbitrary value as a general output port by setting the registers of the I/O port in advance. Set the OAE and OBE bits in the GTIOR register to 0 and the control bit in register that corresponds to the pin to allow the arbitrary values to be output from the pin set as a general output port when an error occurs.
4. Drive the output to a high impedance state using the POEG function.

When automatic dead time setting is made, set the GTDTCR.TDE bit to 0 once after counting is stopped. When counting is stopped, only the values of registers that are changed by a GPT external source change. If counting is resumed, operation carries on from where it was stopped.

If counting was stopped, registers should be initialized before counting is started.

19.10 Usage Notes

19.10.1 Module-Stop Function Setting

The Module Stop Control register can enable or disable operation of the GPT. The initial setting is to halt operation of the GPT. Register access is enabled by clearing the module-stop state. For details, see section 8, Low-Power Consumption Function.

19.10.2 Settings of the GTCCRn Register During Compare Match Operation (n = A to F)

(1) When Automatic Dead Time Setting is Made in Triangle-Wave PWM Mode

The GTCCRA register must satisfy the following conditions:

- GTCCRA register > GTDVU register

- GTCCRA register > GTDVD register
- GTCCRA register < GTPR register

When the GTCCRA register is set to 0x00000000 or a value greater than or equal to the GTPR register value during count operation, the output protection function is activated.

However, if the following conditions are not satisfied, the output protection does not operate normally.

- When the GTCCRA register value at the start of count operation is 0x00000001 or greater, and less than the setting value of the GTPR register

For details, see section 19.8.4. GTIOCnm Protection Function for GTIOCnm Pin Output (n = 0 to 17, m = A, B).

(2) When Automatic Dead Time Setting is not Made in Triangle-Wave PWM Mode

Set a value greater than 0x00000000 and less than the setting value of the GTPR register in the GTCCRA register. When 0x00000000 or the same value as that of the GTPR register is set in the GTCCRA register, compare match is generated in one cycle only when [GTCCRA register = 0x00000000] or [GTCCRA register = GTPR register] is met. Furthermore, a value exceeding the setting value of the GTPR register is set in the GTCCRA register, compare match does not occur.

Similarly, set a value greater than 0x00000000, and less than the setting value of the GTPR register in the GTCCRB register. When 0x00000000 or the same value as that of the GTPR register is set in the GTCCRB register, compare match is generated in one cycle only when [GTCCRB register = 0x00000000] or [GTCCRB register = GTPR register] is met.

Furthermore, a value exceeding the setting value of the GTPR register is set in the GTCCRB register, compare match does not occur.

(3) When Automatic Dead Time Setting is Made in Sawtooth-Wave One-Shot Pulse Mode

The GTCCRC and GTCCRD registers must be set to satisfy the following restrictions. If the restrictions are not satisfied, correct output waveforms with secured dead time may not be obtained.

- In up-counting:

  GTCCRC register < GTCCRD register

  GTCCRC register > GTDVU register

  GTCCRD register < GTPR register − GTDVD register
- In down-counting:

  GTCCRC register > GTCCRD register

  GTCCRC register < GTPR register − GTDVU register

  GTCCRD register > GTDVD register

(4) When Automatic Dead Time Setting is not Made in Sawtooth-Wave One-Shot Pulse Mode

The GTCCRC and GTCCRD registers must be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRC register < GTCCRD register < GTPR register
- In down-counting: GTPR register > GTCCRC register > GTCCRD register > 0

Similarly, GTCCRE and GTCCRF registers must be set to satisfy the following restrictions. If the restrictions are not satisfied, two compare matches do not occur and pulse output cannot be performed.

- In up-counting: 0 < GTCCRE register < GTCCRF register < GTPR register
- In down-counting: GTPR register > GTCCRE register > GTCCRF register > 0

(5) In Sawtooth-Wave PWM Mode

The GTCCRA register must be set with the range of 0x00000000 < GTCCRA register < GTPR register. If GTCCRA register = 0x00000000 or GTCCRA register = GTPR register is set, a compare match occurs within the cycle only when GTCCRA register = 0x00000000 or GTCCRA register = GTPR register is satisfied. If GTCCRA register > GTPR register is set, no compare match occurs.

Similarly, GTCCRB register must be set with the range of 0x00000000 < GTCCRB register < GTPR register. If GTCCRB register = 0x00000000 or GTCCRB register = GTPR register is set, a compare match occurs within the cycle only when GTCCRB register = 0x00000000 or GTCCRB register = GTPR register is satisfied. If GTCCRB register > GTPR register is set, no compare match occurs.

19.10.3 Setting Range of the GTCNT Counter

Set the range of the GTCNT counter within the range of 0 ≤ GTCNT counter ≤ GTPR register.

19.10.4 The GTCNT Counter Start/Stop

The GTCNT counter start/stop control by the GTCR.CST bit is synchronized with the count clock selected by the GTCR.TPCS[3:0] bits. Since the GTCNT counter starts or stops after one count clock cycle selected by the GTCR.TPCS[3:0] bits following the CST bit updating, events until the GTCNT counter actually starts are ignored, where events may be accepted and the interrupt may be generated after the CST bit becomes 0.

19.10.5 Order of Priority in Events

(1) GTCNT Counter

Order of priority in events to update the GTCNT counter is shown below.

Table 19.25 Order of priority in updating GTCNT counter

The GTCNT counter updating source	Order of priority
CPU writing (GTCNT counter writing/GTCLR register writing)	High
Clearing by a hardware source set by the GTCSR register	↑
Up-counting/down-counting by a hardware source set by the GTUPSR and GTDNSR registers	↑
Count operation	Low

In case of contention between up-counting by the GTUPSR register and down-counting by the GTDNSR register, the counter value is not updated.

When updating of the GTCNT counter conflicts with reading by the CPU, the value before updating is reflected.

(2) GTCR.CST Bit

When start/stop by a hardware source set by the GTSSR and GTPSR registers conflicts with the CPU writing (GTCR register writing/GTSTR register writing/GTSTP register writing), the CPU writing takes priority.

When there is contention between start by a hardware source set by the GTSSR register and stop by a hardware source set by the GTPSR register, the state of the CST bit does not change.

When there is contention between the CST bit updating and the CPU reading (GTCR register reading/GTSTR register reading/GTSTP register reading), data before updating is read.

(3) GTCCRm Register (m = A to F)

When there is contention between the GTCCRm register writing and an input capture/buffer transfer, the GTCCRm register writing takes priority over an input capture/buffer transfer.

When an input capture conflicts with the CPU writing or the counter updating by a hardware source, the counter value before updating is captured.

When there is contention between the GTCCRm register updating and the CPU reading, data before updating is read.

(4) GTPR Register

When there is contention between a buffer transfer and the GTPR register writing, the GTPR register writing takes priority over a buffer transfer.

When there is contention between the GTPR register updating and the CPU reading, data before updating is read.

(5) GTADTRm Register (m = A, B)

When there is contention between a buffer transfer and the GTADTRm register writing, the GTADTRm register writing takes priority over a buffer transfer.

When there is contention between the GTADTRm register updating and the CPU reading, data before updating is read.

(6) GTDVm Register (m = U, D)

When there is contention between a buffer transfer and the GTDVm register writing, the GTDVm register writing takes priority over a buffer transfer.

When there is contention between the GTDVm register updating and the CPU reading, data before updating is read.

20. Port Output Enable for GPT (POEG)

20.1 Overview

The POEG issues requests to stop output from output pins of the general PWM timer (GPT). The combination of output pins of the POEG to be disabled can be specified from any channel of the GPT.

Select the method of detection for stopping the output from the list below.

- Input level detection of the GTETRGn pin (n = A to D)
- Detection from the GPT to stop output
- Detection of stopping of oscillation by the oscillation stop detection circuit for the main clock
- Register setting

The GTETRGn pin can be used for output to the external trigger input pins of the GPT.

Table 20.1 lists POEG specifications, Figure 20.1 shows a block diagram of POEG, Figure 20.4 shows POEG external trigger connections, Table 20.2 lists POEG I/O pins, and Table 20.3 lists POEG Interrupt Sources.

Table 20.1 POEG specifications

Item	Description
Number of units	Three units (LLPP (POEG0), NONSAFETY (POEG1), SAFETY (POEG2))
Number of groups	Four groups (A to D)
Output-disable control through input level detection	The GPT output pins can be disabled when a GTETRGA to GTETRGD rising edge or high level is sampled after polarity and filter selection.
Output-disable request from the GPT	• When the GTIOCA pin and the GTIOCB pin are driven to an active level simultaneously, the GPT generates an output-disable request to the POEG. Through reception of these requests, the POEG can control whether the GTIOCA and GTIOCB pins are output-disabled. • GPT output pins can be set to be disabled when the GPT output pins detect a dead time error.
Output-disable control through oscillation stop detection	The GPT output pins can be disabled when oscillation of the clock generation circuit stops.
Output-disable control by software (registers)	The GPT output pins can be disabled by modifying the register settings.
Output-disable control through DSMIF error detection	The GPT output pins can be disabled when DSMIF error 0 or 1 occurs (LLPP only).
Interrupt	• Allows output-disable control by the input level detection • Allows output-disable requests from the GPT
External trigger output to the GPT (count start, count stop, count clear, up-count, down-count, or input capture function)	The GTETRGA to GTETRGD signals can be output to the GPT after polarity and filter selection.
Noise filtering	• Three times sampling for every PCLK/1, PCLK/8, PCLK/32, or PCLK/128 can be set for any of the input pins GTETRGA to GTETRGD. (PCLK is PCLKH for LLPP and PCLKL for NONSAFETY and SAFETY). • Positive or negative polarity can be selected for any of the input pins, GTETRGA to GTETRGD. • Signal state after polarity and filter selection can be monitored.

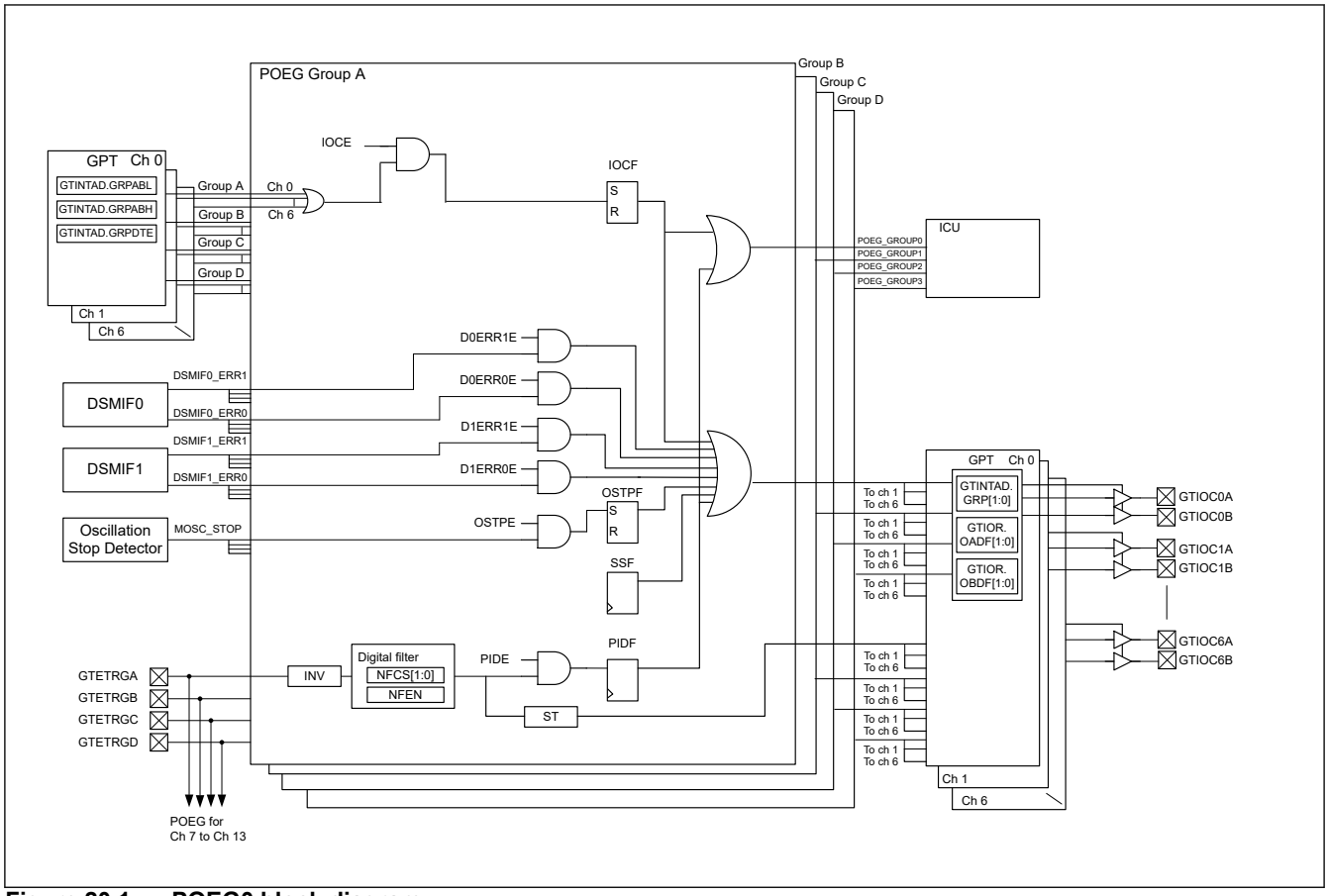


Figure 20.1 POEG0 block diagram

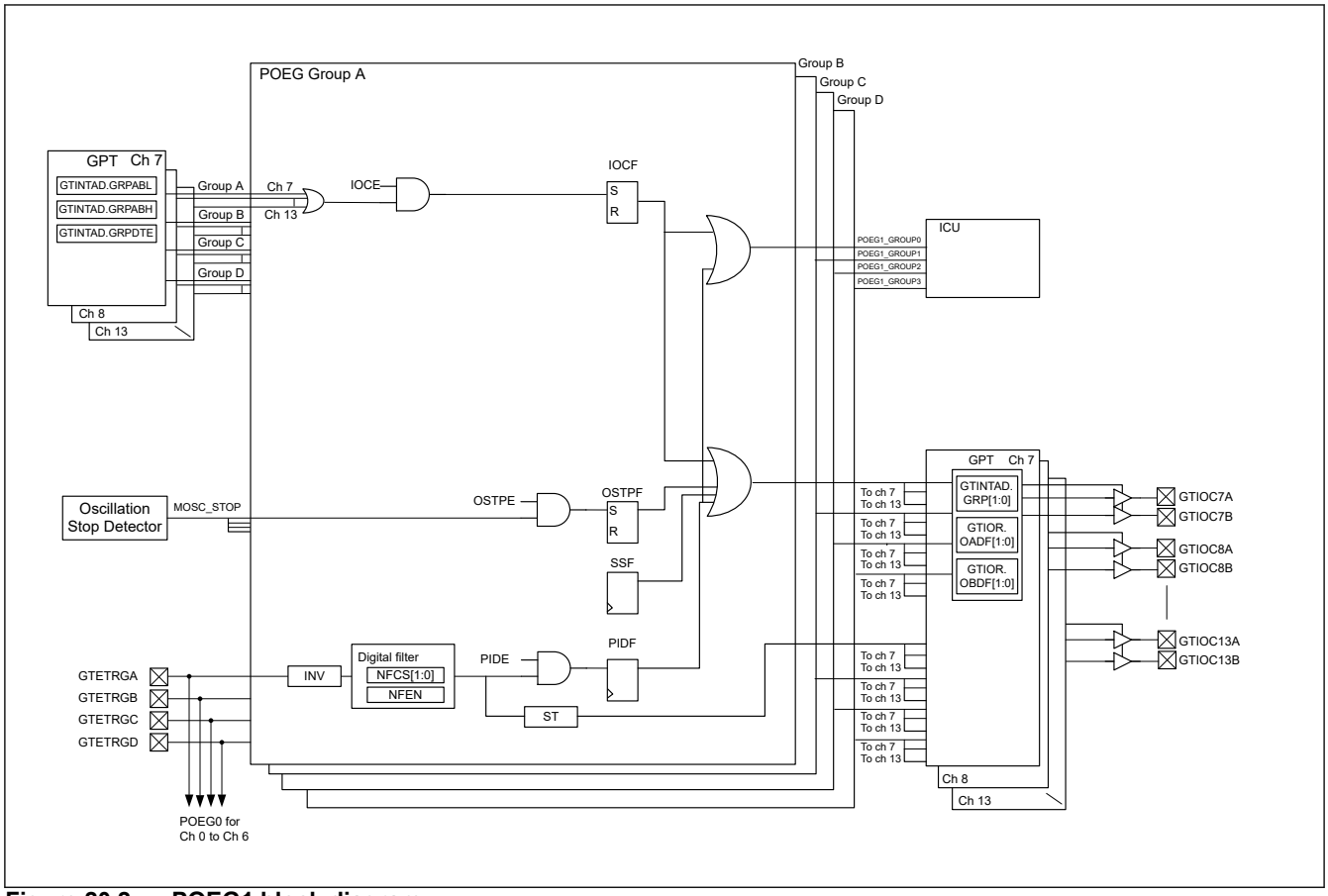


Figure 20.2 POEG1 block diagram

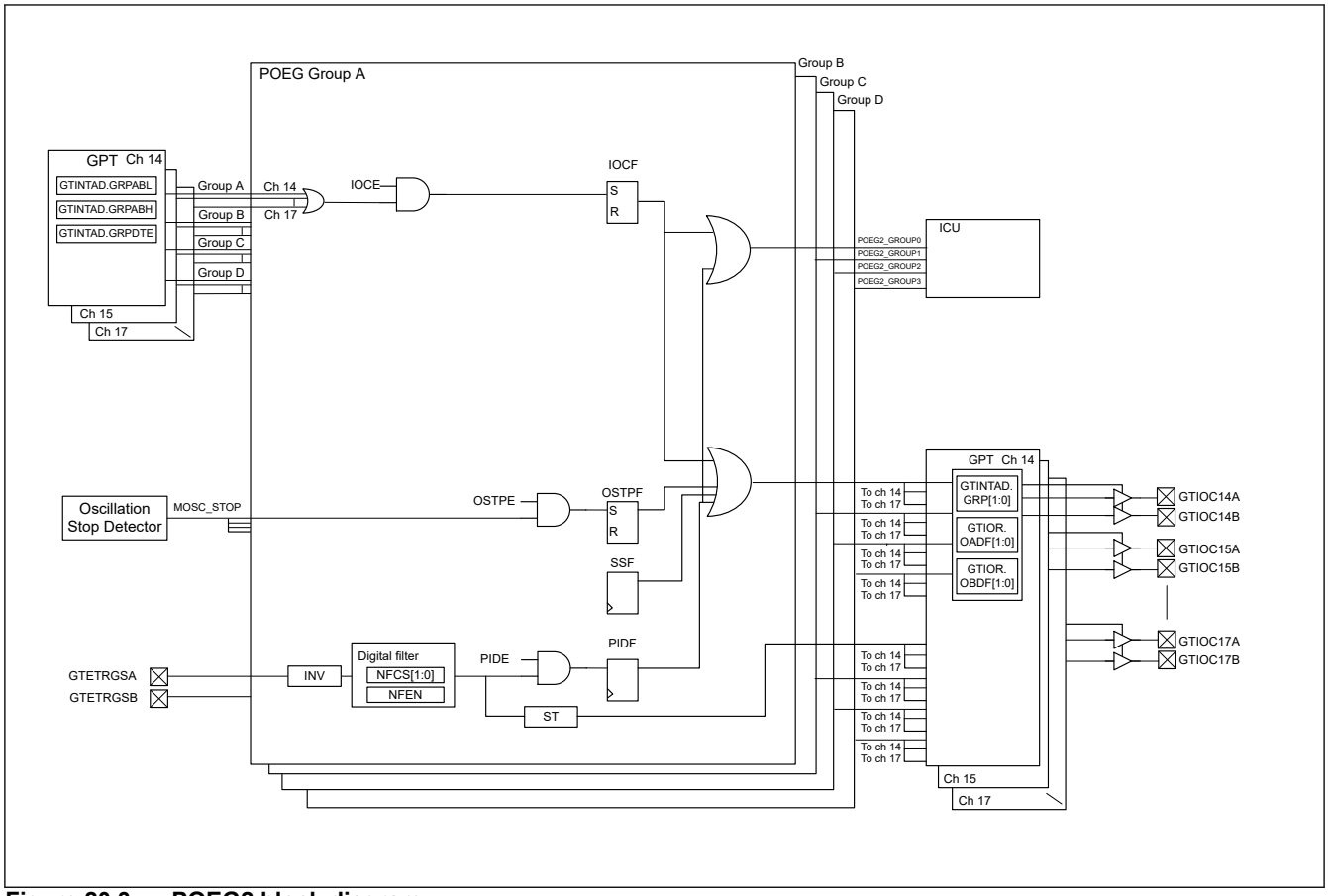


Figure 20.3 POEG2 block diagram

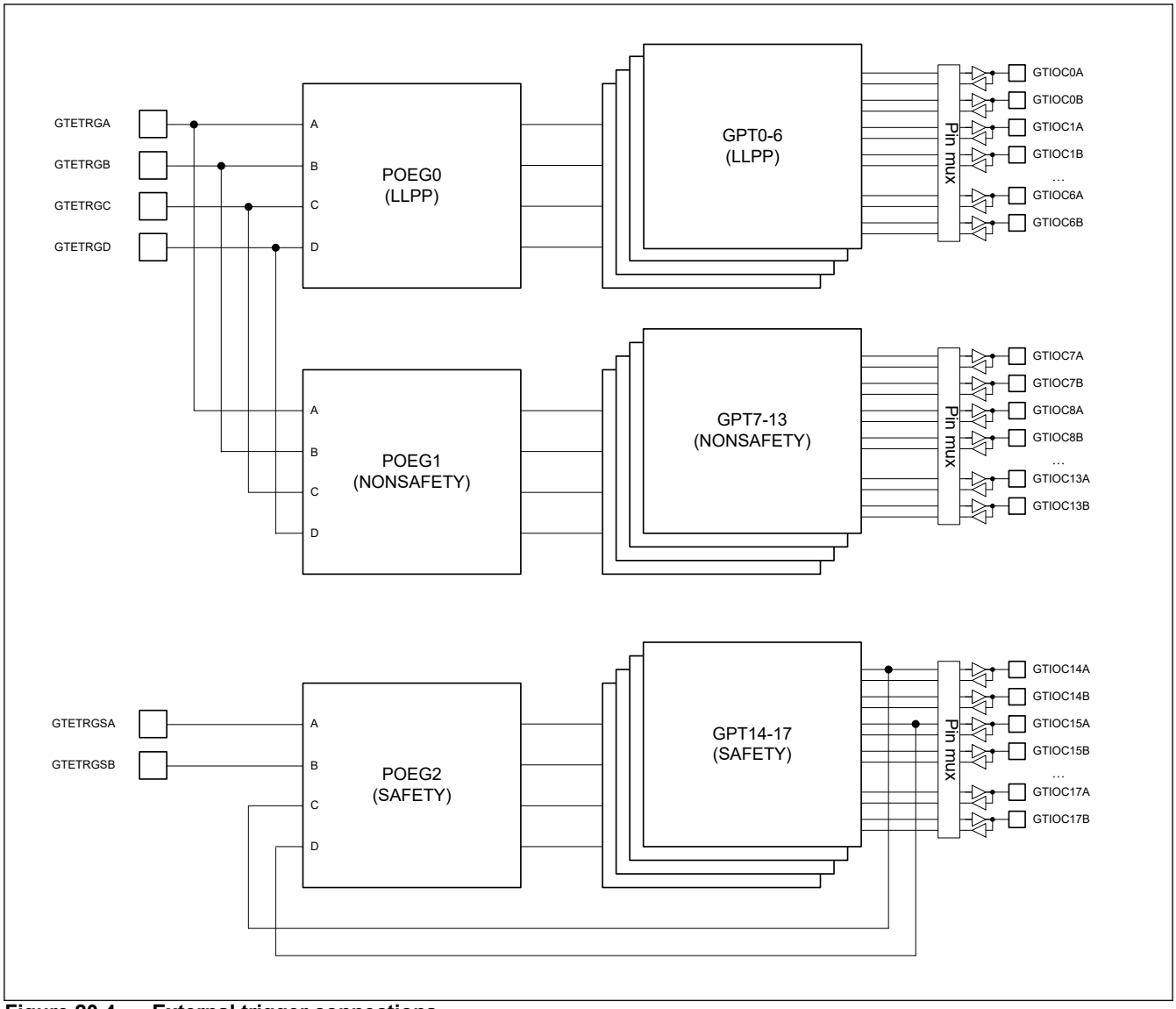


Figure 20.4 External trigger connections

Table 20.2 POEG I/O pins

Pin name	I/O	Function
GTETRGA	Input	GPT output pin output-disable request signal and GPT external trigger input pin A for channels in LLPP and NONSAFETY
GTETRGB	Input	GPT output pin output-disable request signal and GPT external trigger input pin B for channels in LLPP and NONSAFETY
GTETRGC	Input	GPT output pin output-disable request signal and GPT external trigger input pin C for channels in LLPP and NONSAFETY
GTETRGD	Input	GPT output pin output-disable request signal and GPT external trigger input pin D for channels in LLPP and NONSAFETY
GTETRGSA	Input	GPT output pin output-disable request signal and GPT external trigger input pin A for channels in SAFETY
GTETRGSB	Input	GPT output pin output-disable request signal and GPT external trigger input pin B for channels in SAFETY

Table 20.3 POEG interrupt sources (1 of 2)

Name	Interrupt Sources	CPU0 GIC Request	DMAC Activation
POEG0_GROUP0	POEG group A interrupt for channels in LLPP	Possible	Not possible

Table 20.3 POEG interrupt sources (2 of 2)

Name	Interrupt Sources	CPU0 GIC Request	DMAC Activation
POEG0_GROUP1	POEG group B interrupt for channels in LLPP	Possible	Not possible
POEG0_GROUP2	POEG group C interrupt for channels in LLPP	Possible	Not possible
POEG0_GROUP3	POEG group D interrupt for channels in LLPP	Possible	Not possible
POEG1_GROUP0	POEG group A interrupt for channels in NONSAFETY	Possible	Not possible
POEG1_GROUP1	POEG group B interrupt for channels in NONSAFETY	Possible	Not possible
POEG1_GROUP2	POEG group C interrupt for channels in NONSAFETY	Possible	Not possible
POEG1_GROUP3	POEG group D interrupt for channels in NONSAFETY	Possible	Not possible
POEG2_GROUP0	POEG group A interrupt for channels in SAFETY	Possible	Not possible
POEG2_GROUP1	POEG group B interrupt for channels in SAFETY	Possible	Not possible
POEG2_GROUP2	POEG group C interrupt for channels in SAFETY	Possible	Not possible
POEG2_GROUP3	POEG group D interrupt for channels in SAFETY	Possible	Not possible

20.2 Register Descriptions

20.2.1 POEG0Gn : POEG0 Group n Setting Register (n = A to D)

Base address: POEG0 = 0x9000_6000

Offset address: 0x000 (POEG0GA)
0x400 (POEG0GB)
0x800 (POEG0GC)
0xC00 (POEG0GD)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFCS[1:0]		NFEN	INV	D1ERR0E	D0ERR0E	D1ERR0ST	D0ERR0ST	D1ERR1E	D0ERR1E	D1ERR1ST	D0ERR1ST	—	—	—	ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	OSTPE	IOCE	PIDE	SSF	OSTPF	IOCF	PIDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIDF	Port Input Detection Flag 0: The selected input level was not detected on the GTETRGn pin 1: The selected input level was detected on the GTETRGn pin	R/W*1
1	IOCF	GPT Output Stop Request Detection Flag 0: Stopping of GPT output was not detected 1: Stopping of GPT output was detected	R/W*1
2	OSTPF	Oscillation Stop Detection Flag 0: Stopping of oscillation was not detected 1: Stopping of oscillation was detected	R/W*1
3	SSF	Software Stop Flag 0: Software has not stopped output 1: Software has stopped output	R/W
4	PIDE	Port Input Detection Enable 0: Detection of input levels on the corresponding GTETRGn pin is disabled 1: Detection of input levels on the corresponding GTETRGn pin is enabled	R/W*2
5	IOCE	GPT Output Stop Request Enable 0: Detection of stopping of output from the GPT is disabled 1: Detection of stopping of output from the GPT is enabled	R/W*2

Bit	Symbol	Function	R/W
6	OSTPE	Enable Stopping Output on Stopping of Oscillation 0: Detection of stopping of oscillation is disabled 1: Detection of stopping of oscillation is enabled	R/W*2
15:7	—	These bits are read as 0. The write value should be 0.	R/W
16	ST	GTETRGn Input Status Flag 0: GTETRGn input after filtering was 0 1: GTETRGn input after filtering was 1	R
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	D0ERR1ST	DSMIF0 error 1 status 0: Not error detect 1: Error detect	R
21	D1ERR1ST	DSMIF1 error 1 status 0: Not error detect 1: Error detect	R
22	D0ERR1E	Permit output disabled by DSMIF0 error 1 detection 0: Disable 1: Not disabled	R/W*2
23	D1ERR1E	Permit output disabled by DSMIF1 error 1 detection 0: Disable 1: Not disabled	R/W*2
24	D0ERR0ST	DSMIF0 error 0 status 0: Not error detect 1: Error detect	R
25	D1ERR0ST	DSMIF1 error 0 status 0: Not error detect 1: Error detect	R
26	D0ERR0E	Permit output disabled by DSMIF0 error 0 detection 0: Disable 1: Not disabled	R/W*2
27	D1ERR0E	Permit output disabled by DSMIF1 error 0 detection 0: Disable 1: Not disabled	R/W*2
28	INV	GTETRGn Input Inverting 0: Input on the GTETRGn pin is not inverted 1: Input on the GTETRGn pin is inverted	R/W
29	NFEN	Noise filter Enable 0: Digital noise filter on the GTETRGn pin is disabled 1: Digital noise filter on the GTETRGn pin is enabled	R/W
31:30	NFCS[1:0]	Noise filter Clock Select 0 0: Samples the input level of GTETRGn pin three times per PCLKH clock 0 1: Samples the input level of GTETRGn pin three times per PCLKH/8 clock 1 0: Samples the input level of GTETRGn pin three times per PCLKH/32 clock 1 1: Samples the input level of GTETRGn pin three times per PCLKH/128 clock	R/W

Note 1. Only 0 to clear the flag can be written.
Note 2. Can write only once after reset.

The POEG0Gn register (n = A to D) controls requests for stopping output and an external trigger for the GPT based in response to the detection of various signals.

20.2.2 POEGmGn : POEGm Group n Setting Register (m = 1, 2) (n = A to D)

Base address: POEG1 = 0x8004_7000 (m = 1)
POEG2 = 0x8100_A000 (m = 2)

Offset address: 0x000 (POEGmGA)
0x400 (POEGmGB)
0x800 (POEGmGC)
0xC00 (POEGmGD)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	NFCS[1:0]		NFEN	INV	—	—	—	—	—	—	—	—	—	—	—	ST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	OSTPE	IOCE	PIDE	SSF	OSTPF	IOCF	PIDF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PIDF	Port Input Detection Flag 0: The selected input level was not detected on the GTETRGn pin 1: The selected input level was detected on the GTETRGn pin	R/W*1
1	IOCF	GPT Output Stop Request Detection Flag 0: Stopping of GPT output was not detected 1: Stopping of GPT output was detected	R/W*1
2	OSTPF	Oscillation Stop Detection Flag 0: Stopping of oscillation was not detected 1: Stopping of oscillation was detected	R/W*1
3	SSF	Software Stop Flag 0: Software has not stopped output 1: Software has stopped output	R/W
4	PIDE	Port Input Detection Enable 0: Detection of input levels on the corresponding GTETRGn pin is disabled 1: Detection of input levels on the corresponding GTETRGn pin is enabled	R/W
5	IOCE	GPT Output Stop Request Enable 0: Detection of stopping of output from the GPT is disabled 1: Detection of stopping of output from the GPT is enabled	R/W
6	OSTPE	Enable Stopping Output on Stopping of Oscillation 0: Detection of stopping of oscillation is disabled 1: Detection of stopping of oscillation is enabled	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W
16	ST	GTETRGn Input Status Flag 0: GTETRGn input after filtering was 0 1: GTETRGn input after filtering was 1	R
27:17	—	These bits are read as 0. The write value should be 0.	R/W
28	INV	GTETRGn Input Inverting 0: Input on the GTETRGn pin is not inverted 1: Input on the GTETRGn pin is inverted	R/W
29	NFEN	Noise filter Enable 0: Digital noise filter on the GTETRGn pin is disabled 1: Digital noise filter on the GTETRGn pin is enabled	R/W
31:30	NFCS[1:0]	Noise filter Clock Select 0 0: Samples the input level of GTETRGn pin three times per PCLKL clock 0 1: Samples the input level of GTETRGn pin three times per PCLKL/8 clock 1 0: Samples the input level of GTETRGn pin three times per PCLKL/32 clock 1 1: Samples the input level of GTETRGn pin three times per PCLKL/128 clock	R/W

Note 1. Only 0 to clear the flag can be written.

The POEGmGn register (m = 1, 2), (n = A to D) controls requests for stopping output and an external trigger for the GPT based in response to the detection of various signals.

20.3 Operation

20.3.1 Output-Disable Control Operation

If any of the following conditions is satisfied, the GTIOCxA, GTIOCxB, and the three-phase PWM output for motor control pins can be set to output-disable:

- Input level or edge detection of the GTETRGA to GTETRGD pins
  When POEGmGn.PIDE = 1 (m = 0 to 2, n = A to D), the POEGmGn.PIDF flag is set to 1.
- Output-disable request from the GPT
  When POEGmGn.IOCE = 1 (m = 0 to 2, n = A to D), the POEGmGn.IOCF flag is set to 1 if the output-disable request is enabled in GTINTAD. The GTINTAD.GRPDTE, GTINTAD.GRPABH, and GTINTAD.GRPABL settings apply to the group selected in GPT register GTINTAD.GRP[1:0] and OPSCR.GRP[1:0].
- Oscillation stop detection for the Clock Generation Circuit
  When POEGmGn.OSTPE = 1 (m = 0 to 2, n = A to D), the POEGmGn.OSTPF flag is set to 1.
- SSF bit setting
  When POEGmGn.SSF (m = 0 to 2, n = A to D) is set to 1.

The state of output-disable is controlled in the GPT module. The output-disable of the GTIOCxA and GTIOCxB pins are set in the GTINTAD.GRP[1:0] bits, GTIOR.OADF[1:0] bits, and GTIOR.OBDF[1:0] bits in GPTx. The output-disable of the PWM output for motor control pins are set in the OPSCR.GRP[1:0] bits and OPSCR.GODF bit in GPT_OPS.

20.3.1.1 Pin Input Level Detection Operation

If the input conditions set in POEGmGn.PIDE, POEGmGn.NFCS[1:0], POEGmGn.NFEN, and POEGmGn.INV (m = 0 to 2, n = A to D) occur on the GTETRGA to GTETRGD pins, the GPT output pins are output-disabled.

(1) Digital Noise Filter

Figure 20.5 shows high-level detection by the digital filter. When a high level associated with the POEGmGn.INV (m = 0 to 2, n = A to D) polarity setting is detected three times consecutively with the sampling clock selected in POEGmGn.NFCS[1:0] and POEGmGn.NFEN, the detected level is recognized as high, and the GPT output pins are output-disabled. If even one low level is detected during this interval, the detected level is not recognized as high. In addition, in an interval where the sampling clock is not being output, changes of the levels on the GTETRGA to GTETRGD pins are ignored.

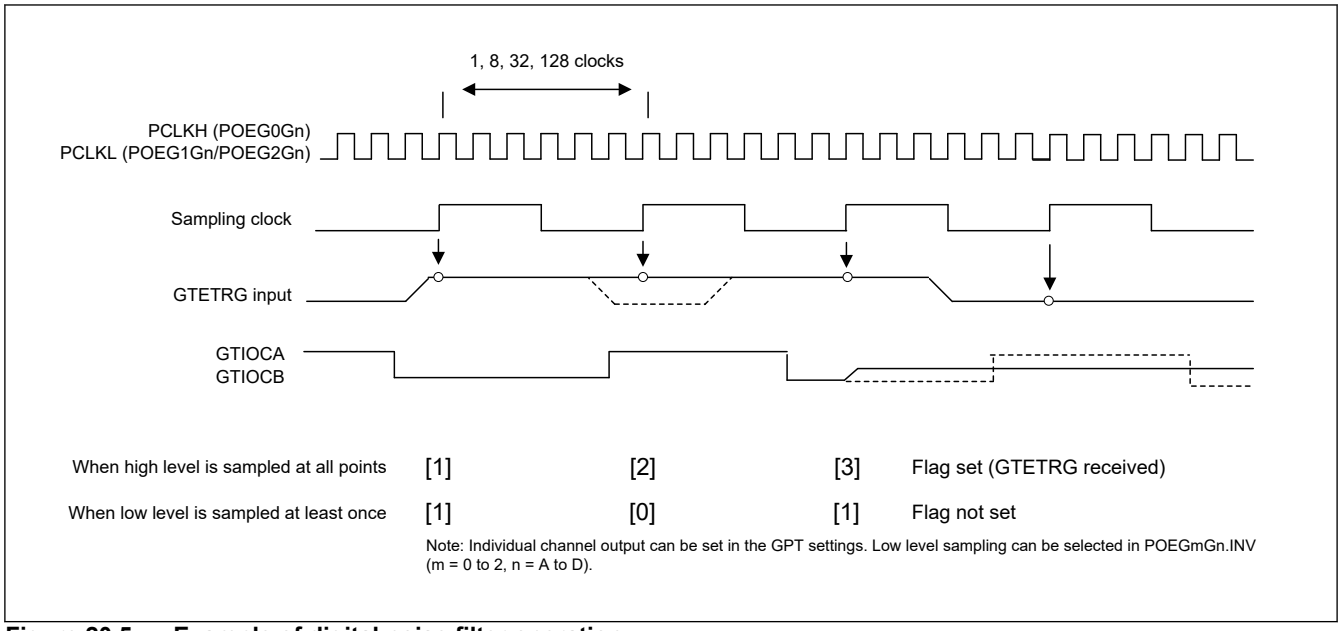


Figure 20.5 Example of digital noise filter operation

20.3.2 Output-Disable Requests from the GPT

For details on this operation, see the description of output-disable control for GTIOC pin outputs in section 19, General PWM Timer (GPT).

20.3.3 Output-Disable Control Using Detection of Stopped Oscillation

When the oscillation stop detection function in the Clock Generation Circuit detects stopped oscillation while POEGmGn.OSTPE = 1 (m = 0 to 2, n = A to D), the GPT output pins are output-disabled for each group.

20.3.4 Output-Disable Control Using DSMIF Error Detection

When the DSMIF detects error while POEG0Gn.DmERR0E =1 or POEG0Gn.DmERR1E = 1 (m = 0, 1), the GPT output pins in the LLPP region can be output-disabled.

20.3.5 Output-Disable Control Using Registers

The GPT output pins can be directly controlled by writing to the software stop flag, POEGmGn.SSF.

20.3.6 Release from Output-Disable

To release the GPT output pins placed in the output-disable state, either return them to their initial state with a reset or clear all of the following (n = A to D):

- POEG0Gn.PIDF flag
- POEG0Gn.IOCF flag
- POEG0Gn.OSTPF flag
- POEG0Gn.SSF flag

Writing 0 to the POEG0Gn.PIDF (n = A to D) flag is ignored (the flag is not cleared) if the external input pins, GTETRGA to GTETRGD, are not disabled and the POEG0Gn.ST bit is not set to 0.

Writing 0 to the POEG0Gn.IOCF (n = A to D) flag is valid (the flag is cleared) only if all of the GTST.DTEF, GTST.OABHF, and GTST.OABLF flags in the GPT are set to 0.

Figure 20.6 shows the release timing for output-disable. The output-disable is released at the beginning of the next count cycle of the GPT after the flag is cleared.

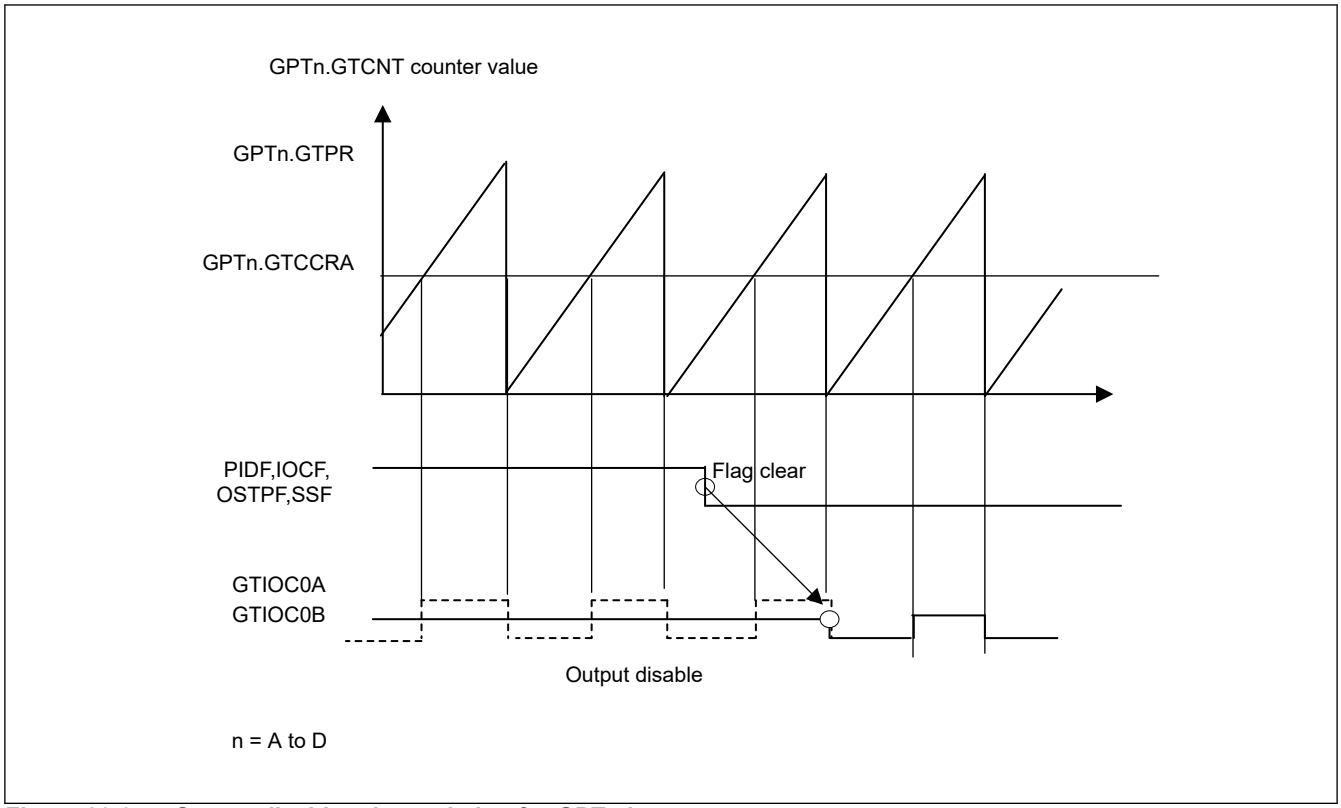


Figure 20.6 Output-disable release timing for GPT pin outputs

21. Trigonometric Function Unit (TFU)

21.1 Overview

A trigonometric Function Unit (TFU) handles the high-speed calculation for sinf, cosf, atan2f, and hypotf functions.

The PCLKH is used as the operating clock for the TFU.

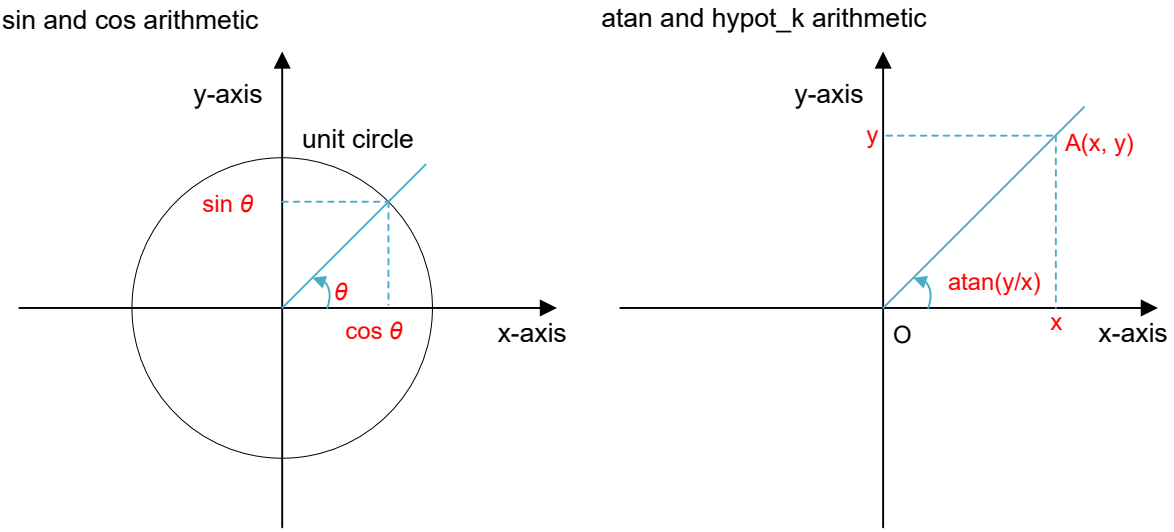
Table 21.1 lists the specifications of the TFU.

Table 21.1 TFU specifications

<table>
<tr><th>Item</th><th colspan="4">Description</th></tr>
<tr><td>Arithmetic Processing</td><td colspan="4">Calculation of sine, cosine, arctangent, and hypot_k ($\sqrt{x^2+y^2}/k$)
● A sine and cosine can be simultaneously calculated.
● An arctangent and hypot_k can be simultaneously calculated.</td></tr>
<tr><td>Range and Unit of Values</td><td colspan="4">
<table>
<tr><th>Floating-point</th><th colspan="2">I/O</th><th>Range</th><th>Unit</th></tr>
<tr><td rowspan="2">Calculating sine</td><td>Input</td><td>Angle θ</td><td>$-\text{float_max} \le \theta \le \text{float_max}$ *1</td><td>radian</td></tr>
<tr><td>Output</td><td>$\sin\theta$</td><td>$-1.0 \le \sin\theta \le 1.0$</td><td>—</td></tr>
<tr><td rowspan="2">Calculating cosine</td><td>Input</td><td>Angle θ</td><td>$-\text{float_max} \le \theta \le \text{float_max}$ *1</td><td>radian</td></tr>
<tr><td>Output</td><td>$\cos\theta$</td><td>$-1.0 \le \cos\theta \le 1.0$</td><td>—</td></tr>
<tr><td rowspan="2">Calculating arctangent</td><td>Input</td><td>x and y coordinates</td><td>$-\text{float_max} \le x \le \text{float_max}$ *1
$-\text{float_max} \le y \le \text{float_max}$ *1</td><td>—</td></tr>
<tr><td>Output</td><td>$\text{atan}(y/x)$</td><td>$-\pi \le \text{atan}(y/x) \le \pi$</td><td>radian</td></tr>
<tr><td rowspan="2">Calculating hypot_k</td><td>Input</td><td>x and y coordinates</td><td>$-\text{float_max} \le x \le \text{float_max}$ *1
$-\text{float_max} \le y \le \text{float_max}$ *1</td><td>—</td></tr>
<tr><td>Output</td><td>$\sqrt{x^2+y^2}/k$</td><td>$0 \le \sqrt{x^2+y^2}/k \le \infty$</td><td>—</td></tr>
<tr><th>Fixed-point</th><th colspan="2">I/O</th><th>Range</th><th>Unit</th></tr>
<tr><td rowspan="3">Calculating sine</td><td rowspan="2">Input</td><td rowspan="2">Angle θ</td><td>$-1.0 \le \theta < 1.0$</td><td>turn</td></tr>
<tr><td>$-4.0 \le \theta < 4.0$</td><td>radian</td></tr>
<tr><td>Output</td><td>$\sin\theta$</td><td>$-1.0 \le \sin\theta \le 1.0$</td><td>—</td></tr>
<tr><td rowspan="3">Calculating cosine</td><td rowspan="2">Input</td><td rowspan="2">Angle θ</td><td>$-1.0 \le \theta < 1.0$</td><td>turn</td></tr>
<tr><td>$-4.0 \le \theta < 4.0$</td><td>radian</td></tr>
<tr><td>Output</td><td>$\cos\theta$</td><td>$-1.0 \le \cos\theta \le 1.0$</td><td>—</td></tr>
<tr><td rowspan="3">Calculating arctangent</td><td>Input</td><td>x and y coordinates</td><td>$-1.0 \le x < 1.0$
$-1.0 \le y < 1.0$</td><td>—</td></tr>
<tr><td rowspan="2">Output</td><td rowspan="2">$\text{atan}(y/x)$</td><td>$-0.5 \le \text{atan}(y/x) \le 0.5$</td><td>turn</td></tr>
<tr><td>$-\pi \le \text{atan}(y/x) \le \pi$</td><td>radian</td></tr>
<tr><td rowspan="2">Calculating hypot_k</td><td>Input</td><td>x and y coordinates</td><td>$-1.0 \le x < 1.0$
$-1.0 \le y < 1.0$</td><td>—</td></tr>
<tr><td>Output</td><td>$\sqrt{x^2+y^2}/k$</td><td>$0 \le \sqrt{x^2+y^2}/k \le \sqrt{2}/k$</td><td>—</td></tr>
</table>
</td></tr>
<tr><td>Data Type for Processing</td><td colspan="4">Single-precision floating-point, Fixed-point (two's complement)</td></tr>
<tr><td>Number of cycles for calculation</td><td colspan="4">Sine: 5
Cosine: 5
Arctangent: 14
hypot_k: 14</td></tr>
</table>

Note: k is a constant. See section 21.3.1. Arithmetic Processing.

Note 1. float_max is the maximum value that can be expressed as single-precision floating-point: $(2 - 2^{-23}) \times 2^{127}$.



Note: sin and cos arithmetic:
Operation to find $\sin\theta$ and $\cos\theta$, which are the triangular ratios of the input angle values θ.
atan arithmetic:
Operation to find the angle value between the vector OA (O is the origin and A is the point indicated by the input coordinates x and y) and the positive part of the x-axis.
hypot_k arithmetic:
Operation to find a value proportional to the distance from the point A indicated by the input coordinates x and y to the origin O.

Note: By multiplying the TFU output result hypot_k by the constant k, the distance between OA ($\sqrt{x^2+y^2}$) can be obtained. for details about k, see section 21.3.1. Arithmetic Processing.

Figure 21.1 Explanation for operations

21.1.1 Precautions on Use of the Trigonometric Function Unit

This section describes precautions on use of the trigonometric function unit.

21.1.1.1 General Precautions

If another operation is started after the operation is started and before the result is read, the result of the preceding operation will be discarded.

21.2 Register Descriptions

21.2.1 FXSCIOC : Fixed-point Sine Cosine Input/Output Setting Register

Base address: TFU = 0x9000_3000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	OF[1:0]		—	—	—	IUF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IUF	Input unit and format setting Specify the unit and format for input value of fixed-point sincos operation 0: unit, format = turn, Q1.31 1: unit, format = radian, Q3.29	R/W

Bit	Symbol	Function	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
5:4	OF[1:0]	Output format setting 0 0: Q1.31 0 1: Q2.30 1 0: Q3.29 1 1: Setting prohibited	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

IUF bit (Input unit and format setting)

The IUF bit specifies the unit and format for input value of fixed-point sincos operation.

OF[1:0] bits (Output format setting)

The OF[1:0] bits specify the format for output value of fixed-point sincos operation.

When Q1.31 is selected, +1.0 cannot be expressed accurately. If $\sin\theta$ or $\cos\theta$ is +1.0 ordinally when Q1.31, +1.0 - 2^{-31} will be read out from the FXSCDT0 and FXSCDT1 registers.

This register should be set before the calculation start. If this register is changed during the calculation, the operation is not guaranteed.

21.2.2 FXATIOC : Fixed-point Arctangent hypot_k Input/Output Setting Register

Base address: TFU = 0x9000_3000

Offset address: 0x05

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	OUF	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	OUF	Output unit and format setting Specify the unit and format for output value of fixed-point arctangent operation 0: unit, format = turn, Q1.31 1: unit, format = radian, Q3.29	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

OUF bit (Output unit and format setting)

The OUF bit specifies the unit and format for output value when arctangent calculation is executed by fixed-point atanhypot_k operation. The output format of hypot_k calculation executed with arctangent calculation in parallel is Q3.29 regardless of this bit setting.

This register should be set before the calculation start. If this register is changed during the calculation, the operation is not guaranteed.

21.2.3 TRGSTS : Trigonometric Status Register

Base address: TFU = 0x9000_3000

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	ERRF	BSYF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BSYF	Calculation in progress flag 0: No calculating 1: Calculating	R/W*1
1	ERRF	Input error flag 0: No input error occurred 1: Input error occurred	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to this bit is invalid.

BSYF bit (Calculation in progress flag)

The BSYF bit specifies the calculation in progress flag. Writing to this bit is ignored.

[Setting condition]

- When calculation started

[Clearing condition]

- When calculation completed

ERRF bit (Input error flag)

The ERRF bit specifies the input error flag.

Do not write to this bit except error flag restoration procedure shown in section 21.3.8. Usage of Interrupt.

[Setting condition]

- When input error occurred or 1 wrote to this bit. For details on input error, see Table 21.8.

[Clearing condition]

- When the next calculation started or 0 wrote to this bit

21.2.4 FPSCDT0 : Floating-point Sine Cosine Data Register 0

Base address: TFU = 0x9000_3000

Offset address: 0x10

Bit position:	31																												0
Bit field:	FPSCDT0[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FPSCDT0[31:0]	Floating-point Sine Cosine Data Register 0	R/W

For the floating-point sincos operation, the FPSCDT0 register is dedicated for the output value $\cos\theta$ of the trigonometric function unit. See Table 21.2 for detail. In terms of usage of this register, see section 21.3.7. Procedure for Trigonometric Function Operation.

Writing to this register is prohibited even during any operations.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

For calculations except the floating-point sincos operation, do not access this register. If other data register 0 such as FPATDT0, FXSCDT0, and FXATDT0 registers is written, this register value will be uncertainty.

When the DTSR0 register is written, this register value will be the same as DTSR0 register writing value.

21.2.5 FPSCDT1 : Floating-point Sine Cosine Data Register 1

Base address: TFU = 0x9000_3000

Offset address: 0x14

Bit position:	31																												0
Bit field:	FPSCDT1[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FPSCDT1[31:0]	Floating-point Sine Cosine Data Register 1	R/W

For the floating-point sincos operation, the FPSCDT1 register is shared for the input angle value θ and the output value $\sin\theta$ of the trigonometric function unit. See Table 21.2 for detail. Writing to this register starts the floating-point sincos operation. In terms of usage of this register, see section 21.3.7. Procedure for Trigonometric Function Operation.

Writing to this register is prohibited even during any operations.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

For calculations except the floating-point sincos operation, do not access this register. If other data register 1 such as FPATDT1, FXSCDT1, and FXATDT1 registers is written, this register value will be uncertainty.

When the DTSR1 register is written, this register value will be the same as DTSR1 register writing value.

Table 21.2 Input/Output value of FPSCDT0 and FPSCDT1

Register	Input value	Output value
FPSCDT0	—	$\cos\theta$
FPSCDT1	Angle θ	$\sin\theta$

21.2.6 FPATDT0 : Floating-point Arctangent hypot_k Data Register 0

Base address: TFU = 0x9000_3000

Offset address: 0x18

Bit position:	31																												0
Bit field:	FPATDT0[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FPATDT0[31:0]	Floating-point Arctangent hypot_k Data Register 0	R/W

For the floating-point atanhypot_k operation, the FPATDT0 register is shared for the input coordinate value x and the output value $\sqrt{x^2+y^2}/k$ of the trigonometric function unit. See Table 21.3 for detail. In terms of usage of this register, see section 21.3.7. Procedure for Trigonometric Function Operation.

Writing to this register is prohibited even during any operations.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

For calculations except the floating-point atanhypot_k operation, do not access this register. If other data register 0 such as FPSCDT0, FXSCDT0, and FXATDT0 registers is written, this register value will be uncertainty.

When the DTSR0 register is written, this register value will be the same as DTSR0 register writing value.

21.2.7 FPATDT1 : Floating-point Arctangent hypot_k Data Register 1

Base address: TFU = 0x9000_3000

Offset address: 0x1C

Bit position:	31																												0
Bit field:	FPATDT1[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FPATDT1[31:0]	Floating-point Arctangent hypot_k Data Register 1	R/W

For the floating-point atanhypot_k operation, the FPATDT1 register is shared for the input coordinate value y and the output value atan(y/x) of the trigonometric function unit. See Table 21.3 for detail. Writing to this register starts the floating-point atanhypot_k operation. In terms of usage of this register, see section 21.3.7. Procedure for Trigonometric Function Operation.

Writing to this register is prohibited even during any operations.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

For calculations except the floating-point atanhypot_k operation, do not access this register. If other data register 1 such as FPSCDT1, FXSCDT1, and FXATDT1 registers is written, this register value will be uncertainty.

When the DTSR1 register is written, this register value will be the same as DTSR1 register writing value.

Table 21.3 Input/Output value of FPATDT0 and FPATDT1

Register	Input value	Output value
FPATDT0	Coordinates x	hypot_k
FPATDT1	Coordinates y	atan(y/x)

21.2.8 FXSCDT0 : Fixed-point Sine Cosine Data Register 0

Base address: TFU = 0x9000_3000

Offset address: 0x20

Bit position:	31																												0
Bit field:	FXSCDT0[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FXSCDT0[31:0]	Fixed-point Sine Cosine Data Register 0	R/W

For the fixed-point sincos operation, the FXSCDT0 register is dedicated for the output value $\cos\theta$ of the trigonometric function unit. See Table 21.4 for detail. In terms of usage of this register, see section 21.3.7. Procedure for Trigonometric Function Operation.

Writing to this register is prohibited even during any operations.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

For calculations except the fixed-point sincos operation, do not access this register. If other data register 0 such as FPSCDT0, FPATDT0, and FXATDT0 registers is written, this register value will be uncertainty.

When the DTSR0 register is written, this register value will be the same as DTSR0 register writing value.

21.2.9 FXSCDT1 : Fixed-point Sine Cosine Data Register 1

Base address: TFU = 0x9000_3000

Offset address: 0x24

Bit position:	31																												0
Bit field:	FXSCDT1[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FXSCDT1[31:0]	Fixed-point Sine Cosine Data Register 1	R/W

For the fixed-point sincos operation, the FXSCDT1 register is shared for the input angle value θ and the output value $\sin\theta$ of the trigonometric function unit. See Table 21.4 for detail. Writing to this register starts the fixed-point sincos operation. In terms of usage of this register, see section 21.3.7. Procedure for Trigonometric Function Operation.

Writing to this register is prohibited even during any operations.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

For calculations except the fixed-point sincos operation, do not access this register. If other data register 1 such as FPSCDT1, FPATDT1, and FXATDT1 registers is written, this register value will be uncertainty.

When the DTSR1 register is written, this register value will be the same as DTSR1 register writing value.

Table 21.4 Input/Output value of FXSCDT0 and FXSCDT1

Register	Input value	Input unit and format	Output value	Output format
FXSCDT0	Unused	—	$\cos\theta$	Depends on FXSCIOC.OF[1:0] setting: Q1.31, Q2.30, or Q3.29
FXSCDT1	Angle θ	When FXSCIOC.IUF = 0: [turn], Q1.31 When FXSCIOC.IUF = 1: [radian], Q3.29	$\sin\theta$	Same as $\cos\theta$

When Q1.31 is selected, +1.0 cannot be expressed accurately. If $\sin\theta$ or $\cos\theta$ is +1.0 ordinally when Q1.31, $+1.0 - 2^{-31}$ will be read out from the FXSCDT0 and FXSCDT1 registers.

21.2.10 FXATDT0 : Fixed-point Arctangent hypot_k Data Register 0

Base address: TFU = 0x9000_3000

Offset address: 0x28

Bit position:	31																												0
Bit field:	FXATDT0[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FXATDT0[31:0]	Fixed-point Arctangent hypot_k Data Register 0	R/W

For the fixed-point atanhypot_k operation, the FXATDT0 register is shared for the input coordinate value x and the output value $\sqrt{x^2 + y^2}/k$ of the trigonometric function unit. See Table 21.5 for detail. In terms of usage of this register, see section 21.3.7. Procedure for Trigonometric Function Operation.

Writing to this register is prohibited even during any operations.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

For calculations except the fixed-point atanhypot_k operation, do not access this register. If other data register 0 such as FPSCDT0, FPATDT0, and FXSCDT0 registers is written, this register value will be uncertainty.

When the DTSR0 register is written, this register value will be the same as DTSR0 register writing value.

21.2.11 FXATDT1 : Fixed-point Arctangent hypot_k Data Register 1

Base address: TFU = 0x9000_3000

Offset address: 0x2C

Bit position:	31																												0
Bit field:	FXATDT1[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	FXATDT1[31:0]	Fixed-point Arctangent hypot_k Data Register 1	R/W

For the fixed-point atanhypot_k operation, the FXATDT1 register is shared for the input coordinate value y and the output value atan(y/x) of the trigonometric function unit. See Table 21.5 for detail. Writing to this register starts the fixed-point atanhypot_k operation. In terms of usage of this register, see section 21.3.7. Procedure for Trigonometric Function Operation.

Writing to this register is prohibited even during any operations.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

For calculations except the fixed-point atanhypot_k operation, do not access this register. If other data register 1 such as FPSCDT1, FPATDT1, and FXSCDT1 registers is written, this register value will be uncertainty.

When the DTSR1 register is written, this register value will be the same as DTSR1 register writing value.

Table 21.5 Input/Output value of FXATDT0 and FXATDT1

Register	Input value	Input format	Output value	Output unit and format
FXATDT0	Coordinates x	Q1.31	hypot_k	[—], Q3.29
FXATDT1	Coordinates y	Q1.31	atan(y/x)	When FXATIOC.OUF = 0: [turn], Q1.31 When FXATIOC.OUF = 1: [radian], Q3.29

21.2.12 DTSR0 : Data Save Restore Register 0

Base address: TFU = 0x9000_3000

Offset address: 0x30

Bit position:	31																												0
Bit field:	DTSR0[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DTSR0[31:0]	Data Save Restore Register 0	R/W

The DTSR0 register is used to save and restore data in the data register 0 such as FPSCDT0, FPATDT0, FXSCDT0, and FXATDT0.

When read this register, the last written value into data register 0 or the result of the last arithmetic operation for data register 0 will be read.

When write to this register, the value for all data register 0 will be set to the written data of this register.

Writing to this register is prohibited during the operation.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

In terms of usage of this register, see section 21.3.8. Usage of Interrupt.

21.2.13 DTSR1 : Data Save Restore Register 1

Base address: TFU = 0x9000_3000

Offset address: 0x34

Bit position:	31																												0
Bit field:	DTSR1[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DTSR1[31:0]	Data Save Restore Register 1	R/W

The DTSR1 register is used to save and restore data in the data register 1 such as FPSCDT1, FPATDT1, FXSCDT1, and FXATDT1.

When read this register, the last written value into data register 1 or the result of the last arithmetic operation for data register 1 will be read.

When write to this register, the value for all data register 1 will be set to the written data of this register.

Writing to this register is prohibited during the operation.

If read access is made to this register during the calculation, the result is read after the calculation is completed. At this time, the bus access is forced to wait until the operation is completed.

In terms of usage of this register, see section 21.3.8. Usage of Interrupt.

21.3 Operation

21.3.1 Arithmetic Processing

The trigonometric function unit has two arithmetic operations, the sincos operation and the atanhypot_k operation. See Table 21.6 for detail.

Table 21.6 Arithmetic processing

Operation	Input value	Output value
sincos	Angle value θ	$\cos\theta$ and $\sin\theta$
atanhypot_k	Coordinates x and y	atan(y/x) and hypot_k

The value of scaling factor k is:

$$k = \prod_{i=0}^{\infty} \frac{1}{\sqrt{1 + 2^{-2i}}} \approx 0.607252935008881256169 4$$

21.3.2 Input and Output Value Formats

The input/output value of the TFU supports single-precision floating-point and fixed-point. For the single-precision floating-point in detail, see Table 21.2 and Table 21.7. For the fixed-point in detail, see Table 21.3. As a side of note, fixed-point input/output data is in 32-bit format, but it is performed with less than 32 bits of precision.

- Floating-point
  Support single-precision floating-point specified by the IEEE754 standard.
  - Single-precision floating-point

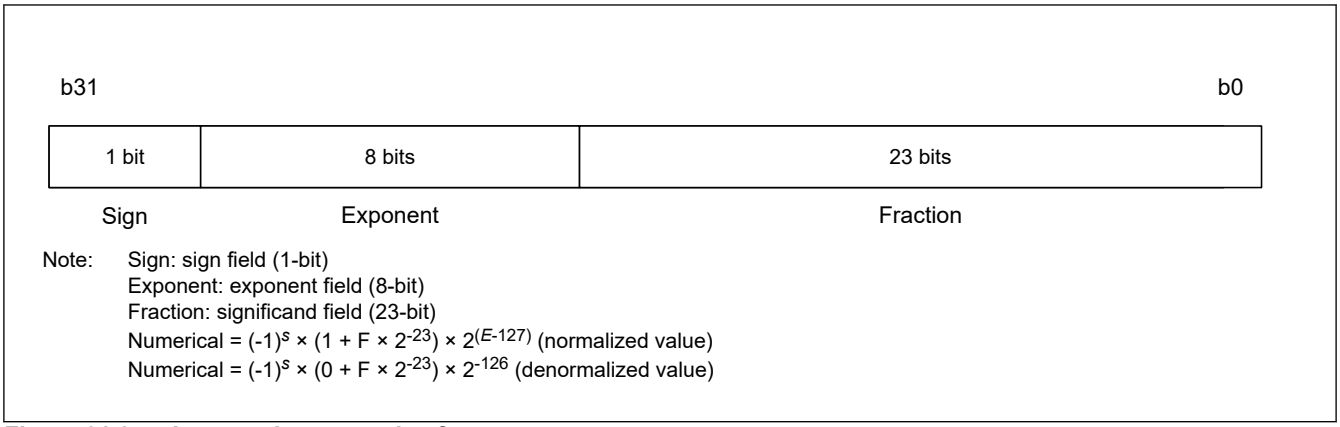


Figure 21.2 Input and output value formats

Table 21.7 Support single-precision floating-point

S	E	F	Numerical
Any value	0 < E < 255	Any value	$(-1)^s \times 1.F \times 2^{(E-127)}$ (normalized value — Normal Numbers)
Any value	E = 0	F > 0	$(-1)^s \times 0.F \times 2^{-126}$ (denormalized value — Subnormal Numbers)
S = 0	E = 0	F = 0	$(-1)^0 \times 0.0$ (positive zero — +0)
S = 1	E = 0	F = 0	$(-1)^{-1} \times 0.0$ (negative zero — -0)
S = 0	E = 255	F = 0	(positive infinity — +∞)
S = 1	E = 255	F = 0	(negative infinity — -∞)
Any value	E = 255	$2^{22} > F > 0$	(non-number — SNaN: Signaling Not a Number)
Any value	E = 255	$F \geq 2^{22}$	(non-number — QNaN: Quiet Not a Number)

- Fixed-point
  Format of two's complement representation shown in Figure 21.3 is used.

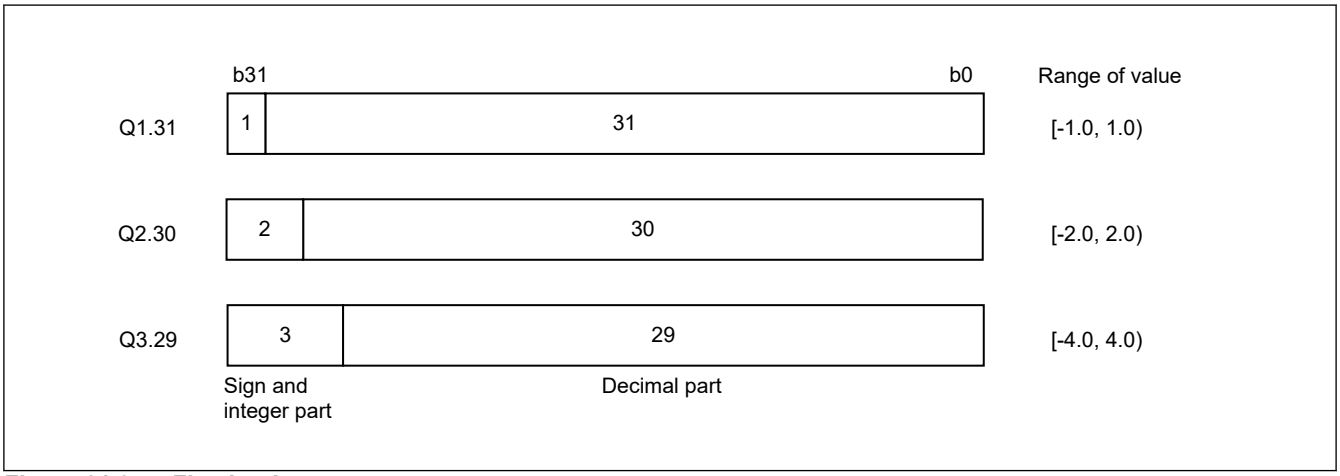


Figure 21.3 Fixed-point

21.3.3 Angle Unit

The units of angle used in the TFU are radian and turn. The radian is one rotation expressed as 0 to 2π. The turn is one rotation expressed as 0 to 1.

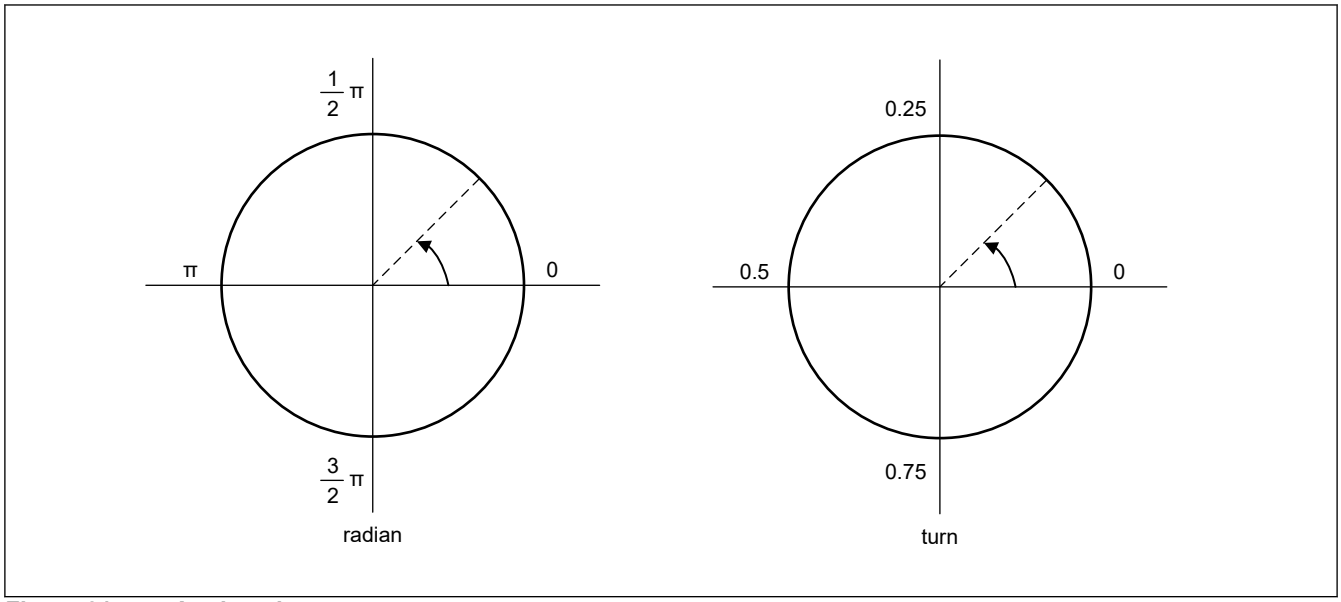


Figure 21.4 Angle unit

21.3.4 Relationship Between Input and Output Values for Sincos Operation

When the input values in the floating-point sincos operation are ±0, ±∞, SNaN (Signaling Not a Number), and QNaN (Quiet Not a Number), a fixed value is output as shown in Table 21.8.

Table 21.8 Relationship between special input value and its output value (for floating-point sincos operation)

Input (θ)	Output ($\cos\theta$)	Output ($\sin\theta$)
-∞	QNaN	QNaN
-0	+1	-0
+0	+1	+0
+∞	QNaN	QNaN
SNaN/QNaN	QNaN	QNaN

Note: The output value of QNaN is 0xFFC0_0000.

21.3.5 Relationship Between Input and Output Values for Atan Operation

When either the input values x or y in the floating-point atan operation are ±0, ±∞, SNaN (Signaling Not a Number), and QNaN (Quiet Not a Number), a fixed value is output as shown in Table 21.9.

If both input values are ±0, it is determined as an input error.

Table 21.9 Relationship between special input value and its output value (for floating-point atan operation)

		x						
		-∞	**Negative value**	**-0**	**+0**	**Positive value**	**+∞**	**SNaN/QNaN**
y	**-∞**	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN
	Negative value	QNaN	—	-π/2	-π/2	—	QNaN	QNaN
	-0	QNaN	-π	QNaN[*1]	QNaN[*1]	-0	QNaN	QNaN
	+0	QNaN	+π	QNaN[*1]	QNaN[*1]	+0	QNaN	QNaN
	Positive value	QNaN	—	+π/2	+π/2	—	QNaN	QNaN
	+∞	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN
	SNaN/QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN

Note 1. An input error occurs for the special input value, and the input error flag (TRGSTS.ERRF) is set.

For fixed-point atan operation, when both x and y input values are 0, it is determined as an input error. When an input error happens, output value of atan operation is 0.

21.3.6 Relationship Between Input and Output Values for hypot_k Operation

When either the input values x or y in the floating-point hypot_k operation are ±0, ±∞, SNaN (Signaling Not a Number), and QNaN (Quiet Not a Number), or when both x and y are ± 0, a fixed value is output as shown in Table 21.10.

If both input values are ±0, it is determined as an input error.

Table 21.10 Relationship between special input value and its output value (for floating-point hypot_k operation)

		x						
		-∞	**Negative value**	**-0**	**+0**	**Positive value**	**+∞**	**SNaN/QNaN**
y	**-∞**	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN
	Negative value	QNaN	—	—	—	—	QNaN	QNaN
	-0	QNaN	—	+0*1	+0*1	—	QNaN	QNaN
	+0	QNaN	—	+0*1	+0*1	—	QNaN	QNaN
	Positive value	QNaN	—	—	—	—	QNaN	QNaN
	+∞	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN
	SNaN/QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN	QNaN

Note 1. An input error occurs for the special input value, and the input error flag (TRGSTS.ERRF) is set.

For fixed-point hypot_k operation, when both x and y input values are 0, it is determined as an input error. Because atan operation becomes error, although it is no problem for hypot_k operation. When an input error happens, output value of hypot_k is 0.

21.3.7 Procedure for Trigonometric Function Operation

Figure 21.5 shows the procedure for sincos operation. There are two procedures. The advantages and disadvantages of each are shown in Table 21.11. For floating-point sincos operation, replace SCDTn in the figure with FPSCDTn. And for fixed-point sincos operation, replace with FXSCDTn. (n = 0 or 1)

Figure 21.6 shows the procedure for atanhypot_k operation. There are two procedures. The advantages and disadvantages of each are shown in Table 21.11. For floating-point atanhypot_k operation, replace ATDTn in the figure with FPATDTn. And for fixed-point atanhypot_k operation, replace with FXATDTn. (n = 0 or 1)

Table 21.11 Advantages and disadvantages of sincos and atanhypot_k operations

Method	Advantages	Disadvantages
Procedure 1	Not occupy the bus.	Operation end determination is required by checking TRGSTS.BSYF.
Procedure 2	Operation end determination is not required by checking TRGSTS.BSYF. (reduced number of execution cycles)	Occupy the bus while waiting for the number of execution cycles when the result of the operation is read.

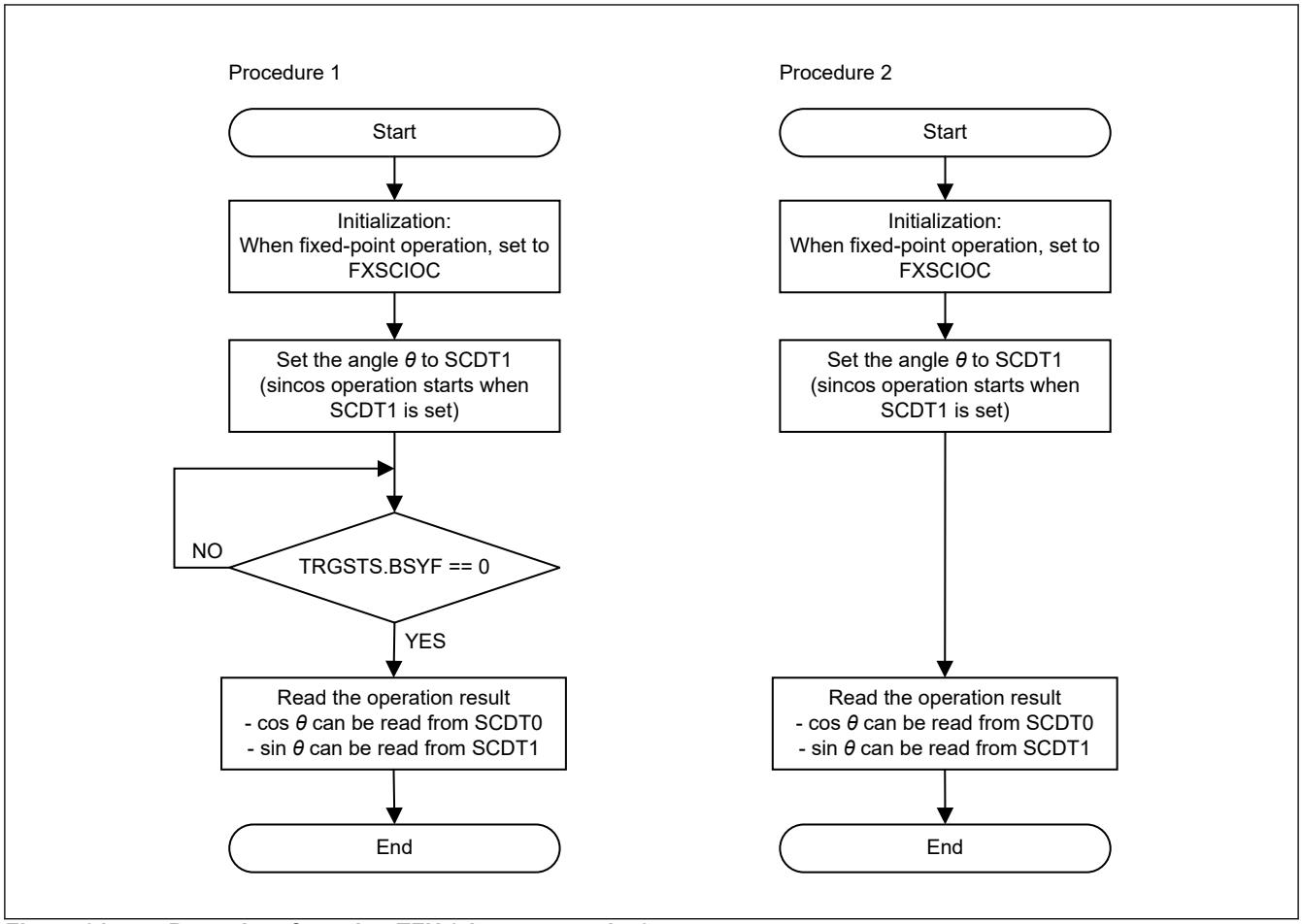


Figure 21.5 Procedure for using TFU (sincos operation)

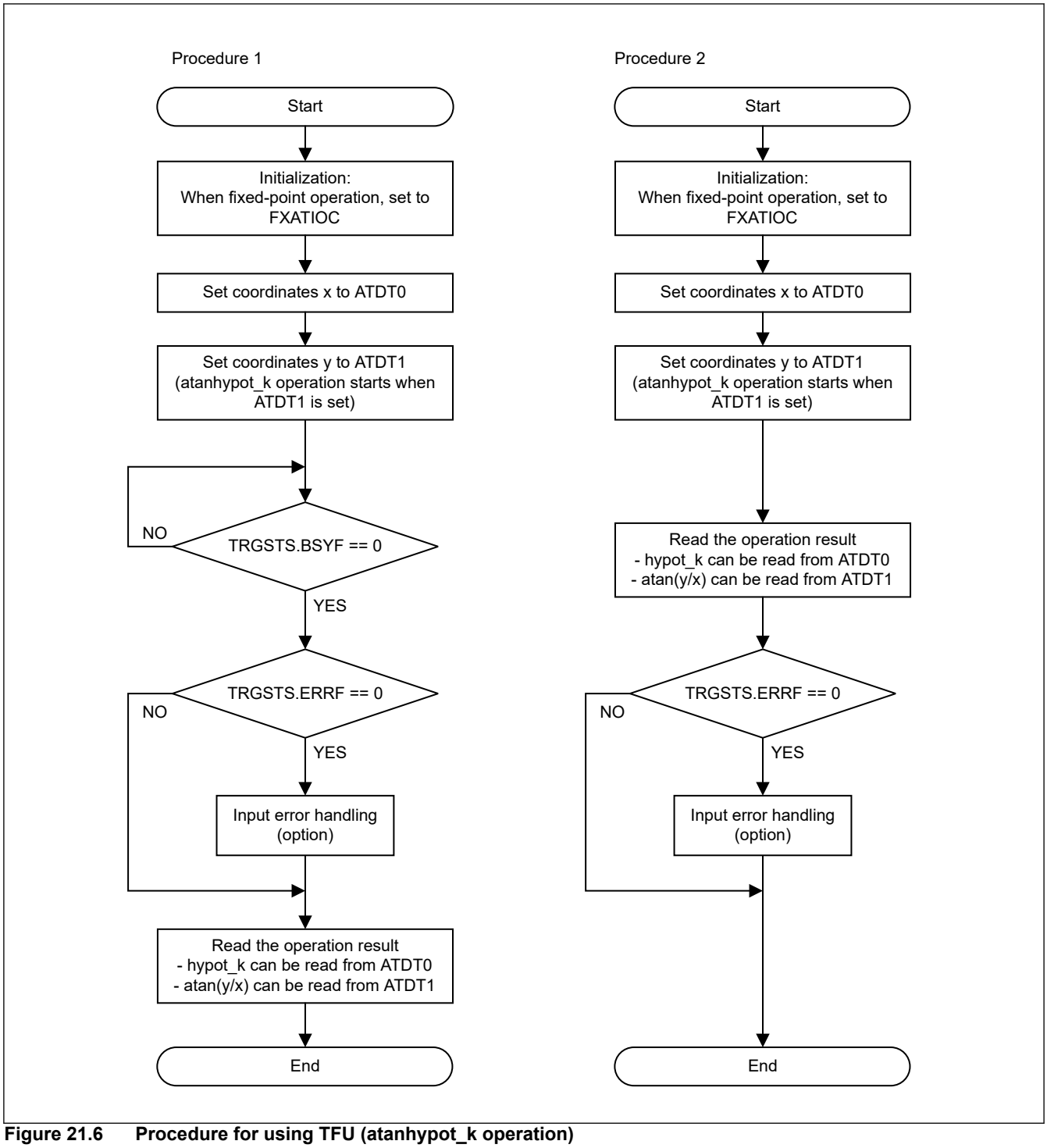


Figure 21.6 Procedure for using TFU (atanhypot_k operation)

21.3.8 Usage of Interrupt

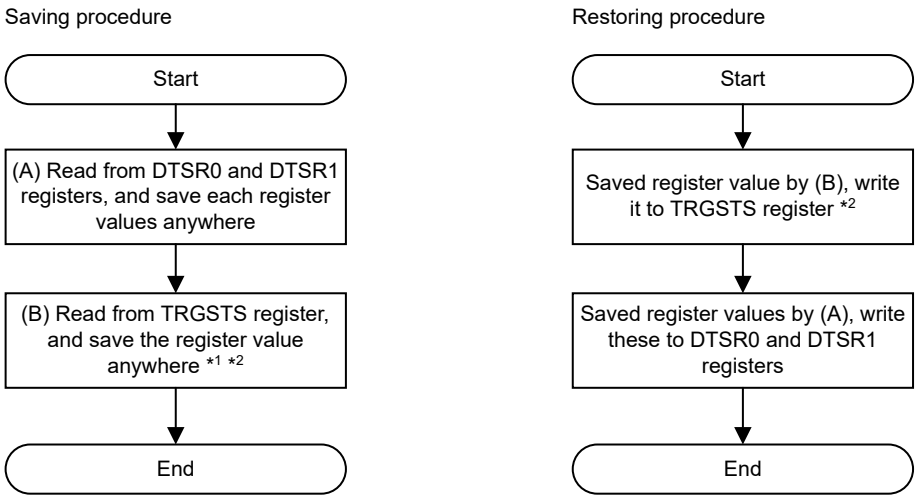
In the middle of procedure for trigonometric function operation, when an interrupt occurs between a data register write and an error flag/data register read and trigonometric function operation is executed in the interrupt handler, malfunction occurs after returning from the interrupt handler.

Therefore, the following measures are required so that malfunction does not occur when multiple interrupts are used, and multiple interrupt handlers use TFU module.

1. Disable interrupts on the process from a data register write to an error flag and data register read
2. Save and restore an error flag and data registers in an interrupt handler

The latter measure is described as below.

Regardless of a kind of an operation in the interrupt handler and an operation being performed before the interrupt is received, data registers and error flag should be saved/restored by using DTSR0, DTSR1, and TRGSTS registers. Figure 21.7 shows the data save/restore procedure. By following this procedure, the operation before the interrupt is received can be continued without malfunction after returning from the interrupt handler.



Note 1. (B) has to be executed after (A).
Note 2. Can be omitted if the error flag is not used in any arithmetic process.

Figure 21.7 Data save/restore procedure in exception handler

When a setting will be changed in the interrupt handler and an arithmetic processing will be performed, in addition to the data registers and error flag, setting registers such as FXSCIOC and FXATIOC need to be saved and restored as well. Save and restore the setting registers as necessary. When saving the setting registers, be sure to do so after saving (A).

22. Compare Match Timer (CMT)

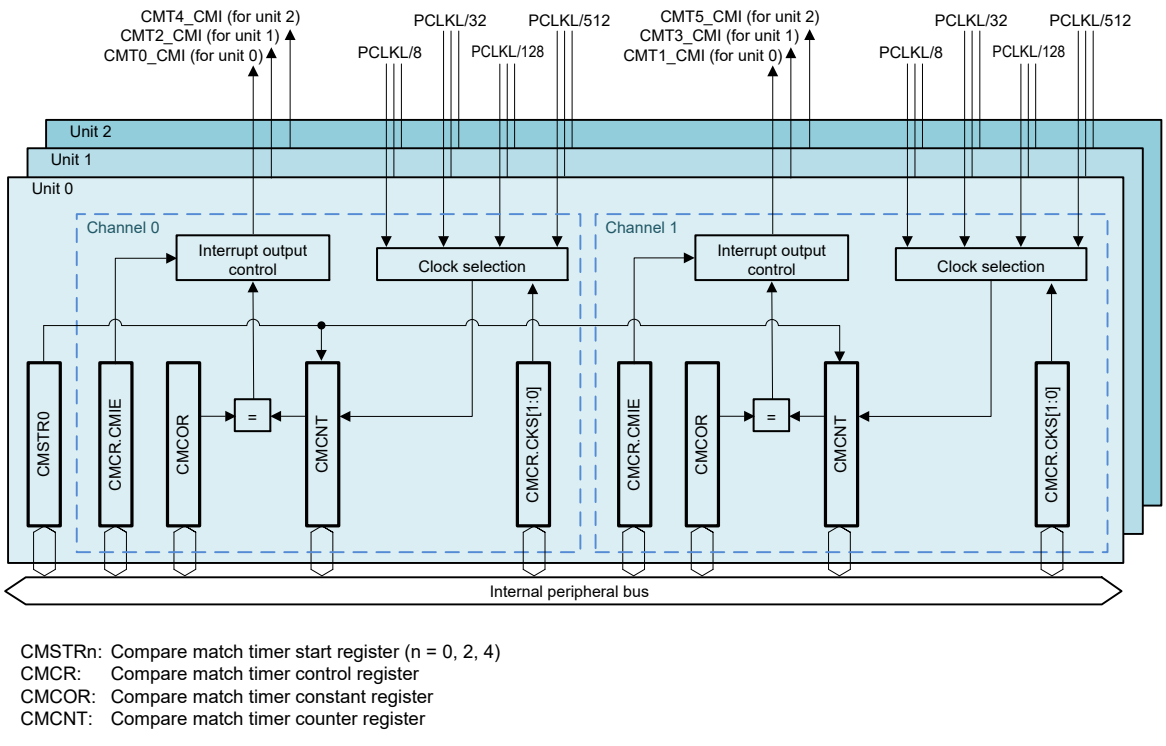
22.1 Overview

The LSI has three on-chip compare match timer (CMT) units, each consisting of a 2-channel 16-bit timer, for a total of six channels. The CMT has a 16-bit counter and can generate interrupts at set intervals.

Table 22.1 lists the specifications for the CMT and Figure 22.1 shows a block diagram of the CMT.

Table 22.1 CMT specifications

Item	Description
Number of internal channels	6 channels (2 channels × 3 units)
Timer counter (per channel)	16-bit up counter (counted according to the count enable signal output by the prescaler.) Returned to 0x0000 after compare match.
Prescaler (per channel)	9-bit counter (linked with enabling/disabling of timer counter operation.) ● Outputs four types of count clocks. The type can be selected from PCLKL/8, PCLKL/32, PCLKL/128, and PCLKL/512.
Interrupt	A compare match interrupt can be requested for each channel.
Module-stop function	Module-stop state can be set to reduce power consumption



CMSTRn: Compare match timer start register (n = 0, 2, 4)
CMCR: Compare match timer control register
CMCOR: Compare match timer constant register
CMCNT: Compare match timer counter register

Figure 22.1 CMT block diagram

22.2 Register Descriptions

22.2.1 CMSTRn : Compare Match Timer Start Register (n = 0, 2, 4)

Base address: CMTC = 0x8004_0000

Offset address: 0x000 + 0x200 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STR1	STR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STR0	CMT Channel n Count Start This bit selects starting or stopping of the CMTn.CMCNT counter operation. The corresponding prescaler starts or stops according to the setting of this bit. 0: The CMTn.CMCNT counter is stopped. 1: The CMTn.CMCNT counter is started.	R/W
1	STR1	CMT Channel n+1 Count Start This bit selects starting or stopping of the CMTn+1.CMCNT counter operation. The corresponding prescaler starts or stops according to the setting of this bit. 0: The CMTn+1.CMCNT counter is stopped. 1: The CMTn+1.CMCNT counter is started.	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W

The CMSTR0 register sets starting or stopping of the CMT0.CMCNT and CMT1.CMCNT counters of unit 0.

The CMSTR2 register sets starting or stopping of the CMT2.CMCNT and CMT3.CMCNT counters of unit 1.

The CMSTR4 register sets starting or stopping of the CMT4.CMCNT and CMT5.CMCNT counters of unit 2.

22.2.2 CMCR : Compare Match Timer Control Register

Base address: CMT0 = 0x8004_0002, CMT2 = 0x8004_0402, CMT4 = 0x8004_0802
CMT1 = 0x8004_0008, CMT3 = 0x8004_0408, CMT5 = 0x8004_0808

Offset address: 0x000

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	CMIE	—	—	—	—	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CKS[1:0]	Clock Select 0 0: PCLKL/8 0 1: PCLKL/32 1 0: PCLKL/128 1 1: PCLKL/512	R/W
5:2	—	These bits are read as 0. The write value should be 0.	R/W
6	CMIE	Compare Match Interrupt Enable 0: Compare match interrupt (CMTn_CMI) disabled 1: Compare match interrupt (CMTn_CMI) enabled	R/W
7	—	The read value is undefined. The write value should be 0.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W

The CMCR register specifies a clock used for count-up operation.

CKS[1:0] bits (Clock Select)

These bits select a clock to be input to the CMTn.CMCNT counter from the internal clocks obtained by dividing the frequency of the low-speed peripheral module clock (PCLKL) (n = 0 to 5). Setting the CMSTRm.STRn bit to 1 starts count-up operation of the corresponding CMCNT counter by using the clock selected in the CKS[1:0] bits (m = 0, 2, 4; n = 0, 1).

CMIE bit (Compare Match Interrupt Enable)

This bit selects whether to enable or disable generation of a compare match interrupt (CMTn_CMI) when the values in the CMCNT counter and in the CMCOR register match (n = 0 to 5).

22.2.3 CMCNT : Compare Match Timer Counter Register

Base address: CMT0 = 0x8004_0002, CMT2 = 0x8004_0402, CMT4 = 0x8004_0802
CMT1 = 0x8004_0008, CMT3 = 0x8004_0408, CMT5 = 0x8004_0808

Offset address: 0x002

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CMCNT counter (the main unit of the compare match timer) is a readable/writable up-counter. When an internal clock is selected by the CMCR.CKS[1:0] bits and the CMSTRm.STRn (m = 0, 2, 4; n = 0, 1) bit is set to 1, the CMCNT counter starts counting up using the selected clock.

When the value in the CMCNT counter and the value in the CMCOR register match, the CMCNT counter is cleared to 0x0000. At the same time, a compare match interrupt (CMTn_CMI) is generated (n = 0 to 5).

22.2.4 CMCOR : Compare Match Timer Constant Register

Base address: CMT0 = 0x8004_0002, CMT2 = 0x8004_0402, CMT4 = 0x8004_0802
CMT1 = 0x8004_0008, CMT3 = 0x8004_0408, CMT5 = 0x8004_0808

Offset address: 0x004

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CMCOR register is a readable/writable register to set a cycle for compare match with the CMCNT counter. The cycle for compare matches is as follows:

Compare-match cycle = (setting of the CMCOR register + 1) × counter-clock cycle[*1]

Note 1. This is a clock cycle set by the CMCR.CKS[1:0] bits.

22.3 Operation

22.3.1 Periodic Count Operation

When an internal clock is selected by the CMCR.CKS[1:0] bits and the CMSTRm.STRn (m = 0, 2, 4; n = 0, 1) bit is set to 1, the CMCNT counter starts counting up using the selected clock.

When the value in the CMCNT counter and the value in the CMCOR register match, the CMCNT counter is cleared to 0x0000, and then a compare match interrupt (CMTn_CMI) is generated (n = 0 to 5). The CMCNT counter then starts counting up again from 0x0000. Figure 22.2 shows the operation of the CMCNT counter.

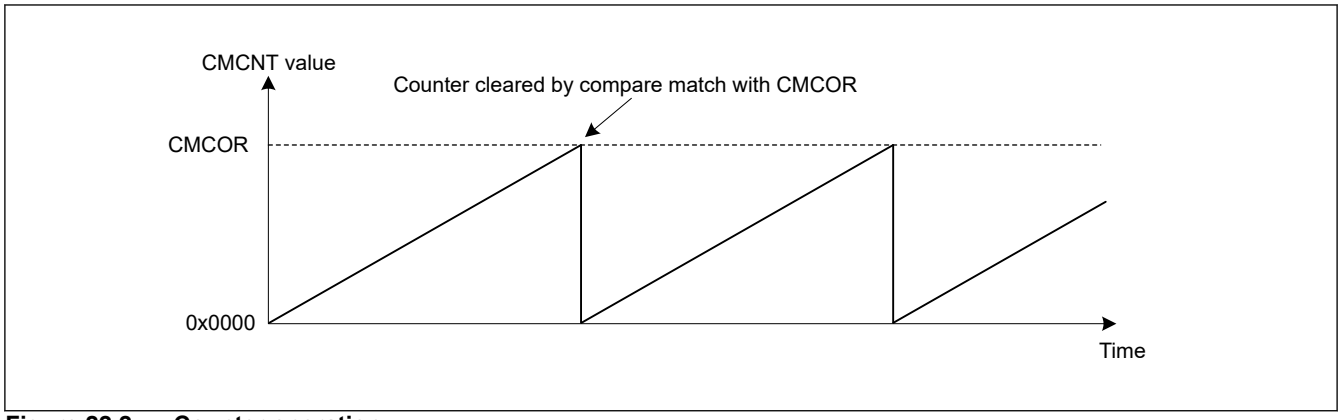


Figure 22.2 Counter operation

22.3.2 CMCNT Count Timing

As the count clock, one of four internal clocks (PCLKL/8, PCLKL/32, PCLKL/128, and PCLKL/512) obtained by dividing the low-speed peripheral module clock (PCLKL) can be selected with the CMCR.CKS[1:0] bits. Figure 22.3 shows the timing of the CMCNT counter.

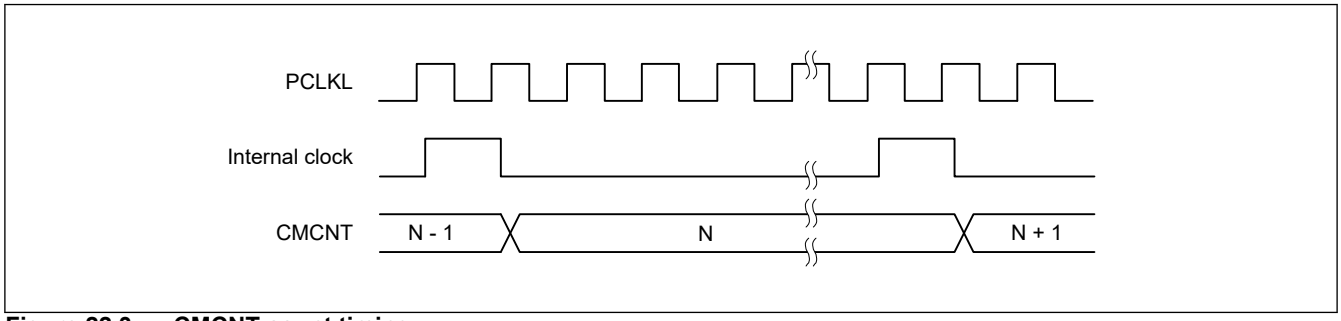


Figure 22.3 CMCNT count timing

22.4 Interrupts

22.4.1 Interrupt Sources

For each CMT channel, a compare-match interrupt (CMTn_CMI) is allocated and has a different event number. The priority of channels can be changed by the interrupt controller settings.

Table 22.2 CMT interrupt sources

Name	Interrupt Sources	CPU0 GIC request	DMAC activation
CMT0_CMI	Compare match between the CMT0.CMCNT counter and CMT0.CMCOR register	Possible	Possible
CMT1_CMI	Compare match between the CMT1.CMCNT counter and CMT1.CMCOR register	Possible	Possible
CMT2_CMI	Compare match between the CMT2.CMCNT counter and CMT2.CMCOR register	Possible	Possible
CMT3_CMI	Compare match between the CMT3.CMCNT counter and CMT3.CMCOR register	Possible	Possible
CMT4_CMI	Compare match between the CMT4.CMCNT counter and CMT4.CMCOR register	Possible	Possible
CMT5_CMI	Compare match between the CMT5.CMCNT counter and CMT5.CMCOR register	Possible	Possible

22.4.2 Timing of Compare Match Interrupt Generation

When the CMCNT counter and the CMCOR register match, a compare match interrupt (CMTn_CMI) is generated.

A compare match signal is generated at the last state in which the values match (the timing when the CMCNT counter updates the matched count value). That is, after a match between the CMCOR register and the CMCNT counter, the compare match signal is not generated until the CMCNT counter input clock is generated (n = 0 to 5).

Figure 22.4 shows the timing of a compare match interrupt.

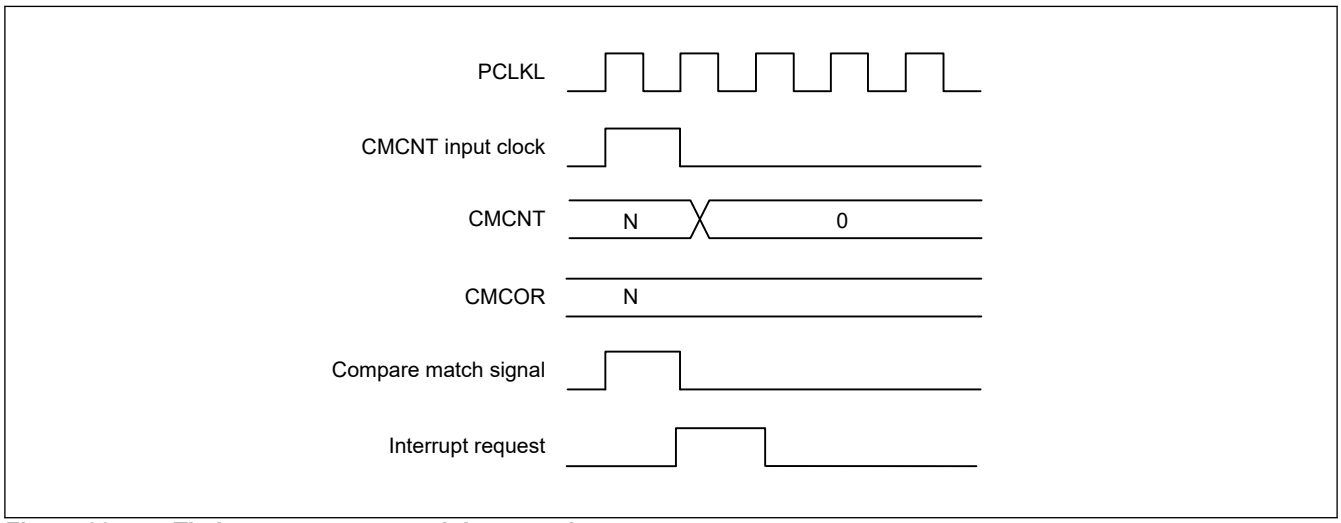


Figure 22.4 Timing a compare match interrupt is set

22.5 Usage Notes

22.5.1 Setting the Module Stop Function

The CMT can be enabled or disabled using the Module Stop Control register. After a reset, the CMT is in the module-stop state. The registers can be accessed by canceling the module-stop state. For details, see section 8, Low-Power Consumption Function.

22.5.2 Conflict between Write and Compare-Match Processes of CMCNT

When the compare match signal is generated while writing to the CMCNT counter, clearing the CMCNT counter is given priority over the writing to it. In this case, the CMCNT counter is not written to. Figure 22.5 shows the timing to clear the CMCNT counter.

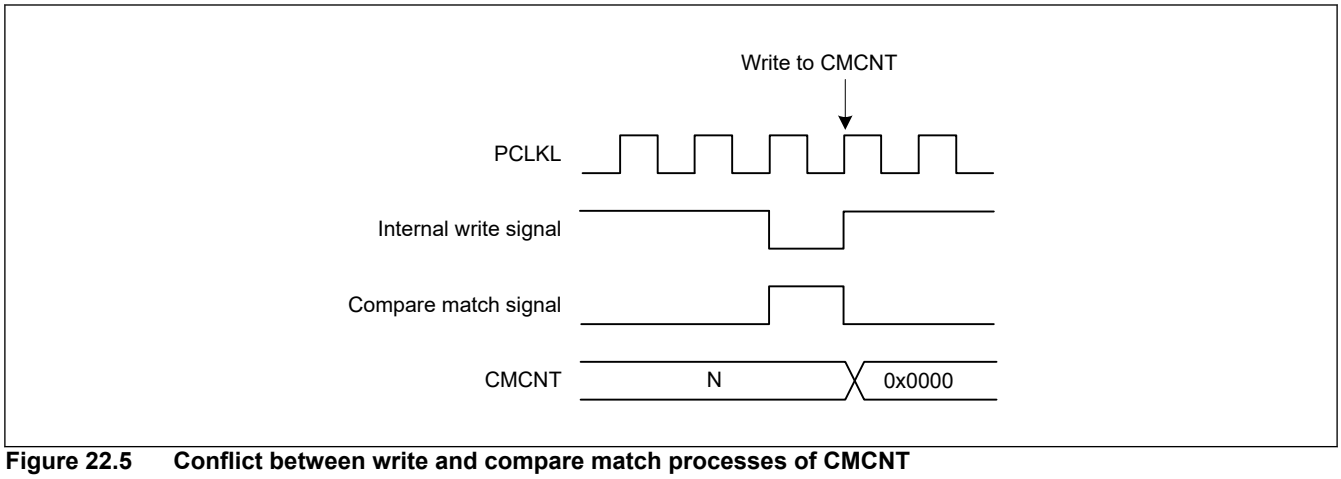


Figure 22.5 Conflict between write and compare match processes of CMCNT

22.5.3 Conflict between Write and Count-Up Processes of CMCNT

If count-up occurs during the write access to the CMCNT counter, that writing has priority over the count-up. Figure 22.6 shows the timing to write the CMCNT counter.

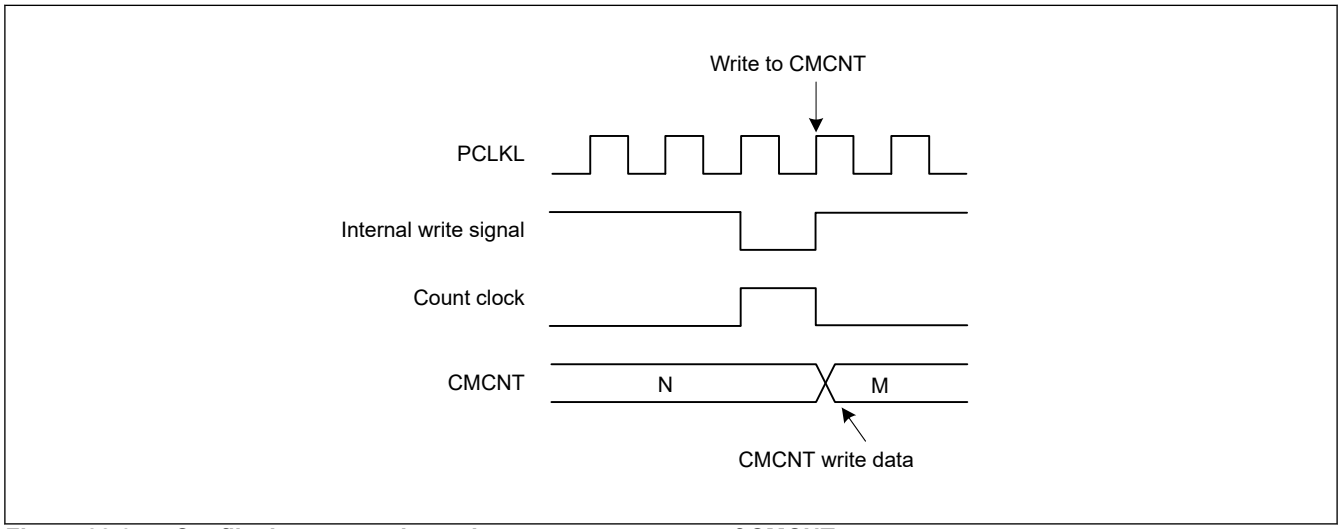


Figure 22.6 Conflict between write and count-up processes of CMCNT

23. Compare Match Timer W (CMTW)

23.1 Overview

The LSI has two channels (1 channel × 2 units) of 32-bit compare match timer W (CMTW). CMTW has a 32-bit counter and can generate interrupts each time a set period elapses.

Table 23.1 shows the specifications of the CMTW and Figure 23.1 shows a block diagram of the CMTW.

Table 23.1 CMTW specifications

Item	Description
Number of channel	Two channels (1 channel × 2 units)
Timer counter	16-bit/32-bit selectable up-counter. The counter returns to 0x00000000 after a compare match.
Prescaler	Four dividing clocks are output. Selectable from any of PCLKL/8, PCLKL/32, PCLKL/128, or PCLKL/512.
Input capture	Up to two input capture input signals available per unit
Output compare	Up to two output compare output signals available per unit
Compare match	One compare match available (no output compare output pin used)
Interrupt	Compare match interrupt Input capture 0 and 1 interrupts per unit Output compare 0 and 1 interrupts per unit
Module-stop function	Module-stop state can be set to reduce power consumption

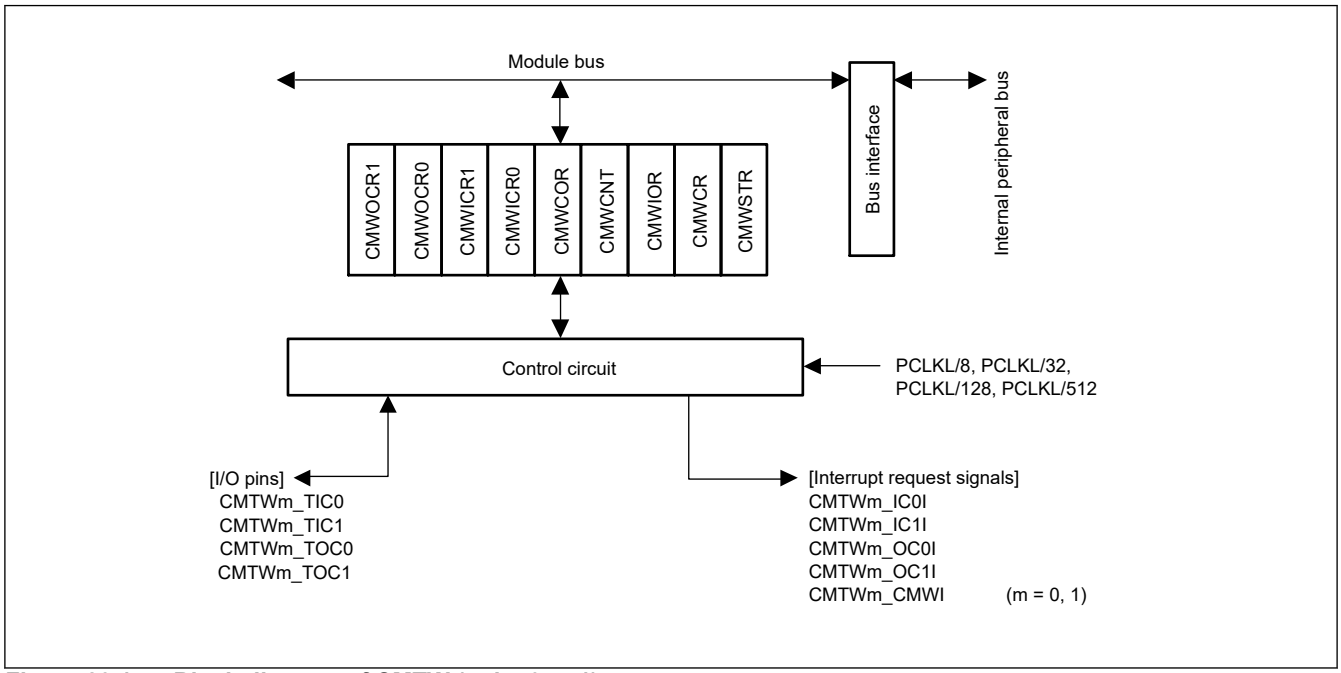


Figure 23.1 Block diagram of CMTW (units 0 to 1)

Table 23.2 shows the CMTW pin configuration.

Table 23.2 Input/Output pins of CMTW

Unit	Pin name	I/O	Description
CMTWm (m = 0, 1)	CMTWm_TIC0	Input	CMTWm input capture input 0
	CMTWm_TIC1	Input	CMTWm input capture input 1
	CMTWm_TOC0	Output	CMTWm output compare output 0
	CMTWm_TOC1	Output	CMTWm output compare output 1

Note: For convention, pin names within this chapter are written without "CMTWm_" (m = 0, 1) notation except this table and the block diagram.

23.2 Register Descriptions

23.2.1 CMWSTR : Timer Start Register

Base address: CMTWm = 0x8004_1000 + 0x0400 × m (m = 0, 1)

Offset address: 0x000

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STR	Counter Start 0: The CMWCNT counter stops counting. The value immediately before a stop of counting is retained and counting is stopped. 1: The CMWCNT counter starts counting.	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W

The CMWSTR register is used to start or stop the CMWCNT counter.

STR bit (Counter Start)

Specifies whether the timer counter operates or is stopped. The relevant prescaler operates or is stopped according to the settings of STR bit.

23.2.2 CMWCR : Timer Control Register

Base address: CMTWm = 0x8004_1000 + 0x0400 × m (m = 0, 1)

Offset address: 0x004

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CCLR[2:0]			—	—	—	CMS	—	OC1IE	OC0IE	IC1IE	IC0IE	CMWIE	—	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CKS[1:0]	Clock Select 0 0: PCLKL/8 0 1: PCLKL/32 1 0: PCLKL/128 1 1: PCLKL/512	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	CMWIE	Compare Match Interrupt Enable 0: Disables a compare match interrupt request (CMTWm_CMWI). 1: Enables a compare match interrupt request (CMTWm_CMWI).	R/W
4	IC0IE	Input Capture 0 Interrupt Enable 0: Disables an interrupt request by the input capture 0 (CMTWm_IC0I). 1: Enables an interrupt request by the input capture 0 (CMTWm_IC0I).	R/W
5	IC1IE	Input Capture 1 Interrupt Enable 0: Disables an interrupt request by the input capture 1 (CMTWm_IC1I). 1: Enables an interrupt request by the input capture 1 (CMTWm_IC1I).	R/W

Bit	Symbol	Function	R/W
6	OC0IE	Output Compare 0 Interrupt Enable 0: Disables an interrupt request by the output compare 0 (CMTWm_OC0I). 1: Enables an interrupt request by the output compare 0 (CMTWm_OC0I).	R/W
7	OC1IE	Output Compare 1 Interrupt Enable 0: Disables an interrupt request by the output compare 1 (CMTWm_OC1I). 1: Enables an interrupt request by the output compare 1 (CMTWm_OC1I).	R/W
8	—	This bit is read as 0. The write value should be 0.	R/W
9	CMS	Timer Counter Size 0: 32 bits 1: 16 bits	R/W
12:10	—	These bits are read as 0. The write value should be 0.	R/W
15:13	CCLR[2:0]	Counter Clear 0 0 0: The CMWCNT counter is cleared by CMWCOR register compare match. 0 0 1: The CMWCNT counter is not cleared. 0 1 0: The CMWCNT counter is not cleared. 0 1 1: The CMWCNT counter is not cleared. 1 0 0: The CMWCNT counter is cleared by CMWICR0 register input capture. 1 0 1: The CMWCNT counter is cleared by CMWICR1 register input capture. 1 1 0: The CMWCNT counter is cleared by CMWOCR0 register compare match. 1 1 1: The CMWCNT counter is cleared by CMWOCR1 register compare match.	R/W

The CMWCR register selects the counter clearing source and the counter input clock, and enables or disables interrupts.

The CMWCR register should be set while the timer counter (CMWCNT) operation is stopped.

CKS[1:0] bits (Clock Select)

Select the clock to be input to the CMWCNT counter among four internal clocks obtained by dividing the peripheral clock (PCLKL). When the CMWSTR.STR bit is set to 1, the CMWCNT counter starts counting up based on the clock selected with the CMWCR.CKS[1:0] bits.

CMWIE bit (Compare Match Interrupt Enable)

Enables or disables compare match interrupt (CMTWm_CMWI) request generation when the CMWCNT counter and CMWCOR register values match.

IC0IE bit (Input Capture 0 Interrupt Enable)

Enables or disables input capture interrupt 0 (CMWm_IC0I) request generation when input capture is generated in the CMWICR0 register.

IC1IE bit (Input Capture 1 Interrupt Enable)

Enables or disables input capture interrupt 1 (CMWm_IC1I) request generation when input capture is generated in the CMWICR1 register.

OC0IE bit (Output Compare 0 Interrupt Enable)

Enables or disables compare match interrupt 0 (CMWm_OC0I) request generation when the CMWCNT counter and CMWOCR0 register values match.

OC1IE bit (Output Compare 1 Interrupt Enable)

Enables or disables compare match interrupt 1 (CMWm_OC1I) request generation when the CMWCNT counter and CMWOCR1 register values match.

CMS bit (Timer Counter Size)

Selects either 16 or 32 bits as the size of the timer counter (CMWCNT). The size selected with the CMS bit is valid in the compare match constant register (CMWCOR), input capture registers (CMWICR0 and CMWICR1), and output compare registers (CMWOCR0 and CMWOCR1).

CCLR[2:0] bits (Counter Clear)

Selects the CMWCNT counter clearing source.

23.2.3 CMWIOR : Timer I/O Control Register

Base address: CMTWm = 0x8004_1000 + 0x0400 × m (m = 0, 1)

Offset address: 0x008

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMWE	—	OC1E	OC0E	OC1[1:0]		OC0[1:0]		—	—	IC1E	IC0E	IC1[1:0]		IC0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	IC0[1:0]	Input Capture Control 0 0 0: Input capture at the rising edge on the TIC0 pin. 0 1: Input capture at the falling edge on the TIC0 pin. 1 0: Input capture at both edges on the TIC0 pin. 1 1: Setting prohibited	R/W
3:2	IC1[1:0]	Input Capture Control 1 0 0: Input capture at the rising edge on the TIC1 pin. 0 1: Input capture at the falling edge on the TIC1 pin. 1 0: Input capture at both edges on the TIC1 pin. 1 1: Setting prohibited	R/W
4	IC0E	Input Capture Enable 0 0: Disables the input capture operation of the CMWICR0 register. 1: Enables the input capture operation of the CMWICR0 register.	R/W
5	IC1E	Input Capture Enable 1 0: Disables the input capture operation of the CMWICR1 register. 1: Enables the input capture operation of the CMWICR1 register.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
9:8	OC0[1:0]	Output Compare Control 0 0 0: Retains the output value.*1 0 1: Initially outputs 0 and toggles the output value upon compare match. 1 0: Initially outputs 1 and toggles the output value upon compare match. 1 1: Setting prohibited	R/W
11:10	OC1[1:0]	Output Compare Control 1 0 0: Retains the output value.*1 0 1: Initially outputs 0 and toggles the output value upon compare match. 1 0: Initially outputs 1 and toggles the output value upon compare match. 1 1: Setting prohibited	R/W
12	OC0E	Compare Match Enable 0 0: Disables the compare match operation using the CMWOCR0 register. 1: Enables the compare match operation using the CMWOCR0 register.	R/W
13	OC1E	Compare Match Enable 1 0: Disables the compare match operation using the CMWOCR1 register. 1: Enables the compare match operation using the CMWOCR1 register.	R/W
14	—	This bit is read as 0. The write value should be 0.	R/W
15	CMWE	Compare Match Enable 0: Disables the compare match operation using the CMWCOR register. 1: Enables the compare match operation using the CMWCOR register.	R/W

Note 1. After reset, 0 is output until the CMWIOR register is set.

The CMWIOR register controls the CMWCOR, CMWICR0, CMWICR1, CMWOCR0, and CMWOCR1 registers.

CMWIOR should be set while the timer counter (CMWCNT) operation is stopped.

IC0[1:0] bits (Input Capture Control 0)

Selects the input capture operation of the CMWICR0 register.

IC1[1:0] bits (Input Capture Control 1)

Selects the input capture operation of the CMWICR1 register.

IC0E bit (Input Capture Enable 0)

Enables or disables the input capture operation of the CMWICR0 register.

IC1E bit (Input Capture Enable 1)

Enables or disables the input capture operation of the CMWICR1 register.

OC0[1:0] bits (Output Compare Control 0)

Sets the output compare operation using the CMWOCR0 register.

OC1[1:0] bits (Output Compare Control 1)

Sets the output compare operation using the CMWOCR1 register.

OC0E bit (Compare Match Enable 0)

Enables or disables the compare match operation using the CMWOCR0 register.

OC1E bit (Compare Match Enable 1)

Enables or disables the compare match operation using the CMWOCR1 register.

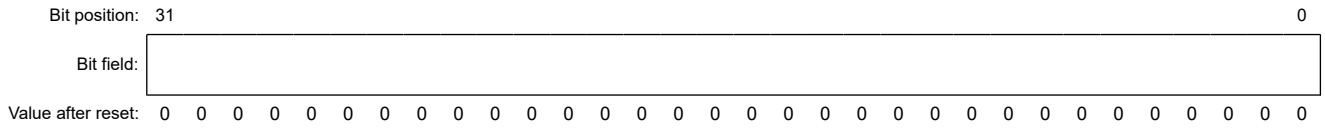
CMWE bit (Compare Match Enable)

Enables or disables the compare match operation using the CMWCOR register.

23.2.4 CMWCNT : Timer Counter

Base address: CMTWm = 0x8004_1000 + 0x0400 × m (m = 0, 1)

Offset address: 0x010



Bit	Symbol	Function	R/W
31:0	n/a	Timer Counter	R/W

The CMWCNT counter is used as a readable/writable up-counter.

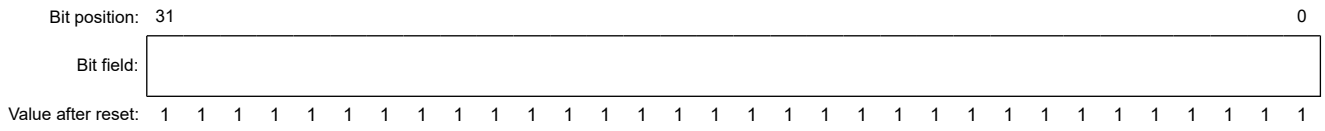
Before starting counter operation, the timer control register (CMWCR) should be set. When the 16-bit counter size is selected with the CMS bit in the timer control register (CMWCR), bits 15 to 0 in the CMWCNT counter are valid. Since access to this register is in 32-bit units, when writing, write 32-bit values to it with 0x0000 in the higher-order bits. If a value other than 0x0000 is set in the higher-order bits, a value greater than 0x0000FFFF may be read when this register is read.

When the STR bit is set to 1, the CMWCNT counter starts counting. When the STR bit is set to 0, the CMWCNT counter retains the value immediately before a stop of counting and stops counting.

23.2.5 CMWCOR : Compare Match Constant Register

Base address: CMTWm = 0x8004_1000 + 0x0400 × m (m = 0, 1)

Offset address: 0x014



Bit	Symbol	Function	R/W
31:0	n/a	The CMWCOR register is a readable/writable register that specifies the time up to a compare match between the timer counter (CMWCNT) value and CMWCOR value. When the 16-bit counter size is selected with the CMS bit in the timer control register (CMWCR), bit 15 to bit 0 in this register are valid. Since access to this register is in 32-bit units, when writing, write 32-bit values to it with 0x0000 in the higher-order bits.	R/W

The cycle for compare matches is as follows:

Compare-match cycle = (setting of the CMWCOR register + 1) × counter-clock cycle[*1]

Note 1. This is a clock cycle set by the CMWCR.CKS[1:0] bits.

23.2.6 CMWICRn : Input Capture Registers (n = 0, 1)

Base address: CMTWm = 0x8004_1000 + 0x0400 × m (m = 0, 1)

Offset address: 0x018 + 0x004 × n

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	The CMWICR0 and CMWICR1 registers are read-only registers in which the CMWCNT counter value is stored when an input capture is generated. When the 16-bit counter size is selected with the CMS bit in the timer control register (CMWCR), bits 15 to 0 in these registers are valid. Writing to these registers is invalid.	R

23.2.7 CMWOCRn : Output Compare Registers (n = 0, 1)

Base address: CMTWm = 0x8004_1000 + 0x0400 × m (m = 0, 1)

Offset address: 0x020 + 0x004 × (n - 1)

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	n/a	The CMWOCR0 and CMWOCR1 registers are readable/writable registers that set the value to be compared when an output compare is generated. When the 16-bit counter size is selected with the CMS bit in the timer control register (CMWCR), bits 15 to 0 of these registers become valid. Since access to this register is in 32-bit units, when writing, write 32-bit values to it with 0x0000 in the higher-order bits.	R/W

23.3 Operation

When the CMWCR register is set and then the STR bit in CMWCR is set to 1, the CMTW starts counter operation.

Setting the CMWSTR.STR bit to 0 enables the CMWCNT counter to retain the value immediately before a stop of counting and stop counting. Setting CMWIOR register enables using the compare match function, input capture input function, and output compare output function.

23.3.1 Period Counting Operation

When the internal clock is selected by using the CMWCNT.CKS[1:0] bits and the CMWSTR.STR bit is set to 1, the CMWCNT counter starts counting up cycles of the selected clock. When the CCLR[2:0] bits in CMWCR are set so that CMWCNT should be cleared by a specific counter clearing source and the counter clearing source is generated, the

CMWCNT counter is cleared to 0x00000000 and continues incrementing. When the CCLR[2:0] bits are set so that CMWCNT should not be cleared by any specific counter clearing source, the CMWCNT counter is cleared to 0x00000000 only when an overflow is generated (0xFFFFFFFF → 0x00000000 (when the counter size is 32 bits) or 0x0000FFFF → 0x00000000 (when the counter size is 16 bits)) and continues incrementing.

23.3.2 Compare Match Function

When the values of the CMWCNT counter and CMWCOR register match, a compare match interrupt (CMTWm_CMWI) is generated. The CMWCNT counter operates as follows according to the setting of CMWCR.CCLR[1:0].

1. When CMWCR.CCLR[2:0] = 000b
   When the values of the CMWCNT counter and CMWCOR register match, the CMWCNT counter is cleared to 0x00000000. The CMWCNT counter then restarts counting up from 0x00000000.
2. When CMWCR.CCLR[2:0] ≠ 000b
   Even when the values of the CMWCNT counter and CMWCOR register match, the CMWCNT counter is not cleared to 0x00000000 but continues counting up until the clearing condition set in CMWCR.CCLR[1:0] is satisfied or the value of the counter reaches 0xFFFFFFFF (when the size of the counter is 32 bits) or 0x0000FFFF (when the size of the counter is 16 bits).
   The CMWCNT counter is then cleared to 0x00000000 and restarts counting up from 0x00000000.

Figure 23.2 shows an example of procedure for setting compare match operation.

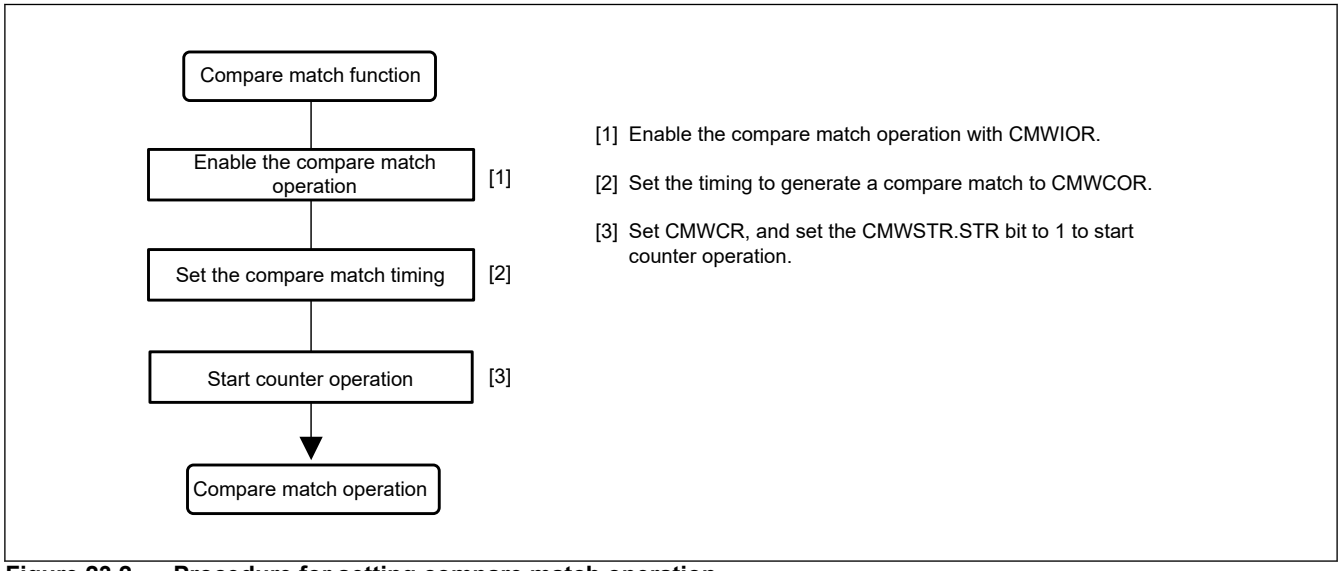


Figure 23.2 Procedure for setting compare match operation

Figure 23.3 shows an example when compare match with CMWCOR is set as a counter clearing source.

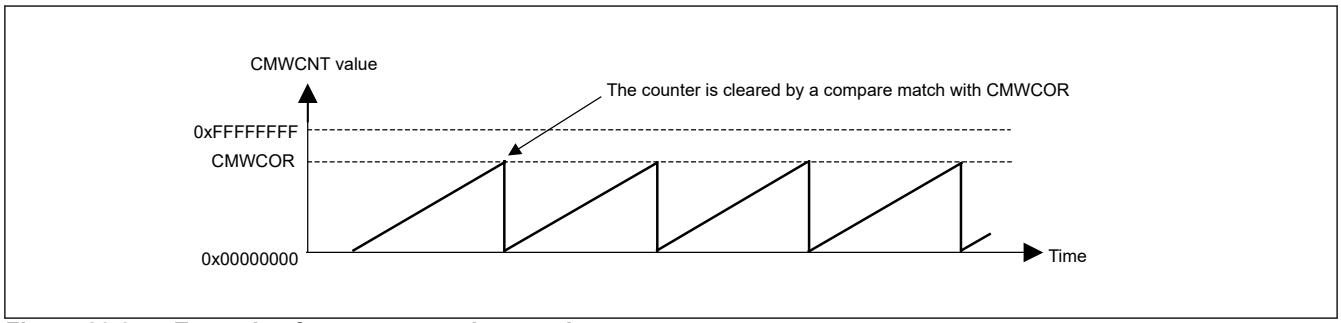


Figure 23.3 Example of compare match operation

Figure 23.4 shows an example when CMWCOR is set to 0xFFFFFFFF and detected as overflow.

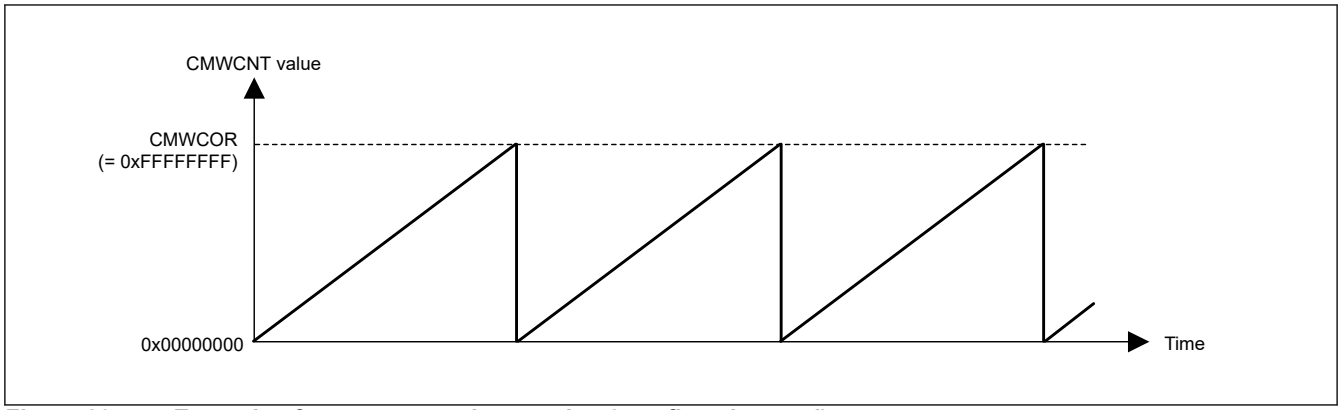


Figure 23.4 Example of compare match operation (overflow detected)

23.3.3 Output Compare Function

Using the output compare function, toggle output from the relevant output pins can be provided. When the CMWCNT counter value matches either of the values of CMWOCR0 or CMWOCR1 register, the output compare interrupt (CMTWm_OC0I or CMTWm_OC1I (m = 0, 1)) is generated. The CMWCNT counter operates as follows according to the setting of CMWCR.CCLR[2:0].

1. When CMWCR.CCLR[2:0] = 110b
   When the values of the CMWCNT counter and CMWOCR0 register match, the CMWCNT counter is cleared to 0x00000000. The CMWCNT counter then restarts counting up from 0x00000000.
2. When CMWCR.CCLR[2:0] = 111b
   When the values of the CMWCNT counter and CMWOCR1 register match, the CMWCNT counter is cleared to 0x00000000. The CMWCNT counter then restarts counting up from 0x00000000.
3. When CMWCR.CCLR[2:0] = others except 110b and 111b
   Even when the values of the CMWCNT counter and CMWOCR0 or CMWOCR1 register match, the CMWCNT counter is not cleared to 0x00000000 but continues counting up until the clearing condition set in CMWCR.CCLR[2:0] is satisfied or the value of the counter reaches 0xFFFFFFFF (when the size of the counter is 32 bits) or 0x0000FFFF (when the size of the counter is 16 bits). The CMWCNT counter is then cleared to 0x00000000 and restarts counting up from 0x00000000.

Figure 23.5 shows an example of procedure for setting output compare operation.

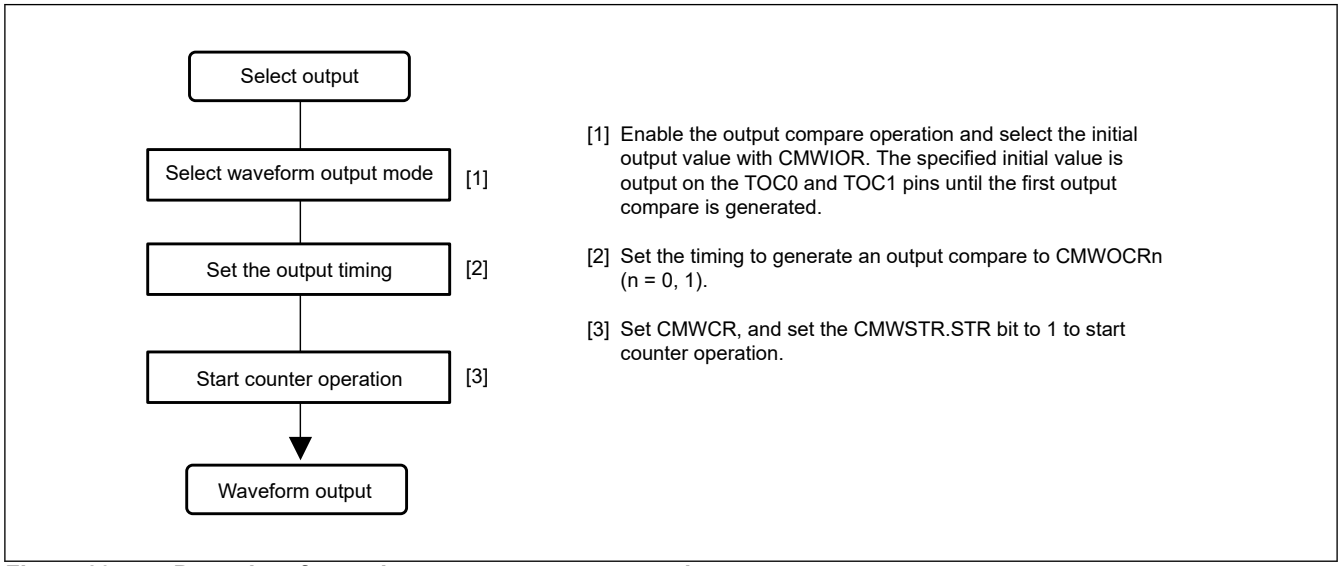


Figure 23.5 Procedure for setting output compare operation

Figure 23.6 shows an example when the counter is cleared upon compare match with CMWOCR1 register and toggle outputs are provided from the TOC0 and TOC1 pins.

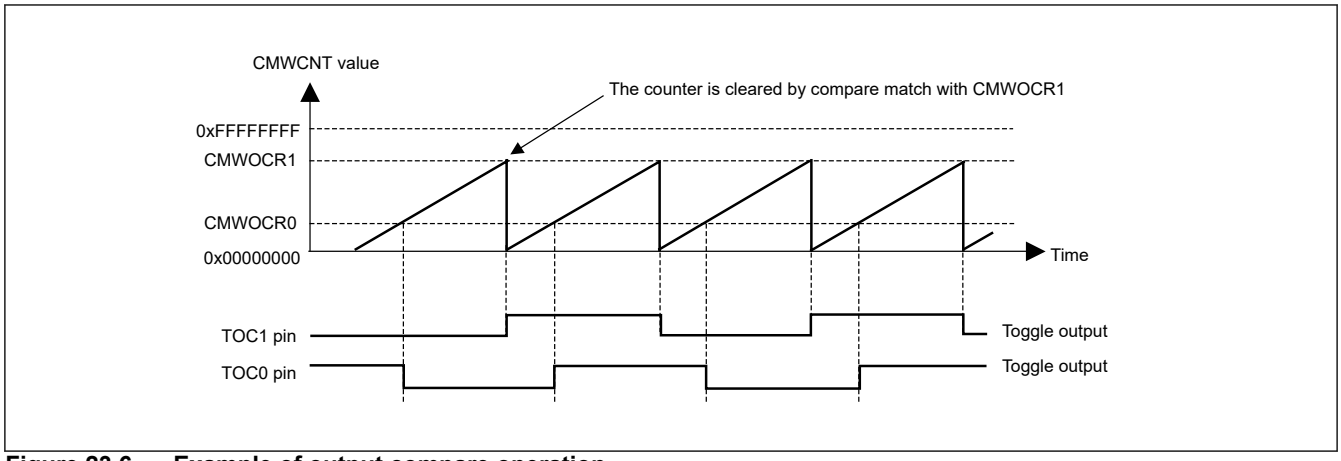


Figure 23.6 Example of output compare operation

23.3.4 Input Capture Function

Through detecting the edge on the TIC0 and TIC1 pin input, the CMWCNT counter value can be transferred to CMWICR0 and CMWICR1 registers, respectively. The edges to be detected can be selected from among the rising edge alone, falling edge alone, and both the rising and falling edges. When the CMWCNT counter value is transferred to CMWICR0 or CMWICR1 register using the input capture function, an input capture interrupt (CMTWm_IC0I or CMTWm_IC1I (m = 0, 1)) is generated. The CMWCNT counter operates as follows according to the setting of CMWCR.CCLR[2:0].

1. When CMWCR.CCLR[2:0] = 100b
   When the CMWCNT counter value is transferred to CMWICR0 using the input capture operation, the CMWCNT counter is cleared to 0x00000000.
   The CMWCNT counter then restarts counting up from 0x00000000.
2. When CMWCR.CCLR[2:0] = 101b
   When the CMWCNT counter value is transferred to CMWICR1 using the input capture operation, the CMWCNT counter is cleared to 0x00000000.
   The CMWCNT counter then restarts counting up from 0x0000000.
3. When CMWCR.CCLR[2:0] = others except 110b and 111b
   Even when the CMWCNT counter value is transferred to CMWICR0 or CMWICR1 using the input capture operation, the CMWCNT counter is not cleared to 0x00000000 but continues counting up until the clearing condition set in CMWCR.CCLR[2:0] is satisfied or the value of the counter reaches 0xFFFFFFFF (when the size of the counter is 32 bits) or 0x0000FFFF (when the size of the counter is 16 bits). The CMWCNT counter is then cleared to 0x00000000 and restarts counting up from 0x00000000.

Figure 23.7 shows an example of procedure for setting input capture operation.

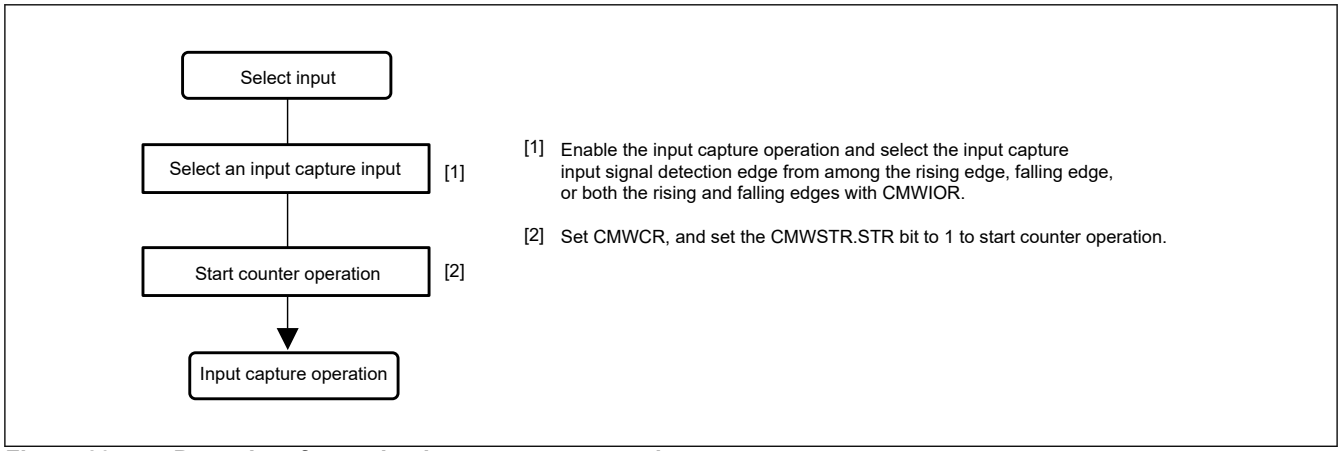


Figure 23.7 Procedure for setting input capture operation

Figure 23.8 shows an example in which both the rising and falling edges are selected for the TIC0 pin input capture input edge and the falling edge for the TIC1 pin, and the CMWCNT counter is cleared by a CMWICR1 register input capture.

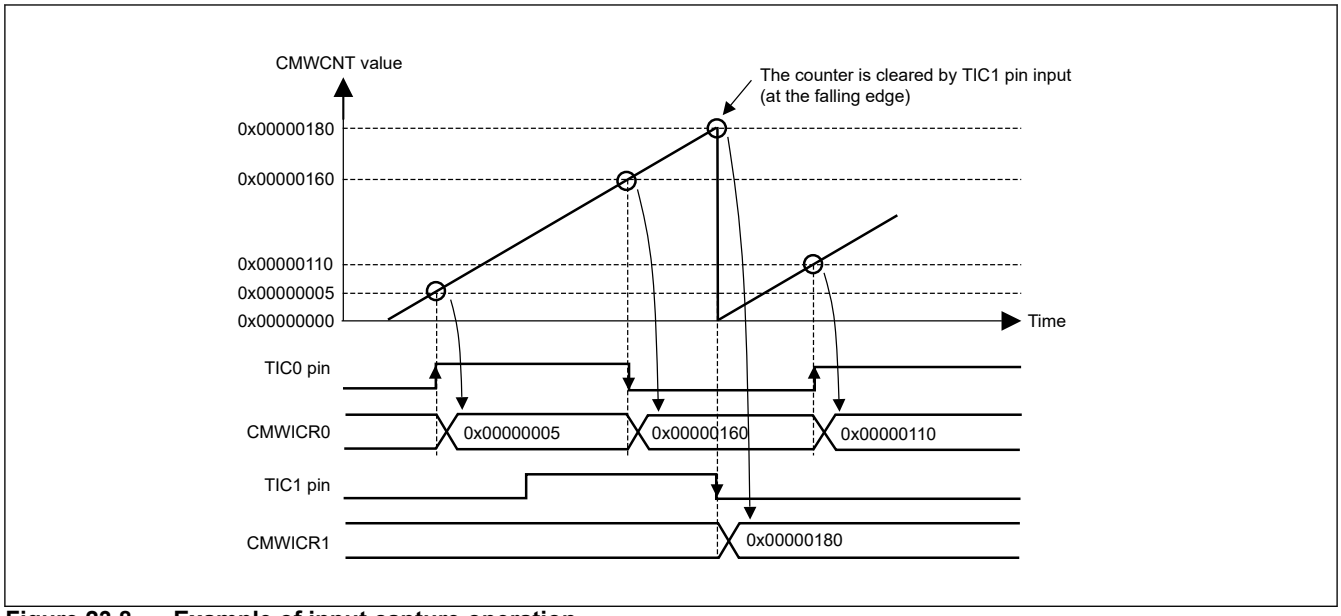


Figure 23.8 Example of input capture operation

23.3.5 Counter Size

With the CMTW, either 16 or 32 bits can be selected as the counter size by using the CMWCR.CMS bit. When the counter is used as a 32-bit counter, set the CMWCOR, CMWOCR0, or CMWOCR1 register to the desired values in 32-bit units. In reading, all 32 bits of CMWICR0 and CMWICR1 are valid. When the counter is used as a 16-bit counter, a 32-bit value should be set to the CMWCOR register with 0x0000 in the higher-order bits. Similarly, a 32-bit value should be set to CMWOCR0 and CMWOCR1 registers with 0x0000 in the higher-order bits.

A 32-bit value with 0x0000 in the higher-order bits is read from CMWICR0 and CMWICR1 registers.

23.3.6 Count Timing Based on CMWCNT

One of four clocks (PCLKL/8, PCLKL/32, PCLKL/128, and PCLKL/512) obtained by dividing the peripheral clock (PCLKL) can be selected with the CMWCR.CKS[1:0] bits. Figure 23.9 shows the timing.

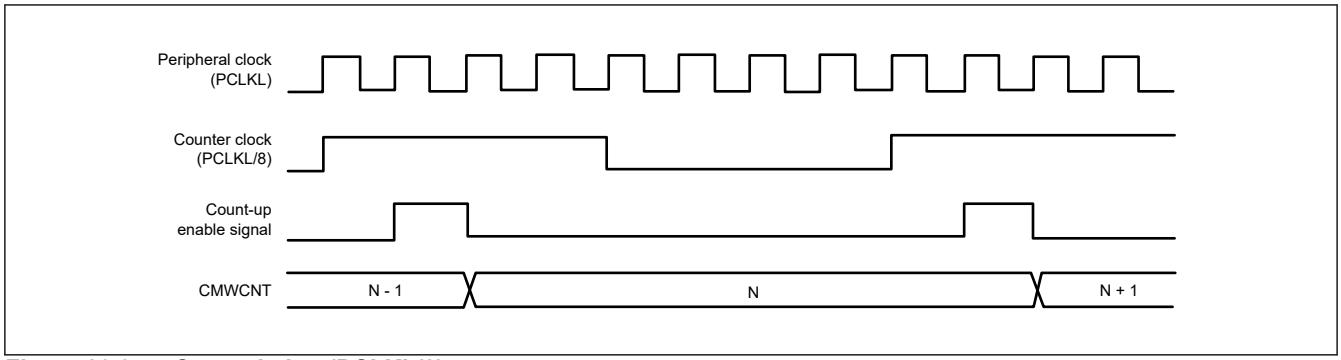


Figure 23.9 Count timing (PCLKL/8)

23.3.7 Output Compare Output Timing

A compare match signal is generated in the last state in which the CMWOCRn (n = 0, 1) register and CMWCNT counter values match (the CMWCNT counter value is updated immediately after the state). That is, the compare match signal is not generated if the CMWCNT counter clock is not input after a match between the CMWOCRn register and CMWCNT counter values. When a compare match signal is generated, the output compare output pin (TOC) changes in accord with the setting of the OC0[1:0] or OC1[1:0] bits in the CMWIOR register. Figure 23.10 shows output compare output timing.

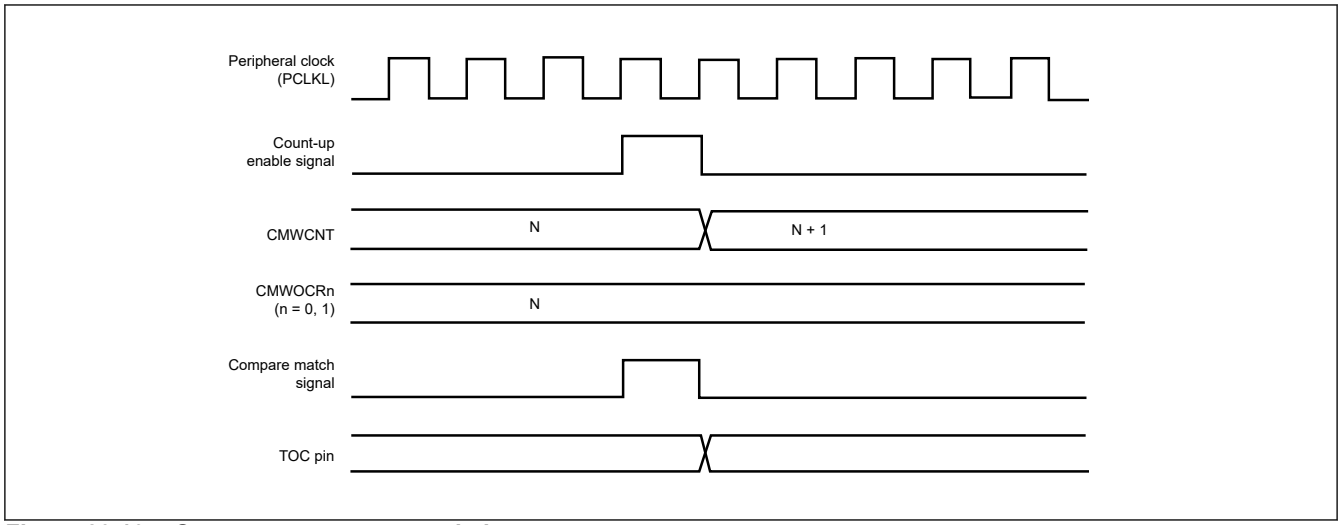


Figure 23.10 Output compare output timing

23.3.8 Input Capture Signal Timing

Figure 23.11 shows the input capture timing.

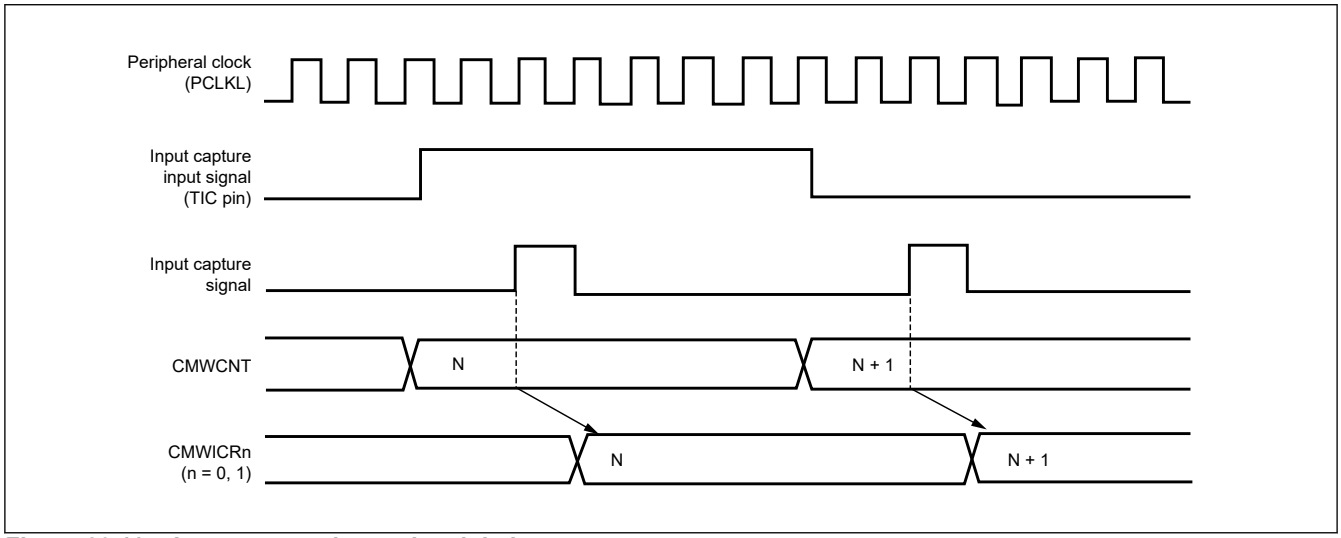


Figure 23.11 Input capture input signal timing

23.4 Interrupts

23.4.1 CMTW Interrupt Sources and DMAC Transfer Requests

The CMTW has five interrupt sources: two input capture interrupt requests (CMTWm_IC0I and CMTWm_IC1I), two output compare interrupt requests (CMTWm_OC0I and CMTWm_OC1I), and a compare match interrupt request (CMTWm_CMWI) (m = 0, 1).

Table 23.3 shows the interrupt sources and priority. The interrupt sources can be enabled or disabled using the IC0IE, IC1IE, OC0IE, OC1IE, and CMWIE bits in CMWCR and are separately issued to the interrupt controller.

Table 23.3 CMTW interrupt sources

Interrupt source	Interrupt	Interrupt enable bit	CPU0 GIC request	DMAC activation	Priority
CMTWm_CMWI	Interrupt caused by compare match	CMWIE	Possible	Possible	High
CMTWm_IC0I	Interrupt caused by input capture 0	IC0IE	Possible	Possible	↑
CMTWm_IC1I	Interrupt caused by input capture 1	IC1IE	Possible	Possible	↑
CMTWm_OC0I	Interrupt caused by output compare 0	OC0IE	Possible	Possible	↑
CMTWm_OC1I	Interrupt caused by output compare 1	OC1IE	Possible	Possible	Low

23.4.2 Timing of Compare Match Interrupt Generation

When the values of the CMWCNT counter and CMWCOR register match, a compare match interrupt (CMTWm_CMWI (m = 0, 1)) is generated. The compare match signal is generated at the end of the cycle where the values matched (i.e. when the CMWCNT counter is updated from the matching counter value). The compare match signal, therefore, is not generated until a further cycle of the input clock (PCLKL/8, PCLKL/32, PCLKL/128, or PCLKL/512) for the CMWCNT counter arrives after the values of the CMWCNT counter and CMWCOR register have matched. Figure 23.12 shows the timing of compare match interrupt generation.

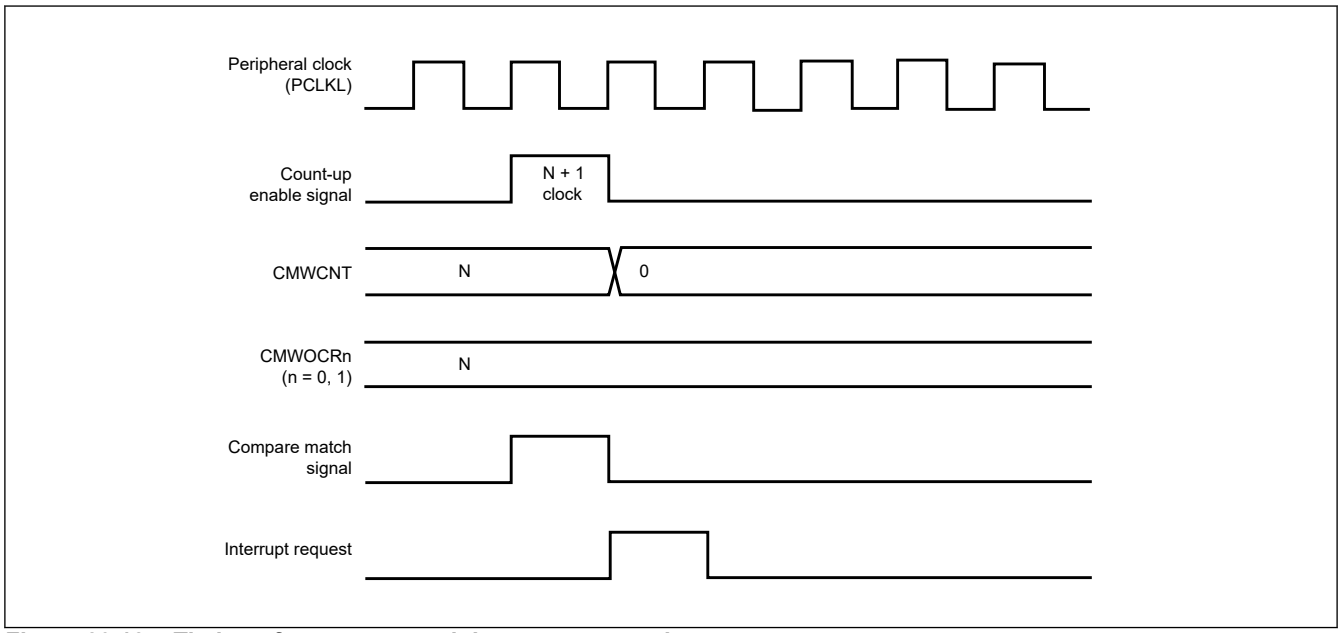


Figure 23.12 Timing of compare match interrupt generation

23.4.3 Timing of Output Compare Interrupt Generation

Figure 23.13 shows the timing of output compare interrupt generation.

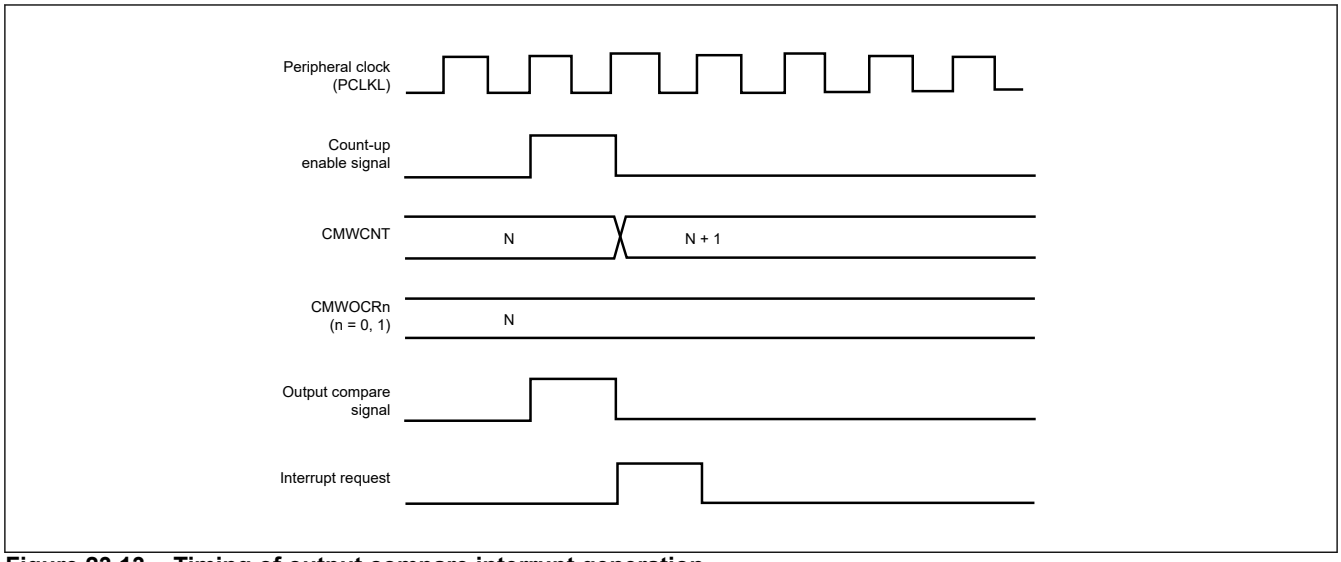


Figure 23.13 Timing of output compare interrupt generation

23.4.4 Timing of Input Capture Interrupt Generation

Figure 23.14 shows the timing of input capture interrupt generation.

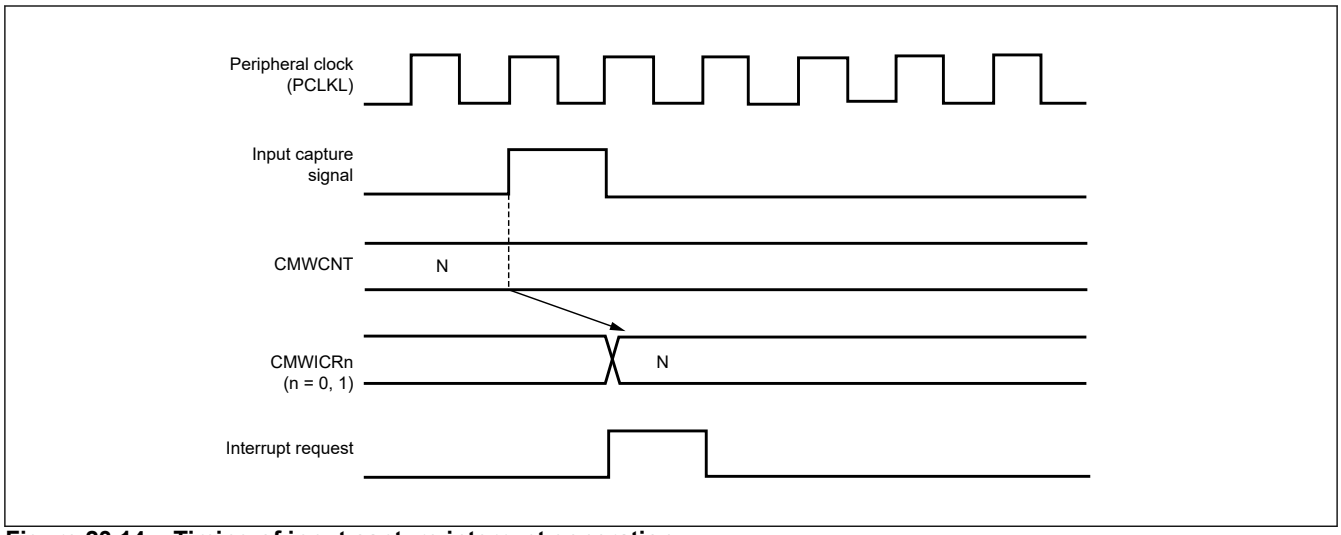


Figure 23.14 Timing of input capture interrupt generation

23.5 Usage Notes

23.5.1 Module-Stop Function

The CMTW operation can be enabled and disabled using the Module-Stop Control Register (MSTPCRD). With the initial value, the CMTW is in module-stop state. Register access is enabled by release from the module-stop state. For details, see section 8, Low-Power Consumption Function.

23.5.2 Contention Between CMWCNT Counter Writing and Compare Match

If the compare match signal is generated during CPU writing to the CMWCNT counter, the compare match interrupt request is output but the counter is not cleared since the CPU writing to the counter is given priority.

Figure 23.15 shows the timing of contention between CMWCNT counter writing and compare match.

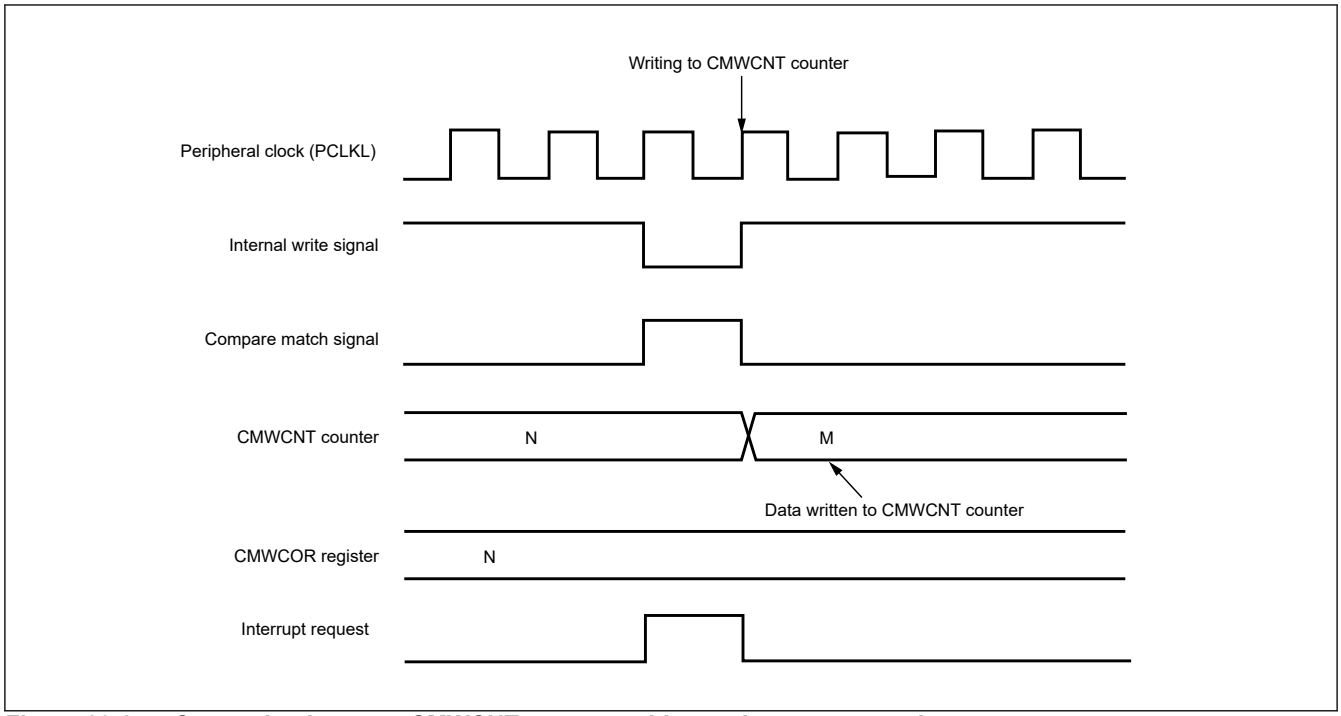


Figure 23.15 Contention between CMWCNT counter writing and compare match

23.5.3 Contention Between CMWCNT Counter Writing and Incrementing or Clearing

In case of contention between incrementation or clearing of the CMWCNT counter and CPU writing to the counter, the counter is not actually incremented or cleared since the CPU writing to the CMWCNT counter is given priority.

Figure 23.16 shows the timing in the case of contention between writing to the CMWCNT counter and incrementation or clearing.

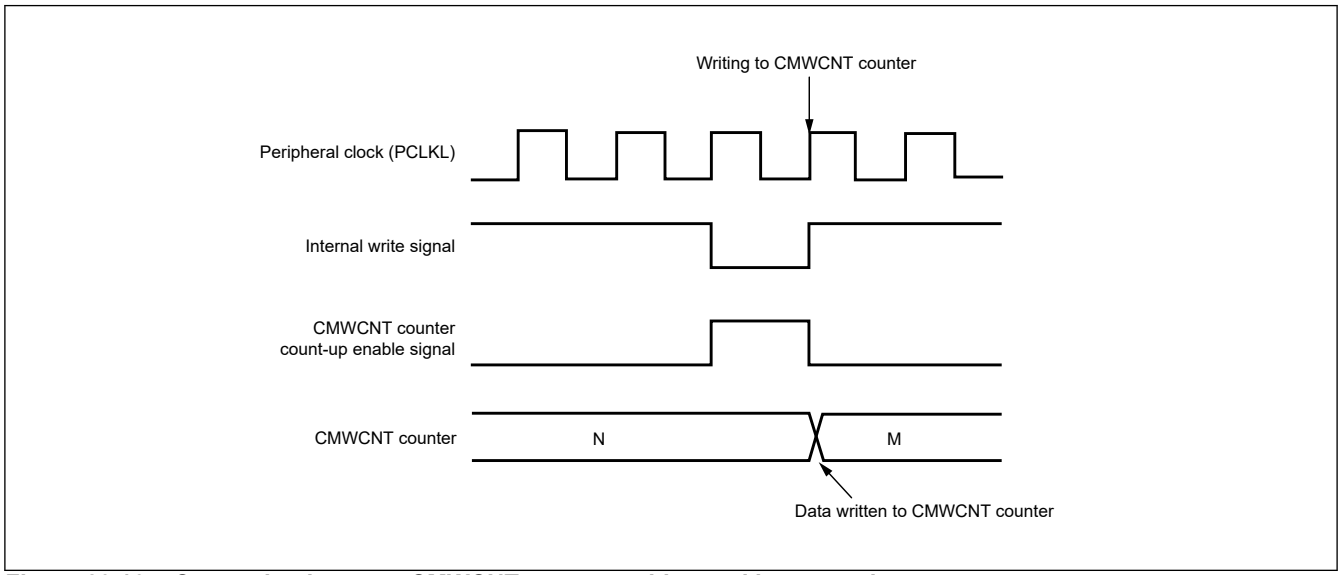


Figure 23.16 Contention between CMWCNT counter writing and incrementing

23.5.4 Contention Between CMWCOR Register Writing and Compare Match

If the compare match is generated during CPU writing to the CMWCOR register, the CPU writing to the CMWCOR register proceeds and also the compare match signal is output.

Figure 23.17 shows the timing of contention between CMWCOR register writing and compare match.

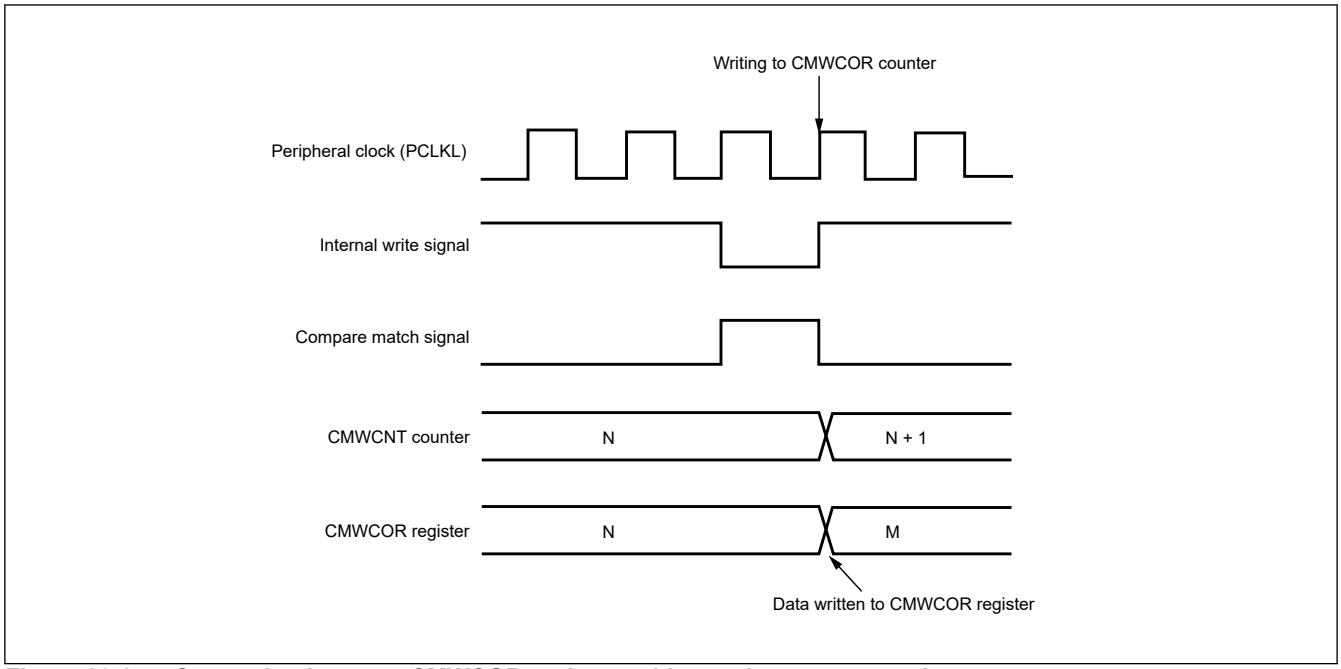


Figure 23.17 Contention between CMWCOR register writing and compare match

23.5.5 Contention Between CMWOCRn Register Writing and Compare Match

If the compare match is generated during CPU writing to the CMWOCRn register, the CPU writing to the CMWOCRn register proceeds and also the compare match signal is output (n = 0, 1).

Figure 23.18 shows the timing of contention between CMWOCRn register writing and compare match.

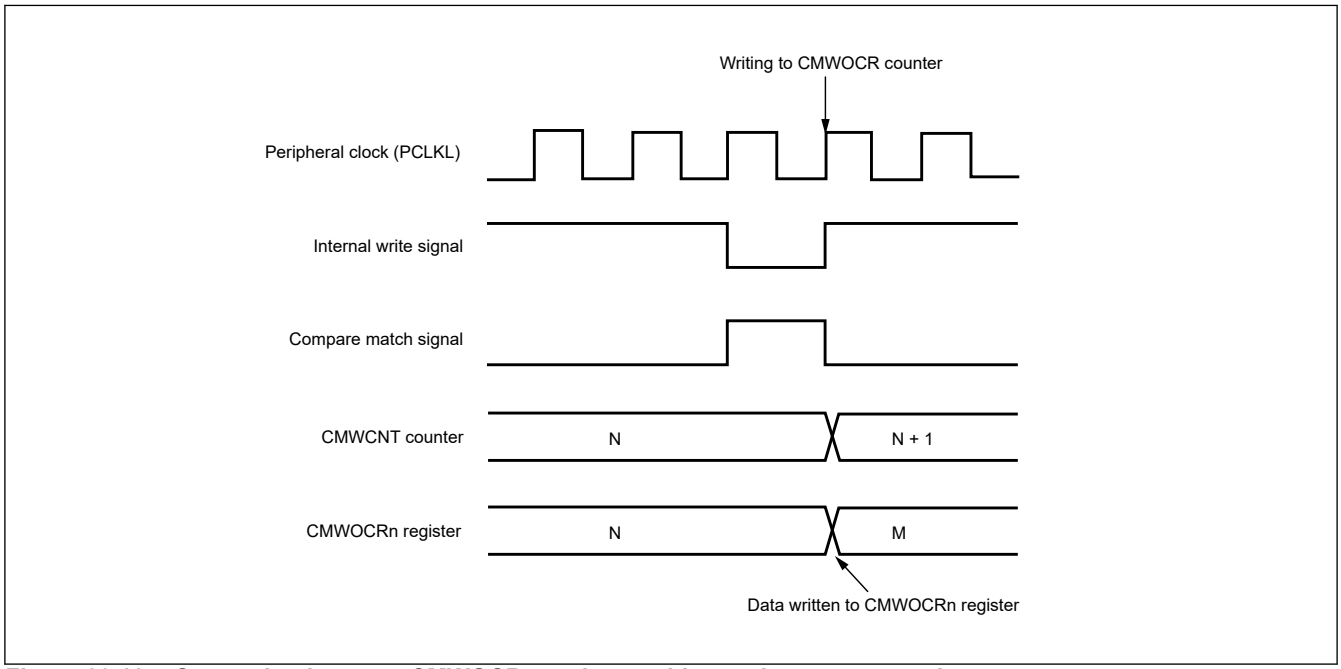


Figure 23.18 Contention between CMWOCRn register writing and compare match

23.5.6 Contention Between CMWCNT Counter Reading and Incrementing or Clearing

If the CMWCNT counter incrementing or clearing process occurs at the same time that the data of the CMWCNT counter is read, the value having been in the CMWCNT counter before incremented or cleared is read.

Figure 23.19 shows the timing of contention between the CMWCNT counter reading and incrementing.

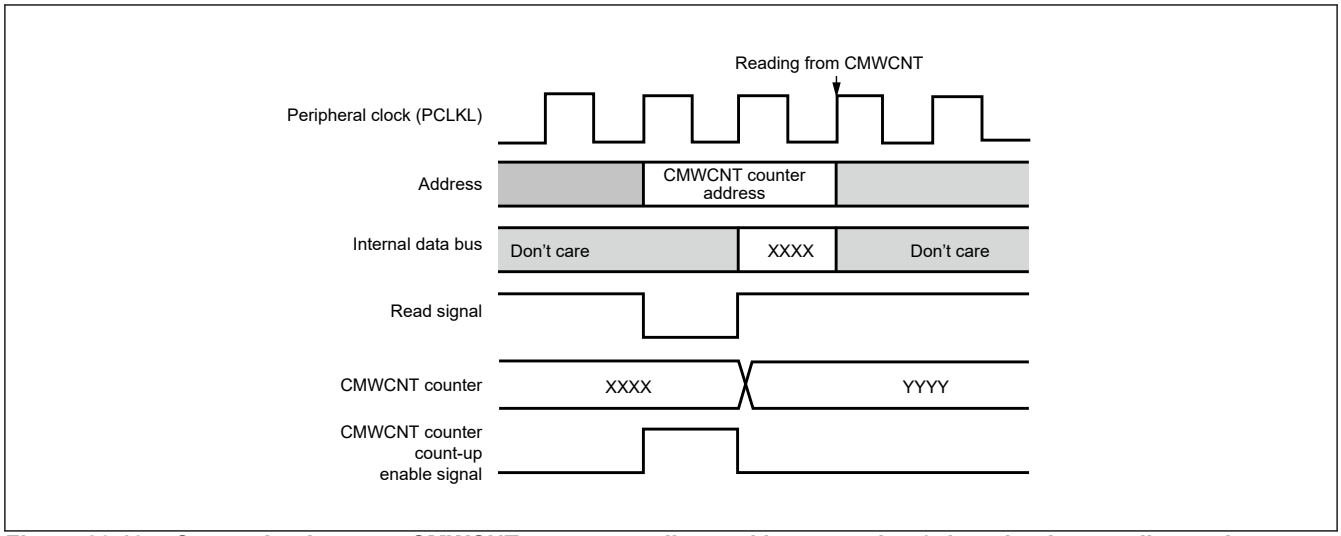


Figure 23.19 Contention between CMWCNT counter reading and incrementing (when the data reading and incrementing process occur simultaneously)

23.5.7 Contention Between CMWICRn Register Reading and Input Capture

If the input capture signal is generated at the same time that the data of CMWICRn register is read, the value having been in CMWICRn register before updated by input capture transfer is read (n = 0, 1).

Figure 23.20 shows the timing of contention between CMWICRn register reading and input capture.

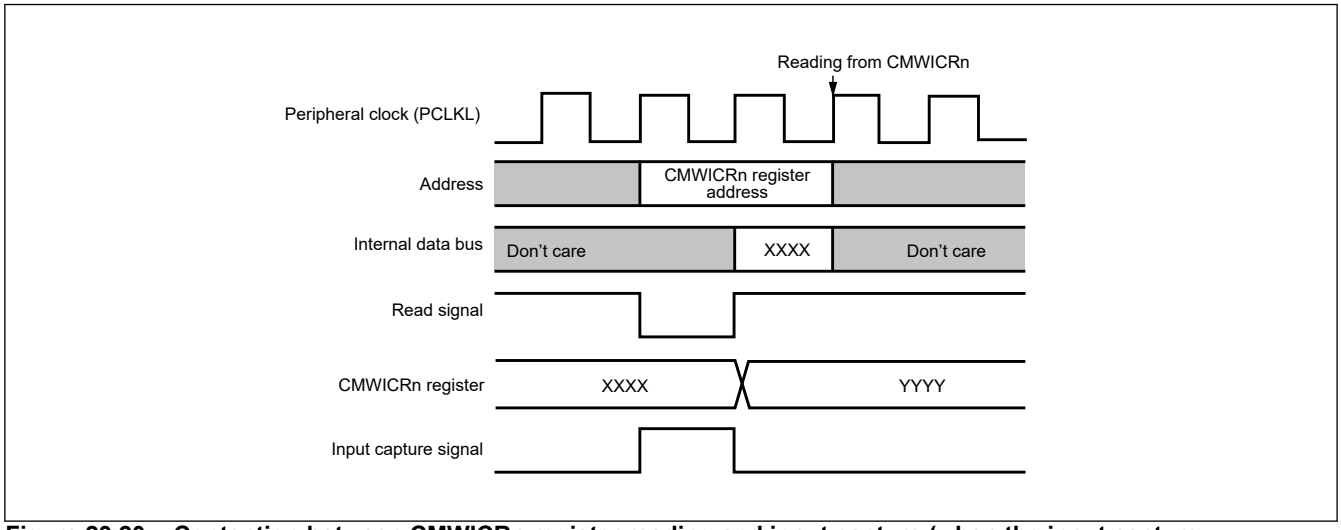


Figure 23.20 Contention between CMWICRn register reading and input capture (when the input capture signal and read signal are generated simultaneously)

Table 23.4 Summary of contention between operations due to the access to registers and changes to the counter's state (1 of 2)

Register access	CMWCNT state	CMWICR0/1 state	Operation to be performed
Writing to CMWCNT	Compare match	—	Output of compare match interrupt request / Writing to CMWCNT
	Counting up	—	Writing to CMWCNT
Writing to CMWCOR	Compare match	—	Compare match
Writing to CMWOCR0	Output compare 0	—	Output compare 0
Writing to CMWOCR1	Output compare 1	—	Output compare 1
Reading from CMWCNT	Counting up	—	Counting up and reading of the previous value

Table 23.4 Summary of contention between operations due to the access to registers and changes to the counter's state (2 of 2)

Register access	CMWCNT state	CMWICR0/1 state	Operation to be performed
Reading from CMWICR0	—	Input capture 0	Input capture 0 and reading of the value before transfer
Reading from CMWICR1	—	Input capture 1	Input capture 1 and reading of the value before transfer

24. Watchdog Timer (WDT)

24.1 Overview

The watchdog timer (WDT) is a 14-bit down-counter and can be used to inform error events to ICU when the counter underflows because the system has run out of control and is unable to refresh the WDT. The refresh-permitted period can be set to refresh the counter and to detect when the system runs out of control.

Table 24.1 lists the specifications of the WDT and Figure 24.1 shows a block diagram of the WDT.

Table 24.1 WDT specifications

Parameter	Specifications
Number of channels	One channel
Count source	Peripheral clock (PCLKL)
Clock division ratio	Divide by 4, 64, 128, 512, 2048, or 8192
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	Register start mode: Counting is started by refresh on a write to the WDTRR register
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values) • A counter underflows or a refresh error occurs • Cortex®-R52 debug state
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
WDT error events	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading the counter value	The down-counter value can be read by the WDTSR register.
Output signal (internal signal)	Interrupt request output

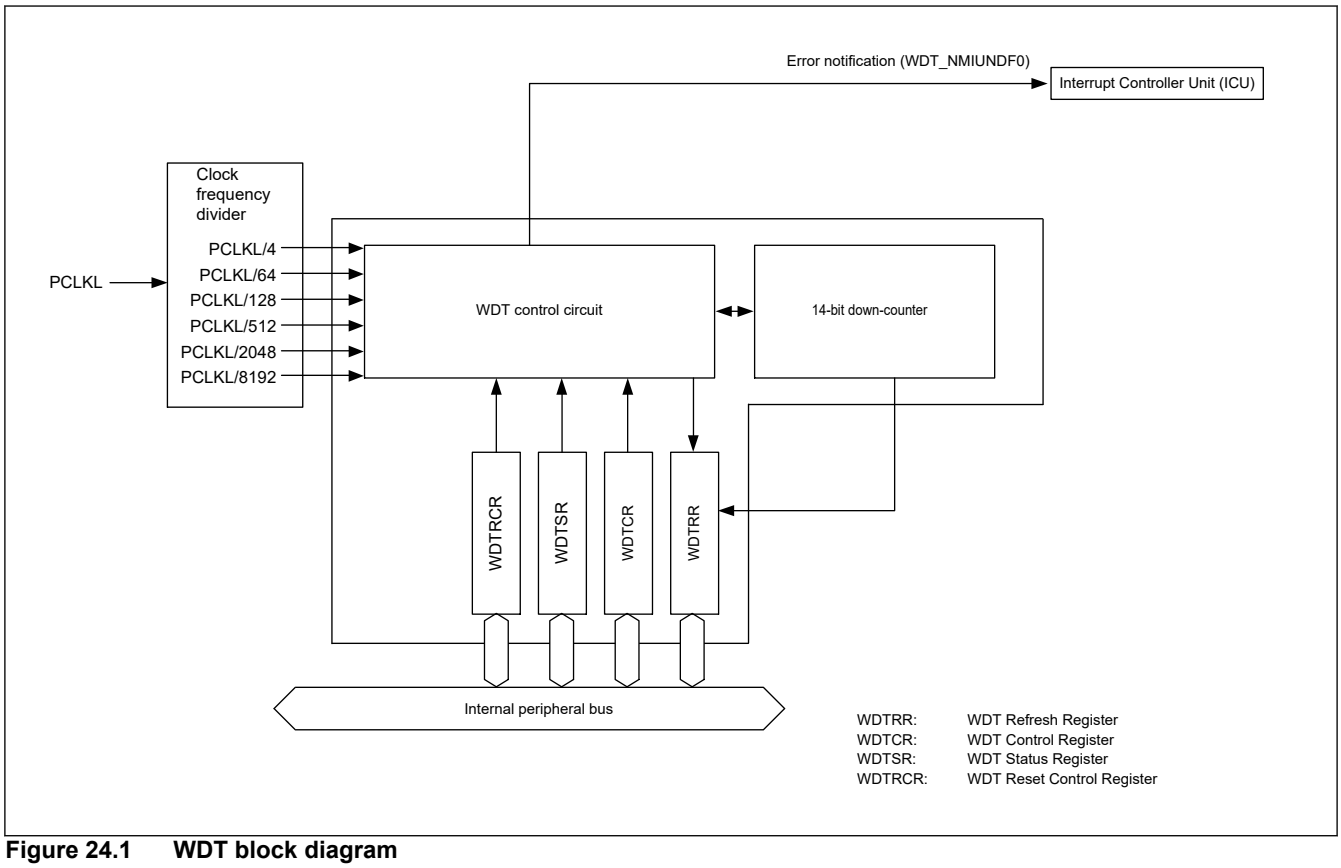


Figure 24.1 WDT block diagram

Table 24.2 WDT interrupt sources

Name	Interrupt sources	CPU0 GIC request	DMAC activation
WDT_NMIUNDF0	Down-counter underflow/Refresh error for CPU0	Possible*1	Not possible

Note 1. Some errors from other peripherals are ORed and connected to the GIC. See section 11, Interrupt Controller (ICU) for detail of error events.

24.2 Register Descriptions

24.2.1 WDTRR : WDT Refresh Register

Base address: WDT0 = 0x8004_2000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:								
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	n/a	The down-counter is refreshed by writing 0x00 and then 0xFF to this register.	R/W

WDTRR register refreshes the down-counter of the WDT.

The down-counter of the WDT is refreshed by writing 0x00 and then 0xFF to WDTRR (refresh operation) within the refresh-permitted period. After being refreshed, the down-counter starts counting down from the value specified with the WDTCR.TOPS[1:0] bits.

When 0x00 is written, the read value is 0x00, when a value other than 0x00 is written, the read value is 0xFF. For details of the refresh operation, see section 24.3.3. Refresh Operation.

24.2.2 WDTCR : WDT Control Register

Base address: WDT0 = 0x8004_2000

Offset address: 0x02

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	RPSS[1:0]		—	—	RPES[1:0]		CKS[3:0]				—	—	TOPS[1:0]	
Value after reset:	0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1

Bit	Symbol	Function	R/W
1:0	TOPS[1:0]	Timeout Period Selection 0 0: 1024 cycles (0x03FF) 0 1: 4096 cycles (0x0FFF) 1 0: 8192 cycles (0x1FFF) 1 1: 16384 cycles (0x3FFF)	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
7:4	CKS[3:0]	Clock Division Ratio Selection 0x1: PCLKL/4 0x4: PCLKL/64 0xF: PCLKL/128 0x6: PCLKL/512 0x7: PCLKL/2048 0x8: PCLKL/8192 Others: Setting prohibited.	R/W

Bit	Symbol	Function	R/W
9:8	RPES[1:0]	Window End Position Selection 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (window end position is not specified.)	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
13:12	RPSS[1:0]	Window Start Position Selection 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (window start position is not specified.)	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

The WDTCR register allows the selecting of a timeout period before the down-counter underflows, a clock division ratio, and a window start/end position for refreshing. There are some restrictions for writing to this register. For details, see section 24.3.2. Control over Writing to the WDTCR and WDTRCR Registers.

TOPS[1:0] bits (Timeout Period Selection)

These bits select a timeout period (period before the down-counter underflows) from among 1024, 4096, 8192, and 16384 cycles, taking the divided clock specified by the CKS[3:0] bits as one cycle.

After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of PCLKL cycles) before the counter underflows.

Table 24.3 lists the relationship between the CKS[3:0] and TOPS[1:0] bit settings, the timeout period, and the number of PCLKL cycles.

CKS[3:0] bits (Clock Division Ratio Selection)

These bits specify the division ratio of the clock used for the down-counter. The division ratio can be selected from among the peripheral module clock (PCLKL) divided by 4, 64, 128, 512, 2048, and 8192. Combined with the TOPS[1:0] bit setting, a count period between 4,096 and 134,217,728 cycles of the PCLKL clock can be selected for the WDT.

RPES[1:0] bits (Window End Position Selection)

These bits allow selection of the window end position of the down counter from among 75%, 50%, 25%, and 0% of the counting period. The selected window end position should be a value smaller than the window start position (window start position > window end position). If the window end position is greater than the window start position, only the window start position setting is valid.

The counter values of the window start and end positions specified by the RPES[1:0] and RPSS[1:0] bit settings vary according to the TOPS[1:0] bit setting.

Table 24.4 lists the counter values of the window start and end positions corresponding to the TOPS[1:0] bit setting.

RPSS[1:0] bits (Window Start Position Selection)

These bits allow the selection of the window start position of the down counter from among 100%, 75%, 50%, and 25% of the counting period (assuming that the count start position as 100% and underflow occurrence as 0%). The period from the window start position to the window end position is the refresh-permitted period, any other periods are the refresh-prohibited periods.

Table 24.4 lists the counter values for the window start and end positions and Figure 24.2 shows the relationship between the settings of the RPSS[1:0] and RPES[1:0] bits, including the refresh-permitted and refresh-prohibited periods.

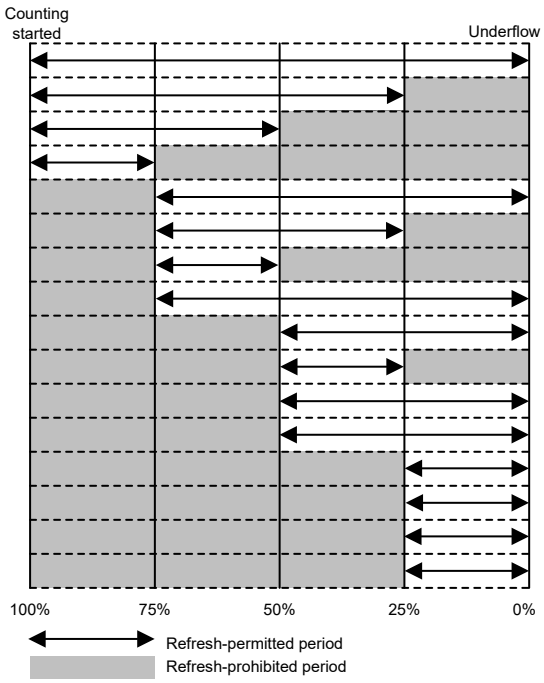
Table 24.3 Timeout period settings

CKS[3:0] bits				TOPS[1:0] bits		Clock division ratio	Timeout period (number of cycles)	No. of Peripheral clock (PCLKL) cycles @Count clock = PCLKL
b7	b6	b5	b4	b1	b0			
0	0	0	1	0	0	Count clock/4	1024	4096
				0	1		4096	16384
				1	0		8192	32768
				1	1		16384	65536
0	1	0	0	0	0	Count clock/64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
1	1	1	1	0	0	Count clock/128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
0	1	1	0	0	0	Count clock/512	1024	524288
				0	1		4096	2097152
				1	0		8192	4194304
				1	1		16384	8388608
0	1	1	1	0	0	Count clock/2048	1024	2097152
				0	1		4096	8388608
				1	0		8192	16777216
				1	1		16384	33554432
1	0	0	0	0	0	Count clock/8192	1024	8388608
				0	1		4096	33554432
				1	0		8192	67108864
				1	1		16384	134217728

Table 24.4 Relationship between timeout period and window start and end counter values

TOPS[1:0] bits		Timeout period		Window start and end counter value			
b1	b0	Cycles	Counter value	100%	75%	50%	25%
0	0	1024	0x03FF	0x03FF	0x02FF	0x01FF	0x00FF
0	1	4096	0x0FFF	0x0FFF	0x0BFF	0x07FF	0x03FF
1	0	8192	0x1FFF	0x1FFF	0x17FF	0x0FFF	0x07FF
1	1	16384	0x3FFF	0x3FFF	0x2FFF	0x1FFF	0x0FFF

RPSS[1:0] bits		RPES[1:0] bits		Window	
b13	b12	b9	b8	Start (%)	End (%)
1	1	1	1	100	0
		1	0		25
		0	1		50
		0	0		75
1	0	1	1	75	0
		1	0		25
		0	1		50
		0	0		75
0	1	1	1	50	0
		1	0		25
		0	1		50
		0	0		75
0	0	1	1	25	0
		1	0		25
		0	1		50
		0	0		75



Note: If window end setting ≥ window start setting, the window end setting is set to 0%.

Figure 24.2 RPSS[1:0] and RPES[1:0] bit settings and the refresh-permitted period

24.2.3 WDTSR : WDT Status Register

Base address: WDT0 = 0x8004_2000

Offset address: 0x04

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	REFEF	UNDF	CNTVAL[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	CNTVAL[13:0]	Down-Counter Value Value counted by the down-counter	R
14	UNDFF	Underflow Flag 0: No underflow occurred 1: Underflow occurred	R/W*1
15	REFEF	Refresh Error Flag 0: No refresh error occurred 1: Refresh error occurred	R/W*1

Note 1. Only 0 can be written to clear the flag.

The WDTSR register displays the counter value of the down-counter and the states of underflow and refresh error occurrences.

CNTVAL[13:0] bits (Down-Counter Value)

Read these bits to confirm the value of the down-counter, but note that the read value may differ from the actual count by a value of one count.

UNDFF flag (Underflow Flag)

Read this flag to confirm whether an underflow has occurred in the down-counter.

The value 1 indicates that the down-counter has underflowed. The value 0 indicates that the down-counter has not underflowed.

Write 0 to the UNDFF flag to set the value to 0. Writing 1 has no effect.

Clearing of the UNDFF flag takes (N + 1) PCLKL cycles. In addition, clearing of the flag is ignored for (N + 1) PCLKL cycles after an underflow. N is specified in the WDTCR.CKS[3:0] bits as follows:

- When WDTCR.CKS[3:0] = 0x1, N = 4
- When WDTCR.CKS[3:0] = 0x4, N = 64
- When WDTCR.CKS[3:0] = 0xF, N = 128
- When WDTCR.CKS[3:0] = 0x7, N = 2048
- When WDTCR.CKS[3:0] = 0x8, N = 8192

REFEF flag (Refresh Error Flag)

Read this flag to confirm whether a refresh error (performing a refresh operation during a refresh-prohibited period) has occurred.

The value 1 indicates that a refresh error has occurred. The value 0 indicates that no refresh error has occurred.

Write 0 to the REFEF flag to set the value to 0. Writing 1 has no effect.

Clearing of the REFEF flag takes (N + 1) PCLKL cycles. In addition, clearing of the flag is ignored for (N + 1) PCLKL cycles after a refresh error. N is specified in the WDTCR.CKS[3:0] bits as follows:

- When WDTCR.CKS[3:0] = 0x1, N = 4
- When WDTCR.CKS[3:0] = 0x4, N = 64
- When WDTCR.CKS[3:0] = 0xF, N = 128
- When WDTCR.CKS[3:0] = 0x6, N = 512
- When WDTCR.CKS[3:0] = 0x7, N = 2048
- When WDTCR.CKS[3:0] = 0x8, N = 8192

24.2.4 WDTRCR : WDT Reset Control Register

Base address: WDT0 = 0x8004_2000

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RSTIRQS	—						
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	RSTIRQS	Reset Interrupt Request Selection 0: Error notification to ICU is permitted 1: Error notification to ICU is not performed	R/W

The WDTRCR register controls whether to send an error notification when underflow occurs in the down-counter of WDT.

There are some restrictions for writing to the WDTRCR register. For details, see section 24.3.2. Control over Writing to the WDTCR and WDTRCR Registers.

RSTIRQS bit (Reset Interrupt Request Selection)

The RSTIRQS bit specifies whether an error notification should be sent to the ICU module when an underflow or refresh error occurs in the down-counter. Set this bit to 0 to enable error notification from the WDT module.

24.3 Operation

24.3.1 Count Operation in Start Mode

Counting starts by a refresh operation of the WDT Refresh Register (WDTRR) when the WDT Control Register (WDTCR) and WDT Reset Control Register (WDTRCR) are set.

24.3.1.1 Register Setting

After the reset state is released, set the clock division ratio, window start and end positions, and timeout period in the WDTCR register. The value specified by the WDTCR.TOPS[1:0] bits is then set in the down-counter by a refresh operation to start counting down.

Thereafter, the value in the counter is reset at each refresh operation and count-down continues if the program runs normally and the counter is refreshed in the refresh-permitted period.

Figure 24.3 shows an example of operation under the following conditions:

- Reset Interrupt Request bit (RSTIRQS): 0b (error notification to ICU is permitted)
- Window Start Position Selection bits (RPSS[1:0]): 10b (75%)
- Window End Position Selection bits (RPES[1:0]): 10b (25%)

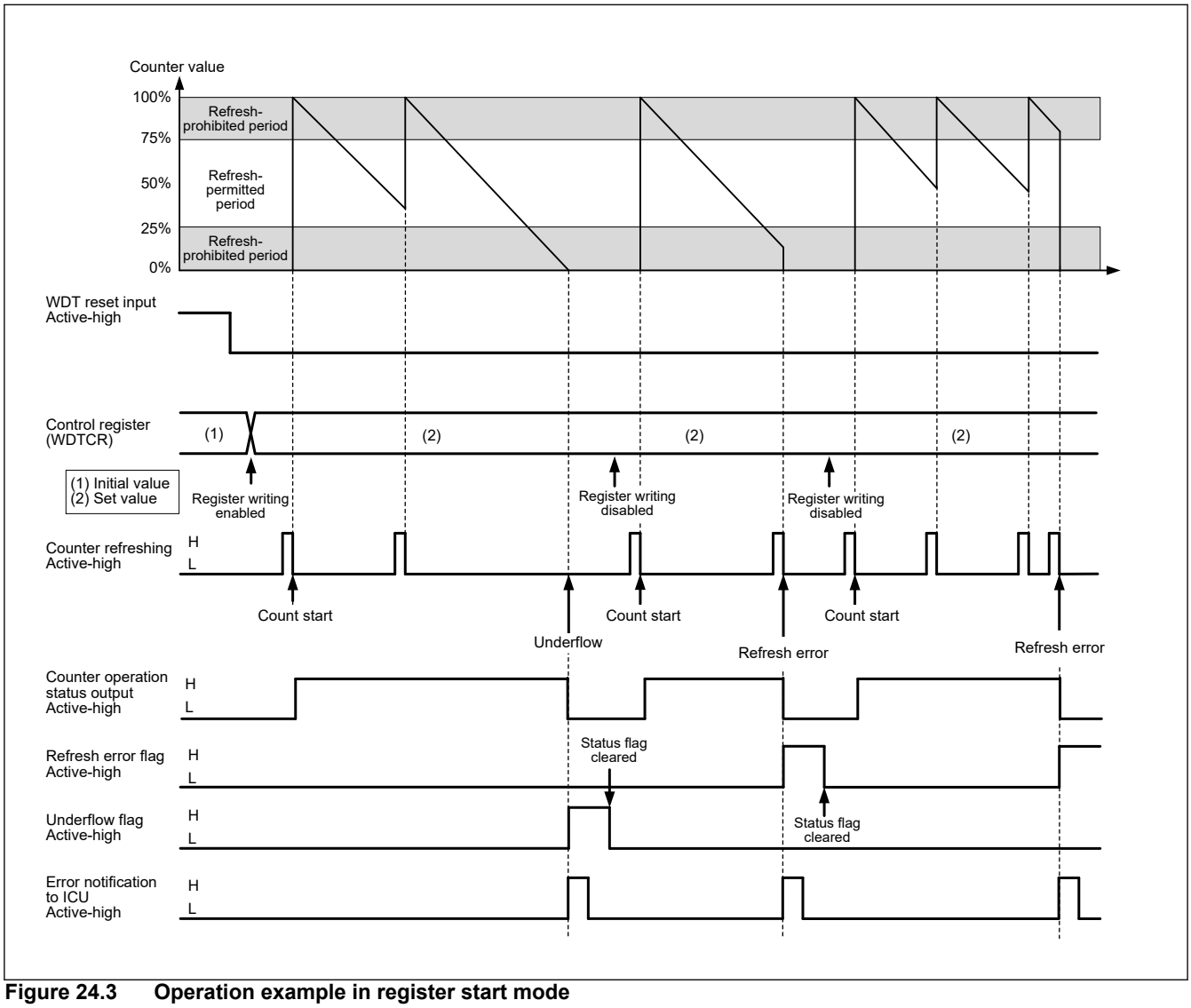


Figure 24.3 Operation example in register start mode

24.3.2 Control over Writing to the WDTCR and WDTRCR Registers

Writing to the WDT Control Register (WDTCR) is only possible once between the release from the reset state and the first refresh operation.

After a refresh operation (counting starts) or by writing to the WDTCR register, the protection signal in the WDT becomes 1 to protect the WDTCR register against subsequent attempts at writing.

Writing to the WDT Reset Control Register (WDTRCR) is also controlled similarly.

This protection is released by the reset source of the WDT. Any other reset sources cannot release the protection. Figure 24.4 shows control waveforms produced in response to writing to the WDTCR register.

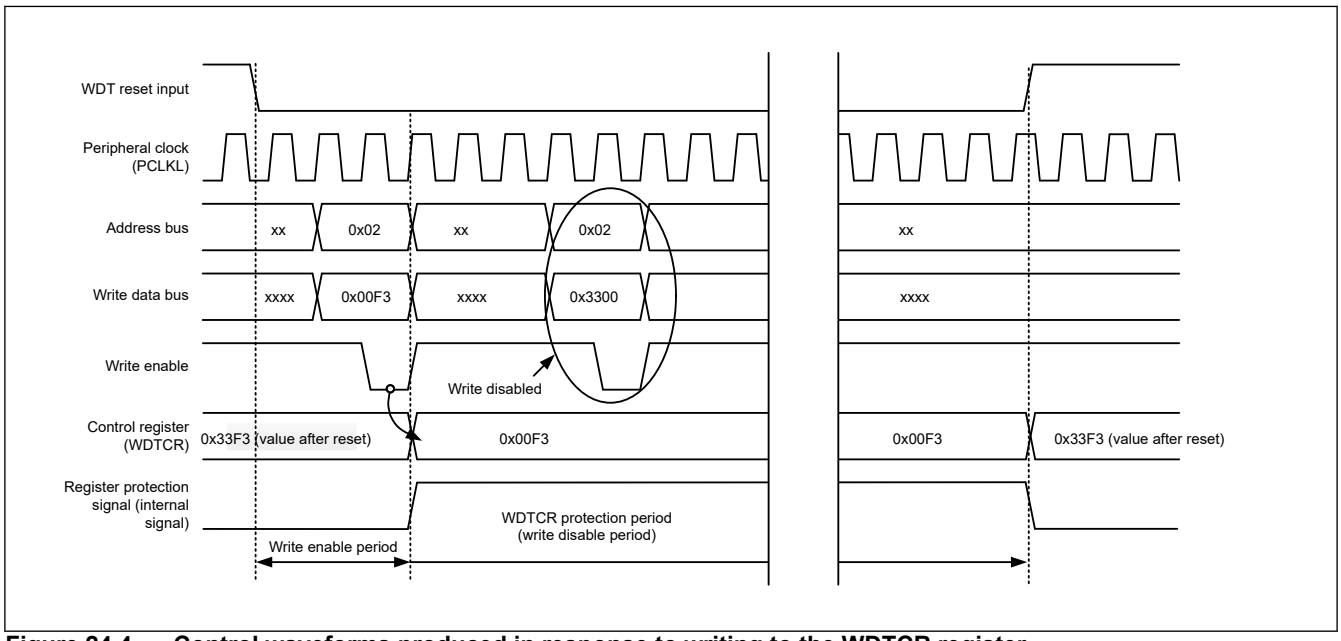


Figure 24.4 Control waveforms produced in response to writing to the WDTCR register

24.3.3 Refresh Operation

To refresh the down-counter and to start the down-counter operation (count start due to refreshing), write the values 0x00 and then 0xFF to the WDT Refresh Register (WDTRR). If a value other than 0xFF is written after 0x00, the down-counter is not refreshed. To perform refreshing after an invalid write, write 0x00 and then 0xFF again to the WDTRR register.

If 0x00 is written twice in succession, writing 0xFF after the second 0x00 refreshes the down-counter because the 0x00 → 0xFF condition is satisfied. The write sequence of 0x00 (n-1th) → 0x00 (nth) → 0xFF also satisfies the refreshing condition and thus refreshes the down-counter.

Even if a register other than WDTRR is accessed or WDTRR is read between writing 0x00 and writing 0xFF to WDTRR, correct refreshing will be done.

[Sample sequences of writing that are valid for refreshing the counter]

- 0x00 → 0xFF
- 0x00 (n - 1th time) → 0x00 (nth time) → 0xFF
- 0x00 → access to another register or read from the WDTRR register → 0xFF

[Sample sequences of writing that are not valid for refreshing the counter]

- 0x23 (a value other than 0x00) → 0xFF
- 0x00 → 0x54 (a value other than 0xFF)
- 0x00→ 0xAA (0x00 and a value other than 0xFF) → 0xFF

If 0xFF is written to the WDTRR register during the refresh-permitted period after 0x00 is written outside of the period, write operation is acknowledged and down-counter is refreshed. Whether writing is made within the refresh-permitted period is determined by when 0xFF is written.

After 0xFF is written to the WDT Refresh Register (WDTRR), refreshing the down-counter requires up to 4 counting cycles. The number of peripheral clock (PCLKL) cycles in a single counting cycle differs depending on the setting of the Clock Division Ratio Selection bits (WDTCR.CKS[3:0]).

Therefore, writing 0xFF to the WDTRR register should be completed within 4 count cycles before the refresh-permitted period end position or before the down-counter underflows. Confirm the down-counter value with the down-counter value bits (WDTSR.CNTVAL[13:0]).

[Sample of refresh operation timing]

- When the window start position is 0x1FFF, if 0x00 is written to the WDTRR register before 0x1FFF (for example, 0x2002), the down-counter can be refreshed by writing 0xFF to the WDTRR register after the value of WDTSR.CNTVAL[13:0] becomes 0x1FFF.
- When the window end position is 0x1FFF, if the value of WDTSR.CNTVAL[13:0] is 0x2003 (4 counts before 0x1FFF) or more immediately after 0x00 and 0xFF are written to the WDTRR register, the down-counter will be refreshed.
- If the refresh-permitted period goes down to 0x0000, refreshing is possible until just before underflow. In this case, if the value of WDTSR.CNTVAL[13:0] is 0x0003 (4 count cycles before underflow) or more immediately after 0x00 and 0xFF are written to the WDTRR register, underflow does not occur and the down-counter is refreshed.

Figure 24.5 shows WDT refresh-operation waveforms when the clock division ratio is PCLKL/64.

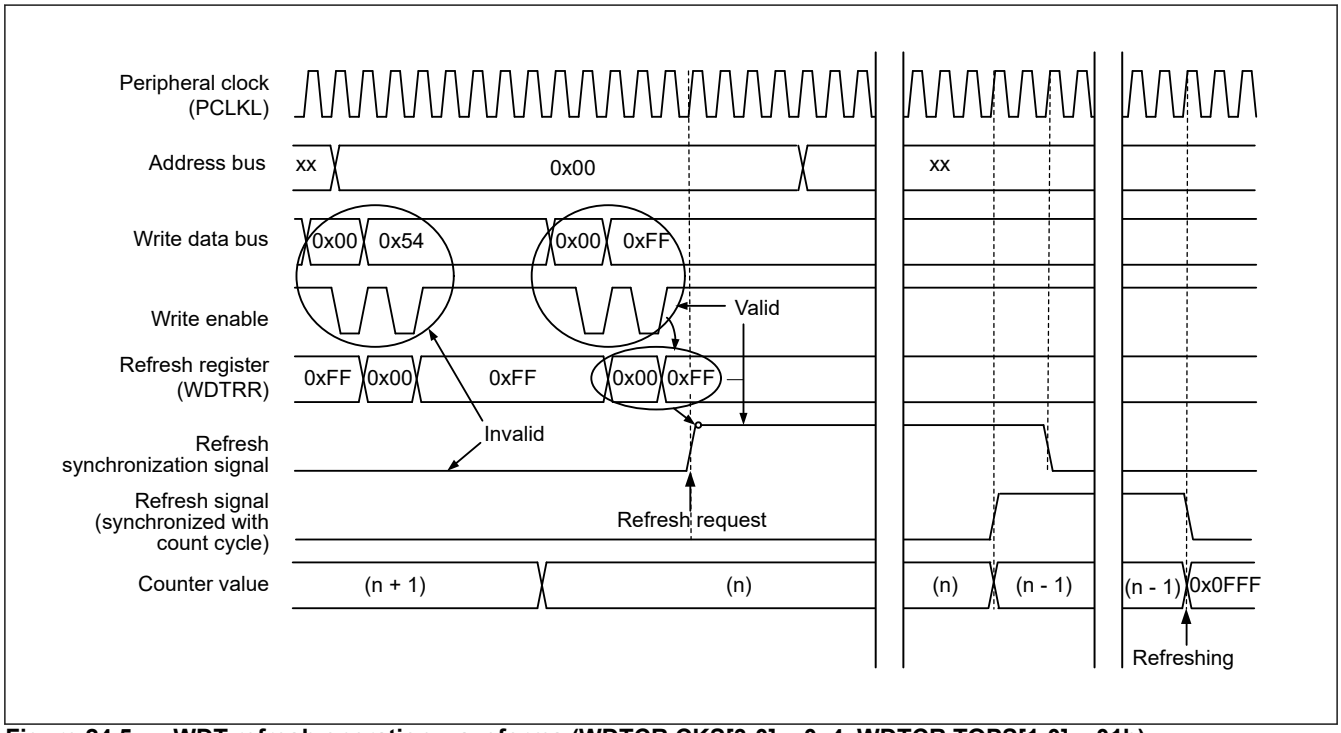


Figure 24.5 WDT refresh operation waveforms (WDTCR.CKS[3:0] = 0x4, WDTCR.TOPS[1:0] = 01b)

24.3.4 Status Flag

The Refresh Error flag (WDTSR.REFEF) and the Underflow flag (WDTSR.UNDFF) retain the error causes after error notifications are output to the Interrupt Controller (ICU).

Occurrence state of error notifications to ICU can be confirmed by reading the WDTSR.REFEF flag or the WDTSR.UNDFF flag after reset is released or on occurrence of an error notification to ICU.

To clear these flags, write 0. Writing 1 is ignored.

If these flags are not cleared, there is no effect on the operation. On occurrence of the next error notification to ICU, the previous error notification is automatically cleared and a new notification to ICU is written.

25. Real Time Clock (RTC)

25.1 Overview

The RTC module can count a 100-year calendar from 2000 to 2099 and automatically adjust dates for leap years. The RTC uses an internal clock acquired by dividing the main clock oscillator or PLL output by a prescaler. Year, month, date, day-of-week, a.m./p.m. (in 12-hour mode), hour, minute, second is counted. No external oscillator is required but battery backup is not supported.

Table 25.1 lists the RTC specifications and Figure 25.1 shows a block diagram. Table 25.2 lists the RTC I/O pin and Table 25.3 lists the interrupt sources.

Table 25.1 RTC specifications

Parameter	Specifications
Number of units	1 unit
Count mode	Calendar count mode
Count source	Main clock oscillator divided by 128 (195.3125 kHz)
Clock and calendar functions	• Year, month, day of the week, day, hour, minute, second counters and sub-counters, counting up to 99 years are supported • Year, month, day of the week, day, hour, minute, and second counters: BCD code display • 12 hours/24 hours mode switching function • Automatic adjustment function for leap years • Start/stop function • 1 Hz pulse output
Interrupts	• Alarm interrupt (RTC_ALM) As an alarm interrupt condition, day-of-week, hour and minute can be selected • 1 second interrupt (RTC_1S) Interrupt can be generated every 1 second • Fixed interval interrupt (RTC_PRD) 0.25 second, 0.5 second, 1 second, 1 minute, 1 hour, 1 day, or 1 month can be selected as an interrupt period
Module-stop function	Module-stop state can be set to reduce power consumption.

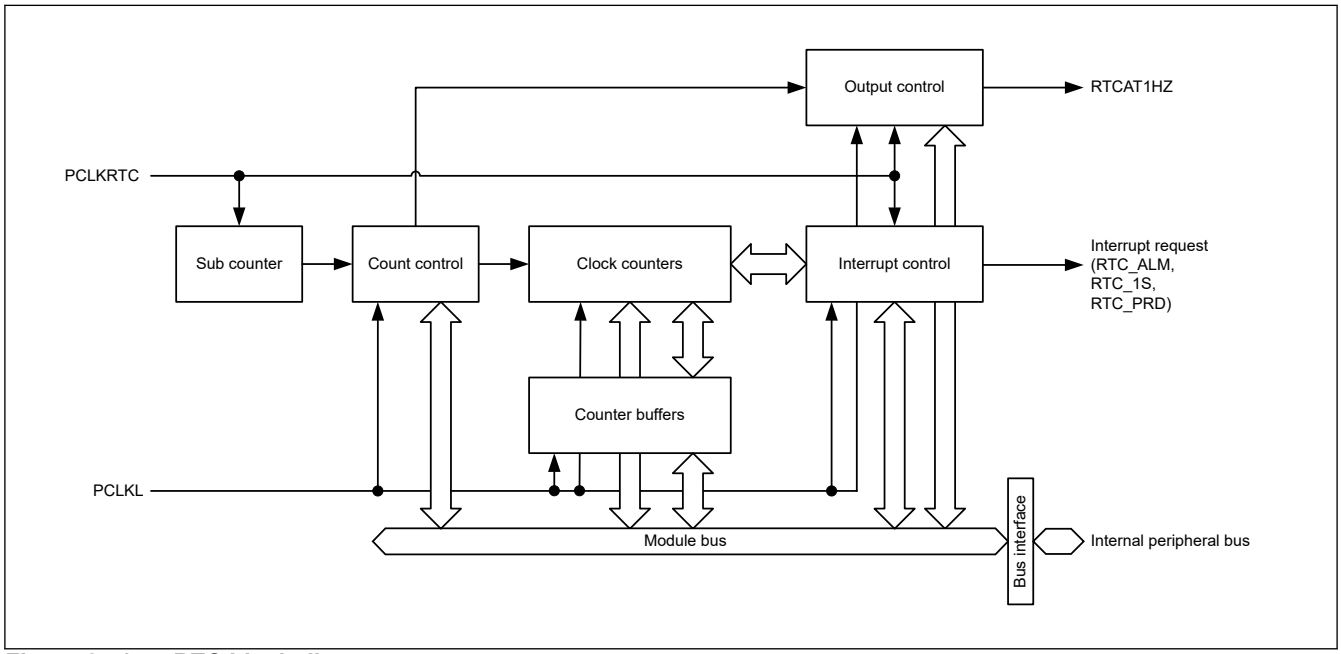


Figure 25.1 RTC block diagram

Table 25.2 RTC I/O pins

Pin name	I/O	Function
RTCAT1HZ	Output	RTC 1 Hz output

Table 25.3 RTC interrupt sources

Name	Interrupt sources	CPU0 GIC request	DMAC activation
RTC_ALM	Alarm interrupt	Possible	Not possible
RTC_1S	1 second interrupt	Possible	Not possible
RTC_PRD	Fixed interval interrupt	Possible	Not possible

25.2 Register Descriptions

25.2.1 RTCA0CTL0 : RTC Control Register 0

Base address: RTC = 0x8100_9000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	RTCA0CE	RTCA0CEST	RTCA0AMPM	RTCA0SLSB	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	RTCA0SLSB	RTCA0SCMP enable/disable setting Set this bit to 1, see the RTCA0SCMP register description. When setting this bit, be sure to follow the flow described in section 25.3.1.1. Initial Setting. It is prohibited to change the setting of this bit while the clock counter operation is enabled (RTCA0CE = 1). If the setting of this bit is changed during clock counter operation, the operation of RTC Controller cannot be guaranteed. When changing the setting of this bit, first perform initialization according to section 25.3.1.5. Initialization of RTC While Clock Counter Operation is Enabled, and then perform the setting of this bit again according to the flow described in section 25.3.1.1. Initial Setting.	R/W
5	RTCA0AMPM	RTCA0HOUR, RTCA0ALH display format selection bit 0: 12-hour display 1: 24-hour display	R/W
6	RTCA0CEST	RTC Controller Enable Status 0: Clock counter operation stopped status 1: Clock counter operation enabled status (the clock counter counts up)	R
7	RTCA0CE	RTC Controller Enable Bit Setting this bit to 0 initializes the internal circuit of RTC Controller other than the control registers*1. Moreover, interrupts (RTC_ALM, RTC_1S, RTC_PRD) are cleared. 0: Clock counter operation stopped 1: Clock counter operation enabled (the clock counter counts up)	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Status flags RTCA0WST, RTCA0RSST, and RTCA0WSST, which cannot be written by the user, are cleared.

25.2.2 RTCA0CTL1 : RTC Control Register 1

Base address: RTC = 0x8100_9000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	RTCA01HZE	RTCA0ALME	RTCA01SE	RTCA0CT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	RTCA0CT[2:0]	Fixed interval interrupt (RTC_PRD) output setting bit See section 25.3.1.7. Changing the Setting of Fixed Interval Interrupt During Clock Counter Operation for changing the setting of this field. RTCA0CT[2]/RTCA0CT[1]/RTCA0CT[0] 0 0 0: Fixed interval interrupt output stop 0 0 1: Interrupt once every 0.25 seconds (in synchronization with second count-up) 0 1 0: Interrupt once every 0.5 seconds (in synchronization with second count-up) 0 1 1: Interrupt once every 1 second (in synchronization with second count-up) 1 0 0: Interrupt once every 1 minute (every 1 minute 00 seconds) 1 0 1: Interrupt once every 1 hour (every 1 hour 00 minutes 00 seconds) 1 1 0: Interrupt once every 1 day (every 1 day 00 hours 00 minutes 00 seconds) 1 1 1: Interrupt once every 1 month (every 1 month first day 00 hours 00 minutes 00 seconds a.m.)	R/W
3	RTCA01SE	1 second interrupt (RTC_1S) output enable bit 0: Stop interrupt output at every 1 second 1: Enable interrupt output at every 1 second	R/W
4	RTCA0ALME	Alarm interrupt (RTC_ALM) output enable bit 0: Disable 1: Enable	R/W
5	RTCA01HZE	This bit enables/disables 1 Hz pulse output (RTCAT1HZ). 0: Disable 1 Hz pulse output (RTCAT1HZ fixed to 0) 1: Enable 1 Hz pulse output	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

25.2.3 RTCA0CTL2 : RTC Control Register 2

Base address: RTC = 0x8100_9000

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	RTCA0WSST	RTCA0RSST	RTCA0RSUB	RTCA0WST	RTCA0WAIT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RTCA0WAIT	RTC Controller Counter Wait Control This bit controls the count-up operation of the counters. Be sure to write 1 to this bit when reading and writing counter values while the clock counter operation is enabled (RTCA0CE = 1). When this bit is 1, on occurrence of RTCA0SUBC overflow, the overflow information is held internally, and after 0 is written to this bit, RTCA0SEC is counted up. However, when the value of the second counter is written while RTCA0WAIT = 1, the overflow information that was held is discarded. 0: Counter operation 1: Count-up operation of RTCA0SEC, RTCA0MIN, RTCA0HOUR, RTCA0WEEK, RTCA0DAY, RTCA0MONTH, RTCA0YEAR is stopped (count-up operation of second, minute, hour, week, day, month, and year count registers is stopped)	R/W
1	RTCA0WST	RTC Controller Counter Wait Status When 1 is written to RTCA0WAIT, the value of this bit becomes 1 when the count-up operation of RTCA0SEC, RTCA0MIN, RTCA0HOUR, RTCA0WEEK, RTCA0DAY, RTCA0MONTH, RTCA0YEAR has stopped completely. Before reading and writing clock counters (RTCA0SEC, RTCA0MIN, RTCA0HOUR, RTCA0WEEK, RTCA0DAY, RTCA0MONTH, RTCA0YEAR) during clock counter operation (RTCA0CE = 1), check that the value of this register is 1. Also, when write to a clock counter (RTCA0SEC, RTCA0MIN, RTCA0HOUR, RTCA0WEEK, RTCA0DAY, RTCA0MONTH, RTCA0YEAR) occurred, the value written to RTCA0WAIT is reflected only when the write operation has completed (status hold). 0: Count-up status of RTCA0SECC, RTCA0MINC, RTCA0HOURC, RTCA0WEEKC, RTCA0DAYC, RTCA0MONC, RTCA0YEARC 1: Count-up stopped status of RTCA0SECC, RTCA0MINC, RTCA0HOURC, RTCA0WEEKC, RTCA0DAYC, RTCA0MONC, RTCA0YEARC (count-up operation of second, minute, hour, week, day, month, and year count registers is stopped)	R
2	RTCA0RSUB	RTCA0SUBC Data Transfer Control This is a control bit for transferring the value of RTCA0SUBC of RTCA0SRBU. Use it when reading the value of RTCA0SRBU while the clock counter operation is enabled (RTCA0CE = 1). When 1 is written to this bit, the value of RTCA0SUBC is loaded to RTCA0SRBU in synchronization with PCLKRTC. For the usage method, see section 25.3.1.4. Reading RTCA0SRBU While Clock Counter Operation is Enabled. 0: RTCA0SRBU data hold 1: Transfer of value of RTCA0SUBC to RTCA0SRBU	R/W
3	RTCA0RSST	RTCA0SRBU Transfer Status When 1 is written to RTCA0RSUB, the value of this bit becomes 1 when the value of RTCA0SUBC is loaded to RTCA0SRBU. Before reading RTCA0SRBU during clock counter operation (RTCA0CE = 1), check that the value of this register is 1. 0: RTCA0SRBU data hold status 1: Transfer of the value of RTCA0SUBC to RTCA0SRBU completed	R
4	RTCA0WSST	RTCA0SCMP Write Status This bit is set to 1 when RTCA0SCMP is written to while the clock counter operation is enabled (RTCA0CE = 1). It is then cleared to 0 on completion of RTCA0SCMP write (RTCA0SUBC overflow). Do not write to RTCA0SCMP while the value of this bit is 1 because write to RTCA0SCMP is in progress. 0: RTCA0SCMP write completed 1: RTCA0SCMP write in progress	R
31:5	—	These bits are read as 0. The write value should be 0.	R/W

25.2.4 RTCA0SUBC : RTC Sub Count Register

Base address: RTC = 0x8100_9000

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	RTCA0SUBC[21:0]																					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
21:0	RTCA0SUBC[21:0]	Register that counts the 1 second reference time This register is cleared on reset or on write to the RTCA0SEC (RTC Sec Count Buffer register) or RTCA0TIME (RTC Time Set register), and arbitrary values cannot be written to this register. Count-up occurs on clock input when the clock counter operation is enabled (RTCA0CE = 1). When this register is read while the clock counter operation is enabled (RTCA0CE = 1), the register value, which changes in synchronization with PCLKRTC, is read asynchronously, so that the read value is not guaranteed. To check the value of this register while the clock counter operation is enabled (RTCA0CE = 1), use the RTCA0SRBU register. The count-up operation starts from 0x0, and on matching the value set to RTCA0SCMP, an overflow signal is output to RTCA0SEC (Second Count Buffer register), and this register is cleared. [Example] Operation using 195.3 kHz clock 195312 - 1 = 195311 (decimal code) = 0x2FAEF is set to RTCA0SCMP. Counting up is performed each time a 195.3 kHz clock is input, taking on values from 0x0 to 0x2FAEF. 1 second is counted when the value changes from 0x2FAEF to 0x0.	R
31:22	—	These bits are read as 0.	R

25.2.5 RTCA0SRBU : RTC Sub Count Register Read Buffer

Base address: RTC = 0x8100_9000

Offset address: 0x10

Bit position:	31										21																				0
Bit field:	—	—	—	—	—	—	—	—	—	—	RTCA0SRBU[21:0]																					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
21:0	RTCA0SRBU[21:0]	Read buffer register of RTCA0SUBC This register is cleared when reset, at which time arbitrary values cannot be written. When 1 is written to the RTCA0RSUB bit while the clock counter operation is enabled (RTCA0CE = 1), the value of RTCA0SUBC is loaded to this register in synchronization with PCLKRTC clock. Be sure to perform RTCA0SRBU read according to the flow described in section 25.3.1.4. Reading RTCA0SRBU While Clock Counter Operation is Enabled.	R
31:22	—	These bits are read as 0.	R

25.2.6 RTCA0SEC : RTC Sec Count Buffer Register

Base address: RTC = 0x8100_9000

Offset address: 0x14

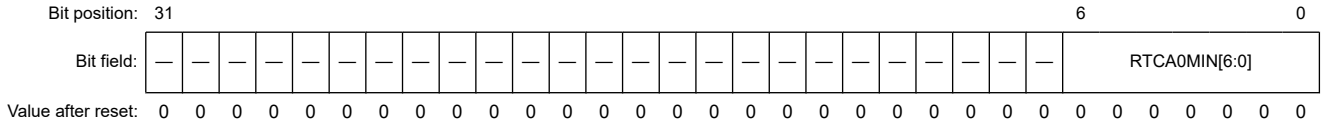
Bit position:	31																								6						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCA0SEC[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	RTCA0SEC[6:0]	Buffer register to read/write RTC Second Count register (RTCA0SECC). This register is used when Second Count register is read and written. When setting this register, a decimal value of 00 to 59 must be set in BCD code.	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

25.2.7 RTCA0MIN : RTC Min Count Buffer Register

Base address: RTC = 0x8100_9000

Offset address: 0x18

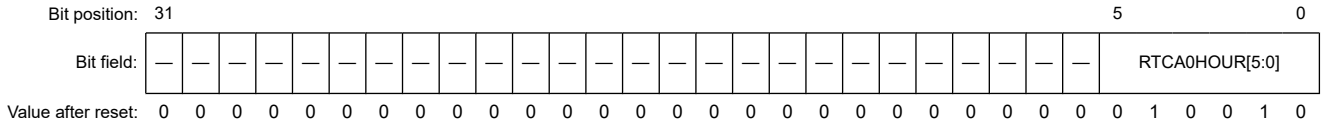


Bit	Symbol	Function	R/W
6:0	RTCA0MIN[6:0]	Buffer register to read/write RTC Minute Count register (RTCA0MINC). This register is used when Minute Count register is read and written. When setting this register, a decimal value of 00 to 59 must be set in BCD code.	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

25.2.8 RTCA0HOUR : RTC Hour Count Buffer Register

Base address: RTC = 0x8100_9000

Offset address: 0x1C



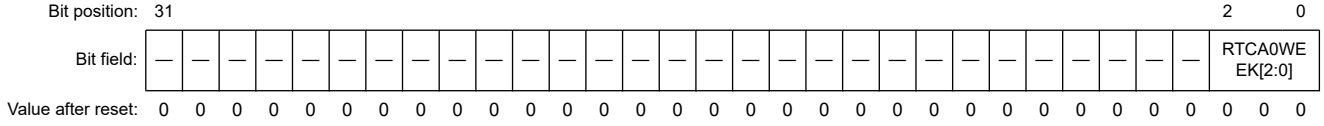
Bit	Symbol	Function	R/W
5:0	RTCA0HOUR[5:0]	Buffer register to read/write RTC Hour Count register (RTCA0HOURC). This register is used when Hour Count register is read and written. When setting this register, a decimal value of 01 to 12 or 21 to 32 must be set when RTCA0AMPM = 0, or a decimal value of 0 to 23 when RTCA0AMPM = 1, in BCD code in both cases. Display of RTCA0HOUR is 12-hour display when the RTCA0AMPM bit is set to 0, and 24-hour display when it is 1. In the case of 12-hour display, am/pm is indicated by the 5th bit of RTCA0HOUR. During am, RTCA0HOUR[5] = 0, and during pm, RTCA0HOUR[5] = 1.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

12-hour display		24-hour display	
Time	RTCA0HOUR display	Time	RTCA0HOUR display
0 am	0x12	0	0x00
1 am	0x01	1	0x01
⋮	⋮	⋮	⋮
9 am	0x09	9	0x09
10 am	0x10	10	0x10
11 am	0x11	11	0x11
0 pm	0x32	12	0x12
1 pm	0x21	13	0x13
⋮	⋮	⋮	⋮
9 pm	0x29	21	0x21
10 pm	0x30	22	0x22
11 pm	0x31	23	0x23

25.2.9 RTCA0WEEK : RTC Week Count Buffer Register

Base address: RTC = 0x8100_9000

Offset address: 0x20

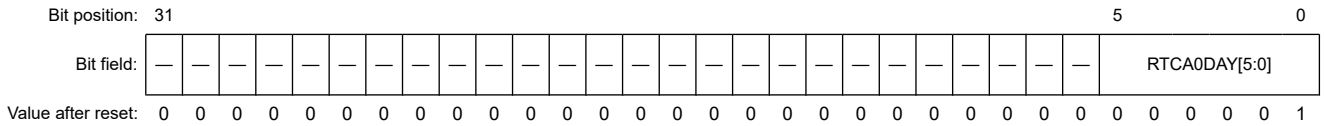


Bit	Symbol	Function	R/W
2:0	RTCA0WEEK[2:0]	Buffer register to read/write RTC Week Count register (RTCA0WEEKC). This register is used when Week Count register is read and written. When setting this register, a decimal value of 00 to 06 must be set in BCD code. There is no particular correspondence between the value of RTCA0WEEK and the day of the week. Set the correspondence according to the application to be used. [Example] 0 → Sunday, 1 → Monday, …, 6 → Saturday	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

25.2.10 RTCA0DAY : RTC Day Count Buffer Register

Base address: RTC = 0x8100_9000

Offset address: 0x24

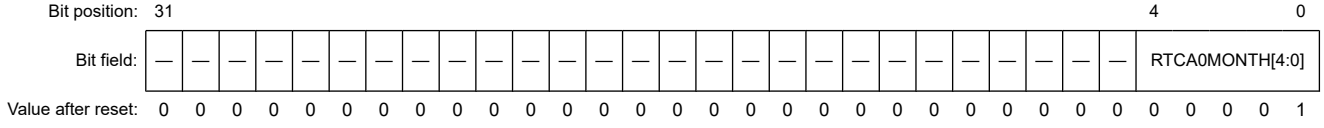


Bit	Symbol	Function	R/W
5:0	RTCA0DAY[5:0]	Buffer register to read/write RTC Day Count register (RTCA0DAYC). This register is used when Day Count register is read and written. When setting this register, a decimal value of 01 to 31 must be set in BCD code. ● 01 to 31 (January, March, May, July, August, October, December) ● 01 to 30 (April, June, September, November) ● 01 to 29 (February, leap year) ● 01 to 28 (February, non-leap year)	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

25.2.11 RTCA0MONTH : RTC Month Count Buffer Register

Base address: RTC = 0x8100_9000

Offset address: 0x28

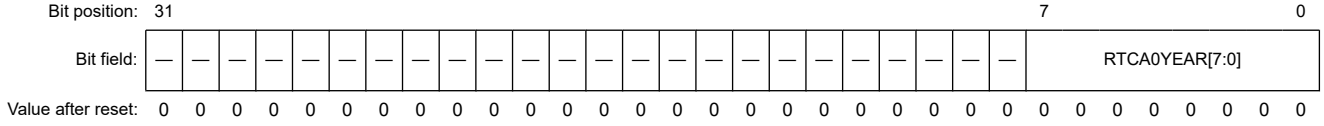


Bit	Symbol	Function	R/W
4:0	RTCA0MONTH[4:0]	Buffer register to read/write RTC Month Count register (RTCA0MONC). This register is used when Month Count register is read and written. When setting this register, a decimal value of 01 to 12 must be set in BCD code.	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

25.2.12 RTCA0YEAR : RTC Year Count Buffer Register

Base address: RTC = 0x8100_9000

Offset address: 0x2C

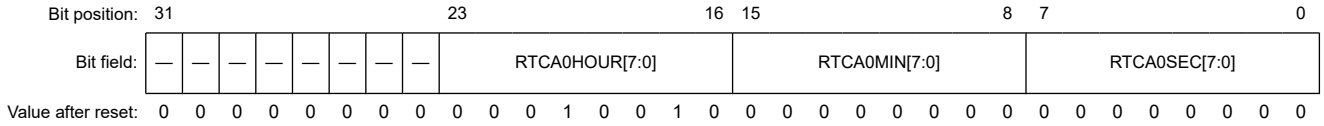


Bit	Symbol	Function	R/W
7:0	RTCA0YEAR[7:0]	Buffer register to read/write RTC Year Count register (RTCA0YEARC). This register is used when Year Count register is read and written. When setting this register, a decimal value of 00 to 99 must be set in BCD code.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

25.2.13 RTCA0TIME : RTC Time Set Register

Base address: RTC = 0x8100_9000

Offset address: 0x30



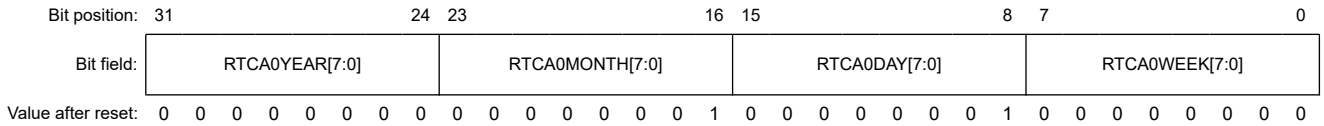
Bit	Symbol	Function	R/W
7:0	RTCA0SEC[7:0]	See RTCA0SEC register	R/W
15:8	RTCA0MIN[7:0]	See RTCA0MIN register	R/W
23:16	RTCA0HOUR[7:0]	See RTCA0HOUR register	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

The RTCA0TIME register is a register for accessing the RTCA0HOUR, RTCA0MIN, and RTCA0SEC registers simultaneously. By using this register, the RTCA0HOUR, RTCA0MIN, and RTCA0SEC registers can be read or written simultaneously.

25.2.14 RTCA0CAL : RTC Calendar Set Register

Base address: RTC = 0x8100_9000

Offset address: 0x34



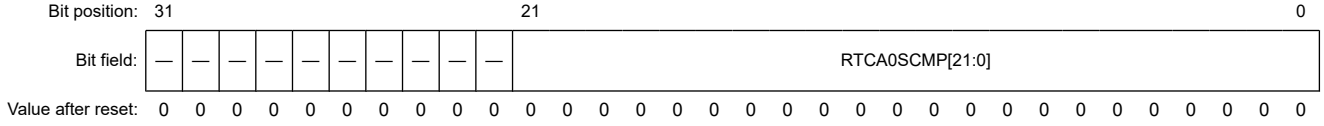
Bit	Symbol	Function	R/W
7:0	RTCA0WEEK[7:0]	See RTCA0WEEK register	R/W
15:8	RTCA0DAY[7:0]	See RTCA0DAY register	R/W
23:16	RTCA0MONTH[7:0]	See RTCA0MONTH register	R/W
31:24	RTCA0YEAR[7:0]	See RTCA0YEAR register	R/W

The RTCA0CAL register is a register for accessing the RTCA0YEAR, RTCA0MONTH, RTCA0DAY, and RTCA0WEEK registers simultaneously. By using this register, the RTCA0YEAR, RTCA0MONTH, RTCA0DAY, and RTCA0WEEK registers can be read or written simultaneously.

25.2.15 RTCA0SCMP : RTC Sub Count Compare Register

Base address: RTC = 0x8100_9000

Offset address: 0x3C

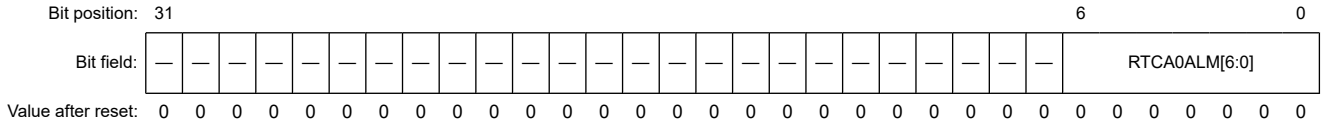


Bit	Symbol	Function	R/W
21:0	RTCA0SCMP[21:0]	Register that sets the compare value of RTCA0SUBC (sub-counter). The setting of this register is valid when RTCA0SLSB = 1. When RTCA0SLSB = 1 is set, RTCA0SUBC starts counting up from 0, and when the value set to this register is matched, an overflow signal is output to RTCA0SEC (Second Count Buffer register), and the register is cleared. Set the value for this register according to the frequency of the selected input clock. Accurate one second counting can be performed to PCLKRTC through the value set to this register. [Example] Setting of 195,312 (decimal) - 1 = 195,311 (decimal) = 0x2FAEF to RTCA0SCMP When performing write to this register during clock counter operation (RTCA0CE = 1), be sure to check that RTCA0WSST = 0. When writing to this register during clock counter operation (RTCA0CE = 1), write ends at the timing of RTCA0SUBC overflow. Be sure to perform RTCA0SCMP write according to the flows described in section 25.3.1.1. Initial Setting and section 25.3.1.6. Writing to RTCA0SCMP During Clock Counter Operation.	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

25.2.16 RTCA0ALM : RTC Alarm Min Set Register

Base address: RTC = 0x8100_9000

Offset address: 0x40



Bit	Symbol	Function	R/W
6:0	RTCA0ALM[6:0]	RTCA0ALM is a register that performs the minute setting for the alarm interrupt. Set a decimal value of 00 to 59 in BCD code (alarm detection is not performed if an out-of-range value is set.)	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

25.2.17 RTCA0ALH : RTC Alarm Hour Set Register

Base address: RTC = 0x8100_9000

Offset address: 0x44

Bit position:	31																							5	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCA0ALH[5:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

Bit	Symbol	Function	R/W
5:0	RTCA0ALH[5:0]	RTCA0ALH is a register that performs the hour setting for the alarm interrupt. When RTCA0AMPM = 0, set a decimal value of 01 to 12 or 21 to 32, and when RTCA0AMPM = 1, set a decimal value of 00 to 23, in BCD code in both cases (alarm detection is not performed if an out-of-range value is set.)	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

25.2.18 RTCA0ALW : RTC Alarm Week Set Register

Base address: RTC = 0x8100_9000

Offset address: 0x48

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	RTCA0ALW6	RTCA0ALW5	RTCA0ALW4	RTCA0ALW3	RTCA0ALW2	RTCA0ALW1	RTCA0ALW0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RTCA0ALW0	Alarm interrupt day of the week setting bit 0 0: Disable alarm 1: Enable alarm	R/W
1	RTCA0ALW1	Alarm interrupt day of the week setting bit 1 0: Disable alarm 1: Enable alarm	R/W
2	RTCA0ALW2	Alarm interrupt day of the week setting bit 2 0: Disable alarm 1: Enable alarm	R/W
3	RTCA0ALW3	Alarm interrupt day of the week setting bit 3 0: Disable alarm 1: Enable alarm	R/W
4	RTCA0ALW4	Alarm interrupt day of the week setting bit 4 0: Disable alarm 1: Enable alarm	R/W
5	RTCA0ALW5	Alarm interrupt day of the week setting bit 5 0: Disable alarm 1: Enable alarm	R/W
6	RTCA0ALW6	Alarm interrupt day of the week setting bit 6 0: Disable alarm 1: Enable alarm	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

The RTCA0ALW register enables the alarm corresponding to RTCA0WEEK value. For example, if RTCA0WEEK = 0 is Sunday, RTCA0ALW0 bit functions as Sunday alarm. If RTCA0WEEK = 6 is Saturday, RTCA0ALW6 bit functions as Saturday alarm.

25.2.19 RTCA0SECC : RTC Second Count Register

Base address: RTC = 0x8100_9000

Offset address: 0x4C

Bit position:	31																								6						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCA0SECC[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	RTCA0SECC[6:0]	Counts up the seconds This register counts from 00 to 59, and when its value changes from 59 to 00 in BCD code, an overflow signal is output to RTCA0MINC.	R
31:7	—	These bits are read as 0.	R

25.2.20 RTCA0MINC : RTC Minute Count Register

Base address: RTC = 0x8100_9000

Offset address: 0x50

Bit position:	31																								6						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCA0MINC[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	RTCA0MINC[6:0]	Counts up the minutes This register counts from 00 to 59, and when its value changes from 59 to 00 in BCD code, an overflow signal is output to RTCA0HOURC.	R
31:7	—	These bits are read as 0.	R

25.2.21 RTCA0HOURC : RTC Hour Count Register

Base address: RTC = 0x8100_9000

Offset address: 0x54

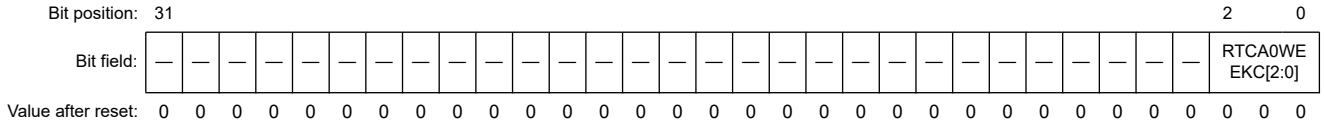
Bit position:	31																									5					0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCA0HOURC[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0

Bit	Symbol	Function	R/W
5:0	RTCA0HOURC[5:0]	Counts up the hours This register counts from 00 to 23, or 01 to 12, or 21 to 32 in BCD code depending on the setting of RTCA0AMPM, and when its value changes from 23 to 00 or from 32 to 01, an overflow signal is output to RTCA0DAY and RTCA0WEEK.	R
31:6	—	These bits are read as 0.	R

25.2.22 RTCA0WEEKC : RTC Week Count Register

Base address: RTC = 0x8100_9000

Offset address: 0x58

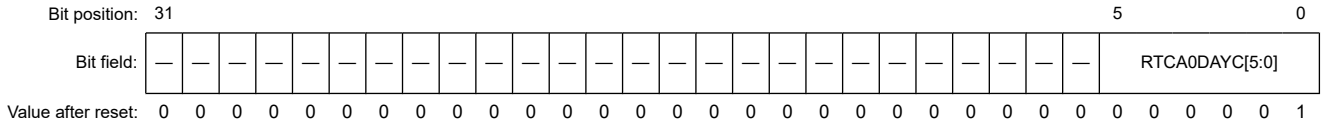


Bit	Symbol	Function	R/W
2:0	RTCA0WEEKC[2:0]	Counts up the weeks This register counts up on occurrence of RTCA0HOURC overflow when the clock counter operation is enabled (RTCA0CE = 1).	R
31:3	—	These bits are read as 0.	R

25.2.23 RTCA0DAYC : RTC Day Count Register

Base address: RTC = 0x8100_9000

Offset address: 0x5C

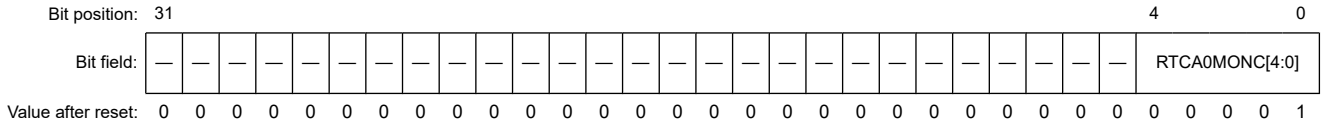


Bit	Symbol	Function	R/W
5:0	RTCA0DAYC[5:0]	Counts up the days This register counts up on occurrence of RTCA0HOURC overflow when the clock counter operation is enabled (RTCA0CE = 1). This register counts according to the value of RTCA0MONC, and on occurrence of an overflow, an overflow signal is output to RTCA0MONC.	R
31:6	—	These bits are read as 0.	R

25.2.24 RTCA0MONC : RTC Month Count Register

Base address: RTC = 0x8100_9000

Offset address: 0x60



Bit	Symbol	Function	R/W
4:0	RTCA0MONC[4:0]	Counts up the months This register counts up on occurrence of RTCA0DAYC overflow while the clock counter operation is enabled (RTCA0CE = 1). This register counts from 01 to 12 in BCD code, and when its value changes from 12 to 01, an overflow signal is output to RTCA0YEARC.	R
31:5	—	These bits are read as 0.	R

25.2.25 RTCA0YEARC : RTC Year Count Register

Base address: RTC = 0x8100_9000

Offset address: 0x64

Bit position:	31																						7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RTCA0YEARC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RTCA0YEARC[7:0]	Counts up the years This register counts up on occurrence of RTCA0MONC overflow while the clock counter operation is enabled (RTCA0CE = 1). This register counts from 00 to 99 in BCD code.	R
31:8	—	These bits are read as 0.	R

25.2.26 RTCA0TIMEC : RTC Time Count Register

Base address: RTC = 0x8100_9000

Offset address: 0x68

Bit position:	31								23							16	15							8	7						0
Bit field:	—	—	—	—	—	—	—	—	RTCA0HOURC[7:0]								RTCA0MINC[7:0]								RTCA0SECC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RTCA0SECC[7:0]	See RTCA0SECC register	R
15:8	RTCA0MINC[7:0]	See RTCA0MINC register	R
23:16	RTCA0HOURC[7:0]	See RTCA0HOURC register	R
31:24	—	These bits are read as 0.	R

This register is for reading the RTCA0HOURC, RTCA0MINC, and RTCA0SECC simultaneously.

25.2.27 RTCA0CALC : RTC Calendar Count Register

Base address: RTC = 0x8100_9000

Offset address: 0x6C

Bit position:	31							24	23							16	15							8	7						0
Bit field:	RTCA0YEARC[7:0]								RTCA0MONC[7:0]								RTCA0DAYC[7:0]								RTCA0WEEKC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RTCA0WEEKC[7:0]	See RTCA0WEEKC register	R
15:8	RTCA0DAYC[7:0]	See RTCA0DAYC register	R
23:16	RTCA0MONC[7:0]	See RTCA0MONC register	R
31:24	RTCA0YEARC[7:0]	See RTCA0YEARC register	R

The RTCA0CALC register is for reading the RTCA0YEARC, RTCA0MONC, RTCA0DAYC, and RTCA0WEEKC simultaneously.

25.3 Operation

25.3.1 Programming RTC

25.3.1.1 Initial Setting

Perform initial setting of RTC according to the flow in Figure 25.2.

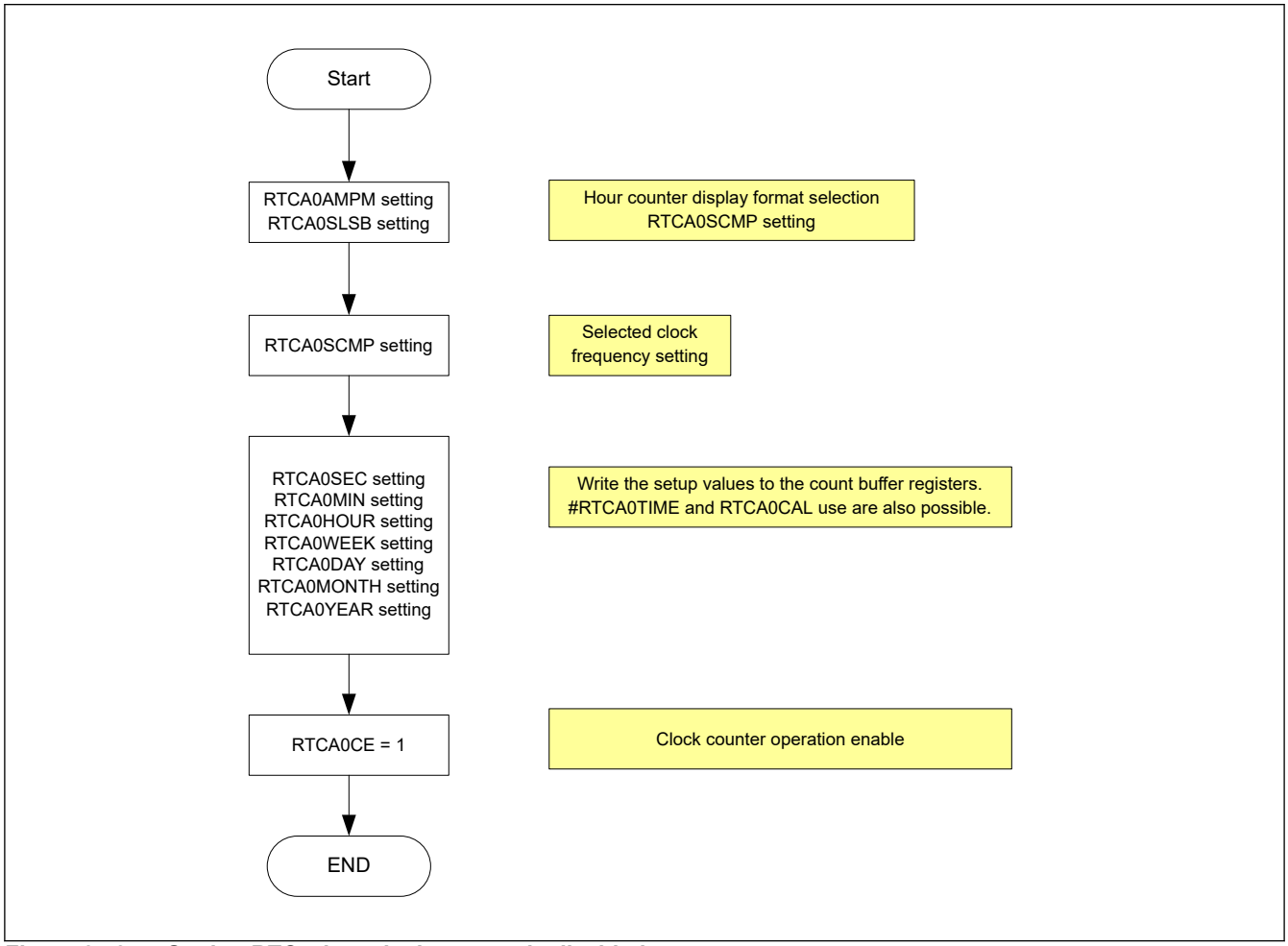


Figure 25.2 Setting RTC when clock counter is disabled

The internal clock counter operates in synchronization with PCLKRTC, and two PCLKRTC cycles are required until completion of the initial value setting procedure described in Figure 25.2. Therefore, PCLKL must be continuously supplied until completion of the initial value setting procedure. To stop PCLKL supply after the initial value setting procedure, check first that RTCA0CEST = 1 (clock counter operation enabled status).

25.3.1.2 Writing to Clock Counters While Clock Counter Operation is Enabled

Be sure to write to the RTCA0SEC, RTCA0MIN, RTCA0HOUR, RTCA0WEEK, RTCA0DAY, RTCA0MONTH, and RTCA0YEAR counters while the clock counter operation is enabled (RTCA0CE = 1) according to the flow in Figure 25.3.

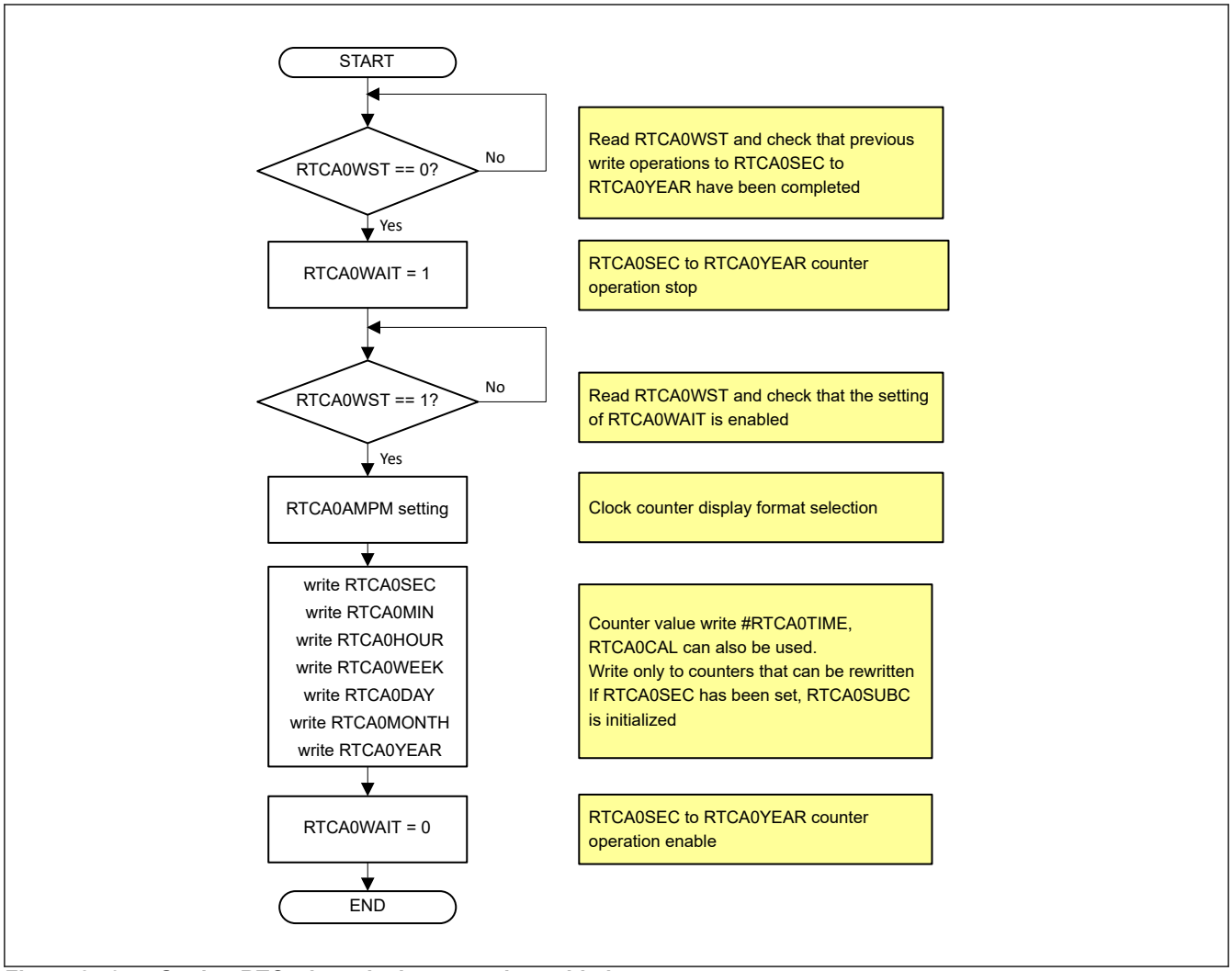


Figure 25.3 Setting RTC when clock counter is enabled

It is prohibited to write two or more times to the same register in the flow described in Figure 25.3. For example, it is prohibited to write twice to the second counter. End the programming flow within 1 second.

When RTCA0WAIT = 1 is set, the operation of RTCA0SEC (second counter buffer) stops. If RTCA0SUBC overflow occurs while RTCA0WAIT = 1, one overflow is held internally. However, if overflow occurs two or more times, the overflow count cannot be held.

The internal clock counter operates in synchronization with PCLKRTC. Two PCLKRTC cycles are required until completion of the above-described initial value setting procedure. Therefore, PCLKL must be continuously supplied until completion of the initial value setting procedure.

To stop PCLKL supply after the initial value setting procedure, check first that RTCA0WST = 0 (clock counter count-up status).

25.3.1.3 Reading Clock Counters While Clock Counter Operation is Enabled

There are two methods to read clock counters while clock counter operation is enabled (RTCA0CE = 1):

- Using Count Buffer Registers
- Reading Count Registers

The advantages and disadvantages of the two methods are mentioned in the next table.

Method	Advantage	Disadvantage
(1) Using count buffer registers	It is unnecessary to read clock counter several times such as in method (2), because clock counters are read synchronously.	Program wait*1 occurs from setting RTCA0WAIT = 1 to completion of data transfer.
(2) Reading count registers	Program wait does not occur.	When the reading of counter and overflow of RTCA0SUBC competed with each other, the clock counters are read several times again*2.

Note 1. The maximum of program wait is 3 clock cycles of PCLKL + 2 clock cycles of PCLKRTC.
Note 2. When reading of counter and overflow of sub-counter competed with each other, unknown value might be read.

(1) Using Count Buffer Registers

This is how to read clock counter value by using buffer registers. The value of clock count registers are transferred to count buffer registers by writing 1 in RTCA0WAIT. The count buffer registers are read after that. In this method, program wait occurs from setting RTCA0WAIT = 1 to completion of data transfer.

Reading clock counters while clock counter operation is enabled (RTCA0CE = 1)

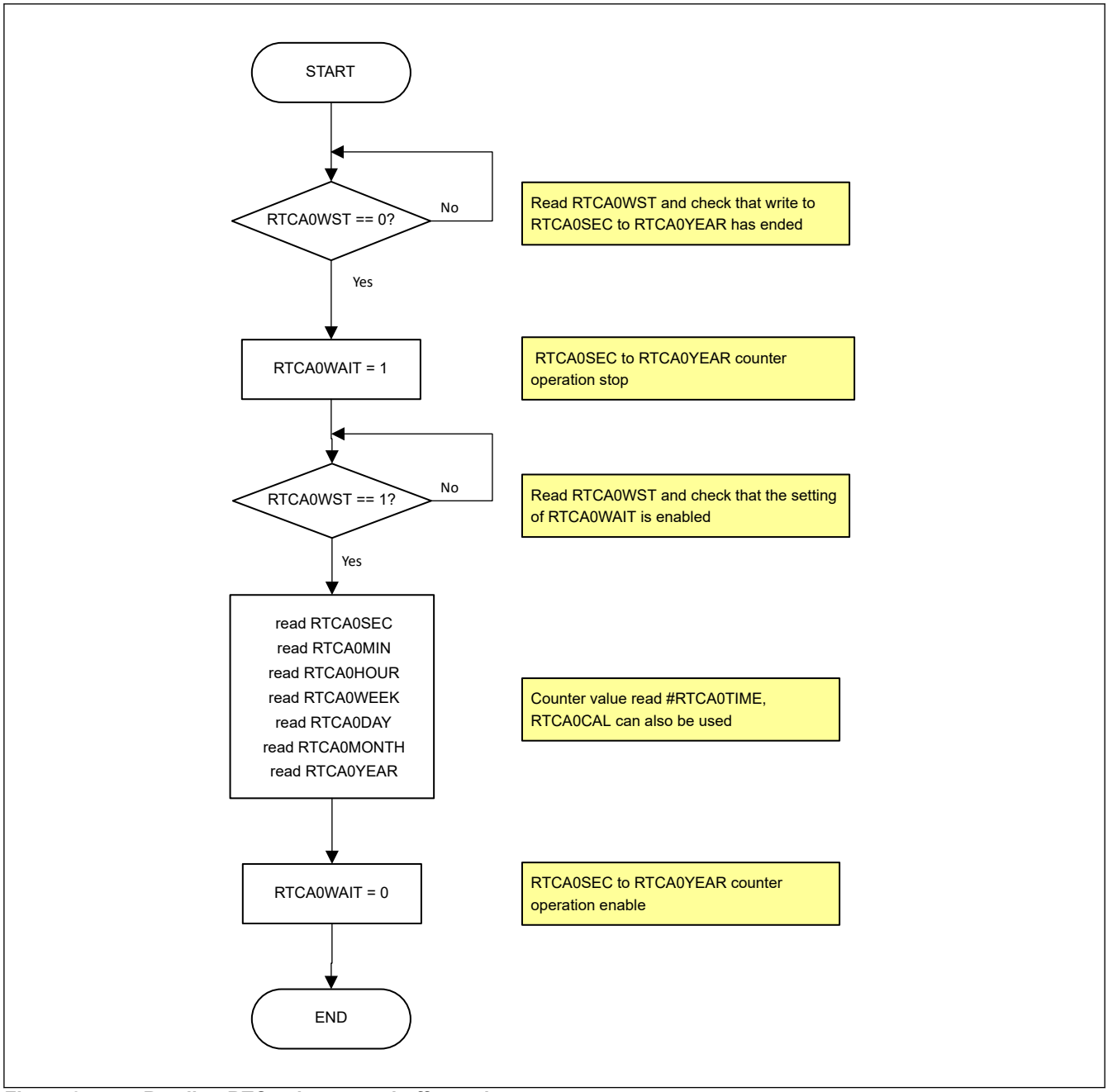


Figure 25.4 Reading RTC using count buffer registers

End this flow within 1 second.

When RTCA0WAIT = 1 is set, the operation of RTCA0SEC (second counter buffer) stops. If RTCA0SUBC overflow occurs while RTCA0WAIT = 1, one overflow is held internally. However, if overflow occurs two or more times, the overflow count cannot be held.

(2) Reading Count Registers

This is how to read clock counter immediately. RTCA0SECC (RTC Second Count Register) is read twice in the beginning and in the end. This is because it confirms whether overflow of sub-counter does not occur while counter is read. After that, the first read value is compared with the second read value. If the first read value is the same as the second read value, it is determined that overflow of sub-counter did not occur during counter read flow. If the first read value is not the same as the second read value, it is determined that overflow of sub-counter occurred during counter read flow, and it begins to read the clock counter again.

Reading clock counters while clock counter operation is enabled (RTCA0CE = 1)

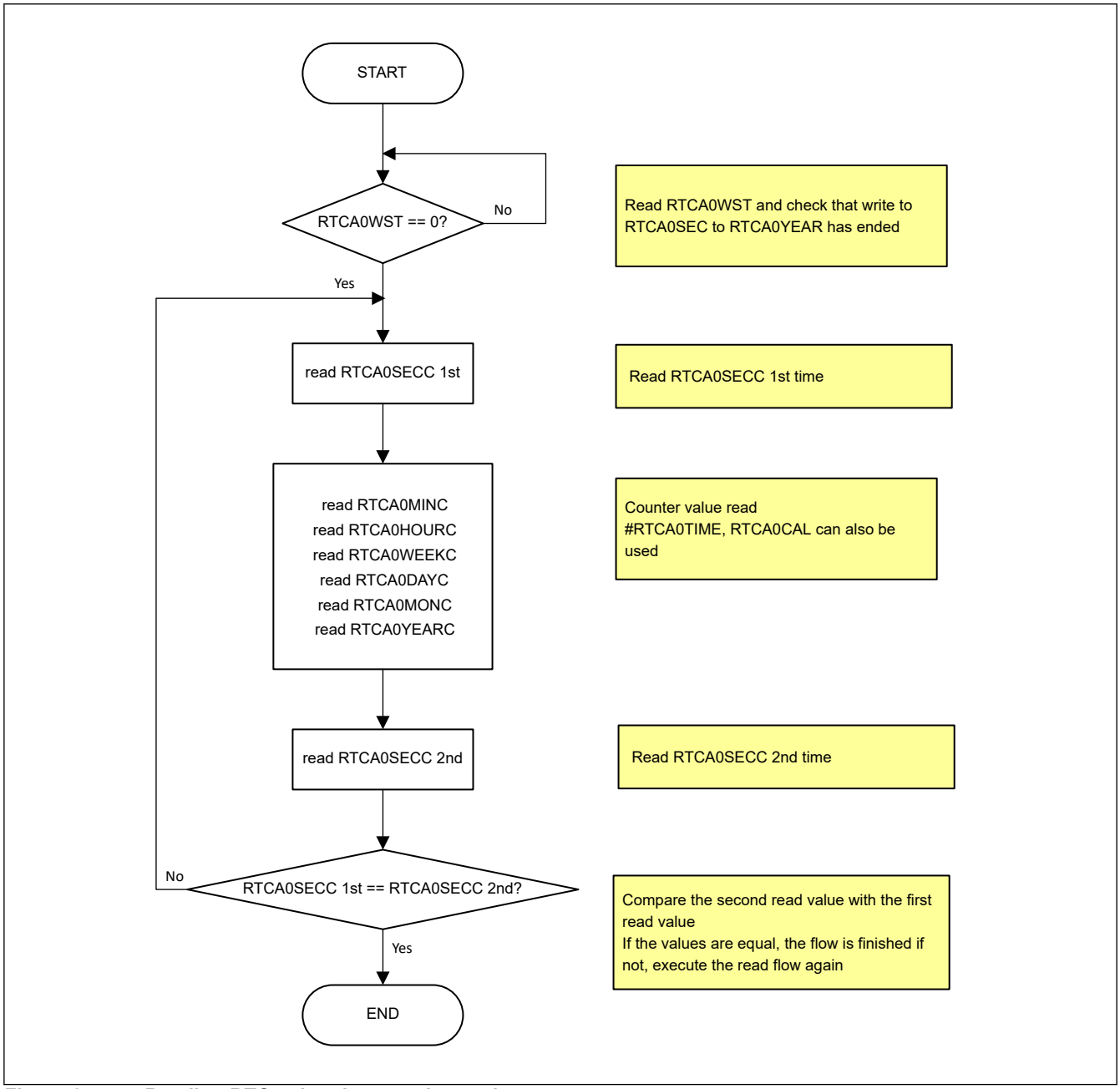


Figure 25.5 Reading RTC using the counting registers

End this flow within 1 second.

When it is not finished within one second, read clock counter value cannot be assured.

25.3.1.4 Reading RTCA0SRBU While Clock Counter Operation is Enabled

Be sure to read RTCA0SRBU while the clock counter operation is enabled (RTCA0CE = 1) according to the flow in Figure 25.6.

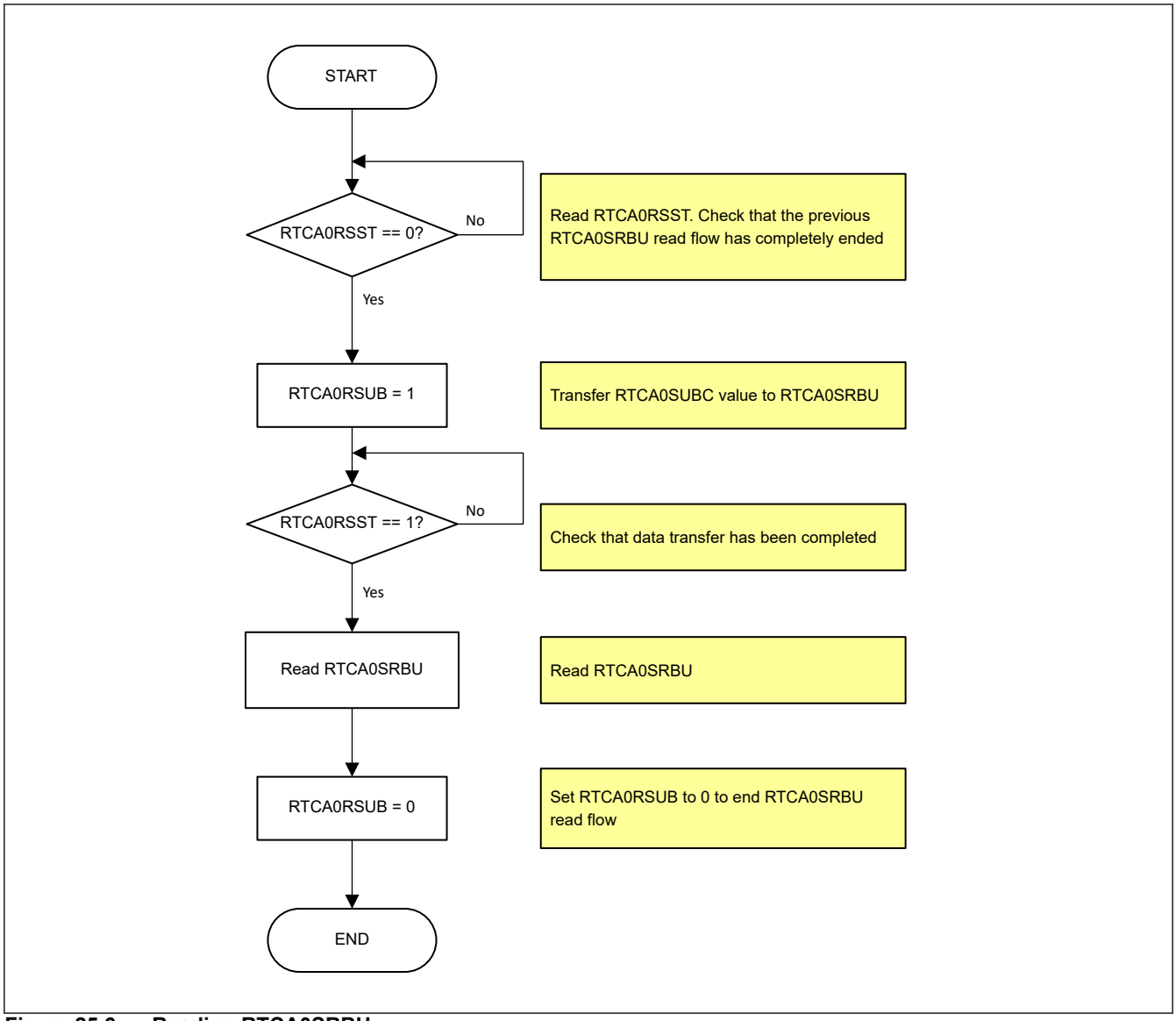


Figure 25.6 Reading RTCA0SRBU

25.3.1.5 Initialization of RTC While Clock Counter Operation is Enabled

Be sure to perform initialization of the RTC controller while the clock counter operation is enabled (RTCA0CE = 1) according to the flow in Figure 25.7. To restart the RTC controller operation at the end of the flow, implement the flow as described in section 25.3.1.1. Initial Setting.

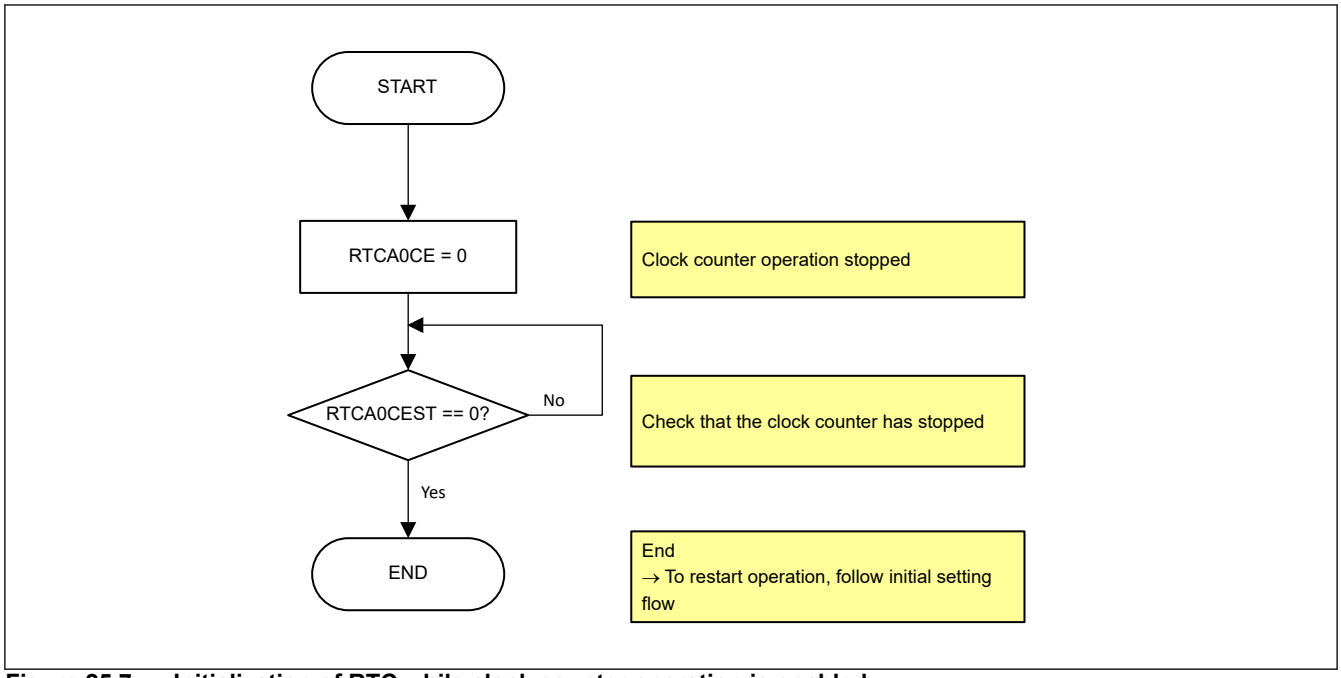


Figure 25.7 Initialization of RTC while clock counter operation is enabled

25.3.1.6 Writing to RTCA0SCMP During Clock Counter Operation

Be sure to write to RTCA0SCMP while the clock counter operation is enabled (RTCA0CE = 1) according to the flow in Figure 25.8.

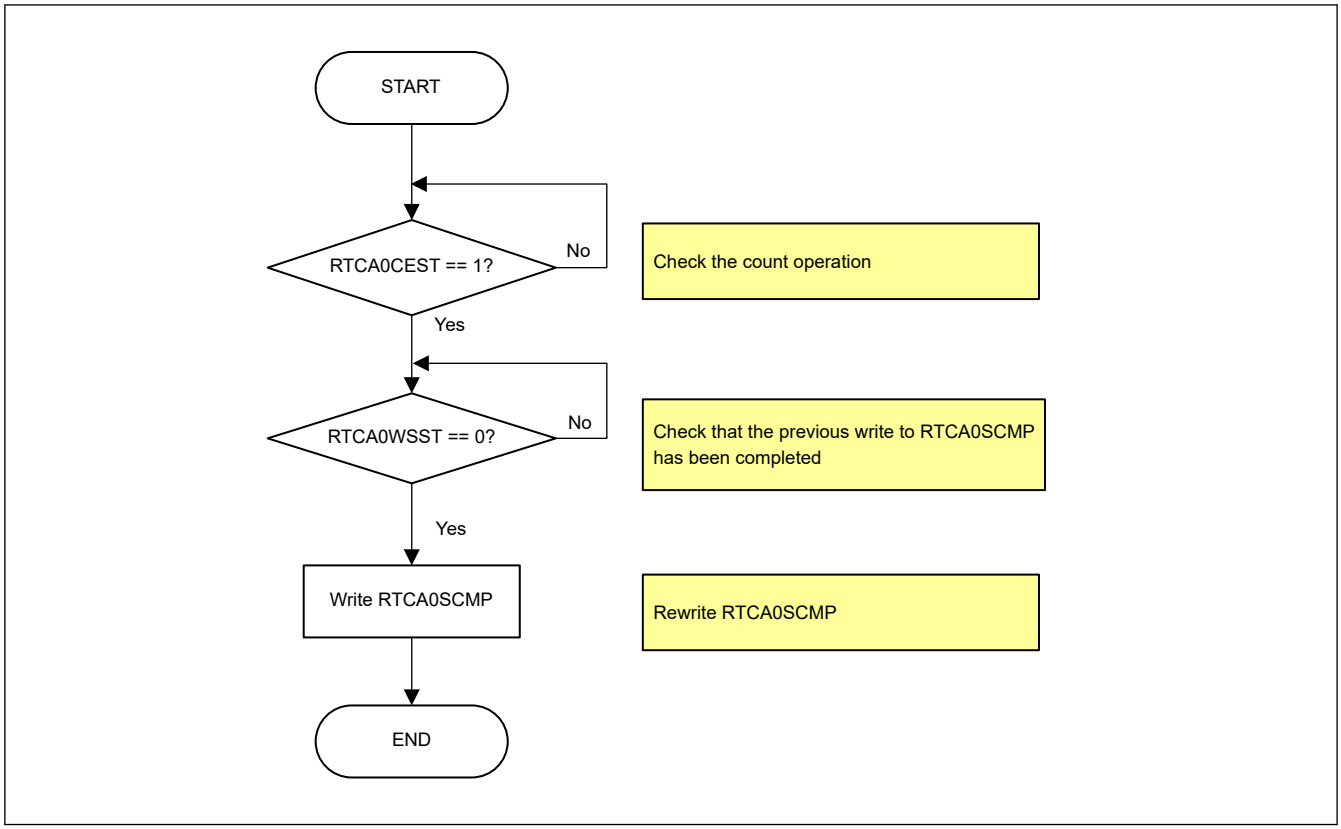


Figure 25.8 Writing to RTCA0SCMP

RTCA0WSST is set to 1 when RTCA0SCMP is written to while the clock counter operation is enabled (RTCA0CE = 1), and on completion of RTCA0SCMP write (RTCA0SUBC overflow), RTCA0WSST is cleared to 0. Because RTCA0WSST displays 1 for up to 1 second for this reason, caution is required when polling RTCA0WSST.

25.3.1.7 Changing the Setting of Fixed Interval Interrupt During Clock Counter Operation

Be sure to change the fixed interval interrupt setting while the clock counter operation is enabled (RTCA0CE = 1) according to the flow in Figure 25.9.

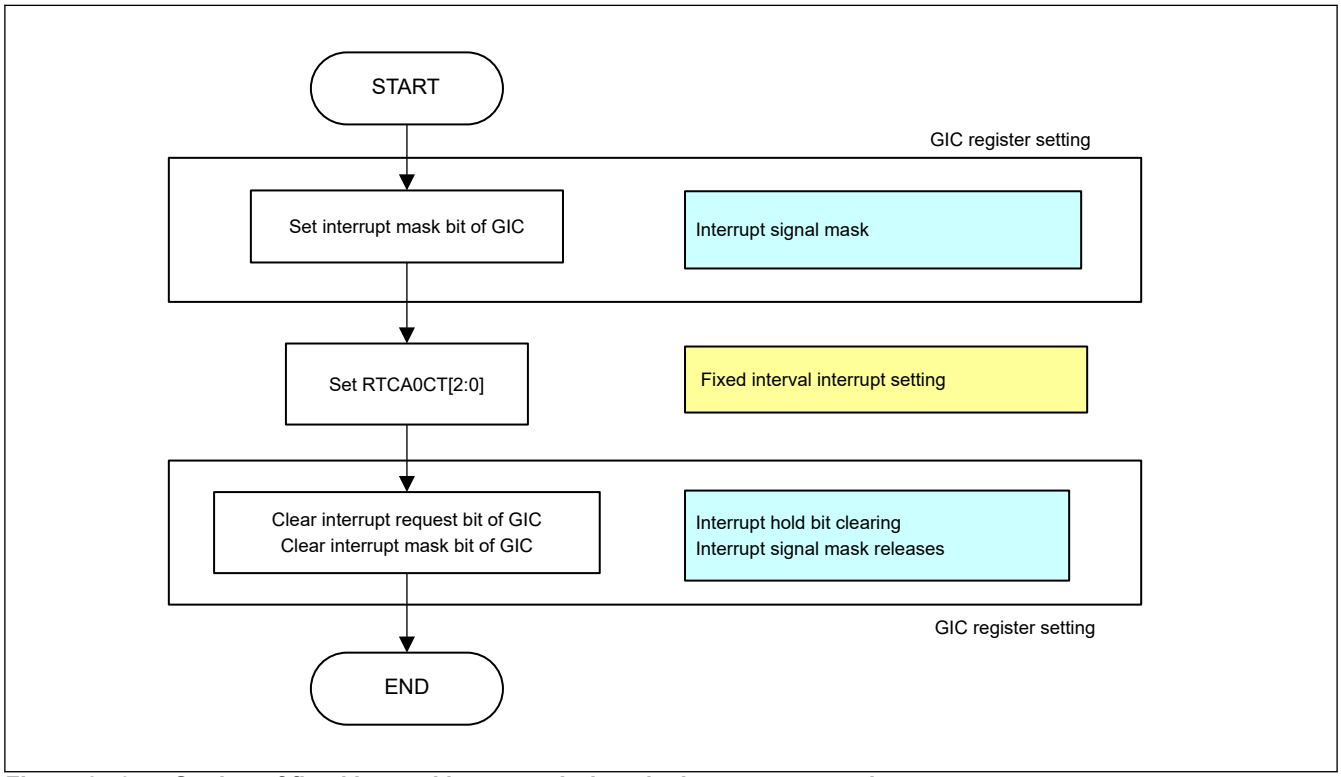


Figure 25.9 Setting of fixed interval interrupt during clock counter operation

25.3.1.8 Changing Alarm Setting During Clock Counter Operation

Be sure to change the alarm interrupt setting while the clock counter operation is enabled (RTCA0CE = 1) according to the flow in Figure 25.10.

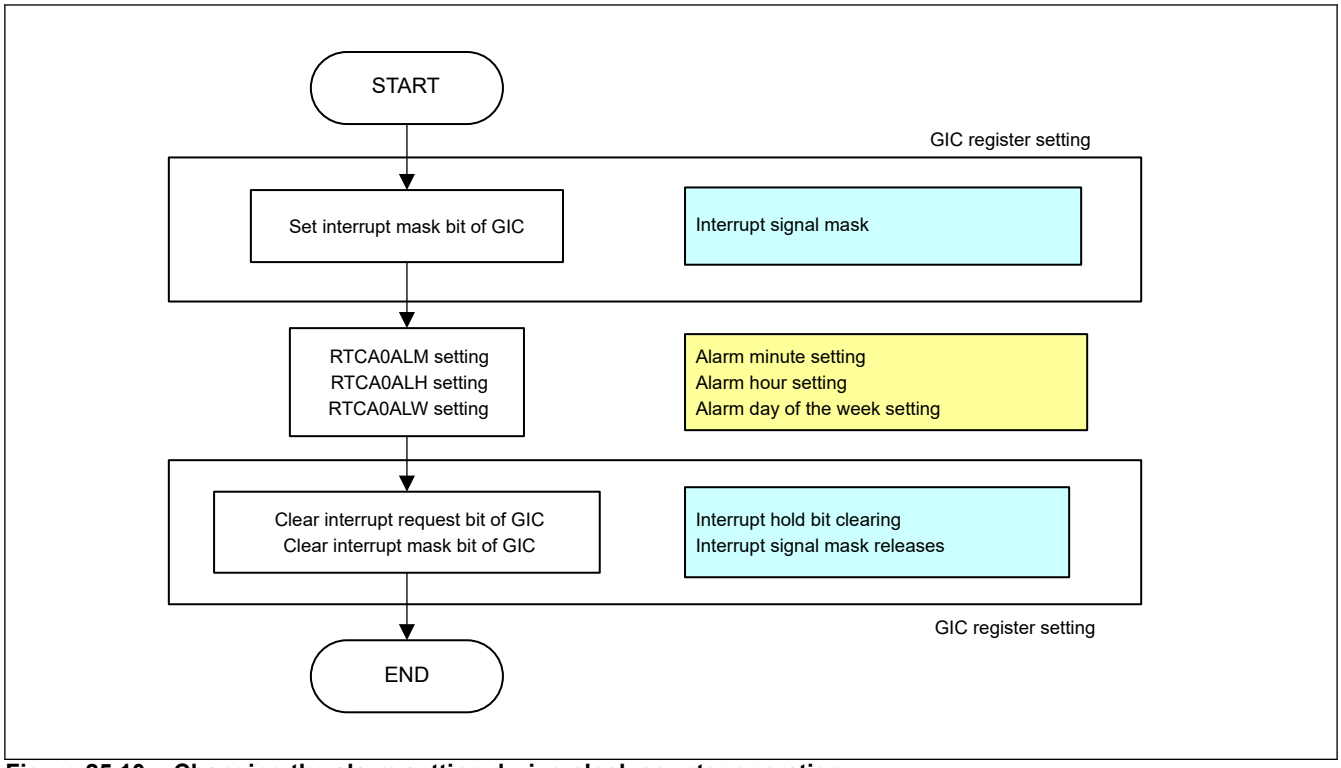


Figure 25.10 Changing the alarm setting during clock counter operation

26. Ethernet Subsystem

26.1 Overview

Ethernet Subsystem (Ethernet_SS) consists of Ethernet MAC (GMAC), EtherCAT Slave Controller (ESC), and RGMII/RMII converters. Three MAC-PHY interfaces supporting RGMII (port 2 only), MII, and RMII are shared with these functions.

Table 26.1 lists the specification of the Ethernet Subsystem and Figure 26.1 shows a block diagram of the Ethernet Subsystem.

Table 26.1 Ethernet Subsystem specifications

Item	Description
Ethernet function	• Gigabit Ethernet MAC (GMAC) • EtherCAT Slave Controller (ESC)
MAC-PHY interface	Supported interface by RGMII/RMII converter • RGMII (2.5 MHz/25 MHz/125 MHz) (port 2 only) • RMII (50 MHz) • MII (2.5 MHz/25 MHz)
Clock output for PHY	• 25 MHz reference clock output for each PHY • 50 MHz reference clock output for RMII
Event link (input)	• GMAC PTP Timer capture • EtherCAT DC capture (LATCH0/1)
Event link (output)	• PTP Timer pulse output • EtherCAT DC event (SYNC0/1)
Module-stop function	Module-stop state can be set to reduce power consumption each module

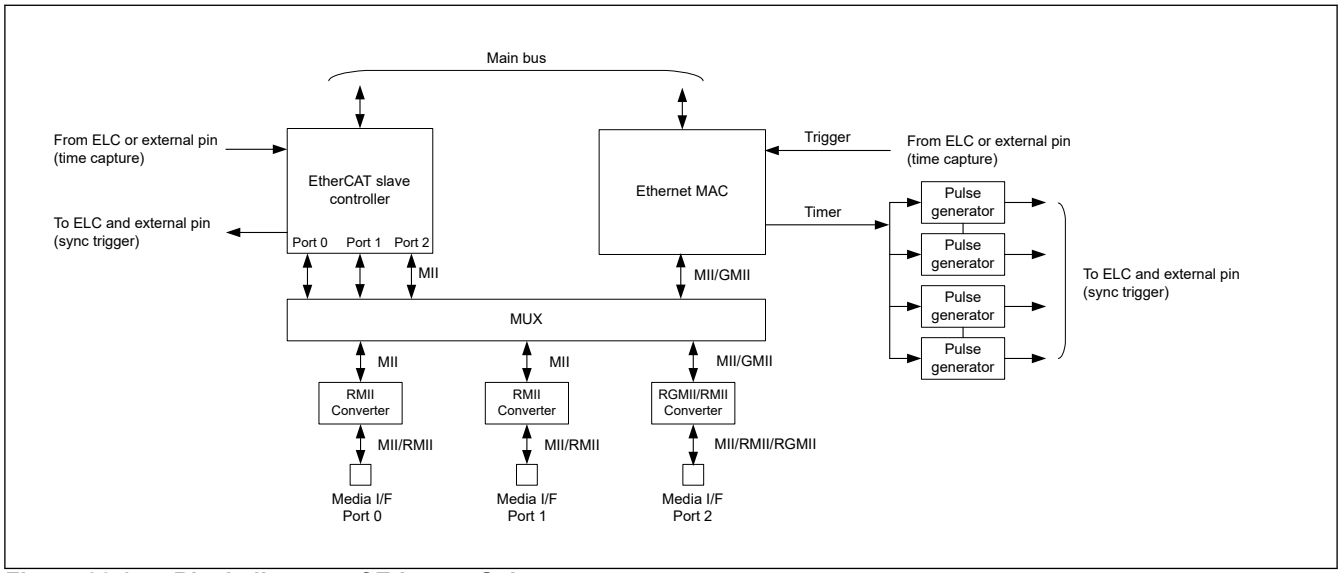


Figure 26.1 Block diagram of Ethernet Subsystem

Table 26.2 Ethernet Subsystem I/O pins (common PHY interface only)

Channel	Pin Name	I/O	Function
ETHn	ETHn_TXCLK	I/O	TX clock input pin (MII mode)/TX clock output pin (RGMII mode)
	ETHn_TXD0	Output	TX data 0 output pin (RGMII, RMII, and MII modes)
	ETHn_TXD1	Output	TX data 1 output pin (RGMII, RMII, and MII modes)
	ETHn_TXD2	Output	TX data 2 output pin (RGMII and MII modes)
	ETHn_TXD3	Output	TX data 3 output pin (RGMII and MII modes)
	ETHn_TXEN	Output	TX data enable pins (RMII and MII modes) TX data enable/TX data error (TX_CTL) pin (RGMII mode)
	ETH2_TXER	Output	TX data error output pin (MII mode)
	ETHn_RXCLK	Input	RX clock input pin (RGMII, RMII, and MII modes)
	ETHn_RXD0	Input	RX data 0 input pin (RGMII, RMII, and MII modes)
	ETHn_RXD1	Input	RX data 1 input pin (RGMII, RMII, and MII modes)
	ETHn_RXD2	Input	RX data 2 input pin (RGMII and MII modes)
	ETHn_RXD3	Input	RX data 3 input pin (RGMII and MII modes)
	ETHn_RXDV	Input	RX data valid pins (MII mode) Carrier sense/RX data valid (CRS_DV) pin (RMII mode) RX data valid/RX error (RX_CTL) pin (RGMII mode)
	ETHn_RXER	Input	RX data error pins (RMII and MII modes)
	ETH2_CRS	Input	Carrier sense pins (MII mode)
	ETH2_COL	Input	Collision detection pins (MII mode)

Note: n = 0 to 2

Table 26.3 Combination of MAC-PHY interface

Pin name	I/O	Normal mode*1			
		MII mode	RMII mode (REF_CLK input)	RMII mode (REF_CLK output)	RGMII mode*2
ETHn_TXCLK	I/O	TX_CLK (Input)	—	—	TXC (Output)
ETHn_TXD0	Output	TXD0	TXD0	TXD0	TXD0
ETHn_TXD1	Output	TXD1	TXD1	TXD1	TXD1
ETHn_TXD2	Output	TXD2	—	—	TXD2
ETHn_TXD3	Output	TXD3	—	—	TXD3
ETHn_TXEN	Output	TX_EN	TX_EN	TX_EN	TX_CTL
ETH2_TXER	Output	TX_ER	—	—	—
ETHn_RXCLK	Input	RX_CLK	REF_CLK	—	RXC
ETHn_RXD0	Input	RXD0	RXD0	RXD0	RXD0
ETHn_RXD1	Input	RXD1	RXD1	RXD1	RXD1
ETHn_RXD2	Input	RXD2	—	—	RXD2
ETHn_RXD3	Input	RXD3	—	—	RXD3
ETHn_RXDV	Input	RX_DV	CRS_DV	CRS_DV	RX_CTL
ETHn_RXER	Input	RX_ER	RX_ER (option)	RX_ER (option)	—
ETH2_CRS	Input	CRS	—	—	—
ETH2_COL	Input	COL	—	—	—
ETHn_REFCLK	Output	—	—	—	—
		(PHY reference clock)			
RMIIn_REFCLK	Output	—	—	REF_CLK	—

Note 1. n = 0 to 2
Note 2. n = 2 only

Table 26.4 MAC-PHY interface mode

Combination*1	ETH0	ETH1	ETH2
No.0	EtherCAT Slave Controller Port 0	EtherCAT Slave Controller Port 1	Ethernet MAC
No.1	EtherCAT Slave Controller Port 0	EtherCAT Slave Controller Port 1	EtherCAT Slave Controller Port 2

Note 1. This is the pin connection inside the Ethernet Subsystem. For example, if Ethernet MAC in ETH2 is not required for No.0, pins which are assigned to ETH2 can be used as another peripheral by pin mux. EtherMAC can be put to the module-stop state (clock is stopped).

26.2 Register Descriptions

26.2.1 PRCMD : Ethernet Protect Register

Base address: ETHSS = 0x8011_0000
Offset address: 0x000

Bit position:	31																												0
Bit field:	PRCMD[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PRCMD[31:0]	Permit write access to protected registers For detail, see Figure 26.2. 0x0000 00A5: DLE to STATE1 at IDLE 0x0000 0001: STATE1 to STATE2 at STATE1 0x0000 FFFE: STATE2 to STATE3 at STATE2 0x0000 0001: STATE3 to Release Protect at STATE3 0x0000 0000: Clear at Release Protect	R/W

The PRCMD register is used to permit write access to protected registers. The PRCMD register mitigates the risk when application system carries out unusual operations by software runaways.

Before setting the PRCMD register to 0x00000001, protected registers are denied write access (no error occurs and the registers are not updated by write data).

[Setting procedure]

1. Writing 0x000000A5 to PRCMD register
2. Writing 0x00000001 to PRCMD register
3. Writing 0x0000FFFE to PRCMD register
4. Writing 0x00000001 to PRCMD register

[Clearing condition]

- Writing 0x00000000 to PRCMD register

Figure 26.2 shows an ethernet protect command sequence.

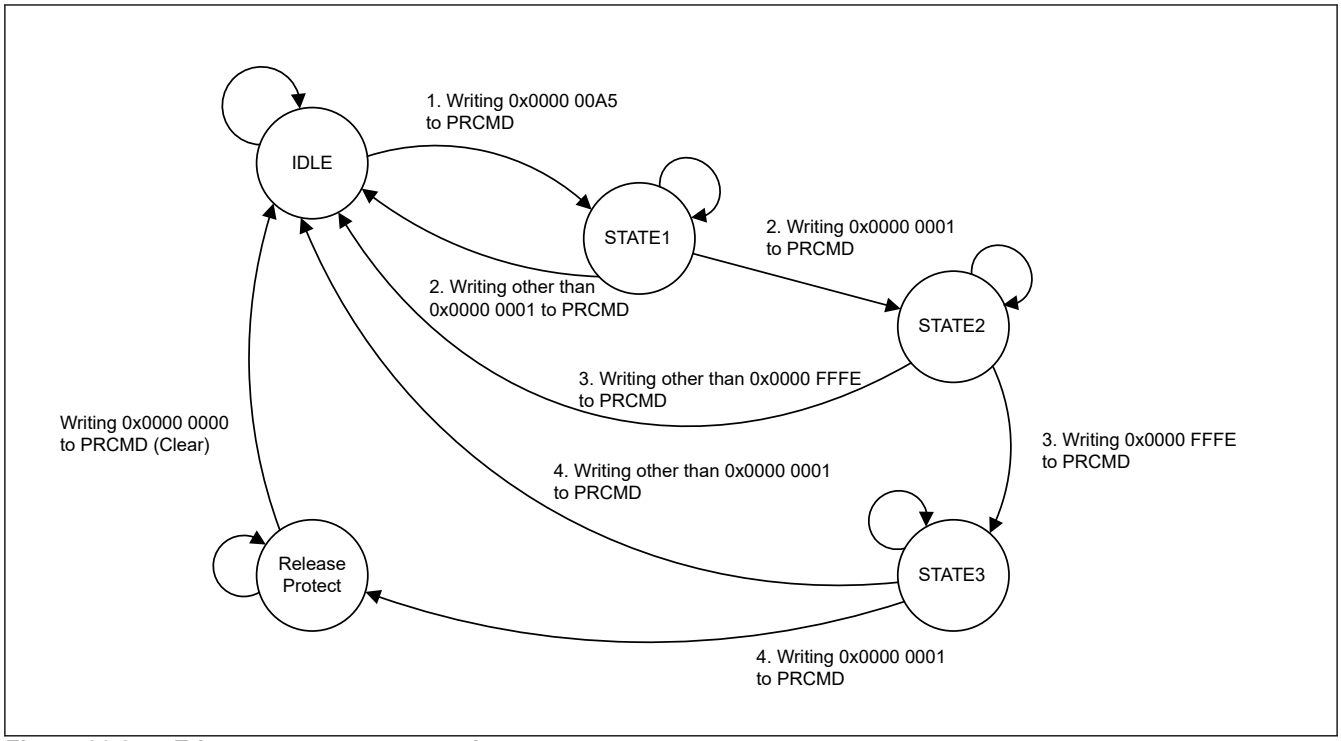


Figure 26.2 Ethernet protect command sequence

26.2.2 MODCTRL : Mode Control Register

Base address: ETHSS = 0x8011_0000

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SW_MODE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>0</td><td>SW_MODE</td><td>Selects the function of the Media interface of the MAC to be used
Media I/F connection
<table>
<tr><th rowspan="2">SW_MODE</th><th colspan="3">Media I/F</th></tr>
<tr><th>Port 0</th><th>Port 1</th><th>Port 2</th></tr>
<tr><td>0</td><td>ESC Port 0</td><td>ESC Port 1</td><td>GMAC Port</td></tr>
<tr><td>1</td><td>ESC Port 0</td><td>ESC Port 1</td><td>ESC Port 2</td></tr>
</table>
Internal Connection between GMAC and ESC
<table>
<tr><th rowspan="2">SW_
MODE</th><th rowspan="2">GMAC</th><th colspan="3">ESC</th></tr>
<tr><th>Port 0</th><th>Port 1</th><th>Port 2</th></tr>
<tr><td>0</td><td>Media I/F Port 2</td><td>Media I/F Port 0</td><td>Media I/F Port 1</td><td>—</td></tr>
<tr><td>1</td><td>—</td><td>Media I/F Port 0</td><td>Media I/F Port 1</td><td>Media I/F Port 2</td></tr>
</table></td><td>R/W</td></tr>
<tr><td>31:1</td><td>—</td><td>These bits are read as 0. The write value should be 0.</td><td>R/W</td></tr>
</table>

The MODCTRL register is used to switch the operation control of Switch mode. See Figure 26.1.

26.2.3 PTPMCTRL : PTP Mode Control Register

Base address: ETHSS = 0x8011_0000

Offset address: 0x00C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PTP_PLS_RSTn
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	PTP_PLS_RSTn	Reset control for Pulse Generator (unit 0 - 3) 0: Assert RESET 1: Release RESET	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

The PTPMCTRL register is used to select the unit number of PTP timer.

26.2.4 PHYLNK : Ethernet PHY Link Mode Register

Base address: ETHSS = 0x8011_0000

Offset address: 0x014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	CATLNK[2:0]			—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
6:4	CATLNK[2:0]	Specify the active level of the ESC_PHYLINKn signal using the EtherCAT interface CATLNK[n] control ESC_PHYLINKn signal (n = 0 to 2) If not used, set the initial value. 0: Active-Low PHYLINK signal 1: Active-High PHYLINK signal	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

The PHYLNK register is used to switch the polarity of the Ethernet Switch and EtherCAT Link.

26.2.5 CONVCTRLn : RGMII/RMII Converter n Control Register (n = 0 to 2)

Base address: ETHSS = 0x8011_0000

Offset address: 0x100 + 0x004 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RGMII_SPEED[1:0]		RGMII_DUPLEX	RGMII_LINK	—	RMII_CRS_MODE	RMII_RX_ER_EN	FULLD	—	—	—	CONV_MODE[4:0]				
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	CONV_MODE[4:0]	Converter operation mode **CONV_MODE[4:0] / Description** x0000b: MII mode (through mode) 00100b: RMII mode 10 Mbps REF_CLK input 00101b: RMII mode 100 Mbps REF_CLK input 10100b: RMII mode 10 Mbps REF_CLK output 10101b: RMII mode 100 Mbps REF_CLK output x1000b*1: RGMII mode 10 Mbps x1001b*1: RGMII mode 100 Mbps x1010b*1: RGMII mode 1000 Mbps Others: Setting prohibited	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	FULLD	Duplex mode (for RMII/RGMII mode) 0: Half-duplex 1: Full-duplex	R/W
9	RMII_RX_ER_EN	ETHn_RXER pin input control (only for RMII mode) 0: RX_ER is always 0 1: RX_ER signal	R/W
10	RMII_CRS_MODE	CRS detection mode (only for RMII mode) 0: CRS | TX_EN 1: CRS | TX_EN | RX_DV	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	RGMII_LINK	Indicates link status (only for RGMII mode)*1 0: Down 1: Up	R
13	RGMII_DUPLEX	Indicates duplex status (only for RGMII mode)*1 0: Half-duplex 1: Full-duplex	R
15:14	RGMII_SPEED[1:0]	Indicates link speed (only for RGMII mode)*1 0 0: 2.5 MHz 0 1: 25 MHz 1 0: 125 MHz 1 1: Reserved	R
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. RGMII mode can be supported for CONVCTRL2 register only.

The CONVCTRLn register is used to select the mode for RGMII/RMII.

Table 26.5 Ethernet interface support mode

Ethernet interface mode	Module	
	GMAC (Port 2)	EtherCAT (Port 0 / 1 / 2)
MII mode, 10 Mbps, Half-duplex	Supported	Not supported
MII mode, 10 Mbps, Full-duplex	Supported	Not supported
MII mode, 100 Mbps, Half-duplex	Supported	Not supported
MII mode, 100 Mbps, Full-duplex	Supported	Supported
RMII mode, 10 Mbps, Half-duplex, Reference clock input	Supported	Not supported
RMII mode, 10 Mbps, Full-duplex, Reference clock input	Supported	Not supported
RMII mode, 100 Mbps, Half-duplex, Reference clock input	Supported	Not supported
RMII mode, 100 Mbps, Full-duplex, Reference clock input	Supported	Supported
RMII mode, 10 Mbps, Half-duplex, Reference clock output	Supported	Not supported
RMII mode, 10 Mbps, Full-duplex, Reference clock output	Supported	Not supported
RMII mode, 100 Mbps, Half-duplex, Reference clock output	Supported	Not supported
RMII mode, 100 Mbps, Full-duplex, Reference clock output	Supported	Supported
RGMII mode, 10 Mbps, Half-duplex	Supported	Not supported
RGMII mode, 10 Mbps, Full-duplex	Supported	Not supported
RGMII mode, 100 Mbps, Half-duplex	Supported	Not supported
RGMII mode, 100 Mbps, Full-duplex	Supported	Not supported
RGMII mode, 1000 Mbps, Half-duplex	Supported	Not supported
RGMII mode, 1000 Mbps, Full-duplex	Supported	Not supported

26.2.6 CONVRST : RGMII/RMII Converter Reset Control Register

Base address: ETHSS = 0x8011_0000

Offset address: 0x114

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	PHYIR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	PHYIR[2:0]	Release reset of RGMII/RMII Converter. Bit 0 = port 0, bit 1= port 1, bit 2 = port 2. 0: Reset RGMII/RMII Converter 1: Activate RGMII/RMII Converter	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

The CONVRST register is used to control the reset for PHY I/F.

26.3 Operation

26.3.1 Ethernet Initializing Operation

For initializing operation, complete the following flowchart.

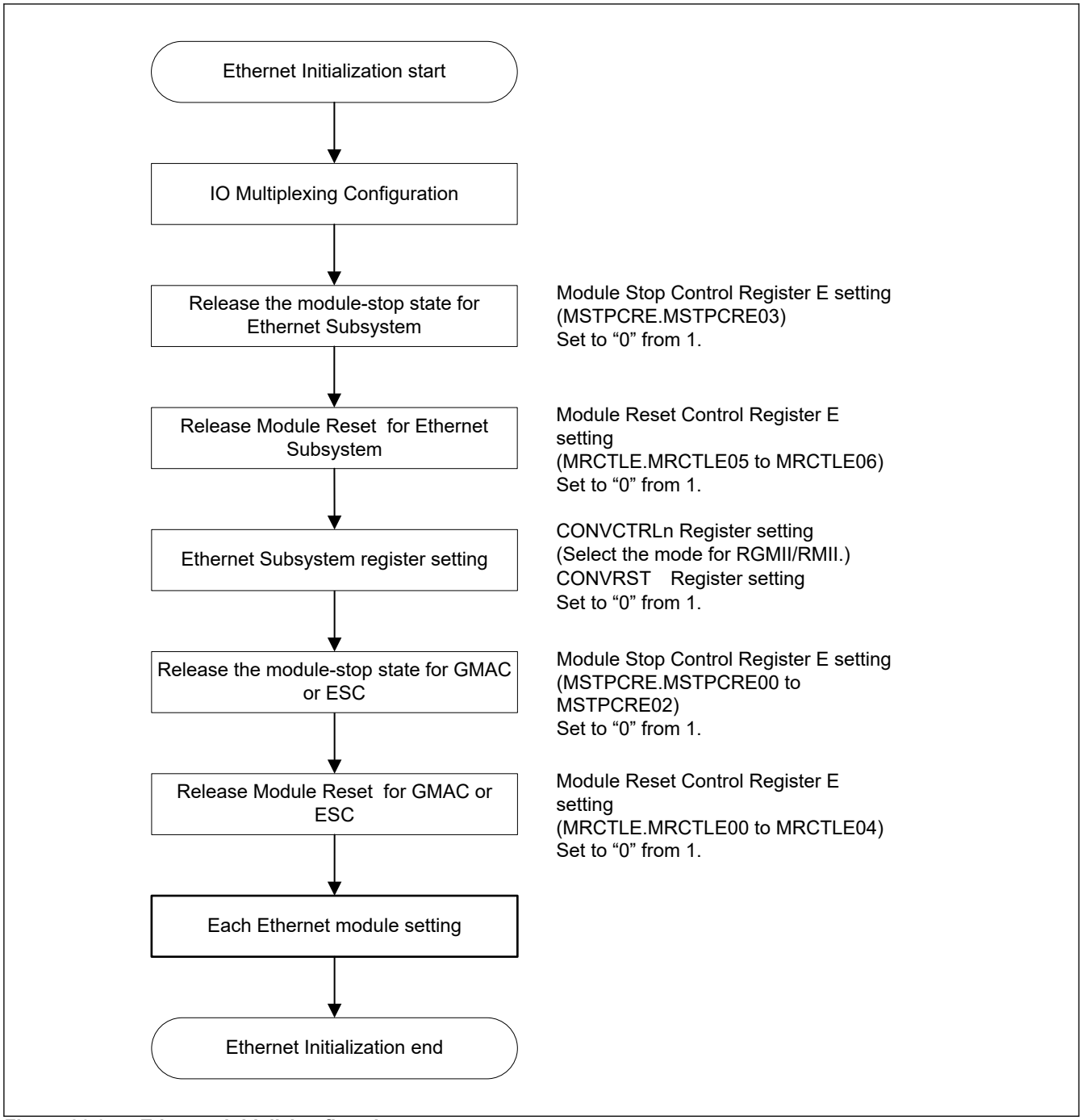


Figure 26.3 Ethernet initializing flowchart

Note: Setting EtherCAT Slave Controller (ESC) to module stop state and module reset release state is prohibited. (MSTPCRE.MSTPCRE02 = 1 and (MRCTLE.MRCTLE03 = 0 and/or MRCTLE.MRCTLE04 = 0))

27. Ethernet MAC (GMAC)

Portions Copyright © 2013, 2020 Synopsys, Inc. Used with permission. All rights reserved. Synopsys & DesignWare are registered trademarks of Synopsys, Inc.

27.1 Overview

The LSI includes an Ethernet MAC (GMAC) that:

- Is compliant with the following standards:
  - IEEE 802.3-2008 for Ethernet MAC, Gigabit Media Independent Interface (GMII), and Media Independent Interface (MII)
  - IEEE 1588-2008 v2 standard for precision networked clock synchronization, IEEE 1588-2008 v2 is compliant with Power IEEE-C37.238 profile
  - IEEE 802.3-az, version D2.0 for Energy Efficient Ethernet (EEE).
- Supports 10/100/1000 Mbps data transfer rates
- Supports both half-duplex and full-duplex operation
- Has programmable frame length to support both standard and jumbo ethernet frames of size up to 16 KB (16 KB - 1):
  - Jumbo mode support in cut-through mode only (not implemented in store and forward due to TX and RX FIFO size).
- Contains 17 MAC Address registers for the Address Filter block
- Supports variety of flexible addresses filtering modes:
  - Perfect (DA) address filters with masks for each byte
  - SA address comparison check with masks for each byte
  - 256-bit Hash filter for multicast and unicast (DA) addresses
  - Option to pass all multicast addressed frames
  - Promiscuous mode support to pass all frames without any filtering for network monitoring
  - Passes all incoming packets (as per filter) with a status report.
- Supports native DMA with simple independent channels transmit and receive engines:
  - 1 RX channel of 4 KB FIFO size for receive channel
  - 1 TX channel of 2 KB FIFO size for transmit channel
  - DMA implements dual buffer (ring) or linked list (chained) descriptor chaining.
- Supports advanced IEEE 1588-2002 & 2008 ethernet frame time stamping:
  - Timer module implemented
  - IEEE1588 time base information, with reference clock of 125 MHz
  - IEEE1588 external snapshot.
- Supports programmable CRC generation and checking
- Supports RMON statistics (L2 layer only)
- Supports Station Management Block and MDIO interface
- Is Ethernet Energy Efficiency compliant with IEEE 802.3az-2010:
  - Supports wakeup on LAN on magic packet and packet filtering
  - Supports energy efficiency feature (LPI mode)
  - Wakeup capability.
- Has interface for:
  - Native mode GMII (used for internal only), MII
  - No Native RMII and RGMII, managed by RMII/RGMII converter (connected on external pins).

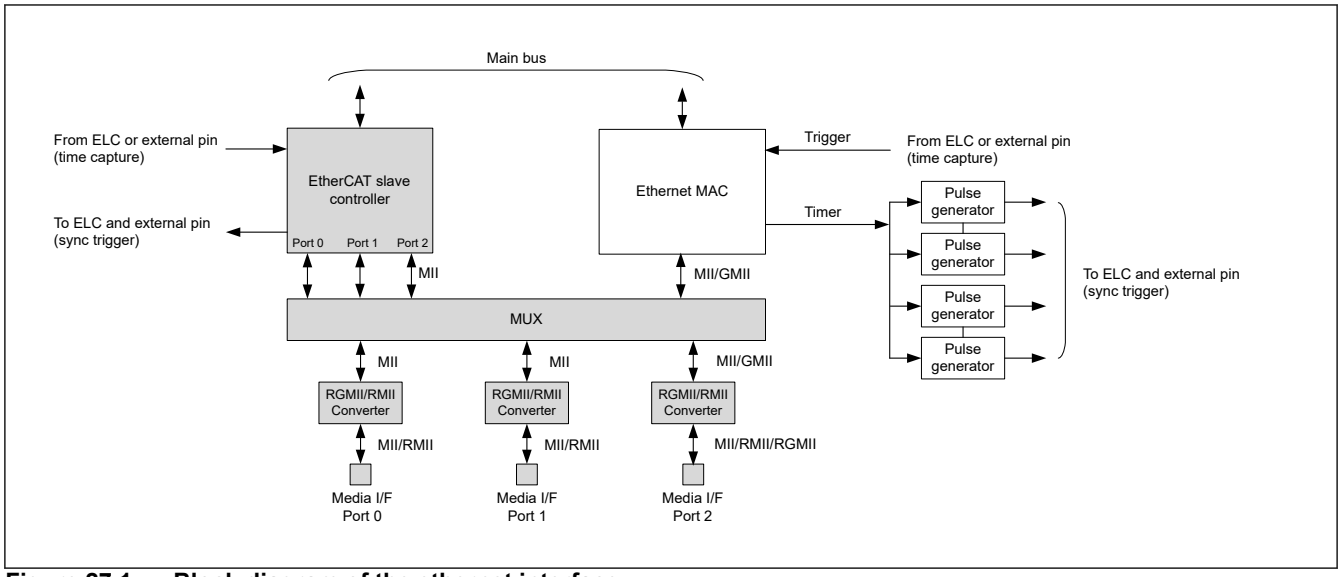


Figure 27.1 Block diagram of the ethernet interface

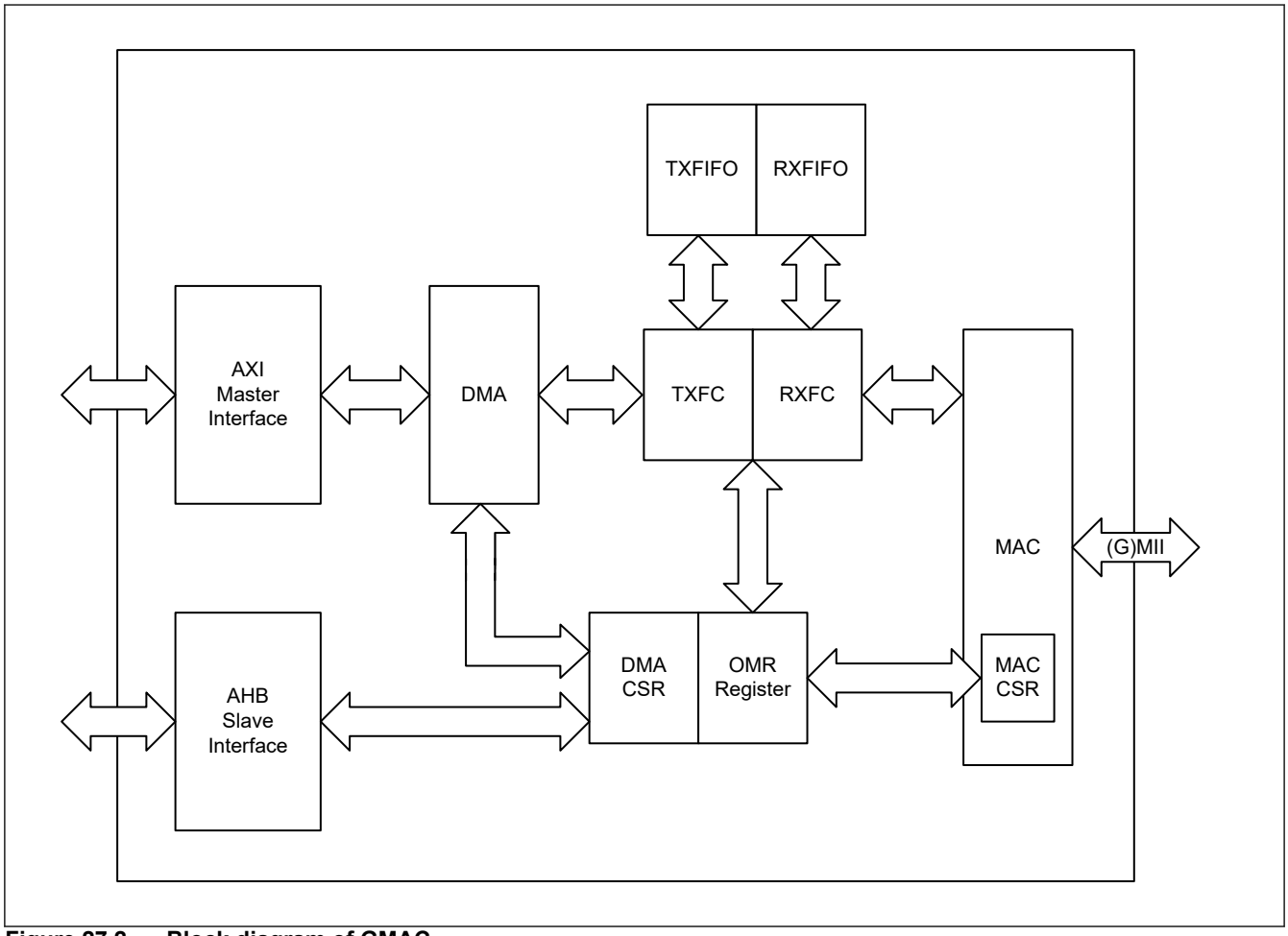


Figure 27.2 Block diagram of GMAC

Table 27.1 Input/output pins of the GMAC (1 of 2)

Pin name	I/O	Description
GMAC_PTPTRG0, GMAC_PTPTRG1	Input	PTP timer trigger signal input pin
GMAC_PTPOUT0, GMAC_PTPOUT1, GMAC_PTPOUT2, GMAC_PTPOUT3	Output	PTP timer pulse output pin

Table 27.1 Input/output pins of the GMAC (2 of 2)

Pin name	I/O	Description
GMAC_MDC	Output	Management data clock output pin
GMAC_MDIO	I/O	Management data I/O pin

27.2 Register Map

27.2.1 GMAC register map

Address	Register symbol	Register name
0x8010_0000	MAC_Configuration	MAC Configuration Register
0x8010_0004	MAC_Frame_Filter	MAC Frame Filter Register
0x8010_0010	GMII_Address	GMII Address Register
0x8010_0014	GMII_Data	GMII Data Register
0x8010_0018	Flow_Control	Flow Control Register
0x8010_001C	VLAN_Tag	VLAN Tag Register
0x8010_0020	Version	Version Register
0x8010_0024	Debug	Debug Register
0x8010_0028	Remote_Wake_Up_Frame_Filter	Remote Wake-Up Frame Filter Register
0x8010_002C	PMT_Control_Status	PMT Control and Status Register
0x8010_0030	LPI_Control_Status	LPI Control and Status Register
0x8010_0034	LPI_Timers_Control	LPI Timers Control Register
0x8010_0038	Interrupt_Status	Interrupt Status Register
0x8010_003C	Interrupt_Mask	Interrupt Mask Register
0x8010_0040 + 0x8 × n	MARn_H (n = 0 to 15)	MAC Address n High Register
0x8010_0044 + 0x8 × n	MARn_L (n = 0 to 15)	MAC Address n Low Register
0x8010_00DC	WDog_Timeout	Watchdog Timeout Register
0x8010_0100	MMC_Control	MMC Control Register
0x8010_0104	MMC_Receive_Interrupt	MMC Receive Interrupt Register
0x8010_0108	MMC_Transmit_Interrupt	MMC Transmit Interrupt Register
0x8010_010C	MMC_Receive_Interrupt_Mask	MMC Receive Interrupt Mask Register
0x8010_0110	MMC_Transmit_Interrupt_Mask	MMC Transmit Interrupt Mask Register
0x8010_0114	Tx_Octet_Count_Good_Bad	Transmit Octet Count for Good and Bad Frames
0x8010_0118	Tx_Frame_Count_Good_Bad	Transmit Frame Count for Good and Bad Frames
0x8010_011C	Tx_Broadcast_Frames_Good	Transmit Frame Count for Good Broadcast Frames
0x8010_0120	Tx_Multicast_Frames_Good	Transmit Frame Count for Good Multicast Frames
0x8010_0124	Tx_64Octets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 64 Byte Frames
0x8010_0128	Tx_65To127Octets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 65 to 127 Bytes Frames
0x8010_012C	Tx_128To255Octets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 128 to 255 Bytes Frames
0x8010_0130	Tx_256To511Octets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 256 to 511 Bytes Frames
0x8010_0134	Tx_512To1023Octets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 512 to 1023 Bytes Frames

Address	Register symbol	Register name
0x8010_0138	Tx_1024ToMaxOctets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 1024 to Maxsize Bytes Frames
0x8010_013C	Tx_Unicast_Frames_Good_Bad	Transmit Frame Count for Good and Bad Unicast Frames
0x8010_0140	Tx_Multicast_Frames_Good_Bad	Transmit Frame Count for Good and Bad Multicast Frames
0x8010_0144	Tx_Broadcast_Frames_Good_Bad	Transmit Frame Count for Good and Bad Broadcast Frames
0x8010_0148	Tx_Underflow_Error_Frames	Transmit Frame Count for Underflow Error Frames
0x8010_014C	Tx_Single_Collision_Good_Frames	Transmit Frame Count for Frames Transmitted after Single Collision
0x8010_0150	Tx_Multiple_Collision_Good_Frames	Transmit Frame Count for Frames Transmitted after Multiple Collision
0x8010_0154	Tx_Deferred_Frames	Transmit Frame Count for Deferred Frames
0x8010_0158	Tx_Late_Collision_Frames	Transmit Frame Count for Late Collision Error Frames
0x8010_015C	Tx_Excessive_Collision_Frames	Transmit Frame Count for Excessive Collision Error Frames
0x8010_0160	Tx_Carrier_Error_Frames	Transmit Frame Count for Carrier Sense Error Frames
0x8010_0164	Tx_Octet_Count_Good	Transmit Octet Count for Good Frames
0x8010_0168	Tx_Frame_Count_Good	Transmit Frame Count for Good Frames
0x8010_016C	Tx_Excessive_Deferral_Error	Transmit Frame Count for Excessive Deferral Error Frames
0x8010_0170	Tx_Pause_Frames	Transmit Frame Count for Good PAUSE Frames
0x8010_0174	Tx_VLAN_Frames_Good	Transmit Frame Count for Good VLAN Frames
0x8010_0178	Tx_OSize_Frames_Good	Transmit Frame Count for Good Oversize Frames
0x8010_0180	Rx_Frames_Count_Good_Bad	Receive Frame Count for Good and Bad Frames
0x8010_0184	Rx_Octet_Count_Good_Bad	Receive Octet Count for Good and Bad Frames
0x8010_0188	Rx_Octet_Count_Good	Receive Octet Count for Good Frames
0x8010_018C	Rx_Broadcast_Frames_Good	Receive Frame Count for Good Broadcast Frames
0x8010_0190	Rx_Multicast_Frames_Good	Receive Frame Count for Good Multicast Frames
0x8010_0194	Rx_CRC_Error_Frames	Receive Frame Count for CRC Error Frames
0x8010_0198	Rx_Alignment_Error_Frames	Receive Frame Count for Alignment Error Frames
0x8010_019C	Rx_Runt_Error_Frames	Receive Frame Count for Runt Error Frames
0x8010_01A0	Rx_Jabber_Error_Frames	Receive Frame Count for Jabber Error Frames
0x8010_01A4	Rx_Undersize_Frames_Good	Receive Frame Count for Undersize Frames
0x8010_01A8	Rx_Oversize_Frames_Good	Receive Frame Count for Oversize Frames
0x8010_01AC	Rx_64Octets_Frames_Good_Bad	Receive Frame Count for Good and Bad 64 Byte Frames
0x8010_01B0	Rx_65To127Octets_Frames_Good_Bad	Receive Frame Count for Good and Bad 65 to 127 Bytes Frames
0x8010_01B4	Rx_128To255Octets_Frames_Good_Bad	Receive Frame Count for Good and Bad 128 to 255 Bytes Frames
0x8010_01B8	Rx_256To511Octets_Frames_Good_Bad	Receive Frame Count for Good and Bad 256 to 511 Bytes Frames
0x8010_01BC	Rx_512To1023Octets_Frames_Good_Bad	Receive Frame Count for Good and Bad 512 to 1,023 Bytes Frames
0x8010_01C0	Rx_1024ToMaxOctets_Frames_Good_Bad	Receive Frame Count for Good and Bad 1,024 to Maxsize Bytes Frames

Address	Register symbol	Register name
0x8010_01C4	Rx_Unicast_Frames_Good	Receive Frame Count for Good Unicast Frames
0x8010_01C8	Rx_Length_Error_Frames	Receive Frame Count for Length Error Frames
0x8010_01CC	Rx_Out_Of_Range_Type_Frames	Receive Frame Count for Out of Range Frames
0x8010_01D0	Rx_Pause_Frames	Receive Frame Count for PAUSE Frames
0x8010_01D4	Rx_FIFO_Overflow_Frames	Receive Frame Count for FIFO Overflow Frames
0x8010_01D8	Rx_VLAN_Frames_Good_Bad	Receive Frame Count for Good and Bad VLAN Frames
0x8010_01DC	Rx_Watchdog_Error_Frames	Receive Frame Count for Watchdog Error Frames
0x8010_01E0	Rx_Receive_Error_Frames	Receive Frame Count for Receive Error Frames
0x8010_01E4	Rx_Control_Frames_Good	Receive Frame Count for Good Control Frames
0x8010_0500 + 0x4 × n	HASH_TABLE_REGn (n = 0 to 7)	Hash Table Register n
0x8010_0588	VLAN_Hash_Table_Reg	VLAN Hash Table Register
0x8010_0700	Timestamp_Control	Timestamp Control Register
0x8010_0704	Sub_Second_Increment	Sub-Second Increment Register
0x8010_0708	System_Time_Seconds	System Time - Seconds Register
0x8010_070C	System_Time_Nanoseconds	System Time - Nanoseconds Register
0x8010_0710	System_Time_Seconds_Update	System Time - Seconds Update Register
0x8010_0714	System_Time_Nanoseconds_Update	System Time - Nanoseconds Update Register
0x8010_0718	Timestamp_Addend	Timestamp Addend Register
0x8010_071C	Target_Time_Seconds	Target Time Seconds Register
0x8010_0720	Target_Time_Nanoseconds	Target Time Nanoseconds Register
0x8010_0728	Timestamp_Status	Timestamp Status Register
0x8010_0730	Auxiliary_Timestamp_Nanoseconds	Auxiliary Timestamp - Nanoseconds Register
0x8010_0734	Auxiliary_Timestamp_Seconds	Auxiliary Timestamp - Seconds Register
0x8010_0800	MAR16_H	MAC Address16 High Register
0x8010_0804	MAR16_L	MAC Address16 Low Register
0x8010_0808	MAR17_H	MAC Address17 High Register
0x8010_080C	MAR17_L	MAC Address17 Low Register
0x8010_1000	Bus_Mode	Bus Mode Register
0x8010_1004	Transmit_Poll_Demand	Transmit Poll Demand Register
0x8010_1008	Receive_Poll_Demand	Receive Poll Demand Register
0x8010_100C	Receive_Descriptor_List_Address	Receive Descriptor List Address Register
0x8010_1010	Transmit_Descriptor_List_Address	Transmit Descriptor List Address Register
0x8010_1014	Status	Status Register
0x8010_1018	Operation_Mode	Operation Mode Register
0x8010_101C	Interrupt_Enable	Interrupt Enable Register
0x8010_1020	Missed_Frame_And_Buffer_Overflow_Counter	Missed Frame and Buffer Overflow Counter Register
0x8010_1024	Receive_Interrupt_Watchdog_Timer	Receive Interrupt Watchdog Timer Register
0x8010_1028	AXI_Bus_Mode	AXI Bus Mode Register
0x8010_102C	AXI_Status	AXI Status Register
0x8010_1048	Current_Host_Transmit_Descriptor	Current Host Transmit Descriptor Register
0x8010_104C	Current_Host_Receive_Descriptor	Current Host Receive Descriptor Register
0x8010_1050	Current_Host_Transmit_Buffer_Address	Current Host Transmit Buffer Address Register

Address	Register symbol	Register name
0x8010_1054	Current_Host_Receive_Buffer_Address	Current Host Receive Buffer Address Register
0x8010_1058	HW_Feature	HW Feature Register
0x8011_0400	GMACTRGSEL	GMAC PTP Trigger Select Register
0x8011_0800 + 0x100 × n	SWTMENn (n = 0 to 3)	PTP Timer Pulse Output Enable n Register
0x8011_0804 + 0x100 × n	SWTMSTSECn (n = 0 to 3)	PTP Timer Pulse Start Second n Register
0x8011_0808 + 0x100 × n	SWTMSTNSn (n = 0 to 3)	PTP Timer Pulse Start Nanosecond n Register
0x8011_080C + 0x100 × n	SWTMPSECn (n = 0 to 3)	PTP Timer Pulse Period Second n Register
0x8011_0810 + 0x100 × n	SWTMPNSn (n = 0 to 3)	PTP Timer Pulse Period Nanosecond n Register
0x8011_0814 + 0x100 × n	SWTMWTHn (n = 0 to 3)	PTP Timer Pulse Width n Register
0x8011_0818 + 0x100 × n	SWTMMAXPn (n = 0 to 3)	PTP Timer Pulse Max Second n Register
0x8011_081C + 0x100 × n	SWTMLATSECn (n = 0 to 3)	PTP Timer Pulse Latch Second n Register
0x8011_0820 + 0x100 × n	SWTMLATNSn (n = 0 to 3)	PTP Timer Pulse Latch Nanosecond n Register

27.3 Register Description

27.3.1 MAC_Configuration : MAC Configuration Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	TWOKPE	—	CST	—	WD	JD	BE	JE	IFG[2:0]			DCRS
Value after reset:	x	x	x	x	0	x	0	x	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PS	FES	DO	LM	DM	IPC	DR	—	ACS	BL[1:0]		DC	TE	RE	PRELEN[1:0]	
Value after reset:	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PRELEN[1:0]	Preamble Length for Transmit Frames These bits control the number of preamble bytes that are added to the beginning of every transmit frame. The preamble reduction occurs only when the GMAC is operating in the full-duplex mode. 0 0: 7 bytes of preamble 0 1: 5 bytes of preamble 1 0: 3 bytes of preamble 1 1: Reserved	R/W
2	RE	Receiver Enable When this bit is set, the receive state machine of the GMAC is enabled for receiving frames from the GMII or MII. When this bit is reset, the GMAC receive state machine is disabled after completion of the reception of the current frame, and does not receive any additional frames from the GMII or MII.	R/W
3	TE	Transmitter Enable When this bit is set, the transmit state machine of the GMAC is enabled for transmission on the GMII or MII. When this bit is reset, the GMAC transmit state machine is disabled after completion of the transmission of the current frame, and does not transmit any additional frames.	R/W

Bit	Symbol	Function	R/W
4	DC	Deferral Check When this bit is set, the deferral check function is enabled in the GMAC. The GMAC issues a Frame Abort status, along with the excessive deferral error bit set in the transmit frame status, when the transmit state machine is deferred for more than 24,288-bit times in the 10 or 100 Mbps mode. If the GMAC is configured for 1000 Mbps operation or if the Jumbo frame mode is enabled in the 10 or 100 Mbps mode, the threshold for deferral is 155,680 bits times. Deferral begins when the transmitter is ready to transmit but is prevented because of an active carrier sense signal (CRS) on GMII or MII. The defer time is not cumulative. For example, if the transmitter defers for 10,000 bit times because the CRS signal is active and then the CRS signal becomes inactive, the transmitter transmits and collision occurs. Because of collision, the transmitter needs to back off and then defer again after back off is complete. In such a scenario, the deferral timer is reset to 0 and it is restarted. When this bit is reset, the deferral check function is disabled and the GMAC defers until the CRS signal goes inactive. This bit is applicable only in the half-duplex mode.	R/W
6:5	BL[1:0]	Back-Off Limit The Back-Off limit determines the random integer number (r) of slot time delays (4,096-bit times for 1000 Mbps and 512-bit times for 10/100 Mbps) for which the GMAC waits before rescheduling a transmission attempt during retries after a collision. This bit is applicable only in the half-duplex mode. where n = retransmission attempt. The random integer r takes the value in the range 0 ≤ r < kth power of 2 0 0: k = min (n, 10) 0 1: k = min (n, 8) 1 0: k = min (n, 4) 1 1: k = min (n, 1)	R/W
7	ACS	Automatic Pad or CRC Stripping When this bit is set, the GMAC strips the Pad or FCS field on the incoming frames only if the value of the length field is less than 1,536 bytes. All received frames with length field greater than or equal to 1,536 bytes are passed to the application without stripping the Pad or FCS field. When this bit is reset, the GMAC passes all incoming frames, without modifying them, to the host.	R/W
8	—	This bit is read as 0. The write value should be 0.	R/W
9	DR	Disable Retry When this bit is set, the GMAC attempts only one transmission. When a collision occurs on the GMII or MII interface, the GMAC ignores the current frame transmission and reports a Frame Abort with excessive collision error in the transmit frame status. When this bit is reset, the GMAC attempts retries based on the settings of the BL field (bits [6:5]). This bit is applicable only in the half-duplex mode.	R/W
10	IPC	Checksum Offload When this bit is set, the GMAC calculates the 16-bit 1's complement of the 1's complement sum of all received Ethernet frame payloads. It also checks whether the IPv4 Header checksum (assumed to be bytes 25-26 or 29-30 (VLAN-tagged) of the received Ethernet frame) is correct for the received frame and gives the status in the receive status word. When this bit is reset, this function is disabled. When this bit is set, enables the IPv4 header checksum checking and IPv4 or IPv6 TCP, UDP, or ICMP payload checksum checking. When this bit is reset, the COE function in the receiver is disabled and the corresponding Payload Checksum Error and IP Header Checksum Error status bits are cleared.	R/W
11	DM	Duplex Mode When this bit is set, the GMAC operates in the full-duplex mode where it can transmit and receive simultaneously.	R/W
12	LM	Loopback Mode When this bit is set, the GMAC operates in the loopback mode at GMII or MII. The receive clock is required for the loopback to work properly, because the transmit clock is not looped back internally.	R/W
13	DO	Disable Receive Own When this bit is set, the GMAC disables the reception of frames when the Transmit Data Enable is asserted in the half-duplex mode. When this bit is reset, the GMAC receives all packets that are given by the PHY while transmitting. This bit is not applicable if the GMAC is operating in the full-duplex mode.	R/W

Bit	Symbol	Function	R/W
14	FES	Speed This bit selects the speed in the MII interface. 0: 10 Mbps 1: 100 Mbps	R/W
15	PS	Port Select This bit selects the Ethernet line speed. In 10 or 100 Mbps operations, this bit, along with FES bit, selects the exact line speed. 0: For 1000 Mbps operations 1: For 10 or 100 Mbps operations	R/W
16	DCRS	Disable Carrier Sense During Transmission When set high, this bit makes the GMAC transmitter ignore the (G)MII CRS signal during frame transmission in the half-duplex mode. This request results in no errors generated because of Loss of Carrier or No Carrier during such transmission. When this bit is low, the GMAC transmitter generates such errors because of Carrier Sense and can even abort the transmissions.	R/W
19:17	IFG[2:0]	Inter-Frame Gap These bits control the minimum IFG between frames during transmission. In the half-duplex mode, the minimum IFG can be configured only for 64-bit times (IFG = 100b). Lower values are not considered. In the 1000-Mbps mode, the minimum IFG supported is 80-bit times and above. When a JAM pattern is being transmitted because of backpressure activation, the GMAC does not consider the minimum IFG. 0 0 0: 96-bit times 0 0 1: 88-bit times 0 1 0: 80-bit times ⋮ 1 1 1: 40-bit times	R/W
20	JE	Jumbo Frame Enable When this bit is set, the GMAC allows Jumbo frames of 9,018 bytes (9,022 bytes for VLAN tagged frames) without reporting a giant frame error in the receive frame status.	R/W
21	BE	Frame Burst Enable When this bit is set, the GMAC allows frame bursting during transmission in the GMII half-duplex mode.	R/W
22	JD	Jabber Disable When this bit is set, the GMAC disables the jabber timer on the transmitter. The GMAC can transfer frames of up to 16,384 bytes. When this bit is reset, the GMAC cuts off the transmitter if the application sends out more than 2,048 bytes of data (10,240 if JE is set high) during transmission.	R/W
23	WD	Watchdog Disable When this bit is set, the GMAC disables the watchdog timer on the receiver. The GMAC can receive frames of up to 16,384 bytes. When this bit is reset, the GMAC does not allow a receive frame which more than 2,048 bytes (10,240 if JE is set high) or the value programmed in Watchdog Timeout Register (WDog_Timeout). The GMAC cuts off any bytes received after the watchdog limit number of bytes.	R/W
24	—	This bit is read as 0. The write value should be 0.	R/W
25	CST	CRC Stripping for Type Frames When this bit is set, the last 4 bytes (FCS) of all frames of Ether type (Length/Type field greater than or equal to 1,536) are stripped and dropped before forwarding the frame to the application.	R/W
26	—	This bit is read as 0. The write value should be 0.	R/W
27	TWOKPE	IEEE 802.3 as Support for 2 K Packets When set, the GMAC considers all frames, with up to 2,000 bytes length, as normal packets. When bit JE[20] is not set, the GMAC considers all received frames of size more than 2 KB as Giant frames. When this bit is reset and bit JE[20] is not set, the GMAC considers all received frames of size more than 1,518 bytes (1,522 bytes for tagged) as Giant frames. When bit [20] is set, setting this bit has no effect on Giant Frame status.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

27.3.2 MAC_Frame_Filter : MAC Frame Filter Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RA	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VTFE
Value after reset:	0	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	HPF	SAF	SAIF	PCF[1:0]		DBF	PM	DAIF	HMC	HUC	PR
Value after reset:	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PR	Promiscuous Mode When this bit is set, the Address Filter module passes all incoming frames regardless of its destination or source address. The SA or DA Filter Fails status bits of the Receive Status Word are cleared when PR is set.	R/W
1	HUC	Hash Unicast When set, GMAC performs destination address filtering of unicast frames according to the hash table. When reset, the GMAC performs a perfect destination address filtering for unicast frames, that is, it compares the DA field with the values programmed in DA registers.	R/W
2	HMC	Hash Multicast When set, GMAC performs destination address filtering of received multicast frames according to the hash table. When reset, the GMAC performs a perfect destination address filtering for multicast frames, that is, it compares the DA field with the values programmed in DA registers.	R/W
3	DAIF	DA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the DA address comparison for both unicast and multicast frames. When reset, normal filtering of frames is performed.	R/W
4	PM	Pass All Multicast When set, this bit indicates that all received frames with a multicast destination address (first bit in the destination address field is 1) are passed. When reset, filtering of multicast frame depends on HMC bit.	R/W
5	DBF	Disable Broadcast Frames When this bit is set, the AFM module filters all incoming broadcast frames. In addition, it overrides all other filter settings. When this bit is reset, the AFM module passes all received broadcast frames.	R/W

Bit	Symbol	Function	R/W
7:6	PCF[1:0]	Pass Control Frames These bits control the forwarding of all control frames (including unicast and multicast PAUSE frames). 0 0: GMAC filters all control frames from reaching the application. 0 1: GMAC forwards all control frames except PAUSE control frames to application even if they fail the Address filter. 1 0: GMAC forwards all control frames to application even if they fail the Address Filter. 1 1: GMAC forwards control frames that pass the Address Filter. The following conditions should be true for the PAUSE control frames processing: ● Condition 1: The GMAC is in the full-duplex mode and flow control is enabled by setting bit RFE[2] of Flow Control Register (Flow_Control) to 1. ● Condition 2: The destination address (DA) of the received frame matches the special multicast address or the MAC Address 0 when bit [3] (UP) of the Flow Control Register (Flow_Control) is set. ● Condition 3: The Type field of the received frame is 0x8808 and the OPCODE field is 0x0001. This field should be set to 01b only when the Condition 1 is true, that is, the GMAC is programmed to operate in the full-duplex mode and the RFE bit is enabled. Otherwise, the PAUSE frame filtering may be inconsistent. When Condition 1 is false, the PAUSE frames are considered as generic control frames. Therefore, to pass all control frames (including PAUSE control frames) when the full-duplex mode and flow control is not enabled, set the PCF field to 10b or 11b (as required by the application).	R/W
8	SAIF	SA Inverse Filtering When this bit is set, the Address Check block operates in inverse filtering mode for the SA address comparison. The frames whose SA matches the SA registers are marked as failing the SA Address filter. When this bit is reset, frames whose SA does not match the SA registers are marked as failing the SA Address filter.	R/W
9	SAF	Source Address Filter Enable When this bit is set, the GMAC compares the SA field of the received frames with the values programmed in the enabled SA registers. If the comparison fails, the GMAC drops the frame. When this bit is reset, the GMAC forwards the received frame to the application with updated SAF bit of the RX status depending on the SA address comparison. According to the IEEE specification, bit [47] of the SA is reserved and set to 0. However, the GMAC compares all 48 bits. The software driver should take this into consideration while programming the MAC address registers for SA.	R/W
10	HPF	Hash or Perfect Filter When this bit is set, it configures the address filter to pass a frame if it matches either the perfect filtering or the hash filtering as set by the HMC or HUC bits. When this bit is low and the HUC or HMC bit is set, the frame is passed only if it matches the Hash filter.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
16	VTFE	VLAN Tag Filter Enable When set, this bit enables the GMAC to drop VLAN tagged frames that do not match the VLAN Tag comparison. When reset, the GMAC forwards all frames irrespective of the match status of the VLAN Tag.	R/W
30:17	—	These bits are read as 0. The write value should be 0.	R/W
31	RA	Receive All When this bit is set, the GMAC Receiver module passes all received frames, irrespective of whether they pass the address filter or not, to the application. The result of the SA or DA filtering is updated (pass or fail) in the corresponding bits in the Receive Status Word. When this bit is reset, the Receiver module passes only those frames to the application that pass the SA or DA address filter.	R/W

27.3.3 GMII_Address : GMII Address Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PA[4:0]					GR[4:0]					CR[3:0]				GW	GB
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	GB	GMII Busy This bit reads logic 0 before writing to GMII Address Register (GMII_Address) and GMII Data Register (GMII_Data). During a PHY register access, the software sets this bit to 1 to indicate that a read or write access is in progress. GMII Data Register (GMII_Data) is invalid until this bit is cleared by the GMAC. Therefore, GMII Data Register (GMII_Data) should be kept valid until the GMAC clears this bit during a PHY Write operation. Similarly, for a read operation, the contents of GMII Data Register (GMII_Data) are not valid until this bit is cleared. The subsequent read or write operation occurs only after the previous operation is complete. Because there is no acknowledgment from the PHY to GMAC after a read or write operation is completed, there is no change in the functionality of this bit even when the PHY is not present.	R/W
1	GW	GMII Write When set, this bit indicates to the PHY that this is a write operation using the GMII Data register. If this bit is not set, it indicates that this is a read operation, placing the data in the GMII Data register.	R/W

Bit	Symbol	Function	R/W
5:2	CR[3:0]	CSR Clock Range The CSR Clock (PCLKM) Range selection determines the frequency of the MDC clock according to the CSR clock frequency. The suggested range of CSR clock frequency applicable for each value (when bit [5] = 0) ensures that the MDC clock is approximately between the frequency range 1.0 MHz to 2.5 MHz. 0x0: The frequency of the CSR clock is 60-100 MHz, and the MDC clock is CSR clock/42. 0x1: The frequency of the CSR clock is 100-150 MHz, and the MDC clock is CSR clock/62. 0x2: The frequency of the CSR clock is 20-35 MHz, and the MDC clock is CSR clock/16. 0x3: The frequency of the CSR clock is 35-60 MHz, and the MDC clock is CSR clock/26. 0x4: The frequency of the CSR clock is 150-250 MHz, and the MDC clock is CSR clock/102. 0x5: The frequency of the CSR clock is 250-300 MHz, and the MDC clock is CSR clock/124. 0x6: Reserved 0x7: Reserved When bit [5] is set, you can achieve higher frequency of the MDC clock than the frequency limit of 2.5 MHz (specified in the IEEE Std 802.3) and program a clock divider of lower value. For example, when CSR clock is of 100 MHz frequency and you program these bits as 0xA, then the resultant MDC clock is of 12.5 MHz which is outside the limit of IEEE 802.3 specified range. Program the following values only if the interfacing chips support faster MDC clocks: 0x8: CSR clock/4 0x9: CSR clock/6 0xA: CSR clock/8 0xB: CSR clock/10 0xC: CSR clock/12 0xD: CSR clock/14 0xE: CSR clock/16 0xF: CSR clock/18	R/W
10:6	GR[4:0]	GMII Register These bits select the desired GMII register in the selected PHY device.	R/W
15:11	PA[4:0]	Physical Layer Address This field indicates which of the 32 possible PHY devices are being accessed.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

27.3.4 GMII_Data : GMII Data Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GD[15:0]															
Value after reset:	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
15:0	GD[15:0]	GMII Data This field contains the 16-bit data value read from the PHY after a Management Read operation or the 16-bit data value to be written to the PHY before a Management Write operation.	R/W

Bit	Symbol	Function	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

27.3.5 Flow_Control : Flow Control Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	DZPQ	—	PLT[1:0]		UP	RFE	TFE	FCA_ BPA
Value after reset:	x	x	x	x	x	x	x	x	0	x	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FCA_BPA	Flow Control Busy or Backpressure Activate This bit initiates a Pause Control frame in the full-duplex mode and activates the backpressure function in the half-duplex mode if the TFE bit is set. In the full-duplex mode, this bit should be read as 0b before writing to the Flow Control register. To initiate a Pause control frame, the application must set this bit to 1b. During a transfer of the Control Frame, this bit continues to be set to signify that a frame transmission is in progress. After the completion of Pause control frame transmission, the GMAC resets this bit to 0b. The Flow Control register should not be written to until this bit is cleared. In the half-duplex mode, when this bit is set (and TFE is set), then backpressure is asserted by the GMAC. During backpressure, when the GMAC receives a new frame, the transmitter starts sending a JAM pattern resulting in a collision. This control register bit is logically ORed with the flow control input signal for the backpressure function. When the GMAC is configured for the full-duplex mode, backpressure is automatically disabled.	R/W
1	TFE	Transmit Flow Control Enable In the full-duplex mode, when this bit is set, the GMAC enables the flow control operation to transmit Pause frames. When this bit is reset, the flow control operation in the GMAC is disabled, and the GMAC does not transmit any Pause frames. In half-duplex mode, when this bit is set, the GMAC enables the backpressure operation. When this bit is reset, the backpressure feature is disabled.	R/W
2	RFE	Receive Flow Control Enable When this bit is set, the GMAC decodes the received Pause frame and disables its transmitter for a specified (Pause) time. When this bit is reset, the decode function of the Pause frame is disabled.	R/W
3	UP	Unicast Pause Frame Detect A Pause frame is processed when it has the unique multicast address specified in the IEEE Std 802.3. When this bit is set, the GMAC can also detect Pause frames with unicast address of the station. This unicast address should be as specified in the MAC Address0 High Register and MAC Address0 Low Register. When this bit is reset, the GMAC only detects Pause frames with unique multicast address. The GMAC does not process a Pause frame if the multicast address of received frame is different from the unique multicast address.	R/W

Bit	Symbol	Function	R/W
5:4	PLT[1:0]	Pause Low Threshold This field configures the threshold of the Pause timer at which the input flow control signal is checked for automatic retransmission of Pause frame. The threshold values should always be less than the Pause Time configured in bits [31:16]. For example, if PT = 0x100 (256 slot-times), and PLT = 01b, then a second Pause frame is automatically transmitted if the flow control signal is asserted at 228 (256 - 28) slot times after the first Pause frame is transmitted. The following list provides the threshold values for different values: 0 0: The threshold is Pause time minus 4 slot times (PT - 4 slot times). 0 1: The threshold is Pause time minus 28 slot times (PT - 28 slot times). 1 0: The threshold is Pause time minus 144 slot times (PT - 144 slot times). 1 1: The threshold is Pause time minus 256 slot times (PT - 256 slot times). The slot time is defined as the time taken to transmit 512 bits (64 bytes) on the GMII or MII interface.	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	DZPQ	Disable Zero-Quanta Pause When this bit is set, it disables the automatic generation of the Zero-Quanta Pause Control frames on the deassertion of the flow control signal from the FIFO layer. When this bit is reset, normal operation with automatic Zero-Quanta Pause Control frame generation is enabled.	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
31:16	PT[15:0]	Pause Time This field holds the value to be used in the Pause Time field in the transmit control frame. Consecutive writes to this register should be performed only after at least 4 clock cycles in the destination clock domain.	R/W

27.3.6 VLAN_Tag : VLAN Tag Register

Base address: GMAC = 0x8010_0000

Offset address: 0x001C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	VTHM	ESVL	VTIM	ETV
Value after reset:	0	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	VL[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	VL[15:0]	VLAN Tag Identifier for Receive Frames This field contains the 802.1Q VLAN tag to identify the VLAN frames and is compared to the 15th and 16th bytes of the frames being received for VLAN frames. The following list describes the bits of this field: Bits [15:13]: User Priority Bit [12]: Canonical Format Indicator (CFI) or Drop Eligibility Indicator (DEI) Bits [11:0]: VLAN Identifier (VID) field of VLAN tag When the ETV bit is set, only bits VID[11:0] are used for comparison. If VL (VL[11:0] if ETV is set) is all 0s, the GMAC does not check the 15th and 16th bytes for VLAN tag comparison, and declares all frames with a Type field value of 0x8100 or 0x88A8 as VLAN frames.	R/W
16	ETV	Enable 12-Bit VLAN Tag Comparison When this bit is set, a 12-bit VLAN identifier is used for comparing and filtering instead of the complete 16-bit VLAN tag. Bits [11:0] of VLAN tag are compared with the corresponding field in the received VLAN-tagged frame. Similarly, when enabled, only 12 bits of the VLAN tag in the received frame are used for hash-based VLAN filtering. When this bit is reset, all 16 bits of the 15th and 16th bytes of the received VLAN frame are used for comparison and VLAN hash filtering.	R/W

Bit	Symbol	Function	R/W
17	VTIM	VLAN Tag Inverse Match Enable When set, this bit enables the VLAN Tag inverse matching. The frames that do not have matching VLAN Tag are marked as matched. When reset, this bit enables the VLAN Tag perfect matching. The frames with matched VLAN Tag are marked as matched.	R/W
18	ESVL	Enable S-VLAN When this bit is set, the GMAC transmitter and receiver also consider the S-VLAN (Type = 0x88A8) frames as valid VLAN tagged frames.	R/W
19	VTHM	VLAN Tag Hash Table Match Enable When set, the most significant 4 bits of the CRC of VLAN tag are used to index the content of VLAN Hash Table Register (VLAN_Hash_Table_Reg). A value of 1 in the VLAN Hash Table register, corresponding to the index, indicates that the frame matched the VLAN hash table. When bit ETV[16] is set, the CRC of the 12-bit VLAN Identifier (VID) is used for comparison whereas when ETV is reset, the CRC of the 16-bit VLAN tag is used for comparison. When reset, the VLAN Hash Match operation is not performed.	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

27.3.7 Version : Version Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VER[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	1	1	0	0	0	1	0	0	1	1	0	1	1	1

Bit	Symbol	Function	R/W
15:0	VER[15:0]	Version (GMAC: 0x3137)	R
31:16	—	These bits are read as 0.	R

27.3.8 Debug : Debug Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0024

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	TXSTSFSTS	TXFSTS	—	TWCSTS	TRCSTS[1:0]		TXPAUSED	TFCSTS[1:0]		TPESTS
Value after reset:	x	x	x	x	x	x	0	0	x	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	RXFSTS[1:0]		—	RRCSTS[1:0]		RWCSTS	—	RFCFCSTS[1:0]		RPESTS
Value after reset:	x	x	x	x	x	x	0	0	x	0	0	0	x	0	0	0

Bit	Symbol	Function	R/W
0	RPESTS	GMAC GMII or MII Receive Protocol Engine Status When high, this bit indicates that the GMAC GMII or MII receive protocol engine is actively receiving data and not in IDLE state.	R
2:1	RFCFCSTS[1:0]	GMAC Receive Frame Controller FIFO Status When high, this field indicates the active state of the small FIFO read and write controllers of the GMAC Receive Frame Controller Module.	R
3	—	This bit is read as 0.	R

Bit	Symbol	Function	R/W
4	RWCSTS	MTL RX FIFO Write Controller Active Status When high, this bit indicates that the MTL RX FIFO write controller is active and is transferring a received frame to the FIFO.	R
6:5	RRCSTS[1:0]	MTL RX FIFO Read Controller State This field gives the state of the RX FIFO read controller: 0 0: IDLE state 0 1: Reading frame data 1 0: Reading frame status (or timestamp) 1 1: Flushing the frame data and status	R
7	—	This bit is read as 0.	R
9:8	RXFSTS[1:0]	MTL RX FIFO Fill-level Status This field gives the status of the fill level of the RX FIFO: 0 0: RX FIFO Empty 0 1: RX FIFO fill level is below the flow-control deactivate threshold 1 0: RX FIFO fill level is above the flow-control activate threshold 1 1: RX FIFO Full	R
15:10	—	These bits are read as 0.	R
16	TPESTS	GMAC GMII or MII Transmit Protocol Engine Status When high, this bit indicates that the GMAC GMII or MII transmit protocol engine is actively transmitting data and is not in the IDLE state.	R
18:17	TFCSTS[1:0]	GMAC Transmit Frame Controller Status This field indicates the state of the GMAC Transmit Frame Controller module: 0 0: IDLE state 0 1: Waiting for status of previous frame, or IFG, or backoff period to be over 1 0: Generating and transmitting a Pause control frame (in the full-duplex mode) 1 1: Transferring input frame for transmission	R
19	TXPAUSED	GMAC transmitter in PAUSE When high, this bit indicates that the GMAC transmitter is in the PAUSE condition (in the full-duplex only mode) and hence does not schedule any frame for transmission.	R
21:20	TRCSTS[1:0]	MTL TX FIFO Read Controller Status This field indicates the state of the TX FIFO read controller: 0 0: IDLE state 0 1: READ state (transferring data to GMAC transmitter) 1 0: Waiting for TX status from GMAC transmitter 1 1: Writing the received TX status or flushing the TX FIFO	R
22	TWCSTS	MTL TX FIFO Write Controller Active Status When high, this bit indicates that the MTL TX FIFO write controller is active and transferring data to the TX FIFO.	R
23	—	This bit is read as 0.	R
24	TXFSTS	MTL TX FIFO Not Empty Status When high, this bit indicates that the MTL TX FIFO is not empty and some data is left for transmission.	R
25	TXSTSFSTS	MTL TX Status FIFO Full Status When high, this bit indicates that the MTL (MAC Transaction Layer) TX status FIFO is full. Therefore, the MTL cannot accept any more frames for transmission.	R
31:26	—	These bits are read as 0.	R

27.3.9 Remote_Wake_Up_Frame_Filter : Remote Wake-Up Frame Filter Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0028

Bit position:	31																												0
Bit field:	WKUPFRMFTR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	WKUPFRMFTR[31:0]	Remote Wake-Up Frame Filter	R/W

27.3.10 PMT_Control_Status : PMT Control and Status Register

Base address: GMAC = 0x8010_0000

Offset address: 0x002C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RWKFILTRST	—	—	—	—	RWKPTR[2:0]			—	—	—	—	—	—	—	—
Value after reset:	0	x	x	x	x	0	0	0	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	GLBLUCAST	—	—	RWKPRCVD	MGKPRCVD	—	—	RWKPKTEN	MGKPKTEN	PWRDWN
Value after reset:	x	x	x	x	x	0	0	x	x	0	0	x	x	0	0	0

Bit	Symbol	Function	R/W
0	PWRDWN	Power Down When set, the GMAC receiver drops all received frames until it receives the expected magic packet or wakeup frame. This bit is then self-cleared and the power-down mode is disabled. The Software can also clear this bit before the expected magic packet or wakeup frame is received. The frames, received by the GMAC after this bit is cleared, are forwarded to the application. This bit must only be set when the Magic Packet Enable, Global Unicast, or Wake-Up Frame Enable bit is set high. You can gate-off the CSR clock (PCLKM) during the power-down mode. However, when the CSR clock is gated-off, you cannot perform any read or write operations on this register. Therefore, software cannot clear this bit.	R/W
1	MGKPKTEN	Magic Packet Enable When set, enables generation of a power management event because of magic packet reception.	R/W
2	RWKPKTEN	Wake-Up Frame Enable When set, enables generation of a power management event because of wake-up frame reception.	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	MGKPRCVD	Magic Packet Received When set, this bit indicates that the power management event is generated because of the reception of a magic packet. This bit is cleared by a read into this register.	R
6	RWKPRCVD	Wake-Up Frame Received When set, this bit indicates the power management event is generated because of the reception of a wakeup frame. This bit is cleared by a read into this register.	R
8:7	—	These bits are read as 0. The write value should be 0.	R/W
9	GLBLUCAST	Global Unicast When set, enables any unicast packet filtered by the MAC address recognition (DA filter) to be a wakeup frame.	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
23:11	—	These bits are read as 0. The write value should be 0.	R/W
26:24	RWKPTR[2:0]	Remote Wake-Up FIFO Pointer This gives the current value (0 to 7) of the Remote Wake-up Frame Filter Register pointer. The contents of the Remote Wake-up Frame Filter Register are transferred to the receive clock domain when a write occurs to that register while this pointer value equals 7.	R
30:27	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
31	RWKFILTRST	Wake-Up Frame Filter Register Pointer Reset When this bit is set, it resets the remote Wake-Up Frame Filter Register pointer to 000b. It is automatically cleared after 1 clock cycle.	R/W

27.3.11 LPI_Control_Status : LPI Control and Status Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0030

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	LPITXA	—	PLS	LPIEN
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	0	x	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	RLPIST	TLPIST	—	—	—	—	RLPIEX	RLPIEN	TLPIEX	TLPIEN
Value after reset:	x	x	x	x	x	x	0	0	x	x	x	x	0	0	0	0

Bit	Symbol	Function	R/W
0	TLPIEN	Transmit LPI Entry When set, this bit indicates that the GMAC transmitter has entered the LPI state because of the setting of the LPIEN bit. This bit is cleared by a read into this register.	R
1	TLPIEX	Transmit LPI Exit When set, this bit indicates that the GMAC transmitter has exited the LPI state after the LPIEN bit is cleared and the LPI TW Timer has expired. This bit is cleared by a read into this register.	R
2	RLPIEN	Receive LPI Entry When set, this bit indicates that the GMAC receiver has received an LPI pattern and entered the LPI state. This bit is cleared by a read into this register. This bit may not get set if the GMAC stops receiving the LPI pattern for a very short duration, such as, less than 3 clock cycles of CSR clock.	R
3	RLPIEX	Receive LPI Exit When set, this bit indicates that the GMAC receiver has stopped receiving the LPI pattern on the GMII or MII interface, exited the LPI state, and resumed the normal reception. This bit is cleared by a read into this register. This bit may not get set if the GMAC stops receiving the LPI pattern for a very short duration, such as, less than 3 clock cycles of CSR clock.	R
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	TLPIST	Transmit LPI State When set, this bit indicates that the GMAC is transmitting the LPI pattern on the GMII or MII interface.	R
9	RLPIST	Receive LPI State When set, this bit indicates that the GMAC is receiving the LPI pattern on the GMII or MII interface.	R
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	LPIEN	LPI Enable When set, this bit instructs the GMAC transmitter to enter the LPI state. When reset, this bit instructs the GMAC to exit the LPI state and resume normal transmission. This bit is cleared when the LPITXA bit is set and the GMAC exits the LPI state because of the arrival of a new packet for transmission.	R/W
17	PLS	PHY Link Status This bit indicates the link status of the PHY. The GMAC transmitter asserts the LPI pattern only when the link status is up (okay) at least for the time indicated by the LPI LS timer. When set, the link is considered to be up (okay) and when reset, the link is considered to be down.	R/W
18	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
19	LPITXA	LPI TX Automate This bit controls the behavior of the GMAC when it is entering or coming out of the LPI mode on the transmit side. If the LPITXA and LPIEN bits are set to 1, the GMAC enters the LPI mode only after all outstanding frames (in the core) and pending frames (in the application interface) have been transmitted. The GMAC comes out of the LPI mode when the application sends any frame for transmission or the application issues a TX FIFO Flush command. In addition, the GMAC automatically clears the LPIEN bit when it exits the LPI state. If TX FIFO Flush bit [20] of Operation Mode Register (Operation_Mode) is set when the GMAC is in the LPI mode, the GMAC exits the LPI mode. When this bit is 0, the LPIEN bit directly controls the behavior of GMAC when it is entering or coming out of the LPI mode.	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

27.3.12 LPI_Timers_Control : LPI Timers Control Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0034

Bit position:	31						25									16	15														0
Bit field:	—	—	—	—	—	—	LST[9:0]										TWT[15:0]														
Value after reset:	x	x	x	x	x	x	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TWT[15:0]	LPI TW Timer This field specifies the minimum time (in microseconds) for which the GMAC waits after it stops transmitting the LPI pattern to the PHY and before it resumes the normal transmission. The TLPIEX status bit is set after the expiry of this timer.	R/W
25:16	LST[9:0]	LPI LS Timer This field specifies the minimum time (in milliseconds) for which the link status from the PHY should be up (okay) before the LPI pattern can be transmitted to the PHY. The GMAC does not transmit the LPI pattern even when the LPIEN bit is set unless the LPI LS Timer reaches the programmed terminal count. The default value of the LPI LS Timer is 1000 (1 sec.) as defined in the IEEE standard.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

27.3.13 Interrupt_Status : Interrupt Status Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0038

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	LPIIS	TSIS	—	MMCRXIPIS	MMCTXIS	MMCRXIS	MMCIS	PMTIS	—	—	—
Value after reset:	x	x	x	x	x	0	0	x	0	0	0	0	0	x	x	x

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
3	PMTIS	PMT Interrupt Status This bit is set when a Magic packet or Wake-on-LAN frame is received in the power-down mode (see bits [5] and [6] in the PMT Control and Status Register). This bit is cleared when both bits [5] and [6] are cleared because of a read operation to the PMT Control and Status register.	R
4	MMCIS	MMC Interrupt Status This bit is set high when any of the bits [7:5] is set high and cleared only when all of these bits are low.	R
5	MMCRXIS	MMC Receive Interrupt Status This bit is set high when an interrupt is generated in the MMC Receive Interrupt Register. This bit is cleared when all the bits in this register are cleared.	R
6	MMCTXIS	MMC Transmit Interrupt Status This bit is set high when an interrupt is generated in the MMC Transmit Interrupt Register. This bit is cleared when all the bits in this register are cleared.	R
7	MMCRXIPIS	MMC Receive Checksum Offload Interrupt Status This bit is set high when an interrupt is generated in the MMC Receive Checksum Offload Interrupt Register. This bit is cleared when all the bits in this register are cleared.	R
8	—	This bit is read as 0.	R
9	TSIS	Timestamp Interrupt Status This bit is set when any of the following conditions is true: ● The system time value equals or exceeds the value specified in the Target Time High and Low registers. ● There is an overflow in the seconds register. ● The Auxiliary snapshot trigger is asserted. This bit is cleared on reading bit [0] of the Timestamp Status Register (Timestamp_Status). If default Timestamping is enabled, when set, this bit indicates that the system time value is equal to or exceeds the value specified in the Target Time registers. In this mode, this bit is cleared after a read completion of this bit.	R
10	LPIIS	LPI Interrupt Status This bit is set for any LPI state entry or exit in the GMAC transmitter or receiver. This bit is cleared on reading bit [0] of LPI Control and Status Register (LPI_Control_Status).	R
31:11	—	These bits are read as 0.	R

27.3.14 Interrupt_Mask : Interrupt Mask Register

Base address: GMAC = 0x8010_0000

Offset address: 0x003C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	LPIIM	TSIM	—	—	—	—	—	PMTIM	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0. The write value should be 0.	R/W
3	PMTIM	PMT Interrupt Mask When set, this bit disables the assertion of the interrupt signal because of the setting of PMT Interrupt Status bit in Interrupt Status Register (Interrupt_Status).	R/W
8:4	—	These bits are read as 0. The write value should be 0.	R/W
9	TSIM	Timestamp Interrupt Mask When set, this bit disables the assertion of the interrupt signal because of the setting of Timestamp Interrupt Status bit in Interrupt Status Register (Interrupt_Status).	R/W

Bit	Symbol	Function	R/W
10	LPIIM	LPI Interrupt Mask When set, this bit disables the assertion of the interrupt signal because of the setting of the LPI Interrupt Status bit in Interrupt Status Register (Interrupt_Status).	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

27.3.15 MAR0_H : MAC Address 0 High Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0040

Bit position:	31															15														0
Bit field:	AE	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADDRHI[15:0]														
Value after reset:	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	ADDRHI[15:0]	MAC Address0[47:32] This field contains the upper 16 bits [47:32] of the first 6-byte MAC address. The GMAC uses this field for filtering the received frames and inserting the MAC address in the Transmit Flow Control (Pause) Frames.	R/W
30:16	—	These bits are read as 0. The write value should be 0.	R/W
31	AE	Address Enable This bit is always set to 1.	R

27.3.16 MAR0_L : MAC Address 0 Low Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0044 + 0x8 × n

Bit position:	31																													0
Bit field:	ADDRLO[31:0]																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	ADDRLO[31:0]	MAC Address0[31:0] This field contains the lower 32 bits [31:0] of the first 6-byte MAC address. This is used by the GMAC for filtering the received frames and inserting the MAC address in the Transmit Flow Control (Pause) Frames.	R/W

27.3.17 MARn_H : MAC ADDRESS High Register (n = 1 to 17)

Base address: GMAC = 0x8010_0000

Offset address: 0x0040 + 0x8 × n (n = 1 to 15)
0x0800 (n = 16), 0x0808 (n = 17)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	AE	SA	MBC[5:0]						—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADDRHI[15:0]															
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
15:0	ADDRHI[15:0]	MAC Address n[47:32] This field contains the upper 16 bits [47:32] of the nth 6-byte MAC address.	R/W
23:16	—	These bits are read as 0. The write value should be 0.	R/W
29:24	MBC[5:0]	Mask Byte Control These bits are mask control bits for comparison of each of the MAC Address bytes. When set high, the GMAC does not compare the corresponding byte of received DA or SA with the contents of MAC Address1 registers. Each bit controls the masking of the bytes as follows: MBC[5]: MARn_H.ADDRHI[15:8] MBC[4]: MARn_H.ADDRHI[7:0] MBC[3]: MADRn_L.ADDRLO[31:24] MBC[2]: MADRn_L.ADDRLO[23:16] MBC[1]: MADRn_L.ADDRLO[15:8] MBC[0]: MADRn_L.ADDRLO[7:0] You can filter a group of addresses (known as group address filtering) by masking one or more bytes of the address.	R/W
30	SA	Source Address When this bit is set, the MAC Address n[47:0] is used to compare with the SA fields of the received frame. When this bit is reset, the MAC Address n[47:0] is used to compare with the DA fields of the received frame.	R/W
31	AE	Address Enable When this bit is set, the address filter module uses the nth MAC address for perfect filtering. When this bit is reset, the address filter module ignores the address for filtering.	R/W

27.3.18 MARn_L : MAC ADDRESS Low Register (n = 1 to 17)

Base address: GMAC = 0x8010_0000

Offset address: 0x0044 + 0x8 × n (n = 1 to 15)
0x0804 (n = 16), 0x080C (n = 17)

Bit position:	31																													0
Bit field:	ADDRLO[31:0]																													
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
31:0	ADDRLO[31:0]	MAC Address n[31:0] This field contains the lower 32 bits [31:0] of the nth 6-byte MAC address. The content of this field is undefined until loaded by the application after the initialization process.	R/W

27.3.19 WDog_Timeout : Watchdog Timeout Register

Base address: GMAC = 0x8010_0000

Offset address: 0x00DC

Bit position:	31															16		13												0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PWE	—	—	WTO[13:0]												
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	WTO[13:0]	Watchdog Timeout When bit PWE[16] is set and bit WD[23] of MAC Configuration Register (MAC_Configuration) is reset, this field is used as watchdog timeout for a received frame. If the length of a received frame exceeds the value of this field, such frame is terminated and declared as an error frame. When bit PWE[16] is set, the value in this field should be more than 1,522 (0x05F2). Otherwise, the IEEE Std 802.3-specified valid tagged frames are declared as error frames and are dropped.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	PWE	Programmable Watchdog Enable When this bit is set and bit WD[23] of MAC Configuration Register (MAC_Configuration) is reset, the WTO field (bits [13:0]) is used as watchdog timeout for a received frame. When this bit is cleared, the watchdog timeout for a received frame is controlled by the setting of bit WD[23] and bit JE[20] in MAC Configuration Register (MAC_Configuration).	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

27.3.20 MMC_Control : MMC Control Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0100

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	UCDBC	—	—	CNTPRSTLVL	CNTPRST	CNTFREEZ	RSTONRD	CNTSTOPRO	CNTRST
Value after reset:	x	x	x	x	x	x	x	0	x	x	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CNTRST	Counters Reset When this bit is set, all counters are reset. This bit is cleared automatically after 1 clock cycle.	R/W
1	CNTSTOPRO	Counters Stop Rollover When this bit is set, after reaching maximum value, the counter does not roll over to 0.	R/W
2	RSTONRD	Reset on Read When this bit is set, the MMC counters are reset to 0 after a read (self-clearing after reset). The counters are cleared when the least significant byte lane (bits [7:0]) is read.	R/W
3	CNTFREEZ	MMC Counter Freeze When this bit is set, it freezes all MMC counters to their current value. Until this bit is reset to 0, no MMC counter is updated because of any transmitted or received frame. If any MMC counter is read with the Reset on Read bit set, then that counter is also cleared in this mode.	R/W
4	CNTPRST	Counters Preset When this bit is set, all counters are initialized or preset to almost full or almost half according to bit [5]. This bit is cleared automatically after 1 clock cycle. This bit, along with bit [5], is useful for debugging and testing the assertion of interrupts because of MMC counter becoming half-full or full.	R/W

Bit	Symbol	Function	R/W
5	CNTPRSTLVL	Full-Half Preset When low and bit [4] is set, all MMC counters get preset to almost-half value. All octet counters get preset to 0x7FFF_F800 (half - 2 KB) and all frame-counters gets preset to 0x7FFF_FFF0 (half - 16). When this bit is high and bit [4] is set, all MMC counters get preset to almost-full value. All octet counters get preset to 0xFFFF_F800 (full - 2 KB) and all frame-counters gets preset to 0xFFFF_FFF0 (full - 16). For 16-bit counters, the almost-half preset values are 0x7800 and 0x7FF0 for the respective octet and frame counters. Similarly, the almost-full preset values for the 16-bit counters are 0xF800 and 0xFFF0.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	UCDBC	Update MMC Counters for Dropped Broadcast Frames When set, this bit enables GMAC to update all the related MMC counters for Broadcast frames dropped due to setting of DBF bit (Disable Broadcast Frames) of MAC Frame Filter Register at offset 0x0004. When reset, MMC counters are not updated for dropped Broadcast frames.	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

27.3.21 MMC_Receive_Interrupt : MMC Receive Interrupt Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0104

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	RXCTRLFIS	RXRCVERRFIS	RXWDOGFIS	RXVLANGBFIS	RXFOVFIS	RXPAUSFIS	RXORANGEFIS	RXLENERFIS	RXUCGFIS	RX1024TMAXOCTGBFIS
Value after reset:	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RX512T1023OCTGBFIS	RX256T511OCTGBFIS	RX128T255OCTGBFIS	RX65T127OCTGBFIS	RX64OCTGBFIS	RXOSIZEGFIS	RXUSIZEGFIS	RXJABERFIS	RXRUNTFIS	RXALGNERFIS	RXCRCERFIS	RXMCGFIS	RXBCGFIS	RXGOCTIS	RXGBOCTIS	RXGBFRMIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RXGBFRMIS	MMC Receive Good Bad Frame Counter Interrupt Status This bit is set when the Rx_Frames_Count_Good_Bad counter reaches half of the maximum value or the maximum value.	R
1	RXGBOCTIS	MMC Receive Good Bad Octet Counter Interrupt Status This bit is set when the Rx_Octet_Count_Good_Bad counter reaches half of the maximum value or the maximum value.	R
2	RXGOCTIS	MMC Receive Good Octet Counter Interrupt Status This bit is set when the Rx_Octet_Count_Good counter reaches half of the maximum value or the maximum value.	R
3	RXBCGFIS	MMC Receive Broadcast Good Frame Counter Interrupt Status This bit is set when the Rx_Broadcast_Frames_Good counter reaches half of the maximum value or the maximum value.	R
4	RXMCGFIS	MMC Receive Multicast Good Frame Counter Interrupt Status This bit is set when the Rx_Multicast_Frames_Good counter reaches half of the maximum value or the maximum value.	R
5	RXCRCERFIS	MMC Receive CRC Error Frame Counter Interrupt Status This bit is set when the Rx_CRC_Error_Frames counter reaches half of the maximum value or the maximum value.	R
6	RXALGNERFIS	MMC Receive Alignment Error Frame Counter Interrupt Status This bit is set when the Rx_Alignment_Error_Frames counter reaches half of the maximum value or the maximum value.	R

Bit	Symbol	Function	R/W
7	RXRUNTFIS	MMC Receive Runt Frame Counter Interrupt Status This bit is set when the Rx_Runt_Error_Frames counter reaches half of the maximum value or the maximum value.	R
8	RXJABERFIS	MMC Receive Jabber Error Frame Counter Interrupt Status This bit is set when the Rx_Jabber_Error_Frames counter reaches half of the maximum value or the maximum value.	R
9	RXUSIZEGFIS	MMC Receive Undersize Good Frame Counter Interrupt Status This bit is set when the Rx_Undersize_Frames_Good counter reaches half of the maximum value or the maximum value.	R
10	RXOSIZEGFIS	MMC Receive Oversize Good Frame Counter Interrupt Status This bit is set when the Rx_Oversize_Frames_Good counter reaches half of the maximum value or the maximum value.	R
11	RX64OCTGBFIS	MMC Receive 64 Octet Good Bad Frame Counter Interrupt Status This bit is set when the Rx_64Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
12	RX65T127OCTGBFIS	MMC Receive 65 to 127 Octet Good Bad Frame Counter Interrupt Status This is set when the Rx_65To127Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
13	RX128T255OCTGBFIS	MMC Receive 128 to 255 Octet Good Bad Frame Counter Interrupt Status This bit is set when the Rx_128To255Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
14	RX256T511OCTGBFIS	MMC Receive 256 to 511 Octet Good Bad Frame Counter Interrupt Status This bit is set when the Rx_256To511Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
15	RX512T1023OCTGBFIS	MMC Receive 512 to 1023 Octet Good Bad Frame Counter Interrupt Status This bit is set when the Rx_512To1023Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
16	RX1024TMAXOCTGBFIS	MMC Receive 1024 to Maximum Octet Good Bad Frame Counter Interrupt Status This bit is set when the Rx_1024ToMaxOctets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
17	RXUCGFIS	MMC Receive Unicast Good Frame Counter Interrupt Status This bit is set when the Rx_Unicast_Frames_Good counter reaches half of the maximum value or the maximum value.	R
18	RXLENERFIS	MMC Receive Length Error Frame Counter Interrupt Status This bit is set when the Rx_Length_Error_Frames counter reaches half of the maximum value or the maximum value.	R
19	RXORANGEFIS	MMC Receive Out Of Range Error Frame Counter Interrupt Status This bit is set when the Rx_Out_Of_Range_Type_Frames counter reaches half of the maximum value or the maximum value.	R
20	RXPAUSFIS	MMC Receive Pause Frame Counter Interrupt Status This bit is set when the Rx_Pause_Frames counter reaches half of the maximum value or the maximum value.	R
21	RXFOVFIS	MMC Receive FIFO Overflow Frame Counter Interrupt Status This bit is set when the Rx_FIFO_Overflow_Frames counter reaches half of the maximum value or the maximum value.	R
22	RXVLANGBFIS	MMC Receive VLAN Good Bad Frame Counter Interrupt Status This bit is set when the Rx_VLAN_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
23	RXWDOGFIS	MMC Receive Watchdog Error Frame Counter Interrupt Status This bit is set when the Rx_Watchdog_Error_Frames counter reaches half of the maximum value or the maximum value.	R
24	RXRCVERRFIS	MMC Receive Error Frame Counter Interrupt Status This bit is set when the Rx_Receive_Error_Frames counter reaches half of the maximum value or the maximum value.	R
25	RXCTRLFIS	MMC Receive Control Frame Counter Interrupt Status This bit is set when the Rx_Control_Frames_Good counter reaches half of the maximum value or the maximum value.	R

Bit	Symbol	Function	R/W
31:26	—	These bits are read as 0.	R

27.3.22 MMC_Transmit_Interrupt : MMC Transmit Interrupt Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0108

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	TXOSIZEGFIS	TXVLANGFIS	TXPAUSFIS	TXEXDEFFIS	TXGFRMIS	TXGOCTIS	TXCARERFIS	TXEXCOLFIS	TXLATCOLFIS	TXDEFFIS
Value after reset:	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TXMCOLGFIS	TXSCOLGFIS	TXUFLOWERFIS	TXBCGBFIS	TXMCGBFIS	TXUCGBFIS	TX1024TMAXOCTGBFIS	TX512T1023OCTGBFIS	TX256T511OCTGBFIS	TX128T255OCTGBFIS	TX65T127OCTGBFIS	TX64OCTGBFIS	TXMCGFIS	TXBCGFIS	TXGBFRMIS	TXGBOCTIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXGBOCTIS	MMC Transmit Good Bad Octet Counter Interrupt Status This bit is set when the Tx_Octet_Count_Good_Bad counter reaches half of the maximum value or the maximum value.	R
1	TXGBFRMIS	MMC Transmit Good Bad Frame Counter Interrupt Status This bit is set when the Tx_Frame_Count_Good_Bad counter reaches half of the maximum value or the maximum value.	R
2	TXBCGFIS	MMC Transmit Broadcast Good Frame Counter Interrupt Status This bit is set when the Tx_Broadcast_Frames_Good counter reaches half of the maximum value or the maximum value.	R
3	TXMCGFIS	MMC Transmit Multicast Good Frame Counter Interrupt Status This bit is set when the Tx_Multicast_Frames_Good counter reaches half of the maximum value or the maximum value.	R
4	TX64OCTGBFIS	MMC Transmit 64 Octet Good Bad Frame Counter Interrupt Status This bit is set when the Tx_64Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
5	TX65T127OCTGBFIS	MMC Transmit 65 to 127 Octet Good Bad Frame Counter Interrupt Status This bit is set when the Tx_65To127Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
6	TX128T255OCTGBFIS	MMC Transmit 128 to 255 Octet Good Bad Frame Counter Interrupt Status This bit is set when the Tx_128To255Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
7	TX256T511OCTGBFIS	MMC Transmit 256 to 511 Octet Good Bad Frame Counter Interrupt Status This bit is set when the Tx_256To511Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
8	TX512T1023OCTGBFIS	MMC Transmit 512 to 1023 Octet Good Bad Frame Counter Interrupt Status This bit is set when the Tx_512To1023Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
9	TX1024TMAXOCTGBFIS	MMC Transmit 1024 to Maximum Octet Good Bad Frame Counter Interrupt Status This bit is set when the Tx_1024ToMaxOctets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
10	TXUCGBFIS	MMC Transmit Unicast Good Bad Frame Counter Interrupt Status This bit is set when the Tx_Unicast_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
11	TXMCGBFIS	MMC Transmit Multicast Good Bad Frame Counter Interrupt Status This bit is set when the Tx_Multicast_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R

Bit	Symbol	Function	R/W
12	TXBCGBFIS	MMC Transmit Broadcast Good Bad Frame Counter Interrupt Status This bit is set when the Tx_Broadcast_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R
13	TXUFLOWERFIS	MMC Transmit Underflow Error Frame Counter Interrupt Status This bit is set when the Tx_Underflow_Error_Frames counter reaches half of the maximum value or the maximum value.	R
14	TXSCOLGFIS	MMC Transmit Single Collision Good Frame Counter Interrupt Status This bit is set when the Tx_Single_Collision_Good_Frames counter reaches half of the maximum value or the maximum value.	R
15	TXMCOLGFIS	MMC Transmit Multiple Collision Good Frame Counter Interrupt Status This bit is set when the Tx_Multiple_Collision_Good_Frames counter reaches half of the maximum value or the maximum value.	R
16	TXDEFFIS	MMC Transmit Deferred Frame Counter Interrupt Status This bit is set when the Tx_Deferred_Frames counter reaches half of the maximum value or the maximum value.	R
17	TXLATCOLFIS	MMC Transmit Late Collision Frame Counter Interrupt Status This bit is set when the Tx_Late_Collision_Frames counter reaches half of the maximum value or the maximum value.	R
18	TXEXCOLFIS	MMC Transmit Excessive Collision Frame Counter Interrupt Status This bit is set when the Tx_Excessive_Collision_Frames counter reaches half of the maximum value or the maximum value.	R
19	TXCARERFIS	MMC Transmit Carrier Error Frame Counter Interrupt Status This bit is set when the Tx_Carrier_Error_Frames counter reaches half of the maximum value or the maximum value.	R
20	TXGOCTIS	MMC Transmit Good Octet Counter Interrupt Status This bit is set when the Tx_Octet_Count_Good counter reaches half of the maximum value or the maximum value.	R
21	TXGFRMIS	MMC Transmit Good Frame Counter Interrupt Status This bit is set when the Tx_Frame_Count_Good counter reaches half of the maximum value or the maximum value.	R
22	TXEXDEFFIS	MMC Transmit Excessive Deferral Frame Counter Interrupt Status This bit is set when the Tx_Excessive_Deferral_Error counter reaches half of the maximum value or the maximum value.	R
23	TXPAUSFIS	MMC Transmit Pause Frame Counter Interrupt Status This bit is set when the Tx_Pause_Frames counter reaches half of the maximum value or the maximum value.	R
24	TXVLANGFIS	MMC Transmit VLAN Good Frame Counter Interrupt Status This bit is set when the Tx_VLAN_Frames_Good counter reaches half of the maximum value or the maximum value.	R
25	TXOSIZEGFIS	MMC Transmit Oversize Good Frame Counter Interrupt Status This bit is set when the Tx_OSize_Frames_Good counter reaches half of the maximum value or the maximum value.	R
31:26	—	These bits are read as 0.	R

27.3.23 MMC_Receive_Interrupt_Mask : MMC Receive Interrupt Mask Register

Base address: GMAC = 0x8010_0000

Offset address: 0x010C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	RXCTRLFIM	RXRCVERRFIM	RXWDOGFIM	RXVLANGBFIM	RXFOVFIM	RXPAUSFIM	RXORANGEFIM	RXLENERFIM	RXUCGFIM	RX1024TMAXOCTGBFIM
Value after reset:	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RX512T1023OCTGBFIM	RX256T511OCTGBFIM	RX128T255OCTGBFIM	RX65T127OCTGBFIM	RX64OCTGBFIM	RXOSIZEGFIM	RXUSIZEGFIM	RXJABERFIM	RXRUNTFIM	RXALGNERFIM	RXCRCERFIM	RXMCGFIM	RXBCGFIM	RXGOCTIM	RXGBOCTIM	RXGBFRMIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RXGBFRMIM	MMC Receive Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Frames_Count_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
1	RXGBOCTIM	MMC Receive Good Bad Octet Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Octet_Count_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
2	RXGOCTIM	MMC Receive Good Octet Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Octet_Count_Good counter reaches half of the maximum value or the maximum value.	R/W
3	RXBCGFIM	MMC Receive Broadcast Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Broadcast_Frames_Good counter reaches half of the maximum value or the maximum value.	R/W
4	RXMCGFIM	MMC Receive Multicast Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Multicast_Frames_Good counter reaches half of the maximum value or the maximum value.	R/W
5	RXCRCERFIM	MMC Receive CRC Error Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_CRC_Error_Frames counter reaches half of the maximum value or the maximum value.	R/W
6	RXALGNERFIM	MMC Receive Alignment Error Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Alignment_Error_Frames counter reaches half of the maximum value or the maximum value.	R/W
7	RXRUNTFIM	MMC Receive Runt Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Runt_Error_Frames counter reaches half of the maximum value or the maximum value.	R/W
8	RXJABERFIM	MMC Receive Jabber Error Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Jabber_Error_Frames counter reaches half of the maximum value or the maximum value.	R/W
9	RXUSIZEGFIM	MMC Receive Undersize Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Undersize_Frames_Good counter reaches half of the maximum value or the maximum value.	R/W
10	RXOSIZEGFIM	MMC Receive Oversize Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Oversize_Frames_Good counter reaches half of the maximum value or the maximum value.	R/W
11	RX64OCTGBFIM	MMC Receive 64 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_64Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
12	RX65T127OCTGBFIM	MMC Receive 65 to 127 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_65To127Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W

Bit	Symbol	Function	R/W
13	RX128T255OCTGBFIM	MMC Receive 128 to 255 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_128To255Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
14	RX256T511OCTGBFIM	MMC Receive 256 to 511 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_256To511Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
15	RX512T1023OCTGBFIM	MMC Receive 512 to 1023 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_512To1023Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
16	RX1024TMAXOCTGBFIM	MMC Receive 1024 to Maximum Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_1024ToMaxOctets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
17	RXUCGFIM	MMC Receive Unicast Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Unicast_Frames_Good counter reaches half of the maximum value or the maximum value.	R/W
18	RXLENERFIM	MMC Receive Length Error Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Length_Error_Frames counter reaches half of the maximum value or the maximum value.	R/W
19	RXORANGEFIM	MMC Receive Out Of Range Error Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Out_Of_Range_Type_Frames counter reaches half of the maximum value or the maximum value.	R/W
20	RXPAUSFIM	MMC Receive Pause Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Pause_Frames counter reaches half of the maximum value or the maximum value.	R/W
21	RXFOVFIM	MMC Receive FIFO Overflow Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_FIFO_Overflow_Frames counter reaches half of the maximum value or the maximum value.	R/W
22	RXVLANGBFIM	MMC Receive VLAN Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_VLAN_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
23	RXWDOGFIM	MMC Receive Watchdog Error Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Watchdog_Error_Frames counter reaches half of the maximum value or the maximum value.	R/W
24	RXRCVERRFIM	MMC Receive Error Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Receive_Error_Frames error counter reaches half the maximum value, and also when it reaches the maximum value.	R/W
25	RXCTRLFIM	MMC Receive Control Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Rx_Control_Frames_Good counter reaches half the maximum value, and also when it reaches the maximum value.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

27.3.24 MMC_Transmit_Interrupt_Mask : MMC Transmit Interrupt Mask Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0110

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	TXOSIZEGFIM	TXVLANGFIM	TXPAUSFIM	TXEXDEFFIM	TXGFRMIM	TXGOCTIM	TXCARERFIM	TXEXCOLFIM	TXLATCOLFIM	TXDEFFIM
Value after reset:	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TXMCOLGFIM	TXSCOLGFIM	TXUFLOWERFIM	TXBCGBFIM	TXMCGBFIM	TXUCGBFIM	TX1024TMAXOCTGBFIM	TX512T1023OCTGBFIM	TX256T511OCTGBFIM	TX128T255OCTGBFIM	TX65T127OCTGBFIM	TX64OCTGBFIM	TXMCGFIM	TXBCGFIM	TXGBFRMIM	TXGBOCTIM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXGBOCTIM	MMC Transmit Good Bad Octet Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Octet_Count_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
1	TXGBFRMIM	MMC Transmit Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Frame_Count_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
2	TXBCGFIM	MMC Transmit Broadcast Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Broadcast_Frames_Good counter reaches half of the maximum value or the maximum value.	R/W
3	TXMCGFIM	MMC Transmit Multicast Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Multicast_Frames_Good counter reaches half of the maximum value or the maximum value.	R/W
4	TX64OCTGBFIM	MMC Transmit 64 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_64Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
5	TX65T127OCTGBFIM	MMC Transmit 65 to 127 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_65To127Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
6	TX128T255OCTGBFIM	MMC Transmit 128 to 255 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_128To255Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
7	TX256T511OCTGBFIM	MMC Transmit 256 to 511 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_256To511Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
8	TX512T1023OCTGBFIM	MMC Transmit 512 to 1023 Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_512To1023Octets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
9	TX1024TMAXOCTGBFIM	MMC Transmit 1024 to Maximum Octet Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_1024ToMaxOctets_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
10	TXUCGBFIM	MMC Transmit Unicast Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Unicast_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
11	TXMCGBFIM	MMC Transmit Multicast Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Multicast_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W

Bit	Symbol	Function	R/W
12	TXBCGBFIM	MMC Transmit Broadcast Good Bad Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Broadcast_Frames_Good_Bad counter reaches half of the maximum value or the maximum value.	R/W
13	TXUFLOWERFIM	MMC Transmit Underflow Error Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Underflow_Error_Frames counter reaches half of the maximum value or the maximum value.	R/W
14	TXSCOLGFIM	MMC Transmit Single Collision Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Single_Collision_Good_Frames counter reaches half of the maximum value or the maximum value.	R/W
15	TXMCOLGFIM	MMC Transmit Multiple Collision Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Multiple_Collision_Good_Frames counter reaches half of the maximum value or the maximum value.	R/W
16	TXDEFFIM	MMC Transmit Deferred Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Deferred_Frames counter reaches half of the maximum value or the maximum value.	R/W
17	TXLATCOLFIM	MMC Transmit Late Collision Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Late_Collision_Frames counter reaches half of the maximum value or the maximum value.	R/W
18	TXEXCOLFIM	MMC Transmit Excessive Collision Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Excessive_Collision_Frames counter reaches half of the maximum value or the maximum value.	R/W
19	TXCARERFIM	MMC Transmit Carrier Error Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Carrier_Error_Frames counter reaches half of the maximum value or the maximum value.	R/W
20	TXGOCTIM	MMC Transmit Good Octet Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Octet_Count_Good counter reaches half of the maximum value or the maximum value.	R/W
21	TXGFRMIM	MMC Transmit Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Frame_Count_Good counter reaches half of the maximum value or the maximum value.	R/W
22	TXEXDEFFIM	MMC Transmit Excessive Deferral Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Excessive_Deferral_Error counter reaches half of the maximum value or the maximum value.	R/W
23	TXPAUSFIM	MMC Transmit Pause Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_Pause_Frames counter reaches half of the maximum value or the maximum value.	R/W
24	TXVLANGFIM	MMC Transmit VLAN Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_VLAN_Frames_Good counter reaches half of the maximum value or the maximum value.	R/W
25	TXOSIZEGFIM	MMC Transmit Oversize Good Frame Counter Interrupt Mask Setting this bit masks the interrupt generated when the Tx_OSize_Frames_Good counter reaches half of the maximum value or the maximum value.	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

27.3.25 Tx_Octet_Count_Good_Bad : Transmit Octet Count for Good and Bad Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0114

Bit position:	31																												0
Bit field:	TXOCTGB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TXOCTGB[31:0]	This field indicates the number of bytes transmitted in good and bad frames exclusive of preamble and retried bytes.	R

27.3.26 Tx_Frame_Count_Good_Bad : Transmit Frame Count for Good and Bad Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0118

Bit position:	31																												0
Bit field:	TXFRMGB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TXFRMGB[31:0]	This field indicates the number of good and bad frames transmitted, exclusive of retried frames.	R

27.3.27 Tx_Broadcast_Frames_Good : Transmit Frame Count for Good Broadcast Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x011C

Bit position:	31																												0
Bit field:	TXBCASTG[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TXBCASTG[31:0]	This field indicates the number of transmitted good broadcast frames.	R

27.3.28 Tx_Multicast_Frames_Good : Transmit Frame Count for Good Multicast Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0120

Bit position:	31																												0
Bit field:	TXMCASTG[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TXMCASTG[31:0]	This field indicates the number of transmitted good multicast frames.	R

27.3.29 Tx_64Octets_Frames_Good_Bad : Transmit Octet Count for Good and Bad 64 Byte Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0124

Bit position:	31																												0
Bit field:	TX64OCTGB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TX64OCTGB[31:0]	This field indicates the number of transmitted good and bad frames with length of 64 bytes, exclusive of preamble and retried frames.	R

27.3.30 Tx_65To127Octets_Frames_Good_Bad : Transmit Octet Count for Good and Bad 65 to 127 Bytes Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0128

Bit position:	31																												0
Bit field:	TX65_127OCTGB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TX65_127OCTGB[31:0]	This field indicates the number of transmitted good and bad frames with length between 65 and 127 (inclusive) bytes, exclusive of preamble and retried frames.	R

27.3.31 Tx_128To255Octets_Frames_Good_Bad : Transmit Octet Count for Good and Bad 128 to 255 Bytes Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x012C

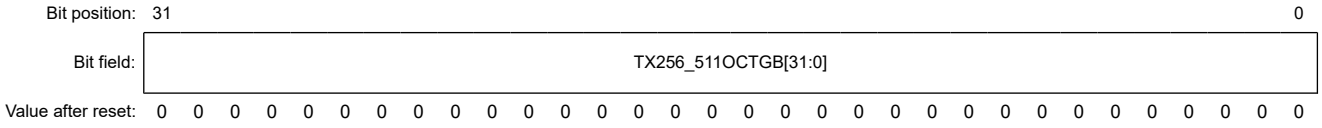
Bit position:	31																												0
Bit field:	TX128_255OCTGB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TX128_255OCTGB[31:0]	This field indicates the number of transmitted good and bad frames with length between 128 and 255 (inclusive) bytes, exclusive of preamble and retried frames.	R

27.3.32 Tx_256To511Octets_Frames_Good_Bad : Transmit Octet Count for Good and Bad 256 to 511 Bytes Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0130

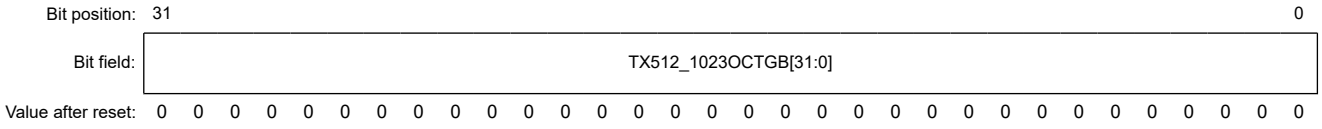


Bit	Symbol	Function	R/W
31:0	TX256_511OCTGB[31:0]	This field indicates the number of transmitted good and bad frames with length between 256 and 511 (inclusive) bytes, exclusive of preamble and retried frames.	R

27.3.33 Tx_512To1023Octets_Frames_Good_Bad : Transmit Octet Count for Good and Bad 512 to 1023 Bytes Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0134

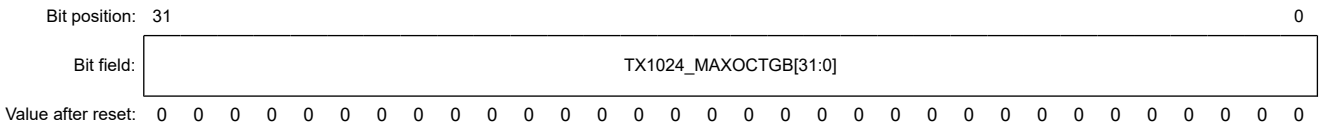


Bit	Symbol	Function	R/W
31:0	TX512_1023OCTGB[31:0]	This field indicates the number of transmitted good and bad frames with length between 512 and 1,023 (inclusive) bytes, exclusive of preamble and retried frames.	R

27.3.34 Tx_1024ToMaxOctets_Frames_Good_Bad : Transmit Octet Count for Good and Bad 1024 to Maxsize Bytes Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0138

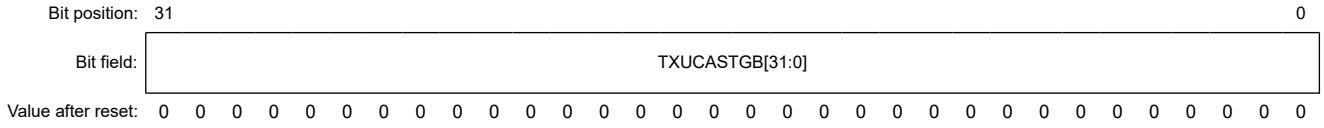


Bit	Symbol	Function	R/W
31:0	TX1024_MAXOCTGB[31:0]	This field indicates the number of good and bad frames transmitted with length between 1,024 and maxsize (inclusive) bytes, exclusive of preamble and retried frames.	R

27.3.35 Tx_Unicast_Frames_Good_Bad : Transmit Frame Count for Good and Bad Unicast Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x013C

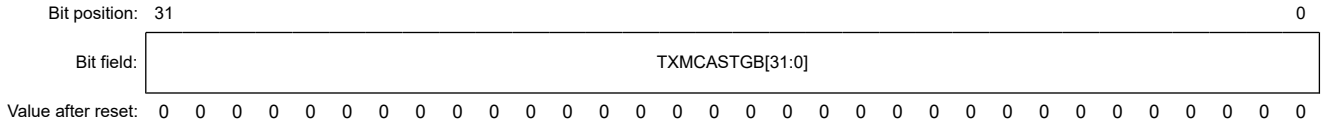


Bit	Symbol	Function	R/W
31:0	TXUCASTGB[31:0]	This field indicates the number of transmitted good and bad unicast frames.	R

27.3.36 Tx_Multicast_Frames_Good_Bad : Transmit Frame Count for Good and Bad Multicast Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0140

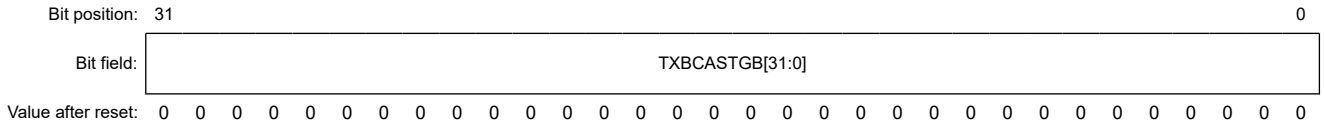


Bit	Symbol	Function	R/W
31:0	TXMCASTGB[31:0]	This field indicates the number of transmitted good and bad multicast frames.	R

27.3.37 Tx_Broadcast_Frames_Good_Bad : Transmit Frame Count for Good and Bad Broadcast Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0144

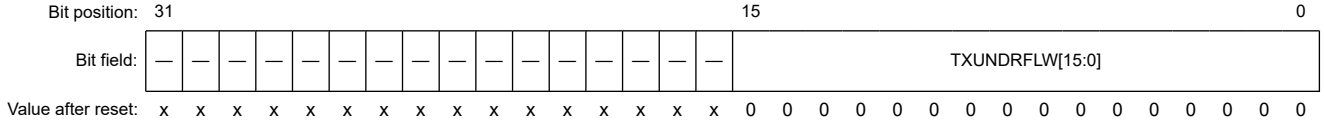


Bit	Symbol	Function	R/W
31:0	TXBCASTGB[31:0]	This field indicates the number of transmitted good and bad broadcast frames.	R

27.3.38 Tx_Underflow_Error_Frames : Transmit Frame Count for Underflow Error Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0148



Bit	Symbol	Function	R/W
15:0	TXUNDRFLW[15:0]	This field indicates the number of frames aborted because of frame underflow error.	R
31:16	—	These bits are read as 0.	R

27.3.39 Tx_Single_Collision_Good_Frames : Transmit Frame Count for Frames Transmitted after Single Collision

Base address: GMAC = 0x8010_0000

Offset address: 0x014C

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXSNGLCOLG[15:0]														
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TXSNGLCOLG[15:0]	This field indicates the number of successfully transmitted frames after a single collision in the half-duplex mode.	R
31:16	—	These bits are read as 0.	R

27.3.40 Tx_Multiple_Collision_Good_Frames : Transmit Frame Count for Frames Transmitted after Multiple Collision

Base address: GMAC = 0x8010_0000

Offset address: 0x0150

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXMULTCOLG[15:0]														
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TXMULTCOLG[15:0]	This field indicates the number of successfully transmitted frames after multiple collisions in the half-duplex mode.	R
31:16	—	These bits are read as 0.	R

27.3.41 Tx_Deferred_Frames : Transmit Frame Count for Deferred Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0154

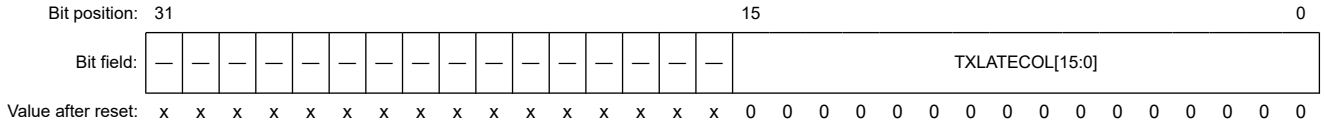
Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXDEFRD[15:0]														
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TXDEFRD[15:0]	This field indicates the number of successfully transmitted frames after a deferral in the half-duplex mode.	R
31:16	—	These bits are read as 0.	R

27.3.42 Tx_Late_Collision_Frames : Transmit Frame Count for Late Collision Error Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0158

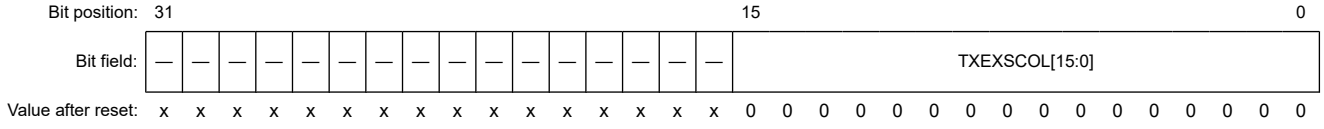


Bit	Symbol	Function	R/W
15:0	TXLATECOL[15:0]	This field indicates the number of frames aborted because of late collision error.	R
31:16	—	These bits are read as 0.	R

27.3.43 Tx_Excessive_Collision_Frames : Transmit Frame Count for Excessive Collision Error Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x015C

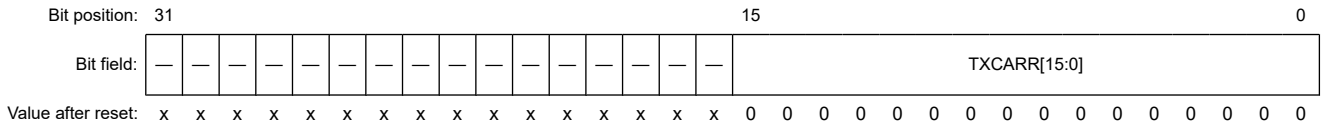


Bit	Symbol	Function	R/W
15:0	TXEXSCOL[15:0]	This field indicates the number of frames aborted because of excessive (16) collision error.	R
31:16	—	These bits are read as 0.	R

27.3.44 Tx_Carrier_Error_Frames : Transmit Frame Count for Carrier Sense Error Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0160

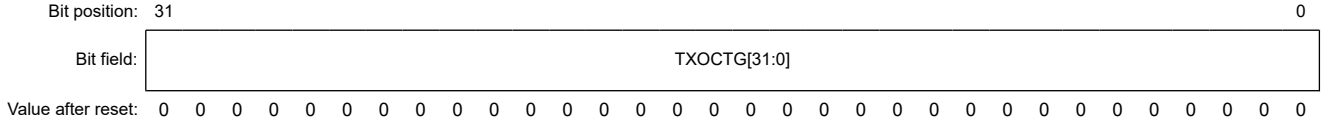


Bit	Symbol	Function	R/W
15:0	TXCARR[15:0]	This field indicates the number of frames aborted because of carrier sense error (no carrier or loss of carrier).	R
31:16	—	These bits are read as 0.	R

27.3.45 Tx_Octet_Count_Good : Transmit Octet Count for Good Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0164

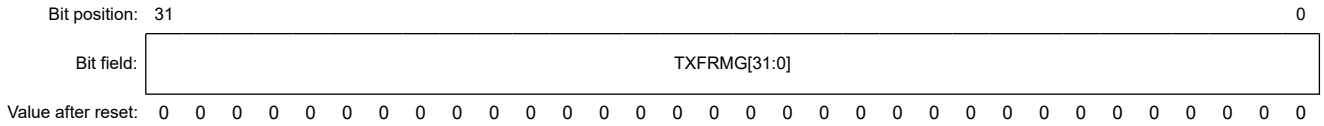


Bit	Symbol	Function	R/W
31:0	TXOCTG[31:0]	This field indicates the number of bytes transmitted, exclusive of preamble, in good frames.	R

27.3.46 Tx_Frame_Count_Good : Transmit Frame Count for Good Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0168

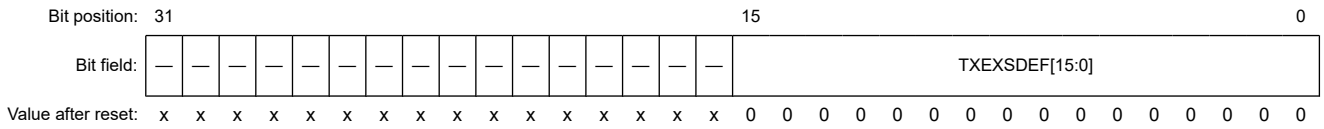


Bit	Symbol	Function	R/W
31:0	TXFRMG[31:0]	This field indicates the number of transmitted good frames, exclusive of preamble.	R

27.3.47 Tx_Excessive_Deferral_Error : Transmit Frame Count for Excessive Deferral Error Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x016C

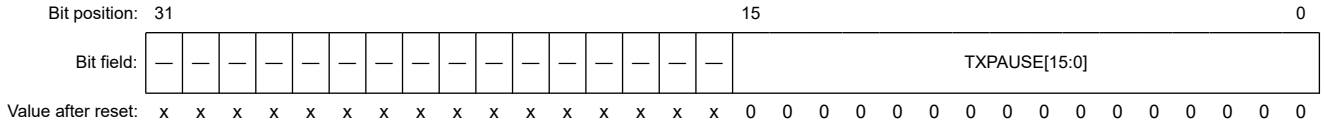


Bit	Symbol	Function	R/W
15:0	TXEXSDEF[15:0]	This field indicates the number of frames aborted because of excessive deferral error, that is, frames deferred for more than two max sized frame times.	R
31:16	—	These bits are read as 0.	R

27.3.48 Tx_Pause_Frames : Transmit Frame Count for Good PAUSE Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0170



Bit	Symbol	Function	R/W
15:0	TXPAUSE[15:0]	This field indicates the number of transmitted good PAUSE frames.	R
31:16	—	These bits are read as 0.	R

27.3.49 Tx_VLAN_Frames_Good : Transmit Frame Count for Good VLAN Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0174

Bit position:	31																												0
Bit field:	TXVLANG[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TXVLANG[31:0]	This register maintains the number of transmitted good VLAN frames, exclusive of retried frames.	R

27.3.50 Tx_OSize_Frames_Good : Transmit Frame Count for Good Oversize Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0178

Bit position:	31															15													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXOSIZG[15:0]													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TXOSIZG[15:0]	This field indicates the number of frames transmitted without errors and with length greater than the maxsize (1,518 or 1,522 bytes for VLAN tagged frames; 2000 bytes if enabled in bit [27] of MAC Configuration Register (MAC_Configuration)).	R
31:16	—	These bits are read as 0.	R

27.3.51 Rx_Frames_Count_Good_Bad : Receive Frame Count for Good and Bad Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0180

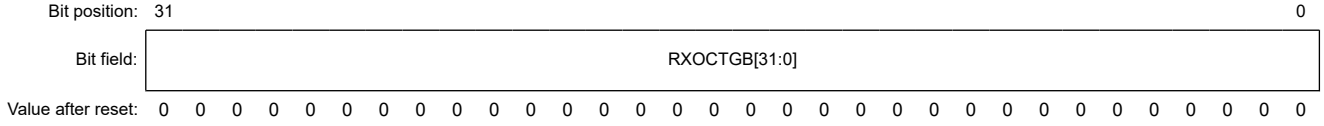
Bit position:	31																												0
Bit field:	RXFRMGB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RXFRMGB[31:0]	This field indicates the number of received good and bad frames.	R

27.3.52 Rx_Octet_Count_Good_Bad : Receive Octet Count for Good and Bad Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0184

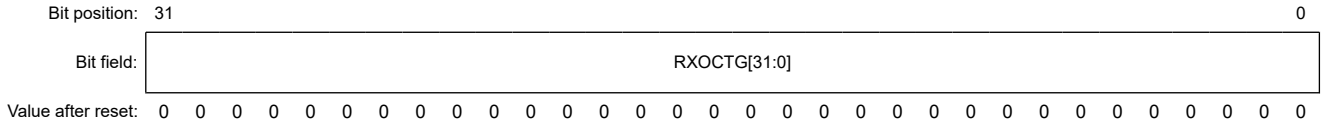


Bit	Symbol	Function	R/W
31:0	RXOCTGB[31:0]	This field indicates the number of bytes received, exclusive of preamble, in good and bad frames.	R

27.3.53 Rx_Octet_Count_Good : Receive Octet Count for Good Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0188

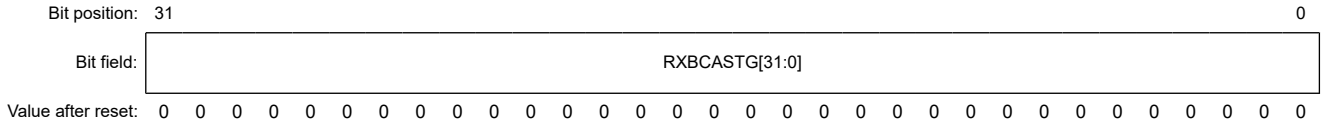


Bit	Symbol	Function	R/W
31:0	RXOCTG[31:0]	This field indicates the number of bytes received, exclusive of preamble, only in good frames.	R

27.3.54 Rx_Broadcast_Frames_Good : Receive Frame Count for Good Broadcast Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x018C

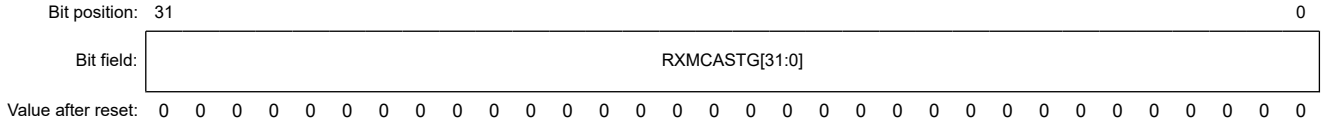


Bit	Symbol	Function	R/W
31:0	RXBCASTG[31:0]	This field indicates the number of received good broadcast frames.	R

27.3.55 Rx_Multicast_Frames_Good : Receive Frame Count for Good Multicast Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0190



Bit	Symbol	Function	R/W
31:0	RXMCASTG[31:0]	This field indicates the number of received good multicast frames.	R

27.3.56 Rx_CRC_Error_Frames : Receive Frame Count for CRC Error Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0194

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RXCRCERR[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RXCRCERR[15:0]	This field indicates the number of frames received with CRC error.	R
31:16	—	These bits are read as 0.	R

27.3.57 Rx_Alignment_Error_Frames : Receive Frame Count for Alignment Error Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x0198

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RXALGNERR[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RXALGNERR[15:0]	This field indicates the number of frames received with alignment (dribble) error. This field is valid only in the 10 or 100 Mbps mode.	R
31:16	—	These bits are read as 0.	R

27.3.58 Rx_Runt_Error_Frames : Receive Frame Count for Runt Error Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x019C

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RXRUNTERR[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RXRUNTERR[15:0]	This field indicates the number of frames received with runt error (< 64 bytes and CRC error).	R
31:16	—	These bits are read as 0.	R

27.3.59 Rx_Jabber_Error_Frames : Receive Frame Count for Jabber Error Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x01A0

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RXJABERR[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RXJABERR[15:0]	This field indicates the number of giant frames received with length (including CRC) greater than 1,518 bytes (1,522 bytes for VLAN tagged) and with CRC error. If Jumbo Frame mode is enabled, then frames of length greater than 9,018 bytes (9,022 for VLAN tagged) are considered as giant frames.	R
31:16	—	These bits are read as 0.	R

27.3.60 Rx_Undersize_Frames_Good : Receive Frame Count for Undersize Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x01A4

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RXUNDERSZG[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RXUNDERSZG[15:0]	This field indicates the number of frames received with length less than 64 bytes and without errors.	R
31:16	—	These bits are read as 0.	R

27.3.61 Rx_Oversize_Frames_Good : Receive Frame Count for Oversize Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x01A8

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RXOVERSZG[15:0]															
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RXOVERSZG[15:0]	This field indicates the number of frames received without errors, with length greater than the maxsize (1,518 or 1,522 for VLAN tagged frames; 2,000 bytes if enabled in bit [27] of MAC Configuration Register (MAC_Configuration)).	R
31:16	—	These bits are read as 0.	R

27.3.62 Rx_64Octets_Frames_Good_Bad : Receive Frame Count for Good and Bad 64 Byte Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x01AC

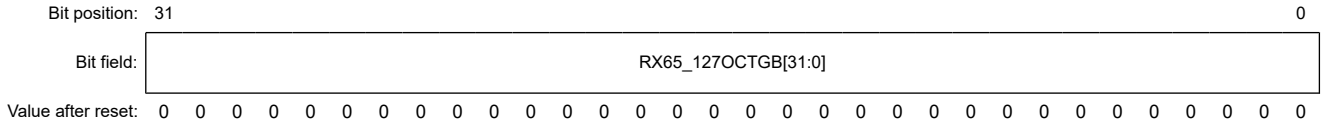
Bit position:	31																														0
Bit field:	RX64OCTGB[31:0]																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RX64OCTGB[31:0]	This field indicates the number of received good and bad frames with length 64 bytes, exclusive of preamble.	R

27.3.63 Rx_65To127Octets_Frames_Good_Bad : Receive Frame Count for Good and Bad 65 to 127 Bytes Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x01B0

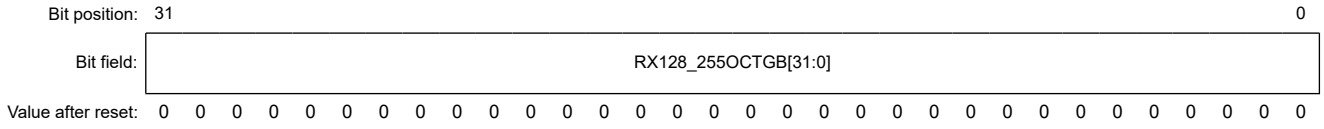


Bit	Symbol	Function	R/W
31:0	RX65_127OCTGB[31:0]	This field indicates the number of received good and bad frames received with length between 65 and 127 (inclusive) bytes, exclusive of preamble.	R

27.3.64 Rx_128To255Octets_Frames_Good_Bad : Receive Frame Count for Good and Bad 128 to 255 Bytes Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x01B4

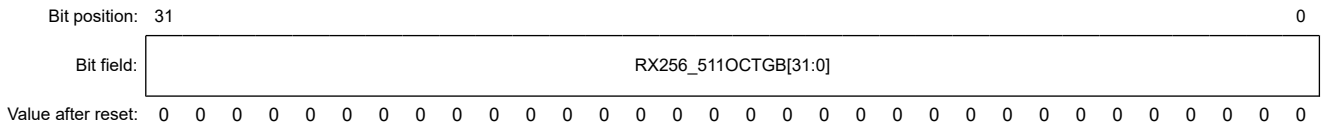


Bit	Symbol	Function	R/W
31:0	RX128_255OCTGB[31:0]	This field indicates the number of received good and bad frames with length between 128 and 255 (inclusive) bytes, exclusive of preamble.	R

27.3.65 Rx_256To511Octets_Frames_Good_Bad : Receive Frame Count for Good and Bad 256 to 511 Bytes Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x01B8

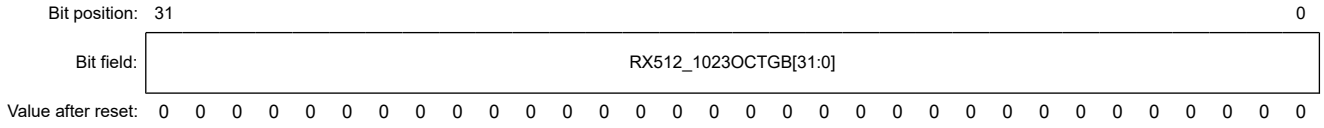


Bit	Symbol	Function	R/W
31:0	RX256_511OCTGB[31:0]	This field indicates the number of received good and bad frames with length between 256 and 511 (inclusive) bytes, exclusive of preamble.	R

27.3.66 Rx_512To1023Octets_Frames_Good_Bad : Receive Frame Count for Good and Bad 512 to 1,023 Bytes Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x01BC

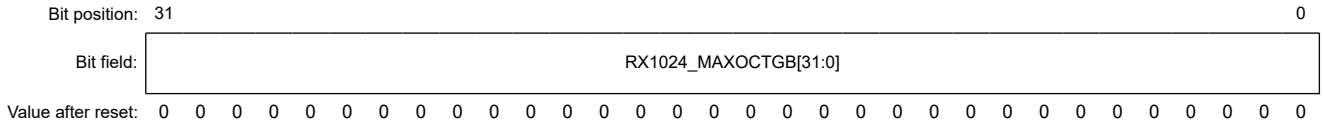


Bit	Symbol	Function	R/W
31:0	RX512_1023OCTGB[31:0]	This field indicates the number of received good and bad frames with length between 512 and 1,023 (inclusive) bytes, exclusive of preamble.	R

27.3.67 Rx_1024ToMaxOctets_Frames_Good_Bad : Receive Frame Count for Good and Bad 1,024 to Maxsize Bytes Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x01C0

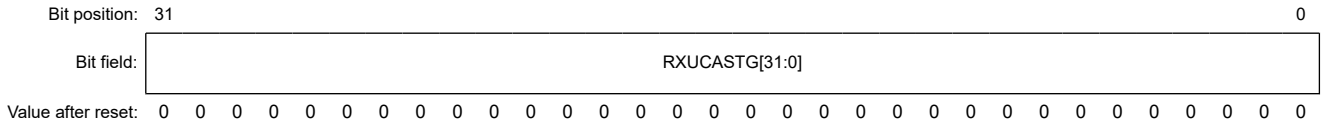


Bit	Symbol	Function	R/W
31:0	RX1024_MAXOCTGB[31:0]	This field indicates the number of received good and bad frames with length between 1,024 and maxsize (inclusive) bytes, exclusive of preamble and retried frames.	R

27.3.68 Rx_Unicast_Frames_Good : Receive Frame Count for Good Unicast Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x01C4

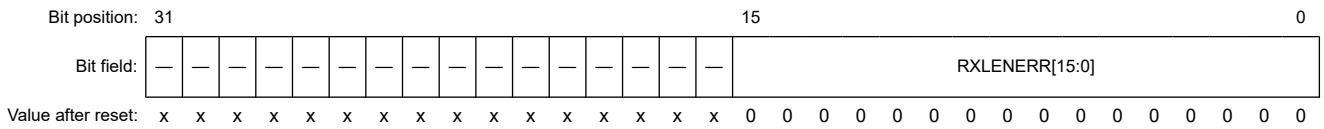


Bit	Symbol	Function	R/W
31:0	RXUCASTG[31:0]	This field indicates the number of received good unicast frames.	R

27.3.69 Rx_Length_Error_Frames : Receive Frame Count for Length Error Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x01C8



Bit	Symbol	Function	R/W
15:0	RXLENERR[15:0]	This field indicates the number of frames received with length error (Length type field not equal to frame size) for all frames with valid length field.	R
31:16	—	These bits are read as 0.	R

27.3.70 Rx_Out_Of_Range_Type_Frames : Receive Frame Count for Out of Range Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x01CC

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RXOUTOFRNG[15:0]														
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RXOUTOFRNG[15:0]	This field indicates the number of received frames with length field not equal to the valid frame size (greater than 1,500 but less than 1,536).	R
31:16	—	These bits are read as 0.	R

27.3.71 Rx_Pause_Frames : Receive Frame Count for PAUSE Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x01D0

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RXPAUSEFRM[15:0]														
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RXPAUSEFRM[15:0]	This field indicates the number of received good and valid PAUSE frames.	R
31:16	—	These bits are read as 0.	R

27.3.72 Rx_FIFO_Overflow_Frames : Receive Frame Count for FIFO Overflow Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x01D4

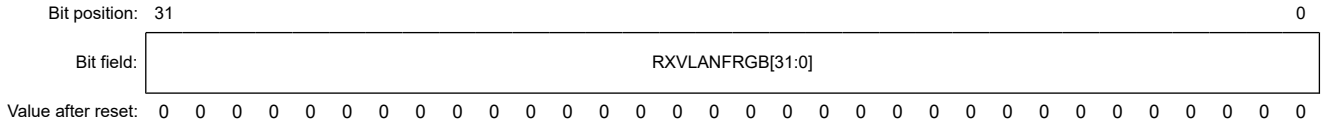
Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RXFIFOOVFL[15:0]														
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RXFIFOOVFL[15:0]	This field indicates the number of received frames missed because of FIFO overflow.	R
31:16	—	These bits are read as 0.	R

27.3.73 Rx_VLAN_Frames_Good_Bad : Receive Frame Count for Good and Bad VLAN Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x01D8

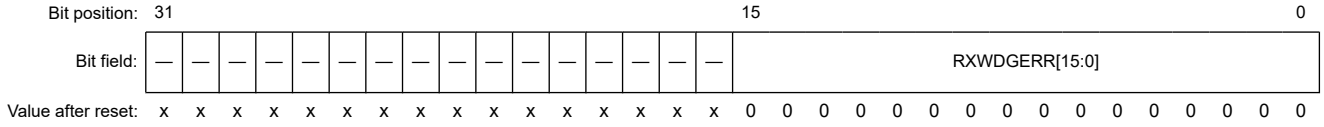


Bit	Symbol	Function	R/W
31:0	RXVLANFRGB[31:0]	This field indicates the number of received good and bad VLAN frames.	R

27.3.74 Rx_Watchdog_Error_Frames : Receive Frame Count for Watchdog Error Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x01DC

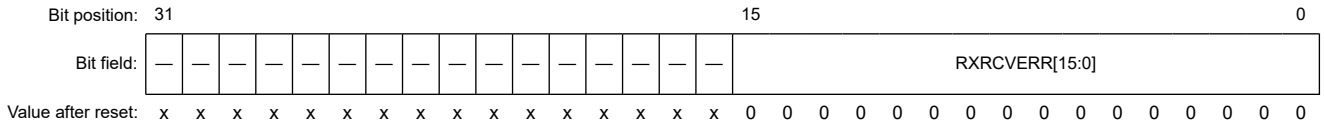


Bit	Symbol	Function	R/W
15:0	RXWDGERR[15:0]	This field indicates the number of frames received with error because of the watchdog timeout error (frames with more than 2,048 bytes or value programmed in Watchdog Timeout Register (WDog_Timeout)).	R
31:16	—	These bits are read as 0.	R

27.3.75 Rx_Receive_Error_Frames : Receive Frame Count for Receive Error Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x01E0

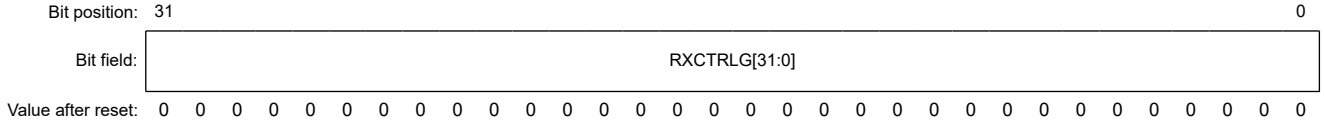


Bit	Symbol	Function	R/W
15:0	RXRCVERR[15:0]	This field indicates the number of frames received with error because of the GMII/MII RXER error or Frame Extension error on GMII.	R
31:16	—	These bits are read as 0.	R

27.3.76 Rx_Control_Frames_Good : Receive Frame Count for Good Control Frames

Base address: GMAC = 0x8010_0000

Offset address: 0x01E4



Bit	Symbol	Function	R/W
31:0	RXCTRLG[31:0]	This field indicates the number of good control frames received.	R

27.3.77 HASH_TABLE_REGn : Hash Table Register n (n = 0 to 7)

Base address: GMAC = 0x8010_0000

Offset address: 0x0500 + 0x4 × n

Bit position:	31																													0
Bit field:	HT[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	HT[31:0]	This field contains the nth 32 bits [31:0] of the Hash table.	R/W

27.3.78 VLAN_Hash_Table_Reg : VLAN Hash Table Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0588

Bit position:	31																15													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VLHT[15:0]													
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	VLHT[15:0]	VLAN Hash Table This field contains the 16-bit VLAN Hash Table.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

27.3.79 Timestamp_Control : Timestamp Control Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0700

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	ATSEN1	ATSEN0	ATSFC	—	—	—	—	—	TSENMACADDR	SNAPTYPSEL	
Value after reset:	x	x	x	x	x	0	0	0	x	x	x	x	x	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TSMSTRENA	TSEVNTENA	TSIPV4ENA	TSIPV6ENA	TSIPENA	TSVER2ENA	TSCTRLSSR	TSENALL	—	—	TSADDREG	TSTRIG	TSUPDT	TSINIT	TSCFUPDT	TSENA
Value after reset:	0	0	1	0	0	0	0	0	x	x	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSENA	Timestamp Enable When set, the timestamp is added for the transmit and receive frames. When disabled, timestamp is not added for the transmit and receive frames and the Timestamp Generator is also suspended. You need to initialize the Timestamp (system time) after enabling this mode. On the receive side, the GMAC processes the IEEE 1588 frames only if this bit is set.	R/W

Bit	Symbol	Function	R/W
1	TSCFUPDT	Timestamp Fine or Coarse Update When set, this bit indicates that the system times update should be done using the fine update method. When reset, it indicates the system timestamp update should be done using the Coarse method.	R/W
2	TSINIT	Timestamp Initialize When set, the system time is initialized (overwritten) with the value specified in the System Time – Seconds Update Register (System_Time_Seconds_Update) and System Time – Nanoseconds Update Register (System_time_Nanoseconds_Update). This bit should be read zero before updating it. This bit is reset when the initialization is complete.	R/W
3	TSUPDT	Timestamp Update When set, the system time is updated (added or subtracted) with the value specified in System Time – Seconds Update Register (System_Time_Seconds) and System Time – Nanoseconds Update Register (System_Time_Nanoseconds). This bit should be read zero before updating it. This bit is reset when the update is completed in hardware.	R/W
4	TSTRIG	Timestamp Interrupt Trigger Enable When set, the timestamp interrupt is generated when the System Time becomes greater than the value written in the Target Time register. This bit is reset after the generation of the Timestamp Trigger Interrupt.	R/W
5	TSADDREG	Addend Reg Update When set, the content of the Timestamp Addend register is updated in the PTP block for fine correction. This is cleared when the update is completed. This register bit should be zero before setting it.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	TSENALL	Enable Timestamp for all Frames When set, the timestamp snapshot is enabled for all frames received by the GMAC.	R/W
9	TSCTRLSSR	Timestamp Digital or Binary Rollover Control When set, the Timestamp Low register rolls over after 0x3B9A_C9FF value (that is, 1 nanosecond accuracy) and increments the timestamp (high) seconds. When reset, the rollover value of sub second register is 0x7FFF_FFFF. The sub second increment must be programmed correctly depending on the PTP reference clock frequency and the value of this bit.	R/W
10	TSVER2ENA	Enable PTP packet Processing for Version 2 Format When set, the PTP packets are processed using the IEEE 1588 version 2 format. Otherwise, the PTP packets are processed using the version 1 format.	R/W
11	TSIPENA	Enable Processing of PTP over Ethernet Frames When set, the GMAC receiver processes the PTP packets encapsulated directly in the Ethernet frames. When this bit is clear, the GMAC ignores the PTP over Ethernet packets.	R/W
12	TSIPV6ENA	Enable Processing of PTP Frames Sent Over IPv6 UDP When set, the GMAC receiver processes PTP packets encapsulated in UDP over IPv6 packets. When this bit is clear, the GMAC ignores the PTP transported over UDP IPv6 packets.	R/W
13	TSIPV4ENA	Enable Processing of PTP Frames Sent over IPv4 UDP When set, the GMAC receiver processes the PTP packets encapsulated in UDP over IPv4 packets. When this bit is clear, the GMAC ignores the PTP transported over UDP IPv4 packets. This bit is set by default.	R/W
14	TSEVNTENA	Enable Timestamp Snapshot for Event Messages When set, the timestamp snapshot is taken only for event messages (SYNC, Delay_Req, Pdelay_Req, or Pdelay_Resp). When reset, the snapshot is taken for all messages except Announce, Management, and Signaling.	R/W
15	TSMSTRENA	Enable Snapshot for Messages Relevant to Master When set, the snapshot is taken only for the messages relevant to the master node. Otherwise, the snapshot is taken for the messages relevant to the slave node.	R/W
17:16	SNAPTYPSEL	Select PTP packets for Taking Snapshots These bits along with bits [15] and [14] determine the set of PTP packet types for which snapshot must be taken.	R/W

Bit	Symbol	Function	R/W
18	TSENMACADDR	Enable MAC address for PTP Frame Filtering When set, the DA MAC address (that matches any MAC Address register) is used to filter the PTP frames when PTP is directly sent over Ethernet.	R/W
23:19	—	These bits are read as 0. The write value should be 0.	R/W
24	ATSFC	Auxiliary Snapshot FIFO Clear When set, it resets the pointers of the Auxiliary Snapshot FIFO. This bit is cleared when the pointers are reset and the FIFO is empty. When this bit is high, auxiliary snapshots get stored in the FIFO.	R/W
25	ATSEN0	Auxiliary Snapshot 0 Enable This field controls capturing the Auxiliary Snapshot Trigger 0. When this bit is set, the Auxiliary snapshot of event on GMAC_PTPTRG0 input is enabled. When this bit is reset, the events on this input are ignored.	R/W
26	ATSEN1	Auxiliary Snapshot 1 Enable This field controls capturing the Auxiliary Snapshot Trigger 1. When this bit is set, the Auxiliary snapshot of event on GMAC_PTPTRG1 input is enabled. When this bit is reset, the events on this input are ignored.	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

27.3.80 Sub_Second_Increment : Sub-Second Increment Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0704

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SSINC[7:0]							
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SSINC[7:0]	Sub-second Increment Value The value programmed in this field is accumulated every clock cycle (of GMAC_PTP_REFCLK_I) with the contents of the sub-second register. For example, when PTP clock is 50 MHz (period is 20 ns), you should program 20 (0x14) when the System Time-Nanoseconds register has an accuracy of 1 ns (TSCTRLSSR bit is set). When TSCTRLSSR is clear, the Nanoseconds register has a resolution of ~0.465 ns. In this case, you should program a value of 43 (0x2B) that is derived by 20 ns/0.465 ns.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

27.3.81 System_Time_Seconds : System Time - Seconds Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0708

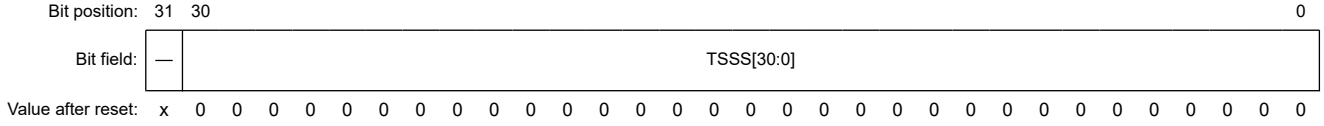
Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TSS[31:0]																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	TSS[31:0]	Timestamp Second The value in this field indicates the current value in seconds of the System Time maintained by the GMAC.	R

27.3.82 System_Time_Nanoseconds : System Time - Nanoseconds Register

Base address: GMAC = 0x8010_0000

Offset address: 0x070C

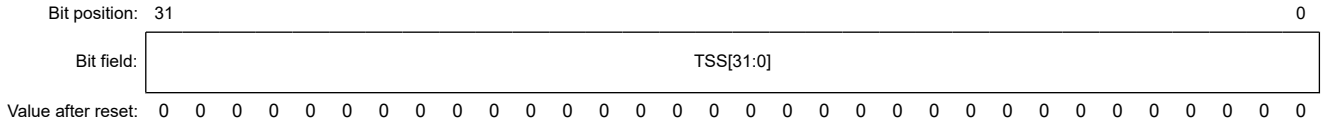


Bit	Symbol	Function	R/W
30:0	TSSS[30:0]	Timestamp Sub Seconds The value in this field has the sub-second representation of time, with an accuracy of 0.46 ns. When bit 9 (TSCTRLSSR) is set in Timestamp Control Register (Timestamp_Control), each bit represents 1 ns and the maximum value is 0x3B9A_C9FF, after which it rolls over to zero.	R
31	—	This bit is read as 0.	R

27.3.83 System_Time_Seconds_Update : System Time - Seconds Update Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0710

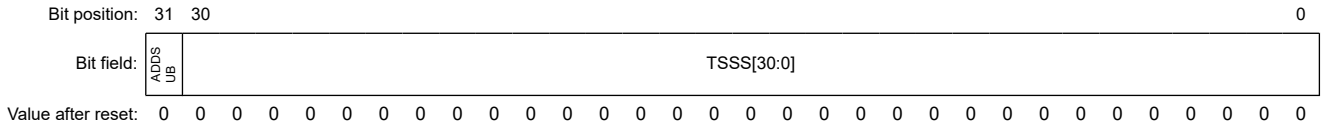


Bit	Symbol	Function	R/W
31:0	TSS[31:0]	Timestamp Second The value in this field indicates the time in seconds to be initialized or added to the system time.	R/W

27.3.84 System_Time_Nanoseconds_Update : System Time - Nanoseconds Update Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0714

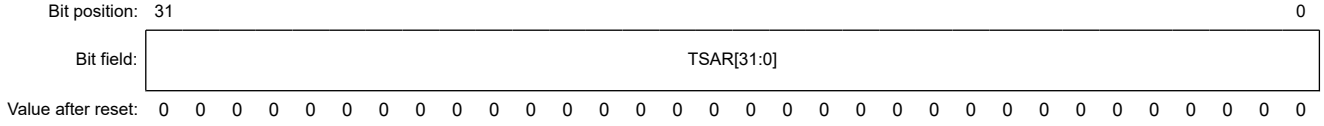


Bit	Symbol	Function	R/W
30:0	TSSS[30:0]	Timestamp Sub Seconds The value in this field has the sub-second representation of time, with an accuracy of 0.46 ns. When bit 9 (TSCTRLSSR) is set in Timestamp Control Register (Timestamp_Control), each bit represents 1 ns and the programmed value should not exceed 0x3B9A_C9FF.	R/W
31	ADDSUB	Add or subtract time When this bit is set, the time value is subtracted with the contents of the update register. When this bit is reset, the time value is added with the contents of the update register.	R/W

27.3.85 Timestamp_Addend : Timestamp Addend Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0718

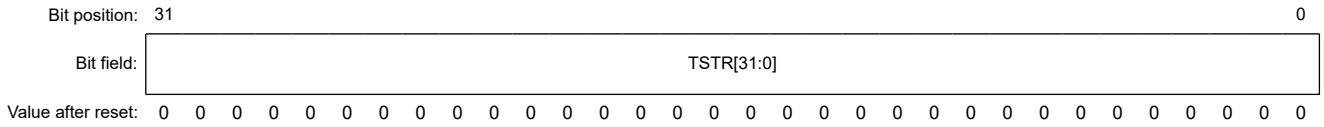


Bit	Symbol	Function	R/W
31:0	TSAR[31:0]	Timestamp Addend This field indicates the 32-bit time value to be added to the Accumulator to achieve time synchronization.	R/W

27.3.86 Target_Time_Seconds : Target Time Seconds Register

Base address: GMAC = 0x8010_0000

Offset address: 0x071C

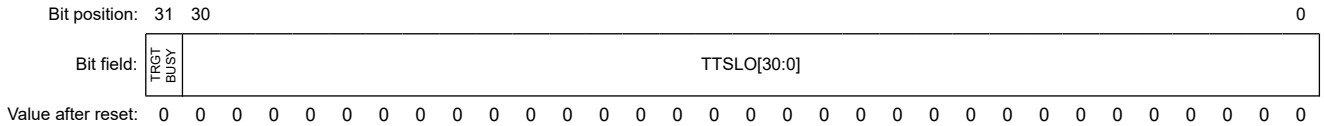


Bit	Symbol	Function	R/W
31:0	TSTR[31:0]	Target Time Seconds This register stores the time in seconds. When the timestamp value matches or exceeds both Target Timestamp registers, then based on bits [6:5] of PPS Control Register (PPS_Control), the GMAC starts or stops the PPS signal output and generates an interrupt (if enabled).	R/W

27.3.87 Target_Time_Nanoseconds : Target Time Nanoseconds Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0720



Bit	Symbol	Function	R/W
30:0	TTSLO[30:0]	Target Timestamp Low Register This register stores the time in (signed) nanoseconds. When the value of the timestamp matches both Target Timestamp registers, then based on the TRGTMODSEL0 field (bits [6:5]) in PPS Control Register (PPS_Control), the GMAC starts or stops the PPS signal output and generates an interrupt (if enabled). This value should not exceed 0x3B9A_C9FF when TSCTRLSSR is set in the Timestamp control register. The actual start or stop time of the PPS signal output may have an error margin up to one unit of sub-second increment value.	R/W

Bit	Symbol	Function	R/W
31	TRGTBUSY	Target Time Register Busy The GMAC sets this bit when the PPSCMD field (bits [2:0]) in PPS Control Register (PPS_Control) is programmed to 010b or 011b. Programming the PPSCMD field to 010b or 011b, instructs the GMAC to synchronize the Target Time Registers to the PTP clock domain. The GMAC clears this bit after synchronizing the Target Time Registers to the PTP clock domain. The application must not update the Target Time Registers when this bit is read as 1. Otherwise, the synchronization of the previously programmed time gets corrupted.	R

27.3.88 Timestamp_Status : Timestamp Status Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0728

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	ATSNS[4:0]					ATSSTM	—	—	—	—	ATSSTN[3:0]			
Value after reset:	x	x	0	0	0	0	0	0	x	x	x	x	0	0	0	0

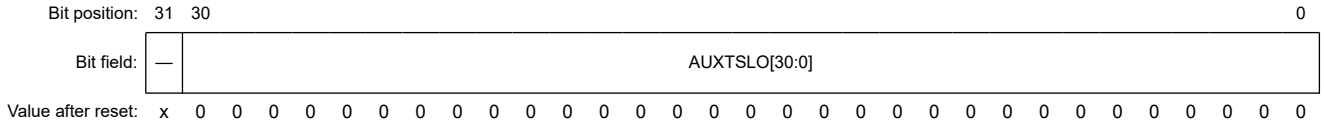
Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	AUXTSTRIG	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	0	x	x

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0.	R
2	AUXTSTRIG	Auxiliary Timestamp Trigger Snapshot This bit is set high when the auxiliary snapshot is written to the FIFO.	R
15:3	—	These bits are read as 0.	R
19:16	ATSSTN[3:0]	Auxiliary Timestamp Snapshot Trigger Identifier These bits identify the Auxiliary trigger inputs for which the timestamp available in the Auxiliary Snapshot Register is applicable. When more than one bit is set at the same time, it means that corresponding auxiliary triggers were sampled at the same clock. These bits are applicable only if the number of Auxiliary snapshots is more than one. One bit is assigned for each trigger as shown in the following list: Bit [16]: Auxiliary trigger 0 Bit [17]: Auxiliary trigger 1 Bit [18]: Auxiliary trigger 2 (not used) Bit [19]: Auxiliary trigger 3 (not used) The software can read this register to find the triggers that are set when the timestamp is taken.	R
23:20	—	These bits are read as 0.	R
24	ATSSTM	Auxiliary Timestamp Snapshot Trigger Missed This bit is set when the Auxiliary timestamp snapshot FIFO is full and external trigger was set. This indicates that the latest snapshot is not stored in the FIFO.	R
29:25	ATSNS[4:0]	Number of Auxiliary Timestamp Snapshots This field indicates the number of snapshots available in the FIFO. A value equal to the selected depth of FIFO (4) indicates that the Auxiliary Snapshot FIFO is full. These bits are cleared (to 00000b) when the Auxiliary snapshot FIFO clear bit is set.	R
31:30	—	These bits are read as 0.	R

27.3.89 Auxiliary_Timestamp_Nanoseconds : Auxiliary Timestamp – Nanoseconds Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0730

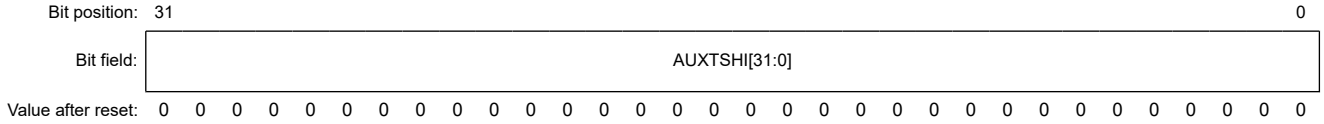


Bit	Symbol	Function	R/W
30:0	AUXTSLO[30:0]	Contains the lower 32 bits (nanoseconds field) of the auxiliary timestamp.	R
31	—	This bit is read as 0.	R

27.3.90 Auxiliary_Timestamp_Seconds : Auxiliary Timestamp - Seconds Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0734

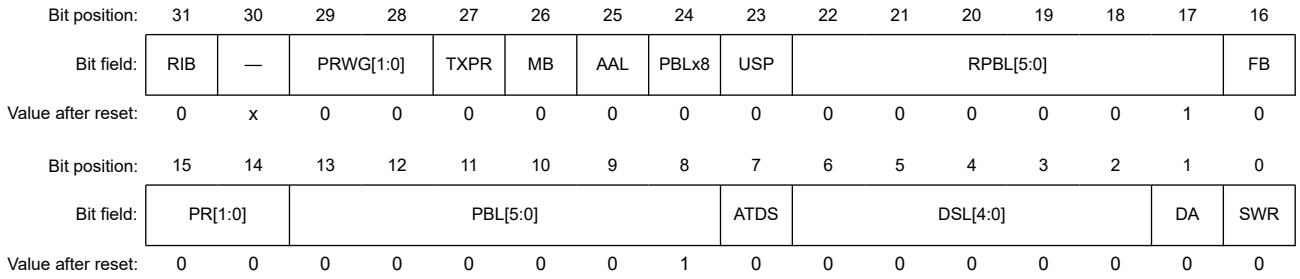


Bit	Symbol	Function	R/W
31:0	AUXTSHI[31:0]	Contains the upper 32 bits (Seconds field) of the auxiliary timestamp.	R

27.3.91 Bus_Mode : Bus Mode Register

Base address: GMAC = 0x8010_0000

Offset address: 0x1000



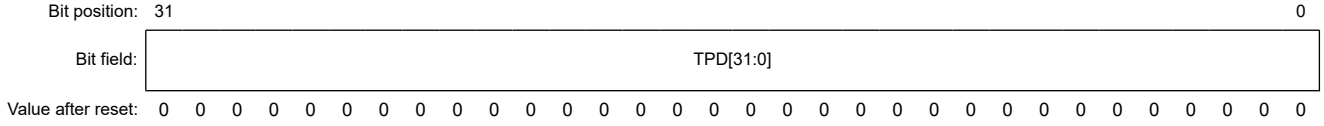
Bit	Symbol	Function	R/W
0	SWR	Software Reset When this bit is set, the MAC DMA Controller resets the logic and all internal registers of the GMAC. It is cleared automatically after the reset operation has completed in all of the GMAC clock domains. Before reprogramming any register of the GMAC, read a 0 value in this bit. The Software reset function is driven only by this bit. The reset operation is completed only when all resets in all active clock domains are deasserted. Therefore, it is essential that all the PHY inputs clocks (applicable for the selected PHY interface) are present for the software reset completion. The time to complete the software reset operation depends on the frequency of the slowest active clock.	R/W

Bit	Symbol	Function	R/W
1	DA	DMA Arbitration Scheme This bit is reserved and read-only (RO).	R
6:2	DSL[4:0]	Descriptor Skip Length This bit specifies the number of word to skip between two unchained descriptors. The address skipping starts from the end of current descriptor to the start of next descriptor. When the DSL value is equal to 0, the descriptor table is taken as contiguous by the DMA in Ring mode.	R/W
7	ATDS	Enhanced Descriptor Size When set, the size of the enhanced descriptor increases to 32 bytes (8 DWORDS). When reset, the descriptor size reverts back to 4 DWORDs (16 bytes).	R/W
13:8	PBL[5:0]	Programmable Burst Length These bits indicate the maximum number of beats to be transferred in one DMA transaction. This is the maximum value that is used in a single block read or write. The DMA always attempts to burst as specified in PBL each time it starts a burst transfer on the host bus. PBL can be programmed with permissible values of 1, 2, 4, 8, and 16. Any other value results in undefined behavior. When USP is set high, this PBL value is applicable only for TX DMA transactions.	R/W
15:14	PR[1:0]	Priority Ratio These bits are reserved and read-only (RO).	R
16	FB	Fixed Burst This bit controls whether the AXI Master interface performs fixed burst transfers or not. When set, the AXI interface uses only SINGLE, INCR4, INCR8, or INCR16 during start of the normal burst transfers. When reset, the AXI interface uses SINGLE and INCR burst transfer operations. For more information, see bit UNDEF[0] of the AXI Bus Mode register.	R/W
22:17	RPBL[5:0]	RX DMA PBL This field indicates the maximum number of beats to be transferred in one RX DMA transaction. This is the maximum value that is used in a single block read or write. The RX DMA always attempts to burst as specified in the RPBL bit each time it starts a burst transfer on the host bus. You can program RPBL with values of 1, 2, 4, 8, and 16. Any other value results in undefined behavior. This field is valid and applicable only when USP is set high.	R/W
23	USP	Use Separate PBL When set high, this bit configures the RX DMA to use the value configured in bits [22:17] as PBL. The PBL value in bits [13:8] is applicable only to the TX DMA operations. When reset to low, the PBL value in bits [13:8] is applicable for both DMA engines.	R/W
24	PBLx8	PBLx8 Mode When set high, this bit multiplies the programmed PBL value (bits [22:17] and bits [13:8]) eight times. Therefore, the DMA transfers the data in 8, 16 beats depending on the PBL value. The maximum number of beats to be transferred in one DMA transaction is 16 beats.	R/W
25	AAL	Address Aligned Beats When this bit is set high and the FB bit is equal to 1, the AXI interface generates all bursts aligned to the start address LS bits. If the FB bit is equal to 0, the first burst (accessing the start address of the data buffer) is not aligned, but subsequent bursts are aligned to the address.	R/W
26	MB	Mixed Burst This bit is reserved and read-only (RO).	R
27	TXPR	Transmit Priority This bit is reserved and read-only (RO).	R
29:28	PRWG[1:0]	Channel Priority Weights This field is reserved and read-only (RO).	R
30	—	This bit is read as 0. The write value should be 0.	R/W
31	RIB	Rebuild INCRx Burst This bit is reserved and read-only (RO).	R

27.3.92 Transmit_Poll_Demand : Transmit Poll Demand Register

Base address: GMAC = 0x8010_0000

Offset address: 0x1004

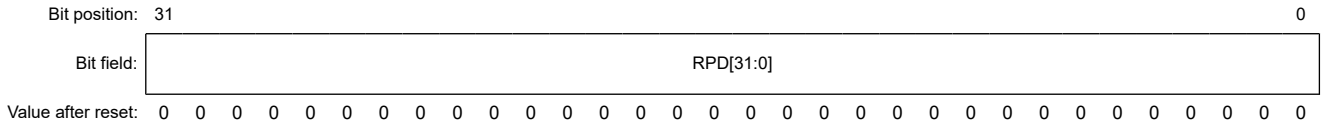


Bit	Symbol	Function	R/W
31:0	TPD[31:0]	Transmit Poll Demand When these bits are written with any value, the DMA reads the current descriptor pointed to by Current Host Transmit Descriptor Register (Current_Host_Transmit_Descriptor). If that descriptor is not available (owned by the host), the transmission returns to the Suspend state and bit TU[2] of Status Register (Status) is asserted. If the descriptor is available, the transmission resumes.	R/W

27.3.93 Receive_Poll_Demand : Receive Poll Demand Register

Base address: GMAC = 0x8010_0000

Offset address: 0x1008

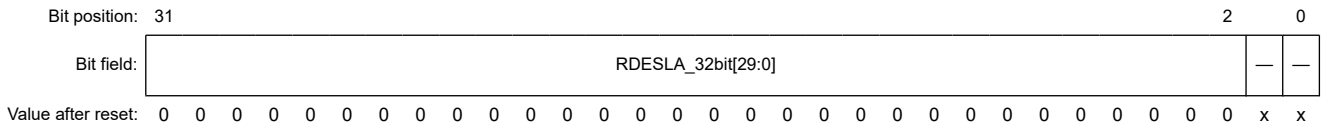


Bit	Symbol	Function	R/W
31:0	RPD[31:0]	Receive Poll Demand When these bits are written with any value, the DMA reads the current descriptor pointed to by Current Host Receive Descriptor Register (Current_Host_Receive_Descriptor). If that descriptor is not available (owned by the host), the reception returns to the Suspended state and bit RU[7] of Status Register (Status) is not asserted. If the descriptor is available, the RX DMA returns to the active state.	R/W

27.3.94 Receive_Descriptor_List_Address : Receive Descriptor List Address Register

Base address: GMAC = 0x8010_0000

Offset address: 0x100C



Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
31:2	RDESLA_32bit[29:0]	Start of Receive List This field contains the base address of the first descriptor in the receive descriptor list. The LSB bits [1:0] for 32-bit bus width are ignored and internally taken as all 0s by the DMA. Therefore, these LSB bits are read-only (RO).	R/W

27.3.95 Transmit_Descriptor_List_Address : Transmit Descriptor List Address Register

Base address: GMAC = 0x8010_0000

Offset address: 0x1010

Bit position:	31																										2	0
Bit field:	TDESLA_32bit[29:0]																										—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x	x

Bit	Symbol	Function	R/W
1:0	—	These bits are read as 0. The write value should be 0.	R/W
31:2	TDESLA_32bit[29:0]	Start of Transmit List This field contains the base address of the first descriptor in the transmit descriptor list. The LSB bits [1:0] for 32-bit bus width are ignored and are internally taken as all 0s by the DMA. Therefore, these LSB bits are read-only (RO).	R/W

27.3.96 Status : Status Register

Base address: GMAC = 0x8010_0000

Offset address: 0x1014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	GLPII	TTI	GPI	GMI	—	EB[2:0]			TS[2:0]			RS[2:0]			NIS
Value after reset:	x	0	0	0	0	x	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AIS	ERI	FBI	—	—	ETI	RWT	RPS	RU	RI	UNF	OVF	TJT	TU	TPS	TI
Value after reset:	0	0	0	x	x	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TI	Transmit Interrupt This bit indicates that the frame transmission is complete. When transmission is complete, bit OWN[31] of TDES0 is reset, and the specific frame status information is updated in the descriptor.	R/W
1	TPS	Transmit Process Stopped This bit is set when the transmission is stopped.	R/W
2	TU	Transmit Buffer Unavailable This bit indicates that the host owns the next descriptor in the transmit list and the DMA cannot acquire it. Transmission is suspended. Bits [22:20] specify the transmit process state transitions. To resume processing transmit descriptors, the host should change the ownership of the descriptor by setting TDES0[31] and then issue a Transmit Poll Demand command.	R/W
3	TJT	Transmit Jabber Timeout This bit indicates that the Transmit Jabber Timer expired, which happens when the frame size exceeds 2,048 (10,240 bytes when the Jumbo frame is enabled). When the Jabber Timeout occurs, the transmission process is aborted and placed in the Stopped state. This causes the Transmit Jabber Timeout TDES0[14] flag to assert.	R/W
4	OVF	Receive Overflow This bit indicates that the receive buffer had an overflow during frame reception. If the partial frame is transferred to the application, the overflow status is set in RDES0[11].	R/W
5	UNF	Transmit Underflow This bit indicates that the transmit buffer had an underflow during frame transmission. Transmission is suspended and an Underflow error TDES0[1] is set.	R/W

Bit	Symbol	Function	R/W
6	RI	Receive Interrupt This bit indicates that the frame reception is complete. When reception is complete, bit RDES1[31] (Disable Interrupt on Completion) is reset in the last descriptor, and the specific frame status information is updated in the descriptor. The reception remains in the Running state.	R/W
7	RU	Receive Buffer Unavailable This bit indicates that the host owns the next descriptor in the receive list and the DMA cannot acquire it. The receive process is suspended. To resume processing receive descriptors, the host should change the ownership of the descriptor and issue a Receive Poll Demand command. If no Receive Poll Demand is issued, the receive process resumes when the next recognized incoming frame is received. This bit is set only when the previous receive descriptor is owned by the DMA.	R/W
8	RPS	Receive Process Stopped This bit is asserted when the receive process enters the Stopped state.	R/W
9	RWT	Receive Watchdog Timeout When set, this bit indicates that the receive watchdog timer expired while receiving the current frame and the current frame is truncated after the watchdog timeout.	R/W
10	ETI	Early Transmit Interrupt This bit indicates that the frame to be transmitted is fully transferred to the MTL Transmit FIFO.	R/W
12:11	—	These bits are read as 0. The write value should be 0.	R/W
13	FBI	Fatal Bus Error Interrupt This bit indicates that a bus error occurred, as described in bits [25:23]. When this bit is set, the corresponding DMA engine disables all of its bus accesses.	R/W
14	ERI	Early Receive Interrupt This bit indicates that the DMA filled the first data buffer of the packet. This bit is cleared when the software writes 1 to this bit or bit RI[6] of this register is set (whichever occurs earlier).	R/W
15	AIS	Abnormal Interrupt Summary Abnormal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in Interrupt Enable Register (Interrupt_Enable): ● Interrupt_Enable [1]: Transmit Process Stopped ● Interrupt_Enable [3]: Transmit Jabber Timeout ● Interrupt_Enable [4]: Receive FIFO Overflow ● Interrupt_Enable [5]: Transmit Underflow ● Interrupt_Enable [7]: Receive Buffer Unavailable ● Interrupt_Enable [8]: Receive Process Stopped ● Interrupt_Enable [9]: Receive Watchdog Timeout ● Interrupt_Enable [10]: Early Transmit Interrupt ● Interrupt_Enable [13]: Fatal Bus Error Only unmasked bits affect the Abnormal Interrupt Summary bit. This is a sticky bit and must be cleared each time a corresponding bit, which causes AIS to be set, is cleared.	R/W
16	NIS	Normal Interrupt Summary Normal Interrupt Summary bit value is the logical OR of the following bits when the corresponding interrupt bits are enabled in Interrupt Enable Register (Interrupt_Enable): ● Interrupt_Enable [0]: Transmit Interrupt ● Interrupt_Enable [2]: Transmit Buffer Unavailable ● Interrupt_Enable [6]: Receive Interrupt ● Interrupt_Enable [14]: Early Receive Interrupt Only unmasked bits (interrupts for which interrupt enable is set in Interrupt Enable Register (Interrupt_Enable)) affect the Normal Interrupt Summary bit. This is a sticky bit and must be cleared (by writing 1 to this bit) each time a corresponding bit, which causes NIS to be set, is cleared.	R/W

Bit	Symbol	Function	R/W
19:17	RS[2:0]	Received Process State This field indicates the Receive DMA FSM state. This field does not generate an interrupt. 0 0 0: Stopped: Reset or Stop Receive Command issued 0 0 1: Running: Fetching Receive Transfer Descriptor 0 1 0: Reserved for future use 0 1 1: Running: Waiting for receive packet 1 0 0: Suspended: Receive Descriptor Unavailable 1 0 1: Running: Closing Receive Descriptor 1 1 0: TIME_STAMP write state 1 1 1: Running: Transferring the receive packet data from receive buffer to host memory	R
22:20	TS[2:0]	Transmit Process State This field indicates the Transmit DMA FSM state. This field does not generate an interrupt. 0 0 0: Stopped: Reset or Stop Transmit Command issued 0 0 1: Running: Fetching Transmit Transfer Descriptor 0 1 0: Running: Waiting for status 0 1 1: Running: Reading Data from host memory buffer and queuing it to transmit buffer (Tx FIFO) 1 0 0: TIME_STAMP write state 1 0 1: Reserved for future use 1 1 0: Suspended: Transmit Descriptor Unavailable or Transmit Buffer Underflow 1 1 1: Running: Closing Transmit Descriptor	R
25:23	EB[2:0]	Error Bits This field indicates the type of error that caused a bus error, for example, error response on the AXI interface. This field is valid only when bit FBI[13] is set. This field does not generate an interrupt. 0 0 0: Error during RX DMA Write Data Transfer 0 0 1: Reserved 0 1 0: Reserved 0 1 1: Error during TX DMA Read Data Transfer 1 0 0: Error during RX DMA Descriptor Write Access 1 0 1: Error during TX DMA Descriptor Write Access 1 1 0: Error during RX DMA Descriptor Read Access 1 1 1: Error during TX DMA Descriptor Read Access	R
26	—	This bit is read as 0. The write value should be 0.	R/W
27	GMI	GMAC MMC Interrupt This bit reflects an interrupt event in the MMC module of the GMAC. The software must read the corresponding registers in the GMAC to get the exact cause of interrupt and clear the source of interrupt to make this bit as 0b. The interrupt signal from the GMAC subsystem (GMAC_SBD) is high when this bit is high.	R
28	GPI	GMAC PMT Interrupt This bit indicates an interrupt event in the PMT module of the GMAC. The software must read the PMT Control and Status Register in the GMAC to get the exact cause of interrupt and clear its source to reset this bit to 0b. The interrupt signal from the GMAC subsystem (GMAC_SBD) is high when this bit is high. The GPI and GMAC_PMT interrupts are generated in different clock domains.	R
29	TTI	Timestamp Trigger Interrupt This bit indicates an interrupt event in the Timestamp Generator block of GMAC. Software must read the corresponding registers in the GMAC to get the exact cause of interrupt and clear its source to reset this bit to 0b. When this bit is high, the interrupt signal from the GMAC subsystem (GMAC_SBD) is high.	R
30	GLPII	GMAC LPI Interrupt This bit indicates an interrupt event in the LPI logic of the GMAC. To reset this bit to 0b, software must read the corresponding registers in the GMAC to get the exact cause of the interrupt and clear its source. When this bit is high, the interrupt signal from the GMAC (GMAC_SBD) is high.	R
31	—	This bit is read as 0. The write value should be 0.	R/W

27.3.97 Operation_Mode : Operation Mode Register

Base address: GMAC = 0x8010_0000

Offset address: 0x1018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	DT	RSF	—	—	—	TSF	FTF	—	—	—	TTC[2:0]
Value after reset:	x	x	x	x	x	0	0	x	x	0	0	0	x	x	x	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TTC[2:0]		ST	RFD[1:0]		RFA[1:0]		EFC	FEF	FUF	DGF	RTC[1:0]		OSF	SR	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	x

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	SR	Start or Stop Receive When this bit is set, the Receive process is placed in the Running state. The DMA attempts to acquire the descriptor from the Receive list and processes the incoming frames. The descriptor acquisition is attempted from the current position in the list, which is the address set by Receive Descriptor List Address Register (Receive_Descriptor_List_Address) or the position retained when the Receive process was previously stopped. If the DMA does not own the descriptor, reception is suspended and bit [7] (Receive Buffer Unavailable) of Status Register (Status) is set. The Start Receive command is effective only when the reception has stopped. If the command is issued before setting Receive Descriptor List Address Register (Receive_Descriptor_List_Address), the DMA behavior is unpredictable. When this bit is cleared, the RX DMA operation is stopped after the transfer of the current frame. The next descriptor position in the receive list is saved and becomes the current position after the receive process is restarted. The Stop Receive command is effective only when the receive process is in either the Running (waiting for receive packet) or in the Suspended state.	R/W
2	OSF	Operate on Second Frame When this bit is set, it instructs the DMA to process the second frame of the transmit data even before the status for the first frame is obtained.	R/W
4:3	RTC[1:0]	Receive Threshold Control These two bits control the threshold level of the MTL Receive FIFO. Transfer (request) to DMA starts when the frame size within the MTL Receive FIFO is larger than the threshold. In addition, full frames with length less than the threshold are transferred automatically. These bits are valid only when the RSF bit is 0, and are ignored when the RSF bit is set to 1. 0 0: 64 0 1: 32 1 0: 96 1 1: 128	R/W
5	DGF	Drop Giant Frames When set, the GMAC drops the received giant frames in the RX FIFO, that is, frames that are larger than the computed giant frame limit. When reset, the GMAC does not drop the giant frames in the RX FIFO.	R/W
6	FUF	Forward Undersized Good Frames When set, the RX FIFO forwards Undersized frames (frames with no error and length less than 64 bytes) including pad-bytes and CRC. When reset, the RX FIFO drops all frames of less than 64 bytes, unless a frame is already transferred because of the lower value of receive threshold, for example, RTC = 01b.	R/W

Bit	Symbol	Function	R/W
7	FEF	Forward Error Frames When this bit is reset, the RX FIFO drops frames with error status (CRC error, collision error, GMII_ER, giant frame, watchdog timeout, or overflow). However, if the start byte (write) pointer of a frame is already transferred to the read controller side (in Threshold mode), then the frame is not dropped. When the FEF bit is set, all frames except runt error frames are forwarded to the DMA. If bit RSF[25] is set and the RX FIFO overflows when a partial frame is written, then the frame is dropped irrespective of the FEF bit setting. However, if the bit RSF[25] is reset and the RX FIFO overflows when a partial frame is written, then a partial frame may be forwarded to the DMA.	R/W
8	EFC	Enable HW Flow Control When this bit is set, the flow control signal operation based on the fill-level of RX FIFO is enabled. When reset, the flow control operation is disabled.	R/W
10:9	RFA[1:0]	Threshold for Activating Flow Control (in half-duplex and full-duplex) These bits control the threshold (fill level of RX FIFO) at which the flow control is activated. 0 0: Full minus 1 KB, that is, FULL - 1 KB 0 1: Full minus 2 KB, that is, FULL - 2 KB 1 0: Full minus 3 KB, that is, FULL - 3 KB 1 1: Reserved These values are applicable only when bit EFC[8] is set high.	R/W
12:11	RFD[1:0]	Threshold for Deactivating Flow Control (in half-duplex and full-duplex) These bits control the threshold (fill-level of RX FIFO) at which the flow control is deasserted after activation. The deassertion is effective only after flow control is asserted. For proper flow control, the value programmed in the RFD field should be equal to or more than the value programmed in the RFA field. 0 0: Full minus 1 KB, that is, FULL - 1 KB 0 1: Full minus 2 KB, that is, FULL - 2 KB 1 0: Full minus 3 KB, that is, FULL - 3 KB 1 1: Full minus 4 KB, that is, FULL - 4 KB	R/W
13	ST	Start or Stop Transmission Command When this bit is set, transmission is placed in the Running state, and the DMA checks the Transmit List at the current position for a frame to be transmitted. Descriptor acquisition is attempted either from the current position in the list, which is the Transmit List Base Address set by Transmit Descriptor List Address Register (Transmit_Descriptor_List_Address), or from the position retained when transmission was stopped previously. If the DMA does not own the current descriptor, transmission enters the Suspended state and bit [2] (Transmit Buffer Unavailable) of Status Register (Status) is set. The Start Transmission command is effective only when transmission is stopped. If the command is issued before setting Transmit Descriptor List Address Register (Transmit_Descriptor_List_Address), then the DMA behavior is unpredictable. When this bit is reset, the transmission process is placed in the Stopped state after completing the transmission of the current frame. The next descriptor position in the transmit list is saved, and it becomes the current position when transmission is restarted. To change the list address, program the Transmit Descriptor List Address Register (Transmit_Descriptor_List_Address) with a new value when this bit is reset. The new value is considered when this bit is set again. The Stop Transmission command is effective only when the transmission of the current frame is complete or the transmission is in the Suspended state.	R/W
16:14	TTC[2:0]	Transmit Threshold Control These bits control the threshold level of the MTL Transmit FIFO. Transmission starts when the frame size within the MTL transmit FIFO is larger than the threshold. In addition, full frames with a length less than the threshold are also transmitted. These bits are used only when bit TSF[21] is reset. 0 0 0: 64 0 0 1: 128 0 1 0: 192 0 1 1: 256 1 0 0: 40 1 0 1: 32 1 1 0: 24 1 1 1: 16	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
20	FTF	Flush Transmit FIFO When this bit is set, the transmit FIFO controller logic is reset to its default values and thus all data in the TX FIFO is lost or flushed. This bit is cleared internally when the flushing operation is completed. The Operation Mode register should not be written to until this bit is cleared. The data which is already accepted by the GMAC transmitter is not flushed. It is scheduled for transmission and results in underflow and runt frame transmission. The flush operation is complete only when the TX FIFO is emptied of its contents and all the pending Transmit status of the transmitted frames are accepted by the host. To complete this flush operation, the PHY transmit clock is required to be active.	R/W
21	TSF	Transmit Store and Forward When this bit is set, transmission starts when a full frame resides in the MTL Transmit FIFO. When this bit is set, the TTC values specified in bits [16:14] are ignored. This bit should be changed only when the transmission is stopped.	R/W
24:22	—	These bits are read as 0. The write value should be 0.	R/W
25	RSF	Receive Store and Forward When this bit is set, the MTL reads a frame from the RX FIFO only after the complete frame has been written to it, ignoring the RTC bits. When this bit is reset, the RX FIFO operates in the cut-through mode, subject to the threshold specified by the RTC bits.	R/W
26	DT	Disable Dropping of TCP/IP Checksum Error Frames When this bit is set, the GMAC does not drop the frames which only have errors detected by the Receive Checksum Offload engine. Such frames do not have any errors (including FCS error) in the Ethernet frame received by the GMAC but have errors only in the encapsulated payload. When this bit is reset, all error frames are dropped if the FEF bit is reset.	R/W
31:27	—	These bits are read as 0. The write value should be 0.	R/W

27.3.98 Interrupt_Enable : Interrupt Enable Register

Base address: GMAC = 0x8010_0000

Offset address: 0x101C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	NIE
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AIE	ERE	FBE	—	—	ETE	RWE	RSE	RUE	RIE	UNE	OVE	TJE	TUE	TSE	TIE
Value after reset:	0	0	0	x	x	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TIE	Transmit Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (bit [16]), the Transmit Interrupt is enabled. When this bit is reset, the Transmit Interrupt is disabled.	R/W
1	TSE	Transmit Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (bit [15]), the Transmission Stopped Interrupt is enabled. When this bit is reset, the Transmission Stopped Interrupt is disabled.	R/W
2	TUE	Transmit Buffer Unavailable Enable When this bit is set with Normal Interrupt Summary Enable (bit [16]), the Transmit Buffer Unavailable Interrupt is enabled. When this bit is reset, the Transmit Buffer Unavailable Interrupt is disabled.	R/W
3	TJE	Transmit Jabber Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (bit [15]), the Transmit Jabber Timeout Interrupt is enabled. When this bit is reset, the Transmit Jabber Timeout Interrupt is disabled.	R/W

Bit	Symbol	Function	R/W
4	OVE	Overflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (bit [15]), the Receive Overflow Interrupt is enabled. When this bit is reset, the Overflow Interrupt is disabled.	R/W
5	UNE	Underflow Interrupt Enable When this bit is set with Abnormal Interrupt Summary Enable (bit [15]), the Transmit Underflow Interrupt is enabled. When this bit is reset, the Underflow Interrupt is disabled.	R/W
6	RIE	Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (bit [16]), the Receive Interrupt is enabled. When this bit is reset, the Receive Interrupt is disabled.	R/W
7	RUE	Receive Buffer Unavailable Enable When this bit is set with Abnormal Interrupt Summary Enable (bit [15]), the Receive Buffer Unavailable Interrupt is enabled. When this bit is reset, the Receive Buffer Unavailable Interrupt is disabled.	R/W
8	RSE	Receive Stopped Enable When this bit is set with Abnormal Interrupt Summary Enable (bit [15]), the Receive Stopped Interrupt is enabled. When this bit is reset, the Receive Stopped Interrupt is disabled.	R/W
9	RWE	Receive Watchdog Timeout Enable When this bit is set with Abnormal Interrupt Summary Enable (bit [15]), the Receive Watchdog Timeout Interrupt is enabled. When this bit is reset, the Receive Watchdog Timeout Interrupt is disabled.	R/W
10	ETE	Early Transmit Interrupt Enable When this bit is set with an Abnormal Interrupt Summary Enable (bit [15]), the Early Transmit Interrupt is enabled. When this bit is reset, the Early Transmit Interrupt is disabled.	R/W
12:11	—	These bits are read as 0. The write value should be 0.	R/W
13	FBE	Fatal Bus Error Enable When this bit is set with Abnormal Interrupt Summary Enable (bit [15]), the Fatal Bus Error Interrupt is enabled. When this bit is reset, the Fatal Bus Error Enable Interrupt is disabled.	R/W
14	ERE	Early Receive Interrupt Enable When this bit is set with Normal Interrupt Summary Enable (bit [16]), the Early Receive Interrupt is enabled. When this bit is reset, the Early Receive Interrupt is disabled.	R/W
15	AIE	Abnormal Interrupt Summary Enable When this bit is set, abnormal interrupt summary is enabled. When this bit is reset, the abnormal interrupt summary is disabled. This bit enables the following interrupts in Status Register: ● Status [1]: Transmit Process Stopped ● Status [3]: Transmit Jabber Timeout ● Status [4]: Receive Overflow ● Status [5]: Transmit Underflow ● Status [7]: Receive Buffer Unavailable ● Status [8]: Receive Process Stopped ● Status [9]: Receive Watchdog Timeout ● Status [10]: Early Transmit Interrupt ● Status [13]: Fatal Bus Error	R/W
16	NIE	Normal Interrupt Summary Enable When this bit is set, normal interrupt summary is enabled. When this bit is reset, normal interrupt summary is disabled. This bit enables the following interrupts in Status Register): ● Status [0]: Transmit Interrupt ● Status [2]: Transmit Buffer Unavailable ● Status [6]: Receive Interrupt ● Status [14]: Early Receive Interrupt	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

27.3.99 Missed_Frame_And_Buffer_Overflow_Counter : Missed Frame and Buffer Overflow Counter Register

Base address: GMAC = 0x8010_0000

Offset address: 0x1020

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	OVFCNTOVF	OVFFRMCNT[10:0]											MISCNTOVF
Value after reset:	x	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MISFRMCNT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	MISFRMCNT[15:0]	Missed Frame Counter This field indicates the number of frames missed by the controller because of the host receive buffer being unavailable. This counter is incremented each time the DMA discards an incoming frame. The counter is cleared when this bit is read.	R
16	MISCNTOVF	Overflow Bit for Missed Frame Counter This bit is set every time Missed Frame Counter (bits [15:0]) overflows, that is, the DMA discards an incoming frame because of the Host Receive Buffer being unavailable with the missed frame counter at maximum value. In such a scenario, the missed frame counter is reset to all 0s and this bit indicates that the rollover occurred.	R
27:17	OVFFRMCNT[10:0]	Overflow Frame Counter This field indicates the number of frames missed by the application. This counter is incremented each time the MTL FIFO overflows. The counter is cleared when this bit is read.	R
28	OVFCNTOVF	Overflow Bit for FIFO Overflow Counter This bit is set every time the Overflow Frame Counter (bits [27:17]) overflows, that is, the RX FIFO overflows with the overflow frame counter at maximum value. In such a scenario, the overflow frame counter is reset to all 0s and this bit indicates that the rollover occurred.	R
31:29	—	These bits are read as 0.	R

27.3.100 Receive_Interrupt_Watchdog_Timer : Receive Interrupt Watchdog Timer Register

Base address: GMAC = 0x8010_0000

Offset address: 0x1024

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RIWT[7:0]							
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RIWT[7:0]	RI Watchdog Timer Count This bit indicates the number of system clock cycles multiplied by 256 for which the watchdog timer is set. The watchdog timer gets triggered with the programmed value after the RX DMA completes the transfer of a frame for which the RI status bit is not set because of the setting in the corresponding descriptor RDES1[31]. When the watchdog timer runs out, the RI bit is set and the timer is stopped. The watchdog timer is reset when the RI bit is set high because of automatic setting of RI as per RDES1[31] of any received frame.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

27.3.101 AXI_Bus_Mode : AXI Bus Mode Register

Base address: GMAC = 0x8010_0000

Offset address: 0x1028

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	EN_LPI	LPI_XIT_FRM	—	—	—	—	—	—	—	—	WR_OSR_LMT[1:0]		—	—	RD_OSR_LMT[1:0]	
Value after reset:	0	0	x	x	x	x	x	x	x	x	0	1	x	x	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	ONEKBBE	AXI_AAL	—	—	—	—	—	—	—	—	BLEN16	BLEN8	BLEN4	UNDEF
Value after reset:	x	x	0	0	x	x	x	x	x	x	x	x	0	0	0	1

Bit	Symbol	Function	R/W
0	UNDEF	AXI Undefined Burst Length This bit is read-only bit and indicates the complement (invert) value of bit FB[16] in Bus Mode Register (Bus_Mode[16]). ● When this bit is set to 1, the GMAC-AXI is allowed to perform any burst length equal to or below the maximum allowed burst length programmed in bits [3]. ● When this bit is set to 0, the GMAC-AXI is allowed to perform only fixed burst lengths as indicated by BLEN16, BLEN8, or BLEN4, or a burst length of 1.	R
1	BLEN4	AXI Burst Length 4 When this bit is set to 1, the GMAC-AXI is allowed to select a burst length of 4 on the AXI Master interface. Setting this bit has no effect when UNDEF is set to 1.	R/W
2	BLEN8	AXI Burst Length 8 When this bit is set to 1, the GMAC-AXI is allowed to select a burst length of 8 on the AXI Master interface. Setting this bit has no effect when UNDEF is set to 1.	R/W
3	BLEN16	AXI Burst Length 16 When this bit is set to 1 or UNDEF is set to 1, the GMAC-AXI is allowed to select a burst length of 16 on the AXI Master interface.	R/W
11:4	—	These bits are read as 0. The write value should be 0.	R/W
12	AXI_AAL	Address-Aligned Beats This bit is read-only bit and reflects bit AAL[25] of Bus Mode Register (Bus_Mode). When this bit is set to 1, the GMAC-AXI performs address-aligned burst transfers on both read and write channels.	R
13	ONEKBBE	1 KB Boundary Crossing Enable for the GMAC-AXI Master When set, the GMAC-AXI Master performs burst transfers that do not cross 1 KB boundary. When reset, the GMAC-AXI Master performs burst transfers that do not cross 4 KB boundary.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
17:16	RD_OSR_LMT[1:0]	AXI Maximum Read OutStanding Request Limit This value limits the maximum outstanding request on the AXI read interface. Maximum outstanding requests = RD_OSR_LMT + 1	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
21:20	WR_OSR_LMT[1:0]	AXI Maximum Write OutStanding Request Limit This value limits the maximum outstanding request on the AXI write interface. Maximum outstanding requests = WR_OSR_LMT + 1	R/W
29:22	—	These bits are read as 0. The write value should be 0.	R/W
30	LPI_XIT_FRM	Unlock on Magic Packet or Remote Wake-Up Frame When set to 1, this bit enables the GMAC-AXI to come out of the LPI mode only when the Magic Packet or Remote Wake-Up Packet is received. When set to 0, this bit enables the GMAC-AXI to come out of LPI mode when any frame is received.	R/W

Bit	Symbol	Function	R/W
31	EN_LPI	Enable Low Power Interface (LPI) When set to 1, this bit enables the LPI mode and accepts the LPI request from the AXI System Clock controller. When set to 0, this bit disables the LPI mode and always denies the LPI request from the AXI System Clock controller.	R/W

27.3.102 AXI_Status : AXI Status Register

Base address: GMAC = 0x8010_0000

Offset address: 0x102C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AXIRDSTS	AXWHSTS
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
0	AXWHSTS	AXI Master Write Channel When high, it indicates that the write channel of the AXI master is active and transferring data.	R
1	AXIRDSTS	AXI Master Read Channel Status When high, it indicates that the read channel of the AXI master is active and transferring data.	R
31:2	—	These bits are read as 0.	R

27.3.103 Current_Host_Transmit_Descriptor : Current Host Transmit Descriptor Register

Base address: GMAC = 0x8010_0000

Offset address: 0x1048

Bit position:	31	0
Bit field:	CURTDESAPTR[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	CURTDESAPTR[31:0]	Host Transmit Descriptor Address Pointer Cleared on reset. Pointer updated by the DMA during operation.	R

27.3.104 Current_Host_Receive_Descriptor : Current Host Receive Descriptor Register

Base address: GMAC = 0x8010_0000

Offset address: 0x104C

Bit position:	31	0
Bit field:	CURRDESAPTR[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	CURRDESAPTR[31:0]	Host Receive Descriptor Address Pointer Cleared on reset. Pointer updated by the DMA during operation.	R

27.3.105 Current_Host_Transmit_Buffer_Address : Current Host Transmit Buffer Address Register

Base address: GMAC = 0x8010_0000

Offset address: 0x1050

Bit position:	31																												0
Bit field:	CURTBUFAPTR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CURTBUFAPTR[31:0]	Host Transmit Buffer Address Pointer Cleared on reset. Pointer updated by the DMA during operation.	R

27.3.106 Current_Host_Receive_Buffer_Address : Current Host Receive Buffer Address Register

Base address: GMAC = 0x8010_0000

Offset address: 0x1054

Bit position:	31																												0
Bit field:	CURRBUFAPTR[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CURRBUFAPTR[31:0]	Host Receive Buffer Address Pointer Cleared on reset. Pointer updated by the DMA during operation.	R

27.3.107 HW_Feature : HW Feature Register

Base address: GMAC = 0x8010_0000

Offset address: 0x1058

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	ACTPHYIF[2:0]			SAVLANINS	FLEXIPPSEN	INTTSEN	ENHDESSEL	TXCHCNT[1:0]		RXCHCNT[1:0]		RXFIFOSIZE	RXTYP2COE	RXTYP1COE	TXCOESEL
Value after reset:	x	0	0	0	0	0	1	1	0	0	0	0	1	1	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AVSEL	EEESEL	TSVER2SEL	TSVER1SEL	MMCSEL	MGKSEL	RWKSEL	SMASEL	L3L4FLTREN	—	ADDMACADRSEL	HASHSEL	EXTHASHEN	HDSEL	GMIISEL	MIISEL
Value after reset:	0	1	1	0	1	1	1	1	0	0	1	1	1	1	1	1

Bit	Symbol	Function	R/W
0	MIISEL	10 or 100 Mbps support	R
1	GMIISEL	1000 Mbps support	R
2	HDSEL	Half-Duplex support	R

Bit	Symbol	Function	R/W
3	EXTHASHEN	Expanded DA Hash Filter	R
4	HASHSEL	HASH Filter	R
5	ADDMACADRSEL	Multiple MAC Address Registers	R
6	—	This bit is read as 0.	R
7	L3L4FLTREN	Layer 3 and Layer 4 Filter Feature	R
8	SMASEL	SMA (MDIO) Interface	R
9	RWKSEL	PMT Remote wakeup	R
10	MGKSEL	PMT Magic Packet	R
11	MMCSEL	RMON Module	R
12	TSVER1SEL	Only IEEE 1588-2002 Timestamp	R
13	TSVER2SEL	IEEE 1588-2008 Advanced Timestamp	R
14	EEESEL	Energy Efficient Ethernet	R
15	AVSEL	AV Feature	R
16	TXCOESEL	Checksum Offload in TX	R
17	RXTYP1COE	IP Checksum Offload (Type 1) in RX	R
18	RXTYP2COE	IP Checksum Offload (Type 2) in RX	R
19	RXFIFOSIZE	RX FIFO > 2,048 Bytes	R
21:20	RXCHCNT[1:0]	Number of additional RX channels	R
23:22	TXCHCNT[1:0]	Number of additional TX channels	R
24	ENHDESSEL	Enhanced Descriptor	R
25	INTTSEN	Timestamping with Internal System Time	R
26	FLEXIPPSEN	Flexible Pulse-Per-Second Output (GMAC: 0)	R
27	SAVLANINS	Source Address or VLAN Insertion	R
30:28	ACTPHYIF[2:0]	Active or Selected PHY interface 0 0 0: GMII or MII Others: Reserved	R
31	—	This bit is read as 0.	R

27.3.108 GMACTRGSEL : GMAC PTP Trigger Select Register

Base address: GMAC = 0x8010_0000

Offset address: 0x0400

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TRGSEL	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	TRGSEL	Select PTP Timestamp Trigger for GMAC IP 0: Select GMAC_PTPTRGn (n = 0, 1) trigger from ELC 1: Select external GMAC_PTPTRGn (n = 0, 1) trigger input pin	R/W
31:2	—	These bits are read as 0.	R

Note: This register can only be written when protection is unlocked by a specific sequence using the Ethernet Protect Register (PRCMD). For the protection unlock procedure, see section 26.2.1. PRCMD : Ethernet Protect Register. The special sequence is not necessary when reading the value of this register.

27.3.109 PTP Timer Pulse Control Registers

The registers control the PTP pulse generator. The PTP pulse generator operates based on the current time of the PTP timer and the register values and generates a pulse signal of any cycle from any start time. The pulse width can be set arbitrarily.

The register controls the pulse signal of GMAC_PTPOUTn (n = 0 to 3) for interrupt signals or external output pins.

27.3.109.1 SWTMENn : PTP Timer Pulse Output Enable n Register (n = 0 to 3)

Base address: GMAC_PTP = 0x8011_0400

Offset address: 0x400 + 0x100 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OUTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OUTEN	Enable GMAC_PTPOUTn Signal Output 0: Disabled 1: Enabled	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: This register should be set after setting completion of SWTMSTSEC, SWTMSTNS, SWTMPSEC, SWTMPNS, SWTMWTH, and SWTMMAXP registers. When this register is set to 1, changing SWTMSTSEC, SWTMSTNS, SWTMPSEC, SWTMPNS, SWTMWTH, and SWTMMAXP has no effect.

27.3.109.2 SWTMSTSECn : PTP Timer Pulse Start Second n Register (n = 0 to 3)

Base address: GMAC_PTP = 0x8011_0400

Offset address: 0x404 + 0x100 × n

Bit position:	31	0
Bit field:	STSEC[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	STSEC[31:0]	Start Time by Second	R/W

27.3.109.3 SWTMSTNSn : PTP Timer Pulse Start Nanosecond n Register (n = 0 to 3)

Base address: GMAC_PTP = 0x8011_0400

Offset address: 0x408 + 0x100 × n

Bit position:	31	0
Bit field:	STNS[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

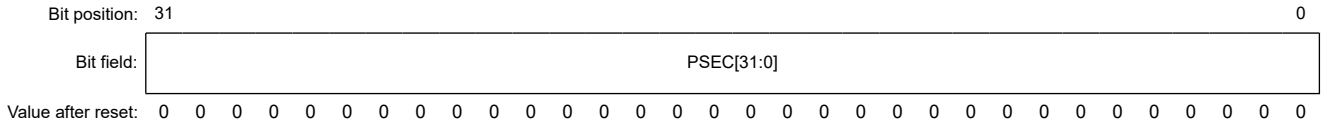
Bit	Symbol	Function	R/W
31:0	STNS[31:0]	Start Time by Nanosecond	R/W

Note: When SWTMMAXP is not 0, the value set in this register must be less than the value set in SWTMMAXP (SWTMSTNS < SWTMMAXP). When SWTMMAXP is 0, this register can be set to any value.

27.3.109.4 SWTMPSECn : PTP Timer Pulse Period Second n Register (n = 0 to 3)

Base address: GMAC_PTP = 0x8011_0400

Offset address: 0x40C + 0x100 × n

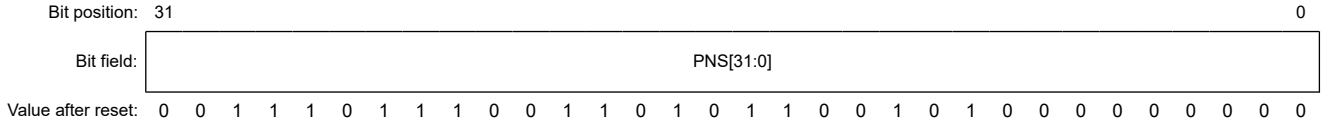


Bit	Symbol	Function	R/W
31:0	PSEC[31:0]	Period by Second	R/W

27.3.109.5 SWTMPNSn : PTP Timer Pulse Period Nanosecond n Register (n = 0 to 3)

Base address: GMAC_PTP = 0x8011_0400

Offset address: 0x410 + 0x100 × n



Bit	Symbol	Function	R/W
31:0	PNS[31:0]	Period by Nanosecond	R/W

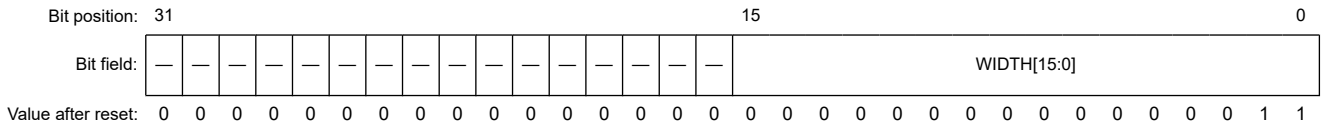
Note: The value set in this register must be larger than the value below.
In case PCLKM = 100 MHz, 18 × SSINC[7:0] bits in Sub_Second_Increment Register (144 ns when SSINC[7:0] = 8).
In case PCLKM = 75 MHz, 20 × SSINC[7:0] bits in Sub_Second_Increment Register (160 ns when SSINC[7:0] = 8).

Note: When SWTMMAXP is not 0, the value set in this register must be less than or equal to the value set in SWTMMAXP (SWTMPNS ≤ SWTMMAXP).

27.3.109.6 SWTMWTHn : PTP Timer Pulse Width n Register (n = 0 to 3)

Base address: GMAC_PTP = 0x8011_0400

Offset address: 0x414 + 0x100 × n



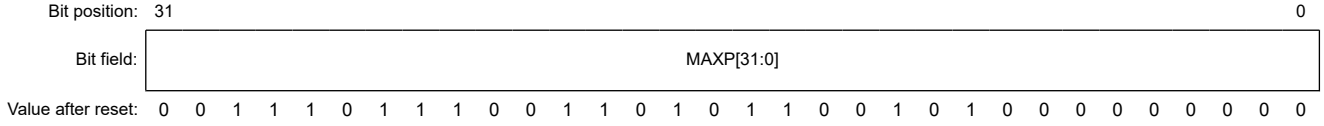
Bit	Symbol	Function	R/W
15:0	WIDTH[15:0]	Set the Pulse Width of GMAC_PTPOUTn in the cycle number of GMAC_PTP_REFCLK_I (8 ns). Output pulse width is the set value × 8 (ns). The GMAC_PTPOUTn is set to 0 when this register is 0x0000.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

Note: If GMAC_PTPOUT is used as pulse event, do not change the value of this register (use the initial value). When it is used as external signal, appropriate width must be specified.

27.3.109.7 SWTMMAXPn : PTP Timer Pulse Max Nanosecond n Register (n = 0 to 3)

Base address: GMAC_PTP = 0x8011_0400

Offset address: 0x418 + 0x100 × n



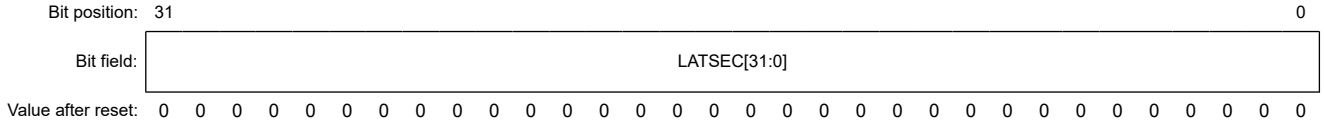
Bit	Symbol	Function	R/W
31:0	MAXP[31:0]	Sets the boundary value in nanoseconds to carry from the nanosecond field to the second field. This register must be set to 0x3B9ACA00 (= 10^9) when TSCTRLSSR bit in Timestamp_Control register is set to 1. This register must be set to 0x80000000 (= 2^{31}) when TSCTRLSSR bit in Timestamp_Control register is reset to 0.	R/W

Note: This register sets the boundary value in nanoseconds to carry from the nanosecond field to the second field. When the nanosecond field reaches the set value of this register, the second field is carried up. This register is normally used with the initial value of 0x3B9ACA00 (= 10^9) and TSCTRLSSR bit in Timestamp_Control register is set to 1. In this case, the nanosecond field counts from 0x0 to 0x3B9AC9FF (= 10^9 - 1) and then carries to the second field. If 0x80000000 (= 2^{31}) is set in this register and TSCTRLSSR bit in Timestamp_Control register is reset to 0, the boundary value for carrying to the second field is interpreted as 0x80000000 (= 2^{31}). In this case, the nanosecond field is fully counted from 0x0 to 0x7FFFFFFF (= 2^{31} - 1) and then carries to the second field.

27.3.109.8 SWTMLATSECn : PTP Timer Pulse Latch Second n Register (n = 0 to 3)

Base address: GMAC_PTP = 0x8011_0400

Offset address: 0x41C + 0x100 × n



Bit	Symbol	Function	R/W
31:0	LATSEC[31:0]	Latch the time of second count at the GMAC_PTPOUTn rising edge.	R/W

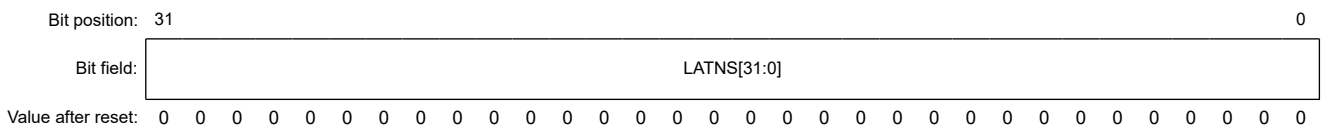
Note: This register is updated when the PTPOUT signal rises, and the updated value is expressed in the formula t_{START} + t_{PERIOD} × n (n = 0, 1, 2, …) in seconds of the rise target time. t_{START} = {SWTMSTSEC, SWTMSTNS}, t_{PERIOD} = {SWTMPSEC, SWTMPNS}

Note: Since SWTMLATSEC is a 32-bit register, it will be 0x00000000 when incremented in the state of 0xFFFFFFFF.

27.3.109.9 SWTMLATNSn : PTP Timer Pulse Latch Nanosecond n Register (n = 0 to 3)

Base address: GMAC_PTP = 0x8011_0400

Offset address: 0x420 + 0x100 × n



Bit	Symbol	Function	R/W
31:0	LATNS[31:0]	Latch the time of nanosecond count at the GMAC_PTPOUTn rising edge.	R/W

Note: This register is updated when the PTPOUT signal rises, and the updated value is expressed in the formula t_{START} + t_{PERIOD} × n (n = 0, 1, 2, …) in nanoseconds of the rise target time. t_{START} = {SWTMSTSEC, SWTMSTNS}, t_{PERIOD} = {SWTMPSEC, SWTMPNS}

27.4 Operation

This section provides the instructions for initializing the DMA or MAC registers in the proper sequence.

27.4.1 Initializing

The initializing sequence in this section is an example for preparation of system environments using GMAC under the configuration below.

GMAC configuration of this example:

GMAC is connected to external port1 through RGMII/RMII Converter.

The sequence for section 27.4.2. Initializing DMA and section 27.4.3. Initializing GMAC is followed after the sequence of this section.

27.4.2 Initializing DMA

The initialization sequence in this section is used for GMAC configurations.

Complete the following steps to initialize the DMA:

1. Provide a software reset. This resets all of the GMAC internal registers and logic. (Bus Mode Register (Bus_Mode) - bit 0).
2. Wait for the completion of the reset process (poll bit 0 of the Bus Mode Register (Bus_Mode), which is only cleared after the reset operation is completed).
3. Poll the bits of AXI Status Register (AXI_Status) to confirm that all previously initiated (before soft reset) or ongoing AXI transactions are complete.

   Note: If the application cannot poll the AXI Status Register (AXI_Status) after a soft reset (because of performance reasons), then it is recommended that you continue with the next steps and check this register again (as mentioned in Step 11) before triggering the DMA operations.
4. Program the following fields to initialize the Bus Mode Register by setting values in Bus Mode Register:
   - AAL
   - Fixed burst or undefined burst
   - Burst length values and burst mode values.
   - Descriptor length
5. Program the AXI Interface options in AXI Bus Mode Register (AXI_Bus_Mode).
6. Create a proper descriptor chain for transmit and receive. In addition, ensure that the receive descriptors are owned by DMA (bit [31] of descriptor should be set). When OSF mode is used, at least two descriptors are required.
7. Make sure that your software creates three or more different transmit or receive descriptors in the chain before reusing any of the descriptors.
8. Initialize receive and transmit descriptor list address with the base address of the transmit and receive descriptor (Receive Descriptor List Address Register (Receive_Descriptor_List_Address) and Transmit Descriptor List Address Register (Transmit_Descriptor_List_Address) respectively).
9. Program the following fields to initialize the mode of operation in Operation Mode Register (Operation_Mode)
   - Receive and Transmit Store and Forward
   - Receive and Transmit Threshold Control (RTC and TTC)
   - Hardware Flow Control enable
   - Flow Control Activation and Deactivation thresholds for MTL Receive and Transmit FIFO (RFA and RFD)
   - Error frame and undersized good frame forwarding enable

- OSF mode

10. Enable the interrupts by programming Interrupt Enable Register (Interrupt_Enable).

Note: Perform Step 11 only if you did not perform Step 3.

11. Read AXI Status Register (AXI_Status) to confirm that all previous AXI transactions are complete.

Note: If any previous transaction is still in progress when you read the AXI Status Register, it is strongly recommended to check the slave components addressed by the AXI master interface of GMAC.

12. Start the Receive and Transmit DMA by setting SR (bit 1) and ST (bit 13) of the Operation Mode Register (Operation_Mode).

27.4.3 Initializing GMAC

The following GMAC Initialization operations can be performed after DMA initialization. If the GMAC initialization is done before the DMA is set up, then enable the GMAC receiver (last step in following sequence) only after the DMA is active. Otherwise, received frames fills the RX FIFO and overflow.

1. Program the GMII Address Register (GMII_Address) for controlling the management cycles for external PHY. For example, Physical Layer Address PA (bits [15:11]). In addition, set bit [0] (GMII Busy) for writing into PHY and reading from PHY.
2. Read the 16-bit data of GMII Data Register (GMII_Data) from the PHY for link up, speed of operation, and mode of operation, by specifying the appropriate address value in bits [15:11] of GMII Address Register (GMII_Address).
3. Provide the MAC address; MAC Address0 High Register (MAR0_H) and MAC Address0 Low Register (MAR0_L).
4. Program the following fields to set the appropriate filters for the incoming frames in MAC Frame Filter Register:
   - Receive All
   - Promiscuous mode
   - Hash or Perfect Filter
   - Unicast, multicast, broadcast, and control frames filter settings
5. Program the following fields for proper flow control in Flow Control Register (Flow_Control):
   - Pause time and other Pause frame control bits
   - Receive and Transmit Flow control bits
   - Flow Control Busy/Backpressure Activate
6. Program the Interrupt Mask register bits, as required.
7. Program the appropriate fields in MAC Configuration Register (MAC_Configuration). For example, Inter frame gap while transmission and jabber disable. Based on the Auto negotiation you can set the Duplex mode (bit [11]) or port select (bit [15]).
8. Set bit [3] (TE) and bit [2] (RE) in MAC Configuration Register (MAC_Configuration).

Note: Do not change the configuration (such as duplex mode, speed, port, or loopback) when the GMAC is actively transmitting or receiving. Software should change these parameters only when the GMAC transmitter and receiver are not active.

27.4.4 Performing Normal Receive and Transmit Operation

For normal operation, complete the following steps:

1. For normal transmit and receive interrupts, read the interrupt status. Then, poll the descriptors, reading the status of the descriptor owned by the Host (either transmit or receive).
2. Set appropriate values for the descriptors, ensuring that transmit and receive descriptors are owned by the DMA to resume the transmission and reception of data.
3. If the descriptors are not owned by the DMA (or no descriptor is available), the DMA goes into SUSPEND state. The transmission or reception can be resumed by freeing the descriptors and issuing a poll demand by writing 0 into the Transmit Poll Demand Register (Transmit_Poll_Demand) and Receive Poll Demand Register (Receive_Poll_Demand).

4. The values of the current host transmitter or receiver descriptor address pointer can be read for the debug process (Current Host Transmit Descriptor Register (Current_Host_transmit_Descriptor) and Current Host Receive Descriptor Register (Current_Host_Receive_Descriptor)).
5. The values of the current host transmit buffer address pointer and receive buffer address pointer can be read for the debug process (Current Host Transmit Buffer Address Register (Current_Host_Transmit_Buffer_Address) and Current Host Receive Buffer Address Register (Current_Host_Receive_Buffer_Address)).

27.4.5 Stopping and Starting Transmission

Complete the following steps to pause the transmission for some time:

1. Disable the Transmit DMA (if applicable) by clearing bit [13] (ST) of Operation Mode Register (Operation_Mode).
2. Wait for any previous frame transmissions to complete. You can check this by reading the appropriate bits of Debug Register (Debug).
3. Disable the GMAC transmitter and GMAC receiver by clearing bit [3] (TE) and bit [2] (RE) in MAC Configuration Register (MAC_Configuration).
4. Disable the Receive DMA (if applicable), after making sure that the data in the RX FIFO is transferred to the system memory (by reading Debug Register (Debug)).
5. Make sure that both TX FIFO and RX FIFO are empty.
6. To restart the operation, first start the DMAs, and then enable the GMAC transmitter and receiver.

27.4.6 Programming Guidelines for GMII Link Transitions

27.4.6.1 Transmit and Receive Clocks are Running when the Link is Down

Complete the following steps when the link is down but the transmit and receive clocks are running:

1. Disable the Transmit DMA (if applicable), by clearing bit [13] (ST) of Operation Mode Register (Operation_Mode).
2. Disable the GMAC receiver by clearing bit [2] (RE) of MAC Configuration Register (MAC_Configuration).
3. Wait for any previous frame transmissions to complete from the TX FIFO. You can do this by reading the appropriate bits of Debug Register (Debug).
   -or-
   Flush the TXFIFO for faster empty operation.
4. Disable the GMAC transmitter by clearing bit [3] (TE) in MAC Configuration Register (MAC_Configuration).
5. After the link is up, read the PHY registers for the latest configuration and program the GMAC registers accordingly.
6. Restart the operation by starting the TX DMA, and then enabling the GMAC transmitter and receiver.

You do not need to disable the RX DMA. When the receiver is disabled, the FIFO does not get any data in the RX FIFO.

27.4.6.2 Transmit and Receive Clocks are Stopped when the Link is Down

Complete the following steps when the link is down and the transmit and receive clocks are stopped:

1. Wait until the link is up and the transmit and receive clocks are active.
   When the transmit and receive clocks are stopped, disabling the transmit or receive operations does not have any effect. Therefore, the software must wait until the link is up again.
2. Disable the Transmit DMA (if applicable), by clearing bit [13] (ST) of the Operation Mode Register (Operation_Mode)
3. Disable the GMAC receiver by clearing bit [2] (RE) of MAC Configuration Register (MAC_Configuration).
4. Wait for any previous frame transmissions to complete from the TX FIFO. You can do this by reading the appropriate bits of Debug Register (Debug).
   -or-
   Flush the TX FIFO for faster empty operation.
5. Disable the GMAC transmitter by clearing bit [3] (TE) in MAC Configuration (MAC_Configuration).
6. After the link is up, read the PHY registers for the latest configuration and program the GMAC registers accordingly.

7. Restart the operation by starting the TX DMA, and then enabling the GMAC transmitter and receiver.

27.4.7 Programming Guidelines for IEEE 1588 Timestamping

27.4.7.1 Initialization Guideline for System Time Generation

You can enable the timestamp feature by setting bit 0 of the Timestamp control register. However, it is essential that the timestamp counter should be initialized after this bit is set. Complete the following steps during GMAC initialization:

1. Mask the Timestamp Trigger interrupt by setting bit 9 of Interrupt Mask Register (Interrupt_Mask).
2. Program bit 0 in Timestamp Control Register (Timestamp_Control) to enable timestamping.
3. Program Sub Second Increment Register (Sub_Second_Increment) based on the PTP clock frequency.
4. If you are using the Fine Correction approach, program Timestamp Addend Register (Timestamp_Addend) and set bit 5 of Timestamp Control Register (Timestamp_Control).
5. Poll the Timestamp Control register until bit 5 is cleared.
6. Program bit 1 of Timestamp Control Register (Timestamp_Control) to select the Fine Update method (if required).
7. Program System Time – Seconds Update Register (System_Time_Seconds_Update) and System Time – Nanoseconds Update Register (System_Time_Nanoseconds_Update) with the appropriate time value.
8. Set bit 2 in Timestamp Control Register (Timestamp_Control). The timestamp counter starts operation as soon as it is initialized with the value written in the Timestamp Update registers.
9. Enable the GMAC receiver and transmitter for proper timestamping.

Note: If timestamp operation is disabled by clearing bit 0 of Timestamp Control Register (Timestamp_Control), you need to repeat all these steps to restart the timestamp operation.

27.4.7.2 System Time Correction

To synchronize or update the system time in one process (coarse correction method), complete the following steps:

1. Set the offset (positive or negative) in the Timestamp Update registers (System Time – Seconds Update Register (System_Time_Seconds_Update) and System Time – Nanoseconds Update Register (System_Time_Nanoseconds_Update)).
2. Set bit 3 (TSUPDT) of Timestamp Control Register (Timestamp_Control). The value in the Timestamp Update registers is added to or subtracted from the system time when the TSUPDT bit is cleared.

To synchronize or update the system time to reduce system time jitter (fine correction method), complete the following steps:

1. With the help of the algorithm explained in section 27.4.8. System Time Register Module, calculate the rate by which you want to make the system time increments slower or faster.
2. Update Timestamp Addend Register (Timestamp_Addend) with the new value and set bit 5 of Timestamp Control Register (Timestamp_Control).
3. Wait for the time for which you want the new value of the Addend register to be active. You can do this by enabling the Timestamp Trigger interrupt after the system time reaches the target value.
4. Program the required target time in Target Time Seconds Register (Target_Time_Seconds) and Target Time Nanoseconds Register (Target_Time_Nanoseconds).
5. Unmask the Timestamp interrupt by clearing bit 9 of Interrupt Mask Register (Interrupt_Mask).
6. Set bit 4 in Timestamp Control Register (Timestamp_Control).
7. When this trigger causes an interrupt, read Interrupt Status Register (Interrupt_Status).
8. Reprogram Timestamp Addend Register (Timestamp_Addend) with the old value and set bit 5 again.

27.4.8 System Time Register Module

The 64-bit time is maintained in this module and updated using the input reference clock (GMAC_PTP_REFCLK_I). This time is the source for taking snapshots (timestamps) of Ethernet frames being transmitted or received at the GMII.

The System Time counter can be initialized or corrected using the coarse correction method. In this method, the initial value or the offset value is written to the Timestamp Update register (see section 27.5.8. IEEE 1588 Timestamp Registers). For initialization, the System Time counter is written with the value in the Timestamp Update registers, while for system time correction, the offset value is added to or subtracted from the system time.

In the fine correction method, a slave clock's (GMAC_PTP_REFCLK_I) frequency drift with respect to the master clock (as defined in IEEE 1588) is corrected over a period of time instead of in one clock, as in coarse correction. This helps maintain linear time and does not introduce drastic changes (or a large jitter) in the reference time between PTP Sync message intervals. In this method, an accumulator sums up the contents of the Addend register. The arithmetic carry that the accumulator generates is used as a pulse to increment the system time counter. The accumulator and the addend are 32-bit registers. Here, the accumulator acts as a high precision frequency multiplier or divider.

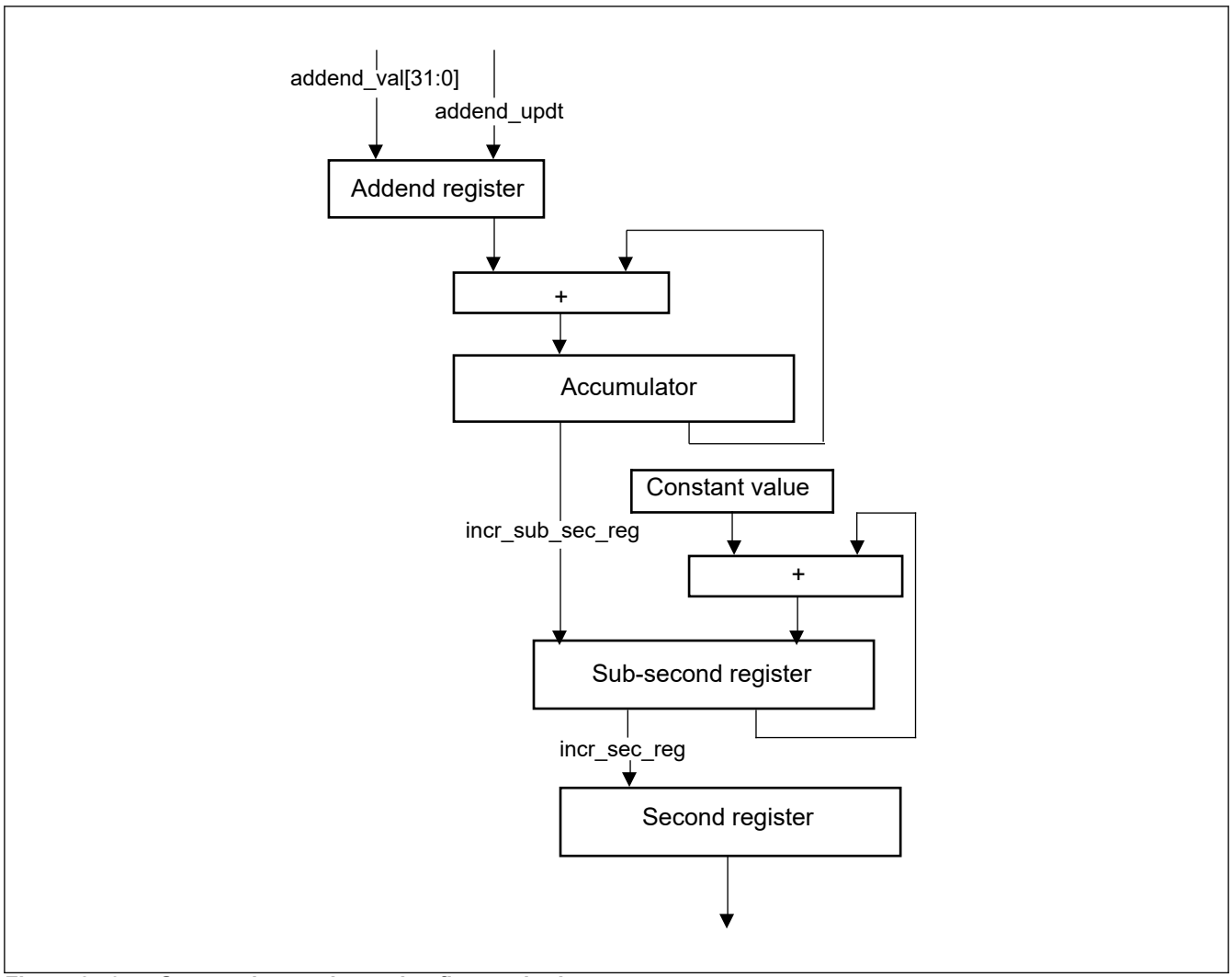


Figure 27.3 System time update using fine method

The System Time Update logic requires a 50 MHz clock frequency to achieve 20 ns accuracy. The frequency division is the ratio of the reference clock frequency to the required clock frequency. Hence, if the reference clock (GMAC_PTP_REFCLK_I) is, for example, 66 MHz, this ratio is calculated as 66 MHz / 50 MHz = 1.32. Hence, the default addend value to be set in the register is 232 / 1.32, 0xC1F07C1F.

If the reference clock drifts lower, to 65 MHz for example, the ratio is 65 / 50, or 1.3 and the value to set in the addend register is 232 / 1.30, or 0xC4EC4EC4. If the clock drifts higher, to 67 MHz for example, the addend register must be set to 0xBF0B7672. When the clock drift is nil, the default addend value of 0xC1F07C1F (232 / 1.32) must be programmed.

In Figure 27.3, the constant value used to accumulate the sub-second register is decimal 43, which achieves an accuracy of 20 ns in the system time (in other words, it is incremented in 20 ns steps). The optional System Time module is unavailable when External Time Update is enabled. Two different methods are used to update the System Time register, depending on which configuration you choose.

The software must calculate the drift in frequency based on the Sync messages and update the Addend register accordingly.

Initially, the slave clock is set with FreqCompensationValue0 in the Addend register. This value is as follows:

$$\mathrm{FreqCompensationValue0} = \frac{2^{32}}{\mathrm{FreqDivisionRatio}}$$

If MasterToSlaveDelay is initially assumed to be the same for consecutive Sync messages, the algorithm described below must be applied. After a few Sync cycles, frequency lock occurs. The slave clock can then determine a precise MasterToSlaveDelay value and re-synchronize with the master using the new value.

The algorithm is as follows:

- At time MasterSyncTime(n), the master sends the slave clock a Sync message. The slave receives this message when its local clock is SlaveClockTime(n) and computes MasterClockTime(n) as:
  $\mathrm{MasterClockTime(n)} = \mathrm{MasterSyncTime(n)} + \mathrm{MasterToSlaveDelay(n)}$
- The master clock count for current Sync cycle, MasterClockCount(n) is given by:
  $\mathrm{MasterClockCount(n)} = \mathrm{MasterClockTime(n)} - \mathrm{MasterClockTime(n-1)}$
  $(\mathrm{assumingthatMasterToSlaveDelayisthesameforSynccyclesnandn-1})$
- The slave clock count for current Sync cycle, SlaveClockCount(n) is given by:
  $\mathrm{SlaveClockCount(n)} = \mathrm{SlaveClockTime(n)} - \mathrm{SlaveClockTime(n-1)}$
- The difference between master and slave clock counts for current Sync cycle, ClockDiffCount(n) is given by:
  $\mathrm{ClockDiffCount(n)} = \mathrm{MasterClockCount(n)} - \mathrm{SlaveClockCount(n)}$
- The frequency scaling factor for slave clock, FreqScaleFactor(n) is given by:
  $\mathrm{FreqScaleFactor}\left(\mathrm{n}\right) = \frac{\mathrm{MasterClockCount(n)} + \mathrm{ClockDiffCount(n)}}{\mathrm{SlaveClockCount(n)}}$
- The frequency compensation value for Addend register, FreqCompensationValue(n) is given by:
  $\mathrm{FreqCompensationValue(n)} = \mathrm{FreqScaleFactor(n)} \times \mathrm{FreqCompensationValue(n-1)}$

In theory, this algorithm achieves lock in one Sync cycle; however, it may take several cycles, because of changing network propagation delays and operating conditions.

This algorithm is self-correcting: If for any reason the slave clock is initially set to a value from the master that is incorrect, the algorithm corrects it at the cost of more Sync cycles.

27.4.9 Programming Guidelines for Energy Efficient Ethernet

27.4.9.1 Entering and Exiting the TX LPI Mode

Complete the following steps during GMAC initialization:

1. Read the PHY register through the MDIO interface, check if the remote end has the EEE capability, and then negotiate the timer values.
2. Program the PHY registers through the MDIO interface (including the RX_CLK_stoppable bit that indicates to the PHY whether to stop RX clock in LPI mode.)
3. Program bits [26:16] and bits [15:0] in LPI Timers Control Register (LPI_Timers_Control).
4. Read the link status of the PHY chip by using the MDIO interface and update bit [17] of LPI Control and Status Register (LPI_Control_Status) accordingly. This update should be done whenever the link status in the PHY chip changes.
5. Set bit [16] of LPI Control and Status Register (LPI_Control_Status) to make the GMAC enter the LPI state. The GMAC enters the LPI mode after completing the transmission in progress and sets bit [0].

Note:
- To make the GMAC enter the LPI state only after it completes the transmission of all queued frames in the TX FIFO, set bit [19] in LPI Control and Status Register (LPI_Control_Status).
- To switch off the CSR clock (PCLKM) or power to the rest of the system during the LPI state, wait for the TLPIEN interrupt of LPI Control and Status Register (LPI_Control_Status) to be generated. Restore the clocks before performing step 6 to come out of the LPI state.

6. Reset bit [16] of LPI Control and Status Register (LPI_Control_Status) to bring the GMAC out of the LPI state. The GMAC waits for the time programmed in bits [15:0] before setting the TLPIEX interrupt status bit and resuming the transmission.

27.4.9.2 Gating Off the CSR Clock in the LPI Mode

You can gate off the CSR clock (PCLKM) to save the power when the GMAC is in the Low Power Idle (LPI) mode.

(1) Gating Off the CSR Clock in the RX LPI Mode

The following operations are performed when the GMAC receives the LPI pattern from the PHY.

1. The MAC RX enters the LPI mode and the RX LPI entry interrupt status [RLPIEN interrupt of LPI Control and Status Register (LPI_Control_Status)] is set.
2. The interrupt signal (GMAC_SBD) is asserted. The GMAC_SBD interrupt is cleared when the host reads the LPI Control and Status Register (LPI_Control_Status).

After the GMAC_SBD interrupt is asserted and the MAC TX is also in the LPI mode, you can gate off the CSR clock. If the MAC TX is not in the LPI mode when you gate off the CSR clock, the events on the GMAC transmitter do not get reported or updated in the CSR.

For restoring the CSR clock, wait for the LPI exit indication from the PHY after which the GMAC asserts the LPI exit interrupt on GMAC_LPI (synchronous to Receive Clock). The GMAC_LPI interrupt is cleared when LPI_Control_Status is read.

(2) Gating Off the CSR Clock in the TX LPI Mode

The following operations are performed when bit [16] (LPIEN) of LPI Control and Status Register (LPI_Control_Status) is set:

1. The Transmit LPI Entry interrupt (TLPIEN bit of register LPI_Control_Status) is set.
2. The interrupt signal (GMAC_SBD) is asserted. The GMAC_SBD interrupt is cleared when the host reads the register LPI_Control_Status.

After the GMAC_SBD interrupt is asserted and the MAC RX is also in the LPI mode, you can gate off the CSR clock. If the MAC RX is not in the LPI mode when you gate off the CSR clock, the events on the GMAC receiver do not get reported or updated in the CSR.

For restoring the CSR clock, switch on the CSR clock when the GMAC must come out of the TX LPI mode. After the CSR clock is resumed, reset bit [16] (LPIEN) of LPI Control and Status Register (LPI_Control_Status) to bring the GMAC out of the LPI mode.

27.4.10 Enhanced Descriptors

The integrated DMA transfers data based on a linked list of descriptors. The descriptors are created in the system memory. Descriptor contains two buffers, two byte-count buffers, and two address pointers, which enable the adapter port to be compatible with various types of memory management schemes.

Note: There is no limit for the number of descriptors that can be used for a single frame.

The enhanced descriptor structure can have 8 DWORDS (32 bytes). The features of the enhanced descriptor structure are:

- The enhanced descriptor structure is implemented to support buffers of up to 8 KB (useful for Jumbo frames).
- There is an assignment of control and status bits in TDES0, TDES1, RDES0, and RDES1.
- The transmit descriptor stores the timestamp in TDES6 and TDES7.
- This receive descriptor structure is also used for storing the extended status (RDES4) and timestamp (RDES6 and RDES7).
- One of the following options for descriptor structure can be selected:
  - If timestamping is enabled in Timestamp Control Register (Timestamp_Control) or Checksum Offload is enabled in MAC Configuration Register (MAC_Configuration), software must allocate 32 bytes (8 DWORDS) of memory for every descriptor. For this, software must set bit [7] (Enhanced Descriptor Size) of Bus Mode Register (Bus_Mode).
  - If timestamping and Checksum Offload is not enabled, the extended descriptors (DES4 to DES7) are not required. Therefore, software can use enhanced descriptors with the default size of 16 bytes (4 DWORDS). For this, software must reset bit [7] (Enhanced Descriptor Size) of Bus Mode Register (Bus_Mode) to 0.

27.4.10.1 Transmit Descriptor

The transmit descriptor structure is shown in Table 27.2. The application software must program the control bits TDES0[31:18] during descriptor initialization. When the DMA updates the descriptor, it write backs all the control bits except the OWN bit (which it clears) and updates the status bits [7:0]. The contents of the transmitter descriptor word 0 (TDES0) through word 3 (TDES3) are given in Table 27.4 through Table 27.7, respectively.

With the advance timestamp support, the snapshot of the timestamp to be taken can be enabled for a given frame by setting bit [25] (TTSE) of TDES0. When the descriptor is closed (that is, when the OWN bit is cleared), the timestamp is written into TDES6 and TDES7. This is indicated by the status bit [17] (TTSS) of TDES0 shown in Table 27.2. The contents of TDES6 and TDES7 are mentioned in Table 27.8 and Table 27.9.

Note: When Advanced Timestamp feature is enabled, software must set bit [7] of Bus Mode Register (Bus_Mode), so that the DMA operates with extended descriptor size. When this control bit is reset, the TDES4 to TDES7 descriptor space is not valid.

Table 27.2 Transmit descriptor fields - enhanced format

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDES0	OWN	Ctrl [30:26]					TTSE	Ctrl [24:18]							TTSS	Status [16:7]										Ctrl/Status [6:3]				Status [2:0]		
TDES1	Reserved						Buffer 2 Byte Count [28:16]										Reserved									Buffer 1 Byte Count [12:0]						
TDES2	Buffer 1 Address [31:0]																															
TDES3	Buffer 2 Address [31:0] or Next Descriptor Address [31:0]																															
TDES4	Reserved																															
TDES5	Reserved																															
TDES6	Transmit Timestamp Low [31:0]																															
TDES7	Transmit Timestamp High [31:0]																															

The DMA always reads or fetches 4 DWORDS of the descriptor from system memory to obtain the buffer and control information as shown in Table 27.3.

Table 27.3 Transmit descriptor fetch (read) for enhanced format

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDES0	OWN	Ctrl [30:26]					TTSE	Ctrl [24:18]							Reserved for Status [17:7]											Reserved				Reserved for Status [2:0]		
TDES1	Reserved						Buffer 2 Byte Count [28:16]										Reserved									Buffer 1 Byte Count [12:0]						
TDES2	Buffer 1 Address [31:0]																															
TDES3	Buffer 2 Address [31:0] or Next Descriptor Address [31:0]																															

Table 27.4 Transmit descriptor word 0 (TDES0) (1 of 3)

Bit position	Bit name	Description
0	DB	Deferred Bit When set, this bit indicates that the GMAC defers before transmission because of the presence of carrier. This bit is valid only in the half-duplex mode.
1	UF	Underflow Error When set, this bit indicates that the GMAC aborted the frame because the data arrived late from the Host memory. Underflow error indicates that the DMA encountered an empty transmit buffer while transmitting the frame. The transmission process enters the Suspended state and sets both bit [5] (Transmit Underflow bit) in Status Register (Status) and bit [0] (Transmit Interrupt bit) in Status Register (Status).
2	ED	Excessive Deferral When set, this bit indicates that the transmission has ended because of excessive deferral of over 24,288-bit times (155,680-bit times in 1000 Mbps mode or if Jumbo Frame is enabled) if bit [4] (Deferral Check bit) in MAC Configuration Register (MAC_Configuration) is set high.

Table 27.4 Transmit descriptor word 0 (TDES0) (2 of 3)

Bit position	Bit name	Description
6:3	CC[3:0]	Collision Count (Status field) These status bits indicate the number of collisions that occurred before the frame was transmitted. This count is not valid when the Excessive Collisions bit (TDES0[8]) is set. The core updates this status field only in the half-duplex mode.
7	VF	VLAN Frame When set, this bit indicates that the transmitted frame is a VLAN-type frame.
8	EC	Excessive Collision When set, this bit indicates that the transmission was aborted after 16 successive collisions while attempting to transmit the current frame. If bit [9] (Disable Retry bit) in the MAC Configuration Register (MAC_Configuration) is set, this bit is set after the first collision, and the transmission of the frame is aborted.
9	LC	Late Collision When set, this bit indicates that frame transmission is aborted because of a collision occurring after the collision window (64 byte-times, including preamble, in MII mode and 512 byte-times, including preamble and carrier extension, in GMII mode). This bit is not valid if the Underflow Error bit is set.
10	NC	No Carrier When set, this bit indicates that the Carrier Sense signal from the PHY was not asserted during transmission.
11	LOC	Loss of Carrier When set, this bit indicates that a loss of carrier occurred during frame transmission (that is, the CRS signal was inactive for one or more transmit clock periods during frame transmission). This is valid only for the frames transmitted without collision when the GMAC operates in the half-duplex mode.
12	IPE	IP Payload Error When set, this bit indicates that GMAC transmitter detected an error in the TCP, UDP, or ICMP IP datagram payload. The transmitter checks the payload length received in the IPv4 or IPv6 header against the actual number of TCP, UDP, or ICMP packet bytes received from the application and issues an error status in case of a mismatch.
13	FF	Frame Flushed When set, this bit indicates that the DMA or MTL flushed the frame because of a software Flush command given by the CPU.
14	JT	Jabber Timeout When set, this bit indicates the GMAC transmitter has experienced a jabber timeout. This bit is only set when bit [22] (Jabber Disable) of MAC Configuration Register (MAC_Configuration) is not set.
15	ES	Error Summary Indicates the logical OR of the following bits: TDES0[14]: Jabber Timeout TDES0[13]: Frame Flush TDES0[11]: Loss of Carrier TDES0[10]: No Carrier TDES0[9]: Late Collision TDES0[8]: Excessive Collision TDES0[2]: Excessive Deferral TDES0[1]: Underflow Error TDES0[16]: IP Header Error TDES0[12]: IP Payload Error
16	IHE	IP Header Error When set, this bit indicates that the GMAC transmitter detected an error in the IP datagram header. The transmitter checks the header length in the IPv4 packet against the number of header bytes received from the application and indicates an error status if there is a mismatch. For IPv6 frames, a header error is reported if the main header length is not 40 bytes. Furthermore, the Ethernet Length/Type field value for an IPv4 or IPv6 frame must match the IP header version received with the packet. For IPv4 frames, an error status is also indicated if the Header Length field has a value less than 0x5. If COE detects an IP header error, it still inserts an IPv4 header checksum if the Ethernet Type field indicates an IPv4 payload.
17	TTSS	Transmit Timestamp Status This field is used as a status bit to indicate that a timestamp was captured for the described transmit frame. When this bit is set, TDES2 and TDES3 have a timestamp value captured for the transmit frame. This field is only valid when the Last Segment Control bit (TDES0[29]) of the descriptor is set.

Table 27.4 Transmit descriptor word 0 (TDES0) (3 of 3)

Bit position	Bit name	Description
19:18	—	These bits are read as 0. The write value should be 0.
20	TCH	Second Address Chained When set, this bit indicates that the second address in the descriptor is the next descriptor address rather than the second buffer address. When TDES0[20] is set, TBS2 (TDES1[28:16]) is a "don't care" value. TDES0[21] takes precedence over TDES0[20].
21	TER	Transmit End of Ring When set, this bit indicates that the descriptor list reached its final descriptor. The DMA returns to the base address of the list, creating a descriptor ring.
23:22	CIC[1:0]	Checksum Insertion Control These bits control the checksum calculation and insertion. The following list describes the bit encoding: 00b: Checksum Insertion Disabled. 01b: Only IP header checksum calculation and insertion are enabled. 10b: IP header checksum and payload checksum calculation and insertion are enabled, but pseudo header checksum is not calculated in hardware. 11b: IP Header checksum and payload checksum calculation and insertion are enabled, and pseudo header checksum is calculated in hardware. This field is valid when the First Segment Control bit (TDES0[28]) is set.
24	—	This bit is read as 0. The write value should be 0.
25	TTSE	Transmit Timestamp Enable When set, this bit enables IEEE 1588 hardware timestamping for the transmit frame referenced by the descriptor. This field is valid only when the First Segment Control bit (TDES0[28]) is set.
26	DP	Disable Pad When set, the GMAC does not automatically add padding to a frame shorter than 64 bytes. When this bit is reset, the DMA automatically adds padding and CRC to a frame shorter than 64 bytes, and the CRC field is added despite the state of the DC bit (TDES0[27]). This is valid only when the first segment (TDES0[28]) is set.
27	DC	Disable CRC When this bit is set, the GMAC does not append a cyclic redundancy check (CRC) to the end of the transmitted frame. This is valid only when the first segment (TDES0[28]) is set.
28	FS	First Segment When set, this bit indicates that the buffer contains the first segment of a frame.
29	LS	Last Segment When set, this bit indicates that the buffer contains the last segment of the frame. When this bit is set, the TBS1 or TBS2 field in TDES1 should have a non-zero value.
30	IC	Interrupt on Completion When set, this bit sets the Transmit Interrupt (bit [0] of Status Register) after the present frame has been transmitted. This bit is valid only when the last segment bit (TDES0[29]) is set.
31	OWN	Own Bit When set, this bit indicates that the descriptor is owned by the DMA. When this bit is reset, it indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame transmission or when the buffers allocated in the descriptor are read completely. The ownership bit of the first descriptor of the frame must be set after all subsequent descriptors belonging to the same frame have been set. This avoids a possible race condition between fetching a descriptor and the driver setting an ownership bit.

Table 27.5 Transmit descriptor word 1 (TDES1) (1 of 2)

Bit position	Bit name	Description
12:0	TBS1[12:0]	Transmit Buffer 1 Size These bits indicate the first data buffer byte size, in bytes. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or the next descriptor, depending on the value of TCH (TDES0[20]).
15:13	—	These bits are read as 0. The write value should be 0.

Table 27.5 Transmit descriptor word 1 (TDES1) (2 of 2)

Bit position	Bit name	Description
28:16	TBS2[12:0]	Transmit Buffer 2 Size This field indicates the second data buffer size in bytes. This field is not valid if TDES0[20] is set. The transmit DMA transfers the exact number of bytes (indicated by buffer size field of TDES1) towards the GMAC. If a descriptor is marked as first (FS bit of TDES1 is set), then the DMA marks the first transfer from the buffer as the start of frame. If a descriptor is marked as last (LS bit of TDES1), then the DMA marks the last transfer from that data buffer as the end of frame to the MTL.
31:29	—	These bits are read as 0. The write value should be 0.

Table 27.6 Transmit descriptor word 2 (TDES2)

Bit position	Bit name	Description
31:0	Buffer 1 Address Pointer[31:0]	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There is no limitation on the buffer address alignment.

Table 27.7 Transmit descriptor word 3 (TDES3)

Bit position	Bit name	Description
31:0	Buffer 2 Address Pointer or Next Descriptor Address[31:0]	Buffer 2 Address Pointer (Next Descriptor Address) Indicates the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (TDES0[20]) bit is set, this address contains the pointer to the physical memory where the next descriptor is present. The buffer address pointer must be aligned to the bus width only when TDES0[20] is set. (LSBs are ignored internally.)

Table 27.8 Transmit descriptor word 6 (TDES6)

Bit position	Bit name	Description
31:0	TTSL[31:0]	Transmit Frame Timestamp Low This field is updated by DMA with the least significant 32 bits of the timestamp captured for the corresponding transmit frame. This field has the timestamp only if the Last Segment bit (LS) in the descriptor is set and Timestamp status (TTSS) bit is set.

Table 27.9 Transmit descriptor word 7 (TDES7)

Bit position	Bit name	Description
31:0	TTSH[31:0]	Transmit Frame Timestamp High This field is updated by DMA with the most significant 32 bits of the timestamp captured for the corresponding transmit frame. This field has the timestamp only if the Last Segment bit (LS) in the descriptor is set and Timestamp status (TTSS) bit is set.

27.4.10.2 Receive Descriptor

The structure of the received descriptor is shown in Table 27.10. This can have 32 bytes of descriptor data (8 DWORDs).

Note: When either of these features is enabled, software must set bit [7] of Bus Mode Register (Bus_Mode) so that the DMA operates with extended descriptor size. When this control bit is reset, the RDES0[0] is always cleared and the RDES4 to RDES7 descriptor space is not valid.

Table 27.10 Receive descriptor fields - enhanced format (1 of 2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDES0	OWN	Status [30:0]																														
RDES1	CTRL	RES [30:29]		Buffer 2 Byte Count [28:16]													CTRL [15:14]		RES	Buffer 1 Byte Count [12:0]												
RDES2	Buffer 1 Address [31:0]																															
RDES3	Buffer 2 Address [31:0] or Next Descriptor Address [31:0]																															
RDES4	Extended Status [31:0]																															

Table 27.10 Receive descriptor fields - enhanced format (2 of 2)

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDES5	Reserved																															
RDES6	Receive Timestamp Low [31:0]																															
RDES7	Receive Timestamp High [31:0]																															

The contents of RDES0 are identified in Table 27.11. The contents of RDES1 through RDES3 are identified in Table 27.12 through Table 27.14, respectively.

Note: The status bits are valid when the LS bit is set. Therefore, when multiple descriptors are used for a single frame, status bits of only last descriptor (that is, descriptor with LS bit set) are valid.

Table 27.11 Receive descriptor fields 0 (RDES0) (1 of 2)

Bit position	Bit name	Description
0	Extended Status Available	Extended Status Available When set, this bit indicates that the extended status is available in descriptor word 4 (RDES4). This is valid only when the Last Descriptor bit (RDES0[8]) is set. This bit is invalid when bit [30] is set. This bit is set even when IP Checksum Offload engine bypasses the processing of received frame. The bypassing may be because of non-IP frame or IP frame with non-TCP/UDP/ICMP payload.
1	CE	CRC Error When set, this bit indicates that a Cyclic Redundancy Check (CRC) Error occurred on the received frame. This field is valid only when the Last Descriptor (RDES0[8]) is set.
2	DE	Dribble Bit Error When set, this bit indicates that the received frame has a non-integer multiple of bytes (odd nibbles). This bit is valid only in the MII Mode.
3	RE	Receive Error When set, this bit indicates that the GMII/MII RXER signal is asserted while GMII/MII RXDV is asserted during frame reception. This error also includes carrier extension error in the GMII and half-duplex mode. Error can be of less or no extension, or error (rxd ≠ 0f) during extension.
4	RWT	Receive Watchdog Timeout When set, this bit indicates that the Receive Watchdog Timer has expired while receiving the current frame and the current frame is truncated after the Watchdog Timeout.
5	FT	Frame Type When set, this bit indicates that the Receive Frame is an Ethernet-type frame (the Length/Type field is greater than or equal to 1,536). When this bit is reset, it indicates that the received frame is an IEEE 802.3 frame. This bit is not valid for Runt frames less than 14 bytes.
6	LC	Late Collision When set, this bit indicates that a late collision has occurred while receiving the frame in the half-duplex mode.
7	Timestamp Available or Giant Frame	Timestamp Available or Giant Frame When set, this bit indicates that a snapshot of the timestamp is written in descriptor words 6 (RDES6) and 7 (RDES7). This is valid only when the Last Descriptor bit (RDES0[8]) is set. Otherwise, this bit, when set, indicates the Giant Frame Status. Giant frames are larger than 1,518-byte (or 1,522-byte for VLAN or 2,000-byte when bit [27] of MAC Configuration register is set) normal frames and larger than 9,018-byte (9,022-byte for VLAN) frame when Jumbo Frame processing is enabled.
8	LS	Last Descriptor When set, this bit indicates that the buffers pointed to by this descriptor are the last buffers of the frame.
9	FS	First Descriptor When set, this bit indicates that this descriptor contains the first buffer of the frame. If the size of the first buffer is 0, the second buffer contains the beginning of the frame. If the size of the second buffer is also 0, the next descriptor contains the beginning of the frame.
10	VLAN	VLAN Tag When set, this bit indicates that the frame to which this descriptor is pointing is a VLAN frame tagged by the GMAC. The VLAN tagging depends on checking the VLAN fields of received frame based on the VLAN Tag Register (VLAN_Tag) setting.

Table 27.11 Receive descriptor fields 0 (RDES0) (2 of 2)

Bit position	Bit name	Description
11	OE	Overflow Error When set, this bit indicates that the received frame was damaged because of buffer overflow in MTL. This bit is set only when the DMA transfers a partial frame to the application. This happens only when the RX FIFO is operating in the threshold mode. In the store-and-forward mode, all partial frames are dropped completely in RX FIFO.
12	LE	Length Error When set, this bit indicates that the actual length of the frame received and that the Length/Type field does not match. This bit is valid only when the Frame Type (RDES0[5]) bit is reset.
13	SAF	Source Address Filter Fail When set, this bit indicates that the SA field of frame failed the SA Filter in the GMAC.
14	DE	Descriptor Error When set, this bit indicates a frame truncation caused by a frame that does not fit within the current descriptor buffers, and that the DMA does not own the next descriptor. The frame is truncated. This field is valid only when the Last Descriptor (RDES0[8]) is set.
15	ES	Error Summary Indicates the logical OR of the following bits: RDES0[1]: CRC Error RDES0[3]: Receive Error RDES0[4]: Watchdog Timeout RDES0[6]: Late Collision RDES0[7]: Giant Frame RDES4[4:3]: IP Header or Payload Error RDES0[11]: Overflow Error RDES0[14]: Descriptor Error This field is valid only when the Last Descriptor (RDES0[8]) is set.
29:16	FL[13:0]	Frame Length These bits indicate the byte length of the received frame that was transferred to host memory. This field is valid when Last Descriptor (RDES0[8]) is set and either the Descriptor Error (RDES0[14]) or Overflow Error (RDES0[11]) bits are reset. The frame length also includes the two bytes appended to the Ethernet frame when IP checksum calculation (Type 1) is enabled and the received frame is not a MAC control frame. This field is valid when Last Descriptor (RDES0[8]) is set. When the Last Descriptor and Error Summary bits are not set, this field indicates the accumulated number of bytes that have been transferred for the current frame. The inclusion of CRC length in the frame length depends on the settings of bit [7] and bit [25] in MAC Configuration Register (MAC_Configuration).
30	AFM	Destination Address Filter Fail When set, this bit indicates a frame that failed in the DA Filter in the GMAC.
31	OWN	Own Bit When set, this bit indicates that the descriptor is owned by the DMA of the GMAC. When this bit is reset, this bit indicates that the descriptor is owned by the Host. The DMA clears this bit either when it completes the frame reception or when the buffers that are associated with this descriptor are full.

Table 27.12 Receive descriptor fields 1 (RDES1) (1 of 2)

Bit position	Bit name	Description
12:0	RBS1[12:0]	Receive Buffer 1 Size Indicates the first data buffer size in bytes. The buffer size must be a multiple of 4, even if the value of RDES2 (buffer 1 address pointer) is not aligned. When the buffer size is not a multiple of 4, the resulting behavior is undefined. If this field is 0, the DMA ignores this buffer and uses Buffer 2 or next descriptor depending on the value of RCH (Bit 14).
13	—	This bit is read as 0. The write value should be 0.
14	RCH	Second Address Chained When set, this bit indicates that the second address in the descriptor is the next descriptor address rather than the second buffer address. When this bit is set, RBS2 (RDES1[28:16]) is a "don't care" value. RDES1[15] takes precedence over RDES1[14].
15	RER	Receive End of Ring When set, this bit indicates that the descriptor list reached its final descriptor. The DMA returns to the base address of the list, creating a descriptor ring.

Table 27.12 Receive descriptor fields 1 (RDES1) (2 of 2)

Bit position	Bit name	Description
28:16	RBS2[12:0]	Receive Buffer 2 Size These bits indicate the second data buffer size, in bytes. The buffer size must be a multiple of 4, even if the value of RDES3 (buffer 2 address pointer) is not aligned to bus width. If the buffer size is not an appropriate multiple of 4, the resulting behavior is undefined. This field is not valid if RDES1[14] is set.
30:29	—	These bits are read as 0. The write value should be 0.
31	DIC	Disable Interrupt on Completion When set, this bit prevents setting the RI bit[6] for the received frame ending in the buffer indicated by this descriptor. This, in turn, disables the assertion of the interrupt to Host because of RI for that frame. This bit is valid only when the Last Descriptor bit (RDES0[8]) is set.

Table 27.13 Receive descriptor fields 2 (RDES2)

Bit position	Bit name	Description
31:0	Buffer 1 Address Pointer	Buffer 1 Address Pointer These bits indicate the physical address of Buffer 1. There are no limitations on the buffer address alignment except for the following condition: The DMA uses the configured value for its address generation when the RDES2 value is used to store the start of frame. The DMA performs a write operation with the RDES2[1:0] bits as 0 during the transfer of the start of frame but the frame data is shifted as per the actual Buffer address pointer. The DMA ignores RDES2[1:0] if the address pointer is to a buffer where the middle or last part of the frame is stored.

Table 27.14 Receive descriptor fields 3 (RDES3)

Bit position	Bit name	Description
31:0	Buffer 2 Address Pointer or Next Descriptor Address[31:0]	Buffer 2 Address Pointer (Next Descriptor Address) These bits indicate the physical address of Buffer 2 when a descriptor ring structure is used. If the Second Address Chained (RDES1[14]) bit is set, this address contains the pointer to the physical memory where the next descriptor is present. If RDES1[14] is set, the buffer (next descriptor) address pointer must be bus width-aligned (RDES3[1:0] = 0. LSBs are ignored internally.) However, when RDES1[14] is reset, there are no limitations on the RDES3 value, except for the following condition: ● The DMA uses the configured value for its buffer address generation when the RDES3 value is used to store the start of frame. ● The DMA ignores RDES3[1:0] if the address pointer is to a buffer where the middle or last part of the frame is stored.

The extended status written is as shown in Table 27.15. The extended status is written only when there is status related to IPC or timestamp available. The availability of extended status is indicated by bit [0] of RDES0.

Table 27.15 Receive descriptor fields 4 (RDES4) (1 of 2)

Bit position	Bit name	Description
2:0	IP Payload Type[2:0]	IP Payload Type These bits indicate the type of payload encapsulated in the IP datagram processed by the Receive Checksum Offload Engine (COE). The COE also sets these bits to 000b if it does not process the payload of the IP datagram due to an IP header error or fragmented IP. 000b: Unknown or did not process IP payload 001b: UDP 010b: TCP 011b: ICMP 1xxb: Reserved This bit is valid when either bit [7] or bit [6] is set.
3	IP Header Error	IP Header Error When set, this bit indicates that either the 16-bit IPv4 header checksum calculated by the core does not match the received checksum bytes, or the IP datagram version is not consistent with the Ethernet Type value. This bit is valid when either bit [7] or bit [6] is set.

Table 27.15 Receive descriptor fields 4 (RDES4) (2 of 2)

Bit position	Bit name	Description
4	IP Payload Error	IP Payload Error When set, this bit indicates that the 16-bit IP payload checksum (that is, the TCP, UDP, or ICMP checksum) that the core calculated does not match the corresponding checksum field in the received segment. It is also set when the TCP, UDP, or ICMP segment length does not match the payload length value in the IP Header field. This bit is valid when either bit [7] or bit [6] is set.
5	IP Checksum Bypassed	IP Checksum Bypassed When set, this bit indicates that the checksum offload engine is bypassed.
6	IPv4 Packet Received	IPv4 Packet Received When set, this bit indicates that the received packet is an IPv4 packet. This bit is updated only when bit [10] (IPC) of MAC Configuration Register (MAC_Configuration) is set.
7	IPv6 Packet Received	IPv6 Packet Received When set, this bit indicates that the received packet is an IPv6 packet. This bit is updated only when bit [10] (IPC) of MAC Configuration Register (MAC_Configuration) is set.
11:8	Message Type[3:0]	Message Type These bits are encoded to give the type of the message received. 0000b: No PTP message received 0001b: SYNC (all clock types) 0010b: Follow_Up (all clock types) 0011b: Delay_Req (all clock types) 0100b: Delay_Resp (all clock types) 0101b: Pdelay_Req (in peer-to-peer transparent clock) 0110b: Pdelay_Resp (in peer-to-peer transparent clock) 0111b: Pdelay_Resp_Follow_Up (in peer-to-peer transparent clock) 1000b: Announce 1001b: Management 1010b: Signaling 1011b to 1110b: Reserved 1111b: PTP packet with Reserved message type
12	PTP Frame Type	PTP Frame Type When set, this bit indicates that the PTP message is sent directly over Ethernet. When this bit is not set and the message type is non-zero, it indicates that the PTP message is sent over UDP-IPv4 or UDP-IPv6. The information about IPv4 or IPv6 can be obtained from bits [6] and [7].
13	PTP Version	PTP Version When set, this bit indicates that the received PTP message is having the IEEE 1588 version 2 format. When reset, it has the version 1 format.
14	Timestamp Dropped	Timestamp Dropped When set, this bit indicates that the timestamp was captured for this frame but got dropped in the MTL RX FIFO because of overflow.
31:15	—	These bits are read as 0. The write value should be 0.

RDES6 and RDES7 contain the snapshot of the timestamp. The availability of the snapshot of the timestamp in RDES6 and RDES7 is indicated by bit [7] in the RDES0 descriptor. The contents of RDES6 and RDES7 are identified in Table 27.16 and Table 27.17, respectively.

Table 27.16 Receive descriptor fields 6 (RDES6)

Bit position	Bit name	Description
31:0	RTSL[31:0]	Receive Frame Timestamp Low This field is updated by DMA with the least significant 32 bits of the timestamp captured for the corresponding receive frame. This field is updated by DMA only for the last descriptor of the receive frame which is indicated by Last Descriptor status bit (RDES0[8]).

Table 27.17 Receive descriptor fields 7 (RDES7)

Bit position	Bit name	Description
31:0	RTSH[31:0]	Receive Frame Timestamp High This field is updated by DMA with the most significant 32 bits of the timestamp captured for the corresponding receive frame. This field is updated by DMA only for the last descriptor of the receive frame which is indicated by Last Descriptor status bit (RDES0[8]).

27.4.11 PTP Pulse Generator

The 4 Pulse generators are implemented beside GMAC. These 4 Pulse generators can generate the pulses which have any of period and any of pulse width by utilizing system timer of GMAC (see section 27.4.8. System Time Register Module).

This Pulse generator can be configured as following Table and following Figure by the register (see section 27.3.109. PTP Timer Pulse Control Registers).

Table 27.18 Parameter list of pulse signal generation

Parameter	Registers	Description
Pulse generation enable	SWTMENn	Enables or disables to output pulse signal
Pulse output start time	SWTMSTSECn, SWTMSTNSn	Set the start time of the output signal in seconds and nanoseconds. After set the start time and set 1 to the SWTMEN register in order to enable to output the pulse signal, the pulse output starts when current time is over the start time set. If pulse output is enabled after the set start time, no pulse is output.
Pulse period	SWTMPSECn, SWTMPNSn	Set the pulse period of the output signal in seconds and nanoseconds. The value set in the SWTMPNS register must be larger than the value below. In case PCLKM = 100 MHz, 18 × SSINC[7:0] bits in Sub_Second_Increment Register (144 ns). In case PCLKM = 75 MHz, 20 × SSINC[7:0] bits in Sub_Second_Increment Register (160 ns). The value must be set before pulse output is enabled.
Pulse width	SWTMWTHn	Set the pulse high level width in the cycle number of GMAC_PTP_REFCLK_I (8 ns). Output pulse width is the set value × 8 (ns). If the pulse width set in the SWTMWTH register is greater than pulse period, output is fixed to high level. If the pulse width is set to 0, no pulse is generated and output is fixed to low level. The value must be set before pulse output is enabled.
Max nanosecond counter	SWTMMAXPn	This register sets the boundary value in nanoseconds to carry from the nanosecond field to the second field. When the nanosecond field reaches the set value of this register, the second field is carried up. This register is normally used with the initial value of 0x3B9ACA00 (= 10^9) and TSCTRLSSR bit in Timestamp_Control register is set to 1. In this case, the nanosecond field counts from 0 to 0x3B9AC9FF (= 10^9 - 1) and then carries to the second field. If 0x80000000 (= 2^{31}) is set in this register and TSCTRLSSR bit in Timestamp_Control register is reset to 0, the boundary value for carrying to the second field is interpreted as 0x80000000 (= 2^{31}). In this case, the nanosecond field is fully counted from 0 to 0x7FFFFFFF (= 2^{31} - 1) and then carries to the second field.
Pulse rising time latch	SWTMLATSECn, SWTMLATNSn	The pulse output rising time is latched into the registers. The value is updated whenever pulse output rises.

Note: n = 0 to 3

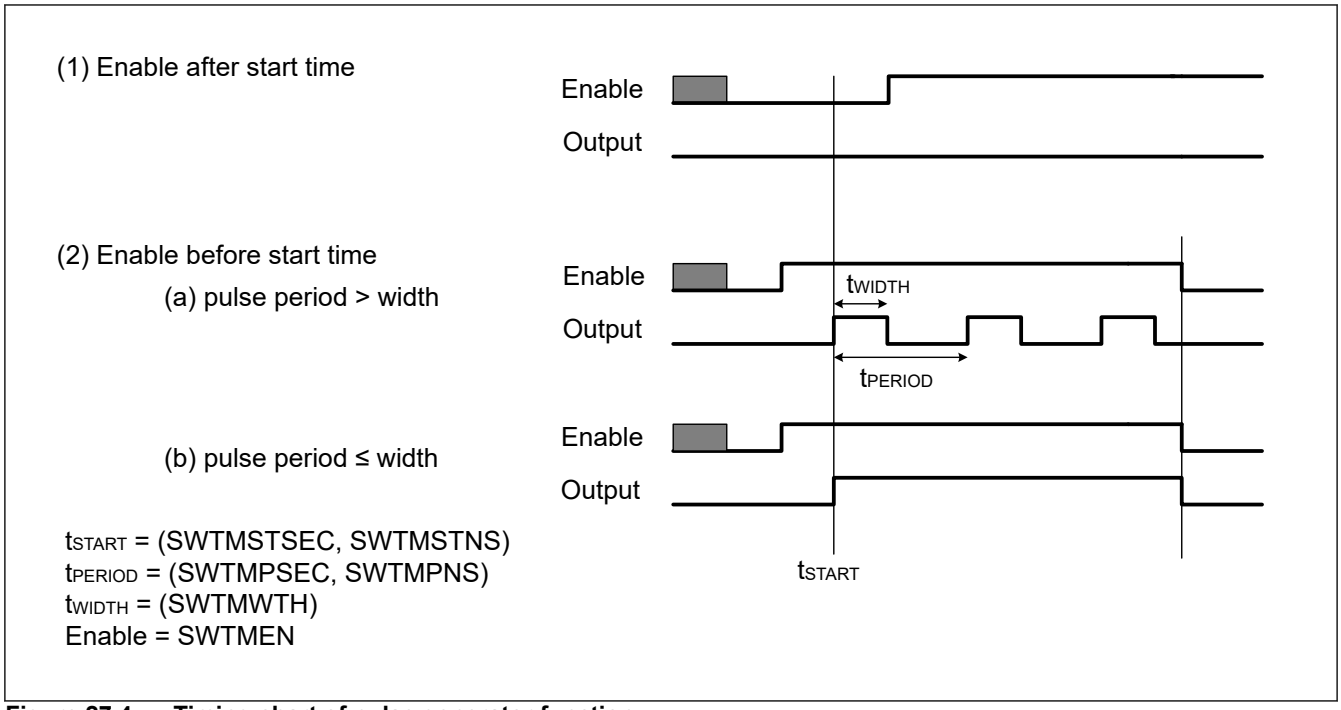


Figure 27.4 Timing chart of pulse generator function

27.5 Usage Notes

27.5.1 Configurations

The configuration depends on implementation is as follows:.

Clock

- CSR clock: AHB clock (PCLKM)

Features

- System Interface Configuration: GMAC-AXI
- IEEE 1588 Timestamp feature: Enabled
- IEEE 1588 Auxiliary Snapshot feature: Enabled
- Advanced Timestamp feature: Enabled
- IP Checksum Offload feature: Enabled
- Full Checksum Offload (Type 2) features: Enabled
- Power Management feature: Enabled
- AV feature: Disabled
- Hash Table width: 256

section 27.3.107. HW_Feature : HW Feature Register also shows the current configuration.

27.5.2 Restriction

27.5.2.1 Ethernet MAC

When any register content is being transferred to a different clock domain after a write operation, there should not be any further writes to the same location until the first write is updated. Otherwise, the second write operation does not get updated to the destination clock domain. Therefore, the delay between two writes to the same register location should be at least 4 cycles of the destination clock (PHY receive clock, PHY transmit clock, or PTP clock).

The transmit and receive data buffers do not have any restrictions on start address alignment. For example, in systems with 32-bit memory, the start address for the buffers can be aligned to any of the 4 bytes. However, the DMA always initiates write transfers, with address aligned to the bus width, with dummy data (old data) in the byte lanes that are not valid. This typically happens during the transfer of the beginning or end of an Ethernet frame. The software driver must discard the dummy bytes based on the start address of the buffer and size of the frame.

27.5.3 MAC Management Counters

The counters in the MAC Management Counters (MMC) module can be viewed as an extension of the register address space of the CSR (Control and Status Registers) module. The MMC module maintains a set of registers for gathering statistics on the received and transmitted frames. The register set includes a control register for controlling the behavior of the registers, two 32-bit registers containing interrupts generated (receive and transmit), and two 32-bit registers containing masks for the Interrupt register (receive and transmit). These registers are accessible from the application through the MAC Control Interface (MCI). Each register is 32 bits width. The write data is qualified with the corresponding byte enable signals. Therefore, non 32-bit accesses are allowed as long as the address is word aligned. The MMCs are accessed using transactions, in the same way the CSR address space is accessed.

The MMC counters are free running. There is no separate enable for the counters to start. It starts counting when corresponding frame is received or transmitted. The receive MMC counters are updated for frames that are passed by the Address Filter (AFM) block. The statistics of frames, dropped by the AFM module, are not updated unless they are runt frames of less than 6 bytes (DA bytes are not received fully). To get statistics of all frames, set bit [0] in the MAC Frame Filter register (MAC_Frame_Filter).

27.5.3.1 Address Assignments

The following definitions define the terminology:

- Transmitted frames are considered good if transmitted successfully. In other words, a transmitted frame is good if the frame transmission is not aborted because of any of the following errors:
  - Jabber Timeout
  - No Carrier or Loss of Carrier
  - Late Collision
  - Frame Underflow
  - Excessive Deferral
  - Excessive Collision
- Received frames are considered good if none of the following errors exists:
  - CRC error
  - Runt Frame (shorter than 64 bytes)
  - Alignment error (in 10/100 Mbps only)
  - Length error (non Type frames only)
  - Out of Range (non Type frames only, longer than 1518 bytes)
  - GMII_RXER Input error
- The maximum frame size depends on the frame type, as follows:
  - Untagged frame maxsize = 1518
  - VLAN Frame maxsize = 1522
  - Envelope Frames maxsize = 2000 (when programmed in bit [27] of MAC Configuration Register (MAC_Configuration))

Note: The MMC counters registers at the following offset addresses are of type read-only and have the default value of 0:
- 0x0114 to 0x0178
- 0x0180 to 0x01E4
- 0x0210 to 0x0244
- 0x0250 to 0x0284

The RX MMC counters and RX IPC MMC counters are updated only for frames that pass the Destination Address filter except when bit UCDBC[8] in MMC Control Register (MMC_Control) is set for broadcast frames.

27.5.4 MMC Registers

MMC registers are from 0x8010_0100 to 0x8010_0110 (GMAC). See section 27.2. Register Map.

27.5.4.1 MMC Control Register (MMC_Control)

The MMC Control register establishes the operating mode of the management counters.

Note: Bit [0] (Counters Reset) has higher priority than bit [4] (Counter Preset). Therefore, when software tries to set both bits in the same write cycle, all counters are cleared and bit [4] is not set.

27.5.4.2 MMC Receive Interrupt Register (MMC_Receive_Interrupt)

The MMC Receive Interrupt register maintains the interrupts that are generated when the following occurs:

- Receive statistic counters reach half of their maximum values (0x8000_0000 for 32-bit counter and 0x8000 for 16-bit counter).
- Receive statistic counters cross their maximum values (0xFFFF_FFFF for 32-bit counter and 0xFFFF for 16-bit counter).

When the Counter Stop Rollover is set, then interrupts are set but the counter remains all 1s. The MMC Receive Interrupt register is a 32-bit width register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (bits [7:0]) of the respective counter must be read in order to clear the interrupt bit.

27.5.4.3 MMC Transmit Interrupt Register (MMC_Transmit_Interrupt)

The MMC Transmit Interrupt register maintains the interrupts generated when transmit statistic counters reach half their maximum values (0x8000_0000 for 32-bit counter and 0x8000 for 16-bit counter), and when they cross their maximum values (0xFFFF_FFFF for 32-bit counter and 0xFFFF for 16-bit counter). When Counter Stop Rollover is set, then interrupts are set but the counter remains all 1s. The MMC Transmit Interrupt register is a 32-bit width register. An interrupt bit is cleared when the respective MMC counter that caused the interrupt is read. The least significant byte lane (bits [7:0]) of the respective counter must be read in order to clear the interrupt bit.

27.5.4.4 MMC Receive Interrupt Mask Register (MMC_Receive_Interrupt_Mask)

The MMC Receive Interrupt Mask register maintains the masks for the interrupts generated when receive statistic counters reach half of their maximum value or the maximum values. This register is 32 bits width.

27.5.4.5 MMC Transmit Interrupt Mask Register (MMC_Transmit_Interrupt_Mask)

The MMC Transmit Interrupt Mask register maintains the masks for the interrupts generated when the transmit statistic counters reach half of their maximum value or the maximum values. This register is 32 bits width.

27.5.5 Power Management Block

Power Management (PMT) Block registers are in 0x8010_0028 to 0x80100_002C (GMAC). See section 27.2. Register Map.

The power management (PMT) block supports the reception of network (remote) wakeup frames and magic packet frames. The PMT block does not perform the clock gate function, but generates interrupts for remote wakeup frames and magic packets that the GMAC receives.

Note:
- The magic packet feature is implemented based on the Magic Packet Technology white paper.
- The remote wake-up frame feature is implemented based on the Device Class Power Management Reference Specification and various implementation-specific white papers.

When you enable the power down mode in the PMT block, the GMAC drops all received frames and does not forward any frame to the MTL RXFIFO or the application. The GMAC comes out of the power down mode only when a magic packet or a remote wake-up frame is received and the corresponding detection is enabled.

The PMT block is available in the receive path of GMAC. You can select both types of power management frames (remote wakeup frame and magic packet). You can use bit [2] and bit [1] of PMT Control and Status Register (PMT_Control_Status) to generate power management events. The application should program these bits. You can access the PMT registers in similar manner as you access the GMAC CSR registers. For mapping information, see section 27.3.9. Remote_Wake_Up_Frame_Filter : Remote Wake-Up Frame Filter Register (Remote_Wake_Up_Frame_Filter) and section 27.3.10. PMT_Control_Status : PMT Control and Status Register (PMT_Control_Status).

27.5.5.1 PMT Block Description

(1) Remote Wake-Up Frame Filter Register

The register Remote_Wake_Up_Frame_Filter (address: 0x0028) accesses the Wake-up Frame Filter register (wkupfmfilter_reg[n], n = 0 to 7). To write to Wake-up Frame Filter register, write to 0x0028 in order to wkupfmfilter_reg0, wkupfmfilter_reg1, … wkupfmfilter_reg7 at eight times. Read is in a similar way. The GMAC updates the current pointer value of wkupfmfilter_reg[n] register in bits [26:24] of PMT Control and Status Register (PMT_Control_Status).

Note:
- If you are accessing these registers in byte or half-word mode, the internal counter, to access the appropriate wkupfmfilter_reg, is incremented when CPU accesses lane3.
- When any register content is being transferred to a different clock domain after a write operation, there should not be any further writes to the same location until the first write is updated. Otherwise, the second write operation does not get updated to the destination clock domain. Therefore, the delay between two writes to the same register location should be at least 4 cycles of the destination clock (PHY receive clock, PHY transmit clock, or PTP clock).

Table 27.19 Wake-up frame filter register

wkupfmfilter_reg0	Filter0 Byte Mask							
wkupfmfilter_reg1	Filter1 Byte Mask							
wkupfmfilter_reg2	Filter2 Byte Mask							
wkupfmfilter_reg3	Filter3 Byte Mask							
wkupfmfilter_reg4	RSVD	Filter3 Command	RSVD	Filter2 Command	RSVD	Filter1 Command	RSVD	Filter0 Command
wkupfmfilter_reg5	Filter3 Offset		Filter2 Offset		Filter1 Offset		Filter0 Offset	
wkupfmfilter_reg6	Filter1 CRC-16				Filter0 CRC-16			
wkupfmfilter_reg7	Filter3 CRC-16				Filter2 CRC-16			

Filter i Byte Mask

The Filter i Byte Mask register defines the bytes of the frame that are examined by filter i (0, 1, 2, and 3) to determine whether or not a frame is a remote wakeup frame. The MSB (31st bit) must be 0.

The bit j [30:0] is the byte mask. If bit j (byte number) of the byte mask is set, then the CRC block processes the Filter i Offset + j of the incoming frame. Otherwise, Filter i Offset + j is ignored.

Filter i Command

The 4-bit filter i command controls the filter i operation.
- Bit [3] specifies the address type, defining the destination address type of the pattern.
  When the bit is set, the pattern applies to only multicast packets. When the bit is reset, the pattern applies only to unicast packet.
- Bit [2] (Inverse Mode), when set, reverses the logic of the CRC16 hash function signal, to reject a packet with matching CRC_16 value.
  Bit [2], along with bit [1], allows a MAC to reject a subset of remote wakeup packets by creating filter logic such as Pattern 1 AND NOT Pattern 2.
- Bit [1] (And_Previous) implements the Boolean logic.
  When set, the result of the current entry is logically ANDed with the result of the previous filter. This AND logic allows a filter pattern longer than 32 bytes by splitting the mask among two, three, or four filters. This depends on the number of filters that have the And_Previous bit set.
- The And_Previous bit is applicable for more than one filter operation, where the filter result is ANDed with the previous filter result. For example, if And_Previous bit is set in Filter 1, the remote wakeup packet is detected and PMT

interrupt generated only if both Filter 0 and Filter 1 satisfy the remote wakeup packet detection and interrupt generation criteria mentioned in Table 27.20.

Note:
- The And_Previous bit setting is applicable within a set of 4 filters.
- Setting And_Previous bit of lowest number filter in the set of 4 filters has no effect. For example, setting of And_Previous bit of Filter 0 has no effect.
- If And_Previous bit is set for a filter and the previous filter is not enabled, And_Previous setting has no effect. For example, if Filter 2 And_Previous bit is set (bit [1] of Filter 2 command is set) but Filter 1 is not enabled (bit [0] of in Filter 1 command is reset), then only Filter 1 result is considered.

- Bit [0] is the enable for filter i. If bit [0] is not set, filter i is disabled.

Filter i Offset

This Filter i Offset register defines the offset (within the frame) from which the filter i examines the frames. This 8-bit pattern offset is the offset for the filter i first byte to be examined. The minimum allowed offset is 12, which refers to the 13th byte of the frame. The offset value 0 refers to the first byte of the frame.

Filter i CRC 16

This Filter i CRC 16 register contains the CRC_16 value calculated from the pattern and the byte mask programmed to the wake-up filter register block.

The 16-bit CRC calculation uses the following polynomial:

$$G(x) = x^{16} + x^{15} + x^2 + 1$$

Each mask, used in the hash function calculation, is compared with a 16-bit value associated with that mask. Each filter has the following:

- 32-bit Mask: Each bit in this mask corresponds to 1 byte in the detected packet. If the bit is 1', the corresponding byte is taken into the CRC16 calculation.
- 8-bit Offset Pointer: Specifies the byte to start the CRC16 computation.

The pointer and the mask are used together to locate the bytes to be used in the CRC16 calculations.

Table 27.20 lists the remote wakeup scenarios in which PMT interrupt is generated.

Table 27.20 Remote wakeup packet and PMT interrupt generation

Filter i command			Frame type and CRC status		Interrupt generation
CAST	INV	EN	Received frame cast type	CRC status	RWK INTR
0	0	1	Unicast	MATCH	Remote wakeup packet is detected and PMT interrupt is generated
0	1	1	Unicast	MISMATCH	Remote wakeup packet is detected and PMT interrupt is generated
1	0	1	Multicast	MATCH	Remote wakeup packet is detected and PMT interrupt is generated
1	1	1	Multicast	MISMATCH	Remote wakeup packet is detected and PMT interrupt is generated

Note: In all other combinations, the remote wakeup packet is not detected and PMT interrupt is not generated.

27.5.6 DMA Register

This section defines the bits for each DMA register. The write data input to the DMA registers are qualified with the corresponding byte enable signal inputs (MCI interface). Therefore, non 32-bit accesses are allowed as long as the address is word aligned. Byte, half word, or word accesses are possible.

DMA registers are in 0x8010_1000 to 0x8010_1058 (GMAC). See section 27.2. Register Map.

27.5.6.1 Bus Mode Register (Bus_Mode)

The Bus Mode register establishes the bus operating modes for the DMA.

27.5.6.2 Transmit Poll Demand Register (Transmit_Poll_Demand)

The Transmit Poll Demand register enables the TX DMA to check whether or not the DMA owns the current descriptor. The Transmit Poll Demand command is given to wake up the TX DMA if it is in the Suspend mode. The TX DMA can go into the Suspend mode because of an Underflow error in a transmitted frame or the unavailability of descriptors it owns. You can issue this command anytime, and the TX DMA resets this command when it again starts to fetch the current descriptor from host memory. When this register is read, it always returns 0.

27.5.6.3 Receive Poll Demand Register (Receive_Poll_Demand)

The Receive Poll Demand register enables the RX DMA to check for new descriptors. This command is given to wake up the RX DMA from the SUSPEND state. The RX DMA can go into the SUSPEND state only because of the unavailability of descriptors it owns. When this register is read, it always returns 0.

27.5.6.4 Receive Descriptor List Address Register (Receive_Descriptor_List_Address)

The Receive Descriptor List Address register points to the start of the receive descriptor list. The descriptor lists reside in the physical memory space of the host and must be word aligned (for 32-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LS bits low. Writing to this register is permitted only when reception is stopped. When stopped, this register must be written to before the receive Start command is given.

You can write to this register only when RX DMA has stopped, that is, bit [1] (SR) is set to 0 in register Operation Mode Register (Operation_Mode). When stopped, this register can be written with a new descriptor list address. When the SR bit is set to 1, the DMA takes the newly programmed descriptor base address.

If this register is not changed when the SR bit is set to 0, then the DMA takes the descriptor address where it was stopped earlier.

27.5.6.5 Transmit Descriptor List Address Register (Transmit_Descriptor_List_Address)

The Transmit Descriptor List Address register points to the start of the transmit descriptor list. The descriptor lists reside in the physical memory space of the host and must be word aligned (for 32-bit data bus). The DMA internally converts it to bus width aligned address by making the corresponding LSB bit to low.

You can write to this register only when the TX DMA has stopped, that is, bit [13] (ST) is set to 0 in Operation Mode Register (Operation_Mode). When stopped, this register can be written with a new descriptor list address.

When the ST bit is set to 1, the DMA takes the newly programmed descriptor base address.

If this register is not changed when the ST bit is set to 0, then the DMA takes the descriptor address where it was stopped earlier.

27.5.6.6 Status Register (Status)

The Status register contains all status bits that the DMA reports to the host. The software driver reads this register during an interrupt service routine or polling. Most of the fields in this register cause the host to be interrupted. The bits of this register are not cleared when read. Writing 1 to (unreserved) bits [16:0] of this register clears these bits and writing 0 has no effect. Each field (bits [16:0]) can be masked by masking the appropriate bit in Interrupt Enable Register (Interrupt_Enable).

27.5.6.7 Operation Mode Register (Operation_Mode)

The Operation Mode register establishes the Transmit and Receive operating modes and commands. This register should be the last CSR to be written as part of the DMA initialization.

27.5.6.8 Interrupt Enable Register (Interrupt_Enable)

The Interrupt Enable register enables the interrupts reported by Status register (Status). Setting a bit to 1 enables a corresponding interrupt. After a hardware reset or software reset, all interrupts are disabled.

27.5.6.9 Missed Frame and Buffer Overflow Counter Register (Missed_Frame_And_Buffer_Overflow_Counter)

The DMA maintains two counters to track the number of frames missed during reception. This register reports the current value of the counter. The counter is used for diagnostic purposes. Bits [15:0] indicate missed frames because of the host

buffer being unavailable. Bits [27:17] indicate missed frames because of buffer overflow conditions (MTL and MAC) and runt frames (good frames of less than 64 bytes) dropped by the MTL.

27.5.6.10 Receive Interrupt Watchdog Timer Register (Receive_Interrupt_Watchdog_Timer)

The Receive Interrupt Watchdog Timer register, when written with a non-zero value, enables the watchdog timer for the Receive Interrupt bit [6] of Status Register (Status).

27.5.6.11 AXI Bus Mode Register (AXI_Bus_Mode)

The AXI Bus Mode Register controls the behavior of the AXI master. It is mainly used to control the burst splitting and the number of outstanding requests.

27.5.6.12 AXI Status Register (AXI_Status)

The AXI Status register provides the active status of the read and write channels of the AXI master interface. This register is useful for debugging purposes.

27.5.6.13 Current Host Transmit Descriptor Register (Current_Host_Transmit_Descriptor)

The Current Host Transmit Descriptor register points to the start address of the current transmit descriptor read by the DMA.

27.5.6.14 Current Host Receive Descriptor Register (Current_Host_Receive_Descriptor)

The Current Host Receive Descriptor register points to the start address of the current receive descriptor read by the DMA.

27.5.6.15 Current Host Transmit Buffer Address Register (Current_Host_Transmit_Buffer_Address)

The Current Host Transmit Buffer Address register points to the current transmit buffer address read by the DMA.

27.5.6.16 Current Host Receive Buffer Address Register (Current_Host_Receive_Buffer_Address)

The Current Host Receive Buffer Address register points to the current receive buffer address read by the DMA.

27.5.6.17 HW Feature Register (HW_Feature)

The HW Feature register indicates the presence of the optional features or functions of the GMAC. The software driver can use this register to dynamically enable or disable programs related to the optional blocks.

27.5.7 MAC Register

MAC registers are in 0x8010_0000 to 0x8010_080C (GMAC). See section 27.2. Register Map.

27.5.7.1 MAC Configuration Register (MAC_Configuration)

The MAC Configuration register establishes receive and transmit operating modes.

Table 27.21 shows how the settings of bit [27] and bit [20] of the MAC Configuration register (MAC_Configuration) impact the giant frame status.

Table 27.21 Giant frame status based on bit [27] and bit [20] (1 of 2)

Length / type field	Received frame length	Bit [27] (TWOKPE)	Bit [20] (JE)	Giant frame status
Untagged packet	> 1,518	0	0	1
	> 2,000	1	0	1
	> 9,018	x	1	1

Table 27.21 Giant frame status based on bit [27] and bit [20] (2 of 2)

Length / type field	Received frame length	Bit [27] (TWOKPE)	Bit [20] (JE)	Giant frame status
VLAN tagged packet	> 1,522	0	0	1
	> 2,000	1	0	1
	> 9,022	x	1	1

Note: For all other combinations, the Giant frame status is 0.

Table 27.22 shows how the settings of bit [7] and bit [25] of the MAC Configuration register (MAC Configuration) impact whether CRC length is included in the frame length.

Note: If Type/Length Field < 1,536, the CST field has no effect. Similarly, if Type/Length field > 1,536, the ACS field has no effect.

Table 27.22 Frame length based on bit [7] and bit [25]

Receive checksum offload engine	Received frame length	Bit [7] (ACS)	Bit [25] (CST)	FCS stripping done
COE Type 2	< 1,536	0	x	No
		1	x	Yes (for Ethernet frames)
	≥ 1,536	x	0	No
		x	1	Yes (for Type frames)

27.5.7.2 MAC Frame Filter Register (MAC_Frame_Filter)

The MAC Frame Filter register contains the filter controls for receiving frames. Some of the controls from this register go to the address check block of the GMAC, which performs the first level of address filtering. The second level of filtering is performed on the incoming frame, based on other controls such as Pass Bad Frames and Pass Control Frames.

27.5.7.3 GMII Address Register (GMII_Address)

The GMII Address register controls the management cycles to the external PHY through the management interface.

27.5.7.4 GMII Data Register (GMII_Data)

The GMII Data register stores write data to be written to the PHY register located at the address specified in GMII Address Register (GMII_Address). This register also stores the read data from the PHY register located at the address specified by GMII_Address.

27.5.7.5 Flow Control Register (Flow_Control)

The Flow Control register controls the generation and reception of the Control (Pause Command) frames by the Flow control module of GMAC. A write to a register with the Busy bit set to 1 triggers the Flow control block to generate a Pause frame. The fields of the control frame are selected as specified in the IEEE 802.3x specification, and the pause time value from this register is used in the Pause Time field of the control frame. The Busy bit remains set until the control frame is transferred onto the cable. The host must make sure that the Busy bit is cleared before writing to the register.

27.5.7.6 VLAN Tag Register (VLAN_Tag)

The VLAN Tag register contains the IEEE 802.1Q VLAN Tag to identify the VLAN frames. The GMAC compares the 13th and 14th bytes of the receiving frame (Length/Type) with 0x8100, and the following two bytes are compared with the VLAN tag. If a match occurs, the GMAC sets the received VLAN bit in the receive frame status. The legal length of the frame is increased from 1,518 bytes to 1,522 bytes.

Consecutive writes to these registers should be performed only after at least 4 clock cycles in the destination clock domain.

27.5.7.7 Version Register (Version)

The Version registers identifies the version of the GMAC.

27.5.7.8 Debug Register (Debug)

The Debug register provides the status of all main modules of the transmit and receive data paths and the FIFOs. An all zero status indicates that the GMAC is in idle state (and FIFOs are empty) and no activity is going on in the data paths.

Note: The reset values given for the Debug register are valid only if clocks are present during reset operation.

27.5.7.9 LPI Control and Status Register (LPI_Control_Status)

The LPI Control and Status Register controls the LPI functions and provides the LPI interrupt status. The status bits are cleared when this register is read.

27.5.7.10 LPI Timers Control Register (LPI_Timers_Control)

The LPI Timers Control register controls the timeout values in the LPI states. It specifies the time for which the GMAC transmits the LPI pattern and also the time for which the GMAC waits before resuming the normal transmission.

27.5.7.11 Interrupt Status Register (Interrupt_Status)

The Interrupt Status register identifies the events in the GMAC that can generate interrupt.

27.5.7.12 Interrupt Mask Register (Interrupt_Mask)

The Interrupt Mask register bits enable you to mask the interrupt signal because of the corresponding event in the Interrupt Status register.

27.5.7.13 MAC Address0 High Register (MAR0_H)

The MAC Address0 High register holds the upper 16 bits of the first 6 bytes MAC address of the station. The first DA byte that is received on the (G)MII interface corresponds to the LS byte (bits [7:0]) of the MAC Address Low register. For example, if 0x112233445566 is received (0x11 in lane 0 of the first column) on the (G)MII as the destination address, then the MAC Address0 Register [47:0] is compared with 0x665544332211.

The synchronization is triggered only when bits [31:24] of the MAC Address0 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low register should be performed after at least 4 clock cycles in the destination clock domain.

27.5.7.14 MAC Address0 Low Register (MAR0_L)

The MAC Address0 Low register holds the lower 32 bits of the 6 bytes first MAC address of the station.

27.5.7.15 MAC Address1 High Register (MAR1_H)

The MAC Address1 High register holds the upper 16 bits of the second 6 bytes MAC address of the station.

The synchronization is triggered only when bits [31:24] of the MAC Address1 Low Register are written. For proper synchronization updates, the consecutive writes to this Address Low register should be performed after at least 4 clock cycles in the destination clock domain.

27.5.7.16 MAC Address1 Low Register (MAR1_L)

The MAC Address1 Low register holds the lower 32 bits of the second 6 bytes MAC address of the station.

Note:
- The descriptions for MAC Address2 High register through MAC Address15 High register are the same as for the MAC Address1 High register.
- The descriptions for MAC Address2 Low register through MAC Address15 Low register are the same as for the MAC Address1 Low register.
- The descriptions for registers MAC Address16 High register through MAC Address17 High register are the same as for the MAC Address1 High register.
- The descriptions for registers MAC Address16 Low register through MAC Address17 Low register are the same as for the MAC Address1 Low register.

27.5.7.17 Watchdog Timeout Register (WDog_Timeout)

The Watchdog Timeout register controls the watchdog timeout for received frames.

27.5.7.18 Hash Table Register 0 (HASH_TABLE_REG0)

The Hash Table register contains the first 32 bits of the hash table.

The 256-bit hash table is used for group address filtering. For hash filtering, the content of the destination address in the incoming frame is passed through the CRC logic and the upper 8 bits of the CRC register are used to index the content of the hash table. The most significant bit determines the register to be used (Hash Table Register n), and the least significant 5 bits determine the bit within the register. For example, a hash value of 10111111b selects bit [31] of the Hash Table Register 5.

The hash value of the destination address is calculated in the following way:

1. Calculate the 32-bit CRC for the DA (see IEEE 802.3, Section 3.2.8 for the steps to calculate CRC32).
2. Perform bitwise reversal for the value obtained in Step 1.
3. Take the upper 7 (or 8) bits from the value obtained in Step 2.

If the corresponding bit value of the register is 1, the frame is accepted. Otherwise, it is rejected. If the bit [1] (Pass All Multicast) is set in MAC Frame Filter Register (MAC_Frame_Filter), then all multicast frames are accepted regardless of the multicast hash values.

The synchronization is triggered only when bits [31:24] of the Hash Table Register X registers are written.

Note:
- Consecutive writes to this register should be performed after at least 4 clock cycles in the destination clock domain.
- Hash Table Register 1 through 7 are similar to Hash Table Register 0.

27.5.7.19 VLAN Hash Table Register (VLAN_Hash_Table_Reg)

The 16-bit hash table is used for group address filtering based on VLAN tag when bit [19] (VTHM) of VLAN Tag Register (VLAN_Tag) is set. For hash filtering, the content of the 16-bit VLAN tag or 12-bit VLAN Identifier (based on bit [16] (ETV) of VLAN Tag Register) in the incoming frame is passed through the CRC logic and the upper 4 bits of the calculated CRC are used to index the contents of the VLAN hash table. For example, a hash value of 1000b selects bit [8] of the VLAN hash table.

The hash value of the destination address is calculated in the following way:

1. Calculate the 32-bit CRC for the VLAN tag or ID (see IEEE 802.3, Section 3.2.8 for the steps to calculate CRC32).
2. Perform bitwise reversal for the value obtained in Step 1.
3. Take the upper four bits from the value obtained in Step 2.

If the corresponding bit value of the register is 1, the frame is accepted. Otherwise, it is rejected. The synchronization is triggered only when bits [15:8] of this register are written.

Note: Consecutive writes to this register should be performed after at least 4 clock cycles in the destination clock domain.

27.5.8 IEEE 1588 Timestamp Registers

This section describes the updated registers required to support the IEEE 1588 functions.

IEEE 1588 Timestamp registers are in 0x8010_0700 to 0x8010_0734 (GMAC). See section 27.2. Register Map.

27.5.8.1 Timestamp Control Register (Timestamp_Control)

The Timestamp Control register controls the operation of the system time generator and the processing of PTP packets for timestamping in the receiver.

Note: Table 27.23 indicates the PTP messages, for which a snapshot is taken depending on bits [17:14] (SNAPTYPSEL[1:0], TSMSTRENA, TSEVNTENA) in the Timestamp Control register (Timestamp_Control).

Table 27.23 Timestamp snapshot dependency on register bits

SNAPTYPSEL[1:0]	TSMSTRENA	TSEVNTENA	PTP messages
00b	x	0	SYNC, Follow_Up, Delay_Req, Delay_Resp
00b	0	1	SYNC
00b	1	1	Delay_Req
01b	x	0	SYNC, Follow_Up, Delay_Req, Delay_Resp, Pdelay_Req, Pdelay_Resp, Pdelay_Resp_Follow_Up
01b	0	1	SYNC, Pdelay_Req, Pdelay_Resp
01b	1	1	Delay_Req, Pdelay_Req, Pdelay_Resp
10b	x	x	SYNC, Delay_Req
11b	x	x	Pdelay_Req, Pdelay_Resp

27.5.8.2 Sub Second Increment Register (Sub_Second_Increment)

This register is present only when the IEEE 1588 timestamp feature is selected without an external timestamp input. In the Coarse Update mode (TSCFUPDT bit in Timestamp Control Register (Timestamp_Control)), the value in this register is added to the system time every clock cycle of GMAC_PTP_REFCLK_I. In the Fine Update mode, the value in this register is added to the system time whenever the Accumulator gets an overflow.

27.5.8.3 System Time – Seconds Register, Nanoseconds Register (System_Time_Seconds, System_Time_Nanoseconds)

The System Time Seconds register, along with System Time Nanoseconds register, indicates the current value of the system time maintained by the GMAC. Though it is updated on a continuous basis, there is some delay from the actual time because of clock domain transfer latencies (from GMAC_PTP_REFCLK_I to CSR clock (GMAC[1]_HCLK)).

These registers (System Time – Seconds and System Time – Nanoseconds) are present only when the IEEE 1588 Timestamp feature is selected without external timestamp input.

27.5.8.4 System Time – Seconds Update Register, Nanoseconds Update Register (System_Time_Seconds_Update, System_Time_Nanoseconds_Update)

The System Time Seconds Update register, along with the System Time Nanoseconds Update register, initializes or updates the system time maintained by the GMAC. You must write both of these registers before setting the TSINIT or TSUPDT bits in the Timestamp Control register.

27.5.8.5 Timestamp Addend Register (Timestamp_Addend)

This register is present only when the IEEE 1588 Timestamp feature is selected without external timestamp input. This register value is used only when the system time is configured for Fine Update mode (TSCFUPDT bit in Timestamp Control Register (Timestamp_Control)). This register content is added to a 32-bit accumulator in every clock cycle (of GMAC_PTP_REFCLK_I) and the system time is updated whenever the accumulator overflows.

27.5.8.6 Target Time Seconds Register, Target Time Nanoseconds Register (Target_Time_Seconds, Target_Time_Nanoseconds)

The Target Time Seconds register, along with Target Time Nanoseconds register, is used to schedule an interrupt event (Timestamp_Status[1] when Advanced Timestamping is enabled; otherwise, TSIS interrupt bit in Interrupt_Status[9]) when the system time exceeds the value programmed in these registers.

27.5.8.7 Timestamp Status Register (Timestamp_Status)

All bits except bits [27:25] gets cleared when the host reads the Timestamp Status register.

27.5.8.8 Auxiliary Timestamp—Nanoseconds Register (Auxiliary_Timestamp_Nanoseconds)

The Auxiliary Timestamp – Nanoseconds register, along with Auxiliary Timestamp – Seconds register (Auxiliary_Timestamp_Seconds), gives the 64-bit timestamp stored as auxiliary snapshot. The two registers together form the read port of a 64-bit width FIFO with a depth of 4. Multiple snapshots can be stored in this FIFO. The ATSNS[4:0] bits in the Timestamp Status register indicate the fill level of this FIFO. The top of the FIFO is removed only when the last byte of Auxiliary Timestamp – Seconds Register is read. This means when bits [31:24] are read.

27.5.9 Known Issue and Workaround

There are known issues in GMAC. Please consider workaround when using related function.

Table 27.24 Known issue and workaround

No.	Issue	Workaround
1	When the Receive FIFO operates in the cut-through mode, the Receive FIFO Read Controller may stop functioning. If this issue happens, software reset is needed to recover functioning.	Operate Receive FIFO in store and forward mode by setting bit 25 (RSF) in operation mode register (Operation_Mode) to 1.
2	Filter i Command in Remote Wake-Up Frame Filter Register (wkupfmfilter_reg[n] n = 0 to 7) is a 4-bit read-write field. When reading it, bit 1 (And_Previous) and bit 2 (INV) always return zero.	Do not use value read from this field. Software should maintain a copy of the value written to this field and use that instead of reading it.
3	Even if bit 3 (CAST) of Filter i Command in Remote Wake-Up Frame Filter Register (wkupfmfilter_reg[n] n = 0 to 7) is set to 0 to process packet with unicast address, if the packet with broadcast address is received, it is also processed to determine whether it is remote wake up packet or not. If CRC matches, it is regarded as remote wake up packet incorrectly.	Use packet with multicast address for remote wake up packet and set CAST to 1. Otherwise, when using packet with unicast address, do not use it which match the condition of packet with broadcast address.
4	When bit 1 (And_Previous) of Filter i Command in Remote Wake-Up Frame Filter Register (wkupfmfilter_reg[n] n = 0 to 7) is used to chain multiple filters, PMT interrupt is generated incorrectly if individual filters match different packets. (For example, the first packet received matches filter 0 only and then the second packet received matches filter 1 only.)	Do not use And_Previous and set it to 0. Otherwise, do not use remote wake up packet which matches defect condition.
5	When LPITXA and LPIEN bits in LPI Control and Status register (LPI_Control_Status) are set to 1, MAC transmitter may enter LPI state prematurely if TX MAC is idle but TX DMA is still active.	Set the LPITXA and LPIEN bits only after ensuring that TX DMA is in IDLE state. When bits [22:20] (TS[2:0]) in the status register (Status) indicate that the TX DMA is in Stopped or Suspend state, the TX DMA is in IDLE.

28. EtherCAT Slave Controller (ESC)

28.1 Overview

The EtherCAT slave controller (ESC) uses an EtherCAT Slave Controller IP Core made by Beckhoff Automation GmbH, Germany. The ESC handles EtherCAT communications as an interface between the EtherCAT field bus and slave applications.

Table 28.1 Specification of the EtherCAT slave controller

Item		Description
Number of ports		3
FMMU		8
SyncManager		8
Process data RAM [KB]		8
Distributed clocks		64 bits
EBus		No
Process data interfaces (PDI)	Digital I/O	No
	SPI slave	No
	Host MPU interface	On-chip bus

Figure 28.1 shows a block diagram of the EtherCAT slave controller.

This section describes the function of the registers and modules in white. The registers and modules in gray are described in section 26, Ethernet Subsystem, section 27, Ethernet MAC (GMAC), and DEADBEEF. Refer to the related sections regarding the registers and modules which are needed to operate the EtherCAT slave controller.

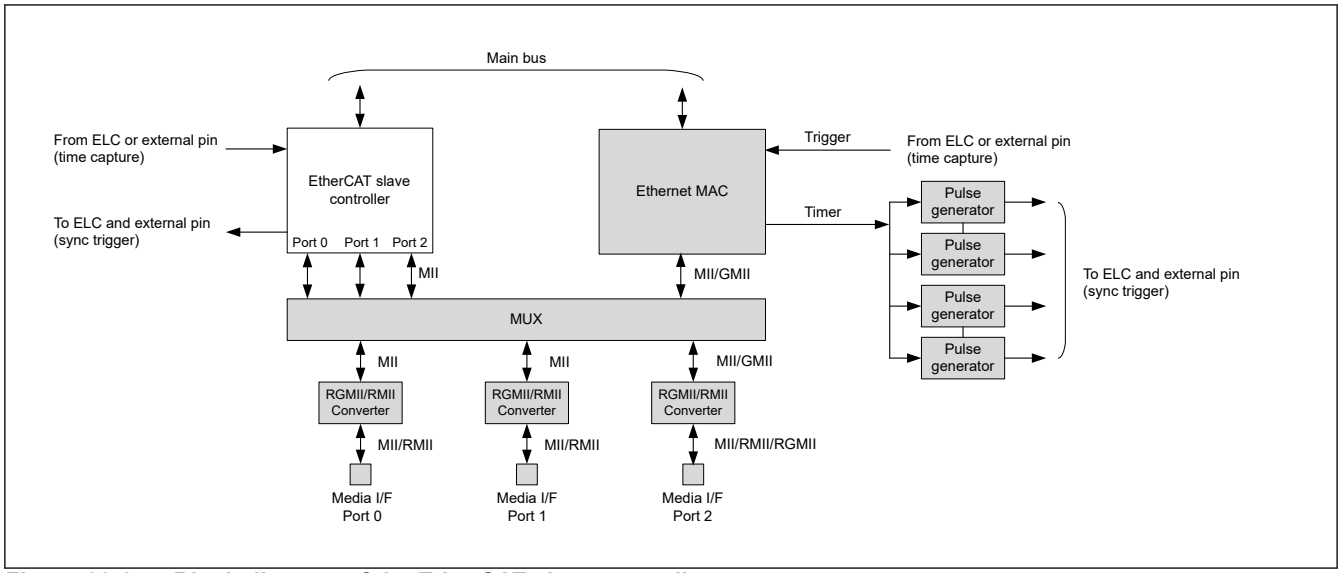


Figure 28.1 Block diagram of the EtherCAT slave controller

Table 28.2 lists the input/output pins of the EtherCAT controller.

Table 28.2 Input/Output pins of the EtherCAT slave controller (excluding PHY MII pins) (1 of 2)

Pin name	I/O	Function
ESC_LEDRUN	O	EtherCAT RUN LED signal output pin
ESC_IRQ	O	EtherCAT IRQ signal output pin
ESC_LEDSTER	O	EtherCAT Dual-color State LED signal output pin
ESC_LEDERR	O	EtherCAT Error LED signal output pin

Table 28.2 Input/Output pins of the EtherCAT slave controller (excluding PHY MII pins) (2 of 2)

Pin name	I/O	Function
ESC_LINKACT0	O	EtherCAT link/Activity LED signal output pin (port 0)
ESC_LINKACT1	O	EtherCAT link/Activity LED signal output pin (port 1)
ESC_LINKACT2	O	EtherCAT link/Activity LED signal output pin (port 2)
ESC_SYNC0	O	EtherCAT SYNC0 signal output pin
ESC_SYNC1	O	EtherCAT SYNC1 signal output pin
ESC_LATCH0	I	EtherCAT LATCH0 signal input pin
ESC_LATCH1	I	EtherCAT LATCH1 signal input pin
ESC_RESETOUT#	O	EtherCAT reset output pin
ESC_I2CCLK	O	EtherCAT EEPROM I2C clock output pin
ESC_I2CDATA	I/O	EtherCAT EEPROM I2C data signal I/O pin
ESC_PHYLINK0	I	EtherCAT PHY0 link status signal input pin
ESC_PHYLINK1	I	EtherCAT PHY1 link status signal input pin
ESC_PHYLINK2	I	EtherCAT PHY2 link status signal input pin
ESC_MDC	O	Management data clock output pin
ESC_MDIO	I/O	Management data I/O pin

28.2 Functional Overview

Typical functions of the EtherCAT slave controller and supported functions by this LSI are shown in Table 28.3. Regarding the detailed specification of EtherCAT and ESC, refer to the documentation (ETG.1000 EtherCAT Specification) provided by EtherCAT Technology Group (ETG) and the EtherCAT Slave Controller IP Core (v2.04) data sheet provided by Beckhoff Automation.

Table 28.3 Typical functions of EtherCAT slave controller and supported functions by this LSI (1 of 3)

Features	Functions	Support
EtherCAT protocol	Handling the following frames: ● Ethernet frames with Ether type 0x88A4 ● EtherCAT frames encapsulated in UDP/IP ● EtherCAT frames with VLAN tag ● Normal Ethernet frames	✓
Addressing modes	Device addressing ● Auto increment address ● Configured station address ● Broadcast address	✓
	Logical addressing	✓
Working counter	Counting the number of read/write from/to the device	✓
EtherCAT command type	Processing the command that master requests slaves to address each addressing mode	✓
Loop control	Loop control and loop state in ESC	✓
Shadow buffer	Shadow buffers function when register is read/written	✓
Circulating frames	Processing of circulating frames during the failure	✓
Link detection	Link MII signal (PHY link signal) (monitoring the PHY register through the management interface)	✓
	MI Link detection and configuration	—
	Enhanced link detection (monitoring the state of transfer by MII RX error monitor)	✓
FIFO size reduction	RX FIFO size reduction because of reduction of propagation delay	✓

Table 28.3 Typical functions of EtherCAT slave controller and supported functions by this LSI (2 of 3)

Features	Functions	Support
Ethernet physical layer	MII	✓
	EBUS	—
	Back-to-Back MII connection	✓
	MII management interface	✓
	Read/write of the PHY register through MII management interface	✓
	PHY address offset	✓
	Manual TX clock shift compensation	✓
	Automatic TX clock shift compensation	✓
FMMU	Mapping between logical address and physical address	✓
SyncManager	Buffer mode	✓
	Mailbox mode	✓
	Interrupt and latch event generation when a buffer was completely and successfully written or read.	✓
	Repeating mailbox communication	✓
	SyncManager deactivation by the PDI	✓
Distributed clocks	Clock synchronization considering propagation delay and drift compensation	✓
	Generation of synchronous output signals (SYNC0 and 1 signals) ● Cyclic mode ● Single shot mode ● Cyclic acknowledge mode ● Single shot acknowledge mode	✓
	Precise time stamping of input events (LATCH0 and 1 signals) ● Single event mode ● Continuous mode ● SyncManager event mode (for debugging)	✓
	Generation of synchronous interrupts	✓
	Synchronous digital output updates / Synchronous digital input sampling	—
	Exclusive control for the SYNC and LATCH signals of the ECAT and PDI	✓
	System time control by the PDI	—
	Communication Timing ● Free run ● Synchronized to output event ● Synchronized to SYNC signal	✓
EtherCAT state machine	Control of state machine/indication of the status and error code	✓
	Device emulation	—
SII EEPROM	SII EEPROM commands	✓
	SII EEPROM error indication	✓
	SII EEPROM access interface	✓
	EEPROM size selection	✓
	EEPROM emulation	—
Interrupt	AL event request (PDI interrupt)	✓
	ECAT event request (ECAT interrupt)	✓
Watchdog	Process data watchdog	✓
	PDI watchdog	✓

Table 28.3 Typical functions of EtherCAT slave controller and supported functions by this LSI (3 of 3)

Features	Functions	Support
Error counters	Port error counters	✓
	Forwarded RX error counter	✓
	ECAT processing unit error counter	✓
	PDI error counter	✓
	Lost link counter	✓
	Watchdog counter process data	✓
	Watchdog counter PDI	✓
LED signals	RUN LED signal	✓
	ERR LED signal	✓
	STATE LED and STATE_RUN LED signals	✓
	LINK/ACT LED signals	✓
	Port error LED signal	—
	RUN/ERR LED override	✓
Process data interface (PDI)	Digital I/O	—
	SPI slave	—
	8-bit/16-bit synchronous/asynchronous microcontroller interface	—
	On-chip bus	✓
	General purpose I/O	—
Write protection	Write protection for the register area (0x0000 to 0x0FFF)	✓
	Write protection for the whole area including the user RAM and process data RAM (0x0000 to 0x2FFF)	✓
ESC reset	ESC reset from the master or PDI	✓

28.3 Description of Registers

28.3.1 ECATOFFADR : EtherCAT PHY Offset Address Setting Register

Base address: ESC_INI = 0x8011_0200

Offset address: 0x0_0000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	OADD[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	OADD[4:0]	PHY Offset Address Setting Set the offset address of PHY of using the EtherCAT.	R/W
31:5	—	The read values are undefined. The write value should be 0.	R/W

Note: This register can only be written when protection is unlocked by the specific sequence using the Ethernet Protect register (PRCMD). For the protection unlock procedure, see section 26.2.1. PRCMD : Ethernet Protect Register. The special sequence is not necessary when reading the value of this register.

The ECATOFFADR register sets the offset address of the Ethernet PHY in case of using the EtherCAT. This register can be read/written in 32-bit units.

28.3.2 ECATOPMOD : EtherCAT Operation Mode Register

Base address: ESC_INI = 0x8011_0200

Offset address: 0x0_0004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EEPROMSIZE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EEPROMSIZE	EEPROM Memory Size Specification Sets the size of the EEPROM memory of the EtherCAT. 0: 16 Kbits or less 1: 32 Kbits to 4 Mbits	R/W
31:1	—	The read values are undefined. The write value should be 0.	R/W

Note: This register can only be written when protection is unlocked by the specific sequence using the Ethernet Protect register (PRCMD). For the protection unlock procedure, see section 26.2.1. PRCMD : Ethernet Protect Register. The special sequence is not necessary when reading the value of this register

The ECATOPMOD register sets the EEPROM memory size in case of using the EtherCAT. This register can be read/written in 32-bit units.

28.3.3 ECATDBGC : EtherCAT Debug Control Register

Base address: ESC_INI = 0x8011_0200

Offset address: 0x0_0008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	TXSFT2[1:0]		TXSFT1[1:0]		TXSFT0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	TXSFT0[1:0]	Set the delay time for ETH0_TXC of the EtherCAT 0 0: 0 ns 0 1: 10 ns 1 0: 20 ns 1 1: 30 ns	R/W
3:2	TXSFT1[1:0]	Set the delay time for ETH1_TXC of the EtherCAT 0 0: 0 ns 0 1: 10 ns 1 0: 20 ns 1 1: 30 ns	R/W

Bit	Symbol	Function	R/W
5:4	TXSFT2[1:0]	Set the delay time for ETH2_TXC of the EtherCAT 0 0: 0 ns 0 1: 10 ns 1 0: 20 ns 1 1: 30 ns	R/W
31:6	—	The read values are undefined. The write value should be 0.	R/W

Note: This register can only be written when protection is unlocked by the specific sequence using the Ethernet Protect register (PRCMD). For the protection unlock procedure, see section 26.2.1. PRCMD : Ethernet Protect Register. The special sequence is not necessary when reading the value of this register.

The ECATDBGC register controls the delay time of TXC. This register can be read/written in 32-bit units.

28.3.4 ECATTRGSEL : EtherCAT DC Latch Trigger Select Register

Base address: ESC_INI = 0x8011_0200

Offset address: 0x0_000C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TRGSEL1	TRGSEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TRGSEL0	Select DC Latch Trigger 0 for ESC 0: Select LATCH0 input from ELC 1: Select LATCH0 input from external	R/W
1	TRGSEL1	Select DC Latch Trigger 1 for ESC 0: Select LATCH1 input from ELC 1: Select LATCH1 input from external	R/W
31:2	—	The read values are undefined. The write value should be 0.	R/W

Note: This register can only be written when protection is unlocked by the specific sequence using the Ethernet Protect register (PRCMD). For the protection unlock procedure, see section 26.2.1. PRCMD : Ethernet Protect Register. The special sequence is not necessary when reading the value of this register.

The ECATTRGSEL register selects the source to LATCHn (n = 0, 1) inputs to ESC. This register can be read/written in 32-bit units.

28.4 ESC Information Register

In the tables, ECAT indicates access by the EtherCAT master controller and PDI indicates access by the local CPU.

28.4.1 TYPE : Type Register

Base address: ESC = 0x8013_0000

Offset address: 0x0000

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TYPE[7:0]							
Value after reset:	1	0	1	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TYPE[7:0]	Type of the EtherCAT slave controller	R

The TYPE register indicates the type of the EtherCAT slave controller.

28.4.2 REVISION : Revision Register

Base address: ESC = 0x8013_0000

Offset address: 0x0001

Bit position:	7	6	5	4	3	2	1	0
Bit field:	REV[7:0]							
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
7:0	REV[7:0]	Revision of the EtherCAT slave controller	R

The REVISION register indicates the revision of the EtherCAT slave controller.

28.4.3 BUILD : Build Register

Base address: ESC = 0x8013_0000

Offset address: 0x0002

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BUILD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	BUILD[15:0]	Build number of the EtherCAT slave controller	R

The BUILD register indicates the build number of the EtherCAT slave controller.

28.4.4 FMMU_NUM : FMMU Supported Register

Base address: ESC = 0x8013_0000

Offset address: 0x0004

Bit position:	7	6	5	4	3	2	1	0
Bit field:	NUMFMMU[7:0]							
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
7:0	NUMFMMU[7:0]	Number of FMMU channels supported in the EtherCAT slave controller	R

The FMMU_NUM register indicates the number of FMMU channels supported in the EtherCAT slave controller.

28.4.5 SYNC_MANAGER : SyncManager Supported Register

Base address: ESC = 0x8013_0000

Offset address: 0x0005

Bit position:	7	6	5	4	3	2	1	0
Bit field:	NUMSYNC[7:0]							
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
7:0	NUMSYNC[7:0]	Number of SyncManager channels supported in the EtherCAT slave controller	R

The SYNC_MANAGER register indicates the number of SyncManager channels supported in the EtherCAT slave controller.

28.4.6 RAM_SIZE : RAM Size Register

Base address: ESC = 0x8013_0000

Offset address: 0x0006

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RAMSIZE[7:0]							
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
7:0	RAMSIZE[7:0]	Process data RAM size supported in the EtherCAT slave controller (unit: KB)	R

The RAM_SIZE register indicates the process data RAM size supported in the EtherCAT slave controller in KB.

28.4.7 PORT_DESC : Port Descriptor Register

Base address: ESC = 0x8013_0000

Offset address: 0x0007

Bit position:	7	6	5	4	3	2	1	0
Bit field:	P3[1:0]		P2[1:0]		P1[1:0]		P0[1:0]	
Value after reset:	0	0	1	1	1	1	1	1

Bit	Symbol	Function	R/W
1:0	P0[1:0]	Port 0 configuration Fixed to the setting for MII connection (11b) in this LSI. 0 0: Not implemented 0 1: Not configured (SII EEPROM) 1 0: EBUS 1 1: MII	R
3:2	P1[1:0]	Port 1 configuration Fixed to the setting for MII connection (11b) in this LSI. 0 0: Not implemented 0 1: Not configured (SII EEPROM) 1 0: EBUS 1 1: MII	R

Bit	Symbol	Function	R/W
5:4	P2[1:0]	Port 2 configuration Fixed to the setting for MII connection (11b) in this LSI. 0 0: Not implemented 0 1: Not configured (SII EEPROM) 1 0: EBUS 1 1: MII	R
7:6	P3[1:0]	Port 3 configuration This LSI does not implement port 3. Fixed to 00b. 0 0: Not implemented 0 1: Not configured (SII EEPROM) 1 0: EBUS 1 1: MII	R

The PORT_DESC register indicates the port configuration.

28.4.8 FEATURE : ESC Features Supported Register

Base address: ESC = 0x8013_0000

Offset address: 0x0008

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	FSCONFIG	RWSUPP	LRW	DCSYNC	FCS	LINKDECMII	—	—	DCWID	DC	—	FMMU
Value after reset:	x	x	x	x	0	0	0	1	1	1	x	x	1	1	x	0

Bit	Symbol	Function	R/W
0	FMMU	FMMU Operation 0: Bit oriented 1: Byte oriented	R
1	—	The read value is undefined.	R
2	DC	Distributed Clock 0: Not available 1: Available	R
3	DCWID	Distributed Clock Width 0: 32 bits 1: 64 bits	R
5:4	—	The read values are undefined.	R
6	LINKDECMII	Enhanced Link Detection in MII 0: Not available 1: Available	R
7	FCS	Separate handling of FCS errors 0: Not supported 1: Supported. Frames with wrong FCS and additional nibble will be counted separately in forwarded RX error counter.	R
8	DCSYNC	Enhanced DC SYNC activation 0: Not available 1: Available	R
9	LRW	EtherCAT LRW command support 0: Supported 1: Not supported	R
10	RWSUPP	EtherCAT read/write command support (BRW, APRW, FPRW) 0: Supported 1: Not supported	R
11	FSCONFIG	Fixed FMMU/SyncManager configuration 0: Variable configuration 1: Fixed configuration	R

Bit	Symbol	Function	R/W
15:12	—	The read values are undefined.	R

The FEATURE register indicates the features supported in the EtherCAT slave controller.

28.5 Station Address Registers

28.5.1 STATION_ADR : Configured Station Address Register

Base address: ESC = 0x8013_0000

Offset address: 0x0010

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	NODADDR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	NODADDR[15:0]	Node Addressing Address Indication Address used for node addressing (FPxx commands) The access type from the EtherCAT master is R/W.	R

The STATION_ADR register indicates the address used for node addressing.

28.5.2 STATION_ALIAS : Configured Station Alias Register

Base address: ESC = 0x8013_0000

Offset address: 0x0012

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	NODALIADDR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	NODALIADDR[15:0]	Alias Address Indication Alias address used for node addressing (FPxx commands). The use of this alias is activated by setting bit 24 of the ESC DL control register (ESC_DL_CONTROL at 0x0100) to 1. The access type from the EtherCAT master is R.	R/W

Note: The initial value, 0, is retained until the EEPROM is loaded. After that, the value becomes the value at address 0x0004 in the EEPROM. This value is only taken over from the EEPROM the first time the EEPROM is loaded after a power-on or reset.

The STATION_ALIAS register indicates the alias address used for node addressing (FPxx commands).

28.6 Write Protection Registers

28.6.1 WR_REG_ENABLE : Write Register Enable Register

Base address: ESC = 0x8013_0000

Offset address: 0x0020

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ENABLE
Value after reset:	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	ENABLE	Register Write Protection Unlock When registers are currently being protected against writing (bit 0 is 1 in the write register protection register, WR_REG_PROTECT, at 0x0021) and freely writing to registers of the given node is to be permitted, the operation to do so by writing to this register has to proceed in the same Ethernet frame and preceding the other desired writing to registers. Write protection will be reactivated once the frame period elapses (unless the value in the write register protection register is changed). The access type from the EtherCAT master is R/W.	R
7:1	—	The read values are undefined.	R

The WR_REG_ENABLE register is used to unlock the write protection temporarily while registers are write-protected.

28.6.2 WR_REG_PROTECT : Write Register Protection Register

Base address: ESC = 0x8013_0000

Offset address: 0x0021

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	PROTECT
Value after reset:	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	PROTECT	Register Write Protection Specification The access type from the EtherCAT master is R/W. 0: Protection disabled 1: Protection enabled	R
7:1	—	The read values are undefined.	R

The WR_REG_PROTECT register is used to protect registers against writing. The registers in the area 0x8013_0000 to 0x8013_0FFF are write-protected (except for the WR_REG_ENABLE register (0x0020) and ESC_WR_ENABLE register (0x0030)).

28.6.3 ESC_WR_ENABLE : ESC Write Enable Register

Base address: ESC = 0x8013_0000

Offset address: 0x0030

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ENABLE
Value after reset:	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	ENABLE	Register/Memory Write Protection Unlock When registers are currently being protected against writing by ESC write protection (bit 0 is 1 in the ESC write protection register, ESC_WR_PROTECT, at 0x0031) and freely writing to registers of the given node is to be permitted, the operation to do so by writing to this register has to proceed in the same Ethernet frame and preceding the other desired writing to registers. Write protection will be reactivated once the frame period elapses (unless the value in the ESC write protection register is changed). The access type from the EtherCAT master is R/W.	R
7:1	—	The read values are undefined.	R

The ESC_WR_ENABLE register is used to unlock the write protection temporarily while registers and memories are write-protected by ESC write protection.

28.6.4 ESC_WR_PROTECT : ESC Write Protection Register

Base address: ESC = 0x8013_0000

Offset address: 0x0031

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	PROTECT
Value after reset:	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	PROTECT	Register/Memory Write Protection Specification The access type from the EtherCAT master is R/W. 0: Protection disabled 1: Protection enabled	R
7:1	—	The read values are undefined.	R

The ESC_WR_PROTECT register is used to protect registers against writing. Registers and memories in the area 0x8013_0000 to 0x8013_2FFF including the process data RAM are write-protected (except for the WR_REG_ENABLE register (0x0020) and ESC_WR_ENABLE register (0x0030)).

28.7 Data Link Layer Registers

28.7.1 ESC_RESET_ECAT_W : ESC Reset ECAT Register for write

Base address: ESC = 0x8013_0000

Offset address: 0x0040

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RESET_ECAT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RESET_ECAT[7:0]	Software Reset Setting A reset is enabled after writing 0x52 ('R'), 0x45 ('E'), and 0x53 ('S') consecutively to this register. The access type from the EtherCAT master is R/W.	R

This register is used to reset the EtherCAT slave controller from the ECAT (master) by software.

28.7.2 ESC_RESET_ECAT_R : ESC Reset ECAT Register for read

Base address: ESC = 0x8013_0000

Offset address: 0x0040

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	RESET_ECAT[1:0]	
Value after reset:	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
1:0	RESET_ECAT[1:0]	Reset Progress Status The access type from the EtherCAT master is R/W. 0 1: After writing 0x52 1 0: After writing 0x45 (if 0x52 was written before) 0 0: Others	R
7:2	—	The read values are undefined. The access type from the EtherCAT master is R/W. In this case, the write value should be 0.	R

This register is used to reset the EtherCAT slave controller from the ECAT (master) by software.

28.7.3 ESC_RESET_PDI_W : ESC Reset PDI Register for write

Base address: ESC = 0x8013_0000

Offset address: 0x0041

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RESET_PDI[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RESET_PDI[7:0]	Software Reset Setting A reset is enabled after writing 0x52 ('R'), 0x45 ('E'), and 0x53 ('S') consecutively to this register. The access type from the EtherCAT master is R.	R/W

This register is used to reset the EtherCAT slave controller from the PDI (slave) by software.

28.7.4 ESC_RESET_PDI_R : ESC Reset PDI Register for read

Base address: ESC = 0x8013_0000

Offset address: 0x0041

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	RESET_PDI[1:0]	
Value after reset:	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
1:0	RESET_PDI[1:0]	Reset Progress Status The access type from the EtherCAT master is R. 0 1: After writing 0x52 1 0: After writing 0x45 (if 0x52 was written before) 0 0: Others	R/W
7:2	—	The read values are undefined. The write value should be 0.	R/W

This register is used to reset the EtherCAT slave controller from the PDI (slave) by software.

28.7.5 ESC_DL_CONTROL : ESC DL Control Register

Base address: ESC = 0x8013_0000

Offset address: 0x0100

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	STAALIAS	—	—	—	—	—	RXFIFO[2:0]		
Value after reset:	x	x	x	x	x	x	x	0	x	x	x	x	x	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	LP3[1:0]		LP2[1:0]		LP1[1:0]		LP0[1:0]		—	—	—	—	—	—	TEMPUSE	FWDRULE
Value after reset:	1	1	0	0	0	0	0	0	x	x	x	x	0	0	0	1

Bit	Symbol	Function	R/W
0	FWDRULE	Forwarding Rule The source MAC address is changed for every frame (SOURCE_MAC[1] is set to 1 (locally administered address)) regardless of the forwarding rule. The access type from the EtherCAT master is R/W. 0: EtherCAT frames are processed. Non-EtherCAT frames are forwarded without processing. 1: EtherCAT frames are processed. Non-EtherCAT frames are destroyed.	R
1	TEMPUSE	Temporary Use of Bits 15 to 8 Settings The access type from the EtherCAT master is R/W. 0: Permanent use 1: Use for about 1 second, then revert to previous settings	R
3:2	—	These bits are read as 0.	R
7:4	—	The read values are undefined.	R
9:8	LP0[1:0]	Loop Port 0 Configuration The access type from the EtherCAT master is R/W. 0 0: Auto 0 1: Auto close 1 0: Open 1 1: Closed	R
11:10	LP1[1:0]	Loop Port 1 Configuration The access type from the EtherCAT master is R/W. 0 0: Auto 0 1: Auto close 1 0: Open 1 1: Closed	R
13:12	LP2[1:0]	Loop Port 2 Configuration Port 2 is not available on this LSI. The access type from the EtherCAT master is R/W. 0 0: Auto 0 1: Auto close 1 0: Open 1 1: Closed	R
15:14	LP3[1:0]	Loop Port 3 Configuration Port 3 is not available on this LSI. The access type from the EtherCAT master is R/W. 0 0: Auto 0 1: Auto close 1 0: Open 1 1: Closed	R

Bit	Symbol	Function	R/W
18:16	RXFIFO[2:0]	RX FIFO Size Set the RX FIFO size. The transfer time can be reduced by reducing the FIFO size. The access type from the EtherCAT master is R/W. 0 x x: Shortened by 40 ns 1 1 1: Default Others: No change	R
23:19	—	The read values are undefined.	R
24	STAALIAS	Station Alias Status The access type from the EtherCAT master is R/W. 0: Ignore station alias 1: Alias can be used for all configured address command types such as FPRD, FPWR.	R
31:25	—	The read values are undefined.	R

Note:
- Changes to loop configurations are delayed until any current reception or transmission of a frame through the port is completed.
- Reducing the size of the RX FIFO depends on all masters and slaves connected to the same network as the EtherCAT having very precise clock sources. An RX FIFO size of 7 (default) is sufficient if the precision of all clocks is 100 ppm or better and RX FIFO sizes of 0 to 3 are possible if the precision is 25 ppm or better.

The ESC_DL_CONTROL register is used to control loop in the EtherCAT slave controller and configure the RX FIFO size and station alias.

28.7.6 PHYSICAL_RW_OFFSET : Physical Read/Write Offset Register

Base address: ESC = 0x8013_0000

Offset address: 0x0108

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RWOFFSET[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RWOFFSET[15:0]	Offset between Read and Write Addresses Offset of R/W commands (FPRW, APRW) between read address and write address. When read, RD_ADR = ADR (the given address is read) When write, WR_ADR = ADR + R/W offset (writing is to the address obtained by adding the offset set in this register to the given address) The access type from the EtherCAT master is R/W.	R

The PHYSICAL_RW_OFFSET register is used to set the offset between read address and write address in the R/W commands.

28.7.7 ESC_DL_STATUS : ESC DL Status Register

Base address: ESC = 0x8013_0000

Offset address: 0x0110

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	COMP 3	LP3	COMP 2	LP2	COMP 1	LP1	COMP 0	LP0	PHYP 3	PHYP 2	PHYP 1	PHYP 0	—	ENHLI NKD	PDIW DST	PDIOP E
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	x	1	0	0

Bit	Symbol	Function	R/W
0	PDIOPE	PDI/EEPROM Load State Indication 0: EEPROM not loaded, the PDI not operational (process data RAM is not accessible) 1: EEPROM loaded correctly, the PDI operational (process data RAM is accessible)	R

Bit	Symbol	Function	R/W
1	PDIWDST	PDI Watchdog Timer Status 0: Timeout of the watchdog timer 1: Watchdog timer reloaded	R
2	ENHLINKD	Enhanced Link Detection Indication This bit is set to the value of bit 9 at address 0x0000 in the EEPROM the first time the EEPROM is loaded after power is initially supplied or after a reset. 0: Deactivated for all ports 1: Activated for at least one port	R
3	—	The read value is undefined.	R
4	PHYP0	Port 0 Link State Indication Physical link on port 0. 0: No link 1: Link detected	R
5	PHYP1	Port 1 Link State Indication Physical link on port 1. 0: No link 1: Link detected	R
6	PHYP2	Port 2 Link State Indication Port 2 is not available on this LSI. 0: No link 1: Link detected	R
7	PHYP3	Port 3 Link State Indication Port 3 is not available on this LSI. 0: No link 1: Link detected	R
8	LP0	Loop Port 0 State Indication 0: Open 1: Closed	R
9	COMP0	Port 0 Communication State Indication 0: No stable communication 1: Communication established	R
10	LP1	Loop Port 1 State Indication 0: Open 1: Closed	R
11	COMP1	Port 1 Communication State Indication 0: No stable communication 1: Communication established	R
12	LP2	Loop Port 2 State Indication Port 2 is not available on this LSI. 0: Open 1: Closed	R
13	COMP2	Port 2 Communication State Indication Port 2 is not available on this LSI. 0: No stable communication 1: Communication established	R
14	LP3	Loop Port 3 State Indication Port 3 is not available on this LSI. 0: Open 1: Closed	R
15	COMP3	Port 3 Communication State Indication Port 3 is not available on this LSI. 0: No stable communication 1: Communication established	R

Note: Reading this DL status register from the ECAT clears bit 2 of the ECAT event request register (ECAT_EVENT_REQ at 0x0210). Avoid reading DL Status register from PDI.

The ESC_DL_STATUS register indicates the state of the EtherCAT slave controller.

28.8 Application Layer Registers

28.8.1 AL_CONTROL : AL Control Register

Base address: ESC = 0x8013_0000

Offset address: 0x0120

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	DEVICEID	ERRINDACK	INISTATE[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	INISTATE[3:0]	Change the state transition of the device state machine. The access type from the EtherCAT master is R/W. 0x1: Initial state request 0x2: Pre-operational state request 0x3: Bootstrap state request 0x4: Safe-operational state request 0x8: Operational state request Others: Reserved	R
4	ERRINDACK	Error Indication Acknowledge (Response) The access type from the EtherCAT master is R/W. 0: Error Indication in AL status register is not acknowledged 1: Error Indication in AL status register is acknowledged	R
5	DEVICEID	Device ID Request The access type from the EtherCAT master is R/W. 0: No request is present 1: A request is present	R
15:6	—	The read values are undefined. The access type from the EtherCAT master is R/W. In this case, the write value should be 0.	R

Note: The PDI has to read the AL control register after the ECAT has written it. Otherwise, the ECAT cannot write again to the AL control register. Reading the AL control register from the PDI clears bit 0 of the AL event request register (AL_EVENT_REQ at 0x0220).

The AL_CONTROL register is used to change the state transition of the device state machine and to acknowledge error indication.

28.8.2 AL_STATUS : AL Status Register

Base address: ESC = 0x8013_0000

Offset address: 0x0130

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	DEVICEID	ERR	ACTSTATE[3:0]			
Value after reset:	x	x	x	x	x	x	x	x	x	x	0	0	0	0	0	1

Bit	Symbol	Function	R/W
3:0	ACTSTATE[3:0]	State Machine State Indication Actual state of the device state machine The access type from the EtherCAT master is R. 0x1: Initial state 0x2: Pre-operational state 0x3: Request bootstrap state 0x4: Safe-operational state 0x8: Operational state	R/W

Bit	Symbol	Function	R/W
4	ERR	Error State Indication The access type from the EtherCAT master is R. 0: The device is in the state as requested or flag was cleared by command 1: The device has not entered the requested state or the state was changed as a result of local action	R/W
5	DEVICEID	Device ID Load State Indication The access type from the EtherCAT master is R. 0: Loading failed 1: Loading was successful	R/W
15:6	—	The read values are undefined. The write value should be 0.	R/W

Note: Reading this AL status register from the ECAT clears bit 3 of the ECAT event request register (ECAT_EVENT_REQ at 0x0210). Avoid reading DL Status register from PDI.

The AL_STATUS register indicates the state of slave application.

28.8.3 AL_STATUS_CODE : AL Status Code Register

Base address: ESC = 0x8013_0000

Offset address: 0x0134

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	STATUSCODE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	STATUSCODE[15:0]	AL status code The access type from the EtherCAT master is R.	R/W

The AL_STATUS_CODE register indicates an error code from slave application.

28.8.4 RUN_LED_OVERRIDE : RUN LED Override Register

Base address: ESC = 0x8013_0000

Offset address: 0x0138

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	OVERRIDEEN	LEDCODE[3:0]			
Value after reset:	x	x	x	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	LEDCODE[3:0]	LED Code Indication (FSM state: Bits [3:0] of the AL Status register, AL_STATUS) 0x0: Off (FSM: 1-Init) 0xD: Blinking (FSM: 2-PreOp) 0xE: Flickering (FSM: 3-Bootstrap) 0xF: On (FSM: 8-Op) Others: Flash 1x - 12x (FSM: 4-SafeOp 1x)	R/W
4	OVERRIDEEN	Override Setting 0: Override disabled 1: Override enabled	R/W
7:5	—	The read values are undefined. The write value should be 0.	R/W

Note: Changing the value of the AL status register to an appropriate value will clear bit 4 (override enable).
Normally RUN LED is controlled by the AL status register (AL_STATUS at 0x0130) automatically. It is not necessary to override RUN LED in order to indicate the state of a general state machine.
For example, this register can be used to run special lighting patterns that indicate the positions of specific slaves.

The RUN_LED_OVERRIDE register is used to override control of the RUN LED pin.

28.8.5 ERR_LED_OVERRIDE : ERR LED Override Register

Base address: ESC = 0x8013_0000

Offset address: 0x0139

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	OVERRIDEEN	LEDCODE[3:0]			
Value after reset:	x	x	x	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	LEDCODE[3:0]	LED Code Indication 0x0: Off 0xD: Blinking 0xE: Flickering 0xF: On Others: Flash 1x to 12x	R/W
4	OVERRIDEEN	Override Setting 0: Override disabled 1: Override enabled	R/W
7:5	—	The read values are undefined. The write value should be 0.	R/W

Note: Bit 4 (override enable) will be cleared if a new error occurs.
The ESC automatically controls an error LED under the conditions below. Regarding other errors, the error LED should be controlled by application using this register.
- SII EEPROM load error
- PDI watchdog timeout

The ERR_LED_OVERRIDE register is used to override control of the error LED pin.

28.9 PDI Registers

28.9.1 PDI_CONTROL : PDI Control Register

Base address: ESC = 0x8013_0000

Offset address: 0x0140

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PDI[7:0]							
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PDI[7:0]	Process Data Interface. In this LSI, the following value is indicated. 0x80: On-chip bus	R

This register indicates the type of PDI.

28.9.2 ESC_CONFIG : ESC Configuration Register

Base address: ESC = 0x8013_0000

Offset address: 0x0141

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENLP3	ENLP2	ENLP1	ENLP0	DCLATCH	DCSYNC	ENLALLP	DEVEMU
Value after reset:	0	0	0	0	1	1	0	0

Bit	Symbol	Function	R/W
0	DEVEMU	Device emulation (control of AL status) 0: The AL status register must be set by the PDI 1: The AL status register is set to a value written to the AL control register	R
1	ENLALLP	Sets enhanced link detection for all ports 0: Disabled (if bits 15 to 12 of address 0 in the EEPROM = 0) 1: Enabled at all ports	R
2	DCSYNC	Sets the SYNC output unit for distributed clocks (fixed to 1 in this LSI) 0: Disabled (power saving) 1: Enabled	R
3	DCLATCH	Sets the latch input unit for distributed clocks Fixed to 1 in this LSI. 0: Disabled (power saving) 1: Enabled	R
4	ENLP0	Port 0 Enhanced Link Detection Setting 0: Disabled (if bit 9 of address 0 in the EEPROM = 0) 1: Enabled	R
5	ENLP1	Port 1 Enhanced Link Detection Setting 0: Disabled (if bit 9 of address 0 in the EEPROM = 0) 1: Enabled	R
6	ENLP2	Port 2 Enhanced Link Detection Setting Port 2 is not available in this LSI. 0: Disabled (if bit 9 of address 0 in the EEPROM = 0) 1: Enabled	R
7	ENLP3	Port 3 Enhanced Link Detection Setting Port 3 is not available in this LSI. 0: Disabled (if bit 9 of address 0 in the EEPROM = 0) 1: Enabled	R

The ESC_CONFIG register indicates configuration of the EtherCAT slave controller.

28.9.3 PDI_CONFIG : PDI Configuration Register

Base address: ESC = 0x8013_0000

Offset address: 0x0150

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ONCHIPBUS[2:0]			ONCHIPBUSCLK[4:0]				
Value after reset:	0	1	0	0	0	1	0	0

Bit	Symbol	Function	R/W
4:0	ONCHIPBUSCLK[4:0]	On-Chip Bus Clock Indication Indicate the frequency of the on-chip bus clock. In this LSI, the value is always 0x04 (corresponding to 100 MHz).	R
7:5	ONCHIPBUS[2:0]	On-Chip Bus Type Indication Indicate the type of on-chip bus. In this LSI, the value is always 010b.	R

The PDI_CONFIG register indicates configuration of the PDI.

28.9.4 SYNC_LATCH_CONFIG : SYNC/LATCH PDI Configuration Register

Base address: ESC = 0x8013_0000

Offset address: 0x0151

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SYNC 1MAP	SYNC LAT1	SYNC1OUT[1:0]		SYNC 0MAP	SYNC LAT0	SYNC0OUT[1:0]	
Value after reset:	1	1	1	0	1	1	1	0

Bit	Symbol	Function	R/W
1:0	SYNC0OUT[1:0]	SYNC0 Output Driver and Polarity Indication Indicate the SYNC0 output driver/polarity. In this LSI, the value is always 10b (push-pull and active-high).	R
2	SYNCLAT0	SYNC0/LATCH0 Indication Indicates the SYNC0/LATCH0 configuration. In this LSI, the value is always 1.*1 0: LATCH0 input 1: SYNC0 output	R
3	SYNC0MAP	SYNC0 State Mapping Indication Indicates enabling or disabling of mapping of the SYNC0 state to bit 2 of the AL event request register (AL_EVENT_REQ at 0x0220). This is always enabled in this LSI, so the value indicated is always 1 (enabled). 0: Disabled 1: Enabled	R
5:4	SYNC1OUT[1:0]	SYNC1 Output Driver and Polarity Indication Indicate the SYNC1 output driver/polarity. In this LSI, the value is always 10b (push-pull and active-high).	R
6	SYNCLAT1	SYNC1/LATCH1 Indication Indicates the SYNC1/LATCH1 configuration. In this LSI, the value is always 1.*1 0: LATCH1 input 1: SYNC1 output	R
7	SYNC1MAP	SYNC1 State Mapping Indication Indicates enabling or disabling of mapping of the SYNC1 state to bit 3 of the AL event request register (AL_EVENT_REQ at 0x0220). This is always enabled in this LSI, so the value indicated is always 1 (enabled). 0: Disabled 1: Enabled	R

Note 1. Latch input is available though the value indicates SYNC output. Use the MPC function in order to switch SYNC output to LATCH input and vice versa.

The SYNC_LATCH_CONFIG register indicates the configuration of SYNC output and LATCH input.

28.9.5 EXT_PDI_CONFIG : Extended PDI Configuration Register

Base address: ESC = 0x8013_0000

Offset address: 0x0152

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DATABUSWID[1:0]	
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
1:0	DATABUSWID[1:0]	PDI Data Bus Width Indication Indicate the data bus width of the PDI. In this LSI, the value is always 00b (4 bytes). 0 0: 4 bytes 0 1: 1 byte 1 0: 2 bytes 1 1: Reserved	R
15:2	—	The read values are undefined.	R

The EXT_PDI_CONFIG register indicates configuration of the PDI.

28.10 Interrupt Registers

28.10.1 ECAT_EVENT_MASK : ECAT Event Mask Register

Base address: ESC = 0x8013_0000

Offset address: 0x0200

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ECATEVMASK[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ECATEVMASK[15:0]	Event Request Mask Setting The access type from the EtherCAT master is R/W. 0: The corresponding bit of the ECAT Event Request register (ECAT_EVENT_REQ at 0x0210) is not mapped 1: The corresponding bit of the ECAT Event Request register is mapped	R

The ECAT Event Request (ECAT interrupt) is used to transmit the slave event to the EtherCAT master. This register is used to set mask to each event of the ECAT Event Request register (ECAT_EVENT_REQ at 0x0210). The logical AND of each effective bit in the ECAT Event Request register and the corresponding bit of this register is taken and the result produces the interrupt signal.

28.10.2 AL_EVENT_MASK : AL Event Mask Register

Base address: ESC = 0x8013_0000

Offset address: 0x0204

Bit position:	31																													0
Bit field:	ALEVMASK[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	1	1	1	1

Bit	Symbol	Function	R/W
31:0	ALEVMASK[31:0]	Event Request Mask Setting The access type from the EtherCAT master is R. 0: The corresponding bit of the AL Event Request register (AL_EVENT_REQ at 0x0220) is not mapped 1: The corresponding bit of the AL Event Request register is mapped	R/W

The AL event request (PDI interrupt) is used to transmit the ESC interrupt to the slave application. This register is used to set mask to each event of the AL Event Request register (AL_EVENT_REQ at 0x0220). The logical AND of each effective bit in the AL Event Request register and the corresponding bit of this register is taken and the result produces the interrupt signal.

28.10.3 ECAT_EVENT_REQ : ECAT Event Request Register

Base address: ESC = 0x8013_0000

Offset address: 0x0210

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	SMSTA7	SMSTA6	SMSTA5	SMSTA4	SMSTA3	SMSTA2	SMSTA1	SMSTA0	ALSTA	DLSTA	—	DCLATCH
Value after reset:	x	x	x	x	0	0	0	0	0	0	0	0	0	0	x	0

Bit	Symbol	Function	R/W
0	DCLATCH	DC Latch Event State Indication This bit is cleared by reading DC latch event times from the ECAT for ECAT controlled latch units, so the latch 0 and 1 status registers (DC_LATCH_STAT0 and DC_LATCH_STAT1 at 0x09AE and 0x09AF) indicate no event. 0: No change on DC latch inputs 1: At least one change on DC latch inputs	R
1	—	The read value is undefined.	R
2	DLSTA	DL Status Event State Indication This bit is cleared by reading out the DL status register (ESC_DL_STATUS at 0x0110 or 0x0111) from the ECAT. 0: No change in DL status 1: DL status change	R
3	ALSTA	AL Status Event State Indication This bit is cleared by reading out the AL status register (AL_STATUS at 0x0130 or 0x0131) from the ECAT. 0: No change in AL status 1: AL status change	R
4	SMSTA0	Mirror value of SyncManager 0 Status Indication 0: No Sync channel 0 event 1: Sync channel 0 event pending	R
5	SMSTA1	Mirror value of SyncManager 1 Status Indication 0: No Sync channel 1 event 1: Sync channel 1 event pending	R
6	SMSTA2	Mirror value of SyncManager 2 Status Indication 0: No Sync channel 2 event 1: Sync channel 2 event pending	R
7	SMSTA3	Mirror value of SyncManager 3 Status Indication 0: No Sync channel 3 event 1: Sync channel 3 event pending	R
8	SMSTA4	Mirror value of SyncManager 4 Status Indication 0: No Sync channel 4 event 1: Sync channel 4 event pending	R
9	SMSTA5	Mirror value of SyncManager 5 Status Indication 0: No Sync channel 5 event 1: Sync channel 5 event pending	R
10	SMSTA6	Mirror value of SyncManager 6 Status Indication 0: No Sync channel 6 event 1: Sync channel 6 event pending	R
11	SMSTA7	Mirror value of SyncManager 7 Status Indication 0: No Sync channel 7 event 1: Sync channel 7 event pending	R
15:12	—	The read values are undefined.	R

The ECAT_EVENT_REQ register indicates the source of ECAT event requests (ECAT interrupts).

28.10.4 AL_EVENT_REQ : AL Event Request Register

Base address: ESC = 0x8013_0000

Offset address: 0x0220

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SMINT 7	SMINT 6	SMINT 5	SMINT 4	SMINT 3	SMINT 2	SMINT 1	SMINT 0	—	WDPD	—	SYNC ACT	DCSY NC1S TA	DCSY NC0S TA	DCLA TCH	ALCT RL
Value after reset:	0	0	0	0	0	0	0	0	x	0	x	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ALCTRL	AL Control Event State Indication This bit is cleared by reading the AL control register (AL_CONTROL at 0x0120 or 0x0121) from the PDI. 0: No change in the AL control register 1: The AL control register has been written	R
1	DCLATCH	DC Latch Event State Indication This bit is cleared by reading DC latch event times from the PDI for PDI controlled latch units, so the latch 0 and 1 status registers (DC_LATCH_STAT0 and DC_LATCH_STAT1 at 0x09AE and 0x09AF) indicate no event. 0: No change on DC latch inputs 1: At least one change on DC latch inputs	R
2	DCSYNC0STA	DC SYNC0 State Indication This bit is cleared by reading the SYNC0 status register (DC_SYNC0_STAT at 0x098E) from the PDI.	R
3	DCSYNC1STA	DC SYNC1 State Indication This bit is cleared by reading the SYNC1 status register (DC_SYNC1_STAT at 0x098F) from the PDI.	R
4	SYNCACT	SyncManager Activation Indication Change of the SyncManager activation register (SMn.ACT at 0x0806 + 0x8 × n) This bit is cleared by reading SyncManager activation registers from the PDI. 0: No change in any SyncManager 1: At least one SyncManager changed	R
5	—	The read value is undefined.	R
6	WDPD	Watchdog Process Data Indication This bit is cleared by reading the watchdog status process data register (WDS_DATA at 0x0440) from the PDI. 0: Valid 1: Timeout	R
7	—	The read value is undefined.	R
8	SMINT0	SyncManager 0 interrupt (bit 0 or 1 of the SyncManager status register (0x0805)) 0: No SyncManager 0 interrupt 1: SyncManager 0 interrupt pending	R
9	SMINT1	SyncManager 1 interrupt (bit 0 or 1 of the SyncManager status register (0x080D)) 0: No SyncManager 1 interrupt 1: SyncManager 1 interrupt pending	R
10	SMINT2	SyncManager 2 interrupt (bit 0 or 1 of the SyncManager status register (0x0815)) 0: No SyncManager 2 interrupt 1: SyncManager 2 interrupt pending	R
11	SMINT3	SyncManager 3 interrupt (bit 0 or 1 of the SyncManager status register (0x081D)) 0: No SyncManager 3 interrupt 1: SyncManager 3 interrupt pending	R

Bit	Symbol	Function	R/W
12	SMINT4	SyncManager 4 interrupt (bit 0 or 1 of the SyncManager status register (0x0825)) 0: No SyncManager 4 interrupt 1: SyncManager 4 interrupt pending	R
13	SMINT5	SyncManager 5 interrupt (bit 0 or 1 of the SyncManager status register (0x082D)) 0: No SyncManager 5 interrupt 1: SyncManager 5 interrupt pending	R
14	SMINT6	SyncManager 6 interrupt (bit 0 or 1 of the SyncManager status register (0x0835)) 0: No SyncManager 6 interrupt 1: SyncManager 6 interrupt pending	R
15	SMINT7	SyncManager 7 interrupt (bit 0 or 1 of the SyncManager status register (0x083D)) 0: No SyncManager 7 interrupt 1: SyncManager 7 interrupt pending	R
31:16	—	The read values are undefined.	R

The AL_EVENT_REQ register indicates the source of AL event requests (PDI interrupts).

28.11 Error Counter Registers

28.11.1 RX_ERR_COUNTn : RX Error Counter n Register (n = 0 to 2)

Base address: ESC = 0x8013_0000

Offset address: 0x0300 + 0x2 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RXERRCNT[7:0]								INVFRMCNT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	INVFRMCNT[7:0]	Invalid Frame Counter Value Indication Counter value of invalid frames for port n counting is stopped when 0xFF is reached. These bits are cleared if one of the RX error counters (RX_ERR_COUNTn, FWD_RX_ERR_COUNTn) is written. The access type from the EtherCAT master is R/W.	R
15:8	RXERRCNT[7:0]	RX Frame Error Counter Value Indication Counter value of RX errors for port n counting is stopped when 0xFF is reached. The number of RX errors of MII interface is counted. These bits are cleared if one of the RX error counters (RX_ERR_COUNTn, FWD_RX_ERR_COUNTn) is written. The access type from the EtherCAT master is R/W.	R

The RX_ERR_COUNTn register counts RX frame errors.

28.11.2 FWD_RX_ERR_COUNTn : Forwarded RX Error Counter n Register (n = 0 to 2)

Base address: ESC = 0x8013_0000

Offset address: 0x0308 + 0x1 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	FWDERRCNT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	FWDERRCNT[7:0]	Forwarded Error Counter Value Indication Counter value of forwarded RX error frames for port n counting is stopped when 0xFF is reached. These bits are cleared if one of the RX error counters (RX_ERR_COUNTn, FWD_RX_ERR_COUNTn) is written. The access type from the EtherCAT master is R/W.	R

The FWD_RX_ERR_COUNTn register counts forwarded RX frame errors.

28.11.3 ECAT_PROC_ERR_COUNT : ECAT Processing Unit Error Counter Register

Base address: ESC = 0x8013_0000

Offset address: 0x030C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	EPUERRCNT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	EPUERRCNT[7:0]	Processing Unit Error Counter Value Indication ECAT processing unit error counter value counting is stopped when 0xFF is reached. This register counts errors of frames passing the processing unit. Writing to this register clears it. The access type from the EtherCAT master is R/W.	R

The ECAT_PROC_ERR_COUNT register counts frame errors passing the ECAT processing unit.

28.11.4 PDI_ERR_COUNT : PDI Error Counter Register

Base address: ESC = 0x8013_0000

Offset address: 0x030D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PDIERRCNT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PDIERRCNT[7:0]	PDI Error Counter Value Indication Counting is stopped when 0xFF is reached. Counting starts when an interface error occurs due to access to the PDI. Writing to this register clears it. The access type from the EtherCAT master is R/W.	R

The PDI_ERR_COUNT register counts PDI access errors.

28.11.5 LOST_LINK_COUNTn : Lost Link Counter n Register (n = 0 to 2)

Base address: ESC = 0x8013_0000

Offset address: 0x0310 + 0x1 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LOSTLINKCNT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	LOSTLINKCNT[7:0]	Lost Link Counter Value Indication Lost link counter value for port n counting is stopped when 0xFF is reached. Counting starts only when port loop is Auto or Auto-Close. Only lost links at open ports are counted. Writing to one of the lost link counter registers clears it. The access type from the EtherCAT master is R/W.	R

The LOST_LINK_COUNTn register counts lost links at port n.

28.12 Watchdog Registers

28.12.1 WD_DIVIDE : Watchdog Divider Register

Base address: ESC = 0x8013_0000

Offset address: 0x0400

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WDDIV[15:0]															
Value after reset:	0	0	0	0	1	0	0	1	1	1	0	0	0	0	1	0

Bit	Symbol	Function	R/W
15:0	WDDIV[15:0]	Watchdog Clock Frequency Divisor Setting Set the frequency divisor of the clock to drive counting by the watchdog timer in units of ticks at 25 MHz. The clock that drives counting by the watchdog timer is obtained by dividing 25 MHz by the value in this register plus 2. The default value is 2498, which corresponds to a period of 100 µs. The access type from the EtherCAT master is R/W.	R

The WD_DIVIDE register is used to set the ratio for dividing 25 MHz to obtain the basic period for incrementing the watchdog timer.

28.12.2 WDT_PDI : Watchdog Time PDI Register

Base address: ESC = 0x8013_0000

Offset address: 0x0410

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WDTIMPDI[15:0]															
Value after reset:	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0

Bit	Symbol	Function	R/W
15:0	WDTIMPDI[15:0]	Watchdog Overflow Time Setting Set the time until the PDI watchdog timer overflows as a number of times the watchdog is incremented. With the default values for these bits and the setting of the watchdog divider, the time for a single incrementation is 100 µs, so the watchdog timer overflows when 100 µs ×1000 = 100 ms elapsed. Setting these bits to 0 disables the watchdog timer. Access to the PDI restarts the watchdog timer. The access type from the EtherCAT master is R/W.	R

The WDT_PDI register is used to set the time until the PDI watchdog timer overflows.

28.12.3 WDT_DATA : Watchdog Time Process Data Register

Base address: ESC = 0x8013_0000

Offset address: 0x0420

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WDTIMPD[15:0]															
Value after reset:	0	0	0	0	0	0	1	1	1	1	1	0	1	0	0	0

Bit	Symbol	Function	R/W
15:0	WDTIMPD[15:0]	Watchdog Overflow Time Setting Set the time until the process data watchdog timer overflows as a number of times the watchdog is incremented. With the default values for these bits and the setting of the watchdog divider, the time for a single incrementation is 100 µs, so the watchdog timer overflows when 100 µs × 1000 = 100 ms elapsed. There is one watchdog for all SyncManagers. Setting these bits to 0 disables the watchdog timer. Access to the Watchdog Trigger Enable bit of SyncManager restarts the watchdog timer. The access type from the EtherCAT master is R/W.	R

The WDT_DATA register is used to set the time until the process data watchdog timer overflows.

28.12.4 WDS_DATA : Watchdog Status Process Data Register

Base address: ESC = 0x8013_0000

Offset address: 0x0440

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WDSTAPD
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	WDSTAPD	Watchdog State Indication Indicates the state of the process data watchdog timer triggered by SyncManagers. Reading this register clears bit 6 of the AL Event Request register (AL_EVENT_REQ at 0x0220). 0: The timeout period of the process data watchdog timer elapses 1: The process data watchdog timer is active or disabled	R
15:1	—	The read values are undefined.	R

The WDS_DATA register indicates the state of the process data watchdog timer.

28.12.5 WDC_DATA : Watchdog Counter Process Data Register

Base address: ESC = 0x8013_0000

Offset address: 0x0442

Bit position:	7	6	5	4	3	2	1	0
Bit field:	WDCNTPD[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	WDCNTPD[7:0]	Watchdog Counter Value Indication Counter value of the process data watchdog timer counting stops when 0xFF is reached. Counting starts on a timeout of the process data watchdog timer. Writing to one of the watchdog counter registers (WDC_DATA, WDC_PDI at 0x0442 and 0x0443) clears the counter. The access type from the EtherCAT master is R/W.	R

The WDC_DATA register indicates the timeout counter value of the process data watchdog timer.

28.12.6 WDC_PDI : Watchdog Counter PDI Register

Base address: ESC = 0x8013_0000

Offset address: 0x0443

Bit position:	7	6	5	4	3	2	1	0
Bit field:	WDCNTPDI[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	WDCNTPDI[7:0]	Watchdog Counter Value Indication Counter value of the PDI watchdog timer counting stops when 0xFF is reached. Counting starts on a timeout of the PDI watchdog timer. Writing to one of the watchdog counter registers (WDC_DATA, WDC_PDI at 0x0442 and 0x0443) clears the counter. The access type from the EtherCAT master is R/W.	R

The WDC_PDI register indicates the timeout counter value of the PDI watchdog timer.

28.13 SII EEPROM Interface Registers

The EtherCAT controls the SII EEPROM interface if bit 0 is 0 in the EEPROM Configuration register (EEP_CONF at 0x0500) and in the EEPROM PDI Access State register (EEP_STATE at 0x0501). Otherwise, the PDI controls the EEPROM interface.

28.13.1 EEP_CONF : EEPROM Configuration Register

Base address: ESC = 0x8013_0000

Offset address: 0x0500

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	FORCEECAT	CTRLPDI
Value after reset:	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
0	CTRLPDI	PDI EEPROM Control Specifies whether EEPROM control is offered to the PDI. The access type from the EtherCAT master is R/W. 0: The PDI has no EEPROM control 1: The PDI has EEPROM control	R
1	FORCEECAT	EEPROM Access Right Change Forcibly changes the right of access to the EEPROM by the ECAT. The access type from the EtherCAT master is R/W. 0: No change 1: Reset bit 0 of the EEPROM PDI Access State register (EEP_STATE at 0x0501) to 0. That is, prohibit access to the EEPROM by the PDI.	R

Bit	Symbol	Function	R/W
7:2	—	The read values are undefined.	R

The EEP_CONF register is used to configure EEPROM access.

28.13.2 EEP_STATE : EEPROM PDI Access State Register

Base address: ESC = 0x8013_0000

Offset address: 0x0501

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	PDIACCESS
Value after reset:	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	PDIACCESS	EEPROM Access Right Setting Write access from the PDI is only possible when bit 0 is 1 and bit 1 is 0 in the EEPROM Configuration register (EEP_CONF at 0x0500). The access type from the EtherCAT master is R. 0: Prohibits the PDI from access to the EEPROM 1: The PDI has access to the EEPROM	R/W
7:1	—	The read values are undefined. The write value should be 0. The access type from the EtherCAT master is R.	R/W

The EEP_STATE register is used to set the right of access to the EEPROM by the PDI.

28.13.3 EEP_CONT_STAT : EEPROM Control/Status Register

Base address: ESC = 0x8013_0000

Offset address: 0x0502

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BUSY	WRENERR	ACKCMDERR	LOADSTA	CKSUMERR	COMMAND[2:0]			PROMSIZE	READBYTE	—	—	—	—	—	ECATWREN
Value after reset:	0	0	0	0	0	0	0	0	0/1*1	0	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	ECATWREN	ECAT Write Enable*3 This bit is always 1 if the PDI has EEPROM control. The access type from the EtherCAT master is R/W. 0: Write requests are disabled 1: Write requests are enabled	R
5:1	—	The read values are undefined. The write value should be 0.	R/W
6	READBYTE	EEPROM Read Byte Indication Indicates supported EEPROM read bytes. 0: 4 bytes 1: 8 bytes	R
7	PROMSIZE	EEPROM Algorithm Indication Indicates the selected EEPROM algorithm. 0: 1 address byte (1-Kbit to 16-Kbit EEPROMs) 1: 2 address bytes (32-Kbit to 4-Mbit EEPROMs)	R

Bit	Symbol	Function	R/W
10:8	COMMAND[2:0]	Command*3 Write: Initiates the commands below. Read: Indicates the currently executed command. Commands: 0 0 0: No command/EEPROM idle (clear error bits) 0 0 1: Read 0 1 0: Write 1 0 0: Reload Others: Reserved/invalid commands (must not be issued)	R/W
11	CKSUMERR	Checksum Error Indication Indicates checksum error in the ESC configuration area. 0: No error in the checksum 1: Error in the checksum	R
12	LOADSTA	EEPROM Loading Status Indication Indicates EEPROM loading status. 0: EEPROM has been loaded and device information has no problem 1: EEPROM has not been loaded and device information is not available (EEPROM loading in progress or finished with a failure)	R
13	ACKCMDERR	Acknowledge/Command Error Indication Indicates acknowledge/command error.*2 0: No error 1: Missing EEPROM acknowledge or invalid command	R
14	WRENERR	Write Enable Error Indication Indicates write enable error.*2 0: No error 1: Write command without write enable	R
15	BUSY	EEPROM Interface State Indication Indicates a busy state of the EEPROM interface. 0: The EEPROM interface is idle 1: The EEPROM interface is busy	R

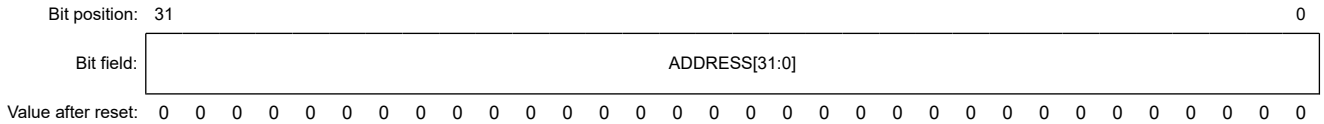
Note: Write access depends on the assignment of the EEPROM interface (ECAT/PDI). Write access is generally blocked if the EEPROM interface is busy (bit 15 = 1).
Note 1. This initial value will be changed by setting of EEPROMSIZE bit in EtherCAT Operation Mode Register (ECATOPMOD).
Note 2. Error bits are cleared by writing 000b (or any valid command) to command bits [10:8].
Note 3. ECAT write enable bit 0 is self-cleared at the SOF of the next frame. Command bits [10:8] are also self-cleared after the command is executed (EEPROM busy ends).
Writing 000b to command bits [10:8] also clears the error bits 14 and 13. Command bits [10:8] are ignored if the acknowledge/command error bit 13 is 1.

The EEP_CONT_STAT register is used to control access to the EEPROM and indicate the status.

28.13.4 EEP_ADR : EEPROM Address Register

Base address: ESC = 0x8013_0000

Offset address: 0x0504



Bit	Symbol	Function	R/W
31:0	ADDRESS[31:0]	EEPROM Address Setting Actually used EEPROM address bits: [9:0]: EEPROM size of up to 16 Kbits [17:0]: EEPROM size of 32 Kbits to 4 Mbits 0: First word (= 16 bits) 1: Second word ⋮	R/W

Note: Write access depends on the assignment of the EEPROM interface (ECAT/PDI). Write access is generally blocked if the EEPROM interface is busy (bit 15 is 1 in the EEPROM Control/Status register, EEP_CONT_STAT, at 0x0502).

The EEP_ADR register is used to set the EEPROM address to be accessed.

28.13.5 EEP_DATA : EEPROM Data Register

Base address: ESC = 0x8013_0000

Offset address: 0x0508

Bit position:	31																16	15														0
Bit field:	HIDATA[15:0]																LODATA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	LODATA[15:0]	Data to be written to the EEPROM or data read from the EEPROM (lower 2 bytes)	R/W
31:16	HIDATA[15:0]	Data read from the EEPROM (upper 2 bytes)	R

Note: Write access depends on the assignment of the EEPROM interface (ECAT/PDI). Write access is generally blocked if the EEPROM interface is busy (bit 15 is 1 in the EEPROM Control/Status register, EEP_CONT_STAT, at 0x0502).

The EEP_DATA register is used to set write data to the EEPROM or indicates read data from the EEPROM. It can be written in 1-word units and read in 2-word units.

28.14 MII Management Interface Registers

28.14.1 MII_CONT_STAT : MII Management Control/Status Register

Base address: ESC = 0x8013_0000

Offset address: 0x0510

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BUSY	CMDERR	READERR	—	—	—	COMMAND[1:0]		PHYOFFSET[4:0]					MILINK	PDICTRL	WREN
Value after reset:	0	0	0	x	x	x	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
0	WREN	Write Enable This bit is always 1 if the PDI controls the MII management interface. The access type from the EtherCAT master is R/W. 0: Disabled 1: Enabled	R
1	PDICTRL	PDI Control Indication Indicates whether the MII management interface can be controlled by the PDI. The interface is controlled by the MII management ECAT Access State register (MII_ECAT_ACS_STAT at 0x0516) and the MII management PDI Access State register (MII_PDI_ACS_STAT at 0x0517). 0: Only ECAT control 1: PDI control possible	R
2	MILINK	MI Link Detection 0: Not available 1: Available	R
7:3	PHYOFFSET[4:0]	PHY Address Offset Indication Indicate the PHY address offset.	R

Bit	Symbol	Function	R/W
9:8	COMMAND[1:0]	Command Write: Initiates the commands below. Read: Indicates the currently executed command. Commands: 0 0: No command/MI idle (clear error bits) 0 1: Read 1 0: Write Others: Setting prohibited	R/W
12:10	—	The read values are undefined. The write value should be 0.	R/W
13	READERR	Read Error Indication Indicates whether a read error occurred. This bit is cleared by writing to this register. 0: No read error 1: Read error occurred (PHY or register not available)	R/W
14	CMDERR	Command Error Indication Indicates whether a command error occurred. This bit is cleared by executing a valid command or writing 00b to COMMAND[1:0]. 0: Last command was successful 1: Invalid command or write command without write enable	R
15	BUSY	MII Management State Indication Indicates that the MII management interface is busy. 0: MII management interface is idle 1: MII management interface is busy	R

The MII_CONT_STAT register is used to control the MII management interface and to indicate the status.

Write enable bit 0 is self-cleared at the SOF of the next frame. Command bits [9:8] are also self-cleared after the command is executed (busy ends). Writing 00b to command bits also clears the error bits 14 and 13. Command bits are cleared after the command is executed.

Write access depends on the assignment of the management interface (ECAT/PDI). Write access is generally blocked if the management interface is busy (bit 15 = 1 in this register).

28.14.2 PHY_ADR : PHY Address Register

Base address: ESC = 0x8013_0000

Offset address: 0x0512

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	PHYADDR[4:0]				
Value after reset:	x	x	x	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	PHYADDR[4:0]	PHY Address Setting	R/W
7:5	—	The read values are undefined.	R

The PHY_ADR register is used to set the PHY address.

Write access depends on the assignment of the management interface (ECAT/PDI). Write access is generally blocked if the management interface is busy (bit 15 is 1 in the MII management control/status register, MII_CONT_STAT, at 0x0510).

28.14.3 PHY_REG_ADR : PHY Register Address Register

Base address: ESC = 0x8013_0000

Offset address: 0x0513

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	PHYREGADDR[4:0]				
Value after reset:	x	x	x	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	PHYREGADDR[4:0]	Address of PHY register	R/W
7:5	—	The read values are undefined.	R/W

The PHY_REG_ADR register is used to set the PHY register address.

Write access depends on the assignment of the management interface (ECAT/PDI). Write access is generally blocked if the management interface is busy (bit 15 is 1 in the MII management Control/Status register, MII_CONT_STAT, at 0x0510).

28.14.4 PHY_DATA : PHY Data Register

Base address: ESC = 0x8013_0000

Offset address: 0x0514

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PHYREGDATA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PHYREGDATA[15:0]	PHY Register Data Indication/Setting	R/W

The PHY_DATA register is used to set data to write to PHY registers or to indicate data read from PHY registers.

Write access depends on the assignment of the management interface (ECAT/PDI). Write access is generally blocked if the management interface is busy (bit 15 is 1 in the MII management Control/Status register, MII_CONT_STAT, at 0x0510).

28.14.5 MII_ECAT_ACS_STAT : MII Management ECAT Access State Register

Base address: ESC = 0x8013_0000

Offset address: 0x0516

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ACSMII
Value after reset:	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	ACSMII	MII Management Interface Access Right Setting Right of access to the MII management interface The access type from the EtherCAT master is R/W. 0: Enables access to the MII management interface by the PDI. 1: Exclusive access to the MII management interface by the ECAT	R
7:1	—	The read values are undefined.	R

Note: Write access is only possible when bit 0 is 1 in the MII management PDI Access State register (MII_PDI_ACS_STAT at 0x0517).

The MII_ECAT_ACS_STAT register is used to set the right of access to the MII management interface.

28.14.6 MII_PDI_ACS_STAT : MII Management PDI Access State Register

Base address: ESC = 0x8013_0000

Offset address: 0x0517

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	FORPDI	ACSMII
Value after reset:	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
0	ACSMII	Right of access to the MII management interface The access type from the EtherCAT master is R. 0: Access to the MII management interface by the ECAT 1: Access to the MII management interface by the PDI	R/W
1	FORPDI	Forced change of access by the PDI (forced change of bit 0) The access type from the EtherCAT master is R/W. 0: The value of bit 0 of this register is not changed 1: The value of bit 0 of this register is reset to 0 (the right of access is changed to the ECAT)	R
7:2	—	The read values are undefined. The access type from the EtherCAT master is R.	R/W

Note: Write access to bit 0 from the PDI is only possible if the following two conditions are satisfied.
- Bit 0 is 0 in the MII management ECAT Access State register (MII_ECAT_ACS_STAT at 0x0516).
- Bit 1 is 0 in the MII management PDI Access State register (MII_PDI_ACS_STAT at 0x0517).

The MII_PDI_ACS_STAT register is used to set the right of access to the MII management interface.

28.15 FMMU Registers

28.15.1 FMMUn_L_START_ADR : FMMU Logical Start Address n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0600 + 0x10 × n

Bit position:	31	0
Bit field:	LSTAADR[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	LSTAADR[31:0]	Logical Start Address Setting Set the start of the logical address within the EtherCAT address space. The access type from the EtherCAT master is R/W.	R

The FMMUn_L_START_ADR register is used to set the logical start address within the EtherCAT address space for FMMU.

28.15.2 FMMUn_LEN : FMMU Length n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0604 + 0x10 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	FMMULEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	FMMULEN[15:0]	Area Size Specification Set the area size in byte units. Address at the end of the logical address range set by the FMMU minus the address at the start of the logical address range set by the FMMU plus 1. The access type from the EtherCAT master is R/W.	R

The FMMUn_LEN register is used to set the length for FMMU area in byte units.

28.15.3 FMMUn_L_START_BIT : FMMU Logical Start Bit n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0606 + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	LSTABIT[2:0]		
Value after reset:	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	LSTABIT[2:0]	Start Bit Setting Set the start bits of the logical start address for FMMU. The access type from the EtherCAT master is R/W.	R
7:3	—	The read values are undefined.	R

The FMMUn_L_START_BIT register is used to set the start bits of the logical start address for FMMU.

28.15.4 FMMUn_L_STOP_BIT : FMMU Logical Stop Bit n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0607 + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	LSTPBIT[2:0]		
Value after reset:	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	LSTPBIT[2:0]	Last Bit Setting Set the last bits of the logical end address for FMMU. The access type from the EtherCAT master is R/W.	R
7:3	—	The read values are undefined.	R

The FMMUn_L_STOP_BIT register is used to set the last bits of the logical end address for FMMU.

28.15.5 FMMUn_P_START_ADR : FMMU Physical Start Address n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0608 + 0x10 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PHYSTAADR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PHYSTAADR[15:0]	Physical Start Address Setting Set the physical start address to which the logical start address will be mapped. The address is set as an offset from the base address (0x8013_0000). The access type from the EtherCAT master is R/W.	R

The FMMUn_P_START_ADR register is used to set the physical start address of the ESC to which the logical start address will be mapped by the FMMU.

28.15.6 FMMUn_P_START_BIT : FMMU Physical Start Bit n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x060A + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	PHYSTABIT[2:0]		
Value after reset:	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
2:0	PHYSTABIT[2:0]	Physical Start Bit Setting Set the start bits of the physical start address to which the start bits of the logical start address will be mapped. The access type from the EtherCAT master is R/W.	R
7:3	—	The read values are undefined.	R

The FMMUn_P_START_BIT register is used to set the start bits of the physical start address of the ESC to which the start bits of the logical start address will be mapped by the FMMU.

28.15.7 FMMUn_TYPE : FMMU Type n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x060B + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	WRITE	READ
Value after reset:	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
0	READ	Read Access Mapping Setting The access type from the EtherCAT master is R/W. 0: Disabled 1: Enabled	R
1	WRITE	Write Access Mapping Setting The access type from the EtherCAT master is R/W. 0: Disabled 1: Enabled	R
7:2	—	The read values are undefined.	R

The FMMUn_TYPE register is used to set the type of FMMU access.

28.15.8 FMMUn_ACT : FMMU Activate n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x060C + 0x10 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ACTIVATE
Value after reset:	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	ACTIVATE	FMMU Enable/Disable The access type from the EtherCAT master is R/W. 0: Disabled 1: Enabled	R
7:1	—	The read values are undefined.	R

The FMMUn_ACT register is used to enable or disable FMMU.

28.16 SyncManager Registers

28.16.1 SMn_P_START_ADR : SyncManager Physical Start Address n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0800 + 0x8 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SMSTAADDR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SMSTAADDR[15:0]	Physical Start Address Setting Specify the physical start address for the area assigned to SyncManager. The access type from the EtherCAT master is R/W.	R

Note: The SMn_P_START_ADR register can only be written when SyncManager is disabled (bit 0 is 0 in the SyncManager Activate n register (SMn.ACT at 0x0806 + 0x8 × n).

This register is used to set the physical start address for the area assigned to SyncManager.

28.16.2 SMn_LEN : SyncManager Length n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0802 + 0x8 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SMLEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SMLEN[15:0]	Area Size Setting Set the number of bytes assigned to SyncManager. Set a value greater than 1. Otherwise, SyncManager is not activated. The access type from the EtherCAT master is R/W.	R

Note: This register can only be written when SyncManager is disabled (bit 0 is 0 in the SyncManager Activate n register (SMn.ACT at 0x0806 + 0x8 × n)).

The SMn_LEN register is used to set the size of the area assigned to SyncManager.

28.16.3 SMn_CONTROL : SyncManager Control n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0804 + 0x8 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	WDTRGEN	IRQPDI	IRQECAT	DIR[1:0]		OPEMODE[1:0]	
Value after reset:	x	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	OPEMODE[1:0]	Operating Mode Setting The access type from the EtherCAT master is R/W. 0 0: Buffer mode (3-buffer mode) 1 0: Mailbox mode (single buffer mode) Others: Reserved	R
3:2	DIR[1:0]	Transfer Direction Setting The access type from the EtherCAT master is R/W. 0 0: Read (ECAT: read access; PDI: write access) 0 1: Write (ECAT: write access; PDI: read access) Others: Reserved	R
4	IRQECAT	ECAT Event Interrupt Setting Enables or disables interrupts (ECAT interrupts) by the ECAT Event Request register (ECAT_EVENT_REQ at 0x0210). The access type from the EtherCAT master is R/W. 0: Disabled 1: Enabled	R
5	IRQPDI	AL Event Interrupt Setting Enables or disables interrupts (PDI interrupts) by the AL Event Request register (AL_EVENT_REQ at 0x0220). The access type from the EtherCAT master is R/W. 0: Disabled 1: Enabled	R
6	WDTRGEN	Watchdog Trigger Setting The access type from the EtherCAT master is R/W. 0: Disabled 1: Enabled	R
7	—	The read value is undefined.	R

Note: This register can only be written when SyncManager is disabled (bit 0 is 0 in the SyncManager Activate n register (SMn.ACT at 0x0806 + 0x8 × n)).

The SMn_CONTROL register is used to set operation of SyncManager.

28.16.4 SMn_STATUS : SyncManager Status n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0805 + 0x8 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	WRBUF	RDBUF	BUFFERED[1:0]		MAILBOX	—	INTRD	INTWR
Value after reset:	0	0	1	1	0	x	0	0

Bit	Symbol	Function	R/W
0	INTWR	Write Complete Interrupt State Indication 0: Indicates that the buffer was successfully written 1: Indicates that the first byte of the buffer was read (interrupt cleared)	R
1	INTRD	Read Complete Interrupt State Indication 0: Indicates that the buffer was successfully read 1: Indicates that the first byte of the buffer was written (interrupt cleared)	R
2	—	The read value is undefined.	R
3	MAILBOX	Mailbox Status Indication This bit is not used in buffer mode. 0: Mailbox empty 1: Mailbox full	R
5:4	BUFFERED[1:0]	Buffer Status Indication Indicates the buffer status in buffer mode (last written buffer). This bit is not used in mailbox mode. 0 0: 1st buffer 0 1: 2nd buffer 1 0: 3rd buffer 1 1: No buffer written	R
6	RDBUF	Read State Indication Indicates that the buffer is being read.	R
7	WRBUF	Write State Indication Indicates that the buffer is being written.	R

The SMn_STATUS register indicates the state of SyncManager.

28.16.5 SMn_ACT : SyncManager Activate n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0806 + 0x8 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LATCHPDI	LATCHECAT	—	—	—	—	REPEATREQ	SMEN
Value after reset:	0	0	x	x	x	x	0	0

Bit	Symbol	Function	R/W
0	SMEN	SyncManager Enable/Disable The access type from the EtherCAT master is R/W. 0: Disabled. Memory is accessed without SyncManager control. 1: Enabled. SyncManager is active and controls memory area set in configuration.	R (ack)
1	REPEATREQ	Repeat Request Toggling of the repeat request signal means that retrying the mailbox is required (primarily used in conjunction with reading of the ECAT mailbox). The access type from the EtherCAT master is R/W.	R (ack)
5:2	—	The read values are undefined.	R
6	LATCHECAT	ECAT Latch Event Specification The access type from the EtherCAT master is R/W. 0: No latch events 1: Generates latch events if the EtherCAT master switches the buffers	R (ack)
7	LATCHPDI	PDI Latch Event Specification The access type from the EtherCAT master is R/W. 0: No latch events 1: Generates latch events if the PDI switches the buffers or accesses the buffer start address	R (ack)

Note: Reading this register from the PDI in all SyncManagers which have changed activation clears bit 4 of the AL Event Request register (AL_EVENT_REQ at 0x0220).

The SMn_ACT register is used to set operation of SyncManager.

28.16.6 SMn_PDI_CONT : SyncManager PDI Control n Register (n = 0 to 7)

Base address: ESC = 0x8013_0000

Offset address: 0x0807 + 0x8 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	REPEATACK	DEACTIVE
Value after reset:	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
0	DEACTIVE	SyncManager Operation Indication/Setting Deactivates SyncManager. Writing 1 is delayed until the end of a frame which is currently processed. The access type from the EtherCAT master is R. 0: Read: Normal operation. SyncManager is activated. Write: Activates SyncManager. 1: Read: SyncManager is deactivated and reset. SyncManager locks access to memory area. Write: Deactivates SyncManager.	R/W
1	REPEATACK	Repeat Acknowledge If this bit is set to the same value as set by bit 0 (repeat request) of the SyncManager activate register (SMn.ACT at 0x0806 + 0x8 × n), the PDI acknowledges the execution of a previous set repeat request. The access type from the EtherCAT master is R.	R/W
7:2	—	The read values are undefined. The write value should be 0.	R/W

The SMn_PDI_CONT register is used to set operation of SyncManager from the PDI.

28.17 Distributed Clock Registers

28.17.1 DC Receive Time Registers

28.17.1.1 DC_RCV_TIME_PORT0 : Receive Time Port 0 Register

Base address: ESC = 0x8013_0000

Offset address: 0x0900

Bit position:	31	0
Bit field:	RCVTIME0[31:0]	
Value after reset:	x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x x	

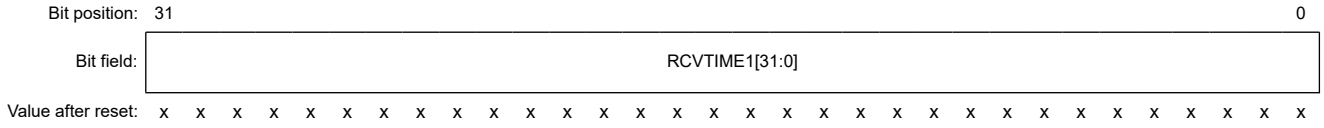
Bit	Symbol	Function	R/W
31:0	RCVTIME0[31:0]	Receive Time Indication/Latch The time stamps cannot be read in the same frame in which this register was written. The access type from the EtherCAT master is R/W. Write: A write access to this register with the BWR or FPWR command latches the local time when each port starts to receive a frame (first start bit of preamble). Read: Indicate the local time at the beginning of reception of the last frame containing write access to this register.	R

Writing to the DC_RCV_TIME_PORT0 register latches the received time of frames at all ports and reading this register indicates the received time of a frame latched at port 0.

28.17.1.2 DC_RCV_TIME_PORT1 : Receive Time Port 1 Register

Base address: ESC = 0x8013_0000

Offset address: 0x0904



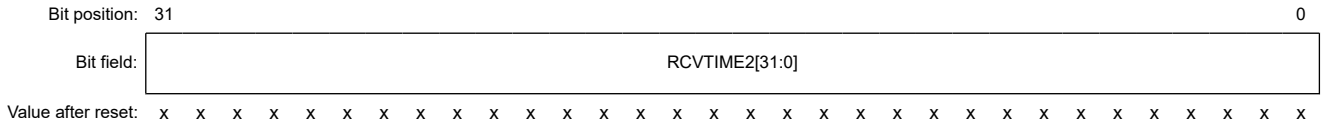
Bit	Symbol	Function	R/W
31:0	RCVTIME1[31:0]	Receive Time Indication Indicate the local time when port 1 starts to receive a frame (first start bit of preamble) containing the BWR or FPWR command to the receive time port 0 register (DC_RCV_TIME_PORT0 at 0x0900).	R

The DC_RCV_TIME_PORT1 register indicates the received time of the frame latched at port 1.

28.17.1.3 DC_RCV_TIME_PORT2 : Receive Time Port 2 Register

Base address: ESC = 0x8013_0000

Offset address: 0x0908



Bit	Symbol	Function	R/W
31:0	RCVTIME2[31:0]	Receive Time Indication Indicate the local time when port 2 starts to receive a frame (first start bit of preamble) containing the BWR or FPWR command to the receive time port 0 register (DC_RCV_TIME_PORT0 at 0x0900).	R

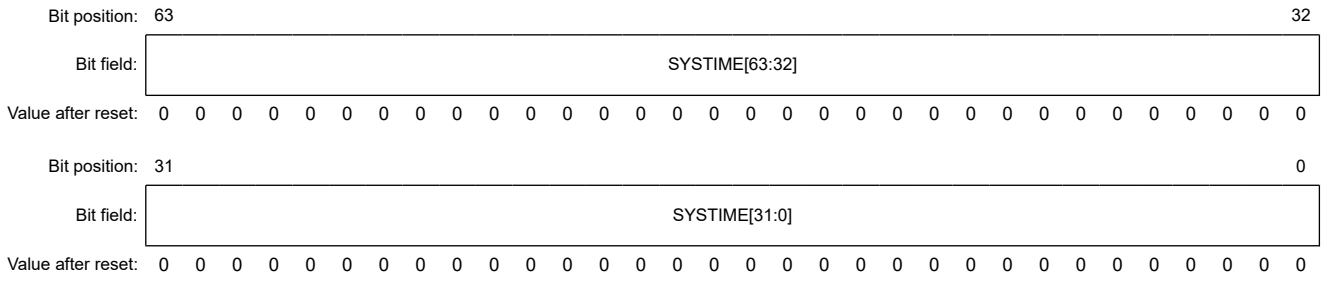
The DC_RCV_TIME_PORT2 register indicates the received time of the frame latched at port 2.

28.17.2 DC Time Loop Control Unit Registers

28.17.2.1 DC_SYS_TIME : System Time Register

Base address: ESC = 0x8013_0000

Offset address: 0x0910



Bit	Symbol	Function	R/W
63:0	SYSTIME[63:0]	System Time Indication Access from the ECAT (read-write) Read: Indicate the local copy of the system time when the frame passed the reference clock (such as including a system time delay). The time latched at the start of frame (SOF) is indicated. Write: A written value is compared with the local copy of the system time. The result is input to the time control loop unit. Access from the PDI (read-only) Indicate the local copy of the system time. The time latched when the first byte of this register was read is indicated.	R

The DC_SYS_TIME register indicates the local copy of the system time.

28.17.2.2 DC_RCV_TIME_UNIT : Receive Time ECAT Processing Unit Register

Base address: ESC = 0x8013_0000

Offset address: 0x0918

Bit position:	63																												32
Bit field:	RCVTIMEEPU[63:32]																												
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit position:	31																												0
Bit field:	RCVTIMEEPU[31:0]																												
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
63:0	RCVTIMEEPU[63:0]	Receive Time Indication This register indicates the local time at the beginning of reception by the EtherCAT processing unit of a frame (such as the first start bit of the preamble), including write access to the Receive Time Port 0 register (DC_RCV_TIME_PORT0 at 0x0900).	R

Note: If port 0 is open, the value in this register reflects the same time as the value in the Receive Time Port 0 register (DC_RCV_TIME_PORT0 at 0x0900), but as 64 bits.

The DC_RCV_TIME_UNIT register indicates the received time of a frame latched at EtherCAT processing unit.

28.17.2.3 DC_SYS_TIME_OFFSET : System Time Offset Register

Base address: ESC = 0x8013_0000

Offset address: 0x0920

Bit position:	63																												32
Bit field:	SYSTIMOFST[63:32]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31																												0
Bit field:	SYSTIMOFST[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

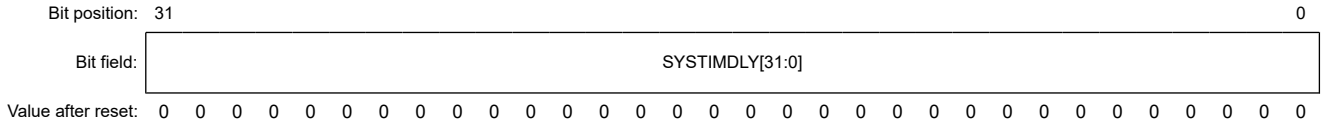
Bit	Symbol	Function	R/W
63:0	SYSTIMOFST[63:0]	System Time and Local Time Difference Indication Indicate a difference between the local time and system time. This offset is added to the local time to obtain the local system time. The access type from the EtherCAT master is R/W.	R

The DC_SYS_TIME_OFFSET register is used to indicate a difference (offset) between the local time and system time.

28.17.2.4 DC_SYS_TIME_DELAY : System Time Delay Register

Base address: ESC = 0x8013_0000

Offset address: 0x0928



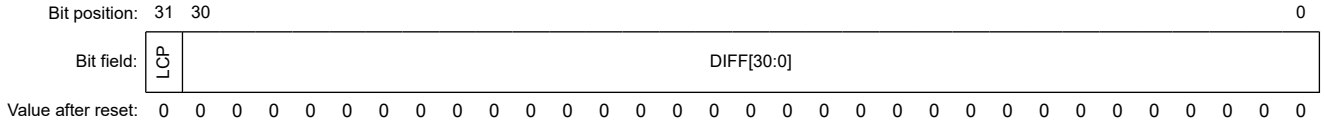
Bit	Symbol	Function	R/W
31:0	SYSTIMDLY[31:0]	Propagation Delay Indication Indicate a delay between the reference clock and the ESC. The access type from the EtherCAT master is R/W.	R

The DC_SYS_TIME_DELAY register indicates a propagation delay between the reference clock and slave (ESC).

28.17.2.5 DC_SYS_TIME_DIFF : System Time Difference Register

Base address: ESC = 0x8013_0000

Offset address: 0x092C



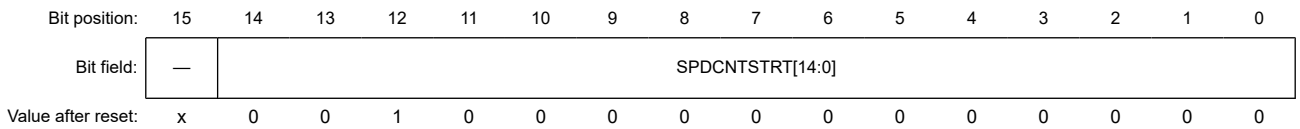
Bit	Symbol	Function	R/W
30:0	DIFF[30:0]	System Time Mean Difference Indication Indicates a mean difference between the local copy of the system time and received system time.	R
31	LCP	System Time Greater/Less Indication Indicates whether the local copy of the system time is greater than or equal to, or is less than, the latest received copy of the system time. 0: Local copy of the system time greater than or equal to the received system time 1: Local copy of the system time less than the received system time	R

The DC_SYS_TIME_DIFF register indicates a mean difference between the local copy of the system time and received system time.

28.17.2.6 DC_SPEED_COUNT_START : Speed Counter Start Register

Base address: ESC = 0x8013_0000

Offset address: 0x0930



Bit	Symbol	Function	R/W
14:0	SPDCNTSTRT[14:0]	Drift Correction Bandwidth Setting Indicate the bandwidth for adjustment of the local copy of the system time (larger values → smaller bandwidth and smoother adjustment). A write access resets the System Time Difference register (DC_SYS_TIME_DIFF at 0x092C) and the Speed Counter Difference register (DC_SPEED_COUNT_DIFF at 0x0932). Valid range: 0x0080 to 0x3FFF The access type from the EtherCAT master is R/W.	R
15	—	The read value is undefined. The access type from the EtherCAT master is R/W. In this case, the write value should be 0.	R

The DC_SPEED_COUNT_START register is used to set the bandwidth for drift correction of the local copy of the system time.

28.17.2.7 DC_SPEED_COUNT_DIFF : Speed Counter Difference Register

Base address: ESC = 0x8013_0000

Offset address: 0x0932

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SPDCNTDIFF[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SPDCNTDIFF[15:0]	Clock Period Deviation Indication Indicate the deviation between the local clock period and the clock period of the reference clock (represented by two's complements). Range: ± (speed counter start value - 0x7F)	R

The DC_SPEED_COUNT_DIFF register indicates the deviation between the local clock period and the clock period of the reference clock.

28.17.2.8 DC_SYS_TIME_DIFF_FIL_DEPTH : System Time Difference Filter Depth Register

Base address: ESC = 0x8013_0000

Offset address: 0x0934

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	SYSTIMDEP[3:0]			
Value after reset:	x	x	x	x	0	1	0	0

Bit	Symbol	Function	R/W
3:0	SYSTIMDEP[3:0]	Filter Depth Setting Set the filter depth for averaging the received system time deviation. A write access resets the System Time Difference register (DC_SYS_TIME_DIFF at 0x092C). The access type from the EtherCAT master is R/W.	R
7:4	—	The read values are undefined. The access type from the EtherCAT master is R/W. In this case, the write value should be 0.	R

The DC_SYS_TIME_DIFF_FIL_DEPTH register is used to set the filter depth for averaging the received system time deviation.

28.17.2.9 DC_SPEED_COUNT_FIL_DEPTH : Speed Counter Filter Depth Register

Base address: ESC = 0x8013_0000

Offset address: 0x0935

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	CLKPERDEP[3:0]			
Value after reset:	x	x	x	x	1	1	0	0

Bit	Symbol	Function	R/W
3:0	CLKPERDEP[3:0]	Filter Depth Setting Set the filter depth for averaging the clock period deviation. A write access resets the internal speed counter filter. The access type from the EtherCAT master is R/W.	R
7:4	—	The read values are undefined. The access type from the EtherCAT master is R/W. In this case, the write value should be 0.	R

The DC_SPEED_COUNT_FIL_DEPTH register is used to set the filter depth for averaging the clock period deviation.

28.17.3 Cyclic Unit Control Registers

28.17.3.1 DC_CYC_CONT : Cyclic Unit Control Register

Base address: ESC = 0x8013_0000

Offset address: 0x0980

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	LATCH1	LATCH0	—	—	—	SYNCOUT
Value after reset:	x	x	0	0	x	x	x	0

Bit	Symbol	Function	R/W
0	SYNCOUT	SYNC Output Unit Control Setting The access type from the EtherCAT master is R/W. 0: ECAT control 1: PDI control	R
3:1	—	The read values are undefined.	R
4	LATCH0	Latch Input Unit 0 Control Setting Latch interrupt is routed to the ECAT or PDI in accordance with this setting. The access type from the EtherCAT master is R/W. 0: ECAT control 1: PDI control	R
5	LATCH1	Latch Input Unit 1 Control Setting Latch interrupt is routed to the ECAT or PDI in accordance with this setting. The access type from the EtherCAT master is R/W. 0: ECAT control 1: PDI control	R
7:6	—	The read values are undefined.	R

The DC_CYC_CONT register sets whether to control SYNC and latch units by the ECAT or PDI.

28.17.4 SYNC Output Unit Registers

28.17.4.1 DC_ACT : Activation Register

Base address: ESC = 0x8013_0000

Offset address: 0x0981

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DBGPULSE	NEARFUTURE	STARTTIME	EXTSTARTTIME	AUTOACT	SYNC1	SYNC0	SYNCACT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SYNCACT	Sync Output Unit Activation Activates the Sync output unit. Write 1 after the start time is written. 0: Deactivated 1: Activated	R/W
1	SYNC0	SYNC0 Output Setting Sets SYNC0 output. 0: Deactivated 1: SYNC0 pulse output is generated	R/W
2	SYNC1	SYNC1 Output Setting Sets SYNC1 output. 0: Deactivated 1: SYNC1 pulse output is generated	R/W
3	AUTOACT	SYNC Output Unit Activation Sets whether to activate the Sync output unit automatically by writing to the Start Time Cyclic Operation register (DC_CYC_START_TIME at 0x0990): 0: Deactivated 1: Activated. Bit 0 is automatically set to 1 in this register after the start time is written.	R/W
4	EXTSTARTTIME	Start Time Cyclic Operation Extension Extends start time cyclic operation. 0: No extension 1: Extends the start time written with 32 bits to 64 bits	R/W
5	STARTTIME	Start Time Plausibility Selects whether checking the plausibility of the start time and response to implausible start times is to proceed. 0: Disabled. Sync signal is generated if the start time is reached. 1: Sync signal is generated immediately if the start time is outside the range of the near future.	R/W
6	NEARFUTURE	Near Future Range Setting Sets the range to be considered the near future. 0: Up to 2^{63} ns from now (1/2 of the DC width) 1: Up to 2^{31} ns from now (about 2.1 sec.)	R/W
7	DBGPULSE	Debug Pulse Setting Sets Sync signal debug pulse. This bit is self-cleared and always read as 0. 0: Deactivated 1: Immediately generates a single debug ping on the SYNC0 and SYNC1 pins in accord with the setting of bits 2 and 1 of this register.	R/W

Note: Writing to this register depends on the setting of bit 0 of the Cyclic Unit Control register (DC_CYC_CONT at 0x0980).

The DC_ACT register is used to activate the Sync output unit.

28.17.4.2 DC_PULSE_LEN : SYNC Signal Pulse Length Register

Base address: ESC = 0x8013_0000

Offset address: 0x0982

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PULSELEN[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PULSELEN[15:0]	SYNC Signal Pulse Length Indication Indicate the pulse length of SYNC signals (in units of 10 ns) 0: Acknowledge mode. In this mode, SYNC signal is cleared by reading the SYNC0 or SYNC1 status register (DC_SYNC0/1_STAT at 0x098E, 0x098F).	R

The DC_PULSE_LEN register indicates the pulse length of SYNC signals.

28.17.4.3 DC_ACT_STAT : Activation Status Register

Base address: ESC = 0x8013_0000

Offset address: 0x0984

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	START TIME	SYNC 1ACT	SYNC 0ACT
Value after reset:	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
0	SYNC0ACT	SYNC0 Status Indication Indicates the activation state of SYNC0. 0: First SYNC0 pulse is not pending 1: First SYNC0 pulse is pending	R
1	SYNC1ACT	SYNC1 Status Indication Indicates the activation state of SYNC1. 0: First SYNC1 pulse is not pending 1: First SYNC1 pulse is pending	R
2	STARTTIME	Plausibility Result Indication Indicates the plausibility check result of the Start Time Cyclic Operation register (DC_CYC_START_TIME at 0x0990) while the Sync output unit is activated. 0: The start time is within the near future 1: The start time is out of the near future	R
7:3	—	The read values are undefined.	R

The DC_ACT_STAT register indicates the activation status of SYNC output signals.

28.17.4.4 DC_SYNC0_STAT : SYNC0 Status Register

Base address: ESC = 0x8013_0000

Offset address: 0x098E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SYNC 0STA
Value after reset:	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	SYNC0STA	SYNC0 State Indication Indicates the SYNC0 state for acknowledge mode. SYNC0 in acknowledge mode is cleared by reading this register from the PDI. This bit is only used in acknowledge mode.	R
7:1	—	The read values are undefined.	R

The DC_SYNC0_STAT register indicates the state of SYNC0 output. It is only used in acknowledge mode.

28.17.4.5 DC_SYNC1_STAT : SYNC1 Status Register

Base address: ESC = 0x8013_0000

Offset address: 0x098F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SYNC 1STA
Value after reset:	x	x	x	x	x	x	x	0

Bit	Symbol	Function	R/W
0	SYNC1STA	SYNC1 State Indication Indicates the SYNC1 state for acknowledge mode. SYNC1 in acknowledge mode is cleared by reading this register from the PDI. This bit is only used in acknowledge mode.	R
7:1	—	The read values are undefined.	R

The DC_SYNC1_STAT register indicates the state of SYNC1 output. It is only used in acknowledge mode.

28.17.4.6 DC_CYC_START_TIME : Start Time Cyclic Operation/Next SYNC0 Pulse Register

Base address: ESC = 0x8013_0000

Offset address: 0x0990

Bit position:	63	62	61	60	59	58	57	56	55	54	53	52	51	50	49	48	47	46	45	44	43	42	41	40	39	38	37	36	35	34	33	32
Bit field:	STATIM[63:32]																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	STATIM[31:0]																															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
63:0	STATIM[63:0]	Start Time Setting/System Time Indication Write: Set the start time (in the system time) of cyclic operation in ns units. Read: Indicate the system time of the next SYNC0 pulse in ns units.	R/W

Note:
- Writing to this register depends on the setting of bit 0 of the Cyclic Unit Control register (DC_CYC_CONT at 0x0980). Only writable when bit 0 is 0 in the SYNC Activation register (DC_ACT at 0x0981).
- When auto-activation is enabled, the upper 32 bits are automatically extended if only the lower 32 bits are written within one frame.

Writing to this register sets the start time of cyclic operation. Reading this register indicates the system time of the next SYNC0 pulse.

28.17.4.7 DC_NEXT_SYNC1_PULSE : Next SYNC1 Pulse Register

Base address: ESC = 0x8013_0000

Offset address: 0x0998

Bit position:	63																												32
Bit field:	SYNC1PULSE[63:32]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	31																												0
Bit field:	SYNC1PULSE[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
63:0	SYNC1PULSE[63:0]	SYNC1 Pulse System Time Indication Indicate the system time of the next SYNC1 pulse in ns units.	R

The DC_NEXT_SYNC1_PULSE register indicates the system time of the next SYNC1 pulse.

28.17.4.8 DC_SYNC0_CYC_TIME : SYNC0 Cycle Time Register

Base address: ESC = 0x8013_0000

Offset address: 0x09A0

Bit position:	31																												0
Bit field:	SYNC0CYC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SYNC0CYC[31:0]	Time Between Consecutive SYNC0 Pulses Set the time between two consecutive SYNC0 pulses in ns units. 0: Single-shot mode. Only one SYNC0 pulse is generated in single-shot mode.	R/W

Note: Writing to this register depends on the setting of bit 0 of the Cyclic Unit Control register (DC_CYC_CONT at 0x0980).

The DC_SYNC0_CYC_TIME register is used to set the time between consecutive SYNC0 pulses.

28.17.4.9 DC_SYNC1_CYC_TIME : SYNC1 Cycle Time Register

Base address: ESC = 0x8013_0000

Offset address: 0x09A4

Bit position:	31																												0
Bit field:	SYNC1CYC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SYNC1CYC[31:0]	Time between SYNC1 and SYNC0 Pulses Set the time between SYNC1 and SYNC0 pulses in ns units.	R/W

Note: Writing to this register depends on the setting of bit 0 of the Cyclic Unit Control register (DC_CYC_CONT at 0x0980).

The DC_SYNC1_CYC_TIME register is used to set the time between SYNC1 and SYNC0 pulses.

28.17.5 Latch Input Unit Registers

28.17.5.1 DC_LATCH0_CONT : Latch 0 Control Register

Base address: ESC = 0x8013_0000

Offset address: 0x09A8

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	NEGEDGE	POSEDGE
Value after reset:	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
0	POSEDGE	Latch 0 Positive Edge Function Setting Sets the function of the rising edge of the latch 0 input signal. 0: Continuous latch active 1: Single event (only first event active)	R/W
1	NEGEDGE	Latch 0 Negative Edge Function Setting Sets the function of the falling edge of the latch 0 input signal. 0: Continuous latch active 1: Single event (only first event active)	R/W
7:2	—	The read values are undefined. The write value should be 0. The access type from the EtherCAT master is R.	R/W

Note: Writing to this register depends on the setting of bit 4 of the Cyclic Unit Control register (DC_CYC_CONT at 0x0980).

The DC_LATCH0_CONT register is used to control the edge function of the latch 0 input signal.

28.17.5.2 DC_LATCH1_CONT : Latch 1 Control Register

Base address: ESC = 0x8013_0000

Offset address: 0x09A9

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	NEGEDGE	POSEDGE
Value after reset:	x	x	x	x	x	x	0	0

Bit	Symbol	Function	R/W
0	POSEDGE	Latch 1 Positive Edge Function Setting Sets the function of the rising edge of the latch 1 input signal. 0: Continuous latch active 1: Single event (only first event active)	R/W
1	NEGEDGE	Latch 1 Negative Edge Function Setting Sets the function of the falling edge of the latch 1 input signal. 0: Continuous Latch active 1: Single event (only first event active)	R/W
7:2	—	The read values are undefined. The write value should be 0. The access type from the EtherCAT master is R.	R/W

Note: Writing to this register depends on the setting of bit 5 of the Cyclic Unit Control register (DC_CYC_CONT at 0x0980).

The DC_LATCH1_CONT register is used to control the edge function of the latch 1 input signal.

28.17.5.3 DC_LATCH0_STAT : Latch 0 Status Register

Base address: ESC = 0x8013_0000

Offset address: 0x09AE

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	PINSTATE	EVENTNEG	EVENTPOS
Value after reset:	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
0	EVENTPOS	Latch 0 Positive Edge Event Indication Indicates detection of rising edges of the event latch 0 signal. This flag is cleared by reading the Latch 0 Time Positive Edge register (DC_LATCH0_TIME_POS at 0x09B0). 0: Rising edge not detected or continuous mode 1: Rising edge detected and mode is single-event	R
1	EVENTNEG	Latch 0 Negative Edge Event Indication Indicates detection of falling edges of the event latch 0 signal. This flag is cleared by reading the Latch 0 Time Negative Edge register (DC_LATCH0_TIME_NEG at 0x09B8). 0: Falling edge not detected or continuous mode 1: Falling edge detected and mode is single-event	R
2	PINSTATE	Latch 0 Input Pin State Indication Indicates the state of the latch 0 input pin.	R
7:3	—	The read values are undefined.	R

The DC_LATCH0_STAT register indicates the state of the latch 0 input signal.

28.17.5.4 DC_LATCH1_STAT : Latch 1 Status Register

Base address: ESC = 0x8013_0000

Offset address: 0x09AF

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	PINSTATE	EVENTNEG	EVENTPOS
Value after reset:	x	x	x	x	x	0	0	0

Bit	Symbol	Function	R/W
0	EVENTPOS	Latch 1 Positive Edge Event Indication Indicates detection of rising edges of the event latch 1 signal. This flag is cleared by reading the Latch 1 Time Positive Edge register (DC_LATCH1_TIME_POS at 0x09C0). 0: Rising edge not detected or continuous mode 1: Rising edge detected and mode is single-event	R
1	EVENTNEG	Latch 1 Negative Edge Event Indication Indicates detection of falling edges of the event latch 1 signal. This flag is cleared by reading the Latch 1 Time Negative Edge register (DC_LATCH1_TIME_NEG at 0x09C8). 0: Falling edge not detected or continuous mode 1: Falling edge detected and mode is single-event	R
2	PINSTATE	Latch 1 Input Pin State Indication Indicates the state of the latch 1 input pin.	R
7:3	—	The read values are undefined.	R

The DC_LATCH1_STAT register indicates the state of the latch 1 input signal.

28.17.5.5 DC_LATCH0_TIME_POS : Latch 0 Time Positive Edge Register

Base address: ESC = 0x8013_0000

Offset address: 0x09B0

Bit position:	63																												32
Bit field:	SYSTIME[63:32]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31																												0
Bit field:	SYSTIME[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
63:0	SYSTIME[63:0]	System Time Indication Indicate the system time captured at the rising edge of the latch 0 input signal. Reading this register clears bit 0 of the Latch 0 Status register (DC_LATCH0_STAT at 0x09AE).	R

Note:
- Bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value
- Clearing the latch 0 status flag function depends on the setting of bit 4 of the Cyclic Unit Control register (DC_CYC_CONT at 0x0980).

The DC_LATCH0_TIME_POS register indicates the system time at the rising edge of the latch 0 input signal.

28.17.5.6 DC_LATCH0_TIME_NEG : Latch 0 Time Negative Edge Register

Base address: ESC = 0x8013_0000

Offset address: 0x09B8

Bit position:	63																												32
Bit field:	SYSTIME[63:32]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	31																												0
Bit field:	SYSTIME[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
63:0	SYSTIME[63:0]	System Time Indication Indicate the system time captured at the falling edge of the latch 0 input signal. Reading this register clears bit 1 of the Latch 0 Status register (DC_LATCH0_STAT at 0x09AE).	R

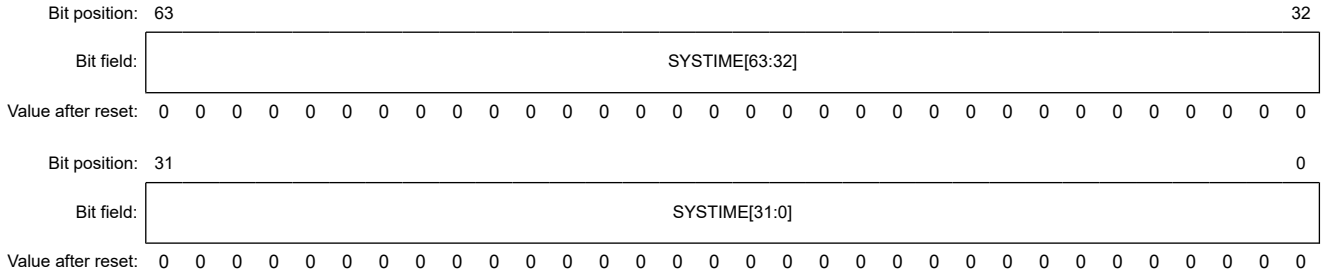
Note:
- Bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value
- Clearing the latch 0 status flag function depends on the setting of bit 4 of the Cyclic Unit Control register (DC_CYC_CONT at 0x0980).

The DC_LATCH0_TIME_NEG register indicates the system time at the falling edge of the latch 0 input signal.

28.17.5.7 DC_LATCH1_TIME_POS : Latch 1 Time Positive Edge Register

Base address: ESC = 0x8013_0000

Offset address: 0x09C0



Bit	Symbol	Function	R/W
63:0	SYSTIME[63:0]	System Time Indication Indicate the system time captured at the rising edge of the latch 1 input signal. Reading this register clears bit 0 of the Latch 1 Status register (DC_LATCH1_STAT at 0x09AF).	R

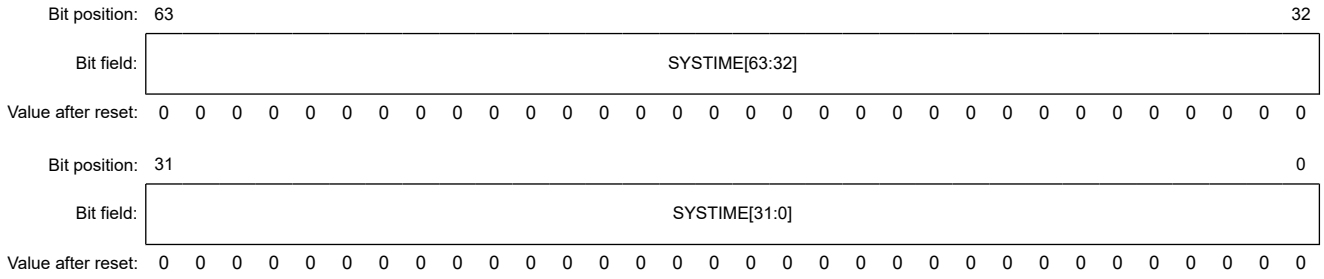
Note:
- Bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value
- Clearing the latch 1 status flag function depends on the setting of bit 5 of the Cyclic Unit Control register (DC_CYC_CONT at 0x0980).

The DC_LATCH1_TIME_POS register indicates the system time at the rising edge of the latch 1 input signal.

28.17.5.8 DC_LATCH1_TIME_NEG : Latch 1 Time Negative Edge Register

Base address: ESC = 0x8013_0000

Offset address: 0x09C8



Bit	Symbol	Function	R/W
63:0	SYSTIME[63:0]	System Time Indication Indicate the system time captured at the falling edge of the latch 1 input signal. Reading this register clears bit 1 of the Latch 1 Status register (DC_LATCH1_STAT at 0x09AF).	R

Note:
- Bits [63:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value
- Clearing the latch 1 status flag function depends on the setting of bit 5 of the Cyclic Unit Control register (DC_CYC_CONT at 0x0980).

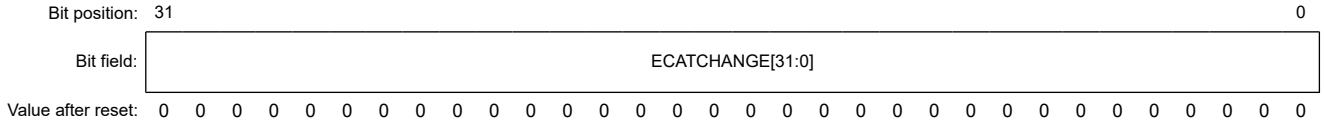
The DC_LATCH1_TIME_NEG register indicates the system time at the falling edge of the latch 1 input signal.

28.17.6 SyncManager Event Time Registers

28.17.6.1 DC_ECAT_CNG_EV_TIME : Buffer Change Event Time Register

Base address: ESC = 0x8013_0000

Offset address: 0x09F0



Bit	Symbol	Function	R/W
31:0	ECATCHANGE[31:0]	Local Time Indication Indicate the local time at the beginning of a frame which causes at least one SyncManager to generate an ECAT event (switching the buffers).	R

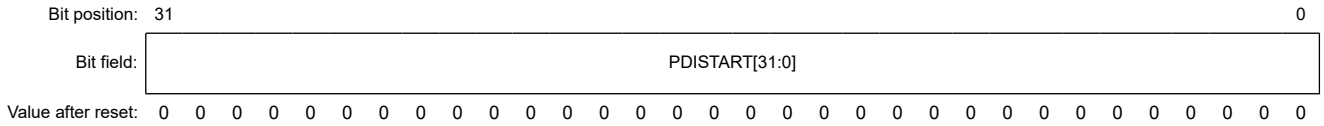
Note: Bits [31:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

The DC_ECAT_CNG_EV_TIME register indicates the local time at the beginning of a frame which causes SyncManager to generate an ECAT event (switching the buffers).

28.17.6.2 DC_PDI_START_EV_TIME : PDI Buffer Start Event Time Register

Base address: ESC = 0x8013_0000

Offset address: 0x09F8



Bit	Symbol	Function	R/W
31:0	PDISTART[31:0]	Local Time Indication Indicate the local time when at least one SyncManager has generated a PDI event (access to the address where a buffer starts).	R

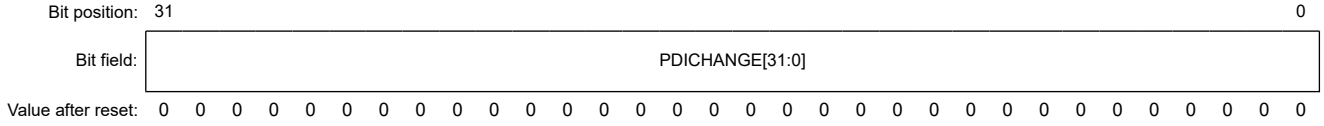
Note: Bits [31:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

The DC_PDI_START_EV_TIME register indicates the local time when SyncManager has generated a PDI event (access to the address where a buffer starts).

28.17.6.3 DC_PDI_CNG_EV_TIME : PDI Buffer Change Event Time Register

Base address: ESC = 0x8013_0000

Offset address: 0x09FC



Bit	Symbol	Function	R/W
31:0	PDICHANGE[31:0]	Local Time Indication Indicate the local time when at least one SyncManager has generated a PDI event (switching the buffers).	R

Note: Bits [31:8] are internally latched (ECAT/PDI independently) when bits [7:0] are read, which guarantees reading a consistent value.

The DC_PDI_CNG_EV_TIME register indicates the local time when SyncManager has generated a PDI event (switching the buffers).

28.18 Other Registers

28.18.1 PRODUCT_ID : Product ID Register

Base address: ESC = 0x8013_0000

Offset address: 0x0E00

Bit position:	63																													32
Bit field:	PROID[63:32]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit position:	31																													0
Bit field:	PROID[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
63:0	PROID[63:0]	Product ID Indication	R

The PRODUCT_ID register indicates the product ID.

28.18.2 VENDOR_ID : Vendor ID Register

Base address: ESC = 0x8013_0000

Offset address: 0x0E08

Bit position:	63																													32
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
Bit position:	31																													0
Bit field:	VENDORID[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	VENDORID[31:0]	Vendor ID Indication	R
63:32	—	The read values are undefined.	R

The VENDOR_ID register indicates the vendor ID.

28.18.3 USER_RAM : User RAM

This area of RAM indicates the supported features dependent on the IP core configuration and takes up the 128 bits from 0x8013_0F80 to 0x8013_0FFF. An initial value of 1 means that the implementation of this module supports the corresponding feature, except in the case of bits [7:0], which indicate the number of bits defined in the user RAM and the initial value is 0x33 for this module.

Bit position	Description	Initial value
7:0	Number of bits for defining extended functionality. The value is 51 (0x33) for this module.	0x33
8	Extended DL Control register (0x0102, 0x0103)	1

Bit position	Description	Initial value
9	AL Status Code register (0x0134, 0x0135)	1
10	ECAT event mask (0x0200, 0x0201)	1
11	Configured station alias (0x0012, 0x0013)	1
12	General input (0x0F18, 0x0F1F)	0
13	General output (0x0F10, 0x0F17)	0
14	AL event mask (0x0204, 0x0207)	1
15	Physical read/write offset (0x0108, 0x0109)	1
16	Watchdog divider writable (0x0400, 0x0401) and watchdog PDI (0x0410, 0x0F11)	1
17	Watchdog counter (0x0442, 0x0443)	1
18	Write protection (0x0020, 0x0031)	1
19	Reset (0x0040, 0x0041)	1
20	Reserved	0
21	DC SyncManager event time (0x09F0, 0x09FF)	1
22	ECAT processing unit/PDI error counter (0x030C, 0x030D)	1
23	EEPROM size configurable (bit 7 at 0x0502) 0: EEPROM size fixed up to 16 Kbits 1: EEPROM size configurable	1
26:24	Reserved	0
27	Lost link counter (0x0310, 0x0313)	1
28	MII management interface (0x0510, 0x0515)	1
29	Enhanced link detection MII	1
30	Enhanced link detection EBUS	0
31	RUN LED	1
32	Link/activity LED	1
33	Reserved	0
35:34	Reserved	1
36	Reserved	0
37	Reserved	1
38	DC Time loop control assigned to PDI	0
39	Link detection and configuration by MI	0
40	MI control by PDI	1
41	Automatic TX shift	1
42	EEPROM emulation	0
49:43	Reserved	0
50	ERR LED, RUN/ERR LED override	1
Others	Reserved	Reserved

28.18.4 DATA_RAM : Process Data RAM

The process data RAM is used for process data and mailbox and takes up 8 KB from 0x8013_1000 to 0x8013_2FFF.

This RAM is only accessible when the EEPROM is correctly loaded (for example, when bit 0 is 1 in the ESC DL Status register, ESC_DL_STATUS, at 0x0110).

28.18.5 Setting the Module-Stop Function

EtherCAT modules are stopped in the initial state. If the modules are to be used, set the MSTPCRE.MSTPCRE02 bit for release from the module-stop state. Resetting the module-stop state is prohibited following release from the module-stop state. Operation of the modules if they are stopped following release and then released again is not guaranteed. Release from the module-stop state can proceed again after the modules have been returned to their initial states (stopped) by a reset.

28.19 Initial Settings

To initialize the EtherCAT, follow the procedure below.

- Set the offset address of the PHY module in the CATOFFADD register.
- Set the size of the EEPROM in the CATEMMD register.
- Set the delay time of the TXC pin in the CATTXCSFT register.
- Release the EtherCAT from the module-stop state by using the MSTPCRE02 bit in the MSTPCRE register.
- Release the ESC and PHY modules from the reset state by using the MRCTLE03 and MRCTLE04 bits in the MRCTLE register.

On release from the reset state, the ESC will automatically load the EEPROM data and be activated.

28.20 Configuration of the Reset Circuit

Figure 28.2 shows the configuration of the reset circuit of the EtherCAT Slave Controller (ESC). When reset request by ESC_RESET_ECAT (0x0040) or ESC_RESET_PDI (0x0041) is received, ESC stops and reset output from ESC becomes high level. At the same time, ESC_RST interrupt is generated and ESC_RESETOUT# pin outputs low level.

To release reset state of ESC, MRCTLE04 bit of MRCTLE register must be changed from 0 → 1 → 0 after ESC_RST interrupt is detected. When reset input to ESC is deasserted, reset output from ESC becomes 0 and simultaneously ESC is rebooted with loading EEPROM. It takes around 1ms to finish EEPROM loading. Adjust clear timing of MRCTLE04 bit of MRCTLE register so that Ethernet PHY have enough reset duration. Figure 28.3 shows the timing diagram.

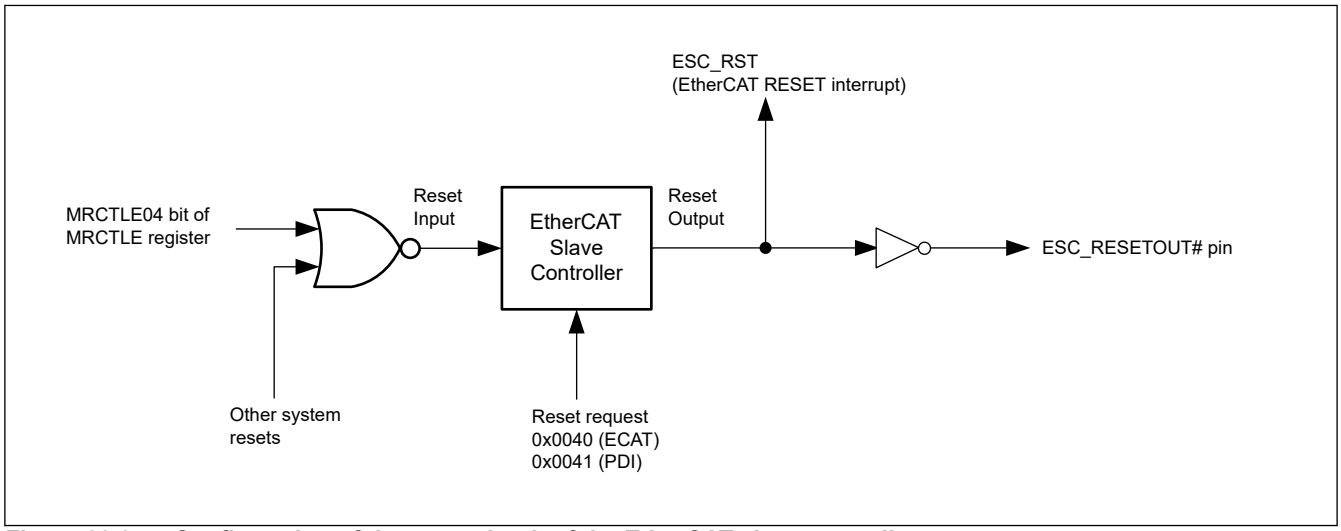


Figure 28.2 Configuration of the reset circuit of the EtherCAT slave controller

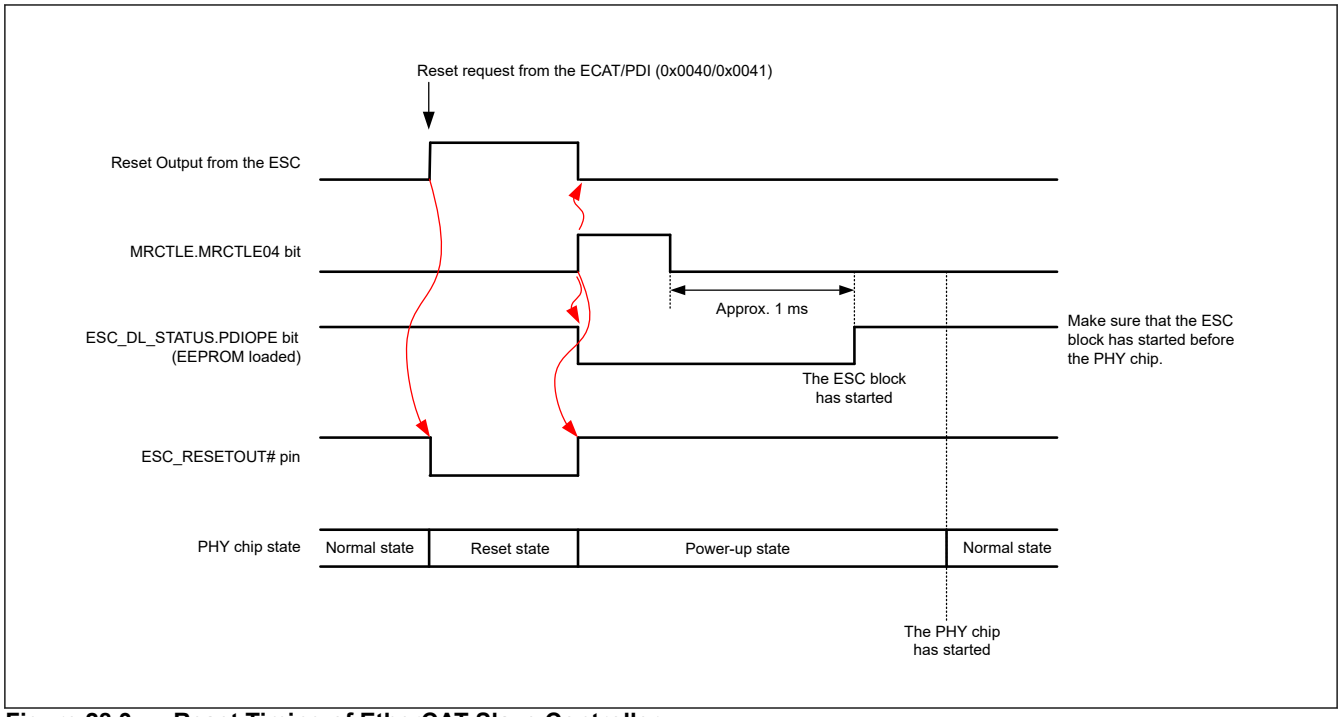


Figure 28.3 Reset Timing of EtherCAT Slave Controller

28.21 Usage Notes

1. Even if write protection is enabled, 0x0138 can be written.
2. When consecutive commands were issued from PDI to EEPROM, the latter command may not be accepted. Issue the commands with enough interval (e.g. 2 µs).
3. When VLAN tagged frame is used, ECAT reset register 0x0040 cannot be written.
4. ESC DL Control (0x0101) loop setting Auto-Close (01): if port waits for write access to be opened, the temporary loop bit (0x0100[1]) is not taken into account when a write command to ESC DL Control occurs.
5. Changing SyncSignal cycle times during activation may lead to extremely long cycle times. Additionally, changing cycle time from PDI may result in unwanted cycle times or extremely long delays. When cycle times are changed, disable SyncSignal.
6. DC Sync1 is missed once, when single shot is used, then DC SyncSignal generation deactivated, then cyclic generation activated.
7. Incorrect station alias address may be used when 0x0012 is changed and frame is received while change of 0x0013 is not completed yet.

29. USB 2.0 Host Module (USBHC)

29.1 Overview

The USB module of this LSI is a dual-role device that has USB 2.0 host and function modules.

The LSI handles a single USB port for either host controller or function controller operation. The Common Control register (COMMCTRL) is used to switch between host controller and function controller operation.

Since operation as a host controller or a function controller is exclusive of each other, dynamic switching between the types of operation is not possible. This section describes host controller operation.

The USB 2.0 host controller:

- Conforms to Universal Serial Bus Specification Revision 2.0
- Conforms to Open Host Controller Interface (OHCI) Specification for USB Rev. 1.0a
- Conforms to Enhanced Host Controller Interface (EHCI) Specification for USB Rev.1.1*1
- Supports USB 2.0 high-speed (480 Mbps), full-speed (12 Mbps), and low-speed (1.5 Mbps) transfer
- Supports the USB 2.0 compliance test function.

Note 1. Hardware prefetching is not supported.

Figure 29.1 shows a block diagram of the USB module.

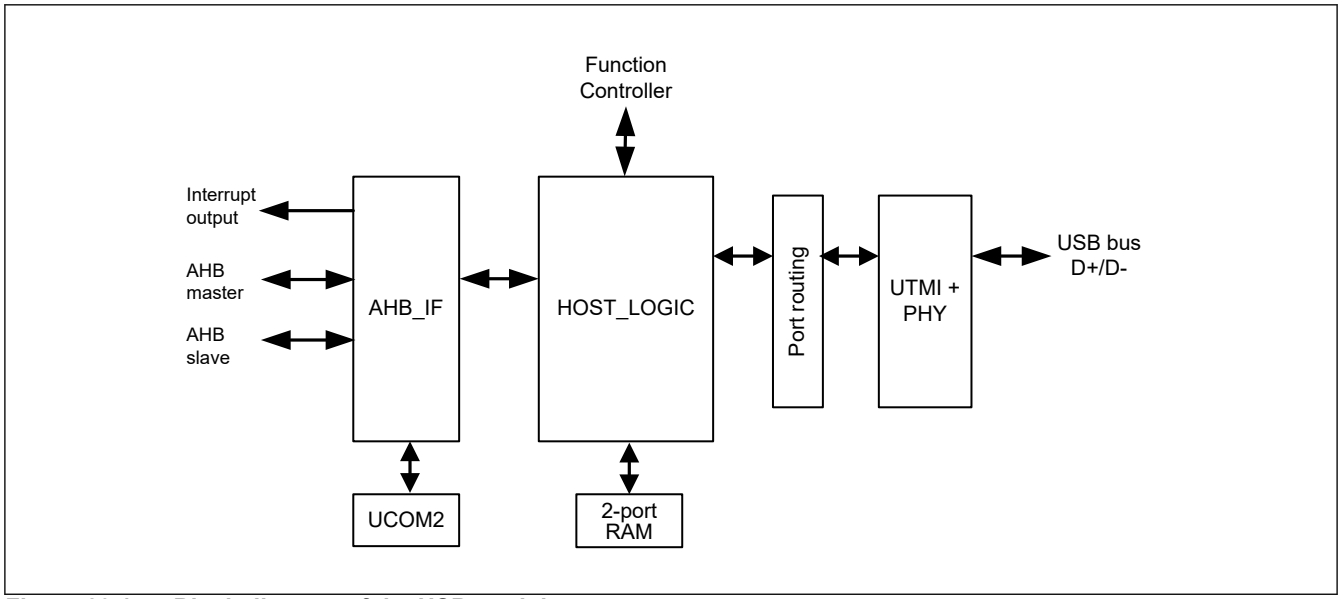


Figure 29.1 Block diagram of the USB module

(1) Host Logic

A USB 2.0 logic conforms to the EHCI and OHCI standards. It has control circuits such as a list processing circuit, a serial-parallel circuit, and a USB buffer. It executes high-speed, low-speed, and full-speed USB transfer.

(2) UTMI+ Transceiver

A USB 2.0 UTMI transceiver with an interface conforms to the UTMI+ standard.

29.1.1 Precautions on the Use of USB Host Controller

29.1.1.1 General Precautions

This section describes precautions on the use of the USB host controller.

1. Dynamic changes in PCLKM and USB_CLK48 are not supported except for the clock stop state.

2. When using isochronous transfer, evaluate performance at the application level to confirm that the expected performance is obtained.
3. The USB host controller outputs the logical OR of multiple internal interrupt signals. See section 29.4. Interrupts.
4. It may take a long time to actually clear an interrupt after a register is accessed to clear the interrupt. For the action necessary during this wait time, see section 29.4.3. Time Required to Clear Interrupt Signals.

29.1.1.2 AHB Interface

The read or written data is not guaranteed for register access while the clock is stopped.

29.1.1.3 Operating Procedure

See the operating procedure described in section 29.6.1. Sample of Initial Settings.

29.2 Register Descriptions

This section describes details of the register functions.

29.2.1 OHCI Operational Registers

Access the OHCI operational registers after starting up the PHY internal PLL. For details, see Figure 29.7.

29.2.1.1 HCREVISION : HcRevision Register

Base address: USBHC = 0x8020_0000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	REV[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
7:0	REV[7:0]	HCI revision These bits indicate the version of the HCI specifications implemented in the host logic. As the host logic conforms to OHCI standard 1.0a, this value is set to 0x10.	R
31:8	—	These bits are read as 0.	R

29.2.1.2 HCCONTROL : HcControl Register

Base address: USBHC = 0x8020_0000

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	RWE	RWC	IR	HCFS[1:0]		BLE	CLE	IE	PLE	CBSR[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CBSR[1:0]	Control/bulk transfer service ratio (ControlBulkServiceRatio) CBSR: Service ratio of bulk ED : Control ED 00b: 1:1 01b: 2:1 10b: 3:1 11b: 4:1	R/W
2	PLE	Periodic list setting (PeriodicListEnable) This bit specifies whether to perform periodic list processing. The setting of this bit takes effect from the next frame. 0: Periodic list processing is not done 1: Period list processing is done	R/W
3	IE	Isochronous ED processing setting (IsochronousEnable) This bit specifies whether to perform isochronous ED processing. The setting of this bit takes effect from the next frame. When an isochronous ED is found during list processing, the host logic checks this bit to determine whether to perform isochronous ED processing. 0: Isochronous ED processing is enabled 1: Isochronous ED processing is disabled	R/W
4	CLE	Control list processing setting (ControlListEnable) This bit specifies whether to perform control list processing. The setting of this bit takes effect from the next frame. The control list can be modified only when this value is 0. 0: Control list processing is not done 1: Control list processing is done	R/W
5	BLE	Bulk list processing setting (BulkListEnable) This bit specifies whether to perform bulk list processing. The setting of this bit takes effect from the next frame. The bulk list can be modified only when this value is 0. 0: Bulk list processing is not done 1: Bulk list processing is done	R/W
7:6	HCFS[1:0]	Host logic operation status (Host Controller FunctionalState) These bits indicate the operating state of the host logic. On entering the USB operational state, the host logic starts management of frames delimited at 1-ms intervals. The operating state is always controlled by software except for transition to the USB resume state due to remote wakeup in the USB suspend state. These bits indicate the USB reset state after a hardware reset. After a software reset, the state shifts to the USB suspend state. 0 0: USB Reset 0 1: USB Resume 1 0: USB Operational 1 1: USB Suspend	R/W
8	IR	HcInterruptStatus interrupt path setting (InterruptRouting) This bit sets the output path for the interrupts indicated in the HcInterruptStatus register. The value set for this bit has no meaning because there is only one OHCI interrupt path for this host logic.	R/W
9	RWC	Remote Wakeup support setting (RemoteWakeUpConnect) This bit indicates whether the host logic supports remote wakeup. To support remote wakeup by the system, set this bit during initialization. 0: Remote Wakeup is not supported 1: Remote Wakeup is supported	R/W
10	RWE	PME assertion control (RemoteWakeUpEnable) This host logic does not use the PME assertion control bit. The value set for this bit has no meaning in this host logic.	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

The HCCONTROL register sets the operation mode for the OHCI host logic.

29.2.1.3 HCCOMMANDSTATUS : HcCommandStatus Register

Base address: USBHC = 0x8020_0000

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	OCR	BLF	CLF	HCR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	HCR	Host logic software reset start (HostController Reset) This bit starts software reset of the host logic. When this bit is set, the USB suspend state is entered regardless of the functional state of the host logic.	W
1	CLF	Control list TD (ControlList Filled) This bit indicates whether TD exists in the control list. The host logic checks this bit when starting the processing of the start ED in the control list. When this bit is 0, the host logic does not begin list processing. When this bit is 1, the host logic starts control list processing and clears this bit to 0. On detecting TD, the host logic restores this bit to 1 and continues control list processing. When the host logic completes list processing, it clears this bit to 0. If TD is not found in the list or software does not set this bit to 1, this bit remains at 0 and list processing stops. When creating a list again and executing processing of the list, set this bit before starting list processing by setting bit 4 (CLE) in the HcControl register. 0: TD exists in the control list 1: TD does not exist in the control list	R/W
2	BLF	Bulk list TD (BulkListFilled) This bit indicates whether TD exists in the bulk list. The host logic checks this bit when starting the processing of the start ED in the bulk list. When this bit is 0, the host logic does not begin list processing. When this bit is 1, the host logic starts bulk list processing and clears this bit to 0. Upon detecting TD, the host logic restores this bit to 1 and continues bulk list processing. When the host logic completes list processing, it clears this bit to 0. If TD is not found in the list or software does not set this bit to 1, this bit remains at 0 and list processing stops. When creating a list again and executing processing of the list, set this bit before starting list processing by setting bit 5 (BLE) in the HcControl register. 0: TD exists in the bulk list 1: TD does not exist in the bulk list	R/W
3	OCR	Host logic control right change (OwnershipChangeRequest) This host logic does not use this bit, which changes the control right of the host logic. The value set for this bit has no meaning in this host logic.	W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
17:16	SOC[1:0]	Schedule overrun count (Scheduling OverrunCount) These bits count the number of times a schedule overrun occurs. This value is incremented every time a schedule overrun occurs. When the value reaches 11b, it returns to 00b. Counting continues even while bit 0 (SO) in the HcInterruptStatus register is set.	R
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The HCCOMMANDSTATUS register is used when the host logic receives commands from software or checks the current host logic status.

29.2.1.4 HCINTERRUPTSTATUS : HcInterruptStatus Register

Base address: USBHC = 0x8020_0000

Offset address: 0x00C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	OC	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	RHSC	FNO	UE	RD	SF	WDH	SO
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SO	USB schedule overrun (Scheduling Overrun) This is an interrupt bit to indicate that the USB schedule has overrun in a frame. When the USB schedule overruns, this bit is set after HccaFrameNumber for the next frame is updated. When this bit is set, bits [17:16] (SOC) in the HcCommandStatus register are also incremented. Writing 1 to this bit clears the interrupt. 0: No SO interrupt has occurred 1: SO interrupt has occurred	R/W
1	WDH	Host logic HccaDoneHead update (Writeback Done Head) This is an interrupt bit to indicate that the host logic has updated the contents of HccaDoneHead. The host logic sets this bit immediately after updating HccaDoneHead, and does not update HccaDoneHead until this bit is cleared. Writing 1 to this bit clears the interrupt. 0: No WDH interrupt has occurred 1: WDH interrupt has occurred	R/W
2	SF	HccaFrameNumber update (StartOfFrame) This is an interrupt bit to indicate that HccaFrameNumber has been updated at the beginning of a frame. The host logic updates HccaFrameNumber and sets this bit when transmitting an SOF packet. Writing 1 to this bit clears the interrupt. 0: No SF interrupt has occurred 1: SF interrupt has occurred	R/W
3	RD	Resume detection (Resume Detected) This is an interrupt bit to indicate that Resume has been detected. This bit is set when assertion of the Resume signal by a device on the USB is detected. This bit is not set when software issues USB Resume. Writing 1 to this bit clears the interrupt. 0: No RD interrupt has occurred 1: RD interrupt has occurred	R/W
4	UE	USB non-related system error detection (Unrecoverable Error) This is an interrupt bit to indicate that a system error on the PCI bus that is not related to the USB has been detected. Writing 1 to this bit clears the interrupt. 0: No UE interrupt has occurred 1: UE interrupt has occurred	R/W
5	FNO	FrameNumber bit MSB change (Frame Number Overflow) This is an interrupt bit to indicate that the MSB of bits [15:0] (FrameNumber) in the HcFmNumber register has changed. This bit is set after HccaFrameNumber is updated in the frame where the MSB of the FrameNumber bits changes from 0 to 1 or from 1 to 0. Writing 1 to this bit clears the interrupt. 0: No FNO interrupt has occurred 1: FNO interrupt has occurred	R/W

Bit	Symbol	Function	R/W
6	RHSC	HcRhStatus/HcRhPortStatus register status (RootHubStatus Change) This is an interrupt bit to indicate that the state of the HcRhStatus register or the HcRhPortStatus register has changed. This bit is set when the HcRhStatus or HcRhPortStatus register state is changed by a hardware source. Writing 1 to this bit clears the interrupt. 0: No RHSC interrupt has occurred 1: RHSC interrupt has occurred	R/W
29:7	—	These bits are read as 0. The write value should be 0.	R/W
30	OC	Host logic control right change (OwnershipChange) This is an interrupt bit to indicate that bit 3 (OCR) in the HcCommandStatus register is set (the OCR bit is not used in this host logic). The value set for this bit has no meaning in this host logic.	R/W
31	—	This bit is read as 0. The write value should be 0.	R/W

The HCINTERRUPTSTATUS register indicates the states of events that trigger interrupts.

If an event occurs, the host logic sets 1 to the corresponding bit. When 1 is set to a bit in this register while 1 is set to the corresponding interrupt enable bit and the MasterInterruptEnable bit in the HcInterruptEnable register, an interrupt is generated.

29.2.1.5 HCINTERRUPTENABLE : HcInterruptEnable Register

Base address: USBHC = 0x8020_0000

Offset address: 0x010

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MIE	OCE	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	RHSCE	FNOE	UEE	RDE	SFE	WDHE	SOE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SOE	SO interrupt source enable (Scheduling OverrunEnable) This bit enables SO as an interrupt source. To set this bit, write 1 to it. To clear this bit, write 1 to the corresponding bit in the HcInterruptDisable register. 0: No monitoring and nothing is to be done 1: SO interrupt is enabled	R/W
1	WDHE	WDH interrupt source enable (WritebackDone HeadEnable) This bit enables WDH as an interrupt source. To set this bit, write 1 to it. To clear this bit, write 1 to the corresponding bit in the HcInterruptDisable register. 0: No monitoring and nothing is to be done. 1: WDH interrupt is enabled	R/W
2	SFE	SF interrupt source enable (StartOfFrame) This bit enables SF as an interrupt source. To set this bit, write 1 to it. To clear this bit, write 1 to the corresponding bit in the HcInterruptDisable register. 0: No monitoring and nothing is to be done 1: SF interrupt is enabled	R/W

Bit	Symbol	Function	R/W
3	RDE	RD interrupt source enable (Resume DetectedEnable) This bit enables RD as an interrupt source. To set this bit, write 1 to it. To clear this bit, write 1 to the corresponding bit in the HcInterruptDisable register. 0: No monitoring and nothing is to be done 1: RD interrupt is enabled	R/W
4	UEE	UE interrupt source enable (Unrecoverable ErrorEnable) This bit enables UE as an interrupt source. To set this bit, write 1 to it. To clear this bit, write 1 to the corresponding bit in the HcInterruptDisable register. 0: No monitoring and nothing is to be done 1: UE interrupt is enabled	R/W
5	FNOE	FNO interrupt source enable (FrameNumber OverflowEnable) This bit specifies FNO as an interrupt source. To set this bit, write 1 to it. To clear this bit, write 1 to the corresponding bit in the HcInterruptDisable register. 0: No monitoring and nothing is to be done 1: FNO interrupt is enabled	R/W
6	RHSCE	RHSC interrupt source enable (RootHubStatus ChangeEnable) This bit enables RHSC as an interrupt source. To set this bit, write 1 to it. To clear this bit, write 1 to the corresponding bit in the HcInterruptDisable register. 0: No monitoring and nothing is to be done 1: RHSC interrupt is enabled	R/W
29:7	—	These bits are read as 0. The write value should be 0.	R/W
30	OCE	OC interrupt source enable (OwnershipChangeEnable) This bit enables OC as an interrupt source. To set this bit, write 1 to it. To clear this bit, write 1 to the corresponding bit in the HcInterruptDisable register. 0: No monitoring and nothing is to be done 1: OC interrupt is enabled	R/W
31	MIE	Interrupt 8 source enable (MasterInterrupt Enable) This bit enables interrupt source settings in bits [30:0] in this register. When this bit is 0, all interrupts are masked. To clear this bit, write 1 to the corresponding bit in the HcInterruptDisable register. 0: No monitoring and nothing is to be done 1: The specified interrupt is enabled	R/W

The HCINTERRUPTENABLE register controls whether to generate an interrupt for each event. The position of each enable bit in this register is the same as the corresponding interrupt bit in the HcInterruptStatus register.

An interrupt is generated when all the following conditions are satisfied:

1. The corresponding bit in the HcInterruptStatus register is set to 1.
2. The corresponding bit in the HcInterruptEnable register is set to 1.
3. The MasterInterruptEnable (MIE) bit in the HcInterruptEnable register is set to 1.

Note: Writing 0 to any of the bits in this register does not change the read value.

29.2.1.6 HCINTERRUPTDISABLE : HcInterruptDisable Register

Base address: USBHC = 0x8020_0000

Offset address: 0x014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MID	OCD	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	RHSCD	FNOD	UED	RDD	SFD	WDHD	SOD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SOD	SO interrupt source disable (Scheduling Overrun Disable) This bit disables the interrupt source SO. Write 1 to this bit to clear the corresponding bit in the HcInterruptEnable register. To enable the interrupt, write 1 to the corresponding bit in the HcInterruptEnable register. 0: Nothing is to be done 1: The interrupt source SO is disabled	R/W
1	WDHD	WDH interrupt source disable (Writeback DoneHead Disable) This bit disables the interrupt source WDH. Write 1 to this bit to clear the corresponding bit in the HcInterruptEnable register. To enable the interrupt, write 1 to the corresponding bit in the HcInterruptEnable register. 0: Nothing is to be done 1: The interrupt source WDH is disabled	R/W
2	SFD	SF interrupt source disable (StartOfFrame Disable) This bit disables the interrupt source SF. Write 1 to this bit to clear the corresponding bit in the HcInterruptEnable register. To enable the interrupt, write 1 to the corresponding bit in the HcInterruptEnable register. 0: Nothing is to be done 1: The interrupt source SF is disabled	R/W
3	RDD	RD interrupt source disable (Resume Detected Disable) This bit disables the interrupt source RD. Write 1 to this bit to clear the corresponding bit in the HcInterruptEnable register. To enable the interrupt, write 1 to the corresponding bit in the HcInterruptEnable register. 0: Nothing is to be done 1: The interrupt source RD is disabled.	R/W
4	UED	UE interrupt source disable (Unrecoverable ErrorDisable) This bit disables the interrupt source UE. Write 1 to this bit to clear the corresponding bit in the HcInterruptEnable register. To enable the interrupt, write 1 to the corresponding bit in the HcInterruptEnable register. 0: Nothing is to be done 1: The interrupt source UE is disabled	R/W
5	FNOD	FNO interrupt source disable (FrameNumberOverflow Disable) This bit disables the interrupt source FNO. Write 1 to this bit to clear the corresponding bit in the HcInterruptEnable register. To enable the interrupt, write 1 to the corresponding bit in the HcInterruptEnable register. 0: Nothing is to be done 1: The interrupt source FNO is disabled	R/W
6	RHSCD	RHSC interrupt source disable (RootHub StatusChange Disable) This bit disables the interrupt source RHSC. Write 1 to this bit to clear the corresponding bit in the HcInterruptEnable register. To enable the interrupt, write 1 to the corresponding bit in the HcInterruptEnable register. 0: Nothing is to be done 1: The interrupt source RHSC is disabled	R/W
29:7	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
30	OCD	OC interrupt source disable (OwnershipChangeDisable) 0: Nothing is to be done 1: OC is deleted from the interrupt sources	R/W
31	MID	Interrupt 8 source disable (Master Interrupt Disable) This bit disables interrupt source settings in bits [30:0] in HcInterruptEnable. When this bit is 0, all interrupts are masked. Write 1 to this bit to clear the corresponding bit in the HcInterruptEnable register. To enable the interrupt, write 1 to the corresponding bit in the HcInterruptEnable register. 0: Nothing is to be done 1: The interrupt set is disabled	R/W

The position of each disable bit in this register is the same as the corresponding interrupt bit in the HcInterruptStatus register.

The HCINTERRUPTDISABLE register is connected to the HcInterruptEnable register. When 1 is set to a bit in this register, the corresponding bit in the HcInterruptEnable register is cleared.

The read values in this register are the current values in the HcInterruptEnable register.

Note: Writing 0 to any of the bits in this register does not change the value of the corresponding bit in the HcInterruptEnable register.

29.2.1.7 HCHCCA : HcHCCA Register

Base address: USBHC = 0x8020_0000

Offset address: 0x018

Bit position:	31																							7							0
Bit field:	RAMBA[23:0]																							—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
31:8	RAMBA[23:0]	RAM base address setting These bits specify the base address of the RAM allocated to the host controller communication area. Specify a value in these bits during initialization. The host logic requests a 256-byte area starting from the base address specified in these bits as HCCA.	R/W

The HCHCCA register indicates the physical address of the Host Controller Communication Area (referred to as HCCA hereafter).

Bits [7:0] of this register are fixed to 0x00 because the HCCA is allocated in units of 256 bytes.

29.2.1.8 HCPERIODCCURRENTIED : HcPeriodicCurrentED Register

Base address: USBHC = 0x8020_0000

Offset address: 0x01C

Bit position:	31																											4				0
Bit field:	PCED[27:0]																											—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
31:4	PCED[27:0]	ED physical address (PeriodicCurrentED) These bits indicate the physical address of the current ED being processed in the periodic list. The host logic updates these bits when completing the ED list processing.	R

The HCPERIODCCURRENTIED register indicates the physical address of the current isochronous or interrupt endpoint descriptor being processed.

29.2.1.9 HCCONTROLHEADED : HcControlHeadED Register

Base address: USBHC = 0x8020_0000

Offset address: 0x020

Bit position:	31																								4			0
Bit field:	CHED[27:0]																								—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
31:4	CHED[27:0]	Start ED physical address (ControlHeadED) These bits specify the physical address of the start ED in the control list. To execute control transfer, specify these bits before setting the CLE bit in the HcControl register.	R/W

The HCCONTROLHEADED register indicates the physical address of the start endpoint descriptor in the control list.

29.2.1.10 HCCONTROLCURRENTED : HcControlCurrentED Register

Base address: USBHC = 0x8020_0000

Offset address: 0x024

Bit position:	31																								4			0
Bit field:	CCED[27:0]																								—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
31:4	CCED[27:0]	ED physical address (ControlCurrentED) These bits indicate the physical address of the current ED being processed in the control list. When processing of the current ED is completed, the host logic moves on to the next ED. The host logic continues list processing until the end of the frame. When the host logic reaches the end of the list, it checks bit 1 (CLF) in the HcCommandStatus register. When the CLF bit is 1, the host logic copies the content of the CHED bits in the HcControlHeadED register to the CCED bits and clears the CLF bit. When the CLF bit is 0, the host logic does nothing. Software is permitted to update the HcControlCurrentED register only when bit 4 (CLE) in the HcControl register is cleared. When the CLE bit is 1, software is only allowed to read the value of the HcControlCurrentED register. To indicate the end of the list, 0x0 is initially set to this register.	R/W

The HCCONTROLCURRENTED register indicates the physical address of the current endpoint descriptor being processed in the control list.

29.2.1.11 HCBULKHEADED : HcBulkHeadED Register

Base address: USBHC = 0x8020_0000

Offset address: 0x028

Bit position:	31																									4			0
Bit field:	BHED[27:0]																									—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
31:4	BHED[27:0]	Start ED physical address (BulkHeadED) These bits specify the physical address of the start ED in the bulk list. To execute bulk transfer, specify these bits before setting bit 5 (BLE) in the HcControl register.	R/W

The HCBULKHEADED register indicates the physical address of the start endpoint descriptor in the bulk list.

29.2.1.12 HCBULKCURRENTED : HcBulkCurrentED Register

Base address: USBHC = 0x8020_0000

Offset address: 0x02C

Bit position:	31																									4			0
Bit field:	BCED[27:0]																									—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
31:4	BCED[27:0]	ED physical address (BulkCurrentED) These bits indicate the physical address of the current ED being processed in the bulk list. When processing of the current ED is completed, the host logic moves on to the next ED. The host logic continues list processing until the end of the frame. When the host logic reaches the end of the list, it checks bit 1 (CLF) in the HcCommandStatus register. When the CLF bit is 1, the host logic copies the content of the BHED bits in the HcBulkHeadED register to the BCED bits and clears the CLF bit. When the CLF bit is 0, the host logic does nothing. Software is permitted to update the HcBulkCurrentED register only when bit 4 (CLE) in the HcControl register is cleared. When the CLE bit is 1, software is only allowed to read the value of the HcBulkCurrentED register. To indicate the end of the list, 0 is initially set to this register.	R/W

The HCBULKCURRENTED register indicates the physical address of the current endpoint descriptor being processed in the bulk list.

29.2.1.13 HCDONEHEAD : HcDoneHead Register

Base address: USBHC = 0x8020_0000

Offset address: 0x030

Bit position:	31																									4			0
Bit field:	DH[27:0]																									—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0.	R
31:4	DH[27:0]	HcDoneHead physical address (DoneHead) These bits indicate the physical address of HcDoneHead in the host logic. This is the physical address of the last completed TD to be added to the Done queue. When processing of the TD is completed, the host logic writes the content of the HcDoneHead register to the NextTD bit of the TD. At that time, the host logic overwrites the content of the HcDoneHead register with the address of the TD. After the host logic writes the content of this register to the HCCA, it sets 0 to this register. In addition, the host logic sets bit 1 (WDH) in the HcInterruptStatus register to 1.	R

The HCDONEHEAD register indicates the physical address of the last completed transfer descriptor (referred to as TD hereafter) to be added to the Done queue.

In normal operation, software is not required to read this register because the content of this register is periodically written to the HCCA.

29.2.1.14 HCFMINTERVAL : HcFmInterval Register

Base address: USBHC = 0x8020_0000

Offset address: 0x034

Bit position:	31	30														16	15	14	13													0
Bit field:	FIT	FSMPS[14:0]															—	—	FI[13:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	1	1	0	1	1	1	1	1

Bit	Symbol	Function	R/W
13:0	FI[13:0]	Frame interval setting (FrameInterval) These bits specify the length (bit time) of a frame used for FS transfer. To conform to the USB standard (one frame = 1 ms), set these bits to 0x2EDF.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
30:16	FSMPS[14:0]	FS transfer packet maximum size setting (FSLagest DataPacket) These bits specify the maximum amount of data that can be transferred without generating a schedule overrun. This setting is compared with the current location in the frame to determine up to which data in the frame can be transferred. The maximum amount depends on the system bus performance, and it should be specified by the driver. The maximum value allowed for these bits is 0x2778. Do not set a value greater than 0x2778.	R/W
31	FIT	Frame synchronization (FrameInterval Toggle) This bit is used to match the frame settings between software and the host logic. When updating the FI bits, software should toggle this bit. When loading the FI bit value, the host logic copies the FIT value to bit 31 (FRT) in the HcFmRemaining register. Software can check if the new FI bit setting has been reflected by comparing the value specified in this bit when the FI bits are written to, and the value read from the FRT bit.	R/W

The HCFMINTERVAL register contains a 14-bit value that indicates the length of a frame (bit time) and a 15-bit value that indicates the maximum size of a packet that the host logic can send and receive at full speed without generating a schedule overrun.

By using the FI bit, adjustments are possible in units of frames.

29.2.1.15 HCFNREMAINING : HcFmRemaining Register

Base address: USBHC = 0x8020_0000

Offset address: 0x038

Bit position:	31													13														0
Bit field:	FRT	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FR[13:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	0	1	1	0	1	1	1	1	1

Bit	Symbol	Function	R/W
13:0	FR[13:0]	Down counter frame (FrameRemaining) These are the register bits of a 14-bit down counter which counts down the current remaining time value during transmission of a frame. The value of these bits is decremented over time. When the value reaches 0x0000, these bits are reloaded with the frame interval value by copying this value from bits [13:0] (FI) in the HcFmInterval register, after which counting down is restarted.	R
30:14	—	These bits are read as 0.	R
31	FRT	Frame synchronization (FrameRemainingToggle) This bit is used to match the frame settings between software and the host logic. When the FR bits reach 0x0000, the host logic copies the value of bits [13:0] (FI) in the HcFmInterval register to the FR bits and also copies the value of bit 31 (FIT) in the HcFmInterval register to this bit. Software can check if the FI bit setting has been copied to the FR bits by comparing the values of the FIT bit and this bit.	R

The HCFNREMAINING register is a 14-bit down counter which counts down the current remaining time during transfer of a frame.

29.2.1.16 HCFMNUMBER : HcFmNumber Register

Base address: USBHC = 0x8020_0000

Offset address: 0x03C

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FN[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	FN[15:0]	Elapsed frame number (FrameNumber) These bits indicate the elapsed frame count. The count in these bits is incremented when the HcFmRemaining register is reloaded. When the value reaches 0xFFFF, it is rolled over to 0x0000. When the host logic enters the USB operational state, the value in these bits is automatically incremented. After the host logic increments the value in these bits at the frame boundary and sends an SOF packet, the value in these bits is written to the HCCA. However, that happens before the host logic reads the first ED of that frame. After writing to the HCCA, the host logic sets bit 2 (SF) in the HcInterruptStatus register.	R
31:16	—	These bits are read as 0.	R

The HCFMNUMBER register indicates a 16-bit counter value.

29.2.1.17 HCPERIODSTART : HcPeriodicStart Register

Base address: USBHC = 0x8020_0000

Offset address: 0x040

Bit position:	31													13															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PS[13:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	PS[13:0]	Periodic list processing start time (PeriodicStart) These bits indicate the time when the host logic starts periodic list processing in a frame. Specify a value in these bits by software during initialization of the host logic. When the value of bits [13:0] (FR) in the HcFmRemaining register is greater than the value specified in these bits, non-periodic lists take priority over periodic lists. The OHCI standard recommends that this value be set to about 90% of the value of bits [13:0] (FI) in the HcFmInterval register. Therefore, the recommended value is 0x2A2F.	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The HCPERIODSTART register indicates the time (using 14 bits) when the host logic starts periodic list processing in a frame.

29.2.1.18 HCLSTHRESHOLD : HcLSThreshold Register

Base address: USBHC = 0x8020_0000

Offset address: 0x044

Bit position:	31																			11									0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LS[11:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0	1	0	0	0

Bit	Symbol	Function	R/W
11:0	LS[11:0]	Transferrable threshold (LSThreshold) These bits specify the threshold value to determine whether LS transfer can be done in the remaining frame time. When the FmRemaining value is greater than this threshold value, LS transfer can be started.	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The HCLSTHRESHOLD register contains the value of 12 bits that are used to determine whether the host logic can perform LS transfer in the remaining frame time.

29.2.1.19 HCRHDESCRIPTORA : HcRhDescriptorA Register

Base address: USBHC = 0x8020_0000

Offset address: 0x048

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	POTPGT[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	NOCP	OCPM	DT	NPS	PSM	NDP[7:0]							
Value after reset:	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
7:0	NDP[7:0]	Downstream port number (NumberDownstreamPorts) These bits specify the number of downstream ports supported by the root hub in the host logic.	R
8	PSM	Power switch control (PowerSwitchingMode) This bit specifies how to control the power switch to each port of the root hub. While bits [31:16] (PPCM) in the HcRhDescriptorB register are set, each port responds only to Set/ClearPortPower. While the bits are cleared, each port is controlled by Set/ClearGlobalPower. This bit setting is valid only when the NPS bit is cleared. 0: Power to all ports is controlled together 1: Power to each port is separately controlled	R/W
9	NPS	Power control (NoPower Switching) This bit specifies how to control the port power. 0: Port power can be switched on and off 1: Port power is always turned on while the host logic is operating	R/W
10	DT	Device type (DeviceType) This bit indicates that the root hub is not a compound device. As the root hub is not allowed to be a compound device, this bit is always read as 0.	R
11	OCPM	Overcurrent state reporting (OverCurrentProtection Mode) This bit specifies how to report the overcurrent state of the root hub. This bit should be set to the same mode as that specified in the PSM bit. This setting is valid only when the NOCP bit is cleared. 0: Overcurrent state of all ports is collectively reported 1: Overcurrent state of each port is separately reported	R/W
12	NOCP	Overcurrent function support (NoOver Current Protection) This bit specifies whether to support the overcurrent protection function for the root hub. 0: Overcurrent protection function is supported 1: Overcurrent protection function is not supported	R/W
23:13	—	These bits are read as 0. The write value should be 0.	R/W
31:24	POTPGT[7:0]	Wait time (PowerOnToPowerGood Time) These bits indicate the time that software should wait before accessing a root hub port after power supply to the port is started. The unit of time is 2 ms. Therefore, the wait time is obtained by POTPGT × 2 ms.	R/W

The HCRHDESCRIPTORA register is one of the registers that indicates the characteristics of the root hub.

29.2.1.20 HCRHDESCRIPTORB : HcRhDescriptorB Register

Base address: USBHC = 0x8020_0000

Offset address: 0x04C

Bit position:	31															16	15														0
Bit field:	PPCM[15:0]																DR[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DR[15:0]	Device Removable These bits indicate whether the device connected to each port in the root hub is removable. Each bit is dedicated to a single port. The USB host controller has port 1 only.	R/W
31:16	PPCM[15:0]	Port Power Control Mask These bits specify power control commands for each port. This setting is valid when bit 8 (PSM) in the HcRhDescriptorA register is set. Each bit is dedicated to a single port. The USB host controller has port 1 only.	R/W

Bit	Value	Description
DR[0]	—	Reserved
DR[1]	0	Device connected to port 1 is not removable.
	1	Device connected to port 1 is removable.
DR[15:2]	—	Reserved

Bit	Value	Description
PPCM[0]	—	Reserved
PPCM[1]	0	All ports are collectively controlled (Set/ClearGlobalPower).
	1	Port 1 is separately controlled (Set/ClearPortPower).
PPCM[15:2]	—	Reserved

The HCRHDESCRIPTORB register is one of the registers that indicate the characteristics of the root hub.

29.2.1.21 HCRHSTATUS : HcRhStatus Register

Base address: USBHC = 0x8020_0000

Offset address: 0x050

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CRWE	—	—	—	—	—	—	—	—	—	—	—	—	—	OCIC	LPSC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DRWE	—	—	—	—	—	—	—	—	—	—	—	—	—	OCI	LPS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LPS	Local power status (LocalPowerStatus) The meaning of this bit differs when it is read and written. When read: LocalPowerStatus This product does not support the local power status function. This bit is always read as 0. When written: ClearGlobalPower Setting this bit to 1 turns off the power to a port. The port to be turned off is determined according to the settings of bit 8 (PSM) in the HcRhDescriptorA register and bits [31:16] (PPCM) in the HcRhDescriptorB register.	R/W

Value	PSM	PPCM[1]	Description
0	—	—	No change
1	0	—	The PPS bit in the HcRhPortStatus1 register is cleared
	1	0	The PPS bit in the HcRhPortStatus1 register is cleared
		1	No change

Writing 0 has no effect.

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>1</td><td>OCI</td><td>Overcurrent indicator (OverCurrent Indicator)
This bit reports the overcurrent state in global overcurrent detection mode.
When overcurrent is reported separately for each port, this value is always 0.
0: Port is in normal state
1: Port is in overcurrent state</td><td>R</td></tr>
<tr><td>14:2</td><td>—</td><td>These bits are read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>15</td><td>DRWE</td><td>Device remote start enable (DeviceRemoteWakeupEnable)
When written: SetRemoteWakeupEnable
This bit is used to set the DRWE bit.
Setting this bit sets the DRWE bit.
Writing 0 has no effect.
When read: DeviceRemoteWakeupEnable
This bit specifies whether to include bit 16 (CSC) in the HcRhPortStatus1 register as a remote wakeup event.
When a connect status change event occurs while this bit is set, transition from the USB suspend state to the USB resume state occurs and the resume detect interrupt is generated.
0: ConnectStatusChange is not a RemoteWakeup source
1: ConnectStatusChange is a RemoteWakeup source</td><td>R/W</td></tr>
<tr><td>16</td><td>LPSC</td><td>Local power status change (LocalPowerStatusChange)
The meaning of this bit differs when it is read and written.
When read: LocalPowerStatusChange
This product does not support the local power status function and this bit is always read as 0.
When written: SetGlobalPower
Setting this bit to 1 turns on the power to a port.
The port to be turned on is determined according to the settings of bit 8 (PSM) in the HcRhDescriptorA register and bits [31:16] (PPCM) in the HcRhDescriptorB register.
<table>
<tr><th>Value</th><th>PSM</th><th>PPCM[1]</th><th>Description</th></tr>
<tr><td>0</td><td>—</td><td>—</td><td>No change</td></tr>
<tr><td rowspan="3">1</td><td>0</td><td>—</td><td>The PPS bit in the HcRhPortStatus1 register is set</td></tr>
<tr><td rowspan="2">1</td><td>0</td><td>The PPS bit in the HcRhPortStatus1 register is set</td></tr>
<tr><td>1</td><td>No change</td></tr>
</table>
Writing 0 has no effect.</td><td>R/W</td></tr>
<tr><td>17</td><td>OCIC</td><td>OCI bit change report (OverCurrent Indicate Change)
This bit notifies that the OCI bit value has changed.
It is set when the OCI bit changes.
Writing 1 to this bit while it is set to 1 clears it.
Writing 0 has no effect.
0: Nothing is to be done in case of an overcurrent
1: OverCurrent state has changed</td><td>R/W</td></tr>
<tr><td>30:18</td><td>—</td><td>These bits are read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>31</td><td>CRWE</td><td>DRWE bit clear (Clear Remote Wakeup Enable)
0: No effect
1: Clears the DRWE bit</td><td>W</td></tr>
</table>

29.2.1.22 HCRHPORTSTATUS1 : HcRhPortStatus1 Register

Base address: USBHC = 0x8020_0000

Offset address: 0x054

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	PRSC	OCIC	PSSC	PESC	CSC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	LSDA	PPS	—	—	—	PRS	POCI	PSS	PES	CCS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CCS	Connection status indication (CurrentConnectStatus) The meaning of this bit differs when it is read and written. When written: ClearPortEnable This bit is used to clear the PES bit. Writing 1 causes the port to enter the disabled state. Writing 0 has no effect. When read: CurrentConnectStatus This bit indicates the current connection state of the downstream port. 0: No device is connected 1: Device is connected	R/W
1	PES	Port enable status (PortEnableStatus) The meaning of this bit differs when it is read and written. When written: SetPortEnable This bit is used to set the PES bit. Writing 1 causes the port to enter the enabled state. Writing 0 has no effect. If this bit is written to while the CCS bit is cleared, this bit is not set but the CSC bit is set to notify the driver of an attempt to enable a disconnected port. When read: PortEnableStatus This bit indicates whether the port is enabled or disabled. The host logic automatically clears this bit on detecting the overcurrent state, disconnection, power-off, or a babble error. At this time, the PESC bit is set. This bit cannot be set while the CCS bit is cleared (no device is connected). The host logic sets this bit when a port reset is completed and the PRSC bit is set, or when a port enters the suspend state and the PRSC bit is set. 0: Port is disabled 1: Port is enabled	R/W
2	PSS	Suspend/Resume status (PortSuspendStatus) The meaning of this bit differs when it is read and written. When written: SetPortSuspend This bit is used to clear the PES bit. Writing 1 makes the port enter the disabled state. Writing 0 has no effect. If this bit is written to while the CCS bit is cleared, this bit is not set but the CSC bit is set to notify the driver of an attempt to suspend a disconnected port. When read: PortSuspendStatus This bit indicates that the port is in the suspend state or the resume sequence. This bit is not set while the CCS bit is cleared (no device is connected). This bit is cleared with the following timing: ● When the resume sequence is completed and the PSSC bit is set ● When a port reset is completed and the PRSC bit is set ● In the USB RESUME state 0: Port is in normal transfer state 1: Port is in suspend state	R/W

Bit	Symbol	Function	R/W
3	POCI	Downstream port overcurrent detection (PortOverCurrentIndicator) The meaning of this bit differs when it is read and written. When written: ClearSuspendStatus This bit terminates the suspend state and starts the resume sequence. Writing 1 starts the resume sequence. Writing 0 has no effect. The resume sequence starts only while the PSS bit is set. When read: PortOverCurrentIndicator This bit indicates that a downstream port has entered the overcurrent state. It is valid only while the overcurrent state is reported separately for each port (OCPM = 1 in the HcRhDescriptorA register). While the overcurrent state in all ports is collectively reported, this bit is 0b. 0: Port is in normal state 1: Port is in overcurrent state	R
4	PRS	Port reset status (PortResetStatus) The meaning of this bit differs when it is read and written. When written: SetPortReset This bit applies a port reset to a downstream port. Writing 1 to this bit starts a 10-ms port reset. Writing 0 has no effect. If this bit is written to while the CCS bit is cleared, this bit is not set but the CSC bit is set to notify software of an attempt to reset a port where no device is connected. When read: PortResetStatus This bit indicates the reset state of the port. It is cleared at the same time as when the PRSC bit is set after a 10-ms port reset is completed. This bit cannot be set while the CCS bit is cleared (no device is connected). 0: Port reset is not in progress 1: Port reset is in progress	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	PPS	Power status (PortPowerStatus) The meaning of this bit differs when it is read and written. When written: SetPortPower This bit turns the port power on when power is separately controlled for each port. Writing 1 turns the port on. Writing 0 has no effect. When read: PortPowerStatus This bit indicates the power state of the port. It is cleared when an overcurrent is detected. 0: Port power is off 1: Port power is on	R/W
9	LSDA	Device speed (LowSpeedDeviceAttached) The meaning of this bit differs when it is read and written. When written: ClearPortPower This bit turns off the port power when power is separately controlled for each port. Writing 1 turns the port off. Writing 0 has no effect. When read: LowSpeedDeviceAttached This bit indicates the speed of the device connected to the port. This status bit is valid only while the CCS bit is set. 0: FS device is connected 1: LS device is connected	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	CSC	CCS bit status (ConnectStatus Change) This bit indicates that the CCS bit value has changed. The host logic sets this bit when the CCS bit changes due to device connection or disconnection. In addition, when a request for port reset, port suspension, or port enable is issued in the disconnection state, this bit is set to make software re-evaluate the device connection state. It is cleared when 1 is written by software. 0: Nothing is to be done regarding CCS 1: CCS has changed	R/W
17	PESC	PES bit status (PortEnable StatusChange) This bit indicates that the PES bit value has changed. It is set when the port state has changed due to a hardware event, such as the overcurrent state, disconnection, power-off, or a babble error. It is cleared when 1 is written by software. 0: Nothing is to be done regarding PES 1: PES state has changed	R/W

Bit	Symbol	Function	R/W
18	PSSC	RESUME sequence complete (PortSuspend StatusChange) This bit indicates that the RESUME sequence has been completed. It is set when all RESUME processing by hardware has been completed. It is cleared when 1 is written by software. In addition, it is cleared when the PRSC bit is set. 0: RESUME has not been completed 1: RESUME has been completed	R/W
19	OCIC	Overcurrent state detection (OverCurrent IndicateChange) This bit is set when the overcurrent state of the port has been detected. It is cleared when 1 is written by software. It is valid only while the overcurrent state is reported separately for each port (OCPM = 1 in the HcRhDescriptorA register). 0: Nothing is to be done regarding overcurrent state 1: Overcurrent state has changed	R/W
20	PRSC	Port reset complete (PortReset StatusChange) This bit indicates that a port reset has been completed. It is set when the host logic has completed a 10-ms hardware reset. It is cleared when 1 is written by software. 0: Port reset has not been completed, or nothing is to be done regarding the PRS bit 1: Port reset has been completed	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The HCRHPORTSTATUS1 register is used to control and report port events.

29.2.2 EHCI Operational Registers

Access the EHCI operational registers after starting up the PHY internal PLL. For details, see Figure 29.7.

29.2.2.1 CAPL_VERSION : Capability Registers Length and EHCI Version Number Register

Base address: USBHC = 0x8020_0000

Offset address: 0x100

Bit position:	31															16							7							0
Bit field:	HCIVN[15:0]																—	—	—	—	—	—	—	—	CRL[7:0]						
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CRL[7:0]	Capability Registers Length These bits indicate the start address of the host logic operational registers. As the host logic operational registers are allocated from address 0x20, this value is set to 0x20.	R
15:8	—	These bits are read as 0.	R
31:16	HCIVN[15:0]	EHCI Version Number These bits indicate the version of the EHCI specifications supported by the host logic. As the host logic conforms to the EHCI Rev. 1.1, this value is set to 0x0110.	R

29.2.2.2 HCSPARAMS : Structural Parameters Register

Base address: USBHC = 0x8020_0000

Offset address: 0x104

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DBGPTNUM[3:0]				—	—	—	P_INDICATOR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	N_CC[3:0]				N_PCC[3:0]				PTRR	—	—	PPC	N_PORTS[3:0]			
Value after reset:	0	0	0	1	0	0	0	1	1	0	0	1	0	0	0	1

Bit	Symbol	Function	R/W
3:0	N_PORTS[3:0]	Number of downstream ports (Number of Ports) These bits indicate the number of physical downstream ports used by the host logic. These bits reflect the setting of bits [1:0] (Port_no) in the PCI Configuration EXT1 register. As the USB host controller has only one port, this value is set to 1. 0x1: 1 port Others: Reserved	R
4	PPC	Port power control (Port Power Control) This bit indicates how to control the port power in the host logic. 0: Port power is always on 1: Port power is controlled using bit 12 (PP) in the PORTSC1 register	R
6:5	—	These bits are read as 0.	R
7	PTRR	Port routing rules This bit indicates how ports are mapped in the OHCI host logic. As the value of the HCSP_PORTROUTE register shows the mapping in the host logic, this value is set to 1.	R
11:8	N_PCC[3:0]	Number of ports (Number of Ports per Companion Controller) These bits indicate the number of the ports supported by one OHCI host logic unit. These bits reflect the setting of bits [1:0] (Port_no) in the PCI Configuration EXT1 register.	R
15:12	N_CC[3:0]	Number of OHCI host logic (Number of Companion Controller) These bits indicate the number of OHCI host logic units related to the EHCI host logic. As the host logic has one OHCI host logic unit, this value is set to 1.	R
16	P_INDICATOR	Port indicator control support This bit indicates whether the host logic supports the port indicator control function. As the host logic does not support the port indicator control function, this value is set to 0.	R
19:17	—	These bits are read as 0.	R
23:20	DBGPTNUM[3:0]	Debug port number These bits indicate that the host logic ports are debug ports. As the host logic has no debug ports, this value is set to 0.	R
31:24	—	These bits are read as 0.	R

29.2.2.3 HCCPARAMS : Capability Parameters Register

Base address: USBHC = 0x8020_0000

Offset address: 0x108

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PL32	PCEC	LPMC	HP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EECP[7:0]								IST[3:0]				—	ASPC	PFLF	AC64
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

Bit	Symbol	Function	R/W
0	AC64	Memory pointer selection This bit indicates whether data structures use 32-bit address memory pointers or 64-bit address memory pointers. As the host logic supports data structures using 32-bit address memory pointers, this value is set to 0. Address memory pointers of 64-bits are not supported.	R
1	PFLF	Programming frame list flag This bit indicates the setting regarding the frame list size that software can use. This value is set to 1 in the host logic. When this bit is set to 1, the available frame list size can be specified through bits [3:2] (Frame List Size) in the USBCMD register. A frame list of size smaller than 4 KB can be specified.	R
2	ASPC	Asynchronous schedule park support capability This bit indicates whether to support the park mode for the high-speed QH (queue head) in the asynchronous schedule. As the host logic supports this function, this value is set to 1.	R
3	—	This bit is read as 0.	R
7:4	IST[3:0]	Isochronous data structure threshold As the host logic in this product does not support caching of the isochronous data structure in the entire frame, these bits are set to 0.	R
15:8	EECP[7:0]	Offset address (EHCI Extend Capabilities Pointer) These bits indicate the offset address for the EHCI extend capabilities registers. As the host logic does not support the EHCI extend capabilities registers, these bits are set to 0.	R
16	HP	Hardware prefetch capability As the host logic does not support the hardware prefetch capability defined in EHCI V1.1, this bit is set to 0.	R
17	LPMC	Link power management capability As the host logic supports Link Power Management (LPM) defined in EHCI V1.1, this bit is set to 1. When LPMC = 1, it means that the host logic supports the LPM L1 state and controls LPM by using bit 9 (LPMCTL), bits [24:23] (SUSPSTS[1:0]), and bits [31:25] (DVADDR[6:0]) in the PORTSC1 register. When LPMC = 0, the applicable bits in the PORTSC1 register mentioned above are treated as reserved bits.	R
18	PCEC	Per-port change event capability As the host logic supports the per-port change event capability defined in EHCI V1.1, this bit is set to 1. When this bit is 1, it means that the host logic supports the per-port change event capability. It also means that this bit is associated with bit 15 (per-port change event enable) in the USBCMD register, bit 2 (Port Change Detect) in the USBSTS register, and bit 2 (Port Change Interrupt Enable) in the USBINTR register. When PCEC = 0, the applicable bits in the registers mentioned above are treated as reserved bits.	R

Bit	Symbol	Function	R/W
19	PL32	32-frame periodic list capability As the host logic supports 32-Frame Periodic List defined in EHCI V1.1, this bit is set to 1. When this bit is 1, it means that the host logic supports 32-Frame Periodic List when 11b is set in bits [3:2] (frame list size) in the USBCMD register.	R
31:20	—	These bits are read as 0.	R

29.2.2.4 HCSP_PORTROUTE : Companion Port Route Description Register

Base address: USBHC = 0x8020_0000

Offset address: 0x10C

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	OHCI Host Port Indication These bits indicate the ports that the OHCI host logic controls. As the host logic has one OHCI host logic unit, these bits are set to 0x00000000.	R

The HCSP_PORTROUTE register indicates how ports are mapped in the OHCI controller. The register is enabled only when bit 7 (port routing rules) in the HCSPARAMS register is 1.

29.2.2.5 USBCMD : USB Command Register

Base address: USBHC = 0x8020_0000

Offset address: 0x120

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	HIRD[3:0]				ITC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PPCEE	—	—	—	ASPME	—	ASPMC[1:0]		LHCR	IAAD	ASYNSE	PSE	FLS[1:0]		HCRESET	RS
Value after reset:	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RS	EHCI host logic run/stop (Run/Stop) This bit activates or stops the EHCI host logic. When this bit is set to 1, the host logic starts operation. As long as this value is 1, the host logic continues operation. Note that this bit should be set to 1 only while the host logic is in the halt state. Bit 12 (HCHalted) in the USBSTS register indicates that the host logic has completed transactions and entered the halt state. 0: Stopped (the host logic has completed transactions and halted) 1: Activated (the host logic executes the schedule)	R/W

Bit	Symbol	Function	R/W
1	HCRESET	Host logic initialization (Host Controller Reset) This bit initializes the host logic. When this bit is set to 1, the host logic initializes the internal pipelines and state machine. Communication through the USB stops immediately. A USB reset is not issued to downstream ports at this time. This reset does not initialize the PCI configuration registers, but it initializes the EHCI operational registers and the port ownership is returned to OHCI. This bit is automatically cleared to 0 by the host logic when the reset is completed. The reset cannot be aborted by software by writing 0 to this bit. This bit should be set only while bit 12 (HCHalted) in the USBSTS register is 1.	R/W
3:2	FLS[1:0]	Frame list size These bits specify the frame list size. The setting in these bits determines the size of the Frame List Current index in the FRINDEX register. 0 0: 1024 elements (4096 bytes) 0 1: 512 elements (2048 bytes) 1 0: 256 elements (1024 bytes) 1 1: 32 elements (128 bytes)	R/W
4	PSE	Periodic schedule enable This bit specifies whether to advance or skip periodic list processing. 0: Periodic list processing is not executed (skipped) 1: Periodic list processing is executed using the PERIODICLISTBASE register	R/W
5	ASYNSE	Asynchronous schedule enable This bit specifies whether to advance or skip asynchronous list processing. 0: Asynchronous list processing is not executed (skipped) 1: Asynchronous list processing is executed using the ASYNCLISTADDR register	R/W
6	IAAD	Interrupt on Async Advance Doorbell This bit is used as a doorbell by software. Software sets this bit to 1 to generate an interrupt when processing proceeds with the next QH (queue head). While bit 5 (Interrupt on Async Advance Enable) in the USBINTR register is set to 1, an interrupt is generated with the next interrupt timing after 1 is written to this bit. If this bit is set while bit 5 (Interrupt on Async Advance Enable) in the USBINTR register is 0, correct operation is not guaranteed. This bit is cleared by the host logic. On completing the processing of a QH normally, the host logic clears this bit to 0 and sets bit 5 (Interrupt on Async Advance) in the USBSTS register to 1.	R/W
7	LHCR	Light host controller reset execution status This bit indicates the state of Light Host Controller Reset execution. As the host logic does not support Light Host Controller Reset, this value is fixed to 0.	R
9:8	ASPMC[1:0]	Asynchronous schedule park mode count These bits specify the number of transactions that can be executed in succession from one QH (queue head). A value from 0x1 to 0x3 can be specified. This setting is valid when bit 11 (Asynchronous Schedule Park Mode Enable) is 1.	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	ASPME	Asynchronous schedule park mode enable This bit enables or disables the park mode. 0: Disabled 1: Enabled	R/W
14:12	—	These bits are read as 0. The write value should be 0.	R/W
15	PPCEE	Per-port change event enable This bit is used by software to enable the per-port change event capability defined by bit 2 (Port Change Detect) in the USBSTS register and bit 2 (Port Change Interrupt Enable) bit in the USBINTR register.	R/W

Bit	Symbol	Function	R/W
23:16	ITC[7:0]	Host logic interrupt generation maximum rate (Interrupt Threshold Control) These bits indicate the maximum rate at which the host logic generates an interrupt. Correct operation is not guaranteed if a value not shown below is written to these bits. 0x00: Reserved 0x01: 1 micro-frame 0x02: 2 micro-frames 0x04: 4 micro-frames 0x08: 8 micro-frames (1 ms) 0x10: 16 micro-frames (2 ms) 0x20: 32 micro-frames (4 ms) 0x40: 64 micro-frames (8 ms)	R/W
27:24	HIRD[3:0]	Host-Initiated Resume Duration (Minimum K-state drive time) These bits are used by software. The bits indicate the minimum amount of time the host controller drives the K-state during a resume from an LPM state (L1). The value of these bits is sent to the connected devices with the LPM capability by the HIRD field contained in the bmAttributes field in an LPM token. Encoding for these bits is defined in the bits named HIRD in an LPM token. Specifically, 0x0 equals 50 µs and each additional increment adds 75 µs. For example, 0x1 equals 125 µs and 0x4 equals 1175 µs.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The USBCMD register indicates the commands that are executed by the host logic.

29.2.2.6 USBSTS : USB Status Register

Base address: USBHC = 0x8020_0000

Offset address: 0x124

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PTCGDETC[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ASS	PSCHSTS	RECLAM	EHCSTS	—	—	—	—	—	—	IAAIS	HSYSE	FLROV	PTCGDET	USBERRINT	USBINT
Value after reset:	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	USBINT	USB transfer complete (USB Interrupt) This bit indicates that USB transfer has been completed. The host logic sets this bit to 1 when either of the following conditions is satisfied: ● USB transfer is completed ● A short packet is received Even if USB transfer has ended with an error, this bit is set to 1 when the IOC (Interrupt On Complete) bit in the TD is 1. Writing 1 can clear this bit. Writing 0 has no effect. 0: USB transfer has not been completed 1: USB transfer has been completed	R/W
1	USBERRINT	USB transaction status (USB Error Interrupt) This bit indicates that a USB transaction has ended with an error. The host logic sets this bit to 1 when a USB transaction has ended with an error. If 1 is set to the IOC bit in the TD that caused an error interrupt, 1 is set to both the USBERRINT bit and the USBINT bit. Writing 1 can clear this bit. Writing 0 has no effect. 0: USB transaction has been completed normally 1: USB transaction has ended with an error	R/W

Bit	Symbol	Function	R/W
2	PTCGDET	Port state change detection This bit indicates that the port state has changed. The host logic sets this bit to 1 when any of the following conditions is satisfied in a port for which bit 13 (Port Owner) in the PORTSC[n] register is set to 0. [1 setting condition] ● Bit 1 (Connect Status Change) in the PORTSC[n] register has changed from 0 to 1 (device connection or disconnection has been detected) ● Bit 3 (Port Enable/Disable Change) in the PORTSC[n] register has changed from 0 to 1 (the port enabled state has changed) ● Bit 5 (Over-current Change) in the PORTSC[n] register has changed from 0 to 1 (the overcurrent state has been detected) ● Bit 6 (Force Port Resume) in the PORTSC[n] register has changed from 0 to 1 (indicating detection of a J-to-K transition on a port while suspended) Writing 1 by software can clear this bit. Writing 0 has no effect.	R/W
3	FLROV	Frame list rollover The host logic sets this bit to 1 when the value of the Frame Index bits in the FRINDEX register returns from the maximum value to 0x000 (rollover). The maximum value at which rollover occurs depends on bits [3:2] (Frame List Size) in the USBCMD register. Writing 1 can clear this bit. Writing 0 has no effect. 0: Frame list has not returned to 0x000 1: Frame list has returned to 0x000	R/W
4	HSYSE	Host system error This bit is set to 1 when a serious error occurs in the host logic. Such an error includes a parity error in the PCI system. If this error occurs, the host logic clears bit 0 (RS) in the USBCMD register to 0 to stop execution of the remaining TDs. Writing 1 can clear this bit. Writing 0 has no effect. 0: No system error has occurred 1: System error has occurred	R/W
5	IAAIS	Async advance interrupt status This bit indicates the Async Advance interrupt state. On fetching QH, the host logic checks bit 6 (Interrupt on Async Advance Doorbell (IAAD)) in the USBCMD register. When IAAD = 1, the host logic clears the IAAD bit and sets this bit after completing QH processing normally. When bit 5 (Interrupt on Async Advance Enable) in the USBINTR register is 1, an interrupt is generated from this source with the next interrupt timing after this bit is set to 1. Writing 1 can clear this bit. Writing 0 has no effect. 0: No Async Advance interrupt has occurred 1: Async Advance interrupt state has been detected	R/W
11:6	—	These bits are read as 0. The write value should be 0.	R/W
12	EHCSTS	EHCI host logic status (HCHalted) This bit is set to 0 when bit 0 (RS) in the USBCMD register is 1. When the RS bit is cleared to 0 by the host logic or software, the EHCI host logic stops execution and this bit is set to 1 by the host logic. 0: EHCI host logic is in execution 1: EHCI host logic is stopped	R
13	RECLAM	Empty asynchronous schedule detection (Reclamation) This bit is used to detect an empty asynchronous schedule. The host logic clears this bit to 0 after a reset or when it fetches QH with H = 1. When the host logic executes an asynchronous transaction or has detected a start event, it sets this bit to 1. When the host logic fetches QH with H = 1 while this bit is 0, the Async Sched Sleeping mode is entered.	R
14	PSCHSTS	Periodic schedule status This bit indicates the current state of the periodic schedule. The periodic schedule is enabled (1) or disabled (0) when the values of this bit and bit 4 (Periodic Schedule Enable) in the USBCMD register are the same. 0: Periodic schedule is disabled 1: Periodic schedule is enabled	R

Bit	Symbol	Function	R/W
15	ASS	Asynchronous schedule status This bit indicates the current state of the asynchronous schedule. The asynchronous schedule is enabled (1) or disabled (0) when the values of this bit and bit 5 (Asynchronous schedule enable) in the USBCMD register are the same. 0: Asynchronous schedule is disabled 1: Asynchronous schedule is enabled	R
31:16	PTCGDETC[15:0]	Port-n Change Detect Each of these bits is the Port Change Detect bit for a specific port. Bit 16 is the port change detect bit for port 1, bit 17 is the port change detect bit for port 2, and so on. For example, when bit 16 is set to 1, it means the status change of port 1 was detected. These bits are used by software only when bit 15 (Per-Port Change Events Enable) in the USBCMD register is set to 1.	R/W

The USBSTS register indicates interrupt sources and various states of the host logic.

The result of a transaction on the USB bus is not indicated in this register.

29.2.2.7 USBINTR : USB Interrupt Enable Register

Base address: USBHC = 0x8020_0000

Offset address: 0x128

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PCGIE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	INTAADVE	HSEE	FMLSTROE	PTCGIE	USBEIE	USBIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	USBIE	USB interrupt enable This bit enables or disables the setting in bit 0 (USBINT) in the USBSTS register. When USBSTS.USBINT = 1 while this bit is set to 1, the host logic generates an interrupt at the next interrupt threshold. The interrupt is acknowledged by software when the USBINT bit is cleared. 0: Disabled 1: Enabled (interrupt indicated by the USBINT bit)	R/W
1	USBEIE	USB error interrupt enable This bit enables or disables the setting in bit 1 (USBERRINT) in the USBSTS register. When USBSTS.USBERRINT = 1 while this bit is set to 1, the host logic generates an interrupt at the next interrupt threshold. The interrupt is acknowledged by software when the USBERRINT bit is cleared. 0: Disabled 1: Enabled (interrupt indicated by the USBERRINT bit)	R/W
2	PTCGIE	Port change interrupt enable This bit enables or disables the setting in bit 2 (Port Change Detect) in the USBSTS register. When the Port Change Detect bit is set to 1 while this bit is set to 1, the host logic generates an interrupt at the next interrupt threshold. The interrupt is acknowledged by software when the Port Change Detect bit is cleared. 0: Disabled 1: Enabled (interrupt indicated by the Port Change Detect bit)	R/W

Bit	Symbol	Function	R/W
3	FMLSTROE	Frame list rollover enable This bit enables or disables the setting in bit 3 (Frame List Rollover) in the USBSTS register. When the Frame List Rollover bit is set to 1 while this bit is set to 1, the host logic generates an interrupt at the next interrupt threshold. The interrupt is acknowledged by software when the Frame List Rollover bit is cleared. 0: Disabled 1: Enabled (interrupt indicated by the Frame List Rollover bit)	R/W
4	HSEE	Host system error enable This bit enables or disables the setting in bit 4 (Host System Error) in the USBSTS register. When the Host System Error bit is set to 1 while this bit is set to 1, the host logic generates an interrupt at the next interrupt threshold. The interrupt is acknowledged by software when the Host System Error bit is cleared. 0: Disabled 1: Enabled (interrupt indicated by the Host System Error bit)	R/W
5	INTAADVE	Interrupt on async advance enable This bit enables or disables the setting in bit 5 (Interrupt on Async Advance) in the USBSTS register. When the Interrupt on Async Advance bit is set to 1 while this bit is set to 1, the host logic generates an interrupt at the next interrupt threshold. The interrupt is acknowledged by software when the Interrupt on Async Advance bit is cleared. 0: Disabled 1: Enabled (interrupt indicated by the Interrupt on Async Advance bit)	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W
31:16	PCGIE[15:0]	Port-n Change Interrupt Enable Each of these bits is the Port Change Interrupt Enable bit for a specific port. Bit 16 is the Port Change Interrupt Enable bit for port 1, bit 17 is the Port Change Interrupt Enable bit for port 2, and so on. When one of these bits is set to 1 and the corresponding bit (one of PTCGDETC[15:0], Port-n Change Detect) in the USBSTS register is set to 1, the host logic generates an interrupt. Only bit 16 is enabled because the USB host controller only supports port 1. The interrupt is acknowledged by software when the Port-n Change Detect bit is cleared.	R/W

The USBINTR register enables or disables reporting of interrupts.

When a bit in this register is set to 1 and the corresponding interrupt bit is set to 1, an interrupt is generated.

The interrupt sources that are disabled in this register can be detected by allowing software to poll for events stored in the USBSTS register.

29.2.2.8 FRINDEX : USB Frame Index Register

Base address: USBHC = 0x8020_0000

Offset address: 0x12C

Bit position:	31														13																0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FRAMEINDEX[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>13:0</td><td>FRAMEINDEX[13:0]</td><td>Frame index
These bits are used by the host logic to add an index to the periodic frame list.
This value is incremented at the end of each micro-frame.
Bits [N:3] are used as the Frame List Current index. This means that the current frame list is accessed 8 times before proceeding with the next index number. The value of N is determined as follows according to the setting of bits [3:2] (Frame List Size) in the USBCMD register.
<table>
<tr><th>Frame List Size</th><th>Number Elements</th><th>N</th></tr>
<tr><td>00b</td><td>(1024)</td><td>12</td></tr>
<tr><td>01b</td><td>(512)</td><td>11</td></tr>
<tr><td>10b</td><td>(256)</td><td>10</td></tr>
<tr><td>11b</td><td>(32)</td><td>12</td></tr>
</table>
This register should be accessed only while the host logic is stopped (bit 12 (HCHalted) = 1 in the USBSTS register).
This setting is reflected in the SOF frame number for the SOF token.</td><td>R/W</td></tr>
<tr><td>31:14</td><td>—</td><td>These bits are read as 0. The write value should be 0.</td><td>R/W</td></tr>
</table>

The FRINDEX register is used by the host logic to add an index to the periodic frame list.

29.2.2.9 CTRLDSSEGMENT : Control Data Structure Segment Register

Base address: USBHC = 0x8020_0000

Offset address: 0x130

Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	The host logic does not support 64-bit addressing and does not use this register. Therefore, do not access this register.	R

29.2.2.10 PERIODICLISTBASE : Periodic Frame List Base Address Register

Base address: USBHC = 0x8020_0000

Offset address: 0x134

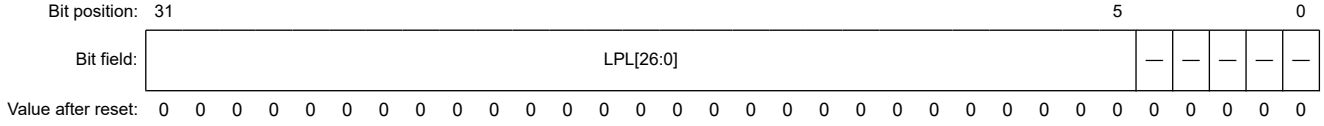
Bit position:	31																				12									0
Bit field:	PFLSA[19:0]																				—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	—	These bits are read as 0. The write value should be 0.	R/W
31:12	PFLSA[19:0]	Periodic frame list start address These bits indicate the start address of the periodic frame list stored in the system memory. Software should specify the start address in this register before the host logic starts list processing. The host logic uses the values in these bits and bits [13:0] (Frame Index) in the FRINDEX register to determine the frame list to be processed. The address of the periodic frame list should be aligned with a 4-KB boundary. Correct operation cannot be guaranteed if the value in these bits is modified during operation.	R/W

29.2.2.11 ASYNCLISTADDR : Next Asynchronous List Address Register

Base address: USBHC = 0x8020_0000

Offset address: 0x138

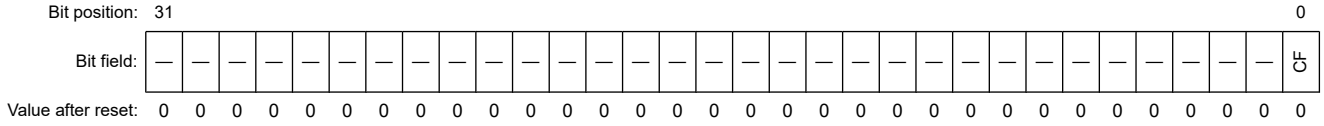


Bit	Symbol	Function	R/W
4:0	—	These bits are read as 0. The write value should be 0.	R/W
31:5	LPL[26:0]	Asynchronous Queue Head link pointer address (Link Pointer Low) These bits indicate the address of the next asynchronous queue head in the system memory to be processed. The address of the asynchronous queue head should be aligned with a 32-byte boundary.	R/W

29.2.2.12 CONFIGFLAG : Configure Flag Register

Base address: USBHC = 0x8020_0000

Offset address: 0x160

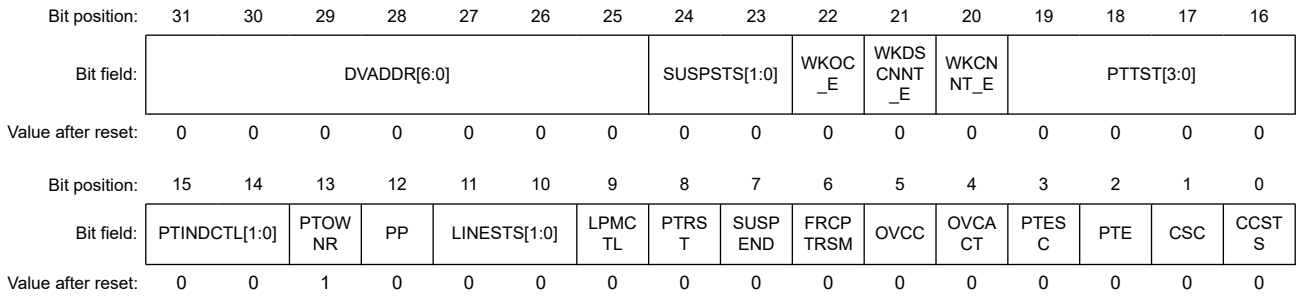


Bit	Symbol	Function	R/W
0	CF	Port routing control circuit configuration flag (Configure Flag) This bit controls the default setting of the port routing control circuit regarding whether the ports are routed to OHCI or EHCI. Software should set this bit to 1 at the end of host logic configuration process. 0: Port routing control circuit routes each port to the OHCI host logic by default 1: Port routing control circuit routes each port to the EHCI host logic by default	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

29.2.2.13 PORTSC1 : Port 1 Status and Control Register

Base address: USBHC = 0x8020_0000

Offset address: 0x164



Bit	Symbol	Function	R/W
0	CCSTS	Port connection status This bit indicates the connection state of the port. On detecting device connection, the host logic sets this bit to 1. The host logic also sets this bit to 1 when PTTST[3:0] = 0101b (Test FORCE_ENABLE) even if no device is connected. On detecting device disconnection, the host logic clears this bit to 0. When PP = 0 or PTOWNR = 0, this bit becomes 0. 0: No device is connected to the port 1: Device is connected to the port	R
1	CSC	Connect status change This bit indicates that the value of bit 0 (Current Connect Status) has changed. Writing 1 can clear this bit. Writing 0 has no effect. When PP = 0, this bit becomes 0. 0: Value of bit 0 (Current Connect Status) has not changed 1: Value of bit 0 (Current Connect Status) has changed	R/W
2	PTE	Port enable/disable status This bit indicates the enabled/disabled state of the port. After resetting a port and detecting that the connected device is an HS device, the host logic enables the port and sets this bit to 1. Software cannot set this bit to 1. On detecting device disconnection or an error, the host logic disables the port and clears this bit to 0. Software can write 0 to disable the port. In this case, however, the bit value changes only after the port state actually changes. While the port is disabled, data transfer to downstream ports is blocked except for a port reset. When PP = 0, this bit becomes 0. Note that when PTTST[3:0] = 0101b (Test FORCE_ENABLE), the port is enabled regardless of the port state and this bit is set to 1. 0: Port is disabled 1: Port is enabled	R/W
3	PTESC	Port enable/disable status change This bit indicates that the enabled/disabled state of the port has changed. On detecting a frame babble error, the host logic disables the port and sets this bit to 1. Writing 1 can clear this bit. Writing 0 has no effect. When PP = 0, this bit becomes 0. 0: Frame babble error has not occurred 1: Frame babble error has been detected	R/W
4	OVCACT	Port overcurrent status This bit indicates the overcurrent state of the port. On detecting an overcurrent, the host logic clears bit 12 (PP) and its related bits and sets this bit to 1. This bit is automatically cleared from 1 to 0 when the overcurrent state is resolved. 0: Port is not in overcurrent state 1: Port is in overcurrent state	R
5	OVCC	Over-current Change This bit indicates that the value of bit 4 (OVCACT) has changed. Writing 1 can clear this bit. Writing 0 has no effect. 0: Value of bit 4 (OVCACT) has not changed 1: Value of bit 4 (OVCACT) has changed	R/W

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>6</td><td>FRCPTRSM</td><td>Force Port Resume (Port resume detection flag)
This bit indicates detection of the resume state at the port.
When a transition from the J to the K state (RemoteWakeup) is detected while the port is suspended, the host logic sets this bit and bit 2 (Port Change Detect) or bits [31:16] (Port-n Change Detect [15:0]) in the USBSTS register to 1. If software writes 1 to the Force Port Resume bit, the host logic does not set the Port Change Detect bit or the Port-n Change Detect bits to 1.
The resume signal (FS K State) is driven on the USB port while this bit is 1. Clear this bit to 0 after an appropriate period has passed.
Writing 0 to this bit while it is 1 returns the port to the HS Idle state. Until then, this bit remains at 1.
After software writes 0 to this bit, the host logic must complete the transition to the HS Idle state within 2 ms. If the port is in the L1 state, software does not need to write 0 to this bit because the host logic sends a resume signal when necessary and clears this bit after the port resumes. However, the length of the resume signal driven by the host logic must be set in bits [27:24] (Host-Initiated Resume Duration) in the USBCMD register.
When PP = 0, this bit becomes 0.
0: Resume (K-state) has not been detected or output
1: Resume (K-state) has been detected or output</td><td>R/W</td></tr>
<tr><td>7</td><td>SUSPEND</td><td>Port suspend
The combination of the settings of this bit and bit 2 (Port Enabled/Disabled) indicate the port state as follows.
<table>
<tr><th>PTE</th><th>SUSPEND</th><th>Port State</th></tr>
<tr><td>0</td><td>x</td><td>Disabled</td></tr>
<tr><td>1</td><td>0</td><td>Enabled</td></tr>
<tr><td>1</td><td>1</td><td>Suspend</td></tr>
</table>
Set this bit to 1 to transition the port to the L1 or L2 suspend state. The type of suspend state (L1 or L2) the host logic supports depends on the value of the Suspend using L1 bit.
In the suspend state, data transfer from this port to the downstream port is blocked unless the port is reset. When this bit is set to 1 during data transfer, reflection in the status bit and blocking of transfer follow completion of the current transfer.
Software should write 1 to this bit. It can write 1 only while PP = 1, PTOWNR = 0, and CCSTS = 1 in the register in the host logic.
This bit is always cleared to 0 under either of the following conditions:
● Software clears the Force Port Resume bit to 0
● Completion of resume is detected
● Software sets the Port Reset (PTRST) bit to 1
● Port Owner (PTOWNR) bit is set to 1 (OHCI)
● Port Power (PP) bit is set to 0
● Port Enabled/Disabled (PTE) bit is set to 0
This bit indicates the suspend state of the port.
0: Port is not in suspend state
1: Port is in suspend state</td><td>R/W</td></tr>
<tr><td>8</td><td>PTRST</td><td>Port reset status
When software writes 1 to this bit while it is 0, the bus reset sequence defined in the USB2.0 standard begins. Write 0 to this bit to terminate the bus reset sequence. Note that this bit must be retained at 1 for long enough time for the bus reset sequence to be completed according to the USB2.0 standard.
When bit 12 (HCHalted) in the USBSTS register is 1, the port should not be reset.
This bit becomes 0 when the PP, PTOWNR, or CCSTS bit satisfies the following conditions. Writing 1 does not start the bus reset sequence.
● PP = 0
● PTOWNR = 1
● CCSTS = 0
This bit indicates the reset state of the port.
0: Port is not in reset state
1: Port is in reset state</td><td>R/W</td></tr>
</table>

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>9</td><td>LPMCTL</td><td>LPM control
This bit is used to select Suspend using L1 (LPM) or Suspend using L2.
When 1 is written to bit 7 (SUSPEND) while this bit is 1, the host logic transitions the attached device to the LPM state.
When this bit is 1 and the Device Address bits are other than 0x0000, setting the Suspend bit to 1 makes the host logic create an LPM token for transitioning the attached device to the L1 state. When this bit is 0, the host logic uses the L2 suspend mechanism.
Writing to this bit is permitted only when bit 7 (SUSPEND) is 0. When SUSPEND = 1, both 0 and 1 cannot be written to this bit.
0: Suspend using L2
1: Suspend using L1 (LPM)</td><td>R/W</td></tr>
<tr><td>11:10</td><td>LINESTS[1:0]</td><td>D+/D− logic level
These bits indicate the current logical levels on the D+ and D− lines of the USB port (bit 11: D+, bit 10: D−).
They are used to detect an LS device before a port reset or a sequence for enabling the port. Therefore, the values in these bits are valid only while bit 3 (Port Enable/Disable) = 0 and bit 0 (Current Connect Status) = 1.
<table>
<tr><th>Bit 11 (D+)</th><th>Bit 10 (D−)</th><th>USB State</th><th>Interpretation</th></tr>
<tr><td>0</td><td>0</td><td>SE0</td><td>Not an LS device.
Moves to the EHCI port reset execution process.</td></tr>
<tr><td>0</td><td>1</td><td>K State</td><td>An LS device has been connected.
Passes the port ownership from EHCI to OHCI.</td></tr>
<tr><td>1</td><td>0</td><td>J State</td><td>Not an LS device.
Moves to the EHCI port reset execution process.</td></tr>
<tr><td>1</td><td>1</td><td>Undefined</td><td>Not an LS device.
Moves to the EHCI port reset execution process.</td></tr>
</table></td><td>R</td></tr>
<tr><td>12</td><td>PP</td><td>Port Power Supply Control (Port Power)
This bit controls power supply to the port.
When this bit is 0, power is not supplied to the port. The port does not operate and connection or disconnection cannot be detected.
If an overcurrent is detected while this bit is 1, the host logic clears this bit to 0 and stops power supply to the port. However, as explained later, when the PPC bit is 0, this bit is fixed to 1 and power supply to the port does not stop.
The function of this bit depends on the setting of bit 4 (PPC) in the HCSPARAMS register.
<table>
<tr><th>PPC</th><th>PP</th><th>Description</th></tr>
<tr><td>0</td><td>1</td><td>This bit is fixed to 1, and power is always supplied to the port</td></tr>
<tr><td>1</td><td>0/1</td><td>Power supply to the port is determined by this bit</td></tr>
</table>
0: No power is supplied to the port
1: Power is supplied to the port</td><td>R/W</td></tr>
<tr><td>13</td><td>PTOWNR</td><td>Port ownership
This bit indicates whether OHCI or EHCI has the port ownership.
This bit is cleared to 0 when bit 0 (CF) in the CONFIGFLAG register changes from 0 to 1.
This bit is set to 1 when the CF bit is 0.
Software should set this bit to 1 to pass the port ownership to OHCI when the connected device is not a high-speed device.
0: EHCI has the port ownership
1: OHCI has the port ownership</td><td>R/W</td></tr>
<tr><td>15:14</td><td>PTINDCTL[1:0]</td><td>As the host logic does not support the port indicator control function, these bits are set to 00b.
Writing to these bits does not affect the operation.</td><td>R</td></tr>
</table>

Bit	Symbol	Function	R/W
19:16	PTTST[3:0]	Pin test control These bits control the test mode. For details of the test mode, see Chapter 7 in the USB Specification Revision 2.0. When the value is other than 0000b, the host logic is operating in test mode. **Port Test Control[3:0] / Mode** 0000b: Normal 0001b: Test J_STATE 0010b: Test K_STATE 0011b: Test SE0_NAK 0100b: Test Packet 0101b: Test FORCE_ENABLE Other than the above: Reserved	R/W
20	WKCNNT_E	Device connection detection enable (Wake on Connect Enable) Writing 1 to this bit enables detection of device connection as a wakeup event. When PP = 0, this bit becomes 0.	R/W
21	WKDSCNNT_E	Device disconnection detection enable (Wake on Disconnect Enable) Writing 1 to this bit enables detection of device disconnection as a wakeup event. When PP = 0, this bit becomes 0.	R/W
22	WKOC_E	Overcurrent state detection enable (Wake on Over-current Enable) Writing 1 to this bit enables detection of the overcurrent state as a wakeup event. When PP = 0, this bit becomes 0.	R/W
24:23	SUSPSTS[1:0]	Suspend status These bits indicate a response from a connected device to an LPM token (L1 transition request). The host logic updates these bits immediately when the L1 transition request processing is completed. To avoid race conditions with hardware, a value must be set to these bits when SUSPEND = 0. 0 0: Device successfully entered the L1 state (ACK is received from the device) 0 1: Device was unable to enter the L1 state at this time (NYET is received from the device) 1 0: Device does not support L1 state transitions (STALL is received from the device) 1 1: Other than above (such as a timeout error)	R/W
31:25	DVADDR[6:0]	USB device address These bits indicate the USB device address (7 bits) of the device attached to the downstream port and are used when an LPM token is sent. When the value of these bits is 0x00, no device that requires these bits is connected.	R/W

29.2.3 AHB Bridge Registers

29.2.3.1 INTENABLE : INT_ENABLE Register

Base address: USBHC = 0x8020_0000

Offset address: 0x200

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	WAKEON_INTEN	UCOM_INTEN	USBH_INTBEN	USBH_INTAEN	AHB_INTEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AHB_INTEN	AHB_INT bit control This bit enables or disables the interrupt source indicated in bit 0 (AHB_INT) in the INT_STATUS register. 0: Disabled 1: Enabled	R/W
1	USBH_INTAEN	USBH_INTA bit control This bit enables or disables the interrupt source indicated in bit 1 (USBH_INTA) in the INT_STATUS register. 0: Disabled 1: Enabled	R/W
2	USBH_INTBEN	USBH_INTB bit control This bit enables or disables the interrupt source indicated in bit 2 (USBH_INTB) in the INT_STATUS register. 0: Disabled 1: Enabled	R/W
3	UCOM_INTEN	UCOM_INT bit control This bit enables or disables the interrupt source indicated in bit 3 (UCOM_INT) in the INT_STATUS register. 0: Disabled 1: Enabled	R/W
4	WAKEON_INTEN	WAKEON_INT bit control This bit enables or disables the interrupt source indicated in bit 4 (WAKEON_INT) in the INT_STATUS register. 0: Disabled 1: Enabled	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

The INTENABLE register enables or disables each interrupt source indicated in the INT_STATUS register. When an interrupt source is disabled, the interrupt signal is not asserted even when the interrupt source is generated and the corresponding bit in the INT_STATUS register is set to 1.

29.2.3.2 INTSTATUS : INT_STATUS Register

Base address: USBHC = 0x8020_0000

Offset address: 0x204

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	WAKEON_INT	UCOM_INT	USBH_INTB	USBH_INTA	AHB_INT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AHB_INT	AHB bus error indication This bit indicates that an AHB bus error has occurred. Writing 1 clears this bit. 0: AHB bus error has not occurred. 1: AHB bus error has occurred	R/W
1	USBH_INTA	OHCI interrupt status This bit indicates the state of the OHCI interrupt. The interrupt can be cleared using the HcInterruptStatus register (one of OHCI operational registers). 0: INTA interrupt has not occurred 1: INTA interrupt has occurred	R

Bit	Symbol	Function	R/W
2	USBH_INTB	USBH_INTB EHCI interrupt status This bit indicates the state of the USBH_INTB EHCI interrupt. The interrupt can be cleared using the USBSTS register (one of EHCI operational registers). 0: INTB interrupt has not occurred 1: INTB interrupt has occurred	R
3	UCOM_INT	UCOM register interrupt status This bit indicates the state of the interrupt from the UCOM register. The interrupt can be cleared using the UCOM2 register. 0: UCOM register interrupt has not occurred 1: UCOM register interrupt has occurred	R
4	WAKEON_INT	WAKEON interrupt status This bit indicates the state of the WAKEON interrupt from the host logic. Writing 1 clears this bit. 0: WAKEON interrupt has not occurred 1: WAKEON interrupt has occurred	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

The INTSTATUS register indicates the state of interrupt sources in the AHB bridge and the state of interrupt signals from the OHCI, EHCI, and UCOM modules.

29.2.3.3 AHBBUSCTR : AHB_BUS_CTR Register

Base address: USBHC = 0x8020_0000

Offset address: 0x208

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PROT_TYPE[3:0]				—	—	—	PROT_MODE	—	—	ALIGN_ADDRESS[1:0]		—	—	MAX_BURST_LEN[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	MAX_BURST_LEN[1:0]	Maximum burst length These bits select the maximum burst length when the AHB master interface initiates a transfer. 0 0: INCR16 0 1: INCR8 1 0: INCR4 1 1: SINGLE	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
5:4	ALIGN_ADDRESS[1:0]	Address boundary setting These bits set the address boundary when the AHB master interface initiates a burst transfer. 0 0: 1-KB boundary is not exceeded when initiating a burst transfer 0 1: 64-byte boundary is not exceeded when initiating a burst transfer 1 0: 32-byte boundary is not exceeded when initiating a burst transfer (The maximum burst length is INCR8 in this case. The 32-byte boundary is exceeded when the maximum burst length is INCR16.) 1 1: 16-byte boundary is not exceeded when initiating a burst transfer. (The maximum burst length is INCR4 in this case. The 16-byte boundary is exceeded when the maximum burst length is INCR8 or greater.)	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>8</td><td>PROT_MODE</td><td>This bit selects the MHPROT[3:0] mode when the AHB master interface initiates a transfer.
0: Value of PROT_TYPE[3:0] is output as MHPROT[3:0]
1: 0000b is set as MHPROT[3:0] for DMA transfer when the last data of a burst is transferred. The value of PROT_TYPE[3:0] is used as MHPROT[3:0] for other data in a burst.</td><td>R/W</td></tr>
<tr><td>11:9</td><td>—</td><td>These bits are read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>15:12</td><td>PROT_TYPE[3:0]</td><td>These bits set MHPROT[3:0] when the AHB master interface initiates a transfer.
<table>
<tr><th>Bit</th><th>Name</th><th>Value</th><th>Description</th></tr>
<tr><td>15</td><td>PROT_TYPE[3]</td><td>0</td><td>Cache is disabled</td></tr>
<tr><td></td><td></td><td>1</td><td>Cache is enabled</td></tr>
<tr><td>14</td><td>PROT_TYPE[2]</td><td>0</td><td>Buffer is disabled</td></tr>
<tr><td></td><td></td><td>1</td><td>Buffer is enabled</td></tr>
<tr><td>13</td><td>PROT_TYPE[1]</td><td>0</td><td>Access by a user</td></tr>
<tr><td></td><td></td><td>1</td><td>Privileged access</td></tr>
<tr><td>12</td><td>PROT_TYPE[0]</td><td>0</td><td>Operation code</td></tr>
<tr><td></td><td></td><td>1</td><td>Data</td></tr>
</table></td><td>R/W</td></tr>
<tr><td>31:16</td><td>—</td><td>These bits are read as 0. The write value should be 0.</td><td>R/W</td></tr>
</table>

The AHBBUSCTR register specifies the AHB master and slave functions.

29.2.3.4 USBCTR : USBCTR Register

Base address: USBHC = 0x8020_0000

Offset address: 0x20C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	DIRPD	PLL_RST	USBH_RST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Function	R/W
0	USBH_RST	Software reset for the core When this bit is set to 1, the entire core is reset. Set this bit when the AHB master interface in the core is inactive. Access to the host logic is enabled in 10 PCLKM clock cycles after this bit is written to. 0: Nothing occurs 1: Host logic is reset	W
1	PLL_RST	Reset of USB PHY PLL Reset to the PHY PLL is asserted. 0: Release PLL reset 1: Assert PLL reset	R/W
2	DIRPD	Direct transition to power-down state If this bit is set to 1, USB subsystem moves to power-down state. 0: Normal operation 1: Direct power-down state	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

29.2.4 Core Defined Registers

29.2.4.1 REVID : Revision and Core ID Register

Base address: USBHC = 0x8020_0000

Offset address: 0x300

Bit position:	31								24								15								8	7							0
Bit field:	COREID[7:0]								—	—	—	—	—	—	—	—	MAJV[7:0]								MINV[7:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
7:0	MINV[7:0]	Minor Version These bits indicate the minor version of the host logic.	R
15:8	MAJV[7:0]	Major Version These bits indicate the major version of the host logic.	R
23:16	—	This bit is read as 0.	R
31:24	COREID[7:0]	Core ID These bits indicate the ID of the host logic.	R

29.2.4.2 OCSLPTIMSET : Overcurrent Detection/Sleep Timer Setting Register

Base address: USBHC = 0x8020_0000

Offset address: 0x310

Bit position:	31			28								20	19																			0
Bit field:	—	—	—	TIMER_SLEEP[8:0]									TIMER_OC[19:0]																				
Value after reset:	0	0	0	0	1	1	0	0	1	0	0	0	0	0	1	1	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
19:0	TIMER_OC[19:0]	Overcurrent Timer setting When the overcurrent input, USB_OVRCUR, is asserted (driven to low) continuously longer than these bit setting time, the USB module determines the occurring overcurrent. This value should be set to more than 1 to 2 ms for the AHB clock and PCLKM frequencies. One bit means 1 clock cycle. 1 ms at PCLKM = 75 MHz: 0x1_24F8 1 ms at PCLKM = 100 MHz: 0x1_86A0 1 ms at PCLKM = 200 MHz: 0x3_0D04 (default setting)	R/W
28:20	TIMER_SLEEP[8:0]	Detection/Sleep Timer Setting These bits are timer setting for RemoteWakeup receive detection time during sleep and measuring Resume-K time. This value should be set to more than 1 µs for the AHB clock and PCLKM frequencies. One bit means 1 clock cycle. PCLKM = 75 MHz: 0x04B PCLKM = 100 MHz: 0x064 PCLKM = 200 MHz: 0x0C8 (default setting)	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

The OCSLPTIMSET register specifies the detection time of overcurent.

29.2.5 UCOM Registers

29.2.5.1 COMMCTRL : Common Control Register

Base address: USBHC = 0x8020_0000

Offset address: 0x800

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	PERI	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
30:0	—	These bits are read as 0. The write value should be 0.	R/W
31	PERI	USB mode setting This bit specified whether to place the USB module in host mode or function mode. 0: Host mode 1: Function mode	R/W

The COMMCTRL register specifies whether to use USB module in host mode or function mode.

29.2.5.2 OBINTSTA : OTG-BC Interrupt Status Register

Base address: USBHC = 0x8020_0000

Offset address: 0x804

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DPMONCHG_STA	DMMONCHG_STA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	VBSTAINT_STA	VBSTACHG_STA	OCINT_STA	IDCHG_STA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	IDCHG_STA	USB_OTGID change status This bit is set when the value that is input from the USB_OTGID pin has changed. The initial value is 1. Before using this bit, clear the status. 0: Value input from the USB_OTGID pin has not changed 1: Value input from the USB_OTGID pin has changed	R/W*1
1	OCINT_STA	USB_OVRCUR assertion status This bit is set when the USB_OVRCUR pin is asserted. 0: USB_OVRCUR pin is not asserted (remains 1) 1: USB_OVRCUR pin is asserted (change to 0)	R/W*1
2	VBSTACHG_STA	VBSTA[3:0] change status This bit is set when the value of bits [31:28] (VBSTA[3:0]) in the VBCTRL register has changed. 0: Value of the VBSTA[3:0] bits has not changed 1: Value of the VBSTA[3:0] bits has changed	R/W*1

Bit	Symbol	Function	R/W
3	VBSTAINT_STA	VBUS voltage status interrupt This bit is set when the value of bits [31:28] (VBSTA[3:0]) in the VBCTRL register equals to the value set for bits [23:20] (VBLVL[3:0]) in the VBCTRL register. This bit is set instantly when the value of the VBSTA[3:0] bits equals to the value of the VBLVL[3:0] bits. After that, if this bit is cleared and the value of the VBSTA[3:0] bits is unchanged, the bit is not set even if the values of the VBSTA[3:0] bits and the VBLVL[3:0] bits are the same. This bit is set again when the value of the VBSTA[3:0] bits changes and the values of the VBSTA[3:0] bits and the VBLVL[3:0] bits become the same again. 0: Value of VBSTA[3:0] bits is not the same as the value of the VBLVL[3:0] bits 1: Value of VBSTA[3:0] bits is the same as the value of the VBLVL[3:0] bits.	R/W*1
15:4	—	These bits are read as 0. The write value should be 0.	R/W
16	DMMONCHG_STA	DMMON change status This bit is set if the DMMON bit of the LINECTRL1 register has changed. 0: The DMMON bit of the LINECTRL1 register has not changed 1: The DMMON bit of the LINECTRL1 register has changed	R/W*1
17	DPMONCHG_STA	DPMON change status This bit is set if the DPMON bit of the LINECTRL1 register has changed. 0: The DPMON bit of the LINECTRL1 register has not changed 1: The DPMON bit of the LINECTRL1 register has changed	R/W*1
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing 1 clears this bit. Writing 0 has no effect.

The OBINTSTA register indicates whether interrupt sources related to USB OTG have occurred.

An OBINT interrupt is generated when a bit is set in this register and the corresponding bit is not masked in the OBINTEN register.

29.2.5.3 OBINTEN : OTG-BC Interrupt Enable Register

Base address: USBHC = 0x8020_0000

Offset address: 0x808

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DPMONCHG_EN	DMMONCHG_EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	VBSTAINT_EN	VBSTACHG_EN	OCINT_EN	IDCHG_EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IDCHG_EN	IDCHG_STA Interrupt enable This bit enables or disables the interrupt flagged by the IDCHG_STA bit. 0: Interrupt flagged by the IDCHG_STA bit is disabled. 1: Interrupt flagged by the IDCHG_STA bit is enabled	R/W
1	OCINT_EN	OCINT_STA interrupt enable This bit enables or disables the interrupt flagged by the OCINT_STA bit. 0: Interrupt flagged by the OCINT_STA bit is disabled 1: Interrupt flagged by the OCINT_STA bit is enabled	R/W
2	VBSTACHG_EN	VBSTACHG_STA interrupt enable This bit enables or disables the interrupt flagged by the VBSTACHG_STA. 0: Interrupt flagged by the VBSTACHG_STA bit is disabled 1: Interrupt flagged by the VBSTACHG_STA bit is enabled	R/W

Bit	Symbol	Function	R/W
3	VBSTAINT_EN	VBSTAINT_STA interrupt enable This bit enables or disables the interrupt flagged by the VBSTAINT_STA. 0: Interrupt flagged by the VBSTAINT_STA bit is disabled 1: Interrupt flagged by the VBSTAINT_STA bit is enabled	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
16	DMMONCHG_EN	DMMONCHG_STA interrupt enable This bit enables or disables the interrupt flagged by the DMMONCHG_STA. 0: Interrupt flagged by the DMMONCHG_STA bit disabled 1: Interrupt flagged by the DMMONCHG_STA bit enabled	R/W
17	DPMONCHG_EN	DPMONCHG_STA interrupt enable This bit enables or disables the interrupt flagged by the DPMONCHG_STA. 0: Interrupt flagged by the DPMONCHG_STA bit disabled 1: Interrupt flagged by the DPMONCHG_STA bit enabled	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The OBINTEN register enables or disables the interrupts that are flagged by the OBINTSTA register.

An OBINT interrupt is generated when a bit is set the OBINTSTA register and the corresponding bit is set in the OBINTEN register.

29.2.5.4 VBCTRL : VBUS Control Register

Base address: USBHC = 0x8020_0000

Offset address: 0x80C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	VBSTA[3:0]				—	—	—	—	VBLVL[3:0]				—	—	OCISEL	OCCLRIEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	VGPUO	—	—	VBUSENSEL	VBOUT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	VBOUT	VBUS drive control (USB_VBUSEN pin) This bit is used to assert the VBUS power path by controlling external power supply IC. 0: VBUS output is disabled. (USB_VBUSEN is driven to low.) 1: VBUS output is enabled. (USB_VBUSEN is driven to high.)	R/W
1	VBUSENSEL	USB_VBUSEN pin control 0: USB_VBUSEN is controlled by PPON1 1: USB_VBUSEN is controlled by VBOUT bit in this register	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	VGPUO	USB_EXICEN pin control This bit is used to control external power supply IC. 0: USB_EXICEN is driven to low 1: USB_EXICEN is driven to high	R/W
15:5	—	These bits are read as 0. The write value should be 0.	R/W
16	OCCLRIEN	USB_VBUSEN pin control at occurrence of overcurrent This bit specifies whether to automatically clear the VBOUT bit if an overcurrent occurs. If an overcurrent occurs, bit 1 (OCINT_STA) in the OBINTSTA register is set. 0: VBOUT bit is not automatically cleared if an overcurrent occurs 1: VBOUT bit is automatically cleared if an overcurrent occurs	R/W

Bit	Symbol	Function	R/W
17	OCISEL	Overcurrent detection This bit selects the signal to be used for detecting an overcurrent. 0: OCI signal that is output by the host logic This signal is treated as an indication of an overcurrent if the USB_OVRCUR pin is asserted longer than the overcurrent detection time set in bits [19:0] (TIMER_OC[19:0]) in the OCSLPTIMSET register. 1: Inverted signal from the USB_OVRCUR pin When this signal is selected, the overcurrent detection time set in the TIMER_OC[19:0] bits is not used. The system assumes an overcurrent has occurred if the USB_OVRCUR pin is asserted for 5 clock cycles or more.	R/W
19:18	—	These bits are read as 0. The write value should be 0.	R/W
23:20	VBLVL[3:0]	VBUS level detection These bits are used to detect, using an interrupt, that VBUS has reached a certain level. When the value of the voltage level of VBUS equals to the value set for these bits, the interrupt flagged by the VBSTAINT_STA bit in the OBINTSTA register is generated.	R/W
27:24	—	These bits are read as 0. The write value should be 0.	R/W
31:28	VBSTA[3:0]	VBUS indication These bits indicate the voltage level of VBUS. The bits indicate the input from the USB_VBUSIN pin as is. VBSTA[3:2] and VBSTA[0] bits are fixed to 0. VBSTA[1] bit indicates USB_VBUSIN input level.	R

The VBCTRL register controls VBUS when the USB module is used in OTG mode.

29.2.5.5 LINECTRL1 : Line Control Port 1 Register

Base address: USBHC = 0x8020_0000

Offset address: 0x810

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	DPRPD_EN	DP_RPD	DMRPD_EN	DM_RPD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	DPMON	DMMON	—	IDMON
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*1

Bit	Symbol	Function	R/W
0	IDMON	Indicates a value of USB_OTGID input pin.	R
1	—	This bit is read as 0.	R
2	DMMON	Indicates a value of USB bus DM.	R
3	DPMON	Indicates a value of USB bus DP.	R
15:4	—	These bits are read as 0. The write value should be 0.	R/W
16	DM_RPD	Controls USB bus (DM) 15 kΩ pulldown resistor when DMPPD_EN = 1. 0: DM-side 15 kΩ pulldown resistor is OFF 1: DM-side 15 kΩ pulldown resistor is ON	R/W
17	DMRPD_EN	Enables DM_RPD to control USB bus (DM) 15 kΩ pulldown resistor. 0: Control of DM-side 15 kΩ pulldown resistor by DM_RPD disabled 1: Control of DM-side 15 kΩ pulldown resistor by DM_RPD enabled	R/W
18	DP_RPD	Controls USB bus (DP) 15 kΩ pulldown resistor when DRPPD_EN = 1. 0: DP-side 15 kΩ pulldown resistor is OFF 1: DP-side 15 kΩ pulldown resistor is ON	R/W
19	DPRPD_EN	Enables DP_RPD to control USB bus (DP) 15 kΩ pulldown resistor. 0: Control of DP-side 15 kΩ pulldown resistor by DP_RPD disabled 1: Control of DP-side 15 kΩ pulldown resistor by DP_RPD enabled	R/W

Bit	Symbol	Function	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. This value depends on the USB_OTGID input pin.

29.3 Clock Lines

29.3.1 Externally Supplied Clocks

Table 29.1 describes the two clock signals that are supplied from within the LSI to the USB host controller.

Table 29.1 List of externally supplied clocks

Clock signal	Function	Frequency
PCLKM	AHB clock	100 MHz or 75 MHz
USB_CLK48	Reference clock	48 MHz

29.4 Interrupts

29.4.1 Interrupt Control Registers

29.4.1.1 U2H_INT Control Registers

U2H_INT is an interrupt signal generated by the AHB bridge. Registers in the AHB bridge are used to check and clear the interrupt state and to enable interrupts.

Table 29.2 U2H_INT Control Registers

Operation	Control Register
Check and clear interrupt state	INT_STATUS register
Enable interrupts	INT_ENABLE register

29.4.1.2 U2H_OHCI_INT Control Registers

U2H_OHCI_INT is the INTA interrupt signal from the host logic. Each interrupt is basically controlled through registers in the host logic, but to assert the interrupt signal, the interrupt enable bit in the AHB bridge must be set.

Table 29.3 U2H_OHCI_INT Control Registers

Operation	Control Register
Check and clear interrupt state	HcInterruptStatus register
Enable interrupts	HcInterruptEnable register HcInterruptDisable register INT_ENABLE register (bit 1 (USBH_INTAEN))

29.4.1.3 U2H_EHCI_INT Control Registers

U2H_EHCI_INT is the INTB interrupt signal from the host logic. Each interrupt is basically controlled through registers in the host logic, but to assert the interrupt signal, the interrupt enable bit in the AHB bridge must be set.

Table 29.4 U2H_EHCI_INT Control Registers

Operation	Control Register
Check and clear interrupt state	USBSTS register
Enable interrupts	USBINTR register INT_ENABLE register (bit 2 (USBH_INTBEN))

29.4.1.4 U2H_OBINT Control Registers

U2H_OBINT is the UCOM interrupt signal from the host logic. Each interrupt is basically controlled through registers in the host logic, but to assert the interrupt signal, the interrupt enable bit in the AHB bridge must be set.

Table 29.5 U2H_OBINT Control Registers

Operation	Control Register
Check and clear interrupt state	OBINTSTA register
Enable interrupts	OBINTEN register INT_ENABLE register (bit 3 (UCOM_INTEN))

29.4.1.5 U2H_WAKEON_INT Control Registers

U2H_WAKEON_INT is the WAKEON interrupt signal from the host logic. Each interrupt is basically controlled through registers in the host logic, but to assert the interrupt signal, the interrupt enable bit in the AHB bridge must be set.

Table 29.6 U2H_WAKEON_INT Control Registers

Operation	Control Register
Check and clear interrupt state	INT_STATUS register (bit 4 (WAKEON_INT))
Enable interrupts	PORTSC1 register (bits 22 to 20) INT_ENABLE register (bit 4 (WAKEON_INTEN))

29.4.2 U2H_BIND_INT

The U2H_BIND_INT interrupt signal is generated as the logical OR of the following five interrupt source signals:

- U2H_INT
- U2H_OHCI_INT
- U2H_EHCI_INT
- U2H_OBINT
- U2H_WAKEON_INT.

The states of the U2H_OBINT, U2H_WAKEON_INT, U2H_OHCI_INT, and U2H_EHCI_INT signals are reflected in the INT_STATUS register in the AHB bridge. Read the register to check which source was responsible for a generated interrupt.

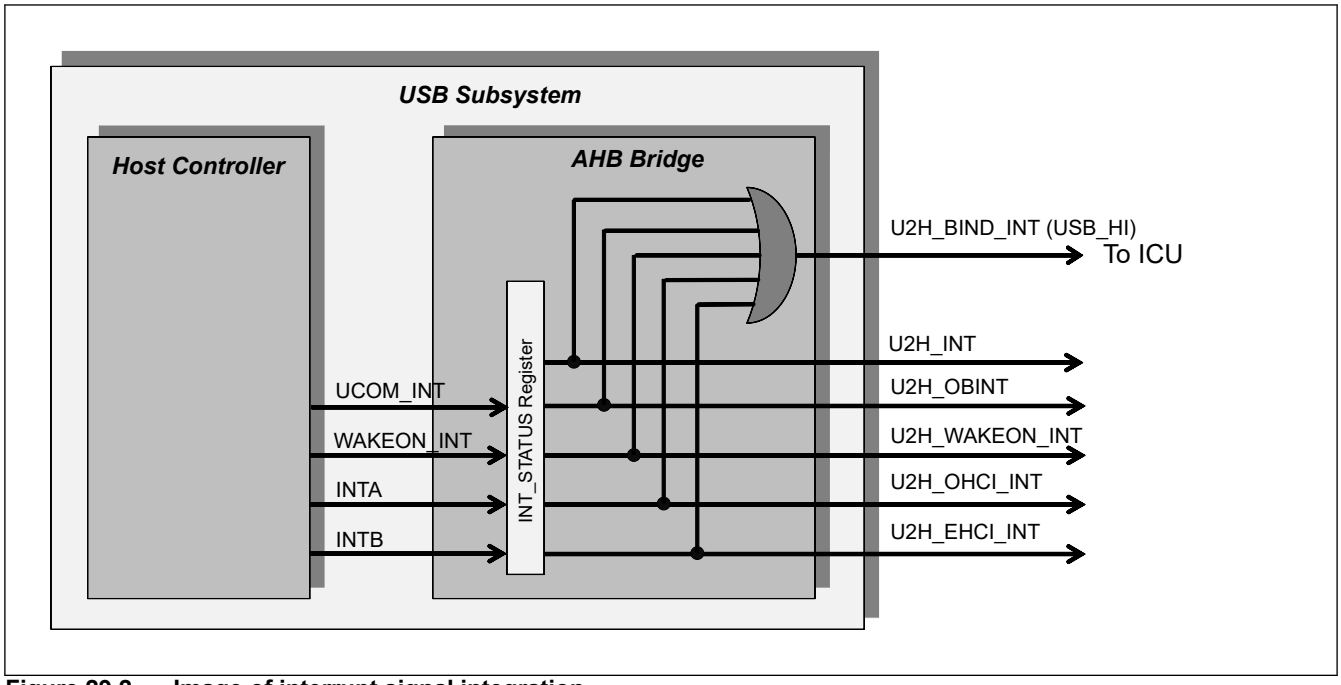


Figure 29.2 Image of interrupt signal integration

29.4.3 Time Required to Clear Interrupt Signals

It may take a long time to actually clear the interrupt after the register for clearing the interrupt source is written to due to the posted write operation of the AHB bridge.

Measure should be taken to prevent false detection before the next interrupt is detected after the register access for clearing the interrupt is completed.

29.5 Overcurrent Control and VBUS Control

29.5.1 Overcurrent Control

This section describes the operation of the USB_OVRCUR and USB_VBUSEN signals used for overcurrent detection at the USB port and VBUS control.

29.5.1.1 USB_OVRCUR and USB_VBUSEN Signal Functions

Table 29.7 shows the meaning of the USB_OVRCUR and USB_VBUSEN signals.

Table 29.7 USB_OVRCUR and USB_VBUSEN

Pin	I/O	Level	Meaning
USB_OVRCUR	Input	1	No overcurrent has been detected
		0	Overcurrent has been detected
USB_VBUSEN	Output	1	Power supply to VBUS is on
		0	Power supply to VBUS is off

29.5.1.2 Conditions for Asserting and Deasserting USB_VBUSEN Output Signal

Figure 29.3 shows a timing diagram for the assertion and deassertion of USB_OVRCUR and USB_VBUSEN.

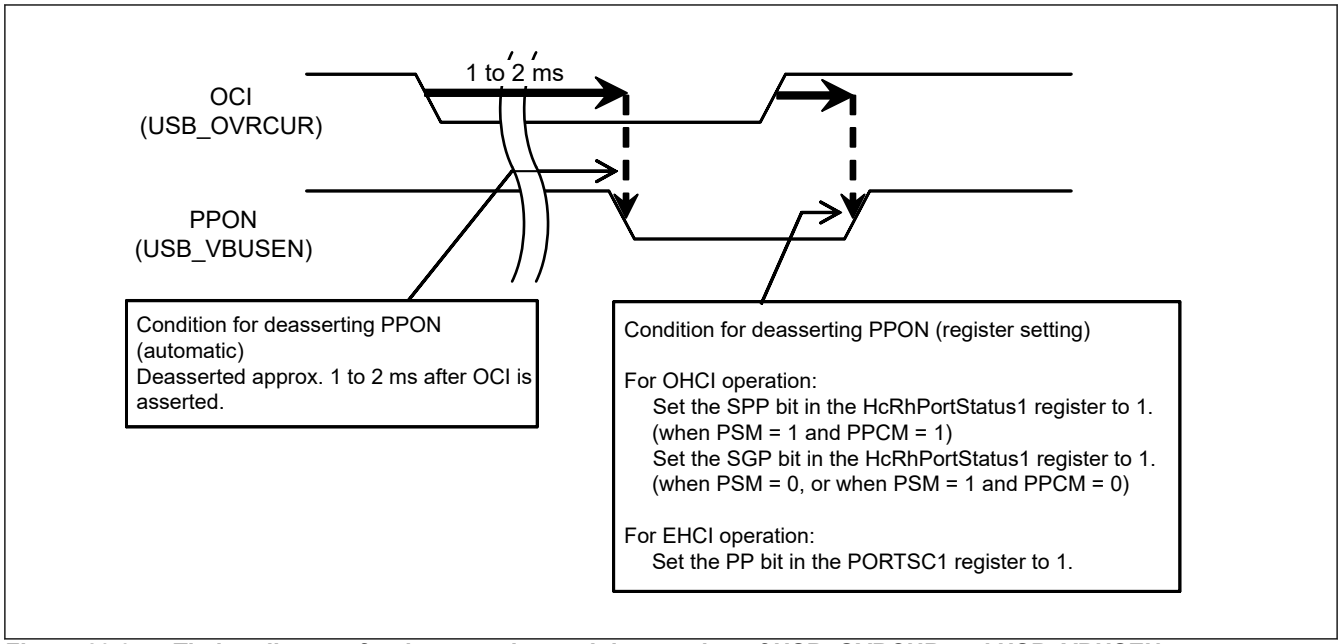


Figure 29.3 Timing diagram for the assertion and deassertion of USB_OVRCUR and USB_VBUSEN

USB_VBUSEN is not asserted automatically after USB_OVRCUR is deasserted. USB_VBUSEN is asserted when software sets the port power bit after USB_OVRCUR is deasserted.

29.5.2 VBUS Control

When the USB port is not used, VBUS can be stopped to reduce power consumption by connecting the USB_OVRCUR and USB_VBUSEN pins to the high-side switch (although this may not be possible depending on the connections with peripheral circuits).

Table 29.8 shows the relationship between USB_VBUSEN and VBUS.

Table 29.8 Relationship between USB_VBUSEN and VBUS

USB_VBUSEN	VBUS
0	Stopped
1	Operated

The USB_VBUSEN operation when USB_OVRCUR is asserted depends on the settings in the OHCI operational register as shown in Table 29.9.

Table 29.9 Register settings and USB_VBUSEN operation

OHCI operational registers				
HcRhDescriptorA Register			HcRhDescriptorB	
NOCP bit	NPS bit	PSM bit	PPCM[1] bit	USB_VBUSEN output pin operation
1	—	—	—	Fixed to 1
—	1	—	—	Fixed to 1
0	0	0	—	Deasserted (0) when USB_OVRCUR is asserted (0)
		1	0	
			1	

Note: When NPS = 1, the host logic detects an overcurrent but does not deassert USB_VBUSEN.

29.5.3 Initial Settings of PPON1

ON and OFF of PPON1 (PortPower) is controlled by the Port Control register of OHCI/EHCI operational registers listed in Table 29.10 for general usage after reset without setting the NOCP and NPS bits in the HcRhDescriptorA register. PPON1 is turned on when at least one of the following bits is ON:

- SPP
- SGP
- PP

In general, bits SPP, SGP, and PP are cleared to 0 when an overcurrent is detected.

Table 29.10 OHCI and EHCI operational registers

Register		Bit	Symbol
OHCI operation	HcRhPortStatus1 register	8	SPP
	HcRhPortStatus register	16	SGP*1
EHCI operation	PORTSC1 register	12	PP

Note 1. This is not generally used with the single-port version. For PPON control by this bit, the bit must be in the state of PSM = 0, or PSM = 1 and PPCM = 0.

PPON1 is turned on by setting the NOCP or NPS bit to 1 regardless of the Port Power bit of the specified ports.

The initial settings of the PPON1 control bits (SPP, SGP, PP, NOCP, and NPS) are all 0 (OFF). Setting these bits to 1 (ON) during initial setting required a procedure in the following flow.

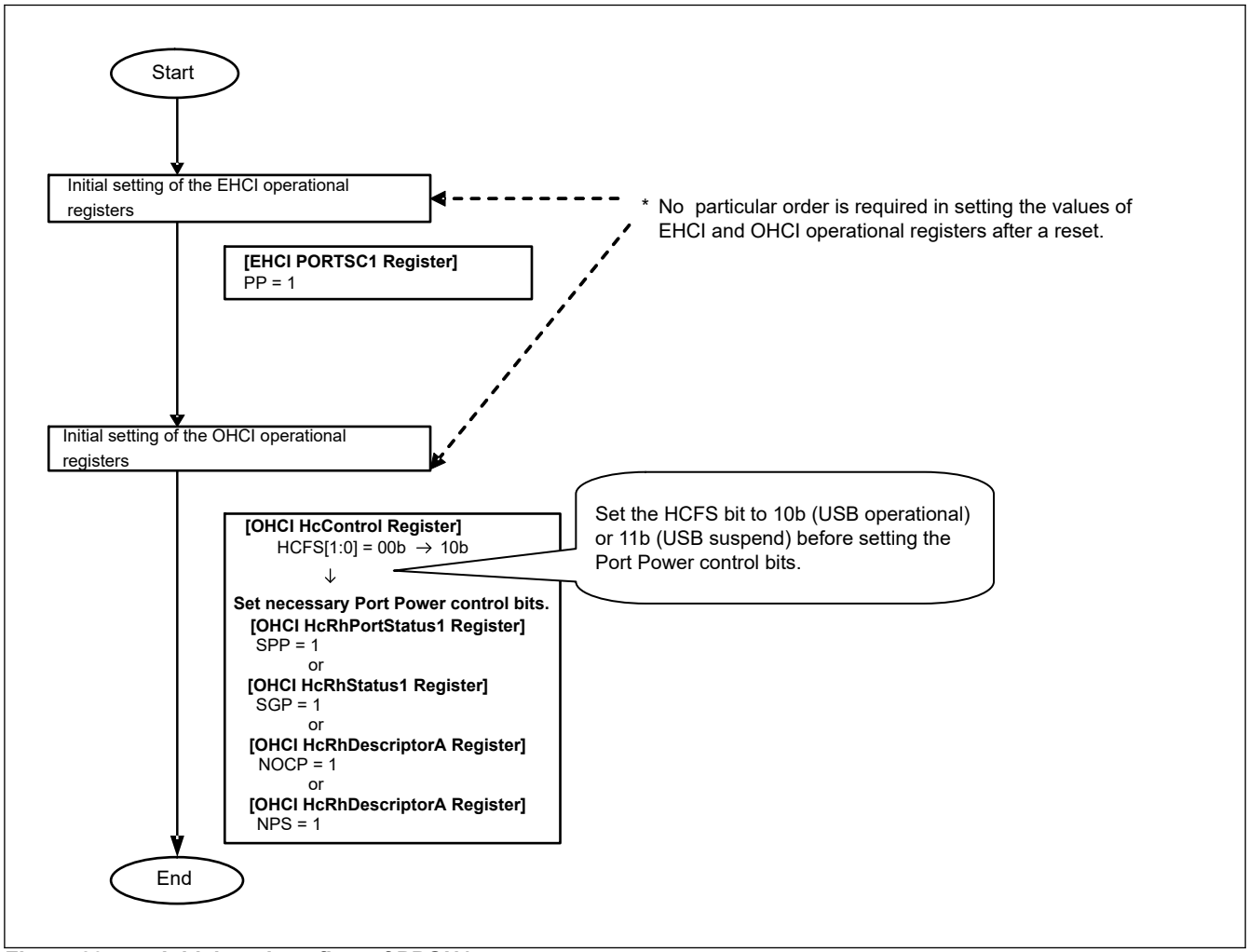


Figure 29.4 Initial settings flow of PPON1

29.5.4 Procedure for USB_VBUSEN Control for Overcurrent Detection

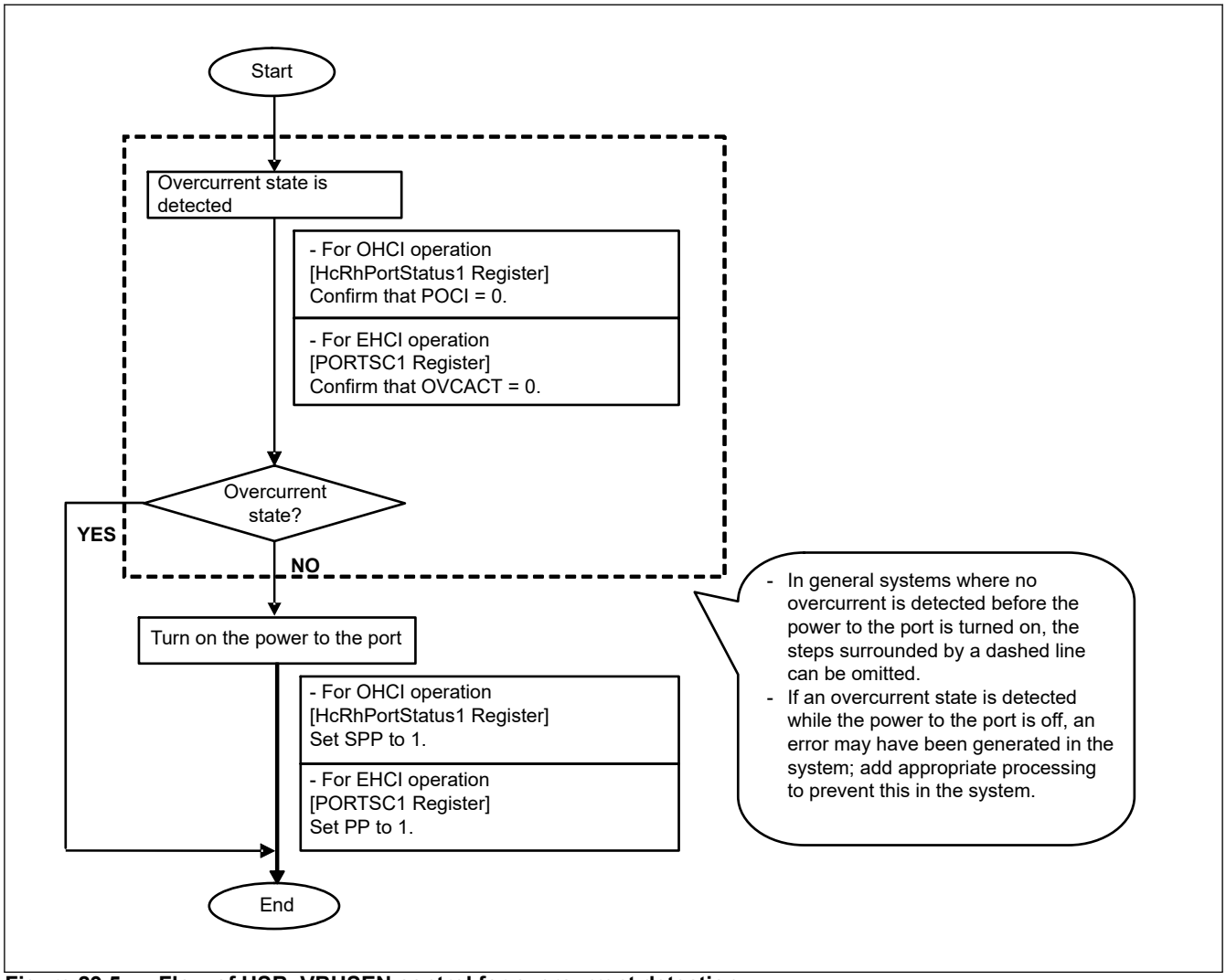


Figure 29.5 Flow of USB_VBUSEN control for overcurrent detection

29.5.5 Procedure for USB_VBUSEN Setting

Figure 29.6 shows the procedure for USB_VBUSEN setting in a system where USB_OVRCUR may be active (0) at system startup.

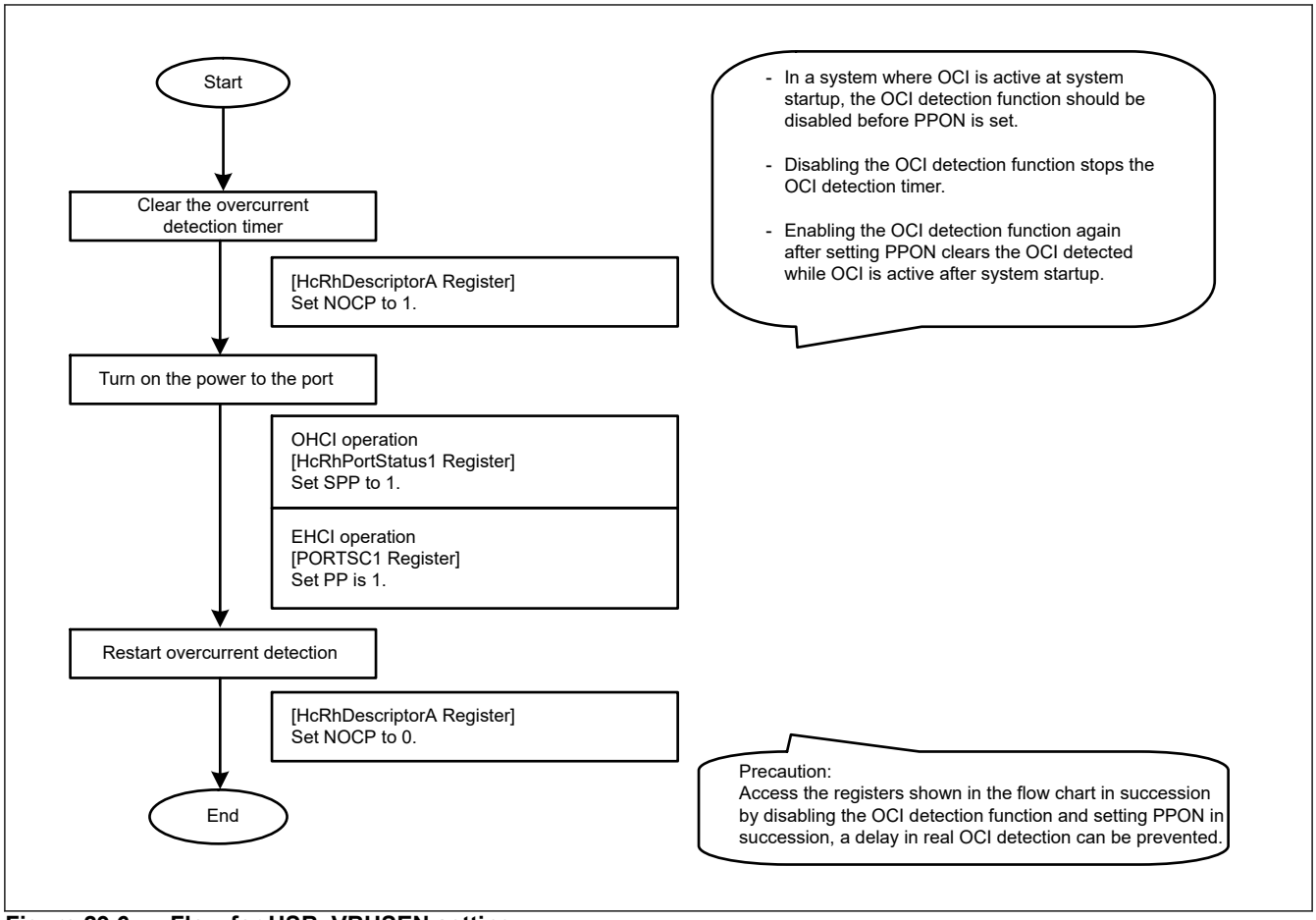


Figure 29.6 Flow for USB_VBUSEN setting

29.6 Initial Setting Sequence

29.6.1 Sample of Initial Settings

This section shows sample initial settings to implement the following functions:

- Access to OHCI/EHCI operational registers
- Data transfer from the host logic to the AHB bus.

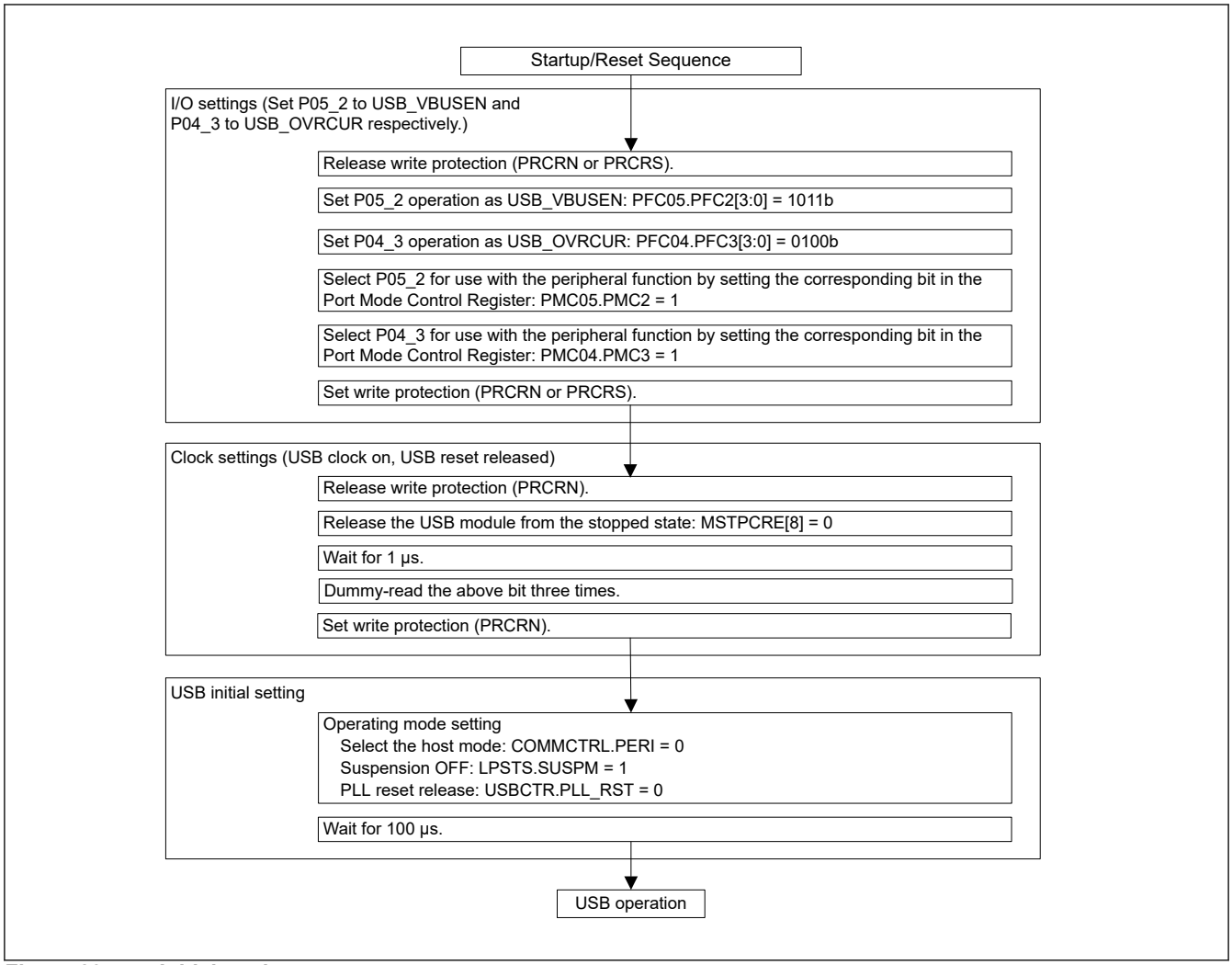


Figure 29.7 Initial setting sequence

29.7 OTG Control

29.7.1 Port Power (VBUS) Control and Connection with External Power-Supply IC

The following procedure describes how to control the port power (VBUS) when using this module for OTG:

1. Control the port power through the associated UCOM registers.
2. Connect the external power-supply IC and the corresponding pins of this module according to the following table.

LSI pin name	I/O	Connection
USB_OTGID	I	Connect this pin as input from the IC power supply
USB_VBUSIN	I	Connect this pin as input from the IC power supply
USB_VBUSEN	O	Connect this pin as output to the IC power supply
USB_EXICEN	O	Connect this pin as output to the IC power supply
USB_OVRCUR	I	Connect this pin as input from the IC power supply If the power-supply IC does not support overcurrent detection capability, this input signal should be tied to high level.

29.7.2 OTG System Configuration Example

Figure 29.8 shows an example of connections to support the OTG function. The configuration includes an external power-supply IC.

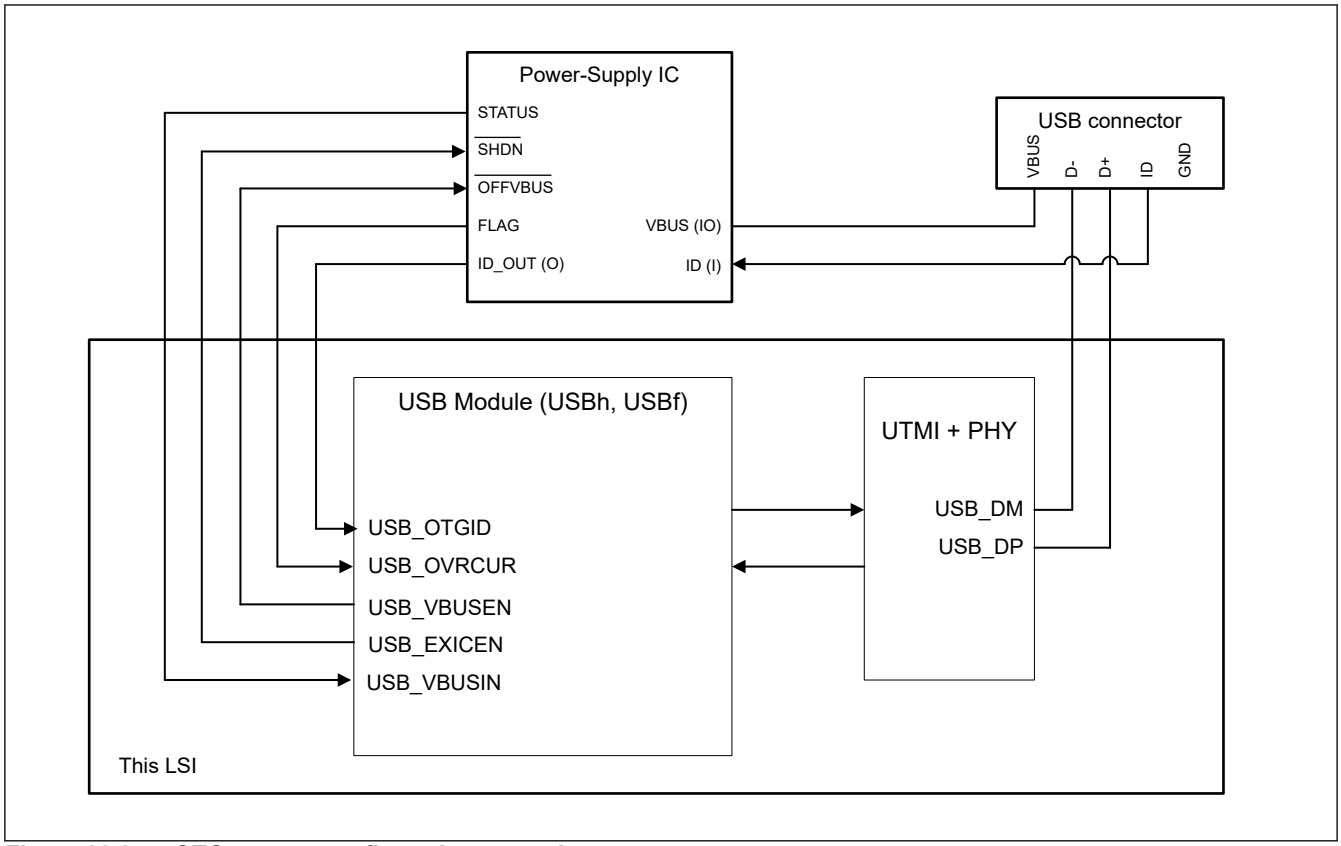


Figure 29.8 OTG system configuration example

29.7.3 OTG Initial Setting Procedure

Figure 29.9 shows the initial setting procedure when this module is used for OTG. The figure shows the steps before detection of A-plug or B-plug connection through the ID and VBUS pins.

Assuming the control of ID and VBUS:

VBUS control: VBCTRL.VBOUT

ID detection: Power-supply IC

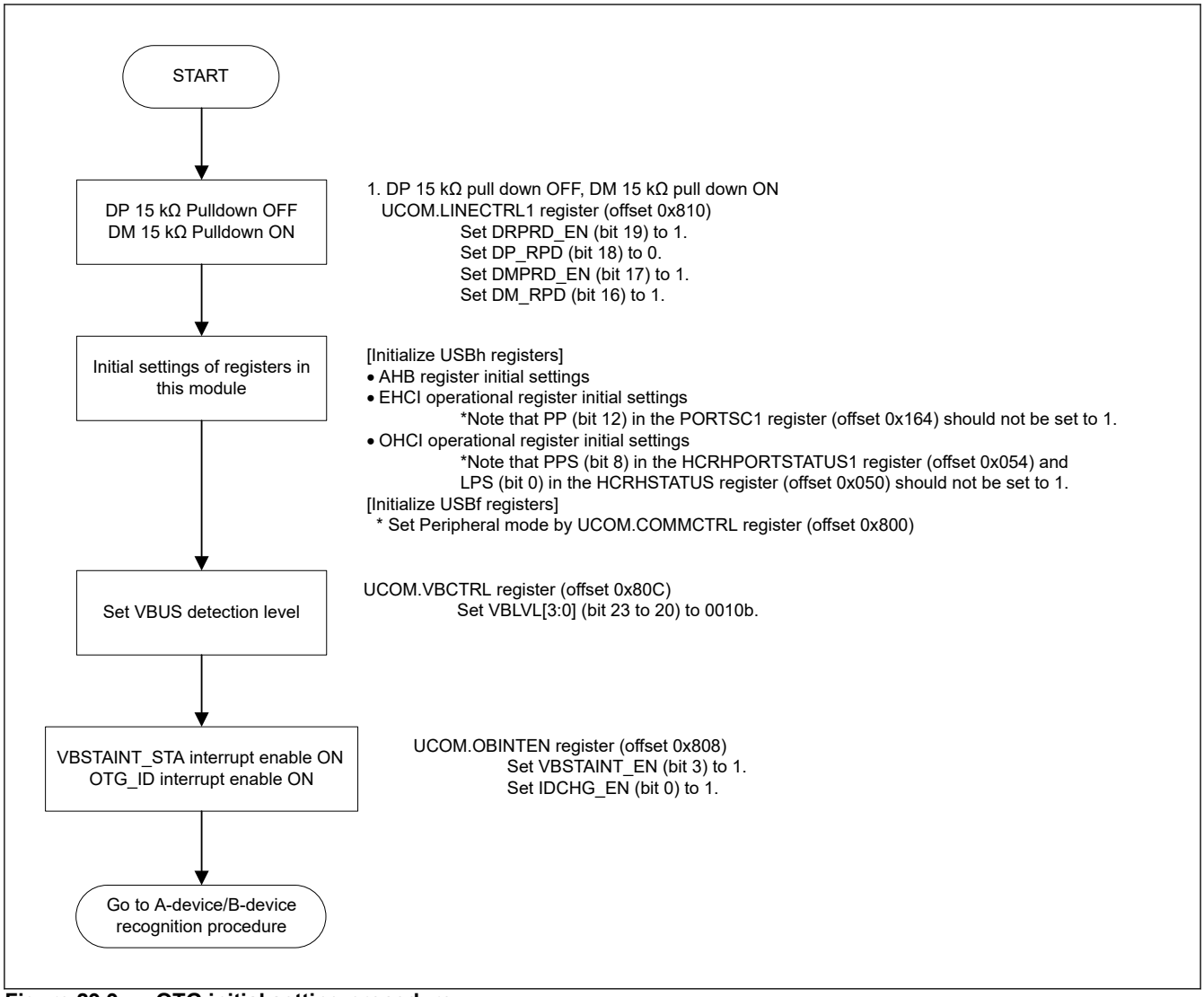


Figure 29.9 OTG initial setting procedure

29.7.4 A-Device/B-Device Recognition Procedure

Figure 29.10 shows the control procedure for operating this module as an A-device or a B-device through A-plug or B-plug connection following the steps described in section 29.7.3. OTG Initial Setting Procedure or section 29.7.5. Disconnect Detection Procedure.

Assuming the control of ID and VBUS:

VBUS control: VBCTRL.VBOUT

ID detection: Power-supply IC

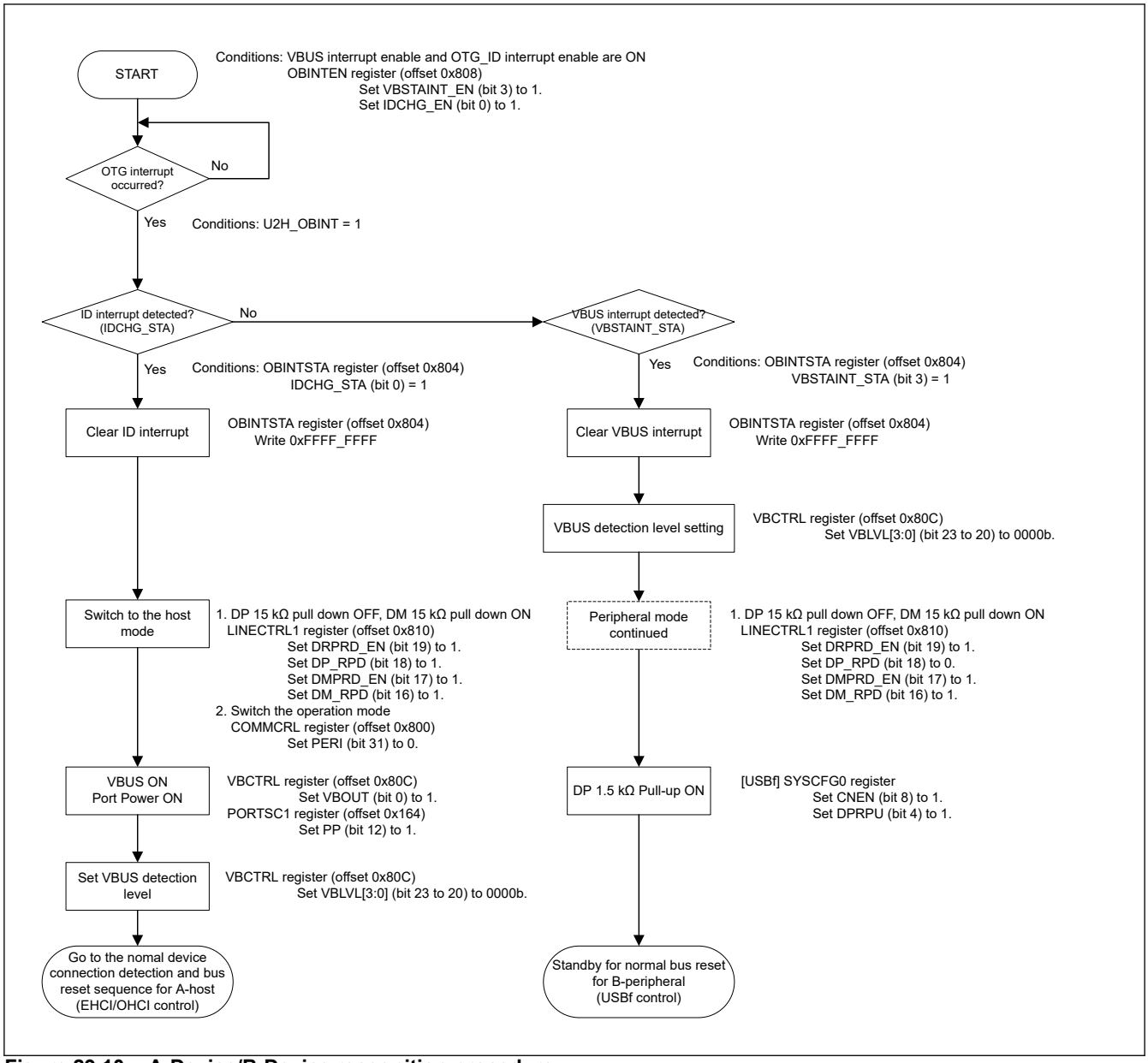


Figure 29.10 A-Device/B-Device recognition procedure

29.7.5 Disconnect Detection Procedure

Figure 29.11 shows the disconnect detection procedure. Disconnect detection means detection of a physical disconnection of an A-plug or B-plug from a USB port.

Assuming the control of ID and VBUS:

VBUS control: VBCTRL.VBOUT

ID detection: Power-supply IC

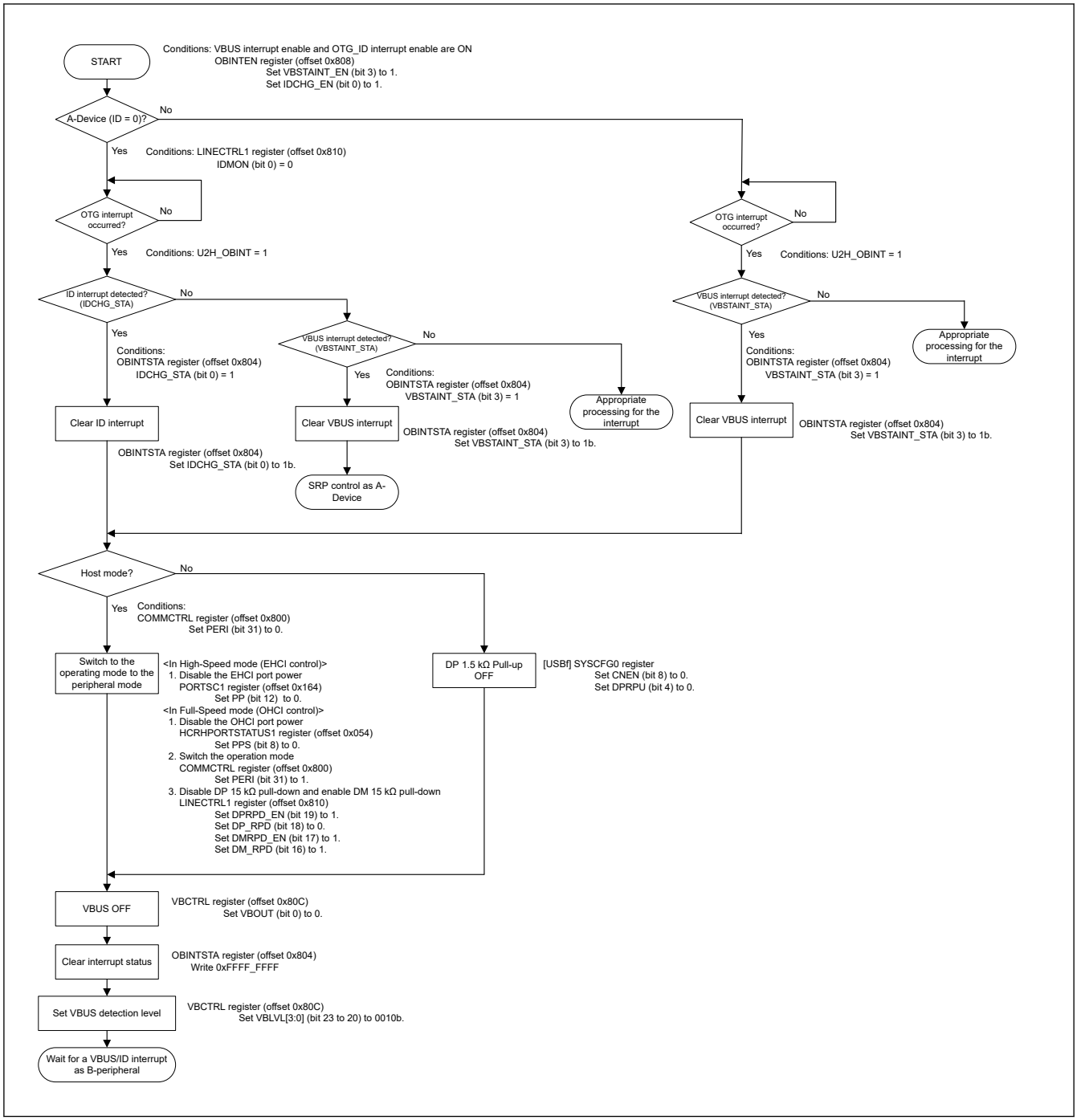


Figure 29.11 Disconnect detection procedure

29.7.6 HNP Control Procedure (Conforming to OTG Revision 2.0)

29.7.6.1 HNP Control Procedure for A-Device (Switching from A-Host to A-Peripheral)

Figure 29.12 shows the control procedure after a request from B-device for the host role is detected from the result of HNP polling for the B-device.

For HNP Control in High-Speed Mode (controlled by EHCI):

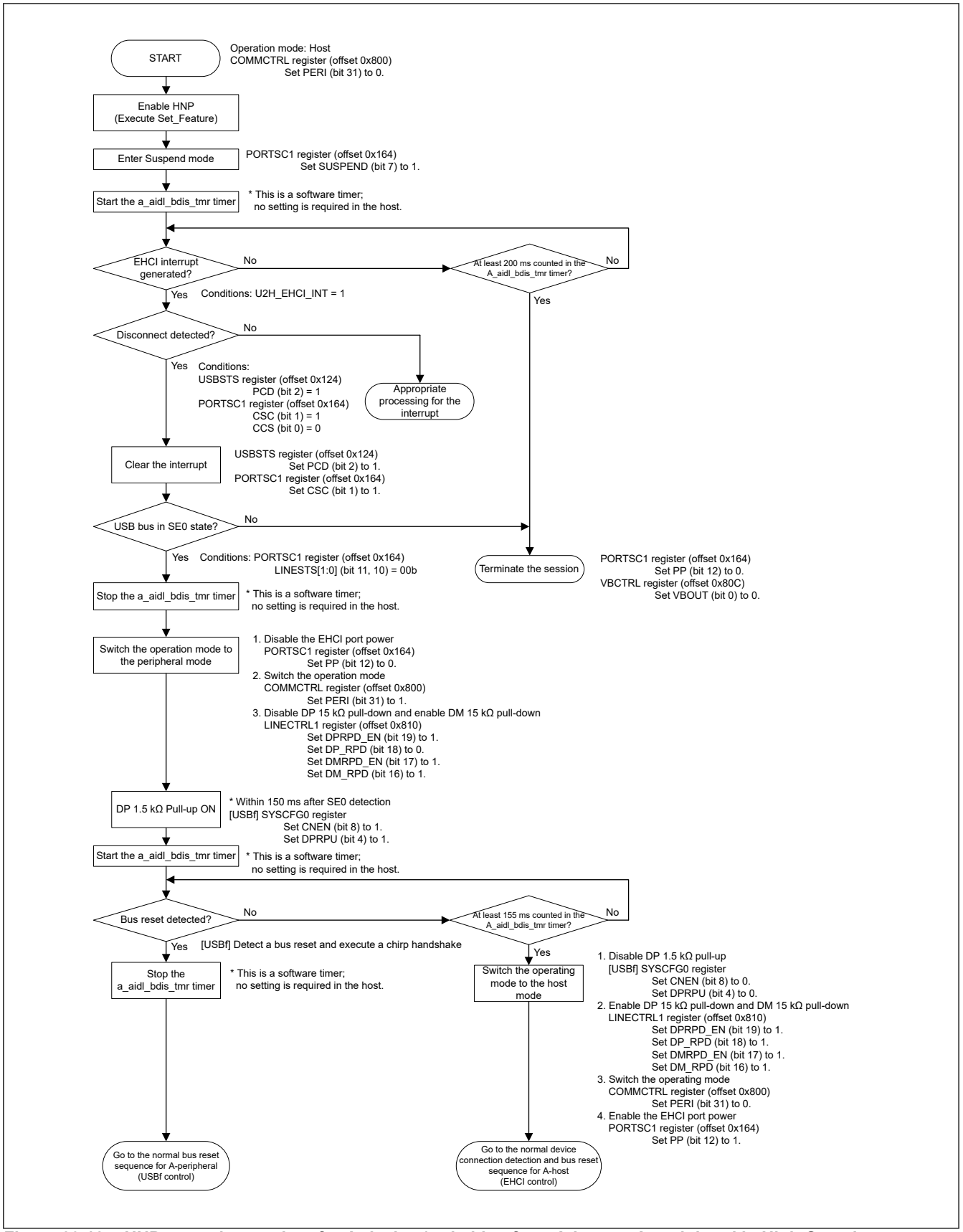


Figure 29.12 HNP control procedure for A-device (switching from A-host to A-peripheral in High-Speed mode)

For HNP control in Full-Speed mode (controlled by OHCI):

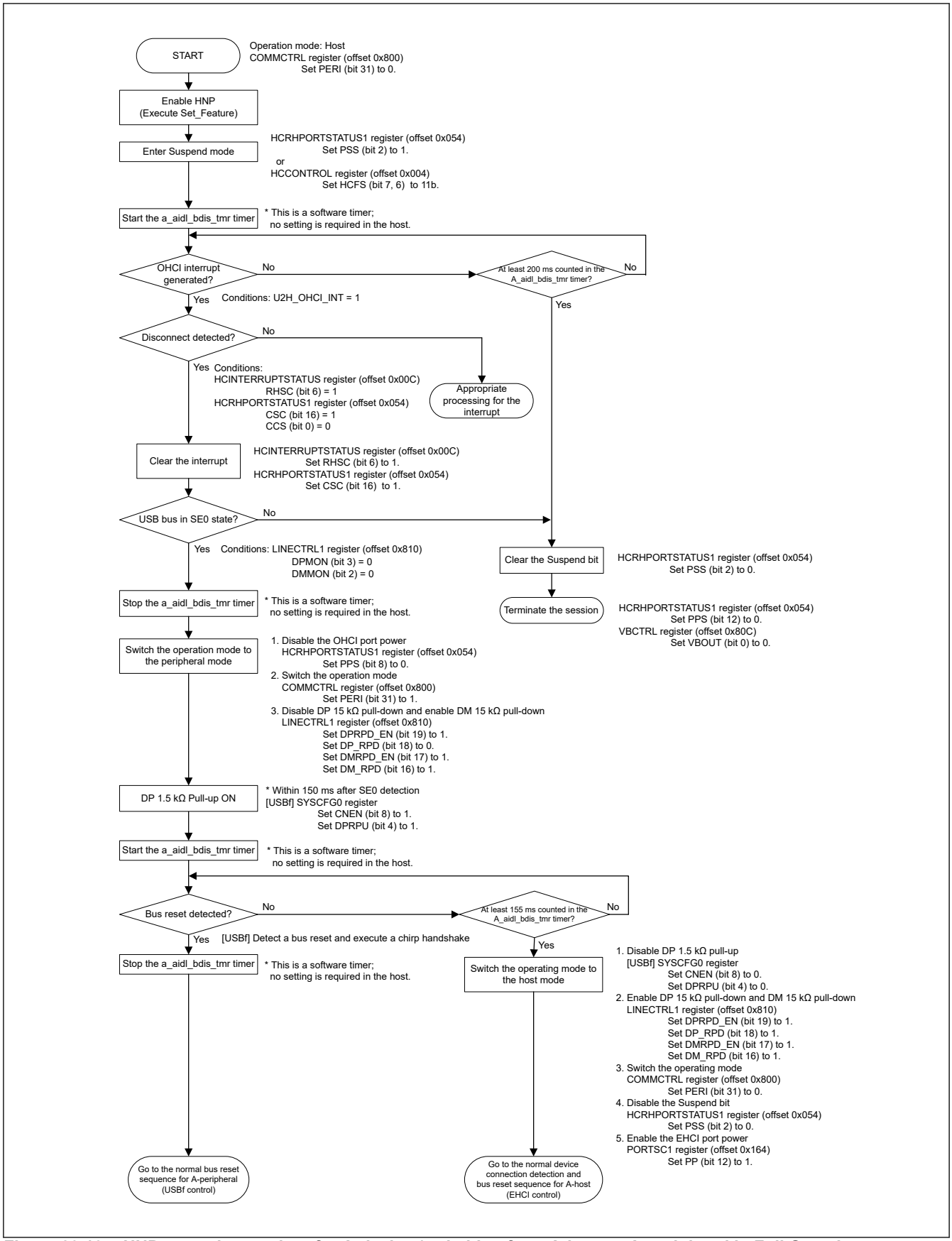


Figure 29.13 HNP control procedure for A-device (switching from A-host to A-peripheral in Full-Speed mode)

29.7.6.2 HNP Control Procedure for A-Device (Switching from A-Peripheral to A-Host)

Assumed conditions:

Even for FS OTG, EHCI is assumed to be the port owner after switching of the role from peripheral to host. Therefore, the procedure for switching from A-peripheral to A-host is not separately provided for each of OHCI and EHCI.

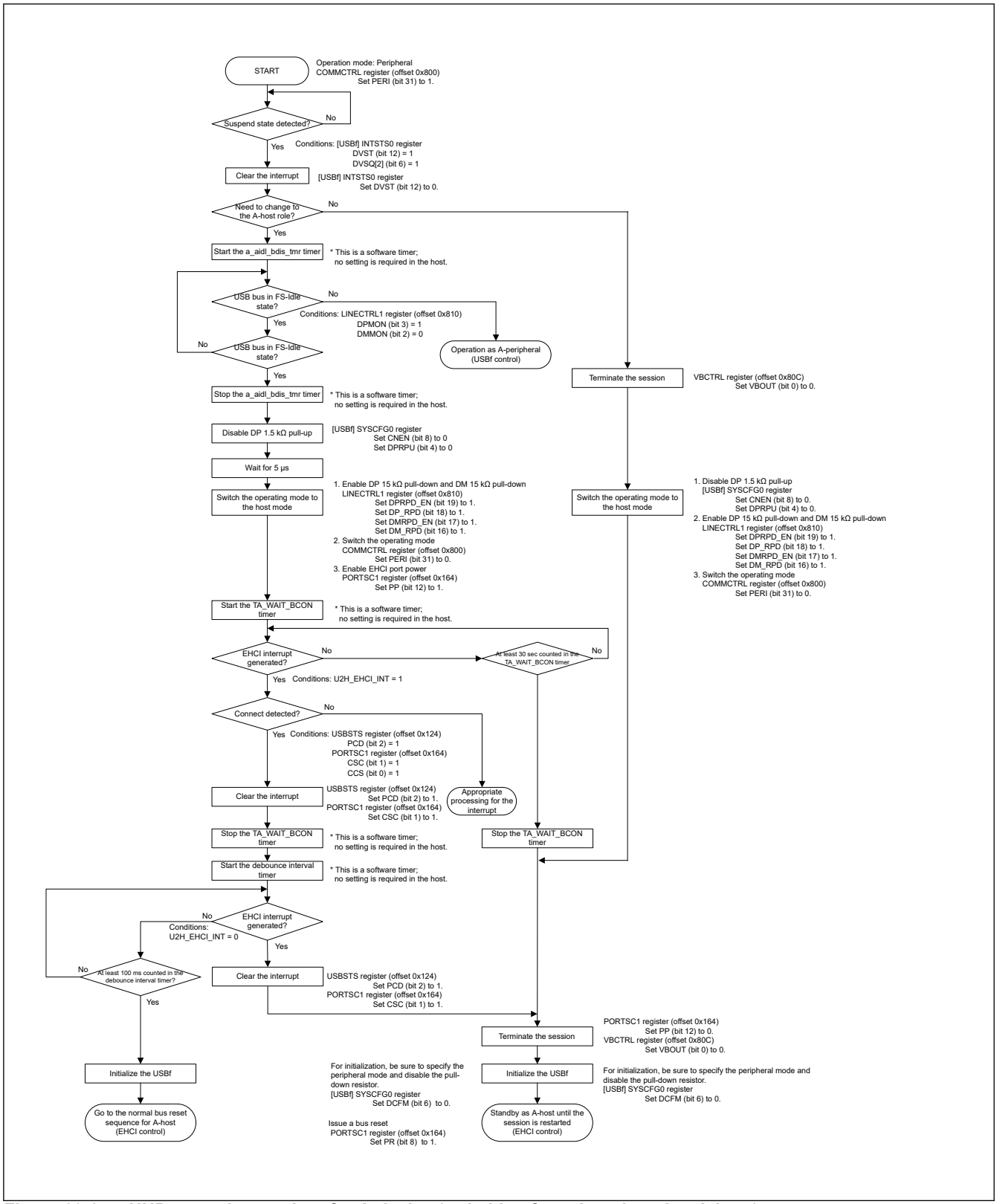


Figure 29.14 HNP control procedure for A-device (switching from A-periperal to A-host)

29.7.6.3 HNP Polling Procedure for A-Device

Figure 29.15 shows the procedure to check whether B-peripheral is requesting the host role when this LSI is an A-host.

For HNP polling in High-Speed mode (controlled by EHCI):

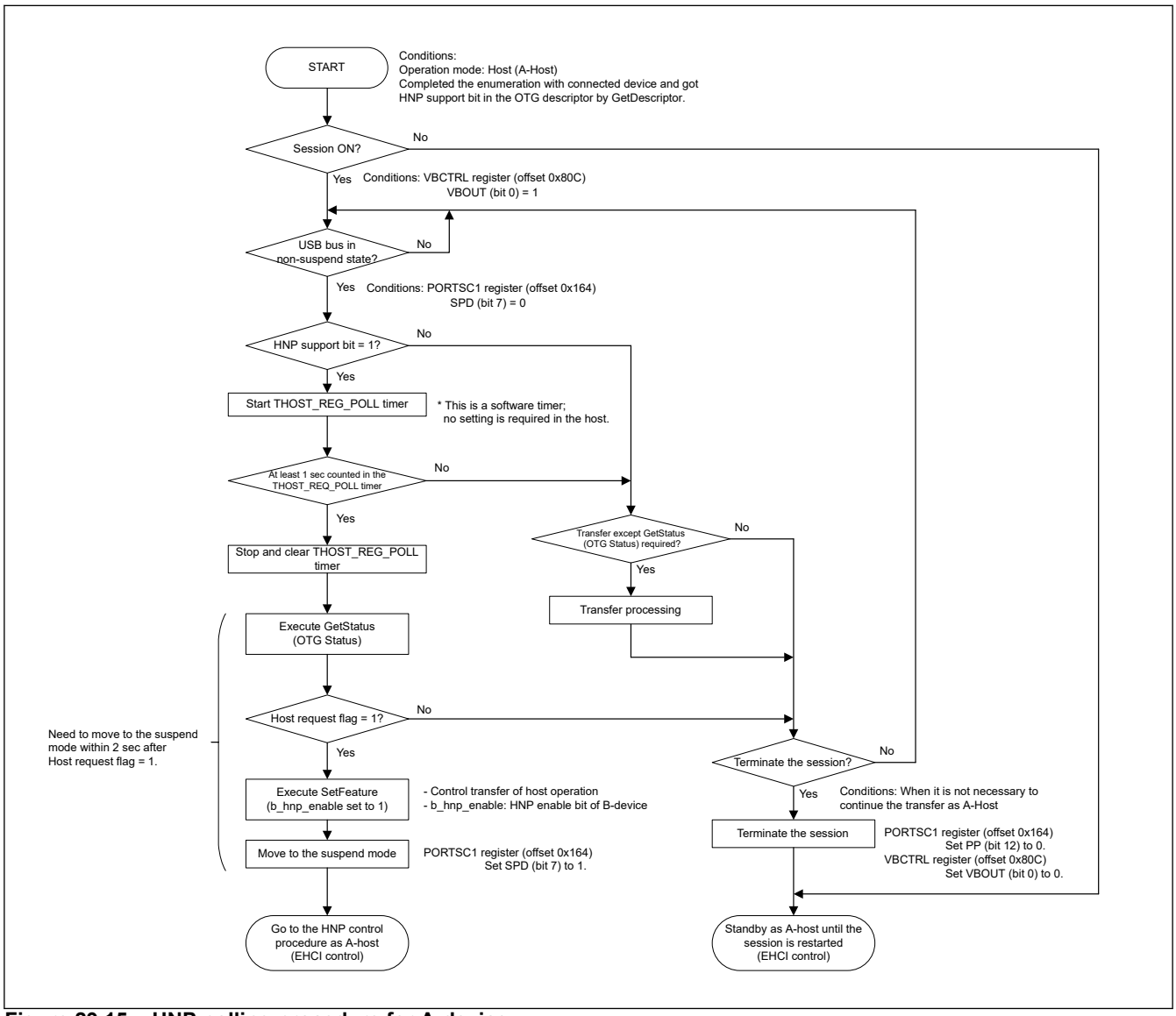


Figure 29.15 HNP polling procedure for A-device

For HNP polling in Full-Speed mode (controlled by OHCI):

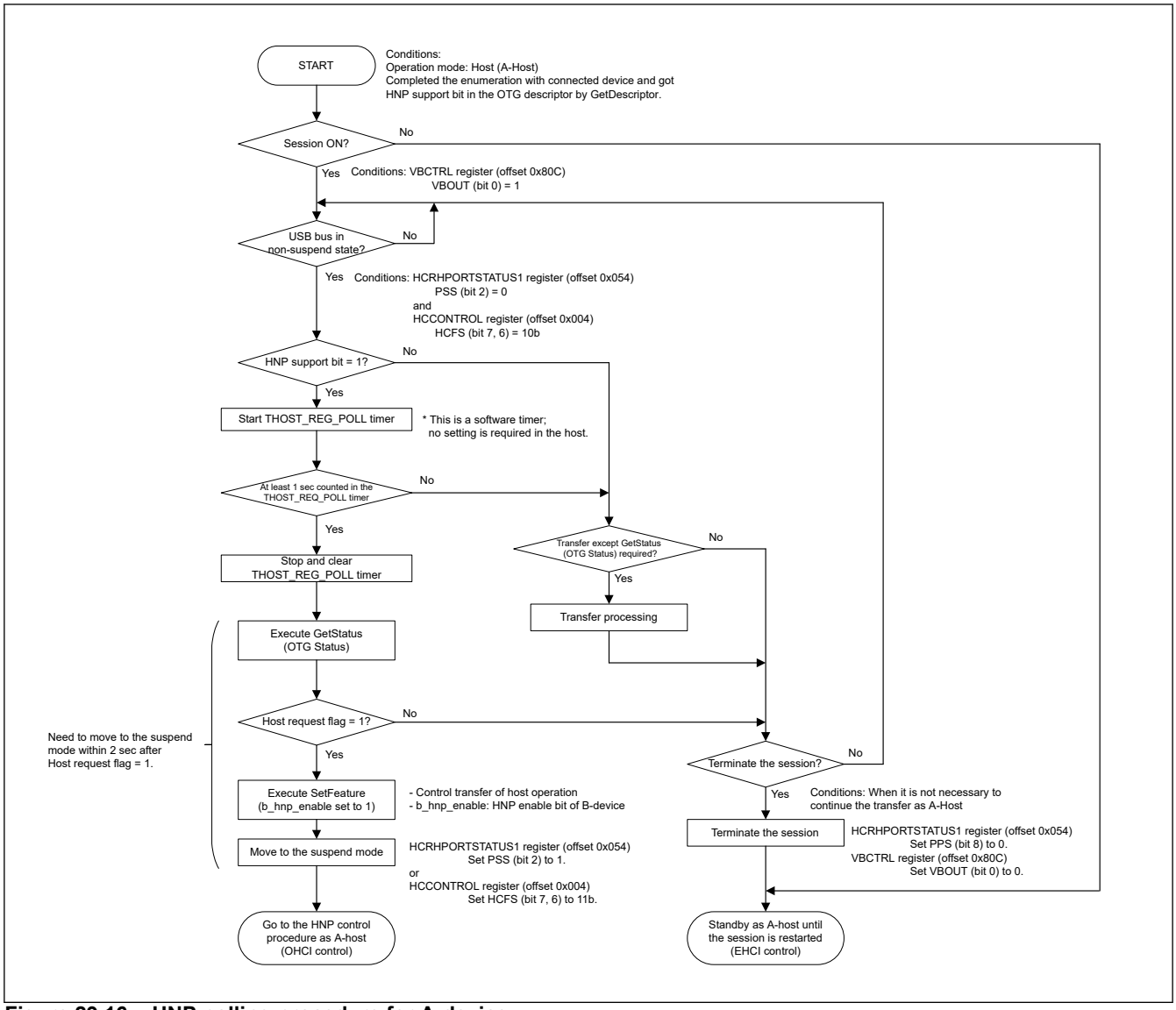


Figure 29.16 HNP polling procedure for A-device

29.7.6.4 HNP Control Procedure for B-Device (Switching from B-Peripheral to B-Host)

Figure 29.17 shows the procedure after the A-device issues set feature.

Assumed conditions:

Even for FS OTG, EHCI is assumed to be the port owner after switching the role from peripheral to host. Therefore, the procedure for switching from the B-peripheral to the B-host is not separately provided for each of OHCI and EHCI.

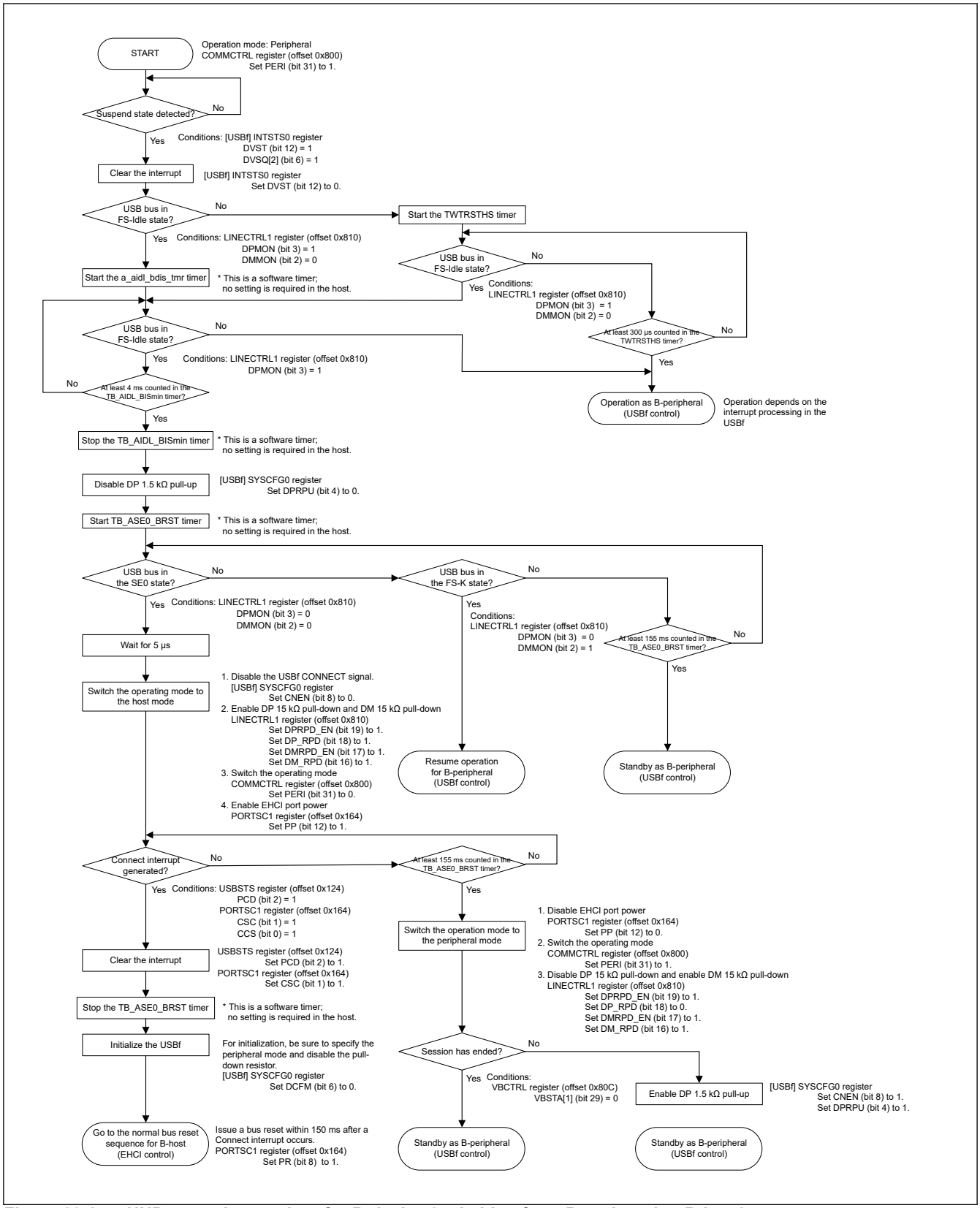


Figure 29.17 HNP control procedure for B-device (switching from B-periperal to B-host)

29.7.6.5 HNP Control Procedure for B-Device (Switching from B-Host to B-Peripheral)

Figure 29.18 shows the procedure after no transfer is performed by the B-host.

Assuming the control of ID and VBUS

VBUS control: VBCTRL.VBOUT

ID detection: Power-supply IC

For HNP control in High-Speed mode (controlled by EHCI):

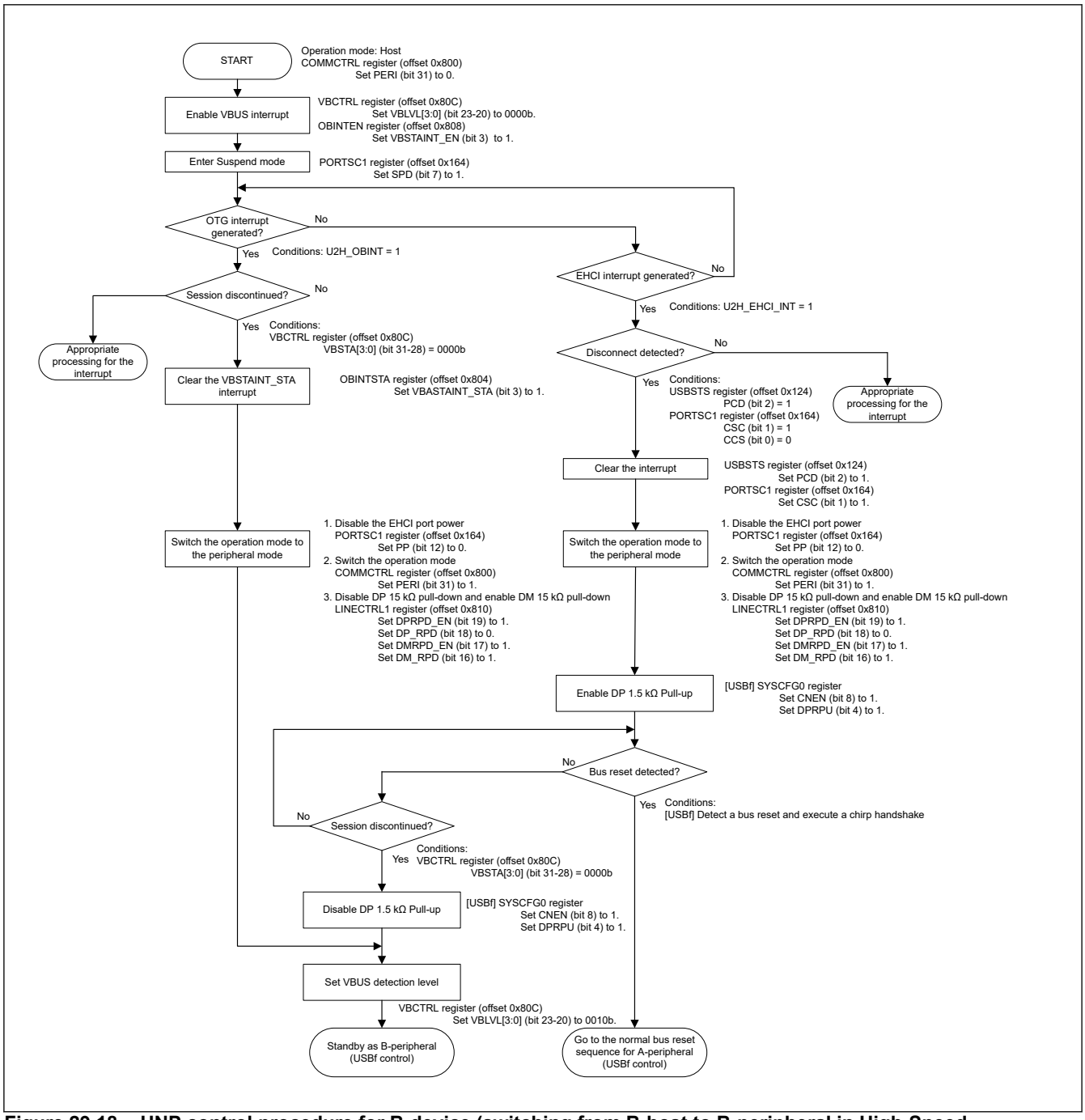


Figure 29.18 HNP control procedure for B-device (switching from B-host to B-peripheral in High-Speed mode)

For HNP control in Full-Speed mode (controlled by OHCI):

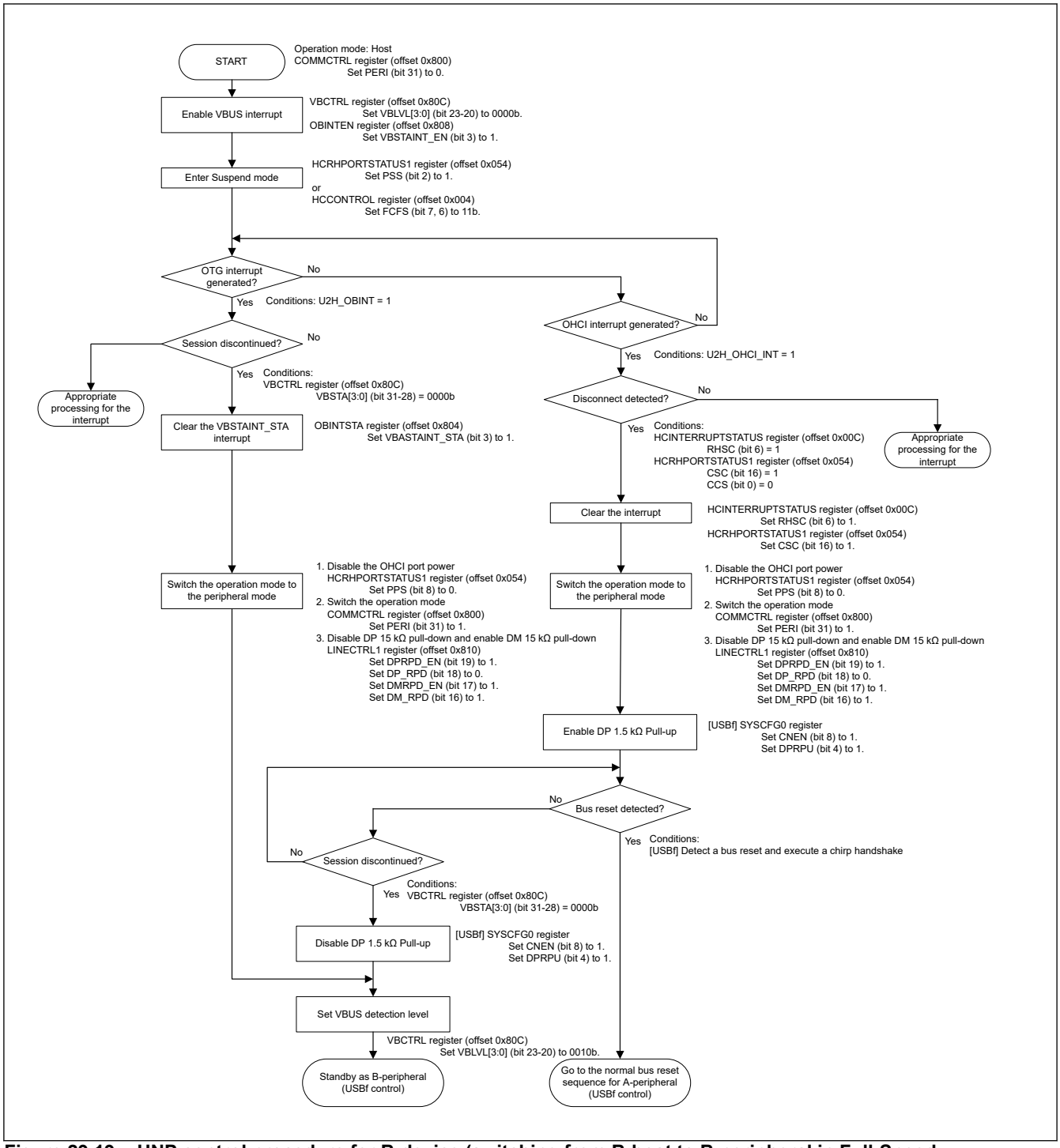


Figure 29.19 HNP control procedure for B-device (switching from B-host to B-peripheral in Full-Speed mode)

29.7.7 SRP Control Procedure (Conforming to OTG Revision 2.0)

29.7.7.1 SRP Control Procedure for A-Device

Figure 29.20 shows the procedure starting with the step for terminating a session because the B-device issues no request for the host role and the A-device does not need to transfer data.

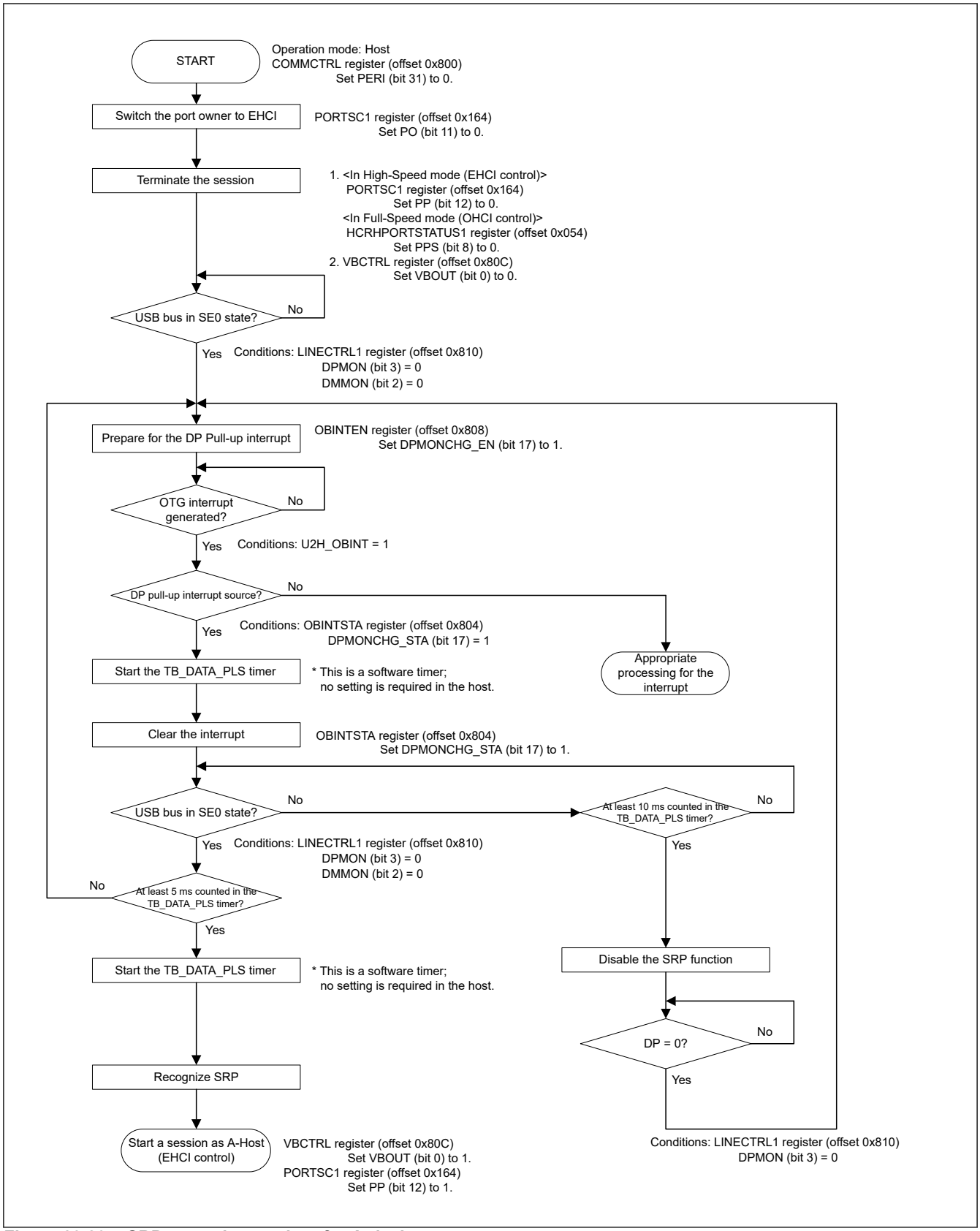


Figure 29.20 SRP control procedure for A-device

29.7.7.2 SRP Control Procedure for B-Device

Figure 29.21 shows the procedure from the standby state for a termination request of a session from the A-device.

Assuming the control of ID and VBUS

VBUS control: VBCTRL.VBOUT

ID detection: Power-supply IC

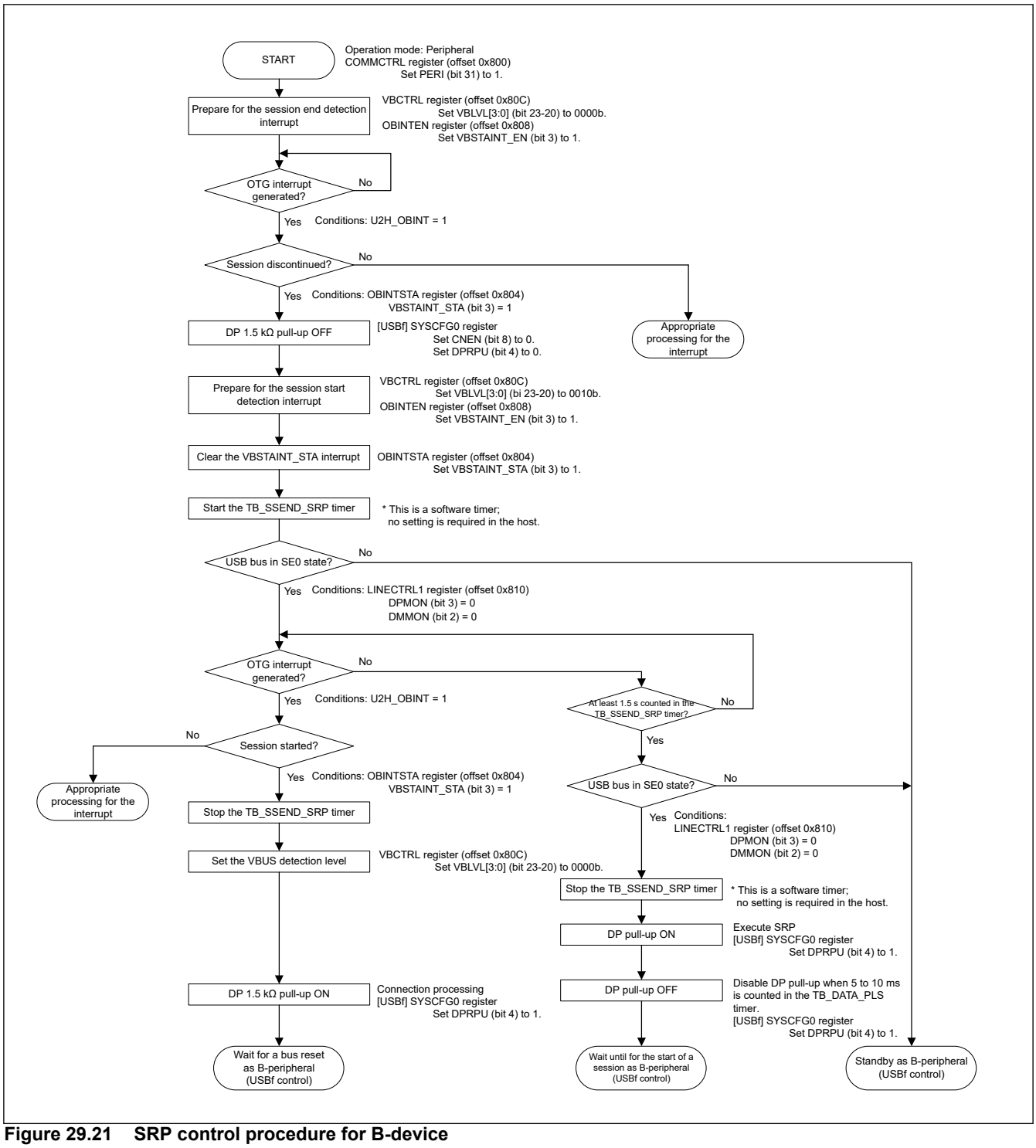


Figure 29.21 SRP control procedure for B-device

30. USB 2.0 HS Function Module (USBf)

30.1 Overview

The USB module of this LSI is a dual-role device that has USB 2.0 host and function modules.

This LSI has a single USB port for two functions for host and function. These functions can be switched by the common control register (COMMCTRL).

Since operation as a host controller or a function controller are exclusive of each other, dynamic switching between the types of operation is not possible.

This section describes function controller operation.

The USBf module supports high-speed transfer and full-speed transfer defined by the USB 2.0 specification.

This LSI supports all transfer types that are defined in the USB specification. It incorporates an 8 KB buffer memory for data transfer and can use up to 10 pipes. Pipes 1 to 9 can be assigned an arbitrary endpoint number according to the user system.

Table 30.1 Specifications of the USB Module

Item	Description
For High-Speed USB	• Built-in USB function controller
For all types of USB transfer	• Support for all types of USB transfer: – Control transfer – Bulk transfer – Interrupt transfer (high-bandwidth transfers not supported) – Isochronous transfer (high-bandwidth transfers not supported)
Bus interface	• Two DMA interfaces available – The dedicated AHB master interface for each DMA channel • High-speed data transfer for access to the internal FIFOs at 60 MB/second (when the bus width is 32 bits)
Pipe configuration	• 8 KB of buffer memory for USB communications • Up to 10 pipes can be selected (including the default control pipe) • Programmable pipe configuration • Any endpoint number can be assigned to pipes 1 to 9 • Transfer conditions that can be set for each pipe are listed below. Pipe 0: Control transfer, single buffer fixed at 64 bytes Pipes 1 and 2: Bulk transfer or isochronous transfer can be selected, continuous transfer mode, programmable buffer size (specifiable as up to 2 KB, double buffer is also specifiable) Pipes 3 to 5: Bulk transfer, continuous transfer mode, programmable buffer size (specifiable as up to 2 KB, double buffer is also specifiable) Pipes 6 to 9: Interrupt transfer, single buffer fixed at 64 bytes
Features of function controller operation	• High-speed (480 Mbps) and full-speed transfer (12 Mbps) are supported • Automatic recognition of high-speed operation or full-speed operation based on automatic response to the reset handshake • Control transfer stage monitoring • Monitoring of device state • Automatic response to SET_ADDRESS requests • NAK response interrupt function (NRDY) • SOF interpolation
Other functions	• Judging the end of transfer on the basis of a transaction count • Adjustable timing for BRDY interrupt event notification (BFRE) • Automatic clearing of the buffer memory after the data for the pipe specified in the DxFIFO port has been read (DCLRM) • NAK setting for response PID generated by end of transfer (SHTNAK)

Figure 30.1 is a block diagram of the USB module.

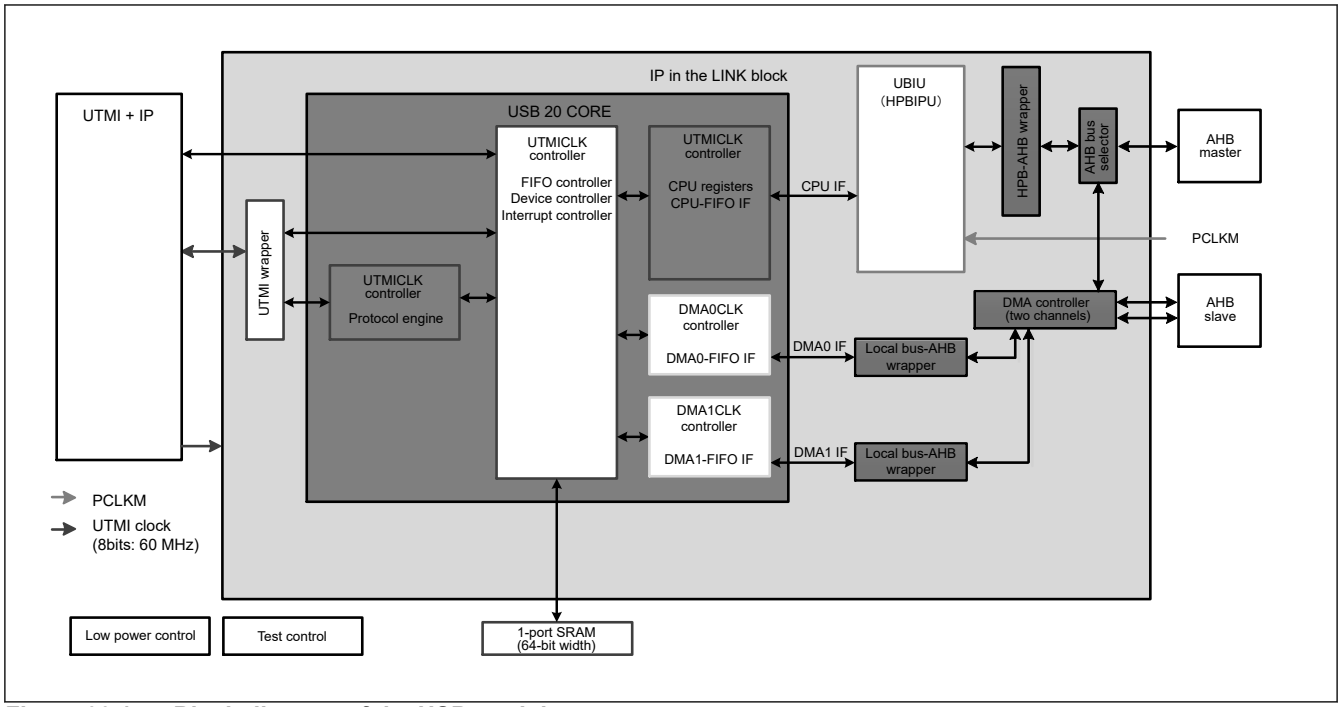


Figure 30.1 Block diagram of the USB module

Data transfer between this controller and the host controller connected to the USB bus uses buffer memory assigned to each pipe. For communication between this controller and the host controller, this controller converts the data stored in buffer memory into USB data packets and outputs them to the USB bus serially, and also inputs data packets from the USB bus and stores them in buffer memory.

30.1.1 Identification of the USB Transfer Speed

The hardware can automatically identify the USB transfer speed.

30.1.2 Bus Interface

(1) How to access the FIFO buffer memory

The following two types of access are available for the FIFO buffer memory for USB data transfers. To read from or write to the FIFO buffer memory, access (read or write) the FIFO ports from the CPU (DMAC).

Two channels of DMAC are provided.

1. CPU access
   Specify a FIFO port address and write data to the FIFO buffer memory or read data from the FIFO buffer memory.
2. DMA access
   Specify the dedicated DMAC and write data to the FIFO buffer memory or read data from the FIFO buffer memory.

USB data transfer is done in little endian. The byte endian swap function is available for FIFO port access; for 16 bit or 32 bit access, the endian can be switched through register settings.

30.1.3 USB Event

This controller sends an interrupt to the ICU to notify an event in USB operation.

Sending interrupts for notification can be enabled or disabled separately for each interrupt type and source through software settings.

30.1.4 USB Data Transfer

This controller performs all types of USB data transfer: control transfer, bulk transfer, interrupt transfer, and isochronous transfer. The following numbers of pipes are available for each transfer type.

- One pipe dedicated to control transfer

- Four pipes dedicated to interrupt transfer
- Three pipes dedicated to bulk transfer
- Two pipes selectively used for bulk transfer or isochronous transfer

Each pipe should be set up as necessary for the desired USB transfer in accordance with the user system; for example, transfer type, endpoint number, or maximum packet size.

This controller has an 8 KB buffer memory. For pipes dedicated for bulk transfer or selective pipes for bulk transfer or isochronous transfer, buffer memory assignment, buffer operating mode setting, or other necessary settings should be made in accordance with the user system. By setting the buffer operating mode, such as double buffer structure or continuous data packet transfer mode, fast data transfer is achieved with fewer interrupts.

CPU and DMA controller access to the buffer memory is made through three FIFO port registers.

30.1.5 DMA Transfer

This controller provides two channels of the DMA interface having the following functions.

- DMA transaction mode: Fetching in both register and link modes are supported
- Transfer size: A transfer size from 1 to 128 bytes can be selected separately for the transfer source and transfer destination
- Skip (scatter/gather) function: The access size and skip size can be specified separately for the transfer source and destination
- Suspend function: A running DMA transaction can be suspended temporarily
- Interval function: The interval of DMA transfers can be specified to control the bus occupancy

30.2 Register Descriptions

30.2.1 System Configuration Control

30.2.1.1 SYSCFG0 : System Configuration Control Register 0

Base address: USBF = 0x8020_1000

Offset address: 0x00

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CNEN	HSE	—	DRPD	DPRPU	—	—	—	USBE
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Bit	Symbol	Function	R/W
0	USBE	USB Block Operation Enable Enables or disables operation of the USB block. 0: USB block operation is disabled. 1: USB block operation is enabled.	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	DPRPU	D+ Line Resistor Control Enables or disables pulling up the D+ line in function controller operation. 0: Pulling up the line is disabled. 1: Pulling up the line is enabled.	R/W
5	DRPD	D+/D− Line Resistor Control Always set 0 at initialization.	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	HSE	High-Speed Operation Enable Enables or disables high-speed operation. 0: High-speed operation is disabled (full-speed). 1: High-speed operation is enabled (the controller detects the communication speed)	R/W

Bit	Symbol	Function	R/W
8	CNEN	Single-End Receiver Operation Enable Enables or disables the single-end receiver to operate. 0: Single-end receiver operation is disabled. 1: Single-end receiver operation is enabled.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

Writing to this register is possible even while the UTMI clock (output of the clock signal from the PLL in the USB) is stopped. Note however that any values that are set while the UTMI clock is stopped are reflected after oscillation of the UTMI clock is restarted.

USBE bit (USB Block Operation Enable)

The setting of the USBE bit enables or disables operation of the USB block.

When the USBE bit is modified from 1 to 0, this controller initializes the bits listed in Table 30.2.

Table 30.2 Registers initialized by writing USBE = 0

Register Name	Bit Name
SYSSTS0	LNST
DVSTCTR0	RHST
INTSTS0	DVSQ
USBADDR	USBADDR
USBREQ	bRequest bmRequestType
USBVAL	wValue
USBINDX	wIndex
USBLENG	wLength

The value of this bit must be changed when SUSPM = 1 and after the UTMI clock oscillation is started.

DRPD and DPRPU bit (D+/D− Line Resistor Control)

Settings related to USB data bus resistance are given in Table 30.3, USB Data Bus Resistance Control. Use the DRPD and DPRPU bits to select USB data bus resistance.

Table 30.3 USB data bus resistance control

Settings		USB data bus resistance control		
DRPD	DPRPU	D− Line	D+ Line	Remarks
0	0	Open	Open	
0	1	Open	Pull-up	Make these settings when the module is used as a function controller.
1	0	Pull-down	Pull-down	Setting prohibited
1	1	Pull-down	Pull-up	Setting prohibited

D+ pull-up resistance control for function controller operation (DPRPU)

When this bit set to 1, this controller pulls up the D+ line to 3.3V and can notify the USB host that the function module is attached.

The controller cancels pulling-up of the D+ line by modifying this bit from 1 to 0, and the state for the USB host can be shown as detached.

HSE bit (High-Speed Operation Enable)

Setting this bit to 1 enables high-speed operation. Setting the HSE bit to 1 lets this controller perform high-speed or fullspeed operation based on the results of reset handshake.

Setting the HSE bit to 0 lets this controller perform full-speed operation.

On the other hand, setting the HSE bit to 1 lets this controller perform the reset handshake protocol and then perform high-speed or full-speed operation automatically according to the results.

This bit can be modified when DPRPU = 0.

CNEN bit (Single-End Receiver Operation Enable)

Setting this bit to 1 enables the single-end receiver to operate. This bit is used to prevent inrush current damage due to the floating state of the single-end receiver when it is detached. Furthermore, this bit makes it possible to see the LNST bits.

Set this bit to 1 when VBUS is detected by a VBUS interrupt, and clear this bit to 0 when VBUS is removed.

30.2.1.2 SYSCFG1 : System Configuration Control Register 1

Base address: USBF = 0x8020_1000

Offset address: 0x02

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	AWAIT[5:0]						—	—	BWAIT[5:0]					
Value after reset:	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1

Bit	Symbol	Function	R/W
5:0	BWAIT[5:0]	CPU Bus Access Wait Specification Specify the number of wait cycles for access to this controller. 0x00: 0 wait cycles (2 access cycles) ⋮ 0x02: 2 wait cycles (4 access cycles) ⋮ 0x04: 4 wait cycles (6 access cycles) ⋮ 0x0F: 15 wait cycles (17 access cycles) (value after reset) ⋮ 0x3F: 63 wait cycles (65 access cycles)	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
13:8	AWAIT[5:0]	AHB-DMA Bridge Bus Access Wait Specification Specify the number of wait cycles for access to this controller. 0x00: 0 wait cycles (2 access cycles) ⋮ 0x02: 2 wait cycles (4 access cycles) ⋮ 0x04: 4 wait cycles (6 access cycles) ⋮ 0x0F: 15 wait cycles (17 access cycles) (value after reset) ⋮ 0x3F: 63 wait cycles (65 access cycles)	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

BWAIT[5:0] bits (CPU Bus Access Wait Specification)

For continuous access to the registers at addresses of this controller beginning at 0x80201004, at least 67 ns must be secured.

To satisfy this restriction, control wait cycles using the frequency of CPU clock. The value after a reset is the maximum value (17 clock cycles), so select an appropriate value no greater than this.

This setting is the same as that for waiting in access to a FIFO port register. The maximum speeds of access to the FIFO ports are as follows:

MBW = 10b (32 bit width): Max 60 MB/sec

MBW = 01b (16 bit width): Max 30 MB/sec

MBW = 00b (8 bit width): Max 15 MB/sec

AWAIT[5:0] bits (AHB-DMA Bridge Bus Access Wait Specification)

These bits specify the number of wait cycles for access to the AHB-DFIFO0/AHB-DFIFO1 bus.

Restrictions for access cycles are the same as those for the CPU bus access wait specification bits (BWAIT).

30.2.1.3 SYSSTS0 : System Configuration Status Register

Base address: USBF = 0x8020_1000

Offset address: 0x04

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LNST[1:0]	
Value after reset:	x	x	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	LNST[1:0]	USB Data Line Status Monitor Indicate the state of the USB line.	R
15:2	—	These bits are read as 0.	R

Note: When a USB bus reset is detected, this register value is undefined.

LNST[1:0] bits (USB Data Line Status Monitor)

The states of USB data bus line of this controller is shown in Table 30.4. The controller monitors the states of USB data bus lines (D+ and D− lines) in the LNST bits of the SYSSTS0 register.

Reference to the LNST bits must be made following attach processing (DPRPU = 1) after USBE has been set to 1.

Table 30.4 States of USB data bus line

LNST[1]	LNST[0]	Full-speed operations	High-speed operations	Chirp operations*1
0	0	SE0	Squelch*2	Squelch*2
0	1	J state	Unsquelch*3	Chirp J*4
1	0	K state	Invalid	Chirp K*5
1	1	SE1	Invalid	Invalid

Note 1. Chirp: Reset handshake protocol being executed in high-speed operations enabled state (HSE = 1)
Note 2. Squelch: SE0 or idle state
Note 3. Unsquelch: High-speed J state or high-speed K state
Note 4. Chirp J: Chirp J-state
Note 5. Chirp K: Chirp K-state

30.2.2 USB Signal Control

30.2.2.1 DVSTCTR0 : Device State Control Register 0

Base address: USBF = 0x8020_1000

Offset address: 0x08

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	WKUP	—	—	—	—	—	RHST[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	RHST[2:0]	Reset Handshake Indicate the state of reset handshake.	R
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	WKUP	Wakeup Output Enables or disables the remote wakeup (resume signal output). Only 1 can be written to this bit. 0: Remote wakeup signal is not output. 1: Remote wakeup signal is output.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, The WKUP is set to 0 and other bits are undefined.

RHST[2:0] bits (Reset Handshake)

These bits indicate the result of reset handshake. Table 30.5 lists the results of reset handshake.

Table 30.5 State of Reset Handshake

Bus state	RHST bit value
Powered or no connection	000b
Reset handshake in progress	100b
Full-speed connection	010b
High-speed connection	011b

When the HSE bit is set to 1, the RHST bits indicate 100b when this controller detects a USB bus reset. Then, these bits indicate 011b when this controller outputs Chirp-K and detects Chirp-JK from the USB host three times. If the connection speed is not fixed to high speed within 2.5 ms after Chirp-K output, these bits indicate 010b.

When the HSE bit is set to 0, these bits indicate 010b when the controller detects a USB bus reset.

A DVST interrupt is generated when the value of the RHST bits is 010b or 011b after the USB bus reset is detected by the controller.

WKUP bit (Wakeup Output)

With the WKUP bit set to 1, this controller outputs the remote wakeup signal to the USB bus.

This controller controls the output time of a remote wakeup signal. When this bit is set to 1 by software, the controller clears this bit to 0 after outputting the 10 ms K-state.

According to the USB Specifications, the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is sent. If this controller writes 1 to this bit right after detection of the suspended state, the K-state will be output after 2 ms.

Do not write 1 to this bit unless the device is in the suspended state (bits DVSQ = 1xxb) and the USB host enables the remote wakeup signal.

When this bit is set to 1, the internal clock must not be stopped even in the suspended state (write 1 to this bit while the SUSPM bit = 1).

30.2.3 Test Mode

30.2.3.1 TESTMODE : USB Test Mode Register

Base address: USBF = 0x8020_1000

Offset address: 0x0C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	UTST[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	UTST[3:0]	Test Mode *See the detailed description below.	R/W

Bit	Symbol	Function	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

UTST[3:0] bits (Test Mode)

Writing values to these bits allows this controller to output a USB test signal during high-speed operations. Table 30.6 shows the test mode operations of the controller.

Table 30.6 Test mode operations

Test mode	UTST settings
Normal operation	0000b
Test_J	0001b
Test_K	0010b
Test_SE0_NAK	0011b
Test_Packet	0100b
Test_Force_Enable	—
Reserved	0101b to 0111b

Write to these bits according to the SetFeature request from the USB host during high-speed communication.

This controller does not enter the suspended state while these bits are 0001b to 0100b.

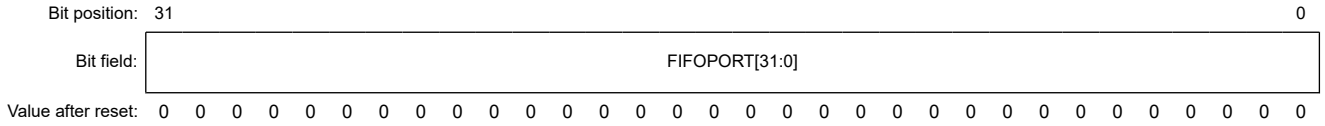
To perform normal USB communications after the test mode is set, apply a hardware reset.

30.2.4 FIFO Ports

30.2.4.1 CFIFO, DnFIFO : FIFO Port Register (n = 0, 1)

Base address: USBF = 0x8020_1000

Offset address: 0x014 (CFIFO/CFIFOL/CFIFOLL)
0x016 (CFIFOH)
0x017 (CFIFOHH)
0x018 + 0x4 × n (DnFIFO/DnFIFOL/DnFIFOLL)
0x01A + 0x4 × n (DnFIFOH)
0x01B + 0x4 × n (DnFIFOHH)



Bit	Symbol	Function	R/W
31:0	FIFOPORT[31:0]	FIFO Port Accessing these bits allows reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

The transmission/reception buffer of this controller is made up of a FIFO structure (FIFO buffer). Use the FIFO port register for access to the FIFO buffer. There are three FIFO ports: CFIFO, D0FIFO, and D1FIFO ports. Also, there is a DxFIFO continuous transfer port for continuous transfer. Each FIFO port consists of a port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer memory and writing of data to the FIFO buffer memory, a port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that is used to select the pipe assigned to the FIFO port, and a port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR).

The FIFO ports have the following restrictions.

1. Access to an FIFO buffer for DCP should be through the CFIFO port.

2. Access to an FIFO buffer by DMA transfer when DFACC = 00b (cycle stealing mode) should be through the DxFIFO port.
3. Access to an FIFO buffer by DMA transfer when DFACC = 01b or 10b (16 byte or 32 byte continuous access mode) should be through the DxFIFO continuous transfer port.
4. The DxFIFO ports can also be accessed by the CPU.
5. When using functions specific to the FIFO port, the pipe number (selected pipe) specified by the CURPIPE bits cannot be changed (when the DMA transfer function is used, etc.).
6. Registers containing the FIFO port do not affect the other FIFO ports.
7. Do not assign the same pipe to separate FIFO ports.
8. In the FIFO buffer state, there are two types of access rights: one assigned to the CPU, and the other to SIE. Access from the CPU is not possible when SIE has the rights to access the FIFO buffer.

FIFOPORT[31:0] bits (FIFO Port)

This controller accesses the FIFO buffer assigned to the pipe number written to the CURPIPE bits of the corresponding select registers (CFIFOSEL, D0FIFOSEL, and D1FIFOSEL) through access to any of these registers by software.

This register can only be accessed when the FRDY bit of each control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) indicates 1 (or while DnFIFOSEL.DREQE = 1).

The valid bits of this register depend on the settings of the corresponding MBW and BIGEND bits.

Table 30.7 to Table 30.9 list the valid bits.

Table 30.7 Endian operation in 32 bit access (MBW = 10)

BIGEND	b31 to b24	b23 to b16	b15 to b8	b7 to b0
0	N + 3 address	N + 2 address	N + 1 address	N + 0 address
1	N + 0 address	N + 1 address	N + 2 address	N + 3 address

Table 30.8 Endian operation in 16 bit access (MBW = 01)

BIGEND	b31 to b24	b23 to b16	b15 to b8	b7 to b0
0	Writing: Invalid Reading: Prohibited*1		N + 1 address	N + 0 address
1	N + 0 address	N + 1 address	Writing: Invalid Reading: Prohibited*1	

Note 1. Reading data from the invalid bits in word or byte units is prohibited.

Table 30.9 Endian operation in 8 bit Access (MBW = 00)

BIGEND	b31 to b24	b23 to b16	b15 to b8	b7 to b0
0	Writing: Invalid Reading: Invalid*1			N + 0 address
1	N + 0 address	Writing: Invalid Reading: Prohibited*1		

Note 1. Reading data from the invalid bits in word or byte units is prohibited.

30.2.4.2 CFIFOSEL : CFIFO Port Select Register

Base address: USBF = 0x8020_1000

Offset address: 0x20

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RCNT	REW	—	—	MBW[1:0]		—	BIGEND	—	—	ISEL	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CURPIPE[3:0]	FIFO Port Access Pipe Specification Specify the pipe number to access the CFIFO port. 0x0: DCP 0x1: Pipe 1 0x2: Pipe 2 ⋮ 0x8: Pipe 8 0x9: Pipe 9 Others: Setting prohibited	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	ISEL	FIFO Port Access Direction when DCP is Selected Specifies access direction of the FIFO port when DCP is selected in CURPIPE bits. 0: Reading from the buffer memory 1: Writing to the buffer memory	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	BIGEND	FIFO Port Endian Control Specifies the CFIFO port byte endian. 0: Little endian 1: Big endian	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
11:10	MBW[1:0]	CFIFO Port Access Bit Width Specify the bit width for accessing the CFIFO port. 0 0: 8 bit width 0 1: 16 bit width 1 0: 32 bit width 1 1: Setting prohibited	R/W
13:12	—	These bits are read as 0. The write value should be 0.	R/W
14	REW	Buffer Pointer Rewind Specifies 1 when rewinding the buffer pointer. This bit is read as 0. 0: The buffer pointer is not rewound. 1: The buffer pointer is rewound.	R/W
15	RCNT	Read Count Mode Specifies the read mode for the value in the DTLN bits in CFIFOCTR. 0: The DTLN bits are cleared when all of the received data has been read from the CFIFO. 1: The DTLN bits are decremented each time the received data is read from the CFIFO.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

CURPIPE[3:0] bits (FIFO Port Access Pipe Specification)

Write the pipe number for the data to be read or written through the CFIFO port.

When modifying this bit field, write this bit first and then read this bit. Only when the written value matches the read one, proceed to the next process.

Do not specify the same pipe in the CURPIPE bits of the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.

Even if the setting of these bits is modified during access to the FIFO buffer, the state of the FIFO buffer is retained, with continued access proceeding after these bits are re-set to the value before the modification.

ISEL bit (FIFO Port Access Direction when DCP is Selected)

To change this bit when the specified pipe is DCP, first write the data to this bit and then read it. Proceed to the next process after checking if the written value matches with the read value.

When the settings of this bit are modified during access to the FIFO buffer, access up to then is saved. Access to the buffer can be continued after rewriting the settings.

Settings in this bit should be made at the same as that of the CURPIPE bits.

BIGEND bit (FIFO Port Endian Control)

In this bit, set the byte endian for the CFIFO port.

For details, see the description of FIFO Port Bits (CFIFO, D0FIFO, and D1FIFO).

MBW[1:0] bits (CFIFO Port Access Bit Width)

In this bit field, set the bit width for accessing the CFIFO port.

When the pipe specified in the CURPIPE bits is in the receiving direction, if reading is started after setting this bit field, do not modify the value of the MBW bits until all data is read.

When the specified pipe is in the receiving direction, modify the value of the CURPIPE bit once and then set the CURPIPE and MBW bits at the same time.

For the procedure for changing the CURPIPE bit, see the description for the same.

When the specified pipe is in the transmitting direction, to switch the bit width from 8 bits to 16 or 32 bits or from 16 bits to 32 bits is not allowed while writing to the buffer memory is in progress.

With the 16 bit or 32 bit width setting, writing to odd bytes is possible by exercising byte access control.

REW bit (Buffer Pointer Rewind)

When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double-buffer operation, re-reading the currently-read FIFO buffer plane from the first data is allowed).

Do not set REW to 1 simultaneously with modifying the CURPIPE bits. Before setting REW to 1, be sure to check that FRDY is 1.

To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

RCNT bit (Read Count Mode)

When 0 is written to this bit, if all received data of the FIFO buffer assigned to the pipe specified in the CURPIPE bits (selected pipe) is read (when the data is read on one side of a double buffer), this controller clears the DTLN bits in the CFIFOCTR register to 0.

When 1 is written to this bit, the controller decrements the value of the DTLN bits in the CFIFOCTR register whenever the received data is read from the FIFO buffer assigned to the specified pipe.

30.2.4.3 DnFIFOSEL : DnFIFO Port Select Register (n = 0, 1)

Base address: USBF = 0x8020_1000

Offset address: 0x028 + 0x4 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RCNT	REW	DCLRM	DREQE	MBW[1:0]		—	BIGEND	—	—	—	—	CURPIPE[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CURPIPE[3:0]	FIFO Port Access Pipe Specification 0x0: Not specified 0x1: Pipe 1 0x2: Pipe 2 ⋮ 0x8: Pipe 8 0x9: Pipe 9 Others: Setting prohibited	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	BIGEND	FIFO Port Endian Control Specifies the DnFIFO port byte endian. 0: Little endian 1: Big endian	R/W

Bit	Symbol	Function	R/W
9	—	This bit is read as 0. The write value should be 0.	R/W
11:10	MBW[1:0]	FIFO Port Access Bit Width Specify the bit width for accessing the FIFO port. 0 0: 8 bit width 0 1: 16 bit width 1 0: 32 bit width 1 1: Setting prohibited	R/W
12	DREQE	DMA Request Enable This bit is used to notify for DMAC. 0: DMA request is disabled 1: DMA request is enabled	R/W
13	DCLRM	Auto Buffer Memory Clear Mode after Specified Pipe Data is Read Enables or disables automatic clearing of the buffer memory after data has been read out using the selected pipe. 0: Auto buffer clear mode is disabled 1: Auto buffer clear mode is enabled	R/W
14	REW	Buffer Pointer Rewind Specifies 1 when rewinding the buffer pointer. This bit is read as 0. 0: The buffer pointer is not rewound 1: The buffer pointer is rewound	R/W
15	RCNT	Read Count Mode Specifies the read mode for the value in the DTLN bits in Dx_FIFOCTR. 0: The DTLN bits are cleared when all of the received data has been read. 1: The DTLN bits are decremented each time the received data is read.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

CURPIPE[3:0] bits (FIFO Port Access Pipe Specification)

Write the pipe number for the data to be read or written through the DxFIFO port.

When modifying this bit field, first write a value to this bit field and then read it. Check if the write value matches the read value and then proceed to the next process.

Do not specify the same pipe in the CURPIPE bits of the CFIFOSEL, D0FIFOSEL, and D1FIFOSEL registers.

Even if the setting of these bits is modified during access to the FIFO buffer, the state of the FIFO buffer is retained, with continued access proceeding after these bits are re-set to the value before the modification.

BIGEND bit (FIFO Port Endian Control)

Write the byte endian for the DxFIFO port in this bit.

For details, see the description of FIFO Port Bits (CFIFO, D0FIFO, and D1FIFO).

MBW[1:0] bits (FIFO Port Access Bit Width)

Write the bit width for accessing the DxFIFO port in this bit.

When the pipe specified in the CURPIPE bits is in the receiving direction, if reading is started after setting this bit field, do not modify the value of the MBW bits until all data is read.

When the specified pipe is in the receiving direction, set the CURPIPE and MBW bits at the same time.

For the procedure for changing the CURPIPE bit, see the description for the same.

For details, see the description of CURPIPE[3:0] Bits (FIFO Port Access Pipe Specification Bits).

DREQE bit (DMA Request Enable)

Use this bit to enable or disable DMA request.

When enabling DMA request, set this bit to 1 after setting the CURPIPE bits.

When modifying the setting of the CURPIPE bits, do so after setting this bit to 0.

DCLRM bit (Auto Buffer Memory Clear Mode after Specified Pipe Data is Read)

After reading data in the specified pipe, disable or enable automatic clearing of the FIFO buffer. When 1 is written to this bit, this controller sets the BCLR bit to 1 for a FIFO buffer if a zero-length packet is received when the FIFO buffer assigned to the specified pipe is empty, or when a short packet is received and the data is completely read while BFRE = 1.

When using this controller with the BRDYM bit set to 1, be sure to set this bit to 0.

REW bit (Buffer Pointer Rewind)

When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double-buffer operation, re-reading the currently-read FIFO buffer plane from the first data is allowed).

Do not set the REW bit to 1 simultaneously with modifying the CURPIPE bits. Before setting the REW bit to 1, be sure to check that the FRDY bit is 1.

To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

RCNT bit (Read Count Mode)

When 0 is written to this bit, if all received data of the FIFO buffer assigned to the pipe specified in the CURPIPE bits (selected pipe) is read (when the data is read on one side of a double buffer), this controller clears the DTLN bits in the DxFIFOCTR register to 0.

When 1 is written to this bit, the controller decrements the value of the DTLN bits in the DxFIFOCTR register whenever the received data is read from the FIFO buffer assigned to the specified pipe.

30.2.4.4 CFIFOCTR, DnFIFOCTR : FIFO Port Control Register (n = 0, 1)

Base address: USBF = 0x8020_1000

Offset address: 0x022 (CFIFOCTR)
0x02A + 0x4 × n (DnFIFOCTR)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BVAL	BCLR	FRDY	—	DTLN[11:0]											
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	DTLN[11:0]	Receive Data Length Indicate the length of the receive data.	R
12	—	This bit is read as 0. The write value should be 0.	R/W
13	FRDY	FIFO Port Ready Indicates whether the FIFO port can be accessed. 0: FIFO port is not accessible. 1: FIFO port is accessible.	R
14	BCLR	CPU Buffer Clear Specifies 1 to clear the FIFO buffer on the CPU side for the selected pipe. Only 1 can be written to this bit. This bit is read as 0. 0: Invalid 1: Clears the CPU buffer memory on the CPU side.	R/W
15	BVAL	Buffer Memory Valid Flag Specifies 1 when writing has ended in the CPU-side FIFO buffer for the pipe specified in CURPIPE (selected pipe). Only 1 can be written to this bit. 0: Invalid 1: Writing ended	R/W

Note: When a USB bus reset is detected, this register value is undefined.

DTLN[11:0] bits (Receive Data Length)

The DTLN bits indicate the length of the receive data. While the FIFO buffer is being read, the DTLN bits indicate different values depending on the RCNT bit value as described below.

1. When RCNT = 0:
   The length of received data is set in these bits, and the value is retained until all received data has been read from a single FIFO buffer plane.
   While BFRE is 1, these bits retain the length of the receive data until BCLR is set to 1 even after all data has been read.
2. When RCNT = 1:
   This controller decrements the value indicated by these bits each time data is read from the FIFO buffer.
   (The value is decremented by 1 when MBW is 00b, by 2 when MBW is 01b, and by 4 when MBW is 10b.)
   This controller sets these bits to 0 when all data has been read from one FIFO buffer plane. However, in double-buffer operation, if data has been received in one FIFO buffer plane before all data has been read from the other plane, this controller sets these bits to indicate the length of the receive data in the latter plane when all data has been read from the former plane.
   When reading these bits during FIFO buffer reading while RCNT = 1, note that these bits are updated within 150 ns after a read cycle for the FIFO port.

FRDY bit (FIFO Port Ready)

This bit indicates whether the FIFO port can be accessed from the CPU (DMAC).

In the following cases, the controller sets FRDY to 1, but data cannot be read via the FIFO port because there is no data to be read. In these cases, set BCLR to 1, clear the FIFO buffer, and then enable transmission and reception of the next data.

1. A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty.
2. A short packet is received and the data is completely read while BFRE is 1.

BCLR bit (CPU Buffer Clear)

When this bit is set to 1, this controller clears the FIFO buffer on the CPU for the selected pipe.

When double-buffer operation is set for the FIFO buffer assigned to the selected pipe, this controller clears only one plane of the FIFO buffer even when both planes are read-enabled.

When the selected pipe is the DCP, setting BCLR to 1 allows this controller to clear the FIFO buffer regardless of whether the FIFO buffer is on the CPU side or SIE side. When clearing the buffer on the SIE side, set the PID bits for the DCP to NAK before setting BCLR to 1.

When the selected pipe is not the DCP, writing 1 to this bit should be done while FRDY indicates 1. When checking the FRDY bit after executing BCLR, allow an interval of at least 80 ns after executing BCLR before referencing FRDY.

BVAL bit (Buffer Memory Valid Flag)

When the pipe specified in the CURPIPE bits (selected pipe) is for transmission, write 1 to this bit in the following cases. This controller sets the FIFO buffer from the CPU side to the SIE side, enabling transmission.

1. To transmit the short packet, set this bit to 1 after data has been written.
2. To transmit a zero-length packet, set this bit to 1 before writing data to FIFO.
3. Set this bit to 1 after the number of data bytes has been written for the pipe in continuous transfer mode, where the number is a natural integral multiple of the maximum packet size and less than the buffer size.

When the data of the maximum packet size has been written for the pipe in non-continuous transfer mode, this controller sets this bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, enabling transmission.

When 1 is written to the BVAL and BCLR bits at the same time when the specified pipe is in the transmitting direction, the controller clears the old data and enables the transmission of a zero-length packet.

Writing 1 to this bit should be done while FRDY indicates 1. When checking the FRDY bit after executing BVAL, allow an interval of at least 80 ns after executing BVAL before referencing FRDY.

When the specified pipe is in the receiving direction, do not write 1 to this bit.

30.2.5 Enabling Interrupts

30.2.5.1 INTENB0 : Interrupt Enable Register 0

Base address: USBF = 0x8020_1000

Offset address: 0x30

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	BRDYE	Buffer Ready Interrupt Enable Enables or disables the USB interrupt output when the BRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
9	NRDYE	Buffer Not Ready Response Interrupt Enable Enables or disables the USB interrupt output when the NRDY interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
10	BEMPE	Buffer Empty Interrupt Enable Enables or disables the USB interrupt output when the BEMP interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
11	CTRE	Control Transfer Stage Transition Interrupt Enable Enables or disables the USB interrupt output when the CTRT interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
12	DVSE	Device State Transition Interrupt Enable Enables or disables the USB interrupt output when the DVST interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
13	SOFE	Frame Number Update Interrupt Enable Enables or disables the USB interrupt output when the SOF interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
14	RSME	Resume Interrupt Enable Enables or disables the USB interrupt output when the RESM interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
15	VBSE	VBUS Interrupt Enable Enables or disables the USB interrupt output when the VBINT interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W

Note: When a USB bus reset is detected, this register value is undefined.

30.2.5.2 INTENB1 : Interrupt Enable Register 1

Base address: USBF = 0x8020_1000

Offset address: 0x32

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PDDETINTE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PDDETINTE	PDDETINT Detection Interrupt Enable Enables or disables the USB interrupt output when a PDDETINT detection interrupt is detected. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

30.2.5.3 BRDYENB : BRDY Interrupt Enable Register

Base address: USBF = 0x8020_1000

Offset address: 0x36

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPEBRDYE[9:0]									
Value after reset:	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPEBRDYE[9:0]	BRDY Interrupt Enable for Each Pipe Enable or disable the BRDY interrupt for each pipe. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
15:10	—	The read values are undefined. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note: The bit number corresponds to the pipe number.

PIPEBRDYE[9:0] bits (BRDY Interrupt Enable for Each Pipe)

On detecting the BRDY interrupt for the pipe corresponding to the bit in this register which has been set to 1 by software, this controller sets 1 to the corresponding PIPEBRDY bit in BRDYSTS and the BRDY bit in INTSTS0, and asserts the interrupt signal through USB_FI.

While at least one PIPEBRDY bit in BRDYSTS indicates 1, this controller asserts the interrupt signal through USB_FI when the corresponding interrupt enable bit of this register is changed from 0 to 1 by software.

30.2.5.4 NRDYENB : NRDY Interrupt Enable Register

Base address: USBF = 0x8020_1000

Offset address: 0x38

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPENRDYE[9:0]									
Value after reset:	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPENRDYE[9:0]	NRDY Interrupt Enable for Each Pipe Enable or disable the NRDY interrupt for each pipe. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
15:10	—	The read values are undefined. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note: The bit number corresponds to the pipe number.

PIPENRDYE[9:0] bits (NRDY Interrupt Enable for Each Pipe)

On detecting the NRDY interrupt for the pipe corresponding to the bit in this register which has been set to 1 by software, this controller sets 1 to the corresponding PIPENRDY bit in NRDYSTS and the NRDY bit in INTSTS0, and asserts the interrupt signal through the USB Function Interrupt (USB_FI).

While at least one PIPENRDY bit in NRDYSTS indicates 1, this controller asserts the interrupt signal through USB_FI when the corresponding interrupt enable bit of this register is changed from 0 to 1 by software.

30.2.5.5 BEMPENB : BEMP Interrupt Enable Register

Base address: USBF = 0x8020_1000

Offset address: 0x3A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPEBEMPE[9:0]									
Value after reset:	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPEBEMPE[9:0]	BEMP Interrupt Enable for Each Pipe Enable or disable the BEMP interrupt for each pipe. 0: Interrupt output disabled 1: Interrupt output enabled	R/W
15:10	—	The read values are undefined. The write value should be 0.	R/W

Note:
- When a USB bus reset is detected, this register value is undefined.
- The bit number corresponds to the pipe number.

PIPEBEMPE[9:0] bits (BEMP Interrupt Enable for Each Pipe)

On detecting the BEMP interrupt for the pipe corresponding to the bit in this register which has been set to 1 by software, this controller sets 1 to the corresponding PIPEBEMP bit in BEMPSTS and the BEMP bit in INTSTS0, and asserts the interrupt signal through USB_FI.

While at least one PIPEBEMP bit in BEMPSTS indicates 1, this controller asserts the interrupt signal through USB_FI when the corresponding interrupt enable bit of this register is changed from 0 to 1 by software.

30.2.6 SOF Control Register

30.2.6.1 SOFCFG : SOF Pin Configuration Register

Base address: USBF = 0x8020_1000

Offset address: 0x3C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	BRDYM	INTL	EDGESTS	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0. The write value should be 0.	R/W
4	EDGESTS	Interrupt Edge Processing Status Indicates the state of processing in response to interrupts when edge-sensing has been selected. 0: Processing in response to an edge-sensed interrupt is not in progress. 1: Processing in response to an edge-sensed interrupt is in progress	R

Bit	Symbol	Function	R/W
5	INTL	Interrupt Output Sense Select*2 Selects sense mode for USB interrupt output. 0: Edge sense 1: Level sense	R/W
6	BRDYM	PIPEBRDY Interrupt Status Clear Timing*1 Specifies the timing for clearing the PIPEBRDY interrupt state. This bit can be set only in the initial setting (before communications). The setting cannot be changed once communication starts. 0: Software clears the state. 1: Hardware clears the state by reading from the FIFO buffer or by writing to the FIFO buffer.	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note 1. When setting the BRDYM bit to 1, set the INTL bit to 1 (level sense).
Note 2. With the INTL bit set to 0, to stop the system clock (SUSPM = 0) after clearing the interrupt state, write 0 to the SUSPM bit after checking that the EDGESTS bit is set to 0.

30.2.7 Interrupt Status

30.2.7.1 INTSTS0 : Interrupt Status Register 0

Base address: USBF = 0x8020_1000

Offset address: 0x40

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	VBINT	RESM	SOFR	DVST	CTRT	BEMP	NRDY	BRDY	VBSTS	DVSQ[2:0]			VALID	CTSQ[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CTSQ[2:0]	Control Transfer Stage Indicate the control transfer stage. 0 0 0: Idle or setup stage 0 0 1: Control read data stage 0 1 0: Control read status stage 0 1 1: Control write data stage 1 0 0: Control write status stage 1 0 1: Control write (no data) status stage 1 1 0: Control transfer sequence error 1 1 1: Reserved	R
3	VALID	USB Request Reception Indicates whether reception of the USB request has been detected. When writing to this bit, only 0 can be written. 0: Not detected 1: Setup packet reception	R/W
6:4	DVSQ[2:0]	Device State Indicate the device state. 0 0 0: Powered state 0 0 1: Default state 0 1 0: Address state 0 1 1: Configured state 1 x x: Suspended state	R
7	VBSTS	VBUS Input Status Indicates the input state of the USB_VBUSIN pin. 0: USB_VBUSIN pin is at the low level 1: USB_VBUSIN pin is at the high level	R

Bit	Symbol	Function	R/W
8	BRDY	BRDY Interrupt Status Indicates the BRDY interrupt state. 0: BRDY interrupts not generated 1: BRDY interrupts generated	R
9	NRDY	NRDY Interrupt Status Indicates the NRDY interrupt state. 0: NRDY interrupts not generated 1: NRDY interrupts generated	R
10	BEMP	BEMP Interrupt Status Indicates the BEMP interrupt state. 0: BEMP interrupts not generated 1: BEMP interrupts generated	R
11	CTRT	Control Transfer Stage Transition Interrupt Status Indicates the control transfer stage transition interrupt state. When writing to this bit, only 0 can be written. 0: Control transfer stage transition interrupts not generated 1: Control transfer stage transition interrupts generated	R/W*1
12	DVST	Device State Transition Interrupt Status Indicates the device state transition interrupt state. When writing to this bit, only 0 can be written. 0: Device state transition interrupts not generated 1: Device state transition interrupts generated	R/W*1
13	SOFR	Frame Number Update Interrupt Status Indicates the frame number update interrupt state. When writing to this bit, only 0 can be written. 0: SOF interrupts not generated 1: SOF interrupts generated	R/W*1
14	RESM	Resume Interrupt Status*2 Indicates the resume detection interrupt state. When writing to this bit, only 0 can be written. 0: Resume interrupts not generated 1: Resume interrupts generated	R/W*1
15	VBINT	VBUS Change Detect Interrupt Status*2 Indicates the VBUS change detection interrupt state. When writing to this bit, only 0 can be written. 0: VBUS interrupts not generated 1: VBUS interrupts generated	R/W*1

Note: When a USB bus reset is detected, The DVST is set to 1 and DVSQ[2:0] are set to 001b and other bits are undefined.
Note 1. To clear the state indicated by the VBINT, RESM, SOFR, DVST or CTRT bits, write 0 only to the bit to be cleared, and write 1 to the other bits. Do not write 0 to the status bit set to 0.
Note 2. The controller detects the change in state indicated by the VBINT and RESM bits of this register, even while the clock is being stopped (SUSPM = 0), and conveys the interrupt if the corresponding interrupt is enabled. When the clock is enabled, clear the state using software.

BRDY bit (BRDY Interrupt Status)

This controller sets a BRDY bit to 1 when, among the PIPEBRDY bits for which the corresponding PIPEBRDYE bits in BRDYENB have been set to 1, at least one PIPEBRDY bit in BRDYSTS is set to 1 (that is, when this controller detects the BRDY interrupt state in at least one pipe among the pipes for which software has enabled the output of BRDY interrupts).

For the conditions for the setting of PIPEBRDY bits, see the description of the BRDYSTS register.

When a BRDY bit has been set to 1, the controller clears the bit in response to software writing 0 to all PIPEBRDY bits corresponding to PIPEBRDYE bits which have been set to 1.

A BRDY bit will not be cleared to 0 in response to software only writing 0 to individual bits that have been set to 1.

NRDY bit (NRDY Interrupt Status)

This controller sets an NRDY bit to 1 when, among the PIPEBRDY bits for which the corresponding PIPENRDYE bits in NRDYENB have been set to 1, at least one PIPENRDY bit in BNRDYSTS is set to 1 (that is, when this controller detects the NRDY interrupt state in at least one pipe among the pipes for which software has enabled the output of NRDY interrupts).

For the conditions for the setting of PIPENRDY bits, see the description of the NRDYSTS register.

When an NRDY bit has been set to 1, the controller clears the bit in response to software writing 0 to all PIPENRDY bits corresponding to PIPENRDYE bits which have been set to 1.

An NRDY bit will not be cleared to 0 in response to software only writing 0 to individual bits that have been set to 1.

BEMP bit (BEMP Interrupt Status)

This controller sets a BEMP bit to 1 when, among the PIPEBRDY bits for which the corresponding PIPEBEMPE bits in BEMPENB have been set to 1, at least one PIPEBEMP bit in BEMPSTS is set to 1 (that is, when this controller detects the BEMP interrupt state in at least one pipe among the pipes for which software has enabled the output of BEMP interrupts).

For the conditions for the setting of PIPEBEMP bits, see the description of the BEMPSTS register.

When a BEMP bit has been set to 1, the controller clears the bit in response to software writing 0 to all PIPEBEMP bits corresponding to PIPEBEMPE bits which have been set to 1.

A BEMP bit will not be cleared to 0 in response to software only writing 0 to individual bits that have been set to 1.

CTRT bit (Control Transfer Stage Transition Interrupt Status)

If this controller detects the stage transition of control transfer, it updates the CTSQ value and sets 1 to this bit.

When this interrupt occurs, clear the state before the controller detects the next control transfer stage transition.

DVST bit (Device State Transition Interrupt Status)

If this controller detects a change in the device state, it updates the DVSQ value and sets 1 to this bit.

When this interrupt occurs, clear the state before the controller detects the next device state transition.

SOFR bit (Frame Number Update Interrupt Status)

The conditions when the controller sets 1 in this bit are below.

When updating the frame number, this controller sets 1 to this bit (this interrupt is detected every 1 ms).

The controller detects the SOFR interrupt by internal interpolation even if the SOF packet from the USB host is corrupted.

RESM bit (Resume Interrupt Status)

If this controller is in the suspended state (DVSQ = 1xxb) and the DP pin falling edge is detected, 1 is set to this bit.

VBINT bit (VBUS Change Detect Interrupt Status)

When this controller detects a change in the USB_VBUSIN pin input value (from high to low and from low to high), 1 is written to this bit. The controller writes the input value of the USB_VBUSIN pin to the VBSTS bit. When the VBINT interrupt occurs, use the software to execute a consistency check several times during reading the VBSTS bit, and remove the chattering effect.

30.2.7.2 INTSTS1 : Interrupt Status Register 1

Base address: USBF = 0x8020_1000

Offset address: 0x42

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PDDETINT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PDDETINT	PDDET Detection Interrupt Status*1 Indicates the PDDETINT interrupt status. 0: PDDET interrupts not generated 1: PDDET interrupts generated	R/W*2
15:1	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, The bit 12 is set to 1 and other bits are undefined.

Note 1. To clear the status indicated by the PDDETINT bit, write 0 to bit0 only and 1 to other bits. Do not write 0 to status bits that indicate 0.

Note 2. This controller detects changes in the status indicated by the PDDETINT bit in this register even while the clock is stopped (SUSPM = 0), and reports corresponding interrupts if they are enabled. Perform a software status clear after the clock is enabled.

PDDETINT bit (PDDET Detection Interrupt Status)

This bit is set to 1 when a change (high to low or low to high) in EXH_PDDETSTS (BC1/2-PD detection status signal) is detected. When a PDDETINT interrupt is generated, read the PDDETSTS bit several times by software to perform debounce processing.

30.2.7.3 BRDYSTS : BRDY Interrupt Status Register

Base address: USBF = 0x8020_1000

Offset address: 0x46

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPEBRDY[9:0]									
Value after reset:	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPEBRDY[9:0]	BRDY Interrupt Status for Each Pipe*1 Indicate the BRDY interrupt state for each pipe. When writing to these bits, only 0 can be written. 0: Interrupts not generated 1: Interrupts generated	R/W*2*3
15:10	—	The read values are undefined. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note 1. The bit number corresponds to the pipe number.
Note 2. When BRDYM = 0, to clear the state of each bit of this register, write 0 only to the bit to be cleared and 1 to other bits.
Note 3. When BRDYM = 0, clearing these interrupt status bits should be done before accessing the FIFO.

PIPEBRDY[9:0] bits (BRDY Interrupt Status for Each Pipe)

When the BRDY interrupt is detected for a pipe by this controller, the controller sets 1 in the corresponding PIPEBRDY[9:0] bit of the BRDYSTS register. Here, when 1 is written to the corresponding bit of BRDYENB register by using the software, the controller sets 1 to the BRDY bit of the INTSTS0 register.

The conditions for generating and clearing the BRDY interrupt depend on the settings of the SOFCFG.BRDYM bit and PIPECFG.BFRE bit for each pipe as described below.

- When BRDYM = 0 and BFRE = 0
  With these settings, the BRDY interrupt indicates that the FIFO port is accessible.
  On any of the following conditions, this controller generates the internal BRDY interrupt request trigger and sets 1 to the PIPEBRDY bit corresponding to the selected pipe.
  1. For the pipe in the transmitting direction
     (a) When the software has modified the DIR bit from 0 to 1
     (b) When the controller ends the packet transmission of the selected pipe in the condition where writing is not possible from the CPU to the FIFO buffer that has been assigned to the selected pipe (when the BSTS bit is read as 0).
     In continuous transmission/reception mode, the request trigger is generated when data of one plane of the FIFO buffer is complete.
     (c) In double-buffer operation, one FIFO buffer being empty on completion of the writing of data to the other FIFO buffer.
     The request trigger is not generated until completion of writing data to the currently-written FIFO buffer plane even if transmission to the other FIFO buffer is completed.
     (d) When the hardware flushes the buffer of the pipe for isochronous transfers.
     (e) The FIFO buffer making the transition from the write-disabled to the write-enabled state in response to writing of 1 to the ACLRM bit.
     The request trigger is not generated for the DCP (that is, during data transmission for control transfers).

2. For the pipe in the receiving direction

   (a) When packet reception is completed successfully thus enabling the FIFO buffer to be read when read-access from the CPU to the FIFO buffer for the selected pipe is disabled (when the BSTS bit is read as 0).
   The request trigger is not generated for the transaction in which DATA-PID disagreement occurs.
   When a short packet is received, the request trigger is generated even if the FIFO buffer has available space.
   When the transaction counter is used, the request trigger is generated on receiving the specified number of packets.
   In this case, the request trigger is generated even if the FIFO buffer has available space.

   (b) In double-buffer operation, one FIFO buffer being ready for reading on completion of the reading of data from the other FIFO buffer.
   The request trigger is not generated until completion of reading data from the currently-read FIFO buffer plane even if reception by the other FIFO buffer is completed.
   The BRDY interrupt is not generated in the status stage of control transfers.
   The PIPEBRDY interrupt state of the selected pipe can be cleared to 0 by writing 0 to the corresponding PIPEBRDY interrupt status bit in the BRDYSTS register. In this case, 1s should be written to the PIPEBRDY interrupt status bits for the other pipes.
   Be sure to clear the BRDY state before accessing the FIFO buffer.

- When BRDYM = 0 and BFRE = 1
  With these settings, this controller generates the BRDY interrupt on completion of the reading of all data for a single transfer using the pipe in the receiving direction, and sets 1 to the PIPEBRDY bit corresponding to the selected pipe.
  On any of the following conditions, this controller determines that the last data for a single transfer has been received.

  1. When a short packet including a zero-length packet is received.
  2. When the transaction counter register (TRNCNT bits) is used and the number of packets specified by the TRNCNT bits is completely received.

  When the pertinent data is completely read out after any of the above determination conditions has been satisfied, this controller determines that all data for a single transfer has been completely read out.
  When a zero-length packet is received when the FIFO buffer is empty, the controller determines that all data for a single transfer has been completely read out upon the FRDY and the DTLN bits of the FIFO port control register are set to 1 and 0, respectively.
  In this case, write 1 to the BCLR bit of the corresponding FIFOCTR register by software to start the next transfer.
  With these settings, this controller does not detect the BRDY interrupt for the pipe in the transmitting direction.
  The PIPEBRDY interrupt state of the selected pipe can be cleared to 0 by writing 0 to the corresponding PIPEBRDY interrupt status bit. In this case, 1s should be written to the PIPEBRDY interrupt status bits for the other pipes.
  In this mode, the BFRE bit setting should not be modified until all data for a single transfer has been processed.
  When it is necessary to modify the BFRE bit before completion of processing, all the FIFO buffers for the selected pipe should be cleared using the ACLRM bit.

- When BRDYM = 1 and BFRE = 0
  With these settings, the PIPEBRDY values are linked to the BSTS bit settings for each pipe. In other words, the BRDY interrupt status bits (PIPEBRDY) are set to 1 or 0 by this controller depending on the FIFO buffer state.

  1. For the pipe in the transmitting direction
     The setting of a BRDY interrupt status bit is 1 while the port register of the corresponding FIFO buffer is ready for written data and 0 when it is not able to accept written data.
     However, the BRDY interrupt is not generated in response to setting of the bit to 1 if writing to the DCP in the transmitting direction can proceed.
  2. For the pipe in the receiving direction
     The setting of a BRDY interrupt status bit is 1 while the port register of the FIFO buffer is ready for reading and 0 when all data have been read.

  When a zero-length packet is received when the FIFO buffer is empty, the corresponding bit is set to 1 and the BRDY interrupt continues to be effective until 1 is written to BCLR.
  With this setting, the PIPEBRDY bit cannot be cleared to 0.
  When BRDYM is set to 1, all of the BFRE bits (for all pipes) should be cleared to 0.
  When BRDYM is set to 1, the INTL bit should be set to 1 (level control).

30.2.7.4 NRDYSTS : NRDY Interrupt Status Register

Base address: USBF = 0x8020_1000

Offset address: 0x48

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPENRDY[9:0]									
Value after reset:	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPENRDY[9:0]	NRDY Interrupt Status for Each Pipe*1 Indicate the NRDY interrupt state for each pipe. When writing to these bits, only 0 can be written. 0: Interrupts not generated 1: Interrupts generated	R/W*2
15:10	—	The read values are undefined. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note 1. The bit number corresponds to the pipe number.
Note 2. To clear the state indicated by each bit of this register, write 0 only to the bit to be cleared and 1 to the other bits.

PIPENRDY[9:0] bits (NRDY Interrupt Status for Each Pipe)

On generating the internal NRDY interrupt request for the pipe whose PID bits are set to BUF, this controller sets the corresponding PIPENRDY bit in NRDYSTS in the register to 1. If the corresponding bit in the NRDYENB register is set to 1, this controller sets the NRDY bit in INTSTS0 in the register to 1.

The conditions on which this controller generates the internal NRDY interrupt request for a given pipe are described below.

However, the internal NRDY interrupt request is not generated during status stage execution of the control transfer.

1. When the pipe is in the transmitting direction
   (a) When an IN Token is received while there is no transmission data in the FIFO buffer and the corresponding PIPE PID bit is set to BUF (01):
   In this case, this controller generates an NRDY interrupt request at the reception of the IN token, setting the PIPENRDY bit to 1.
   For the pipe for the isochronous transfers in which an interrupt is generated, this controller transmits a zerolength packet, setting the OVRN bit to 1.
2. For the pipe in the receiving direction
   (a) When the PID bits for the corresponding pipe are set to BUF (01) and an OUT token is received while there is no open space in the FIFO buffer:
   For the pipe for the isochronous transfers in which an interrupt is generated, this controller generates an NRDY interrupt request at the reception of the OUT token, setting the PIPENRDY bit to 1 and OVRN bit to 1.
   For the pipe for the transfers other than isochronous transfers in which an interrupt is generated, this controller generates an NRDY interrupt request when an NAK handshake is transferred after the data following the OUT token was received, setting the PIPENRDY bit to 1.
   However, during re-transmission (due to DATA-PID disagreement), the NRDY interrupt request is not generated.
   In addition, if an error occurs in the DATA packet, the NRDY interrupt request is not generated.
   (b) When the PID bits for the corresponding pipe are set to BUF (01) and a PING token is received while there is no open space in the FIFO buffer:
   In this case, this controller generates an NRDY interrupt request at the reception of the PING token, setting the PIPENRDY bit to 1.
   (c) In an isochronous transfer pipe, when the PID bits are set to BUF (01) and data is not received successfully within the interval frame:
   In this case, this controller generates an NRDY interrupt request at the reception of an SOF, and sets the PIPENRDY bit to 1.

30.2.7.5 BEMPSTS : BEMP Interrupt Status Register

Base address: USBF = 0x8020_1000

Offset address: 0x4A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	PIPEBEMP[9:0]									
Value after reset:	x	x	x	x	x	x	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	PIPEBEMP[9:0]	BEMP Interrupt Status for Each Pipe*1 Indicate the BEMP interrupt state for each pipe. When writing to these bits, only 0 can be written. 0: Interrupts not generated 1: Interrupts generated	R/W*2
15:10	—	The read values are undefined. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note 1. The bit number corresponds to the pipe number.
Note 2. To clear the state indicated by each bit of this register, write 0 only to the bit to be cleared and 1 to the other bits.

PIPEBEMP[9:0] bits (BEMP Interrupt Status for Each Pipe)

On detecting the BEMP interrupt for the pipe whose PID bits are set to BUF by software, this controller sets the corresponding PIPEBEMP bit in BEMPSTS to 1. If the corresponding bit in BEMPENB is set to 1 by software, this controller sets the BEMP bit in INTSTS0 to 1.

The following describes the conditions on which this controller generates the internal BEMP interrupt request.

1. For the pipe in the transmitting direction, when the FIFO buffer of the corresponding pipe is empty on completion of transmission (including zero-length packet transmission).
   In single-buffer operation, the internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for the pipe other than DCP.
   However, the internal BEMP interrupt request is not generated on any of the following conditions.
   (a) When writing data to the FIFO buffer on the CPU (DMAC) side is started by software on completion of transmitting data of one plane in double-buffer operation.
   (b) When the buffer is cleared (emptied) by setting the ACLRM or BCLR bit to 1.
   (c) When IN transfer (zero-length packet transmission) is performed during the control transfer status stage.
2. For the pipe in the receiving direction
   When the successfully-received data packet size exceeds the specified maximum packet size.
   In this case, this controller generates the BEMP interrupt request, setting the corresponding PIPEBEMP bit to 1, and discards the received data and modifies the setting of the PID bits of the corresponding pipe to STALL (11). However, the internal BEMP interrupt request is not generated on any of the following conditions.
   (a) When a CRC error or bit stuffing error is detected in the received data.
   (b) When a setup transaction is being performed.

Writing 0 to the PIPEBEMP bit clears the state; writing 1 to the PIPEBEMP bit has no effect.

30.2.8 Frame Number Registers

30.2.8.1 FRMNUM : Frame Number Register

Base address: USBF = 0x8020_1000

Offset address: 0x4C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OVRN	CRCE	—	—	—	FRNM[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	FRNM[10:0]	Frame Number Indicate the latest frame number.	R
13:11	—	These bits are read as 0. The write value should be 0.	R/W
14	CRCE	CRC Error Detection Status Indicates whether a CRC error has been detected in the pipe during isochronous transfer. When writing to this bit, only 0 can be written. 0: No error 1: An error occurred.	R/W
15	OVRN	Overrun/Underrun Detection Status*1 Indicates whether an overrun/underrun error has been detected in the pipe during isochronous transfer. When writing to this bit, only 0 can be written. 0: No error 1: An error occurred.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note 1. The OVRN bit is for use in debugging. When designing a system, control the timing so that neither overrun nor underrun occurs.

FRNM[10:0] bits (Frame Number)

This controller sets these bits to indicate the latest frame number, which is updated every time an SOF packet is issued or received (every 1 ms).

When reading these bits by software, repeat reading until the same value is read twice.

CRCE bit (CRC Error Detection Status)

In an isochronous transfer pipe, this controller sets 1 to this bit when a CRC error or bit stuffing error has been detected.

This bit can be cleared by writing 0 to it by software. In this case, write 0x8000 when OVRN is not to be cleared at the same time.

This controller generates an internal NRDY interrupt request when it detects a CRC error. For details, see NRDY Interrupt Enable Bit for Each Pipe (PIPENRDYE).

OVRN bit (Overrun/Underrun Detection Status)

For the isochronous transfer pipe, this controller sets 1 to this bit when an overrun/underrun is detected.

When an overrun/underrun is detected, the controller issues an internal NRDY request. See NRDY Interrupt Enable Bit for Each Pipe (PIPENRDYE) for details.

This bit can be cleared writing 0 to it by software. In this case, write 0x4000 when CRCE is not to be cleared at the same time.

This controller sets this bit to 1 on any of the following conditions.

1. For the isochronous transfer pipe in the transmitting direction, the IN token is received before all the transmit data has been written to the FIFO buffer.
2. For the isochronous transfer pipe in the receiving direction, the OUT token is received when no FIFO buffer planes are empty.

30.2.8.2 UFRMNUM : Frame Number Register

Base address: USBF = 0x8020_1000

Offset address: 0x4E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	UFRNM[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	UFRNM[2:0]	Microframe Number Indicate the µframe number.	R
15:3	—	These bits are read as 0.	R

Note: When a USB bus reset is detected, this register value is undefined.

UFRNM[2:0] bits (Microframe Number)

This controller sets these bits to indicate the µframe number during high-speed operation. During operation other than high-speed operation, this controller sets these bits to 0x00.

When reading these bits, repeat reading until the same value is read twice.

30.2.9 USB Address

30.2.9.1 USBADDR : USB Address Register

Base address: USBF = 0x8020_1000

Offset address: 0x50

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	USBADDR[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	USBADDR[6:0]	USB Address Indicate the USB address assigned by the host.	R
15:7	—	These bits are read as 0.	R

Note: When a USB bus reset is detected, The USBADDR[6:0] is set to 0000000b and other bits are undefined.

USBADDR[6:0] bits (USB Address)

These bits indicate the USB address assigned by the host when the SetAddress request is successfully processed.

If a USB bus reset is detected by the controller, 0x00 is set to this bit.

30.2.10 USB Request Registers

The USB request registers are used to store the setup requests for control transfers. The values of the USB request that have been received are stored.

30.2.10.1 USBREQ : USB Request Type Register

Base address: USBF = 0x8020_1000

Offset address: 0x54

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BREQUEST[7:0]								BMREQUESTTYPE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	BMREQUESTTYPE[7:0]	Request Type USB request bmRequestType value	R
15:8	BREQUEST[7:0]	Request USB request bRequest value	R

BMREQUESTTYPE[7:0] bits (Request Type)

These bits indicate the USB request data value received by this controller in the setup transaction. Writing to these bits by software has no effect.

BREQUEST[7:0] bits (Request)

These bits indicate the USB request data value received by this controller in the setup transaction. Writing to these bits by software has no effect.

30.2.10.2 USBVAL : USB Request Value Register

Base address: USBF = 0x8020_1000

Offset address: 0x56

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WVALUE[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WVALUE[15:0]	Value USB request wValue value	R

WVALUE[15:0] bits (Value)

These bits are used to read the value of the USB request wValue. Bits 7 to 0 are lower-order byte.

These bits indicate the USB request wValue value received by this controller in the setup transaction. Writing to these bits by software has no effect.

30.2.10.3 USBINDX : USB Request Index Register

Base address: USBF = 0x8020_1000

Offset address: 0x58

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WINDEX[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WINDEX[15:0]	Index USB request wIndex value	R

WINDEX[15:0] bits (Index)

These bits are used to read the value of the USB request wIndex. Bits 7 to 0 are lower-order byte.

These bits indicate the USB request wIndex value received by this controller in the setup transaction. Writing to these bits by software has no effect.

30.2.10.4 USBLENG : USB Request Length Register

Base address: USBF = 0x8020_1000

Offset address: 0x5A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WLENGTH[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WLENGTH[15:0]	Length USB request wLength value	R

WLENGTH[15:0] bits (Length)

These bits are used to read the value of the USB request wLength. Bits 7 to 0 are lower-order byte.

These bits indicate the USB request wLength value received in setup transaction by the controller. Writing to these bits by software has no effect.

30.2.11 DCP Configuration

30.2.11.1 DCPCFG : DCP Configuration Register

Base address: USBF = 0x8020_1000

Offset address: 0x5C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CNTMD	SHTNAK	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0. The write value should be 0.	R/W
7	SHTNAK	Disabling PIPE at the End of Transfer Specifies whether to change PID to NAK at the end of transfer when receiving direction is specified for the default control pipe. 0: The pipe continues at the end of transfer. 1: The pipe is disabled at the end of transfer.	R/W
8	CNTMD	Continuous Transfer Mode Specifies whether to enable the default control pipe communication in continuous transfer mode. 0: Continuous transfer mode disabled 1: Continuous transfer mode enabled	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

30.2.11.2 DCPMAXP : DCP Maximum Packet Size Register

Base address: USBF = 0x8020_1000

Offset address: 0x5E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	MXPS[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	MXPS[6:0]	Maximum Packet Size Specify the maximum data payload (maximum packet size) for the DCP.	R/W
15:7	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

MXPS[6:0] bits (Maximum Packet Size)

Set the maximum data payload (maximum packet size) for the DCP in these bits. 0x40 (64 bytes) is the value after a reset.

The MXPS bits should be set to the value based on the USB specification.

The MXPS bits should be set while PID = NAK and before setting the CURPIPE bits.

Before modifying these bits after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

While MXPS is 0, do not write to the FIFO buffer or do not set PID to BUF.

30.2.11.3 DCPCTR : DCP Control Register

Base address: USBF = 0x8020_1000

Offset address: 0x60

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSTS	—	—	—	—	—	—	SQCLR	SQSET	SQMON	PBUSY	—	—	CCPL	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PID[1:0]	Response PID These bits control the response type of this controller during control transfer. 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
2	CCPL	Control Transfer End Enable Setting this bit to 1 enables the status stage of the control transfer to be completed. 0: End of the control transfer disabled 1: End of the control transfer enabled	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	PBUSY	PIPE Busy Indicates whether the selected pipe is currently being used by the USB bus. 0: The pipe is not being used by the USB bus. 1: The pipe is being used by the USB bus.	R

Bit	Symbol	Function	R/W
6	SQMON	Sequence Toggle Bit Monitor Indicates the expected value of the sequence toggle bit for the next transaction during the DCP transfer. 0: DATA0 1: DATA1	R
7	SQSET	Toggle Bit Set Specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. This bit is read as 0. When writing to this bit, only 1 can be written. 0: Writing has no effect 1: Specifies DATA1	R/W
8	SQCLR	Toggle Bit Clear Specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. This bit is read as 0. When writing to this bit, only 1 can be written. 0: Writing has no effect 1: Specifies DATA0	R/W
10:9	—	These bits are read as 0. The write value should be 0.	R/W
11	—	This bit is read as 0. The write value should be 1.	R/W
12	—	The read value is undefined. The write value should be 0.	R/W
14:13	—	These bits are read as 0. The write value should be 1.	R/W
15	BSTS	Buffer Status Indicates whether access to the DCP FIFO buffer is enabled or disabled. 0: Buffer access disabled 1: Buffer access enabled	R

Note: When a USB bus reset is detected, The CCPL is set to 0 and PID[1:0] are set to 0 and other bits are undefined.

PID[1:0] bits (Response PID)

Change the setting of these bits from NAK to BUF by software during data stage or status stage of control transfers.

This controller changes the setting of these bits in the following conditions:

1. This controller changes the setting of the PID bits to select NAK response (00) on receiving the setup packet. Here, this controller sets VALID to 1. The setting of the PID bits cannot be changed until VALID is set to 0.
2. This controller sets the PID bits to select STALL response (11) on receiving the data of the size exceeding the maximum packet size when the PID bits have been set to select BUF response.
3. This controller sets the PID bits to select STALL response (1x) on detecting the control transfer sequence error.
4. This controller sets the PID bits to select NAK response on detecting the USB bus reset.

This controller does not reference to the setting of the PID bits while the SET_ADDRESS request is processed (auto processing).

CCPL bit (Control Transfer End Enable)

Setting this bit to 1 while the corresponding PID bits are set to BUF enables the stage of the control transfer to be completed.

Specifically, during control read transfer, this controller transmits the ACK handshake in response to the OUT transaction from the USB host, and outputs the zero-length packet in response to the IN transaction from the USB host during control writing or no-data control transfer. However, on detecting the SET_ADDRESS request, this controller operates in auto response mode from the setup stage up to the status stage completion irrespective of the setting of this bit.

This controller modifies this bit from 1 to 0 on receiving the new setup packet.

1 cannot be written by software to this bit while VALID is 1.

PBUSY bit (PIPE Busy)

This controller modifies this bit from 0 to 1 upon start of the USB transaction for the selected pipe, and modifies the bit from 1 to 0 upon completion of one transaction.

Reading this bit after the PID bits have been set to NAK allows checking that modification of the pipe settings is possible.

SQMON bit (Sequence Toggle Bit Monitor)

This bit indicates the expected value of the sequence toggle bit of the selected pipe.

This bit is toggled upon normal completion of the transaction. However, this bit is not toggled when a DATA-PID disagreement occurs during the receiving transfer.

This controller sets this bit to 1 (specifies DATA1 as the expected value) upon normal reception of the setup packet.

This controller does not reference to this bit during the IN/OUT transaction of the status stage, and does not allow this bit to toggle upon normal completion.

SQSET bit (Toggle Bit Set)

Setting this bit to 1 by software allows this controller to set DATA1 as the expected value of the sequence toggle bit of the selected pipe. This bit always indicates 0.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

Set this bit to 1 while PID is NAK and before setting the CURPIPE bits.

Before setting this bit to 1 after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

SQCLR bit (Toggle Bit Clear)

Setting this bit to 1 by software allows this controller to set DATA0 as the expected value of the sequence toggle bit of the selected pipe. This bit always indicates 0.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

Set this bit to 1 while PID is NAK and before setting the CURPIPE bits.

Before setting this bit to 1 after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

BSTS bit (Buffer Status)

This bit indicates whether access from the CPU to the FIFO buffer assigned to the DCP is possible.

The meaning of the BSTS bit depends on the ISEL bit setting as follows.

1. When ISEL = 0: indicates whether the received data can be read from the buffer.
2. When ISEL = 1: indicates whether the data for transmission can be written to the buffer.

30.2.12 Pipe Configuration Register

Pipes 1 to 9 should be set using the PIPESEL, PIPECFG, PIPEBUF, PIPEMAXP, PIPEPERI, PIPExCTR, PIPExTRE and PIPExTRN registers.

After selecting the pipe using the PIPESEL register, functions of the pipe should be set using the PIPECFG, PIPEBUF, PIPEMAXP and PIPEPERI registers. The PIPExCTR, PIPExTRE and PIPExTRN registers can be set regardless of the pipe selection in the PIPESEL register.

30.2.12.1 PIPESEL : Pipe Window Select Register

Base address: USBF = 0x8020_1000

Offset address: 0x64

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	PIPESEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	PIPESEL[3:0]	Pipe Window Select Specify the pipe corresponding to the PIPECFG, PIPEBUF, PIPEMAXP, PIPEPERI registers. 0x0: No pipe selected 0x1: Pipe 1 0x2: Pipe 2 0x3: Pipe 3 0x4: Pipe 4 0x5: Pipe 5 0x6: Pipe 6 0x7: Pipe 7 0x8: Pipe 8 0x9: Pipe 9 Others: Reserved	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

PIPESEL[3:0] bit (Pipe Window Select)

When a value between 0001b and 1001b is set in these bits, the information and settings for the corresponding pipe can be read from the PIPECFG, PIPEBUF, PIPEMAXP, PIPEPERI registers. After a pipe is specified by setting these bits, the values set by software to the PIPECFG, PIPEBUF, PIPEMAXP, PIPEPERI registers are reflected in the corresponding pipe transfer type by this controller.

When 0000b is set in PIPESEL, 0 is read from all of the bits in the PIPECFG, PIPEBUF, PIPEMAXP, and PIPEPERI registers. When 0000b is set in PIPESEL, writing to the bits in these registers has no effect.

30.2.12.2 PIPECFG : Pipe Configuration Register

Base address: USBF = 0x8020_1000

Offset address: 0x68

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TYPE[1:0]		—	—	—	BFRE	DBLB	CNTMD	SHTNAK	—	—	DIR	EPNUM[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	EPNUM[3:0]	Endpoint Number Specify endpoint number for the selected pipe.	R/W
4	DIR	Transfer Direction Specifies pipe transfer direction for the selected pipe. 0: Receiving direction 1: Transmitting direction	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	SHTNAK	Pipe Disable at the End of Transfer Specifies whether to change PID to NAK at the end of transfer when the selected pipe is in the receiving direction. 0: Pipe continued at end of transfer 1: Pipe disabled at end of transfer	R/W
8	CNTMD	Continuous Transfer Mode Specifies whether to use the selected pipe in continuous transfer mode. 0: Non-continuous transfer mode 1: Continuous transfer mode	R/W
9	DBLB	Double Buffer Mode Selects either single- or double-buffer operation for the FIFO buffer used by the selected pipe. 0: Single-buffer operation 1: Double-buffer operation	R/W

Bit	Symbol	Function	R/W
10	BFRE	BRDY Interrupt Operation Specification Specifies the BRDY interrupt generation timing from this controller to the CPU with respect to the selected pipe. 0: BRDY interrupt notification upon transmitting or receiving data 1: BRDY interrupt notification upon reading data	R/W
13:11	—	These bits are read as 0. The write value should be 0.	R/W
15:14	TYPE[1:0]	Transfer Type Specify the transfer type for the pipe selected by the PIPESEL bits (selected pipe). 0 0: Pipe is not in use. 0 1: Bulk transfer 1 0: Interrupt transfer 1 1: Isochronous transfer	R/W

Note: When a USB bus reset is detected, this register value is undefined.

EPNUM[3:0] bits (Endpoint Number)

Specify the endpoint number for the selected pipe in these bits by software.

Setting 0000b indicates that the pipe is not being used.

These bits should be modified while PID is NAK, and the pipe is not selected by the CURPIPE bits.

Before modifying these bits after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

Do not make the settings such that the combination of the set values in the DIR and EPNUM bits should be the same for two or more pipes (EPNUM = 0000b (the selected pipe is not in use) can be set for all the pipes).

DIR bit (Transfer Direction)

When this bit is set to 0 by software, this controller uses the selected pipe in the receiving direction, and when this bit is set to 1, this controller uses the selected pipe in the transmitting direction.

This bit should be modified while PID is NAK and the pipe is not selected by the CURPIPE bits.

To modify this bit after completing USB communication using the selected pipe, write 1 and then 0 to ACLRM continuously to clear the FIFO buffer assigned to the selected pipe while the PID and CURPIPE bits are in the abovedescribed state.

Before modifying this bit after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

SHTNAK bit (Pipe Disable at the End of Transfer)

This bit is valid when pipes 1 to 5 are selected in the receiving direction.

When this bit is set to 1 for the selected pipe in the receiving direction, this controller modifies the PID bits for the selected pipe to NAK on determining the end of the transfer. This controller determines that the transfer has ended on any of the following conditions.

1. A short packet (including a zero-length packet) is successfully received.
2. The transaction counter is used and the number of packets specified for the counter is successfully received.

This bit should be modified while PID is NAK.

Before modifying this bit after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

This bit should be cleared to 0 for the pipe in the transmitting direction.

CNTMD bit (Continuous Transfer Mode)

This bit is valid when pipes 1 to 5 are selected and the transfer type of the selected pipe is set to bulk.

According to the setting value of this bit, this controller determines transmission/reception completion for the FIFO buffer assigned to the selected pipe. See Table 30.10.

Table 30.10 Relationship between transfer mode settings by CNTMD bit and timings at which reading data or transmitting data from FIFO buffer is enabled

CNTMD bit setting	When reading data or transmitting data is enabled
0	In the receiving direction (DIR = 0), the FIFO buffer is ready for reading in the following case. This controller receives one packet.
	In the transmitting direction (DIR = 1), the FIFO buffer is ready for transmission in either of the following cases. 1. Data of the maximum packet size is written to the FIFO buffer by software (or DMAC). 2. Data of the short packet size (including 0 byte data) is written to the FIFO buffer and then 1 is written to BVAL by software (or DMAC).
1	In the receiving direction (DIR = 0), the FIFO buffer is ready for reading in any of the following cases. 1. The number of the data bytes received in the FIFO buffer assigned to the selected pipe is equal to the number of assigned data bytes ((BUFSIZE + 1) × 64). 2. This controller receives a short packet other than a zero-length packet. 3. This controller receives a zero-length packet when data is already stored in the FIFO buffer assigned to the selected pipe. 4. This controller receives the number of packets equal to the transaction counter value specified for the selected pipe by software.
	In the transmitting direction (DIR = 1), the FIFO buffer is ready for transmission in any of the following cases 1 to 3. 1. The number of the data bytes written to the FIFO buffer by software (or DMAC) is equal to the number of data bytes in a single FIFO buffer plane assigned to the selected pipe. 2. The number of data bytes less than the size of a single FIFO buffer plane (including 0 byte data) assigned to the selected pipe is written to the FIFO buffer and then 1 is written to BVAL by software (or DMAC). 3. The number of data bytes less than the size of a single FIFO buffer plane (including 0 byte data) assigned to the selected pipe is written to the FIFO buffer and then the DENDx_N signal is asserted at the same time as writing the last data by software (or DMAC).

This bit should be modified while PID is NAK and the pipe is not selected by the CURPIPE bits.

To modify this bit after completing USB communication using the selected pipe, write 1 and then 0 to ACLRM continuously to clear the FIFO buffer assigned to the selected pipe while the PID and CURPIPE bits are in the abovedescribed state.

Before modifying this bit after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

DBLB bit (Double Buffer Mode)

This bit is valid when pipes 1 to 5 are selected.

When this bit is set to 1, this controller assigns two planes of the FIFO buffer size specified by the BUFSIZE bits in PIPEBUF to the selected pipe.

Specifically, the following expression determines the FIFO buffer size assigned to the selected pipe by this controller.

(BUFSIZE + 1) × 64 × (DBLB + 1) [bytes]

This bit should be modified while PID is NAK and the pipe is not selected by the CURPIPE bits.

To modify this bit after completing USB communication using the selected pipe, write 1 and then 0 to ACLRM continuously by software to clear the FIFO buffer assigned to the selected pipe while the PID and CURPIPE bits are in the above-described state.

Before modifying this bit after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

BFRE bit (BRDY Interrupt Operation Specification)

This bit is valid when pipes 1 to 5 are selected.

When this bit is set to 1 and the selected pipe is in the receiving direction (the DIR bit is set to 0), this controller detects the transfer completion and generates the BRDY interrupt on having read the pertinent packet.

When the BRDY interrupt is generated with the above conditions, 1 must be written to BCLR. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to BCLR.

When this bit is set to 1 and the selected pipe is in the transmitting direction (the DIR bit is set to 1), this controller does not generate the BRDY interrupt. For details, see the description of the PIPEBRDY interrupt register.

This bit should be modified while PID is NAK and the pipe is not selected by the CURPIPE bits.

To modify this bit after completing USB communication using the selected pipe, write 1 and then 0 to ACLRM continuously to clear the FIFO buffer assigned to the selected pipe while the PID and CURPIPE bits are in the abovedescribed state.

Before modifying this bit after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

TYPE[1:0] bits (Transfer Type)

Specify the USB transfer type for the pipe selected by the PIPESEL bits (selected pipe) in the TYPE bits.

Table 30.11 lists the selected pipes and transfer types that can be set in the TYPE bits.

Table 30.11 Selected pipes and transfer types that can be set in TYPE bits

Selected pipe	TYPE bits	USB transfer type
Pipe 1 or Pipe 2	01b or 11b	Bulk or isochronous transfer
Pipe 3 to Pipe 5	01b	Bulk transfer
Pipe 6 to Pipe 9	10b	Interrupt transfer

Before setting PID to BUF for the selected pipe (before starting USB communication using the selected pipe), be sure to set these bits to the value other than 00b.

These bits should be modified while the PID bits for the selected pipe are set to NAK. Before modifying these bits after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

30.2.12.3 PIPEBUF : Pipe Buffer Specification Register

Base address: USBF = 0x8020_1000

Offset address: 0x6A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	BUFSIZE[4:0]					—	—	BUFNMB[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	BUFNMB[7:0]	Buffer number Specify the pipe FIFO buffer number for the selected pipe. (0x04 to 0x80)	R/W
9:8	—	These bits are read as 0. The write value should be 0.	R/W
14:10	BUFSIZE[4:0]	Buffer size Specify the FIFO buffer size for the pipe specified in PIPESEL bits (selected pipe). 0x00: 64 bytes 0x01: 128 bytes ⋮ 0x1F: 2 KB	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note: The bits in this register should be modified while PID is NAK and the pipe is not selected by the CURPIPE bits.
Note: Before modifying the bits of this register after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

BUFNMB[7:0] bits (Buffer number)

The first block number in the FIFO buffer to be allocated for the selected pipe should be set in these bits. The FIFO buffer blocks allocated for the selected pipe by this controller are determined as follows:

Block number: BUFNMB to block number of BUFNMB + (BUFSIZE + 1) × (DBLB + 1) − 1

These bits must be set to a value that does not exceed the size of installed memory (0 [0x00] to 8192 [0x80] for 8 KB memory). Observe the following conditions:

- 0x00 is used exclusively for DCP.
- 0x04 is used exclusively for pipe 6. However, when pipe 6 is not used, 0x04 can be used for other pipes. When pipe 6 is selected, writing to these bits has no effect and 0x04 is automatically assigned by this controller.
- 0x05 is used exclusively for pipe 7. However, when pipe 7 is not used, 0x05 can be used for other pipes. When pipe 7 is selected, writing to these bits has no effect and 0x05 is automatically assigned by this controller.
- 0x06 is used exclusively for pipe 8. However, when pipe 8 is not used, 0x06 can be used for other pipes. When pipe 8 is selected, writing to these bits has no effect and 0x06 is automatically assigned by this controller.
- 0x07 is used exclusively for pipe 9. However, when pipe 9 is not used, 0x07 can be used for other pipes. When pipe 9 is selected, writing to these bits has no effect and 0x07 is automatically assigned by this controller.

BUFSIZE[4:0] bits (Buffer size)

Specify the size of the FIFO buffer for the selected pipe in these bits in terms of blocks, where one block comprises 64 bytes. When the DBLB bit is set to 1 by software, this controller assigns two planes of the FIFO buffer size specified by the BUFSIZE bits to the selected pipe. Specifically, the following expression determines the FIFO buffer size assigned to the selected pipe by this controller.

(BUFSIZE + 1) × 64 × (DBLB + 1) [bytes]

Set the following value in these bits.

1. When pipes 1 to 5 are selected, any value from 0x00 to 0x1F can be set.
2. When pipes 6 to 9 are selected, only 0x0 should be set.

When used with CNTMD = 1, set an integral multiple of the maximum packet size to the BUFSIZE bits.

30.2.12.4 PIPEMAXP : Pipe Maximum Packet Size Register

Base address: USBF = 0x8020_1000

Offset address: 0x6C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	MXPS[10:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
10:0	MXPS[10:0]*1	Maximum Packet Size Specify the maximum data payload (maximum packet size) of the selected pipe. Pipes 6 to 9 can be set to 0x01 to 0x40 bytes.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

Note 1. The value after a reset of the MXPS bit is 0x00 when no pipe is selected with the PIPESEL bits in PIPESEL and 0x40 when a pipe is selected with the PIPESEL bits in PIPESEL.

MXPS[10:0] bits (Maximum Packet Size)

Set the maximum data payload (maximum packet size) for the selected pipe in these bits.

For pipes 1 and 2, a value between 1 byte (0x1) to 1024 bytes (0x400) can be set.

For pipes 3 to 5, any value from 8 bytes (0x8), 16 bytes (0x10), 32 bytes (0x20), 64 bytes (0x40) and 512 bytes (0x200) can be set (bits [2:0] are not provided).

The value after a reset is 0x040 (64 bytes).

These bits should be set to the appropriate value for each transfer type based on the USB specification.

Set these bits to 1 while PID is NAK and before setting the CURPIPE bits.

Before modifying these bits after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

While MXPS is 0, do not write to the FIFO buffer or set PID to BUF.

30.2.12.5 PIPEPERI : Pipe Timing Control Register

Base address: USBF = 0x8020_1000

Offset address: 0x6E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	IFIS	—	—	—	—	—	—	—	—	—	IITV[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	IITV[2:0]	Interval Error Detection Spacing Specify the interval error detection timing for the selected pipe in terms of frames, which is expressed as n-th power of 2.	R/W
11:3	—	These bits are read as 0. The write value should be 0.	R/W
12	IFIS	Isochronous IN Buffer Flush Specifies whether to flush the buffer when the pipe selected by the PIPESEL bits (selected pipe) is used for isochronous IN transfers. 0: The buffer is not flushed 1: The buffer is flushed	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

IITV[2:0] bits (Interval Error Detection Spacing)

Specify the interval error detection timing in terms of frames, which is expressed as n-th power of 2, in these bits.

These bits should be modified while PID is NAK and before setting the CURPIPE bits.

Before modifying these bits after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

Before modifying these bits after USB communication has been completed with these bits set to a certain value, set PID to NAK and then set ACLRM to 1 to initialize the interval timer.

The IITV bits are invalid for pipes 3 to 5; set these bits to 0 for these pipes.

These bits can be set when the selected pipe transfer is for isochronous transfers.

When the selected pipe is for isochronous OUT transfers

This controller generates the NRDY interrupt when it fails to receive a data packet within the interval set for (μ) frames by the IITV bits.

This controller generates the NRDY interrupt when this controller fails to receive a data packet because of a CRC error or other errors contained in the packet, or because of the FIFO buffer being full because reading data from the FIFO buffer by software (DMAC) is slow.

This controller generates the NRDY interrupt on receiving an SOF packet. Even if the SOF packet is corrupted, the internal interpolation is used and allows the interrupt to be generated at the timing to receive the SOF packet. However, when the IITV bits are set to the value other than 0, this controller generates the NRDY interrupt on receiving an SOF packet for every interval after starting interval counting operation. When the PID bits are set to NAK by software after starting the

interval timer, this controller does not generate the NRDY interrupt on receiving an SOF packet. The interval counting starts at the different timing depending on the IITV bit setting as follows.

- When IITV = 0: The interval counting starts when the PID bits for the selected pipe are set to BUF. See Figure 30.2.
- When IITV ≠ 0: The interval counting starts on completion of successful reception of the first data packet after the PID bits for the selected pipe have been modified to BUF. See Figure 30.3

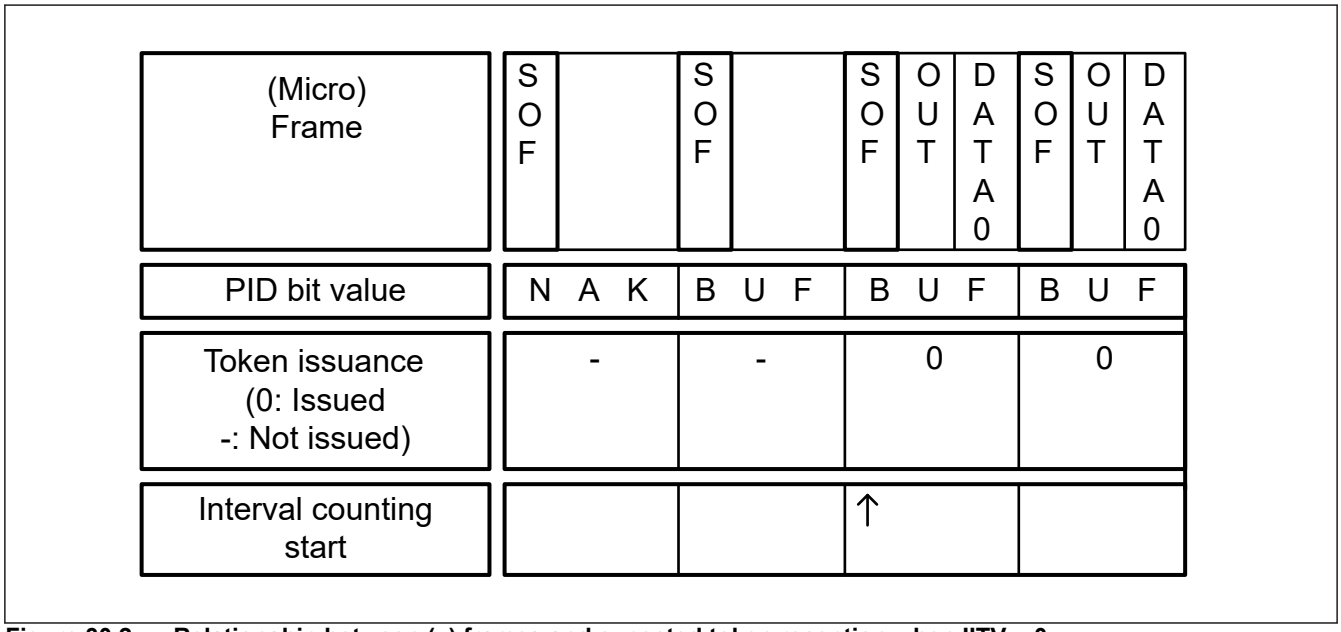


Figure 30.2 Relationship between (μ) frames and expected token reception when IITV = 0

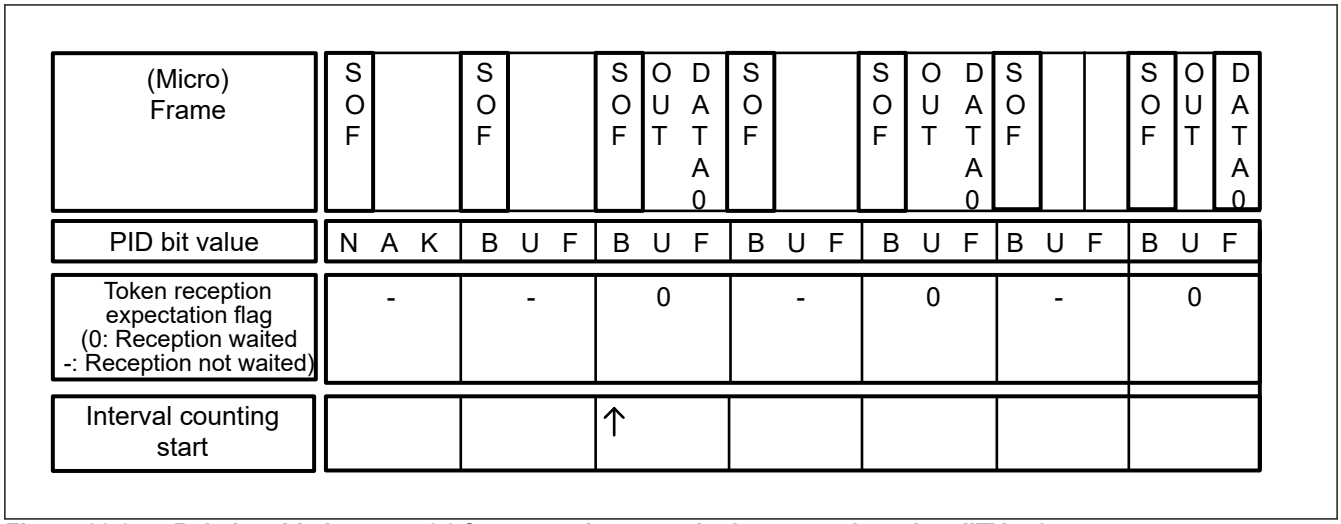


Figure 30.3 Relationship between (μ) frames and expected token reception when IITV = 1

When the selected pipe is for isochronous IN transfers

The IFIS bit should be 1 for this use. When IFIS = 0, this controller transmits a data packet in response to the received IN token irrespective of the IITV bit setting.

When IFIS = 1, this controller clears the FIFO buffer when this controller fails to receive an IN token within the interval set for (μ) frames by the IITV bits in a state in which there is data to be transmitted in the FIFO buffer.

This controller also clears the FIFO buffer when this controller fails to receive an IN token successfully because of a bus error such as a CRC error contained in the token.

This controller clears the FIFO buffer on receiving an SOF packet. Even if the SOF packet is corrupted, the internal interpolation is used and allows the FIFO buffer to be cleared at the timing to receive the SOF packet. The interval counting starts at the different timing depending on the IITV bit setting (similar to the timing during OUT transfers).

The counting conditions for the interval counter are any of the following.

- When a hardware reset is applied to this controller (at this point, the value of the IITV bits is also cleared to 0).
- When the ACLRM bit is set to 1 by software.
- When the controller detects a USB bus reset.

IFIS bit (Isochronous IN Buffer Flush)

When the selected pipe is for isochronous IN transfers, this controller automatically clears the FIFO buffer when this controller fails to receive the IN token from the USB host within the interval set by the IITV bits in terms of (μ) frames.

In double-buffer operation (DBLB = 1), this controller only clears the data in the plane used earlier.

This controller clears the FIFO buffer on receiving the SOF packet immediately after the (μ) frame in which this controller has expected to receive the IN token. Even if the SOF packet is corrupted, this controller also clears the FIFO buffer at the right timing to receive the SOF packet by using the internal interpolation.

When the selected pipe is not for the isochronous transfer, set this bit to 0.

30.2.13 Pipe Control Registers

30.2.13.1 PIPEnCTR : PIPEn Control Register (n = 1 to 5)

Base address: USBF = 0x8020_1000

Offset address: 0x070 + 0x2 × (n - 1)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSTS	INBUFM	—	—	—	ATREPM	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PID[1:0]	Response PID Specify the response type for the next transaction of the selected pipe. 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
4:2	—	These bits are read as 0. The write value should be 0.	R/W
5	PBUSY	PIPE Busy Indicates whether or not the selected pipe is being currently used for the USB bus. 0: The pipe is not being currently used for the USB bus. 1: The pipe is being currently used for the USB bus.	R
6	SQMON	Toggle Bit Confirm Sets the expected value of the sequence toggle bit in the next transaction of the selected pipe. 0: DATA0 1: DATA1	R
7	SQSET	Toggle Bit Set Specifies 1 when setting the expected value of the sequence toggle bit in the next transaction of the selected pipe, to DATA1. This bit is read as 0. When writing to this bit, only 1 can be written. 0: Writing has no effect. 1: Specifies DATA1	R/W
8	SQCLR	Toggle Bit Clear Specifies 1 when clearing the expected value of the sequence toggle bit in the next transaction of the selected pipe, to DATA0. This bit is read as 0. When writing to this bit, only 1 can be written. 0: Writing has no effect. 1: Specifies DATA0	R/W

Bit	Symbol	Function	R/W
9	ACLRM	Auto Buffer Clear Mode Enables or disables automatic buffer clear mode for the selected pipe. 0: Disabled 1: Enabled (all buffers are initialized)	R/W
10	ATREPM	Auto Response Mode Enables or disables auto response mode for the selected pipe. 0: Disabled 1: Enabled (a zero-length packet response while sending, NAK response and an NRDY interrupt is issued while receiving)	R/W
12:11	—	These bits are read as 0. The write value should be 0.	R/W
13	—	This bit is read as 0. The write value should be 1.	R/W
14	INBUFM	Transfer Buffer Monitor Indicates the selected FIFO buffer state when the selected pipe is in the transmitting direction. 0: FIFO buffer contains no transmittable data. 1: FIFO buffer contains transmittable data.	R
15	BSTS	Buffer Status Indicates the FIFO buffer state of the selected pipe. 0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note: When a USB bus reset is detected, The PID[1:0] is set to 00b and other bits are undefined.

PID[1:0] bits (Response PID)

Set a response of the controller in each pipe in this bit by the software.

The default value of this bit is NAK. When executing a USB transfer through the selected pipe, change the setting of the PID bits to select BUF. The basic operations (operations when there is no error in the communication packet) of this controller for each value of the PID bit are given in Table 30.12.

If the selected pipe is in USB communication, when the setting of the PID bits is changed from BUF to NAK by software, after writing NAK, to check whether the USB transfer of the selected pipe is actually shifted to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

In following cases, the controller modifies the value of this bit:

1. When the selected pipe is for receiving and when the software has written 1 to the SHTNAK bit of the selected pipe, the controller sets PID = NAK upon identifying the transfer end.
2. For the selected pipe, when the data packet of payload exceeding the maximum packet size is received, the controller sets PID = STALL (11b).
3. When the USB bus reset is detected, the controller sets PID = NAK.

Write 10b to shift from PID = NAK (00b) to PID = STALL.

Write 11b to shift from BUF (01b) to STALL.

First write 10b and then write 00b to shift form STALL (11b) to NAK.

First, shift to NAK and then to BUF to shift from STALL to BUF.

Table 30.12 List of controller operations according to the PID bit (1 of 2)

PID bit value	Transfer type (Type bits value)	Transfer direction (DIR bit value)	Controller operations
00b (NAK)	Bulk (TYPE = 01b) or Interrupt (TYPE = 10b)	Not dependent on setup value	NAK response is sent to the token from the USB host.
	Isochronous (TYPE = 11b)	Reception (DIR = 0)	Does not respond to the token from the USB host.
		Transmission (DIR = 1)	A zero-length packet is sent to the token from the USB host.

Table 30.12 List of controller operations according to the PID bit (2 of 2)

PID bit value	Transfer type (Type bits value)	Transfer direction (DIR bit value)	Controller operations
01b (BUF)	Bulk (TYPE = 01b)	Reception (DIR = 0)	When an OUT token is sent from the USB host, if the FIFO buffer for the selected pipe is ready for reception, the data is received and an ACK or NYET response is returned. A NAK response is returned if not ready. When a PING token is sent from the USB host, if the FIFO buffer of the selected pipe is ready for reception, an ACK response is returned. A NAK response is returned if not ready.
	Interrupt (TYPE = 10b)	Reception (DIR = 0)	For the Out token from the USB host, if the FIFO buffer of the selected pipe is ready for reception, the data is received and an ACK response is sent. A NAK response is returned if not ready.
	Bulk (TYPE = 01b) or Interrupt (TYPE = 10b)	Transmission (DIR = 1)	If the corresponding FIFO buffer is ready for transmission, the data is sent for the token from the USB. A NAK response is returned if not ready.
	Isochronous (TYPE = 11b)	Reception (DIR = 0)	For the Out token from the USB host, if the FIFO buffer of the selected pipe is ready for reception, the data is received. The data is deleted if not ready.
		Transmission (DIR = 1)	If the corresponding FIFO buffer is ready for transmission, the data is sent for the token from the USB. A zero-length packet is sent if not ready.
10b (STALL) or 11b (STALL)	Bulk (TYPE = 01b) or Interrupt (TYPE = 10)	Not dependent on setup value	A STALL response is sent to the token from the USB host.
	Isochronous (TYPE = 11b)	Not dependent on setup value	Returns nothing in response to the token from the USB host.

PBUSY bit (PIPE Busy)

This controller modifies this bit from 0 to 1 upon start of the USB transaction for the selected pipe, and modifies the bit from 1 to 0 upon completion of one transaction.

Reading this bit after PID has been set to NAK allows checking that modification of the pipe settings is possible.

SQMON bit (Toggle Bit Confirm)

This bit indicates the expected value of the sequence toggle bit of the selected pipe.

When the selected pipe is not for the isochronous transfer, this bit is toggled upon normal completion of the transaction. However, this bit is not toggled when a DATA-PID disagreement occurs during the receiving transfer.

SQSET bit (Toggle Bit Set)

Setting this bit to 1 by software allows this controller to set DATA1 as the expected value of the sequence toggle bit of the selected pipe. This bit always indicates 0.

Set this bit to 1 while PID is NAK.

Before setting this bit to 1 after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

SQCLR bit (Toggle Bit Clear)

Setting this bit to 1 by software allows this controller to set DATA0 as the expected value of the sequence toggle bit of the selected pipe. This bit always indicates 0.

Set this bit to 1 while PID is NAK.

Before setting this bit to 1 after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

ACLRM bit (Auto Buffer Clear Mode)

To clear the contents in the FIFO buffer assigned to the selected pipe completely, write 1 and then 0 to this bit continuously.

Table 30.13 shows the contents cleared by writing 1 and 0 to this bit continuously.

Table 30.14 shows the cases in which clearing the contents is necessary.

Table 30.13 Contents cleared by this controller by setting ACLRM = 1

No.	Contents cleared by ACLRM bit manipulation
(1)	All the contents in the FIFO buffer assigned to the selected pipe (all the information in two FIFO buffer planes in double-buffer operation)
(2)	The interval count value when the selected pipe is for isochronous transfer

Table 30.14 Cases where setting ACLRM = 1 is required

No.	Case where clearing is required
(1)	All the contents in the FIFO buffer assigned to the selected pipe are to be cleared.
(2)	The interval counter is to be reset.
(3)	When the BFRE setting is modified
(4)	When the DBLB setting is modified
(5)	When the transaction count function is forcibly terminated

Set this bit to 1 while PID is NAK and before specifying the pipe in CURPIPE bits. Before setting this bit to 1 after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

ATREPM bit (Auto Response Mode)

When the selected pipe is for bulk transfer, this bit can be set to 1.

When this bit is set to 1, this controller responds to the token from the USB host as described below.

1. When the selected pipe is for bulk IN transfer (when TYPE = 01b and DIR = 1)
   When ATREPM = 1 and PID = BUF, this controller transmits a zero-length packet in response to the IN token. This controller updates (toggles) the sequence toggle bit (DATAPID) each time this controller receives the ACK from the USB host (in a single transaction, IN token is received, zero-length packet is transmitted, and then ACK is received). In this case, this controller does not generate the BRDY or BEMP interrupt.
2. When the selected pipe is for bulk OUT transfer (when TYPE = 01b and DIR = 0)
   When ATREPM = 1 and PID = BUF, this controller returns NAK in response to the OUT token (or PING token) and generates the NRDY interrupt.

This bit should be modified while PID is NAK. Before setting this bit to 1 after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

For USB communication in auto response mode, set this bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode.

When the selected pipe is for isochronous transfer, be sure to set this bit to 0.

INBUFM bit (Transfer Buffer Monitor)

When the selected pipe is in the transmitting direction (DIR = 1), this controller sets this bit to 1 when writing to at least one FIFO buffer plane is completed by software (or DMAC).

This controller sets this bit to 0 when this controller completes transmitting the data from the FIFO buffer plane to which all data has been written. In double-buffer operation (DBLB = 1), this controller sets this bit to 0 after this controller has completed transmitting the data from both FIFO buffer planes but before it has completed writing data to a single FIFO buffer plane.

This bit indicates the same value as the BSTS bit when the selected pipe is in the receiving direction (DIR = 0).

BSTS bit (Buffer Status)

This bit indicates whether access from the CPU to the FIFO buffer assigned to the selected pipe is enabled or disabled. The meaning of the BSTS bit depends on the setting of the DIR, BFRE and DCLRM bits as follows.

Table 30.15 BSTS bit operations

DIR bit value	BFRE bit value	DCLRM bit value	Meaning of BSTS bit
0	0	0	1: The received data can be read from the FIFO buffer. 0: The received data has been completely read from the FIFO buffer.
		1	Setting prohibited
	1	0	1: The received data can be read from the FIFO buffer. 0: The BCLR bit has been set to 1 after the received data has been completely read from the FIFO buffer.
		1	1: The received data can be read from the FIFO buffer. 0: The received data has been completely read from the FIFO buffer.
1	0	0	1: The transmit data can be written to the FIFO buffer. 0: The transmit data has been completely written to the FIFO buffer.
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

30.2.13.2 PIPEnCTR : PIPEn Control Register (n = 6 to 9)

Base address: USBF = 0x8020_1000

Offset address: 0x07A + 0x2 × (n - 6)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BSTS	—	—	—	—	—	ACLRM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PID[1:0]	Response PID Specify the response type for the next transaction of the selected pipe. 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
4:2	—	These bits are read as 0. The write value should be 0.	R/W
5	PBUSY	PIPE Busy Indicates whether or not the selected pipe is being currently used for the USB bus. 0: The pipe is not being currently used for the USB bus. 1: The pipe is being currently used for the USB bus.	R
6	SQMON	Toggle Bit Confirm Sets the expected value of the sequence toggle bit in the next transaction of the selected pipe. 0: DATA0 1: DATA1	R
7	SQSET	Toggle Bit Set Specifies 1 while clearing the expected value of the sequence toggle bit in the next transaction of the selected pipe to DATA1.This bit is read as 0. When writing to this bit, write 1. 0: Writing has no effect. 1: Specifies DATA1	R/W

Bit	Symbol	Function	R/W
8	SQCLR	Toggle Bit Clear Specifies 1 while clearing the expected value of the sequence toggle bit in the next transaction of the selected pipe to DATA0. This bit is read as 0. When writing to this bit, write 1. 0: Writing has no effect. 1: Specifies DATA0	R/W
9	ACLRM	Auto Buffer Clear Mode Enables or disables automatic buffer clear mode for the selected pipe. 0: Disabled 1: Enabled (all buffers are initialized)	R/W
12:10	—	These bits are read as 0. The write value should be 0.	R/W
13	—	This bit is read as 0. The write value should be 1.	R/W
14	—	This bit is read as 0. The write value should be 0.	R/W
15	BSTS	Buffer Status Indicates the FIFO buffer state of the selected pipe. 0: Buffer access is disabled 1: Buffer access is enabled	R

Note: When a USB bus reset is detected, The PID[1:0] are set to 00b and other bits are undefined.

As for the bits listed below, see the description of each bit in section 30.2.13.1. PIPEnCTR : PIPEn Control Register (n = 1 to 5)

- Response PID Bits (PID[1:0])
- PIPE Busy Bit (PBUSY)
- Buffer Status Bit (BSTS)
- Sequence Toggle Bit Monitor Bit (SQMON)
- Sequence Toggle Bit Set Bit (SQSET)
- Sequence Toggle Bit Clear Bit (SQCLR)

ACLRM bit (Auto Buffer Clear Mode)

To clear all contents from the FIFO buffer allocated to the selected pipe, write 1 and 0 sequentially in the ACLRM bit.

Table 30.16 shows the contents of data to be cleared by the controller when 1 and 0 are written to this bit sequentially.

The cases that require this processing are listed in Table 30.17.

Table 30.16 Contents cleared by the controller when ACLRM = 1

No.	Contents cleared by ACLRM bit manipulation
(1)	Entire contents of the FIFO buffer allocated to the selected pipe

Table 30.17 Cases where setting ACLRM = 1 is required

No.	Cases where clearing is required
(1)	All the contents in the FIFO buffer assigned to the selected pipe are to be cleared.
(2)	The interval counter is to be reset.
(3)	The value of the BFRE bit is changed.
(4)	The transaction count function is forcibly terminated.

The ACLRM bit should be set while PID = NAK and before specifying the pipe in the CURPIPE bits.

Before modifying this bit after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY is 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

30.2.14 Transaction Counters

30.2.14.1 PIPEnTRE : PIPEn Transaction Counter Enable Register (n = 1 to 5)

Base address: USBF = 0x8020_1000

Offset address: 0x090 + 0x4 × (n - 1)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TRENB	TRCLR	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	TRCLR	Transaction Counter Clear The transaction counter can be cleared to 0 by writing 1 to this bit. This bit is read as 0. When writing to this bit, write 1. 0: Invalid 1: The current counter value is cleared	R/W
9	TRENB	Transaction Counter Enable Enable or disables the transaction counter 0: The transaction counter is disabled 1: The transaction counter is enabled	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

Note: Modify each bit in this register while PID is NAK.
Before modifying each bit after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

TRCLR bit (Transaction Counter Clear)

When this bit is set to 1 by software, this controller clears the current counter value of the transaction counter corresponding to the selected pipe and then sets this bit to 0.

TRENB bit (Transaction Counter Enable)

For the pipe in the receiving direction, setting this bit is set to 1 by software after setting the total number of the packets to be received in the TRNCNT bits allows the following control on having received the number of packets equal to the set value in the TRNCNT bits.

1. In continuous transmission/reception mode (CNTMD = 1), this module switches the FIFO buffer to the CPU side even if the FIFO buffer is not full on completion of reception.
2. While SHTNAK is 1, this module changes the setting of the PID bits to NAK for the corresponding pipe on having received the number of packets equal to the set value in the TRNCNT bits.
3. While DENDE is 1 and PKTMD is 0, the DEND signal is asserted when the number of packets specified in the TRNCNT bits is received and the last data is to be read.
4. While BFRE is 1, this module asserts the BRDY interrupt on having received the number of packets equal to the set value in the TRNCNT bits and then reading out the last received data.

For the pipe in the transmitting direction, set this bit to 0.

When the transaction counter is not used, set this bit to 0.

When the transaction counter is used, set the TRNCNT bits before setting this bit to 1. Set this bit to 1 before receiving the first packet to be counted by the transaction counter.

30.2.14.2 PIPEnTRN : PIPEn Transaction Counter Register (n = 1 to 5)

Base address: USBF = 0x8020_1000

Offset address: 0x092 + 0x4 × (n - 1)

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TRNCNT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TRNCNT[15:0]	Transaction Counter When written to: Specify the total number of reception packets (number of transactions) to be received by the selected pipe. When read from: Indicate the specified number of transactions if TRENB is 0. Indicate the number of currently counted transaction if TRENB is 1.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

TRNCNT[15:0] bits (Transaction Counter)

For the pipe in the receiving direction, setting these bits to 1 by software after setting the total number of the packets to be received in the TRNCNT bits allows the control of the transaction counter clear bit (TRCLR).

When TRENB = 0, these bits indicate the number of transactions set by software.

When TRENB = 1, these bits indicate the number of transactions being currently counted.

This controller increments the value of these bits by one when all of the following conditions are satisfied on receiving the packet.

- TRENB = 1
- (TRNCNT setting ≠ current counter value + 1) on receiving the packet
- The payload of the received packet agrees with the set value in the MXPS bits.

This module clears the value of these bits to 0 when any of the following conditions are satisfied.

When all the following conditions are satisfied:

- TRENB = 1
- (TRNCNT setting ≠ current counter value + 1) on receiving the packet
- The payload of the received packet agrees with the set value in the MXPS bits.

When both of the following conditions are satisfied:

- TRENB = 1
- A short packet is received.

When the following condition is satisfied:

- When the TRCLR bit is set to 1 by software

For the pipe in the transmitting direction, set these bits to 0. When the transaction counter is not used, set these bits to 0.

These bits should be modified while PID is NAK and TRENB is 0.

Before modifying these bits after changing the setting of the PID bits for the selected pipe from BUF to NAK, check that PBUSY = 0. However, checking of the PBUSY bit is not required if the setting of the PID bits has been changed to select NAK response by the controller.

To modify the value of these bits, set TRCLR to 1 before setting TRENB to 1.

30.2.14.3 LPSTS : Low Power Status Register

Base address: USBF = 0x8020_1000

Offset address: 0x102

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	SUSPM	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	—	These bits are read as 0. The write value should be 0.	R/W
14	SUSPM	UTMI SuspendM Control Controls the SuspendM signal to the UTMI. 0: UTMI suspend mode 1: UTMI normal mode	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.

SUSPM bit (UTMI SuspendM Control)

The USBf module controls output of the clock signal from the PLL in the USB-PHY layer. Accordingly, supply of the clock signal to the USBf module is stopped while SUSPM = 0.

After setting this bit to 1, wait for at least 100 µs to allow oscillation of the UTMI clock to become stable.

Writing to this controller is disabled when the SUSPM bit is set to 0 (the UTMI clock is stopped). Reading from the controller area is allowed. However, writing to the registers listed in Table 30.18 is enabled even when the SUSPM bit is set to 0.

Table 30.18 List of registers that can be written by software when SUSPM = 0

Address	Register name
0x8020_1000	SYSCFG0
0x8020_1002	SYSCFG1
0x8020_1032	INTENB1*1
0x8020_1102	LPSTS

Note 1. Bit 0 in the INTENB1 register can be written when SUSPM = 0.

However, the value written to the SYSCFG0 register while the UTMI clock is stopped (SUSPM = 0) is reflected after the UTMI clock oscillation is started (SUSPM = 1).

30.2.15 Next Register Set

30.2.15.1 N0SA_n, N1SA_n : Next Source Address Register n (n = 0, 1)

Base address: USBF = 0x8020_1000

Offset address: 0x400 + 0x40 × n (N0SA_n)
0x40C + 0x40 × n (N1SA_n)

Bit position:	31	0
Bit field:	SAWD[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	SAWD[31:0]	Source Address or Write Data These bits set the start address of the DMA transfer source in normal mode. These bits set write data in write-only mode.	R/W

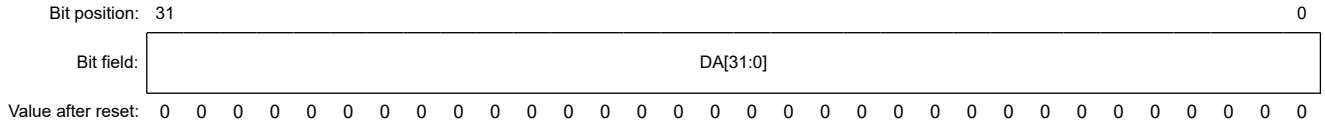
Note: When a USB bus reset is detected, this register value is undefined.
Note: The N0SA_n register is overwritten by descriptor read data during a transfer in link mode.

This register is used to set write data in write-only mode (when the WONLY field in the CHCFG_n register = 1). For details, see section 30.14.1.3. Write-Only Mode.

30.2.15.2 N0DA_n, N1DA_n : Next Destination Address Register n (n = 0, 1)

Base address: USBF = 0x8020_1000

Offset address: 0x404 + 0x40 × n (N0DA_n)
0x410 + 0x40 × n (N1DA_n)



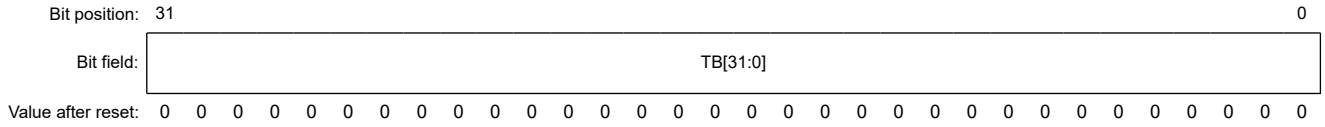
Bit	Symbol	Function	R/W
31:0	DA[31:0]	Destination Address These bits set the start address of the DMA transfer destination.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note: The N0DA_n register is overwritten by descriptor read data during a transfer in link mode.

30.2.15.3 N0TB_n, N1TB_n : Next Transaction Byte Register n (n = 0, 1)

Base address: USBF = 0x8020_1000

Offset address: 0x408 + 0x40 × n (N0TB_n)
0x414 + 0x40 × n (N1TB_n)



Bit	Symbol	Function	R/W
31:0	TB[31:0]	Transaction Byte These bits set the total number of transaction bytes. Do not start a DMA transaction with these bits set to 0.	R/W

Note: When a USB bus reset is detected, this register value is undefined.
Note: The N0TB_n register is overwritten by descriptor read data during a transfer in link mode.

30.2.16 Current Register Set

The current register set indicates the DMA transfer source address, the DMA transfer destination address, and the total number of transfer bytes. The set value is loaded from the Next 0/1 register set (in register mode) or from the descriptor read data (in link mode) by hardware. This current register set cannot be written by software.

30.2.16.1 CRSA_n : Current Source Address Register n (n = 0, 1)

Base address: USBF = 0x8020_1000

Offset address: 0x418 + 0x40 × n

Bit position:	31																												0
Bit field:	CRSA[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CRSA[31:0]	Source Address These bits indicate the read address of the next DMA transaction. The CRSA[31:0] value is automatically incremented during a DMA transaction, but remains unchanged when the SAD bit in the CHCFG_n register is 1 and is undefined when the WONLY bit in the CHCFG_n register is 1. The initial value of these bits is loaded from the following register. In register mode: The transfer source address is loaded from the Next 0/1 register set. In link mode: The transfer source address is loaded from the descriptor. (Descriptor read data is input to the N0SA_n register by hardware, and is loaded to the CRSA_n register at the beginning of transfer.) The value of these bits is incremented upon completion of a read transfer. Read this register after the ongoing DMA stops (TACT = 0 in the CHSTAT_n register). (Handle the value of these bits during DMA operation as a reference value.)	R

Note: When a USB bus reset is detected, this register value is undefined.

This register indicates the DMA transfer source address of DMA channel n (n = 1, 0).

30.2.16.2 CRDA_n : Current Destination Address Register n (n = 0, 1)

Base address: USBF = 0x8020_1000

Offset address: 0x41C + 0x40 × n

Bit position:	31																												0
Bit field:	CRDA[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CRDA[31:0]	Destination Address These bits indicate the write address of the next DMA transaction. The CRDA[31:0] value is automatically incremented during a DMA transaction, but remains unchanged when the DAD bit in the CHCFG_n register is 1. The initial value of these bits is loaded from the following register. In register mode: The transfer destination address is loaded from the Next 0/1 register set. In link mode: The transfer destination address is loaded from the descriptor. (Descriptor read data is input to the N0DA_n register by hardware, and is loaded to the CRDA_n register at the beginning of transfer.) The value of these bits is incremented upon completion of a write transfer. Read this register after the ongoing DMA stops (TACT = 0 in the CHSTAT_n register). (Handle the value of these bits during DMA operation as a reference value.)	R

Note: When a USB bus reset is detected, this register value is undefined.

This register indicates the DMA transfer destination address of DMA channel n (n = 1, 0).

30.2.16.3 CRTB_n : Current Transaction Byte Register n (n = 0, 1)

Base address: USBF = 0x8020_1000

Offset address: 0x420 + 0x40 × n

Bit position:	31																												0
Bit field:	CRTB[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CRTB[31:0]	Transaction Byte These bits indicate the remaining number of transfer bytes of the DMA transaction in progress. The CRTB[31:0] value is automatically decremented during a DMA transaction. The initial value of these bits is loaded from the following register. In register mode: The number of transfer bytes is loaded from the Next 0/1 register set. In link mode: The number of transfer bytes is loaded from the descriptor. (Descriptor read data is input to the N0TB_n register by hardware, and is loaded to the CRTB_n register at the beginning of transfer.) The value of these bits is decremented upon completion of a write transfer. Read this register after the ongoing DMA stops (TACT = 0 in the CHSTAT_n register). (Handle the value of these bits during DMA operation as a reference value.)	R

Note: When a USB bus reset is detected, this register value is undefined.

This register indicates the total number of transfer bytes of DMA channel n (n = 1, 0). This register is cleared to 0 at the end of a transfer.

30.2.17 Channel Register Set

30.2.17.1 CHSTAT_n : Channel Status Register n (n = 0, 1)

Base address: USBF = 0x8020_1000

Offset address: 0x424 + 0x40 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DNUM[7:0]								—	—	—	—	—	SWPRQ	DMARQM	INTM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	MODE	DER	DW	DL	SR	TC	END	ER	SUS	TACT	RQST	EN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EN	Enable Indicates that operation of DMA channel n is enabled or disabled. 0: Operation disabled 1: Operation enabled	R
1	RQST	Request Indicates that a transfer request has been received. 0: No DMA transfer request has been received. 1: A DMA transfer request has been received.	R
2	TACT	Transaction Active Indicates that the DMAC is operating to confirm that the channel is completely stopped. For details, see section 30.15.8. Transfer Status. 0: DMA of channel_n is inactive. 1: DMA of channel_n is in progress.	R

Bit	Symbol	Function	R/W
3	SUS	Suspend Indicates that the channel is suspended. For details, see section 30.15.8.2. Suspending a Transfer. 0: Channel_n is not suspended. 1: Channel_n is suspended.	R
4	ER	Error Indicates that an error response was received during a DMA transfer and a USB_FDMAERR interrupt was generated. 0: No error response was received. 1: An error response was received.	R
5	END	USB_FDMAn Interrupted Indicates that a DMA transaction has been completed and an USB_FDMAn interrupt was generated. 0: DMA transfer not completed 1: DMA transfer completed	R
6	TC	Terminal Count Indicates that a DMA transaction has been completed. 0: DMA transfer not completed 1: DMA transfer completed	R
7	SR	Selected Register Set Indicates the selected register set in register mode. 0: Next0 register set 1: Next1 register set	R
8	DL	Descriptor Load Indicates that the internal state is the descriptor read state. This bit remains 1 when a bus error is received during descriptor read. 0: Other than descriptor read 1: (When ER = 0) While the descriptor is read in link mode (When ER = 1) A bus error occurred while the descriptor was read in link mode.	R
9	DW	Descriptor WriteBack Indicates that the internal state is the descriptor write-back state. This bit remains 1 when a bus error is received during descriptor write-back. 0: Other than header write-back in link mode 1: (When ER in the CHSTAT_n register is 0) During header write-back in link mode (When ER in the CHSTAT_n register is 1) A bus error occurred during header write-back in link mode.	R
10	DER	Descriptor Error Indicates that the read descriptor is invalid (LV = 0). (Does not depend on the level of the DIM bit in the CHCFG_n register.) 0: No descriptor error occurred. 1: A descriptor error occurred.	R
11	MODE	DMA Mode Indicates the DMA mode that is specified by the DMS bit in the CHCFG_n register. 0: Register mode 1: Link mode	R
15:12	—	These bits are read as 0.	R
16	INTM	Interrupt Mask Indicates that the USB_FDMAn interrupt is in the temporary mask status. 0: The temporary mask status is cleared. 1: In temporary mask status	R
17	DMARQM	DMAREQ Mask Indicates that a DMA transfer request from the USB control is in the temporary mask status. 0: The temporary mask status is cleared. 1: In temporary mask status	R

Bit	Symbol	Function	R/W
18	SWPRQ	Sweep Request Indicates the forced ejection request status. Indicates the software forced ejection request (activated by the SETSSWPRQ bit in the CHCTRL_n register). 0: The forced ejection request is not asserted. 1: The forced ejection request is asserted.	R
23:19	—	These bits are read as 0.	R
31:24	DNUM[7:0]	Data Number These bits indicate the amount of valid data in the buffer. Data is read from the transfer source, and then the amount of data that has not been written to the transfer destination is displayed (in bytes).	R

This register indicates the status of DMA channel n (n = 1, 0).

When the ER bit in the CHSTAT_n register is set to 1, handle the series of transfers as invalid.

To suspend the ongoing DMA transaction, mask or clear the transfer request or clear the EN bit in the CHSTAT_n register. For details on the procedure, see section 30.15.8.3. Aborting a Transfer.

If a DMA transfer request from the USB control and a transfer request by software (by setting the STG bit in the CHCFG_n register to 1) are used together for the same channel, enabled activation sources cannot be identified. Use only a transfer request in the system.

To make a transfer request by software, make sure that the previously requested DMA transfer operation has been completed (by checking the current register or in other ways), and then set the next STG bit.

EN bit (Enable)

[Setting condition]

- 1 is written to SETEN in the CHCTRL_n register.

[Clearing conditions]

- 1 is written to SWRST in the CHCTRL_n register.
- 1 is written to CLREN in the CHCTRL_n register.
- When an error response is received during a transfer.
- When all DMA transactions are completed in register mode (REN = 0 in the CHCFG_n register)
- When DMA transfer of the last descriptor (LE = 1) is completed in link mode (write-back when WBD = 0)
- When descriptor read operation stops in link mode (LV = 0 and DRRP = 0 in the CHCFG_n register)

RQST bit (Request)

[Setting conditions]

- 1 is written to STG in the CHCTRL_n register.
- When a DMA transfer request is received from the USB control

[Clearing conditions]

- 1 is written to SWRST in the CHCTRL_n register.
- 1 is written to CLRRQ in the CHCTRL_n register.
- When a transfer is performed on the side specified by REQD in the CHCFG_n register in single transfer mode (TM = 0 in the CHCFG_n register)
- When all DMA transactions are completed in register mode (REN = 0 in the CHCFG_n register)
- When DMA transfer of the last descriptor (LE = 1) is completed in link mode
- When descriptor read operation stops in link mode (LV = 0 and DRRP = 0 in the CHCFG_n register)
- When a DMA transaction is completed in link mode with DEM in the CHCFG_n register cleared to 0
- When the master interface receives a bus error

TACT bit (Transaction Active)

[Setting condition]

- 1 is written to SETEN in the CHCTRL_n register (descriptor read start or waiting for a DMA request)

[Clearing condition]

- Idle internal state (EN in the CHSTAT_n register is cleared to 0 and all transfers are completed.)

SUS bit (Suspend)

[Setting condition]

- When the internal state becomes suspended state after 1 is written to SETSUS in the CHCTRL_n register while DMA transfer of channel n is in progress.

[Clearing conditions]

- 1 is written to CLRSUS in the CHCTRL_n register.
- 1 is written to CLREN in the CHCTRL_n register.
- Conditions for clearing EN in the CHSTAT_n register are met.

ER bit (Error)

[Setting condition]

- When an error response is received during a bus cycle.

[Clearing condition]

- 1 is written to SWRST in the CHCTRL_n register.

END bit (USB_FDMAn Interrupted)

[Setting conditions]

- Conditions for setting the TC bit are met when DEM in the CHCFG_n register is 0.
- When LV = 0 in the header, DRRP = 0 in the CHCFG_n register, and DIM = 0 during the descriptor read operation in link mode.

[Clearing conditions]

- 1 is written to CLREND in the CHCTRL_n register.
- 1 is written to SWRST in the CHCTRL_n register.

TC bit (Terminal Count)

[Setting conditions]

- When transfers for the total number of transfer bytes specified in the CRTB register are completed in register mode.
- When transfers for the total number of transfer bytes specified in the CRTB register are completed in link mode while WBD = 1 in the descriptor's header.
- When descriptor write-back is completed in link mode while WBD = 0 in the descriptor's header.

[Clearing conditions]

- 1 is written to CLRTC in the CHCTRL_n register.
- 1 is written to SWRST in the CHCTRL_n register.

SR bit (Selected Register Set)

[Setting condition]

- 1 is written to the RSEL bit in the CHCFG_n register to set it.

[Clearing condition]

- 0 is written to the RSEL bit in the CHCFG_n register to clear it.

DL bit (Descriptor Load)

[Setting condition]

- When descriptor read starts in link mode

[Clearing conditions]

- Descriptor read is completed with an OK response in link mode.
- 1 is written to SWRST in the CHCTRL_n register. If 1 is retained with an error response, this bit can be cleared to 0 only with the SWRST bit.

DW bit (Descriptor WriteBack)

[Setting condition]

- When header write-back starts in link mode.

[Clearing conditions]

- Header write-back is completed with an OK response in link mode.
- 1 is written to SWRST in the CHCTRL_n register. If 1 is retained with an error response, this bit can be cleared to 0 only with the SWRST bit.

DER bit (Descriptor Error)

[Setting condition]

- LV of the read descriptor is 0 when DRRP in the CHCFG_n register is 0 in link mode.

[Clearing conditions]

- 1 is written to the CLRDER bit in the CHCTRL_n register.
- 1 is written to the SWRST bit in the CHCTRL_n register.

INTM bit (Interrupt Mask)

[Setting condition]

- 1 is written to the SETINTM bit in the CHCTRL_n register.

[Clearing conditions]

- 1 is written to the CLRINTM bit in the CHCTRL_n register.
- 1 is written to the SWRST bit in the CHCTRL_n register.

DMARQM bit (DMAREQ Mask)

[Setting condition]

- 1 is written to the SETDMARQM bit in the CHCTRL_n register to set it.

[Clearing conditions]

- Header write-back in link mode is completed with an OK response.
- 1 is written to the CLRDMARQM bit in the CHCTRL_n register.
- 1 is written to the SWRST bit in the CHCTRL_n register.

SWPRQ bit (Sweep Request)

[Setting condition]

- When SETSSWPRQ in the CHCTRL_n register is asserted

[Clearing conditions]

- The amount of data in the buffer becomes 0 because of forced ejection.

- 1 is written to the CLRHSWPRQM bit in the CHCTRL_n register.
- 1 is written to the SWRST bit in the CHCTRL_n register.

DNUM[7:0] bits (Data Number)

[Increment condition]

- When DMA read transfer is completed.

[Decrement condition]

- When DMA write transfer is completed.

[Clearing conditions]

- Conditions for clearing the EN bit are met.
- 1 is written to the SWRST bit in the CHCTRL_n register.

30.2.17.2 CHCTRL_n : Channel Control Register n (n = 0, 1)

Base address: USBF = 0x8020_1000

Offset address: 0x428 + 0x40 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	CLRDMARQM	SETDMARQM	CLRINTM	SETINTM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	SETSWPRQ	—	SETREN	—	—	CLRSUS	SETSUS	CLRDER	CLRTC	CLRENDD	CLRRQ	SWRST	STG	CLREN	SETEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SETEN	Set Enable Enables DMA transfer of DMA channel n. If this bit is set together with the SWRST bit, clearing by the SWRST bit takes precedence, and a transfer does not start. 0: No effect 1: DMA transfer is enabled. (The EN bit in the CHSTAT_n register is set)	W
1	CLREN	Clear Enable Clears the EN bit in the CHSTAT_n register. For details, see section 30.15.8.3. Aborting a Transfer. 0: No effect 1: DMA transfer is disabled. (The EN bit in the CHSTAT_n register is cleared)	W
2	STG	Software Trigger Activates the software (setting the internal transfer request). If this bit is set together with the SWRST bit, clearing by the SWRST bit takes precedence. 0: No effect 1: A transfer request is set by software. (The RQST bit in the CHSTAT_n register is set)	W
3	SWRST	Software Reset Clears each bit in the CHSTAT_n register. (For details on bits to be cleared, see descriptions of each bit.) Set this bit to 1 when the EN bit and the TACT bit are 0. 0: No effect 1: Clears each bit in the CHSTAT_n register	W

Bit	Symbol	Function	R/W
4	CLRRQ	Clear Request Clears the RQST bit in the CHSTAT_n register. 0: No effect 1: Clears the RQST bit in the CHSTAT_n register	W
5	CLREND	Clear End Clears the END bit in the CHSTAT_n register, and clears USB_FDMAn to low level. 0: No effect 1: Clears the END bit in the CHSTAT_n register	W
6	CLRTC	Clear TC Clears the TC bit in the CHSTAT_n register. 0: No effect 1: Clears the TC bit in the CHSTAT_n register	W
7	CLRDER	Clear DER Clears the DER bit in the CHSTAT_n register, and clears USB_FDMAn to low level. 0: No effect 1: Clears the DER bit in the CHSTAT_n register	W
8	SETSUS	Set Suspend Setting this bit to 1 when the EN bit in the CHSTAT_n register is 1 suspends the ongoing DMA transfer. 0: No effect 1: Suspends the ongoing DMA transfer	W
9	CLRSUS	Clear Suspend Setting this bit to 1 when the SUS bit in the CHSTAT_n register is 1 clears the suspended state. 0: No effect 1: Clears the suspended status of the ongoing DMA transfer	W
11:10	—	The write value should be 0.	W
12	SETREN	Set Register Set Enable Sets the REN bit in the CHCFG_n register. 0: No effect 1: Sets the REN bit in the CHCFG_n register to 1	W
13	—	The write value should be 0.	W
14	SETSSWPRQ	Set Software Sweep Request Forcibly ejects the data in the buffer to the transfer destination. See section 30.15.3. Forced Ejection Request. If the transfer destination asserts a hardware request (REQD = 1), forced ejection is not available (disabled by hardware). 0: No effect 1: Writes the buffer data (that has not been written to the transfer destination) to the transfer destination	W
15	—	The write value should be 0.	W
16	SETINTM	Set Interrupt Mask Temporarily masks USB_FDMAn and sets the INTM bit in the CHSTAT_n register to 1. 0: No effect 1: Masks USB_FDMAn	W
17	CLRINTM	Clear Interrupt Mask Clears the NT_DMA [n] pin output mask status, and clears the INTM bit in the CHSTAT_n register to 0. 0: No effect 1: Clears the mask status set by the SETINTM bit	W
18	SETDMARQM	SET DMAREQ Mask Temporarily masks a DMA transfer request from the USB control, and sets the DMARQM bit in the CHSTAT_n register to 1. 0: No effect 1: Masks a DMA transfer request from the USB control	W

Bit	Symbol	Function	R/W
19	CLRDMARQM	Clear DMAREQ Mask Clears the temporary mask status of the DMA transfer request from the USB control, and clears the DMARQM bit in the CHSTAT_n register to 0. 0: No effect 1: Clears the mask status set by the SETDMARQM bit	W
31:20	—	The write value should be 0.	W

The CHCTRL_n register controls DMA transfer of DMA channel n (n = 1, 0). This register is used to activate each function, and it does not retain written values.

30.2.17.3 CHCFG_n : Channel Configuration Register n (n = 0, 1)

Base address: USBF = 0x8020_1000

Offset address: 0x42C + 0x40 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DMS	REN	RSW	RSEL	SBE	DIM	—	DEM	WONLY	TM	DAD	SAD	DDS[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SDS[3:0]				DRRP	AM[2:0]			—	LVL	HIEN	LOEN	REQD	—	—	SEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SEL	Terminal Select Selects the FIFO channel to be used on the USB control side. 0: D0FIFO is used 1: D1FIFO is used	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	REQD	Request Direction Specifies whether the USB control side is source or destination. 0: The USB control side is source (initial value) 1: The USB control side is destination	R/W
4	LOEN	Sets the transfer request signal between the USB control and the DMAC. Always set this bit to 0.	R/W
5	HIEN	Sets the transfer request signal between the USB control and the DMAC. Always set this bit to 1.	R/W
6	LVL	Sets the transfer request signal between the USB control and the DMAC. Always set this bit to 1.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	AM[2:0]	These bits set the transfer request signal between the USB control and the DMAC. Always set these bits to 000b.	R/W
11	DRRP	Descriptor Read Repeat Switches operation when LV in the header is 0 after reading the descriptor. (See Figure 30.22.) 0: Sets the DER bit in the CHSTAT_n register to stop operation (initial value). 1: Reads the same descriptor continuously until LV becomes 1. When LV = 1, DMA transfer using the descriptor value starts. The descriptor read interval is controlled by the DSCITVL register.	R/W

Bit	Symbol	Function	R/W
15:12	SDS[3:0]	Source Data Size These bits set the DMA transfer size. When SDS[3] is set to 0, transfer mode is changed to normal mode. When SDS[3] is set to 1, transfer mode is changed to skip mode. Transfer size is set by SDS[2:0]. (For settable values, see Table 30.19.) 0 0 0: 8 bits (initial value) 0 0 1: 16 bits 0 1 0: 32 bits 0 1 1: 64 bits 1 0 0: 128 bits 1 0 1: 256 bits 1 1 0: 512 bits 1 1 1: 1024 bits	R/W
19:16	DDS[3:0]	Destination Data Size These bits set the DMA transfer size. When the USB control side is transfer destination, use the normal mode. When DDS[3] is set to 0, transfer mode is changed to normal mode. When SDS[3] is set to 1, transfer mode is changed to skip mode. Transfer size is set by DDS[2:0]. (For settable values, see Table 30.19.) 0 0 0: 8 bits (initial value) 0 0 1: 16 bits 0 1 0: 32 bits 0 1 1: 64 bits 1 0 0: 128 bits 1 0 1: 256 bits 1 1 0: 512 bits 1 1 1: 1024 bits	R/W
20	SAD	Source Address Direction Sets the count direction of the transfer source address of DMA channel n. When the USB control side is transfer source, set this bit to 1 (fixed). When using the skip mode on the transfer source side or when the transfer source is beat-unaligned, do not specify SAD = 1 (fixed). 0: Increment (initial value) 1: Fixed	R/W
21	DAD	Destination Address Direction Sets the count direction of the transfer destination address of DMA channel n. When the USB control side is transfer destination, set this bit to 1 (fixed). When using the skip mode on the transfer destination side or when the transfer destination is beat-unaligned, do not specify DAD = 1 (fixed). 0: Increment (initial value) 1: Fixed	R/W
22	TM	Sets the transfer request signal between the USB control and the DMAC. Always set this bit to 0.	R/W
23	WONLY	Write Only Mode Sets the write-only mode. For details, see section 30.14.1.3. Write-Only Mode. 0: Normal operation (initial value) 1: Write-only mode	R/W
24	DEM	USB_FDMAn Mask Masks USB_FDMAn interrupt detection. When this bit is 1 at USB_FDMAn timing, USB_FDMAn and END in the CHSTAT_n register are not asserted. In register mode, the DEM bit is automatically cleared to 0. In link mode, this bit is not cleared. 0: Does not mask USB_FDMAn interrupt detection (initial value) 1: Masks USB_FDMAn interrupt detection	R/W
25	—	This bit is read as 0. The write value should be 0.	R/W
26	DIM	Descriptor Interrupt Mask Sets masking for USB_FDMAn when LV = 0 after the descriptor's header is read. 0: Does not mask USB_FDMAn (initial value) 1: Masks USB_FDMAn	R/W

Bit	Symbol	Function	R/W
27	SBE	Sweep Buffer Enable Selects whether to sweep (by writing) the read data in the buffer before aborting the transfer when EN in the CHSTAT_n register is cleared to 0 during a DMA transaction. Sweep mode can be used only when REQD = 0. 0: Does not sweep buffer data and aborts the transfer (initial value) 1: Sweeps buffer data and aborts the transfer	R/W
28	RSEL	Register Set Select Selects the next register set to be executed next. This bit is valid only in register mode. When RSW = 1, this bit is automatically reversed upon completion of a DMA transaction. 0: Executes Next0 register set (initial value) 1: Executes Next1 register set	R/W
29	RSW	Register Select Switch The RSEL bit is automatically reversed upon completion of a DMA transaction. This bit is valid only in register mode. 0: Does not reverse the RSEL bit upon completion of a DMA transaction (initial value) 1: Reverses the RSEL bit upon completion of a DMA transaction	R/W
30	REN	Register Set Enable Performs DMA transaction of the next register set selected by the RSEL bit continuously after the current DMA transaction is completed. This bit is valid only in register mode. 0: Does not perform DMA transaction continuously 1: Performs DMA transaction continuously	R/W
31	DMS	DMA Mode Select Sets DMA mode. 0: Register mode (initial value) 1: Link mode	R/W

The CHCFG_n register controls DMA transfer of DMA channel n (n = 1, 0).

The settable size ranges of the SDS[2:0] and DDS[2:0] bits depend on the data bus width, number of implemented buffers, and alignment type of addresses to be accessed (beat-aligned or beat-unaligned). The following table shows the settable size ranges. These ranges are specified to prevent this controller from malfunctioning. Set appropriate values according to the module to be accessed.

Table 30.19 Settable size ranges of SDS[2:0] and DDS[2:0] bits

Data bus width [bit]	Number of buffers	Transfer address	Settable values of SDS[2:0] and DDS[2:0] bits	
32	16	Beat-aligned	8 to 32 bits 128 to 512 bits	(000b to 010b) (100b to 110b)
		Beat-unaligned	8 to 32 bits 128 to 256 bits	(000b to 010b) (100b to 101b)
		Beat-unaligned	8 to 32 bits	(000b to 010b)

If both transfer source and destination or either of these is beat-unaligned, set the SDS[2:0] and DDS[2:0] bits to values within the settable ranges for the beat-unaligned addresses from among the CHCFG_n register settings.

- Even when beat-aligned setting is made at the beginning of DMA transaction, if a skip transfer is used, beat-aligned setting may change to beat-unaligned setting during the DMA transaction. In this case, make beat-unaligned setting from the beginning.
- If the software cannot distinguish beat-aligned or beat-unaligned setting, use the beat-unaligned setting range.

DEM bit (USB_FDMAn Mask)

[Clearing condition]

A DMA transaction is completed when DEM = 1.

RSEL bit (Register Set Select)

[Transition condition]

A DMA transaction is completed when RSW = 1.

REN bit (Register Set Enable)

[Setting conditions]

- 1 is written to this bit.
- 1 is written to the SETREN bit in the CHCTRL_n register.

[Clearing conditions]

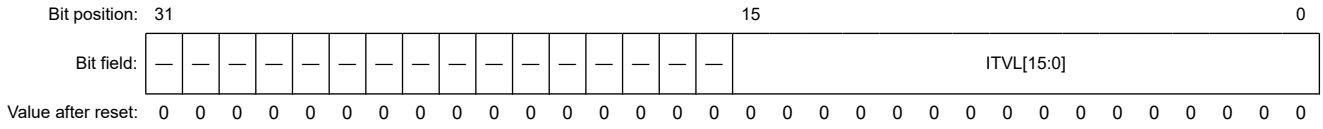
- 0 is written to this bit.
- A DMA transaction is completed when REN = 1.

To re-set the REN bit during a DMA transaction, use the SETREN bit in the CHCTRL_n register.

30.2.17.4 CHITVL_n : Channel Interval Register n (n = 0, 1)

Base address: USBF = 0x8020_1000

Offset address: 0x430 + 0x40 × n



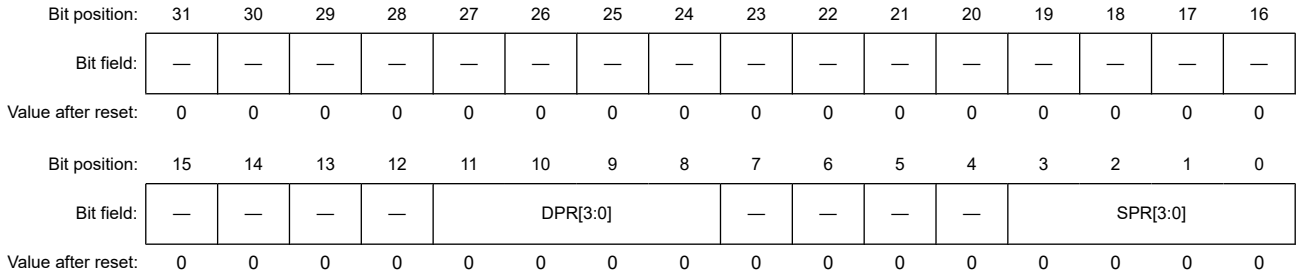
Bit	Symbol	Function	R/W
15:0	ITVL[15:0]	Interval These bits set the DMA transfer interval.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The CHITVL_n register sets the DMA transfer interval of DMA channel n (n = 1, 0).

30.2.17.5 CHEXT_n : Channel Extension Register n (n = 0, 1)

Base address: USBF = 0x8020_1000

Offset address: 0x434 + 0x40 × n



Bit	Symbol	Function	R/W
3:0	SPR[3:0]	Source PROT These bits set the value to be output to the MHPROT[3:0] pins during DMA read transfer.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
11:8	DPR[3:0]	Destination PROT These bits set the value to be output to the MHPROT[3:0] pins during DMA write transfer.	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The CHEXT_n register is an extension register for DMA channel n.

30.2.18 Link Register Set

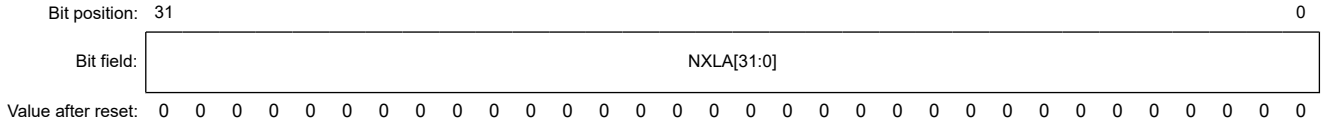
The link register set is used to set and display the descriptor address in link mode.

When the descriptor address is set in the NXLA_n register by software to activate the DMAC, the NXLA_n register value is loaded to the CRLA_n register by hardware and the descriptor is read. The DMAC starts a DMA transaction according to the read descriptor value. The NXLA_n register value is automatically updated by the next link address value in the read descriptor to be used as the descriptor address for the next DMA transaction.

30.2.18.1 NXLA_n : Next Link Address Register n (n = 0, 1)

Base address: USBF = 0x8020_1000

Offset address: 0x438 + 0x40 × n



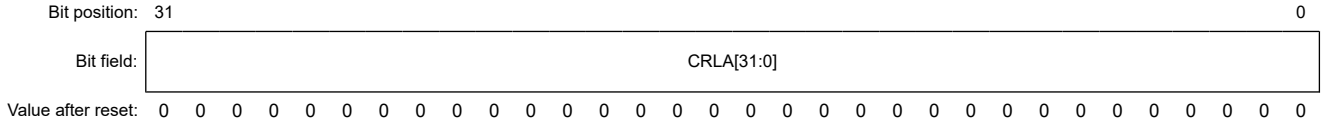
Bit	Symbol	Function	R/W
31:0	NXLA[31:0]	Next Link Address These bits set the address of the link destination. Because the two lower-order bits are fixed to 0, only word-aligned addresses can be set.	R/W

The NXLA_n register sets the link address of DMA channel n (n = 1, 0).

30.2.18.2 CRLA_n : Current Link Address Register n (n = 0, 1)

Base address: USBF = 0x8020_1000

Offset address: 0x43C + 0x40 × n



Bit	Symbol	Function	R/W
31:0	CRLA[31:0]	Current Link Address These bits display the address of the descriptor that is being executed.	R

The CRLA_n register displays the link address of DMA channel n (n = 1, 0).

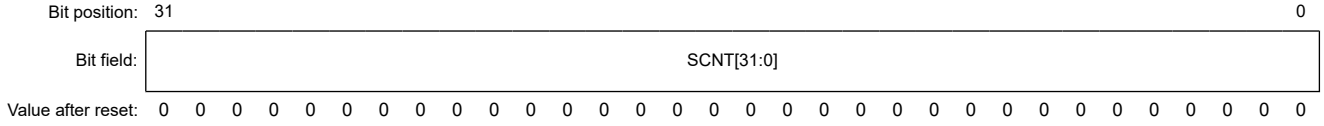
30.2.19 Skip Register Set

The skip register set is used for settings for skip (scatter/gather) transfer.

30.2.19.1 SCNT_n : Source Continuous Register n (n = 0, 1)

Base address: USBF = 0x8020_1000

Offset address: 0x600 + 0x20 × n



Bit	Symbol	Function	R/W
31:0	SCNT[31:0]	Source Continuous These bits set the size (in bytes) of continuous access space for access to the transfer source address.	R/W

The SCNT_n register sets the space size for continuous access to the transfer source address of DMA channel n (n = 1, 0).

Use this register together with the SSKP_n register (see Figure 30.4).

When using this mode, set the SDS[3] bit in the CHCFG_n register to 1.

When performing a skip transfer on the transfer source side, do not set the SAD bit in the CHCFG_n register to 1 (fixed). In addition, do not perform a skip transfer when SCNT_n is set to 0x00000000.

30.2.19.2 SSKP_n : Source Skip Register n (n = 0, 1)

Base address: USBF = 0x8020_1000

Offset address: 0x604 + 0x20 × n

Bit position:	31																												0
Bit field:	SSKP[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SSKP[31:0]	Source Skip These bits set the skip amount (in bytes) during access to the source address.	R/W

The SSKP_n register sets the skip amount for skipping the next source address after accessing the data size set by the SCNT_n register during access to the source address of DMA channel n (n = 1, 0).

Use this register together with the SCNT_n register (see Figure 30.4).

When using this mode, set the SDS[3] bit in the CHCFG_n register to 1.

When performing a skip transfer on the transfer source side, do not set the SAD bit in the CHCFG_n register to 1 (fixed).

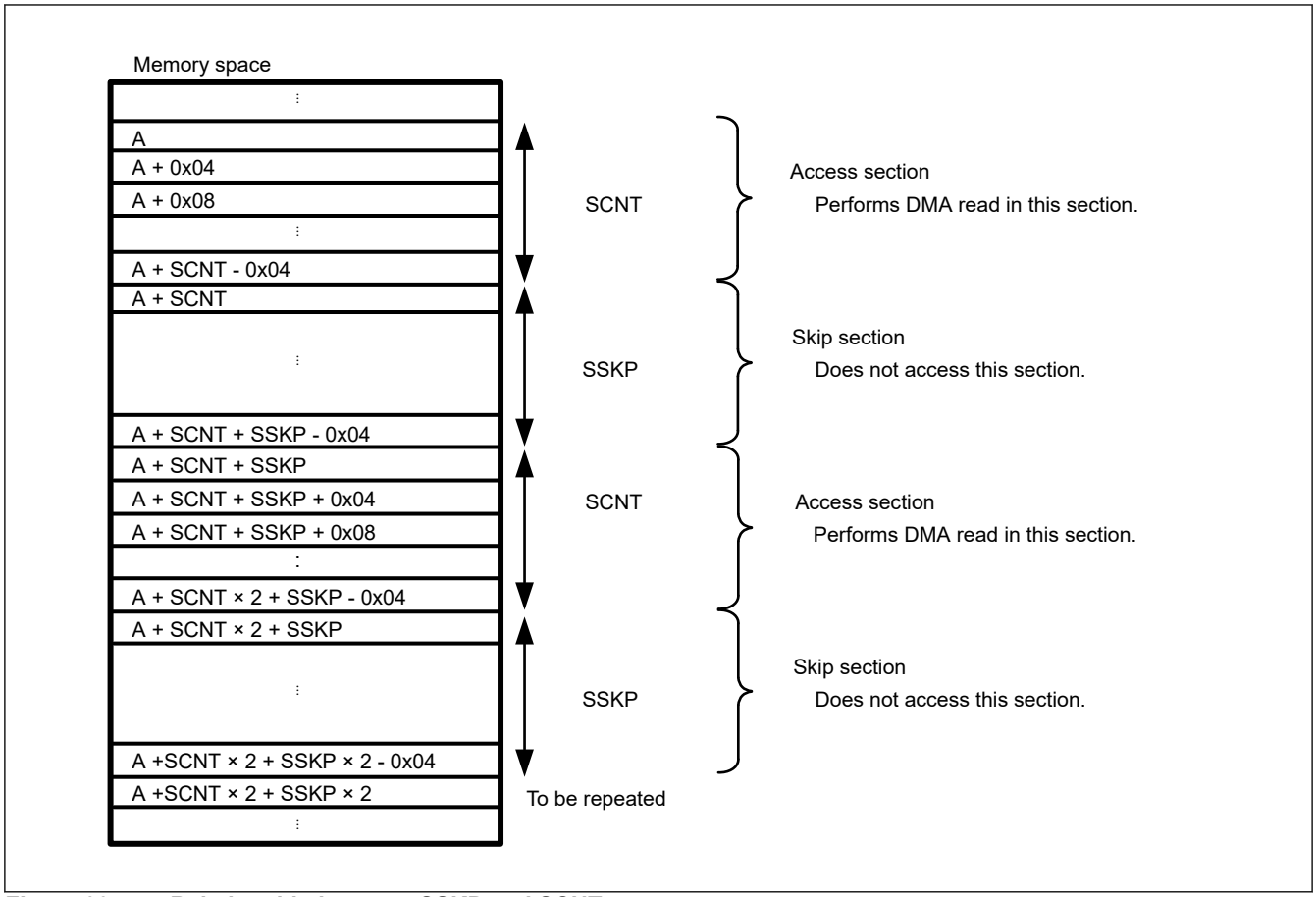


Figure 30.4 Relationship between SSKP and SCNT

The SCNT and SSKP_n values can be set regardless of the source address and the set value of the SDS field in the CHCFG_n register. The DMAC accesses a memory space with the size set in the SDS field, and acquires only valid data into the buffer. For details, see section 30.16.1.1. Read Access.

30.2.19.3 DCNT_n : Destination Continuous Register n (n = 0, 1)

Base address: USBF = 0x8020_1000

Offset address: 0x608 + 0x20 × n

Bit position:	31																												0
Bit field:	DCNT[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DCNT[31:0]	Destination Continuous These bits set the space size (in bytes) for continuous access during an access to the destination address.	R/W

The DCNT_n register sets the space size for continuous access during an access to the destination address of DMA channel n (n = 1, 0).

Use this register together with the DSKP_n register (see Figure 30.5).

When using this mode, set the DDS[3] bit in the CHCFG_n register to 1.

When performing a skip transfer on the transfer destination side, do not set the DAD bit in the CHCFG_n register to 1 (fixed). In addition, do not perform a skip transfer when DCNT is set to 0x00000000.

30.2.19.4 DSKP_n : Destination Skip Register n (n = 0, 1)

Base address: USBF = 0x8020_1000

Offset address: 0x60C + 0x20 × n

Bit position:	31																												0
Bit field:	DSKP[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DSKP[31:0]	Destination Skip These bits set the skip amount (in bytes) during access to the destination address.	R/W

The DSKP_n register sets the skip amount for skipping the next destination address after accessing the data size set by the DCNT_n register during access to the destination address of DMA channel n (n = 1, 0).

Use this register together with the DCNT_n register (see Figure 30.5).

When using this mode, set the DDS[3] bit in the CHCFG_n register to 1.

When performing a skip transfer on the transfer destination side, do not set the DAD bit in the CHCFG_n register to 1 (fixed).

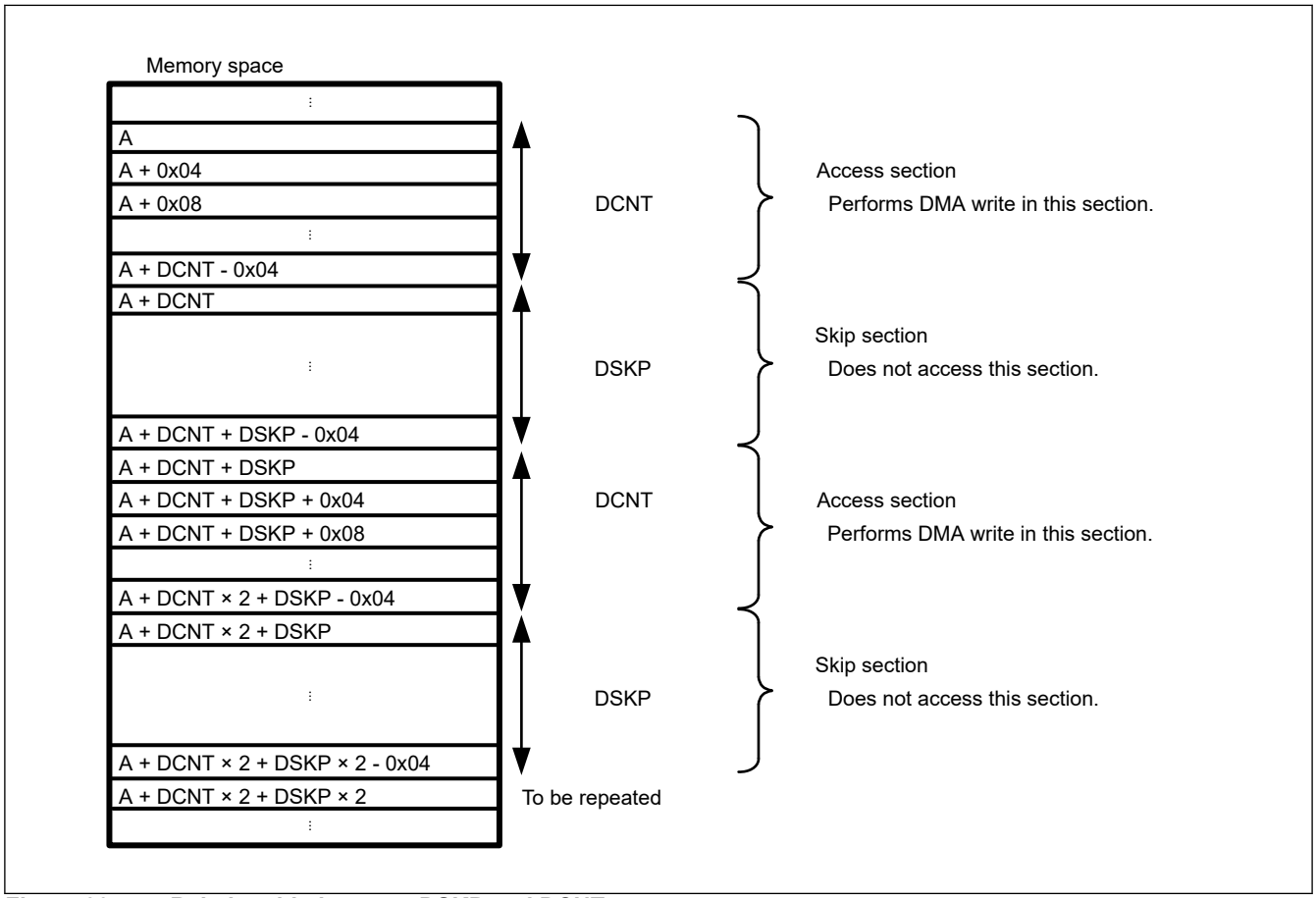


Figure 30.5 Relationship between DSKP and DCNT

The DCNT and DSKP_n values can be set regardless of the destination address and the set value of the DDS field in the CHCFG_n register. The DMAC performs write access only to the specified space with the size equal to or smaller than the value set in the DDS field. For details, see section 30.16.1.2. Write Access.

30.2.20 DMA Register Set

The following registers are common to all channels.

30.2.20.1 DCTRL : DMA Control Register

Base address: USBF = 0x8020_1000

Offset address: 0x700

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	LWPR[3:0]				—	—	—	—	LDPR[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PR	Priority Sets transfer priority control mode. See section 30.15.2. DMA Channel Priority Control. 0: Fixed priority mode 1: Round-robin mode	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
19:16	LDPR[3:0]	Link Descriptor PROT These bits set the value to be output to the MHPROT[3:0] pins during descriptor read in link mode.	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
27:24	LWPR[3:0]	Link WriteBack PROT These bits set the value to be output to the MHPROT[3:0] pins during descriptor writeback in link mode.	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The DCTRL register sets the transfer type for descriptor access and the arbitration between channels.

30.2.20.2 DSCITVL : Descriptor Interval Register

Base address: USBF = 0x8020_1000

Offset address: 0x704

Bit position:	31															15								8							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DITVL[7:0]								—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
15:8	DITVL[7:0]	Descriptor Interval These bits set the descriptor read interval. The descriptor is read again with an interval of (DITVL × 256) cycles.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The DSCITVL register sets the descriptor read interval.

By setting the DRRP bit in the CHCFG_n register to 1, the descriptor is continuously read until it reaches LV = 1. This register sets the read interval.

30.2.20.3 DSTAT_EN : DMA Status EN Register

Base address: USBF = 0x8020_1000

Offset address: 0x710

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EN1	EN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EN0	Channel 0 EN Displays the EN bit status of DMA channel 0.	R
1	EN1	Channel 1 EN Displays the EN bit status of DMA channel 1.	R
31:2	—	These bits are read as 0.	R

The DSTAT_EN register displays the EN bit status of all channels.

Writing data to this register does not change the value of each bit.

30.2.20.4 DSTAT_ER : DMA Status ER Register

Base address: USBF = 0x8020_1000

Offset address: 0x714

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ER1	ER0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ER0	Channel 0 ER Displays the ER bit status of DMA channel 0.	R
1	ER1	Channel 1 ER Displays the ER bit status of DMA channel 1.	R
31:2	—	These bits are read as 0.	R

The DSTAT_ER register displays the ER bit status of all channels.

Writing data to this register does not change the value of each bit.

30.2.20.5 DSTAT_END : DMA Status END Register

Base address: USBF = 0x8020_1000

Offset address: 0x718

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	END1	END0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	END0	Channel 0 END Displays the END bit status of DMA channel 0.	R
1	END1	Channel 1 END Displays the END bit status of DMA channel 1.	R
31:2	—	These bits are read as 0.	R

The DSTAT_END register displays the END bit status of all channels.

Writing data to this register does not change the value of each bit.

30.2.20.6 DSTAT_TC : DMA Status TC Register

Base address: USBF = 0x8020_1000

Offset address: 0x71C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TC1	TC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TC0	Channel 0 TC Displays the TC bit status of DMA channel 0.	R
1	TC1	Channel 1 TC Displays the TC bit status of DMA channel 1.	R
31:2	—	These bits are read as 0.	R

The DSTAT_TC register displays the TC bit status of all channels.

Writing data to this register does not change the value of each bit.

30.2.20.7 DSTAT_SUS : DMA Status SUS Register

Base address: USBF = 0x8020_1000

Offset address: 0x720

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SUS1	SUS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SUS0	Channel 0 SUS Displays the SUS bit status of DMA channel 0.	R
1	SUS1	Channel 1 SUS Displays the SUS bit status of DMA channel 1.	R
31:2	—	These bits are read as 0.	R

The DSTAT_SUS register displays the SUS bit status of all channels.

Writing data to this register does not change the value of each bit.

30.3 Operation

30.3.1 System Controls and Oscillation Controls

This section describes the register operations required for initial settings of this module, and the registers necessary for power consumption control. The startup sequence is as follows.

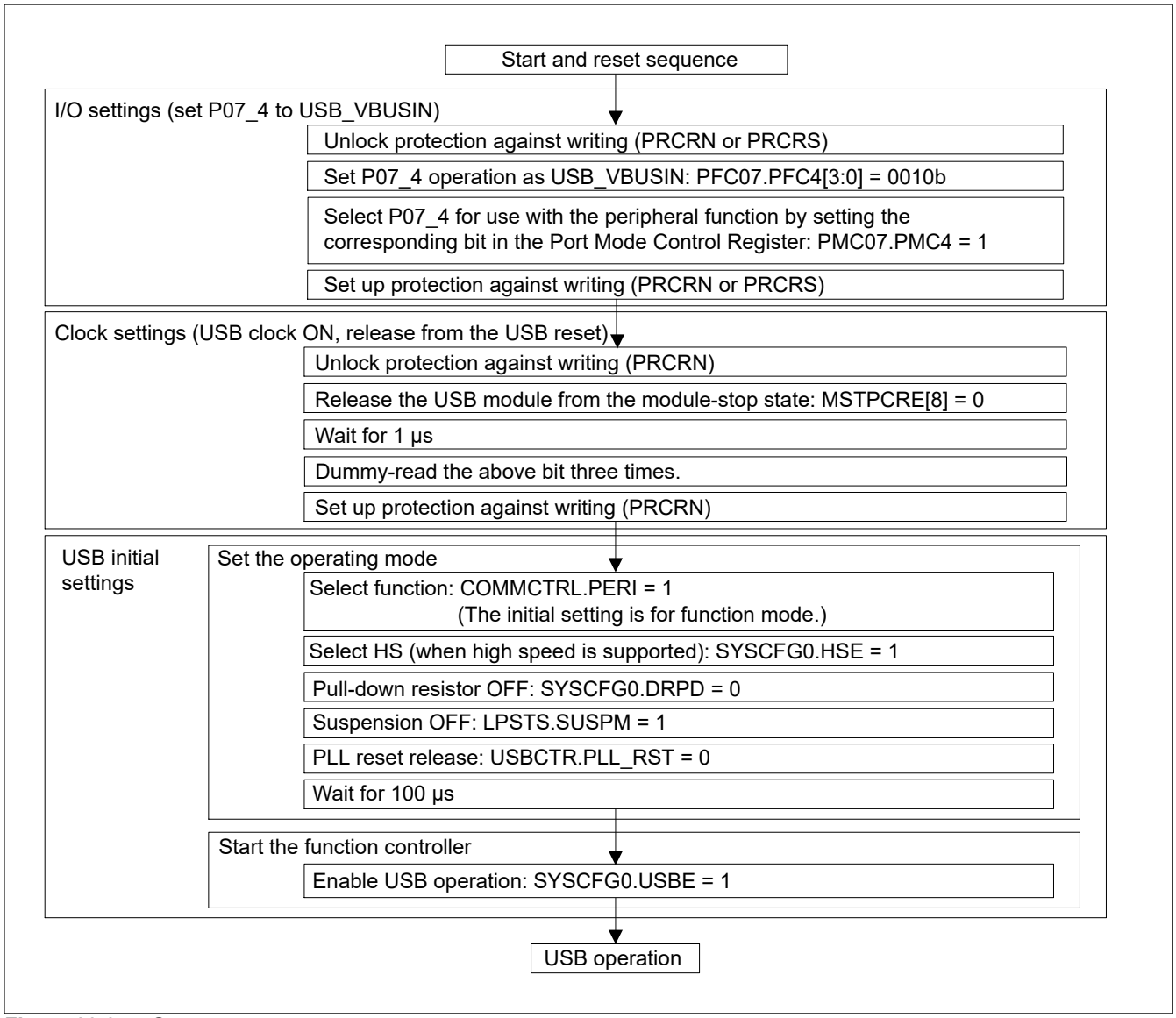


Figure 30.6 Startup sequence

30.3.2 Resets

Table 30.20 lists the types of reset for this controller. For the initial states of the registers following the reset operations, see .

Table 30.20 Types of reset

Name	Operation
Hardware reset	Low level input from the RES# pin
USB bus reset	Automatically detected by this module from the D+ and D- lines

30.3.3 USB Data Bus Resistor Controller

This controller controls the pull-up resistors for the D+ signal of the USB2.0-PHY port.

Make pull-up setting for the D+ signal using the DPRPU and DRPD bits of the SYSCFG0 register.

Confirm the connection to the USB Host before setting the DPRPU bit of the SYSCFG0 register to 1 to pull up D+.

This controller incorporates termination resistors for the D+ and D− signals (in high-speed operation mode) and output resistors (in full-speed operation mode). The switching of the internal resistors following the connection to the PC is automatically done by the controller during reset handshake, suspend, and resume processing.

If the DPRPU bit of the SYSCFG0 register is set to 0 during communication with the PC, the controller disables the pull-up resistors (or termination resistors) of the USB data lines, so that it can notify the host controller of the disconnection of the device.

30.3.4 Notes on Stopping Clocks

- The PHY clock can be stopped with the SUSPM register in suspend mode.
- If a clock is to be stopped while the controller is in USB suspended state, however, it is necessary to resume the supply of the clock during resume processing. The resumption of the PHY clock need be accomplished within 5.5 ms after a resume interrupt occurs.
- When the USB function is not in use, place the USB module in the module-stop state (MSTPCRE[8] = 1).

30.4 Interrupt Functions

30.4.1 Overview of Interrupt Functions

Table 30.21 lists the interrupt functions of this controller.

Table 30.21 List of interrupt functions

Bit	Interrupt name	Interrupt source	Related status
VBINT	VBUS interrupt	When a change in the state of the USB_VBUSIN input pin has been detected (low to high or high to low).	VBSTS
RESM	Resume interrupt	When a change in the state of the USB bus has been detected in the suspended state (J-state to K-state or J-state to SE0)	—
SOFR	Frame number update interrupt	If SOFRM = 0: When an SOF packet with a different frame number is received. If SOFRM = 1: When an SOF packet for a µ frame number of 0 cannot be received due to a corrupted packet.	—
DVST	Device state transition interrupt	When transition in device state is detected: A USB bus reset detected The suspended state detected Set Address request received Set Configuration request received	DVSQ
CTRT	Control transfer stage transition interrupt	When a stage transition is detected in control transfer: Setup stage completed Control write transfer status stage transition Control read transfer status stage transition Control transfer completed A control transfer sequence error occurred	CTSQ
BEMP	Buffer empty interrupt	When the buffer becomes empty after all data in the buffer memory has been transmitted. When a packet exceeding the maximum packet size is received.	PIPEBEMP
NRDY	Buffer not ready interrupt	When a token is received when PID = BUF and the buffer memory is not ready to transmit or receive data. When a CRC error or bit stuff error occurs while receiving data in isochronous transfer. When an interval error occurs while receiving data in isochronous transfer.	PIPENRDY
BRDY	Buffer ready interrupt	When the buffer is ready (readable or writable).	PIPEBRDY

Table 30.22 shows operations for a USBf interrupt output from this controller. In case more than one interrupt source is generated, the USBf interrupt output method can be set by using the INTL bit in the SOFCFG register. Set the USBf interrupt output operation according to the user system.

Table 30.22 Operations for USBf Interrupt (USB_FI)

INTL setting	USBf interrupt output pin operation	
	When one interrupt source is generated	When more than one interrupt source is generated
Edge sense (INTL = 0)	Holds the pin low until the interrupt source is cleared.	Negated (L pulse output) for 40 clock cycles at 60 MHz when one source is cleared.
Level sense (INTL = 1)	Holds the pin low until the interrupt source is cleared.	Holds the pin low until all sources are cleared.

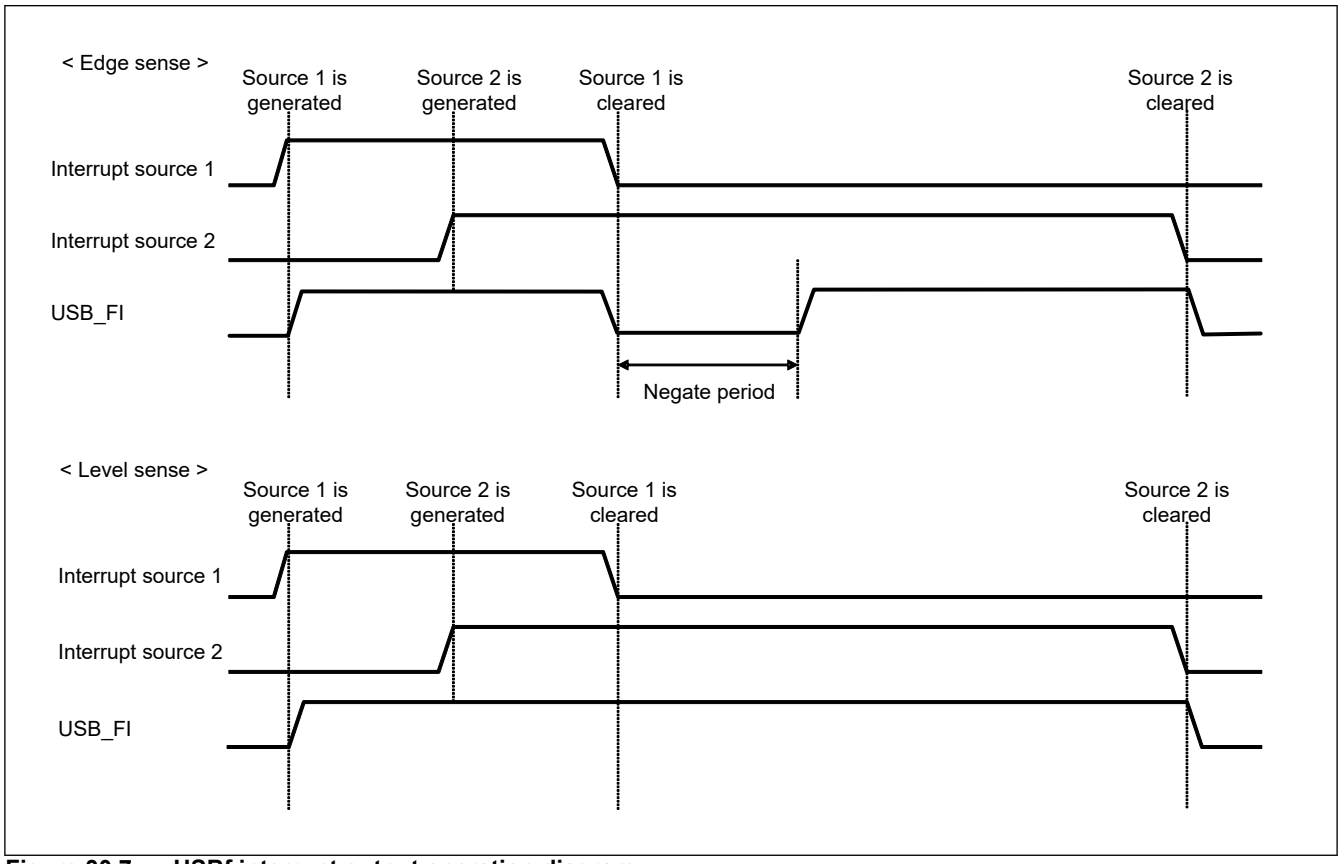


Figure 30.7 USBf interrupt output operation diagram

Figure 30.8 shows the interrupt configurations for the controller.

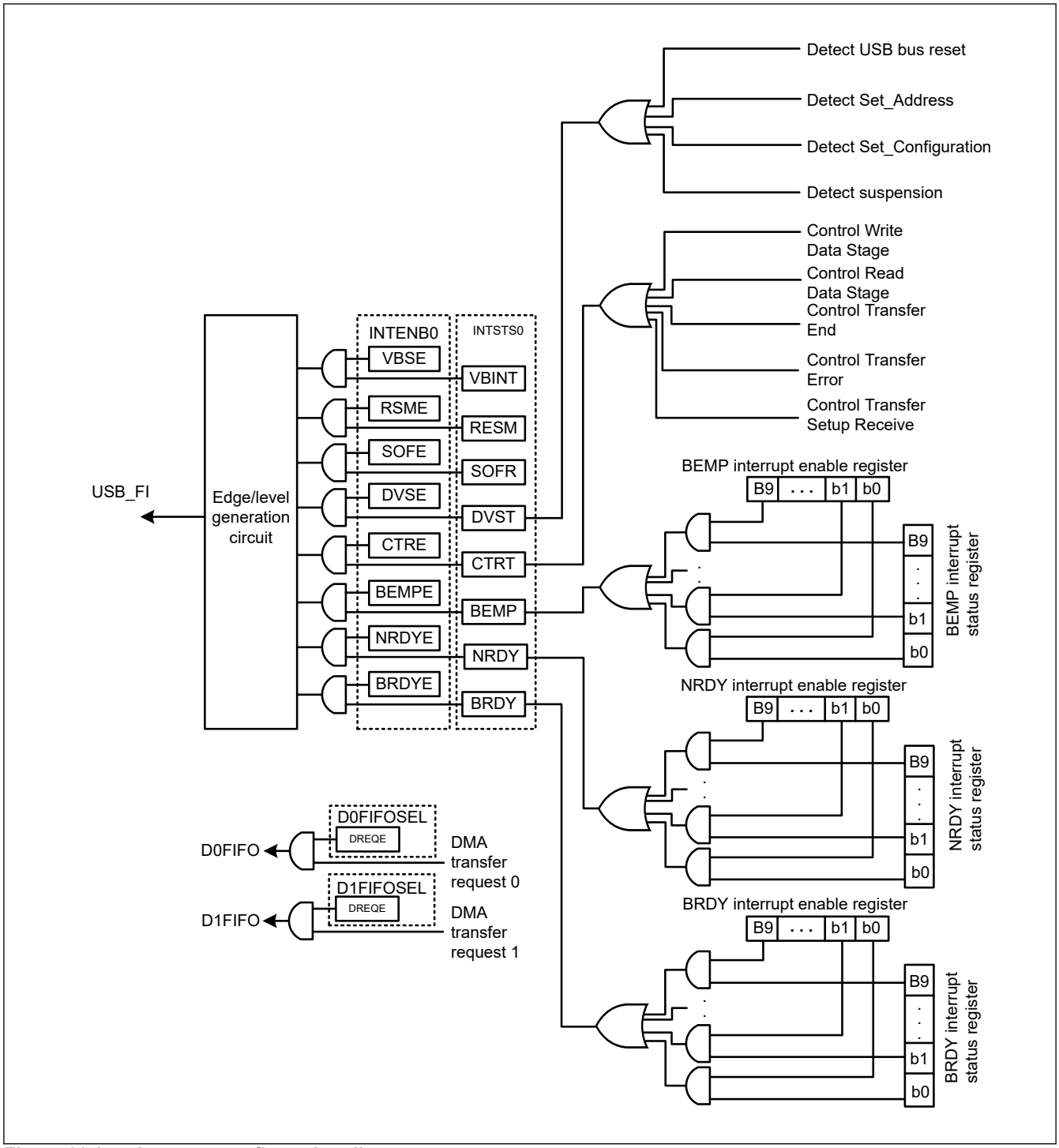
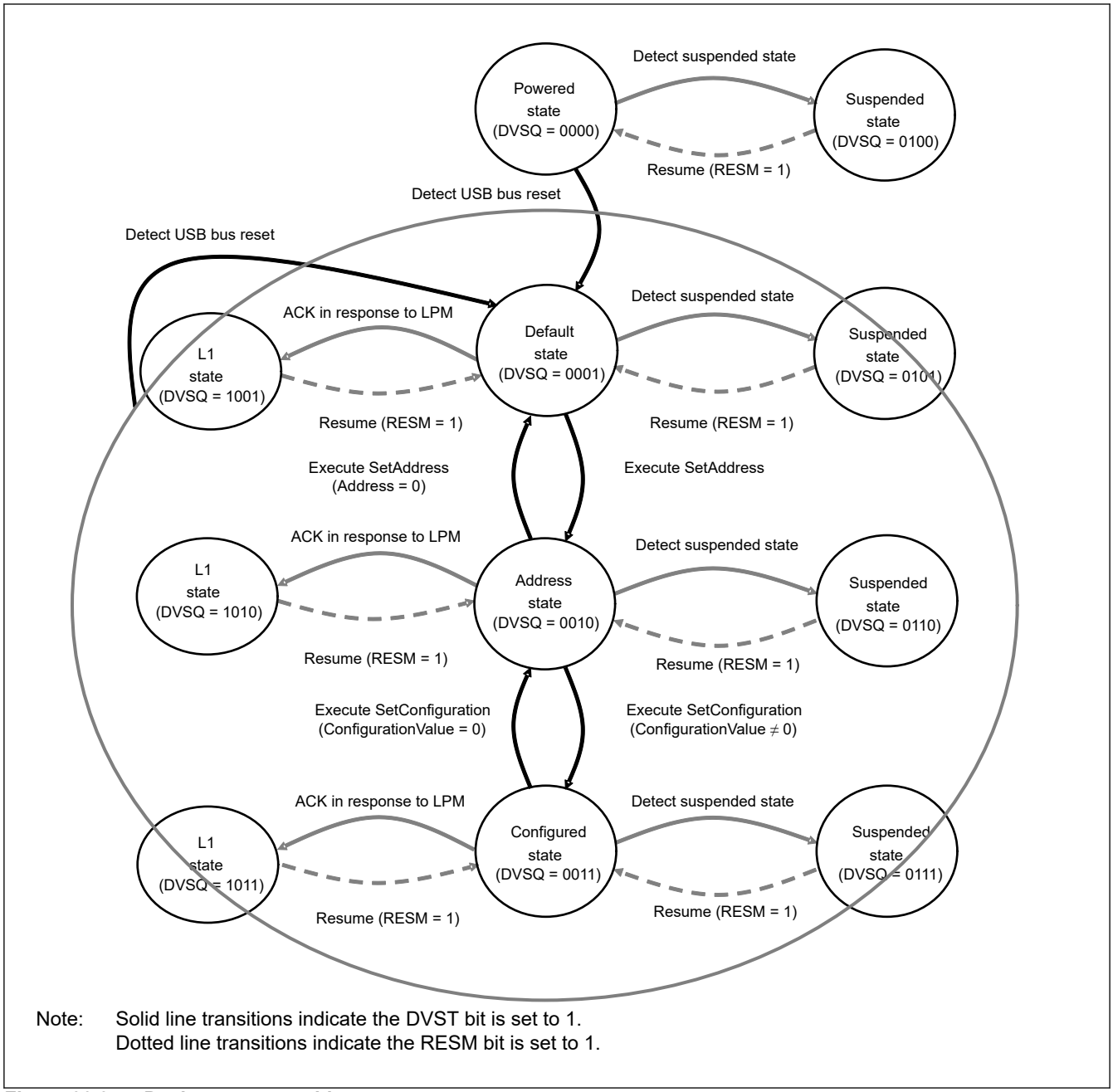


Figure 30.8 Interrupt configuration diagram

30.4.2 Device State Transition Interrupt

Figure 30.9 shows a diagram of how this module handles the device state transitions. The controller monitors the device states and generates the device state transition interrupts. However, recovery from the suspended state (resume signal detection) is detected by the resume interrupt. The device state transition interrupts can be enabled or disabled individually by setting the INTENB0 register. The device state after a transition can be confirmed using the DVSQ bits in the INTSTS0 register.

When making a transition to the default state, the device state transition interrupt is generated after the reset handshake protocol has been completed.



Note: Solid line transitions indicate the DVST bit is set to 1.
Dotted line transitions indicate the RESM bit is set to 1.

Figure 30.9 Device state transitions

30.4.3 Control Transfer Stage Transition Interrupt

Figure 30.10 shows a diagram of how this module handles the control transfer stage transition. The controller monitors the control transfer sequence and generates the control transfer stage transition interrupts. The control transfer stage transition interrupts can be enabled or disabled individually in the INTENB0 register. The control transfer stage after a transition can be confirmed in the CTSQ bits in the INTSTS0 register.

The control transfer sequence errors are described below. If an error occurs, the PID bits in DCPCTR are set to 1xb (STALL).

1. During control read transfers
   (a) At the IN token of the data stage, an OUT or PING token is received when there have been no data transfers at all.
   (b) An IN token is received at the status stage.
   (c) A packet is received at the status stage for which the data packet is DATAPID = DATA0.
2. During control write transfers

(a) At the OUT token of the data stage, an IN token is received when there have been no ACK response at all.

(b) A packet is received at the data stage for which the first data packet is DATAPID = DATA0.

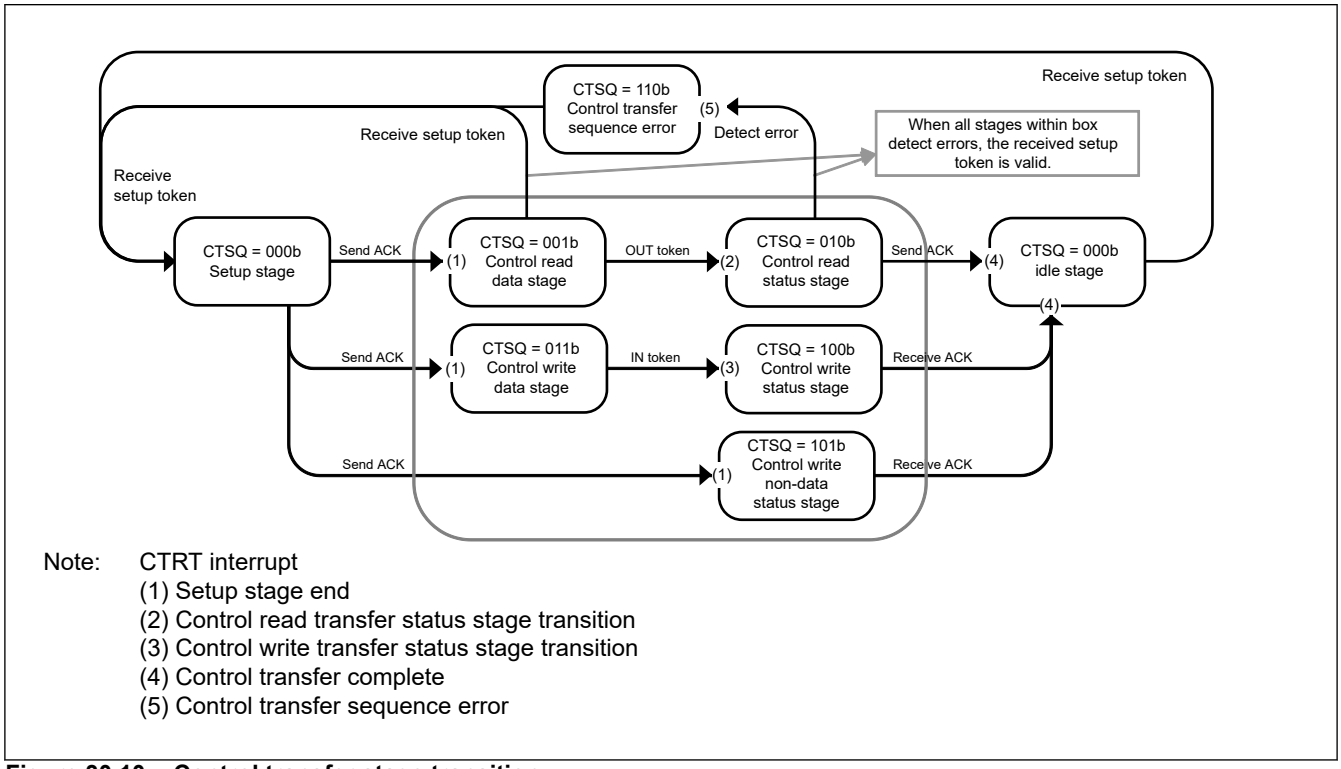
(c) At the status stage, an OUT or PING token is received.

3. During control write no-data transfers

(a) At the status stage, an OUT or PING token is received.

Note that in the control write transfer data stage, if the number of received data is more than the USB request wLength value, the control transfer sequence error cannot be recognized. Also, in the control read transfer status stage, when a packet other than a zero-length packet is received, an ACK response is returned and the transfer is successfully completed.

When a CTRT interrupt is generated due to a sequence error (SERR = 1), the value of CTSQ = 110b is retained until CTRT = 0 is written by the user system (clearing the interrupt state). Therefore, while CTSQ = 110b is retained, the CTRT interrupt for the completion of the setup stage is not generated, even when a new USB request is received. Events occurring after the setup stage are saved by the controller and the CTRT interrupt is generated after the interrupt state is cleared by software.



Note: CTRT interrupt
(1) Setup stage end
(2) Control read transfer status stage transition
(3) Control write transfer status stage transition
(4) Control transfer complete
(5) Control transfer sequence error

Figure 30.10 Control transfer stage transition

30.5 Pipe Control

Table 30.23 provides a list of pipe settings for the controller. In USB data transfers, data transmission is executed in logic pipes called endpoints. This controller comes with ten pipes for data transfer. Each pipe can be set to meet the requirements of the user system.

Table 30.23 Pipe settings

Register name	Bit name	Setting	Remarks
PIPECFG	TYPE	Specifies the transfer type	Can be set for pipes 1 to 9.
	BFRE	Selects BRDY interrupt mode	Can be set for pipes 1 to 5.
	DBLB	Selects double buffer configuration	Can be set for pipes 1 to 5.
	CNTMD	Selects continuous transfer or noncontinuous transfer	Can be set for pipes 1 and 2 only in bulk transfers. Can be set for pipes 3 to 5.
	DIR	Selects transfer direction	Set to IN or OUT
	EPNUM	Endpoint number	Can be set for pipes 1 to 9. Set to a value other than 0x0 when a pipe is in use.
	SHTNAK	Disables pipe when transfer is completed.	Can be set for pipes 1 and 2 only in bulk transfers. Can be set for pipes 3 to 5.
PIPEBUF	BUFSIZE	Buffer memory size	Cannot be set for DCP (fixed to 256 bytes). Up to 2 KB can be set for pipes 1 to 5. Cannot be set for pipes 6 to 9 (fixed to 64 bytes).
	BUFNMB	Buffer memory number	Cannot be set for DCP (fixed at areas 0x00 to 0x03). Can be set for pipes 1 to 5 (specifiable in area ranging from 0x08 to 0x80). Cannot be set for pipes 6 to 9 (fixed at areas ranging from 0x04 to 0x07).
DCPMAXP PIPEMAXP	MXPS	Maximum packet size	Setting conforming to the USB standard
PIPEPERI	IFIS	Buffer flush	Can be set for pipes 1 and 2 only in isochronous transfers. Cannot be set for pipes 3 to 5. Cannot be set for pipes 6 to 9.
	IITV	Interval counter	Can be set for pipes 1 and 2 only in isochronous transfers. Cannot be set for pipes 3 to 5. Cannot be set for pipes 6 to 9.
DCPCTR PIPExCTR	BSTS	Buffer status	DCP state switched between receive and transmission buffer by ISEL bit
	INBUFM	IN buffer monitor	Available only for pipes 3 to 5.
	ATREPM	Auto response mode	Can be set for pipes 1 to 5.
	ACLRM	Auto buffer clear	Can be set for pipes 1 to 9.
	SQCLR	Sequence clear	Clears data toggle bit.
	SQSET	Sequence set	Sets data toggle bit.
	SQMON	Sequence check	Monitors data toggle bit.
	PBUSY	Pipe busy check	—
	PID	Response PID	—
PIPExTRE	TRENB	Transaction count enable	Can be set for pipes 1 to 5.
	TRCLR	Current transaction counter clear	Can be set for pipes 1 to 5.
PIPExTRN	TRNCNT	Transaction counter	Can be set for pipes 1 to 5.

30.5.1 Maximum Packet Size Setting

The MXPS bits in DCPMAXP and PIPEMAXP are used to specify the maximum packet size for each pipe. DCP and PIPE1 to PIPE5 can be set to any of the maximum packet sizes defined by the USB Specification. For PIPE6 to PIPE9, 64 bytes are the upper limit of the maximum packet size. The maximum packet size should be set before beginning the transfer (PID = BUF).

DCP: 64 should be set when using high-speed operation.

DCP: Select and set 8, 16, 32, or 64 when using full-speed operation.

Pipes 1 to 5: 512 should be set when using high-speed bulk transfer.

Pipes 1 to 5: Select and set 8, 16, 32, or 64 when using full-speed bulk transfer.

Pipes 1 to 2: Set a value between 1 and 1024 when using high-speed isochronous transfer.

Pipes 1 to 2: Set a value between 1 and 1023 when using full-speed isochronous transfer.

Pipes 6 to 9: Set a value between 1 and 64.

The high bandwidth transfers used with interrupt transfers and isochronous transfers are not supported.

30.5.2 Response PID

Set the response PID for each pipe with the PID bits of the DCPCTR and PIPExCTR registers.

1. Response PID setting
   The response PID specifies the response to a transaction from the Host.
   (a) NAK setting: Always sends a NAK response when a transaction is issued.
   (b) BUF setting: Responds to the transaction in accordance with the buffer memory state.
   (c) STALL setting: Always sends a STALL response when a transaction is issued.

   Regardless of the value set in the PID bit, an ACK is always sent as a response to a setup transaction and the USB request is stored in corresponding registers.
   Based on the results of the transaction, the controller may trigger the PID bits to be written.
   The controller will trigger a write event to the PID bit in the following cases.
2. Hardware setting of response PID
   (a) NAK setting:
      (i) When SETUP token is received normally (only DCP)
      (ii) In bulk transfers when the SHTNAK bit of the PIPECFG register is set to 1 and short packet is received.
      (iii) In bulk transfers when SHTNAK bit is set to 1 and the transaction counter is completed.
   (b) BUF setting: The BUF cannot be written by the controller.
   (c) STALL setting:
      (i) When a maximum packet size over error is detected in the received data packet
      (ii) When a control transfer sequence error is detected

30.5.3 Pipe Control Register Switching Procedures

The following bits in the pipe control registers can be re-written only when USB transmission is disabled (PID = NAK). Figure 30.11 shows the procedure for switching the pipe control register from the USB transmission enabled (PID = BUF) state.

Registers for which settings are prohibited when the USB transmission is enabled (PID = BUF):

- All bits of the DCPMAXP register
- Bits SQCLR and SQSET of the DCPCTR register
- All bits of the PIPECFG, PIPEBUF, PIPEMAXP and PIPEPERI registers
- ATREPM, ACLRM, SQCLR, and SQSET bits of the PIPExCTR register
- All bits of the PIPExTRE and PIPExTRN registers

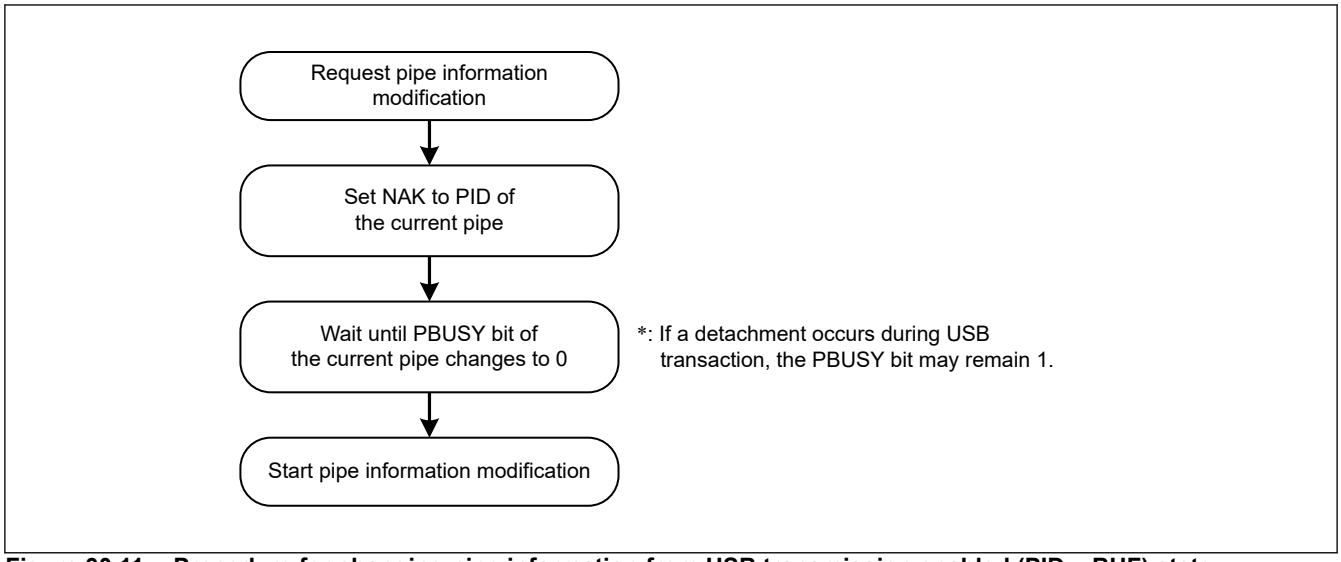


Figure 30.11 Procedure for changing pipe information from USB transmission enabled (PID = BUF) state

In addition, the following bits of the pipe control registers can only be re-written with pipe information that is not set in the CURPIPE bits of CPU/DMA0/DMA1-FIFO ports.

- Register for which settings are prohibited during setting of the CURPIPE bits in the FIFO port register:
  - All bits of the DCPMAXP register
  - All bits of the PIPECFG, PIPEBUF, PIPEMAXP and PIPEPERI registers
  - Bit ACLRM of the PIPExCTR register

When modifying information of a pipe, specify other pipe number in the CURPIPE bits. Also, after setting the DCP pipe information, clear the buffer using the BCLR bit.

30.5.4 Data PID Sequence Bit

When a normal data transfer occurs in the control transfer data stage, bulk transfer or interrupt transfer, the controller automatically toggles the data PID sequence bit. The next data PID sequence bit for data transfer can be confirmed in the SQMON bit in the DCPCTR or PIPExCTR registers. The sequence bit is switched in the ACK handshake receive timing when data is sent or in the ACK handshake send timing when data is received. The data PID sequence bit can also be modified for the SQCLR and SQSET bits of the DCPCTR and PIPExCTR registers.

In control transfers, the controller automatically sets the sequence bit for stage transitions. DATA1 is established at the end of the setup stage. In the status stage, the controller does not reference the sequence bit and returns a response with PID = DATA1. Therefore, the bit does not need to be set with software. Note that the data PID sequence bit must be set with software when a ClearFeature request is received.

Finally, the sequence bit cannot be manipulated through the SQSET bit for the isochronous transfer setup pipe.

30.6 FIFO Buffers

This section describes the operation of the FIFO buffers in this controller.

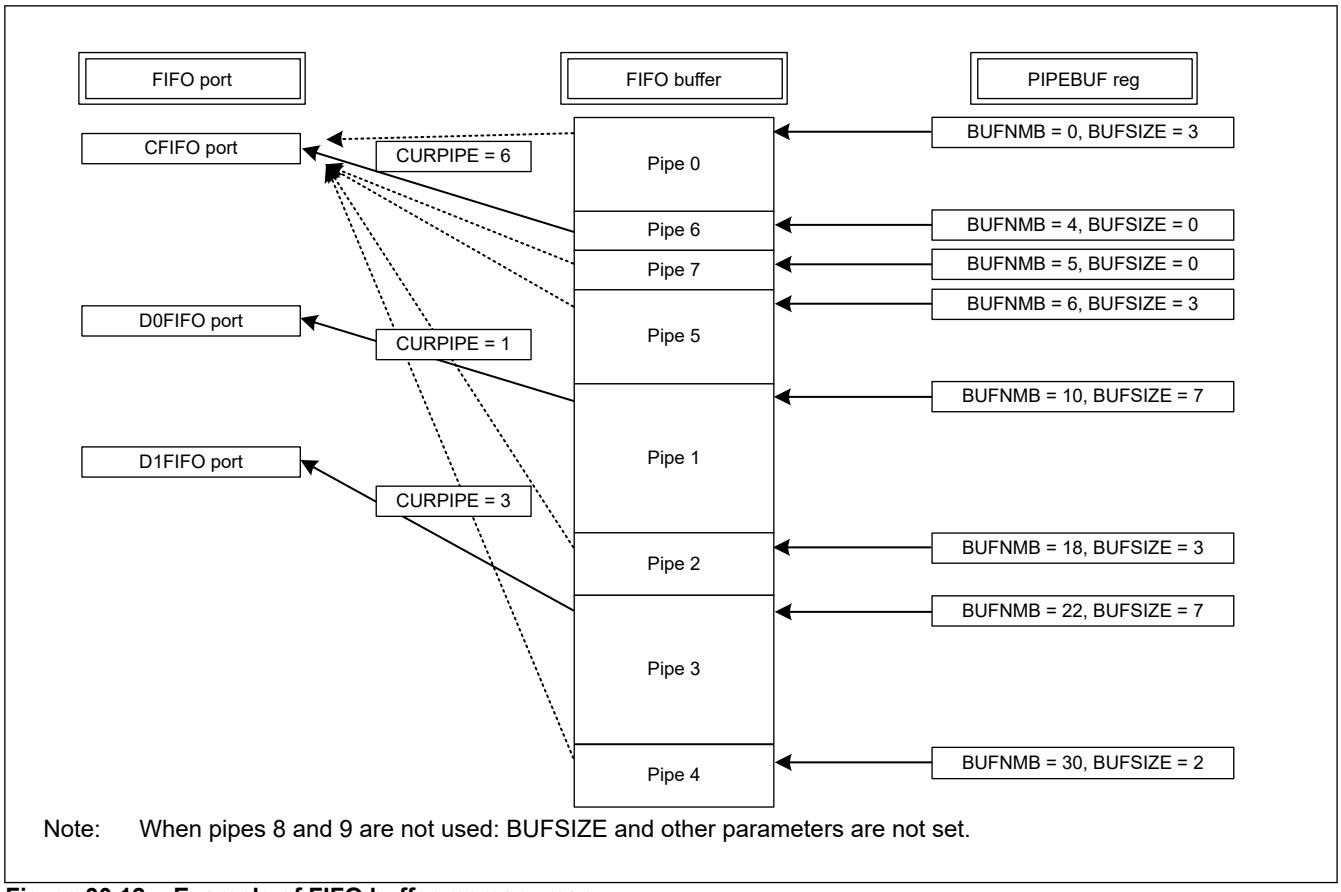
30.6.1 FIFO Buffer Allocation

Figure 30.12 shows an example of memory map for the FIFO buffers of this controller. The FIFO buffer area is shared by the CPU controlling the user system and this controller. The access right of the FIFO buffers may be given to the user system (CPU side) or to this controller (SIE side).

An independent FIFO buffer area is allocated for each pipe. The memory area is made up of memory blocks of 64 bytes and defined by the starting block number (1 block is 64 bytes long) and the number of blocks (specified by the BUFNMB and BUFSIZE bits of the PIPEBUF register). If the CNTMD bit of the PIPExCFG register is set to continuous transfer mode, the value specified in the BUFSIZE bits must be an integral multiple of the maximum packet size. If the double buffer configuration is selected through the DBLB bit of the PIPExCFG register, two planes of memory area the size of which is specified by the BUFSIZE bits of the PIPEBUF register are allocated to a single pipe.

Two FIFO ports are used to access an FIFO buffer (data read/write). The pipe to be assigned to an FIFO port is designated by specifying the pipe number in the CURPIPE bits of the CFIFOSEL or DxFIFOSEL register.

The FIFO buffer state of each pipe can be confirmed by using the BSTS and INBUFM bits of the DCPCTR and PIPExCTR registers. The access right of a FIFO port can be confirmed by using the FRDY bit of the CFIFOCTR or DxFIFOCTR register.



Note: When pipes 8 and 9 are not used: BUFSIZE and other parameters are not set.

Figure 30.12 Example of FIFO buffer memory map

30.6.2 Clearing FIFO Buffers

Table 30.24 shows a list of modes of clearing the FIFO buffers by this controller. The FIFO buffers can be cleared by the 3 bits that are listed in the table.

Table 30.24 List of FIFO buffer clearing modes

Bit name	BCLR	DCLRM	ACLRM
Register	CFIFOCTR register DxFIFOCTR register	DxFIFOSEL register	PIPExCTR register
Function	The FIFO buffer on the CPU side is cleared.	The FIFO buffer is automatically cleared after the data is read from the designated pipe.	The buffer is automatically cleared to discard all the received packets.
Clearing method	Write 1 to clear.	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

30.7 FIFO Port Function

This section describes FIFO port functions. Table 30.25 lists settings of FIFO port functions of this controller. If writing data is continued until the buffer becomes full (up to the maximum packet size in discontinuous transfer mode) during the data write access, the FIFO port automatically enters the state where data can be transmitted to the USB bus. To make data with a size less than the buffer full (the maximum packet size in discontinuous transfer mode) transmittable, write completion must

be set by the BVAL bit in the CFIFOCTR or DxFIFOCTR register. Furthermore, to transmit a zero-length packet, clear the buffer by the BCLR bit in the CFIFOCTR or DxFIFOCTR register and set write completion by the BVAL bit.

When all data has been read during the read access, the FIFO port automatically enters the state where a new packet can be received. However, when a zero-length packet is received (DTLN = 0), the data cannot be read. In that case, clear the buffer by the BCLR bit in the CFIFOCTR or DxFIFOCTR register. The receive data length is checked by the DTLN bits in the CFIFOCTR or DxFIFOCTR register.

Table 30.25 FIFO Port function settings

Register name	Bit name	Function	Notes
C/DxFIFOSEL	RCNT	DTLN read mode selection	
	REW	Buffer memory rewind (re-read, re-write)	
	DCLRM	Automatic clearing of buffer memory after specified pipe received data is read	DxFIFO only
	DREQE	DREQ signal assertion	DxFIFO only
	MBW	FIFO port access bit width	
	BIGEND	FIFO port endian selection	
	ISEL	FIFO port access direction	DCP only
	CURPIPE	Current pipe selection	
C/DxFIFOCTR	BVAL	Buffer memory write completion	
	BCLR	Clearing of CPU-side buffer memory	
	FRDY	Monitoring of FIFO port ready	
	DTLN	Confirmation of received data length	

30.7.1 FIFO Port Selection

Table 30.26 shows the list of pipes that can be selected in each FIFO port. The pipes to be accessed are selected with the CURPIPE bits of the CFIFOSEL or DxFIFOSEL register. After selecting the pipes, confirm that the value of the CURPIPE bits written was read correctly (if the previous pipe number is read out, this indicates the controller is still changing the pipe), then confirm that FRDY = 1 and access the FIFO port. Figure 30.13 shows the procedure for switching pipes for access to the FIFO port.

Also, select the bus width for the FIFO port access with the MBW bit. The buffer memory access direction is determined by the ISEL bit for DCP, and the DIR bit of the PIPExCFG register for all other pipes.

Table 30.26 FIFO port access by pipe

Pipe	Access method	Usable ports
DCP	CPU access	CFIFO port register
Pipes 1 to 9	CPU access	CFIFO port register
	DMA access	DxFIFO port register

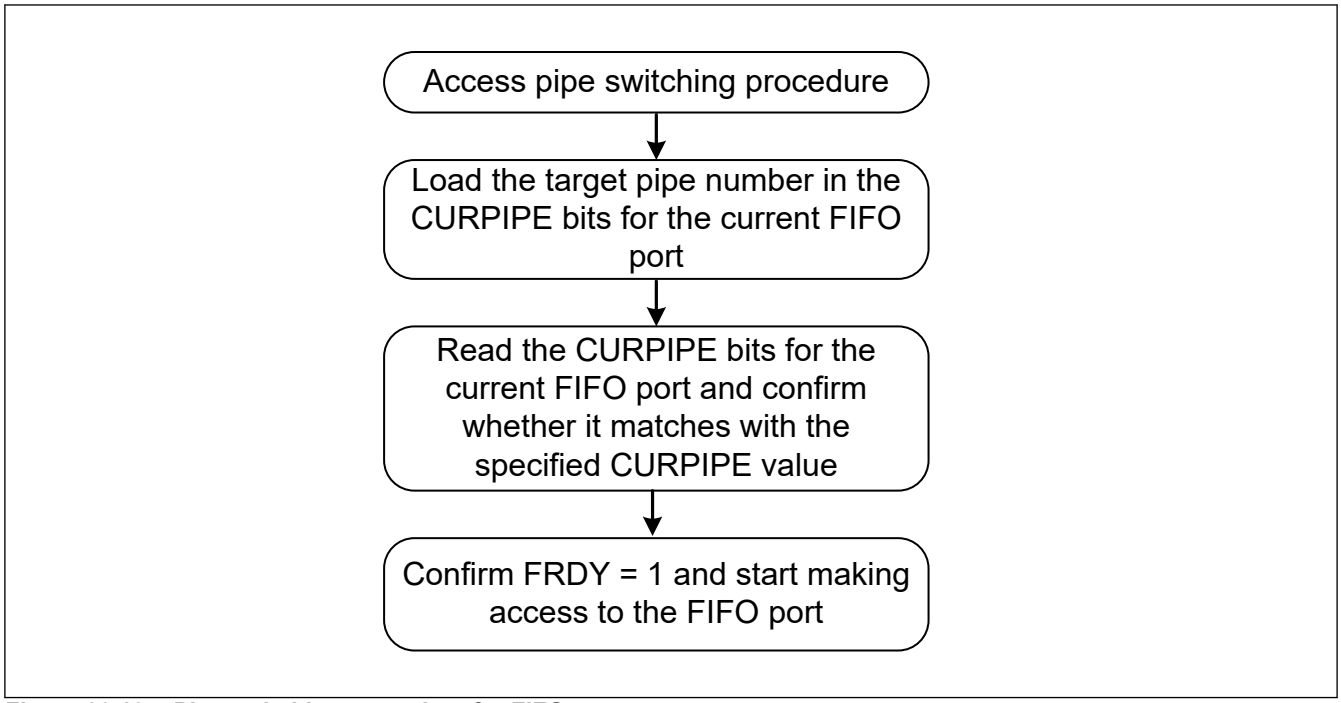


Figure 30.13 Pipe switching procedure for FIFO port access

30.7.2 DxFIFO Automatic Clear Mode (DxFIFO Port Read Direction)

When a data read event of the controller buffer memory is completed with setting the DCLRM bit of the DxFIFOSEL register to 1, the buffer memory of the corresponding pipe is automatically cleared.

Table 30.27 shows the correspondence between packet reception and buffer memory clearing by software in each setting.

As indicated in Table 30.27, the buffer clear conditions differ according to the set value of the BFRE bit, even for states in which clear is normally required, using the DCLRM bit eliminates the need for clearing of the buffer by software, enabling DMA transfers without the use of software.

Note that this function only supports the buffer memory read direction setting.

Table 30.27 Correspondence of packet reception and buffer memory clearing by software

Buffer state when packet is received	DCLRM = 0		DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	Clearing not required	Clearing not required	Clearing not required	Clearing not required
Zero-length packet received	Clearing required	Clearing required	Clearing not required	Clearing not required
Normal short packet received	Clearing not required	Clearing required	Clearing not required	Clearing not required
Transaction count end	Clearing not required	Clearing required	Clearing not required	Clearing not required

30.7.3 BRDY Interrupt Timing Selection

The BFRE bit of the PIPECFG register can be set so that the BRDY interrupt is not generated when a data packet of maximum packet size is received.

When using a DMA transfer, this function enables an interrupt to be generated only when the last data is received. The last data indicates either a short packet reception or the transaction count end. By setting BFRE = 1, the BRDY interrupt will be generated after the received data is read. By reading the DTLN bit of the DnFIFOCTR register, the receive data length of last data packet received just before the BRDY interrupt was generated can be confirmed.

Table 30.28 shows the timing of the BRDY interrupt.

Table 30.28 BRDY interrupt generation timing

Buffer state when packet is received	BFRE = 0	BFRE = 1
Buffer full (normal packet received)	When packet is received	No interrupt generated
Zero-length packet received	When packet is received	When packet is received
Normal short packet received	When packet is received	When read event of data received from buffer memory is completed
Transaction count end	When packet is received	When read event of data received from buffer memory is completed

The BFRE bit function is only valid in reading direction of the buffer memory. When in writing direction, fix the BFRE bit to 0.

30.8 Control Transfer (DCP)

Data transfers of the data stage of control transfers are done using the default control pipe (DCP). The DCP buffer memory is a 64 byte single buffer, and is a fixed area that is shared for both control reading and control writing. The buffer memory can be accessed only through the CFIFO port.

30.8.1 Setup Stage

The controller always responds with an ACK when it receives a normal setup packet. The controller operations in the setup stage are as follows.

1. When a new setup packet is received, the controller sets the following bits.
   (a) Sets the VALID bit of the INTSTS0 register to 1.
   (b) Sets the PID bits of the DCPCTR register to NAK.
   (c) Sets the CCPL bit of the DCPCTR register to 0.
2. When a data packet is received following the setup packet, the USB request parameters are stored in the following registers: USBREQ, USBVAL, USBINDX and USBLENG.

Always set VALID = 0 before the response process to a control transfer. While VALID = 1, PID = BUF will not be set and the data stage cannot be completed.

The function of the VALID bit allows the controller to temporarily stop a request in-process when it receives a new USB request during a control transfer, and respond to the newest request.

In addition, the controller automatically judges the direction bit (bmRequestType bit 8) and the request data length (wLength) of the received USB request and determines whether it is a control read transfer, control write transfer or control write no-data transfer, and then handles the stage transition. If the sequence is incorrect, a sequence error for the control transfer stage transition interrupt is generated and is notified to the software. For more information concerning the controller stage management, see Figure 30.10.

30.8.2 Data Stage

Use the DCP for data transfers in response to receiving a USB request. Before accessing the DCP buffer memory, set the access direction in the ISEL bit of the CFIFOSEL register.

The transaction is executed by setting the PID bits of the DCPCTR register to BUF.

Data transfer completion is detected by the BRDY and BEMP interrupts. Use the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

For control write transfers in high-speed operation, a NYET handshake is sent in accordance with the buffer memory state.

30.8.3 Status Stage

When the setting of the PID bits of the DCPCTR register is BUF, set the CCPL bit to 1 to complete the control transfer.

After the above settings, the controller automatically executes the status stage in accordance with the data transfer direction fixed in the setup stage. The detailed process is as follows.

1. Control read transfers:

The controller receives a zero-length packet from the USB host and sends an ACK response.

2. Control write transfers and no-data control transfers:
   The controller sends a zero-length packet and receives an ACK response from the USB host controller.

30.8.4 Control Transfer Automatic Response

The controller automatically sends a response to a normal SET_ADDRESS request. If one of the following errors occurs, a response must be sent by software.

1. bmRequestType ≠ 0x00
2. wIndex ≠ 0x00
3. wLength ≠ 0x00
4. wValue > 0x7F
5. DVSQ = 011b (configured)

All requests other than the SET_ADDRESS request must be responded to by software.

30.9 Bulk Transfer (Pipes 1 to 5)

The user can select the buffer memory usage method (single/double buffer, continuous/non-continuous transfer mode) for bulk transfer. The maximum size that can be set for the buffer memory is 2 KB. The controller manages the buffer memory state and automatically responds to PING packets and NYET handshakes.

30.9.1 NYET Handshake Control

Table 30.29 shows the list of responses to a token received in a bulk or control transfer. When an OUT token is received in a bulk or control transfer and there is only enough open space for one packet in the buffer memory, the controller sends a NYET response. However, when a short packet is received, the controller sends an ACK response instead of a NYET response, even under these conditions.

Table 30.29 List of responses to received tokens

PID bits setting	Buffer memory state	Received token	Response	Notes
NAK/STALL	—	SETUP	ACK	—
	—	IN/OUT/PING	NAK/STALL	—
BUF	—	SETUP	ACK	—
	RCV-BRDY*1	OUT/PING	ACK	When OUT token is received, data packet is received.
	RCV-BRDY*2	OUT	NYET	Data packet is received.
	RCV-BRDY*2	OUT (Short)	ACK	Data packet is received.
	RCV-BRDY*2	PING	ACK	
	RCV-NRDY*3	OUT/PING	NAK	
	TRN-BRDY*4	IN	DATA0/1	Data packet is transmitted.
	TRN-NRDY*5	IN	NAK	

Note 1. RCV-BRDY: Buffer memory has enough space for 2 packets or more when an OUT or PING token is received.
Note 2. RCV-BRDY: Buffer memory has only enough space for one packet when an OUT token is received.
Note 3. RCV-NRDY: Buffer memory has not enough space when a PING token is received.
Note 4. TRN-BRDY: Buffer memory has data for transmission when an IN token is received.
Note 5. TRN-NRDY: Buffer memory does not have data for transmission when an IN token is received.

30.10 Interrupt Transfer (Pipes 6 to 9)

This controller executes an interrupt transfer in accordance with the period managed by the host controller. The controller ignores (no response) PING packets in interrupt transfers. In addition, the controller does not send a NYET handshake, but responds with ACK, NAK or STALL.

The controller does not support high-bandwidth interrupt transfers.

30.11 Isochronous Transfer (Pipes 1 and 2)

The controller provides the following functions for isochronous transfers.

1. Isochronous transfer error information notification
2. Interval counter (IITV bit setting)
3. Isochronous IN transfer data setup control (IDLY function)
4. Isochronous IN transfer buffer flush function (IFIS bit setting)
5. SOF pulse output function

The controller does not support high-bandwidth isochronous transfers.

30.11.1 Isochronous Transfer Error Detection

The controller manages isochronous transfer errors by software and therefore has the following error information detection functions. Table 30.30 and Table 30.31 describe the procedure in which errors are confirmed and the interrupts that are generated.

1. PID error
   When PID of the received packet is corrupted
2. CRC error and bit stuffing error
   When an error occurs in CRC of the received packet or when the bit stuffing is corrupted
3. Maximum packet size over
   This indicates the data size of the received packet is larger than the value set for the maximum packet size.
4. Overrun and underrun
   When there is no data in the buffer memory when an IN token is received in an IN-direction (send) transfer When there is no empty space in the buffer memory when an OUT token is received in an OUT-direction (receive) transfer
5. Interval error
   The following will generate interval errors.
   (a) When an IN token could not be received in the interval frame of an isochronous IN transfer
   (b) When an OUT token could not be received in the interval frame of an isochronous OUT transfer

Table 30.30 Errors detected in token reception/transmission

Detection priority	Error type	Generated interrupts and states at time of error detection
1	PID error	No interrupt generated (ignored as corrupted packet)
2	CRC error or bit stuffing error	No interrupt generated (ignored as corrupted packet)
3	Overrun or underrun errors	NRDY interrupt is generated and OVRN bit is set. A zero-length packet is sent in response to an IN token. A data packet is not received in response to an OUT token.
4	Interval error	NRDY interrupt generated

Table 30.31 Errors detected in data packet reception

Detection priority	Error type	Generated interrupts and states
1	PID error	No interrupt generated (ignored as corrupted packet)
2	CRC error or bit stuffing error	NRDY interrupt is generated and CRCE bit is set.
3	Maximum packet size over error	BEMP interrupt is generated and PID is set to STALL.

30.11.2 DATA-PID

This controller does not support high-bandwidth transfers. The following occurs in response to a received PID.

1. IN direction:
   (a) DATA0: Transmitted as PID of the data packet
   (b) DATA1: Not transmitted
   (c) DATA2: Not transmitted
   (d) mDATA: Not transmitted
2. OUT direction (in full-speed operation):
   (a) DATA0: Received successfully as PID of the data packet
   (b) DATA1: Received successfully as PID of the data packet
   (c) DATA2: The packet is ignored.
   (d) mDATA: The packet is ignored.
3. OUT direction (in high-speed operation):
   (a) DATA0: Received successfully as PID of the data packet
   (b) DATA1: Received successfully as PID of the data packet
   (c) DATA2: Received successfully as PID of the data packet
   (d) mDATA: Received successfully as PID of the data packet

30.11.3 Interval Counter

30.11.3.1 Overview of Operation

The isochronous transfer interval can be set in the IITV bit of the PIPEPERI register. Table 30.32 shows the functions of the interval counter.

Table 30.32 Interval counter functions

Transfer direction	Function	Detection conditions
IN	Transfer buffer flush function	Cannot successfully receive IN token in interval frame during isochronous IN transfer.
OUT	Notifies that a token not being received	Cannot successfully receive OUT token in interval frame during isochronous OUT transfer.

Since the interval counting is performed upon reception of SOF or by the complemented SOF, the isochronism can be maintained even if the SOF is damaged. Frame intervals are set as 2^{IITV} (μ) frames.

30.11.3.2 Interval Counter Initialization

The controller initializes the interval counter under the following conditions.

1. Hardware reset
   Initializes the IITV bit.
2. Clearing of the buffer memory by the ACLRM bit
   This initializes the counter but not the IITV bit.
3. USB bus reset

After the interval counter is initialized and a packet is successfully transferred, the interval count starts under the following conditions.

1. SOF is received after data is sent in response to an IN token when PID = BUF.
2. SOF is received after data is received in response to an OUT token when PID = BUF.

Note that the interval counter is not initialized in the following conditions.

1. When the PID is set to NAK or STALL
   The interval timer is not stopped at this time. The transaction will be attempted at the next interval.

2. USB bus reset or USB suspend
   The IITV bit is not initialized at this time. When the SOF is received, the count starts from the value before the reception.

30.11.4 Isochronous Transfer Transmission Data Setup

In the isochronous data transmission by this controller, after data is written to the buffer memory, the data packet can be sent out in the next frame after the SOF packet is detected. This function, called the isochronous transfer transmission data setup, allows specification of the frame that started transmission.

When the buffer memory is used for double buffering and writing to both buffers has been completed, only transfer from the first buffer to have received data can proceed. Therefore, even when several IN tokens are received in the same frame, only one packet of data is sent by the buffer memory.

When an IN token is received, if the buffer memory is ready for transmission, the data is transferred and a normal response is returned. However, if the buffer memory is not ready for transmission, a zero-length packet is sent and an underrun error occurs.

Figure 30.14 shows an example of transmission using the isochronous transfer transmission data setup function with this controller when IITV = 0 (for each frame) is set.

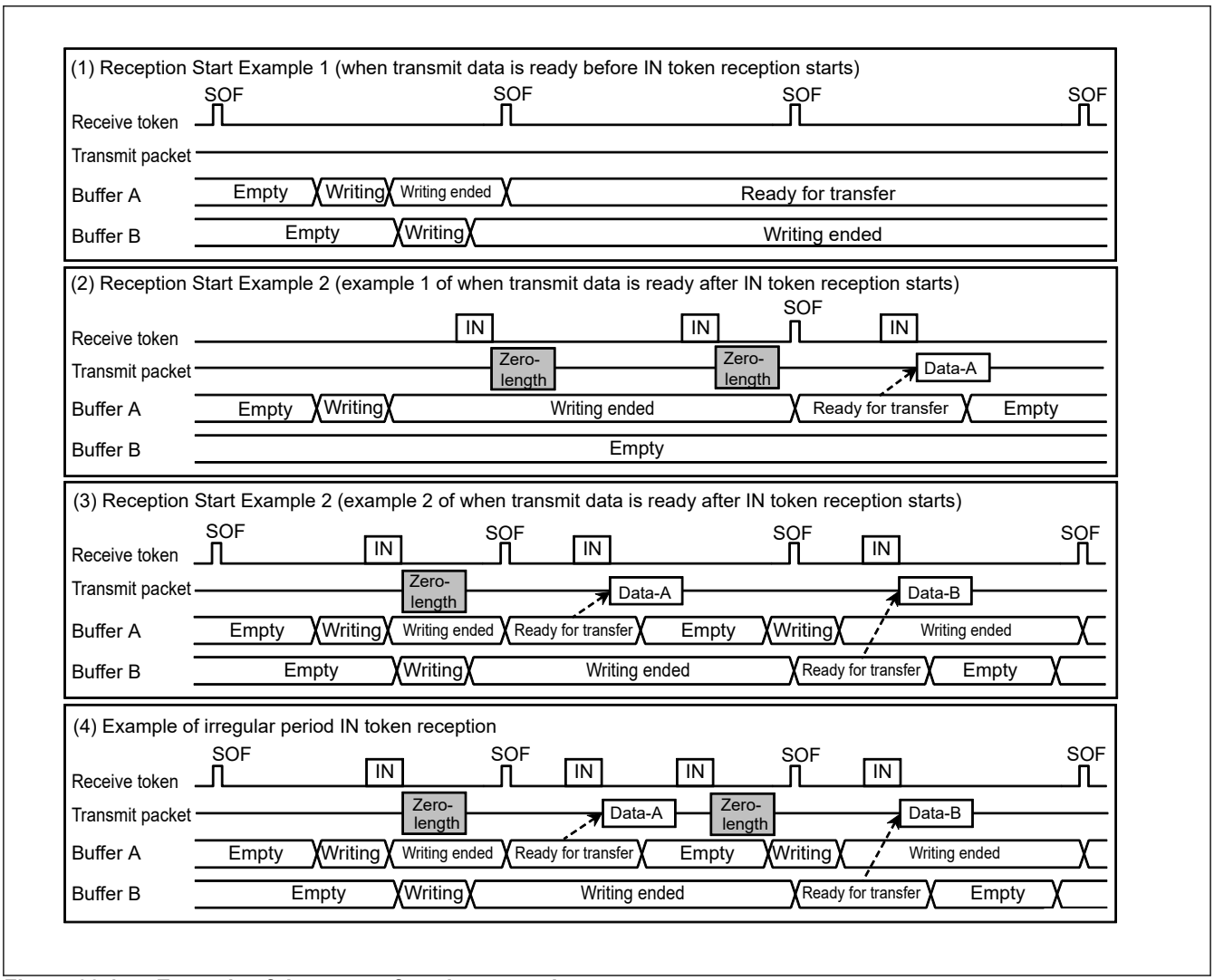


Figure 30.14 Example of data setup function operation

30.11.5 Isochronous Transfer Transmission Buffer Flush

When an SOF packet or a µSOF packet of the next frame is received without receiving an IN token in the interval frame during isochronous data transmission, this controller operates as if a corrupted IN token was received, and clears the buffer which is ready for transmission, making that buffer ready for writing.

If a double buffer is being used and writing to both buffers has been completed, data are considered to have been sent from the buffer memory that was cleared in the same interval frame, and transmission is enabled for the buffer memory that is not discarded with SOF or µSOF packets reception.

The timing at which the buffer flush function is activated varies depending on the setting of the IITV bits.

1. When IITV = 0
   The buffer flush operation proceeds from the first frame after the pipe becomes valid.
2. When IITV ≠ 0
   The buffer flush operation proceeds after the first successful transaction.

Figure 30.15 shows an operation example of the buffer flush function of this controller. When an unanticipated token is received prior to the interval frame, this controller sends the written data or a zero-length packet as an underrun error according to the data setup state.

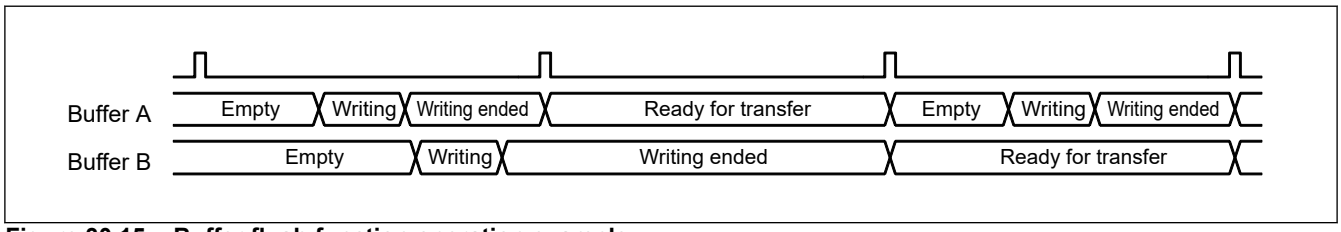


Figure 30.15 Buffer flush function operation example

Figure 30.16 shows an example of an interval error generated in the controller. There are five types of interval errors, as listed below. Timing 1 in the figure shows when the interval error occurs and how the buffer flush function operates.

When an interval error occurs during an IN transfer, the buffer flush function goes into operation; during an OUT transfer, the NRDY interrupt is generated.

Use the OVRN bit to determine whether an error is an NRDY interrupt, such as a receive packet error, or an overrun error.

Responses to the tokens in the shaded boxes are executed in accordance to the buffer memory state.

1. IN direction:
   (a) If the buffer is ready for transfer, data is transferred as a normal response.
   (b) If the buffer is not ready for transfer, a zero-length packet is sent and an underrun error occurs.
2. OUT direction:
   (a) If the buffer is ready for reception, data is received as a normal response.
   (b) If the buffer is not ready for reception, data is discarded and an overrun error occurs.

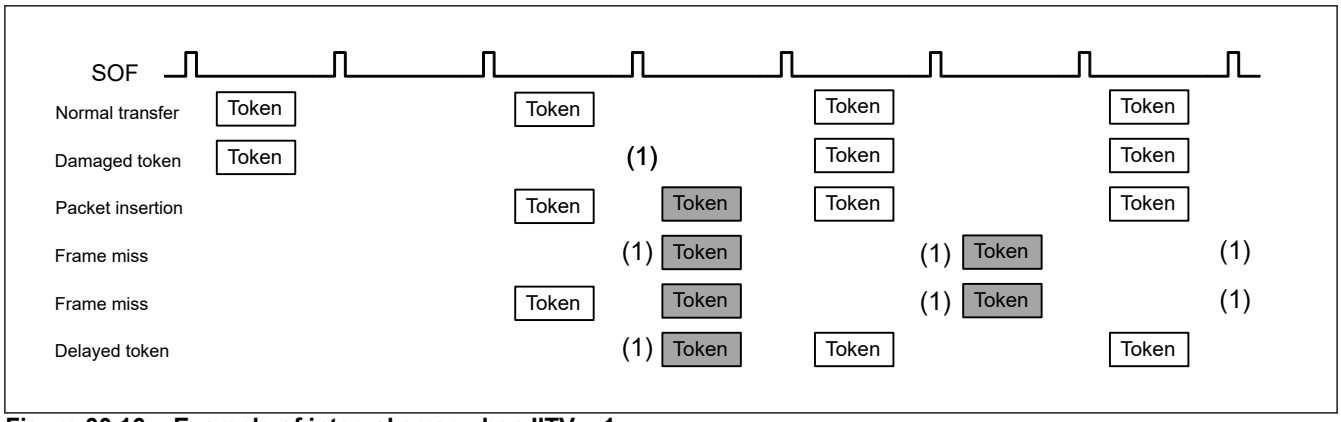


Figure 30.16 Example of interval error when IITV = 1

30.12 SOF Interpolation

If an SOF packet could not be received at intervals of 1 ms (in full-speed operation) or 125 µs (in high-speed operation) because of corruption or missing, this controller interpolates the SOF. The SOF interpolation operation begins when both the USBE and SUSPM bits have been set to 1 and an SOF packet is received. The interpolation is initialized under the following conditions.

1. Hardware reset
2. USB bus reset
3. Suspended state detected

The SOF interpolation operates according to the following specifications.

1. The frame interval (125 µs or 1 ms) is based on the results of the reset handshake protocol.
2. The interpolation does not operate until the SOF packet is received.
3. After a first SOF packet is received, interpolation of another SOF packet proceeds after a 125-µs or 1-ms interval counted in cycles of the internal clock running at 48 MHz clock has elapsed.
4. Interpolation is performed in the previous reception intervals after the 2nd and subsequent SOF packets are received.
5. Interpolation is not performed in the suspended state or during reception of a USB bus reset.
   When the controller goes to the suspended state in high-speed operation, interpolation continues for 3 ms from the last packet.

The SOF interpolation works for the following.

1. Updating of the frame number or micro-frame number
2. SOFR interrupt and µSOF lock
3. SOF pulse output
4. Isochronous transfer interval count

When an SOF packet is lost during full-speed operation, the FRNM bit of the FRMNUM register is not updated.

When a µSOF packet is lost during high-speed operation, the UFRNM bit of the UFRMNUM register is updated.

However, when a µSOF packet when µFRNM = 000b is lost, the FRNM bit is not updated. At this time, even if µSOF packets when µFRNM ≠ 000b are received successfully, the FRNM bit is not updated.

30.13 Link Power Management Processing

The Link Power Management standard redefines the existing suspended state as L2 state, and further defines L1 state that enables transitions and recovery with lower latency than the L2 (suspended) state.

The following table shows comparison between L2 (suspended) and L1.

Table 30.33 Comparison between L2 (suspended) and L1

Item	L1	L2 (suspended)
Transition	LPM transaction	3 ms idle
Recovery due to host	(Host) The host can specify the minimum drive period (75 µs to 1.175 ms). (Device) 10 µs K drive	(Host) Min 20 ms K drive (Device) 10 ms K drive
Recovery due to device	(Device) 50 µs K drive (Host) 60 µs to 990 µs K drive (Device) 10 µs K drive	(Device) 1 ms to 15 ms K drive (Host) Min 20 ms K drive (Device) 10 ms K drive
Signaling	Low and full speed idle	Low and full speed idle

The following describes the transition (to L1 state) processing and recovery processing.

30.13.1 Descriptor

When the GetDescriptor command is received, the own descriptor must be sent back.

It is necessary to change the contents of the descriptor to be returned depending on whether to respond to the transition to L1 and recovery processing because of the LPM transaction, as shown in the table below.

Table 30.34 Relationship between response to LPM and descriptor

Response to LPM	bcdUSB	Provision of USB2.0Ex.Desc	LPM of USB2.0Ex.Desc	Response when LPM is received	Remarks
Does not respond	0x0200	Not provided	—	Not Respond	Standard operation when the descriptor does not respond to LPM
	0x0201	Provided	LPM = 0	STALL	Clear declaration that the descriptor does not respond to LPM. In this case, no response must be avoided and STALL must be sent in response.
Responds	0x0201	Provided	LPM = 1	ACK or NYET	Standard operation when the descriptor responds to LPM

Whether to respond to the transition to L1 and recovery processing is declared with the LPM bit in the USB2.0 extension descriptor. To provide the USB2.0 extension descriptor, a value of 0x0201 or larger must be set for the bcdUSB field of the device descriptor.

If the descriptor does not respond to LPM, the USB2.0 extension descriptor is not provided and the bcdUSB field is set to 0x0200. In this case, even if LPM token is received, ignore it.

If the descriptor does not respond to LPM, it is also possible to set the bcdUSB field to 0x0201 and clear the LPM bit in the USB2.0 extension descriptor to 0 (does not respond). In this case, however, LPM cannot be ignored and STALL must be sent in response. When the descriptor responds to LPM, set the bcdUSB field to 0x0201 and set the LPM bit in the USB2.0 extension descriptor to 1 (responds). This enables acknowledgment of NYET or ACK response to the LPM token.

30.13.2 Basic Processing

The following processing is required.

1. Send no response, ACK, NYET, or STALL response to the LPM token received from the host in accordance with the own state.
2. If re-transmission of LPM token is not detected for 8 µs after an ACK response was sent, a transition to the L1 state is made.
3. Detect the K drive of the host and perform the processing for recovery to the idle state.
4. Perform the processing for recovery to the idle state by remote wakeup.

For step 1, hardware receives the LPM token, it sends the response. Hardware performs processing in step 2 in addition to re-transmission control and transition to the L1 state.

A transition to the L1 state can be checked by a DVST interrupt.

For step 3, a RESM interrupt is generated by detecting the K drive of the host in the L1 state.

For step 4, setting the WKUP bit to 1 by software gives an instruction to start remote wakeup to the hardware. In the specifications, software clears this bit at the time of recovery from the L2 state, but hardware clears this bit at the time of recovery from the L1 state.

30.13.3 HIRD Value Negotiation

The HIRD value included in the LPM token indicates the host K period for recovery from the L1 state. It is supposed that the desired HIRD value may vary depending on applicable applications. It is supposed that a small HIRD value is desired for applications focusing on improvement in transfer efficiency, but a large HIRD value is desired for applications focusing on low power consumption.

According to the L1NEGOMD and HIRDTHR bits in the L1CTRL register, an ACK response is returned if the received HIRD value is within the desired range. Otherwise, a NYET response is returned to request the host to correct the HIRD value.

Note: Using this HIRD value negotiation function requires that the negotiation processing be also supported on the host side.

30.14 DMA Mode

30.14.1 Register Mode/Link Mode

With the DMS bit in the CHCFG_n register, DMA mode can be switched between register mode and link mode.

Table 30.35 DMA mode setting

DMS(CHCFG)	Mode	Description
0	Register Mode	Performs a DMA transfer with the values set for the next register set.
1	Link Mode	Accesses the descriptor area and performs a DMA transfer with the value set for the descriptor. Repeats descriptor read and DMA transfer unless you set the descriptor or use the control register to stop them.

30.14.1.1 Register Mode

In register mode, you can perform a DMA transfer using the value set in the internal register. You can set two sets (Next0 register set and Next1 register set) of transfer source addresses, the transfer destination addresses, and the numbers of transfer bytes.

You can select the next register set to perform a transfer, or use two next register sets for continuous transfer.

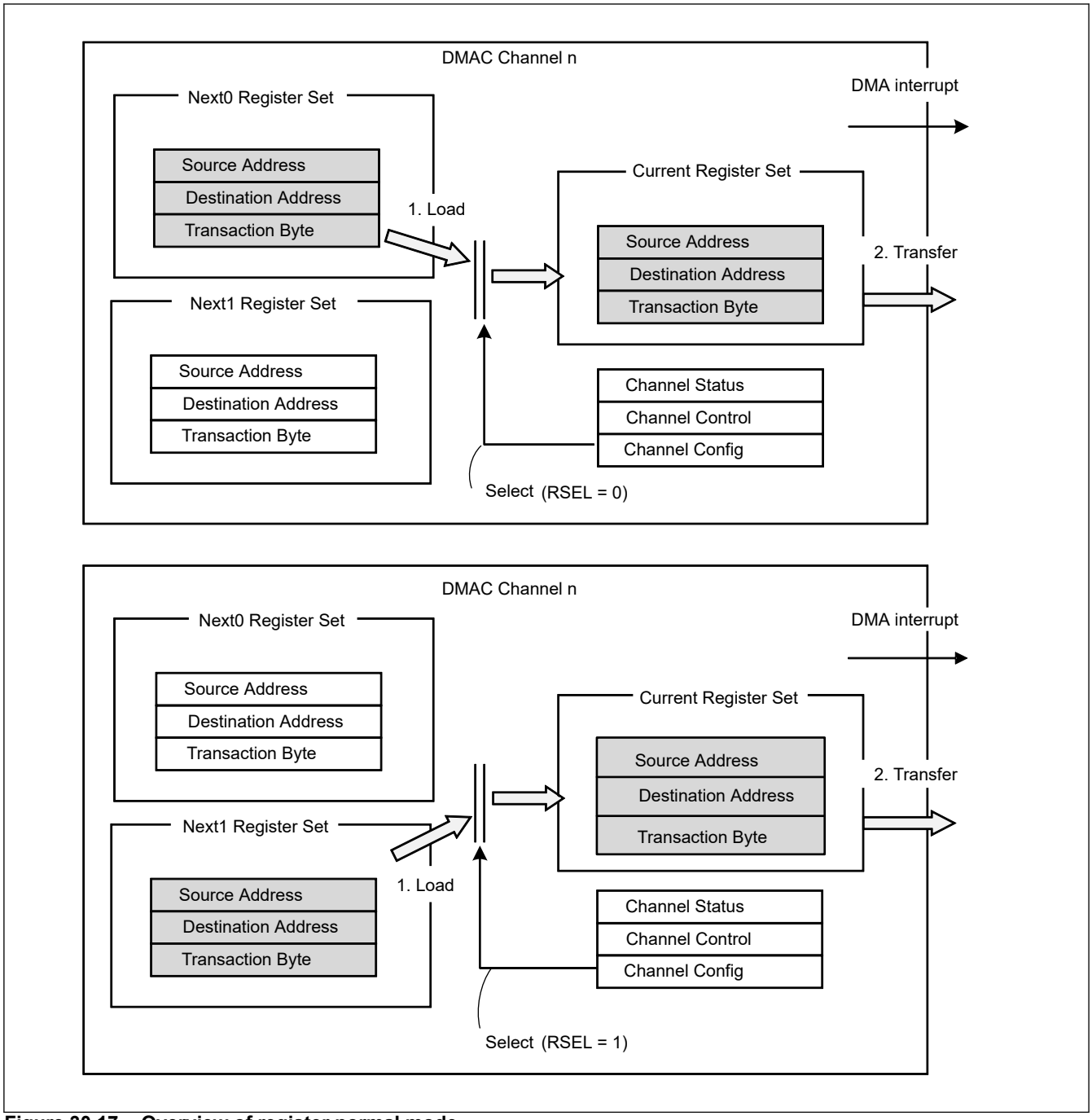


Figure 30.17 Overview of register normal mode

The above figure illustrates operation when Next0 register set is executed (above) and operation when Next1 register set is executed (below).

(1) Operation Flow in Register Mode

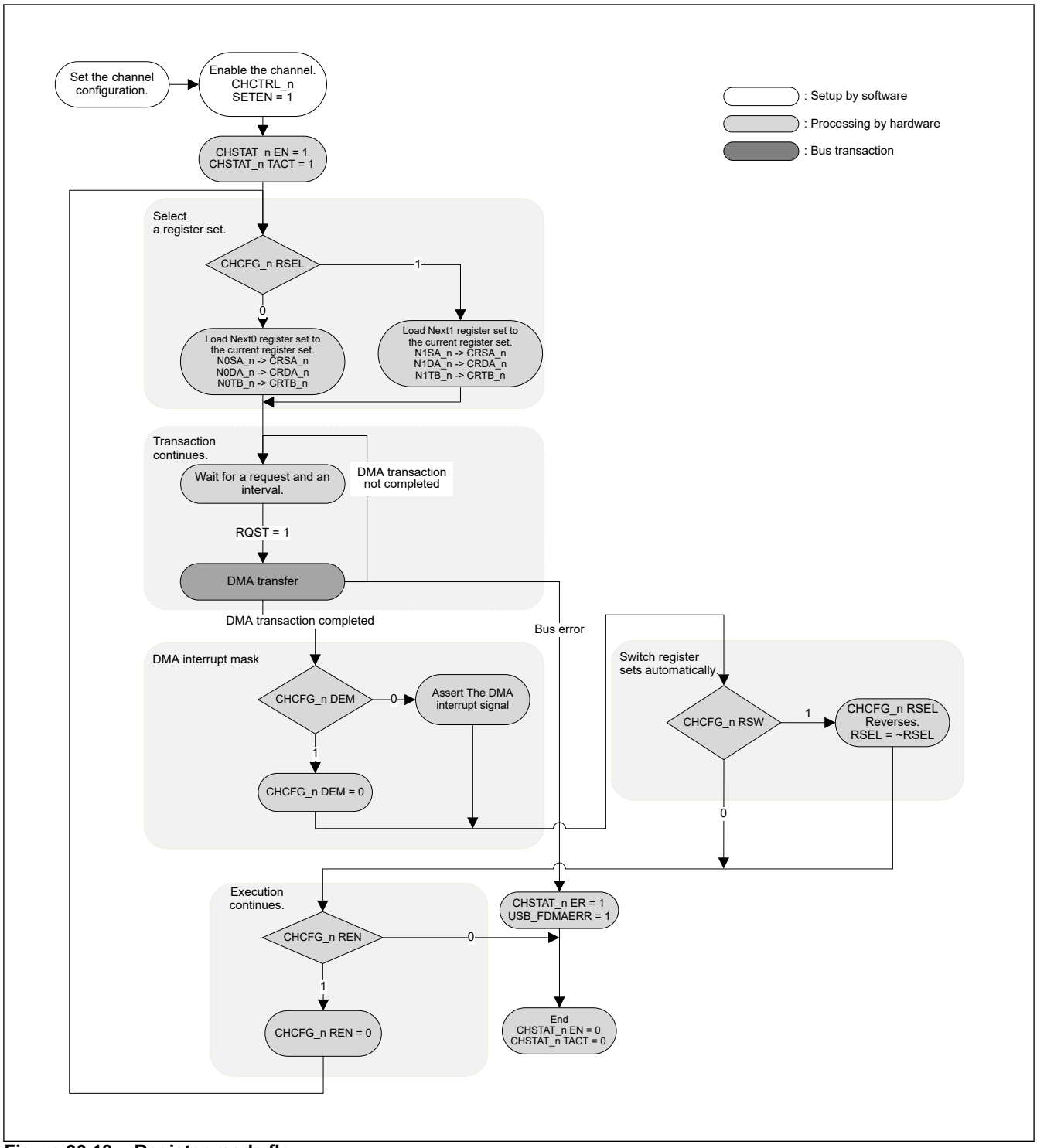


Figure 30.18 Register mode flow

<Description of register mode flow>

1. Channel setting (set channel configuration)
   Set Next0 or Next1 register set (the transfer destination address, the transfer source address, and the total number of transfer bytes). Also set the USB-controlled FIFO channels and the amount of transfer used for channel register set. (See section 30.15. DMA Transfer.)
2. Selecting register set

When 1 is written to the SETEN bit in the CHCTRL_n register, the EN bit and the TACT bit in the CHSTAT_n register are set to 1, and the set value of the next register set selected by the RSEL bit in the CHCFG_n register is loaded to the current register set.

3. DMA transaction (processing transaction)
   A DMA transaction is performed according to the set value. For details on transfer, see section 30.15. DMA Transfer.
4. Making USB_FDMAn (USB_FDMAn mask)
   The USB_FDMAn is masked according to the set DEM bit value in the CHCFG_n register. When DEM = 1, the USB_FDMAn is masked. Immediately after that, the DEM bit is automatically cleared to 0.
5. Switching register sets automatically
   Switching to the other next register set is determined according to the set RSW bit value in the CHCFG_n register.
6. Continuous execution
   Whether to perform DMA transfers continuously is determined according to the set REN bit value in the CHCFG_n register. When REN = 0, the EN bit and the TACT bit in the CHSTAT_n register are cleared to 0, and the DMAC stops operation. When REN = 1, DMA transfers are continued. Immediately after that, the REN bit is automatically cleared to 0.

(2) Setting Register Mode

Register mode settings

Select the register set to execute.

Table 30.36 Register mode settings

DMS (CHCFG_n)	RSEL (CHCFG_n)	Description
0	0	Executes Next0 register set.
	1	Executes Next1 register set.

USB_FDMAn mask settings

The USB_FDMAn can be masked.

Table 30.37 USB_FDMAn mask settings

DEM (CHCFG_n)	Description
0	When a DMA transaction is completed, the USB_FDMAn is asserted.
1	Even if a DMA transaction is completed, the USB_FDMAn is not asserted. Upon completion of a DMA transaction, the DEM bit is cleared to 0 by hardware.

Settings for executing register sets automatically

After a DMA transaction is completed, another DMA transaction can be performed successively.

Table 30.38 Settings for automatic execution of register sets

REN (CHCFG_n)	Description	Remarks
0	When a DMA transaction of the register set that is set in the RSEL bit is completed, the EN bit is cleared to 0 and the DMA operation ends.	Set this value if you want to perform a DMA transaction once.
1	After a DMA transaction is completed, a DMA transfer is performed for the register set that was selected to be performed successively. When the successive transfer is performed, the REN bit is cleared to 0.	Set this value if you want to perform DMA transactions for register sets successively.

Settings for switching register sets automatically

After a DMA transaction is completed, the register set to be used for the next DMA transaction can be switched.

Table 30.39 Settings for automatic switching of register sets

RSW (CHCFG_n)	Description	Remarks
0	After a DMA transaction is completed, register sets are not switched.	Set this value when you want to use only one register set.
1	When a DMA transaction is completed while REN = 1, the RSEL bit is reversed automatically and the other register set is selected.	Set this value when you want to switch register sets.

(3) Register Mode Setting Examples

Using the Next0 register set only

Table 30.40 Register mode setting example 1

DMS (CHCFG_n)	RSEL (CHCFG_n)	DEM (CHCFG_n)	RSW (CHCFG_n)	REN (CHCFG_n)
0 (Register mode)	0 (Next0)	0 (Do not mask)	0 (Do not switch)	0 (Continuous execution not in progress)

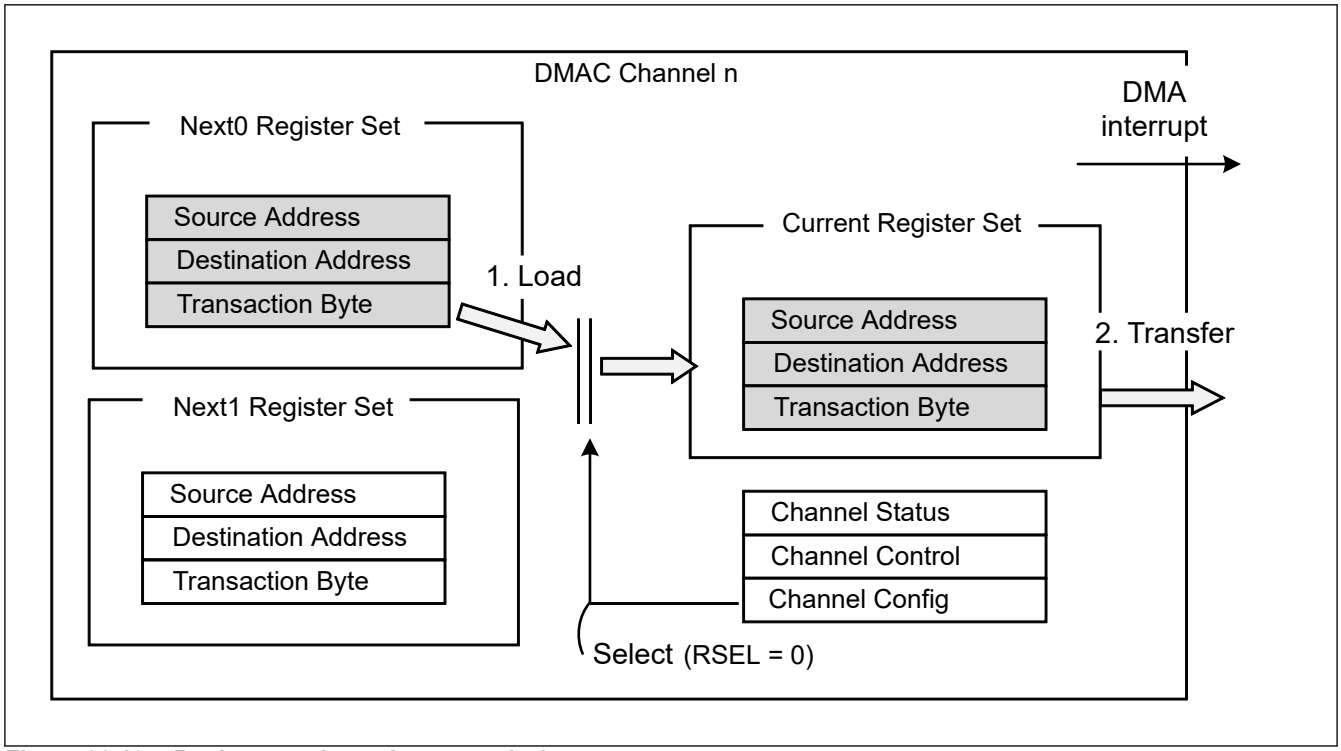


Figure 30.19 Register mode setting example 1

1. Writing 1 to the SETEN bit in the CHCTRL_n register sets the EN bit in the CHSTAT_n register to 1, and loads Next0 register set to the current register set.
2. According to the values set for the current register set and the channel register set, a DMA transaction is performed.
3. Because the DEM bit in the CHCFG_n register is 0, USB_FDMAn is asserted after a DMA transaction is completed.
4. Because the REN bit in the CHCFG_n register is 0, the EN bit in the CHSTAT_n register is cleared to 0, and the operation ends.

Using two register sets successively

Table 30.41 Register mode setting example 2

DMS (CHCFG_n)	RSEL (CHCFG_n)	DEM (CHCFG_n)	RSW (CHCFG_n)	REN (CHCFG_n)
0 (Register mode)	0 (Next0)	1 (Mask)	1 (Switch)	1 (Continuous execution in progress)

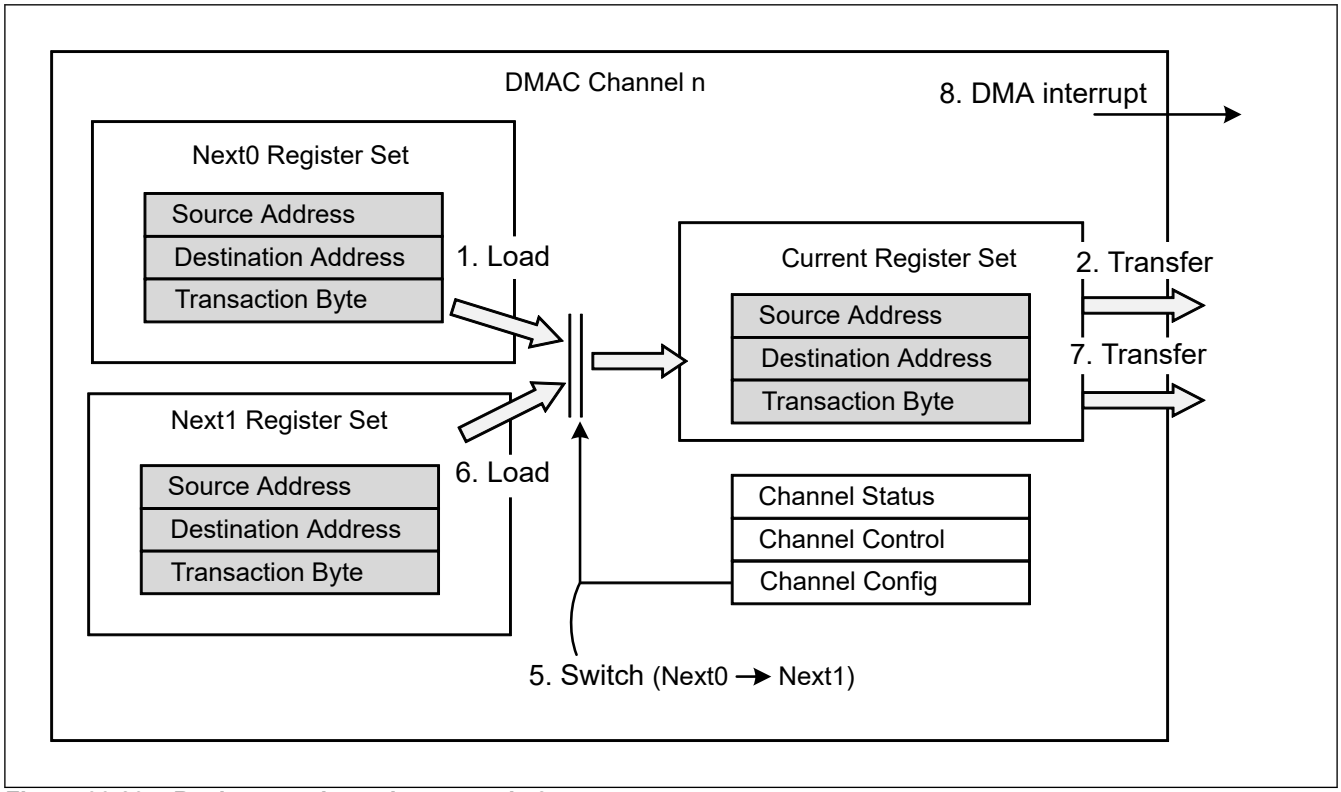


Figure 30.20 Register mode setting example 2

1. Writing 1 to the SETEN bit in the CHCTRL_n register sets the EN bit in the CHSTAT_n register to 1, and loads Next0 register set to the current register set.
2. According to the values of the current register set and the channel register set, a DMA transaction is performed.
3. Because the DEM bit in the CHCFG_n register is 1, USB_FDMAn is not asserted after a DMA transaction is completed, and the DEM bit is automatically cleared to 0.
4. Because the REN bit in the CHCFG_n register is 1, DMA transactions are performed successively, and the REN bit is automatically cleared to 0.
5. Because the RSW bit in the CHCFG_n register is 1, the register set to be executed next is switched (RSEL = 0 → 1).
6. Loads Next1 register set to the current register set.
7. According to the values of the current register set and the channel register set, DMA transactions are performed.
8. Because the DEM bit in the CHCFG_n register is 0, USB_FDMAn is asserted after a DMA transaction is completed.
9. Because the REN bit in the CHCFG_n register is 0, the EN bit in the CHSTAT_n register is cleared to 0, and the operation ends.

30.14.1.2 Link Mode

In link mode, a DMA transaction is performed by reading a descriptor in an external memory area as a setting value. The DMAC incorporates the next link address (NXLA_n) register and the current link address (CRLA_n) register for each channel. Each of them is used to set the address of the descriptor to be executed next and to display the descriptor address of the current DMA transaction respectively.

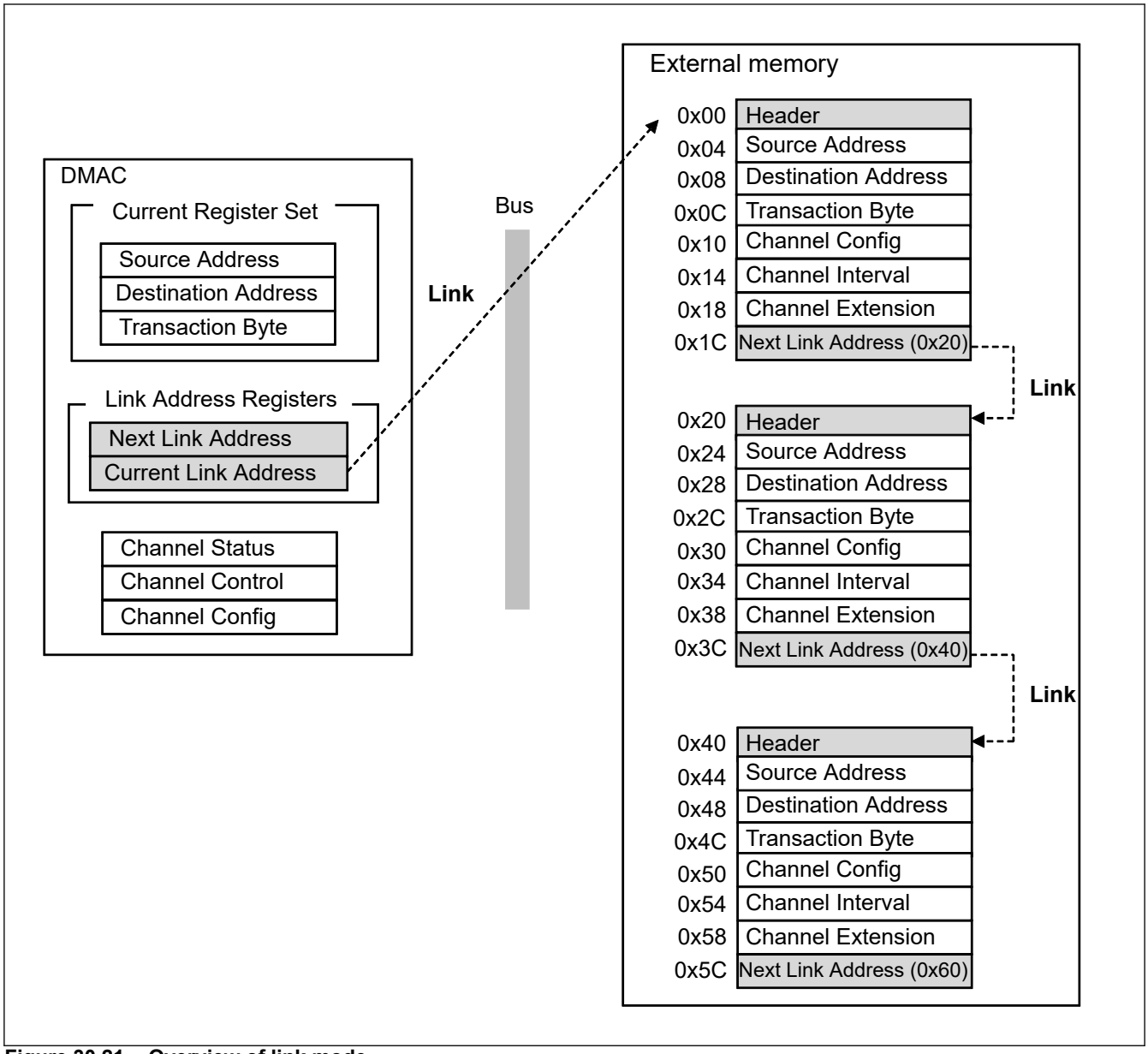


Figure 30.21 Overview of link mode

(1) Link Mode Operation Flow

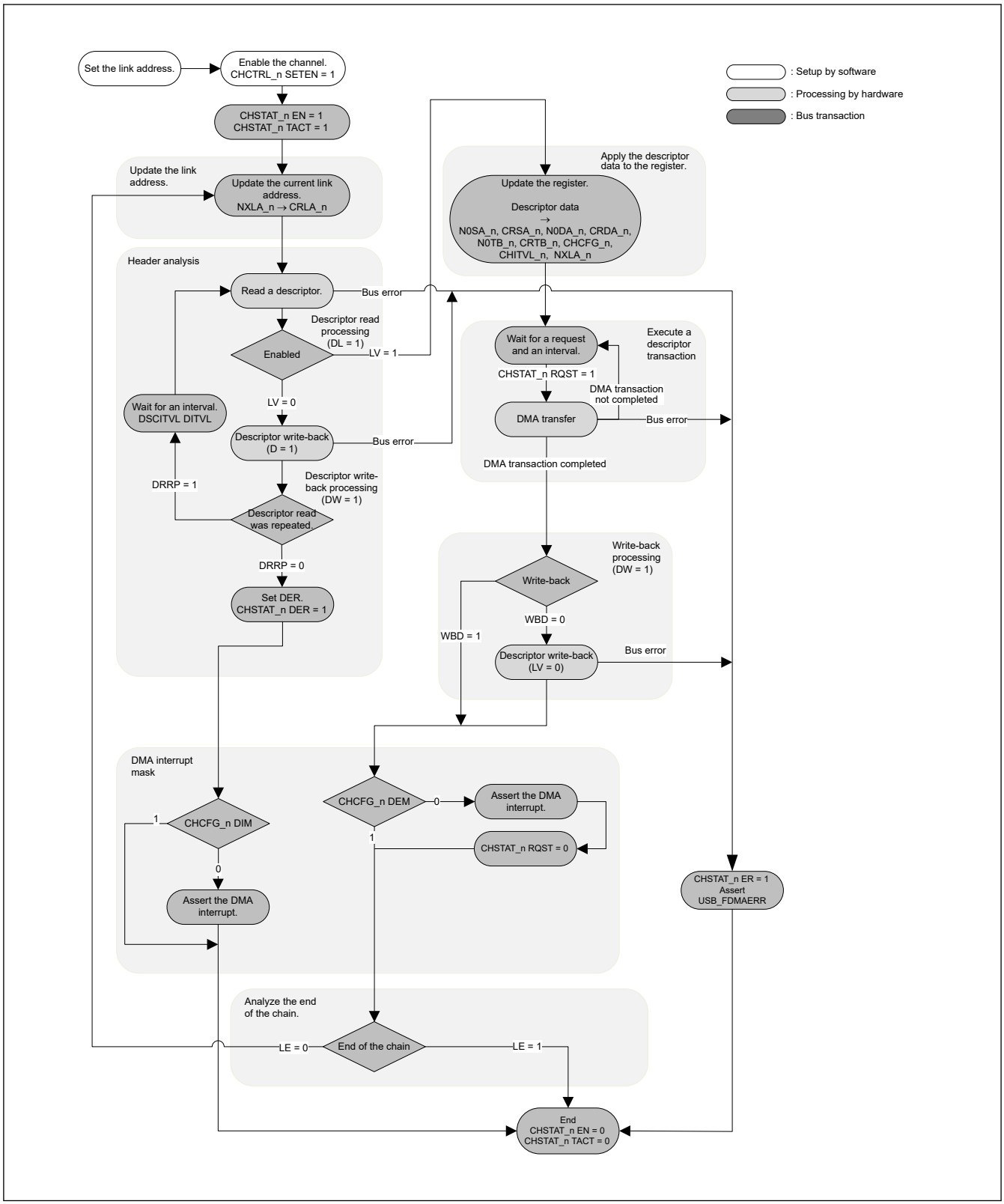


Figure 30.22 Link mode operation flow

Description of Link Mode Operation Flow

1. Channel setting
   Set the start address of the link destination in the NXLA_n register.

2. Updating the link address
   Writing 1 to the SETEN bit in the CHCTRL_n register sets the EN bit and the TACT bit in the CHSTAT_n register to 1, and loads the link address set for the NXLA_n register to the CRLA_n register.
3. Descriptor read and header judgment
   The DMAC starts reading a descriptor and checks the header contents. When LV = 0, 1 is written back to the D bit in the header. After that, when the DRRP bit in the CHCFG_n register = 1, the same descriptor is read again after the number of cycles set in the DSCITVL register elapsed. When DRRP = 0, DER in the CHSTAT_n register is set to 1. This indicates the end state (EN = 0 and TACT = 0 in the CHSTAT_n register). At that time, if the DIM bit in the CHCFG_n register is 0, USB_FDMAn is asserted.
4. Descriptor setting
   When LV = 1, the read descriptor data is loaded to the current register set and the channel register set. In addition, the next link destination is loaded to the NXLA_n register.
5. DMA transaction
   According to the set values, a DMA transaction is performed. For details on transfers, see section 30.15. DMA Transfer.
6. Writing back of header
   When WBD in the header = 0, the DMAC writes LV = 0 back to the header area.
7. USB_FDMAn mask
   When the DEM bit in the CHCFG_n register is 0, USB_FDMAn is asserted.
8. Link end judgment
   When LE in the header = 1, the EN bit and the TACT bit in the CHSTAT_n register are cleared to 0, and the DMAC stops operation. When LE = 0, the current register set is updated and the next descriptor read is started.

(2) Register Settings

Link mode settings

When using link mode, set the DMS bit in the CHCFG_n register to 1.

Table 30.42 Link mode settings

DMS (CHCFG_n)	Description
1	Operates in link mode.This bit cannot be modified by using a descriptor.

LINK address settings

As registers that indicate the link destination, the next link address (NXLA_n) register and the current link address (CRLA_n) register are available.

To start link mode, set the link destination in the NXLA_n register.

The NXLA_n register is updated to the next link after a descriptor is read. In addition, the CRLA_n register indicates the link address of the currently executed descriptor.

Table 30.43 Link address register set

Register	Description
Next link address register (NXLA_n)	Sets and displays the next link destination. Before starting link mode, set the address of the link destination in this register.
Current link address register (CRLA_n)	Displays the currently executed link destination. This register is read-only.

(3) Descriptor Settings

The DMAC supports multiple descriptor formats. To switch formats, use the DSCFM field of bits [31:28] in the first word (header) of a descriptor.

The following table shows the relationship between the DSCFM bit value and descriptor formats.

Table 30.44 Descriptor format

DSCFM	Descriptor size	Next link address	Channel extension	Channel interval	Channel config	Transaction size	Destination address	Source address	Header
0x3	Four words	Y	— (Reload)	— (Reload)	— (Reload)	— (Header)	Y	Y	Y (with STS)
0x1	Eight words	Y	Y	Y	Y	Y	Y	Y	Y (without STS)
Other than above	If DSCFM is set to a value other than 1 and 3, operation is not guaranteed. Do not set a value other than 0x1 and 0x3 for DSCFM.								

Table 30.45 Description of availability

Field	Availability	Description	Remarks
Header	Y(with STS)	Indicates that the STS field of [15:0] in the header is enabled. The value set in the STS field is used as the total number of transfer bytes (transaction size).	—
	Y(without STS)	The STS field of [15:0] in the header is disabled. The transaction size of the descriptor is used as the total number of transfer bytes.	—
Source address	Y	Specifies the source address.	—
Destination address	Y	Specifies the destination address.	—
Transaction size	Y	Specifies the transaction size.	—
	— (Header)	Omits the transaction size. The value set in the STS field in the header is used as the total number of transfer bytes (transaction size).	Because the STS field is 16 bits, up to 65,535 bytes can be set.
Channel config Channel interval Channel extension	Y	Specifies the channel config, channel interval, and channel extension.	—
	— (Reload)	Omits the channel config, channel interval, and channel extension. Inherits the previous setting values (of the CHCFG_n, CHITVL_n, and CHEXT_n registers at that time).	—
Next link address	Y	Specifies the next descriptor address (next link address) to be read after a DMA transfer of this descriptor.	—

The DMAC interprets data obtained through descriptor read in order. If a value less than eight words is specified in the DSCFM field, place descriptor data marked with Y in Table 30.44 in memory.

Table 30.46 Descriptor placement example

	Address (link address + n)							
DSCFM	+ 0x1C	+ 0x18	+ 0x14	+ 0x10	+ 0x0C	+ 0x08	+ 0x04	+ 0x00
0x3	—	—	—	—	Next link address	Destination address	Source address	Header
0x1	Next link address	Extension	Interval	Config	Transaction byte	Destination address	Source address	Header

Header

The header indicates descriptor states as shown below.

This area is read by the DMAC before starting a DMA transfer in link mode. Furthermore, the transfer status is written back by the DMAC after the DMA transfer.

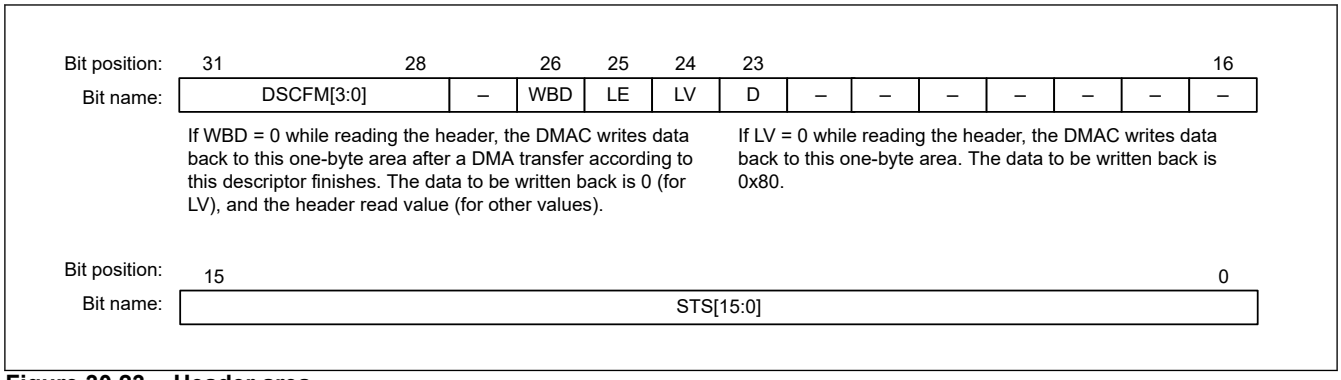


Figure 30.23 Header area

Table 30.47 Header area

Bit position	Bit name	Description
15:0	STS	Short Transaction Size When DSCFM = 0x3, these bits set the transaction size in bytes. Up to 65535 bytes can be set. When DSCFM = 0x3, do not set the STS bits to 0. If 0 is set, the operation is not guaranteed.
22:16	—	A reserved area. Set 0.
23	D	Descriptor Error Indicates an access error of a descriptor. When LV = 0 while a descriptor is being read, the DMAC writes 1 back to this bit. 0: No descriptor error 1: LV = 0 during descriptor read
24	LV	Link Valid Indicates that this descriptor is enabled. When WBD = 0, the DMAC writes 0 after the DMA transaction written in the descriptor is completed. Set 1 when setting header. 0: The descriptor is disabled. 1: The descriptor is enabled.
25	LE	Link End Indicates that the link ends with the DMA transaction of this descriptor. To indicate the end of the link, set this bit to 1. 0: The link continues. 1: The link ends.
26	WBD	Write Back Disable Masks write-back execution of the LV bit. When this bit is 1, the DMAC does not perform writeback operation. 0: The LV bit is written back to 0. 1: The LV bit is not written back.
27	—	A reserved area. Set 0.
31:28	DSCFM	Descriptor Format Specifies the descriptor format (descriptor length and combination). For details, see Table 30.44.

If descriptors are added sequentially while the DMAC is operating, an access of the CPU to set the LV bit to 1 and an access of the DMAC to write 1 back to the D bit may conflict. Because of this, the data that was written first is overwritten by the data that was written later.

To avoid this problem, the byte lane of the D bit and the byte lane of the LV bit are placed differently to separate write areas. The DMAC uses the byte write method to write back the D bit. Therefore, the CPU must set LV = 1 by using the byte write method.

Setting descriptors other than header

Data in descriptors except header has the same specifications as an on-chip register. For details on the on-chip register specifications, see section 30.2. Register Descriptions.

Setting AHB during descriptor access

The MHPROT pin output during the descriptor access can be set in the LWPR and LDPR fields in the DCTRL register. Set it according to the access destination for which descriptor is placed.

Descriptor areas and DMA transfer areas

Figure 30.24 provides an overview of descriptor areas and DMA transfer areas which the DMAC accesses.

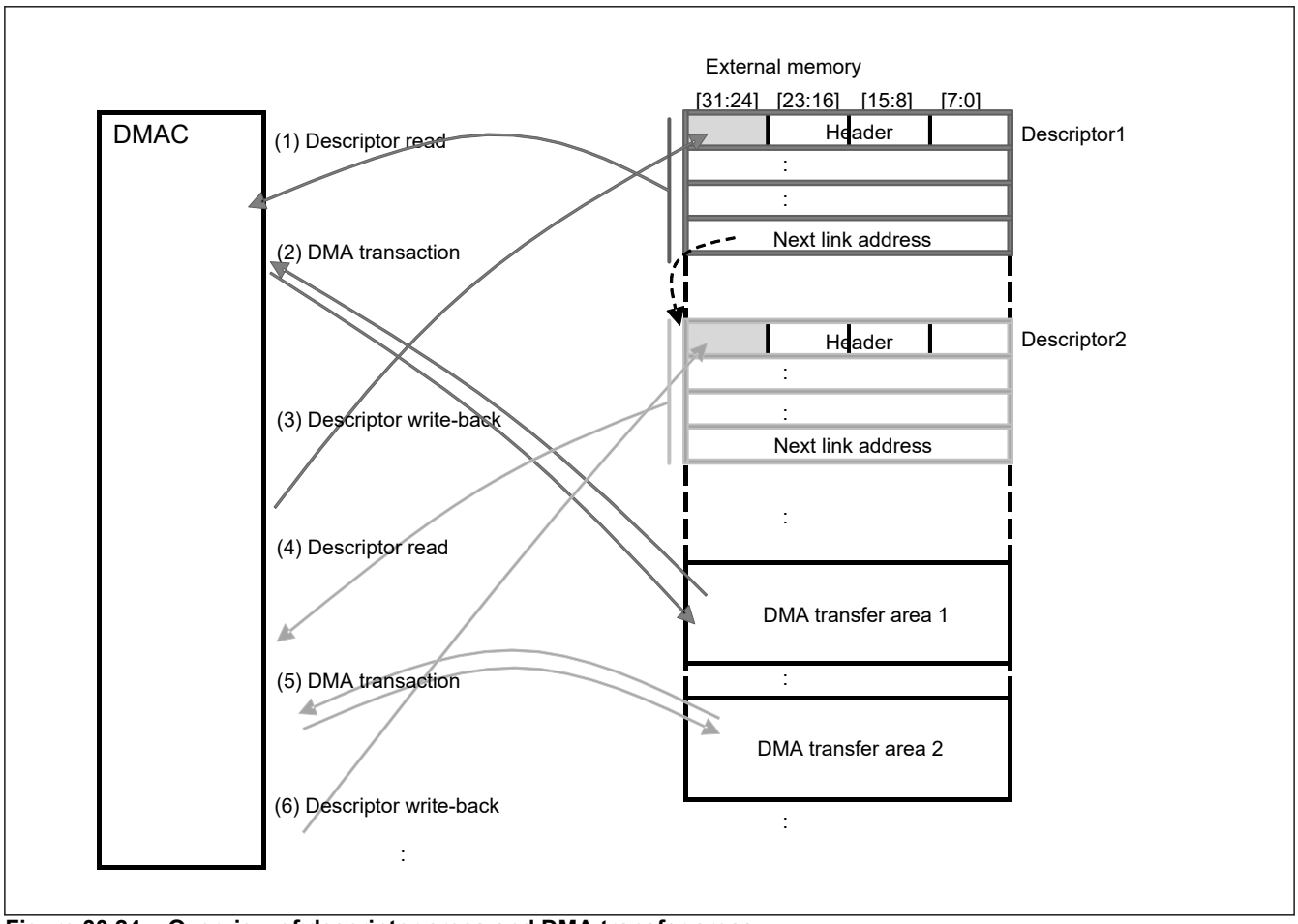


Figure 30.24 Overview of descriptor areas and DMA transfer areas

1. Descriptor read
   Loads the value set for the on-chip NXLA_n register to the CRLA_n register, and then reads the descriptor from the external memory space (descriptor 1) indicated by the CRLA_n register.
2. DMA transaction
   When the LV bit in the header of the descriptor is 1, performs a DMA transfer according to the descriptor information.
3. Descriptor write-back
   If the WBD bit in the header is 0 after a DMA transfer for the set number of bytes, data (LV bit = 0 and the data that was read in step 1 in other fields) is written back to header[31:24] of descriptor 1 with a byte size.
4. Descriptor read
   If the LE bit in the header of the descriptor that was read in step 1 is 0, reads the next descriptor from the address (descriptor 2) indicated by the next link address in the descriptor.
5. DMA transaction
   When the LV bit in the header of the descriptor is 1, performs a DMA transfer according to the descriptor information.
6. Descriptor write-back
   If the WBD bit in the header is 0 after a DMA transfer for the set number of bytes, data (LV bit = 0 and the data that was read in step 4 in other fields) is written back to header[31:24] of descriptor 2 with a byte size.

Hereafter, repeats steps 4 through 6.

When LE in the header is 1 and WBD = 0, a DMA transfer is performed with the descriptor settings, and 0 is written back to the LV bit in the header, and then operation ends.

When LE in the header is 1 and WBD = 1, a DMA transfer is performed with the descriptor settings, and then operation ends (without write-back).

When the LV bit in the header is 0, 1 is written back to the D bit in the header. After that, if the DRRP bit in the CHCFG_n register is 1, the descriptor is read again after the interval specified in the DITVL field in the DSCITVL_n register. When DRRP = 0, the operation stops.

Notes on descriptors

- In link mode, settings can be changed by reading a descriptor, but the timing for changing settings and for a hardware request cannot be synchronized. For this reason, when using a hardware request, set the AM, LVL, HIEN, LOEN, and SEL bits in the CHCFG_n register to 1 before setting the SETEN bit in the CHCTRL_n register. Besides, do not change these bits in a descriptor.
- In a descriptor, the settings of the DMS field in the CHCFG_n register cannot be changed (always link mode). In addition, settings of the REN, RSW, and RSEL fields in the CHCFG_n register can be changed in the descriptor, but that does not affect the operation.
- The DMAC references the DSCFM field and the LV bit in the header to determine if the descriptor is enabled or disabled. Therefore, initialize (DSCFM = 0x1 or 0x3 and LV = 1) the memory area equivalent to the DSCFM field and the LV bit of the descriptor before the DMAC accesses the memory area.
- If you want to set the next descriptor in the memory while the DMAC is operating, write 1 to the LV bit after setting the descriptors after header (source address, destination address, ... next link address). This is to avoid DMA transfer using descriptor values (source address, destination address, and so on) before the setting if descriptor settings by the CPU and descriptor read of the DMAC conflict and descriptor read by the DMAC interrupts descriptor settings by the CPU.
- If you want to keep the information written back to the D bit in the header, perform a byte access to write 1 to the LV bit in the header.

(4) LINK Configuration Example

In link mode, descriptors can be configured as shown below.

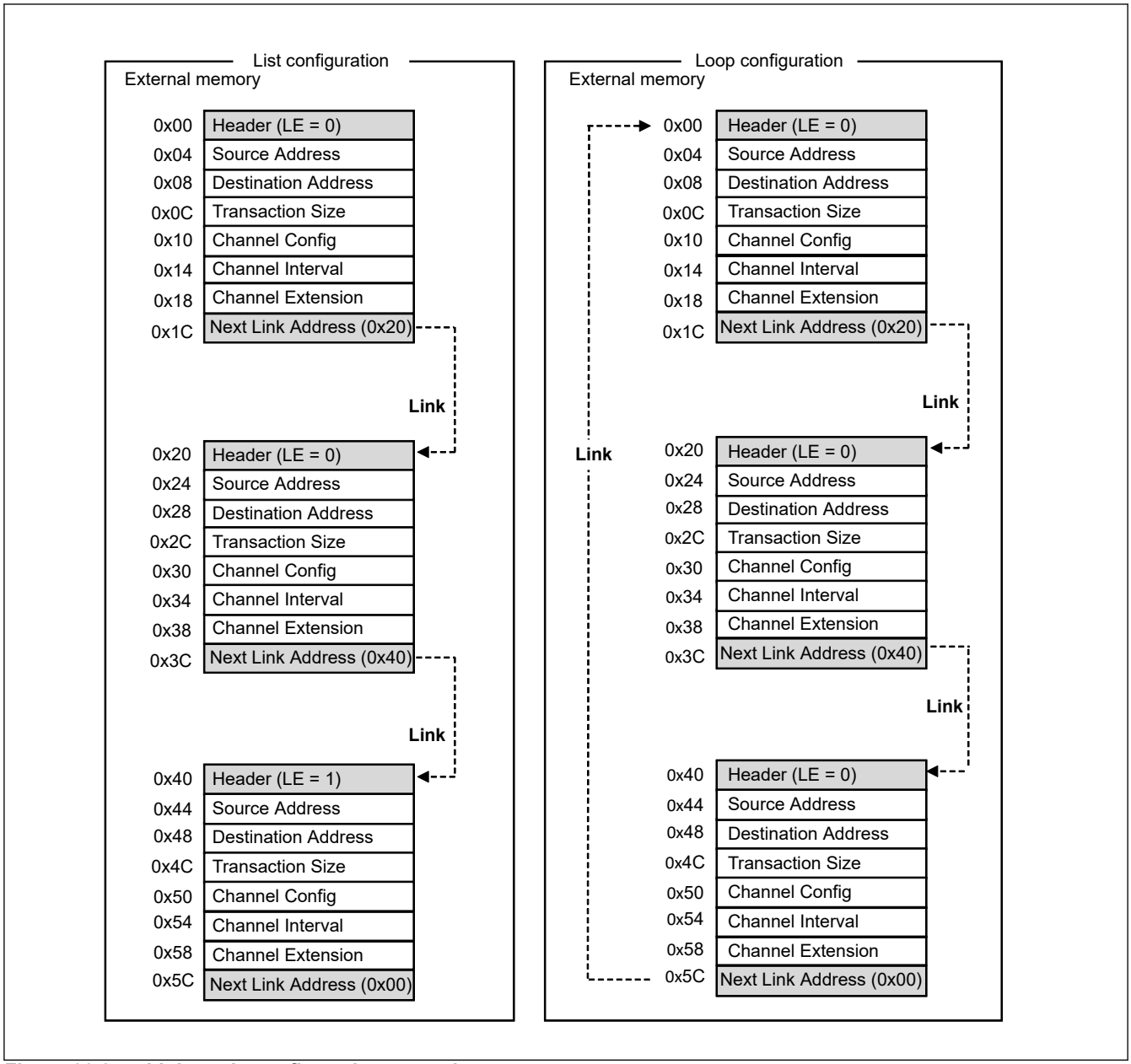


Figure 30.25 Link mode configuration example

List configuration

Writing 1 to the LE bit in the header of the last descriptor terminates the link.

Loop configuration

Setting the link destination of the last descriptor to the address of the previous descriptor configures descriptors in a loop. To terminate the loop, modify the LE bit in the header to 1 before the DMAC reads a descriptor, or follow the procedure to suspend a transfer.

30.14.1.3 Write-Only Mode

Setting the WONLY bit in the CHCFG_n register to 1 enables write-only mode.

Table 30.48 Write-only mode settings (1 of 2)

WONLY(CHCFG)	Mode	Description
0	Normal mode	Performs DMA transfers with the values set for the next register set.

Table 30.48 Write-only mode settings (2 of 2)

WONLY(CHCFG)	Mode	Description
1	Write-only mode	Performs DMA write transfers only without performing DMA read transfers.

In write-only mode, read operation for DMA transfers is not performed (descriptor read is performed in the same way as normal mode). In register mode, use the values set for the NxSA_n register (x = 0 when RSEL = 0, and x = 1 when RSEL = 1) as write data. In link mode, use the value in the SA field of a descriptor as write data.

Use this mode for initializing a memory area.

30.15 DMA Transfer

This section describes basic operation of DMA transfer.

30.15.1 Transfer Mode

Only single transfer mode is supported.

When a DMA transfer request from the USB control is accepted, a DMA transfer is performed on the side (transfer source or transfer destination) indicated by the REQD bit in the CHCFG_n register to assert the internal DMA acknowledge signal from the internal USB control to the DMAC control. Each time a transfer request is accepted, transfer proceeds. This operation is repeated until it reaches the transfer size loaded to the CRTB_n register. (Arbitration between channels is performed for each DMA transfer.)

The DMA acknowledge timing differs depending on the setting for the REQD bit in the CHCFG_n register or settings for the transfer size (DDS[2:0] and SDS[2:0] in the CHCFG_n register). For details, see section 30.15.7. Differences in Operation According to the Transfer Data Size.

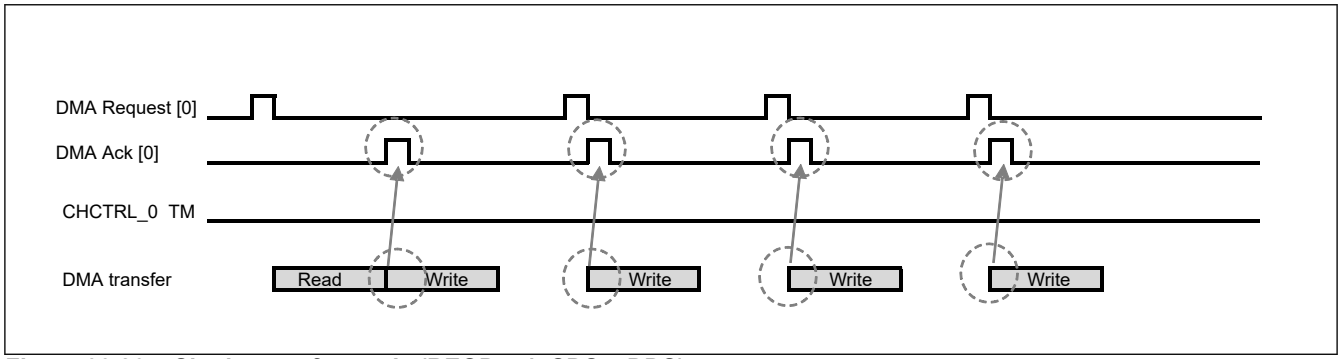


Figure 30.26 Single transfer mode (REQD = 1, SDS > DDS)

30.15.2 DMA Channel Priority Control

As an arbitration method between channels, fixed priority mode and round-robin mode are supported. To select a mode, use the PR bit in the DCTRL register. When the PR bit is 0, fixed priority mode is selected. When the PR bit is 1, round-robin mode is selected.

Table 30.49 Priority control settings

Mode	PR (DCTRL)	Description	Application
Fixed priority	0	Controls requests in fixed priority (high: CH0 > CH1: low) mode.	Use this mode if channels have priority.
Round-robin	1	Controls requests in round-robin mode.	Use this mode if you want to execute requests equally.

30.15.2.1 Fixed Priority Mode

In fixed priority mode, the priority among channels is fixed as shown below.

[High] CH0 > CH1 [Low]

If DMA transfer requests are generated in multiple channels simultaneously, a DMA transfer request with a smaller channel number takes precedence. The following figure provides an example where another DMA transfer request with higher priority is generated during a DMA transfer in fixed priority mode.

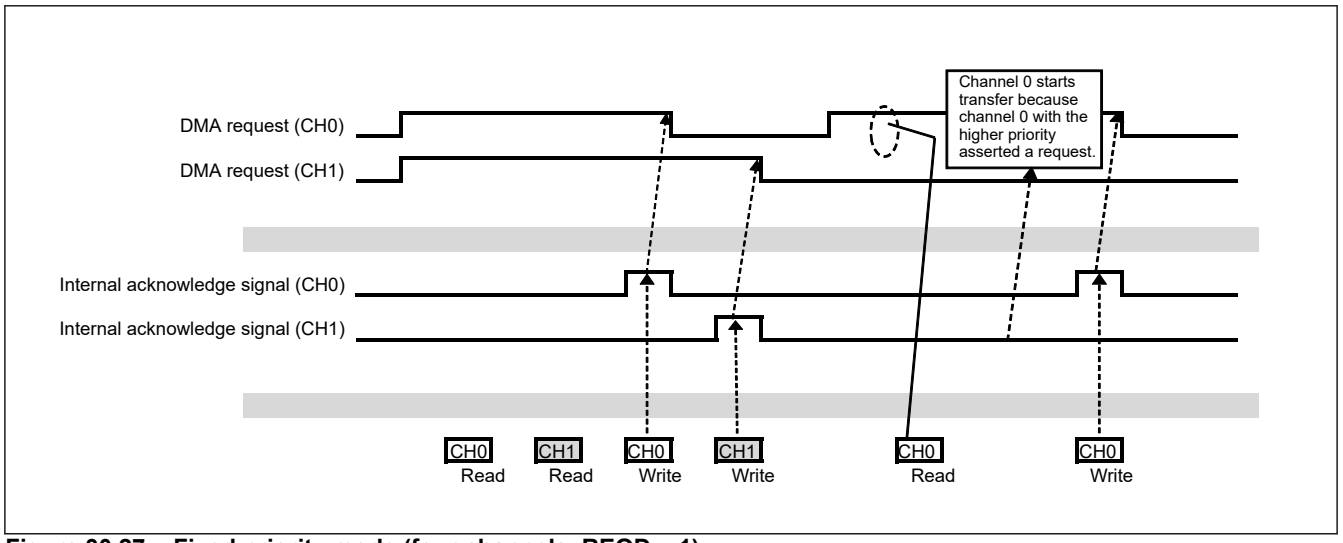


Figure 30.27 Fixed priority mode (four channels, REQD = 1)

Note: Channel 0 is handled with the highest priority, but the transfer with the next highest priority is performed to increase the bus usage rate while the transfer of channel 0 switches.

30.15.2.2 Round-Robin Mode

In round-robin mode, priority is changed each time a transfer of each channel is accepted so that the lowest priority is given to the channel in which the last transfer is performed.

The priority immediately after a reset is the same as the one in fixed priority mode as shown below.

[High] CH0 > CH1 [Low]

In this state, if no transfer request is generated in DMA channel 0, but a request is generated in DMA channel 1, a transfer of DMA channel 1 is performed. The priority after the transfer is shown below.

[High] CH1 > CH0 [Low]

The following figure provides an example of a DMA transfer in round-robin mode.

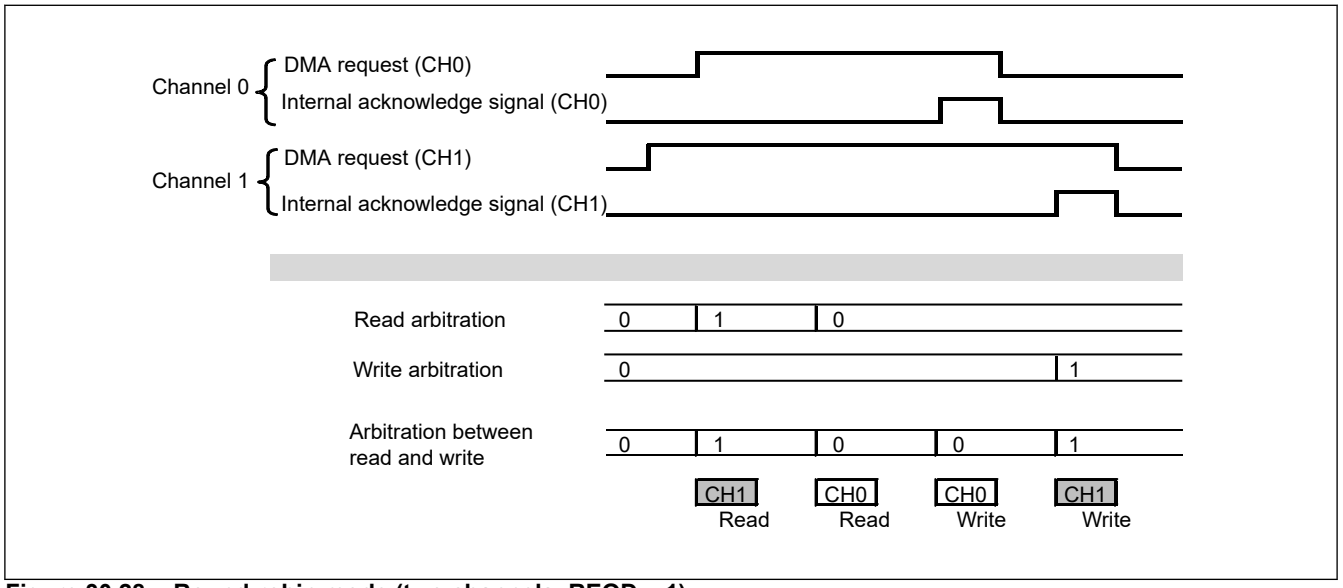


Figure 30.28 Round-robin mode (two channels, REQD = 1)

Note: In the DMAC, arbitration between read channels and arbitration between write channels are performed. As a result of these arbitrations, further arbitration is performed, and a bus access is issued.

30.15.3 Forced Ejection Request

When a forced ejection request is input, data that is not yet transferred in a buffer is transferred to the DMA transfer destination address. After data is flushed, the DMA transfer resumes.

The following describes notes on forced ejection requests:

- If a forced ejection request conflicts with a USB control DMA transfer request, the forced ejection takes precedence, and then the DMA transfer is performed.
- When the transfer destination is a system on the USB control side (the REQD bit in the CHCFG_n register = 1), the unit on the DMA transfer destination side receives data even if the USB control side does not request a DMA transfer. Therefore, a malfunction, such as buffer overflow, may occur. Because of this, when REQD =1, no forced ejection is performed by hardware.
- Differences from flush mode described in (2)Aborting a Transfer (with Buffer Flush: SBE = 1) (The EN bit is cleared to 0 when the SBE bit in the CHCFG_n register = 1.)
  Flush mode: The DMAC stops operation after data in a buffer is written.
  Forced ejection request: A DMA transfer continues after flush operation ends.

30.15.3.1 Software Forced Ejection Request

For software forced ejection requests, use the SETSSWPRQ bit in the CHCTRL_n register. To make a forced ejection request, write 1 to the SETSSWPRQ bit. The DMAC outputs data in a buffer to the DMA transfer destination.

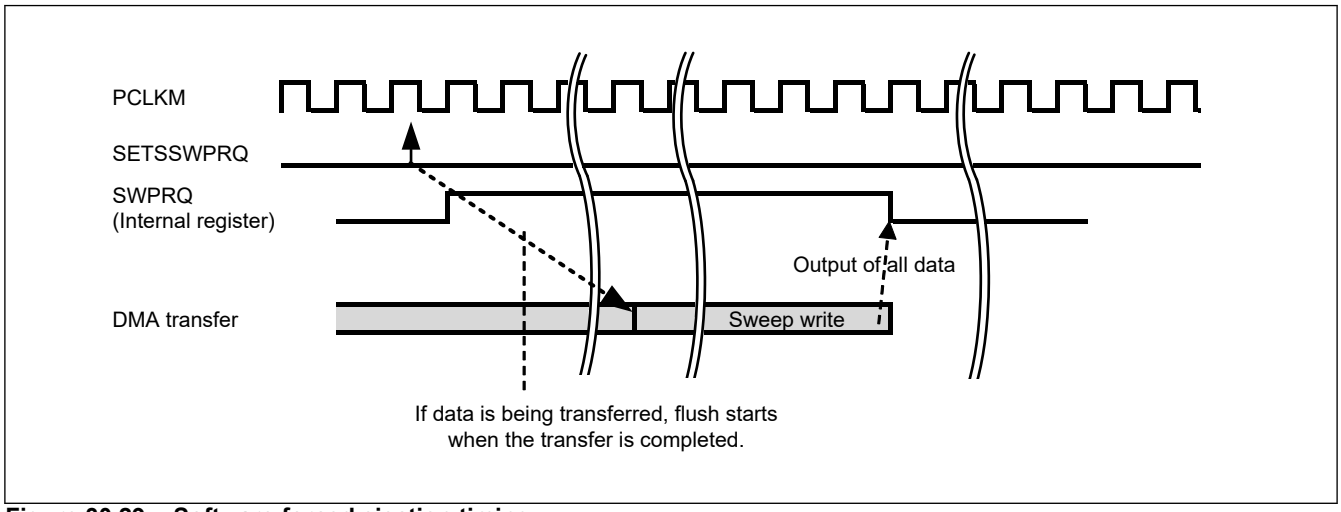


Figure 30.29 Software forced ejection timing

30.15.4 DMA Transfer Completion Interrupt (USB_FDMAn)

The USB_FDMAn[1:0] interrupt request signals indicate that a DMA transaction is completed.

Each bit of USB_FDMAn[1:0] corresponds to each channel.

When a transfer for the total number of transfer bytes loaded to the CRTB_n register is completed with an OKAY response, the END bit in the CHSTAT_n register is set to 1. If the DEM bit in the CHCFG_n register = 0 at this time, USB_FDMAn is asserted to high level (n = 1, 0). (When write-back is performed in link mode, this pin is asserted after the write-back operation.)

In addition, when the DRRP bit in the CHCFG_n register = 0 in link mode and the header of the read descriptor is LV = 0, the DER bit in the CHSTAT_n register is set to 1. If the DIM bit in the CHCFG_n register = 0 at this time, USB_FDMAn is asserted to high level.

Use these signals to detect a transfer completion interrupt in the interrupt controller.

Table 30.50 USB_FDMAn assert conditions

Source	Condition	USB_FDMAn mask signal
DMA transaction completed	When a transfer for the total number of transfer bytes loaded to the CRTB_n register is completed with an OKAY response (after write-back when write-back is performed in link mode)	DEM bit in the CHCFG_n register
Descriptor invalid	When the header of the read descriptor is LV = 0 while DRRP and DIM in the CHCFG_n register are 0 in link mode	DIM bit in the CHCFG_n register

30.15.5 DMA Error Interrupt (USB_FDMAERR)

If an error response is received during a DMA transfer or descriptor access, this module determines it as an error and aborts the transfer. When an error response is received, the EN bit in the CHSTAT_n register for channel n where a transfer is in progress is cleared to 0 and the ER bit is set to 1 (n = 1, 0). In addition, high level is output from the USB_FDMAERR pin.

The USB_FDMAERR signal cannot be masked.

Data for a series of error transfers is not guaranteed. Be sure to perform the following procedure to retry the transfer from the beginning.

1. Set the SWRST bit in the CHCTRL_n register to 1.
2. Set each register again.

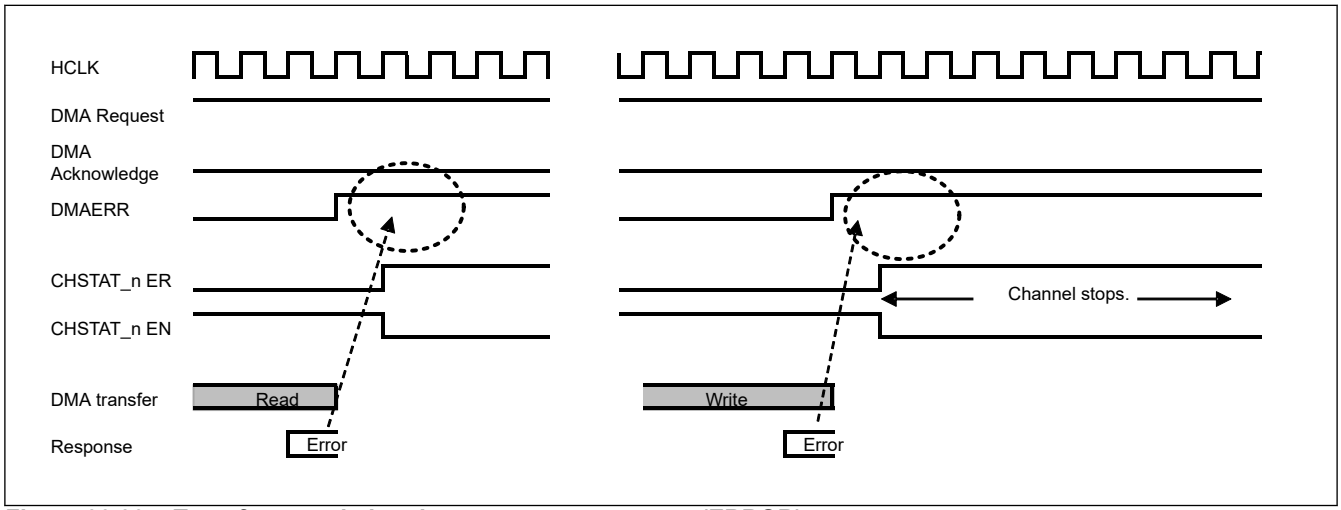


Figure 30.30 Transfer stop timing due to an error response (ERROR)

Note: The USB_FDMAERR pin output is not asserted even if an error occurs during an access to the slave of this controller.

30.15.6 Interval Count Function

By the setting for the ITVL field in the CHITVL_n register, the execution interval of DMA transfers can be adjusted. This function is used to avoid occupation of a bus by the DMAC. Until the count value becomes 0, no DMA transfer for the next DMA request is performed.

The following figure provides an operation example.

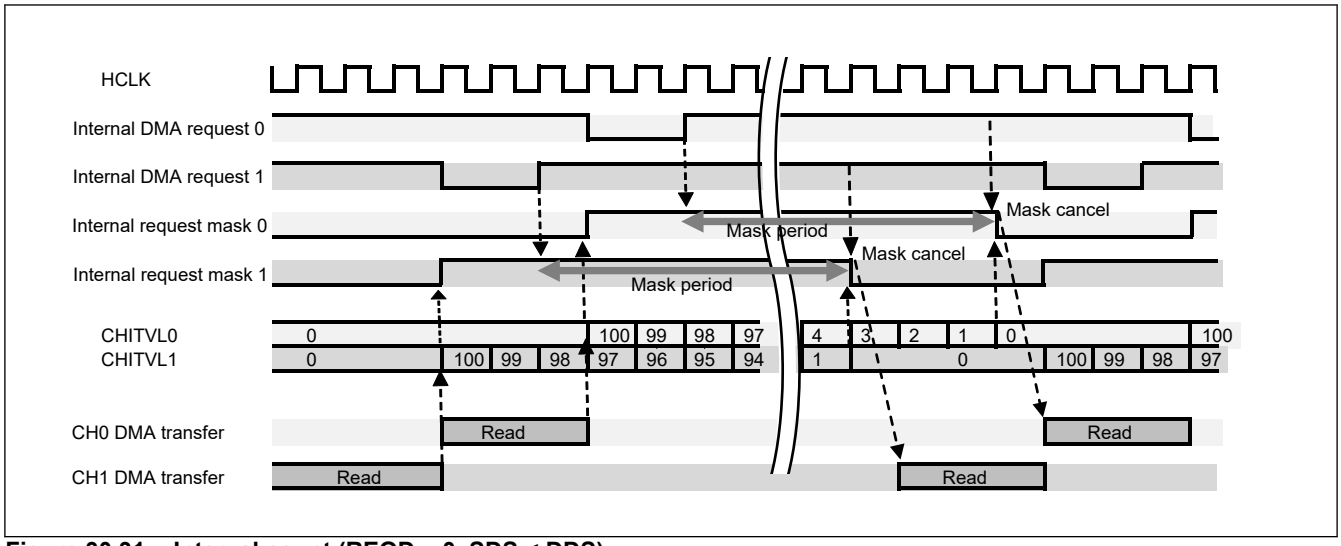


Figure 30.31 Interval count (REQD = 0, SDS < DDS)

An interval is inserted after a transfer specified by the REQD bit in the CHCFG_n register.

The following figure shows the relationship between the setting values for the REQD, SDS, and DDS bits in the CHCFG_n register and the cycles applied to intervals.

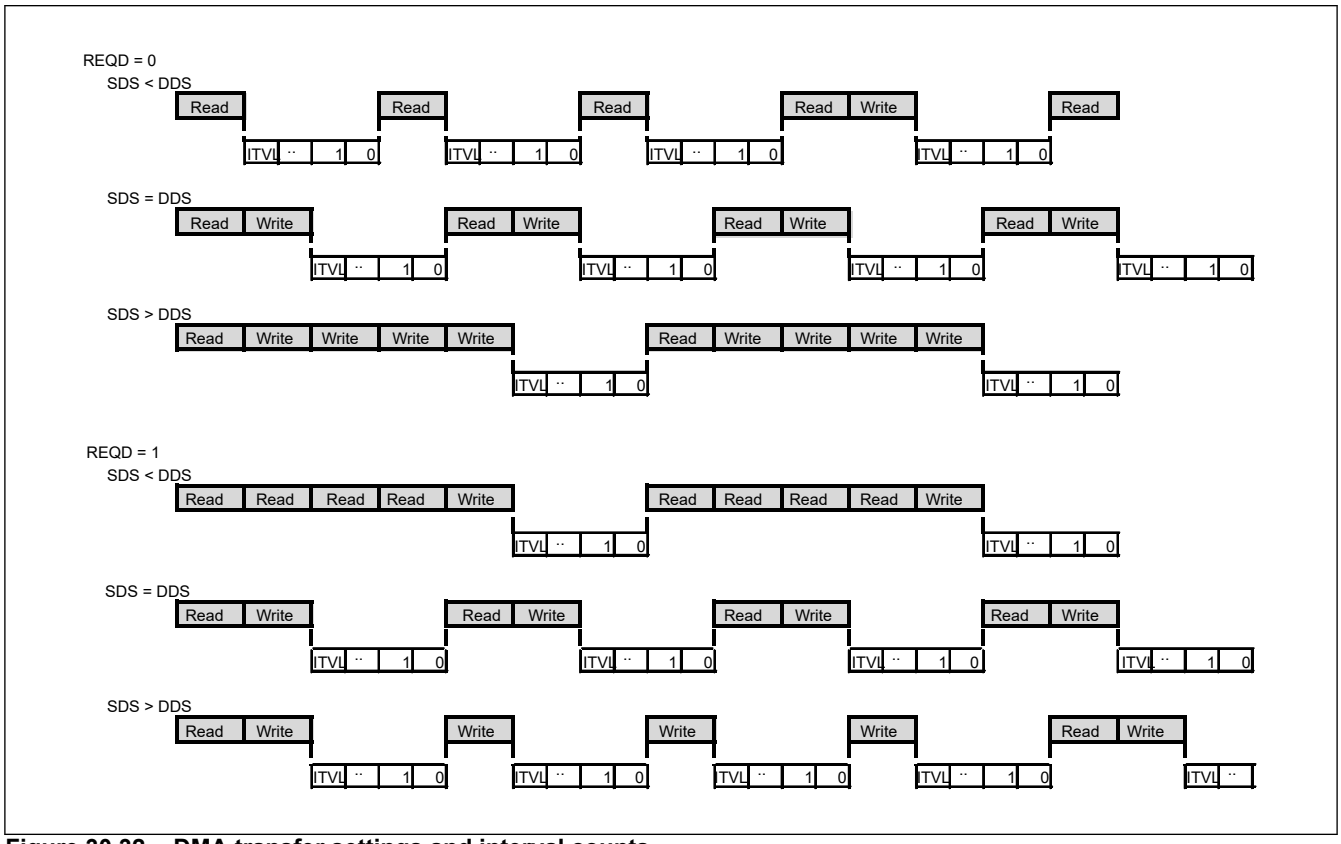


Figure 30.32 DMA transfer settings and interval counts

30.15.7 Differences in Operation According to the Transfer Data Size

30.15.7.1 When the Transfer Data Size on the Transfer Source Side is Small

When read operation for the transfer data size on the transfer destination side is completed, write operation to the transfer destination starts.

The following figure shows the timing chart (rising edge detection) when the transfer source data size is 8 bits and the transfer destination data size is 32 bits (SDS = 0 and DDS = 2 in the CHCFG_n register).

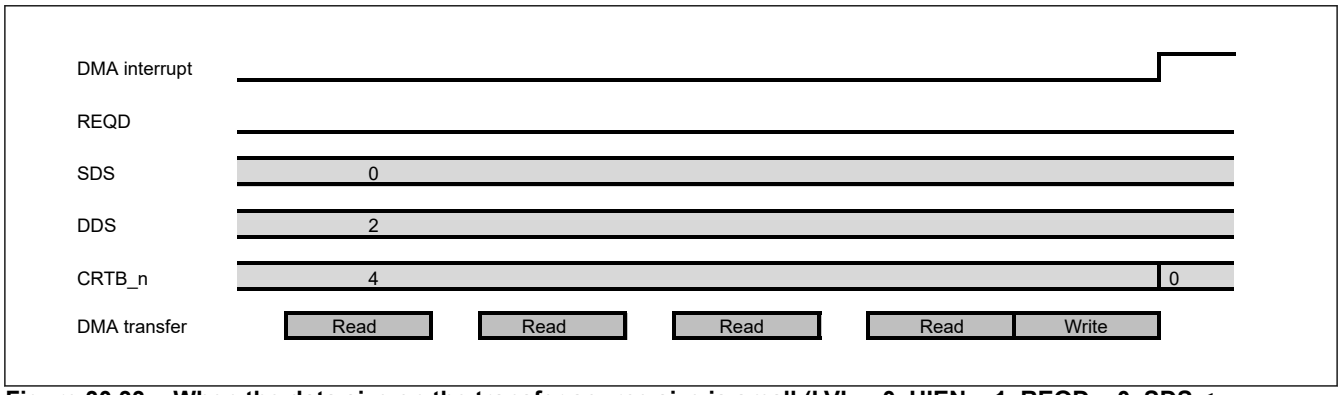


Figure 30.33 When the data size on the transfer source size is small (LVL = 0, HIEN = 1, REQD = 0, SDS < DDS in CHCFG_n)

30.15.7.2 When the Transfer Data Size on the Transfer Destination is Small

Because the transfer data size on the transfer source side is large, after a single read operation of the transfer source, write operation to the transfer destination is performed a few times. The following figure shows the timing chart (rising edge detection) when the transfer source data size is 64 bits and the transfer destination data size is 16 bits (SDS = 3 and DDS = 1 in the CHCFG_n register).

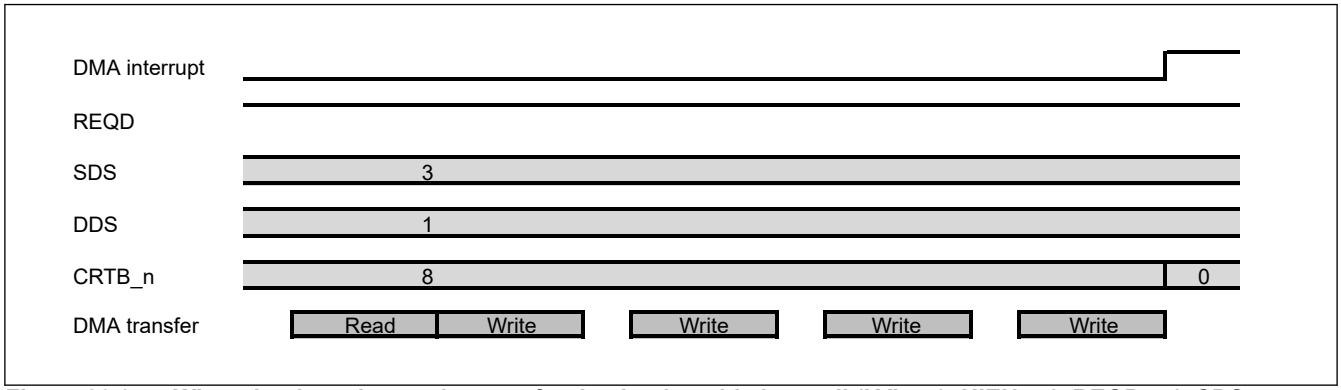


Figure 30.34 When the data size on the transfer destination side is small (LVL = 0, HIEN = 1, REQD = 1, SDS > DDS in CHCFG_n)

30.15.7.3 When Transfer Data Sizes on the Transfer Source Side and the Transfer Destination Side are Equal

Each time a DMA transfer request is detected, read operation is performed on the transfer source side and write operation is performed on the transfer destination side. The following figure shows the timing chart (rising edge detection) when the transfer source data size and the transfer destination data size are 8 bits (SDS = 0 and DDS = 0 in the CHCFG_n register).

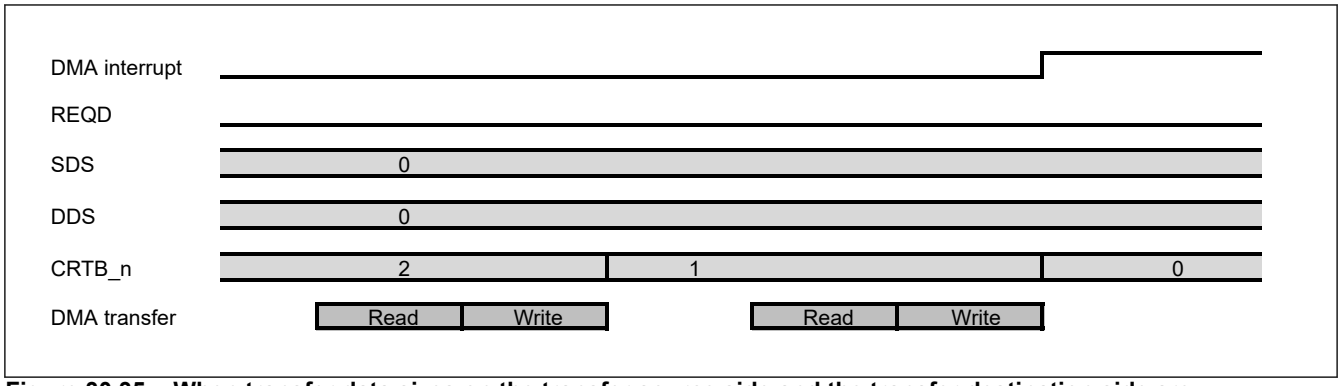


Figure 30.35 When transfer data sizes on the transfer source side and the transfer destination side are equal (LVL = 0, HIEN = 1, REQD = 0, and SDS = DDS in CHCFG_n)

30.15.8 Transfer Status

The CHSTAT_n register indicates the transfer status of each channel.

30.15.8.1 Transfer Status

The TACT bit in the CHSTAT_n register indicates that channel n is operating. Writing 1 to the SETEN bit in the CHCTRL_n register sets 1. The TACT bit remains 1 during an access to a descriptor or waiting for a DMA request.

The TACT bit is cleared when the EN bit in the CHSTAT_n register is cleared (for details on clearing conditions, see section 30.2.17.1. CHSTAT_n : Channel Status Register n (n = 0, 1), and the ongoing DMA transfer is completed.

The TACT bit is not cleared when the EN bit is not cleared (when the REN bit in the CHCFG_n register = 1 in register mode or the next descriptor access is performed in link mode) even if a DMA transaction finishes.

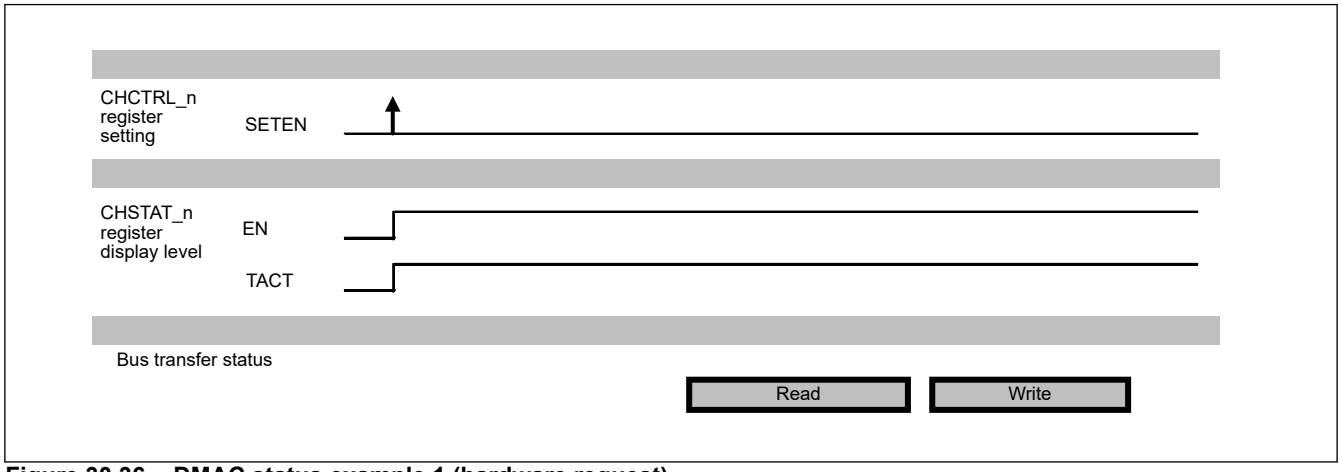


Figure 30.36 DMAC status example 1 (hardware request)

30.15.8.2 Suspending a Transfer

A DMA transfer can be suspended using the SETSUS bit in the CHCTRL_n register. At that time, if there is an active bus cycle, the DMA transfer is suspended after the bus cycle finishes. Writing 1 to the CLRSUS bit in the CHCTRL_n register resumes from the suspended state.

To check whether the transfer is suspended, set the SETSUS bit in the CHCTRL_n register, and then make sure that the SUS bit in the CHSTAT_n register or the SUS bit in the DSTAT_SUS register of the applicable channel is set to 1.

30.15.8.3 Aborting a Transfer

If 1 is written to the CLREN bit in the CHCTRL_n register during a DMA transaction, the DMA transaction of the channel can be aborted. As processing after aborting the transaction, you can use the SBE bit in the CHCFG_n register to select a mode to flush data remaining in a buffer when the transaction is aborted or a mode that does not flush buffer data. SBE = 0 (mode not to flush data) is selected by default.

When this mode (flush data) is enabled, if an ongoing transfer is aborted when the CLREN bit in the CHCTRL_n register = 1, data remaining in the buffer of the DMAC is flushed and the operation stops.

(1) Aborting a Transfer (without Buffer Flush: SBE = 0)

If 1 is written to the CLREN bit in the CHCTRL_n register during a DMA transfer, the DMA transfer is aborted and then stopped. The timing to stop the transfer depends on the set REQD bit value. After the transfer stops, write 1 to the SWRST bit in the CHCTRL_n register to clear the contents of the DMAC, and then make settings for the next transfer.

- When the TACT bit in the CHSTAT_n register is cleared to 0, you can confirm that the channel has completely stopped.
- If a DMA transfer is aborted, USB_FDMAn is not asserted.
- When the REQD bit in the CHCFG_n register = 0, the transfer stops when the next read operation is completed. (However, if data that can be written is remaining in the buffer, the data is written and then the transfer stops.)
- When the REQD bit in the CHCFG_n register = 1, the transfer stops when the next write operation is completed.

(2) Aborting a Transfer (with Buffer Flush: SBE = 1)

If 1 is written to the CLREN bit in the CHCTRL_n register during a DMA transfer, the DMA transfer is aborted. When the REQD bit in the CHCFG_n register = 0, already read data is flushed (written), and then the DMA transfer stops. When REQD = 1, flush mode cannot be used by hardware.

After the transfer stops, set the SWRST bit in the CHCTRL_n register to clear the contents of the DMAC, and then make settings for the next transfer.

- When the TACT bit in the CHSTAT_n register is cleared to 0, you can confirm that the channel has completely stopped.

(3) Checking Channel Stop

Even if 1 is written to the CLREN bit in the CHCTRL_n register, and the EN bit in the CHSTAT_n register is cleared to 0, if a transfer is already in progress on the bus, the DMAC cannot stop the transfer immediately. To check that DMA has completely stopped, make sure that the EN bit is 0 and the TACT bit in the CHSTAT_n register is 0.

(4) Transfer Aborting Procedure

The following describes the procedure for aborting a transfer:

1. Write 1 to the CLREN bit in the CHCTRL_n register.
2. When the SBE bit in the CHCFG_n register = 0, the transfer stops according to the REQD bit value in the CHCFG_n register. If SBE = 1 at this time, flush mode is enabled.
3. Read the CHSTAT_n register and confirm that the TACT bit is 0. When TACT = 0, DMA has completely stopped. When TACT = 1, continue polling until the TACT bit is cleared to 0.
4. If you want to perform the next DMA transfer after a transfer is aborted, be sure to set the SWRST (software reset) bit in the CHCTRL_n register to 1 before the next transfer starts.

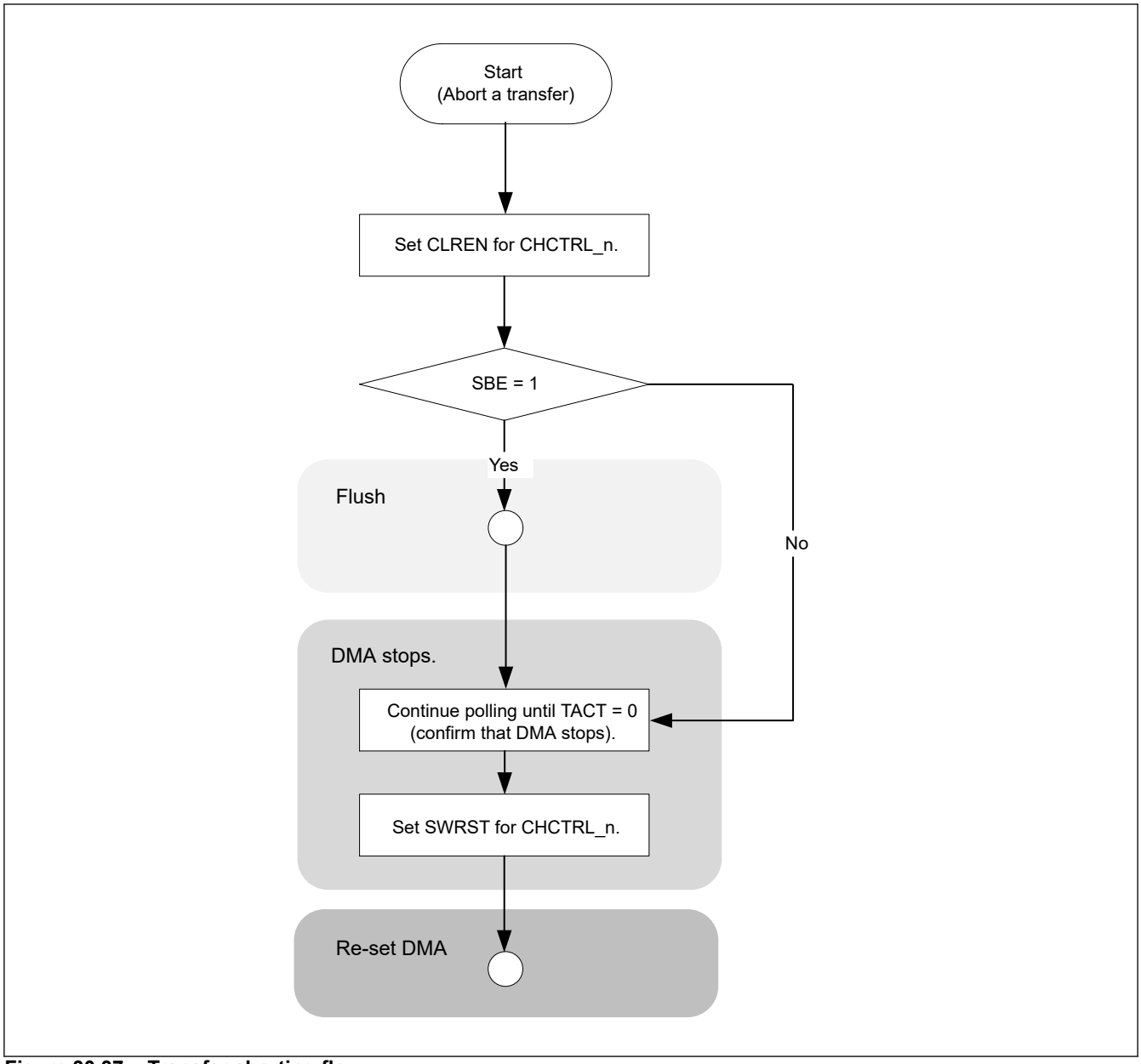


Figure 30.37 Transfer aborting flow

30.16 Access Type

30.16.1 AHB Master DMA Transfer Combinations

30.16.1.1 Read Access

The DMA read access issuance type has the following features.

- Read access is made to a beat-aligned space (including the source address indicated in the CRSA_n register) with the size specified by the SDS[2:0] bits in the CHCFG_n register. Beat-unaligned transfer is not performed for the bus. An unnecessary area may be read depending on the set values of the CRSA_n register and each SKIP register. In this case, necessary data in the read data is loaded to a buffer.
- The size and the burst length are determined by the value set for the SDS[2:0] field.
  When the value set for the SDS[2:0] field is equal to or smaller than the bus width:

  Size: SDS[2:0] set value

Burst type: Single

When the value set for the SDS[2:0] field is larger than the bus width:

Size: Bus width

Burst type: Fixed-length burst (burst length = SDS[2:0] set value/bus width)

The following table shows AHB bus access types.

Table 30.51 DMA read transfer combinations

SDS	Source address	AHB transfer					
		First transfer			Second transfer		
		Address	Size	Burst	Address	Size	Burst
0 (8 bits)	—	addr	8	Single	—	—	—
1 (16 bits)	2-byte align	{addr[31:1], 0b}	16	Single	—	—	—
	2-byte unalign				{addr[31:1], 0b} + 0x2	16	Single
2 (32 bits)	4-byte align	{addr[31:2], 00b}	32	Single	—	—	—
	4-byte unalign				{addr[31:2], 00b} + 0x4	32	Single
4 (128 bits)	16-byte align	{addr[31:4], 0x0}	32	INCR4	—	—	—
	16 bytes				{addr[31:4], 0x0} + 0x10	32	INCR4
5 (256 bits)	32-byte align	{addr[31:5], 0x00}	32	INCR8	—	—	—
	32 bytes				{addr[31:5], 0x00} + 0x20	32	INCR8
6 (512 bits)	64-byte align	{addr[31:6], 0x00}	32	INCR16	—	—	—
	64 bytes				{addr[31:6], 0x00} + 0x40	32	INCR16

Note: If the transfer becomes EBT during the burst transfer, remaining transfer is performed by 32-bit INCR.

30.16.1.2 Write Access

The DMA write access issuance type has the following features.

- Write access is made from the destination address (indicated in the CRDA_n register) to a beat-aligned boundary with the size specified by the DDS[2:0] bits in the CHCFG_n register.
- The size and the burst length are determined by the value set for the DDS[2:0] field.
  When the value set for the DDS[2:0] field is equal to or smaller than the bus width:

  Size: DDS[2:0] set value

  Burst type: Single

  When the value set for the DDS[2:0] field is larger than the bus width:

  Size: Bus width

  Burst type: Fixed-length burst (burst length = DDS[2:0] set value/bus width)
- Write access to any space other than the specified space is not performed. In the following cases, write access is made with a combination of sizes smaller than the value set for the DDS[2:0] field.
  - When the destination address is beat-unaligned with the value set for the DDS[2:0] field
  - When an access with the DDS field size is made over the SKIP boundary
  - When the DDS[2:0] size is larger than the remaining number of transfer bytes

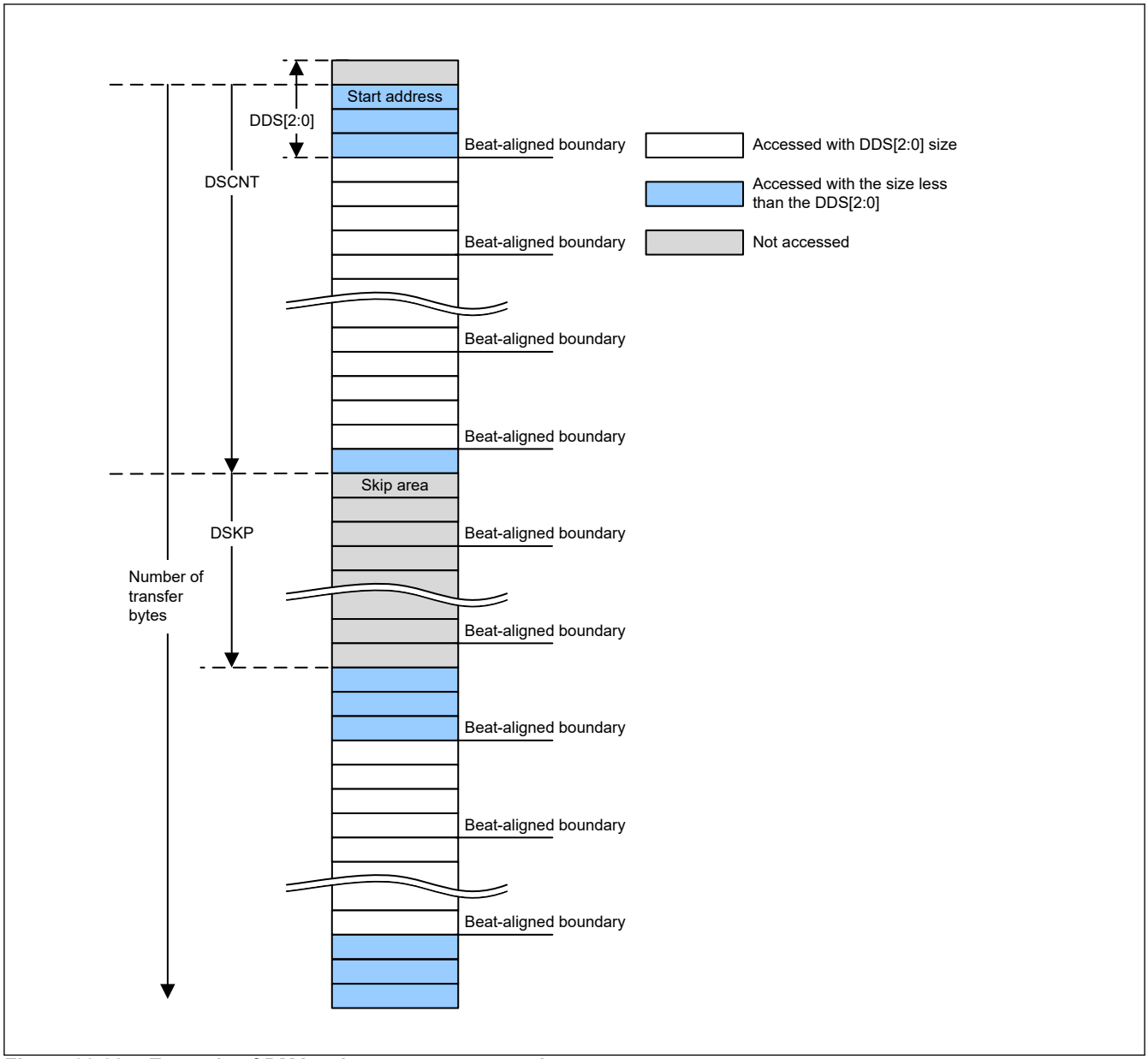


Figure 30.38 Example of DMA write access space and access type

31. Serial Communications Interface (SCI)

31.1 Overview

This LSI has six independent serial communications interface (SCI) channels. The SCI is configurable to five asynchronous and synchronous serial interfaces:

- Asynchronous interfaces (UART and Asynchronous Communications Interface Adapter (ACIA))
- 8-bit clock synchronous interface
- Simple I2C (master-only)
- Simple SPI
- Smart card interface

The smart card interface complies with the ISO/IEC 7816-3 standard for electronic signals and transmission protocol. Each SCI has FIFO buffers to enable continuous and full-duplex communication, and the data transfer speed can be configured independently using a baud rate generator.

Table 31.1 lists the SCI specifications, Figure 31.1 shows a block diagram of SCI channel n, and Table 31.2 lists the I/O pins.

Table 31.1 SCI specifications (1 of 3)

Item	Specifications
Number of channels	6 channels (ch0 to ch4 in NONSAFETY and ch5 in SAFETY)
Serial communication modes	• Asynchronous • Clock synchronous • Smart card interface • Simple I2C • Simple SPI
Transfer speed	Bit rate specifiable with the baud rate generator
Full-duplex communications	• Transmitter: Continuous transmission possible using double-buffering • Receiver: Continuous reception possible using double-buffering
Half-duplex communication	Possible by using TXD pins
Data transfer	Selectable as LSB-first or MSB-first transfer
RXD/TXD polarity	Selectable polarity for TXD and RXD independently
Interrupt sources	• Transmit end, transmit data empty, receive data full, receive error, receive data ready, and address match • Completion of generation of a start condition, restart condition, or stop condition (for simple I2C mode)
RS-485 driver control function	Output DE pin to enable transmission mode of external transceiver
Loopback function	Self-diagnosis of communication function inside module supported
Synchronizing circuit bypass function	Bus clock (PCLKM) can be used as operation clock (PCLKSCIn) when bypass function is enabled.
Module-stop function	Module-stop state can be set for each channel

Table 31.1 SCI specifications (2 of 3)

Item		Specifications
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Reception sampling timing adjustment	Adjustable reception sampling timing from default timing
	Transmission timing adjustment	Adjustable timing of transmission wave edge by register
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	Transmission and reception controllable with CTSn# and RTSn# pins
	Transmission/reception	Selectable to 1-stage register or 16-stage FIFO
	Address match	Interrupt request/event output can be issued upon detecting a match between received data and the value in the compare match register.
	Start-bit detection	Selectable to low level or falling edge detection
	Break detection	Breaks from framing errors detectable by reading register
	Clock source	Selectable to internal or external clock
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication enabled between multiple processors
	Noise cancellation	Digital noise filters included on signal paths from RXDn pin inputs
Clock synchronous mode	Data length	8 bits
	Reception sampling timing adjustment	Adjustable reception sampling timing to backward from default timing in case internal clock used
	Receive error detection	Overrun error
	Clock source	Selectable to internal clock (master mode) or external clock (slave mode)
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Hardware flow control	Transmission and reception controllable with CTSn# and RTSn# pins
	Transmission/ reception	Selectable to 1-stage register or 16-stage FIFO
Smart card interface mode	Error processing	Error signal can be automatically transmitted on detecting a parity error during reception.
		Data can be automatically retransmitted on receiving an error signal during transmission.
	Data type	Both direct and inverse convention supported
Simple I2C mode	Transfer format	I2C bus format (MSB-first only)
	Operating mode	Master (single-master operation only)
	Transfer rate	Up to 400 kbps
	Noise cancellation	The signal paths from input on the SCLn and SDAn pins incorporate digital noise filters and provide an adjustable interval for noise cancellation.

Table 31.1 SCI specifications (3 of 3)

Item		Specifications
Simple SPI mode	Data length	8 bits
	Error detection	Overrun error
	Clock source	Selectable to internal clock (master mode) or external clock (slave mode)
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Reception sampling timing adjustment	Adjustable reception sampling timing to backward from default timing in case internal clock used
	SS input pin function	High impedance state can be invoked on the output pins by driving the SSn# pin high.
	Clock settings	Configurable between four clock phase and clock polarity settings
	Transmission/ reception	Selectable to 1-stage register or 16-stage FIFO
Bit rate modulation function		Error reduction through correction of outputs from the baud rate generator

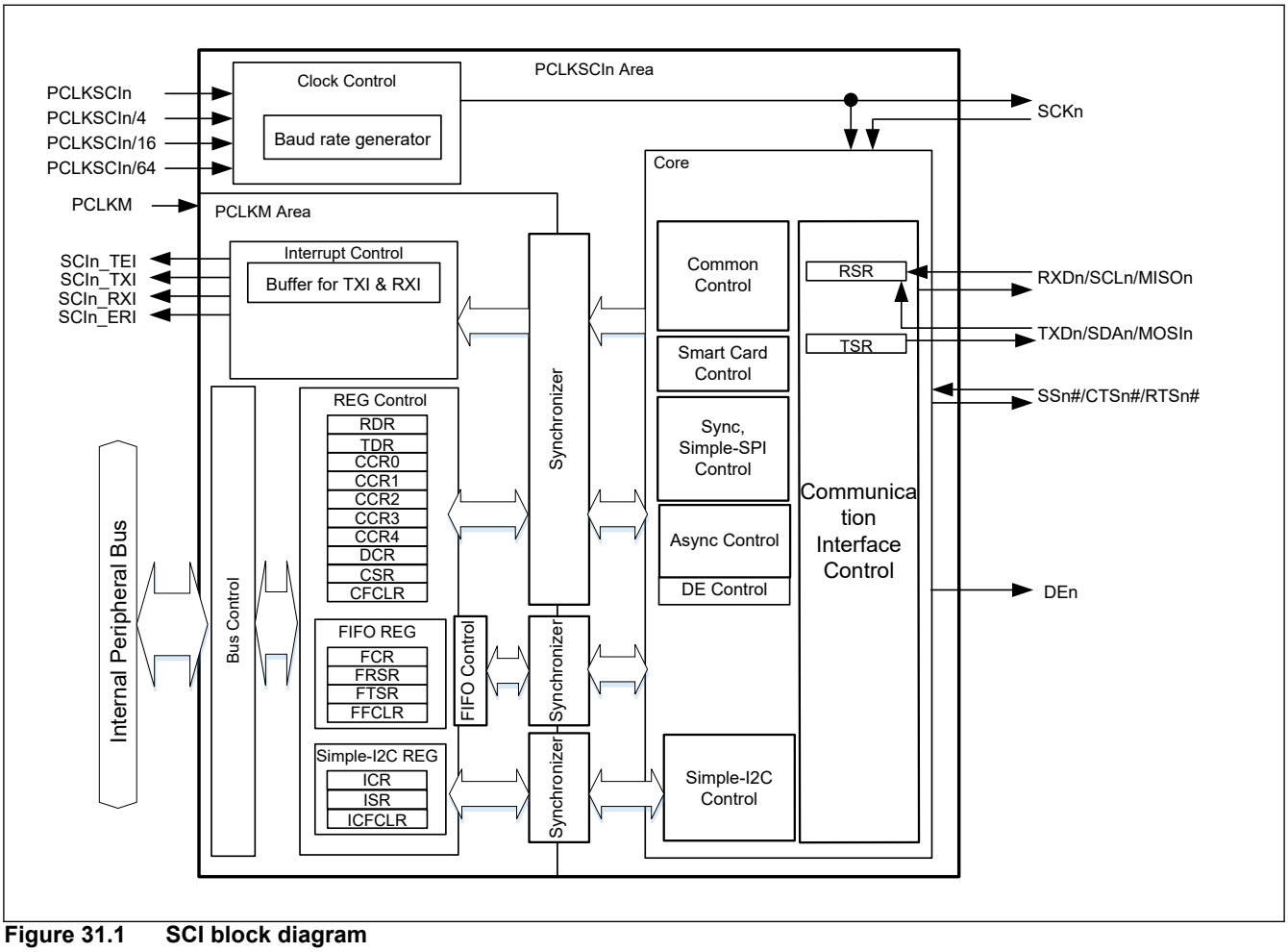


Figure 31.1 SCI block diagram

Table 31.2 SCI input/output pins

Channel	Pin name	I/O	Function
SCIn	SCKn	I/O	SCIn clock input/output
	RXDn	Input	SCIn receive data input
	SCLn	I/O	SCIn I2C clock input/output
	MISOn	I/O	SCIn data input (master), data output (slave)
	TXDn	Output	SCIn transmit data output
	SDAn	I/O	SCIn I2C data input/output
	MOSIn	I/O	SCIn data output (master), data input (slave)
	SSn#	Input	SCIn chip select input, active-low
	RTSn#	Output	SCIn transfer start control output, active-low
	CTSn#	Input	SCIn transfer start control input, active-low
	DEn	Output	SCIn Driver Enable output

Note: n = 0 to 5

31.2 Register Descriptions

This section explains each register specification.

Table 31.3 shows a list of registers/bits that can not be rewritten during communication. Also, check the notes on register rewriting described in section 31.16. Usage Notes.

Table 31.3 Bits that cannot be rewritten during communication

Register	Bit	Write constraint
CCR0	RE	In Clock synchronous mode, Simple-SPI and Simple-I2C, 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written in TE and RE.
CCR0	TE	
CCR0	IDSEL	Writable only when TE = 0 and RE = 0.
CCR0	SSE	Writable only when TE = 0 and RE = 0.
CCR1	CTSE	Writable only when TE = 0 and RE = 0.
CCR1	CTSPEN	Writable only when TE = 0 and RE = 0.
CCR1	PE	Writable only when TE = 0 and RE = 0.
CCR1	PM	Writable only when TE = 0 and RE = 0.
CCR1	TINV	Writable only when TE = 0 and RE = 0.
CCR1	RINV	Writable only when TE = 0 and RE = 0.
CCR1	SPLP	Writable only when TE = 0 and RE = 0.
CCR1	SHARPS	Writable only when TE = 0 and RE = 0.
CCR1	NFCS[2:0]	Writable only when TE = 0 and RE = 0.
CCR1	NFEN	Writable only when TE = 0 and RE = 0.
CCR2	All bits	Writable only when TE = 0 and RE = 0.
CCR3	All bits	Writable only when TE = 0 and RE = 0.
CCR4	All bits	Writable only when TE = 0 and RE = 0.
ICR	IICDL[4:0]	Writable only when TE = 0 and RE = 0.
ICR	IICINTM	Writable only when TE = 0 and RE = 0.
ICR	IICCSC	Writable only when TE = 0 and RE = 0.
FCR	All bits	Writable only when TE = 0 and RE = 0.
DCR	All bits	Writable only when TE = 0 and RE = 0.

31.2.1 RSR : Receive Shift Register

RSR is a shift register which is used to receive serial data input from the RXDn pin and converts it into parallel data. When one frame of data has been received, it is automatically transferred to the RDR register.

The RSR register cannot be directly accessed by the CPU.

31.2.2 RDR : Receive Data Register

Base address: SCIn = 0x8000_1000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_1000

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	FER	PER	—	—	ORER	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	FFER	FPER	DR	MPB	RDAT[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	RDAT[8:0]	Serial receive data RDAT is a 9-bit register for storing received data. Received data is stored in [6:0] when 7-bit data is selected, in [7:0] when 8-bit data is selected, and in [8:0] when 9-bit data is selected. And 0 is stored in the unused bit.	R
9	MPB	Multi-processor flag 0: Data transmission cycles 1: ID transmission cycles	R
10	DR	Receive data ready flag FRSR.DR can be read.	R
11	FPER	FIFO parity error flag (Valid only in Asynchronous mode) 0: There is no parity error in the data read from the receive FIFO. 1: There is parity error in the data read from the receive FIFO.	R
12	FFER	FIFO framing error flag (Valid only in Asynchronous mode) 0: There is no framing error in the data read from the receive FIFO. 1: There is framing error in the data read from the receive FIFO.	R
23:13	—	These bits are read as 0.	R
24	ORER	Overrun Error flag CSR.ORER can be read.	R
26:25	—	These bits are read as 0.	R
27	PER	Parity error flag CSR.PER can be read.	R
28	FER	Framing error flag CSR.FER can be read.	R
31:29	—	These bits are read as 0.	R

In FIFO mode (CCR3.FM = 1), this register has 16-stage FIFO structure.

RDAT[8:0] bits (Serial receive data)

After one frame of data is received, the received data is transferred from the RSR register to this registers, thus allowing the RSR register to receive the next data.

The RSR and RDR registers have a double-buffered structure to enable continuous reception.

Read RDR only once when a receive data full interrupt (RXI) request is issued. Without reading received data from RDR, if the next one frame is received, an overrun error occurs.

In FIFO mode, continuous reception is executed until 16 stages are stored. If data is read when there is no received data in the receive FIFO (RDR), the value is undefined. When the receive FIFO (RDR) are full of received data, subsequent serial receive data is lost.

The CPU cannot write to RDR.

"0" is stored in the bit position which is not received (RDR.bit8 or RDR.bit7) at the time of 7 bit or 8 bit communication of Asynchronous mode.

FFER and FPER store error information of received data only in FIFO mode. When not FIFO, 0 is stored in RDR.bit 12 and bit 11.

MPB bit (Multi-processor flag)

In asynchronous mode, during multi-processor communication (CCR3.MP = 1), the value of the multi-processor bit corresponding to the received data (RDAT[8:0]) can be read.

FPER bit (FIFO parity error flag)

Indicates whether the data read from the receive FIFO has a parity error.

FFER bit (FIFO framing error flag)

Indicates whether the data read from receive FIFO has a framing error.

31.2.3 TDR : Transmit Data Register

Base address: SCIn = 0x8000_1000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_1000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	MPBT	TDAT[8:0]								
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
8:0	TDAT[8:0]	Serial transmit data TDAT is a 9-bit register for setting transmit data. Transmit data is set in [6:0] when 7-bit data is selected, in [7:0] when 8-bit data is selected, and in [8:0] when 9-bit data is selected. When byte access, write TDR[15:8] and then write TDR[7:0].	R/W
9	MPBT	Multi-processor transfer bit flag Value of the multi-processor bit in the transmission frame. This bit is use in Asynchronous mode. When writing to this bit when not used, write the initial value. 0: Data transmission cycles 1: ID transmission cycles	R/W
31:10	—	These bits are read as 1. The write value should be 1.	R/W

In FIFO mode (CCR3.FM = 1), this register has 16-stage FIFO structure.

TDAT[8:0] bits (Serial transmit data)

The TDR is a 9-bit register for storing transmit data.

When empty space is detected in the TSR register, the transmit data stored in the TDR registers is transferred to TSR, and transmitting is started.

The TSR and TDR registers have a double-buffer structure to realize continuous transmission. When the next data to be transmitted is stored in TDR after one frame of data has been transmitted, the transmitting operation is continued by transfer to the TSR register.

In FIFO mode, when the SCI detects that the transmit shift register (TSR) is empty, it transmits data in the transmit FIFO (TDR) into TSR and starts serial transmission. Continuous serial transmission is executed until there is no transmit data left in the transmit FIFO (TDR).

When transmit FIFO is full of transmit data (16 frames), no more data can be written. If new data is written, the data is ignored.

Write transmit data to the TDR only once, after TXI interrupt is issued when CCR0.TE is 1. And when using bytes access, write TDR[15:8] and then write TDR[7:0].

MPBT bit (Multi-processor transfer bit flag)

Selects the multi-processor bit of transmit frame.

31.2.4 TSR : Transmit Shift Register

TSR is a shift register that transmits serial data. TSR cannot be directly accessed by the CPU.

To perform serial data transmission, the SCI first automatically transfers transmit data from TDR to TSR, and then sends the data to the TXDn pin.

31.2.5 CCR0 : Common Control Register 0

Base address: SCIn = 0x8000_1000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_1000

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	SSE	—	—	TEIE	TIE	—	—	—	RIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	IDSEL	DCME	MPIE	—	—	—	TE	—	—	—	RE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RE	Receive Enable In clock synchronous mode and simple-SPI mode, receive only setting is prohibited (TE = 0 and RE = 1 setting prohibited). 0: Serial reception is disabled. 1: Serial reception is enabled.	R/W*1
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	TE	Transmit Enable 0: Serial transmission is disabled. 1: Serial transmission is enabled.	R/W*1
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	MPIE	Multi-Processor Interrupt Enable (Valid in asynchronous mode when CCR3.MP = 1.) This bit should set 0 in smart card interface mode. When the data with the multiprocessor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and non-multiprocessor reception is resumed. If you want to continue receiving operation using the multiprocessor function, set this bit to 1 sufficiently earlier than receiving the STOP bit of the next received frame. (Consider the synchronization delay time.) 0: Non-Multi-Processor reception 1: Multi-Processor reception	R/W*2

Bit	Symbol	Function	R/W
9	DCME	Data Compare Match Enable (Valid only in asynchronous mode) 0: Address match function is disabled. 1: Address match function is enabled.	R/W*2
10	IDSEL	ID frame select (Valid only in asynchronous mode with multi-processor mode) 0: All data is to be compared. 1: The data with RDR.MPB = 1 is to be compared.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
16	RIE	Receive Interrupt Enable 0: RXI and ERI interrupt requests are disabled. 1: RXI and ERI interrupt requests are enabled.	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	TIE	Transmit Interrupt Enable 0: TXI interrupt request is disabled. 1: TXI interrupt request is enabled.	R/W
21	TEIE	Transmit End Interrupt Enable This bit should set 0 in smart card interface mode. 0: TEI interrupt request is disabled. 1: TEI interrupt request is enabled.	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
24	SSE	SSn# Pin Function Enable (Valid in Simple-SPI mode.) In slave mode (CCR3.CKE[1:0] = 1xb), set 1 to this bit. 0: SSn# pin function is disabled. 1: SSn# pin function is enabled.	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. In clock-synchronouse mode (CCR3.MOD[2:0] = 010b), simple-SPI mode (CCR3.MOD[2:0] = 011b), and simple-I^2C mode (CCR3.MOD[2:0] = 100b), 1 can be written only when TE = 0 and RE = 0. After setting TE or RE to 1, only 0 can be written in TE and RE. In other modes, writing is enabled under any condition.

Note 2. This bit is cleared by hardware. Note that unintentional setting of this bit may be occurred by read-modify-write operation.

RE bit (Receive Enable)

Enables or disables serial receive operation.

When this bit is set to 1, serial reception becomes possible after the synchronization delay time has elapsed in asynchronous mode or the synchronous clock input in clock synchronous mode or start bit in smart-card-interface mode.

Note that CCR3 should be set prior to setting the RE bit to 1 in order to designate the reception format.

Except smart-card-interface-mode, even if reception is halted by setting the RE bit to 0, the CSR.RDRF, FER, PER, ORER, FRSR.DR flags are not affected and the previous values are retained. In smart-card-interface-mode, even if reception is halted by setting the RE bit to 0, the CSR.FER, PER, ORER flags are not affected and the previous value is retained. Also, to stop the reception operation, synchronization delay time will be required from when the RE bit is set to 0 until the reception operation is stopped.

TE bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission becomes possible after the synchronization delay time has elapsed. After the synchronization delay time, transmission is started by writing transmit data to TDR. Note that CCR3 should be set prior to setting the TE bit to 1 in order to designate the transmission format. In addition, the synchronization delay time is required until the transmission control circuit is stopped after the TE bit is set to 0.

MPIE bit (Multi-Processor Interrupt Enable)

When this bit is set to 1 and the data with the multi-processor bit set to 0 is received, the data is not read and setting the status flags (CSR.RDRF, ORER, FER, FRSR.DR) are not set.

When the data with the multi-processor bit set to 1 is received, the MPIE is automatically cleared to 0, and normal reception is resumed. For details, see section 31.4. Multi-Processor Communication Function. If you want to continue receiving operation using the multiprocessor function, set this bit to 1 sufficiently earlier than receiving the STOP bit of the next received frame.

When the receive data includes the MPB bit set to 0, the receive data is not transferred from the RSR to the RDR, a receive error is not detected, and setting the flags ORER and FER to 1 is disabled.

When the receive data includes the MPB bit set to 1, the MPIE bit is automatically cleared to 0, the RXI and ERI interrupt requests are enabled (if CCR0.RIE is set to 1), and setting the flags ORER, FER to 1 is enabled.

MPIE should be set to 0 if multi-processor communications function is not to be used.

DCME bit (Data Compare Match Enable)

It can select whether the Address match function (data compare match function) uses or not.

When DCME is 1, if SCI detects the match to the comparison data (CCR4.CMPD) with receive data, DCME is cleared automatically, and after that, SCI operation mode will be receive mode without data compare match function.

See section 31.3.6. Address Match (receive data match detection) Function.

The write value should be 0 other than asynchronous mode.

IDSEL bit (ID frame select)

It can select whether it's compared in spite of the value of MPB bit or it's compared only the data of MPB bit = 1 (ID frame) when the Address match function is valid. Please set at the same time as DCME.

RIE bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.

RXI and ERI interrupt request is disabled by setting the RIE bit to 0.

An ERI interrupt request can be canceled by reading 1 from the CSR.ORER, FER, or PER and then setting the flag to 0, or setting the RIE bit to 0.

TIE bit (Transmit Interrupt Enable)

Enables or disables a TXI interrupt request.

A TXI interrupt request is disabled by setting the TIE bit to 0. At the beginning of transmission, set 1 to CCR0.TE and CCR0.TIE simultaneously. Then the TXI interrupt request is generated.

TEIE bit (Transmit End Interrupt Enable)

Enables or disables a TEI interrupt request.

A TEI interrupt request is disabled by setting the TEIE bit to 0.

In simple I2C mode, the TEI is allocated to the interrupt on completion of issuing a start, restart, or stop condition (STI). In this case, the TEIE bit can be used to enable or disable the STI.

SSE bit (SSn# Pin Function Enable)

Set this bit to 1 if the SSn# pin is to be used in control of transmission and reception (in simple SPI mode). Set this bit to 0 in any other mode. Do not set both the SSE and CTSE (even if this setting is made, operation is the same as that when these bits are set to 0).

In the slave mode (CCR3.CKE[1:0] = 10b or 11b), SSE should be set 1.

In the master mode (CCR3.CKE[1:0] = 00b or 01b) and single-master, the SSn# pin on the master side is not required to control reception and transmission, so SSE should be set 0.

31.2.6 CCR1 : Common Control Register 1

Base address: SCIn = 0x8000_1000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_1000

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	NFEN	—	NFCS[2:0]			—	—	—	SHARPS	—	—	—	SPLP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	RINV	TINV	—	—	PM	PE	—	—	SPB2IO	SPB2DT	—	—	CTSPEN	CTSE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
0	CTSE	CTS Enable 0: CTS function is disabled (RTS output function is enabled). 1: CTS function is enabled.	R/W
1	CTSPEN	CTS external pin Enable 0: 1-pin mode, CTS and RTS share 1 pin. 1: 2-pin mode, CTS and RTS have dedicated pins.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	SPB2DT	Serial port break data select The output level of TXD pin is selected when CCR0.TE = 0.*1 0: Low level is output in TXD pin, when TINV = 0. High level is output in TXD pin, when TINV = 1. 1: High level is output in TXD pin, when TINV = 0. Low level is output in TXD pin, when TINV = 1.	R/W
5	SPB2IO	Serial port break I/O This bit selects output or not output of TXD pin when the CCR0.TE = 0.*1 0: The value of SPB2DT bit is not output in TXD pin. 1: The value of SPB2DT bit is output in TXD pin.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	PE	Parity Enable (Valid only in asynchronous mode. In Smart Card Interface mode, set 1 to this bit.) 0: Parity bit addition is not performed, when transmitting. Parity bit checking is not performed, when receiving. 1: The parity bit is added, when transmitting. The parity bit is checked, when receiving.	R/W
9	PM	Parity Mode (Valid only when PE = 1) 0: Selects even parity 1: Selects odd parity	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	TINV	TXD invert 0: Transmit data is not inverted and output to TXD.*2 1: Transmit data is inverted and output to TXD.	R/W
13	RINV	RXD invert 0: Received data from RXD is not inverted and input.*2 1: Received data from RXD is inverted and input.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	SPLP	Loopback Control This function is available in asynchronous mode and clock synchronous mode. 0: Normal mode 1: Loopback mode	R/W

Bit	Symbol	Function	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
20	SHARPS	Half-duplex communication select In the simple I2C mode or in the simple SPI mode, this bit should be set 0. 0: The TXD pin and the RXD pin independent 1: Half-duplex communication using TXD pin	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
26:24	NFCS[2:0]	Noise Filter Clock Select (Valid in asynchronous mode and simple-I2C mode.) Select for the noise filter's clock source. In simple-I2C mode, 000b setting is prohibited. "The baud rate generator source clock" means the clock selected by CCR2.CKS[1:0]. 0 0 0: The base clock signal divided by 1. 0 0 1: The baud rate generator source clock divided by 1. 0 1 0: The baud rate generator source clock divided by 2. 0 1 1: The baud rate generator source clock divided by 4. 1 0 0: The baud rate generator source clock divided by 8. Others: Setting prohibited	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
28	NFEN	Digital Noise Filter Function Enable (Valid in asynchronous mode and simple-I2C mode) Noise filter function is available for RXD only in asynchronous mode, for SCL and SDA inputs in simple-I2C mode. 0: Noise filter disabled 1: Noise filter enabled	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. You should use this bit in asynchronous mode only. Not guaranteed in other mode.
Note 2. RINV/TINV should be set 0b in smart card interface mode and simple I2C mode.

CTSE bit (CTS Enable)

Set this bit to 1 if the SSn# pin is to be used for CTS input. The RTS signal is output when this bit is set to 0. Set this bit to 0 in smart card interface mode, simple SPI mode, and simple I2C mode. Do not set both the CTSE and SSE bits (even if this setting is made, operation is the same as that when these bits are set to 0).

CTSPEN bit (CTS external pin Enable)

When CTSE is 1, select the terminals usage method when using the CTS and RTS functions.

SPB2DT bit (Serial port break data select), SPB2IO bit (Serial port break I/O)

The TXD pin status decided by combination of CCR0.TE bit, CCR1.SPB2IO bit and CCR1.SPB2DT bit is indicated in Table 31.4.

Table 31.4 TXD pin status

CCR0.TE	CCR1.SPB2IO	CCR1.SPB2DT	TXD pin status (When CCR1.TINV = 0)
0	0	x	Hi-Z (initial value)
0	1	0	Low level output
0	1	1	High level output
1	x	x	Serial transmission data is output.

PE bit (Parity Enable)

When PE bit to 1, the parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

In the multiprocessor format, the parity bit is not added or checked regardless of this bit setting.

PM bit (Parity Mode)

Selects the parity mode for transmission and reception (even or odd). In multi-processor mode, this bit is invalid.

For details on the usage of this bit in smart card interface mode, see section 31.5.2. Data Format (Except in Block Transfer Mode).

TINV bit (TXD invert), RINV bit (RXD invert)

The data of RDR is controlled by RINV and CCR3.SINV. And the data from TXD pin is controlled by TINV and CCR3.SINV. The control by RINV/TINV are done to communication pins (RXD/TXD), so they can control not only data-bits but also other bits (start bit, stop bit, parity bit). For details, see Figure 31.2.

During half-duplex communication and slave operation in simple SPI mode, use the TXD pin for reception, so set the inversion control of the received data with the TINV bit.

Note: Description and timing charts in this chapter assumes CCR1.TINV = 0 and CCR1.RINV = 0 when TINV and RINV value are not specified.

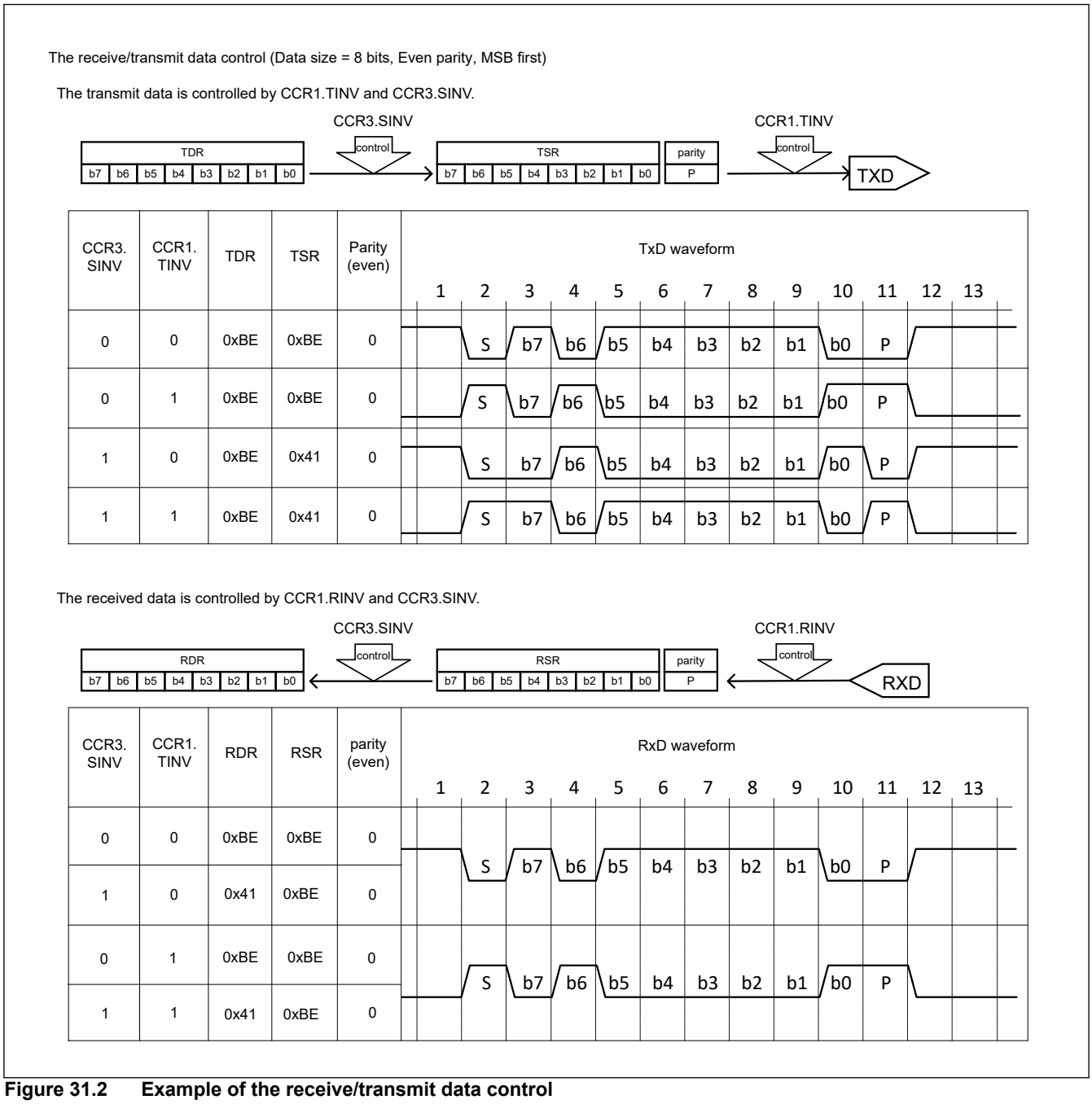


Figure 31.2 Example of the receive/transmit data control

SPLP bit (Loopback Control)

When SPLP = 1, TXD is connected to RXD internally.

Transmit data can be inverted and received by TINV = 1.

Set to 0 in asynchronous mode with external clock and clock synchronous mode slave.

SHARPS bit (Half-duplex communication select)

Setting this bit to 1 enables half-duplex communication using the TXD pin. However, it cannot be used in Simple-SPI mode, Simple-I2C mode, and Smart Card Interface mode.

NFCS[2:0] bits (Noise Filter Clock Select)

These bits select the sampling clock for the digital noise filter.

To use the noise filter in asynchronous mode, set these bits from 000b to 100b. In simple I2C mode, set the bits to a value in the range from 001b to 100b.

NFEN bit (Digital Noise Filter Function Enable)

This bit enables or disables the digital noise filter function. When the function is enabled, noise cancellation is applied to the RXDn input in asynchronous mode, and noise cancellation is applied to the SDAn and SCLn input in simple I2C mode. In any mode other than above, set the NFEN bit to 0.

31.2.7 CCR2 : Common Control Register 2

Base address: SCIn = 0x8000_1000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_1000

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	MDDR[7:0]								—	—	CKS[1:0]		—	—	—	BRME
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BRR[7:0]								—	ABCSE	ABCS	BGDM	—	BCP[2:0]		
Value after reset:	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
2:0	BCP[2:0]	Base Clock Pulse Selects the number of base clock cycles in smart card interface mode. 0 0 0: 93 clock cycles (S = 93)*1 0 0 1: 128 clock cycles (S = 128)*1 0 1 0: 186 clock cycles (S = 186)*1 0 1 1: 512 clock cycles (S = 512)*1 1 0 0: 32 clock cycles (S = 32)*1 (Initial value) 1 0 1: 64 clock cycles (S = 64)*1 1 1 0: 372 clock cycles (S = 372)*1 1 1 1: 256 clock cycles (S = 256)*1	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	BGDM	Baud Rate Generator Double-Speed Mode Select Valid in asynchronous/clock-synchronous/simple-SPI mode and CCR3.CKE[1] = 0. 0: Baud rate generator outputs the clock with single frequency. 1: Baud rate generator outputs the clock with doubled frequency.	R/W
5	ABCS	Asynchronous Mode Base Clock Select (Valid only in Asynchronous mode) 0: Selects 16 base clock cycles for 1-bit period. 1: Selects 8 base clock cycles for 1-bit period.	R/W
6	ABCSE	Asynchronous Mode Extended Base Clock Select (Valid only in asynchronous mode and CCR3.CKE[1] = 0) 0: Clock cycles for 1-bit period are decided with combination between CCR2.BGDM and CCR2.ABCS. 1: 6 base clock cycles for 1-bit period and doubled frequency is output from the baud rate generator.	R/W

Bit	Symbol	Function	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
15:8	BRR[7:0]	Bit rate setting BRR is an 8-bit register that adjusts the bit rate.	R/W
16	BRME	Bit Rate Modulation Enable 0: Bit rate modulation function is disabled. 1: Bit rate modulation function is enabled.	R/W
19:17	—	These bits are read as 0. The write value should be 0.	R/W
21:20	CKS[1:0]	Clock Select 0 0: PCLKSCIn clock (n = 0)*2 0 1: PCLKSCIn/4 clock (n = 1)*2 1 0: PCLKSCIn/16 clock (n = 2)*2 1 1: PCLKSCIn/64 clock (n = 3)*2	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
31:24	MDDR[7:0]	Modulation Duty setting MDDR corrects the bit rate adjusted by the BRR register.	R/W

Note 1. S is the value of S in Table 31.6.
Note 2. n is the decimal notation of the value of n in Table 31.6.

BCP[2:0] bits (Base Clock Pulse)

The BCP[2:0] bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode.

For details, see section 31.5.4. Receive Data Sampling Timing and Reception Margin.

BGDM bit (Baud Rate Generator Double-Speed Mode Select)

The BGDM bit is valid when the baud rate generator is selected as the clock source (CCR3.CKE[1] = 0) in asynchronous mode, clock synchronous mode, simple-SPI mode. When external clock is selected (CCR3.CKE[1] = 1), set it to 0. For the clock output from the baud rate generator, either single or doubled frequency can be selected. The base clock is generated by the clock output from the baud rate generator. When the BGDM bit is set to 1, the base clock cycle is halved and the bit rate is doubled.

Set this bit to 0 in modes other than asynchronous mode or clock synchronous mode or simple-SPI.

ABCS bit (Asynchronous Mode Base Clock Select)

Selects the clock cycles for 1-bit period.

Set it to 0 in modes other than asynchronous mode.

ABCSE bit (Asynchronous Mode Extended Base Clock Select)

The pulse number for a base clock at 1-bit period is 6 and the clock of a double frequency is output from baud rate generator. Only when the bit rate is set to 6 dividing frequency of the bus clock, please use this bit and set CCR2.CKS[1:0] = 00b and BRR = 0.

Set it to 0 in modes other than asynchronous mode. Even in asynchronous mode, set it to 0 when using external clock.

Table 31.5 Base clock cycle number per 1-bit

ABCSE bit	ABCS bit	BGDM bit	The base clock cycles / 1 bit	The frequency of the baud rate generator
0	0	0	16	×1
0	0	1	16	×2
0	1	0	8	×1
0	1	1	8	×2
1	x	x	6	×2

BRR[7:0] bits (Bit rate setting)

BRR is an 8-bit register that adjusts the bit rate.

SCI has independent baud rate generator control, different bit rates can be set for each. Table 31.6 shows the relationship between the setting (N) in the BRR and the bit rate (B) for asynchronous mode, multiprocessor transfer, clock synchronous mode, smart card interface mode, simple SPI mode, and simple I2C mode.

Table 31.6 Relationship between N setting in BRR and bit rate B

Mode	BGDM bit	ABCS bit	ABCSE bit	BRR[7:0] setting	Error (%)
Asynchronous multiprocessor transfer	0	0	0	$N = \frac{PCLKSCIn \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error = \{\frac{PCLKSCIn \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$
	1	0	0	$N = \frac{PCLKSCIn \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error = \{\frac{PCLKSCIn \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{PCLKSCIn \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$Error = \{\frac{PCLKSCIn \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$
	x	x	1	$N = \frac{PCLKSCIn \times 10^6}{12 \times 2^{2n-1} \times B} - 1$	$Error = \{\frac{PCLKSCIn \times 10^6}{B \times 12 \times 2^{2n-1} \times (N+1)} - 1\} \times 100$
Clock synchronous, Simple-SPI	0	0	0	$N = \frac{PCLKSCIn \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	—
	1	0	0	$N = \frac{PCLKSCIn \times 10^6}{4 \times 2^{2n-1} \times B} - 1$	—
Smart card interface				$N = \frac{PCLKSCIn \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error = \{\frac{PCLKSCIn \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1\} \times 100$
Simple-I2C*1				$N = \frac{PCLKSCIn \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	—

Note: B: Bit rate (bps)
N: BRR setting for baud rate generator (0 ≤ N ≤ 255)
PCLKSCIn: Operating frequency (MHz)
n and S: Determined by the settings of the CCR2 registers as listed in the table below. Please be careful about 2^{2n+1} is used in the expression for Smart card interface, 2^{2n-1} is used in other modes.

Note 1. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I2C mode satisfy the I2C standard.

Table 31.7 Calculating widths at high and low level for SCL

Mode	SCL	Formula (result in seconds)
I^2C	Width at high level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{PCLKSCIn \times 10^6}$
	Width at low level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{PCLKSCIn \times 10^6}$

Table 31.8 Clock source settings

CCR2 setting	Clock source	n
CKS[1:0] = 00b	PCLKSCIn clock	0
CKS[1:0] = 01b	PCLKSCIn/4 clock	1
CKS[1:0] = 10b	PCLKSCIn/16 clock	2
CKS[1:0] = 11b	PCLKSCIn/64 clock	3

Table 31.9 Base clock settings in smart card interface mode (1 of 2)

CCR2 setting	Base clock cycles for 1-bit period	S
BCP[2:0] = 000b	93 clock cycles	93
BCP[2:0] = 001b	128 clock cycles	128
BCP[2:0] = 010b	186 clock cycles	186

Table 31.9 Base clock settings in smart card interface mode (2 of 2)

CCR2 setting	Base clock cycles for 1-bit period	S
BCP[2:0] = 011b	512 clock cycles	512
BCP[2:0] = 100b	32 clock cycles	32
BCP[2:0] = 101b	64 clock cycles	64
BCP[2:0] = 110b	372 clock cycles	372
BCP[2:0] = 111b	256 clock cycles	256

Table 31.10 lists examples of N settings in BRR in asynchronous mode. Table 31.11 and Table 31.16 list the maximum bit rate settable for each operating frequency. Examples of BRR (N) settings in clock synchronous mode and simple SPI mode are listed in Table 31.13. Examples of BRR (N) settings in smart card interface mode are listed in Table 31.15. Examples of BRR (N) settings in simple I2C mode are listed in Table 31.17. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, see section 31.5.4. Receive Data Sampling Timing and Reception Margin. Table 31.12 and Table 31.14 list the maximum bit rates with external clock input. Table 31.18 lists the minimum widths at high and low level for SCL at various bit rates.

When either the asynchronous mode base clock select bit (ABCS) or the baud rate generator double-speed mode select bit (BGDM) is set to 1 in asynchronous mode, the bit rate becomes twice that listed in Table 31.10.

When both of those registers are set to 1, the bit rate becomes four times the listed value.

Table 31.10 Examples of BRR settings for various bit rates (asynchronous mode)

	PCLKSCIn = 75 MHz			PCLKSCIn = 96 MHz			PCLKSCIn = 100 MHz		
Bit rate (bps)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	243	0.06	—	—	—	—	—	—
19200	0	121	0.06	0	155	0.16	0	162	-0.15
38400	0	60	0.06	0	77	0.16	0	80	0.47
57600	0	40	-0.76	0	51	0.16	0	53	0.47
115200	0	19	1.73	0	25	0.16	0	26	0.47

Note: This is an example when CCR2.ABCS = 0, CCR2.BGDM = 0 and CCR2.ABCSE = 0.
When either ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS = 1 and BGDM = 1, the bit rate increases four times.

Table 31.11 Maximum bit rate for each operating frequency (asynchronous mode) (1 of 2)

PCLKSCIn (MHz)	BGDM bit	ABCS bit	ABCSE bit	n	N	MAX baud rate (baud)
75	0	0	0	0	0	2,343,750
		1	0	0	0	4,687,500
	1	0	0	0	0	
		1	0	0	0	9,375,000
	x	x	1	0	0	12,500,000
96	0	0	0	0	0	3,000,000
		1	0	0	0	6,000,000
	1	0	0	0	0	
		1	0	0	0	12,000,000
	x	x	1	0	0	16,000,000

Table 31.11 Maximum bit rate for each operating frequency (asynchronous mode) (2 of 2)

PCLKSCIn (MHz)	BGDM bit	ABCS bit	ABCSE bit	n	N	MAX baud rate (baud)
100	0	0	0	0	0	3,125,000
		1	0	0	0	6,250,000
	1	0	0	0	0	
		1	0	0	0	12,500,000
	x	x	1	0	0	16,666,667

Table 31.12 Maximum bit rate with external clock input (asynchronous mode)

PCLKSCIn (MHz)	External clock (MHz)	Max baud rate (baud)	
		CCR2.ABCS = 0	CCR2.ABCS = 1
75	18.75	1,171,875	2,343,750
96	24	1,500,000	3,000,000
100	25	1,562,500	3,125,000

Table 31.13 BRR settings for various bit rates (clock synchronous mode, simple SPI mode)

Bit rate (bps)	PCLKSCIn = 75 MHz			PCLKSCIn = 96 MHz			PCLKSCIn = 100 MHz		
	BGDM bit	n	N	BGDM bit	n	N	BGDM bit	n	N
2.5 M	1	0	14	—	—	—	0	0	9
4 M	—	—	—	0	0	5	—	—	—
5 M	—	—	—	—	—	—	0	0	4
7.5 M	1	0	4	—	—	—	—	—	—
8 M	—	—	—	0	0	2	—	—	—
10 M	—	—	—	—	—	—	1	0	4
12 M	—	—	—	0	0	1	—	—	—
12.5 M	1	0	2	—	—	—	0	0	1
24 M	—	—	—	0	0	0	—	—	—
18.75 M	0	0	0	—	—	—	—	—	—
25 M	—	—	—	—	—	—	0	0	0
37.5 M	1	0	0	—	—	—	—	—	—

Note: '—' can be set, but an error over 10% will occur.

Table 31.14 Maximum bit rate with external clock input (clock synchronous mode, simple SPI mode)

PCLKSCIn (MHz)	External clock (MHz)	MAX bit rate (Mbps)
75	37.50	37.50
96	24.00	24.00
100	25.00	25.00

Note: 37.5 MHz is max AC spec in this LSI.

Table 31.15 BRR settings for various bit rates (smart card interface mode, n = 0, S = 372)

Bit rate (bps)	PCLKSCIn = 75 MHz			PCLKSCIn = 96 MHz			PCLKSCIn = 100 MHz		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
9600	0	10	-4.54	0	12	3.39	0	13	0.01

Table 31.16 Maximum bit rate for each operating frequency (smart card interface mode, S = 32)

PCLKSCIn (MHz)	MAX Bit rate (bps)	n	N
75	1,171,875	0	0
96	1,500,000	0	0
100	1,562,500	0	0

Table 31.17 BRR settings for various bit rates (simple I2C mode)

Bit rate (kbps)	PCLKSCIn = 75 MHz			PCLKSCIn = 96 MHz			PCLKSCIn = 100 MHz		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
100	0	22	1.90	0	29	0.00	0	30	0.81
400	0	5	-2.34	0	7	-6.25	0	7	-2.34

Table 31.18 Minimum widths at high and low level for SCL at various bit rates (simple I2C mode)

Bit rate (kbps)	PCLKSCIn = 75 MHz			PCLKSCIn = 96 MHz			PCLKSCIn = 100 MHz		
	n	N	Minimum width of high/low level (µs)	n	N	Minimum width of high/low level (µs)	n	N	Minimum width of high/low level (µs)
100	0	22	4.29 / 4.91	0	29	4.38 / 5.00	0	30	4.34 / 4.96
400	0	5	1.12 / 1.28	0	7	1.17 / 1.33	0	7	1.12 / 1.28

BRME bit (Bit Rate Modulation Enable)

Enables and disables the bit rate modulation function. The bit rate generated by baud rate generator is evenly corrected when this function is enabled.

Set to 0 in Clock-synchronous mode, Simple-SPI mode, and Smart Card Interface mode.

CKS[1:0] bits (Clock Select)

These bits select the clock source for the baud rate generator.

For the relationship between the settings of these bits and the baud rate, refer to BRR explanation.

MDDR[7:0] bits (Modulation Duty setting)

When the BRME bit is set to 1, the bit rate generated by the baud rate generator is evenly corrected according to the settings of MDDR (M/256). The relationship between the MDDR setting (M) and the bit rate (B) is given in Table 31.19.

The initial value of MDDR is 0xFF. Bit 7 in this register is fixed to 1.

Table 31.19 Relationship between MDDR setting (M) and bit rate (B) using bit rate modulation function

Mode*1	BGDM bit	ABCS bit	ABCSE bit	BRR setting	Error (%)
Asynchronous Multiprocessor transfer	0	0	0	$N = \frac{\mathrm{PCLKSCIn} \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\mathrm{Error} = \{\frac{\mathrm{PCLKSCIn} \times 10^6}{B \times 64 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\} \times 100$
	1	0	0	$N = \frac{\mathrm{PCLKSCIn} \times 10^6}{32 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\mathrm{Error} = \{\frac{\mathrm{PCLKSCIn} \times 10^6}{B \times 32 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\} \times 100$
	0	1	0		
	1	1	0	$N = \frac{\mathrm{PCLKSCIn} \times 10^6}{16 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\mathrm{Error} = \{\frac{\mathrm{PCLKSCIn} \times 10^6}{B \times 16 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\} \times 100$
	x	x	1	$N = \frac{\mathrm{PCLKSCIn} \times 10^6}{12 \times 2^{2n-1} \times (256/M) \times B} - 1$	$\mathrm{Error} = \{\frac{\mathrm{PCLKSCIn} \times 10^6}{B \times 12 \times 2^{2n-1} \times (256/M) \times (N+1)} - 1\} \times 100$
Simple-I2C*2				$N = \frac{\mathrm{PCLKSCIn} \times 10^6}{64 \times 2^{2n-1} \times (256/M) \times B} - 1$	

Note: B: Bit rate (bps)
M: MDDR setting (128 ≤ M ≤ 255)
N: BRR setting for bound rate generator (0 ≤ N ≤ 255)
PCLKSCIn: Operating frequency (MHz)
n and S: Determined by the settings of the CCR2.BCP[2:0] and CKS[1:0] as listed in Table 31.8 and Table 31.9.
Please be careful about 2^{2n+1} is used in the expression for Smart card interface, 2^{2n-1} is used in other modes.
Note 1. Do not use this function in Clock-synchronous mode, Simple-SPI mode, and Smart card Interface mode.
Note 2. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I2C mode satisfy the I2C standard.

Table 31.20 lists examples of N settings in BRR and M settings in MDDR in asynchronous mode.

Table 31.20 Examples of BRR and MDDR settings for various bit rates (asynchronous mode)

	PCLKSCIn = 75 MHz					PCLKSCIn = 96 MHz					PCLKSCIn = 100 MHz				
Bit rate (bps)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)	n	N	M	BGDM bit	Error (%)
38400	0	35	151	0	0.00	0	46	154	0	-0.00	0	47	151	0	0.00
57600	0	23	151	0	0.00	0	47	236	0	0.03	0	31	151	0	0.00
115200	0	11	151	0	0.00	0	23	236	0	0.03	0	15	151	0	0.00
230400	0	5	151	0	0.00	0	11	236	0	0.03	0	7	151	0	0.00
460800	0	2	151	0	0.00	0	5	236	0	0.03	0	3	151	0	0.00

Note: In this example when CCR2.ABCS = 0 and CCR2.ABCSE = 0.
CCR2.BRME = 0 (M = 256) disables the bit rate modulation function.

31.2.8 CCR3 : Common Control Register 3

Base address: SCIn = 0x8000_1000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_1000

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	BLK	GM	—	—	CKE[1:0]		—	—	DEN	FM	MP	MOD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RXDE SEL	STP	SINV	LSBF	—	—	CHR[1:0]		BPEN	—	—	—	—	—	CPOL	CPHA
Value after reset:	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	CPHA	Clock Phase Select (Valid in Clock-synchronous mode and Simple-SPI mode. Set this bit only when CCR0.TE = 0 and RE = 0.) 0: Data is sampled at an odd edge and changes at an even edge. (Clock is delayed.) 1: Data changes at an odd edge and is sampled at an even edge. (Clock is not delayed.)	R/W
1	CPOL	Clock Polarity Select (Valid in Clock-synchronous mode and Simple-SPI mode. Set this bit only when CCR0.TE = 0 and RE = 0.) 0: SCKn in idle state is 0. 1: SCKn in idle state is 1.	R/W
6:2	—	These bits are read as 0. The write value should be 0.	R/W
7	BPEN	Synchronizer bypass enable This bit controls whether to bypass the synchronizer circuit between the bus clock and operation clock. When BPEN = 1, PCLKSPIn (operation clock) is input from PCLKM (Bus clock) and the synchronization circuit is bypassed. See section 6, Clock Generation Circuit. 0: Synchronizer circuit is not bypassed. 1: Synchronizer circuit is bypassed.	R/W

Bit	Symbol	Function	R/W
9:8	CHR[1:0]	Character Length (Valid in Asynchronous mode)*1 Select the data length for transmission and reception. 0 0: Transmit/receive in 9-bit data length 0 1: Transmit/receive in 9-bit data length 1 0: Transmit/receive in 8-bit data length (initial value) 1 1: Transmit/receive in 7-bit data length*2	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	LSBF	LSB First select Set this bit to 0 in simple I2C mode. 0: MSB first 1: LSB first	R/W
13	SINV	Transmitted/Received Data Invert The level of communication pins (RXD/TXD) are controlled by combination of this bit and CCR1.TINV/RINV. For details, see Figure 31.2. Set this bit to 0 in simple-I2C mode. 0: TDR contents are transmitted to TSR as they are. RSR contents are stored to RDR as they are. 1: TDR contents are inverted before being transmitted to TSR. RSR contents are inverted and stored to RDR.	R/W
14	STP	Stop Bit Length (Valid in Asynchronous mode) 0: 1 stop bit 1: 2 stop bits	R/W
15	RXDESEL	Asynchronous Start Bit Edge Detection Select (Valid only in asynchronous mode) 0: The low level on the RXDn pin is detected as the start bit. 1: A falling edge on the RXDn pin is detected as the start bit.	R/W
18:16	MOD[2:0]	Communication mode select Select the SCI communication mode. 0 0 0: Asynchronous mode (Multi-processor mode) 0 0 1: Smart card interface mode 0 1 0: Clock synchronous mode 0 1 1: Simple SPI mode 1 0 0: Simple I2C mode Others: Setting prohibited	R/W
19	MP	Multi-Processor Mode (Valid in Asynchronous mode) 0: Multi-processor communications function is disabled. 1: Multi-processor communications function is enabled.	R/W
20	FM	FIFO Mode select (Valid in Asynchronous mode (including multi-processor mode), Clock synchronous mode, Simple SPI mode) 0: Non-FIFO mode (TDR and RDR registers are non-FIFO). 1: FIFO mode (TDR and RDR registers are FIFO).	R/W
21	DEN	Driver enable (Valid only in Asynchronous mode) 0: RS-485 Driver control function disable. 1: RS-485 Driver control function enable.	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
25:24	CKE[1:0]	Clock enable In the case of asynchronous mode When using the external clock 16 times the bit rate should be input from the SCKn pin when CCR2.ABCS bit is 0. Input a clock signal with a frequency 8 times the bit rate when the CCR2.ABCS bit is 1. 0 0: The baud rate generator The SCKn pin is available for use as an I/O port in accord with the I/O port settings. 0 1: The baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. 1 x: External clock In the case of Clock synchronous mode, Simple-SPI 0 x: Internal clock (Master operation) The SCKn pin functions as the clock output pin. 1 x: External clock (Slave operation) The SCKn pin functions as the clock input pin. In the case of Smart card interface mode When CCR3.GM = 0 0 0: Output disabled (The SCKn pin can be used for other peripherals.) 0 1: Clock output 1 x: Setting prohibited When CCR3.GM = 1 0 0: Output fixed low 1 0: Output fixed high x 1: Clock output	R/W
27:26	—	These bits are read as 0. The write value should be 0.	R/W
28	GM	GSM Mode (Valid only in Smart card interface mode) 0: Non-GSM mode operation 1: GSM mode operation	R/W
29	BLK	Block Transfer Mode (Valid only in Smart card interface mode) 0: Non-block transfer mode operation 1: Block transfer mode operation	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. In other than asynchronous mode, this bit setting is invalid and a fixed data length of 8 bits is used.
Note 2. LSB first is fixed and the MSB (bit 7) in TDR is not transmitted in transmission.

CPHA bit (Clock Phase Select)

This bit selects the phase of the clock signal output through the SCKn pin. For details, see Figure 31.85.

Set the bit to 1 in other than simple SPI mode and clock synchronous mode.

CPOL bit (Clock Polarity Select)

This bit selects the polarity of the clock signal output through the SCKn pin. For details, see Figure 31.85.

Set the bit to 1 in other than simple SPI mode and clock synchronous mode.

BPEN bit (Synchronizer bypass enable)

The synchronization circuit can be bypassed by this bit only when the same clock is input to the bus clock and the operation clock (PCLKSCIn). For details, see section 31.14. Synchronizer Bypass Function.

CHR[1:0] bits (Character Length)

Selects the data length for transmission and reception.

Except of asynchronous mode, a fixed data length of 8 bits is used.

LSBF bit (LSB First select)

Select whether to transmit/receive data in MSB first or LSB first.

SINV bit (Transmitted/Received Data Invert)

SINV can invert the transmit data-bit from TDR to TSR, and also can invert the received data from RSR to RDR. This bit does not affect the logic level of the parity bit. To invert the parity bit, invert the CCR1.PM.

STP bit (Stop Bit Length)

Selects the stop bit length in transmission.

In reception, only the first stop bit is checked regardless of this bit setting. If the second stop bit is 0, it is treated as the start bit of the next transmit frame.

RXDESEL bit (Asynchronous Start Bit Edge Detection Select)

Selects the detection method of the start bit for reception in asynchronous mode. When a break occurs, data receiving operation depends on the settings of this bit. Set this bit to 1 when reception should be stopped while a break occurs or when reception should be started without retaining the RXDn pin input at high level for the period of one data frame or longer after completion of the break.

Set this bit to 0 in modes except of asynchronous mode.

MOD[2:0] bits (Communication mode select)

Selects the SCI communication mode.

Table 31.21 Relationship between communication mode selection bits (MOD[2:0]), other operation mode setting bits

Communication mode	Asynchronous								SMIF*1	Clock synchronous		Simple SPI		Simple I2C
CCR3.MOD[2:0]	000b								001b	010b		011b		100b
CCR3.MP	0				1				—	—		—		—
CCR3.FM	0		1		0		1		—	0	1	0	1	—
CCR3.DEN	0	1	0	1	0	1	0	1	—	—		—		—
CCR0.SSE	—								—	—		0	1	—

Note: '— ' means setting prohibited.
Note 1. Smart card interface

MP bit (Multi-Processor Mode)

Disables/enables the multi-processor communications function. The settings of the PE bit and PM bit are invalid in multi-processor mode.

FM bit (FIFO Mode select)

When FM = 1, the TDR register / RDR register switches to FIFO mode.

DEN bit (Driver enable)

Select RS-485 Driver control function disable or enable.

CKE[1:0] bits (Clock enable)

These bits select the clock source and SCKn pin function.

In smart card interface mode, these bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, see section 31.5.8. Clock Output Control.

GM bit (GSM Mode)

Setting this bit to 1 allows GSM mode operation.

In GSM mode, the CSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, see section 31.5.6. Serial Data Transmission (Except in Block Transfer Mode) and section 31.5.8. Clock Output Control.

BLK bit (Block Transfer Mode)

Setting this bit to 1 allows block transfer mode operation.

For details, see section 31.5.3. Block Transfer Mode.

31.2.9 CCR4 : Common Control Register 4

Base address: SCIn = 0x8000_1000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_1000

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	AET	ATT[2:0]			AJD	AST[2:0]			—	—	—	—	—	—	ATEN	ASEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CMPD[8:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	CMPD[8:0]	Compare Match Data (Valid only in Asynchronous mode) Set the compare data pattern for address match function.	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	ASEN	Adjust receive sampling timing enable (Valid in Asynchronous mode using internal clock, Clock-synchronous mode operating as master, Simple-SPI mode operating as master) 0: Adjust sampling timing disable. 1: Adjust sampling timing enable.	R/W
17	ATEN	Adjust transmit timing enable (Valid only in asynchronous mode using internal clock) 0: Adjust transmit timing disable. 1: Adjust transmit timing enable.	R/W
23:18	—	These bits are read as 0. The write value should be 0.	R/W
26:24	AST[2:0]	Adjustment value for receive Sampling Timing This bit enables only when ASEN is 1. In asynchronous mode using internal clock The sampling timing of RXDn pin is adjusted from the middle of bit by the following formula. Adjustment sampling timing = base clock × the setting value of AST[2:0]. In Clock-synchronous mode and Simple-SPI mode using internal clock The RxD sampling timing can be adjusted by delaying 1 to 4 PCLKSCIn. 0 0 0: 1 PCLKSCIn delay 0 0 1: 2 PCLKSCIn delay 0 1 0: 3 PCLKSCIn delay 0 1 1: 4 PCLKSCIn delay Others: Setting prohibited	R/W
27	AJD	Adjustment Direction for receive sampling timing (Valid in Asynchronous mode using internal clock) This bit enables only when ASEN is 1. Adjustment direction for RxD receive sampling timing is determined by this bit. For details, see section 31.3.10. The function of adjust receive sampling timing (Asynchronous Mode). 0: The sampling timing is adjusted backward to the middle of bit. 1: The sampling timing is adjusted forward to the middle of bit.	R/W

Bit	Symbol	Function	R/W
30:28	ATT[2:0]	Adjustment value for Transmit timing (Valid in Asynchronous mode using internal clock) This bit enables only when ATEN is 1. The selected edge timing of TxD is adjusted by the following formula. Adjustment edge timing = base clock × the setting value of ATT[2:0] This setting timing is limited by setting the base clock cycles. For details, see section 31.3.11. The function of adjust transmit timing (Asynchronous Mode).	R/W
31	AET	Adjustment edge for transmit timing (Valid in Asynchronous mode using internal clock) The adjustable edge is set by this bit. This bit enables only when ATEN = 1. For details, see section 31.3.11. The function of adjust transmit timing (Asynchronous Mode). 0: Adjust the rising edge timing, when CCR1.TINV = 0. Adjust the falling edge timing, when CCR1.TINV = 1. 1: Adjust the falling edge timing, when CCR1.TINV = 0. Adjust the rising edge timing, when CCR1.TINV = 1.	R/W

Note: If the description in this document and the timing chart do not specify the ASEN / ATEN setting value, it means that the reception sampling adjustment function / transmission timing adjustment function are OFF (CCR4.ASEN = 0, CCR4.ATEN = 0).

CMPD[8:0] bits (Compare Match Data)

Set the comparison data for receive data, when Address match function is enabled (CCR0.DCME = 1). CCR4.CMPD[8:0] should be written while CCR0.DCME is 0.

For the comparison data, it can select length from 3 types, they are CMPD[6:0] with 7-bit length, CMPD[7:0] with 8-bit length, and CMPD[8:0] with 9-bit length.

ASEN bit (Adjust receive sampling timing enable)

When this bit is 1, the receive sampling timing adjustment function is enabled. Control is different in Asynchronous mode, Clock-synchronous mode, and Simple-SPI mode.

In Asynchronous mode using internal clock, see section 31.3.10. The function of adjust receive sampling timing (Asynchronous Mode) in details.

In Clock-synchronous mode mode as master, Simple-SPI mode operating as master, see section 31.7.7. Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used in details. Only the digital delay of the master mode receive sampling clock (MRCLK) can be controlled by this bit.

ATEN bit (Adjust transmit timing enable)

When this bit is 1, the transmission timing adjustment function is enabled. The transmission timing adjustment function can adjust the edge timing of the waveform output from the TXD pin. See section 31.3.11. The function of adjust transmit timing (Asynchronous Mode) in details.

AST[2:0] bits (Adjustment value for receive Sampling Timing)

When ASEN = 1, the receive sampling timing can be adjusted according to this bit setting value.

In Asynchronous mode using internal clock.

The sampling timing of RXD pin is adjusted from the middle of bit by the following formula. This setting value is limited by setting the base clock cycles. For details, see section 31.3.10. The function of adjust receive sampling timing (Asynchronous Mode).

Adjustment sampling timing = base clock × the setting value of AST[2:0].

In Clock-synchronous mode and Simple-SPI mode using internal clock

The RxD sampling timing can be adjusted by delaying 1 to 4 PCLKSCIn. For details, see section 31.7.7. Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used.

000: 1 PCLKSCIn delay

001: 2 PCLKSCIn delay

010: 3 PCLKSCIn delay

011: 4 PCLKSCIn delay

1xx: Setting prohibited

AJD bit (Adjustment Direction for receive sampling timing)

Set the RXD pin sampling timing adjustment direction from the bit center to the rear or front. See section 31.3.10. The function of adjust receive sampling timing (Asynchronous Mode) for details.

ATT[2:0] bits (Adjustment value for Transmit timing)

The edge timing of the TXD pin specified by the AET bit is adjusted by the base clock × ATT[2:0] setting value. The upper limit of the adjustment time that can be set is limited by the number of base clock cycles. See section 31.3.11. The function of adjust transmit timing (Asynchronous Mode) for details.

AET bit (Adjustment edge for transmit timing)

Set the TXD pin edge for timing adjustment. See section 31.3.11. The function of adjust transmit timing (Asynchronous Mode) for details.

31.2.10 ICR : Simple I2C Control Register

Base address: SCIn = 0x8000_1000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_1000

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	IICSCLS[1:0]		IICSDAS[1:0]		—	IICSTPREQ	IICRSTAREQ	IICSTAREQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	IICACKT	—	—	—	IICCSC	IICINTM	—	—	—	IICDL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	IICDL[4:0]	SDA Delay Output Select (Cycles below are of the clock signal from the baud rate generator.) 0x00: No output delay 0x01: 0 to 1 cycle 0x02: 1 to 2 cycles 0x03: 2 to 3 cycles 0x04: 3 to 4 cycles 0x05: 4 to 5 cycles ⋮ 0x1E: 29 to 30 cycles 0x1F: 30 to 31 cycles	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	IICINTM	I2C Interrupt Mode Select 0: Use ACK/NACK interrupts. 1: Use reception and transmission interrupts.	R/W
9	IICCSC	Clock Synchronization 0: No synchronization with the clock signal 1: Synchronization with the clock signal	R/W
12:10	—	These bits are read as 0. The write value should be 0.	R/W
13	IICACKT	ACK Transmission Data 0: ACK transmission 1: NACK transmission and reception of ACK/NACK	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	IICSTAREQ	Start Condition Generation 0: A start condition is not generated. 1: A start condition is generated. *1 *3 *4 *5	R/W

Bit	Symbol	Function	R/W
17	IICRSTAREQ	Restart Condition Generation 0: A restart condition is not generated. 1: A restart condition is generated. *2 *3 *4 *5	R/W
18	IICSTPREQ	Stop Condition Generation 0: A stop condition is not generated. 1: A stop condition is generated. *2 *3 *4 *5	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
21:20	IICSDAS[1:0]	SDA Output Select 0 0: Serial data output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SDAn pin. 1 1: Place the SDAn pin in the high-impedance state.	R/W
23:22	IICSCLS[1:0]	SCL Output Select 0 0: Serial clock output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SCLn pin. 1 1: Place the SCLn pin in the high-impedance state.	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. In the bus free state, perform the start condition generation.
Note 2. In the bus busy state, perform restart or stop condition generation when the SCLn pin after acknowledgment described in section 31.6. Simple I2C mode in Figure 31.55 and Figure 31.56 is low level.
Note 3. Do not set more than one from among the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits to 1 at a given time.
Note 4. Execute the generation of a condition after the value of the IICSTIF flag is 0.
Note 5. Do not write 0 to this bit while it is 1. Generation of a condition is suspended by writing 0 to this bit while it is 1.

IICDL[4:0] bits (SDA Delay Output Select)

These bits are used to set a delay for output on the SDAn pin relative to the falling edge of the output on the SCLn pin. The available delay settings range from no delay to 31 cycles, with the clock signal from the baud rate generator as the base. The signal obtained by frequency-dividing PCLKSCIn by the divisor set in CCR2.CKS[1:0] is supplied as the clock signal from the baud rate generator.

Set these bits to 0x00 unless operation is in simple I2C mode. In simple I2C mode, set these bits from 0x01 to 0x1F.

IICINTM bit (I2C Interrupt Mode Select)

This bit selects the sources of interrupt requests in simple I2C mode.

IICCSC bit (Clock Synchronization)

Set the IICCSC bit to 1 if the internally generated SCLn clock signal is to be synchronized when the SCLn pin has been placed at the low level in the case of a wait inserted by the other device, etc.

The SCLn clock signal is not synchronized if the IICCSC bit is 0. The SCLn clock signal is generated in accord with the rate selected in the BRR regardless of the level being input on the SCLn pin.

Set the IICCSC bit to 1 except during debugging.

IICACKT bit (ACK Transmission Data)

Transmitted data contains ACK bits. Set this bit to 1 when ACK and NACK bits are received.

IICSTAREQ bit (Start Condition Generation)

When a start condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTAREQ bit to 1.

If you want to generate the start condition after generating the stop condition, start the generation of the start condition with a half cycle period of the bit rate from the stop condition generation interrupt request output.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the start condition

IICRSTAREQ bit (Restart Condition Generation)

When a restart condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICRSTAREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of generation of the restart condition

IICSTPREQ bit (Stop Condition Generation)

When a stop condition is to be generated, set both the IICSDAS[1:0] and IICSCLS[1:0] bits to 01b as well as setting the IICSTPREQ bit to 1.

[Setting condition]

- Writing 1 to the bit

[Clearing condition]

- Completion of stop condition

IICSDAS[1:0] bits (SDA Output Select)

These bits control output from the SDAn pin.

IICSCLS[1:0] bits (SCL Output Select)

These bits control output from the SCLn pin.

31.2.11 FCR : FIFO Control Register

Base address: SCIn = 0x8000_1000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_1000

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	RSTRG[4:0]					RFRST	—	—	RTRG[4:0]				
Value after reset:	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TFRST	—	—	TTRG[4:0]					—	—	—	—	—	—	—	DRES
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DRES	Receive data ready error select (Valid in Asynchronous mode) This bit select the interrupt request for a reception data ready detection. 0: reception data full interrupt (RXI) 1: receive error interrupt (ERI)	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
12:8	TTRG[4:0]	Transmit FIFO data trigger number (Valid in Asynchronous mode (including multi-processor mode), Clock-synchronous mode, Simple-SPI mode) Trigger number must be set 15 or less. When DMAC transfer is used, set TTRG[4:0] = 0x0F. 0x00: Trigger number 0 ⋮ 0x0F: Trigger number 15 Others: Setting prohibited	R/W
14:13	—	These bits are read as 0. The write value should be 0.	R/W
15	TFRST	Transmit FIFO Data Register Reset This bit enables only when CCR3.FM is 1. The read value is always 0. 0: It is invalid. It does not affect the operation. 1: The number of data stored in transmit FIFO (TDR register) are made 0.	W
20:16	RTRG[4:0]	Receive FIFO data trigger number (Valid in Asynchronous mode (including multi-processor mode), Clock-synchronous mode, Simple-SPI mode) Trigger number must be set 15 or less. When DMAC transfer is used, set RTRG[4:0] = 0x00. 0x00: Trigger number 0 ⋮ 0x0F: Trigger number 15 Others: Setting prohibited	R/W
22:21	—	These bits are read as 0. The write value should be 0.	R/W
23	RFRST	Receive FIFO Data Register Reset This bit enables only when CCR3.FM is 1. The read value is always 0. 0: It is invalid. It does not affect the operation. 1: The number of data stored in receive FIFO (RDR register) are made 0.	W
28:24	RSTRG[4:0]	RTS# Output Active Trigger Number Select (Valid in Asynchronous mode (including multi-processor mode), Clock-synchronous mode) These bits enable only when CCR3.FM = 1 and CCR1.CTSE = 0 and CCR0.SSE = 0. Trigger number must be set 15 or less. 0x00: Trigger number 0 ⋮ 0x0F: Trigger number 15 Others: Setting prohibited	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

DRES bit (Receive data ready error select)

Select whether the detection of receive data ready (FRSR.DR = 1) is the cause of RXI interrupt request or the cause of ERI interrupt request.

TTRG[4:0] bits (Transmit FIFO data trigger number)

The TDRE flag is set to 1 when the quantity of transmit data in the transmit FIFO (TDR register) is equal to or less than the specified transmit triggering number. If CCR0.TIE = 1, TXI interrupt request is occurred.

Note: In case the trigger number is set 16 or more, unexpected TXI interrupt will occur.

TFRST bit (Transmit FIFO Data Register Reset)

When the TFRST bit is set to 1, the number of the transmission data stored in transmit FIFO (TDR register) is made 0.

RTRG[4:0] bits (Receive FIFO data trigger number)

The CSR.RDRF flag is set to 1 when the quantity of receive data in the receive FIFO (RDR register) is equal to or greater than the specified receive triggering number. If CCR0.RIE = 1, RXI interrupt request is occurred. When FCR.RTRG is set to 0, RDRF bit is set if the quantity of data in receive FIFO is greater than or equal to 1.

Note: In case the trigger number is set 16 or more, unexpected RXI interrupt will occur.

RFRST bit (Receive FIFO Data Register Reset)

When the RFRST bit is set to 1, the number of the reception data stored in receive FIFO (RDR register) is made 0.

RSTRG[4:0] bits (RTS# Output Active Trigger Number Select)

When the quantity of receive data stored in the receive FIFO (RDR register) is equal to or greater than this number, the RTS# signal is in the High state. When FCR.RSTRG is set to 0, RTS# is in the high state if the quantity of data in receive FIFO is greater than or equal to 1.

Note: In case the trigger number is set 16 or more, RTS# will go to High state at unexpected timing.

31.2.12 DCR : Driver Control Register

Base address: SCIn = 0x8000_1000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_1000

Offset address: 0x30

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	DENGT[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DEAST[4:0]					—	—	—	—	—	—	—	DEPOL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DEPOL	Driver effective polarity select (Valid only in Asynchronous mode) 0: The DE signal is active high. 1: The DE signal is active low.	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
12:8	DEAST[4:0]	Driver Assertion Time (Valid only in Asynchronous mode) Set the driver assertion time. When CCR3.DEN = 1, the following driver assertion time is inserted in addition to the normal transmission waiting time. Driver assertion time = DEAST[4:0] set value × base clock period Setting DEAST[4:0] = 0x00 is prohibited.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
20:16	DENGT[4:0]	Driver negate time (Valid only in Asynchronous mode) Set the driver negation time. When CCR3.DEN = 1, the following driver negate time is inserted after STOP bit transmission end. Driver negate time = DENGT[4:0] set value × base clock period Setting DENGT[4:0] = 0x00 is prohibited.	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

DEPOL bit (Driver effective polarity select)

Select the active level of the DE signal.

DEAST[4:0] bits (Driver Assertion Time)

Set the driver assertion time (= time from the activation of the DE (Driver Enable) signal to the start of the start bit). It is expressed in base clock units (1/8 or 1/16 bit period).

DENGT[4:0] bits (Driver negate time)

Set the driver negation time (= time from the end of the last stop bit of the transmitted message until the DE (Driver Enable) signal is disabled). It is expressed in base clock units (1/8 or 1/16 bit period).

If the transmission data is written during the driver negate time, transmit starting operation is different depends on the writing timing. (The DE signal remains valid, and transmission of the start bit may start after the transmission wait time has elapsed. Also, the DE signal may become invalid once, and start bit transmission may start after the DEAST time + transmission wait time has elapsed.)

31.2.13 CSR : Common Status Register

Base address: SCIn = 0x8000_1000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_1000

Offset address: 0x48

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RDRF	TEND	TDRE	FER	PER	MFF	—	ORER	—	—	—	—	—	DFER	DPER	DCMF
Value after reset:	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RXDMON	—	—	—	—	—	—	—	—	—	—	ERS	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	These bits are read as 0.	R
4	ERS	Error Signal Status Flag (Valid only in Smart card interface mode) 0: Error signal Low not responded 1: Error signal Low responded	R
14:5	—	These bits are read as 0.	R
15	RXDMON	Serial input data monitor The state of the RXD pin without synchronizing by bus clock is shown. 0: RXD pin is the Low level, when CCR1.RINV = 0. RXD pin is the High level, when CCR1.RINV = 1. 1: RXD pin is the High level, when CCR1.RINV = 0. RXD pin is the Low level, when CCR1.RINV = 1.	R
16	DCMF	Data Compare Match Flag (Valid only in Asynchronous mode) 0: No matched 1: Matched	R
17	DPER	Data Compare Match Parity Error Flag (Valid only in Asynchronous mode) 0: No parity error occurred at address match detection. 1: A parity error has occurred at address match detection.	R
18	DFER	Data Compare Match Framing Error Flag (Valid only in Asynchronous mode) 0: No framing error occurred at address match detection. 1: A framing error has occurred at address match detection.	R
23:19	—	These bits are read as 0.	R
24	ORER	Overrun Error Flag 0: No overrun error occurred. 1: An overrun error has occurred.	R
25	—	This bit is read as 0.	R
26	MFF	Mode Fault Error Flag (Valid only in Simple-SPI mode) 0: No mode fault error 1: Mode fault error	R

Bit	Symbol	Function	R/W
27	PER	Parity Error Flag 0: No parity error occurred (Non-FIFO mode) No parity error in all received data in receive FIFO (FIFO mode) 1: A parity error has occurred (Non-FIFO mode) One or more parity errors occurred in received data in receive FIFO (FIFO mode)	R
28	FER	Framing Error Flag 0: No framing error occurred (Non-FIFO mode) No framing error in all received data in receive FIFO (FIFO mode) 1: A framing error has occurred (Non-FIFO mode) One or more framing errors occurred in received data in receive FIFO (FIFO mode)	R
29	TDRE	Transmit Data Empty Flag 0: Transmit data is in TDR register (Non-FIFO mode) The quantity of transmit data written in transmit FIFO exceeds the specified transmit triggering number. (FIFO mode) 1: No transmit data is in TDR register (Non-FIFO mode) The quantity of transmit data written in transmit FIFO is equal to or less than the specified transmit triggering number. (FIFO mode)	R
30	TEND	Transmit End Flag 0: A character is being transmitted or standing by for transmission. 1: Character transfer has been completed.	R
31	RDRF	Receive Data Full Flag 0: No received data is in RDR register (Non-FIFO mode) The quantity of receive data written in receive FIFO falls below the specified receive triggering number (FIFO mode) 1: Received data is in RDR register (Non-FIFO mode) The quantity of receive data written in receive FIFO is equal to or greater than the specified receive triggering number (FIFO mode)	R

ERS bits (Error Signal Status Flag)

[Setting condition]

- When an error signal LOW is sampled.

[Clearing condition]

- When write 1 to CFCLR.ERSC.

DCMF bit (Data Compare Match Flag)

Indicates that SCI detects the match to the comparison data (CCR4.CMPD) with receive data.

Clearing the CCR0.RE bit to 0 does not affect the DCMF flag, which retains its previous state.

[Setting condition]

- Matched to the comparison data (CCR4.CMPD) with receive data, while CCR0.DCME = 1.

[Clearing condition]

- When write 1 to CFCLR.DCMFC.

DPER bit (Data Compare Match Parity Error Flag)

It indicates that a parity error occurred at Address Match detection (reception data match detection).

Clearing the CCR0.RE bit to 0 does not affect the DPER flag, which retains its previous state.

[Setting condition]

- When a parity error was detected by the frame in which an address match was detected.

[Clearing condition]

- When write 1 to CFCLR.DPERC.

DFER bit (Data Compare Match Framing Error Flag)

It indicates that a framing error occurred at Address Match detection (reception data match detection).

Clearing the CCR0.RE bit to 0 does not affect the DFER flag, which retains its previous state.

[Setting condition]

- When a stop bit of the frame in which an Address Match was detected is 0.
  When it is a 2-stop mode, the 1st bit of stop bit judges only whether it's 1 and does not check the 2nd bit of stop bit.

[Clearing condition]

- When write 1 to CFCLR.DFERC.

ORER bit (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

Clearing the CCR0.RE bit to 0 does not affect the ORER flag, which retains its previous state. In simple-I2C mode, this bit is not use.

[Setting condition with Non-FIFO mode (CCR3.FM = 0)]

- When the next data is received before reading out RDR with no-error reception data stored in RDR.
  In RDR, receive data prior to an overrun error occurrence is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, reception data is not forwarded to RDR register.
  Note that, in clock synchronous mode and simple-SPI mode, serial reception will be stop.

[Setting condition with FIFO mode (CCR3.FM = 1)]

- When the next serial reception is completed while the receive FIFO is full of N_{FIFO} (= 16) receive data.
  (N_{FIFO}: Indicates the number of available FIFO stages)

[Clearing condition]

- When write 1 to CFCLR.ORERC.

MFF bit (Mode Fault Error Flag)

This bit indicates mode fault errors. In a multi-master mode, determine the mode fault error occurrence by reading the MFF flag.

[Setting condition]

- Input on the SSn# pin being at the low level during master operation (CCR3.CKE[1:0] = 00b or 01b) in simple SPI mode.

[Clearing condition]

- When write 1 to CFCLR.MFFC.

PER bit (Parity Error Flag)

Indicates that a parity error has occurred during reception and the reception ends abnormally.

Clearing the CCR0.RE bit to 0 does not affect the PER flag, which retains its previous state.

When performing parity check in asynchronous mode, set the address match function disabled (CCR0.DCME = 0). In

Clock-synchronous mode, Simple-SPI mode, and Simple-I2C mode, this bit not used.

[Setting condition]

- When a parity error is detected during reception.
  In FIFO select mode, when one or more parity error is detected in receive FIFO data.
  Although receive data when the parity error occurs is transferred to RDR, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When write 1 to CFCLR.PERC.

FER bit (Framing Error Flag)

Indicates that a framing error has occurred during reception and the reception ends abnormally.

Clearing the CCR0.RE bit to 0 does not affect the FER flag, which retains its previous state.

When performing framing check in asynchronous mode, set the address match function disabled (CCR0.DCME = 0).

In Clock-synchronous mode, Simple-SPI mode, and Simple-I2C mode, this bit not used.

[Setting condition]

- When 0 is sampled as the stop bit during reception.
  In FIFO select mode, when one or more framing error is detected in receive FIFO data.
  In 2-stop-bit mode, only the first stop bit is checked whether it is 1 but the second stop bit is not checked. Note that although receive data when the framing error occurs is transferred to RDR, no RXI interrupt request occurs. In addition, when the FER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When write 1 to CFCLR.FERC.

TDRE bit (Transmit Data Empty Flag)

[Non-FIFO selected (CCR3.FM = 0)]

Indicates the presence of transmit data in the TDR register.

The condition of CCR0.TE = 0 has priority over the condition of 0.

If other conditions that become 1 and conditions that become 0 are satisfied at the same time, the TDRE flag is set to 0.

[Setting condition]

- When CCR0.TE is 0.
- When data is transmitted from the TDR register to the TSR register.

[Clearing condition]

- When write 1 to CFCLR.TDREC.
- When the transmission data is written to the TDR register during CCR0.TE = 1.

[FIFO selected (CCR3.FM = 1)]

- Indicates that data has been transferred from the transmit FIFO (TDR) into the transmit shift register (TSR), the quantity of data in transmit FIFO has fallen below the specified transmit triggering number.
  When the condition which becomes 1 and the condition which becomes 0 were formed at the same time, TDRE flag will be 0. After that, when the number of data stored in transmit FIFO is judged, and if that is equal to or less than TTRG value, TDRE is set to 1 after 1 PCLKM.

[Setting condition]

- When the quantity of transmit data written in transmit FIFO is equal to or less than the specified transmit triggering number.*1

[Clearing condition]

- When write 1 to CFCLR.TDREC.
- When the transmission data is written to transmit FIFO by the DMAC (except ch5).

Note 1. The transmit FIFO is 16-stage FIFO, the maximum number of data that can be written when the TDRE flag is 1 is "N_{FIFO} (= 16) - FTSR.T[5:0]". Even if data any more is written, data is discarded.

TEND bit (Transmit End Flag)

[Non-FIFO selected (CCR3.FM = 0), and Not Smart card interface mode (CCR3.MOD[2:0] ≠ 001b)]

Indicates completion of transmission.

[Setting condition]

- When CCR0.TE is 0.
- When the CCR0.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When the TDR register is not updated at transmission of the tail-end bit of a character.
- When the TDR register is not updated at the end of DE negate time with DE control function enable (CCR3.DEN = 1).

[Clearing condition]

- After the synchronization delay time has elapsed since the transmission data was written to the TDR register during CCR0.TE = 1.
- When write 1 to CFCLR.TDREC during CCR0.TE = 1.

[Non-FIFO selected (CCR3.FM = 0), and Smart card interface mode (CCR3.MOD[2:0] = 001b)]

With no error signal from the receiving side, this bit is set to 1 when next data is ready to be transferred to the TDR register.

[Setting condition]

- When CCR0.TE is 0.
- When the CCR0.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When ERS flag is 0 and TDR register is not updated after 1-byte transmission + fixed time, the set timing is determined by setting CCR3 register as listed below.
  - When GM = 0 and BLK = 0, 12.5 etu after the start of transmission
  - When GM = 0 and BLK = 1, 11.5 etu after the start of transmission
  - When GM = 1 and BLK = 0, 11.0 etu after the start of transmission
  - When GM = 1 and BLK = 1, 11.0 etu after the start of transmission

[Clearing condition]

- After the synchronization delay time has elapsed since the transmission data was written to the TDR register during CCR0.TE = 1.
- When write 1 to CFCLR.TDREC during CCR0.TE = 1.

[FIFO selected (CCR3.FM = 1)]

Indicates that the transmit FIFO does not contain valid data when transmitting the last bit of a serial character, so the transmission is halted.

[Setting condition]

- TEND is set to 1 when transmit FIFO does not contain transmit data when the last bit of a one-byte serial character is transmitted.
- When the TDR register is not updated at the end of DE negate time with DE control function enable (CCR3.DEN = 1).

[Clearing condition]

- After the synchronization delay time has elapsed since the transmission data was written to the TDR register during CCR0.TE = 1.

RDRF bit (Receive Data Full Flag)

[Non-FIFO selected (CCR3.FM = 0)]

Indicates the presence of receive data in the RDR register.

[Setting condition]

- When the reception ends normally, and receive data is forwarded from the RSR register to the RDR register.

[Clearing condition]

- When write 1 to CFCLR.RDRFC.
- When a data is read from the RDR register.

[FIFO selected (CCR3.FM = 1)]

Indicates that receive data has been transferred to the receive FIFO data register (RDR), and the quantity of data in receive FIFO is equal to or greater than the specified receive triggering number. When FCR.RTRG is set to 0, RDRF is set if receive FIFO is not empty.

[Setting condition]

RDRF is set to 1 when the quantity of receive data in receive FIFO is equal to or greater than the specified receive triggering number.*1

[Clearing condition]

- When write 1 to CFCLR.RDRFC.
- When the reception data is read from receive FIFO by the DMAC (except ch5).
  When setting condition and clearing condition are occurred at the same time, RDRF flag will be 0. After that, if the data count in receive FIFO is equal to or greater than RTRG value, RDRF is set to 1 after 1 PCLKM.

Note 1. Since the receive FIFO is N_{FIFO} (= 16) stage, the maximum quantity of data that can be read when RDRF is 1 is equal to FRSR.R[5:0]. Further data can be read, but the value is undefined.

Note: Except when interrupting communication, RDRF and TDRE should not be cleared by CFCLR register.

31.2.14 ISR : Simple I2C Status Register

Base address: SCIn = 0x8000_1000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_1000

Offset address: 0x4C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	IICSTIF	—	—	IICACKR
Value after reset:	0	0	0	0	0	0	0	0	0	0	x	x	0	x	0	0

Bit	Symbol	Function	R/W
0	IICACKR	ACK Reception Data Flag 0: ACK received 1: NACK received	R
1	—	This bit is read as 0.	R
2	—	The read value is undefined.	R
3	IICSTIF	Issuing of Start, Restart, or Stop Condition Completed Flag 0: There are no requests for generating conditions or a condition is being generated. 1: A start, restart, or stop condition is completely generated.	R
4	—	The read value is undefined.	R
5	—	The read value is undefined.	R
31:6	—	These bits are read as 0.	R

IICACKR bit (ACK Reception Data Flag)

Received ACK and NACK bits can be read from this bit.

The IICACKR flag is updated at the rising of SCLn clock for the ACK/NACK receiving bit.

IICSTIF bit (Issuing of Start, Restart, or Stop Condition Completed Flag)

After generating a condition, this bit indicates that the generation is completed. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ, do it after clearing the IICSTIF flag.

When the IICSTIF flag is 1 while CCR0.TEIE = 1, an STI request is output.

[Setting condition]

Completion of the generation of a start, restart, or stop condition (when setting condition and clearing condition are occurred at the same time, IICSTIF will be 0).

[Clearing condition]

- Writing 1 to ICFCLR.IICSTIFC bit
- When operation is not in simple-I2C
- Writing 0 to CCR0.TE bit

31.2.15 FRSR : FIFO Receive Status Register

Base address: SCIn = 0x8000_1000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_1000

Offset address: 0x50

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	FNUM[5:0]						—	—	PNUM[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	R[5:0]						—	—	—	—	—	—	—	DR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DR	Receive Data Ready flag 0: Receiving is in progress, or no received data has remained in receive FIFO after normally completed receiving. (Receive FIFO is empty.) 1: The following receive data does not come for a fixed period after storing data below the threshold.	R
7:1	—	These bits are read as 0.	R
13:8	R[5:0]	Receive FIFO Data Count (Valid in Asynchronous mode (including multi-processor), Clock synchronous mode, Simple-SPI mode, when CCR3.FM is 1.) Indicate the quantity of data in receive FIFO.	R
15:14	—	These bits are read as 0.	R
21:16	PNUM[5:0]	Parity Error Count (Valid only in Asynchronous mode) Indicates the quantity of data with a parity error among the data in receive FIFO.	R
23:22	—	These bits are read as 0.	R
29:24	FNUM[5:0]	Framing Error Count (Valid only in Asynchronous mode) Indicates the quantity of data with a framing error among the data in receive FIFO.	R
31:30	—	These bits are read as 0.	R

DR bit (Receive Data Ready flag)

Indicates that the quantity of data stored in the receive FIFO falls below the specified receive triggering number, and that no next data has been received yet after the elapse of 15 etu from the last stop bit in asynchronous mode. This bit is valid only asynchronous mode (including multi-processor mode) and FIFO selected. In other modes, this bit does not set to 1.

[Setting conditions]

DR is set to 1 when both of the following conditions are met.

- Data in receive FIFO is less than the specified receive triggering number, and no next data is received after the elapse of 15 etu[*1] from the last stop bit.

- CSR.FER, PER flags are 0.

[Clearing conditions]

- When all receive data in the receive FIFO is read and 1 is written to FFCLR.DRC.
- When CCR3.FM bit is 0.

Note 1. This is equivalent to one and a half (1.5) frames in the 8-bit format with one stop bit (etu: elementary time unit).

R[5:0] bits (Receive FIFO Data Count)

Indicate the quantity of receive data stored in receive FIFO.

0x00 means no receive data. N_{FIFO} (= 16) means receive FIFO is full.

PNUM[5:0] bits (Parity Error Count)

Indicate the quantity of data with parity error in the received FIFO.

FNUM[5:0] bits (Framing Error Count)

Indicate the quantity of data with framing error in the received FIFO.

31.2.16 FTSR : FIFO Transmit Status Register

Base address: SCIn = 0x8000_1000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_1000

Offset address: 0x54

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	T[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	T[5:0]	Transmit FIFO Data Count (Valid in Asynchronous mode (including multi-processor), Clock synchronous mode, Simple-SPI mode, when CCR3.FM is 1.) Indicate the quantity of non-transmitted data stored in transmit FIFO.	R
31:6	—	These bits are read as 0.	R

T[5:0] bits (Transmit FIFO Data Count)

Indicate the quantity of non-transmitted data stored in transmit FIFO.

0x00 means no un-transmit data. N_{FIFO} (= 16) means transmit FIFO is full.

31.2.17 CFCLR : Common Flag Clear Register

Base address: SCIn = 0x8000_1000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_1000

Offset address: 0x68

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RDRFC	—	TDREC	FERC	PERC	MFFC	—	ORERC	—	—	—	—	—	DFERC	DPERC	DCMFC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	ERSC	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	—	The write value should be 0.	W
4	ERSC	ERS clear 0: No effect 1: Clear the CSR.ERS bit	W
15:5	—	The write value should be 0.	W
16	DCMFC	DCMF clear 0: No effect 1: Clear the CSR.DCMF bit	W
17	DPERC	DPER clear 0: No effect 1: Clear the CSR.DPER bit	W
18	DFERC	DFER clear 0: No effect 1: Clear the CSR.DFER bit	W
23:19	—	The write value should be 0.	W
24	ORERC	ORER clear 0: No effect 1: Clear the CSR.ORER bit	W
25	—	The write value should be 0.	W
26	MFFC	MFF clear 0: No effect 1: Clear the CSR.MFF bit	W
27	PERC	PER clear 0: No effect 1: Clear the CSR.PER bit	W
28	FERC	FER clear 0: No effect 1: Clear the CSR.FER bit	W
29	TDREC	TDRE clear 0: No effect 1: Clear the CSR.TDRE bit	W
30	—	The write value should be 0.	W
31	RDRFC	RDRF clear 0: No effect 1: Clear the CSR.RDRF bit	W

31.2.18 ICFCLR : Simple I2C Flag Clear Register

Base address: SCIn = 0x8000_1000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_1000

Offset address: 0x6C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	IICSTIFC	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	The write value should be 0.	W
3	IICSTIFC	IICSTIF clear 0: No effect 1: Clear the ISR.IICSTIF bit	W
31:4	—	The write value should be 0.	W

31.2.19 FFCLR : FIFO Flag Clear Register

Base address: SCIn = 0x8000_1000 + 0x0400 × n (n = 0 to 4)
SCI5 = 0x8100_1000

Offset address: 0x70

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DRC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DRC	DR clear 0: No effect 1: Clear the FRSR.DR bit	W
31:1	—	The write value should be 0.	W

31.3 Asynchronous Mode

Figure 31.3 shows the general format for asynchronous serial communications.

One frame consists of a start bit (low level), transmit/receive data, a parity bit, and stop bits (high level).

In asynchronous serial communications, the communications line is held in the mark state (high level) when not communicating.

The SCI monitors the communications line. When the SCI detects a start bit, it starts serial communication. The detection condition of the start bit changes according to the CCR3.RXDESEL setting. SCI regards space (Low level) as a start bit when CCR3.RXDESEL is 0. SCI regards a fall edge as a start bit when RXDESEL is 1.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications. Both the transmitter and the receiver also have a double-buffered structure (it has also FIFO mode), so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

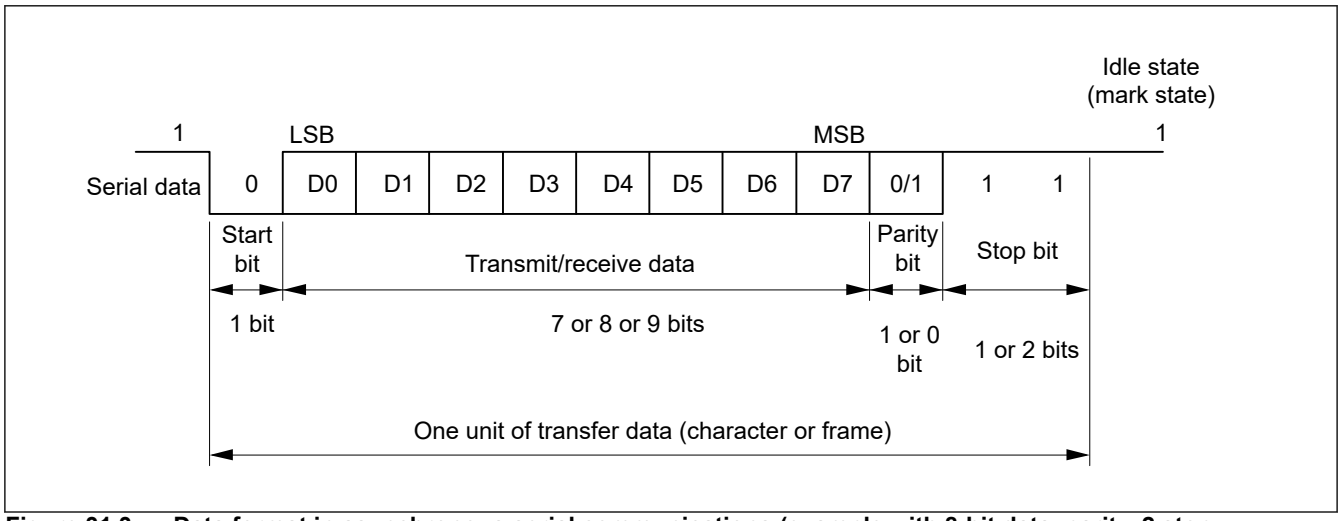


Figure 31.3 Data format in asynchronous serial communications (example with 8-bit data, parity, 2 stop bits)

31.3.1 Serial Data Transfer Format

Table 31.22 lists the serial data transfer formats that can be used in asynchronous mode. Any of 18 transfer formats can be selected according to the CCR1 and CCR3 settings.

For details of multi-processor function, see section 31.4. Multi-Processor Communication Function.

Table 31.22 Serial transfer formats (asynchronous mode) (1 of 2)

CCR3				CCR1	Serial transfer format and frame length
CHR[1]	CHR[0]	MP	STP	PE	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	0	0	0	St 9-bit data STP
0	0	0	1	0	St 9-bit data STP STP
0	0	0	0	1	St 9-bit data P STP
0	0	0	1	1	St 9-bit data P STP STP
1	0	0	0	0	St 8-bit data STP
1	0	0	1	0	St 8-bit data STP STP
1	0	0	0	1	St 8-bit data P STP
1	0	0	1	1	St 8-bit data P STP STP
1	1	0	0	0	St 7-bit data STP
1	1	0	1	0	St 7-bit data STP STP

Table 31.22 Serial transfer formats (asynchronous mode) (2 of 2)

CCR3				CCR1	Serial transfer format and frame length
CHR[1]	CHR[0]	MP	STP	PE	1 2 3 4 5 6 7 8 9 10 11 12 13
1	1	0	0	1	St | 7-bit data | P | STP
1	1	0	1	1	St | 7-bit data | P | STP | STP
0	0	1	0	—	St | 9-bit data | MPB | STP
0	0	1	1	—	St | 9-bit data | MPB | STP | STP
1	0	1	0	—	St | 8-bit data | MPB | STP
1	0	1	1	—	St | 8-bit data | MPB | STP | STP
1	1	1	0	—	St | 7-bit data | MPB | STP
1	1	1	1	—	St | 7-bit data | MPB | STP | STP

[Symbol explanation]

St : Start bit
STP : Stop bit
P : Parity bit
MPB : Multi-processor bit

31.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode

In asynchronous mode, the SCI operates on a base clock with a frequency of 16 times[*1] the bit rate. In reception, the SCI samples the falling edge of the start bit using the base clock and performs internal synchronization[*2]. When sampling timing does not adjust ("CCR4.ASEN = 0b" or "CCR4.ASEN = 1b and CCR4.AST[2:0] = 000b"), receive data is sampled at the rising edge of the 8th pulse[*1] of the base clock, data is latched at the middle of each bit, as shown in Figure 31.4. Thus, the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left|(0.5 - \frac{1}{2N}) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F)\right| \times 100[\%] \cdots \text{Equation (1)}$$

M: Reception margin

N: Ratio of bit rate to clock

(N = 16, when CCR2.ABCSE = 0 and CCR2.ABCS = 0, N = 8, when CCR2.ABCS = 1, N = 6, when CCR2.ABCSE = 1)

D: Duty cycle of clock (D = 0.5 to 1.0)

L: Frame length (L = 9 to 13)

F: Absolute value of clock frequency deviation

Assuming values of N = 16, F = 0 and D = 0.5 in formula (1), the reception margin is determined by the formula below.

$M = \{0.5 - 1/(2 \times 16)\} \times 100(\%) = 46.875\%$

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

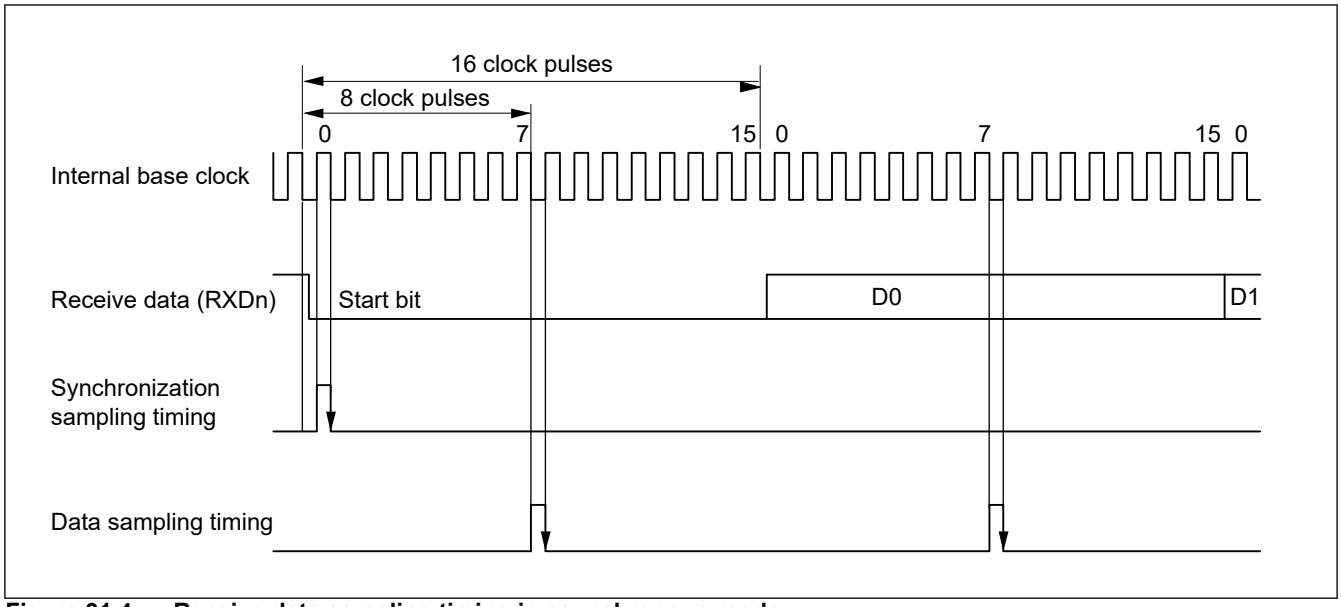


Figure 31.4 Receive data sampling timing in asynchronous mode

Note 1. This is an example when CCR2.ABCS = 0 and CCR2.ABCSE = 0. When CCR2.ABCS = 1 and CCR2.ABCS = 0, a frequency of 8 times the bit rate is used as a base clock and receive data is sampled at the rising edge of the 4th pulse of the base clock. When CCR2.ABCSE is 1, a sextuple frequency of a bit rate is a base clock and receive data is sampled at the rising edge of the 3rd pulse of the base clock.

Note 2. The determination condition of the start bit is as follows.

In the case of the function of adjust sampling timing is OFF (CCR4.ASEN = 0):

The determination condition of a start bit is that Low beyond half bit length continues. It is same as the sampling timing.

In Figure 31.4, the low period should be kept over 8 cycles to detect a start bit. If Low period does not keep over 8 cycles, the module judges this as a noise. So, the module does not start reception and wait start bit.

In the case of the function of adjust sampling timing is ON (CCR4.ASEN = 1):

The determination condition of a start bit is that Low keeps up until the sampling timing.

Adjusting the sampling timing forward (CCR4.AJD = 1) increases the possibility of erroneously determining a noise as the start bit.

31.3.3 Clock

Either an internal clock generated by the baud rate generator or an external clock input to the SCKn pin can be selected as the SCI's transfer clock, according to the setting of CCR3.CKE[1:0]. When an external clock is input to the SCKn pin, the clock frequency should be 16 times the bit rate (when CCR2.ABCS = 0) and 8 times the bit rate (when CCR2.ABCS = 1).

When the SCI is operated on an internal clock, the clock can be output from the SCKn pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in Figure 31.5.

If you selected an internal clock, the SCK pin is outputted after CCR0.TE is set to 1 or CCR0.RE is set to 1.

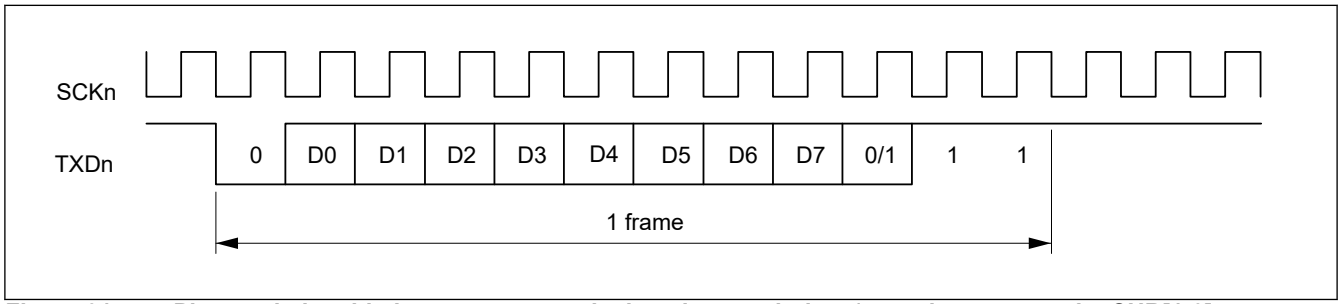


Figure 31.5 Phase relationship between output clock and transmit data (asynchronous mode: CHR[1:0] = 10b, PE = 1, MP = 0, STP = 1)

31.3.4 Double-Speed Operation and frequency of 6 times the bit rate

When CCR2.ABCS is set to 1, the SCI operates on the bit rate twice that in the case where ABCS is set to 0. And when CCR2.BGDM is set to 1, the cycle of the base clock is halved and the bit rate is doubled from that in the case where BGDM is set to 0. When CCR3.CKE[1] is set to 0 and the baud rate generator is selected, setting the ABCS and BGDM bits to 1 allows the SCI to operate on a bit rate four times that in the case ABCS = 0 and BGDM = 0.

When CCR2.ABCSE is set to 1, the number of base clock pulses are 6 during a period of 1 bit, and the base clock frequency is half. And SCI works 16/3 times of bit rate compared with a case of CCR2.ABCS = 0, CCR2.BGDM = 0 and CCR2.ABCSE = 0.

As shown by Formula (1) in section 31.3.2. Receive Data Sampling Timing and Reception Margin in Asynchronous Mode, the reception margin decreases when CCR2.ABCS is set to 1 or CCR2.ABCSE is set to 1. Therefore, if the desired bit rate can be obtained with CCR2.ABCS set to 0 and/or CCR2.ABCSE set to 0, it is recommended to use the SCI with CCR2.ABCS set to 0 and/or CCR2.ABCSE set to 0.

31.3.5 CTS and RTS Function

The CTS function is the transmission control function by the CTSn# pin. Setting the CCR1.CTSE bit to 1 enables the CTS function. For the functions of CTS and RTS, you can select the alternate setting that uses either function with one terminal or the dedicated setting that uses each function independently with two terminals. This setting is done with the CCR1.CTSPEN bit.

When the CTS function is enabled, placing the low level on the CTSn# pin, causes transmission to start.

Even if the CTSn# pin goes high after transmission starts, the frame being transmitted is not affected and transmission will continue.

The RTS function is the transmission request function by the RTSn# pin. In the RTS function, the RTSn# pin output low level, when reception becomes possible. Conditions for output of the low and high levels are shown below.

[Conditions of low-level output]

(1) Non-FIFO selected when all of the following conditions are satisfied.

- CCR0.RE is 1
- When in receivable state
- There are no received data yet to be read
- CSR.ORER, FER, PER flags are all 0

(2) FIFO selected when all of the following conditions are satisfied.

- CCR0.RE is 1
- When the quantity of receive data written in receive FIFO (RDR) are less than the setting value of FCR.RSTRG[4:0]
- CSR.ORER (RDR.ORER) is 0

[Condition for high-level output]

When the conditions of low-level output have not been satisfied.

31.3.6 Address Match (receive data match detection) Function

The address match function can be used only for the asynchronous mode.

If CCR0.DCME is set to 1,[*2] when one frame of data has been received, SCI compares that receive data with the data which is set to CCR4.CMPD. If SCI detects the match to the comparison data (CCR4.CMPD[*1]) with receive data, SCI can issue RXI interrupt request.

If CCR3.MP bit is set to 0, this comparative target in communication data is valid only data field in receive format. In multi-processor mode (CCR3.MP = 1), if CCR0.IDSEL bit is set to 1, the reception data at which MPB bit is 1 detects address match or unmatch, and the reception data at which MPB bit is 0 detects always unmatch. If CCR0.IDSEL bit is set to 0, SCI detects address match or unmatch despite the value of the MPB bit of the reception data at every reception complete.

Until SCI detects the match to the comparison data (CCR4.CMPD) with receive data, the communication data is skipped (discarded), and SCI can not detect parity error, framing error.

When SCI detects the match, the CCR0.DCME is automatically cleared, and CSR.DCMF is set to 1. If CCR0.IDSEL bit is set to 1 at this time, CCR0.MPIE bit is automatically cleared. If CCR0.IDSEL bit is set to 0 at this time, CCR0.MPIE bit is kept. At the same time, if CCR0.RIE is set to 1, SCI issues RXI interrupt request.

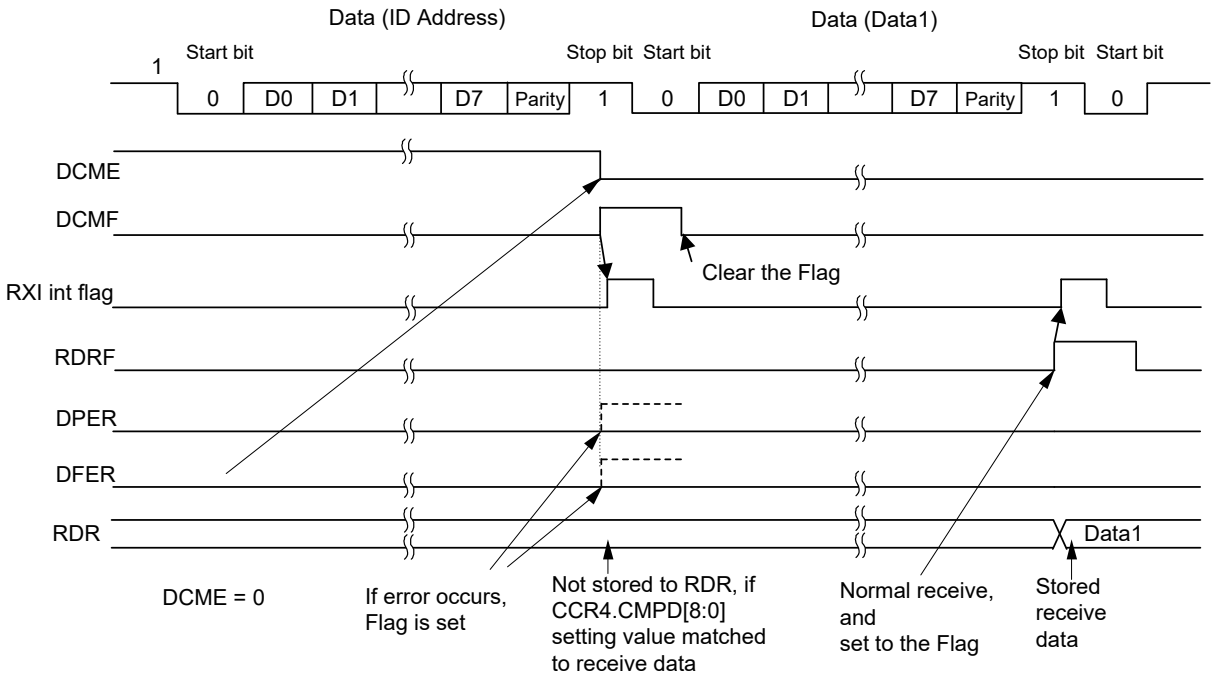
If SCI detects framing error in comparative receive data which is detected the match, CSR.DFER is set to 1, and if SCI detects parity error in that frame, CSR.DPER is set to 1. That comparative receive data and MPB bit are not stored to RDR register, and CSR.RDRF is retained to 0.

After SCI detects the match, and the CCR0.DCME is automatically cleared, it receives next data continuously in current register setting.

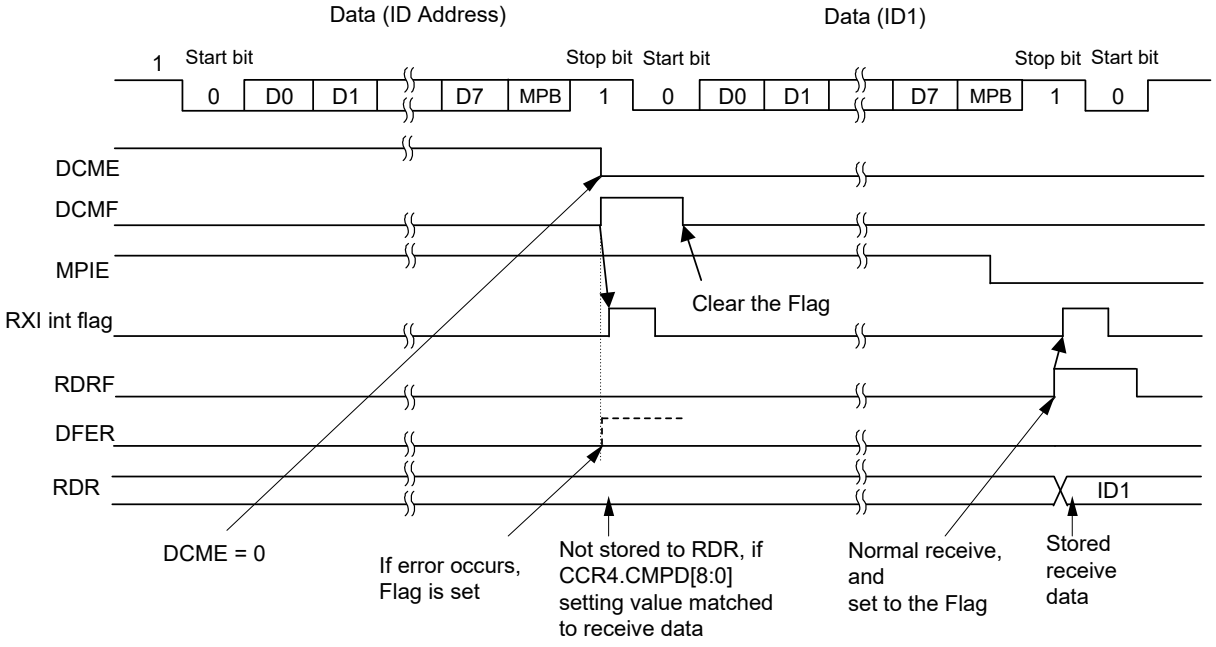
When CSR.DFER flag or CSR.DPER flag is set, the address much is not detected. Before making the address match function effective, please be sure to set CSR.DFER and CSR.DPER flag as 0.

Figure 31.6 and Figure 31.7 show the address match function example.

Note 1. This comparative target can select one length of 3 types, they are CMPD[6:0] with 7-bit length enable, CMPD[7:0] with 8-bit, and CMPD[8:0] with 9-bit length.
Note 2. Set the CCR0.DCME bit to 1 before receiving the start bit of the received frame that performs address matching.

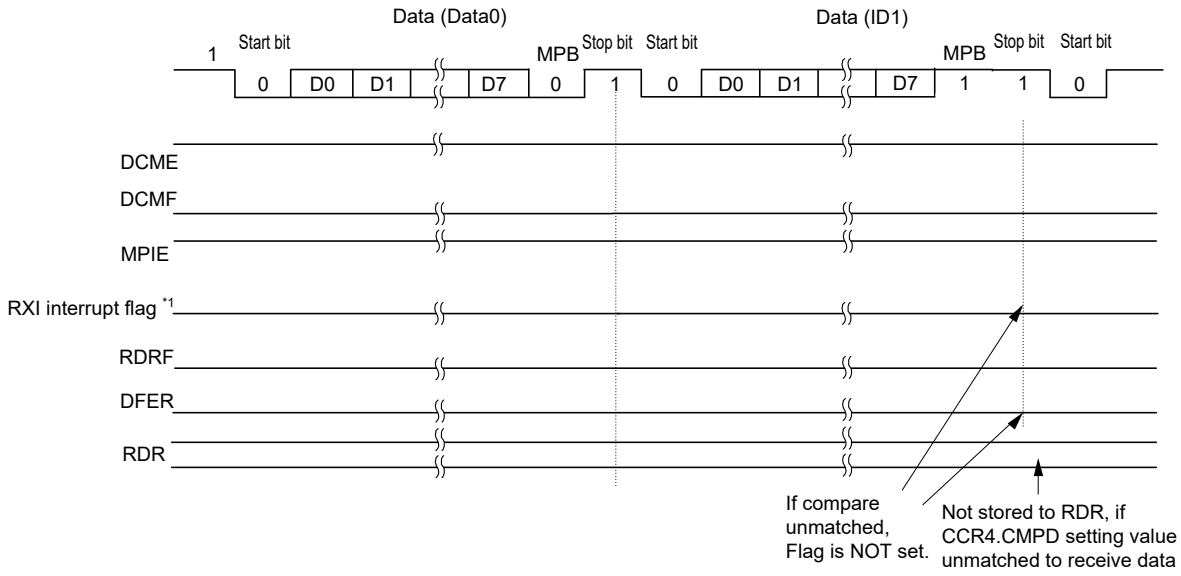


(a) Example of Compare matched between receive data and CCR4.CMPD[8:0] (8-bit length/parity/Normal mode)

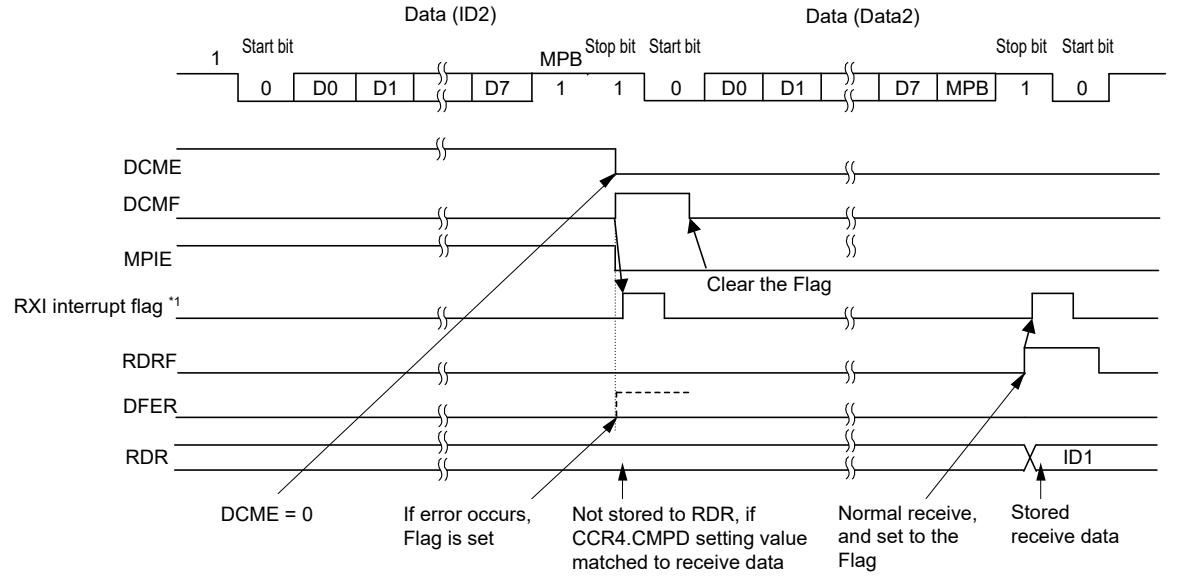


(b) Example of Compare matched between receive data and CCR4.CMPD[8:0] (8-bit length/Multi-processor mode)

Figure 31.6 Example of addresss match (1) non-multi-processor mode / 8-bit data



(a) Example of Compare unmatched between receive data and CCR4.CMPD (8-bit length / IDSEL = 1 / Multi-processor mode)



(b) Example of Compare matched between receive data and CCR4.CMPD (8-bit length / IDSEL = 1 / Multi-processor mode)

Note 1. See section 11, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 31.7 Example of address match (2) multi-processor mode / 8bit-data

31.3.7 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, start by writing 0 to CCR0.TE and CCR0.RE (or writing the initial value to CCR0) and then continue through the Procedure (select to Non-FIFO or FIFO) for SCI given in Figure 31.8 or Figure 31.9. Whenever the operating mode or transfer format is changed, CCR0.TE and CCR0.RE must be initialized before the change is made.

When the external clock is used in asynchronous mode, ensure that the clock signal is supplied even during initialization.

Note that setting the CCR0.RE bit to 0 initializes neither the ORER, FER, PER, RDRF, DR and RDAT. When FIFO selected, even if the TE bit is set as 0, the TEND flag is not initialized, so please be careful.

Please be also careful at the time of change in the operation mode.

Moreover, note that switching the value of the CCR0.TE bit from 0 to 1 while the CCR0.TIE bit is 1 leads to the generation of a TXI interrupt request.

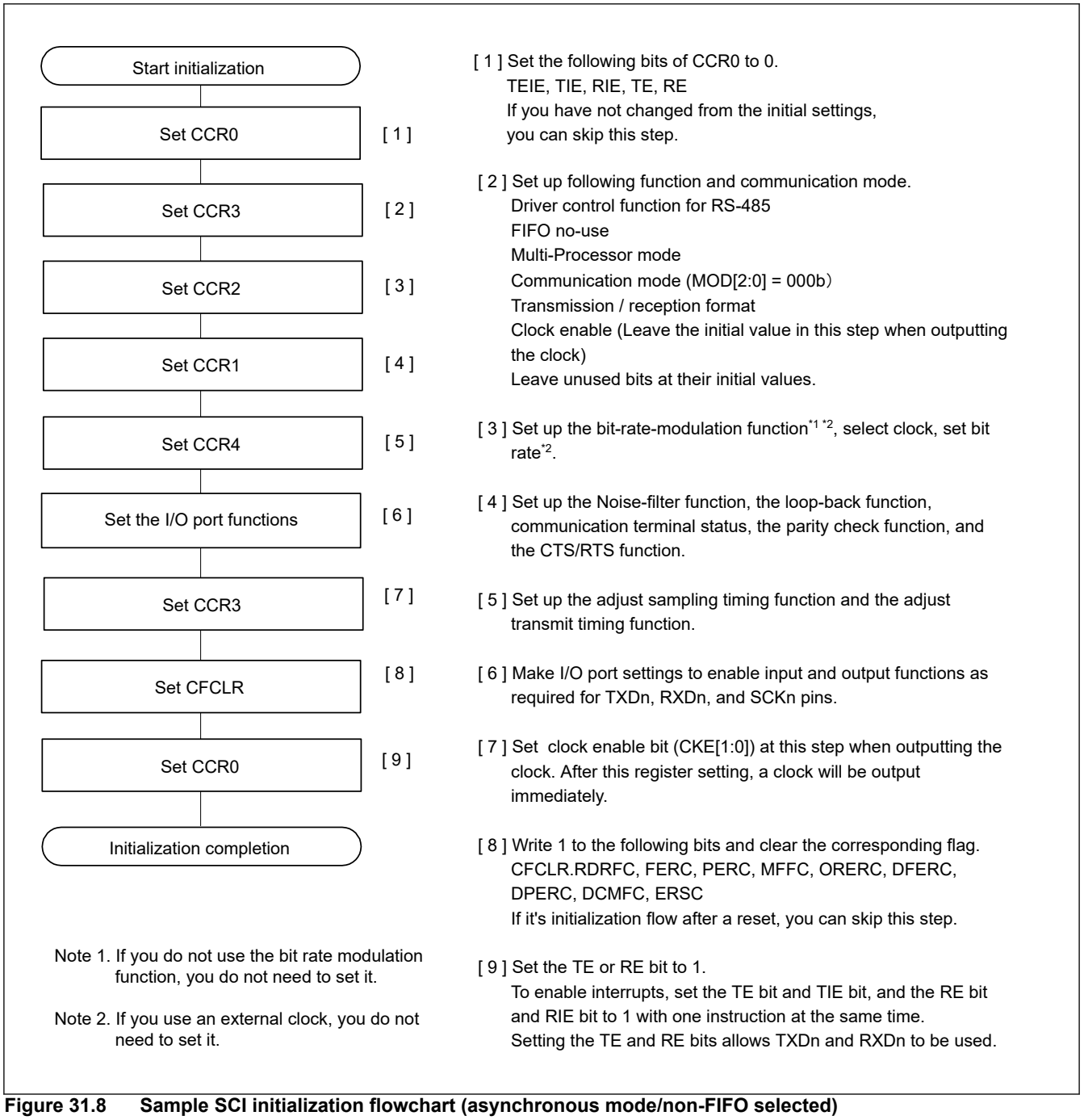
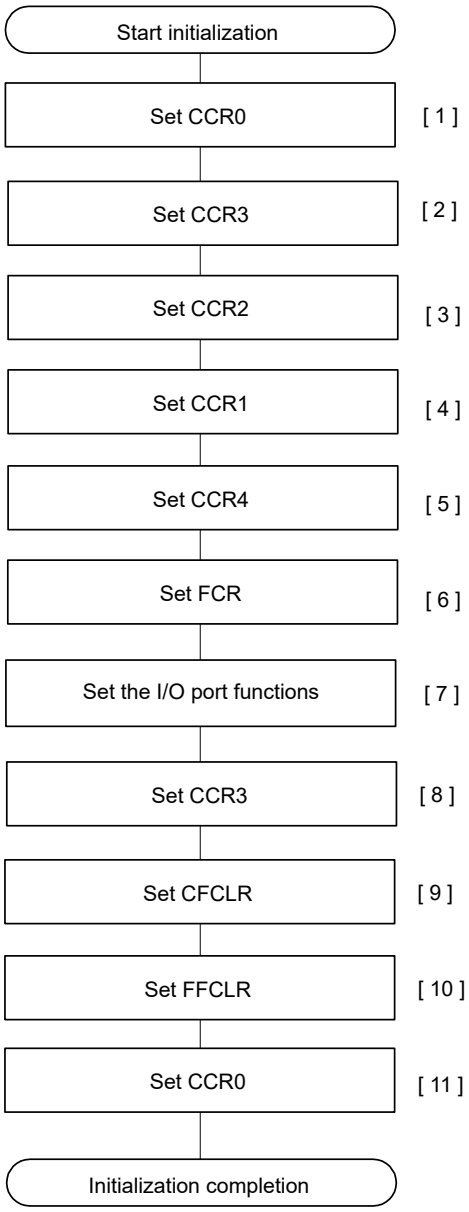


Figure 31.8 Sample SCI initialization flowchart (asynchronous mode/non-FIFO selected)



Note 1. If you do not use the bit rate modulation function, you do not need to set it.
Note 2. If you use an external clock, you do not need to set it.

[1] Set the following bits of CCR0 to 0.
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set up following function and communication mode.
Driver control function for RS-485
FIFO use
Multi-Processor mode
Communication mode (MOD[2:0] = 000b)
Transmission / reception format
Clock enable (Leave the initial value in this step when outputting the clock)
Leave unused bits at their initial values.

[3] Set up the bit-rate-modulation function*1 *2, select clock, set bit rate*2.

[4] Set up the Noise-filter function, the loop-back function, communication terminal status, the parity check function, and the CTS/RTS function.

[5] Set up the adjust sampling timing function and the adjust transmit timing function.

[6] Set the TFRST and RFRST to 1 to empty FIFO.
Set the DRES, TTRG[4:0], RTRG[4:0], and RSTRG[4:0] bits

[7] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[8] Set clock enable bit (CKE[1:0]) at this step when outputting the clock. After this register setting, a clock will be output immediately.

[9] Write 1 to the following bits and clear the corresponding flag.
CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
If it's initialization flow after a reset, you can skip this step.

[10] Write 1 to the FFCLR.DRC and clear the corresponding flag.
If it's initialization flow after a reset, you can skip this step.

[11] Set the TE or RE bit to 1.
To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Figure 31.9 Sample SCI initialization flowchart (asynchronous mode/FIFO selected)

Figure 31.10 shows an example of the timing when data is transmitted after reset is released, and SCI is set to asynchronous mode according to Figure 31.8 or Figure 31.9. As shown in the figure, when the pin function is set to the TXD pin, the CCR0.TE bit is 0, so the pin is high impedance. When transmit data is written after setting the CCR0.TE bit to 1, data transmission starts. There is a transmission wait time from writing TDR to data transmission starts. In asynchronous mode, TXD is high during this period.

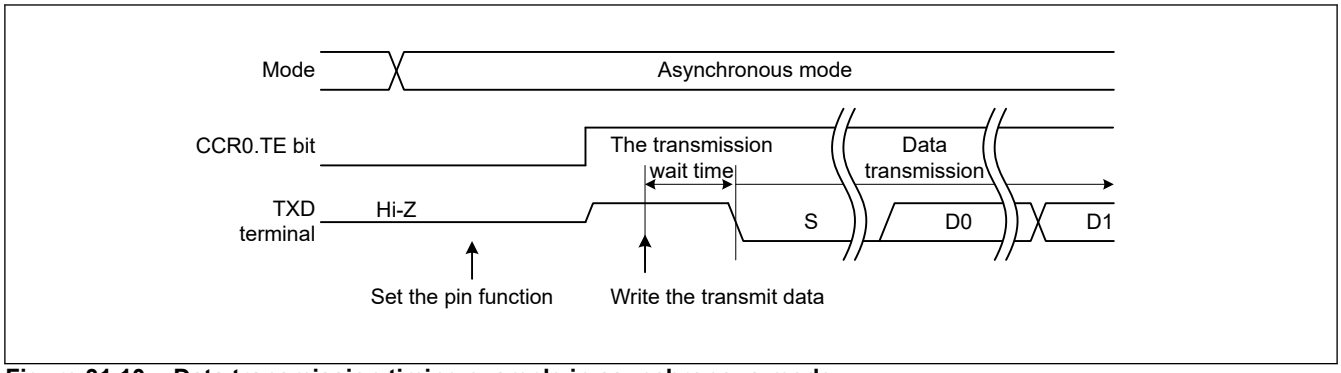


Figure 31.10 Data transmission timing example in asynchronous mode

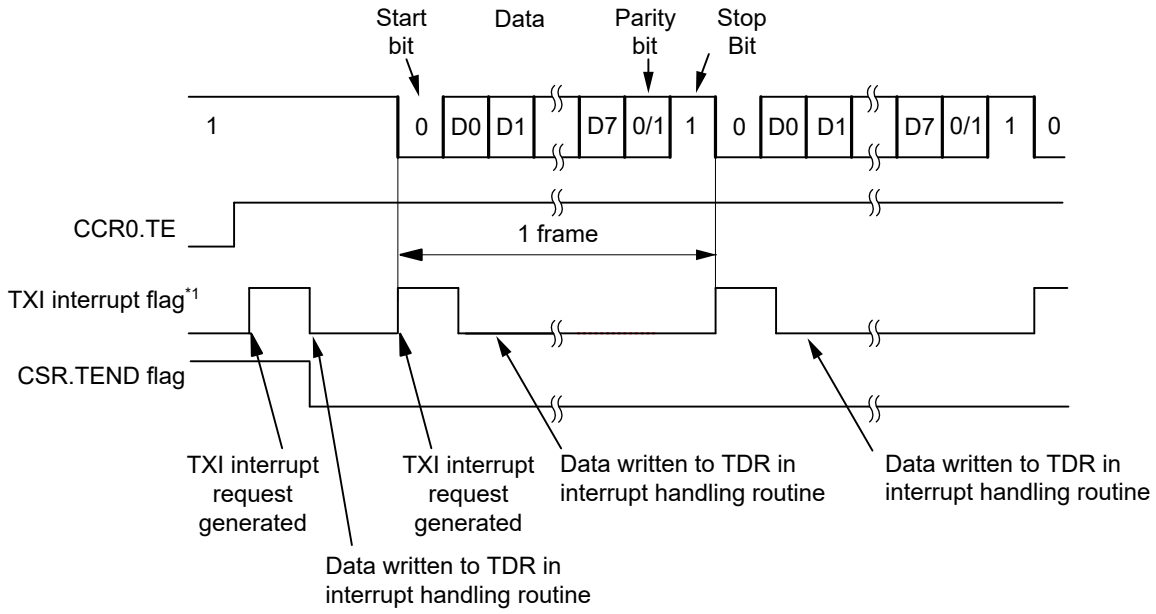
31.3.8 Serial Data Transmission (Asynchronous Mode)

(1) Non-FIFO selected

Figure 31.11, Figure 31.12, and Figure 31.13 show an example of the operation for serial transmission in asynchronous mode. In serial transmission, the SCI operates as described below.

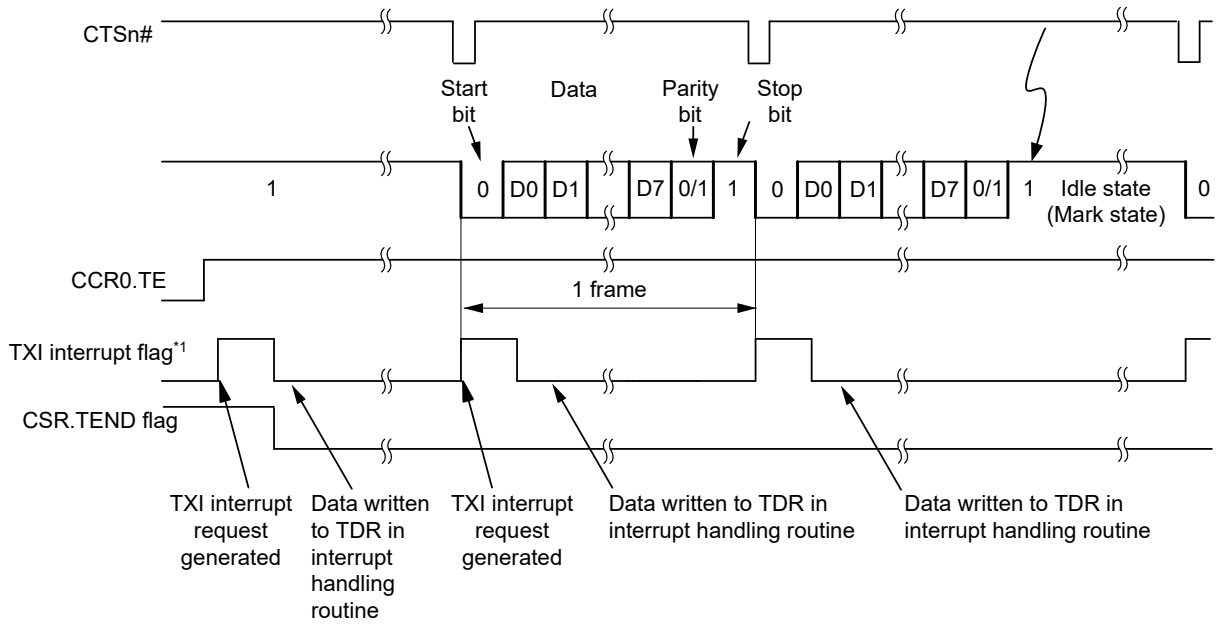
1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt handling routine. At the beginning of transmission, set 1 to CCR0.TE and CCR0.TIE simultaneously. Then the TXI interrupt request is generated.
2. Transmission starts at data transfer from TDR to TSR when CCR1.CTSE = 0 (CTS function is disabled) or when the CTS# pin level is low. If CCR0.TIE is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next transmit data to TDR in the TXI interrupt handling routine before transmission of the current transmit data is completed. Write the last transmission data, then the last data transmit start, and TXI is generated. When TEI interrupt requests are in use, before the last data transmit end, set the CCR0.TIE bit to 0 (a TXI interrupt request is disabled) and the CCR0.TEIE bit to 1 (a TEI interrupt request is enabled) using the TXI handling routine.
3. Data is sent from the TXDn pin in the following order: start bit, transmit data, parity bit or multi-processor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks for updating of (writing to) TDR at the time of stop bit output.
5. When TDR is updated, setting of CCR1.CTSE to 0 (CTS function is disabled) or a low level input on the CTSn# pin cause the next transfer of the next transmit data from TDR to TSR and sending of the stop bit, after which serial transmission of the next frame starts.
6. If TDR is not updated, CSR.TEND is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output. If the CCR0.TEIE is 1 at this time, a TEI interrupt request is generated.

Figure 31.15 shows the example of Serial Transmission Flowchart.



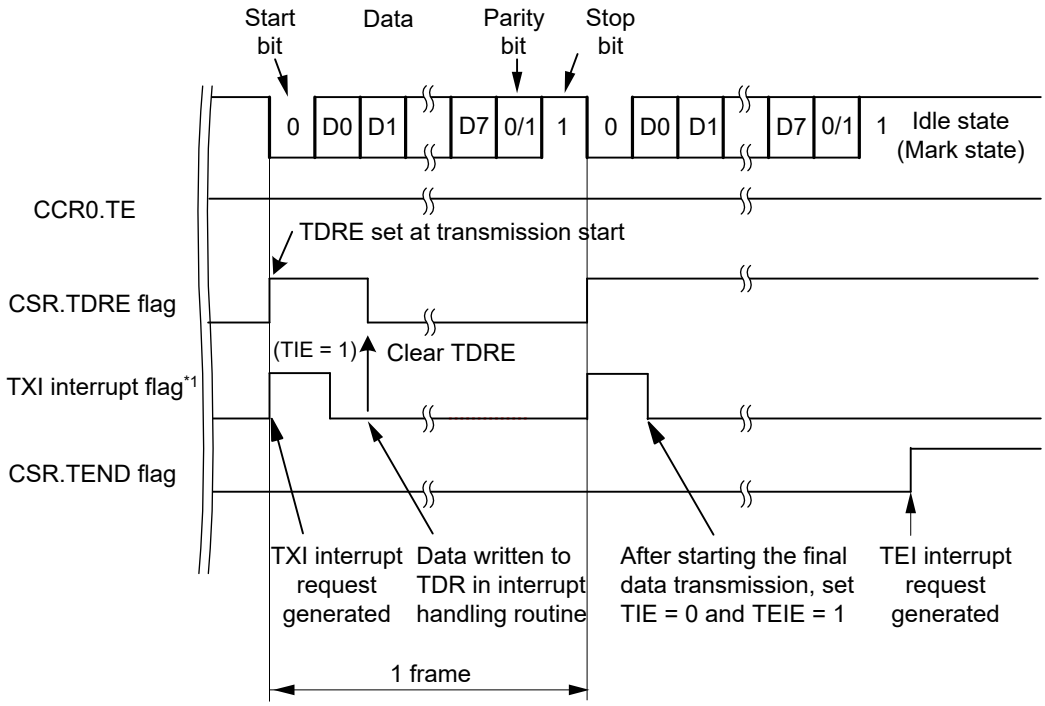
Note 1. See section 11, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 31.11 Example of operation for serial transmission in asynchronous mode (1) (with 8-bit data, parity, 1 stop bit, CTS function not used, at the beginning of transmission)



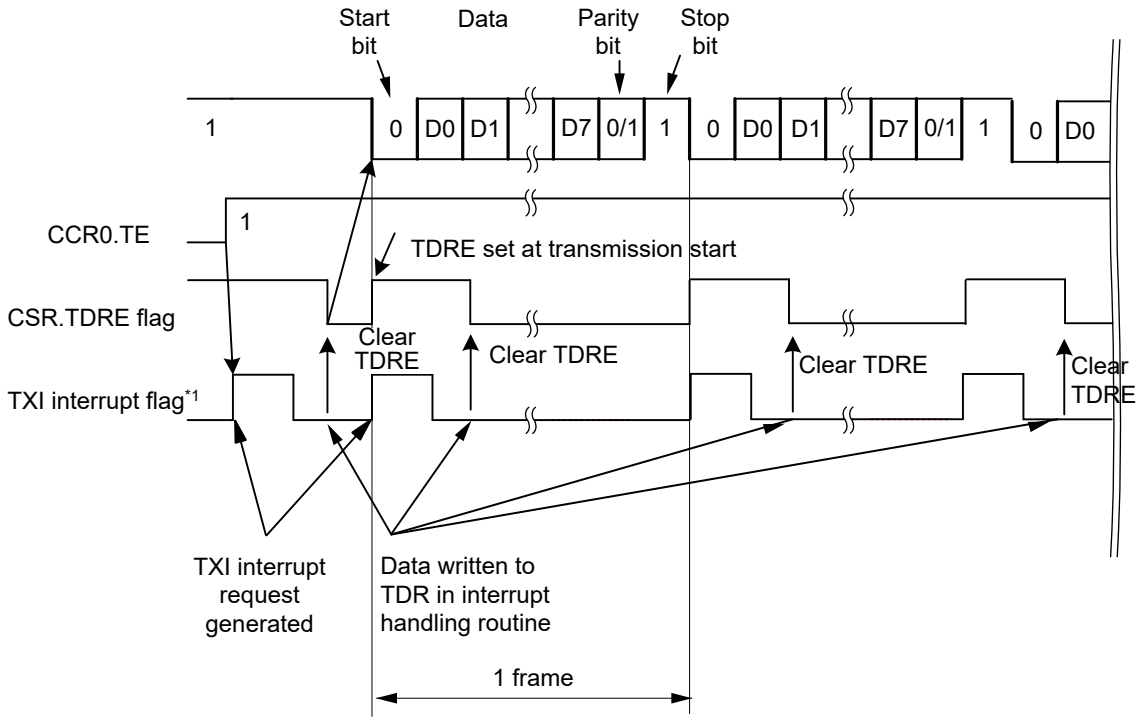
Note 1. See section 11, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 31.12 Example of operation for serial transmission in asynchronous mode (2) (with 8-bit data, parity, 1 stop bit, CTS function used, at the beginning of transmission)



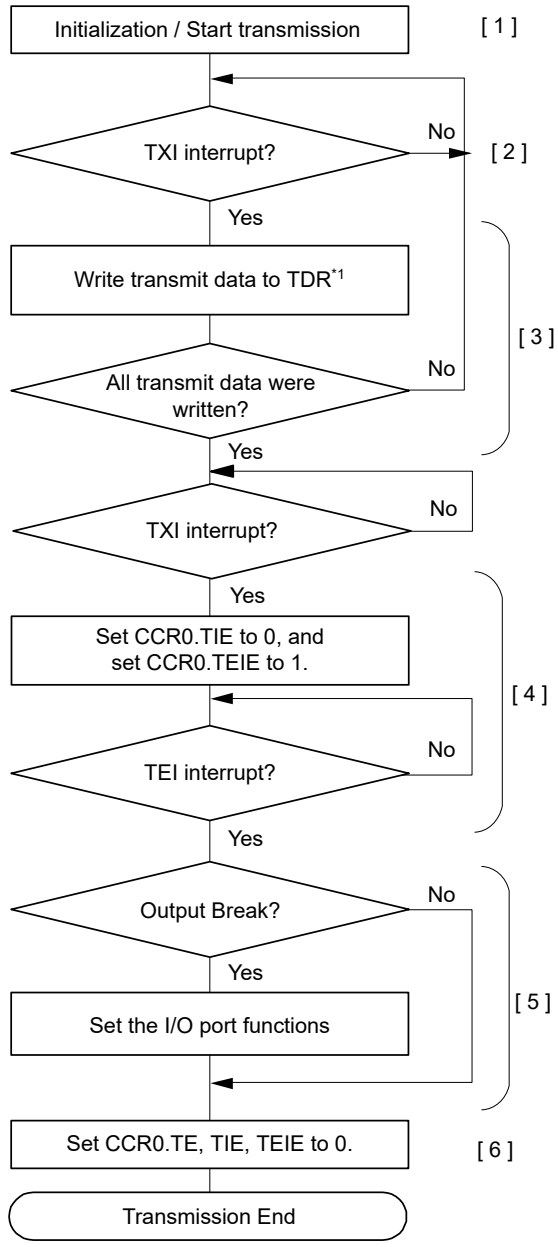
Note 1. See section 11, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 31.13 Example of operation for serial transmission in asynchronous mode (3) (with 8-bit data, parity, 1 stop bit, CTS function not used, from the middle of transmission until transmission completion)



Note 1. See section 11, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 31.14 Example of operation for serial transmission in asynchronous mode (4) (with 8-bit data, parity, 1 stop bit, CTS function not used, in the middle of transmission)



[1] Initialize SCI:
To enable TXI interrupt at the start of transmission, set TE and TIE to 1 with one instruction at the same time.

[2] TXI interrupt request:
The transmit-data-empty-interrupt (TXI) is generated at following conditions.
When set TE and TIE to 1 at same time.
When transmission-data forwards to TSR from TDR.

[3] Serial transmission continuation procedure:
To continue serial transmission write transmit data to TDR*1 once by using a TXI.
Transmit data can also be written by activating the DMAC.

[4] If you use TEI, set TIE = 0 and TEIE = 1 after writing the last-transmit-data to TDR*1 and transmission is started.

[5] Break output at the end of serial transmission:
To output a break, it is necessary to set SPB2IO and SPB2DT, for output status setting when TE is 0.
(If you set these bits at initialization, you can skip this step.)

[6] Set CCR0.TE, TIE, TEIE to 0.

Note 1. In case of byte access with data length 9 bits, write transmit data in order of TDR[15:8] and TDR[7:0].

Figure 31.15 Example of serial transmission flowchart in asynchronous mode (non-FIFO selected)

(2) FIFO selected

Figure 31.16 shows an example of data format that is written to transmit FIFO (TDR) in asynchronous mode with FIFO selected. MPBT write to transmit FIFO (TDR) bit 9. Data is set to TDR.TDAT[8:0] corresponded to data length. It should write to 0 for unused bits. It should write it in order of the TDR[15:8] register and the TDR[7:0] register at byte-access.

Data Length	Register setting		Transmit data in TDR[15:0]															
	CCR3. CHR[1:0]		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	–	–	–	–	–	–	MPBT	–	–	TDAT[6:0]						
8 bits	1	0	–	–	–	–	–	–	MPBT	–	TDAT[7:0]							
9 bits	0	x	–	–	–	–	–	–	MPBT	TDAT[8:0]								

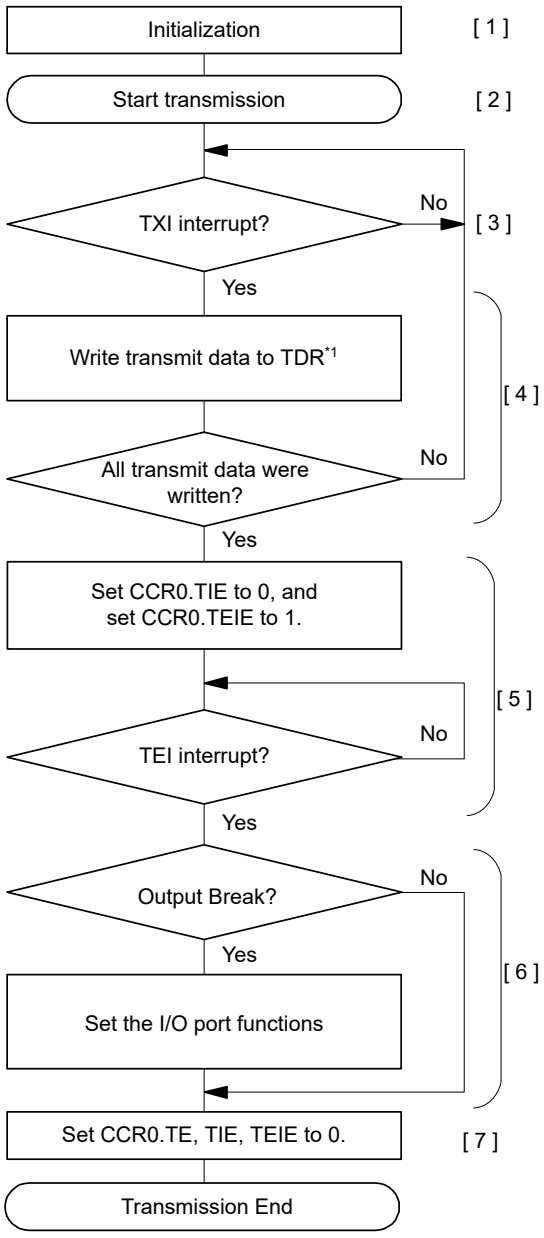
Note: –: Do not used. It should write to 0

Figure 31.16 Data format that is written to transmit FIFO (TDR) (FIFO selected)

In serial transmission, the SCI operates as described below.

1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt handling routine. The writable transmit data number is until (N_{FIFO} (= 16) - FTSR.T) bytes. At the beginning of transmission, set 1 to CCR0.TE and CCR0.TIE simultaneously. Then the TXI interrupt request is generated.
2. Transmission starts at data transfer from TDR to TSR when CCR1.CTSE = 0 (CTS function is disabled) or when the CTS# pin level is low. When the quantity of transmit data written in transmit FIFO (TDR) is equal to or less than the specified transmit triggering number, CSR.TDRE is set to 1. If CCR0.TIE is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next transmit data to TDR in the TXI interrupt handling routine before transmission of the current transmit data is completed. Write the last transmission data, then the last data transmit start, and TXI is generated. When TEI interrupt requests are in use, before the last data transmit end, set the CCR0.TIE bit to 0 (a TXI interrupt request is disabled) and the CCR0.TEIE bit to 1 (a TEI interrupt request is enabled) using the TXI handling routine.
3. Data is sent from the TXDn pin in the following order: start bit, transmit data, parity bit or multi-processor bit (may be omitted depending on the format), and stop bit.
4. The SCI checks whether non-transmitted data in transmit FIFO (TDR) or not at the time of stop bit output.
5. When data is set to transmit FIFO (TDR), setting of CCR1.CTSE to 0 (CTS function is disabled) or a low level input on the CTSn# pin cause the next transfer of the next transmit data from transmit FIFO (TDR) to TSR and sending of the stop bit, after which serial transmission of the next frame starts.
6. If data is not set to transmit FIFO (TDR), CSR.TEND is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output. If CCR0.TEIE is 1 at this time, a TEI interrupt request is generated.

Figure 31.17 shows a sample flowchart for serial transmission in asynchronous mode at FIFO selected.



[1] Initialize SCI:
To enable TXI interrupt, set TE and TIE to 1 with one instruction at the same time.

[2] Set transmit-data to the transmit FIFO (TDR):
The number of transmittable data that can be set is 16.

[3] TXI interrupt request:
The transmit-data-empty-interrupt (TXI) is generated when transmission-data forwards to TSR from the transmit FIFO (TDR) and the number of transmit-data written in the transmit FIFO is equal to or less than the transmit FIFO threshold.

[4] Serial transmission continuation procedure:
To continue serial transmission, write all transmit data to the transmit FIFO (TDR) by using a TXI interrupt and clear TDRE flag to 0. (Transmit data can also be written to the transmit FIFO (TDR) by activating the DMAC. In this case, the TDRE flag is cleared automatically, so please do not operate the TDRE flag.)
The number of transmission data it's possible to write in is N_{FIFO} (= 16) - (the number of stored transmit FIFO data).

[5] If you use TEI, set TIE = 0 and TEIE = 1 after writing the last-transmit-data to TDR.*1

[6] Break output at the end of serial transmission:
To output a break, it is necessary to set SPB2IO and SPB2DT, for output status setting when TE is 0.
(If you set these bits at initialization, you can skip this step.)

[7] Set CCR0.TE, TIE, TEIE to 0.

Note 1. In case of byte access with data length 9 bits, write transmit data in order of TDR[15:8] and TDR[7:0].

Figure 31.17 Example of serial transmission flowchart in asynchronous mode (FIFO selected)

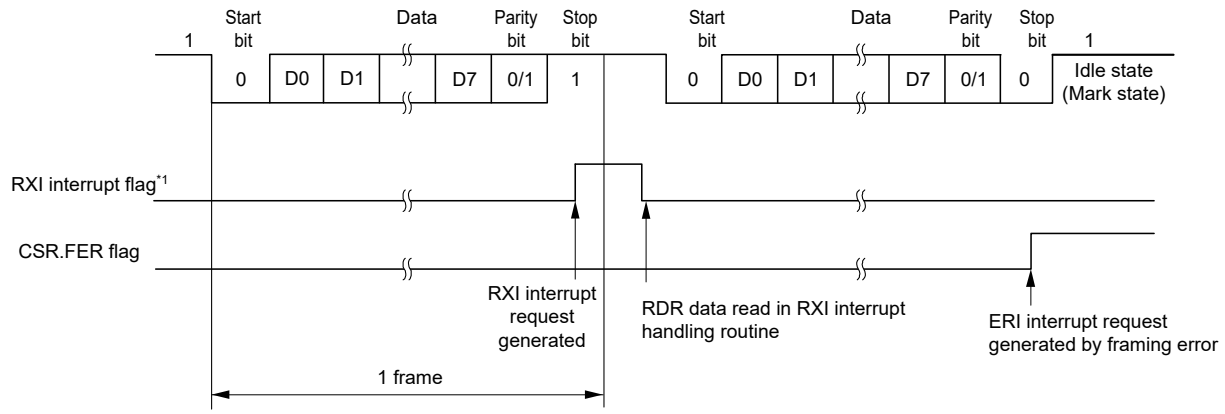
31.3.9 Serial Data Reception (Asynchronous Mode)

(1) Non-FIFO selected

Figure 31.18 and Figure 31.19 show an example of the operation for serial data reception in asynchronous mode. In serial data reception, the SCI operates as described below.

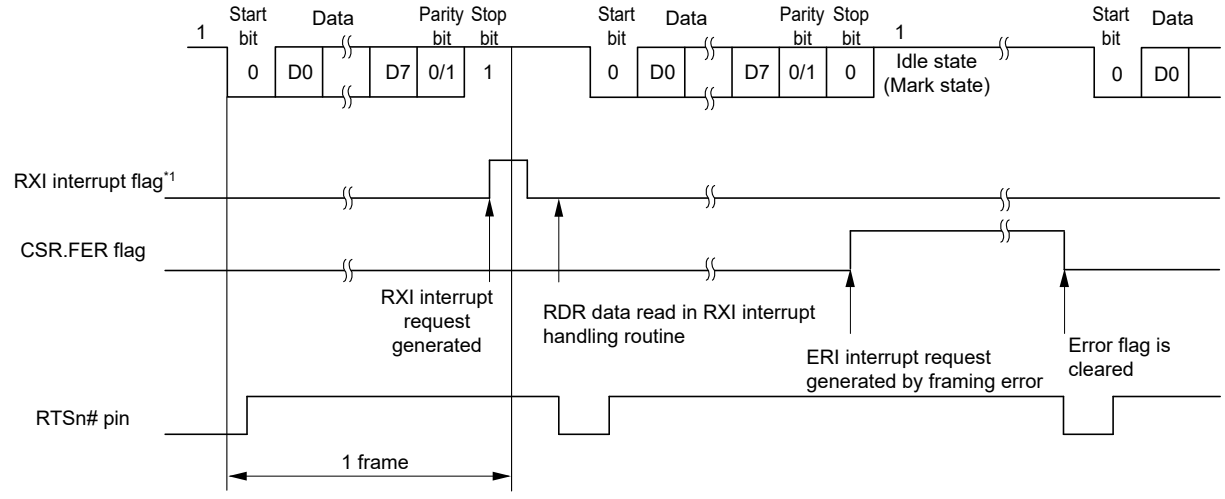
1. When the value of CCR0.RE becomes 1, the output signal on the RTSn# pin goes to the low level in the case of RTS function use.
2. When the SCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in RSR, and checks the parity bit and stop bit.
3. If an overrun error occurs, CSR.ORER is set to 1. If CCR0.RIE is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR.

4. If a parity error is detected, CSR.PER is set to 1 and receive data is transferred to RDR. If the CCR0.RIE is 1 at this time, an ERI interrupt request is generated.
5. If a framing error (when the stop bit is 0) is detected, CSR.FER is set to 1 and receive data is transferred to RDR. If CCR0.RIE is 1 at this time, an ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to RDR. If CCR0.RIE is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to RDR in this RXI interrupt handling routine before reception of the next receive data is completed. Reading the received data that have been transferred to RDR causes the RTSn# pin to output the low level in the case of RTS function use. If you do not want to turn the RTSn# pin output low after receiving the last data, set CCR0.RE bit to 0, before reading the RDR.



Note 1. See section 11, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 31.18 Example of SCI operation for serial reception in asynchronous mode (1) (example with 8-bit data, parity, 1 stop bit, RTS function is not used)



Note 1. See section 11, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

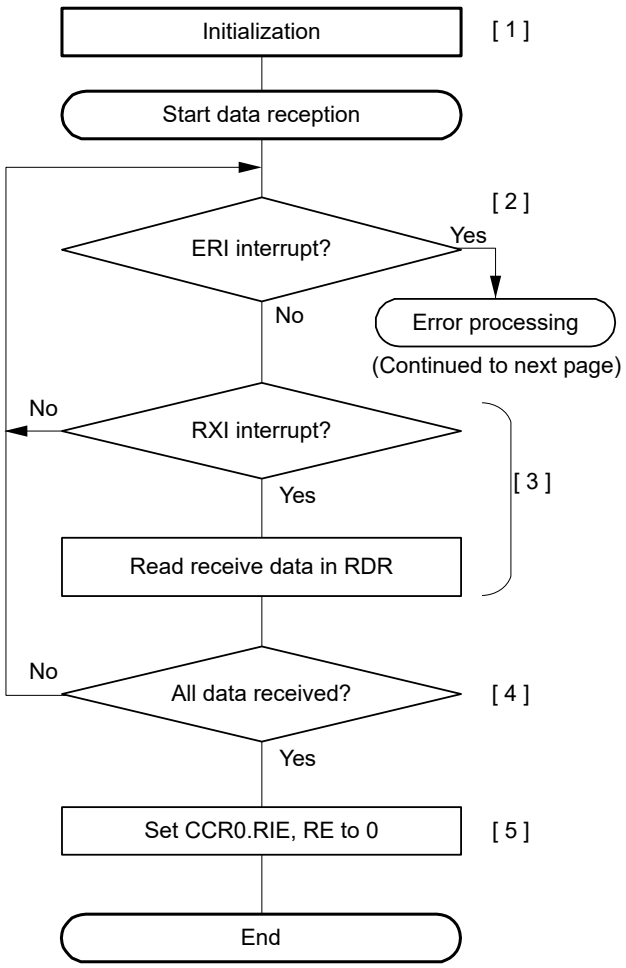
Figure 31.19 Example of SCI operation for serial reception in asynchronous mode (2) (example with 8-bit data, parity, 1 stop bit, RTS function is used)

Table 31.23 lists the states of the flags in CSR status register and receive data handling when a receive error is detected. If a receive error is detected, an ERI interrupt request is generated but an RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, and PER bits to 0 before resuming reception. Moreover, be sure to read the RDR during overrun error processing. When a reception is forcibly terminated by

setting CCR0.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in RDR. Figure 31.20 and Figure 31.21 show samples of flowcharts for serial data reception.

Table 31.23 Flags in the CSR status register and receive data handling

Flags in the CSR status register			Receive data	Receive error type
ORER	FER	PER		
1	0	0	Lost	Overrun error
0	1	0	Transferred RDR	Framing error
0	0	1	Transferred RDR	Parity error
1	1	0	Lost	Overrun error + Framing error
1	0	1	Lost	Overrun error + Parity error
0	1	1	Transferred RDR	Framing error + Parity error
1	1	1	Lost	Overrun error + Framing error + Parity error



[1] Initialize SCI.
(Set CCR0.RE, RIE to 1.)

[2] Receive error processing and break detection:
If a receive error occurs, an ERI interrupt is generated. After performing the appropriate error processing, be sure to set the ORER, PER, and FER flags to 0. Reception cannot be resumed if any of these flags is set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RXDn pin.

[3] Read the receive data in RDR once in the RXI interrupt handling routine.

[4] Serial reception continuation procedure:
To continue serial reception, before the stop bit of the current frame is received, read data from RDR in the RXI interrupt handling routine. The RDR data can also be read by activating the DMAC.

[5] Set CCR0.RIE, RE to 0.

Figure 31.20 Example flowchart of serial reception in asynchronous mode (non-FIFO selected) (1)

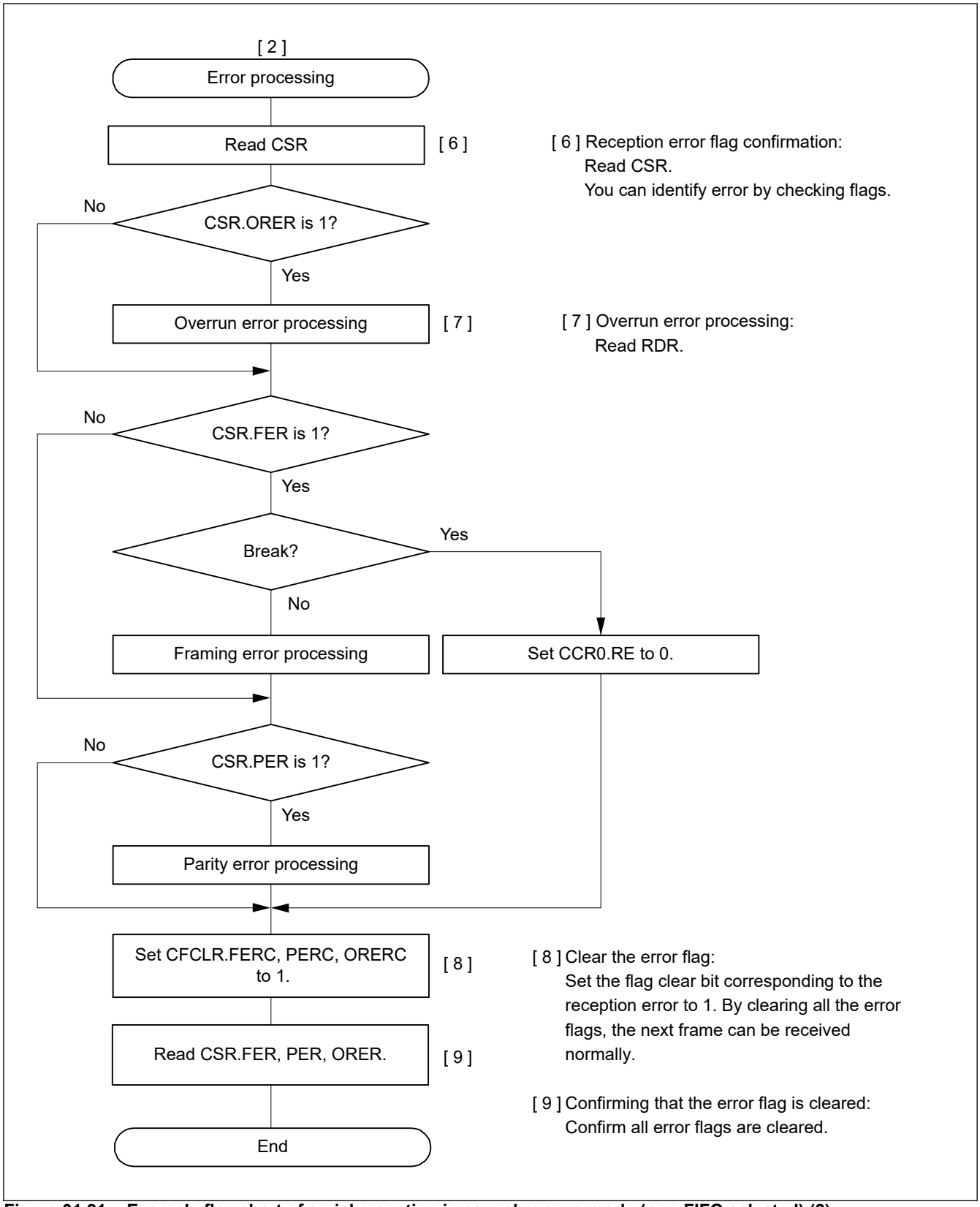


Figure 31.21 Example flowchart of serial reception in asynchronous mode (non-FIFO selected) (2)

(2) FIFO selected

Figure 31.22 shows an example of data format that is stored to receive FIFO (RDR) in asynchronous mode.

MPB (Receive FIFO (RDR) bit 9) is stored 0. Data is stored to receive FIFO (RDR) corresponded to data length. It is stored 0 for unused bits. If reading receive FIFO (RDR), SCI updates to next data which are FER, PER and receive data

(RDAT[8:0]) in receive FIFO. The flags which are BRK, RDRF, ORER and DR in receive FIFO, are always indicated to the flags corresponded to CSR register.

Data Length	Register Setting		Receive flag in RDR[31:0], MPB, RDAT[8:0]															
	CCR3.CHR[1:0]		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	–	–	–	FFER	FPER	DR	MPB	0	0	RDAT[6:0]						
8 bits	1	0	–	–	–	FFER	FPER	DR	MPB	0	RDAT[7:0]							
9 bits	0	x	–	–	–	FFER	FPER	DR	MPB	RDAT[8:0]								
	CCR3.CHR[1:0]		b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
7 bits	1	1	–	–	–	FER	PER	–	–	ORER	–	–	–	–	–	–	–	–
8 bits	1	0	–	–	–	FER	PER	–	–	ORER	–	–	–	–	–	–	–	–
9 bits	0	x	–	–	–	FER	PER	–	–	ORER	–	–	–	–	–	–	–	–

Note: The MPB flag, the RDR[9] bit, is always read as 0.
When a 7-bit data length is selected, the RDAT[8:7] bits are read as 00b.
When an 8-bit data length is selected, the RDAT[8] bit is read as 0.

Figure 31.22 Data format that is stored to receive FIFO (RDR) (FIFO selected)

Table 31.24 lists the states of the flags in CSR status register and receive data handling when a receive error is detected with FIFO selected. Figure 31.23 and Figure 31.24 show samples of flowcharts for serial data reception with FIFO selected.

In serial data reception, the SCI operates as described below.

1. When the value of CCR0.RE becomes 1, the output signal on the RTSn# pin goes to the low level in the case of RTS function use.
2. When the SCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in RSR, and checks the parity bit and stop bit.
3. If there is no space in RDR, CSR.ORER is set to 1. If CCR0.RIE is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to receive FIFO (RDR).
4. If a parity error is detected, CSR.PER is set to 1 and receive data is transferred to receive FIFO (RDR). If the CCR0.RIE is 1 at this time, an ERI interrupt request is generated.
5. If a framing error (when the stop bit is 0) is detected, CSR.FER is set to 1 and receive data is transferred to RDR. If CCR0.RIE is 1 at this time, an ERI interrupt request is generated.
6. After framing error is detected, when SCI detects that continuous receive data is 0 for 1 frame, reception stops.
7. When quantity of data stored in the receive FIFO data register (RDR) falls the specified receive triggering number, and that no next data has been received yet after the elapse of 15 etu from the last stop bit in asynchronous mode, FRSR.DR is set to 1. If RIE bit is set to 1, SCI occurs RXI interrupt request when FCR.DRES bit is 0 and SCI occurs ERI interrupt request when FCR.DRES = 1.
8. When reception finishes successfully, receive data is transferred to receive FIFO (RDR). RDRF is set to 1 when the quantity of receive data which is equal to or greater than the specified receive triggering number are stored in receive FIFO (RDR). If CCR0.RIE is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to receive FIFO (RDR) in this RXI interrupt handling routine before overrun error is occurred. Reading the received data that have been transferred to receive FIFO (RDR), and if it is less than RTS trigger number, causes the RTSn# pin to output the low level in the case of RTS function use.

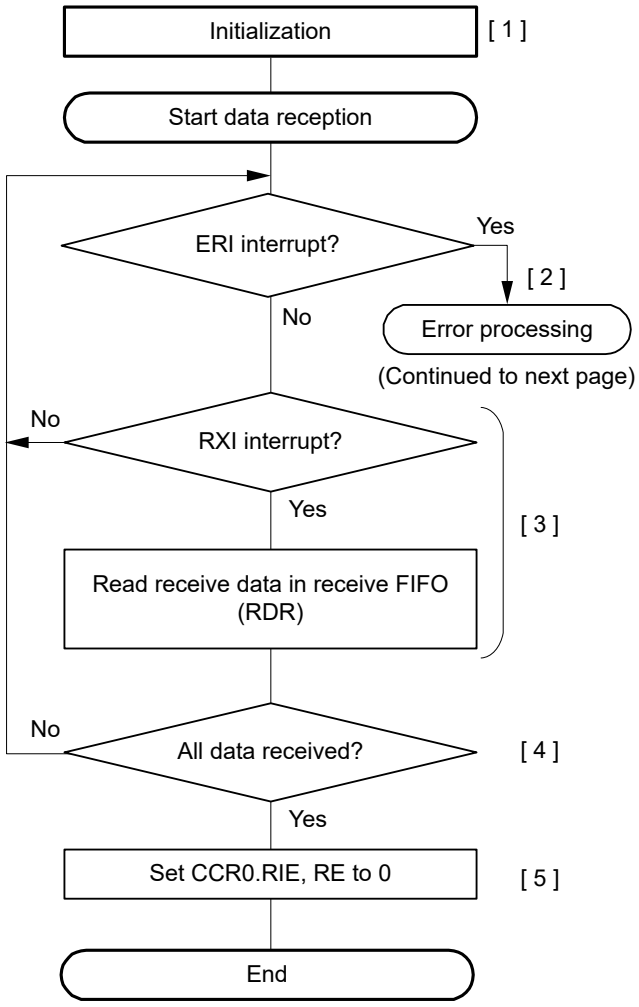
Table 31.24 Flags in the CSR status register and receive data handling (FIFO selected) (1 of 2)

CSR value			Receive FIFO (RDR)	
ORER	FER*1	PER*1	RDAT[8:0]	Receive Error Type
1	0	0	Lost	Overrun error
0	1	0	Transferred RDR	Framing error
0	0	1	Transferred RDR	Parity error

Table 31.24 Flags in the CSR status register and receive data handling (FIFO selected) (2 of 2)

CSR value			Receive FIFO (RDR)	
ORER	FER*1	PER*1	RDAT[8:0]	Receive Error Type
1	1	0	Lost	Overrun error + Framing error
1	0	1	Lost	Overrun error + Parity error
0	1	1	Transferred RDR	Framing error + Parity error
1	1	1	Lost	Overrun error + Framing error + Parity error

Note 1. This flag indicates whether there is an error in received data when reception is completed.



[1] Initialize SCI.
(Set CCR0.RE, RIE to 1.)

[2] Receive error processing and break detection:
If a receive error occurs, an ERI interrupt is generated. After performing the appropriate error processing, be sure to set the ORER, flags to 0. Reception cannot be resumed if ORER is set to 1. Reception can continue even if PER, FER, DR flags are 1.

[3] Read the receive data in the receive FIFO (RDR) in the RXI interrupt handling routine. The reception data stored in the receive FIFO (RDR) is read until the number of stored data will be below the FCR.RTRG value at least. You can confirm the number of the reception data in the FIFO by FRSR.R[5:0].

[4] Serial reception continuation procedure:
To continue serial reception, before an overrun error occurs, please read data from the receive FIFO (RDR) in the RXI interrupt handling routine and clear CSR.RDRF flag and FRSR.DR flag to 0. The receive FIFO data can also be read by activating the DMA. The RDRF flag is cleared automatically in this case, so please do not operate the RDRF flag.

[5] Set CCR0.RIE, RE to 0.

Figure 31.23 Example flowchart of serial reception in asynchronous mode (FIFO selected) (1)

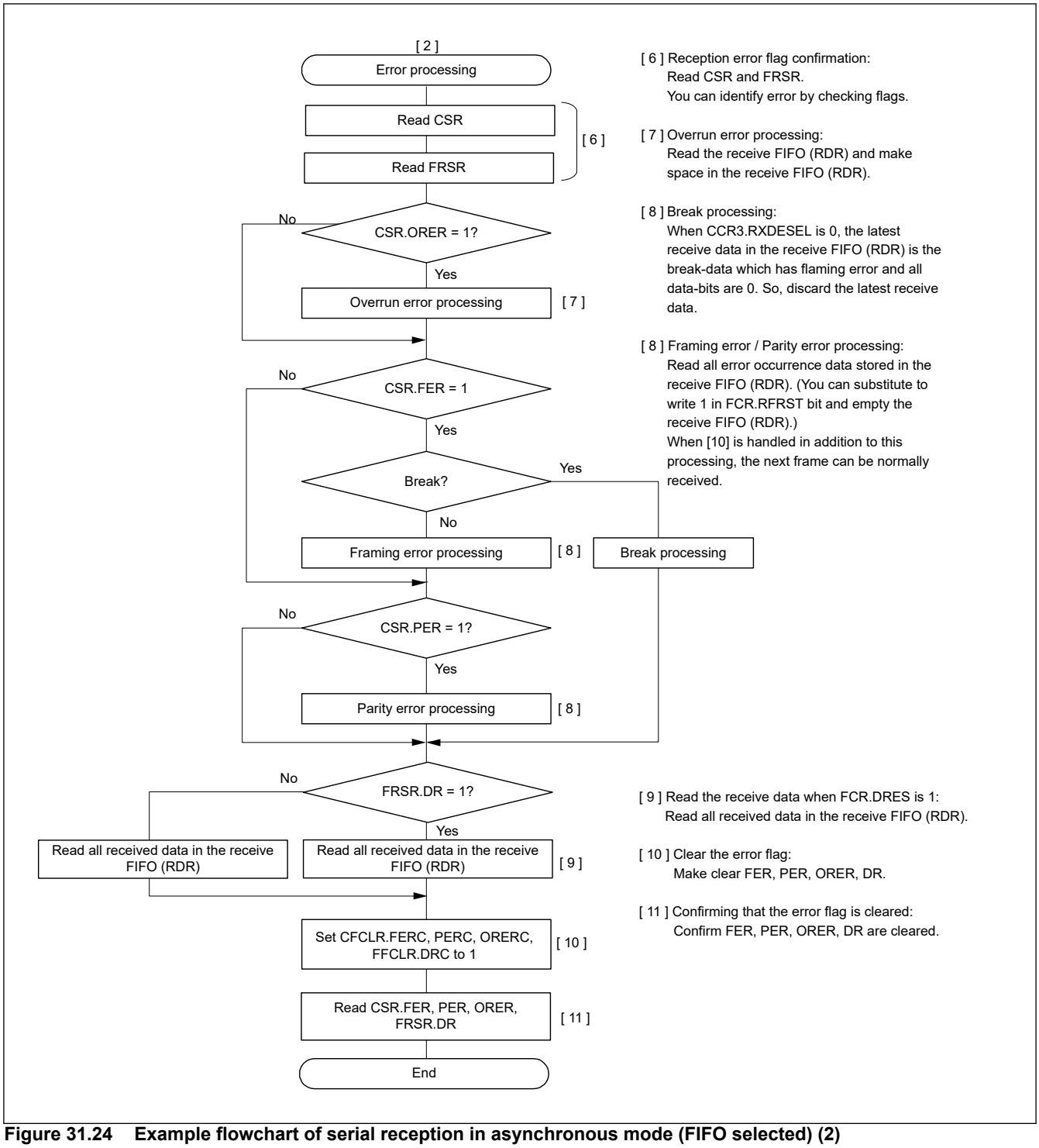


Figure 31.24 Example flowchart of serial reception in asynchronous mode (FIFO selected) (2)

31.3.10 The function of adjust receive sampling timing (Asynchronous Mode)

When there is the difference between the rising transfer time and the falling transfer time through a photo coupler, the receive sampling timing at middle of bit affects the reception margin. In this case, the receive sampling timing can adjust from the middle of bit to the optimum timing by using this function.

The receive sampling timing is adjusted from the middle of bit by following formula. And the adjustable direction is set by CCR4.AJD. You can select forward or backward from the middle of bit. When adjusting backward (CCR4.AJD = 0), substitute AJD = +1 and substitute AJD = -1 when adjusting forward (CCR4.AJD = 1).

Adjusted sampling timing = the middle of bit + AJD × (base clock × the setting value of CCR4.AST[2:0])

The setting timing is limited by base clock cycles per 1 bit. Please see Table 31.25 in detail.

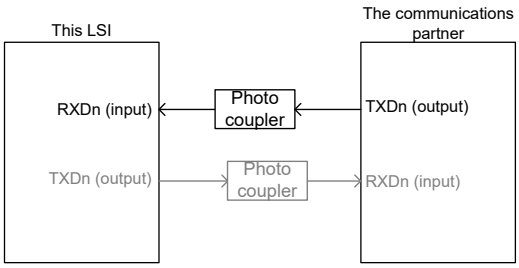
A reception movement image figure of the communication through a photo coupler with this function is shown in Figure 31.25 and Figure 31.26, the movement explanation of this function is shown in Figure 31.27.

Do not use this function when there is no difference between the rising transfer time and the falling transfer time, because there is a possibility of deteriorating the reception margin.

Table 31.25 The acceptable value of setting register (asynchronous mode using internal clock)

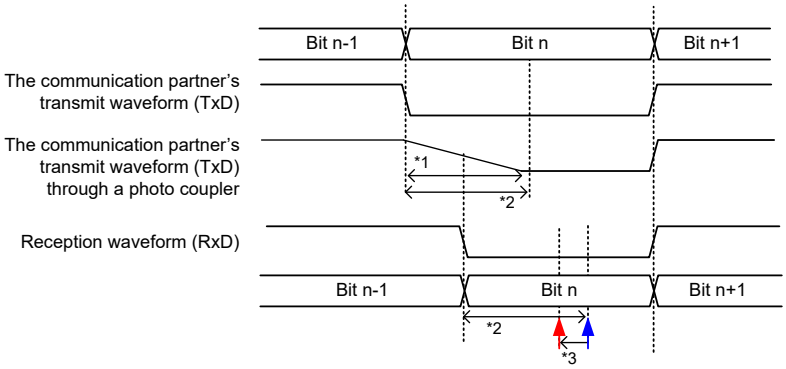
CCR2.ABCSE	CCR2.ABCS	The number of base cycles / 1 bit	The acceptable value	
			CCR4.AJD	CCR4.AST
1	x	6	0	000b to 010b*1
			1	
0	1	8	0	000b to 011b*1
			1	
0	0	16	0	000b to 111b
			1	

Note 1. When the value of CCR4.AST is over the acceptable value, sampling is done at default timing. (Adjustment of sampling is not done.)



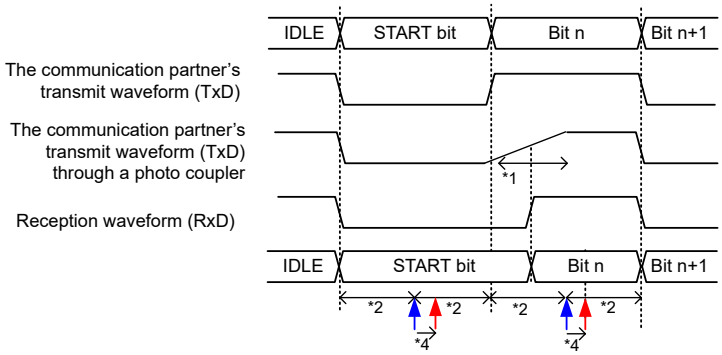
(a) In the case of the falling transfer time >> rising transfer time

The falling edge of reception waveform is made dull like following chart.
In this case, you can sampling at the middle of bit if you adjust the receive sampling timing to forward (AJD = 1b).



(b) In the case of the falling transfer time << rising transfer time

The rising edge of reception waveform is made dull like following chart.
So, the reception margin of communications partner will be bad. In this case, you can improve the reception margin if you adjust the receive sampling timing to back.



The receive sampling timing when unadjusted (middle of bit)

The adjusted receive sampling timing

Note 1. The dull period by a photo coupler

Note 2. Bit center timing at set communication rate

Note 3. When CCR4.AJD is 1, the receive sampling timing is shifted to forward by the setting value of CCR4.AST[2:0].

Note 4. When CCR4.AJD is 0, the Receive sampling timing is shifted to backward by the setting value of CCR4.AST[2:0].

Note: This waveform shows the operation image of adjustment receive sampling timing.

Figure 31.25 The reception movement image figure of the communication through a photo coupler

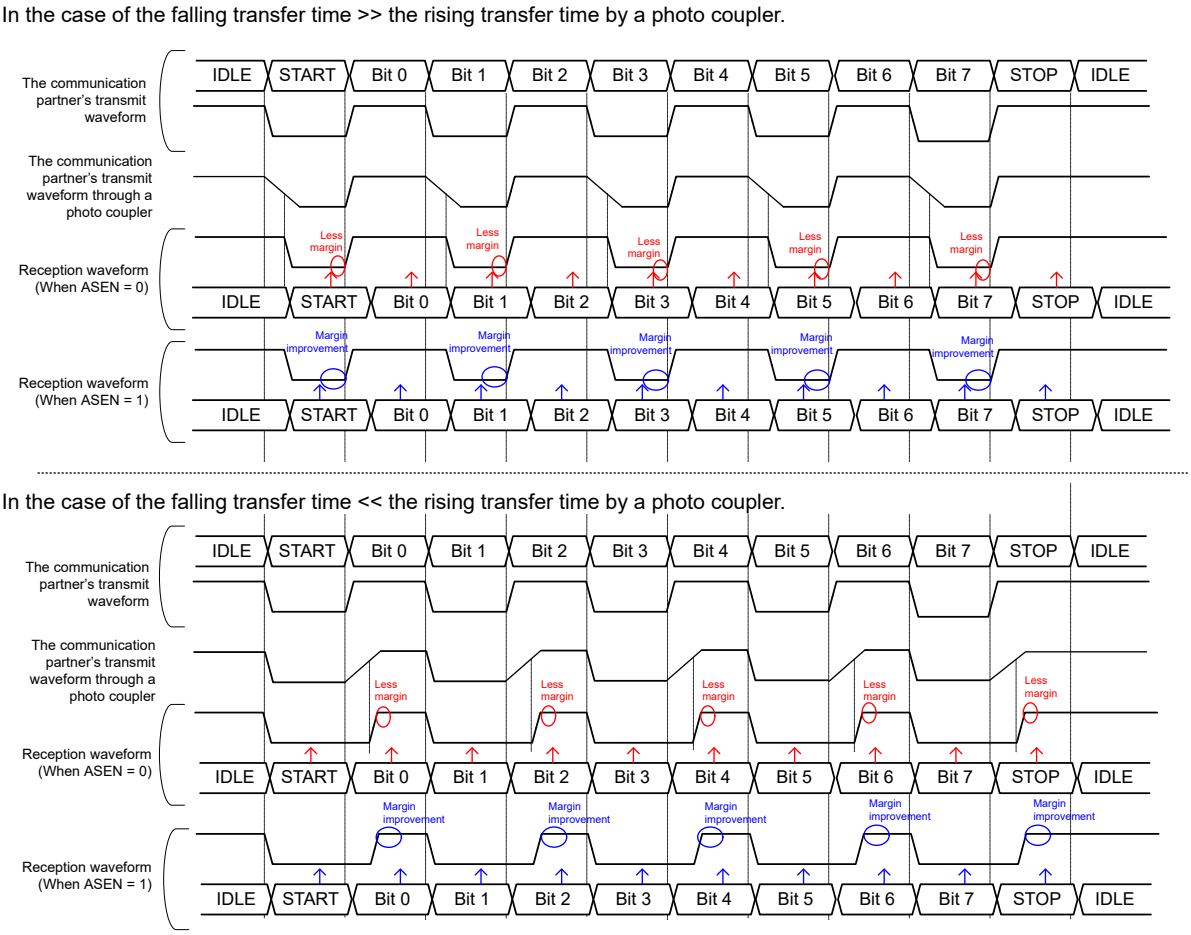


Figure 31.26 Improvement of reception margin when receive sampling timing adjustment function Image

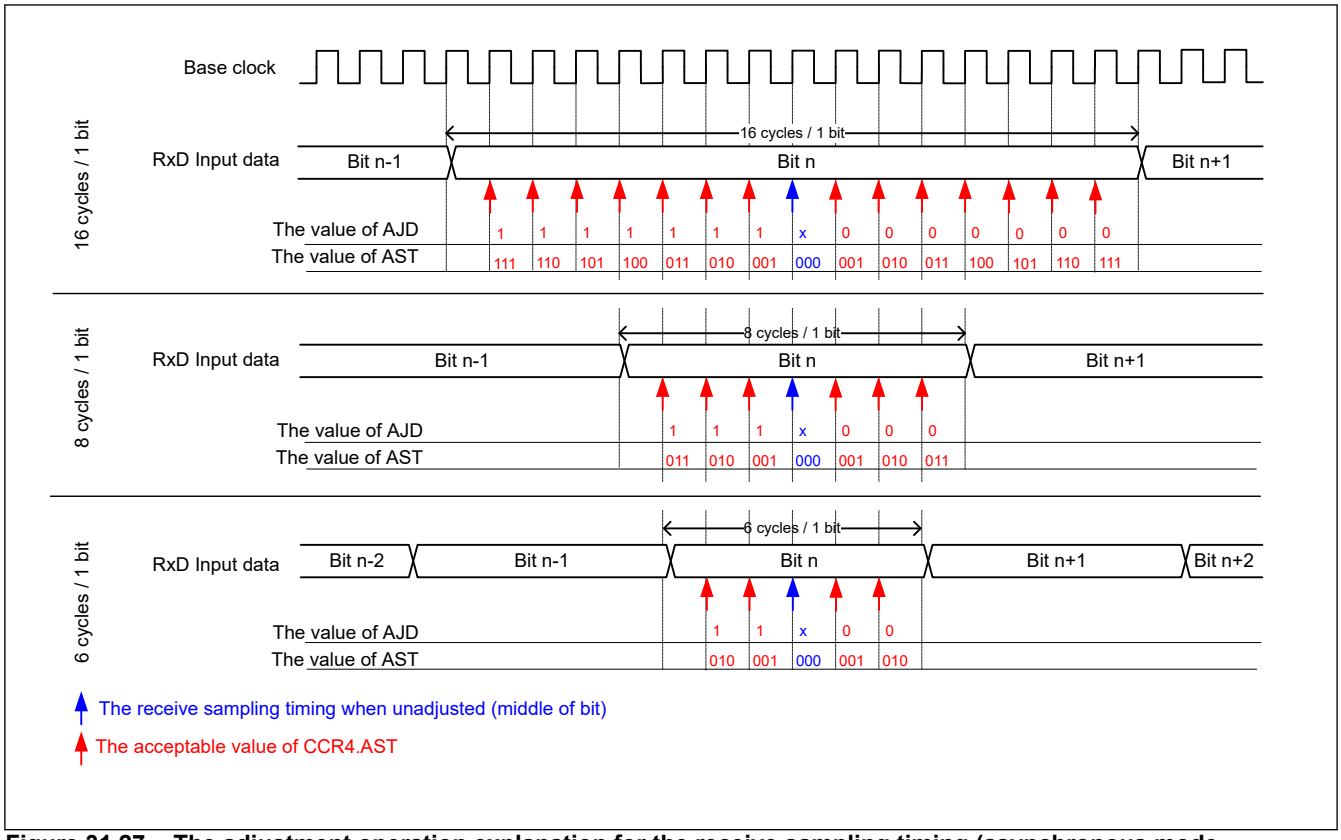


Figure 31.27 The adjustment operation explanation for the receive sampling timing (asynchronous mode using internal clock)

31.3.11 The function of adjust transmit timing (Asynchronous Mode)

In communication via a photo coupler or the like, when either the rising or falling transition time of the TxD output signal is long, then a communication partner receive dulled waveform. In this case, the reception margin may be affected.

In these cases, make a communication partner to be sampling at middle of bit using the function of adjust transmit timing. When CCR4.ATEN is 1, this function can adjust the edge timing at the timing calculated by the following formula for the edge set with CCR4.AET.

The adjustment edge timing = base clock × CCR4.ATT[2:0]

In addition, the upper limit of the adjustment edge timing is limited by setting the base clock cycles. For details, see Table 31.26.

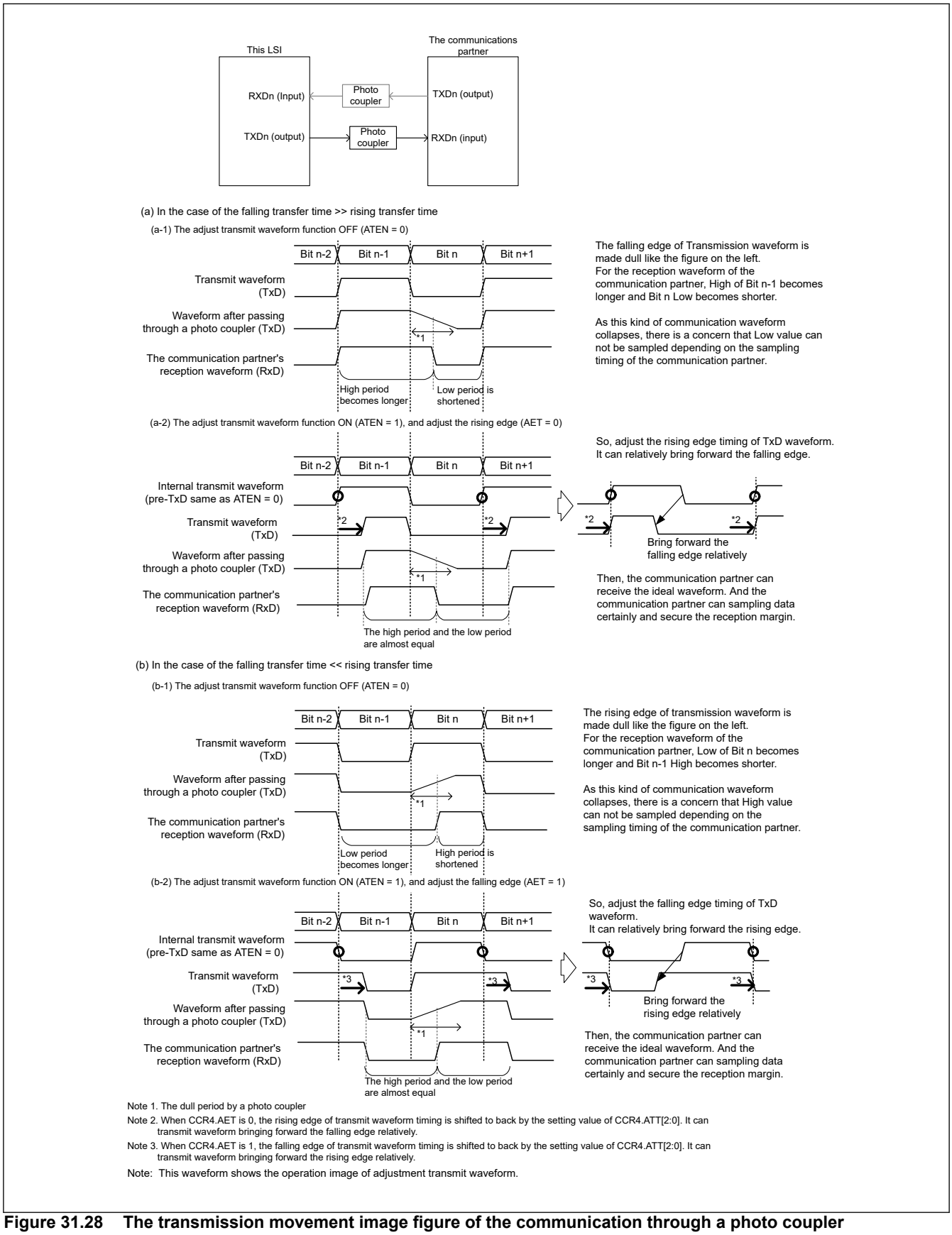
A transmission movement image figure of the communication through a photo coupler with this function is shown in Figure 31.28 and Figure 31.29, the movement explanation of this function is shown in Figure 31.30 and Figure 31.31.

Do not use this function when there is not the difference between the rising transfer time and the falling transfer time, there is a possibility of deteriorating the reception margin of a communication partner.

Table 31.26 The acceptable value of CCR4.AET and CCR4.ATT (asynchronous mode using internal clock)

ABCSE	ABCS	The number of base clock cycles / 1 bit	The acceptable value	
			AET	ATT[2:0]
1	x	6	0	000b to 101b
			1	
0	1	8	0	000b to 111b
			1	
0	0	16	0	000b to 111b
			1	

Note: When the value of CCR4.AET/ATT is out of the acceptable value, this module does not adjust transmit timing.



Note 1. The dull period by a photo coupler

Note 2. When CCR4.AET is 0, the rising edge of transmit waveform timing is shifted to back by the setting value of CCR4.ATT[2:0]. It can transmit waveform bringing forward the falling edge relatively.

Note 3. When CCR4.AET is 1, the falling edge of transmit waveform timing is shifted to back by the setting value of CCR4.ATT[2:0]. It can transmit waveform bringing forward the rising edge relatively.

Note: This waveform shows the operation image of adjustment transmit waveform.

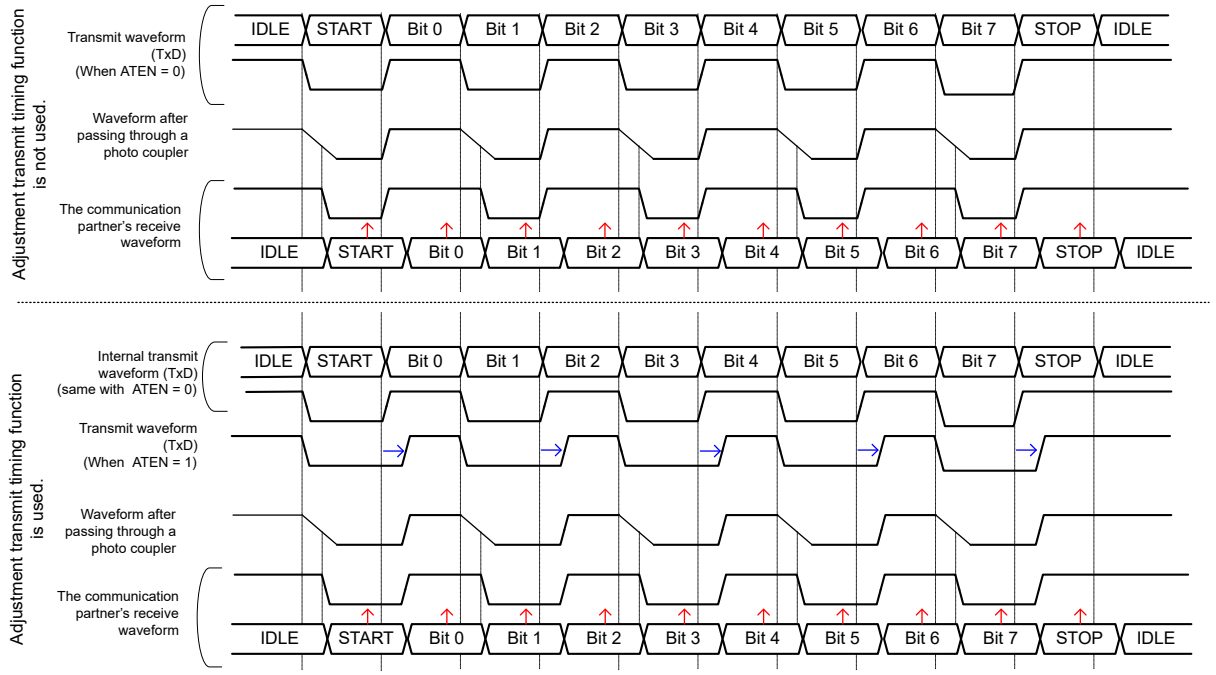
Figure 31.28 The transmission movement image figure of the communication through a photo coupler

The explanation of transmit waveforms of the communication through a photo coupler using adjust transmit timing function

When using the transmission timing adjustment function, adjust the edge timing of the transmission waveform and correct the reception waveform of the communication partner

The following example is 8 bit long data.

(a) In the case of the falling edge transfer time >> the rising transfer time



(b) In the case of the falling edge transfer time << the rising transfer time

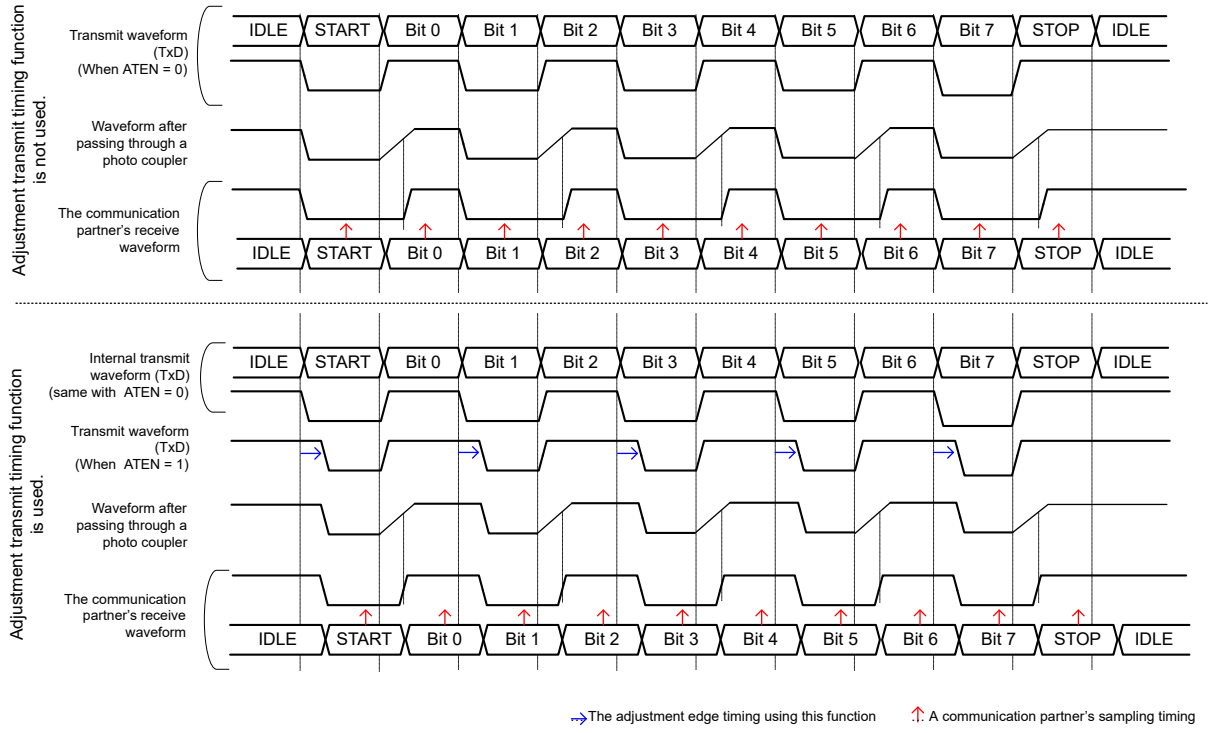
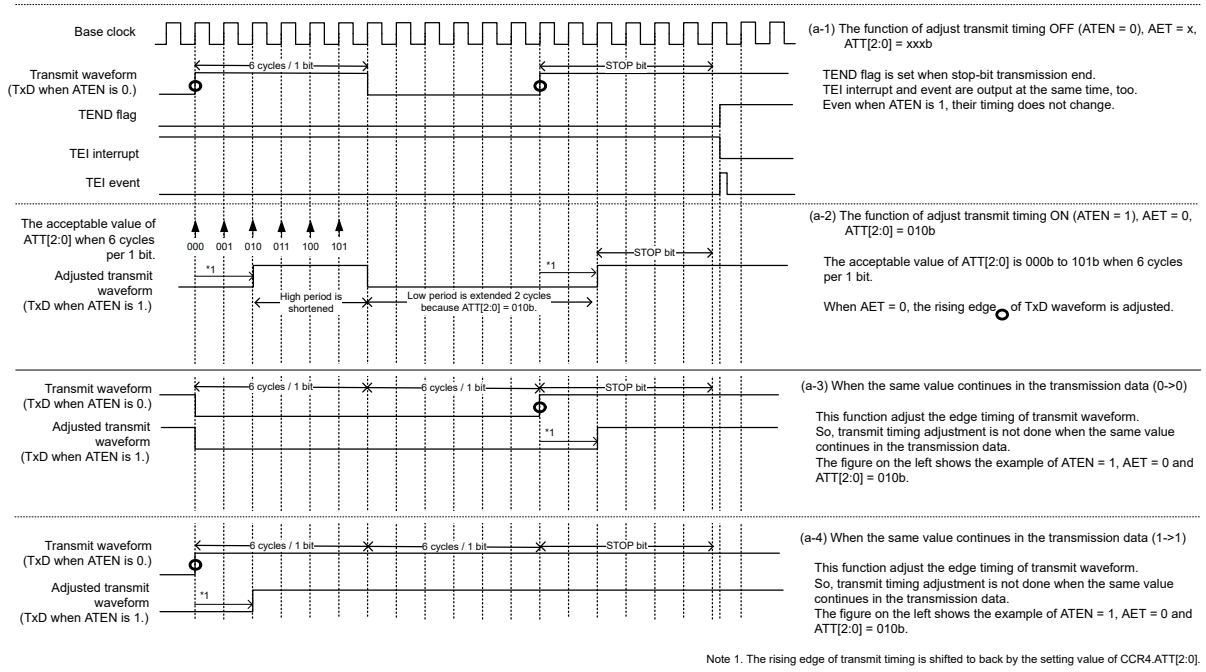


Figure 31.29 The explanation for the transmit waveform through a photo coupler

The operation explanation of adjustment the transmit timing

(a) In the case of the falling transfer time >> rising transfer time
In this case, the high period of a communication partner's reception waveform is made long, and the low period is made short.
Therefore, the LSI transmits the waveform with the falling edge relatively brought forward by adjusting the rising edge timing.
Then adjust value (ATT[2:0]) should be set to make equal the low-period/1 bit and high-period/1 bit for a communication partner.

This function's operation is explained as an example of 6 cycles/1 bit.



Note 1. The rising edge of transmit timing is shifted to back by the setting value of CCR4.ATT[2:0].

Figure 31.30 The adjustment operation explanation for the transmit timing when AET is 0

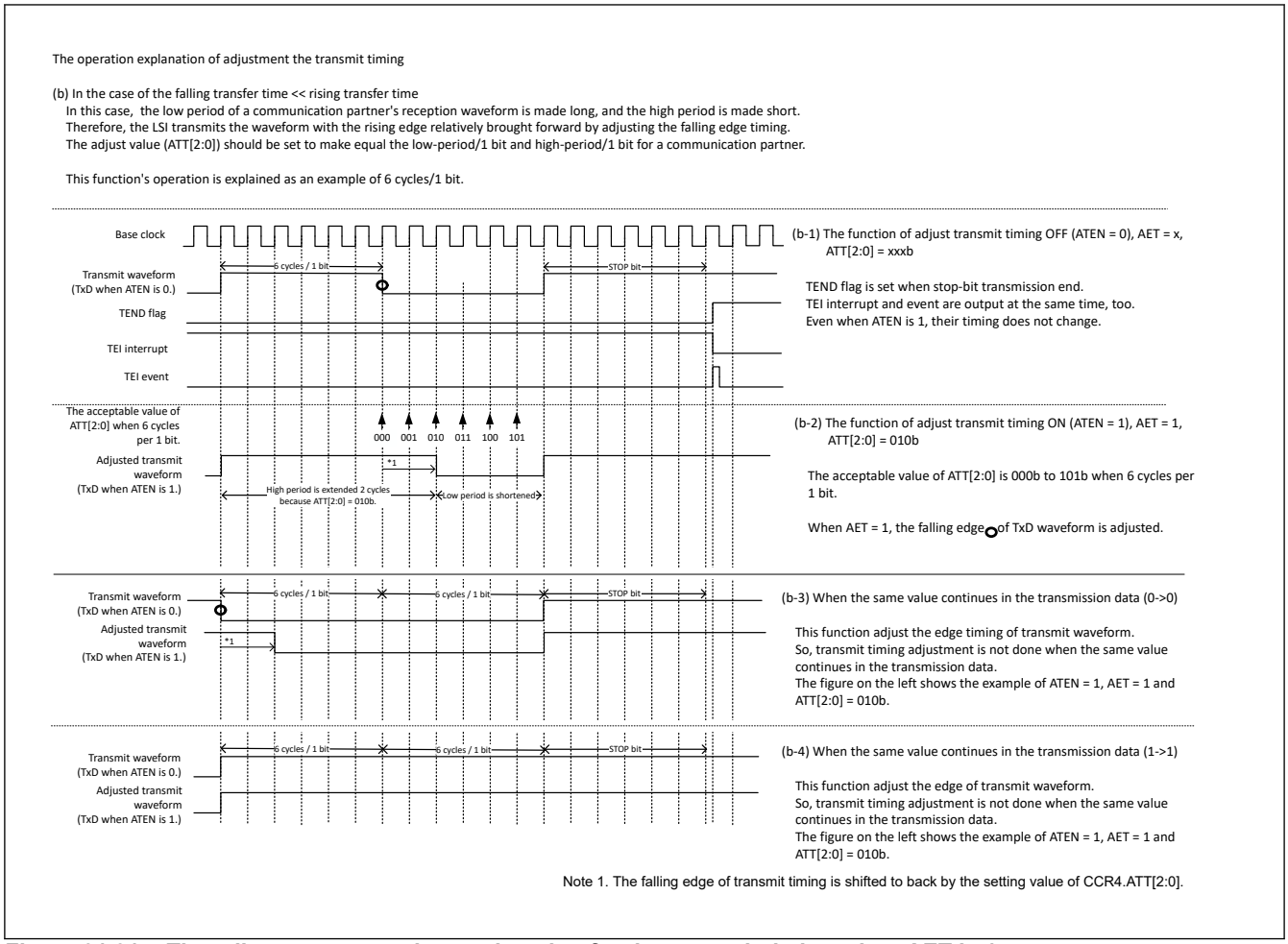


Figure 31.31 The adjustment operation explanation for the transmit timing when AET is 1

31.4 Multi-Processor Communication Function

Using the multi-processor communication functions enables to transmit and receive data by sharing a communication line between multiple processors by using asynchronous serial communication in which the multi-processor bit is added. In multi-processor communication, a unique ID code is allocated to each receiving station. Serial communication cycles consist of an ID transmission cycle to specify the receiving station and a data transmission cycle to transmit data to the specified receiving station. The multi-processor bit is used to distinguish between the ID transmission cycle and the data transmission cycle. When the multi-processor bit is set to 1, it indicates the ID transmission cycle and when the multiprocessor bit is set to 0, it indicates the data transmission cycle. Figure 31.32 shows an example of communication between processors by using a multi-processor format. First, a transmitting station transmits communication data in which the multi-processor bit set to 1 is added to the ID code of the receiving station. Next, the transmitting station transmits the communication data in which the multi-processor bit set to 0 is added to the transmit data. On receiving the communication data in which the multi-processor bit is set to 1, the receiving station compares the received ID with the ID of the receiving station itself and if the two matches, receives the communication data that is subsequently transmitted. If the received ID does not match with the ID of the receiving station, the receiving station skips the communication data until again receiving the communication data in which the multi-processor bit is set to 1.

RTS control cannot be used at the time of multi-processor communication function use, because this is a function corresponding to one-to-many communications.

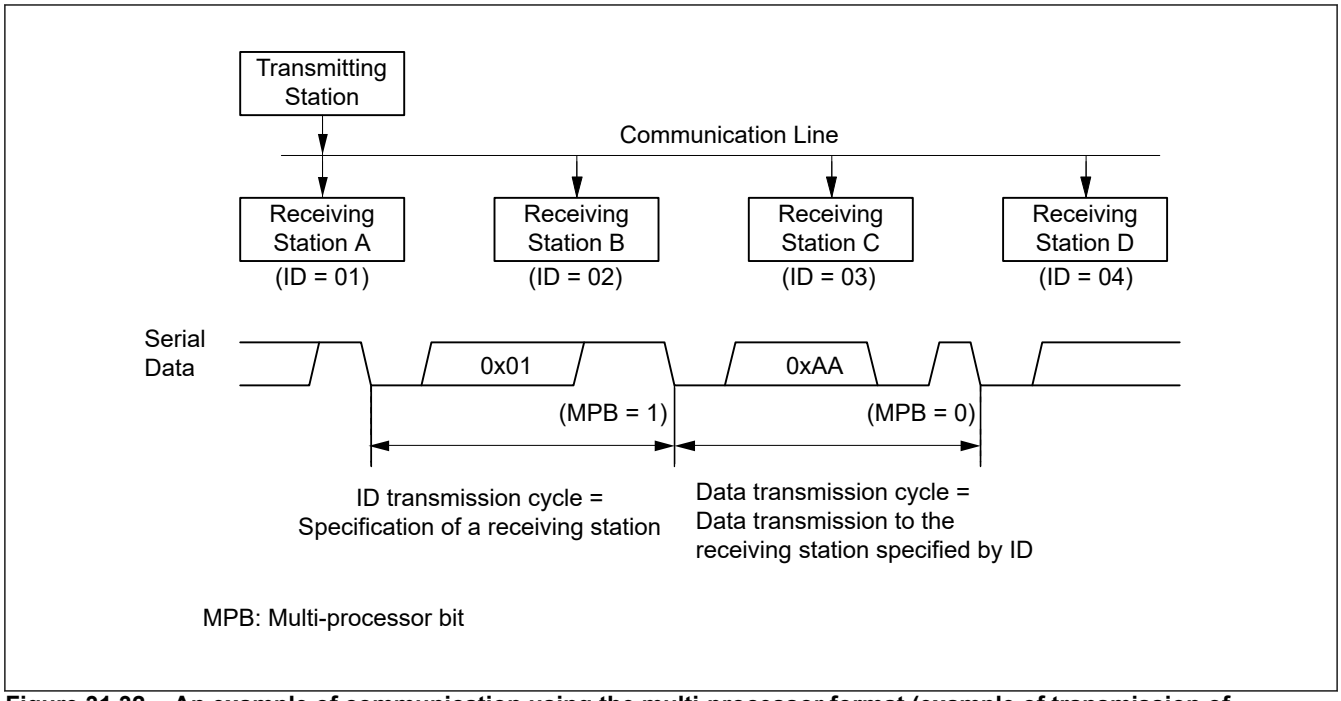


Figure 31.32 An example of communication using the multi-processor format (example of transmission of data 0xAA to receiving station A)

(1) Non-FIFO selected

For supporting this function, the SCI provides the MPIE bit in CCR0. When the MPIE bit is set to 1, transfer of receive data from the RSR to the RDR, detection of a receive error, and setting the respective status flags RDRF, ORER, and FER in CSR are disabled until reception of data in which the multi-processor bit is set to 1.

On receiving a reception character in which the multi-processor bit is set to 1, the MPB bit in RDR is set to 1 and the MPIE bit in CCR0 is automatically cleared, thus returning to a non-multi-processor reception operation. At this time, an RXI interrupt is generated if the RIE bit in CCR0 is set.

When the multi-processor format is specified, specification of the parity bit is disabled.

Apart from this, there is no difference from the operation in the non-multi-processor asynchronous mode. A clock which is used for the multi-processor communication is also the same as the clock used in the non-multi-processor asynchronous mode.

(2) FIFO selected

For transmission, it should write to TDR.MPBT (Multi-Processor Bit Transfer) which corresponds to transmit data in TDR.TDAT. For reception, Multi-processor bit that is a part of receive data, is stored to RDR.MPB, and receive data is stored to RDR.RDAT. When the MPIE bit is set to 1, transfer of receive data from the RSR to the RDR.RDAT, detection of a receive error, and detection of DR, and setting the respective status flags RDRF, ORER, and FER in CSR are disabled until reception of data in which the multi-processor bit is set to 1.

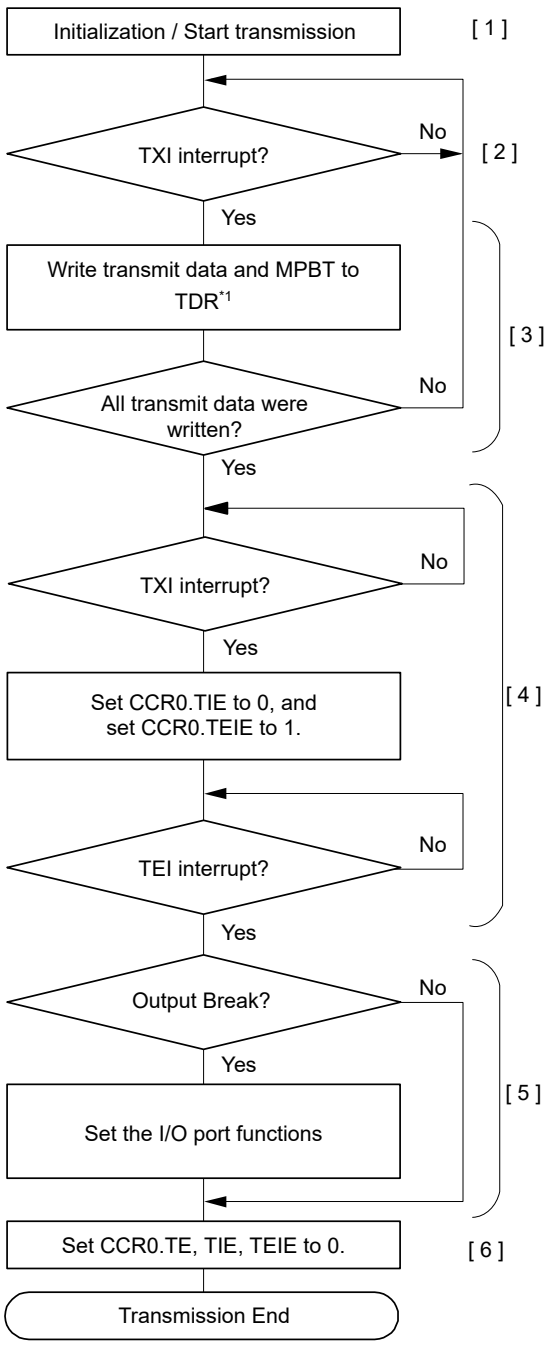
On receiving a reception character in which the multi-processor bit is set to 1, the MPB bit in RDR is set to 1, and receive data is stored to receive FIFO (RDR.RDAT), and the MPIE bit in CCR0 is automatically cleared, thus returning to a non-multi-processor reception operation. At this time, an RXI interrupt is generated if the RIE bit in CCR0 is set.

When the multi-processor format is specified, specification of the parity bit is disabled. Apart from this, there is no difference from the operation in the non-multi-processor asynchronous mode and non-FIFO selected.

31.4.1 Multi-Processor Serial Data Transmission

(1) Non-FIFO selected

Figure 31.33 shows a sample flowchart of multi-processor data transmission. In the ID transmission cycle, the ID should be transmitted with the MPBT bit in TDR set to 1. In the data transmission cycle, the data should be transmitted with the MPBT bit set to 0. The other operations are the same as the operations in asynchronous mode.



[1] Initialize SCI.
To enable TXI interrupt at the start of transmission, set TE and TIE to 1 with one instruction at the same time. And set CCR3.MP to 1.

[2] TXI interrupt request:
The transmit-data-empty-interrupt (TXI) is generated at following conditions:
When set TE and TIE to 1 at same time.
When transmission-data forwards to TSR from TDR.

[3] Serial transmission continuation procedure:
To continue serial transmission write transmit data and MPBT setting value to TDR*1 once by using a TXI. Transmit data and MPBT setting value can also be written by activating the DMAC.

[4] If you use TEI, set TIE = 0 and TEIE = 1 after writing the last-transmit-data to TDR*1 and transmission is started.

[5] Break output at the end of serial transmission:
To output a break, it is necessary to set SPB2IO and SPB2DT, for output status setting when TE is 0.
(If you set these bits at initialization, you can skip this step.)

[6] Set CCR0.TE, TIE, TEIE to 0.

Note 1. In case of byte access with data length 9 bits, write transmit data in order of TDR[15:8] and TDR[7:0].

Figure 31.33 Example of multi-processor serial transmission flowchart (non-FIFO selected)

(2) FIFO selected

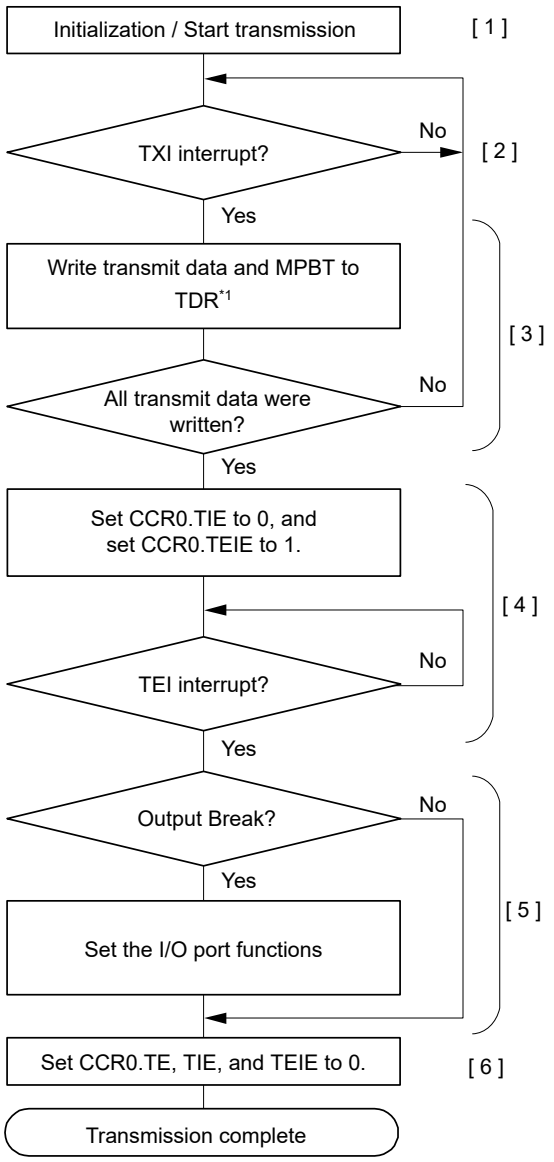
Figure 31.34 shows an example of data format that is written to transmit FIFO (TDR.TDAT) in multi-processor mode. Write MPBT in bit 9 of TDR. And write data to transmit FIFO (TDR.TDAT) corresponded to data length. It should write to 0 for unused bits.

Data Length	Register setting		Transmit data in TDR[15:0]															
	CCR3. CHR[1:0]		b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	—	—	—	—	—	—	MPBT	—	—	TDAT[6:0]						
8 bits	1	0	—	—	—	—	—	—	MPBT	—	TDAT[7:0]							
9 bits	0	x	—	—	—	—	—	—	MPBT	TDAT[8:0]								

Note: The bits Indicated by "—" are not used. Therefore, those bits should be set to 0.

Figure 31.34 Data format in multi-processor mode that is written to transmit FIFO (TDR.TDAT)

Figure 31.35 shows a sample flowchart for multi-processor data transmission with FIFO selected. In the ID transmission cycle, the ID should be transmitted with the MPBT bit in TDR set to 1. In the data transmission cycle, the data should be transmitted with the MPBT bit set to 0. The other operations are the same as the operations in asynchronous mode with FIFO selected.



[1] Initialize SCI.
To enable TXI interrupt at he start of transmission, set TE and TIE to 1 with one instructions at the same time. And set CCR3.FM and MP to 1.

[2] TXI interrupt request:
The transmit-data-empty-interrupt (TXI) is generated when transmission-data forwards to TSR from the transmit FIFO (TDR) and the number of transmit-data written in the transmit FIFO is equal to or less than the transmit FIFO threshold.

[3] Serial transmission continuation procedure:
To continue serial transmission, write all transmit data and MPBT setting value to the transmit FIFO (TDR) by using a TXI interrupt. Transmit data and MPBT setting value can also be written to the transmit FIFO (TDR) by activating the DMAC.

[4] If you use TEI, set TIE = 0 and TEIE = 1 after writing the last-transmit-data to TDR*1.

[5] Break output at the end of serial transmission:
To output a break, it is necessary to set SPB2IO and SPB2DT, for output status setting when TE is 0.
(If you set these bits at initialization, you can skip this step.)

[6] Set CCR0.TE, TIE, and TEIE to 0.

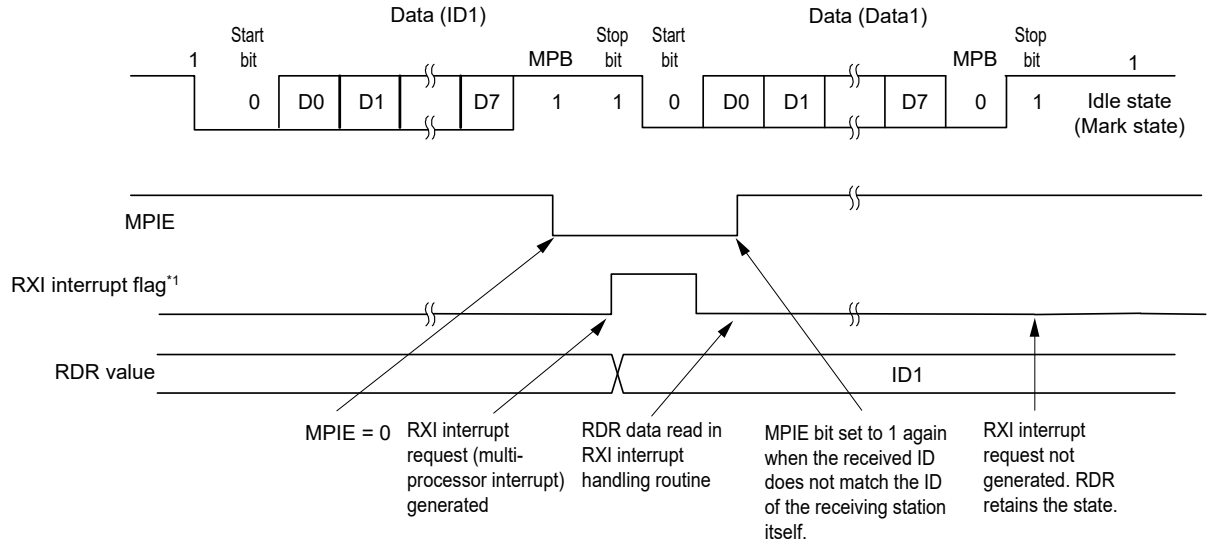
Note 1. In case of byte access with data length 9 bits, write transmit data in order of TDR[15:8] and TDR[7:0].

Figure 31.35 Example of serial transmission flowchart in multi-processor mode (FIFO selected)

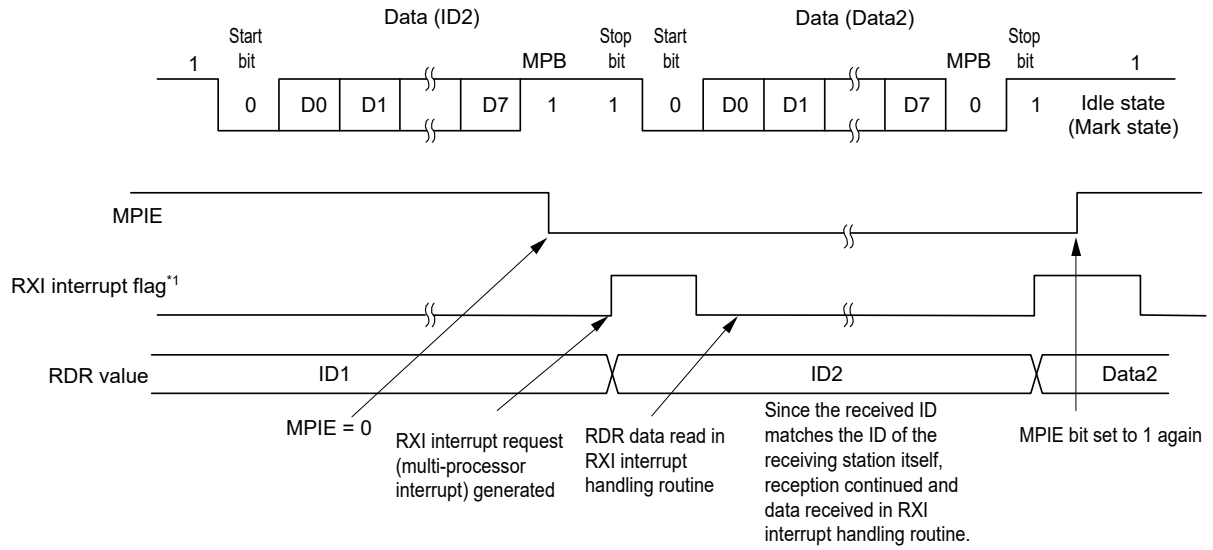
31.4.2 Multi-Processor Serial Data Reception

(1) Non-FIFO selected

Figure 31.37 and Figure 31.38 are sample flowcharts of multi-processor data reception. When the MPIE bit in CCR0 is set to 1, reading the communication data is skipped until reception of the communication data in which the multi-processor bit is set to 1. When the communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to RDR. At this time, the RXI interrupt request is generated. The other operations are the same as the operations in asynchronous mode. Figure 31.36 is the example of operation for reception.



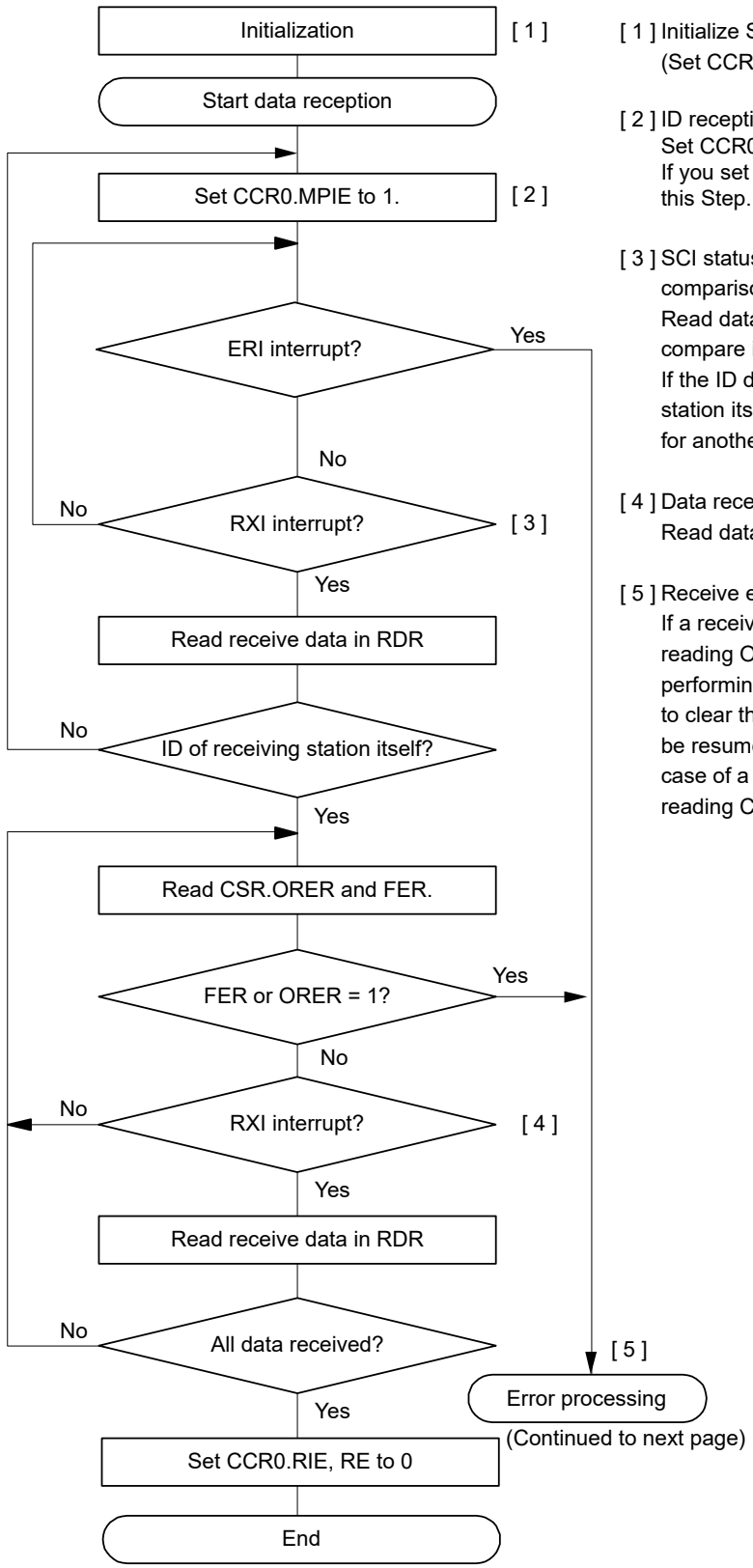
(a) When the received ID does not match the ID of the receiving station itself



(b) When the received ID matches the ID of the receiving station itself

Note 1. See section 11, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 31.36 Example of SCI reception (8-bit data/multi-processor bit/1 stop bit)



[1] Initialize SCI.
(Set CCR0.RE, RIE to 1. And set CCR3.MP to 1.)

[2] ID reception cycle:
Set CCR0.MPIE to 1. and wait for ID reception.
If you set this bits at initialization ([1]), you can skip this Step.

[3] SCI status confirmation and reception and comparison of ID:
Read data in RDR at the first RXI interrupt and compare it with the ID of the receiving station itself.
If the ID does not match the ID of the receiving station itself, set the MPIE bit to 1 again, and wait for another RXI interrupt request.

[4] Data reception at an RXI interrupt:
Read data in RDR once in the RXI interrupt routine.

[5] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading ORER and FER flags in CSR. After performing the appropriate error processing be sure to clear the ORER and FER flags. Reception cannot be resumed if any of these flags is set to 1. In the case of a framing error, a break can be detected by reading CSR.RXDMON bit.

Figure 31.37 Example of multi-processor serial reception flowchart (1) (non-FIFO selected)

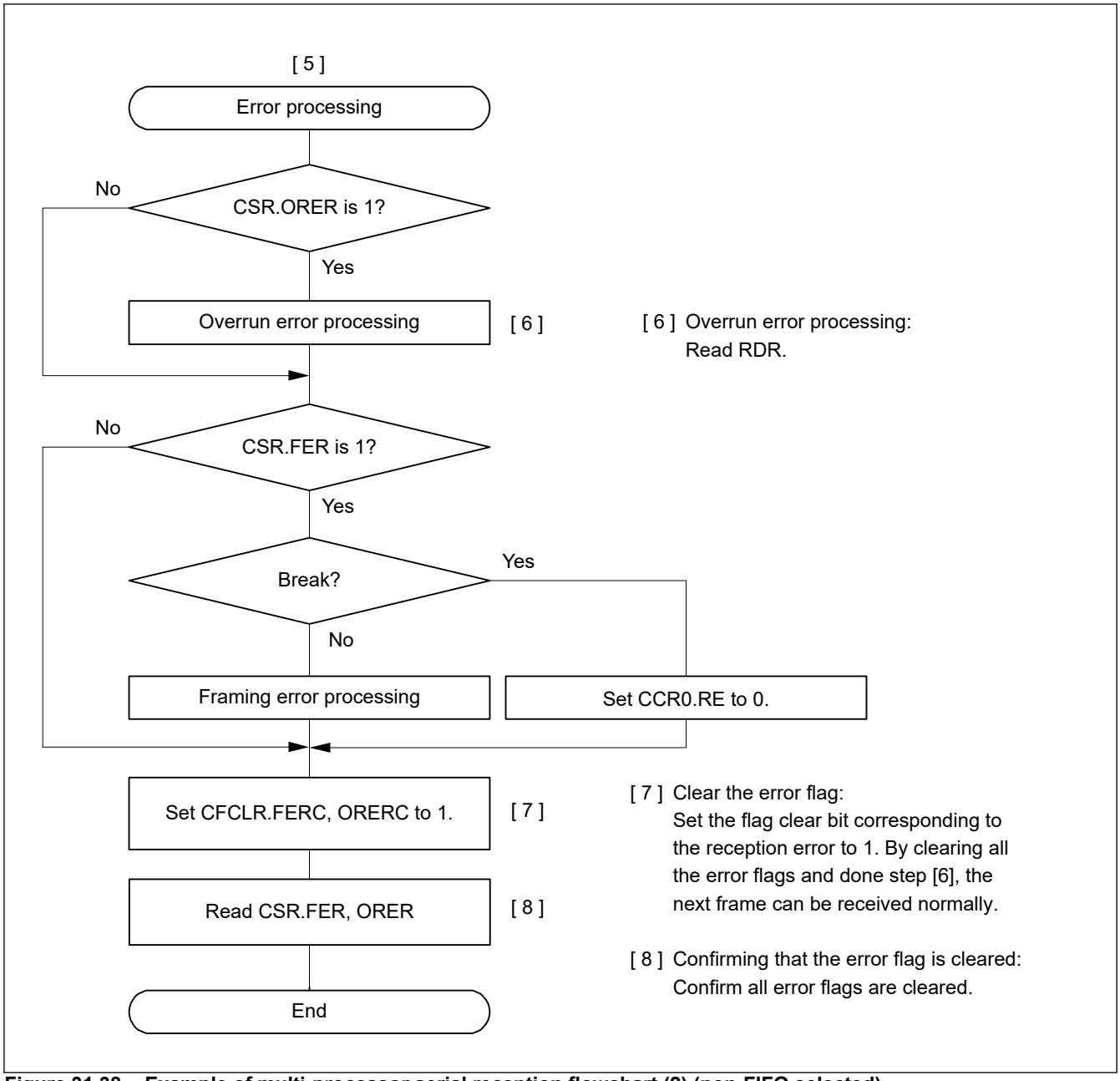


Figure 31.38 Example of multi-processor serial reception flowchart (2) (non-FIFO selected)

(2) FIFO selected

Figure 31.39 shows an example of data format that is stored to receive FIFO (RDR) in multi-processor mode. MPB is stored in bit 9 of RDR. 0 is stored to FPER and PER flag bit. Data is stored to receive FIFO (RDR) corresponded to data length. It is stored to 0 for unused bits. Reading the receive FIFO (RDR register) updates the FFER, FPER, MPB flags and receive data (RDAT[8:0]) in the receive FIFO (RDR register) with the next received data. The FER, PER, and ORER flags in the receive FIFO (RDR register) always reflect the status of the corresponding flags in the CSR and FRSR registers.

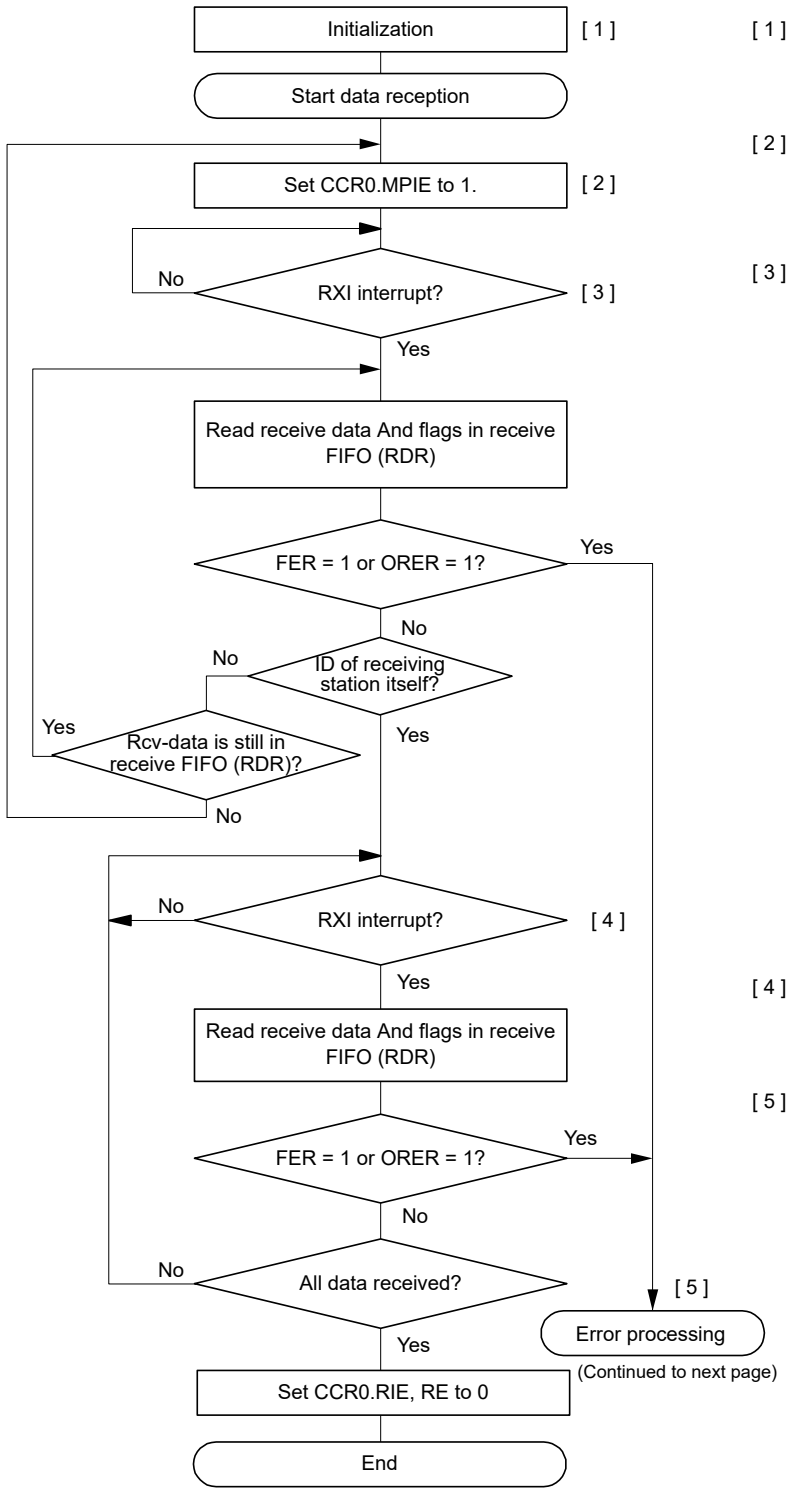
Data Length	Register Setting CCR3.CHR[1:0]		Receive data in RDR[31:0] b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
7 bits	1	1	0	0	0	FFER	FPER	DR	MPB	0	0	RDAT[6:0]						
8 bits	1	0	0	0	0	FFER	FPER	DR	MPB	0	RDAT[7:0]							
9 bits	0	x	0	0	0	FFER	FPER	DR	MPB	RDAT[8:0]								
Data Length	CCR3.CHR[1:0]		b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
7 bits	1	1	0	0	0	FER	PER	0	0	ORER	0	0	0	0	0	0	0	0
8 bits	1	0	0	0	0	FER	PER	0	0	ORER	0	0	0	0	0	0	0	0
9 bits	0	x	0	0	0	FER	PER	0	0	ORER	0	0	0	0	0	0	0	0

Note: When a 7-bit data length is selected, the RDAT[8:7] bits are always read as 00b.
When an 8-bit data length is selected, the RDAT[8] bit is always read as 0.

Figure 31.39 Data format in multi-processor mode that is stored to receive FIFO (FIFO selected)

Figure 31.40 shows a sample flowchart for multi-processor data reception at FIFO selected. When the MPIE bit in CCR0 is set to 1, reading the communication data is skipped until reception of the communication data in which the multiprocessor bit is set to 1. When the communication data in which the multi-processor bit is set to 1 is received, the received data, MPB, and each errors are transferred to receive FIFO (RDR), and the MPIE bit in CCR0 is automatically cleared, thus returning to a normal reception operation. After a framing error occurred and CSR.FER flag is set to 1, but SCI continues data reception.

The other operations are the same as the operations in asynchronous mode at non-FIFO selected.



[1] Initialize SCI.
(Set CCR0.RE, RIE to 1.
And set CCR3.MP and FM to 1.)

[2] ID reception cycle:
Set CCR0.MPIE to 1. and wait for ID reception. If you set this bits at Initialization ([1]), you can skip this Step.

[3] SCI status confirmation and reception and comparison of ID:
The subsequent receive data including the data with MPB bit 1 is stored in receive FIFO (RDR). When the number of receive data stored in the receive FIFO (RDR register) reaches or exceeds the receive FIFO threshold, the CSR.RDRF flag is set to 1 and an RXI interrupt request is generated. When quantity of data stored in the receive FIFO data register (RDR) falls the specified receive triggering number and received data is equal to or greater than 1, and that no next data has been received yet after the elapse of 15 etus from the last stop bit, FRSR.DR is set to 1. Read data in receive FIFO (RDR) at the first RXI interrupt, and compare it with the ID of the receiving station itself. If the ID does not match the ID of the receiving station itself, it reads until the data with MPB = 1, and compares next ID. If it is not a data with MPB = 1 in receive FIFO (RDR), it is set the MPIE bit to 1 again, and wait for another RXI interrupt request.

[4] Data reception at an RXI interrupt:
Read data in the receive FIFO (RDR) once in the RXI interrupt routine.

[5] Receive error processing and break detection:
If a receive error occurs, an error is identified by reading ORER and FER flags in CSR. After performing the appropriate error processing be sure to clear the ORER and FER flags. In the case of a framing error, a break can be detected by reading the CSR.RXDMON bit.

Figure 31.40 Example flowchart of serial reception in multi-processor mode (1) (FIFO selected)

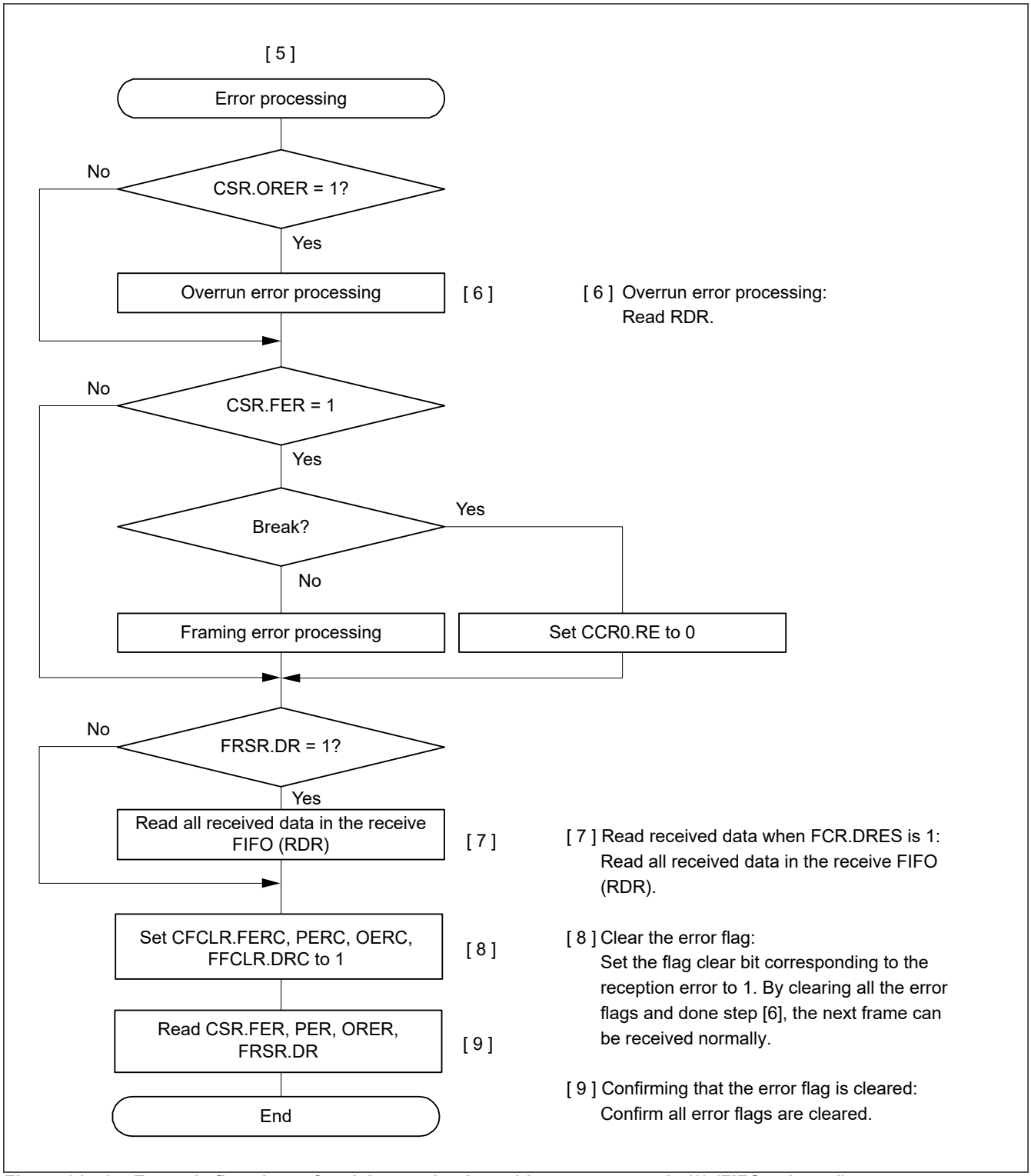


Figure 31.41 Example flowchart of serial reception in multi-processor mode (2) (FIFO selected)

31.5 Smart Card Interface Mode

The SCI supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards), as an extended function.

Smart card interface mode can be selected using the appropriate register.

31.5.1 Sample Connection

Figure 31.42 shows a sample connection between a smart card (IC card) and this LSI.

As in the figure, since this LSI communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to VCC using a resistor. Setting the TE and RE bits in CCR0 to 1 with an IC card disconnected enables closed transmission/reception allowing self-diagnosis.

To supply an IC card with the clock pulses generated by the SCI, input the SCKn pin output to the CLK pin of an IC card.

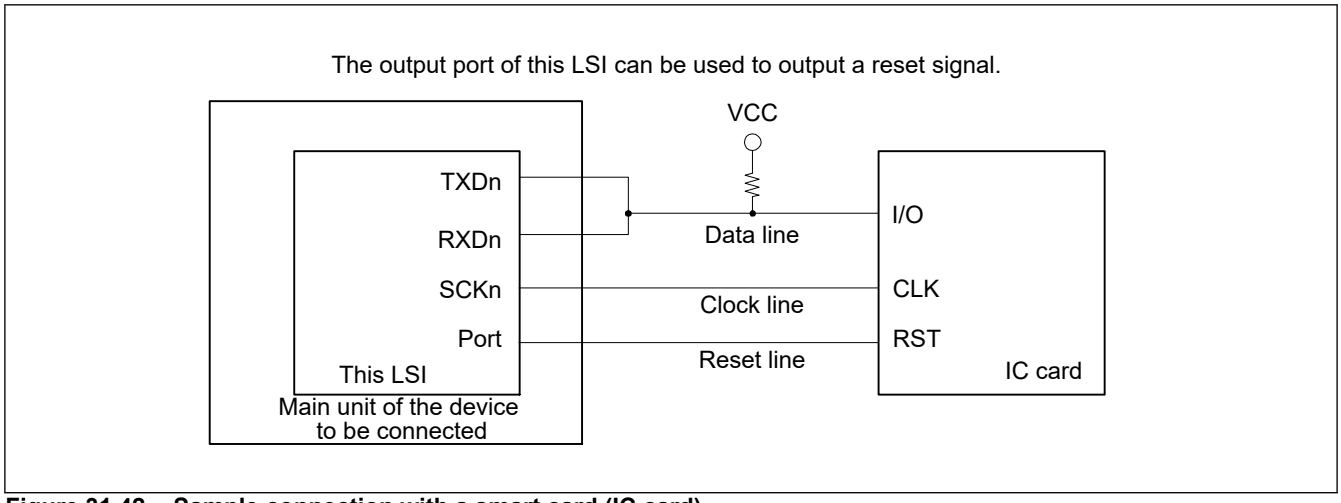


Figure 31.42 Sample connection with a smart card (IC card)

31.5.2 Data Format (Except in Block Transfer Mode)

Figure 31.43 shows the data transfer formats in smart card interface mode.

- One frame consists of 8-bit data and a parity bit in asynchronous mode.
- During transmission, at least 2 etu (elementary time unit: time required for transferring 1 bit) is secured as a guard time from the end of the parity bit until the start of the next frame.
- If a parity error is detected during reception, a low-level error signal is output for 1 etu after 10.5 etu has passed from the start bit.
- If an error signal is sampled during transmission, the same data is automatically retransmitted after at least 2 etu.

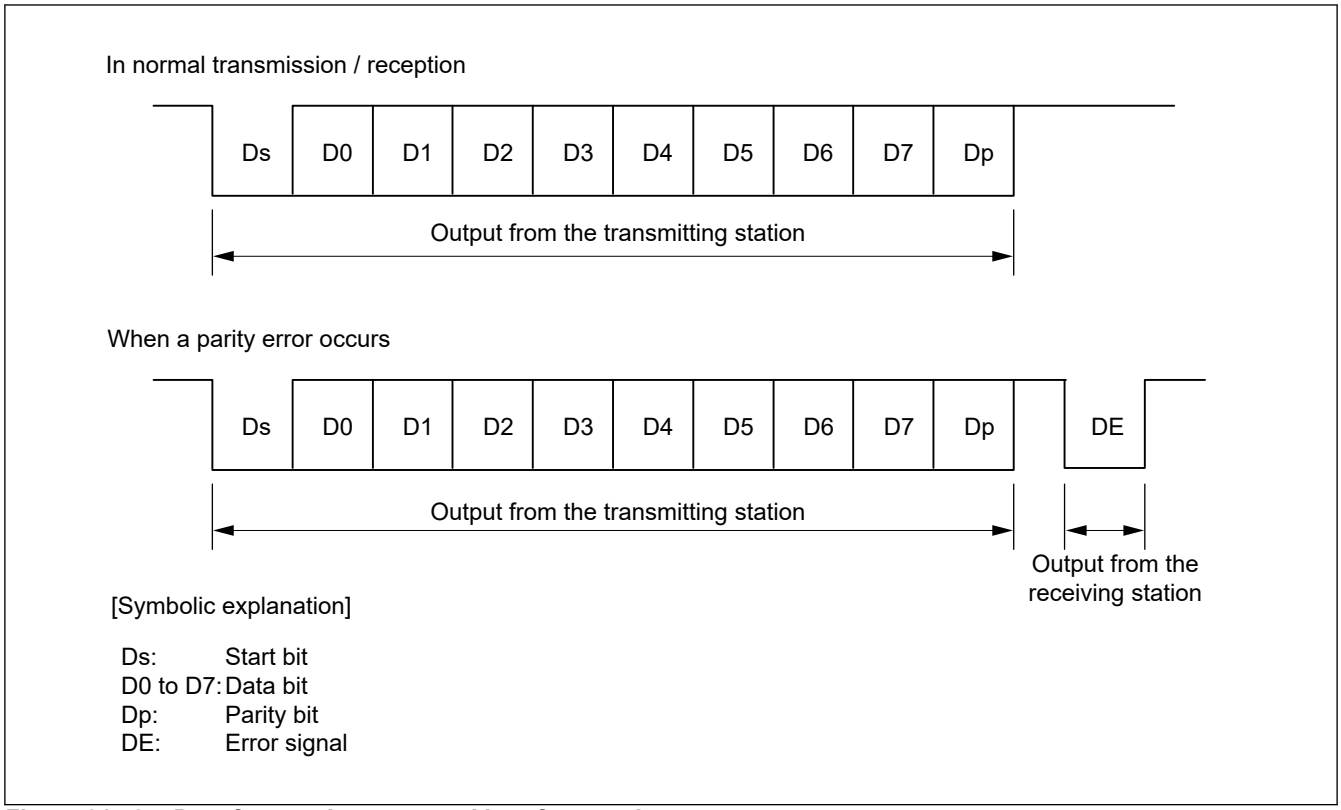


Figure 31.43 Data formats in smart card interface mode

For communications with IC cards of the direct convention type and inverse convention type, follow the procedure below.

(1) Direct Convention Type

For the direct convention type, logic levels 1 and 0 correspond to states Z and A, respectively, and data is transferred with LSB first as the start character, as shown in Figure 31.44. Therefore, data in the start character in the figure is 0x3B.

When using the direct convention type, write 0 to the CCR3.SINV bit and write 1 to the CCR3.LSBF bit. Write 0 to CCR1.PM in order to use even parity, which is prescribed by the smart card standard.

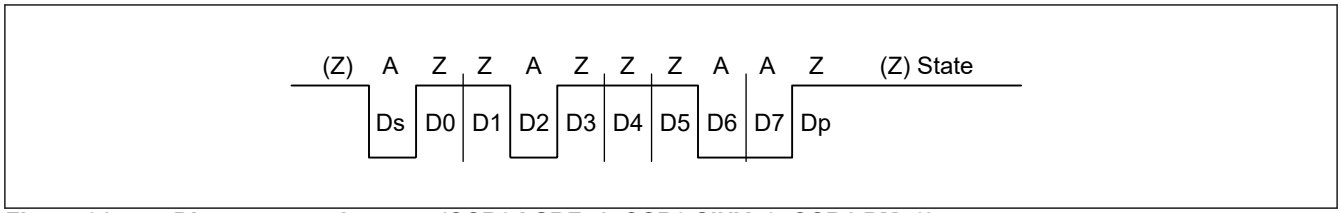


Figure 31.44 Direct convention type (CCR3.LSBF=1, CCR3.SINV=0, CCR1.PM=0)

(2) Inverse Convention Type

For the inverse convention type, logic levels 1 and 0 correspond to states A and Z, respectively and data is transferred with MSB first as the start character, as shown in Figure 31.45. Therefore, data in the start character in the figure is 0x3F.

When using the inverse convention type, write 1 to the CCR3.SINV bit and write 0 to the CCR3.LSBF bit. The parity bit is logic level 0 to produce even parity, which is prescribed by the smart card standard, and corresponds to state Z. Since the SINV bit of this LSI only inverts data bits D7 to D0, write 1 to CCR1.PM to invert the parity bit for both transmission and reception.

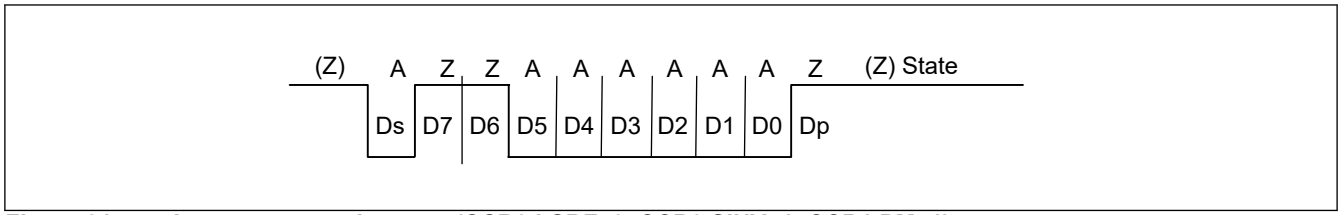


Figure 31.45 Inverse convention type (CCR3.LSBF=0, CCR3.SINV=1, CCR1.PM=1)

31.5.3 Block Transfer Mode

Block transfer mode is different from non-block-transfer-mode of smart card interface mode in the following respects.

- Even if a parity error is detected during reception, no error signal is output. Since CSR.PER is set by error detection, clear the PER bit before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is secured as a guard time from the end of the parity bit until the start of the next frame.
- Since the same data is not retransmitted during transmission, CSR.TEND is set 11.5 etu after transmission start.
- In block transfer mode, CSR.ERS flag indicates the error signal status as in non-block-transfer-mode of smart card interface mode, but the flag is always read as 0 because no error signal is transferred.

31.5.4 Receive Data Sampling Timing and Reception Margin

Only the internal clock generated by the baud rate generator can be used as a transfer clock in smart card interface mode. In this mode, the SCI can operate on a base clock with a frequency of 32, 64, 372, 256, 93, 128, 186, or 512 times the bit rate according to the settings of CCR2.BCP[2:0]. For data reception, the falling edge of the start bit is sampled with the base clock to perform synchronization. Receive data is sampled on the 16th, 32nd, 186th, 128th, 46th, 64th, 93rd, and 256th rising edges of the base clock so that it can be latched at the middle of each bit as shown in Figure 31.46. The reception margin here is determined by the following formula.

$$M = \left|(0.5 - \frac{1}{2N}) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F)\right| \times 100\%$$

M: Reception margin (%)

N: Ratio of bit rate to clock (N = 32, 64, 372, 256)

D: Duty cycle of clock (D = 0 to 1.0)

L: Frame length (L = 10)

F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5, and N = 372 in the above formula, the reception margin is determined by the formula below.

$$M = \{0.5 - 1/(2 \times 372)\} \times 100\% = 49.866\%$$

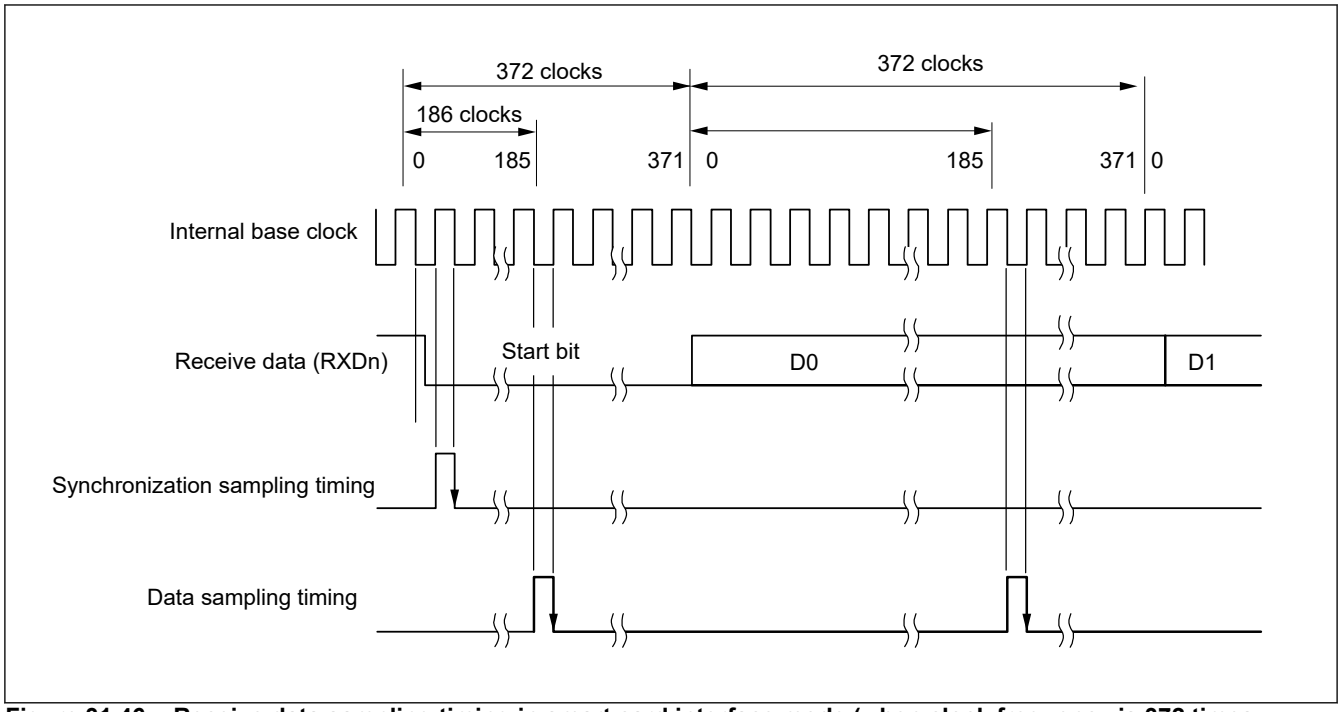


Figure 31.46 Receive data sampling timing in smart card interface mode (when clock frequency is 372 times the bit rate)

31.5.5 Initialization of SCI (Smart Card Interface Mode)

Before transmitting and receiving data, write 0 to CCR0.TE and CCR0.RE (Or write the initial value to CCR0). And initialize SCI following example flowchart in Figure 31.47.

Be sure to set the initial value in the TIE, RIE, TE, RE, and TEIE bits in CCR0 register before switching from transmission mode to reception mode and vice versa. Even if the RE bit is set to 0, the RDR register is not initialized. In transmission mode, set 1 to the TE bit and TIE bit simultaneously, then the TXI interrupt request is generated. To change reception mode to transmission mode, first check that reception has completed, and then initialize SCI. At the end of initialization, set TE = 1 and RE = 0. Reception completion can be verified by RXI request, CSR.ORER, or CSR.PER. To change transmission mode to reception mode, first check that transmission has completed, and then initialize SCI. At the end of initialization, set TE = 0 and RE = 1. Transmission completion can be verified by reading CSR.TEND.

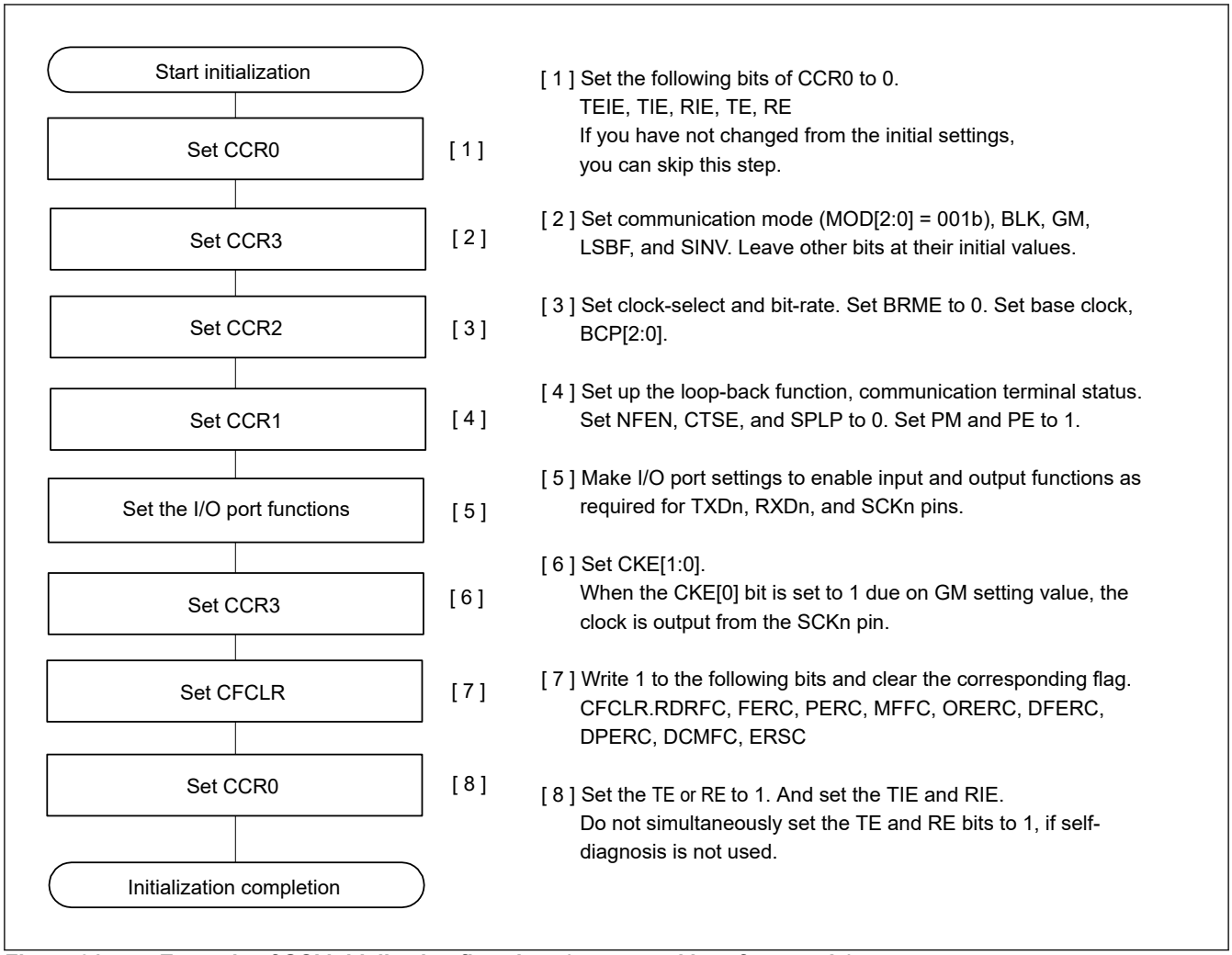


Figure 31.47 Example of SCI initialization flowchart (smart card interface mode)

Figure 31.48 is a timing chart when data transmission is performed by making transition to the Smart Card Interface mode according to the above flow chart. The figure shows the case when CCR3.GM bit is 0. As shown in the figure, when the pin function is set to the SCK pin, the SCK pin is high impedance because the CCR3.CKE[0] bit is 0. When the TXD pin is set, the TXD pin is high impedance because the CCR0.TE bit is 0. Start clock output to the SCK pin with the clock output setting CCR3.CKE[0] to 1, start data transmission by writing transmit data after setting CCR0.TE to 1.

In the smart card interface mode, even if not communicating at CCR0.TE = 0 and CCR0.RE = 0, the clock is continuously output if the clock output setting is used.

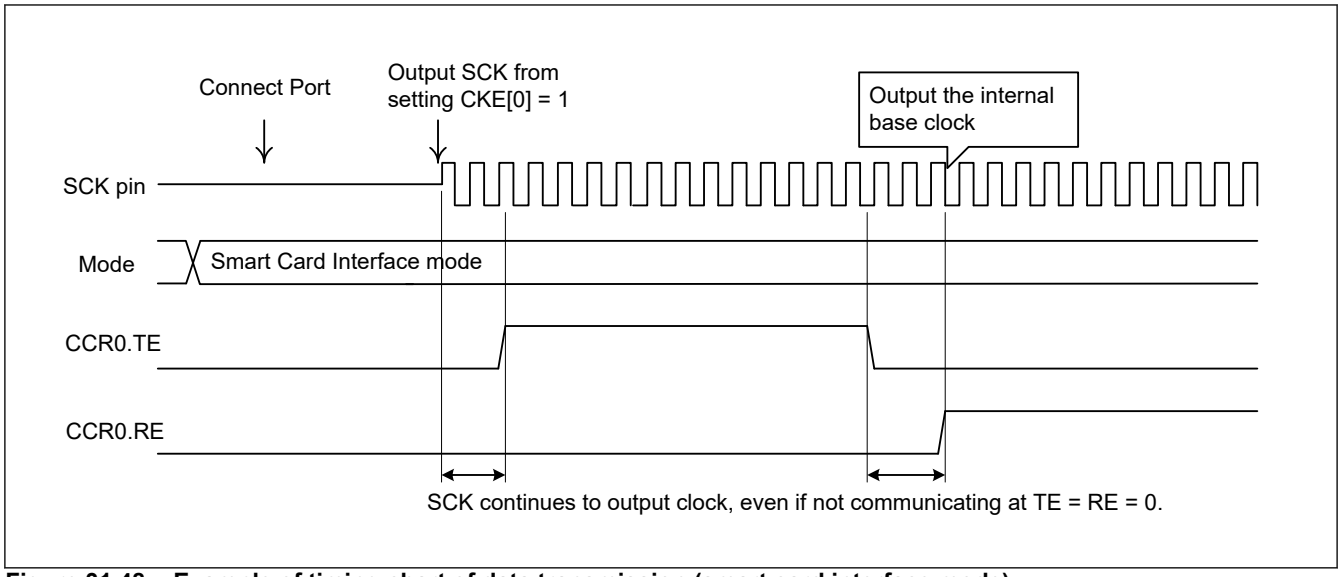
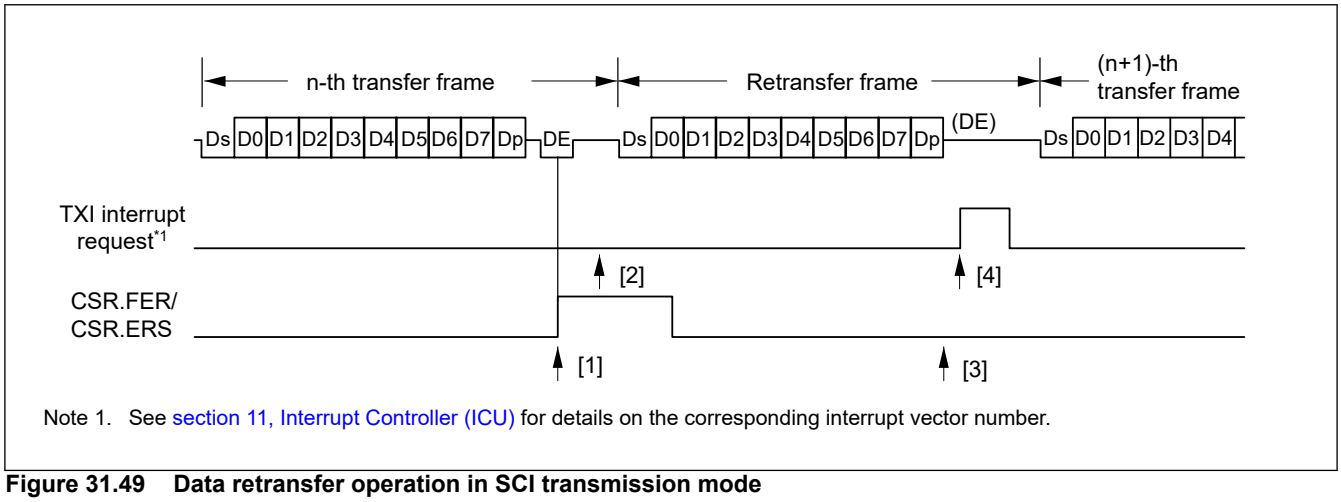


Figure 31.48 Example of timing chart of data transmission (smart card interface mode)

31.5.6 Serial Data Transmission (Except in Block Transfer Mode)

Serial data transmission in smart card interface mode (except in block transfer mode), in that an error signal is sampled and data can be retransmitted, is different from that in non-smart card interface mode. Figure 31.49 shows the data retransfer operation during transmission.

1. When an error signal from the receiver end is sampled after one-frame data has been transmitted, CSR.ERS is set to 1. If CCR0.RIE is 1 at this time, an ERI interrupt request is generated. Clear the ERS flag to 0 before the next parity bit is sampled.
2. For a frame in which an error signal is received, CSR.TEND is not set. Data is retransferred from TDR to TSR allowing data retransmission automatically.
3. If no error signal is returned from the receiver, the ERS flag is not set to 1.
4. In this case, the SCI judges that transmission of one-frame data (including retransfer) has been completed, and the TEND flag is set. If CCR0.TIE is 1 at this time, a TXI interrupt request is generated. Writing transmit data to TDR starts transmission of the next data.



Note 1. See section 11, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 31.49 Data retransfer operation in SCI transmission mode

Figure 31.51 shows a sample flowchart of serial transmission. All the processing steps are automatically performed by using a TXI interrupt request to activate the DMAC. When CSR.TEND is set to 1 in transmission, if the CCR0.TIE is 1, a TXI interrupt request is generated. The DMAC is activated by a TXI interrupt request if the TXI interrupt request is specified as a source of DMAC activation beforehand, allowing transfer of transmit data. The TEND flag is automatically set to 0 when the DMAC transfers the data. If an error occurs, the SCI automatically retransmits the same data. During this retransmission, the TEND flag is kept to 0 and the DMAC is not activated. Therefore, the SCI and DMAC automatically

transmit the specified number of bytes, including retransmission in the case of error occurrence. However, since the ERS flag is not automatically cleared, set the RIE bit to 1 beforehand to enable an ERI interrupt request to be generated at error occurrence, and clear the ERS flag to 0.

When transmitting/receiving data using the DMAC, be sure to make settings to enable the DMAC before making SCI settings. For DMAC settings, see section 14, DMA Controller (DMAC).

Note that CSR.TEND flag is set in different timings depending on the CCR3.GM bit setting. Figure 31.50 shows the TEND flag generation timing.

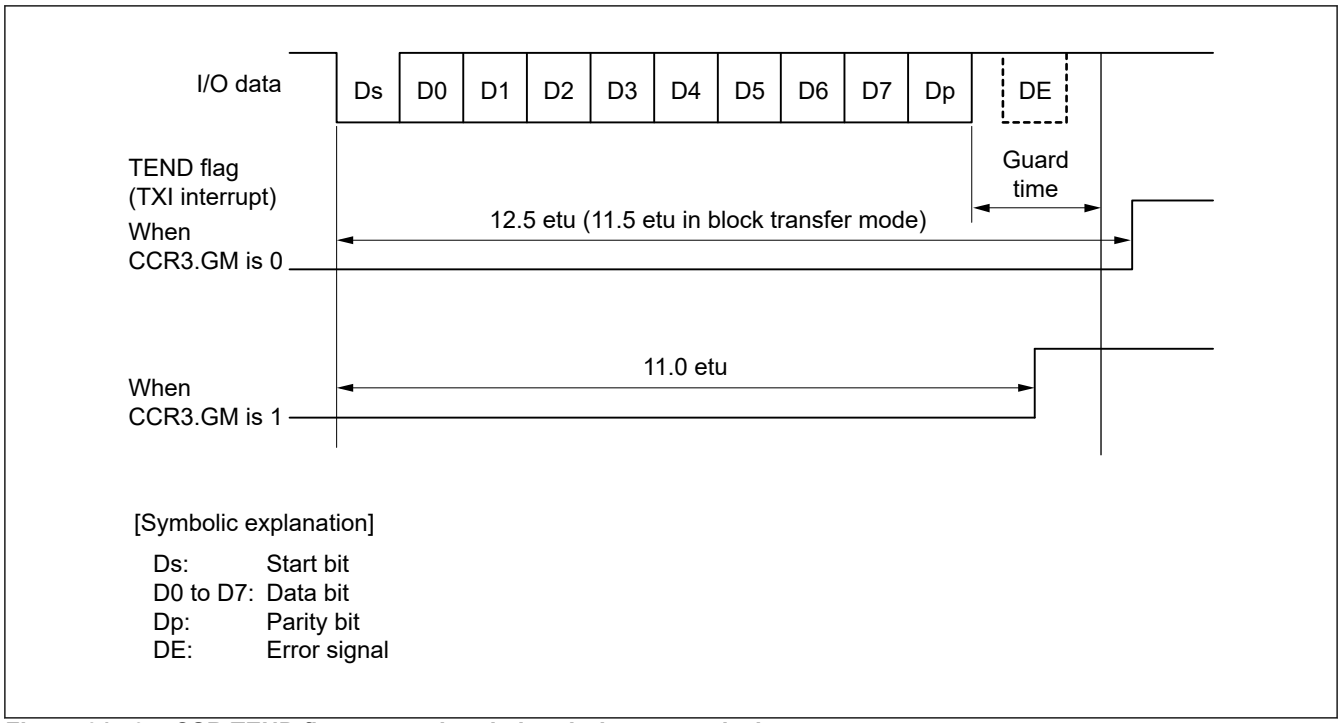


Figure 31.50 CSR.TEND flag generation timing during transmission

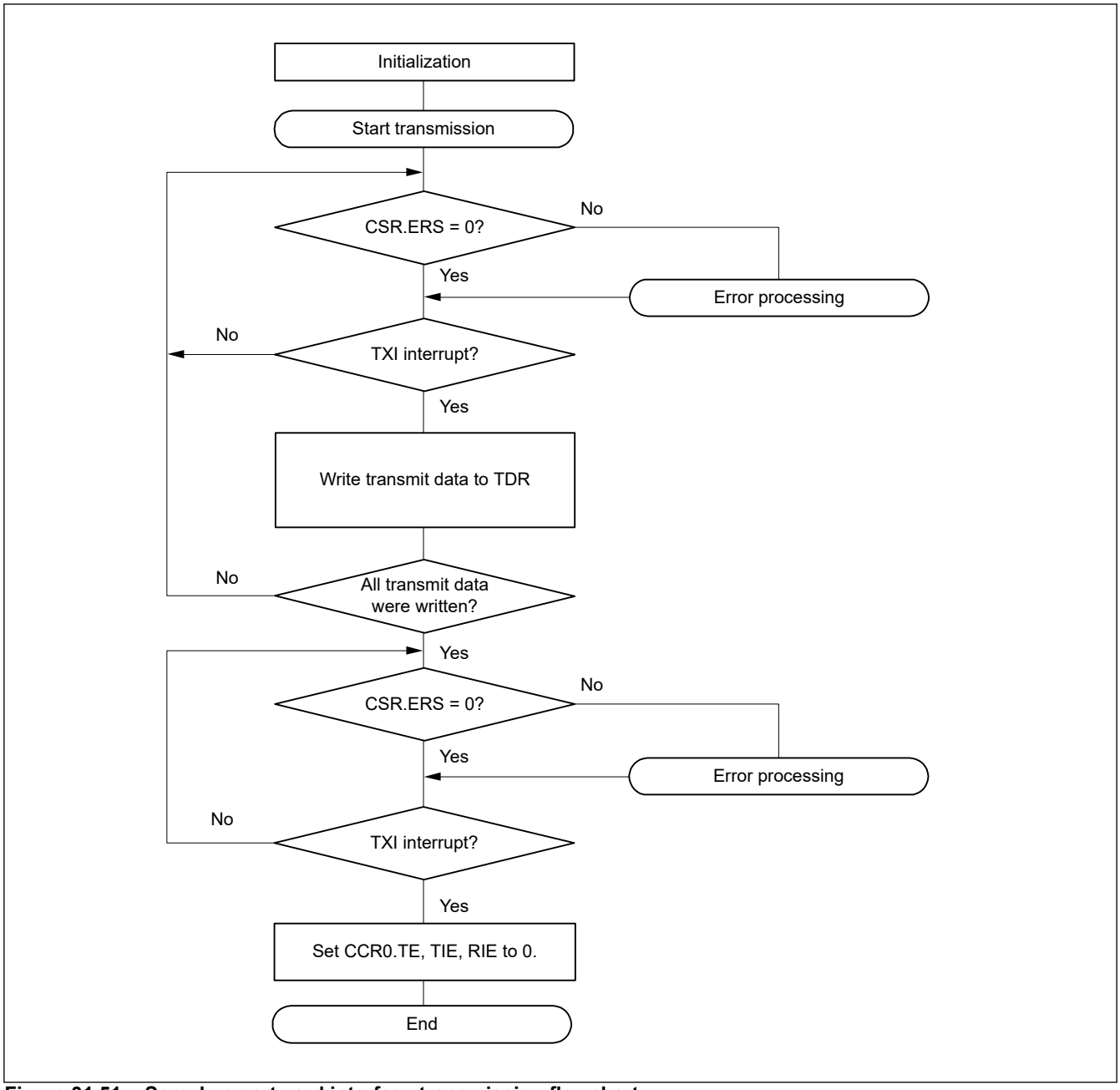
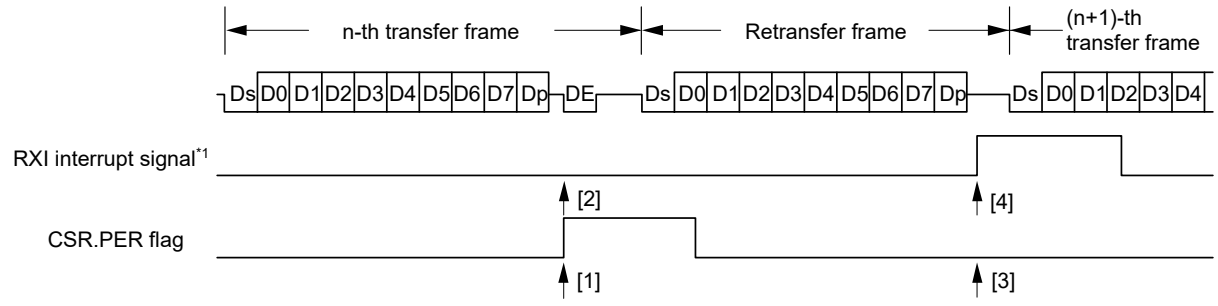


Figure 31.51 Sample smart card interface transmission flowchart

31.5.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is like that in non-smart card interface mode. Figure 31.52 shows the data retransfer operation in reception mode.

1. If a parity error is detected in receive data, the CSR.PER flag is set to 1. When the CCR0.RIE is 1 at this time, an ERI interrupt request is generated. Clear the PER flag to 0 before the next parity bit is sampled.
2. For a frame in which a parity error is detected, no RXI interrupt is generated.
3. When no parity error is detected, the CSR.PER flag is not set to 1.
4. In this case, data is determined to have been received successfully. When the CCR0.RIE is 1, an RXI interrupt request is generated.



Note 1. See section 11, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 31.52 Data retransfer operation in SCI reception mode (data retransfer operation during reception)

Figure 31.53 shows a sample flowchart for serial data reception. All the processing steps are automatically performed using an RXI interrupt request to activate the DMAC. In reception, setting the RIE bit to 1 allows an RXI interrupt request to be generated. The DMAC is activated by an RXI interrupt request if the RXI interrupt request is specified as a source of DMAC activation beforehand, allowing transfer of receive data. If an error occurs during reception and either the CSR.ORER or PER flag is set to 1, a receive error interrupt (ERI) request is generated. Clear the error flag after the error occurrence. If an error occurs, the DMAC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DMAC is transferred.

Even if a parity error occurs and the PER flag is set to 1 during reception, receive data is transferred to RDR, thus allowing the data to be read. When a reception is forcibly terminated by setting the CCR0.RE to 0 during operation, read the RDR register because the received data which has not yet been read may be left in RDR.

Note: For operations in block transfer mode, see section 31.3. Asynchronous Mode.

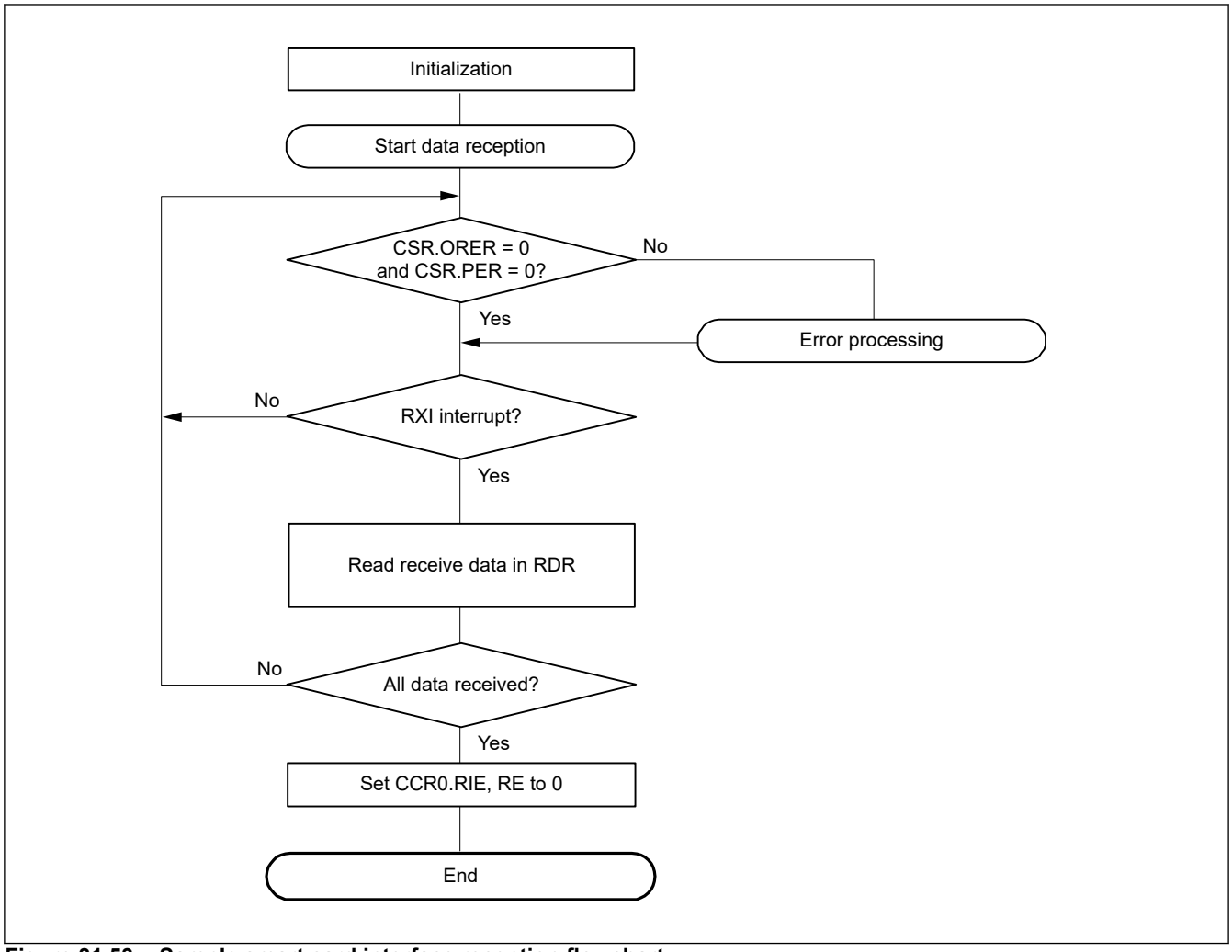


Figure 31.53 Sample smart card interface reception flowchart

31.5.8 Clock Output Control

When the CCR3.GM is set to 1, the clock output can be controlled by the CCR3.CKE[1:0]. Refer to the description of the CKE in section 31.2.8. CCR3 : Common Control Register 3. When setting the clock output, the base clock described in section 31.3.3. Clock is output, so the width of the clock pulse can be kept to the width specified by setting the bit rate. It is described in section 31.2.7. CCR2 : Common Control Register 2, the bit rate is set by CCR2.CKS[1:0], CCR2.BCP[2:0], and BRR[7:0].

Figure 31.54 shows the timing chart for explaining clock output control. This is an example when the CCR3.CKE[1] is set to 0 and the CCR3.CKE[0] is controlled.

When the CCR3.GM is 0, output control by the CCR3.CKE[0] is immediately reflected on the SCK pin, so there is a possibility that pulses with an unintended width may be output from the SCK pin.

When the CCR3.GM is 1, the output pulse control by the CCR3.CKE[0] controls the pulse width set to be based on the state of the base clock.

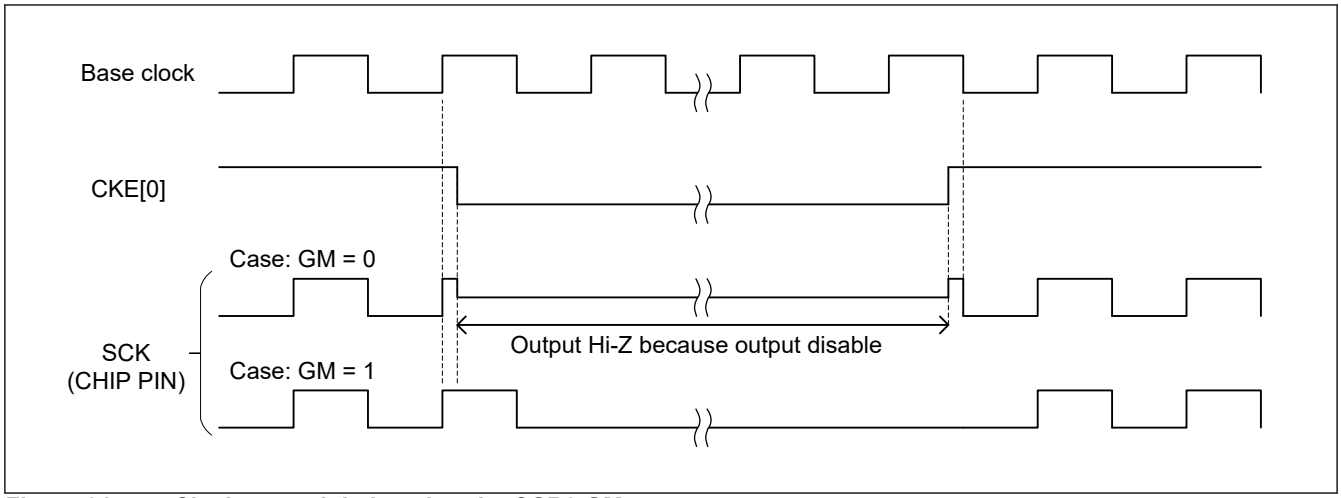


Figure 31.54 Clock control timing chart by CCR3.GM

31.6 Simple I2C mode

Simple I2C mode supports master operation only. Simple I2C bus format is composed of 8 data bits and an acknowledge bit. A slave-address frame follows start condition or restart condition. And the slave-address frame is used to specify a slave device as the partner by master device. The currently specified slave device remains valid until a new slave device is specified or a stop condition is satisfied. The 8-bit data of each frame is transmitted from MSB.

Figure 31.55 shows I2C bus format, and Figure 31.56 shows the timing of I2C.

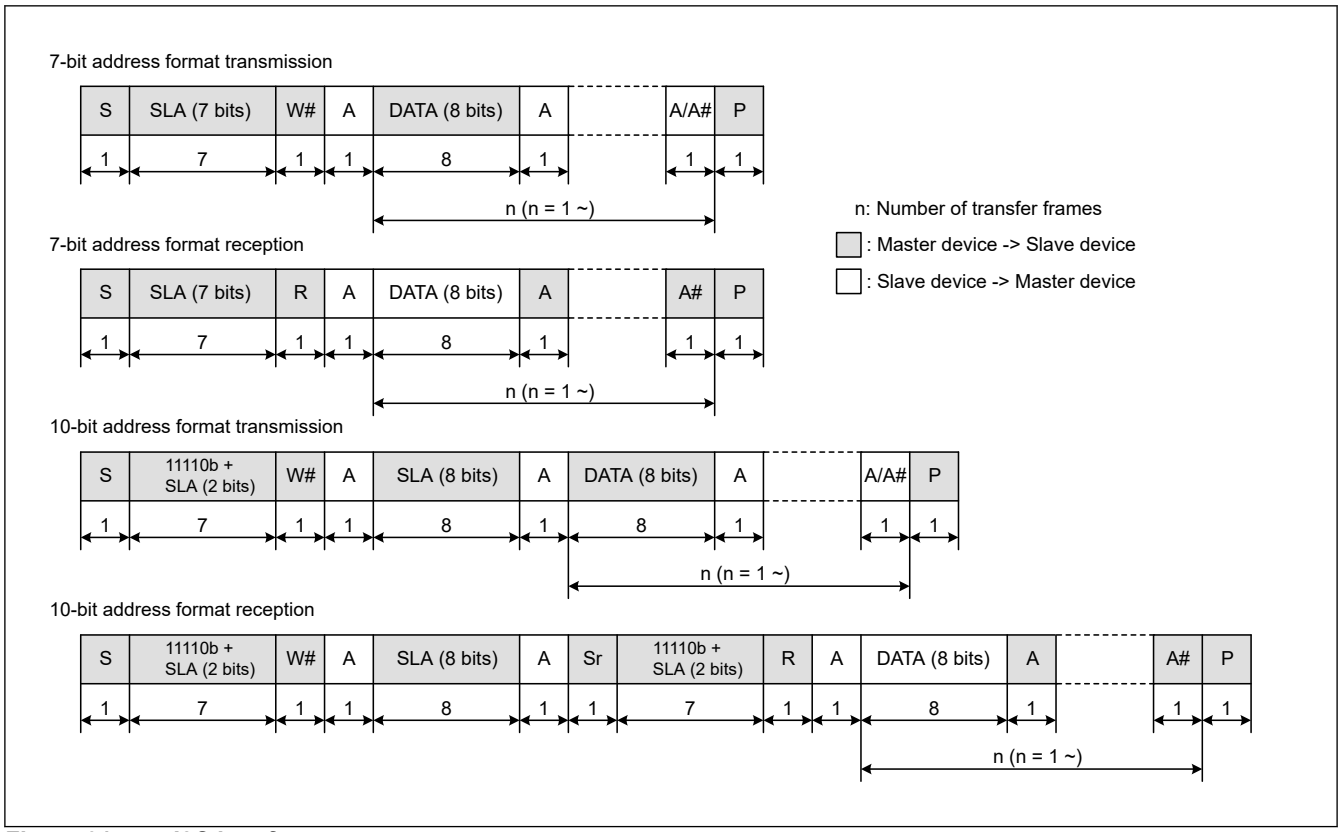


Figure 31.55 I2C bus format

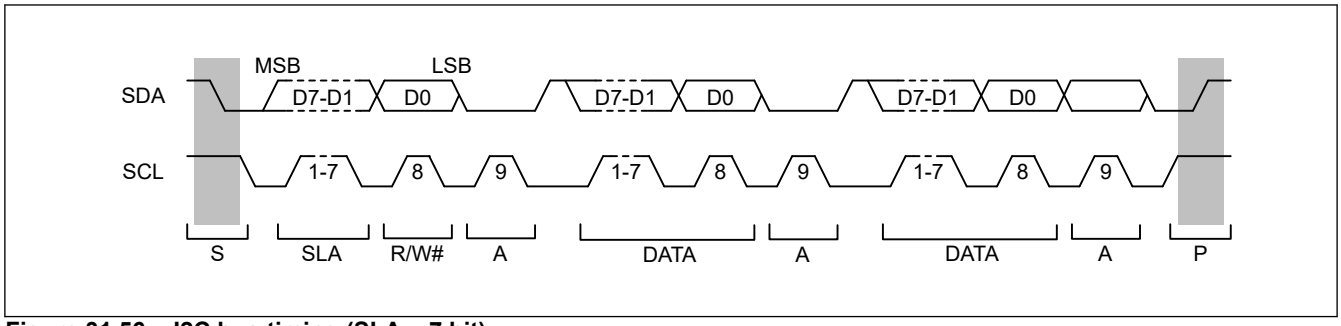


Figure 31.56 I2C bus timing (SLA = 7 bit)

[Symbol explanation]

S: Indicates a start condition. The master device changing the level on the SDAn line from the high to the low level when the SCLn line is at the high level.

SLA: Indicates a slave address. The master device selects a slave device.

R/W#: Indicates the direction of transfer (reception or transmission). When R/W# is high, the transfer direction is from the slave device to the master device. When R/W# is low, the transfer direction is from the master device to the slave device.

A/A#: Indicates an acknowledge. This is returned by the slave device for master transmission and by the master device for master reception. The low level indicates ACK and the high level indicates NACK.

Sr: Indicates a restart condition. The master device changing the level on the SDAn line from the high to the low level when the SCLn line is at the high level.

DATA: Indicates the received or transmitted data.

P: Indicates a stop condition. The master device changing the level on the SDAn line from the low to the high level when the SCLn line is at the high level.

31.6.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to ICR.IICSTAREQ causes the generation of a start condition. The following operations are done at the generation of a start condition.

- The level on the SDAn line falls (from the high level to the low level) and the SCLn line is kept in the released state.
- The hold time for the start condition is secured as half of a bit period at the bit rate determined by the setting of the CCR2.BRR.
- The level on the SCLn line falls (from the high level to the low level), the ICR.IICSTAREQ is set (to 0), and a start condition generated STI interrupt (SCIn_TEI) is output.

Writing 1 to ICR.IICRSTAREQ causes the generation of a restart condition. The following operations are done at the generation of a restart condition.

- The SDAn line is released and the SCLn line is kept at the low level.
- The period at low level for the SCLn line is secured as half of a bit period at the bit rate determined by the setting of the CCR2.BRR.
- The SCLn line is released (transition from the low to the high level).
- Once the high level on the SCLn line is detected, the setup time for the restart condition is secured as half of a bit period at the bit rate determined by the setting of the CCR2.BRR.
- The level on the SDAn line falls (from the high level to the low level).
- The hold time for the restart condition is secured as half of a bit period at the bit rate determined by the setting of the CCR2.BRR.
- The level on the SCLn line falls (from the high level to the low level), the ICR.IICRSTAREQ is set (to 0), and a restart condition generated STI interrupt (SCIn_TEI) is output.

Writing 1 to ICR.IICSTPREQ causes the generation of a stop condition. The following operations are done at the generation of a stop condition.

- The level on the SDAn line falls (from the high level to the low level) and the SCLn line is kept at the low level.
- The period at low level for the SCLn line is secured as half of a bit period at the bit rate determined by the setting of the CCR2.BRR.
- The SCLn line is released (transition from the low to the high level).
- Once the high level on the SCLn line is detected, the setup time for the stop condition is secured as half of a bit period at the bit rate determined by the setting of the CCR2.BRR.
- The SDAn is released (transition from the low to the high level), the ICR.IICSTPREQ is set (to 0), and a stop condition generated STI interrupt (SCIn_TEI) is output.

Figure 31.57 shows the timing of operations in the generation of start, restart and stop conditions.

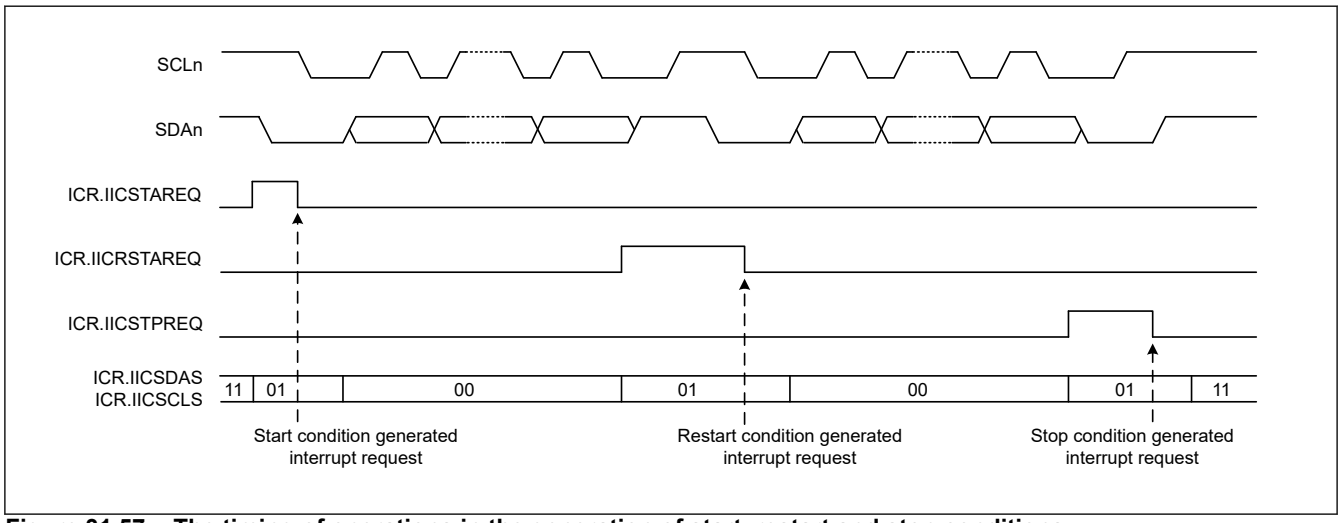


Figure 31.57 The timing of operations in the generation of start, restart and stop conditions

31.6.2 Clock Synchronization

The slave device of the communication partner may make SCLn line Low-level with a view to insert a wait. Setting the ICR.IICCSC to 1, applies control to obtain synchronization when the levels of the internal SCLn clock signal and the level being input on the SCLn pin differ.

When the ICR.IICCSC is set to 1, the level of the internal SCLn clock signal changes from low to high, counting to determine the period at high level is stopped while the low level is being input on the SCLn pin, and counting to determine the period at high level starts after the transition of the input on the SCLn pin to the high level. The interval until counting to determine the period at high level starts on the transition of the SCLn pin to the high level is the total time which contains the SCLn input delay, the noise filtering delay of the SCLn pin (2 or 3 cycles of the filtering clock), and the internal processing delay (1 or 2 cycles of PCLKSCIn). The period at high level of the internal SCLn clock is extended even if other devices are not placing the low level on the SCLn line.

If the ICR.IICCSC is 1, synchronization is obtained for the transmission and reception of data by taking the logical AND of the input on the SCLn pin and the internal SCLn clock. If the ICR.IICCSC is 0, synchronization with the internal SCLn clock is obtained for the transmission and reception of data.

If a slave device inserts a period of waiting into the interval until the transition of the internal SCLn clock signal from the low to the high level after a request for the generation of a start, restart, or stop condition is issued, the time until generation is prolonged by that period.

If a slave device inserts a period of waiting after the transition of the internal SCLn clock signal from the low to the high level, although the generation-completed interrupt is issued without stopping the period of waiting, generation of the condition itself is not guaranteed.

Figure 31.58 shows an example of operations to synchronize the clocks.

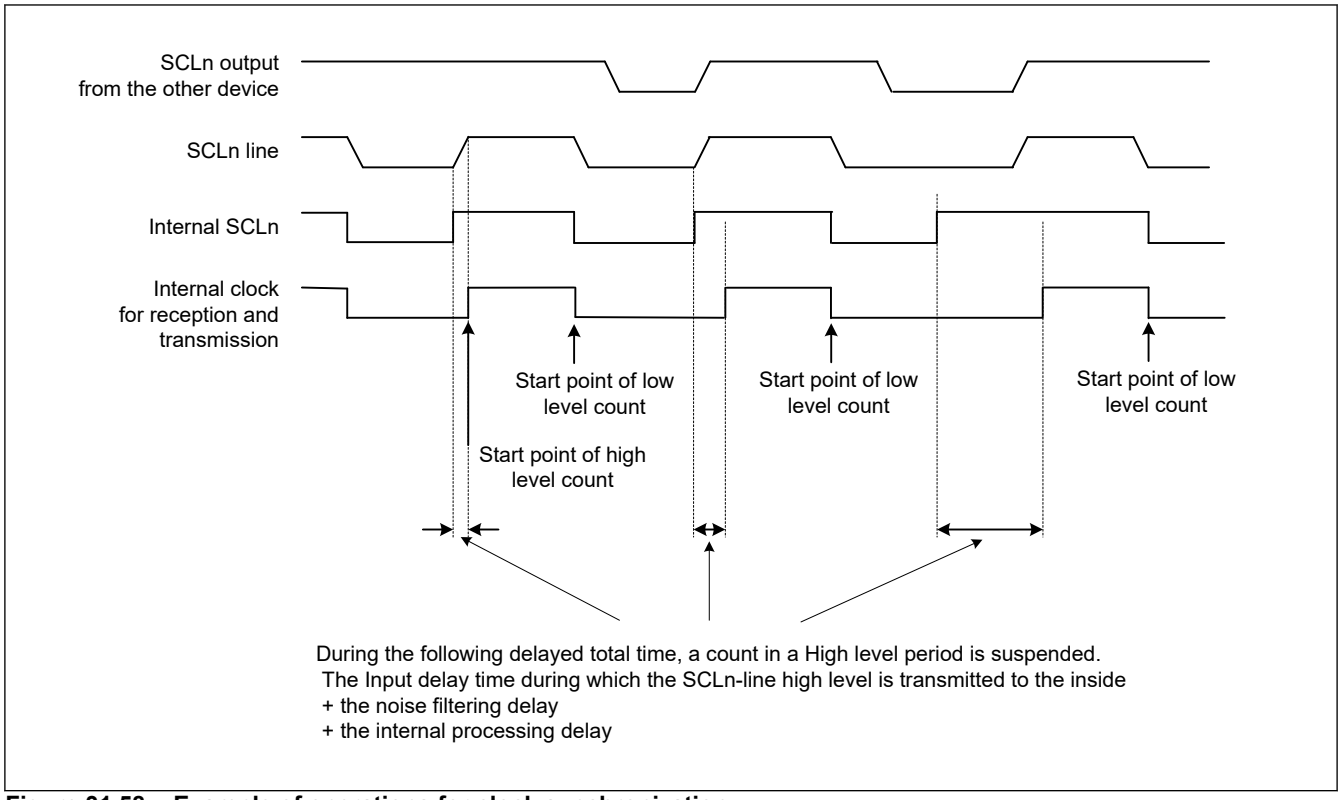


Figure 31.58 Example of operations for clock synchronization

31.6.3 SDA Output Delay

The ICR.IICDL[4:0] can be used to set a delay for output on the SDAn pin relative to falling edges of output on the SCLn pin. The delay-time settings are selectable from 0 to 31 cycles of the clock signal from the baud rate generator (The base is PCLKSCIn and selected the divided clock by the CCR2.CKS[1:0]). About Start/Restart/Stop conditions, 8-bit transmission data and acknowledge, the SDAn pin output can be delayed.

If the SDA output delay is shorter than the falling time of the SCLn output pin, the change of the SDAn output pin will start while the SCLn output pin level is falling, then there is a possibility of erroneous operation for slave devices. Ensure setting the SDA output delay greater than the SCLn maximum falling time (300 ns for I2C normal/fast mode).

Figure 31.59 shows the timing of SDA output delay.

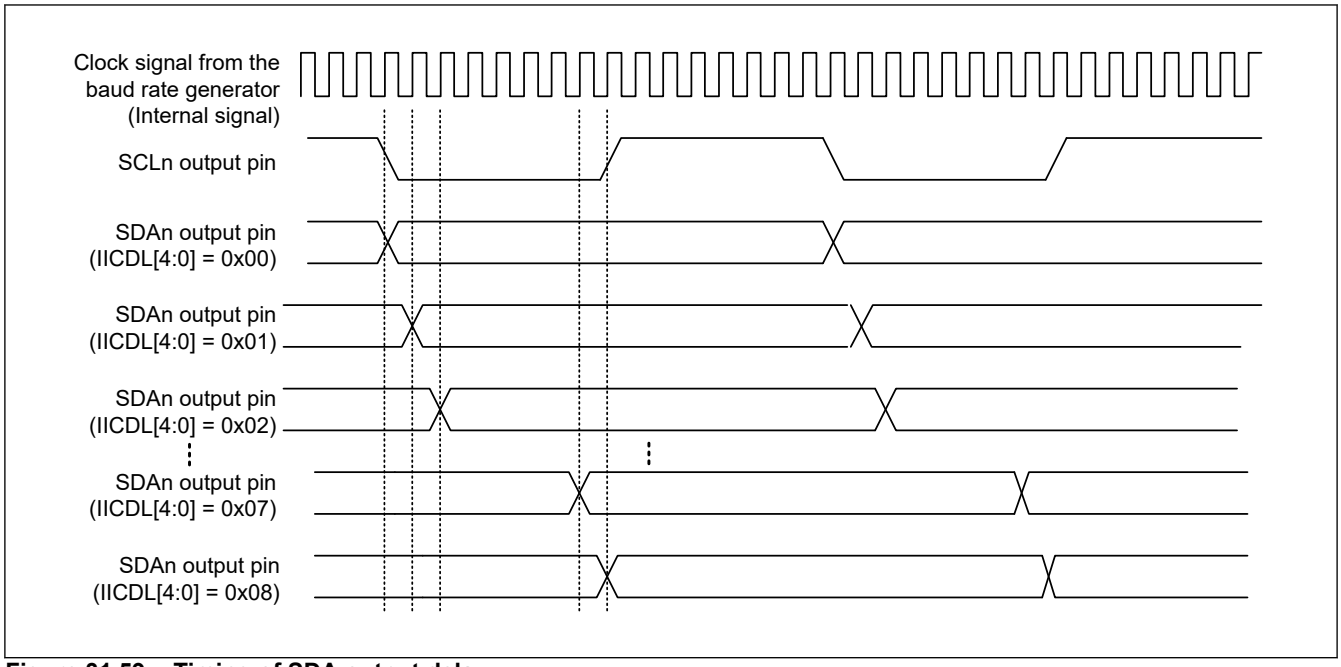
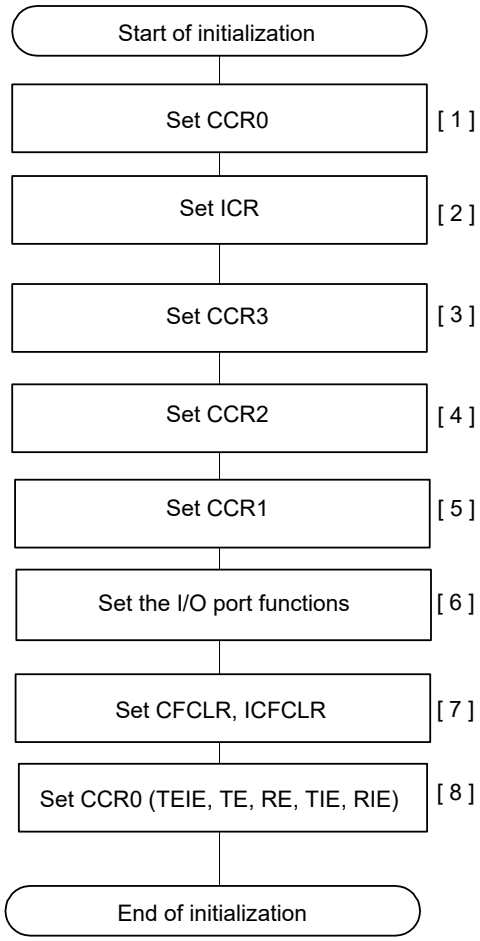


Figure 31.59 Timing of SDA output delay

31.6.4 SCI Initialization (Simple I2C mode)

Write initial value (all 0) to CCR0, then initialize SCI according to Figure 31.60.

When changing the operating mode, transfer format, and so on, be sure to set 0 to CCR0.TE and CCR0.RE before proceeding with the changes. (Or write initial value to CCR0 again.)



Note 1. If you do not use the bit rate modulation function, you do not need to set it.

Note: Set the CCR0.TE and RE bits to 0 or 1 at the same time.

[1] Set the following bits of CCR0 to 0.
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set the IICSDAS[1:0] and IICSCLS[1:0] to 11b.
Set the IICDL[4:0] and IICINTM as required.
Set the IICACKT and the IICCSC bits to 1.

[3] Set the transmission/reception format as the communication mode (MOD[2:0] = 100b) and CCR3.LSBF = 0.

[4] Set the bit rate modulation function*1, the clock selection, and the bit rate.

[5] Set up the Noise-filter function.

[6] Set I/O port settings that allow use of the SCLn and SDAn pin functions.

[7] Write 1 to the following bits and clear the corresponding flag.
CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
ICFCLR.IICSTIFC

[8] Set the TE and RE to 1.
To enable interrupts, set the TE bit, TIE bit, TEIE bit, RE bit, and RIE bit to 1 with one instruction at the same time.
(for transmission and when the IICINTM bit is 1, clear the RIE bit.)
Setting the TE and RE bits to 1 makes the SCLn and SDAn pin functions available.

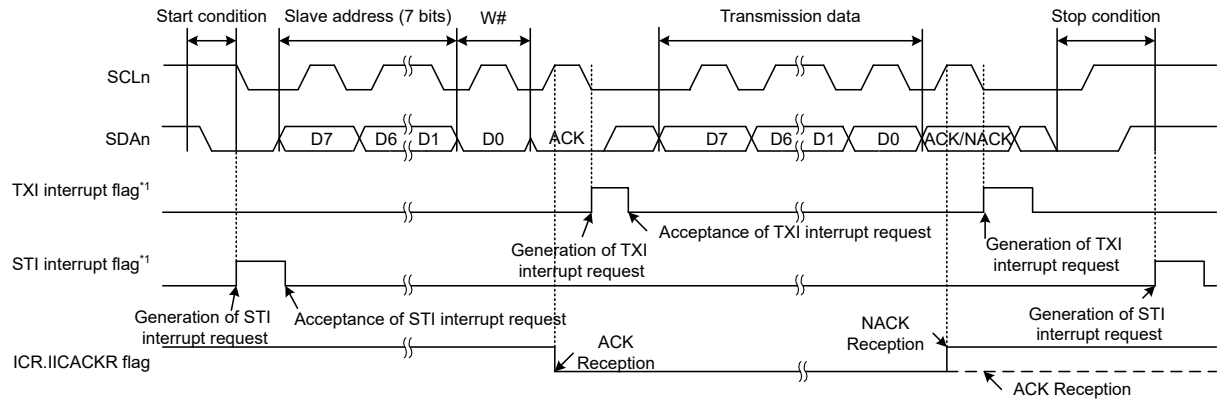
Figure 31.60 Example of the flow of SCI initialization (for simple I2C mode)

31.6.5 Operation in Master Transmission (Simple I2C mode)

Figure 31.61 and Figure 31.62 show examples of operations in master transmission, Figure 31.63, Figure 31.64, and Figure 31.65 show the example flowcharts. See Table 31.33 about the STI interrupt.

Figure 31.61 shows the operation example when ICR.IICINTM is 1 (Reception/Transmission interrupts are in use). In this case, you can start DMAC by TXI interrupt. However, if use DMAC, ACK/NACK can not be confirmed. So, if you want to confirm ACK/NACK, prepare transmit data by CPU. In simple I2C mode, TXI interrupt is generated when communication of one frame is completed. And it is not used RXI interrupt in master transmission, so the CCR0.RIE set to 0.

Figure 31.63 shows a flow chart in the case of ICR.IICINTM is 1 and address transmission by CPU and data transmission by DMAC. Figure 31.64 shows a flow chart of address and data transmission by CPU. When 10-bit slave addresses are in use, steps [3] and [4] are repeated twice.



Note 1. See section 11, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

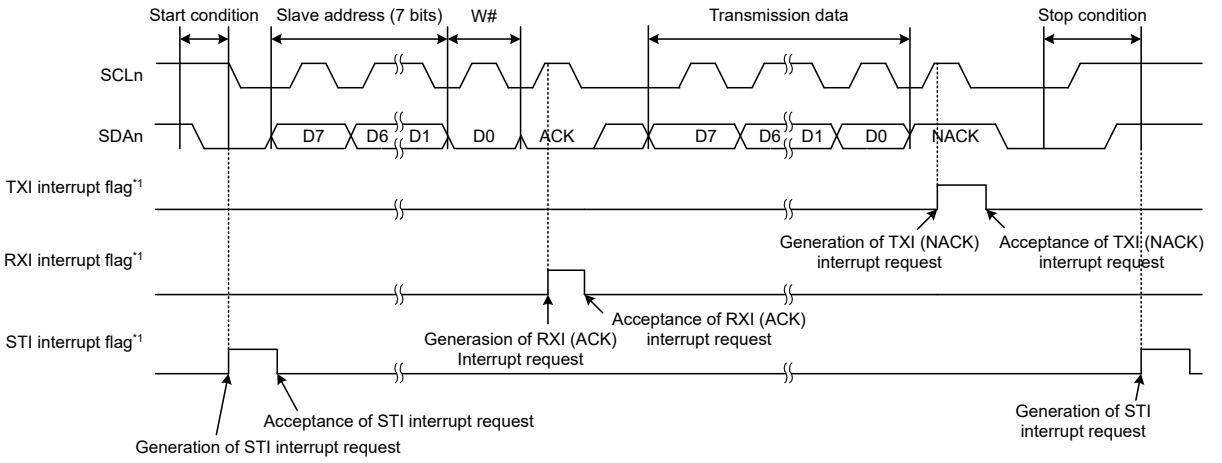
Figure 31.61 Example 1 of operations for master transmission in simple I2C bus mode (7-bit slave address, transmission and reception interrupt in use (ICR.IICINTM = 1))

Figure 31.62 shows an example of operations when ICR.IICINTM is 0 (ACK and NACK interrupt in use). In this case, DMAC is activated by the ACK (RXI) interrupt, and necessary number of data bytes are transmitted. When the NACK is received, error processing, such as transmission stop and retransmission, is performed by the NACK (TXI) interrupt as the trigger.

Figure 31.65 shows a flow chart of ICR.IICINTM is 0.

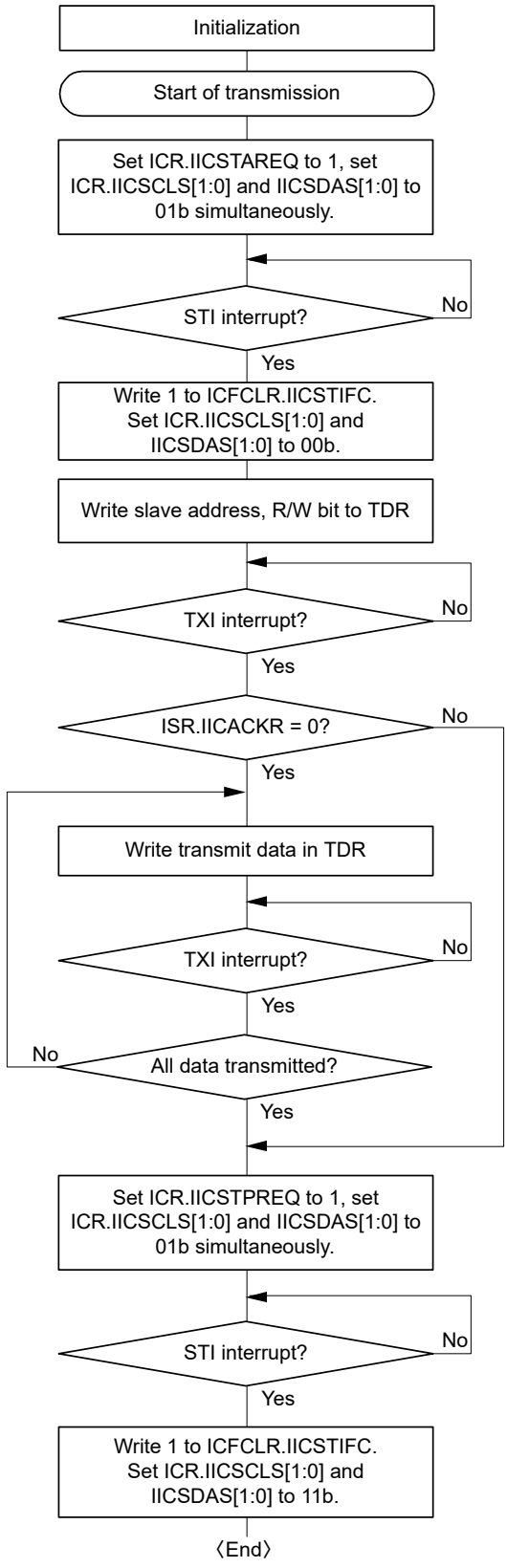
To resume communication after interrupting communication for some reason after writing transmit data to TDR, follow the procedure below.

1. Set the CCR0.TE and CCR0.RE bits to 0 to stop communication.
2. Set ICR.IICSCLS[1:0] and ICR.IICSDAS[1:0] to 11b, release the I2C bus, and clear various condition generation requests.
3. When CSR.RDRF = 1, the RDR register is read by dummy and the RDRF bit is set to 0.
4. Set the CCR0.TE and CCR0.RE bits to 1 and restart communication.



Note 1. See section 11, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 31.62 Example 2 of operations for master transmission in simple I2C bus mode (7-bit slave address, ACK and NACK interrupt in use (ICR.IICINTM = 0))



[1] Initialization for simple I2C mode
For transmission, set the CCR0.RIE to 0.
(RXI and ERI interrupts requests are disabled)

[2] Generate a start condition.

[3] Writing to TDR:
Writing the slave address and value for the R/W bit to TDR.

[4] Confirm ACK response from the slave device:
Check the ISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

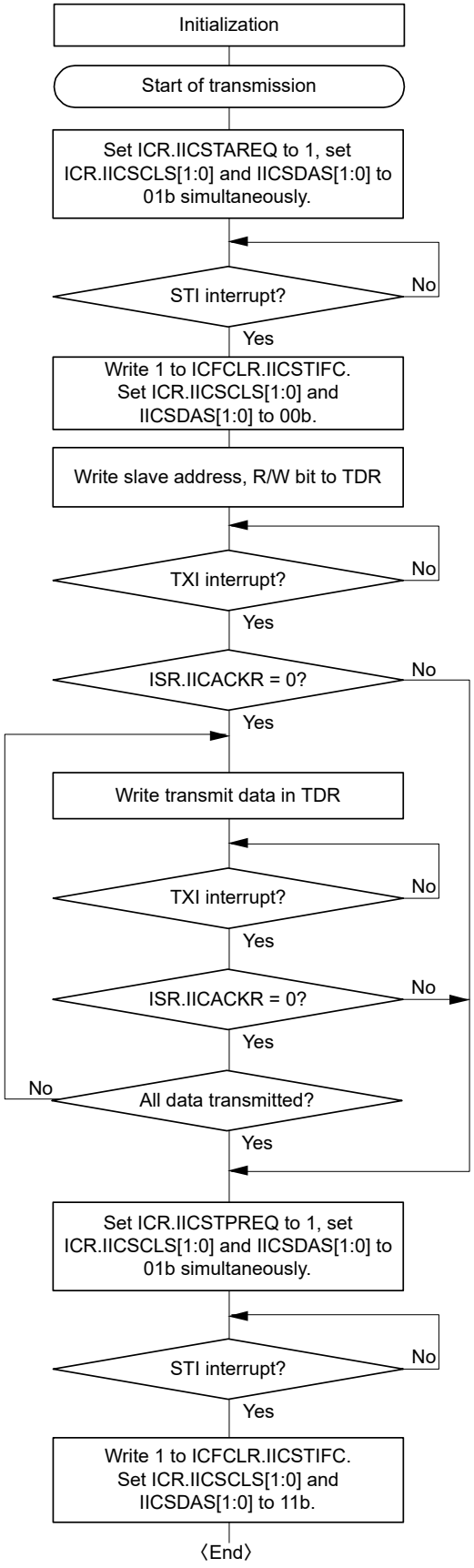
[5] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to TDR. Except for the first data to be transmitted, a TXI interrupt request can activate the DMAC to handle writing of data to TDR.
But it cannot be confirmed status of ACK or NACK in this case.

[6] Generation of a stop condition.

If 10-bits slave address are in use, processing from [3] to [4] is repeated twice.

Note: In simple I2C mode, the TXI interrupt request is generated when communication is completed.

Figure 31.63 Example of the procedure for master transmission operations in simple I2C mode (when ICR.IICINTM is 1, and when confirming ACK / NACK by address transmission only.)



[1]

[2]

[3]

[4]

[5]

[6]

[1] Initialization for simple I2C mode
For transmission, set the CCR0.RIE to 0.
(RXI and ERI interrupts requests are disabled)

[2] Generate a start condition.

[3] Writing to TDR:
Writing the slave address and value for the R/W bit to TDR.

[4] Confirm ACK response from the slave device:
Check the ISR.IICACKR bit. If its value is 0, it is indicated that the slave device responded with ACK and operations proceed. If its value is 1, it is indicated that there was no response from a slave device so the next transition is to generation of the stop condition.

[5] Steps for continuing with serial transmission:
When transmission is to continue, write further transmit data to TDR. If ACK or NACK is checked in every transmission, DMAC cannot be used.

[6] Generation of a stop condition.

If 10-bits slave address are in use, processing from [3] to [4] is repeated twice.

Note: In simple I2C mode, the TXI interrupt request is generated when communication is completed.

Figure 31.64 Example of the procedure for master transmission operations in simple I2C mode (when ICR.IICINTM is 1, and when confirming ACK / NACK in all transmissions.)

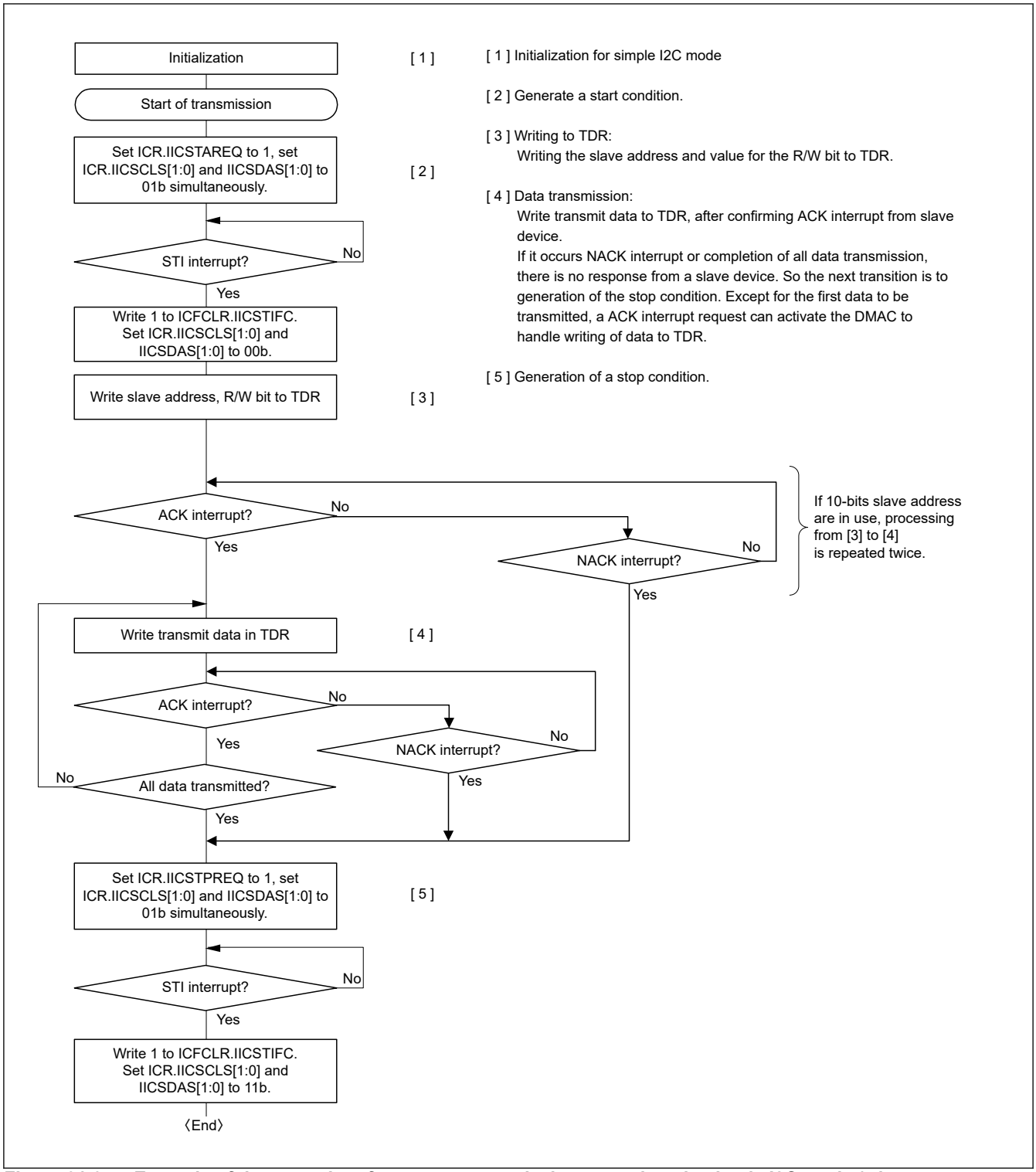
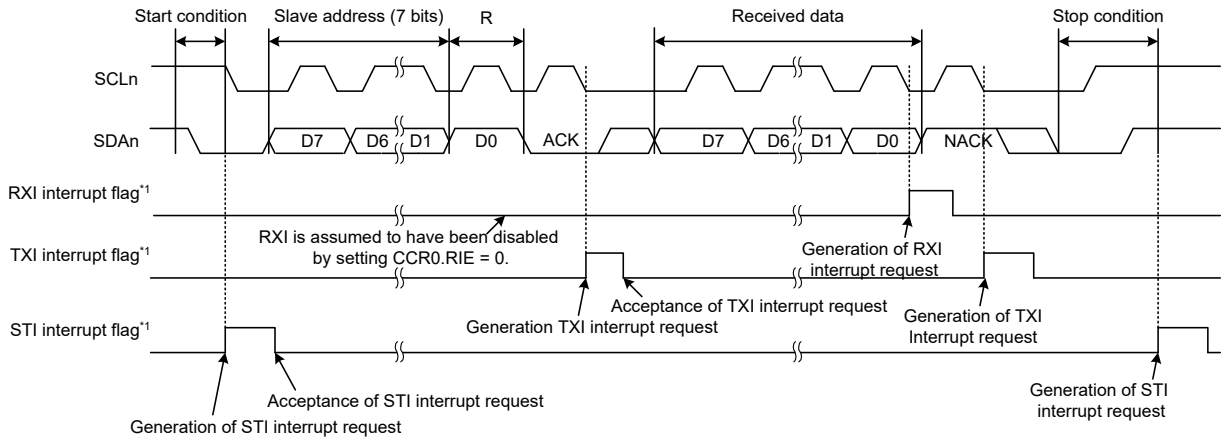


Figure 31.65 Example of the procedure for master transmission operations in simple I2C mode (when ICR.IICINTM is 0.)

31.6.6 Master Reception (Simple I2C mode)

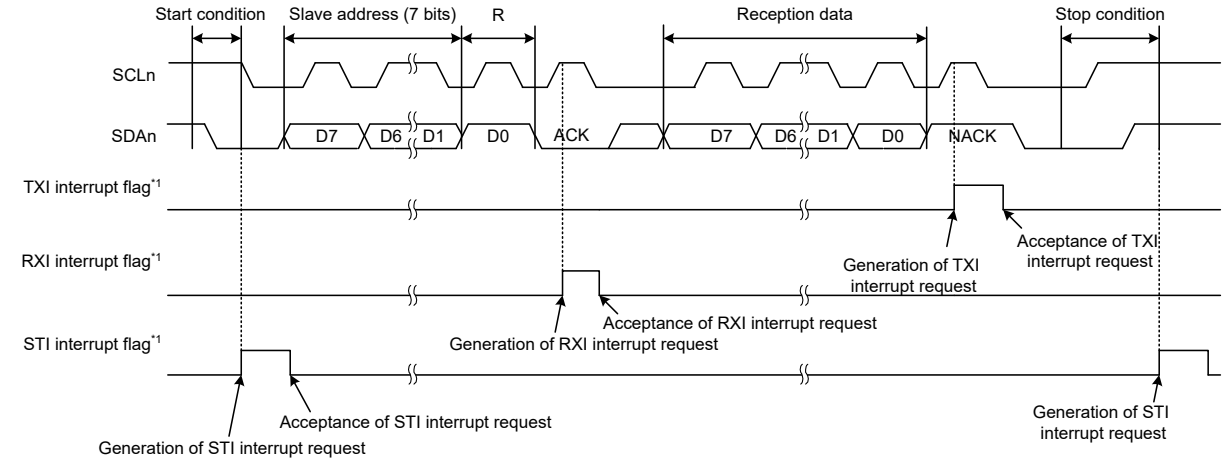
Figure 31.66 and Figure 31.67 show example of operations in simple I2C mode master reception. Figure 31.68 and Figure 31.69 show flowchart of the master reception. The value of the ICR.IICINTM bit is assumed to be 1 (use reception and transmission interrupts) and 0 (use ACK and NACK interrupts).

In simple I2C mode, the transmit data empty interrupt (TXI) is generated when communication of one frame is completed.



Note 1. See section 11, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 31.66 Example of operations for master reception in simple I2C bus mode (7-bit slave address, transmission and reception interrupt in use (ICR.IICINTM = 1))



Note 1. See section 11, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 31.67 Example of operations for master reception in simple I2C bus mode (7-bit slave address, ACK and NACK interrupt in use (ICR.IICINTM = 0))

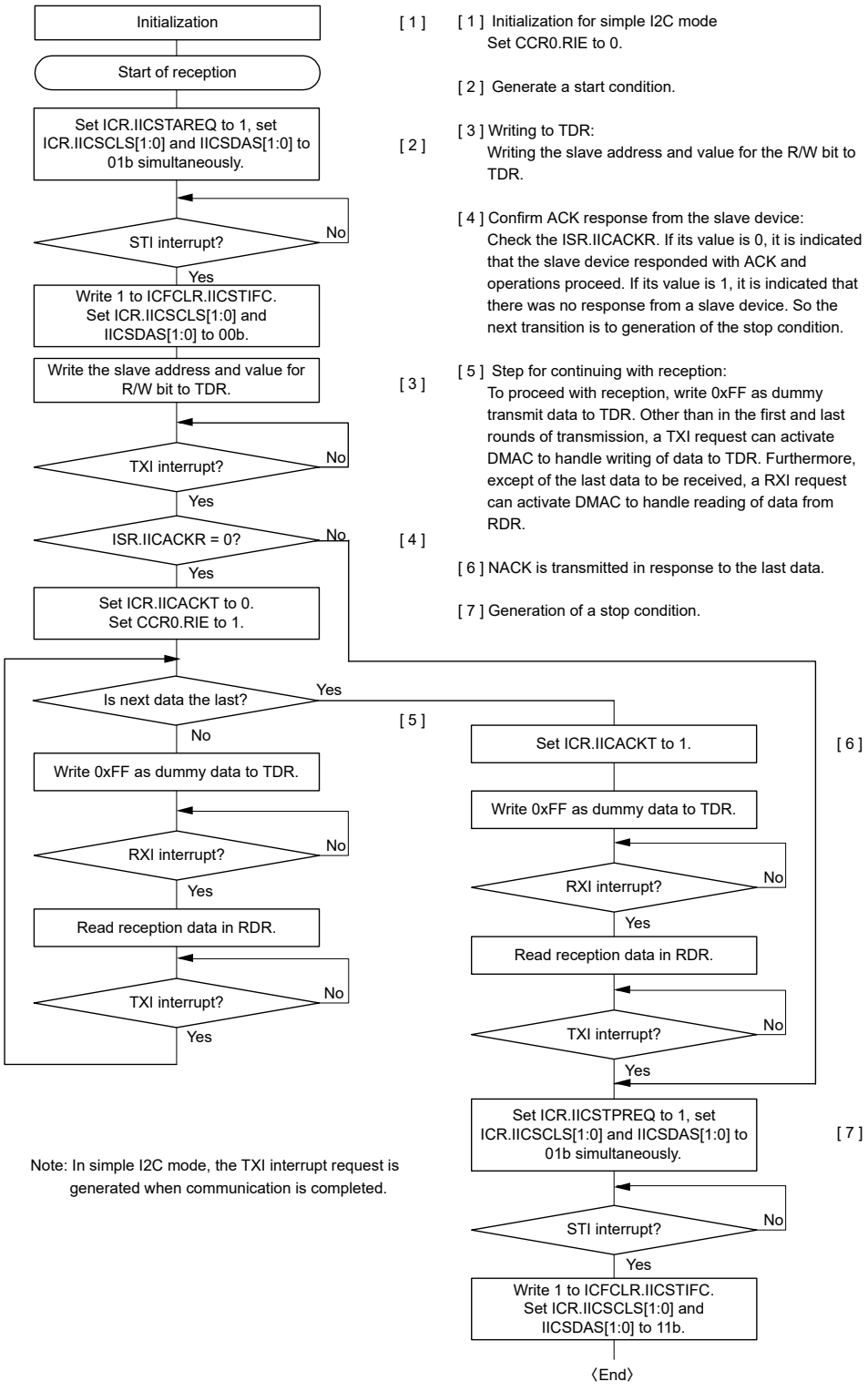


Figure 31.68 Example of the procedure for master reception operations in simple I2C mode (when ICR.IICINTM is 1, and transmission interrupts and reception interrupts are in use.)

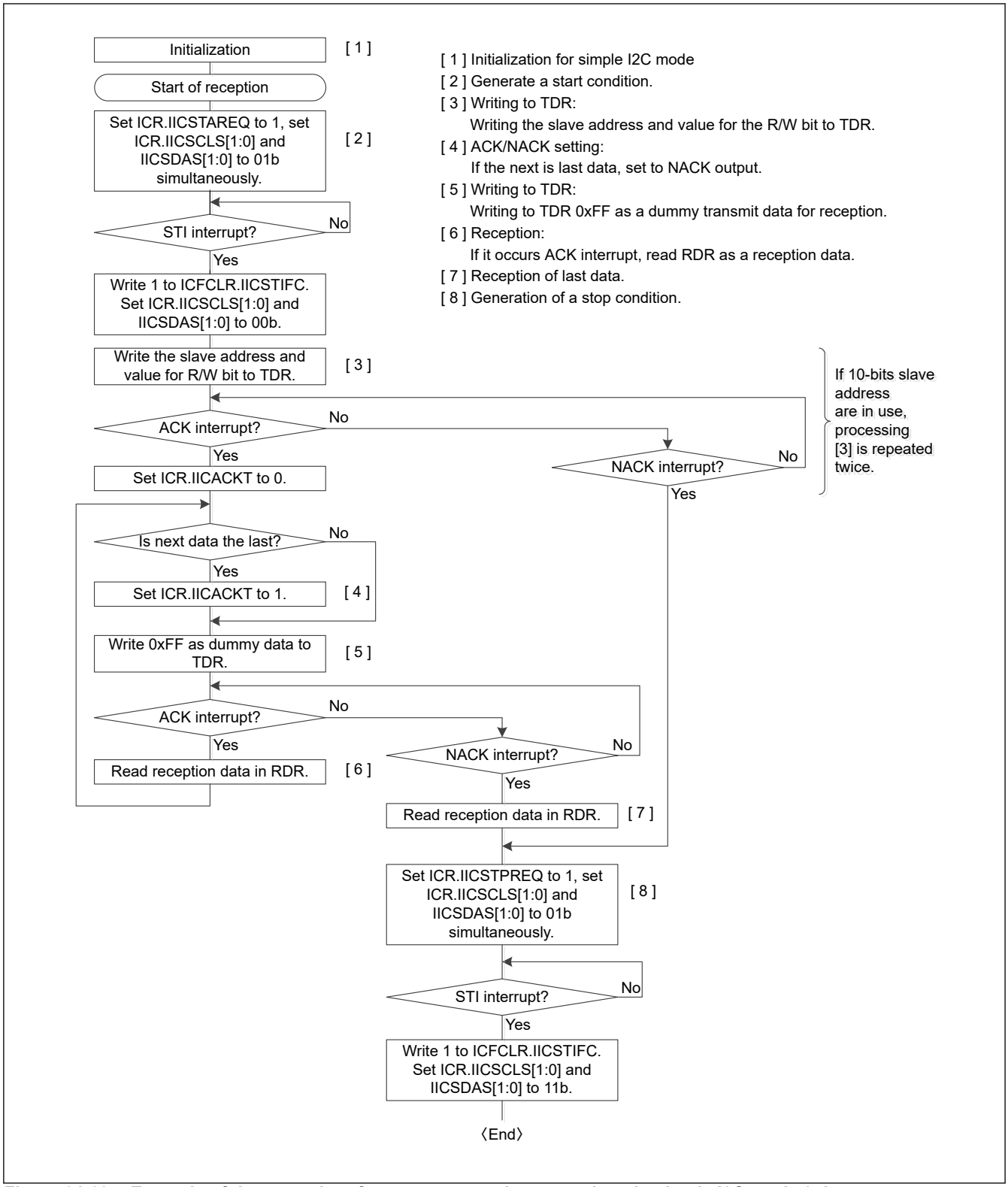


Figure 31.69 Example of the procedure for master reception operations in simple I2C mode (when ICR.IICINTM is 0, and ACK interrupts and NACK interrupts are in use.)

31.7 Clock Synchronous Mode

Figure 31.70 shows the communication data format of clock synchronous serial communication.

In clock synchronous mode, data is transmitted and received in synchronization with clock pulses. A communication data character consists of 8-bit data. In clock synchronous mode, no parity bit can be added. In data transmission when CPHA = 1 and CPOL = 1, the SCI outputs data from the falling edge of the sync clock until the next falling edge. In data reception,

data is read at the rising edges of the sync clock. After 8-bit data is output, the communication line holds the final-bit output state. In slave communication when CPHA = 1, however, the communication line holds the first-bit output state.

Because the SCI has an internal transmitter and a receiver independently, SCI enable full-duplex communication by sharing a communication clock of the transmitter and the receiver. Furthermore, because both the transmitter and the receiver have a double- buffer structure, continuous transmission and reception are possible by writing the next transmit data during transmission and reading the previous receive data during reception.

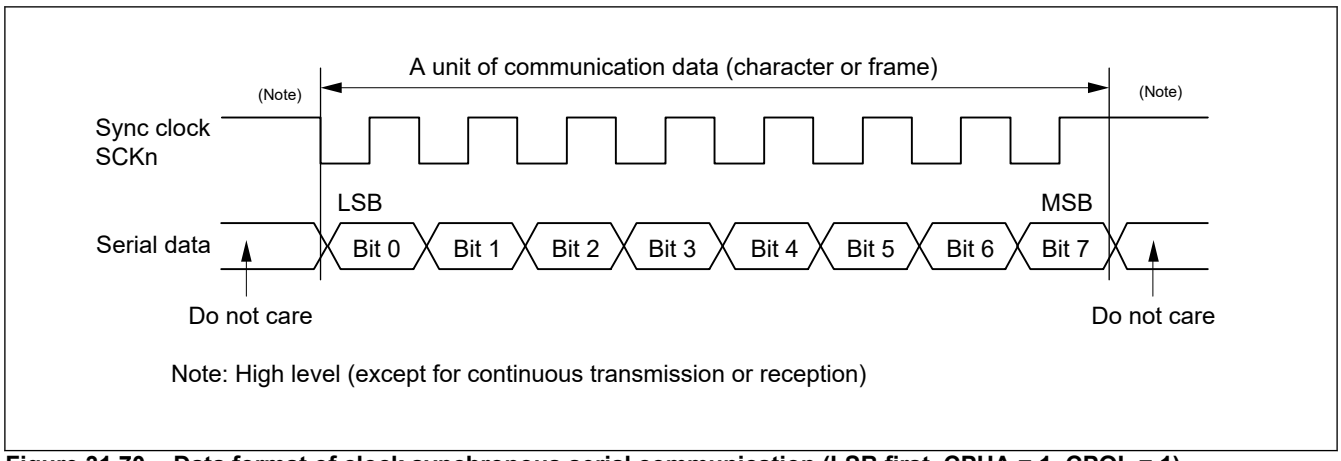


Figure 31.70 Data format of clock synchronous serial communication (LSB first, CPHA = 1, CPOL = 1)

31.7.1 Clock

(1) When the internal clock is selected

When the CCR3.CKE[1:0] bits are set to 00b or 01b (master mode), the internal clock generated by the baud rate generator can be selected and the sync clock is output from the SCKn pin. Eight pulses of the sync clock are output during single-character transmission/reception. The sync clock remains at a high level*1 while no transmission or reception is performed. In transmission-only or transmission/reception, the sync clock is not output unless transmit data is prepared.

When the internal clock is selected, the clock with a delay from the SCKn signal is used for the master reception sampling clock. This ensures the data setup time and hold time for high-speed communication.

Note 1. When CCR3.CPOL = 1, the sync clock stops at a high level. When CCR3.CPOL = 0, the sync clock stops at a low level.

(2) When the external clock is selected

When the CCR3.CKE[1:0] bits are set to 10b or 11b (slave mode), data is transmitted and received using the external clock that is input from the SCKn pin.

31.7.2 CTS and RTS Functions

The CTS function performs transmission/reception and controls transmission start using the CTSn# pin input when the internal clock is selected. Setting the CCR1.CTSE bit to 1 enables the CTS function. Two settings are possible for the CTS/RTS pin: dual setting for using either function with a single pin, and dedicated setting for using each function simultaneously with two pins. The CCR1.CTSPEN register is used for these settings.

When the CTS function is enabled, transmission/reception and transmission start only when the CTSn# pin input level is low. Even if the CTSn# pin input becomes high level during transmission/reception or transmission operation, frames that are being transmitted/received or being transmitted are not affected and transmission/reception or transmission operation continues.

The RTS function makes a serial communication start request using the RTSn# pin output when the external sync clock is selected. When serial communication is enabled, the RTSn# pin outputs a low level. A low level and a high level are output under the following conditions.

[Conditions for low level]

(1) When a non-FIFO buffer is selected, all the following conditions are met.

- CCR0.RE bit or TE bit = 1
- Serial communication is enabled.
- No receive data is present before reading. (when CCR0.RE bit = 1)
- Data before transmission start is remaining in the TSR register. (when CCR0.TE bit = 1)
- CSR.ORER flag = 0

(2) When a FIFO buffer is selected, all the following conditions are met.

- CCR0.RE bit or TE bit = 1
- Serial communication is enabled
- The number of receive data stored in the receive FIFO (RDR register) is less than the value set in FCR.RSTRG[4:0]. (when CCR0.RE bit = 1)
- Unsent data is remaining in the transmit FIFO (TDR register). (when CCR0.TE bit = 1 and CCR3.CKE[1] bit = 0)
- Data before transmission start is remaining in the TSR register. (when CCR0.TE bit = 1 and CCR3.CKE[1] bit = 1)
- ORER (RDR.ORER) flag = 0

[Conditions for high level]

(1) When a non-FIFO buffer is selected

If conditions for low level are not met:

When CCR0.RE is set to 0 without reading the RDR register to terminate reception after reception is complete, the RTSn# pin output level remains high. At this time, write 0 to CCR0.RE.

(2) When a FIFO buffer is selected

If conditions for low level are not met:

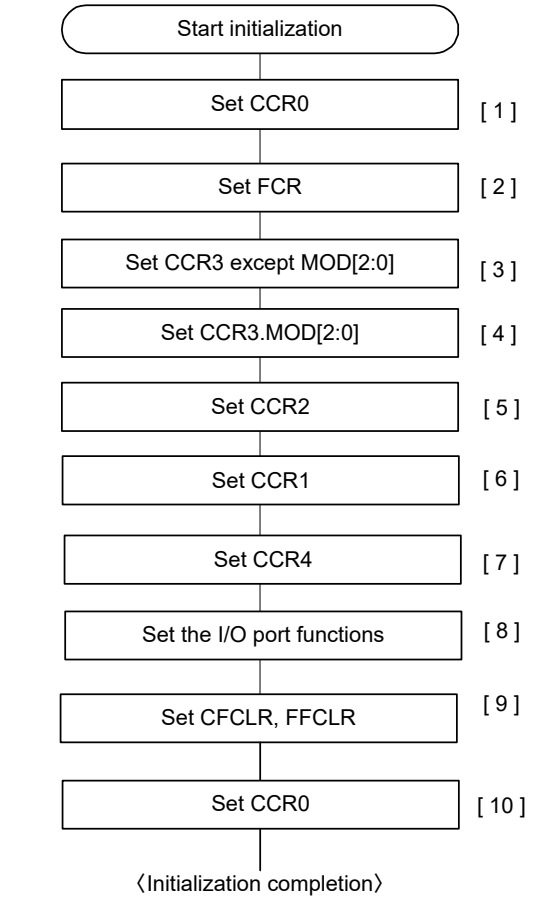
31.7.3 Initializing the SCI (Clock Synchronous Mode)

Before starting data transmission/reception, write 0 to CCR0.TE and CCR0.RE (or write initial values to the CCR0 register) and initialize the SCI according to the flowchart example in Figure 31.71.

Before changing operating mode or communication format, also be sure to write 0 to CCR0.TE and CCR0.RE.

Note that writing 0 to the RE bit does not initialize the ORER, FER, PER, and RDRF flags in CSR and the RDR register. Also note that writing 0 to the TE bit does not initialize the TEND flag when a FIFO buffer is selected. Attention is also needed for changing operating mode.

When CCR0.TIE = 1, note that setting the TE bit to 1 from 0 generates a TXI interrupt.



[1] Set the following bits of CCR0 to 0[*1].
TEIE, TIE, RIE, TE, RE
If you have not changed from the initial settings, you can skip this step.

[2] Set the TFRST and RFRST to 1 to empty FIFO.
Set the TTRG[4:0], RTRG[4:0], and RSTRG[4:0] bits.

[3] Set CCR3 except of communication mode.
FIFO use / no-use
Transmission / reception format
Clock setting
Leave unused bits (CHR[1:0], STP, RXDESEL, MP, DE, ACS0,GM,BLK) at their initial values.

[4] Set communication mode (MOD[2:0] = 010b)[*2].

[5] Set up the bit-rate-modulation function[*3 *4], select clock, set bit rate[*4]. Leave unused bits (BCP[2:0], ABCS, ABCSE) at their initial values.

[6] Set up the Noise-filter function, the loop-back function, communication terminal status, the parity check function, and the CTS/RTS function.

[7] Set up the adjust sampling timing function.
Leave unused bits (CMPD[8:0], ATEN, ATT[2:0], AET) at their initial values.

[8] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, SCKn, CTS#, and RTSn# pins.

[9] Write 1 to the following bits and clear the corresponding flag.
CFCLR.RDRFC, FERC, PERC, MFFC, ORERC, DFERC, DPERC, DCMFC, ERSC
FFCLR.DRC

[10] Set the TE or RE bit to 1.[*1 *5] To enable interrupts, set the TE bit and TIE bit, and the RE bit and RIE bit to 1 with one instruction at the same time.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Note 1. In simultaneous transmit and receive operations, the TE and RE bits in CCR0 should both be to 0 or set to 1 simultaneously.

Note 2. Set CPOL and CPHA before setting the Communication mode.

Note 3. If you do not use the bit rate modulation function, you do not need to set it.

Note 4. If you use an external clock, you do not need to set it.

Note 5. When using the internal clock (master), the setting of reception only is prohibited.

Figure 31.71 SCI initialization flowchart example (clock synchronous mode)

31.7.4 Serial Data Transmission (Clock Synchronous Mode)

(1) Non-FIFO selected

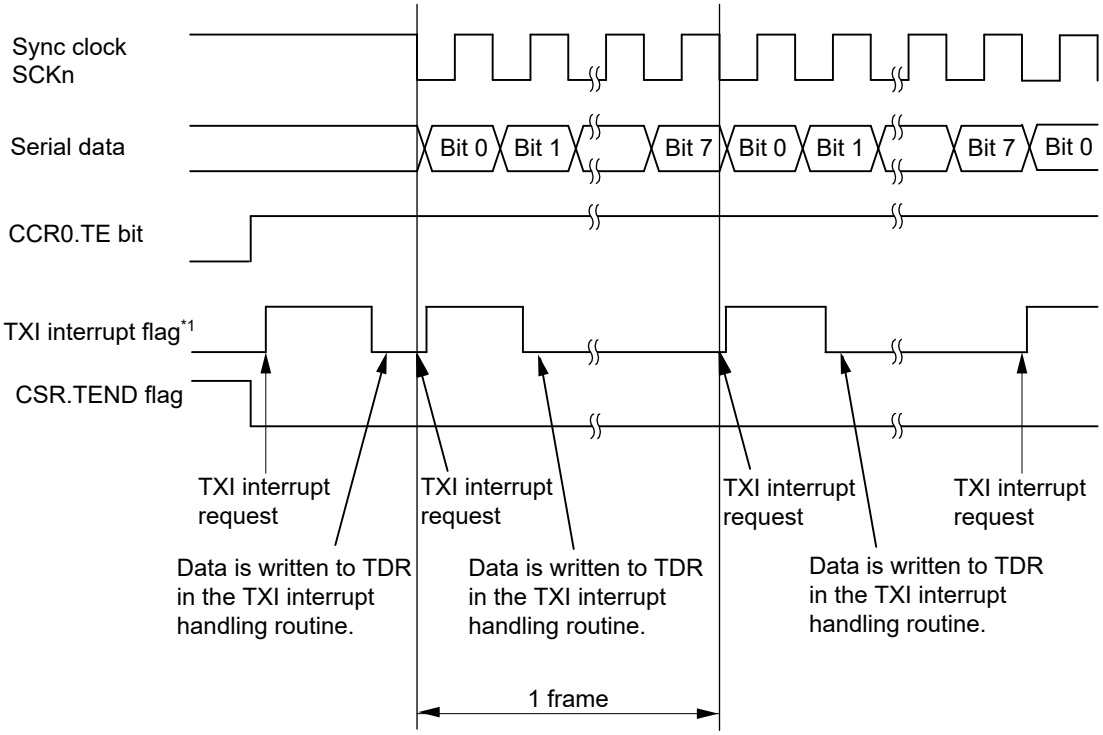
Figure 31.72, Figure 31.73, and Figure 31.74 show operation examples of serial transmission in clock synchronous mode. The SCI operates as follows during serial data transmission.

1. When data is written to the TDR register in the TXI interrupt routine, the SCI transfers the data from the TDR register to the TSR register. When starting data transmission, set the CCR0.TIE bit and the CCR0.TE bit to 1 simultaneously by a single instruction. Then a TXI interrupt request is generated.
2. The written data is transferred from the TDR register to the TSR register, which starts transmission. When the CCR0.TIE bit is set to 1 at this time, a TXI interrupt request is generated. Writing the next transmit data to the TDR register before transmission of data transferred previously in the TXI interrupt routine is complete enables continuous transmission. When a TEI interrupt request is used, after the final transmit data is written to the TDR register in the TXI

interrupt request processing routine and the final data's transmission is started, set 0 to the CCR0.TIE bit and set 1 to the TEIE bit.

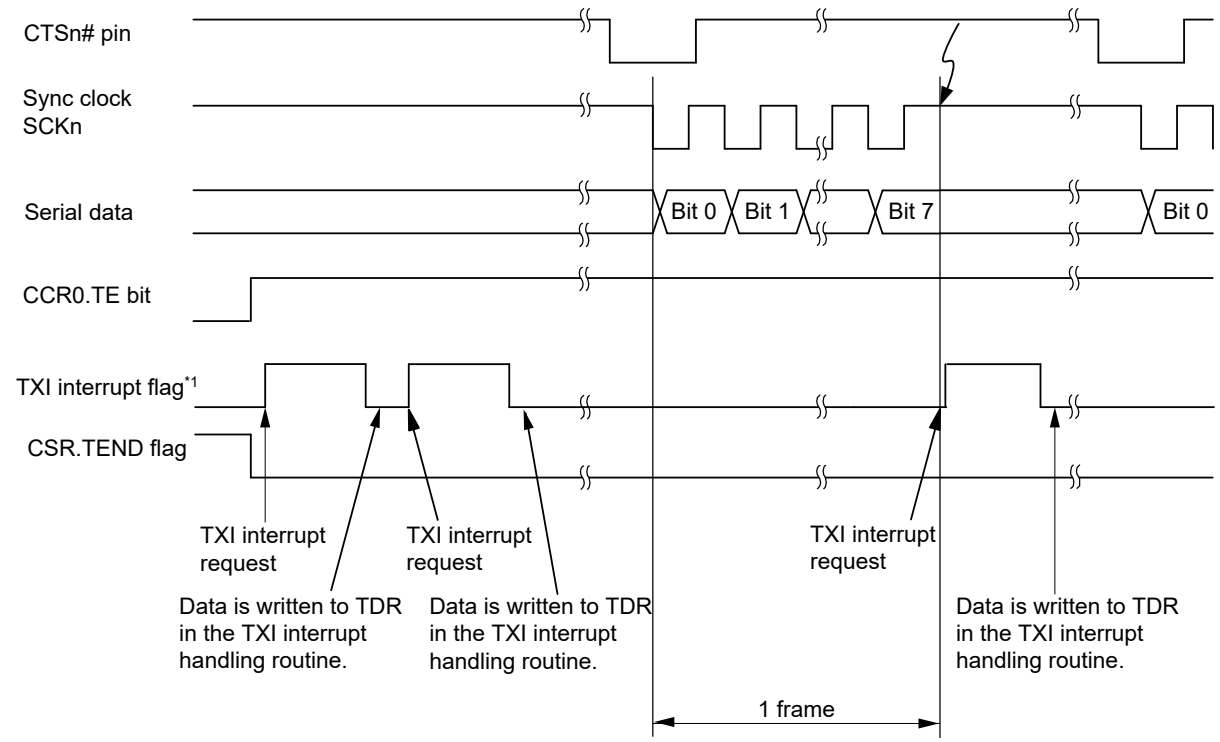
3. The TXDn pin outputs 8-bit data in synchronization with the output clock (in clock output mode) or with the input clock (when external clock is selected). When the CCR1.CTSE bit = 1 (CTS function enabled), the output clock starts after the CTS signal input becomes low level.
4. Update (data write) of the TDR register is checked at the final-bit transmission timing.
5. When the TDR register has been updated, data is transferred from the TDR register to the TSR register to start sending the next frame.
6. If the TDR register has not been updated, the CSR.TEND flag is set to 1 and the final-bit output state is retained. When the CCR0.TEIE bit is set to 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Figure 31.75 shows an example of data transmission flowchart.



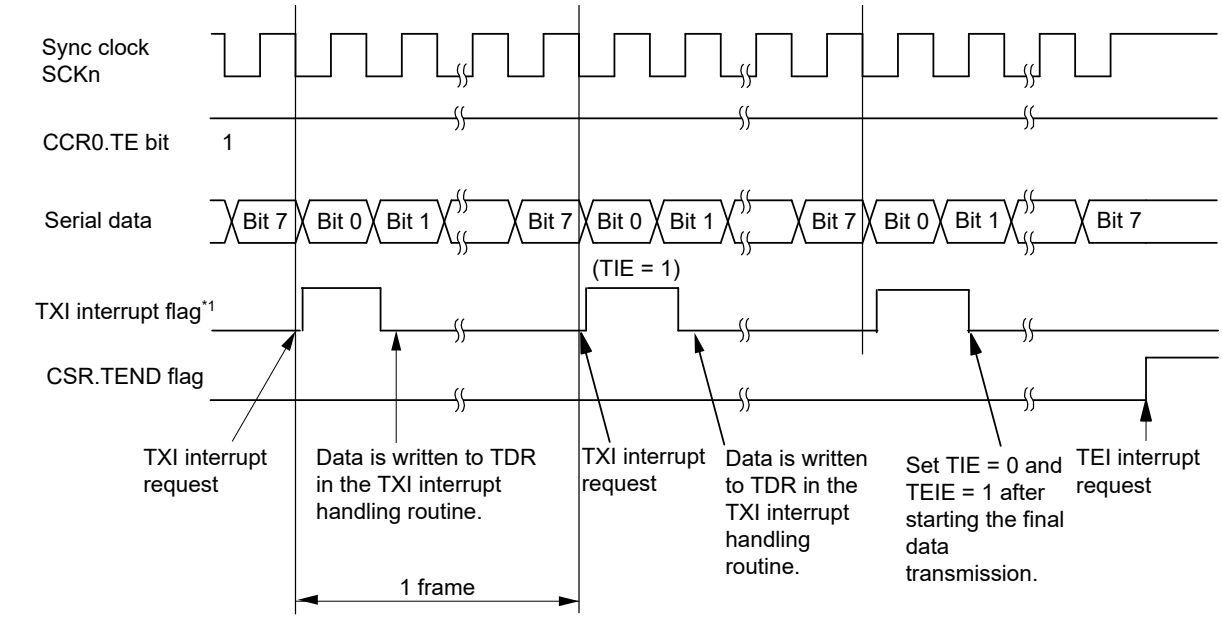
Note 1. For corresponding interrupt source numbers, see section 11, Interrupt Controller (ICU).

Figure 31.72 Serial transmission example in clock synchronous mode (1) (CTS function not used / transmission start / CPHA = 1, CPOL = 1)



Note 1. For corresponding interrupt source numbers, see section 11, Interrupt Controller (ICU).

Figure 31.73 Serial transmission example in clock synchronous mode (2) (CTS function used / transmission start / CPHA = 1, CPOL = 1)



Note 1. For corresponding interrupt source numbers, see section 11, Interrupt Controller (ICU).

Figure 31.74 Serial transmission example in clock synchronous mode (3) (during transmission to transmission end / CPHA = 1, CPOL = 1)

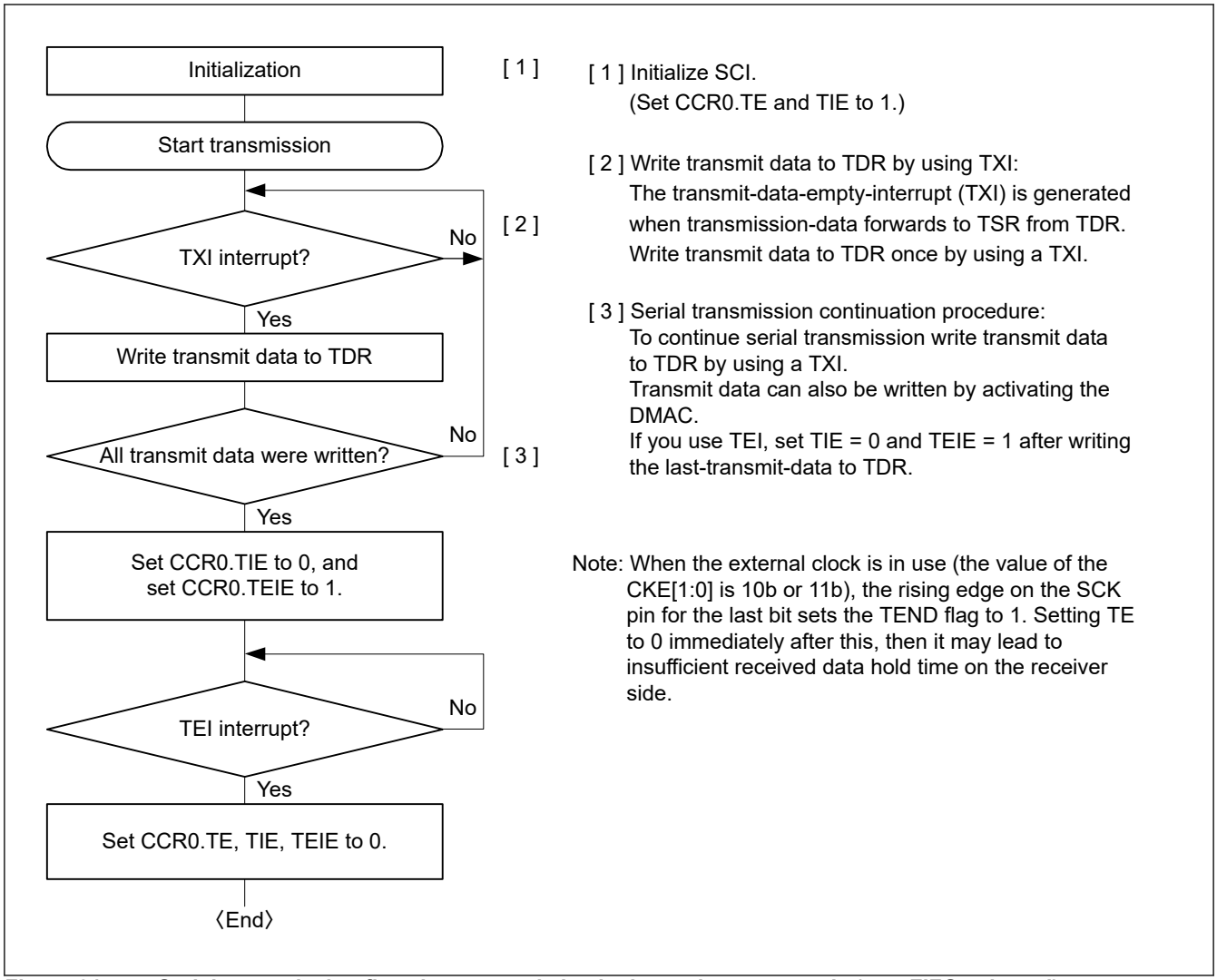


Figure 31.75 Serial transmission flowchart example in clock synchronous mode (non-FIFO selected)

(2) FIFO selected

Figure 31.76 shows an example of flowchart of serial transmission (a FIFO buffer selected) in clock synchronous mode. The SCI operates as follows when serial data transmission.

1. When data is written to the transmit FIFO (TDR register) in the TXI interrupt routine, the SCI transfers the data from the transmit FIFO (TDR register) to the TSR register. The number of writable transmit data is [N_{FIFO} (= 16) - number of unsent transmit data stored in the transmit FIFO (TDR register)]. In addition, when starting data transmission, set the CCR0.TIE bit and the CCR0.TE bit to 1 simultaneously by a single instruction. Then a TXI interrupt request is generated.
2. Data is transferred from the transmit FIFO (TDR register) to the TSR register and transmission starts. When the number of data stored in the transmit FIFO (TDR register) is equal to or less than the threshold value of the transmit FIFO, the CSR.TDRE flag is set to 1. When the CCR0.TIE bit is set to 1 at this time, a TXI interrupt request is generated. Writing the next transmit data to the transmit FIFO (TDR register) in the TXI interrupt routine before transmission of data written to the transmit FIFO (TDR register) is complete enables continuous transmission. When a TEI interrupt request is used, after the final transmit data is written to the transmit FIFO (TDR register) in the TXI interrupt request processing routine, set 0 to the CCR0.TIE bit and set 1 to the TEIE bit.
3. The TXDn pin outputs 8-bit data in synchronization with the output clock (in clock output mode) or with the input clock (when external clock is selected). When CCR1.CTSE = 1 (CTS function enabled), the output clock starts after the CTS signal input becomes low level.
4. The SCI checks whether unsent transmit data is remaining in the transmit FIFO (TDR register)[*1] at the final-bit transmission timing.

5. When data is remaining in the transmit FIFO (TDR register), the data is transferred from the transmit FIFO (TDR register) to the TSR register to start sending the next frame.
6. If no data is remaining in the transmit FIFO (TDR register), the CSR.TEND flag is set to 1 and the final-bit output state is retained. When the CCR0.TEIE bit is set to 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Note 1. The number of unsent transmit data stored in the TDR register (transmit FIFO) can be monitored by reading the FTSR.T[5:0] bits.

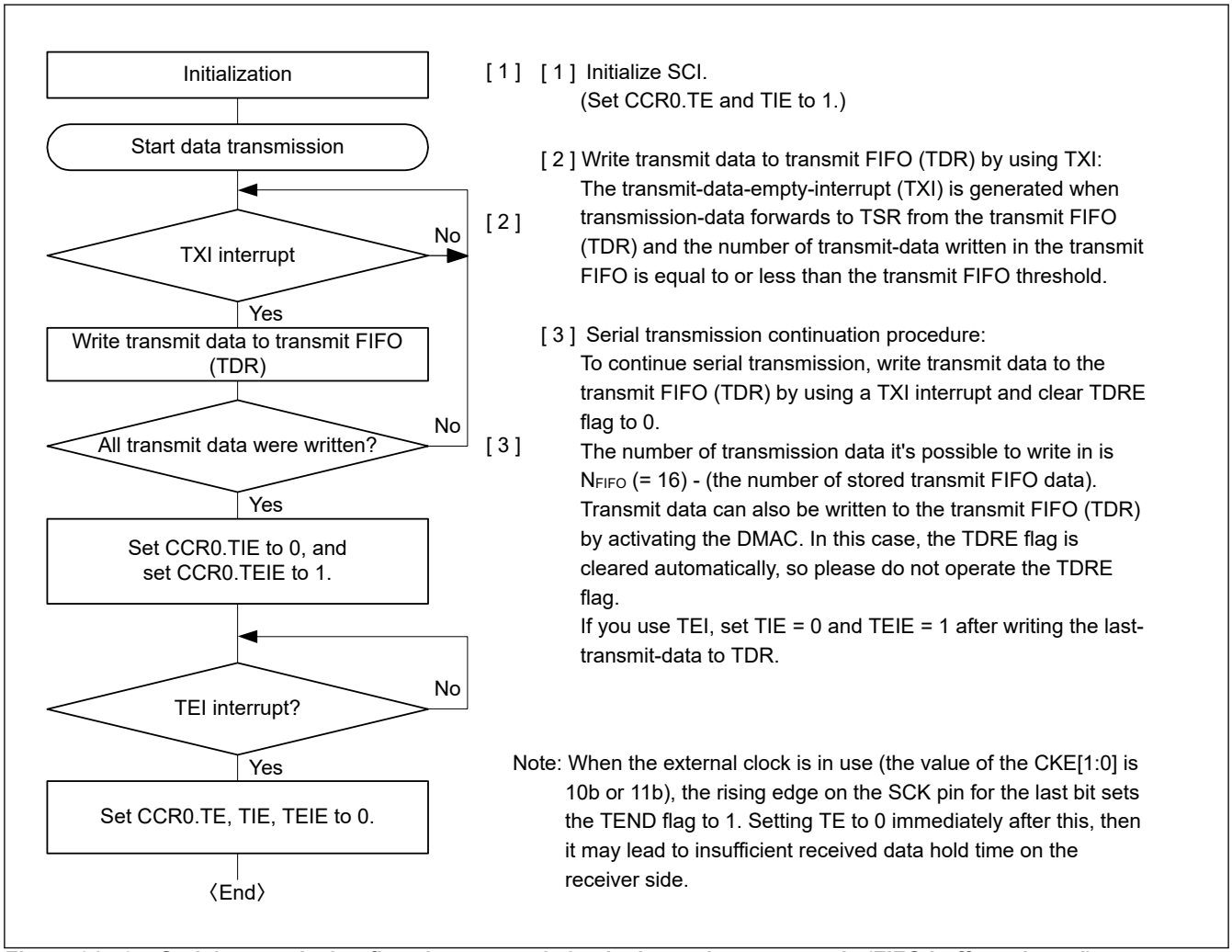


Figure 31.76 Serial transmission flowchart example in clock synchronous mode (FIFO buffer selected)

31.7.5 Serial Data Reception (Clock Synchronous Mode)

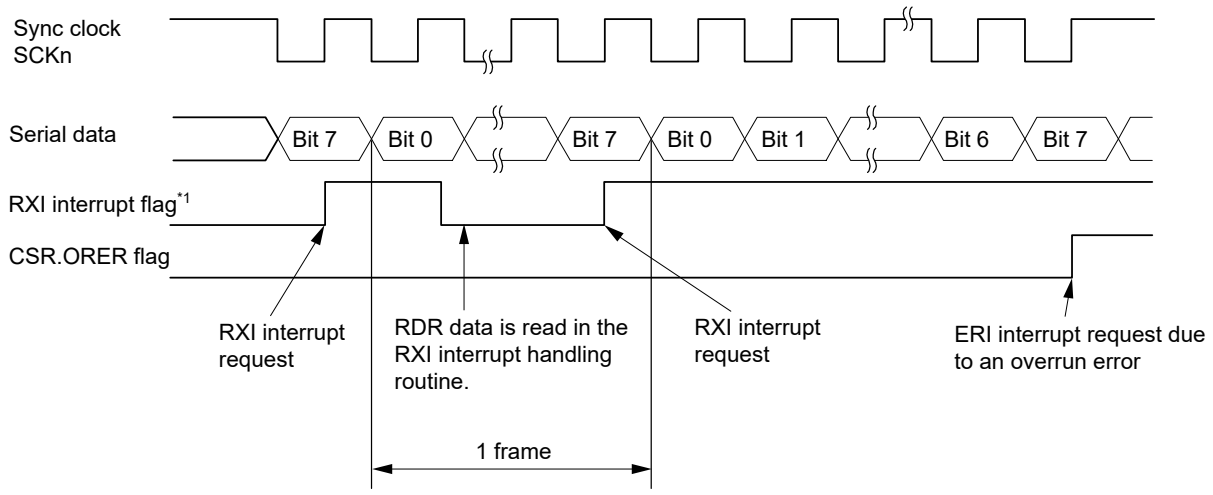
(1) Non-FIFO selected

Figure 31.77 and Figure 31.78 show operation examples of serial data reception in clock synchronous mode.

The SCI operates as follows during serial data reception. Reception-only operation is possible only in slave mode. (In master mode, reception-only operation is prohibited.)

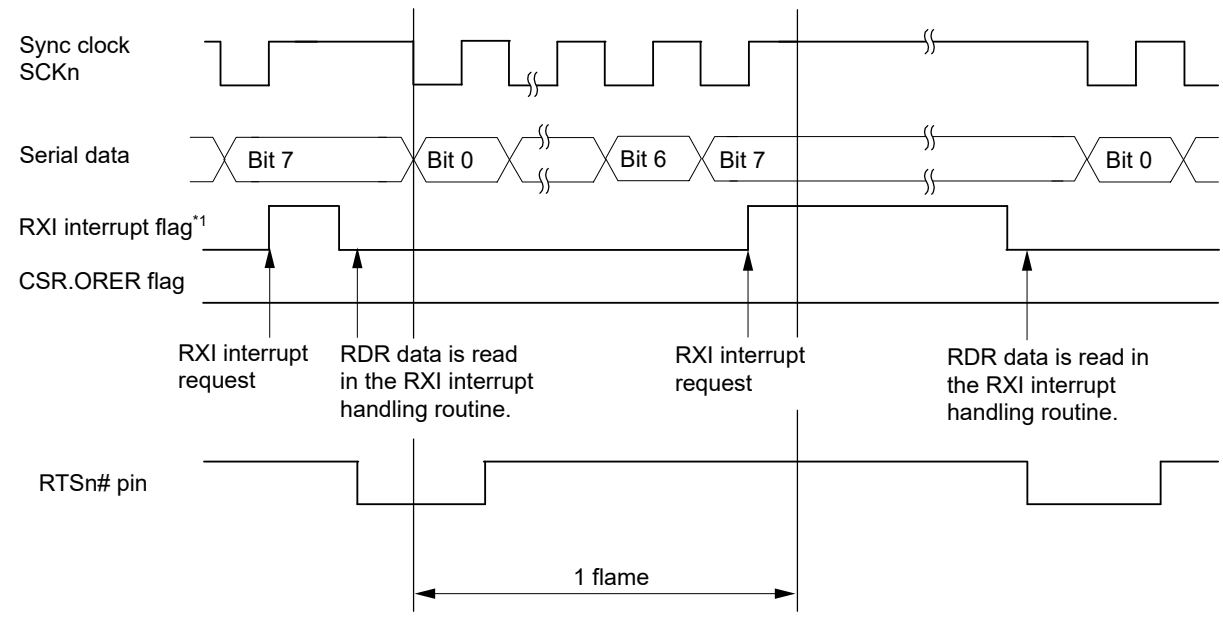
1. When the CCR0.RE bit is set to 1, the RTSn# pin output becomes low level (when the RTS function is used).
2. The SCI starts data reception in synchronization with input or output of the sync clock, and transfers receive data to the RSR register.
3. When an overrun error occurs, the CSR.ORER flag is set to 1. When the CCR0.RIE bit = 1 at this time, an ERI interrupt request is generated and the received data is not transferred to the RDR register.

4. When data is normally received, the received data is transferred to the RDR register. When the RIE bit = 1 at this time, an RXI interrupt request is generated. Reading the received data transferred to the RDR register in the RXI interrupt handling routine before the next data is completely received enables continuous reception. When the received data transferred to the RDR register is read, the RTSn# pin output becomes low level (when the RTS function is used). If you want to prevent the RTSn# pin output from turning low level after the final data is received, clear the CCR0.RE bit to 0 and then read the RDR register.



Note 1. For corresponding interrupt source numbers, see section 11, Interrupt Controller (ICU).

Figure 31.77 Serial data reception example in clock synchronous mode (1) (RTS function not used / CPHA = 1, CPOL = 1)



Note 1. For corresponding interrupt source numbers, see section 11, Interrupt Controller (ICU).

Figure 31.78 Serial data reception example in clock synchronous mode (2) (RTS function used / CPHA = 1, CPOL = 1)

While the reception error flag is set to 1, subsequent reception is disabled. Therefore, before continuing reception, be sure to clear the ORER flag in CSR to 0. Also be sure to read the RDR register in the overrun error processing. If the CCR0.RE bit

is set to 0 during reception to forcibly terminate the reception operation, unread receive data may be remaining in the RDR register. In this case, read the RDR register.

Figure 31.79 shows an example of serial data reception flowchart.

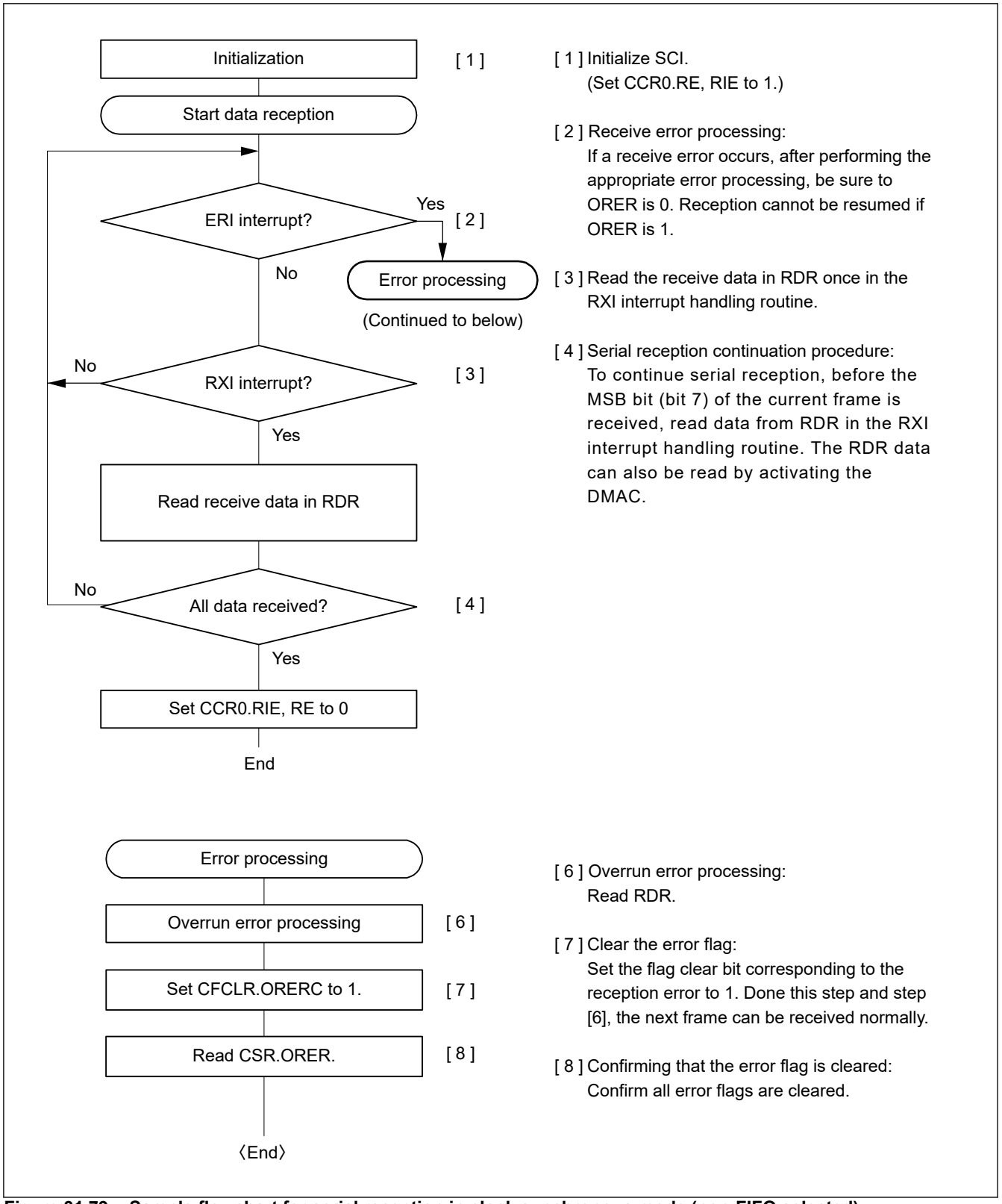


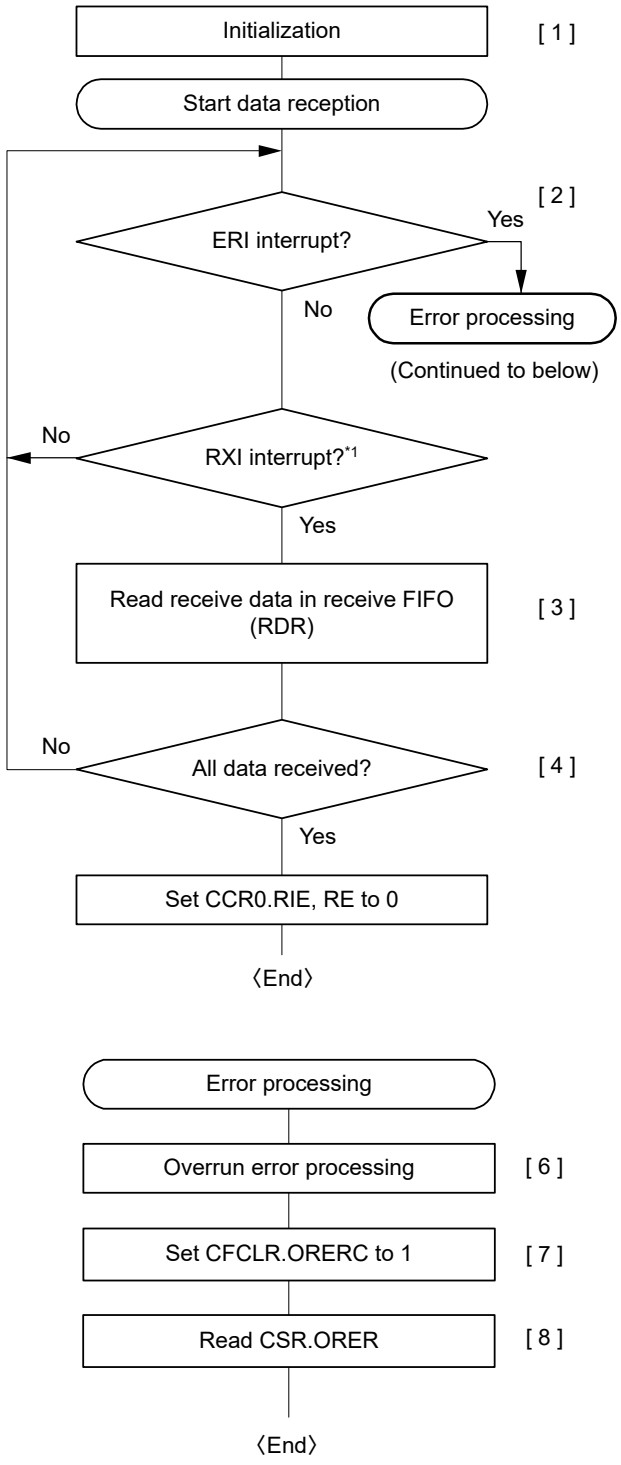
Figure 31.79 Sample flowchart for serial reception in clock synchronous mode (non-FIFO selected)

(2) FIFO selected

Figure 31.80 shows an example of serial data reception flowchart (a FIFO buffer selected) in clock synchronous mode. The SCI operates as follows during serial data reception. Reception-only operation is possible only in slave mode. (In master mode, reception-only operation is prohibited.)

1. When the CCR0.RE bit is set to 1, the RTSn# pin output turns low level (when the RTS function is used).
2. The SCI starts receiving data in synchronization with input or output of the sync clock and transfers the received data to the receive FIFO (RDR register).
3. When an overrun error occurs, the CSR.ORER flag is set to 1. When the CCR0.RIE bit = 1 at this time, an ERI interrupt request is generated and the received data is not transferred to the receive FIFO (RDR register)*1.
4. When data is normally received, the received data is transferred to the receive FIFO (RDR register)*1. When the number of receive data stored in the receive FIFO (RDR register) is equal to or more than the threshold value of the receive FIFO, the CSR.RDRF flag is set to 1. When the RIE bit = 1 at this time, an RXI interrupt request is generated. Reading the received data transferred to the receive FIFO (RDR register) in the RXI interrupt handling routine before an overrun error occurs enables continuous reception. When the received data transferred to the receive FIFO (RDR register) is read and the number of data becomes lower than the RTS# output threshold value, the RTSn# pin output becomes low level (when the RTS function is used).

Note 1. The RDR.RDAT[8] bit is not used.



[1] Initialize SCI.
(Set CCR0.RE, RIE to 1.)

[2] Receive error processing:
If a receive error occurs, after performing the appropriate error processing, be sure to ORER is 0. Reception cannot be resumed if ORER is 1.

[3] Read the receive data in the receive FIFO (RDR) in the RXI interrupt handling routine. The reception data stored in the receive FIFO (RDR) is read until the number of stored data will be below the FCR.RTRG value at least. You can confirm the number of the reception data in the FIFO by FRSR.R[5:0].

[4] Serial reception continuation procedure:
To continue serial reception, before the overrun error occurs, read data from RDR and clear RDRF flag to 0. The RDR data can also be read by activating the DMAC. In this case, the RDRF flag is cleared automatically, so please do not operate the RDRF flag.

Note 1. Set all received data to an integer multiple of the receive FIFO threshold. If it is not an integral multiple, return to step [2] and it will become an infinite loop, so set a timeout time separately using a timer etc.

[6] Overrun error processing:
Read RDR.

[7] Clear the error flag:
Set the flag clear bit corresponding to the reception error to 1. Done this step and step [6], the next frame can be received normally.

[8] Confirming that the error flag is cleared:
Confirm all error flags are cleared.

Figure 31.80 Example of serial reception in clock synchronous mode (FIFO selected)

31.7.6 Serial Data Transmission and Reception (Clock Synchronous Mode)

(1) Non-FIFO selected

Figure 31.81 shows an example of serial data concurrent transmission/reception flowchart in clock synchronous mode. After the SCI is initialized, perform the following procedure for serial data concurrent transmission/reception. When switching mode from transmission to concurrent transmission/reception, check that the CSR.TEND flag is set to 1 to ensure that the

SCI is in the transmission complete state. Then set CCR0.TE = 0 and RE = 0 and then set the TE, RE, TIE, and RIE bits in CCR0 to 1 simultaneously by a single instruction.

When switching mode from reception to concurrent transmission/reception, check that the SCI is in the reception complete state, and then set CCR0.TE = 0 and RE = 0. After that, check that the error flags (ORER, FER, and PER) in CSR are cleared to 0, and then set the TE, RE, TIE, and RIE bits in CCR0 to 1 simultaneously by a single instruction.

When the RTS function is used in the concurrent transmission/reception operation, if you want to prevent the RTSn# pin output from turning to low after the final data is received as in the reception operation, clear the RE and TE bits in CCR0 to 0 simultaneously, and then read the RDR register.

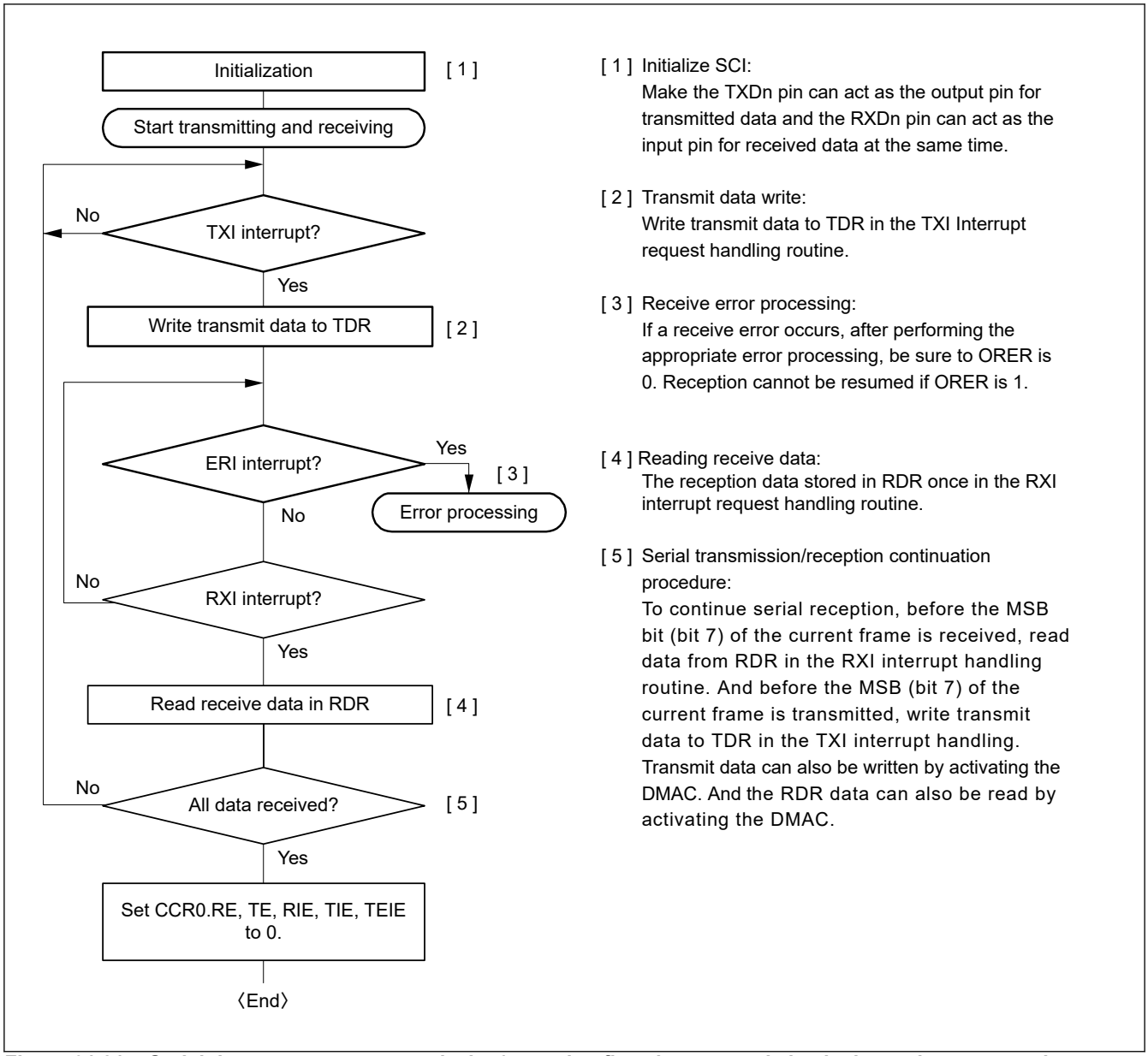


Figure 31.81 Serial data concurrent transmission/reception flowchart example in clock synchronous mode (non- FIFO selected)

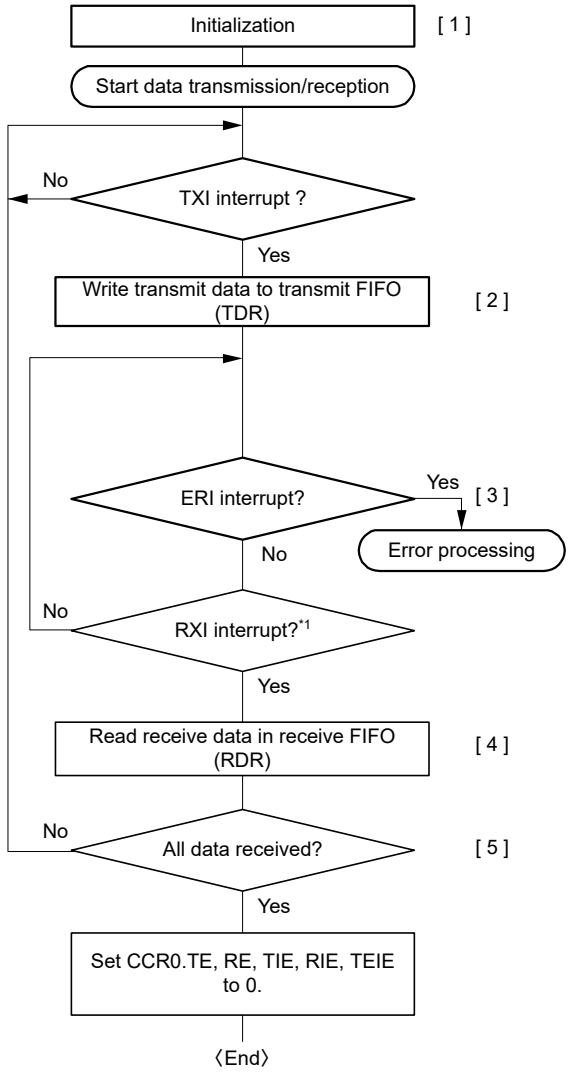
(2) FIFO selected

Figure 31.82 shows an example of serial data concurrent transmission/reception flowchart (a FIFO buffer selected) in clock synchronous mode.

After the SCI is initialized, perform the following procedure for serial data concurrent transmission/reception. When switching mode from transmission to concurrent transmission/reception, check that the CSR.TEND flag is set to 1 to ensure

that the SCI is in the transmission complete state. Then set CCR0.TE = 0 and RE = 0 and then set the TE, RE, TIE, and RIE bits in CCR0 to 1 simultaneously by a single instruction.

When switching mode from reception to concurrent transmission/reception, check that the SCI is in the reception complete state, and then set CCR0.TE = 0 and RE = 0. After that, check that the error flags (ORER, FER, and PER) in CSR are cleared to 0, and then set the TE, RE, TIE, and RIE bits in CCR0 to 1 simultaneously by a single instruction.



[1] Initialize SCI:
Make the TXDn pin can act as the output pin for transmitted data and the RXDn pin can act as the input pin for received data at the same time.

[2] Transmit data write:
Write transmit data to transmit FIFO (TDR) in the TXI Interrupt request handling routine.
The number of transmission data it's possible to write in is N_{FIFO} (= 16) - (the number of stored transmit FIFO data).

[3] Receive error processing:
If a receive error occurs, after performing the appropriate error processing, be sure to ORER is 0. Reception cannot be resumed if ORER is 1.

[4] Read receive data:
Read the receive data in the receive FIFO (RDR) in the RXI interrupt handling routine. The reception data stored in the receive FIFO (RDR) is read until the number of stored data will be below the FCR.RTRG value at least. You can confirm the number of the reception data in the FIFO by FRSR.R[5:0].

[5] Serial transmission/reception continuation procedure:
To continue serial reception, before the overrun error occurs, read data from RDR and clear CSR.RDRF.
To continue serial transmission, write transmit data to the transmit FIFO (TDR) by using a TXI interrupt and clear TDRE flag to 0.
Transmit data can also be written to the transmit FIFO (TDR) by activating the DMAC. And The RDR data can also be read by activating the DMAC. In this case, the TDRE and RDRF are cleared automatically, so please do not operate the TDRE and RDRF.

Note 1. Set all received data to an integer multiple of the receive FIFO threshold. If it is not an integral multiple, return to step [2] and it will become an infinite loop, so set a timeout time separately using a timer etc.

Figure 31.82 Serial data concurrent transmission/reception flowchart example in clock synchronous mode (FIFO selected)

31.7.7 Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used

When the clock synchronous internal clock is used (master mode), MRCLK is used as a reception sampling clock.

This function adjusts the reception sampling timing by delaying MRCLK by 1 to 4 PCLKSCIn and adding a digital delay.

Setting the CCR4.ASEN bit to 1 enables this function. The delay value is set in CCR4.AST[1:0].

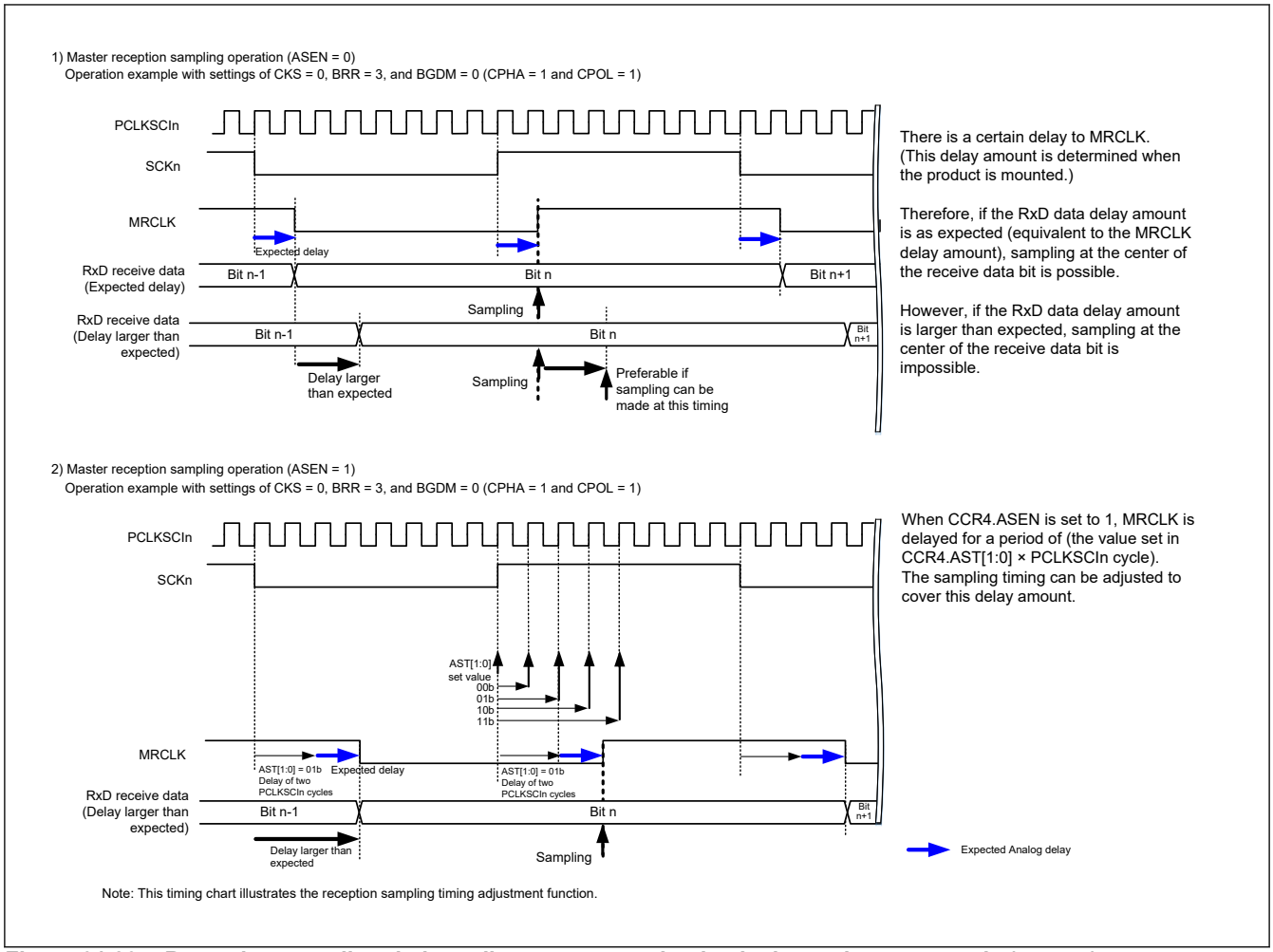


Figure 31.83 Reception sampling timing adjustment operation in clock synchronous mode (master)

31.8 Simple-SPI Mode

As an extended function of the SCI, the SCI supports Simple-SPI (4-wire serial bus) mode that allows communication from one or more master devices to multiple slave devices.

Simple-SPI mode is enabled by the Simple-SPI mode setting (CCR3.MOD[2:0] = 011b) and by setting the CCR0.SSE bit to 1. When using the 4-wire serial bus in master mode with a single master, the SS pin function on the master side is not required. Therefore, the CCR0.SSE bit is set to 0.

Figure 31.84 shows an example of Simple-SPI connection.

In simple-SPI mode, data is transmitted and received in synchronization with clock pulses in the same way as clock synchronous mode. A character of communication data consists of 8-bit data. No parity bit can be attached.

Because the SCI has an internal transmitter and a receiver independently, the transmitter and the receiver can share a clock to enable full-duplex communication. Furthermore, because both the transmitter and the receiver have a double-buffer structure, continuous transmission and reception are possible by writing the next transmit data during transmission and reading the previous receive data during reception.

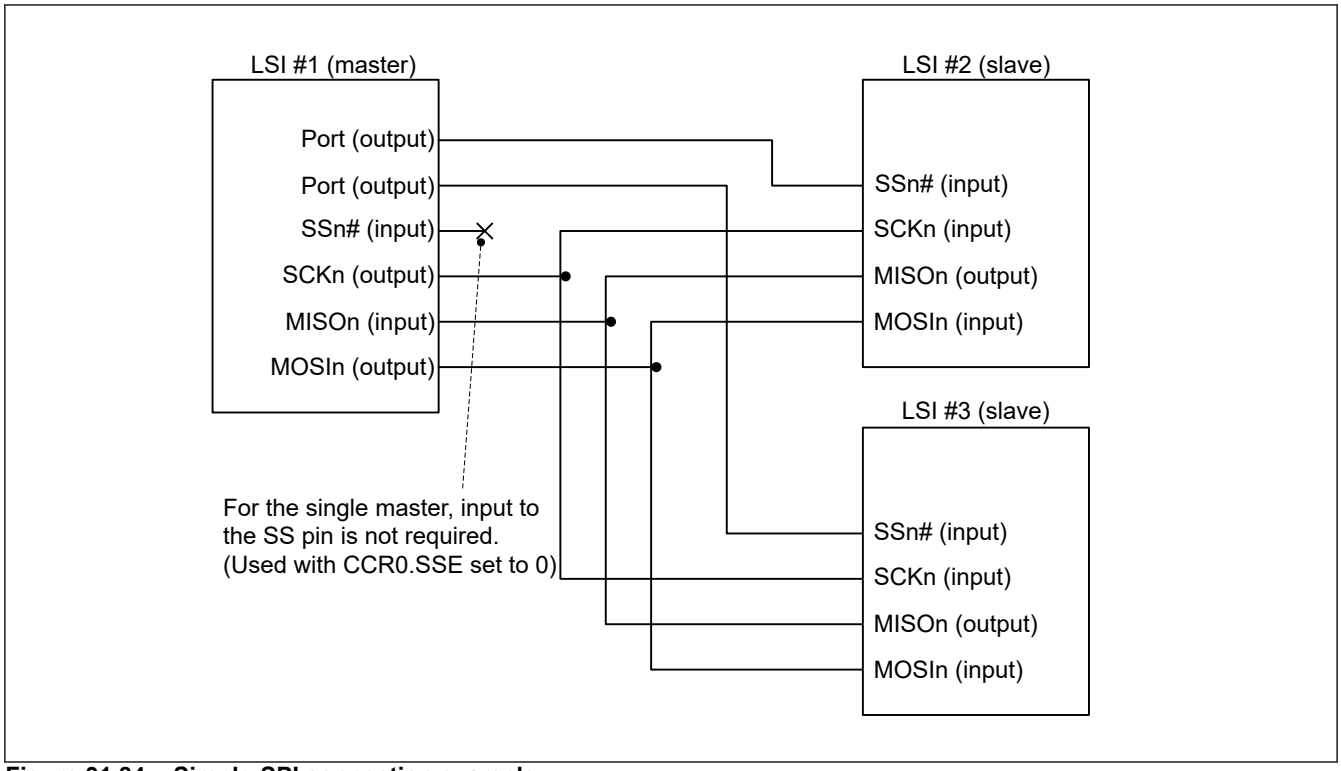


Figure 31.84 Simple-SPI connection example

31.8.1 Pin Status in Master and Slave Mode

In simple-SPI mode, input and output directions of each pin vary depending on master mode (CCR3.CKE[1:0] = 00b or 01b) and slave mode (CCR3.CKE[1:0] = 10b or 11b).

Table 31.27 shows the relationship among mode, SSn# pin input, and each pin state.

Table 31.27 Relationship among mode, SSn# pin input, and each pin state

Mode	SSn# pin input	TXDn (MOSIn) pin state	RXDn (MISOn) pin state	SCKn pin state
Master mode*1	High level (Communication enabled)	Transmit data output*2	Receive data input	Clock output*3
	Low level (Communication disabled)	High impedance	Receive data input (disabled)	High impedance
Slave mode	High level (Communication disabled)	Receive data input (disabled)	High impedance	Clock input (disabled)
	Low level (Communication enabled)	Receive data input	Transmit data output*2	Clock input

Note 1. When single master is selected (SSE bit = 0), communication is enabled (equivalent to the state when the SSn# pin input is high level) regardless of the SSn# pin input level. The SSn# pin is not used and is available for other purposes.
Note 2. High impedance when transmission is disabled (CCR0.TE bit = 0)
Note 3. High impedance when multi-master (SSE bit = 1) is selected and transmission/reception is disabled (CCR0.TE, RE bits = 00b)

31.8.2 SS Function in Master Mode

Setting the CCR3.CKE[1:0] bits to 00b or 01b enables master mode.

In single master mode (SSE bit = 0), the SSn# pin is not used and data transmission and reception are enabled regardless of the SSn# pin input level. The SSn# pin is available for other purposes.

When in multi-master mode (SSE bit = 1) and the SSn# pin input is high, the master outputs a clock from the SCKn pin and performs transmission and reception operations. And outputting clock indicates "There are no other masters" or "Another master is not performing reception or transmission". When the SSn# pin input level is low in multi-master mode (SSE bit = 1), this means that another master exists and it is performing data transmission/reception. At this time, the SCI makes the TXDn pin output and the SCKn pin output high impedance and does not start data transmission/reception. In addition, the

CSR.MFF bit is set to 1 as a mode fault error. In multi-master mode, read this flag bit to perform the error processing. If a mode fault error occurs during the transmission/reception operation, the SCKn pin and the TXDn pin output are made high impedance while the SSn# pin input level is low. When the SSn# pin input becomes high level, the SCKn pin outputs a clock signal and the TXDn pin outputs data. Even if the SCKn pin and the TXDn pin are in the high impedance state, internal transmission/reception operation continues, but it stops after transmission/reception of a single character is complete. In this case, any of TXI, RXI, and TEI interrupts occurs.

Control the SSn# pin output in master mode with a general-purpose port.

31.8.3 SS Function in Slave Mode

Setting the CCR3.CKE[1:0] bits to 10 or 11 enables slave mode.

When the SSn# pin input level is high, the RXDn pin output becomes high impedance and the clock input from the SCKn pin is ignored. When the SSn# pin input level is low, the clock input from the SCKn pin becomes effective, enabling data transmission and reception.

When the SSn# pin input changes from low to high level during the transmission/reception operation, the RXDn pin output is made high impedance and the transmission/reception operation is immediately suspended. If the transmission is in progress, the CSR.TEND flag will not be set, a transmit end interrupt will not be output, and an abnormal stop status will occur. So, do not negate the SSn# pin during slave transmission/reception. If an abnormal stop occurs, set CCR0.RE and CCR0.TE to 0 to stop transmission / reception. To resume transmission/reception, set CCR0.RE and CCR0.TE to 1 after at least PCLKSCIn × 3 cycles + PCLKM × 3 cycles.

31.8.4 Relationship between Clock and Transmit/Receive Data

The clock to be used for data transmission/reception is selectable from four types using the CCR3.CPOL and CPHA bits. Figure 31.85 shows the relationship between clock and transmit data/receive data. The same relationship applies to master mode and slave mode.

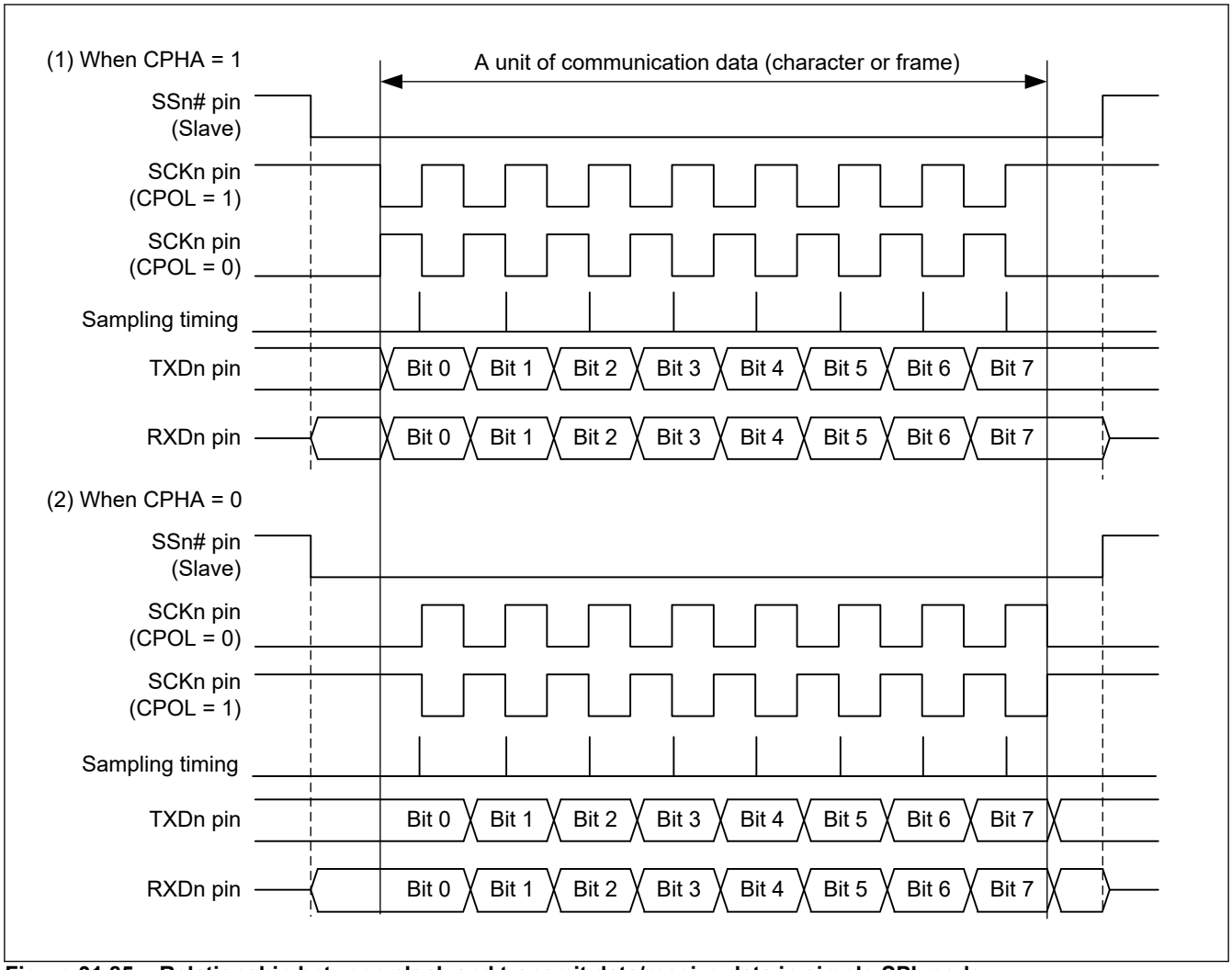


Figure 31.85 Relationship between clock and transmit data/receive data in simple-SPI mode

31.8.5 SCI Initialization in Simple-SPI Mode

The SCI can be initialized using the same initialization procedure as for clock synchronous mode (Figure 31.71). The master devices and slave devices use the same clock type selected by the CCR3.CPOL and CPHA bits.

Before performing initialization or changing operating mode or communication format, be sure to stop communication (CCR0.RE = 0 and CCR0.TE = 0).

Note that setting the RE bit to 0 does not initialize the ORER, FER, and PER flags in CSR and the RDR register.

Note that, when CCR0.TIE = 1, setting the TE bit to 1 from 0 generates a TXI interrupt.

31.8.6 Serial Data Transmission and Reception in Simple-SPI Mode

In master mode, set the SSn# pin of the destination slave device to low level before starting data transmission/reception and set to high level after the end of data transmission/reception. In multiple master operation with CCR0.SSE = 1 even in master mode, a mode fault error will occur if the SSn# pin goes low. Therefore, make sure that no mode fault error has occurred before starting communication, and start communication, and make sure that no mode fault error has occurred even after communication ends. If a mode fault error has occurred, communication may be incomplete, so measures such as retransmission are required. The other procedures are the same as in clock synchronous mode.

In slave mode, it operates according to the SSn# pin input level. Other steps are the same as those of clock synchronous mode.

31.8.7 Reception Sampling Timing Adjustment Function in Simple-SPI Mode with Internal Clock Used

The reception sampling timing adjustment function in simple SPI mode is the same as the reception sampling timing adjustment function in clock synchronous mode. For the description of operation, see section 31.7.7. Reception Sampling Timing Adjustment Function in Clock Synchronous Mode with Internal Clock Used.

31.9 Bit Rate Modulation Function

Using the bit rate modulation function, the bit rate can be corrected by evenly enabling the clocks of the number specified in the CCR2.MDDR[7:0] bits among 256 clock cycles of internal clocks which is selected by the CKS[1:0] bits in CCR2.

Figure 31.86 shows an example where the PCLKSCIn is selected by the CKS[1:0] bits in CCR2 and the BRR and MDDR are set to 0 and 160 respectively in asynchronous mode. In this example, the cycle of the base clock is evenly corrected (256/160) and the bit rate is also corrected (160/256). The enable of internal clock has non-balanced. So, notice the pulse width of the internal base clock is caused bias.

Do not use this function in Clock Synchronous mode, Simple-SPI mode, Smart Card Interface mode.

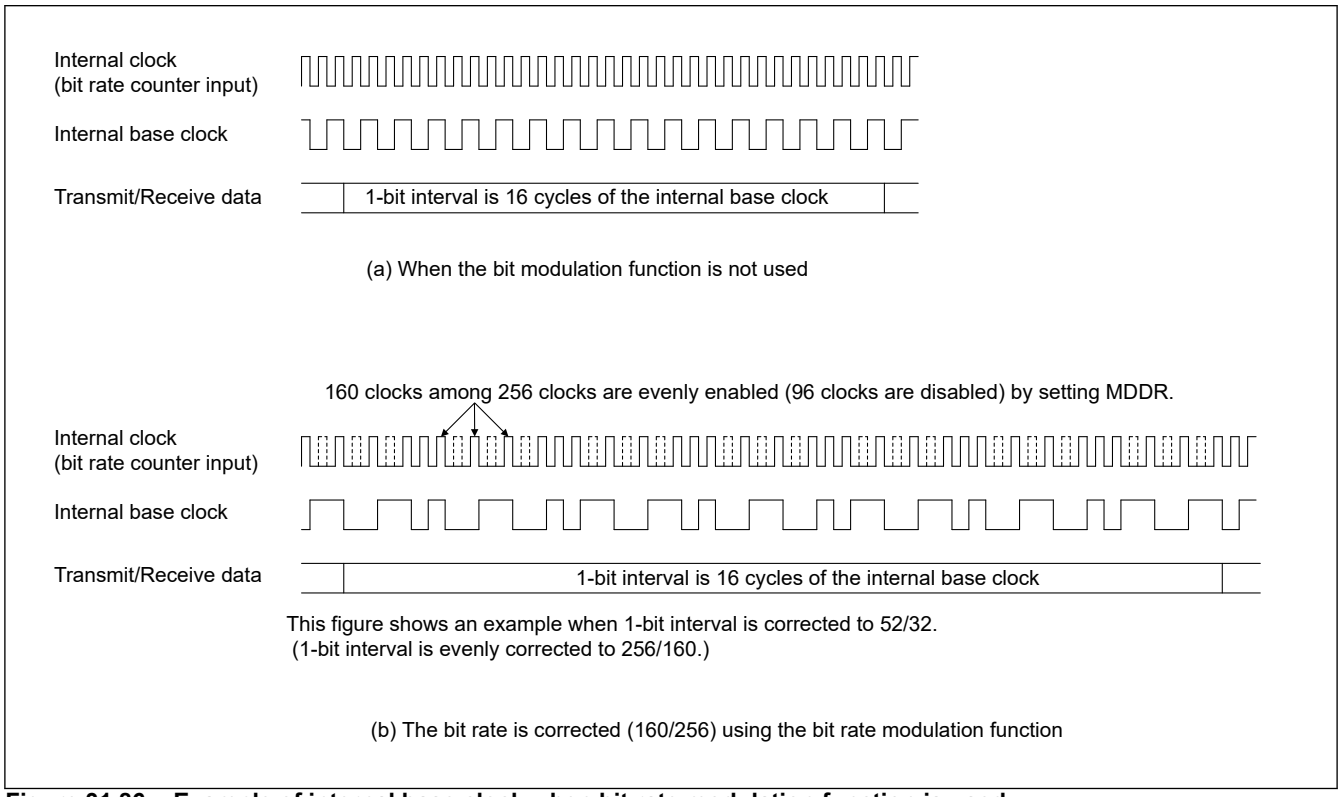


Figure 31.86 Example of internal base clock when bit rate modulation function is used

31.10 Noise Cancellation Function

Figure 31.87 shows the configuration of the noise filter used for noise cancellation. The noise filter consists of a two-stage flip-flop circuits and a match detection circuit. When the input signals of the noise filter and the output signals of the two-stage flip-flop circuits completely match, the matched level is conveyed as an internal signal. Unless otherwise matched, the previous value is retained. (When the same level is retained for three cycles or longer on the sampling clock of the noise filter, it is considered as a valid receive signal. A change in pulse for three cycles or shorter is considered as a noise, not as a receive signal.)

In asynchronous mode, the noise cancellation function can be applied to the receive signal input to the RXDn pin. The sampling period of the noise filter can be selected from the base clock period and the divided clock of the baud rate generator clock source by CCR1.NFCS[2:0].

(When CCR1.NFCS[2:0] = 000b, CCR2.ABCS = 0 and CCR2.ABCSE = 0, the cycle is 1/16 of a period 1 transfer bit.

When CCR1.NFCS[2:0] = 000b, CCR2.ABCS = 1 and CCR2.ABCSE = 0, the cycle is 1/8 of a period 1 transfer bit.

When CCR1.NFCS[2:0] = 000b, CCR2.ABCSE = 1, the cycle is 1/6 of a period 1 transfer bit.)

In simple I2C mode, the noise elimination function can be used for the input pins of SDAn and SCLn. The sampling period of the noise filter can be selected from the divided clock of the baud rate generator clock source by CCR1.NFCS[2:0].

If the base clock is stopped once with the noise filter enabled and then the base clock input is restarted again, the noise filter operation resumes from the state where the clock was stopped. When CCR0.TE and CCR0.RE are set to 0 during base clock input, all of the noise filter flip-flop values are initialized to 1. Accordingly, if the input data is 1 when reception operation resumes, it is determined that a level match is detected and is conveyed as an internal signal. When the level being input corresponds to 0, the initial output of the noise filter is retained until the level matches in three consecutive sampling cycles.

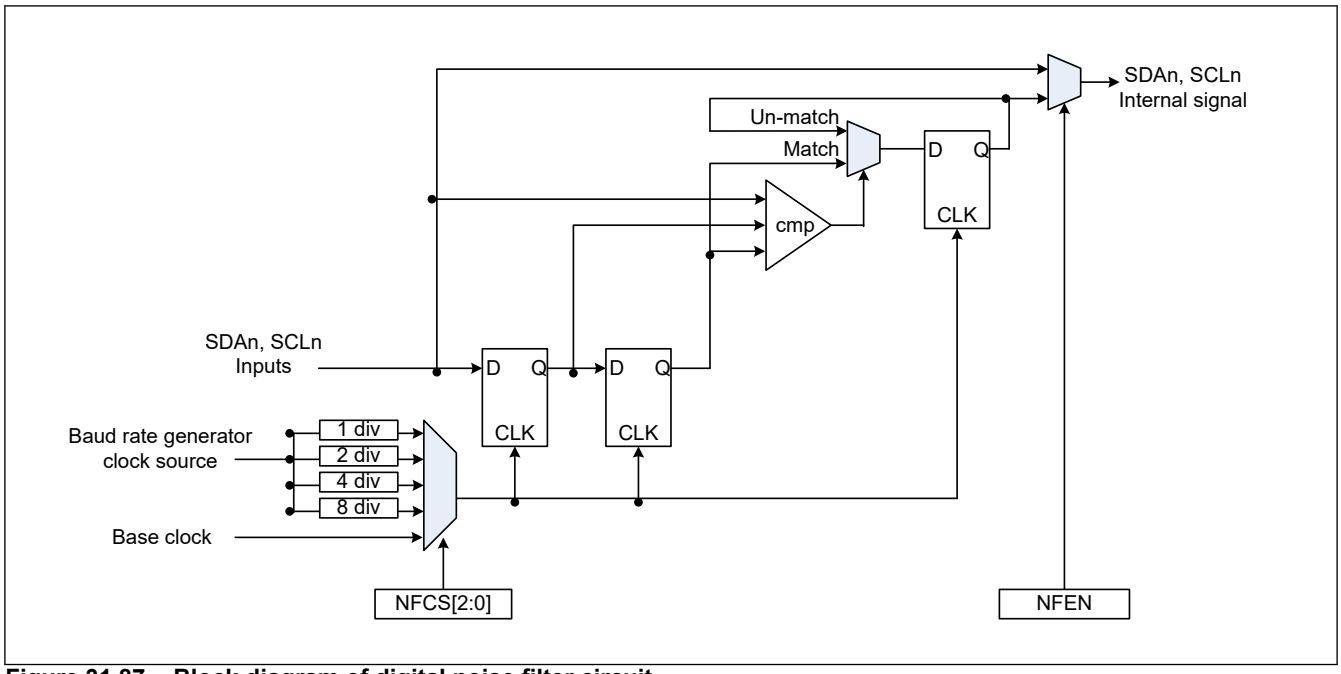


Figure 31.87 Block diagram of digital noise filter circuit

31.11 RS-485 Driver Control Function

Setting the DEN bit in the SCI Common Control Register 3 (CCR3) to 1 enables the RS-485 driver control function and generates a DE (Driver Enable) signal that enables the external transceiver transmission mode. The DE signal outputs a valid level for the period with assertion time and negation time added before and after data transmission. The DE signal valid level is set by the DEPOL bit in the driver control register (DCR).

The Assert time is the time from when the DE signal is valid until the start bit starts. Set by DEAST[4:0] of driver control register (DCR).

The Negate time is the time from the end of the last stop bit of the transmitted message to the invalidation of the DE signal. Set with DENGT[4:0] of the driver control register (DCR).

DEAST and DENGT are expressed in base clock units. For details, see section 31.2.12. DCR : Driver Control Register.

When this function is used (DEN = 1), the TEND set timing and TEI interrupt output timing are at the end of the driver negation time.

When transmission is completed and the next transmission data is not written before the DE signal is negated, the DE signal is negated once. If the timing for writing the next transmit data is not in time, assert the DE signal after negating it again, insert the set assert time, and transmit the next data. If you want to perform the next transmission with the DE signal asserted, write the next transmission data to the TDR quickly enough in consideration of the synchronization delay time of the register.

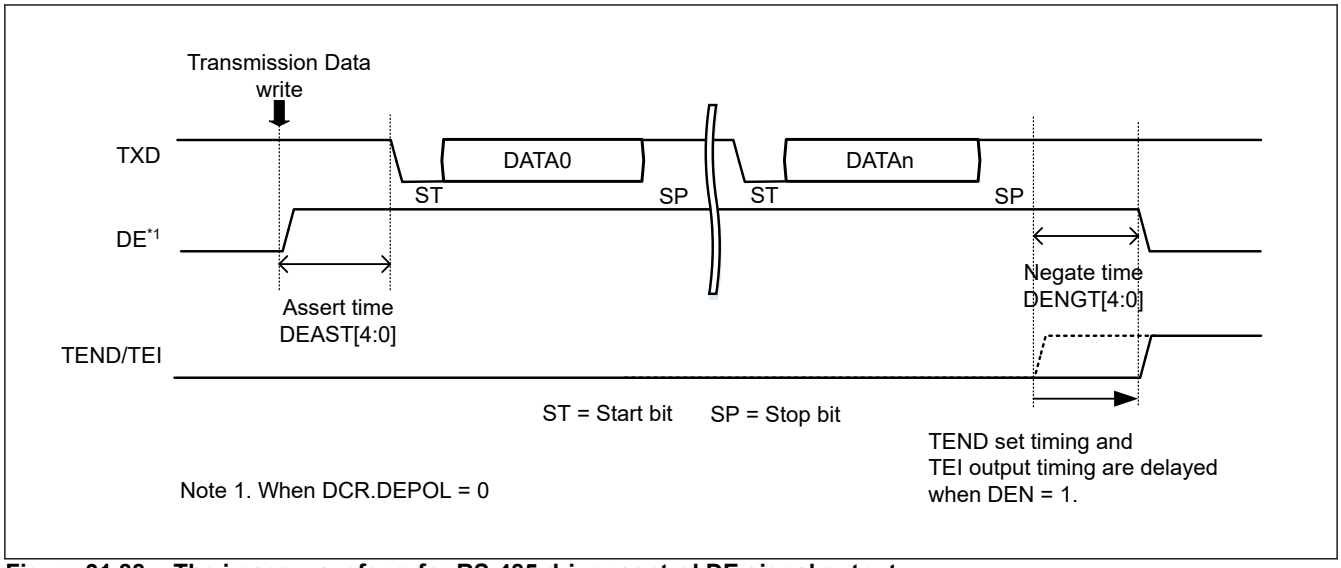


Figure 31.88 The image waveform for RS-485 driver control DE signal output

31.12 Loopback Function

The loopback function can be used in asynchronous mode with the internal clock, and clock synchronous mode with the internal clock.

When 1 is written to the SPLP bit in the CCR1 register, SCI blocks the external input (RXD) path and connects the output path of the transmit data register and the input path of the receive data register.

When this function is used with TINV = 1, inversion of transmission data becomes reception data. However, this function can be used with TINV = 1 only when operating in clock synchronous mode internal clock.

Table 31.28 shows the relationship between the TINV and SPLP bit settings and the received data.

Table 31.28 TINV and SPLP bit settings and received data

TINV	SPLP	Receive data	Communication mode	
			Async internal clock	Clock sync internal clock
x	0	Receive Data from RXD pin	Possible	Possible
0	1	Transmit Data	Possible	Possible
1	1	Inverted transmit data	Impossible	Possible

Figure 31.89 shows the configuration of the shift register input/output path in loopback mode.

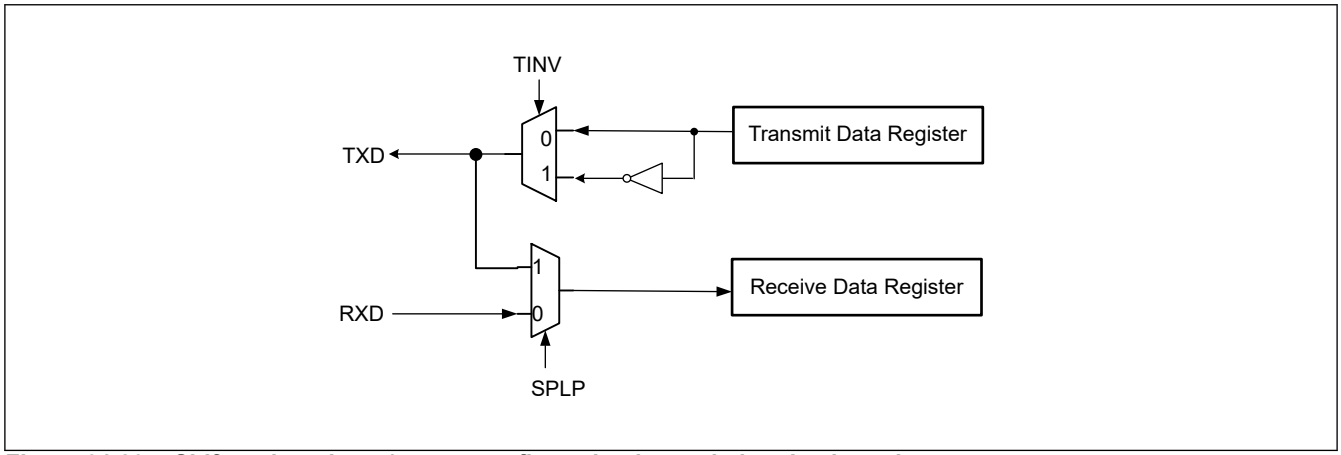


Figure 31.89 Shift register input/output configuration image in loopback mode

31.13 Half-Duplex communication Function

Do not use the half-duplex communication function in Simple-I2C mode, Simple-SPI mode, and Smart Card Interface mode.

In other communication modes, if the CCR1.SHARPS bit is set to 1, half-duplex communication using the TXD pin is possible. When half-duplex communication is used, transmission and reception must be performed exclusively. Transmission and reception settings (CCR0.TE = 1 and CCR0.RE = 1) is prohibited.

However, if half-duplex communication is performed as the master in clock synchronous mode, perform transmission/reception settings (CCR0.TE = 1 and CCR0.RE = 1) and perform dummy transmission. By dummy transmission (arbitrary transmission data is written to TDR), SCKn is output and reception is enabled. The dummy transmission data is discarded inside the module and is not actually transmitted.

During half-duplex communication, the only communication port terminal used is the TXD pin. Output when CCR0.TE = 1, input when CCR0.TE = 0.

31.14 Synchronizer Bypass Function

This module has a bus clock and the operation clock (PCLKSCIn). And these have each operating circuit. Therefore, there is a synchronization circuit for signal transfer between different clocks, and synchronization delay time is required for signal propagation between different clocks.

However, the synchronization circuit can be bypassed by the CCR3.BPEN bit only when the same clock is input to the bus clock and the operation clock. In this case, eliminates synchronization delay time and improves responsiveness. Figure 31.90 shows the image waveform of the bypass function.

This module also has a synchronization circuit between the communication clock (SCK) and the operation clock (PCLKSCIn), but this synchronization circuit can not be bypassed.

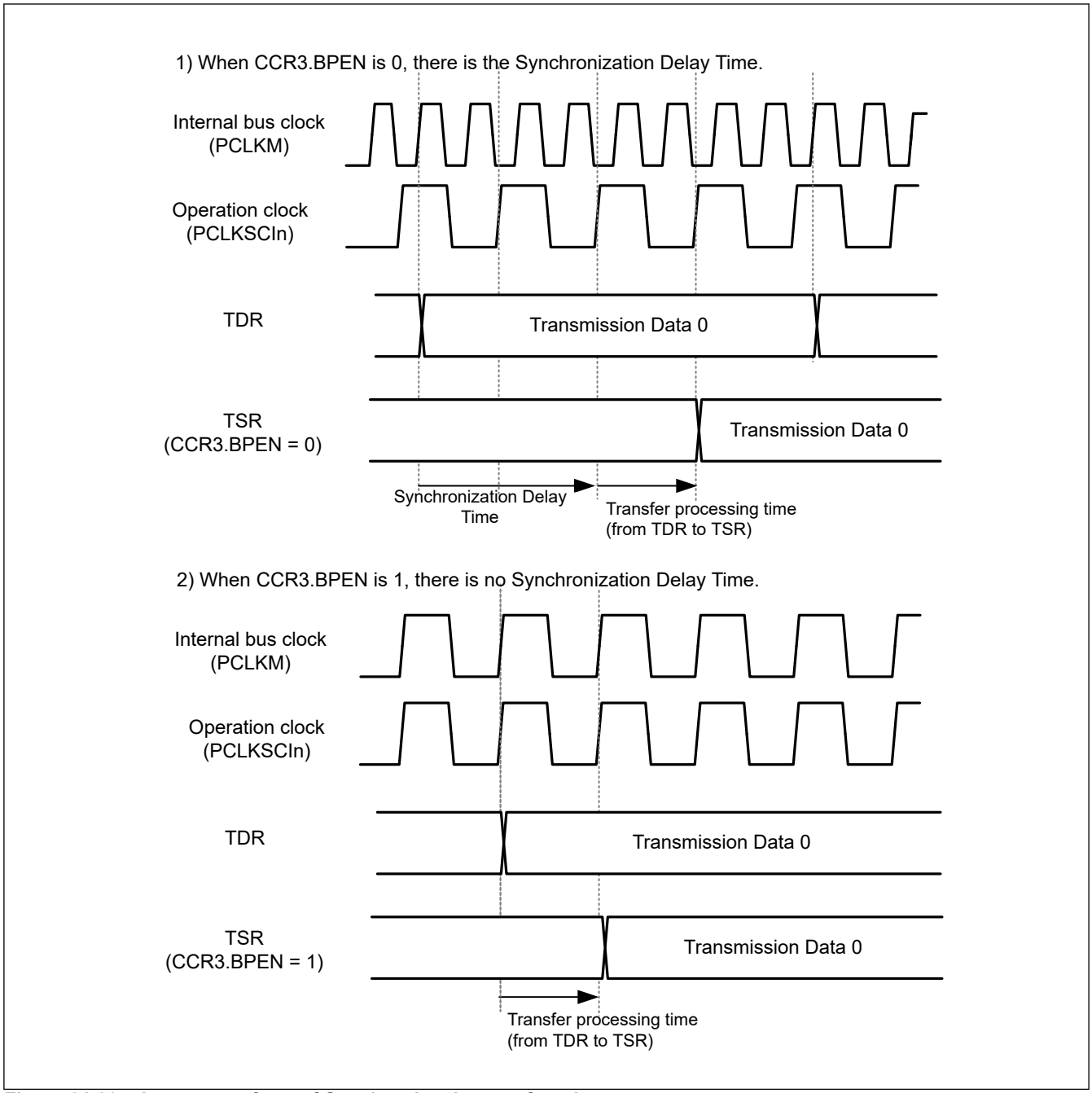


Figure 31.90 Image waveform of Synchronizer bypass function

31.15 Interrupt signal

Table 31.29 lists SCI interrupt signals.

When performing transmission and reception using DMAC, be sure to set DMAC first, and then set SCI. See section 14, DMA Controller (DMAC) for how to set DMAC.

Table 31.29 SCI interrupt sources (1 of 2)

Name	Interrupt sources	CPU0 GIC request	DMAC activation
SCIn_ERI	Error interrupt	Possible	Not possible
SCIn_RXI	Simple-I2C: Reception end interrupt Other mode: Receive data full interrupt	Possible	Possible*1

Table 31.29 SCI interrupt sources (2 of 2)

Name	Interrupt sources	CPU0 GIC request	DMAC activation
SCIn_TXI	Simple-I2C and Smart Card Interface: Transmit end interrupt Other mode: Transmit data empty interrupt	Possible	Possible*1
SCIn_TEI	Simple-I2C: Completion of generation of a start, restart, or stop condition (STI) Other mode: Transmit end interrupt	Possible	Not possible

Note: n = 0 to 5
Available interrupt sources change dependent on operation mode.
Note 1. DMAC activation is not available for n = 5.

31.15.1 Buffer Operations for TXI and RXI interrupts

The TXI and RXI interrupts have an interrupt buffer function. When the first interrupt request is generated during interrupt handling and the next interrupt request is generated (when the status flag of the Interrupt Controller Unit (ICU) is 1), the module does not output the interrupt request, and the SCI holds it internally. The interrupt that can be held is up to one.

31.15.2 Interrupt in Asynchronous Mode, Clock Synchronous Mode, and Simple-SPI Mode

A different interrupt vector is assigned to each interrupt source, and individual interrupt sources can be enabled or disabled with the enable bits in CCR0 register.

(1) Non-FIFO selected

Table 31.30 lists interrupt sources in Asynchronous Mode, Clock Synchronous Mode, and Simple-SPI Mode with Non-FIFO selected.

If the CCR0.TIE bit is 1, a TXI interrupt request is generated when transmit data is transferred from the TDR to the TSR. At the start of transmission, a TXI interrupt request can also be generated by using a single instruction to set the CCR0.TE and CCR0.TIE bit to 1 at the same time. A TXI interrupt request can activate the DMAC to handle data transfer.

A TXI interrupt request is not generated by setting the CCR0.TIE bit to 1 while the setting of the CCR0.TE bit is 1.*1

When the CCR0.TEIE bit is 1, the CSR.TEND flag becomes 1 and the TEI interrupt request is generated if the next data is not written to the TDR register by the timing to transmit the last bit of transmission data. In addition, the TEND flag holds 1 during the period from setting the CCR0.TE bit to 1 until writing transmit data to the TDR register, and if the TEIE bit is set to 1, a TEI interrupt request is generated.

Writing data to the TDR register clears the TEND flag and cancels the TEI interrupt request, but it takes time to cancel it.

If the CCR0.RIE bit is 1, an RXI interrupt request is generated when received data is stored in the RDR. An RXI interrupt request can activate the DMAC to handle data transfer.

Setting of any from among the ORER, FER, PER flags in the CSR to 1 while the CCR0.RIE bit is 1 leads to the generation of an ERI interrupt request.

An RXI interrupt request is not generated at this time. Clearing all three flags (ORER, FER, PER) leads to discarding of the ERI interrupt request.

Note 1. To temporarily prohibit TXI interrupts at the time of transmission of the last of the data and so on when you wish a new round of transmission to start after handling of the transmit end interrupt, control prohibiting and permitting of the interrupt by using the interrupt request enable bit in the interrupt controller rather than using the TIE bit. This can prevent the suppression of TXI interrupt requests in the transfer of new data.

Table 31.30 SCI interrupt sources with Non-FIFO selected

Name	Interrupt source	Interrupt flag	Level/pulse output	Interrupt enable	DMAC activation	Priority
ERI	Receive error	ORER, FER, PER, DFER, DPER	Level	RIE	Not possible	High
RXI	Receive data full	RDRF	Pulse	RIE	Possible	↑
	Address match	DCMF				
TXI	Transmit data empty	TDRE	Pulse	TIE	Possible	↑
	TE = 0 -> 1 detection					
TEI	Transmit end	TEND	Level	TEIE	Not possible	Low

(2) FIFO selected

Table 31.31 lists interrupt sources in Asynchronous Mode, Clock Synchronous Mode, and Simple-SPI Mode with FIFO selected.

If the CCR0.TIE bit is 1, a TXI interrupt request is generated when the stored number of data in transmit FIFO becomes the threshold value indicated in FCR.TTRG or below. A TXI interrupt request can also be generated by using a single instruction to set the CCR0.TE and CCR0.TIE bit to 1 at the same time. A TXI interrupt request is not generated by setting the CCR0.TIE bit to 1 while the setting of the CCR0.TE bit is 1. If CCR0.TEIE bit is 1, when the next data is not being written in transmit FIFO by the timing to which the last bit of the transmission data is sent, CSR.TEND flag will be 1 and TEI interrupt request is generated.

If the CCR0.RIE bit is 1, RXI interrupt request is generated when the stored number of data in receive FIFO becomes the threshold value indicated in FCR.RTRG or above. When FCR.RTRG is set to 0, RXI interrupt request is occurred if the quantity of data in receive FIFO is greater than or equal to 1.

If the CCR0.RIE bit is 1, when CSR.ORER flag is set to 1 or the data a framing error or a parity error generated is stored in receive FIFO, ERI interrupt request is generated.

When the number of data stored in a receive FIFO at this time is a threshold value or above, RXI interrupt request is also generated. The ERI interrupt request can be canceled by clearing all flags (CSR.ORER, FER, and PER).

Table 31.31 SCI interrupt sources with FIFO selected

Name	Interrupt source	Interrupt flag	Level/pulse output	Interrupt enable	DMAC activation	Priority
ERI	Receive error	ORER, FER, PER, DFER, DPER	Level	RIE	Not possible	High
		DR (FCR.DRES = 1)				
RXI	Receive FIFO data full	RDRF	Pulse	RIE	Possible	↑
	Receive data ready	DR (FCR.DRES = 0)				
	Address match	DCMF				
TXI	Transmit FIFO data empty	TDRE	Pulse	TIE	Possible	↑
	TE = 0 -> 1 detection					
TEI	Transmit end	TEND	Level	TEIE	Not possible	Low

31.15.3 Interrupt in Smart Card Interface Mode

Table 31.32 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 31.32 SCI interrupt sources in smart card interface mode

Name	Interrupt source	Interrupt flag	Level/pulse output	Interrupt enable	DMAC activation	Priority
ERI	Receive error or error signal detection	ORER, PER, ERS	Level	RIE	Not possible	High
RXI	Receive data full	RDRF	Pulse	RIE	Possible	↑
TXI	Transmit end	TEND	Pulse	TIE	Possible	Low
	When set TE = 0 -> 1					

Data transmission/reception using the DMAC is also possible in smart card interface mode. In transmission operation, when the TEND flag in CSR is set to 1, a TXI interrupt request is generated. This TXI interrupt request activates the DMAC allowing transfer of transmit data if the TXI request is specified beforehand as a source of DMAC activation. The TEND flag is automatically set to 0 when the DMAC transfers the data.

If an error occurs, the SCI automatically retransmits the same data. During the retransmission, the TEND flag is kept to 0 and the DMAC is not activated. Therefore, the SCI and DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, the ERS flag in CSR is not automatically cleared to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the RIE bit in CCR0 to 1 to enable an ERI interrupt request to be generated at error occurrence.

In reception operation, an RXI interrupt request is generated when receive data is set to RDR. This RXI interrupt request activates the DMAC allowing transfer of receive data if the RXI request is specified beforehand as a source of DMAC activation. If an error occurs, the error flag is set. Therefore, the DMAC is not activated and an ERI interrupt request is issued to the CPU instead; the error flag must be cleared.

When transmitting/receiving data using the DMAC, be sure to make settings to enable the DMAC before making SCI settings. For DMAC settings, see section 14, DMA Controller (DMAC).

31.15.4 Interrupts in Simple-I2C Mode

Table 31.33 lists SCI interrupts in simple-I2C mode.

The STI interrupt is allocated to the transmit end interrupt (TEI) request. The receive error interrupt (ERI) request cannot be used.

The DMAC can also be used to handle transfer in simple I2C mode.

When the value of the IICINTM bit in ICR is 1, a RXI request will be generated on the falling edge of the SCLn signal for the eighth bit. If the RXI has been set up as an activating request for the DMAC beforehand, the RXI request will activate the DMAC to handle transfer of the received data. Furthermore, a TXI request is generated on the falling edge of the SCLn signal for the ninth bit (acknowledge bit). If the TXI has been set up as an activating request for the DMAC beforehand, the TXI request will activate the DMAC to handle transfer of the transmit data. (In this case, ACK/NACK judging are impossible.)

When the value of the IICINTM bit in ICR is 0, SCI operates as follows.

RXI request (ACK detection) is generated if the input on the SDAn pin is at the low level on the rising edge of the SCLn signal for the ninth bit (acknowledge bit).

TXI request (NACK detection) is generated if the input on the SDAn pin is at the high level on the rising edge of the SCLn signal for the ninth bit (acknowledge bit).

If the RXI has been set up as an activating request for the DMAC beforehand, the RXI request will activate the DMAC to handle transfer of the received data. Also, if the DMAC is used for data transfer in reception or transmission, be sure to set up and enable the DMAC before setting up the SCI.

When the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits in ICR are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

Table 31.33 SCI interrupt sources in Simple-I2C mode

Name	Interrupt source		Interrupt flag	Level/ pulse output	Interrupt enable	DMAC activation	Priority
	IICINTM = 1	IICINTM = 0					
RXI	Reception end	—	—	Pulse	RIE	Possible*1	High
	—	ACK detection	—			Possible	↑
TXI	Transmit end	—	—	Pulse	TIE	Possible*1	↑
	—	NACK detection	—			Possible	↑
STI	Completion of generation of a start, restart, or stop condition		IICSTIF	Level	TEIE	Not Possible	Low

Note 1. If the DMAC is being used, you can not confirm whether ACK or NACK.

31.16 Usage Notes

31.16.1 SCI Operations during Low Power Consumption State

(1) Transmission

Before using the power consumption reduction function to reduce SCI's power consumption, please do the following to confirming transmission end (CSR.TEND = 1):

- Set the output pin state after transmission operation is stopped by CCR1.SPB2DT, SPB2IO.
- Stop the transmission (CCR0.TIE = 0, TE = 0, TEIE = 0).

When transitions to these states are made during transmission, the data being transmitted become indeterminate.

To transmit data in the same transmission mode after cancellation of the low power consumption state, set the TE bit to 1, read CSR, and write data to TDR sequentially to start data transmission. To transmit data with a different transmission mode, initialize the SCI first.

To start transmission using the DMAC after cancellation of the low power consumption state, set the TE and TIE bit to 1. The TXI interrupt flag is set to 1 and transmission starts using the DMAC.

Figure 31.91 shows a sample flowchart for transition to the low power consumption state during transmission. Figure 31.92 and Figure 31.93 show the port pin states during transition to the low power consumption state.

(2) Reception

(a) When Address match function is non used as condition of resumption (wake-up)

Before specifying the module stop state or making a transition to the low power consumption state, stop the receive operations (CCR0.RE = 0). If transition is made during data reception, the data being received will be invalid.

Figure 31.94 shows a sample flowchart for reception to software standby mode during reception.

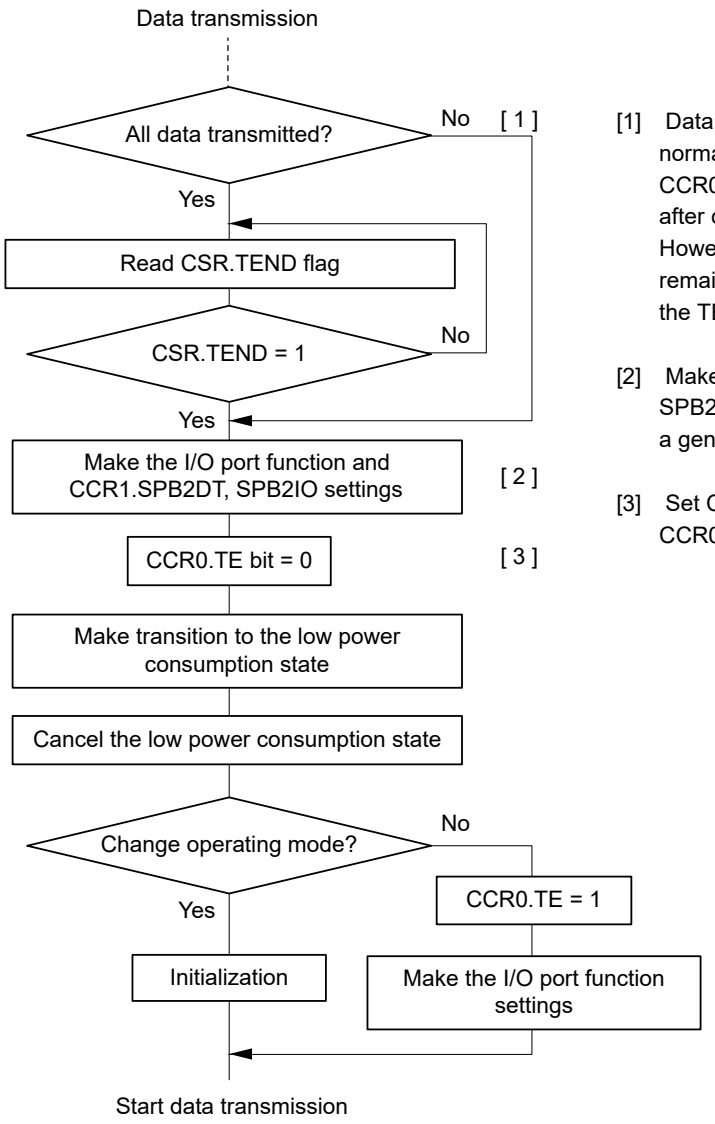
(b) When Address match function is used as condition of resumption (wake-up)

When using the power consumption reduction function to reduce SCI's power consumption, please do the following:

- Set the operation mode of after released.
- Set the compare data to CCR4.CMPD and set 1 to CCR0.DCME.
- Set the receive operation (CCR0.RE = 1).

When SCI transfers to the low power consumption mode if the reception data pin (RXD) is Low level, set CCR3.RXDESEL = 0. When setting it as CCR3.RXDESEL = 1, there is a possibility that a start bit (fall edge of RXDn pin) can not be detected at the time of low power consumption mode release.

Figure 31.95 shows a sample flowchart for reception to the low power consumption state during reception.



[1] Data being transmitted is lost halfway. Data can be normally transmitted from the CPU by setting the CCR0.TE to 1, reading CSR, and writing data to TDR after canceling the low power consumption state. However, if the DMAC has been activated, the data remaining in the DMAC will be transmitted when both the TE and TIE bits in CCR0 are set to 1.

[2] Make the I/O port function and CCR1.SPB2DT, SPB2IO settings to switch the TXDn pin to operate as a general I/O port.

[3] Set CCR0.TE bit to 0. Set also 0 to CCR0.TIE and CCR0.TEIE if you set these bits were 1.

Figure 31.91 Example of flowchart for transition to the low power consumption state during transmission

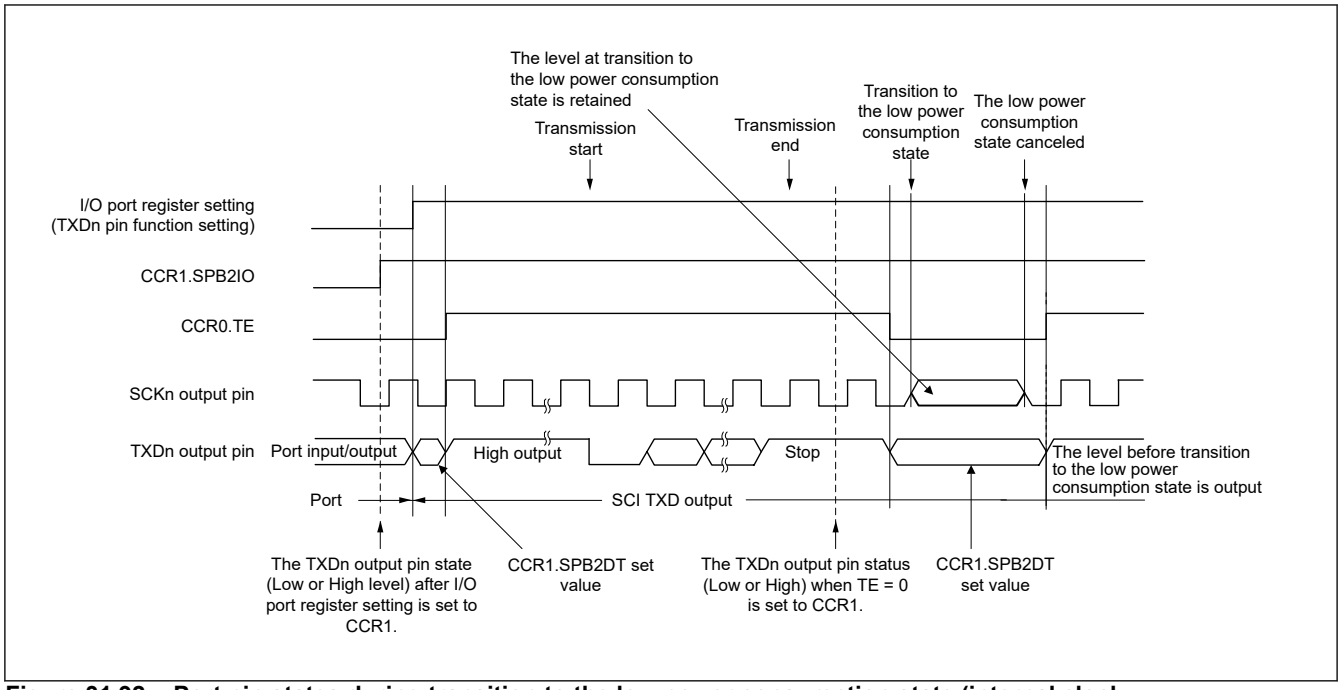


Figure 31.92 Port pin states during transition to the low power consumption state (internal clock, asynchronous transmission)

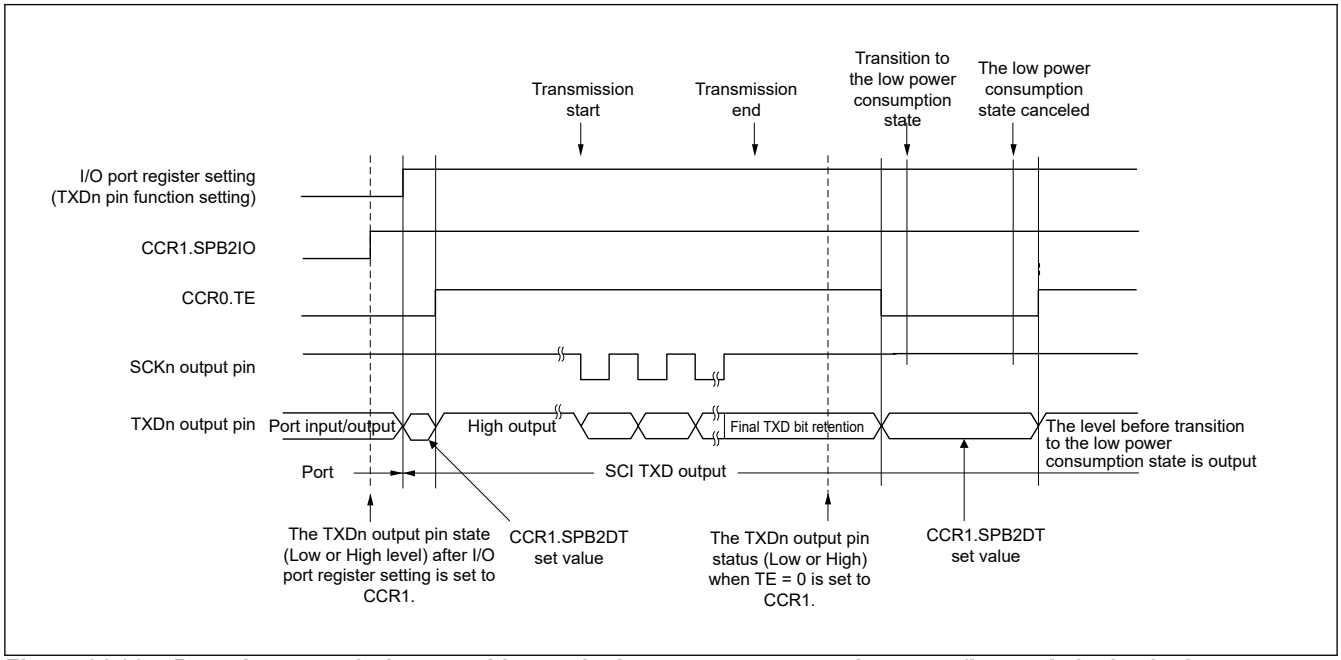


Figure 31.93 Port pin states during transition to the low power consumption state (internal clock, clock synchronous transmission)

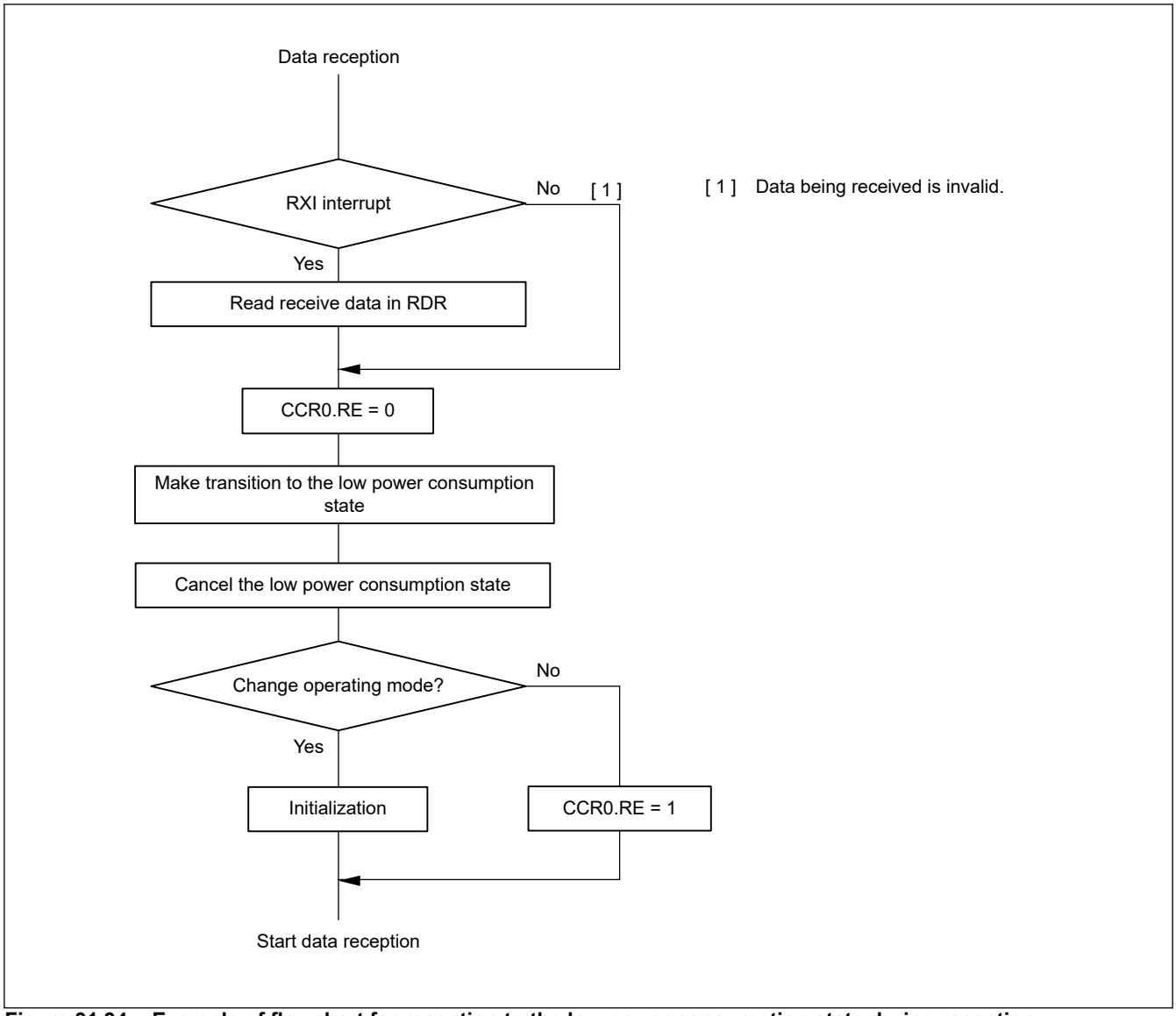


Figure 31.94 Example of flowchart for reception to the low power consumption state during reception

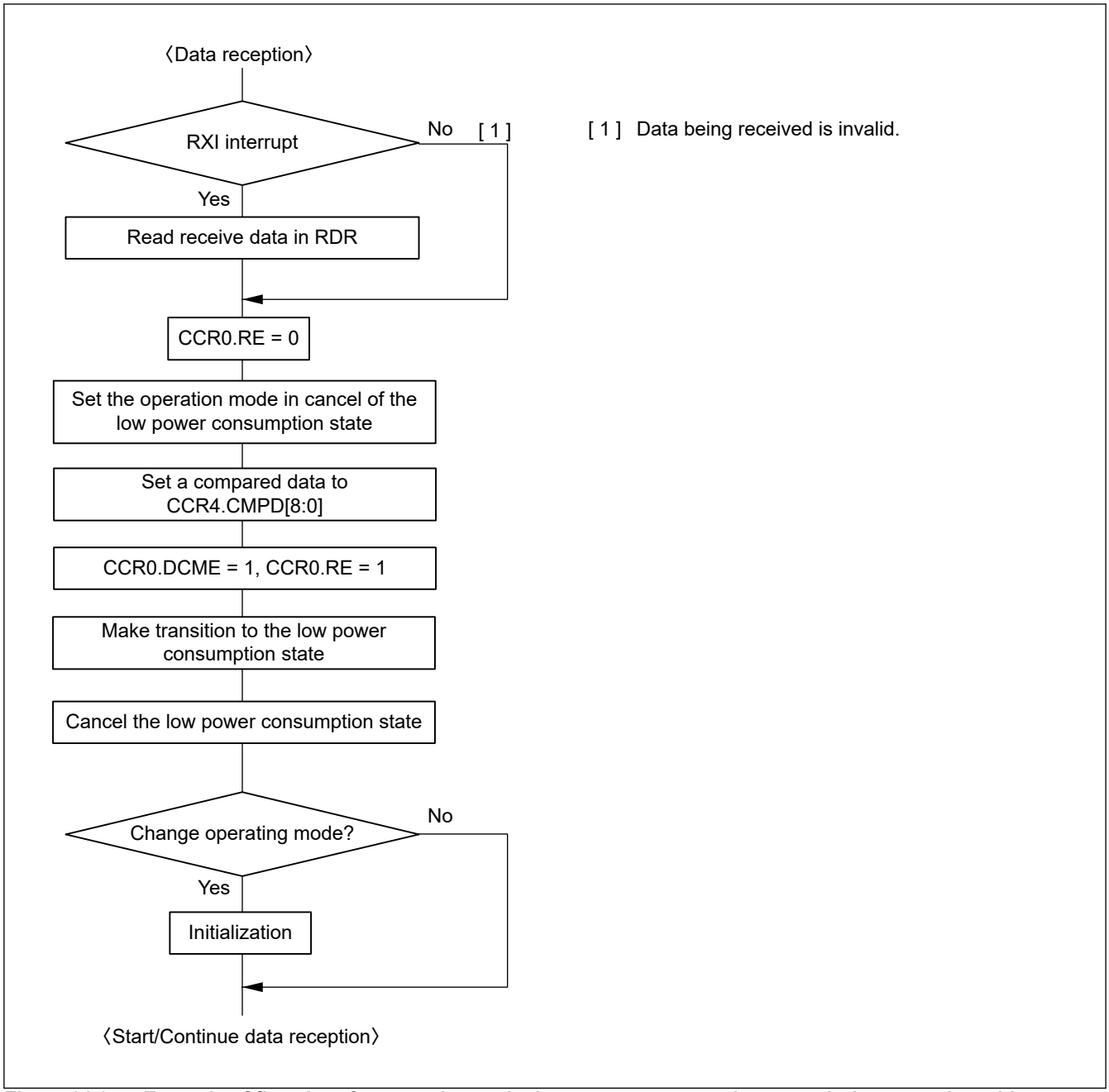


Figure 31.95 Example of flowchart for reception to the low power consumption state during reception with address match

31.16.2 Break Detection and Processing

(1) Non-FIFO selected

When a framing error is detected, a break can be detected by reading CSR.RXDMON bit value. In a break, the input from the RXDn pin becomes all 0s, and so the CSR.FER flag is set to 1 (framing error has occurred), and the CSR.PER flag may also be set to 1 (parity error has occurred). When the CCR3.RXDESEL bit is 0, the SCI continues the receive operation even after a break is received. Therefore, note that even if the FER flag is set to 0 (no framing error occurred), it will be set to 1 again. When the CCR3.RXDESEL bit is 1, the SCI sets the CSR.FER flag to 1 and stops receiving operation until a start bit of the next data frame is detected. If the CSR.FER flag is set to 0 at this time, the CSR.FER flag retains 0 during the break. When the RXDn pin is set to 1 and the break ends, detecting the beginning of the start bit at the first falling edge of the RXDn pin allows the SCI to start the receiving operation.

(2) FIFO selected

After a framing error is detected, when SCI detects that continuous receive data is 0 for 1 frame, the data stored into the receive FIFO (RDR) and reception stops. When a framing error is detected, a break can be detected by reading CSR.RXDMON bit value. After the RxD signal is in the mark state and it has finished the break, a stock of reception data to the receive FIFO (RDR) resumes.

31.16.3 Mark State and Production of Breaks

When CCR0.TE = 0 (serial transmission is disabled), the state of the TXDn pin can be set by CCR1.SPB2IO bit and CCR1.SPB2DT bit. Using this, it's possible to do a TXDn pin in the mark state and send a breakout. When you want to make a communication-line in the mark state (the state of 1) until the CCR0.TE bit is set to 1 (serial transmission is enabled), execute the following operations. First, set output level High by the CCR1.SPB2IO and CCR1.SPB2DT bits. Next, it's changed to a TXDn pin by I/O port function. On the other hand, if you want to send a break when sending data, set the CCR0.TE bit to 0 after setting the SPB2IO and SPB2DT bits in the CCR1 register to low level output. Setting the TE bit to 0 initializes the transmitter regardless of the current transmission status.

31.16.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode)

Transmission can be started by writing transmit-data to TDR even if CSR.ORER is 1. However, reception can not be started. Note also that the receive error flags cannot be set to 0 even if the CCR0.RE is set to 0 (serial reception is disabled).

31.16.5 Writing Data to TDR (Transmit FIFO)

(1) Non-FIFO selected

Data can be written to TDR anytime when CCR0.TE = 1. However, if new data is written to TDR when transmit data is remaining in TDR, the previous data in TDR is lost because it has not been transferred to TSR yet. If you use DMAC, be sure to write transmit data to TDR in the TXI interrupt request handling routine.

(2) FIFO selected

Data can be written to transmit FIFO (TDR) when TE is 1. Check the number of writable data with the FTSR.T[4:0] bits.

31.16.6 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode)

When the external clock source is used as a synchronization clock, the following restrictions apply.

(1) Start of transmission

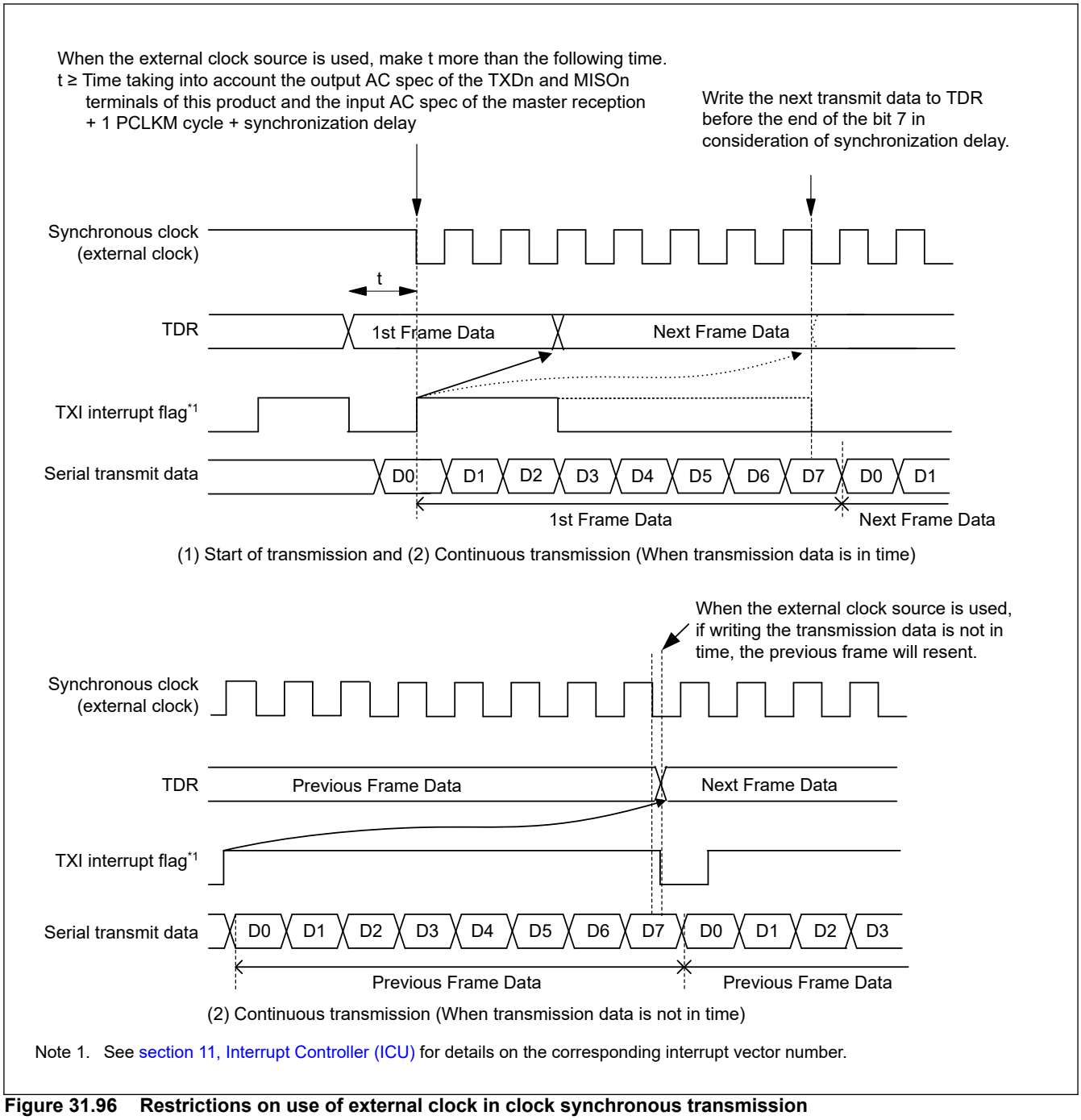
Update TDR by the CPU or DMAC and wait at least the following time until the start of the external clock input: (See Figure 31.96.)

Time taking into account the output AC specification of the TXDn and MISOn pins of this product and the input AC specification of the master reception + 1 PCLKM cycle + synchronization delay (2 to 3 PCLKSCIn).

(2) Continuous transmission

Write the next transmit data to TDR before the falling edge[*1] of the transmit clock for bit 7. Please write the transmit data to TDR in consideration of synchronization delay [2 to 3 PCLKSCIn]. If the transmit data can not be written in time, the previous frame data is resent. (See Figure 31.96.)

Note 1. When CCR3.CPOL = 1 and CCR3.CPHA = 0, or CCR3.CPOL = 0 and CCR3.CPHA = 1. In the case of CCR3.CPOL = 0 and CCR3.CPHA = 0, or CCR3.CPOL = 1 and CCR3.CPHA = 1, it is the rising edge.



Note 1. See section 11, Interrupt Controller (ICU) for details on the corresponding interrupt vector number.

Figure 31.96 Restrictions on use of external clock in clock synchronous transmission

31.16.7 Notes on Starting Transfer

At the point where transfer starts, when the interrupt status flag in the interrupt controller is 1, follow the procedure below to clear interrupt requests before permitting operations (by setting the CCR0.TE or CCR0.RE bit to 1).

For details on the interrupt status flag, see section 11, Interrupt Controller (ICU).

- Confirm that transfer has stopped (the setting of the CCR0.TE or CCR0.RE bits is 0).
- Set the corresponding interrupt enable bit (CCR0.TIE, CCR0.TEIE, or CCR0.RIE) to 0.
- Read the corresponding interrupt enable bit (CCR0.TIE, CCR0.TEIE, or CCR0.RIE bit) to check that it has actually become 0.
- Set the interrupt status flag in the interrupt controller to 0.

31.16.8 Limitations on Simple SPI Mode

(1) Master Mode

- Use a resistor to pull up or pull down the clock line matching the initial settings for the transfer clock set by the CCR3.CPHA and CPOL bits when CCR0.SSE = 1. This prevents the clock line from being placed in the high-impedance state when the CCR0.TE bit is set to 0 or unexpected edges from being generated on the clock line when the CCR0.TE bit is changed from 0 to 1.
  In a single-master mode, pull up or pull down the clock line is not necessary because the clock line does not become high-impedance state when CCR0.SSE = 0 and CCR0.TE = 0.
- In the case of the setting for clock delay (CCR3.CPHA = 0), the receive data full interrupt (RXI) is generated before the final clock edge on the SCKn pin as indicated in Figure 31.97. If the TE and RE bits become 0 at this time before the final edge of the clock signal on the SCKn pin, the SCKn pin is placed in the high-impedance state, so the width of the last clock pulse of the transfer clock is shortened. Furthermore, an RXI interrupt may lead to the input signal on the SSn# pin of a connected slave going to the high level before the final edge of the clock signal on the SCKn pin, leading to incorrect operation of the slave.
- In a multi-master mode, take care because the SCKn pin output becomes high-impedance while the input on the SSn# pin is at the low level if a mode fault error occurs as the current character is being transferred. And stopping supply of the clock signal to the connected slave. Remake the settings for the connected slave to avoid misaligned bits when transfer is restarted.

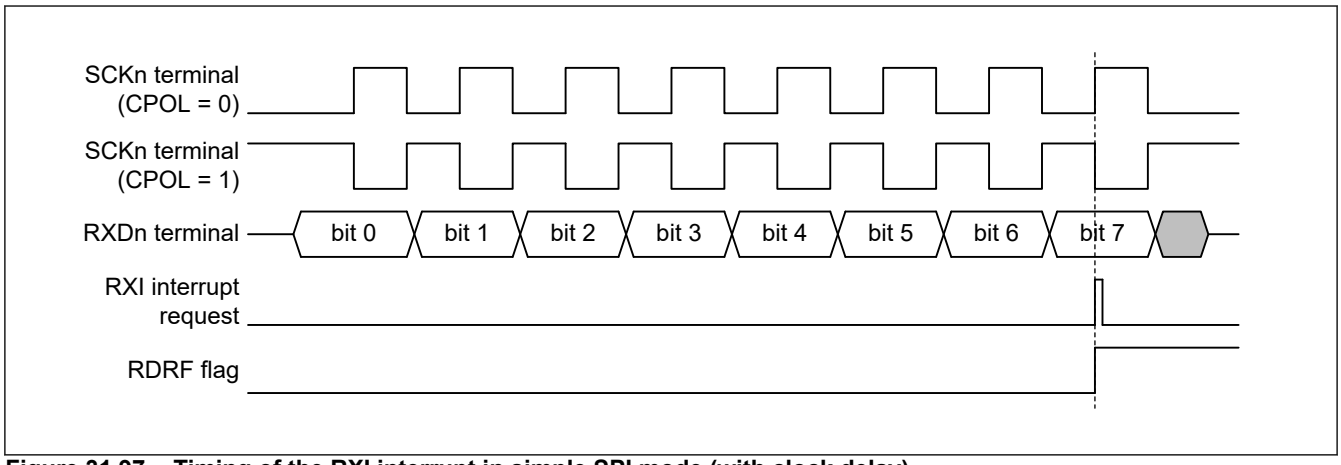


Figure 31.97 Timing of the RXI interrupt in simple SPI mode (with clock delay)

(2) Slave Mode

- It takes "1 PCLKM + synchronization delay time + data output delay time (AC spec)" from writing the transmit data to the TDR register until the data is output to the MSIOn pin. Take these into account when starting external clock input.
- Provide an external clock signal to the master the same as the data length for transfer.
- Secure the SS input setup time (AC spec) from the SS# low-level input to the start of external clock input.
- Control the input on the SSn# pin before the start and after the end of data transfer.
- When the level being input on the SSn# pin is to be changed from low to high while the current character is being transferred, the transmission and reception are stopped immediately. Set the TE and RE bits in the CCR0 to 0 and, after remaking the settings, restart transfer of the first byte.

31.16.9 Notes on Transmit Enable bit (TE)

In initial register value, when CCR0.TE = 0, and the terminal function is TXDn, the pin outputs high impedance.

So please make sure that the TXDn line will not be high impedance by the following one of ways.

1. The pull-up resistance is connected to the TXDn line.
2. Before CCR0.TE bit is 0, the function of the terminal is changed to general-purpose input port or output port. And after CCR0.TE bit is 1, the function of the pin is changed to TXDn.

3. In asynchronous mode, you can set CCR1 and decided level of TXDn pin during TE is 0.

In the simple SPI mode slave operation, the MSIOn pin operates in the same way as the above TXDn pin, so please deal with 1 or 2 in the same way. (3 can not be used.)

31.16.10 Notes on RS-485 Driver Control function

- RS-485 Driver control function is valid only in asynchronous mode.
- When RS-485 Driver control function is active (CCR3.DEN = 1), the TEND set timing/TEI output timing changes as follows. Wait for the TEI interrupt and set the TE bit in CCR0 to 0.
  When RS-485 Driver control function is inactive: When STOP bit output is completed.
  When RS-485 Driver control function is active: At the end of DE negation time.

31.16.11 Notes on Loopback function

The Loopback function is valid in asynchronous mode with internal clock and clock synchronous mode with internal clock.

31.16.12 Notes Regarding Register Access When Operation Clock (PCLKSCIn) Is Slower Than Bus Clock (PCLKM)

If the operating clock (PCLKSCIn) is slower than the bus clock (PCLKM), the interval for writing access to the same register should be at least PCLKSCIn × 3 cycles + PCLKM × 3 cycles. Synchronization delay time and signal processing time are required to ensure that the setting value of the first write access is propagated to the subsequent logic circuit and the circuit is controlled correctly.

31.16.13 Notes on interrupting operation

If 0 is written to CCR0.RE during data reception and the reception operation is interrupted, there is a possibility of an invalid state, so please do not use the received data (RDR register stored value) and the flag value of each status register. To interrupt the reception operation, stop the interrupt or event link reception side and then write 0 to the CCR0.RE bit.

In addition, when 0 is written to each operation enable bit (CCR0.RE, CCR0.TE) and the operation is interrupted, the following time is required to safely stop the internal circuit. During this time, access to this register is prohibited.

PCLKSCIn × 3 cycles + PCLKM × 3 cycles

31.16.14 Notes on communication mode transition

When changing the communication mode setting bits CCR3.MOD[2:0] and switching the communication mode, set CCR0.RE = 0 and CCR0.TE = 0 to stop the communication operation. In order to transition the internal circuit state safely, the register access interval should be at least PCLKSCIn × 3 cycles + PCLKM × 3 cycles. During this time, write access to this register is prohibited. As an example, the following describes the transition from Asynchronous mode to Clock-synchronous mode.

1. Confirm stop of reception and transmission at asynchronous mode (CCR3.MOD[2:0] = 000b).
2. Write 0 to CCR0.TE and CCR0.RE, stop communication.
3. Wait "PCLKSCIn × 3 cycles + PCLKM × 3 cycles". During this time, access to this register is prohibited.
4. Write 010b to CCR3.MOD[2:0].
5. Wait "PCLKSCIn × 3 cycles + PCLKM × 3 cycles". During this time, access to this register is prohibited.

32. I²C Bus Interface (IIC)

32.1 Overview

The LSI has a two-channel I²C Bus Interface (IIC) module that conforms with and provides a subset of the NXP I²C bus (inter-integrated circuit bus) interface functions. Table 32.1 lists the IIC specifications, Figure 32.1 shows a block diagram, and Figure 32.2 shows an example of I/O pin connections to external circuits, with an I²C bus configuration. Table 32.2 lists the I/O pins.

Table 32.1 IIC specifications (1 of 2)

Parameter	Specifications
Number of channels	3 channels (ch0 to ch1 in NONSAFETY and ch2 in SAFETY)
Communications format	• I²C bus format or SMBus format. • Master or slave mode selectable. • Automatic securing of the setup times, hold times, and bus-free times for the transfer rate.
Transfer rate	Fast-mode supported, up to 400 kbps.
SCL clock	For master operation, the duty cycle of the SCL clock is selectable in the range from 4% to 96%.
Issuing and detecting conditions	• Start, restart, and stop conditions are automatically generated. • Start conditions (including restart conditions) and stop conditions are detectable.
Slave address	• Configurable for up to three different slave addresses. • 7-bit and 10-bit address formats supported, including simultaneous use. • General call addresses, device ID addresses, and SMBus host addresses detectable.
Acknowledgment	• For transmission, automatic loading of the acknowledge bit. Transfer of the next transmit data can be automatically suspended on detection of a not-acknowledge bit. • For reception, automatic transmission of the acknowledge bit. If a wait between the 8th and 9th clock cycles is selected, software can control the value in the acknowledge field in response to the received value.
Wait function	During reception, the following wait periods are available by holding the SCL clock low: • Waiting between the 8th and 9th clock cycles. • Waiting between the 9th clock cycle and the 1st clock cycle of the next transfer.
SDA output delay function	Output timing of transmitted data, including the acknowledge bit, can be delayed.
Arbitration	• For multi-master operation: – SCL clock synchronization is possible when conflict occurs with the SCL signal from another master. – When issuing the start condition creates conflict on the bus, loss of arbitration is detected by testing for mismatching between the internal signal for the SDA line and the level on the SDA line. – In master operation, loss of arbitration is detected by testing for mismatching between the signal on the SDA line and the internal signal for the SDA line. • Loss of arbitration because the start condition occurs while the bus is busy is detectable, to prevent the issuing of double start conditions. • Loss of arbitration is detectable on transfer of a not-acknowledge bit because the internal signal for the SDA line and the level on the SDA line do not match. • Loss of arbitration because mismatching of internal and line levels for data is detectable in slave transmission.
Timeout function	Internal detection of long-interval stops of the SCL clock.
Noise cancellation	• Digital noise filters for both the SCL and SDA signals. • Programmable window for noise cancellation by the filters.
Interrupt sources	• Transfer error or event generation: arbitration detection, NACK, timeout, start or restart condition, or stop condition. • Receive data full, including matching with a slave address. • Transmit data empty, including matching with a slave address. • Transmit end.
Module-stop function	Module-stop state can be set to reduce power consumption.

Table 32.1 IIC specifications (2 of 2)

Parameter	Specifications
IIC operating modes	• Master transmit • Master receive • Slave transmit • Slave receive

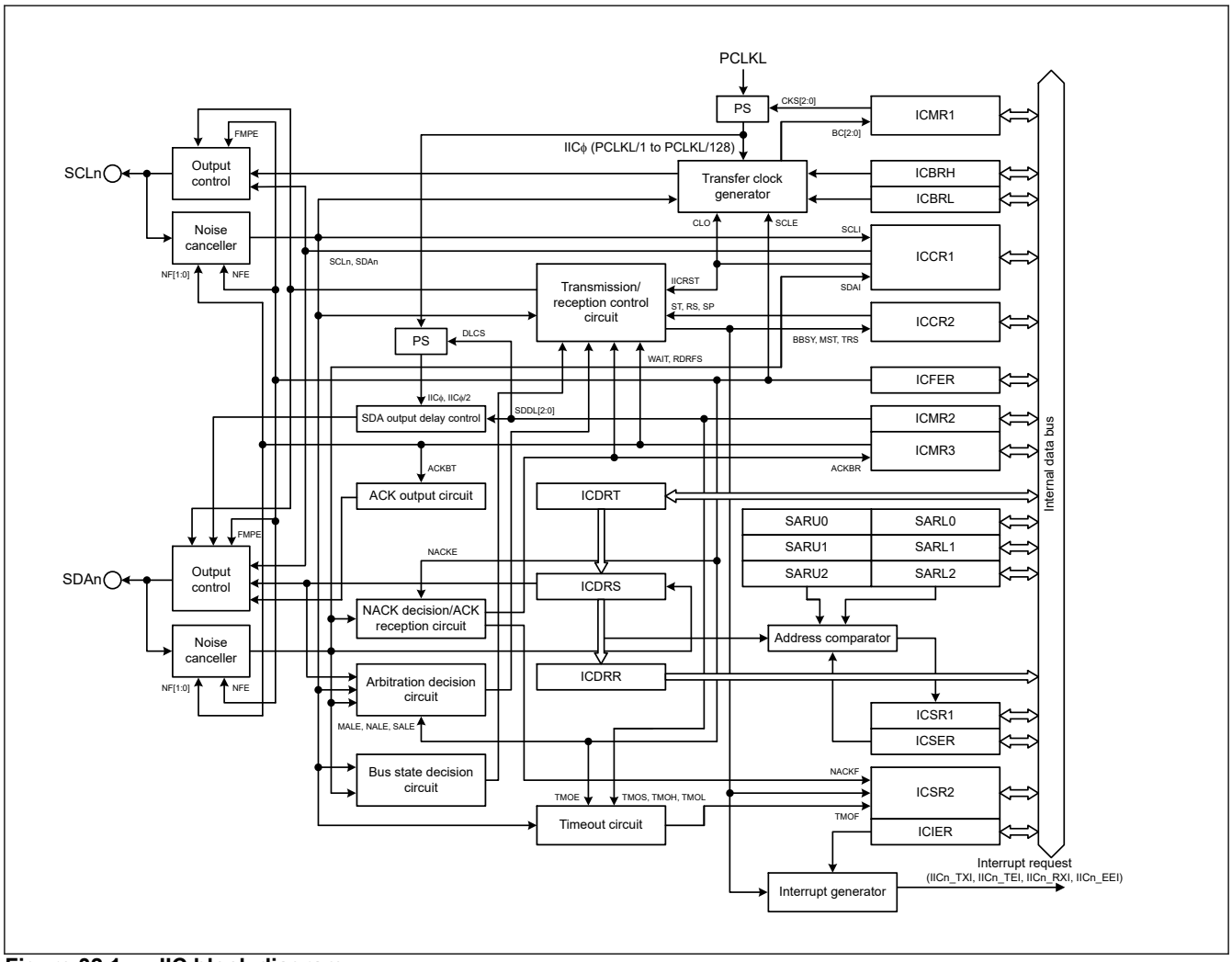


Figure 32.1 IIC block diagram

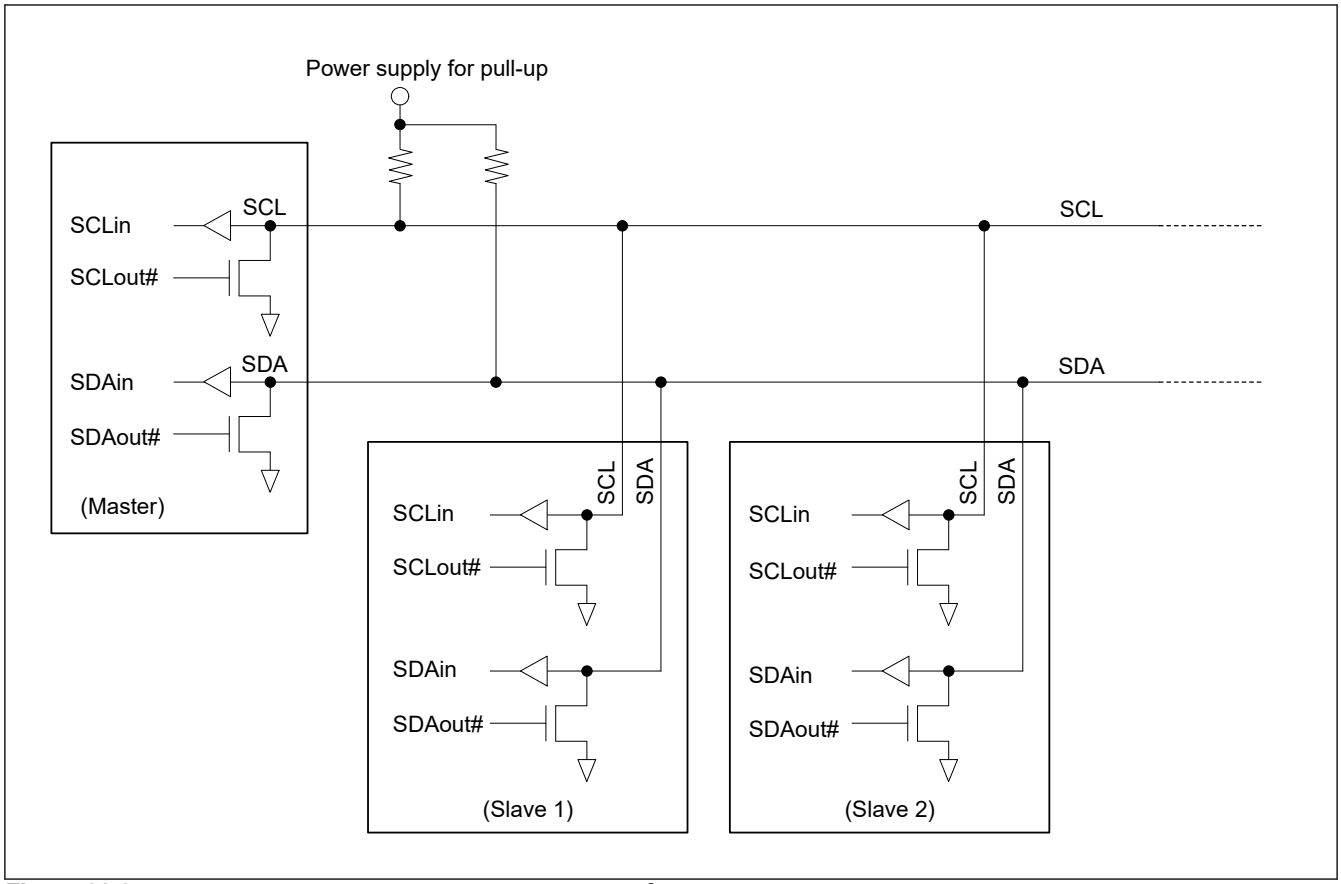


Figure 32.2 I/O pin connection to an external circuit (I^2C bus configuration example)

The input level of the signals for IIC is CMOS. Change to TTL is not supported.

Table 32.2 IIC I/O pins

Channel	Pin name*1	I/O	Function
IIC0	IIC_SCL0	I/O	IIC0 serial clock I/O pin
	IIC_SDA0	I/O	IIC0 serial data I/O pin
IIC1	IIC_SCL1	I/O	IIC1 serial clock I/O pin
	IIC_SDA1	I/O	IIC1 serial data I/O pin
IIC2	IIC_SCL2	I/O	IIC2 serial clock I/O pin
	IIC_SDA2	I/O	IIC2 serial data I/O pin

Note 1. For convention, pin names within this chapter are written without "IIC_" notation except this table.

Table 32.3 IIC interrupt sources

Name	Interrupt sources	CPU0 GIC request	DMAC activation
IICn_EEI	Transfer error or event generation	Possible	Not possible
IICn_RXI	Receive data full	Possible	Possible*1
IICn_TXI	Transmit data empty	Possible	Possible*1
IICn_TEI	Transmit end	Possible	Not possible

Note: n = 0 to 2
Note 1. DMAC activation is not available for n = 2.

32.2 Register Descriptions

32.2.1 ICCR1 : I2C Bus Control Register 1

Base address: IICn = 0x8004_3000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Function	R/W
0	SDAI	SDA Line Monitor 0: SDAn line is low 1: SDAn line is high	R
1	SCLI	SCL Line Monitor 0: SCLn line is low 1: SCLn line is high	R
2	SDAO	SDA Output Control/Monitor 0: Read: IIC drives SDAn pin low Write: IIC drives SDAn pin low 1: Read: IIC releases SDAn pin Write: IIC releases SDAn pin	R/W
3	SCLO	SCL Output Control/Monitor Use an external pull-up resistor to drive the signal high. 0: Read: IIC drives SCLn pin low Write: IIC drives SCLn pin low 1: Read: IIC releases SCLn pin Write: IIC releases SCLn pin	R/W
4	SOWP	SCLO/SDAO Write Protect 0: Write enable SCLO and SDAO bits 1: Write protect SCLO and SDAO bits	W
5	CLO	Extra SCL Clock Cycle Output This bit clears automatically after one clock cycle is output. 0: Do not output extra SCL clock cycle (default) 1: Output extra SCL clock cycle	R/W
6	IICRST	IIC-Bus Interface Internal Reset This setting clears the bit counter and the SCLn/SDAn output latch. 0: Release IIC reset or internal reset 1: Initiate IIC reset or internal reset	R/W
7	ICE	IIC-Bus Interface Enable Used in combination with the IICRST bit to select either IIC or internal reset. 0: Disable (SCLn and SDAn pins in inactive state) 1: Enable (SCLn and SDAn pins in active state)	R/W

SDAO bit (SDA Output Control/Monitor) and SCLO bit (SCL Output Control/Monitor)

The SDAO and SCLO bits directly control the SDAn and SCLn signals output from the IIC.

When writing to these bits, also write 0 to the SOWP bit. Setting these bits results in input to the IIC by the input buffer.

When slave mode is selected, a start condition might be detected and the bus might be released, depending on the bit settings.

Do not rewrite these bits during a start condition, stop condition, restart condition, or during transmission or reception.

Operation after rewriting under the specified conditions is not guaranteed. When reading these bits, the state of signals output from the IIC can be read.

CLO bit (Extra SCL Clock Cycle Output)

The CLO bit allows the output of an extra SCL clock cycle for debugging or error processing. Normally, set this bit to 0.

Setting this bit to 1 in a normal communication state causes a communication error. For details on this function, see section 32.11.2. Extra SCL Clock Cycle Output Function.

IICRST bit (IIC-Bus Interface Internal Reset)

The IICRST bit initiates an internal state reset of the IIC. Setting this bit to 1 initiates an IIC reset or internal reset.

Whether an IIC reset or internal reset is initiated is determined by setting this bit in combination with the ICE bit.

Table 32.4 lists the IIC resets.

The IIC reset initializes all registers except ICCR1.ICE and ICCR1.IICRST bits, and internal states of the IIC. In addition to the internal states of the IIC, the internal reset initializes the following:

- Bit counter (ICMR1.BC[2:0] bits)
- I^2C Bus Shift Register (ICDRS)
- I^2C Bus Status Registers (ICSR1 and ICSR2)
- SDAO and SCLO Output Control/Monitor (ICCR1.SCLO and ICCR1.SDAO bits)
- I^2C Bus Control Register 2 (except ICCR2.BBSY bit)

For the reset conditions of each register, see section 32.14. Resets, Registers, and Function States When Issuing Each Condition.

An internal reset initiated with the IICRST bit set to 1 during operation (with the ICE bit set to 1) resets the internal states of the IIC without initializing the port settings and the control and setting registers of the IIC when the bus or IIC hangs up because of a communication error. If the IIC hangs up in a low level output state, resetting the internal states cancels the low level output state and releases the bus with the SCLn pin and SDAn pin at high impedance.

Note: If an internal reset is initiated using the IICRST bit for a bus hang-up that occurs during communication with the master device in slave mode, the slave and master devices might enter different states, because the bit counter information differs. For this reason, do not initiate an internal reset in slave mode. Initiate recovery processing from the master device. If an internal reset is necessary because the IIC hangs with the SCLn line in a low level output state in slave mode, initiate an internal reset, and then issue a restart condition from the master device, or issue a stop condition and resume communication from the start condition. If communication is restarted by initiating a reset solely in the slave device without issuing a start or restart condition from the master device, synchronization is lost because the master and slave devices operate asynchronously.

Table 32.4 IIC resets

IICRST	ICE	State	Specifications
1	0	IIC reset	Resets all registers except ICCR1.ICE and ICCR1.IICRST bits, and the internal states of the IIC
	1	Internal reset	Resets the following: • ICMR1.BC[2:0] bits • ICCR1, ICSR2, ICDRS registers • SDAO and SCLO Output Control/Monitor (ICCR1.SCLO and ICCR1.SDAO bits) • I^2C Bus Control Register 2 (except ICCR2.BBSY bit) • Internal states of the IIC.

ICE bit (IIC-Bus Interface Enable)

The ICE bit selects the active or inactive state of the SCLn and SDAn pins. It can also be combined with the IICRST bit to initiate two types of resets. See Table 32.4 for the reset types.

Set the ICE bit to 1 when using the IIC. The SCLn and SDAn pins are placed in the active state when the ICE bit is set to 1. Set the ICE bit to 0 when the IIC is not used. The SCLn and SDAn pins are placed in the inactive state when the ICE bit is set to 0. Do not assign the SCLn or SDAn pin to the IIC when setting up the pin function control. Slave address comparison is performed if the pins are assigned to the IIC.

32.2.2 ICCR2 : I²C Bus Control Register 2

Base address: IICn = 0x8004_3000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	ST	Start Condition Issuance Request 0: Do not issue a start condition request 1: Issue a start condition request	R/W
2	RS	Restart Condition Issuance Request 0: Do not issue a restart condition request 1: Issue a restart condition request	R/W
3	SP	Stop Condition Issuance Request 0: Do not issue a stop condition request 1: Issue a stop condition request	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	TRS	Transmit/Receive Mode 0: Receive mode 1: Transmit mode	R/W*1
6	MST	Master/Slave Mode 0: Slave mode 1: Master mode	R/W*1
7	BBSY	Bus Busy Detection Flag 0: I²C bus released (bus free state) 1: I²C bus occupied (bus busy state)	R

Note 1. The MST and TRS bits can be written to when the ICMR1.MTWP bit is set to 1.

ST bit (Start Condition Issuance Request)

The ST bit requests transition to master mode and triggers a start condition.

When this bit is set to 1, a start condition is issued when the BBSY flag is set to 0 (bus free state). For details on issuing a start condition, see section 32.10. Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the ST bit.

[Clearing conditions]

- When 0 is written to the ST bit.
- When a start condition is issued (a start condition is detected).
- When the AL (arbitration-lost) flag in ICSR2 is set to 1.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Set the ST bit to 1 (start condition issuance request) when the BBSY flag is set to 0 (bus free state). Arbitration might be lost if the ST bit is set to 1 (start condition request) when the BBSY flag is 1 (bus busy state).

RS bit (Restart Condition Issuance Request)

The RS bit requests that a restart condition be issued in master mode.

When this bit is set to 1 to request a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode). For details on issuing a restart condition, see section 32.10. Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the RS bit with the BBSY flag in ICCR2 set to 1.

[Clearing conditions]

- When 0 is written to the RS bit.
- When a restart condition is issued (a start condition is detected).
- When the AL (arbitration-lost) flag in ICSR2 is set to 1.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Do not set the RS bit to 1 while issuing a stop condition.

Note: If 1 (restart condition request) is written to the RS bit in slave mode, the restart condition is not issued, but the RS bit remains set to 1. If the operating mode changes to master mode without the bit being cleared, a restart condition might be issued.

SP bit (Stop Condition Issuance Request)

The SP bit requests that a stop condition be issued in master mode.

When this bit is set to 1, a stop condition is issued when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode). For details on this function, see section 32.10. Start, Restart, and Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the SP bit with both the BBSY flag and the MST bit in ICCR2 set to 1.

[Clearing conditions]

- When 0 is written to the SP bit.
- When a stop condition is issued (a stop condition is detected).
- When the AL (arbitration-lost) flag in ICSR2 is set to 1.
- When a start condition and a restart condition are detected.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: Writing to the SP bit is not possible while the BBSY flag is 0 (bus free state).

Note: Do not set the SP bit to 1 while a restart condition is being issued.

TRS bit (Transmit/Receive Mode)

The TRS bit indicates transmit or receive mode.

The IIC is in receive mode when the TRS bit is 0 and in transmit mode when the bit is 1. The combination of the TRS bit and the MST bit indicates the operating mode of the IIC.

The value of the TRS bit automatically changes to 1 for transmit mode or 0 for receive mode when a start condition is issued or detected and the R/W# bit is set. Although writing to the TRS bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not required during normal usage.

[Setting conditions]

- When a start condition is issued normally because of a start condition request (when a start condition is detected with the ST bit set to 1).
- When a restart condition is issued normally because of a restart condition request (when a restart condition is detected with the RS bit set to 1).
- When the R/W# bit added to the slave address is set to 0 in master mode.
- When the address received in slave mode matches the address enabled in ICSER, with the R/W# bit set to 1.
- When 1 is written to the TRS bit with the MTWP bit in ICMR1 set to 1.

[Clearing conditions]

- When a stop condition is detected.
- When the AL (arbitration-lost) flag in ICSR2 is set to 1.
- In master mode, on the reception of a slave address to which a R/W# bit with the value 1 is appended.
- In slave mode, on a match between the received address and the address enabled in ICSER when the value of the received R/W# bit is 0, including when the received address is the general call address.
- In slave mode, when a restart condition is detected (a start condition is detected with ICCR2.BBSY = 1 and ICCR2.MST = 0).
- When 0 is written to the TRS bit with the MTWP bit in ICMR1 set to 1.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

MST bit (Master/Slave Mode)

The MST bit indicates master or slave mode. The IIC is in slave mode when the MST bit is set to 0 and is in master mode when the bit is set to 1. The combination of the MST bit and the TRS bit indicates the operating mode of the IIC.

The value of the MST bit automatically changes to 1 for master mode or 0 for slave mode when a start condition is issued, or when a stop condition is issued or detected. Although writing to the MST bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not necessary during normal usage.

[Setting conditions]

- When a start condition is issued normally because of a start condition request (when a start condition is detected with the ST bit set to 1).
- When 1 is written to the MST bit with the MTWP bit in ICMR1 set to 1.

[Clearing conditions]

- When a stop condition is detected.
- When the AL (arbitration-lost) flag in ICSR2 is set to 1.
- When 0 is written to the MST bit with the MTWP bit in ICMR1 set to 1.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

BBSY bit (Bus Busy Detection Flag)

The BBSY flag indicates whether the I^2C bus is occupied (bus busy state) or released (bus free state).

The flag is set to 1 when the SDAn line changes from high to low when the SCLn line is high, assuming that a start condition was issued.

The flag is set to 0 when the SDAn line changes from low to high with the SCLn line high, if the bus free time (ICBRL setting) start condition is not detected, assuming that a stop condition was issued.

[Setting condition]

- When a start condition is detected.

[Clearing conditions]

- When the bus free time (ICBRL setting) start condition is not detected after detecting a stop condition.
- When 1 is written to the IICRST bit in ICCR1 with the ICE bit in ICCR1 set to 0 (IIC reset).

32.2.3 ICMR1 : I2C Bus Mode Register 1

Base address: IICn = 0x8004_3000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x02

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MTWP	CKS[2:0]			BCWP	BC[2:0]		
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
2:0	BC[2:0]	Bit Counter 0 0 0: 9 bits 0 0 1: 2 bits 0 1 0: 3 bits 0 1 1: 4 bits 1 0 0: 5 bits 1 0 1: 6 bits 1 1 0: 7 bits 1 1 1: 8 bits	R/W*1
3	BCWP	BC Write Protect 0: Write enable BC[2:0] bits 1: Write protect BC[2:0] bits	W*1
6:4	CKS[2:0]	Internal Reference Clock Select Select the internal reference clock source (IICΦ) for the IIC. 0 0 0: PCLKL clock 0 0 1: PCLKL/2 clock 0 1 0: PCLKL/4 clock 0 1 1: PCLKL/8 clock 1 0 0: PCLKL/16 clock 1 0 1: PCLKL/32 clock 1 1 0: PCLKL/64 clock 1 1 1: PCLKL/128 clock	R/W
7	MTWP	MST/TRS Write Protect 0: Write protect MST and TRS bits in ICCR2 1: Write enable MST and TRS bits in ICCR2	R/W

Note 1. Rewrite the BC[2:0] bits and set the BCWP bit to 0 at the same time.

BC[2:0] bits (Bit Counter)

The BC[2:0] bits function as a counter that indicates the number of bits remaining to be transferred on detection of a rising edge on the SCLn line. Although BC[2:0] are read/write bits, it is not required to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one, for an additional acknowledge bit, between transferred frames when the SCLn line is at a low level.

The value in the BC[2:0] bits return to 000b at the end of a data transfer, including the acknowledge bit, or when a start or restart condition is detected.

32.2.4 ICMR2 : I2C Bus Mode Register 2

Base address: IICn = 0x8004_3000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Value after reset:	0	0	0	0	0	1	1	0

Bit	Symbol	Function	R/W
0	TMOS	Timeout Detection Time Select 0: Select long mode 1: Select short mode	R/W
1	TMOL	Timeout L Count Control 0: Disable count while SCLn line is low 1: Enable count while SCLn line is low	R/W
2	TMOH	Timeout H Count Control 0: Disable count while SCLn line is high 1: Enable count while SCLn line is high	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	SDDL[2:0]	SDA Output Delay Counter 0 0 0: No output delay 0 0 1: 1 IICΦ cycle (When ICMR2.DLCS = 0 (IICΦ)) 1 or 2 IICΦ cycles (When ICMR2.DLCS = 1 (IICΦ/2)) 0 1 0: 2 IICΦ cycles (When ICMR2.DLCS = 0 (IICΦ)) 3 or 4 IICΦ cycles (When ICMR2.DLCS = 1 (IICΦ/2)) 0 1 1: 3 IICΦ cycles (When ICMR2.DLCS = 0 (IICΦ)) 5 or 6 IICΦ cycles (When ICMR2.DLCS = 1 (IICΦ/2)) 1 0 0: 4 IICΦ cycles (When ICMR2.DLCS = 0 (IICΦ)) 7 or 8 IICΦ cycles (When ICMR2.DLCS = 1 (IICΦ/2)) 1 0 1: 5 IICΦ cycles (When ICMR2.DLCS = 0 (IICΦ)) 9 or 10 IICΦ cycles (When ICMR2.DLCS = 1 (IICΦ/2)) 1 1 0: 6 IICΦ cycles (When ICMR2.DLCS = 0 (IICΦ)) 11 or 12 IICΦ cycles (When ICMR2.DLCS = 1 (IICΦ/2)) 1 1 1: 7 IICΦ cycles (When ICMR2.DLCS = 0 (IICΦ)) 13 or 14 IICΦ cycles (When ICMR2.DLCS = 1 (IICΦ/2))	R/W
7	DLCS	SDA Output Delay Clock Source Select 0: Select internal reference clock (IICΦ) as clock source for SDA output delay counter 1: Select internal reference clock divided by 2 (IICΦ/2) as clock source for SDA output delay counter*1	R/W

Note 1. The setting DLCS = 1 (IICΦ/2) is only valid when SCL is low. When SCL is high, the DLCS = 1 setting becomes invalid and the clock source becomes the internal reference clock (IICΦ).

TMOS bit (Timeout Detection Time Select)

The TMOS bit selects long or short mode for the timeout detection time when the timeout function is enabled (ICFER.TMOE = 1).

When this bit is set to 0, long mode is selected. When it is set to 1, short mode is selected.

In long mode, the timeout detection internal counter functions as a 16 bit-counter. In short mode, the counter functions as a 14 bit counter. While the SCLn line is in the state that enables this counter as specified in the TMOH and TMOL bits, the counter counts up in synchronization with the internal reference clock (IICΦ) as a count source.

For details on this function, see section 32.11.1. Timeout Function.

TMOL bit (Timeout L Count Control)

The TMOLbit enables or disables up-counting on the internal counter of the timeout function while the SCLn line is held low and the timeout function is enabled (ICFER.TMOE = 1).

TMOH bit (Timeout H Count Control)

The TMOH bit enables or disables up-counting on the internal counter of the timeout function while the SCLn line is held high and the timeout function is enabled (ICFER.TMOE = 1).

SDDL[2:0] bits (SDA Output Delay Counter)

The SDDL[2:0] bits can be used to delay the SDA output. This counter works with the clock source selected in the DLCS bit. This setting can be used for all types of SDA output, including the transmission of the acknowledge bit.

Set the SDA output delay to meet the I^2C bus standard for the data enable time/acknowledge enable time,*1 or the SMBus standard, within [data hold time (300 ns or more + the SCL-clock low-level period) - the data setup time (250 ns)]. If a

value outside the standard is set, communication between the devices might malfunction or falsely indicate a start or stop condition, depending on the bus state.

For details on this function, see section 32.5. SDA Output Delay Function.

Note 1. Data enable time/acknowledge enable time
3,450 ns for up to 100 kbps: Standard-mode (Sm)
900 ns for up to 400 kbps: Fast-mode (Fm)

32.2.5 ICMR3 : I2C Bus Mode Register 3

Base address: IICn = 0x8004_3000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SMBS	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	NF[1:0]	Noise Filter Stage Select 0 0: Filter out noise of up to 1 IICΦ cycle (single-stage filter) 0 1: Filter out noise of up to 2 IICΦ cycles (2-stage filter) 1 0: Filter out noise of up to 3 IICΦ cycles (3-stage filter) 1 1: Filter out noise of up to 4 IICΦ cycles (4-stage filter)	R/W
2	ACKBR	Receive Acknowledge 0: 0 received as the acknowledge bit (ACK reception) 1: 1 received as the acknowledge bit (NACK reception)	R
3	ACKBT	Transmit Acknowledge 0: Send 0 as the acknowledge bit (ACK transmission) 1: Send 1 as the acknowledge bit (NACK transmission)	R/W*1
4	ACKWP	ACKBT Write Protect 0: Write protect ACKBT bit 1: Write enable ACKBT bit	R/W*1
5	RDRFS	RDRF Flag Set Timing Select Low-hold is released by writing to ACKBT. 0: Set the RDRF flag on the rising edge of the 9th SCL clock cycle. The SCLn line is not held low on the falling edge of the 8th clock cycle. 1: Set the RDRF flag on the rising edge of the 8th SCL clock cycle. The SCLn line is held low on the falling edge of the 8th clock cycle.	R/W*2
6	WAIT	WAIT Low-hold is released by reading ICDRR. 0: No wait. SCLn is not held low during the period between 9th clock cycle and 1st clock cycle. 1: Wait. SCLn is held low during the period between 9th clock cycle and 1st clock cycle.	R/W*2
7	SMBS	SMBus/IIC-Bus Select 0: Select I2C bus 1: Select SMBus	R/W

Note 1. Write to the ACKBT bit only while the ACKWP bit is already 1. If the application writes 1 to both the ACKWP and ACKBT bits at the same time, the ACKBT bit is not set to 1.
Note 2. The WAIT and RDRFS bits are only valid in receive mode (invalid in transmit mode).

NF[1:0] bits (Noise Filter Stage Select)

The NF[1:0] bits select the number of stages in the digital noise filter. For details on this function, see section 32.6. Digital Noise Filter Circuits.

Note: Set the noise range to be filtered out by the noise filter within a range less than the SCLn line high-level period or low-level period. If the noise range is set to a value of [SCL clock width: high-level period or low-level period,

whichever is shorter] - [1.5 internal reference clock (IICΦ) cycles] or more, the SCL clock is regarded as noise by the noise filter function of the IIC, which might prevent the IIC from operating normally.

ACKBR bit (Receive Acknowledge)

The ACKBR bit stores the acknowledge bit information received from the receive device in transmit mode.

[Setting condition]

- When 1 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1.

[Clearing conditions]

- When 0 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1.
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (IIC reset).

ACKBT bit (Transmit Acknowledge)

The ACKBT bit sets the acknowledge bit to be sent in receive mode.

[Setting condition]

- When 1 is written to this bit with the ACKWP bit set to 1.

[Clearing conditions]

- When 0 is written to this bit with the ACKWP bit set to 1.
- When stop condition issuance is detected with the SP bit in ICCR2 set to 1.
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (IIC reset).

ACKWP bit (ACKBT Write Protect)

The ACKWP bit controls write enabling of the ACKBT bit.

RDRFS bit (RDRF Flag Set Timing Select)

The RDRFS bit selects the RDRF flag set timing in receive mode and also selects whether to hold the SCLn line low on the falling edge of the 8th SCL clock cycle.

When the RDRFS bit is 0, the SCLn line is not held low on the falling edge of the 8th SCL clock cycle, and the RDRF flag is set to 1 on the rising edge of the 9th SCL clock cycle.

When the RDRFS bit is 1, the RDRF flag is set to 1 on the rising edge of the 8th SCL clock cycle, and the SCLn line is held low on the falling edge of the 8th SCL clock cycle. The low-hold of the SCLn line is released by a write to the ACKBT bit.

After data is received with this setting, the SCLn line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT = 0) or NACK (ACKBT = 1), based on the receive data.

WAIT bit (WAIT)

The WAIT bit controls whether to forcefully hold the period between the 9th SCL clock cycle and the 1st SCL clock cycle low until the receive data buffer (ICDRR) is completely read each time a single-byte data is received in receive mode.

When the WAIT bit is 0, the receive operation is continued without holding the period between the 9th and the 1st SCL clock cycle low. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.

When the WAIT bit is 1, the SCLn line is held low from the falling edge of the 9th clock cycle until the ICDRR value is read each time a single-byte data is received. This enables the receive operation in byte units.

Note: When the value of the WAIT bit is to be read, be sure to first read the ICDRR.

SMBS bit (SMBus/IIC-Bus Select)

Setting the SMBS bit to 1 selects the SMBus and enables the HOAE bit in ICSER.

32.2.6 ICFER : I²C Bus Function Enable Register

Base address: IICn = 0x8004_3000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x05

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Value after reset:	0	1	1	1	0	0	1	0

Bit	Symbol	Function	R/W
0	TMOE	Timeout Function Enable 0: Disable 1: Enable	R/W
1	MALE	Master Arbitration-Lost Detection Enable 0: Disable the arbitration-lost detection function and disable automatic clearing of the MST and TRS bits in ICCR2 when arbitration is lost 1: Enable the arbitration-lost detection function and enable automatic clearing of the MST and TRS bits in ICCR2 when arbitration is lost	R/W
2	NALE	NACK Transmission Arbitration-Lost Detection Enable 0: Disable 1: Enable	R/W
3	SALE	Slave Arbitration-Lost Detection Enable 0: Disable 1: Enable	R/W
4	NACKE	NACK Reception Transfer Suspension Enable 0: Do not suspend transfer operation during NACK reception (disable transfer suspension) 1: Suspend transfer operation during NACK reception (enable transfer suspension)	R/W
5	NFE	Digital Noise Filter Circuit Enable 0: Do not use the digital noise filter circuit 1: Use the digital noise filter circuit	R/W
6	SCLE	SCL Synchronous Circuit Enable 0: Do not use the SCL synchronous circuit 1: Use the SCL synchronous circuit	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W

TMOE bit (Timeout Function Enable)

The TMOE bit enables or disables the timeout function. For details on this function, see section 32.11.1. Timeout Function.

MALE bit (Master Arbitration-Lost Detection Enable)

The MALE bit specifies whether to use the arbitration-lost detection function in master mode. For normal operation, set this bit to 1.

NALE bit (NACK Transmission Arbitration-Lost Detection Enable)

The NALE bit specifies whether to cause loss of arbitration when ACK is detected during the transmission of NACK in receive mode, for example, when slaves with the same address exist on the bus, or when two or more masters select the same slave device simultaneously with a different number of receive bytes.

SALE bit (Slave Arbitration-Lost Detection Enable)

The SALE bit specifies whether to cause loss of arbitration when a value different from the value being transmitted is detected on the bus in slave transmit mode, for example, when slaves with the same address exist on the bus, or when a mismatch with the transmit data occurs because of noise.

NACKE bit (NACK Reception Transfer Suspension Enable)

The NACKE bit specifies whether to continue or discontinue the transfer operation when NACK is received from the slave device in transmit mode. For normal operation, set this bit to 1.

When NACK is received with the NACKE bit set to 1, the next transfer operation is suspended. When the NACKE bit is 0, the next transfer operation is continued regardless of the received acknowledge content.

For details, see section 32.8.2. NACK Reception Transfer Suspension Function.

SCLE bit (SCL Synchronous Circuit Enable)

The SCLE bit specifies whether to synchronize the SCL clock with the SCL input clock. For normal operation, set this bit to 1.

When the SCLE bit is set to 0 (no SCL synchronous circuit used), the IIC does not synchronize the SCL clock with the SCL input clock. With this setting, the IIC outputs the SCL clock with the transfer rate set in ICBRH and ICBRL, regardless of the SCLn line state. For this reason, if the bus load of the I^2C bus line is much larger than the specification value, or if the SCL clock output overlaps in multiple masters, a short-cycle SCL clock that does not meet the specification might be output. When no SCL synchronous circuit is used, it also affects the issuance of the start, restart, and stop conditions, and the continuous output of extra SCL clock cycles.

Do not set this bit to 0 except when checking the output of the set transfer rate.

32.2.7 ICSER : I^2C Bus Status Enable Register

Base address: IICn = 0x8004_3000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	HOAE	—	DIDE	—	GCAE	SAR2E	SAR1E	SAR0E
Value after reset:	0	0	0	0	1	0	0	1

Bit	Symbol	Function	R/W
0	SAR0E	Slave Address Register 0 Enable 0: Disable slave address in SARL0 and SARU0 1: Enable slave address in SARL0 and SARU0	R/W
1	SAR1E	Slave Address Register 1 Enable 0: Disable slave address in SARL1 and SARU1 1: Enable slave address in SARL1 and SARU1	R/W
2	SAR2E	Slave Address Register 2 Enable 0: Disable slave address in SARL2 and SARU2 1: Enable slave address in SARL2 and SARU2	R/W
3	GCAE	General Call Address Enable 0: Disable general call address detection 1: Enable general call address detection	R/W
4	—	This bit is read as 0. The write value should be 0.	R/W
5	DIDE	Device ID Address Detection Enable 0: Disable device ID address detection 1: Enable device ID address detection	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	HOAE	Host Address Enable 0: Disable host address detection 1: Enable host address detection	R/W

SARyE bit (Slave Address Register y Enable) (y = 0 to 2)

The SARyE bit enables or disables the received slave address and the slave address set in SARLy and SARUy.

When this bit is set to 1, the slave address set in SARLy and SARUy is enabled and is compared with the received slave address. When this bit is set to 0, the slave address set in SARLy and SARUy is disabled and is ignored even if it matches the received slave address.

GCAE bit (General Call Address Enable)

The GCAE bit specifies whether to ignore the general call address (0000 000b + 0 [W]: All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the IIC recognizes the received slave address as the general call address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2), and performs the data receive operation. When this bit is set to 0, the received slave address is ignored even if it matches the

general call address.

DIDE bit (Device ID Address Detection Enable)

The DIDE bit specifies whether to recognize and execute the device ID address when a device ID (1111 100b) is received in the first frame after a start or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the IIC recognizes that the device ID address was received. When the next R/W# bit is 0 (W), the IIC recognizes the second and the subsequent frames as slave addresses and continues the receive operation. When this bit is set to 0, the IIC ignores the received first frame even if it matches the device ID address, and recognizes the first frame as a normal slave address.

For details on the device ID function, see section 32.7.3. Device ID Address Detection.

HOAE bit (Host Address Enable)

The HOAE bit specifies whether to ignore the received host address (0001 000b) when the SMBS bit in ICMR3 is 1.

When this bit is set to 1 while the SMBS bit in ICMR3 is 1, if the received slave address matches the host address, the IIC recognizes the received slave address as the host address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2), and performs the receive operation.

When the SMBS bit in ICMR3 or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

32.2.8 ICIER : I2C Bus Interrupt Enable Register

Base address: IICn = 0x8004_3000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x07

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMOIE	Timeout Interrupt Request Enable 0: Disable timeout interrupt IICn_EEI (TMOF) request 1: Enable timeout interrupt IICn_EEI (TMOF) request	R/W
1	ALIE	Arbitration-Lost Interrupt Request Enable 0: Disable arbitration-lost interrupt IICn_EEI (AL) request 1: Enable arbitration-lost interrupt IICn_EEI (AL) request	R/W
2	STIE	Start Condition Detection Interrupt Request Enable 0: Disable start condition detection interrupt IICn_EEI (START) request 1: Enable start condition detection interrupt IICn_EEI (START) request	R/W
3	SPIE	Stop Condition Detection Interrupt Request Enable 0: Disable stop condition detection interrupt IICn_EEI (STOP) request 1: Enable stop condition detection interrupt IICn_EEI (STOP) request	R/W

Bit	Symbol	Function	R/W
4	NAKIE	NACK Reception Interrupt Request Enable 0: Disable NACK reception interrupt IICn_EEI (NACKF) request 1: Enable NACK reception interrupt IICn_EEI (NACKF) request	R/W
5	RIE	Receive Data Full Interrupt Request Enable 0: Disable receive data full interrupt (IICn_RXI) request 1: Enable receive data full interrupt (IICn_RXI) request	R/W
6	TEIE	Transmit End Interrupt Request Enable 0: Disable transmit end interrupt (IICn_TEI) request 1: Enable transmit end interrupt (IICn_TEI) request	R/W
7	TIE	Transmit Data Empty Interrupt Request Enable 0: Disable transmit data empty interrupt (IICn_TXI) request 1: Enable transmit data empty interrupt (IICn_TXI) request	R/W

TMOIE bit (Timeout Interrupt Request Enable)

The TMOIE bit enables or disables timeout interrupt IICn_EEI (TMOF) requests when the TMOF flag in ICSR2 is 1. To cancel a IICn_EEI (TMOF) interrupt request, set the TMOF flag or the TMOIE bit to 0.

ALIE bit (Arbitration-Lost Interrupt Request Enable)

The ALIE bit enables or disables arbitration-lost interrupt IICn_EEI (AL) requests when the AL flag in ICSR2 is 1. To cancel an IICn_EEI (AL) interrupt request, set the AL flag or the ALIE bit to 0.

STIE bit (Start Condition Detection Interrupt Request Enable)

The STIE bit enables or disables start condition detection interrupt IICn_EEI (START) requests when the START flag in ICSR2 is 1. To cancel an IICn_EEI (START) interrupt request, set the START flag or the STIE bit to 0.

SPIE bit (Stop Condition Detection Interrupt Request Enable)

The SPIE bit enables or disables stop condition detection interrupt IICn_EEI (STOP) requests when the STOP flag in ICSR2 is 1. To cancel an IICn_EEI (STOP) interrupt request, set the STOP flag or the SPIE bit to 0.

NAKIE bit (NACK Reception Interrupt Request Enable)

The NAKIE bit enables or disables NACK reception interrupt IICn_EEI (NACKF) requests when the NACKF flag in ICSR2 is 1. To cancel an IICn_EEI (NACKF) interrupt request, set the NACKF flag or the NAKIE bit to 0.

RIE bit (Receive Data Full Interrupt Request Enable)

The RIE bit enables or disables receive data full interrupt (IICn_RXI) requests when the RDRF flag in ICSR2 is 1.

TEIE bit (Transmit End Interrupt Request Enable)

The TEIE bit enables or disables transmit end interrupt (IICn_TEI) requests when the TEND flag in ICSR2 is 1. To cancel an IICn_TEI interrupt request, set the TEND flag or the TEIE bit to 0.

TIE bit (Transmit Data Empty Interrupt Request Enable)

The TIE bit enables or disables transmit data empty interrupt (IICn_TXI) requests when the TDRE flag in ICSR2 is 1.

32.2.9 ICSR1 : I²C Bus Status Register 1

Base address: IICn = 0x8004_3000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AAS0	Slave Address 0 Detection Flag 0: Slave address 0 not detected 1: Slave address 0 detected	R/W*1
1	AAS1	Slave Address 1 Detection Flag 0: Slave address 1 not detected 1: Slave address 1 detected	R/W*1
2	AAS2	Slave Address 2 Detection Flag 0: Slave address 2 not detected 1: Slave address 2 detected	R/W*1
3	GCA	General Call Address Detection Flag 0: General call address not detected 1: General call address detected	R/W*1
4	—	This bit is read as 0. The write value should be 0.	R/W
5	DID	Device ID Address Detection Flag This bit is set to 1 when the first frame received immediately after a start condition is detected, matches a value of (device ID (1111 100b) + 0 [W]). 0: Device ID command not detected 1: Device ID command detected	R/W*1
6	—	This bit is read as 0. The write value should be 0.	R/W
7	HOA	Host Address Detection Flag This bit is set to 1 when the received slave address matches the host address (0001 000b). 0: Host address not detected 1: Host address detected	R/W*1

Note 1. Only 0 can be written to clear the flag.

AASy bit (Slave Address y Detection Flag) (y = 0 to 2)

The AASy flag indicates whether slave address y was detected.

[Setting conditions]

For 7-bit address format (SARUy.FS = 0):

- When the received slave address matches the SVA[6:0] value in SARLy, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 1 on the rising edge of the 9th SCL clock cycle in the frame.

For 10-bit address format: (SARUy.FS = 1):

- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy), and the subsequent address matches the SARLy value, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 1 on the rising edge of the 9th SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the AASy flag after reading AASy = 1
- When a stop condition is detected.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

For 7-bit address format (SARUy.FS = 0):

- When the received slave address does not match the SVA[6:0] value in SARLy, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 0 on the rising edge of the 9th SCL clock cycle in the frame.

For 10-bit address format (SARUy.FS = 1):

- When the received slave address does not match a value of 11110b + SVA[1:0] in SARUy, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 0 on the rising edge of the 9th SCL clock cycle in the frame.

- When the received slave address matches a value of 11110b + SVA[1:0] in SARUy, and the subsequent address does not match the SARLy value, with the SARyE bit in ICSER set to 1 (slave address y detection enabled).
  The AASy flag is set to 0 on the rising edge of the 9th SCL clock cycle in the frame.

GCA bit (General Call Address Detection Flag)

The GCA flag indicates whether the general call address was detected.

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 [W]), with the GCAE bit in ICSER set to 1 (general call address detection enabled).
  The GCA flag is set to 1 on the rising edge of the 9th SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA flag after reading GCA = 1.
- When a stop condition is detected.
- When the received slave address does not match the general call address (0000 000b + 0 [W]), with the GCAE bit in ICSER set to 1 (general call address detection enabled).
  The GCA flag is set to 0 on the rising edge of the 9th SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

DID bit (Device ID Address Detection Flag)

The DID flag indicates whether the device ID address was detected.

[Setting condition]

- When the first frame received immediately after a start or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]), with the DIDE bit in ICSER set to 1 (device ID address detection enabled).
  The DID flag is set to 1 on the rising edge of the 9th SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID flag after reading DID = 1.
- When a stop condition is detected.
- When the first frame received immediately after a start or restart condition is detected does not match a value of the device ID (1111 100b), with the DIDE bit in ICSER set to 1 (device ID address detection enabled).
  The DID flag is set to 0 on the rising edge of the 9th SCL clock cycle in the frame.
- When the first frame received immediately after a start or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]), and the second frame does not match any slave address from 0 to 2, with the DIDE bit in ICSER set to 1 (device ID address detection enabled).
  The DID flag is set to 0 on the rising edge of the 9th SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

HOA bit (Host Address Detection Flag)

The HOA flag indicates whether the host address was detected.

[Setting condition]

- When the received slave address matches the host address (0001 000b), with the HOAE bit in ICSER set to 1 (host address detection enabled).
  The HOA flag is set to 1 on the rising edge of the 9th SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the HOA flag after reading HOA = 1.
- When a stop condition is detected.
- When the received slave address does not match the host address (0001 000b), with the HOAE bit in ICSER set to 1 (host address detection enabled).
  The HOA flag is set to 0 on the rising edge of the 9th SCL clock cycle in the frame.

- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

32.2.10 ICSR2 : I²C Bus Status Register 2

Base address: IICn = 0x8004_3000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x09

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMOF	Timeout Detection Flag 0: Timeout not detected 1: Timeout detected.	R/W*1
1	AL	Arbitration-Lost Flag 0: Arbitration not lost 1: Arbitration lost	R/W*1
2	START	Start Condition Detection Flag 0: Start condition not detected 1: Start condition detected	R/W*1
3	STOP	Stop Condition Detection Flag 0: Stop condition not detected 1: Stop condition detected	R/W*1
4	NACKF	NACK Detection Flag 0: NACK not detected 1: NACK detected	R/W*1
5	RDRF	Receive Data Full Flag 0: ICDRR contains no receive data 1: ICDRR contains receive data	R/W*1
6	TEND	Transmit End Flag 0: Data being transmitted 1: Data transmit complete	R/W*1
7	TDRE	Transmit Data Empty Flag 0: ICDRT contains transmit data 1: ICDRT contains no transmit data	R

Note 1. Only 0 can be written, to clear the flag.

TMOF bit (Timeout Detection Flag)

The TMOF flag is set to 1 when IIC detects a timeout because the SCLn line state remains unchanged for the set period.

[Setting condition]

- When the SCLn line state remains unchanged for the period specified in the ICMR2.TMOH, TMOL, and TMOS bits while the ICFER.TMOE bit is 1 (timeout function enabled) in master or in slave mode and the received slave address matches.

[Clearing conditions]

- When 0 is written to the TMOF flag after reading TMOF = 1.
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

AL bit (Arbitration-Lost Flag)

The AL flag indicates that the bus mastership is lost in arbitration because of a bus conflict or because of some other reason, when a start condition was issued, or an address and data was transmitted.

The IIC monitors the level on the SDAn line during transmission. If the level on the line does not match the value of the bit being output, the IIC sets the AL flag to 1 to indicate that the bus is occupied by another device. The IIC can also set the flag to indicate the detection of arbitration loss during NACK transmission in master mode or during data transmission in slave mode.

[Setting conditions]

When master arbitration-lost detection is enabled (ICFER.MALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of the SCL clock except for the ACK period during data transmission in master transmit mode.
- When a start condition is detected while the ST bit in ICCR2 is 1 (start condition requested) or the internal SDA output state does not match the SDAn line level.
- When the ST bit in ICCR2 is 1 (start condition requested), with the BBSY flag in ICCR2 set to 1.

When NACK arbitration-lost detection is enabled (ICFER.NALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of the SCL clock in the ACK period during NACK transmission in receive mode.

When slave arbitration-lost detection is enabled (ICFER.SALE = 1):

- When the internal SDA output state does not match the SDAn line level on the rising edge of the SCL clock, except for the ACK period during data transmission in slave transmit mode.

[Clearing conditions]

- When 0 is written to the AL flag after reading AL = 1.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Table 32.5 Relationship between arbitration-lost generation sources and arbitration-lost enable functions

ICFER			ICSR2		
MALE	NALE	SALE	AL	Error	Arbitration-lost generation source
1	×	×	1	Start condition issuance error	When internal SDA output state does not match SDAn line level when a start condition is detected, while the ST bit in ICCR2 is 1.
					When ST in ICCR2 is set to 1 while BBSY in ICCR2 is 1.
			1	Transmit data mismatch	When transmit data (including slave address) does not match the bus state in master transmit mode.
×	1	×	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master or slave receive mode.
×	×	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode.

Note: ×: Don't care

START bit (Start Condition Detection Flag)

The START flag indicates whether a start or restart condition is detected.

[Setting condition]

- When a start or restart condition is detected.

[Clearing conditions]

- When 0 is written to the START flag after reading START = 1.
- When a stop condition is detected.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

STOP bit (Stop Condition Detection Flag)

The STOP flag indicates whether a stop condition is detected.

[Setting condition]

- When a stop condition is detected.

[Clearing conditions]

- When 0 is written to the STOP flag after reading STOP = 1.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

NACKF bit (NACK Detection Flag)

The NACKF flag indicates whether a NACK was detected.

[Setting condition]

- When acknowledge is not received (NACK received) from the receive device in transmit mode, with the NACKE bit in ICFER set to 1 (transfer suspension enabled).

[Clearing conditions]

- When 0 is written to the NACKF flag after reading NACKF = 1.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: When the NACKF flag is set to 1, the IIC suspends data transmission and reception. Writing to ICDRT in transmit mode or reading from ICDRR in receive mode with the NACKF flag set to 1 does not enable data transmit or receive operation. To restart data transmission or reception, set the NACKF flag to 0.

RDRF bit (Receive Data Full Flag)

The RDRF flag indicates whether the IDCRR contains receive data.

[Setting conditions]

- When receive data is transferred from ICDRS to ICDRR.
  The RDRF flag is set to 1 on the rising edge of the 8th or 9th SCL clock cycle (selected in the RDRFS bit in ICMR3).
- When the received slave address matches after a start (or restart) condition is detected with the TRS bit in ICCR2 set to 0.

[Clearing conditions]

- When 0 is written to the RDRF flag after reading RDRF = 1.
- When data is read from ICDRR.
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

TEND bit (Transmit End Flag)

The TEND flag indicates whether data transmission is still being transmitted or is complete.

[Setting condition]

- On the rising edge of the 9th SCL clock cycle while the TDRE flag is 1.

[Clearing conditions]

- When 0 is written to the TEND flag after reading TEND = 1.
- When data is written to ICDRT.
- When a stop condition is detected.
- When 1 is written to the IICRST bit in ICCR1 to initiate an IIC reset or an internal reset.

TDRE bit (Transmit Data Empty Flag)

The TDRE flag indicates whether the ICDRT contains transmit data.

[Setting conditions]

- When data is transferred from ICDRT to ICDRS and ICDRT becomes empty.
- When the TRS bit in ICCR2 is set to 1.

- When the received slave address matches while the TRS bit is 1.

[Clearing conditions]

- When data is written to ICDRT.
- When the TRS bit in ICCR2 is set to 0.
- When 1 is written to the IICRST bit in ICCR1 to apply an IIC reset or an internal reset.

Note: When the NACKF flag is set to 1 while the NACKE bit in ICFER is 1, the IIC suspends data transmission and reception. In this case, if the TDRE flag is 0 (next transmit data written), data is transferred to the ICDRS register and the ICDRT register becomes empty on the rising edge of the 9th clock cycle, but the TDRE flag is not set to 1.

32.2.11 SARLy : Slave Address Register L y (y = 0 to 2)

Base address: IICn = 0x8004_3000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x0A + 0x02 × y

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SVA[6:0]							SVA0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SVA0	10-bit Address LSB Slave address setting	R/W
7:1	SVA[6:0]	7-bit Address/10-bit Address Lower Bits Slave address setting	R/W

SVA0 bit (10-bit Address LSB)

When the 10-bit address format is selected (SARUy.FS = 1), the SVA0 bit functions as the LSB of a 10-bit address and is combined with the SVA[6:0] bits to form the lower 8 bits of a 10-bit address.

This bit is valid when the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1. When the SARUy.FS or SARyE bit is 0, the setting in this bit is ignored.

SVA[6:0] bits (7-bit Address/10-bit Address Lower Bits)

When the 7-bit address format is selected (SARUy.FS = 0), the SVA[6:0] bits function as a 7-bit address. When the 10-bit address format is selected (SARUy.FS = 1), these bits combined with the SVA0 bit to form the lower 8 bits of a 10-bit address. When the SARyE bit in ICSER is 0, the setting in these bits is ignored.

32.2.12 SARUy : Slave Address Register U y (y = 0 to 2)

Base address: IICn = 0x8004_3000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x0B + 0x02 × y

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SVA[1:0]		FS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FS	7-bit/10-bit Address Format Select 0: Select 7-bit address format 1: Select 10-bit address format	R/W

Bit	Symbol	Function	R/W
2:1	SVA[1:0]	10-bit Address Upper Bits Slave address setting	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

FS bit (7-bit/10-bit Address Format Select)

The FS bit selects a 7-bit or 10-bit address format for slave address y (in SARLy and SARUy).

When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 0, the 7-bit address format is selected for slave address y, the SVA[6:0] setting in SARLy is valid, and the SVA[1:0] and SVA0 settings in SARLy are ignored.

When the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, the 10-bit address format is selected for slave address y and the SVA[1:0] and SARLy settings are valid.

When the SARyE bit in ICSER is 0 (SARLy and SARUy disabled), the SARUy.FS setting is invalid.

SVA[1:0] bits (10-bit Address Upper Bits)

When the 10-bit address format is selected (FS = 1), the SVA[1:0] bits function as the upper 2 bits of a 10-bit address.

These bits are valid when the SARyE bit in ICSER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1. When the SARUy.FS or SARyE bit is 0, the setting in these bits is ignored.

32.2.13 ICBRL : I2C Bus Bit Rate Low-Level Register

Base address: IICn = 0x8004_3000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	BRL[4:0]				
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
4:0	BRL[4:0]	Bit Rate Low-Level Period Low-level period setting of SCL clock	R/W
7:5	—	These bits are read as 1. The write value should be 1.	R/W

ICBRL is a 5-bit register that sets the low-level period of the SCL clock. ICBRL also works to generate the data setup time for the automatic SCL low-hold operation (see section 32.8. Automatic Low-Hold Function for SCL). When the IIC is used only in slave mode, this register must be set to a value longer than the data setup time.*1

ICBRL counts the low-level period with the internal reference clock source (IICΦ) specified in the CKS[2:0] bits in ICMR1.

If the digital noise filter is enabled (the NFE bit in ICFER is 1), set the ICBRL register to a value at least one greater than the number of stages in the noise filter. For this number, see the description of the ICMR3.NF[1:0] bits.

Note 1. Data setup time (tSU: DAT)
250 ns for up to 100 kbps: Standard-mode (Sm)
100 ns for up to 400 kbps: Fast-mode (Fm)

32.2.14 ICBRH : I²C Bus Bit Rate High-Level Register

Base address: IICn = 0x8004_3000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x11

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	BRH[4:0]				
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
4:0	BRH[4:0]	Bit Rate High-Level Period High-level period setting of SCL clock	R/W
7:5	—	These bits are read as 1. The write value should be 1.	R/W

ICBRH is a 5-bit register that sets the high-level period of the SCL clock. ICBRH is valid in master mode. If the IIC is used only in slave mode, no setting is required in this register.

ICBRH counts the high-level period with the internal reference clock source (IICΦ) specified in the CKS[2:0] bits in ICMR1.

If the digital noise filter is enabled (the NFE bit in ICFER is 1), set the ICBRH register to a value at least one greater than the number of stages in the noise filter. For this number, see the description of the ICMR3.NF[1:0] bits.

The IIC transfer rate and the SCL clock duty are calculated using the following expressions (1) to (5):

(1) ICFER.SCLE = 0
Transfer rate = 1/{[(BRH + 1) + (BRL + 1)]/IICΦ*1 + tr*2 + tf*2}
Duty cycle = {tr + [(BRH + 1)/IICΦ]}/{tr + tf + [(BRH + 1) + (BRL + 1)]/IICΦ}

(2) ICFER.SCLE = 1 and ICFER.NFE = 0 and CKS[2:0] = 000b (IICΦ = PCLKL)
Transfer rate = 1/{[(BRH + 3) + (BRL+ 3)]/IICΦ + tr + tf}
Duty cycle = {tr + [(BRH + 3)/IICΦ]}/{tr + tf + [(BRH + 3) + (BRL + 3)]/IICΦ}

(3) ICFER.SCLE = 1 and ICFER.NFE = 1 and CKS[2:0] = 000b (IICΦ = PCLKL)
Transfer rate = 1/{[(BRH + 3 + nf*3) + (BRL + 3 + nf)]/IICΦ + tr + tf}
Duty cycle = {tr + [(BRH + 3 + nf)/IICΦ]}/{tr + tf + [(BRH + 3 + nf) + (BRL + 3 + nf)]/IICΦ}

(4) ICFER.SCLE = 1 and ICFER.NFE = 0 and CKS[2:0] ≠ 000b
Transfer rate = 1/{[(BRH + 2) + (BRL + 2)]/IICΦ + tr + tf}
Duty cycle = {tr + [(BRH + 2)/IICΦ]}/{tr + tf + [(BRH + 2) + (BRL + 2)]/IICΦ}

(5) ICFER.SCLE = 1 and ICFER.NFE = 1 and CKS[2:0] ≠ 000b
Transfer rate = 1/{[(BRH + 2 + nf) + (BRL + 2 + nf)]/IICΦ + tr + tf}
Duty cycle = {tr + [(BRH + 2 + nf)/IICΦ]}/{tr + tf + [(BRH + 2 + nf) + (BRL + 2 + nf)]/IICΦ}

Note 1. IICΦ = PCLKL × Division ratio.
Note 2. The SCLn line rise time (tr) and SCLn line fall time (tf) depend on the total bus line capacitance (Cb) and the pull-up resistor (Rp). For details, see the I²C bus standard from NXP Semiconductors.
Note 3. nf = Number of digital noise filter stages selected in the ICMR3.NF bit.

Table 32.6 Example of ICBRH/ICBRL settings for transfer rate when SCLE = 0

Transfer rate (kbps)	CKS[2:0]	BRH[4:0]	BRL[4:0]	PCLKL (MHz)	NF[1:0]	Computation expression
100	100b	12 (0x0C)	12 (0x0C)	50	—	(1)
400	001b	23 (0x17)	23 (0x17)	50	—	(1)

Table 32.7 Example of ICBRH/ICBRL settings for transfer rate when SCLE = 1 and NFE = 0

Transfer rate (kbps)	CKS[2:0]	BRH[4:0]	BRL[4:0]	PCLKL (MHz)	NF[1:0]	Computation expression
100	011b	25 (0x19)	25 (0x19)	50	—	(4)
400	001b	22 (0x16)	22 (0x16)	50	—	(4)

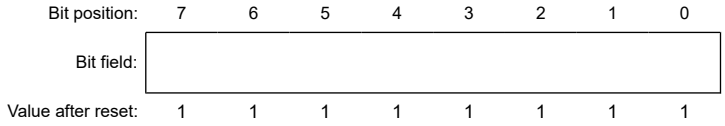
Table 32.8 Example of ICBRH/ICBRL settings for transfer rate when SCLE = 1 and NFE = 1

Transfer rate (kbps)	CKS[2:0]	BRH[4:0]	BRL[4:0]	PCLKL (MHz)	NF[1:0]	Computation expression
100	011b	24 (0x18)	24 (0x18)	50	01b	(5)
400	001b	21 (0x15)	21 (0x15)	50	01b	(5)

32.2.15 ICDRT : I2C Bus Transmit Data Register

Base address: IICn = 0x8004_3000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x12



Bit	Symbol	Function	R/W
7:0	n/a	Transmit Data	R/W

When ICDRT detects a space in the I2C Bus Shift Register (ICDRS), it transfers the transmit data that is written to ICDRT to ICDRS and starts transmitting data in transmit mode.

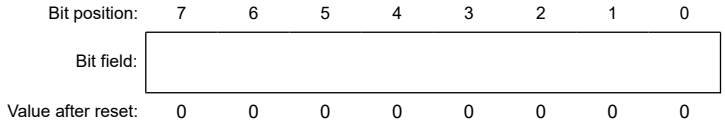
The double-buffer structure of ICDRT and ICDRS allows continuous transmit operation if the next transmit data is written to ICDRT while the ICDRS data is being transmitted.

ICDRT can always be read from and written to. Write transmit data to ICDRT when a transmit data empty interrupt (IICn_TXI) request is generated.

32.2.16 ICDRR : I2C Bus Receive Data Register

Base address: IICn = 0x8004_3000 + 0x0400 × n (n = 0, 1)
IIC2 = 0x8100_8000

Offset address: 0x13



Bit	Symbol	Function	R/W
7:0	n/a	Receive Data	R

When 1 byte of data is received, the received data is transferred from the I2C Bus Shift Register (ICDRS) to ICDRR to enable the next data to be received.

The double-buffer structure of ICDRS and ICDRR allows continuous receive operation if the received data is read from ICDRR while ICDRS is receiving data. ICDRR cannot be written to. Read data from ICDRR when a receive data full interrupt (IICn_RXI) request is generated.

If ICDRR receives the next receive data before the current data is read from ICDRR while the RDRF flag in ICSR2 is 1, the IIC automatically holds the SCL clock low for 1 cycle before the RDRF flag is set to 1 again.

32.2.17 ICDRS : I²C Bus Shift Register

Base address: n/a

Offset address: n/a

Bit position:	7	6	5	4	3	2	1	0
Bit field:								
Value after reset:	—	—	—	—	—	—	—	—

ICDRS is an 8-bit shift register to transmit and receive data.

During transmission, transmit data is transferred from ICDRT to ICDRS and is sent from the SDAn pin. During reception, data is transferred from ICDRS to ICDRR after 1 byte of data is received. ICDRS cannot be accessed directly.

32.3 Operation

32.3.1 Communication Data Format

The I²C bus format consists of 8-bit data and 1-bit acknowledge. The frame following a start or restart condition is an address frame that specifies a slave device with which the master device communicates. The specified slave is valid until a new slave is specified or a stop condition is issued.

Figure 32.3 shows the I²C bus format, and Figure 32.4 shows the I²C bus timing.

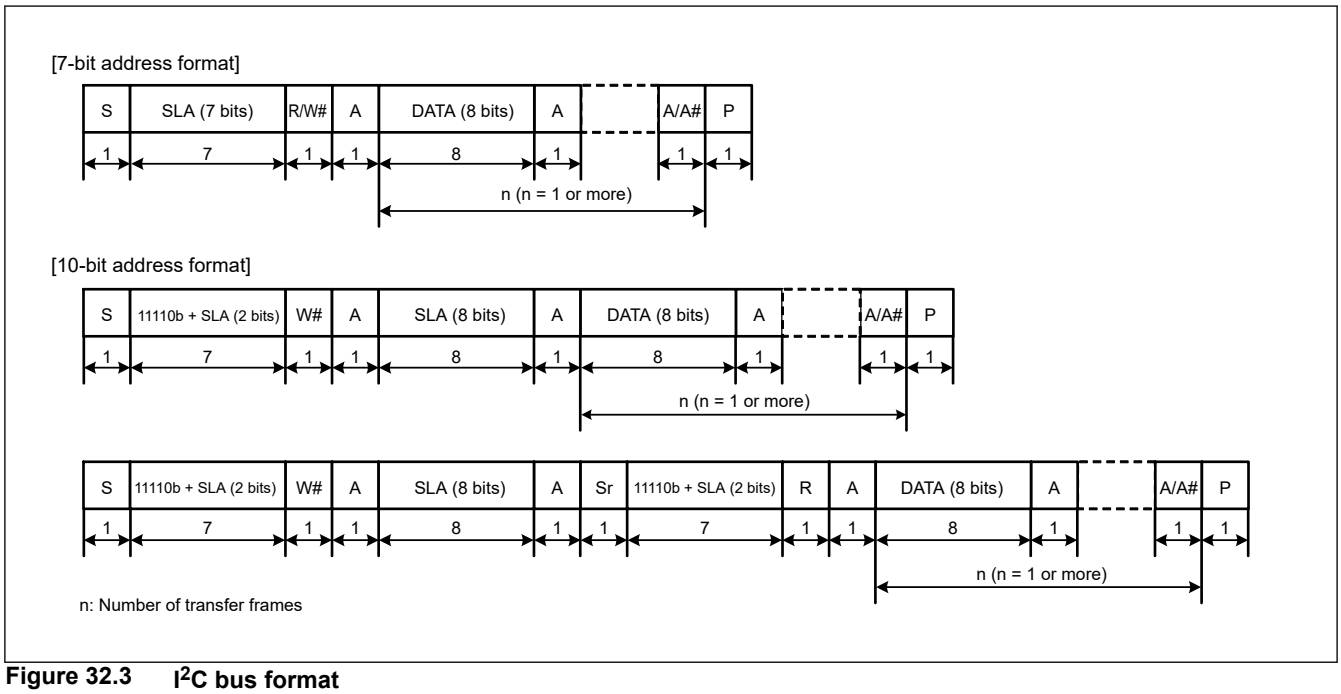


Figure 32.3 I²C bus format

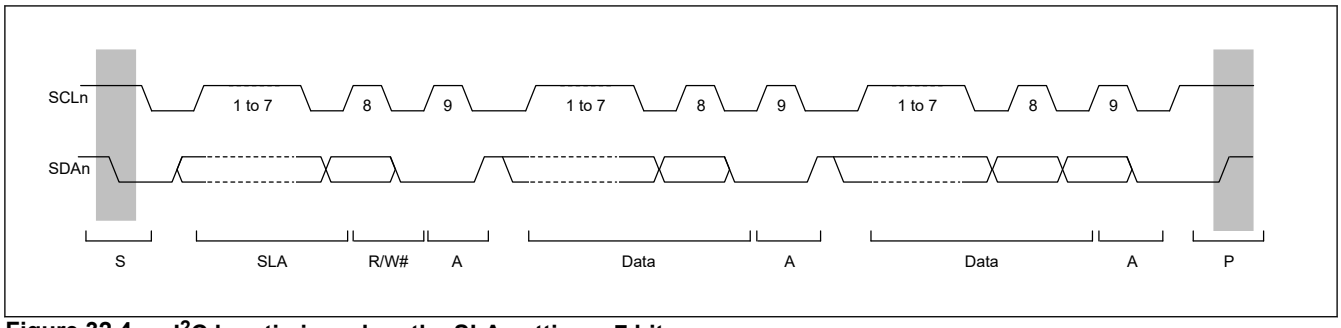


Figure 32.4 I²C bus timing when the SLA setting = 7 bits

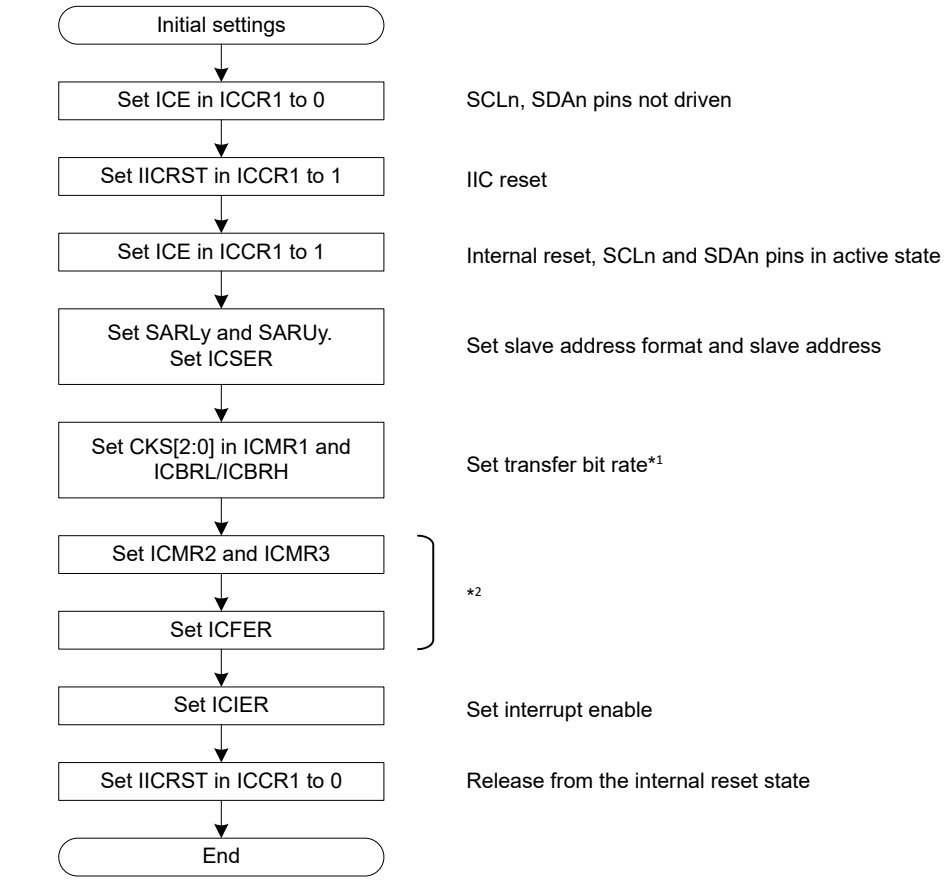
S:	Start condition. The master device drives the SDAn line low from high while the SCLn line is high.
SLA:	Slave address, by which the master device selects a slave device.
R/W#:	Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.
A:	Acknowledge. The receive device drives the SDAn line low. In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.
A#:	Not Acknowledge. The receive device drives the SDAn line high.
Sr:	Restart condition. The master device drives the SDAn line low from the high level after the setup time has elapsed with the SCLn line high.
DATA:	Transmitted or received data
P:	Stop condition. The master device drives the SDAn line high from low while the SCLn line is high.

32.3.2 Initial Settings

Before starting data transmission and reception, initialize the IIC using the procedure shown in Figure 32.5.

1. Set the ICCR1.ICE bit set to 0 to set the SCLn and SDAn pins to the inactive state.
2. Set the ICCR1.IICRST bit to 1 to initiate IIC reset.
3. Set the ICCR1.ICE bit to 1 to initiate internal reset, which initializes the flags and the internal state of the ICSR1 register.
4. Set the SARLy, SARUy, ICSER, ICMR1, ICBRH, and ICBRL registers (y = 0 to 2), and set the other registers as required. For the initial IIC settings, see Figure 32.5.
5. When the required register settings are complete, set the ICCR1.IICRST bit to 0, to release the IIC reset.

Note: This procedure is not necessary if the IIC initialization is already complete.



y = 0 to 2

Note 1. When the IIC is used only in slave mode, set the ICBRL register to a value longer than the data setup time.
Note 2. Set these registers as required.

Figure 32.5 Example IIC initialization flow

32.3.3 Master Transmit Operation

In master transmit operation, the IIC outputs the SCL clock and transmit data signals as the master device, and the slave device returns the acknowledgments. Figure 32.6 shows an example of master transmission, and Figure 32.7 to Figure 32.9 show the operation timing in master transmission.

To set up and perform master transmission:

1. To initialize the IIC, follow the procedure in section 32.3.2. Initial Settings.
2. Read the BBSY flag in ICCR2 to check that the bus is free, and then set the ST bit in ICCR2 to 1 (start condition request). On receiving the request, the IIC issues a start condition. At the same time, the BBSY and START flags in ICSR2 automatically set to 1 and the ST bit is automatically set to 0. If the start condition is detected and the internal levels for the SDA output state and the levels on the SDAn line match while the ST bit is 1, the IIC recognizes that the start condition requested by the ST bit is successfully complete, and the MST and TRS bits in ICCR2 automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 also automatically sets to 1 in response to the setting of the TRS bit to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. When the transmit data is written to ICDRT, the TDRE flag is automatically set to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag again sets to 1. After the byte containing the slave address and R/W# bit is transmitted, the value of the TRS bit automatically updates to select master transmit or master receive mode based on the value of the transmitted R/W# bit. If the value of the R/W# bit is 0, the IIC continues in master transmit mode.

If the ICSR2.NACKF flag is 1, indicating that no slave device recognized the address or that there was an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.
To transmit data with an address in the 10-bit format, start by writing 11110b, the 2 upper bits of the slave address, and W to ICDRT as the first address transmission. For the second address transmission, write the 8 lower bits of the slave address to ICDRT.

4. After confirming that the TDRE flag in ICSR2 is 1, write the transmit data to the ICDRT register. The IIC automatically holds the SCLn line low until the transmit data is ready or a stop condition is issued.
5. After all bytes of transmit data are written to the ICDRT register, wait until the value in the TEND flag in ICSR2 returns to 1, and then set the SP bit in ICCR2 to 1 (stop condition requested). On receiving a stop condition request, the IIC issues the stop condition.
6. On detecting the stop condition, the IIC automatically sets the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. In addition, the IIC automatically sets the TDRE and TEND flags to 0, and sets the STOP flag in ICSR2 to 1.
7. After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

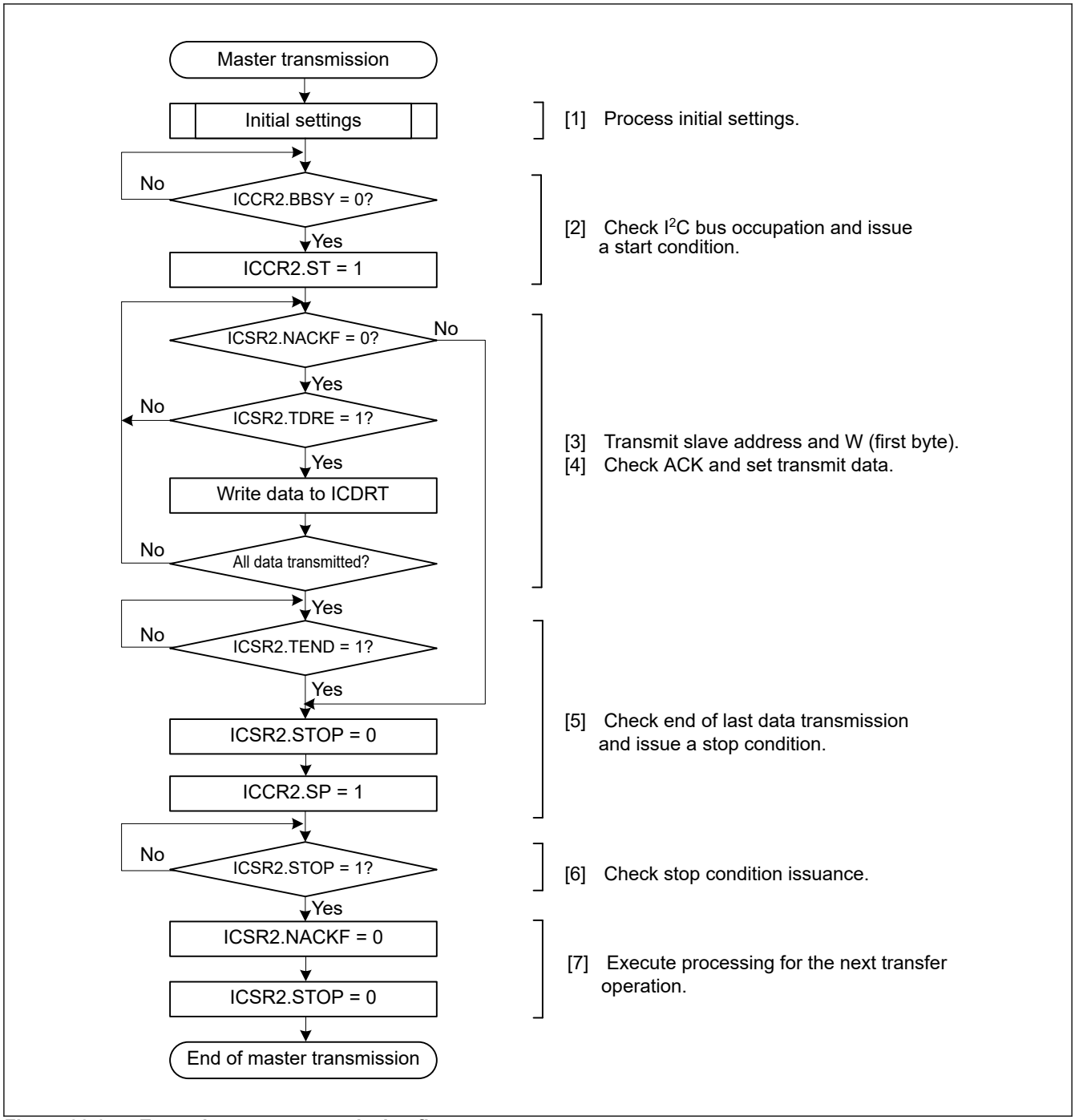


Figure 32.6 Example master transmission flow

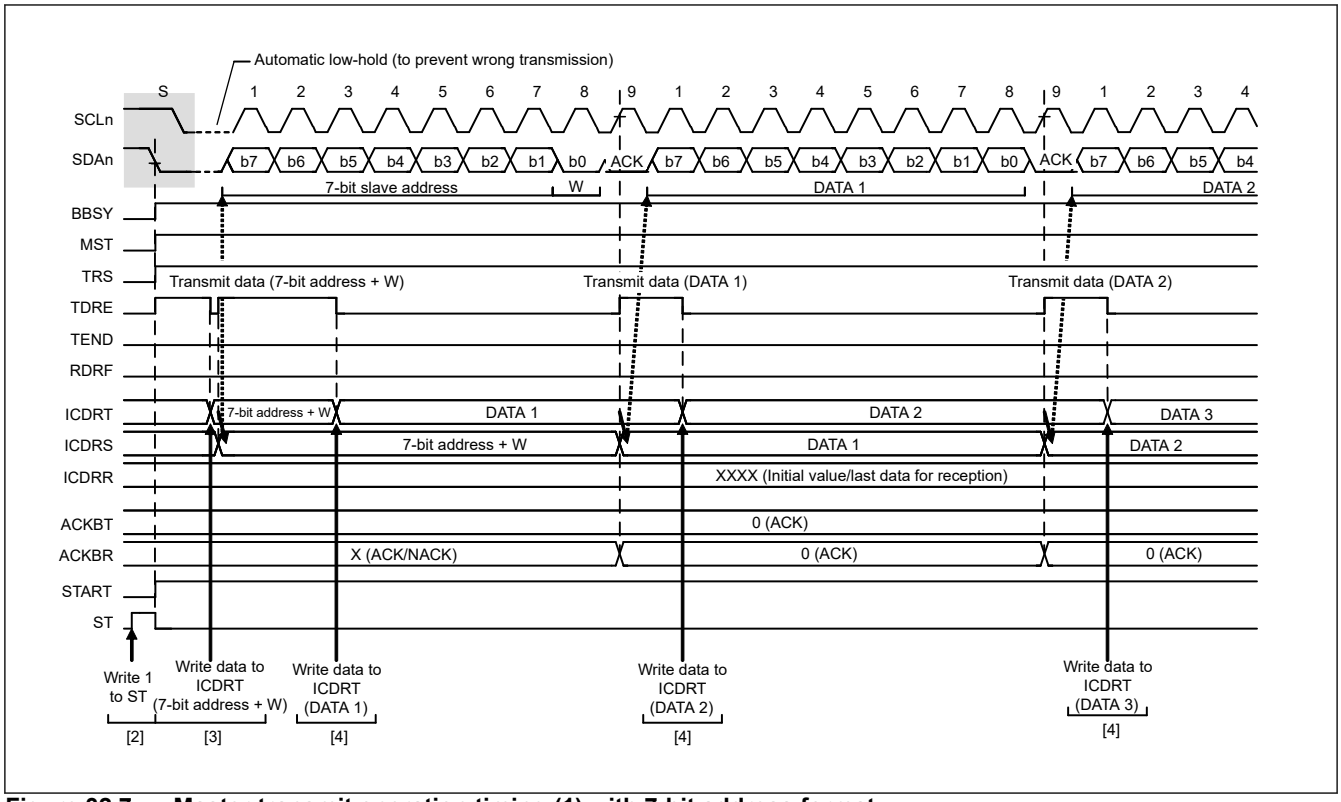


Figure 32.7 Master transmit operation timing (1) with 7-bit address format

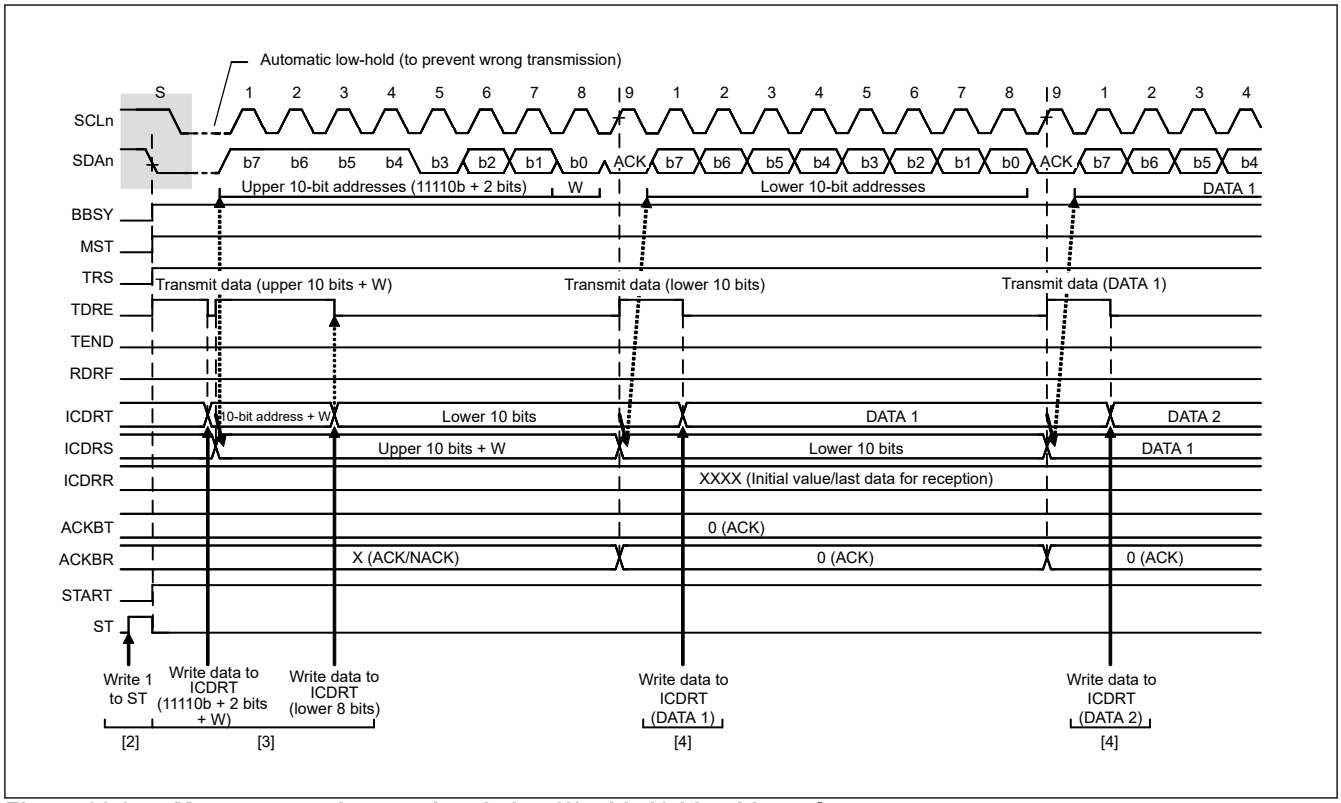


Figure 32.8 Master transmit operation timing (2) with 10-bit address format

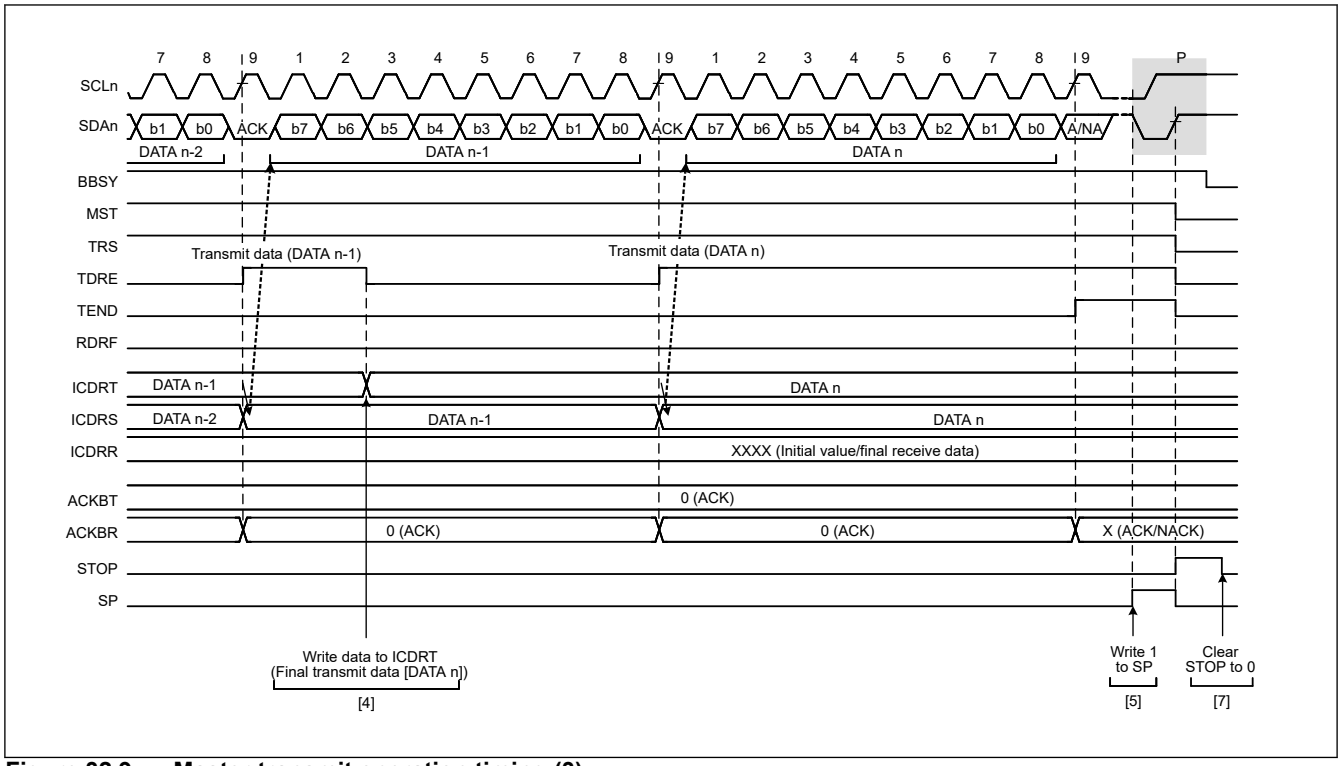


Figure 32.9 Master transmit operation timing (3)

32.3.4 Master Receive Operation

In master receive operation, the IIC as a master device outputs the SCL clock, receives data from the slave device, and returns acknowledgments. Because the IIC must start by sending a slave address to the associated slave device, the slave address part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.

Figure 32.10 and Figure 32.11 show examples of master reception (7-bit address format), and Figure 32.12 to Figure 32.14 show the operation timing in master reception.

To set up and perform master reception:

1. To initialize the IIC, follow the procedure in section 32.3.2. Initial Settings.
2. Read the BBSY flag in ICCR2 to check that the bus is free, and then set the ST bit in ICCR2 to 1 to request the issue of a start condition. On receiving the request, the IIC issues a start condition. When the IIC detects the start condition, the BBSY and START flags in ICSR2 automatically set to 1, and the ST bit automatically sets to 0. If the start condition is detected and the levels for the SDA output and the levels on the SDAn line match while the ST bit is 1, the IIC recognizes that issuance of the start condition as requested by the ST bit is successfully complete, and the MST and TRS bits in ICCR2 automatically set to 1, placing the IIC in master transmit mode. The TDRE flag in ICSR2 also is automatically set to 1 in response to the setting of the TRS bit to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to ICDRT. When the transmit data is written to ICDRT, the TDRE flag is automatically set to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag again is set to 1. When the byte containing the slave address and R/W# bit is transmitted, the value of the ICCR2.TRS bit automatically updates to select transmit or receive mode based on the value of the transmitted R/W# bit. If the value of the R/W# bit is 1, the TRS bit is set to 0 on the rising edge of the 9th cycle of the SCL clock, placing the IIC in master receive mode. At this time, the TDRE flag is set to 0 and the ICSR2.RDRF flag is automatically set to 1.
   If the ICSR2.NACKF flag is 1, indicating that no slave device recognized the address or that there is an error in communications, write 1 to the ICCR2.SP bit to issue a stop condition.
   For master reception from a device with a 10-bit address, start by using master transmission to issue the 10-bit address, and then issue a restart condition. After that, transmit 11110b, the two upper bits of the slave address, and the R bit to place the IIC in master receive mode.
4. Dummy read ICDRR after confirming that the RDRF flag in ICSR2 is 1. Doing so causes the IIC to start output of the SCL clock and start data reception.

5. After 1 byte of data is received, the RDRF flag in ICSR2 is set to 1 on the rising edge of the 8th or 9th cycle of the SCL clock, as selected in the RDRFS bit in ICMR3. Reading ICDRR produces the received data, and automatically sets the RDRF flag to 0. The value of the acknowledgment field received during the 9th cycle of the SCL clock is returned as the value set in the ICMR3.ACKBT bit. If the next byte to be received is the second-to-last byte, set the ICMR3.WAIT bit to 1 for wait insertion before reading ICDRR, containing the second-to-last byte. In addition to enabling NACK output, even when interrupts or other operations result in delays in setting the ICMR3.ACKBT bit to 1 (NACK) in step (6), this fixes the SCLn line to the low level on the rising edge of the 9th clock cycle in reception of the last byte, which enables the issuing of a stop condition.
6. When the ICMR3.RDRFS bit is 0, and the slave device must be notified that it is to end transfer for data reception after transfer of the next and final byte, set the ICMR3.ACKBT bit to 1 (NACK).
7. After reading the second-to-last byte from the ICDRR register, if the value of the ICSR2.RDRF flag is 1, write 1 to the SP bit in ICCR2 (stop condition requested), and then read the last byte from ICDRR. When ICDRR is read, the IIC is released from the wait state and issues the stop condition after low-level output in the 9th clock cycle is complete or the SCLn line is released from the low-hold state.
8. On detecting the stop condition, the IIC automatically sets the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Additionally, detection of the stop condition sets the ICSR2.STOP flag to 1.
9. Check that the ICSR2.STOP flag is 1, and then set the ICSR2.NACKF and ICSR2.STOP flags to 0 for the next transfer operation.

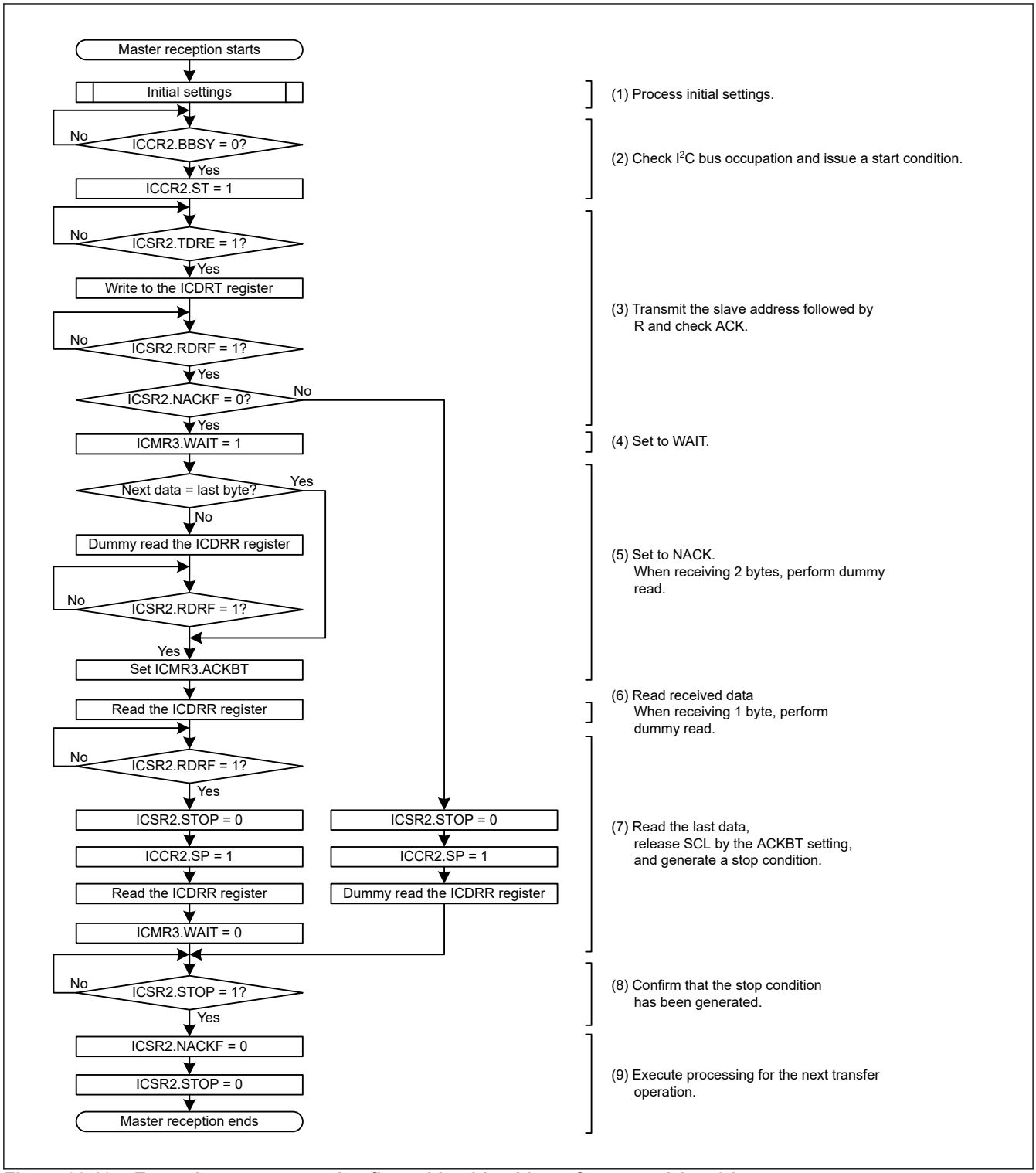


Figure 32.10 Example master reception flow with 7-bit address format and 1 or 2 bytes

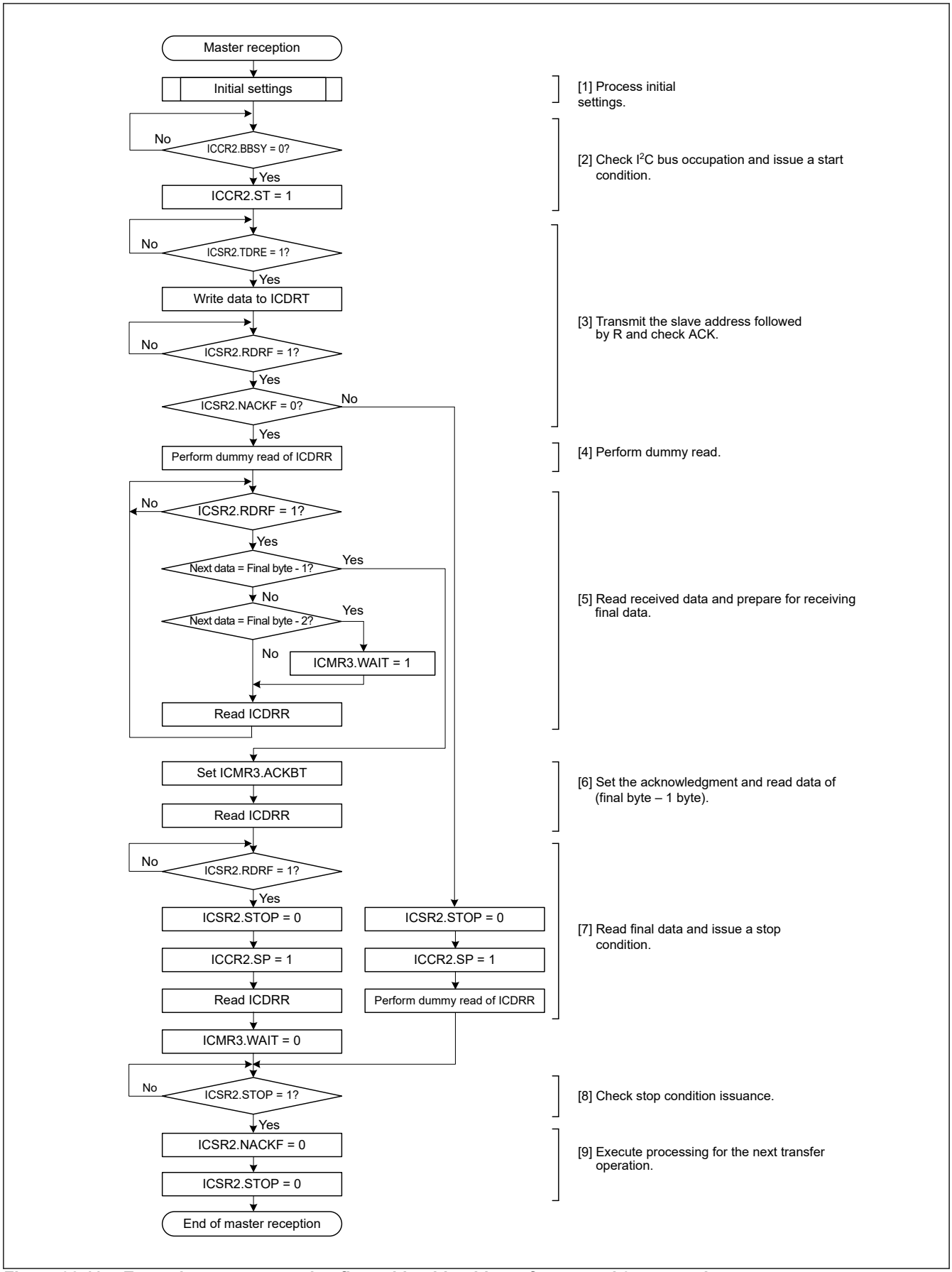


Figure 32.11 Example master reception flow with 7-bit address format and 3 or more bytes

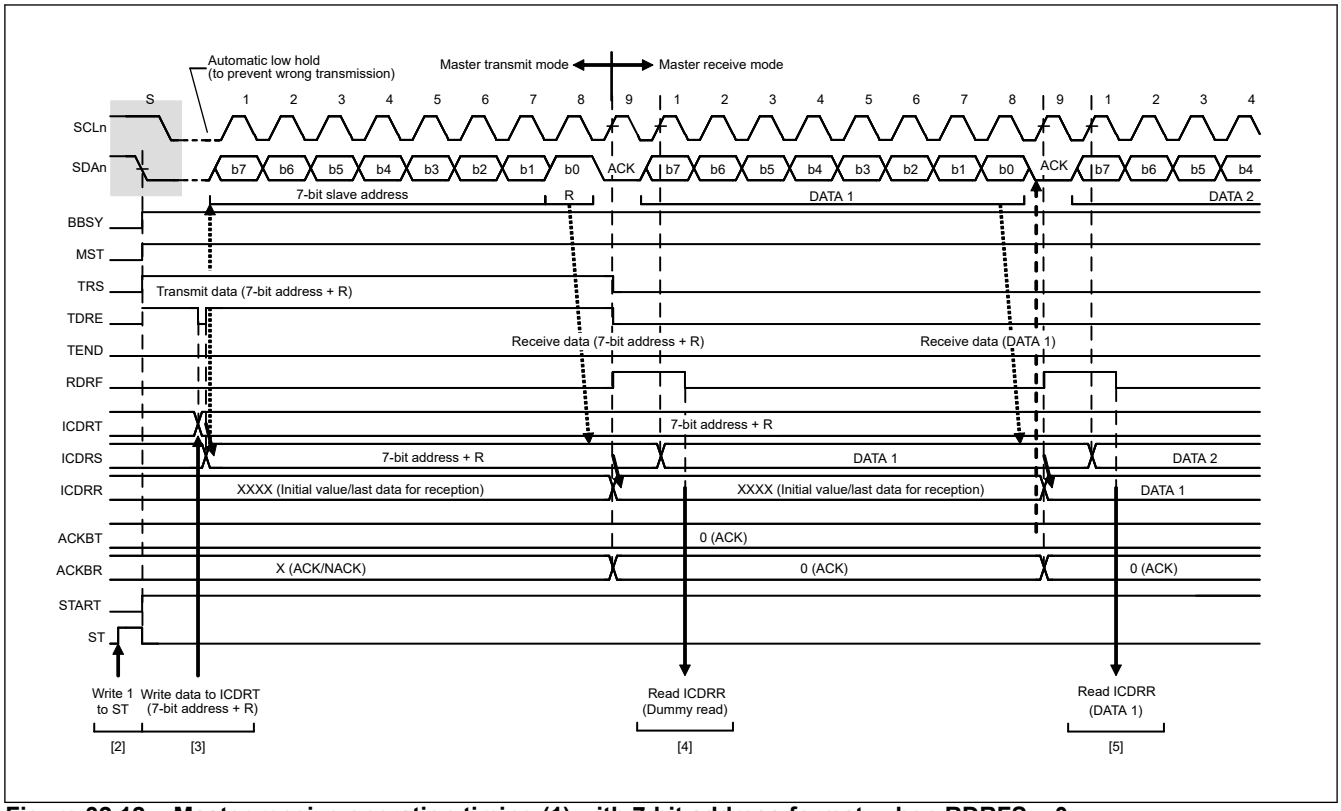


Figure 32.12 Master receive operation timing (1) with 7-bit address format, when RDRFS = 0

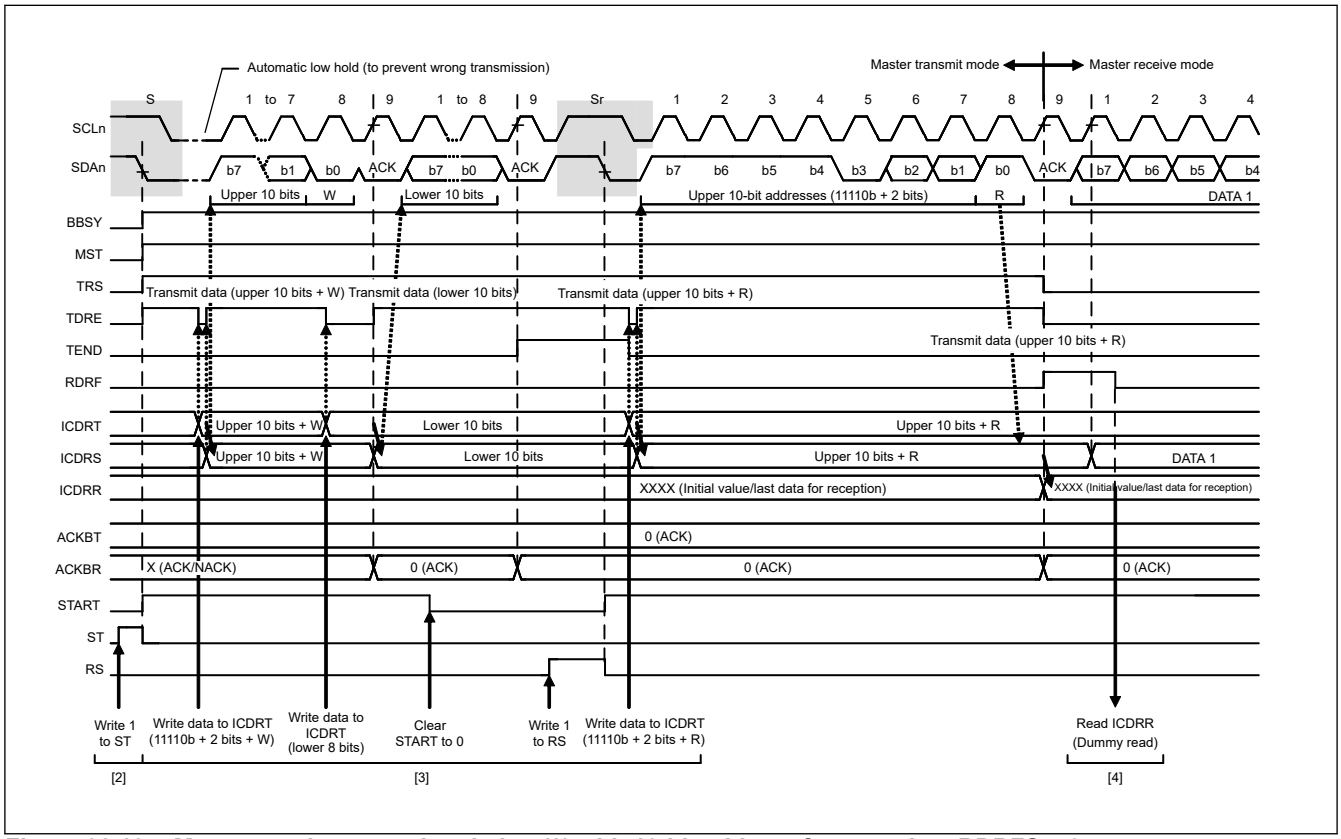


Figure 32.13 Master receive operation timing (2) with 10-bit address format, when RDRFS = 0

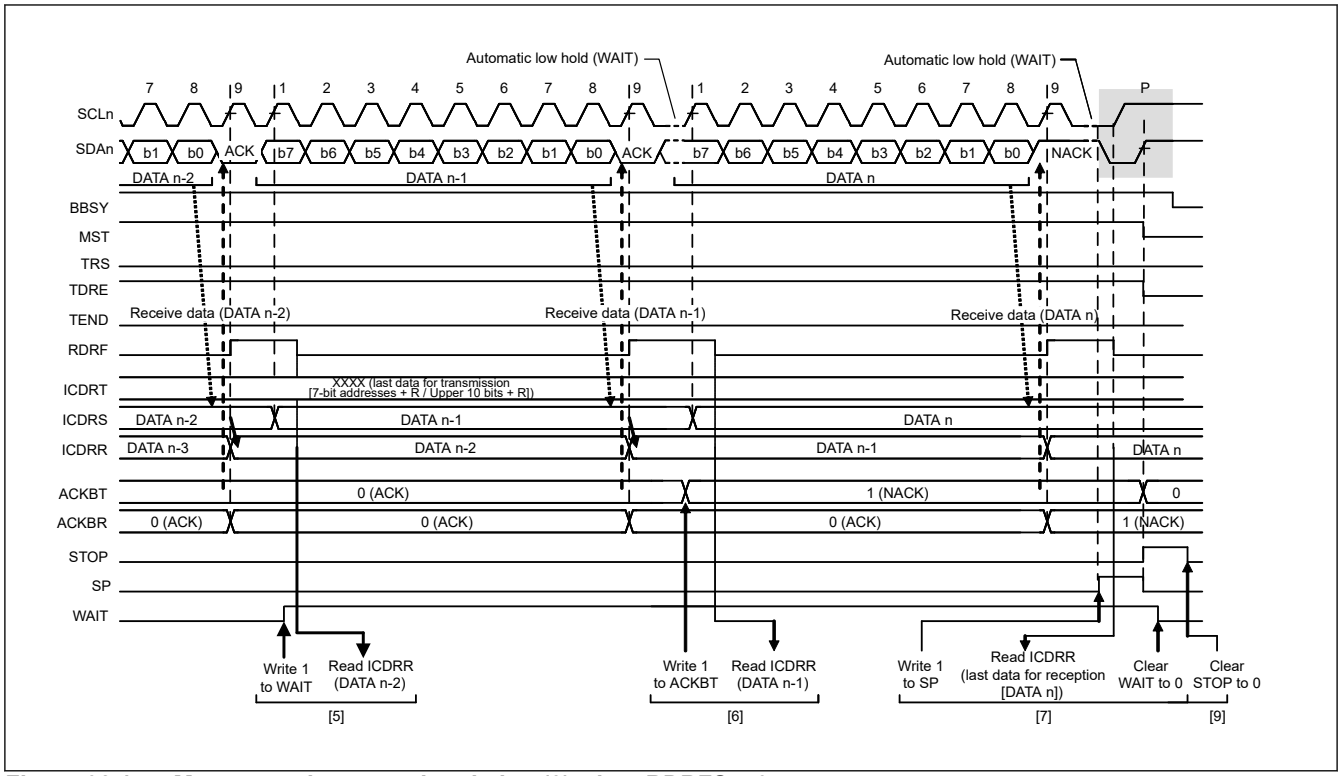


Figure 32.14 Master receive operation timing (3) when RDRFS = 0

32.3.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL clock, the IIC transmits data as a slave device, and the master device returns the acknowledgments.

Figure 32.15 shows an example of slave transmission, and Figure 32.16 and Figure 32.17 show the operation timing in slave transmission.

To set up and perform slave transmission:

1. To initialize the IIC, follow the procedure in section 32.3.2. Initial Settings.
   After the initialization, the IIC stays in the standby state until it receives a slave address that matches.
2. After receiving a matching slave address, the IIC sets one of the associated bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) flags to 1 on the rising edge of the 9th cycle of the SCL clock and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the 9th cycle of the SCL clock. If the value of the received R/W# bit is 1, the IIC automatically places itself in slave transmit mode by setting both the ICCR2.TRS bit and the ICSR2.TDRE flag to 1.
3. Check that the ICSR2.TEND flag is 1, and write the transmit data to the ICDRT register. If the IIC receives no acknowledge from the master device (receives an NACK signal) while the ICFER.NACKE bit is 1, the IIC suspends transfer of the next data.
4. Wait until the ICSR2.TEND flag is set to 1 while the ICSR2.TDRE flag is 1, after the ICSR2.NACKF flag is set to 1 or the last byte for transmission is written to the ICDRT register. When the ICSR2.NACKF flag or the TEND flag is 1, the IIC drives the SCLn line low on the 9th falling edge of the SCL clock.
5. When the ICSR2.NACKF flag or the ICSR2.TEND flag is 1, dummy read ICDRR to complete the processing. This releases the SCLn line.
6. On detecting the stop condition, the IIC automatically sets the ICSR1.HOA, GCA, and AASy (y = 0 to 2) flags, the ICSR2.TDRE and TEND flags, and the ICCR2.TRS bit to 0, and enters slave receive mode.
7. Check that the ICSR2.STOP flag is 1, and then set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

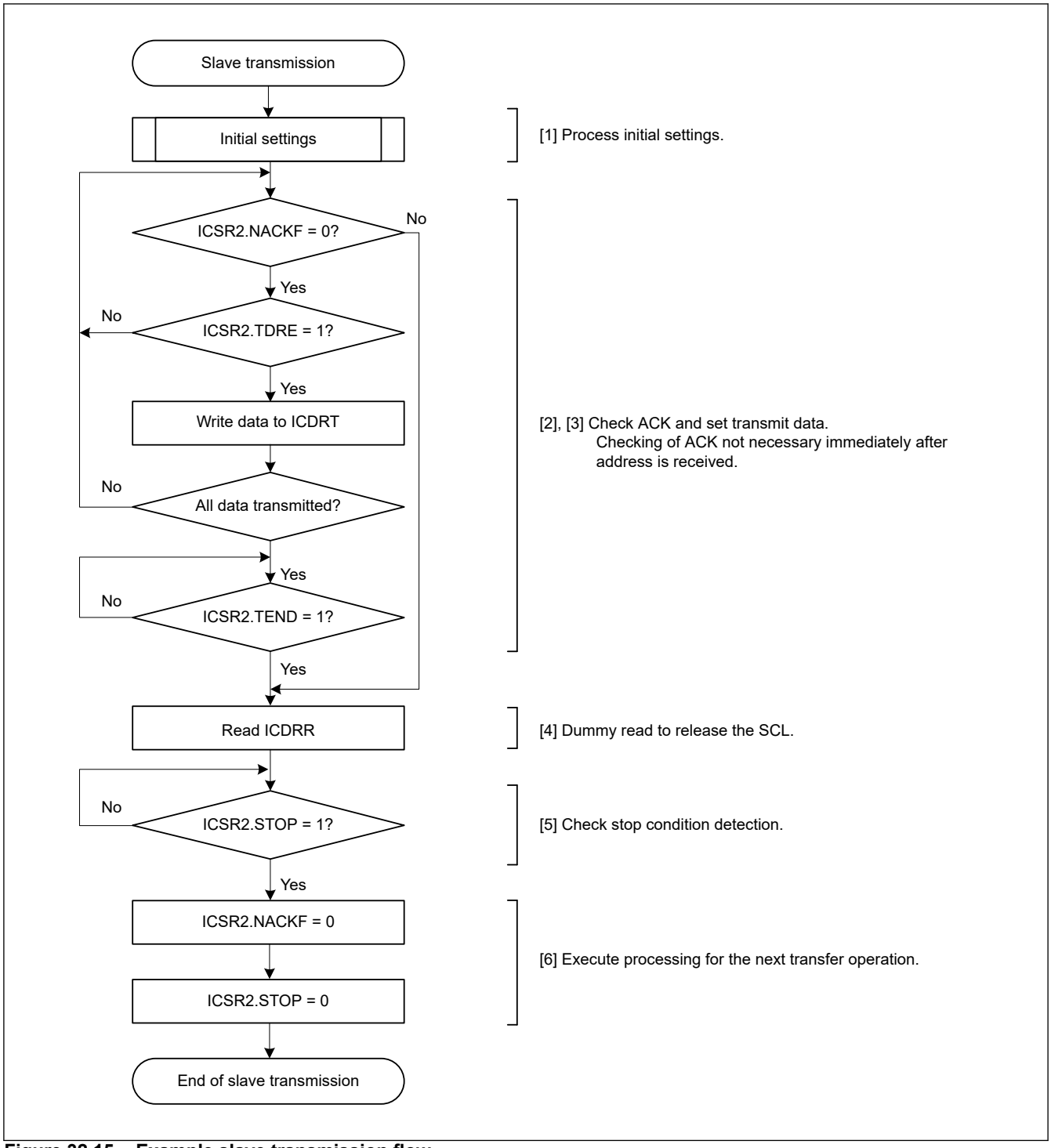


Figure 32.15 Example slave transmission flow

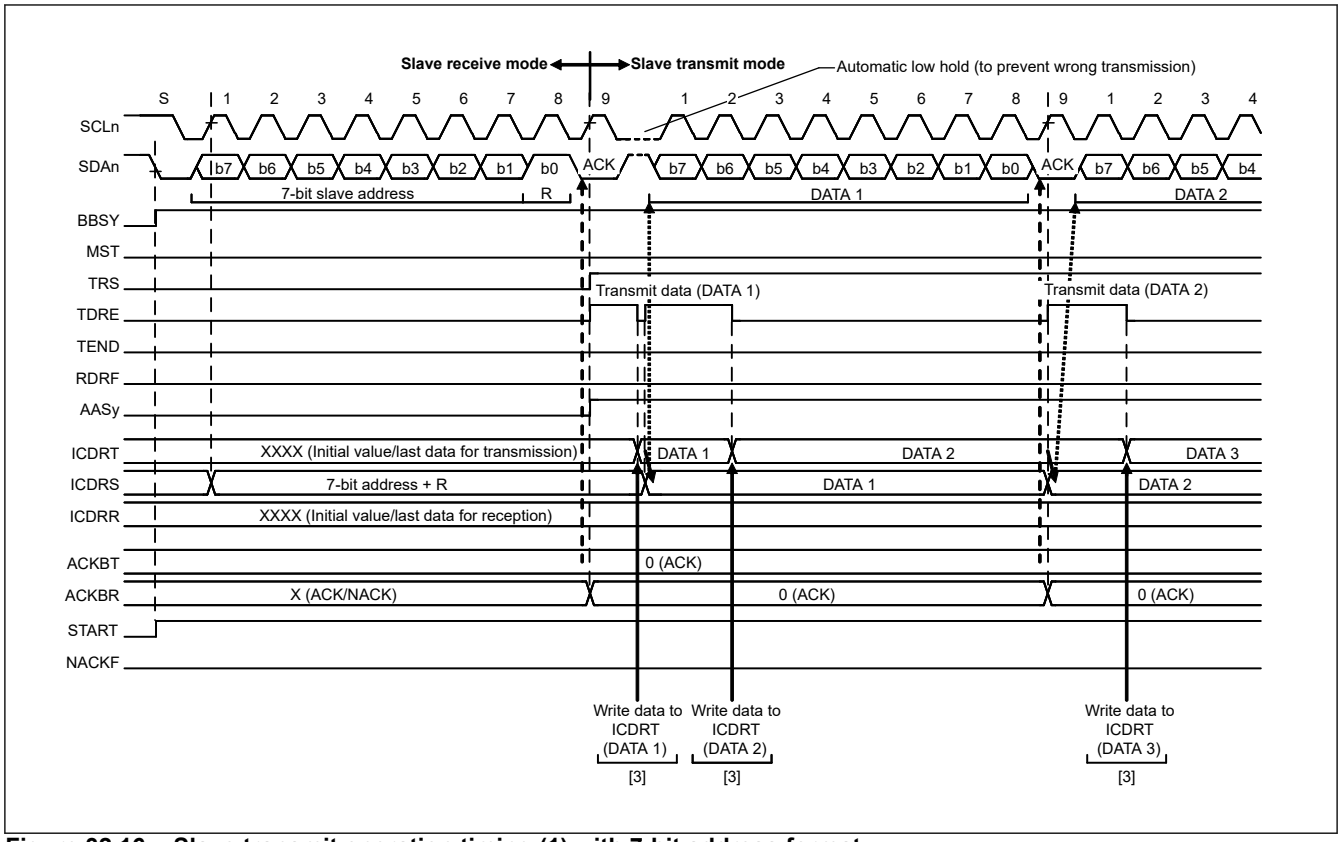


Figure 32.16 Slave transmit operation timing (1) with 7-bit address format

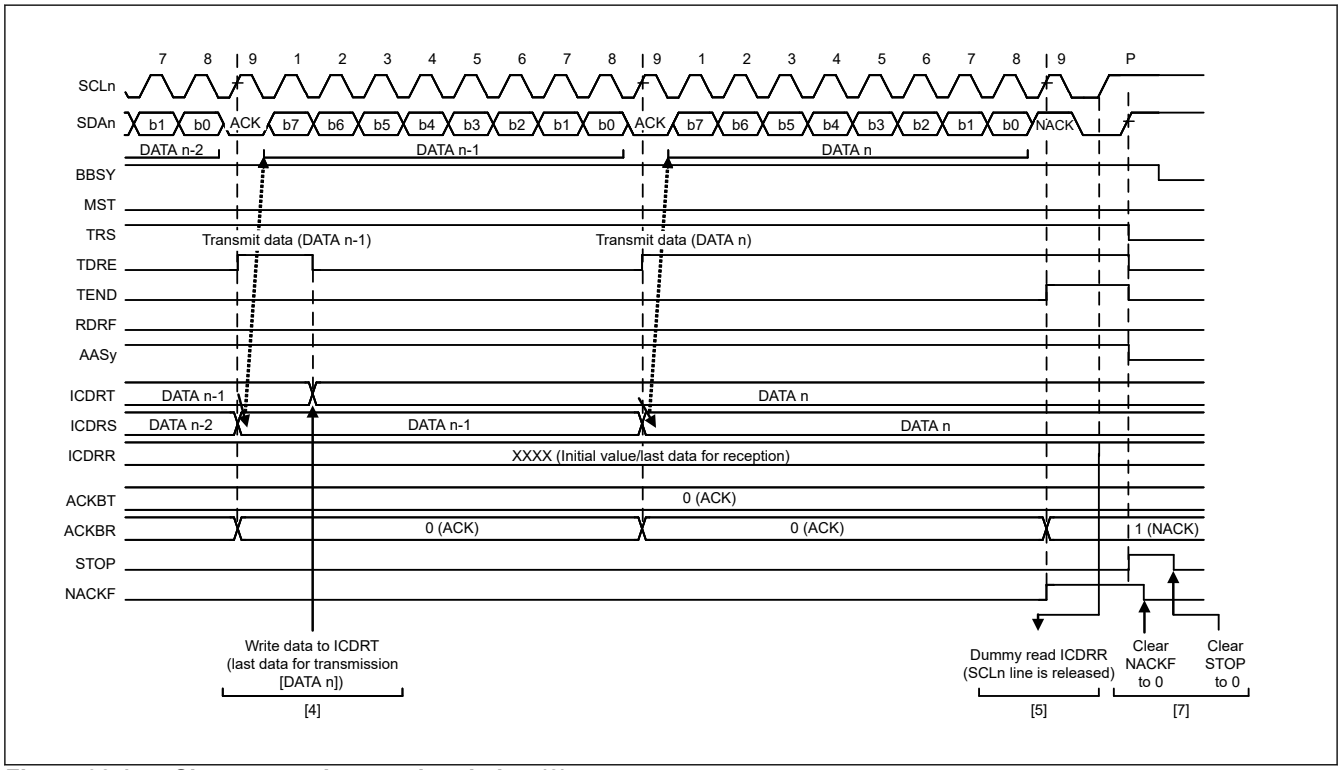


Figure 32.17 Slave transmit operation timing (2)

32.3.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the IIC returns acknowledgments as a slave device.

Figure 32.18 shows an example of slave reception, and Figure 32.19 and Figure 32.20 show the operation timing in slave reception.

To set up and perform slave reception:

1. To initialize the IIC, follow the procedure in section 32.3.2. Initial Settings.
   After the initialization, the IIC stays in the standby state until it receives a slave address that matches.
2. After receiving a matching slave address, the IIC sets one of the associated ICSR1.HOA, GCA, and AASy (y = 0 to 2) flags to 1 on the rising edge of the 9th cycle of the SCL clock and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the 9th cycle of the SCL clock. If the value of the received R/W# bit is 0, the IIC continues to place itself in slave receive mode and sets the RDRF flag in ICSR2 to 1.
3. Check that the ICSR2.STOP flag is 0 and the ICSR2.RDRF flag is 1, and then dummy read the ICDRR register.
   The dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower 8 bits when the 10-bit address format is selected.
4. When ICDRR is read, the IIC automatically sets the ICSR2.RDRF flag to 0. If reading of ICDRR is delayed and the next byte is received while the RDRF flag is still set to 1, the IIC holds the SCLn line low until 1 SCL cycle before the point where RDRF must be set. In this case, reading ICDRR releases the SCLn line from being held at the low level. When the ICSR2.STOP flag is 1 and the ICSR2.RDRF flag is also 1, read ICDRR until all the data is completely received.
5. On detecting the stop condition, the IIC automatically clears the ICSR1.HOA, GCA, and AASy (y = 0 to 2) flags to 0.
6. Check that the ICSR2.STOP flag is 1, then set the ICSR2.STOP flag to 0 for the next transfer operation.

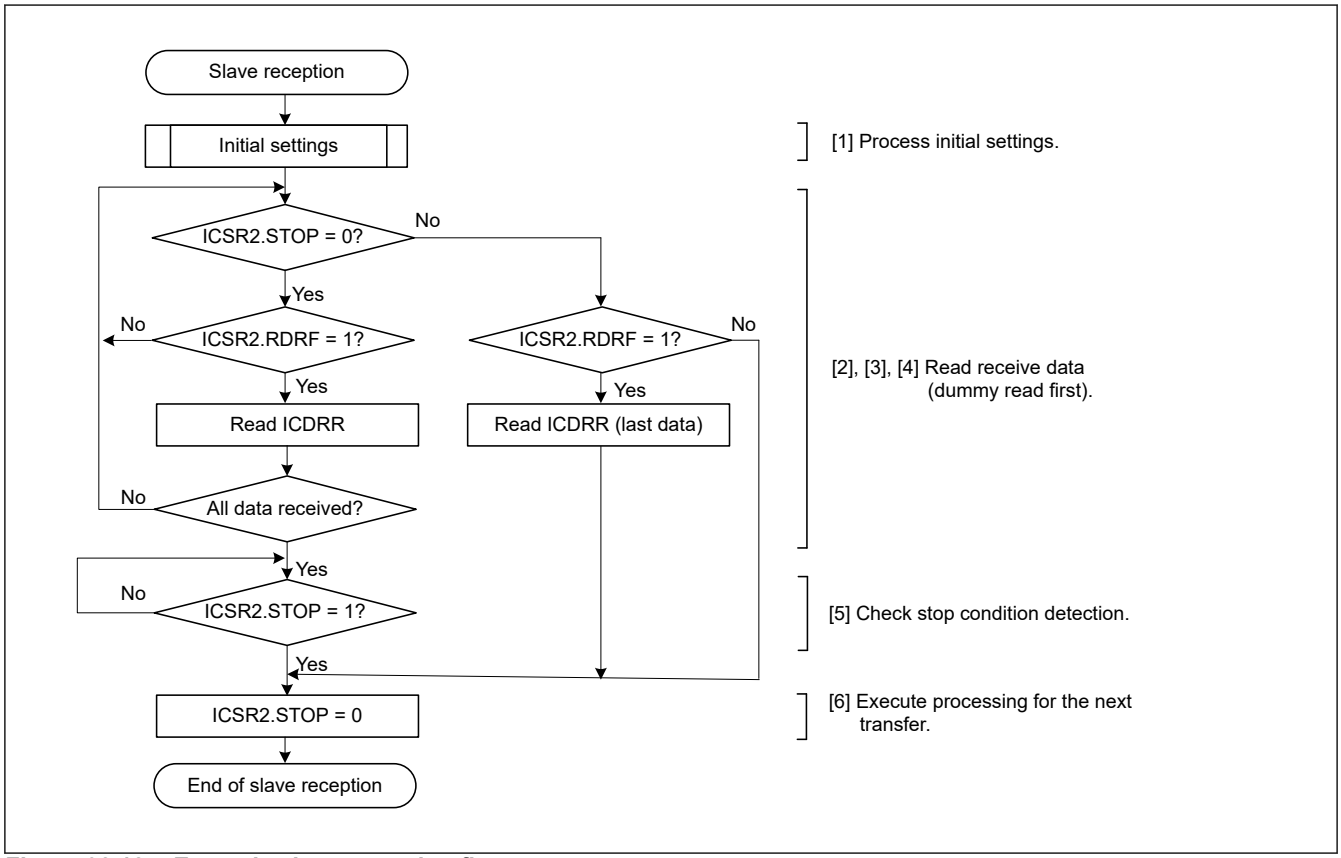


Figure 32.18 Example slave reception flow

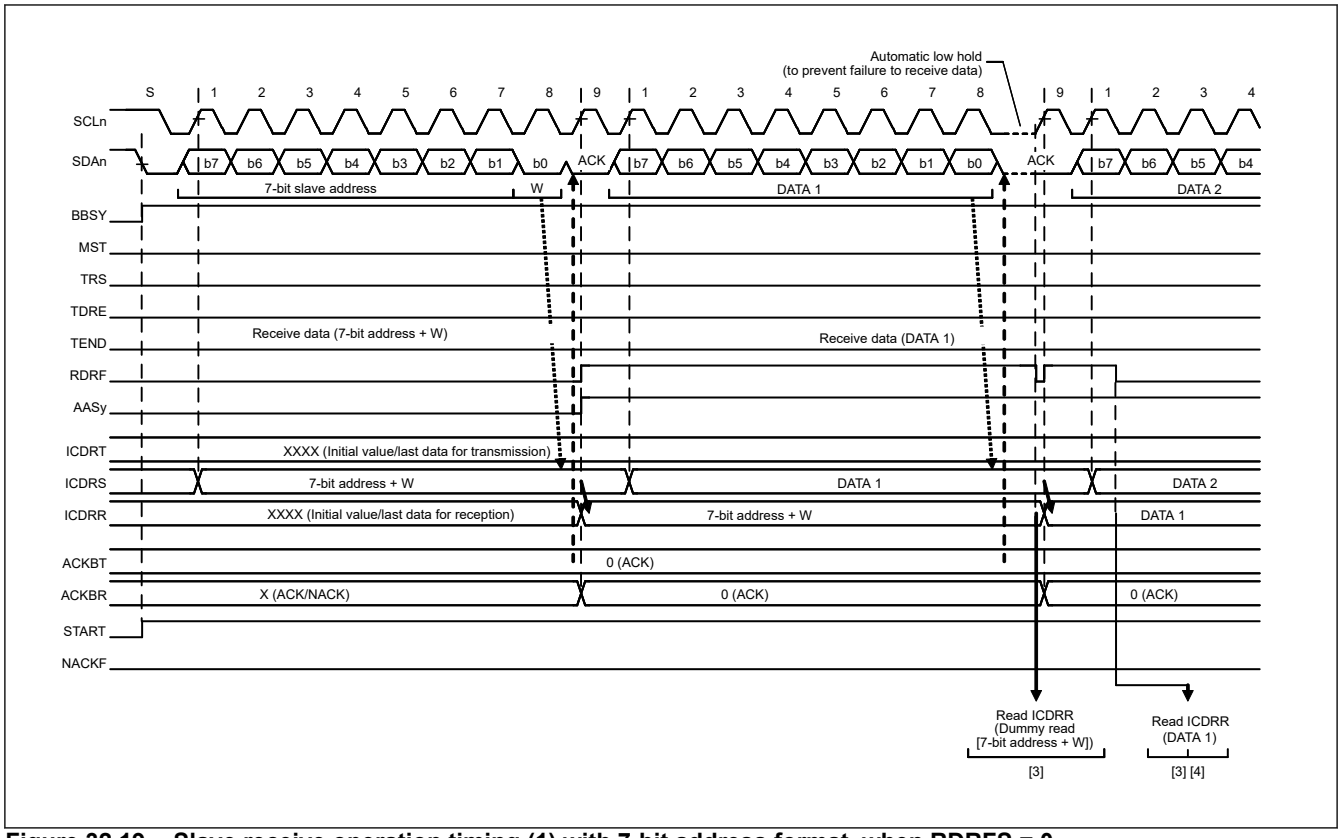


Figure 32.19 Slave receive operation timing (1) with 7-bit address format, when RDRFS = 0

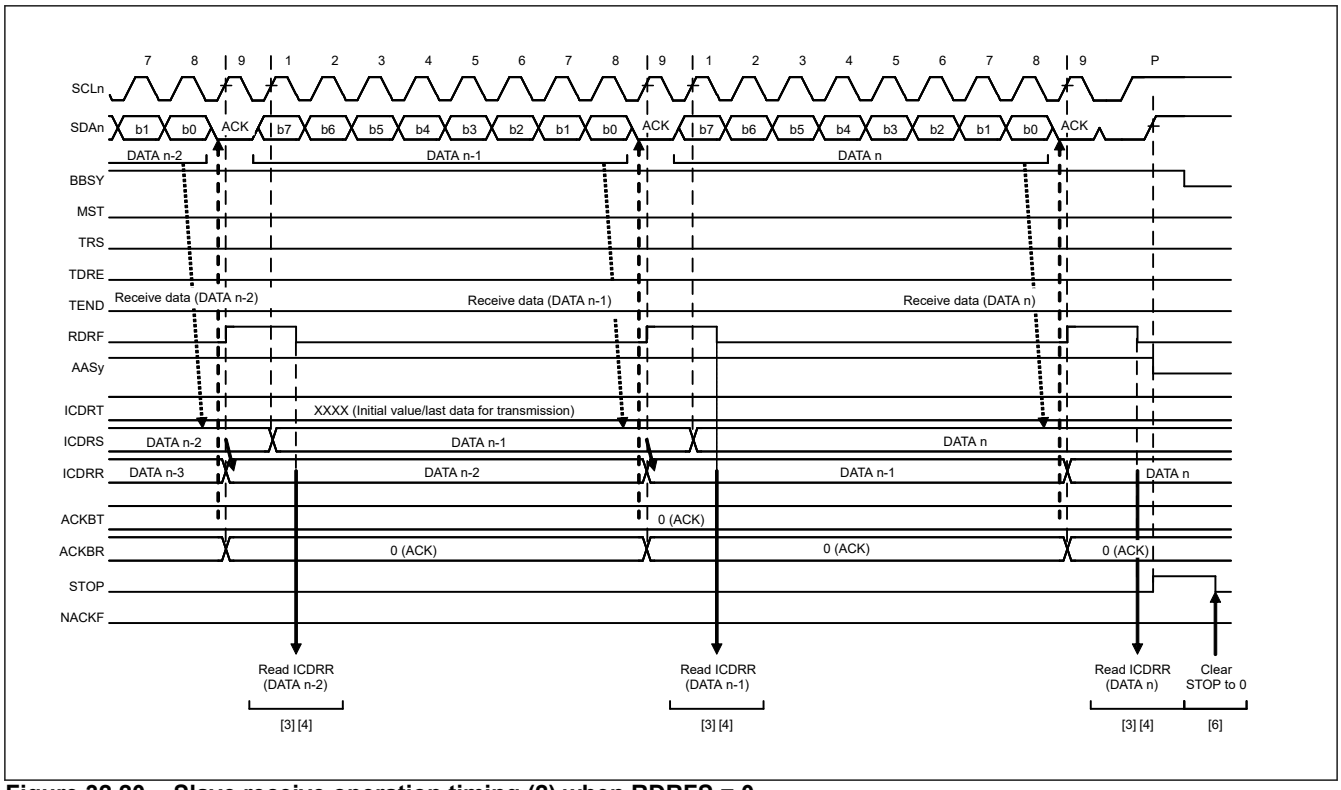


Figure 32.20 Slave receive operation timing (2) when RDRFS = 0

32.4 SCL Synchronization Circuit

For generation of the SCL clock, IIC starts counting the value for the high-level period specified in ICBRH when it detects a rising edge on the SCLn line, and drives the SCLn line low when it completes counting. When IIC detects the falling edge

of the SCLn line, it starts counting the value for the low-level period specified in ICBRL, and then stops driving the SCLn line, releasing the line when it completes counting. The IIC repeats this process to generate the SCL clock.

If multiple master devices are connected to the I^2C bus, a collision of SCL signals might arise because of contention with another master device. In such cases, the master devices must synchronize their SCL signals. Because this synchronization of SCL signals must be bit by bit, IIC is equipped with an SCL synchronization circuit to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCLn line while in master mode.

When IIC detects a rising edge on the SCLn line and starts counting the high-level period specified in ICBRH, and the level on the SCLn line falls because an SCL signal is being generated by another master device, IIC performs the following:

1. Stops counting when it detects the falling edge.
2. Drives the level on the SCLn line low.
3. Starts counting the low-level period specified in ICBRL.

When the IIC finishes counting the low-level period, it stops driving the SCLn line low to release the line. If the lowlevel period of the SCL clock signal from the other master device is longer than the low-level period set in the IIC, the low-level period of the SCL signal is extended. When the low-level period for the other master device ends, the SCL signal rises because the SCLn line is released. When the IIC finishes outputting the low-level period of the SCL clock, the SCLn line is released and the SCL clock rises. That is, when SCL signals from more than one master are contending, the high-level period of the SCL signal is synchronized with that of the clock with the narrower period, and the low-level period of the SCL signal is synchronized with that of the clock with the broader period. However, such synchronization of the SCL signal is only enabled when the SCLE bit in ICFER is set to 1.

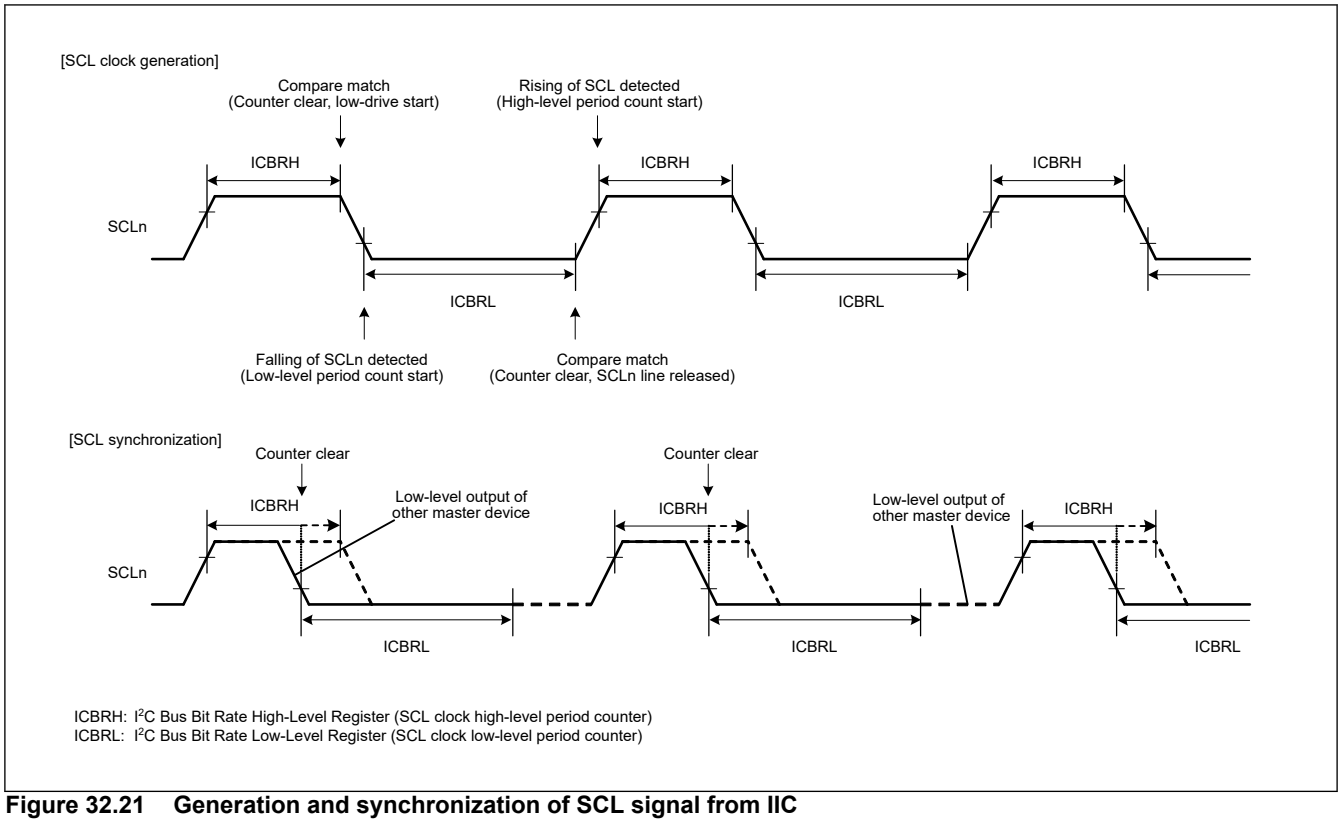


Figure 32.21 Generation and synchronization of SCL signal from IIC

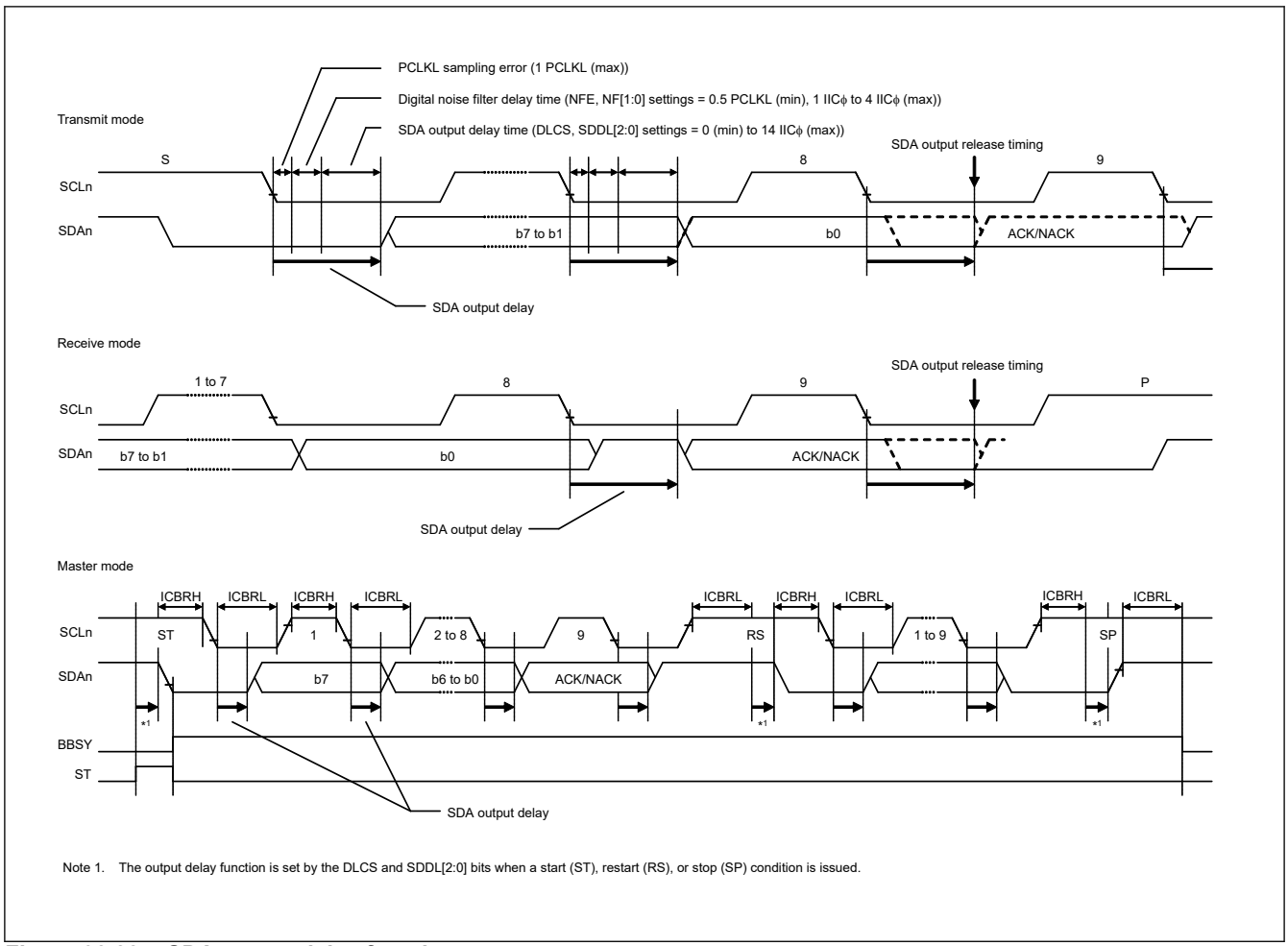
32.5 SDA Output Delay Function

The IIC module provides a function for delaying the output on the SDA line. The delay can be applied to all outputs on the SDA line, including the issuing of start, restart, and stop conditions, data, and the ACK and NACK signals.

With this function, SDA output is delayed from the detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval during which the SCL clock is low. This approach helps prevent erroneous operation of communications devices, with the aim of satisfying the 300 ns minimum data-hold time requirement of the SMBus

specification. The output delay function is enabled by setting the SDDL[2:0] bits in ICMR2 to any value other than 000b, and disabled by setting the same bits to 000b.

When the SDA output delay function is enabled, for example, the DLCS bit in ICMR2 selects the clock source for the SDA output delay counter, either as the internal base clock (IICΦ) for the IIC module or as the internal base clock divided by two (IICΦ/2). The counter counts the number of cycles set in the SDDL[2:0] bits in ICMR2. When the delay count is reached, the IIC module places the required output (start, restart, or stop condition, data, or an ACK or NACK signal) on the SDA line.



Note 1. The output delay function is set by the DLCS and SDDL[2:0] bits when a start (ST), restart (RS), or stop (SP) condition is issued.

Figure 32.22 SDA output delay function

32.6 Digital Noise Filter Circuits

The internal circuitry sees the states of the SCLn and SDAn pins through analog and digital noise-filter circuits. Figure 32.23 shows a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of the IIC consists of four flip-flop circuit stages connected in series, and a match-detection circuit.

The number of valid stages in the digital noise filter is selected in the NF[1:0] bits in ICMR3. The selected number of valid stages determines the noise-filtering capability as a period from 1 to 4 IICΦ cycles.

The input signal to the SCLn pin (or SDAn pin) is sampled on falling edges of the IICΦ signal. When the input signal level matches the output level of the number of valid flip-flop circuit stages as selected in the NF[1:0] bits in ICMR3, the signal level is conveyed to the subsequent stage. If the signal levels do not match, the previous value is saved.

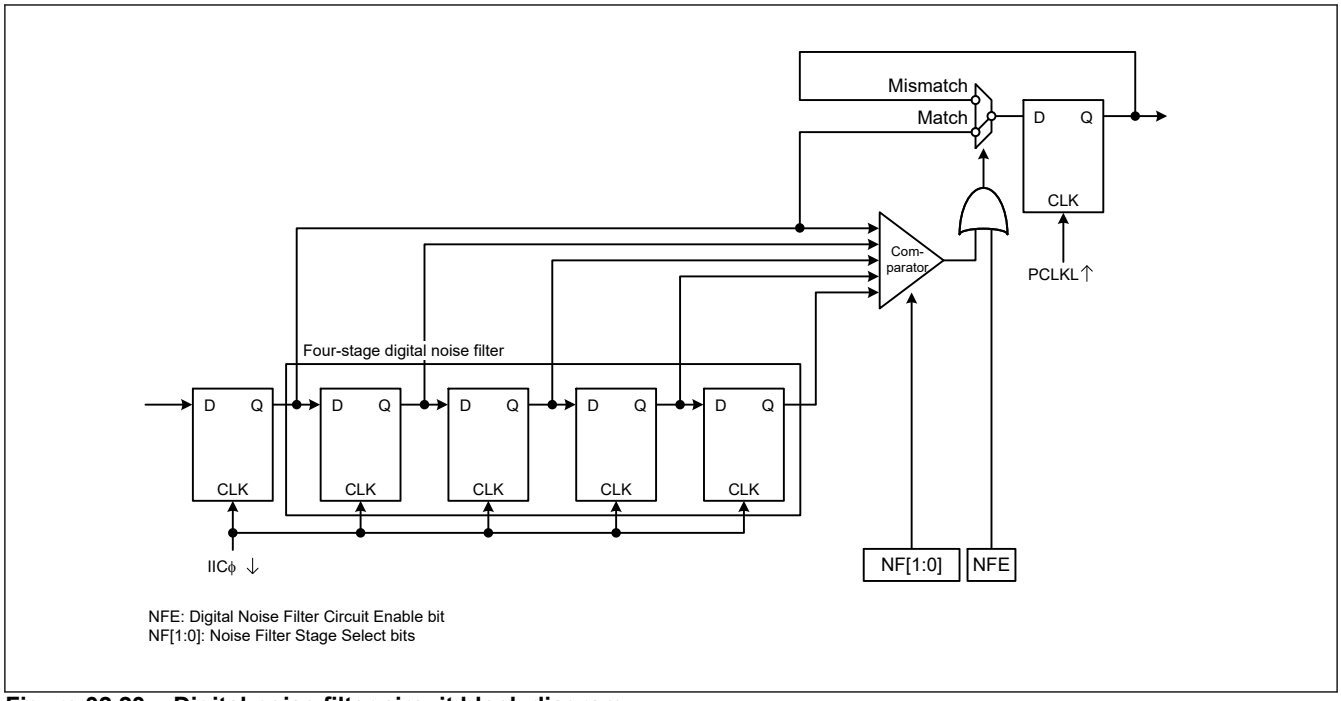


Figure 32.23 Digital noise filter circuit block diagram

32.7 Address Match Detection

IIC can set three unique slave addresses in addition to the general call address and host address. The slave addresses can be 7-bit or 10-bit slave addresses.

32.7.1 Slave-Address Match Detection

IIC can set three unique slave addresses and has a slave address detection function for each unique slave address. When the SARyE bit (y = 0 to 2) in ICSER is set to 1, the slave addresses set in SARUy and SARLy (y = 0 to 2) can be detected.

When the IIC detects a match of the set slave address, the associated AASy (y = 0 to 2) flag in ICSR1 is set to 1 on the rising edge of the 9th SCL clock cycle, and the RDRF flag in ICSR2 or the TDRE flag in ICSR2 is set to 1 by the subsequent R/W# bit. This causes a receive data full interrupt (IICn_RXI) or transmit data empty interrupt (IICn_TXI) to be generated. The AASy flag identifies which slave address is specified.

Figure 32.24 to Figure 32.26 show the AASy flag set timing in three cases.

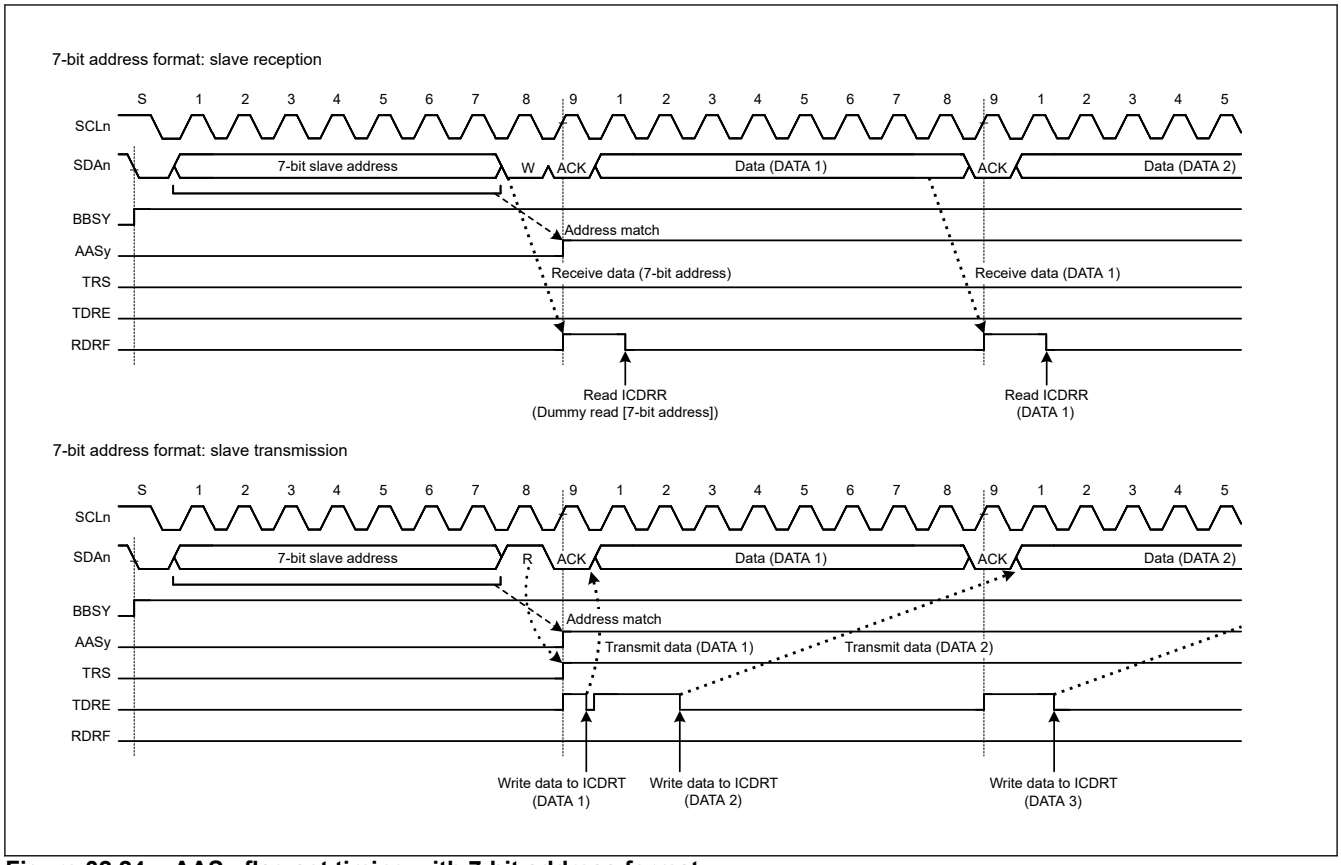


Figure 32.24 AASy flag set timing with 7-bit address format

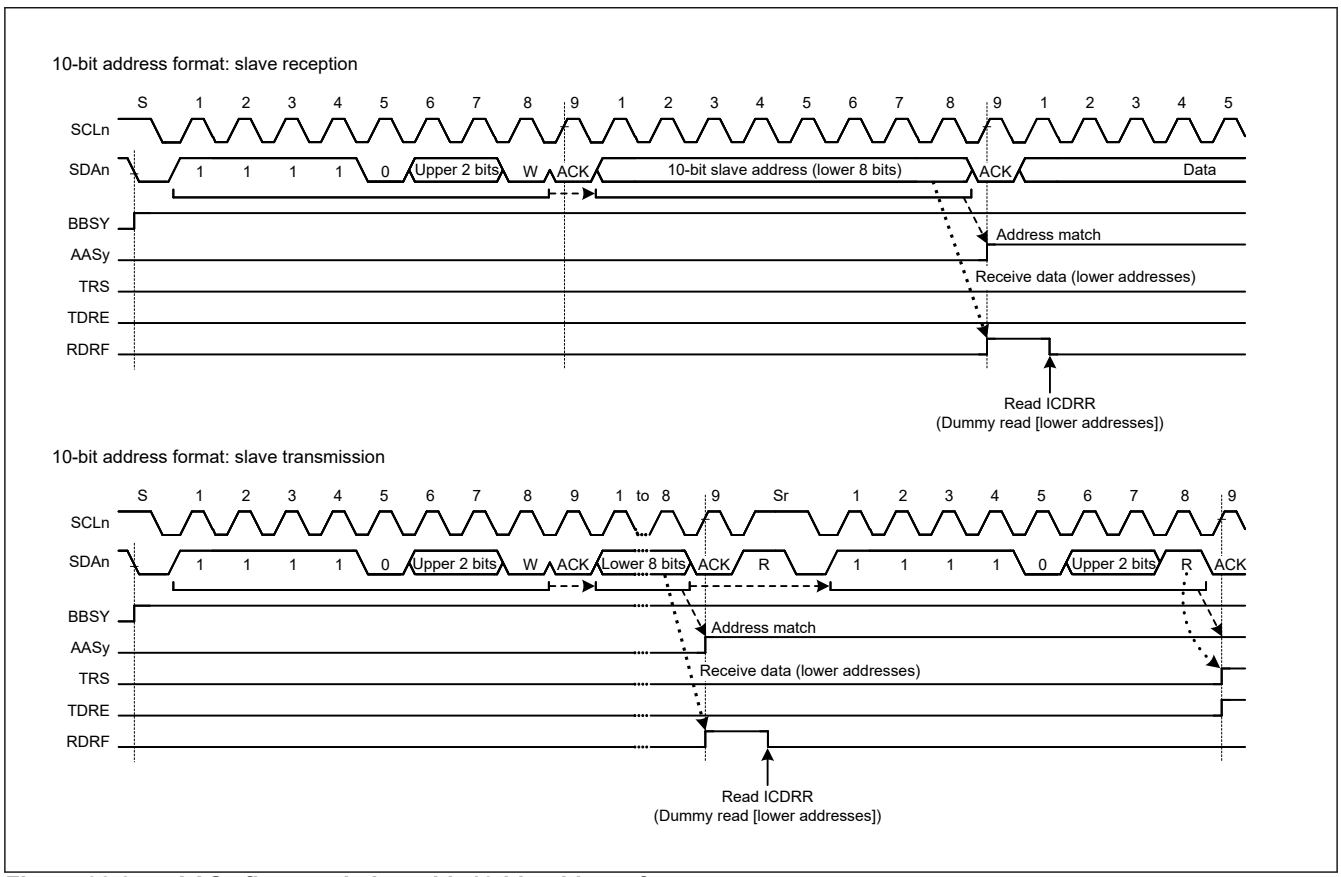


Figure 32.25 AASy flag set timing with 10-bit address format

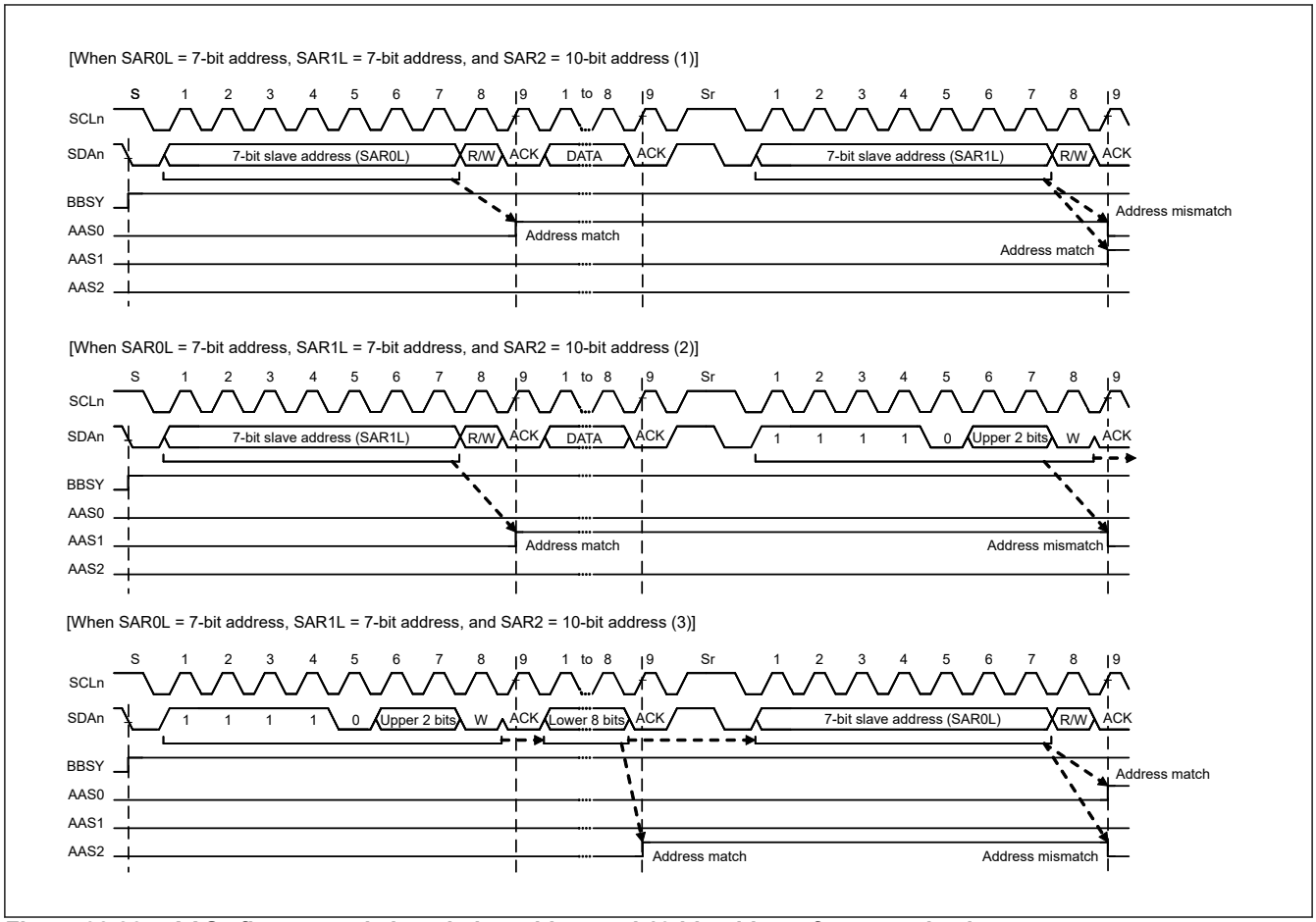


Figure 32.26 AASy flag set and clear timing with 7- and 10-bit address formats mixed

32.7.2 Detection of General Call Address

The IIC provides detection of the general call address (0000 000b + 0 [W]). General call address detection is enabled by setting the GCAE bit in ICSER to 1.

If the address received after a start or restart condition is issued is 0000 000b + 1 [R] (start byte), the IIC recognizes this as the address of a slave device with an all-zero address, but not as the general call address.

When the IIC detects the general call address, both the GCA flag in ICSR1 and the RDRF flag in ICSR2 set to 1 on the rising edge of the 9th cycle of the SCL clock. This leads to the generation of a receive data full interrupt (IICn_RXI). The value of the GCA flag can be checked to confirm that the general call address is transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

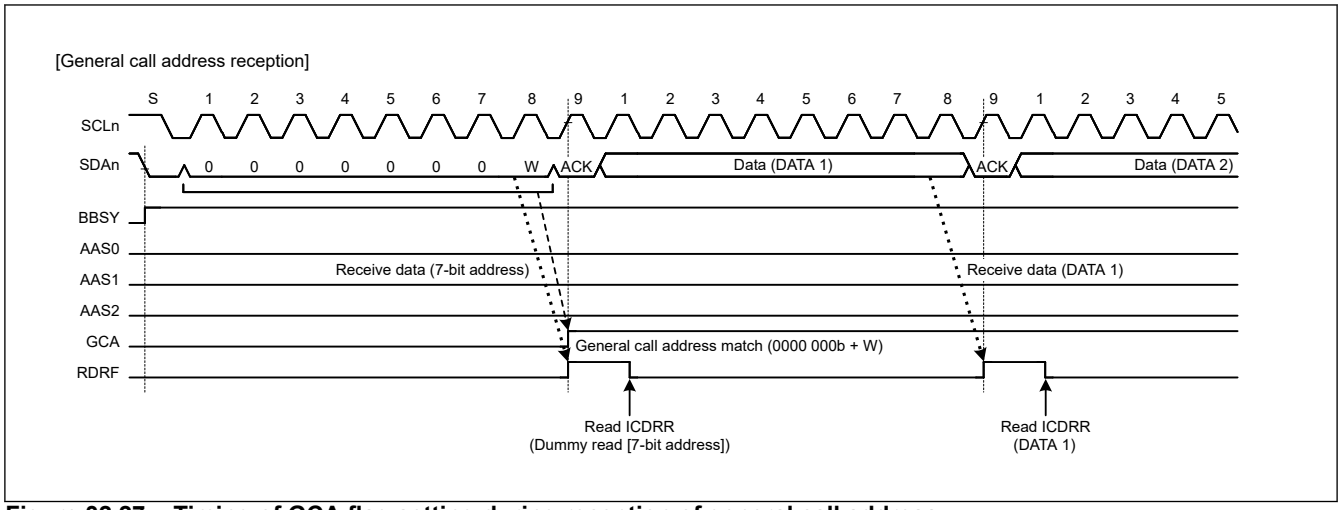


Figure 32.27 Timing of GCA flag setting during reception of general call address

32.7.3 Device ID Address Detection

The IIC module provides detection of the device ID address in compliance with the I^2C bus specification, revision 03.

When IIC receives 1111 100b as the first byte after a start or restart condition is issued with the DIDE bit in ICSER set to 1, it recognizes the address as a device ID, sets the DID flag in ICSR1 to 1 on the rising edge of the 8th SCL clock cycle when the subsequent R/W# bit is 0, and then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the IIC sets the associated AASy (y = 0 to 2) flag in ICSR1 to 1.

When the first byte received after the issue of a start or restart condition matches the device ID address (1111 100b) again and the subsequent R/W# bit is 1, IIC does not compare the second and subsequent bytes, and sets the ICSR2.TDRE flag to 1.

In the device ID address detection function, IIC sets the DID flag to 0 if a match with the IIC slave address is not obtained, or a match with the device ID address is not obtained after a match with the IIC slave address, and a restart condition is detected. If the first byte after the detection of a start or restart condition matches the device ID address (1111 100b) and the R/W# bit is 0, IIC sets the DID flag to 1 and compares the second and subsequent bytes with the slave address of IIC. If the R/W# bit is 1, the DID flag holds the previous value and IIC does not compare the second and subsequent bytes. In this way, reception of a device ID address can be checked by reading the DID flag after confirming that TDRE = 1.

Additionally, prepare the device ID fields (3 bytes: 12 bits indicating the manufacturer + 9 bits identifying the part + 3 bits indicating the revision) that must be sent to the host after reception of a continuous device ID field as normal transmit data. For details on the information that must be included in device ID fields, contact NXP Semiconductors.

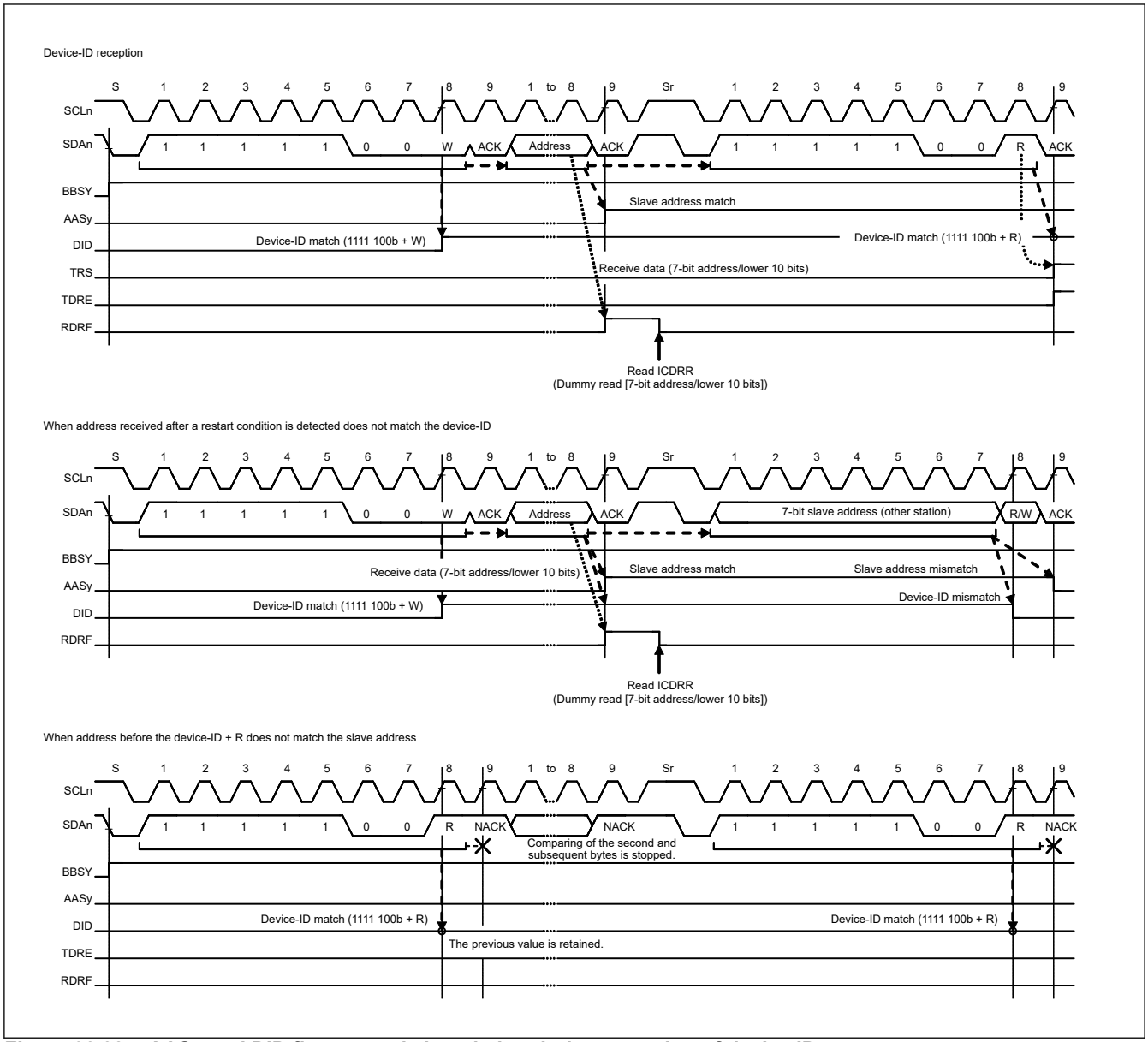


Figure 32.28 AASy and DID flag set and clear timing during reception of device ID

32.7.4 Host Address Detection

The IIC provides host address detection while the SMBus is operating. When the HOAE bit in ICSER is set to 1 while the SMBS bit in ICMR3 is 1, the IIC can detect the host address (0001 000b) in slave receive mode (MST and TRS bits = 00b in ICCR2).

When the IIC detects the host address, the HOA flag in ICSR1 is set to 1 on the rising edge of the 9th SCL clock cycle. At the same time, the RDRF flag in ICSR2 is set to 1 when the R/W# bit is 0. This causes a receive data full interrupt (IICn_RXI) to be generated. The HOA flag indicates that the host address was sent from another device.

If the bit following the host address (0001 000b) is a read bit (R/W# bit = 1), the IIC can also detect the host address. After the host address is detected, the IIC operates in the same manner as in normal slave operation.

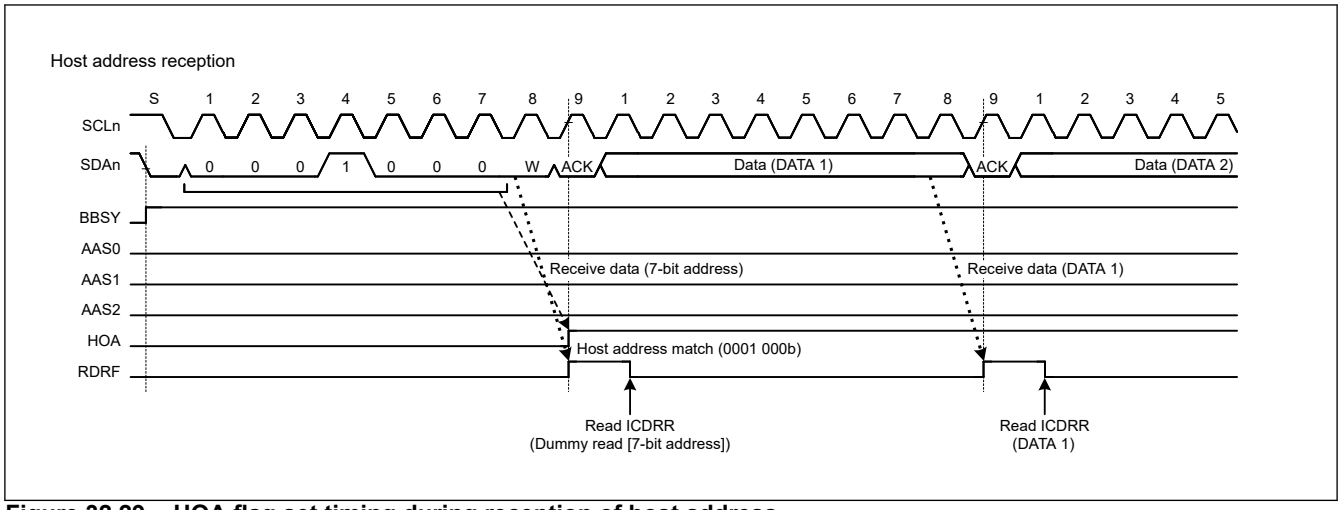


Figure 32.29 HOA flag set timing during reception of host address

32.8 Automatic Low-Hold Function for SCL

32.8.1 Function to Prevent Wrong Transmission of Transmit Data

If the I^{2}C Bus Shift Register (ICDRS) is empty when data has not been written to the I^{2}C Bus Transmit Data Register (ICDRT) with the IIC in transmission mode (TRS bit = 1 in ICCR2), the SCLn line is automatically held at the low level over subsequent intervals. This low-hold period is extended until the transmit data is written, which prevents the unintended transmission of erroneous data.

Master transmit mode

- Low-level interval after a start or restart condition is issued
- Low-level interval between the 9th clock cycle of one transfer and the 1st clock cycle of the next.

Slave transmit mode

- Low-level interval between the 9th clock cycle of one transfer and the 1st clock cycle of the next.

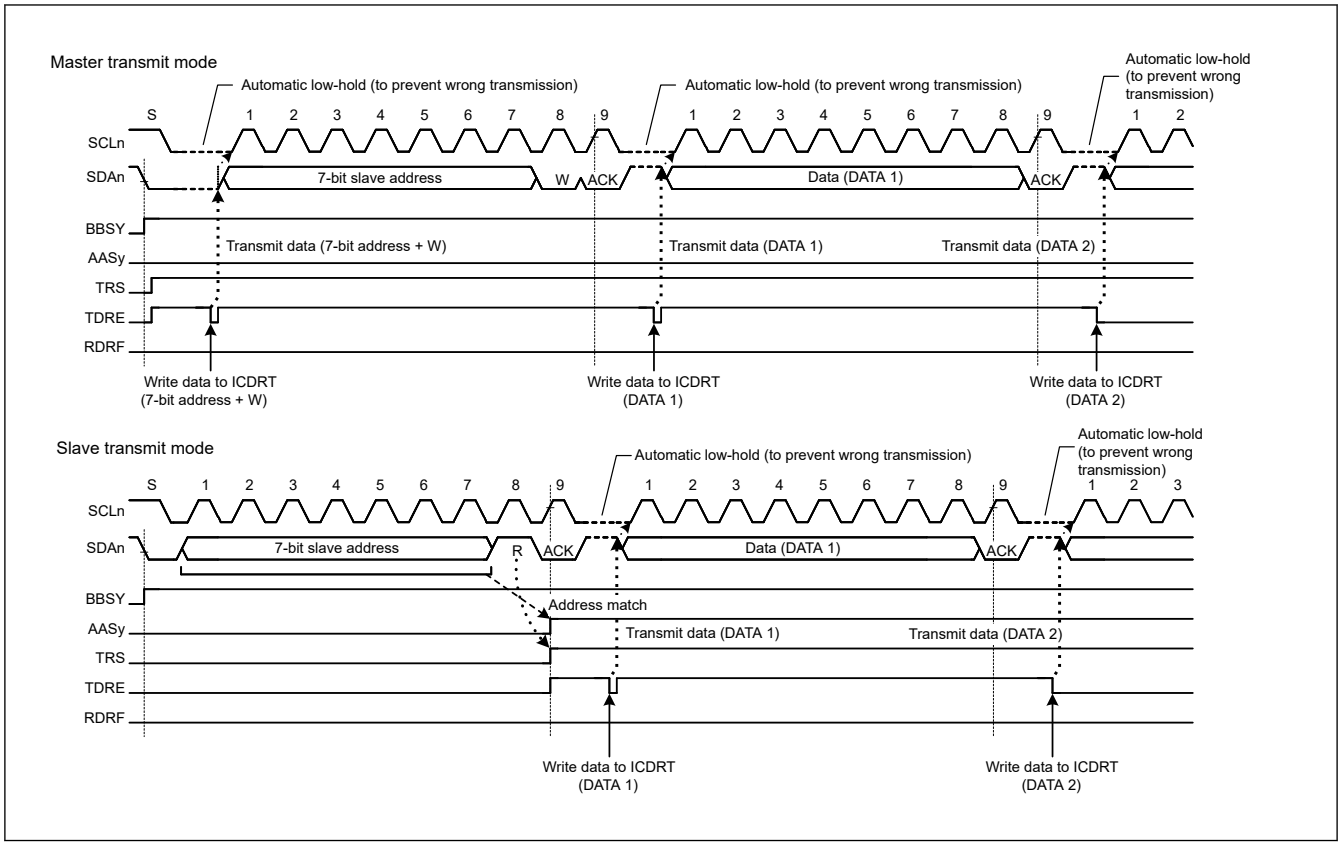


Figure 32.30 Automatic low-hold operation in transmit mode

32.8.2 NACK Reception Transfer Suspension Function

This function suspends transfer operation when NACK is received in transmit mode (TRS bit = 1 in ICCR2). This function is enabled when the NACKE bit in ICFER is set to 1. If the next transmit data is already written (TDRE flag = 0 in ICSR2) when NACK is received, the next data transmission on the falling edge of the 9th SCL clock cycle is automatically suspended. This prevents the SDAn line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (NACKF flag = 1 in ICSR2), transmit and receive operations are discontinued. To restore transmit or receive operations, set the NACKF flag to 0. In master transmit mode, after a restart or stop condition is issued, set the NACKF flag to 0, and then issue a start condition again.

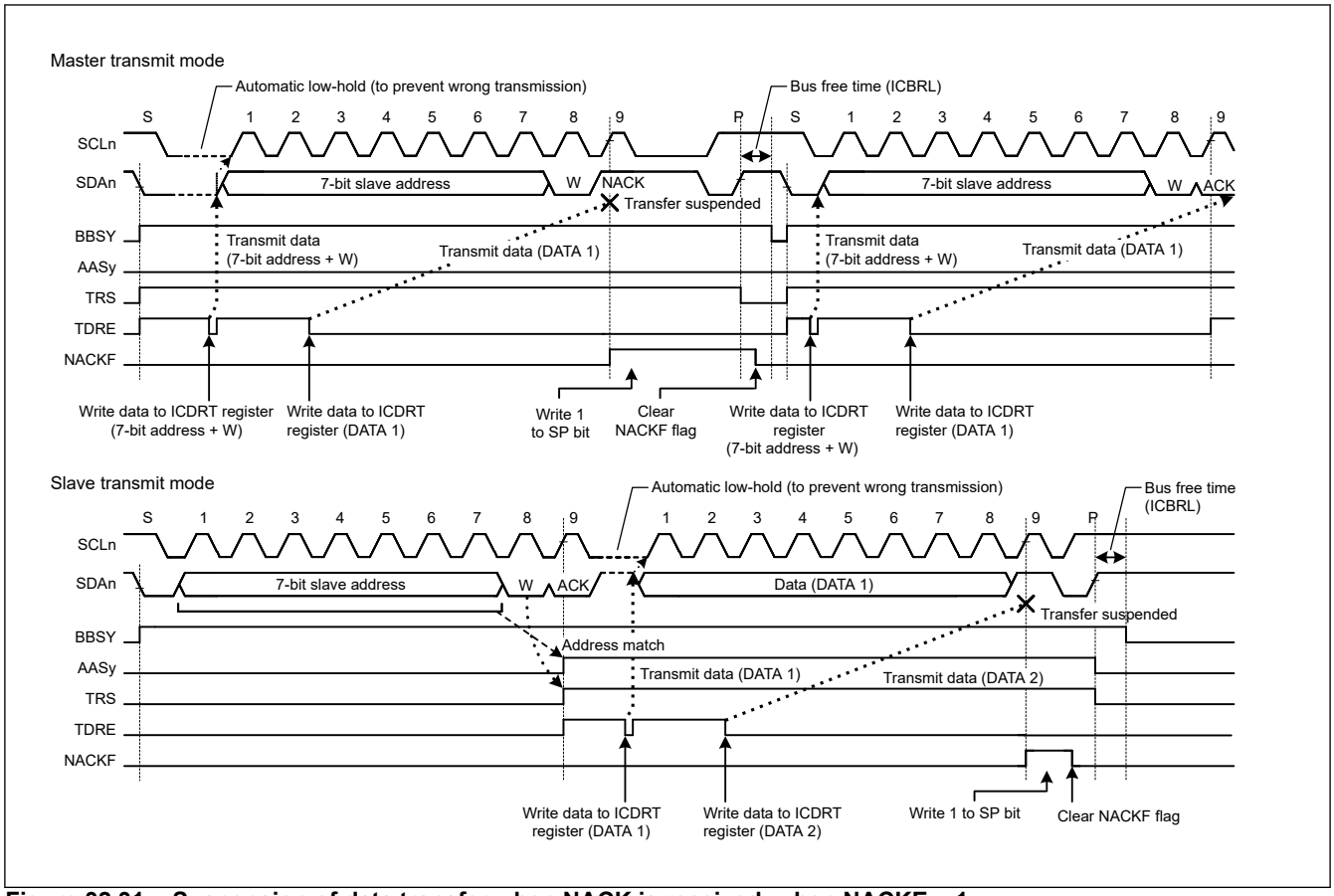


Figure 32.31 Suspension of data transfer when NACK is received, when NACKE = 1

32.8.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (ICDRR) read is delayed for a period of one transfer frame or more with receive data full (RDRF flag = 1 in ICSR2) in receive mode (TRS = 0 in ICCR2), the IIC holds the SCLn line low automatically immediately before the next data is received to prevent failure to receive data.

This function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, the IIC slave address is designated after a stop condition is issued. This function does not interfere with other communication because the IIC does not hold the SCLn line low when a mismatch with its own slave address occurs after a stop condition is issued.

Periods in which the SCLn line is held low can be selected with a combination of the WAIT and RDRFS bits in ICMR3.

(1) 1-byte receive operation and automatic low-hold function using the WAIT bit

When the WAIT bit in ICMR3 is set to 1, IIC performs a 1-byte receive operation using the WAIT bit function. Additionally, when the ICMR3.RDRFS bit is 0, IIC automatically sends the ACKBT bit value in ICMR3 for the acknowledge bit in the period from the falling edge of the 8th SCL clock cycle to the falling edge of the 9th SCL clock cycle, and automatically holds the SCLn line low on the falling edge of the 9th SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from ICDRR, which enables byte-wise receive operation.

The WAIT bit function is enabled for receive frames after a match with the IIC slave address, including the general call address and host address, is obtained in master or slave receive mode.

(2) 1-byte receive operation (ACK/NACK transmission control) and automatic low-hold function using the RDRFS bit

When the RDRFS bit in ICMR3 is set to 1, the IIC performs a 1-byte receive operation using the RDRFS bit function.

When the RDRFS bit is set to 1, the RDRF (receive data full) flag in ICSR2 is set to 1 on the rising edge of the 8th SCL clock cycle, and the SCLn line is automatically held low on the falling edge of the 8th SCL clock cycle. This low-hold is

released by writing a value to the ACKBT bit in ICMR3, but cannot be released by reading data from ICDRR, which enables receive operation through the ACK or NACK transmission control based on the data received in byte units.

The RDRFS bit function is enabled for receive frames after a match with the IIC slave address, including the general call address and host address, is obtained in master or slave receive mode.

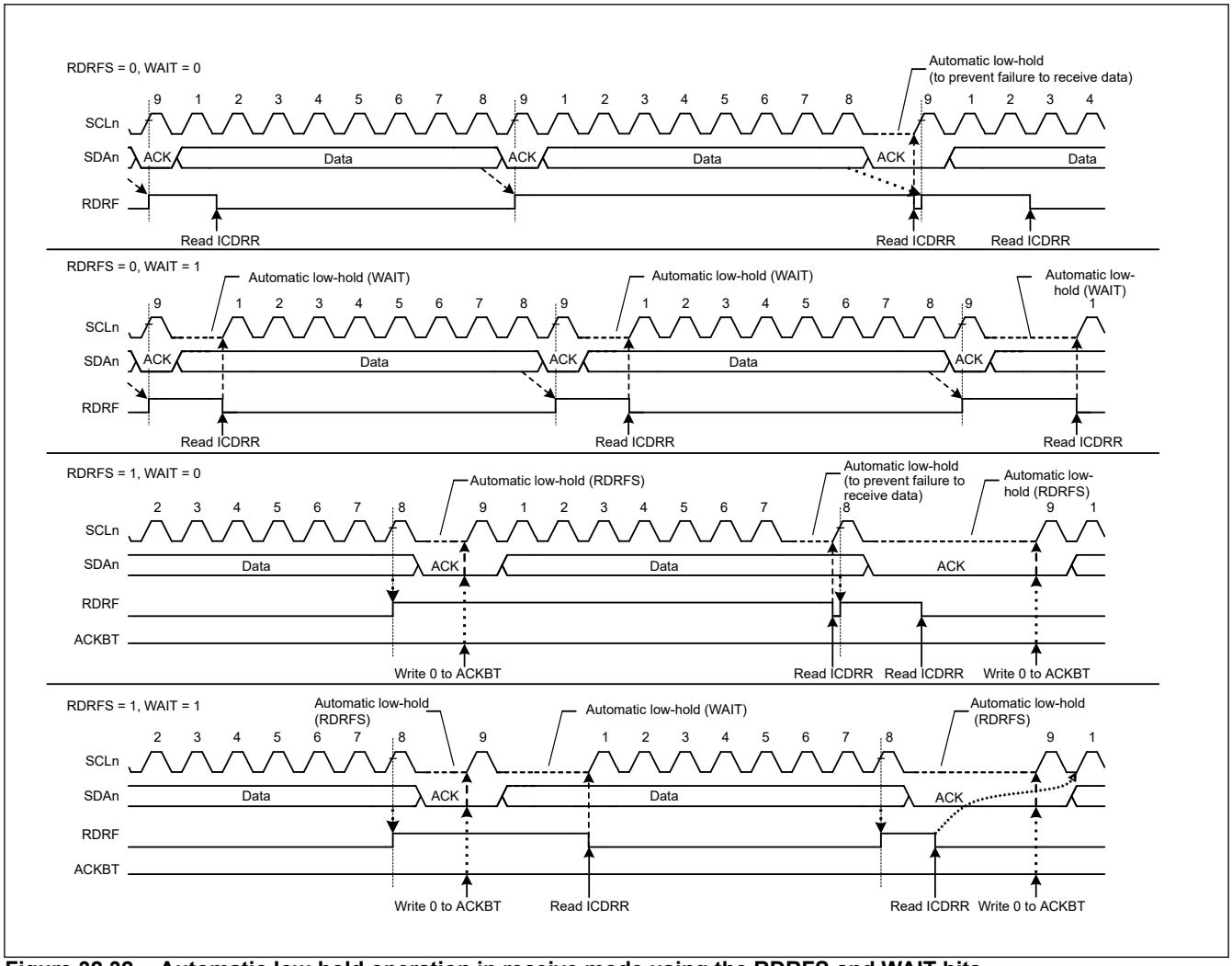


Figure 32.32 Automatic low-hold operation in receive mode using the RDRFS and WAIT bits

32.9 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I^2C bus standard, IIC provides functions to prevent double-issue of a start condition, detect arbitration-lost during the transmission of NACK, and detect arbitration lost in slave transmit mode.

32.9.1 Master Arbitration-Lost Detection (MALE Bit)

IIC drives the SDAn line low to issue a start condition. However, if the SDAn line was already driven low by another master device issuing a start condition, IIC regards its own start condition as an error and considers this a loss in arbitration.

Priority is given to transfer by the other master device. Similarly, if a request to issue a start condition is made by setting the ST bit in ICCR2 to 1 while the bus is busy (BBSY flag = 1 in ICCR2), IIC regards this as a double-issuing-of-start-condition error and considers itself to have lost the arbitration. This prevents a failure of transfer resulting from a start condition issued while transfer is in progress.

When a start condition is issued successfully, if the transmit data including the address bits (internal SDA output level) and the level on the SDAn line do not match (high output as the internal SDA output, meaning the SDAn pin is in the high-impedance state) and a low level is detected on the SDAn line, IIC loses the arbitration.

After a loss in arbitration of mastership, the IIC immediately enters slave receive mode. If a slave address, including the general call address, matches its own address at this time, the IIC continues in slave operation.

A loss in arbitration of mastership is detected when the following conditions are met while the MALE bit in ICFER is 1 (master arbitration-lost detection enabled).

[Master arbitration-lost conditions]

- Mismatching of the internal level for output on SDA and the level on the SDAn line after a start condition was issued by setting the ST bit in ICCR2 to 1 while the BBSY flag in ICCR2 is set to 0 (erroneous issuing of a start condition)
- Setting of the ST bit in ICCR2 to 1 (start condition double-issue error) while the BBSY flag is 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDAn line in master transmit mode (MST and TRS bits = 11b in ICCR2).

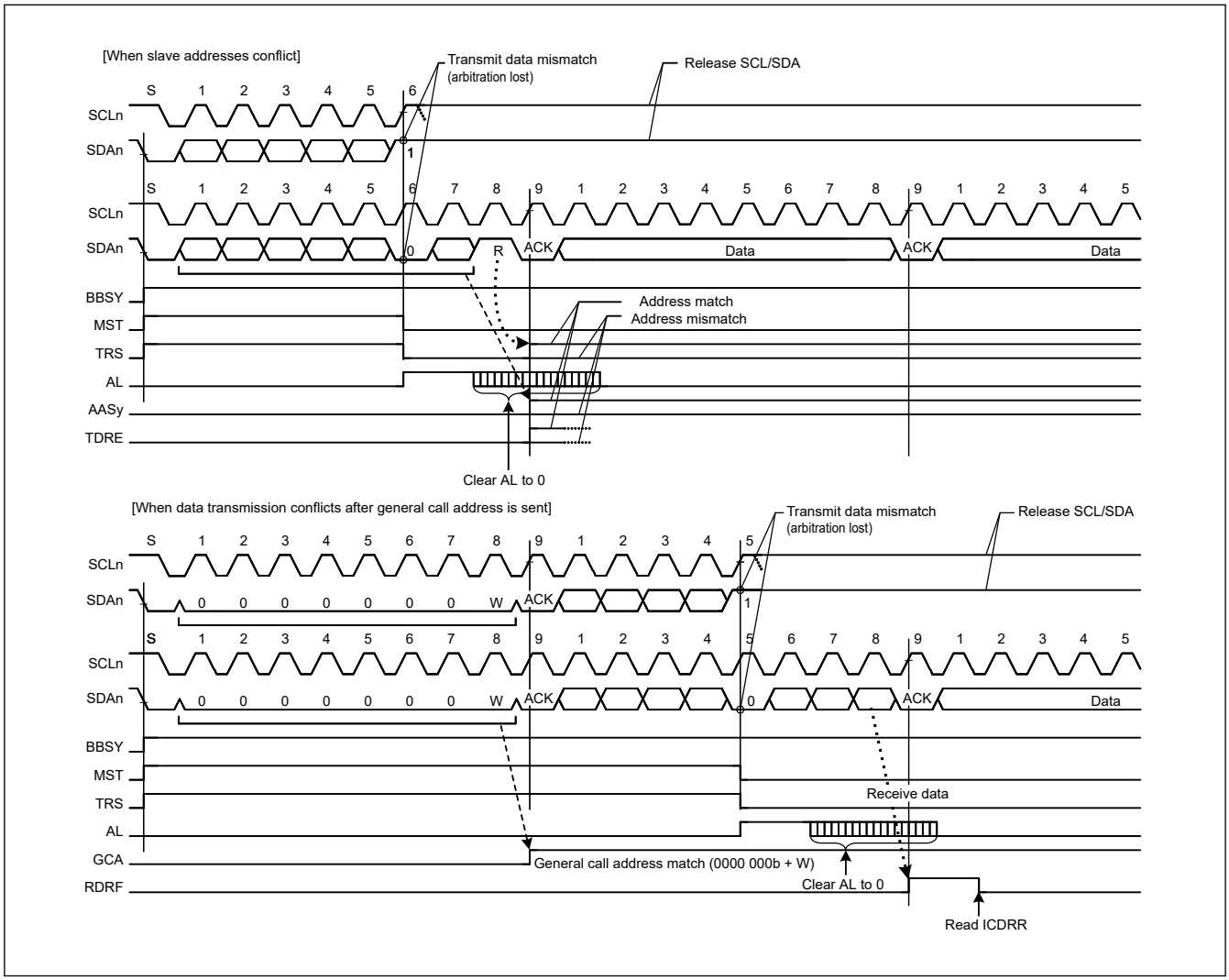


Figure 32.33 Examples of master arbitration-lost detection when MALE = 1

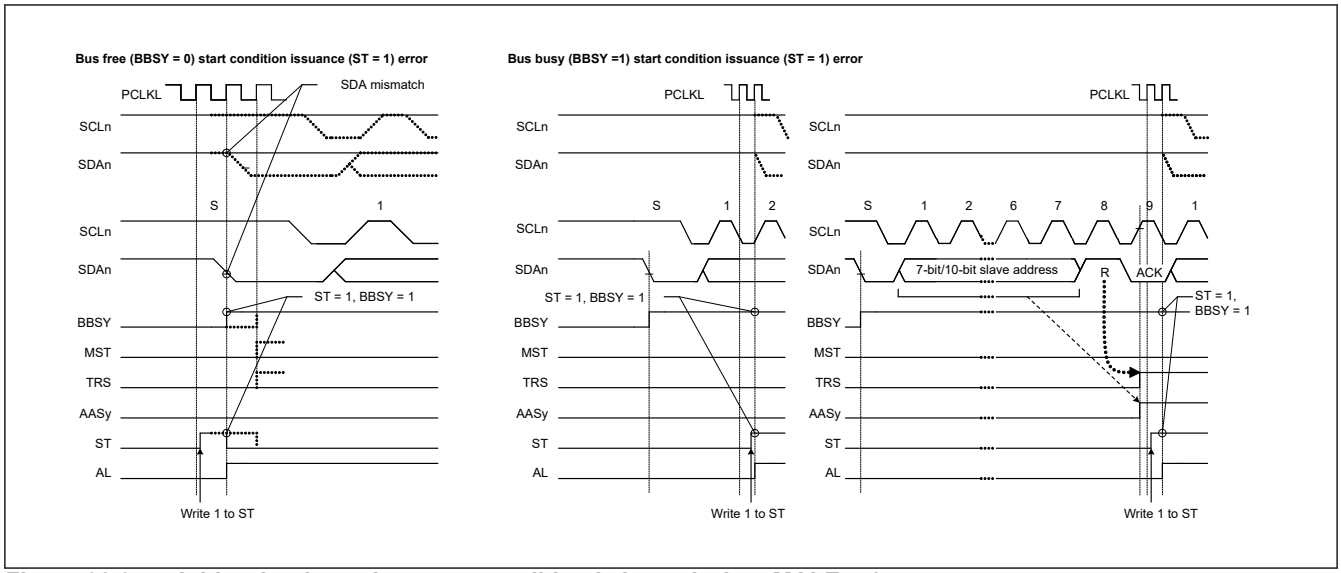


Figure 32.34 Arbitration-lost when start condition is issued when MALE = 1

32.9.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit)

This function causes arbitration to be lost if the internal SDA output level does not match the level on the SDAn line during transmission of NACK in receive mode. Arbitration is lost because of a conflict between NACK and ACK transmissions when two or more master devices receive data from the same slave device simultaneously in a multi-master system. Such conflict occurs when multiple master devices send or receive the same information through a single slave device. Figure 32.35 shows an example of arbitration-lost detection during the transmission of NACK.

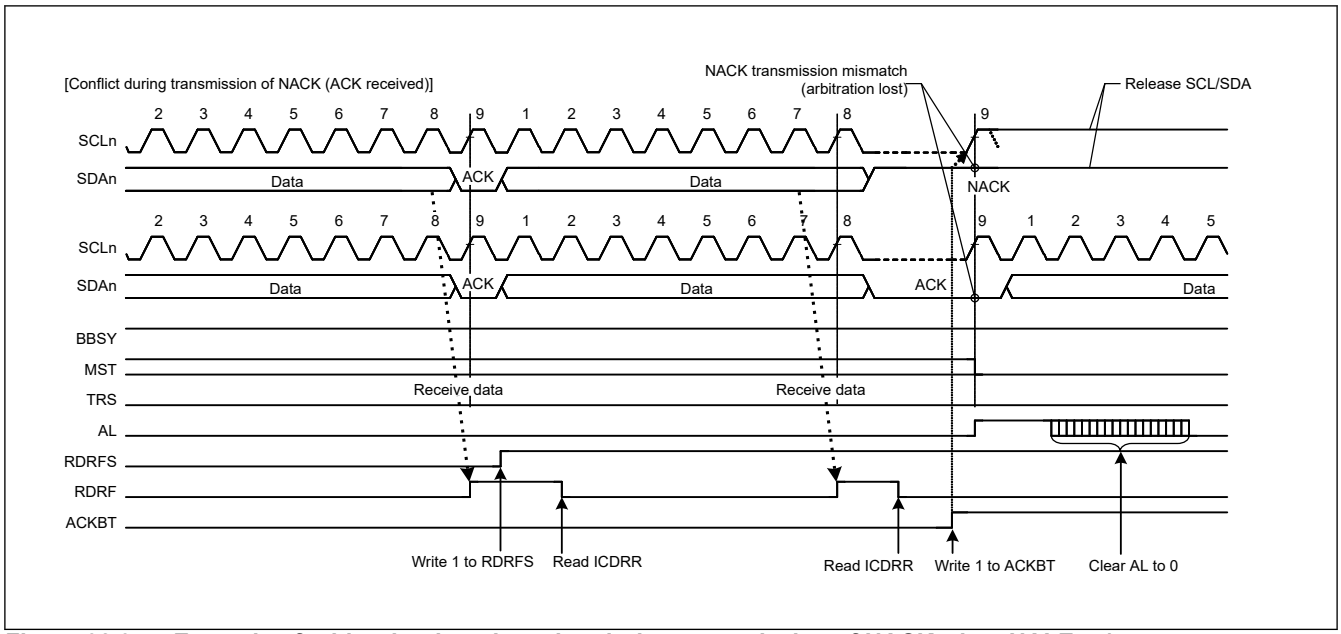


Figure 32.35 Example of arbitration-lost detection during transmission of NACK when NALE = 1

The following description explains arbitration-lost detection using an example in which two master devices (master A and master B) and a single slave device are connected through the bus. In this example, master A receives 2 bytes of data from the slave device, and master B receives 4 bytes of data from the slave device.

If master A and master B access the slave device simultaneously, because the slave address is identical, arbitration is not lost in either master A or B during access to the slave device. Therefore, both master A and master B recognize that they have obtained the bus mastership and operate as such. Master A sends NACK when it has received the 2 final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received the required 4 bytes of data.

The NACK transmission from master A and the ACK transmission from master B conflict. In general, if a conflict like this occurs, master A cannot detect the ACK transmitted by master B and issues a stop condition. The stop condition issue conflicts with the SCL clock output of master B, which disrupts communication.

When IIC receives ACK during transmission of NACK, it detects a defeat in conflict with other master devices and causes arbitration to be lost. If arbitration is lost during transmission of NACK, IIC immediately cancels the slave match condition and enters slave receive mode. This prevents a stop condition from being issued, preventing a communication failure on the bus.

Similarly, in the ARP command processing of SMBus, the function to detect loss of arbitration during transmission of NACK is also available for eliminating the extra clock cycle processing, such as 0xFF transmission processing, which is required if the UDID (Unique Device Identifier) of the assigned address does not match in the Get UDID general processing after the Assign Address command.

The IIC detects arbitration-lost during transmission of NACK when the following condition is met with the NALE bit in ICFER set to 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

- When the internal SDA output level does not match the SDAn line (ACK is received) during transmission of NACK (ACKBT bit = 1 in ICMR3).

32.9.3 Slave Arbitration-Lost Detection (SALE Bit)

This function causes arbitration to be lost if the transmit data (internal SDA output level) and the level on the SDAn line do not match (high output as the internal SDA output, meaning the SDAn pin is in the high-impedance state), and the low level is detected on the SDAn line in slave transmit mode. This arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) over an SMBus.

When IIC loses slave arbitration, IIC is immediately released from the slave-matched state and enters slave receive mode.

This function can detect conflicts of data during transmission of UDIDs over an SMBus and eliminates subsequent redundant processing for the transmission of 0xFF.

IIC detects slave arbitration-lost when the following condition is met with the SALE bit in ICFER set to 1 (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

- When transmit data excluding acknowledge (internal SDA output level) does not match the SDAn line in slave transmit mode (MST and TRS bits = 01b in ICCR2).

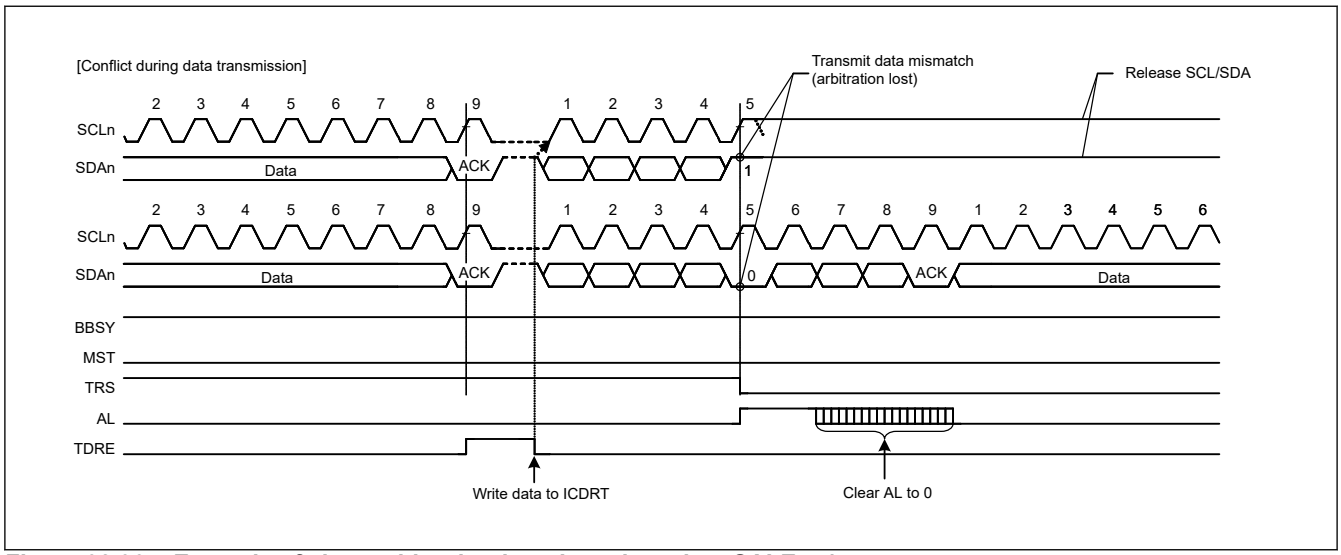


Figure 32.36 Example of slave arbitration-lost detection when SALE = 1

32.10 Start, Restart, and Stop Condition Issuing Function

32.10.1 Issuing a Start Condition

IIC issues a start condition when the ST bit in ICCR2 is set to 1. When the ST bit is set to 1, a start condition request is made, and IIC issues a start condition when the BBSY flag in ICCR2 is 0 (bus free state). When a start condition is issued normally, IIC automatically shifts to the master transmit mode.

To issue a start condition:

1. Drive the SDAn line low (high level to low level).
2. Ensure that the time set in ICBRH and the start condition hold time elapse.
3. Drive the SCLn line low (high level to low level).
4. Detect low level on the SCLn line and ensure the low-level period of the SCLn line set in ICBRL elapses.

32.10.2 Issuing a Restart Condition

IIC issues a restart condition when the RS bit in ICCR2 is set to 1.

When the RS bit is set to 1, a restart condition request is made, and IIC issues a restart condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).

To issue a restart condition:

1. Release the SDAn line.
2. Ensure that the low-level period of the SCLn line set in ICBRL elapses.
3. Release the SCLn line (low level to high level).
4. Detect a high level on the SCLn line and ensure the time set in ICBRL and the restart condition setup time elapse.
5. Drive the SDAn line low (high level to low level).
6. Ensure the time set in ICBRH and the restart condition hold time elapse.
7. Drive the SCLn line low (high level to low level).
8. Detect a low level on the SCLn line and ensure the low-level period of the SCLn line set in ICBRL elapses.

Note: When issuing restart condition requests, write the slave address to ICDRT after confirming that ICCR2.RS = 0. Data written while ICCR2.RS = 1 is not forwarded because of the retransmission condition before the occurrence.

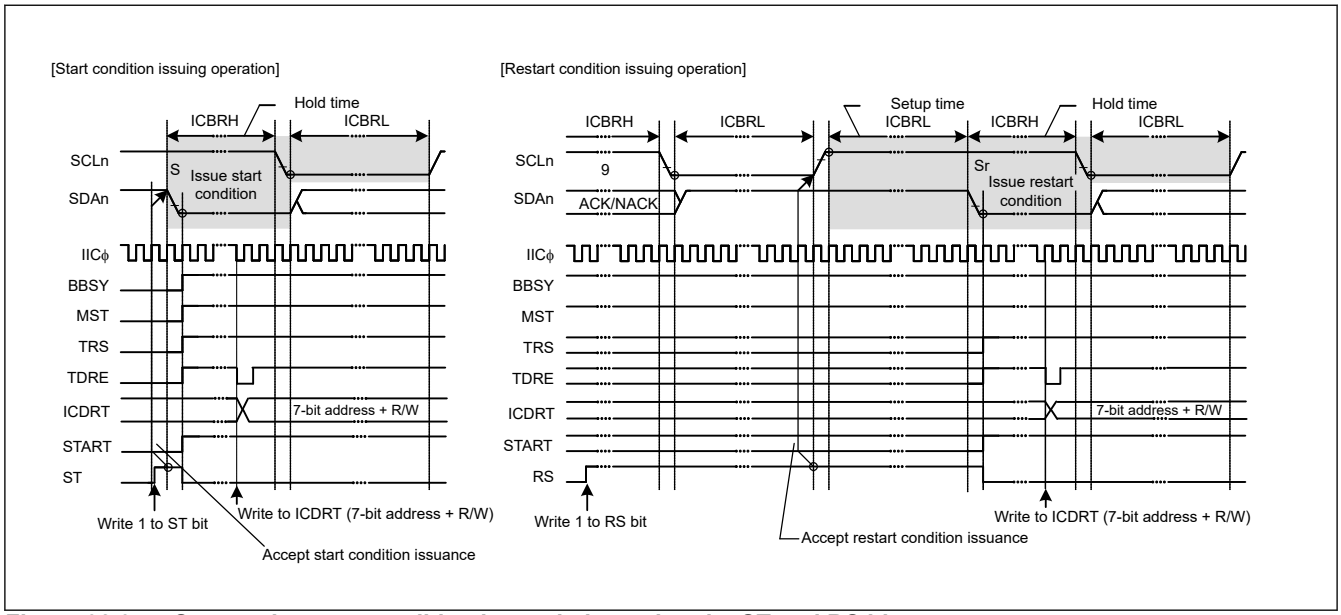


Figure 32.37 Start and restart condition issue timing using the ST and RS bits

Figure 32.38 shows the operation timing when a restart condition is issued after the master transmission.

To issue a restart condition after the master transmission:

1. Initialize the IIC using the details provided in section 32.3.2. Initial Settings.
2. Read the BBSY flag in IICR2 to check that the bus is free, and then set the ST bit in ICCR2 to 1 (start condition issuance request). On receiving the request, IIC issues a start condition. At the same time, the BBSY flag and the START flag in ICSR2 are automatically set to 1 and the ST bit automatically sets to 0. If the start condition is detected and the internal levels for the SDA output state and the levels on the SDAn line match while the ST bit is 1, IIC recognizes that a start condition is successfully issued as requested by the ST bit. The MST and TRS bits in ICCR2 automatically set to 1, placing IIC in master transmit mode. The TDRE flag in ICSR2 also automatically sets to 1 when the TRS bit is set to 1.
3. Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. After the transmit data is written to ICDRT, the TDRE flag automatically sets to 0, the data is transferred from ICDRT to ICDRS, and the TDRE flag again sets to 1. After the byte containing the slave address and R/W# bit has been transmitted, the value of the TRS bit is automatically updated to select master transmit or master receive mode according to the value of the transmitted R/W# bit. If the value of the R/W# bit is 0, IIC continues in master transmit mode. If the ICSR2.NACKF flag is 1 at this time, indicating that no slave device recognized the address or that there was an error in communications, write 1 to ICCR2.SP bit to issue a stop condition. To transmit data with an address in the 10-bit format, start by writing 11110b, the 2 upper bits of the slave address, and W to ICDRT as the first address transmission. Then, as the second address transmission, write the 8 lower bits of the slave address to ICDRT.
4. After confirming that the TDRE flag in ICSR2 is 1, write the data for transmission to the ICDRT register. IIC automatically holds the SCLn line low until the data for transmission is ready, and a restart or stop condition is issued.
5. After all bytes of data for transmission are written to the ICDRT register, wait until the value of the TEND flag in ICSR2 returns to 1. Then, after checking that the START flag in ICSR2 is 1, set the START flag in ICSR2 to 0.
6. Set the RS bit in ICCR2 to 1 (restart condition issuance request). Upon receiving the request, IIC issues a restart condition.
7. After checking that the START flag in ICSR2 is 1, write the value for transmission (the slave address and the R/W# bit) to ICDRT.

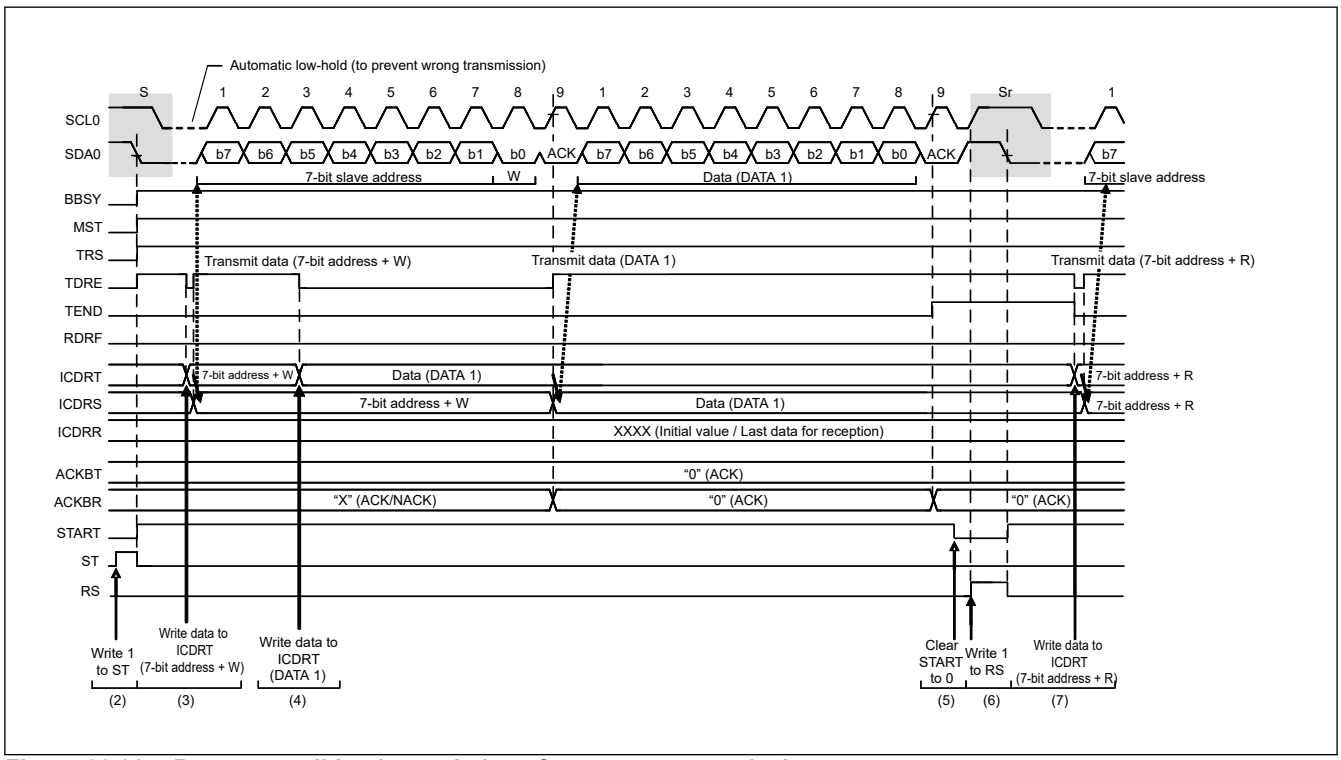


Figure 32.38 Restart condition issue timing after master transmission

32.10.3 Issuing a Stop Condition

IIC issues a stop condition when the SP bit in ICCR2 is set to 1. When the SP bit is set to 1, a stop condition request is made, and the IIC issues a stop condition when the BBSY flag in ICCR2 is 1 (bus busy state) and the MST bit in ICCR2 is 1 (master mode).

To issue a stop condition:

1. Drive the SDAn line low (high level to low level).
2. Ensure that the low-level period of the SCLn line set in ICBRL elapses.
3. Release the SCLn line (low level to high level).
4. Detect a high level on the SCLn line and ensure that the time set in ICBRH and the stop condition setup time elapse.
5. Release the SDAn line (low level to high level).
6. Ensure the time set in ICBRL and the bus free time elapse.
7. Clear the BBSY flag to 0 to release the bus mastership.

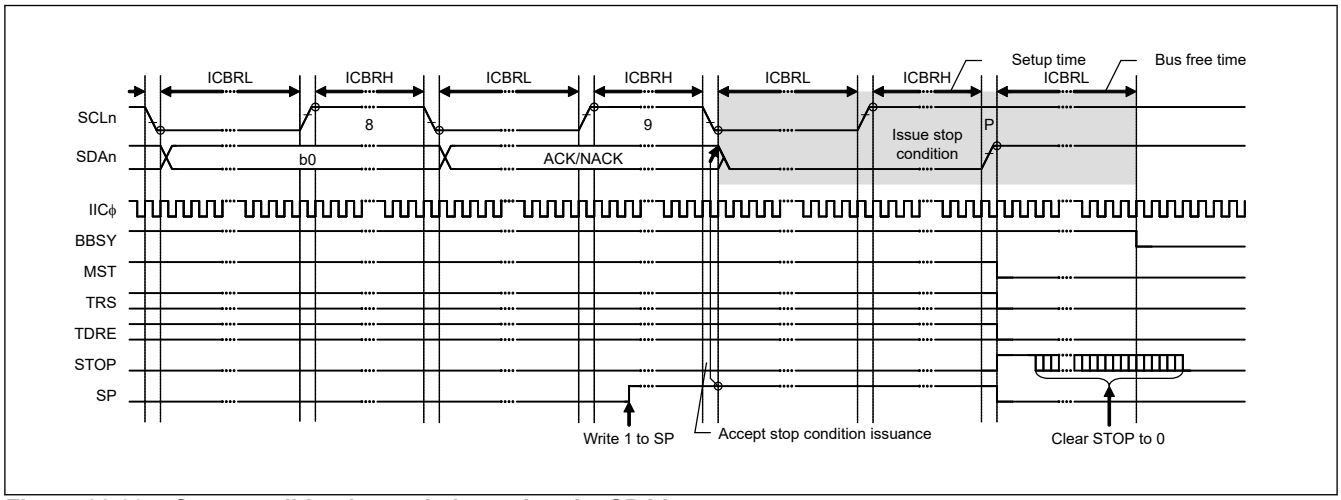


Figure 32.39 Stop condition issue timing using the SP bit

32.11 Bus Hanging

If the clock signals from the master and slave devices are out of synchronization because of noise or other factors, the I^2C bus might hang with a fixed level on the SCLn line or SDAn line.

To manage bus hanging, the IIC has the following:

- A timeout function to detect hanging by monitoring the SCLn line
- A function for the output of an extra SCL clock cycle to release the bus from a hung state because of clock signals being out of synchronization
- An IIC reset function
- An internal reset function

By checking the SCLO, SDAO, SCLI, and SDAI bits in ICCR1, it is possible to see whether the IIC or its communicating partner is placing the low level on the SCLn or SDAn lines.

32.11.1 Timeout Function

The timeout function can detect when the SCLn line is stuck longer than the predetermined time. IIC can detect an abnormal bus state by monitoring that the SCLn line is stuck low or high for a predetermined time.

The timeout function monitors the SCLn line state and counts the low-level or high-level period using the internal counter.

The timeout function resets the internal counter each time the SCLn line changes (rises or falls), but continues to count unless the SCLn line changes. If the internal counter overflows because no SCLn line changes, IIC can detect the timeout and report the bus hung state.

This timeout function is enabled when the ICFER.TMOE bit is 1. It detects a hung state when the SCLn line is stuck low or high during the following conditions:

- The bus is busy (ICCR2.BBSY flag is 1) in master mode (ICCR2.MST bit is 1)
- IIC slave address is detected (ICSR1 register is not 0x00) and the bus is busy (ICCR2.BBSY flag is 1) in slave mode (ICCR2.MST bit is 0)
- The bus is free (ICCR2.BBSY flag is 0) while a start condition is requested (ICCR2.ST bit is 1)

The internal counter of the timeout function uses the internal reference clock (IICΦ) set in the CKS[2:0] bits in ICMR1 as a count source. It functions as a 16-bit counter when long mode is selected (TMOS bit = 0 in ICMR2) or a 14-bit counter when short mode is selected (TMOS bit = 1).

The SCLn line level (low, high, or both levels) during which this counter is activated can be selected in the TMOH and TMOL bits in ICMR2. If both TMOL and TMOH bits are set to 0, the internal counter is disabled.

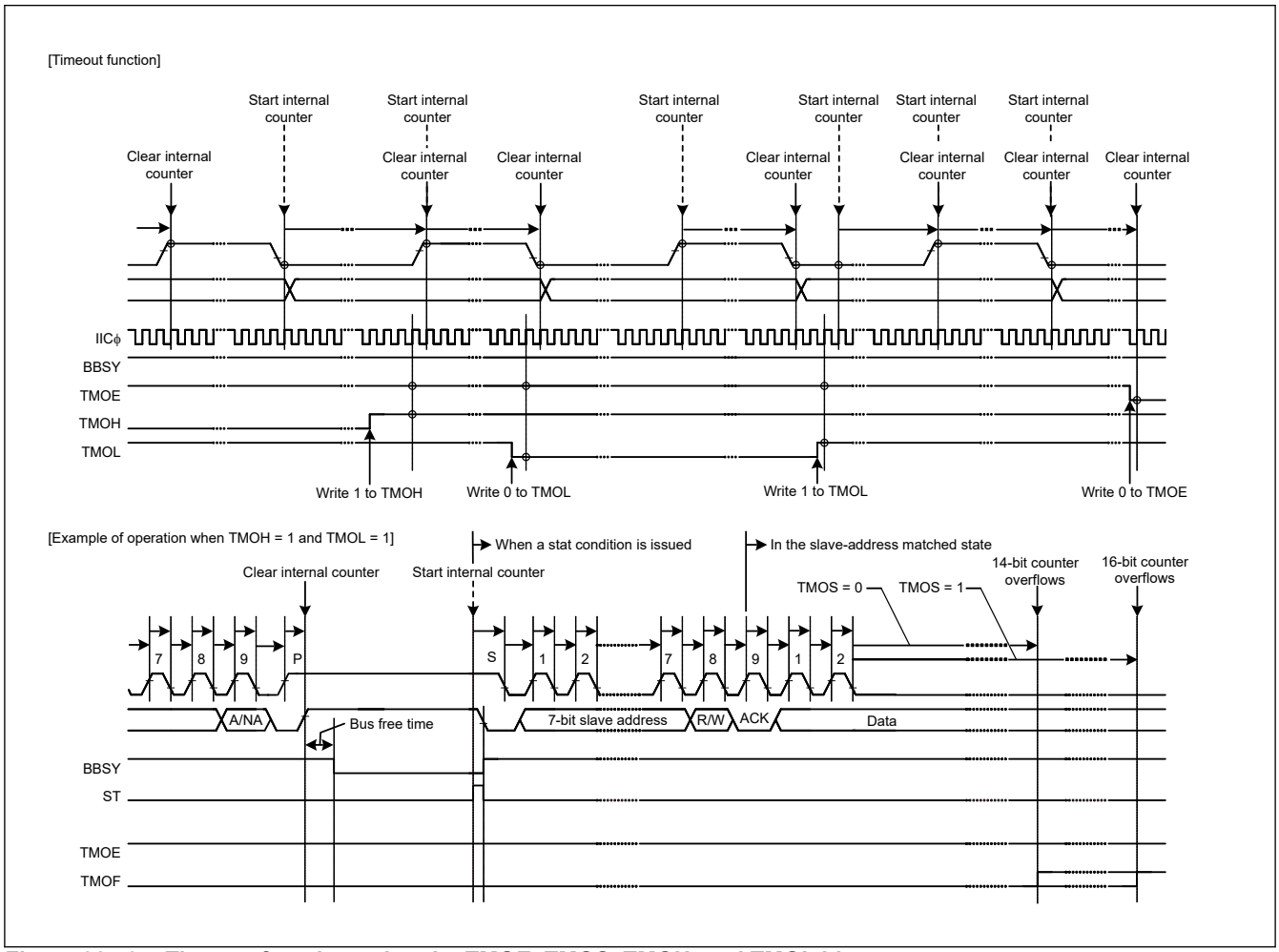


Figure 32.40 Timeout function using the TMOE, TMOS, TMOH, and TMOL bits

32.11.2 Extra SCL Clock Cycle Output Function

In master mode, this function outputs extra SCL clock cycles to release the SDAn line of the slave device from being held at the low level because the master is out of synchronization with the slave device.

This function is mainly used in master mode to release the SDAn line of the slave device from being fixed low by including extra cycles of SCL output from IIC. It uses single cycles of the SCL clock for a bus error where IIC cannot issue a stop condition because the slave device is holding the SDAn line at the low level. Do not use this function in normal situations.

Using it when communications are proceeding correctly leads to malfunctioning.

When the CLO bit in ICCR1 is set to 1 in master mode, a single cycle of the SCL clock at the transfer rate specified in the CKS[2:0] bits in ICMR1, and in the ICBRH and ICBRL registers, is output as an extra clock cycle. After output of this single cycle of the SCL clock, the CLO bit is automatically set to 0. More extra clock cycles can be output consecutively by writing 1 to the CLO bit through software after having read CLO = 0.

When the IIC module is in master mode and the slave device is holding the SDAn line at the low level because synchronization with the slave device is lost because of effects like noise, the output of a stop condition is not possible.

This function can be used to output extra cycles of SCL one by one to make the slave device release the SDAn line from being held at the low level, and recover the bus from an unusable state. Release of the SDAn line by the slave device can be monitored by reading the SDAI bit in ICCR1. After confirming the release of the SDAn line by the slave device, complete the communication by reissuing the stop condition.

Use this function with the MALE bit in ICFER set to 0 (master arbitration-lost detection disabled). If the MALE bit is set to 1 (enabled), arbitration is lost when the value of the SDAO bit in ICCR1 does not match the state of the SDAn line.

[Output conditions for using the CLO bit in ICCR1]

- When the bus is free (BBSY flag in ICCR2 = 0) or in master mode (MST bit = 1 and BBSY flag = 1 in ICCR2)
- When the communication device does not hold the SCLn line low.

Figure 32.41 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

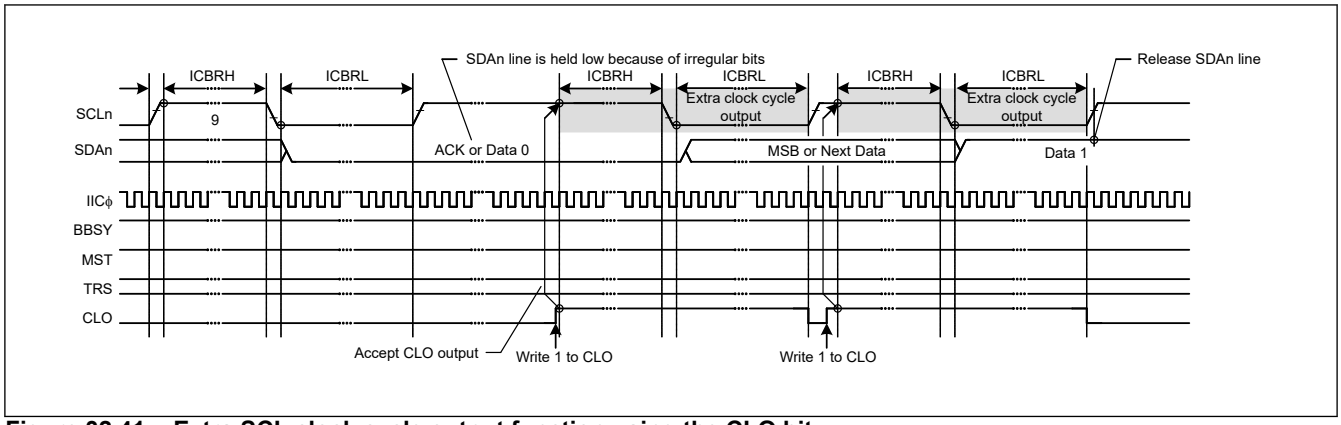


Figure 32.41 Extra SCL clock cycle output function using the CLO bit

32.11.3 IIC Reset and Internal Reset

The IIC module has two types of resets:

- IIC reset, which initializes all registers, including the BBSY flag in ICCR2
- Internal reset, which releases the IIC from the slave-address matched state and initializes the internal counter while saving other settings.

After issuing a reset, always set the IICRST bit in ICCR1 to 0.

Both types of resets are valid for release from bus-hung states, because both restore the output state of the SCLn and SDAn pins to the high-impedance state.

Issuing a reset during slave operation might lead to a loss of synchronization between the master device clock and the slave device clock, therefore, avoid this when possible. In addition, monitoring of the bus state, such as for the presence of a start condition, is not possible during an IIC reset (ICE and IICRST bits = 01b in ICCR1).

For a detailed description of the IIC and internal resets, see section 32.14. Resets, Registers, and Function States When Issuing Each Condition.

32.12 SMBus Operation

IIC supports data communication conforming to the SMBus Specification, version 2.0. To perform SMBus communication, set the SMBS bit in ICMR3 to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus standard, set the CKS[2:0] bits in ICMR1, ICBRH, and ICBRL. In addition, specify the values in the DLCS bit in ICMR2 and the

SDDL[2:0] bits in ICMR2 to meet the data hold time specification of 300 ns or more. When the IIC is used only as a slave device, the transfer rate setting is not required, but ICBRL must be set to a value longer than the data setup time (250 ns).

For the SMBus device default address (1100 001b), use one of the slave address registers L0 to L2 (SARL0, SARL1, and SARL2), and set the associated FS bit (7- or 10-bit address format select) in SARUy (y = 0 to 2) to 0 (7-bit address format).

When transmitting the UDID (Unique Device Identifier), set the SALE bit in ICFER to 1 to enable the slave arbitration-lost detection function.

32.12.1 SMBus Timeout Measurement

(1) Measuring slave device timeout

The following period (timeout interval: $T_{LOW:\ SEXT}$) must be measured for slave devices in SMBus communication:

- From start condition to stop condition

To measure timeout for slave devices, measure the period from start condition detection to stop condition detection with the internal timer using the IIC start condition detection interrupt (STIn) and stop condition detection interrupt (SPIn). The measured timeout period must be within the total clock low-level period [slave device] $T_{LOW:\ SEXT}$: 25 ms (maximum) of the SMBus standard.

If the time measured with the internal timer exceeds the clock low-level detection timeout $T_{TIMEOUT}$: 25 ms (minimum) of the SMBus standard, the slave device must release the bus by writing 1 to the IICRST bit in ICCR1 to issue an internal reset of the IIC. When an internal reset is issued, the IIC stops driving the bus for the SCLn and SDAn pins and makes the SCLn/SDAn pin output high-impedance, which releases the bus.

(2) Measuring master device timeout

The following periods (timeout interval: $T_{LOW:\ MEXT}$) must be measured for master devices in SMBus communication:

- From start condition to acknowledge bit
- Between acknowledge bits
- From acknowledge bit to stop condition

To measure timeout for master devices, measure these periods with the internal timer using the IIC start condition detection interrupt (STIn), stop condition detection interrupt (SPIn), transmit end interrupt (IICn_TEI), or receive data full interrupt (IICn_RXI). The measured timeout period must be within the total clock low-level extended period (master device) $T_{LOW:\ MEXT}$: 10 ms (maximum) of the SMBus standard, and the total of all $T_{LOW:\ MEXT}$ values from the start condition to stop condition must be within $T_{LOW:\ SEXT}$: 25 ms (maximum).

For the ACK receive timing (rising edge of the 9th SCL clock cycle), monitor the TEND flag in ICSR2 in master transmit mode (master transmitter) and the RDRF flag in ICSR2 in master receive mode (master receiver). Perform byte-wise transmit operations in master transmit mode, and hold the RDRFS bit in ICMR3 at 0 until the byte immediately before the reception of the final byte in master receive mode. While the RDRFS bit is 0, the RDRF flag is set to 1 on the rising edge of the 9th SCL clock cycle.

If the period measured with internal timer exceeds the total clock low-level extended period (master device), $T_{LOW:\ MEXT}$: 10 ms (maximum) of the SMBus standard or the total of measured periods exceeds the clock low-level detection timeout, $T_{TIMEOUT}$: 25 ms (minimum) of the SMBus standard, the master device must stop the transaction by issuing a stop condition. In master transmit mode, immediately stop the transmit operation (stop writing data to ICDRT).

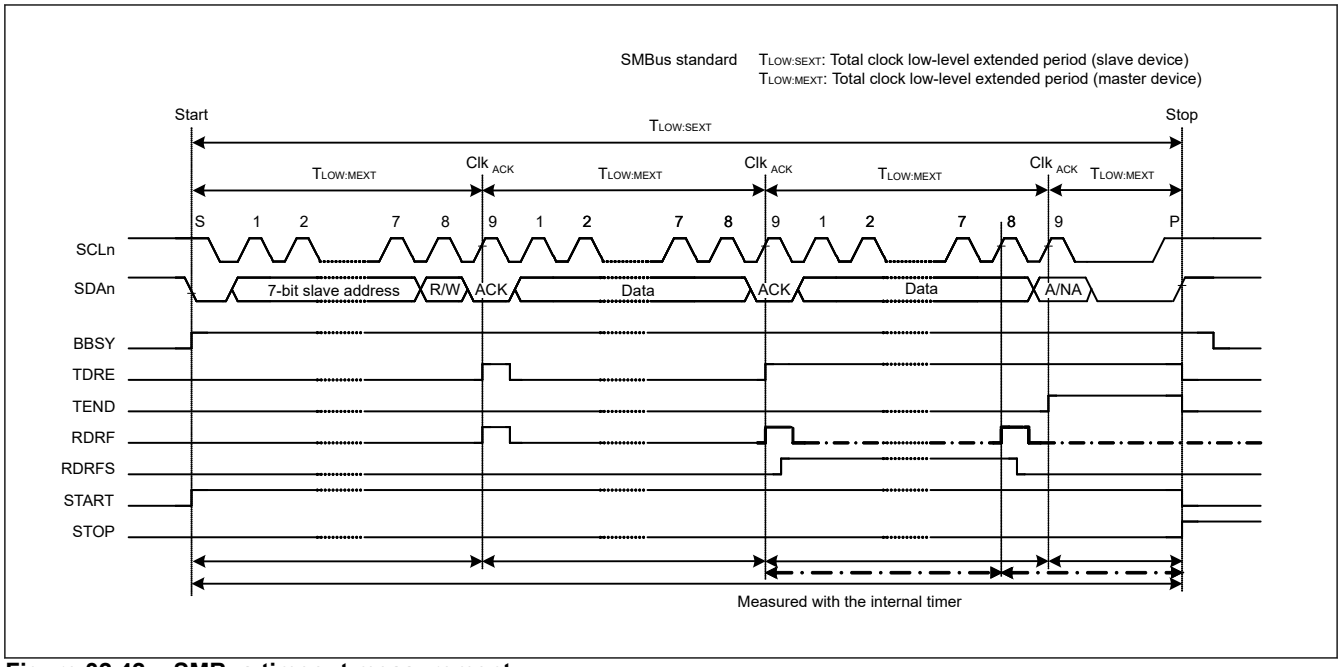


Figure 32.42 SMBus timeout measurement

32.12.2 Packet Error Code (PEC)

The LSI provides a CRC calculator, which enables transmission of a packet error code (PEC) or allows checking the received data in SMBus data communication for the IIC. For the CRC-generating polynomials of the CRC calculator, see section 36, CRC Operation Unit (CRC).

In master transmit mode, the PEC data can be generated by writing all transmit data to the CRC Data Input register (CRCDIR) in the CRC calculator.

In master receive mode, the PEC data can be checked by writing all receive data to CRCDIR in the CRC calculator and comparing the obtained value in the CRC Data Output Register (CRCDOR) with the received PEC data.

To send ACK or NACK based on the match or mismatch result when the final byte is received as a result of the PEC code check, set the RDRFS bit in ICMR3 to 1 before the rising edge of the 8th SCL clock cycle during reception of the final byte, and hold the SCLn line low on the falling edge of the 8th clock cycle.

32.12.3 SMBus Host Notification Protocol (Notify ARP Master Command)

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of its own slave address or to request its own slave address from the SMBus host.

For a product using the LSI to operate as an SMBus host or ARP master, the host address (0001 000b) sent from the slave device must be detected as a slave address, and so the IIC has a function for detecting the host address. To detect the host address as a slave address, set the SMBS bit in ICMR3 and the HOAE bit in ICSER to 1. Operation after the host address is detected is the same as normal slave operation.

32.13 Interrupt Sources

The IIC issues four types of interrupt requests:

- Transfer error or event occurrence (detection of arbitration-lost, NACK, timeout, start or restart condition, or stop condition)
- Receive data full
- Transmit data empty
- Transmit end

Table 32.9 lists details about the interrupt requests. The receive data full and transmit data empty interrupts can activate data transfer by the DMAC.

Table 32.9 Interrupt sources

Symbol	Interrupt source	Interrupt flag	DMAC activation	Interrupt condition
IICn_EEI*4	Transfer error or event generation	AL	Not possible	AL = 1, ALIE = 1
		NACKF		NACKF = 1, NAKIE = 1
		TMOF		TMOF = 1, TMOIE = 1
		START		START = 1, STIE = 1
		STOP		STOP = 1, SPIE = 1
IICn_RXI*2 *4	Receive data full	RDRF	Possible	RDRF = 1, RIE = 1
IICn_TXI*1 *4	Transmit data empty	TDRE	Possible	TDRE = 1, TIE = 1
IICn_TEI*3 *4	Transmit end	TEND	Not possible	TEND = 1, TEIE = 1

Note: There is a delay between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. When an interrupt flag is cleared or masked, read the relevant flag again to check whether clearing or masking is complete, and then return from interrupt handling. Not doing so creates the possibility of repeated processing of the same interrupt.
Note 1. Because IICn_TXI is an edge-detected interrupt, it does not require clearing. Additionally, the TDRE flag in ICSR2 (condition for IICn_TXI) is automatically set to 0 when transmit data is written to ICDRT or a stop condition is detected (STOP flag = 1 in ICSR2).
Note 2. Because IICn_RXI is an edge-detected interrupt, it does not require clearing. Additionally, the RDRF flag in ICSR2 (condition for IICn_RXI) automatically is set to 0 when data is read from ICDRR.
Note 3. When using the IICn_TEI interrupt, clear the TEND flag in ICSR2 in the IICn_TEI interrupt handling. The TEND flag in ICSR2 automatically is set to 0 when transmit data is written to ICDRT or a stop condition is detected (STOP flag = 1 in ICSR2).
Note 4. Channel number (n = 0 to 2).

Clear or mask each flag during interrupt handling.

32.14 Resets, Registers, and Function States When Issuing Each Condition

IIC provides chip reset, IIC reset, and internal reset functions. Table 32.10 lists the IIC resets, registers, and function states when issuing each condition.

Table 32.10 Resets, registers, and function states when issuing each condition (1 of 2)

Registers		Chip reset	IIC reset (ICE = 0, IICRST = 1)	Internal reset (ICE = 1, IICRST = 1)	Start or restart condition detection	Stop condition detection
ICCR1	ICE, IICRST	Reset	Saved	Saved	Saved	Saved
	SCLO, SDAO		Reset	Reset		
	Others			Saved		
ICCR2	BBSY	Reset	Reset	Saved	Set	Saved
	ST			Reset	Saved	Saved
	TRS, MST				Set or saved	Reset
	Others				Reset	Reset or Saved
ICMR1	BC[2:0]	Reset	Reset	Reset	Reset	Saved
	Others			Saved	Saved	
ICMR2		Reset	Reset	Saved	Saved	Saved
ICMR3		Reset	Reset	Saved	Saved	Saved
ICFER		Reset	Reset	Saved	Saved	Saved
ICSER		Reset	Reset	Saved	Saved	Saved
ICIER		Reset	Reset	Saved	Saved	Saved
ICSR1		Reset	Reset	Reset	Saved	Reset

Table 32.10 Resets, registers, and function states when issuing each condition (2 of 2)

Registers		Chip reset	IIC reset (ICE = 0, IICRST = 1)	Internal reset (ICE = 1, IICRST = 1)	Start or restart condition detection	Stop condition detection
ICSR2	TDRE, TEND	Reset	Reset	Reset	Saved	Reset
	START				Set	
	STOP				Saved	Set
	Others				Saved	Saved
SARL0, SARL1, SARL2, SARU0, SARU1, SARU2		Reset	Reset	Saved	Saved	Saved
ICBRH, ICBRL		Reset	Reset	Saved	Saved	Saved
ICDRT		Reset	Reset	Saved	Saved	Saved
ICDRR		Reset	Reset	Saved	Saved	Saved
ICDRS		Reset	Reset	Reset	Saved	Saved
Timeout function		Reset	Reset	Operating	Operating	Operating
Bus free time measurement		Reset	Reset	Operating	Operating	Operating

32.15 Usage Notes

32.15.1 Settings for the Module-Stop Function

IIC operation can be disabled or enabled using Module Stop Control Register G (MSTPCRG01). IIC is initially stopped after reset. Releasing the module-stop state enables access to the registers. For details, see section 8, Low-Power Consumption Function.

33. CAN-FD Interface (CANFD)

33.1 Overview

The LSI has a 2-channel CAN-FD module (CANFD) that complies with ISO 11898-1 (2015) Standards. CANFD transmits and receives both formats of messages, namely the standard identifier (11 bits) (identifier is hereafter referred to as ID) and extended ID (29 bits). Table 33.1 lists the specifications of the CANFD, Figure 33.1 shows a block diagram of the CANFD, and Table 33.2 lists the I/O pins.

Table 33.1 CANFD specifications (1 of 2)

Parameter	Description
Number of channels	Two channels
Protocol	CAN-FD ISO 11898-1 (2015)
Communication speed	• Classical CAN mode: 1 Mbps • CAN FD mode Nominal bit rate: max. 1 Mbps Data bit rate: max. 8 Mbps
Buffer	Total 192 buffers (when frame size is 76 bytes) • Individual buffers: 64 buffers (32 buffers × 2 channels) Transmit buffer: 32 buffers per channel Transmit queue: 4 queues per channel (shared with transmit buffer, up to 16 buffers allocatable) • Shared buffers: 128 buffers for all channels Receive buffer: 0 to 32 buffers Receive FIFO buffer: 8 FIFO buffers (up to 128 buffers allocatable to each) Transmit/receive FIFO buffer: 3 FIFO buffers per channel (up to 128 buffers allocatable to each)
Reception function	• Receives data frames and remote frames • Selects ID format (standard ID, extended ID, or both IDs) to be received • Sets interrupt enable/disable for each FIFO • Mirror function (reception of messages transmitted from the CAN node itself) • Timestamp function (to record message reception time as a 16-bit timer value)
Reception filter function	• Selects receive message according to 256 receive rules • Sets the number of receive rules (0 to 255) for each channel • Acceptance filter processing: Sets ID and mask for each receive rule • DLC filter processing: Enables DLC filter check for each acceptance rule
Receive message transfer function	• Routing function Transfers receive messages to arbitrary destinations (can be transferred to up to 8 buffers) Transfer destination: Receive buffer, receive FIFO buffer, and/or transmit/receive FIFO buffer • Label addition function Stores label information together with a message in a receive buffer and FIFO buffer
Transmission function	• Transmits data frames and remote frames • Selects ID format (standard ID, extended ID, or both IDs) to be transmitted • Sets interrupt enable/disable for each transmit buffer and transmit/receive FIFO buffer • Selects ID priority transmission or transmit buffer number priority transmission • Transmit request can be aborted (possible to confirm with a flag) • One-shot transmission function
Interval transmission function	Transmit message at configurable intervals (transmit mode or gateway mode of transmit/receive FIFO buffers)
Transmit ques function	Transmits all stored messages according to the ID priority
Transmit history function	Stores the history information of transmission-completed messages Adds timestamp (recording message transmission time as a 16-bit timer value) to the history information
Gateway function	Transmits a received message automatically

Table 33.1 CANFD specifications (2 of 2)

Parameter	Description
Bus off recovery mode selection	Selects the method for returning from bus-off state. ● ISO 11898-1 compliant ● Automatic entry to channel halt mode at bus-off entry ● Automatic entry to channel halt mode at bus-off end ● Transition to channel halt mode by program request ● Transition to the error-active state by program request (forcible return from the bus-off state)
Error status monitoring	● Monitors CAN protocol errors (stuff error, from error, ACK error, CRC error, bit error, ACK delimiter error, and bus dominant lock) ● Defects error status transitions (error warning, error passive, bus-off entry, and bus-off recovery) ● Reads the error counter ● Monitors DLC errors
Interrupt source	8 interrupt sources ● Global interrupts (2 sources) RX FIFO interrupt Global error interrupt ● Channel interrupts (3 sources/channel) Channel transmit interrupt Channel error interrupt Common RX FIFO interrupt 10 DMA requests ● 8 for RX FIFO ● 2 for the first common FIFO (1 source/channel)
CAN clock source	Selects the PCLKM or the PCLKCAN (40 MHz or 80 MHz)
Test function	Test function for user evaluation ● Listen-only mode ● Self-test mode 0 (external loopback) ● Self-test mode 1 (internal loopback) ● Restricted operation mode ● RAM test (read/write test) ● Inter-channel communication test (CRC error test enabled)
Low-power consumption function	Module-stop state can be set to reduce power consumption

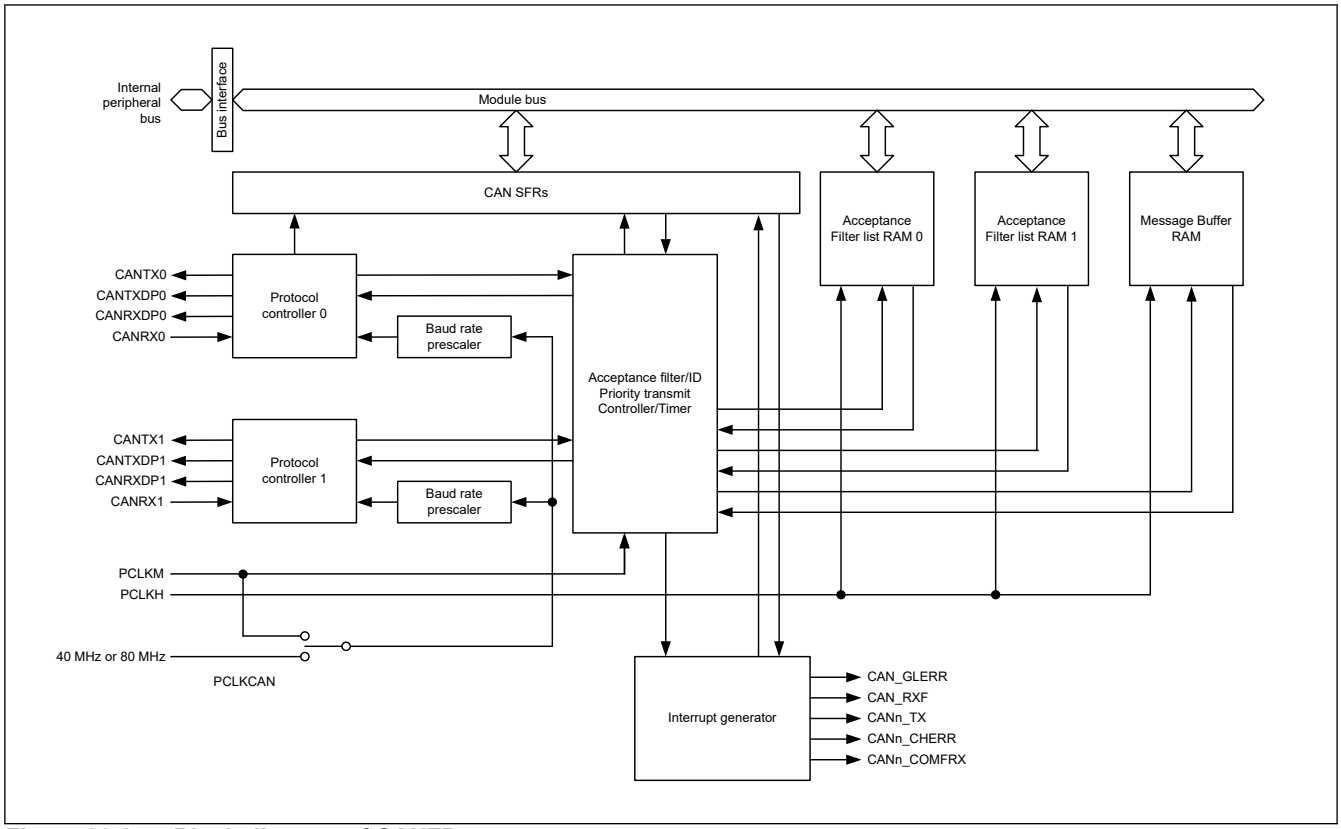


Figure 33.1 Block diagram of CANFD

Table 33.2 lists the input/output pins of the CANFD module.

Table 33.2 Pin configuration of CANFD

Channel	Pin name	I/O	Function
CAN0	CANRX0	Input	CAN0 receive data input
	CANTX0	Output	CAN0 transmit data output
	CANRXDP0	Output	CAN0 receive data phase output
	CANTXDP0	Output	CAN0 transmit data phase output
CAN1	CANRX1	Input	CAN1 receive data input
	CANTX1	Output	CAN1 transmit data output
	CANRXDP1	Output	CAN1 receive data phase output
	CANTXDP1	Output	CAN1 transmit data phase output

Table 33.3 CANFD interrupt sources

Name	Interrupt sources	CPU0 GIC request	DMAC activation
CAN_RXF	RX FIFO interrupt	Possible	Not possible
CAN_GLERR	Global error interrupt	Possible	Not possible
CAN_RF_DMAREQm	RX FIFO m DMA request	Not possible	Possible
CANn_TX	Channel n TX interrupt	Possible	Not possible
CANn_CHERR	Channel n CAN error interrupt	Possible	Not possible
CANn_COMFRX	Channel n Common RX FIFO or TXQ interrupt	Possible	Not possible
CANn_CF_DMAREQ	Channel n First common FIFO DMA request	Not possible	Possible

Note: m = 0 to 7, n = 0, 1

33.2 Register Descriptions

33.2.1 CFDCnNCFG : Channel n Nominal Bit Rate Configuration Register (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x0000 + 0x10 × n

Bit position:	31						25	24						17	16						10	9						0
Bit field:	NTSEG2[6:0]							NTSEG1[7:0]								NSJW[6:0]							NBRP[9:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
9:0	NBRP[9:0]	Nominal Bit Rate Prescaler Nominal bit rate prescaler division ratio setting	R/W
16:10	NSJW[6:0]	Nominal Bit Rate Resynchronization Jump Width Control 0x00: Setting prohibited 0x01: 2 Tq ⋮ 0x7E: 127 Tq 0x7F: 128 Tq	R/W
24:17	NTSEG1[7:0]	Nominal Bit Rate Time Segment 1 Control 0x00: Setting prohibited 0x01: 2 Tq 0x02: 3 Tq ⋮ 0xFE: 255 Tq 0xFF: 256 Tq	R/W
31:25	NTSEG2[6:0]	Nominal Bit Rate Time Segment 2 Control 0x00: Setting prohibited 0x01: 2 Tq ⋮ 0x7E: 127 Tq 0x7F: 128 Tq	R/W

The CFDCnNCFG register is used to configure the transmission/reception nominal bit rate parameters of the channels. Do not write this register in CH_OPERATION or CH_SLEEP mode. Set this register in CH_RESET or CH_HALT mode.

NBRP[9:0] bits (Nominal Bit Rate Prescaler)

The NBRP[9:0] bits are used to define the peripheral bus clock periods contained in a time quantum.

NSJW[6:0] bits (Nominal Bit Rate Resynchronization Jump Width Control)

The NSJW[6:0] bits set the synchronization jump width. A value from 1 to 128 time quanta can be set.

NTSEG1[7:0] bits (Nominal Bit Rate Time Segment 1 Control)

The NTSEG1[7:0] bits set the segment TSEG1 to compensate for edges on the CAN bus with a positive phase error. These bits contain the propagation segment. Configure a Tq value only between 2 and 256, inclusive. See section 33.4.1.2. CAN Bit Timing for more details.

NTSEG2[6:0] bits (Nominal Bit Rate Time Segment 2 Control)

The NTSEG2[6:0] bits set the segment TSEG2 to compensate for edges on the CAN bus with a negative phase error. Configure a Tq value only between 2 and 128, inclusive.

33.2.2 CFDCnCTR : Channel n Control Register (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x0004 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ROM	CRCT	—	—	—	CTMS[1:0]		CTME	ERRD	BOM[1:0]		—	TDCV FIE	SOCO IE	EOCO IE	TAIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ALIE	BLIE	OLIE	BORIE	BOEIE	EPIE	EWIE	BEIE	—	—	—	—	RTBO	CSLP R	CHMDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
1:0	CHMDC[1:0]	Mode Select 0 0: Channel Communication mode 0 1: Channel Reset mode 1 0: Channel Halt mode 1 1: Setting prohibited	R/W
2	CSLPR	Channel Stop Mode 0: Other than Channel Stop mode 1: Channel Stop mode	R/W
3	RTBO	Forcible Return from Bus-Off When this bit is set to 1, forcible return from the bus-off state is made. This bit is always read as 0.	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	BEIE	Bus Error Interrupt Enable 0: Bus error interrupt disabled 1: Bus error interrupt enabled	R/W
9	EWIE	Error Warning Interrupt Enable 0: Error warning interrupt disabled 1: Error warning interrupt enabled	R/W
10	EPIE	Error Passive Interrupt Enable 0: Error passive interrupt disabled 1: Error passive interrupt enabled	R/W
11	BOEIE	Bus-Off Entry Interrupt Enable 0: Bus-off entry interrupt disabled 1: Bus-off entry interrupt enabled	R/W
12	BORIE	Bus-Off Recovery Interrupt Enable 0: Bus-off recovery interrupt disabled 1: Bus-off recovery interrupt enabled	R/W
13	OLIE	Overload Interrupt Enable 0: Overload interrupt disabled 1: Overload interrupt enabled	R/W
14	BLIE	Bus Lock Interrupt Enable 0: Bus lock interrupt disabled 1: Bus lock interrupt enabled	R/W
15	ALIE	Arbitration Lost Interrupt Enable 0: Arbitration lost interrupt disabled 1: Arbitration lost interrupt enabled	R/W
16	TAIE	Transmission Abort Interrupt Enable 0: Transmission abort interrupt disabled 1: Transmission abort interrupt enabled	R/W

Bit	Symbol	Function	R/W
17	EOCOIE	Error Occurrence Counter Overflow Interrupt Enable 0: Error occurrence counter overflow interrupt disabled 1: Error occurrence counter overflow interrupt enabled	R/W
18	SOCOIE	Successful Occurrence Counter Overflow Interrupt Enable 0: Successful occurrence counter overflow interrupt disabled 1: Successful occurrence counter overflow interrupt enabled	R/W
19	TDCVFIE*1	Transceiver Delay Compensation Violation Interrupt Enable 0: Transceiver delay compensation violation interrupt disabled 1: Transceiver delay compensation violation interrupt enabled	R/W
20	—	This bit is read as 0. The write value should be 0.	R/W
22:21	BOM[1:0]	Bus-Off Recovery Mode Select 0 0: ISO11898-1 compliant 0 1: Entry to Channel Halt mode automatically at bus-off entry 1 0: Entry to Channel Halt mode automatically at bus-off end 1 1: Entry to Channel Halt mode (in bus-off state) by program request	R/W
23	ERRD	Error Display Mode Select 0: Error flags are displayed only for the first error information after bits [14:8] in the CFDCnERFL register are all cleared 1: Error flag for all error information are displayed	R/W
24	CTME	Communication Test Mode Enable 0: Communication test mode is disabled 1: Communication test mode is enabled	R/W
26:25	CTMS[1:0]	Communication Test Mode Select 0 0: Standard test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (External loopback mode) 1 1: Self-test mode 1 (Internal loopback mode)	R/W
29:27	—	These bits are read as 0. The write value should be 0.	R/W
30	CRCT	CRC Error Test Enable 0: The first bit of reception ID field is not inverted 1: The first bit of reception ID field is inverted	R/W
31	ROM*1	Restricted Operation Mode Enable 0: Restricted operation mode disabled 1: Restricted operation mode enabled	R/W

Note 1. These bits are not available in the classical CAN function.

The CFDCnCTR register controls the modes of the related channel. It is used to enable generation of interrupts if errors are detected on the CAN bus connected to this channel. It is also used to configure the channel in test mode.

CHMDC[1:0] bits (Mode Select)

The CHMDC[1:0] bits can be used to configure modes of the CAN channel.

CAN mode transitions are described in more details in section 33.3.3. Channel Modes.

Setting CHMDC[1:0] bits to 11b has no effect. When the CAN-FD module is in GL_HALT mode, these bits can only be set to 10b or 01b. These bits cannot be set in CH_SLEEP mode.

These bits can change automatically when transitioning to Halt mode by the CFDCnCTR.BOM settings.

If CPU write access to CFDCnCTR.CHMDC occurs at the same time when the CAN channel enters Halt mode (at the start of bus-off when CFDCnCTR.BOM[1:0] = 01b, or at the end of bus-off when CFDCnCTR.BOM[1:0] = 10b), then the CPU write access has the highest priority.

The CAN channel changes the value of CFDCnCTR.CHMDC within the Channel Control Registers for the specified cases only if the CFDCnCTR.CHMDC[1:0] value is 00b (Operation mode).

CSLPR bit (Channel Stop Mode)

When the CSLPR bit is 1, a Sleep mode request is generated for the corresponding CAN channel

When this bit is 0, a request to exit Sleep mode is generated for the related CAN-FD channel.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_SLEEP mode.

RTBO bit (Forcible Return from Bus-Off)

When the protocol controller of the CAN channel enters bus-off state, you can force a recovery from bus-off state by setting the RTBO bit in the Channel Control Register to 1.

The error state changes from bus-off state to integrating with a maximum delay of 1 CAN bit time.

When the RTBO bit is set to 1, the REC and TEC registers are initialized and the Bus-Off Status bit (Channel Bus-off Status, CFDCnSTS.BOSTS) is set to 0.

Registers other than the REC and TEC registers are not initialized by this command. Even if CFDCnCTR.BORIE is set, a bus-off recovery interrupt is not generated by this recovery from the bus-off state.

The RTBO bit cannot be set in CH_SLEEP mode. Setting this bit in any state other than bus-off state has no effect and the bit is cleared immediately. The read value is always 0.

Return from the Bus-Off command should be used only when CFDCnCTR.BOM[1:0] is set to 00b.

Only write to this bit when the related CAN-FD channel is in CH_OPERATION mode. This bit is automatically cleared when set by software.

BEIE bit (Bus Error Interrupt Enable)

When the BEIE and the CFDCnERFL.BEF bits are both 1, an error interrupt request is generated.

This bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

EWIE bit (Error Warning Interrupt Enable)

When the EWIE and the CFDCnERFL.EWF bits are both 1, an error interrupt request is generated.

The EWIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

EPIE bit (Error Passive Interrupt Enable)

An error interrupt request is generated when the EPIE bit and the CFDCnERFL.EPF are both 1.

The EPIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

BOEIE bit (Bus-Off Entry Interrupt Enable)

When the BOEIE and the CFDCnERFL.BOEF bits are both 1, an error interrupt request is generated.

The BOEIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

BORIE bit (Bus-Off Recovery Interrupt Enable)

When the BORIE and the CFDCnERFL.BORF bits are both 1, an error interrupt request is generated.

The BORIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

OLIE bit (Overload Interrupt Enable)

When the OLIE and the CFDCnERFL.OVLF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

BLIE bit (Bus Lock Interrupt Enable)

When the BLIE and the CFDCnERFL.BLF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

ALIE bit (Arbitration Lost Interrupt Enable)

When the ALIE and the CFDCnERFL.ALF bits are both 1, an error interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

TAIE bit (Transmission Abort Interrupt Enable)

When the TAIE bit is 1 and a transmission is successfully aborted from the transmit buffer, an interrupt request is generated.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

EOCOIE bit (Error Occurrence Counter Overflow Interrupt Enable)

When the EOCOIE bit is 1 and the CFDCnFDSTS.EOCO bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The EOCOIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

SOCOIE bit (Successful Occurrence Counter Overflow Interrupt Enable)

When the SOCOIE bit is 1 and the CFDCnFDSTS.SOCO bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The SOCOIE bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

TDCVFIE bit (Transceiver Delay Compensation Violation Interrupt Enable)

When the TDCVFIE bit is 1 and the CFDCnFDSTS.TDCVF bit belonging to the corresponding CAN channel is 1, an error interrupt request is generated.

The TDCVFIE bit cannot be set in CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET mode. Do not set this bit when in Classical-only mode.

Note: This bit is not available in the classical CAN function.

BOM[1:0] bits (Bus-Off Recovery Mode Select)

The BOM[1:0] bits control the timing of the recovery from Bus-Off mode of the CAN-FD Channel.

Do not write to these bits in CH_SLEEP mode. Only write to these bits when the related CAN-FD channel is in CH_RESET mode.

ERRD bit (Error Display Mode Select)

The ERRD bit controls the display mode of the error flag bits [14:8] in the Channel Error Flag Register (CFDCnERFL).

If the ERRD bit is 0 and more than one errors occur at the same time, the error flag bits are set for all the errors that occurred at the same time. No further errors are flagged until the error flag bits [14:8] are cleared.

Do not write to the ERRD bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

CTME bit (Communication Test Mode Enable)

The CTME bit enables the channel test modes.

Do not write to this bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_HALT mode.

CTMS[1:0] bits (Communication Test Mode Select)

The CTMS[1:0] bits are used to select the required test mode.

Do not write to these bits in CH_SLEEP or CH_RESET mode. Only write to these bits when the related CAN-FD channel is in CH_HALT mode.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

CRCT bit (CRC Error Test Enable)

The CRCT bit checks the internal CRC generator logic of the protocol controller.

This bit inverts the first bit (ID bit) of the CAN message data stream that is being received, so that the internal generated CRC result does not match the received CRC value of the frame. Refer to the bit stuffing rule when using this feature for the possibility of receiving a stuff error (due to the inversion) rather than a CRC error.

The internal generated CRC value is always observed in the following registers:

- CFDCnERFL.CRCREG (classical CAN frames)
- CFDCnFDCRC.CRCREG (CAN-FD frames)

Some limitations exist when using this bit:

- It is not possible to use this feature with CAN nodes connected to the device externally, only with nodes connected to the internal CAN bus communication can be used
- One CAN node can send a reference message and the receiver node can invert one bit of the incoming bit stream

Note: The transmitter and receiver modes share the same CRC generator, therefore it is not necessary to consider the modes separately when testing this limitation.

The CRC Error Test mode is enabled if the CRCT (new control signal that inverts the first bit of the bit stream) and CTME bits are both 1.

Do not write to the CRCT bit in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_HALT mode.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

ROM bit (Restricted Operation Mode Enable)

When the ROM and CTME bits are both 1, the restricted operation mode is enabled. This mode should only be used in basic test mode (CFDCnCTR.CTMS[1:0] = 00b).

The ROM bit cannot be set in CH_SLEEP mode. Only write to this bit when the related CAN-FD channel is in CH_HALT mode.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. Do not set this bit when in Classical-only mode.

Note: This bit is not available in the classical CAN function.

33.2.3 CFDCnSTS : Channel n Status Register (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x0008 + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TEC[7:0]								REC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	ESIF	COMSTS	RECSTS	TRMSTS	BOSTS	EPSTS	CSLPSTS	CHLTSTS	CRSTSTS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
0	CRSTSTS	Channel Reset Status Flag 0: Not in Channel Reset mode 1: In Channel Reset mode	R

Bit	Symbol	Function	R/W
1	CHLTSTS	Channel Halt Status Flag 0: Not in Channel Halt mode 1: In Channel Halt mode	R
2	CSLPSTS	Channel Stop Status Flag 0: Not in Channel Sleep mode 1: In Channel Sleep mode	R
3	EPSTS	Error Passive Status Flag 0: Not in error passive state 1: In error passive state	R
4	BOSTS	Bus-Off Status Flag 0: Not in bus-off state 1: In bus-off state	R
5	TRMSTS	Transmit Status Flag 0: Bus idle or in reception 1: In transmission or bus off state	R
6	RECSTS	Receive Status Flag 0: Bus idle, in transmission or bus off state 1: In reception	R
7	COMSTS	Communication Status Flag 0: Communication is not ready 1: Communication is ready	R
8	ESIF*1	Error State Indication Flag 0: No CANFD message whose ESI bit is recessive has been received 1: At least one CANFD message whose ESI bit is recessive has been received	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
23:16	REC[7:0]	Reception Error Count The receive error counter (REC) can be read.	R
31:24	TEC[7:0]	Transmission Error Count The transmit error counter (TEC) can be read.	R

Note 1. This bit is not available in the classical CAN function.

The CFDCnSTS register shows the mode, error, and transmission/reception status of the related channel together with its reception and transmission error count values.

CRSTSTS bit (Channel Reset Status Flag)

The CRSTSTS bit indicates whether the related CAN channel is in Reset mode.

This bit is set automatically when the related CAN channel enters Channel Reset mode. When the mode is changed from Reset mode to Sleep mode, the CRSTSTS bit remains 1.

This bit is cleared automatically when the related CAN channel exits the Channel Reset mode, except when changing to Sleep mode.

CHLTSTS bit (Channel Halt Status Flag)

The CHLTSTS bit indicates whether the related CAN channel is in Halt mode.

This bit is set automatically when the related CAN module enters Halt mode, and is cleared automatically when the related CAN module exits Halt mode.

CSLPSTS bit (Channel Stop Status Flag)

The CSLPSTS bit indicates whether the related CAN channel is in Sleep mode.

This bit is set automatically when the related CAN-FD channel enters Sleep mode, and is cleared automatically when the related CAN-FD channel exits Sleep mode.

EPSTS bit (Error Passive Status Flag)

The EPSTS bit indicates whether the related CAN-FD channel has entered the error passive state.

This bit is set automatically when the value of the CAN Transmission or Reception Counter Register exceeds the value of 0x7F.

This bit is cleared automatically when the related CAN-FD channel exits the error passive state or enters Reset mode.

BOSTS bit (Bus-Off Status Flag)

The BOSTS bit indicates whether the related CAN-FD channel has entered the error bus-off state.

This bit is set automatically when the value of the related CAN Transmission Error Count Register exceeds 0xFF and the related CAN-FD channel is in the bus-off state (CAN Transmission Error Count Register > 0xFF).

This bit is cleared automatically when the related CAN-FD channel exits bus-off state.

TRMSTS bit (Transmit Status Flag)

The TRMSTS bit indicates whether the related CAN-FD channel is transmitting a message.

This bit is set automatically when the related CAN-FD channel is operating as a transmitter node or is in the bus-off state.

This bit is cleared automatically when the related CAN-FD channel is in the bus-idle state or starts operating as a receiver node.

RECSTS bit (Receive Status Flag)

The RECSTS bit indicates whether the related CAN-FD channel is receiving a message.

This bit is set automatically when the related CAN-FD channel is operating as a receiver node.

This bit is cleared automatically when the related CAN-FD channel is in the bus-idle state or starts operating as a transmitter node.

COMSTS bit (Communication Status Flag)

The COMSTS bit indicates whether the related CAN-FD channel is ready for communication.

This bit is set automatically when the related CAN-FD channel is ready to perform communication following the detection of 11 consecutive recessive bits after exiting the Reset or Halt mode.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

Note: This bit is 1 during bus-off state.

ESIF bit (Error State Indication Flag)

The ESIF bit is set when the ESI bit is sampled recessive for a reception CAN message without any error. When in Loopback or Mirror mode, the self-transmitted messages are considered reception messages.

If a set from the CAN-FD channel occurs simultaneously with a clear by a write access, then the bit is set.

This bit is cleared by writing 0 to it. This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

Note: This bit is not available in the classical CAN function.

REC[7:0] bits (Reception Error Count)

The REC[7:0] bits increment or decrement the counter value according to error status of the CAN-FD channel during reception, and display the value of the REC error counter.

The value in bus-off state is indeterminate.

These bits are cleared automatically when the CAN-FD module enters GL_RESET or the CAN-FD channel is in CH_RESET mode.

TEC[7:0] bits (Transmission Error Count)

The TEC[7:0] bits increment or decrement the counter value according to error status of the CAN-FD channel during transmission, and display the value of the TEC error counter.

Only write to these bits when in test mode and CAN-FD channel is in CH_HALT mode.

Do not read data from this register while CFDCnCTR.TRWE is set.

These bits are cleared automatically when CAN-FD module is in GL_RESET or CAN-FD channel is in CH_RESET mode.

33.2.4 CFDCnERFL : Channel n Error Flag Register (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x000C + 0x10 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	CRCREG[14:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	ADERR	B0ERR	B1ERR	CERR	AERR	FERR	SERR	ALF	BLF	OVLF	BORF	BOEF	EPF	EWF	BEF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BEF	Bus Error Flag 0: Channel bus error not detected 1: Channel bus error detected	R/W*1
1	EWF	Error Warning Flag 0: Channel error warning not detected 1: Channel error warning detected	R/W*1
2	EPF	Error Passive Flag 0: Channel error passive not detected 1: Channel error passive detected	R/W*1
3	BOEF	Bus-Off Entry Flag 0: Channel bus-off entry not detected 1: Channel bus-off entry detected	R/W*1
4	BORF	Bus-Off Recovery Flag 0: Channel bus-off recovery not detected 1: Channel bus-off recovery detected	R/W*1
5	OVLF	Overload Flag 0: Channel overload not detected 1: Channel overload detected	R/W*1
6	BLF	Bus Lock Flag 0: Channel bus lock not detected 1: Channel bus lock detected	R/W*1
7	ALF	Arbitration Lost Flag 0: Channel arbitration-lost not detected 1: Channel arbitration-lost detected	R/W*1
8	SERR	Stuff Error Flag 0: Channel stuff error not detected 1: Channel stuff error detected	R/W*1
9	FERR	Form Error Flag 0: Channel form error not detected 1: Channel form error detected	R/W*1
10	AERR	Acknowledge Error Flag 0: Channel acknowledge error not detected 1: Channel acknowledge error detected	R/W*1
11	CERR	CRC Error Flag 0: Channel CRC error not detected 1: Channel CRC error detected	R/W*1

Bit	Symbol	Function	R/W
12	B1ERR	Recessive Bit Error Flag 0: Channel recessive bit error not detected 1: Channel recessive bit error detected	R/W*1
13	B0ERR	Dominant Bit Error Flag 0: Channel dominant bit error not detected 1: Channel dominant bit error detected	R/W*1
14	ADERR	Acknowledge Delimiter Error Flag 0: Channel acknowledge delimiter error not detected 1: Channel acknowledge delimiter error detected	R/W*1
15	—	This bit is read as 0. The write value should be 0.	R/W
30:16	CRCREG[14:0]	CRC Calculation Data (CRC length: 15 bits) These bits show the CRC value calculated based on the transmit message or receive message.	R
31	—	This bit is read as 0. The write value should be 0.	R/W

Note 1. To clear each flag of this register, software must write 0 to the corresponding bit. These flags cannot be set to 1 by software.

The CFDCnERFL register shows the status of various error conditions detectable regardless of the setting of the related CAN Channel Error Interrupt Enable Register. It also shows the status of the various bus errors detectable by the CAN channel. Refer to the CAN specification (ISO 11898-1) for a description of error occurrence conditions.

Only write to this register when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

BEF bit (Bus Error Flag)

The BEF bit indicates a detection of a CAN channel bus error state, flagged by bits [14:8] in this register.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

This bit is set automatically when a bus error is detected, and is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

EWF bit (Error Warning Flag)

The EWF bit indicates whether an error warning condition has been detected for the CAN channel.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

This bit is set automatically when either TEC or REC exceeds 0x5F.

The setting of this bit only occurs when the TEC or REC initially exceeds 0x5F. Therefore, if the TEC or REC remains > 0x5F and the EWF bit is cleared by software, it is not set again until both the TEC and REC go below 0x60 and either TEC or REC crosses over again from a value 0x5F to a value > 0x5F.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

EPF bit (Error Passive Flag)

The EPF bit indicates a detection of a CAN channel error passive state.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

This bit is set automatically when the CAN error state becomes error passive state.

The setting of this bit only occurs when the TEC or REC initially exceeds 0x7F. Therefore, if the TEC or REC remains > 0x7F and the bit is cleared by software, it is not set again until both the TEC and REC go below 0x80 and either TEC or REC crosses over again from a value ≤ 0x7F to a value > 0x7F.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

BOEF bit (Bus-Off Entry Flag)

The BOEF bit indicates a detection of a CAN channel bus-off entry state.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

This bit is set automatically when the CAN error state enters the bus-off state.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If a set condition occurs simultaneously with a clear condition, then the bit is set.

BORF bit (Bus-Off Recovery Flag)

The BORF bit indicates a detection of a CAN channel bus-off recovery state.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

This bit is set automatically if CAN channel recovers from bus-off state in the following conditions:

- When CFDCnCTR.BOM[1:0] is 00b and normal recovery (11 consecutive recessive bits × 128 times detected) occurs
- When CFDCnCTR.BOM[1:0] is 10b and normal recovery (11 consecutive recessive bits × 128 times detected) occurs
- When CFDCnCTR.BOM[1:0] is 11b and normal recovery (11 consecutive recessive bits × 128 times detected) occurs.

The bit is not set if CAN channel recovers from bus-off state in the following conditions:

- When CAN Reset mode is requested
- When CFDCnCTR.RTBO is set to 1 (the CAN channel returns to error active)
- When CFDCnCTR.BOM[1:0] is 01b
- When CFDCnCTR.BOM[1:0] is 11b and a halt request is asserted before the CAN channel reaches the end of the bus-off state.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If a set condition occurs simultaneously with a clear condition, the flag is set.

OVLF bit (Overload Flag)

The OVLF flag indicates a detection of a CAN channel overload state.

The OVLF bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

This bit is set automatically when an overload condition is detected. If a set condition occurs simultaneously with a clear condition, then the bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

BLF bit (Bus Lock Flag)

The BLF bit indicates a detection of a CAN channel bus lock condition.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

This bit is set automatically when 32 consecutive dominant bits are detected on the CAN bus while the CAN channel is in Operation mode.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

It is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

ALF bit (Arbitration Lost Flag)

The ALF bit indicates a detection of a CAN channel bus arbitration-lost condition.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

The bit is set automatically when an arbitration-lost condition is detected on the CAN bus while the CAN channel is in Operation mode.

If a set condition occurs simultaneously with a clear condition, then the bit is set. It is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

SERR bit (Stuff Error Flag)

The SERR bit indicates a detection of a CAN stuff error.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a stuff error is detected. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 0000000b.

FERR bit (Form Error Flag)

The FERR bit indicates a detection of a CAN form error.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a form error is detected. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 0000000b.

AERR bit (Acknowledge Error Flag)

The AERR bit indicates a detection of a CAN acknowledge error.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when an acknowledge error is detected. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 0000000b.

CERR bit (CRC Error Flag)

The CERR bit indicates a detection of a CAN CRC error.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a CRC error is detected. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 0000000b.

B1ERR bit (Recessive Bit Error Flag)

The B1ERR bit indicates a detection of a recessive bit error.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a recessive bit error (expected recessive bit, sampled as dominant bit) is detected. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 0000001b.

B0ERR bit (Dominant Bit Error Flag)

The B0ERR bit indicates a detection of a dominant bit error.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a dominant bit error (expected dominant bit, sampled as recessive bit) is detected. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 0000000b.

ADERR bit (Acknowledge Delimiter Error Flag)

The ADERR bit indicates a detection of an acknowledge delimiter bit error.

This bit is cleared by writing 0 to it, and can only be set by CAN-FD module logic. Writing 1 has no effect.

To clear this bit, use the following sequence:

1. Clear the corresponding flag bit.
2. Read if the flag bit is cleared.
3. If yes, continue, else go back to step 1.

This bit is set automatically when a form error is detected during the acknowledge delimiter state of frame transmission. If CFDCnCTR.ERRD bit is 1 and if the set and clear conditions occur at the same time for this bit, then this bit is set.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode. If CFDCnCTR.ERRD bit is 0 and the set and clear conditions occur at the same time for this bit, then it is cleared if a bit at the error flag bits [14:8] in the CFDCnERFL register is already set. Otherwise, this bit is set if the error flag bits [14:8] in the CFDCnERFL register is 0000000b.

CRCREG[14:0] bits (CRC Calculation Data (CRC length: 15 bits))

The CRCREG[14:0] bits read the calculated CRC value when CFDCnCTR.CTME bit is 1 for the channel.

If CFDCnCTR.CTME bit is 0, then these bits are always read as 0.

These bits show the CAN2.0 CRC value calculated by the CAN-FD channel logic when the CTME bit is enabled.

The CFDCnERFL.CRCREG value is updated in the first bit of the CRC field of the CAN frame (reception and transmission).

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

33.2.5 CFDCnDCFG : Channel n Data Bit Rate Configuration Register (n = 0, 1)

This register is not available in the classical CAN function.

Base address: CANFD = 0x8002_0000

Offset address: 0x1400 + 0x20 × n

Bit position:	31				27				24				19				16				12				8	7						0
Bit field:	—	—	—	—	DSJW[3:0]				—	—	—	—	DTSEG2[3:0]				—	—	—	DTSEG1[4:0]					DBRP[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	DBRP[7:0]	Data Bit Rate Prescaler Division Ratio Setting When the set value = P (0 to 255), the data bit rate prescaler divides fCAN by (P + 1).	R/W
12:8	DTSEG1[4:0]	Data Bit Rate Time Segment 1 Control 0x00: Setting prohibited 0x01: 2 Tq 0x02: 3 Tq 0x03: 4 Tq ⋮ 0x1E: 31 Tq 0x1F: 32 Tq	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
19:16	DTSEG2[3:0]	Data Bit Rate Time Segment 2 Control 0x0: Setting prohibited 0x1: 2 Tq 0x2: 3 Tq 0x3: 4 Tq ⋮ 0xE: 15 Tq 0xF: 16 Tq	R/W
23:20	—	These bits are read as 0. The write value should be 0.	R/W
27:24	DSJW[3:0]	Data Bit Rate Resynchronization Jump Width Control 0x0: 1 Tq 0x1: 2 Tq ⋮ 0xE: 15 Tq 0xF: 16 Tq	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

The CFDCnDCFG register configures the transmission/reception data bit rate parameters of the channels.

The channel of Classical-only mode does not perform configuration of this register.

Do not write to this register in CH_OPERATION or CH_SLEEP mode. Only write to this register in CH_RESET or CH_HALT mode.

DBRP[7:0] bits (Data Bit Rate Prescaler Division Ratio Setting)

The DBRP[7:0] bits define the peripheral bus clock periods contained in a time quantum.

DTSEG1[4:0] bits (Data Bit Rate Time Segment 1 Control)

The DTSEG1[4:0] bits set the segment TSEG1 to compensate for edges on the CAN bus with a positive phase error. A value from 2 to 32 time quanta can be set.

The DTSEG1[4:0] bits are also used to set the propagation segment.

Do not write any other value to these bits. See section 33.4.1.2. CAN Bit Timing for more details.

DTSEG2[3:0] bits (Data Bit Rate Time Segment 2 Control)

The DTSEG2[3:0] bits set the segment TSEG2 to compensate for edges on the CAN bus with a negative phase error. A value from 2 to 16 time quanta can be set.

Do not write any other value to these bits.

DSJW[3:0] bits (Data Bit Rate Resynchronization Jump Width Control)

The DSJW[3:0] bits set the synchronization jump width. A value from 1 to 16 time quanta can be set.

33.2.6 CFDCnFDCFG : Channel n CAN-FD Configuration Register (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x1404 + 0x20 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CFDTE	CLOE	REFE	FDOE	—	GWBRS	GWFDF	GWEN	TDCO[7:0]							
Value after reset:	0	0*1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ESIC	TDCE	TDCOC	—	—	—	—	—	EOCCFG[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	EOCCFG[2:0]	Error Occurrence Counter Configuration 0 0 0: All transmit messages and receive messages 0 0 1: All transmit messages 0 1 0: All receive messages 0 1 1: Setting prohibited 1 0 0: Only data phase of transmitted or received CANFD message 1 0 1: Only data phase of transmitted CANFD message 1 1 0: Only data phase of received CANFD message 1 1 1: Setting prohibited	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	TDCOC*2	Transmitter Delay Compensation Offset Configuration 0: Measurement and offset 1: Offset-only	R/W
9	TDCE*2	Transceiver Delay Compensation Enable 0: Transceiver delay compensation disabled 1: Transceiver delay compensation enabled	R/W
10	ESIC*2	Error State Indication Configuration 0: The ESI bit in the frame represents the error state of the node itself 1: The ESI bit in the frame represents the error state of the message buffer if the node itself is not in error passive. If the node is in error passive, then the ESI bit is driven by the node itself.	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
23:16	TDCO[7:0]*2	Transceiver Delay Compensation Offset These bits are set to the transmitter delay compensation offset value.	R/W

Bit	Symbol	Function	R/W
24	GWEN*2	CAN2.0, CAN-FD Multi Gateway Enable 0: Multi gateway disabled 1: Multi gateway enabled	R/W
25	GWFDF*2	Gateway FDF Configuration Bit 0: Gateway frame is transmitted as Classical CAN frame 1: Gateway frame is transmitted as CAN-FD frame	R/W
26	GWBRS*2	Gateway BRS Configuration Bit 0: Gateway frame is transmitted with BRS = 0 1: Gateway frame is transmitted with BRS = 1	R/W
27	—	This bit is read as 0. The write value should be 0.	R/W
28	FDOE*2	FD-Only Enable 0: FD-only mode disabled 1: FD-only mode enabled	R/W
29	REFE	RX Edge Filter Enable 0: RX edge filter disabled 1: RX edge filter enabled	R/W
30	CLOE*2	Classical CAN-Only Enable 0: Classical-only mode disabled 1: Classical-only mode enabled	R/W
31	CFDTE	CAN-FD Frame Distinction Enable 0: CAN-FD frame distinction disabled 1: CAN-FD frame distinction enabled	R/W

Note 1. This bit is set to 1 for RZ/T2M device with CANFD disabled.
Note 2. These bits are not available in the classical CAN function.

The CFDCnFDCFG register configures which communication direction (transmitter/receiver) errors are counted.

EOCCFG[2:0] bits (Error Occurrence Counter Configuration)

The EOCCFG[2:0] bits select which type of CAN frame configuration and direction, including protocol errors are counted.

Do not write to these bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

TDCOC bit (Transmitter Delay Compensation Offset Configuration)

The TDCOC bit selects which offset is used when defining the position of the secondary sample point (SSP) for the CAN-FD channel. If the bit is set to 0, the position of the SSP is the measured transceiver delay plus the fixed offset. If the bit is 1, the position of the SSP is defined only by the offset.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical-only mode.

This bit is not available in the classical CAN function.

TDCE bit (Transceiver Delay Compensation Enable)

The TDCE bit enables the transceiver delay compensation for the CAN-FD channel.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical-only mode.

This bit is not available in the classical CAN function.

ESIC bit (Error State Indication Configuration)

Bus controllers that are used as CAN-to-CAN gateway support that in every forwarded CAN-FD message. The ESI flag does not change to reflect the status of the gateway, bridge, or router but instead the flag is sent as it was in the original message.

The ESIC bit controls the transmission of either the ESI flag information or the message of ESI flag information (CFDCFFDCSTSn.CFESI or CFDTMFDCTRn.TMESI).

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode. Do not set this bit when in Classical-only mode.

This bit is not available in the classical CAN function.

TDCO[7:0] bits (Transceiver Delay Compensation Offset)

The TDCO[7:0] bits set the secondary sample point offset. How this value is used, depends on the CFDCnFDCFG.TDCOC setting.

If CFDCnFDCFG.TDCOC = 0, the transceiver delay compensation result is equal to the Trv_Delay (measured delay) + the value in CFDCnFDCFG.TDCO, rounded down to the nearest integer number of time quanta. Otherwise, the result is equal to the value in CFDCnFDCFG.TDCO. See section 33.4.1.5. Transmitter Delay Compensation for details on how CFDCnFDCFG.TDCO is used.

The actual offset value is interpreted as TDCO + 1. For example, if 4 is set in TDCO, the offset is 5 clock cycles. Clock cycle is 1 cycle of CAN channel DLL clock.

Do not write to the TDCO[7:0] bits in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode. Do not set to this bit when in Classical-only mode.

This bit is not available in the classical CAN function.

GWEN bit (CAN2.0, CAN-FD Multi Gateway Enable)

When the GWEN bit is enabled, a multi gateway is enabled. Message received on one node can be routed to another node using the COM FIFO when they are configured as gateway FIFO (CFDCFCCn.CFM[1:0] = 10b). Furthermore, when TX Queue is set as Gateway mode, the message received on one node can be stored in TX Queue and can be sent to another node.

The FDF and BRS bits of the routed message can be changed by the configuration value of the CFDCnFDCFG.GWFDF and CFDCnFDCFG.GWBRS bits. By this, the transmitted value of these bits can be replaced.

Example:

CFDCnFDCFG.GWEN = 1 on channel y

CFDCnFDCFG.GWFDF = 1

If a Classical CAN frame is received on channel x and routed to a gateway FIFO or TX Queue of channel y, then this CAN frame is sent on channel y as a CAN-FD frame because of the CFDCnFDCFG.GWFDF bit.

Table 33.4 shows how the message information is changed depending on the received and configured data.

Table 33.4 Modified message information by received and configured data (1 of 2)

Routed CAN frame	Routed received DLC	CAN BRS bit	Configured CFDCnFDCFG.GWFDF bit	Gateway message DLC	Gateway message BRS bit	Gateway message frame type
CAN2.0	≤ 8	N/A	1	≤ 8	Based on configuration of CFDCnFDCFG.GWBRS	CAN-FD
CAN2.0	> 8	N/A	1	= 8	Based on configuration of CFDCnFDCFG.GWBRS	CAN-FD
CAN-FD	≤ 8	None	1	≤ 8	Based on configuration of CFDCnFDCFG.GWBRS	CAN-FD
CAN-FD	> 8	None	1	> 8	Based on configuration of CFDCnFDCFG.GWBRS	CAN-FD
CAN2.0	≤ 8	N/A	0	≤ 8	N/A	CAN2.0
CAN2.0	> 8	N/A	0	> 8	N/A	CAN2.0

Table 33.4 Modified message information by received and configured data (2 of 2)

Routed CAN frame	Routed received DLC	CAN BRS bit	Configured CFDCnFDCFG.GWFDF bit	Gateway message DLC	Gateway message BRS bit	Gateway message frame type
CAN-FD	≤ 8	None	0	≤ 8	N/A	CAN2.0
CAN-FD	> 8	None	0	= 8	N/A	CAN2.0

Note:
- This gateway is limited to an 8-byte data payload for different frame type. If routing and target frame type is the same, the data length code (DLC) value remains the same. If the source frame is a CAN-FD with more than 8 data bytes, then on classical destination node, the data payload is reduced to 8 bytes. Only the first 8 bytes of data perform gateway transmission.
- Transmission buffers other than the gateway FIFO are not affected by this feature.

Do not route remote frames through the gateway when CFDCnFDCFG.GWEN is set. When a destination node is CFDCnFDCFG.FDOE = 1, set CFDCnFDCFG.GWEN and CFDCnFDCFG.GWFDF to 1. When a destination node is CFDCnFDCFG.CLOE = 1, set CFDCnFDCFG.GWEN = 1 and CFDCnFDCFG.GWFDF = 0.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET mode.

This bit is not available in the classical CAN function.

GWFDF bit (Gateway FDF Configuration Bit)

When the GWEN bit is set to 1, the FDF bit of the transmitting gateway frame is replaced by the value of the GWFDF bit.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to these bits when the CAN-FD module is in CH_RESET mode. Do not set this bit when in Classical-only mode.

This bit is not available in the classical CAN function.

GWBRS bit (Gateway BRS Configuration Bit)

When the GWEN bit is set to 1, the BRS bit of the transmitting gateway frame is replaced by the value of CFDCnFDCFG.GWBRS.

In classical CAN frames, the GWBRS bit is invalid.

Do not write to this bit in CH_OPERATION or CH_SLEEP mode.

Only write to this bit when the related CAN-FD channel is in CH_RESET mode. Do not set this bit when in Classical-only mode.

This bit is not available in the classical CAN function.

FDOE bit (FD-Only Enable)

The FDOE bit enables the reception and transmission of CAN-FD-only frames. If enabled, communication in Classical CAN frame format is disabled. Transmission of Classical CAN frames is not possible because the FDF bit of the message buffer is a don't care (CFDCFFDCSTSn.CFFDF/CFDTMFDCTRn.TMFDF).

If messages with Classical CAN frame format are received, the protocol controller treats them as invalid frames and response with error frames. When a Classical CAN frame is configured for transmitting, the FDF bit is sent as recessive, therefore an FD frame is sent. If the data length code (DLC) is configured of greater than 8 bytes, the remaining data bytes are padded with 0xCC.

The FDOE bit cannot be written in CH_OPERATION, CH_HALT or CH_SLEEP mode.

Do not set CFDCnFDCFG.FDOE and CFDCnFDCFG.CLOE simultaneously.

This bit is not available in the classical CAN function.

REFE bit (RX Edge Filter Enable)

The REFE bit enables the RX edge filter during the IDLE detection (bus integration). When the bit is enabled, two consecutive dominant time quanta are required to detect a synchronization edge.

The REFE bit cannot be written in CH_OPERATION, CH_HALT and CH_SLEEP mode. Do not set this bit when in Classical-only mode and when CFDCnFDCFG.CFDTE = 0 (disabled CAN-FD frame distinction).

CLOE bit (Classical CAN-Only Enable)

The CLOE bit enables the Classical CAN-only mode. If this bit is 1, the protocol controller can only send classical frames and response with a form or CRC error on FD frames.

Do not set CFDCnFDCFG.CLOE and CFDCnFDCFG.FDOE simultaneously.

CFDCnFDCFG.CLOE	CFDCnFDCFG.FDOE	Channel mode
0	0	CAN-FD mode (Dual mode)
0	1	FD-only mode
1	0	Classical CAN-only mode
1	1	Reserved

Do not write to this bit in CH_OPERATION, CH_HALT or CH_SLEEP mode.

Only write to these bits when the CAN-FD channel is in CH_RESET mode.

This bit is not available in the classical CAN function.

CFDTE bit (CAN-FD Frame Distinction Enable)

The CFDTE bit enables the CAN-FD frame distinction function. The CFDTE bit is required for Classical CAN-only mode (CFDCnFDCFG.CLOE = 1).

If this bit is 0, the protocol controller can only send Classical frames and response with a Form or CRC error on FD frames.

If this bit is 1, the protocol controller behaves according to the ISO 11898-1 (DIS 2015) specification. If the FDF bit is detected recessive, then the protocol controller enters the protocol exception state, and attempts to integrate back to the CAN communication.

Do not write this bit in CH_OPERATION, CH_HALT or CH_SLEEP mode.

Only write to these bits when the related CAN-FD channel is in CH_RESET mode.

33.2.7 CFDCnFDCTR : Channel n CAN-FD Control Register (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x1408 + 0x20 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SOCCLR	EOCCLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EOCCLR	Error Occurrence Counter Clear 0: No error occurrence counter clear 1: Clear error occurrence counter	R/W
1	SOCCLR	Successful Occurrence Counter Clear 0: No successful occurrence counter clear 1: Clear successful occurrence counter	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

The CFDCnFDCTR register (n = 0, 1) controls the error and successful occurrence counters.

EOCCLR bit (Error Occurrence Counter Clear)

The EOCCLR bit is used to clear the error occurrence counter.

Do not write to this bit in CH_SLEEP or CH_RESET mode. The read value is always 0.

This bit is cleared automatically by the CAN-FD logic and when the related CAN-FD channel is in CH_RESET mode.

SOCCLR bit (Successful Occurrence Counter Clear)

The SOCCLR bit is used to clear the successful occurrence counter.

Do not write to this bit in CH_SLEEP or CH_RESET mode. The read value is always 0.

This bit is cleared automatically by the CAN-FD logic and when the related CAN-FD channel is in CH_RESET mode.

33.2.8 CFDCnFDSTS : Channel n CAN-FD Status Register (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x140C + 0x20 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SOC[7:0]								EOC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TDCVF	—	—	—	—	—	SOCO	EOCO	TDCR[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TDCR[7:0]	Transceiver Delay Compensation Result	R
8	EOCO	Error Occurrence Counter Overflow Flag 0: Error occurrence counter has not overflowed 1: Error occurrence counter has overflowed	R/W
9	SOCO	Successful Occurrence Counter Overflow Flag 0: Successful occurrence counter has not overflowed 1: Successful occurrence counter has overflowed	R/W
14:10	—	These bits are read as 0. The write value should be 0.	R/W
15	TDCVF	Transceiver Delay Compensation Violation Flag 0: Transceiver delay compensation violation has not occurred 1: Transceiver delay compensation violation has occurred	R/W
23:16	EOC[7:0]	Error Occurrence Counter These bits show the error occurrence counter value.	R
31:24	SOC[7:0]	Successful Occurrence Counter These bits show the successful occurrence counter value.	R

The CFDCnFDSTS register (n = 0, 1) indicates the transceiver compensation delay result and its related FIFO message lost status.

TDCR[7:0] bits (Transceiver Delay Compensation Result)

The TDCR[7:0] bits are set when the transceiver delay has been measured.

The measured delay is a multiple of the CAN channel DLL clock. The result depends on the CFDCnFDCFG.TDCOC configuration and the offset value in CFDCnFDCFG.TDCO. See section 33.4.1.5. Transmitter Delay Compensation for details on how this value is derived.

The TDCR[7:0] bits are updated at the falling edge between the FDF bit and the RES bit when CFDCnFDCFG.TDCOC = 0 and the transceiver delay compensation is enabled (CFDCnFDCFG.TDCE = 1).

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

EOCO bit (Error Occurrence Counter Overflow Flag)

The EOCO bit indicates whether the related CAN channel error occurrence counter has overflowed. This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when CFDCnFDSTS.EOC is 0xFF and a CAN bus error is detected based on the configuration defined in CFDCnFDCFG.EOCCFG.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit.

SOCO bit (Successful Occurrence Counter Overflow Flag)

The SOCO bit indicates whether the related CAN channel successful occurrence counter has overflowed. This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when CFDCnFDSTS.SOC is 0xFF and a successful message reception or successful message transmission occurs.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit.

TDCVF bit (Transceiver Delay Compensation Violation Flag)

The CAN-FD module captures internally the transmitted data bit-by-bit. This data is then compared against the received CAN bus level which is delayed by the transceiver loop delay.

The transceiver delay has some variations depending on the physical parameters such as temperature. The result bit CFDCnFDSTS.TDCR is updated by each message. However, temporary maximum delay violation could be missed. Therefore, the TDCVF bit captures this violation.

This bit is cleared by writing 0 to it. Writing 1 has no effect.

This bit is set automatically when the transceiver delay compensation is greater than the maximum delay compensation (6 data bit times - 2 clk_dlc) and the internal bit is overrun.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

Do not use the bit clear instruction to clear this bit.

EOC[7:0] bits (Error Occurrence Counter)

The EOC[7:0] bits are used together with the SOC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.

This higher error rate can be detected depending on the configuration of the CFDCnFDCFG.EOCCFG bits.

The EOC[7:0] bits are set only by CAN-FD module logic. These bits are cleared by writing 1 to CFDCnFDCTR.EOCCLR. Writing any other value has no effect.

These bits are updated when an error occurs, according to the configuration of the CFDCnFDCFG.EOCCFG bits. When the counter reaches the value of 0xFF, the update stops.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

SOC[7:0] bits (Successful Occurrence Counter)

The SOC[7:0] bits are used together with the EOC[7:0] bits to support an option for host-controlled fall-back to payload bit rate identical to arbitration bit rate when messages utilizing the reduced payload bit length have significant higher error rates compared to other messages.

The SOC[7:0] bits are set only by CAN-FD module logic. Writing any other value has no effect.

These bits are updated when the occurrence of any error-free messages on the bus is detected through reception or transmission. When the counter reaches the value of 0xFF, the update stops. In Loopback mode, the counter is incremented twice.

These bits are cleared by writing 1 to CFDCnFDCTR.SOCCLR.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

33.2.9 CFDCnFDCRC : Channel n CAN-FD CRC Register (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x1410 + 0x20 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	SCNT[3:0]				—	—	—	CRCREG[20:0]																					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
20:0	CRCREG[20:0]	CRC Register Value These bits show the CRC value calculated for the CAN-FD frame.	R
24:21	—	These bits are read as 0.	R
28:25	SCNT[3:0]	Stuff Bit Count These bits shows the stuff bit count (mod 8) for the CAN-FD frame.	R
31:29	—	These bits are read as 0.	R

The CFDCnFDCRC register (n = 0, 1) holds the CRC value calculated for the CAN-FD frame.

CRCREG[20:0] bits (CRC Register Value)

The CRCREG[20:0] bits contain the CRC value calculated by the CAN-FD channel logic when the CFDCnCTR.CTME bit is enabled.

The CFDCnFDCRC.CRCREG value is updated in the first bit of the CRC field of the CAN-FD frame (reception and transmission).

When the CFDCnCTR.CTME bit is 0, the CRCREG[20:0] bits are always read as 0.

When bit 17th of the CRC field is used, CRCREG[20:17] are always read as 0.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

SCNT[3:0] bits (Stuff Bit Count)

The SCNT[3:0] bits contain the stuff count value of the CAN-FD frame. These bits indicate the number of inserted stuff bits (modulo 8, Gray-coded) for a CAN-FD frame when the CFDCnCTR.CTME bit is enabled in CFDCnFDCRC.SCNT[3:1]. SCNT[0] is the parity bit.

When the CFDCnCTR.CTME bit is 0, the SCNT[3:0] bits are always read as 0.

The SCNT value is updated in the first bit of CRC field of the CAN-FD frame (reception and transmission).

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

33.2.10 CFDGIPV : Global IP Version Register

Base address: CANFD = 0x8002_0000

Offset address: 0x0080

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	PSI[13:0]														—	—	—	—	—	—	IPT[1:0]		IPV[7:0]							
Value after reset:	0	0	1	1	0	0	0	1	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	*1	0	1	0	0	0	0	1	0

Bit	Symbol	Function	R/W
7:0	IPV[7:0]	IP Version These bits show the main release number.	R
9:8	IPT[1:0]	IP Type These bits show the IP type used in the product. 0 0: CAN IP 0 1: CAN-FD IP 1 x: Reserved	R
15:10	—	These bits are read as 0.	R
29:16	PSI[13:0]	Parameter Status Information These bits show the IP configuration.	R
31:30	—	These bits are read as 0.	R

Bits	Description
[29:27]	Number of channel (110b: 2 channels)
[26:24]	Number of TXMB (001b: 32/channel)
[23:21]	Number of AFL entry (001b: 128/channel)
[20:17]	Number of pool buffer (0010b: 64/channel)
[16]	Same ID overwrite function of TXQ (0b: Support)

Note 1. The value depends on RZ/T2 package type.

The CFDGIPV register shows the release version of the CAN-FD module.

33.2.11 CFDGCFG : Global Configuration Register

Base address: CANFD = 0x8002_0000

Offset address: 0x0084

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ITRCP[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TSBTCS[2:0]			TSSS	TSP[3:0]				—	—	CMPOC	DCS	MME	DRE	DCE	TPRI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TPRI	Transmission Priority 0: ID priority 1: Message buffer number priority	R/W
1	DCE	DLC Check Enable 0: DLC check disabled 1: DLC check enabled	R/W
2	DRE	DLC Replacement Enable 0: DLC replacement disabled 1: DLC replacement enabled	R/W
3	MME	Mirror Mode Enable 0: Mirror mode disabled 1: Mirror mode enabled	R/W
4	DCS	Data Link Controller Clock Select 0: PCLKM 1: PCLKCAN	R/W

Bit	Symbol	Function	R/W
5	CMPOC	CAN-FD Message Payload Overflow Configuration 0: Message is rejected 1: Message payload is cut to fit to configured message size	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
11:8	TSP[3:0]	Timestamp Prescaler 0x0: Timestamp prescaler = 1 0x1: Timestamp prescaler = 2 0x2: Timestamp prescaler = 4 0x3: Timestamp prescaler = 8 ⋮ 0xD: Timestamp prescaler = 8192 0xE: Timestamp prescaler = 16384 0xF: Timestamp prescaler = 32768	R/W
12	TSSS	Timestamp Source Select 0: Source clock for timestamp counter is peripheral clock 1: Source clock for timestamp counter is bit time clock	R/W
15:13	TSBTCS[2:0]	Timestamp Bit Time Channel Select 0 0 0: Select clock from channel 0 0 0 1: Select clock from channel 1 Others: Setting prohibited	R/W
31:16	ITRCP[15:0]	Interval Timer Reference Clock Prescaler FIFO interval timer prescaler value	R/W

The CFDGCFG register is used to select the transmission priority to be used for all the TX message buffers and the clock source for the CAN protocol engine of all CAN channels. The CFDGCFG register is also used to select the source for the timestamp clock and to configure the frequency for the timestamp clock and interval timer reference clock.

TPRI bit (Transmission Priority)

The TPRI bit selects the transmission priority for all CAN channels.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

Message buffer number priority should not be used together with TX queue transmission.

DCE bit (DLC Check Enable)

The DCE bit enables data length code (DLC) check for all CAN channels.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

DRE bit (DLC Replacement Enable)

When the DRE bit is 1 and the DCE is 1, the CAN-FD stores the configured value (CFDGAFLP0n.GAFLDLC) of the DLC in the destination RX message buffer or FIFO buffer if the DLC check passes. Otherwise, the DLC value in the destination RX message buffer or FIFO buffer is unchanged.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

MME bit (Mirror Mode Enable)

The MME bit enables the Mirror mode for all CAN channels.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

DCS bit (Data Link Controller Clock Select)

The DCS bit selects the clock source for CAN communication.

Do not write to this bit in GL_SLEEP or GL_OPERATION mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

CMPOC bit (CAN-FD Message Payload Overflow Configuration)

The CMPOC bit controls the message payload acceptance mechanism when the received payload is higher than the message buffer payload size CFDRMNB.RMPLS, CFDRFCCn.RFPLS, and CFDCFCCn.CFPLS. The received message payload is always compared with the available message payload size in the message buffer.

Do not write to this bit in GL_SLEEP or GL_OPERATION mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

When this bit is set and payload overflow occurs, the DLC value is stored in the RX message buffer or FIFO buffer unchanged.

TSP[3:0] bits (Timestamp Prescaler)

The value configured in the TSP[3:0] bits defines the period of the clock source used for the timestamp counter.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

TSSS bit (Timestamp Source Select)

The TSSS bit allows the selection of the clock source for the timestamp counter.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode. Additionally, do not set this bit to 1 when CAN-FD communication is used.

Note: The bit time clock varies depending on the nominal and data rate bit configuration.

TSBTCS[2:0] bits (Timestamp Bit Time Channel Select)

The TSBTCS[2:0] bits allow the selection of the bit time clock of a particular channel for the timestamp counter.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

ITRCP[15:0] bits (Interval Timer Reference Clock Prescaler)

The ITRCP[15:0] bits allow the definition of a reference clock for the FIFO interval timer source clock.

When these bits are 0x0000, the timer is disabled.

Do not write to this bit in GL_SLEEP mode. Only write to this bit when CAN-FD module is in GL_RESET mode.

33.2.12 CFDGCTR : Global Control Register

Base address: CANFD = 0x8002_0000

Offset address: 0x0088

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TSRST
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MOWEIE	QMEIE	—	QOWEIE	CMPOFIE	THLEIE	MEIE	DEIE	—	—	—	—	—	GSLPR	GMDC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

Bit	Symbol	Function	R/W
1:0	GMDC[1:0]	Global Mode Control 0 0: Global operation mode request 0 1: Global reset mode request 1 0: Global halt mode request 1 1: Keep current value	R/W
2	GSLPR	Global Sleep Request 0: Global sleep request disabled 1: Global sleep request enabled	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	DEIE	DLC Check Interrupt Enable 0: DLC check interrupt disabled 1: DLC check interrupt enabled	R/W
9	MEIE	Message Lost Error Interrupt Enable 0: Message lost error interrupt disabled 1: Message lost error interrupt enabled	R/W
10	THLEIE	TX History List Entry Lost Interrupt Enable 0: TX history list entry lost interrupt disabled 1: TX history list entry lost interrupt enabled	R/W
11	CMPOFIE	CAN-FD Message Payload Overflow Flag Interrupt Enable 0: CAN-FD message payload overflow flag interrupt disabled 1: CAN-FD message payload overflow flag interrupt enabled	R/W
12	QOWEIE	TXQ Message Overwrite Error Interrupt Enable 0: TXQ message overwrite error interrupt disabled 1: TXQ message overwrite error interrupt enabled	R/W
13	—	This bit is read as 0. The write value should be 0.	R/W
14	QMEIE	TXQ Message Lost Error Interrupt Enable 0: TXQ message lost error interrupt disabled 1: TXQ message lost error interrupt enabled	R/W
15	MOWEIE	Message Lost Error Interrupt Enable 0: GW FIFO message lost error interrupt disabled 1: GW FIFO message lost error interrupt enabled	R/W
16	TSRST	Timestamp Reset 0: Timestamp not reset 1: Timestamp reset	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

The CFDGCTR register controls the global mode of the CAN-FD module and the timestamp function. The register also enables and disables the global error interrupts.

GMDC[1:0] bits (Global Mode Control)

The GMDC[1:0] bits can be used to configure the modes for the CAN-FD module. Additionally, if CFDGCTR.GSLPR bit is 1 when the CAN-FD module is in Reset mode, the CAN-FD module transits to Global Sleep mode. Additionally, if CFDGCTR.GSLPR is 1 when the CAN-FD module is in Reset Mode, then the CAN-FD module transits to Global Sleep Mode.

Setting the GMDC[1:0] bits to 11b has no effect. Mode transition is described in detail in section 33.3.2. Global Modes.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

GSLPR bit (Global Sleep Request)

The GSLPR bit globally selects the sleep request for CAN-FD module including all CAN channels. Channel sleep request is set automatically for all channels.

Only write to this bit when the CAN-FD module is in GL_RESET or GL_SLEEP mode.

DEIE bit (DLC Check Interrupt Enable)

When the DEIE bit is 1, an interrupt is generated if a DLC error is detected in the received frames.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

MEIE bit (Message Lost Error Interrupt Enable)

When the MEIE bit is 1, an interrupt is generated if a message lost condition occurs.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

THLEIE bit (TX History List Entry Lost Interrupt Enable)

When the THLEIE bit is 1, an interrupt is generated if a TX history list entry lost condition occurs.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

CMPOFIE bit (CAN-FD Message Payload Overflow Flag Interrupt Enable)

When the CMPOFIE bit is 1, an interrupt is generated when a CAN-FD message payload overflow condition occurs.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

QOWEIE bit (TXQ Message Overwrite Error Interrupt Enable)

When the QOWEIE bit is 1, an interrupt is generated when a TXQ message overwrite condition occurs.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

QMEIE bit (TXQ Message Lost Error Interrupt Enable)

When the QMEIE bit is 1, an interrupt is generated when a TXQ message lost condition occurs.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

MOWEIE bit (Message Lost Error Interrupt Enable)

When the MOWEIE bit is 1, an interrupt is generated in GW mode and a GW FIFO message over write condition occurs.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

TSRST bit (Timestamp Reset)

When the TSRST bit is 1, the Global Timestamp Register is reset to 0x0000.

Do not write to this bit when the CAN-FD module is in GL_SLEEP or GL_RESET mode.

Read value is always 0.

This bit is cleared automatically by the CAN-FD logic.

33.2.13 CFDGSTS : Global Status Register

Base address: CANFD = 0x8002_0000

Offset address: 0x008C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	GRAM INIT	GSLP STS	GHLT STS	GRST STS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1

Bit	Symbol	Function	R/W
0	GRSTSTS	Global Reset Status 0: Not in Reset mode 1: In Reset mode	R
1	GHLTSTS	Global Halt Status 0: Not in Halt mode 1: In Halt mode	R
2	GSLPSTS	Global Sleep Status 0: Not in Sleep mode 1: In Sleep mode	R
3	GRAMINIT	Global RAM Initialization 0: RAM initialization is complete 1: RAM initialization is ongoing	R
31:4	—	These bits are read as 0.	R

The CFDGSTS register indicates the global status of the CAN-FD module.

GRSTSTS bit (Global Reset Status)

The GRSTSTS bit indicates the status of Global CAN-FD module Reset mode.

This bit is set automatically when the CAN-FD module enters GL_RESET mode. When the mode changes from GL_RESET mode to GL_SLEEP mode, this bit remains set.

This bit is cleared automatically when the CAN-FD module exits the GL_RESET mode.

GHLTSTS bit (Global Halt Status)

The GHLTSTS bit indicates the status of Global CAN-FD module Halt mode.

This bit is set automatically when the CAN-FD module enters GL_HALT mode.

This bit is cleared automatically when the CAN-FD module exits the GL_HALT mode.

GSLPSTS bit (Global Sleep Status)

The GSLPSTS bit indicates the status of Global CAN-FD module Sleep mode.

This bit is set automatically when the CAN-FD module enters GL_SLEEP mode.

This bit is cleared automatically when the CAN-FD module exits the GL_SLEEP mode.

GRAMINIT bit (Global RAM Initialization)

The GRAMINIT bit indicates the status of Global CAN-FD module RAM initialization.

This bit is set automatically when the CAN-FD module enters GL_SLEEP mode after a hardware reset.

This bit is cleared automatically when the CAN-FD module completed RAM initialization.

33.2.14 CFDGERFL : Global Error Flag Register

Base address: CANFD = 0x8002_0000

Offset address: 0x0090

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EEF1	EEF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	MOWES	QMES	—	QOWES	CMPOF	THLES	MES	DEF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DEF	DLC Error Flag 0: DLC error not detected 1: DLC error detected	R/W
1	MES	Message Lost Error Status 0: Message lost error not detected 1: Message lost error detected	R
2	THLES	TX History List Entry Lost Error Status 0: TX history list entry lost error not detected 1: TX history list entry lost error detected	R
3	CMPOF	CAN-FD Message Payload Overflow Flag 0: CAN-FD message payload overflow not detected 1: CAN-FD message payload overflow detected	R/W

Bit	Symbol	Function	R/W
4	QOWES	TXQ Message Overwrite Error Status 0: TXQ message overwrite error not detected 1: TXQ message overwrite error detected	R
5	—	This bit is read as 0. The write value should be 0.	R/W
6	QMES	TXQ Message Lost Error Status 0: TXQ message lost error not detected 1: TXQ message lost error detected	R
7	MOWES	Message Overwrite Error Status 0: Message overwrite error not detected 1: Message overwrite error detected	R
15:8	—	These bits are read as 0. The write value should be 0.	R/W
16	EEF0	ECC Error Flag for Channel 0 0: ECC error not detected during TX-SCAN 1: ECC error detected during TX-SCAN	R/W
17	EEF1	ECC Error Flag for Channel 1 0: ECC error not detected during TX-SCAN 1: ECC error detected during TX-SCAN	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The CFDGERFL egister indicates the detection of global errors.

DEF bit (DLC Error Flag)

The DEF bit indicates the error status of the DLC.

Do not write to this bit when the CAN-FD module is in GL_SLEEP or GL_RESET mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when a DLC error is detected in a received frame.

The bit is cleared by writing 0 to it.

This bit is cleared automatically in GL_RESET mode.

MES bit (Message Lost Error Status)

The MES bit indicates status of the message lost error.

This bit is set automatically when a FIFO message lost error is detected.

This bit is cleared automatically when:

- All FIFO message lost flags are cleared
- The CAN-FD module is in GL_RESET mode.

THLES bit (TX History List Entry Lost Error Status)

The THLES bit indicates status of the TX history list entry lost error.

This bit is set automatically when a TX history list entry lost error is detected.

This bit is cleared automatically when:

- All TX history list entry lost flags are cleared
- The CAN-FD module is in GL_RESET mode.

CMPOF bit (CAN-FD Message Payload Overflow Flag)

The CMPOF bit is set automatically when a CAN-FD message payload overflow is detected on at least one channel.

Do not write to this bit when the CAN-FD module is in GL_SLEEP or GL_RESET mode.

This bit is cleared by writing 0 to it. Writing 1 to this bit has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is cleared automatically in GL_RESET mode.

QOWES bit (TXQ Message Overwrite Error Status)

The QOWES bit is set automatically when a TXQ message overwrite error is detected.

This bit is cleared automatically when all TXQ message overwrite flags are cleared.

This bit is cleared automatically in GL_RESET mode.

QMES bit (TXQ Message Lost Error Status)

The QMES bit is set automatically when a TXQ message lost error is detected.

This bit is cleared automatically when all TXQ message lost flags are cleared.

This bit is cleared automatically in GL_RESET mode.

MOWES bit (Message Overwrite Error Status)

The MOWES bit is set automatically when GW mode and COMFIFO message overwrite error is detected.

This bit is cleared automatically when all COMFIFO Message Overwrite flags are cleared.

This bit is cleared automatically in GL_RESET mode.

EEF0 bit (ECC Error Flag for Channel 0)

The EEF0 bit specifies whether an ECC error has occurred on Channel 0.

Do not write to this bit when the CAN-FD module is in GL_SLEEP or GL_RESET mode. Writing 1 to this bit has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

The bit is cleared by writing 0 to it. This bit is cleared automatically in GL_RESET mode.

EEF1 bit (ECC Error Flag for Channel 1)

The EEF1 bit specifies whether an ECC error has occurred on Channel 1.

Do not write to this bit when the CAN-FD module is in GL_SLEEP or GL_RESET mode. Writing 1 to this bit has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

The bit is cleared by writing 0 to it. This bit is cleared automatically in GL_RESET mode.

33.2.15 CFDGTINTSTS0 : Global TX Interrupt Status Register 0

Base address: CANFD = 0x8002_0000

Offset address: 0x1300

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CFOTIF1	TQOFIF1	THIF1	CFTIF1	TQIF1	TAIF1	TSIF1	—	CFOTIF0	TQOFIF0	THIF0	CFTIF0	TQIF0	TAIF0	TSIF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSIF0	TX Successful Interrupt Flag Channel 0 0: Channel 0 TX Successful Interrupt flag not set 1: Channel 0 TX Successful Interrupt flag set	R
1	TAIF0	TX Abort Interrupt Flag Channel 0 0: Channel 0 TX Abort Interrupt flag not set 1: Channel 0 TX Abort Interrupt flag set	R
2	TQIF0	TX Queue Interrupt Flag Channel 0 0: Channel 0 TX Queue Interrupt flag not set 1: Channel 0 TX Queue Interrupt flag set	R
3	CFTIF0	COM FIFO TX/GW Mode Interrupt Flag Channel 0 0: Channel 0 COM FIFO TX/GW Mode Interrupt flag not set 1: Channel 0 COM FIFO TX/GW Mode Interrupt flag set	R
4	THIF0	TX History List Interrupt Channel 0 0: Channel 0 TX History List Interrupt flag not set 1: Channel 0 TX History List Interrupt flag set	R
5	TQOFIF0	TX Queue One Frame Transmission Interrupt Flag Channel 0 0: Channel 0 TX Queue One Frame Transmission Interrupt flag not set 1: Channel 0 TX Queue One Frame Transmission Interrupt flag set	R
6	CFOTIF0	COM FIFO One Frame Transmission Interrupt Flag Channel 0 0: Channel 0 COM FIFO One Frame Transmission Interrupt flag not set 1: Channel 0 COM FIFO One Frame Transmission Interrupt flag set	R
7	—	This bit is read as 0.	R
8	TSIF1	TX Successful Interrupt Flag Channel 1 0: Channel 1 TX Successful Interrupt flag not set 1: Channel 1 TX Successful Interrupt flag set	R
9	TAIF1	TX Abort Interrupt Flag Channel 1 0: Channel 1 TX Abort Interrupt flag not set 1: Channel 1 TX Abort Interrupt flag set	R
10	TQIF1	TX Queue Interrupt Flag Channel 1 0: Channel 1 TX Queue Interrupt flag not set 1: Channel 1 TX Queue Interrupt flag set	R
11	CFTIF1	COM FIFO TX/GW Mode Interrupt Flag Channel 1 0: Channel 1 COM FIFO TX/GW Mode Interrupt flag not set 1: Channel 1 COM FIFO TX/GW Mode Interrupt flag set	R
12	THIF1	TX History List Interrupt Channel 1 0: Channel 1 TX History List Interrupt flag not set 1: Channel 1 TX History List Interrupt flag set	R
13	TQOFIF1	TX Queue One Frame Transmission Interrupt Flag Channel 1 0: Channel 1 TX Queue One Frame Transmission Interrupt flag not set 1: Channel 1 TX Queue One Frame Transmission Interrupt flag set	R
14	CFOTIF1	COM FIFO One Frame Transmission Interrupt Flag Channel 1 0: Channel 1 COM FIFO One Frame Transmission Interrupt flag not set 1: Channel 1 COM FIFO One Frame Transmission Interrupt flag set	R
31:15	—	These bits are read as 0.	R

The CFDGTINTSTS0 register indicates the detection of transmit specific interrupts.

TSIFn bit (TX Successful Interrupt Flag Channel n (n = 0, 1))

The CFDGTINTSTS0.TSIFn bit is set to 1 when the TX Successful Interrupt flag of the related channel is set (when the interrupt is enabled).
This bit is cleared automatically:

- When the related TX MB Result Status bits are cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

TAIFn bit (TX Abort Interrupt Flag Channel n (n = 0, 1))

The CFDGTINTSTS0.TAIFn bit is set to 1 when the TX Abort Interrupt flag of the related channel is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX MB Result Status bits are cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

TQIFn bit (TX Queue Interrupt Flag Channel n (n = 0, 1))

The CFDGTINTSTS0.TQIFn bit is set to 1 when the TX Queue Interrupt flag of the related channel is set (when the interrupt is enabled).
This bit is cleared automatically:

- When the related TX Queue Interrupt flag is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

CFTIFn bit (COM FIFO in TX/GW Mode Interrupt Flag Channel n (n = 0, 1))

The CFDGTINTSTS0.CFTIFn bit is set to 1 when the related COM TX/GW FIFO Mode Interrupt flag (CFDCFSTSn.CFTXIF) is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related COM TX/GW FIFO Mode Interrupt flag (CFDCFSTSn.CFTXIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

THIFn bit (TX History List Interrupt Flag Channel n (n = 0, 1))

The CFDGTINTSTS0.THIFn bit is set to 1 when the related TX History List Interrupt flag (CFDTHLSTSn.THLIF) is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX History List Interrupt flag (CFDTHLSTSn.THLIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

TQOFIFn bit (TX Queue One Frame Transmission Interrupt Flag Channel n (n = 0, 1))

The CFDGTINTSTS0.TQOFIFn bit is set to 1 when the TX Queue One Frame Transmission Interrupt flag of the related channel is set (when the interrupt is enabled).

This bit is cleared automatically:

- When the related TX Queue One Frame Transmission Interrupt flag is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

CFOTIFn bit (COM FIFO One Frame Transmission Interrupt Flag Channel n (n = 0, 1))

The CFDGTINTSTS0.CFOTIFn bit is set to 1 when the related COM FIFO One Frame Transmission Interrupt flag (CFDCFSTSn.CFOFTXIF) is set (when the Interrupt is enabled).

This bit is cleared automatically:

- When the related COM FIFO One Frame Transmission Interrupt flag (CFDCFSTSn.CFOFTXIF) is cleared (when the interrupt enable is disabled)
- When in GL_RESET or CH_RESET mode.

33.2.16 CFDGTSC : Global Timestamp Counter Register

Base address: CANFD = 0x8002_0000

Offset address: 0x0094

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TS[15:0]	Timestamp value	R
31:16	—	These bits are read as 0.	R

The CFDGTSC register stores the timestamp based on the selected configuration.

TS[15:0] bits (Timestamp value)

The Timestamp value is stored in the Global Timestamp Counter register based on the configuration of TSSS, TSBTCS, and TSP. The proper incrementing of the timestamp counter cannot be guaranteed when transitioning to halt state.

Do not write to bits TS[15:0] when the CAN-FD module is in GL_RESET or GL_SLEEP mode.

The TS[15:0] bits are cleared automatically in GL_RESET mode.

33.2.17 CFDGAFLECTR : Global Acceptance Filter List Entry Control Register

Base address: CANFD = 0x8002_0000

Offset address: 0x0098

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	AFLDAE	—	—	—	—	AFLPN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	AFLPN[3:0]	Acceptance Filter List Page Number Select an Acceptance Filter List page	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	AFLDAE	Acceptance Filter List Data Access Enable 0: Acceptance Filter List data access disabled 1: Acceptance Filter List data access enabled	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

The CFDGAFLECTR register is used to select the Global Acceptance Filter List page for reading or writing entries into the Global Acceptance Filter List.

AFLPN[3:0] bits (Acceptance Filter List Page Number)

The AFLPN[3:0] bits select the page number to access the desired RAM area of the Acceptance Filter List. One Acceptance Filter List page consists of 16 Acceptance Filter List entries.

Read/write accesses to the Acceptance Filter List can only be performed through a fixed window.

Do not write to these bits when the CAN-FD module is in GL_SLEEP mode. Enter only the values between 0x0 and 0xF, inclusive.

AFLDAE bit (Acceptance Filter List Data Access Enable)

The AFLDAE bit prevents write access to the Acceptance Filter List when cleared after configuration of the Acceptance Filter List.

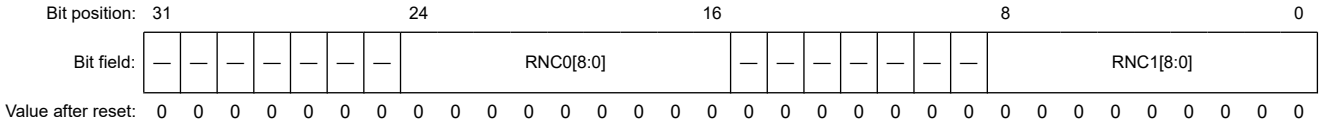
Data can be read from the Acceptance Filter List independent of the status of this bit.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode. Set this bit to enable write access for the Acceptance Filter List.

33.2.18 CFDGAFLCFG0 : Global Acceptance Filter List Configuration Register 0

Base address: CANFD = 0x8002_0000

Offset address: 0x009C



Bit	Symbol	Function	R/W
8:0	RNC1[8:0]	Rule Number for Channel 1 Number of rules dedicated to channel 1	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
24:16	RNC0[8:0]	Rule Number for Channel 0 Number of rules dedicated to channel 0	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The CFDGAFLCFG0 register is used to define the number of rules for entries in the Acceptance Filter List, applicable for channels 0 to 1.

The total number of available entries in the Acceptance Filter List is 256 for 2 CAN channels. However, the filters can be allocated flexibly to the different channels depending on requirements as long as both of the following conditions are satisfied:

- The maximum number of acceptance filter per channel is 256
- The total number of rules defined for all channels is not exceeding the number of available entries in the Acceptance Filter List.

RNCn[8:0] bits (Rule Number for Channel n (n = 0, 1))

The RNCn[8:0] bits define the number of rules in the Acceptance Filter List for channel n.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

33.2.19 CFDGAFLIDn : Global Acceptance Filter List ID Register n (n = 1 to 16)

Base address: CANFD = 0x8002_0000

Offset address: 0x1800 + 0x10 × (n - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	GAFLIDE	GAFLRTR	GAFLLB	GAFLID[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GAFLID[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
28:0	GAFLID[28:0]	Global Acceptance Filter List Entry ID Field ID part of the Global Acceptance Filter List entry	R/W
29	GAFLLB	Global Acceptance Filter List Entry Loopback Configuration 0: Global Acceptance Filter List entry ID for acceptance filtering with attribute RX 1: Global Acceptance Filter List entry ID for acceptance filtering with attribute TX	R/W
30	GAFLRTR	Global Acceptance Filter List Entry RTR Field 0: Data frame 1: Remote frame	R/W
31	GAFLIDE	Global Acceptance Filter List Entry IDE Field 0: Standard identifier of rule entry ID is valid for acceptance filtering 1: Extended identifier of rule entry ID is valid for acceptance filtering	R/W

The CFDGAFLIDn register (n = 1 to 16) is used to configure the ID field for the rules of entries in the Global Acceptance Filter List.

GAFLID[28:0] bits (Global Acceptance Filter List Entry ID Field)

The GAFLID[28:0] bits represent the CAN identifier (ID) field of each entry in the Global Acceptance Filter List.

The acceptance filter process compares this field against the ID of a received CAN message. For alignment of these bits in standard and extended frame formats, see section 33.2.88. Identifier Bits Alignment.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLLB bit (Global Acceptance Filter List Entry Loopback Configuration)

The GAFLLB bit selects whether entry in the Global Acceptance Filter List gets the attribute RX or TX.

This attribute determines the validity of the entry in Mirror mode, Loopback test mode, and during standard (non-loopback) reception. See section 33.5.5. Loopback Modes for detailed description of the validity of the Global Acceptance Filter List entry depending on transmitter/receiver case, the type of loopback mode, and RX/TX attribute.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLRTR bit (Global Acceptance Filter List Entry RTR Field)

The GAFLRTR bit allows the configuration of the specified frame format (data frame or remote frame) for each entry of the Global Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the RTR bit of the received CAN message.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLIDE bit (Global Acceptance Filter List Entry IDE Field)

The GAFLIDE bit allows the configuration of the ID format (standard ID or extended ID) for each entry in the Global Acceptance Filter List. For each rule entry in a CAN channel, the acceptance filter process compares this bit against the IDE bit of the received CAN message.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

33.2.20 CFDGAFLMn : Global Acceptance Filter List Mask Register n (n = 1 to 16)

Base address: CANFD = 0x8002_0000

Offset address: 0x1804 + 0x10 × (n - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	GAFLIDEM	GAFLRTRM	GAFLIFL1	GAFLIDM[28:16]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GAFLIDM[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
28:0	GAFLIDM[28:0]	Global Acceptance Filter List ID Mask Field Global Acceptance Filter List Mask field bits for ID field	R/W
29	GAFLIFL1	Global Acceptance Filter List Information Label 1 Global Acceptance Filter List information label bit 1	R/W
30	GAFLRTRM	Global Acceptance Filter List Entry RTR Mask 0: RTR bit is not used for ID matching 1: RTR bit is used for ID matching	R/W
31	GAFLIDEM	Global Acceptance Filter List IDE Mask 0: IDE bit is not used for ID matching 1: IDE bit is used for ID matching	R/W

The CFDGAFLMn register (n = 1 to 16) is used to configure the Mask field of each rule for entries in the Global Acceptance Filter List.

GAFLIDM[28:0] bits (Global Acceptance Filter List ID Mask Field)

GAFLIDM[28:0] bits are the filter mask bits for the related bits in the CAN Identifier field of each Global Acceptance Filter List entry.

0	Corresponding STD-ID/EXT-ID bit is not used for ID matching
1	Corresponding STD-ID/EXT-ID bit is used for ID matching

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLIFL1 bit (Global Acceptance Filter List Information Label 1)

The GAFLIFL1 bit allows the configuration of a 2-bit information label to be attached to a received message accepted by the associated entry in the Global Acceptance Filter List. This bit is a MSB bit of an information label.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

This bit is stored in the Information Label Field [1] (CFDRMFDSTS.RMIFL[1], CFDRFFDSTS.RFIFL[1], CFDCFFDCSTSn.CFIFL[1]) of the storage location of an incoming message.

Note: This bit is stored in CFDTHLACC1n.TIFL[1] when CFDTHLCCn.THLDGE = 1 is set up using Gateway function.

GAFLRTRM bit (Global Acceptance Filter List Entry RTR Mask)

The GAFLRTRM bit allows the configuration of the RTR mask bit for each entry in the Global Acceptance Filter List.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLIDEM bit (Global Acceptance Filter List IDE Mask)

The GAFLIDEM bit allows the configuration of the IDE mask bit for each entry in the Global Acceptance Filter List.

When the IDE mask bit is 0, the ID comparison depends on the received IDE bit.

If the received IDE bit is 0, the STD-ID comparison takes place.

If the received IDE bit is 1, the EXT-ID comparison takes place.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to this bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

33.2.21 CFDGAFLP0n : Global Acceptance Filter List Pointer 0 Register n (n = 1 to 16)

Base address: CANFD = 0x8002_0000

Offset address: 0x1808 + 0x10 × (n - 1)

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	GAFLPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GAFLRMV	—	—	GAFLRMDP[4:0]					GAFLIFL0	GAFLSRD2	GAFLSRD1	GAFLSRD0	GAFLDLC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	GAFLDLC[3:0]	Global Acceptance Filter List DLC Field Minimum number of data bytes in a data frame required for acceptance	R/W
4	GAFLSRD0	Global Acceptance Filter List Select Routing Destination 0 0: Routing target is CFIFO0 1: Routing target is TX Queue 0 instead of CFIFO0	R/W
5	GAFLSRD1	Global Acceptance Filter List Select Routing Destination 1 0: Routing target is CFIFO1 1: Routing target is TX Queue 1 instead of CFIFO1	R/W
6	GAFLSRD2	Global Acceptance Filter List Select Routing Destination 2 0: Routing target is CFIFO2 1: Routing target is TX Queue 2 instead of CFIFO2	R/W
7	GAFLIFL0	Global Acceptance Filter List Information Label 0	R/W
12:8	GAFLRMDP[4:0]	Global Acceptance Filter List RX Message Buffer Direction Pointer RX message buffer number for storage of received messages	R/W
14:13	—	These bits are read as 0. The write value should be 0.	R/W
15	GAFLRMV	Global Acceptance Filter List RX Message Buffer Valid 0: Single message buffer direction pointer is invalid 1: Single message buffer direction pointer is valid	R/W
31:16	GAFLPTR[15:0]	Global Acceptance Filter List Pointer	R/W

The CFDGAFLP0n register (n = 1 to 16) is used to configure the data length code (DLC), software pointer, single message buffer select, and message buffer direction pointer for each rule entry in the Global Acceptance Filter List.

GAFLDLC[3:0] bits (Global Acceptance Filter List DLC Field)

The GAFLDLC[3:0] bits allow the configuration of a minimum data length code (DLC) value for a message to be accepted by the associated entry in the Global Acceptance Filter List (automatic DLC filter function).

DLC filter process is only passed if the DLC value of the message accepted by an entry in the Global Acceptance Filter List is equal to or higher than the DLC value configured for this associated Global Acceptance Filter List entry. Automatic DLC filter function is disabled for the corresponding rule entry when this field is set to 0.

Table 33.5 shows DLC value that can be configured.

Table 33.5 Configuration of DLC value

Format	DLC[3:0]	Description
CAN and CAN-FD	0000b	DLC of received message = 0 or more (DLC filter check is disabled)
CAN and CAN-FD	0001b	DLC of received message = 1 or more
CAN and CAN-FD	0010b	DLC of received message = 2 or more
CAN and CAN-FD	0011b	DLC of received message = 3 or more
CAN and CAN-FD	0100b	DLC of received message = 4 or more
CAN and CAN-FD	0101b	DLC of received message = 5 or more
CAN and CAN-FD	0110b	DLC of received message = 6 or more
CAN and CAN-FD	0111b	DLC of received message = 7 or more
CAN	1xxxb	DLC of received message = 8 or more
CAN-FD	1000b	DLC of received message = 8 or more*1
CAN-FD	1001b	DLC of received message = 12 or more*1
CAN-FD	1010b	DLC of received message = 16 or more*1
CAN-FD	1011b	DLC of received message = 20 or more*1
CAN-FD	1100b	DLC of received message = 24 or more*1
CAN-FD	1101b	DLC of received message = 32 or more*1
CAN-FD	1110b	DLC of received message = 48 or more*1
CAN-FD	1111b	DLC of received message = 64*1

Note 1. This setting is not available in the classical CAN function.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLSRD0 bit (Global Acceptance Filter List Select Routing Destination 0)

The GAFLSRD0 bit changes a copy destination to CFIFO0 or TXQ0 by routing.

If this bit is set as 1, the preset value of CFDGAFLP1n.GAFLFDP selects TXQ0.

If this bit is set to 0, the preset value of CFDGAFLP1n.GAFLFDP selects CFIFO0.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to the bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLSRD1 bit (Global Acceptance Filter List Select Routing Destination 1)

The GAFLSRD1 bit changes a copy destination to CFIFO1 or TXQ1 by routing.

If this bit is set to 1, the preset value of CFDGAFLP1n.GAFLFDP selects TXQ1.

If this bit is set to 0, the preset value of CFDGAFLP1n.GAFLFDP selects CFIFO1.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to the bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLSRD2 bit (Global Acceptance Filter List Select Routing Destination 2)

The GAFLSRD2 bit changes a copy destination to CFIFO2 or TXQ2 by routing.

If this bit is set to 1, the preset value of CFDGAFLP1n.GAFLFDP selects TXQ2.

If this bit is set to 0, the preset value of CFDGAFLP1n.GAFLFDP selects CFIFO2.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to the bit when the CAN-FD module is in CH_RESET or CH_HALT mode.

GAFLIFL0 bit (Global Acceptance Filter List Information Label 0)

The GAFLIFL0 bit allows the configuration of a 2-bit information label that can be attached to a received message accepted by the related Global Acceptance Filter List entry. This bit is a LSB bit of an information label.

Do not write to this bit when CFDGAFLECTR.AFLDAE bit is 0.

Only write to the bit when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

This bit is stored in Information Label Field[0] (CFDRMFDSTS.RMIFL[0], CFDRFFDSTS.RFIFL[0], CFDCFFDCSTSn.CFIFL[0]) of the storage location of an incoming message.

This bit is stored in CFDTHLACC1n.TIFL[0] when CFDTHLCCn.THLDGE = 1 is set up using the gateway function.

GAFLRMDP[4:0] bits (Global Acceptance Filter List RX Message Buffer Direction Pointer)

The GAFLRMDP[4:0] bits allow the configuration of a single reception message buffer as the destination target for a received message that passes the acceptance check of the related Global Acceptance Filter List entry. The value entered is the single destination message buffer number.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

CFDRMNB.NRXMB[7:0] is the value entered in the RX Message Buffer Number Register to configure the number of RX message buffers. The value to be entered in CFDGAFLP0n.GAFLRMDP[4:0] bits should only be between 0x00 and CFDRMNB.NRXMB[7:0] to 1 less.

If CFDRMNB.NRXMB[7:0] = 0x00, the GAFLRMV bit should be configured as 0.

GAFLRMV bit (Global Acceptance Filter List RX Message Buffer Valid)

The GAFLRMV bit allows the enabling or disabling of a single reception message buffer as the target for a received message that passes the acceptance check of the related Global Acceptance Filter List entry.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

GAFLPTR[15:0] bits (Global Acceptance Filter List Pointer)

The GAFLPTR[15:0] bits allow the configuration of a 16-bit pointer to be attached to a received message accepted by the related Global Acceptance Filter List entry. The pointer is added during message storage in the Message Buffer area and can be used by the application as a support function. The pointer information can be used for example, to support PDU Identifier allocation for the received message in AUTOSAR systems.

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0.

Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

33.2.22 CFDGAFLP1n : Global Acceptance Filter List Pointer 1 Register n (n = 1 to 16)

Base address: CANFD = 0x8002_0000

Offset address: 0x180C + 0x10 × (n - 1)

Bit position:	31														13																0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	GAFLFDP[13:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
13:0	GAFLFDP[13:0]	Global Acceptance Filter List FIFO Direction Pointer FIFO direction pointer bits for received message storage	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The CFDGAFLP1n register (n = 1 to 16) is used to configure the FIFO direction pointer fields in each Rule Entry of the Global Acceptance Filter List.

GAFLFDP[13:0] bits (Global Acceptance Filter List FIFO Direction Pointer)

The GAFLFDP[13:0] bits allow the configuration of FIFO buffers as the target for a received message that passes the acceptance check of the related Global Acceptance Filter List entry. Each bit of the CFDGAFLP1n.GAFLFDP[13:0] is configured as dedicated FIFO.

Bit	Value (binary)	Function
0	0	Disable RX FIFO 0 as target for reception
	1	Enable RX FIFO 0 as target for reception
1	0	Disable RX FIFO 1 as target for reception
	1	Enable RX FIFO 1 as target for reception
2	0	Disable RX FIFO 2 as target for reception
	1	Enable RX FIFO 2 as target for reception
3	0	Disable RX FIFO 3 as target for reception
	1	Enable RX FIFO 3 as target for reception
4	0	Disable RX FIFO 4 as target for reception
	1	Enable RX FIFO 4 as target for reception
5	0	Disable RX FIFO 5 as target for reception
	1	Enable RX FIFO 5 as target for reception
6	0	Disable RX FIFO 6 as target for reception
	1	Enable RX FIFO 6 as target for reception
7	0	Disable RX FIFO 7 as target for reception
	1	Enable RX FIFO 7 as target for reception
8	0	Disable Common FIFO 0 and Channel 0 TX Queue 0 as target for reception
	1	GAFLSRD0 = 0: Enable Common FIFO 0 as target for reception GAFLSRD0 = 1: Enable Channel 0 TX Queue 0 as target for reception
9	0	Disable Common FIFO 1 and Channel 0 TX Queue 1 as target for reception
	1	GAFLSRD1 = 0: Enable Common FIFO 1 as target for reception GAFLSRD1 = 1: Enable Channel 0 TX Queue 1 as target for reception
10	0	Disable Common FIFO 2 and Channel 0 TX Queue 2 as target for reception
	1	GAFLSRD2 = 0: Enable Common FIFO 2 as target for reception GAFLSRD2 = 1: Enable Channel 0 TX Queue 2 as target for reception
11	0	Disable Common FIFO 3 and Channel 1 TX Queue 0 as target for reception
	1	GAFLSRD0 = 0: Enable Common FIFO 3 as target for reception GAFLSRD0 = 1: Enable Channel 1 TX Queue 0 as target for reception
12	0	Disable Common FIFO 4 and Channel 1 TX Queue 1 as target for reception
	1	GAFLSRD1 = 0: Enable Common FIFO 4 as target for reception GAFLSRD1 = 1: Enable Channel 1 TX Queue 1 as target for reception
13	0	Disable Common FIFO 5 and Channel 1 TX Queue 2 as target for reception
	1	GAFLSRD2 = 0: Enable Common FIFO 5 as target for reception GAFLSRD2 = 1: Enable Channel 1 TX Queue 2 as target for reception

Do not write to these bits when CFDGAFLECTR.AFLDAE bit is 0. Only write to these bits when the related CAN-FD channel is in CH_RESET or CH_HALT mode.

For storage in TX queue, TX queue buffers of a target that is in GW mode is possible.

Only one of the following configurations is valid:

- Up to 8 destination FIFO buffers
- 7 destination FIFO buffers plus one RX message buffer
- 8 destination TX queue buffers
- 7 destination TX queue buffers plus one RX message buffer
- A maximum of 8 destinations in all at FIFO buffer and TX queue buffer.

33.2.23 CFDRMNB : RX Message Buffer Number Register

Base address: CANFD = 0x8002_0000

Offset address: 0x00AC

Bit position:	31																				10	9	8	7								0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RMPLS[2:0]			NRXMB[7:0]								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	NRXMB[7:0]	Number of RX Message Buffers	R/W
10:8	RMPLS[2:0]*1	Reception Message Buffer Payload Data Size 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
31:11	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits are not available in the classical CAN function.

The CFDRMNB register is used to configure the total number of RX message buffers allocated to all channels.

NRXMB[7:0] bits (Number of RX Message Buffers)

The NRXMB[7:0] bits are used to configure the number of RX message buffers.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

Enter only values between 0 and 32 inclusive, with 0x00 indicating that no RX message buffer is allocated.

RMPLS[2:0] bits (Reception Message Buffer Payload Data Size)

The RMPLS[2:0] bits are used to configure the message buffer payload data size.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

Note: These bits are not available in the classical CAN function.

33.2.24 CFDRMND0 : RX Message Buffer New Data Register 0

Base address: CANFD = 0x8002_0000

Offset address: 0x00B0

Bit position:	31																												0
Bit field:	RMNS[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RMNS[31:0]	RX Message Buffer New Data Status 0: New data not stored in corresponding RX message buffer 1: New data stored in corresponding RX message buffer	R/W

The CFDRMND0 specifies the new data storage status of the RX message buffers.

RMNS[31:0] bits (RX Message Buffer New Data Status)

The RMNSu[31:0] bits indicate the status of new data for the corresponding RX message buffer. RMNS bit [0] corresponds to RX message buffer [0] and so on.

no_of_channels = 2

no_of_CFDRMBCPs_per_channel = (number of RX message buffer components per channel) = 16

no_of_CFDRMBCPs = no_of_channels × no_of_CFDRMBCPs_per_channel (2 * 16 = 32)

no_of_bits_per_register = 32

no_of_CFDRMNDs (number of CFDRMND Registers) = no_of_CFDRMBCPs / no_of_bits_per_register (32 / 32 = 1)

t = [0 ... no_of_CFDRMNDs - 1]

u = [t * 32 ... (no_of_CFDRMBCPs − ((no_of_CFDRMNDs − 1 − t) * 32) − 1)]

t can be calculated from the target of the New Data Status flag (u) using the formula t = floor (u / 32)

Bit position can be calculated using the formula (u − (t * 32))

Do not write to these bits when the CAN-FD module is in GL_RESET or GL_SLEEP mode. Writing 1 has no effect.

These bits cannot be cleared when message storage in the corresponding RX message buffer is in progress.
Do not use the bit clear instruction to clear these bits. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

These bits are set automatically when storage of new messages are in the corresponding RX message buffer. These bits are cleared by writing 0. These bits are cleared automatically when the CAN-FD module is in GL_RESET mode.

When CFDRMNB.RMPLS = 000b (maximum 8 bytes payload), the duration of message storage is 6 PCLKM cycles.

When CFDRMNB.RMPLS > 000b, the duration of message storage is 6 PCLKM cycles + 1 for each 4 bytes (maximum of 20 PCLKM cycles for 64 bytes).

Note: This feature is not available in the classical CAN function.

33.2.25 CFDRFCCn : RX FIFO Configuration/Control Register n (n = 0 to 7)

Base address: CANFD = 0x8002_0000

Offset address: 0x00C0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFFIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RFIGCV[2:0]			RFIM	—	RFDC[2:0]			—	RFPLS[2:0]			—	—	RFIE	RFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFE	RX FIFO Enable 0: FIFO disabled 1: FIFO enabled	R/W
1	RFIE	RX FIFO Interrupt Enable 0: FIFO interrupt generation disabled 1: FIFO interrupt generation enabled	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
6:4	RFPLS[2:0]*1	Rx FIFO Payload Data Size Configuration 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
10:8	RFDC[2:0]	RX FIFO Depth Configuration 0 0 0: FIFO Depth = 0 messages 0 0 1: FIFO Depth = 4 messages 0 1 0: FIFO Depth = 8 messages 0 1 1: FIFO Depth = 16 messages 1 0 0: FIFO Depth = 32 messages 1 0 1: FIFO Depth = 48 messages 1 1 0: FIFO Depth = 64 messages 1 1 1: FIFO Depth = 128 messages	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
12	RFIM	RX FIFO Interrupt Mode 0: Interrupt generated when RX FIFO counter reaches RFIGCV value from values smaller than RFIGCV 1: Interrupt generated at the end of every received message storage	R/W
15:13	RFIGCV[2:0]	RX FIFO Interrupt Generation Counter Value 0 0 0: Interrupt generated when FIFO is 1/8th full 0 0 1: Interrupt generated when FIFO is 1/4th full 0 1 0: Interrupt generated when FIFO is 3/8th full 0 1 1: Interrupt generated when FIFO is 1/2 full 1 0 0: Interrupt generated when FIFO is 5/8th full 1 0 1: Interrupt generated when FIFO is 3/4th full 1 1 0: Interrupt generated when FIFO is 7/8th full 1 1 1: Interrupt generated when FIFO is full	R/W
16	RFFIE	RX FIFO Full Interrupt Enable 0: FIFO interrupt generation disabled 1: FIFO interrupt generation enabled	R/W

Bit	Symbol	Function	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits are not available in the classical CAN function.

The CFDRFCCn registers (n = 0 to 7) are used to configure and control the eight RX FIFOs.

RFE bit (RX FIFO Enable)

The RFE bit enables the FIFO. When this bit is set to 0, the RX FIFO is cleared to empty.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

This bit can only be set if the configured FIFO depth is greater than 000b (CFDRFCCn.RFDC[2:0] > 000b).

Set the RFE bit with a separate write access to the CFDRFCCn register, after all the other bits in the CFDRFCCn register are set.

This bit is cleared automatically when the CAN-FD module is in GL_RESET mode.

RFIE bit (RX FIFO Interrupt Enable)

The RFIE bit enables generation of the FIFO interrupt.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

RFPLS[2:0] bits (Rx FIFO Payload Data Size Configuration)

The RFPLS[2:0] bits define the message data payload allocation in the RAM.

This is the maximum number of bytes which can be received by this FIFO.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

Note: These bits are not available in the classical CAN function.

RFDC[2:0] bits (RX FIFO Depth Configuration)

The RFDC[2:0] bits select the depth of the FIFO in terms of the number of messages. If the FIFO depth is configured to 0 messages, the FIFO cannot be used.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

RFIM bit (RX FIFO Interrupt Mode)

The RFIM bit selects the interrupt generation condition for the FIFO.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

RFIGCV[2:0] bits (RX FIFO Interrupt Generation Counter Value)

The RFIGCV[2:0] bits select the counter value of the FIFO for generation of FIFO interrupts. These values represent fractions of the FIFO depth for which an interrupt is generated.

Do not write to these bits when the CAN-FD module is in GL_SLEEP mode.

The setting of the RFIGCV[2:0] bits should be synchronized with the RFDC[2:0] bits.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

RFFIE bit (RX FIFO Full Interrupt Enable)

The RFFIE bit enables generation of the RX FIFO full interrupt. Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

The following content shows examples of interruptions:

- Interruption output in number of arbitrary stages (CFDRFCCn.RFIGCV)
- Interruption output in FIFO full state.

Note: Management of the receiving data of FIFO can be performed by these notices of interruption.

33.2.26 CFDRFSTSn : RX FIFO Status Register n (n = 0 to 7)

Base address: CANFD = 0x8002_0000

Offset address: 0x00E0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFFIF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RFMC[7:0]								—	—	—	—	RFIF	RFMLT	RFFLL	RFEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	RFEMP	RX FIFO Empty 0: FIFO not empty 1: FIFO empty	R
1	RFFLL	RX FIFO Full 0: FIFO not full 1: FIFO full	R
2	RFMLT	RX FIFO Message Lost 0: No message lost in FIFO 1: FIFO message lost	R/W
3	RFIF	RX FIFO Interrupt Flag 0: FIFO interrupt condition not satisfied 1: FIFO interrupt condition satisfied	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
15:8	RFMC[7:0]	RX FIFO Message Count Number of messages stored in FIFO	R
16	RFFIF	RX FIFO Full Interrupt Flag 0: FIFO full interrupt condition not satisfied 1: FIFO full interrupt condition satisfied	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

The CFDRFSTSn register (n = 0 to 7) shows the status of messages stored in the corresponding FIFO buffers.

RFEMP bit (RX FIFO Empty)

The RFEMP bit is set automatically when:

- The RFMC bit is 0
- RX FIFO is disabled by setting the CFDRFCCn.RFE bit to 0
- The CAN-FD module is in GL_RESET mode.

The RFEMP bit is cleared automatically when the first message is stored in the RX FIFO buffer.

RFFLL bit (RX FIFO Full)

The RFFLL bit is set automatically when the number of CAN messages stored in the FIFO buffer matches the configured FIFO depth.

The RFFLL is cleared automatically when:

- The number of CAN messages stored in the FIFO buffer is less than the configured FIFO depth
- RX FIFO is disabled by setting the CFDRFCCn.RFE bit to 0
- The CAN-FD module is in GL_RESET mode.

RFMLT bit (RX FIFO Message Lost)

Only write to the RFMLT bit when CAN-FD module is in GL_HALT or GL_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically whenever a message is lost due to attempted storage when the FIFO buffer is already full. If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode.

RFIF bit (RX FIFO Interrupt Flag)

The RFIF bit is set automatically when the configured interrupt condition is satisfied. This bit is not automatically cleared when the RX FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then this bit is set.

The bit is cleared by writing 0 to it. The bit is also cleared when CAN-FD module is in GL_RESET mode.

RFMC[7:0] bits (RX FIFO Message Count)

The RFMC[7:0] bits indicate the number of CAN messages stored in the RX FIFO buffer that can be read by the CPU.

These bits are cleared automatically when the FIFO is disabled and when the CAN-FD module is in GL_RESET mode.

RFFIF bit (RX FIFO Full Interrupt Flag)

The RFFIF bit is not cleared automatically when the RX FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the FIFO full interrupt condition is satisfied. If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The bit is cleared by writing 0 to it.

The bit is also cleared when the CAN-FD module is in GL_RESET mode.

33.2.27 CFDRFPCTRn : RX FIFO Pointer Control Register n (n = 0 to 7)

Base address: CANFD = 0x8002_0000

Offset address: 0x0100 + 0x04 × n

Bit position:	31																								7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RFPC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RFPC[7:0]	RX FIFO Pointer Control Increments read pointer of the corresponding RX FIFO buffers	R/W
31:8	—	The write value should be 0.	R/W

The CFDRFPCTRn register (n = 0 to 7) can be used to increment the read pointer of the corresponding RX FIFO buffers.

RFPC[7:0] bits (RX FIFO Pointer Control)

When the value 0xFF is written to the RFPC bits, the pointer of the corresponding RX FIFO buffer is moved to the next FIFO entry. Only write 0xFF to these registers when the corresponding RX FIFO buffer is enabled and not empty.

The read value from these bits is always 0x00.

Only write to these bits when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

Do not write to the RX FIFO Pointer Control registers when DMA is enabled.

33.2.28 CFDCFCCn : Common FIFO Configuration/Control Register n (n = 0 to 5)

Base address: CANFD = 0x8002_0000

Offset address: 0x0120 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CFITT[7:0]								CFDC[2:0]			CFTML[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CFIGCV[2:0]			CFIM	CFITR	CFITSS	CFM[1:0]		—	CFPLS[2:0]			—	CFTXIE	CFRXIE	CFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CFE	Common FIFO Enable 0: FIFO disabled 1: FIFO enabled	R/W
1	CFRXIE	Common FIFO RX Interrupt Enable 0: FIFO interrupt generation disabled for Frame RX 1: FIFO interrupt generation enabled for Frame RX	R/W
2	CFTXIE	Common FIFO TX Interrupt Enable 0: FIFO interrupt generation disabled for Frame TX 1: FIFO interrupt generation enabled for Frame TX	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
6:4	CFPLS[2:0]*1	Common FIFO Payload Data Size Configuration 0 0 0: 8 bytes 0 0 1: 12 bytes 0 1 0: 16 bytes 0 1 1: 20 bytes 1 0 0: 24 bytes 1 0 1: 32 bytes 1 1 0: 48 bytes 1 1 1: 64 bytes	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
9:8	CFM[1:0]	Common FIFO Mode 0 0: RX FIFO mode 0 1: TX FIFO mode 1 0: CAN – CAN GW FIFO mode 1 1: Reserved	R/W
10	CFITSS	Common FIFO Interval Timer Source Select 0: Reference clock (× 1 / × 10 period) 1: Bit time clock of related channel (FIFO is linked to fixed channel)	R/W
11	CFITR	Common FIFO Interval Timer Resolution 0: Reference clock period × 1 1: Reference clock period × 10	R/W

Bit	Symbol	Function	R/W
12	CFIM	Common FIFO Interrupt Mode 0: RX FIFO mode: RX interrupt generated when Common FIFO counter reaches CFIGCV value from a lower value TX FIFO mode: TX interrupt generated when Common FIFO transmits the last message successfully GW FIFO mode: For RX interrupt flag: Interrupt generated when FIFO counter increments and reaches the value configured in CFIGCV For TX interrupt flag: Interrupt generated when FIFO transmits the last message successfully 1: RX FIFO mode: RX interrupt generated at the end of every received message storage TX FIFO mode: TX interrupt generated for every successfully transmitted message GW FIFO mode: For RX interrupt flag: Interrupt generated when a message is stored in the FIFO For TX interrupt flag: Interrupt generated when a message is successfully transmitted from the FIFO	R/W
15:13	CFIGCV[2:0]	Common FIFO Interrupt Generation Counter Value 0 0 0: Interrupt generated when FIFO is 1/8th full 0 0 1: Interrupt generated when FIFO is 1/4th full 0 1 0: Interrupt generated when FIFO is 3/8th full 0 1 1: Interrupt generated when FIFO is 1/2 full 1 0 0: Interrupt generated when FIFO is 5/8th full 1 0 1: Interrupt generated when FIFO is 3/4th full 1 1 0: Interrupt generated when FIFO is 7/8th full 1 1 1: Interrupt generated when FIFO is full	R/W
20:16	CFTML[4:0]	Common FIFO TX Message Buffer Link Transmission scan link position of the corresponding channel	R/W
23:21	CFDC[2:0]	Common FIFO Depth Configuration 0 0 0: FIFO Depth = 0 messages 0 0 1: FIFO Depth = 4 messages 0 1 0: FIFO Depth = 8 messages 0 1 1: FIFO Depth = 16 messages 1 0 0: FIFO Depth = 32 messages 1 0 1: FIFO Depth = 48 messages 1 1 0: FIFO Depth = 64 messages 1 1 1: FIFO Depth = 128 messages	R/W
31:24	CFITT[7:0]	Common FIFO Interval Transmission Time Delay the start of transmission from the FIFO if configured in TX or GW mode, delay is a multiple of basic Interval Timer Clock Source unit	R/W

Note 1. These bits are not available in the classical CAN function.

The CFDCFCCn register (n = 0 to 5) is used to configure the Common FIFOs.

CFE bit (Common FIFO Enable)

The CFE bit enables the FIFO when set. FIFO is disabled when this bit is cleared.

This bit can also be used, by clearing it, to abort transmission from Common FIFO when configured in TX mode or GW mode, or to stop reception into the Common FIFO in RX mode.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFOs configured as TX or GW FIFO.

This bit can only be set if the configured FIFO depth is greater than 0 (CFDC bit > 0).

Set the CFE bit with a separate write access to the CFDCFCCn register, after all the other bits in this register are set.

This bit is cleared automatically when the CAN-FD module is in GL_RESET mode.

This bit is also cleared automatically when the related channel is in CH_RESET mode if the FIFO is configured in TX or GW mode.

CFRXIE bit (Common FIFO RX Interrupt Enable)

The CFRXIE bit enables generation of FIFO interrupts when the interrupt flag is set after reception of a frame in the corresponding FIFO buffer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

CFTXIE bit (Common FIFO TX Interrupt Enable)

The CFTXIE bit enables generation of common FIFO interrupts when the interrupt flag is set after transmission of a frame from the corresponding FIFO buffer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

CFPLS[2:0] bits (Common FIFO Payload Data Size Configuration)

The CFPLS[2:0] bits define the message data payload allocation in the RAM. This is the maximum number of bytes which can be received or transmitted by the FIFO buffer.

For details, see section 33.6. FIFO Buffers and Normal Message Buffer Configuration.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

Note: These bits are not available in the classical CAN function.

CFM[1:0] bits (Common FIFO Mode)

The CFM[1:0] bits select the mode of the FIFO. When a hardware reset is applied, all the Common FIFO buffers are configured in RX FIFO mode. Do not configure these bits to 11b.

Do not write to these bits in GL_OPERATION or GL_SLEEP mode.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

CFITSS bit (Common FIFO Interval Timer Source Select)

The CFITSS bit selects the basic clock source for the Interval Transmission Timer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode. In addition, do not write to this bit when the CFE bit is set to 1.

Do not write 1 to this bit when CAN-FD communication is used.*1

Note: The bit time clock can vary depending on the nominal and data rate bit configuration.

Note 1. This feature is not available in the classical CAN function.

CFITR bit (Common FIFO Interval Timer Resolution)

The CFITR bit selects the resolution of the reference clock for the Interval Transmission Timer (peripheral clock is the source for the reference clock).

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode. Also, do not write to this bit when the CFE bit is set to 1.

CFIM bit (Common FIFO Interrupt Mode)

The CFIM bit selects the interrupt generation condition for the FIFO buffer.

Do not write to this bit in GL_SLEEP mode.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

CFIGCV[2:0] bits (Common FIFO Interrupt Generation Counter Value)

The CFIGCV[2:0] bits select the message counter value for the generation of FIFO interrupts. These values represent fractions of the FIFO depth at which the interrupt is to be generated.

Do not write to these bits when the CAN-FD module is in GL_SLEEP mode.

The setting of these bits should be synchronized with the CFDC[2:0] bits.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

CFTML[4:0] bits (Common FIFO TX Message Buffer Link)

The CFTML[4:0] bits select the normal transmit message buffer position where the TX or GW FIFO is linked to, for transmission scanning.

Do not write to these bits in GL_OPERATION or GL_SLEEP mode.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

CFDC[2:0] bits (Common FIFO Depth Configuration)

The CFDC[2:0] bits select the depth of the common FIFO in terms of the number of messages. If the FIFO depth is configured to 0 message, the FIFO cannot be used.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

CFITT[7:0] bits (Common FIFO Interval Transmission Time)

The CFITT[7:0] bits select the delay in the start of transmission for all messages transmitted from this FIFO buffer when configured in TX or GW mode. The delay is a multiple of the basic interval timer clock source period (reference clock × 1, reference clock × 10, or bit time clock of the related CAN channel).

Do not write to these bits when the CAN-FD module is in GL_SLEEP mode.

Do not write to these bits when the CFE bit is set to 1.

When CFDGCFG.ITRCP[15:0] = 0x0000, set the CFITT[7:0] bits to 0x0000.

33.2.29 CFDCFCCEn : Common FIFO Configuration/Control Enhancement Register n (n = 0 to 5)

Base address: CANFD = 0x8002_0000

Offset address: 0x0180 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFBME
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CFMOWM	—	—	—	—	—	CFOFTXIE	CFOFRXIE	CFFIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CFFIE	Common FIFO Full Interrupt Enable 0: FIFO Interrupt generation disabled 1: FIFO Interrupt generation enabled	R/W
1	CFOFRXIE	Common FIFO One Frame Reception Interrupt Enable 0: One Frame RX interrupt generation disabled 1: One Frame RX interrupt generation enabled	R/W
2	CFOFTXIE	Common FIFO One Frame Transmission Interrupt Enable 0: One Frame TX interrupt generation disabled 1: One Frame TX interrupt generation enabled	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
8	CFMOWM	Common FIFO Message Overwrite Mode 0: Message discarded mode 1: Message overwrite mode	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
16	CFBME	Common FIFO Buffering Mode Enable 0: Transmission from Common FIFO 1: Transmission halt from Common FIFO	R/W

Bit	Symbol	Function	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

The CFDCFCCEn register (n = 0 to 5) is used to configure the Common FIFOs.

CFFIE bit (Common FIFO Full Interrupt Enable)

The CFFIE bit enables generation of the FIFO full interrupt when the interrupt flag is set after reception of a frame in the corresponding FIFO buffer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

The following content shows examples of interruptions:

1. Interruption output in number of arbitrary stages (CFDCFCCn.CFIGCV).
2. Interruption output in FIFO full state.

Management of the receiving data of FIFO can be performed by these notices of interruption.

CFOFRXIE bit (Common FIFO One Frame Reception Interrupt Enable)

The CFOFRXIE bit enables generation of the one frame reception interrupt when the interrupt flag is set after reception of a frame in the corresponding FIFO buffer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

CFOFTXIE bit (Common FIFO One Frame Transmission Interrupt Enable)

The CFOFTXIE bit enables generation of the one frame transmission interrupt when the interrupt flag is set after transmission of a frame in the corresponding FIFO buffer.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

CFMOWM bit (Common FIFO Message Overwrite Mode)

When the CFMOWM bit is 0, a receiving message is discarded and FIFO is full. When the CFMOWM bit is 1, a receiving message is overwritten and FIFO is full.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode.

Only write 1 to this bit when the common FIFO is in GW mode.

Do not write 1 to this bit when the CFE bit is 1.

CFBME bit (Common FIFO Buffering Mode Enable)

When the CFBME bit is 0, messages are transmitted from FIFO. When the CFBME bit is 1, messages are not transmitted from FIFO.

Do not write to this bit when the CAN-FD module is in GL_SLEEP mode. Additionally, do not write 1 to this bit when the CFE bit is 1.

33.2.30 CFDCFSTSn : Common FIFO Status Register n (n = 0 to 5)

Base address: CANFD = 0x8002_0000

Offset address: 0x01E0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	CFMOW	—	—	—	—	—	CFOFTXIF	CFOFRXIF	CFFIF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CFMC[7:0]								—	—	—	CFTXIF	CFRXIF	CFMLT	CFFLL	CFEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	CFEMP	Common FIFO Empty 0: FIFO not empty 1: FIFO empty	R
1	CFFLL	Common FIFO Full 0: FIFO not full 1: FIFO full	R
2	CFMLT	Common FIFO Message Lost 0: FIFO message not lost 1: FIFO message lost	R/W
3	CFRXIF	Common RX FIFO Interrupt Flag 0: FIFO interrupt condition not satisfied after frame reception 1: FIFO interrupt condition satisfied after frame reception	R/W
4	CFTXIF	Common TX FIFO Interrupt Flag 0: FIFO interrupt condition not satisfied after frame transmission 1: FIFO Interrupt condition satisfied after frame transmission	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
15:8	CFMC[7:0]	Common FIFO Message Count Number of messages stored in FIFO	R
16	CFFIF	Common FIFO Full Interrupt Flag 0: Interrupt condition not satisfied for FIFO full interrupt 1: Interrupt condition satisfied for FIFO full interrupt	R/W
17	CFOFRXIF	Common FIFO One Frame Reception Interrupt Flag For each FIFO that receives a frame, a corresponding interrupt is set.	R/W
18	CFOFTXIF	Common FIFO One Frame Transmission Interrupt Flag For each FIFO that transmits a frame, a corresponding interrupt is set.	R/W
23:19	—	These bits are read as 0. The write value should be 0.	R/W
24	CFMOW	Common FIFO Message Overwrite 0: No message overwrite occurred in FIFO 1: Message overwrite occurred in FIFO	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The CFDCFSTSn register (n = 0 to 5) shows the status of messages stored in the corresponding FIFO buffers.

CFEMP bit (Common FIFO Empty)

The CFEMP bit is set automatically when:

- The CPU has read all messages from the FIFO configured in RX mode
- All messages have been transmitted from the FIFO configured in TX or GW mode
- The FIFO is disabled by setting the CFE bit to 0
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET when FIFO configured in TX or GW mode.

The CFEMP bit is cleared automatically when:

- The first reception message is stored in the FIFO buffer when configured in RX mode
- The first message to be transmitted is stored in the FIFO buffer when configured in TX or GW mode.

CFFLL bit (Common FIFO Full)

The CFFLL bit is set automatically when the number of CAN messages stored in the FIFO matches the configured FIFO depth.

The CFFLL bit is cleared automatically when:

- The number of CAN messages stored in the FIFO is less than the configured FIFO depth
- The FIFO is disabled by setting the CFE bit to 0

- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode when FIFO buffer is configured in TX or GW mode.

CFMLT bit (Common FIFO Message Lost)

The CFMLT bit is set automatically whenever a message is lost due to attempted storage of a new message when FIFO is already full in RX or GW mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then this bit is set.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFO configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

The CFMLT bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

CFRXIF bit (Common RX FIFO Interrupt Flag)

The CFRXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFO configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the configured interrupt condition is satisfied for Common FIFO buffers when configured in GW mode or RX mode.

The CFRXIF bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in GW mode.

CFTXIF bit (Common TX FIFO Interrupt Flag)

The CFTXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFO buffer configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the configured interrupt condition is satisfied for Common FIFO buffers configured in GW or TX mode.

The CFTXIF bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

CFMC[7:0] bits (Common FIFO Message Count)

The CFMC[7:0] bits indicate the following:

- Number of CAN messages stored by the CPU in the FIFO buffer configured in TX mode pending for transmission
- Number of CAN messages stored in the FIFO buffer configured in RX mode by CAN-FD to be read by the CPU
- Number of CAN messages stored by the CAN-FD in the GW FIFO pending for transmission.

The CFMC[7:0] bits are cleared automatically when:

- The FIFO is disabled
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

CFFIF bit (Common FIFO Full Interrupt Flag)

The CFFIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFO configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the FIFO Full Interrupt condition is satisfied for Common FIFO buffers when configured in GW or RX mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The CFFIF bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

CFOFRXIF bit (Common FIFO One Frame Reception Interrupt Flag)

The CFOFRXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for the FIFO buffer configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the One Frame Reception Interrupt condition is satisfied for the Common FIFO buffers when configured in GW or RX mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The CFOFRXIF bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

CFOFTXIF bit (Common FIFO One Frame Transmission Interrupt Flag)

The CFOFTXIF bit is not cleared automatically if the Common FIFO buffer is disabled.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFOs configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is set automatically when the One Frame Transmission Interrupt condition is satisfied for Common FIFO buffers configured in GW mode or TX mode.

If a set from the CAN channel occurs simultaneously with a clear by a write access, then the bit is set.

The bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO is configured in TX or GW mode.

CFMOW bit (Common FIFO Message Overwrite)

The CFMOW bit is set automatically whenever a message is an overwrite storage of a new message when CFDCFCCEn.CFMOWM = 1 and FIFO is already full in GW mode.

If the set from the CAN channel occurs simultaneously with the clear by a write access, the bit is set.

Only write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode and the related CAN-FD channel is not in CH_RESET mode for FIFO buffer configured as TX or GW FIFO. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

The CFMOW bit is cleared:

- By writing 0 to it
- When the CAN-FD module is in GL_RESET mode
- When the related CAN-FD channel is in CH_RESET mode if the FIFO buffer is configured in TX or GW mode.

33.2.31 CFDCFPCTRn : Common FIFO Pointer Control Register n (n = 0 to 5)

Base address: CANFD = 0x8002_0000

Offset address: 0x0240 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFPC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	CFPC[7:0]	Common FIFO Pointer Control Increments read or write pointer of the corresponding Common FIFO buffers depending on the mode configuration.	W
31:8	—	The write value should be 0.	W

The CFDCFPCTRn register (n = 0 to 5) can be used to increment the read or write pointer of the corresponding Common FIFO buffer.

CFPC[7:0] bits (Common FIFO Pointer Control)

When the value 0xFF is written into the CFPC[7:0] bits, the read pointer of the corresponding Common FIFO buffer (when configured in RX mode), or the write pointer of the corresponding Common FIFO buffer (when configured in TX mode) moves to the next FIFO entry.

The read value from these bits is always 0x00.

Only write to these bits when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

Only write 0xFF to this register when:

- The Common FIFO buffer is enabled and is not empty if configured in RX mode
- The Common FIFO buffer is enabled and is not full if configured in TX mode
- The Common FIFO buffer is enabled and is not configured in GW mode.

Do not write to the Common FIFO Pointer Control registers when DMA is enabled.

33.2.32 CFDFESTS : FIFO Empty Status Register

Base address: CANFD = 0x8002_0000

Offset address: 0x02A0

Bit position:	31																	13					8	7						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFXEMP[5:0]						RFXEMP[7:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	RFXEMP[7:0]	RX FIFO Empty Status 0: Corresponding FIFO not empty 1: Corresponding FIFO empty	R
13:8	CFXEMP[5:0]	Common FIFO Empty Status 0: Corresponding FIFO not empty 1: Corresponding FIFO empty	R
31:14	—	These bits are read as 0.	R

The CFDFESTS register shows status of the empty bits of the FIFO buffers.

RFXEMP[7:0] bits (RX FIFO Empty Status)

Bit [7] (RFXEMP[7]) is associated with FIFO index 7 and bit [0] (RFXEMP[0]) is associated with FIFO index 0.

The RFXEMP[7:0] bits are set when the CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

CFXEMP[5:0] bits (Common FIFO Empty Status)

Bit [13] (CFXEMP[5]) is associated with common FIFO index 5 and bit [8] (CFXEMP[0]) is associated with common FIFO index 0.

The CFXEMP[5:0] bits are set when the CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

33.2.33 CFDFFSTS : FIFO Full Status Register

Base address: CANFD = 0x8002_0000

Offset address: 0x02A4

Bit position:	31																	13					8	7						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFXFLL[5:0]						RFXFLL[7:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RFXFLL[7:0]	RX FIFO Full Status 0: Corresponding FIFO not full 1: Corresponding FIFO full	R
13:8	CFXFLL[5:0]	Common FIFO Full Status 0: Corresponding FIFO not full 1: Corresponding FIFO full	R
31:14	—	These bits are read as 0.	R

The CFDFFSTS register shows status of the full bits of the FIFO buffers.

RFXFLL[7:0] bits (RX FIFO Full Status)

The RFXFLL[7:0] bits are cleared when CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

CFXFLL[5:0] bits (Common FIFO Full Status)

The CFXFLL[5:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

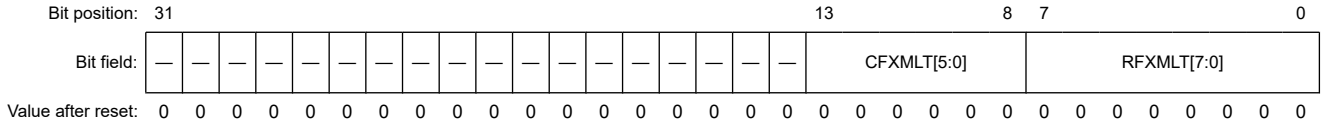
Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

33.2.34 CFDFMSTS : FIFO Message Lost Status Register

Base address: CANFD = 0x8002_0000

Offset address: 0x02A8



Bit	Symbol	Function	R/W
7:0	RFXMLT[7:0]	RX FIFO Message Lost Status 0: Corresponding FIFO Message Lost flag not set 1: Corresponding FIFO Message Lost flag set	R
13:8	CFXMLT[5:0]	Common FIFO Message Lost Status 0: Corresponding FIFO Message Lost flag not set 1: Corresponding FIFO Message Lost flag set	R
31:14	—	These bits are read as 0.	R

The CFDFMSTS register shows status of the message lost bits of the FIFO buffers.

RFXMLT[7:0] bits (RX FIFO Message Lost Status)

The RFXMLT[7:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

CFXMLT[5:0] bits (Common FIFO Message Lost Status)

The CFXMLT[5:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

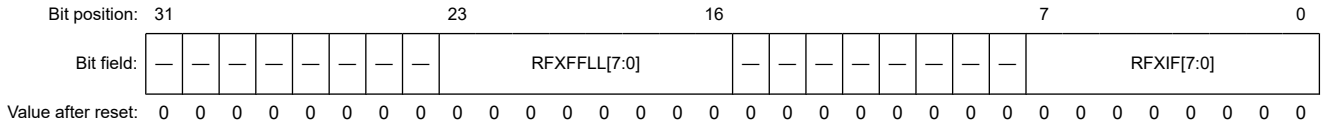
Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

33.2.35 CFDRFISTS : RX FIFO Interrupt Flag Status Register

Base address: CANFD = 0x8002_0000

Offset address: 0x02AC



Bit	Symbol	Function	R/W
7:0	RFXIF[7:0]	RX FIFO[x] Interrupt Flag Status 0: Corresponding RX FIFO Interrupt flag not set 1: Corresponding RX FIFO Interrupt flag set	R
15:8	—	These bits are read as 0.	R
23:16	RFXFFLL[7:0]	RX FIFO[x] Interrupt Full Flag Status 0: Corresponding RX FIFO Interrupt Full flag not set 1: Corresponding RX FIFO Interrupt Full flag set	R
31:24	—	These bits are read as 0.	R

The CFDRFISTS register shows status of the interrupt flag bits of the RX FIFO buffers.

RFXIF[7:0] bits (RX FIFO[x] Interrupt Flag Status)

Each bit is set automatically when the corresponding interrupt flag bit is set in the RX FIFO Status Registers.

The RFXIF[7:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding interrupt flag bit is cleared in the RX FIFO Status Registers.

RFXFFLL[7:0] bits (RX FIFO[x] Interrupt Full Flag Status)

Each bit is set automatically when the corresponding interrupt full flag bit is set in the RX FIFO Status Registers.

The RFXFFLL[7:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding interrupt full flag bit is cleared in the RX FIFO Status Registers.

33.2.36 CFDCFRISTS : Common FIFO RX Interrupt Flag Status Register

Base address: CANFD = 0x8002_0000

Offset address: 0x02B0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFXRXIF[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	CFXRXIF[5:0]	Common FIFO RX Interrupt Flag Status 0: Corresponding Common FIFO RX Interrupt flag not set 1: Corresponding Common FIFO RX Interrupt flag set	R
31:6	—	These bits are read as 0.	R

The CFDCFRISTS register shows status of the interrupt flag bits of the Common FIFO buffers.

CFXRXIF[5:0] bits (Common FIFO RX Interrupt Flag Status)

Each bit is set automatically when the corresponding RX interrupt flag bit is set in the Common FIFO Status Registers.

The CFXRXIF[5:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding RX interrupt flag bit is cleared in the Common FIFO Status Registers.

33.2.37 CFDCFTISTS : Common FIFO TX Interrupt Flag Status Register

Base address: CANFD = 0x8002_0000

Offset address: 0x02B4

Bit position:	31																								5	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFXTXIF[5:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	CFXTXIF[5:0]	Common FIFO TX Interrupt Flag Status 0: Corresponding Common FIFO TX Interrupt flag not set 1: Corresponding Common FIFO TX Interrupt flag set	R
31:6	—	These bits are read as 0.	R

The CFDCFTISTS register shows status of the interrupt flag bits of the Common FIFO buffers.

CFXTXIF[5:0] bits (Common FIFO TX Interrupt Flag Status)

Each bit is set automatically when the corresponding TX interrupt flag bit is set in the Common FIFO Status Registers.

The CFXTXIF[5:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding TX interrupt flag bit is cleared in the Common FIFO Status Registers.

33.2.38 CFDFFFSTS : FIFO FDC Full Status Register

Base address: CANFD = 0x8002_0000

Offset address: 0x02C4

Bit position:	31																	13					8	7						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFXFFLL[5:0]						RFXFFLL[7:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RFXFFLL[7:0]	RX FIFO FDC Level Full Status 0: Corresponding FIFO full interrupt not set 1: Corresponding FIFO full interrupt set	R
13:8	CFXFFLL[5:0]	COMMON FIFO FDC Level Full Status 0: Corresponding FIFO full interrupt not set 1: Corresponding FIFO full interrupt set	R
31:14	—	These bits are read as 0.	R

The CFDFFFSTS register shows status of the full interrupt flag bits of the FIFO buffers.

RFXFFLL[7:0] bits (RX FIFO FDC Level Full Status)

Each bit is set automatically when the corresponding bit is set in the RX FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the RX FIFO Status Registers.

The RFXFFLL[7:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

CFXFFLL[5:0] bits (COMMON FIFO FDC Level Full Status)

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

The CFXFFLL[5:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

33.2.39 CFDCFMOWSTS : Common FIFO Message Overwrite Status Register

Base address: CANFD = 0x8002_0000

Offset address: 0x02C0

Bit position:	31																								5	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFXMOW[5:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	CFXMOW[5:0]	Common FIFO Massage Overwrite Status 0: Corresponding FIFO Overwrite flag not set 1: Corresponding FIFO Overwrite flag set	R
31:6	—	These bits are read as 0.	R

The CFDCFMOWSTS register shows the status of the Interrupt Flag bits of the Common FIFO Buffers.

CFXMOW[5:0] bits (Common FIFO Massage Overwrite Status)

The CFXMOW[5:0] bits are cleared when the CAN-FD module is in GL_RESET mode. This register is only valid in GW mode.

Each bit is set automatically when the corresponding bit is set in the Common FIFO Status Registers.

Each bit is cleared automatically when the corresponding bit is cleared in the Common FIFO Status Registers.

33.2.40 CFDCFOFRISTS : Common FIFO One Frame RX Interrupt Flag Status Register

Base address: CANFD = 0x8002_0000

Offset address: 0x02B8

Bit position:	31																								5	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFXOFRXIF[5:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	CFXOFRXIF[5:0]	Common FIFO One Frame RX Interrupt Flag Status 0: Corresponding Common FIFO One Frame RX Interrupt flag not set 1: Corresponding Common FIFO One Frame RX Interrupt flag set	R
31:6	—	These bits are read as 0.	R

The CFDCFOFRISTS register shows status of the interrupt flag bits of the Common FIFO buffers.

CFXOFRXIF[5:0] bits (Common FIFO One Frame RX Interrupt Flag Status)

Each bit is set automatically when the corresponding One Frame RX Interrupt flag bit is set in the Common FIFO Status Registers.

The CFXOFRXIF[5:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding One Frame RX Interrupt flag bit is cleared in the Common FIFO Status Registers.

33.2.41 CFDCFOFTISTS : Common FIFO One Frame TX Interrupt Flag Status Register

Base address: CANFD = 0x8002_0000

Offset address: 0x02BC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFXOFTXIF[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	CFXOFTXIF[5:0]	Common FIFO One Frame TX Interrupt Flag Status 0: Corresponding Common FIFO One Frame TX Interrupt flag not set 1: Corresponding Common FIFO One Frame TX Interrupt flag set	R
31:6	—	These bits are read as 0.	R

The CFDCFOFTISTS register shows status of the interrupt flag bits of the Common FIFO buffers.

CFXOFTXIF[5:0] bits (Common FIFO One Frame TX Interrupt Flag Status)

Each bit is set automatically when the corresponding One Frame TX Interrupt flag bit is set in the Common FIFO Status Registers.

The CFXOFTXIF[5:0] bits are cleared when the CAN-FD module is in GL_RESET mode.

Each bit is cleared automatically when the corresponding One Frame TX Interrupt flag bit is cleared in the Common FIFO Status Registers.

33.2.42 CFDCDTCT : DMA Transfer Control Register

Base address: CANFD = 0x8002_0000

Offset address: 0x1330

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CFDMAE1	CFDMAE0	RFDMAE7	RFDMAE6	RFDMAE5	RFDMAE4	RFDMAE3	RFDMAE2	RFDMAE1	RFDMAE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFDMAE0	DMA Transfer Enable for RX FIFO 0 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
1	RFDMAE1	DMA Transfer Enable for RX FIFO 1 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
2	RFDMAE2	DMA Transfer Enable for RX FIFO 2 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
3	RFDMAE3	DMA Transfer Enable for RX FIFO 3 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W

Bit	Symbol	Function	R/W
4	RFDMAE4	DMA Transfer Enable for RX FIFO 4 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
5	RFDMAE5	DMA Transfer Enable for RX FIFO 5 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
6	RFDMAE6	DMA Transfer Enable for RX FIFO 6 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
7	RFDMAE7	DMA Transfer Enable for RX FIFO 7 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
8	CFDMAE0	DMA Transfer Enable for Common FIFO 0 of Channel 0 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
9	CFDMAE1	DMA Transfer Enable for Common FIFO 0 of Channel 1 0: DMA transfer request disabled 1: DMA transfer request enabled	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

The CFDCDTCT register controls the start and stop of DMA transfer operation.

RFDMAEn (n = 0 to 7) bit (DMA Transfer Enable for RX FIFOn)

Number of RX FIFOs = 8

The RFDMAEn bit cannot be set in GL_SLEEP or GL_RESET mode.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

CFDMAEn (n = 0, 1) bit (DMA Transfer Enable for Common FIFO 0 of Channel 0, 1)

The CFDMAEn bit enables or disables DMA transfer request for common FIFO 0 of channel 0 or 1. Only Common FIFO 0 can be linked to a DMA channel with this bit. Common FIFO 1 cannot be linked to a DMA channel.

To link Common FIFO 2, see bit CFDCDTTCT.CFDMAEn in section 33.2.44. CFDCDTTCT : DMA TX Transfer Control Register.

The CFDMAEn bit cannot be set in GL_SLEEP or GL_RESET mode.

Do not enable a DMA transfer for a Common FIFO that is configured as TX or GW FIFO.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

33.2.43 CFDCDTSTS : DMA Transfer Status Register

Base address: CANFD = 0x8002_0000

Offset address: 0x1334

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CFDMASTS1	CFDMASTS0	RFDMASTS7	RFDMASTS6	RFDMASTS5	RFDMASTS4	RFDMASTS3	RFDMASTS2	RFDMASTS1	RFDMASTS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFDMASTS0	DMA Transfer Status for RX FIFO 0 0: DMA transfer stopped 1: DMA transfer on going	R
1	RFDMASTS1	DMA Transfer Status for RX FIFO 1 0: DMA transfer stopped 1: DMA transfer on going	R
2	RFDMASTS2	DMA Transfer Status for RX FIFO 2 0: DMA transfer stopped 1: DMA transfer on going	R
3	RFDMASTS3	DMA Transfer Status for RX FIFO 3 0: DMA transfer stopped 1: DMA transfer on going	R
4	RFDMASTS4	DMA Transfer Status for RX FIFO 4 0: DMA transfer stopped 1: DMA transfer on going	R
5	RFDMASTS5	DMA Transfer Status for RX FIFO 5 0: DMA transfer stopped 1: DMA transfer on going	R
6	RFDMASTS6	DMA Transfer Status for RX FIFO 6 0: DMA transfer stopped 1: DMA transfer on going	R
7	RFDMASTS7	DMA Transfer Status for RX FIFO 7 0: DMA transfer stopped 1: DMA transfer on going	R
8	CFDMASTS0	DMA Transfer Status only for Common FIFO 0 of Channel 0 0: DMA transfer stopped 1: DMA transfer on going	R
9	CFDMASTS1	DMA Transfer Status only for Common FIFO 0 of Channel 1 0: DMA transfer stopped 1: DMA transfer on going	R
31:10	—	These bits are read as 0.	R

The CFDCDTSTS register shows the status of the DMA transfer.

RFDMASTSn (n = 0 to 7) bit (DMA Transfer Status for RX FIFO n)

Number of RX FIFOs = 8

Each bit is set automatically when the corresponding DMA enable bit is set and the corresponding DMA FIFO is not empty.

Each bit is cleared automatically when the DMA transfer stops either because the DMA is disabled or the DMA FIFO is empty.

When CFDCDTCT.RFDMAEn (see CFDCDTCT.RFDMAEn bit in section 33.2.42. CFDCDTCT : DMA Transfer Control Register) is set to 0 while DMA transfer for the corresponding FIFO is on going, the RFDMASTSn bit becomes 0 when the DMA transfer is complete.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

CFDMASTSn bit (DMA Transfer Status only for Common FIFO 0 of Channel 0, 1)

Each bit is set automatically when the corresponding DMA enable bit is set and the corresponding DMA FIFO is not empty.

Each bit is cleared automatically when the DMA transfer stops either because the DMA is disabled or the DMA FIFO is empty.

When CFDCDTCT.CFDMAEn (see CFDCDTCT.CFDMAEn bit in section 33.2.42. CFDCDTCT : DMA Transfer Control Register) is set to 0 while DMA transfer for the corresponding FIFO is on going, the CFDMASTSn bit becomes 0 when the DMA transfer is complete.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

33.2.44 CFDCDTTCT : DMA TX Transfer Control Register

Base address: CANFD = 0x8002_0000

Offset address: 0x1340

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFDMAE1	CFDMAE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TQ3DMAE1	TQ3DMAE0	—	—	—	—	—	—	TQ0DMAE1	TQ0DMAE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TQ0DMAE0	DMA TX Transfer Enable for TXQ 0 of Channel 0 0: DMA TX transfer request disabled 1: DMA TX transfer request enabled	R/W
1	TQ0DMAE1	DMA TX Transfer Enable for TXQ 0 of Channel 1 0: DMA TX transfer request disabled 1: DMA TX transfer request enabled	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
8	TQ3DMAE0	DMA TX Transfer Enable for TXQ 3 of Channel 0 0: DMA TX transfer request disabled 1: DMA TX transfer request enabled	R/W
9	TQ3DMAE1	DMA TX Transfer Enable for TXQ 3 of Channel 1 0: DMA TX transfer request disabled 1: DMA TX transfer request enabled	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
16	CFDMAE0	DMA TX Transfer Enable for Common FIFO 2 of Channel 0 0: DMA TX transfer request disabled 1: DMA TX transfer request enabled	R/W
17	CFDMAE1	DMA TX Transfer Enable for Common FIFO 2 of Channel 1 0: DMA TX transfer request disabled 1: DMA TX transfer request enabled	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

The CFDCDTTCT register controls the start and stop of DMA transfer operation.

TQ0DMAEn (n = 0, 1) bit (DMA TX Transfer Enable for TXQ 0 of Channel n)

The TQ0DMAEn bit cannot be set in GL_SLEEP or GL_RESET mode.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

TQ3DMAEn (n = 0, 1) bit (DMA TX Transfer Enable for TXQ 3 of Channel n)

The TQ3DMAEn bit cannot be set in GL_SLEEP or GL_RESET mode.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

CFDMAEn (n = 0, 1) bit (DMA TX Transfer Enable for Common FIFO 2 of Channel n)

The CFDMAEn bit cannot be set in GL_SLEEP or GL_RESET mode.

Only common FIFO 2 can be linked to a DMA channel with this bit. Common FIFO 1 cannot be linked to a DMA channel.

To link Common FIFO 0, see CFDCDTCT.CFDMAEn bit in section 33.2.42. CFDCDTCT : DMA Transfer Control Register.

Do not enable a DMA transfer for a Common FIFO that is configured as RX or GW FIFO.

The CFDMAEn bit is cleared when the CAN-FD module is in GL_RESET mode.

33.2.45 CFDCDTTSTS : DMA TX Transfer Status Register

Base address: CANFD = 0x8002_0000

Offset address: 0x1344

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CFDMASTS1	CFDMASTS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TQ3DMASTS1	TQ3DMASTS0	—	—	—	—	—	—	TQ0DMASTS1	TQ0DMASTS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TQ0DMASTS0	DMA TX Transfer Status for TXQ0 of Channel 0 0: DMA TX transfer stopped 1: DMA TX transfer enabled	R
1	TQ0DMASTS1	DMA TX Transfer Status for TXQ0 of Channel 1 0: DMA TX transfer stopped 1: DMA TX transfer enabled	R
7:2	—	These bits are read as 0.	R
8	TQ3DMASTS0	DMA TX Transfer Status for TXQ3 of Channel 0 0: DMA TX transfer stopped 1: DMA TX transfer enabled	R
9	TQ3DMASTS1	DMA TX Transfer Status for TXQ3 of Channel 1 0: DMA TX transfer stopped 1: DMA TX transfer enabled	R
15:10	—	These bits are read as 0.	R
16	CFDMASTS0	DMA TX Transfer Status for Common FIFO 2 of Channel 0 0: DMA TX transfer stopped 1: DMA TX transfer enabled	R
17	CFDMASTS1	DMA TX Transfer Status for Common FIFO 2 of Channel 1 0: DMA TX transfer stopped 1: DMA TX transfer enabled	R
31:18	—	These bits are read as 0.	R

The CFDCDTTSTS register shows the status of the DMA TX transfer.

TQ0DMASTSn (n = 0, 1) bit (DMA TX Transfer Status for TXQ 0 of Channel n)

The TQ0DMASTSn bit is set when the CFDCDTTCT.TQ0DMAEn bit in the corresponding CFDCDTTCT register is set (see section 33.2.44. CFDCDTTCT : DMA TX Transfer Control Register).

This bit is cleared when:

- The CFDCDTTCT.TQ0DMAEn bit in the corresponding CFDCDTTCT register is cleared
- The CAN-FD module is in GL_RESET mode.

TQ3DMASTSn (n = 0, 1) bit (DMA TX Transfer Status for TXQ 3 of Channel n)

The TQ3DMASTSn bit is set when the CFDCDTTCT.TQ3DMAEn bit in the corresponding CFDCDTTCT register is set (see section 33.2.44. CFDCDTTCT : DMA TX Transfer Control Register).

This bit is cleared when:

- The CFDCDTTCT.TQ3DMAEn bit in the corresponding CFDCDTTCT register is cleared

- The CAN-FD module is in GL_RESET mode.

CFDMASTSn (n = 0, 1) bit (DMA TX Transfer Status only for Common FIFO 2 of Channel n)

The CFDMASTSn bit is set when the CFDCDTTCT.CFDMAEn bit in the corresponding CFDCDTTCT register is set (see section 33.2.44. CFDCDTTCT : DMA TX Transfer Control Register).

This bit is cleared when:

- The CFDCDTTCT.CFDMAEn bit in the corresponding CFDCDTTCT register is cleared
- The CAN-FD module is in GL_RESET mode.

33.2.46 CFDGRINTSTSn : Global RX Interrupt Status Register n (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x1350 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CFOFRIF[2:0]			—	CFRFIF[2:0]			—	—	—	—	—	CFRIF[2:0]			—	—	BQOFRIF[1:0]		—	QOFRIF[2:0]			—	BQFIF[1:0]			—	—	QFIF[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	QFIF[2:0]	TXQ Full Interrupt Flag Channel n (n = 0, 1) 0: Corresponding TXQ Full Interrupt flag is not set 1: Corresponding TXQ Full Interrupt flag is set	R
3	—	This bit is read as 0.	R
5:4	BQFIF[1:0]	Borrowed TXQ Full Interrupt Flag Channel n (n = 0, 1) 0: Corresponding TXQ Full Interrupt flag is not set 1: Corresponding TXQ Full Interrupt flag is set	R
7:6	—	These bits are read as 0.	R
10:8	QOFRIF[2:0]	TXQ One Frame RX Interrupt Flag Channel n (n = 0, 1) 0: Corresponding TXQ One Frame RX Interrupt flag is not set 1: Corresponding TXQ One Frame RX Interrupt flag is set	R
11	—	This bit is read as 0.	R
13:12	BQOFRIF[1:0]	Borrowed TXQ One Frame RX Interrupt Flag Channel n (n = 0, 1) 0: Corresponding TXQ One Frame RX Interrupt flag is not set 1: Corresponding TXQ One Frame RX Interrupt flag is set	R
15:14	—	These bits are read as 0.	R
18:16	CFRIF[2:0]	Common FIFO RX Interrupt Flag Channel n (n = 0, 1) 0: Corresponding Common FIFO RX Interrupt flag is not set 1: Corresponding Common FIFO RX Interrupt flag is set	R
23:19	—	These bits are read as 0.	R
26:24	CFRFIF[2:0]	Common FIFO FDC Level Full Interrupt Flag Channel n (n = 0, 1) 0: Corresponding Common FIFO Full Interrupt flag is not set 1: Corresponding Common FIFO Full Interrupt flag is set	R
27	—	This bit is read as 0.	R
30:28	CFOFRIF[2:0]	Common FIFO One Frame RX Interrupt Flag Channel n (n = 0, 1) 0: Corresponding Common FIFO One Frame RX Interrupt flag is not set 1: Corresponding Common FIFO One Frame RX Interrupt flag is set	R
31	—	This bit is read as 0.	R

QFIF[2:0] bits (TXQ Full Interrupt Flag Channel n (n = 0, 1))

The QFIF[2:0] bits are set automatically when the TXQ Full Interrupt flag of the related channel is set when the interrupt is enabled.

The QFIF[2:0] bits are cleared automatically when:

- The related TXQ result status bits are cleared or the interrupt enable is disabled
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

BQFIF[1:0] bits (Borrowed TXQ Full Interrupt Flag Channel n (n = 0, 1))

The BQFIF[1:0] bits are set when a flexible transmission buffer assignment function is used and the borrowed TXQ is in full status. Operation is the same as CFDGRINTSTSn.QFIF.

The bit of the channel that lends TXMB is a reserved bit.

QOFRIF[2:0] bits (TXQ One Frame RX Interrupt Flag Channel n (n = 0, 1))

The QOFRIF[2:0] bits are set automatically when the TXQ One Frame RX Interrupt flag of the related channel is set when the interrupt is enabled.

The QOFRIF[2:0] bits are cleared automatically when:

- The related TXQ result status bits are cleared or the interrupt enable is disabled
- The CAN-FD module is in GL_RESET or CH_RESET mode.

BQOFRIF[1:0] bits (Borrowed TXQ One Frame RX Interrupt Flag Channel n (n = 0, 1))

The BQOFRIF[1:0] bits are set when a flexible transmission buffer assignment function is used and the borrowed TXQ receives one frame. Operation is the same as CFDGRINTSTSn.QOFRIF.

The bit of the channel that lends TXMB is a reserved bit.

CFRIF[2:0] bits (Common FIFO RX Interrupt Flag Channel n (n = 0, 1))

The CFRIF[2:0] bits are set automatically when the Common FIFO RX Interrupt flag of the related channel is set when the interrupt is enabled.

The CFRIF[2:0] bits are cleared automatically when:

- The related Common FIFO RX result status bits are cleared or the interrupt enable is disabled
- The CAN-FD module is in GL_RESET or CH_RESET mode.

CFRFIF[2:0] bits (Common FIFO FDC Level Full Interrupt Flag Channel n (n = 0, 1))

The CFRFIF[2:0] bits are set automatically when the Common FIFO Full Interrupt flag of the related channel is set when the interrupt is enabled.

The CFRFIF[2:0] bits are cleared automatically when:

- The related Common FIFO RX result status bits are cleared or the interrupt enable is disabled
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

CFOFRIF[2:0] bits (Common FIFO One Frame RX Interrupt Flag Channel n (n = 0, 1))

The CFOFRIF[2:0] bits are set automatically when the Common FIFO One Frame RX Interrupt flag of the related channel is set when the interrupt is enabled.

The CFOFRIF[2:0] bits are cleared automatically when:

- The related Common FIFO RX result status bits are cleared or the interrupt enable is disabled
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

33.2.47 CFDTMCn : TX Message Buffer Control Register n (n = 0 to 127)

Base address: CANFD = 0x8002_0000

Offset address: 0x02D0 + 0x01 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TMOM	TMTAR	TMTR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMTR	TX Message Buffer Transmission Request 0: TX Message buffer transmission not requested 1: TX message buffer transmission requested	R/W
1	TMTAR	TX Message Buffer Transmission Abort Request 0: TX message buffer transmission request abort not requested 1: TX message buffer transmission request abort requested	R/W
2	TMOM	TX Message Buffer One-shot Mode 0: TX message buffer not configured in one-shot mode 1: TX message buffer configured in one-shot mode	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTMCn register (n = 0 to 127) configures the TX message buffer functions.

TMTR bit (TX Message Buffer Transmission Request)

When the TMTR bit is set, the CAN-FD module logic tries to transmit the message stored in the corresponding message buffer.

Only write to this bit when the related CAN-FD module is in CH_HALT or CH_OPERATION mode.

Do not set this bit if the corresponding TX message buffer is linked to a COM FIFO in TX or GW mode or is a part of TX Queue.

This bit cannot be directly cleared by a CPU write access.

This bit can only be set when the Transmission Result flag bits (CFDTMSTSn.TMTRF[1:0]) in the CFDTMSTSn register corresponding to the message buffer are cleared to 00b.

The TMTR bit is automatically cleared by the:

- CAN-FD module logic at the end of a successful transmission
- CAN-FD module logic at the end of a transmission abort, requested by the corresponding CFDTMCn.TMTAR bit
- CAN-FD module logic when there is a detection of a CAN bus error or arbitration loss if CFDTMCn.TMOM bit is set for the message buffer
- CAN-FD module logic when the CAN-FD module is in GL_RESET mode or the related channel is in CH_RESET mode.

TMTAR bit (TX Message Buffer Transmission Abort Request)

When the TMTAR bit is set, the CAN-FD module logic tries to abort the transmission of the frame stored in the corresponding message buffer.

In most cases, transmission cannot be aborted if the internal scan for transmission is complete and the message buffer has already been selected for transmission. In this case, frame may be transmitted successfully from the message buffer. The message buffer selection is released by entering CH_HALT mode.

However, message buffer selected for transmission can be aborted by an abort request when the CAN node detects a new message on the bus (RX pin) before it starts transmission from the selected message buffer.

Only write to the TMTAR bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode. This bit can only be set when the related transmit request TMTR bit is set.

The TMTAR bit cannot be cleared by a CPU write access. Clearing of this bit by CAN-FD has priority over setting by a CPU write access.

The TMTAR bit is automatically cleared by:

- The CAN-FD module logic at the end of a successful transmission
- The CAN-FD module logic at the end of a transmission abort
- The CAN-FD module logic when there is detection of a CAN bus error or arbitration loss
- The CAN-FD module logic when the CAN-FD module is in GL_RESET mode or the related channel enters CH_RESET mode.

TMOM bit (TX Message Buffer One-shot Mode)

When the TMOM bit is set, the CAN-FD module logic tries to transmit the message only once.

If the transmission is successful, the CFDTMSTSn.TMTRF[1:0] bits are set to 10b or 11b. Otherwise, the transmission is automatically aborted and CFDTMSTSn.TMTRF[1:0] bits are set to 01b due to a bus error or a bus arbitration lost.

The TMOM bit remains set if the transmission has completed successfully or aborted due to an error or a loss of arbitration.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

Set this bit at the same time as the TMTR bit. Clear this bit with a write access.

If a message has already been requested for transmission, do not write to this bit until the message has been successfully transmitted or transmission has been aborted.

The TMOM bit is automatically cleared by the CAN-FD module logic when the CAN-FD module is in GL_RESET mode or the related channel is in CH_RESET mode.

33.2.48 CFDTMSTSn : TX Message Buffer Status Register n (n = 0 to 127)

Base address: CANFD = 0x8002_0000

Offset address: 0x07D0 + 0x01 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TMTARM	TMTRM	TMTRF[1:0]		TMTSTS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMTSTS	TX Message Buffer Transmission Status 0: No on-going transmission 1: On-going transmission	R
2:1	TMTRF[1:0]	TX Message Buffer Transmission Result Flag 0 0: No result 0 1: Transmission aborted from the TX message buffer 1 0: Transmission successful from the TX message buffer and transmission abort was not requested 1 1: Transmission successful from the TX message buffer and transmission abort was requested	R/W
3	TMTRM	TX Message Buffer Transmission Request Mirrored 0: TX message buffer transmission not requested 1: TX message buffer transmission requested	R
4	TMTARM	TX Message Buffer Transmission Abort Request Mirrored 0: TX message buffer transmission request abort not requested 1: TX message buffer transmission request abort requested	R
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTMSTSn register (n = 0 to 127) shows status of the transmission and transmission abort for the corresponding message buffers.

TMTSTS bit (TX Message Buffer Transmission Status)

The TMTSTS bit is set automatically at the start of the transmission from the corresponding TX message buffer.

This bit is cleared automatically when:

- Transmission stops
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

TMTRF[1:0] bits (TX Message Buffer Transmission Result Flag)

The TMTRF[1:0] bits show the result for the corresponding TX message buffer. The status is as follows:

- 00b: Transmission in progress or has not been requested
- 01b: Transmission has been aborted from the corresponding TX message buffer
- 10b: Transmission was successful from the corresponding TX message buffer and the CFDTMCn.TMTAR bit was not set for this TX message buffer
- 11b: Transmission was successful from the corresponding TX message buffer, but the CFDTMCn.TMTAR bit was set for this TX message buffer.

Only write to these bits when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

The TMTRF[1:0] bits are cleared automatically when the CAN-FD module is in GL_RESET mode or the related channel is in CH_RESET mode.

TMTRM bit (TX Message Buffer Transmission Request Mirrored)

The TMTRM bit is set when the CFDTMCn.TMTR bit in the corresponding CFDTMCn register is set.

This bit is cleared when the CFDTMCn.TMTR bit in the corresponding CFDTMCn register is cleared.

TMTARM bit (TX Message Buffer Transmission Abort Request Mirrored)

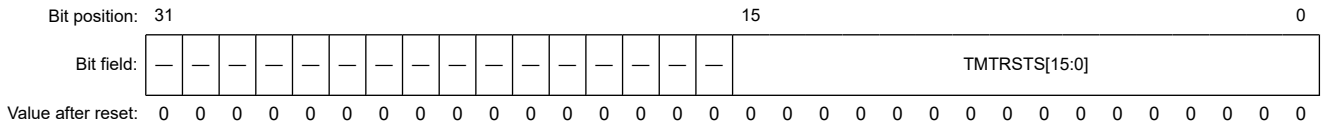
The TMTARM bit is set when the CFDTMCn.TMTAR bit in the corresponding CFDTMCn register is set.

This bit is cleared when the CFDTMCn.TMTAR bit in the corresponding CFDTMCn register is cleared.

33.2.49 CFDTMTRSTSf : TX Message Buffer Transmission Request Status Register f (f = 0 to 3)

Base address: CANFD = 0x8002_0000

Offset address: 0x0CD0 + 0x04 × f



Bit	Symbol	Function	R/W
15:0	TMTRSTS[15:0]	TX Message Buffer Transmission Request Status 0: Transmission not requested for corresponding TX message buffer 1: Transmission requested for corresponding TX message buffer	R
31:16	—	These bits are read as 0.	R

TMTRSTS[15:0] bits (TX Message Buffer Transmission Request Status)

The TMTRSTS[15:0] bits show status of the CFDTMCn.TMTR bits of the TX Message Buffer Control Registers.

Alignment of the TMTRSTS[15:0] bits is shown in Table 33.6.

Table 33.6 Alignment of TMTRSTS[15:0] mirror bits

Bit position	TX message buffer number
n × 64 - fmin × 32	n × 64 + 0
n × 64 + 1 - fmin × 32	n × 64 + 1
⋮	⋮
n × 64 + 31 - fmin × 32	n × 64 + 31
n × 64 + 32 - fmax × 32	n × 64 + 32
n × 64 + 33 - fmax × 32	n × 64 + 33
⋮	⋮
n ×64 + 62 - fmax × 32	n × 64 + 62
n × 64 + 63 - fmax × 32	n × 64 + 63

Note: When n = 0, fmin = 0, fmax = 1
When n = 1, fmin = 2, fmax = 3

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Control Registers (CFDTMCn), and only when the message buffer does not belong to a TX Queue.

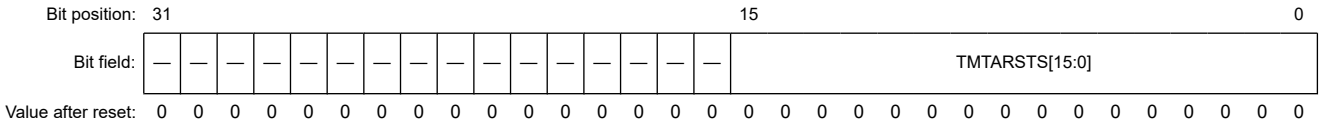
Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Control Registers
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

33.2.50 CFDTMTARSTSf : TX Message Buffer Transmission Abort Request Status Register f (f = 0 to 3)

Base address: CANFD = 0x8002_0000

Offset address: 0x0D70 + 0x04 × f



Bit	Symbol	Function	R/W
15:0	TMTARSTS[15:0]	TX Message Buffer Transmission Abort Request Status 0: Transmission abort not requested for corresponding TX message buffer 1: Transmission abort requested for corresponding TX message buffer	R
31:16	—	These bits are read as 0.	R

TMTARSTS[15:0] bits (TX Message Buffer Transmission Abort Request Status)

The TMTARSTS[15:0] bits show status of the CFDTMCn.TMTAR bits of the TX Message Buffer Control Registers.

Alignment of the TMTARSTS[15:0] bits is shown in Table 33.7.

Table 33.7 Alignment of TMTARSTS[15:0] mirror bits (1 of 2)

Bit position	TX message buffer number
n × 64 - fmin × 32	n × 64 + 0
n × 64 + 1 - fmin × 32	n × 64 + 1
⋮	⋮
n × 64 + 31 - fmin × 32	n × 64 + 31
n × 64 + 32 - fmax × 32	n × 64 + 32
n × 64 + 33 - fmax × 32	n × 64 + 33

Table 33.7 Alignment of TMTARSTS[15:0] mirror bits (2 of 2)

Bit position	TX message buffer number
⋮	⋮
n × 64 + 62 - fmax × 32	n × 64 + 62
n × 64 + 63 - fmax × 32	n × 64 + 63

Note: When n = 0, fmin = 0, fmax = 1
When n = 1, fmin = 2, fmax = 3

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Control Registers, and when the message buffer belongs to a TX Queue.

Each bit is cleared automatically when:

- The corresponding bit is cleared in the TX Message Buffer Control Registers
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

33.2.51 CFDTMTCSTSf : TX Message Buffer Transmission Completion Status Register f (f = 0 to 3)

Base address: CANFD = 0x8002_0000

Offset address: 0x0E10 + 0x04 × f

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TMTCSTS[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	TMTCSTS[15:0]	TX Message Buffer Transmission Completion Status 0: Transmission not completed for corresponding TX message buffer 1: Transmission completed for corresponding TX message buffer	R
31:16	—	These bits are read as 0.	R

TMTCSTS[15:0] bits (TX Message Buffer Transmission Completion Status)

The TMTCSTS[15:0] bits show status of successful completion of the TX Message Buffer Status Registers.

Alignment of the TMTCSTS[15:0] bits is shown in Table 33.8.

Table 33.8 Alignment of TMTCSTS[15:0] mirror bits

Bit position	TX message buffer number
n × 64 - fmin × 32	n × 64 + 0
n × 64 + 1 - fmin × 32	n × 64 + 1
⋮	⋮
n × 64 + 31 - fmin × 32	n × 64 + 31
n × 64 + 32 - fmax × 32	n × 64 + 32
n × 64 + 33 - fmax × 32	n × 64 + 33
⋮	⋮
n × 64 + 62 - fmax × 32	n × 64 + 62
n × 64 + 63 - fmax × 32	n × 64 + 63

Note: When n = 0, fmin = 0, fmax = 1
When n = 1,fmin = 2, fmax = 3

Each bit is set automatically when the corresponding bit is set in the TX Message Buffer Status Registers.

Each bit is cleared automatically when:

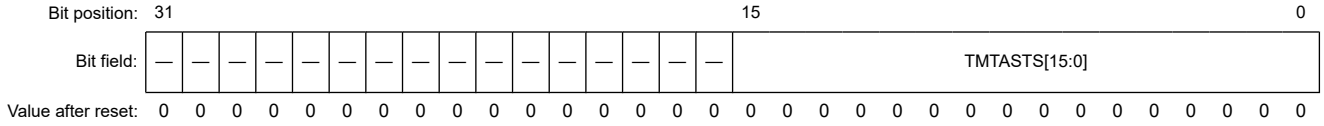
- The corresponding bit is cleared in the TX Message Buffer Status Registers
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

If a CAN channel enters CH_RESET mode, then the bits related to that channel are cleared.

33.2.52 CFDTMTASTSf : TX Message Buffer Transmission Abort Status Register f (f = 0 to 3)

Base address: CANFD = 0x8002_0000

Offset address: 0x0EB0 + 0x04 × f



Bit	Symbol	Function	R/W
15:0	TMTASTS[15:0]	TX Message Buffer Transmission Abort Status 0: Transmission not aborted for corresponding TX message buffer 1: Transmission aborted for corresponding TX message buffer	R
31:16	—	These bits are read as 0.	R

TMTASTS[15:0] bits (TX Message Buffer Transmission Abort Status)

The TMTASTS[15:0] bits show status of the successful transmission abort of the corresponding TX message buffer.

Alignment of the TMTASTS[15:0] bits is shown in Table 33.9.

Table 33.9 Alignment of TMTASTS[15:0] mirror bits

Bit position	TX message buffer number
n × 64 - fmin × 32	n × 64 + 0
n × 64 + 1 - fmin × 32	n × 64 + 1
⋮	⋮
n × 64 + 31 - fmin × 32	n × 64 + 31
n × 64 + 32 - fmax × 32	n × 64 + 32
n × 64 + 33 - fmax × 32	n × 64 + 33
⋮	⋮
n × 64 + 62 - fmax × 32	n × 64 + 62
n × 64 + 63 - fmax × 32	n × 64 + 63

Note: When n = 0, fmin = 0, fmax = 1
When n = 1, fmin = 2, fmax = 3

Each bit is set automatically when the CFDTMSTSn.TMTRF[1:0] bits are set to 01b in the corresponding TX Message Buffer Status Register.

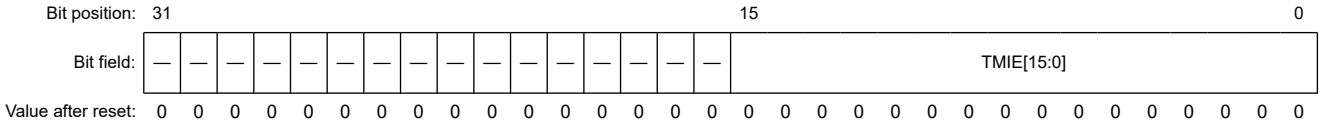
Each bit is cleared automatically when:

- The CFDTMSTSn.TMTRF[1:0] bits are cleared in the corresponding TX Message Buffer Status Register
- The CAN-FD module is in GL_RESET mode
- The related CAN-FD channel is in CH_RESET mode.

33.2.53 CFDTMIECf : TX Message Buffer Transmission Interrupt Enable Register f (f = 0 to 3)

Base address: CANFD = 0x8002_0000

Offset address: 0x0F50 + 0x04 × f



Bit	Symbol	Function	R/W
15:0	TMIE[15:0]	TX Message Buffer Interrupt Enable 0: TX message buffer interrupt disabled for corresponding TX message buffer 1: TX message buffer interrupt enabled for corresponding TX message buffer	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

TMIE[15:0] bits (TX Message Buffer Interrupt Enable)

If the TMIE[15:0] bits are set, an interrupt is generated at the end of a successful transmission from the corresponding message buffer.

See section 33.7. Interrupt and DMA for TX Message Buffer Interrupt specification.

Alignment of the TMIE[15:0] bits is shown in Table 33.10.

Table 33.10 Alignment of TMIE[15:0] bits

Bit position	TX message buffer number
n × 64 - fmin × 32	n × 64 + 0
n × 64 + 1 - fmin × 32	n × 64 + 1
⋮	⋮
n × 64 + 31 - fmin × 32	n × 64 + 31
n × 64 + 32 - fmax × 32	n × 64 + 32
n × 64 + 33 - fmax × 32	n × 64 + 33
⋮	⋮
n × 64 + 62 - fmax × 32	n × 64 + 62
n × 64 + 63 - fmax × 32	n × 64 + 63

Note: When n = 0, fmin = 0, fmax = 1
When n = 1, fmin = 2, fmax = 3

Do not write to the TMIE[15:0] bits when:

- The CAN-FD module is in GL_SLEEP mode
- The related CAN-FD channel is in CH_SLEEP mode
- The corresponding TX message buffer is part of a TX Queue
- The corresponding TX message buffer is linked to a Common FIFO with the CFDCFCCn.CFTML[4:0] bits.

33.2.54 CFDTXQCC0n : TX Queue Configuration/Control Register 0n (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x1000 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQOFTXIE	TXQOFRXIE	TXQFIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TXQDC[4:0]					TXQIM	—	TXQTXIE	—	—	TXQOWE	TXQGWE	TXQE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXQE	TX Queue Enable 0: TX Queue disabled 1: TX Queue enabled	R/W
1	TXQGWE	TX Queue Gateway Mode Enable 0: TX Queue GW mode disabled 1: TX Queue GW mode enabled	R/W
2	TXQOWE	TX Queue Overwrite Mode Enable 0: TX Queue OW mode disabled 1: TX Queue OW mode enabled	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	TXQTXIE	TX Queue TX Interrupt Enable 0: TX Queue TX interrupt disabled 1: TX Queue TX interrupt enabled	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	TXQIM	TX Queue Interrupt Mode 0: When the last message is successfully transmitted 1: At every successful transmission	R/W
12:8	TXQDC[4:0]	TX Queue Depth Configuration 0x00: 0 messages 0x01: Reserved 0x02: 3 messages 0x03: 4 messages ⋮ 0x0F: 16 messages Others: Setting prohibited	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
16	TXQFIE	TXQ Full Interrupt Enable 0: TX Queue full interrupt generation disabled 1: TX Queue full interrupt generation enabled	R/W
17	TXQOFRXIE	TXQ One Frame Reception Interrupt Enable 0: One Frame RX interrupt generation disabled 1: One Frame RX interrupt generation enabled	R/W
18	TXQOFTXIE	TXQ One Frame Transmission Interrupt Enable 0: One Frame TX interrupt generation disabled 1: One Frame TX interrupt generation enabled	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTXQCC0n register (n = 0, 1) is used to configure the TX Queue transmission.

TXQ0 is composed of TXMB0 to TXMB31 (at the maximum) when TXQE is enabled.

TXQE bit (TX Queue Enable)

The TXQE bit cannot be set if the configured TX Queue depth is 0x00 (CFDTXQCC0n.TXQDC[4:0] = 0x00).

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_RESET or CH_SLEEP mode.

The TXQE bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

TXQGWE bit (TX Queue Gateway Mode Enable)

When the TXQGWE bit is set, the TX Queue is in TX Queue GW mode.

When this bit is set, CPU must not access the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQOWE bit (TX Queue Overwrite Mode Enable)

When the TXQOWE bit is set, the TX Queue is in TX Queue Overwrite mode.

An overwrite function is valid when the same ID such as ID of the data written in from the gateway or CPU is in TX Queue. For example, when a frame is received and is stored into the TX Queue, if a message with the same ID is stored in the TX Queue, the old message is overwritten by the new message.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

When using the function in GW mode, the depth of TXQ (CFDTXQCC0n.TXQDC) should be configured to the value which is the various number of ID that is used in the TX Queue plus 3.

The function is valid for the standard ID frame but is invalid for the extended ID frame.

Do not modify this bit when the CFDTXQCC0n.TXQE bit is 1.

TXQTXIE bit (TX Queue TX Interrupt Enable)

When the TXQTXIE bit is set, an interrupt is generated based on the setting of the TXQIM bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

TXQIM bit (TX Queue Interrupt Mode)

The TXQIM bit selects the interrupt generation condition for the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQDC[4:0] bits (TX Queue Depth Configuration)

The TXQDC[4:0] bits select the depth of the transmission queue. The message buffer selection starts from MB[0] up to MB[31] depending on the configured depth.

When using TXQ1 and TXQ0 simultaneously, the total depth of TXQ1 and TXQ0 should be 32 or less.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQFIE bit (TXQ Full Interrupt Enable)

When the TXQFIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS0n.TXQFIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

Only write 1 to this bit in Gateway mode (CFDTXQCC0n.TXQGWE = 1).

TXQOFRXIE bit (TXQ One Frame Reception Interrupt Enable)

When the TXQOFRXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS0n.TXQOFRXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

Only write 1 to this bit in GW mode (CFDTXQCC0n.TXQGWE = 1).

TXQOFTXIE bit (TXQ One Frame Transmission Interrupt Enable)

When the TXQOFTXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS0n.TXQOFTXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

33.2.55 CFDTXQCC1n : TX Queue Configuration/Control Register 1n (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x1060 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQOFTXIE	TXQOFRXIE	TXQFIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TXQDC[4:0]					TXQIM	—	TXQTXIE	—	—	TXQOWE	TXQGWE	TXQE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXQE	TX Queue Enable 0: TX Queue disabled 1: TX Queue enabled	R/W
1	TXQGWE	TX Queue Gateway Mode Enable 0: TX Queue GW mode disabled 1: TX Queue GW mode enabled	R/W
2	TXQOWE	TX Queue Overwrite Mode Enable 0: TX Queue OW mode disabled 1: TX Queue OW mode enabled	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
5	TXQTXIE	TX Queue TX Interrupt Enable 0: TX Queue TX interrupt disabled 1: TX Queue TX interrupt enabled	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	TXQIM	TX Queue Interrupt Mode 0: When the last message is successfully transmitted 1: At every successful transmission	R/W
12:8	TXQDC[4:0]	TX Queue Depth Configuration 0x00: 0 messages 0x01: Reserved 0x02: 3 messages 0x03: 4 messages ⋮ 0x0F: 16 messages Others: Setting prohibited	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
16	TXQFIE	TXQ Full Interrupt Enable 0: TX Queue full interrupt generation disabled 1: TX Queue full interrupt generation enabled	R/W
17	TXQOFRXIE	TXQ One Frame Reception Interrupt Enable 0: One Frame RX interrupt generation disabled 1: One Frame RX interrupt generation enabled	R/W
18	TXQOFTXIE	TXQ One Frame Transmission Interrupt Enable 0: One Frame TX interrupt generation disabled 1: One Frame TX interrupt generation enabled	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTXQCC1n register (n = 0, 1) is used to configure the TX Queue transmission.

TXQ1 is composed of TXMB31 to TXMB0 (at the maximum) when TXQE is enabled.

TXQE bit (TX Queue Enable)

The TXQE bit cannot be set if the configured TX Queue depth is 0x00 (CFDTXQCC1n.TXQDC[4:0] = 0x00).

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do write to this bit when the related CAN-FD channel is in CH_RESET or CH_SLEEP mode.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

TXQGWE bit (TX Queue Gateway Mode Enable)

When the TXQGWE bit is set, the TX Queue is in TX Queue Gateway mode.

When this bit is set, CPU must not access the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQOWE bit (TX Queue Overwrite Mode Enable)

When the TXQOWE bit is set, the TX Queue is in TX Queue Overwrite mode.

An overwrite function is valid when the same ID such as ID of the data written in from the gateway or CPU is in TX Queue. For example, when a frame is received and is stored into the TX Queue, if a message with the same ID is stored in the TX Queue, the old message is overwritten by the new message.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

When using the function in GW mode, the depth of TXQ (CFDTXQCC1n.TXQDC) should be configured to the value which is the various number of ID that is used in the TX Queue plus 3.

The function is valid for the standard ID frame but is invalid for the extended ID frame.

Do not modify this bit when the CFDTXQCC1n.TXQE bit is 1.

TXQTXIE bit (TX Queue TX Interrupt Enable)

When the TXQTXIE bit is set, an interrupt is generated based on the setting of the TXQIM bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

TXQIM bit (TX Queue Interrupt Mode)

The TXQIM bit selects the interrupt generation condition for the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQDC[4:0] bits (TX Queue Depth Configuration)

The TXQDC[4:0] bits select the depth of the transmission queue. The message buffer selection starts from MB[31] down to MB[0] depending on the configured depth.

When using TXQ1 and TXQ0 simultaneously, the total depth of TXQ1 and TXQ0 should be 32 or less.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQFIE bit (TXQ Full Interrupt Enable)

When the TXQFIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS1n.TXQFIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

Only write 1 to this bit when in Gateway mode (CFDTXQCC1n.TXQGWE = 1).

TXQOFRXIE bit (TXQ One Frame Reception Interrupt Enable)

When the TXQOFRXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS1n.TXQOFRXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

Only write 1 to this bit when in Gateway mode (CFDTXQCC1n.TXQGWE = 1).

TXQOFTXIE bit (TXQ One Frame Transmission Interrupt Enable)

When the TXQOFTXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS1n.TXQOFTXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

33.2.56 CFDTXQCC2n : TX Queue Configuration/Control Register 2n (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x10C0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQOFTXIE	TXQOFRXIE	TXQFIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TXQDC[4:0]					TXQIM	—	TXQTXIE	—	—	TXQOWE	TXQGWE	TXQE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXQE	TX Queue Enable 0: TX Queue disabled 1: TX Queue enabled	R/W
1	TXQGWE	TX Queue Gateway Mode Enable 0: TX Queue GW mode disabled 1: TX Queue GW mode enabled	R/W
2	TXQOWE	TX Queue Overwrite Mode Enable 0: TX Queue OW mode disabled 1: TX Queue OW mode enabled	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	TXQTXIE	TX Queue TX Interrupt Enable 0: TX Queue TX interrupt disabled 1: TX Queue TX interrupt enabled	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	TXQIM	TX Queue Interrupt Mode 0: When the last message is successfully transmitted 1: At every successful transmission	R/W
12:8	TXQDC[4:0]	TX Queue Depth Configuration 0x00: 0 messages 0x01: Reserved 0x02: 3 messages 0x03: 4 messages ⋮ 0x0F: 16 messages Others: Setting prohibited	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
16	TXQFIE	TXQ Full Interrupt Enable 0: TX Queue full interrupt generation disabled 1: TX Queue full interrupt generation enabled	R/W
17	TXQOFRXIE	TXQ One Frame Reception Interrupt Enable 0: One Frame RX interrupt generation disabled 1: One Frame RX interrupt generation enabled	R/W
18	TXQOFTXIE	TXQ One Frame Transmission Interrupt Enable 0: One Frame TX interrupt generation disabled 1: One Frame TX interrupt generation enabled	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTXQCC2n register (n = 0, 1) is used to configure the TX Queue transmission.

TXQ2 is composed of TXMB32 to TXMB63 (at the maximum) when TXQE is enabled.

TXQE bit (TX Queue Enable)

The TXQE bit cannot be set if the configured TX Queue depth is 0x00 (CFDTXQCC2n.TXQDC[4:0] = 0x00).

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_RESET or CH_SLEEP mode.

TXQGWE bit (TX Queue Gateway Mode Enable)

When the TXQGWE bit is set, the TX Queue is in TX Queue Gateway mode. When this bit is set, CPU must not access the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQOWE bit (TX Queue Overwrite Mode Enable)

When the TXQOWE bit is set, the TX Queue is in TX Queue Overwrite mode.

An overwrite function is valid when the same ID such as ID of the data written in from the gateway or CPU is in TX Queue. For example, when a frame is received and is stored into the TX Queue, if a message with the same ID is stored in the TX Queue, the old message is overwritten by the new message.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

When using the function in GW mode, the depth of TXQ (CFDTXQCC2n.TXQDC) should be configured to the value which is the various number of ID that is used in the TX Queue plus 3.

The function is valid for the standard ID frame but is invalid for the extended ID frame.

Do not modify this bit when the CFDTXQCC2n.TXQE bit is 1.

TXQTXIE bit (TX Queue TX Interrupt Enable)

When the TXQTXIE bit is set, an interrupt is generated based on the setting of the TXQIM bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

TXQIM bit (TX Queue Interrupt Mode)

The TXQIM bit selects the interrupt generation condition for the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQDC[4:0] bits (TX Queue Depth Configuration)

The TXQDC[4:0] bits select the depth of the transmission queue. The message buffer selection starts from MB[32] up to MB[63] depending on the configured depth.

When using TXQ3 and TXQ2 simultaneously, the total depth of TXQ3 and TXQ2 should be 32 or less.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION

TXQFIE bit (TXQ Full Interrupt Enable)

When the TXQFIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS2n.TXQFIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

Only write 1 to this bit when in GW mode (CFDTXQCC2n.TXQGWE = 1).

TXQOFRXIE bit (TXQ One Frame Reception Interrupt Enable)

When the TXQOFRXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS2n.TXQOFRXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

Only write 1 to this bit when in GW mode (CFDTXQCC2n.TXQGWE = 1).

TXQOFTXIE bit (TXQ One Frame Transmission Interrupt Enable)

When the TXQOFTXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS2n.TXQOFTXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

33.2.57 CFDTXQCC3n : TX Queue Configuration/Control Register 3n (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x1120 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQOFTXIE	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	TXQDC[4:0]					TXQIM	—	TXQTXIE	—	—	TXQOWE	—	TXQE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXQE	TX Queue Enable 0: TX Queue disabled 1: TX Queue enabled	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	TXQOWE	TX Queue Overwrite Mode Enable 0: TX Queue OW mode disabled 1: TX Queue OW mode enabled	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	TXQTXIE	TX Queue TX Interrupt Enable 0: TX Queue TX interrupt disabled 1: TX Queue TX interrupt enabled	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
7	TXQIM	TX Queue Interrupt Mode 0: When the last message is successfully transmitted 1: At every successful transmission	R/W
12:8	TXQDC[4:0]	TX Queue Depth Configuration 0x00: 0 messages 0x01: Reserved 0x02: 3 messages 0x03: 4 messages ⋮ 0x0F: 16 messages Others: Setting prohibited	R/W
17:13	—	These bits are read as 0. The write value should be 0.	R/W
18	TXQOFTXIE	TXQ One Frame Transmission Interrupt Enable 0: One Frame TX interrupt generation disabled 1: One Frame TX interrupt generation enabled	R/W
31:19	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTXQCC3n register (n = 0, 1) is used to configure the TX Queue transmission.

TXQ3 is composed of TXMB63 to TXMB32 (at the maximum) when TXQE is enabled.

TXQE bit (TX Queue Enable)

The TXQE bit cannot be set if the configured TX Queue depth is 0x00 (CFDTXQCC3n.TXQDC[4:0] = 0x00).

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_RESET or CH_SLEEP mode.

TXQOWE bit (TX Queue Overwrite Mode Enable)

When the TXQOWE bit is set, the TX Queue is in TX Queue Overwrite mode.

An overwrite function is valid when the same ID such as ID of the data written in from the gateway or CPU is in TX Queue. For example, when a frame is received and is stored into the TX Queue, if a message with the same ID is stored in the TX Queue, the old message is overwritten by the new message.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

When using the function in GW mode, the depth of TXQ (CFDTXQCC3n.TXQDC) should be configured to the value which is the various number of ID that is used in the TX Queue plus 3.

The function is valid for the standard ID frame but is invalid for the extended ID frame.

Do not modify this bit when the CFDTXQCC3n.TXQE bit is 1.

TXQTXIE bit (TX Queue TX Interrupt Enable)

When the TXQTXIE bit is set, an interrupt is generated based on the setting of the TXQIM bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

TXQIM bit (TX Queue Interrupt Mode)

The TXQIM bit selects the interrupt generation condition for the TX Queue.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQDC[4:0] bits (TX Queue Depth Configuration)

The TXQDC[4:0] bits select the depth of the transmission queue. The message buffer selection starts from MB[63] down to MB[32] depending on the configured depth.

When using TXQ3 and TXQ2 simultaneously, the total depth of TXQ3 and TXQ2 should be 32 or less.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in any of the following modes:

- CH_SLEEP
- CH_HALT
- CH_OPERATION.

TXQOFTXIE bit (TXQ One Frame Transmission Interrupt Enable)

When the TXQOFTXIE bit is set, an interrupt is generated based on the setting of the CFDTXQSTS3n.TXQOFTXIF bit.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the related CAN-FD channel is in CH_SLEEP mode.

33.2.58 CFDTXQSTS0n : TX Queue Status Register 0n (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x1020 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	TXQMOW	TXQMLT	TXQOFTXIF	TXQOFRXIF	TXQFIF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	TXQMC[5:0]						—	—	—	—	—	TXQTXIF	TXQFLL	TXQEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	TXQEMP	TX Queue Empty 0: TX Queue not empty 1: TX Queue empty	R
1	TXQFLL	TX Queue Full 0: TX Queue not full 1: TX Queue full	R
2	TXQTXIF	TX Queue TX Interrupt Flag 0: TX Queue interrupt condition not satisfied after a frame TX 1: TX Queue interrupt condition satisfied after a frame TX	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
13:8	TXQMC[5:0]	TX Queue Message Count Number of messages in the TX Queue.	R
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	TXQFIF	TXQ Full Interrupt Flag When TXQ is in full status, an interrupt is set.	R/W

Bit	Symbol	Function	R/W
17	TXQOFRXIF	TXQ One Frame Reception Interrupt Flag When TXQ receives one frame, an interrupt is set.	R/W
18	TXQOFTXIF	TXQ One Frame Transmission Interrupt Flag When one frame transmits from TXQ, an interrupt is set.	R/W
19	TXQMLT	TXQ Message Lost 0: No message lost in TXQ 1: TXQ message lost	R/W
20	TXQMOW	TXQ Message Overwrite 0: No message overwrite in TXQ 1: Message overwrite in TXQ	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTXQSTS0n register (n = 0, 1) shows the status of the TX Queue of corresponding CAN channel.

TXQEMP bit (TX Queue Empty)

The TXQEMP bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

This bit is set automatically when:

- The last message is transmitted from the TX Queue
- The related CAN-FD channel is in CH_RESET mode.

The bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

TXQFLL bit (TX Queue Full)

The TXQFLL bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when:

- The number of CAN messages stored in the TX Queue is less than the configured TX Queue depth
- The related CAN-FD channel is in CH_RESET mode.

TXQTXIF bit (TX Queue TX Interrupt Flag)

The TXQTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMC[5:0] bits (TX Queue Message Count)

The TXQMC[5:0] bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

TXQFIF bit (TXQ Full Interrupt Flag)

The TXQFIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of the TX Queue.

Only in Gateway mode (CFDTXQCC0n.TXQGWE = 1) that this bit is set automatically when the TX Queue transits to a buffer full status.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQOFRXIF bit (TXQ One Frame Reception Interrupt Flag)

The TXQOFRXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When receiving data is stored in the TX Queue in Gateway mode, this bit is set automatically.

This function can only be used in Gateway mode of the TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQOFTXIF bit (TXQ One Frame Transmission Interrupt Flag)

The TXQOFTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When transmission is successful in the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMLT bit (TXQ Message Lost)

The TXQMLT bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When a message lost occurs in Gateway mode of the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMOW bit (TXQ Message Overwrite)

The TXQMOW bit is not cleared automatically if the TX Queue is disabled.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When CFDTXQCC0n.TXQOWE = 1 and message overwrite occurs in TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

33.2.59 CFDTXQSTS1n : TX Queue Status Register 1n (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x1080 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	TXQMOW	TXQMLT	TXQOFTXIF	TXQOFRXIF	TXQFIF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	TXQMC[5:0]						—	—	—	—	—	TXQTXIF	TXQFLL	TXQEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	TXQEMP	TX Queue Empty 0: TX Queue not empty 1: TX Queue empty	R
1	TXQFLL	TX Queue Full 0: TX Queue not full 1: TX Queue full	R
2	TXQTXIF	TX Queue TX Interrupt Flag 0: TX Queue interrupt condition not satisfied after a frame TX 1: TX Queue interrupt condition satisfied after a frame TX	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
13:8	TXQMC[5:0]	TX Queue Message Count Number of messages in the TX Queue	R
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	TXQFIF	TXQ Full Interrupt Flag When TXQ is in full status, an interrupt is set.	R/W
17	TXQOFRXIF	TXQ One Frame Reception Interrupt Flag When TXQ receives one frame, an interrupt is set.	R/W
18	TXQOFTXIF	TXQ One Frame Transmission Interrupt Flag When one frame transmits from TXQ, an interrupt is set.	R/W
19	TXQMLT	TXQ Message Lost 0: No message lost in TXQ 1: TXQ message lost	R/W
20	TXQMOW	TXQ Message Overwrite 0: No message overwrite in TXQ 1: Message overwrite in TXQ	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTXQSTS1n register (n = 0, 1) shows the status of the TX Queue of corresponding CAN channel.

TXQEMP bit (TX Queue Empty)

The TXQEMP bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

The bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

This bit is set automatically when:

- The last message is transmitted from the TX Queue
- The related CAN-FD channel is in CH_RESET mode.

TXQFLL bit (TX Queue Full)

The TXQFLL bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when:

- The number of CAN messages stored in the TX Queue is less than the configured TX Queue depth
- The related CAN-FD channel is in CH_RESET mode.

TXQTXIF bit (TX Queue TX Interrupt Flag)

The TXQTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMC[5:0] bits (TX Queue Message Count)

The TXQMC[5:0] bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

TXQFIF bit (TXQ Full Interrupt Flag)

The TXQFIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

Only in Gateway mode (CFDTXQCC1n.TXQGWE = 1) that this bit is set automatically when TX Queue transits to a buffer full status.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQOFRXIF bit (TXQ One Frame Reception Interrupt Flag)

The TXQOFRXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When receiving data is stored in the TX Queue in Gateway mode, this bit is set automatically.

This function can only be used in Gateway mode of the TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQOFTXIF bit (TXQ One Frame Transmission Interrupt Flag)

The TXQOFTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When transmission is successful in the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMLT bit (TXQ Message Lost)

The TXQMLT bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When a message lost occurs in Gateway mode of the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMOW bit (TXQ Message Overwrite)

The TXQMOW bit is not cleared automatically if the TX Queue is disabled.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When CFDTXQCC1n.TXQOWE = 1 and message overwrite occurs in TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

33.2.60 CFDTXQSTS2n : TX Queue Status Register 2n (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x10E0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	TXQMOW	TXQMLT	TXQOFTXIF	TXQOFRXIF	TXQFIF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	TXQMC[5:0]						—	—	—	—	—	TXQTXIF	TXQFLL	TXQEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	TXQEMP	TX Queue Empty 0: TX Queue not empty 1: TX Queue empty	R
1	TXQFLL	TX Queue Full 0: TX Queue not full 1: TX Queue full	R
2	TXQTXIF	TX Queue TX Interrupt Flag 0: TX Queue interrupt condition not satisfied after a frame TX 1: TX Queue interrupt condition satisfied after a frame TX	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
13:8	TXQMC[5:0]	TX Queue Message Count Number of messages in the TX Queue.	R
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	TXQFIF	TXQ Full Interrupt Flag When TXQ is in full status, an interrupt is set.	R/W
17	TXQOFRXIF	TXQ One Frame Reception Interrupt Flag When TXQ receives one frame, an interrupt is set.	R/W
18	TXQOFTXIF	TXQ One Frame Transmission Interrupt Flag When one frame transmits from TXQ, an interrupt is set.	R/W
19	TXQMLT	TXQ Message Lost 0: No message lost in TXQ 1: TXQ message lost	R/W
20	TXQMOW	TXQ Message Overwrite 0: No message overwrite in TXQ 1: Message overwrite in TXQ	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTXQSTS2n register (n = 0, 1) shows the status of the TX Queue of corresponding CAN Channel.

TXQEMP bit (TX Queue Empty)

The TXQEMP bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

This bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

This bit is set automatically when:

- The last message is transmitted from the TX Queue
- The related CAN-FD channel is in CH_RESET mode.

TXQFLL bit (TX Queue Full)

The TXQFLL bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when:

- The number of CAN messages stored in the TX Queue is less than the configured TX Queue depth
- The related CAN-FD channel is in CH_RESET mode.

TXQTXIF bit (TX Queue TX Interrupt Flag)

The TXQTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of the TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMC[5:0] bits (TX Queue Message Count)

The TXQMC[5:0] bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

TXQFIF bit (TXQ Full Interrupt Flag)

The TXQFIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of the TX Queue.

Only in Gateway mode (CFDTXQCC2n.TXQGWE = 1) that this bit is set automatically when the TX Queue transits to a buffer full status.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQOFRXIF bit (TXQ One Frame Reception Interrupt Flag)

The TXQOFRXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When receiving data is stored in the TX Queue in Gateway mode, this bit is set automatically.

This function can only be used in Gateway mode of the TX queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQOFTXIF bit (TXQ One Frame Transmission Interrupt Flag)

The TXQOFTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When transmission is successful in the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

The bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMLT bit (TXQ Message Lost)

The TXQMLT bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When a message lost occurs in Gateway mode of the TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMOW bit (TXQ Message Overwrite)

The TXQMOW bit is not cleared automatically if the TX Queue is disabled.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When CFDTXQCC2n.TXQOWE = 1 and message overwrite occurs in TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

33.2.61 CFDTXQSTS3n : TX Queue Status Register 3n (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x1140 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	TXQMOW	—	TXQOFTXIF	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	TXQMC[5:0]						—	—	—	—	—	TXQTXIF	TXQFLL	TXQEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	TXQEMP	TX Queue Empty 0: TX Queue not empty 1: TX Queue empty	R
1	TXQFLL	TX Queue Full 0: TX Queue not full 1: TX Queue full	R
2	TXQTXIF	TX Queue TX Interrupt Flag 0: TX Queue interrupt condition not satisfied after a frame TX 1: TX Queue interrupt condition satisfied after a frame TX	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
13:8	TXQMC[5:0]	TX Queue Message Count Number of messages in the TX Queue	R
17:14	—	These bits are read as 0. The write value should be 0.	R/W
18	TXQOFTXIF	TXQ One Frame Transmission Interrupt Flag When one frame transmits from TXQ, an interrupt is set.	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
20	TXQMOW	TXQ Message Overwrite 0: No message overwrite in TXQ 1: Message overwrite in TXQ	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTXQSTS3n register (n = 0, 1) shows the status of the TX Queue of corresponding CAN Channel.

TXQEMP bit (TX Queue Empty)

The TXQEMP bit is set automatically when the TX Queue is disabled or no messages are stored in the TX Queue.

This bit is set automatically when:

- The last message is transmitted from the TX Queue
- The related CAN-FD channel is in CH_RESET mode.

This bit is cleared automatically when the first message to be transmitted is stored in the TX Queue.

TXQFLL bit (TX Queue Full)

The TXQFLL bit is set automatically when the number of CAN messages stored in the TX Queue matches the configured TX Queue depth.

This bit is cleared automatically when:

- The number of CAN messages stored in the TX Queue is less than the configured TX Queue depth
- The related CAN-FD channel is in CH_RESET mode.

TXQTXIF bit (TX Queue TX Interrupt Flag)

The TXQTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of the TX Queue.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is set automatically when the configured interrupt condition is satisfied for the TX Queue.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMC[5:0] bits (TX Queue Message Count)

The TXQMC[5:0] bits show the number of CAN messages in the TX Queue.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

TXQOFTXIF bit (TXQ One Frame Transmission Interrupt Flag)

The TXQOFTXIF bit is not cleared automatically if the TX Queue is disabled.

When stopping the TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue. When transmission is successful in the TX Queue, this bit is set automatically.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

TXQMOW bit (TXQ Message Overwrite)

The TXQMOW bit is not cleared automatically if the TX Queue is disabled.

When stopping TX Queue, this bit should be cleared, after disabling TXQE and checking an empty state of TX Queue.

When CFDTXQCC3n.TXQOWE = 1 and message overwrite occurs in TX Queue, this bit is set automatically.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1. Writing 1 has no effect.

You cannot write to this bit when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

33.2.62 CFDTXQPCTR0n : TX Queue Pointer Control Register 0n (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x1040 + 0x04 × n

Bit position:	31																							7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQPC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TXQPC[7:0]	TX Queue Pointer Control Increments the write pointer to the TX Queue buffer in the corresponding channel	W
31:8	—	The write value should be 0.	W

The CFDTXQPCTR0n register (n = 0, 1) is used to confirm storage of a full message in the corresponding TX Queue buffers.

TXQPC[7:0] bits (TX Queue Pointer Control)

When the value 0xFF is written to the TXQPC[7:0] bits, the write pointer of the corresponding TX Queue buffer is updated and a transmit request is initiated for this message.

The read value from these bits is always 0x00. Do not write to the FIFO control registers when DMA is enabled.

You cannot write to these bits when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

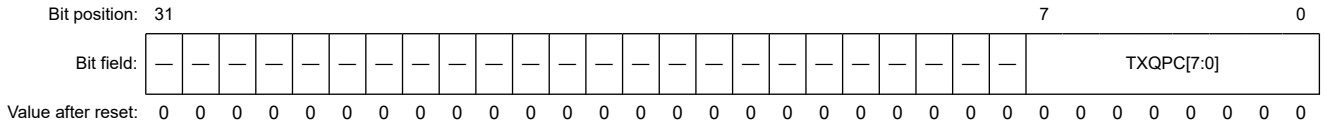
Only write 0xFF to this register when:

- The corresponding TX Queue is enabled and not full
- The Common FIFO is enabled and is not configured in GW mode.

33.2.63 CFDTXQPCTR1n : TX Queue Pointer Control Register 1n (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x10A0 + 0x04 × n



Bit	Symbol	Function	R/W
7:0	TXQPC[7:0]	TX Queue Pointer Control Increments the write pointer to the TX Queue buffer in the corresponding channel	W
31:8	—	The write value should be 0.	W

The CFDTXQPCTR1n register (n = 0, 1) is used to confirm storage of a full message in the corresponding TX Queue buffers.

TXQPC[7:0] bits (TX Queue Pointer Control)

When the value 0xFF is written to the TXQPC[7:0] bits, the write pointer of the corresponding TX Queue buffer is updated and a transmit request is initiated for this message.

The read value from these bits is always 0x00. You cannot write to the FIFO control registers when DMA is enabled.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

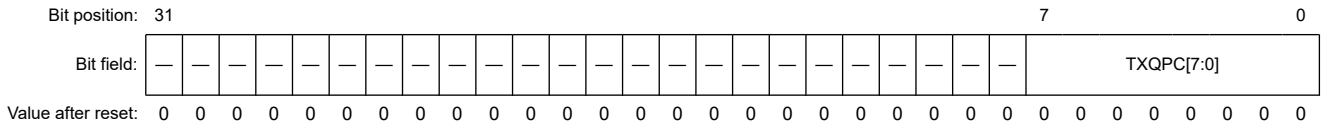
Only write 0xFF to this register when:

- The corresponding TX Queue is enabled and not full
- The Common FIFO is enabled and is not configured in GW mode.

33.2.64 CFDTXQPCTR2n : TX Queue Pointer Control Register 2n (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x1100 + 0x04 × n



Bit	Symbol	Function	R/W
7:0	TXQPC[7:0]	TX Queue Pointer Control Increments the write pointer to the TX Queue buffer in the corresponding channel	W
31:8	—	The write value should be 0.	W

The CFDTXQPCTR2n register (n = 0, 1) is used to confirm storage of a full message in the corresponding TX Queue buffers.

TXQPC[7:0] bits (TX Queue Pointer Control)

When the value 0xFF is written to the TXQPC[7:0] bits, the write pointer of the corresponding TX Queue buffer is updated and a transmit request is initiated for this message.

The read value from these bits is always 0x00. You cannot write to the FIFO control registers when DMA is enabled.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

Only write 0xFF to this register when:

- The corresponding TX Queue is enabled and not full
- The Common FIFO is enabled and is not configured in GW mode.

33.2.65 CFDTXQPCTR3n : TX Queue Pointer Control Register 3n (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x1160 + 0x04 × n

Bit position:	31																						7							0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQPC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TXQPC[7:0]	TX Queue Pointer Control Increments the write pointer to the TX Queue buffer in the corresponding channel	W
31:8	—	The write value should be 0.	W

The CFDTXQPCTR3n register (n = 0, 1) is used to confirm storage of a full message in the corresponding TX Queue buffers.

TXQPC[7:0] bits (TX Queue Pointer Control)

When the value 0xFF is written to the TXQPC[7:0] bits, the write pointer of the corresponding TX Queue buffer is updated and a transmit request is initiated for this message.

The read value from these bits is always 0x00. Do not write to the FIFO control registers when DMA is enabled.

You cannot write to these bits when the related CAN-FD channel is in CH_SLEEP or CH_RESET mode.

Only write 0xFF to this register when:

- The corresponding TX Queue is enabled and not full
- The Common FIFO is enabled and is not configured in Gateway mode.

33.2.66 CFDTXQESTS : TX Queue Empty Status Register

Base address: CANFD = 0x8002_0000

Offset address: 0x1180

Bit position:	31																						8	7						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQxEMP[7:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	TXQxEMP[7:0]	TXQ Empty Status 0: TXQ not empty 1: TXQ empty	R
31:8	—	These bits are read as 0.	R

The CFDTXQESTS register shows the status of the empty bits of the TXQ buffers.

TXQxEMP[7:0] bits (TXQ Empty Status)

Each bit is set automatically when the corresponding bit is set in the TX Queue Empty Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue Empty Status Register.

This bit is set when the CAN-FD module is in GL_RESET mode.

Bit position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Channel 0 TX Queue 3
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Channel 1 TX Queue 3

33.2.67 CFDTXQFISTS : TX Queue Full Interrupt Status Register

Base address: CANFD = 0x8002_0000

Offset address: 0x1184

Bit position:	31																								6	4	2	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQ1FULL[2:0]	—	TXQ0FULL[2:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TXQ0FULL[2:0]	TXQ Full Interrupt Status Flag for Channel 0 0: TXQ full interrupt not set 1: TXQ full interrupt set	R
3	—	This bit is read as 0.	R
6:4	TXQ1FULL[2:0]	TXQ Full Interrupt Status Flag for Channel 1 0: TXQ full interrupt not set 1: TXQ full interrupt set	R
31:7	—	These bits are read as 0.	R

The CFDTXQFISTS register shows the status of the full interrupt bits of the TXQ buffers.

TXQnFULL[2:0] (n = 0, 1) bits (TXQ Full Interrupt Status Flag)

Each bit is set automatically when the corresponding bit is set in the TX Queue Full Interrupt Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue Full Interrupt Status Register.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

Bit position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Reserved
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Reserved

33.2.68 CFDTXQMSTS : TX Queue Message Lost Status Register

Base address: CANFD = 0x8002_0000

Offset address: 0x1188

Bit position:	31																								6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQ1ML[2:0]			—	TXQ0ML[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TXQ0ML[2:0]	TXQ Message Lost Status Flag for Channel 0 0: TXQ message lost flag not set 1: TXQ message lost flag set	R
3	—	This bit is read as 0.	R
6:4	TXQ1ML[2:0]	TXQ Message Lost Status Flag for Channel 1 0: TXQ message lost flag not set 1: TXQ message lost flag set	R
31:7	—	These bits are read as 0.	R

The CFDTXQMSTS register shows the status of the message lost bits of the TXQ buffers.

TXQnML[2:0] (n = 0, 1) bits (TXQ Message Lost Status Flag)

Each bit is set automatically when the corresponding bit is set in the TX Queue Message Lost Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue Message Lost Status Register.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

Bit position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Reserved
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Reserved

33.2.69 CFDTXQISTS : TX Queue Interrupt Status Register

Base address: CANFD = 0x8002_0000

Offset address: 0x1190

Bit position:	31																								7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQ1ISF[3:0]				TXQ0ISF[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	TXQ0ISF[3:0]	TXQ Interrupt Status Flag for Channel 0 0: TXQ Interrupt flag not set 1: TXQ Interrupt flag set	R

Bit	Symbol	Function	R/W
7:4	TXQ1ISF[3:0]	TXQ Interrupt Status Flag for Channel 1 0: TXQ Interrupt flag not set 1: TXQ Interrupt flag set	R
31:8	—	These bits are read as 0.	R

The CFDTXQISTS register shows the status of the interrupt flag of the TXQ buffers.

TXQnISF[3:0] (n = 0, 1) bits (TXQ Interrupt Status Flag)

Each bit is set automatically when the corresponding bit is set in the TX Queue Interrupt Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue Interrupt Status Register.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

Bit position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Channel 0 TX Queue 3
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Channel 1 TX Queue 3

33.2.70 CFDTXQOFTISTS : TX Queue One Frame TX Interrupt Status Register

Base address: CANFD = 0x8002_0000

Offset address: 0x1194

Bit position:	31																								7				3			
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQ1OFTISF[3:0]				TXQ0OFTISF[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	TXQ0OFTISF[3:0]	TXQ One Frame TX Interrupt Status Flag for Channel 0 0: TXQ One Frame TX Interrupt flag not set 1: TXQ One Frame TX Interrupt flag set	R
7:4	TXQ1OFTISF[3:0]	TXQ One Frame TX Interrupt Status Flag for Channel 1 0: TXQ One Frame TX Interrupt flag not set 1: TXQ One Frame TX Interrupt flag set	R
31:8	—	These bits are read as 0.	R

The CFDTXQOFTISTS register shows the status of the One Frame TX Interrupt flag of the TXQ buffers.

TXQnOFTISF[3:0] (n = 0, 1) bits (TXQ One Frame TX Interrupt Status Flag)

Each bit is set automatically when the corresponding bit is set in the TX Queue One Frame TX Interrupt Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue One Frame TX Interrupt Status Register.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

Bit position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Channel 0 TX Queue 3
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Channel 1 TX Queue 3

33.2.71 CFDTXQOFRISTS : TX Queue One Frame RX Interrupt Status Register

Base address: CANFD = 0x8002_0000

Offset address: 0x1198

Bit position:	31																								6	4	2	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQ1OFRISF[2:0]		—	TXQ0OFRISF[2:0]
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	TXQ0OFRISF[2:0]	TXQ One Frame RX Interrupt Status Flag for Channel 0 0: TXQ One Frame RX Interrupt flag not set 1: TXQ One Frame RX Interrupt flag set	R
3	—	This bit is read as 0.	R
6:4	TXQ1OFRISF[2:0]	TXQ One Frame RX Interrupt Status Flag for Channel 1 0: TXQ One Frame RX Interrupt flag not set 1: TXQ One Frame RX Interrupt flag set	R
31:7	—	These bits are read as 0.	R

The CFDTXQOFRISTS register shows the status of the One Frame RX Interrupt flag of the TXQ buffers.

TXQnOFRISF[2:0] (n = 0, 1) bits (TXQ One Frame RX Interrupt Status Flag)

Each bit is set automatically when the corresponding bit is set in the TX Queue One Frame RX Interrupt Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue One Frame RX Interrupt Status Register.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

Bit position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Reserved
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Reserved

33.2.72 CFDTXQFSTS : TX Queue Full Status Register

Base address: CANFD = 0x8002_0000

Offset address: 0x119C

Bit position:	31																							7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXQ1FSF[3:0]				TXQ0FSF[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	TXQ0FSF[3:0]	TXQ Full Status Flag for Channel 0 0: TXQ Full flag not set 1: TXQ Full flag set	R
7:4	TXQ1FSF[3:0]	TXQ Full Status Flag for Channel 1 0: TXQ Full flag not set 1: TXQ Full flag set	R
31:8	—	These bits are read as 0.	R

The CFDTXQFSTS register shows the status of the Full Status flag bits of the TXQ buffers.

TXQnFSF[3:0] (n = 0, 1) bits (TXQ Full Status Flag)

Each bit is set automatically when the corresponding bit is set in the TX Queue Full Status Register.

Each bit is cleared automatically when the corresponding bit is cleared in the TX Queue Full Status Register.

This bit is cleared when the CAN-FD module is in GL_RESET mode.

Bit position	Corresponding TX Queue
0	Channel 0 TX Queue 0
1	Channel 0 TX Queue 1
2	Channel 0 TX Queue 2
3	Channel 0 TX Queue 3
4	Channel 1 TX Queue 0
5	Channel 1 TX Queue 1
6	Channel 1 TX Queue 2
7	Channel 1 TX Queue 3

33.2.73 CFDTHLCCn : TX History List Configuration/Control Register n (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x1200 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	THLDGE	THLDTE	THLIM	THLIE	—	—	—	—	—	—	—	THLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	THLE	TX History List Enable 0: TX History List disabled 1: TX History List enabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	THLIE	TX History List Interrupt Enable 0: TX History List Interrupt disabled 1: TX History List Interrupt enabled	R/W
9	THLIM	TX History List Interrupt Mode 0: Interrupt generated if TX History List level reaches ¾ of the TX History List depth 1: Interrupt generated for every successfully stored entry	R/W
10	THLDTE	TX History List Dedicated TX Enable 0: TX FIFO + TX Queue 1: Flat TX MB + TX FIFO + TX Queue	R/W
11	THLDGE	TX History List Dedicated Gateway Enable 0: Not dedicated Gateway FIFO + Gateway TX Queue 1: Dedicated Gateway FIFO + Gateway TX Queue	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTHLCCn register (n = 0, 1) configures the TX History List functions.

THLE bit (TX History List Enable)

The THLE bit enables the TX History List buffer when it is set.

You cannot write to this bit when the related CAN-FD channel is in CH_RESET or CH_SLEEP mode.

This bit is cleared automatically when the related CAN-FD channel is in CH_RESET mode.

THLIE bit (TX History List Interrupt Enable)

The THLIE bit enables the generation of the TX History List interrupt when it is set.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

THLIM bit (TX History List Interrupt Mode)

The THLIM bit selects the interrupt generation condition for the FIFO.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

THLDTE bit (TX History List Dedicated TX Enable)

The THLDTE bit selects the condition for storing an entry in the TX History List after successful transmission.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

THLDGE bit (TX History List Dedicated Gateway Enable)

The THLDGE bit selects the condition for storing an entry in the TX History List after successful transmission.

You cannot write to this bit when the CAN-FD module is in GL_SLEEP mode.

Do not write to this bit when the CAN-FD module is in GL_HALT or GL_OPERATION mode.

33.2.74 CFDTHLSTSn : TX History List Status Register n (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x1220 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	THLMC[5:0]						—	—	—	—	THLIF	THLELT	THLFLL	THLEMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	THLEMP	TX History List Empty 0: TX History List not empty 1: TX History List empty	R
1	THLFLL	TX History List Full 0: TX History List not full 1: TX History List full	R
2	THLELT	TX History List Entry Lost 0: No entry lost in TX History List 1: TX History List entry Lost	R/W
3	THLIF	TX History List Interrupt Flag 0: TX History List interrupt condition not satisfied 1: TX History List interrupt condition satisfied	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
13:8	THLMC[5:0]	TX History List Message Count Number of messages stored in TX History List	R
31:14	—	These bits are read as 0. The write value should be 0.	R/W

The CFDTHLSTSn register (n = 0, 1) shows the status of data stored in the TX History List buffer.

THLEMP bit (TX History List Empty)

The THLEMP bit is set automatically when the CPU has read all the entries from the TX History List buffer.

This bit is cleared automatically when the first entry is stored to the TX History List.

This bit is set automatically when:

- TX History List is disabled
- The related CAN-FD channel is in CH_RESET mode.

THLFLL bit (TX History List Full)

The THLFLL bit is set automatically when the number of entries in the TX History List buffer matches the TX History List depth.

Each TX History List can store up to 32 entries (each channel has a dedicated TX History List).

This bit is cleared automatically when:

- The number of entries in the TX History List buffer is less than the TX History List depth
- The TX History List is disabled
- The related CAN-FD channel is in CH_RESET mode.

THLELT bit (TX History List Entry Lost)

The THLELT bit is set when a new entry cannot be stored because the related TX History List buffer is already full.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

THLIF bit (TX History List Interrupt Flag)

The THLIF bit is set when the configured interrupt condition is satisfied.

Only write to this bit when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode. Writing 1 has no effect.

Do not use the bit clear instruction to clear this bit. Use the MOV instruction to ensure that only the specified bit is cleared. Other bits remain 1.

This bit is cleared:

- By writing 0 to it
- When the related CAN-FD channel is in CH_RESET mode.

This bit is automatically cleared in CH_RESET mode.

THLMC[5:0] bits (TX History List Message Count)

The THLMC[5:0] bits show the number of transmitted messages stored in the TX History List.

These bits are cleared automatically when the related CAN-FD channel is in CH_RESET mode.

33.2.75 CFDTHLACC0n : Channel n TX History List Access Register 0 (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x8000 + 0x08 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TMTS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TGW	—	—	—	—	—	BN[6:0]							BT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	BT[2:0]	Buffer Type 0 0 1: Flat TX message buffer 0 1 0: TX FIFO message buffer number and gateway FIFO message number 1 0 0: TX Queue message buffer number Others: Setting prohibited	R
9:3	BN[6:0]	Buffer Number Number of the message buffer	R
14:10	—	These bits are read as 0.	R
15	TGW	Transmit Gateway Buffer Indication 0: No transmission from gateway 1: Transmission from gateway	R
31:16	TMTS[15:0]	Transmit Timestamp Transmit timestamp value for software drivers	R

The CFDTHLACC0n register (n = 0, 1) provides access to the entry in the TX History List based on the read timestamp value.

BT[2:0] bits (Buffer Type)

The BT[2:0] bits indicate whether data has been stored following a transmission from a FIFO buffer, a TX Queue or a TX message buffer.

BN[6:0] bits (Buffer Number)

The BN[6:0] bits show the message buffer from which transmission was successfully completed. If a message from a Common FIFO is transmitted, then these bits show the message buffer that is linked to the Common FIFO for transmission.

TGW bit (Transmit Gateway Buffer Indication)

The TGW bit is automatically set to 1 when transmission is completed in GW mode.

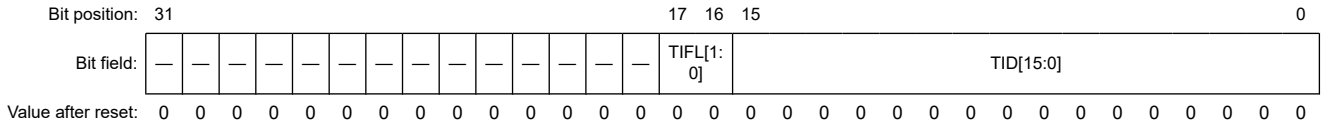
TMTS[15:0] bits (Transmit Timestamp)

The TMTS[15:0] bits indicate the timestamp for use by software drivers.

33.2.76 CFDTHLACC1n : Channel n TX History List Access Register 1 (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x8004 + 0x08 × n



Bit	Symbol	Function	R/W
15:0	TID[15:0]	Transmit ID These bits indicate that message buffer reference ID, TX FIFO reference ID, or AFL pointer field is stored for software drivers.	R
17:16	TIFL[1:0]	Transmit Information Label These bits indicate that message buffer information label, TX FIFO information label, or AFL information label is stored for software drivers.	R
31:18	—	These bits are read as 0.	R

The CFDTHLACC1n register (n = 0, 1) provides access to entry in the TX History List based on the read pointer value.

TID[15:0] bits (Transmit ID)

The TID[15:0] bits indicate whether the message buffer reference ID (CFDTMFDCTRn.TMPTR) or the TX FIFO reference ID (CFDCFFDCSTSn.CFPTR) is for use by software drivers.

When transmission in Gateway mode, these bits indicate the AFL pointer field (CFDGAFLP0n.GAFLPTR) instead of the message buffer reference ID (CFDTMFDCTRn.TMPTR).

TIFL[1:0] bits (Transmit Information Label)

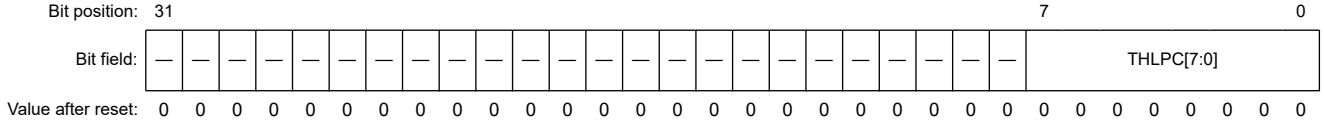
The TIFL[1:0] bits indicate whether the message buffer information label (CFDTMFDCTRn.TMIFL) or the TX FIFO information label (CFDCFFDCSTSn.CFIFL) is for use by software drivers.

When transmission in Gateway mode, these bits indicate the AFL pointer field (CFDGAFLMn.GAFLIFL1 and CFDGAFLP0n.GAFLIFL0) instead of the MB information label (CFDTMFDCTRn.TMIFL).

33.2.77 CFDTHLPCTRn : TX History List Pointer Control Register n (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x1240 + 0x04 × n



Bit	Symbol	Function	R/W
7:0	THLPC[7:0]	TX History List Pointer Control Increments the write pointer to the TX History List in the corresponding channel	W
31:8	—	The write value should be 0.	W

The CFDTHLPCTRn register (n = 0, 1) is used to increment the read pointer of the TX History List.

THLPC[7:0] bits (TX History List Pointer Control)

When 0xFF is written to the THLPC[7:0] bits, the read pointer of the TX History List is moved to the next TX History List entry address.

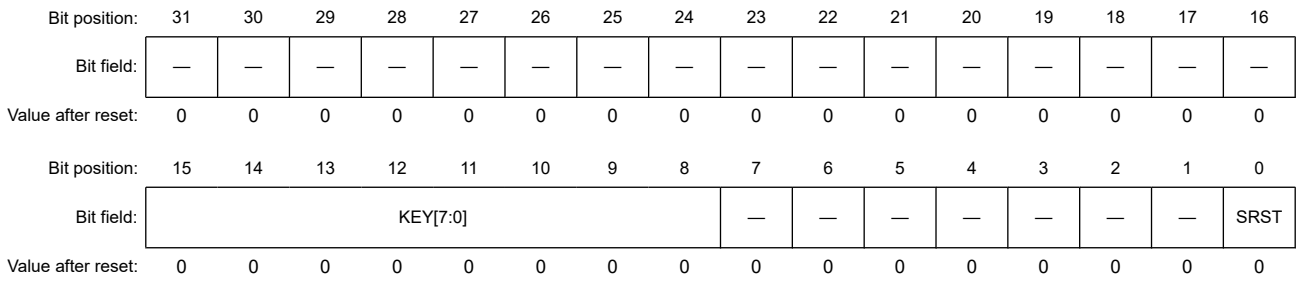
The read value from these bits is always 0x00. Only write to these bits when the related CAN-FD channel is in CH_HALT or CH_OPERATION mode.

Only write 0xFF to these registers when the corresponding TX History List is enabled and not empty.

33.2.78 CFDGRSTC : Global Reset Control Register

Base address: CANFD = 0x8002_0000

Offset address: 0x1380



Bit	Symbol	Function	R/W
0	SRST	Software Reset 0: Normal state 1: Software reset state	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
15:8	KEY[7:0]	Key Code These bits control the validity of rewriting of the SRST bit.	W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

SRST bit (Software Reset)

When the SRST bit is set, the CAN-FD module is in the same state as hardware reset. When a reset is required, write 1 then write 0 to this bit.

This bit is cleared when the CAN-FD module is in GL_SLEEP mode.

When this bit is cleared, the RAM initialization sequence does not operate. The configuration of RAM is performed by software.

The RAM is not initialized when software reset is performed during the initialization of RAM. Software must perform the initialization of RAM.

KEY[7:0] bits (Key Code)

When 0xC4 is written in the KEY[7:0] bits, a write to the SRST bit is valid.

The read value from these bits is always 0x00.

33.2.79 CFDGFCMC : Global Flexible CAN Mode Configuration Register

Base address: CANFD = 0x8002_0000

Offset address: 0x1384

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FLXC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FLXC0	Flexible CAN Mode between Channel 0 and Channel 1 0: Normal mode 1: Flexible CAN mode	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Flexible CAN mode configured in the CFDGFCMC register and Flexible transmission buffer assignment configured in the CFDGFTBAC register should not be used simultaneously.

FLXC0 bit (Flexible CAN Mode between Channel 0 and Channel 1)

When the FLXC0 bit is set, Channel 0 and Channel 1 of a CAN-FD module are in Flexible CAN mode.

Channel 1 uses TX/RX terminal of Channel 0. The TX/RX terminal of Channel 1 cannot be used.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

33.2.80 CFDGFTBAC : Global Flexible Transmission Buffer Assignment Configuration Register

Base address: CANFD = 0x8002_0000

Offset address: 0x138C

Bit position:	31																												3	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	FLXMB0[3:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	FLXMB0[3:0]	Flexible Transmission Buffer Assignment between Channel 0 and Channel 1 By setting these bits, the even channel can use the configured number of TX mailboxes of the odd channel 0x0: 0 0x1: 4 0x2: 8 0x3: 12 0x4: 16 0x5: 20 0x6: 24 0x7: 28 0x8: 32 Others: Setting prohibited	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Flexible transmission buffer assignment configured in the CFDGFTBAC register and Flexible CAN mode configured in the CFDGFCMC register should not be used simultaneously.

FLXMB0[3:0] bits (Flexible Transmission Buffer Assignment between Channel 0 and Channel 1)

Channel 0 can use the number TXMB of channel 1 from 0 to 32 by the configuration of these bits.

Only write to this bit when the CAN-FD module is in GL_RESET mode.

33.2.81 CFDGTSTCFG : Global Test Configuration Register

Base address: CANFD = 0x8002_0000

Offset address: 0x1308

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	RTMPS[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	C1ICBCE	C0ICBCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	C0ICBCE	Channel 0 Internal CAN Bus Communication Test Mode Enable 0: Channel 0 internal CAN bus communication disabled 1: Channel 0 internal CAN bus communication enabled	R/W
1	C1ICBCE	Channel 1 Internal CAN Bus Communication Test Mode Enable 0: Channel 1 internal CAN bus communication disabled 1: Channel 1 internal CAN bus communication enabled	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W
25:16	RTMPS[9:0]	RAM Test Mode Page Select Select a RAM test mode page	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The CFDGTSTCFG register is used to configure the CAN channels joining the internal CAN bus communication test mode and the RAM test mode page.

CnICBCE (n = 0, 1) bits (Channel n Internal CAN Bus Communication Test Mode Enable)

When the C0ICBCE and C1ICBCE bits are set and CAN-FD module is configured in the internal CAN bus communication test mode, then CAN channel n joins the internal CAN bus communication test mode operation.

Do not write to these bits when the CAN-FD module is in GL_RESET or GL_SLEEP mode.

Only write to these bits when the CAN-FD module is in GL_HALT mode.

These bits are cleared automatically when the CAN-FD module is in GL_RESET mode.

RTMPS[9:0] bits (RAM Test Mode Page Select)

The RTMPS[9:0] bits select the RAM page mode for CPU read/write access when the CAN-FD module is configured in RAM test mode.

See section 33.9.2.1. RAM Test Mode for the RAM test mode specification.

Do not write to these bits when the CAN-FD module is in GL_RESET or GL_SLEEP mode.

Only enter values from 0 to 15 (0x00F) for the AFL RAM and 16 to 76 (0x04C) for the message buffer RAM.

The setting range of these bits depends on the combination of parameters.

Only write to these bits when the CAN-FD module is in GL_HALT mode.

These bits are cleared automatically when the related CAN-FD channel is in GL_RESET mode.

33.2.82 CFDGTSTCTR : Global Test Control Register

Base address: CANFD = 0x8002_0000

Offset address: 0x130C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	RTME	—	ICBCTME
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ICBCTME	Internal CAN Bus Communication Test Mode Enable 0: Internal CAN Bus Communication test mode disabled 1: Internal CAN Bus Communication test mode enabled	R/W
1	—	This bit is read as 0. The write value should be 0.	R/W
2	RTME	RAM Test Mode Enable 0: RAM test mode disabled 1: RAM test mode enabled	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

The CFDGTSTCTR register is used to control the global test modes of the CAN-FD module.

ICBCTME bit (Internal CAN Bus Communication Test Mode Enable)

When the ICBCTME bit is set, internal CAN bus communication is enabled for the CAN channels that are configured for internal CAN bus communication participation. See section 33.9.2.2. Internal CAN Bus Communication Mode for the specification of internal CAN bus communication test mode.

Only write to this bit when the CAN-FD module is in GL_HALT mode.

Clear this bit when the CAN-FD module is in GL_HALT mode.

This bit is cleared automatically when the CAN-FD module is in GL_RESET mode.

RTME bit (RAM Test Mode Enable)

When the RTME bit is set, the CAN-FD module is configured in RAM test mode. See section 33.9.2.1. RAM Test Mode for RAM test mode specification.

Only write to this bit when the CAN-FD module is in GL_HALT mode.

Clear this bit when the CAN-FD module is in GL_HALT mode.

This bit is cleared automatically when the CAN-FD module is in GL_RESET mode.

33.2.83 CFDGFDCFG : Global FD Configuration Register

Base address: CANFD = 0x8002_0000

Offset address: 0x1314

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TSCCFG[1:0]		—	—	—	—	—	—	—	RPED
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RPED	RES Bit Protocol Exception Disable 0: Protocol exception event detection enabled 1: Protocol exception event detection disabled	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
9:8	TSCCFG[1:0]	Timestamp Capture Configuration 0 0: Timestamp capture at the sample point of SOF (start of frame) 0 1: Timestamp capture at frame valid indication 1 0: Timestamp capture at the sample point of RES bit 1 1: Reserved	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

RPED bit (RES Bit Protocol Exception Disable)

The RPED bit configures the protocol exception event handling according to ISO 11898-1.

When this bit is enabled, the protocol exception event detection is disabled, and the protocol controller transmits an error frame when the protocol exception event is detected (RES bit is sampled recessive).

Only write to this bit when the CAN-FD module is in GL_RESET mode.

TSCCFG[1:0] bits (Timestamp Capture Configuration)

The TSCCFG[1:0] bits configure the different capture points of the timestamp for transmission and reception.

When CFDGFDCFG.TSCCFG[1:0] = 10b, the timestamp capture is performed for CAN-FD frames at RES bit and for Classical frames at the start of frame.

Only write to these bits when the CAN-FD module is in GL_RESET mode.

33.2.84 CFDGLOCKK : Global Lock Key Register

Base address: CANFD = 0x8002_0000

Offset address: 0x131C

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	LOCK[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	LOCK[15:0]	Lock Key Key bits for unlocking the protection of test modes	W
31:16	—	The write value should be 0.	W

The CFDGLOCKK register is a write-only register that is used to unlock the protection for special test bits.

See section 33.9.2. Global Test Modes for Lock key specification.

LOCK[15:0] bits (Lock Key)

The unlock key sequence must be written in the LOCK[15:0] bits to configure the CAN-FD module in RAM test modes.

The read value from these bits is always 0x0000.

You cannot write to these bits when the CAN-FD module is in GL_SLEEP or GL_RESET mode.

Do not write to these bits when the CAN-FD module is in GL_OPERATION mode.

33.2.85 CFDRPGACCn : RAM Test Page Access Register n (n = 0 to 63)

Base address: CANFD = 0x8002_0000

Offset address: 0x8400 + 0x04 × n

Bit position:	31																													0
Bit field:	RDTA[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RDTA[31:0]	RAM Data Test Access RAM data bytes	R/W

RDTA[31:0] bits (RAM Data Test Access)

Data can be read from or written into the RDTA[31:0] bits when the CAN-FD module is configured in RAM test mode.

Only write to this bit when the CAN-FD module is in GL_HALT mode and RAM test mode is enabled.

If data is read when RAM test mode is not enabled, then it is always read as 0x00000000.

Software data should be read/written in the RAM Test Page Access registers during RAM test mode.

33.2.86 CFDCnBLCT : Channel n Bus Load Control Register (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x1418 + 0x20 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	BLCLD	—	—	—	—	—	—	—	BLCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BLCE	Bus Load Counter Enable 0: Bus load counter disabled 1: Bus load counter enabled	R/W

Bit	Symbol	Function	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W
8	BLCLD	Bus Load Counter Load When CFDCnBLCT.BLCLD is set, it is reset after a bus load counter value is loaded.	W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

BLCE bit (Bus Load Counter Enable)

The BLCE bit indicates the enabling signal of a bus load counter. Write to this bit after setting up the CFDCnNCFG register.

When this bit is 0, the bus load counter stops but the counter is not clear.

The bit is cleared when the related CAN-FD channel is in CH_RESET mode.

Do not write to this bit in CH_SLEEP mode.

BLCLD bit (Bus Load Counter Load)

When the BLCLD bit is set, the bus load counter value is loaded into CFDCnBLSTS.BLC and the bus load counter is reset.

The read value is always 0.

33.2.87 CFDCnBLSTS : Channel n Bus Load Status Register (n = 0, 1)

Base address: CANFD = 0x8002_0000

Offset address: 0x141C + 0x20 × n

Bit position:	31																										3		0
Bit field:	BLC[28:0]																										—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	—	These bits are read as 0.	R
31:3	BLC[28:0]	Bus Load Counter Status These bits indicate the bus load counter value.	R

BLC[28:0] bits (Bus Load Counter Status)

When the CFDCnBLCT.BLCLD bit is set, the bus load counter value is loaded to BLC[28:0] bits and the bus load counter is reset.

These bits are set only by the CAN-FD module. Writing any value has no effect.

33.2.88 Identifier Bits Alignment

Standard Identifier (11 bits) format: ID28 – ID18 is aligned to b10 to b0, b11 to b28 should be 0.

Extended Identifier (29 bits) format: ID28 – ID0 is aligned to b28 – b0

Table 33.11 Standard Identifier (11-bit format)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IDE = 0	RTR	—	0	0	0	0	0	0	0	0	0	0	0	0	0

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
0	0	0	0	0	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18

Table 33.12 Extended Identifier (29-bit format)

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
IDE = 1	RTR	—	ID28	ID27	ID26	ID25	ID24	ID23	ID22	ID21	ID20	ID19	ID18	ID17	ID16

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
ID15	ID14	ID13	ID12	ID11	ID10	ID9	ID8	ID7	ID6	ID5	ID4	ID3	ID2	ID1	ID0

33.2.89 Message Buffer Component Structure

33.2.89.1 Start Addresses

The start address for each of the Message Buffer component is calculated using the number of related Message Buffer components and the number of channels.

number_of_channels	2
number_of_RMBCPs_per_channel	16
number_of_RFMBCPs	8
number_of_CFMBCPs_per_channel	3
number_of_TMBCPs_per_channel	64

The start addresses for each register in the Message Buffer component are depicted in Table 33.13.

Table 33.13 Message Buffer Component Register Start Addresses

b = Message buffer component index	MBCP	Register	p	Start address n = 0, 1
0 to number_of_RMBCPs_per_channel - 1	RMBCPb[i]	RMID	x	0x2000 + b × 0x0080 + n × 0x800
		RMPTR	x	0x2004 + b × 0x0080 + n × 0x800
		RMFDSTS	x	0x2008 + b × 0x0080 + n × 0x800
		RMDFp	0 to 15	0x200C + p × 0x0004 + b × 0x0080 + n × 0x800
0 to number_of_RFMBCPs - 1	RFMBCPb[i]	RFID	x	0x6000 + b × 0x0080
		RFPTR	x	0x6004 + b × 0x0080
		RFFDSTS	x	0x6008 + b × 0x0080
		RFDFp	0 to 15	0x600C + p × 0x0004 + b × 0x0080
0 to number_of_CFMBCPs_per_channel - 1	CFMBCPb[i]	CFID	x	0x6400 + b × 0x0080 + n × 0x180
		CFPTR	x	0x6404 + b × 0x0080 + n × 0x180
		CFFDCSTS	x	0x6408 + b × 0x0080 + n × 0x180
		CFDFp	0 to 15	0x640C + p × 0x0004 + b × 0x0080 + n × 0x180
0 to number_of_TMBCPs_per_channel - 1	TMBCPb[i]	TMID	x	0x10000 + b × 0x0080 + n × 0x2000
		TMPTR	x	0x10004 + b × 0x0080 + n × 0x2000
		TMFDCTR	x	0x10008 + b × 0x0080 + n × 0x2000
		TMDFp	0 to 15	0x1000C + p × 0x0004 + b × 0x0080 + n × 0x2000

The message buffer configuration consists of four types of Message Buffer components:

- RX Message Buffer Component (CFDRMBCPb[i])
- RX FIFO Access Message Buffer Component (CFDRFMBCPb[i])
- Common FIFO Access Message Buffer Component (CFDCFMBCPb[i])
- TX Message Buffer Component (CFDTMBCPb[i]).

Where b = the Message Buffer component index having a range that varies based on the type of Message Buffer component and i = channel index that has a range from 0 to n.

For a summary of this configuration, see Figure 33.28. For a detailed description of the number of and the different types of message buffers, see section 33.6. FIFO Buffers and Normal Message Buffer Configuration

As described in section 33.2. Register Descriptions, each Message Buffer component consists of the following registers:

- Identifier (ID)
- Pointer (PTR)
- Data Field (DFp).

p is the Data Field register index having a range that varies based on the type of Message Buffer component.

Rc is the Message Buffer Component register where c = Message Buffer Component register index having a range that varies based on the type of Message Buffer component.

A description of the registers, their associated bits and their accessibility are shown below the summary and detailed figures of each component.

In each of the figures, a cell that contains '-' means reserved and has the same behavior as reserved bits for registers in section 33.2.89. Message Buffer Component Structure.

33.2.89.2 CFDRMIDn : RX Message Buffer ID Register n (n = 0 to 31)

Base address: CANFD = 0x8002_0000

Offset address: 0x2000 + 0x080 × n

Bit position:	31	30	29 to 28	27 to 0
Bit field:	RMIDE	RMRTR	—	RMID[28:0]
Value after reset:	0	0	0 0	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
28:0	RMID[28:0]	RX Message Buffer ID Field STD-ID/EXT-ID fields	R
29	—	These bits are read as 0.	R
30	RMRTR	RX Message Buffer RTR Bit 0: Data frame 1: Remote frame	R
31	RMIDE	RX Message Buffer IDE Bit 0: STD-ID is stored 1: EXT-ID is stored	R

The CFDRMIDn register (n = 0 to 31) stores the ID field, IDE bit, and RTR bit of the received message.

RMID[28:0] bits (RX Message Buffer ID Field)

The RMID[28:0] are the bits of the STD-ID/EXT-ID fields of the message stored in the RX message buffer.

For alignment of these bits in standard and extended frame format, see section 33.2.88. Identifier Bits Alignment

See section 33.2.89.1. Start Addresses for details on how to interpret the structure of this buffer component.

RMRTR bit (RX Message Buffer RTR Bit)

The RMRTR bit shows whether a data frame or a remote frame was stored in the RX message buffer.

Note: There are no remote frames in CAN-FD format. When a CAN-FD frame is received, the register reflects the state of the received value (the RRS bit in FD frame format).

RMIDE bit (RX Message Buffer IDE Bit)

The RMIDE bit shows whether message with Standard Identifier or Extended Identifier was stored in the RX message buffer.

33.2.89.3 CFDRMPTRn : RX Message Buffer Pointer Register (n = 0 to 31)

Base address: CANFD = 0x8002_0000

Offset address: 0x2004 + 0x080 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RMDLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	RMTS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RMTS[15:0]	RX Message Buffer Timestamp Field Timestamp value stored for the message in the RX message buffer	R/W
27:16	—	These bits are read as 0. The write value should be 0.	R/W
31:28	RMDLC[3:0]	RX Message Buffer DLC Field Number of data bytes received in a CAN frame.	R/W

The CFDRMPTRn register (n = 0 to 31) stores the DLC and Timestamp fields for the received message.

RMTS[15:0] bits (RX Message Buffer Timestamp Field)

The RMTS[15:0] bits store the timestamp value taken at the capture point as configured by CFDGFDCFG.TSCCFG of the received message.

RMDLC[3:0] bits (RX Message Buffer DLC Field)

The RMDLC[3:0] bits store the number of data bytes that were received in the RX message buffer.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes that were received.

Note: The maximum capacity of the buffer belongs to CFDRMNB.RMPLS and this is not available in the classical CAN function.

33.2.89.4 CFDRMFDSTSn : RX Message Buffer CAN-FD Status Register (n = 0 to 31)

Base address: CANFD = 0x8002_0000

Offset address: 0x2008 + 0x080 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RMPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	RMIFL[1:0]		—	—	—	—	—	RMFDF	RMBRS	RMESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RMESI*1	Error State Indicator bit 0: CAN-FD frame received from error active node 1: CAN-FD frame received from error passive node	R/W
1	RMBRS*1	Bit Rate Switch bit 0: CAN-FD frame received with no bit rate switch 1: CAN-FD frame received with bit rate switch	R/W

Bit	Symbol	Function	R/W
2	RMFDF*1	CAN FD Format bit 0: Non CAN-FD frame received 1: CAN-FD frame received	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
9:8	RMIFL[1:0]	RX Message Buffer Information Label Field	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
31:16	RMPTR[15:0]	RX Message Buffer Pointer Field	R/W

Note 1. This bit is not available in the classical CAN function.

The CFDRMFDSTSn register (n = 0 to 31) shows the status of the FDF, BRS, and ESI bits, and pointer of the received CAN-FD frame.

RMESI bit (Error State Indicator bit)

The RMESI bit has the same value as the ESI bit of the received CAN-FD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

RMBRS bit (Bit Rate Switch bit)

The RMBRS bit has the same value as the BRS bit of the received CAN-FD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

RMFDF bit (CAN FD Format bit)

The RMFDF bit has the same value as the FDF bit of the received CAN-FD frame.

Note: This bit is not available in the classical CAN function.

RMIFL[1:0] bits (RX Message Buffer Information Label Field)

The RMIFL[1:0] bits store the information label value from the related Global Acceptance Filter List entry.

RMPTR[15:0] bits (RX Message Buffer Pointer Field)

The RMPTR[15:0] bits store the pointer value from the related Global Acceptance Filter List entry.

33.2.89.5 CFDRMDFp_n : RX Message Buffer Data Field p Register n (p = 0 to 15, n = 0 to 31)

Base address: CANFD = 0x8002_0000

Offset address: 0x200C + 0x004 × p + 0x080 × n

Bit position:	31 24	23 16	15 8	7 0
Bit field:	RMDB_HH[7:0]	RMDB_HL[7:0]	RMDB_LH[7:0]	RMDB_LL[7:0]
Value after reset:	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0

Bit	Symbol	Function	R/W
7:0	RMDB_LL[7:0]	RX Message Buffer Data Byte (4 × p)	R/W
15:8	RMDB_LH[7:0]	RX Message Buffer Data Byte (4 × p + 1)	R/W
23:16	RMDB_HL[7:0]	RX Message Buffer Data Byte (4 × p + 2)	R/W
31:24	RMDB_HH[7:0]*1	RX Message Buffer Data Byte (4 × p + 3)	R/W

Note 1. These bits are not available in the classical CAN function.

The CFDRMDFp_n register (p = 0 to 15, n = 0 to 31) stores the data bytes (4 × p) to data bytes (4 × p + 3) of the received message.

RMDB_LL[7:0] bits (RX Message Buffer Data Byte (4 × p))

The RMDB_LL[7:0] bits store data bytes (4 × p) of the message in the RX message buffer. Unused data bytes are filled with 0x00.

RMDB_LH[7:0] bits (RX Message Buffer Data Byte (4 × p + 1))

The RMDB_LH[7:0] bits store data bytes (4 × p + 1) of the message in the RX message buffer. Unused Data Bytes are filled with 0x00.

RMDB_HL[7:0] bits (RX Message Buffer Data Byte (4 × p + 2))

The RMDB_HL[7:0] bits store data bytes (4 × p + 2) of the message in the RX message buffer. Unused data bytes are filled with 0x00.

RMDB_HH[7:0] bits (RX Message Buffer Data Byte (4 × p + 3))

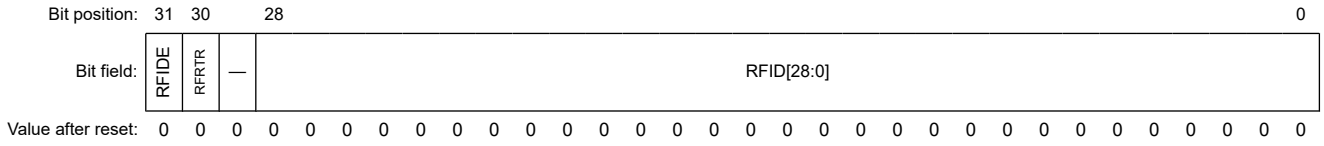
The RMDB_HH[7:0] bits store data bytes (4 × p + 3) of the message in the RX message buffer. Unused data bytes are filled with 0x00.

These bits are not available in the classical CAN function.

33.2.89.6 CFDRFIDn : RX FIFO Access ID Register n (n = 0 to 7)

Base address: CANFD = 0x8002_0000

Offset address: 0x6000 + 0x080 × n



Bit	Symbol	Function	R/W
28:0	RFID[28:0]	RX FIFO Buffer ID Field STD-ID/EXT-ID fields	R
29	—	This bit is read as 0.	R
30	RFRTR	RX FIFO Buffer RTR bit 0: Data frame 1: Remote frame	R
31	RFIDE	RX FIFO Buffer IDE bit 0: STD-ID has been received 1: EXT-ID has been received	R

The CFDRFIDn register (n = 0 to 7) stores the ID field, IDE bit, and RTR bit of the message.

RFID[28:0] bits (RX FIFO Buffer ID Field)

The RFID[28:0] bits are the bits of the STD-ID/EXT-ID fields of the message in the FIFO buffer.

For alignment of these bits in standard and extended frame format, see section 33.2.88. Identifier Bits Alignment.

RFRTR bit (RX FIFO Buffer RTR bit)

The RFRTR bit shows whether a data frame or a remote frame was stored in the FIFO buffer.

Note: There are no remote frames in CAN-FD format. When a CAN-FD frame was received, the register reflects the state of the received value (RRS bit in FD frame format).

RFIDE bit (RX FIFO Buffer IDE bit)

The RFIDE bit shows whether message with the Standard Identifier or Extended Identifier was received in the FIFO buffer.

33.2.89.7 CFDRFPTRn : RX FIFO Access Pointer Register n (n = 0 to 7)

Base address: CANFD = 0x8002_0000

Offset address: 0x6004 + 0x080 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RFDLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	RFTS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RFTS[15:0]	RX FIFO Timestamp Value Timestamp value of the received CAN frame	R
27:16	—	These bits are read as 0.	R
31:28	RFDLC[3:0]	RX FIFO Buffer DLC Field Number of data bytes received in a CAN frame	R

The CFDRFPTRn register (n = 0 to 7) stores the DLC and Timestamp fields for the received message.

RFTS[15:0] bits (RX FIFO Timestamp Value)

The RFTS[15:0] bits store the timestamp value taken at the capture point as configured by the CFDGFDCFG.TSCCFG bit of the received message.

RFDLC[3:0] bits (RX FIFO Buffer DLC Field)

The RFDLC[3:0] bits store the number of data bytes that were received in the RX FIFO buffer.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes that were received.

33.2.89.8 CFDRFFDSTSn : RX FIFO Access CAN-FD Status Register n (n = 0 to 7)

Base address: CANFD = 0x8002_0000

Offset address: 0x6008 + 0x080 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CFDRFPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	RFIFL[1:0]		—	—	—	—	—	RFFDF	RFBRS	RFESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RFESI*1	Error State Indicator bit 0: CAN-FD frame received from error active node 1: CAN-FD frame received from error passive node	R
1	RFBRS*1	Bit Rate Switch bit 0: CAN-FD frame received with no bit rate switch 1: CAN-FD frame received with bit rate switch	R
2	RFFDF*1	CAN FD Format bit 0: Non CAN-FD frame received 1: CAN-FD frame received	R
7:3	—	These bits are read as 0.	R
9:8	RFIFL[1:0]	RX FIFO Buffer Information Label Field	R
15:10	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
31:16	CFDRFPTR[15:0]	RX FIFO Buffer Pointer Field	R

Note 1. This bit is not available in the classical CAN function.

The CFDRFFDSTSn register (n = 0 to 7) shows the status of the FDF, BRS, and ESI bits, including the pointer of the received CAN-FD frame.

RFESI bit (Error State Indicator bit)

The RFESI bit has the same value as the ESI bit of the received CAN-FD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

RFBRS bit (Bit Rate Switch bit)

The RFBRS bit has the same value as the BRS bit of the received CAN-FD frame.

When the received FDF bit is 0, this means a CAN2.0 frame is received and 0 is stored to this bit.

Note: This bit is not available in the classical CAN function.

RFFDF bit (CAN FD Format bit)

The RFFDF bit has the same value as the FDF bit of the received CAN-FD frame.

Note: This bit is not available in the classical CAN function.

RFIFL[1:0] bits (RX FIFO Buffer Information Label Field)

The RFIFL[1:0] bits store the information label value from the related Global Acceptance Filter List entry.

CFDRFPTR[15:0] bits (RX FIFO Buffer Pointer Field)

The CFDRFPTR[15:0] bits store the pointer value from the related Global Acceptance Filter List entry.

33.2.89.9 CFDRFDFpn : RX FIFO Access Data Field p Register n (p = 0 to 15, n = 0 to 7)

Base address: CANFD = 0x8002_0000

Offset address: 0x600C + 0x004 × p + 0x080 × n

Bit position:	31							24	23							16	15							8	7							0
Bit field:	RFDB_HH[7:0]								RFDB_HL[7:0]								RFDB_LH[7:0]								RFDB_LL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	RFDB_LL[7:0]	RX FIFO Buffer Data Byte (4 × p)	R
15:8	RFDB_LH[7:0]	RX FIFO Buffer Data Byte (4 × p + 1)	R
23:16	RFDB_HL[7:0]	RX FIFO Buffer Data Byte (4 × p + 2)	R
31:24	RFDB_HH[7:0]	RX FIFO Buffer Data Byte (4 × p + 3)	R

The CFDRFDFpn register (p = 0 to 15, n = 0 to 7) stores data bytes (4 × p) to data bytes (4 × p + 3) of the received message.

RFDB_LL[7:0] bits (RX FIFO Buffer Data Byte (4 × p))

The RFDB_LL[7:0] bits store data bytes (4 × p) of the message present in the FIFO buffer. Unused data bytes are filled with 0x00 according to the configured data payload size CFDRFCCn.RFPLS.

RFDB_LH[7:0] bits (RX FIFO Buffer Data Byte (4 × p + 1))

The RFDB_LH[7:0] bits store data bytes (4 × p + 1) of the message present in the FIFO buffer. Unused data bytes are filled with 0x00.

RFDB_HL[7:0] bits (RX FIFO Buffer Data Byte (4 × p + 2))

The RFDB_HL[7:0] bits store data bytes (4 × p + 2) of the message present in the FIFO buffer. Unused data bytes are filled with 0x00.

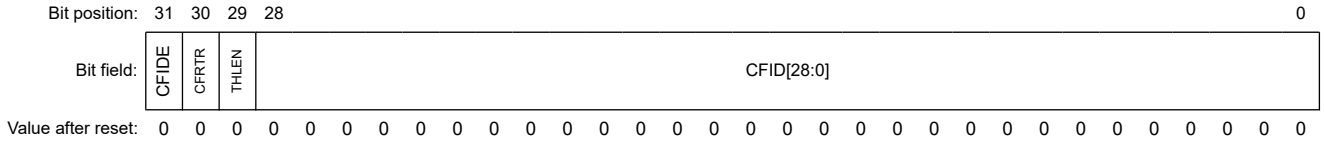
RFDB_HH[7:0] bits (RX FIFO Buffer Data Byte (4 × p + 3))

The RFDB_HH[7:0] bits store data bytes (4 × p + 3) of the message present in the FIFO buffer. Unused data bytes are filled with 0x00.

33.2.89.10 CFDCFIDn : Common FIFO Access ID Register n (n = 0 to 5)

Base address: CANFD = 0x8002_0000

Offset address: 0x6400 + 0x080 × n



Bit	Symbol	Function	R/W
28:0	CFID[28:0]	Common FIFO Buffer ID Field STD-ID/EXT-ID fields	R/W
29	THLEN	THL Entry Enable TX FIFO mode: 0: Entry is not to be stored in THL after successful TX 1: Entry is to be stored in THL after successful TX RX FIFO mode: Reserved, this bit is read as 0.	R/W
30	CFRTR	Common FIFO Buffer RTR bit 0: Data frame 1: Remote frame	R/W
31	CFIDE	Common FIFO Buffer IDE bit 0: STD-ID is to be transmitted or has been received 1: EXT-ID is to be transmitted or has been received	R/W

The CFDCFIDn register (n = 0 to 5) stores the ID field, IDE bit, and RTR bit of the message.

In TX mode, you can read data from the FIFO, only for the current entry based on the write pointer value, and not for the other entries.

CFID[28:0] bits (Common FIFO Buffer ID Field)

The CFID[28:0] bits are the bits of the STD-ID/EXT-ID fields of the message in the FIFO buffer.

For alignment of these bits in standard and extended frame format, see section 33.2.88. Identifier Bits Alignment.

In TX mode, you can write and read data from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

THLEN bit (THL Entry Enable)

The THLEN bit controls the storage of an entry corresponding to the transmitted message in the TX History list at the end of a successful transmission.

In TX mode, you can write and read data from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

CFRTR bit (Common FIFO Buffer RTR bit)

The CFRTR bit selects whether a data frame or a remote frame is to be transmitted from or was received in the FIFO buffer.

Note: There are no remote frames in CAN FD format. When a CAN-FD frame is received (RX mode), the register reflects the state of the received value (RRS bit in FD frame format). When CAN-FD transmission (TX or GW mode CFDCFID.CFFDF = 1), the bit is always transmitted dominant (data frame).

In TX mode, you can write and read data from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

CFIDE bit (Common FIFO Buffer IDE bit)

The CFIDE bit selects whether a message with EXT-ID or STD-ID is to be transmitted from or was received in the FIFO buffer.

In TX mode, you can write and read data from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

33.2.89.11 CFDCFPTRn : Common FIFO Access Pointer Register n (n = 0 to 5)

Base address: CANFD = 0x8002_0000

Offset address: 0x6404 + 0x080 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CFDLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	CFTS[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CFTS[15:0]	Common FIFO Timestamp Value Timestamp value of the received CAN frame (FIFO in RX mode).	R/W
27:16	—	These bits are read as 0. The write value should be 0.	R/W
31:28	CFDLC[3:0]	Common FIFO Buffer DLC Field Number of data bytes received in a CAN frame, or transmitted in a CAN frame.	R/W

The CFDCFPTRn register (n = 0 to 5) stores the DLC and Timestamp fields.

In TX mode, you can read data from the FIFO buffer, only for the current entry based on the write pointer value, and not for the other entries.

CFTS[15:0] bits (Common FIFO Timestamp Value)

The CFTS[15:0] bits store the timestamp value taken at the capture point as configured by the CFDGFDCFG.TSCCFG bit of the received message (if FIFO is configured in RX mode).

In TX mode, you can read and write data from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

CFDLC[3:0] bits (Common FIFO Buffer DLC Field)

The CFDLC[3:0] bits store the number of data bytes that were received in the FIFO buffer or are to be transmitted.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes.

In TX mode, you can read and write data from the FIFO buffers. Do not read data for the other entries in the FIFO when configured in TX mode.

In RX mode, you can only read data from the FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

33.2.89.12 CFDCFFDCSTSn : Common FIFO Access CAN-FD Control/Status Register n (n = 0 to 5)

Base address: CANFD = 0x8002_0000

Offset address: 0x6408 + 0x080 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CFPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CFIFL[1:0]		—	—	—	—	—	CFFDF	CFBRS	CFESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CFESI*1	Error State Indicator bit 0: CAN-FD frame received or to transmit by error active node 1: CAN-FD frame received or to transmit by error passive node	R/W
1	CFBRS*1	Bit Rate Switch bit 0: CAN-FD frame received or to transmit with no bit rate switch 1: CAN-FD frame received or to transmit with bit rate switch	R/W
2	CFFDF*1	CAN FD Format bit 0: Non CAN-FD frame received or to transmit 1: CAN-FD frame received or to transmit	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W
9:8	CFIFL[1:0]	COMMON FIFO Buffer Information Label Field	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W
31:16	CFPTR[15:0]	Common FIFO Buffer Pointer Field	R/W

Note 1. This bit is not available in the classical CAN function.

The CFDCFFDCSTSn register (n = 0 to 5) shows the status of the FDF, BRS, and ESI bits, including the pointer of the received CAN-FD frame or the CAN-FD frame to transmit.

In TX mode, you can read data from the FIFO, only for the current entry based on the write pointer value, and not for the other entries.

CFESI bit (Error State Indicator bit)

In TX mode, you can read and write data from FIFO buffers. In this mode, when the CAN-FD module is not in error passive, the CFESI bit equals the write value. Otherwise, it is a don't care and the bit is transmitted as 1 on the CAN bus, indicating this is an error passive node.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, the CFESI bit is updated with the ESI bit value of the CAN-FD frame when it has been received, indicating the error state of the transmitting node. In RX or GW mode, 0 is stored to this bit when the received FDF bit is 0, this means a CAN 2.0 frame is received.

Note: This bit is not available in the classical CAN function.

CFBRS bit (Bit Rate Switch bit)

In TX mode, you can read and write data from FIFO buffers. In this mode, the CAN-FD module either transmits a 0 to indicate no bit rate switch in the frame to be transmitted or a 1 to indicate a bit rate switch in the frame to be transmitted.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, the CFBRS bit is updated with the BRS bit value of the CAN-FD frame when it has been received, indicating whether there is a bit rate switch (1) or (0) on the CAN-FD frame.

In RX or GW mode, 0 is stored to the CFBRS bit when the received FDF bit is 0, this means a CAN 2.0 frame is received.

Note: This bit is not available in the classical CAN function.

CFFDF bit (CAN FD Format bit)

In TX mode, you can read and write data from FIFO buffers. In this mode, the CAN-FD module either transmits a 0 to indicate a CAN 2.0 frame is to be transmitted or a 1 to indicate a CAN-FD frame is to be transmitted.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, the CFFDF bit is updated with the FDF bit value of the CAN frame when it has been received, indicating whether it is a CAN 2.0 frame (0) or a CAN-FD frame (1).

Note: This bit is not available in the classical CAN function.

CFIFL[1:0] bits (COMMON FIFO Buffer Information Label Field)

If the Common FIFO is configured in TX mode, the value programmed in CFDCFFDCSTSn.CFIFL[1:0] is stored together with additional message information, to the TX History List after successful transmission of the message.

The information label value from the related Global Acceptance Filter List entry is stored in these bits (if FIFO is configured in either RX or GW mode).

In TX mode, you can read and write data from FIFO buffers.

In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

CFPTR[15:0] bits (Common FIFO Buffer Pointer Field)

If the Common FIFO is configured in TX mode, the value programmed in CFDCFFDCSTSn.CFPTR[15:0] is stored together with additional message information, to the TX History List after successful transmission of the message.

The pointer value from the related Global Acceptance Filter List entry is stored in these bits (if FIFO is configured in either RX or GW mode).

In TX mode, you can read and write data from FIFO buffers.

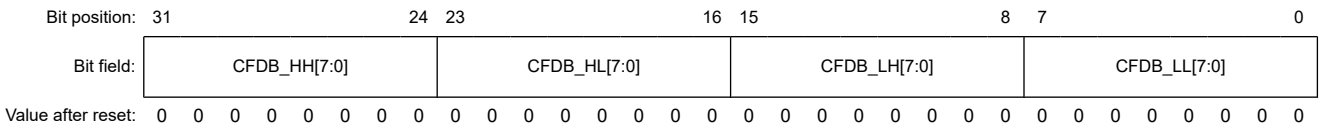
In RX mode, you can only read data from FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

33.2.89.13 CFDCFDFpn : Common FIFO Access Data Field p Register n (p = 0 to 15, n = 0 to 5)

Base address: CANFD = 0x8002_0000

Offset address: 0x640C + 0x004 × p + 0x080 × n



Bit	Symbol	Function	R/W
7:0	CFDB_LL[7:0]	Common FIFO Buffer Data Byte (4 × p)	R/W
15:8	CFDB_LH[7:0]	Common FIFO Buffer Data Byte (4 × p + 1)	R/W
23:16	CFDB_HL[7:0]	Common FIFO Buffer Data Byte (4 × p + 2)	R/W
31:24	CFDB_HH[7:0]	Common FIFO Buffer Data Byte (4 × p + 3)	R/W

The CFDCFDFpn register (p = 0 to 15, n = 0 to 5) stores data bytes (4 × p) to data bytes (4 × p + 3) of the message.

In TX mode, you can read data from the FIFO, only for the current entry based on the write pointer value, and not for the other entries.

CFDB_LL[7:0] bits (Common FIFO Buffer Data Byte (4 × p))

The CFDB_LL[7:0] bits store data bytes (4 × p) of the message present in the FIFO buffer.

In TX mode, you can read and write data from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCCn.CFPLS.*1

CFDB_LH[7:0] bits (Common FIFO Buffer Data Byte (4 × p + 1))

The CFDB_LH[7:0] bits store data bytes (4 × p + 1) of the message present in the FIFO buffer.

In TX mode, you can read and write data from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCCn.CFPLS.*1

CFDB_HL[7:0] bits (Common FIFO Buffer Data Byte (4 × p + 2))

The CFDB_HL[7:0] bits store data bytes (4 × p + 2) of the message present in the FIFO buffer.

In TX mode, you can read and write data from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCCn.CFPLS.*1

CFDB_HH[7:0] bits (Common FIFO Buffer Data Byte (4 × p + 3))

The CFDB_HH[7:0] bits store data bytes (4 × p + 3) of the message present in the FIFO buffer.

In TX mode, you can read and write data from the FIFO buffers.

In RX mode, you can only read data from the FIFO buffers.

In GW mode, you cannot write data to the FIFO buffers.

In RX or GW mode, unused data bytes are filled with 0x00, according to their configured data payload size CFDCFCCn.CFPLS.*1

Note 1. In RX or GW mode, unused data bytes are filled with 0x00 according to the configured data payload size CFDCFCCn.CFPLS, which is a CAN-FD feature not found in classical CAN.

33.2.89.14 CFDTMIDn : TX Message Buffer ID Register n (n = 0 to 127)

Base address: CANFD = 0x8002_0000

Offset address: 0x1_0000 + 0x080 × n

Bit position:	31	30	29	28																										0
Bit field:	TMIDE	TMRTR	THLEN	TMID[28:0]																										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
28:0	TMID[28:0]	TX Message Buffer ID Field STD-ID/EXT-ID fields	R/W
29	THLEN	Tx History List Entry 0: Entry not stored in THL after successful TX 1: Entry stored in THL after successful TX	R/W
30	TMRTR	TX Message Buffer RTR bit 0: Data frame 1: Remote frame	R/W
31	TMIDE	TX Message Buffer IDE bit 0: STD-ID is transmitted 1: EXT-ID is transmitted	R/W

The CFDTMIDn register (n = 0 to 127) stores the ID, IDE, RTR fields and history configuration of the message to be transmitted from the associated buffer.

TMID[28:0] bits (TX Message Buffer ID Field)

The TMID[28:0] bits are bits of the STD-ID/EXT-ID fields of the message stored in this TX message buffer.

For alignment of these bits in standard and extended frame format, see section 33.2.88. Identifier Bits Alignment.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

THLEN bit (Tx History List Entry)

The THLEN bit controls the storage of an entry corresponding to the transmitted message in the TX History list at the end of a successful transmission.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

TMRTR bit (TX Message Buffer RTR bit)

The TMRTR bit selects whether a data frame or remote frame is to be transmitted from this TX message buffer.

Note: There are no remote frames in CAN-FD format. For a CAN-FD transmission (CFDTMFDCTR.CFFDF = 1), this bit is always transmitted dominant (data frame).

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

TMIDE bit (TX Message Buffer IDE bit)

The TMIDE bit selects whether a message with EXT-ID or STD-ID is to be transmitted from this TX message buffer.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

33.2.89.15 CFDTMPTRn : TX Message Buffer Pointer Register n (n = 0 to 127)

Base address: CANFD = 0x8002_0000

Offset address: 0x1_0004 + 0x080 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TMDLC[3:0]				—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
27:0	—	The read values are undefined. The write value should be 0.	R/W
31:28	TMDLC[3:0]	TX Message Buffer DLC Field Number of data bytes to be transmitted in a CAN frame.	R/W

The CFDTMPTRn register (n = 0 to 127) stores the DLC fields of the message to transmit from the associated buffer.

TMDLC[3:0] bits (TX Message Buffer DLC Field)

The TMDLC[3:0] bits select the number of data bytes to be transmitted from this TX message buffer when the corresponding TMRTR bit is configured as 0.

See Table 5 in ISO 11898-1 (2015) Specification for details in defining the number of data bytes to be transmitted.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

33.2.89.16 CFDTMFDCTRn : TX Message Buffer CAN-FD Control Register n (n = 0 to 127)

Base address: CANFD = 0x8002_0000

Offset address: 0x1_0008 + 0x080 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	TMPTR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TMIFL[1:0]		—	—	—	—	—	TMFDF	TMBRS	TMESI
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TMESI*1	Error State Indicator bit 0: CAN-FD frame to transmit by error active node 1: CAN-FD frame to transmit by error passive node	R/W
1	TMBRS*1	Bit Rate Switch bit 0: CAN-FD frame to transmit with no bit rate switch 1: CAN-FD frame to transmit with bit rate switch	R/W
2	TMFDF*1	CAN FD Format bit 0: Non CAN-FD frame to transmit 1: CAN-FD frame to transmit	R/W
7:3	—	The read values are undefined. The write value should be 0.	R/W
9:8	TMIFL[1:0]	TX Message Buffer Information Label Field	R/W
15:10	—	The read values are undefined. The write value should be 0.	R/W
31:16	TMPTR[15:0]	TX Message Buffer Pointer Field	R/W

Note 1. This bit is not available in the classical CAN function.

The CFDTMFDCTRn register (n = 0 to 127) shows the status of the FDF, BRS, and ESI bits, including the pointer fields of the CAN-FD frame to be transmitted.

TMESI bit (Error State Indicator bit)

If the channel is not in error passive, then the TMESI bit equals the write value, otherwise it is a don't care and the bit is transmitted as 1 on the CAN bus, indicating this is an error passive node.

Do not write to the TMESI bit when the related CAN-FD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

TMBRS bit (Bit Rate Switch bit)

Do not write to the TMBRS bit when the related CAN-FD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

TMFDF bit (CAN FD Format bit)

Do not write to the TMFDF bit when the related CAN-FD channel is in CH_SLEEP mode.

Note: This bit is not available in the classical CAN function.

TMIFL[1:0] bits (TX Message Buffer Information Label Field)

The TMIFL[1:0] bits store the information label value to be copied, together with additional message information, in the TX History List after successful transmission of the message.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

TMPTR[15:0] bits (TX Message Buffer Pointer Field)

The TMPTR[15:0] bits store the pointer value to be copied, together with additional message information in the TX History List after successful transmission of the message.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

33.2.89.17 CFDTMDFp_n : TX Message Buffer Data Field p Register n (p = 0 to 15, n = 0 to 127)

Base address: CANFD = 0x8002_0000

Offset address: 0x1_000C + 0x004 × p + 0x080 × n

Bit position:	31							24	23							16	15							8	7							0
Bit field:	TMDB_HH[7:0]								TMDB_HL[7:0]								TMDB_LH[7:0]								TMDB_LL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TMDB_LL[7:0]	TX Message Buffer Data Byte (4 × p)	R/W
15:8	TMDB_LH[7:0]	TX Message Buffer Data Byte (4 × p + 1)	R/W
23:16	TMDB_HL[7:0]	TX Message Buffer Data Byte (4 × p + 2)	R/W
31:24	TMDB_HH[7:0]	TX Message Buffer Data Byte (4 × p + 3)	R/W

The CFDTMDFp_n register (p = 0 to 15, n = 0 to 127) stores data bytes (4 × p) to data bytes (4 × p + 3) of the message to transmit from the associated buffer.

TMDB_LL[7:0] bits (TX Message Buffer Data Byte (4 × p))

Data bytes (4 × p)[7:0] of the message stored in the TX message buffer.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

TMDB_LH[7:0] bits (TX Message Buffer Data Byte (4 × p + 1))

Data bytes (4 × p + 1)[7:0] of the message stored in the TX message buffer.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

TMDB_HL[7:0] bits (TX Message Buffer Data Byte (4 × p + 2))

Data bytes (4 × p + 2)[7:0] of the message stored in the TX message buffer.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

TMDB_HH[7:0] bits (TX Message Buffer Data Byte (4 × p + 3))

Data bytes (4 × p + 3)[7:0] of the message stored in the TX message buffer.

Do not write to these bits when the related CAN-FD channel is in CH_SLEEP mode.

33.3 Operation

33.3.1 Overview

The modes of the CAN-FD module can be classified into 2 groups:

- Global modes
- Channel modes

33.3.2 Global Modes

Global modes are applicable for the complete CAN-FD module. The global modes of the CAN-FD module are:

- Global Sleep
- Global Reset
- Global Halt
- Global Operation

Figure 33.2 shows the possible transitions between the Global modes.

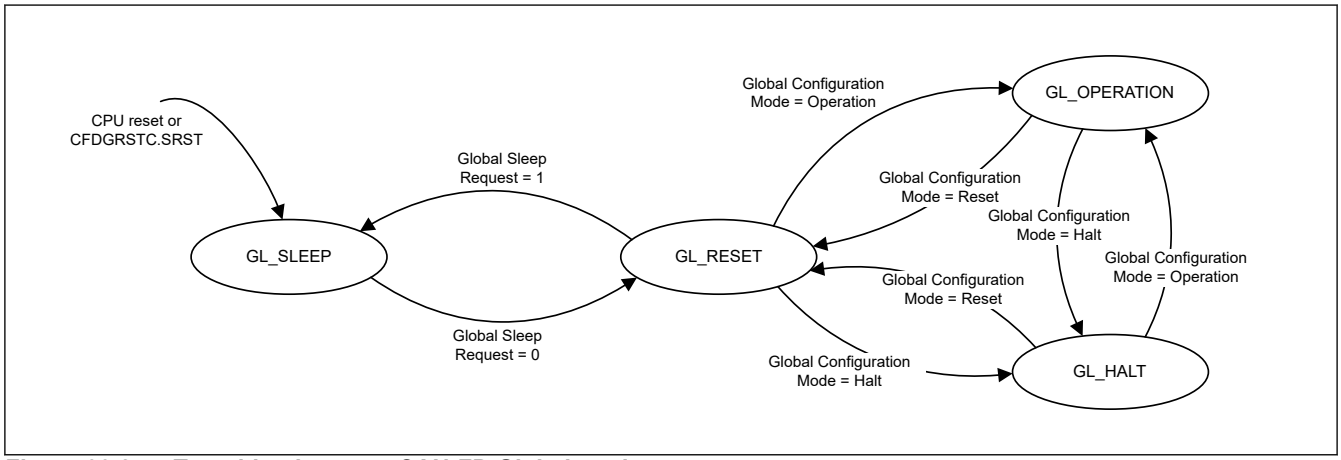


Figure 33.2 Transition between CAN-FD Global modes

Changes in the Global mode can affect the Channel mode. Table 33.14 shows the effect of a Global mode transition on a Channel mode.

Table 33.14 Possible CAN-FD Channel modes and Global modes

Current Global mode	Target Global mode			
	Sleep	**Reset**	**Halt**	**Operation**
Sleep	—	Ch-Sleep: Keep Ch-Reset: N/A Ch-Halt: N/A Ch-Oper: N/A	—	—
Reset	Ch-Sleep: Keep Ch-Reset: → Ch-Sleep Ch-Halt: N/A Ch-Oper: N/A	—	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: N/A Ch-Oper: N/A	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: N/A Ch-Oper: N/A
Halt	—	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: → Ch-Reset Ch-Oper: N/A	—	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: Keep Ch-Oper: N/A
Operation	—	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: → Ch-Reset Ch-Oper: → Ch-Reset	Ch-Sleep: Keep Ch-Reset: Keep Ch-Halt: Keep Ch-Oper: → Ch-Halt	—

33.3.2.1 Global Sleep Mode

After the release of a hardware reset or after setting and clearing the CFDGRSTC.SRST bit, the CAN-FD module automatically enters Global Sleep mode.

The CAN-FD module also enters the Global Sleep mode when the Global Sleep Request bit is set while it is in Global Reset mode. This control bit cannot be set in Global Halt mode or Global Operation mode.

Setting the Global Sleep Request bit sets all Channel Sleep Request bits and forces all channels into the Channel Sleep mode. Sleep mode is used for power saving purpose. When CAN-FD module is in Global Sleep mode, only the clock for CPU write access to the Global Sleep Mode Request bit is active. All other clocks are stopped and all other functions of the CAN-FD module are suspended.

Read access from all registers is still possible and all register values are preserved.

In Global Sleep mode, RAM access is prohibited because the logic which generates a RAM address does not operate.

After setting the Global Sleep Request bit, it is necessary to confirm that the Global Sleep status has been updated, indicating successful transition to Global Sleep mode before the Global Sleep Request bit can be cleared again.

Figure 33.3 shows the procedure for entering Global Sleep mode and Figure 33.4 shows the procedure for exiting Global Sleep mode.

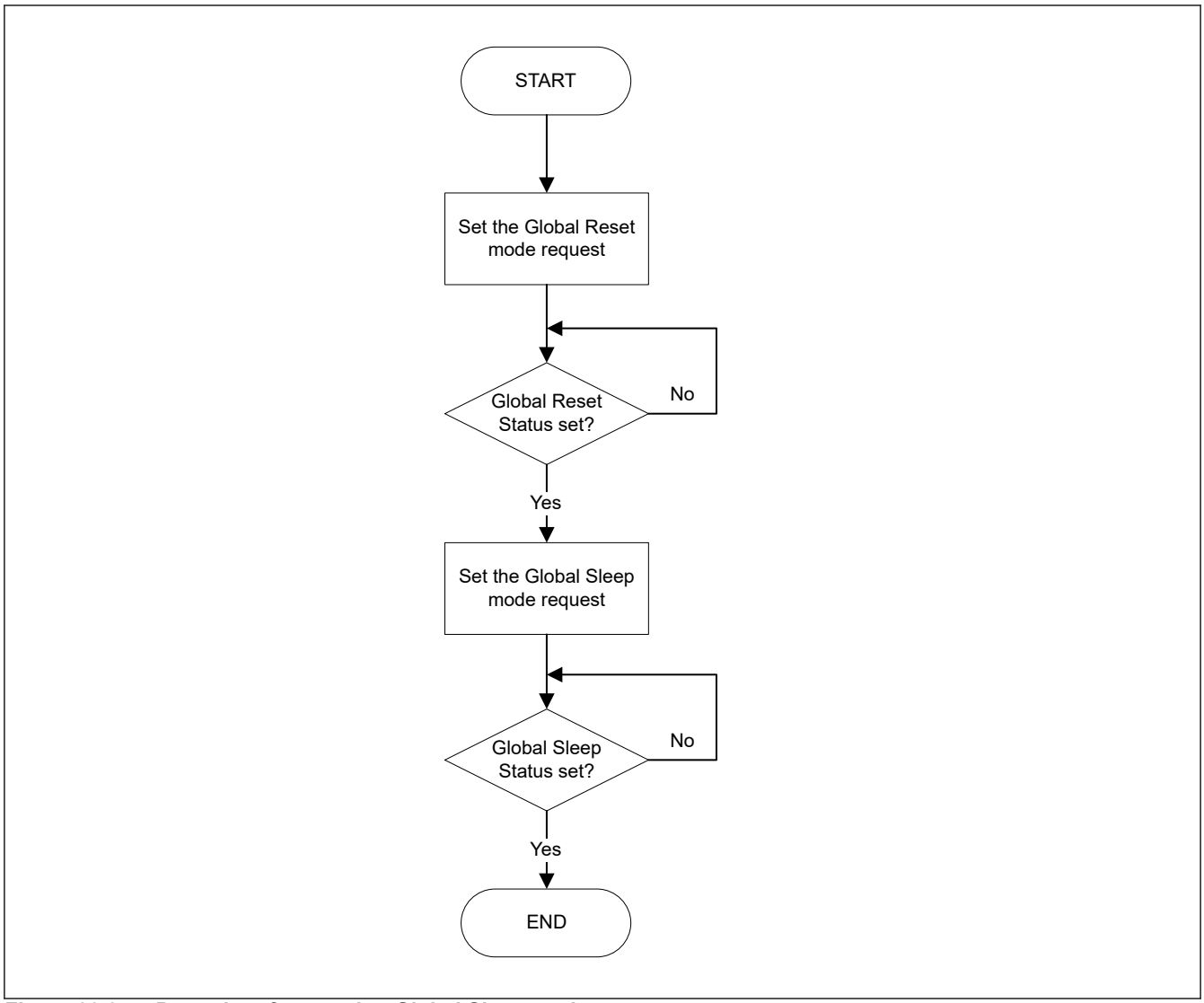


Figure 33.3 Procedure for entering Global Sleep mode

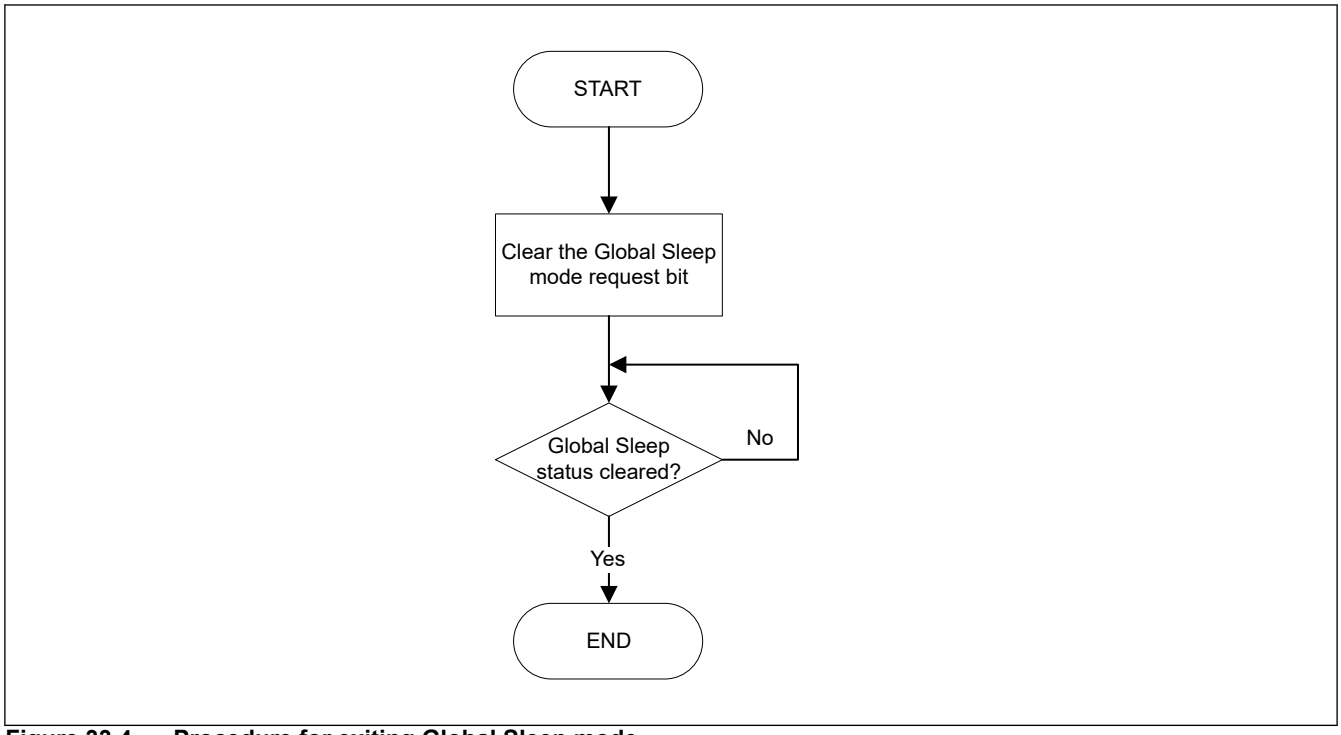


Figure 33.4 Procedure for exiting Global Sleep mode

33.3.2.2 Global Reset Mode

The CAN-FD module enters the Global Reset mode in the following ways:

- Global Mode Control bits CFDGCTR.GMDC[1:0] in the Global Control Register is configured for Global Reset mode while the CAN-FD module is in Global Halt or Global Operation mode.
- Global Sleep Mode Request bit is cleared while CAN-FD module is in Global Sleep mode.

In Global Reset mode, all CAN-FD module functions are suspended and all status and flag registers are initialized.

Additionally all FIFOs and all channel TX Queues are disabled and transmission control bits are cleared.

Configuration registers (except the test mode registers) are not initialized in this mode to their device reset values and the CAN-FD module can be configured.

See section 33.3.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Global Reset mode.

Setting the Global mode to Reset by setting the Global Mode Control bits CFDGCTR.GMDC[1:0] in the Global Control Register to 01b sets all Channel Mode Control bits CFDCnCTR.CHMDC[1:0] in the Channel Control Registers to 01b and forces all channels into the Channel Reset mode.

For channels that are already in Channel Reset mode or Channel Sleep mode, this automatic transition is not performed (CFDCnCTR.CHMDC[1:0] of related channel already set to 01b).

After setting Global Mode Control bits CFDGCTR.GMDC[1:0] to Reset mode, it is necessary to confirm that the Reset Mode Status bit CFDGSTS.GRSTSTS in the Global Status Register has been updated, indicating successful transition to Global Reset mode before CFDGCTR.GMDC can be changed again.

Figure 33.5 shows the procedure for entering Global Reset mode and Figure 33.6 shows the procedure for exiting Global Reset mode.

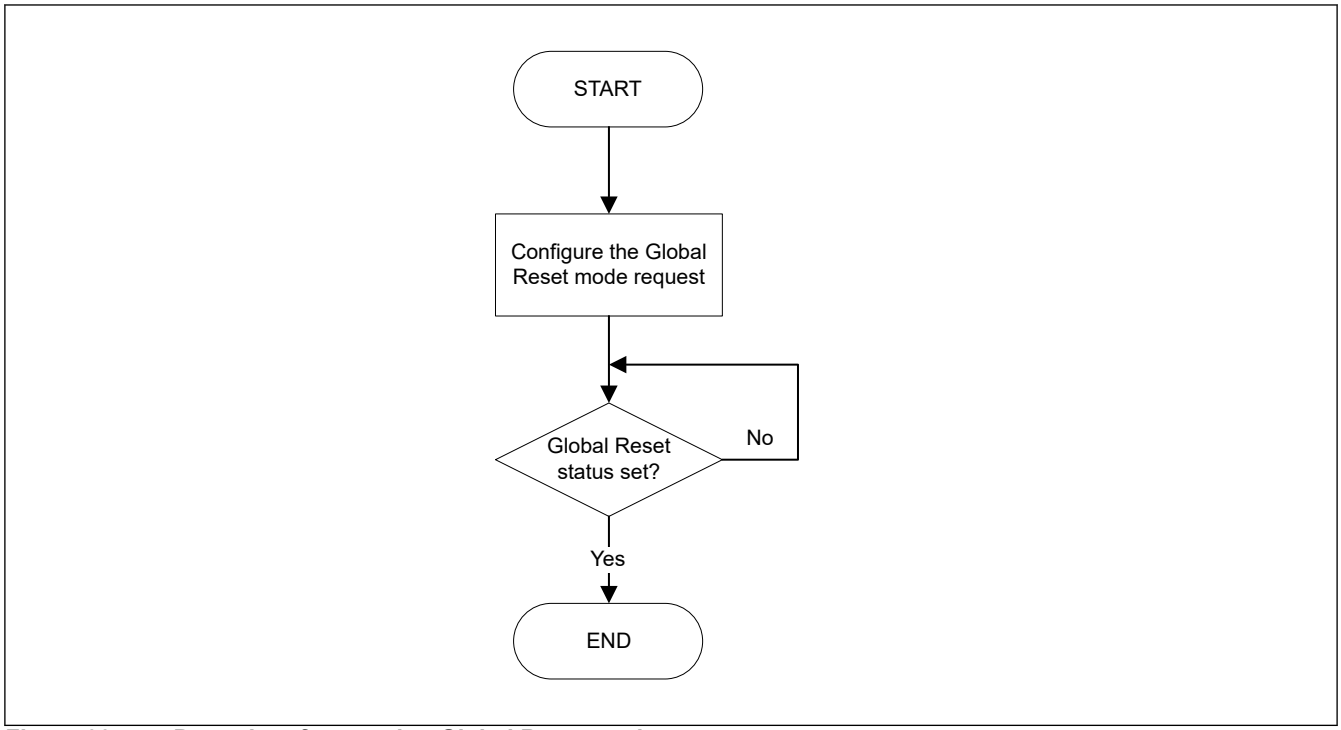


Figure 33.5 Procedure for entering Global Reset mode

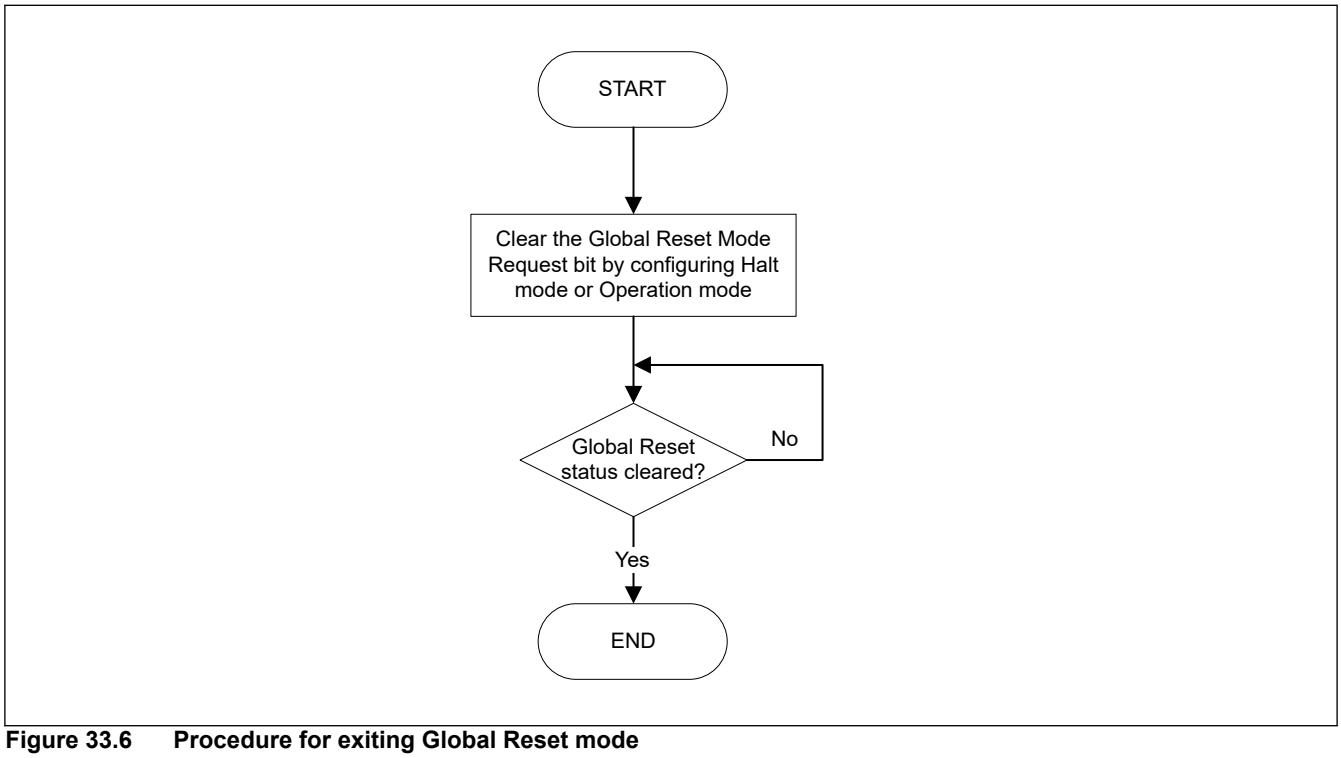


Figure 33.6 Procedure for exiting Global Reset mode

33.3.2.3 Global Halt Mode

The CAN-FD module enters the Global Halt mode in the following ways:

- Global Mode Control bits CFDGCTR.GMDC[1:0] in the Global Control Register is configured for Global Halt mode while the CAN-FD module is in Global Reset mode:
  - The channel is in either Channel Reset or Channel Sleep mode and remains in this mode.
- Global Mode Control bits CFDGCTR.GMDC[1:0] in the Global Control Register is configured for Global Halt mode while the CAN-FD module is in Global Operation mode:

- All channels in Channel Reset, Channel Halt, or Channel Sleep mode remain in this mode
- All channels in Channel Operation mode transit to Channel Halt mode
- Global Halt Mode Status bit is set when all channels have left Channel Operation mode.

If a transmission or reception is ongoing for a channel, the transition to Channel Halt mode is delayed until completion of the communication.

Similarly, if a channel is in bus-off state, the full bus-off recovery sequence may be delayed depending on the channel configuration.

In Global Halt mode, all communications are suspended and CAN-FD logic does not cause any change to the Status and Flag registers (only when a channel is in the bus-off state that its REC and TEC values are cleared). Additionally, the test mode configuration and control registers are not initialized in this mode.

The Global Halt mode should be used to configure global module test modes.

See section 33.3.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Global Halt mode is performed.

Setting the Global mode to Halt mode by setting the Global Mode Control bits CFDGCTR.GMDC[1:0] in the Global Control Register to 10b sets all Channel Mode Control bits CFDCnCTR.CHMDC[1:0] in the Channel Control Registers to 10b for the channels that are in Channel Operation mode and forces these channels into the Channel Halt mode.

For channels that are already in Channel Reset, Channel Halt, or Channel Sleep mode, this automatic transition is not performed.

Therefore, the Global Halt mode request can be used to shut down all CAN-FD channel communications without loss of messages and disruption on the related CAN bus (no interruption of reception/transmission processes on the channels).

After setting the Global Mode Control bits CFDGCTR.GMDC[1:0] to Halt mode, it is necessary to confirm that the Halt Mode Status bit CFDGSTS.GHLTSTS in the Global Status Register has been updated to indicate a successful transition to Global Halt mode. Do not specify any other SFR setting until confirming CFDGSTS.GHLTSTS is set.

Figure 33.7 shows the procedure for entering Global Halt mode and Figure 33.8 shows the procedure for exiting Global Halt mode.

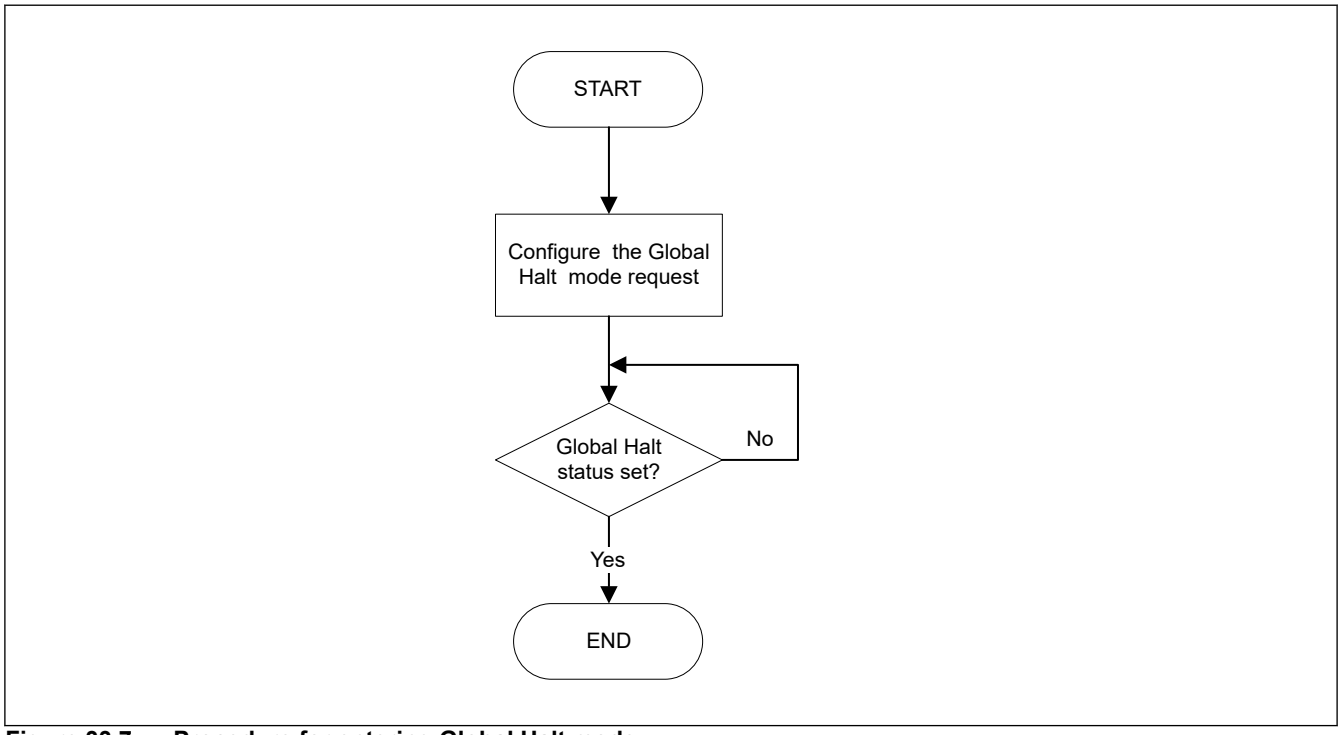


Figure 33.7 Procedure for entering Global Halt mode

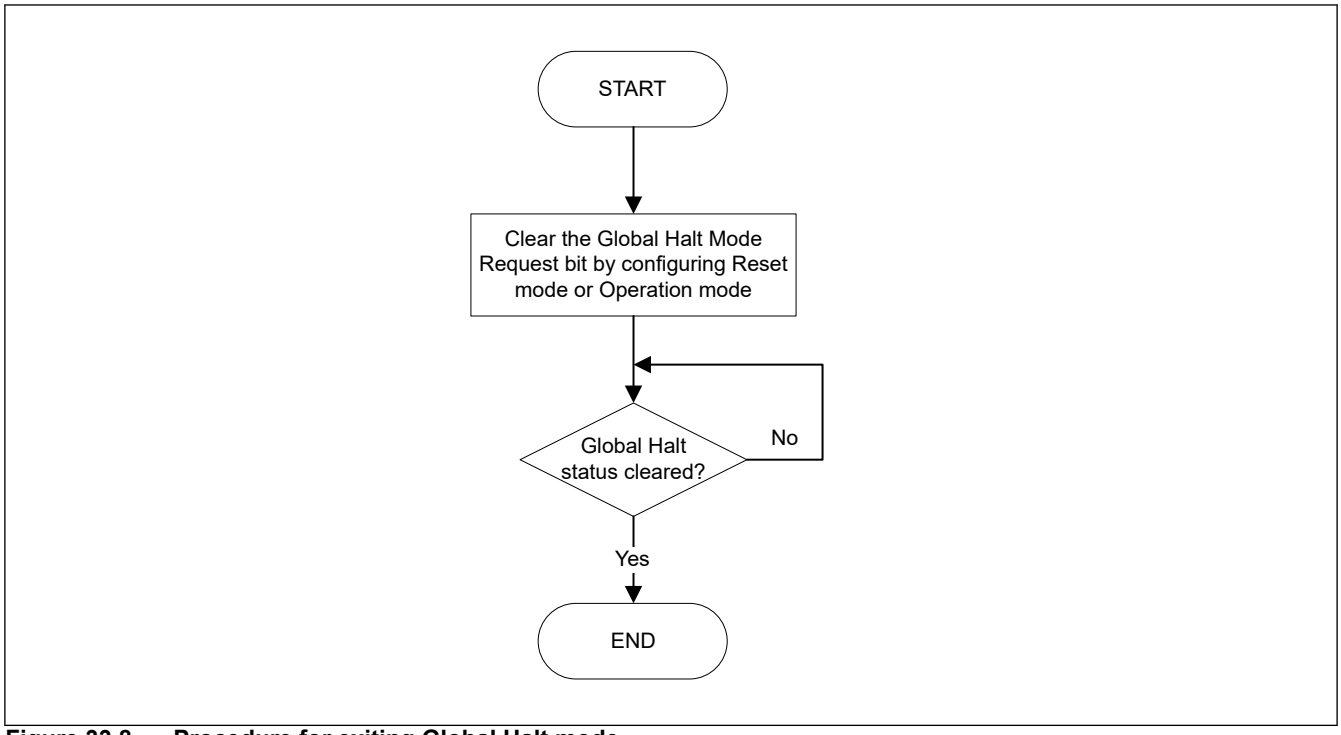


Figure 33.8 Procedure for exiting Global Halt mode

33.3.2.4 Global Operation Mode

The CAN-FD module enters the Global Operation mode when the Global Mode Configuration bits are set to Global Operation mode.

The CAN-FD channels can only be set to Channel Operation mode and start CAN communication when CAN-FD is in Global Operation mode.

After setting the Global Mode Control bits CFDGCTR.GMDC[1:0] to Global Operation mode, it is necessary to confirm that the Global Reset Mode Status bit CFDGSTS.GRSTSTS and the Global Halt Mode Status bit CFDGSTS.GHLTSTS in the Global Status Register have been cleared to indicate a successful transition to Global Operation mode before CFDGCTR.GMDC can be modified again.

Figure 33.9 shows the procedure for entering Global Operation mode and Figure 33.10 shows the procedure for exiting Global Operation mode.

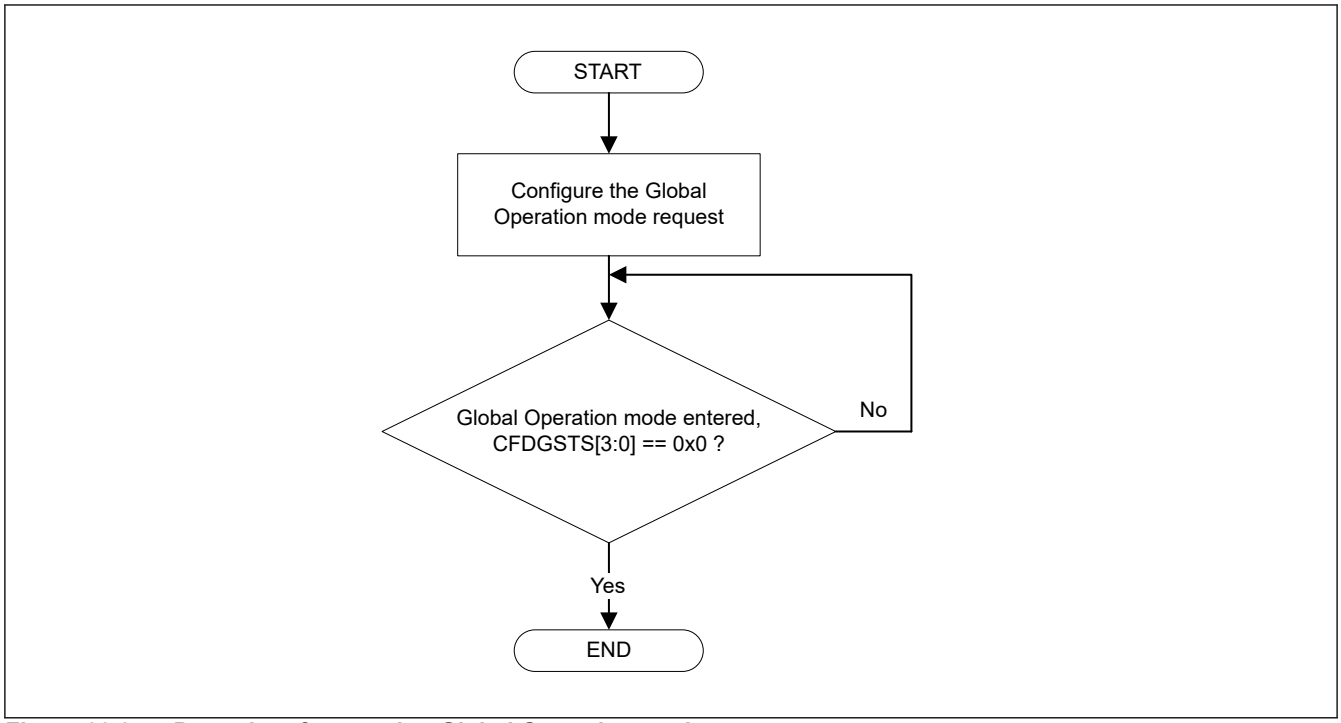


Figure 33.9 Procedure for entering Global Operation mode

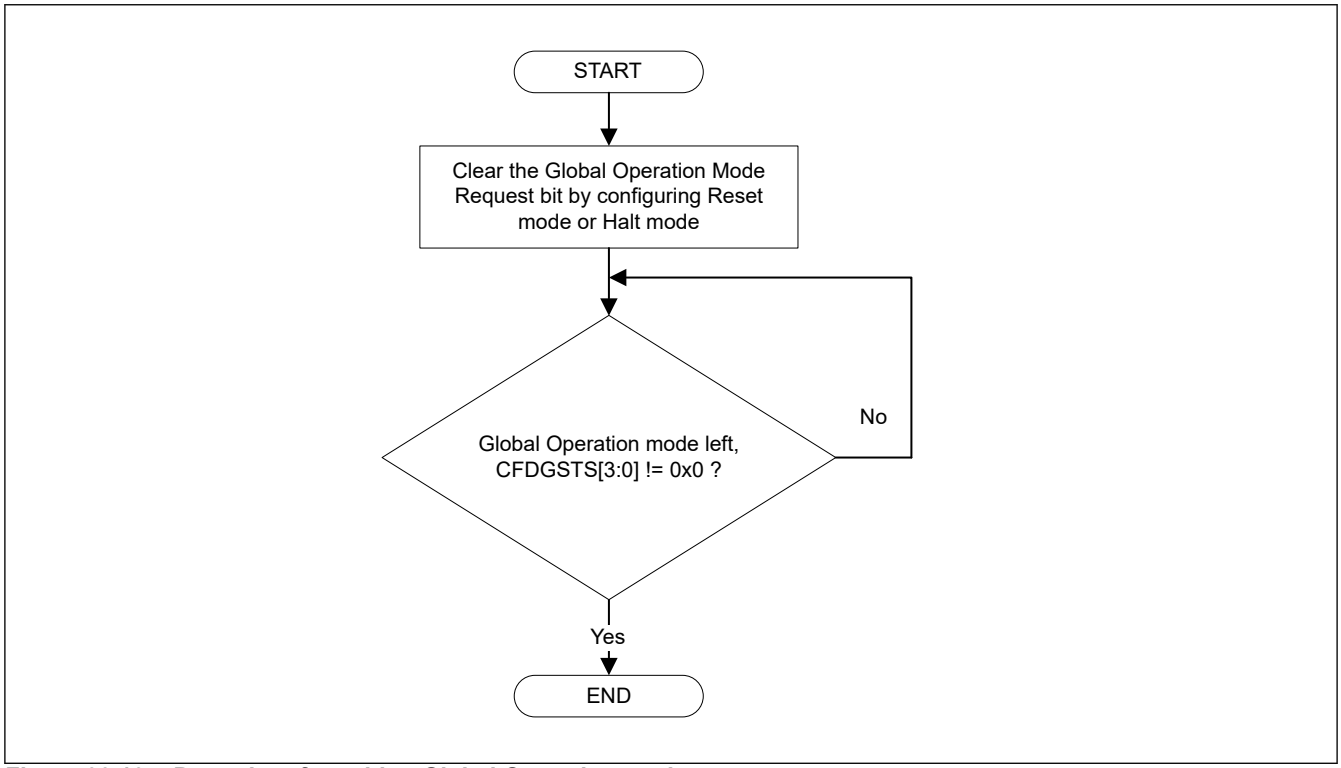


Figure 33.10 Procedure for exiting Global Operation mode

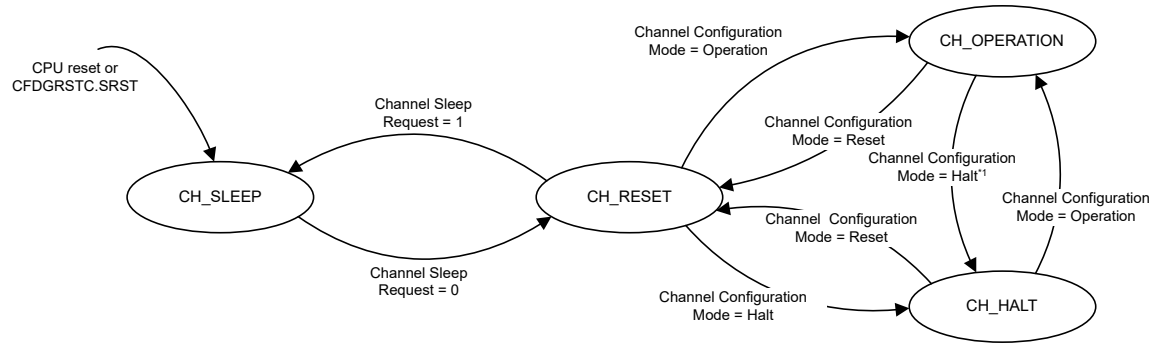
33.3.3 Channel Modes

Each CAN channel can be in one of the following four channel modes:

- Reset
- Halt
- Operation

- Sleep

Figure 33.11 shows possible transitions between the channel modes.



Note 1. The transition timing depends on BOM[1:0] setting in the Channel Control Register (CFDCnCTR):
When BOM[1:0] = 01b, the state transition timing is immediate after entering bus-off state
When BOM[1:0] = 10b, the state transition timing is at the end of bus-off state
When BOM[1:0] = 11b, the state transition timing is matching with the setting of CHMDC to HALT mode

Figure 33.11 Transition between CAN channel modes

33.3.3.1 CAN Channel Sleep Mode

After the release of a hardware reset or after setting and clearing the CFDGRSTC.SRST bit, each CAN channel of the CAN-FD module automatically enters Channel Sleep mode.

Each CAN channel also enters Channel Sleep mode when the related Channel Sleep Mode Request bit is set while the CAN channel is in Channel Reset mode. Do not set this control bit in Channel Halt mode or Channel Operation mode.

Entering the CAN Channel Sleep mode instantly stops the clock supplied to the CAN channel unit and therefore reduces power consumption.

After setting the Channel Sleep Mode Request bit, it is necessary to confirm that the Channel Sleep mode status has been updated to indicate a successful transition to Channel Sleep mode before the Channel Sleep Mode Request bit can be cleared again.

During Channel Sleep mode, do not write to channel related registers. Read operation is still possible.

33.3.3.2 CAN Channel Reset Mode

A CAN-FD CAN channel enters the CAN Channel Reset mode in the following ways:

- Channel Mode Control bits CFDCnCTR.CHMDC[1:0] in the Channel Control Registers is configured for Channel Reset mode while the related CAN channel is in Channel Halt mode or Channel Operation mode
- Channel Sleep Mode Request bit is cleared while the related CAN channel is in Channel Sleep mode
- Global Mode Control bits CFDGCTR.GMDC[1:0] is set to Global Reset mode and CAN channel is not in Channel Sleep mode or Channel Reset mode.

In Channel Reset mode, all CAN channel status and flag registers are initialized.

Additionally all channel-related transmission control bits are cleared and the channel-related TX Queue is disabled.

Configuration registers (except the Channel Test Mode registers) are not initialized in this mode and the CAN channel can be configured for communication.

See section 33.3.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Channel Reset mode is performed.

After setting the Channel Mode Control bits CFDCnCTR.CHMDC[1:0] to Channel Reset mode, it is necessary to confirm that the Reset Mode Status bit CFDCnSTS.CRSTSTS in the related Channel Status Registers has been updated to indicate a successful transition to Channel Reset mode before the related CFDCnCTR.CHMDC[1:0] bits can be modified again.

See Table 33.15 for the behavior of transitioning to Channel Reset mode while CAN communication is ongoing.

Table 33.15 Transition behavior in CAN Reset mode and Halt mode

Mode	State		
	Receiver	Transmitter	Bus-Off
CAN Channel Reset mode (CFDCnCTR.CHMDC[1:0] = 01b)	The CAN channel transits to Channel Reset mode without waiting for the completion of the ongoing reception.*1	The CAN channel transits to Channel Reset mode without waiting for the completion of the ongoing transmission.*1	The CAN channel transits to Channel Reset mode without waiting for the completion of the bus-off recovery.
CAN Channel Halt Mode (CFDCnCTR.CHMDC[1:0] = 10b)	The CAN channel transits to Channel Halt mode at the end of the ongoing reception or error.*2	The CAN channel transits to Channel Halt mode after completion of the ongoing transmission.	When CFDCnCTR.BOM[1:0] is set to 00b, a Channel Halt mode request is accepted only after the completion of the full bus-off recovery sequence. When CFDCnCTR.BOM[1:0] is set to 10b, the CAN channel transits automatically to Channel Halt mode after waiting for the completion of the bus-off recovery. When CFDCnCTR.BOM[1:0] is set to 01b, the CAN channel transits automatically to Channel Halt mode without waiting for the completion of the bus-off recovery. When CFDCnCTR.BOM[1:0] is set to 11b, the CAN channel transits to Channel Halt mode as soon as Channel Halt mode is requested (without waiting for the completion of the bus-off recovery).

Note 1. If the entry to Channel Reset is required only at the end of an ongoing communication, then Channel Halt mode can be requested first to prevent interruption of CAN communication by direct transition to Channel Reset mode. After the CAN channel enters Channel Halt mode, the Channel Reset mode can be requested.

Note 2. If CAN communication is locked at a dominant level after an error flag, software can detect this situation by monitoring the channel related Bus Lock flag and resolve the lock condition by setting the CAN channel to Channel Reset mode.

33.3.3.3 CAN Channel Halt Mode

A CAN-FD CAN channel enters the CAN Channel Halt mode in the following ways:

- Channel Mode Control bits CFDCnCTR.CHMDC[1:0] in the Channel Control Registers are configured for Channel Halt mode while the related CAN channel is in Channel Reset mode or Channel Operation mode
- Global Mode Control bits CFDGCTR.GMDC[1:0] are set to Global Halt mode and CAN channel is in Channel Operation mode.

In Channel Halt mode, all channel CAN communication is suspended but all status and flag registers remain unchanged during Channel Halt mode entry (except for the bus-off case where REC and TEC values are cleared for the channel).

In addition, the Channel Test Mode Configuration and Control registers are not initialized in this mode.

The Channel Halt mode should be used to configure Channel Test modes.

See section 33.3.4. Global Mode and Channel Mode Transition Interactions for a detailed description of the behavior of all registers when transition to Channel Halt mode is performed.

After setting the Channel Mode Control bits CFDCnCTR.CHMDC[1:0] to Channel Halt mode, it is necessary to confirm that the Halt Mode Status bit CFDCnSTS.CHLTSTS in the related Channel Status Register is updated to indicate a successful transition to Channel Halt mode before the related CFDCnCTR.CHMDC[1:0] bits can be modified again.

See Table 33.15 for the behavior of transitioning to Channel Halt mode while CAN communication is ongoing.

33.3.3.4 CAN Channel Operation Mode

The Channel Operation mode is activated by setting the CFDCnCTR.CHMDC[1:0] bits to 00b. If 11 consecutive recessive bits are detected after entering CAN Channel Operation mode, the CFDCnSTS.COMSTS bit is set and the CAN channel:

- Enables the functions of the channel communication by allowing the channel to become an active node on the CAN network
- Releases the internal fault confinement logic including receive and transmit error counters

At this point, the CAN channel can start transmission and reception of CAN messages.

Within the CAN Channel Operation mode, the channel may be in four different sub-modes, depending on which type of communication functions are performed (see Figure 33.12):

- Channel idle: The CAN channel is neither receiving nor transmitting
- Channel receives: The channel is receiving a CAN message sent by another CAN node
- Channel transmits: The channel is transmitting a CAN message

  Note: The channel may receive its own message simultaneously when Self Test mode is enabled.
- Channel is in bus-off state: The CAN channel is cut-off from CAN bus communication.

After setting the Channel Mode Control bits CFDCnCTR.CHMDC[1:0] to Channel Operation mode, it is necessary to confirm that the Channel Reset Mode Status bit CFDCnSTS.CRSTSTS and the Channel Halt Mode Status bit CFDCnSTS.CHLTSTS in the Channel Status Register have been updated to indicate a successful transition to Channel Operation mode before the related CFDCnCTR.CHMDC[1:0] bits can be changed again.

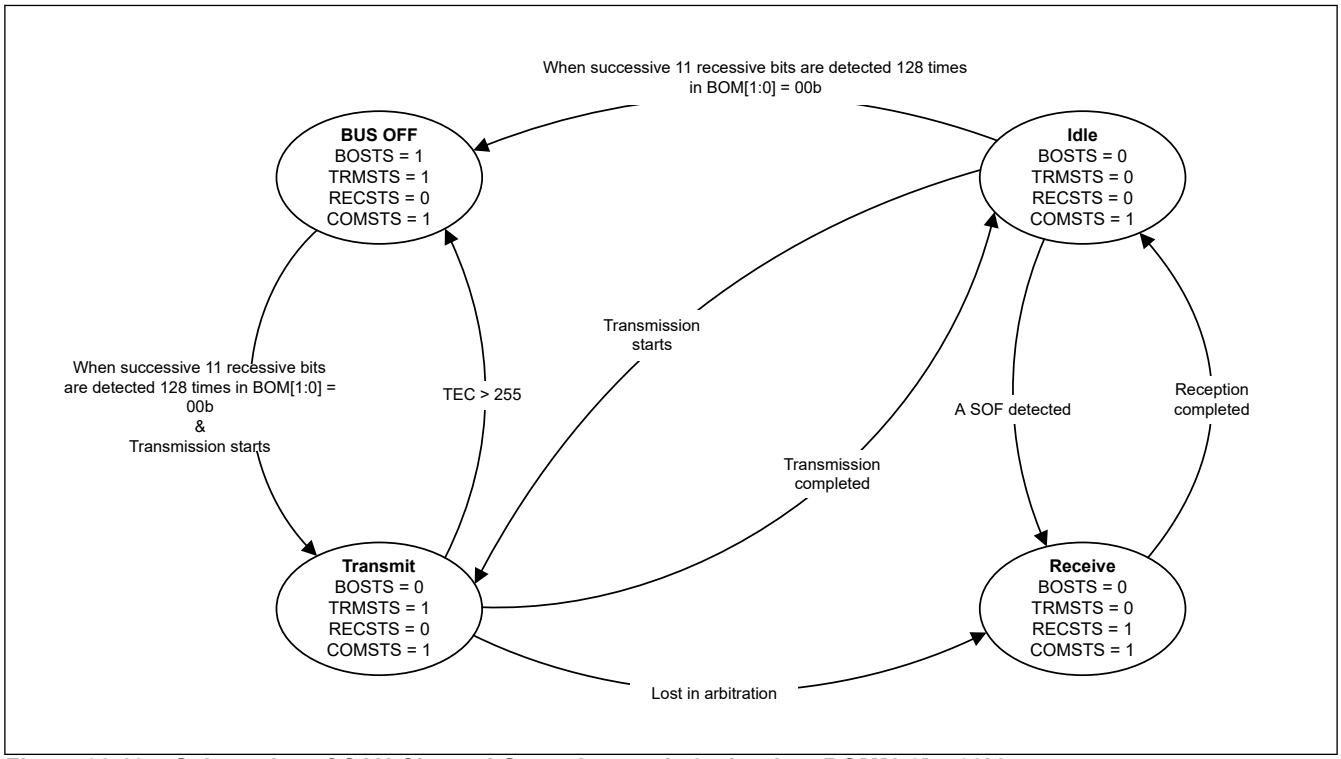


Figure 33.12 Sub-modes of CAN Channel Operation mode (only when BOM[1:0] = 00b)

33.3.3.5 CAN Channel Bus-Off State

The CAN channel bus-off state is entered according to the fault confinement rules of the CAN specification. The following modes can be configured for returning to the CAN Channel Operation mode from the bus-off state:

- CFDCnCTR.BOM[1:0] = 00b:

Bus-Off recovery is compliant to ISO 11898-1, namely the CAN channel re-enters CAN communication (error active state) after 11 consecutive recessive bits are detected 128 times. TEC and REC counters are initialized to 0. The Bus-Off Recovery Flag CFDCnERFL.BORF is set in this case.

- CFDCnCTR.BOM[1:0] = 01b:
  The CAN channel changes the value of the CFDCnCTR.CHMDC[1:0] bits within the CAN Channel Control Register to 10b and switches immediately to Channel Halt mode automatically after entering bus-off state. TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDCnERFL.BORF is not set in this case.
- CFDCnCTR.BOM[1:0] = 10b:
  The CAN channel changes the value of the CFDCnCTR.CHMDC[1:0] bits within the CAN Channel Control Register to 10b as soon as it reaches bus-off state and enters Channel Halt mode automatically after the CAN channel has completed the bus-off recovery sequence (after 11 consecutive recessive bits are detected 128 times). TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDCnERFL.BORF is set in this case.
- CFDCnCTR.BOM[1:0] = 11b:
  Bus-off recovery is initiated but CAN channel can immediately enter Channel Halt mode when still in bus-off state if a request is made to enter Channel Halt mode.
  TEC and REC counters are initialized to 0 and the Bus-Off Recovery Flag CFDCnERFL.BORF is not set.
  Without setting CFDCnCTR.CHMDC[1:0] = 10b and when 11 recessive bits is detected 128 times continuously, transition conditions become the same as CFDCnCTR.BOM[1:0] = 00b.

Note: If the recovery from bus-off occurs normally in Channel Halt mode (after waiting for 128 sequences of 11 consecutive recessive bits), and no halt request has been generated during this period, then the Bus-Off Recovery flag CFDCnERFL.BORF is set.

When software writes to the CFDCnCTR.CHMDC[1:0] bit at the same time as the CAN channel enters Halt mode (at the start of bus-off when CFDCnCTR.BOM[1:0] = 01b, or at the end of bus-off when CFDCnCTR.BOM[1:0] = 10b), the software request has the highest priority.

Note: In the above case, the automatic setting of the CFDCnCTR.CHMDC[1:0] bits to Channel Halt mode request is performed when the CFDCnCTR.CHMDC[1:0] bits value is previously 00b (Channel Operation mode).

Additionally, it is possible to force the CAN channel to recover from the bus-off state by setting CFDCnCTR.RTBO to 1. The error state changes from bus-off state to integrating state with a maximum delay of 1 CAN bit time, and the CAN communication becomes possible again after 11 consecutive recessive bits are detected. The Bus-Off Recovery Flag is not set in this case, and the TEC and REC counters are initialized to 0.

Before setting CFDCnCTR.RTBO to 1, all pending transmissions from the TX message buffers, TX Queues and/or Common FIFO in TX or GW mode should be disabled.

The disable of the pending transmission message buffer, TX Queue or FIFO must be confirmed by the corresponding acknowledge flags.

For the TX message buffer, the acknowledge flags are the Transmission Result Flags (CFDTMSTSn.TMTRF[1:0]). For the TX Queue, it is the TX Queue Empty flag (CFDTXQSTSn.TXQEMP). For the FIFO, it is the FIFO Empty flag (CFDCFSTSn.CFEMP).

The CFDCnCTR.RTBO bit should be used for bus-off recovery only when CFDCnCTR.BOM[1:0] is set to 00b.

Setting this bit in any state other than bus-off has no effect and the bit is cleared immediately.

Table 33.16 shows the settings for the Bus-Off Entry flag CFDCnERFL.BOEF and the Bus-Off Recovery flag CFDCnERFL.BORF for the different configurations of CFDCnCTR.BOM[1:0].

Table 33.16 Behavior of Bus-off Entry and Recovery flags (1 of 2)

BOM[1:0]	BOEF bit setting	BORF bit setting
00b	Always (on entry to bus-off)	Always (on exit from bus-off)
00b CFDCnCTR.RTBO set to 1	Always (on entry to bus-off)	Only if normal bus-off recovery occurs before software sets CFDCnCTR.RTBO to 1
01b	Always (on entry to bus-off)	Never
10b	Always (on entry to bus-off)	Always (on exit from bus-off)

Table 33.16 Behavior of Bus-off Entry and Recovery flags (2 of 2)

BOM[1:0]	BOEF bit setting	BORF bit setting
11b	Always (on entry to bus-off)	Only if normal bus-off recovery occurs before software issues a Halt request

For an efficient software procedure, it is not necessary to wait for the bus-off recovery sequence to end.

It is possible to perform a transmission re-initialization during bus-off recovery. To do this, follow the recommended software flow in Figure 33.13.

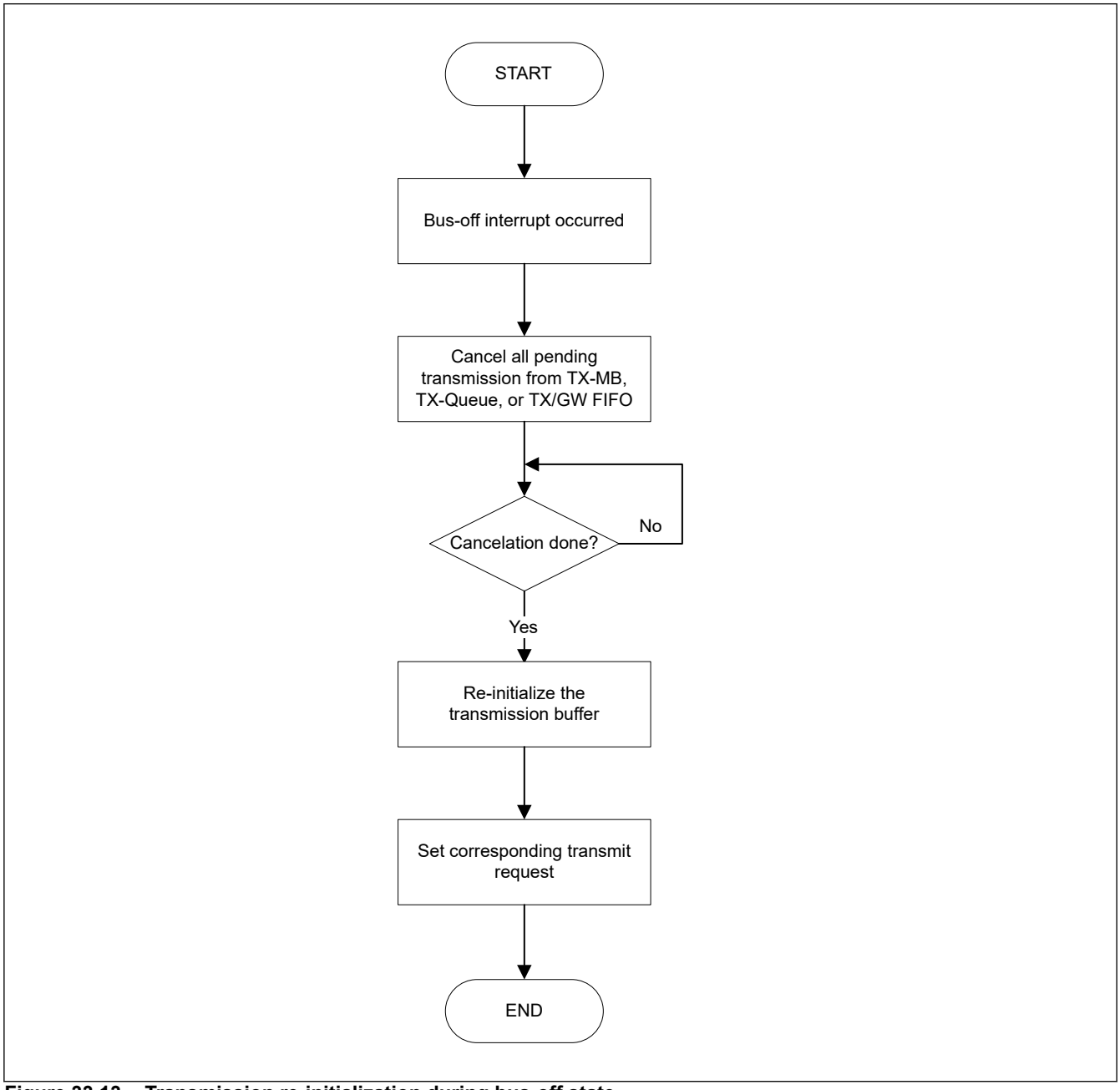


Figure 33.13 Transmission re-initialization during bus-off state

33.3.4 Global Mode and Channel Mode Transition Interactions

The interaction between Global mode setting and Channel mode setting is as follows:

- Changing the Channel Mode Control bits CFDCnCTR.CHMDC[1:0] in the Channel Control Registers does not affect the Global Mode Control bits CFDGCTR.GMDC[1:0].

- Changing the Global Mode Control bits CFDGCTR.GMDC[1:0] affects the channel mode control as described in Table 33.17.

Table 33.17 Instruction between Global and Channel mode transition

Global mode change	Channel mode	Channel mode transition action
Sleep → Reset	Sleep	Channel remains in Sleep mode
Sleep → Halt	— (Global mode change not possible)	
Sleep → Operation	— (Global mode change not possible)	
Reset → Sleep	Sleep	Channel remains in Sleep mode
	Reset	Channel Sleep request bit is set automatically, channel transits to Sleep mode
Reset → Halt	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
Reset → Operation	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
Halt → Sleep	— (Global mode change not possible)	
Halt → Reset	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel mode control is set to Reset mode, channel transits to Reset mode
Halt → Operation	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel remains in Halt mode
Operation → Sleep	— (Global mode change not possible)	
Operation → Reset	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel mode control is set to Reset mode, channel transits to Reset mode
	Operation	Channel mode control is set to Reset mode, channel transits to Reset mode
Operation → Halt	Sleep	Channel remains in Sleep mode
	Reset	Channel remains in Reset mode
	Halt	Channel remains in Halt mode
	Operation	Channel mode control is set to Halt mode, channel transits to Halt mode after communication finished

33.3.4.1 Timing of Global Mode Change

The transition time for the Global mode changes are shown in the following table.

From	To	Maximum transition time
GL_SLEEP	GL_RESET	3 peripheral clock cycles*2
GL_RESET	GL_SLEEP	3 peripheral clock cycles
GL_RESET	GL_HALT	10 peripheral clock cycles
GL_RESET	GL_OPERATION	10 peripheral clock cycles
GL_HALT	GL_RESET	2 CAN bit times
GL_HALT	GL_OPERATION	3 peripheral clock cycles
GL_OPERATION	GL_RESET	2 CAN bit times
GL_OPERATION	GL_HALT	3 CAN frames*1, *3

Note 1. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked RX lines or continued error conditions.

Note 2. Exit GL_SLEEP mode only when CFDGSTS.GRAMINIT is cleared.
Note 3. TQ, CAN frame, and CAN bits are related to the individual channels. For maximum transition time, the channel with the lowest baud rate must be used.

33.3.4.2 Timing of Channel Mode Change

The transition time for the Channel mode changes are shown in the following table.

From	To	Maximum transition time
CH_SLEEP	CH_RESET	3 peripheral clock cycles
CH_RESET	CH_SLEEP	3 peripheral clock cycles
CH_RESET	CH_HALT	3 CAN bit times
CH_RESET	CH_OPERATION	4 CAN bit times
CH_HALT	CH_RESET	2 CAN bit times
CH_HALT	CH_OPERATION	4 CAN bit times*2
CH_OPERATION	CH_RESET	2 CAN bit times
CH_OPERATION	CH_HALT	2 CAN frames*1 *3

Note 1. The time specified for this transition does not include the case where channel enters bus-off state. For bus-off, the timing depends on the configuration of the CFDCnCTR.BOM[1:0] bits.
Note 2. The given transition time is the time without any errors on the bus. In case of an error condition, the transition time can lengthen to an uncalculated result. The transition time can also come to a stuck condition for locked RX lines or continued error conditions.
Note 3. In general, if the baud rate prescaler value CFDCnNCFG.NBRP is changed in CH_HALT mode, the transition time can deviate. The internal prescaler is a free running down counter that creates the TQ clock, and new BRP value is captured when the counter reaches the value 0.

33.4 Initialization

Before joining CAN communications, configure the following settings:

- Clock setting
- Bit timing setting (nominal and data rate)
- Baud rate setting (nominal and data rate)
- CAN-FD setting
- Acceptance filter setting (configuration of Global Acceptance Filter List)
- Reception, Transmission and GW-FIFO setting
- CAN Operation mode setting

33.4.1 Initialization of CAN Clock, Bit Timing and Baud Rate

33.4.1.1 Bit Timing Conditions

The following lines describe the composition of each segment and the limitations that apply to the segment setting.

1. Each segment setting
   SS = Fixed to 1 TQ
   TSEG1 = Refer to (CFDCnNCFG) and (CFDCnDCFG)
   TSEG2 = Refer to (CFDCnNCFG) and (CFDCnDCFG)
   SJW = Refer to (CFDCnNCFG) and (CFDCnDCFG)
   SS + TSEG1 + TSEG2 = 5 to 49 TQs for data bit rate and 8 to 385 for nominal bit rate
2. Limitations on TSEG1, TSEG2 and SJW
   TSEG1 (N) > TSEG2 (N) ≥ SJW (N)
   TSEG1 (D) ≥ TSEG2 (D) ≥ SJW (D)
   When only classical frames are used, configure the bit fields TSEG1 and TSEG2 of CFDCnDCFG to valid values.
   Table 33.18 shows an example of how to set the bit timing to achieve the required Sample Point settings.

Table 33.18 Transition behavior in CAN Reset mode and Halt mode

1 Bit	Set value (TQ)				Sample Point*1 (%)
	SS	TSEG1	TSEG2	SJW	
5 TQ	1	2	2	1	60.00
8 TQ	1	4	3	1	62.50
	1	5	2	1	75.00
10 TQ	1	6	3	1	70.00
	1	7	2	1	80.00
12 TQ	1	8	3	1	75.00
	1	9	2	1	83.33
15 TQ	1	10	4	1	73.33
	1	11	3	1	80.00
16 TQ	1	10	5	1	68.75
	1	11	4	1	75.00
20 TQ	1	12	7	1	65.00
	1	13	6	1	70.00
24 TQ	1	15	8	1	66.66
	1	16	7	1	70.83
50 TQ	1	39	10	4	80.00

Note 1. Sample point (in case of 75%)

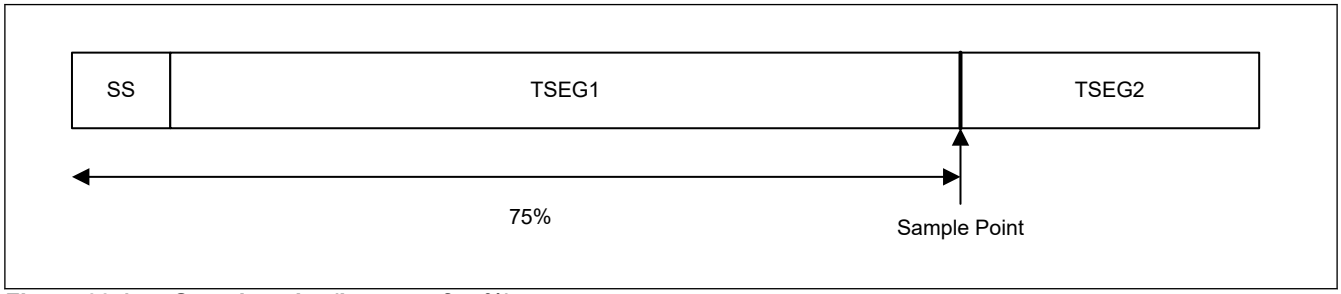


Figure 33.14 Sample point (in case of 75%)

33.4.1.2 CAN Bit Timing

In the CAN protocol, each bit in a communication frame is composed of three segments that can be configured individually for each channel using the related CFDCnNCFG and CFDCnDCFG registers.

Figure 33.15 shows the segment composition of a bit and the sample point in it.

Of these segments, the Time Segment 1 (TSEG1) and Time Segment 2 (TSEG2) are used to specify the position of the sample point, so that the timing at which each bit on the CAN bus is sampled can be altered by changing the values of these segments.

The minimum resolution for this timing is referred to as Time Quantum (TQ), which is determined by the clock frequency supplied to the CAN channel and the divide-by-N value of the baud rate prescaler (nominal and data rate).

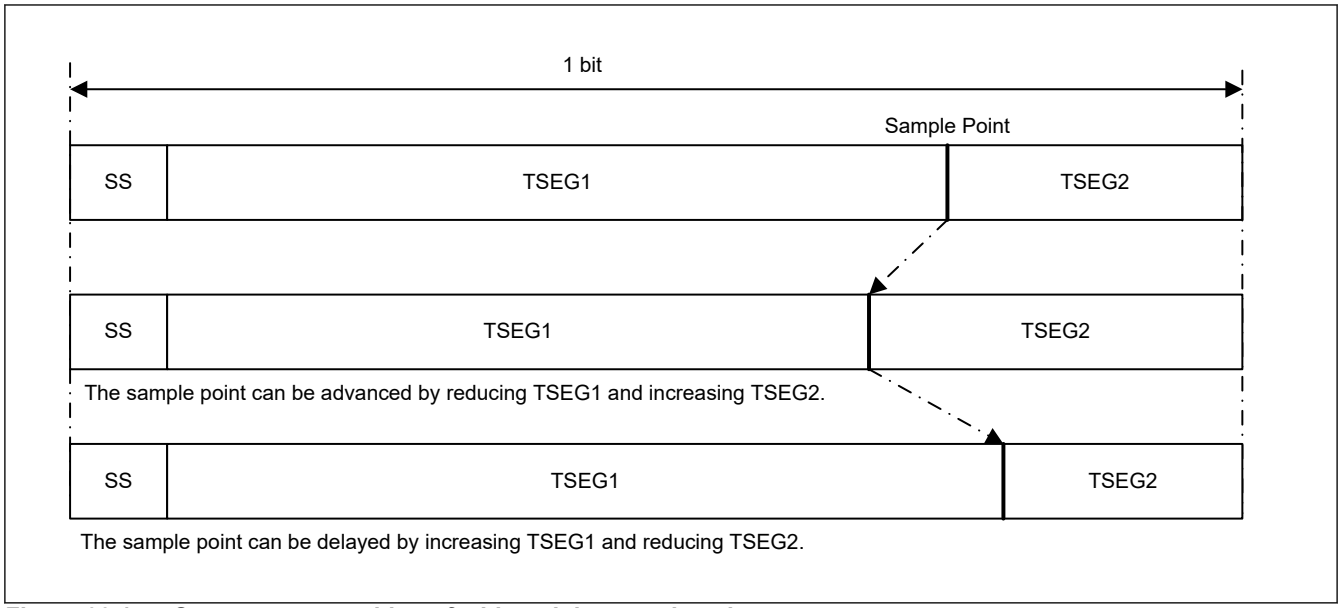


Figure 33.15 Segment composition of a bit and the sample point

1. SS: Synchronization Segment
   This segment is used to synchronize bits by monitoring a recessive-to-dominant edge during the interframe space. This comprises of intermission, suspend transmission, bus idle, during bus idle, and all nodes that can start transmission.
2. TSEG1: Time Segment 1
   This segment absorbs physical delays on the CAN network. A physical delay on the network is two times the total sum of a bus delay, input comparator delay, and output driver delay. It can be lengthened by SJW.
3. TSEG2: Time Segment 2
   This segment is used to correct a phase error by performing resynchronization. It can be shortened by SJW. While sending or receiving a message, communication frames between some nodes may get out of sync due to a drift in the oscillator frequency or a delay in the transmission path. This is referred to as a phase error.
4. SJW: Resynchronization Jump Width
   This is the maximum width by which bits that have become out of sync due to a phase error may be corrected.

Figure 33.15 shows only one symbolic sample point.

33.4.1.3 Baud Rate

Either PCLKM or PCLKCAN can be selected globally for all CAN channels as CAN communication clock.

The transfer speed is determined by the DLL clock, the divide-by-N value of the baud rate prescaler, and the number of TQs in one bit.

$$\text{baud rate} = \frac{\text{DLL_Clock}}{(\text{number_of_time_quanta_per_bit}) \times (\text{BRP} + 1)}$$

Figure 33.16 shows a block diagram of the circuit that generates the CAN channel system clock, Table 33.19 and Table 33.20 show baud rate examples.

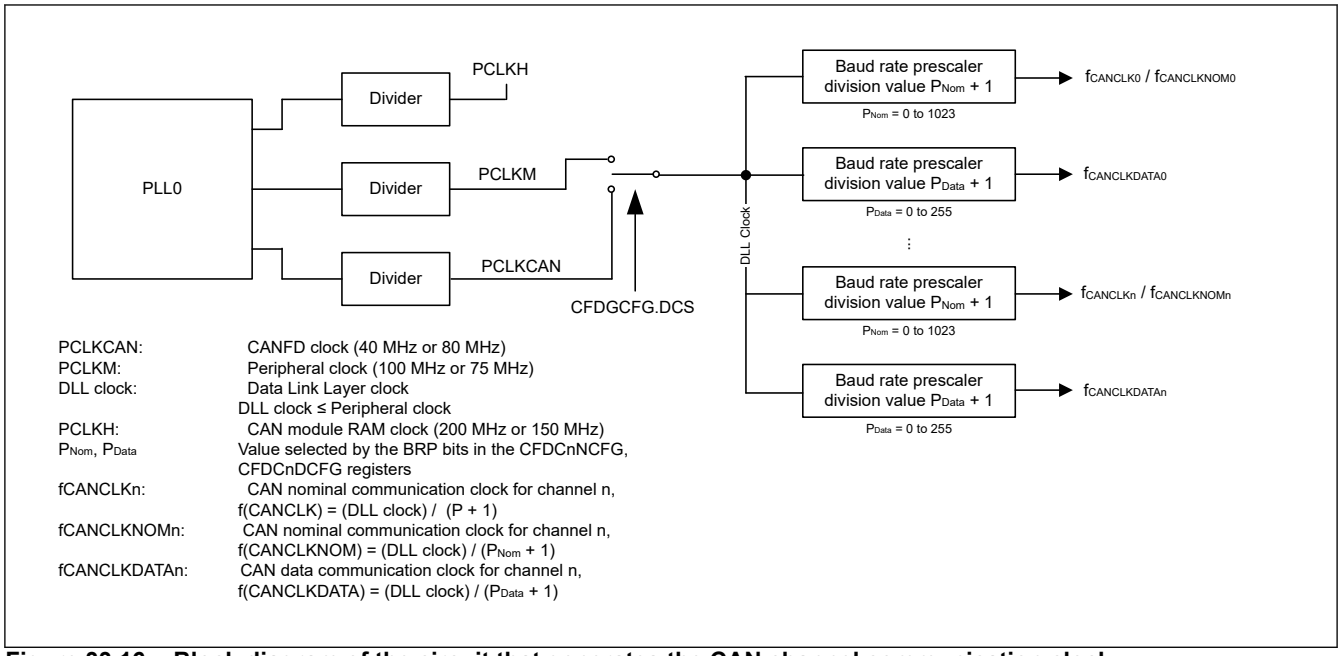


Figure 33.16 Block diagram of the circuit that generates the CAN channel communication clock

Table 33.19 Nominal baud rate calculation formula and example CAN communication configurations

Baud rate calculation formula	(DLL clock) (Baud rate prescaler divide-by-N value[*1]) × (Number of TQs in one bit)			
	80 MHz	**40 MHz**	**100 MHz**	**75 MHz**
1 Mbps	8 TQ (10) 20 TQ (4)	8 TQ (5) 20 TQ (2)	10 TQ (10) 20 TQ (5)	15 TQ (5)
500 Kbps	8 TQ (20) 20 TQ (8)	8 TQ (10) 20 TQ (4)	10 TQ (20) 20 TQ (10)	15 TQ (10)
250 Kbps	8 TQ (40) 20 TQ (16)	8 TQ (20) 20 TQ (8)	10 TQ (40) 20 TQ (20)	15 TQ (20)
125 Kbps	8 TQ (80) 20 TQ (32)	8 TQ (40) 20 TQ (16)	10 TQ (80) 20 TQ (40)	15 TQ (40)
83.3 Kbps	8 TQ (120) 12 TQ (80) 16 TQ (60) 24 TQ (40)	8 TQ (60) 12 TQ (40) 16 TQ (30) 24 TQ (20)	8 TQ (150) 10 TQ (120) 12 TQ (100) 15 TQ (80) 16 TQ (75) 20 TQ (60) 24 TQ (50)	10 TQ (90) 12 TQ (75) 15 TQ (60) 20 TQ (45)
33.3 Kbps	8 TQ (300) 12 TQ (200) 16 TQ (150) 20 TQ (120) 24 TQ (100)	8 TQ (150) 12 TQ (100) 16 TQ (75) 20 TQ (60) 24 TQ (50)	8 TQ (375) 10 TQ (300) 12 TQ (250) 15 TQ (200) 20 TQ (150) 24 TQ (125)	10 TQ (225) 15 TQ (150) 18 TQ (125)

Note: Shown in ( ) are the baud rate prescaler divided-by-N values.
Note 1. Baud rate prescaler divide-by-N value = P + 1 (P = 0 to 1023), P: value selected by the BRP bits in the Channel Configuration Registers.

Table 33.20 Baud rate calculation example for nominal and data bit rate CAN communication configurations

Baud rate calculation formula	(DLL clock) (Baud rate prescaler divide-by-N value*1) × (Number of TQs in one bit)			
	80 MHz	40 MHz	100 MHz	75 MHz
Nominal 1 Mbps Data 8 Mbps	80 TQ (1)	40 TQ (1)	100 TQ (1)	75 TQ (1)
	10 TQ (1)	5 TQ (1)	Not possible	Not possible
Nominal 1 Mbps Data 5 Mbps	80 TQ (1)	40 TQ (1)	100 TQ (1)	75 TQ (1)
	16 TQ (1)	8 TQ (1)	20 TQ (1)	15 TQ (1)
Nominal 500 Kbps Data 2 Mbps	160 TQ (1)	80 TQ (1)	200 TQ (1)	150 TQ (1)
	40 TQ (1)	20 TQ (1)	50 TQ (1)	Not possible

Note: Shown in ( ) are the baud rate prescaler divided-by-N values.
Note 1. Baud rate prescaler divide-by-N value = P + 1 (P = 0 to 1023), P: value selected by the BRP bits in the Channel Configuration Registers.

For optimum clock tolerance in networks using the FD frame format, the length of the time quantum should be the same in nominal bit time and in data bit time. This means CFDCnNCFG.NBRP = CFDCnDCFG.DBRP.

Additionally, if transceiver delay compensation is used, do not program the CFDCnDCFG.DBRP[7:0] bits to be greater than 1, as 1 means divide by 2.

33.4.1.4 Setting of CAN Clock, Bit Timing and Baud Rate

Figure 33.17 shows the procedure for setting the CAN clock and the baud rate for each channel.

These settings should be performed during Channel Reset mode (Configuration mode) for the CAN channels.

Before going to channel communication state, the baud rate must be configured, otherwise the mode does not switch correctly.

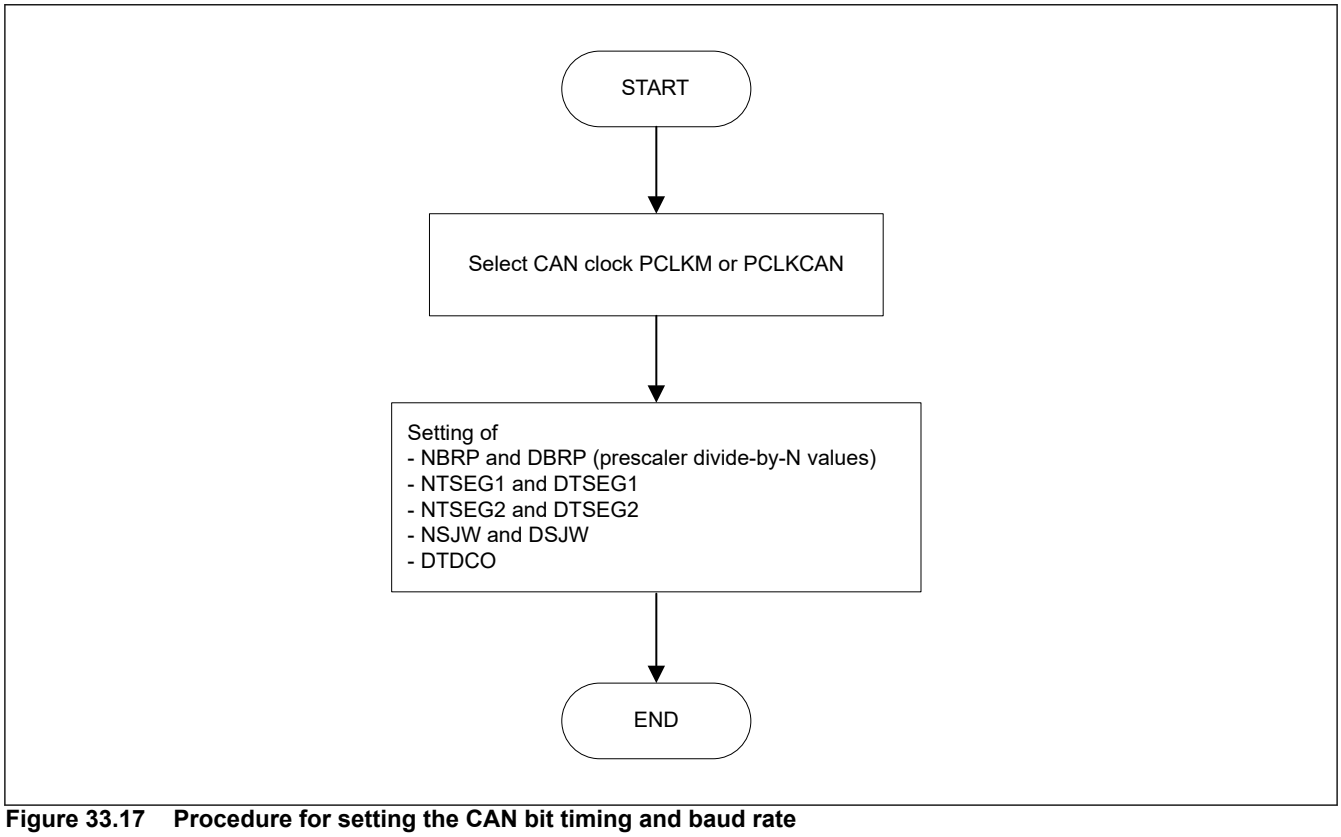


Figure 33.17 Procedure for setting the CAN bit timing and baud rate

33.4.1.5 Transmitter Delay Compensation

When a high baud rate is used such as 5 to 8 Mbps for the data phase, the transmitter delay can become greater than TSEG1. In this case, the transmitter always detects a bit-error in the data phase of the CAN-FD frame. The TDC compensates for the inability of the transmitter to receive its own transmitted bit at the sample point of that bit.

There is another symbolic sample point known as the Secondary Sample Point (SSP) that is used only during the data phase of CAN-FD frames. This is derived from the Transceiver Delay Compensation Result bit (CFDCnFDSTS.TDCR) as shown in Figure 33.18.

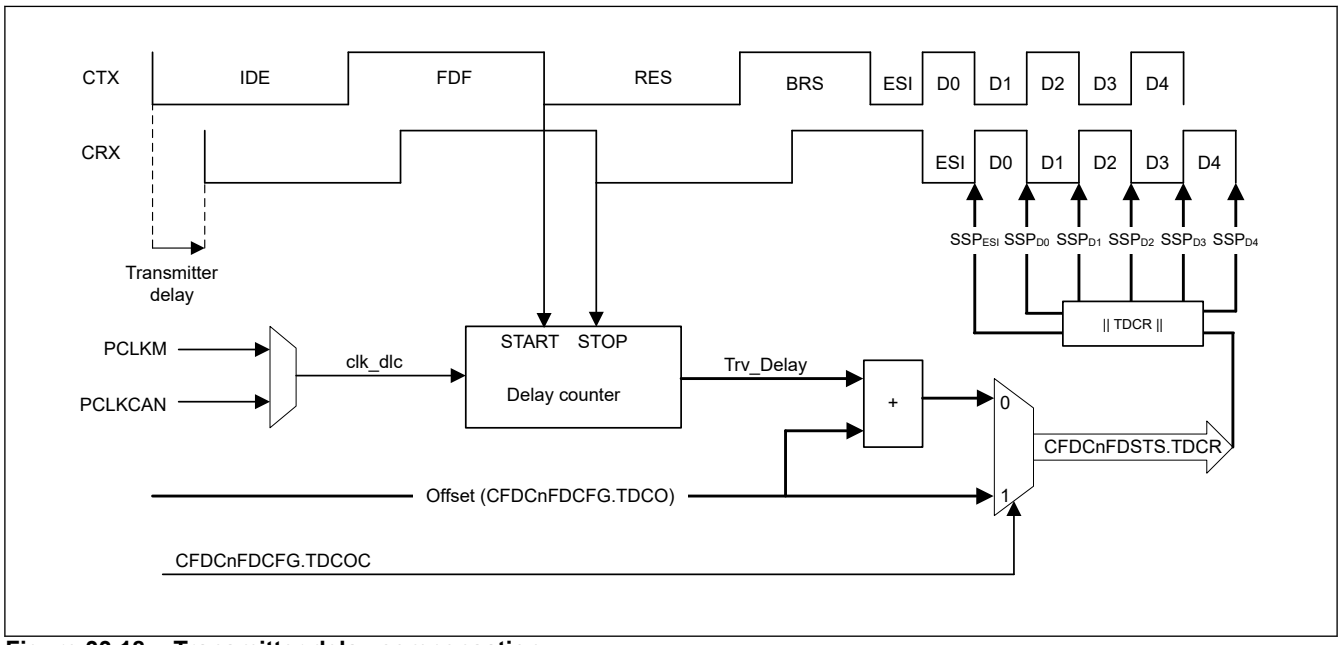


Figure 33.18 Transmitter delay compensation

The measured Trv_Delay is based on the number of clk_dlc clock cycles. The delay is counted up by 1 for each started clock until the dominant value is seen on CAN_RX. Figure 33.19 shows the measured result. Trv_Delay counted to maximum 127 with a clk_dlc clock.

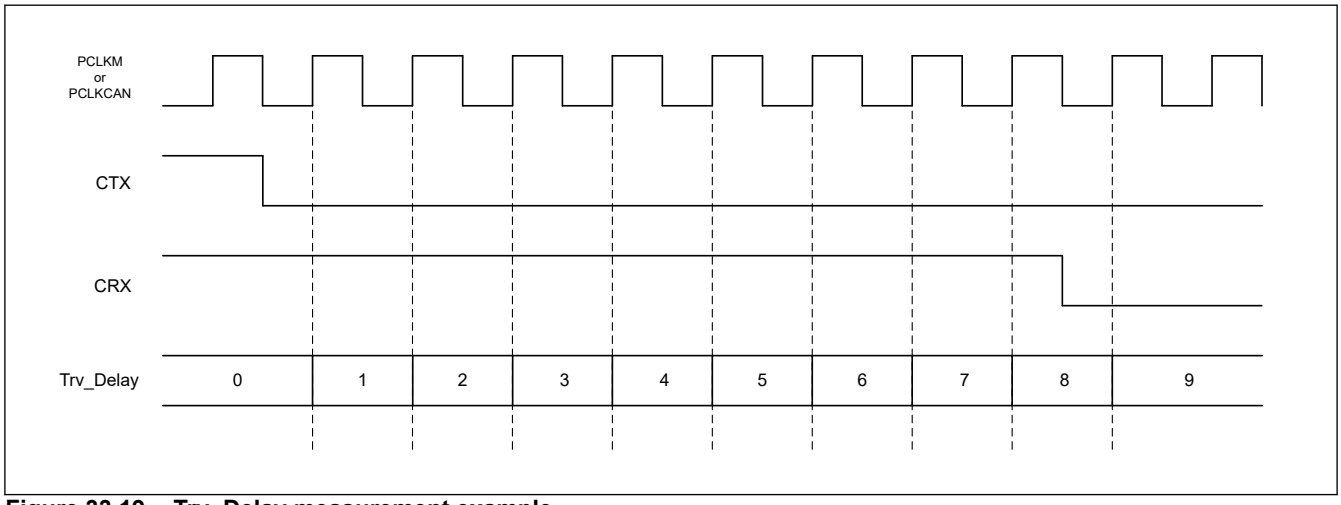


Figure 33.19 Trv_Delay measurement example

The SSP is calculated by taking the result from CFDCnFDSTS.TDCR and rounding the value down to the nearest integer number of data time quanta.

Figure 33.20 shows the positioning of the secondary sample point. When CFDCnFDCFG.TDCOC is equal to 0, the SSP is equal to the Trv_Delay (measured delay) + CFDCnFDCFG.TDCO, rounded down to the nearest integer number of time quanta. Usually, the TDCO value should have the size of (Sync Segment data + TSEG1 data) to position the SSP to a theoretical location of the sample point.

If the CFDCnFDCFG.TDCOC is equal to 1, the SSP is defined by CFDCnFDCFG.TDCO. If CFDCnDCFG.DBRP[7:0] is greater than 0, the value is also rounded down to the nearest integer number of time quanta.

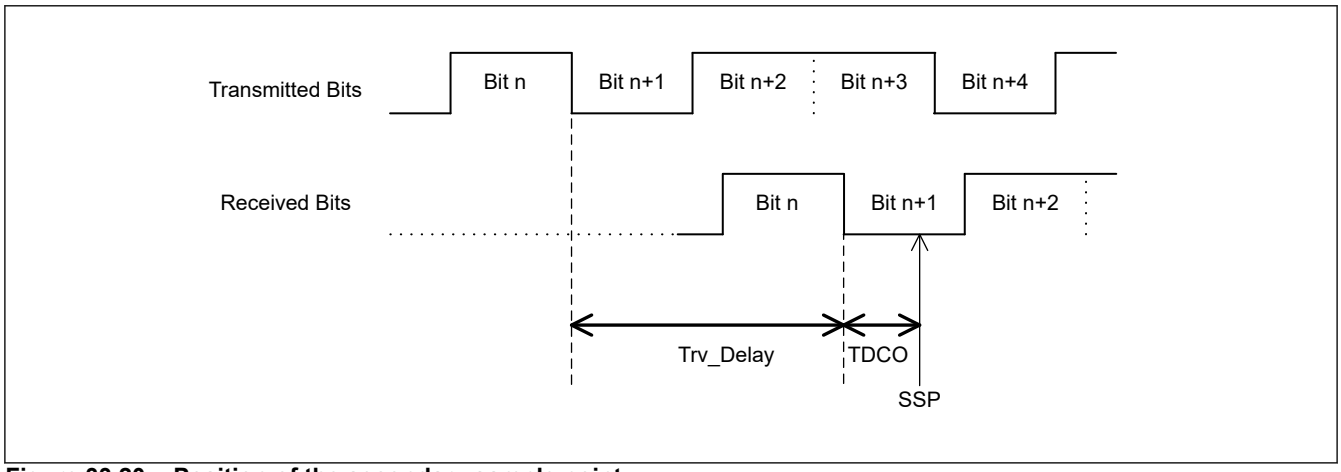


Figure 33.20 Position of the secondary sample point

The maximum delay (Trv_Delay + TDCO) which can be compensated by the CAN-FD module is (6 data bits - 2clk_dlc).

The ISO 11898-1 allows you to set different values for BRP_data and BRP_nom.

If different values are used for CFDCnNCFG.NBRP and CFDCnDCFG.DBRP, then two CAN nodes may be out of synchronization at the point when the bit rate changes from nominal bit rate to data bit rate after sample point of the BRS bit. This condition is shown in Figure 33.21.

The length of the time quantum should be the same in the nominal bit time and in the data bit time. This means CFDCnNCFG.NBRP = CFDCnDCFG.DBRP.

Different bit rates can be achieved by selecting different configuration values for the Time Segments. The nominal bit rate can be configured from 8 to 385 TQs and the data bit rate from 5 to 49 TQs.

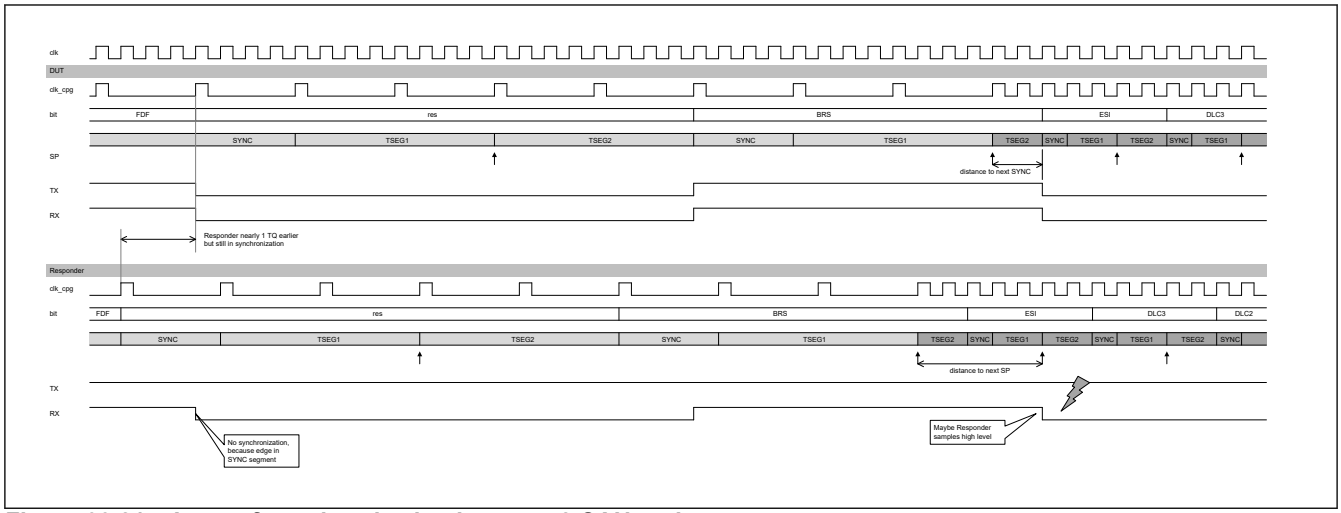


Figure 33.21 Loss of synchronization between 2 CAN nodes

The transmitter delay compensation measurement result is updated at the falling edge from FDF bit to RES bit when configured accordingly (CFDCnFDCFG.TDCE = 1, CFDCnFDCFG.TDCOC = 0).

Figure 33.22 shows the read flow to get the measured transmitter delay compensation result.

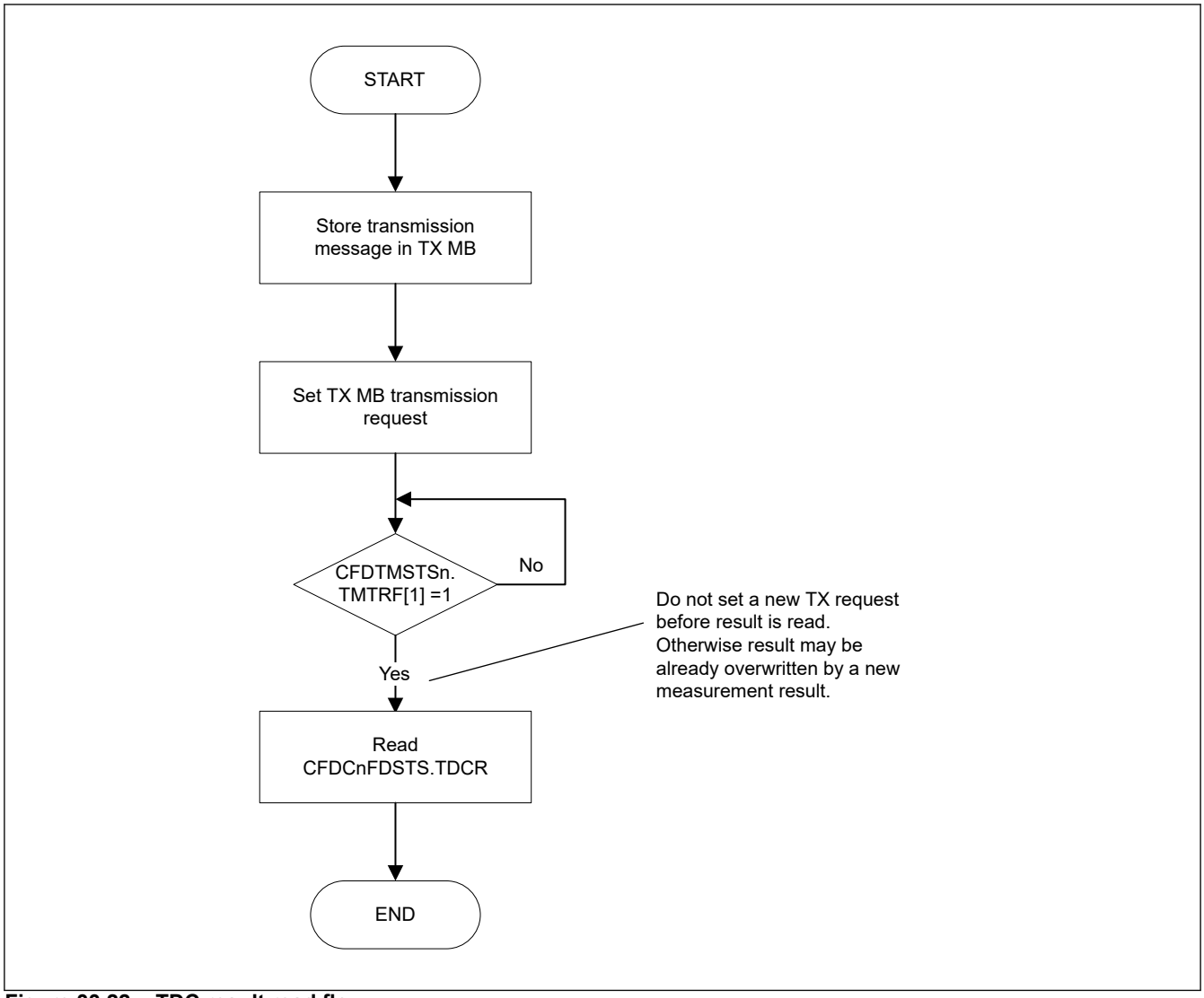


Figure 33.22 TDC result read flow

33.4.2 CAN Module Configuration after Hardware Reset

After a hardware reset (power on reset) or after setting and clearing the CFDGRSTC.SRST bit, the CAN-FD module enters Global Sleep mode automatically.

To enable configuration of the CAN-FD module, you must exit Sleep mode by clearing the Global Sleep Request bit CFDGCTR.GSLPR to 0.

After a hardware reset, the module starts RAM initialization, the CFDGSTS.GRAMINIT bit in the Global Status Register is set automatically to indicate that the CAN-FD logic is initializing the RAM.

After RAM initialization is complete, this bit is cleared automatically.

RAM initialization is necessary to avoid setting of false ECC error flag after hardware resets the random data presented in the RAM.

Do not access registers of the CAN-FD in either read or write until RAM initialization is complete and the CFDGSTS.GRAMINIT bit is cleared.

Before going to communication mode, the Global Acceptance Filter List and message FIFO buffers must be configured. In addition, each required CAN channel must be configured such as CAN bit timing. For this configuration, all required CAN channels must be released from Channel Sleep mode and must be configured for communication in Channel Reset mode (Configuration mode).

For this, all required CAN channels must be released from Channel Sleep mode and must be configured for communication in Channel Reset mode (Configuration mode).

Figure 33.23 shows the configuration procedure. For details about each step, see section 33.5. Acceptance Filtering Function using Global Acceptance Filter List (AFL), section 33.6. FIFO Buffers and Normal Message Buffer Configuration, section 33.7. Interrupt and DMA, and section 33.4.1.3. Baud Rate.

The CAN-FD module does not perform RAM initialization sequence after executing a software reset by setting CFDGRSTC.SRST.

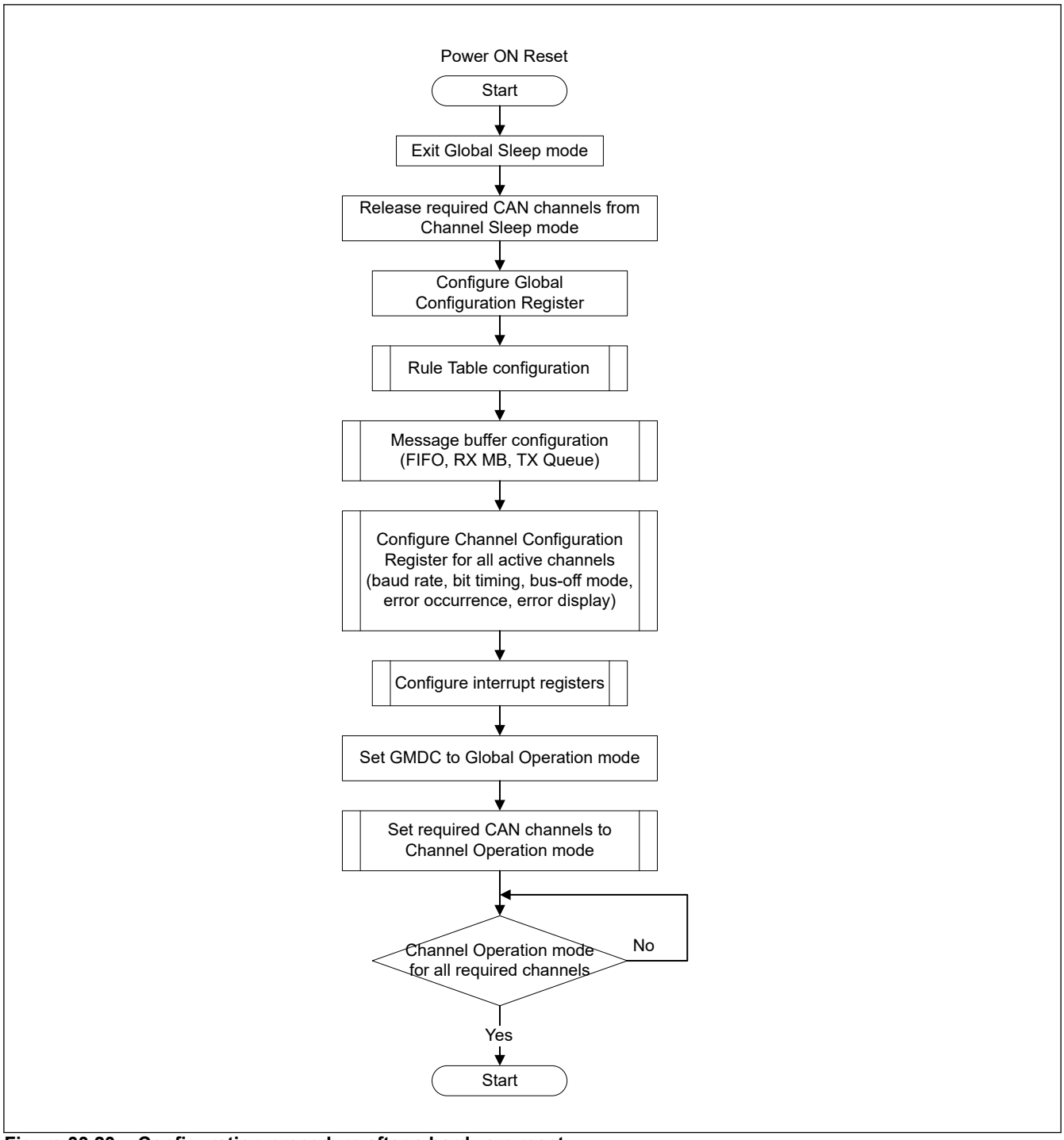


Figure 33.23 Configuration procedure after a hardware reset

33.5 Acceptance Filtering Function using Global Acceptance Filter List (AFL)

33.5.1 Overview

The CAN-FD module can handle message acceptance filtering for all channels with a global Acceptance Filter List (AFL). Each element of the AFL defines a filter rule for messages received on a specific channel.

The following actions are performed based on the AFL entries:

- Acceptance filtering based on received CAN identifier and masking
- DLC filtering based on received DLC value
- Message data payload according to the CFDGCFG.CMPOC bit
- Storage of accepted messages in the message buffer objects defined in the related AFL entry
- Attaching a 16-bit pointer to the stored messages defined in the related AFL entry, for example to support AUTOSAR applications
- Attaching a 2-bit information label to the stored messages defined in the related AFL entry

The 2-channel CAN-FD module allows a maximum of 256 AFL entries across all channels with a maximum of 128 AFL entries per channel.

During the acceptance filtering process, each AFL entry in a channel is checked against the received message by the acceptance filter unit. The check starts from the lowest AFL entry number for this channel.

The AFL search stops when a match of the received identifier with a configured identifier/mask combination occurs or when the received identifier has been compared against all AFL entries defined for the related channel. If no match occurs, then the received message is rejected. No notification is given to the application in this case.

Additionally, an automatic DLC filtering is performed for each accepted message if DLC check is globally enabled. If the DLC value of the received message is equal to or higher than the configured DLC value in the matching AFL entry, the DLC check is passed.

If DLC replacement (CFDGCFG.DRE bit) is enabled, DLC value configured in the matching AFL entry is greater than 0x0 and DLC check passes, then the configured value of DLC in the matching AFL entry is stored in the destination RXMB or FIFO Buffer.

If the received value of DLC is greater than the configured DLC value in the matching AFL entry, then the additional data bytes received on the CAN bus are not stored in the destination RXMB or FIFO Buffer. These additional data bytes are stored as 0x00 in the destination RXMB or FIFO Buffer.

If DLC replacement is enabled and DLC value of matching AFL entry is 0x0, then the received value of DLC is stored in the destination RX MB or FIFO Buffer.

If DLC replacement (the CFDGCFG.DRE bit) is disabled and DLC check passes, then the received value of DLC on the CAN bus is stored in the destination RXMB or FIFO buffer.

If the received value of DLC is greater than the configured DLC value in the matching AFL entry, then the additional data bytes received from the CAN bus are also stored in the destination RXMB or FIFO buffer.

If DLC value of the received message is less than the configured DLC value in the matching AFL entry, then DLC check fails. In this case, the received message is rejected and is not stored in any RXMB or FIFO buffer.

Additionally, DLC check failure is flagged by the DLC Error Flag in the Global Error Flag Register. If configured, an error interrupt is also generated. The DLC replacement configuration has no impact if the DLC check fails.

If a message has passed both acceptance filtering and DLC filtering, it is stored in a single reception message buffer and/or in FIFO buffers configured for reception or gateway function.

This message storage target information is also defined in the same AFL entry. Do not set a target at the AFL entry which is not configured.

Each accepted received message can be stored into a maximum of 8 different target destinations (single reception message buffer and/or FIFO buffers).

The programming of more than eight target destinations is not allowed. When more destinations are programmed then internal timing, a race condition can occur and received message may not be stored to the message RAM. Correct configuration of the numbers of target destination is the responsibility of the application.

Additional protection mechanism is made for the case when a received message contains more data payload bytes than possible to store in the target destination (CFDRMNB.RMPLS[2:0], CFDRFCCn.RFPLS[2:0] or CFDCFCCn.CFPLS[2:0]).

If CFDGCFG.CMPOC = 0, the message is completely rejected and is stored in the target destination. When CFDGCFG.CMPOC = 0 and RX or Common FIFO full including the received message contains more data payload bytes than possible to store in the target destination (CFDRMNB.RMPLS[2:0], CFDRFCCn.RFPLS[2:0] or CFDCFCCn.CFPLS[2:0]), the corresponding CFDFMSTS.RFXMLT[7:0] or CFDFMSTS.CFXMLT[5:0] bits are not set to 1, respectively.

When CFDGCFG.CMPOC = 1, the received data bytes greater than CFDRMNB.RMPLS is rejected. When CFDGCFG.CMPOC = 1 and RX or Common FIFO full including the received message contains more data payload bytes than possible to store in the target destination (CFDRMNB.RMPLS[2:0], CFDRFCCn.RFPLS[2:0] or CFDCFCCn.CFPLS[2:0]), the corresponding CFDFMSTS.RFXMLT[7:0] or CFDFMSTS.CFXMLT[5:0] bits are set to 1, respectively.

Depending on the CFDGCFG.DRE bit, the original received DLC or the DLC value configured at the AFL entry is stored.

Regardless of the CFDGCFG.CMPOC bit setting, CFDGERFL.CMPOF is set to 1 if a payload overflow condition is detected.

The DLC filtering is performed before the payload overflow function. So for one reception frame, only one flag can be set at the same time with CFDGERFL.DEF or CFDGERFL.CMPOF.

33.5.2 Allocation of AFL Entries to each CAN Channel

The number of AFL entries per channel can be configured using the dedicated field in the related Global Acceptance Filter Configuration Registers (see Figure 33.24).

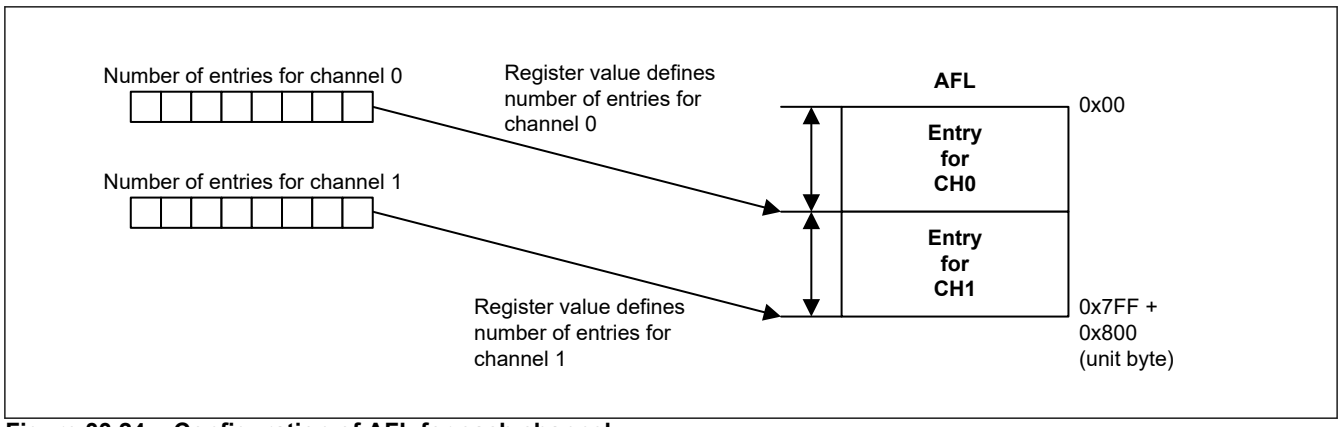


Figure 33.24 Configuration of AFL for each channel

The minimum number of entries for one channel is 0 (no entries defined for the channel) and the maximum number of entries for one channel is 128. The total number of entries for all channels should not exceed the maximum limit of (n + 1) × 128.

All entries are unique for a channel and overlapping or sharing of entries is not supported. Correct configuration of the AFL is the responsibility of the application.

The CAN-FD module does not flag errors related to the configuration of the AFL.

33.5.3 AFL Entry Description

Each AFL entry consists of 16 bytes. The fields in all entries are identical.

Each entry contains the following information for acceptance filtering and DLC filtering:

- Identifier (11 bits for Standard Frame format, 29 bits for Extended Frame format):
  Acceptance filter unit checks the identifier field of the received message against the identifier field of each AFL entry (full 29 bits masking of identifier bits is possible, see information that follows).
- IDE bit:
  Acceptance filter unit checks the IDE bit of the received message against this bit and selects the relevant part of the identifier field for acceptance filtering (masking of IDE bit is possible, see the information that follows).

- RTR bit:
  Acceptance filter unit only accepts data frames (RTR = 0) or remote frames (RTR = 1) according to the setting of this bit (masking of RTR bit is possible, see the information that follows).
- Loopback Configuration bit:
  This bit can enable or disable the AFL entry depending on the Loopback Configuration or Mirror mode condition.
- Mask for Identifier bits (29 bits):
  Each bit in the identifier mask can mask the corresponding identifier bit in the AFL entry during acceptance filtering, see Figure 33.25.
- Mask for IDE bit:
  If this Mask bit masks the IDE bit of the AFL entry in both Standard Identifier and Extended Identifier format, messages can be accepted by this AFL entry. The identifier of the received message is compared against the Standard Identifier part of the AFL entry for Standard Identifier format messages and against the Extended Identifier part of the AFL entry for Extended Identifier format messages.
- Mask for RTR bit:
  If this Mask bit masks the RTR bit of the AFL entry in both frame formats, data frame and remote frame formats are accepted by this AFL entry.
- Pointer information (16 bits):
  This 16-bit pointer is attached to a received message accepted by the related AFL entry. The pointer is added during message storage in the message buffer area and can be used by application as support function. The pointer information can be used for example to support PDU identifier allocation for the received message in AUTOSAR systems.
- Information label (2 bits):
  This 2-bit label is attached to a received message accepted by the related AFL entry. The label is added during message storage in the message buffer area and can be used by application as support function.
- DLC value for automatic DLC filtering:
  If the DLC value of the received message is equal or higher than the configured DLC value, the DLC check is passed. If the DLC value in this AFL entry is configured to 0, DLC filtering is effectively disabled for this entry (all accepted messages pass DLC filtering).

Each AFL entry contains the following information for the handling of received messages:

- Message buffer number of one single reception message buffer as target for received message storage
- Single reception message buffer enable bit to configure the single reception message buffer number to be valid or invalid, as target for received message storage
- FIFO direction pointer - each bit of the FIFO direction pointer configures a dedicated FIFO as possible target for a received message

Note: A message received on channel A can be routed to Common FIFO buffer of another channel. If this Common FIFO buffer is configured in Gateway mode, then the message stored in this Common FIFO Buffer is transmitted on that channel because Common FIFO buffer is associated with channel.

There is no hardware protection against such storage of message. Therefore, the FIFO direction pointer must be configured carefully.

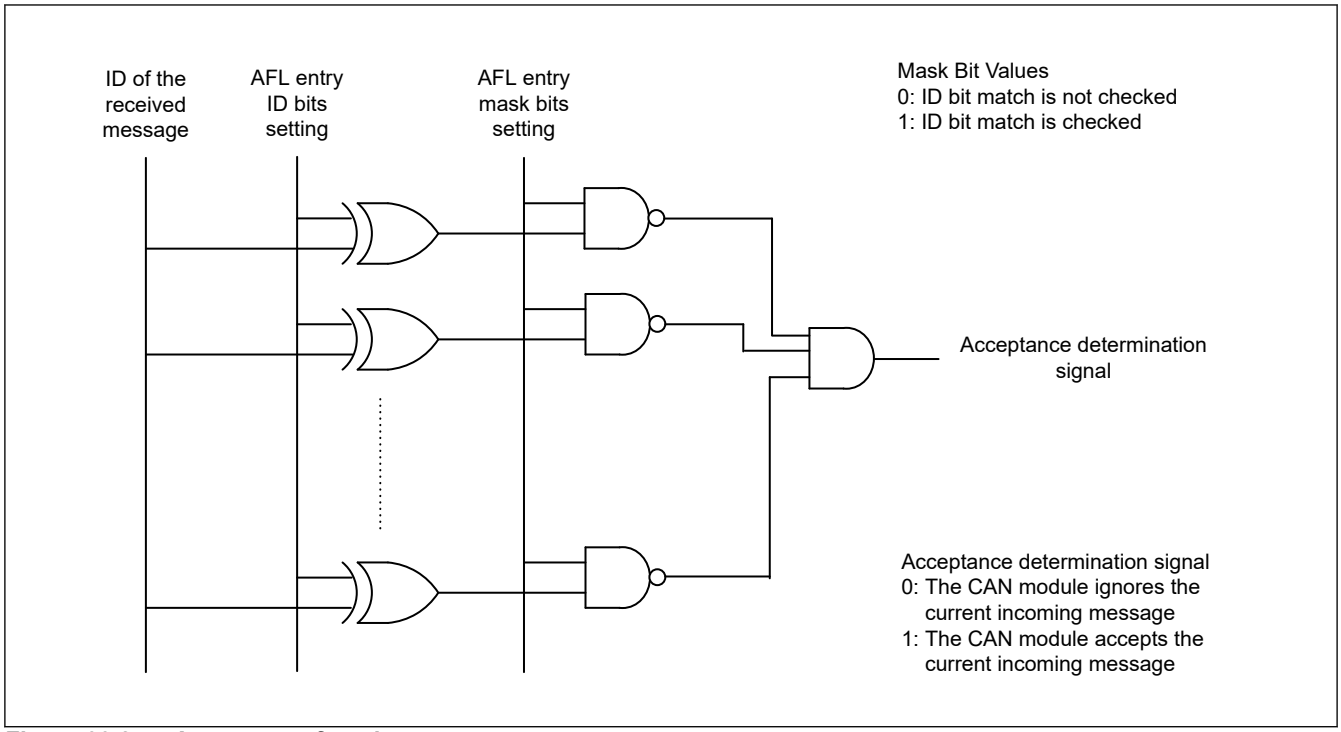


Figure 33.25 Acceptance function

33.5.4 Entering Entries in the AFL

Application software can enter one full entry into the AFL using the following registers:

- Global AFL ID Entry Register: Part 1 of the AFL entry
- Global AFL Mask Entry Register: Part 2 of the AFL entry
- Global AFL Pointer 0 Entry Register: Part 3 of the AFL entry
- Global AFL Pointer 1 Entry Register: Part 4 of the AFL entry

16 sets of these registers form a group of AFL entries. Each group can be accessed through a page mechanism. For the CAN-FD module, 2-channel version, 16 of these pages exist to allow access to the whole AFL range. The AFL should only be configured in CH_RESET or CH_HALT mode. Pages are linked to the AFL entries in the following way:

Table 33.21 Mapping between pages and AFL entries (1 of 2)

Page 0	Entry 0 – 15
Page 1	Entry 16 – 31
Page 2	Entry 32 – 47
Page 3	Entry 48 – 63
Page 4	Entry 64 – 79
Page 5	Entry 78 – 95
Page 6	Entry 96 – 111
Page 7	Entry 112 – 127
Page 8	Entry 128 – 143
Page 9	Entry 144 – 159
Page 10	Entry 160 – 175
Page 11	Entry 176 – 191
Page 12	Entry 192 – 207
Page 13	Entry 208 – 223

Table 33.21 Mapping between pages and AFL entries (2 of 2)

Page 14	Entry 224 – 239
Page 15	Entry 240 – 255

The selection of the AFL access page is done using the Global Acceptance Filter List Entry Control Register (CFDGAFLECTR) (see Figure 33.26). This register has the following fields:

- 4 bits to select the AFL page number
- 1 bit to enable or disable the AFL data access to prevent unwanted write access to the AFL

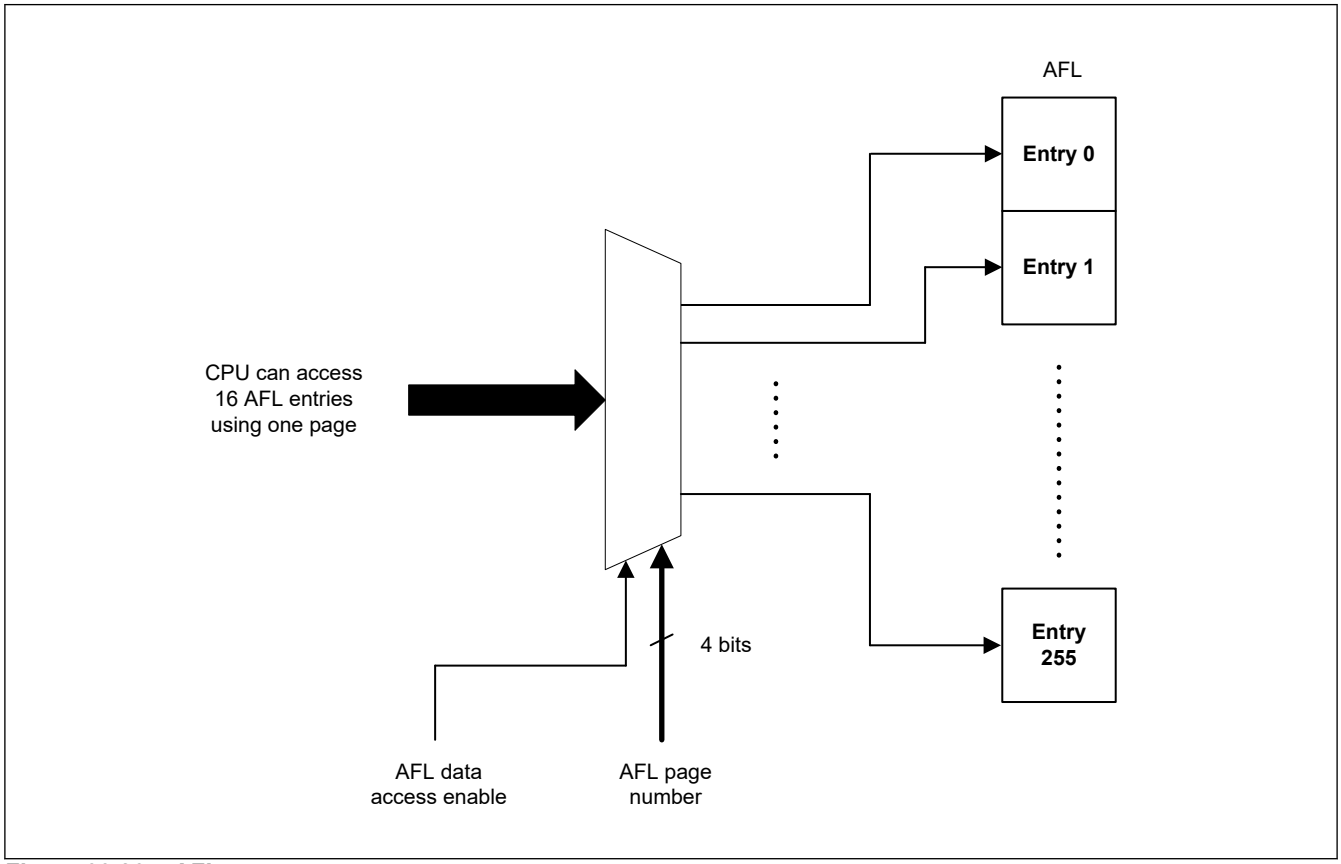


Figure 33.26 AFL page access

Application software should not write numbers higher than 0xF for the AFL page number.

Follow the configuration shown in Figure 33.27 to program the AFL.

After entering all entries in Configuration mode, locking of the AFL access should be performed to protect unwanted write access to the AFL.

Write protection is active during all Global modes (GL_RESET, GL_HALT, and GL_OPERATION) if the lock bit is set.

Read access to AFL is still possible during all Global modes even when AFL data access is disabled (consistency check of AFL contents is possible during run time).

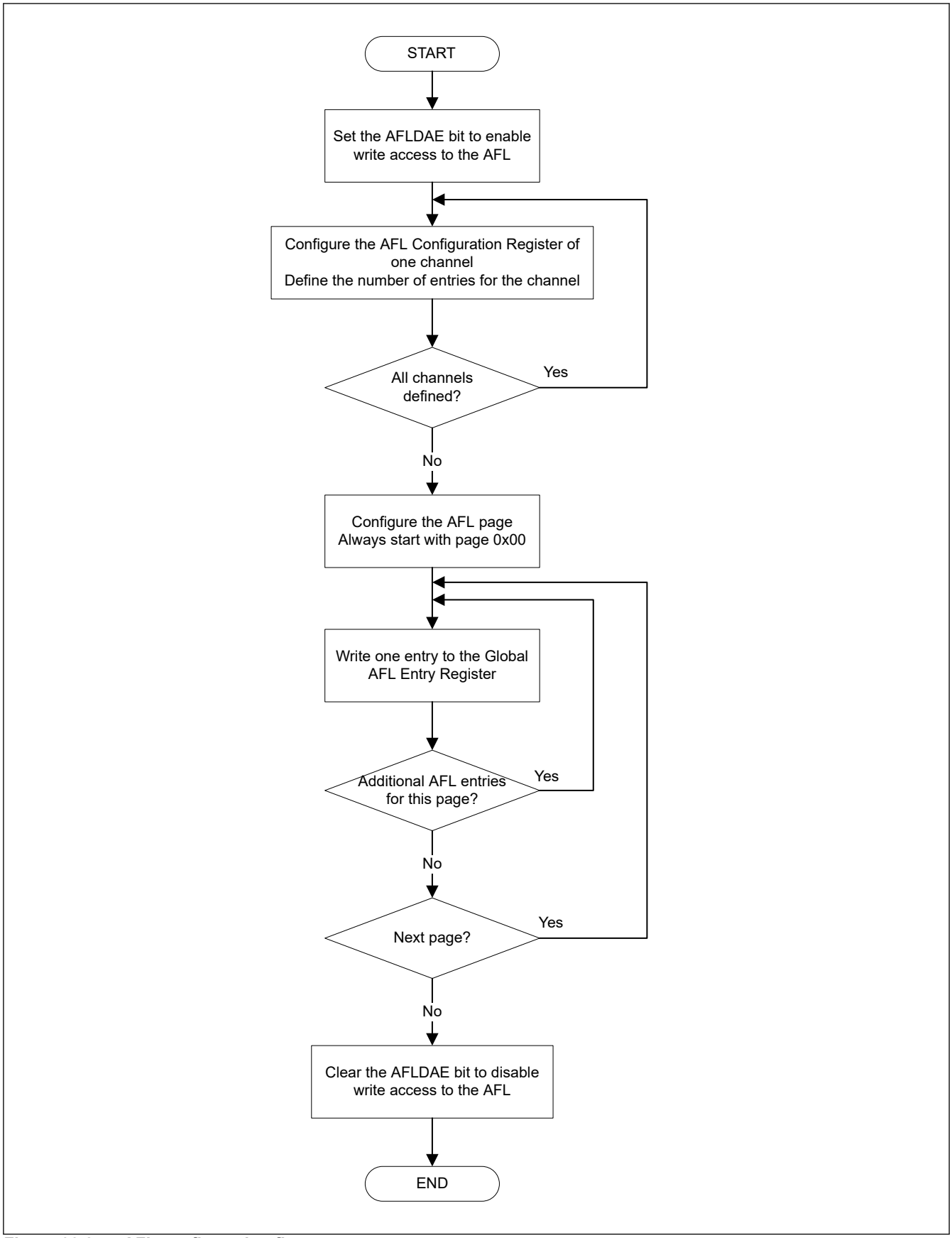


Figure 33.27 AFL configuration flow

33.5.5 Loopback Modes

If the Loopback Configuration bit is set, the AFL entry is only valid in loopback test mode (Self test mode 0 or Self test mode 1) or in mirror mode when receiving messages that were transmitted by the respective CAN channel itself.

The AFL entry is not valid for received messages in loopback mode transmitted by other CAN nodes on the bus. The expression valid or invalid for the related entry means that this AFL entry is or is not compared against the received message ID respectively.

If the Loopback Configuration bit is 0, the AFL entry is only valid for:

- Received messages transmitted by other CAN nodes on the bus in normal (non-loopback mode) and mirror modes
- Received messages transmitted by other CAN nodes or the CAN channel itself in loopback test mode.

The mirror mode can be enabled with the CFDGCFG.MME bit in the Global Configuration Register. If CFDGCFG.MME bit is set, then a successfully transmitted message can be stored back in an RX message buffer or FIFO buffer if a matching entry is configured in the AFL for that channel.

The Loopback Configuration bit in the matching AFL entry must be set to store this frame.

If mirror mode and loopback test mode are configured at the same time, the loopback test mode behavior applies.

Table 33.22 shows the behavior of the acceptance filter unit depending on the setting of the related input signals.

Table 33.22 Behavior of acceptance filter based on the loopback configuration setting in AFL entry

Mirror mode enable (MME Configuration bit)	Loopback in Test mode (Self test mode 0 or Self test mode 1)	Channel mode	Loopback Configuration bit in AFL entry	AFL entry
0	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Invalid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid
1	0	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Invalid
			1	Valid
	1	Receiver	0	Valid
			1	Invalid
		Transmitter	0	Valid
			1	Valid

Note: The expression valid or invalid for the related entry means that this AFL entry is or is not compared against the received message ID, respectively.

33.5.6 IDE Masking

When the GAFLIDEM bit is 0 in an AFL entry, the IDE bit configured in the AFL entry is not used for ID matching. In this case, the use of ID[10:0] or ID[28:0] matching is based on the received IDE bit.

Consider the following example:

- The ID and Mask fields of an AFL entry x is configured as follows:
  - CFDGAFLID [x] = 0xC0553A20 → IDE = 1, RTR = 1, LLB = 0, ID[10:0] = 0x220 / ID[28:0] = 0x0553A20
  - CFDGAFLMn = 0x0000FFFF → IDEM = 0, RTRM = 0, IDM[10:0] = 0x7FF / IDM[28:0] = 0x0000FFFF
- The comparison result for the four different received IDs with AFL entry x is described as follows:

– If a frame with IDE = 0 and ID = 0x220 is received, this is considered as a match
– If a frame with IDE = 0 and ID = 0x320 is received, this is not a match
– If a frame with IDE = 1 and ID = 0x1FFF3A20 is received, this is considered as a match
– If a frame with IDE = 1 and ID = 0x08803220 is received, this is not a match.

33.6 FIFO Buffers and Normal Message Buffer Configuration

This section describes the process for configuring the number of RX message buffers, the FIFO buffers, and the flat TX message buffers in the CAN-FD module. The message buffers are mapped as shown in Figure 33.28.

The RX message buffers can be accessed with the RX Message Buffer Registers.

The RX FIFO buffers and the common FIFO buffers configured in RX mode, TX mode, or GW mode can only be accessed with the FIFO Access Registers.

If the common FIFO is configured in TX mode, you can only write data into the FIFO buffer using the FIFO Access registers.

If the common FIFO is configured in GW mode or RX mode, you can only read data from the FIFO Access Registers.

The TX message buffers can be accessed with the TX Message Buffer Registers.

If unused message buffer locations are read, the message buffer locations are read as unknown values.

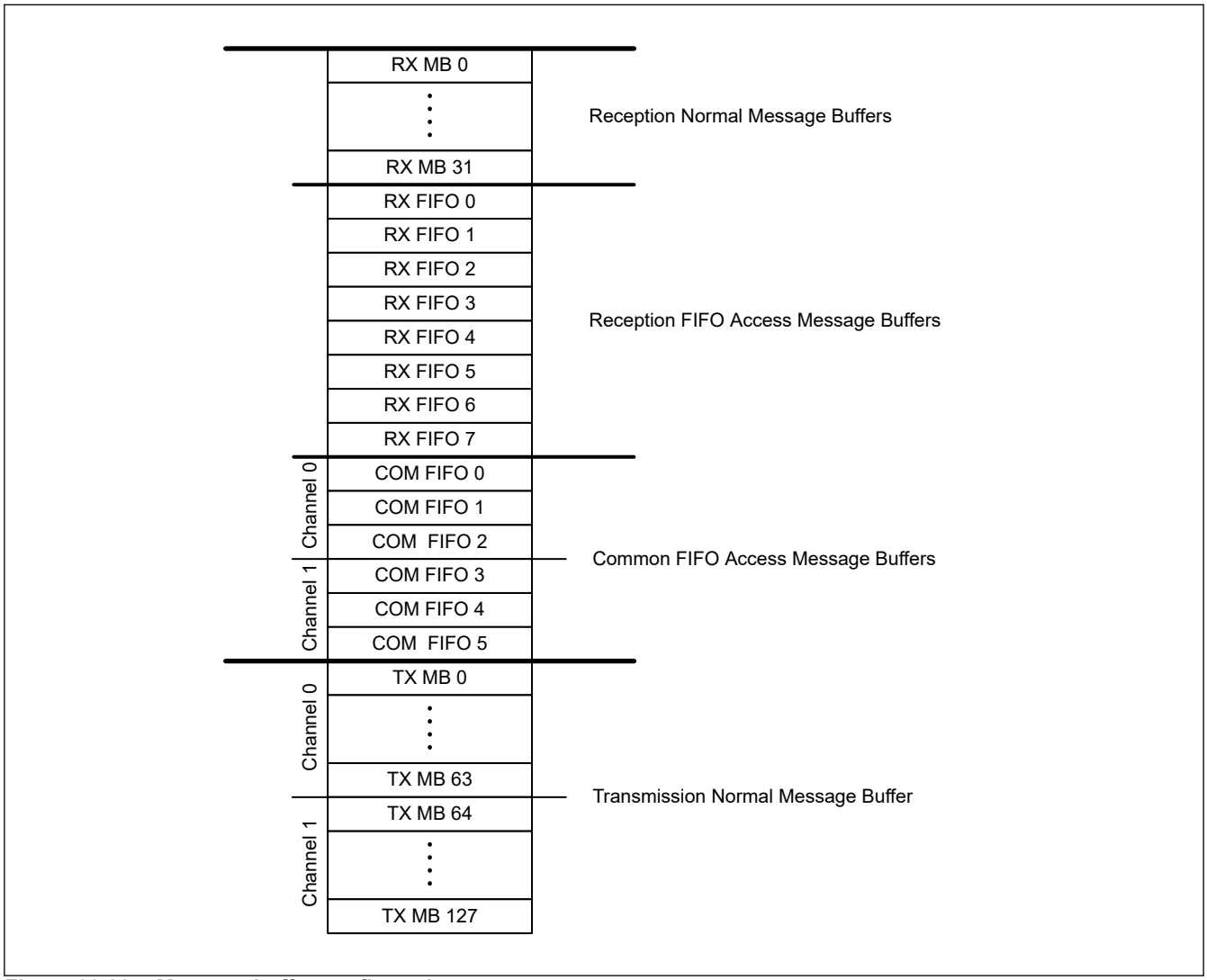


Figure 33.28 Message buffer configuration

33.6.1 Normal RX Message Buffers

In CAN-FD module, the frames received by various channels can be stored in normal RX message buffers based on the configuration of the AFL entries.

Additionally, the number of normal RX message buffers required in the system can be chosen up to a fixed maximum limit.

33.6.1.1 Normal RX Message Buffer Configuration

In CAN-FD module, the number of normal RX message buffers can be configured by writing to the RX Message Buffer Number Register.

The limiting values for the configuration of number of message buffers are:

- Minimum value = 0x00 (no normal RX MB)
- Maximum value = (16 × number of CAN channels) = 16 × 2 = 32 = 0x20 (32 flat RX MBs for 2 channels)

Do not use values outside these limits.

The AFL entries for routing the received messages to normal RX message buffers must be configured to match the requirements of the system.

The AFL entries must also be configured properly, and an AFL entry for normal RX message buffers should not exceed the number of message buffers configured in the RX Message Buffer Number Register.

Note: There is no internal check procedure provided in CAN-FD module against incorrect configuration of the AFL.

The data field size of the RX message buffer can be configured with the CFDRMNB.RMPLS[2:0] bits. The default size is 8 bytes and the maximum data payload size is 64 bytes.

When the receiving frame exceeds the data field size, then the acceptance depends on the configuration of CFDGCFG.CMPOC (message rejecting or data payload cut).

33.6.2 FIFO Buffer

The CAN-FD module provides a fixed number of FIFO buffers to support storage of frames for reception, transmission and gateway functions for various CAN channels.

The number of reception-only FIFO buffers is fixed to 8. However, 3 common FIFO buffers per channel can be configured to store messages for transmission, reception, or gateway function.

These FIFO buffers can be enabled or disabled, and the following parameters can be configured to match the system requirements:

- Size
- Interrupt structure
- Message lost mechanism
- Message overwrite mechanism of the FIFO buffers
- Location of the TX FIFO or GW FIFO

When the receiving frame exceeds the data field size, the acceptance depends on the configuration of the CFDGCFG.CMPOC bit (message rejecting or data payload cut).

33.6.2.1 FIFO Buffers Configuration

In CAN-FD module, the FIFO buffers can be configured to match the system requirements.

The total number of FIFO buffers = 8 RX FIFO buffers + 6 common FIFO buffers = 14 FIFO buffers for 2 channels and message overwrite mechanism.

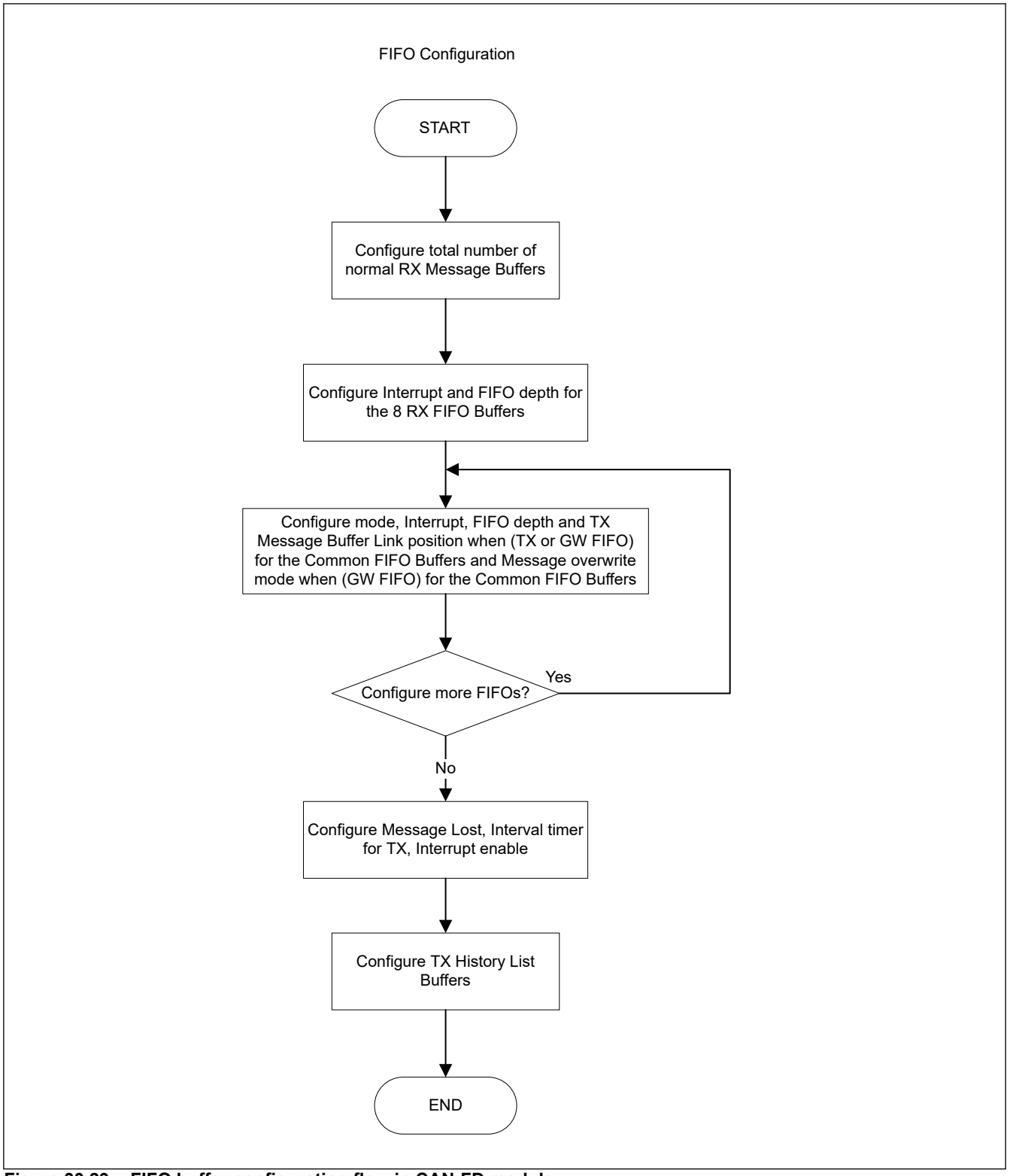


Figure 33.29 FIFO buffer configuration flow in CAN-FD module

As shown in Figure 33.29, the various FIFO buffers can be configured by writing to the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers.

For the 8 RX FIFO buffers, the following parameters can be configured:

- Interrupts
- FIFO depth
- FIFO payload data size

For the common FIFO buffers, the following parameters can be configured:

- Mode
- Interrupts FIFO depth
- FIFO payload data size
- FIFO TX link position

(1) FIFO Mode Configuration of Common FIFO Buffers

The mode of the common FIFO buffers can be configured by writing to the CFDCFCCn.CFM[1:0] bits in the Common FIFO Configuration/Control Registers. The possible modes of configuration for Common FIFO buffers are:

- 00b: RX mode (default mode after hardware reset)
- 01b: TX mode
- 10b: GW mode
- 11b: Reserved (do not write this value to the register bits)

Messages can only be read from the RX FIFO buffers and the Common FIFO buffers configured in RX mode. Messages are stored by the CAN module in these FIFO buffers based on the AFL entries.

Messages can be read and written into the Common FIFO buffers configured in TX mode. These messages are transmitted on the appropriate CAN channel.

Messages can only be read from the Common FIFO buffers configured in GW mode. However, the CPU read access has no impact on the read or write pointers. The pointers can only be incremented when a new message is stored in the FIFO buffer and decremented when a message is transmitted on the corresponding CAN channel by the CAN-FD module.

After a hardware reset, all the Common FIFO buffers are configured in RX mode by default. Only enable the FIFO buffers after configuring the Common FIFO buffers in the required modes.

In GW mode, when a transmit/receive FIFO buffer is trying to receive a new message while the transmit/receive FIFO buffer is already full of data, the oldest data of the buffer is overwritten with the message received or the message is discarded. The behavior is determined by setting the CFDCFCCEn.CFMOWM bit.

- When CFDCFCCEn.CFMOWM = 0:
  When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the received message is discarded, and CFDCFSTSn.CFMLT bit is set to 1.
- When CFDCFCCEn.CFMOWM = 1:
  When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the oldest data in the buffer is overwritten with the received message. The read pointer of the transmit/receive FIFO buffer simultaneously moves to the next oldest message. The CFDCFSTSn.CFMOW bit is then set to 1, which notifies that the oldest message has been overwritten with the received message.
  In addition, when a CAN bus error or arbitration-lost for the transmitting message occurs in the transmit/receive FIFO buffer full, the transmitting message is lost and retransmission for the message is not performed. The read point is then moves to the next message automatically. Do not write to this bit when the CFDCFCCn.CFE bit is 1.

(2) FIFO TX Message Buffer Link Configuration

When the common FIFO is configured as TX or GW FIFO, the FIFO buffer must be linked to a normal TX message buffer to participate in the transmission scan of a CAN channel.

The link to a normal TX message buffer must be unique, for example the same TX message buffer cannot be shared between 2 or more common FIFO buffers.

Do not write data into a TX message buffer that is linked to a Common FIFO buffer. Also, the TX message buffer linked to a Common FIFO buffer should not be a part of the TX Queue.

The TX message buffer link of each Common FIFO buffer can be configured by writing to the CFDCFCCn.CFTML[4:0] bits in the Common FIFO Configuration/Control Registers. Available options for TX message buffer link configuration are:

- 0x00: TX Message Buffer 32
- 0x01: TX Message Buffer 33
  ⋮

- 0x1F: TX Message Buffer 63

(3) FIFO Depth Configuration

The depth of each FIFO buffer can be configured by writing to the CFDRFCCn.RFDC[2:0] bits and CFDCFCCn.CFDC[2:0] bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/ Control Registers. The 8 available options for depth configuration are:

- 000b: 0 Messages (FIFO buffer cannot be enabled)
- 001b: 4 Messages
- 010b: 8 Messages
- 011b: 16 Messages
- 100b: 32 Messages
- 101b: 48 Messages
- 110b: 64 Messages
- 111b: 128 Messages

The RAM allocation for RX message buffers along with FIFO buffers is limited to (n + 1) × 256 messages. Configuration of the RX message buffers, along with FIFO buffers, that exceeds this maximum limit should not be done.

CAN-FD module logic does not check the validity of the configuration.

Note:
- If the FIFO depth of a common FIFO is 4 messages or more (CFDCFCCn.CFDC[2:0] > 000b), then the Common FIFO TX message buffer link is valid when the FIFO is disabled or enabled.
- If FIFO depth is 0 messages, then the Common FIFO TX message buffer link is not valid when the FIFO is disabled or enabled.

(4) **FIFO Payload Size Configuration**

The data size of each FIFO buffer can be configured by writing to the CFDRFCCn.RFPLS[2:0] bits and CFDCFCCn.CFPLS[2:0] bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/ Control Registers. The eight available options for depth configuration are:

- 000b: 8 bytes
- 001b: 12 bytes
- 010b: 16 bytes
- 011b: 20 bytes
- 100b: 24 bytes
- 101b: 32 bytes
- 110b: 48 bytes
- 111b: 64 bytes

(5) FIFO Interrupt Configuration

The interrupt generation conditions for the FIFO buffers can be configured by writing to the CFDRFCCn.RFIM bit and the CFDCFCCn.CFIM bit in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers. The two available options are:

- 0:
  - RX FIFO mode: Interrupt generated when the Common FIFO counter reaches CFDRFCCn.RFIGCV/ CFDCFCCn.CFIGCV value
  - TX FIFO mode: Interrupt generated when the Common FIFO transmits the last message successfully
  - GW FIFO mode
    Frame RX: Interrupt generated when message counter increments and reaches the interrupt threshold value
    Frame TX: Interrupt generated when the last message is transmitted successfully from FIFO
- 1:

– RX FIFO mode: Interrupt generated at the end of storage of every received message
– TX FIFO mode: Interrupt generated for every successfully transmitted message
– GW FIFO mode
Frame RX: Interrupt generated when message is stored in the FIFO
Frame TX: Interrupt generated when message is successfully transmitted from the FIFO

If the Interrupt Mode bit is 0 for a RX FIFO, then interrupt is generated based on the configuration of the CFDRFCCn.RFIGCV[2:0] bits.

Similarly, if the Interrupt Mode bit is 0 for a Common FIFO configured in RX mode, then interrupt is generated based on the configuration of CFDCFCCn.CFIGCV[2:0] bits.

The eight available options for configuring the FIFO counter value for generation of an interrupt are:

- 000b: Interrupt generated when FIFO is 1/8th Full
- 001b: Interrupt generated when FIFO is 1/4th Full
- 010b: Interrupt generated when FIFO is 3/8th Full
- 011b: Interrupt generated when FIFO is 1/2 Full
- 100b: Interrupt generated when FIFO is 5/8th Full
- 101b: Interrupt generated when FIFO is 3/4th Full
- 110b: Interrupt generated when FIFO is 7/8th Full
- 111b: Interrupt generated when FIFO is Full

In this case, an interrupt is generated when the message count matches the configured value.

However, there are some limitations on the configuration of the CFDRFCCn.RFIGCV[2:0] and CFDCFCCn.CFIGCV[2:0] bits depending on the FDC[2:0] bits (FIFO Depth Configuration), see Table 33.23.

Table 33.23 FIFO interrupt generation counter and FIFO depth configuration

RFDC[2:0] (FDC[2:0])	RFIGCV[2:0] (CFGCV[2:0])							
	111b	110b	101b	100b	011b	010b	001b	000b
000b	Don't care (FIFO cannot be enabled)							
001b	Allowed	Not allowed	Allowed	Not allowed	Allowed	Not allowed	Allowed	Not allowed
010b	Allowed							
011b	Allowed							
100b	Allowed							
101b	Allowed							
110b	Allowed							
111b	Allowed							

Common FIFO buffer can set an interrupt output at the completion time of transmitting one frame, or the completion of reception. In addition, Common FIFO and RX FIFO can set an interrupt output, when stored to the setup number (CFDC/RFDC) of FIFO stages.

33.6.2.2 FIFO Buffers Control

The FIFO interrupt must be enabled by setting any one of the following bits in the RX FIFO Configuration/Control Registers:

- CFDRFCCn.RFIE
- CFDRFCCn.RFFIE

In addition, the FIFO interrupt must be enabled by setting any one of the following bits in the Common FIFO Configuration/Control Registers:

- CFDCFCCn.CFRXIE

- CFDCFCCn.CFTXIE
- CFDCFCCEn.CFFIE
- CFDCFCCEn.CFOFRXIE
- CFDCFCCEn.CFOFTXIE

After configuration is complete, each FIFO can be enabled by setting the CFDRFCCn.RFE and CFDCFCCn.CFE bits in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers to allow transmission and reception of messages.

When CFDCFCCEn.CFBME = 1, it becomes FIFO buffering mode, send data is stored in the Common FIFO, and transmission is stopped. Transmission starts when CFDCFCCEn.CFBME = 0.

Do not write 1 from 0 for this bit when the CFDCFCCn.CFE bit is 1.

33.7 Interrupt and DMA

33.7.1 Interrupts

The CAN-FD module generates several interrupts. The interrupt output, which is connected to the Interrupt Controller Unit (ICU), can be controlled by the corresponding interrupt enable bit.

The status flag is set independent from this enable bit.

The channel transmission interrupt has an additional Status Flag register. These status bits are only set when the corresponding interrupt enables are set.

The Status Flag register supports the identification of the interrupt source for the channel transmission, as this interrupt is driven by several trigger sources.

The interrupts in the CAN-FD module can be classified into 2 groups, global interrupts and channel interrupts:

- Global interrupts:
  The CAN-FD module can generate 2 global interrupts:
  1. One Global interrupt for successful reception into the 8 RX FIFO buffers
  2. One Global error interrupt
- Channel interrupts:
  Each channel of the CAN-FD module can generate 3 channel interrupts:
  1. Channel transmission
     Transmission completion from channel
     Transmission abort from channel
     Transmission from TX Queue for a channel
     Channel THL Interrupt
     Successful transmission from a Common FIFO in TX or GW mode for a channel
  2. Channel error interrupt
  3. Successful reception in a Common FIFO in RX or GW mode for a channel or successful routing in a TXQ

The interrupts are cleared when the corresponding flag bits are cleared or interrupt enable bits are cleared.

Table 33.24 provides an overview of interrupt.

Table 33.24 Overview of interrupt source

Parameter	Interrupt	Interrupt source	Interrupt clearing
Global Interrupts	Successful reception into at least one RX FIFO	Interrupt flag of corresponding RX FIFO for which interrupt is enabled	Clear the interrupt flag of corresponding RX FIFO buffer for which interrupt is enabled
	FIFO full into at least one RX FIFO	FIFO Full Interrupt flag of corresponding RX FIFO for which interrupt is enabled	Clear the FIFO Full Interrupt flags of corresponding RX FIFO buffer for which interrupt is enabled
	Global error	Any of the following: • DLC Error flag • Message Lost Status bit • Message Overwrite Status bit • TXQ Message Lost Status bit • TXQ Message Overwrite Status bit • TX History Entry Lost Status bit • CAN-FD Message Payload overflow flag	Clear all of: • DLC Error flag • Message Lost flags in all of the FIFO Status Registers • Message Overwrite flags in all of the Common FIFO Status Registers • Message Lost flags in all of the TXQ Status Registers • Message Overwrite Flags in all of the TXQ Status Registers • TX History List Entry Lost flag • CAN-FD Message Payload Overflow flag
Channel Transmission Interrupts	Channel n successful transmission	Any channel related TX MB Successful flag when interrupt is enabled Separate interrupts are provided for common FIFO buffers and TX Queue*1	Clear all channel related TX MB Result Status bits for which the interrupt is enabled
	Channel n abort	Any channel related TX MB Abort flag when interrupt is enabled Separate interrupts are provided for common FIFO buffers and TX Queue*1	Clear all channel related TX MB Result Status bits for which the interrupt is enabled globally
	Channel n transmission from TX Queue	Related channel TX Queue Interrupt flag	Clear related channel TX Queue Interrupt flag
	Channel n THL Interrupt	Channel n THL Interrupt Status flag	Clear the relevant THL Interrupt Status flag
	Channel n COM FIFO TX Interrupt	Interrupt flag for Common FIFOs in TX or GW mode belonging to the related channel	Clear the interrupt flags of Common FIFOs in TX or GW mode belonging to the related channel
	Channel n COM FIFO One Frame TX Interrupt	One Frame Transmission Interrupt flag for Common FIFOs belonging to the related channel	Clear the One Frame Transmission Interrupt flags of Common FIFOs belonging to the related channel
	Channel n TXQ One Frame TX Interrupt	One Frame Transmission Interrupt flag for TXQs belonging to the related channel	Clear the One Frame Transmission Interrupt flags of TXQs belonging to the related channel
Channel COM RX FIFO Interrupt	Channel n COM FIFO RX Interrupt	Interrupt flag for Common FIFOs in RX or GW mode belonging to the related channel	Clear the interrupt flags of Common FIFOs in RX or GW mode belonging to the related channel
	Channel n COM FIFO One Frame RX Interrupt	One Frame Reception Interrupt flag for Common FIFOs belonging to the related channel	Clear the One Frame Reception Interrupt flags of Common FIFOs belonging to the related channel
	Channel n COM FIFO Full Interrupt	FIFO Full Interrupt flag for Common FIFOs in RX or GW mode belonging to the related channel	Clear the FIFO Full Interrupt flags of Common FIFOs in RX or GW mode belonging to the related channel
	Channel n TXQ One Frame Routing Interrupt	One Frame Routing Interrupt flag for TXQs in GW mode belonging to the related channel	Clear the One Frame Routing Interrupt flags of TXQs in GW mode belonging to the related channel
	Channel n TXQ Full Interrupt	TXQ Full Interrupt flag for TXQs in GW mode belonging to the related channel	Clear the FIFO Full Interrupt flags of TXQs in GW mode belonging to the related channel
Channel Error Interrupt	Channel n Error	Any channel related error flag in the Channel Error Flag Register for which interrupt is enabled in the Channel Error Interrupt Enable Register	Clear all channel related error flags in the Channel Error Flag Register for which interrupt is enabled in the Channel Error Interrupt Enable Register

Note 1. These interrupts are only set for TX Message Buffers that do not belong to an enabled TX Queue and are not pointing to a common FIFO.

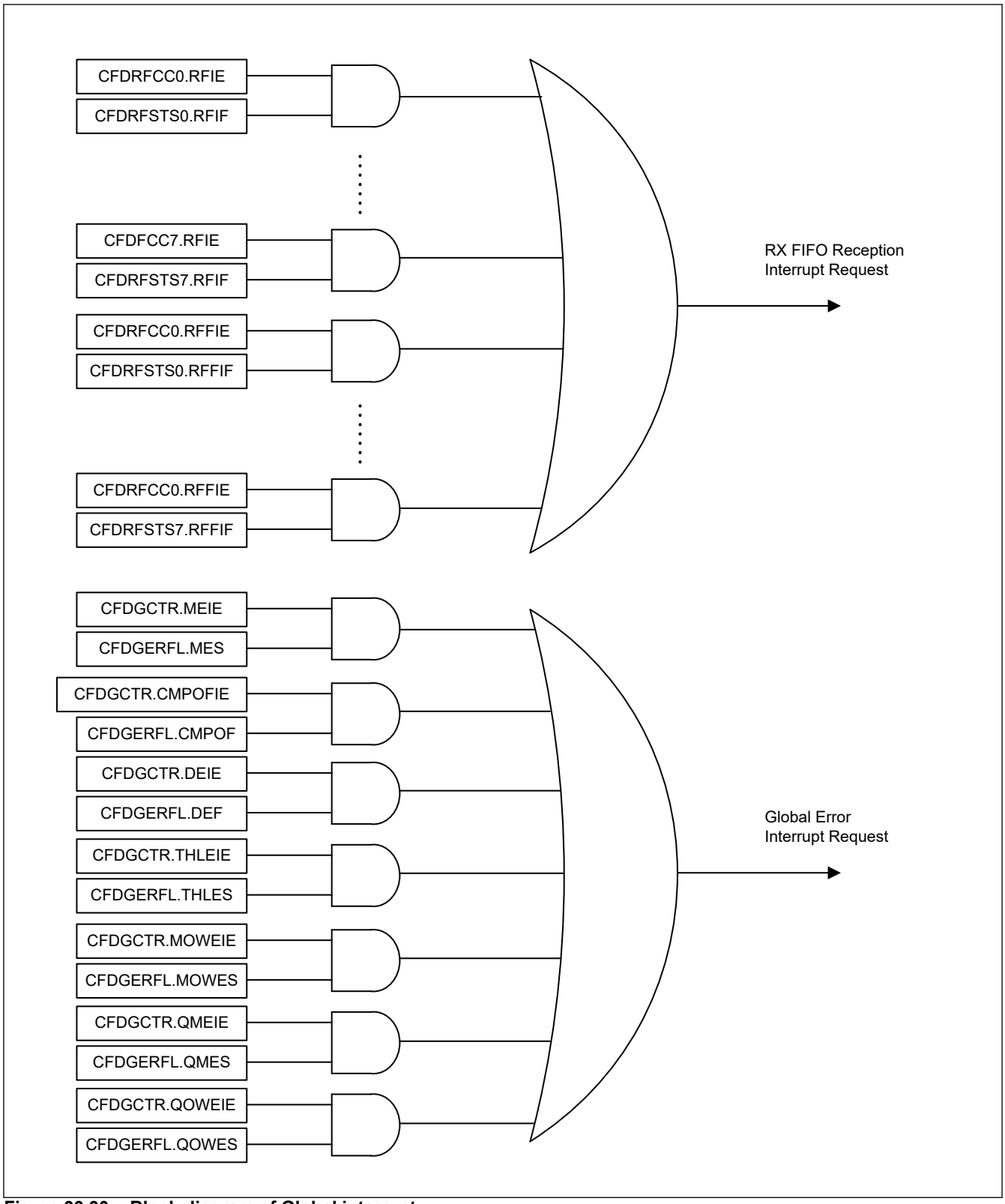


Figure 33.30 Block diagram of Global interrupt

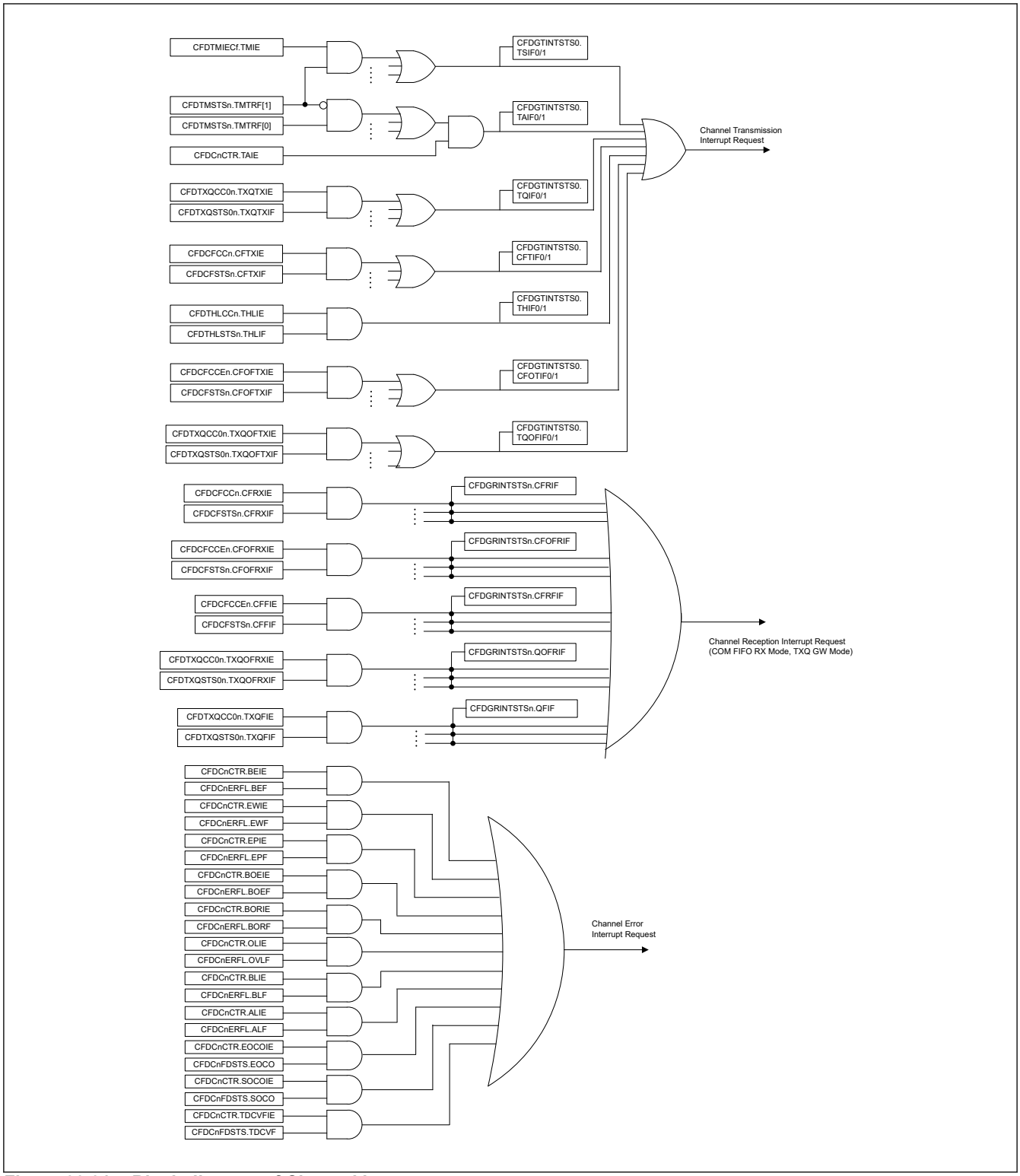


Figure 33.31 Block diagram of Channel interrupt

33.7.2 DMA Transfer

The CAN-FD module has some message buffer which can be associated with a DMA channel:

- Reception DMA
  - 8 RX FIFO Message Buffers
  - 2 Common FIFO Message Buffers
- Transmission DMA

– 16 TXQ Message Buffers (TXQ0, TXQ3)
– 2 Common FIFO Message Buffer

Figure 33.32 shows the potential DMA channels.

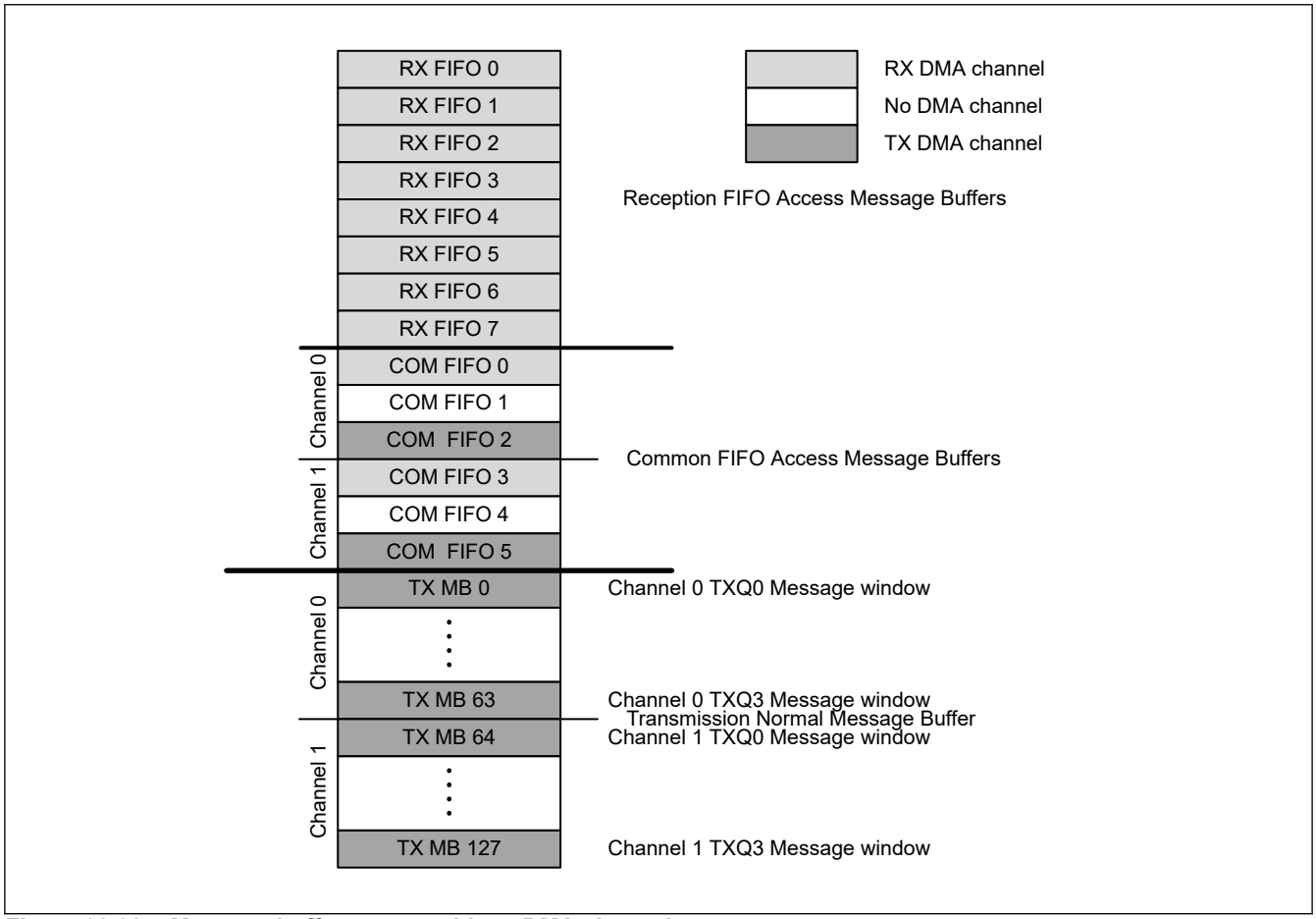


Figure 33.32 Message buffer connectable to DMA channel

A DMA channel transfer request is generated for each FIFO entry to the DMAC when the related CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE is set to 1 and the belonging FIFO is not empty.

Reception FIFO interrupt should be disabled for this particular FIFO (CFDRFCCn.RFIE or CFDCFCCn.CFRXIE).

Use the regular start address for the DMA access window address and add 0x8000 to the regular start address for the debugger access window. See Table 33.25.

Table 33.25 DMA channel access window address

b = Message buffer component.index	MBCP	Register	p	Start address n = 0, 1
0 to number_of_RFMBCPs - 1	RFMBCPb	RFID	x	0x6000 + b × 0x0080
		RFPTR	x	0x6004 + b × 0x0080
		RFFDSTS	x	0x6008 + b × 0x0080
		RFDFp	0 to 15	0x600C + p × 0x0004 + b × 0x0080
0 to number_of_CFMBCPs_per_channel - 1	CFMBCPb	CFID	x	0x6400 + b × 0x0080 + n × 0x0180
		CFPTR	x	0x6404 + b × 0x0080 + n × 0x0180
		CFFDCSTS	x	0x6408 + b × 0x0080 + n × 0x0180
		CFDFp	0 to 15	0x640C + p × 0x0004 + b × 0x0080 + n × 0x0180

DMA FIFO pointer decrement is done automatically by reading the last configured data payload byte (CFDRFCCn.RFPLS or CFDCFCCn.CFPLS).

Note: The DMA must read the exact length of the configured data payload size (CFDRFCCn.RFPLS or CFDCFCCn.CFPLS).

The software debugger must access outside of the regular SFR address range from 0xE000 to 0xEFFF.

Do not write to the FIFO and TXQ control registers when DMA is enabled.

The DMA enable of the particular DMA FIFO (CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE) can be set at any time, the procedure in this section is a configuration flow for an initial setup.

When CFDCDTTCT.TQ0DMAE or CFDCDTTCT.TQ3DMAE or CFDCDTTCT.CFDMAE is set, the messages of the corresponding TXQ or Common FIFO can be handled by DMA controller.

Use the following procedure when the TXQ or the Common FIFO can be handled by DMA controller.

1. CPU checks the TXQ or the Common FIFO is not full.
2. When transmit data can be used, CPU sets this data to Common FIFO or TXQ.
   When using Common FIFO, transmit data is write in CFDCFID, CFDCFPTR, CFDCFFDCSTS, and CFDTMBCPb[i] registers.
   When using TXQ, transmit data is write in CFDTMID, CFDTMPTR, CFDTMFDCTR, and CFDTMDFp registers.
3. For Common FIFO, the common FIFO pointer is incremented automatically when DMA controller writes the last data payload byte configured by CFDCFCCn.CFPLS.
   For TXQ, if the data of 64 data payload is written, a TXQ pointer increases automatically. When payload data is less than 64 bytes, dummy data must be written in and 64 data payload size must be done.

   Note: Only 32-bit write-access can be possible on the DMA message handling.

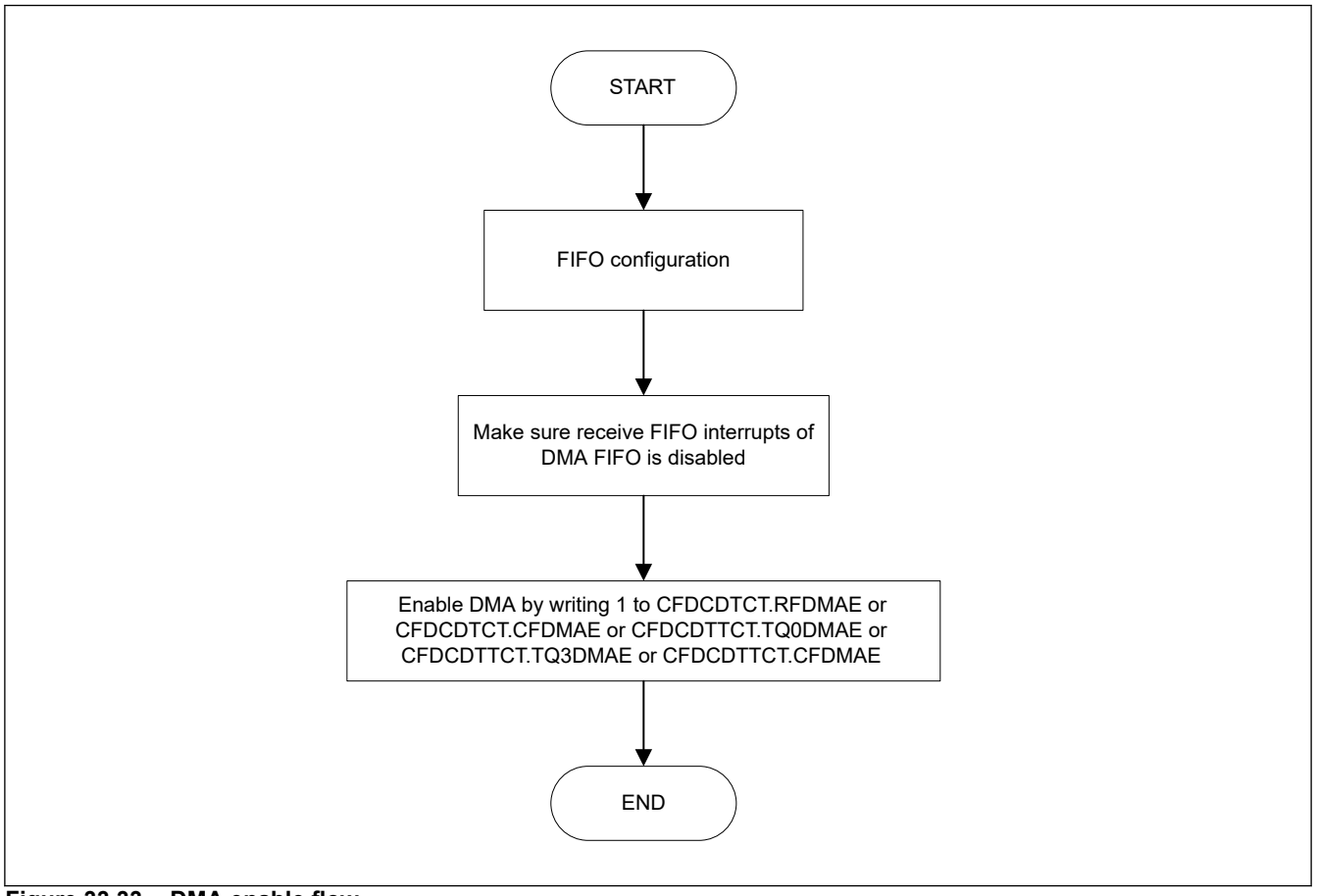


Figure 33.33 DMA enable flow

To disable a DMA transfer requested, disable the particular DMA enable bit (CFDCDTCT.RFDMAE or CFDCDTCT.CFDMAE). If the disable is made during an ongoing transfer, then this must be completed first before further action can be taken. The transfer status can be identified by the CFDCDTSTS.RFDMASTS or CFDCDTSTS.CFDMASTS. For reference, see the flow in Figure 33.34. When the DMA is disabled, then consider what to do with the remaining or new incoming messages to this particular FIFO reception.

When the FIFO is not disabled, reception to the FIFO continues.

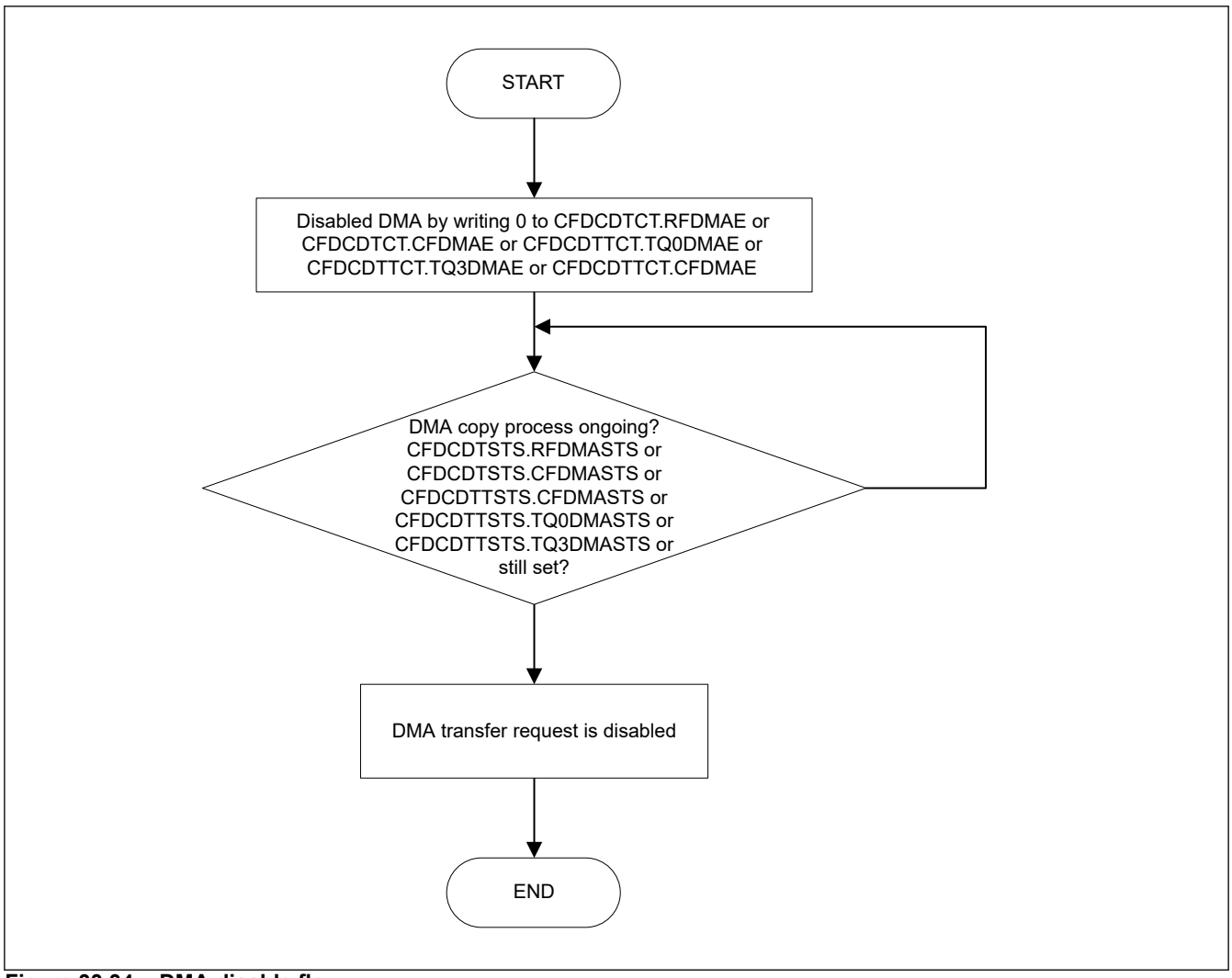


Figure 33.34 DMA disable flow

33.8 Reception and Transmission

33.8.1 Reception

FIFO buffers or Common FIFO buffers configured in RX mode or GW mode depending on the Acceptance Filter List entries are as follows:

- Up to 32 RX Message Buffers can be configured
- 8 RX FIFO buffers available
- Up to 6 Common FIFO buffers can be configured in RX mode or GW mode
- Up to 4 TX Queues can be configured in GW mode

33.8.1.1 Message Storage in RX Message Buffers

When a message is successfully received and stored in a RX Message Buffer, the corresponding New Data flag is set in the RX Message Buffer New Data Register.

The CAN message can be read from the corresponding RX Message Buffer.

If a new message is stored into a RX Message Buffer before the previous message in this message buffer can be read, then the original message is overwritten. There is no mechanism for preventing a new message from overwriting the current message in the RX Message Buffer. If such a loss of messages is not acceptable, then RX FIFO should be used to store related messages.

Note:
- Interrupts are not provided for the RX Message Buffers in the CAN-FD module and so the RX Message Buffer New Data Register should be accessed periodically to check if a new message has been stored in the RX Message Buffers.
- Unused data bytes are filled with 0x00 depending on the DLC value.

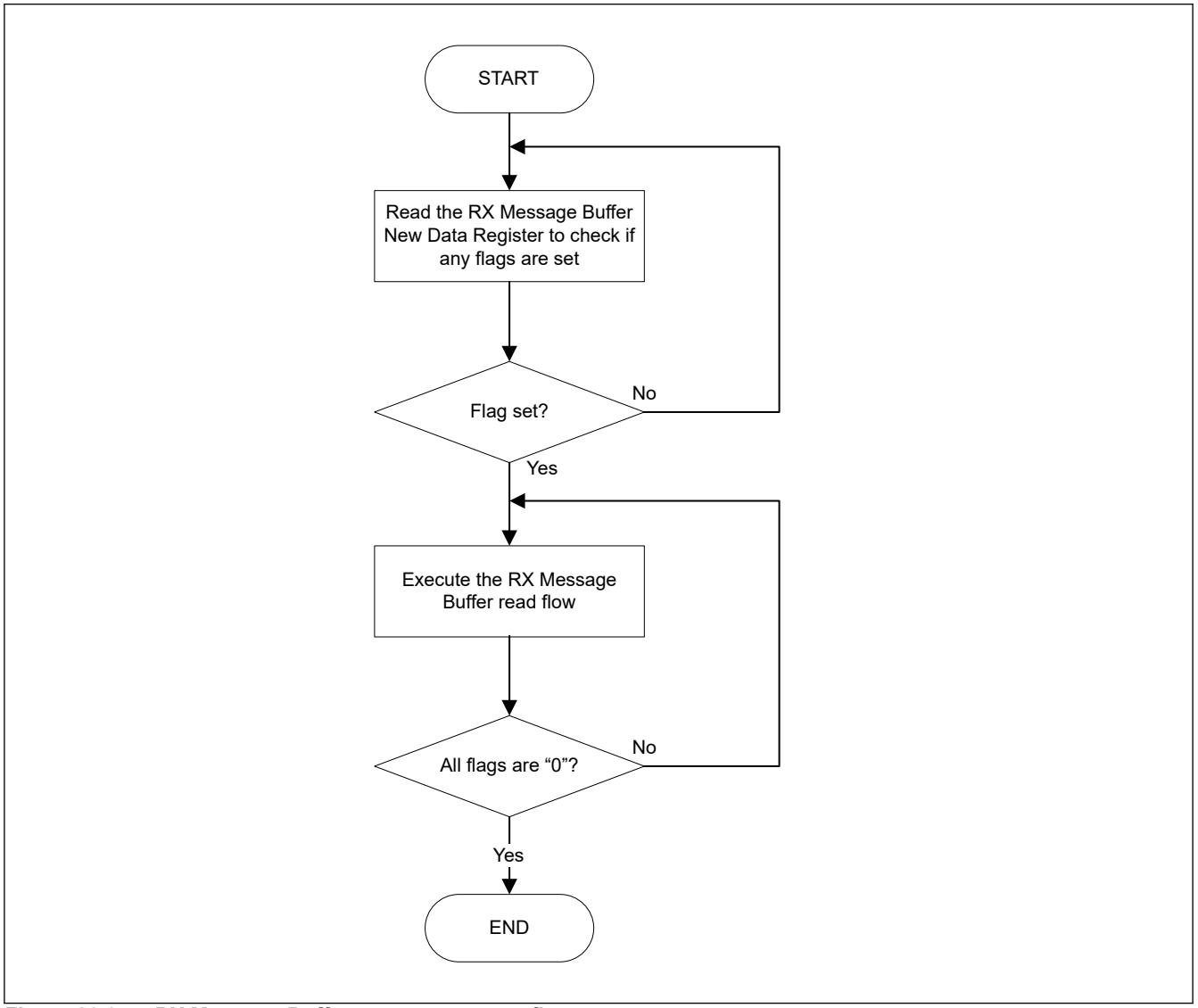


Figure 33.35 RX Message Buffer message access flow

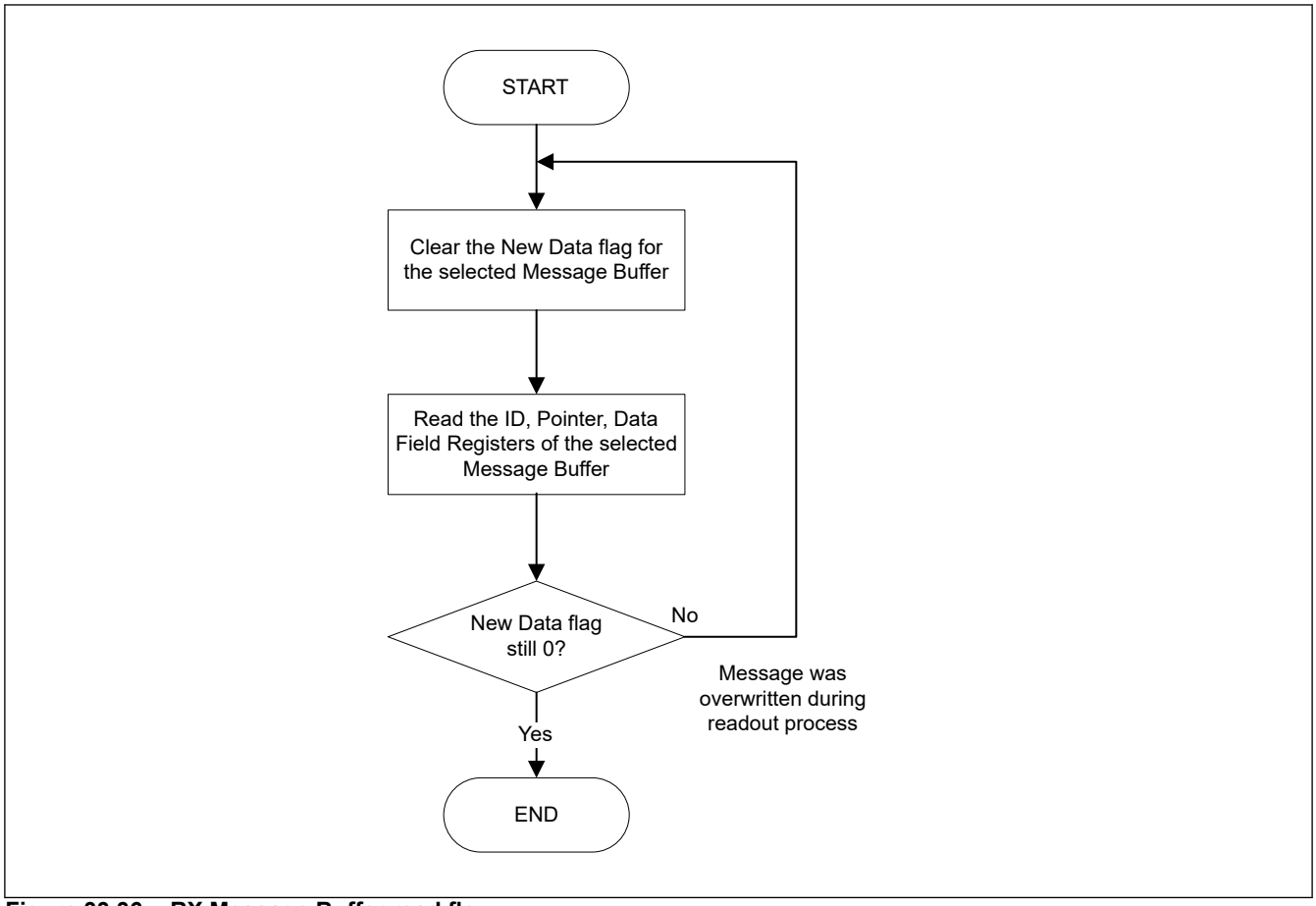


Figure 33.36 RX Message Buffer read flow

33.8.1.2 Message Storage in FIFO Buffers

The AFL entries for routing the received messages to RX FIFO buffers or Common FIFO buffers configured in RX or GW mode should be configured based on system requirements.

The CFDGAFLP1n.GAFLFDP[31:0] field in the matching AFL entry selects the FIFO buffers to which the related reception message is stored.

When the received message is stored in one or more RX FIFO buffers or Common FIFO buffers configured in RX mode or GW mode, the message counter value is incremented in the corresponding RX FIFO Status Registers or Common FIFO Status Registers.

Depending on the configuration of the FIFO buffers, an interrupt may also be generated.

The message can be read from the corresponding FIFO Access registers.

Note: Because many messages can be stored in the FIFO buffers, reading more than one message may be required to read the latest message stored in a FIFO buffer.

If the message count value matches the FIFO depth, the FIFO Full flag is set.

When the value 0xFF is written to the corresponding FIFO Pointer Control Register, the message count is decremented by 1.

Only write 0xFF to the FIFO Pointer Control register after reading the complete message from the FIFO Access registers of the corresponding FIFO.

When all the messages stored in the FIFO are read, the FIFO Empty flag is set.

If a new message is stored into the FIFO when the FIFO message count matches the FIFO depth (FIFO Full condition), the FIFO Message Lost flag is set and the new message is lost (no overwrite of already stored messages takes place).

An appropriate value can be configured as warning level to generate an interrupt before the FIFO full condition occurs to avoid loss of a message due to overrun condition.

In GW mode, when a transmit/receive FIFO buffer is trying to receive a new message while the transmit/receive FIFO buffer is already full of data, the oldest data of the buffer is overwritten with the message received or the message is discarded. The behavior is determined by setting CFDCFCCEn.CFMOWM bit.

- When CFDCFCCEn.CFMOWM = 0:
  When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the received message is discarded and CFDCFSTSn.CFMLT bit is set to 1.
- When CFDCFCCEn.CFMOWM = 1:
  When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the oldest data in the buffer is overwritten with the received message.
  The read pointer of the transmit/receive FIFO buffer simultaneously moves to the next oldest message.
  The CFDCFSTSn.CFMOW bit is then set to 1, which notifies that the oldest message has been overwritten with the received message.
  In addition, if a CAN bus error or arbitration-lost for the transmitting message occurs in the transmit/receive FIFO buffer full, the transmitting message is lost and retransmission of the message is not performed. The read point moves to the next message automatically.

Do not write change for this bit when the CFDCFCCn.CFE bit is 1.

Common FIFO can set interrupt when:

- CAN frame reception is completed
- FIFO is in full status in RX mode or GW mode

Note:
- The message lost can be set only in RX or GW mode by CAN, the flag is not set when the CPU is overloading the FIFO buffers.
- When CFDGAFLP0n.GAFLSRDi (i = 0 to 2) is set and CFDTXQCCin.TXQGWE (i = 0 to 2, n = 0, 1) is also set, a receiving frame is stored in the target TXQ as send data by routing.

The RX FIFO Buffers and the Common FIFO Buffers configured in RX or GW mode can be disabled at any time by clearing the CFDRFCCn.RFE or CFDCFCCn.CFE bit in the RX FIFO Configuration/Control Registers and the Common FIFO Configuration/Control Registers.

When the CFDRFCCn.RFE or CFDCFCCn.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Therefore, all messages in the FIFO Buffers are lost and no further messages can be stored into the FIFO.

When the RX FIFO Buffers or Common FIFO Buffers configured in RX mode are assigned as DMA channel, software should not access the FIFO Access Register of this FIFO buffer or write 0xFF to the FIFO Pointer Control Register (CFDCFPCTRn.CFPC or CFDRFPCTRn.RFPC), because this could lead to unintended FIFO message decrement. The DMA channel controls the FIFO decrement automatically.

Note: If the interrupt flag is set for a FIFO Buffer and then the FIFO is disabled, the interrupt flag is not automatically cleared. The interrupt flag should be cleared before disabling the FIFO.

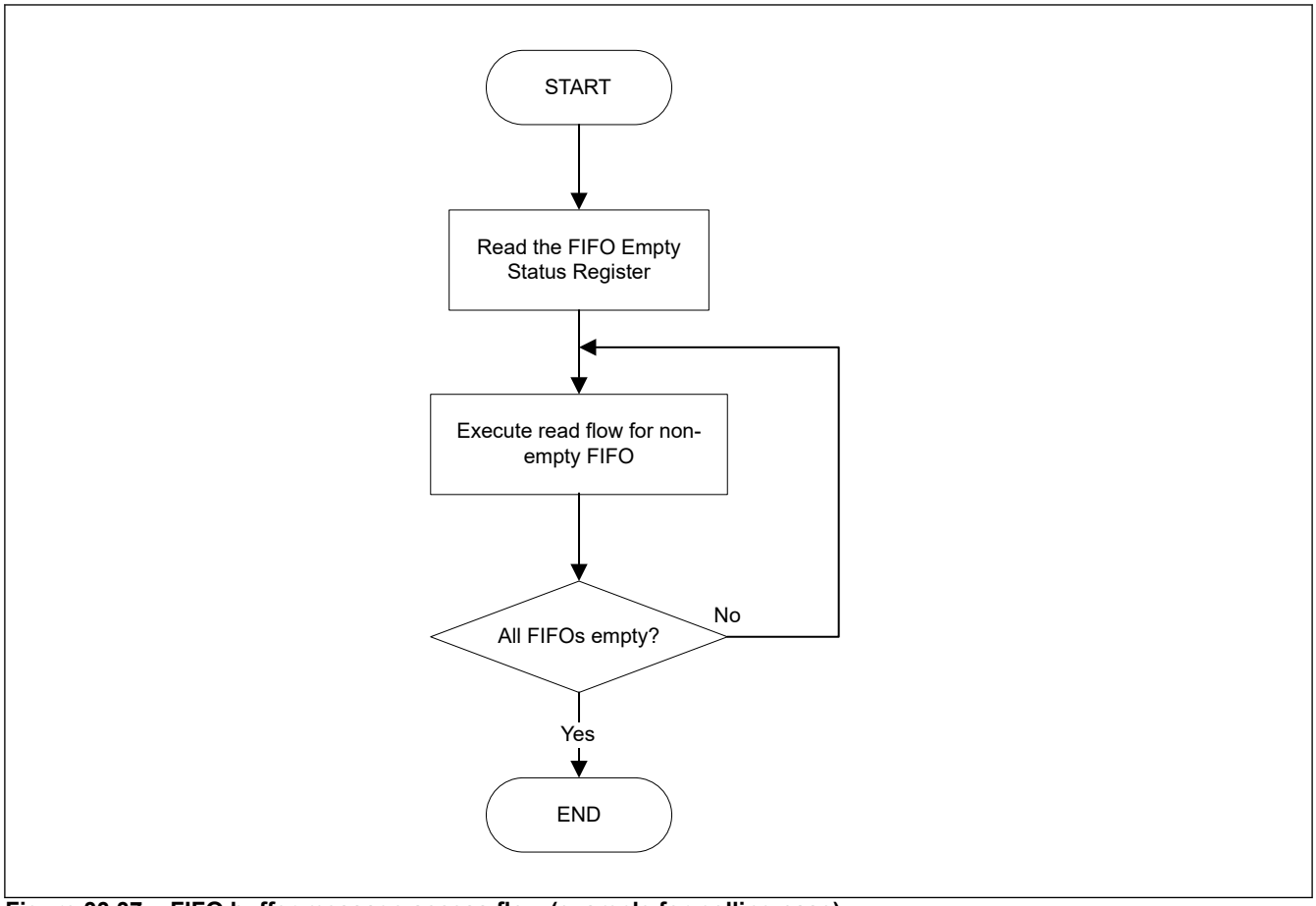


Figure 33.37 FIFO buffer message access flow (example for polling case)

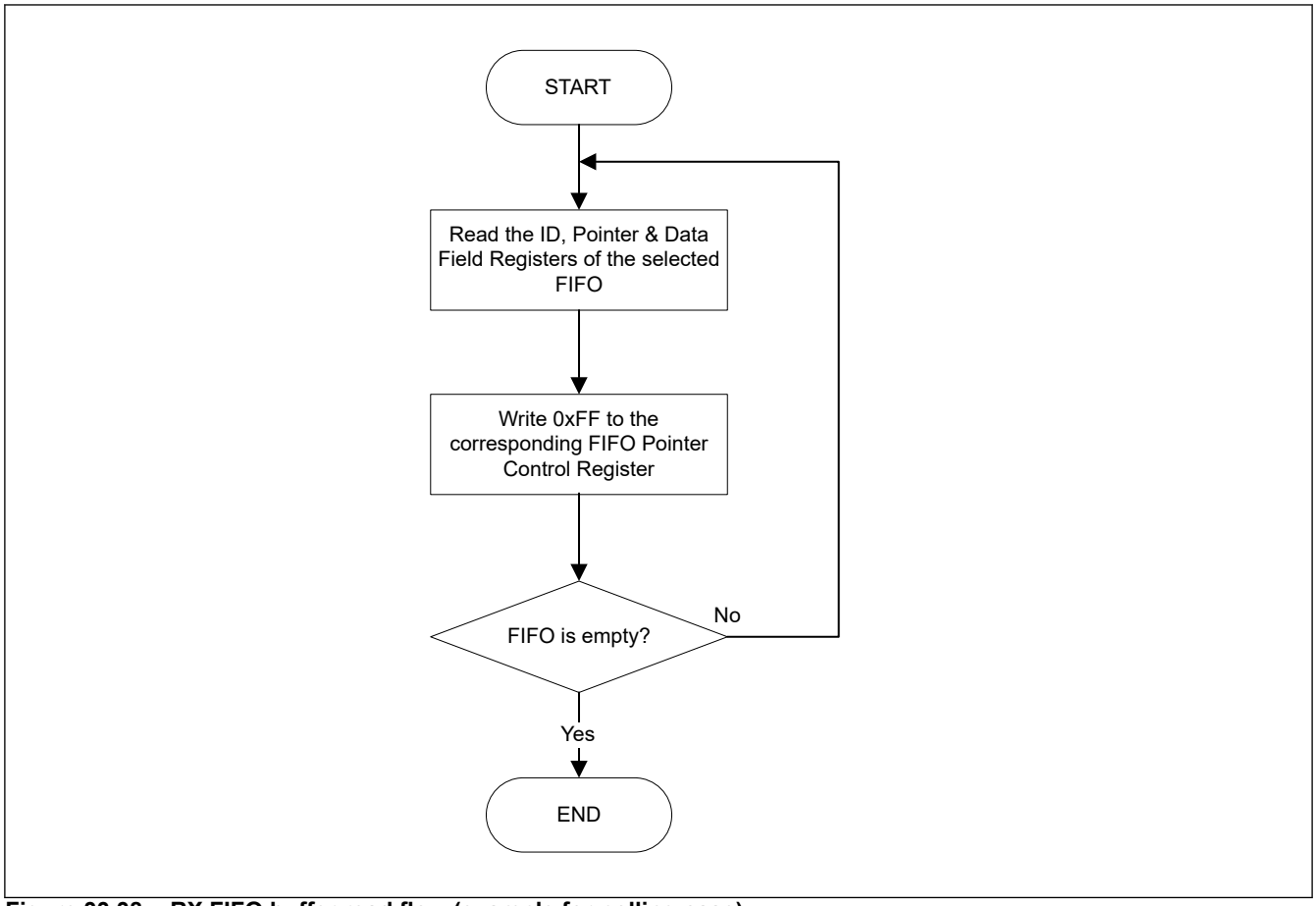


Figure 33.38 RX FIFO buffer read flow (example for polling case)

Note: When the next frame is received before clearing the completion interrupt flag of reception, the completion interrupt of reception is not set again.

Even if it clears an interruption flag after the completion processing of reception, the already received interrupt flag is not set.

It is necessary to perform the completion processing of reception even before the next completion of frame reception, and to clear an interruption flag.

When processing does not meet the deadline, after checking that receiving data is empty, interrupt flag is cleared and it checks that receiving data is empty again.

33.8.1.3 Timestamp

The Timestamp counter is a free-running counter that can be used to check reception time of an incoming message or transmission time of successful transmitted messages. The Timestamp counter value is captured based on the CFDGFDCFG.TSCCFG[1:0] configuration (at the sample point of start of frame, point in time when the frame is valid, or for CAN-FD frames also at the sample point of the RES bit). For reception, it is stored together with the message ID and Data into the target RX Message Buffer or RX/GW FIFO.

For transmit message, the Timestamp counter value is stored as part of the TX History List entry.

The counter can be clocked with the peripheral clock or with the CAN channel bit timing clock. The counter source clock can be configured with the CFDGCFG.TSSS bit of the Global Configuration Register. If it is 0, the peripheral clock is used. If it is 1, the selected CAN channel bit time clock is used.

The channel selection is done with the CFDGCFG.TSBTCS bit of the Global Configuration Register.

Care must be taken when using selected CAN channel bit time clock as the clock source. When entering Channel Halt mode or Channel Reset mode for this channel, the Timestamp counter is stopped. Therefore, for other CAN channels, the Timestamp counter value is also not updated.

If peripheral clock is selected as the Timestamp counter clock source, Channel modes do not influence the Timestamp counter function.

The source clock for the Timestamp counter can be divided by a factor defined by the CFDGCFG.TSP bits (Timestamp Prescaler) in the Global Configuration Register.

The Timestamp counter can be reset to 0x0000 with the CFDGCTR.TSRST bit (Timestamp Reset).

33.8.2 Transmission

There are several possible transmission configurations for each channel:

- Normal transmission
- FIFO transmission
- Gateway transmission
- TX Queue transmission
  A fixed number of transmission message buffers (64 TX message buffers) are dedicated for each channel. These message buffers are only used for transmission and cannot be configured for reception.
  Additionally transmission from TX Queue and/or Common FIFO in TX or GW mode can be configured in the following way (see Figure 33.39):
  - TX Queue: Up to 32 transmission message buffers for one channel can be grouped to form a TX Queue with a common access window.
    Upper transmission message buffers are used to form the TXQ1 or TXQ3.
    Lower transmission message buffers are used to form the TXQ0 or TXQ2.
    Transmission control and status registers of these transmission message buffers should not be used.
    One channel has four TX Queues. Each TXQ has an access window:
    - TXQ0 is transmission Message Buffer 0 of each channel
    - TXQ1 is transmission Message Buffer 31 of each channel
    - TXQ2 is transmission Message Buffer 32 of each channel
    - TXQ3 is transmission Message Buffer 63 of each channel

    When using TXQ1 and TXQ0 simultaneously, the sum of the depths of TXQ1 and TXQ0 should not exceed 32.
    When using TXQ3 and TXQ2 simultaneously, the sum of the depths of TXQ3 and TXQ2 should not exceed 32.
- Common FIFO (TX/GW mode): Each Common FIFO in TX or GW mode is linked to a dedicated channel.
  Each channel has a fixed number of three Common FIFOs assigned to it. Within the channel, a Common FIFO configured in TX or GW mode, can be freely linked (assigned) between 32 and 39 transmission message buffers (only one FIFO to one transmission message buffer).
  The Common FIFO buffer then replaces the transmission message buffer linked to it.
  Transmission control and status registers of these transmission message buffers should not be used.

See Figure 33.28 for information on Common FIFO buffer assignment to related channels.

Note: Common FIFO buffers should not be linked to TX message buffers that are already part of a TX Queue.

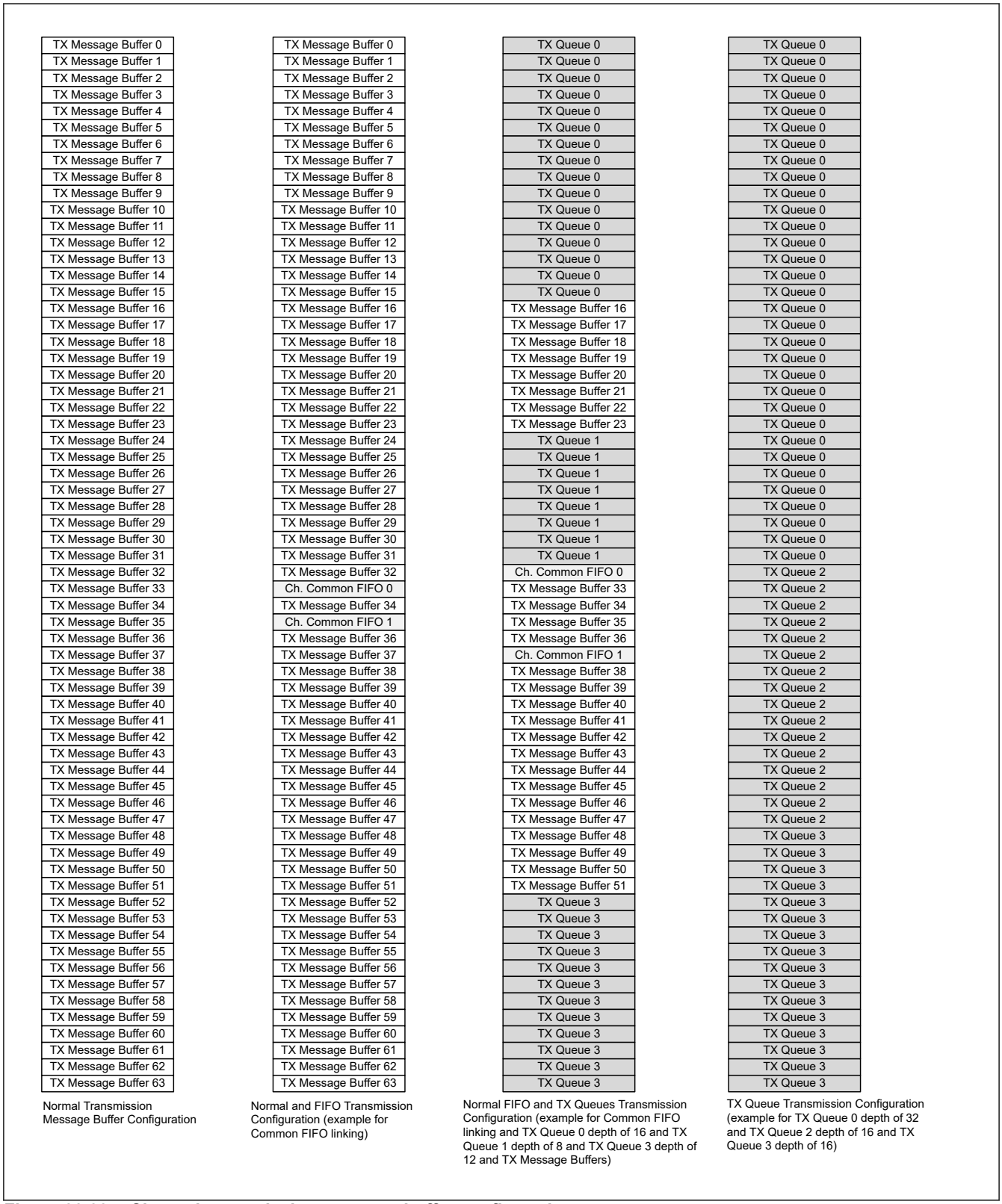


Figure 33.39 Channel transmission message buffer configuration

33.8.2.1 Transmission Priority

If two or more transmission message buffers of a channel are configured for transmission, the transmission priority in the CAN-FD module can be selected from the following two modes:

- CAN ID priority

- Message buffer number priority

The transmission priority mode is common for all message buffers and all CAN channels. It can be configured with the CFDGCFG.TPRI bit in the Global Configuration Register.

For message buffer number priority transmission, the smallest message buffer number with transmission request has the highest priority for transmission. This also includes the TX message buffers linked to the Common FIFO buffers configured in TX mode or GW mode.

However, message buffer number priority should not be used if TX Queue is enabled.

For CAN ID priority transmission, ID priority complies with the CAN bus arbitration rule (as specified in ISO 11898-1 specification). All TX message buffers can enter the ID priority comparison for message buffers configured for transmission. This also includes the TX message buffers linked to the Common FIFO buffers configured in TX mode or GW mode and includes the TX Queue message buffers.

If the ID of two or more message buffers is the same, the smaller message buffer number has higher priority for transmission.

Note: For Common FIFO buffers configured in TX mode or GW mode, only the message currently being pointed to by the FIFO read pointer can be included in the transmission arbitration.

If the message is being transmitted from the FIFO, the next pending message within the same FIFO will be considered in the transmission arbitration.

In contrast to this, all transmission message buffers of a TX Queue participate in internal transmission arbitration.

Figure 33.40 shows the transmission configuration flow.

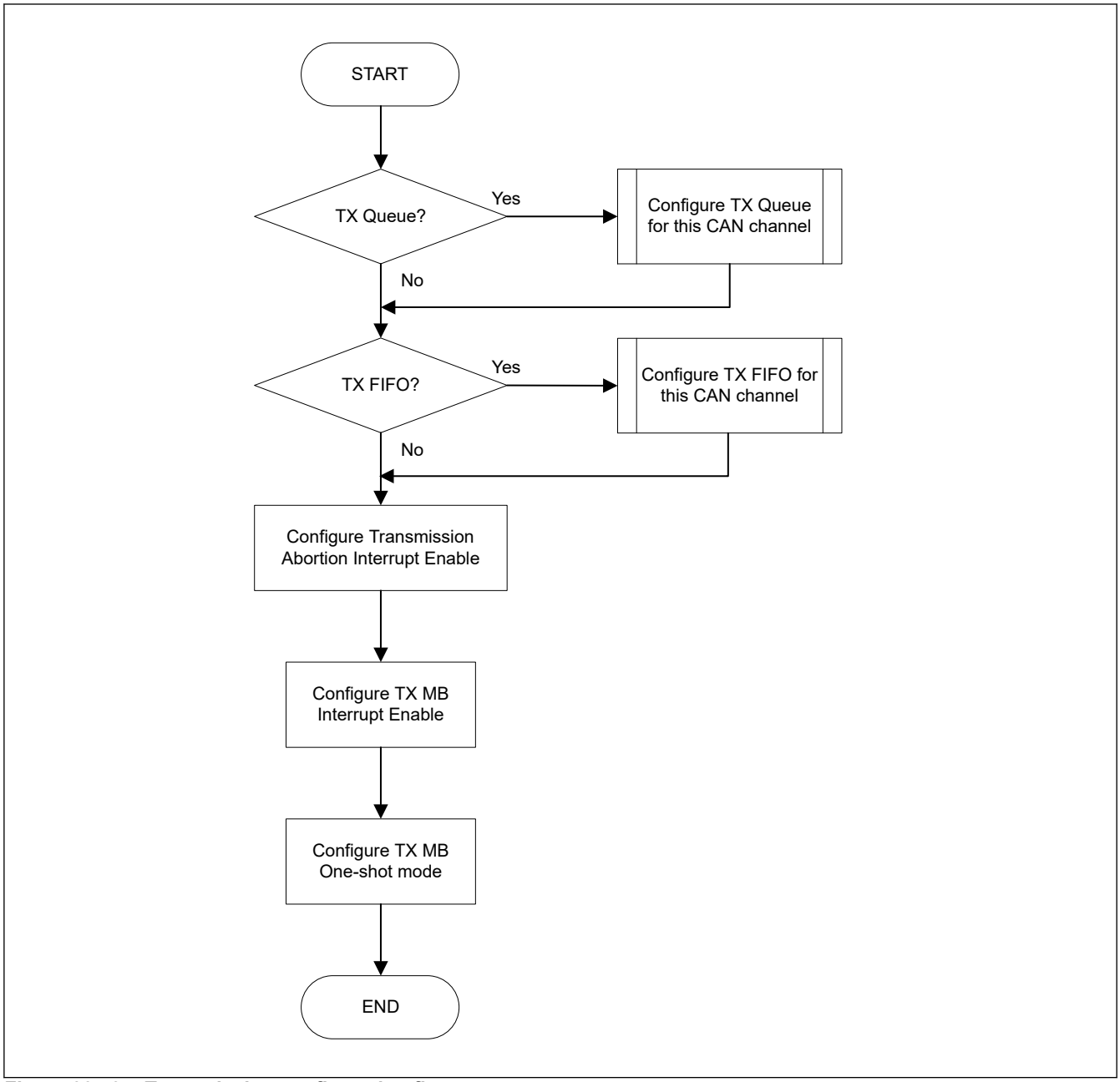


Figure 33.40 Transmission configuration flow

33.8.2.2 Normal Transmission

Each transmission message buffer has two modes of message transmission:

- Regular transmission mode
  If the message buffer is placed in regular transmission mode, the data frame or remote frame set in that message buffer can be transmitted.
  Completion of regular transmission can be checked through the related TX Message Buffer Transmission Result flag bits (CFDTMSTSn.TMTRF[1:0]) in the TX Message Buffer Status Registers. These bits are set to 10b or 11b when the regular transmission is successful.
  When arbitration is lost or an error occurs, message transmission will be attempted further if no transmission abort request is set for this transmission message buffer.
  New internal transmission arbitration for this channel is performed for all message buffers with transmission request.
- One-shot transmission mode
  When the CFDTMCn.TMOM bit of the TX Message Buffer Control Registers is set for a transmission message buffer, then the message buffer is placed in one-shot transmission mode and attempts to transmit a message only once.

Completion of one-shot transmission can be checked through the related TX Message Buffer Transmission Result flag bits (CFDTMSTSn.TMTRF[1:0]) in the TX Message Buffer Status Registers. The CFDTMSTSn.TMTRF[1:0] bits are set to 10b or 11b when the one-shot transmission is successful.
The CFDTMSTSn.TMTRF[1:0] bits are set to 01b when arbitration is lost or an error occurs during the transmission of the related message buffer.
Additional message transmission will not be attempted in this case.

The regular transmission request procedure after a configuration is shown in Figure 33.41.

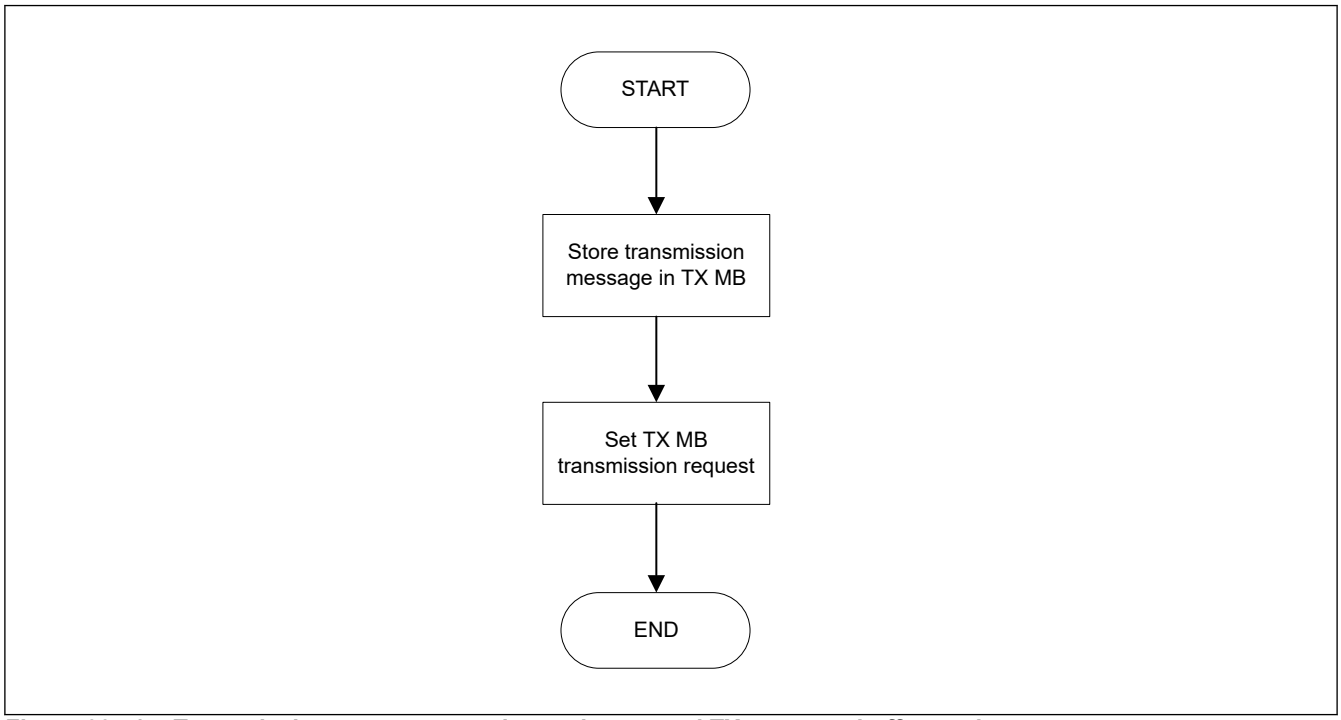


Figure 33.41 Transmission request procedure using normal TX message buffer mode

(1) TX Message Buffer Control Register Setting

Table 33.26 shows configuration of the normal CAN transmission mode.

Table 33.26 Configuration of CAN transmission mode

Transmission request CFDTMCn.TMTR	Transmission abortion request CFDTMCn.TMTAR	One Shot Enable CFDTMCn.TMOM	Communication activity
0	0	0	Message buffer is disabled
0	0	1	Message buffer is disabled
1	0	0	Configured as a transmission message buffer for a data frame or a remote frame
1	0	1	Configured as a one shot transmission message buffer for a data frame or a remote frame
1	1	0	Transmission abortion is requested
1	1	1	One shot transmission abortion is requested

The configuration bits can be configured in the TX Message Buffer Control Registers.

Figure 33.42 shows timings for successful transmission for two message buffers of one channel.

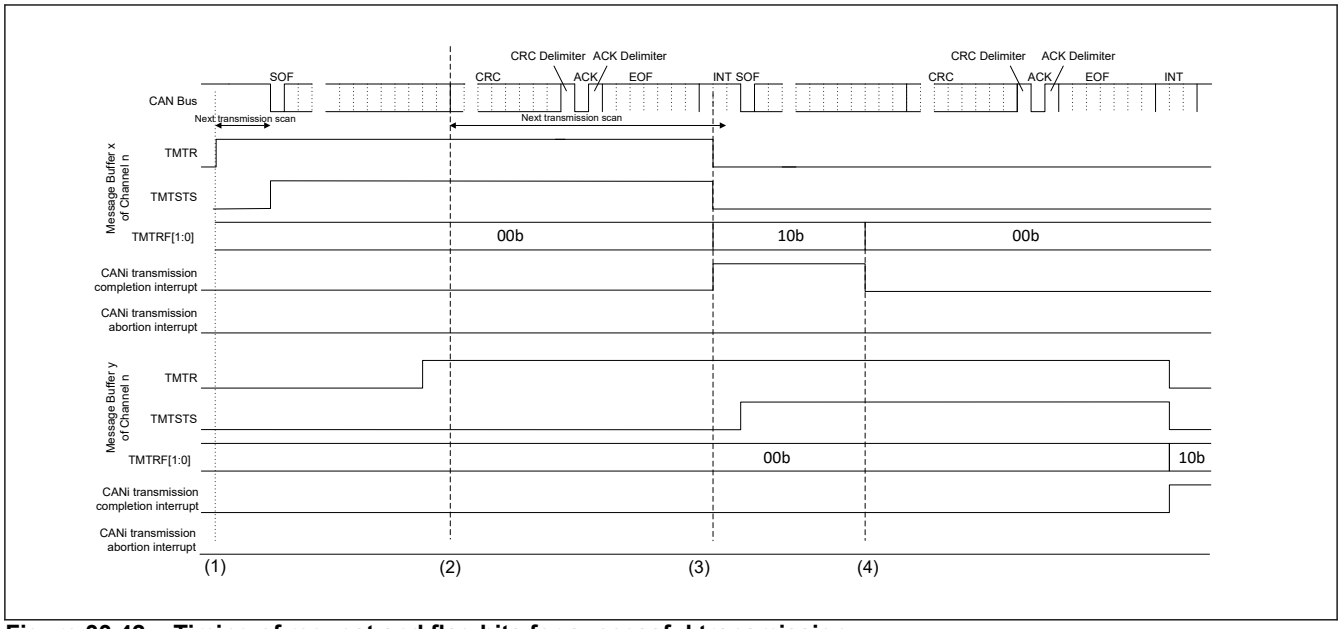


Figure 33.42 Timing of request and flag bits for successful transmission

1. If the CFDTMCn.TMTR bit in the TX Message Buffer Control Registers is set in the bus idle state, message buffer scanning procedure starts to determine the highest priority message buffer for transmission.
   When the transmission message buffer is determined, the CFDTMSTSn.TMTSTS bit in the related TX Message Buffer Status Registers is set (Transmitting/Transmitter), and CAN channel starts the transmission[*1].
2. On the first bit of CRC, the transmission scanning procedure starts for the next possible transmission when pending transmission requests exist.
   The scan time can be delayed due to other transmission scan on other channels, but it will be finished before Intermission 3 to be able to continue transmission without any gaps.
3. If the message is successfully transmitted, the CFDTMSTSn.TMTRF[1:0] bits in the corresponding TX Message Buffer Status Registers are set to 10b and CFDTMSTSn.TMTSTS and the CFDTMCn.TMTR bits are cleared.
   When the TMIE bits in the TX Message Buffer Interrupt Enable Configuration Registers is set (interrupt enabled), the CAN successful transmission interrupt request is generated.
   To clear the related interrupt line the CFDTMSTSn.TMTRF flag bits must be cleared.
4. Before starting the next transmission, clear the CFDTMSTSn.TMTRF[1:0] bits. Load the next message in the transmission message buffer and set the CFDTMCn.TMTR bit again. CFDTMCn.TMTR bit cannot be set again before CFDTMSTSn.TMTRF[1:0] bits are cleared.

The transmission scanning procedure is performed again to search for the highest priority transmission message buffer from the beginning of the first CRC bit.

If an error occurs either during the transmission or following the loss of arbitration, then during Error Frame, the transmission scanning procedure is performed again to search for the highest priority transmission message buffer.

Note: The setting point of CFDTMSTSn.TMTSTS is not always fixed at the start of the SOF. It may be delayed up to the start of the standard ID due to the synchronization logic implemented for the PLL bypass.

Figure 33.43 shows timings for transmission abort for two message buffers of one channel.

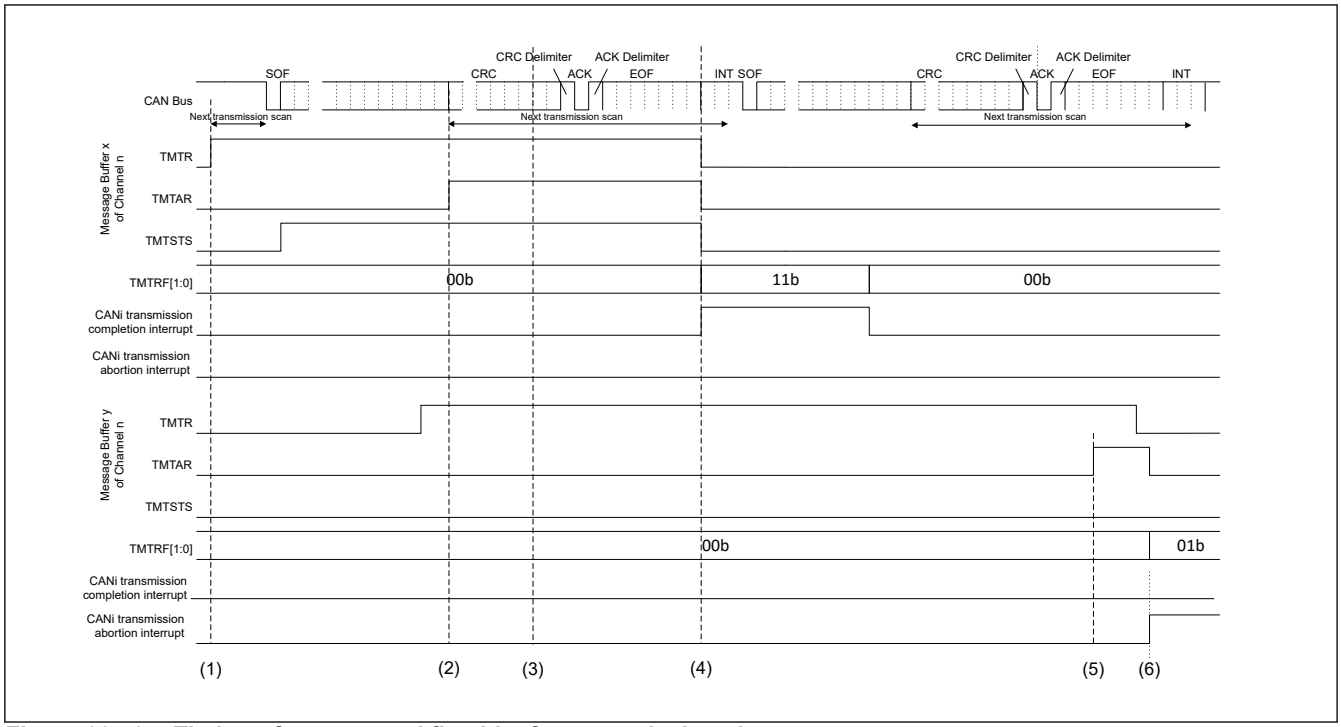


Figure 33.43 Timing of request and flag bits for transmission abort

Note 1. If arbitration is lost after the CAN channel starts the transmission, the CFDTMSTSn.TMTSTS bit is cleared.

1. If the CFDTMCn.TMTR bit in the TX Message Buffer Control Registers is set in the bus idle state, message buffer scanning procedure starts to determine the highest priority message buffer for transmission.
   When the transmission message buffer is determined, the CFDTMSTSn.TMTSTS bit in the TX Message Buffer Status Registers is set (Transmitting/Transmitter), and CAN channel starts the transmission[*1].
2. If the CFDTMCn.TMTAR bit is set when the related message buffer is already selected for transmission or currently transmitting, the message is not aborted, if no error occurs or arbitration is lost.
3. On the first CRC bit, the transmission scanning procedure starts for the next transmission. In this example timing diagram, message buffer y is not selected as the next transmission message buffer.
   The scan time can be delayed due to other transmission scan on other channels, but it will be finished before Intermission 3 to be able to continue transmission without any gaps.
4. If the message is successfully transmitted, the CFDTMSTSn.TMTRF[1:0] bits in the corresponding TX Message Buffer Status Registers are set to 11b and CFDTMSTSn.TMTSTS and the CFDTMCn.TMTR bits are cleared.
   When the TMIE bits in the TX Message Buffer Interrupt Enable Configuration Registers is set (interrupt enabled), the CAN successful transmission interrupt request is generated.
   To clear the related interrupt line, clear the CFDTMSTSn.TMTRF[1:0] bits.
5. Another CAN node is transmitting on the CAN bus (CFDTMSTSn.TMTSTS not set). If the CFDTMCn.TMTAR bit is set when the related channel is under transmission scan, the transmission request cannot be cleared.
6. After internal processing time, the transmission is aborted and the CFDTMSTSn.TMTRF[1:0] bits are set to 01b. If the message buffer is not transmitting or selected as the next transmission message buffer or under transmit scan, then the abort is immediately accepted and the corresponding CFDTMSTSn.TMTRF[1:0] bits in the TX Message Buffer Status Registers are set to 01b.
   In addition, CFDTMCn.TMTR, and CFDTMCn.TMTAR bits are cleared automatically.
   When the transmission abort interrupt enable TAIE bit of the related Channel Control Register is set, an interrupt is generated for successful transmission abort.
   To clear the related interrupt line, clear the CFDTMSTSn.TMTRF[1:0] bits.

Note 1. If arbitration is lost after the CAN channel starts the transmission, the CFDTMSTSn.TMTSTS is cleared. The transmission scanning procedure is performed again to search for the highest priority transmission message buffer from the beginning of the first CRC bit. If an error occurs, either during the transmission, or following the loss of arbitration, then during Error Frame, the transmission scanning procedure is performed again to search for the highest priority transmission Message Buffer.

33.8.2.3 TX FIFO or GW FIFO Transmission

Three common FIFO buffers are assigned to each channel. The three FIFO buffers can be linked to any normal TX Message Buffer position for this channel by the CFDCFCCn.CFTML[4:0] bits in the Common FIFO Configuration/Control Register if configured in TX or GW mode.

When the transmission scan starts and the FIFO buffer corresponding to this TX Message Buffer is enabled, the relevant message in the FIFO buffer participates in the transmission scan.

Configuration of a TX Message Buffer linked to a FIFO buffer configured in TX or GW mode should not be done.

(1) TX FIFO Operation

CAN messages can be written into the TX FIFO by writing to the corresponding FIFO Access registers.

When the value 0xFF is written into the corresponding FIFO Pointer Control Register, the message count of the related FIFO is incremented by 1.

Only write to the FIFO Pointer Control register after writing the complete message to the corresponding FIFO Access registers.

If the message count matches the FIFO Depth, the FIFO Full flag is set.

The oldest message in the TX FIFO is included in the scan for transmission by the corresponding CAN-FD module channel logic.

When a message is successfully transmitted from the TX FIFO, the message count value is decremented by 1. When all the messages from the FIFO are transmitted, the FIFO Empty flag is set.

The interrupt generation conditions for the TX FIFO buffers can be configured by configuring the CFDCFCCn.CFIM bit in the corresponding Common FIFO Configuration/Control Registers.

If CFDCFCCn.CFIM bit is 0, an interrupt is generated when the last message is successfully transmitted from the TX FIFO buffer.

If CFDCFCCn.CFIM bit is 1, an interrupt is generated for every successfully transmitted message from the TX FIFO buffer.

Common FIFO can set interrupt, when CAN frame transmitted is completed.

The Common FIFO buffers configured in TX mode can be disabled by clearing the CFDCFCCn.CFE bit in the Common FIFO Configuration/Control Registers. If this bit is cleared to 0, the FIFO Empty flag is set as follows:

- Immediately if the message from the TX FIFO is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Halt mode if the transmission from the TX FIFO is already scheduled for transmission or already in transmission.

Note: The Common FIFO buffer is considered as disabled after clearing the CFDCFCCn.CFE bit only when the Empty flag is set for the corresponding Common FIFO Buffer.

Other possible messages pending from the TX FIFO are lost and their transmission must be requested again. Before CFDCFCCn.CFE is set again, ensure that CFDCFSTSn.CFEMP bit is set and that there is no pending abort from the TX FIFO.

When the CFDCFCCn.CFE bit is cleared, the message read and write pointers of the FIFO are cleared and are no longer active. Therefore, all messages in the FIFO buffers are lost and no further message can be stored into the FIFO.

The FIFO transmission request procedure after configuration is shown in Figure 33.44.

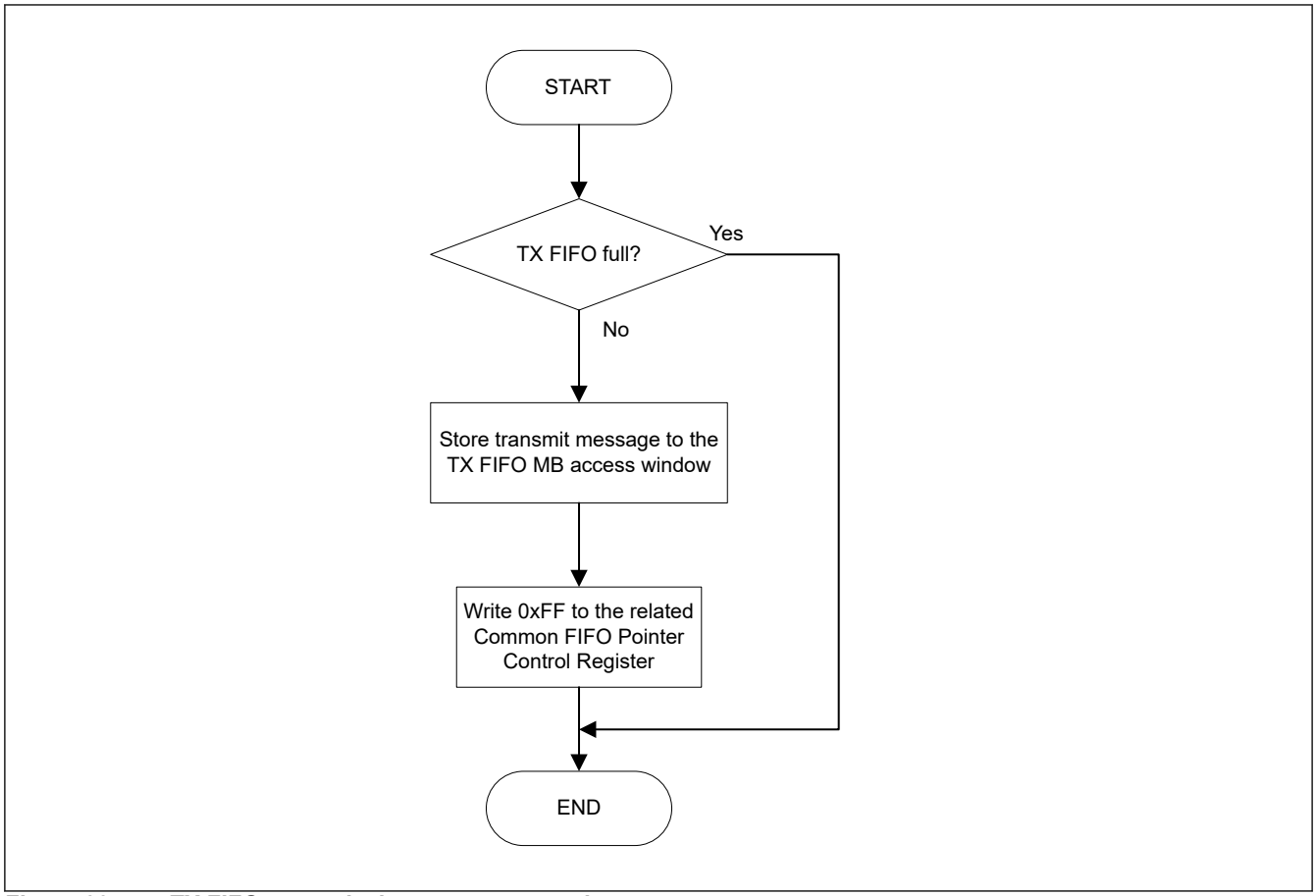


Figure 33.44 TX FIFO transmission request procedure

(2) GW FIFO Operation

The AFL entries for routing the received messages to GW FIFO buffers should be configured based on the requirements of the system. The matching AFL entry selects the GW FIFO buffer for storage of a received message on any of the CAN channels.

When a message is successfully received and stored in a GW FIFO buffer, the FIFO message count in the corresponding FIFO Status Register is incremented by 1.

If the message count matches the FIFO depth, then the FIFO Full flag is set.

The oldest message in the GW FIFO is included in the scan for transmission by the corresponding CAN-FD module channel logic.

When a message is successfully transmitted from the GW FIFO, the message count value is decremented by 1. When all the messages from the GW FIFO are transmitted, the FIFO Empty flag is set.

If a new message is stored into the FIFO when the FIFO message count matches the FIFO depth (FIFO full condition), the FIFO Message Lost flag is set and the new message is lost (no overwrite of already stored messages takes place).

In GW mode, when a transmit/receive FIFO buffer is trying to receive a new message while the transmit/receive FIFO buffer is already full of data, the oldest data of the buffer is overwritten with the message received or the message is discarded. The behavior is determined by setting CFDCFCCEn.CFMOWM bit.

- When CFDCFCCEn.CFMOWM = 0:
  When writing of data is required due to reception of a new message while a transmit/receive FIFO buffer is full of data, the received message is discarded and CFDCFSTSn.CFMLT bit is set to 1.
- When CFDCFCCEn.CFMOWM = 1:
  When writing of data is required due to reception of a new message while a transmit/receive.
  FIFO buffer is full of data, the oldest data in the buffer is overwritten with the received message.
  The read pointer of the transmit/receive FIFO buffer simultaneously moves to the next oldest message.

The CFDCFSTSn.CFMOW bit is then set to 1, which notifies that the oldest message is overwritten with the received message.
In addition, when a CAN bus error or arbitration-lost for the transmitting message occurs in transmit/receive FIFO buffer full, the transmitting message is lost and retransmission for the message is not performed. The read point moves to the next message automatically.

The interrupt generation conditions for the GW FIFO buffers can be configured by configuring the CFDCFCCn.CFIM bit in the corresponding Common FIFO Configuration/Control Registers.

If CFDCFCCn.CFIM bit is 0, then RX interrupt flag is set when FIFO counter increments and reaches value configured by CFDCFCCn.CFIGCV and the TX interrupt flag is set when FIFO transmits the last message successfully.

If CFDCFCCn.CFIM bit is 1, then RX interrupt flag is set at the end of storage of every received message and TX interrupt flag is set if a message is successfully transmitted from the FIFO.

Common FIFO can set interrupt when:

- CAN frame transmitted is completed
- CAN frame reception is completed
- FIFO is in full status in RX mode or GW mode

When CFDCFCCEn.CFBME = 1, it becomes FIFO buffering mode, send data is stored in Common FIFO, and transmission is stopped. Transmission will be started if it is set as CFDCFCCEn.CFBME = 0.

The Common FIFO buffers configured in GW mode can be disabled by clearing the CFDCFCCn.CFE bit in the Common FIFO Configuration/Control Register. If this bit is cleared, the GW FIFO becomes empty as follows:

- Immediately if the message from the GW FIFO is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Halt mode if the transmission from the GW FIFO is already scheduled for transmission or already in transmission

Other possible messages pending from the GW FIFO are lost.

Before CFDCFCCn.CFE is set again, ensure that the CFDCFSTSn.CFEMP bit is set and that there is no pending abort from the GW FIFO.

When the CFDCFCCn.CFE bit is cleared and the CFDCFSTSn.CFEMP bit is set, the message read and write pointers of the GW FIFO are cleared and are no longer active. Therefore, all messages in the GW FIFO buffers are lost and no further message can be stored into the GW FIFO.

In applications intended to be used as CAN-to-CAN gateways, it is useful if the Error State Indication (ESI) information of the routing messages is not replaced by the sending node Error State Indication. For this, each channel has the control function register CFDCnFDCFG.ESIC to replace their own ESI information by the routing ESI information.

Note: If the sending node is error passive, the ESI bit is sent anyway as error passive (ESI = 1).

(3) Interval Timer for FIFO Transmission

For each Common FIFO in TX or GW mode, it is possible to specify a delay between two consecutive messages that are configured for transmission from the same FIFO buffer. This delay is called interval time. This interval time starts after the first message has been successfully transmitted from the FIFO buffer after the CFDCFCCn.CFE bit is set.

When the Common FIFO in TX or GW mode is enabled, the first message is transmitted without considering this interval time.

The interval timer stops counting when:

- FIFO is disabled by clearing the CFDCFCCn.CFE bit
- CAN channel is in CH_RESET mode

The interval time is specified by the CFDCFCCn.CFITT value in the Common FIFO Configuration/Control Register and can be specified from 0 to 255 timer units.

The timer unit can be defined based on two different source clocks for the interval timer. To disable the interval timer for FIFO transmission, a value of 0 should be selected.

The timer source can be selected by the configuration bit CFITSS in the Common FIFO Configuration/Control Register. For the timer source the CAN bit timing clock of the FIFO related channel or a global reference clock can be selected.

If CAN channel bit time clock is configured as clock source and the CAN channel enters CH_HALT or CH_RESET or CH_SLEEP mode, the interval timer is stopped for that channel.

If peripheral clock is selected as interval timer clock source, then the interval timer is stopped only when the CAN channel is in CH_RESET or CH_SLEEP mode.

The reference clock can be used to configure the interval time in fixed time units. It is based on the peripheral clock. The reference clock prescaler value CFDGCFG.ITRCP in the Global Configuration Register defines the relation between the peripheral clock frequency/period and the reference clock period.

See Table 33.27 for CFDGCFG.ITRCP configuration values to achieve different reference clock periods based on the peripheral clock frequency/period.

Table 33.27 Configuration example for the FIFO interval timer reference clock

Peripheral clock (PCLKM)	Reference clock		
	1 µs	100 µs	500 µs
75 MHz / 13.33 ns	75	7500	37500
100 MHz / 10 ns	100	10000	50000

Additionally, the reference clock resolution can be specified by the interval timer reference clock resolution value CFDCFCCn.CFITR in the Common FIFO Configuration/Control Register.

The interval time is based on the reference clock period multiplied by the configured value (× 1 or × 10).

The reference clock based interval timer can be used to follow the requirements of the ISO 15765-2 Separation Time. The whole range for the separation time from 100 µs to 127 ms can be covered.

The specified interval time starts after successful transmission event (after EOF7 state of the CAN protocol).

When the interval time has elapsed, the next transmission request is raised by the related TX/GW FIFO. Therefore, the interval time defines the minimum time between two messages transmitted from one FIFO.

The next message is sent at earliest after this interval time.

Figure 33.45 shows an example timing of the internal processing.

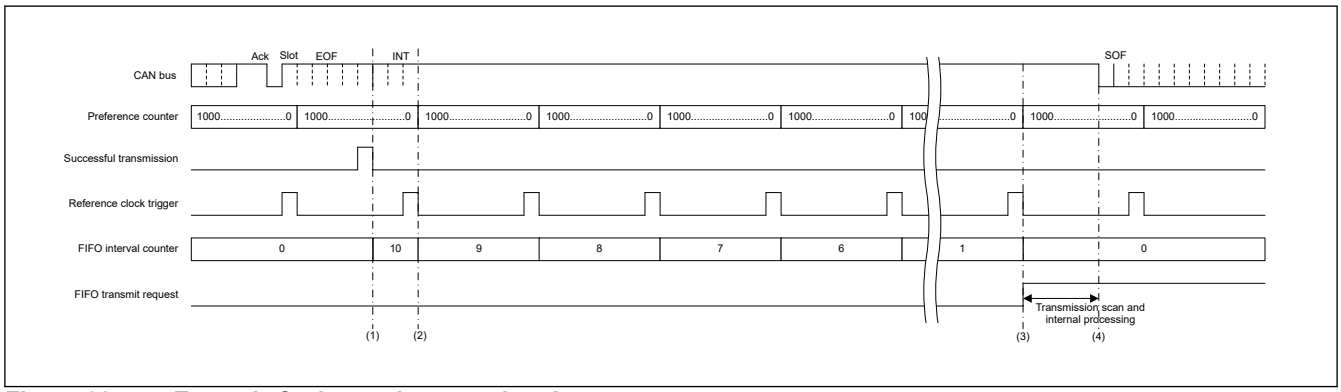


Figure 33.45 Example for interval processing time

The configuration for the above timing is as follows:

- Peripheral clock frequency = 100 MHz
- Interval timer reference clock (CFDGCFG.ITRCP) = 1000 times
- Reference clock due to the above settings = 10 µs
- Common FIFO interval timer source selection (CFDCFCCn.CFITSS) = 0
- Common FIFO interval timer resolution (CFDCFCCn.CFITR) = 0
- Common FIFO interval transmission time (CFDCFCCn.CFITT) = 10 times
- Theoretical message separation interval = 100 µs

1. Internal FIFO interval timer is restarted with the occurrence of successful transmission result. This restart is not synchronized to the reference clock trigger. Therefore, the first interval is counting less or equal to one reference clock interval.
2. With the next reference clock trigger, the FIFO interval timer is decremented.
3. When the FIFO interval timer reached the value 0 the FIFO transmit request is set.
4. When the FIFO is selected for transmission, the transmission starts. Due to internal processing this usually takes less than 3 CAN bit time, between internal FIFO transmit request set 3 and actual transmission.

In the worst case when multi events such as reception scan, internal message routing, transmit scan on all channels occur, then it can take up to 432 peripheral clock cycles.

As shown in Figure 33.45, it is not guaranteed that the minimum interval is always equal to the configured value. If a minimum time must never be breached, configure CFDCFCCn.CFITT to the required minimum value + 1.

If additional TX message buffers or TX/GW FIFOs are configured for transmission for the same channel, the real delay between two messages transmitted from a TX FIFO can be much longer than specified by the interval time due to higher priority message transmission from these TX message buffers or TX/GW FIFOs.

Figure 33.46 shows a block diagram of the FIFO interval time generation circuit.

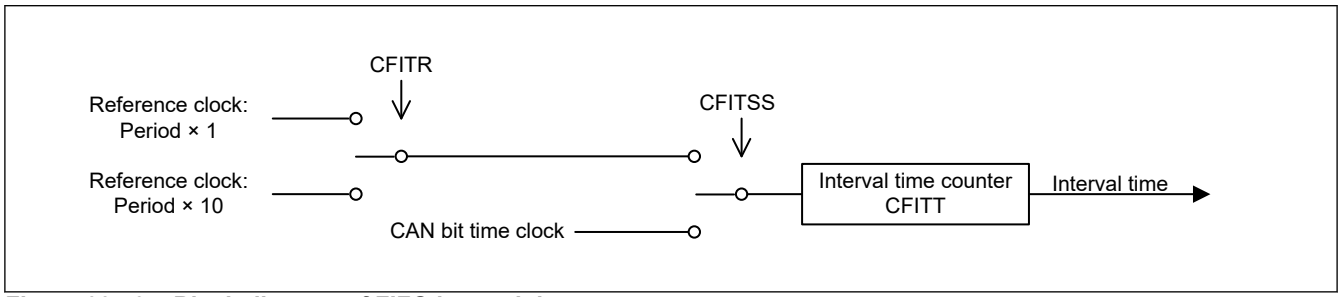


Figure 33.46 Block diagram of FIFO interval timer

33.8.2.4 TX Queue

Each enabled TX Queue for a specific channel consists of 3 to 32 TX message buffers, which are accessed by one access window. One channel has 4 TX Queues. One TX Queue can be configured with a depth of 3 up to 32 buffers and it uses TX Message Buffer No. 0 as access window (referred to as TXQ0). The second TX Queue can be configured with a depth of 3 up to 32 buffers and it uses TX Message Buffer No. 31 as access window (referred to as TXQ1). The third TX Queue can be configured with a depth of 3 up to 32 buffers and it uses TX Message Buffer No. 32 as access window (referred to as TXQ2). The fourth TX Queue can be configured with a depth of 3 up to 32 buffers and it uses TX Message Buffer No. 63 as access window (referred to as TXQ3).

All TXQ0, TXQ1, TXQ2, and TXQ3 messages enter the priority comparison for the transmission, which should be only ID Priority (CFDGCFG.TPRI = 0).

The registers for TXQ0 are CFDTXQCC0n, CFDTXQSTS0n, and CFDTXQPCTR0n.

The registers for TXQ1 are CFDTXQCC1n, CFDTXQSTS1n, and CFDTXQPCTR1n.

The registers for TXQ2 are CFDTXQCC2n, CFDTXQSTS2n, and CFDTXQPCTR2n.

The registers for TXQ3 are CFDTXQCC3n, CFDTXQSTS3n, and CFDTXQPCTR3n.

When access window TX Message Buffer No.63 (TXQ3) or TX Message Buffer No.32 (TXQ2) or TX Message Buffer No.31 (TXQ1) or TX Message Buffer No.0 (TXQ0) is used, refer to the related access registers TX Message Buffer ID Registers (CFDTMIDn), TX Message Buffer Pointer Registers (CFDTMPTRn), TX Message Buffer Data Field 0 Registers (CFDTMDF0_n), and TX Message Buffer Data Field 1 Registers (CFDTMDF1_n).

The depth of each TXQ0 buffer can be configured by writing to the CFDTXQCC0n.TXQDC[4:0] bits of the TX Queue Configuration/Control Register.

TXQ0 can set from TXMB0 to TXMB31 as a queue buffer at the maximum.

The 31 available options for depth configuration are:

- 0x00: TX Queue disabled

- 0x01: Reserved
- 0x02: 3 messages
- ⋮
- 0x1D: 30 messages
- 0x1E: 31 messages
- 0x1F: 32 messages

The depth of each TXQ1 buffer can be configured by writing to the CFDTXQCC1n.TXQDC[4:0] bits of the TX Queue Configuration/Control Register.

TXQ1 can set from TXMB31 to TXMB0 as a queue buffer at the maximum.

The 31 available options for depth configuration are:

- 0x00: TX Queue disabled
- 0x01: Reserved
- 0x02: 3 messages
- ⋮
- 0x1D: 30 messages
- 0x1E: 31 messages
- 0x1F: 32 messages

The depth of each TXQ2 buffer can be configured by writing to the CFDTXQCC2n.TXQDC[4:0] bits of the TX Queue Configuration/Control Register.

TXQ2 can set from TXMB32 to TXMB63 as a queue buffer at the maximum.

The 31 available options for depth configuration are:

- 0x00: TX Queue disabled
- 0x01: Reserved
- 0x02: 3 messages
- ⋮
- 0x1D: 30 messages
- 0x1E: 31 messages
- 0x1F: 32 messages

The depth of each TXQ3 buffer can be configured by writing to the CFDTXQCC3n.TXQDC[4:0] bits of the TX Queue Configuration/Control Register.

TXQ3 can set from TXMB63 to TXMB32 as a queue buffer at the maximum.

The 31 available options for depth configuration are:

- 0x00: TX Queue disabled
- 0x01: Reserved
- 0x02: 3 messages
- ⋮
- 0x1D: 30 messages
- 0x1E: 31 messages
- 0x1F: 32 messages

When using TXQ1 and TXQ0 simultaneously, the depth of TXQ is 32 or less in total.

When using TXQ3 and TXQ2 simultaneously, the depth of TXQ is 32 or less in total.

Do not access all the TX message buffers forming the TX Queue directly (except TX Message Buffer No. 63, TX Message Buffer No. 32, TX Message Buffer No. 31 and TX Message Buffer No. 0, which acts as TX Queue access window).

When CFDGAFLP0n.GAFLSRD i (i = 0 to 2) is set and the CFDTXQCCin.TXQGWE (i = 0 to 2, n = 0, 1) is also set, a receiving frame is stored in the target TXQ as send data by routing.

When CFDTXQCCn.TXQOWE bit is 1, the TX Queue is in TX Queue overwrite mode. If the message of the same ID is stored in TX Queue when a frame is received and it is stored in TX Queue, an old message is overwritten by a new message. Therefore, an old message is not transmitted. When the old message of the same ID is transmitting and a CAN bus error and an arbitration-lost occur, the message of old ID is not resent.

When using the function in GW mode and TX Queue overwrite mode, the depth of TXQ (CFDTXQCC0n.TXQDC) should be configured to the value which is the various number of ID that is used in the TX Queue plus 3. If it accesses by routing in gateway mode when a TXQ buffer is full, CFDTXQSTS.TXQMLT is set and send data is thrown away.

The function is valid for the standard ID frame and is invalid for the extended ID frame.

Explanation of operation of the TX Queue with same ID over-writing function in GW mode is shown in Figure 33.47.

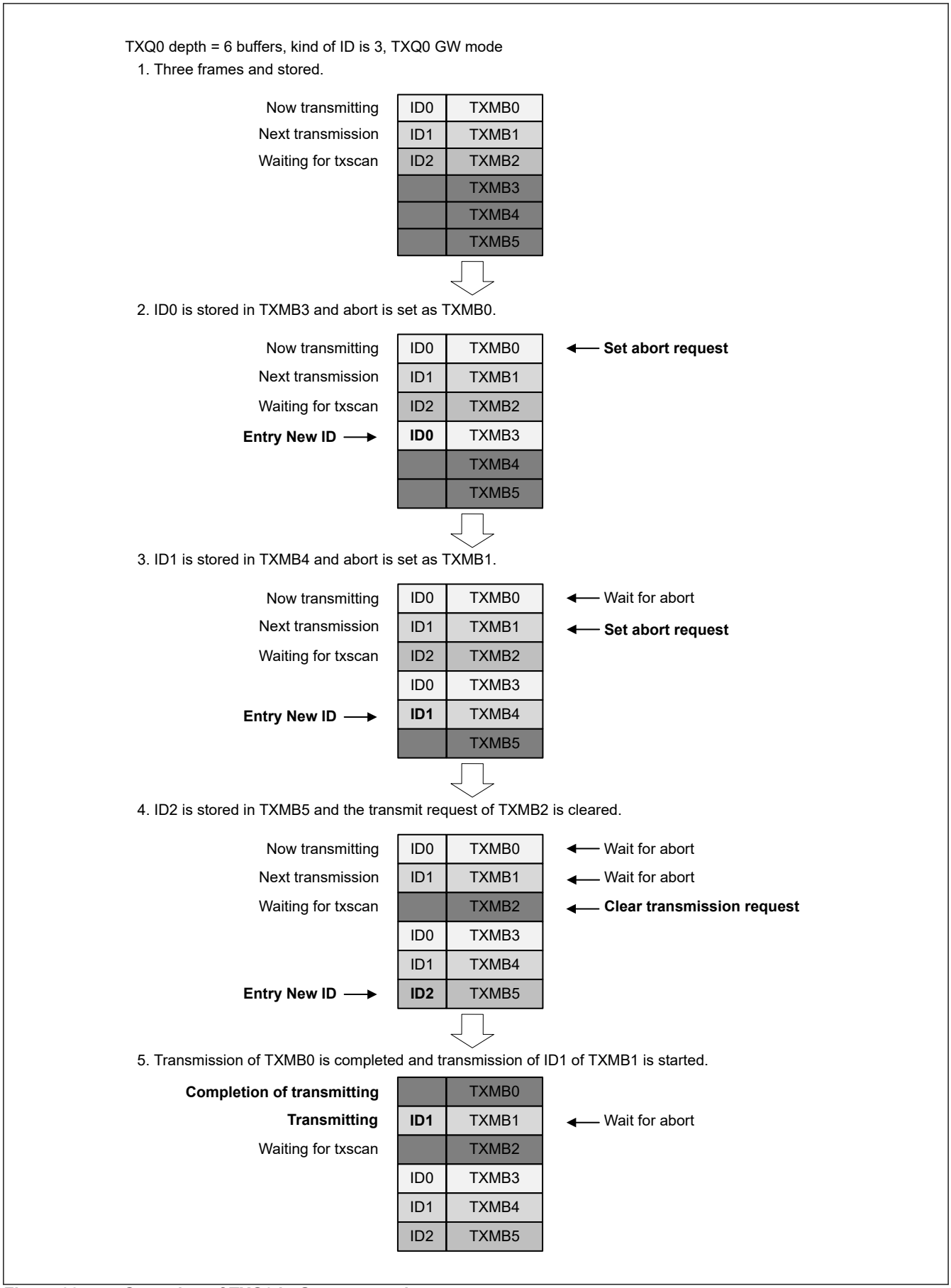


Figure 33.47 Operation of TXQ0 in Gateway mode

When a system writes in TXQ, a system should write in send data, after checking the state of TXQ.

Do not access or configure the related TX Message Buffer Control Registers.

The messages stored to the TX Queue access window are internally stored to a free buffer of the TX Queue.

When the buffer is full then no further access should be done to the queue, until it is no longer full. If it accesses by software writing in when the buffer of TXQ is full, send data is overwritten.

The TX Queue can be disabled by clearing the TXQE bit in the TX Queue Configuration/Control Register. If this bit is cleared, the TX Queue Empty flag is set as follows:

- Immediately if the message from the TX Queue is neither scheduled for the next transmission nor in transmission
- Following the transmission completion, the detection of an error on the CAN bus, loss of arbitration or transition to Channel or Global Halt mode if the transmission from the TX Queue is already scheduled for transmission or already in transmission.

Note: The TX Queue is disabled only when the Empty flag is set after clearing the TXQE bit for the corresponding TX Queue.

Other possible messages pending from the TX Queue are lost and their transmission must be requested again.

Before TXQE is set again, ensure that the CFDTXQSTSn.TXQEMP bit is set and that there is no pending abort from the TX Queue.

When the TXQE bit is cleared, all messages in the TX Queue buffers are lost and no further message should be stored into the TX Queue.

When a message has been stored to the TX Queue, 0xFF must be written into the TX Queue Pointer Control Register. This sets the transmit request automatically and changes the internal message buffer pointer to the next free message buffer location of the TX Queue.

Note: If two messages with the same identifier are stored in the TX Queue, then the order of transmission of these messages can be different from the order in which they were stored in the TX Queue.

To avoid this condition, it is important to confirm that the previous message with the same ID was successfully transmitted before a new message with the same identifier is stored in the TX Queue. Or if TX queue overwrite mode is used, the frame of the same ID is rewritten on a new frame.

For the TX Queue a dedicated interrupt can be enabled by setting the TXQIE bit of the TX Queue Configuration/Control Register.

The interrupt mode can be configured with the CFDTXQCCn.TXQIM bit of the same register either to generate an interrupt for every transmitted message or for the last transmitted message.

The TX Queue transmission request procedure after configuration is shown in Figure 33.48.

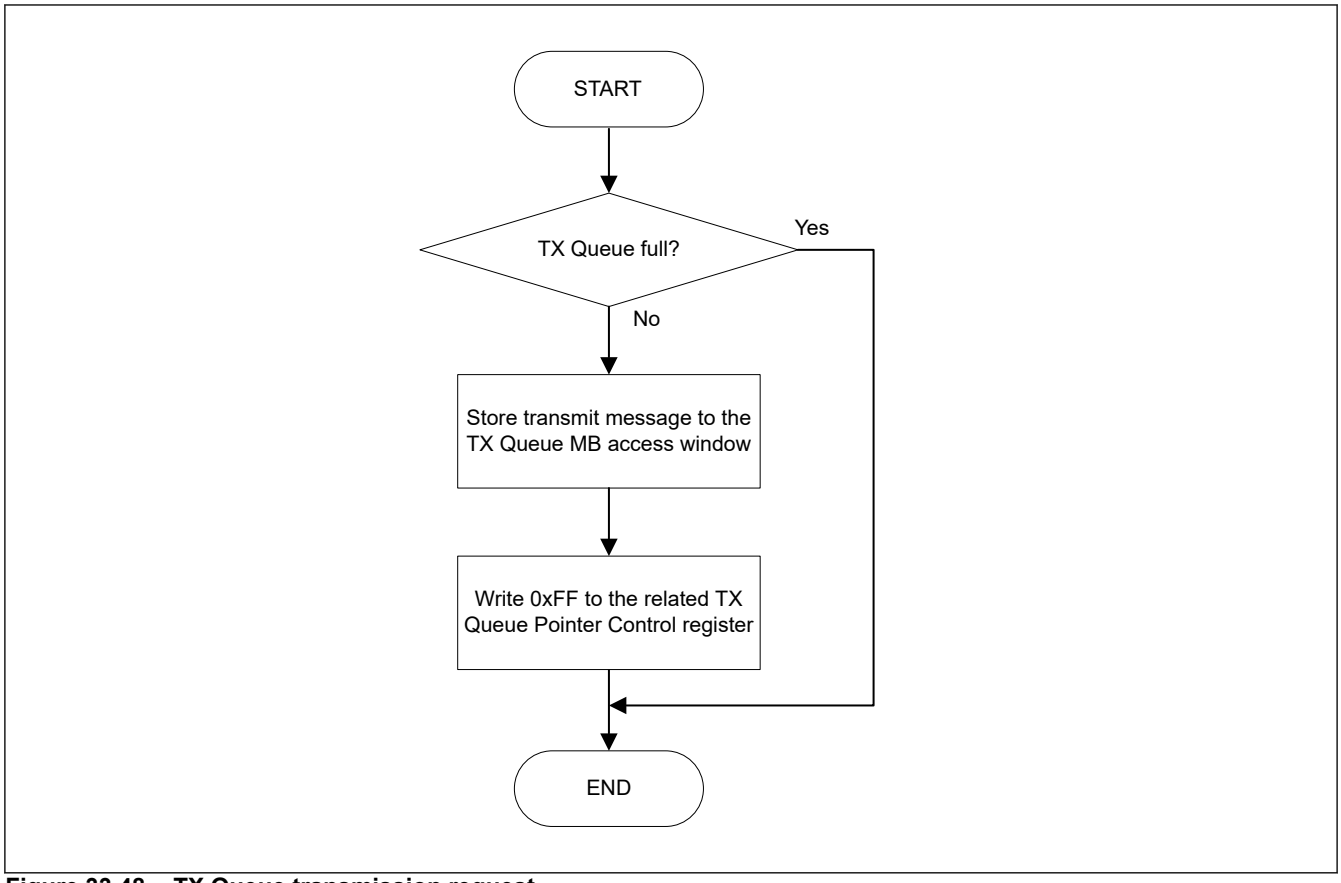


Figure 33.48 TX Queue transmission request

TXQ name	Access window	Range width	Direction	Hardware routing access point	CPU access point	DMA access point	Note
TXQ0	TXMB0	0, 3-32	TXMB0 → TXMB31	Yes	Yes	Yes	When using both TXQ0/1, the total number of stages is 32 or less
TXQ1	TXMB31	0, 3-32	TXMB31 → TXMB0	Yes	Yes	No	
TXQ2	TXMB32	0, 3-32	TXMB32 → TXMB63	Yes	Yes	No	When using both TXQ2/3, the total number of stages is 32 or less
TXQ3	TXMB63	0, 3-32	TXMB63 → TXMB32	No	Yes	Yes	

TXQ0 can use hardware routing, CPU access, and DMA access.

Hardware routing access, CPU access, and DMA access should not be used simultaneously. Choose one access method.

33.8.2.5 TX History List

The TX History List function records the information of the successfully transmitted message in the TX History List buffers for each CAN channel. Two TX History List buffers are provided for each CAN channel and each TX History List buffer can store up to 16 TX History List entries for a CAN channel.

The CFDTHLCCn.THLDTE bit of the TX History List Configuration/Control Register can be used to configure if only message information from TX FIFOs/TX Queue is stored or if all transmit message information from TX Queue, TX FIFO or normal TX message buffers should be stored in the TX History List for a CAN channel.

When a CFDTHLCCn.THLDGE bit is set up, the information on all the frames transmitted in GW mode is stored in TX History List.

Each transmit message can be individually configured for acceptance to the TX History List by the CFDCFID.THLEN bit in the Message Buffer Pointer Register.

The message information is stored to the TX History List Buffer of a CAN channel after the message is successfully transmitted on that CAN channel.

Storing to the List is not synchronized with the status of CFDTMSTSn.TMTRF[1:0] bits in the TX Message Buffer Status Register.

Due to internal processing, the storage to the List can happen with a delay after the successful transmission indication.

Storing the TX History List data can be recognized by the condition that the THLIF bit is set to 1 when the THLIE is configured to 1 or when the TX History List counter CFDTHLSTSn.THLMC[5:0] is increased.

The delay time is dependent on the number of channels due to internal processing.

- Maximum delay time from setting the CFDTMSTSn.TMTRF to storing the TX History List data is 224 peripheral bus clock cycles.

The History list records the following information of the transmitted message:

- Buffer type:
  - 001b: TX Message Buffer
  - 010b: TX FIFO
  - 100b: TX Queue
- Buffer number:
  TX Message Buffer, TX Queue Message Buffer or TX Message Buffer Link for the Common FIFO Buffer from which the transmission occurred. The number depends on the buffer type, see Table 33.28.
- Transmission ID:
  Transmission pointer stored in the transmission message
- Transmit timestamp:
  Message timestamp captured at the capture point as configured by CFDGFDCFG.TSCCFG.
- Transmission information label:
  Transmission information label stored in the transmission message.
- Transmit gateway buffer indication:
  For data transmitted from the gateway, CFDTHLACC0n.TGW bit is set to 1.

Table 33.28 TX History List buffer number entry

CFDTHLACC0n.BT[2:0] buffer type		
001b	**101b**	**100b**
TX Message Buffer	**TX FIFO**	**TX Queue**
TXMB0	Number shown corresponds to the common FIFO. TX Message Buffer Link CFTML of the related Common FIFO Configuration	Number shown corresponds to the message buffer belonging to the TX Queue for which the frame was transmitted.
TXMB1		
TXMB2		
TXMB3		
⋮		
TXMB30		
TXMB31		
TXMB32		
TXMB33		
TXMB34		
⋮		
TXMB61		
TXMB62		
TXMB63		

The Transmission ID entry is used to identify which message of a TX FIFO or TX Queue has been successfully transmitted because the TX FIFO or TX Queue number alone is not sufficient.

Therefore, a unique number can be attached to each transmission message stored in a TX FIFO or TX Queue. This unique identification number should be written to the CFDCFFDCSTSn.CFPTR[15:0] part of the Common FIFO Access Pointer Register for a TX FIFO or to the CFDTMFDCTRn.TMPTR[15:0] part of the TX Message Buffer Pointer Register of the TX Queue access window message buffer.

When the message is successfully transmitted, the identification number is then stored together with the other message related information to the TX History List and can be read with the Transmission ID (TID) of the TX History List Access Register.

Also for normal TX message buffers, the CFDTMFDCTRn.TMPTR[15:0] part of the TX Message Buffer Pointer Register is stored in the Transmission History List. Information label is the same.

Figure 33.49 shows a transmission preparation flow when TX History List is used.

Read access to the TX History List Access Register is done for every single entry.

After reading one entry, 0xFF must be written to the corresponding TX History List Pointer Control Register to be able to access the next entry until TX History List is empty.

Figure 33.50 shows an example flow for processing the TX History List information.

The TX History Lists have dedicated interrupts, which can be configured with the CFDTHLCCn.THLIM bit of the corresponding TX History List Configuration/Control Registers and enabled with the CFDTHLCCn.THLIE bit of the same registers, either to generate an interrupt when the history list reached a filling level of 75% or for every new TX History List entry.

An entry lost indication is flagged by the CFDTHLSTSn.THLELT bit in the TX History List Status Register.

Status of this bit is also shown by the THLES bit in the Global Error Flag Register.

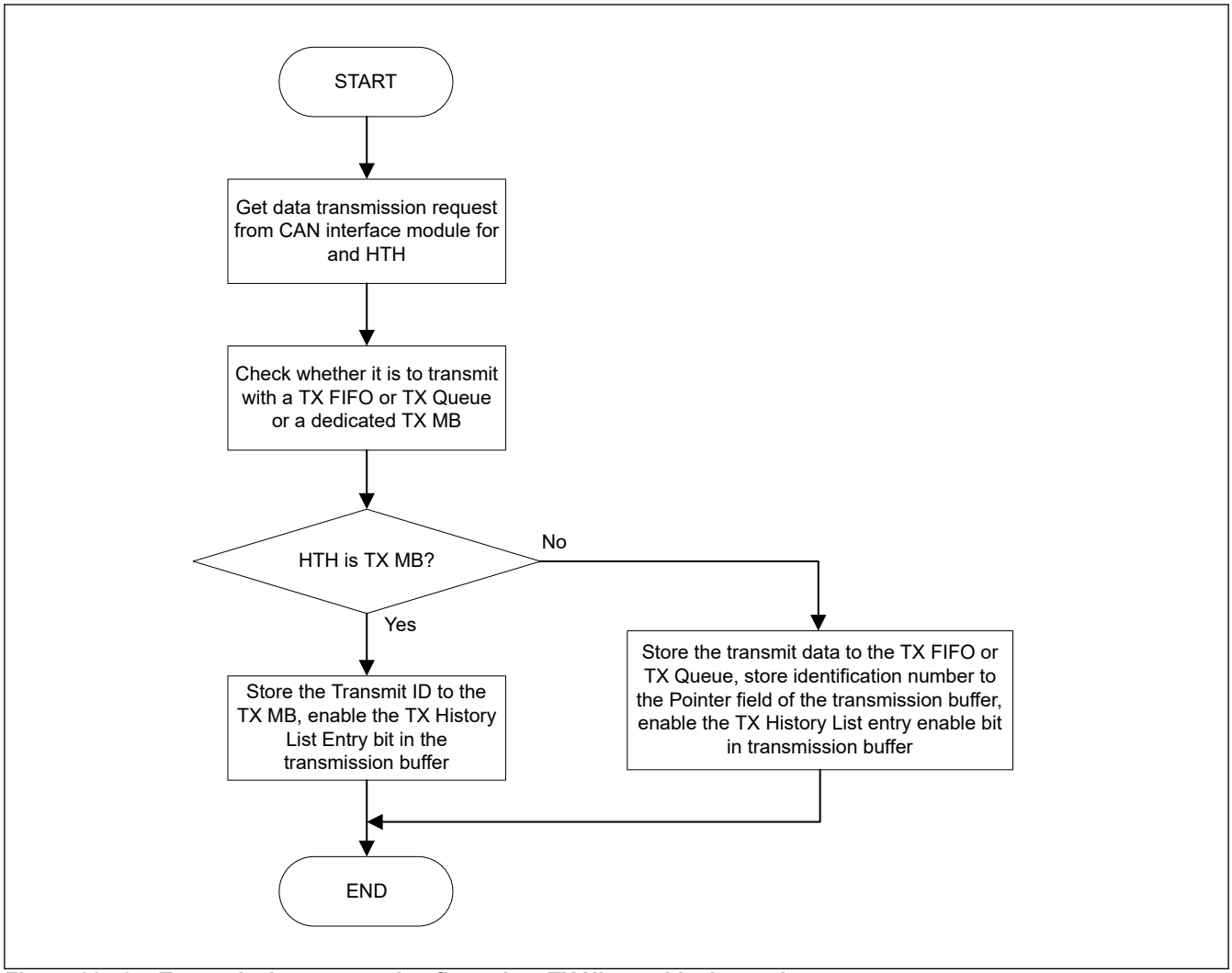


Figure 33.49 Transmission preparation flow when TX History List is used

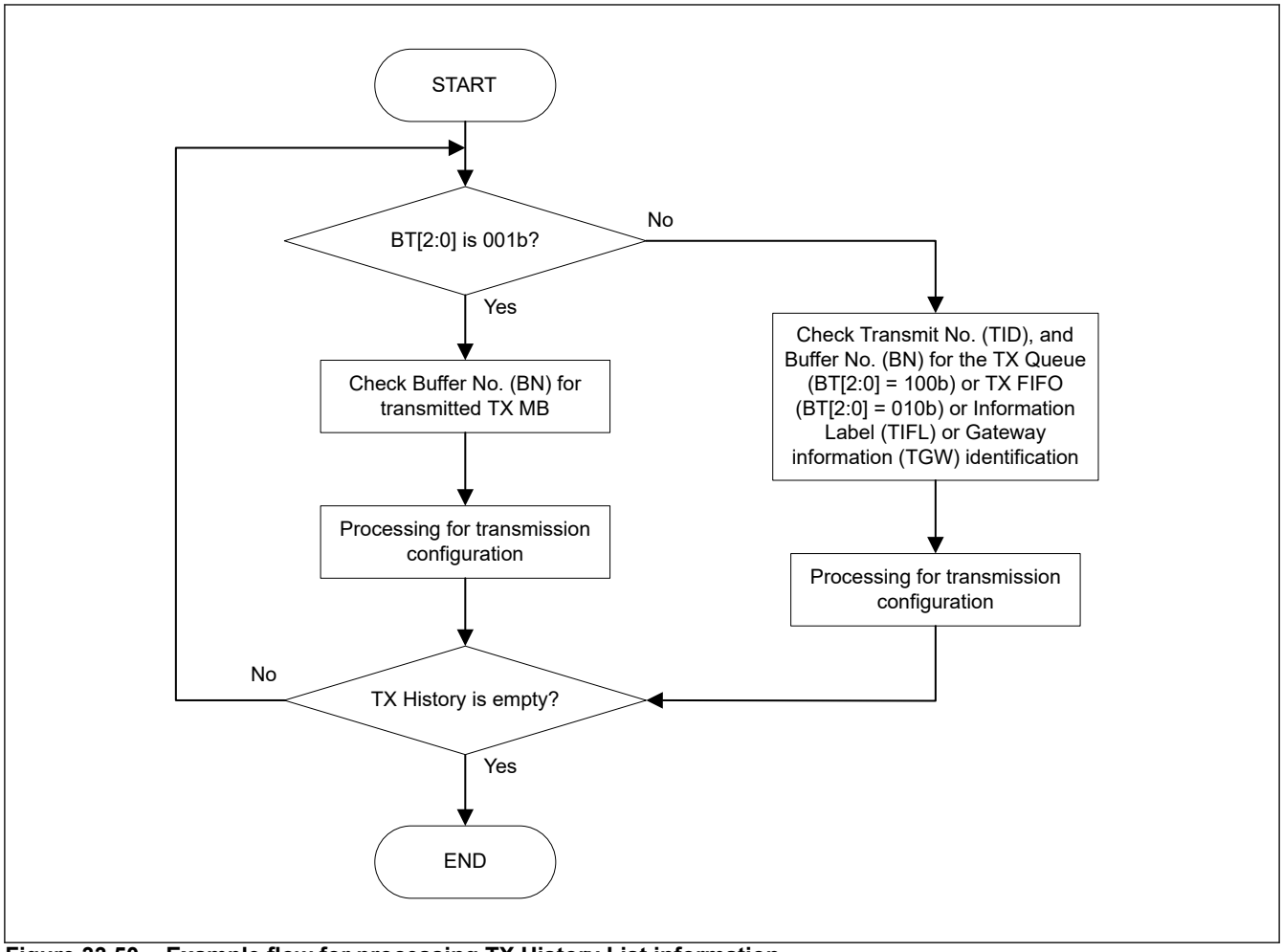


Figure 33.50 Example flow for processing TX History List information

33.8.2.6 TX Data Padding

If the data length code (DLC) of the transmitting message has a higher number of data bytes than the buffer size, then the data bytes beyond the restricted range are replaced by bytes with the value of CC HEX.

This can happen for Common FIFOs configured as (TX or GW) when the transmit message DLC is higher than CFDCFCCn.CFPLS.

This can also happen in FD-only mode, if a Classical Frame is configured with a DLC larger than 8.

33.9 Test Mode

The CAN-FD module can be configured into test modes to allow testing of certain features. These features are provided only for special purposes and care must be taken when configuring the CAN-FD module in test modes.

All test modes are mutually exclusive unless it is explicitly stated that some functions can be enabled across other test modes.

Do not enable any combinations of the various test modes specified in this section.

The test modes can be broadly split into 2 groups:

- Channel specific test modes
- Global test modes

33.9.1 Channel Specific Test Modes

Each CAN channel can be configured into the following test modes:

- Basic test mode

- Listen-only mode
- Self-test mode 0 (External Loop Back mode)
- Self-test mode 1 (Internal Loop Back mode)
- Restricted operation mode

33.9.1.1 Basic Test Mode

The Basic test mode should be used when a particular test setting must be enabled other than when in Listen-Only and Self-test modes.

33.9.1.2 Listen-Only Mode

The ISO 11898-1 recommends an optional Bus-monitoring mode. In this mode, the CAN channel is able to receive valid data frames and valid remote frames. However, it sends only recessive bits on the CAN bus and is not allowed to transmit.

If the CAN engine is required to send a dominant bit (ACK bit, Overload flag, Active Error flag), the bit is routed internally so that the CAN engine monitors this as dominant. The external TX pin remains in recessive state.

This mode can be used for baud rate detection. In this mode, an error interrupt is generated if a bus error occurs and the interrupt is enabled.

In this mode, it is not permitted to request transmission from any normal TX Message Buffer or TX-/GW-FIFO of this channel.

Note: If a message is stored in GW FIFO or Routing TXQ, ensure that the transmitting channel is not in Listen-only mode so that transmission is not requested for this channel from the GW FIFO or Routing TXQ.

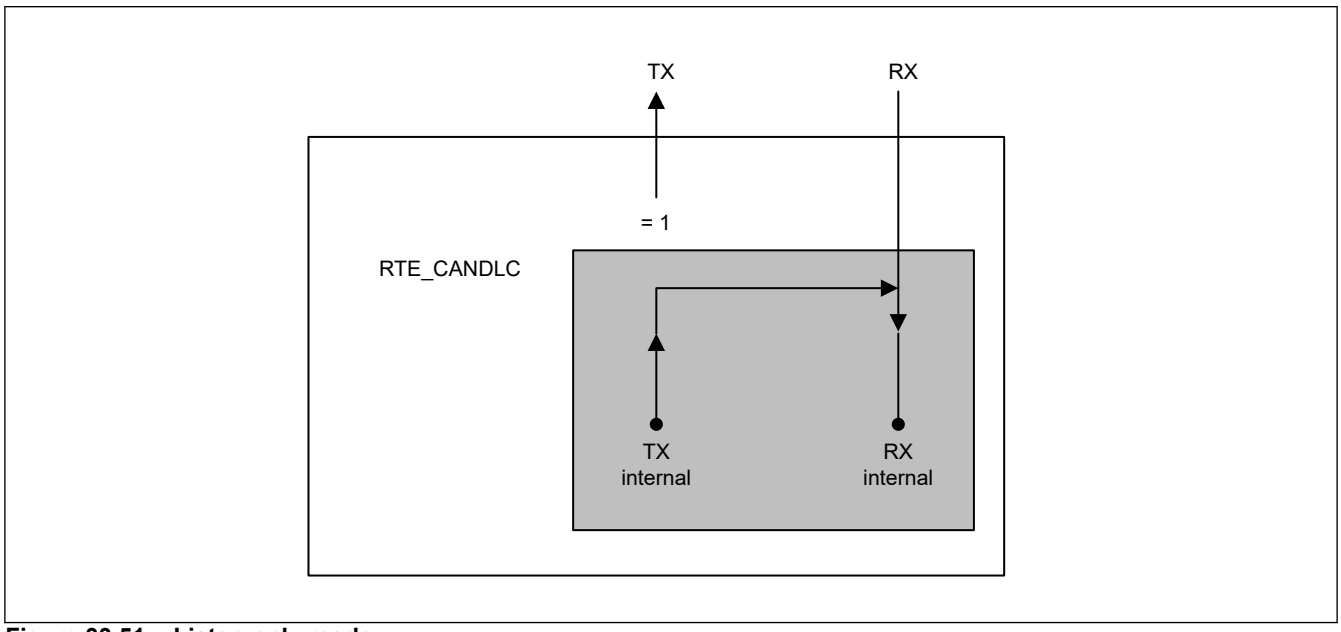


Figure 33.51 Listen-only mode

33.9.1.3 Self-Test Mode 0 (External Loop Back Mode)

In Self-test mode 0, the CAN engine treats its own transmitted messages as received messages through the CAN transceiver and can store them into its receive message buffers.

To be independent from external stimulation, the engine generates its own Acknowledge bit.

This test can be used for CAN transceiver tests.

The RX/TX pins should be connected to the transceiver.

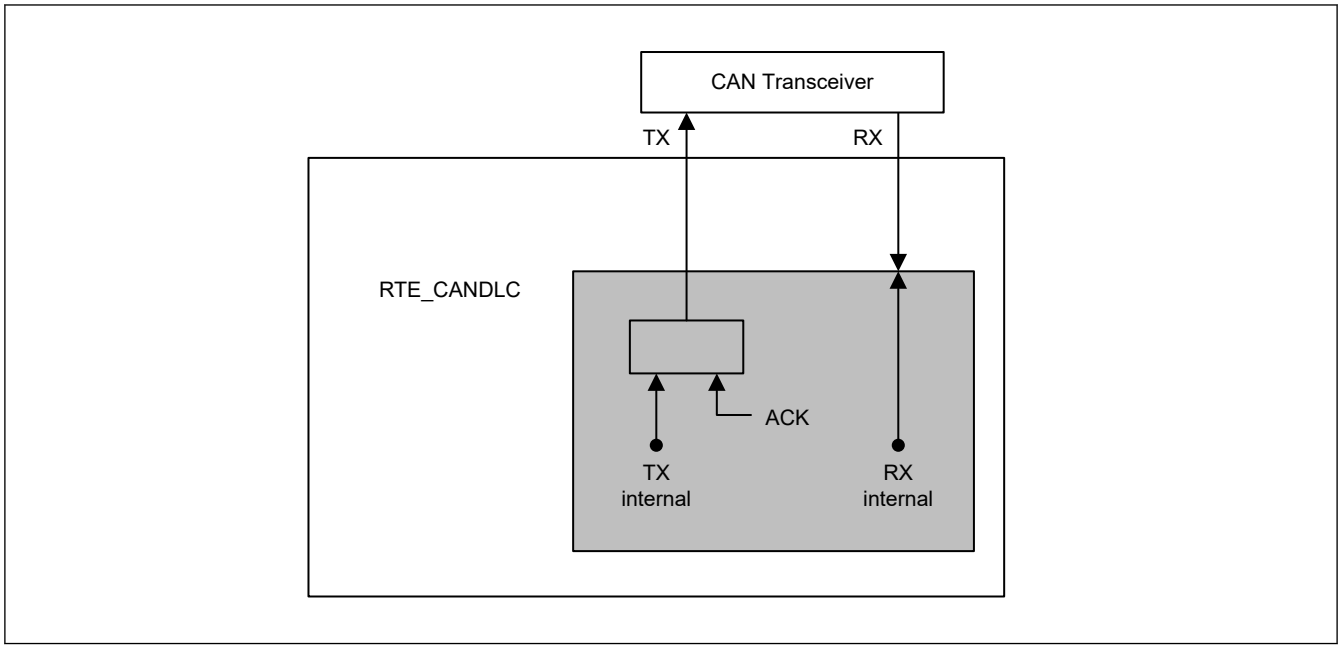


Figure 33.52 Self-test mode 0 (External Loop Back mode)

33.9.1.4 Self-Test Mode 1 (Internal Loop Back Mode)

In Self-test mode 1, the CAN engine treats its own transmitted messages as received messages and stores them into the receive buffer. This mode is provided for self-test functions. To be independent from external stimulation, the CAN engine generates its own Acknowledge bit. In this mode the CAN engine performs an internal feedback from TX internal to RX internal. The actual value of the external RX input is disregarded by the CAN engine.

The external TX pin outputs only recessive bits.

The RX/TX pins do not need to be connected to the CAN bus or any external device.

Note: The channel pins are also disconnected from the internal CAN bus communication line.

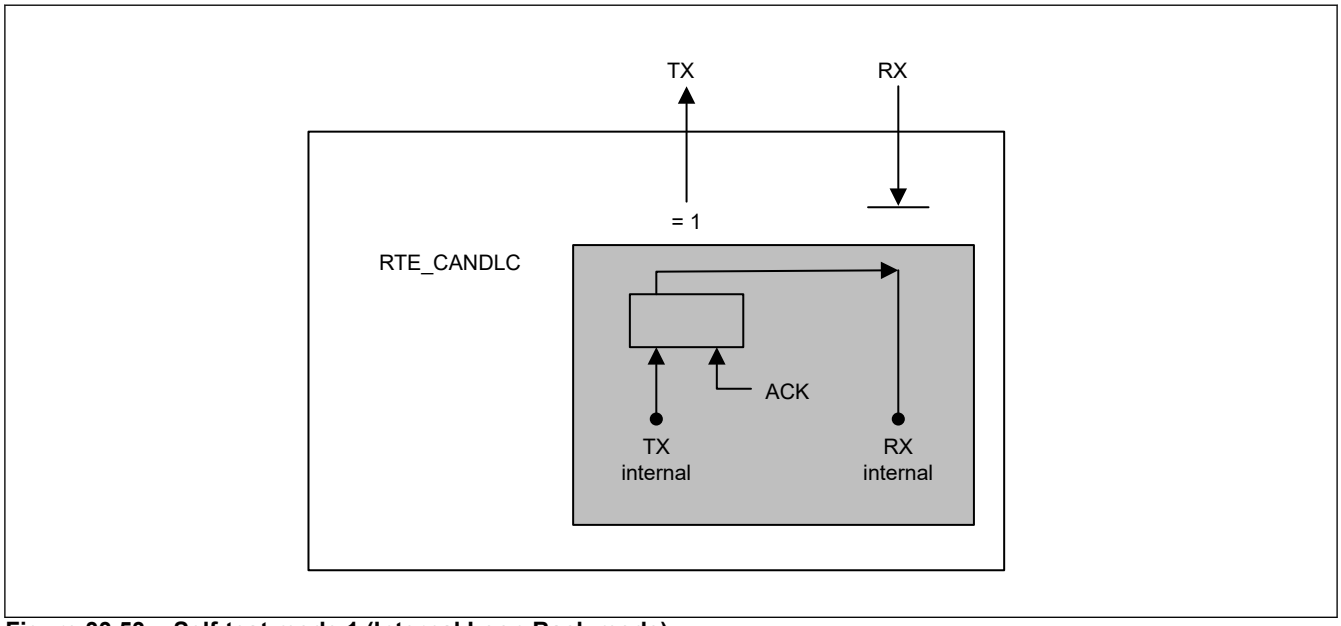


Figure 33.53 Self-test mode 1 (Internal Loop Back mode)

33.9.1.5 Restricted Operation Mode

In Restricted operation mode, the CAN node is able to receive valid data and remote frames that generates the Acknowledge bit.

Active error and overload frames cannot be transmitted, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication after an error or overload condition occurs.

Additionally, the Receive and Transmit Error Counter (REC and TEC) are frozen independently from the occurrence of errors.

The mode is specified as in ISO 11898-1. However, it is permitted to set any requested transmit.

33.9.2 Global Test Modes

The CAN-FD module can be configured into the following test modes:

- RAM test mode
- Internal CAN bus communication mode
- CRC error test

These test modes are protected by a special software procedure to enable the mode. This software procedure enables write access to the test mode by a specific unlock key, the related unlock key can be seen in the following table.

Test mode	Unlock key 1	Unlock key 2
RAM test mode	0x7575	0x8A8A

If the software sequence of the two consecutive unlock key write accesses (half-word or word accesses) is interrupted by any other write access to the SFR or if incorrect data is written to the Global Unlock Key Register then the corresponding test mode cannot be set and the sequence must be restarted.

After the two unlock key write accesses, the next write access should be to set the corresponding Test Mode Enable bit. If this is not followed, the unlock mechanism resets and the Test Mode Enable bit cannot be set. At this time, the unlock sequence must be restarted.

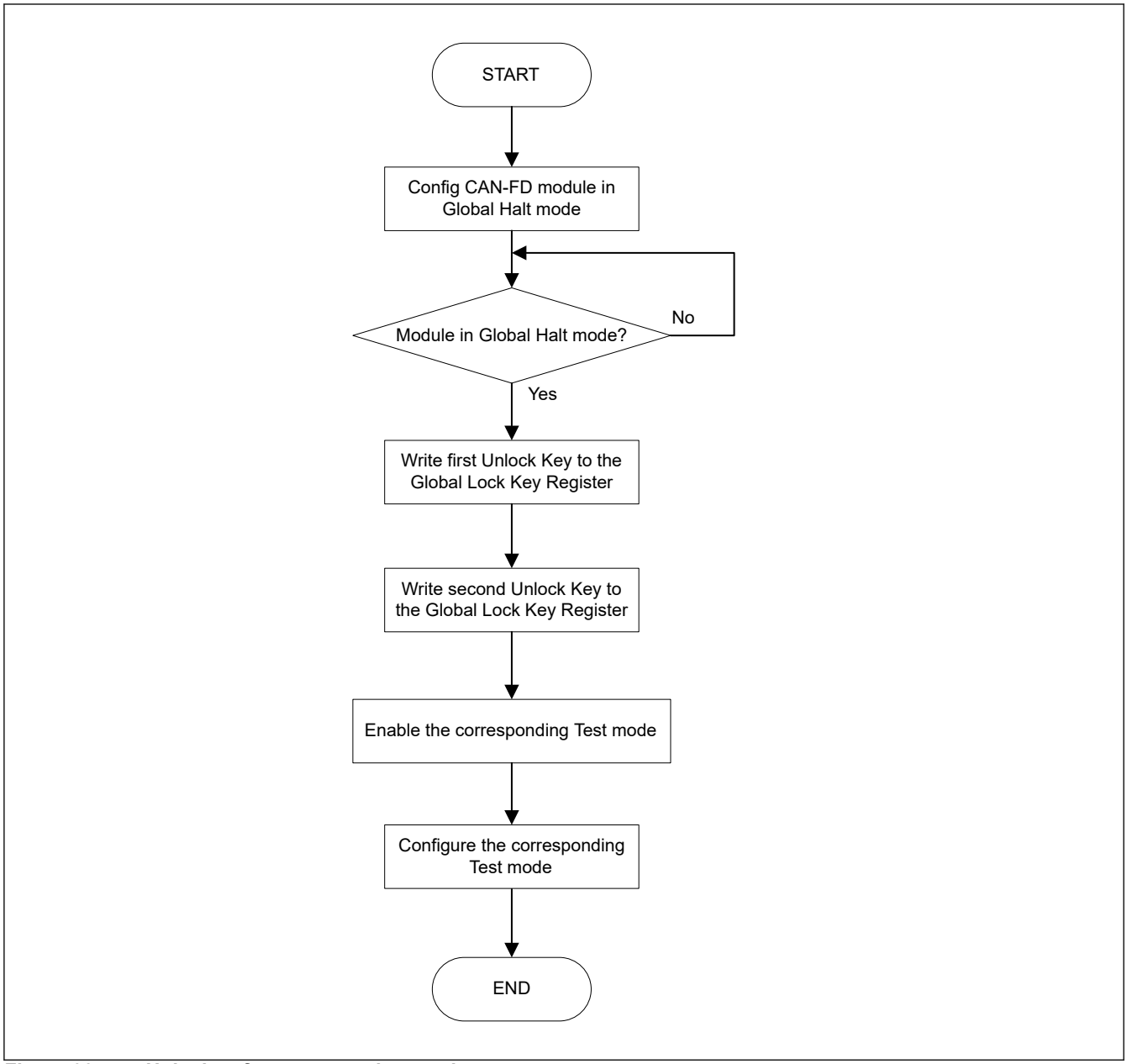


Figure 33.54 Unlock software protection routine

33.9.2.1 RAM Test Mode

The CAN-FD module can be configured in RAM test mode by setting the CFDGTSTCTR.RTME bit in the Global Test Control Register when the corresponding lock key is previously written. This is a special test mode, in which, the complete RAM area can be accessed.

In this mode, the RAM area is split into number of pages (pn) of 256 bytes each. Which can be accessed with the CFDRPGACCn register.

The page should be selected for read/write access by writing to the CFDGTSTCFG.RTMPS[9:0] bits in the Global Test Control Register. Data can then be read from or written into the RAM Test Page Access Registers.

Figure 33.55 shows the structure of the pages in the RAM when performing a RAM test mode.

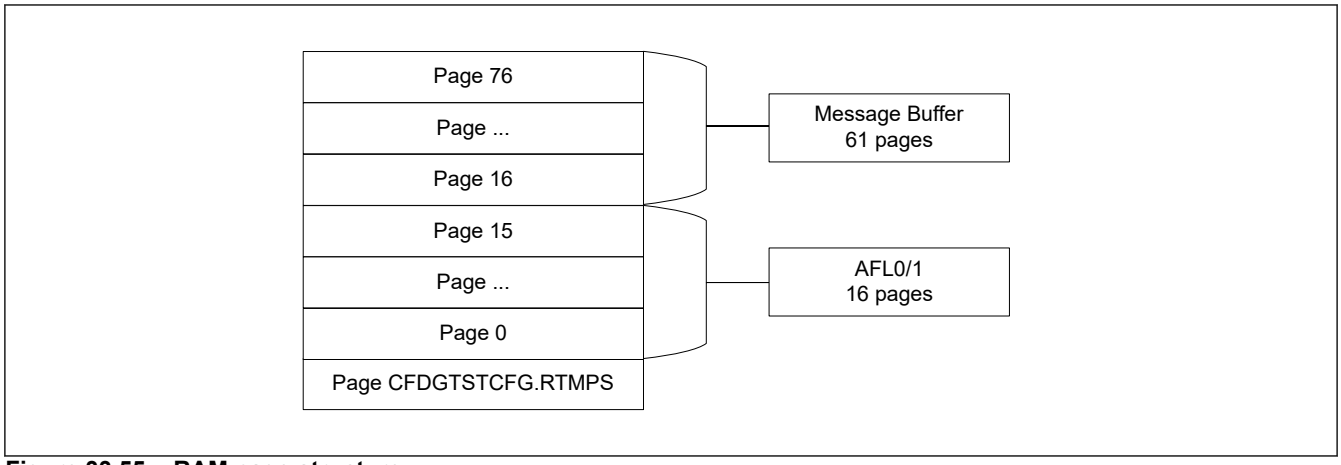


Figure 33.55 RAM page structure

The total available RAM size for a 2-CAN channel version is 4096 bytes for the AFL RAM and 15488 bytes for the Message Buffer RAM.

AFL RAM0/1 can treat RAM test mode as one RAM.

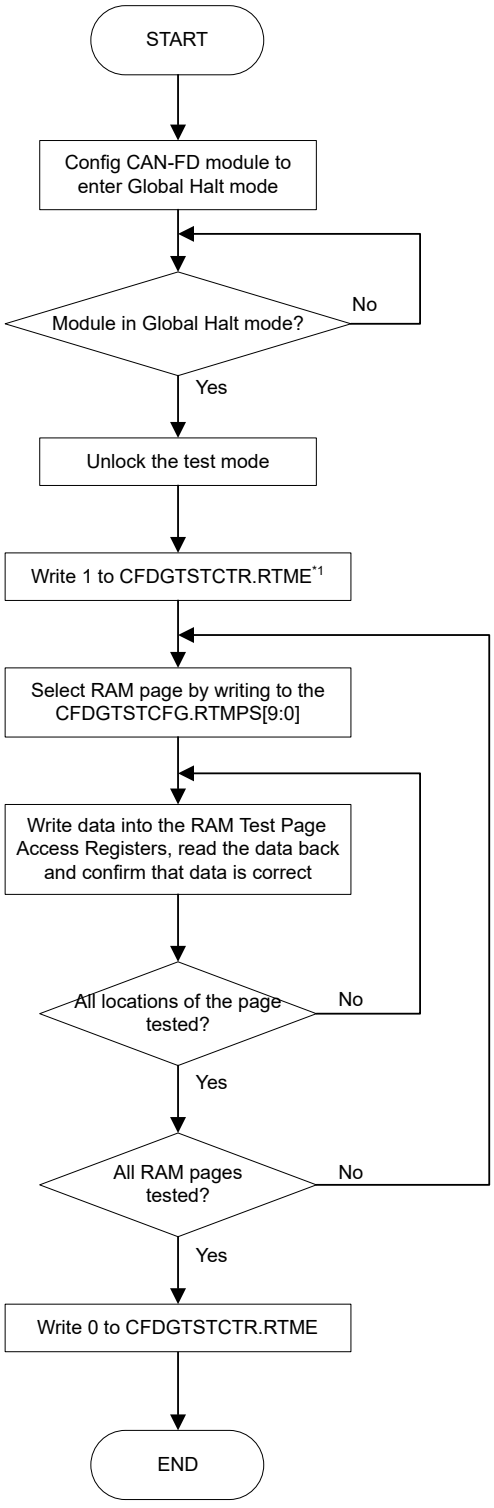
The number of pages (pn) and CFDGTSTCFG.RTMPS[9:0] values for the AFL and MB RAMs are calculated in the following way:

pn = ceil (total RAM size in bytes / number of bytes per page)

- AFL RAM:
  pn = ceil (4096 / 256) = 16 pages
  CFDGTSTCFG.RTMPS[9:0] = 0 to 15 (0x00F) inclusive
- MB RAM:
  pn = ceil (15488 / 256) = 61 pages
  CFDGTSTCFG.RTMPS[9:0] = 16 to 76 (0x04C) inclusive

Do not access more than 128 bytes of RAM on the last page (RTMPS[9:0] = 0x04C).

Figure 33.56 shows the software flow for RAM test mode.



Note 1: Change into the following status before changing to RAMTEST.

- Cancel of a request of transmission
- Disable of all the FIFO and TXQ
- Clear of the receiving flag of a receiving buffer

Figure 33.56 Software flow of RAM test mode

To exit this test mode, clear the CFDGTSTCTR.RTME bit. The CFDGTSTCTR.RTME bit is cleared by writing 0 to it.

The CFDGTSTCTR.RTME bit is cleared automatically when the CAN-FD module enters Global Reset mode from the test mode.

33.9.2.2 Internal CAN Bus Communication Mode

The CAN-FD module can be configured in Internal CAN bus communication test mode by setting the CFDGTSTCTR.ICBCTME bit in the Global Test Control Register. This is a special test mode, in which the CAN channels can be connected together internally to generate a CAN cluster within the CAN-FD module.

Only use the following sequence to enter Internal CAN bus communication test mode:

1. Configure all channels in Halt mode and check that all channels have entered Global Halt mode.
2. Write data into the Global Test Configuration Register to select the channels participating in the Internal CAN bus communication test.
3. Set the CFDGTSTCTR.ICBCTME bit of the Global Test Control Register.
4. Check that CFDGTSTCTR.ICBCTME bit is set in the Global Test Control Register.

In this mode, the TXD outputs of the channels participating (configured) in Internal CAN bus communication mode are connected together using AND gate. The output of the AND gate is connected to the RXD inputs of all participating channels to create a CAN cluster within the CAN-FD module. The channels are isolated from the external CAN bus while the CAN-FD module is in this test mode.

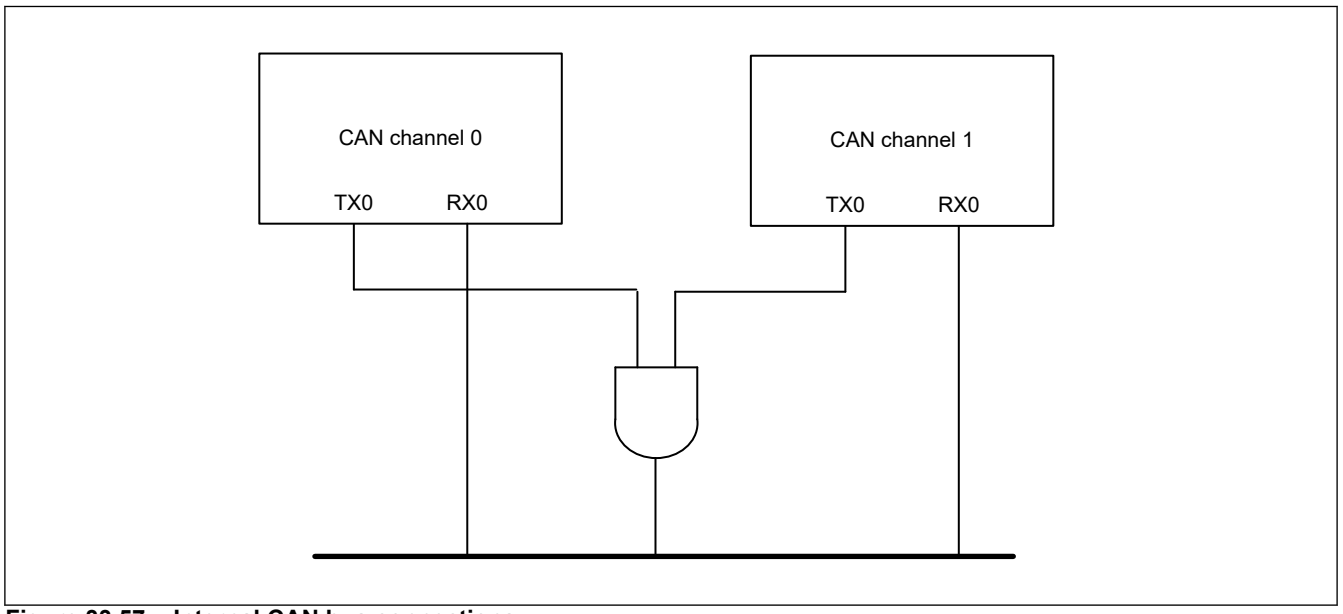


Figure 33.57 Internal CAN bus connections

The AFL, Flat RX message buffers, FIFO buffers, Flat TX message buffers, and various registers can now be configured as normal to start communication between channels.

The channels not participating in Internal CAN bus should only be configured in Halt mode.

33.9.2.3 CRC Error Test

After the CAN-FD module has been configured in Internal CAN bus communication test mode, use the following sequence to perform CRC error testing. In the following sequence, channel x is the reference transmitter CAN-FD module and channel y is the receiver CAN-FD module where (x, y = [0,…, n] and $x \neq y$):

1. Configure channel x node to transmit one reference message.
2. Set the CFDCyCTR.CRCT bit to 1, in order to invert the first bit of the incoming bit stream from channel x.
3. Set the CFDTMCx.TMTR.
4. Read either CFDCyERFL.CRCREG or CFDCyFDCRC.CRCREG (depending on the received frame type: Classical or FD). The value should be different from the received CRC value of the reference message from channel x.

5. Check that CFDCyERFL.CERR is 1.

As the CRC generator logic is shared for RX and TX there is no need to create a separate TX CRC error test.

33.10 Bus Traffic Measurement

The idle time of the CAN bus can be measured using PCLKM or PCLKCAN. Bus traffic can be calculated based on the measurement results.

33.10.1 How to count the CAN Bus Idle Time

Figure 33.58 shows the concept of measuring the idle time of the CAN bus.

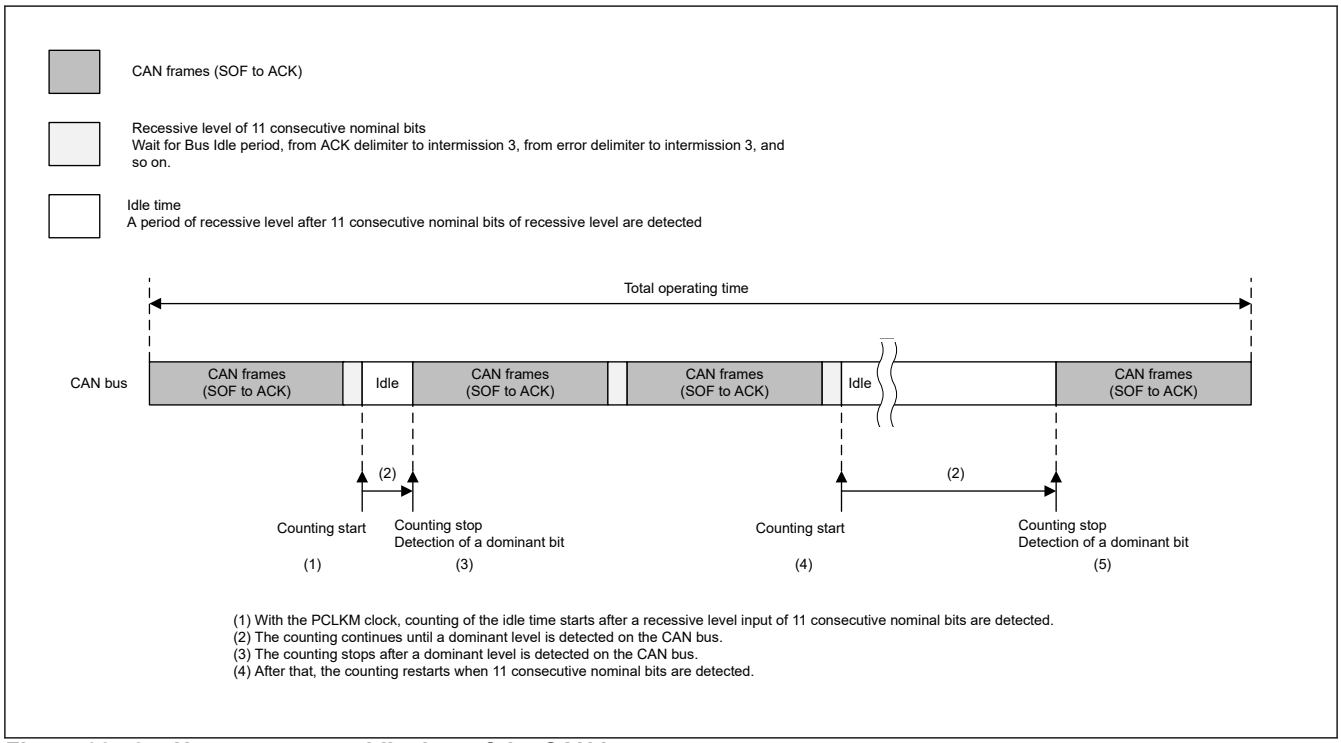


Figure 33.58 How to measure idle time of the CAN bus

33.10.2 Operations and Measurement Procedure

The following procedure is for measuring the idle time of the CAN bus.

1. The channel to be measured transits to operation mode.
2. Write 1 to CFDCnBLCT.BLCE bit to set the measuring counter to operating mode.
3. Write 1 to CFDCnBLCT.BLCLD bit to clear the counter register.
4. Detect a recessive level input of 11 consecutive nominal bits.
5. Start counting the bus idle time.
6. Detect a dominant level.
7. The counting stops.
8. Detect a recessive level input of 11 consecutive nominal bits.
9. The counting starts.
10. Write 1 to CFDCnBLCT.BLCLD bit to clear the counter register and simultaneously load the counter value to CFDCnBLSTS.
11. Read the value of CFDCnBLSTS.

To stop the measurement counter, write 0 to CFDCnBLCT.BLCE bit.

To initialize the counter, write 1 to CFDCnBLCT.BLCLD bit.

This measurement is enabled when the channel to be measured is in Operation mode.

When the relevant channels are in reset mode, the counter does not operate.

Also, accurate measurements are not available in test mode.

Write 1 to CFDCnBLCT.BLCLD bit to clear the counter register and simultaneously load the value of the counter to CFDCnBLSTS.

The lower 3 bits of CFDCnBLSTS are fixed to 0.

Based on the values of the counter, software can calculate the CAN bus traffic according to the following formulas:

$$\frac{(\text{Total operating time} - \text{Total idle time})}{\text{Total operating time}} = \frac{\text{Bus operating time}}{\text{Total operating time}} = \text{Bus usage ratio}$$

- Total idle time: a value read from CFDCnBLSTS × a clock cycle of PCLKM
- Total operating time: a setting interval of CFDCnBLCT.BLCLD bit

Example: Below is a calculation example under the following conditions:

- Conditions: nominal bit rate = 1 Mbps
- PCLKCAN clock = 40 MHz (= 25 ns)
- A setting interval of CFDCnBLCT.BLCLD bit = cycle of 1 ms
- A read value of CFDCnBLSTS register = 0x4E20

$$\frac{(\text{Total operating time} - \text{Total idle time})}{\text{Total operating time}} = \frac{(1000000\ \text{ns} - 20000 \times 25\ \text{ns})}{1000000\ \text{ns}} = 50\%$$

33.11 Flexible CAN Mode

This is a mode in which it is possible to connect the CAN modules of 2 channels to a single CAN driver.

The pair of Channel in flexible CAN mode is as follows:

When CFDGFCMC.FLXC0 bit is set, Channel 0 and Channel 1 of a CAN-FD module are Flexible CAN mode.

The Channel 1 uses TX/RX terminal of Channel 0, and the TX/RX terminal of Channel 1 cannot use.

In Flexible CAN mode, each channel performs communication processing independently. However, when one of the channels transmits, the other channel will not return an acknowledge bit.

Note: When operating in Flexible CAN mode, the error counters (TEC/REC) of the two CAN nodes are not synchronized with each other.

Flexible transmission buffer assignment configured in CFDGFTBAC register and Flexible CAN mode configured in CFDGFCMC register should not be used simultaneously.

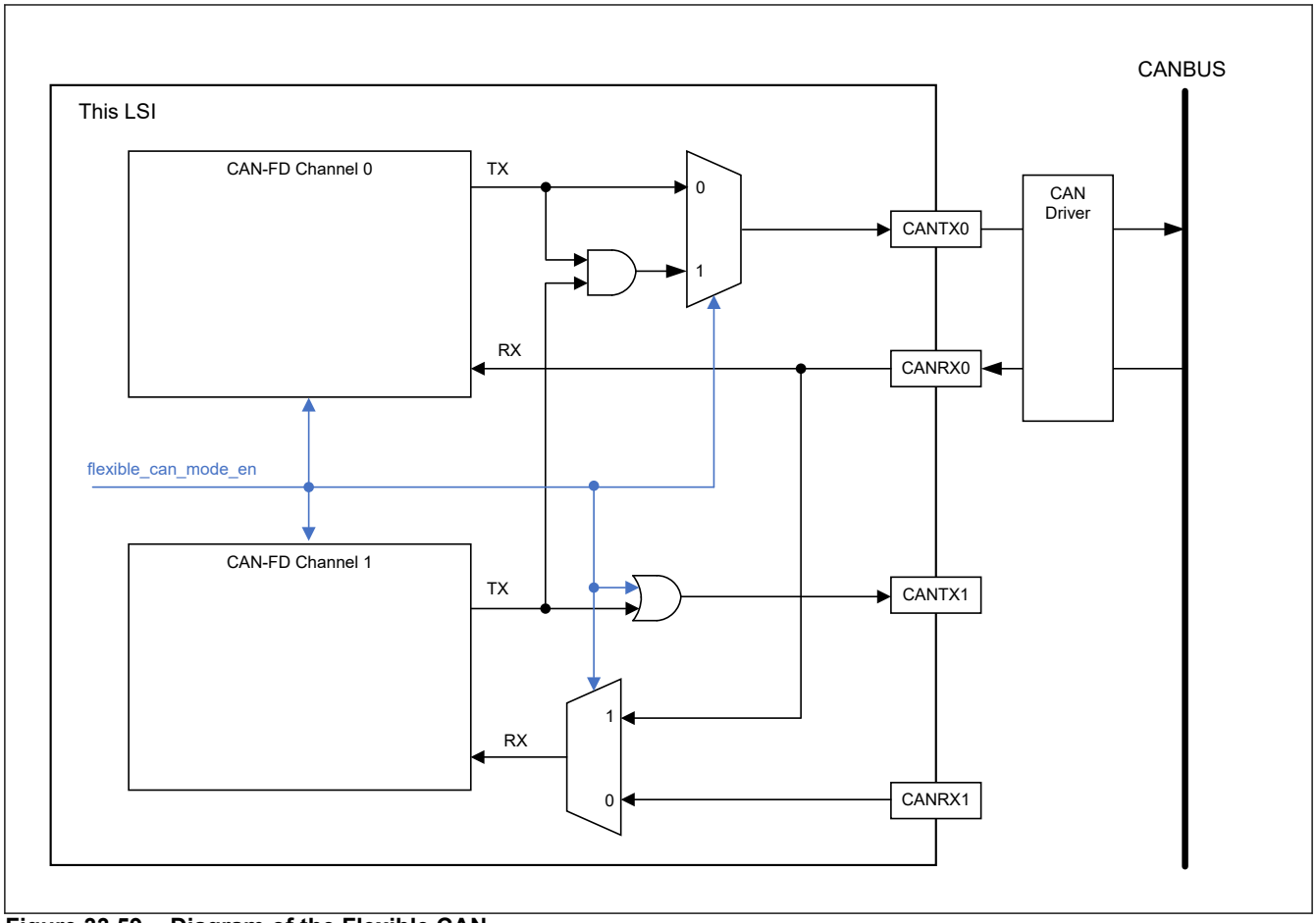


Figure 33.59 Diagram of the Flexible CAN

33.12 Flexible Transmission Buffer Assignment

Each Channel has 64 transmit buffers by exclusive use.

In order to correspond to an insufficient case by a TXMB, this Channel can rent a maximum of 32 TXBMs from the next Channel.

The buffer which can be rented becomes every four units by 4 to 32 buffers.

When CFDGFTBAC.FLXMB0 bit is set, flexible transmission buffer assignment is performed between Channel 0 and Channel 1.

Flexible transmission buffer assignment configured in CFDGFTBAC register and Flexible CAN mode configured in CFDGFCMC register should not be used simultaneously.

Interrupt of the rented buffer is output to interrupt of the rented channel.

When using TXQ by the rented buffer, TXQ should only set within the rented range.

The TXMB lent out operates in the mode of the channel to be used. For example, when the channel 1 is in a reset mode, using rented TXMB and transmission of the channel 0 is possible. Moreover, the rented TXMB is affected by the transmitting status, TX SCAN PROCESS, or transmitting abortion of the rented channel.

In the case of message buffer number priority mode, the priority of TXMB0 is high and the priority of TXMB95 (case which rented 32 buffers) becomes low. Priority of rented TXMB becomes low.

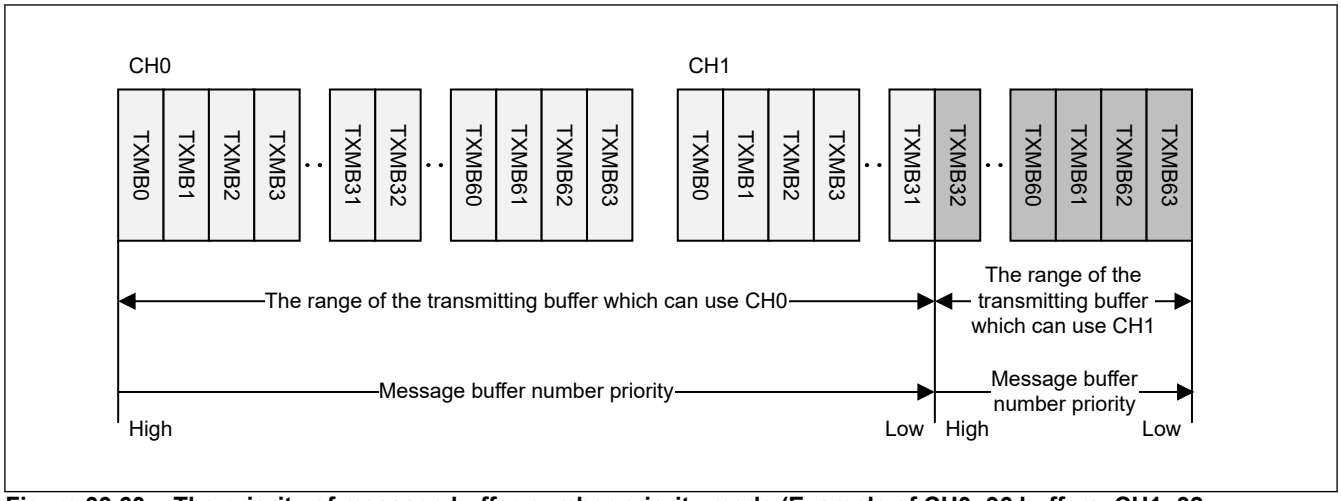


Figure 33.60 The priority of message buffer number priority mode (Example of CH0: 96 buffers, CH1: 32 buffers)

The example of a rental between channel 0 and channel 1 is shown below.

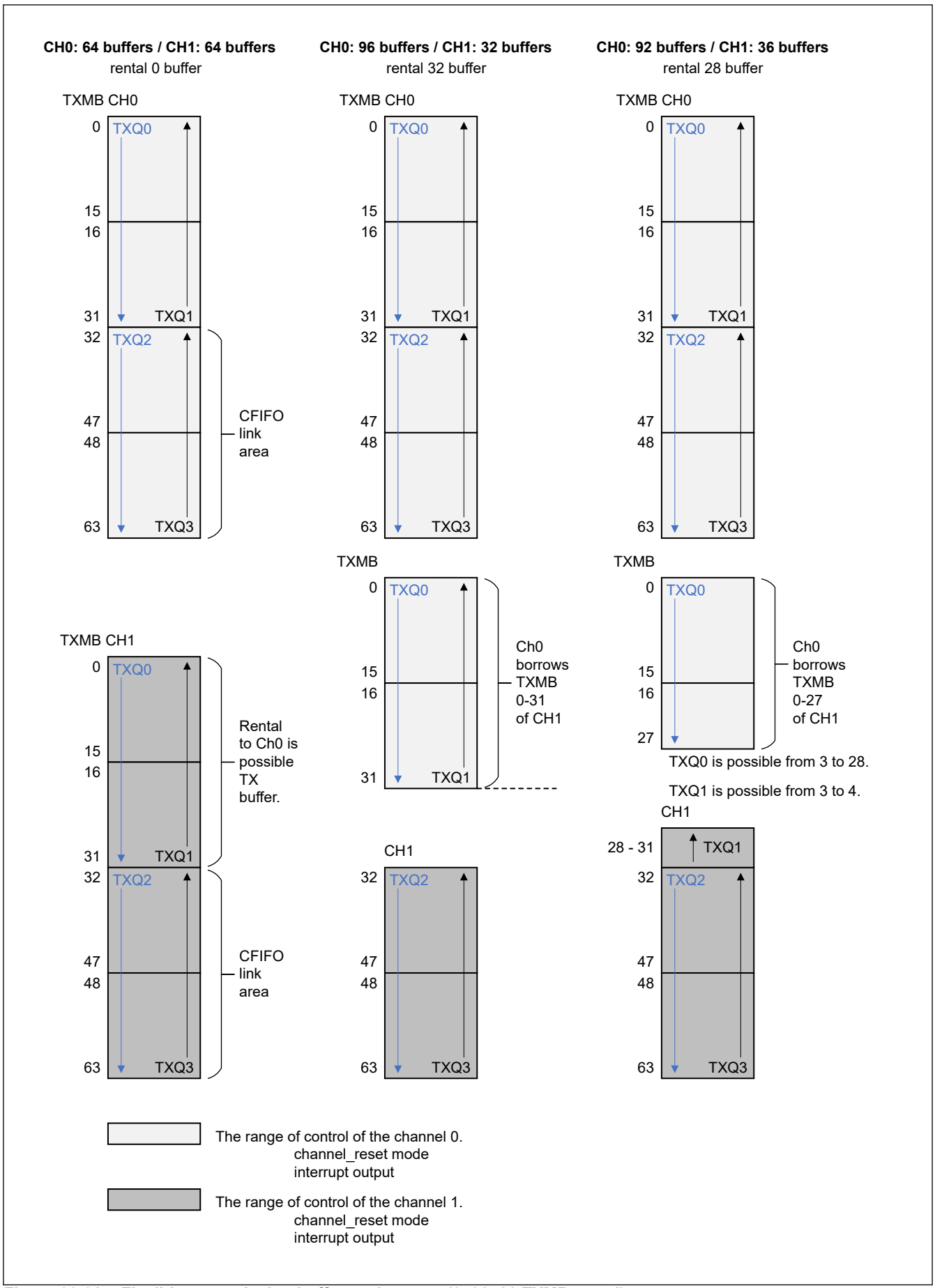


Figure 33.61 Flexible transmission buffer assignment (0, 32, 28 TXMB rental)

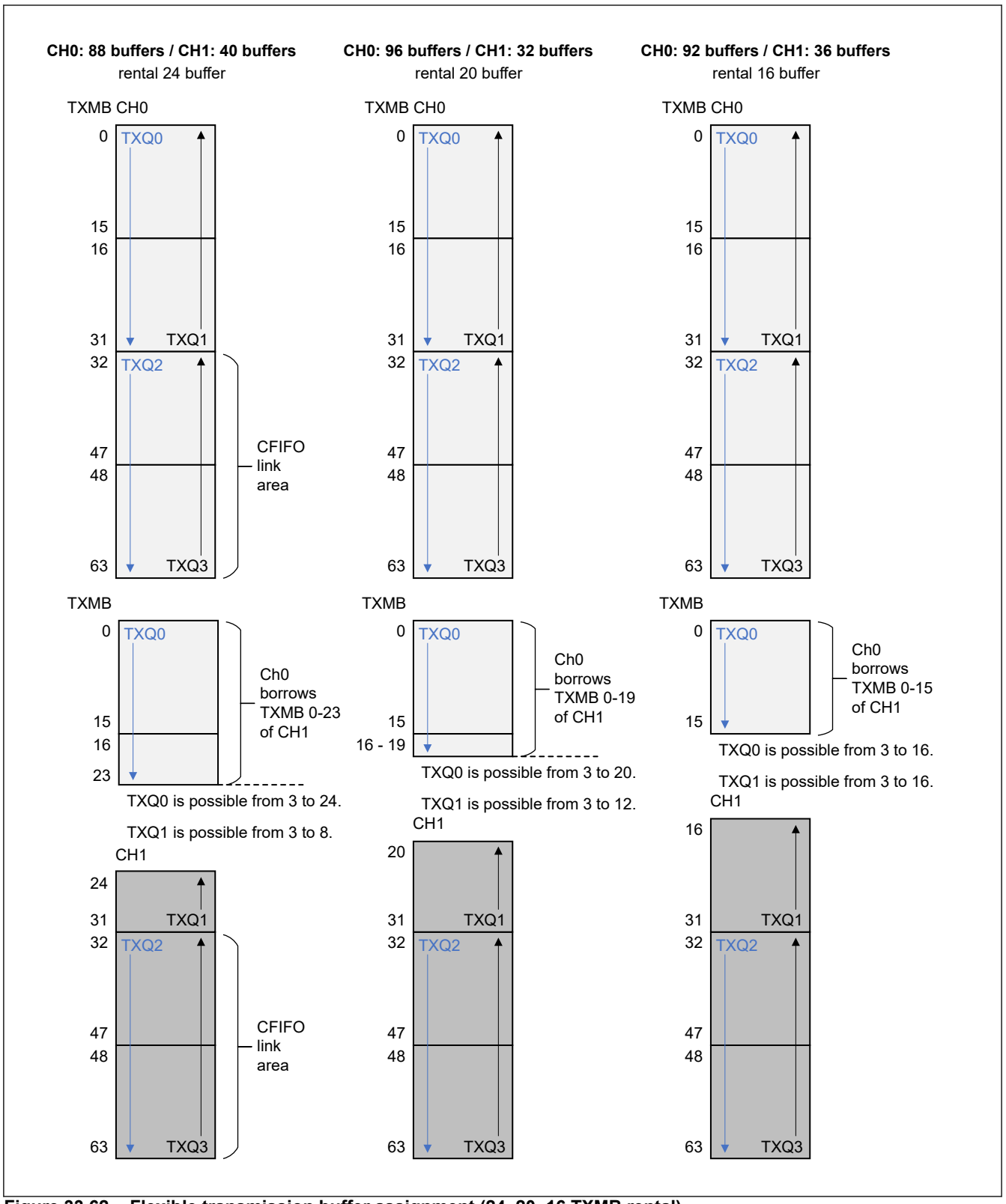


Figure 33.62 Flexible transmission buffer assignment (24, 20, 16 TXMB rental)

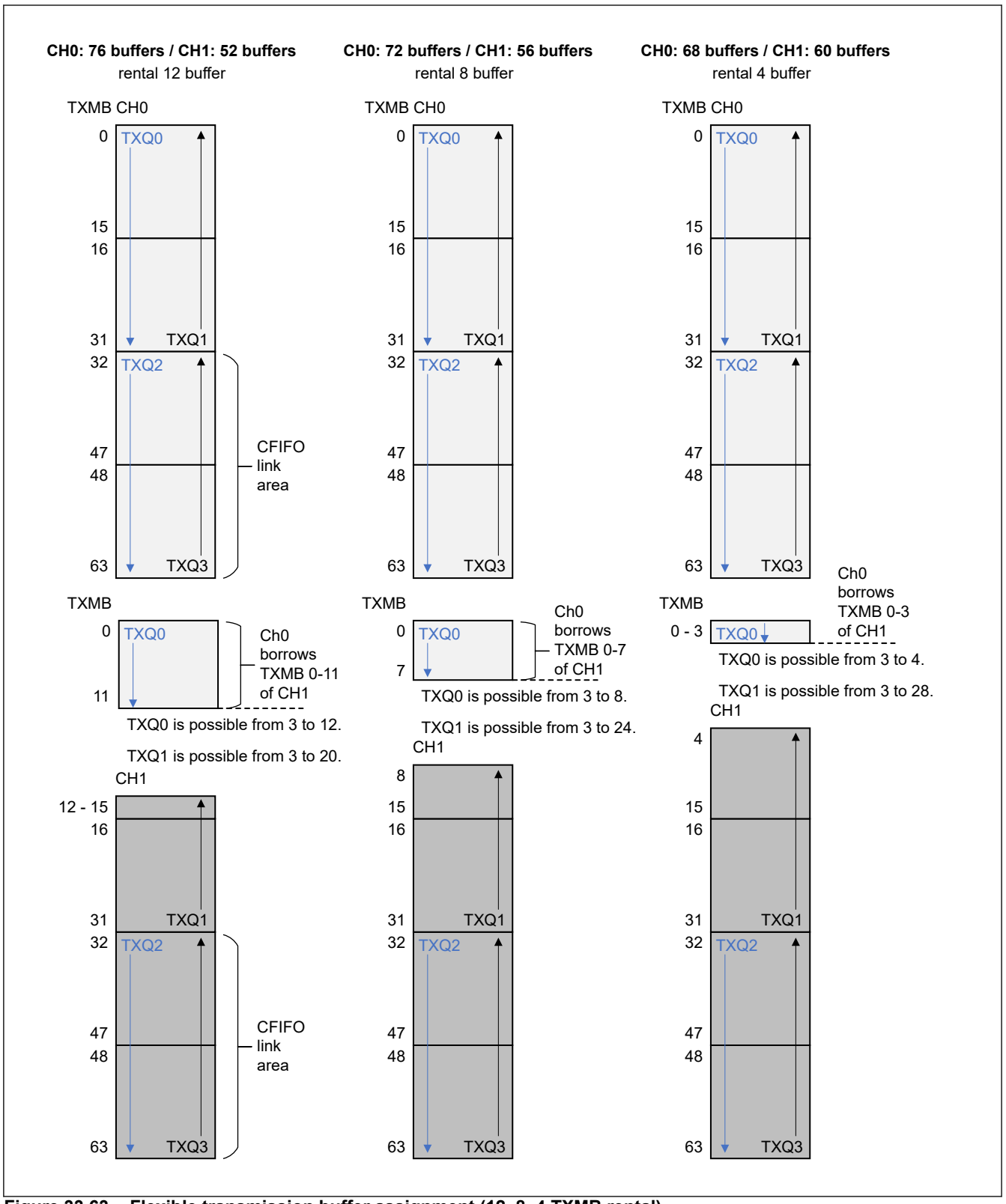


Figure 33.63 Flexible transmission buffer assignment (12, 8, 4 TXMB rental)

34. Serial Peripheral Interface (SPI)

34.1 Overview

The SPI supports serial communications for transmit-receive, transmit only, and receive only. The SPI is provided with a function for high-speed serial communication with multiple processors and peripheral devices.

34.1.1 Features

Table 34.1 lists the specifications of the SPI, and Figure 34.1 shows a block diagram of the SPI.

Table 34.1 SPI specifications (1 of 2)

Item	Description
Number of channels	4 channels (ch0 to ch2 in NON-SAFETY and ch3 in SAFETY)
Transfer functions	• SPI serial communication (4-wire) and clock synchronous (3-wire) serial communication are possible by using the MOSI (Master Out Slave In), MISO (Master In Slave Out), SSL (Slave Select), and RSPCK (SPI Clock) signals. • Transmit-only operation is available • Receive-only operation is available • Serial communication is possible in master mode and slave mode • RSPCK polarity switching • RSPCK phase switching
Data format	• MSB first or LSB first selectable. • Transfer bit length selectable from 4 to 32 bits • 32 bit x 4 stage FIFO transmit and receive buffers • Up to four frames transferable in one round of transmission or reception (each frame consisting of up to 32 bits) • Byte swap operating function • Transmit/receive data inversion
Bit rate	• In master mode, the on-chip baud rate generator divides the PCLKSPIn to generate RSPCK. A division ratio of 2 to 4096 is settable. • In slave mode, an external input clock is used as a serial clock. The maximum frequency = PCLKSPIn/2 (High width: 1 PCLKSPIn cycles, low width: 1 PCLKSPIn cycles)
Buffer configuration	• Transmit buffer and receive buffer are configured independently.
Error detection	• Mode fault error detection • Underrun error detection • Overrun error detection • Parity error detection

Table 34.1 SPI specifications (2 of 2)

Item	Description
SSL control function	[Motorola SPI mode] ● Four SSL pins (SSLn0 to SSLn3) each channel ● In single master mode, SSLn0 to SSLn3 pins are output. ● In multi-master mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins either for output or unused ● In slave mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins unused ● Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) ● Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) ● Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) ● Configurable delay between frames in burst transfer ● Function for changing SSL polarity [TI SSP mode] ● Four SSL pins (SSL0 to SSL3) each channel ● In single master mode, SSLn0 to SSLn3 pins are output. ● In multi-master mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins either for output or unused ● In slave mode, SSLn0 pin for input, and SSLn1 to SSLn3 pins unused ● Controllable delay from SSL output assertion to SSL output negation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) ● Controllable delay from RSPCK stop to Output disable (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) ● Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units) ● Configurable delay between frames in burst transfer ● Function for changing SSL polarity
Communication Protocol	● Motorola SPI ● TI SSP (Synchronous Serial Protocol)
Synchronization bypass function	● Synchronization circuit can be bypassed using bus clock (PCLKM) as operation clock (PCLKSPIn)
Control in master transfer	[Motorola SPI mode] ● Transfers of up to eight commands each can be executed sequentially in looped execution ● For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity and phase, transfer data length, LSB first/MSB first, burst, RSPCK delay, SSL negation delay, next-access delay ● Transfers can be initiated by writing to the transmit buffer ● MOSI signal value specifiable in SSL negation ● RSPCK auto-stop function [TI SSP mode] ● Transfers of up to eight commands each can be executed sequentially in looped execution ● For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity and phase, transfer data length, LSB first/MSB first, burst, RSPCK delay, SSL negation delay (Output disable delay), next-access delay ● Transfers can be initiated by writing to the transmit buffer ● MOSI signal value specifiable in SSL negation ● RSPCK auto-stop function
Interrupt sources	● Maskable Interrupt sources SPI receive buffer full / Receive data ready interrupt SPI transmit buffer empty interrupt SPI communication end interrupt SPI error interrupt (mode fault, overrun, parity error, receive data ready) SPI idle interrupt (SPI idle)
Others	● Switching between CMOS output and open-drain output ● SPI disable (initialization) function ● Loopback mode function
Module-stop function	Module-stop state can be set to reduce power consumption*1

Note 1. Module stop condition: It depends on the setting higher than this module. This module is not reset, and the clock stops on module stop condition. It contributes to low power consumption, because the clock stops.

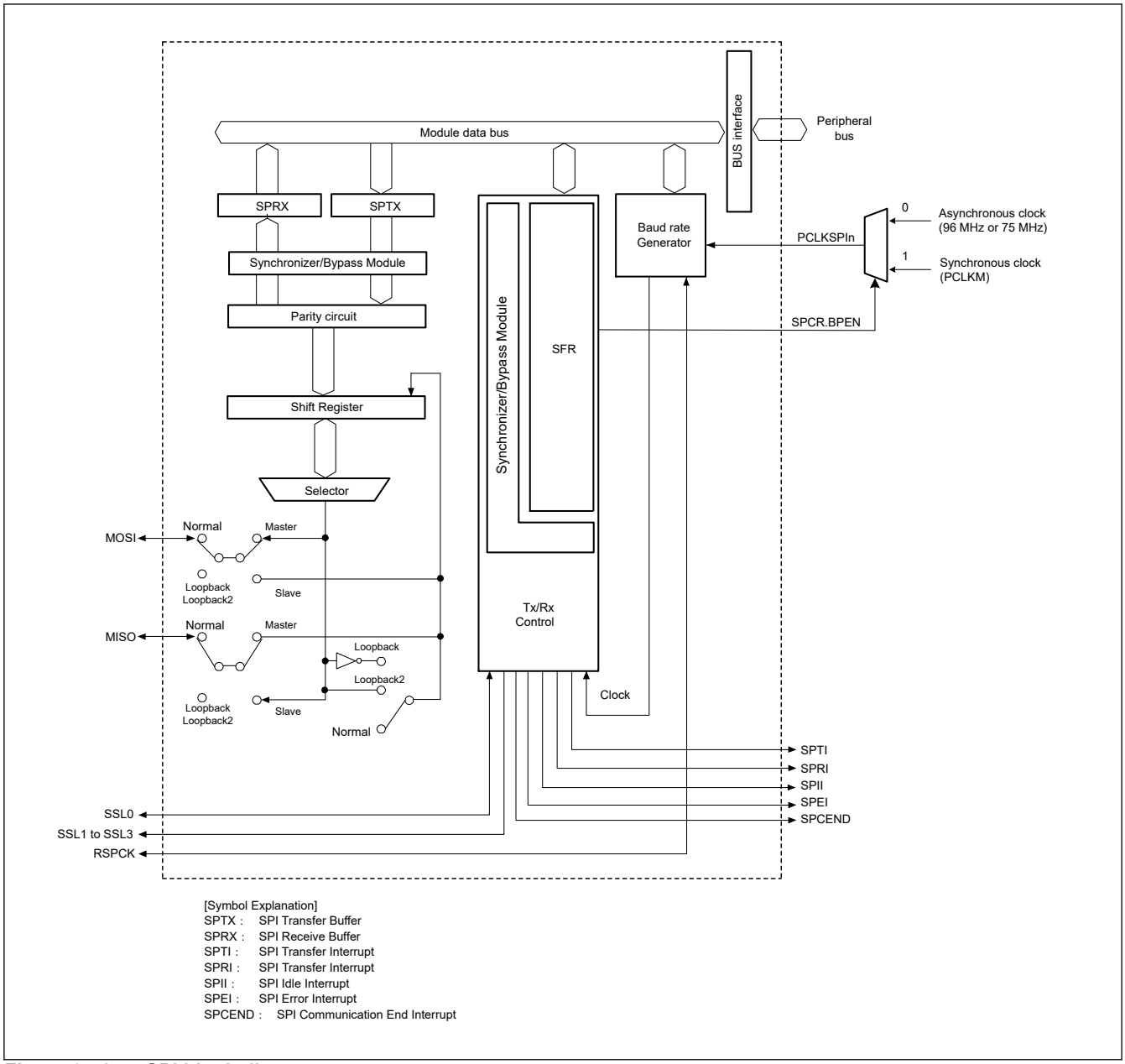


Figure 34.1 SPI block diagram

The SPI automatically switches the SSL0 pin input and output directions. The SSL0 pin is switched to output when single master is set, and it is switched to input when multi-master or slave is set. Furthermore, the SPI automatically switches the input and output directions of pins RSPCK, MOSI, MISO, SSL1 to SSL3 according to the master/slave setting and SPI operation (4-wire)/clock synchronous operation (3-wire) SSL0 input level. (See section 34.3.2. SPI Pin Control.)

Table 34.2 lists the I/O pins used in the SPI.

Table 34.2 SPI pin configuration

Channel	Pin name	I/O	Function
SPIn (n = 0 to 3)	SPI_RSPCKn	I/O	Clock I/O
	SPI_MOSIn	I/O	Master transmit data I/O
	SPI_MISOn	I/O	Slave transmit data I/O
	SPI_SSLn0	I/O	Slave selection signal I/O
	SPI_SSLn1	Output	Slave selection signal output
	SPI_SSLn2	Output	Slave selection signal output
	SPI_SSLn3	Output	Slave selection signal output

Note: For convention, pin names within this chapter are written without "SPI_" notation except this table.

34.2 Register Descriptions

34.2.1 SPDR/SPDR_HA/SPDR_BY : SPI Data Register

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

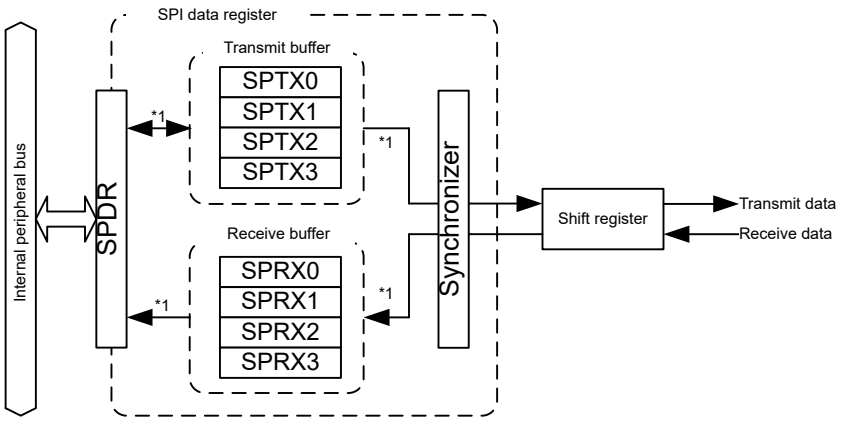
Offset address: 0x00

Bit position:	31																												0
Bit field:	SPD[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	SPD[31:0]	The SPI data register (SPDR) is used to store SPI's transmit data and receive data. Transmit buffers and receive buffers independently function.	R/W

SPDR is the interface with the buffers that hold data for transmission and reception by the SPI.

Figure 34.2 shows the structure of SPDR.



Note 1. The destination buffer and stage for access is automatically switched by hardware.

Figure 34.2 Structure of SPDR

32 bit × 4 stage transmit FIFO and 32 bit × 4 receive FIFO are provided. These 8 stage FIFO are mapped to one address in the SPDR. Transmit buffers (SPTXn, n = 0 to 3) can be written by writing data to SPDR to transmit written data.

Upon completion of receiving data, receive buffers store received data. When an overrun error occurs, data in the receive buffer is not updated.

Bus Interface

The SPI data register has 32 bit × 4 stage transmit FIFO and 32 bit × 4 receive FIFO (32 bytes in total). These 32 bytes are mapped to the 4-byte space of SPDR. Write transmit data from the LSB. Received data is stored from the LSB.

SPDR register write operation and read operation are described below.

(a) Write

A transmit buffer write pointer is provided for transmit buffers. When data is written to SPDR, the pointer automatically switches to the next buffer. The following illustrates the structure of the transmit buffer bus interface (write).

Figure 34.3 shows the configuration of the bus interface with the transmission buffer in the case of writing to SPDR.

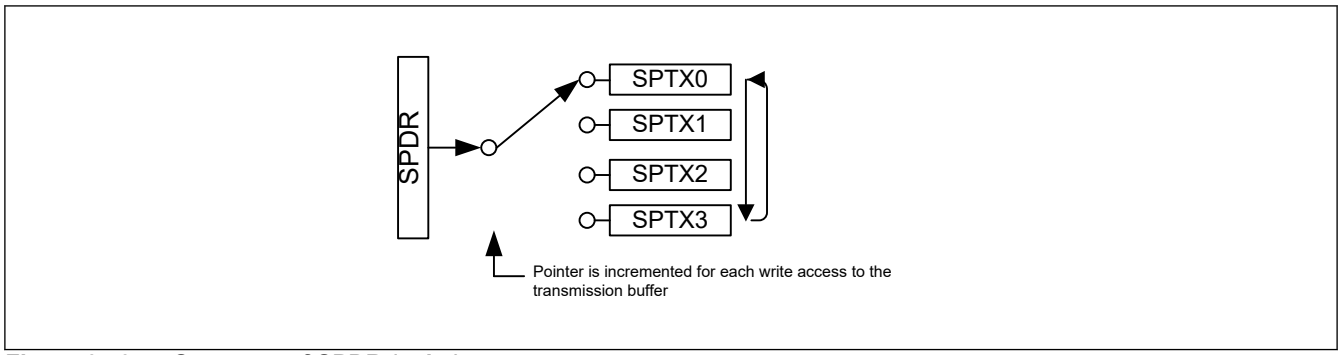


Figure 34.3 Structure of SPDR (write)

The transmit buffer (SPTX0 to SPTX3) switching order:

SPTX0→SPTX1→SPTX2→SPTX3→SPTX0→SPTX1→…

When writing transmit data to transmit buffers (SPTXn), write transmit data of frames +1 specified by the Transmission FIFO threshold setting bits of SPI data control register 2 (SPDCR2.TTRG[1:0]) while an SPI transmit buffer empty interrupt is present (SPSR.SPTEF flag = 1). Writing to the transmit buffer (SPTXn, n = 0 to 3) in the state where there is no empty stage in the transmit FIFO does not update the buffer value.

(b) Read

Values can be read from receive buffers (SPRXn, n = 0 to 3) or transmit buffers (SPTXn, n = 0 to 3) by reading the SPDR register. Reading a receive buffer or reading a transmit buffer can be selected by the SPI receive data or transmit data select bit (SPDCR.SPRDTD).

The SPDR register is read according to the independent receive buffer read pointer and the transmit buffer read pointer.

The following illustrates the structure of the receive buffer and transmit buffer bus interface (read).

Figure 34.4 shows the configuration of the bus interface with the receive and transmission buffers in the case of reading from SPDR.

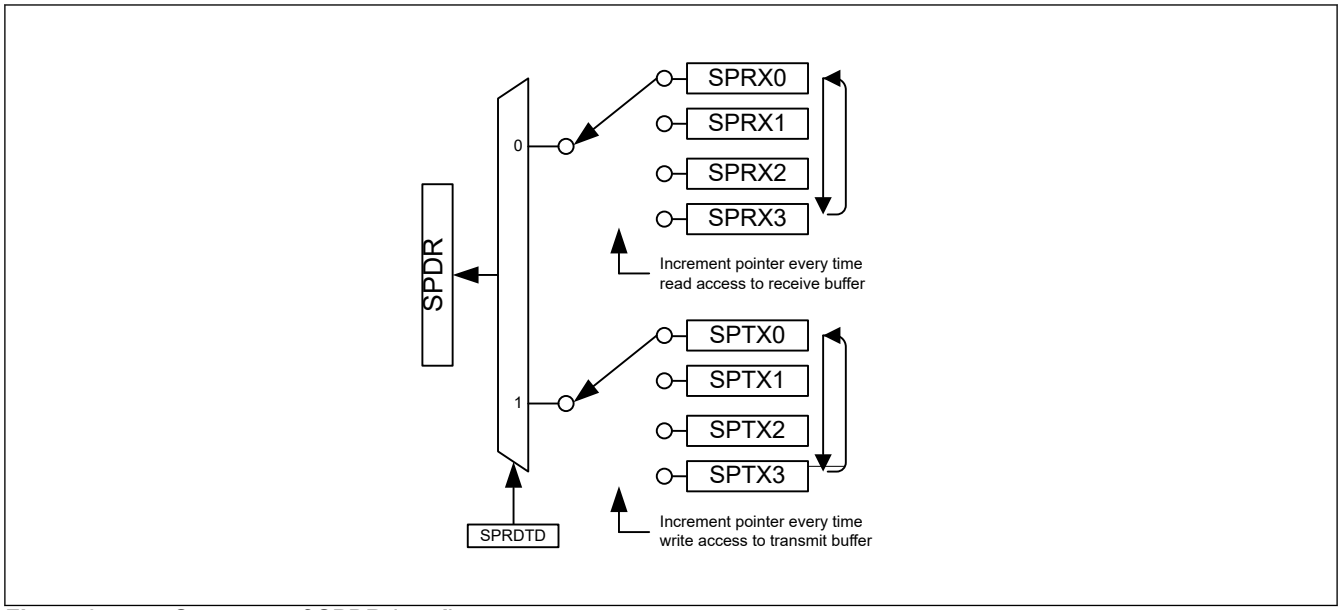


Figure 34.4 Structure of SPDR (read)

When a receive buffer is read, the receive buffer read pointer automatically switches to the next buffer. The receive buffer read pointer switches in the same order as the transmit buffer write pointer.

The transmit buffer read pointer is updated during the SPDR write access, but it is not updated during the transmit buffer read access. When a transmit buffer is read, the value written to SPDR last can be read.

34.2.2 SPCKD : SPI Clock Delay Register

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SCKDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SCKDL[2:0]	RSPCK Delay Setting 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

[In the Motorola-SPI case]

The SPCKD register is used to set the period (RSPCK delay) from SSL signal assertion start until RSPCK oscillates while SPCMD.SCKDEN = 1. If SPCKD is modified while SPCR.MSTR = 1 and SPCR.SPE = 1, subsequent operation is not guaranteed.

[In the TI-SSP case]

The SPCKD register is used to set the period (RSPCK delay) from SSL signal assertion start until RSPCK oscillates while SPCMD.SCKDEN = 1. Also that is used to set the period until the SSL signal is negated. If SPCKD is modified while SPCR.MSTR = 1 and SPCR.SPE = 1, subsequent operation is not guaranteed.

SCKDL[2:0] bits (RSPCK Delay Setting)

These bits are used to set the RSPCK delay value when SPCMD.SCKDEN = 1.

To use the SPI in slave mode, set SCKDL[2:0] bits to 000b.

34.2.3 SSLND : SPI Slave Select Negation Delay Register

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x05

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SLNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SLNDL[2:0]	SSL Negation Delay Bits This is the synchronization delay for replacing the RSPCK input with PCLKSPIn. RSPCK Delay Setting 0 0 0: 1 RSPCK [Master mode] 1 PCLKSPIn + (2~3 PCLKSPIn (Synchronization delay)) [TI-SSP case in Slave Mode] 0 0 1: 2 RSPCK [Master mode] 2 PCLKSPIn + (2~3 PCLKSPIn (Synchronization delay)) [TI-SSP case in Slave Mode] 0 1 0: 3 RSPCK [Master mode] 3 PCLKSPIn + (2~3 PCLKSPIn (Synchronization delay)) [TI-SSP case in Slave Mode] 0 1 1: 4 RSPCK [Master mode] 4 PCLKSPIn + (2~3 PCLKSPIn (Synchronization delay)) [TI-SSP case in Slave Mode] 1 0 0: 5 RSPCK [Master mode] 5 PCLKSPIn + (2~3 PCLKSPIn (Synchronization delay)) [TI-SSP case in Slave Mode] 1 0 1: 6 RSPCK [Master mode] 6 PCLKSPIn + (2~3 PCLKSPIn (Synchronization delay)) [TI-SSP case in Slave Mode] 1 1 0: 7 RSPCK [Master mode] 7 PCLKSPIn + (2~3 PCLKSPIn (Synchronization delay)) [TI-SSP case in Slave Mode] 1 1 1: 8 RSPCK [Master mode] 8 PCLKSPIn + (2~3 PCLKSPIn (Synchronization delay)) [TI-SSP case in Slave Mode]	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

[In the Motorola-SPI case]

The SSLND register is used to set the period (SSL negation delay) after the SPI in master mode sends the final RSPCK edge during serial transfer until it negates the SSL signal while SPCMD.SCKDEN = 1. If SSLND is modified while SPCR.MSTR = 1 and SPCR.SPE = 1, subsequent operation is not guaranteed.

[In the TI-SSP case]

The SSLND register is used to set the period (Output disable delay) after the SPI in master mode sends the final RSPCK edge during serial transfer until it negates the Output enable signal while SPCMD.SCKDEN = 1. Also, that is used to set the period from when the SPI in slave mode detects the last RSPCK edge of serial transfer to when the Output enable signal is negated. If SSLND is modified while the SPE bit in SPCR.SPE = 1, subsequent operation is not guaranteed.

SLNDL[2:0] bits (SSL Negation Delay Bits)

[In the Motorola-SPI case]

These bits are used to set the SSL negation delay value when SPCMD.SLNDEN = 1.

To use the SPI in slave mode except TI-SSP, set SLNDL[2:0] bits to 000b.

[In the TI-SSP case]

These bits are used to set the Output disable delay value when SPCMD.SLNDEN = 1.

34.2.4 SPND : SPI Next-Access Delay Register

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SPNDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SPNDL[2:0]	SPI Next-Access Delay Bits RSPCK Delay Setting 0 0 0: 1 RSPCK + 5 PCLKSPIn 0 0 1: 2 RSPCK + 5 PCLKSPIn 0 1 0: 3 RSPCK + 5 PCLKSPIn 0 1 1: 4 RSPCK + 5 PCLKSPIn 1 0 0: 5 RSPCK + 5 PCLKSPIn 1 0 1: 6 RSPCK+ 5 PCLKSPIn 1 1 0: 7 RSPCK + 5 PCLKSPIn 1 1 1: 8 RSPCK + 5 PCLKSPIn	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The SPND register is used to set the SSL signal inactive period (next-access delay) after completion of serial transfer while SPCMD.SCKDEN = 1. If SPND is modified while SPCR.MSTR = 1 and SPCR.SPE = 1, subsequent operation is not guaranteed.

SPNDL[2:0] bits (SPI Next-Access Delay Bits)

These bits are used to set the next-access delay value when SPCMD.SCKDEN = 1. To use the SPI in slave mode, set SPNDL[2:0] bits to 000b.

34.2.5 MRCKD : SPI ClocK Digital control Register for Master Receive

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x07

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	ARST[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	ARST[2:0]	Receive Sampling Timing Adjustment Bits The sampling of the serial data (MISO) input in master mode can be adjusted by delaying 0 to 7 PCLKSPIn from the center of the bit. 0 0 0: 0 PCLKSPIn delay 0 0 1: 1 PCLKSPIn delay 0 1 0: 2 PCLKSPIn delay 0 1 1: 3 PCLKSPIn delay 1 0 0: 4 PCLKSPIn delay 1 0 1: 5 PCLKSPIn delay 1 1 0: 6 PCLKSPIn delay 1 1 1: 7 PCLKSPIn delay	R/W

Bit	Symbol	Function	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The MRCKD register is a register for adjusting the reception sampling timing in master mode. If the ARST[2:0] bits are rewritten while SPCR.SPE = 1, subsequent operations are not guaranteed.

ARST[2:0] bits (Receive Sampling Timing Adjustment Bits)

These bits are used to select the amount of reception sampling timing adjustment. The sampling timing can be adjusted by delaying 0 to 7 PCLKSPIn from the center of the bit.

Delay setting exceeding the dividing ratio is prohibited.

This bit setting is valid only when the loopback function is not used (when SPPCR.SPLP = 0, when MRCLK is selected (SPCR.SPSCKSEL = 1), and SPPCR.SPLP2 = 0) and in master mode. For details, see SPSCKSEL bit of section 34.2.6. SPCR : SPI Control Register.

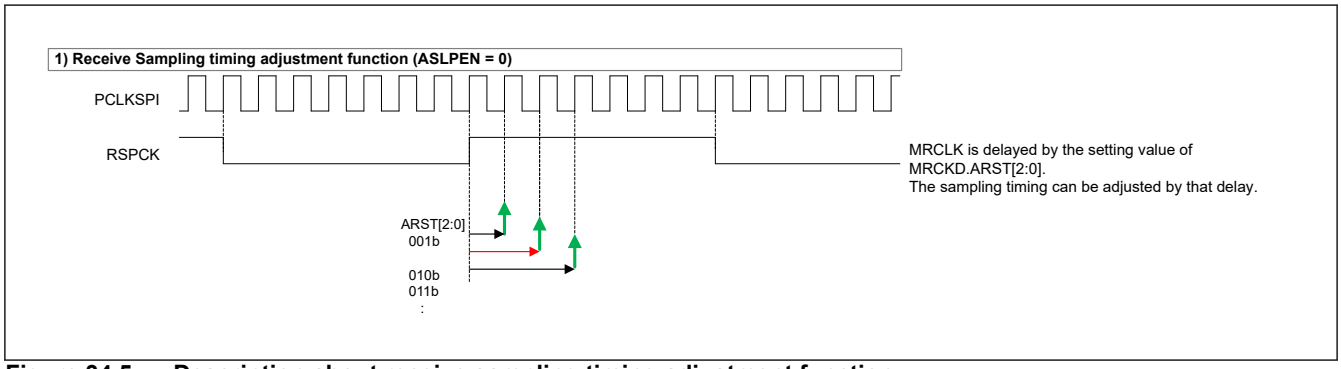


Figure 34.5 Description about receive sampling timing adjustment function

34.2.6 SPCR : SPI Control Register

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	BPEN	MSTR	TXMD[1:0]		—	—	SPFRF	SPMS	—	—	CENDIE	SPTIE	SPDRES	SPIIE	SPRIE	SPEIE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	MODFEN	BFDS	SCKASE	PTE	—	SPOE	SPPE	SPSCKSEL	—	—	—	—	—	—	SPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPE	SPI Function Enable 0: SPI function is disabled. 1: SPI function is enabled.	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	SPSCKSEL	SPI Master Receive Clock Select 0: MRIOCLK (Adjust with Analog Delay (SPCR2.SPSCKDL)) 1: MRCLK (Adjust with Digital Delay (MRCKD.ARST))	R/W
8	SPPE	Parity Enable 0: A parity bit is not added to transmit data. Received-data parity check is not performed. 1: A parity bit is added to transmit data. Received-data parity check is performed.	R/W

Bit	Symbol	Function	R/W
9	SPOE	Parity Mode 0: Even parity is used for transmission and reception. 1: Odd parity is used for transmission and reception.	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	PTE	Parity Self-Diagnosis Enable 0: Parity circuit self-diagnosis function is disabled. 1: Parity circuit self-diagnosis function is enabled.	R/W
12	SCKASE	RSPCK Auto-Stop Function Enable 0: RSPCK auto-stop function is disabled. 1: RSPCK auto-stop function is enabled.	R/W
13	BFDS	Between Burst Transfer Frames Delay Select 0: Delay (RSPCK delay, SSL negation delay and next-access delay) between frames is inserted in burst transfer. 1: Delay between frames is not inserted in burst transfer.	R/W
14	MODFEN	Mode Fault Error Detection Enable 0: Mode fault error detection is disabled. 1: Mode fault error detection is enabled.	R/W
15	—	This bit is read as 0. The write value should be 0.	R/W
16	SPEIE	SPI Error Interrupt Enable 0: SPI error interrupt request is disabled. 1: SPI error interrupt request is enabled.	R/W
17	SPRIE	SPI Receive Buffer Full Interrupt Enable 0: SPI receive buffer full interrupt request is disabled. 1: SPI receive buffer full interrupt request is enabled.	R/W
18	SPIIE	SPI Idle Interrupt Enable 0: Idle interrupt request is disabled. 1: Idle interrupt request is enabled.	R/W
19	SPDRES	SPI Receive Data Ready Error Select 0: Receive data full interrupt 1: Error interrupt	R/W
20	SPTIE	SPI Transmit Buffer Empty Interrupt Enable 0: SPI transmit buffer empty interrupt request is disabled. 1: SPI transmit buffer empty interrupt request is enabled.	R/W
21	CENDIE	SPI Communication End Interrupt Enable 0: Communication end interrupt request is disabled. 1: Communication end interrupt request is enabled.	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
24	SPMS	SPI Function Enable 0: SPI operation (4-wire) 1: Clock synchronous operation (3-wire)	R/W
25	SPFRF	SPI Frame Format Select When SPMS = 1 (clock synchronous operation (3-wire)), this bit setting is invalid. 0: Motorola-SPI 1: TI-SSP	R/W
27:26	—	These bits are read as 0. The write value should be 0.	R/W
29:28	TXMD[1:0]	Communication Mode Select 0 0: Transmit-Receive 0 1: Transmit only 1 x: Receive only	R/W
30	MSTR	SPI Master/Slave Mode Select 0: Slave mode 1: Master mode	R/W
31	BPEN	Synchronization Circuit Bypass Enable 0: Non-Bypass 1: Bypass	R/W

The SPCR register is used to set operating mode of the SPI. If the set MSTR, TXMD[1:0], SPFRF, SPMS, MODFEN, BFDS, SCKASE, PTE, SPOE, SPPE, SPSCKSEL, and BPEN bit value is modified while SPCR.SPE = 1, subsequent operation is not guaranteed.

SPE bit (SPI Function Enable)

The SPE bit is used to enable or disable SPI functions. Setting this bit to 1 enables SPI functions. When SPSR.MODF = 1, the SPCR.SPE bit is cleared to 0 and the SPCR.SPE bit cannot be set to 1 until the MODF flag is cleared to 0. (See section 34.3.10. Error Detection.) Setting the SPE bit to 0 disables SPI functions and initializes a part of module functions. (See section 34.3.12. SPI Initialization.)

SPSCKSEL bit (SPI Master Receive Clock Select)

The SPSCKSEL bit selects the clock used to sample the received data in master mode. By setting this bit, the sampling timing of the MISO can be adjusted. This bit setting is valid only when the loopback function is not used (when SPPCR.SPLP = 0 and SPPCR.SPLP2 = 0) and in master mode.

Figure 34.6 shows a Master Receive Clock Delay configuration. By selecting this bit, you can select MRIOCLK (adjust with analog delay (SPCR2.SPSCKDL)) or MRCLK (adjust with digital delay (MRCKD.ARST)).

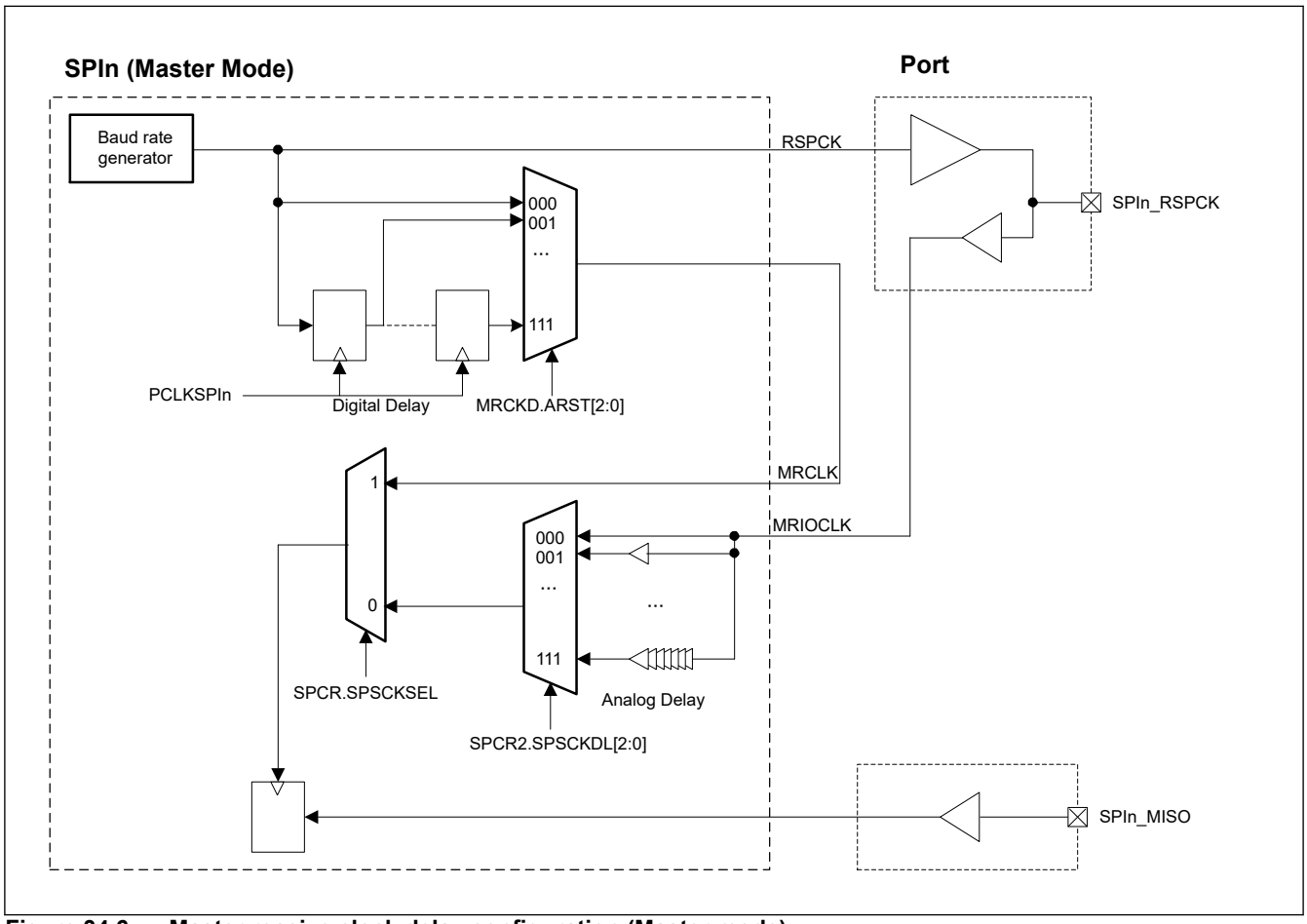


Figure 34.6 Master receive clock delay configuration (Master mode)

SPPE bit (Parity Enable)

The SPPE bit is used to enable or disable the parity function.

SPOE bit (Parity Mode)

The SPOE bit is used to specify even parity or odd parity.

In even parity mode, the parity bit is determined so that the sum of 1 (parity bit + transmit characters or receive characters) becomes an even number. In the same way, in odd parity mode, a parity bit is determined so that the sum of 1 (parity bit +

transmit characters or receive characters) becomes an odd number. The SPOE bit is valid only when the SPCR.SPPE is set to 1.

PTE bit (Parity Self-Diagnosis Enable)

The PTE bit is used to enable or disable self-diagnosis of the parity circuit to confirm that the parity function is normal.

SCKASE bit (RSPCK Auto-Stop Function Enable)

The SCKASE bit is used to enable or disable the RSPCK auto-stop function. When this function is enabled, the RSPCK clock stops immediately before an overrun error occurs during data reception in master mode. For details, see section 34.3.10.1. Overrun Error.

BFDS bit (Between Burst Transfer Frames Delay Select)

The BFDS bit controls whether insert the delay time between the burst transfer frames.

Valid in the master mode (SPCR.MSTR = 1) for frames with the SPCMDm.SSLKP bit set to 1.

This bit shold be set to 0 in slave mode. The usage of SSL delay control between transfer frames is shown as below.

For details, see (4)Burst Transfer in section 34.3.13.2. Slave Mode Operation.

(1) Non-burst transmits

(2) Burst transmit with delay between frames

(2-1) From the 1st frame to the last previous frame

(2-2) The last frame

(3) Burst transmit with no delay between frames

(3-1) From the 1st frame to the last previous frame

(3-2) The last frame

Table 34.3 Usage of SSL delay control between transfer frames (master mode)

	SPCMDm SSLKP bit	SPCR BFDS bit	SSL delay control register*1 (RSPCK clock delay, SSL negation delay, next access delay)
(1)	0	0	Any given value. You can control each delay value according to setting for RSPCK clock delay, SSL negation delay, and next access delay.
(2-1)	1	0	
(2-2)	0	0	
(3-1)	1	1	Any given value. But delay is inserted only below. • RSPCK clock delay of the 1st frame • SSL negation delay and next access delay of the last frame
(3-2)	0	1	

Note 1. Whether the setting value of following bits are valid or not depends on the setting value of the SPCMD.SPNDEN bit. (See section 34.2.14. SPCMDm : SPI Command Register m (m = 0 to 7).)
The SPCKD.SCKDL[2:0] bits: RSPCK delay
The SSLND.SLNDL[2:0] bits: SSL negate delay
The SPND.SPNDL[2:0] bits: Next access delay

< Setting / operation example > (Motorola SPI, BFDS = 1 Case)

SPCMD0.SSLKP = 1 → Burst transfer / no interframe delay between 0 and 1 (SSL keep active)

SPCMD1.SSLKP = 1 → Burst transfer / no interframe delay between 1 and 2 (SSL keep active)

SPCMD2.SSLKP = 1 → Burst transfer / no interframe delay between 2 and 3 (SSL keep active)

SPCMD3.SSLKP = 1 → Burst transfer / no interframe delay between 3 and 4 (SSL keep active)

SPCMD4.SSLKP = 0 → do not Burst Transfer, and once in-activate SSL.

(BFDS setting is invalid because it does not Burst Transfer.)

SPCMD5.SSLKP = 1 → Burst transfer / no interframe delay between 5 and 6 (SSL keep active)

SPCMD6.SSLKP = 1 → Burst transfer / no interframe delay between 6 and 7 (SSL keep active)

SPCMD7.SSLKP = 0 → do not Burst Transfer, and once in-activate SSL.

(BFDS setting is invalid because it does not Burst Transfer.)

MODFEN bit (Mode Fault Error Detection Enable)

The MODFEN bit is used to enable or disable detection of a mode fault error. (See section 34.3.10. Error Detection.) The SPI determines SSL0 pin input or output direction according to the combination of the MODFEN and MSTR bits. (See section 34.3.2. SPI Pin Control.)

SPEIE bit (SPI Error Interrupt Enable)

The SPEIE bit is used to enable or disable an SPI error interrupt request when the SPI detects a mode fault error or an underrun error and sets the MODF flag in the SPI status register (SPSR) to 1, when the SPI detects an overrun error and sets the OVRF flag in SPSR to 1, or when the SPI detects a parity error and sets the PERF flag in SPSR to 1. (See section 34.3.10. Error Detection.)

SPRIE bit (SPI Receive Buffer Full Interrupt Enable)

The SPRIE bit is used to enable or disable a receive buffer full interrupt request of the SPI.

SPIIE bit (SPI Idle Interrupt Enable)

The SPIIE bit is used to enable or disable an idle interrupt request of the SPI after the SPI detects the idle state and sets the IDLNF flag in the SPI status register (SPSR) to 0.

SPDRES bit (SPI Receive Data Ready Error Select)

When a receive data ready is detected (SPSR.SPDRF = 1), select whether to use SPRI interrupt request or SPEI interrupt request.

SPTIE bit (SPI Transmit Buffer Empty Interrupt Enable)

The SPTIE bit is used to enable or disable a transmit buffer empty interrupt request of the SPI.

A transmit buffer empty interrupt request at the beginning of transmission is generated by setting the SPE bit to 1 simultaneously when or after the SPTIE bit is set to 1. Note that a transmit buffer empty interrupt is generated while the SPTIE bit is 1 even though SPI functions are disabled (SPE bit = 0).

CENDIE bit (SPI Communication End Interrupt Enable)

The CENDIE bit controls generation of a communication end interrupt request.

SPMS bit (SPI Function Enable)

The SPMS bit is used to select SPI operation (4-wire) or clock synchronous operation (3-wire).

For clock synchronous operation, the SSL pin is not used but three pins RSPCK, MOSI, and MISO are used for communication. When SPMS = 1 (clock synchronous operation (3-wire)), the setting of the SPFRF bit is invalid.

To perform clock synchronous operation in master mode (SPCR.MSTR = 1), the SPCMD.CPHA bit can be set to 0 or 1. To perform clock synchronous operation in slave mode (SPCR.MSTR = 0), set the CPHA bit to 1. If this bit is set to 0 for clock synchronous operation in slave mode (SPCR.MSTR = 0), subsequent operation is not guaranteed.

The communication status according to the settings of SPCR.MSTR, SPCR.TXMD[1:0], SPCR.SPFRF, and SPCR.SPMS as follows.

Table 34.4 SPI communication status (1 of 2)

SPCR MSTR	SPCR TXMD[1]	SPCR TXMD[0]	SPCR SPFRF	SPCR SPMS	Communication Status	Communication Status No.
1	0	0	0	0	Transmit-Receive Master / Motorola SPI / SPI operation (4-wire)	1-(1)
1	0	0	1	0	Transmit-Receive Master / TI-SSP / SPI operation (4-wire)	1-(2)
1	0	0	—	1	Transmit-Receive Master / Clock synchronous operation (3-wire)	1-(3)
1	0	1	0	0	Transmit only Master / Motorola SPI / SPI operation (4-wire)	1-(4)
1	0	1	1	0	Transmit only Master / TI-SSP / SPI operation (4-wire)	1-(5)
1	0	1	—	1	Transmit only Master / Clock synchronous operation (3-wire)	1-(6)
1	1	—	0	0	Receive only Master / Motorola SPI / SPI operation (4-wire)	1-(7)

Table 34.4 SPI communication status (2 of 2)

SPCR MSTR	SPCR TXMD[1]	SPCR TXMD[0]	SPCR SPFRF	SPCR SPMS	Communication Status	Communication Status No.
1	1	—	1	0	Receive only Master / TI-SSP / SPI operation (4-wire)	1-(8)
1	1	—	—	1	Receive only Master / Clock synchronous operation (3-wire)	1-(9)
0	0	0	0	0	Transmit-Receive Slave / Motorola SPI / SPI operation (4-wire) (default)	0-(1)
0	0	0	1	0	Transmit-Receive Slave / TI-SSP / SPI operation (4-wire)	0-(2)
0	0	0	—	1	Transmit-Receive Slave / Clock synchronous operation (3-wire)	0-(3)
0	0	1	0	0	Transmit only Slave / Motorola SPI / SPI operation (4-wire)	0-(4)
0	0	1	1	0	Transmit only Slave / TI-SSP / SPI operation (4-wire)	0-(5)
0	0	1	—	1	Transmit only Slave / Clock synchronous operation (3-wire)	0-(6)
0	1	—	0	0	Receive only Slave / Motorola SPI / SPI operation (4-wire)	0-(7)
0	1	—	1	0	Receive only Slave / TI-SSP / SPI operation (4-wire)	0-(8)
0	1	—	—	1	Receive only Slave / Clock synchronous operation (3-wire)	0-(9)

SPFRF bit (SPI Frame Format Select)

The SPFRF bit selects the communication protocol.

The format of the SPI terminal (RSPCK, SSL0 to 3) can be set according to the set communication protocol.

When SPMS = 1 (clock synchronous operation (3-wire)), this bit is invalid because SSL is not used.

TXMD[1:0] bits (Communication Mode Select)

The TXMD[1:0] bits are used to select the transmit-receive, transmit-only, and receive-only serial communication.

TXMD[1:0] = 01b

→ Transmit-only is performed without reception.

→ Receive buffer full interrupt request cannot be used.

TXMD[1] = 1b

→ Receive-only is performed without transmission.

→ Transmit buffer empty interrupt request cannot be used.

For details, see section 34.3.6. Communications Operating Mode.

MSTR bit (SPI Master/Slave Mode Select)

The MSTR bit is used to select master mode or slave mode of the SPI. The SPI determines input/output directions of pins RSPCK, MOSI, MISO, and SSL1 to SSL3 according to the MSTR bit setting.

BPEN bit (Synchronization Circuit Bypass Enable)

When BPEN = 1, bus clock (PCLKM) is used as operation clock (PCLKSPIn) and synchronization circuit is bypassed.

34.2.7 SPCRRM : SPI Control Register for Master Receive only

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RMSTTG	RMEDTG	—	RMFM[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	RMFM[4:0]	Frame processing count setting in Master Receive only 0x00: This function is not used.*1 0x01: Automatically stop communication after processing 1 received frame. ⋮ 0x1F: Automatically stop communication after processing 31 received frames.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	RMEDTG	1: Receive End (writable only when in Master Receive) Reading value is always 0.	W
7	RMSTTG	1: Receive Start (writable only when in Master Receive) Reading value is always 0.	W

Note 1. See (9)Software Processing Flow in section 34.3.13.1. Master Mode Operation.

The SPCRRM register controls the start and completion of communication for master mode reception only operation. If the RMFM[4:0] bits are rewritten while SPCR.SPE = 1, subsequent operations are not guaranteed.

RMFM[4:0] bits (Frame processing count setting in Master Receive only)

The number of received frames can be adjusted when operating in master receive only. Valid only when the master mode (SPCR.MSTR = 1) and the communication operation mode select bits (SPCR.TXMD[1:0]) are 10b.

Only the start bit in master mode reception automatically stops communication after "starts frame processing according to the value set in this bit" after reception starts.

RMEDTG bit (1: Receive End (writable only when in Master Receive))

This bit is used to end reception when master receive only. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.TXMD[1:0]) are 10b.

RMSTTG bit (1: Receive Start (writable only when in Master Receive))

This bit is used to start reception when master receive only. Valid only when the master mode (SPCR.MSTR = 1) and the communication mode select bits (SPCR.TXMD[1:0]) are 10b.

Writing 1 to this bit during reception is not accepted. Write again after reception is completed.

34.2.8 SPDRCR : SPI Control Register for Received Data Ready Detection

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x0D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SPDRC[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SPDRC[7:0]	SPI Receive Data Ready Detect Adjustment 0x00: Disable receive data ready detection function 0x01: Performs reception data ready judgment after 1 PCLKSPIn ⋮ 0xFF: Performs reception data ready judgment after 255 PCLKSPIn	R/W

The SPDRCR register is used to set the SPI receive data ready detection function. If the set value is changed while SPCR.SPE = 1, subsequent operations are not guaranteed.

SPDRC[7:0] bits (SPI Receive Data Ready Detect Adjustment)

The receive data ready detection function can be disabled or, if used, the period until detection can be set from 1 to 255 PCLKSPIn.

The value set in the SPDRC[7:0] bits is used to 1 set the SPDRF flag. For details, see the description of SPDRF in section 34.2.18. SPSR : SPI Status Register.

34.2.9 SPPCR : SPI Pin Control Register

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x0E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	MOIFE	MOIFV	—	SPOM	SPLP2	SPLP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPLP	SPI Loopback 0: Normal mode 1: Loopback mode (data is inverted for transmission)	R/W
1	SPLP2	SPI Loopback 2 0: Normal mode 1: Loopback mode (data is not inverted for transmission)	R/W
2	SPOM	SPI Output Pin Mode 0: CMOS output 1: Open-drain output	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	MOIFV	MOSI Idle Fixed Value 0: The fixed value of MOSI idle = 0. 1: The fixed value of MOSI idle = 1.	R/W
5	MOIFE	MOSI Idle Value Fixing Enable 0: The MOSI output value is the last data of previous transfer. 1: The MOSI output value is the set MOIFV bit value.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

The SPPCR register is used to set pin mode of the SPI. If SPPCR is modified with the SPCR.SPE bit = 1, subsequent operation is not guaranteed.

SPLP bit (SPI Loopback)

When the SPLP bit is set to 1, the SPI shuts down the route between the MISO pin and the shift register (when SPCR.MSTR = 1) or shuts down the route between the MOSI pin and the shift register, inverts the input route value in the shift register, and then connects the route to the output route (when SPCR.MSTR = 0) (loopback mode).

SPLP2 bit (SPI Loopback 2)

When the SPLP2 bit is set to 1, the SPI shuts down the route between the MISO pin and the shift register (when SPCR.MSTR = 1) or shuts down the route between MOSI pin and the shift register and then connects the route to the output route without inverting the input route value in the shift register (when SPCR.MSTR = 0) (loopback mode). If this bit is set to 1 together with the SPLP bit, setting this bit takes precedence.

SPOM bit (SPI Output Pin Mode)

This bit is used to select CMOS output pin or open drain output pin as SPI's output pins.

MOIFV bit (MOSI Idle Fixed Value)

This bit is used to select the MOSI pin output value during the SSL negation period (including SSL retention period in burst transfer) when SPCR.MOIFE = 1 in master mode.

MOIFE bit (MOSI Idle Value Fixing Enable)

This bit is used for the SPI in master mode to fix the MOSI output value during the SSL negation period (including SSL retention period in burst transfer). When SPCR.MOIFE = 0, the SPI outputs the last data of the previous serial transfer to MOSI during the SSL negation period. When SPCR.MOIFE = 1, the SPI outputs the fixed MOIFV bit value to MOSI.

34.2.10 SPCR2 : SPI Control Register 2

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x0F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SPSCKDL[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SPSCKDL[2:0]	SPI Master Receive Clock Analog Delay Valid only in master mode 0 0 0: No delay 0 0 1: 1.1 ns (max) 0 1 0: 2.2 ns (max) 0 1 1: 3.3 ns (max) 1 0 0: 4.4 ns (max) 1 0 1: 5.5 ns (max) 1 1 0: 6.6 ns (max) 1 1 1: 7.7 ns (max)	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The SPCR2 register is used to set the analog delay of the master receive clock. If the value set in this register is changed while SPCR.SPE = 1, subsequent operations are not guaranteed.

SPSCKDL[2:0] bits (SPI Master Receive Clock Analog Delay)

These bits set the analog delay for MRIOCLK input to SPI in master mode. By setting these bits, the sampling timing of the received data can be adjusted. For details, see the SPSCKSEL bit of section 34.2.6. SPCR : SPI Control Register.

34.2.11 SSLP : SPI Slave Select Polarity Register

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x10

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SSL0P	SSL0 Signal Polarity Setting 0: In the Motorola-SPI case, the SSL0 signal is active low (0). In the TI-SSP case, the SSL0 signal is active high (1). 1: In the Motorola-SPI case, the SSL0 signal is active high (1). In the TI-SSP case, the SSL0 signal is active low (0).	R/W
1	SSL1P	SSL1 Signal Polarity Setting 0: In the Motorola-SPI case, the SSL1 signal is active low (0). In the TI-SSP case, the SSL1 signal is active high (1). 1: In the Motorola-SPI case, the SSL1 signal is active high (1). In the TI-SSP case, the SSL1 signal is active low (0).	R/W

Bit	Symbol	Function	R/W
2	SSL2P	SSL2 Signal Polarity Setting 0: In the Motorola-SPI case, the SSL2 signal is active low (0). In the TI-SSP case, the SSL2 signal is active high (1). 1: In the Motorola-SPI case, the SSL2 signal is active high (1). In the TI-SSP case, the SSL2 signal is active low (0).	R/W
3	SSL3P	SSL3 Signal Polarity Setting 0: In the Motorola-SPI case, the SSL3 signal is active low (0). In the TI-SSP case, the SSL3 signal is active high (1). 1: In the Motorola-SPI case, the SSL3 signal is active high (1). In the TI-SSP case, the SSL3 signal is active low (0).	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W

The SSLP register is used to set the polarity of SSL0 to SSL3 signals of the SPI. If any of these SSLP bits is modified with the SPCR.SPE bit = 1, subsequent operation is not guaranteed.

SSLnP bits (SSLn Signal Polarity Setting)

These bits are used to specify the polarity of SSL signals. The set SSLnP bit (n = 3 to 0) values indicate the active polarity of SSLn signals.

SSL0 is different from SSL1, SSL2, and SSL3. In slave or multi-master mode, it functions as an input.

For details, see section 34.3.3.2. Single Master/Single Slave (This LSI = Slave) and section 34.3.3.5. Multi-Master/Multi-Slave (with This LSI Acting as Master) .

34.2.12 SPBR : SPI Bit Rate Register

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x11

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SPR[7:0]							
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	SPR[7:0]	The SPBR register is used to set the bit rate in master mode. If SPBR is modified while SPCR.MSTR = 1 and SPCR.SPE = 1, subsequent operation is not guaranteed.	R/W

When the SPI is used in slave mode, the bit rate depends on the input clock bit rate regardless of the SPBR.BRDV setting. (Specify a bit rate that meets electrical characteristics.)

The bit rate is determined by a combination of the set SPBR and BRDV[1:0] in the SPCMDm register (SPCMD0 to SPCMD7).

The bit rate is calculated by the following expression, where "n" is the set SPBR value (0 to 255) and "N" is the set BRDV[1:0] bits value (0 to 3).

$$\text{Bit rate} = \frac{f(\text{PCLKSPIn})}{2 \times (n+1) \times 2^{N}}$$

Table 34.5 shows an example of correspondence between bit rates and set values of SPBR and BRDV[1:0].

Table 34.5 Corresponding between bit rates and set values (examples) (1 of 2)

SPBR (n)	BRDV[1:0] bits (N)	Division ratio	Bit rate		
			PCLKSPIn = 75 MHz	PCLKSPIn = 96 MHz	PCLKSPIn = 100 MHz
0	0	2	37.50 Mbps	48.0 Mbps	50.0 Mbps
1	0	4	18.75 Mbps	24.0 Mbps	25.0 Mbps
2	0	6	12.50 Mbps	16.0 Mbps	16.7 Mbps

Table 34.5 Corresponding between bit rates and set values (examples) (2 of 2)

SPBR (n)	BRDV[1:0] bits (N)	Division ratio	Bit rate		
			PCLKSPIn = 75 MHz	PCLKSPIn = 96 MHz	PCLKSPIn = 100 MHz
3	0	8	9.38 Mbps	12.0 Mbps	12.5 Mbps
4	0	10	7.50 Mbps	9.6 Mbps	10.0 Mbps
5	0	12	6.25 Mbps	8.0 Mbps	8.33 Mbps
5	1	24	3.13 Mbps	4.0 Mbps	4.17 Mbps
5	2	48	1.56 Mbps	2.0 Mbps	2.08 Mbps
5	3	96	781 kbps	1.0 Mbps	1.04 Mbps
255	3	4096	18.3 kbps	23.4 kbps	24.4 kbps

34.2.13 SPSCR : SPI Sequence Control Register

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x13

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SPSLN[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SPSLN[2:0]	SPI Sequence Length Specification The order in which the SPCMD0 to SPCMD07 registers are to be referenced is changed in accordance with the sequence length that is set in these bits. The relationship among the setting of these bits, the sequence length, and the SPCMD0 to SPCMD7 register numbers referenced by SPI is as follows. However, the SPI in slave mode always refers to SPCMD0. 0 0 0: Sequence Length = 1 (0→0→⋯) 0 0 1: Sequence Length = 2 (0→1→0→⋯) 0 1 0: Sequence Length = 3 (0→1→2→0→⋯) 0 1 1: Sequence Length = 4 (0→1→2→3→0→⋯) 1 0 0: Sequence Length = 5 (0→1→2→3→4→0→⋯) 1 0 1: Sequence Length = 6 (0→1→2→3→4→5→0→⋯) 1 1 0: Sequence Length = 7 (0→1→2→3→4→5→6→0→⋯) 1 1 1: Sequence Length = 8 (0→1→2→3→4→5→6→7→0→⋯)	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

The SPSCR register is used to set the sequence length for the SPI to perform master operation.

Before modifying the SPSLN[2:0] bits in SPSCR while SPCR.MSTR = 1 and SPCR.SPE = 1, confirm that SPSR.IDLNF = 0.

SPSLN[2:0] bits (SPI Sequence Length Specification)

These bits are used to set the sequence length for the SPI in master mode to perform sequence operation.

According to the sequence length specified by SPSLN[2:0] bits, the SPI in master mode changes the SPCMDm registers (SPCMD0 to SPCMD7) to be referenced and the reference sequence. For details, see (3)Sequence Control.

The SPI in slave mode always references SPCMD0.

34.2.14 SPCMDm : SPI Command Register m (m = 0 to 7)

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x14 + 0x04 × m

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	SSLA[1:0]		—	—	—	SPB[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SCKDEN	SLNDEN	SPNDEN	LSBF	—	—	—	—	SSLKP	—	—	—	BRDV[1:0]		CPOL	CPHA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CPHA	RSPCK Phase This register is valid only for Motorola-SPI cases. (In the TI-SSP case, it is fixed at 1). 0: Data is sampled at an odd edge and changes at an even edge. 1: Data changes at an odd edge and is sampled at an even edge.	R/W
1	CPOL	RSPCK Polarity 0: RSPCK in idle state is 0. 1: RSPCK in idle state is 1.	R/W
3:2	BRDV[1:0]	Bit Rate Division 0 0: The base bit rate is selected. 0 1: Two-divided base bit rate is selected. 1 0: Four-divided base bit rate is selected. 1 1: Eight-divided base bit rate is selected.	R/W
6:4	—	These bits are read as 0. The write value should be 0.	R/W
7	SSLKP	SSL Signal Level Hold 0: All SSL signals are negated at the end of transfer. 1: SSL signal level is held after the transfer ends until the next access starts.	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
12	LSBF	SPI LSB First 0: MSB first 1: LSB first	R/W
13	SPNDEN	SPI Next-Access Delay Enable 0: Next-access delay is 1 RSPCK + 5 PCLKSPIn 1: Next-access delay is the set value of the SPI next-access delay register (SPND).	R/W
14	SLNDEN	SSL Negation Delay Setting Enable 0: [Master] SSL negation delay is 1 RSPCK. [Slave in the TI-SSP] SSL negation delay is 1 PCLKSPIn + 2~3 PCLKSPIn (Synchronization delay) 1: SSL negation delay is the set value of the slave select negation delay register (SSLND).	R/W
15	SCKDEN	RSPCK Delay Setting Enable 0: RSPCK delay is 1 RSPCK. [In the Motorola-SPI case] RSPCK delay is 0 RSPCK. [In the TI-SSP case] 1: RSPCK delay is the set value of the RSPCK delay register (SPCKD).	R/W

Bit	Symbol	Function	R/W
20:16	SPB[4:0]	SPI Data Length 0x00: Setting prohibited 0x01: Setting prohibited 0x02: Setting prohibited 0x03: 4 bits 0x04: 5 bits 0x05: 6 bits ⋮ 0x1E: 31 bits 0x1F: 32 bits	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
25:24	SSLA[1:0]	SSL Signal Assertion 0 0: SSL0 0 1: SSL1 1 0: SSL2 1 1: SSL3	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

The SPI has eight SPI command registers (SPCMD0 to SPCMD7) that are used to set the transfer format of the SPI in master mode. Furthermore, some bits in SPCMD0 are used to set the transfer format of the SPI in slave mode.

The SPI in master mode sequentially references SPCMD0 to SPCMD7 according to the setting of SPSCR.SPSLN[2:0], and then performs serial transfer specified in the referenced SPCMD.

SPCMD should be set while the transmit buffer is empty (next-transfer data is not set) and data which refers to SPCMD has not been written.

The SPCMD referenced by the SPI in master mode is indicated by SPCP[2:0] in the SPSSR register. If SPCMD0 is modified while the SPI in slave mode is enabled (SPCR.SPE = 1), subsequent operation is not guaranteed.

CPHA bit (RSPCK Phase)

This bit is used to set the RSPCK phase of the SPI in master mode or slave mode. To perform data communication between SPI modules, the same RSPCK phase must be set for both modules.

When SPCR.SPMS = 0 and SPCR.SPFRF = 1 (in TI SSP mode), setting CPHA = 0 is invalid.

This register is valid only for Motorola-SPI cases.

CPOL bit (RSPCK Polarity)

This bit is used to set the RSPCK polarity of the SPI in master mode or slave mode. To perform data communication between SPI modules, the same RSPCK polarity must be set for both modules.

BRDV[1:0] bits (Bit Rate Division)

This register is used to determine the bit rate with a combination of the set values of the BRDV[1:0] bits and the SPBR register. The set SPBR value determines the base bit rate. The set BRDV[1:0] bits value is used to select undivided, 2-divided, 4-divided, or 8-divided base bit rate. SPCMD0 to SPCMD7 enable setting of different BRDV[1:0] values. This makes it possible to perform serial transfer with a different bit rate for each command.

SSLKP bit (SSL Signal Level Hold)

This bit is used to set whether to hold or negate the SSL signal level of the current command during a period from the SSL negation timing for the current command to the SSL assertion timing for the next command when the SPI in master mode performs serial transfer. Setting this bit to 1 enables burst transfer in SPI operation master mode. For details, see (4)Burst Transfer in section 34.3.13.1. Master Mode Operation.

To use the SPI in slave mode, set SSLKP bit to 0.

LSBF bit (SPI LSB First)

This bit is used to set the data format of the SPI in master mode or slave mode to MSB first or LSB first.

SPNDEN bit (SPI Next-Access Delay Enable)

This bit is used to set the period from SSL negation to SSL assertion in SPI master mode. When SPNDEN = 0, the SPI sets the next-access delay to 1 RSPCK + 5 PCLKSPIn. When SPNDEN = 1, the SPI inserts the next-access delay in accordance with the SPND register setting.

To use the SPI in slave mode, set SPNDEN bit to 0.

SLNDEN bit (SSL Negation Delay Setting Enable)

[In the Motorola-SPI case]

This bit is used to set the period from RSPCK oscillation stop to SSL negation in SPI master mode. When SLNDEN = 0, the SPI sets the SSL negation delay to 1 RSPCK. When SLNDEN = 1, the SPI negates the SSL signal with the RSPCK delay in accordance with the SSLND register setting.

To use the SPI in slave mode, set SLNDEN bit to 0.

[In the TI-SSP case]

This bit is used to set the period from RSPCK oscillation stop to output disable in SPI master mode, or last edge of RSPCK to output disable in slave mode. When the SLNDEN bit is 0, the SSL negate delay is 1 RSPCK in master mode and 1 PCLKSPIn + 2~3 PCLKSPIn (Synchronization delay) in slave mode. When SLNDEN = 1, the SPI negates the SSL signal with the RSPCK delay in accordance with the SSLND register setting.

When using SPI in slave mode except TI SSP setting, set the SLNDEN bit to 0.

SCKDEN bit (RSPCK Delay Setting Enable)

[In the Motorola-SPI case]

This bit is used to set the period (RSPCK delay) after the SPI in master mode activates the SSL signal until it oscillates RSPCK. When SCKDEN = 0, the SPI sets the RSPCK delay to 1 RSPCK. When SCKDEN = 1, the SPI starts RSPCK oscillation with the RSPCK delay in accordance with the SPCKD register setting.

To use the SPI in slave mode, set SCKDEN bit to 0.

[In the TI-SSP case]

This bit is used to set the period from the start of assertion of the SSL signal to the RSPCK oscillation (RSPCK delay) and the period of the SSL signal to negation by the SPI in master mode. When SCKDEN = 0, the SPI does not sets the RSPCK delay. When SCKDEN = 1, the SPI starts RSPCK oscillation with the RSPCK delay in accordance with the SPCKD register setting.

To use the SPI in slave mode, set SCKDEN bit to 0.

SPB[4:0] bits (SPI Data Length)

These bits are used to set the transfer data length of the SPI in master mode or slave mode.

SSLA[1:0] bits (SSL Signal Assertion)

These bits are used to control SSL signal assertion for the SPI in master mode to perform serial transfer. The set SSLA[1:0] bits value controls assertion of the SSL3 to SSL0 signals. The signal polarity when the SSL signal is asserted depends on the set value of the SSLP register. When SSLA[1:0] bits are set to 00b in multi-master mode, serial transfer is performed with all SSL signals negated (because SSL0 is input).

To use the SPI in slave mode, set SSLA[1:0] bits to 00b.

34.2.15 SPDCR : SPI Data Control Register

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x40

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SPFC[1:0]		—	—	—	SINV	SPRDTD	SLSEL[1:0]		BYSW
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BYSW	Byte Swap Operating Mode Select 0: Byte swap is disabled 1: Byte swap is enabled	R/W
2:1	SLSEL[1:0]	SSL Pin Output Select See Table 34.6.	R/W
3	SPRDTD	SPI Receive Data or Transmit Data Selection 0: The SPDR reads the receive buffer 1: The SPDR reads the transmit buffer	R/W
4	SINV	Serial data invert 0: Not invert serial data 1: Invert serial data	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W
9:8	SPFC[1:0]	Frame Count Number of Frames Specification 0 0: 1 frame 0 1: 2 frames 1 0: 3 frames 1 1: 4 frames	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

The SPDCR register controls the data format.

If the value set in this register is changed while SPCR.SPE = 1, subsequent operations are not guaranteed.

BYSW bit (Byte Swap Operating Mode Select)

It is a setting bit, that is to swap a transmit/receive data in byte units. A data after byte swap is different by a data length (setting of SPCMD.SPB[4:0]).

When byte swap, A data length (setting of SPB[4:0]) must be set to 32 bits or 16 bits. Other case of data length (for example, 4 to 15, 17 to 31 bit length), byte swap is not guaranteed. For the arrangement of data before and after swapping data lengths of 32 bits and 16 bits, see section 34.3.4. Data Format and section 34.3.4.5. Byte swap reception.

When the parity function set to valid, the behavior is not guaranteed.

SLSEL[1:0] bits (SSL Pin Output Select)

The SLSEL[1:0] bits are used to select SSL output or I/O as an SSL pin function in master mode.

Table 34.6 SSL pin output selection

SSLn pin	SLSEL[1:0] = 00b	SLSEL[1:0] = 01b	SLSEL[1:0] = 10b	SLSEL[1:0] = 11b
SSL0	Output/Input	Output/Input	Output/Input	Setting prohibited
SSL1	Output	I/O	Output	
SSL2, SSL3	Output	I/O	I/O	

Note: Input or output of SSL0 is determined by the SPCR.MODFEN bit. For details, see Table 34.7.

SPRDTD bit (SPI Receive Data or Transmit Data Selection)

The SPRDTD bit is used to select receive buffer or transmit buffer from which the SPI data register (SPDR) value is read.

When the transmit buffer is read, the value written to SPDR the last time is read.

SINV bit (Serial data invert)

The SINV bit is used to invert transmit data and receive data.

When the SINV bit is set to 1, transmit buffer (SPTX) data is inverted to invert transmit data and receive data, and then the inverted data is stored in the receive buffer (SPRX). The parity bit is the value corresponding to the inverted transmission/reception data.

SPFC[1:0] bits (Frame Count)

The SPFC[1:0] bits are used for the condition to set the CENDF flag in slave receive only mode. For details on the CENDF flag setting conditions, see section 34.2.18. SPSR : SPI Status Register.

The SPFC[1:0] bits are invalid except in the slave receive only mode.

34.2.16 SPDCR2 : SPI Data Control Register 2

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x44

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	TTRG[1:0]		—	—	—	—	—	—	RTRG[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	RTRG[1:0]	Receive FIFO threshold setting When using DMA, set the threshold to 0 for RTRG and TTRG bits. 0 0: Threshold 0 0 1: Threshold 1 1 0: Threshold 2 1 1: Threshold 3	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
9:8	TTRG[1:0]	Transmission FIFO threshold setting When using DMA, set the threshold to 0 for RTRG and TTRG bits. 0 0: Threshold 0 0 1: Threshold 1 1 0: Threshold 2 1 1: Threshold 3	R/W
15:10	—	These bits are read as 0. The write value should be 0.	R/W

The SPDCR2 register controls the FIFO threshold. If the value set in this register is changed while SPCR.SPE = 1, subsequent operations are not guaranteed.

RTRG[1:0] bits (Receive FIFO threshold setting)

Set the receive FIFO threshold.

When the number of data stored in the receive FIFO > the number of frames set by RTRG[1:0], the receive buffer full flag is set.

TTRG[1:0] bits (Transmission FIFO threshold setting)

Set the transmit FIFO threshold.

When the number of empty stages in the transmit FIFO > the number of frames set in TTRG[1:0], the transmit buffer empty flag is set.

34.2.17 SPSSR : SPI Sequence Status Register

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x51

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	SPECM[2:0]			—	SPCP[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SPCP[2:0]	SPI Command Pointer 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
3	—	This bit is read as 0.	R
6:4	SPECM[2:0]	SPI Error Command 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
7	—	This bit is read as 0.	R

The SPSSR register indicates the sequence control status when the SPI performs master operation. Write access to SPSSR is ignored.

SPCP[2:0] bits (SPI Command Pointer)

These bits show the SPCMDm registers 0 to 7 (SPCMD0 to SPCMD7) indicated by the current pointer in the SPI sequence control. For details about the SPI sequence control, see section 34.3.13.1. Master Mode Operation.

SPECM[2:0] bits (SPI Error Command)

These bits show the SPCMDm registers 0 to 7 (SPCMD0 to SPCMD7) indicated by the command pointer (SPCP[2:0] bits) when an error was detected in the SPI sequence control. The SPI updates the SPECM[2:0] bits value only when an error is detected. When no error is present (SPSR.OVRF = 0, SPSR.MODF = 0, SPSR.PERF = 0), the SPECM[2:0] bits value has no meaning. For the SPI's error detection function, see section 34.3.10. Error Detection. For the SPI's sequence control, see section 34.3.13.1. Master Mode Operation.

34.2.18 SPSR : SPI Status Register

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x52

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SPRF	CENDF	SPTEF	UDRF	PERF	MODF	IDLNF	OVRF	SPDRF	—	—	—	—	—	—	—
Value after reset:	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0.	R
7	SPDRF	SPI Receive Data Ready Flag 0: Receive data ready not detected 1: Receive data ready detected	R
8	OVRF	Overrun Error Flag 0: No overrun error is present. 1: An overrun error is present.	R

Bit	Symbol	Function	R/W
9	IDLNF	SPI Idle Flag 0: The SPI is in the idle state. 1: The SPI is in the transfer state.	R
10	MODF	Mode Fault Error Flag 0: Neither mode fault error nor underrun error is present. 1: A mode fault error or underrun error is present.	R
11	PERF	Parity Error Flag 0: No parity error is present. 1: A parity error is present.	R
12	UDRF	Underrun Error Flag This bit indicates error status in combination with the MODF flag. The UDRF bit is valid when MODF flag is 1. 0: Mode fault error occurred (MODF = 1) 1: Underrun error occurred (MODF = 1)	R
13	SPTEF	SPI Transmit Buffer Empty Flag 0: The number of empty stages in the transmit FIFO ≤ the value set in SPDCR2.TTRG 1: The number of empty stages in the transmit FIFO > the value set in SPDCR2.TTRG	R
14	CENDF	Communication End Flag 0: The SPI is not communicating or communicating. 1: The SPI communication completed.	R
15	SPRF	SPI Receive Buffer Full Flag 0: The number of data stored in the receive FIFO ≤ number of frames set by the SPDCR2.RTRG bit. 1: The number of data stored in the receive FIFO > number of frames set by the SPDCR2.RTRG bit.	R

SPDRF bit (SPI Receive Data Ready Flag)

During communication (SPCR.SPE = 1), a certain period of time has elapsed while the number of data stored in the reception FIFO ≤ the reception FIFO threshold.

This bit is set to 0 when the reception operation is not performed (SPCR.TXMD[1:0] = 01b).

[Setting (to 1) condition]

All the following two conditions are met:

- SPDRCR.SPDRC[7:0] ≠ 0x00
- After the receive FIFO has been written, when the number of data stored in the receive FIFO ≤ the receive FIFO threshold and the value set by SPDRC[7:0] has elapsed

[Clearing (to 0) condition]

- When 1 is written to the SPSRC.SPDRFC bit

OVRF bit (Overrun Error Flag)

This flag indicates whether an overrun error is present. When the RSPCK clock auto-stop function is enabled (SPCR.SCKASE bit = 1) in master mode (SPCR.MSTR bit = 1), no overrun error occurs and, therefore, this flag is not set to 1. For details, see section 34.3.10.1. Overrun Error.

[Setting (to 1) condition]

When serial transfer is completed in one of the following two conditions with data stored in the receive FIFO for the number of FIFO stages.

- The SPCR.TXMD[1:0] bits = 00b. (transmit-receive mode)
- The SPCR.TXMD[1:0] bits = 10b. (receive only)

[Clearing (to 0) condition]

- When 1 is written to the SPSRC.OVRFC bit

IDLNF bit (SPI Idle Flag)

This flag indicates transfer status of the SPI.

[Setting (to 1) condition]

[Transmit-Receive, Transmit-only Master mode]

- None of the [Transmit-Receive, Transmit-only Master mode] in [Clearing (to 0) conditions]

[Receive-only Master mode]

- When 1 is written to SPCRRM.RMSTTG bit

[Slave mode]

- SPCR.SPE = 1 (SPI function enabled)

[Clearing (to 0) conditions]

Communication status: 1-(1) to (6) * For details of communication status, see Table 34.4.

[Transmit-Receive, Transmit-only Master mode]

Any of the following two conditions is met:

- SPCR.SPE = 0 (SPI initialization)
- All the following three conditions are met:
  - The next transfer data is not set in the transmission buffer (SPTXn, n = 0 to 3)
  - SPSSR.SPCP[2:0] = 000b (at the beginning of sequence control)
  - The operation completed until the next access delay (the master main state machine has transitioned to the idle state)

[Receive-only Master mode]

Communication status: 1-(7) to (9)

Any of the following two conditions is met:

- SPCR.SPE = 0 (SPI initialization)
- Any of the following three conditions is met:
  - When RMFM[4:0] = 0x00, after writing 1 to RMEDTG, operation completed until the next access delay (the master main state machine has transitioned to the idle state)
  - When RMFM[4:0] ≠ 0x00, after writing 1 to RMEDTG, operation completed until the next access delay (the master main state machine has transitioned to the idle state)
  - When RMFM[4:0] ≠ 0x00, the operation is completed up to the next access delay after processing is completed for the number of received frames set in RMFM[4:0] (the master main state machine has transitioned to the idle state)

[Slave mode]

Communication status: 0 - (1) to (9)

- SPCR.SPE = 0 (SPI initialization)

MODF bit (Mode Fault Error Flag)

This flag indicates whether a mode fault error or an underrun error is present. The UDRF flag allows you to see which error (mode fault error or underrun error) has occurred.

[Setting (to 1) condition]

[Multi-master mode]

- The SSL0 pin input level becomes active level while the SPCR.MSTR bit = 1 (master mode) and the SPCR.MODFEN bit = 1 (mode fault error detection enabled), and then the SPI has detected a mode fault error.

[Slave, Motorola-SPI mode]

Any of the following two conditions is met:

- The SSL0 pin is negated before the RSPCK cycles necessary for data transfer end while the SPCR.MSTR bit = 0 (slave mode), SPCR.SPFRF bit = 0 (Motorola-SPI) and the SPCR.MODFEN bit = 1 (mode fault error detection enabled), and then the SPI has detected a mode fault error.
- Serial transfer is started before transmit data output becomes ready while the SPCR.SPE bit = 1 (SPI function enabled), and then the SPI has detected an underrun error.

[Slave, TI-SSP mode]

Any of the following two conditions is met:

- The SSL0 pin is asserted before the RSPCK cycles necessary for data transfer end while the SPCR.MSTR bit = 0 (slave mode), SPCR.SPFRF bit = 1 (TI-SSP) and the SPCR.MODFEN bit = 1 (mode fault error detection enabled), and then the SPI has detected a mode fault error.
- Serial transfer is started before transmit data output becomes ready while the SPCR.SPE bit = 1 (SPI function enabled), and then the SPI has detected an underrun error.

The SSL signal active level depends on the SSLP.SSLiP bits (SSL signal polarity bits).

[Clearing (to 0) condition]

- When 1 is written to the SPSRC.MODFC bit

PERF bit (Parity Error Flag)

This flag indicates whether a parity error is present.

[Setting (to 1) condition]

When the serial transfer ends and a parity error is detected with the SPCR.SPPE bit set to 1 under any of the following two conditions:

- The SPCR.TXMD[1:0] bits = 00b (transmit-receive)
- The SPCR.TXMD[1:0] bits = 10b (receive-only)

[Clearing (to 0) condition]

- When 1 is written to the SPSRC.PERFC bit

UDRF bit (Underrun Error Flag)

This flag indicates that a mode fault error or an underrun error is present.

[Setting (to 1) condition]

- Serial transfer is started before transmit data output becomes ready while the SPCR.MSTR bit = 0 (slave mode) and the SPCR.TXMD[1:0] bits = 00b or 01b (transmit-receive or transmit-only) and the SPCR.SPE bit = 1 (SPI function enabled), and then the SPI has detected an underrun error.

[Clearing (to 0) condition]

- When 1 is written to the SPSRC.UDRFC bit

SPTEF bit (SPI Transmit Buffer Empty Flag)

This flag indicates the transmit buffer (SPTX) status in the SPI data register (SPDR).

[Setting (to 1) condition]

Any of the following three conditions is met:

- The SPCR.SPE bit is set to 0 (SPI initialization)
- When the number of empty transmission FIFO stages > SPDCR2.TTRG[1:0]
- When 1 is written to SPFCR.SPFRST

[Clearing (to 0) condition]

Any of the following two conditions is met:

- At the time of final access when transmission data is written to SPDR (SPTXn, n = 0 to 2) in one processing routine using DMAC
- When 1 is written to the SPSRC.SPTEFC bit

Writing a value to the SPDR register is enabled only while the SPTEF flag = 1. If a value is written to the SPDR register while the SPTEF flag = 0, transmit buffer data is not updated.

CENDF bit (Communication End Flag)

This flag indicates communication end status of SPI. It turns 1 at communication end and turns 0 at starting next communication.

[Setting (to 1) condition]

Transmit-Receive / Transmit-only Master mode

Communication status: 1 - (1) to (6). For details of communication status, see Table 34.4.

The following three conditions are met:

- The next transfer data is not set in the transmission buffer (SPTXn, n = 0 to 3)
- The SPSSR.SPCP[2:0] are 000b (it means the head of the sequential control)
- Operation completed until the next access delay (the master main state machine has transitioned to the idle state)

Receive-only Master mode

Communication status: 1 - (7) to (9)

Any of the following three conditions is met:

- When RMFM[4:0] = 0x00, after writing 1 to RMEDTG, operation completed until the next access delay (the master main state machine has transitioned to the idle state)
- When RMFM[4:0] ≠ 0x00, after writing 1 to RMEDTG, operation completed until the next access delay (the master main state machine has transitioned to the idle state)
- When RMFM[4:0] ≠ 0x00, the operation is completed up to the next access delay after processing is completed for the number of received frames set in RMFM[4:0] (the master main state machine has transitioned to the idle state)

Transmit-receive / transmit-only slave, Motorola-SPI mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0).

Communication status: 0 - (1), (4)

The following three conditions are met:

- The next transfer data is not set in the transmission buffer
- The transmission shift register is empty (it means SPI does not do serial transfer)
- SSL0 was negated

Transmit-receive / transmit-only slave, TI-SSP mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0).

Communication status: 0 - (2), (5)

The following three conditions are met:

- The next transfer data is not set in the transmission buffer
- The transmission shift register is empty (it means SPI does not do serial transfer)
- When the SSL negate delay is completed

Transmit-receive / transmit only slave mode at clock synchronous (3-wire: the SPCR.SPMS bit is 1)

Communication status: 0 - (3), (6)

The following three conditions are met:

- The next transfer data is not set in the transmission buffer
- The transmission shift register is empty (it means SPI does not do serial transfer)
- The last even edge of RSPCK of the frame was detected (when the SPCMD.CPHA bit is 1)

Receive only slave, Motorola-SPI mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0).

Communication status: 0 - (7)

The following condition is met:

- SSL0 input was negated after getting frames for SPDCR.SPFC set value in the receive buffer

Receive only slave, TI-SSP mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0).

Communication status: 0 - (8)

The following condition is met:

- SSL0 negate delay is completed after getting frames for SPDCR.SPFC set value in the receive buffer

Receive only slave mode at clock synchronous (3-wire: the SPCR.SPMS bit is 1).

Communication status: 0 - (9)

The following condition is met:

- The last even edge of RSPCK of the last frame received for SPFC set value was detected (when the SPCMD.CPHA bit is 1)

[Clearing (to 0) condition]

Transmit-Receive / Transmit-only Master mode

Communication status: 1 - (1) to (6)

Any of the following two conditions is met:

- The next transmit data was written to the transmit buffer (SPTX)
- When 1 is written to the SPSRC.CENDFC bit

Receive-only Master mode

Communication status: 1 - (7) to (9)

Any of the following two conditions is met:

- When 1 is written to the SPCRRM.RMSTTG bit with SPCR.SPE = 1
- When 1 is written to the SPSRC.CENDFC bit

Transmit-receive / transmit only slave mode

Communication status: 0 - (1) to (6)

Satisfy one of following conditions:

- The next transmit data is written to the transmit buffer (SPTX)
- When 1 is written to the SPSRC.CENDFC bit

Receive-only slave mode at SPI serial communication (4-wire: the SPCR.SPMS bit is 0).

Communication status: 0 - (7) to (8)

Satisfy one of following conditions:

- SSL0 assertion of next data is detected
- When 1 is written to the SPSRC.CENDFC bit

Receive-only slave mode at clock synchronous (3-wire: the SPCR.SPMS bit is 1).

Communication status: 0 - (9)

Satisfy one of following conditions:

- The first edge of RSPCK of the next data is detected
- When 1 is written to the SPSRC.CENDFC bit

SPRF bit (SPI Receive Buffer Full Flag)

This flag indicates the receive buffer (SPRX) status in the SPDR register.

[Setting (to 1) condition]

When the number of data stored in the receive FIFO > the number of frames set in the SPDCR2.RTRG[1:0] bits in Transmit-Receive, Receive-only modes. However, the SPRF flag does not change from 0 to 1 while the OVRF flag = 1. (See section 34.3.10. Error Detection.)

[Clearing (to 0) condition]

Any of the following three conditions is met:

- At the last access when read data is read from SPDR (SPRXn, n = 0 to 2) in one processing routine using DMAC
- When 1 is written to the SPSRC.SPRFC bit
- When 1 is written to the SPFCR.SPFRST bit

34.2.19 SPTFSR : SPI Transfer FIFO Status Register

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x58

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	TFDN[2:0]		
Value after reset:	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
2:0	TFDN[2:0]	Transmit FIFO data empty stage number 0 0 0: Number of empty stages 0 0 0 1: Number of empty stages 1 0 1 0: Number of empty stages 2 0 1 1: Number of empty stages 3 1 0 0: Number of empty stages 4	R
7:3	—	These bits are read as 0.	R

Note: TFDN[2:0] is initialized by SPCR.SPE = 0.

34.2.20 SPRFSR : SPI Receive FIFO Status Register

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x5C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	RFDN[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	RFDN[2:0]	Receive FIFO data store stage number 0 0 0: Number of store stages 0 0 0 1: Number of store stages 1 0 1 0: Number of store stages 2 0 1 1: Number of store stages 3 1 0 0: Number of store stages 4	R
7:3	—	These bits are read as 0.	R

Note: RFDN[2:0] is initialized by SPCR.SPE = 0.

34.2.21 SPPSR : SPI Poling Register

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x60

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPEPS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPEPS	SPI Polling Status 0: SPCR.SPE is 0 1: SPCR SPE is 1	R
31:1	—	These bits are read as 0.	R

Note: This bit shows a result of SPCR.SPE after synchronization by the operation clock (PCLKSPIn).

34.2.22 SPSRC : SPI Status Clear Register

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x6A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SPRFC	CENDFC	SPTEFC	UDRFC	PERFC	MODFC	—	OVRFC	SPDRFC	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	—	These bits are read as 0.	R
7	SPDRFC	SPI Receive Data Ready Flag Clear The SPI receive data ready flag can be cleared by writing 1. When read, 0 is read.	W
8	OVRFC	Overrun Error Flag Clear By writing 1, the Overrun Error Flag can be cleared. Reading value is always 0.	W
9	—	This bit is read as 0.	R
10	MODFC	Mode Fault Error Flag Clear By writing 1, the Mode Fault Error Flag can be cleared. Reading value is always 0. Before setting MODFC and UDRFC, make sure that SPSR.MODF and UDRF are set to 1.	W
11	PERFC	Parity Error Flag Clear By writing 1, the Parity Error Flag can be cleared. Reading value is always 0.	W
12	UDRFC	Underrun Error Flag Clear By writing 1, the Underrun Error Flag can be cleared. Reading value is always 0. When clearing the UDRF flag, clear the MODF flag at the same time (MODFC = 1).	W
13	SPTEFC	SPI Transmit Buffer Empty Flag Clear By writing 1, the SPI Transmit Buffer Empty Flag can be cleared. Reading value is always 0.	W
14	CENDFC	Communication End Flag Clear By writing 1, the Communication End Flag can be cleared. Reading value is always 0.	W
15	SPRFC	SPI Receive Buffer Full Flag Clear By writing 1, the SPI Receive Buffer Full Flag can be cleared. Reading value is always 0.	W

The SPSRC register clears the status flag (SPSR) that indicates the operating status of SPI.

34.2.23 SPFCR : SPI FIFO Clear Register

Base address: SPIn = 0x8000_3000 + 0x0400 × n (n = 0 to 2)
SPI3 = 0x8100_2000

Offset address: 0x6C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	SPFRST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPFRST	SPI FIFO clear By writing 1, the pointer in the FIFO and the stored data are initialized. Reading value is always 0.	W
7:1	—	The write value should be 0.	W

The SPFCR register is used to clear the FIFO.

If SPFCR is rewritten while SPCR.SPE = 1, subsequent operations are not guaranteed.

34.3 Operation

In this section, a word "serial transfer period" is used as a word that means a period from the start of valid data drive to latching of the final valid data.

34.3.1 Overview of SPI Operations

The SPI can transfer data in the following five modes.

- Slave mode (SPI operation)
- Single master mode (SPI operation)
- Multi-master mode (SPI operation)
- Slave mode (clock synchronous operation)
- Master mode (clock synchronous operation)

SPI modes can be set by the MSTR, MODFEN, SPMS and SPFRF bits in the SPCR register.

Table 34.7 lists the relationship between SPI modes and SPCR settings, and outline of each mode.

Table 34.7 Relationship between SPI modes and SPCR settings and description of each mode (1 of 2)

Mode	Slave (SPI operation)	Single-Master (SPI operation)	Multi-Master (SPI operation)	Slave (Clock synchronous operation)	Master (Clock synchronous operation)
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0 or 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
SPFRF bit setting	valid	valid	valid	In-valid	In-valid
RSPCK signal	Input	Output	Output/Hi-Z	Input	Output
MOSI signal	Input	Output	Output/Hi-Z	Input	Output
MISO signal	Output/Hi-Z	Input	Input	Output	Input
SSL0 signal	Input	Output	Input	Hi-Z (not used)	Hi-Z (not used)

Table 34.7 Relationship between SPI modes and SPCR settings and description of each mode (2 of 2)

Mode	Slave (SPI operation)	Single-Master (SPI operation)	Multi-Master (SPI operation)	Slave (Clock synchronous operation)	Master (Clock synchronous operation)
SSL1 to SSL3 signals	Hi-Z (not used)	Output	Output/Hi-Z	Hi-Z (not used)	Hi-Z (not used)
Output pin mode	CMOS/open drain	CMOS/open drain	CMOS/open drain	CMOS/open drain	CMOS/open drain
SSL polarity modification function	Supported	Supported	Supported	—	—
Transfer rate	Up to PCLKSPIn/2	Up to PCLKSPIn/2	Up to PCLKSPIn/2	Up to PCLKSPIn/2	Up to PCLKSPIn/2
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator
Clock polarity	2 types				
Clock phase	2 types*5	2 types*5	2 types*5	One (CPHA = 1)	2 types
First transfer bit	MSB/LSB				
Transfer data length	4 to 32 bits				
Burst transfer	Enabled (CPHA = 1)	Enabled (CPHA = 0, 1)	Enabled (CPHA = 0, 1)	—	—
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported*6	Supported	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Supported	Not supported	Supported
Transfer activation method	SSL input active or RSPCK oscillation	Transmit buffer empty interrupt request or transmit buffer write when SPTEF = 1	Transmit buffer empty interrupt request or transmit buffer write when SPTEF = 1	RSPCK oscillation	Transmit buffer empty interrupt request or transmit buffer write when SPTEF = 1
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmission buffer empty detection	Supported*4	Supported	Supported	Supported*4	Supported
Reception buffer full detection	Supported*1				
Overrun error detection	Supported*1	Supported*1 *3	Supported*1 *3	Supported*1	Supported*1 *3
Parity error detection	Supported*1 *2				
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported
Underrun error detection	Supported*4	Not supported	Not supported	Supported*4	Not supported

Note 1. When SPI is transmit-master mode or transmit-slave mode (see Table 34.4), none of receive buffer full error, overrun error, and parity error is detected.
Note 2. When the SPCR.SPPE bit is 0, parity error is not detected.
Note 3. When the SPCR.SCKASE bit is 1, overrun error is not detected.
Note 4. When SPI is receive only slave mode (see Table 34.4), none of transmit buffer empty and underrun error is detected.
Note 5. CPHA = 0 is invalid in TI SSP mode. (Even if it is set, the operation is the same as when CPHA = 1.)
Note 6. Available only in TI SSP mode.

34.3.2 SPI Pin Control

The SPI automatically switches pin directions and output modes according to the settings of the SPCR.MSTR, SPCR.MODFEN and SPCR.SPMS bits and the SPPCR.SPOM bit.

When the SPPCR.SPOM bit is set to 0, each output becomes CMOS output, when the SPOM bit is set to 1, each output becomes open drain output.

Table 34.8 shows the relationship between pin state and set values of each bit.

For details of OE function, see section 34.3.5. Transfer Format (Frame Format).

Table 34.8 Relationship between pin states and set control bit values

Mode	Pin	Pin State*1	
		SPOM = 0	SPOM = 1
Single-master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0)	RSPCK	CMOS output	Open-drain output
	SSL0 to SSL3	CMOS output	Open-drain output
	MOSI	CMOS output	Open-drain output
	MISO	Input	Input
Multi-master mode (SPI operation) (MSTR = 1, MODFEN = 1, SPMS = 0)	RSPCK*2	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSL0	Input	Input
	SSL1 to SSL3*2	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSI*2	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MISO	Input	Input
Slave mode (SPI operation) (MSTR = 0, SPMS = 0)	RSPCK	Input	Input
	SSL0	Input	Input
	SSL1 to SSL3*4	Hi-Z (not used)	Hi-Z (not used)
	MOSI	Input	Input
	MISO*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
Master mode (Clock synchronous operation) (MSTR = 1, MODFEN = 0, SPMS = 1)	RSPCK	CMOS output	Open-drain output
	SSL0 to SSL3*4	Hi-Z (not used)	Hi-Z (not used)
	MOSI	CMOS output	Open-drain output
	MISO	Input	Input
Slave mode (Clock synchronous operation) (MSTR = 0, SPMS = 1)	RSPCK	Input	Input
	SSL0 to SSL3*4	Hi-Z (not used)	Hi-Z (not used)
	MOSI	Input	Input
	MISO	CMOS output	Open-drain output

Note 1. The set SPI value is not applied to multi-function pins for which the SPI function is not selected.
Note 2. When SSL0 is at the active level, the pin state is Hi-Z. Motorola-SPI: When SSL0 is active level, the pin state becomes Hi-Z. TI-SSP: The terminal status becomes Hi-Z until SSL0 is asserted after SPCR.SPE = 1 and communication completed.
Note 3. Motorola-SPI: When SSL0 is inactive level or when SPCR.SPE = 0, the pin state becomes Hi-Z. TI-SSP: When SSL0 is except the communication period or when SPCR.SPE = 0 (assertion after SPE = 1 and communication completed), the pin status changes to Hi-Z.
Note 4. These pins are available for use as I/O port pins.

The SPI in single master mode (SPI operation) or multi-master mode (SPI operation) determines the MOSI signal value in the SSL negation period (including the SSL hold period in burst transfer) according to the settings of the SPPCR.MOIFE and SPPCR.MOIFV bits, as listed in Table 34.9.

Table 34.9 Determining the MOSI signal value in the SSL negation period

MOIFE bit	MOIFV bit	MOSI signal value during SSL negation period
0	0, 1	Final data from previous transfer
1	0	Always low
1	1	Always high

34.3.3 SPI System Configuration Examples

This configuration example describes that 0 level of SSL signals is active level. When connecting and using in a multi-slave or multi-master mode, the transfer format of the connected device should be unified to either Motorola-SPI or TISSP.

34.3.3.1 Single Master/Single Slave (This LSI = Master)

Figure 34.7 shows an example of single master/single slave SPI system configuration where this LSI is used as a master. In the single master/single slave configuration, the SSL0 to SSL3 output signals of this LSI (master) are not used.

SSL input signals of the SPI slave are fixed to 0 level to always select the SPI slave. When SPCMD.CPHA = 0, some slave devices cannot fix SSL signals to active level in the relevant transfer format. If the SSL signal level cannot be fixed, connect the SSL output of this LSI to the SSL input of the slave device.

This LSI (master) always drives the RSPCK and MOSI pins. The SPI slave always drives the MISO pin.

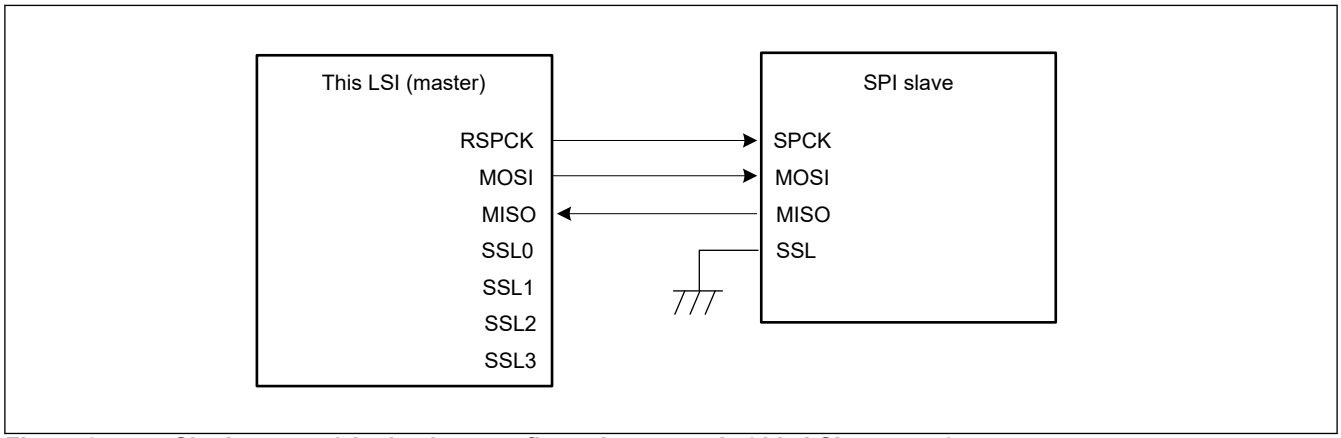


Figure 34.7 Single-master/single-slave configuration example (this LSI = master)

34.3.3.2 Single Master/Single Slave (This LSI = Slave)

Figure 34.8 shows an example of single master/single slave SPI system configuration where this LSI is used as a slave. When this LSI is used as a slave, the SSL0 pin is used as SSL input. The SPI master always drives the RSPCK and MOSI pins. This LSI (slave) always drives the MISO pin. When the SSL0 level is inactive, the pin state becomes Hi-Z.

In the single slave configuration with the SPCMD.CPHA bit set to 1, set the SPCR.SPFRF bit to 0, and set the SPMS bit to 0. There is the SSL0 input level of this LSI (slave) is fixed to 0 so that this LSI (slave) can be always selected and serial transfer can also be performed (Figure 34.9). However, the communication end interrupt does not output when SSL0 input was fixed as Figure 34.9.

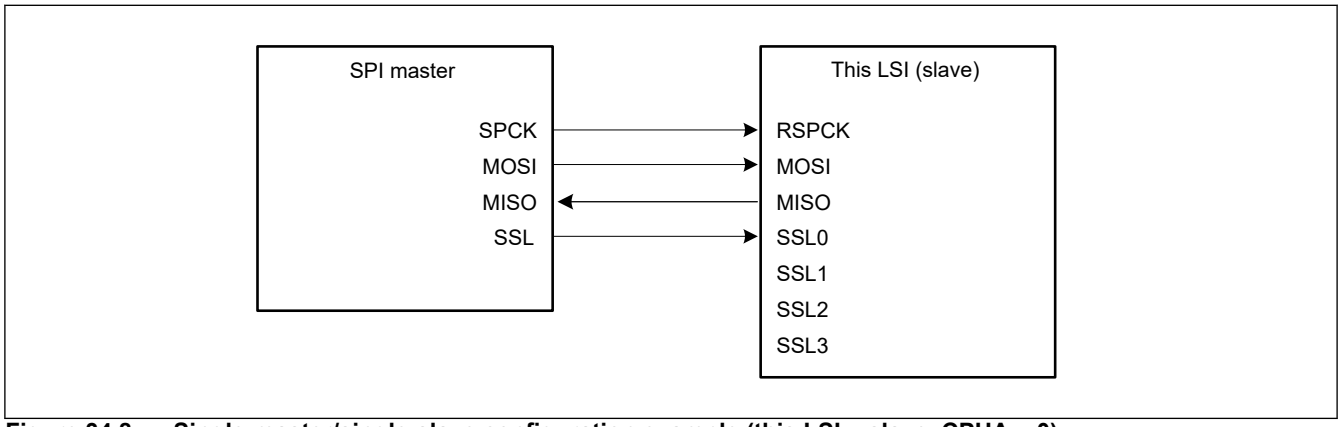


Figure 34.8 Single-master/single-slave configuration example (this LSI = slave, CPHA = 0)

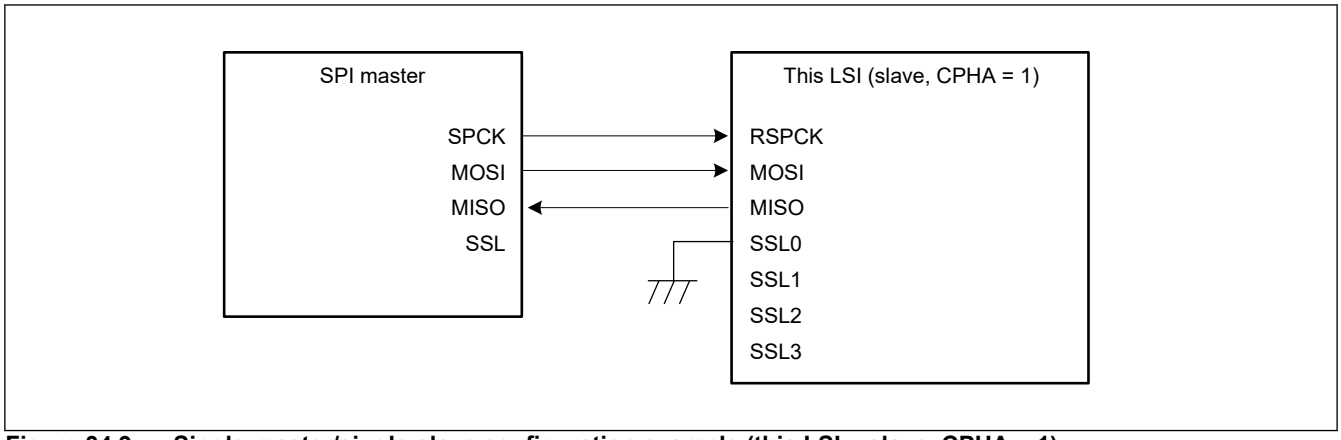


Figure 34.9 Single-master/single-slave configuration example (this LSI = slave, CPHA = 1)

34.3.3.3 Single Master/Multi-Slave (This LSI = Master)

Figure 34.10 shows an example of single master/multi-slave SPI system configuration where this LSI is used as a master. In the example in Figure 34.10, the SPI system is comprised of this LSI (master) and four slaves (SPI slave 0 to SPI slave 3).

The RSPCK output and MOSI output pins of the master are connected to the RSPCK input and MOSI input pins of SPI slave 0 to SPI slave 3. All of the MISO output pins of SPI slave 0 to SPI slave 3 are connected to the MISO input pin of the master. The SSL0 to SSL3 output pins of this LSI (master) are connected to the SSL input pin of SPI slave 0 to SPI slave 3.

The master always drives RSPCK, MOSI, and SSL0 to SSL3 pins. The slave (out of SPI slave 0 to SPI slave 3) where 0 level is input to SSL input pin drives the MISO pin.

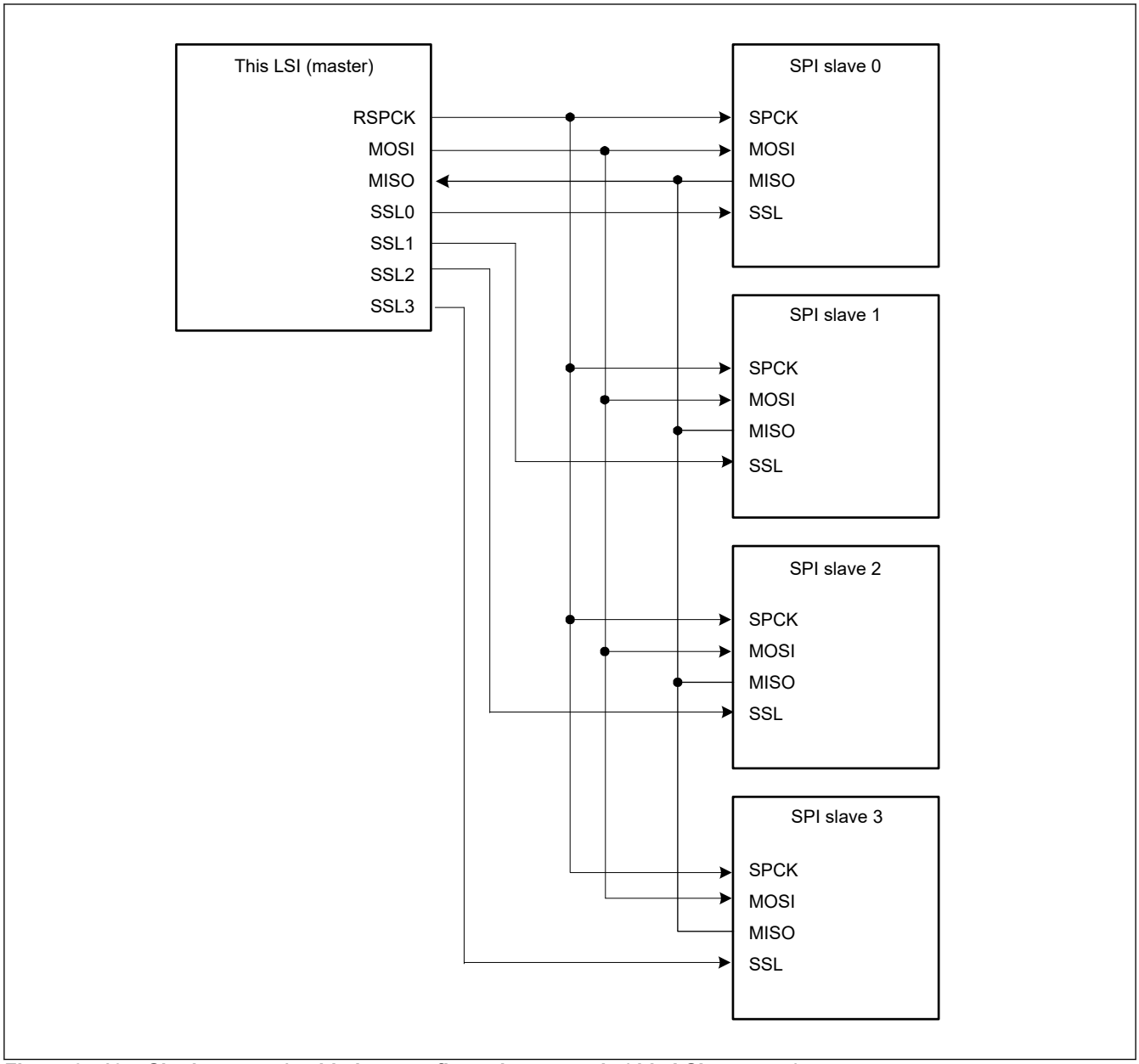


Figure 34.10 Single-master/multi-slave configuration example (this LSI = master)

34.3.3.4 Single Master/Multi-Slave (with This LSI Acting as Slave)

Figure 34.11 shows an example of single master/multi-slave SPI system configuration where this LSI is used as a slave. In the example in Figure 34.11, the SPI system consists of the SPI master and two these LSIs (slave X, slave Y).

The RSPCK output and MOSI output pins of the SPI master are connected to the RSPCK input and MOSI input pins of the LSIs (slave X, slave Y). The MISO output pin of the LSIs (slave X, slave Y) is connected to the MISO input pin of the SPI master. The SSLX output and SSLY output pins of the SPI master are connected to the SSL0 input pin of the LSIs (slave X, slave Y).

The SPI master always drives the RSPCK, MOSI, SSLX, and SSLY pins. The slave X or slave Y (this LSI) where 0 level is input to the SSL0 input pin drives the MISO pin.

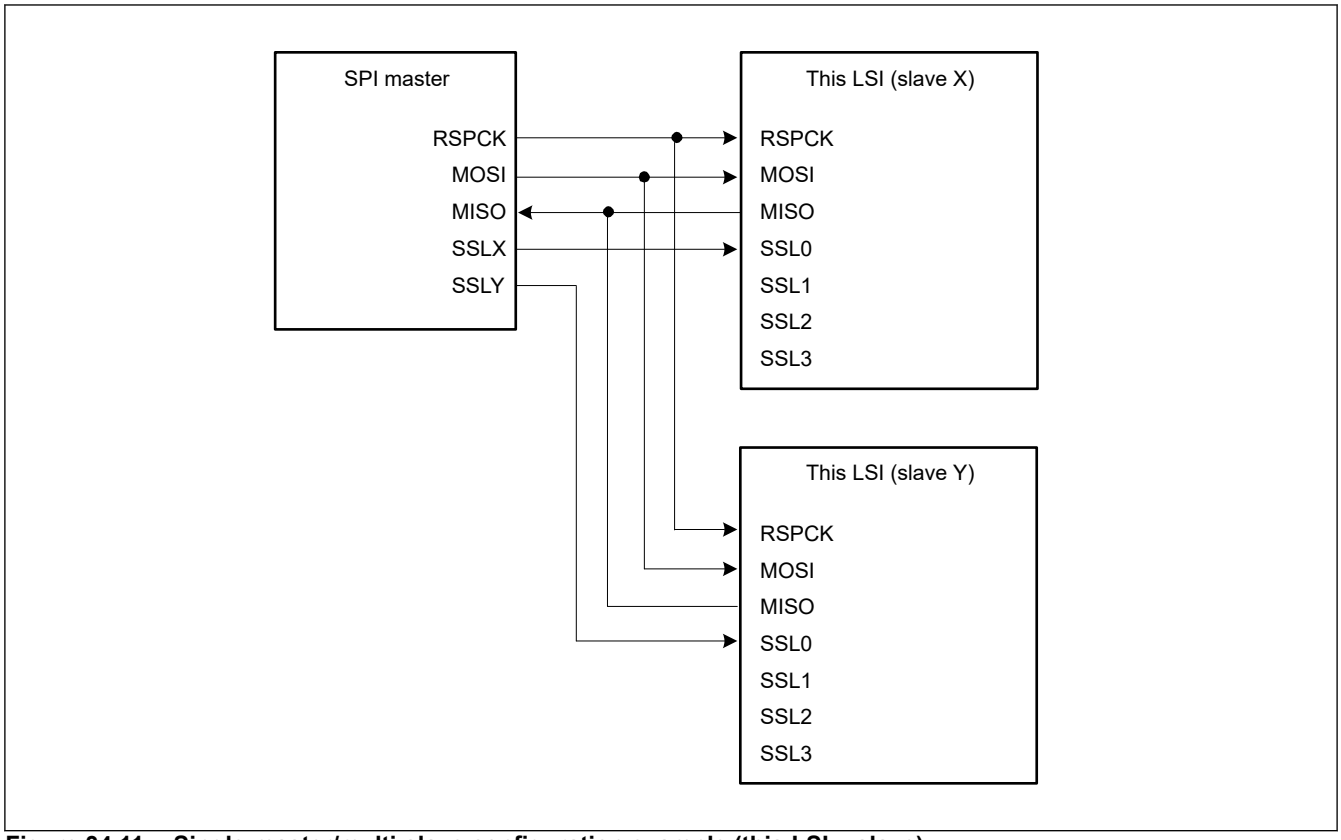


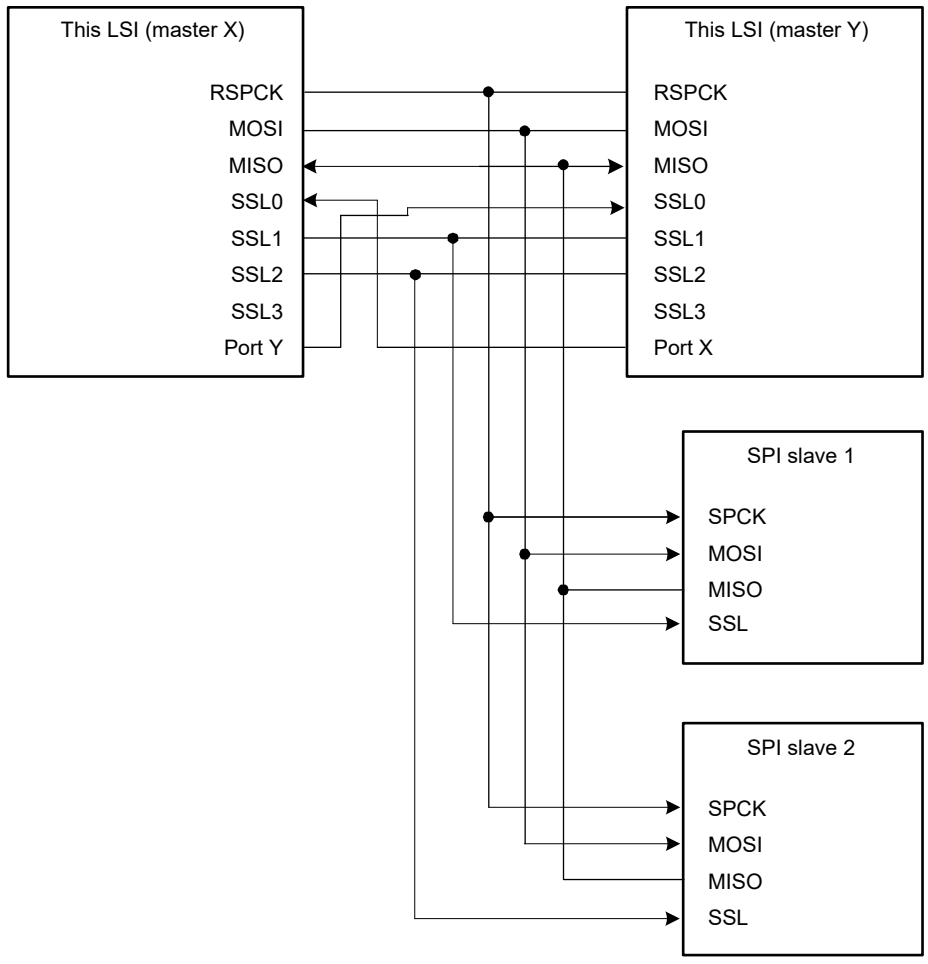
Figure 34.11 Single-master/multi-slave configuration example (this LSI = slave)

34.3.3.5 Multi-Master/Multi-Slave (with This LSI Acting as Master)

Figure 34.12 shows an example of multi-master/multi-slave SPI system configuration where this LSI is used as a master. In the example in Figure 34.12, the SPI system consists of two these LSIs (master X, master Y) and two SPI slaves (SPI slave 1, SPI slave 2).

The RSPCK output and MOSI output pins of these LSIs (master X, master Y) are connected to the RSPCK input and MOSI input pins of SPI slave 1 and SPI slave 2. The MISO output pin of SPI slave 1 and SPI slave 2 is connected to the MISO input pin of these LSIs (master X, master Y). The Port Y (general port) output pin of this LSI (master X) is connected to the SSL0 input pin of this LSI (master Y). The Port X (general port) output pin of this LSI (master Y) is connected to the SSL0 input pin of this LSI (master X). The SSL1 output and SSL2 output pins of these LSIs (master X, master Y) are connected to the SSL input pin of SPI slave 1 and SPI slave 2. In this configuration example, SSL3 output pins of this LSI are not used because the system can be configured only with SSL0 input pin and SSL1 output and SSL2 output pins for connecting slaves.

While the SSL0 input level is 1, this LSI drives the RSPCK, MOSI, SSL1, and SSL2 pins. While the SSL0 input level is 0, this LSI detects a mode fault error and changes the RSPCK, MOSI, SSL1, SSL2 pin levels to Hi-Z to release the SPI bus mastership for another master. SPI slave 1 or SPI slave 2 where 0 level is input to the SSL input pin drives the MISO pin.



Note: When setting TI SSP, enter the following levels for port X and port Y.
- Start of communication: the value of SSLP.SSL0P of the other master.
- End of communication: the inverted value of SSLP.SSL0P of the other master.

Figure 34.12 Multi-master/multi-slave configuration example (this LSI = master)

34.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (This LSI = Master)

Figure 34.13 shows an example of master (clock synchronous operation)/slave (clock synchronous operation) SPI system configuration where this LSI is used as a master. In this configuration, the SSL0 to SSL3 output pins of this LSI (master) are not used.

This LSI (master) always drives the RSPCK and MOSI pins. The SPI slave always drives the MISO pin.

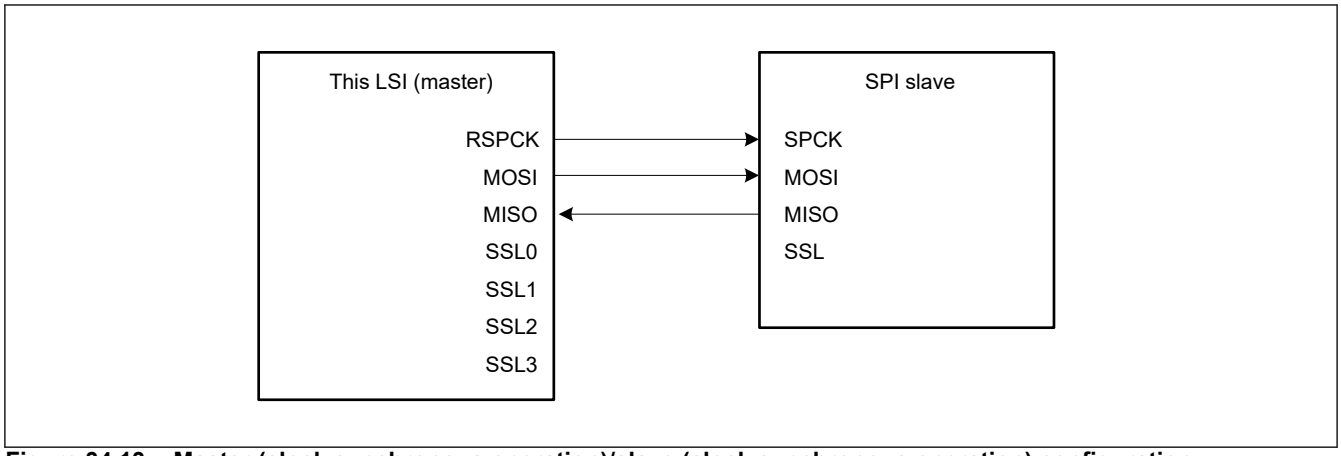


Figure 34.13 Master (clock synchronous operation)/slave (clock synchronous operation) configuration example (this LSI = master)

34.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (This LSI = Slave)

Figure 34.14 shows an example of master (clock synchronous operation)/slave (clock synchronous operation) SPI system configuration where this LSI is used as a slave. When this LSI is used as a slave (clock synchronous operation), this LSI (slave) always drives the MISO pin and the SPI master always drives the RSPCK and MOSI pins.

Only in the single slave configuration with the SPCMD.CPHA bit set to 1, this LSI (slave) can perform serial transfer.

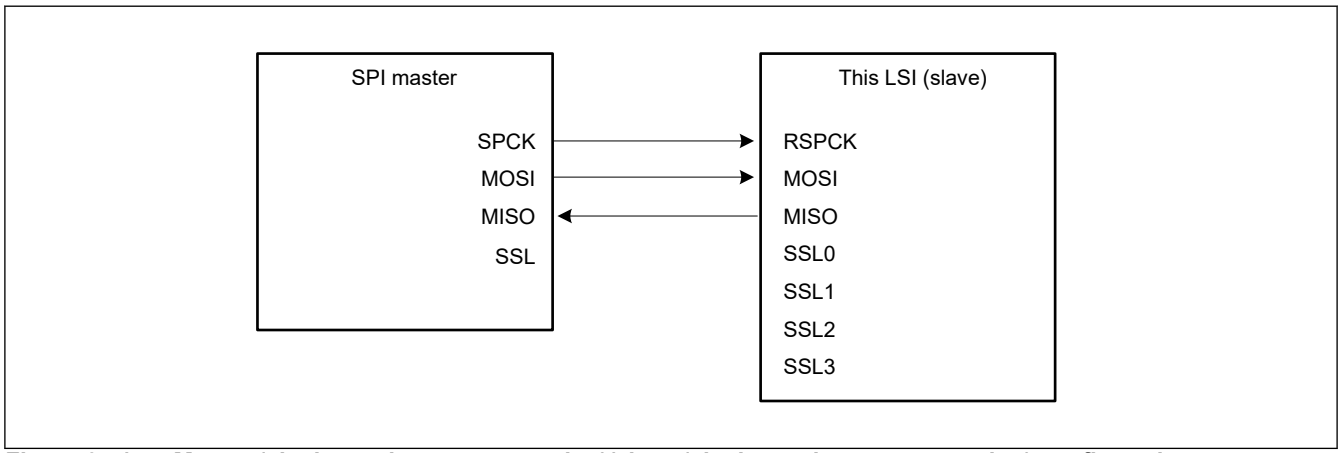


Figure 34.14 Master (clock synchronous operation)/slave (clock synchronous operation) configuration example (this LSI = slave, CPHA = 1)

34.3.4 Data Format

Data format of the SPI depends on the set values of the SPCMD register and the SPCR.SPPE bit. The SPI handles data of the set data length from the LSB in the SPDR register as transfer data regardless of MSB first or LSB first.

34.3.4.1 Data Format of One Frame

The data format of one frame for transmission and reception is shown below.

(1) With Parity Disabled (SPPE=0)

When the parity function is disabled, data of the bit length specified by SPCMDm.SPB[4:0] is transmitted and received.

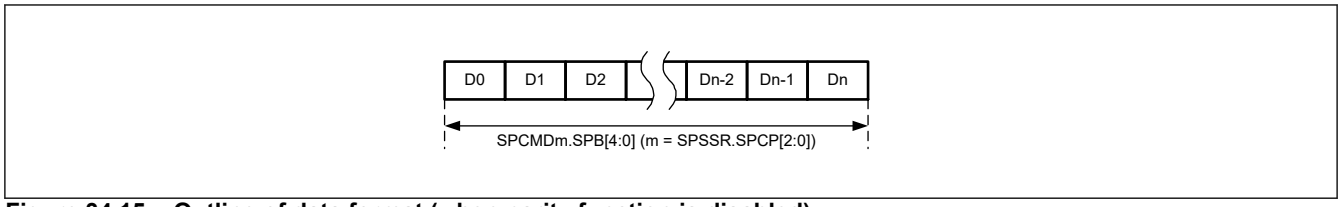


Figure 34.15 Outline of data format (when parity function is disabled)

(2) With Parity Enabled (SPPE = 1)

When the parity function is enabled, data of the bit length specified by SPCMDm.SPB[4:0] is transmitted and received. However, the final bit is used as a parity bit.

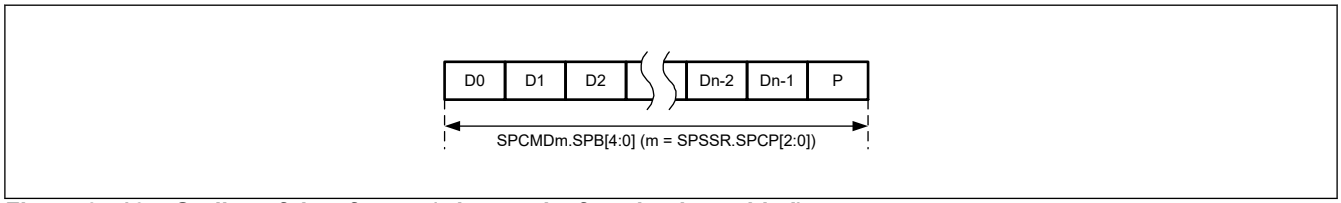


Figure 34.16 Outline of data format (when parity function is enabled)

34.3.4.2 When Parity is Disabled (SPPE = 0)

When the parity function is disabled, transmit buffer data is copied to the shift register without processing it. The following describes the relationship between SPDR, and shift register by using a combination of MSB first/LSB first and bit length.

(1) MSB First Transfer (32-Bit Data)

Figure 34.17 shows operations of the SPDR register and the shift register when the SPI performs 32-bit MSB-first transfer with the parity function disabled.

In transmission, T31 to T00 in the transmit buffer are copied to the shift register. Bit values in the shift register are shifted and transmitted in the order of T31 → T30 → … → T00 as transmit data.

In reception, received data is stored in bit 0 of the shift register, and received data is shifted in units of data. When necessary RSPCK cycles are input and data is stored from R31 to R00, the shift register value is copied to the receive buffer.

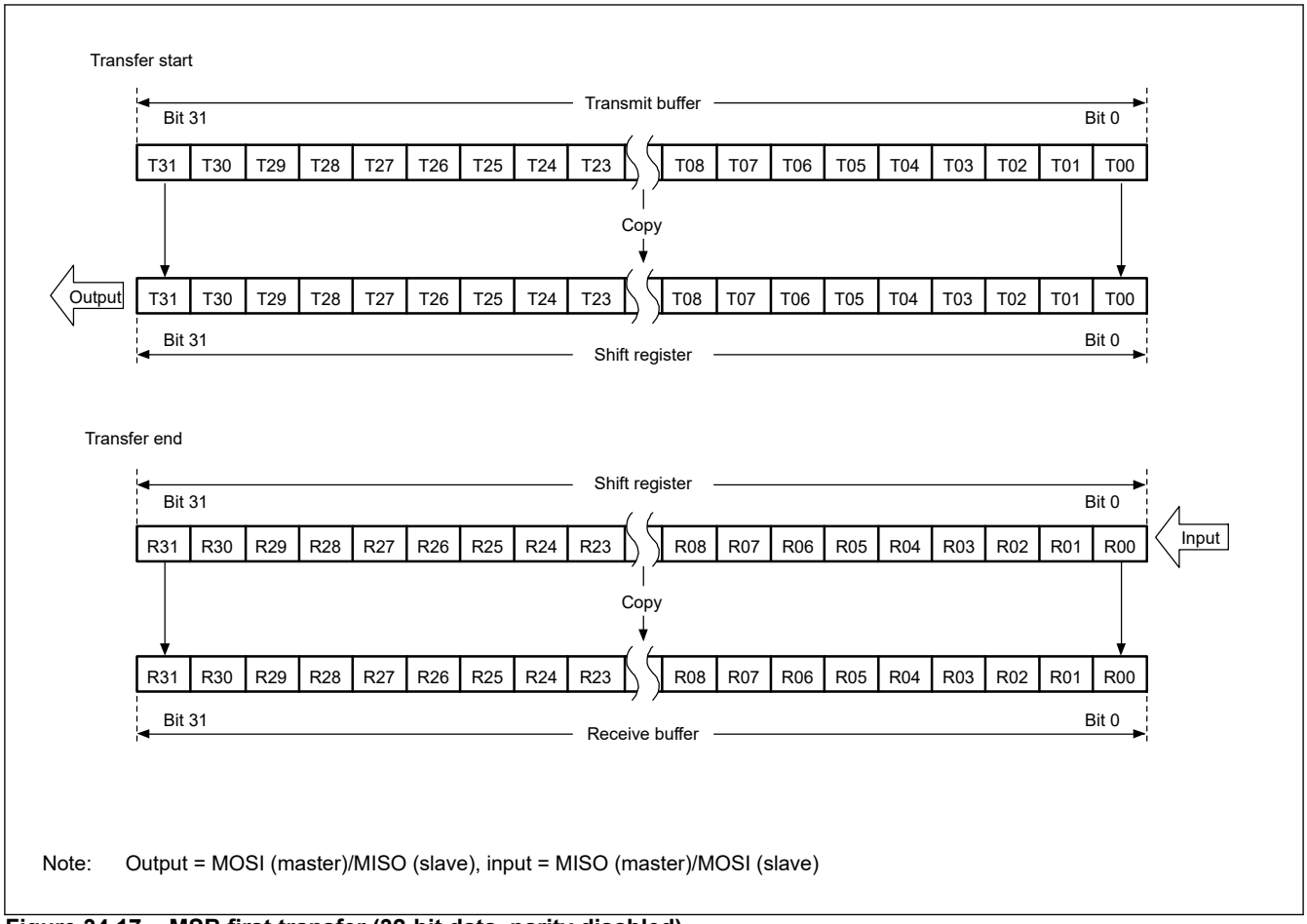


Figure 34.17 MSB first transfer (32-bit data, parity disabled)

(2) MSB First Transfer (24-Bit Data)

As an example of MSB-first transfer of data other than 32 bits with the parity function disabled, Figure 34.18 shows operations of the SPDR register and the shift register when the SPI performs 24-bit data transfer.

In transmission, lower 24 bits (T23 to T00) in the transmit buffer are copied to the shift register. Bit values in the shift register are shifted and transmitted in the order of T23 → T22 → … → T00 as transmit data.

In reception, received data is stored in bit 0 of the shift register, and received data is shifted in units of data. When necessary RSPCK cycles are input and data is stored from R23 to R00, the shift register value is copied to the receive buffer.

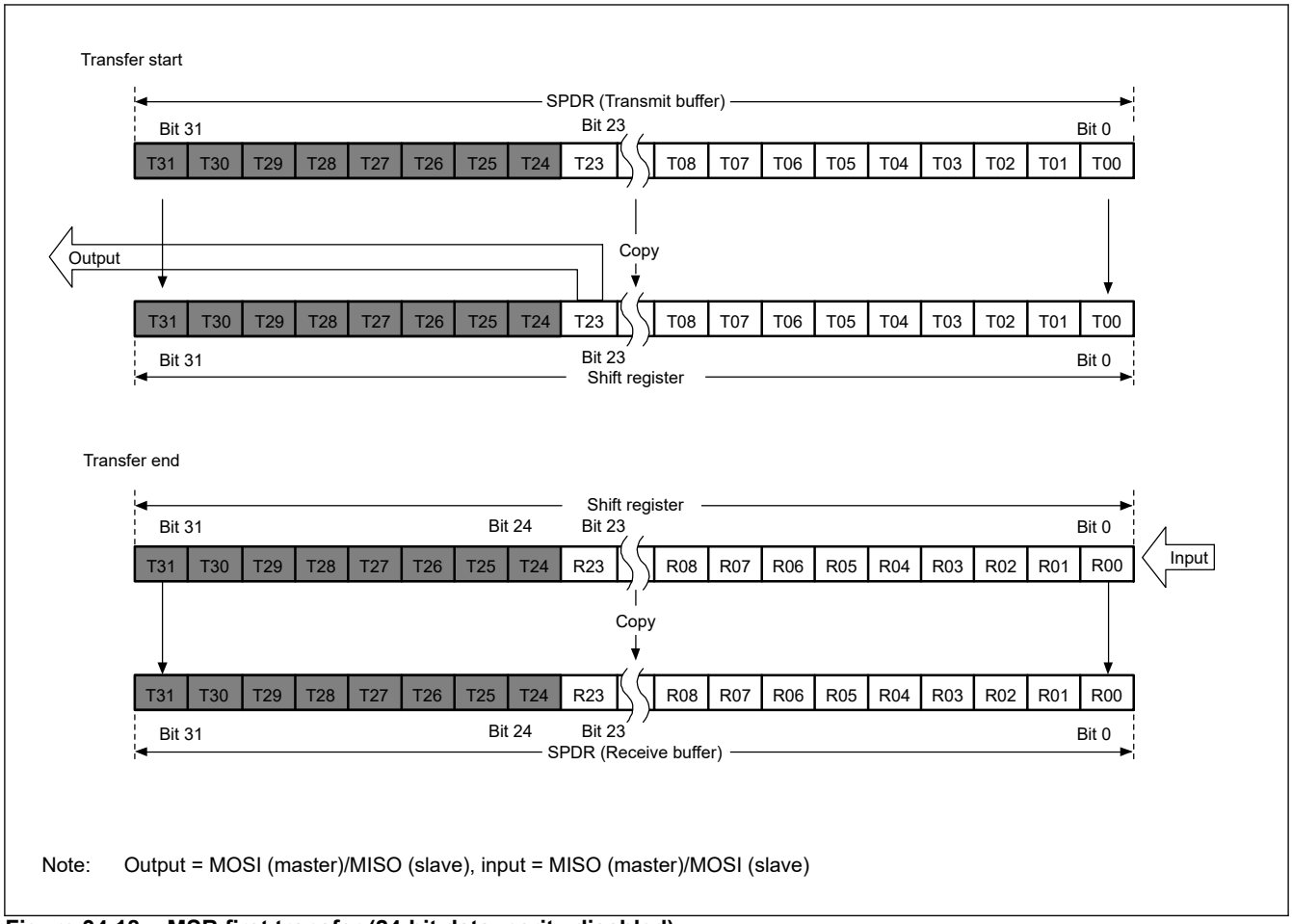


Figure 34.18 MSB first transfer (24-bit data, parity disabled)

(3) LSB First Transfer (32-Bit Data)

Figure 34.19 shows operations of the SPDR register and the shift register when the SPI performs 32-bit LSB-first transfer with the parity function disabled.

In transmission, bit values of the transmit buffer (T31 to T00) are reversed in bit units and are copied to the shift register in the order of T00 to T31. Bit values in the shift register are shifted and transmitted in the order of T00 → T01 → … → T31 as transmit data.

In reception, the first received data is stored in bit 0 of the shift register, and received data is shifted in units of data.

When necessary RSPCK cycles are input and data is stored from R00 to R31, bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of R31 to R00.

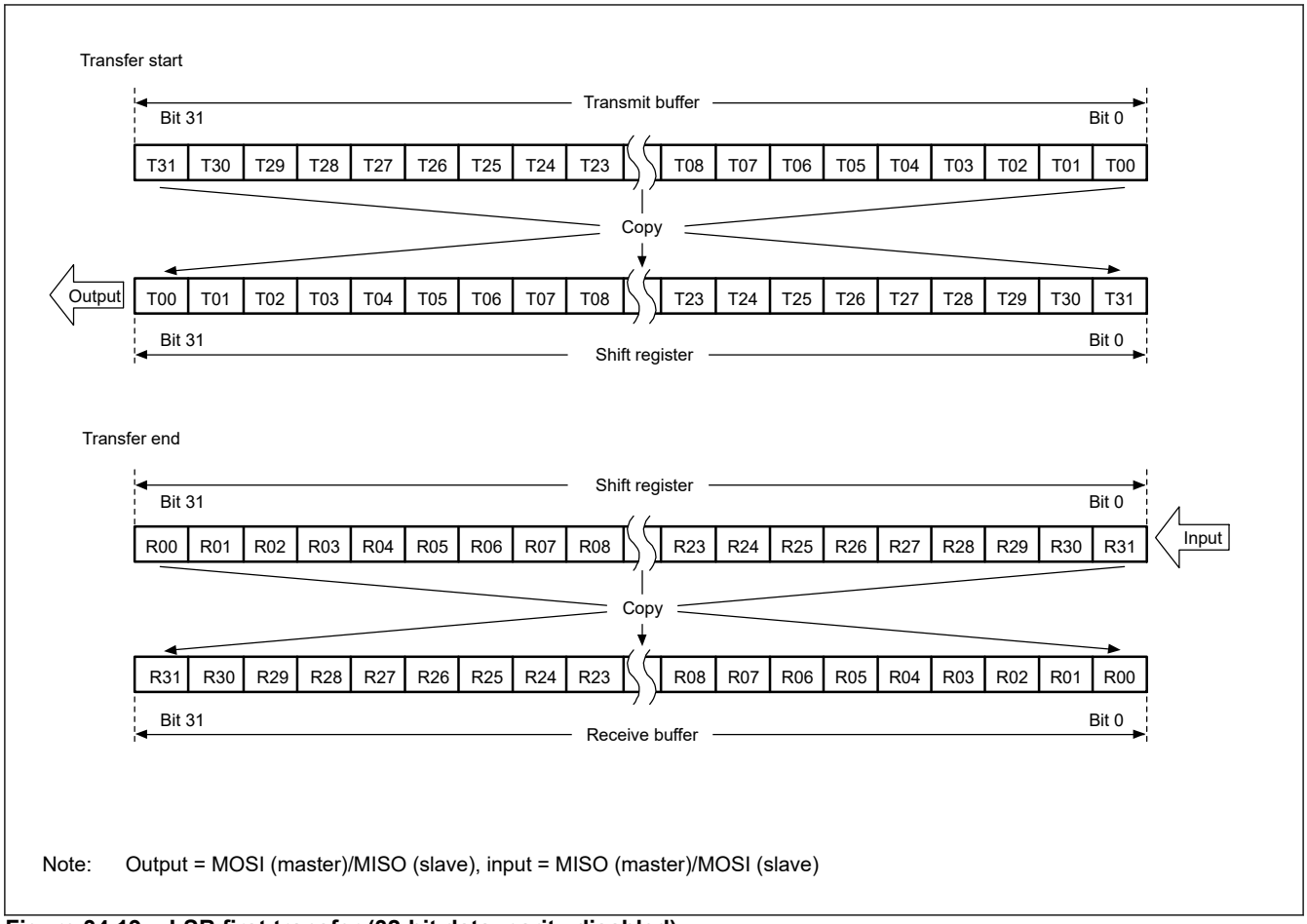


Figure 34.19 LSB first transfer (32-bit data, parity disabled)

(4) LSB First Transfer (24-Bit Data)

As an example of LSB-first transfer of data other than 32 bits with the parity function disabled, Figure 34.20 shows operations of the SPDR register and the shift register when the SPI performs 24-bit data transfer.

In transmission, lower 24-bit values of the transmit buffer (T23 to T00) are reversed to the order from T00 to T23 in bit units and are copied to the shift register. Bit values in the shift register are shifted and transmitted in the order of T00→ T01 → … → T23 as transmit data.

In reception, received data is stored in bit 8 of the shift register, and received data is shifted in units of data. When necessary RSPCK cycles are input and data is stored from R00 to R23, bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of R23 (bit 23) to R00 (bit 0).

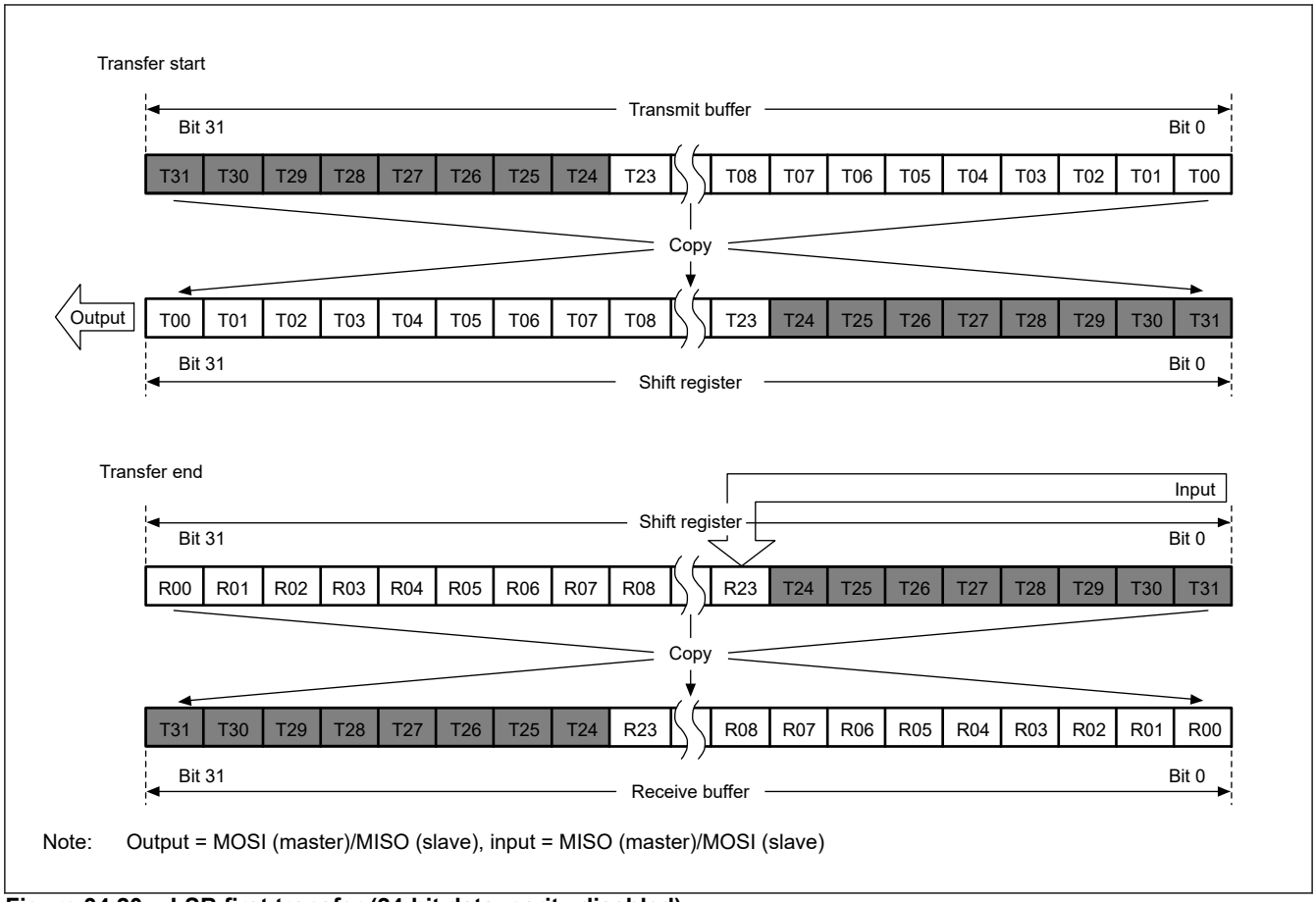


Figure 34.20 LSB first transfer (24-bit data, parity disabled)

34.3.4.3 When Parity is Enabled (SPPE = 1)

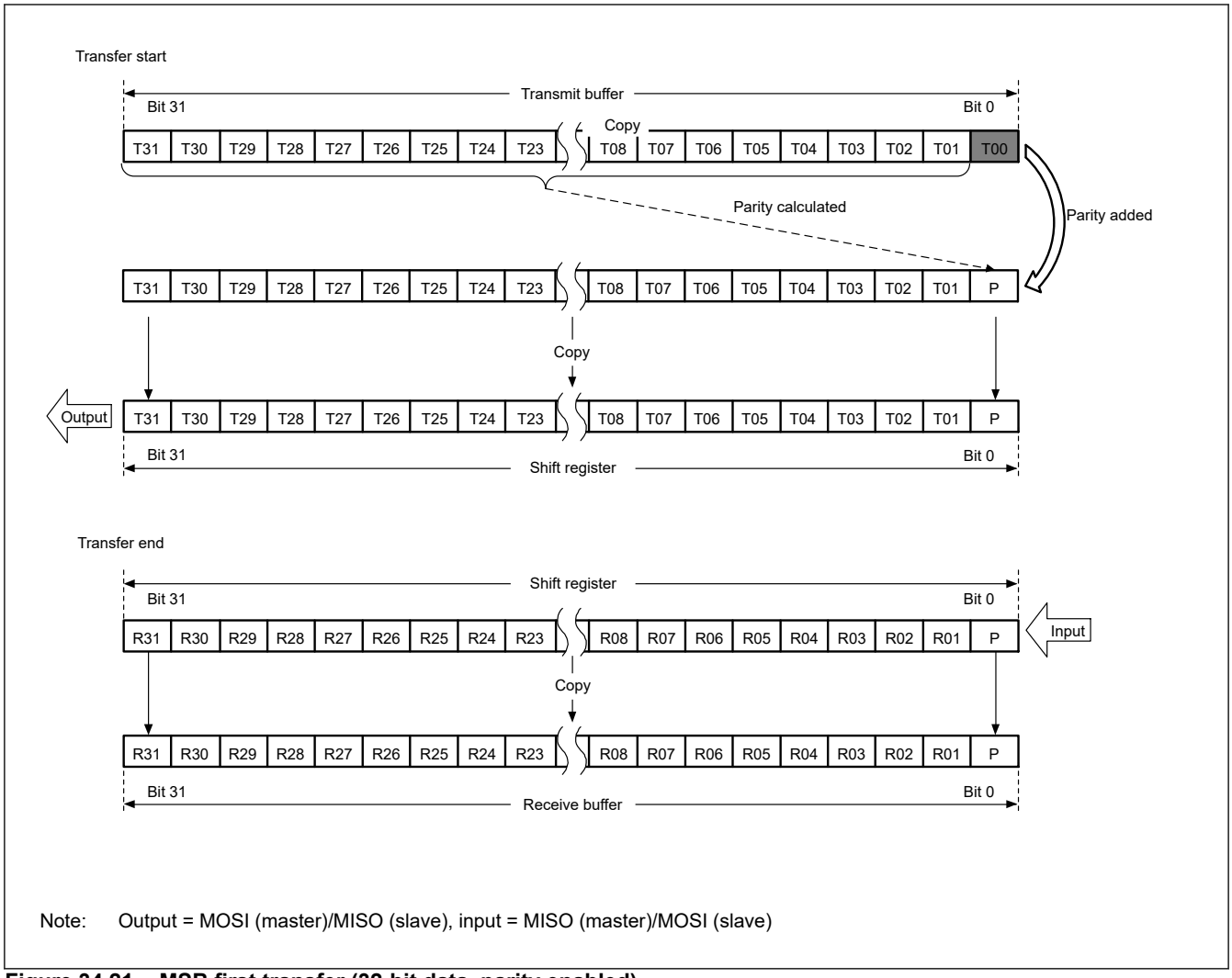
When the parity function is enabled, the LSB of transmit data or received data is converted to a parity bit. The parity bit value is calculated and converted by the hardware.

(1) MSB First Transfer (32-Bit Data)

Figure 34.21 shows operations of the SPDR register and the shift register when the SPI performs 32-bit MSB-first transfer with the parity function enabled.

In transmission, the parity bit (P) value is calculated first from the data value T31 to T01, then the final bit T00 is replaced with the calculated P value, and then transmit data is copied to the shift register. Transmit data is transmitted in the order of T31 → T30 → … → T01 → P.

In reception, received data is stored in bit 0 of the shift register, and is then shifted in units of data. When necessary RSPCK cycles are input and received data is stored from R31 to P, the shift register value is copied to the receive buffer, and at the same time, R31 to P data is checked to determine whether a parity error is present.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

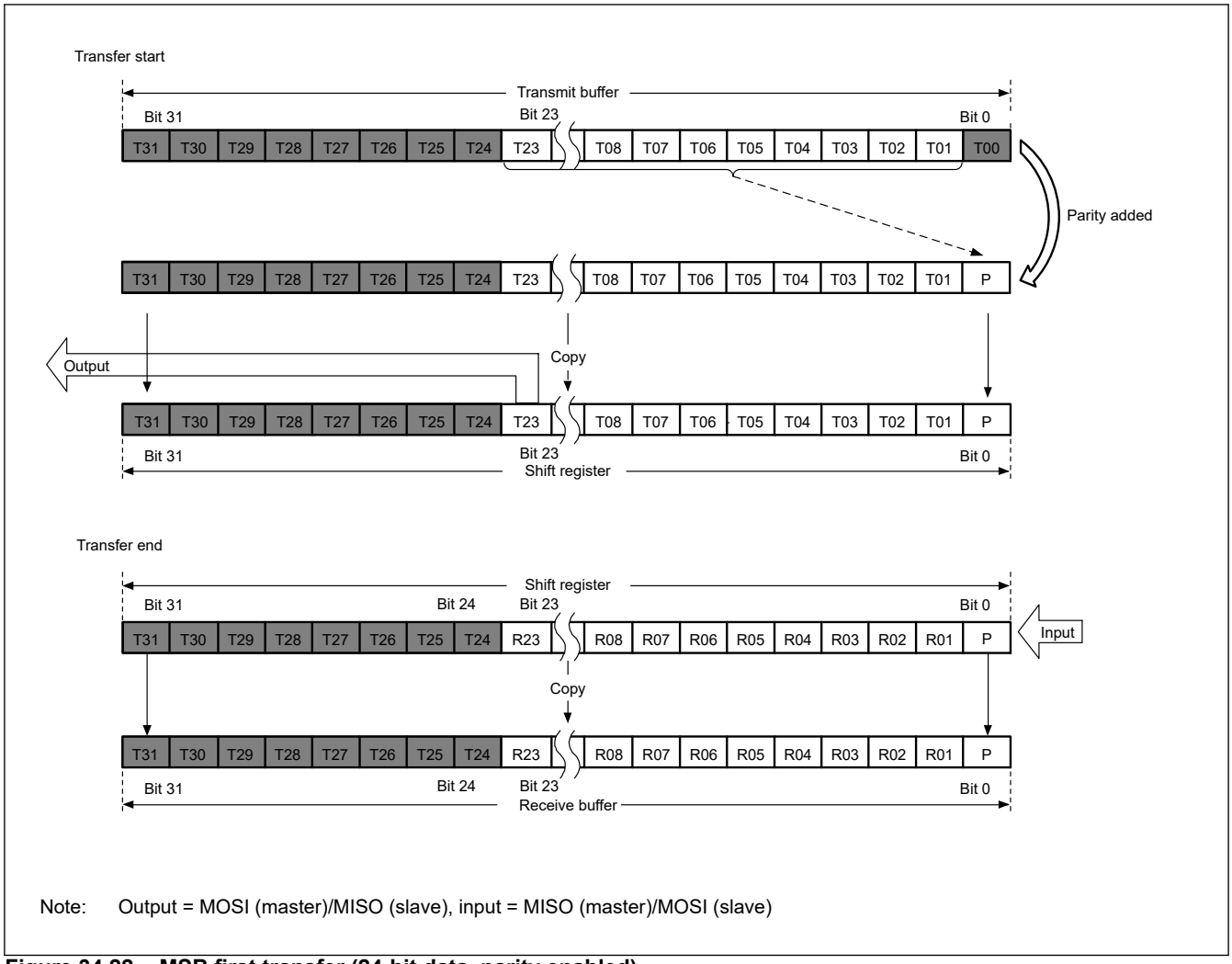
Figure 34.21 MSB first transfer (32-bit data, parity enabled)

(2) MSB First Transfer (24-Bit Data)

As an example of MSB-first transfer of data other than 32 bits with the parity function enabled, Figure 34.22 shows operations of the SPDR register and the shift register when the SPI performs 24-bit data transfer.

In transmission, the parity bit (P) value is calculated first from the data value T23 to T01, then the final bit T00 is replaced with the calculated P value, and then transmit data is copied to the shift register. Transmit data is transmitted in the order of T23 → T22 → … → T01 → P.

In reception, received data is stored in bit 0 of the shift register, and is then shifted in units of data. When necessary RSPCK cycles are input and received data is stored from R23 to P, the shift register value is copied to the receive buffer, and at the same time, R23 to P data is checked to determine whether a parity error is present.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

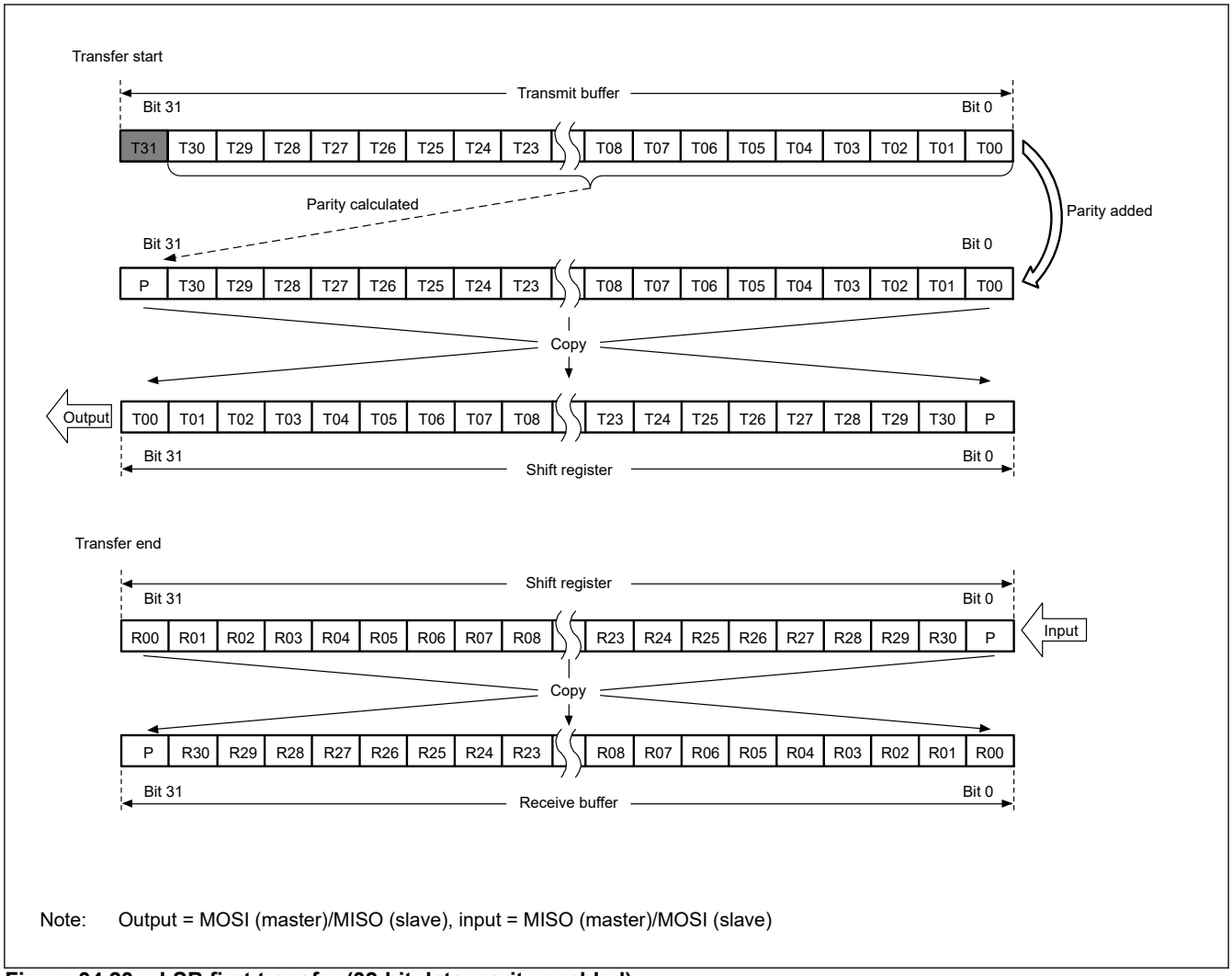
Figure 34.22 MSB first transfer (24-bit data, parity enabled)

(3) LSB First Transfer (32-Bit Data)

Figure 34.23 shows operations of the SPDR register and the shift register when the SPI performs 32-bit LSB-first transfer with the parity function enabled.

In transmission, the parity bit (P) value is calculated first from the data value T30 to T00, then the final bit T31 is replaced with the calculated P value, and then transmit data is copied to the shift register. Transmit data is transmitted in the order of T00 → T01 → … → T30 → P.

In reception, received data is stored in bit 0 of the shift register, and is then shifted in units of data. When necessary RSPCK cycles are input and received data is stored from R00 to P, bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of P to R00, and at the same time, R00 to P data is checked to determine whether a parity error is present.



Note: Output = MOSI (master)/MISO (slave), input = MISO (master)/MOSI (slave)

Figure 34.23 LSB first transfer (32-bit data, parity enabled)

(4) LSB First Transfer (24-Bit Data)

As an example of LSB-first transfer of data other than 32 bits with the parity function enabled, Figure 34.24 shows operations of the SPDR register and the shift register when the SPI performs 24-bit data transfer.

In transmission, the parity bit (P) value is calculated first from the data value T22 to T00, then the final bit T23 is replaced with the calculated P value, and then transmit data is copied to the shift register. Transmit data is transmitted in the order of T00→ T01 → … → T22 → P.

In reception, received data is stored in bit 8 of the shift register, and is then shifted in units of data. When necessary RSPCK cycles are input and received data is stored from R00 to P, bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of P (bit 23) to R00 (bit 0), and at the same time, R00 to P data is checked to determine whether a parity error is present.

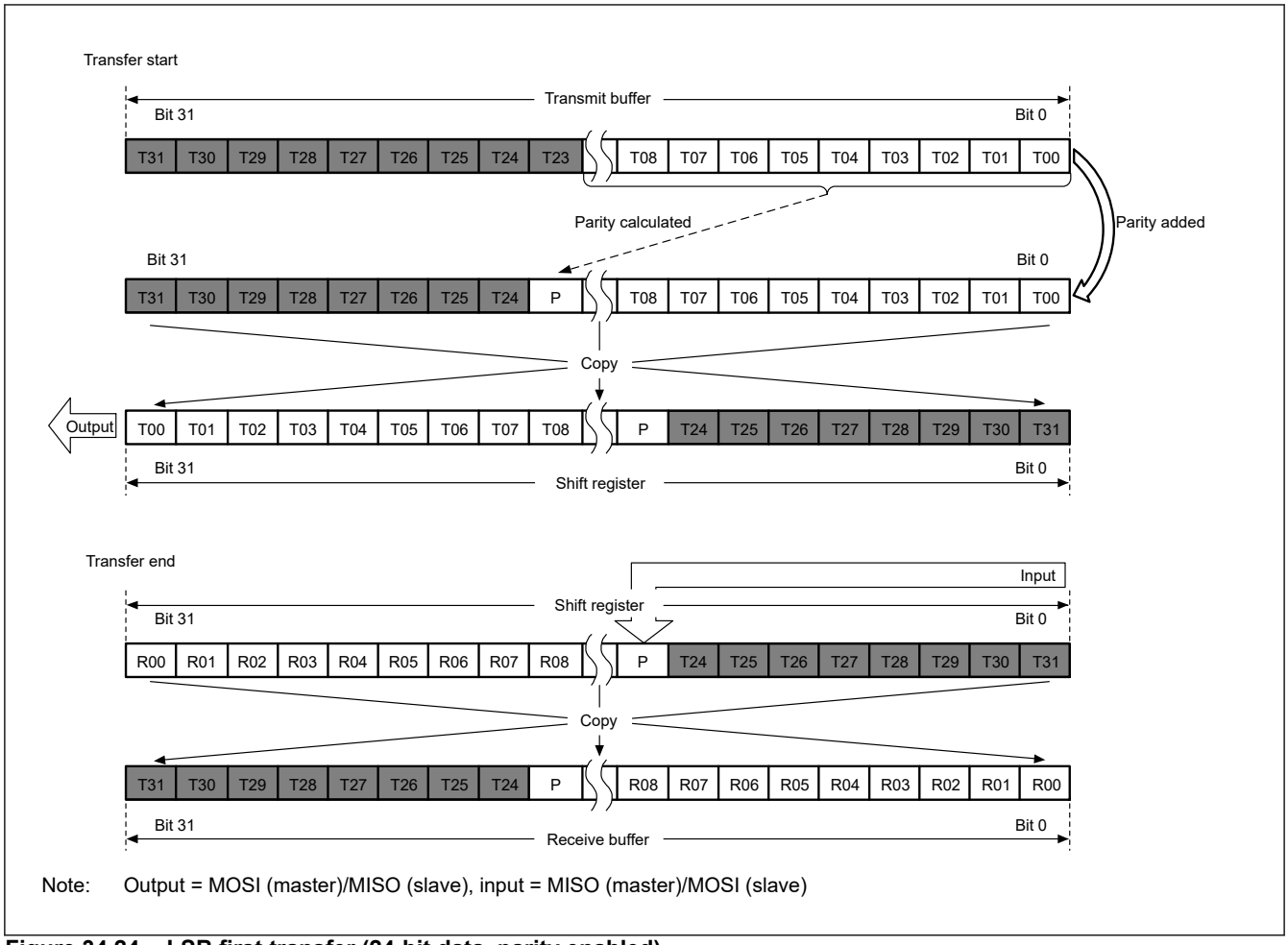


Figure 34.24 LSB first transfer (24-bit data, parity enabled)

34.3.4.4 Byte swap transmission

When byte swapping is enabled, the data in the transmission buffer, swapped in 8-bit units, is copied to the shift register.

(1) 32-bit data length

Figure 34.25 shows the relationship between the SPDR (transmit buffer) and the shift register when transferring data with a 32-bit data length, using a combination of MSB/LSB first and with/without byte swap.

1. **MSB First Transfer (When the byte swap is disabled.)**
   Data (Byte3 [T31 to T24] to Byte0 [T07 to T00]) in the transmit buffer are copied to the shift register.
   Bit values in the shift register are shifted and transmitted in the order of T31 → T30 → … → T00 as transmit data.
2. **MSB First Transfer (When the byte swap is enabled.)**
   Byte values of the transmit buffer (Byte3 [T31 to T24] to Byte0 [T07 to T00]) are reversed in byte units and are copied to the shift register in the order of Byte3 [T07 to T00] to Byte0 [T31 to T24].
   Bit values in the shift register are shifted and transmitted in the order of T07 → T06 → … → T00 → T15 → T14→ … → T08 → T23 → T22 → … → T16 → T31 → T30 → … → T24 as transmit data.
3. **LSB-first transfer (When the byte swap is disabled.)**
   Bit values of the transmit buffer (Byte3 [T31 to T24] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte0 [T00 to T07] to Byte3 [T24 to T31].
   Bit values in the shift register are shifted and transmitted in the order of T00 → T01 → … → T31 as transmit data.
4. **LSB-first transfer (When the byte swap is enabled.)**
   Bit values of each byte of the transmit buffer (Byte3 [T31 to T24] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte3 [T24 to T31] to Byte0 [T00 to T07].
   Bit values in the shift register are shifted and transmitted in the order of T24 → T25 → … → T31 → T16 → T17 → … → T23 → T08 → T09 → … → T15 → T00 → T01 → … → T07 as transmit data.

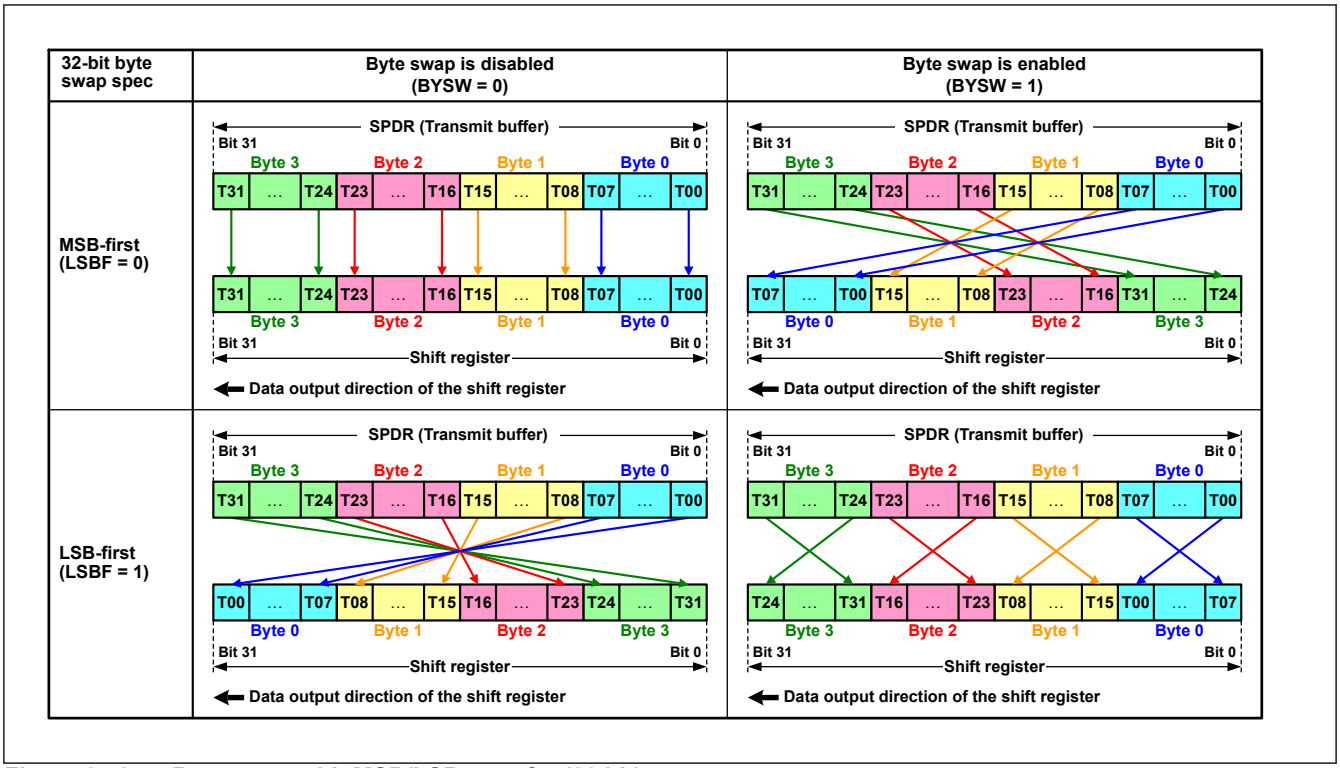


Figure 34.25 Byte swap with MSB/LSB transfer (32 bit)

(2) 16-bit data length

Figure 34.26 shows the relationship between the SPDR (transmit buffer) and the shift register when transferring data with a 16-bit data length, using a combination of MSB/LSB first and with/without byte swap.

1. **MSB First Transfer (When the byte swap is disabled.)**
   Data (Byte1 [T15 to T08] to Byte0 [T07 to T00]) in the transmit buffer are copied to the shift register.
   Bit values in the shift register are shifted and transmitted in the order of T15 → T14 → … → T00 as transmit data.
2. **MSB First Transfer (When the byte swap is enabled.)**
   Byte values of the transmit buffer (Byte1 [T15 to T08] to Byte0 [T07 to T00]) are reversed in byte units and are copied to the shift register in the order of Byte1 [T07 to T00] to Byte0 [T15 to T08].
   Bit values in the shift register are shifted and transmitted in the order of T07 → T06 → … → T00 → T15 → T14→ … → T08 as transmit data.
3. **LSB-first transfer (When the byte swap is disabled.)**
   Bit values of the transmit buffer (Byte1 [T15 to T08] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte0 [T00 to T07] to Byte1 [T08 to T15].
   Bit values in the shift register are shifted and transmitted in the order of T00 → T01 → … → T15 as transmit data.
4. **LSB-first transfer (When the byte swap is enabled.)**
   Bit values of each byte of the transmit buffer (Byte1 [T15 to T08] to Byte0 [T07 to T00]) are reversed in bit units and are copied to the shift register in the order of Byte1 [T08 to T15] to Byte0 [T00 to T07].
   Bit values in the shift register are shifted and transmitted in the order of T08 → T09 → … → T15 → T00 → T01 → … → T07 as transmit data.



Figure 34.26 Byte swap with MSB/LSB transfer (16 bit)

Note: When using the byte swap, set 16 bits or 32 bits to the data length (SPCMD.SPB[4:0] setting).
If setting the other length, the behavior is not guaranteed.

Note: When the byte swap is valid, set the parity function as invalid (SPCR.SPPE bit = 0) .
If setting the parity function as valid (SPPE bit = 1), the behavior is not guaranteed.

Note: Set SPDCR.BYSW bit, when SPCR.SPE bit is 0. If rewriting BYSW bit, when SPE bit is 1, the behavior after it is not guaranteed.

34.3.4.5 Byte swap reception

When byte swap is enabled, the data in the shift register, swapped in 8-bit units, is copied to the receive buffer.

(1) 32-bit data length

Figure 34.27 shows the relationship between the shift register and SPDR (reception buffer) when transferring data with a 32-bit data length, using a combination of MSB/LSB first and with/without byte swap.

1. **MSB First Transfer (When the byte swap is disabled.)**
   The first received data (R31) is stored in bit 0 of the shift register, and received data is shifted in the order of R31 → R30 → … → R00.
   When necessary RSPCK cycles are input and data is stored from Byte3 [R31 to R24] to Byte0 [R07 to R00], the shift register value is copied to the receive buffer.
2. **MSB First Transfer (When the byte swap is enabled.)**
   The first received data (R07) is stored in bit 0 of the shift register, and received data is shifted in the order of R07 → R06 → … → R00 → R15 → R14 → … → R08 → R23 → R22 → … → R16 → R31 → R30 → … → R24.
   When necessary RSPCK cycles are input and data is stored from Byte0 [R07 to R00] to Byte3 [R31 to R24], byte values in the shift register are reversed in byte units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].
3. **LSB-first transfer (When the byte swap is disabled.)**
   The first received data (R00) is stored in bit 0 of the shift register, and received data is shifted in the order of R00 → R01 → … → R31.

When necessary RSPCK cycles are input and data is stored from Byte0 [R00 to R07] to Byte3 [R24 to R31], bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].

4. **LSB-first transfer (When the byte swap is enabled.)**
   The first received data (R24) is stored in bit 0 of the shift register, and received data is shifted in the order of R24 → R25 → … → R31 → R16 → R17 → … → R23 → R08 → R09 → … → R15 → R00 → R01 → … → R07.
   When necessary RSPCK cycles are input and data is stored from Byte3 [R24 to R31] to Byte0 [R00 to R07], bit values of each byte in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte3 [R31 to R24] to Byte0 [R07 to R00].

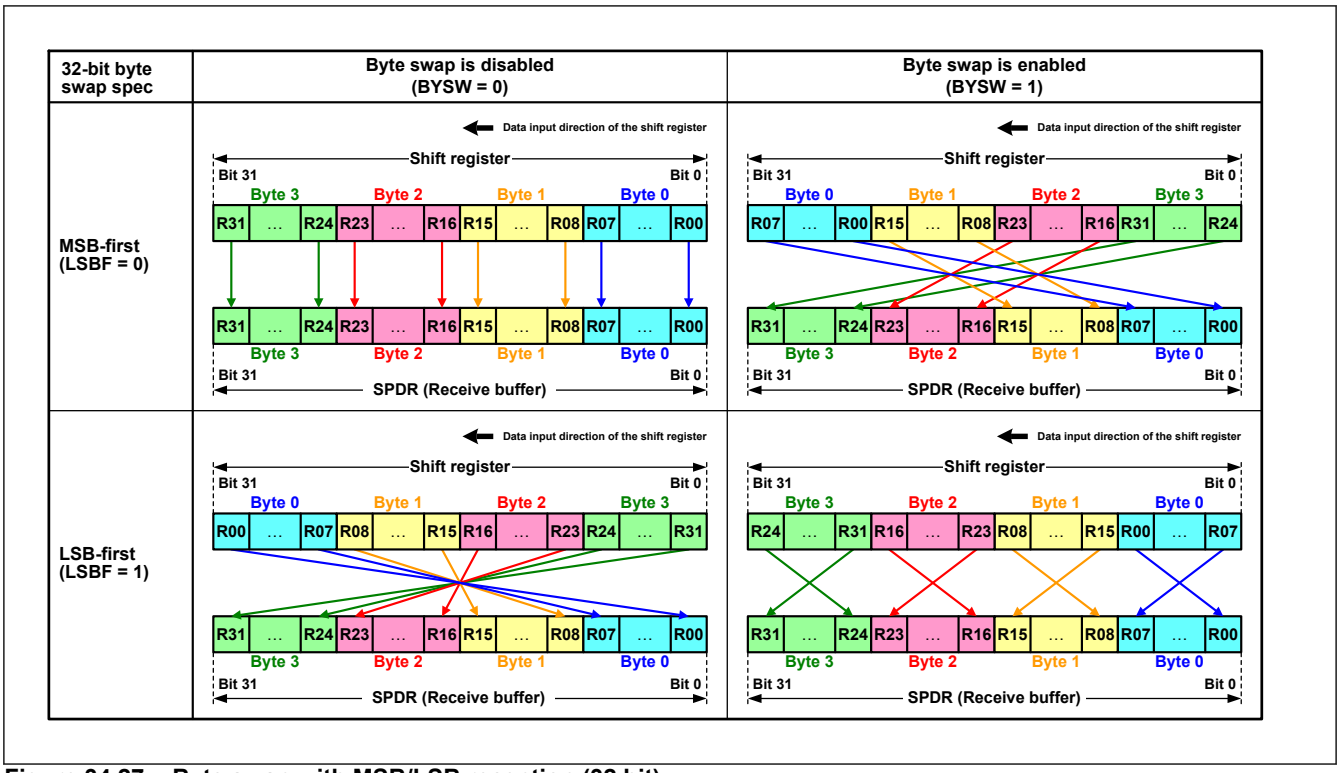


Figure 34.27 Byte swap with MSB/LSB reception (32 bit)

(2) 16-bit data length

Figure 34.28 shows the relationship between the shift register and SPDR (reception buffer) when transferring data with a 16-bit data length, using a combination of MSB/LSB first and with/without byte swap.

1. **MSB First Transfer (When the byte swap is disabled.)**
   The first received data (R15) is stored in bit 0 of the shift register, and received data is shifted in the order of R15 → R14 → … → R00.
   When necessary RSPCK cycles are input and data is stored from Byte1 [R15 to R08] to Byte0 [R07 to R00], the shift register value is copied to the receive buffer.

2. **MSB First Transfer (When the byte swap is enabled.)**
   The first received data (R07) is stored in bit 0 of the shift register, and received data is shifted in the order of R07 → R06 → … → R00 → R15 → R14 → … → R08.
   When necessary RSPCK cycles are input and data is stored from Byte0 [R07 to R00] to Byte1 [R15 to R08], byte values in the shift register are reversed in byte units and are copied to the receive buffer in the order of Byte1 [R15 to R08] to Byte0 [R07 to R00].

3. **LSB-first transfer (When the byte swap is disabled.)**
   The first received data (R00) is stored in bit 15 of the shift register, and received data is shifted in the order of R00 → R01 → … → R15.
   When necessary RSPCK cycles are input and data is stored from Byte0 [R00 to R07] to Byte1 [R08 to R15], bit values in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte1 [R15 to R08] to Byte0 [R07 to R00].

4. **LSB-first transfer (When the byte swap is enabled.)**
   The first received data (R08) is stored in bit 0 of the shift register, and received data is shifted in the order of R08 → R09 → … → R15 → R00 → R01 → … → R07.
   When necessary RSPCK cycles are input and data is stored from Byte1 [R08 to R15] to Byte0 [R00 to R07], bit values of each byte in the shift register are reversed in bit units and are copied to the receive buffer in the order of Byte1 [R15 to R08] to Byte0 [R07 to R00].

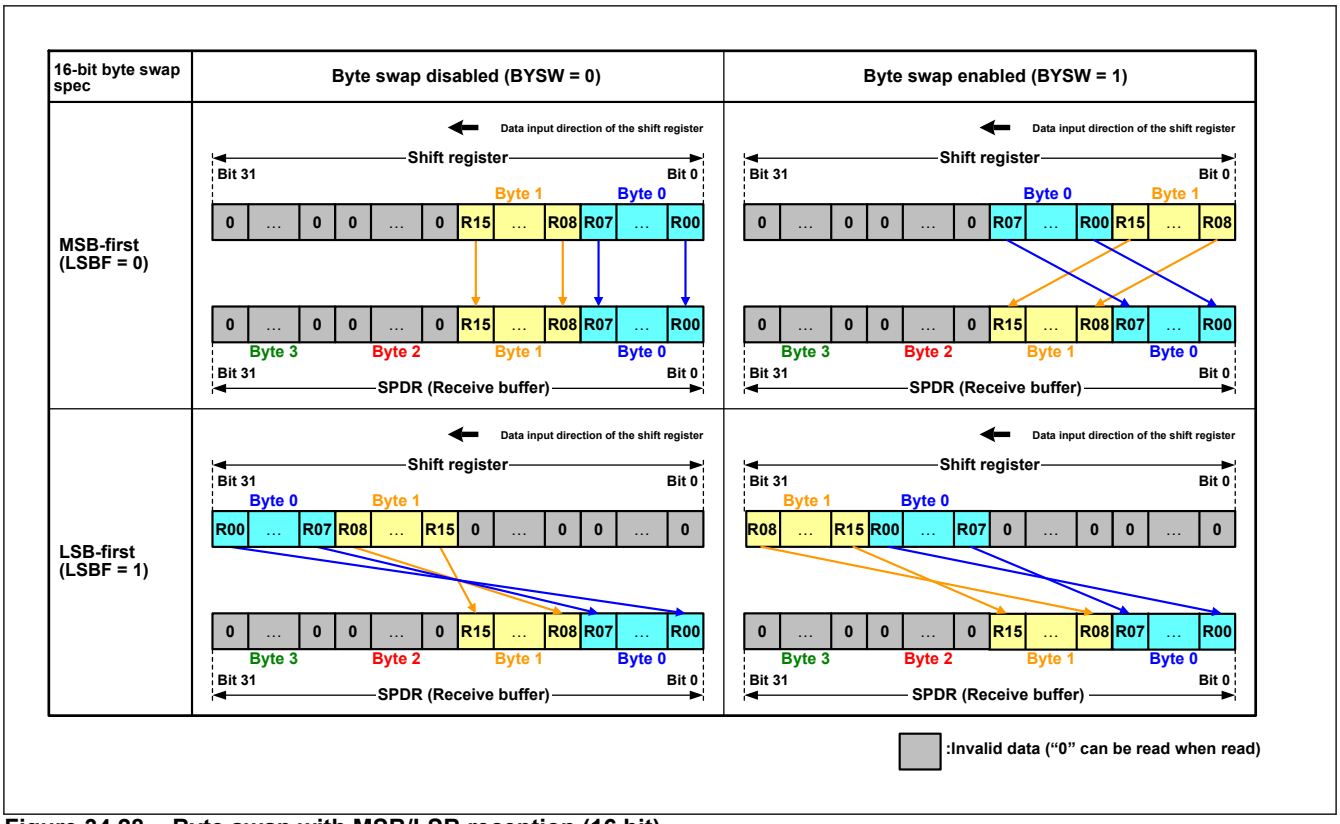


Figure 34.28 Byte swap with MSB/LSB reception (16 bit)

Note: When using the byte swap, set 16 bits or 32 bits to the data length (SPCMD.SPB[4:0] setting) .
If setting the other length, the behavior is not guaranteed.

Note: When the byte swap is valid, set the parity function as invalid (SPCR.SPPE bit = 0).
If setting the parity function as valid (SPPE bit = 1), the behavior is not guaranteed.

Note: Set SPDCR2.BYSW bit, when SPCR.SPE bit is 0. If rewriting BYSW bit, when SPE bit is 1, the behavior after it is not guaranteed.

34.3.5 Transfer Format (Frame Format)

34.3.5.1 CPHA = 0

Figure 34.29 shows an example of transfer format of 8-bit data serial transfer when SPCMD.CPHA = 0. However, when the SPI is in slave mode (SPCR.MSTR = 0) with the CPHA bit set to 0, clock synchronous operation (SPCR.SPMS = 1) is not guaranteed. In Figure 34.29, RSPCK (CPOL = 0) is the RSPCK signal waveform when SPCMD.CPOL = 0, and RSPCK (SPCMD.CPOL = 1) is the RSPCK signal waveform when SPCMD.CPOL = 1. The sampling timing shows the time when the SPI latches serial transfer data in the shift register. Input and output directions of each signal depend on SPI settings. (See section 34.3.2. SPI Pin Control.)

When SPCMD.CPHA = 0, output of valid data to the MOSI signal and valid data drive to the MISO signal start at the SSL signal assertion timing. The first RSPCK signal change timing after the SSL signal is asserted is the first transfer data latch timing. After this timing, data is sampled in each RSPCK cycle. The MOSI signal and the MISO signal change always after the half RSPCK cycles of the transfer data latch timing. The set CPOL bit value does not affect the RSPCK signal operation timing but affects only the signal polarity.

Period t1 is the period (RSPCK delay) from SSL signal assertion to RSPCK oscillation. Period t2 is the period (SSL negation delay) from RSPCK oscillation stop to SSL signal negation. Period t3 is the period (next-access delay) to inhibit SSL signal assertion for the next transfer after serial transfer is completed. Periods t1, t2, and t3 are controlled by the master device in the SPI system. For t1, t2, and t3 when this LSI (SPI) is in master mode, see section 34.3.13.1. Master Mode Operation.

[In the Motorola-SPI case]

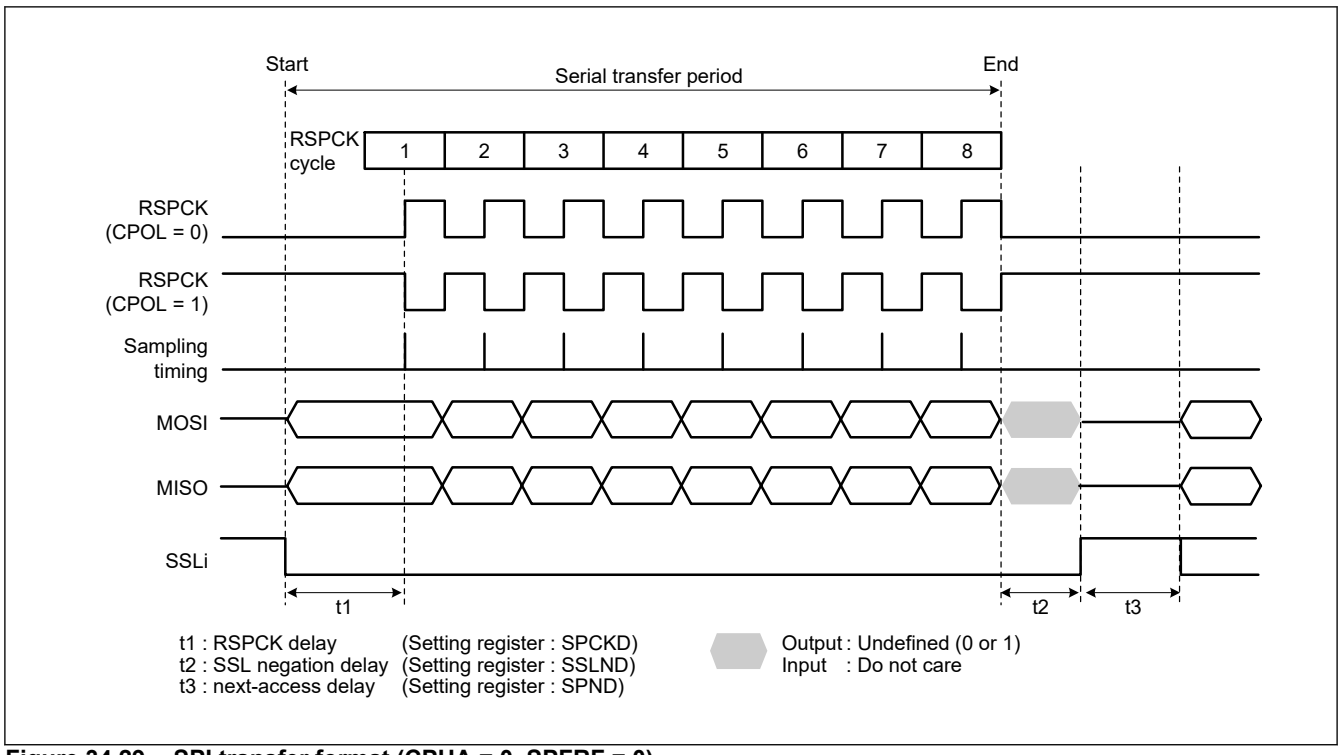


Figure 34.29 SPI transfer format (CPHA = 0, SPFRF = 0)

34.3.5.2 CPHA = 1

Figure 34.30 shows an example of transfer format of 8-bit data serial transfer when SPCMD.CPHA = 1. However, when SPCR.SPMS = 1, communication is performed by using only RSPCK, MOSI, and MISO signals without using the SSL signal. In Figure 34.30, RSPCK (CPOL = 0) is the RSPCK signal waveform when SPCMD.CPOL = 0, and RSPCK (SPCMD.CPOL = 0) is the RSPCK signal waveform when SPCMD.CPOL = 0. The sampling timing shows the time when the SPI latches serial transfer data in the shift register. Input and output directions of each signal depend on SPI mode (master or slave mode). (See section 34.3.2. SPI Pin Control.)

When SPCMD.CPHA = 1, invalid data drive to the MISO signal starts at the SSL signal assertion timing. Output of valid data to the MOSI and MISO signals starts at the first RSPCK signal change timing after the SSL signal is asserted.

After this timing, data is updated in each RSPCK cycle. Transfer data is always latched after 1/2 RSPCK cycle of this timing. The set CPOL bit value does not affect the RSPCK signal operation timing but affects only the signal polarity.

Periods t1, t2, and t3 are the same as when CPHA = 0. For t1, t2, and t3 when this LSI (SPI) is in master mode, see section 34.3.13.1. Master Mode Operation.

[In the Motorola-SPI case]

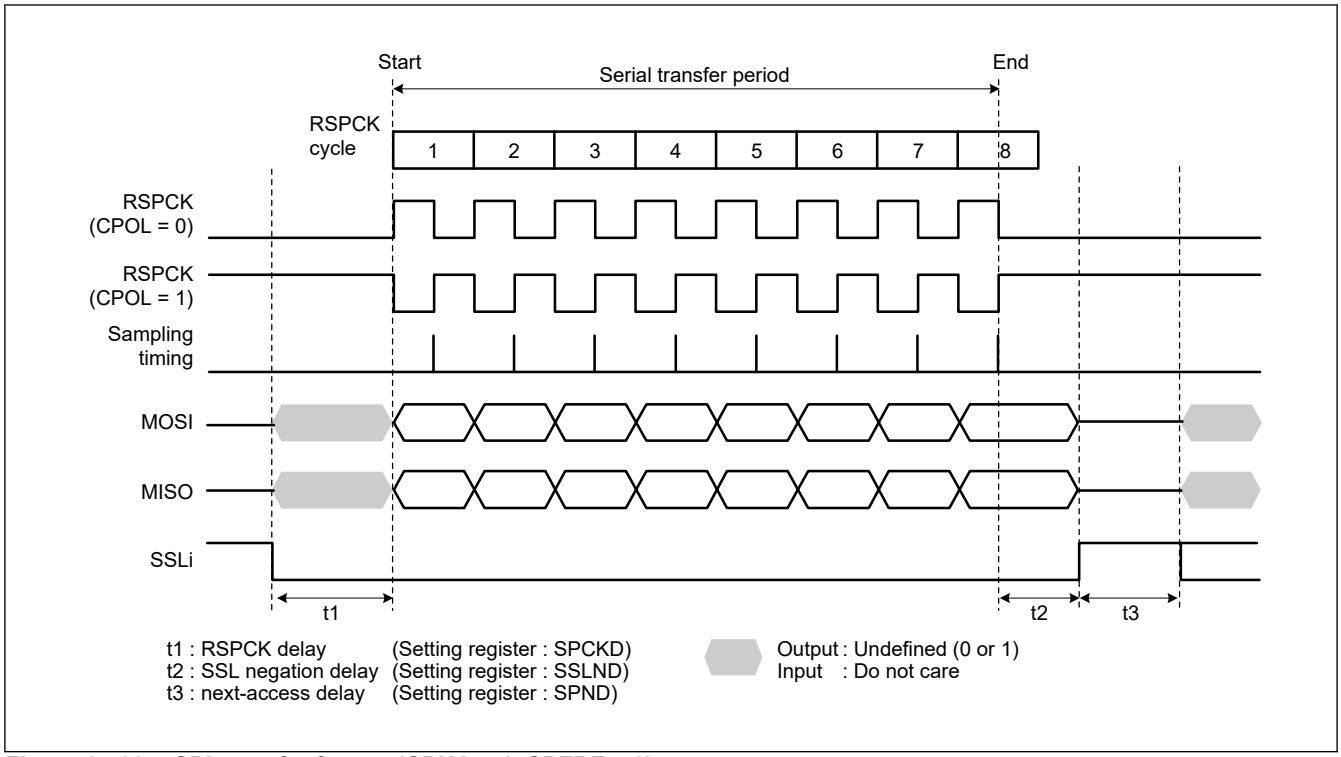


Figure 34.30 SPI transfer format (CPHA = 1, SPFRF = 0)

34.3.6 Communications Operating Mode

Transmit-Receive serial communication, transmit-only operation, and Receive-only operation are selected by setting the Communication Mode Select bits (TXMD[1:0]) of the SPCR register.

SPDR access described in Figure 34.31, Figure 34.32, and Figure 34.33 shows an access to the SPDR register. W shows a write cycle.

34.3.6.1 Transmit-Receive Serial Communication (TXMD[1:0] = 00b)

Figure 34.31 shows an example of operation when the communication mode select bit (TXMD[1:0]) in the SPI control register (SPCR) is set to 00b. In the example in Figure 34.31, the SPI performs 8-bit data serial transfer with the settings of RTRG = FIFO stage - 1, SPDCR2.TTRG = 0, SPCMD.CPHA = 1, and SPCMD.CPOL = 0. Numbers under the RSPCK waveform show the number of RSPCK cycles (number of transfer bits).

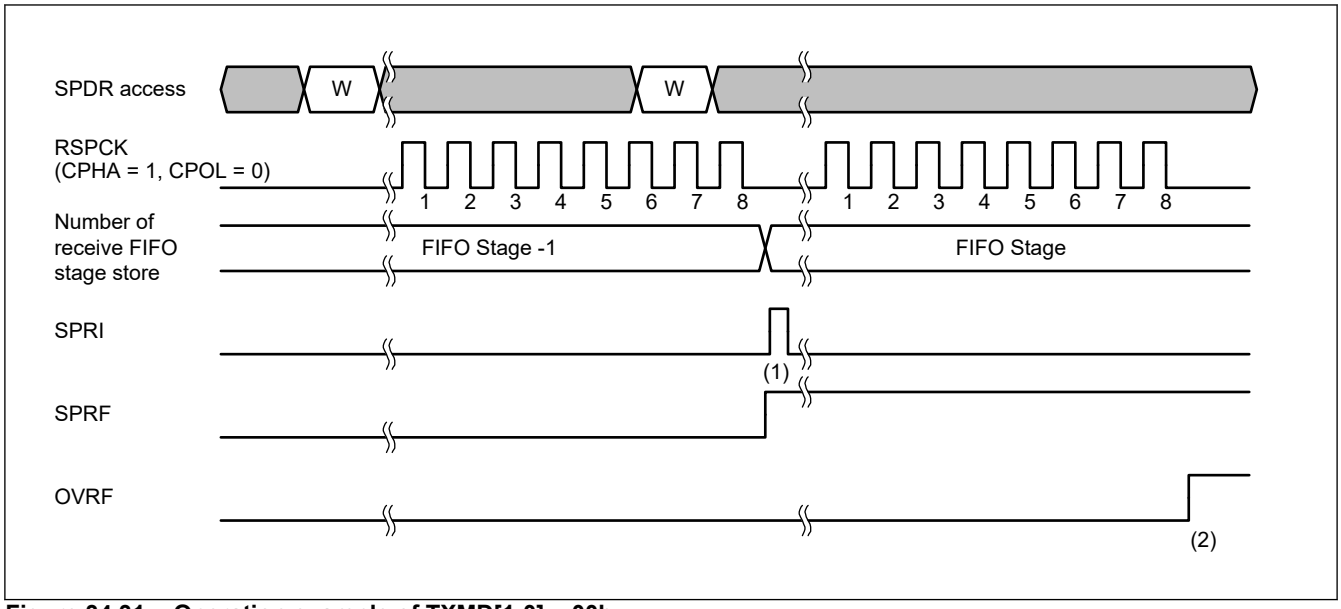


Figure 34.31 Operation example of TXMD[1:0] = 00b

The following describes operation of flags at timings (1) and (2) in the figure above.

(1) When serial transfer ends while the number of SPDR receive buffer store matches the number of frames set in SPDCR2.RTRG, the SPI generates a receive buffer full interrupt request SPRI (setting the SPSR.SPRF flag to 1) and copies the received data in the shift register to the receive buffer.

(2) When serial transfer ends with data for the number of FIFO stages stored in the SPDR receive buffer, the SPI sets the OVRF flag in the SPSR register to 1 and discards the received data in the shift register.

In Transmit-Receive serial communication (TXMD[1:0] = 00b), transmit data is transmitted and receive data is received. Therefore, the SPRF flag and the OVRF flag are set to 1 at timings (1) and (2) respectively.

34.3.6.2 Transmit-Only Serial Communication (TXMD[1:0] = 01b)

Figure 34.32 shows an example of operation when the communication mode select bits (TXMD[1:0]) in the SPCR register are set to 01b. In the example in Figure 34.32, the SPI performs 8-bit data serial transfer with the settings of SPDCR2.TTRG[1:0] = 00b, SPDCR2.RTRG[1:0] = 00b, SPCMD.CPHA = 1, and SPCMD.CPOL = 0. Numbers under the RSPCK waveform show the number of RSPCK cycles (number of transfer bits).

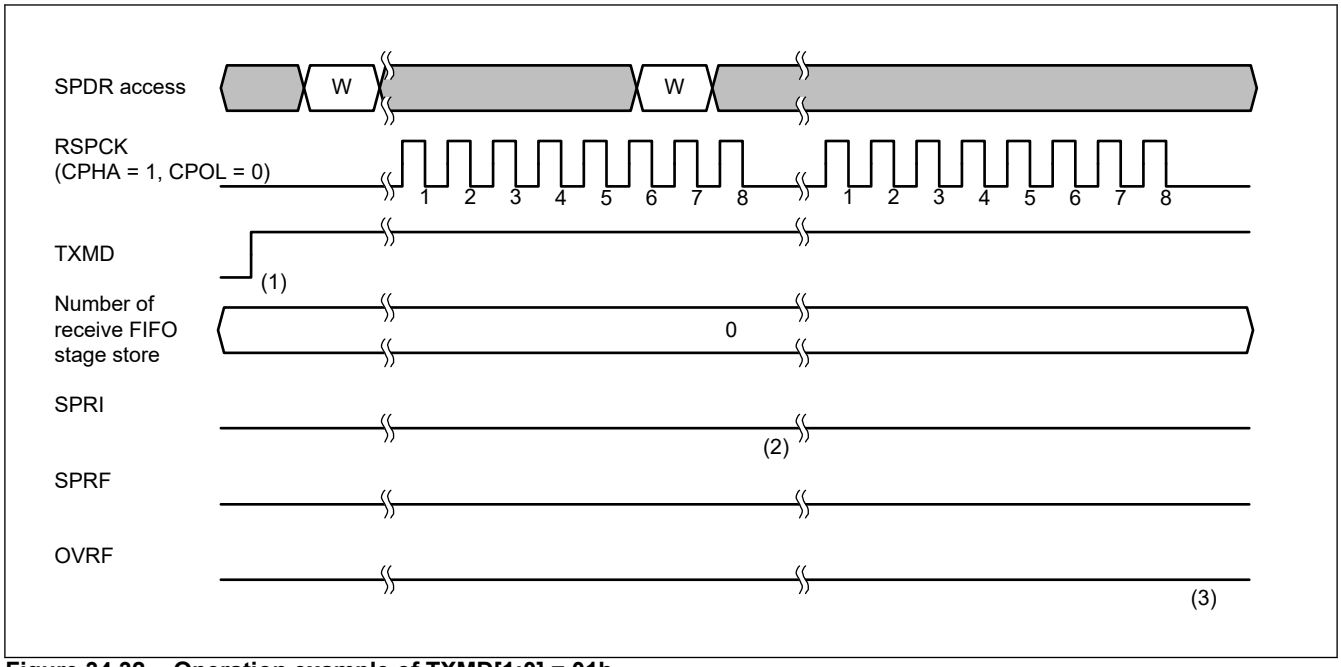


Figure 34.32 Operation example of TXMD[1:0] = 01b

The following describes operation of flags at timings (1) to (3) in the figure above.

(1) Before setting TXMD[1:0] to 01b (transmit-only serial communication), confirm that the receive buffer is empty and that the SPRF and OVRF flags in the SPSR register are 0.

(2) In transmit-only serial communication (TXMD[1:0] = 01b), when serial transfer ends without receiving data in the receiving FIFO of SPDR, the SPRF flag remains 0 and the shift register data is not copied to the receive buffer.

(3) The OVRF flag remains 0 even after serial transfer ends because previously received data is not remaining in the SPDR's receive buffer, and the shift register data is not copied to the receive buffer.

In transmit-only serial communication (TXMD[1:0] = 01b), transmit data is transmitted but receive data is not received.

Therefore, the SPRF flag and the OVRF flag remain 0 at each timing of (1) to (3).

34.3.6.3 Receive-Only Serial Communication (TXMD[1:0] = 10b)

Figure 34.33 shows an example of operation when the communication mode select bit (TXMD[1]) in the SPI control register (SPCR) is set to 1. In the example in Figure 34.33, the SPI performs 8-bit data serial transfer with the settings of SPDCR2.TTRG[1:0] = 0, SPDCR2.RTRG[1:0] = FIFO stage - 1, SPCMD.CPHA = 1, and CPOL in SPCMD = 0. Numbers under the RSPCK waveform show the number of RSPCK cycles (number of transfer bits).

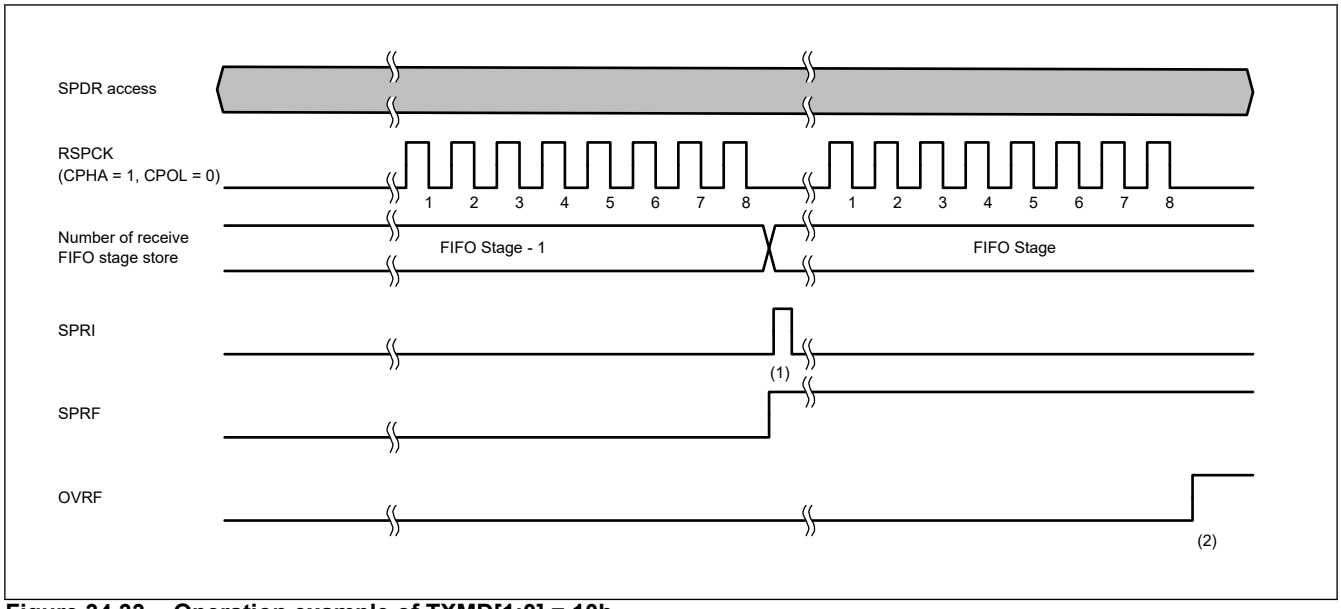


Figure 34.33 Operation example of TXMD[1:0] = 10b

The following describes operation of flags at timings (1) and (2) in the figure above.

(1) When serial transfer ends while the number of SPDR receive buffer store matches the number of frames set in SPDCR2.RTRG, the SPI generates a receive buffer full interrupt request SPRI (setting the SPSR.SPRF flag to 1) and copies the received data in the shift register to the receive buffer.

(2) When serial transfer ends with data for the number of FIFO stages stored in the SPDR receive buffer, the SPI sets the OVRF flag in the SPSR.SPI register to 1 and discards the received data in the shift register.

34.3.7 Transmission Buffer Empty/Receive Buffer Full Interrupt

Figure 34.34 show an example of operation of SPI transmit buffer empty interrupt (SPTI) and SPI receive buffer full interrupt (SPRI) respectively. "SPDR access" described in Figure 34.34 and shows an access to the SPDR register. W shows a write cycle and R shows a read cycle. In each example of operation, the SPI performs 8-bit data serial transfer with the settings of SPCR.TXMD[1:0] = 00b, SPDCR2.TTRG = 0, SPDCR2.RTRG = 0, SPCMD.CPHA = 0 (Figure 34.34), and SPCMD.CPOL = 0. Numbers under the RSPCK waveform show the number of RSPCK cycles (number of transfer bits).

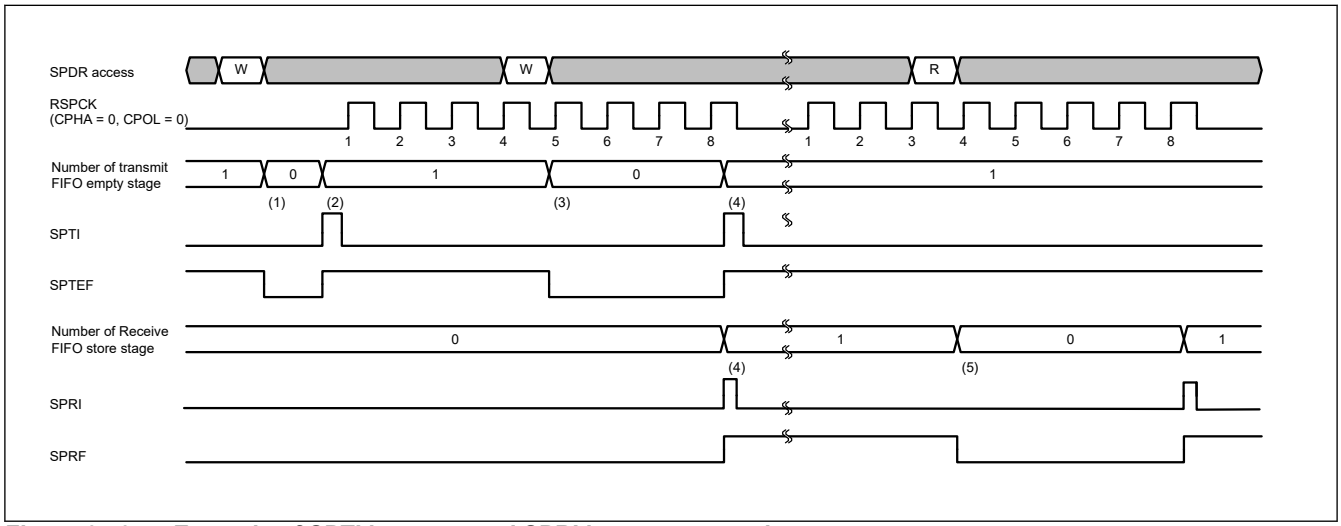


Figure 34.34 Example of SPTI interrupt and SPRI interrupt operations

The following describes operation at timings (1) to (5) in the figure above.

(1) When transmit data is written to SPDR before the data for the next transfer is not set to the FIFO of the SPDR, the SPI writes data to the transmission buffer. When transmit data is written to SPDR in one processing routine using DMAC, the SPSR.SPTEF flag is cleared to 0 at the last access.

(2) When the shift register is empty, the SPI copies transmit buffer data to the shift register. At this time, if transmit FIFO empty stage number > TTRG value, then the SPI generates a transmit buffer empty interrupt request (SPTI) and sets the SPTEF flag to 1. How to start serial transfer depends on SPI mode. (See section 34.3.13. SPI Operation and section 34.3.14. Clock Synchronous Operation.)

(3) When transmit data is written to SPDR upon a transmit buffer empty interrupt request (SPTI) or in the transmit buffer empty determination processing due to the SPTEF flag, the SPI writes data to the transmit buffer. When the transmit data is written to SPDR in one processing routine using DMAC, the SPTEF flag is cleared to 0 at the last access. The SPI does not copy transmit buffer data to the shift register because the shift register contains serial transfer data.

(4) When serial transfer ends (data sampling clock edge in the final bit is detected) while the receive buffer of SPDR > FIFO stage number, the SPI copies the received data in the shift register to the receive buffer. At this time, a receive buffer full interrupt request is generated and the SPRF flag is set to 1. When serial transfer ends, the shift register becomes empty. If the next transfer data is set in the transmit FIFO before serial transfer ends, the SPI set the SPTEF flag to 1 and copies transmit buffer data to the shift register. Even if received data is not copied from the shift register to the receive buffer while an overrun error is present, the SPI determines that the shift register is empty at the end of serial transfer, enabling data transfer from the transmit buffer to the shift register.

(5) When SPDR is read upon a receive buffer full interrupt request (SPRI) or in the receive buffer full determination processing due to the SPRF flag, received data can be read. If the received data is read from SPDR in one processing routine using DMAC, the SPRF flag is cleared to 0 at the last access.

When transmit data is written to the SPDR register while no empty stages in the transmit FIFO, the SPI does not update transmit buffer data. Write transmit data to the SPDR register upon a transmit buffer empty interrupt request or check the empty or not the transmission buffer by the SPTEF flag.

While SPCR.SPTIE = 1, when the SPI is disabled (SPCR.SPE = 0), an SPI transmit buffer empty interrupt is generated. However, SPI transmit buffer empty interrupt is inhibited by disabling SPI transmit buffer empty interrupt (SPCR.SPTIE = 0) at the same time as SPE bit setting.

When serial transfer ends while data is stored in the receive FIFO for the number of FIFO stages, the SPI detects an overrun error without copying data from the shift register to the receive buffer (as described in section 34.3.10. Error Detection). To prevent overrun of received data, read received data before the next serial transfer ends.

Transmit buffer state and receive buffer state can be monitored by an SPI transmit buffer empty interrupt and an SPI receive buffer full interrupt or a corresponding interrupt flag or by the SPTEF and SPRF flags.

34.3.8 Idle Interrupt

The idle interrupt during master mode operation is when the SPCP[2:0] of the SPSSR register becomes 000b (start of sequence control), the IDLNF flag in the SPSR register is set to 1 and an idle interrupt request is made during master mode operation. An interrupt request is also made by clearing the SPCR.SPE bit to 0.

(1) In the Motorola-SPI case

Figure 34.35 shows an example of idle interrupt operation during normal operation.

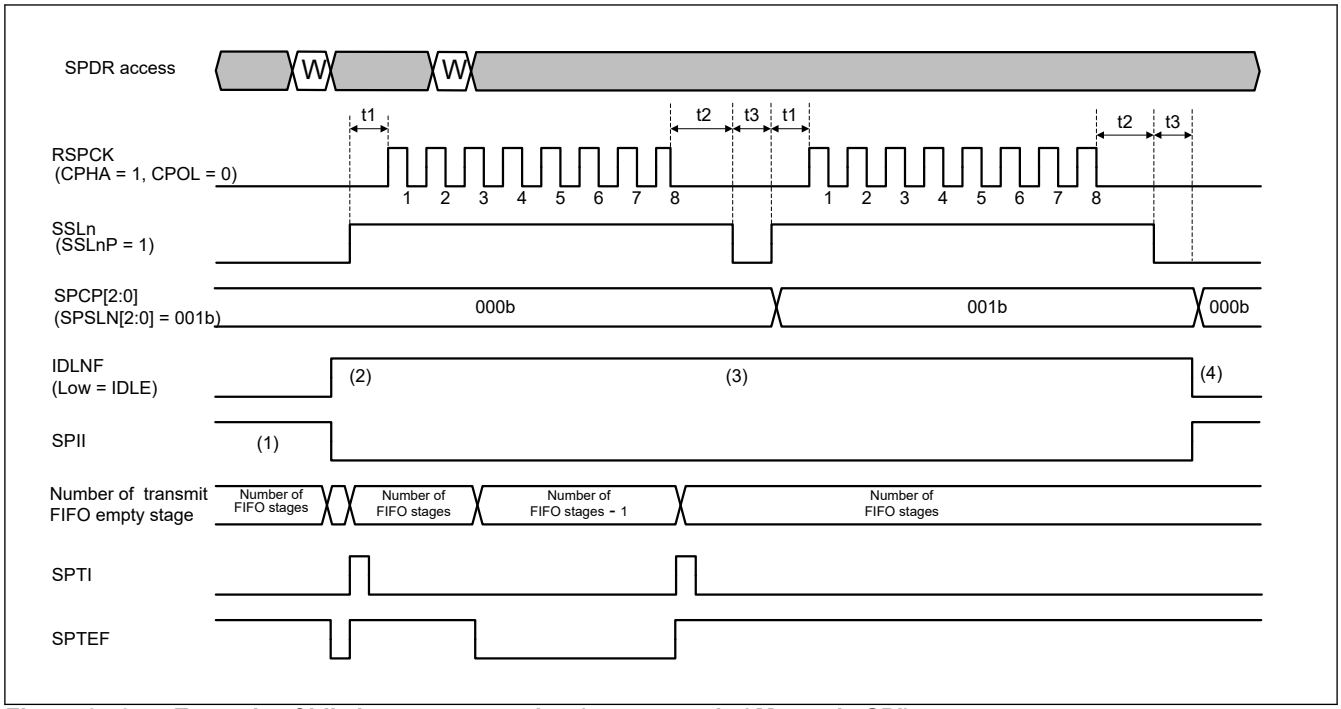


Figure 34.35 Example of idle interrupt operation (master mode / Motorola-SPI)

(1) At the start of transmission, if the next transfer data is not set in the transmission buffer, the IDLNF flag is 0 (IDLE). Writing transmit data makes sets the IDLNF flag to 1 (BUSY). When the SPIIE bit in the SPI control register (SPCR) is set to 1 before transmit data is written, interrupt processing is required before transmission start. For this reason, set the SPIIE bit to 0 before starting transmission.

(2) After transmission has started, the IDLNF flag remains 1 (BUSY) regardless of the transmit buffer state.

(3) The SPCP[2:0] bits change the command to the next command the end of t3 cycle. When the next command is not 000b, the IDLNF flag remains unchanged even when the next transmit data has not been written.

(4) The IDLNF flag is cleared to 0 (IDLE) the end of t3 cycle because the next command is 000b and the next transmit data is not present. When the SPIIE bit is 1 currently, an SPII interrupt is output.

(2) In the TI-SSP case

Figure 34.36 shows an example of idle interrupt operation during normal operation.

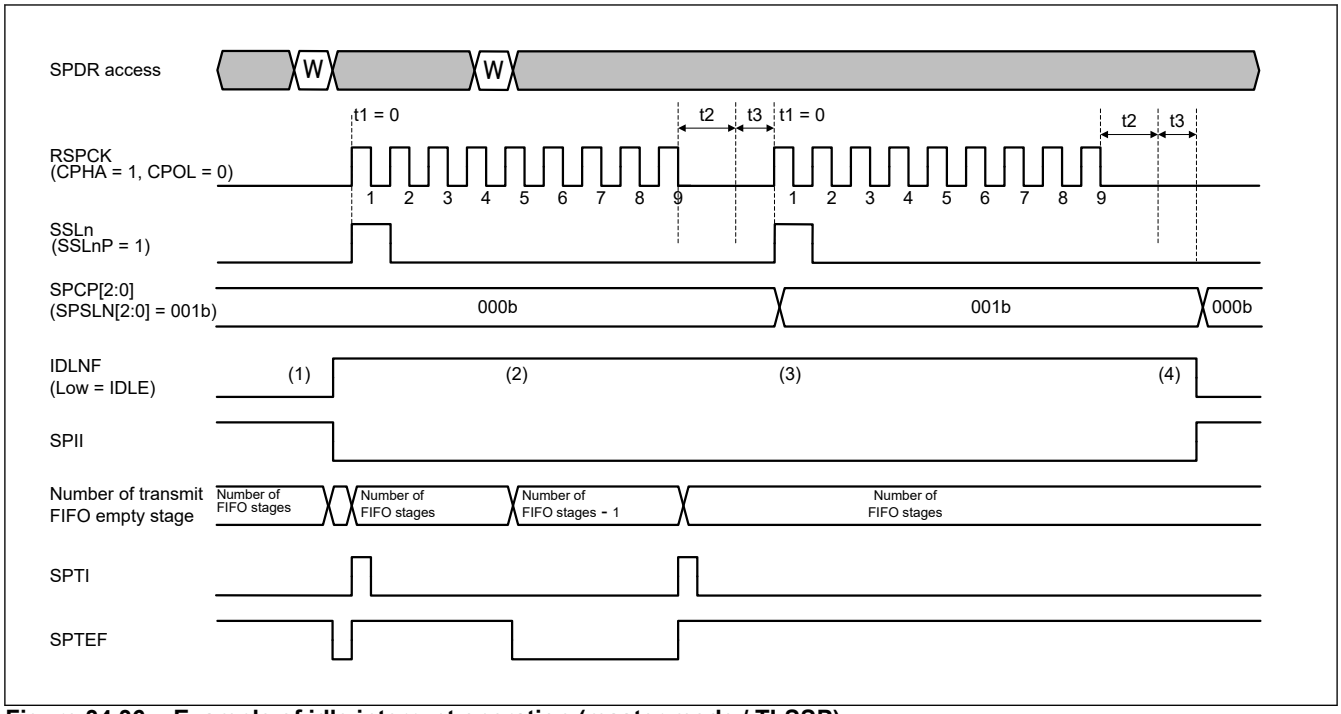


Figure 34.36 Example of idle interrupt operation (master mode / TI-SSP)

(1) At the start of transmission, if the next transfer data is not set in the transmission buffer, the IDLNF flag is 0 (IDLE). Writing transmit data makes sets the IDLNF flag to 1 (BUSY). When the SPCR.SPIIE bit is set to 1 before transmit data is written, interrupt processing is required before transmission start. For this reason, set the SPIIE bit to 0 before starting transmission.

(2) After transmission has started, the IDLNF flag remains 1 (BUSY) regardless of the transmit buffer state.

(3) The SPCP[2:0] bits change the command to the next command the end of t3 cycle. When the next command is not 000b, the IDLNF flag remains unchanged even when the next transmit data has not been written.

(4) The IDLNF flag is cleared to 0 (IDLE) the end of t3 cycle because the next command is 000b and the next transmit data is not present. When SPCR.SPIIE = 1 currently, an SPII interrupt is output.

34.3.9 Communication End Interrupt

34.3.9.1 Transmit-Receive/Transmit-only in Master Mode

Refer to the description of the CENDF bit in section 34.2.19. SPTFSR : SPI Transfer FIFO Status Register and section 34.2.18. SPSR : SPI Status Register for the setting/clearing conditions of the communication completion flag during Transmit-Receive/Transmit-only in Master Mode.

(1) In the Motorola-SPI case

Figure 34.37 shows an example of communication end interrupt operation during transmit-receive/transmit master mode.

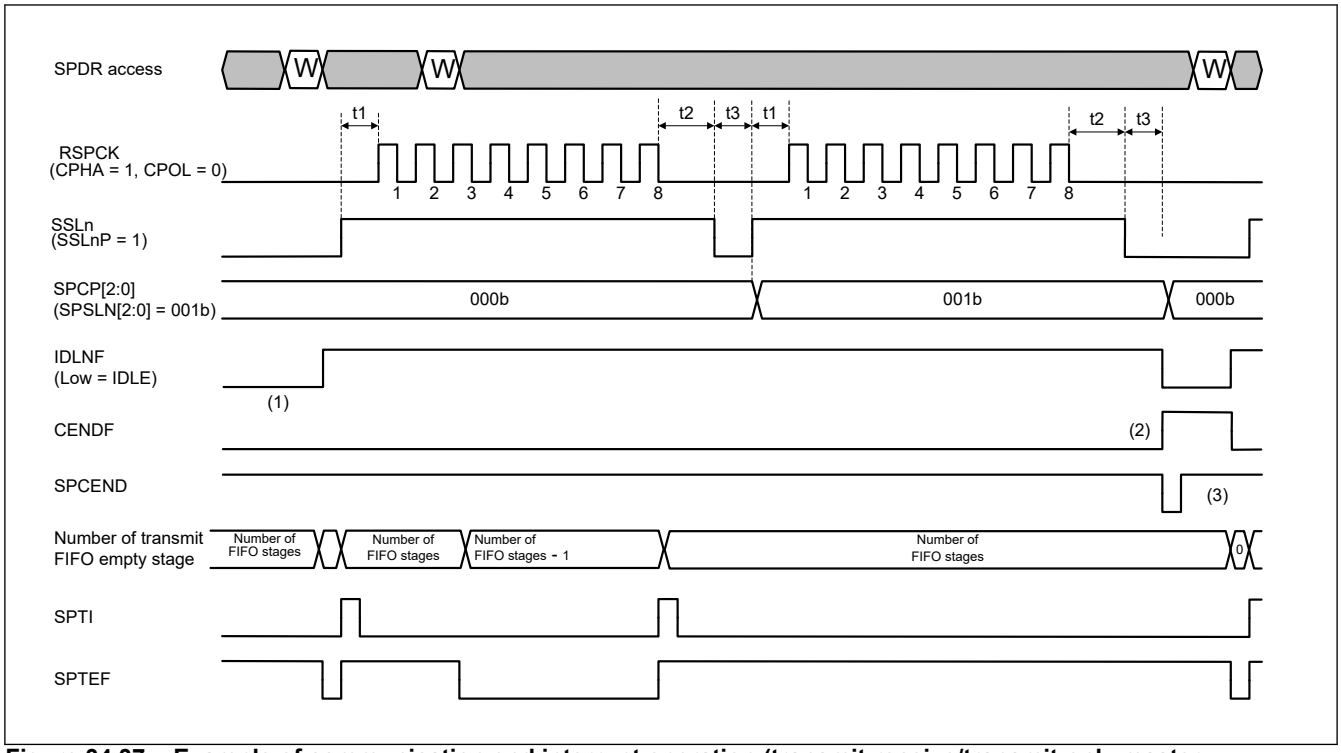


Figure 34.37 Example of communication end interrupt operation (transmit-receive/transmit-only master mode / Motorola-SPI)

(1) The CENDF flag is 0 and the level of SPCEND is 1 before communication starts, and these are kept during communication.

(2) The CENDF flag is 1 (communication end) at the end of t3 cycle, because the next command is 000b and there is no next transmit data. The SPCEND interrupt is then output with 1 PCLKM cycle width if the CENDIE bit is 1.

(3) The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX), or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

(2) In the TI-SSP case

Figure 34.38 shows an example of communication end interrupt operation during transmit-receive/transmit-only master mode.

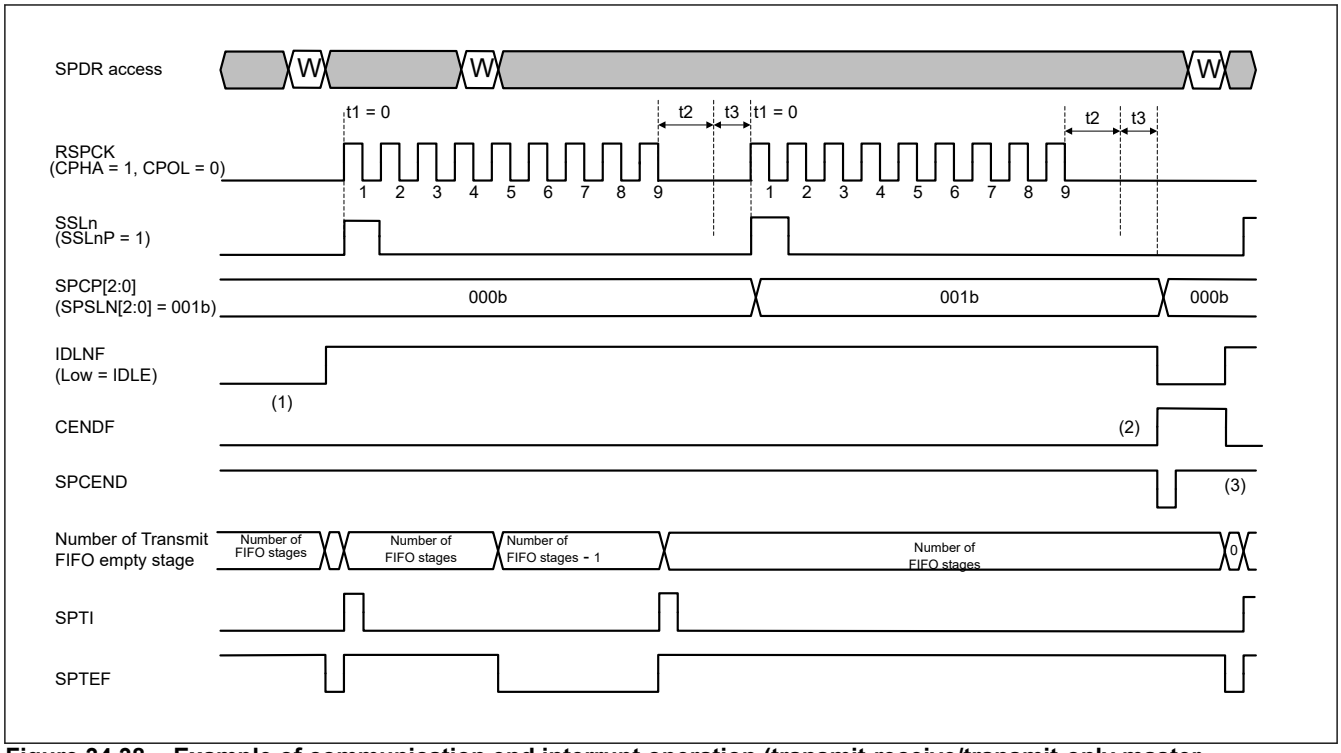


Figure 34.38 Example of communication end interrupt operation (transmit-receive/transmit-only master mode / TI-SSP)

(1) The CENDF flag is 0 and the level of SPCEND is 1 before communication starts, and these are kept during communication.

(2) The CENDF flag is 1 (communication end) at the end of t3 cycle, because the next command is 000b and there is no next transmit data. The SPCEND interrupt is then output with 1 PCLKM cycle width if the CENDIE bit is 1.

(3) The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX), or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

34.3.9.2 Receive-only in Master Mode

Refer to the description of the CENDF bit in section 34.2.18. SPSR : SPI Status Register for the setting/clearing conditions of the communication completion flag during Receive-only in Master Mode.

Figure 34.39 shows an example of communication end interrupt operation during receive-only master mode at RMFM[4:0] = 0x00.

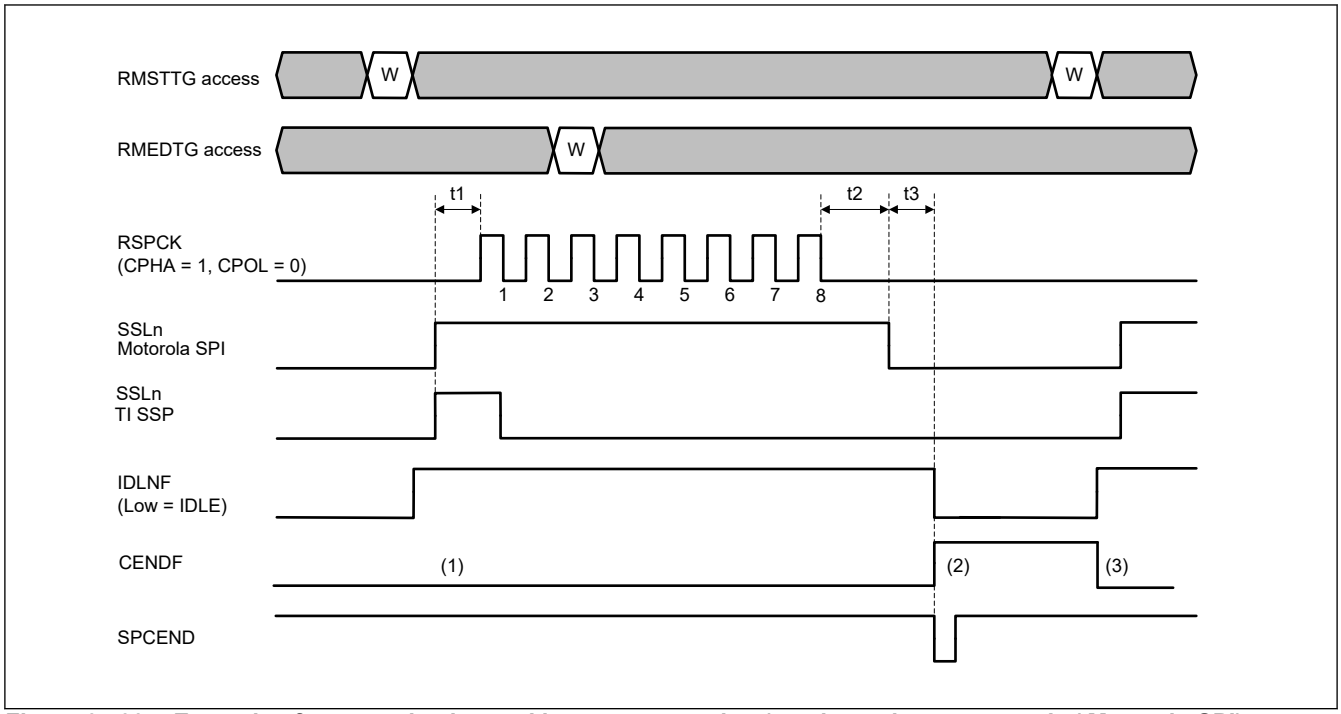


Figure 34.39 Example of communication end interrupt operation (receive-only master mode / Motorola-SPI) at RMFM[4:0] = 0x00

(1) The CENDF flag is 0 and the level of SPCEND is 1 before communication starts, and these are kept during communication.

(2) The CENDF flag is 1 (communication end) at the end of t3 cycle, by writing 1 to RMEDTG during the communication frame. The SPCEND interrupt is then output with 1 PCLKM cycle width if SPCR.CENDIE = 1.

(3) The CENDF flag is cleared when writing 1 to RMSTTG, or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

Figure 34.40 shows an example of communication end interrupt operation during receive-only master mode at RMFM[4:0] ≠ 0x00.

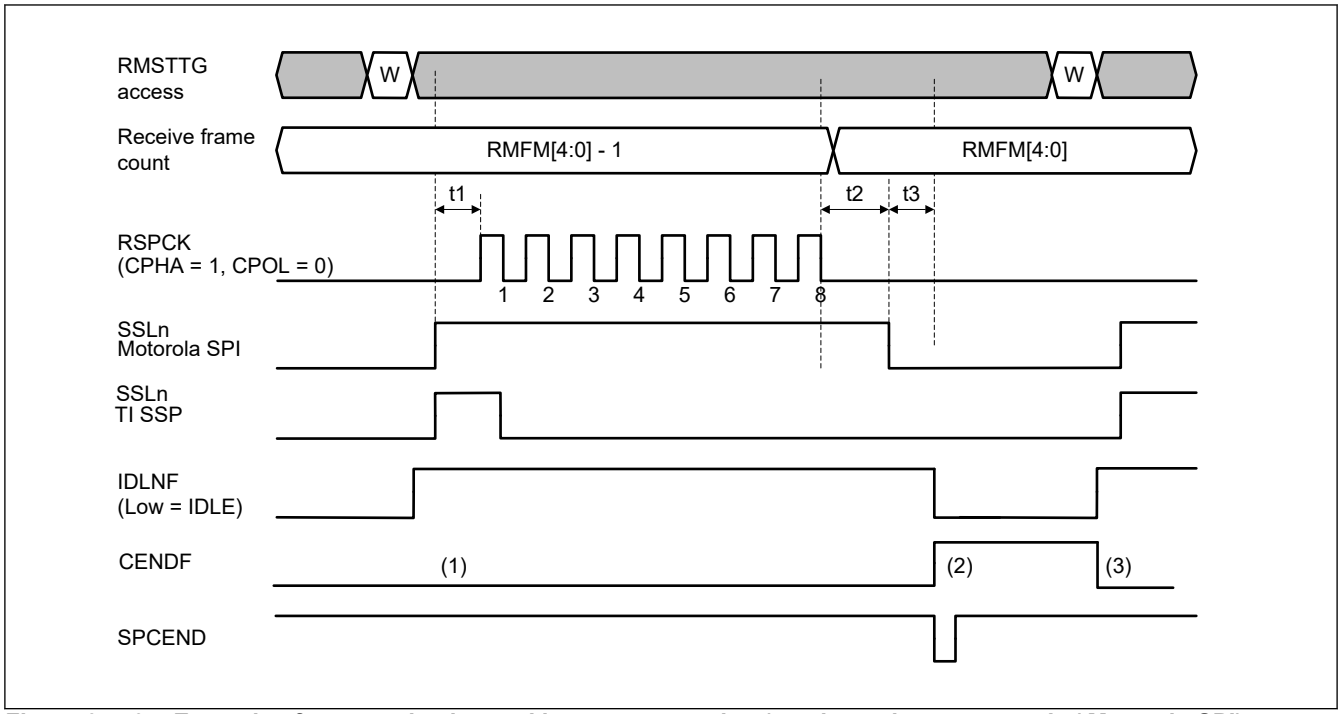


Figure 34.40 Example of communication end interrupt operation (receive-only master mode / Motorola-SPI) at RMFM[4:0] ≠ 0x00

(1) The CENDF flag is 0 and the level of SPCEND is 1 before communication starts, and these are kept during communication.

(2) The CENDF flag is 1 (communication end) at the end of t3 cycle, after receiving the number of frames set by RMFM[4:0]. The SPCEND interrupt is then output with 1 PCLKM cycle width if SPCR.CENDIE = 1.

(3) The CENDF flag is cleared when writing 1 to RMSTTG, or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

In slave mode operation, the output timing of the communication end interrupt is dependent on the value of the SPCR.SPMS bit (SPI mode select bit). The clear timing of the communication end interrupt is dependent on the communication mode (transmit-receive or transmit-only or receive-only).

34.3.9.3 Transmit-Receive/Transmit-only in Slave Mode on SPI Operation (4-wire)

See the description for the CENDF bit in section 34.2.18. SPSR : SPI Status Register for the setting/clearing conditions of the communication completion flag during Transmit-Receive/Transmit-only in Slave Mode (4-wire).

(1) In the Motorola-SPI case

Figure 34.41 shows an example of communication end interrupt operation during transmit-receive/transmit-only slave mode on SPI operation.

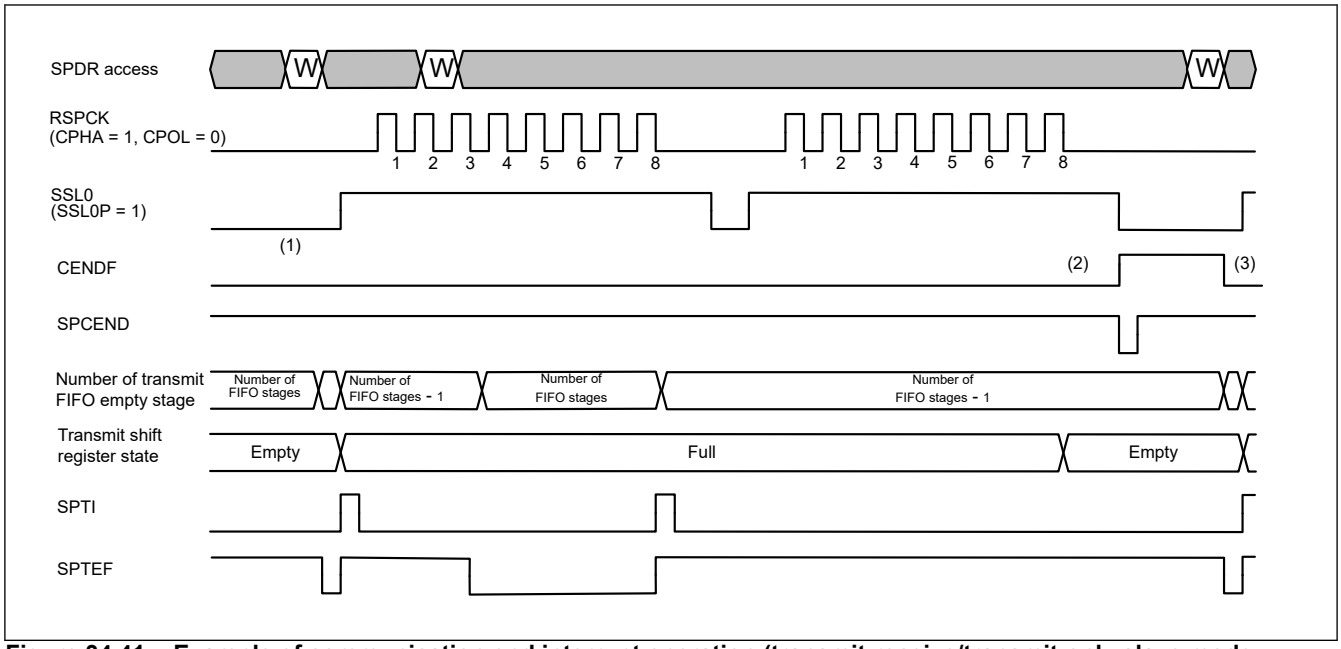


Figure 34.41 Example of communication end interrupt operation (transmit-receive/transmit-only slave mode on SPI operation / Motorola-SPI)

(1) The CENDF flag is 0 and the level of SPCEND is 1 before communication starts, and these are kept during communication.

(2) The CENDF flag is 1 (communication end) at the timing of SSL0 negate, when the next transfer data is not set in the transmit FIFO and the transmit shift register is empty. The SPCEND interrupt is then output with 1 PCLKM cycle width if SPCR.CENDIE = 1.

(3) The CENDF flag is cleared when the next transmition data is written to the transmit buffer (SPTX), or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

(2) In the TI-SSP case

Figure 34.42 shows an example of communication end interrupt operation during transmit-receive/transmit-only slave mode on SPI operation.

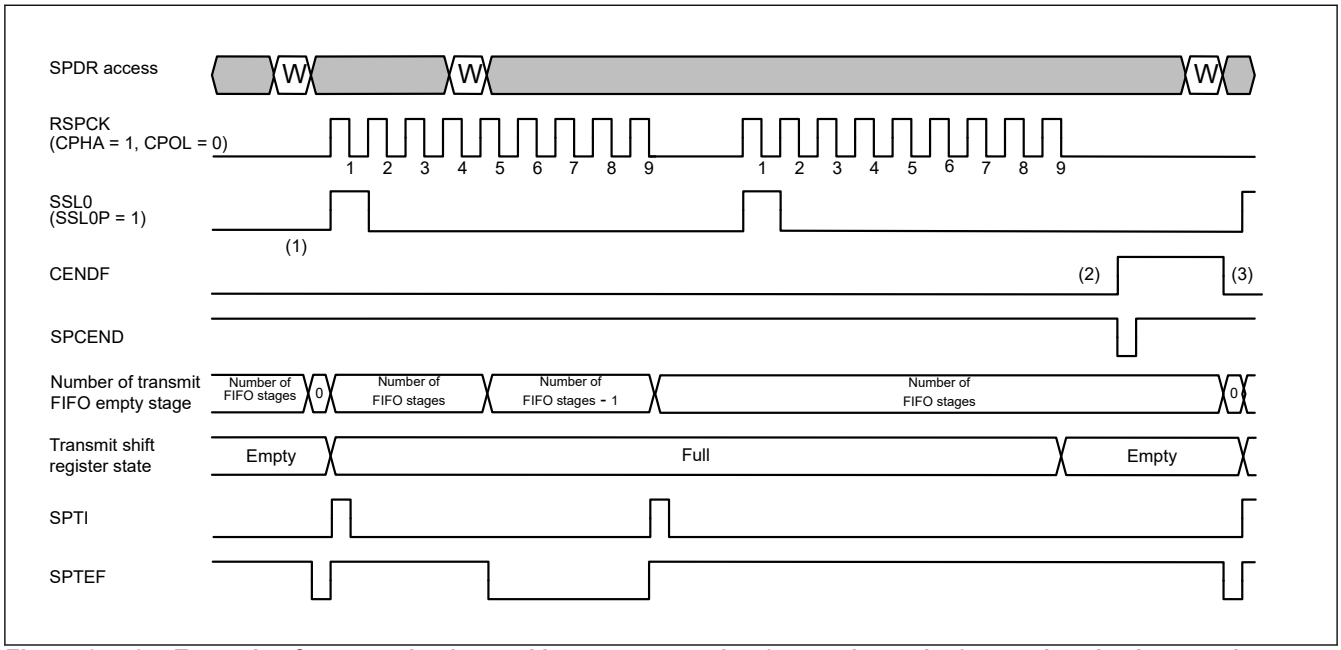


Figure 34.42 Example of communication end interrupt operation (transmit-receive/transmit-only slave mode on SPI operation / TI-SSP)

(1) The CENDF flag is 0 and the level of SPCEND is 1 before communication start, and these have kept during communication.

(2) The CENDF flag is 1 (communication end) at the RSPCK last data bit sampling, when the next transfer data is not set in the transmit FIFO and the transmit shift register is empty. The SPCEND interrupt is then outputs with PCLKM 1 cycle width if the CENDIE bit is 1.

(3) The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX), or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

34.3.9.4 Receive Only in Slave Mode on SPI Operation (4-wire)

Refer to the description of the CENDF bit in section 34.2.18. SPSR : SPI Status Register for the setting/clearing conditions of the communication completion flag during Receive-only in Slave Mode (4-wire).

(1) In the Motorola-SPI case

Figure 34.43 shows an example of communication end interrupt operation during receive only slave mode on SPI operation (4-wire).

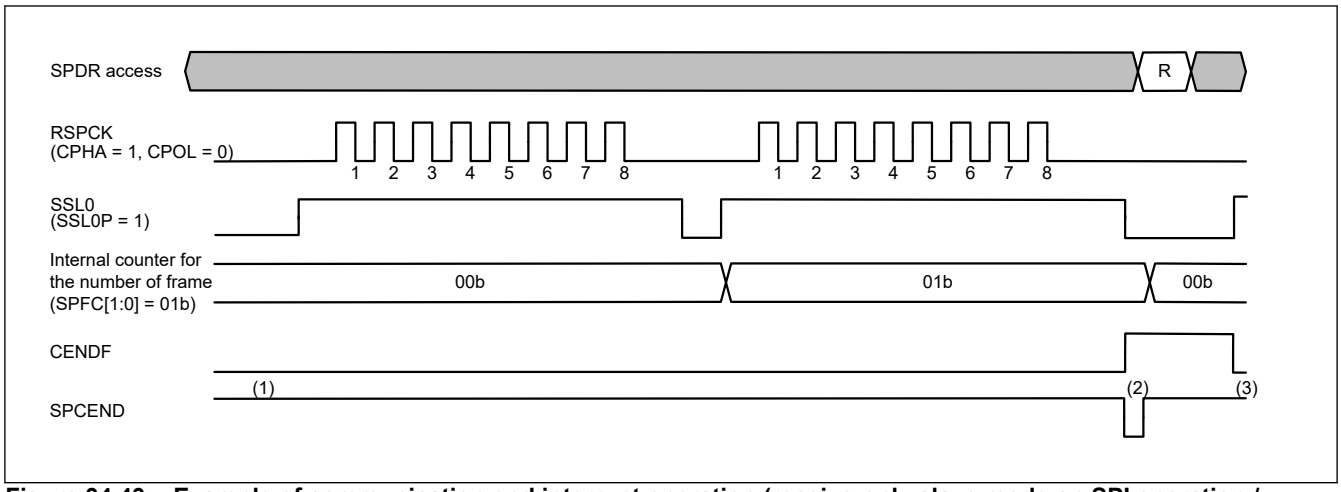


Figure 34.43 Example of communication end interrupt operation (receive-only slave mode on SPI operation / Motorola-SPI)

(1) The CENDF flag is 0 and the level of SPCEND is 1 before communication start, and these have kept during communication.

(2) After store the frames for the SPDCR.SPFC register set value in the receive buffer, the CENDF flag becomes 1 (communication completed) at the timing of SSL0 negation. The SPCEND interrupt is then outputs with PCLKM 1 cycle width if SPCR.CENDIE = 1.

(3) The CENDF flag is cleared at the SSL0 assert when the next transmission start, or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

(2) In the TI-SSP case

Figure 34.44 shows an example of communication end interrupt operation during receive only slave mode on SPI operation (4-wire).

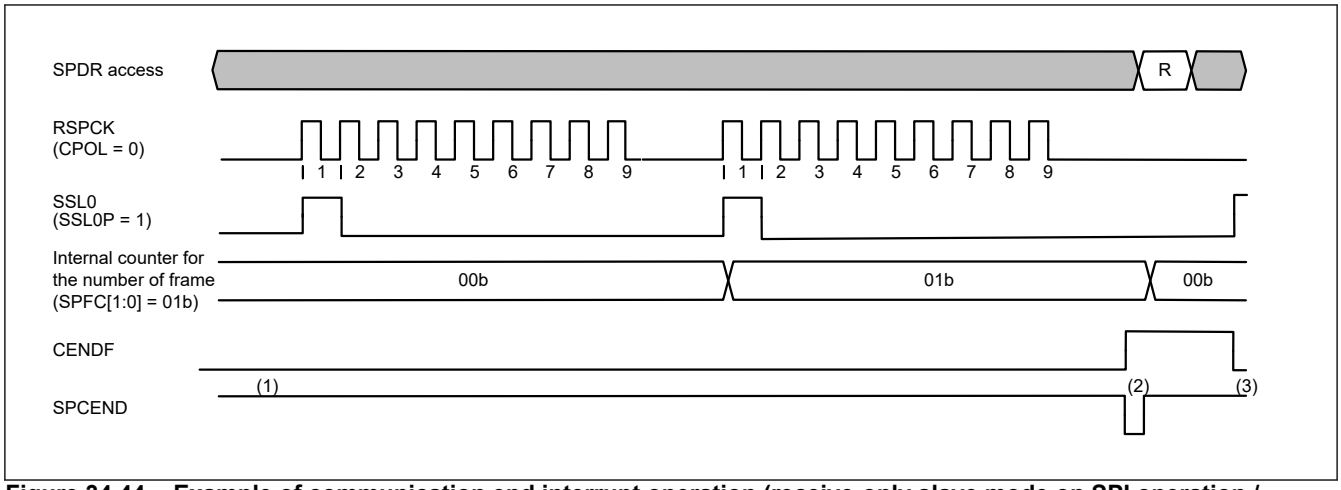


Figure 34.44 Example of communication end interrupt operation (receive-only slave mode on SPI operation / TI-SSP)

(1) The CENDF flag is 0 and the level of SPCEND is 1 before communication start, and these have kept during communication.

(2) The CENDF flag is 1 (communication end) at the RSPCK last data bit sampling, when the last frame transmission ends. The SPCEND interrupt is then outputs with PCLKM 1 cycle width if SPCR.CENDIE = 1.

(3) The CENDF flag is cleared at the SSL0 assert when the next transmission start, or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

34.3.9.5 Transmit-Receive/Transmit-only in Slave Mode on Clock Synchronous Operation (3-wire)

Refer to the description of the CENDF bit in section 34.2.18. SPSR : SPI Status Register for the setting/clearing conditions of the communication completion flag during Receive-only in Slave Mode (4-wire).

Figure 34.45 shows an example of communication end interrupt operation during transmit-receive/transmit slave mode on clock synchronous operation (3-wire).

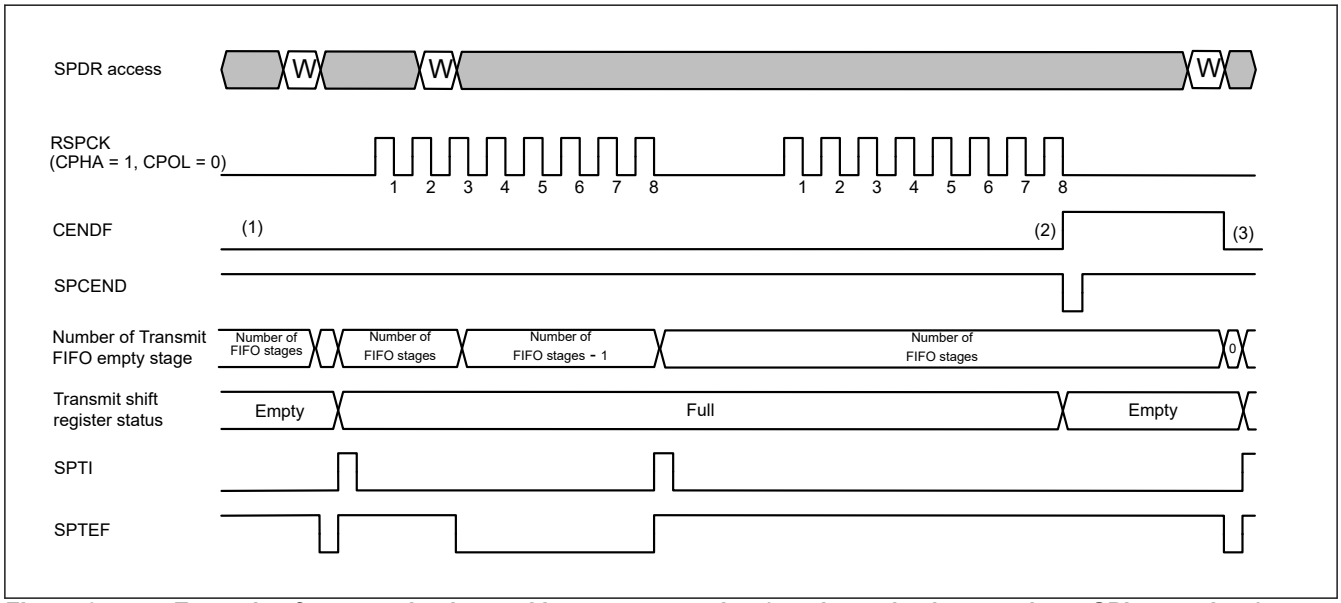


Figure 34.45 Example of communication end interrupt operation (receive-only slave mode on SPI operation / Motorola-SPI)

(1) The CENDF flag is 0 and the level of SPCEND is 1 before communication start, and these have kept during communication.

(2) When the next transfer data is not set in the transmit FIFO and the transmit shift register is empty, the SPCEND interrupt is then outputs with PCLKM 1 cycle width if SPCR.CENDIE = 1.

(3) The CENDF flag is cleared when the next transmission data is written to the transmit buffer (SPTX), or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

34.3.9.6 Receive only in Slave Mode on Clock Synchronous Operation (3-wire)

Refer to the description of the CENDF bit in section 34.2.19. SPTFSR : SPI Transfer FIFO Status Register and section 34.2.18. SPSR : SPI Status Register for the setting/clearing conditions of the communication completion flag during Receive -only in Slave Mode on Clock Synchronous (3-wire).

Figure 34.46 shows an example of communication end interrupt operation during receive only slave mode on clock synchronous operation.

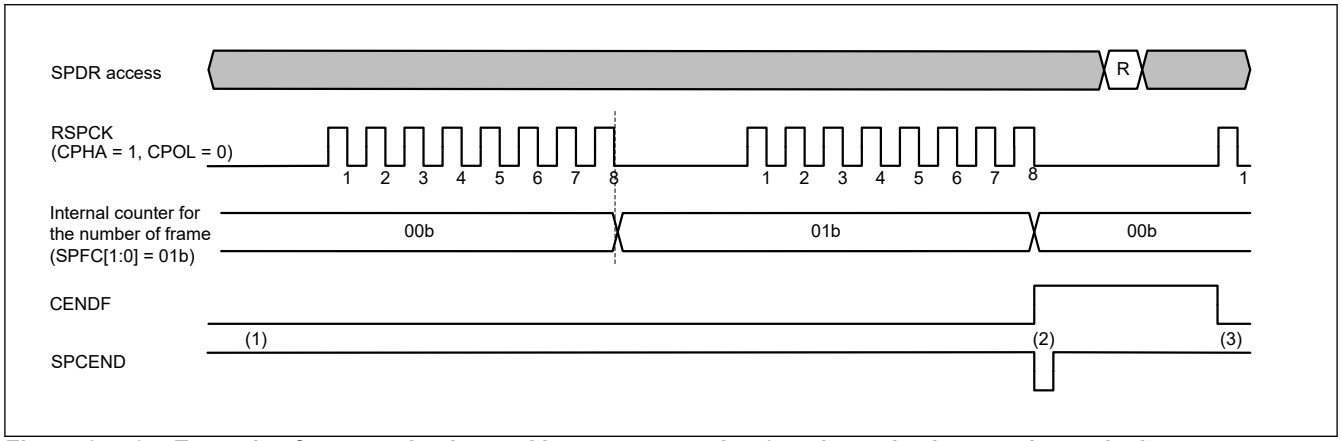


Figure 34.46 Example of communication end interrupt operation (receive-only slave mode on clock synchronous operation)

(1) The CENDF flag is 0 and the level of SPCEND is 1 before communication start, and these have kept during communication.

(2) The CENDF flag is set to 1 (communication completed) at the timing of the last data bit sampling of RSPCK in the last frame communication when the last frame of the SPDCR.SPFC bit set value is received. The SPCEND interrupt is then outputs with PCLKM 1 cycle width if SPCR.CENDIE = 1.

(3) The CENDF flag is cleared at the first edge of RSPCK for the next transmission, or when 1 is written to the SPSRC.CENDFC bit, then the CENDF flag is 0.

34.3.9.7 Common Operation

In this Section, the operation common to each mode communication in section 34.3.9.1. Transmit-Receive/Transmit-only in Master Mode to section 34.3.9.6. Receive only in Slave Mode on Clock Synchronous Operation (3-wire) is explained. When the enable of SPI communication end interrupt (CENDIE) is 0, at the time of communication completion, a flag of communication end (CENDF) is set, but no interrupt is output. However, if the enable of communication end interrupt (CENDIE) is set to 1 before clearing the flag of communication end (CENDF) while the enable of SPI function (SPE) is 1, the communication end interrupt is output.

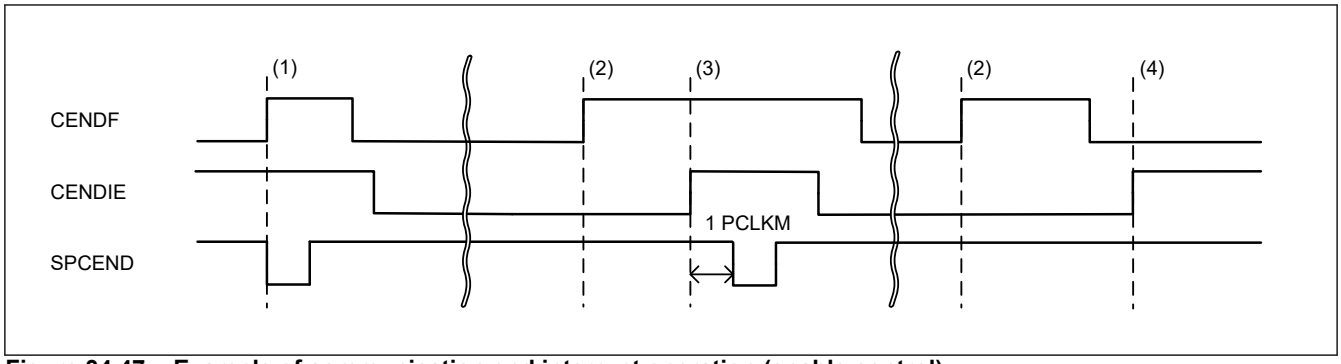


Figure 34.47 Example of communication end interrupt operation (enable control)

(1) When the enable of SPI communication end interrupt (CENDIE) is 1, at the time of communication completion, the following two are the same timing

a flag of communication end (CENDF)

the communication end interrupt

(2) When the enable of SPI communication end interrupt (CENDIE) is 0, at the time of communication completion, the following occurs, but no interrupt.

a flag of communication end (CENDF)

(3) After (2), if the enable of communication end interrupt (CENDIE) is set when the enable of SPI function (SPE) and the flag of communication end (CENDF) are 1, the communication end interrupt is output after 1 PCLKM.

(4) After (2), even if the enable of communication end interrupt (CENDIE) is set when the enable of SPI function (SPE) or the flag of communication end (CENDF) is 0, the communication end interrupt is not output.

34.3.10 Error Detection

In normal serial transfer of SPI, data written to the transmit buffer in the SPDR register is transmitted in serial and received serial data can be read from the SPDR's receive buffer. Depending on the transmit buffer or receive buffer status when SPDR is accessed and the SPI status at the start or end of serial transfer, abnormal transfer is performed.

When some abnormal transfer operation takes place, the SPI detects it as an underrun error, overrun error, parity error, or mode fault error. Table 34.10 shows the relationship between abnormal transfer operation and the SPI's error detection function.

Table 34.10 Abnormal transfer occurrence conditions and error detection function of SPI (1 of 2)

	Transfer occurrence condition	SPI operation	Error detection
1	SPDR is written when while no empty stages in the transmit FIFO.	• Transmit buffer data is retained. • No write data is present.	None
2	SPDR is read while no data stored in receive FIFO.	Received data and previously received serial data are read.	None
3	Serial transfer starts in transmit slave mode or transmit-only slave mode, before transmit data output is ready.	• Serial transfer is suspended. • No transmit data or receive data is present. • Driving of the MISO output signal is stopped. • SPI function is disabled.	Underrun error
4	Serial transfer ends when data is stored in the receive FIFO for the number of FIFO stages.	• Receive FIFO data is retained. • No serial receive data is present.	Overrun error
5	Incorrect parity bit has been received with the parity function enabled in following mode. • Transmit / receive-only master mode • Transmit / receive slave mode • Receive-only slave mode	The parity error flag is asserted.	Parity error
6	The SSL0 input signal is asserted when the serial transfer is idle state in multi-master mode.	• Driving of the RSPCK, MOSI, SSL1 to SSL3 output signals is stopped. • SPI function is disabled.	Mode fault error

Table 34.10 Abnormal transfer occurrence conditions and error detection function of SPI (2 of 2)

	Transfer occurrence condition	SPI operation	Error detection
7	The SSL0 input signal is asserted during serial transfer in multi-master mode.	• Serial transfer is suspended. • Missing transmit/receive data. • Driving of the RSPCK, MOSI, SSL1 to SSL3 output signals is stopped. • SPI function is disabled.	Mode fault error
8	[In the Motorola-SPI case] The SSL0 input signal is negated during serial transfer in slave mode.	• Serial transfer is suspended. • No transmit data or receive data is present. • Driving of the MISO output signal is stopped. • SPI function is disabled.	Mode fault error
9	[In the TI-SSP case] The SSL0 input signal is asserted during serial transfer in slave mode.	• Serial transfer is suspended. • No transmit data or receive data is present. • Driving of the MISO output signal is stopped. • SPI function is disabled.	Mode fault error

The SPI does not detect any error for the operation shown in No.1 in Table 34.10. To prevent missing of data during SPDR writing, write data to SPDR upon a transmit buffer empty interrupt request or while the SPSR.SPTEF flag is 1.

The SPI does not detect any error also for the operation shown in No.2 in Table 34.10. To prevent reading of unnecessary data, read data from SPDR upon a receive buffer full interrupt request or while the SPSR.SPRF flag is 1.

The underrun error shown in No.3 in Table 34.10 is described in section 34.3.10.4. Underrun Error.

The overrun error shown in No.4 in Table 34.10 is described in section 34.3.10.1. Overrun Error.

The parity error shown in No.5 in Table 34.10 is described in section 34.3.10.2. Parity Error.

The mode fault error shown in No.6 to No.9 in Table 34.10 is described in section 34.3.10.3. Mode Fault Error.

For transmission and reception interrupts, see section 34.3.7. Transmission Buffer Empty/Receive Buffer Full Interrupt.

34.3.10.1 Overrun Error

When serial transfer ends while the receive buffer in the SPDR register is full, the SPI detects an overrun error and sets the OVRF flag in SPSR to 1. While the OVRF flag is 1, the SPI does not copy shift register data to the receive buffer. Therefore, data before an error occurs is retained in the receive buffer. To clear the OVRF flag in SPSR to 0, issue a system reset or 1 is written to the SPSRC.OVRFC bit.

Figure 34.48 shows operation of the SPRF flag and the OVRF flag in SPSR. "SPSR access" and "SPDR access" described in Figure 34.48 show an access to SPSR and SPDR respectively. "W" shows a write cycle and "R" shows a read cycle. In the example in Figure 34.48, the SPI performs 8-bit data serial transfer with the settings of SPCMD.CPHA = 1 and SPCMD.CPOL = 0. Numbers under the RSPCK waveform show the number of RSPCK cycles (number of transfer bits).

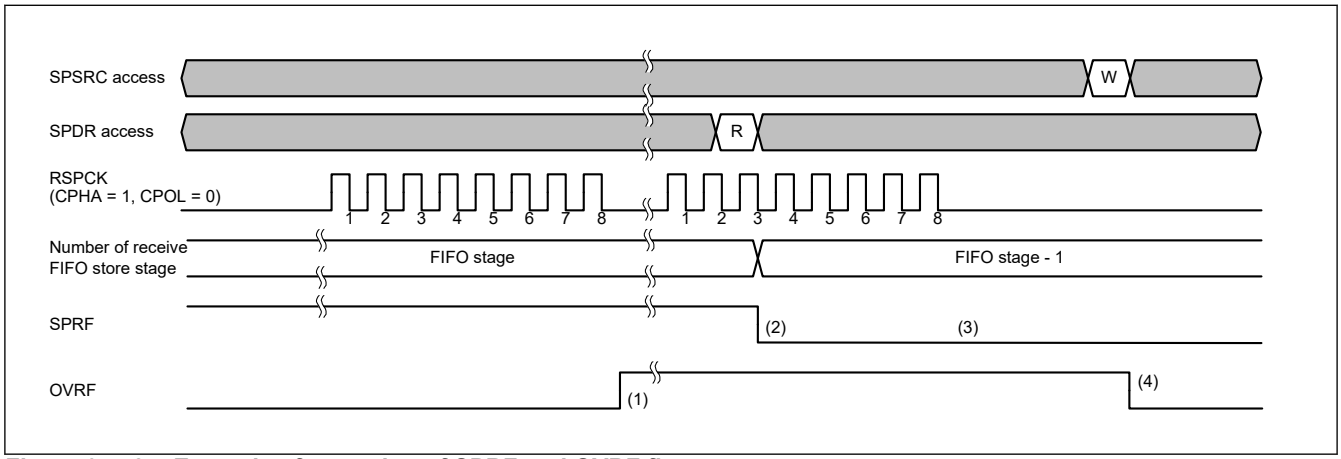


Figure 34.48 Example of operation of SPRF and OVRF flags

The following describes operation of flags at timings (1) to (4) in the figure above.

(1) When serial transfer ends while data is stored for the number of FIFO stages, the SPI detects an overrun error and sets the OVRF flag to 1. The SPI does not copy shift register data to the receive buffer. The SPI does not detect a parity

error even when SPPE = 1. In master mode, the SPI copies the value of pointer to the SPCMD register to the SPSSR.SPECM[2:0] bits.

(2) The SPI can read receive buffer data by reading SPDR. At this time, the SPRF flag is cleared to 0 at the last access when the received data is read from SPDR in one processing routine using DMAC.

(3) When serial transfer ends while the OVRF flag is 1 (overrun error), the SPI does not copy shift register data to the receive buffer (SPRF flag remains 0) and no SPI receive buffer full interrupt is generated. The SPI does not detect a parity error even when SPPE = 1. In master mode, the SPI does not update the SPSSR.SPECM[2:0] bits. When the SPI does not copy received data from the shift register to receive buffer while an overrun error is present, the SPI also determines that the shift register is empty at the end of serial transfer, enabling data transfer from the transmit buffer to the shift register.

(4) When 1 is written to the SPSRC.OVRFC bit.

Whether an overrun error is present can be checked by reading SPSR or by reading an SPI error interrupt and SPSR. To perform serial transfer without using an SPI error interrupt, read SPDR and then immediately read SPSR to quickly detect an overrun error. When the SPI is used in master mode, the pointer value to SPCMD when an error is present can be checked by reading the SPSSR.SPECM bits. When an overrun error occurs and the OVRF flag is set to 1, successful reception is disabled until the OVRF flag is cleared to 0.

When the RSPCK auto-stop function is enabled in master mode, an overrun error does not occur. Figure 34.49 and Figure 34.50 show the clock stop waveform when a serial transfer continues while the reception buffer is full in master mode.

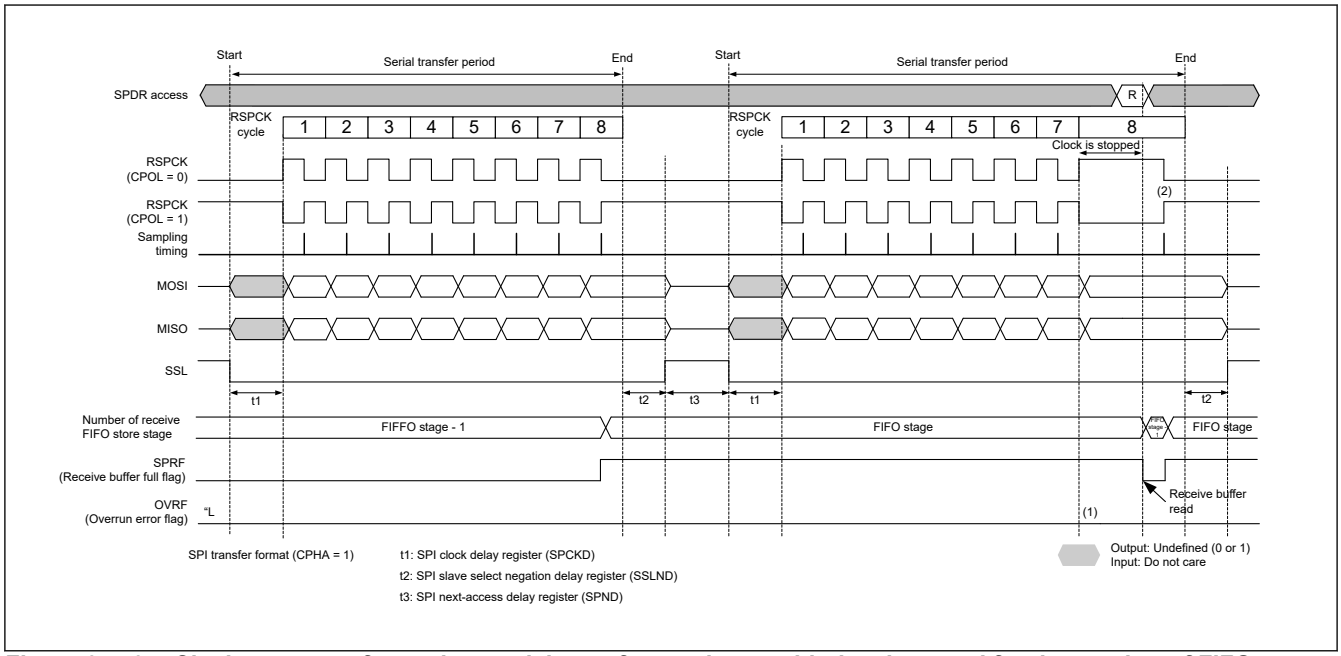


Figure 34.49 Clock stop waveform when serial transfer continues with data is stored for the number of FIFO stages in master mode (CPHA = 1)

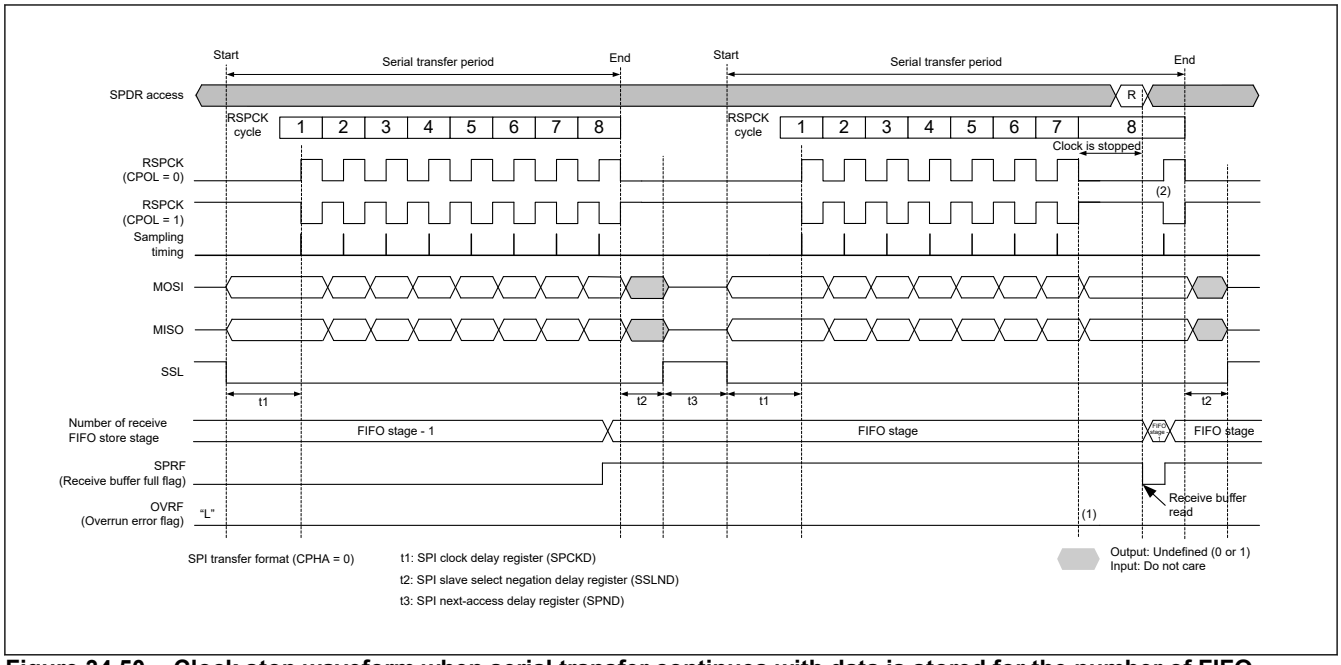


Figure 34.50 Clock stop waveform when serial transfer continues with data is stored for the number of FIFO stages in master mode (CPHA = 0)

The following describes operation of flags at timings (1) and (2) in the figure above.

(1) While data is stored in the receive FIFO for the number of FIFO stages, the RSPCK clock is deactivated and no overrun error occurs.

(2) Receive buffer data can be read by reading SPDR during clock stop. After the receive buffer data has been read, the RSPCK clock restarts.

Overrun error does not occur when RSPCK automatic stop function is enabled for transfer with no delay of between frames during burst transfer in master mode. The Figure 34.51 and the Figure 34.52 show the clock stop waveform, when there is no delay between frames at burst transfer and the serial transfer continues in the data is stored in the receive FIFO for the number of FIFO stages.

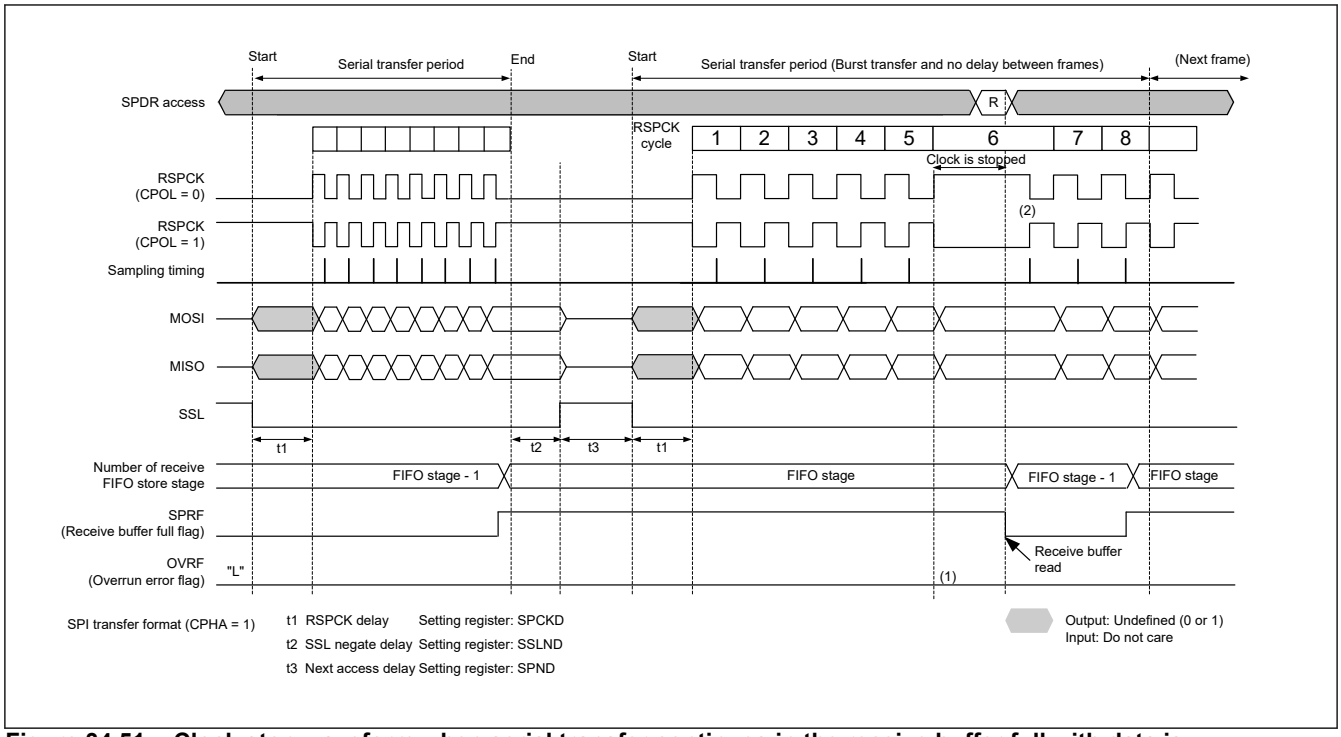


Figure 34.51 Clock stop waveform when serial transfer continues in the receive buffer full with data is stored for the number of FIFO stages in master mode (at burst transfer and no delay between frames CPHA = 1)

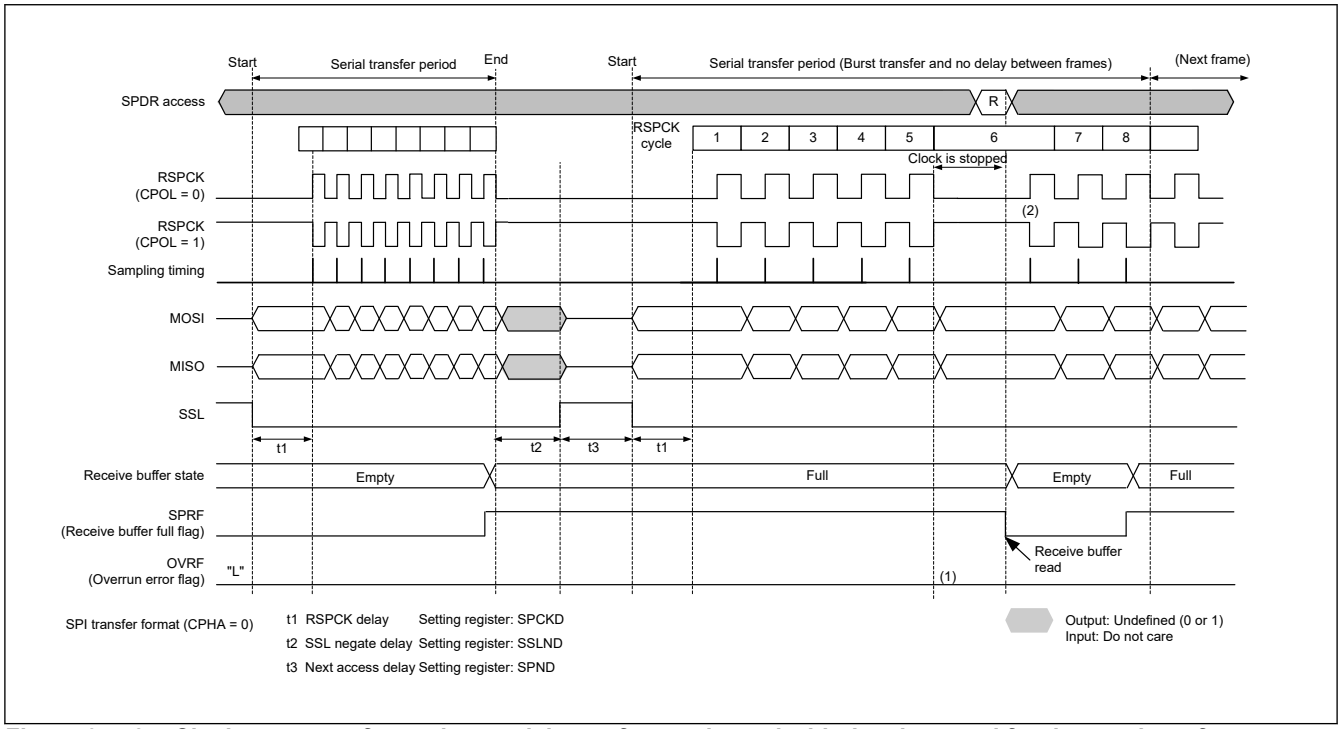


Figure 34.52 Clock stop waveform when serial transfer continues i with data is stored for the number of FIFO stages in master mode (at burst transfer and no delay between frames CPHA = 0)

The following describes operation of flags at timings (1) and (2) in the figure above.

(1) While the data is stored for the number of FIFO stages, the RSPCK clock is deactivated and no overrun error occurs.

(2) Receive buffer data can be read by reading SPDR during clock stop. After the receive buffer data has been read, the RSPCK clock restarts.

34.3.10.2 Parity Error

After transfer in transmit-receive or receive-only master mode, transmit-receive slave mode or receive only slave mode while the SPCR.SPPE = 1, the SPI checks occurrence of a parity error. When the SPI detects a parity error in received data, the PERF flag in the SPSR register is set to 1. While the OVRF flag is 1, the SPI does not copy shift register data to the receive buffer. Therefore, parity error in received data is not detected. To clear the PERF flag in SPSR to 0, issue a system reset or write 1 to the SPSRC.PERFC bit.

Figure 34.53 shows operation of the OVRF and PERF flags in SPSR. "SPSR access" described in Figure 34.53 shows an access to SPSR. "W" shows a write cycle and "R" shows a read cycle. In the example in Figure 34.53, the SPI performs transmit-receive serial communication while SPCR.SPPE = 1. The SPI performs 8-bit data serial transfer with the settings of SPCMD.CPHA = 1 and SPCMD.CPOL = 0. Numbers under the RSPCK waveform show the number of RSPCK cycles (number of transfer bits).

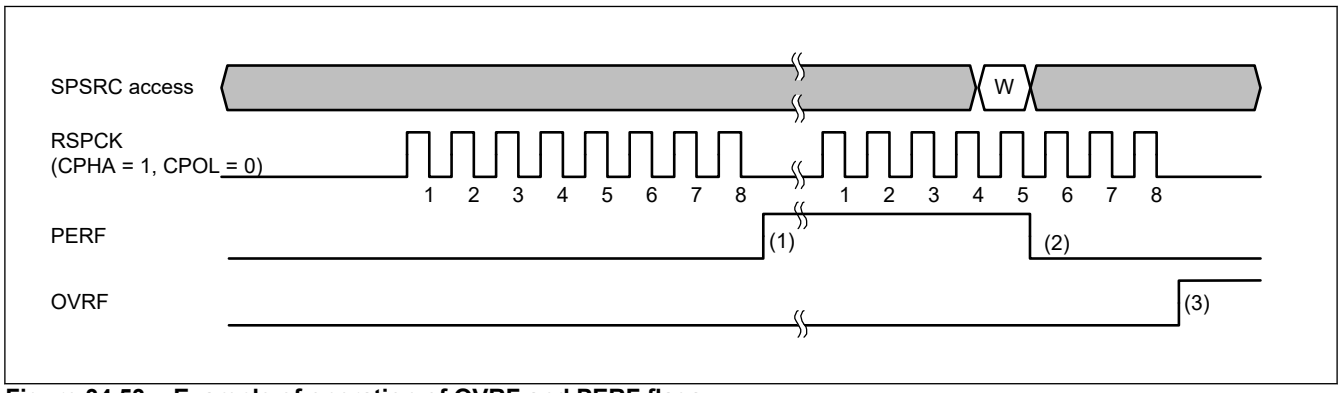


Figure 34.53 Example of operation of OVRF and PERF flags

The following describes operation of flags at timings (1) to (3) in the figure above.

(1) When the SPI does not detect an overrun error and terminates the serial transfer, the SPI copies shift register data to the receive buffer. When the SPI checks the received data and detects a parity error at this time, the PERF flag is set to 1. In master mode, the SPI copies the value of pointer to the SPCMD register to the SPSSR.SPCM [2:0] bits.

(2) When 1 is written to the SPSRC.PERFC bit.

(3) When the SPI detects an overrun error and terminates the serial transfer, the SPI does not copy shift register data to the receive buffer and does not detect a parity error.

Whether a parity error is present can be checked by reading SPSR or by reading an SPI error interrupt and SPSR. To use an SPI error interrupt, set the SPCR.SPEIE bit to 1. To perform serial transfer without using an SPI error interrupt, read SPDR to quickly detect a parity error. When the SPI is used in master mode, the pointer value to SPCMD when an error is present can be checked by reading the SPSSR.SPECM[2:0] bits.

34.3.10.3 Mode Fault Error

When the SPCR.MSTR = 1, SPCR.SPMS = 0, and SPCR.MODFEN = 1, the SPI operates in multi-master mode. When an active level is input to the SSL0 input signal of the SPI in multi-master mode, the SPI detects a mode fault error regardless of serial transfer status and sets the MODF flag in the SPSR register to 1. When a mode fault error is detected, the SPI copies the value of pointer to the SPCMD register to the SPSSR.SPECM[2:0] bits. The SSL0 signal active level depends on the SSL0P bit in the SSLP register.

While SPCR.MSTR = 0, the SPI operates in slave mode. When SPCR.SPMS = 0 and SPCR.MODFEN = 1 in slave mode, if the SSL0 input signal is negated during the serial transfer period (from valid data drive start to final valid data latch), the SPI detects a mode fault error while any of the following 2 conditions is met.

(1) In the Motorola-SPI case

When the SSL0 input signal is negated while serial data transfer.

(2) In the TI-SSP case

When the SSL0 input signal is asserted while serial data transfer. However, during burst transfer, no error is detected even if the SSL0 input signal is asserted during the last bit of frame.

When the SPI detects a mode fault error, it stops driving output signals and clears the SPCR.SPE bit.

When the SPE bit is cleared, the SPI function is disabled (as described in section 34.3.12. SPI Initialization). In a multi-master configuration, the mastership can be released by stopping driving output signals and disabling the SPI function by using a mode fault error.

Whether a mode fault error is present can be checked by reading SPSR or by reading an SPI error interrupt and SPSR. To detect a mode fault error without using an SPI error interrupt, poll SPSR. When the SPI is used in master mode, the pointer value to SPCMD when an error is present can be checked by reading the SPSSR.SPECM[2:0] bits.

While the MODF flag = 1, the SPI ignores writing 1 to the SPE bit. To enable the SPI function after a mode fault error is detected, clear the MODF flag to 0 without fail.

34.3.10.4 Underrun Error

While the SPI is operating in slave mode (SPCR.MSTR = 0) and the communication mode select SPCR.TXMD[1:0] bits are set to 00b or 01b, if serial transfer is started before transmit data output is ready with the SPCR.SPE bit set to 1 (SPI function enabled), the SPI detects an underrun error and sets the MODF and UDRF flags in the SPSR register to 1.

When the SPI detects an underrun error, it stops driving output signals and clears the SPE bit to 0. When the SPE bit is cleared, the SPI function is disabled (as described in section 34.3.12. SPI Initialization)

Whether an underrun error is present can be checked by reading SPSR or by reading an SPI error interrupt and SPSR.

To detect an underrun error without using an SPI error interrupt, poll SPSR.

While the MODF flag = 1, the SPI ignores writing 1 to the SPE bit. To enable the SPI function after an underrun error is detected, clear the MODF flag to 0 without fail.

34.3.11 Received Data Ready Detection

When SPCR.TXMD[1:0] = 00b, 01b, or 11b, and SPDRCR.SPDRC[7:0] ≠ 0x00, after receiving data in the receive FIFO during communication (SPE = 1), SPDR.SPDRF flag is set to 1 when the received data is not stored even after the number of received FIFOs ≤ the threshold value and the value set in SPDRCR.SPDRC[7:0] has elapsed.

When the receive data ready is detected, the interrupt can be selected as SPRI or SPEI with the SPCR.SPDRES bit.

Figure 34.54 shows an example of reception data ready detection operation.

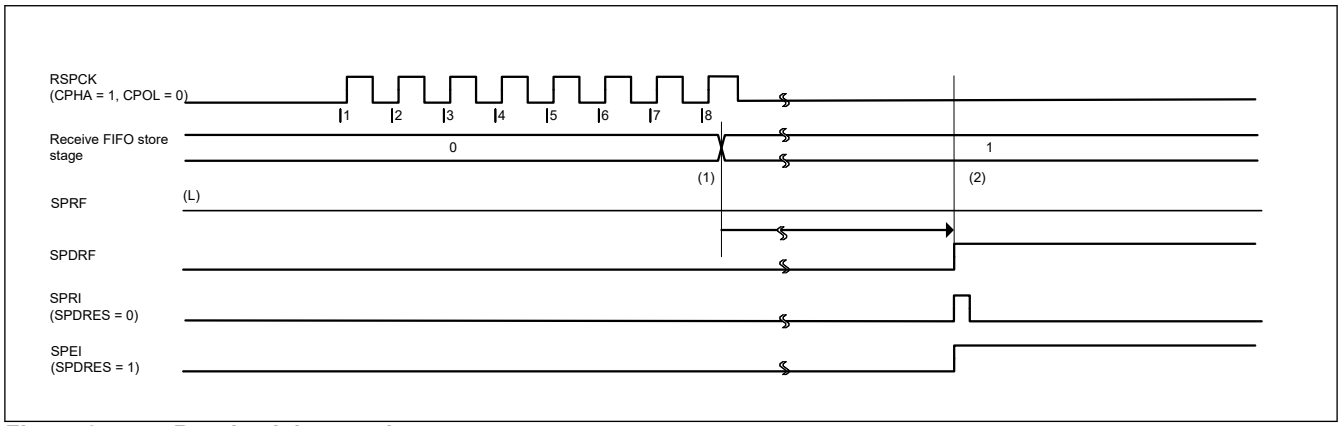


Figure 34.54 Received data ready

The following describes operation at the timings indicated by (1) and (2) in the figure.

(1) Store the received data in the receive FIFO. SPRF is 0, because receive FIFO store stage ≤ RTRG.

(2) Set SPDRF and assert SPRI or SPEI because there is no writing to the receive FIFO for the amount of SPDRC[7:0] set from (1).

34.3.12 SPI Initialization

When 0 is written to the SPCR.SPE bit or when the SPI clears the SPE bit to 0 after it detects a mode fault error or an underrun error, the SPI disables the SPI function and initializes a part of module functions. Furthermore, the SPI initializes

all module functions when a system reset occurs. The following describes initialization by clearing the SPCR.SPE bit and by using a system reset.

34.3.12.1 Initialization by Clearing the SPE Bit

When the SPCR.SPE bit is cleared to 0, the SPI performs the following initialization:

- Suspends the ongoing serial transfer.
- Stops driving output signals (Hi-Z) in slave mode.
- Resets the internal SPI state.
- SPSR.SPTEF flag = 1.

In initialization by clearing the SPE bit, control bits in SPI are not initialized. For this reason, re-setting the SPE bit to 1 activates the SPI with the pre-initialization transfer mode.

Values of the CENDF flag, the SPRF flag, OVRF flag, MODF flag, PERF flag, and UDRF flag in SPSR are not initialized. The SPSSR register value is not initialized either. For this reason, it is possible to read receive buffer data and check the communication completion status and the error status during SPI transfer also after the SPI is initialized.

SPSR.SPTEF flag is initialized to 1. Therefore, if the SPCR.SPTIE bit is set to 1 after the SPI is initialized, an SPI transmit buffer empty interrupt is generated. When the CPU initializes the SPI, write 0 to the SPE bit and the SPTIE bit to disable an SPI transmit buffer empty interrupt.

34.3.12.2 System Reset

In initialization by using a system reset, all SPI control bits, status bits, and data register are initialized in addition to the initialization described in section 34.3.12.1. Initialization by Clearing the SPE Bit to completely initialize the SPI.

34.3.13 SPI Operation

34.3.13.1 Master Mode Operation

Single master mode operation and multi-master mode operation are different from each other only in mode fault error detection (see section 34.3.10. Error Detection). The SPI in single master mode (SPI) detects no mode fault error.

The SPI in multi-master mode detects a mode fault error. The following describes operations common to single master mode and multi-master mode.

(1) Starting a Serial Transfer

When data is written to the SPDR register while the next transfer data is not set in the transmit FIFO, the SPI updates the transmit buffer (SPTXn, n = 0 to 3) data in SPDR.

While the shift register is empty, the SPI copies transmit buffer data to the shift register to start serial transfer. After the SPI copies transmit data to the shift register, it changes the shift register status to full. Upon completion of serial transfer, the SPI changes the shift register status to empty. The shift register status cannot be monitored.

For details about the SPI transfer format, see section 34.3.5. Transfer Format (Frame Format). The SSL output signal polarity depends on the set SSLP register value.

(2) Terminating a Serial Transfer

[Except Receive-only in Master Mode]

After the SPI detects the RSPCK edge corresponding to the final sampling timing regardless of the SPCMD.CPHA bit value, the SPI terminates serial transfer. When the number of data stored in the receive FIFO < the number of FIFO stages, data is copied from the shift register to the receive buffer in the SPDR register after serial transfer.

The final sampling timing varies depending on the transfer data bit length. The data length of the SPI in master mode depends on the set value of the SPCMD.SPB[4:0] bits. The SSL output signal polarity depends on the set SSLP register value. For details about the SPI transfer format, see section 34.3.5. Transfer Format (Frame Format).

[Receive-only in Master Mode]

When any of the following 2 conditions is met, then SPI terminating the serial transfer.

- After the SPI detects the RSPCK edge corresponding to the final sampling timing regardless of the SPCMD.CPHA bit value, the SPI terminates serial transfer.
- When writing SPCRRM.RMEDTG = 1 during the serial transfer period, the SPI terminates the serial transfer.

When the number of data stored in the receive FIFO < the number of FIFO stages, data is copied from the shift register to the receive buffer in the SPDR register after serial transfer.

The final sampling timing varies depending on the transfer data bit length. The data length of the SPI in master mode depends on the set value of the SPCMD.SPB[4:0] bits. The SSL output signal polarity depends on the set SSLP register value. For details about the SPI transfer format, see section 34.3.5. Transfer Format (Frame Format).

(3) Sequence Control

The transfer format in master mode is determined as follows.

The transfer format in master mode is determined by the SPSCR register, SPCMDm (m = 0 to 7) register, SPBR register, SPCKD register, SSLND register, and SPND register.

The SPSCR register is used to determine the sequence configuration for serial transfer to be performed by the SPI in master mode. The SSL output signal value, MSB first/LSB first, data length, a part of bit rate settings, RSPCK polarity and phase, SPCKD enable/disable, SSLND enable/disable, and SPND enable/disable are set in SPCMD0 to SPCMD7.

A part of bit rate settings is set in SPBR, the SPI clock delay value is set in SPCKD, the SSL negation delay value is set in SSLND, and the access delay value is set in SPND.

The SPI configures the sequence that structures a part of or whole of SPCMD0 to SPCMD7 according to the sequence length specified in SPSCR. The SPI has a pointer to SPCMD that configures the sequence. This pointer value can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPE bit in the SPCR register is set to 1 to enable the SPI function, the SPI sets the pointer to commands in SPCMD0 and applies the SPCMD0 setting to the transfer format at the start of serial transfer. The SPI increments the pointer the next-access delay period ends in each data transfer. When the serial transfer corresponding to the final command that configures the sequence is completed, the SPI sets the pointer in SPCMD0. Thus sequence is repeatedly executed.

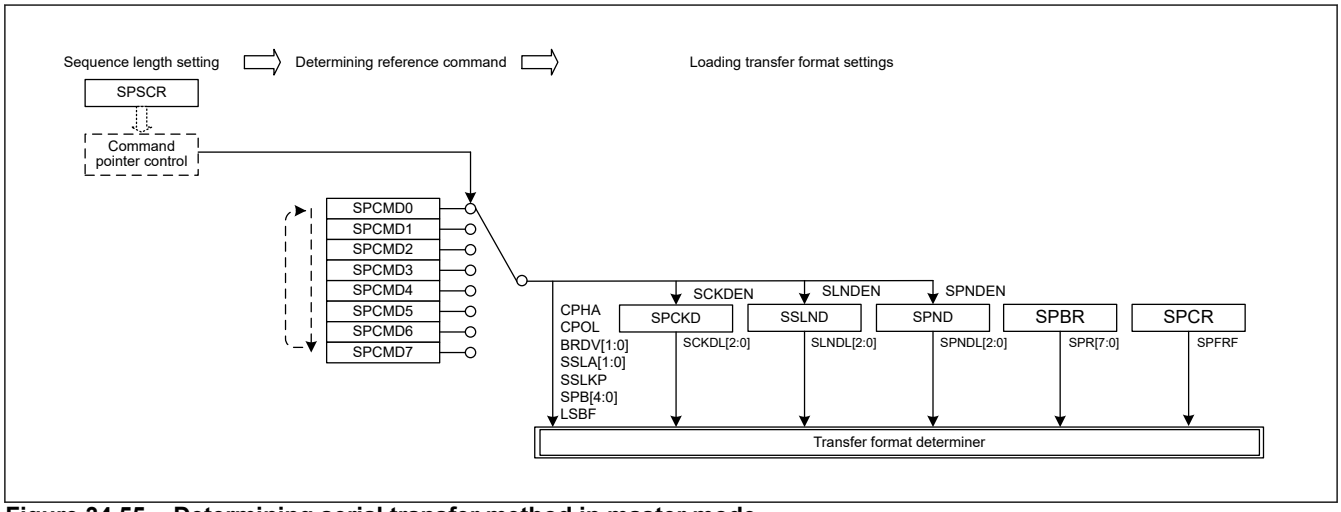


Figure 34.55 Determining serial transfer method in master mode

In this section, a frame consists of data (SPDR) and configuration (SPCMDm).

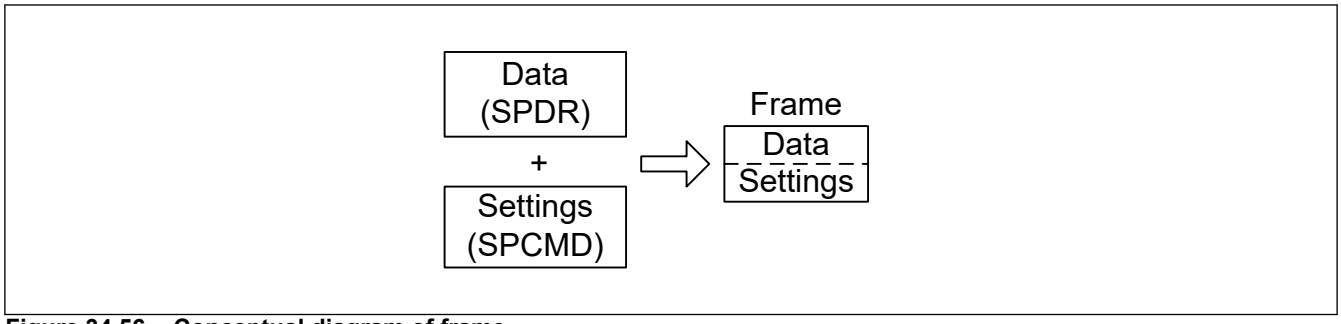


Figure 34.56 Conceptual diagram of frame

Figure 34.57 shows the correspondence between the SPI command register and transmit buffer (SPTXn, n = 0 to 3)/receive buffer (SPRXn, n = 0 to 3) in the sequence operation.

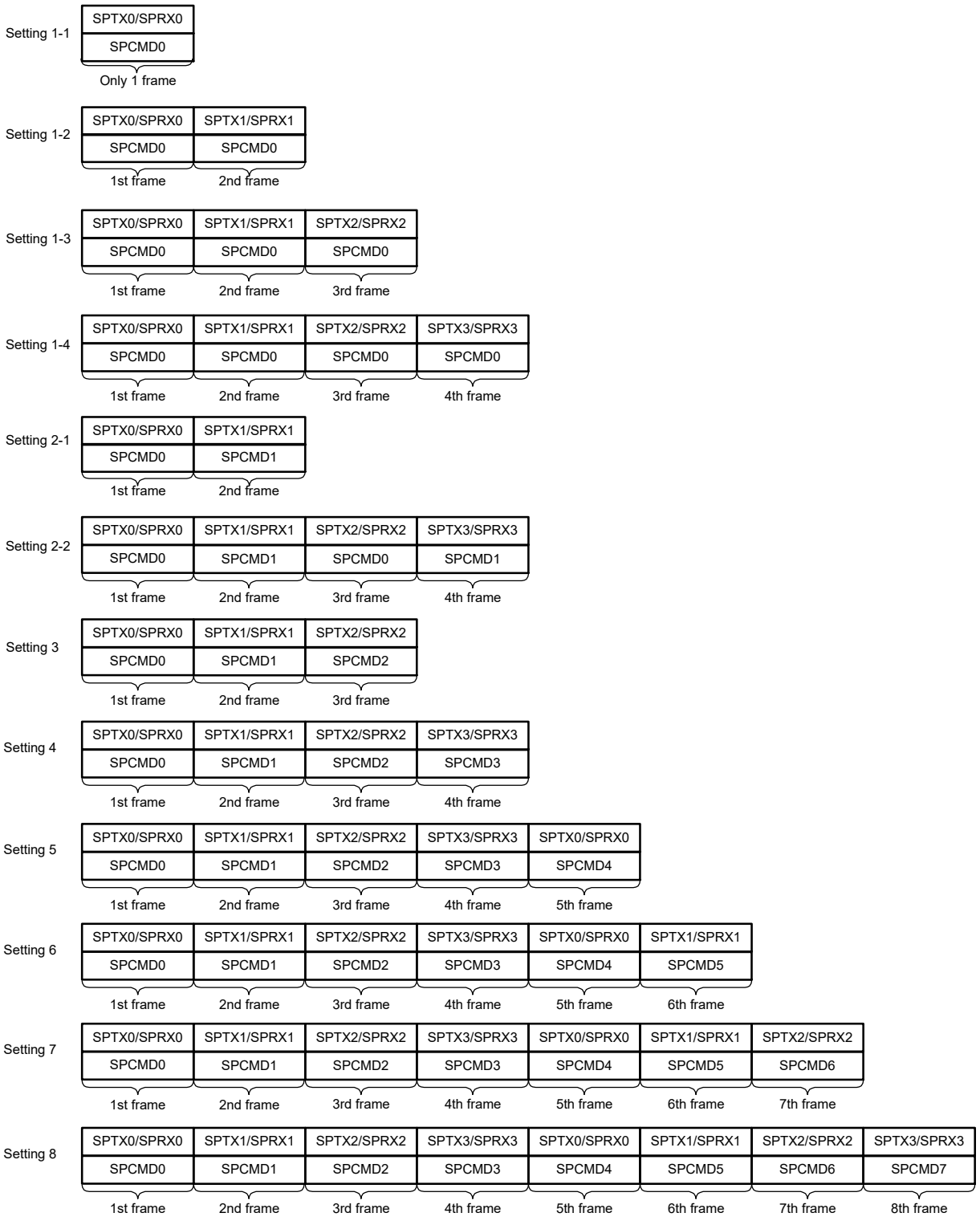


Figure 34.57 Correspondence between SPI command register and transmit/receive buffers in sequence operation

(4) Burst Transfer

This section describes burst transfer during transmit-receive / transmit-only operation.

[In the Motorola-SPI case]

When the SPCMD.SSLKP bit, which the SPI references in the current serial transfer, is 1, the SPI retains the SSL signal level during serial transfer until the SSL signal assertion of the next serial transfer starts. When the SSL signal level in the next serial transfer is the same as the SSL signal level in the current serial transfer, the SPI can continuously perform serial transfer while holding the SSL signal assertion status (burst transfer).

(a) SPCR.BFDS = 0

Figure 34.58 shows an example of SSL signal operation when burst transfer is achieved by using the settings of SPCMD0 and SPCMD1. The following describes SPI operations of (1) to (8) shown in Figure 34.58. The SSL output signal polarity depends on the set the SSLP register.

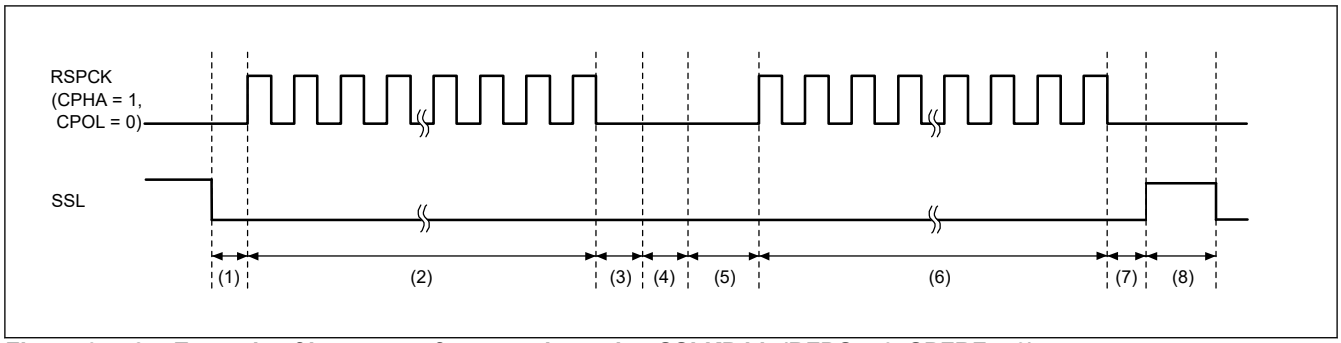


Figure 34.58 Example of burst transfer operation using SSLKP bit (BFDS = 0, SPFRF = 0)

(1) Assert the SSL signal and insert an RSPCK delay according to SPCMD0.

(2) Perform serial transfer according to SPCMD0.

(3) Insert an SSL negation delay.

(4) The SSL signal value in SPCMD0 is retained because SPCMD0.SSLKP = 1. This period additionally continues for 5 PCLKSPIn cycles (at minimum) that is the same as the next-access delay time of SPCMD0. When the shift register is empty after the minimum period has passed, this period continues until transmit data for the next transfer is stored in the shift register.

(5) Assert the SSL signal and insert an RSPCK delay according to SPCMD1.

(6) Perform serial transfer according to SPCMD1.

(7) Insert SSL negate delay.

(8) The SSL signal is negated because SPCMD1.SSLKP = 0. Furthermore, the next-access delay is inserted according to SPCMD1.

If the SSL signal output setting in SPCMD with the SSLKP bit set to 1 differs from the SSL signal output setting in SPCMD to be used for the next transfer, the SPI changes the SSL signal state when the SSL signal corresponding to the next-transfer command is asserted (5). Note that, if an SSL signal change like this takes place, slaves that drive the MISO signal may conflict with each other, which may cause collision of signal level.

The SPI in master mode references the SSL signal operation in the module when SSLKP is not used. Even when SPCMD.CPHA = 0, the SPI can accurately start serial transfer by using the next-transfer SSL signal assertion detected internally. For this reason, burst transfer in master mode is enabled regardless of the set CPHA bit value. (See section 34.3.13. SPI Operation.)

(b) SPCR.BFDS = 1

Figure 34.59 shows an example of SSL signal operation when burst transfer is achieved by using the settings of SPCMD0 and SPCMD1. The following describes SPI operations of (1) to (6) shown in Figure 34.59. The SSL output signal polarity depends on the set SSSLP register value.

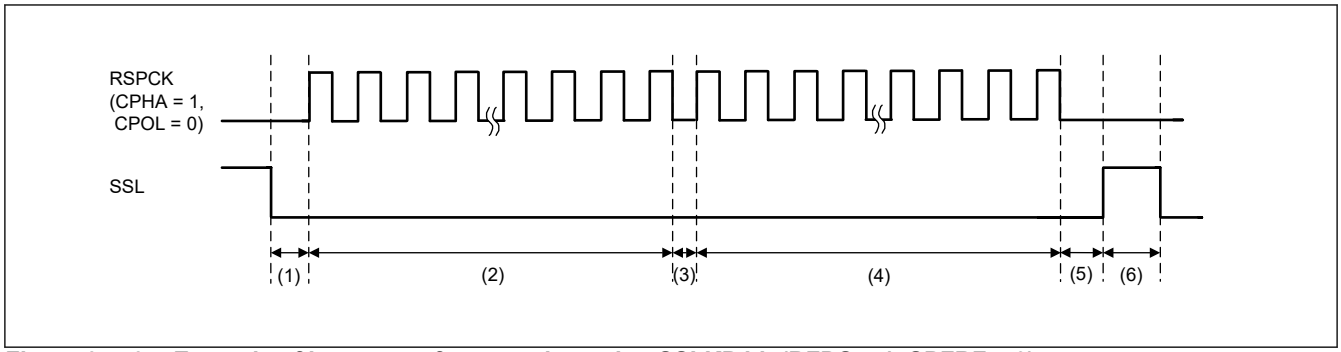


Figure 34.59 Example of burst transfer operation using SSLKP bit (BFDS = 1, SPFRF = 0)

(1) Assert the SSL signal and insert an RSPCK delay according to SPCMD0. The RSPCK delay is inserted only the first frame of burst transmisshon.

(2) Perform serial transfer according to SPCMD0. Wait last clock until the next transmit data is stored in the shift register, if the shift register is empty during RSPCK negate period between frames.

(3) The value of SSL signal according to SPCMD0 was hold, because SPCMD0.SSLKP = 1. RSPCK negate period between frames is 0.5RSPCK, if the shift register is not empty.

(4) Perform serial transfer according to SPCMD1.

(5) Insert SSL negate delay for the last frame.

(6) The SSL signal is negated because SPCMD1.SSLKP = 0. Furthermore, the next-access delay is inserted according to SPCMD1.

[In the TI-SSP case]

SPI asserts the SSL signal for one cycle at the start of serial transfer.

Serial transfer can be executed continuously by asserting the SSL signal for one cycle at the start of the next serial transfer (burst transfer).

(a) SPCMD.SSLKP = 1 and SPCR.BFDS = 1

SPCMD0 to SPCMD1 are shown in Figure 34.60. The following shows an example of SSL signal operation and serial data MISO/MOSI when burst transfer is realized using the settings. The SSL output signal polarity depends on the set SPI slave select polarity register (SSLP) value.

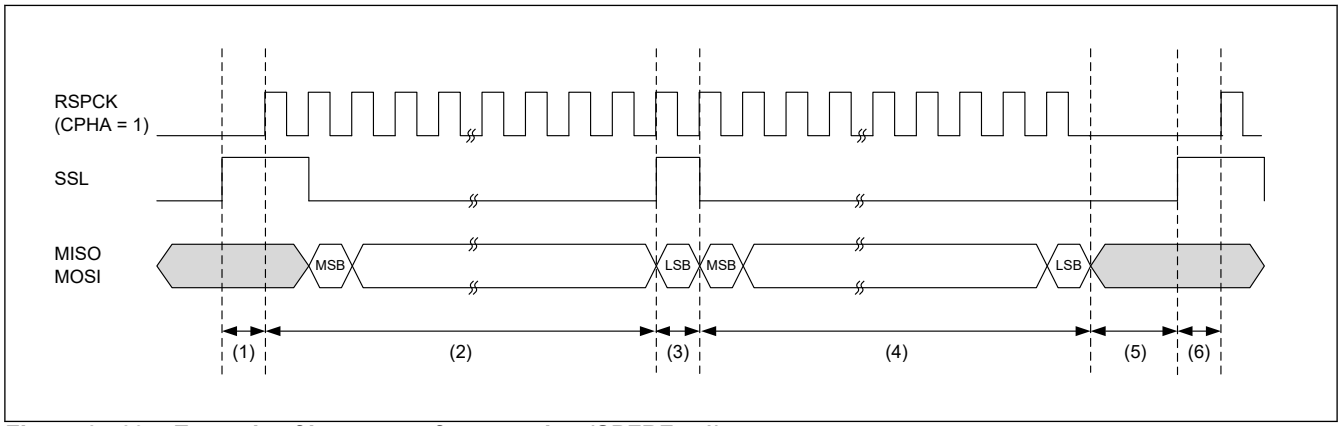


Figure 34.60 Example of burst transfer operation (SPFRF = 1)

(1) Assert the SSL signal and insert an RSPCK delay according to SPCMD0. The RSPCK delay is inserted only the first frame of burst transmission.

(2) Perform serial transfer according to SPCMD0.

(3) Final data transfer and SSL assertion are performed simultaneously. If the shift register is empty during the RSPCK negation period between frames, wait for the output of the last clock until the transmission data for the next transfer is stored in the shift register.

(4) Perform serial transfer according to SPCMD1.

(5) Insert SSL negate delay for the last frame.

(6) The SSL signal is negated because the SPCMD1.SSLKP = 0.

If the SSL signal output setting in SPCMD with the SSLKP bit set to 1 differs from the SSL signal output setting in SPCMD to be used for the next transfer, the SPI changes the SSL signal state when the SSL signal corresponding to the next-transfer command is asserted (5). Note that, if an SSL signal change like this takes place, slaves that drive the MISO signal may conflict with each other, which may cause collision of signal level.

This section describes burst transfer during receive-only operation.

[In the Motorola-SPI case]

When the SPCMD.SSLKP bit, which the SPI references in the current serial transfer, is 1, the SPI retains the SSL signal level during serial transfer until the SSL signal assertion of the next serial transfer starts. When the SSL signal level in the next serial transfer is the same as the SSL signal level in the current serial transfer, the SPI can continuously perform serial transfer while holding the SSL signal assertion status (burst transfer).

(a) SPCR.BFDS = 0

Figure 34.61 shows an example of SSL signal operation when burst transfer is achieved by using the settings of SPCMD0 and SPCMD1. The following describes SPI operations of (1) to (8) shown in Figure 34.61. The SSL output signal polarity depends on the set SSLP register value.

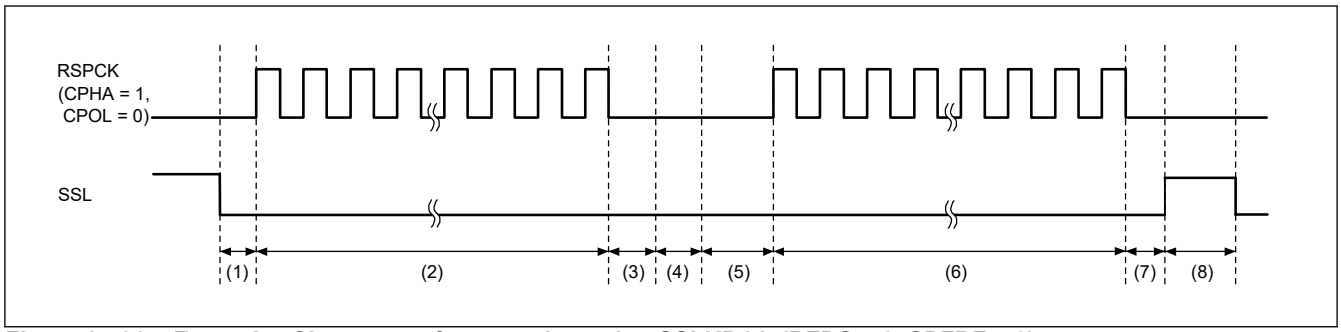


Figure 34.61 Example of burst transfer operation using SSLKP bit (BFDS = 0, SPFRF = 0)

(1) Assert the SSL signal and insert an RSPCK delay according to SPCMD0.

(2) Perform serial transfer according to SPCMD0.

(3) Insert an SSL negation delay.

(4) The SSL signal value in SPCMD0 is retained because SPCMD0.SSLKP = 1. This period additionally continues for 5 PCLKSPIn cycles (at minimum) that is the same as the next-access delay time of SPCMD0.

(5) Assert the SSL signal and insert an RSPCK delay according to SPCMD1.

(6) Perform serial transfer according to SPCMD1.

(7) Insert SSL negate delay.

(8) The SSL signal is negated because SPCMD1.SSLKP = 0. Furthermore, the next-access delay is inserted according to SPCMD1

If the SSL signal output setting and the SSL signal output setting between SPCMDs used for burst transfer are different, SPI switches the SSL signal state when the SSL signal corresponding to the next transfer command is asserted (5). Note that, if an SSL signal change like this takes place, slaves that drive the MISO signal may conflict with each other, which may cause collision of signal level.

The SPI in master mode references the SSL signal operation in the module when SSLKP is not used. Even when SPCMD.CPHA = 0, the SPI can accurately start serial transfer by using the next-transfer SSL signal assertion detected internally. For this reason, burst transfer in master mode is enabled regardless of the set CPHA bit value. (See section 34.3.13. SPI Operation.)

(b) SPCR.BFDS = 1

Figure 34.62 shows an example of SSL signal operation when burst transfer is achieved by using the settings of SPCMD0 and SPCMD1. The following describes SPI operations of (1) to (6) shown in Figure 34.62. The SSL output signal polarity depends on the set SSLP register value.

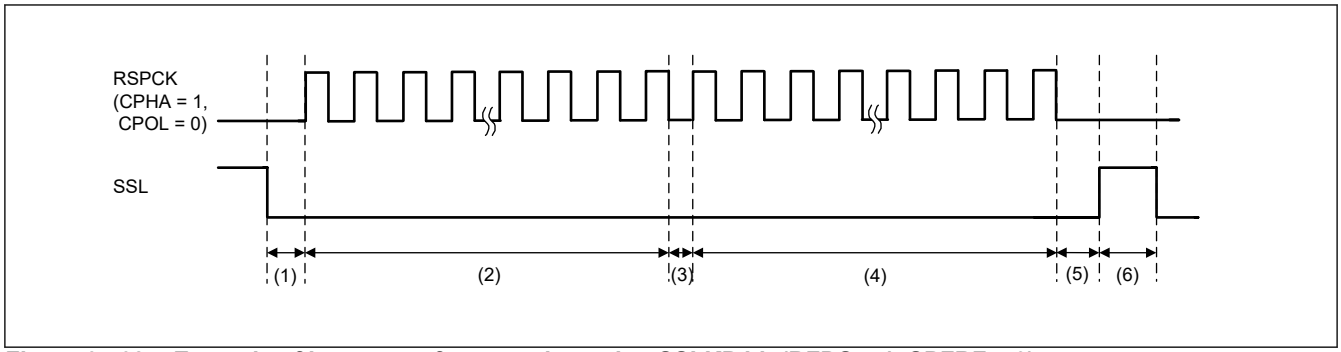


Figure 34.62 Example of burst transfer operation using SSLKP bit (BFDS = 1, SPFRF = 0)

(1) Assert the SSL signal and insert an RSPCK delay according to SPCMD0. The RSPCK delay is inserted only the first frame of burst.

(2) Perform serial transfer according to SPCMD0.

(3) Since it is not the last frame, the SSL signal value at SPCMD0 is retained. RSPCK negation between frames is 0.5 RSPCK for the next frame.

(4) Perform serial transfer according to SPCMD1.

(5) Insert SSL negate delay for the last time.

(6) The SSL signal is negated because SPCMD1.SSLKP = 0. Furthermore, the next-access delay is inserted according to SPCMD1.

Note: "Last frame": Frame set by RMFM[4:0] bits when SPCRRM.RMFM[4:0] ≠ 0x00
Or, a frame in which SPCRRM.RMEDTG = 1 has been accepted.

[In the TI-SSP case]

SPI asserts the SSL signal for one cycle at the start of serial transfer.

Serial transfer can be executed continuously by asserting the SSL signal for one cycle at the start of the next serial transfer (burst transfer).

(a) SPCMD.SSLKP = 1 and SPCR.BFDS = 1

SPCMD0 to SPCMD1 are shown in Figure 34.63. The following shows an example of SSL signal operation and serial data MISO/MOSI when burst transfer is realized using the settings. The SSL output signal polarity depends on the set SSLP register value.

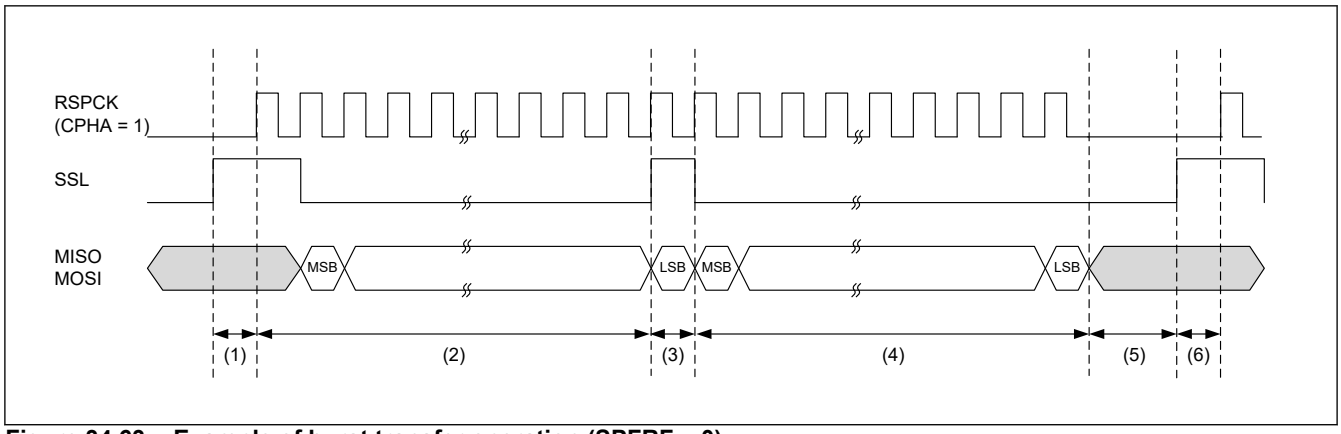


Figure 34.63 Example of burst transfer operation (SPFRF = 0)

(1) Assert the SSL signal and insert an RSPCK delay according to SPCMD0. The RSPCK delay is inserted only the first frame of burst.

(2) Perform serial transfer according to SPCMD0.

(3) Final data transfer and SSL assertion are performed simultaneously.

(4) Perform serial transfer according to SPCMD1.

(5) Insert Output disable delay for the last frame.

(6) The SSL signal is negated because SPCMD1.SSLKP = 0.

Note: "Last frame": Frame set by RMFM[4:0] bits when SPCRRM.RMFM[4:0] ≠ 0x00
Or, a frame in which SPCRRM.RMEDTG = 1 has been accepted.

If the SSL signal output setting between the SPCMDs used for burst transfer differs from the SSL signal output setting, SPI switches the SSL signal state when the SSL signal corresponding to the next transfer command is asserted (5). Note that, if an SSL signal change like this takes place, slaves that drive the MISO signal may conflict with each other, which may cause collision of signal level.

(5) RSPCK Delay (t1)

The RSPCK delay value of the SPI in master mode depends on the settings of the SPCMD.SCKDEN and SPCKD.SCKDL[2:0] bits. The SPI determines SPCMD to be referenced in serial transfer by the pointer control, and then determines the RSPCK delay value for serial transfer by using the selected SPCMD.SCKDEN bit and SPCKD.SCKDL[2:0] bits as shown in Table 34.11. For the definition of RSPCK delay, see section 34.3.5. Transfer Format (Frame Format).

RSPCK delay insert to only the first frame of burst transmission, when transmit without "Between Burst Transfer Frames Delay" (SPCMD.SSLKP = 1, SPCR.BFDS = 1).

Table 34.11 Relationship between SCKDEN/SPCKD and RSPCK delay value

SPCMDm.SCKDEN bit	SPCKD.SCKDL[2:0] bits	RSPCK delay value	
		Motorola-SPI	TI-SSP
0	000b to 111b	1 RSPCK	0 RSPCK
1	000b	1 RSPCK	1 RSPCK
	001b	2 RSPCK	2 RSPCK
	010b	3 RSPCK	3 RSPCK
	011b	4 RSPCK	4 RSPCK
	100b	5 RSPCK	5 RSPCK
	101b	6 RSPCK	6 RSPCK
	110b	7 RSPCK	7 RSPCK
	111b	8 RSPCK	8 RSPCK

(6) SSL Negation Delay (t2)

The SSL negation delay value of the SPI in master mode depends on the settings of the SPCMD.SLNDEN bit and SSLND.SLNDL[2:0]. The SPI determines SPCMD to be referenced in serial transfer by the pointer control, and then determines the SSL negation delay value for serial transfer by using the selected SPCMD.SLNDEN and SSLND.SLNDL[2:0] as shown in Table 34.12. For the definition of SSL negation delay, see section 34.3.5. Transfer Format (Frame Format).

SSL negate delay insert to only the last frame of burst transmission, when transmit without "Between Burst Transfer Frames Delay" (SPCMD.SSLKP = 1, SPCR.BFDS = 1).

Table 34.12 Relationship between SSLND and SSL negation delay value (1 of 2)

SLNDEN	SSLND.SLNDL[2:0] bits	SSL negation delay value
0	000b to 111b	1 RSPCK cycle

Table 34.12 Relationship between SSLND and SSL negation delay value (2 of 2)

SLNDEN	SSLND.SLNDL[2:0] bits	SSL negation delay value
1	000b	1 RSPCK cycle
	001b	2 RSPCK cycle
	010b	3 RSPCK cycle
	011b	4 RSPCK cycle
	100b	5 RSPCK cycle
	101b	6 RSPCK cycle
	110b	7 RSPCK cycle
	111b	8 RSPCK cycle

(7) Next-Access Delay (t3)

The next-access delay value of the SPI in master mode depends on the settings of the SPCMD.SPNDEN bit and SPND.SPNDL[2:0] bits. The SPI determines SPCMD to be referenced in serial transfer by the pointer control, and then determines the RSPCK delay value for serial transfer by using the selected SPCMD.SPNDEN bit and SPND.SPNDL[2:0] bits as shown in Table 34.13. For the definition of next-access delay, see section 34.3.5. Transfer Format (Frame Format).

Next-Access delay insert to only the last frame of burst transmission, when transmit without "Between Burst Transfer Frames Delay" (SPCMD.SSLKP = 1, SPCR.BFDS = 1).

Table 34.13 Relationship between SPNDEN bit, SPND bits, and next-access delay value

SPNDEN bit	SPND.SPNDL[2:0] bits	Next-access delay value
0	000b to 111b	1 RSPCK + 5 PCLKSPIn cycle
1	000b	1 RSPCK + 5 PCLKSPIn cycle
	001b	2 RSPCK + 5 PCLKSPIn cycle
	010b	3 RSPCK + 5 PCLKSPIn cycle
	011b	4 RSPCK + 5 PCLKSPIn cycle
	100b	5 RSPCK + 5 PCLKSPIn cycle
	101b	6 RSPCK + 5 PCLKSPIn cycle
	110b	7 RSPCK + 5 PCLKSPIn cycle
	111b	8 RSPCK + 5 PCLKSPIn cycle

(8) Initialization Flowchart

Figure 34.64 shows an example of initialization flow when using the SPI in master-mode SPI operation. For how to set the interrupt controller, DMAC, and input/output ports, see descriptions of each block.

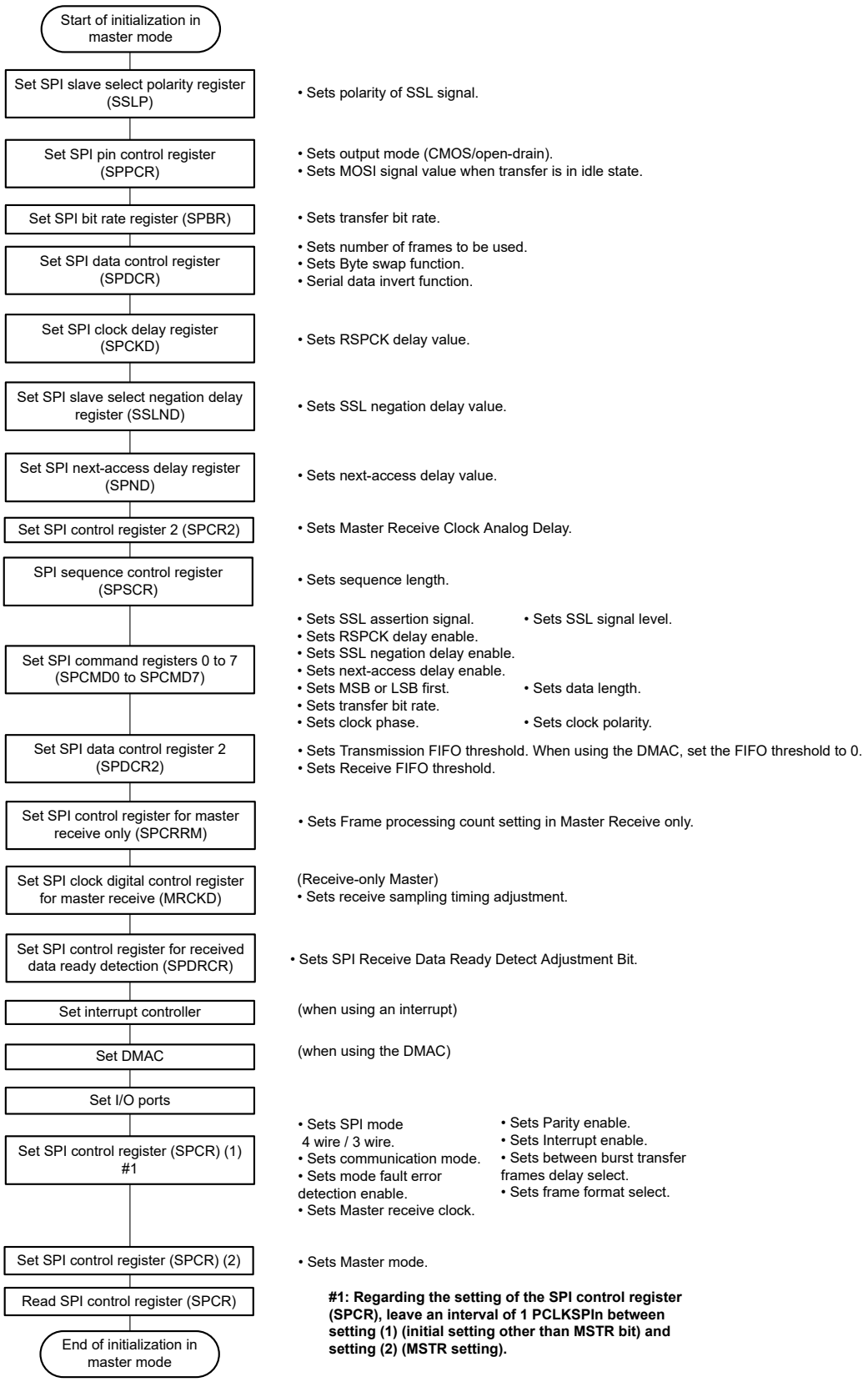


Figure 34.64 Example of initialization flowchart in master mode (SPI operation)

(9) Software Processing Flow

Figure 34.65 to Figure 34.68 show examples of the flow of software processing.

(a) Transmit Processing Flow

Enabling an SPII or SPCEND interrupt after final transmit data is written makes it possible to notify the CPU that transmission of all data has been completed.

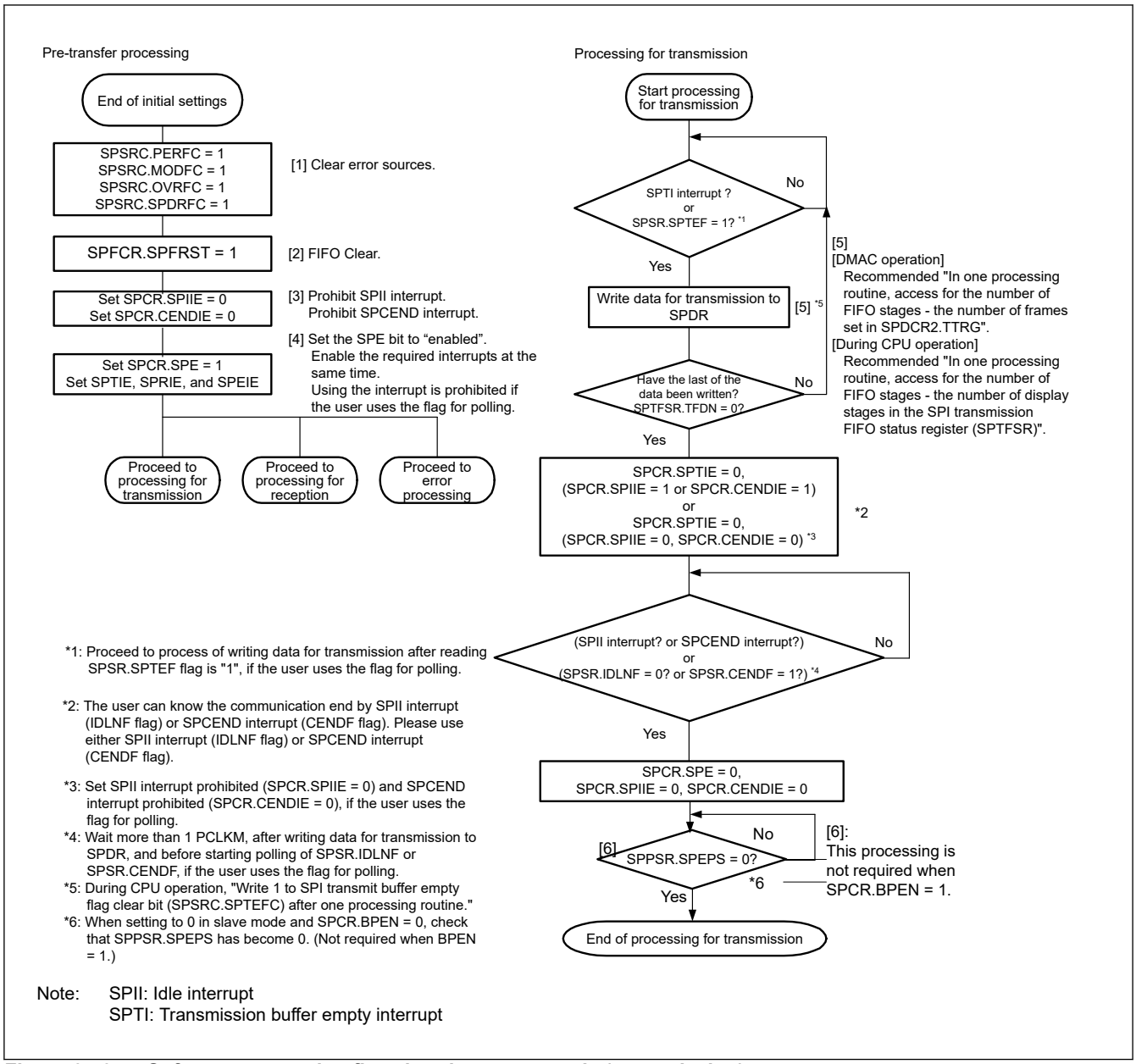


Figure 34.65 Software processing flowchart in master mode (transmission)

(b) Reception Processing Flow

The SPI always requires transmission because it does not have reception-only operation.

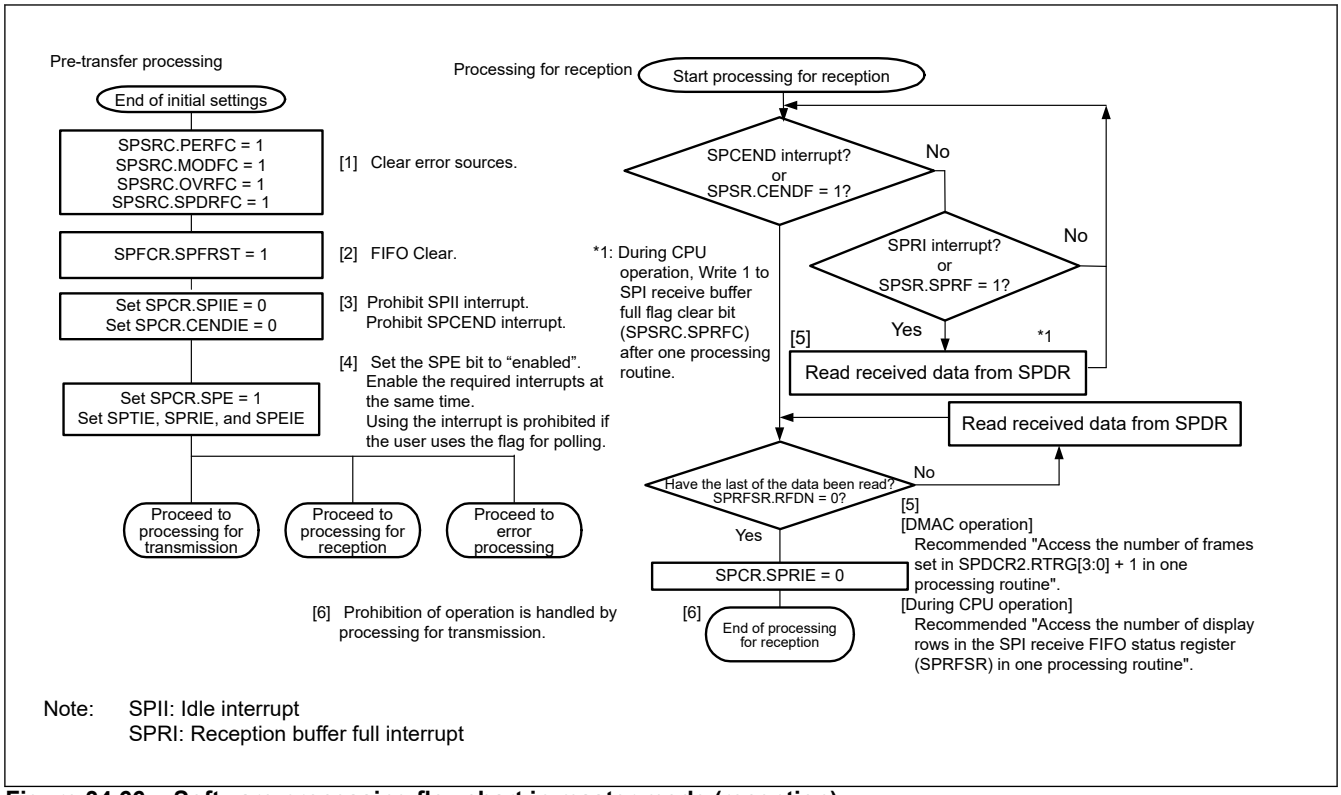


Figure 34.66 Software processing flowchart in master mode (reception)

(c) Master Reception-only processing flow

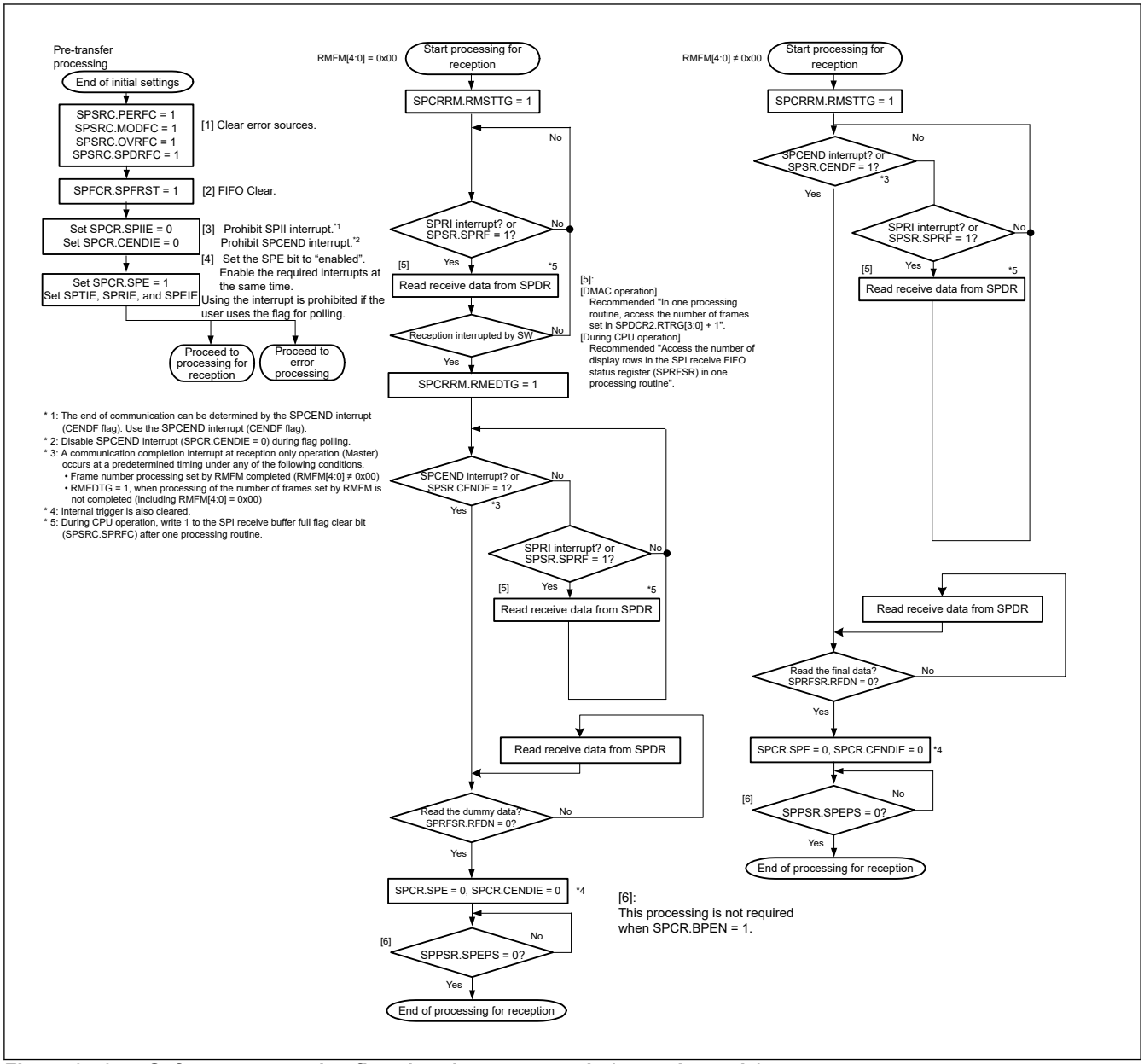


Figure 34.67 Software processing flowchart in master mode (reception-only)

(d) Error processing flow

When a mode fault error occurs, the SPE bit is automatically cleared to stop transmission and reception operations.

However, other error sources do not clear the SPE bit or stop transmission and reception operations. For this reason, if an error occurs due to another source different from the source of the error occurred first, the SPECM[2:0] value is updated. Therefore, we recommend that you clear the SPE bit to stop the ongoing operation.

When interrupts are used and an error has occurred, clear an Interrupt flag in the error processing because an SPTI interrupt request or an SPRI interrupt request may be remaining in an Interrupt flag. Furthermore, when an SPRI interrupt request is remaining, read the receive buffer to initialize the SPI's internal sequencer.

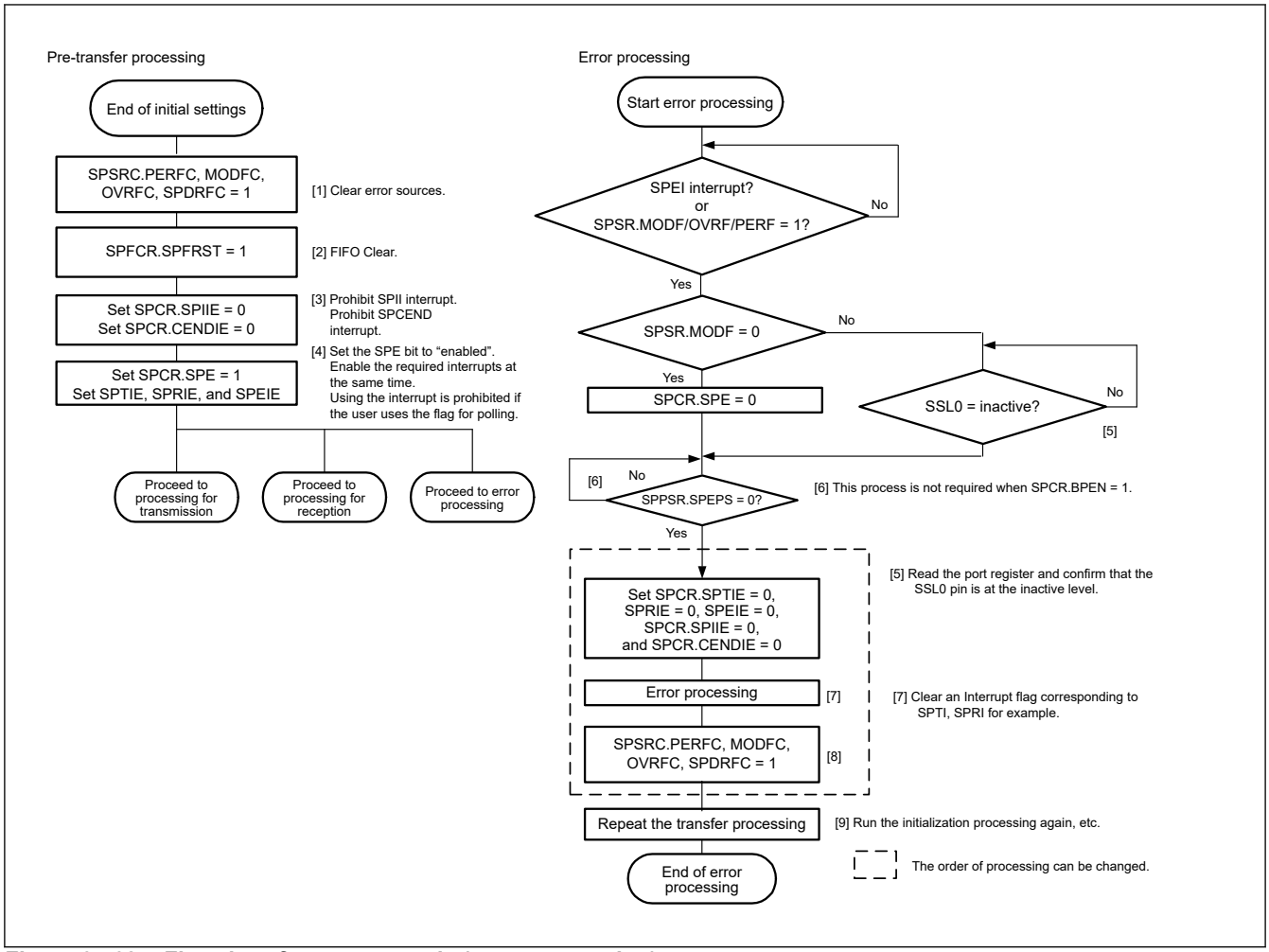


Figure 34.68 Flowchart for master mode (error processing)

34.3.13.2 Slave Mode Operation

(1) Starting a Serial Transfer

When the SPI detects SSL0 input signal assertion while SPCMD0.CPHA = 0, valid data drive to the MISO output signal must be started. For this reason, the SSL0 input signal assertion becomes a trigger to start serial transfer.

While SPCMD0.CPHA = 1, when the SPI detects the first RSPCK edge with the SSL0 input signal asserted, valid data drive to the MISO output signal must be started. For this reason, the first RSPCK edge in the SSL0 signal assertion state becomes a trigger to start serial transfer when SPCMD0.CPHA = 1.

The SPI starts driving the MISO output signal at the SSL0 signal assertion timing independently of the CPHA bit setting. Data that the SPI outputs is enabled or disabled depending on the set CPHA bit value.

For details about the SPI transfer format, see section 34.3.5. Transfer Format (Frame Format). The SSL0 input signal polarity depends on the set SSL0P bit value in the SSLP register.

(2) Terminating a Serial Transfer

Upon detecting the RSPCK edge corresponding to the final sampling timing regardless of the value of SPCMD0.CPHA bit, the SPI terminates serial transfer. When the number of data stored in the receive FIFO is smaller than the number of FIFO stages, the SPI copies received data from the shift register to the receive buffer in the SPDR register after serial transfer. The SPI also changes the shift register status to empty after serial transfer regardless of the receive buffer status. When the SPI detects SSL0 input signal negation during a period from serial transfer start to the end of serial transfer, a mode fault error occurs. (See section 34.3.10. Error Detection.)

The final sampling timing varies depending on the transfer data bit length. The data length of the SPI in slave mode depends on the set value of the SPCMD0.SPB[4:0] bits. The SSL0 input signal polarity depends on the set value of SSLP.SSL0P bit. For details about the SPI transfer format, see section 34.3.5. Transfer Format (Frame Format).

(3) Notes on Single-Slave Operations

[In the Motorola-SPI case]

Upon detecting the SSL0 input signal assertion edge while SPCMD0.CPHA = 0, the SPI starts serial transfer. When using the SPI in single slave mode in a configuration as shown in the example in Figure 34.9, the SPI cannot start serial transfer correctly while CPHA is set to 0 because the SSL0 input signal is always active. To correctly perform transmission and reception of the SPI in slave mode in a configuration where the SSL0 input signal is fixed to active state, set the CPHA bit to 1. If the CPHA bit needs to be set to 0, do not fix the SSL0 input signal.

[In the TI-SSP case]

When TI SSP is set, when SPI is used as a single slave in the configuration shown in the example of Figure 34.9, the SSL0 input signal is always fixed to the inactive state, SPI cannot start serial transfer correctly.

When using a single slave, use the configuration shown in the example in Figure 34.8.

(4) Burst Transfer

[In the Motorola-SPI case]

When SPCMD0.CPHA = 1, continuous serial transfer (burst transfer) is possible while maintaining the SSL0 input signal assertion state. When SPCMD0.CPHA = 1, the period (from the first RSPCK edge in the SSL0 input signal active state until the sampling timing for receiving the final bit) corresponds to the serial transfer period. Even if the SSL0 input signal is always at the active level, start of an access can be detected, which allows burst transfer.

When SPCMD0.CPHA = 0, the second and subsequent serial burst transfers cannot be performed correctly for the same reason as (3).

[In the TI-SSP case]

In serial transfer, data transfer starts after the SSL input signal is asserted for RSPCK 1 cycle. Since frame transfer starts from the SSL input signal, SSL must be asserted between frames.

(5) Initialization Flowchart

Figure 34.69 shows an example of initialization flow when using the SPI in slave-mode SPI operation. For how to set the interrupt controller, DMAC, and input/output ports, see descriptions of each block.

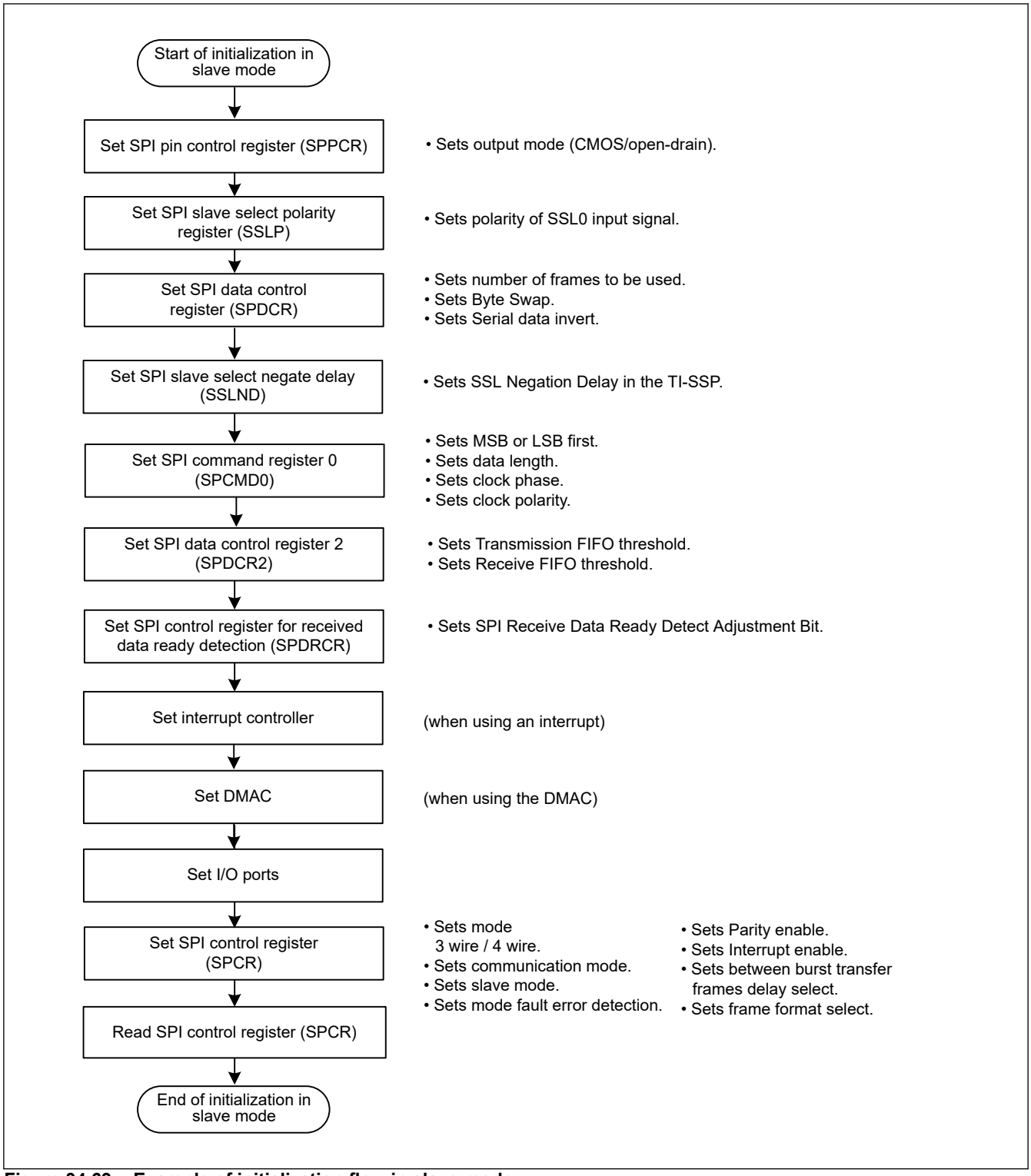


Figure 34.69 Example of initialization flow in slave mode

(6) Software Processing Flow

Figure 34.70 to Figure 34.73 show examples of the flow of software processing.

(a) Transmit Processing Flow

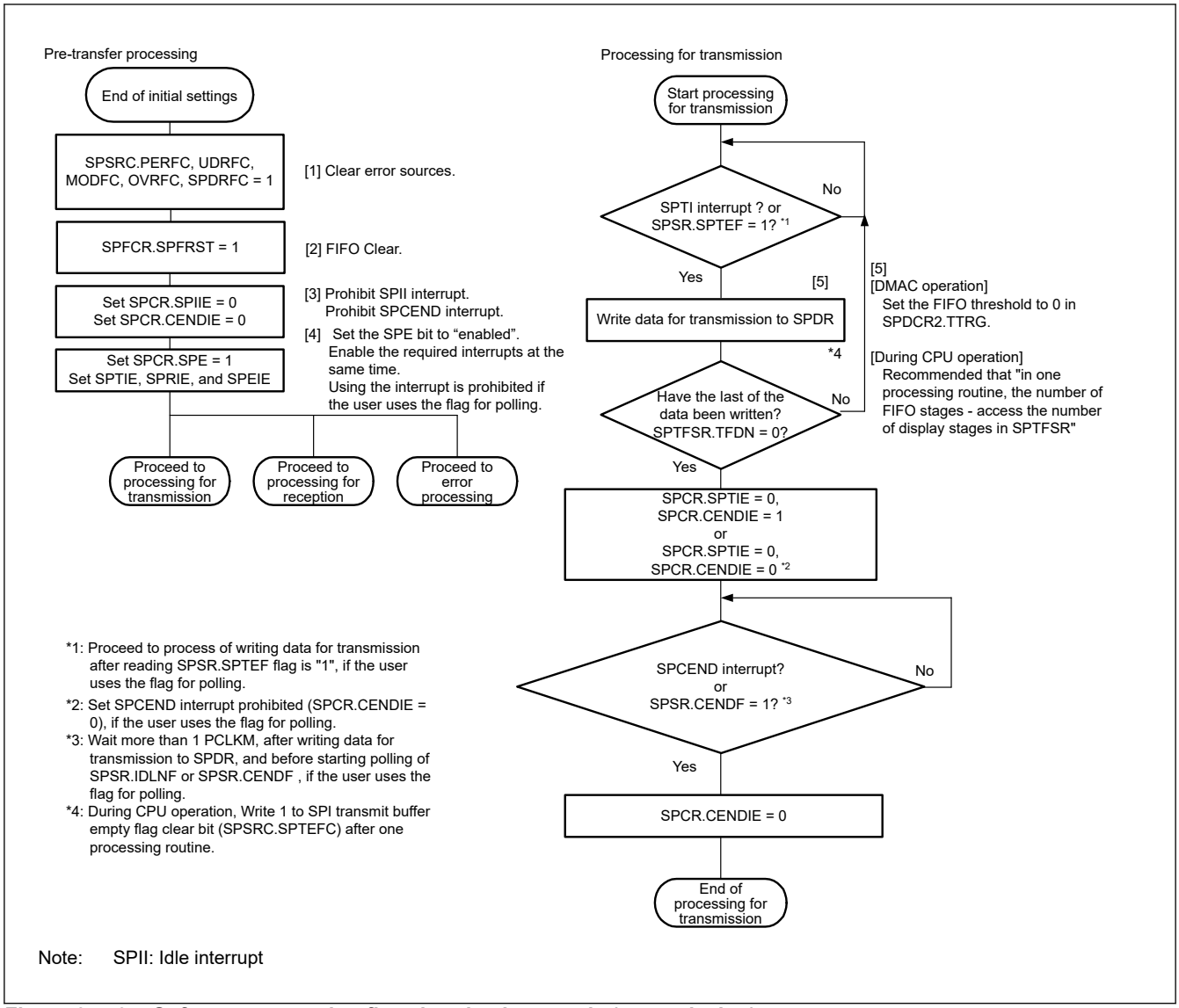
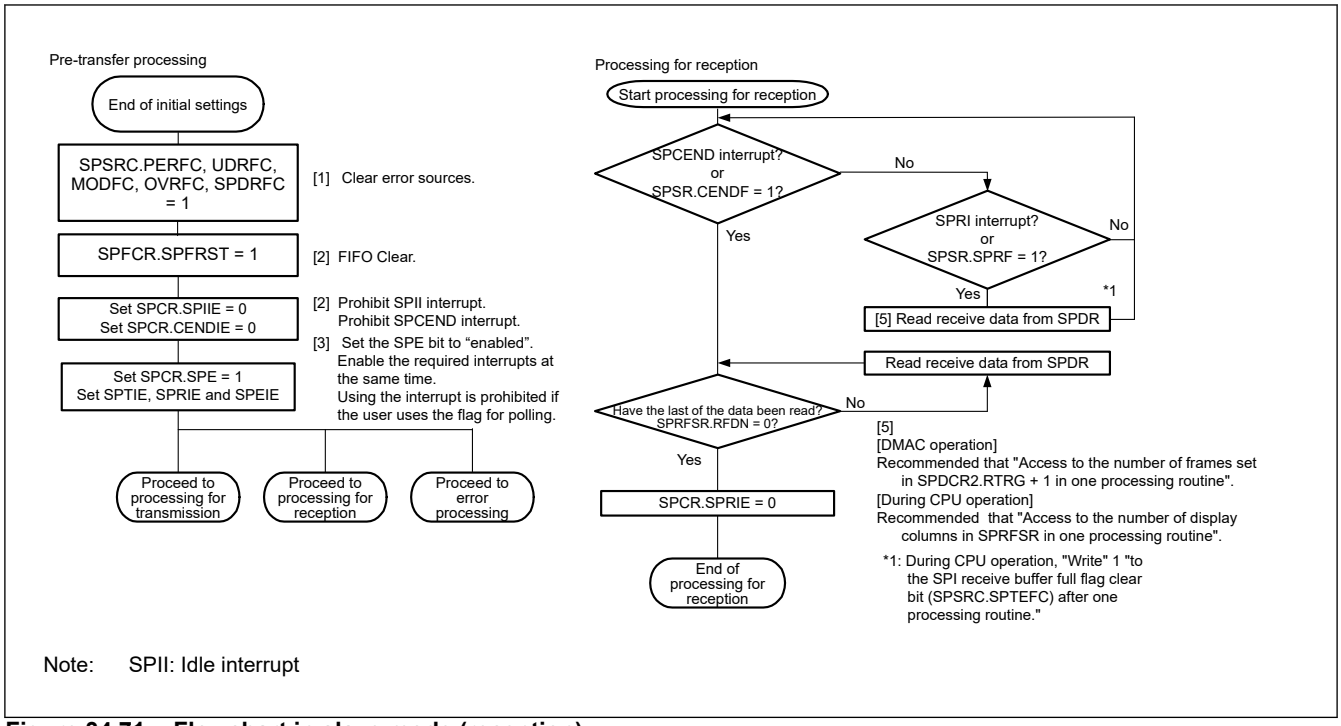


Figure 34.70 Software processing flowchart in slave mode (transmission)

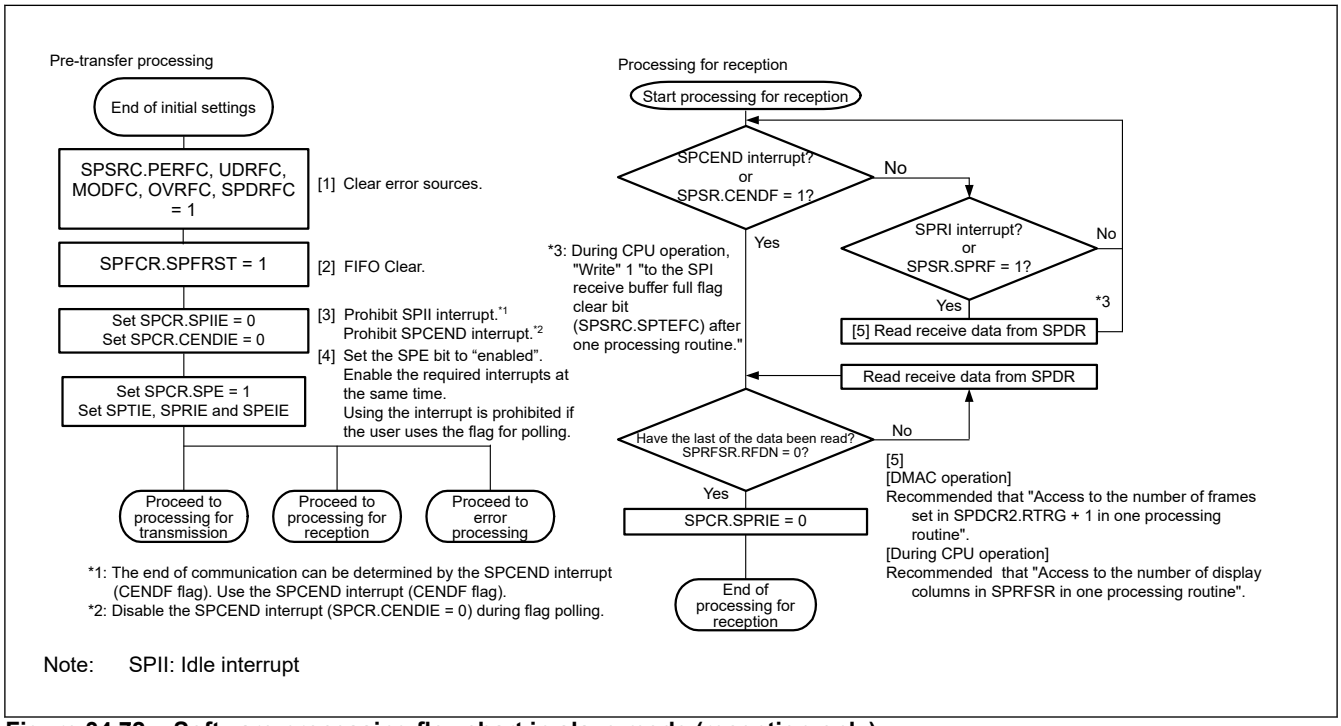
(b) Reception Processing Flow



Note: SPII: Idle interrupt

Figure 34.71 Flowchart in slave mode (reception)

(c) Master Reception-only processing Flow



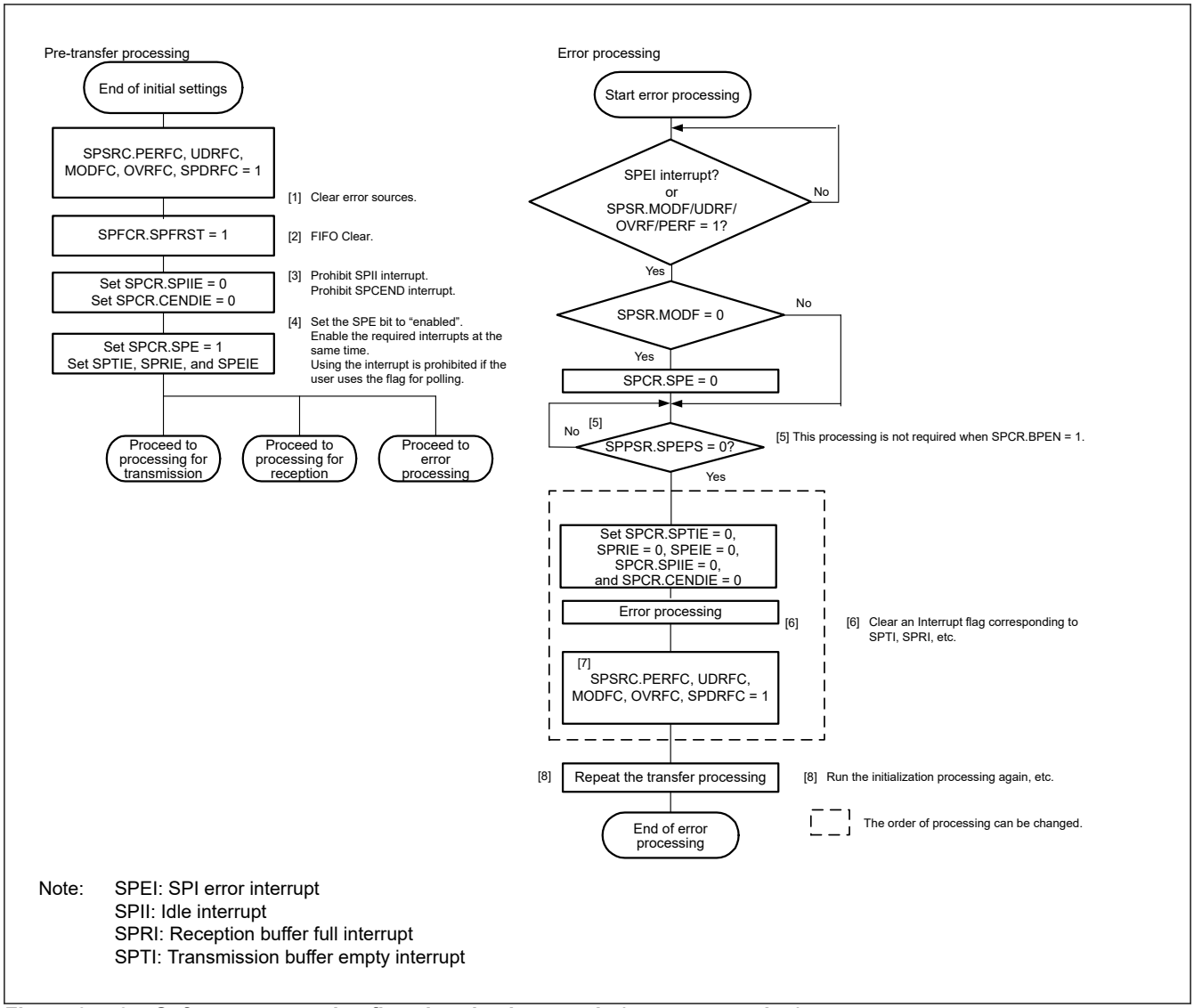
Note: SPII: Idle interrupt

Figure 34.72 Software processing flowchart in slave mode (reception-only)

(d) Error processing flow

In slave mode operation, the MODF flag can be cleared regardless of the SSL0 pin status even when a mode fault error is present.

When interrupts are used and an error has occurred, clear an Interrupt flag in the error processing because an SPTI interrupt request or an SPRI interrupt request may be remaining in an Interrupt flag. Furthermore, when an SPRI interrupt request is remaining, read the receive buffer to initialize the SPI's internal sequencer.



Note: SPEI: SPI error interrupt
SPII: Idle interrupt
SPRI: Reception buffer full interrupt
SPTI: Transmission buffer empty interrupt

Figure 34.73 Software processing flowchart in slave mode (error processing)

34.3.14 Clock Synchronous Operation

When SPCR.SPMS bit = 1, the SPI operates in synchronization with clock. Clock synchronous operation enables communication by using three pins RSPCK, MOSI, and MISO without using SSL pins. SSL pins can be used as I/O ports.

Clock synchronous operation does not use SSL pins for communication, but the module operates internally in the similar manner to SPI operation. In master mode operation and slave mode operation, communication is possible according to the flow similar to SPI operation, but no mode fault error is detected because SSL pins are not used.

In clock synchronous operation, if the SPCMD.CPHA bit is set to 0 in slave mode (SPCR.MSTR = 0), subsequent operation is not guaranteed.

34.3.14.1 Master Mode Operation

(1) Starting a Serial Transfer

When data is written to the SPDR register while the next transfer data is not set in the transmit FIFO, the SPI updates the transmit buffer (SPTXn, n = 0 to 3) data in SPDR. While the shift register is empty, the SPI copies transmit buffer data to the shift register to start serial transfer. After the SPI copies transmit data to the shift register, it changes the shift register status to full. Upon completion of serial transfer, the SPI changes the shift register status to empty. The shift register status cannot be monitored.

For details about the SPI transfer format, see section 34.3.5. Transfer Format (Frame Format). In clock synchronous operation, however, the SSL0 output signal is not used for communication.

(2) Terminating a Serial Transfer

Upon detecting the RSPCK edge corresponding to the final sampling timing, the SPI terminates serial transfer. When the number of data stored in the receive FIFO < the number of FIFO stages, the SPI copies received data from the shift register to the receive buffer in the SPDR register after serial transfer.

The final sampling timing varies depending on the transfer data bit length. The data length of the SPI in master mode depends on the set value of the SPCMD.SPB[4:0] bits. For details about the SPI transfer format, see section 34.3.5. Transfer Format (Frame Format). In clock synchronous operation, however, the SSL0 output signal is not used for communication.

(3) Sequence Control

The transfer format in master mode is determined by the SPSCR, SPCMDm (m = 0 to 7), SPBR, SPCKD, SSLND, and SPND registers. These settings are valid though SSL signals are not output in clock synchronous operation.

The SPSCR register is used to determine the sequence configuration for serial transfer to be performed by the SPI in master mode. MSB first/LSB first, data length, a part of bit rate settings, RSPCK polarity and phase, SPCKD enable/disable, SSLND enable/disable, and SPND enable/disable are set in SPCMD0 to SPCMD7. A part of bit rate settings is set in SPBR, the SPI clock delay value is set in SPCKD, the SSL negation delay value is set in SSLND, and the access delay value is set in SPND.

The SPI configures the sequence that structures a part of or whole of SPCMD0 to SPCMD7 according to the sequence length specified in SPSCR. The SPI has a pointer to SPCMD that configures the sequence. This pointer value can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 to enable the SPI function, the SPI sets the pointer to commands in SPCMD0 and applies the SPCMD0 setting to the transfer format at the start of serial transfer. The SPI increments the pointer when the next-access delay period ends in each data transfer. When the serial transfer corresponding to the final command that configures the sequence is completed, the SPI sets the pointer in SPCMD0. Thus sequence is repeatedly executed.

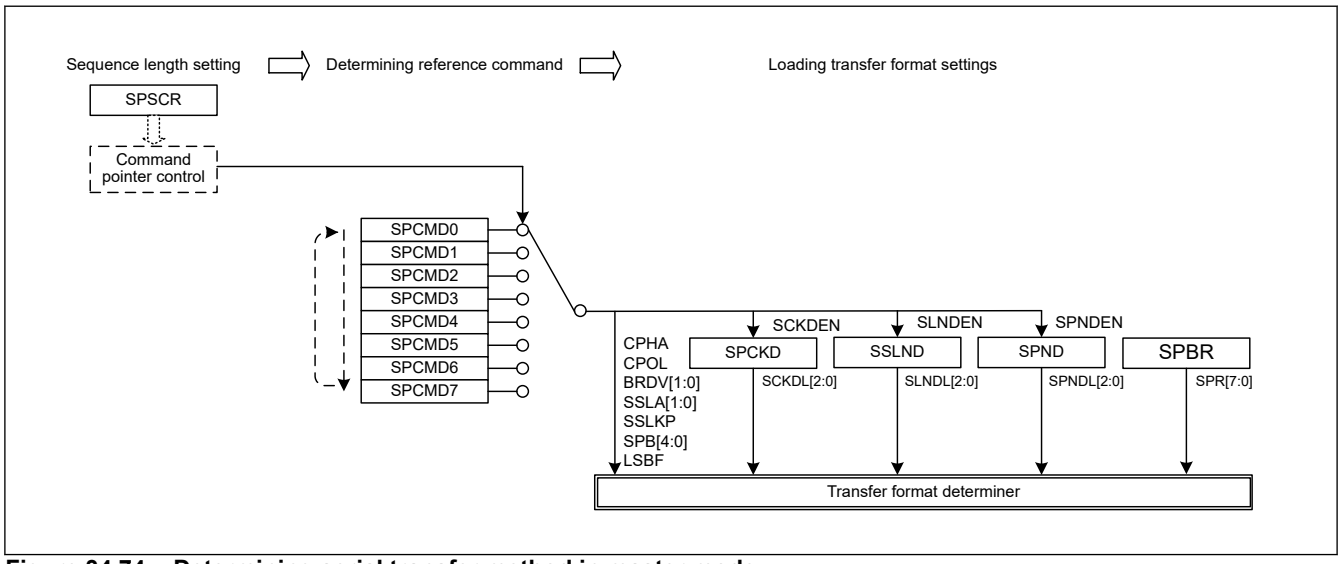


Figure 34.74 Determining serial transfer method in master mode

In this section, a frame consists of data (SPDR) and configuration (SPCMDm (m = 0 to 7)).

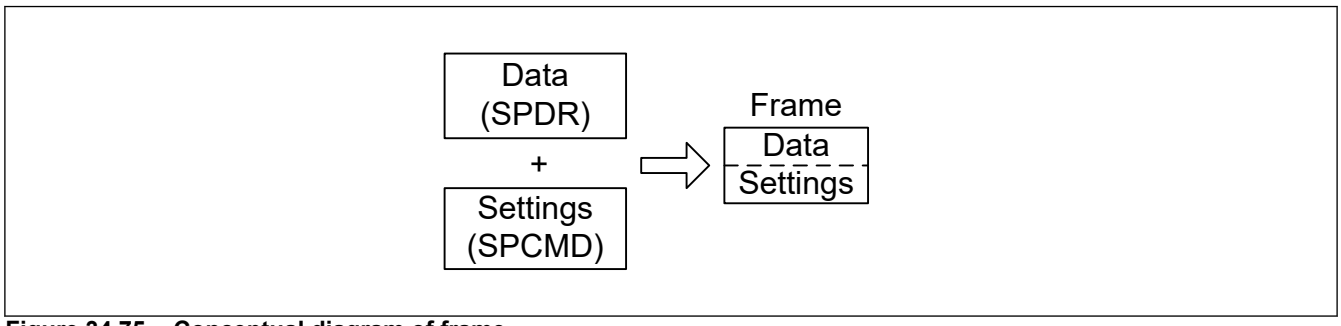


Figure 34.75 Conceptual diagram of frame

Figure 34.76 shows the correspondence between the command in the sequence operation performed and the transmit buffer/ receive buffer.

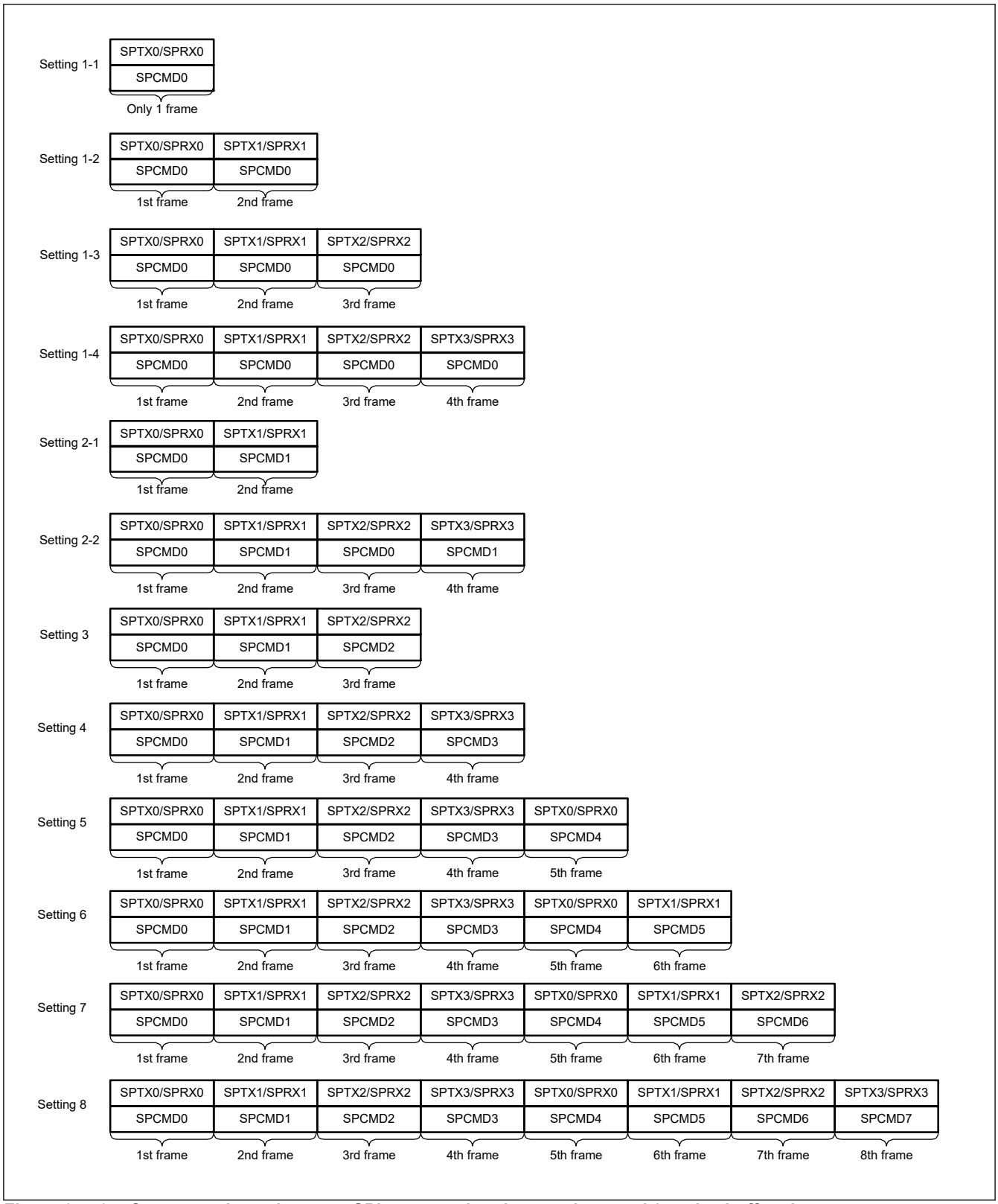


Figure 34.76 Correspondence between SPI command register and transmit/receive buffers in sequence operation

(4) Initialization Flow

Figure 34.77 shows an example of initialization flow when using the SPI in master-mode clock synchronous operation. For how to set the interrupt controller, DMAC, and input/output ports, see descriptions of each block.

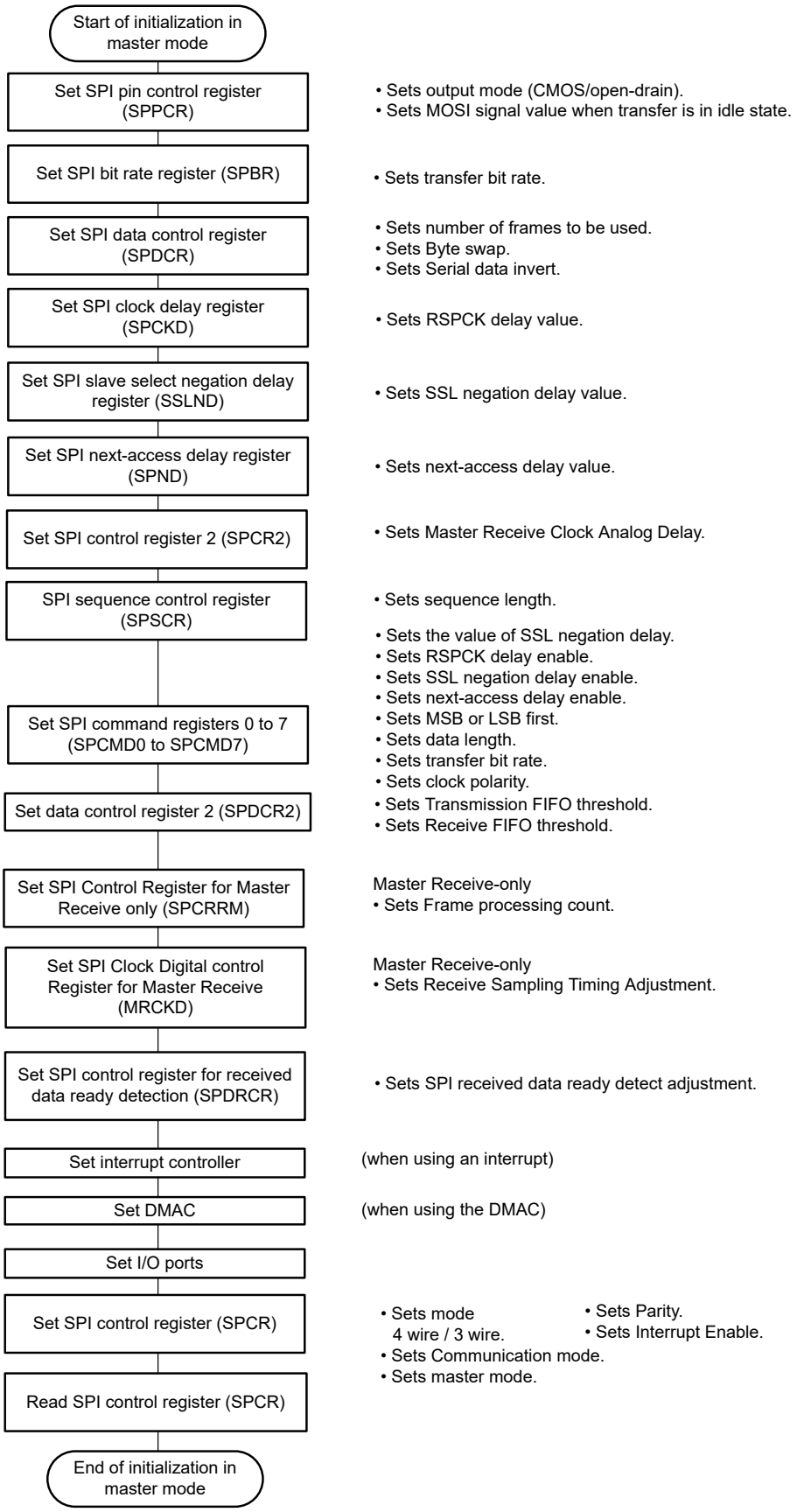


Figure 34.77 Example of initialization flowchart in master mode (clock synchronous operation)

(5) Flow of Software Processing

The software processing flow is omitted here because it is similar to the flow for SPI operation. However, no mode fault error occurs.

34.3.14.2 Slave Mode Operation

(1) Starting a Serial Transfer

When SPCR.SPMS = 1, the first RSPCK edge becomes a trigger to start serial transfer of the SPI.

While SPCR.SPMS = 1, the SPI always drives the MISO output signal.

For details about the SPI transfer format, see section 34.3.5. Transfer Format (Frame Format). In clock synchronous operation, however, the SSL0 input signal is not used.

(2) Terminating a Serial Transfer

Upon detecting the RSPCK edge corresponding to the final sampling timing, the SPI terminates serial transfer. When the number of data stored in the receive FIFO < the number of FIFO stages, the SPI copies received data from the shift register to the receive buffer in the SPDR register after serial transfer. The SPI also changes the shift register status to empty after serial transfer regardless of the receive buffer status. The final sampling timing varies depending on the transfer data bit length. The data length of the SPI in slave mode depends on the set value of the SPCMD0.SPB[4:0] bits. For details about the SPI transfer format, see section 34.3.5. Transfer Format (Frame Format).

(3) Initialization Flowchart

Figure 34.78 shows an example of initialization flow when using the SPI in slave-mode clock synchronous operation.

For how to set the interrupt controller, DMAC, and input/output ports, see descriptions of each block.

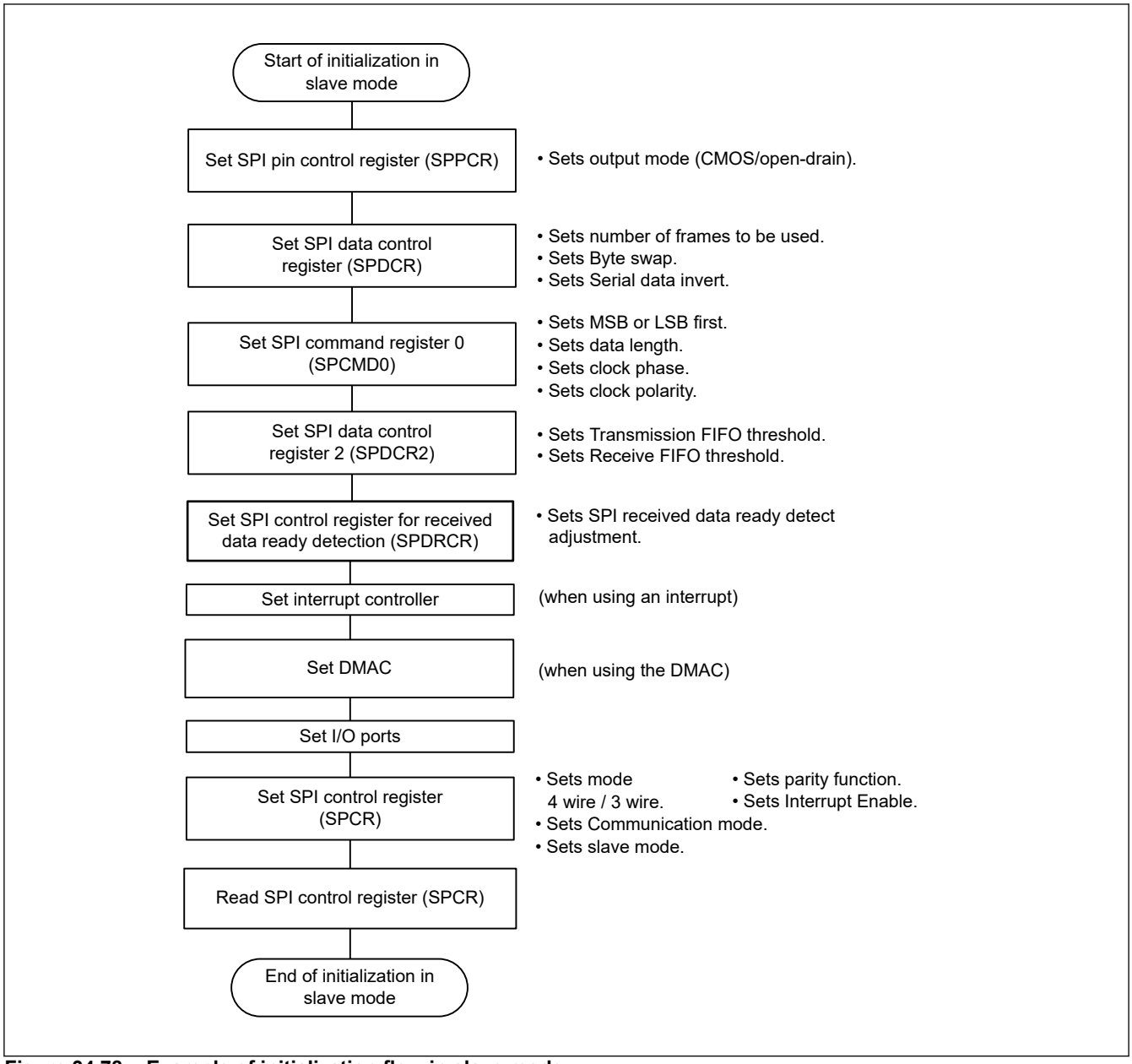


Figure 34.78 Example of initialization flow in slave mode

(4) Flow of Software Processing

The software processing flow is omitted here because it is similar to the flow for SPI operation. However, no mode fault error occurs.

34.3.15 Loopback Mode

When 1 is written to the SPPCR.SPLP bit or SPPCR.SPLP2 bit, the SPI disconnects the path between MISO pin and shift register (when SPCR.MSTR = 1) or between MOSI pin and shift register (when SPCR.MSTR = 0), and then connects the shift register input path to the output path. When SPCR.MSTR = 1, the SPI does not disconnect the path between MOSI pin and shift register. When SPCR.MSTR = 0, the SPI does not disconnect the path between MISO pin and shift register.

When serial transfer is performed in loopback mode, transmit data or inverted transmit data of the SPI becomes received data of the SPI.

Figure 34.79 shows the relationship between settings of the SPLP2 and SPLP bits and received data.

Table 34.14 SPLP2 and SPLP bit settings and received data

SPPCR.SPLP2 bit	SPPCR.SPLP bit	Received data
0	0	Input data from the MOSI pin or MISO pin
0	1	Inversed transmit data
1	0	Transmit data
1	1	Transmit data

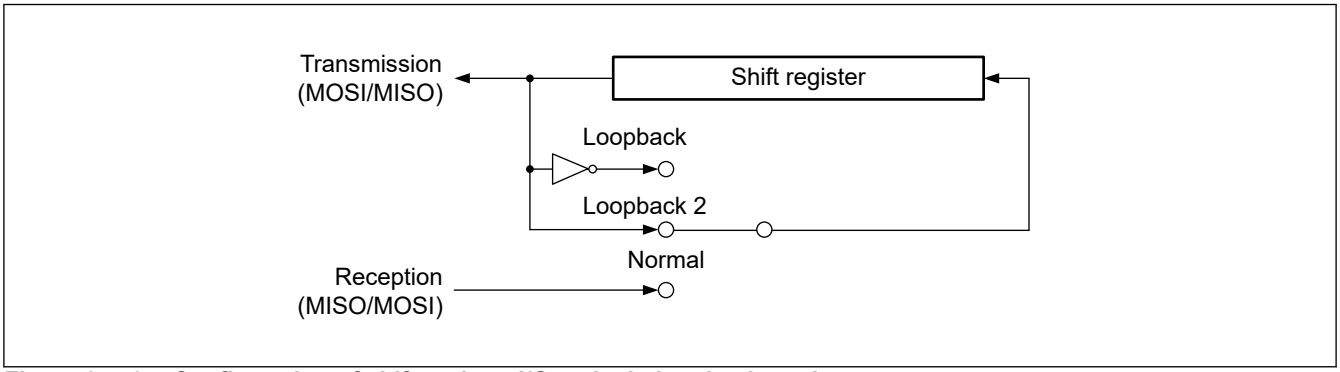


Figure 34.79 Configuration of shift register I/O paths in loopback mode

34.3.16 Self-Diagnosis of Parity Function

The parity circuit consists of a section to add a parity bit to transmit data and a section to detect an error in receive data.

To detect a failure of these sections of the parity circuit, self-diagnosis of the parity circuit is performed according to the flow in Figure 34.80.

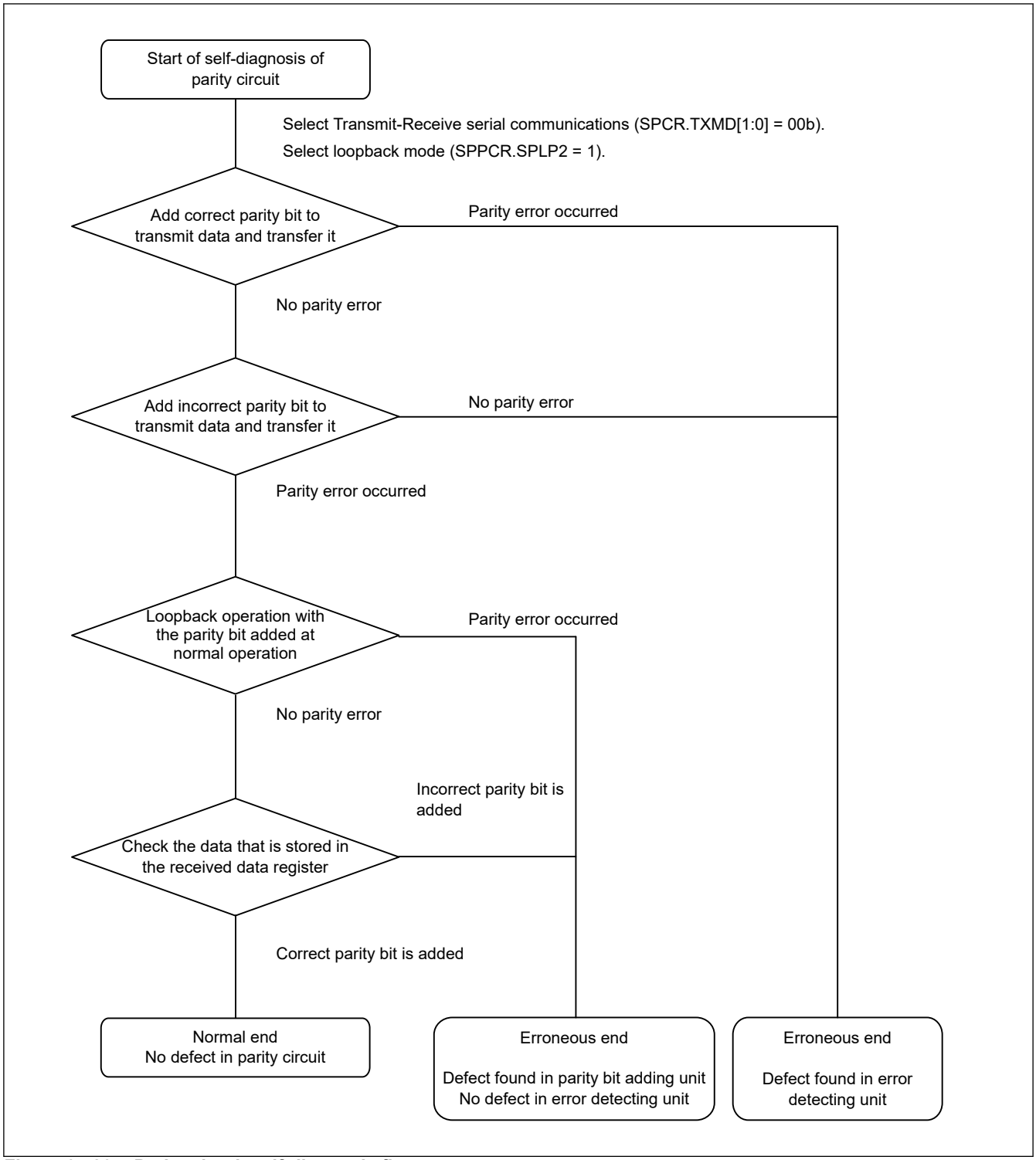


Figure 34.80 Parity circuit self-diagnosis flow

34.3.17 Interrupt Requests

The SPI has the following interrupt sources: receive buffer full, transmit buffer empty, mode fault, underrun, overrun, parity error, SPI idle, communication end, and received data ready. Furthermore, the DMAC can be activated by a receive buffer full interrupt request and a transmit buffer empty interrupt request to start data transfer.

Mode fault, underrun, overrun, parity error, and received data ready (SPCR.SPDRES = 1) interrupt requests are allocated to vector addresses in SPEI. Therefore, identify these interrupt sources by reading respective flags. Table 34.15 lists SPI's interrupt sources. When each interrupt condition in Table 34.15 is met, an interrupt is generated. Clear the receive buffer full and transmit buffer empty interrupt sources by performing data transfer.

When starting transmission or reception by using the DMAC, configure the DMAC to enable it, and then configure the SPI.

Even when the transmit buffer empty interrupt or receive buffer full interrupt condition is met while an Interrupt flag is 1, an interrupt request is not output to ICU but is internally retained. (Only one request per source can be internally retained.) When an Interrupt flag is cleared to 0, the retained interrupt request is output to ICU and then automatically cleared. An internally retained interrupt request can also be cleared by setting the corresponding interrupt enable bit (SPCR.SPTIE bit or SPCR.SPRIE bit) to 0.

Table 34.15 Interrupt sources of SPI

Interrupt source	Symbol	Interrupt condition	DMAC activation
Reception buffer full	SPRI	SPRIE = 1 and (receive buffer full (SPRF = 1) | (SPDRES = 0 and Receive data ready (SPDRF = 1)))	Activated
Transmission buffer empty	SPTI	SPTIE = 1 and transmit buffer empty (SPTEF = 1)	Activated
SPI errors (mode fault, underrun, overrun, and parity error, received data ready)	SPEI	SPEIE = 1 and (MODF = 1 | UDRF = 1 | OVRF = 1 | PERF = 1 | (SPDRES = 1 and SPDRF = 1))	—
SPI idle	SPII	SPIIE = 1 and IDLNF = 0	—
Communication end*1	SPCEND	CENDIE = 1 and CENDF = 1	—

Note 1. See section 34.3.9.7. Common Operation for details on the communication completion interrupt output conditions.

34.4 Usage Notes

34.4.1 Notes on Communication Start

When an Interrupt flag is 1 at the beginning of communication, perform the following procedure to clear an interrupt request before enabling operation (setting the SPCR.SPE bit to 1).

1. Confirm that communication is not going (SPCR.SPE = 0).
2. Set the corresponding interrupt enable bit (SPCR.SPTIE bit or SPCR.SPRIE bit) to 0.
3. Read the corresponding interrupt enable bit (SPCR.SPTIE bit or SPCR.SPRIE bit) and confirm that it is 0.
4. Set the Interrupt flag to 0.

34.4.2 Notes on the Module-Stop Function

To stop this module using the power consumption reducing function, be sure to set the SPE bit in SPCR to 0 to terminate communication and then use the power consumption reducing function.

34.4.3 Notes on SPRF and SPTEF Flags

When using the SPRF and SPTEF flags in SPSR by polling them, set the SPRIE and SPTIE bits in SPCR to 0.

34.4.4 Notes on Receive Sampling Function

When using the receive sampling function (sampling receive data with MRIOCLK or MRCLK), set the SSLND register considering the delay of the receive data sampling clock (MRIOCLK or MRCLK) from RSPCK. Please see section 34.2.3. SSLND : SPI Slave Select Negation Delay Register.

34.4.5 Notes on Burst Transfer in Master Mode

During burst transfer, changing the following settings between SPCMD with the SSLKP bit set to 1 and SPCMD used in the next transfer is prohibited.

1. SSL Signal Assertion setting (SSLA[1:0] bits)
2. RSPCK output setting (CPHA, CPOL, BRDV[1:0] bits)

34.4.6 Notes on TI-SSP in Slave Mode

In the slave TI SSP mode, the delay between frames must observe the following intervals.

Secure the following intervals on the master side.

[Master side: interval from RSPCK edge of last bit to RSPCK edge of next SSL] > [Slave side: Output enable delay time = PCLKSPIn × (1 to 2) + SLNDL[2:0] setting value]

34.4.7 Notes at SPE = 1

(1) When SPCR.SPE = 1, if the following register bits are rewritten, subsequent operations are not guaranteed.

Table 34.16 Communication end event generating conditions (receive only slave mode) (1 of 2)

Register name	Bit name
SPCKD	SCKDL[2:0]
SSLND	SLNDL[2:0]
SPND	SPNDL[2:0]
MRCKD	ARST[2:0]
SPCR	BPEN
	MSTR
	TXMD[1:0]
	SPFRF
	SPMS
	MODFEN
	BFDS
	SCKASE
	PTE
	SPOE
	SPPE
	SPSCKSEL
SPCRRM	RMFM[4:0]
SPDRCR	SPDRC[7:0]
SPPCR	MOIFE
	MOIFV
	SPOM
	SPLP2
	SPLP
SPCR2	SPSCKDL[2:0]
SSLP	SSL3P
	SSL2P
	SSL1P
	SSL0P
SPBR	SPR[7:0]
SPSCR	SPSLN[2:0]

Table 34.16 Communication end event generating conditions (receive only slave mode) (2 of 2)

Register name	Bit name
SPCMD0*1	SSLA[1:0]
	SPB[4:0]
	SCKDEN
	SLNDEN
	SPNDEN
	LSBF
	SSLKP
	BRDV[1:0]
	CPOL
	CPHA
SPDCR	SPFC[1:0]
	SINV
	SPRDTD
	SLSEL[1:0]
	BYSW
SPDCR2	TTRG[1:0]
	RTRG[1:0]
SPFCR	SPFRST

Note 1. Rewriting prohibited in slave mode. In master mode, this is possible only when there is no next transfer data in the transmit FIFO.

(2) After rewriting from SPE = 1 to 0 when SPCR.BPEN = 0, or when SPSR.MODF is set to 1, check that the SPPSR.SPEPS bit is set to 0, then do the next operation.

35. Expanded Serial Peripheral Interface (xSPI)

35.1 Overview

This LSI has two units of Expanded Serial Peripheral Interface (xSPI). The xSPI protocol specifies the interface for Memory Devices, which provides high data throughput, low signal count, and limited backward compatibility with legacy SPI devices. The electrical interface can deliver up to 200 MB per second raw data throughput.

Table 35.1 lists the xSPI specifications, Figure 35.1 shows a block diagram, and Table 35.2 lists the I/O pins.

Table 35.1 xSPI Specifications

Item	Description
Number of channels	2 units
Protocol	Compliant for the xSPI protocol
Data transmission and reception	Issue the transaction for up to 2 Slave as Master
Transfer speed	Support the transfer at xSPI200
Mode	• Support Protocol modes below – 1/4/8pin with SDR/DDR (1S-1S-1S, 4S-4D-4D, 8D-8D-8D) – 2/4pin with SDR (1S-2S-2S, 2S-2S-2S, 1S-4S-4S, 4S-4S-4S) • Configurable address length • Configurable initial access latency cycle • Support XiP mode
xSPI function	• Support Write Data Mask • Support In-band Reset • Memory-mapping – Support up to 64 MB address space each CS – Prefetch function for burst-read with low latency – Outstanding buffer for burst-write with high throughput • Manual command – Configurable up to 4 commands – Status Register Polling function • Input Strobe port timing shift
Interrupt source	2 interrupts
Module-stop function	Module-stop state can be set to reduce power consumption

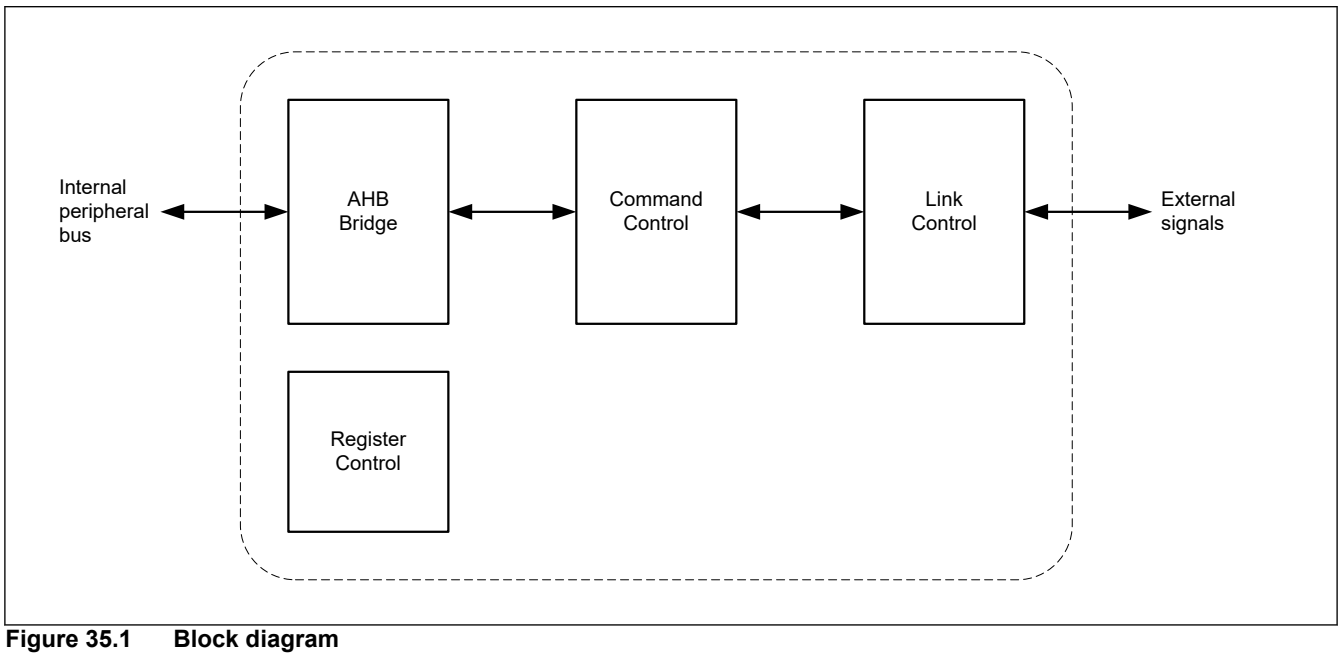


Figure 35.1 Block diagram

Table 35.2 xSPI I/O pins

Channel	Pin name	I/O	Function
xSPIm (m = 0, 1)	XSPIm_CKP	Output	Clock Positive
	XSPIm_CKN	Output	Clock Negative
	XSPIm_CS0#	Output	Chip Select for slave0
	XSPIm_CS1#	Output	Chip Select for slave1
	XSPIm_DS	I/O	Read Data Strobe / Write Data Mask
	XSPIm_IO0	I/O	Data 0 input/output
	XSPIm_IO1	I/O	Data 1 input/output
	XSPIm_IO2	I/O	Data 2 input/output
	XSPIm_IO3	I/O	Data 3 input/output
	XSPIm_IO4	I/O	Data 4 input/output
	XSPIm_IO5	I/O	Data 5 input/output
	XSPIm_IO6	I/O	Data 6 input/output
	XSPIm_IO7	I/O	Data 7 input/output
	XSPIm_RESET0#	Output	Master reset status for slave0
	XSPIm_RESET1#	Output	Master reset status for slave1
	XSPIm_RSTO0#	Input	Slave reset status for slave0
	XSPIm_RSTO1#	Input	Slave reset status for slave1
	XSPIm_INT0#	Input	Interrupt for slave0
	XSPIm_INT1#	Input	Interrupt for slave1
	XSPIm_ECS0#	Input	Error Correction Status for slave0
	XSPIm_ECS1#	Input	Error Correction Status for slave1
	XSPIm_WP0#	Output	Write Protect for slave0
	XSPIm_WP1#	Output	Write Protect for slave1

Table 35.3 Address map

Channel	Internal address	Space
xSPI0	0x4000_0000 to 0x43FF_FFFF	CS0 mirror
	0x4400_0000 to 0x47FF_FFFF	CS1 mirror
	0x6000_0000 to 0x63FF_FFFF	CS0
	0x6400_0000 to 0x67FF_FFFF	CS1
xSPI1	0x4800_0000 to 0x4BFF_FFFF	CS0 mirror
	0x6800_0000 to 0x6BFF_FFFF	CS0

35.2 Register Descriptions

35.2.1 xSPI Configuration Registers

These registers configure xSPI Master function. These registers should be configured in the initialization phase before issuing xSPI transaction.

35.2.1.1 WRAPCFG : xSPI Wrapper Configuration Register

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	DSSFTCS1[4:0]					—	—	—	—	—	—	—	—	—	—	—	DSSFTCS0[4:0]					—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
12:8	DSSFTCS0[4:0]	DS shift for slave0 These bits configure the number of delay cell for XSPIm_DS pin. It is used to adjust the DS sampling timing. 1 cell delay is around 125 ps. When enabled automatic calibration, it can be updated automatically. 0x00: No shift 0x01: Add a delay of 1 cell ⋮ 0x1E: Add a delay of 30 cells 0x1F: Add a delay of 31 cells	R/W
23:13	—	These bits are read as 0. The write value should be 0.	R/W
28:24	DSSFTCS1[4:0]	DS shift for slave1 The function is same as one of slave0.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to configure xSPI Master function.

35.2.1.2 COMCFG : xSPI Common Configuration Register

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x004

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OENEGEX	OEASTEX
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	These bits are read as 0. The write value should be 0.	R/W
16	OEASTEX	Output Enable Asserting extension This bit extends 1 cycle output enable of Data and DS during output enable asserting. When set to 1, CS asserting should be extended (LIOCFGCSn.CSASTEX = 1 (n = 0, 1)). This bit shall not be used in case of no latency cycle. Because xSPI output data could be conflicted with xSPI input data. 0: No extend 1 cycle Output enable 1: Extend 1 cycle Output enable	R/W
17	OENEGEX	Output Enable Negating extension This bit extends 1 cycle output enable of Data and DS during output enable negating. This bit should not be used in case of no latency cycle. Because SPI output data could be conflicted with SPI input data. 0: No extend 1 cycle Output enable 1: Extend 1 cycle Output enable	R/W

Bit	Symbol	Function	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to configure xSPI Master function.

35.2.1.3 BMCFG : xSPI Bridge Map Configuration Register

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	PREEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	MWRSIZE[7:0]								MWRCOMB	—	—	—	—	—	—	WRMD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	WRMD	AHB Write Response mode This bit selects the timing of AHB write response in memory-mapping mode. When set to 1, it returns the response after transmitting a frame on xSPI bus. When enabled this mode, Memory Write Combination mode must be disabled. 0: Return response after storing to Internal Write Buffer 1: Return response after issuing write transaction to xSPI bus	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	MWRCOMB	Memory Write Combination mode This bit selects to combine the xSPI data in write access of memory-mapping mode. When set to 0, xSPI data size depends on AHB Burst type. When set to 1, it depends on Memory Write Size bit. When set to 1, any write transaction could be held in this xSPI master temporarily. 0: Disable combination mode 1: Enable combination mode	R/W
15:8	MWRSIZE[7:0]	Memory Write Size These bits select the size to combine incremental address in memory-mapping mode. It transmits an xSPI frame with the data combined up to the configured size while the address is incremental. When detected non-incremental address or a read transaction before reaching to the target size, it transmits the pending data into xSPI bus. 0x00: Combine incremental address up to 4 bytes 0x01: Combine incremental address up to 8 bytes ⋮ 0x0E: Combine incremental address up to 60 bytes 0x0F: Combine incremental address up to 64 bytes Others: Setting prohibited	R/W
16	PREEN	Prefetch enable This bit enables prefetch function for read transaction in memory-mapping mode. It could reduce the latency for AHB read transaction with incremental address. 0: Disable prefetch function 1: Enable prefetch function	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to configure xSPI Master function.

35.2.1.4 CMCFG0CSn : xSPI Command Map Configuration Register 0 CSn (n = 0, 1)

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x010 + 0x010 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ADDRPCD[7:0]								ADDRPEN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	ADDSIZE[1:0]		FFMT[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	FFMT[1:0]	Frame format These bits configure xSPI frame format in memory-mapping mode. Please see Table 35.9 for detail. 0 0: Normal format: Command 1 byte, Address ADDSIZE, Data to AHB transaction. 0 1: 8D-8D-8D profile 1.0 format: Command 2 bytes, Address ADDSIZE, Data to AHB transaction 1 0: 8D-8D-8D profile 2.0 Command Modifier format: Command & Modifier 6 bytes, Data to AHB transaction 1 1: 8D-8D-8D profile 2.0 Commands with Extended Command Modifier format: Command & Modifier 6 bytes, Data to AHB transaction	R/W
3:2	ADDSIZE[1:0]	Address size These bits configure the number of address byte in memory-mapping mode. In case of 8D-8D-8D profile 2.0, it should be configured to 4 bytes. 0 0: 1 byte (256-byte address space) 0 1: 2 bytes (64 KB address space) 1 0: 3 bytes (16 MB address space) 1 1: 4 bytes (4 GB address space)	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W
23:16	ADDRPEN[7:0]	Address Replace Enable These bits select the bits to replace for MSByte of AHB address in memory-mapping mode. 0: No replacement (xSPI frame address field is same as AHB address) 1: Replacement	R/W
31:24	ADDRPCD[7:0]	Address Replace Code These bits configure the code to replace the MSByte of AHB address in memory-mapping mode. It replaces the corresponding bits when Address Replace Enable bit is set to 1.	R/W

This register has functions to configure xSPI Master function.

35.2.1.5 CMCFG1CSn : xSPI Command Map Configuration Register 1 CSn (n = 0, 1)

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x014 + 0x010 × n

Bit position:	31	20	16	15	0
Bit field:	—	RDLATE[4:0]		RDCMD[15:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0	0 1 0 0 0		0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
15:0	RDCMD[15:0]	Read command These bits configure the command field of read transaction in memory-mapping mode. Normal format and 8D-8D-8D profile 2.0 format use only upper 1 byte. 8D-8D-8D profile 1.0 format uses 2 bytes.	R/W
20:16	RDLATE[4:0]	Read latency cycle These bits configure the latency cycle of read transaction in memory-mapping mode. 0x00: No latency 0x01: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to configure xSPI Master function.

35.2.1.6 CMCFG2CSn : xSPI Command Map Configuration Register 2 CSn (n = 0, 1)

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x018 + 0x010 × n

Bit position:	31											20			16	15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	WRLATE[4:0]				WRCMD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	WRCMD[15:0]	Write command These bits configure the command field of write transaction in memory-mapping mode. Normal format and 8D-8D-8D profile 2.0 format use only upper 1 byte. 8D-8D-8D profile 1.0 format uses 2 bytes.	R/W
20:16	WRLATE[4:0]	Write latency cycle These bits configure the latency cycle of write transaction in memory-mapping mode. 0x00: No latency 0x01: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to configure xSPI Master function.

35.2.1.7 LIOCFGCSn : xSPI Link I/O Configuration Register CSn (n = 0, 1)

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x050 + 0x004 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	DDRSMPEX[3:0]				SDRSMPSFT[3:0]				SDRS MPMD	SDRD RV	CSNE GEX	CSAS TEX	CSMIN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	WRMS KMD	LATE MD	PRTMD[9:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<table>
<tr><th>Bit</th><th>Symbol</th><th>Function</th><th>R/W</th></tr>
<tr><td>9:0</td><td>PRTMD[9:0]</td><td>Protocol mode
These bits configure the protocol mode and the pin to sample data inputs.
In case of using not SPI clock but Data strobe for sampling in SDR mode, it is required to set PRTMD[9] to 1.
0x000: 1S-1S-1S
0x3B2: 4S-4D-4D
0x3FF: 8D-8D-8D
0x048: 1S-2S-2S
0x049: 2S-2S-2S
0x090: 1S-4S-4S
0x092: 4S-4S-4S
Others: Setting prohibited</td><td>R/W</td></tr>
<tr><td>10</td><td>LATEMD</td><td>Latency mode
This bit selects the behavior of initial access latency phase for both direct-manual mode and memory-mapping mode. When set to 0, the latency cycle is equal to each configured cycle from transmitting address field. When set to 1, the latency cycle is incremented from the last byte-pair of Address field and is extended 2 times of each configured cycle depending on data strobe port.
It is used only for profile 2.0 frame format of 8D-8D-8D protocol mode with 6 bytes command/address field. Please refer to xSPI protocol for detail.
<table>
<tr><th>Value: Function</th><th>Frame format</th><th>Usage</th></tr>
<tr><td rowspan="2">0: Configurable latency</td><td>profile 2.0</td><td>The configurable latency cycle should be set as minus 1.</td></tr>
<tr><td>Others</td><td>Latency cycle increments after address field.</td></tr>
<tr><td rowspan="2">1: Variable latency</td><td>profile 2.0</td><td>Latency cycle increments from address [23:16]. And it should not be set to 1.</td></tr>
<tr><td>Others</td><td>Not support</td></tr>
</table></td><td>R/W</td></tr>
<tr><td>11</td><td>WRMSKMD</td><td>Write mask mode
This bit selects to use Data strobe port as write data mask. It could be useful for write access of odd byte.
It is used only for 8D-8D-8D protocol mode.
0: Write mask disable
1: Write mask enable</td><td>R/W</td></tr>
<tr><td>15:12</td><td>—</td><td>These bits are read as 0. The write value should be 0.</td><td>R/W</td></tr>
<tr><td>19:16</td><td>CSMIN[3:0]</td><td>CS minimum idle term
This bit configures the minimum cycle between xSPI frames.
0x0: 1 cycle
0x1: 2 cycles
⋮
0xE: 15 cycles
0xF: 16 cycles</td><td>R/W</td></tr>
<tr><td>20</td><td>CSASTEX</td><td>CS asserting extension
This bit extends 1 cycle chip select pins when asserting.
0: No extension
1: Extend 1 cycle</td><td>R/W</td></tr>
<tr><td>21</td><td>CSNEGEX</td><td>CS negating extension
This bit extends 1 cycle chip select pins when negating.
0: No extension
1: Extend 1 cycle</td><td>R/W</td></tr>
<tr><td>22</td><td>SDRDRV</td><td>SDR driving timing
This bit configures the timing of data output in SDR.
This bit should not be used in case of no latency cycle. Because SPI output data could be conflicted with SPI input data.
0: Drive at 1/2 cycle before CK rising-edge
1: Drive at CK rising-edge</td><td>R/W</td></tr>
</table>

Bit	Symbol	Function	R/W
23	SDRSMPMD	SDR Sampling mode This bit selects the edge of sampling in SDR. In DDR, regardless of this setting, it samples data input ports with both edges of DS. When set to 1, it samples at rising-edge before falling-edge. 0: Samples data input at falling-edge 1: Samples data input at rising-edge	R/W
27:24	SDRSMPSFT[3:0]	SDR Sampling window shift These bits shift the timing of CK sampling in SDR. In case of using DS in SDR, there is no influence on the behavior. In case of DDR or using DS in SDR, it should be set to 0. 0x0: Sample without delay 0x1: Sample at 1 cycle delay ⋮ 0x6: Sample at 6 cycle delay 0x7: Sample at 7 cycle delay Others: Setting prohibited	R/W
31:28	DDRSMPEX[3:0]	DDR sampling window extend These bits configure the cycle of extending the sampling window in DDR. In DDR, the input data is sampled during the expected cycle soon after latency cycle. The input data out of range is ignored. It can be configured depending on DS propagation delay. 0x0: Expand no cycle 0x1: Expand 1 cycle ⋮ 0x6: Expand 6 cycles 0x7: Expand 7 cycles Others: Setting prohibited	R/W

This register has functions to configure xSPI Master function.

35.2.2 xSPI Control Registers

These registers control xSPI Master function.

35.2.2.1 BMCTL0 : xSPI Bridge Map Control Register 0

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x060

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	CS1ACC[1:0]		CS0ACC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
1:0	CS0ACC[1:0]	AHB channel to slave0 memory area access enable These bits enable the access from AHB channel to CS0 memory. 0 0: Read/Write disable 0 1: Read enable, Write disable 1 0: Read disable, Write enable 1 1: Read/Write enable	R/W

Bit	Symbol	Function	R/W
3:2	CS1ACC[1:0]	AHB channel to slave1 memory area access enable These bits enable the access from AHB channel to CS1 memory. 0 0: Read/Write disable 0 1: Read enable, Write disable 1 0: Read disable, Write enable 1 1: Read/Write enable	R/W
7:4	—	These bits are read as 1. The write value should be 1.	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to control xSPI Master function.

35.2.2.2 BMCTL1 : xSPI Bridge Map Control Register 1

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x064

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	PBUFCLR	—	MWRPUSH	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	The write value should be 0.	W
8	MWRPUSH	Memory Write Data Push This bit requests to push the pending data in combination mode. 0: No command 1: Push request	W
9	—	The write value should be 0.	W
10	PBUFCLR	Prefetch Buffer clear This bit requests to clear the prefetch buffer when enabled the prefetch function. It should not be set during memory access (COMSTT.MEMACC = 1). 0: No command 1: Clear request	W
31:11	—	The write value should be 0.	W

This register has functions to control xSPI Master function.

35.2.2.3 CMCTL : xSPI Command Map Control Register

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x068

Bit position:	31	16	15	8	7	0
Bit field:	—	XIPEN	XIPEXCODE[7:0]		XIPENCODE[7:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0	0 0 0 0 0 0 0 0		0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
7:0	XIPENCODE[7:0]	XiP mode enter code These bits configure the code to enter XiP mode in memory-mapping mode.	R/W

Bit	Symbol	Function	R/W
15:8	XIPEXCODE[7:0]	XiP mode exit code These bits configure the code to exit XiP mode in memory-mapping mode.	R/W
16	XIPEN	XiP mode enable This bit enables XiP mode in memory-mapping mode. When set to 1, XiP enter code is inserted in the latency field, and the command field in next transaction is omitted. When set to 0, XiP exit code is inserted in the latency field. And it is set to 0 automatically when transmitting XiP disable pattern. It should not be used for 8D-8D-8D protocol mode profile 2.0 frame format. 0: Disable XiP mode 1: Enable XiP mode	R/W
31:17	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to control xSPI Master function.

35.2.2.4 CSSCTL : xSPI CS Size Control Register

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x06C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CS1SIZE[5:0]						—	—	CS0SIZE[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	CS0SIZE[5:0]	CS0 (slave0) size This bit selects the size of CS0 (slave0). 0x00: 1 MB 0x01: 2 MB 0x03: 4 MB 0x07: 8 MB 0x0F: 16 MB 0x1F: 32 MB 0x3F: 64 MB Others: Setting prohibited	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
13:8	CS1SIZE[5:0]	CS1 (slave1) size This bit selects the size of CS1 (slave1). 0x00: 1 MB 0x01: 2 MB 0x03: 4 MB 0x07: 8 MB 0x0F: 16 MB 0x1F: 32 MB 0x3F: 64 MB Others: Setting prohibited	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to control xSPI Master function.

35.2.2.5 CDCTL0 : xSPI Command Manual Control Register 0

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x070

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	PEREP[3:0]				—	—	—	PERITV[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	TRNUM[1:0]		CSSEL	—	PERMD	TRREQ
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TRREQ	Transaction request This bit requests to issue the transaction of manual-command. When set to 1, it starts the transaction. It is cleared to 0 when the transaction completed. The transaction is canceled by clearing to 0 while the transaction is ongoing. 0: No transaction 1: Request transaction	R/W
1	PERMD	Periodic mode This bit enables the periodic transaction mode. When set to 1, it repeats a transaction periodically and compares the read value with the expected value. It alternates the status polling operation for external memory. 0: Direct manual-command mode 1: Periodic manual-command mode	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	CSSEL	Chip select This bit selects a target memory to issue manual-command. 0: CS0 1: CS1	R/W
5:4	TRNUM[1:0]	Transaction number These bits configure the number of transactions in normal manual-command mode. In periodic manual-command, regardless of this setting, it issues the transaction using only command buffer 0. 0 0: Issue 1 command (using command buffer 0) 0 1: Issue 2 commands (using command buffer 0-1) 1 0: Issue 3 commands (using command buffer 0-2) 1 1: Issue 4 commands (using command buffer 0-3)	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W
20:16	PERITV[4:0]	Periodic transaction interval These bits configure the interval of transaction in periodic manual-command mode. Too short interval compared with CPU bus cycle could result in no store into command buffer0. The interval should be longer than 4 times of CPU bus cycle. 0x00: 2 (= 2^1) cycles 0x01: 4 (= 2^2) cycles ⋮ 0x1E: 2,147,483,648 (= 2^{31}) cycles 0x1F: 4,294,967,296 (= 2^{32}) cycles	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
27:24	PERREP[3:0]	Periodic transaction repeat These bits configure the number of transaction repetitions in periodic manual-command mode. 0x0: 1 (= 2^0) time 0x1: 2 (= 2^1) times ⋮ 0xE: 16384 (= 2^{14}) times 0xF: 32768 (= 2^{15}) times	R/W
31:28	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to control xSPI Master function.

35.2.2.6 CDCTL1 : xSPI Command Manual Control Register 1

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x074

Bit position:	31																												0
Bit field:	PEREXP[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PEREXP[31:0]	Periodic transaction expected value These bits configure the expected value to compare with the read value in periodic manual-command mode. For example, in case of comparing 1 byte, the lower byte should be configured.	R/W

This register has functions to control xSPI Master function.

35.2.2.7 CDCTL2 : xSPI Command Manual Control Register 2

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x078

Bit position:	31																												0
Bit field:	PERMSK[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	PERMSK[31:0]	Periodic transaction masked value These bits configure the masked value for the expected value in periodic manual-command mode. When set 1 to any bit, the corresponding bit configured as expected value (CDCTL1.PEREXP[31:0]) is ignored. In 8D-8D-8D, the data bytes are transferred only in byte pairs on SPI bus. It means the dummy read data could be stored. It should be masked for unused bits. For example, in case of read lower 1 byte, it should be configured to 0xFFFFFF00.	R/W

This register has functions to control xSPI Master function.

35.2.2.8 CDTBUFn : xSPI Command Manual Type Buf n (n = 0 to 3)

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x080 + 0x010 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CMD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	TRTYPE	—	LATE[4:0]					DATASIZE[3:0]				ADDSIZE[2:0]			CMDSIZE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CMDSIZE[1:0]	Command Size These bits configure the size of command field. In case of 8D-8D-8D, it should be fixed to 10b. It should not be configured both command size and address size to zero. 0 0: 0 bytes (No command phase) 0 1: 1 byte 1 0: 2 bytes Others: Setting prohibited	R/W
4:2	ADDSIZE[2:0]	Address size These bits configure the size of address field. 0 0 0: 0 bytes (No address phase) 0 0 1: 1 byte 0 1 0: 2 bytes 0 1 1: 3 bytes 1 0 0: 4 bytes Others: Setting prohibited	R/W
8:5	DATASIZE[3:0]	Write/Read Data Size These bits configure the size of data field. In 8D-8D-8D, the data bytes are transferred only in byte pairs on SPI bus. For example, even if configuring 1 byte for read, 2 bytes data is received. The last byte should be ignored. The 0 bytes must not configured for read transaction. 0x0: 0 bytes (No data phase) 0x1: 1 byte ⋮ 0x7: 7 bytes 0x8: 8 bytes Others: Setting prohibited	R/W
13:9	LATE[4:0]	Latency cycle These bits configure the latency cycle in manual-command mode. 0x0: No latency 0x1: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
14	—	This bit is read as 0. The write value should be 0.	R/W
15	TRTYPE	Transaction Type This bit selects the type of transaction. 0: Read transaction (Readout data from slave device) 1: Not read transaction	R/W

Bit	Symbol	Function	R/W
31:16	CMD[15:0]	Command (1-2 bytes) These bits configure the command field in manual-command mode. The number of bytes configured in Command Size bit is transferred. 1S-1S-1S, 4S-4D-4D: CMD[15:8] is command field, CMD[7:0] is not used. 8D-8D-8D profile 1.0: CMD[15:8] is command field, CMD[7:0] is extension field. 8D-8D-8D profile 2.0: CMD[15:0] is upper 2 bytes of command & modifier field. (bit 47-32 in xSPI protocol)	R/W

This register has functions to control xSPI Master function.

35.2.2.9 CDABUFn : xSPI Command Manual Address Buf n (n = 0 to 3)

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x084 + 0x10 × n

Bit position:	31																												0
Bit field:	ADD[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	ADD[31:0]	Address These bits configure the address field in manual-command mode. 1S-1S-1S, 4S-4D-4D, 8D-8D-8D profile 1.0: It is address field. 8D-8D-8D profile 2.0: It is lower 4 bytes of command & modifier field. (bit 31-0 in xSPI protocol)	R/W

This register has functions to control xSPI Master function.

35.2.2.10 CDD0BUFn : xSPI Command Manual Data 0 Buf n (n = 0 to 3)

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x088 + 0x10 × n

Bit position:	31																												0
Bit field:	DATA[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

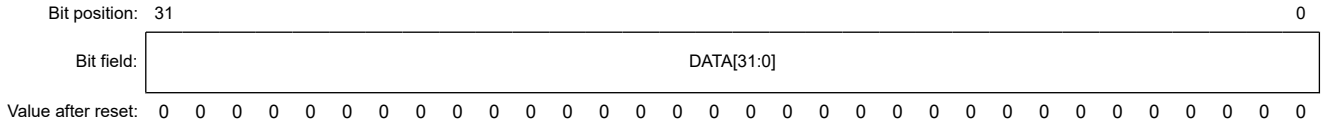
Bit	Symbol	Function	R/W
31:0	DATA[31:0]	Write/Read Data These bits configure the data field in manual-command mode. In case of write transaction, the write data should be configured. In case of read transaction, the read data is stored.	R/W

This register has functions to control xSPI Master function.

35.2.2.11 CDD1BUFn : xSPI Command Manual Data 1 Buf n (n = 0 to 3)

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x08C + 0x10 × n



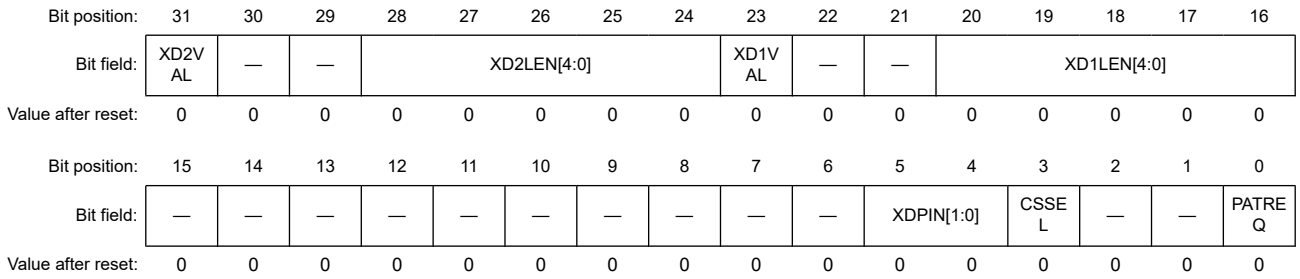
Bit	Symbol	Function	R/W
31:0	DATA[31:0]	Write/Read Data These bits configure the data field in manual-command mode. In case of write transaction, the write data should be configured. In case of read transaction, the read data is stored.	R/W

This register has functions to control xSPI Master function.

35.2.2.12 LPCTL0 : xSPI Link Pattern Control Register 0

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x100



Bit	Symbol	Function	R/W
0	PATREQ	Pattern request This bit requests to issue the pattern. When set to 1, it starts the pattern. It is cleared to 0 when the pattern completed. 0: No request XiP Disable pattern 1: Request XiP Disable pattern	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	CSSEL	Chip select This bit selects a target memory to issue a pattern. 0: slave0 (CS0) 1: slave1 (CS1)	R/W
5:4	XDPIN[1:0]	XiP Disable pattern pin These bits select the data output pins to transmit XiP Disable pattern. 0 0: 1 pin 0 1: 2 pins 1 0: 4 pins 1 1: 8 pins	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
20:16	XD1LEN[4:0]	XiP Disable pattern 1st phase length These bits select the length of 1st phase in XiP disable pattern. The pattern with zero-length both 1st phase and 2nd phase should not be configured. 0x0: 0 cycles 0x1: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
22:21	—	These bits are read as 0. The write value should be 0.	R/W
23	XD1VAL	XiP Disable pattern 1st phase value This bit selects the value of 1st phase in XiP disable pattern. 0: Low drive 1: High drive	R/W
28:24	XD2LEN[4:0]	XiP Disable pattern 2nd phase length These bits select the length of 2nd phase in XiP disable pattern. 0x00: 0 cycles 0x01: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
30:29	—	These bits are read as 0. The write value should be 0.	R/W
31	XD2VAL	XiP Disable pattern 2nd phase value This bit selects the value of 2nd phase in XiP disable pattern. 0: Low drive 1: High drive	R/W

This register has functions to control xSPI Master function.

35.2.2.13 LPCTL1 : xSPI Link Pattern Control Register 1

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x104

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	RSTSU[2:0]			—	RSTWID[2:0]			—	—	RSTREP[1:0]		CSSEL	—	PATREQ[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	PATREQ[1:0]	Pattern request These bits request to issue the pattern. When set to 01b or 10b, it starts the pattern. It is cleared to 00b when the pattern completed. 0 0: No request 0 1: Request Reset pattern 1 0: Request CS only pattern 1 1: Setting prohibited	R/W
2	—	This bit is read as 0. The write value should be 0.	R/W
3	CSSEL	Chip select This bit selects a target memory to issue a pattern. 0: slave0 (CS0) 1: slave1 (CS1)	R/W

Bit	Symbol	Function	R/W
5:4	RSTREP[1:0]	Reset pattern repeat These bits select the repeating time to toggle CS from LOW to HIGH. 0 0: 4 times (Specified on Reset Signaling Protocol) 0 1: 5 times 1 0: 6 times 1 1: 7 times	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
10:8	RSTWID[2:0]	Reset pattern width These bits configure the width of cycle in reset pattern and CS only pattern. It toggles CS with the configured cycle. 0 0 0: 2 (= 2^1) cycles 0 0 1: 4 (= 2^2) cycles ⋮ 1 1 0: 128 (= 2^7) cycles 1 1 1: 256 (= 2^8) cycles	R/W
11	—	This bit is read as 0. The write value should be 0.	R/W
14:12	RSTSU[2:0]	Reset pattern data output setup time These bits configure the number of setup cycle for data output based on the edge of CS in reset pattern. It needs enough setup time because xSPI slave samples any data at the rising edge of CS. This cycle of setup time should be less than the cycle of reset pattern width (RSTWID[2:0]). 0 0 0: 1 cycle 0 0 1: 2 cycles ⋮ 1 1 0: 7 cycles 1 1 1: 8 cycles	R/W
31:15	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to control xSPI Master function.

35.2.2.14 LIOCTL : xSPI Link I/O Control Register

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x108

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RSTCS1	RSTCS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	WPCS1	WPCS0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	WPCS0*1	WP drive for slave0 This bit controls the value of Write Protect (WP0#) port. It can be useful only for xSPI slave with write protect port. 0: Drive Low level 1: Drive High level	R/W
1	WPCS1*1	WP drive for slave1 This function is same as WPCS0. 0: Drive Low level 1: Drive High level	R/W
15:2	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
16	RSTCS0*1	Reset drive for slave0 This bit controls the value of xSPI reset (RESET0#) port. It can be useful only for xSPI slave with reset port. 0: Drive Low level 1: Drive High level	R/W
17	RSTCS1*1	Reset drive for slave1 This function is same as RSTCS0. 0: Drive Low level 1: Drive High level	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. For unit 1, these bits are reserved. The write value should be 0.

This register has functions to control xSPI Master function.

35.2.2.15 CCCTL0CSn : xSPI Command Calibration Control Register 0 CSn (n = 0, 1)

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x130 + 0x020 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	CASFTEND[4:0]					—	—	—	CASFTSTA[4:0]				
Value after reset:	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	CAITV[4:0]					—	—	—	—	—	—	CANOWR	CAEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CAEN	Automatic Calibration Enable This bit enables the automatic calibration. When set to 1, it transmits the calibration sequence periodically and adjusts the value of phase shift. When set to 0 during the calibration sequence, it stops after completed ongoing calibration sequence, and then this bit is cleared. 0: Disable automatic calibration 1: Enable automatic calibration	R/W
1	CANOWR	Calibration no write mode This bit selects to omit write command in calibration sequence. It can be used for any slave device with fixed calibration pattern data. 0: Calibration sequence with write command 1: Calibration sequence without write command	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
12:8	CAITV[4:0]	Calibration interval These bits configure the interval between calibration patterns. 0x00: 2 (= 2^1) cycle wait 0x01: 4 (= 2^2) cycle wait ⋮ 0x1E: 2,147,483,648 (= 2^{31}) cycle wait 0x1F: 4,294,967,296 (= 2^{32}) cycle wait	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
20:16	CASFTSTA[4:0]	Calibration DS shift start value These bits configure the start value of DS shift.	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W
28:24	CASFTEND[4:0]	Calibration DS shift end value These bits configure the end value of DS shift. It should be equal or more than the start value (CASFTSTA).	R/W

Bit	Symbol	Function	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to control xSPI Master function.

35.2.2.16 CCCTL1CSn : xSPI Command Calibration Control Register 1 CSn (n = 0, 1)

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x134 + 0x020 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	CARDLATE[4:0]					—	—	—	CAWRLATE[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CADATASIZE[3:0]				CAADDSIZE[2:0]			CACMDSIZE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CACMDSIZE[1:0]	Command Size These bits configure the size of command field. In case of 8D-8D-8D, it should be fixed to 10b. It should not be configured both command size and address size to zero. 0 0: 0 bytes (No command phase) 0 1: 1 byte 1 0: 2 bytes 1 1: Setting prohibited	R/W
4:2	CAADDSIZE[2:0]	Address size These bits configure the size of address field. 0 0 0: 0 bytes (No address phase) 0 0 1: 1 byte 0 1 0: 2 bytes 0 1 1: 3 bytes 1 0 0: 4 bytes Others: Setting prohibited	R/W
8:5	CADATASIZE[3:0]	Write/Read Data Size These bits configure the size of data field. In 8D-8D-8D, it should be configured with even byte. 0x0: 1 byte 0x1: 2 bytes ⋮ 0xE: 15 bytes 0xF: 16 bytes	R/W
15:9	—	These bits are read as 0. The write value should be 0.	R/W
20:16	CAWRLATE[4:0]	Write Latency cycle These bits configure the latency cycle in calibration frame. 0x00: No latency 0x01: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
23:21	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
28:24	CARDLATE[4:0]	Read Latency cycle These bits configure the latency cycle in calibration frame. 0x00: No latency 0x01: 1 cycle ⋮ 0x1E: 30 cycles 0x1F: 31 cycles	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

35.2.2.17 CCCTL2CSn : xSPI Command Calibration Control Register 2 CSn (n = 0, 1)

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x138 + 0x020 × n

Bit position:	31															16	15														0
Bit field:	CARDCMD[15:0]																CAWRCMD[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CAWRCMD[15:0]	Calibration pattern write command These bits configure the calibration pattern write command.	R/W
31:16	CARDCMD[15:0]	Calibration pattern read command These bits configure the calibration pattern read command.	R/W

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

35.2.2.18 CCCTL3CSn : xSPI Command Calibration Control Register 3 CSn (n = 0, 1)

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x13C + 0x020 × n

Bit position:	31																														0
Bit field:	CAADD[31:0]																														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

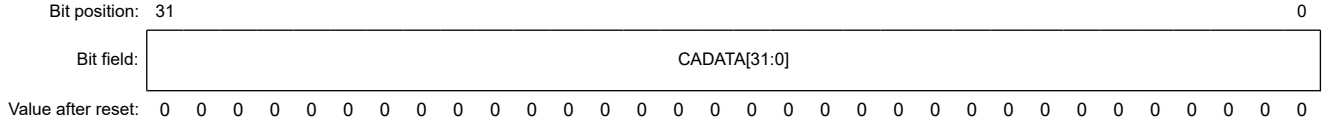
Bit	Symbol	Function	R/W
31:0	CAADD[31:0]	Calibration pattern address These bits configure the calibration pattern address.	R/W

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

35.2.2.19 CCCTL4CSn : xSPI Command Calibration Control Register 4 CSn (n = 0, 1)

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x140 + 0x020 × n



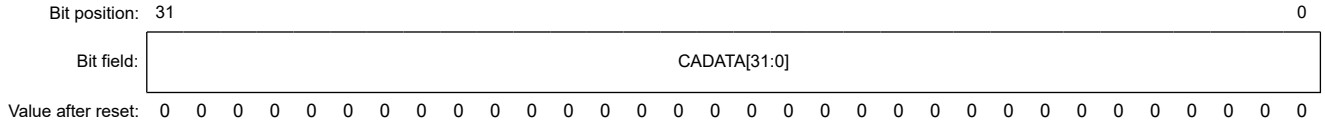
Bit	Symbol	Function	R/W
31:0	CADATA[31:0]	Calibration pattern data These bits configure the calibration pattern data.	R/W

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

35.2.2.20 CCCTL5CSn : xSPI Command Calibration Control Register 5 CSn (n = 0, 1)

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x144 + 0x020 × n



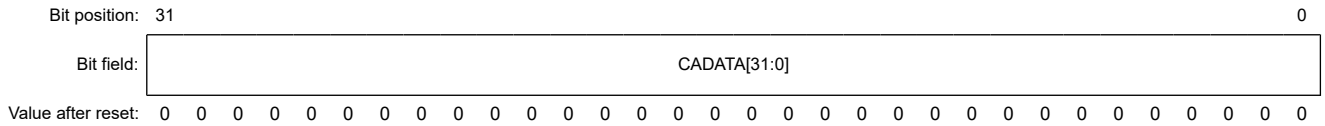
Bit	Symbol	Function	R/W
31:0	CADATA[31:0]	Calibration pattern data These bits configure the calibration pattern data.	R/W

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

35.2.2.21 CCCTL6CSn : xSPI Command Calibration Control Register 6 CSn (n = 0, 1)

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x148 + 0x020 × n



Bit	Symbol	Function	R/W
31:0	CADATA[31:0]	Calibration pattern data These bits configure the calibration pattern data.	R/W

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

35.2.2.22 CCCTL7CSn : xSPI Command Calibration Control Register 7 CSn (n = 0, 1)

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x14C + 0x020 × n

Bit position:	31																												0
Bit field:	CADATA[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CADATA[31:0]	Calibration pattern data These bits configure the calibration pattern data.	R/W

This register has functions to control xSPI Master function. It should be updated while Automatic Calibration Enable is disabled.

35.2.3 xSPI Status Registers

These registers monitor the status of xSPI Master.

35.2.3.1 VERSTT : xSPI Version Register

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x180

Bit position:	31																												0
Bit field:	VER[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	VER[31:0]	Version These bits indicate this IP version.	R

This register indicates the status of xSPI Master.

35.2.3.2 COMSTT : xSPI Common Status Register

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x184

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	RSTOCS1	INTCS1	ECSCS1	—	RSTOCS0	INTCS0	ECSCS0
Value after reset:	0	0	0	0	0	0	0	0	0	*1	*1	*1	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	WRBUFNE	—	PBUFNE	—	—	—	MEMACC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MEMACC	Memory access ongoing 0: AHB channel is not accessing to memory. 1: AHB channel is accessing to memory.	R
3:1	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
4	PBUFNE	Prefetch Buffer Not Empty 0: Empty 1: Not empty	R
5	—	This bit is read as 0.	R
6	WRBUFNE	Write Buffer Not Empty 0: Empty 1: Not empty	R
15:7	—	These bits are read as 0.	R
16	ECSCS0*2	ECS monitor for slave0 This bit indicates the value of ECS port. 0: Low level 1: High level	R
17	INTCS0*2	INT monitor for slave0 This bit indicates the value of INT port. 0: Low level 1: High level	R
18	RSTOCS0*2	RSTO monitor for slave0 This bit indicates the value of RSTO port. 0: Low level 1: High level	R
19	—	This bit is read as 0.	R
20	ECSCS1*2	ECS monitor for slave1 This function is same as ECSCS0. 0: Low level 1: High level	R
21	INTCS1*2	INT monitor for slave1 This function is same as INTCS0. 0: Low level 1: High level	R
22	RSTOCS1*2	RSTO monitor for slave1 This function is same as RSTOCS0. 0: Low level 1: High level	R
31:23	—	These bits are read as 0.	R

Note 1. Value after reset is 1 for unit 0 and 0 for unit 1.
Note 2. These bits are reserved for unit 1.

This register indicates the status of xSPI Master.

35.2.3.3 CASTTCSn : xSPI Calibration Status Register CSn (n = 0, 1)

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x188 + 0x004 × n

Bit position:	31																												0
Bit field:	CASUC[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	CASUC[31:0]	Calibration Success These bits indicate the calibration success for each DS shift value. It is updated when completed each calibration sequence. CASUC[x] indicates calibration success in DS shift value = x.	R

This register indicates the status of xSPI Master.

35.2.4 xSPI Interrupt Registers

These registers control the interrupt function of xSPI Master.

35.2.4.1 INTS : xSPI Interrupt Status Register

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x190

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CASUCCS1	CASUCCS0	CAFAILCS1	CAFAILCS0	—	—	—	—	—	—	—	BUSERR	—	BRGUF	—	BRGOF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	INTCS1	INTCS0	—	—	ECSCS1	ECSCS0	—	—	DSTOCS1	DSTOCS0	PERTO	INICMP	PATCMP	CMDCMP
Value after reset:	0	0	*1	*1	0	0	*1	*1	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMDCMP	Command Completed This bit is set to 1 when completed the requested manual-command. In direct manual-command, it means all transactions completed. In periodic manual-command, it means the read data matches with the expected data. 0: No detection 1: Detection	R
1	PATCMP	Pattern Completed This bit is set to 1 when completed the requested pattern. 0: No detection 1: Detection	R
2	INICMP	Initial Sequence Completed This bit is set to 1 when completed the initial sequence. 0: No detection 1: Detection	R
3	PERTO	Periodic transaction timeout This bit is set to 1 when the read value does not match with the expected value in periodic manual-command mode. 0: No detection 1: Detection	R
4	DSTOCS0	DS timeout for slave0 This bit is set to 1 when lost DS in read transaction with using DS. It means not receiving the data during expected read phase. In this case, xSPI master stops the read transaction and the following transaction. 0: No detection 1: Detection	R
5	DSTOCS1	DS timeout for slave1 This function is same as DSTOCS0. 0: No detection 1: Detection	R
7:6	—	These bits are read as 0.	R
8	ECSCS0*2	ECC error detection for slave0 This bit is set to 1 when detected the falling edge on ECS0# port. It can be useful only for xSPI slave with ECC detection function. 0: No detection 1: Detection	R
9	ECSCS1*2	ECC error detection for slave1 This function is same as ECSCS0. 0: No detection 1: Detection	R
11:10	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
12	INTCS0*2	Interrupt detection for slave0 This bit is set to 1 when detected the falling edge on INT0# port. It can be useful only for xSPI slave with interrupt function. 0: No detection 1: Detection	R
13	INTCS1*2	Interrupt detection for slave1 This function is same as INTCS0. 0: No detection 1: Detection	R
15:14	—	These bits are read as 0.	R
16	BRGOF	Bridge Buffer overflow This bit is read as 0.	R
17	—	This bit is read as 0.	R
18	BRGUF	Bridge Buffer underflow This bit is read as 0.	R
19	—	This bit is read as 0.	R
20	BUSERR	AHB bus error This bit is set to 1 when an error response occurs on AHB channel.	R
27:21	—	These bits are read as 0.	R
28	CAFAILCS0	Calibration failed for slave0 This bit is set to 1 when failed calibration. 0: No detection 1: Detection	R
29	CAFAILCS1	Calibration failed for slave1 This function is same as CAFAILCS0. 0: No detection 1: Detection	R
30	CASUCCS0	Calibration success for slave0 This bit is set to 1 when success calibration. 0: No detection 1: Detection	R
31	CASUCCS1	Calibration success for slave1 This function is same as CASUCCS0. 0: No detection 1: Detection	R

Note 1. Value after reset is 0 for unit 0 and 1 for unit 1.
Note 2. These bits are reserved for unit 1.

This register indicates the status of interrupt. The bits in this register are cleared to 0 when writing 1 on the corresponding bit of INTC register.

35.2.4.2 INTC : xSPI Interrupt Clear Register

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x194

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CASUCCS1C	CASUCCS0C	CAFAILCS1C	CAFAILCS0C	—	—	—	—	—	—	—	BUSERRC	—	BRGUFC	—	BRGOFC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	INTCS1C	INTCS0C	—	—	ECSCS1C	ECSCS0C	—	—	DSTOCS1C	DSTOCS0C	PERTOC	INICMPC	PATCMPC	CMDCMPC
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMDCMPC	Command Completed interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
1	PATCMPC	Pattern Completed interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
2	INICMPC	Initial Sequence Completed interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
3	PERTOC	Periodic transaction timeout interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
4	DSTOCS0C	DS timeout for slave0 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
5	DSTOCS1C	DS timeout for slave1 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
7:6	—	The write value should be 0.	W
8	ECSCS0C*1	ECC error detection for slave0 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
9	ECSCS1C*1	ECC error detection for slave1 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
11:10	—	The write value should be 0.	W
12	INTCS0C*1	Interrupt detection for slave0 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
13	INTCS1C*1	Interrupt detection for slave1 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
15:14	—	The write value should be 0.	W
16	BRGOFC	Bridge Buffer overflow interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
17	—	The write value should be 0.	W
18	BRGUFC	Bridge Buffer underflow interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
19	—	The write value should be 0.	W
20	BUSERRC	AHB bus error interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
27:21	—	The write value should be 0.	W
28	CAFAILCS0C	Calibration failed for slave0 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
29	CAFAILCS1C	Calibration failed for slave1 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W
30	CASUCCS0C	Calibration success for slave0 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W

Bit	Symbol	Function	R/W
31	CASUCCS1C	Calibration success for slave1 interrupt clear 0: No change interrupt status 1: Clear interrupt status	W

Note 1. For unit 1, these bits are reserved. The write value should be 1.

This register clears the status of interrupt.

35.2.4.3 INTE : xSPI Interrupt Enable Register

Base address: XSPIm = 0x8022_0000 + 0x1000 × m (m = 0, 1)

Offset address: 0x198

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CASUCCS1E	CASUCCS0E	CAFAILCS1E	CAFAILCS0E	—	—	—	—	—	—	—	BUSERRE	—	BRGUFE	—	BRGOFE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	INTCS1E	INTCS0E	—	—	ECSCS1E	ECSCS0E	—	—	DSTOCS1E	DSTOCS0E	PERTOE	INICMPE	PATCMPE	CMDCMPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMDCMPE	Command Completed interrupt enable 0: Disabled 1: Enabled	R/W
1	PATCMPE	Pattern Completed interrupt enable 0: Disabled 1: Enabled	R/W
2	INICMPE	Initial Sequence Completed interrupt enable 0: Disabled 1: Enabled	R/W
3	PERTOE	Periodic transaction timeout interrupt enable 0: Disabled 1: Enabled	R/W
4	DSTOCS0E	DS timeout for slave0 interrupt enable 0: Disabled 1: Enabled	R/W
5	DSTOCS1E	DS timeout for slave1 interrupt enable 0: Disabled 1: Enabled	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	ECSCS0E*1	ECC error detection for slave0 interrupt enable 0: Disabled 1: Enabled	R/W
9	ECSCS1E*1	ECC error detection for slave1 interrupt enable 0: Disabled 1: Enabled	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	INTCS0E*1	Interrupt detection for slave0 interrupt enable 0: Disabled 1: Enabled	R/W
13	INTCS1E*1	Interrupt detection for slave1 interrupt enable 0: Disabled 1: Enabled	R/W

Bit	Symbol	Function	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	BRGOFE	Bridge Buffer overflow interrupt enable 0: Disabled 1: Enabled	R/W
17	—	This bit is read as 0. The write value should be 0.	R/W
18	BRGUFE	Bridge Buffer underflow interrupt enable 0: Disabled 1: Enabled	R/W
19	—	This bit is read as 0. The write value should be 0.	R/W
20	BUSERRE	AHB bus error interrupt enable 0: Disabled 1: Enabled	R/W
27:21	—	These bits are read as 0. The write value should be 0.	R/W
28	CAFAILCS0E	Calibration failed for slave0 interrupt enable 0: Disabled 1: Enabled	R/W
29	CAFAILCS1E	Calibration failed for slave1 interrupt enable 0: Disabled 1: Enabled	R/W
30	CASUCCS0E	Calibration success for slave0 interrupt enable 0: Disabled 1: Enabled	R/W
31	CASUCCS1E	Calibration success for slave1 interrupt enable 0: Disabled 1: Enabled	R/W

Note 1. For unit 1, these bits are reserved. The write value should be 0.

This register enables the interrupt.

35.3 Operation

xSPI Master interface has the functions to issue the transaction for external memory with xSPI Slave interface. It allows to write to registers in external memory or read from it.

This xSPI Master mainly has two modes to issue the transaction. One is a manual-command mode; Software configures all fields of xSPI frame and starts the transaction by software request. The other is a memory-mapping mode; it automatically converts AHB access for pre-configured memory area into xSPI transaction. It enables to access from AHB to external memory area outside of chip via xSPI bus.

This section describes the xSPI bus operation, the direct control of xSPI frame (manual-command), the control of memory access (memory-mapping), the error operation, and the flow to operation.

35.3.1 xSPI Bus

This section describes the xSPI bus operation.

35.3.1.1 Supported Protocol Mode

This xSPI Master supports various protocol modes. It is configured by Protocol mode bits (LIOCFGCSn.PRTMD[9:0]). Table 35.4 lists the supported protocol mode.

Table 35.4 Supported protocol mode (1 of 2)

Protocol mode	Function	PRTMD[9:0]	Note
1S-1S-1S	Command, Address, and Data fields are transferred at SDR with using 1 pin. Read data is sampled with CK.	0x000	Specified by xSPI protocol

Table 35.4 Supported protocol mode (2 of 2)

Protocol mode	Function	PRTMD[9:0]	Note
4S-4D-4D	Command field is transferred at SDR with using 4 pins. Address and Data fields are transferred at DDR with using 4 pins. Read data is sampled with DS.	0x3B2	Specified by xSPI protocol
8D-8D-8D	Command, Address, and Data fields are transferred at DDR with using 8 pins. Read data is sampled with DS.	0x3FF	Specified by xSPI protocol
1S-2S-2S	Command field is transferred at SDR with using 1 pin. Address and Data fields are transferred at SDR with using 2 pins. Read data is sampled with CK.	0x048	—
2S-2S-2S	Command, Address, and Data fields are transferred at SDR with using 2 pins. Read data is sampled with CK.	0x049	—
1S-4S-4S	Command field is transferred at SDR with using 1 pin. Address and Data fields are transferred at SDR with using 4 pins. Read data is sampled with CK.	0x090	—
4S-4S-4S	Command, Address, and Data fields are transferred at SDR with using 4 pins. Read data is sampled with CK.	0x092	—

Note: In case of XiP mode enable, XiP code is inserted in Latency field. It is valid only for memory-mapping mode.

Table 35.5 lists a description of the main internal signals.

Table 35.5 Internal signals

Signal name	I/O	Function
clk_spi	I	xSPI control clock
spi_ck	O	xSPI Clock (Same as XSPIm_CKP pin)
spi_cs0	O	xSPI Chip Select for slave0 (Same as XSPIm_CS0# pin)
spi_cs1	O	xSPI Chip Select for slave1 (Same as XSPIm_CS1# pin)
spi_doe[7:0]	O	xSPI Data Output Enable
spi_do[7:0]	O	xSPI Data Output
spi_di[7:0]	I	xSPI Data Input
spi_dsoe	O	xSPI Data Strobe Output Enable
spi_dso	O	xSPI Data Strobe Output
spi_dsi	I	xSPI Data Strobe Input

The bytes of Command and Address fields are transferred in highest order to lowest order sequence. The sequential bytes of Data field are transferred in lowest address to highest address order. In case of using multiple pins, the least significant bit of each byte is placed on spi_do/di[0] with each higher order bit on the successively higher numbered spi_do/di signals.

Figure 35.2 shows timing-chart for 1S-1S-1S protocol mode. The spi_do[0] signal is used for output data and the spi_di[1] signal is used for input data.

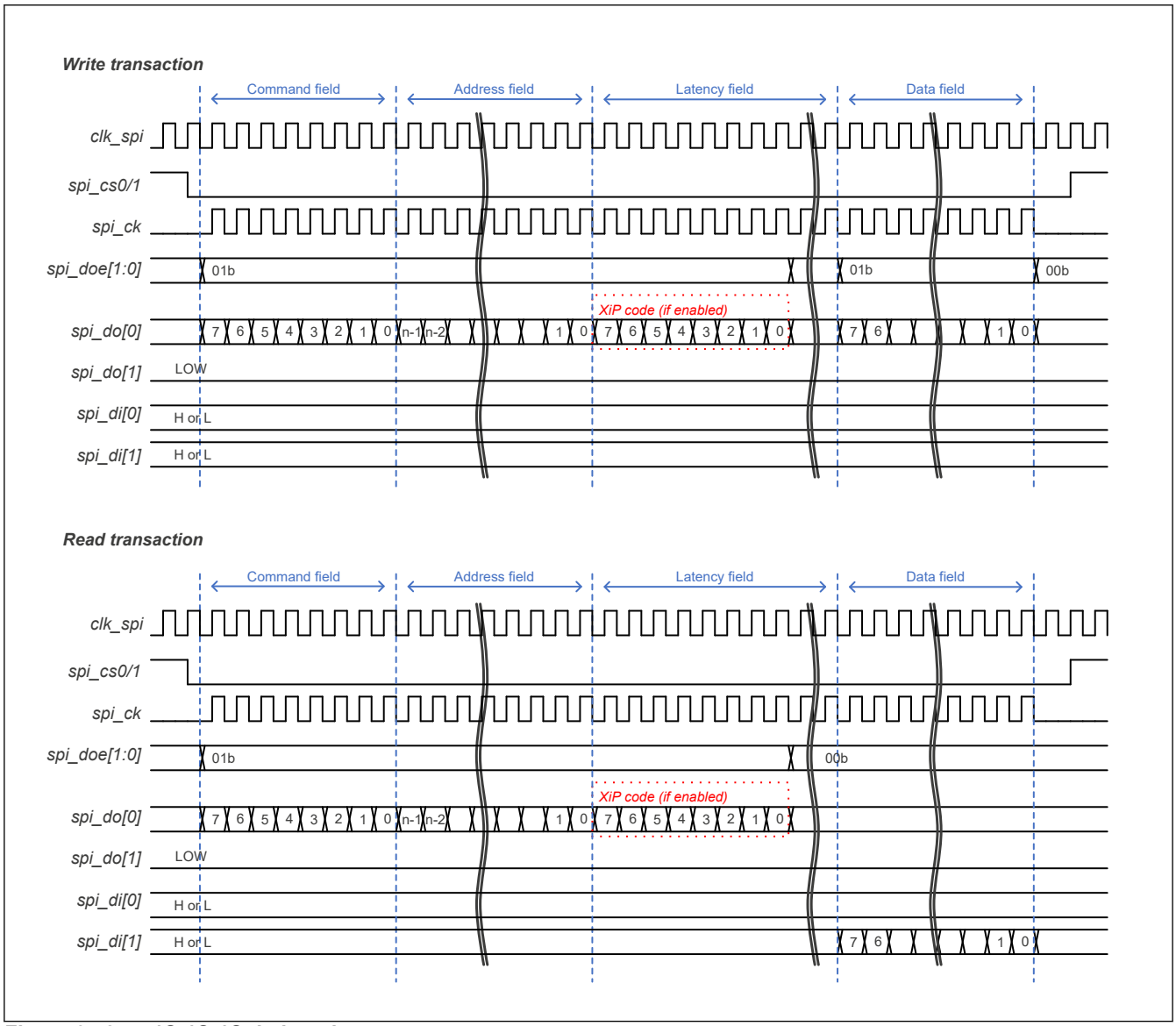


Figure 35.2 1S-1S-1S timing-chart

Figure 35.3 shows timing-chart for 1S-2S-2S protocol mode.

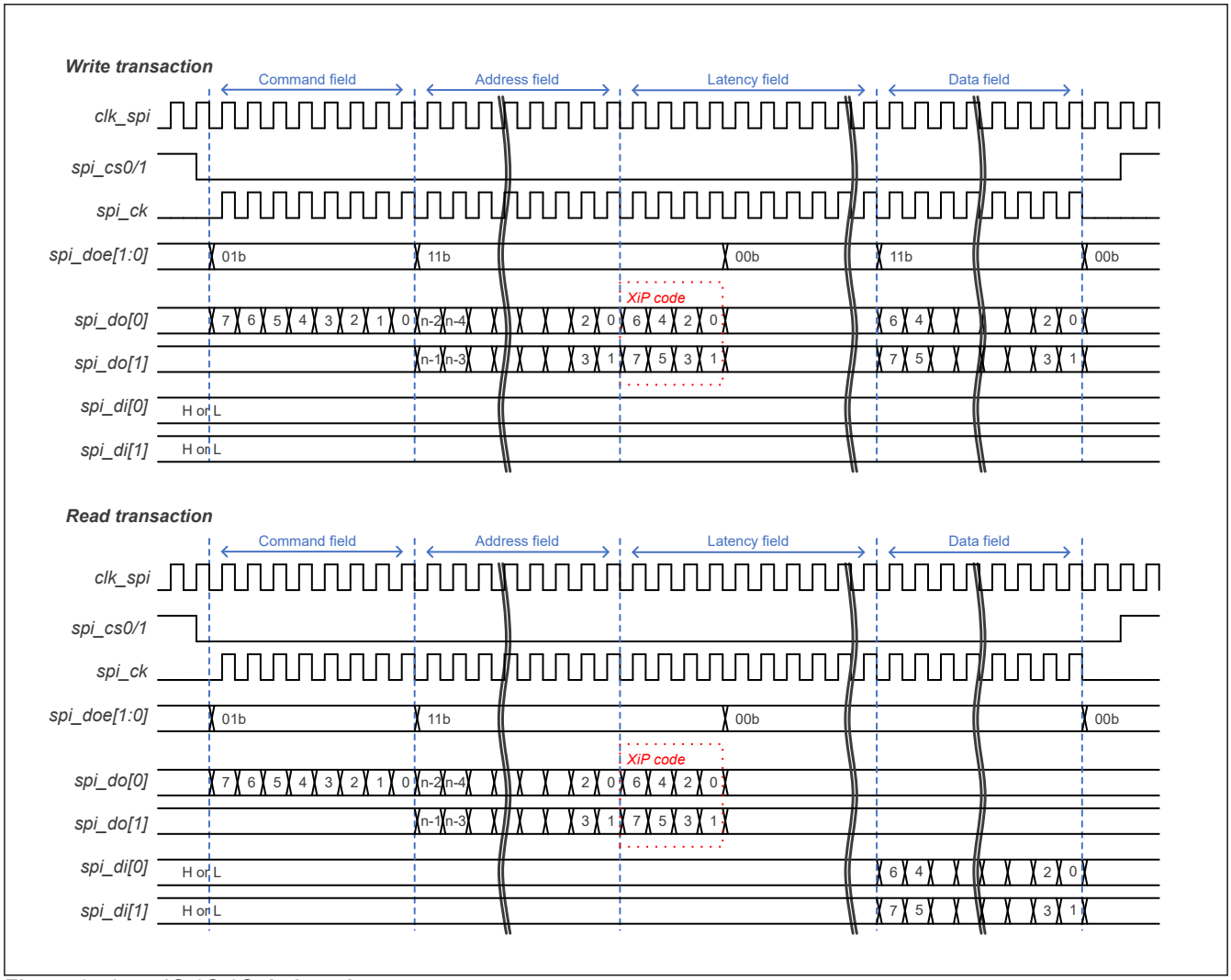


Figure 35.3 1S-2S-2S timing-chart

Figure 35.4 shows timing-chart for 4S-4D-4D protocol mode.

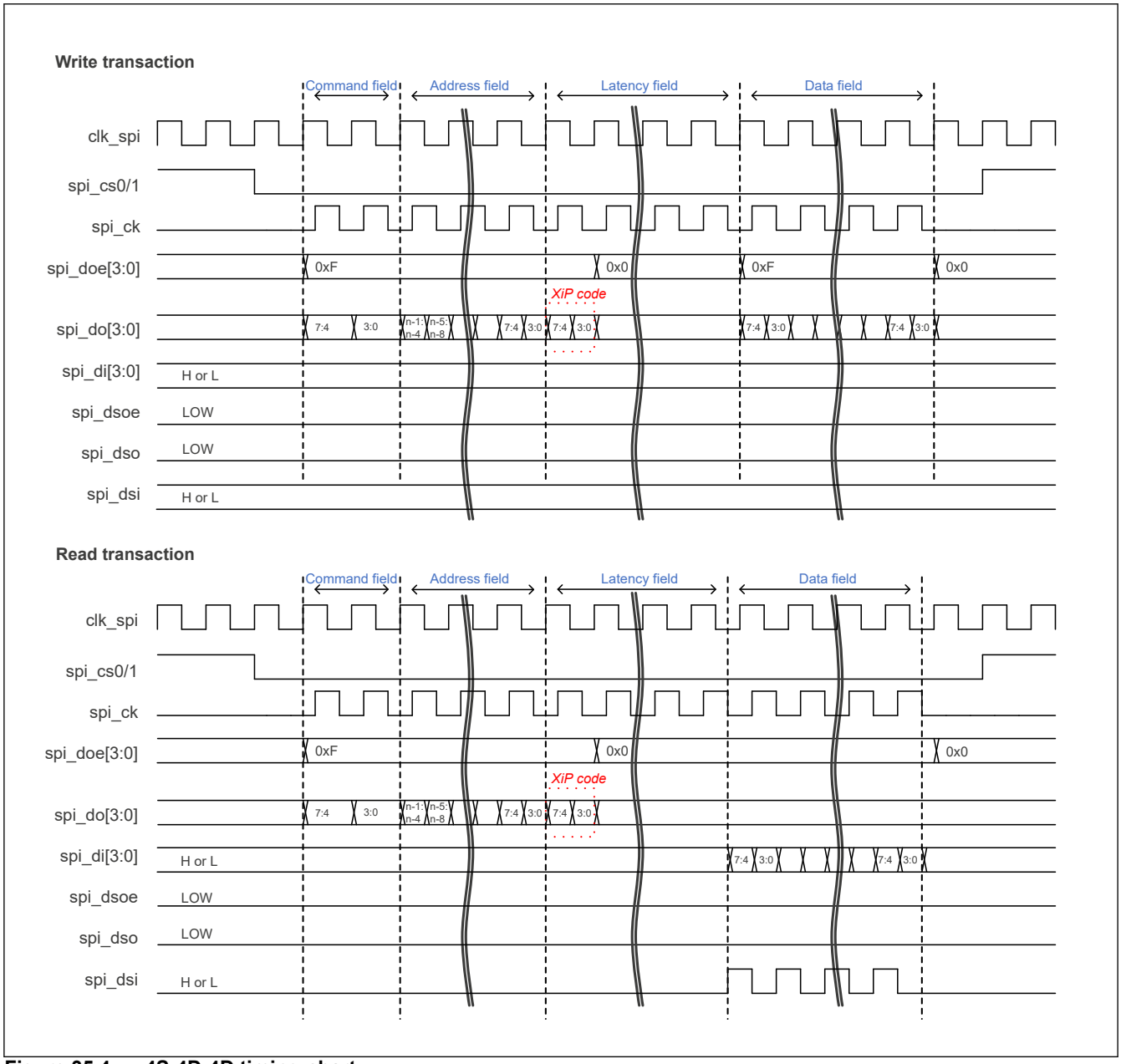


Figure 35.4 4S-4D-4D timing-chart

Figure 35.5 shows timing-chart for 8D-8D-8D profile 1.0 protocol mode. In 8D-8D-8D, the data is always transmitted with byte-pair. In case each field is odd byte, last one byte is padded with invalid data.

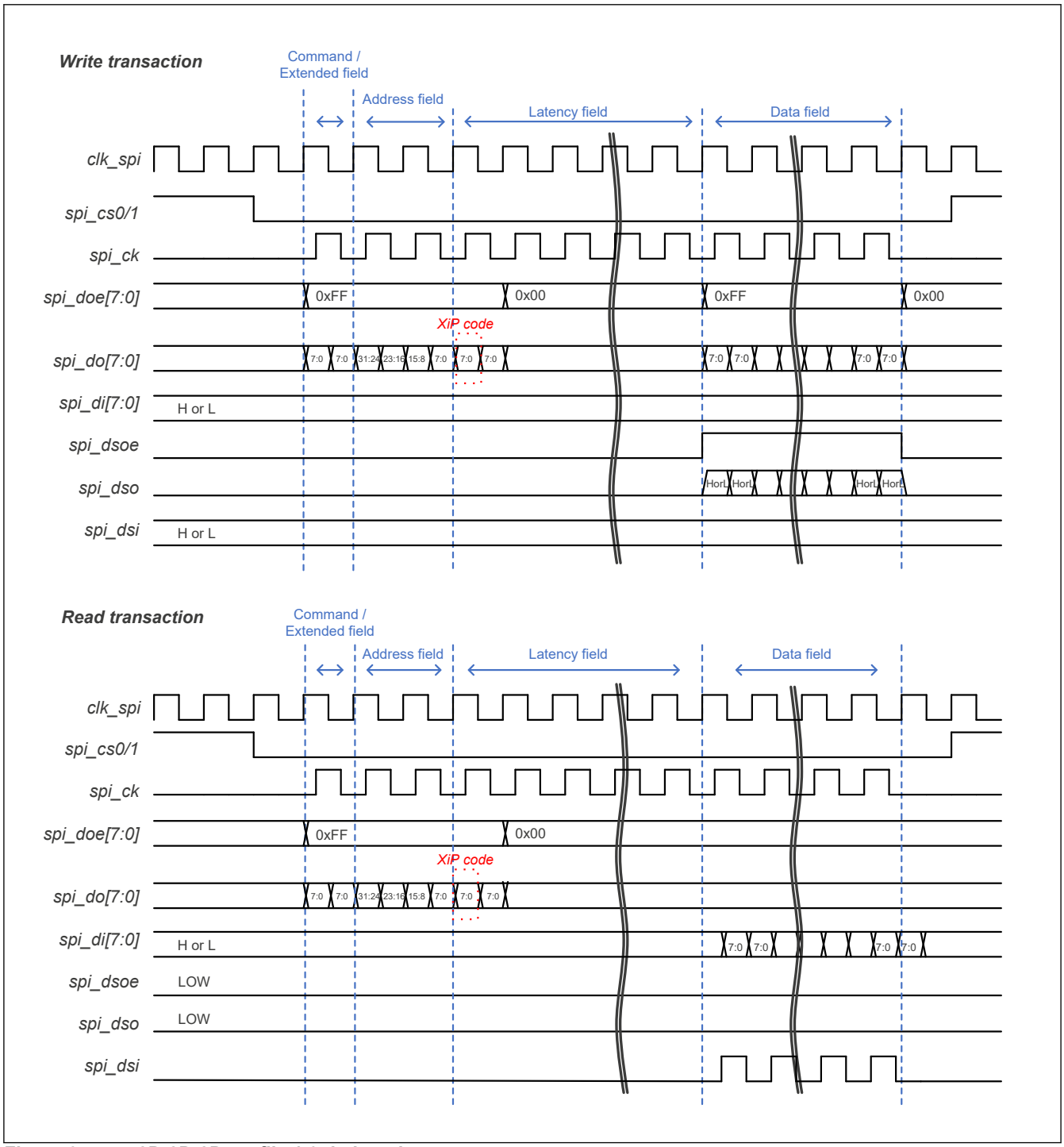


Figure 35.5 8D-8D-8D profile 1.0 timing-chart

Figure 35.6 shows timing-chart for 8D-8D-8D profile 2.0 protocol mode. In 8D-8D-8D, the data is always transmitted with byte-pair. In case each field is odd byte, last one byte is padded with invalid data.

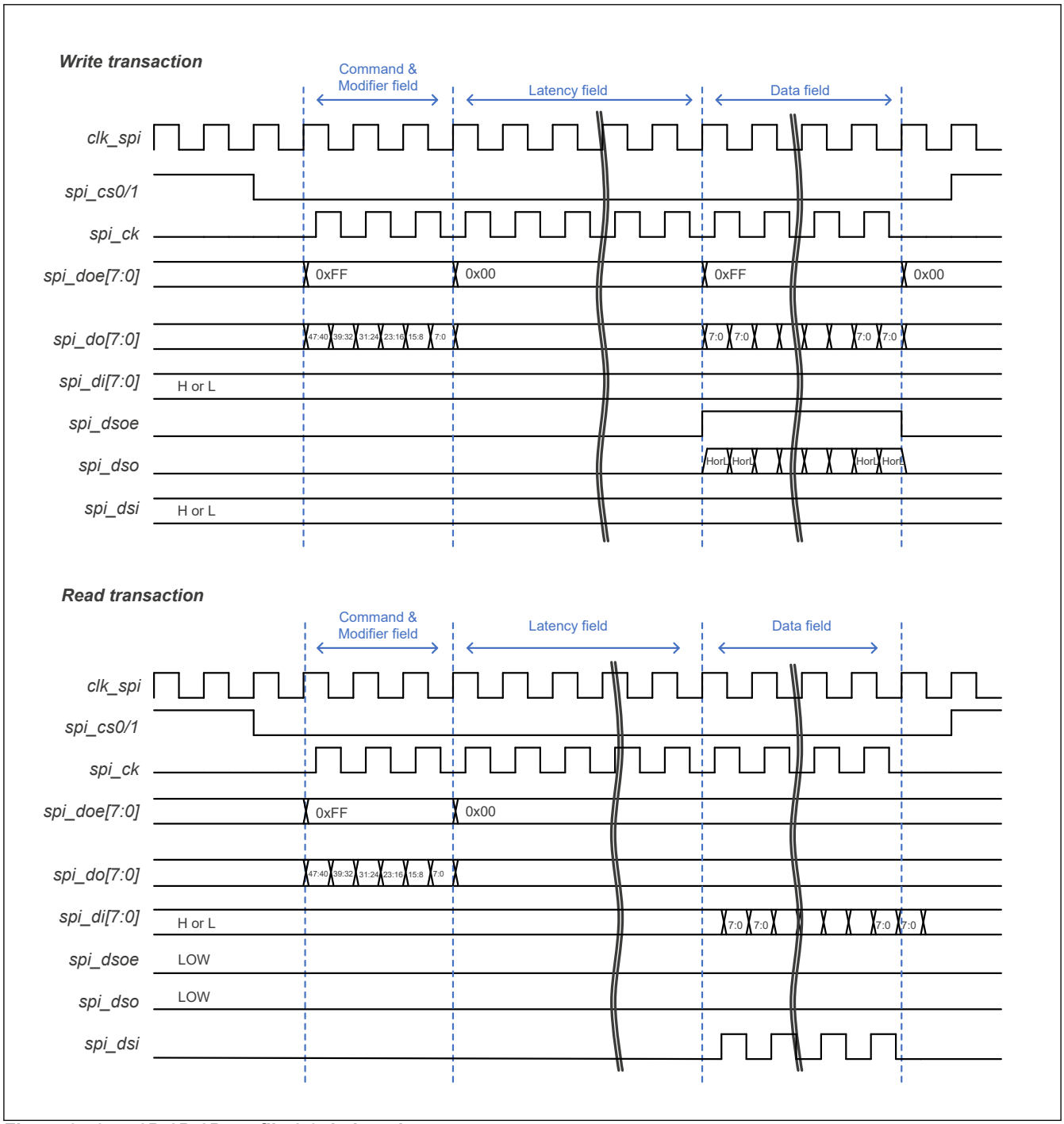


Figure 35.6 8D-8D-8D profile 2.0 timing-chart

35.3.1.2 xSPI Frame Interval

The interval between xSPI frames is configured with CS minimum idle term bits (LIOCFGCSn.CSMIN[3:0]). It depends on the specification of xSPI slave device.

35.3.1.3 xSPI Signals Timing Control

This xSPI Master supports both SDR and DDR. And it is possible to sample input data with Data-Strobe (DS) signal at SDR. For various modes and easy implementation, this xSPI Master could adjust the timing to drive/sample xSPI interface signals statically. Table 35.6 lists the summary of xSPI Interface signal timing control.

Table 35.6 Summary of xSPI signals timing control

Signal	Mode	Default operation	Timing Control (n = 0, 1)
CS (spi_cs) drive	—	Asserting 1 cycle before the rising-edge of first spi_ck	1 cycle extension for asserting with LIOCFGCSn.CSASTEX bit
		Negating 1.5 cycle after the falling-edge of last spi_ck	1 cycle extension for negating with LIOCFGCSn.CSNEGEX bit
CK (spi_ck) drive	SDR without DS SDR with DS	Reference point	—
	DDR with DS	Reference point	—
DO/DOE (spi_do/doe) drive	SDR without DS SDR with DS	Falling edge of clk_spi	0 or 0.5 cycle shift with LIOCFGCSn.SDRDRV bit
	DDR with DS	Both edges of clk_spi	—
DI (spi_di) sample	SDR without DS	Falling edge of spi_ck on expected data size	0 to 7 cycle shift (1 cycle unit) with LIOCFGCSn.SDRSMPSFT[3:0] bits 0 or 0.5 cycle shift with LIOCFGCSn.SDRSMPMD bit
	SDR with DS	Falling edge of spi_dsi signal on expected data size	Sample at rising edge with LIOCFGCSn.SDRSMPMD bit 0 to 1 cycle phase shift with WRAPCFG.DSSFTCSn[4:0] bits 0 to 7 cycle extension with LIOCFGCSn.DDRSMPEX[3:0] bits
	DDR with DS	Both edges of spi_dsi signal on expected data size	0 to 1 cycle phase shift with WRAPCFG.DSSFTCSn[4:0] bits 0 to 7 cycle extension with LIOCFGCSn.DDRSMPEX[3:0] bits

Note: In DDR on xSPI protocol, CK or DS should be aligned for center of data. It means to shift the phase by 0.25 cycle (90 degrees). This xSPI master supports to adjust this phase depending on the usage conditions.

Figure 35.7 shows the default operation and timing control for SDR without DS.

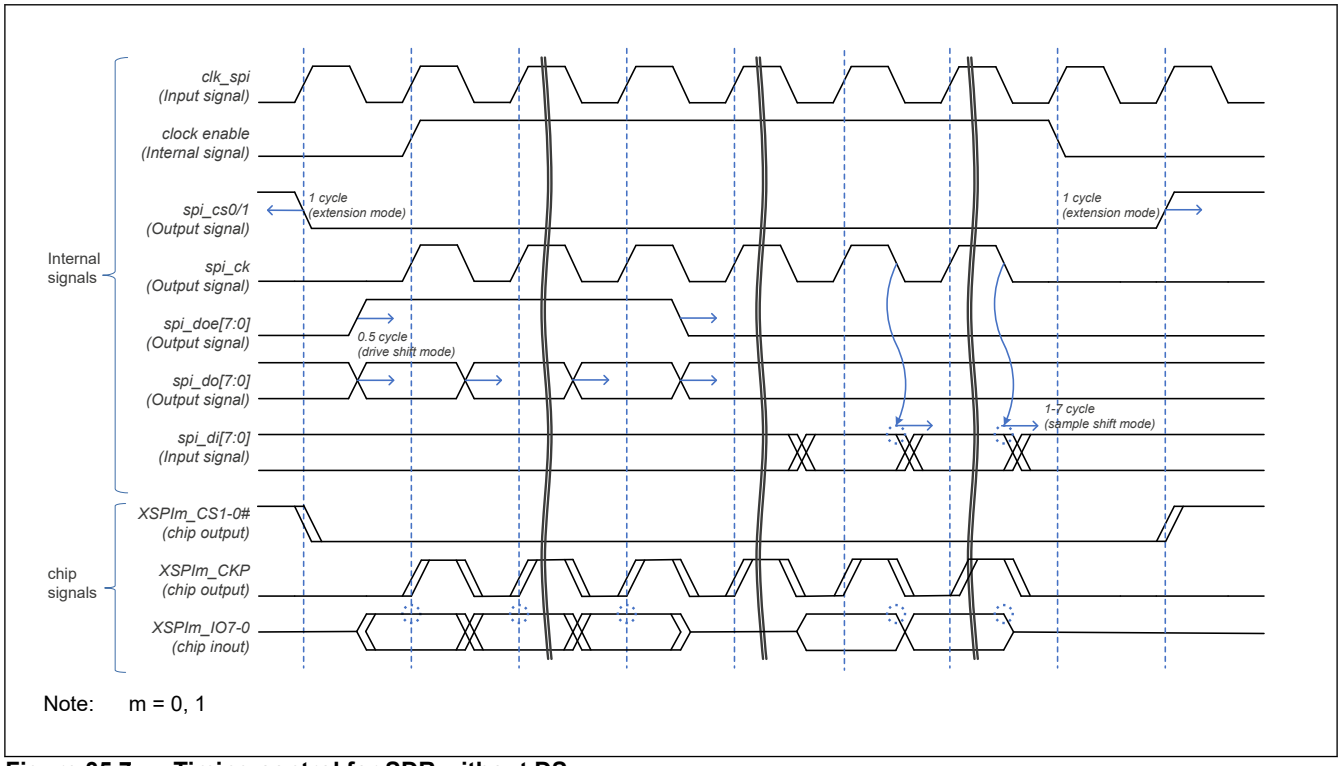


Figure 35.7 Timing control for SDR without DS

Figure 35.8 shows the default operation and timing control for SDR with DS.

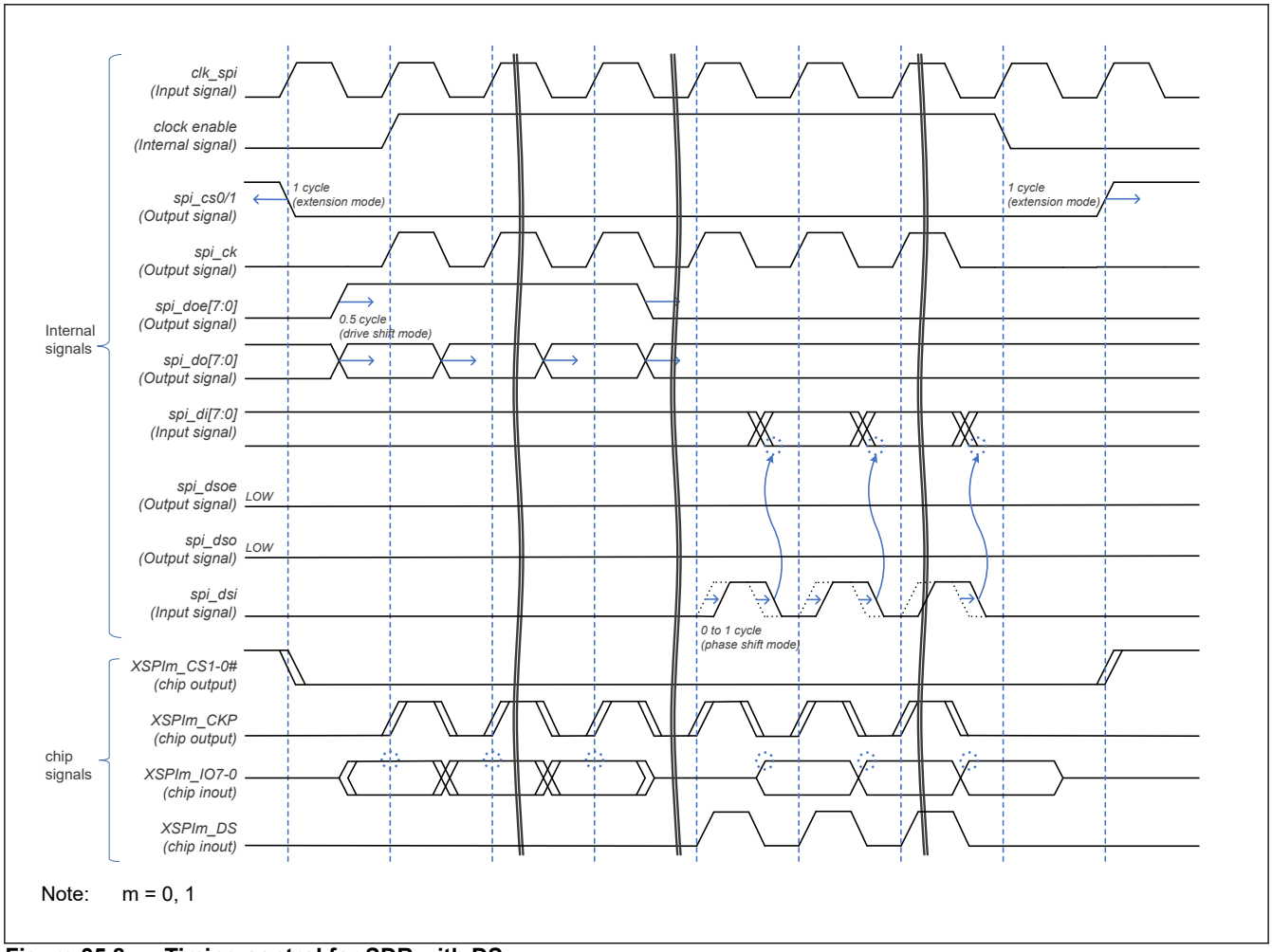


Figure 35.8 Timing control for SDR with DS

Figure 35.9 shows the default operation and timing control for DDR with DS.

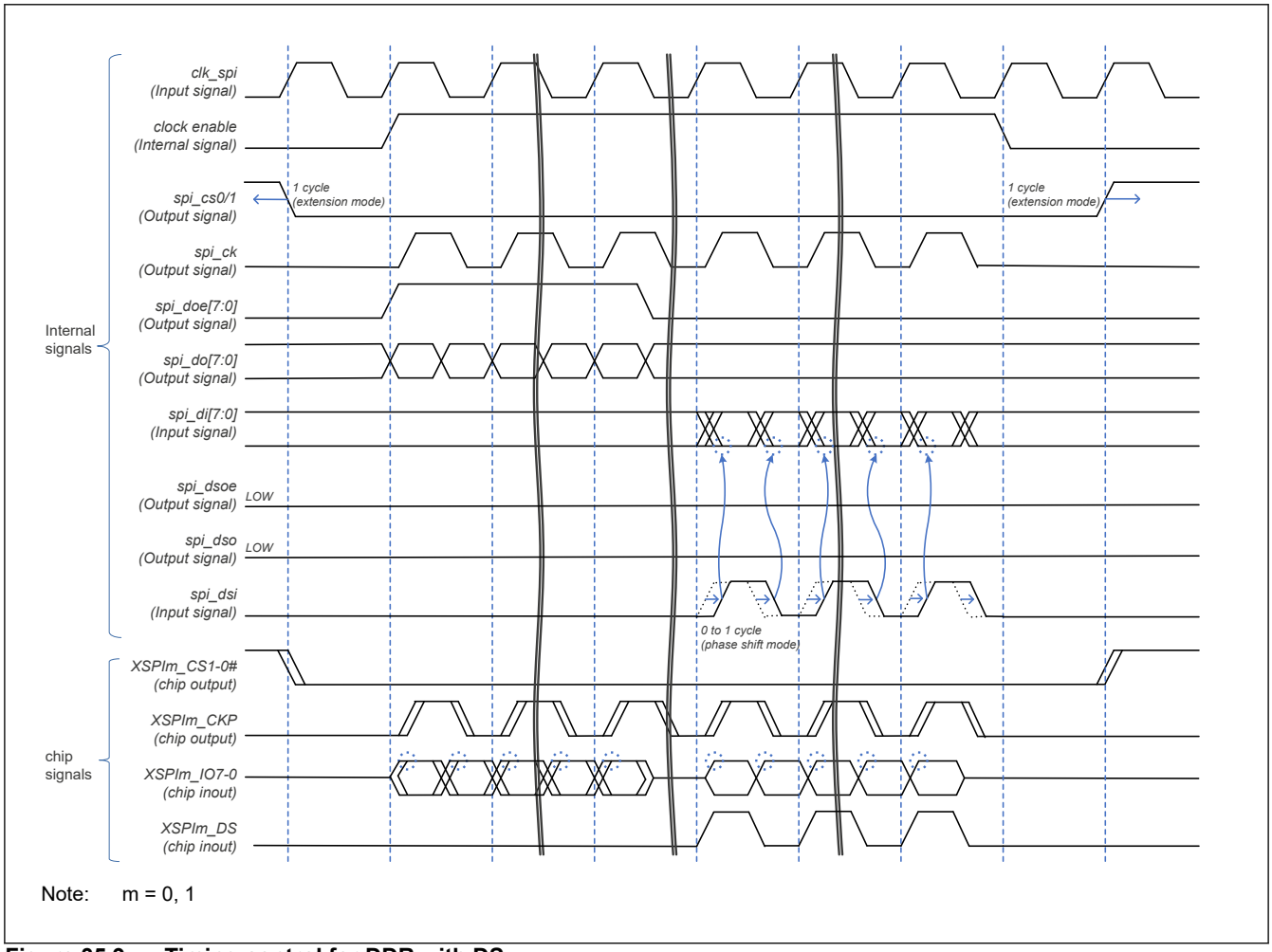


Figure 35.9 Timing control for DDR with DS

35.3.2 Manual-command

This section describes the manual-command mode. The manual-command has two functional modes: direct mode and periodic mode.

35.3.2.1 Direct Mode

This mode sequentially can issue up to four xSPI transactions configured and requested by Software. A series of transaction can be issued by a transaction request (CDCTL0.TRREQ = 1 with PERMD = 0). The number of transactions (CDCTL0.TRNUM[1:0]) can be configured up to 4. It can be used to change the mode or read the status of xSPI slave device.

Table 35.7 lists the configured register bits for direct manual-command. The operating flow is illustrated in Figure 35.22.

Table 35.7 Manual-command configuration for direct mode (1 of 2)

Transaction	Transaction type	Command	Command size	Address	Address size	Data (x = 0, 1)	Data size	Latency cycle
1st Transaction	CDTBUF0.TRTYPE	CDTBUF0.CMD[15:0]	CDTBUF0.CMDSIZE[1:0]	CDABUF0.ADD[31:0]	CDTBUF0.ADDSIZE[2:0]	CDDxBUF0.DATA[31:0]	CDTBUF0.DATASIZE[3:0]	CDTBUF0.LATE[4:0]
2nd Transaction	CDTBUF1.TRTYPE	CDTBUF1.CMD[15:0]	CDTBUF1.CMDSIZE[1:0]	CDABUF1.ADD[31:0]	CDTBUF1.ADDSIZE[2:0]	CDDxBUF1.DATA[31:0]	CDTBUF1.DATASIZE[3:0]	CDTBUF1.LATE[4:0]

Table 35.7 Manual-command configuration for direct mode (2 of 2)

Transaction	Transaction type	Command	Command size	Address	Address size	Data (x = 0, 1)	Data size	Latency cycle
3rd Transaction	CDTBUF2.TRTYPE	CDTBUF2.CMD[15:0]	CDTBUF2.CMDSIZE[1:0]	CDABUF2.ADD[31:0]	CDTBUF2.ADDSIZE[2:0]	CDDxBUF2.DATA[31:0]	CDTBUF2.DATASIZE[3:0]	CDTBUF2.LATE[4:0]
4th Transaction	CDTBUF3.TRTYPE	CDTBUF3.CMD[15:0]	CDTBUF3.CMDSIZE[1:0]	CDABUF3.ADD[31:0]	CDTBUF3.ADDSIZE[2:0]	CDDxBUF3.DATA[31:0]	CDTBUF3.DATASIZE[3:0]	CDTBUF3.LATE[4:0]

35.3.2.2 Periodic Mode

This mode periodically issues an xSPI read transaction configured and requested by Software. And it can compare the read value up to 4 bytes with expected value. The transaction is issued by a transaction request (CDCTL0.TRREQ = 1 with PERMD = 1). It can be used to alternate the status polling operation of xSPI slave device.

The periodic term is configured in Periodic transaction interval bits (CDCTL0.PERITV[4:0]). The expected value is configured in Periodic transaction expected and masked value bits (CDCTL1.PEREXP[31:0] and CDCTL2.PERMSK[31:0]). Table 35.8 lists the configured register bits for periodic manual-command. The operating flow is illustrated in Figure 35.23.

Table 35.8 Manual-command configuration for periodic mode

Transaction	Transaction type	Command	Command size	Address	Address size	Data (x = 0, 1)	Data size	Latency cycle
Read Transaction	CDTBUF0.TRTYPE	CDTBUF0.CMD[15:0]	CDTBUF0.CMDSIZE[1:0]	CDABUF0.ADD[31:0]	CDTBUF0.ADDSIZE[2:0]	CDDxBUF0.DATA[31:0]	CDTBUF0.DATASIZE[3:0]	CDTBUF0.LATE[4:0]

35.3.3 Memory-mapping

This section describes the memory-mapping mode. This mode automatically converts AHB access for pre-configured memory area into xSPI transaction.

35.3.3.1 Configuration

In this operation, the payload of address and data field are delivered from AHB signals. The information of command field and size are delivered from the configured register bits. Table 35.9 lists the register bits configured for memory-mapping.

Table 35.9 Memory-mapping configuration for memory area access (n = 0, 1) (1 of 2)

AHB Transaction	Format Change mode	Command	Command size	Address size	Latency cycle
Write for slave n memory area	Normal	CMCFG2CSn.WRCMD[15:8]	1 byte	CMCFG0CSn.ADDSIZE[1:0]	CMCFG2CSn.WRLATE[4:0]
	8D-8D-8D profile 1.0	CMCFG2CSn.WRCMD[15:0]	2 bytes		
	8D-8D-8D profile 2.0 Command Modifier	CMCFG2CSn.WRCMD[15:8]	1 byte	5 bytes	
	8D-8D-8D profile 2.0 Extended Command Modifier	CMCFG2CSn.WRCMD[15:13]	3 bits	45 bits	

Table 35.9 Memory-mapping configuration for memory area access (n = 0, 1) (2 of 2)

AHB Transaction	Format Change mode	Command	Command size	Address size	Latency cycle
Read for slave n memory area	Normal	CMCFG1CSn. RDCMD[15:8]	1 byte	CMCFG0CSn. ADDSIZE[1:0]	CMCFG1CSn. RDLATE[4:0]
	8D-8D-8D profile 1.0	CMCFG1CSn. RDCMD[15:0]	2 bytes		
	8D-8D-8D profile 2.0 Command Modifier	CMCFG1CSn. RDCMD[15:8]	1 byte	5 bytes	
	8D-8D-8D profile 2.0 Extended Command Modifier	CMCFG1CSn. RDCMD[15:13]	3 bits	45 bits	

Note: The MSByte of Address can be replaced with Address Replace Enable and Code bits (CMCFG0CSn.ADDRPEN[7:0] / ADDRPCD[7:0]).

35.3.3.2 Write Access Operation

In AHB write access for memory area, this xSPI Master stores all payload data in internal bridge buffer and then issues a write transaction into xSPI slave. Figure 35.10 shows the operation summary.

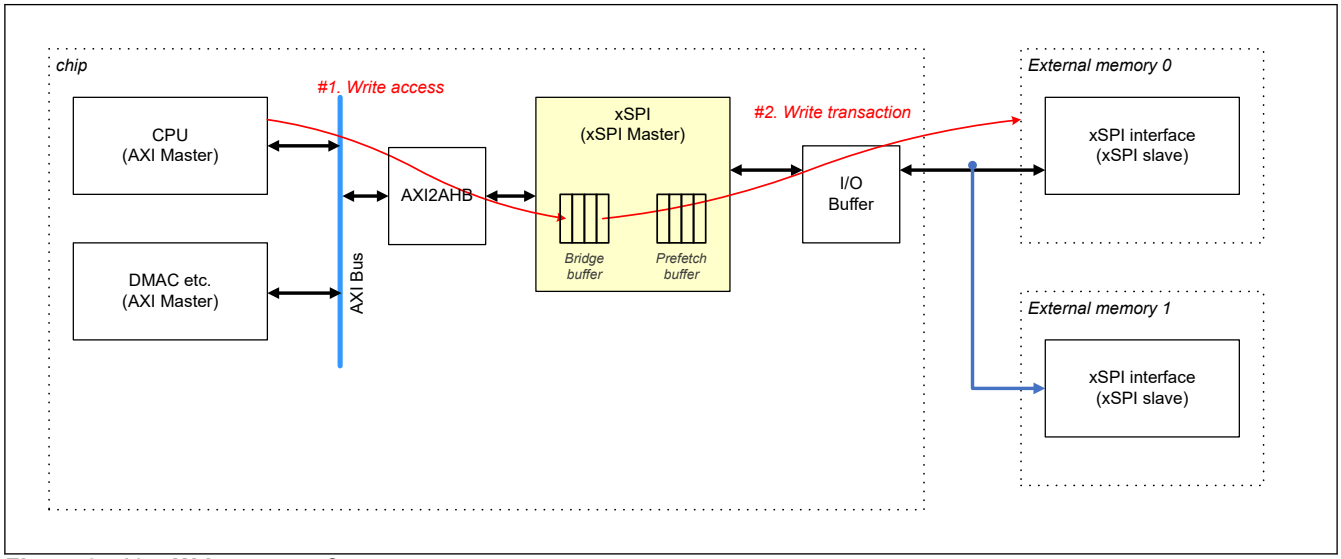


Figure 35.10 Write access for memory area

The operation of xSPI bus changes depending on AHB burst type. When HBURST is SINGLE or INCR, one AHB transaction occurs for one xSPI frame. When HBURST is INCR4, INCR8, or INCR16, AHB transactions equal to burst size occur for one xSPI frame. When HBURST is WRAP4, WRAP8, or WRAP16, AHB transactions occur for two xSPI frames. Figure 35.11 shows the relationship between AHB and xSPI frames.

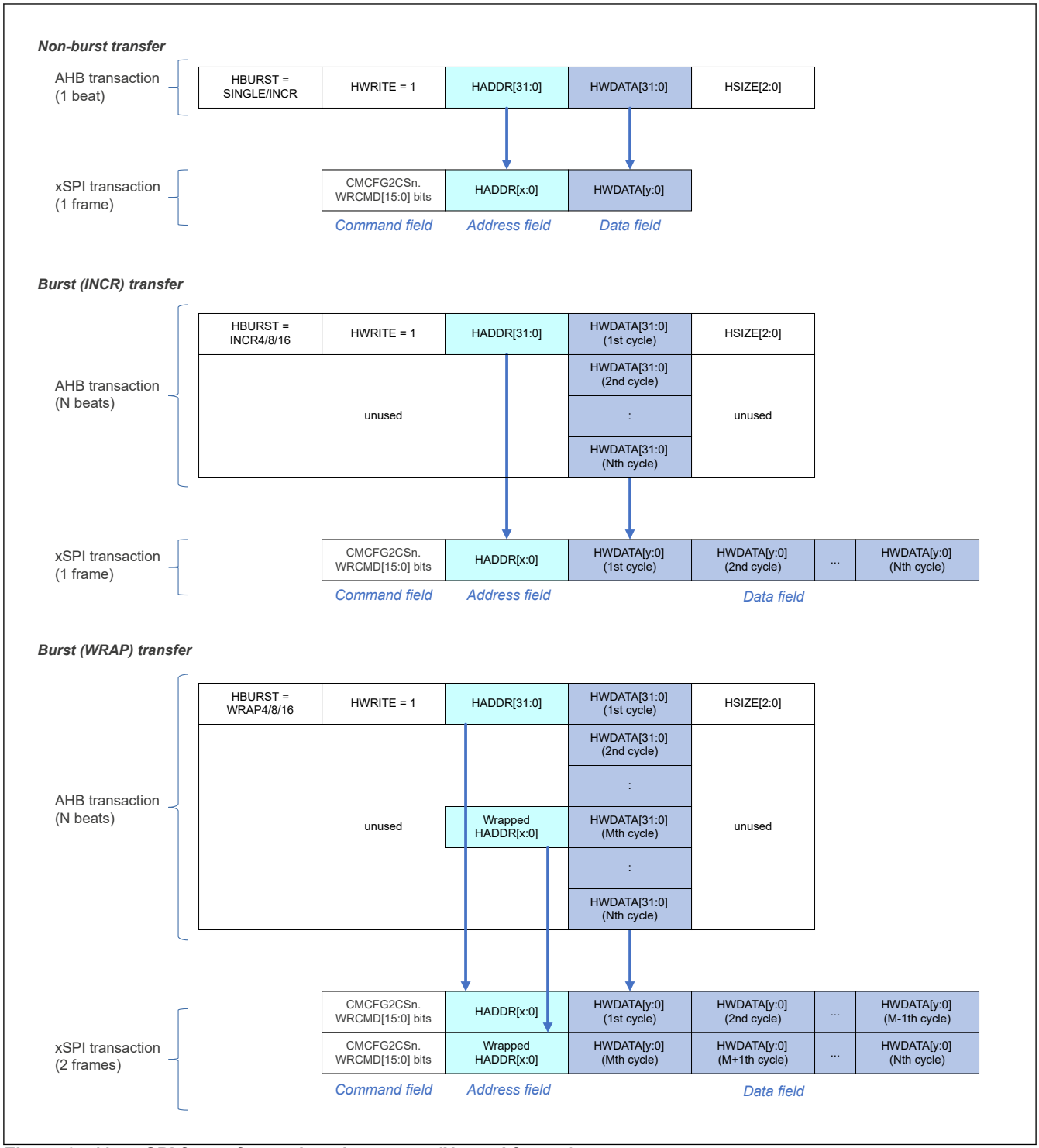


Figure 35.11 xSPI frame format in write access (Normal format)

35.3.3.3 Combination Function

In AHB write access for memory area, this xSPI master has the function to combinate the write data for high throughput on xSPI bus. When enabled this function (BMCFG.MWRCOMB = 1), this xSPI master transmits an xSPI frame with the selected size while the sequential address is incremental. When detected any transaction with non-incremental address, any transaction for different slave, or a read transaction before reaching to the target size (BMCFG.MWRSIZE[7:0]), it transmits the pending data into xSPI bus. And also, when set Memory Write Data Push bit (BMCTL1.MWRPUSH), it transmits the pending data into xSPI bus.

This function could be useful for any slave device to request a chunk of data at a time. In the case, AHB Master should continue to provide the fixed data size with incremental address. Ex. there is any device to request to write in page unit.

Figure 35.12 shows the operation when enable the combination function.

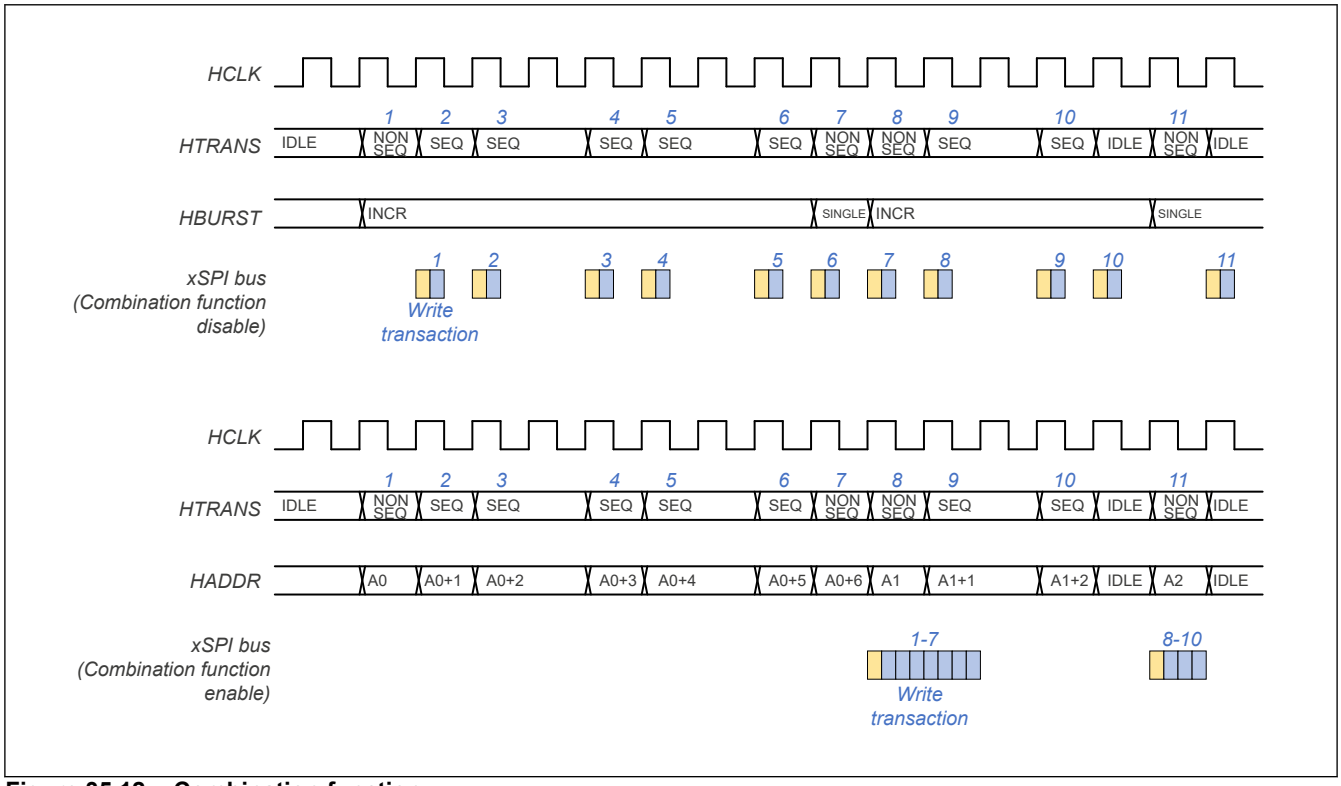


Figure 35.12 Combination function

35.3.3.4 Read Access Operation

In AHB read access for memory area, soon after detected the read access, this xSPI Master issues a read transaction into xSPI slave. Figure 35.13 shows the operation summary.

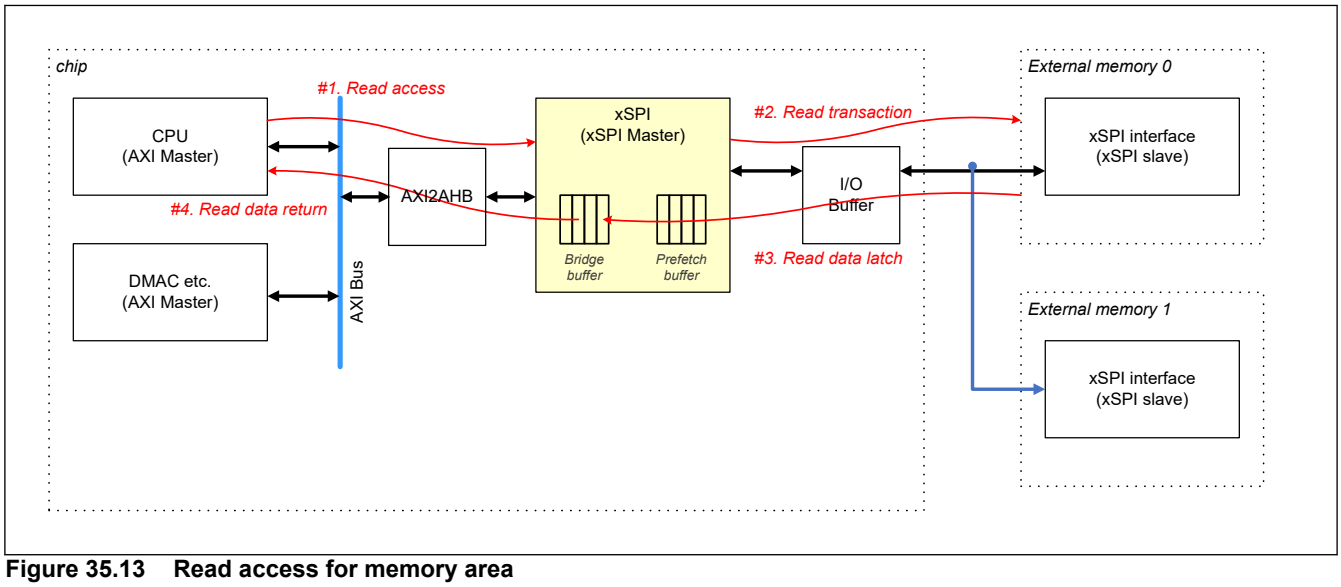


Figure 35.13 Read access for memory area

The operation of xSPI bus changes depending on AHB burst type. When HBURST is SINGLE or INCR, one AHB transaction triggers for one xSPI frame. When HBURST is INCR4, INCR8, or INCR16, AHB transactions equal to burst size triggers for one xSPI frame. When HBURST is WRAP4, WRAP8, or WRAP16, AHB transactions trigger for two xSPI frames. Figure 35.14 shows the relationship between AHB and xSPI frames.

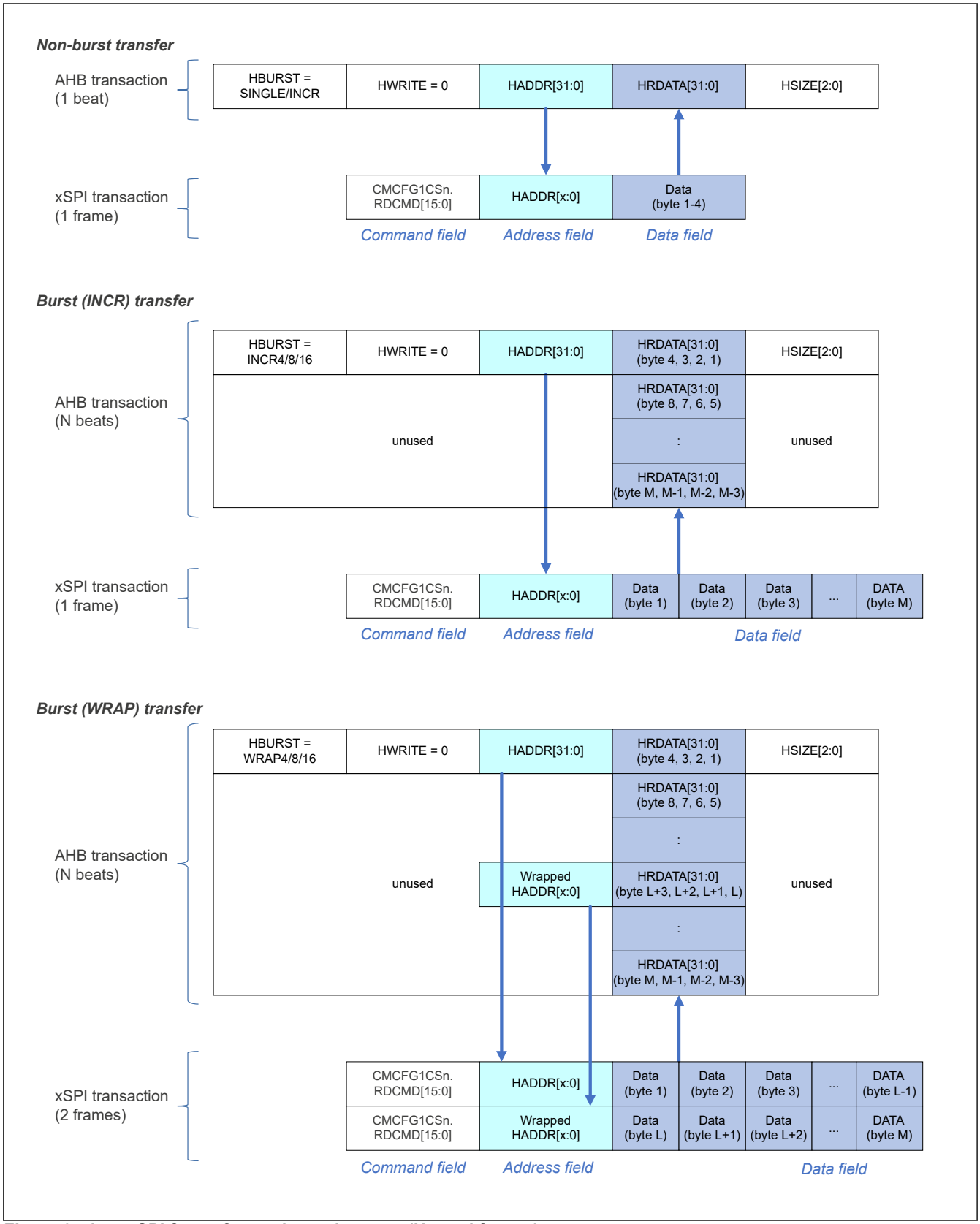


Figure 35.14 xSPI frame format in read access (Normal format)

35.3.3.5 Prefetch Function

In AHB read access for memory area, this xSPI Master has the function to prefetch the read data for lower latency. When enabled this function (BMCFG.PREEN = 1), this xSPI Master continues to read the incremental address and store the read

data from xSPI slave in internal prefetch buffer. And the following AHB read access searches for prefetch buffer. If found the target read data in prefetch buffer, this xSPI Master returns the data from prefetch buffer. If not found, this xSPI Master clears prefetch buffer and newly issues a read transaction into xSPI slave. This function is effective in application such as the consecutive read addresses are close. But it is not effective in application such as the consecutive read addresses are not incremental because xSPI read frame for prefetch uses xSPI bus. Figure 35.15 shows the operation summary.

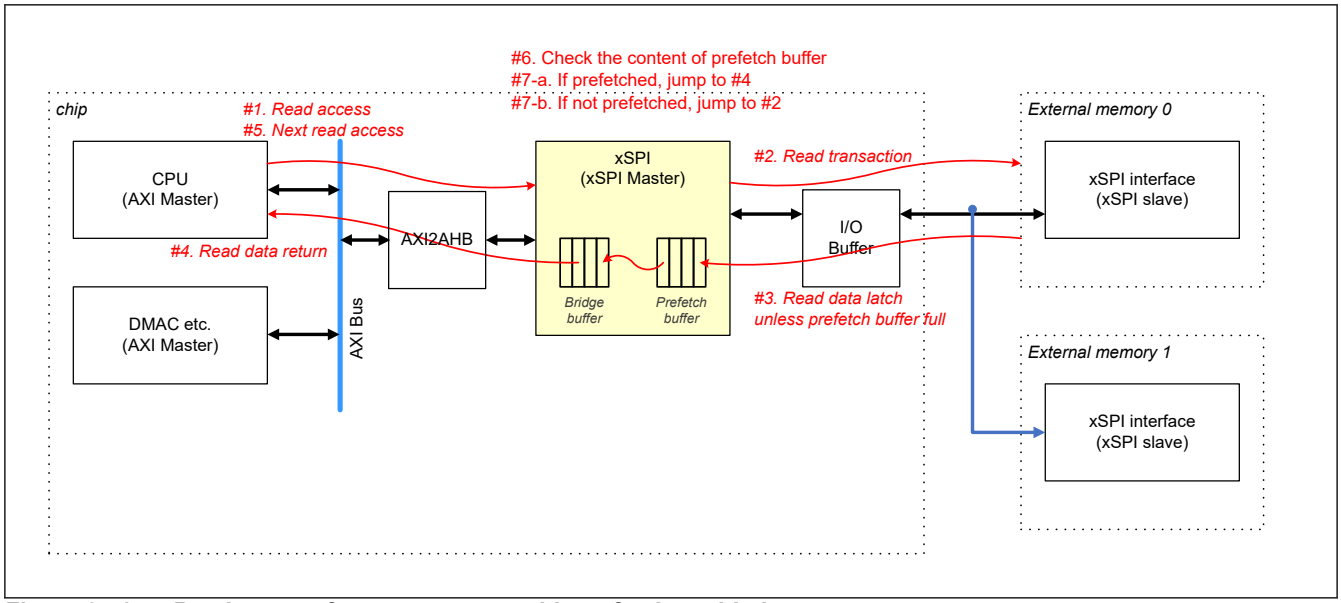


Figure 35.15 Read access for memory area with prefetch enabled

Note: When enabled this prefetch function, AHB Master could read from not a slave device but the internal prefetch buffer. When accessed to the same address from multiple AHB Masters, this xSPI Master does not guarantee to read the latest data. If AHB Master wish to read the latest data from a slave device, it should read after cleared the prefetch buffer (BMCTL1.PBUFCLR).

Note: Prefetch buffer is implemented as FIFO-based, and when read access is issued, the data before the access address is discarded from the buffer. And when next access is issued to the region which is discarded at previous access, this module issues the xSPI read access again to fill the prefetch buffer.

35.3.3.6 XiP Mode

Some slave devices have a mode (XiP mode) in which the command phase is not required for lower latency. While in this mode, the xSPI master skips sending the command and the slave device implicitly performs the command that was executed in the previous transaction. When enabled XiP mode bit (CMCTL.XIPEN = 1), this xSPI master inserts XiP enter code (CMCTL.XIPENCODE[7:0]) in latency field. When disabled XiP mode bit (CMCTL.XIPEN = 0), this xSPI master inserts XiP exit code (CMCTL.XIPEXCODE[7:0]) in latency field. This function is available only for memory-mapping mode.

And when transmitting XiP disable pattern, this xSPI master clears XiP mode bit and disables XiP mode.

Note: In case of no cycle inserting XiP code in latency field, this xSPI Master could not insert XiP code.

Note: XiP mode could be used only for unidirectional access to a slave. The write transaction and read transaction should be separated.

35.3.4 Pattern Control

This xSPI Master has the function to transmit any pattern which is not xSPI frame format. It starts to transmit by the pattern issue trigger bits (LPCTL0-1.PATREQ). It is possible to transmit 3 types of patterns.

35.3.4.1 XiP Disable Pattern

XiP Disable pattern transmits any pattern with the configured length and value (LPCTL0.XD1LEN[4:0] / XD1VAL / XD2LEN[4:0] / XD2VAL). It uses spi_ck, spi_doe/do signals. The number of output pin can be configured by XiP Disable pattern pin bits (LPCTL0.XDPIN[1:0]). It may be used to disable XiP mode for legacy SPI. Figure 35.16 shows the timing-chart.

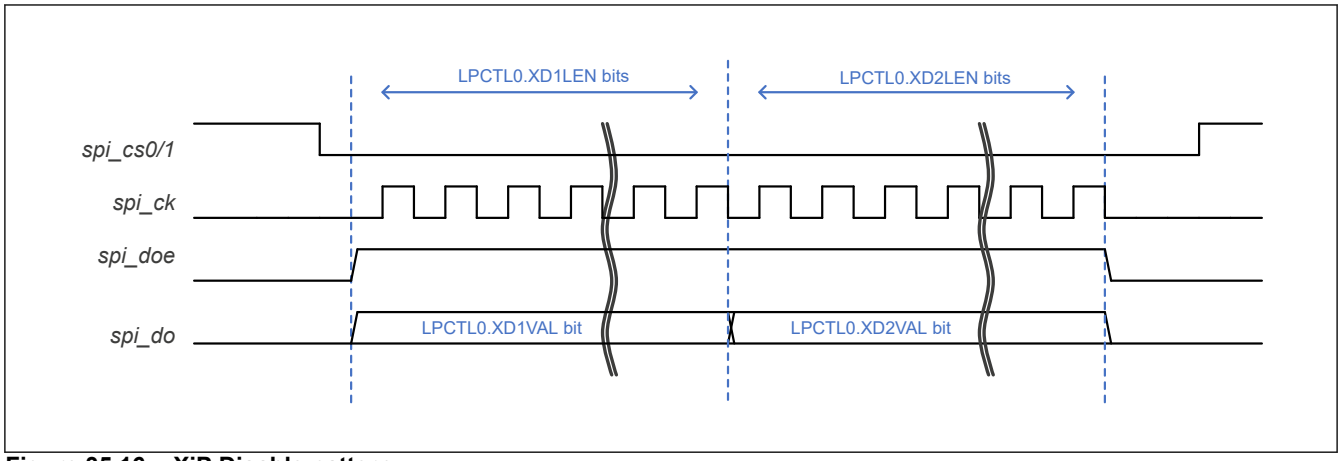


Figure 35.16 XiP Disable pattern

35.3.4.2 Reset Pattern

Reset pattern transmits the pattern specified in Serial Flash Reset Signaling Protocol. CS Low/High width is configured with Reset Pattern Length bits (LPCTL1.RSTWID[2:0]). xSPI slave will sample the data input at the rising edge of CS. Setup time for data output is configured with Reset pattern data output setup time bits (LPCTL1.RSTSU[2:0]). The setup time should be less than Reset pattern width always. Figure 35.17 shows the timing-chart.

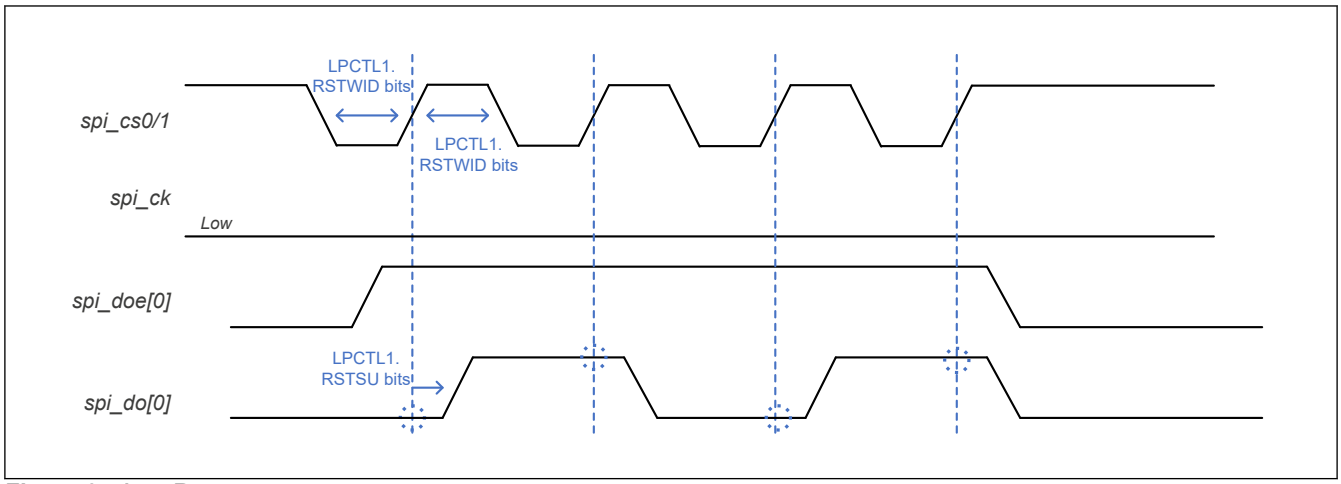


Figure 35.17 Reset pattern

Note: In the protocol, CS Low/High width is defined as minimum 500 ns and Setup time is defined as minimum 6 ns.

35.3.4.3 CS Only Pattern

CS Only pattern activates CS port with the configured length bits (LPCTL1.RSTWID[2:0]). It may be used to resume from Deep Power Down state. Figure 35.18 shows the timing-chart.

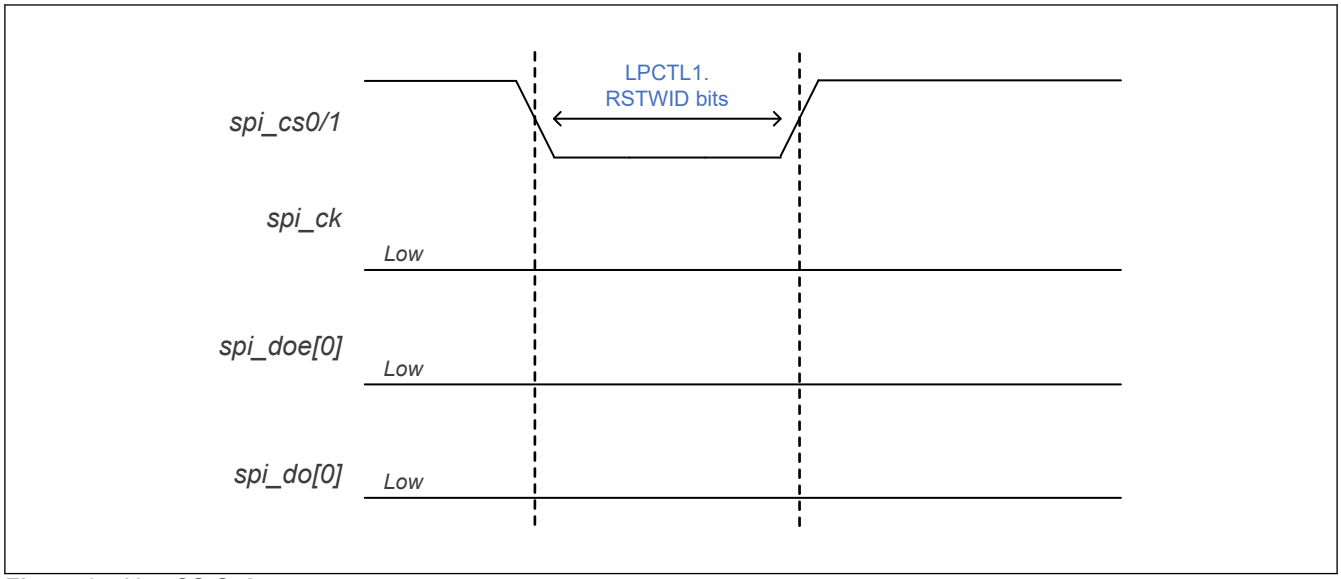


Figure 35.18 CS Only pattern

35.3.5 Integrity Checking

This xSPI Master can detect some errors. Table 35.10 lists the error list and the detail behavior.

When reset of xSPI master is needed due to fatal error, set module reset register (MRCTLA) to reset xSPI master. After confirming it is reset surely, release module reset.

Table 35.10 Error list

Error type	Event	Flag bit	Note (action)
Calibration failed	When the read data did not match the expected value during automatic calibration	INTS. CAFAILCS0-1	It results in writing unexpected data to xSPI slave.
AHB bus error	When an error response occurred on AHB slave interface for memory-mapping	INTS. BUSERR	This is a fatal error. This xSPI master shall be initialized. When detected this error in AHB write transaction, the internal bridge buffer is cleared. It means the pending write data is lost.
ECC error detection	When detected the falling edge on ECSn# port It can be useful only for xSPI slave with ECC detection function.	INTS. ECSCS0-1	Only notify the error event of xSPI slave.
DS timeout	When DS does not toggle in read transaction with using DS	INTS. DSTOCS0-1	Both xSPI master and xSPI slave should be reset for fatal error.
Periodic transaction timeout	When the read value does not match with the expected value in periodic manual-command mode	INTS. PERTO	Depending on the status of function

35.3.6 Interrupts

This xSPI Master has an interrupt port.

It can monitor with Interrupt Status Register (INTS). In case of initialization phase, it can be programmable with Interrupt Enable register (INTE). Table 35.11 lists the related register bit.

Table 35.11 xSPI interrupt sources

Name	Interrupt Sources	CPU0 GIC Request	DMAC Activation
XSPIm_INT	Interrupt	Possible	Not possible
XSPIm_INTERR	Error interrupt	Possible	Not possible

Note: m = 0, 1

Table 35.12 Interrupt register bit

Flag bit	Enable bit	Clear bit	Interrupt Sources (m = 0, 1)
CASUCCS1	CASUCCS1E	CASUCCS1C	XSPIm_INT
CASUCCS0	CASUCCS0E	CASUCCS0C	
CAFAILCS1	CAFAILCS1E	CAFAILCS1C	XSPIm_INTERR
CAFAILCS0	CAFAILCS0E	CAFAILCS0C	
BUSERR	BUSERRE	BUSERRC	XSPIm_INTERR
BRGUF	BRGUFE	BRGUFC	XSPIm_INTERR
BRGOF	BRGOFE	BRGOFC	
INTCS1	INTCS1E	INTCS1C	XSPIm_INTERR
INTCS0	INTCS0E	INTCS0C	
ECSCS1	ECSCS1E	ECSCS1C	XSPIm_INTERR
ECSCS0	ECSCS0E	ECSCS0C	
DSTOCS1	DSTOCS1E	DSTOCS1C	XSPIm_INTERR
DSTOCS0	DSTOCS0E	DSTOCS0C	
PERTO	PERTOE	PERTOC	XSPIm_INTERR
INICMP	INICMPE	INICMPC	XSPIm_INT
PATCMP	PATCMPE	PATCMPC	XSPIm_INT
CMDCMP	CMDCMPE	CMDCMPC	XSPIm_INT

35.3.7 Flows of Operations

35.3.7.1 Flow of Configuration

Figure 35.19 shows flow of configuration.

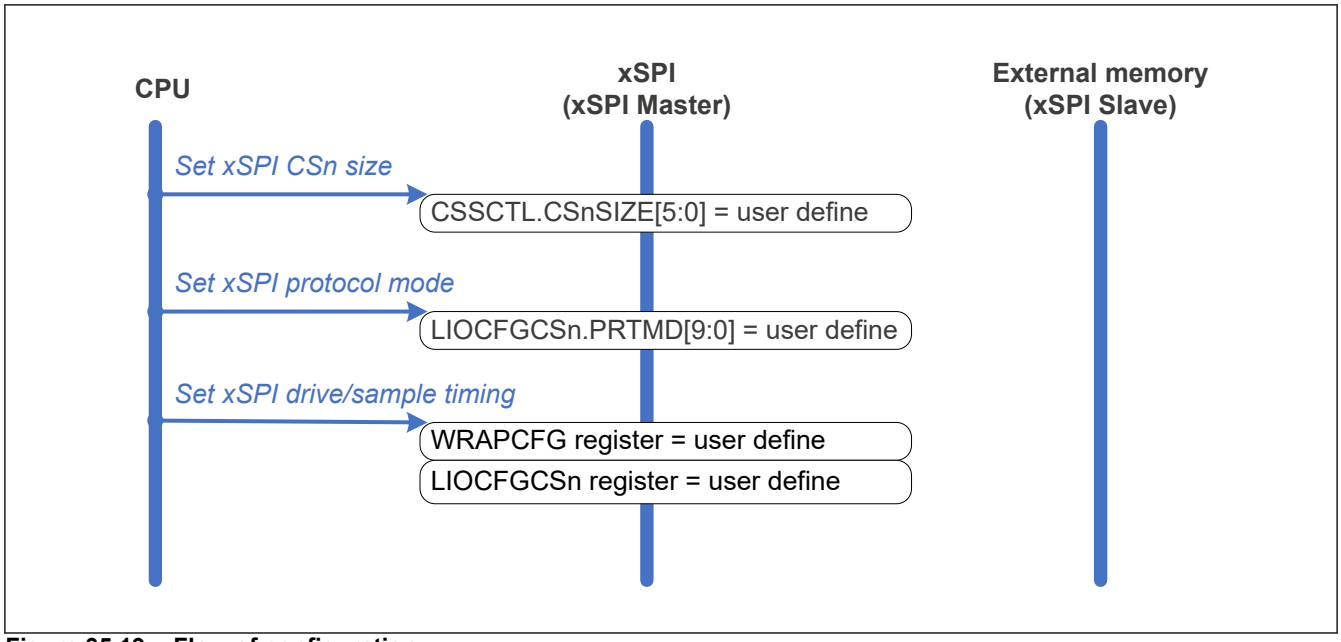


Figure 35.19 Flow of configuration

35.3.7.2 Flow of Communication Stop

Figure 35.20 shows flow of communication stop.

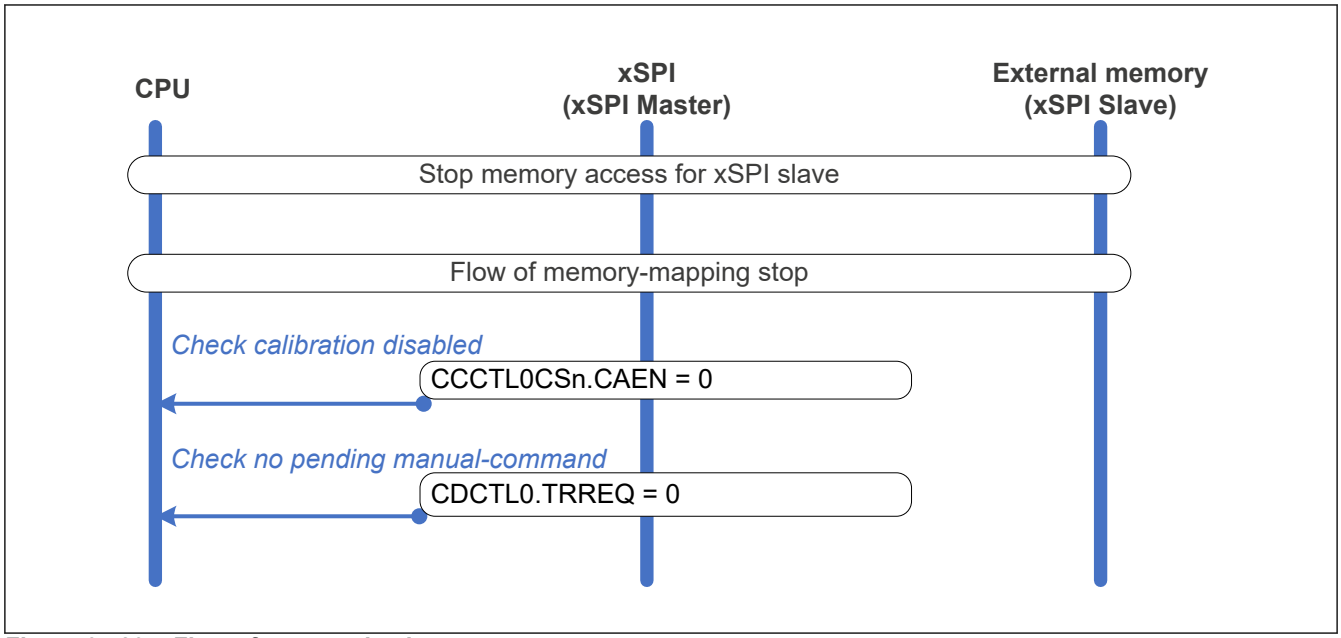


Figure 35.20 Flow of communication stop

Note: In case of re-config of any configuration register, all communication with xSPI slave shall be stopped completely. It means that the automatic calibration is disabled, and there is no pending manual-command and memory-mapping access.

35.3.7.3 Flow of Automatic Calibration

Figure 35.21 shows flow of automatic calibration.

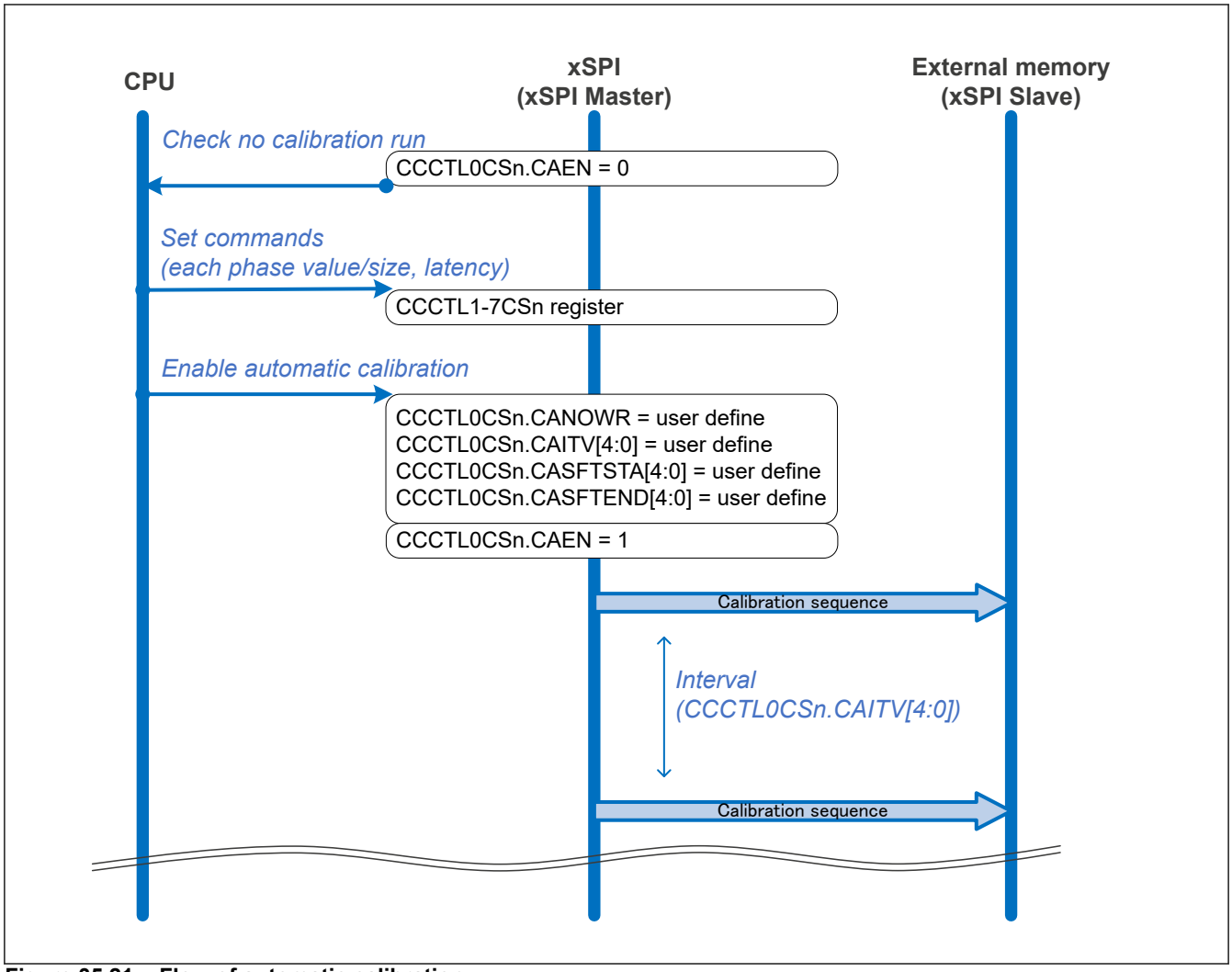


Figure 35.21 Flow of automatic calibration

35.3.7.4 Flow of Manual-command Procedure

Figure 35.22 shows manual-command procedure for direct mode.

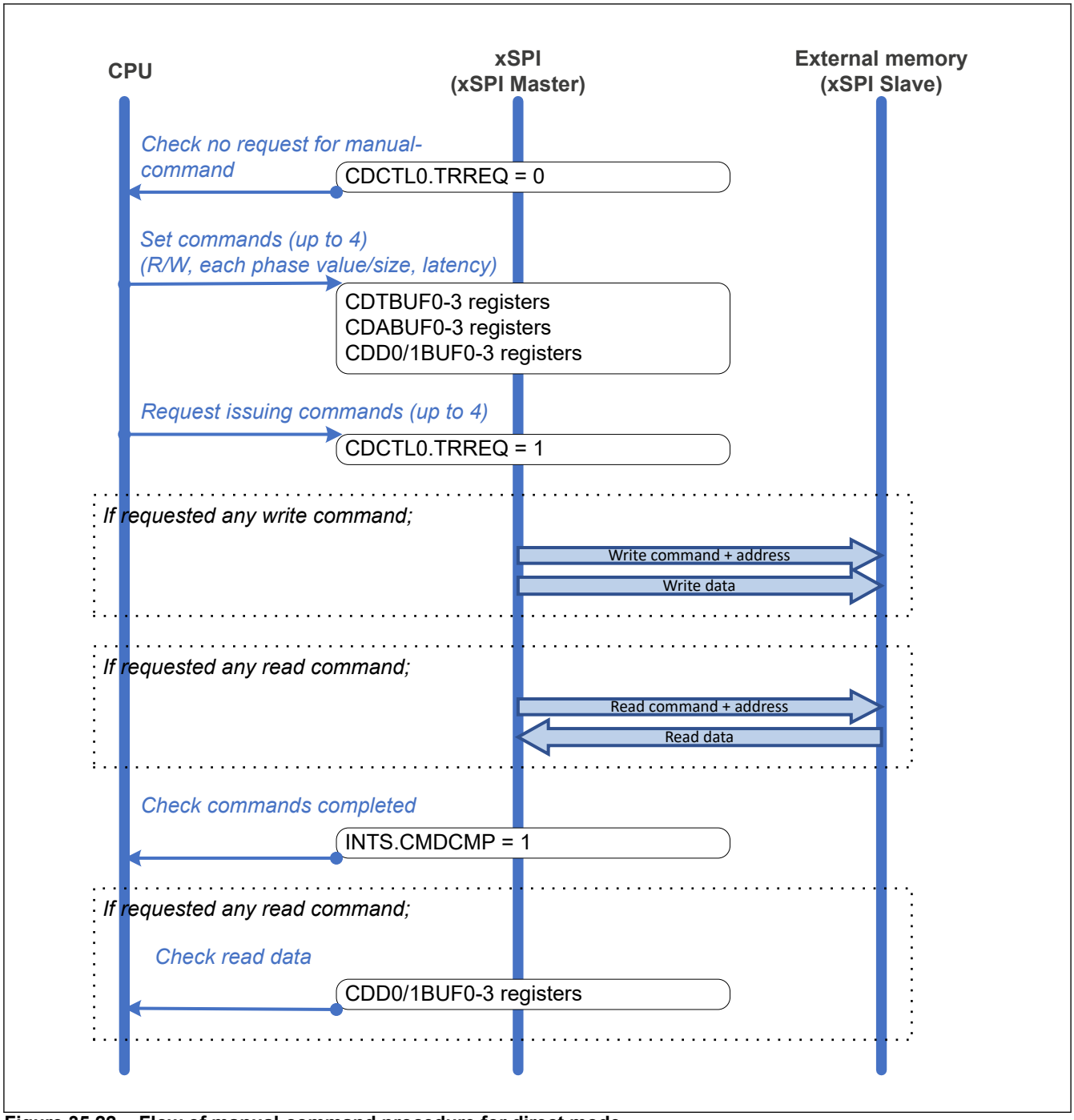


Figure 35.22 Flow of manual-command procedure for direct mode

Figure 35.23 shows manual-command procedure for periodic mode.

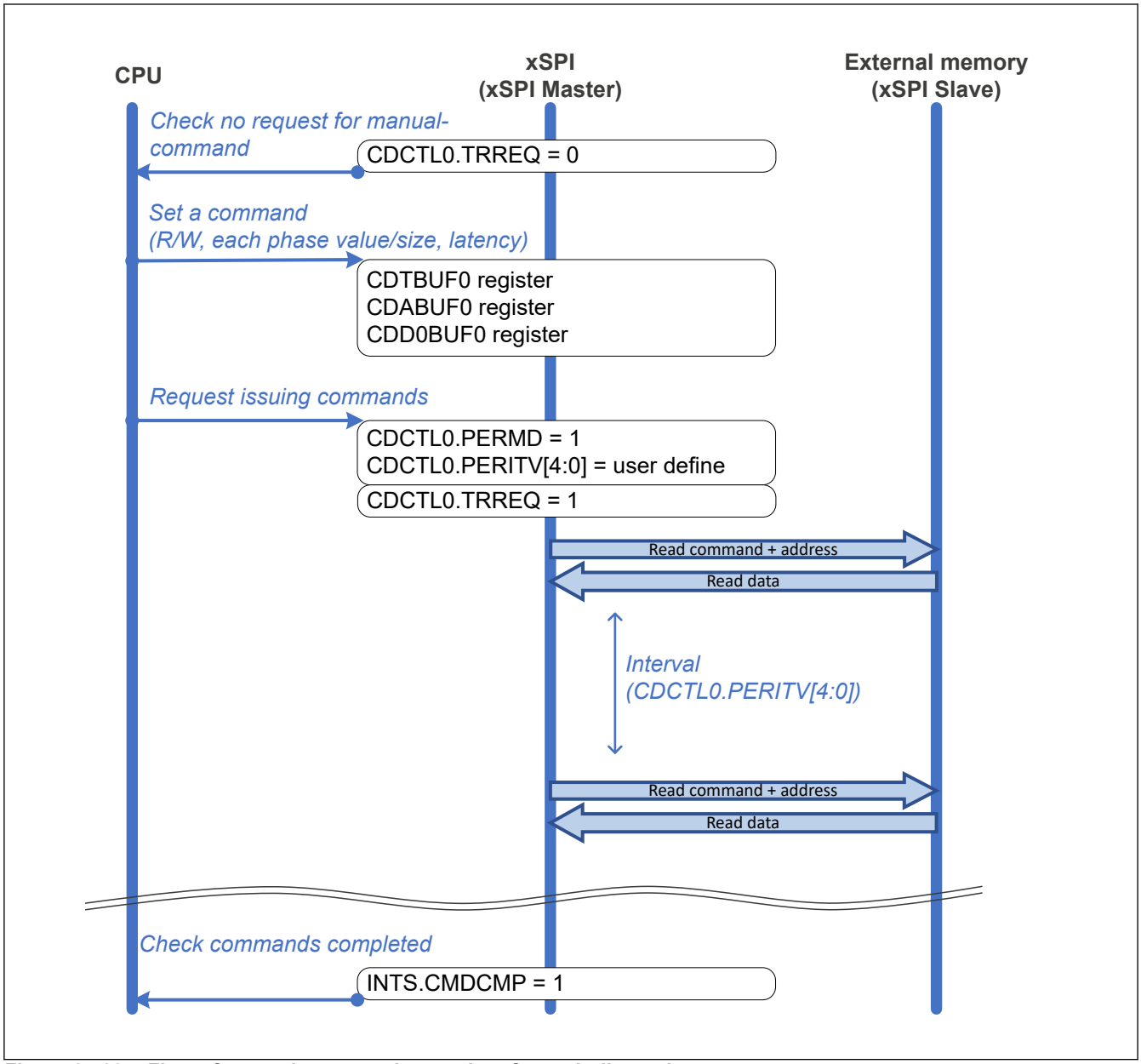


Figure 35.23 Flow of manual-command procedure for periodic mode

35.3.7.5 Flow of Memory-mapping

Figure 35.24 shows flow of memory-mapping.

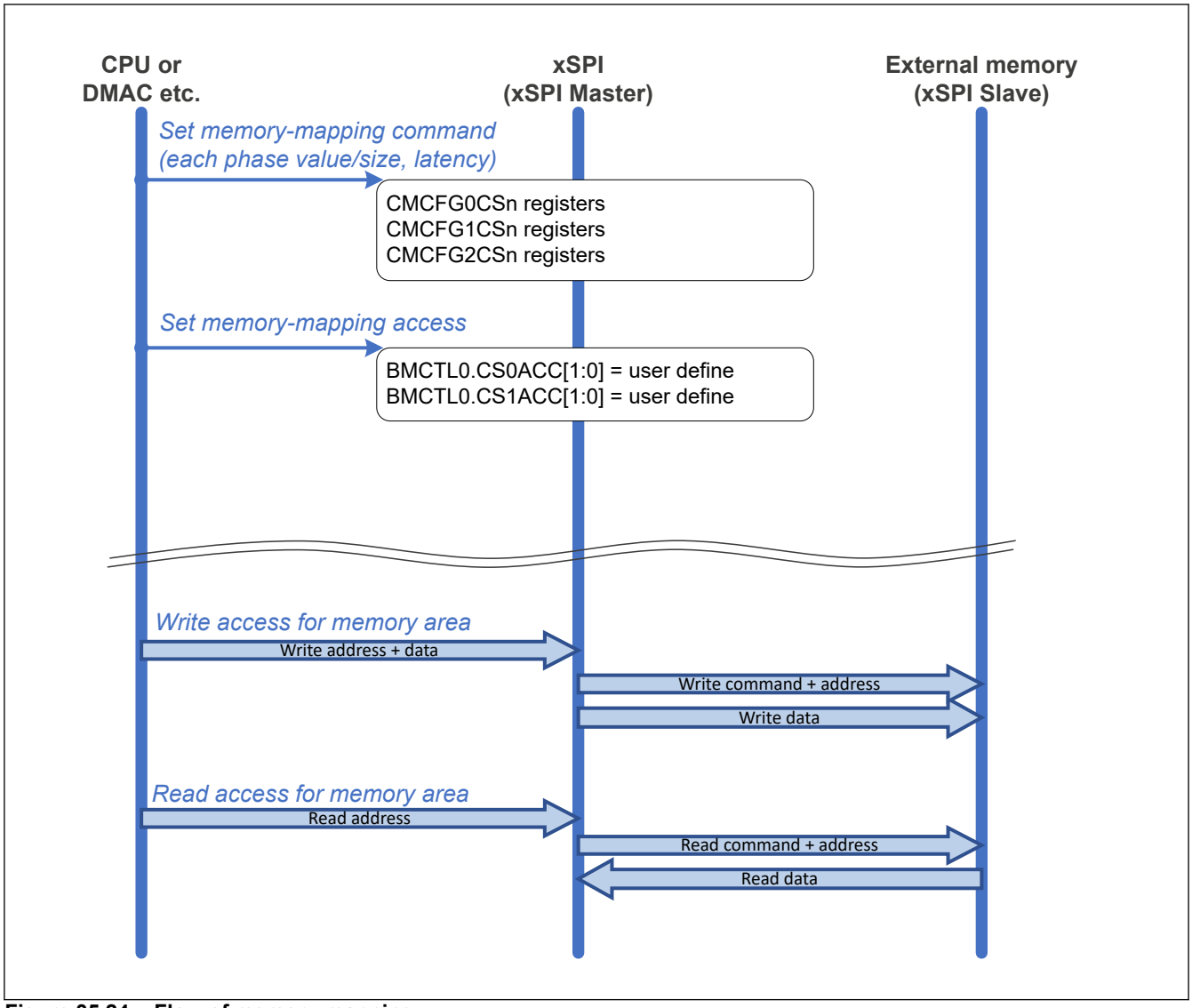


Figure 35.24 Flow of memory-mapping

35.3.7.6 Flow of Memory-mapping Stop

Figure 35.25 shows flow of memory-mapping stop.

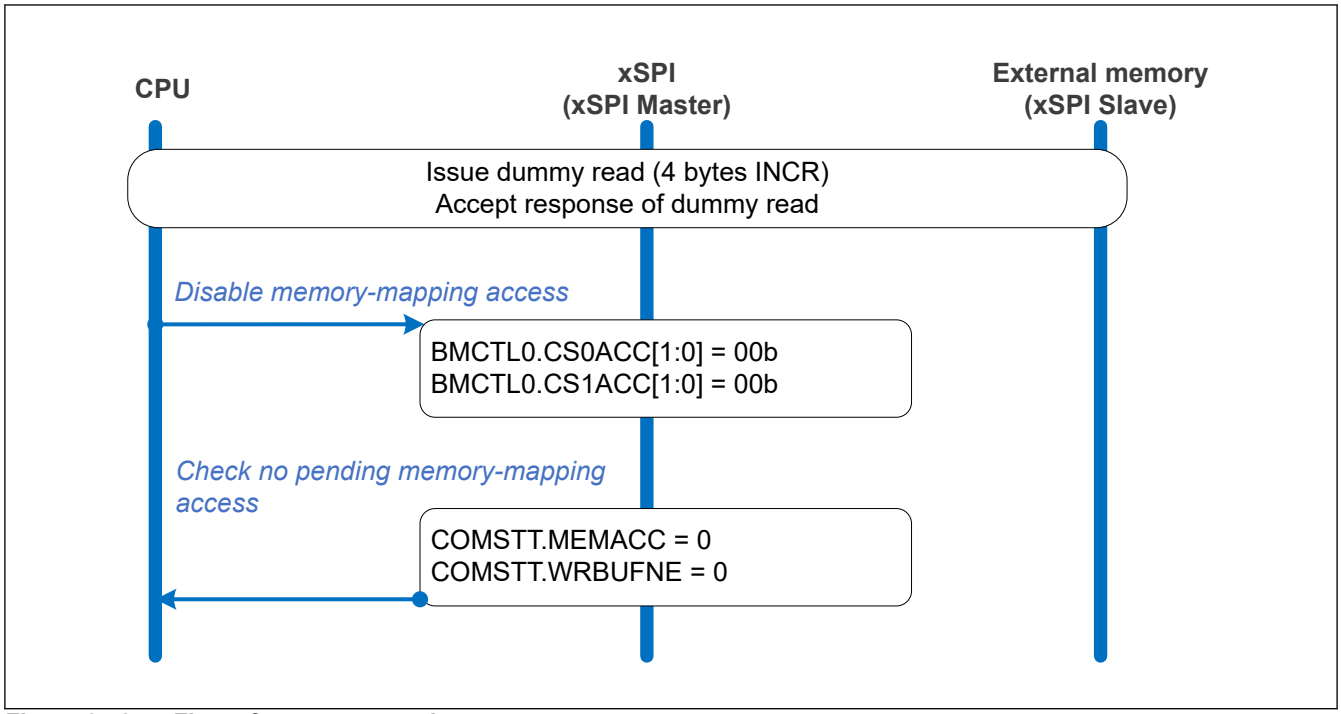


Figure 35.25 Flow of memory-mapping stop

35.3.7.7 Flow of Pattern Request

Figure 35.26 shows flow of pattern request. Before requesting any pattern, any ongoing commands should be completed or canceled.

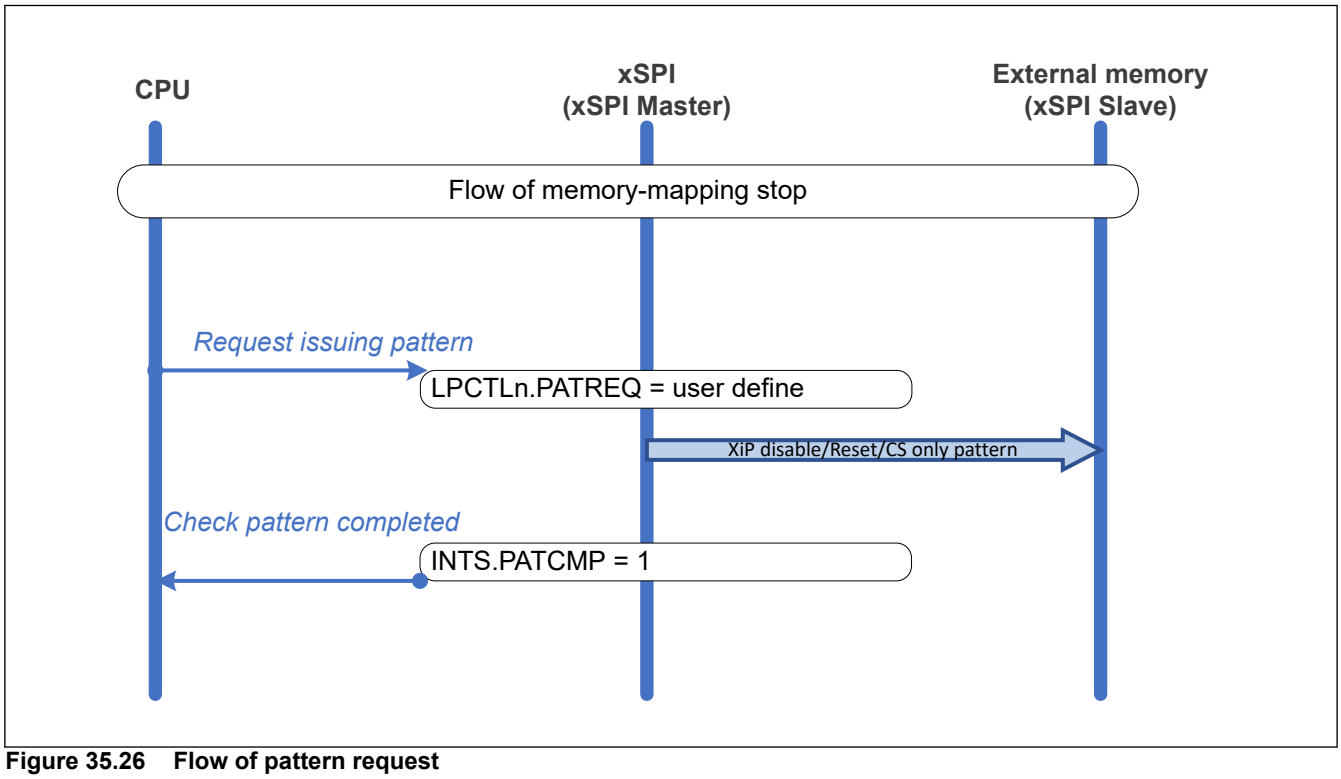


Figure 35.26 Flow of pattern request

35.3.7.8 Flow of XiP Mode

Figure 35.27 shows flow of XiP mode enable/disable.

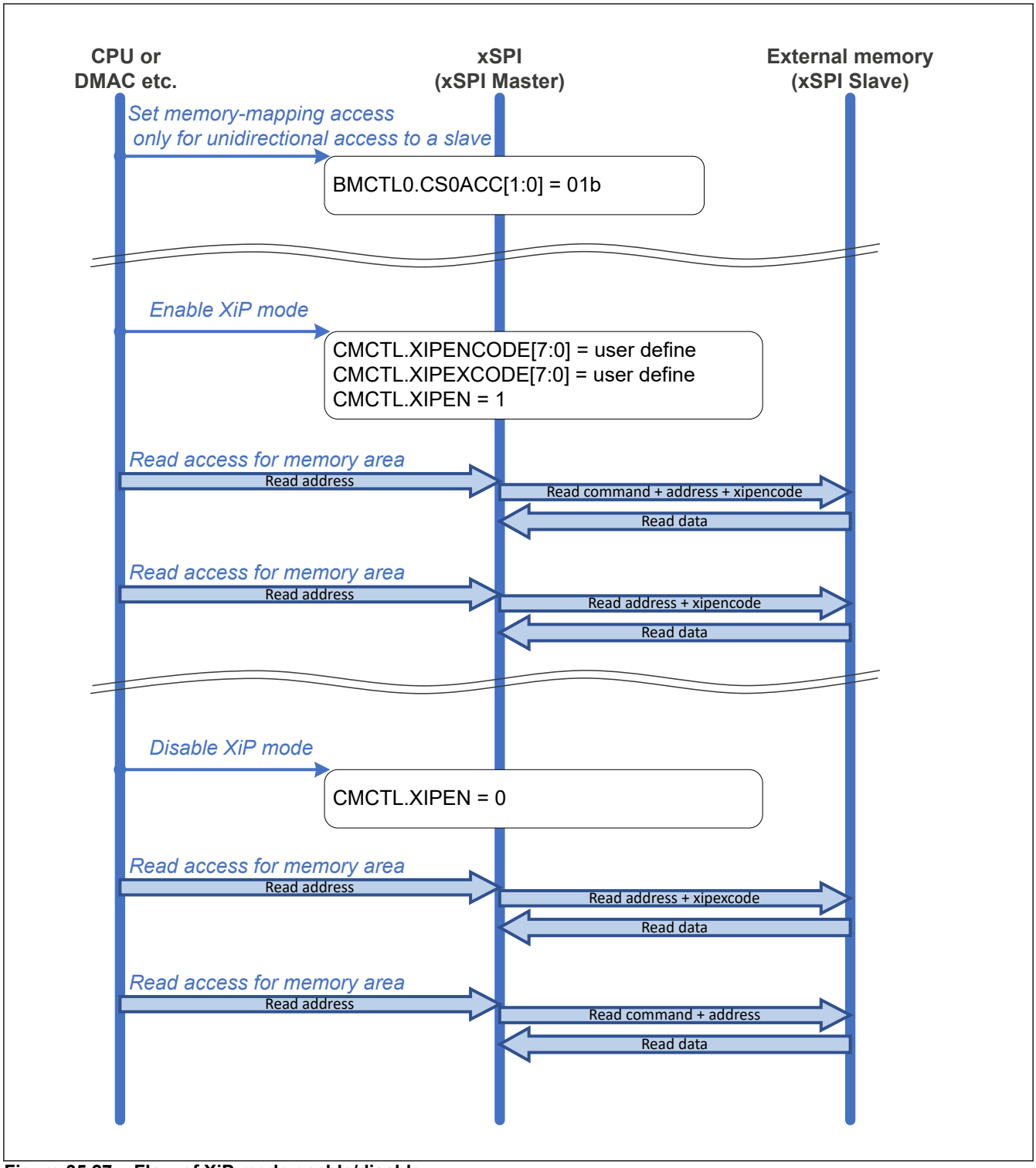


Figure 35.27 Flow of XiP mode enable/disable

35.3.8 Usage Notes

- If prefetch function is enabled and CS size control register (CSSCTL) is not set correctly compared to size of connected slave device, xSPI master may access outside of the slave device address. It may cause DS timeout in case of 8D-8D-8D. Make sure to set the correct slave device size to CSSCTL register.
- DDR access without DS is not supported.

36. CRC Operation Unit (CRC)

36.1 Overview

Cyclic redundancy check (CRC) operation units generate CRC codes. Table 36.1 lists the specifications of the CRC operation unit, and Figure 36.1 shows a block diagram of the CRC operation unit.

Table 36.1 CRC specifications

Parameter	Description	
Number of channels	2 channels	
Data size	8 bits	32 bits
Data for CRC calculation*1	CRC codes are generated for any desired data in 8n-bit units (where n is a natural number)	CRC codes are generated for any desired data in 32n-bit units (where n is a natural number)
CRC processor unit	8-bit parallel processing	32-bit parallel processing
CRC generating polynomial	One of three generating polynomials is selectable • 8-bit CRC $X^8 + X^2 + X + 1$ (CRC-8) • 16-bit CRC $X^{16} + X^{15} + X^2 + 1$ (CRC-16) $X^{16} + X^{12} + X^5 + 1$ (CRC-CCITT)	One of two generating polynomials is selectable • 32-bit CRC $X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$ (CRC-32) $X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$ (CRC-32C)
CRC calculation switching	The order of the bits produced by CRC calculation can be switched for LSB first or MSB first communication	
Low power consumption	Module-stop state can be set.	

Note 1. The circuit does not have a function to divide data for calculation into CRC calculation units. Write data in 8-bit or 32-bit units.

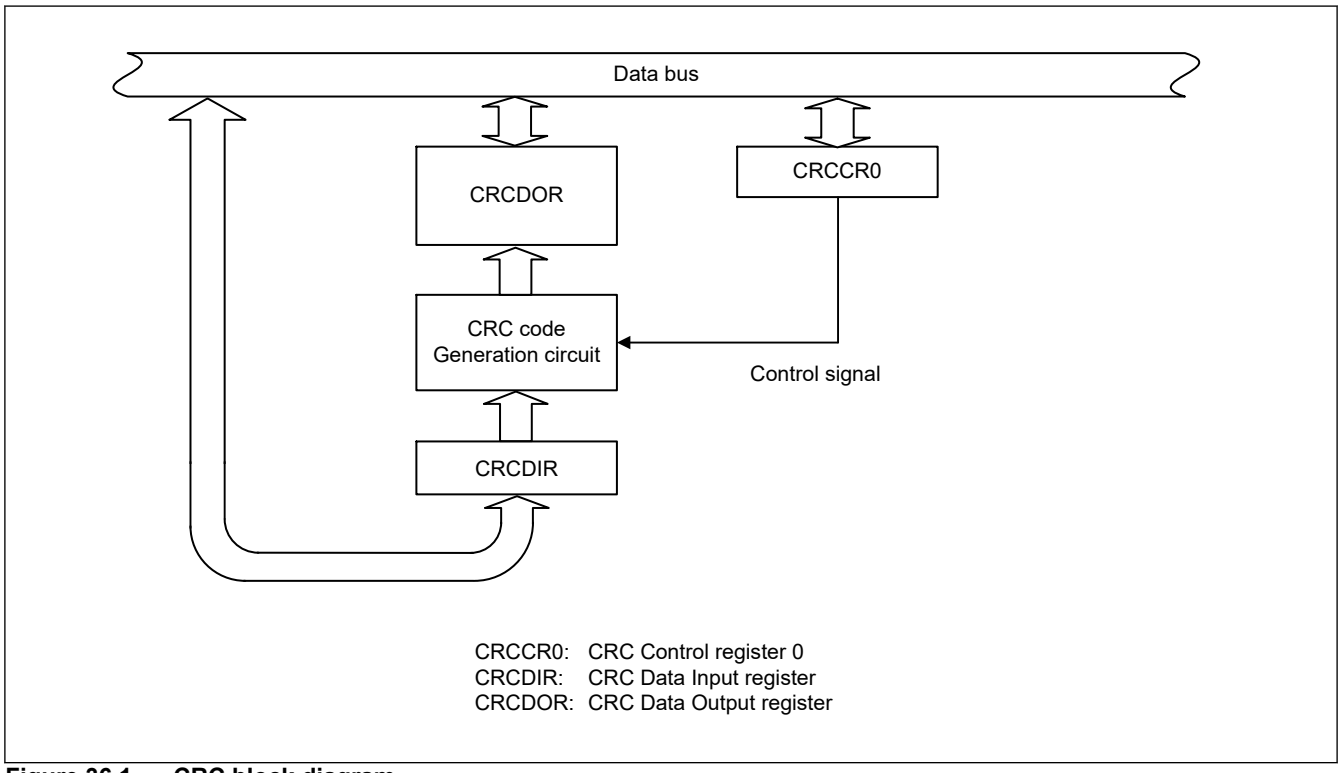


Figure 36.1 CRC block diagram

36.2 Register Descriptions

36.2.1 CRCCR0 : CRC Control Register 0

Base address: CRC0 = 0x8000_4000
CRC1 = 0x8100_3000

Offset address: 0x0

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DORCLR	LMS	—	—	—	GPS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	GPS[2:0]	CRC Generating Polynomial Switching 0 0 0: No calculation is executed. 0 0 1: 8-bit CRC-8 ($X^8 + X^2 + X + 1$) 0 1 0: 16-bit CRC-16 ($X^{16} + X^{15} + X^2 + 1$) 0 1 1: 16-bit CRC-CCITT ($X^{16} + X^{12} + X^5 + 1$) 1 0 0: 32-bit CRC-32 ($X^{32} + X^{26} + X^{23} + X^{22} + X^{16} + X^{12} + X^{11} + X^{10} + X^8 + X^7 + X^5 + X^4 + X^2 + X + 1$) 1 0 1: 32-bit CRC-32C ($X^{32} + X^{28} + X^{27} + X^{26} + X^{25} + X^{23} + X^{22} + X^{20} + X^{19} + X^{18} + X^{14} + X^{13} + X^{11} + X^{10} + X^9 + X^8 + X^6 + 1$) Others: No calculation is executed.	R/W
5:3	—	These bits are read as 0. The write value should be 0.	R/W
6	LMS	CRC Calculation Switching 0: Generates CRC for LSB first communication. 1: Generates CRC for MSB first communication.	R/W
7	DORCLR	CRCDOR Register Clear 0: No effect 1: Clears the CRCDOR register	W

GPS[2:0] bits (CRC Generating Polynomial Switching)

Set these bits to select the CRC Generating Polynomial.

LMS bit (CRC Calculation Switching)

The setting this bit selects the order of the bits of generated CRC codes. The bit selects transmission of the lower-order byte of the CRC code first for LSB first communication, or the higher-order byte first for MSB first communication. For details on the transmission and reception of CRC codes, see section 36.3. Operation.

DORCLR bit (CRCDOR Register Clear)

Write 1 to this bit so that the CRCDOR register is set to 0x00000000.

36.2.2 CRCDIR/CRCDIR_BY : CRC Data Input Register

Base address: CRC0 = 0x8000_4000
CRC1 = 0x8100_3000

Offset address: 0x4

Bit position:	31	0
Bit field:		
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	n/a	CRC Data Input	R/W

CRCDIR or CRCDIR_BY is written with data for the CRC calculation.

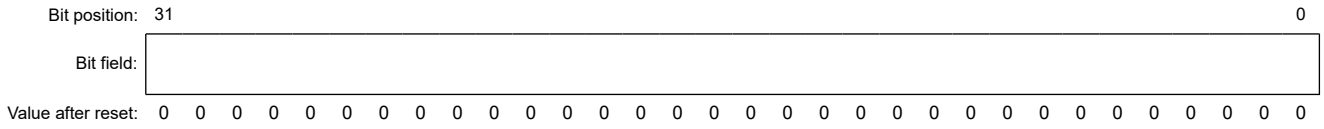
CRCDIR is the register symbol for R/W access in 32-bit units. When generating a 32-bit CRC code, the input data should be written to CRCDR.

CRCDIR_BY is the register symbol for R/W access in 8-bit units. When generating an 8- or 16-bit CRC code, the input data should be written to CRCDID_BY.

36.2.3 CRCDOR/CRCDOR_HA/CRCDOR_BY : CRC Data Output Register

Base address: CRC0 = 0x8000_4000
CRC1 = 0x8100_3000

Offset address: 0x8



Bit	Symbol	Function	R/W
31:0	n/a	CRC Data Output	R/W

When calculating with the initial value set to a value other than 0x00000000, rewrite the CRCDOR register.

The CRC code should be written to the CRCDIR register following the communication data. If the calculation result is 0x00000000, there is no error in the communication data.

If a 32-bit CRC is selected, the CRCDOR register should be accessed. If a 16-bit CRC is selected, the CRCDOR_HA register should be accessed. If an 8-bit CRC is selected, the CRCDOR_BY register should be accessed.

36.3 Operation

The CRC calculator generates CRC codes for use in LSB first or MSB first communication.

The following shows examples of generating the CRC code for input data (0xF0) using the 16-bit CRC-CCITT generating polynomial ($X^{16} + X^{12} + X^5 + 1$). In these examples, the value of the CRC Data Output Register (CRCDOR_HA) is cleared before the CRC calculation (when a 16-bit CRC is in use, bits [31:16] of CRCDOR are not updated).

When an 8-bit CRC is in use, the valid bits of the CRC code is obtained in the lower-order byte of CRCDOR. When a 32-bit CRC is in use, the valid bits of the CRC code is obtained in bits [31:0] of CRCDOR.

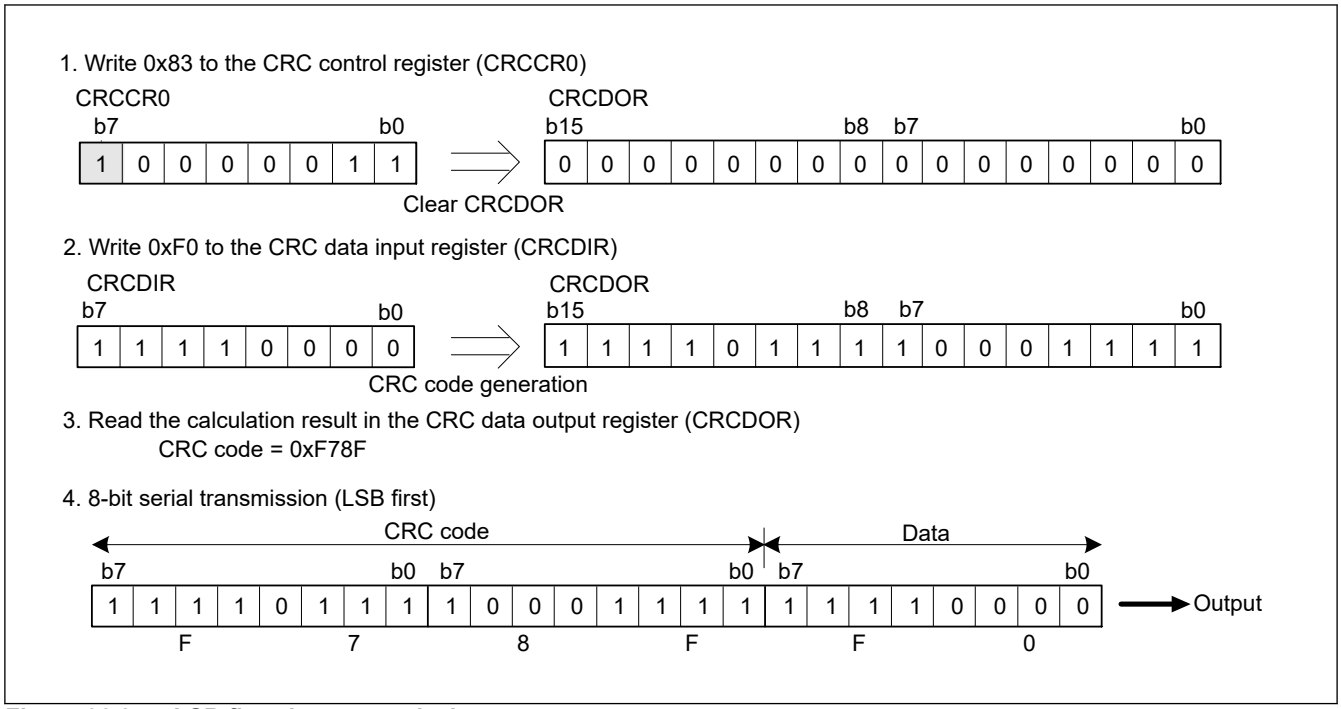


Figure 36.2 LSB first data transmission

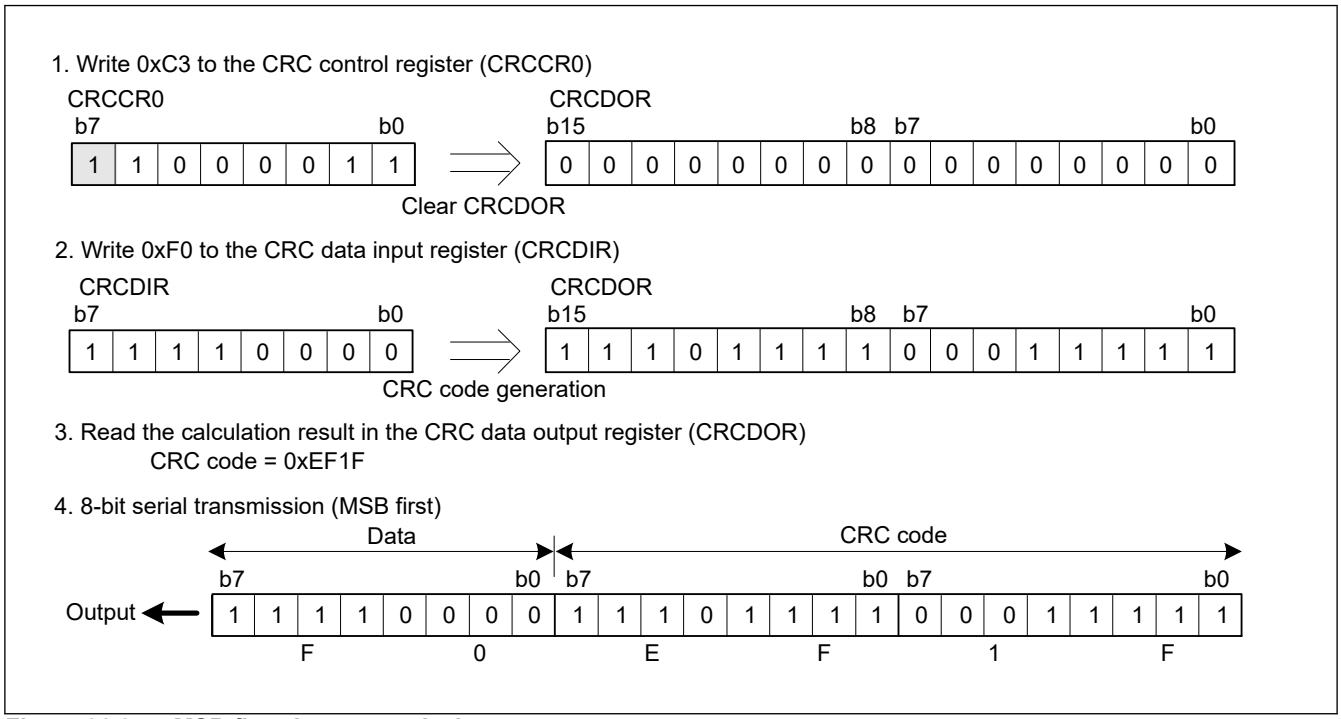


Figure 36.3 MSB first data transmission

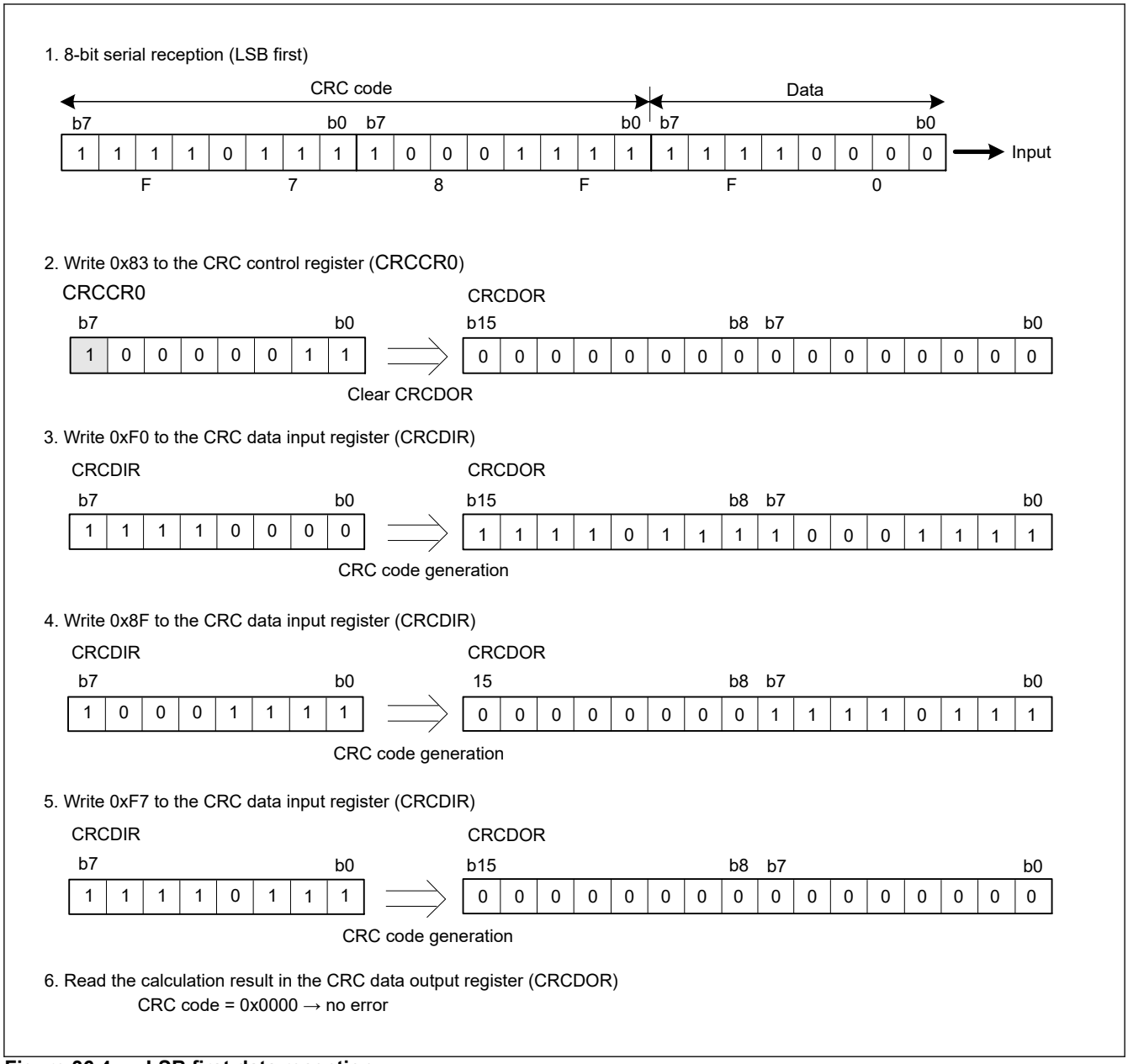


Figure 36.4 LSB first data reception

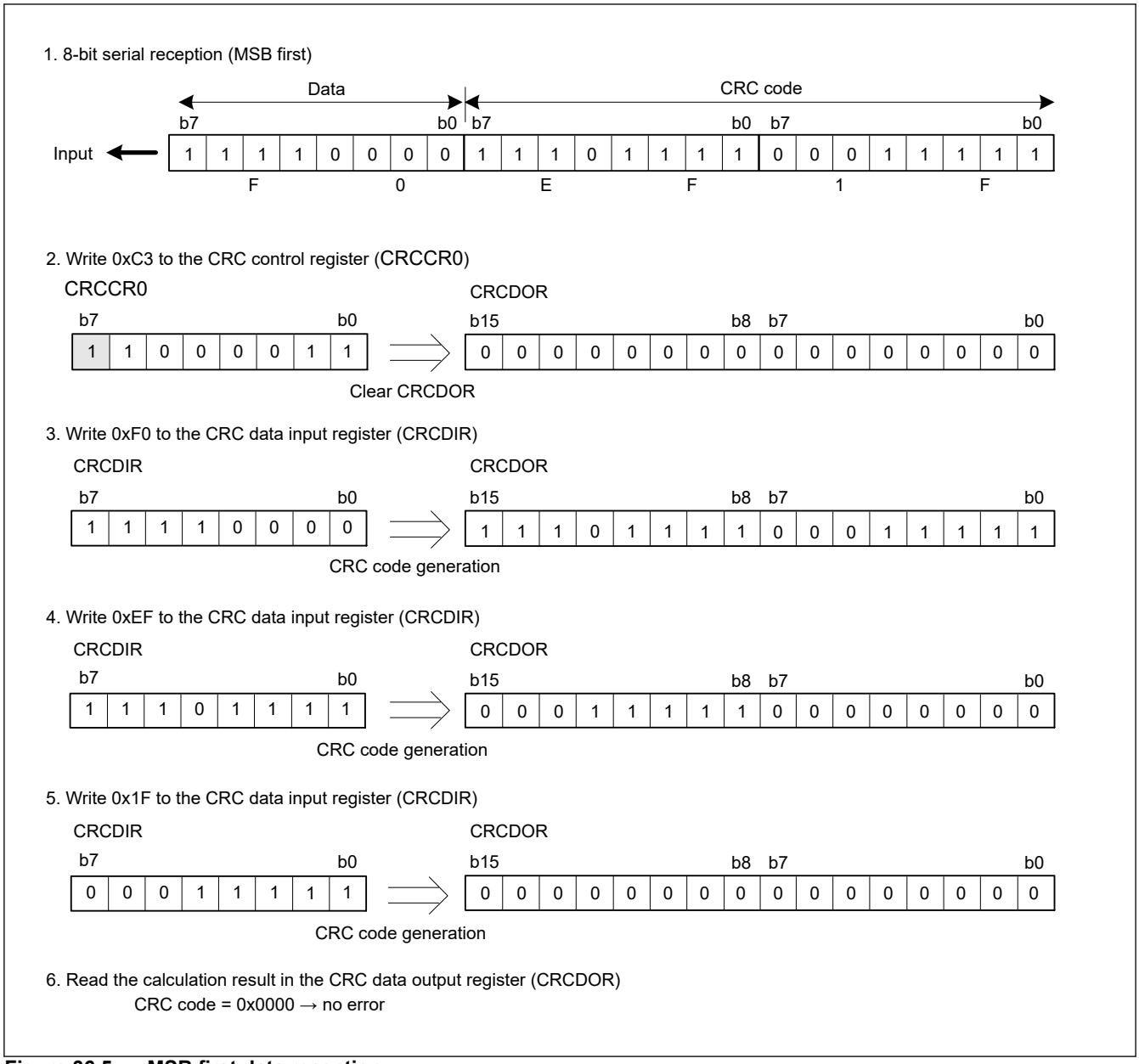


Figure 36.5 MSB first data reception

36.4 Usage Notes

36.4.1 Module-Stop Function Setting

Operation of the CRC calculator can be disabled or enabled using the Module Stop Control Register D and G (MSTPCRD and MSTPCRG). After a reset, the CRC is in the module-stop state. Register access is enabled by releasing the module-stop state.

For details, see section 8, Low-Power Consumption Function.

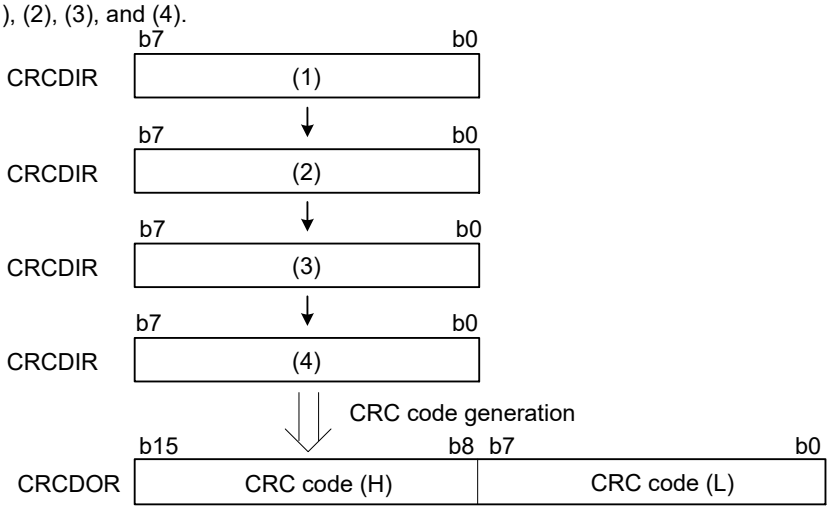
36.4.2 Note on Transmission

Note that the sequence of transmission for the CRC code differs according to whether transmission is LSB first or MSB first.

When transmitting 32-bit data (in case operation executed on 8 bits in parallel)

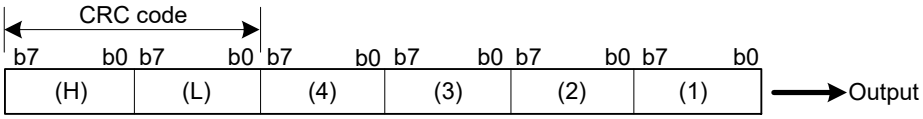
1. CRC code

After specifying the method for generation calculation, write data to the CRCDIR register in the order of (1), (2), (3), and (4).



2. Transmit data

(i) When transmission is LSB first



(ii) When transmission is MSB first

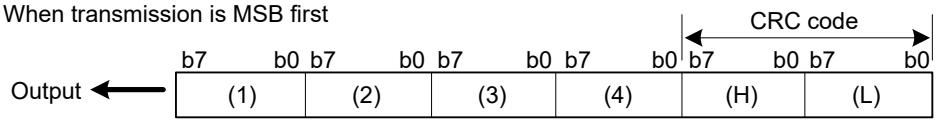


Figure 36.6 LSB first and MSB first data transmission

37. Boundary Scan

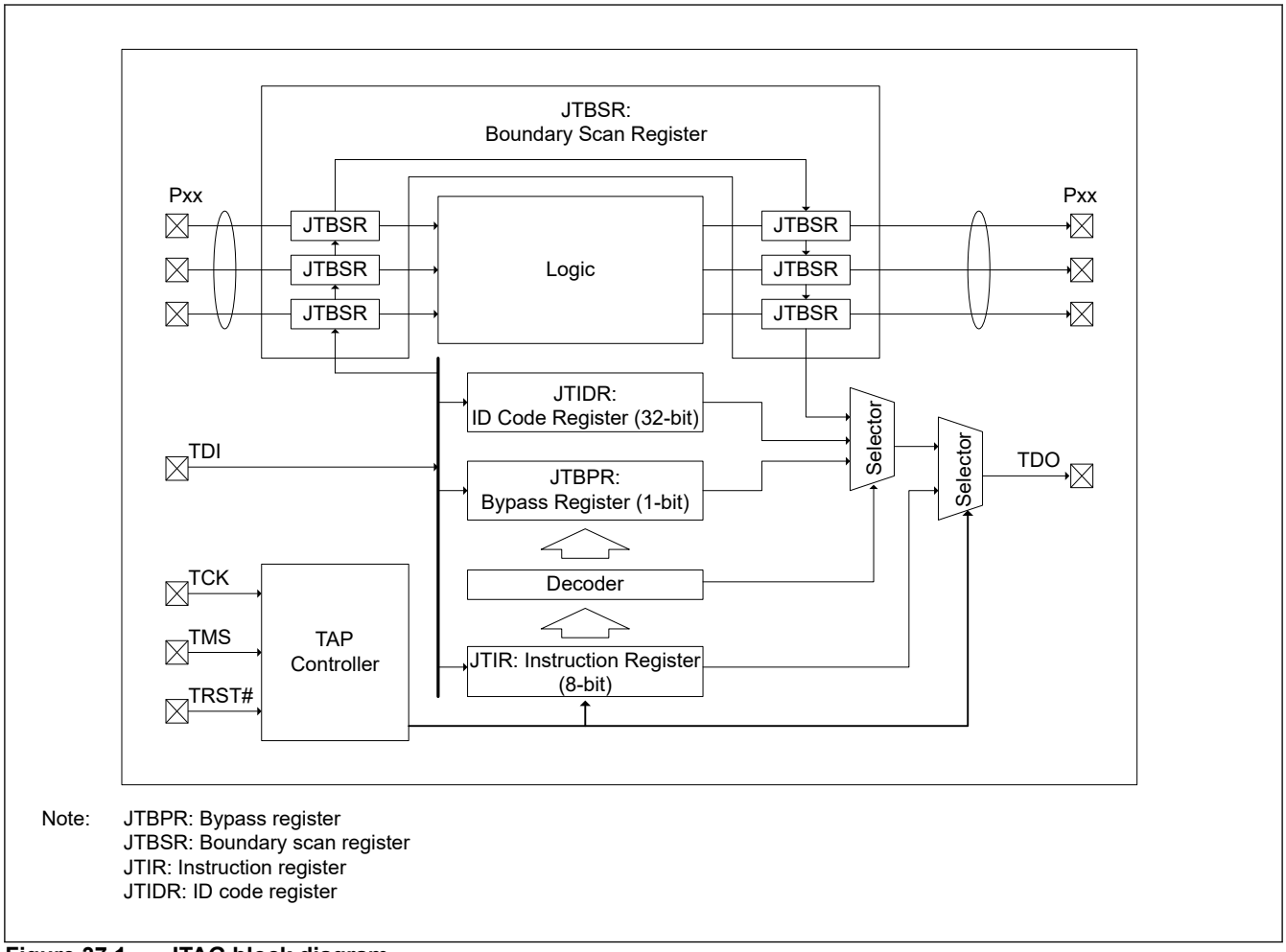
37.1 Overview

The LSI has a boundary scan function. The boundary scan is a serial I/O interface based on the JTAG (Joint Test Action Group), IEEE Std.1149.1, and IEEE Standard Test Access Port and Boundary-Scan Architecture.

Table 37.1 lists the specifications of boundary scan and Figure 37.1 shows a block diagram of the boundary scan function.

Table 37.1 Boundary scan specifications

Parameter	Specifications
Boundary scan enabled/disabled	Boundary scan function is enabled when the BSCANP pin is driven high.
Dedicated boundary scan pins	The TDO, TCK, TDI, TMS, and TRST# pins are dedicated for the JTAG when the boundary scan function is enabled.
Six test modes	• BYPASS mode • EXTEST mode • SAMPLE/PRELOAD mode • CLAMP mode • HIGHZ mode • IDCODE mode



Note: JTBPR: Bypass register
JTBSR: Boundary scan register
JTIR: Instruction register
JTIDR: ID code register

Figure 37.1 JTAG block diagram

Table 37.2 shows the I/O pins used in the boundary scan function.

Table 37.2 Pin configuration of JTAG

Pin name	I/O	Description
TCK	Input	Test clock input pin Clock signal for boundary scan. Input the clock the duty cycle of which is 50% when boundary scan function is used.
TMS	Input	Test mode select pin
TDI	Input	Test data input pin
TDO	Output	Test data output pin
TRST#	Input	Test reset input pin

37.2 Register Descriptions

Instructions can be input to the instruction register via the TDI pin by serial transfer.

The bypass register, which is a 1-bit register, is connected between the TDI and TDO pins in BYPASS mode.

The boundary scan register, which is configured according to the BSDL description, is connected between the TDI and TDO pins when test data are being shifted in.

None of the registers is accessible from the CPU. Table 37.3 shows the availability of serial transfer for the registers.

Table 37.3 Serial transfer for the registers

Register	Serial input	Serial output
Instruction register	Available	Available
ID code register	Available	Available
Bypass register	Available	Available
Boundary scan register	Available	Available

Note: Any serial transfer is available if standards for the boundary scan function are satisfied.

37.2.1 JTIR : Instruction Register

Base address: —

Offset address: —

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TS[7:0]							
Value after reset:	0	1	0	1	0	1	0	1

Bit	Symbol	Function	R/W
7:0	TS[7:0]	Test Bit Set The command configuration is as shown in Table 37.4.	—

The JTIR register is an 8-bit register.

Boundary scan instructions can be transferred to the instruction register by serial input from the TDI pin.

The instruction register is initialized when the TRST# pin is driven to the low level or when the TAP controller is in the Test-Logic-Reset state.

Table 37.4 Command configuration (1 of 2)

TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0	Instruction
0	0	0	0	0	0	0	0	EXTEST
0	1	0	0	0	0	0	0	SAMPLE/PRELOAD
0	1	0	1	0	1	0	1	IDCODE (value after reset)

Table 37.4 Command configuration (2 of 2)

TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0	Instruction
1	1	0	1	0	0	0	0	CLAMP
1	0	0	0	0	0	0	0	HIGHZ
1	1	1	1	1	1	1	1	BYPASS
Other than above								Reserved

37.2.2 JTIDR : ID Code Register

Base address: —

Offset address: —

Bit position:	31																												0
Bit field:	DID[31:0]																												
Value after reset:	0	0	0	0	1	0	0	0	0	1	0	1	1	1	0	1	1	0	1	1	0	1	0	0	0	1	0	0	0	1	1	1

Bit	Symbol	Function	R/W
31:0	DID[31:0]	This register has the fixed value that indicates the device IDCODE.	—

The JTIDR register is a 32-bit register.

ID code register data is output from the TDO pin when the IDCODE instruction is executed.

37.2.3 JTBPR : Bypass Register

The JTBPR register is a 1-bit register.

This register is connected between the TDI and TDO pins when BYPASS mode is set.

The JTBPR register cannot be read from or written to by the CPU.

37.2.4 JTBSR : Boundary Scan Register

The JTBSR register is a shift register to control the external input and output pins of this LSI and is distributed across the pad.

The EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ instructions are issued to apply the JTBSR register in boundary-scan testing.

Refer to the BSDL file for the correspondences between the pins of this LSI and the bits of the boundary scan register.

The value is undefined after a reset.

37.3 Operation

The boundary scan function is valid when the RES# and BSCANP pins are driven high.

37.3.1 TAP Controller

Figure 37.2 shows the state transition diagram of the TAP controller. Table 37.5 describes each state.

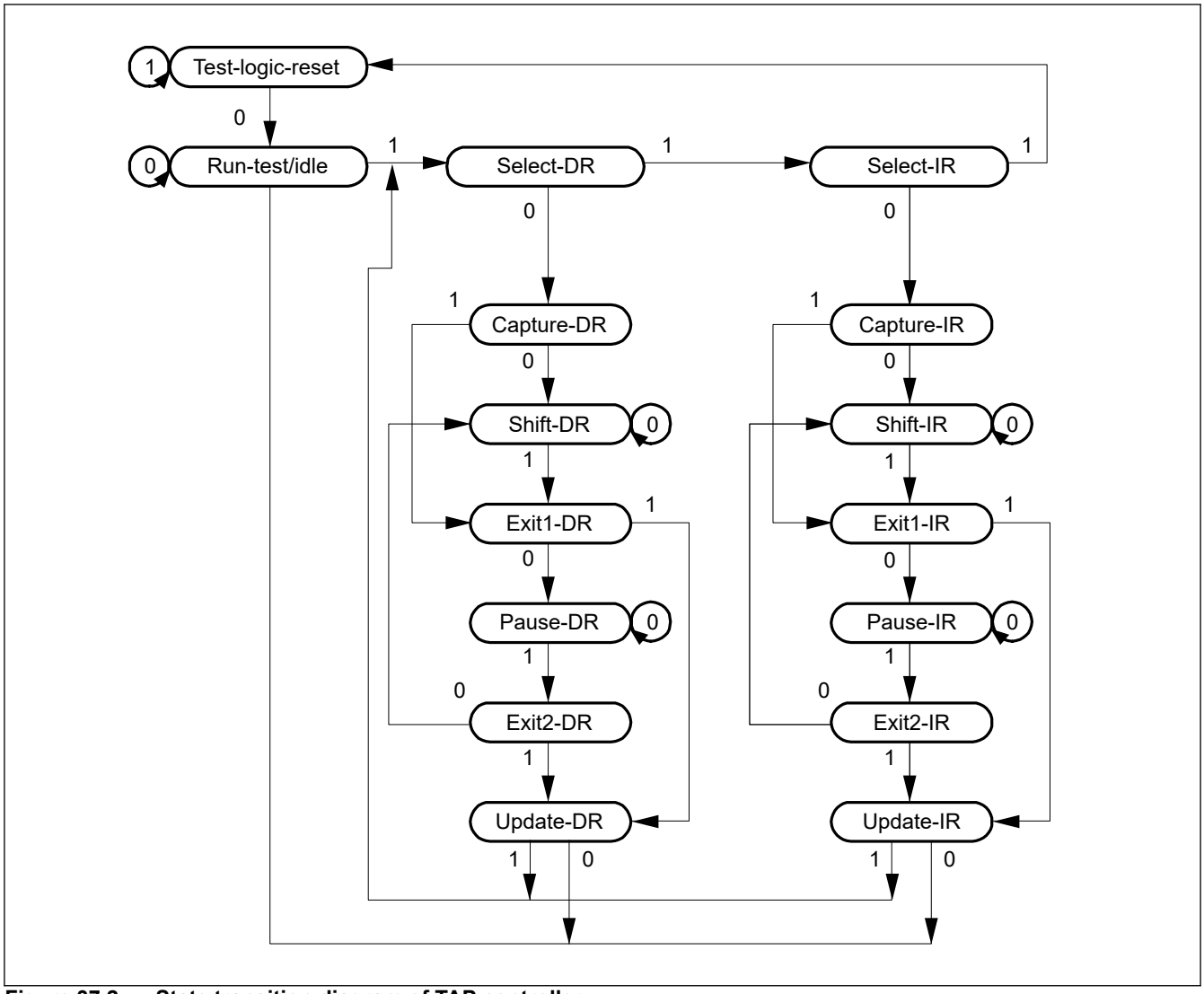


Figure 37.2 State transition diagram of TAP controller

Table 37.5 Explanation of states (1 of 2)

State	Explanation
Test Logic Reset	Reset state of the TAP controller The LSI is in this state during normal operation.
Run Test/Idle	Test execution state
Select DR Scan	Temporary state used for selecting a data register
Select IR Scan	Temporary state used for selecting an instruction register
Capture DR	Data of the test data register corresponding to the current instruction is captured in parallel.
Shift DR	The test data register corresponding to the current instruction is connected between the TDI and TDO pins, and data is transferred serially.
Exit DR	Temporary state
Pause DR	Clocks are applied while the value input in the Shift DR state is retained.
Exit2 DR	Temporary state
Update DR	Output latches of the test data register corresponding to the current instruction are updated.
Capture IR	A fixed value is input to the instruction register.
Shift IR	The instruction register is connected between the TDI and TDO pins, and data is transferred serially.
Exit IR	Temporary state

Table 37.5 Explanation of states (2 of 2)

State	Explanation
Pause IR	Clocks are applied while the value input in the Shift IR state is retained.
Exit2 IR	Temporary state
Update IR	The current instruction is updated to the instruction input in the Shift IR state.

37.3.2 List of Commands

(1) BYPASS [Instruction Code: 1111 1111b]

The BYPASS instruction drives the bypass register. This instruction shortens the shift path, facilitating the transfer of serial data to other LSIs on a printed-circuit board at higher speeds. While this instruction is being executed, the test circuit does not affect the system circuit.

The bypass register is connected between the TDI and TDO pins. Bypass operation is initiated from shift-DR operation. The TDO is set to 0 in the first clock cycle in the Shift-DR state. In the subsequent clock cycles, the TDI signal is output on the TDO pin.

(2) EXTEST [Instruction Code: 0000 0000b]

The EXTEST instruction is used to test external circuits when this LSI is installed on the printed circuit board. If this instruction is executed, output pins are used to output test data (specified by the SAMPLE/PRELOAD instruction) from the boundary scan register to the print circuit board, and input pins are used to input test result to the boundary scan register from the print circuit board.

(3) SAMPLE/PRELOAD [Instruction Code: 0100 0000b]

The SAMPLE/PRELOAD instruction is used to input data from this LSI's internal circuits to the boundary scan register, output data from the scan path, and reload the data to the scan path. While this instruction is executed, input signals are directly input to this LSI and output signals are also directly output to the external circuits. The system circuit of this LSI is not affected by this instruction.

In SAMPLE operation, the boundary scan register latches the snapshot of data transferred from input pins to internal circuit or data transferred from internal circuit to output pins. The latched data is read from the scan path. The scan register latches the snapshot at the rising edge of the TCK in Capture-DR state. The scan register latches snapshots without affecting the LSI normal operation.

In PRELOAD operation, the value after reset is written from the scan path to the parallel output latch of the boundary scan register prior to the EXTEST instruction execution. If the EXTEST is executed without executing this PRELOAD operation, undefined values are output from the beginning to the end (transfer to the output latch) of the EXTEST sequence. (In the EXTEST instruction, output parallel latches are always output to the output pins.)

(4) IDCODE [Instruction Code: 0101 0101b]

When the IDCODE instruction is selected, the ID code register value is output from the TDO in LSB-first order in Shift-DR state of the TAP controller. While this instruction is being executed, the test circuit does not affect the system circuit. The instruction register is initialized by the IDCODE instruction in Test-Logic-Reset state of the TAP controller.

(5) CLAMP [Instruction Code: 1101 0000b]

When the CLAMP instruction is selected, output pins output the boundary scan register value which was specified by the SAMPLE/PRELOAD instruction in advance. While the CLAMP instruction is selected, the status of the boundary scan register is maintained regardless of the TAP controller state.

The bypass register is connected between the TDI and TDO pins, leading to the same operation as when BYPASS mode has been selected.

(6) HIGHZ [Instruction Code: 1000 0000b]

When the HIGHZ instruction is selected, all output pins enter high-impedance state. While the HIGHZ instruction is selected, the status of the boundary scan register is maintained regardless of the TAP controller state. The bypass register is connected between the TDI and TDO pins, leading to the same operation as when the BYPASS instruction has been selected.

37.4 Usage Notes

1. For serial transfer, data are input or output in LSB-first order. See Figure 37.3.
2. Pins of the boundary scan (TDI, TMS, and TRST#) have to be pulled up by pull-up resistors.
   However, if an on-chip emulator is in use, handle the TRST# pin according to the manual for the given on-chip emulator.
3. Power supply pins (VDD, VSS, VCC1833_n, VCC33, VCC18_PLL0, VCC18_PLL1, AVCC18_TSU, VCC33_USB, VCC18_USB, VSS_USB, AVCC18_USB, VCC18_ADC0, VCC18_ADC1) cannot be boundary-scanned.
4. Analog reference pins (VREFH0, VREFH1, and USB_RREF) cannot be boundary-scanned.
5. Analog input pins (AN000 to AN007 and AN100 to AN115) cannot be boundary-scanned.
6. Clock pins (EXTAL, XTAL, and EXTCLKIN) cannot be boundary-scanned.
7. The reset pin (RES#) cannot be boundary-scanned.
8. The operating mode input pin (MDX) cannot be boundary-scanned.
9. USB dedicated pins (USB_DP and USB_DM) cannot be boundary-scanned.
10. The boundary-scan pin (BSCANP) cannot be boundary-scanned.
11. The boundary-scan pins (TCK, TMS, TRST#, TDI, and TDO) cannot be boundary-scanned.
12. The boundary scan function is not available when the chip is in the reset state.
13. For a pin that incorporates open-drain functionality and for which the open-drain function is enabled, if the boundary-scan function sets the corresponding bit in the output scan register and output enable register to 1, executing an EXTEST, CLAMP, or SAMPLE/PRELOAD instruction makes the pin output the high level rather than placing it in the high-impedance state.*1

Note 1. For details on boundary-scan target pins, refer to the BSDL file. Boundary-scan is not applicable to some pins and power pins.

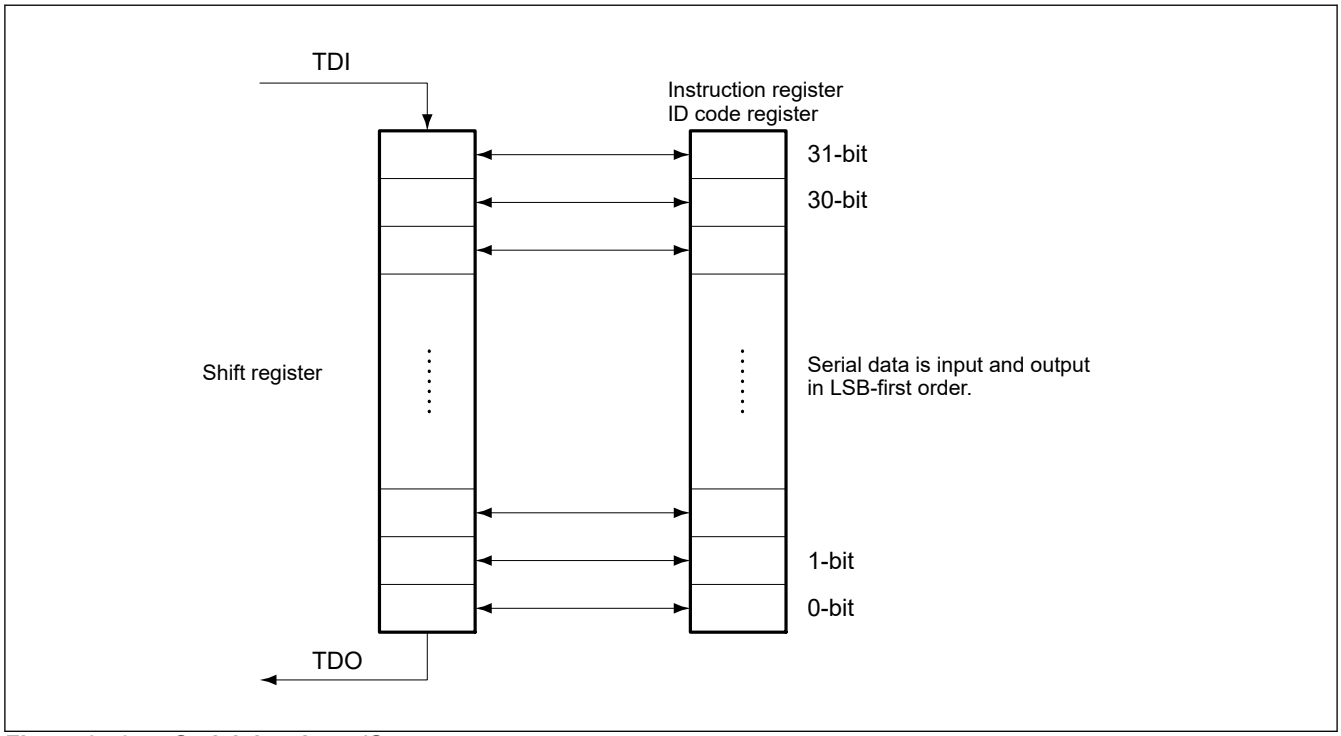


Figure 37.3 Serial data Input/Output

38. ΔΣ Interface (DSMIF)

38.1 Overview

The ΔΣ interfaces (DSMIF) can be connected with up to three external ΔΣ modulators. It is capable of filtering delta-sigma modulated 1-bit digital input data and converting it into 16-bit digital data.

Table 38.1 lists the specifications of DSMIF and Table 38.2 lists the functions of DSMIF. Figure 38.1 shows a block diagram of DSMIF.

Table 38.1 DSMIF specifications

Parameter	Description
Number of units	2 units
Number of channels	3 channels per unit for a total of 6 channels
Decimation filter characteristics	Filter are selectable as Sinc 1st order, Sinc 2nd order, Sinc 3rd order Can select each channel.
Over-sampling rate (decimation rate)	Selectable from 4 to 256 Can select each channel.
Interface clock	Support Master mode (MCLK is used as output)/Slave mode (MCLK is used as input) by channel Master mode supports the following output clock: ● 25/20/12.5/10/6.25/5 MHz (PCLKH 200 MHz) ● 25/18.75/12.5/9.375/6.25/5 MHz (PCLKH 150 MHz) Filter received clock inversion function.
Data capture	Crests and troughs can trigger capture, allowing capture of values for current with the same timing. Can select one capture trigger from six triggers for each trigger input by channel. The PWM synchronization function is realized by initializing the decimation counter for the differential stage of the SINC filter with the trigger output from GPT.
Unit synchronization	Not supported
Reset function of differential filter clock divider	Support channel by channel. Can select one from three triggers.
Error source ouput (interrupt)	12 kinds of interrupt sources in 1 unit: ● Current data register update interrupt ● Current capture data register A update interrupt ● Current capture data register B update interrupt ● Short-circuit detection error interrupt ● Lower threshold overcurrent detection 0 error interrupt ● Upper threshold overcurrent detection 0 error interrupt ● Lower threshold overcurrent detection 1 error interrupt ● Upper threshold overcurrent detection 1 error interrupt ● Lower threshold overcurrent detection 2 error interrupt ● Upper threshold overcurrent detection 2 error interrupt ● Lower threshold current detection interrupt ● Upper threshold detection interrupt
Low-power consumption function	The module-stop state can be set (clock gating function is implemented)

Table 38.2 DSMIF functions (1 of 3)

Parameter			Unit 0 (DSMIF0)	Unit 1 (DSMIF1)
I/O pin	Clock	Clock I/O pin DSMIF0: ch0-2 DSMIF1: ch3-5	MCLK0 MCLK1 MCLK2	MCLK3 MCLK4 MCLK5
	Data	DATA input pin DSMIF0: ch0-2 DSMIF1: ch3-5	MDAT0 MDAT1 MDAT2	MDAT3 MDAT4 MDAT5

Table 38.2 DSMIF functions (2 of 3)

Parameter			Unit 0 (DSMIF0)	Unit 1 (DSMIF1)
Interrupt source	Normal	Current data register update (Combined with ch0 to ch2 for DSMIF0, combined with ch3 to ch5 for DSMIF1)	DSMIF0_CDRUI	DSMIF1_CDRUI
		Current data register update common	DSMIF0_CDUPDC	DSMIF1_CDUPDC
		Current capture data register A update common	DSMIF0_CDACUPDC	DSMIF1_CDACUPDC
		Current capture data register B update common	DSMIF0_CDBCUPDC	DSMIF1_CDBCUPDC
	Error	Lower threshold current sum error*1	DSMIF0_LTCSE	DSMIF1_LTCSE
		Upper threshold current sum error*1	DSMIF0_UTCSE	DSMIF1_UTCSE
		Short-circuit detection 0*1	DSMIF0_SCDE0	DSMIF1_SCDE0
		Short-circuit detection 1*1	DSMIF0_SCDE1	DSMIF1_SCDE1
		Short-circuit detection 2*1	DSMIF0_SCDE2	DSMIF1_SCDE2
		Lower threshold overcurrent detection 0 (ch0/ch3)*1	DSMIF0_OVC0EL0	DSMIF1_OVC0EL0
		Lower threshold overcurrent detection 0 (ch1/ch4)*1	DSMIF0_OVC0EL1	DSMIF1_OVC0EL1
		Lower threshold overcurrent detection 0 (ch2/ch5)*1	DSMIF0_OVC0EL2	DSMIF1_OVC0EL2
		Upper threshold overcurrent detection 0 (ch0/ch3)*1	DSMIF0_OVC0EH0	DSMIF1_OVC0EH0
		Upper threshold overcurrent detection 0 (ch1/ch4)*1	DSMIF0_OVC0EH1	DSMIF1_OVC0EH1
		Upper threshold overcurrent detection 0 (ch2/ch5)*1	DSMIF0_OVC0EH2	DSMIF1_OVC0EH2
		Lower threshold overcurrent detection 1 (ch0/ch3)	DSMIF0_OVC1EL0	DSMIF1_OVC1EL0
		Lower threshold overcurrent detection 1 (ch1/ch4)	DSMIF0_OVC1EL1	DSMIF1_OVC1EL1
		Lower threshold overcurrent detection 1 (ch2/ch5)	DSMIF0_OVC1EL2	DSMIF1_OVC1EL2
		Upper threshold overcurrent detection 1 (ch0/ch3)	DSMIF0_OVC1EH0	DSMIF1_OVC1EH0
		Upper threshold overcurrent detection 1 (ch1/ch4)	DSMIF0_OVC1EH1	DSMIF1_OVC1EH1
		Upper threshold overcurrent detection 1 (ch2/ch5)	DSMIF0_OVC1EH2	DSMIF1_OVC1EH2
		Lower threshold overcurrent detection 2 (ch0/ch3)	DSMIF0_OVC2EL0	DSMIF1_OVC2EL0
		Lower threshold overcurrent detection 2 (ch1/ch4)	DSMIF0_OVC2EL1	DSMIF1_OVC2EL1
		Lower threshold overcurrent detection 2 (ch2/ch5)	DSMIF0_OVC2EL2	DSMIF1_OVC2EL2
		Upper threshold overcurrent detection 2 (ch0/ch3)	DSMIF0_OVC2EH0	DSMIF1_OVC2EH0
		Upper threshold overcurrent detection 2 (ch1/ch4)	DSMIF0_OVC2EH1	DSMIF1_OVC2EH1
		Upper threshold overcurrent detection 2 (ch2/ch5)	DSMIF0_OVC2EH2	DSMIF1_OVC2EH2

Table 38.2 DSMIF functions (3 of 3)

Parameter			Unit 0 (DSMIF0)	Unit 1 (DSMIF1)
Interrupt source	Error	Overcurrent detection window notification 0 (ch0/ch3)	DSMIF0_OVC0WN0	DSMIF1_OVC0WN0
		Overcurrent detection window notification 1 (ch0/ch3)	DSMIF0_OVC1WN0	DSMIF1_OVC1WN0
		Overcurrent detection window notification 2 (ch0/ch3)	DSMIF0_OVC2WN0	DSMIF1_OVC2WN0
		Overcurrent detection window notification 3 (ch0/ch3)*2	DSMIF0_OVC3WN0	DSMIF1_OVC3WN0
		Overcurrent detection window notification 0 (ch1/ch4)	DSMIF0_OVC0WN1	DSMIF1_OVC0WN1
		Overcurrent detection window notification 1 (ch1/ch4)	DSMIF0_OVC1WN1	DSMIF1_OVC1WN1
		Overcurrent detection window notification 2 (ch1/ch4)	DSMIF0_OVC2WN1	DSMIF1_OVC2WN1
		Overcurrent detection window notification 3 (ch1/ch4)*2	DSMIF0_OVC3WN1	DSMIF1_OVC3WN1
		Overcurrent detection window notification 0 (ch2/ch5)	DSMIF0_OVC0WN2	DSMIF1_OVC0WN2
		Overcurrent detection window notification 1 (ch2/ch5)	DSMIF0_OVC1WN2	DSMIF1_OVC1WN2
		Overcurrent detection window notification 2 (ch2/ch5)	DSMIF0_OVC2WN2	DSMIF1_OVC2WN2
		Overcurrent detection window notification 3 (ch2/ch5)*2	DSMIF0_OVC3WN2	DSMIF1_OVC3WN2
Trigger input ELC destination	Capture	Current data capture trigger 0	DSMIF0 CAP_TRG0	DSMIF1 CAP_TRG0
		Current data capture trigger 1	DSMIF0 CAP_TRG1	DSMIF1 CAP_TRG1
		Current data capture trigger 2	DSMIF0 CAP_TRG2	DSMIF1 CAP_TRG2
		Current data capture trigger 3	DSMIF0 CAP_TRG3	DSMIF1 CAP_TRG3
		Current data capture trigger 4	DSMIF0 CAP_TRG4	DSMIF1 CAP_TRG4
		Current data capture trigger 5	DSMIF0 CAP_TRG5	DSMIF1 CAP_TRG5
	Counter Reset	Decimation dividing counter initialization trigger 0	DSMIF0 CDCNT_INT_TRG0	DSMIF1 CDCNT_INT_TRG0
		Decimation dividing counter initialization trigger 1	DSMIF0 CDCNT_INT_TRG1	DSMIF1 CDCNT_INT_TRG1
		Decimation dividing counter initialization trigger 2	DSMIF0 CDCNT_INT_TRG2	DSMIF1 CDCNT_INT_TRG2

Note 1. binded unit error signal connected to POE3 and POEG as error 0
Note 2. binded unit error signal connected to POE3 and POEG as error 1

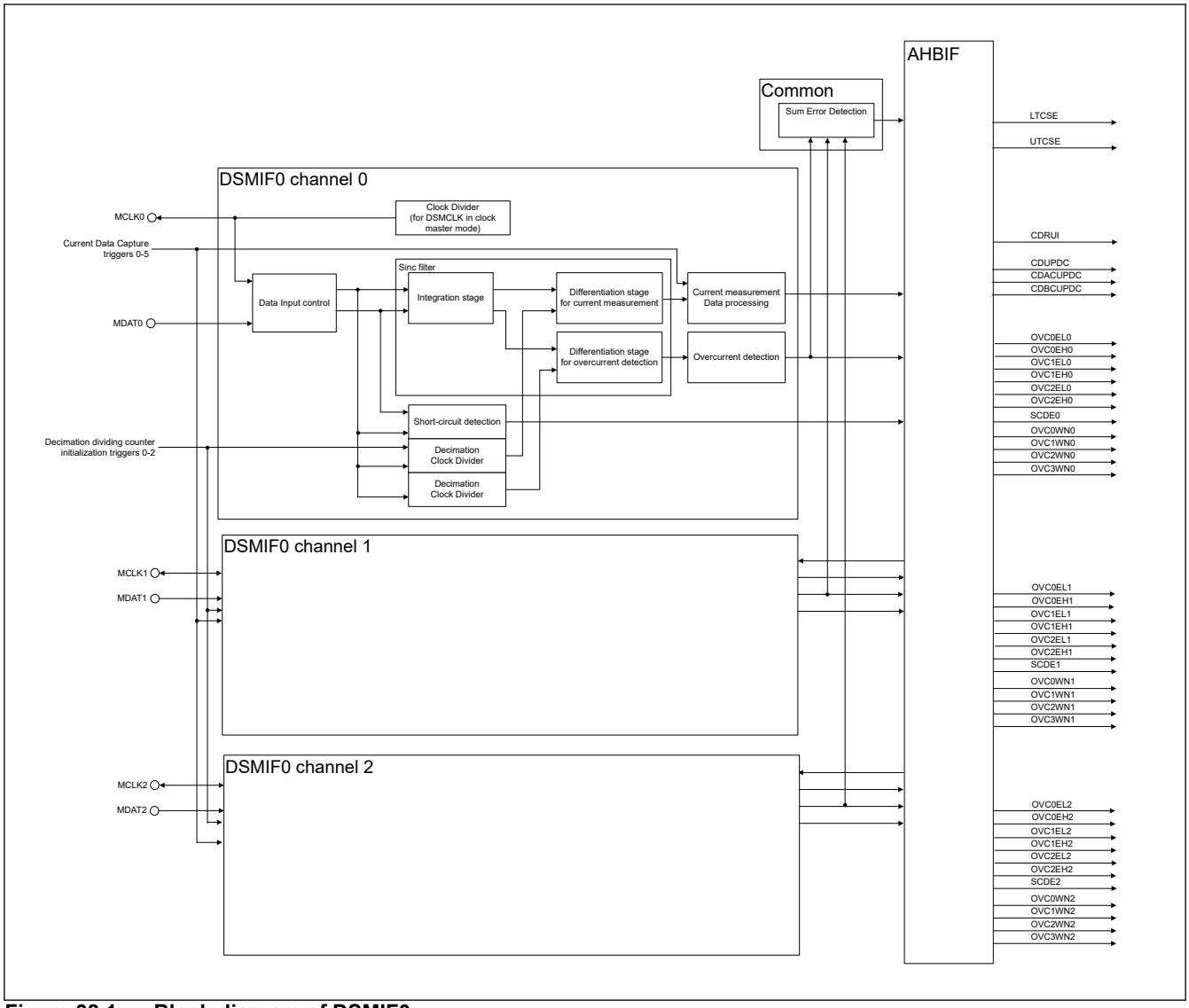


Figure 38.1 Block diagram of DSMIF0

38.2 Register Descriptions

38.2.1 DSMIF Common Control Registers

38.2.1.1 DSSEICR : Overcurrent Sum Error Detect Interrupt Control Register

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x080

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ISEH	ISEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ISEL	Overcurrent sum error lower limit detection interrupt enable bit 0: Do not output overcurrent sum error lower limit detection interrupt 1: Overcurrent sum error lower limit detection interrupt can be output	R/W
1	ISEH	Overcurrent sum error upper limit detection interrupt enable bit 0: Do not output overcurrent sum error upper limit detection interrupt 1: Overcurrent sum error lower upper detection interrupt can be output	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

38.2.1.2 DSSECSR : Overcurrent Sum Error Detect Channel Setting Register

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x084

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SEDM[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SEDM[2:0]	Overcurrent sum error detect mode setting bit 0 0 0: Detects error of sum value of overcurrent data of CH0, CH1, CH2 0 0 1: Detects error of sum value of overcurrent data of CH0, CH1 0 1 0: Detects error of overcurrent data of CH0 0 1 1: Detects error of overcurrent data of CH1 1 0 0: Detects error of overcurrent data of CH2 Others: Setting prohibited (detects error of sum value of overcurrent data of CH0, CH1, CH2)	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

38.2.1.3 DSSELTR : Overcurrent Sum Error Detect Low Threshold Register

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x088

Bit position:	31														17																0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCMPTBL[17:0]																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
17:0	SCMPTBL[17:0]	Overcurrent sum error detect lower limit These bits set the lower limit of the active current value for overcurrent sum error detection. Set these bits to the lower limit of the sum value of the overcurrent data for each channel.	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

38.2.1.4 DSSEHTR : Overcurrent Sum Error Detect High Threshold Register

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x08C

Bit position:	31														17															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SCMPTBH[17:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
17:0	SCMPTBH[17:0]	Overcurrent sum error detect upper limit These bits set the upper limit of the active current value for overcurrent sum error detection. Set these bits to the upper limit of the sum value of the overcurrent data for each channel.	R/W
31:18	—	These bits are read as 0. The write value should be 0.	R/W

38.2.1.5 DSSECR : Overcurrent Sum Error Detect Control Register

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x090

Bit position:	31																												1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEEH	SEEL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SEEL	Overcurrent sum error lower limit detection enable bit 0: The lower limit exceeded detection function of overcurrent sum error detection is invalid 1: The lower limit exceeded detection function of overcurrent sum error detection is valid	R/W
1	SEEH	Overcurrent sum error upper limit detection enable bit 0: The upper limit exceeded detection function of overcurrent sum error detection is invalid 1: The upper limit exceeded detection function of overcurrent sum error detection is valid	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

38.2.1.6 DSSECDRn : Overcurrent Sum Error Detect Capture Data Register n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x0A0 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SECDR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	SECDR[15:0]	Overcurrent sum error detect capture data n Captures the overcurrent data of the target channel when the upper limit error or the lower limit error of the over sum current error detection occurs. See Table 38.9 for details of the data to be captured.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

38.2.1.7 DSCMSR : Common Mode Setting Register

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x0C0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	CISM[1:0]		DFS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DFS	Data Format Select 0: Left justified 1: Right justified	R/W
2:1	CISM[1:0]	Common Interrupt Synchronous channel Mode 0 0: Common interrupts are not used 0 1: Common interrupts for ch0, ch1, and ch2 (ch3, ch4, and ch5) can be used 1 0: Common interrupts for ch0 and ch1 (ch3 and ch4) can be used 1 1: Setting prohibited	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

38.2.1.8 DSCICR : Common Interrupt Control Register

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x0C4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	IBUCE	IAUCE	IUCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IUCE	Current data register update channel common interrupt enable 0: Do not output capture current data register update channel common interrupt 1: Capture current data register update channel common interrupt can be output	R/W
1	IAUCE	Capture current data register A update channel common interrupt enable 0: Do not output capture current data register A update channel common interrupt 1: Capture current data register A update channel common interrupt can be output	R/W

Bit	Symbol	Function	R/W
2	IBUCE	Capture current data register B update channel common interrupt enable 0: Do not output capture current data register B update channel common interrupt 1: Capture current data register B update channel common interrupt can be output	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

38.2.1.9 DSCSTRTR : Channel Software Start Trigger Register

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x200

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	STRTRG2	STRTRG1	STRTRG0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STRTRG0	Channel 0 start trigger 0: Do nothing 1: Start channel	R/W
1	STRTRG1	Channel 1 start trigger 0: Do nothing 1: Start channel	R/W
2	STRTRG2	Channel 2 start trigger 0: Do nothing 1: Start channel	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

38.2.1.10 DSCSTPTR : Channel Software Stop Trigger Register

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x204

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	STPTRG2	STPTRG1	STPTRG0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STPTRG0	Channel 0 stop trigger 0: Do nothing 1: Stop channel	R/W
1	STPTRG1	Channel 1 stop trigger 0: Do nothing 1: Stop channel	R/W

Bit	Symbol	Function	R/W
2	STPTRG2	Channel 2 stop trigger 0: Do nothing 1: Stop channel	R/W
31:3	—	These bits are read as 0. The write value should be 0.	R/W

38.2.1.11 DSCESR : Channel Error Status Register

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x220

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SUMERRH	SUMERRL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	SCF2	SCF1	SCF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SCF0	Channel 0 short circuit detection flag This bit becomes 1 when 1 is continuously input to the MDAT0 pin, exceeding the number of times set in DSSCTSRCH0.SCNTH, or when 0 is continuously input to the MDAT0 pin, exceeding the number of times set in DSSCTSRCH0.SCNTH. This bit becomes 0 when SCF0 = 1 and writing 1 to the channel 0 short circuit detection flag clear (DSCESCR.CLRSCF0 / DSCSCRCH0.CLRSCF). 0: Indicates that the MDAT0 pin short circuit is not detected 1: Indicates that a short circuit on the MDAT0 pin has been detected	R
1	SCF1	Channel 1 short circuit detection flag This bit becomes 1 when 1 is continuously input to the MDAT1 pin, exceeding the number of times set in DSSCTSRCH1.SCNTH, or when 0 is continuously input to the MDAT1 pin, exceeding the number of times set in DSSCTSRCH1.SCNTH. This bit becomes 0 when SCF1 = 1 and writing 1 to the channel 1 short circuit detection flag clear (DSCESCR.CLRSCF1 / DSCSCRCH1.CLRSCF). 0: Indicates that the MDAT1 pin short circuit is not detected 1: Indicates that a short circuit on the MDAT1 pin has been detected	R
2	SCF2	Channel 2 short circuit detection flag This bit becomes 1 when 1 is continuously input to the MDAT2 pin, exceeding the number of times set in DSSCTSRCH2.SCNTH, or when 0 is continuously input to the MDAT2 pin, exceeding the number of times set in DSSCTSRCH2.SCNTH. This bit becomes 0 when SCF2 = 1 and writing 1 to the channel 2 short circuit detection flag clear (DSCESCR.CLRSCF2 / DSCSCRCH2.CLRSCF). 0: Indicates that the MDAT2 pin short circuit is not detected 1: Indicates that a short circuit on the MDAT2 pin has been detected	R
15:3	—	These bits are read as 0.	R
16	SUMERRL	Overcurrent sum error lower limit detection flag This bit becomes 1 when the sum value of the overcurrent data is updated and this sum value is smaller than the lower limit (DSSELTR.SCMPTBL). This bit becomes 0 when SUMERRL = 1 and writing 1 to the overcurrent sum error lower limit detection flag clear (DSCESCR.CLRSUMERRL). 0: Indicates that the sum value of overcurrent data is not less than the lower limit 1: Indicates that the sum value of overcurrent data is less than the lower limit	R

Bit	Symbol	Function	R/W
17	SUMERRH	Overcurrent sum error upper limit detection flag This bit becomes 1 when the sum value of the overcurrent data is updated and this sum value is larger than the upper limit (DSSEHTR.SCMPTBH). This bit becomes 0 when SUMERRH = 1 and writing 1 to the overcurrent sum error upper limit detection flag clear (DSCESCR.CLRSUMERRH). 0: Indicates that the sum value of overcurrent data is not more than the upper limit 1: Indicates that the sum value of overcurrent data is more than the upper limit	R
31:18	—	These bits are read as 0.	R

38.2.1.12 DSCOCESR : Channel Overcurrent Error Status Register

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x224

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	OC2FH2	OC2FH1	OC2FH0	—	OC2FL2	OC2FL1	OC2FL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	OC1FH2	OC1FH1	OC1FH0	—	OC1FL2	OC1FL1	OC1FL0	—	OC0FH2	OC0FH1	OC0FH0	—	OC0FL2	OC0FL1	OC0FL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OC0FL0	Channel 0 overcurrent lower limit detection 0 flag This bit becomes 1 when the overcurrent data is updated and this data is smaller than the lower limit 0. This bit becomes 0 when OC0FL0 = 1 and writing 1 to the channel 0 overcurrent lower limit detection flag 0 clear (DSCOCESCR.CLROC0FL0 / DSCSCRCH0.CLROC0FL). 0: Indicates that the overcurrent data is not less than the lower limit 0 1: Indicates that the overcurrent data has been less than the lower limit 0	R
1	OC0FL1	Channel 1 overcurrent lower limit detection 0 flag This bit becomes 1 when the overcurrent data is updated and this data is smaller than the lower limit 0. This bit becomes 0 when OC0FL1 = 1 and writing 1 to the channel 1 overcurrent lower limit detection flag 0 clear (DSCOCESCR.CLROC0FL1 / DSCSCRCH1.CLROC0FL). 0: Indicates that the overcurrent data is not less than the lower limit 0 1: Indicates that the overcurrent data has been less than the lower limit 0	R
2	OC0FL2	Channel 2 overcurrent lower limit detection 0 flag This bit becomes 1 when the overcurrent data is updated and this data is smaller than the lower limit 0. This bit becomes 0 when OC0FL2 = 1 and writing 1 to the channel 2 overcurrent lower limit detection flag 0 clear (DSCOCESCR.CLROC0FL2 / DSCSCRCH2.CLROC0FL). 0: Indicates that the overcurrent data is not less than the lower limit 0 1: Indicates that the overcurrent data has been less than the lower limit 0	R
3	—	This bit is read as 0.	R
4	OC0FH0	Channel 0 overcurrent upper limit exceeded 0 flag This bit becomes 1 when the overcurrent data is updated and this data is larger than the upper limit 0. This bit becomes 0 when OC0FH0 = 1 and writing 1 to the channel 0 overcurrent upper limit exceeded flag 0 clear (DSCOCESCR.CLROC0FH0 / DSCSCRCH0.CLROC0FH). 0: Indicates that the overcurrent data doesn't exceed the upper limit 0 1: Indicates that the overcurrent data has exceeded the upper limit 0	R

Bit	Symbol	Function	R/W
5	OC0FH1	Channel 1 overcurrent upper limit exceeded 0 flag This bit becomes 1 when the overcurrent data is updated and this data is larger than the upper limit 0. This bit becomes 0 when OC0FH1 = 1 and writing 1 to the channel 1 overcurrent upper limit exceeded flag 0 clear (DSCOCESCR.CLROC0FH1 / DSCSCRCH1.CLROC0FH). 0: Indicates that the overcurrent data doesn't exceed the upper limit 0 1: Indicates that the overcurrent data has exceeded the upper limit 0	R
6	OC0FH2	Channel 2 overcurrent upper limit exceeded 0 flag This bit becomes 1 when the overcurrent data is updated and this data is larger than the upper limit 0. This bit becomes 0 when OC0FH2 = 1 and writing 1 to the channel 2 overcurrent upper limit exceeded flag 0 clear (DSCOCESCR.CLROC0FH2 / DSCSCRCH2.CLROC0FH). 0: Indicates that the overcurrent data doesn't exceed the upper limit 0 1: Indicates that the overcurrent data has exceeded the upper limit 0	R
7	—	This bit is read as 0.	R
8	OC1FL0	Channel 0 overcurrent lower limit detection 1 flag This bit becomes 1 when the overcurrent data is updated and this data is smaller than the lower limit 1. This bit becomes 0 when OC1FL0 = 1 and writing 1 to the channel 0 overcurrent lower limit detection flag 1 clear (DSCOCESCR.CLROC1FL0 / DSCSCRCH0.CLROC1FL). 0: Indicates that the overcurrent data is not less than the lower limit 1 1: Indicates that the overcurrent data has been less than the lower limit 1	R
9	OC1FL1	Channel 1 overcurrent lower limit detection 1 flag This bit becomes 1 when the overcurrent data is updated and this data is smaller than the lower limit 1. This bit becomes 0 when OC1FL1 = 1 and writing 1 to the channel 1 overcurrent lower limit detection flag 1 clear (DSCOCESCR.CLROC1FL1 / DSCSCRCH1.CLROC1FL). 0: Indicates that the overcurrent data is not less than the lower limit 1 1: Indicates that the overcurrent data has been less than the lower limit 1	R
10	OC1FL2	Channel 2 overcurrent lower limit detection 1 flag This bit becomes 1 when the overcurrent data is updated and this data is smaller than the lower limit 1. This bit becomes 0 when OC1FL2 = 1 and writing 1 to the channel 2 overcurrent lower limit detection flag 1 clear (DSCOCESCR.CLROC1FL2 / DSCSCRCH2.CLROC1FL). 0: Indicates that the overcurrent data is not less than the lower limit 1 1: Indicates that the overcurrent data has been less than the lower limit 1	R
11	—	This bit is read as 0.	R
12	OC1FH0	Channel 0 overcurrent upper limit exceeded 1 flag This bit becomes 1 when the overcurrent data is updated and this data is larger than the upper limit 1. This bit becomes 0 when OC1FH0 = 1 and writing 1 to the channel 0 overcurrent upper limit exceeded flag 1 clear (DSCOCESCR.CLROC1FH0 / DSCSCRCH0.CLROC1FH). 0: Indicates that the overcurrent data doesn't exceed the upper limit 1 1: Indicates that the overcurrent data has exceeded the upper limit 1	R
13	OC1FH1	Channel 1 overcurrent upper limit exceeded 1 flag This bit becomes 1 when the overcurrent data is updated and this data is larger than the upper limit 1. This bit becomes 0 when OC1FH1 = 1 and writing 1 to the channel 1 overcurrent upper limit exceeded flag 1 clear (DSCOCESCR.CLROC1FH1 / DSCSCRCH1.CLROC1FH). 0: Indicates that the overcurrent data doesn't exceed the upper limit 1 1: Indicates that the overcurrent data has exceeded the upper limit 1	R
14	OC1FH2	Channel 2 overcurrent upper limit exceeded 1 flag This bit becomes 1 when the overcurrent data is updated and this data is larger than the upper limit 1. This bit becomes 0 when OC1FH2 = 1 and writing 1 to the channel 2 overcurrent upper limit exceeded flag 1 clear (DSCOCESCR.CLROC1FH2 / DSCSCRCH2.CLROC1FH). 0: Indicates that the overcurrent data doesn't exceed the upper limit 1 1: Indicates that the overcurrent data has exceeded the upper limit 1	R
15	—	This bit is read as 0.	R

Bit	Symbol	Function	R/W
16	OC2FL0	Channel 0 overcurrent lower limit detection 2 flag This bit becomes 1 when the overcurrent data is updated and this data is smaller than the lower limit 2. This bit becomes 0 when OC2FL0 = 1 and writing 1 to the channel 0 overcurrent lower limit detection flag 2 clear (DSCOCESCR.CLROC2FL0 / DSCSCRCH0.CLROC2FL). 0: Indicates that the overcurrent data is not less than the lower limit 2 1: Indicates that the overcurrent data has been less than the lower limit 2	R
17	OC2FL1	Channel 1 overcurrent lower limit detection 2 flag This bit becomes 1 when the overcurrent data is updated and this data is smaller than the lower limit 2. This bit becomes 0 when OC2FL1 = 1 and writing 1 to the channel 1 overcurrent lower limit detection flag 2 clear (DSCOCESCR.CLROC2FL1 / DSCSCRCH1.CLROC2FL). 0: Indicates that the overcurrent data is not less than the lower limit 2 1: Indicates that the overcurrent data has been less than the lower limit 2	R
18	OC2FL2	Channel 2 overcurrent lower limit detection 2 flag This bit becomes 1 when the overcurrent data is updated and this data is smaller than the lower limit 2. This bit becomes 0 when OC2FL2 = 1 and writing 1 to the channel 2 overcurrent lower limit detection flag 2 clear (DSCOCESCR.CLROC2FL2 / DSCSCRCH2.CLROC2FL). 0: Indicates that the overcurrent data is not less than the lower limit 2 1: Indicates that the overcurrent data has been less than the lower limit 2	R
19	—	This bit is read as 0.	R
20	OC2FH0	Channel 0 overcurrent upper limit exceeded 2 flag This bit becomes 1 when the overcurrent data is updated and this data is larger than the upper limit 2. This bit becomes 0 when OC2FH0 = 1 and writing 1 to the channel 0 overcurrent upper limit exceeded flag 0 clear (DSCOCESCR.CLROC2FH0 / DSCSCRCH0.CLROC2FH). 0: Indicates that the overcurrent data doesn't exceed the upper limit 2 1: Indicates that the overcurrent data has exceeded the upper limit 2	R
21	OC2FH1	Channel 1 overcurrent upper limit exceeded 2 flag This bit becomes 1 when the overcurrent data is updated and this data is larger than the upper limit 2. This bit becomes 0 when OC2FH1 = 1 and writing 1 to the channel 1 overcurrent upper limit exceeded flag 2 clear (DSCOCESCR.CLROC2FH1 / DSCSCRCH1.CLROC2FH). 0: Indicates that the overcurrent data doesn't exceed the upper limit 2 1: Indicates that the overcurrent data has exceeded the upper limit 2	R
22	OC2FH2	Channel 2 overcurrent upper limit exceeded 2 flag This bit becomes 1 when the overcurrent data is updated and this data is larger than the upper limit 2. This bit becomes 0 when OC2FH2 = 1 and writing 1 to the channel 2 overcurrent upper limit exceeded flag 2 clear (DSCOCESCR.CLROC2FH2 / DSCSCRCH2.CLROC2FH). 0: Indicates that the overcurrent data doesn't exceed the upper limit 2 1: Indicates that the overcurrent data has exceeded the upper limit 2	R
31:23	—	These bits are read as 0.	R

38.2.1.13 DSCOCNSR : Channel Overcurrent Notification Status Register

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x228

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	OWD3N2	OWD3N1	OWD3N0	—	OWD2N2	OWD2N1	OWD2N0	—	OWD1N2	OWD1N1	OWD1N0	—	OWD0N2	OWD0N1	OWD0N0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OWD0N0	Channel 0 overcurrent detection window notification 0 flag This bit becomes 1 when the condition of overcurrent detection window notification 0 is satisfied. See section 38.3.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD0N0 = 1 and writing 1 to the channel 0 overcurrent detection window notification flag 0 clear (DSCOCNSCR.CLROWD0N0 / DSCSCRCH0.CLROWD0). 0: Indicates that the condition of overcurrent detection window notification 0 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 0 was satisfied	R
1	OWD0N1	Channel 1 overcurrent detection window notification 0 flag This bit becomes 1 when the condition of overcurrent detection window notification 0 is satisfied. See section 38.3.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD0N1 = 1 and writing 1 to the channel 1 overcurrent detection window notification flag 0 clear (DSCOCNSCR.CLROWD0N1 / DSCSCRCH1.CLROWD0). 0: Indicates that the condition of overcurrent detection window notification 0 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 0 was satisfied	R
2	OWD0N2	Channel 2 overcurrent detection window notification 0 flag This bit becomes 1 when the condition of overcurrent detection window notification 0 is satisfied. See section 38.3.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD0N2 = 1 and writing 1 to the channel 2 overcurrent detection window notification flag 0 clear (DSCOCNSCR.CLROWD0N2 / DSCSCRCH2.CLROWD0). 0: Indicates that the condition of overcurrent detection window notification 0 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 0 was satisfied	R
3	—	This bit is read as 0.	R
4	OWD1N0	Channel 0 overcurrent detection window notification 1 flag This bit becomes 1 when the condition of overcurrent detection window notification 1 is satisfied. See section 38.3.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD1N0 = 1 and writing 1 to the channel 0 overcurrent detection window notification flag 1 clear (DSCOCNSCR.CLROWD1N0 / DSCSCRCH0.CLROWD1). 0: Indicates that the condition of overcurrent detection window notification 1 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 1 was satisfied	R
5	OWD1N1	Channel 1 overcurrent detection window notification 1 flag This bit becomes 1 when the condition of overcurrent detection window notification 1 is satisfied. See section 38.3.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD1N1 = 1 and writing 1 to the channel 1 overcurrent detection window notification flag 1 clear (DSCOCNSCR.CLROWD1N1 / DSCSCRCH1.CLROWD1). 0: Indicates that the condition of overcurrent detection window notification 1 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 1 was satisfied	R
6	OWD1N2	Channel 2 overcurrent detection window notification 1 flag This bit becomes 1 when the condition of overcurrent detection window notification 1 is satisfied. See section 38.3.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD1N2 = 1 and writing 1 to the channel 2 overcurrent detection window notification flag 1 clear (DSCOCNSCR.CLROWD1N2 / DSCSCRCH2.CLROWD1). 0: Indicates that the condition of overcurrent detection window notification 1 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 1 was satisfied	R
7	—	This bit is read as 0.	R

Bit	Symbol	Function	R/W
8	OWD2N0	Channel 0 overcurrent detection window notification 2 flag This bit becomes 1 when the condition of overcurrent detection window notification 2 is satisfied. See section 38.3.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD2N0 = 1 and writing 1 to the channel 0 overcurrent detection window notification flag 2 clear (DSCOCNSCR.CLROWD2N0 / DSCSCRCH0.CLROWD2). 0: Indicates that the condition of overcurrent detection window notification 2 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 2 was satisfied	R
9	OWD2N1	Channel 1 overcurrent detection window notification 2 flag This bit becomes 1 when the condition of overcurrent detection window notification 2 is satisfied. See section 38.3.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD2N1 = 1 and writing 1 to the channel 1 overcurrent detection window notification flag 2 clear (DSCOCNSCR.CLROWD2N1 / DSCSCRCH1.CLROWD2). 0: Indicates that the condition of overcurrent detection window notification 2 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 2 was satisfied	R
10	OWD2N2	Channel 2 overcurrent detection window notification 2 flag This bit becomes 1 when the condition of overcurrent detection window notification 2 is satisfied. See section 38.3.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD2N2 = 1 and writing 1 to the channel 2 overcurrent detection window notification flag 2 clear (DSCOCNSCR.CLROWD2N2 / DSCSCRCH2.CLROWD2). 0: Indicates that the condition of overcurrent detection window notification 2 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 2 was satisfied	R
11	—	This bit is read as 0.	R
12	OWD3N0	Channel 0 overcurrent detection window notification 3 flag This bit becomes 1 when the condition of overcurrent detection window notification 3 is satisfied. See section 38.3.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD3N0 = 1 and writing 1 to the channel 0 overcurrent detection window notification flag 3 clear (DSCOCNSCR.CLROWD3N0 / DSCSCRCH0.CLROWD3). 0: Indicates that the condition of overcurrent detection window notification 3 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 3 was satisfied	R
13	OWD3N1	Channel 1 overcurrent detection window notification 3 flag This bit becomes 1 when the condition of overcurrent detection window notification 3 is satisfied. See section 38.3.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD3N1 = 1 and writing 1 to the channel 1 overcurrent detection window notification flag 3 clear (DSCOCNSCR.CLROWD3N1 / DSCSCRCH1.CLROWD3). 0: Indicates that the condition of overcurrent detection window notification 3 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 3 was satisfied	R
14	OWD3N2	Channel 2 overcurrent detection window notification 3 flag This bit becomes 1 when the condition of overcurrent detection window notification 3 is satisfied. See section 38.3.13.2. Overcurrent Detection Window Notification for detail. This bit becomes 0 when OWD3N2 = 1 and writing 1 to the channel 2 overcurrent detection window notification flag 3 clear (DSCOCNSCR.CLROWD3N2 / DSCSCRCH2.CLROWD3). 0: Indicates that the condition of overcurrent detection window notification 3 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 3 was satisfied	R
31:15	—	These bits are read as 0.	R

38.2.1.14 DSCOCRMR : Channel Overcurrent Comparator Result Monitor Register

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x22C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	OC2C MPH2	OC2C MPH1	OC2C MPH0	—	OC2C MPL2	OC2C MPL1	OC2C MPL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	OC1C MPH2	OC1C MPH1	OC1C MPH0	—	OC1C MPL2	OC1C MPL1	OC1C MPL0	—	OC0C MPH2	OC0C MPH1	OC0C MPH0	—	OC0C MPL2	OC0C MPL1	OC0C MPL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OC0CMPL0	Channel 0 overcurrent detect 0 lower limit compare result 0: Indicates that the overcurrent data is not less than the lower limit 0 1: Indicates that the overcurrent data has been less than the lower limit 0	R
1	OC0CMPL1	Channel 1 overcurrent detect 0 lower limit compare result 0: Indicates that the overcurrent data is not less than the lower limit 0 1: Indicates that the overcurrent data has been less than the lower limit 0	R
2	OC0CMPL2	Channel 2 overcurrent detect 0 lower limit compare result 0: Indicates that the overcurrent data is not less than the lower limit 0 1: Indicates that the overcurrent data has been less than the lower limit 0	R
3	—	This bit is read as 0.	R
4	OC0CMPH0	Channel 0 overcurrent detect 0 upper limit compare result 0: Indicates that the overcurrent data does not exceed the upper limit 0 1: Indicates that the overcurrent data has exceeded the upper limit 0	R
5	OC0CMPH1	Channel 1 overcurrent detect 0 upper limit compare result 0: Indicates that the overcurrent data does not exceed the upper limit 0 1: Indicates that the overcurrent data has exceeded the upper limit 0	R
6	OC0CMPH2	Channel 2 overcurrent detect 0 upper limit compare result 0: Indicates that the overcurrent data does not exceed the upper limit 0 1: Indicates that the overcurrent data has exceeded the upper limit 0	R
7	—	This bit is read as 0.	R
8	OC1CMPL0	Channel 0 overcurrent detect 1 lower limit compare result 0: Indicates that the overcurrent data is not less than the lower limit 1 1: Indicates that the overcurrent data has been less than the lower limit 1	R
9	OC1CMPL1	Channel 1 overcurrent detect 1 lower limit compare result 0: Indicates that the overcurrent data is not less than the lower limit 1 1: Indicates that the overcurrent data has been less than the lower limit 1	R
10	OC1CMPL2	Channel 2 overcurrent detect 1 lower limit compare result 0: Indicates that the overcurrent data is not less than the lower limit 1 1: Indicates that the overcurrent data has been less than the lower limit 1	R
11	—	This bit is read as 0.	R
12	OC1CMPH0	Channel 0 overcurrent detect 1 upper limit compare result 0: Indicates that the overcurrent data does not exceed the upper limit 1 1: Indicates that the overcurrent data has exceeded the upper limit 1	R
13	OC1CMPH1	Channel 1 overcurrent detect 1 upper limit compare result 0: Indicates that the overcurrent data does not exceed the upper limit 1 1: Indicates that the overcurrent data has exceeded the upper limit 1	R
14	OC1CMPH2	Channel 2 overcurrent detect 1 upper limit compare result 0: Indicates that the overcurrent data does not exceed the upper limit 1 1: Indicates that the overcurrent data has exceeded the upper limit 1	R

Bit	Symbol	Function	R/W
15	—	This bit is read as 0.	R
16	OC2CMPL0	Channel 0 overcurrent detect 2 lower limit compare result 0: Indicates that the overcurrent data is not less than the lower limit 2 1: Indicates that the overcurrent data has been less than the lower limit 2	R
17	OC2CMPL1	Channel 1 overcurrent detect 2 lower limit compare result 0: Indicates that the overcurrent data is not less than the lower limit 2 1: Indicates that the overcurrent data has been less than the lower limit 2	R
18	OC2CMPL2	Channel 2 overcurrent detect 2 lower limit compare result 0: Indicates that the overcurrent data is not less than the lower limit 2 1: Indicates that the overcurrent data has been less than the lower limit 2	R
19	—	This bit is read as 0.	R
20	OC2CMPH0	Channel 0 overcurrent detect 2 upper limit compare result 0: Indicates that the overcurrent data does not exceed the upper limit 2 1: Indicates that the overcurrent data has exceeded the upper limit 2	R
21	OC2CMPH1	Channel 1 overcurrent detect 2 upper limit compare result 0: Indicates that the overcurrent data does not exceed the upper limit 2 1: Indicates that the overcurrent data has exceeded the upper limit 2	R
22	OC2CMPH2	Channel 2 overcurrent detect 2 upper limit compare result 0: Indicates that the overcurrent data does not exceed the upper limit 2 1: Indicates that the overcurrent data has exceeded the upper limit 2	R
31:23	—	These bits are read as 0.	R

The DSCOCRMR register has monitor bits that return the comparison result with the upper/lower limit table for each overcurrent detection every time the overcurrent data is updated. There is no status clear bit of this register because the result updates when the overcurrent data is updated.

38.2.1.15 DSCSR : Channel Status Register

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x240

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	CBUF2	CBUF1	CBUF0	—	CAUF2	CAUF1	CAUF0	—	DUF2	DUF1	DUF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DUF0	Channel 0 Data Update flag This bit is the Current Data Register Update Interrupt Status bit. This bit becomes 0 when DUF0 = 1 and 1 to the Channel n Data Update Clear flag (DSCSCR.CLRDUF0/ DSCSCRCH0.CLRDUF) or when the Current Data Register Channel 0 (DSCDRCH0) is read. 0: Indicates that the Current Data register has not been updated 1: Indicates that the Current Data register has been updated	R
1	DUF1	Channel 1 Data Update flag This bit is the Current Data Register Update Interrupt Status bit. This bit becomes 0 when DUF1 = 1 and 1 to the Channel n Data Update Clear flag (DSCSCR.CLRDUF1/ DSCSCRCH1.CLRDUF) or when the Current Data Register Channel 1 (DSCDRCH1) is read. 0: Indicates that the Current Data register has not been updated 1: Indicates that the Current Data register has been updated	R

Bit	Symbol	Function	R/W
2	DUF2	Channel 2 Data Update flag This bit is the Current Data Register Update Interrupt Status bit. This bit becomes 0 when DUF2 = 1 and 1 to the Channel n Data Update Clear flag (DSCSCR.CLRDUF2/ DSCSCRCH2.CLRDUF) or when the Current Data Register Channel 2 (DSCDRCH2) is read. 0: Indicates that the Current Data register has not been updated 1: Indicates that the Current Data register has been updated	R
3	—	This bit is read as 0.	R
4	CAUF0	Channel 0 capture data A update flag This bit becomes 0 when CAUF0 = 1 and 1 to the capture data update flag A clear (DSCSCR.CLRCAUF00 / DSCESCRCH0.CLRCAUF) or when the capture current data register A channel 0 (DSCCDRACH0) is read. 0: Indicates that the capture register A has not been updated 1: Indicates that the capture register A has been updated	R
5	CAUF1	Channel 1 capture data A update flag This bit becomes 0 when CAUF1 = 1 and 1 to the capture data update flag A clear (DSCSCR.CLRCAUF01 / DSCESCRCH1.CLRCAUF) or when the capture current data register A channel 1 (DSCCDRACH1) is read. 0: Indicates that the capture register A has not been updated 1: Indicates that the capture register A has been updated	R
6	CAUF2	Channel 2 capture data A update flag This bit becomes 0 when CAUF2 = 1 and 1 to the capture data update flag A clear (DSCSCR.CLRCAUF02 / DSCESCRCH2.CLRCAUF) or when the capture current data register A channel 2 (DSCCDRACH2) is read. 0: Indicates that the capture register A has not been updated 1: Indicates that the capture register A has been updated	R
7	—	This bit is read as 0.	R
8	CBUF0	Channel 0 capture data B update flag This bit becomes 0 when CBUF0 = 1 and 1 to the capture data update flag B clear (DSCSCR.CLRCBUF00 / DSCESCRCH0.CLRCBUF) or when the capture current data register B channel 0 (DSCCDRBCH0) is read. 0: Indicates that the capture register B has not been updated 1: Indicates that the capture register B has been updated	R
9	CBUF1	Channel 1 capture data B update flag This bit becomes 0 when CBUF1 = 1 and 1 to the capture data update flag B clear (DSCSCR.CLRCBUF01 / DSCESCRCH1.CLRCBUF) or when the capture current data register B channel 1 (DSCCDRBCH1) is read. 0: Indicates that the capture register B has not been updated 1: Indicates that the capture register B has been updated	R
10	CBUF2	Channel 2 capture data B update flag This bit becomes 0 when CBUF2 = 1 and 1 to the capture data update flag B clear (DSCSCR.CLRCBUF02 / DSCESCRCH2.CLRCBUF) or when the capture current data register B channel 2 (DSCCDRBCH2) is read. 0: Indicates that the capture register B has not been updated 1: Indicates that the capture register B has been updated	R
31:11	—	These bits are read as 0.	R

38.2.1.16 DSCSSR : Channel State Status Register

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x244

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CHSTATE2	—	—	—	CHSTATE1	—	—	—	CHSTATE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CHSTATE0	Channel 0 state 0: Channel stop 1: Channel in operation	R
3:1	—	These bits are read as 0.	R
4	CHSTATE1	Channel 1 state 0: Channel stop 1: Channel in operation	R
7:5	—	These bits are read as 0.	R
8	CHSTATE2	Channel 2 state 0: Channel stop 1: Channel in operation	R
31:9	—	These bits are read as 0.	R

38.2.1.17 DSCESCR : Channel Error Status Clear Register

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x260

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRSUMERRH	CLRSUMERRL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	CLRSCF2	CLRSCF1	CLRSCF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLRSCF0	Channel 0 Overcurrent Lower Limit Detection Flag Clear 0: No effect 1: Clear DSCESR.OCFL0 / DSCSRCH0.OCFL	W
1	CLRSCF1	Channel 1 Overcurrent Lower Limit Detection Flag Clear 0: No effect 1: Clear DSCESR.OCFL1 / DSCSRCH1.OCFL	W
2	CLRSCF2	Channel 2 Overcurrent Lower Limit Detection Flag Clear 0: No effect 1: Clear DSCESR.OCFL2 / DSCSRCH2.OCFL	W
15:3	—	The write value should be 0.	W

Bit	Symbol	Function	R/W
16	CLRSUMERRL	Overcurrent Sum Error Lower Limit Detection Flag Clear 0: No effect 1: Clear DSCESR.SUMERRL	W
17	CLRSUMERRH	Overcurrent Sum Error Upper Limit Detection Flag Clear 0: No effect 1: Clear DSCESR.SUMERRH	W
31:18	—	The write value should be 0.	W

38.2.1.18 DSCOCESCR : Channel Overcurrent Error Status Clear Register

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x264

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	CLROC2FH2	CLROC2FH1	CLROC2FH0	—	CLROC2FL2	CLROC2FL1	CLROC2FL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CLROC1FH2	CLROC1FH1	CLROC1FH0	—	CLROC1FL2	CLROC1FL1	CLROC1FL0	—	CLROC0FH2	CLROC0FH1	CLROC0FH0	—	CLROC0FL2	CLROC0FL1	CLROC0FL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLROC0FL0	Channel 0 overcurrent lower limit detection 0 flag clear 0: No effect 1: Clear DSCOCESR.OC0FL0 / DSCSRCH0.OC0FL	W
1	CLROC0FL1	Channel 1 overcurrent lower limit detection 0 flag clear 0: No effect 1: Clear DSCOCESR.OC0FL1 / DSCSRCH1.OC0FL	W
2	CLROC0FL2	Channel 2 overcurrent lower limit detection 0 flag clear 0: No effect 1: Clear DSCOCESR.OC0FL2 / DSCSRCH2.OC0FL	W
3	—	The write value should be 0.	W
4	CLROC0FH0	Channel 0 overcurrent upper limit exceeded 0 flag clear 0: No effect 1: Clear DSCOCESR.OC0FH0 / DSCSRCH0.OC0FH	W
5	CLROC0FH1	Channel 1 overcurrent upper limit exceeded 0 flag clear 0: No effect 1: Clear DSCOCESR.OC0FH1 / DSCSRCH1.OC0FH	W
6	CLROC0FH2	Channel 2 overcurrent upper limit exceeded 0 flag clear 0: No effect 1: Clear DSCOCESR.OC0FH2 / DSCSRCH2.OC0FH	W
7	—	The write value should be 0.	W
8	CLROC1FL0	Channel 0 overcurrent lower limit detection 1 flag clear 0: No effect 1: Clear DSCOCESR.OC1FL0 / DSCSRCH0.OC1FL	W
9	CLROC1FL1	Channel 1 overcurrent lower limit detection 1 flag clear 0: No effect 1: Clear DSCOCESR.OC1FL1 / DSCSRCH1.OC1FL	W
10	CLROC1FL2	Channel 2 overcurrent lower limit detection 1 flag clear 0: No effect 1: Clear DSCOCESR.OC1FL2 / DSCSRCH2.OC1FL	W
11	—	The write value should be 0.	W

Bit	Symbol	Function	R/W
12	CLROC1FH0	Channel 0 overcurrent upper limit exceeded 1 flag clear 0: No effect 1: Clear DSCOCESR.OC1FH0 / DSCSRCH0.OC1FH	W
13	CLROC1FH1	Channel 1 overcurrent upper limit exceeded 1 flag clear 0: No effect 1: Clear DSCOCESR.OC1FH1 / DSCSRCH1.OC1FH	W
14	CLROC1FH2	Channel 2 overcurrent upper limit exceeded 1 flag clear 0: No effect 1: Clear DSCOCESR.OC1FH2 / DSCSRCH2.OC1FH	W
15	—	The write value should be 0.	W
16	CLROC2FL0	Channel 0 overcurrent lower limit detection 2 flag clear 0: No effect 1: Clear DSCOCESR.OC2FL0 / DSCSRCH0.OC2FL	W
17	CLROC2FL1	Channel 1 overcurrent lower limit detection 2 flag clear 0: No effect 1: Clear DSCOCESR.OC2FL1 / DSCSRCH1.OC2FL	W
18	CLROC2FL2	Channel 2 overcurrent lower limit detection 2 flag clear 0: No effect 1: Clear DSCOCESR.OC2FL2 / DSCSRCH2.OC2FL	W
19	—	The write value should be 0.	W
20	CLROC2FH0	Channel 0 overcurrent upper limit exceeded 2 flag clear 0: No effect 1: Clear DSCOCESR.OC2FH0 / DSCSRCH0.OC2FH	W
21	CLROC2FH1	Channel 1 overcurrent upper limit exceeded 2 flag 0: No effect 1: Clear DSCOCESR.OC2FH1 / DSCSRCH1.OC2FH	W
22	CLROC2FH2	Channel 2 overcurrent upper limit exceeded 2 flag clear 0: No effect 1: Clear DSCOCESR.OC2FH2 / DSCSRCH2.OC2FH	W
31:23	—	The write value should be 0.	W

38.2.1.19 DSCOCNSCR : Channel Overcurrent Notification Status Clear Register

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x268

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	CLROWD3N2	CLROWD3N1	CLROWD3N0	—	CLROWD2N2	CLROWD2N1	CLROWD2N0	—	CLROWD1N2	CLROWD1N1	CLROWD1N0	—	CLROWD0N2	CLROWD0N1	CLROWD0N0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLROWD0N0	Channel 0 overcurrent detection window notification 0 flag clear 0: No effect 1: Clear DSCOCNSR.OWD0N0 / DSCSRCH0.OWD0N	W
1	CLROWD0N1	Channel 1 overcurrent detection window notification 0 flag clear 0: No effect 1: Clear DSCOCNSR.OWD0N1 / DSCSRCH1.OWD0N	W

Bit	Symbol	Function	R/W
2	CLROWD0N2	Channel 2 overcurrent detection window notification 0 flag clear 0: No effect 1: Clear DSCOCNSR.OWD0N2 / DSCSRCH2.OWD0N	W
3	—	The write value should be 0.	W
4	CLROWD1N0	Channel 0 overcurrent detection window notification 1 flag clear 0: No effect 1: Clear DSCOCNSR.OWD1N0 / DSCSRCH0.OWD1N	W
5	CLROWD1N1	Channel 1 overcurrent detection window notification 1 flag clear 0: No effect 1: Clear DSCOCNSR.OWD1N1 / DSCSRCH1.OWD1N	W
6	CLROWD1N2	Channel 2 overcurrent detection window notification 1 flag clear 0: No effect 1: Clear DSCOCNSR.OWD1N2 / DSCSRCH2.OWD1N	W
7	—	The write value should be 0.	W
8	CLROWD2N0	Channel 0 overcurrent detection window notification 2 flag clear 0: No effect 1: Clear DSCOCNSR.OWD2N0 / DSCSRCH0.OWD2N	W
9	CLROWD2N1	Channel 1 overcurrent detection window notification 2 flag clear 0: No effect 1: Clear DSCOCNSR.OWD2N1 / DSCSRCH1.OWD2N	W
10	CLROWD2N2	Channel 2 overcurrent detection window notification 2 flag clear 0: No effect 1: Clear DSCOCNSR.OWD2N2 / DSCSRCH2.OWD2N	W
11	—	The write value should be 0.	W
12	CLROWD3N0	Channel 0 overcurrent detection window notification 3 flag clear 0: No effect 1: Clear DSCOCNSR.OWD3N0 / DSCSRCH0.OWD3N	W
13	CLROWD3N1	Channel 1 overcurrent detection window notification 3 flag clear 0: No effect 1: Clear DSCOCNSR.OWD3N1 / DSCSRCH1.OWD3N	W
14	CLROWD3N2	Channel 2 overcurrent detection window notification 3 flag clear 0: No effect 1: Clear DSCOCNSR.OWD3N2 / DSCSRCH2.OWD3N	W
31:15	—	The write value should be 0.	W

38.2.1.20 DSCSCR : Channel Status Clear Register

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x280

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	CLRC BUF2	CLRC BUF1	CLRC BUF0	—	CLRC AUF2	CLRC AUF1	CLRC AUF0	—	CLRD UF2	CLRD UF1	CLRD UF0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLRDUF0	Channel 0 Data Update Flag Clear 0: No effect 1: Clear DSCSR.DUF0 / DSCSRCH0.DUF	W

Bit	Symbol	Function	R/W
1	CLRDUF1	Channel 1 Data Update Flag Clear 0: No effect 1: Clear DSCSR.DUF1 / DSCSRCH1.DUF	W
2	CLRDUF2	Channel 2 Data Update Flag Clear 0: No effect 1: Clear DSCSR.DUF2 / DSCSRCH2.DUF	W
3	—	The write value should be 0.	W
4	CLRCAUF0	Channel 0 Capture Data A Update Flag Clear 0: No effect 1: Clear DSCSR.CAUF0 / DSCSRCH0.CAUF	W
5	CLRCAUF1	Channel 1 Capture Data A Update Flag Clear 0: No effect 1: Clear DSCSR.CAUF1 / DSCSRCH1.CAUF	W
6	CLRCAUF2	Channel 2 Capture Data A Update Flag Clear 0: No effect 1: Clear DSCSR.CAUF2 / DSCSRCH2.CAUF	W
7	—	The write value should be 0.	W
8	CLRCBUF0	Channel 0 Capture Data B Update Flag Clear 0: No effect 1: Clear DSCSR.CBUF0 / DSCSRCH0.CBUF	W
9	CLRCBUF1	Channel 1 Capture Data B Update Flag Clear 0: No effect 1: Clear DSCSR.CBUF1 / DSCSRCH1.CBUF	W
10	CLRCBUF2	Channel 2 Capture Data B Update Flag Clear 0: No effect 1: Clear DSCSR.CBUF2 / DSCSRCH2.CBUF	W
31:11	—	The write value should be 0.	W

38.2.2 DSMIF Channel Control Registers

38.2.2.1 DSICRCHn : Interrupt Control Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x300 + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	OWNE3	OWNE2	OWNE1	OWNE0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	IOEH2	IOEL2	IOEH1	IOEL1	IOEH0	IOEL0	—	—	—	—	ISE	IBUE	IAUE	IUE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	IUE	Current data register update interrupt enable 0: Do not output current data register update interrupt 1: Current data register update interrupt can be output	R/W
1	IAUE	Capture current data register A update interrupt enable 0: Do not output capture current data register A update interrupt 1: Capture current data register A update interrupt can be output	R/W

Bit	Symbol	Function	R/W
2	IBUE	Capture current data register B update interrupt enable 0: Do not output capture current data register B update interrupt 1: Capture current data register B update interrupt can be output	R/W
3	ISE	Short circuit detection error interrupt enable bit 0: Do not output short circuit detection interrupt 1: Short circuit detection interrupt can be output	R/W
7:4	—	These bits are read as 0. The write value should be 0.	R/W
8	IOEL0	Overcurrent lower limit detection interrupt 0 0: Do not output the overcurrent lower limit detection interrupt 0 1: Overcurrent lower limit detection interrupt 0 can be output	R/W
9	IOEH0	Overcurrent upper limit exceeded detection interrupt 0 0: Do not output the overcurrent upper limit detection interrupt 0 1: Overcurrent upper limit exceeded detection interrupt 0 can be output	R/W
10	IOEL1	Overcurrent lower limit detection interrupt 1 0: Do not output the overcurrent lower limit detection interrupt 1 1: Overcurrent lower limit detection interrupt 1 can be output	R/W
11	IOEH1	Overcurrent upper limit exceeded detection interrupt 1 0: Do not output the overcurrent upper limit detection interrupt 1 1: Overcurrent upper limit exceeded detection interrupt 1 can be output	R/W
12	IOEL2	Overcurrent lower limit detection interrupt 2 0: Do not output the overcurrent lower limit detection interrupt 2 1: Overcurrent lower limit detection interrupt 2 can be output	R/W
13	IOEH2	Overcurrent upper limit exceeded detection interrupt 2 0: Do not output the overcurrent upper limit detection interrupt 2 1: Overcurrent upper limit exceeded detection interrupt 2 can be output	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W
16	OWNE0	Overcurrent detection window notification 0 output enable 0: Do not output overcurrent detection window notification 0 1: Overcurrent detection window notification 0 can be output	R/W
17	OWNE1	Overcurrent detection window notification 1 output enable 0: Do not output overcurrent detection window notification 1 1: Overcurrent detection window notification 1 can be output	R/W
18	OWNE2	Overcurrent detection window notification 2 output enable 0: Do not output overcurrent detection window notification 2 1: Overcurrent detection window notification 2 can be output	R/W
19	OWNE3	Overcurrent detection window notification 3 output enable 0: Do not output overcurrent detection window notification 3 1: Overcurrent detection window notification 3 can be output	R/W
31:20	—	These bits are read as 0. The write value should be 0.	R/W

38.2.2.2 DSCMCCRCHn : Current Measurement Clock Control Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x304 + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CKDIV[5:0]						SEDGE	—	—	—	—	—	—	CKDIR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CKDIR	A/D conversion clock master/slave switching 0: MCLKn pin is input (slave operation) 1: MCLKn pin is output (master operation)	R/W
6:1	—	These bits are read as 0. The write value should be 0.	R/W
7	SEDGE	Sampling edge selection 0: Capture MDATn at the negative edge of MCLKn 1: Capture MDATn at the positive edge of MCLKn	R/W
13:8	CKDIV[5:0]	A/D conversion clock division ratio Set the division ratio in clock master mode. Set to a division ratio of 2 × (CKDIV[5:0] + 1) with the core clock	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

38.2.2.3 DSCMFCRCHn : Current Measurement Filter Control Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x308 + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	CMSH[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMDEC[7:0]								—	—	—	—	—	—	CMSINC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CMSINC[1:0]	Current measurement filter order setting 0 0: Sinc3 (3rd order) 0 1: Sinc1 (1st order) 1 0: Sinc2 (2nd order) 1 1: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
15:8	CMDEC[7:0]	Decimation ratio selection for current measurement M indicates the decimation ratio. Decimation ratio M is set as follows: M = CMDEC + 1 The decimation ratio M can be set within the following range. 4 ≤ M ≤ 256 (0x3 ≤ CMDEC ≤ 0xFF) Setting less than M = 4 is prohibited. When these value is set, Sinc filter operation is not guaranteed.	R/W

Bit	Symbol	Function	R/W
20:16	CMSH[4:0]	Data shift setting for current measurement*1 0x00: [23:8] 0x01: [22:7] 0x02: [21:6] 0x03: [20:5] 0x04: [19:4] 0x05: [18:3] 0x06: [17:2] 0x07: [16:1] 0x08: [15:0] 0x09: [14:0]<<1 0x0A: [13:0]<<2 0x0B: [12:0]<<3 0x0C: [11:0]<<4 0x0D: [10:0]<<5 0x0E: [9:0]<<6 0x0F: [8:0]<<7 0x10: [7:0]<<8 0x11: [6:0]<<9 0x12: [5:0]<<10 0x13: [4:0]<<11 Others: Setting prohibited	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. In CMSH[4:0] = 0x09 to 0x13, the lower bits are filled with 0s.

38.2.2.4 DSCMCTCRCHn : Current Measurement Capture Trigger Control Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x30C + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	DEDGE	—	—	—	—	—	DITSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	CTSELB[2:0]			—	—	—	—	—	CTSELA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	CTSELA[2:0]	Current capture trigger A selection bit 0x0: Do not capture 0x1: Select current data capture trigger 0 0x2: Select current data capture trigger 1 0x3: Select current data capture trigger 2 0x4: Select current data capture trigger 3 0x5: Select current data capture trigger 4 0x6: Select current data capture trigger 5 0x7: Setting prohibited (do not capture)	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
10:8	CTSELB[2:0]	Current capture trigger B selection bit 0x0: Do not capture 0x1: Select current data capture trigger 0 0x2: Select current data capture trigger 1 0x3: Select current data capture trigger 2 0x4: Select current data capture trigger 3 0x5: Select current data capture trigger 4 0x6: Select current data capture trigger 5 0x7: Setting prohibited (do not capture)	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W
17:16	DITSEL[1:0]	Current measurement filter initialization trigger selection bit for frequency division counter for decimation. 0 0: Do not initialize 0 1: Select decimation dividing counter initialization trigger 0 1 0: Select decimation dividing counter initialization trigger 1 1 1: Select decimation dividing counter initialization trigger 2	R/W
22:18	—	These bits are read as 0. The write value should be 0.	R/W
23	DEDGE	Current measurement filter initialization trigger for division counter for decimation edge selection bit. The trigger from ELC is usually used positive edge. Change from the initial value if necessary. 0: The frequency division counter for decimation is initialized at the negative edge of initialization trigger 1: The frequency division counter for decimation is initialized at the positive edge of initialization trigger	R/W
31:24	—	These bits are read as 0. The write value should be 0.	R/W

38.2.2.5 DSEDCRCHn : Error Detect Control Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x320 + 0x140 × n

Bit position:	31																													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SDE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SDE	Short circuit detection enable bit 0: Short circuit detection is invalid 1: Short circuit detection is valid	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

38.2.2.6 DSOCFCRCHn : Overcurrent Detect Filter Control Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x330 + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	OCSH[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	OCDEC[7:0]								—	—	—	—	—	—	OCSINC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	OCSINC[1:0]	Overcurrent detection filter order setting 0 0: Sinc3 (3rd order) 0 1: Sinc1 (1st order) 1 0: Sinc2 (2nd order) 1 1: Setting prohibited	R/W
7:2	—	These bits are read as 0. The write value should be 0.	R/W
15:8	OCDEC[7:0]	Decimation ratio selection for overcurrent detection M indicates the decimation ratio. Decimation ratio M is set as follows: M = OCDEC + 1 The decimation ratio M can be set within the following range. 4 ≤ M ≤ 256 (0x3 ≤ OCDEC ≤ 0xFF) Setting less than M = 4 is prohibited. When these value is set, Sinc filter operation is not guaranteed.	R/W
20:16	OCSH[4:0]	Data shift setting for overcurrent detection*1 0x00: [23:8] 0x01: [22:7] 0x02: [21:6] 0x03: [20:5] 0x04: [19:4] 0x05: [18:3] 0x06: [17:2] 0x07: [16:1] 0x08: [15:0] 0x09: [14:0]<<1 0x0A: [13:0]<<2 0x0B: [12:0]<<3 0x0C: [11:0]<<4 0x0D: [10:0]<<5 0x0E: [9:0]<<6 0x0F: [8:0]<<7 0x10: [7:0]<<8 0x11: [6:0]<<9 0x12: [5:0]<<10 0x13: [4:0]<<11 Others: Setting prohibited	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. In OCSH[4:0] = 0x09 to 0x13, the lower bits are filled with 0s.

38.2.2.7 DSSCTSRCHn : Short Circuit Threshold Setting Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x324 + 0x140 × n

Bit position:	31			28												16			12												0
Bit field:	—	—	—	SCNTH[12:0]												—	—	—	SCNTL[12:0]												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
12:0	SCNTL[12:0]	Short circuit detection low continuous detection count Set the upper limit for the number of times 0 is input continuously to the MDATn pin.	R/W
15:13	—	These bits are read as 0. The write value should be 0.	R/W
28:16	SCNTH[12:0]	Short circuit detection high continuous detection count Set the upper limit for the number of times 1 is input continuously to the MDATn pin.	R/W
31:29	—	These bits are read as 0. The write value should be 0.	R/W

38.2.2.8 DSODCRCHn : Overcurrent Detect Control Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x334 + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	OWFE3	OWFE2	OWFE1	OWFE0	—	—	ODEH2	ODEL2	ODEH1	ODEL1	ODEH0	ODEL0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ODEL0	Current lower limit detection 0 enable 0: Overcurrent lower limit detection 0 is invalid 1: Overcurrent lower limit detection 0 is valid	R/W
1	ODEH0	Current upper limit exceeded detection 0 enable 0: Overcurrent upper limit exceeded detection 0 is invalid 1: Overcurrent upper limit exceeded detection 0 is valid	R/W
2	ODEL1	Current lower limit detection 1 enable 0: Overcurrent lower limit detection 1 is invalid 1: Overcurrent lower limit detection 1 is valid	R/W
3	ODEH1	Current upper limit exceeded detection 1 enable 0: Overcurrent upper limit exceeded detection 1 is invalid 1: Overcurrent upper limit exceeded detection 1 is valid	R/W
4	ODEL2	Current lower limit detection 2 enable 0: Overcurrent lower limit detection 2 is invalid 1: Overcurrent lower limit detection 2 is valid	R/W
5	ODEH2	Current upper limit exceeded detection 2 enable 0: Overcurrent upper limit exceeded detection 2 is invalid 1: Overcurrent upper limit exceeded detection 2 is valid	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
8	OWFE0	Overcurrent lower limit detection interrupt 0 0: Do not output the overcurrent lower limit detection interrupt 0 1: Overcurrent lower limit detection interrupt 0 can be output	R/W
9	OWFE1	Overcurrent detection window function 0 enable 0: Overcurrent detection window function 0 is invalid 1: Overcurrent detection window function 0 is valid	R/W
10	OWFE2	Overcurrent detection window function 1 enable 0: Overcurrent detection window function 1 is invalid 1: Overcurrent detection window function 1 is valid	R/W
11	OWFE3	Overcurrent detection window function 2 enable 0: Overcurrent detection window function 2 is invalid 1: Overcurrent detection window function 2 is valid	R/W
31:12	—	These bits are read as 0. The write value should be 0.	R/W

38.2.2.9 DSODWCRCHn : Overcurrent Detect Window Control Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x338 + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	OWNM3[3:0]				OWNM2	OWNM1	OWNM0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OWNM0	Channel n overcurrent detection window notification 0 mode select 0: (overcurrent data channel n < lower limit 0) or (overcurrent data channel n > upper limit 0) 1: lower limit 0 channel n ≤ overcurrent data channel n ≤ upper limit 0 channel n	R/W
1	OWNM1	Channel n overcurrent detection window notification 1 mode select 0: (overcurrent data channel n < lower limit 1) or (overcurrent data channel n > upper limit 1) 1: lower limit 1 channel n ≤ overcurrent data channel n ≤ upper limit 1 channel n	R/W
2	OWNM2	Channel n overcurrent detection window notification 2 mode select 0: (overcurrent data channel n < lower limit 2) or (overcurrent data channel n > upper limit 2) 1: lower limit 2 channel n ≤ overcurrent data channel n ≤ upper limit 2 channel n	R/W

Bit	Symbol	Function	R/W
6:3	OWNM3[3:0]	Channel n overcurrent detection window notification 3 mode select 0x0: Overcurrent detection window notification 0 0x1: Overcurrent detection window notification 1 0x2: Overcurrent detection window notification 2 0x3: Overcurrent detection window notification 0 OR overcurrent detection window notification 1 0x4: Overcurrent detection window notification 0 AND overcurrent detection window notification 1 0x5: Overcurrent detection window notification 0 OR overcurrent detection window notification 2 0x6: Overcurrent detection window notification 0 AND overcurrent detection window notification 1 0x7: Overcurrent detection window notification 1 OR overcurrent detection window notification 2 0x8: Overcurrent detection window notification 1 AND overcurrent detection window notification 2 0x9: Overcurrent detection window notification 0 OR overcurrent detection window notification 1 OR overcurrent detection window notification 2 0xA: Overcurrent detection window notification 0 AND overcurrent detection window notification 1 AND overcurrent detection window notification 2 Others: Setting prohibited	R/W
31:7	—	These bits are read as 0. The write value should be 0.	R/W

38.2.2.10 DSOCLTR0CHn : Overcurrent Low Threshold Register 0 Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x3A0 + 0x140 × n

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OCMPTBL0[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	OCMPTBL0[15:0]	Overcurrent detection lower limit 0 This bit sets the lower limit of the active current value for overcurrent detection 0.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

38.2.2.11 DSOCHTR0CHn : Overcurrent High Threshold Register 0 Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x3A4 + 0x140 × n

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OCMPTBH0[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	OCMPTBH0[15:0]	Overcurrent detection upper limit 0 This bit sets the upper limit of the active current value for overcurrent detection 0.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

38.2.2.12 DSOCLTR1CHn : Overcurrent Low Threshold Register 1 Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x3A8 + 0x140 × n

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OCMPTBL1[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	OCMPTBL1[15:0]	Overcurrent detection lower limit 1 This bit sets the lower limit of the active current value for overcurrent detection 1.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

38.2.2.13 DSOCHTR1CHn : Overcurrent High Threshold Register 1 Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x3AC + 0x140 × n

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OCMPTBH1[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	OCMPTBH1[15:0]	Overcurrent detection upper limit 1 This bit sets the upper limit of the active current value for overcurrent detection 1.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

38.2.2.14 DSOCLTR2CHn : Overcurrent Low Threshold Register 2 Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x3B0 + 0x140 × n

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OCMPTBL2[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	OCMPTBL2[15:0]	Overcurrent detection lower limit 2 This bit sets the lower limit of the active current value for overcurrent detection 2.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

38.2.2.15 DSOCHTR2CHn : Overcurrent High Threshold Register 2 Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x3B4 + 0x140 × n

Bit position:	31																15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	OCMPTBH2[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	OCMPTBH2[15:0]	Overcurrent detection upper limit 2 This bit sets the upper limit of the active current value for overcurrent detection 2.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

38.2.2.16 DSCSTRTRCHn : Software Start Trigger Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x3E0 + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STRTRG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STRTRG	Channel start trigger Mirror register of DSCSTRTR.STRTRGn 0: Do nothing 1: Start channel n	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

38.2.2.17 DSCSTPTRCHn : Software Stop Trigger Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x3E4 + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STPTRG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STPTRG	Channel stop trigger Mirror register of DSCSTRTR.STPTRGn 0: Do nothing 1: Stop channel n	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

38.2.2.18 DSCDRCHn : Current Data Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x3F0 + 0x140 × n

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADDR[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ADDR[15:0]	Current data Stores the current value generated by the current measurement data block.	R
31:16	—	These bits are read as 0.	R

38.2.2.19 DSCCDRACHn : Capture Current Data Register A Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x3F4 + 0x140 × n

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CDRA[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CDRA[15:0]	Capture current data A The current data value when the capture trigger A is input is stored.	R
31:16	—	These bits are read as 0.	R

38.2.2.20 DSCCDRBCHn : Capture Current Data Register B Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x3F8 + 0x140 × n

Bit position:	31															15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CDRB[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CDRB[15:0]	Capture current data B The current data value when the capture trigger B is input is stored.	R
31:16	—	These bits are read as 0.	R

38.2.2.21 DSOCDRCHn : Overcurrent Data Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x3FC + 0x140 × n

Bit position:	31															15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ODR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ODR[15:0]	Overcurrent data Stores the current value generated by the overcurrent data block.	R
31:16	—	These bits are read as 0.	R

38.2.2.22 DSCOCDR0CHn : Capture Overcurrent Data Register 0 Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x400 + 0x140 × n

Bit position:	31															15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CODR0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CODR0[15:0]	Capture overcurrent data 0 When either the overcurrent upper limit or the overcurrent lower limit is detected by the overcurrent data block, the overcurrent data 0 used for comparison is stored.	R
31:16	—	These bits are read as 0.	R

38.2.2.23 DSCOCDR1CHn : Capture Overcurrent Data Register 1 Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x404 + 0x140 × n

Bit position:	31															15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CODR1[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CODR1[15:0]	Capture overcurrent data 1 When either the overcurrent upper limit or the overcurrent lower limit is detected by the overcurrent data block, the overcurrent data 1 used for comparison is stored.	R
31:16	—	These bits are read as 0.	R

38.2.2.24 DSCOCDR2CHn : Capture Overcurrent Data Register 2 Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x408 + 0x140 × n

Bit position:	31															15															0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CODR2[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CODR2[15:0]	Capture overcurrent data 2 When either the overcurrent upper limit or the overcurrent lower limit is detected by the overcurrent data block, the overcurrent data 2 used for comparison is stored.	R
31:16	—	These bits are read as 0.	R

38.2.2.25 DSCSRCHn : Status Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x420 + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	OC2CMPH	OC2CMPL	OC1CMPH	OC1CMPL	OC0CMPH	OC0CMPL	—	—	—	—	OWD3N	OWD2N	OWD1N	OWD0N
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	OC2FH	OC2FL	OC1FH	OC1FL	OC0FH	OC0FL	CHSTATE	—	—	—	SCF	CBUF	CAUF	DUF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	DUF	Channel n data update flag Mirror register of DSCSR.DUFLn 0: Indicates that the current data register has not been updated 1: Indicates that the current data register has been updated	R
1	CAUF	Channel n capture data A update flag Mirror register of DSCSR.CAUFn 0: Indicates that the capture data register A has not been updated 1: Indicates that the capture data register A has been updated	R
2	CBUF	Channel n capture data B update flag Mirror register of DSCSR.CBUFn 0: Indicates that the capture data register B has not been updated 1: Indicates that the capture data register B has been updated	R
3	SCF	Channel n short circuit detection flag Mirror register of DSCESR.SCFn 0: Indicates that the MDATn pin short circuit is not detected 1: Indicates that a short circuit on the MDATn pin has been detected	R
6:4	—	These bits are read as 0.	R
7	CHSTATE	Channel n state Mirror register of DSCSSR.CHSTATEn 0: Channel stop 1: Channel in operation	R
8	OC0FL	Channel n overcurrent lower limit detection 0 flag Mirror register of DSCOCESR.OC0FLn 0: Indicates that the overcurrent data is not less than the lower limit 0 1: Indicates that the overcurrent data has been less than the lower limit 0	R

Bit	Symbol	Function	R/W
9	OC0FH	Channel n overcurrent upper limit exceeded 0 flag Mirror register of DSCOCESR.OC0FHn 0: Indicates that the overcurrent data doesn't exceed the upper limit 0 1: Indicates that the overcurrent data has exceeded the upper limit 0	R
10	OC1FL	Channel n overcurrent lower limit detection 1 flag Mirror register of DSCOCESR.OC1FLn 0: Indicates that the overcurrent data is not less than the lower limit 1 1: Indicates that the overcurrent data has been less than the lower limit 1	R
11	OC1FH	Channel n overcurrent upper limit exceeded 1 flag Mirror register of DSCOCESR.OC1FHn 0: Indicates that the overcurrent data doesn't exceed the upper limit 1 1: Indicates that the overcurrent data has exceeded the upper limit 1	R
12	OC2FL	Channel n overcurrent lower limit detection 2 flag Mirror register of DSCOCESR.OC2FLn 0: Indicates that the overcurrent data is not less than the lower limit 2 1: Indicates that the overcurrent data has been less than the lower limit 2	R
13	OC2FH	Channel n overcurrent upper limit exceeded 2 flag Mirror register of DSCOCESR.OC2FHn 0: Indicates that the overcurrent data doesn't exceed the upper limit 2 1: Indicates that the overcurrent data has exceeded the upper limit 2	R
15:14	—	These bits are read as 0.	R
16	OWD0N	Channel n overcurrent detection window notification 0 Mirror register of DSCOCNSR.OWD0Nn 0: Indicates that the condition of overcurrent detection window notification 0 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 0 was satisfied	R
17	OWD1N	Channel n overcurrent detection window notification 1 Mirror register of DSCOCNSR.OWD1Nn 0: Indicates that the condition of overcurrent detection window notification 1 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 1 was satisfied	R
18	OWD2N	Channel n overcurrent detection window notification 2 Mirror register of DSCOCNSR.OWD2Nn 0: Indicates that the condition of overcurrent detection window notification 2 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 2 was satisfied	R
19	OWD3N	Channel n overcurrent detection window notification 3 Mirror register of DSCOCNSR.OWD3Nn 0: Indicates that the condition of overcurrent detection window notification 3 is not satisfied 1: Indicates that the condition of overcurrent detection window notification 3 was satisfied	R
23:20	—	These bits are read as 0.	R
24	OC0CMPL	Channel n overcurrent detect 0 lower limit compare result Mirror register of DSCOCRMR.OC0CMPLn 0: Indicates that the overcurrent data is not less than the lower limit 0 1: Indicates that the overcurrent data has fallen the lower limit 0	R
25	OC0CMPH	Channel n overcurrent detect 0 upper limit compare result Mirror register of DSCOCRMR.OC0CMPHn 0: Indicates that the overcurrent data doesn't exceed the upper limit 0 1: Indicates that the overcurrent data has exceeded the upper limit 0	R
26	OC1CMPL	Channel n overcurrent detect 1 lower limit compare result Mirror register of DSCOCRMR.OC1CMPLn 0: Indicates that the overcurrent data is not less than the lower limit 1 1: Indicates that the overcurrent data has fallen the lower limit 1	R

Bit	Symbol	Function	R/W
27	OC1CMPH	Channel n overcurrent detect 1 upper limit compare result Mirror register of DSCOCRMR.OC1CMPHn 0: Indicates that the overcurrent data doesn't exceed the upper limit 1 1: Indicates that the overcurrent data has exceeded the upper limit 1	R
28	OC2CMPL	Channel n overcurrent detect 2 lower limit compare result Mirror register of DSCOCRMR.OC2CMPLn 0: Indicates that the overcurrent data is not less than the lower limit 2 1: Indicates that the overcurrent data has fallen the lower limit 2	R
29	OC2CMPH	Channel n overcurrent detect 2 upper limit compare result Mirror register of DSCOCRMR.OC2CMPHn 0: Indicates that the overcurrent data doesn't exceed the upper limit 2 1: Indicates that the overcurrent data has exceeded the upper limit 2	R
31:30	—	These bits are read as 0.	R

The DSCSRCHn register is a mirror register of various registers in section 38.2.1. DSMIF Common Control Registers.

38.2.2.26 DSCSCRCHn : Status Clear Register Channel n (n = 0 to 2)

Base address: DSMIFm = 0x9000_8000 + 0x0800 × m (m = 0, 1)

Offset address: 0x430 + 0x140 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	CLROWD3N	CLROWD2N	CLROWD1N	CLROWD0N
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	CLROC2FH	CLROC2FL	CLROC1FH	CLROC1FL	CLROC0FH	CLROC0FL	—	—	—	—	CLRSCF	CLRCBUF	CLRCAUF	CLRDUF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CLRDUF	Channel n data update flag clear Mirror register of DSCSCR.CLRDUFn 0: Do nothing 1: Clear DSCSR.DUFn /DSCSRCHn.DUF	W
1	CLRCAUF	Channel n capture data A update flag clear Mirror register of DSCSCR.CLRCAUFn 0: Do nothing 1: Clear DSCSR.CAUFn /DSCSRCHn.CAUF	W
2	CLRCBUF	Channel n capture data B update flag clear Mirror register of DSCSCR.CLRCBUFn 0: Do nothing 1: Clear DSCSR.CBUFn /DSCSRCHn.CBUF	W
3	CLRSCF	Channel n short circuit detection flag clear Mirror register of DSCESCR.CLRSCFn 0: Do nothing 1: Clear DSCESR.SCFn / DSCSRCHn.SCF	W
7:4	—	The write value should be 0.	W
8	CLROC0FL	Channel n overcurrent lower limit detection flag 0 clear Mirror register of DSCOCESCR.CLROC0FLn 0: Do nothing 1: Clear DSCOCESR.OSC0FLn / DSCSRCHn.OC0FL	W
9	CLROC0FH	Channel n overcurrent upper limit exceeded flag 0 clear Mirror register of DSCOCESCR.CLROC0FHn 0: Do nothing 1: Clear DSCOCESR.OSC0FHn / DSCSRCHn.OC0FH	W

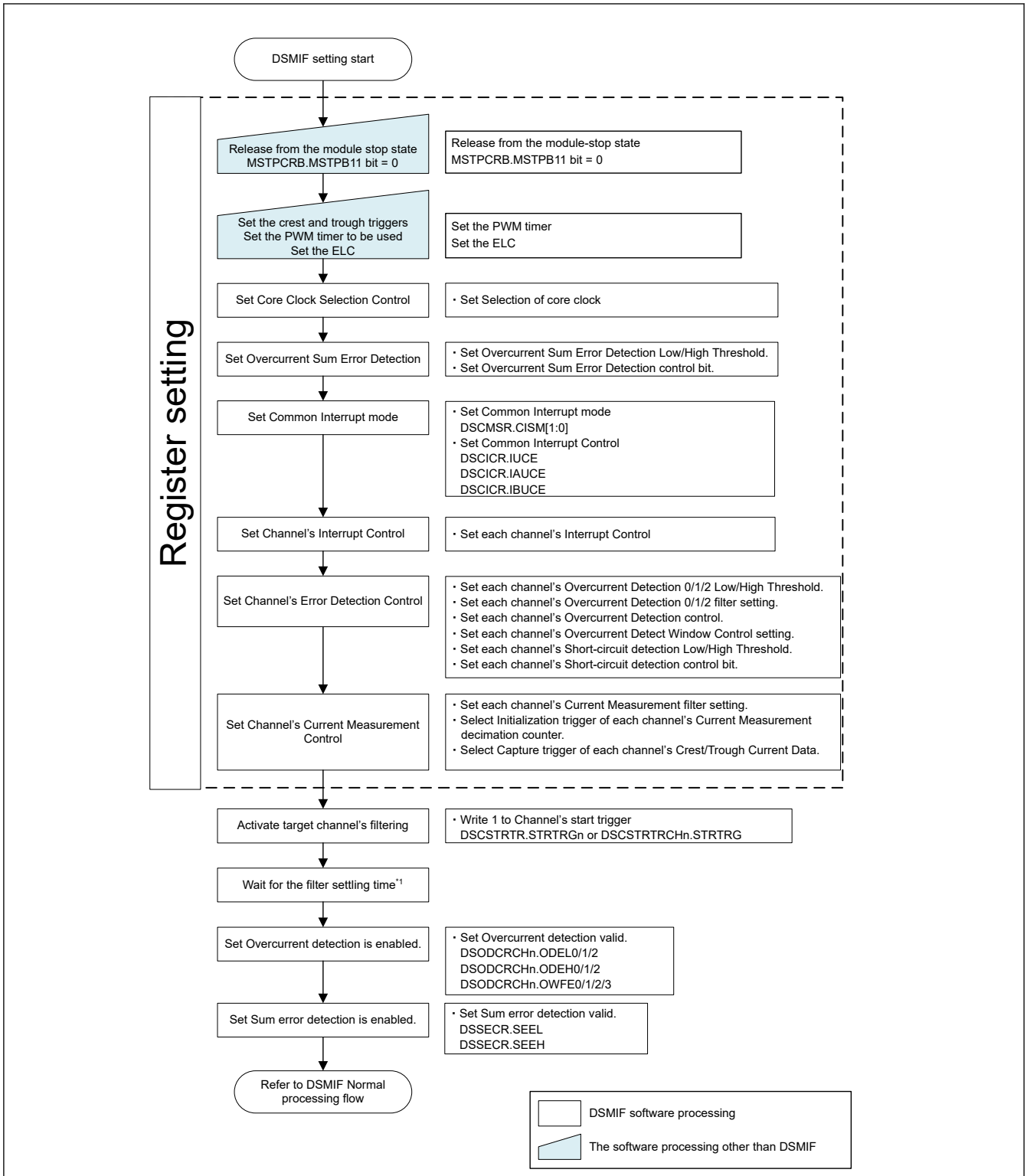
Bit	Symbol	Function	R/W
10	CLROC1FL	Channel n overcurrent lower limit detection flag 1 clear Mirror register of DSCOCESCR.CLROC1FLn 0: Do nothing 1: Clear DSCOCESR.OSC1FLn / DSCSRCHn.OC1FL	W
11	CLROC1FH	Channel n overcurrent upper limit exceeded flag 1 clear Mirror register of DSCOCESCR.CLROC1FHn 0: Do nothing 1: Clear DSCOCESR.OSC1FHn / DSCSRCHn.OC1FH	W
12	CLROC2FL	Channel n overcurrent lower limit detection flag 2 clear Mirror register of DSCOCESCR.CLROC2FLn 0: Do nothing 1: Clear DSCOCESR.OSC2FLn / DSCSRCHn.OC2FL	W
13	CLROC2FH	Channel n overcurrent upper limit exceeded flag 2 clear Mirror register of DSCOCESCR.CLROC2FHn 0: Do nothing 1: Clear DSCOCESR.OSC2FHn / DSCSRCHn.OC2FH	W
15:14	—	The write value should be 0.	W
16	CLROWD0N	Channel n overcurrent detection window notification 0 flag clear Mirror register of DSCOCNSCR.CLROWD0n 0: Do nothing 1: Clear DSCOCNSR.OWD0Nn / DSCSRCHn.OWD0N	W
17	CLROWD1N	Channel n overcurrent detection window notification 1 flag clear Mirror register of DSCOCNSCR.CLROWD1n 0: Do nothing 1: Clear DSCOCNSR.OWD1Nn / DSCSRCHn.OWD1N	W
18	CLROWD2N	Channel n overcurrent detection window notification 2 flag clear Mirror register of DSCOCNSCR.CLROWD2n 0: Do nothing 1: Clear DSCOCNSR.OWD2Nn / DSCSRCHn.OWD2N	W
19	CLROWD3N	Channel n overcurrent detection window notification 3 flag clear Mirror register of DSCOCNSCR.CLROWD3n 0: Do nothing 1: Clear DSCOCNSR.OWD3Nn / DSCSRCHn.OWD3N	W
31:20	—	The write value should be 0.	W

The DSCSCRCHn register is a mirror register of various registers in section 38.2.1. DSMIF Common Control Registers.

38.3 Operation

38.3.1 Start Flow

Figure 38.2 shows an example of a start procedure for DSMIF.

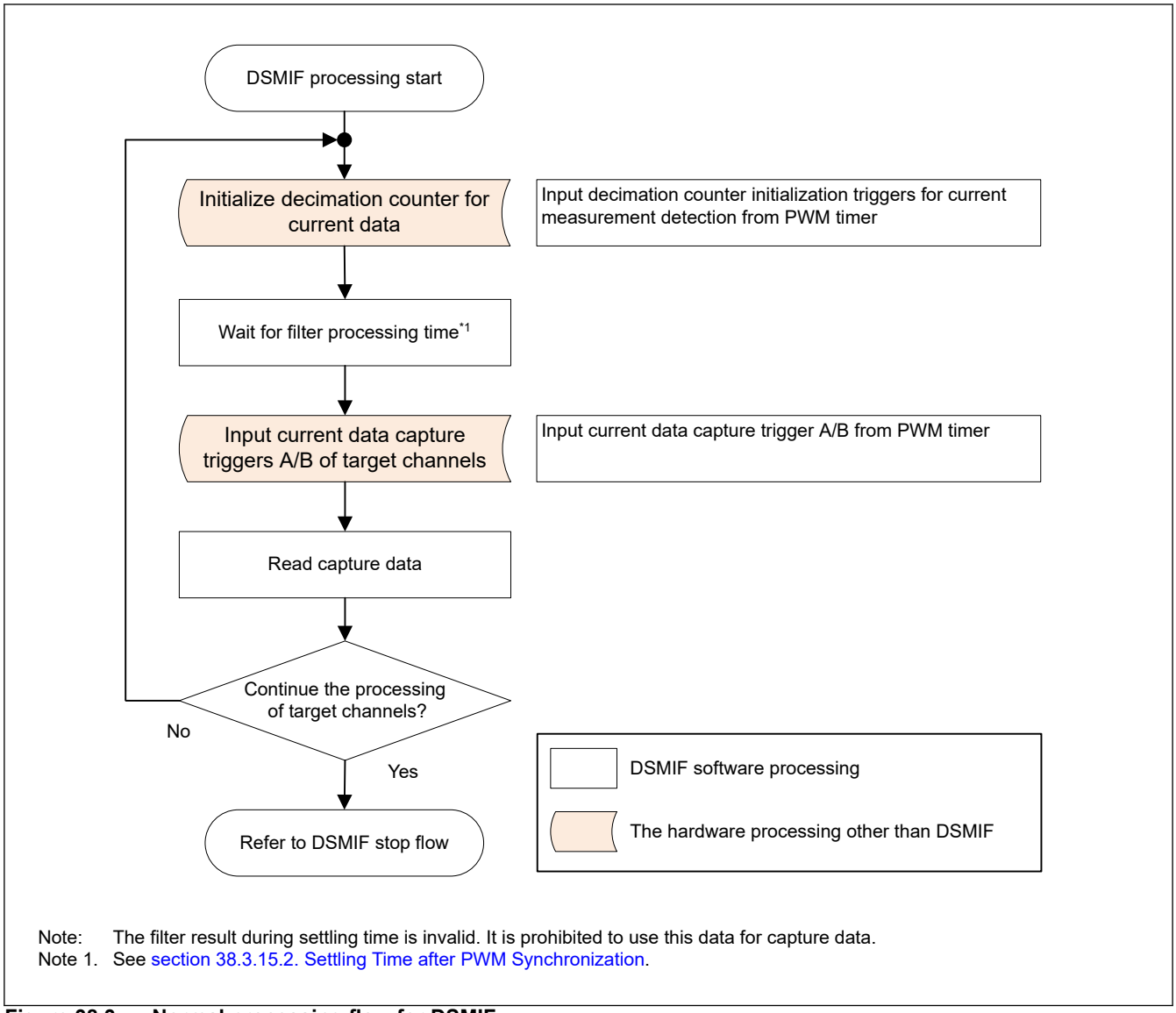


Note 1. See section 38.3.15.1. Settling Time of Channel Activation.

Figure 38.2 Start flow of DSMIF

38.3.2 Normal Processing Flow

Figure 38.3 shows an example of a normal processing procedure for DSMIF.



Note: The filter result during settling time is invalid. It is prohibited to use this data for capture data.
Note 1. See section 38.3.15.2. Settling Time after PWM Synchronization.

Figure 38.3 Normal processing flow for DSMIF

38.3.3 Stop Flow

Figure 38.4 shows an example of a stop procedure in channel synchronization control mode for DSMIF. Figure 38.5 shows the example of the stop procedure in channel individual control mode of DSMIF.

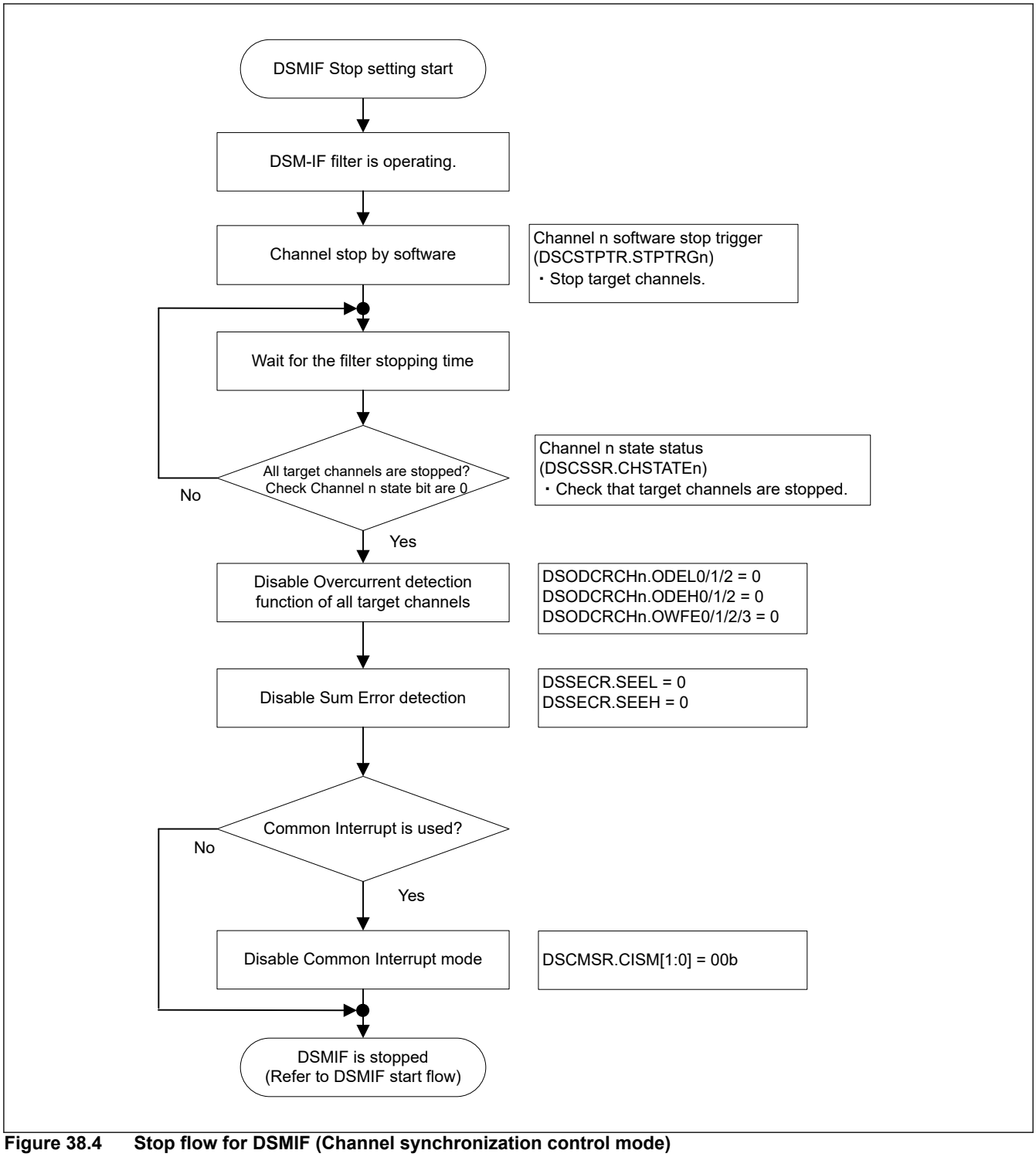


Figure 38.4 Stop flow for DSMIF (Channel synchronization control mode)

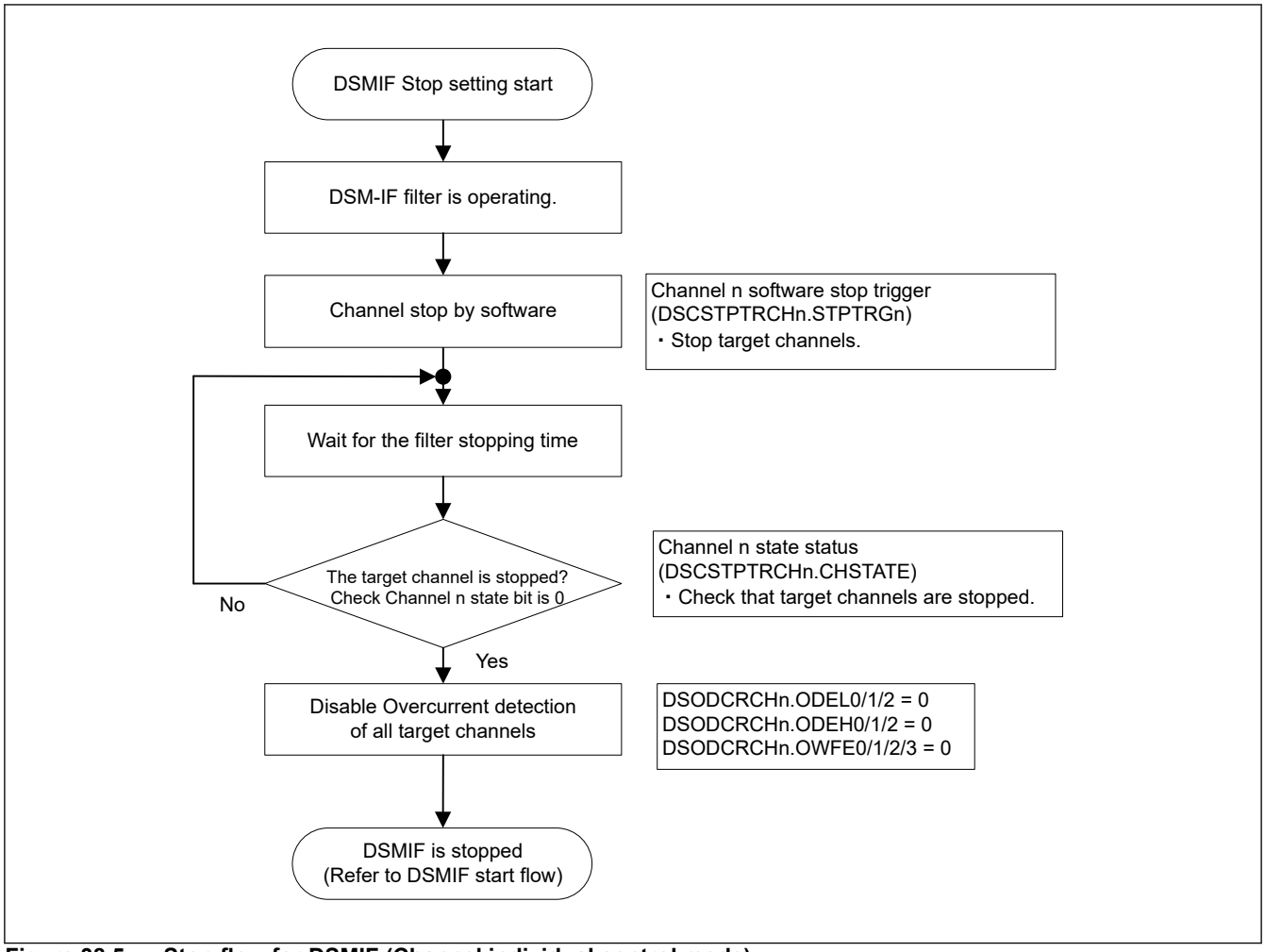
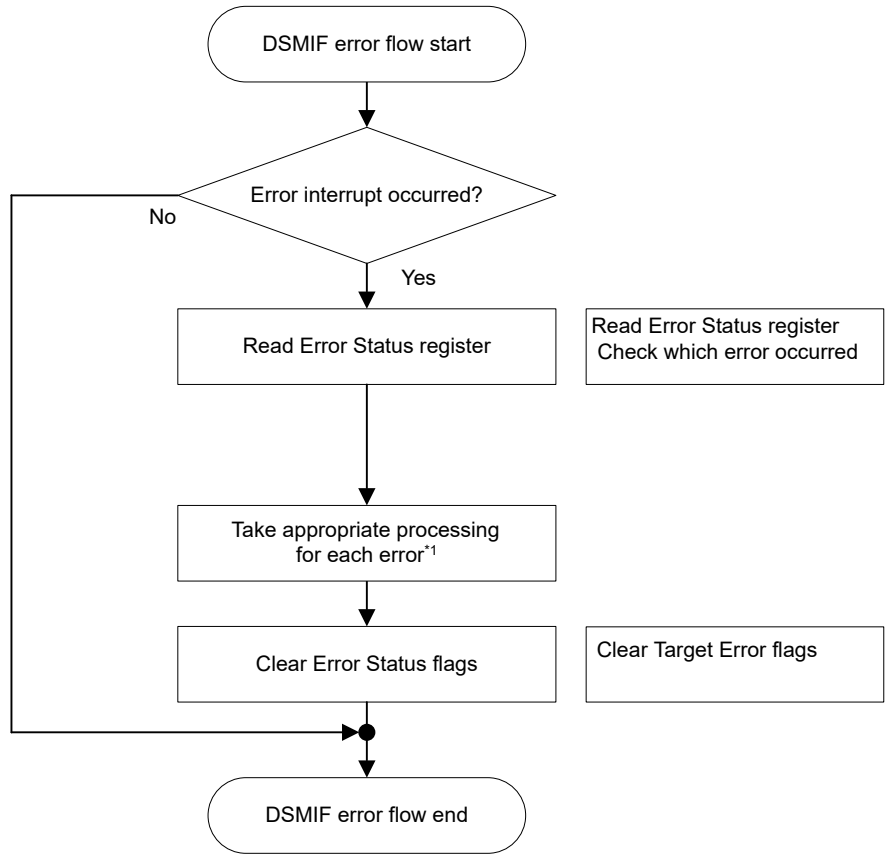


Figure 38.5 Stop flow for DSMIF (Channel individual control mode)

38.3.4 Error Flow

Figure 38.6 shows an example of an error procedure for DSMIF.



Note 1. The processing after each error detection depends on the user's system. Therefore, execute the error processing after the error detection according to each system.

Figure 38.6 Error flow for DSMIF

38.3.5 Clock Specification

When DSCMCCRCHn.CKDIR is set to 0 (clock slave mode), an external delta-sigma modulator clock is input to MCLKn. When DSCMCCRCHn.CKDIR is set to 1 (clock master mode), the frequency-divided core clock is output to MCLKn.

The frequency of the MCLKn can be selected by the DSCMCCRCHn.CKDIV[5:0] bits.

Table 38.3 shows the clock division ratio of master mode in DSMIF.

Table 38.3 Clock division ratio of Master mode in DSMIF

No.	Core clock (MHz)	MCLKn (MHz)	Supported core clock division ratio	Setting value (DSCMCCRCHn.CKDIV[5:0])
1	200	25	8	00_0011b
2		20	10	00_0100b
3		12.5	16	00_0111b
4		10	20	00_1001b
5		6.25	32	00_1111b
6		5	40	01_0011b

38.3.6 Channel Start and Channel Stop

The channel start and channel stop are controlled by the register in Table 38.4. Set the channels to be synchronized to 1 simultaneously. Set the channel start according to section 38.3.1. Start Flow. Set the channel stop according to section 38.3.3. Stop Flow.

Table 38.4 Channel start trigger and channel stop trigger

Channel mode	Category	Register bit name
Channel synchronization control mode	Channel start	DSCSTRTR.STRTRG0
		DSCSTRTR.STRTRG1
		DSCSTRTR.STRTRG2
	Channel stop	DSCSTPTR.STPTRG0
		DSCSTPTR.STPTRG1
		DSCSTPTR.STPTRG2
Channel individual control mode	Channel start	DSCSTRTRCHn.STRTRG
	Channel stop	DSCSTPTRCHn.STPTRG

38.3.7 Channel Synchronization

To synchronize the Sinc filters of multiple channels, use the following procedure. To allow for multiple channels to be synchronized, follow the procedure in section 38.3.1. Start Flow.

1. Set the same clock settings.
2. Select the same trigger for division counter initialization trigger.
   Set the same value to Division counter initialization trigger select bits (DSCMCTCRCHn.DITSEL[1:0]).
3. Select the same trigger for capture trigger A and B.
   (a) Set the same value to the Current Capture Trigger A Select bits (DSCMCTCRCHn.CTSELA[2:0]).
   (b) Set the same value to the Current Capture Trigger B Select bits (DSCMCTCRCHn.CTSELB[2:0]).
4. Activate the channels simultaneously
   Write 1 to Channel n start trigger (DSCSTRTR.STRTRGn) at the same time.

38.3.8 Data Input Control

By using the DSCMCCRCHn.SEDGE bit, data receive can be selected to receive on the falling edge of MCLKn or the rising edge of MCLKn. Data input control receives clock and data from DSM. Synchronized data and clock are output to the sinc filter and short-circuit detection.

38.3.9 Sinc Filter

The differentiation stage is operated by decimation clock (the frequency is 1/M of MCLKn). M is the decimation ratio and is set in the DSCMFCRCHn.CMDEC[7:0] bits for current measurement or in the DSOCFCRCHn.OCDEC[7:0] bits for overcurrent detection. The filter result is stored in register every decimation clock. In the case of current data, an interrupt is output at the same time when the filter result is stored.

Figure 38.7 shows a block diagram for the Sinc Filter.

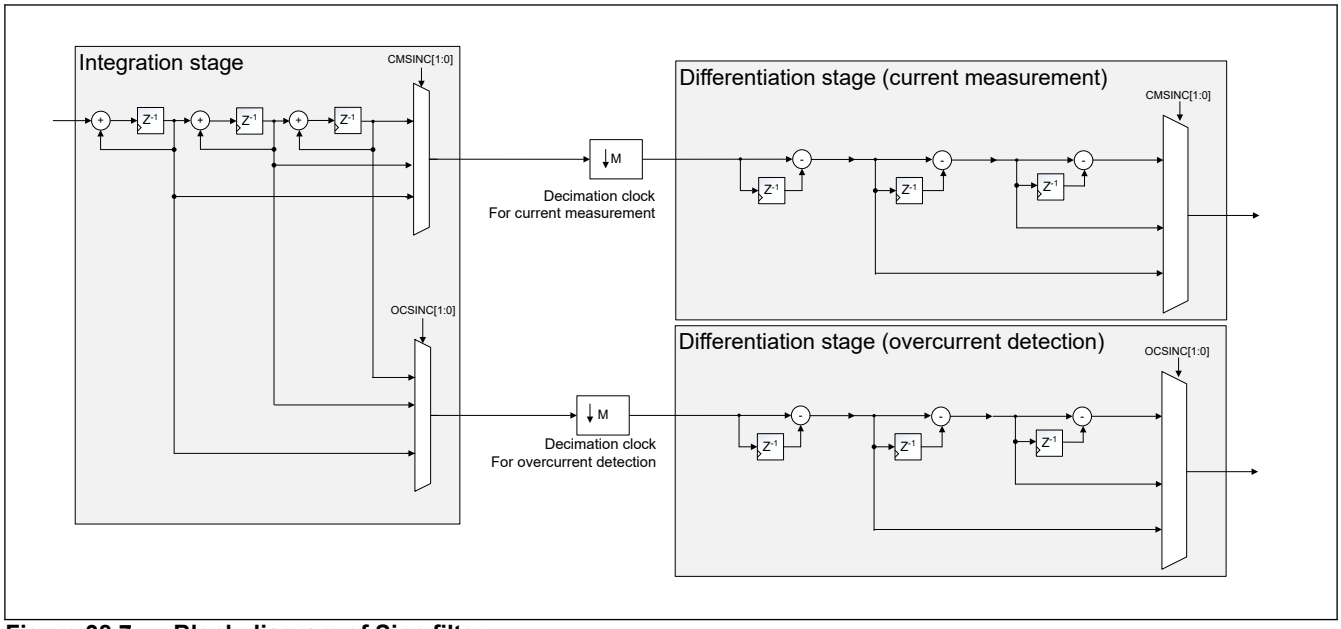


Figure 38.7 Block diagram of Sinc filter

38.3.10 Available Settings of Sinc Filter

Table 38.5 lists the available settings of the sinc filter and data shifting.

Table 38.5 Available settings of Sinc filter (1 of 2)

Sinc filter order	Decimation rate	Effective bits	Peak value (Decimal)	Peak value (Hexadecimal)	Stored data		CMSH[4:0] OCSH[4:0]	Resolution
					DFS = 0	DFS = 1		
3 (Sinc3)	4	[6:0]	63*1	000003F	[5:0], 00_0000_0000	00_0000_0000, [5:0]	0x12	6 bits
	5	[6:0]	125	000007D	[6:0], 0_0000_0000	0_0000_0000, [6:0]	0x11	7 bits
	6	[7:0]	216	00000D8	[7:0], 0000_0000	0000_0000, [7:0]	0x10	8 bits
	7	[8:0]	343	0000157	[8:0], 000_0000	000_0000, [8:0]	0x0F	9 bits
	8	[9:0]	511*1	00001FF	[8:0], 000_0000	000_0000, [8:0]	0x0F	9 bits
	9-10	[9:0]	1000	00003E8	[9:0], 00_0000	00_0000, [9:0]	0x0E	10 bits
	11-12	[10:0]	1728	00006C0	[10:0], 0_0000	0_0000, [10:0]	0x0D	11 bits
	13-15	[11:0]	3375	0000D2F	[11:0], 0000	0000, [11:0]	0x0C	12 bits
	16	[12:0]	4095*1	0000FFF	[11:0], 0000	0000, [11:0]	0x0C	12 bits
	17-20	[12:0]	8000	0001F40	[12:0], 000	000, [12:0]	0x0B	13 bits
	21-25	[13:0]	15625	0003D09	[13:0], 00	00, [13:0]	0x0A	14 bits
	26-31	[14:0]	29791	000745F	[14:0], 0	0, [14:0]	0x09	15 bits
	32	[15:0]	32767*1	0007FFF	[14:0], 0	0, [14:0]	0x09	15 bits
	33-40	[15:0]	64000	000FA00	[15:0]		0x08	16 bits
	41-50	[16:0]	125000	001E848	[16:1]		0x07	16 bits
	51-63	[17:0]	250047	003D0BF	[17:2]		0x06	16 bits
	64	[18:0]	262143*1	003FFFF	[17:2]		0x06	16 bits
	65-80	[18:0]	512000	007D000	[18:3]		0x05	16 bits
	81-101	[19:0]	1030301	00FB89D	[19:4]		0x04	16 bits
	102-127	[20:0]	2048383	01F417F	[20:5]		0x03	16 bits
	128	[21:0]	2097151*1	01FFFFF	[20:5]		0x03	16 bits
	129-161	[21:0]	4173281	03FADE1	[21:6]		0x02	16 bits
	162-203	[22:0]	8365427	07FA573	[22:7]		0x01	16 bits
	204-255	[23:0]	16581375	0FD02FF	[23:8]		0x00	16 bits
	256	[24:0]	16777215*1	0FFFFFF	[23:8]		0x00	16 bits
2 (Sinc2)	16	[7:0]	255*1	00000FF	[7:0], 0000_0000	0000_0000, [7:0]	0x10	8 bits
	17-20	[8:0]	400	0000190	[8:0], 000_0000	000_0000, [8:0]	0x0F	9 bits
	21-31	[9:0]	961	00003C1	[9:0], 00_0000	00_0000, [9:0]	0x0E	10 bits
	32	[10:0]	1023*1	00003FF	[9:0], 00_0000	00_0000, [9:0]	0x0E	10 bits
	33-40	[10:0]	1600	0000640	[10:0], 0_0000	0_0000, [10:0]	0x0D	11 bits
	41-63	[11:0]	3969	0000F81	[11:0], 0000	0000, [11:0]	0x0C	12 bits
	64	[12:0]	4095*1	0000FFF	[11:0], 0000	0000, [11:0]	0x0C	12 bits
	65-90	[12:0]	8100	0001FA4	[12:0], 000	000, [12:0]	0x0B	13 bits
	91-127	[13:0]	16129	0003F01	[13:0], 00	00, [13:0]	0x0A	14 bits
	128	[14:0]	16383*1	0003FFF	[13:0], 00	00, [13:0]	0x0A	14 bits
	129-181	[14:0]	32761	0007FF9	[14:0], 0	0, [14:0]	0x09	15 bits
	182-255	[15:0]	65025	000FE01	[15:0]		0x08	16 bits
	256	[16:0]	65535*1	000FFFF	[15:0]		0x08	16 bits

Table 38.5 Available settings of Sinc filter (2 of 2)

Sinc filter order	Decimation rate	Effective bits	Peak value (Decimal)	Peak value (Hexadecimal)	Stored data		CMSH[4:0] OCSH[4:0]	Resolution
					DFS = 0	DFS = 1		
1 (Sinc1)	32	[5:0]	31*1	000001F	[4:0], 000_0000_0000	000_0000_0000, [4:0]	0x13	5 bits
	33-63	[5:0]	63	000003F	[5:0], 00_0000_0000	00_0000_0000, [5:0]	0x12	6 bits
	64	[6:0]	63*1	000003F	[5:0], 00_0000_0000	00_0000_0000, [5:0]	0x12	6 bits
	65-127	[6:0]	127	000007F	[6:0], 0_0000_0000	0_0000_0000, [6:0]	0x11	7 bits
	128	[7:0]	127*1	000007F	[6:0], 0_0000_0000	0_0000_0000, [6:0]	0x11	7 bits
	129-255	[7:0]	255	00000FF	[7:0], 0000_0000	0000_0000, [7:0]	0x10	8 bits
	256	[8:0]	255*1	00000FF	[7:0], 0000_0000	0000_0000, [7:0]	0x10	8 bits

Note 1. When the filter result is the maximum value, it is rounded to the maximum value of -1.

Figure 38.8 shows the relationship between the filter settings and the stored values.

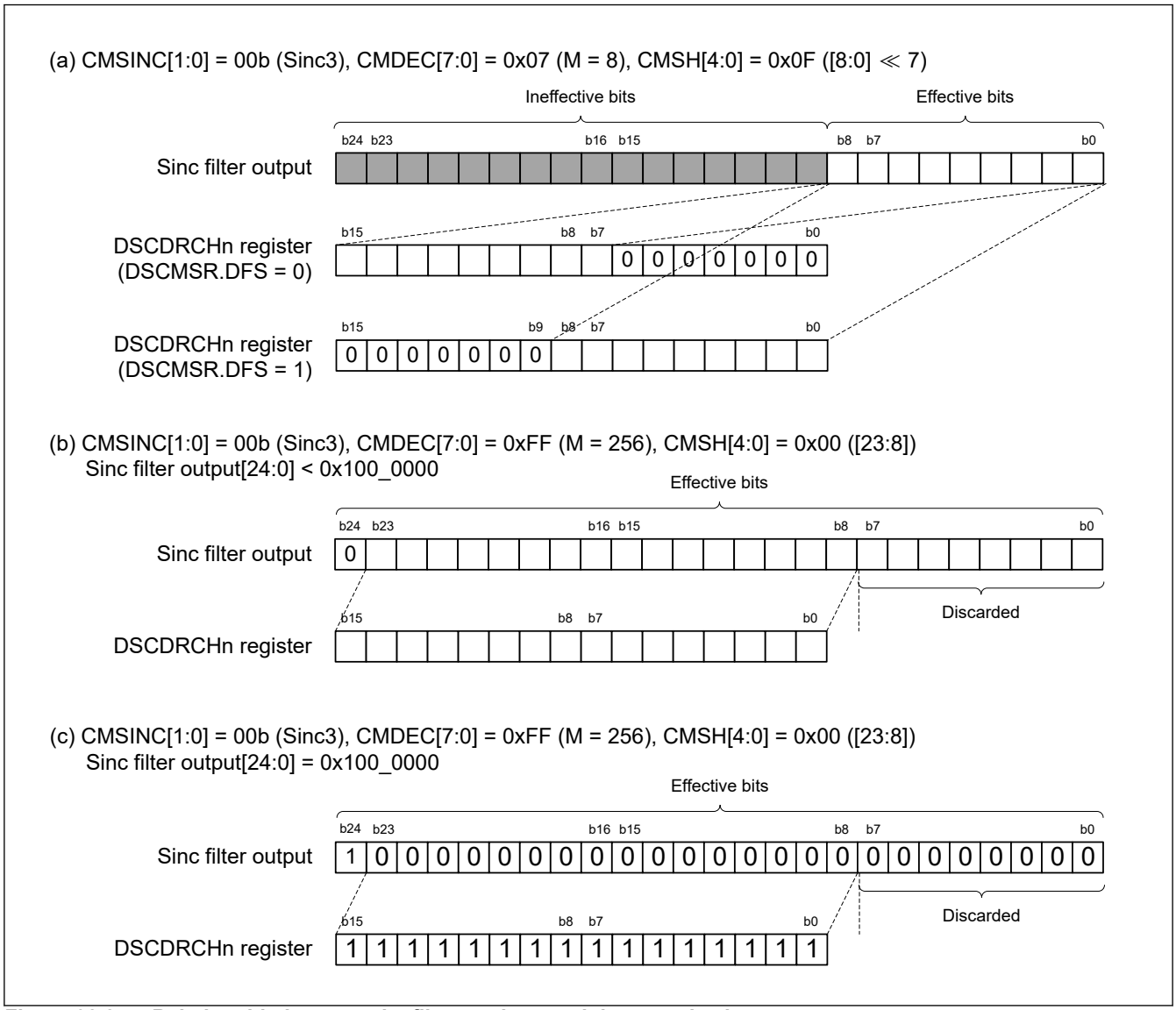


Figure 38.8 Relationship between the filter settings and the stored values

38.3.11 Current Measurement Data Processing

Each channel of DSMIF has the data registers specified in Table 38.6. Capture trigger A is selected by DSCMCTCRCHn.CTSELA[2:0] and capture trigger B is selected by DSCMCTCRCHn.CTSELB[2:0].

The trigger capturing function allows the capture of the values for current at the time of capture trigger A and capture trigger B using the event link controller (ELC) to set the capture trigger A or capture trigger B.

Table 38.6 Current measurement data list

No.	Register name	Symbol	Stored data
1	Current Data register	DSCDRCHn	The measured values for current
2	Capture Current Data Register A	DSCCDRACHn	The captured value of Current Data register by capture trigger A from the internal timer module
3	Capture Current Data Register B	DSCCDRBCHn	The captured value of Current Data register by capture trigger B from the internal timer module

38.3.12 Short-circuit Detection

When the short-circuit detection enable (DSEDCRCHn.SDE) is valid, the short-circuit detection can operate. A dedicated 13-bit counter counts consecutive 0's or 1's input from dsmdata n (n = 0 to 2). When the number of consecutive 0's exceeds the value set in the DSSCTSRCHn.SCNTL[12:0] register or the number of consecutive 1's exceeds the value set in the DSSCTSRCHn.SCNTH[12:0] register, a short-circuit detection interrupt request is generated.

38.3.13 Overcurrent Detection

The overcurrent detection function is used for abnormally detection. Overcurrent data is generated by the sinc filter result that is setting in section 38.3.9. Sinc Filter. The overcurrent data is compared with the value of the upper limit value or the lower limit value. The interrupt of overcurrent detection is output by these results.

It has 3 sets of upper limit value and lower limit value for each channel. And the interrupts can be output from each comparison results.

In addition, the overcurrent detection function also has overcurrent detection window notification function that notifies within range or out of range from the comparison result of the upper limit and the lower limit.

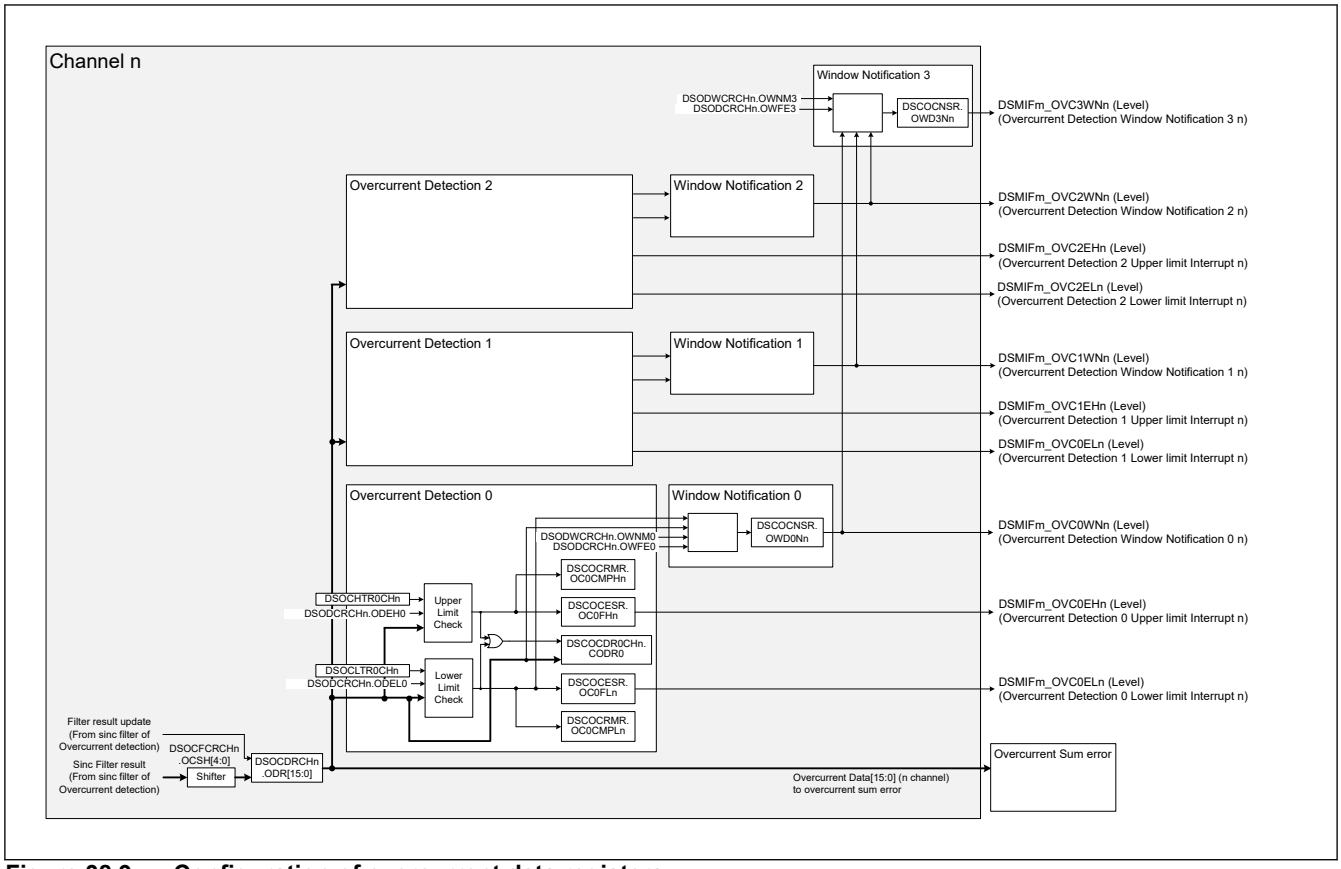


Figure 38.9 Configuration of overcurrent data registers

The measure values from current are stored as 16-bit values obtained by shifting the 25-bit data output from the sinc-filter in accordance with the setting of the DSOCFCRCHn.OCSH[4:0] bits.

The format of stored data can be selected by the DSCMSR.DFS bit. Either left-justified or right-justified can be selected.

38.3.13.1 Overcurrent Detection 0 to 2

Overcurrent detection has three upper limit excess judgments and three lower limit judgments for each channel. Table 38.7 shows the list of overcurrent detection.

Table 38.7 List of overcurrent detection

Function	Upper Lower	Enable	Comparator table	Error flag	Comparator result	Interrupt	Error capture data
Overcurrent Detection 0	Upper	DSODCRCHn.ODEH0	DSOCHTR0CHn.OCMPTBH0	DSCOCESR.OC0FHn	DSCOCRMR.OC0CMPHn	OVC0EHn	DSCOCDR0CHn.CODR0
	Lower	DSODCRCHn.ODEL0	DSOCHTR0CHn.OCMPTBL0	DSCOCESR.OC0FLn	DSCOCRMR.OC0CMPLn	OVC0ELn	
Overcurrent Detection 1	Upper	DSODCRCHn.ODEH1	DSOCHTR1CHn.OCMPTBH1	DSCOCESR.OC1FHn	DSCOCRMR.OC1CMPHn	OVC1EHn	DSCOCDR1CHn.CODR1
	Lower	DSODCRCHn.ODEL1	DSOCHTR1CHn.OCMPTBL1	DSCOCESR.OC1FLn	DSCOCRMR.OC1CMPLn	OVC1ELn	
Overcurrent Detection 2	Upper	DSODCRCHn.ODEH2	DSOCHTR2CHn.OCMPTBH2	DSCOCESR.OC2FHn	DSCOCRMR.OC2CMPHn	OVC2EHn	DSCOCDR2CHn.CODR2
	Lower	DSODCRCHn.ODEL2	DSOCHTR2CHn.OCMPTBL2	DSCOCESR.OC2FLn	DSCOCRMR.OC2CMPLn	OVC2ELn	

The overcurrent detection function is explained using overcurrent detection 0 as an example.

When the overcurrent upper limit detection 0 enable (DSODCRCHn.ODEH0) is valid, the overcurrent upper limit detection can operate. And when the overcurrent lower limit detection 0 enable (DSODCRCHn.ODEL0) is valid, the overcurrent lower limit detection can operate.

The current data is stored in the DSOCDRCHn as a 16-bit data shifted from 25-bit data output from sinc filter for overcurrent detection in accordance with the setting of the DSOCFCRCHn.OCSH[4:0] bits.

An overcurrent upper limit detection interrupt is requested when the value in the DSOCDRCHn register is upper than the value of OCMPTBH0.

An overcurrent lower limit detection interrupt is requested when the value in the DSOCDRCHn register is lower than the value of OCMPTBL0.

Overcurrent detection has the following two statuses.

1. Error flags that are set when the upper limit is exceeded / less than the lower limit is detected and cleared with the clear bit.
2. Monitor bits that return the comparison result for each decimation cycle.

38.3.13.2 Overcurrent Detection Window Notification

Overcurrent detection window notification is event notification function to notify within a range or out of range by comparison with upper limit and lower limit of overcurrent detection. There are four kinds of notification for each channel. Table 38.8 shows a list of overcurrent detection notification.

Table 38.8 List of overcurrent detection notification

Function	Enable	Mode bit	Flag	Notification
Overcurrent detection Window notification 0	DSODCRCHn.OWFE0	DSODWCRCHn.OWNM0	DSCOCNSR.OWD0Nn	OVC0WNn
Overcurrent detection Window notification 1	DSODCRCHn.OWFE1	DSODWCRCHn.OWNM1	DSCOCNSR.OWD1Nn	OVC1WNn
Overcurrent detection Window notification 2	DSODCRCHn.OWFE2	DSODWCRCHn.OWNM2	DSCOCNSR.OWD2Nn	OVC2WNn
Overcurrent detection Window notification 3	DSODCRCHn.OWFE3	DSODWCRCHn.OWNM3	DSCOCNSR.OWD3Nn	OVC3WNn

Overcurrent detection window notification 0 to 2 will be generated by comparison with upper limit and lower limit for overcurrent detection 0 to 2. The notification can be set for within a range or out of range as Figure 38.10.

1. OWNMm = 0: (overcurrent data channel n < lower limit m) or (overcurrent data channel n > upper limit m)
2. OWNMm = 1: lower limit m ≤ overcurrent data channel n ≤ upper limit m

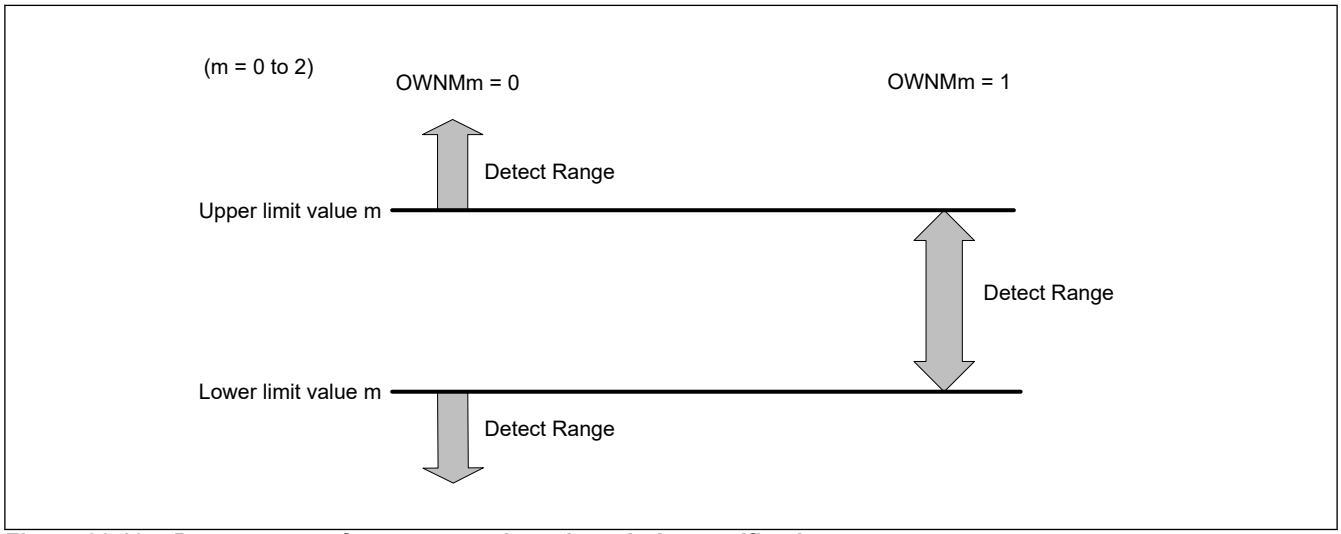


Figure 38.10 Detect range of overcurrent detection window notification

38.3.14 Overcurrent Sum Error Detection

When the overcurrent sum error upper limit detection enable (DSSECR.SEEH) is valid, the overcurrent sum error upper limit detection can operate. When the overcurrent sum error lower limit detection enable (DSSECR.SEEL) is valid, the

overcurrent sum error lower limit detection can operate. An overcurrent upper limit sum error detection interrupt is requested when the sum of the values for each mode (see Table 38.9) is higher than the value of SCMPTBH.

An overcurrent lower limit sum error detection interrupt is requested when the sum of the values for each mode is lower than the value of SCMPTBL.

Table 38.9 List of sum error mode detect mode and capture data

SEDM[2:0]	Sum error detect mode	CH0	CH1	CH2	DSSECDR0	DSSECDR1	DSSECDR2
000b	Three value sum error detect	✓	✓	✓	DSOCDRCH0	DSOCDRCH1	DSOCDRCH2
001b	Two value sum error detect	✓	✓	—	DSOCDRCH0	DSOCDRCH1	—*1
010b	One channel error detect	✓	—	—	DSOCDRCH0	—	—
011b		—	✓	—	DSOCDRCH1	—	—
100b		—	—	✓	DSOCDRCH2	—	—

Note 1. Do not store data, hold previous value.

38.3.15 Settling Time

38.3.15.1 Settling Time of Channel Activation

Settling time of channel activation is the time from when the channel is activated until the filter result is stable and output.

The settling time is calculated as follows.

Table 38.10 Settling time of channel activation

Sinc order	Processing time of settling time of channel activation
$Sinc^1$	(MCLK period × Decimation ratio) × sinc order + MCLK period × 1 cycle + PCLKH period × 18 cycles
$Sinc^3$ $Sinc^2$	(MCLK period × Decimation ratio) × (sinc order + 1) + MCLK period × 1 cycle + PCLKH period × 18 cycles

38.3.15.2 Settling Time after PWM Synchronization

Settling time after PWM synchronization is the time from the input of the decimation dividing counter initialization trigger to the stable output of the filter result.

Table 38.11 Settling time after PWM synchronization

Processing time of settling time of PWM synchronization
(MCLK period × Decimation ratio) × sinc order + PCLKH period × 13 cycles

38.3.16 Common Interrupt

The common interrupt is a function that outputs a single common interrupt that is output for each channel. Common interrupts are provided for the following data update interrupts.

When using common interrupts, it is recommended that channel-specific data update interrupts are disabled.

Table 38.12 Data update common interrupt

Function	Interrupt	Pulse or Level	Enable
Current data	Current data register update common interrupt	Pulse	DSCICR.IUCE
Capture data A	Current capture data register A update common interrupt	Pulse	DSCICR.IAUCE
Capture data B	Current capture data register B update common interrupt	Pulse	DSCICR.IBUCE

Common interrupts have two modes, 3ch synchronization mode and 2ch synchronization mode.

As shown in Table 38.13, common interrupts can be output when synchronization mode is enabled and all data on the target channel has been updated.

Table 38.13 Mode of common interrupt

CISM[1:0]	Function	Target channel			Target function		
		CH0	CH1	CH2	Current data	Capture data A	Capture data B
00b	Common interrupts can not be used	—	—	—	—	—	—
01b	Three channel synchronous mode	✓	✓	✓	✓	✓	✓
10b	Two channel synchronous mode	✓	✓	—	✓	✓	✓
11b	Setting prohibited	—	—	—	—	—	—

The operation timing of the common interrupt in the three channel synchronization mode is explained using the current data update common interrupt as an example. When the current data is updated for each of the three channels, the current data common interrupt is output at the same timing as the latest channel current data update interrupt.

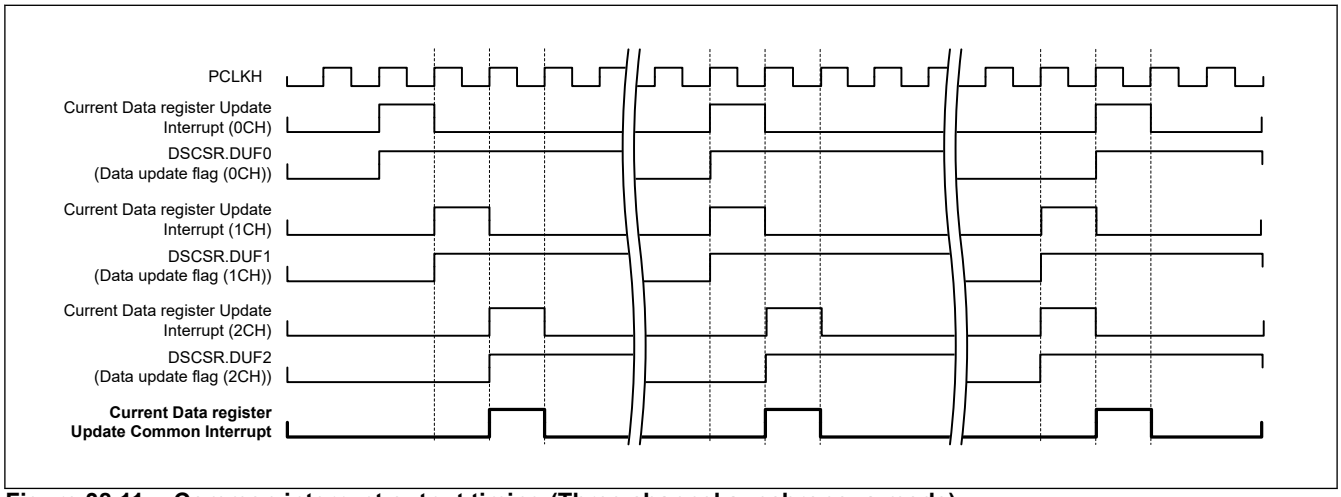


Figure 38.11 Common interrupt output timing (Three channel synchronous mode)

Similarly, in the two channel synchronous mode, the current data common interrupt is output at the same timing as the latest channel current data update interrupt.

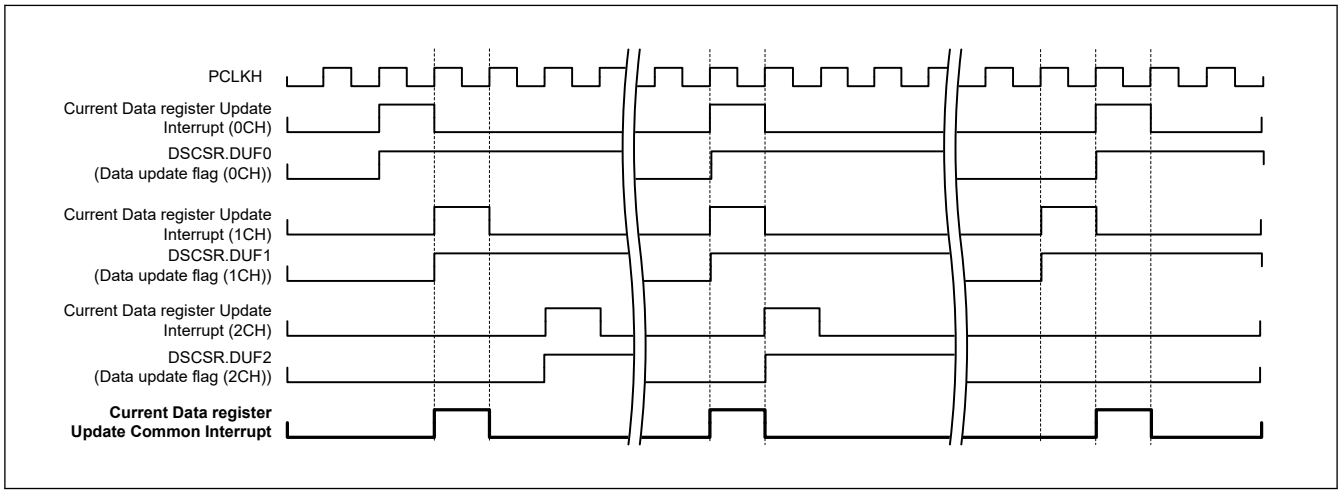


Figure 38.12 Common interrupt output timing (Two channel synchronous mode)

39. 12-Bit A/D converter (ADC12)

39.1 Overview

This LSI incorporates two unit of a 12-bit successive approximation A/D converter. Analog inputs of up to 4 channels are selectable for unit 0 and unit 1 (LLPP of CPU).

The 12-bit A/D converter converts analog inputs of up to 4 selected channels to a 12-bit digital value through successive approximation.

The 12-bit A/D converter has three operating modes: single scan mode in which analog inputs selected arbitrarily are converted only once in ascending order of channel number; continuous scan mode in which analog inputs selected arbitrarily are continuously converted in ascending order of channel number; and group scan mode in which analog inputs are arbitrarily selected and divided into two groups (group A and group B) or three groups (group A, group B and group C), and then analog inputs of channels selected for each group are converted in ascending order of channel number.

In group scan mode, the conditions for scanning start of each groups (A, B or A, B, C) (synchronous trigger) can be independently selected, thus allowing A/D conversion of each groups to be started independently. In addition to the above actions, the group priority operation in group scan mode accepts the start of scanning the priority group during the scan of the low priority group, interrupts the scan of the low priority group, and starts the scan of the priority group. The priority of group priority operations is group A > group B > group C. In group priority operation, if the scan start of group B is accepted during the scan of group C, the scan of group C is interrupted and the scan of group B is started. If the scan start of group A is accepted during the scan of group C, the scan of group A is started. The scan of Group C is interrupted and the scan of Group A is started. Similarly, if the scan start of Group A is accepted during the scan of Group B, the scan of Group B is interrupted and the scan of Group A is started. The interrupted group scan is able to resume after the priority group scan is complete also.

In double trigger mode, one analog input channel arbitrarily selected is converted in single scan mode or group scan mode (group A), and the resulting data of A/D conversion started by the first and second synchronous triggers are stored into different registers (duplication of A/D conversion data).

The compare function is a function that outputs an interrupt (ADCn_CMPAI/ADCn_CMPBI) when the A/D conversion value of the selected channel by specifying the upper and lower reference values for each window A/B matches the comparison conditions. The event (ADCn_WCMPM/ADCn_WCMPUM) is output according to the event conditions (A or B, A and B, A exor B). In addition, a comparator operation that compares the A/D conversion value with the lower reference value is also possible.

Table 39.1 lists the specifications of the 12-bit A/D converter and Table 39.2 indicates the functions of the 12-bit A/D converter. Figure 39.1 shows a block diagram of the 12-bit A/D converter.

Table 39.1 ADC12 specifications (1 of 2)

Item	Description
Number of units	Two units, 0 and 1
Input channels	• Unit 0: Up to 4 channels in LLPP of CPU0 • Unit 1: Up to 4 channels in LLPP of CPU0
A/D conversion method	Successive approximation method
Resolution	12 bits
Conversion time	0.84 µs: Channel-dedicated sample-and-hold circuits not in use 1.52 µs: Channel-dedicated sample-and-hold circuits in use At PCLKADC in the 12-bit A/D converter: 25 MHz
A/D conversion clock	PCLKADC (25 MHz or 18.75 MHz)

Table 39.1 ADC12 specifications (2 of 2)

Item	Description
Operating modes	• Single scan mode: A/D conversion is performed only once on the analog inputs of arbitrarily selected channels • Continuous scan mode: A/D conversion is performed repeatedly on the analog inputs of arbitrarily selected channels • Group scan mode: Two (groups A and B) or three (groups A, B, and C) can be selected as the number of the groups to be used. Only the combination of groups A and B can be selected when the number of the groups is two. Analog inputs that are arbitrarily selected are divided into two groups (group A and B) or three groups (group A, B, and C), and A/D conversion of the analog input selected on a group basis is performed only once. The conditions for scanning start of groups A, B, and C (synchronous trigger) can be independently selected, thus allowing A/D conversion of each group to be started independently. • Group scan mode (when Group A is given priority): If a priority-group trigger is input during scanning of the low-priority group, scan of the low-priority group is stopped and scan of the priority group is started. The priority order is group A (highest) > group B > group C (lowest). Whether or not to restart scanning of the low-priority group after processing for the high-priority group completes, is selectable. Rescan can also be set to start either from the beginning of the selected channel or the channel on which A/D conversion is not completed.
Conditions for A/D conversion start	• Software trigger • Synchronous triggers Trigger by the multi-function timer pulse unit (MTU3) and event link controller (ELC) • Asynchronous trigger A/D conversion can be triggered by the external trigger ADTRG0# (unit 0) and ADTRG1# (unit1).
Functions	• Dedicated sample-and-hold function with optional constant sampling and 3 channels in unit 0 and 1 • Variable sampling state count • Selectable A/D-converted value addition mode or average mode • Double-trigger mode (duplication of A/D conversion data) • Automatic clear function for A/D data registers • Digital comparison of values in the comparison and data registers, and between values in the data registers • Overwrite checking function of the A/D data register
Interrupt sources and ELC events	• ADCn_ADI (for ELC, trigger of interrupt is slightly different for CPU SPI) • ADCn_GBADI: A/D scan end interrupt for Group B • ADCn_GCADI: A/D scan end interrupt for Group C • ADCn_CMPAI: Window A compare match • ADCn_CMPBI: Window B compare match • ADCn_WCMPM: compare match • ADCn_WCMPUM: compare mismatch
ELC interface	• An ELC event is generated upon completion of all scans • Able to start scanning by a trigger from the ELC
Module-stop function	Module-stop state can be set to reduce power consumption

Table 39.2 Function of ADC12 (1 of 2)

Parameter	Unit 0 (ADC120)	Unit 1 (ADC121)
Analog input channel	AN000 to AN003	AN100 to AN103

Table 39.2 Function of ADC12 (2 of 2)

Parameter			Unit 0 (ADC120)	Unit 1 (ADC121)
Conditions for A/D conversion start	Software	Software trigger	Enabled	Enabled
	Asynchronous trigger	Trigger input pin (Use only synchronous trigger as the scan start in group scan mode)	ADTRG0#	ADTRG1#
	Synchronous	Compare match with or input capture to MTU0.TGRA	TRGA0N	TRGA0N
		Compare match with or input capture to MTU1.TGRA	TRGA1N	TRGA1N
		Compare match with or input capture to MTU2.TGRA	TRGA2N	TRGA2N
		Compare match with or input capture to MTU3.TGRA	TRGA3N	TRGA3N
		Compare match with or input capture to MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	TRGA4N	TRGA4N
		Compare match with or input capture to MTU6.TGRA	TRGA6N	TRGA6N
		Compare match with or input capture to MTU7.TGRA or underflow (trough) of MTU7.TCNT in complementary PWM mode	TRGA7N	TRGA7N
		Compare match with MTU0.TGRE	TRG0N	TRG0N
		Compare match between MTU4.TADCORA and MTU4.TCNT	TRG4AN	TRG4AN
		Compare match between MTU4.TADCORB and MTU4.TCNT	TRG4BN	TRG4BN
		Compare match between MTU4.TADCORA and MTU4.TCNT or compare match between MTU4.TADCORB and MTU4.TCNT	TRG4AN or TRG4BN	TRG4AN or TRG4BN
		Compare match between MTU4.TADCORA and MTU4.TCNT and compare match between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is used)	TRG4ABN	TRG4ABN
		Compare match between MTU7.TADCORA and MTU7.TCNT	TRG7AN	TRG7AN
		Compare match between MTU7.TADCORB and MTU7.TCNT	TRG7BN	TRG7BN
		Compare match between MTU7.TADCORA and MTU7.TCNT or compare match between MTU7.TADCORB and MTU7.TCNT	TRG7AN or TRG7BN	TRG7AN or TRG7BN
		Compare match between MTU7.TADCORA and MTU7.TCNT and compare match between MTU7.TADCORB and MTU7.TCNT (when interrupt skipping function 2 is used)	TRG7ABN	TRG7ABN
		ELC trigger	ADC0 A, ADC0 B	ADC1 A, ADC1 B
Channel-dedicated sample-and-hold function	Target channel		AN000 to AN002	AN100 to AN102
Interrupt			ADC0_ADI ADC0_GBADI ADC0_GCADI ADC0_CMPAI ADC0_CMPBI	ADC1_ADI ADC1_GBADI ADC1_GCADI ADC1_CMPAI ADC1_CMPBI
Output to ELC			ADC0_ADI*1 ADC0_WCMPM ADC0_WCMPUM	ADC1_ADI*1 ADC1_WCMPM ADC1_WCMPUM
Module-stop function settings			Available	Available

Note 1. Trigger of event is slightly different from normal interrupt for CPU SPI.

Table 39.3 ADC12 interrupt sources

Name	Interrupt sources	CPU0 GIC request	DMAC activation
ADCn_ADI	A/D scan end interrupt	Possible	Possible*1
ADCn_GBADI	A/D scan end interrupt for Group B	Possible	Possible
ADCn_GCADI	A/D scan end interrupt for Group C	Possible	Possible
ADCn_CMPAI	Window A compare match	Possible	Not possible
ADCn_CMPBI	Window B compare match	Possible	Not possible
ADCn_WCMPM	Compare match	Not possible	Possible
ADCn_WCMPUM	Compare mismatch	Not possible	Possible

Note: n = 0 to 1.
Note 1. Trigger of event is slightly different from normal interrupt for CPU SPI.

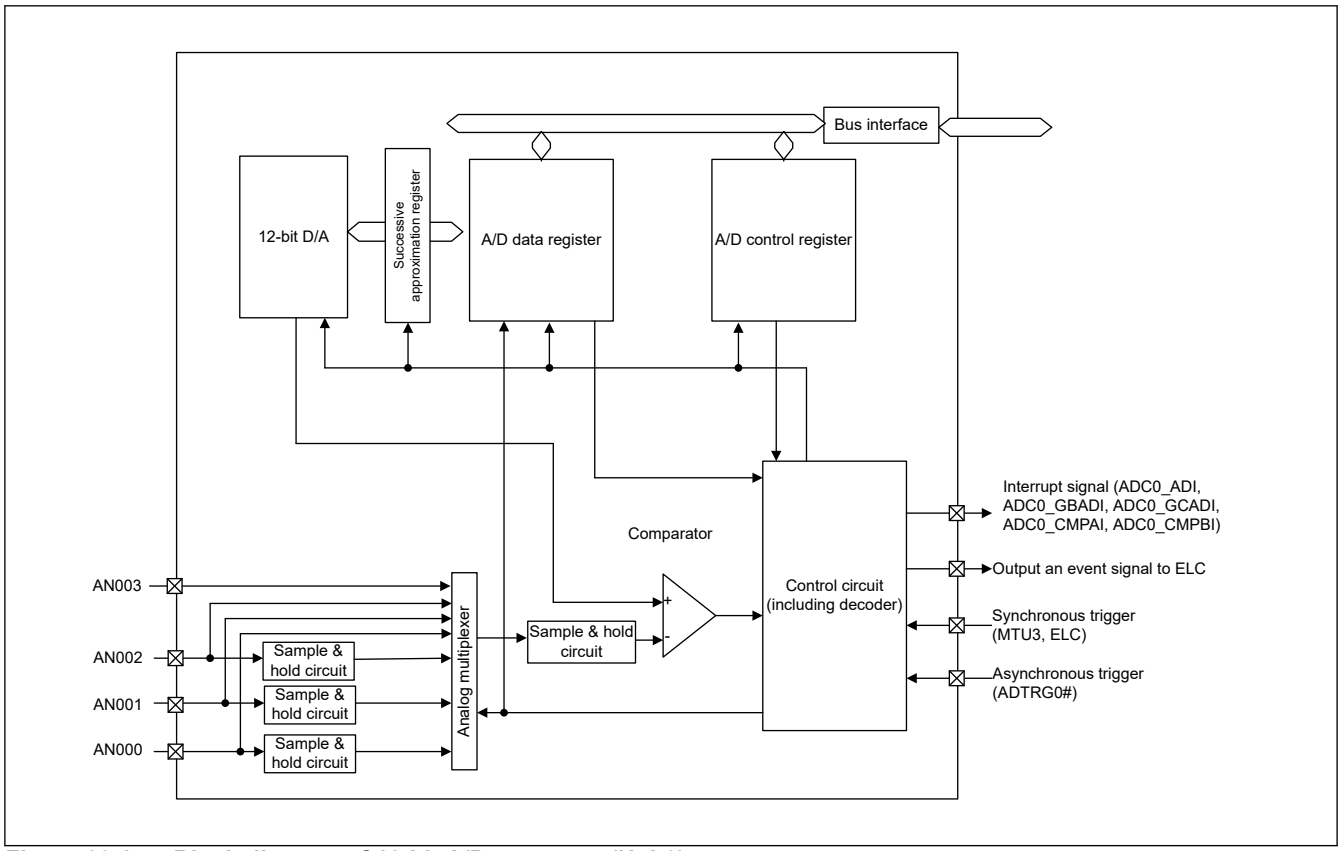


Figure 39.1 Block diagram of 12-bit A/D converter (Unit0)

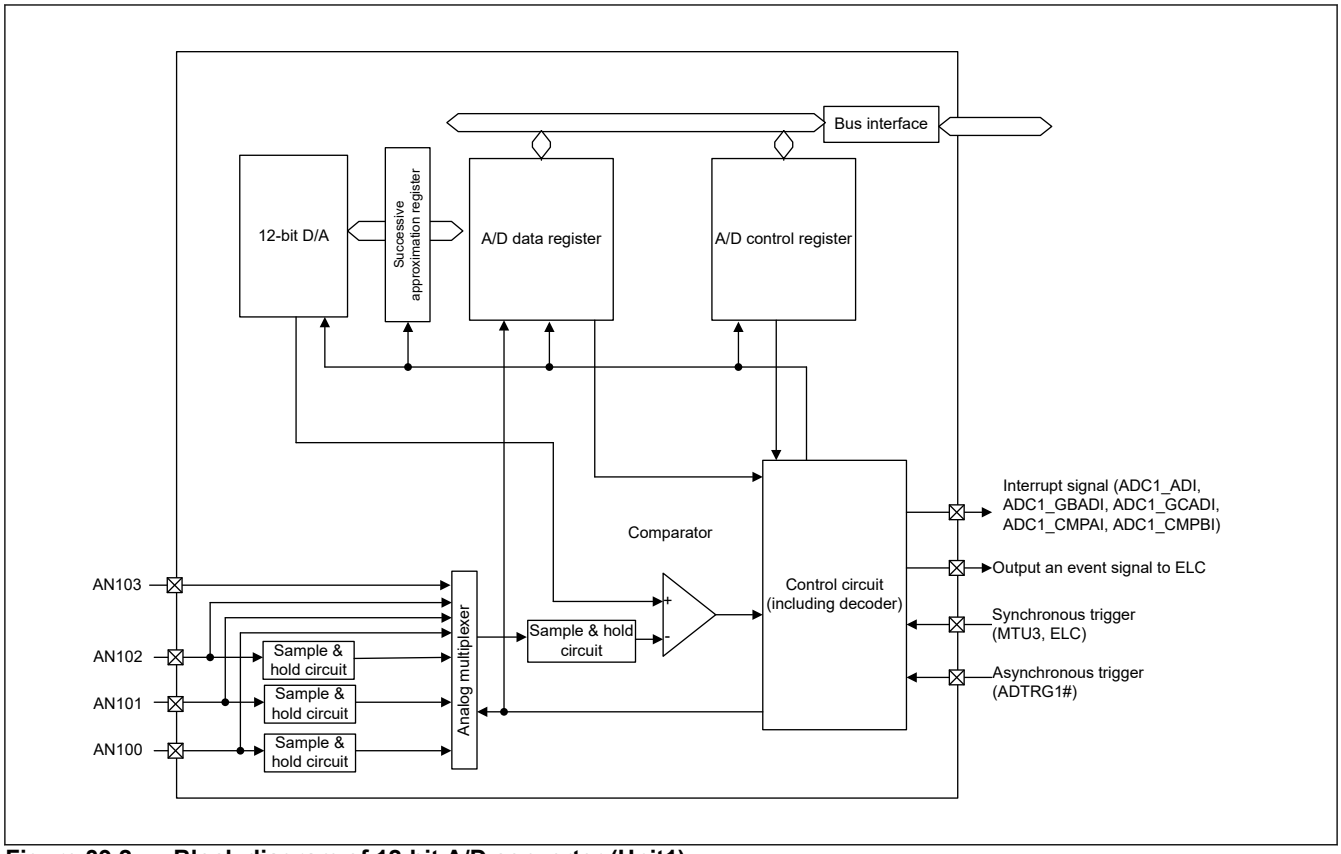


Figure 39.2 Block diagram of 12-bit A/D converter (Unit1)

39.2 Register Descriptions

39.2.1 ADCSR : A/D Control Register

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0x00

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADST	ADCS[1:0]		ADIE	—	—	TRGE	EXTRG	DBLE	GBADIE	—	DBLANS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	DBLANS[4:0]	Double Trigger Channel Select These bits select one analog input channel for double triggered operation. The setting is only valid while double trigger mode is selected.	R/W
5	—	This bit is read as 0. The write value should be 0.	R/W
6	GBADIE	Group B Scan End Interrupt Enable 0: Disables ADCn_GBADI interrupt generation upon group B scan completion 1: Enables ADCn_GBADI interrupt generation upon group B scan completion	R/W
7	DBLE	Double Trigger Mode Select 0: Deselects double trigger mode 1: Selects double trigger mode	R/W
8	EXTRG	Trigger Select 0: A/D conversion is started by a synchronous trigger (MTU3, ELC) 1: A/D conversion is started by the asynchronous trigger (ADTRGn#)	R/W

Bit	Symbol	Function	R/W
9	TRGE	Trigger Start Enable 0: Disables A/D conversion to be started by the synchronous or asynchronous trigger 1: Enables A/D conversion to be started by the synchronous or asynchronous trigger	R/W
11:10	—	These bits are read as 0. The write value should be 0.	R/W
12	ADIE	Scan End Interrupt Enable 0: Disables ADCn_ADI interrupt generation upon scan completion 1: Enables ADCn_ADI interrupt generation upon scan completion	R/W
14:13	ADCS[1:0]	Scan Mode Select 0 0: Single scan mode 0 1: Group scan mode 1 0: Continuous scan mode 1 1: Setting prohibited	R/W
15	ADST	A/D conversion Start 0: Stops A/D conversion process 1: Starts A/D conversion process	R/W

39.2.2 ADANSA0 : A/D Channel Select Register A0

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0x04

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	ANSA0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	ANSA0[3:0]	A/D conversion Analog input Channel Select ANSA0[0] bit corresponds to ch0 and ANSA0[3] corresponds ch3. 0: ch0-ch3 are not subjected to conversion. 1: ch0-ch3 are subjected to conversion.	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

39.2.3 ADADS0 : A/D-Converted Value Addition/Average Function Channel Select Register 0

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0x08

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	ADS0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	ADS0[3:0]	A/D-Converted Value Addition/Average Channel Select 0: A/D-converted value addition/average mode for ch0 to ch3 is not selected 1: A/D-converted value addition/average mode for ch0 to ch3 is selected	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

39.2.4 ADADC : A/D-Converted Value Addition/Average Count Select Register

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0x0C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	AVEE	—	—	—	—	ADC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	ADC[2:0]	Addition Count Select 0 0 0: 1-time conversion (no addition; same as normal conversion) 0 0 1: 2-time conversion (addition once) 0 1 0: 3-time conversion (addition twice)*1 0 1 1: 4-time conversion (addition three times) 1 0 1: 16-time conversion (addition 15 times)*1 Others: Setting is prohibited	R/W
6:3	—	These bits are read as 0. The write value should be 0.	R/W
7	AVEE	Average Mode Enable 0: Addition mode is selected 1: Average mode is selected	R/W

Note 1. When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the Addition count to 3 times and 16times (ADADC.ADC[1:0] = 010b, 101b).

39.2.5 ADCER : A/D Control Extended Register

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0x0E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ADRFMT	—	—	—	—	—	—	—	—	—	ACE	—	—	ADPRC[1:0]		—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
2:1	ADPRC[1:0]	A/D Conversion Accuracy Specify 0 0: A/D conversion is performed with 12-bit accuracy Others: Setting prohibited	R/W
4:3	—	These bits are read as 0. The write value should be 0.	R/W
5	ACE	A/D Data Register Automatic Clearing Enable 0: Disables automatic clearing 1: Enables automatic clearing	R/W
14:6	—	These bits are read as 0. The write value should be 0.	R/W
15	ADRFMT	A/D Data Register Format Select 0: Right-alignment is selected for the A/D data register format 1: Left-alignment is selected for the A/D data register format	R/W

39.2.6 ADSTRGR : A/D Conversion Start Trigger Select Register

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0x10

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	TRSA[5:0]						—	—	TRSB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	TRSB[5:0]	A/D Conversion Start Trigger Select for Group B Select the A/D conversion start trigger for group B in group scan mode. See Table 39.4 for details.	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
13:8	TRSA[5:0]	A/D Conversion Start Trigger Select Select the A/D conversion start trigger in single scan mode and continuous mode. In group scan mode, the A/D conversion start trigger for group A is selected. See Table 39.5 for details.	R/W
15:14	—	These bits are read as 0. The write value should be 0.	R/W

Table 39.4 Selection of A/D activation sources by the TRSB[5:0] bits (for group B only) (1 of 2)

Module	Source	Remarks	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
Trigger source de-selection state			1	1	1	1	1	1
MTU3	TRGA0N	Compare match with or input capture to MTU0.TGRA	0	0	0	0	0	1
MTU3	TRGA1N	Compare match with or input capture to MTU1.TGRA	0	0	0	0	1	0
MTU3	TRGA2N	Compare match with or input capture to MTU2.TGRA	0	0	0	0	1	1
MTU3	TRGA3N	Compare match with or input capture to MTU3.TGRA	0	0	0	1	0	0
MTU3	TRGA4N	Compare match with or input capture to MTU4.TGRA, or an underflow of MTU4.TCNT (in the trough) in complementary PWM mode	0	0	0	1	0	1
MTU3	TRGA6N	Compare match with MTU6.TGRA	0	0	0	1	1	0
MTU3	TRGA7N	Compare match with MTU7.TGRA	0	0	0	1	1	1
MTU3	TRG0N	Compare match with MTU0.TGRE	0	0	1	0	0	0
MTU3	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT	0	0	1	0	0	1
MTU3	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	0
MTU3	TRG4AN or TRG4BN	Compare match between MTU4.TADCORA and MTU4.TCNT, or between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	1
MTU3	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT, and between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is in use)	0	0	1	1	0	0
MTU3	TRG7AN	Compare match between MTU7.TADCORA and MTU7.TCNT	0	0	1	1	0	1
MTU3	TRG7BN	Compare match between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	0

Table 39.4 Selection of A/D activation sources by the TRSB[5:0] bits (for group B only) (2 of 2)

Module	Source	Remarks	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
MTU3	TRG7AN or TRG7BN	Compare match between MTU7.TADCORA and MTU7.TCNT, or between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	1
MTU3	TRG7ABN	Compare match between MTU7.TADCORA and MTU7.TCNT, and between MTU7.TADCORB and MTU7.TCNT (when interrupt skipping function 2 is in use)	0	1	0	0	0	0
ELC	ADCn A	—	0	1	0	0	0	1
ELC	ADCn B	—	0	1	0	0	1	0

Table 39.5 Selection of A/D activation sources by the TRSA[5:0] bits (1 of 2)

Module	Source	Remarks	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
Trigger source de-selection state			1	1	1	1	1	1
External	ADTRGn#	Input pin for the trigger	0	0	0	0	0	0
MTU3	TRGA0N	Compare match with or input capture to MTU0.TGRA	0	0	0	0	0	1
MTU3	TRGA1N	Compare match with or input capture to MTU1.TGRA	0	0	0	0	1	0
MTU3	TRGA2N	Compare match with or input capture to MTU2.TGRA	0	0	0	0	1	1
MTU3	TRGA3N	Compare match with or input capture to MTU3.TGRA	0	0	0	1	0	0
MTU3	TRGA4N	Compare match with or input capture to MTU4.TGRA, or in complementary PWM mode, an underflow of MTU4.TCNT (in the trough)	0	0	0	1	0	1
MTU3	TRGA6N	Compare match with MTU6.TGRA	0	0	0	1	1	0
MTU3	TRGA7N	Compare match with MTU7.TGRA	0	0	0	1	1	1
MTU3	TRG0N	Compare match with MTU0.TGRE	0	0	1	0	0	0
MTU3	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT	0	0	1	0	0	1
MTU3	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	0
MTU3	TRG4AN or TRG4BN	Compare match between MTU4.TADCORA and MTU4.TCNT, or between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	1
MTU3	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT, and between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is in use)	0	0	1	1	0	0
MTU3	TRG7AN	Compare match between MTU7.TADCORA and MTU7.TCNT	0	0	1	1	0	1
MTU3	TRG7BN	Compare match between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	0
MTU3	TRG7AN or TRG7BN	Compare match between MTU7.TADCORA and MTU7.TCNT, or between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	1
MTU3	TRG7ABN	Compare match between MTU7.TADCORA and MTU7.TCNT, and between MTU7.TADCORB and MTU7.TCNT (when interrupt skipping function 2 is in use)	0	1	0	0	0	0
ELC	ADCn A	—	0	1	0	0	0	1

Table 39.5 Selection of A/D activation sources by the TRSA[5:0] bits (2 of 2)

Module	Source	Remarks	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
ELC	ADCn B	—	0	1	0	0	1	0

39.2.7 ADANSB0 : A/D Channel Select Register B0

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0x14

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	ANSB0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	ANSB0[3:0]	A/D Conversion Analog Input Channel Select ANSB0[0] bit corresponds to ch0 and ANSB0[3] corresponds ch3. 0: ch0-ch3 are not subjected to conversion 1: ch0-ch3 are subjected to conversion	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

39.2.8 ADDBLDR : A/D Data Duplication Register

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0x18

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DBLDR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DBLDR[15:0]	The result of A/D conversion in response to the second trigger in double trigger mode.	R

A/D-converted value is 12-bit accuracy. The formats for data in the ADDRy, ADDBLDR, ADDBLDRA, and ADDBLDRB registers vary according to the following conditions.

- The setting of the A/D data register format select bit (ADCER.ADRFMT) (determining whether the data are flush-left or flush-right in the registers)
- The setting of the Addition count select bits (ADADC.ADC[2:0]) (once, twice, three times, four times, or sixteen times is selectable.)
- The setting of the average mode enable bit (ADADC.AVEE) (Addition or average is selectable.)

When the number of the conversion count is set one to four in the addition mode, the A/D conversion addition result is expanded by 2 bits and stored in the A/D data register. When the number of the conversion count is sixteen in the addition mode, the A/D conversion addition result is expanded by 4 bits and stored in the A/D data register.

39.2.9 ADDRn : A/D Data Register n (n = 0 to 3)

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0x020 + 0x002 × n

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DR[15:0]	The result of A/D conversion (n: Number of channel)	R

A/D-converted value is 12-bit accuracy. The formats for data in the ADDRy, ADDBLDR, ADDBLDRA, and ADDBLDRB registers vary according to the following conditions.

- The setting of the A/D data register format select bit (ADCER.ADRFMT) (determining whether the data are flush-left or flush-right in the registers)
- The setting of the Addition count select bits (ADADC.ADC[2:0]) (once, twice, three times, four times, or sixteen times is selectable.)
- The setting of the average mode enable bit (ADADC.AVEE) (Addition or average is selectable.)

When the number of the conversion count is set one to four in the addition mode, the A/D conversion addition result is expanded by 2 bits and stored in the A/D data register. When the number of the conversion count is sixteen in the addition mode, the A/D conversion addition result is expanded by 4 bits and stored in the A/D data register.

39.2.10 ADSHCR : A/D Sample and Hold Control Register

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0x66

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SHANS[2:0]			SSTSH[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

Bit	Symbol	Function	R/W
7:0	SSTSH[7:0]	Sample and hold period setting These bits set the sample and hold period in the range from 4 to 255 states. S/H period is set 13 + this register value.	R/W
10:8	SHANS[2:0]	Sample and hold use or bypass select for ch0-2 SHANS[2] corresponds to ch2, SHANS[0] corresponds to ch0. 0: Bypass S/H 1: Use S/H	R/W
15:11	—	These bits are read as 0. The write value should be 0.	R/W

39.2.11 ADELCCR : A/D Event Link Control Register

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0x7D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	GCELC	ELCC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	ELCC[1:0]	Event link control bits Control the end of scan event (ADCn_ADI). To enable the ELCC[1:0] setting, set GCELC to 0. 0 0: Output event signal at the end of scan except the end of group scan of group A, B 0 1: Output event signal at the end of group scan of group B 1 x: Output event signal at the end of scan	R/W
2	GCELC	Event control bit for Group C 0: Enable the setting of ELCC[1:0] 1: Disable the setting of ELCC[1:0] and output event signal at the end of group scan of group C	R/W
7:3	—	These bits are read as 0. The write value should be 0.	R/W

39.2.12 ADGSPCR : A/D Group Scan Priority Control Register

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0x80

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	GBRP	LGRRS	—	—	—	—	—	—	—	—	—	—	—	—	GBRSCN	PGS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PGS	Group Priority Control Setting*1 0: Operation is without group priority control 1: Operation is with group priority control	R/W
1	GBRSCN	Group B Restart Setting (Enabled only when PGS = 1. Set 0 when PGS = 0.) 0: Scanning for group is not restarted when priority control is performed 1: Scanning for group is restarted when priority control is performed	R/W
13:2	—	These bits are read as 0. The write value should be 0.	R/W
14	LGRRS	Restart Channel Select (Enabled only when PGS = 1 and GBRSCN = 1, Set 0 when PGS = 0.) 0: Restart scan with first channel 1: Restart scan with suspended channel	R/W
15	GBRP	Group B Single Scan Continuous Start*2 (Enabled only when PGS = 1. Set 0 when PGS = 0.) 0: Single scan for group B is not continuously activated 1: Single scan for group B is continuously activated	R/W

Note 1. When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode). If the bits are set to any other values, proper operation cannot be guaranteed.
Note 2. When the GBRP bit has been set to 1, single scan is performed continuously for low priority group regardless of the setting of the GBRSCN bit.

39.2.13 ADDBLDRA : A/D Data Duplication Register A

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0x84

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DBLDRA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DBLDRA[15:0]	The result of A/D conversion during extended operation in double trigger mode	R

A/D-converted value is 12-bit accuracy. The formats for data in the ADDRy, ADDBLDR, ADDBLDRA, and ADDBLDRB registers vary according to the following conditions.

- The setting of the A/D data register format select bit (ADCER.ADRFMT) (determining whether the data are flush-left or flush-right in the registers)
- The setting of the Addition count select bits (ADADC.ADC[2:0]) (once, twice, three times, four times, or sixteen times is selectable.)
- The setting of the average mode enable bit (ADADC.AVEE) (Addition or average is selectable.)

When the number of the conversion count is set one to four in the addition mode, the A/D conversion addition result is expanded by 2 bits and stored in the A/D data register. When the number of the conversion count is sixteen in the addition mode, the A/D conversion addition result is expanded by 4 bits and stored in the A/D data register.

39.2.14 ADDBLDRB : A/D Data Duplication Register B

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0x86

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	DBLDRB[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DBLDRB[15:0]	The result of A/D conversion during extended operation in double trigger mode	R

A/D-converted value is 12-bit accuracy. The formats for data in the ADDRy, ADDBLDR, ADDBLDRA, and ADDBLDRB registers vary according to the following conditions.

- The setting of the A/D data register format select bit (ADCER.ADRFMT) (determining whether the data are flush-left or flush-right in the registers)
- The setting of the Addition count select bits (ADADC.ADC[2:0]) (once, twice, three times, four times, or sixteen times is selectable.)
- The setting of the average mode enable bit (ADADC.AVEE) (Addition or average is selectable.)

When the number of the conversion count is set one to four in the addition mode, the A/D conversion addition result is expanded by 2 bits and stored in the A/D data register. When the number of the conversion count is sixteen in the addition mode, the A/D conversion addition result is expanded by 4 bits and stored in the A/D data register.

39.2.15 ADWINMON : A/D Compare Function Window A/B Status Monitoring Register

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0x8C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	MONCMPB	MONCMPA	—	—	—	MONCOMB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MONCOMB	Combination result monitor 0: Window A/B combination condition does not match. 1: Window A/B combination condition match.	R
3:1	—	These bits are read as 0.	R
4	MONCMPA	Comparing result monitor for window A 0: Window A comparing condition does not match. 1: Window A comparing condition match.	R
5	MONCMPB	Comparing result monitor for window B 0: Window B comparing condition does not match. 1: Window B comparing condition match.	R
7:6	—	These bits are read as 0.	R

39.2.16 ADCMPCR : A/D Compare Function Control Register

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0x90

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPAIE	WCMPE	CMPBIE	—	CMPAE	—	CMPBE	—	—	—	—	—	—	—	CMPAB[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	CMPAB[1:0]	Window A/B combination condition setting This bit is valid when both windows A/B are valid (CMPAE = 1 and CMPBE = 1). 0 0: Output ADCn_WCMPM when window A comparison condition match OR window B comparison condition match, otherwise output ADCn_WCMPUM 0 1: Output ADCn_WCMPM when window A comparison condition match EXOR window B comparison condition match, otherwise output ADCn_WCMPUM 1 0: Output ADCn_WCMPM when window A comparison condition match AND window B comparison condition match, otherwise output ADCn_WCMPUM 1 1: Setting prohibited.	R/W
8:2	—	These bits are read as 0. The write value should be 0.	R/W
9	CMPBE	Window B operation permission 0: Window B is not in operation. Output of ADCn_WCMPM/ADCn_WCMPUM is not permitted. 1: Window B is in operation.	R/W
10	—	This bit is read as 0. The write value should be 0.	R/W
11	CMPAE	Window A operation permission 0: Window A is not in operation. Output of ADCn_WCMPM/ADCn_WCMPUM is not permitted. 1: Window A is in operation.	R/W
12	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
13	CMPBIE	Compare window B Interrupt Enable 0: Generation of an ADCn_CMPBI interrupt in response to matches with a condition for comparison is disabled. 1: Generation of an ADCn_CMPBI interrupt in response to matches with a condition for comparison is enabled.	R/W
14	WCMPE	Window Function enable 0: Window function disabled 1: Window function enabled	R/W
15	CMPAIE	Compare window A Interrupt Enable 0: Generation of an ADCn_CMPAI interrupt in response to matches with a condition for comparison is disabled. 1: Generation of an ADCn_CMPAI interrupt in response to matches with a condition for comparison is enabled.	R/W

39.2.17 ADCMPANSR0 : A/D Compare Function Window A Channel Select Register 0

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0x94

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPCHA0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPCHA0[15:0]	Window A Channel Select Set the CMPCHA0[15:0] bits when DCSR.ADST = 0. 0: Channel is not target 1: Channel is target	R/W

39.2.18 ADCMPLR0 : A/D Compare Function Window A Comparison Condition Setting Register 0

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0x98

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	CMPLCHA0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CMPLCHA0[3:0]	Window A comparison condition for target channel (ch0-3) setting CMPLCHA0[3] corresponds channel 3 and CMPLCHA0[0] corresponds channel 0. When the comparison result of each channel input matches the set condition, the ADCMPSR0.CMPSTCHA0n bit is set to 1 and a compare interrupt (ADCn_CMPAI) is generated. 0: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 register value > A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): A/D-converted value < ADCMPDR0 register value or ADCMPDR1 register value < A/D-converted value 1: When window function is disabled (ADCMPCR.WCMPE = 0): ADCMPDR0 register value < A/D-converted value When window function is enabled (ADCMPCR.WCMPE = 1): ADCMPDR0 register value < A/D-converted value < ADCMPDR1 register value	R/W

Bit	Symbol	Function	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

39.2.19 ADCMPDR0 : A/D Comparison Function Window A Lower Level Setting Register

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0x9C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPLLA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPLLA[15:0]	Reference data setting when using the compare function window A	R/W

39.2.20 ADCMPDR1 : AD Comparison Function Window A Upper Level Setting Register

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0x9E

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPULA[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPULA[15:0]	Reference data setting when using the compare function window A	R/W

39.2.21 ADCMPSR0 : A/D Comparison Function Window A Channel Status Register 0

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0xA0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPSTCHA0[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPSTCHA0[15:0]	Window A Status Flag This flag shows the comparison result of each targeted channel, when window A is operating (ADCMPCR.CMPAE = 1b). If this flag is set while ADCMPCR.CMPAIE = 1, a compare interrupt (ADCn_CMPAI) request is generated. 0: Condition does not match 1: Condition matches	R/W

39.2.22 ADWINLLB : A/D Compare Function Window B Lower-Side Level Setting Register

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0xA8

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPLLB[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPLLB[15:0]	Reference lower data setting when using the compare function window B	R/W

39.2.23 ADWINULB : A/D Compare Function Window B Upper-Side Level Setting Register

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0xAA

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CMPULB[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	CMPULB[15:0]	Reference upper data setting when using the compare function window B	R/W

39.2.24 ADCMPBSR : A/D Compare Function Window B Status Register

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0xAC

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	CMPSTB
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMPSTB	Window B Flag When window B operation is enabled (ADCMPCR.CMPBE = 1b), this bit indicates the comparison result of channels to which window B comparison conditions are applied. 0: Comparison conditions are not met 1: Comparison conditions are met	R/W
7:1	—	These bits are read as 0. The write value should be 0.	R/W

39.2.25 ADCMPBNSR : A/D Compare Function Window B Channel Select Register

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0xA6

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CMPLB	—	CMPCHB[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	CMPCHB[5:0]	Window B Channel Select Select channels for comparing with window B condition. Set the CMPCHB[5:0] bits when the ADCSR.ADST bit is 0. 0x00: ch0 0x01: ch1 0x02: ch2 0x03: ch3 0x3F: No channel is selected Others: Setting prohibited	R/W
6	—	This bit is read as 0. The write value should be 0.	R/W
7	CMPLB	Window B Comparison Condition Setting When the comparison result of target input matches the set condition, the ADCMPBSR.CMPSTB bit is set to 1 and a compare interrupt (ADCn_CMPBI) is generated. 0: When window function is disabled (ADCMPCR.WCMPE = 0): ADWINLLB register value > A/D-converted value. When window function is enabled (ADCMPCR.WCMPE = 1): A/D-converted value < ADWINLLB register value or ADWINULB register value < A/D-converted value. 1: When window function is disabled (ADCMPCR.WCMPE = 0): ADWINLLB register value < A/D-converted value. When window function is enabled (ADCMPCR.WCMPE = 1): ADWINLLB register value < A/D-converted value < ADWINULB register value.	R/W

39.2.26 ADANSC0 : A/D Channel Select Register C0

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0xD4

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	ANSC0[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	ANSC0[3:0]	A/D-Converted Channel Select for Group C in Group Scan Mode ANSC0[3] corresponds ch3, and ANSC0[0] corresponds ch0. 0: Channel is not target 1: Channel is target	R/W
15:4	—	These bits are read as 0. The write value should be 0.	R/W

39.2.27 ADGCTRGR : A/D Group C Trigger Select Register

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0xD9

Bit position:	7	6	5	4	3	2	1	0
Bit field:	GRCE	GCADIE	TRSC[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	TRSC[5:0]	Group C A/D Conversion Start Trigger Select Select starting trigger for A/D conversion of group C in group scan mode. See Table 39.6 for more details.	R/W
6	GCADIE	Group C Scan Completion Interrupt Enable 0: Group C scan completed interrupt (ADCn_GCADI) is disabled 1: Group C scan completed interrupt (ADCn_GCADI) is enabled	R/W
7	GRCE	Group C A/D Conversion Enable 0: Group C A/D conversion is disabled 1: Group C A/D conversion is enabled	R/W

Table 39.6 A/D convert start trigger for Group C in group scan mode

Module	Trigger	Description	TRSC					
			[5]	[4]	[3]	[2]	[1]	[0]
No Trigger is selected			1	1	1	1	1	1
MTU3	TRGA0N	MTU0.TGRA Compare match/Input capture	0	0	0	0	0	1
MTU3	TRGA1N	MTU1.TGRA Compare match/Input capture	0	0	0	0	1	0
MTU3	TRGA2N	MTU2.TGRA Compare match/Input capture	0	0	0	0	1	1
MTU3	TRGA3N	MTU3.TGRA Compare match/Input capture	0	0	0	1	0	0
MTU3	TRGA4N	MTU4.TGRA Compare match/Input capture, or underflow (trough) of MTU4.TCNT in PWM mode	0	0	0	1	0	1
MTU3	TRGA6N	MTU6.TGRA Compare match/Input capture	0	0	0	1	1	0
MTU3	TRGA7N	MTU7.TGRA Compare match/Input capture, or underflow (trough) of MTU7.TCNT in PWM mode	0	0	0	1	1	1
MTU3	TRG0N	MTU0.TGRE Compare match	0	0	1	0	0	0
MTU3	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT	0	0	1	0	0	1
MTU3	TRG4BN	Compare match between MTU4.TADCORB and NTU4.TCNT	0	0	1	0	1	0
MTU3	TRG4AN or TRG4BN	Compare match between MTU4.TADCORA and MTU4.TCNT, or Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	1
MTU3	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT, and Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	1	0	0
MTU3	TRG7AN	Compare match between MTU7.TADCORA and MTU7.TCNT	0	0	1	1	0	1
MTU3	TRG7BN	Compare match between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	0
MTU3	TRG7AN or TRG7BN	Compare match between MTU7.TADCORA and MTU7.TCNT, or Compare match between MTU7.TADCORB and MTU7.TCNT	0	0	1	1	1	1
MTU3	TRG7ABN	Compare match between MTU7.TADCORA and MTU7.TCNT, and Compare match between MTU7.TADCORB and MTU7.TCNT	0	1	0	0	0	0
ELC	ADCn A		0	1	0	0	0	1
ELC	ADCn B		0	1	0	0	1	0

39.2.28 ADSSTRn : A/D Sampling State Register n (n = 0 to 3)

Base address: ADC120 = 0x9000_4000
ADC121 = 0x9000_4800

Offset address: 0xE0 + 0x01 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SST[7:0]							
Value after reset:	0	0	0	0	1	0	1	1

Bit	Symbol	Function	R/W
7:0	SST[7:0]	Sampling Time Setting These bits set the sampling time in the range from 8 to 255 states.	R/W

39.3 Operation

39.3.1 Scanning Operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.

There are three operating modes: single scan mode, continuous scan mode, and group scan mode. In single scan mode, one or more specified channels are scanned once. In continuous scan mode, one or more specified channels are scanned repeatedly until the ADCSR.ADST bit is cleared to 0 from 1. In group scan mode, the selected channels of group A and the selected channels of group B (and the selected channels of group C) are scanned once after starting to be scanned according to the respective synchronous triggers (MTU3, ELC).

In single scan mode and continuous scan mode, A/D conversion is performed for ANn pins selected by the ADANSA0 register, starting from the pin with the smallest number n. In group scan mode, A/D conversion is performed for ANn pins selected by the ADANSA0 register for group A, and performed for ANn pins selected by the ADANSB0 register for group B and performed for ANn pins selected by the ADANSC0 register for group C, respectively, starting from the pin with the smallest number n.

Double trigger mode is to be used with single scan mode or group scan mode. With double trigger mode being enabled, A/D conversion data of a channel selected by the ADCSR.DBLANS[4:0] bits is duplicated only if the conversion is started by any of the synchronous triggers (MTU3, ELC) selected by the ADSTRGR.TRSA[5:0] bits. In group scan mode, only group A can use the double trigger.

To enable double trigger extended mode, select double trigger mode, and then select the synchronous trigger by the ADSTRGR.TRSA[5:0].

- ADSTRGR.TRSA[5:0] = 001011b
  TRG4AN or TRG4BN
- ADSTRGR.TRSA[5:0] = 001111b
  TRG7AN or TRG7BN

In the double trigger extended mode, in addition to the normal double trigger mode operation, the A/D conversion data activated by the odd number trigger (TRG4AN, TRG7AN) is stored in the A/D data duplex register A (ADDBLDRA), and the A/D conversion data activated by the even number trigger (TRG4BN, TRG7BN) is stored in the A/D Data Duplication register B (ADDBLDRB).

If two types of trigger source occur at the same time, the data is not sorted by the trigger factor and the A/D conversion data is stored in the Data Duplication register B (ADDBLDRB). If one trigger is performing A/D conversion and the other trigger is input, the other trigger will be ignored.

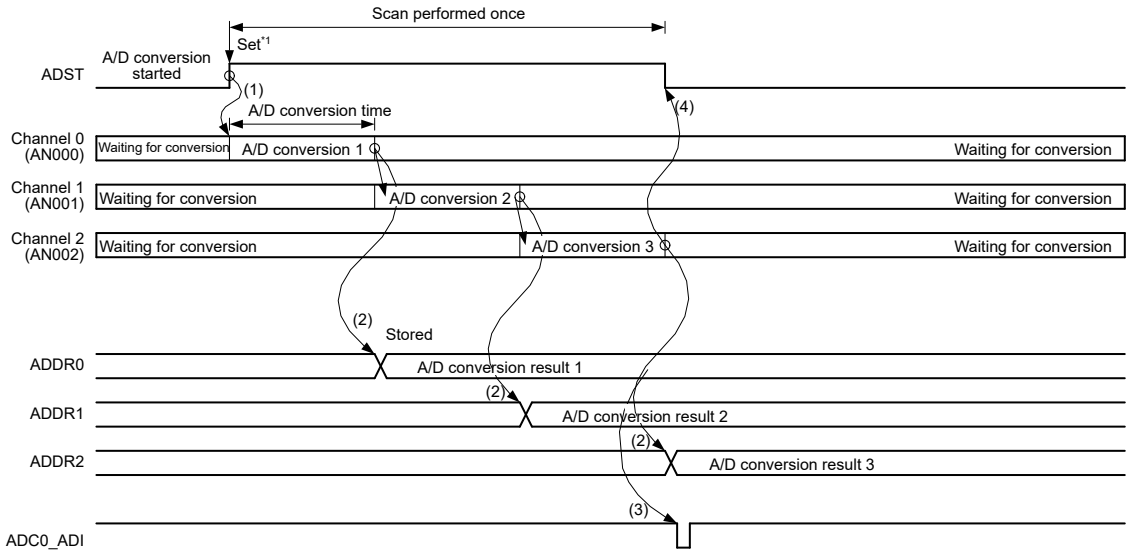
Analog input AN000 to AN002 can use the S/H circuit by setting of ADSHCR.SHANS[2:0].

39.3.2 Single Scan Mode

39.3.2.1 Basic Operation (Not use S/H circuits)

In basic operation of single scan mode, A/D conversion is performed once on the analog input of the specified channels as below.

1. When the ADCSR.ADST bit becomes 1 (A/D conversion start) due to software or synchronous trigger (MTU3, ELC) input, A/D conversion starts in ascending order of n of ANn selected in the ADANSA0 register.
2. When the A/D conversion of one channel is completed, the A/D conversion result is stored in the corresponding A/D data register (ADDRy).
3. If the ADCSR.ADIE bit is set to 1 (ADC0_ADI interrupt enabled at the end of scanning) after A/D conversion of all selected channels is completed, an ADC0_ADI interrupt request (pulse output) is generated.
4. ADCSR.ADST bit holds 1 (A/D conversion start) during A/D conversion and is automatically cleared when A/D conversion of all selected channels is completed, and the 12-bit A/D converter goes into a standby state.



Note 1. The arrows in this figure indicate instruction execution by software.

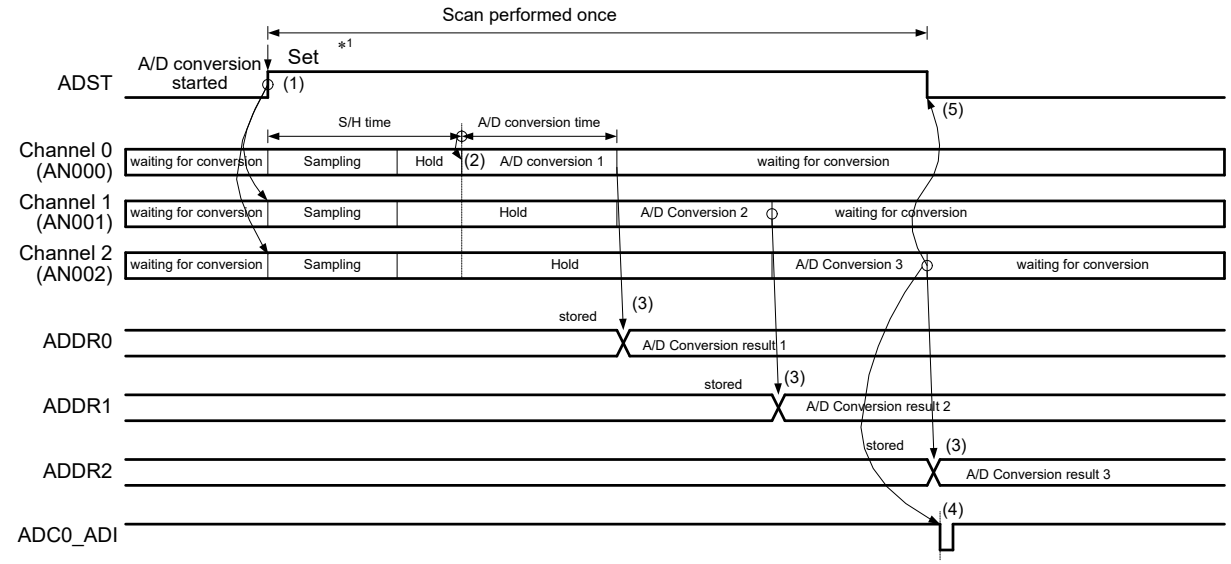
Figure 39.3 Example of operation in single scan mode (basic operation AN004 to AN006 selected)

39.3.2.2 Basic Operation (Use S/H circuits)

The S/H circuit is used to A/D Convert the analog inputs of all selected channels after S/H is performed only once.

The channel that uses the S/H circuit is selected with the ADSHCR.SHANS[2:0] bits.

1. When the ADCSR.ADST bit becomes 1 (A/D conversion start) by software or synchronization trigger (MTU3, ELC), all channels using the S/H circuit start sampling analog inputs.
2. After sampling and holding, A/D conversion is started in ascending order of n of the channel ANn selected in the ADANSA0 register.
3. When the A/D conversion of the channel is completed, the A/D conversion result is stored in the corresponding A/D data register (ADDRy).
4. If the ADCSR.ADIE bit is set to 1 (ADC0_ADI interrupt enabled at the end of scanning) after A/D conversion of all selected channels is completed, an ADC0_ADI interrupt request (pulse output) is generated.
5. The ADCSR.ADST bit holds 1 (A/D conversion start) during A/D conversion and is automatically cleared when A/D conversion of all selected channels is completed, and the 12-bit A/D converter goes into a standby state.



Note 1. The arrows in this figure indicate instruction execution by software.

Figure 39.4 Example of operation in single scan mode (basic operation with S/H, AN000 to AN002 selected)

39.3.2.3 A/D conversion in Double Trigger Mode

When double trigger mode is selected in single scan mode, A/D conversion is performed for two rounds of single scan operation started by a synchronous trigger (MTU3, ELC).

A/D conversion Data Duplication is enabled by setting the channel number to be duplicated to the ADCSR.DBLANS[4:0] bits and setting ADCSR.DBLE to 1. When ADCSR.DBLE is set to 1, the channel selection of the ADANSA0 register is disabled.

To select the double trigger mode, use ADSTRGR.TRSA[5:0] to select the synchronous trigger (MTU3, ELC), set the ADCSR.EXTRG bit to 0, and set the ADCSR.TRGE bit to 1. Also, do not use software trigger.

1. When the ADCSR.ADST bit becomes 1 (A/D conversion start) due to the synchronous trigger (MTU3, ELC) input, the A/D conversion of the 1 channel selected by the ADCSR.DBLANS[4:0] bits starts.
2. When the A/D conversion is completed, the A/D conversion result is stored in the corresponding A/D data register (ADDRy).
3. ADCSR.ADST is automatically cleared and the 12-bit A/D converter is in the standby state. At this time, the ADC0_ADI interrupt does not occur regardless of the setting of the ADCSR.ADIE bit (ADC0_ADI interrupt enabled by the end of scanning).
4. When the ADCSR.ADST bit becomes 1 (A/D conversion start) by the second synchronous trigger input, A/D conversion of the 1 channel selected by the ADCSR.DBLANS[4:0] bits starts.
5. When the A/D conversion is completed, the A/D conversion result is stored in the A/D Data Duplication register (ADDBLDR) dedicated to the double trigger mode.
6. If the ADCSR.ADIE bit is set to 1 (ADC0_ADI interrupt enabled at the end of scanning), an ADC0_ADI interrupt request (pulse output) is generated.
7. The ADCSR.ADST bit holds 1 (A/D conversion start) during A/D conversion, is automatically cleared when A/D conversion is completed, and the 12-bit A/D converter is in the standby state.

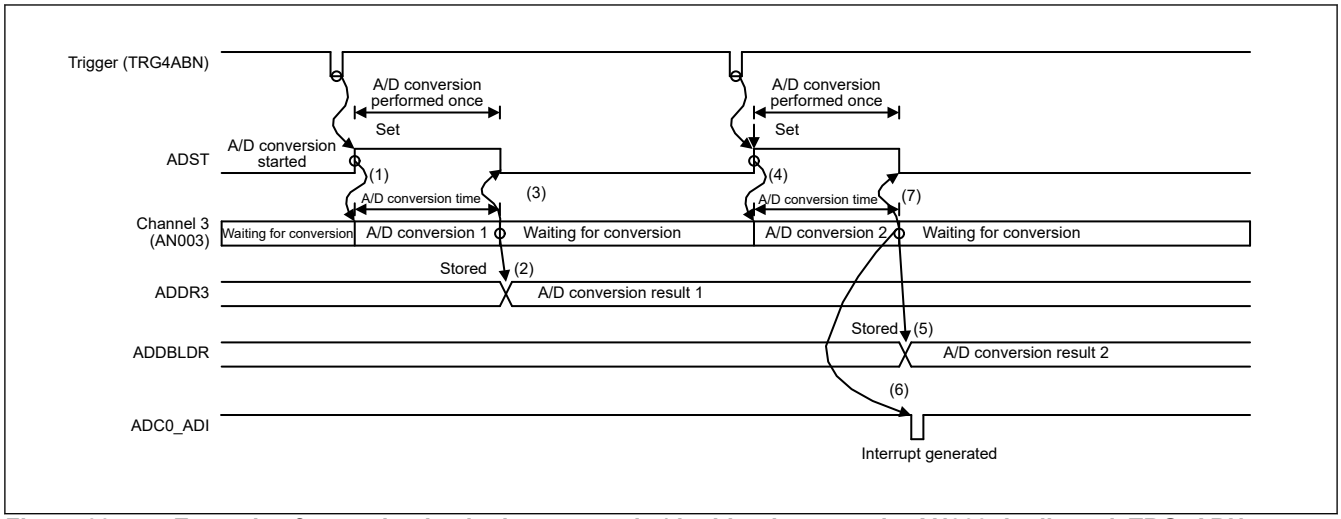


Figure 39.5 Example of operation in single scan mode (double trigger mode, AN003 duplicated, TRG4ABN is selected as the trigger)

39.3.2.4 Extended Operations When Double Trigger Mode is Selected

When double trigger mode is selected in single scan mode, single scan is performed twice.

To enable double trigger extended mode, and then select the following synchronous trigger by the ADSTRGR.TRSA[5:0].

- ADSTRGR.TRSA[5:0] = 001011b
  TRG4AN or TRG4BN
- ADSTRGR.TRSA[5:0] = 001111b
  TRG7AN or TRG7BN

And set the ADCSR.EXTRG bit to 0 and the ADCSR.TRGE bit to 1. Also, do not use software trigger.

A/D conversion Data Duplication is enabled by setting the channel number to be duplicated to the ADCSR.DBLANS[4:0] bits and setting ADCSR.DBLE to 1. When ADCSR.DBLE is set to 1, the channel selection of the ADANSA0 register is disabled.

1. When the ADCSR.ADST bit becomes 1 (A/D conversion start) by TRG4AN, A/D conversion of 1 channel selected by the ADCSR.DBLANS[4:0] bits starts.
2. When the A/D conversion is completed, the A/D conversion result is stored in the corresponding A/D data register (ADDRy) and A/D Data Duplication register A (ADDBLDRA).
3. ADCSR.ADST is automatically cleared and the 12-bit A/D converter is in the standby state. At this time, the ADC0_ADI interrupt does not occur regardless of the setting of the ADCSR.ADIE bit (ADC0_ADI interrupt enabled by the end of scanning).
4. When the ADCSR.ADST bit becomes 1 (A/D conversion start) by TRG4BN input, A/D conversion of 1 channel selected by the ADCSR.DBLANS[4:0] bits starts.
5. When the A/D conversion is completed, the A/D conversion result is stored in the A/D Data Duplication register (ADDBLDR) and the A/D Data Duplication register B (ADDBLDRB).
6. If the ADCSR.ADIE bit is set to 1 (ADC0_ADI interrupt enabled at the end of scanning), an ADC0_ADI interrupt request (pulse output and level output) is generated.
7. The ADCSR.ADST bit holds 1 (A/D conversion start) during A/D conversion, is automatically cleared when A/D conversion is completed, and the 12-bit A/D converter is in the standby state.

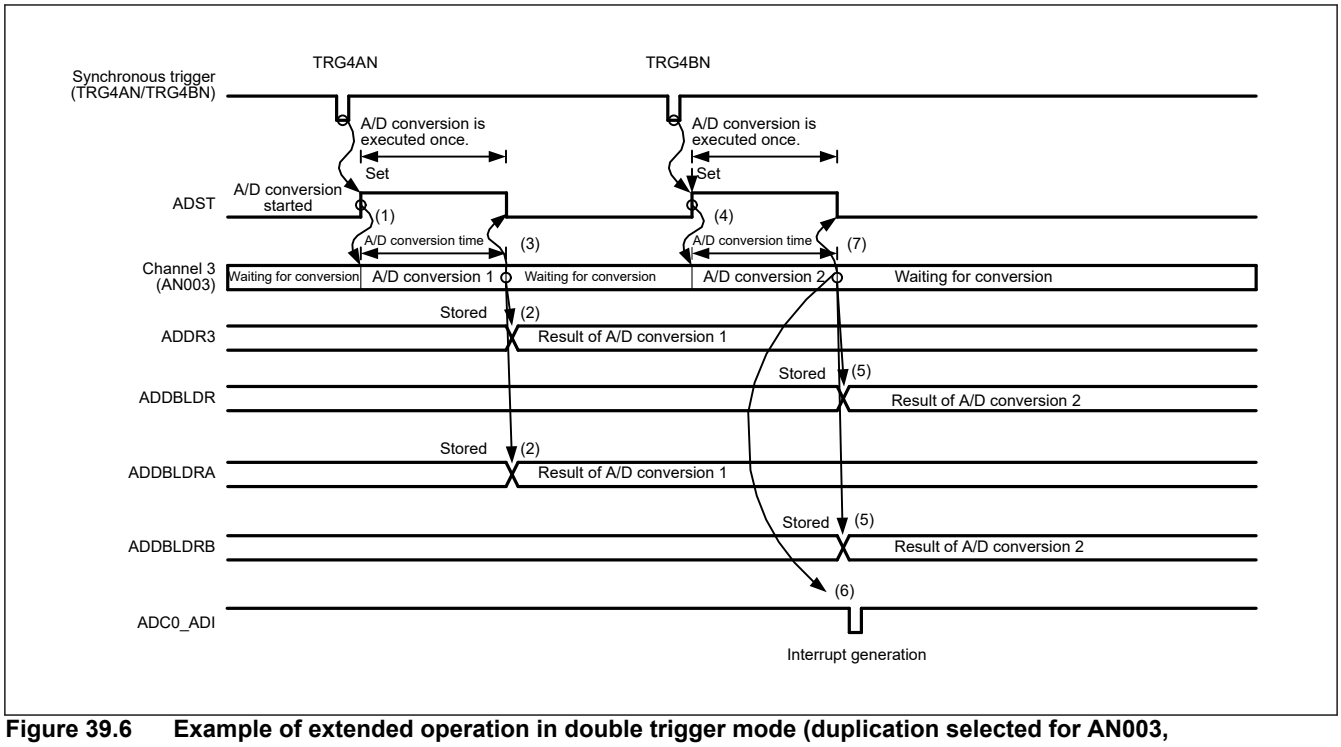


Figure 39.6 Example of extended operation in double trigger mode (duplication selected for AN003, TRG4AN, TRG4BN selected)

39.3.3 Continuous Scan Mode

39.3.3.1 Basic Operation (Not use S/H circuits)

In basic operation of continuous scan mode, A/D conversion is performed repeatedly on the analog input of the channels selected by the ADANSA0 register.

1. When the ADCSR.ADST bit becomes 1 (A/D conversion start) by software, synchronous trigger (MTU3, ELC), or asynchronous trigger, A/D conversion starts in ascending order of n of ANn selected in the ADANSA0 register.
2. When the A/D conversion of one channel is completed, the A/D conversion result is stored in the corresponding A/D data register (ADDRy).
3. If the ADCSR.ADIE bit is set to 1 (ADC0_ADI interrupt enabled at the end of scanning) after A/D conversion of all selected channels is completed, an ADC0_ADI interrupt request (pulse output) is generated. In addition, the 12-bit A/D converter continuously starts A/D conversion in ascending order of n of ANn selected in the ADANSA0 register.
4. The ADCSR.ADST bit is not cleared automatically, and (2) to (3) are repeated during 1 (A/D conversion start). When the ADCSR.ADST bit is set to 0 (A/D conversion stopped), A/D conversion is stopped and the 12-bit A/D converter is in the standby state.
5. After that, when the ADCSR.ADST bit becomes 1 (A/D conversion starts), A/D conversion starts again in ascending order of n of ANn selected in the ADANSA0 register.

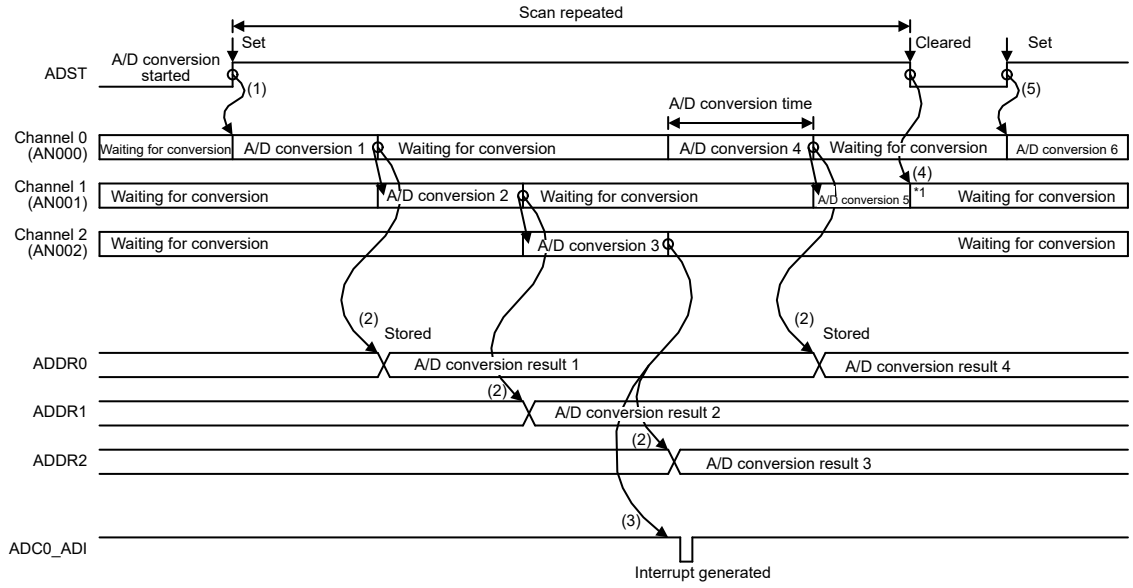


Figure 39.7 Example of operation in continuous scan mode (basic operation AN000 to AN002 selected)

39.3.3.2 Basic Operation (Use S/H circuits)

When the S/H circuit is used, the operation of A/D conversion of the analog inputs of all the selected channels is repeated after the sample and hold is performed. The channel that uses the S/H circuit is selected with the ADSHCR.SHANS[2:0] bits.

1. When the ADCSR.ADST bit becomes 1 (A/D conversion start) by software, synchronization trigger (MTU3, ELC), or asynchronization trigger, all channels using the S/H circuit start sampling analog inputs.
2. After sampling and holding, A/D conversion is started in ascending order of n of the channel ANn selected in the ADANSA0 register.
3. When the A/D conversion of the channel is completed, the A/D conversion result is stored in the corresponding A/D data register (ADDRy).
4. If the ADCSR.ADIE bit is set to 1 (ADC0_ADI interrupt enabled at the end of scanning) after A/D conversion of all selected channels is completed, ADC0_ADI interrupt request (pulse output) is generated. It also begins sampling the analog inputs of all channels that use the S/H circuitry.
5. The ADCSR.ADST bit is not cleared automatically, and (2) to (4) are repeated while it is set to 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stopped), A/D conversion is stopped and the 12-bit A/D converter is in the standby state.
6. After that, when the ADCSR.ADST bit becomes 1 (A/D conversion starts), sampling of analog inputs of all channels using the S/H circuit starts again.

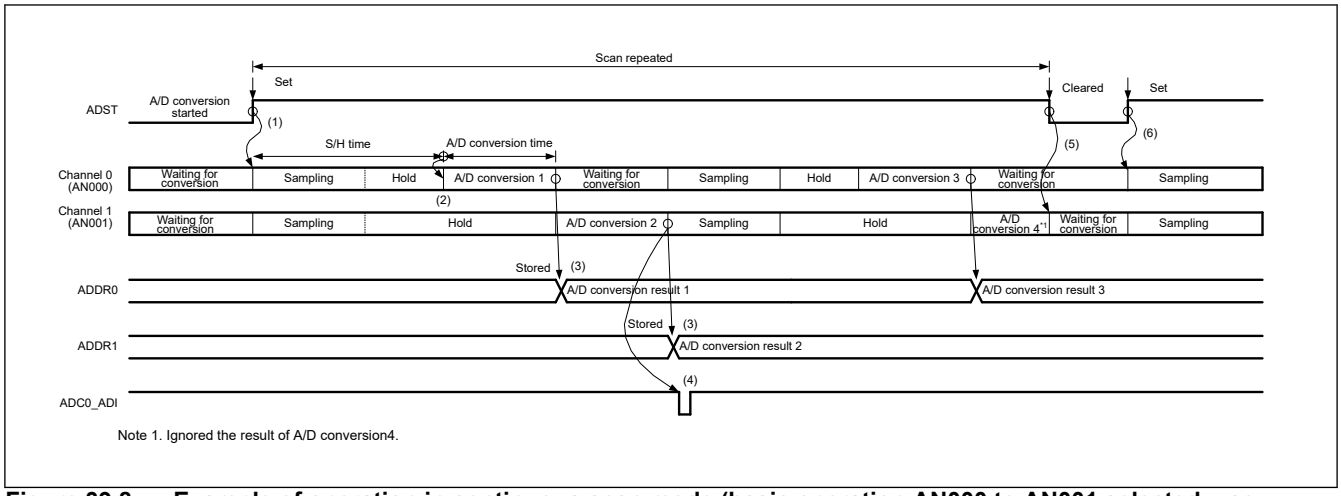


Figure 39.8 Example of operation in continuous scan mode (basic operation AN000 to AN001 selected, use S/H circuit)

39.3.4 Group Scan Mode

39.3.4.1 Basic Operation

In basic operation of group scan mode, A/D conversion is performed once on the analog inputs of all the specified channels in each group (A and B or A, B and C) after scanning is started by a synchronous trigger (MTU3, ELC). Scan operation of each group is similar to the scan operation in single scan mode.

To set the trigger for group scan mode, select the trigger for group A with the ADSTRGR.TRSA[5:0] bits, select the trigger for group B with the ADSTRGR.TRSB[5:0] bits, and select the trigger for group C with the ADGCTRGR.TRSC[5:0] bits. Triggers for groups A, B, and C should be separate triggers so that scans for groups A, B, and C do not occur at the same time. Also, do not use software triggers. For the channels to be scanned, select the group A channel in the ADANSA0 register, select the group B channel in the ADANSB0 register, and select the group C channel in the ADANSC0 register.

The following is an example of the operation of the group scan mode by the synchronous trigger from the MTU3. Group A is set to start conversion with a TRG4AN trigger from MTU3, Group B is set to start conversion with a TRG4BN trigger from MTU3, and Group C is set to start conversion with a TRG4ABN trigger from MTU3.

1. Start scanning of group A with TRG4AN trigger from MTU3.
2. If the ADCSR.ADIE bit is set to 1 (ADC0_ADI interrupt enabled) at the end of group A scanning, the ADC0_ADI interrupt is output.
3. Start scanning Group B with TRG4BN trigger from MTU3.
4. If the ADCSR.GBADIE bit is set to 1 (ADC0_GBADI interrupt enabled) at the end of Group B scanning, an ADC0_GBADI interrupt is output.
5. Trigger TRG4ABN from MTU3 to start scanning for Group C.
6. If the ADGCTRGR.GCADIE bit is set to 1 (ADC0_GCADI interrupt enabled) at the end of Group C scanning, an ADC0_GCADI interrupt is output.

In group scan mode, different channels and triggers should be selected for group A and group B.

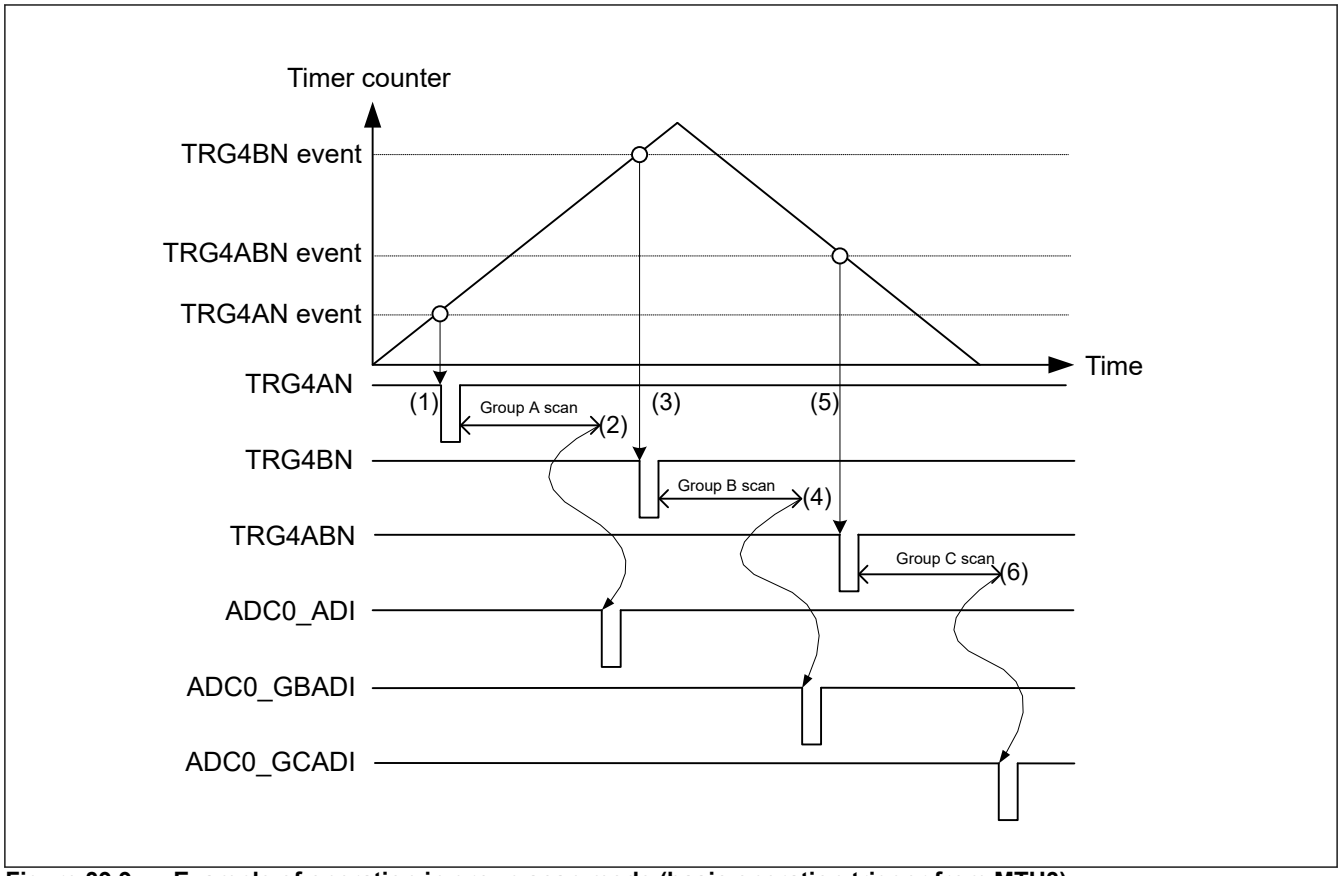


Figure 39.9 Example of operation in group scan mode (basic operation trigger from MTU3)

39.3.4.2 A/D conversion in Double Trigger Mode

When double trigger mode is selected in group scan mode, group A controls two single scan mode executions starting with a synchronous trigger (MTU3, ELC) as a series of operations. Group B and Group C behave in the same way as single scan mode started with a synchronous trigger (MTU3, ELC).For group scan mode trigger settings, the ADSTRGR.TRSA[5:0] bits select the trigger for group A, the ADSTRGR.TRSB[5:0] bits select the trigger for group B, and ADGCTRGR.TRSC[5:0] bits selects the trigger for Group C. Triggers for groups A, B, and C should be separate triggers so that scans for groups A, B, and C do not occur at the same time. Also, do not use software triggers.

If you select "TRG4AN" or "TRG4BN", "TRG7AN" or "TRG7BN" as the trigger for group A in the ADSTRGR.TRSA[5:0] bits, it operates in double trigger extended mode.

For the channels to be scanned, select the group A channel with the ADCSR.DBLANS[4:0] bits, select the group B channel with the ADANSB0 register, and select the group C channel with the ADANSC0 register.

A/D conversion Data Duplication is enabled by setting the channel number to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1.

The following shows an operation example when the group scan mode and double trigger mode are set by the synchronous trigger from the MTU3. Group A is set to start conversion with a TRG4ABN trigger from MTU3, Group B is set to start conversion with a TRGA0N trigger from MTU3, and Group C is set to start conversion with a TRGA1N trigger from MTU3.

1. Start scanning Group C with the TRGA1N trigger from the MTU3.
2. If the ADGCTRGR.GCADIE bit is set to 1 (ADC0_GCADI interrupt enabled) at the end of Group C scanning, the ADC0_GCADI interrupt is output.
3. The TRGA0N trigger from the MTU3 starts scanning for group B.
4. If the ADCSR.GBADIE bit is set to 1 (ADC0_GBADI interrupt enabled) at the end of Group B scanning, an ADC0_GBADI interrupt is output.
5. The first TRG4ABN trigger from the MTU3 will start the first scan of Group A.

6. At the end of the first scan of Group A, the converted data is stored in ADDRy, and the ADC0_ADI interrupt request does not occur regardless of the setting of the ADCSR.ADIE bit.
7. The second TRG4ABN trigger from the MTU3 will start the second scan of Group A.
8. At the end of the second scan of group A, the conversion data is stored in ADDBLDR, and if the ADCSR.ADIE bit is set to 1 (ADC0_ADI interrupt enabled), the ADC0_ADI interrupt is output.
9. The second TRGA0N trigger from the MTU3 starts the second scan of group B.
10. At the end of the second scan of Group B, if the ADCSR.GBADIE bit is set to 1 (ADC0_GBADI interrupt enabled), an ADC0_GBADI interrupt is output.
11. The second TRGA1N trigger from the MTU3 starts the second scan of Group C.
12. At the end of the second scan of Group C, if the ADGCTRGR.GCADIE bit is set to 1 (ADC0_GCADI interrupt enabled), an ADC0_GCADI interrupt is output.

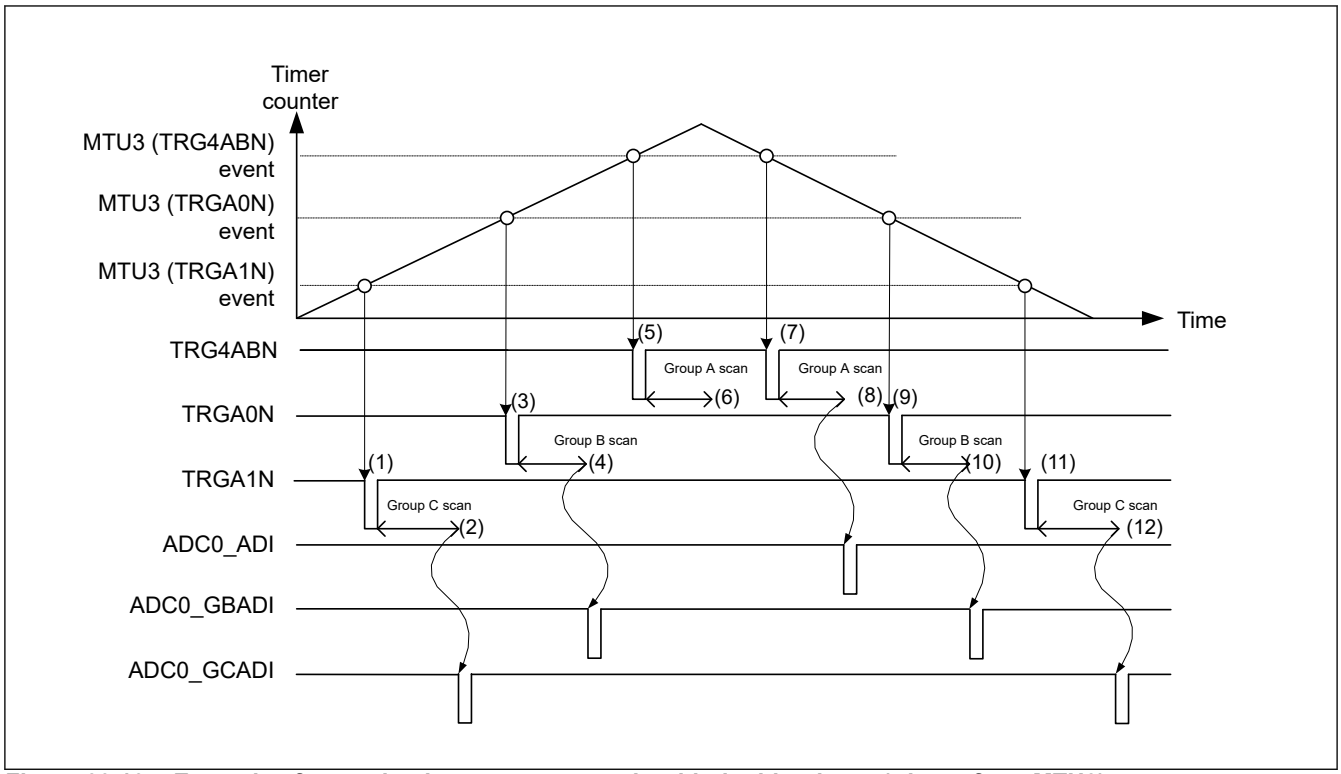


Figure 39.10 Example of operation in group scan mode with double trigger (trigger from MTU3)

39.3.4.3 Operation under Group Priority Control

When the ADGSPCR.PGS bit is set to 1 in group scan mode, group priority operation is performed. Group priority is in the order of group A > group B > group C. The number of groups used in group scan mode can be selected from either 2 (Groups A, B) or 3 (Groups A, B, C) depending on the setting of ADGCTRGR.GRCE. When setting the PGS bit in the ADPGSCR register to 1, follow the procedure in Figure 39.11 to set the relevant registers. If the procedure is not followed, A/D conversion operation and stored data are not guaranteed.

In the basic operation of group scan mode, the trigger input generated during scanning of groups A, B, and C is ignored, and the scanning operation of groups A, B, and C is the same as that of single scan mode.

In the group priority operation, if there is a trigger input of the priority group during the scan of the low priority group, the scan of the low priority group is interrupted and the priority group is scanned.

When the ADGSPCR.GBRSCN bit is 0, the low priority group is in the standby state after the priority group scan is completed. Also, the low priority group trigger input that occurs during the scan is ignored.

When the ADGSPCR.GBRSCN bit is 1, the scan of the low priority group is automatically re-executed after the scan of the priority group is completed. In addition, the trigger input of the low priority group that occurred during the scan of the

priority group is valid, and the scan of the low priority group is automatically executed after the scan of the priority group is completed.

When the ADGSPCR.GBRSCN bit is 1 and the ADGSPCR.LGRRS is 0, the scan of the low priority group is restarted from the beginning. Also, when ADGSPCR.LGRRS is 1, the scan of the low priority group is re-executed from the interrupted channel.

Table 39.7 shows the setting of the ADGSPCR.GBRSCN bit and the operation when a trigger is input during scanning.

When the ADGSPCR.GBRP bit is set to 1, the scan operation of the lowest priority group is the operation of executing a single scan continuously.

For the trigger setting in group scan mode, select the group A synchronous trigger with the ADSTRGR.TRSA[5:0] bits, and set the group B synchronous trigger different from the group A trigger with the ADSTRGR.TRSB[5:0] bits. Select a Group C synchronous trigger that is different from the Group A and B triggers on the ADGCTRGR.TRSC[5:0] bits.

When the group scan mode is set to 2 groups (ADGCTRGR.GRCE bit is set to 0) and the ADGSPCR.GBRP bit is set to 1, the ADSTRGR.TRSB[5:0] bits is set to 0x3F.

If the group scan mode is set to 3 groups (ADGCTRGR.GRCE bit is set to 1) and the ADGSPCR.GBRP bit is set to 1, set the ADGCTRGR.TRSC[5:0] bits to 0x3F.

For the channels to be scanned, select the group A channel in the ADANSA0 register, select the group B channel in the ADANSB0 register, and select the group C channel in the ADANSC0 register.

Table 39.8 shows Group priority operation setting and operation mode of 2 groups, and Table 39.9 shows Group priority operation setting and operation mode of 3 groups.

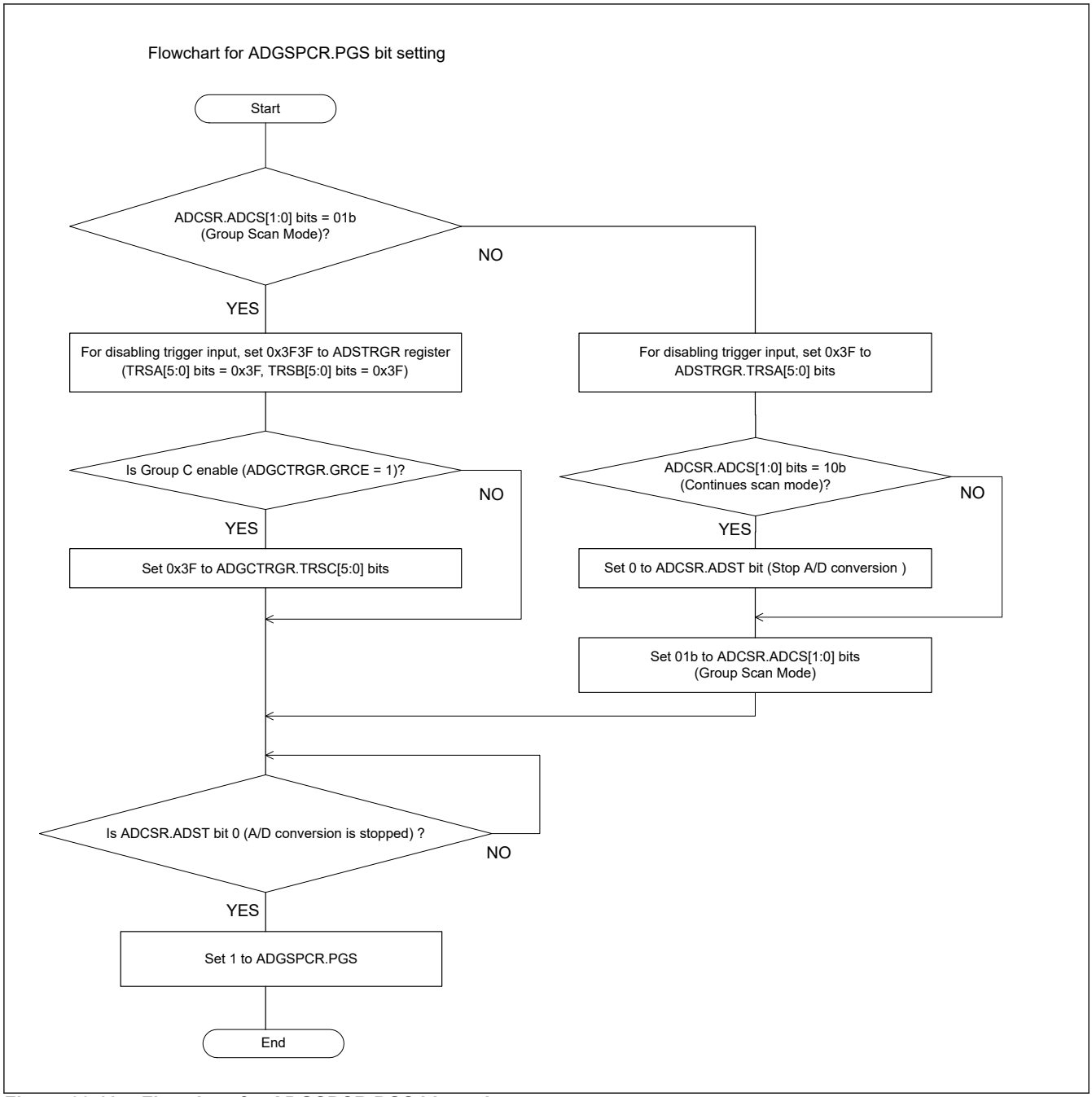


Figure 39.11 Flowchart for ADGSPCR.PGS bit setting

Table 39.7 Control of A/D conversion operations according to the settings of the ADGSPCR.GBRSCN bit (1 of 2)

A/D conversion operation	Trigger input	ADGSPCR.GBRSCN = 0	ADGSPCR.GBRSCN = 1
When A/D conversion for group A is in progress	Input of trigger for group A	Trigger input is ineffective.	Trigger input is ineffective.
	Input of trigger for group B	Trigger input is ineffective.	A/D conversion is performed on group B after A/D conversion on group A is completed.
	Input of trigger for group C	Trigger input is ineffective.	A/D conversion is performed on group C after A/D conversion on group A is completed.

Table 39.7 Control of A/D conversion operations according to the settings of the ADGSPCR.GBRSCN bit (2 of 2)

A/D conversion operation	Trigger input	ADGSPCR.GBRSCN = 0	ADGSPCR.GBRSCN = 1
When A/D conversion for group B is in progress	Input of trigger for group A	Conversion for group B that is in progress is discontinued and conversion for group A starts.	• Conversion in progress for group B is discontinued and conversion for group A starts. • Conversion for group B starts after conversion for group A is completed.
	Input of trigger for group B	Trigger input is ineffective.	Trigger input is ineffective.
	Input of trigger for group C	Trigger input is ineffective.	A/D conversion is performed on group C after A/D conversion on group B is completed.
When A/D conversion for group C is in progress	Input of trigger for group A	Conversion for group C that is in progress is discontinued and conversion for group A starts.	• Conversion in progress for group C is discontinued and conversion for group A starts. • Conversion for group C starts after conversion for group A is completed.
	Input of trigger for group B	Conversion for group C that is in progress is discontinued and conversion for group B starts.	• Conversion in progress for group C is discontinued and conversion for group B starts. • Conversion for group C starts after conversion for group B is completed.
	Input of trigger for group C	Trigger input is ineffective.	Trigger input is ineffective.

Table 39.8 Group priority operation setting and operation mode for 2 groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 0)

ADGSPCR			Operation mode
GBRSCN	LGRRS	GBRP	
0	—	0	Group priority operation of 2 groups (groups A and B). When the trigger of group A is input, the scan of group B ends (does not re-execute).
1	0	0	Group priority operation of 2 groups (groups A and B). After the scan of group B is interrupted, group B restarts the scan from the beginning of the channel specified by ADANSB0 after the scan of group A is completed.
1	1	0	Group priority operation of 2 groups (groups A and B). After the scan of group B is interrupted, group B resumes the scan from the interrupted channel among the channels specified by ADANSB0 after the scan of group A is completed.
—	0	1	Group priority operation of 2 groups (groups A and B). Group B continuously performs a single scan without inputting a start trigger. After the group B scan is interrupted, the single scan is restarted from the beginning of the channel specified by ADANSB0 after the group A scan is completed.
1	1	1	Group priority operation of 2 groups (groups A and B). Group B continuously performs a single scan without inputting a start trigger. After the scan of group B is interrupted, the single scan is restarted from the interrupted channel among the channels specified by ADANSB0 after the scan of group A is completed.

Table 39.9 Group priority operation setting and operation mode for 3 groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 1) (1 of 2)

ADGSPCR			Operation mode
GBRSCN	LGRRS	GBRP	
0	—	0	Group priority operation of 3 groups (groups A, B, C). When the trigger of group A is input, the scan of group B ends (does not re-execute). When the trigger of group A or group B is input, the scan of group C ends (does not re-execute).

Table 39.9 Group priority operation setting and operation mode for 3 groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 1) (2 of 2)

ADGSPCR			
GBRSCN	LGRRS	GBRP	Operation mode
0	—	1	Group priority operation of 3 groups (groups A, B, C). When the trigger of group A is input, the scan of group B ends (does not re-execute). Group C continuously performs a single scan without inputting a start trigger. After the group C scan is interrupted, the scan is restarted from the beginning of the channel specified by ADANSC0 after the group A/B scan is completed.
1	0	0	Group priority operation of 3 groups (groups A, B, C). After the scan of group B is suspended, the scan restarts from the top channel of the channels specified by ADANSB0 after the group A scan is completed. After the scan of group C is suspended, the scan restarts from the top channel of the channels specified by ADANSC0 after the group A/B scan is completed.
1	1	0	Group priority operation of 3 groups (groups A, B, C). After the scan of group B is suspended, the scan restarts from the interrupted channel of the channels specified by ADANSB0 after the group A scan is completed. After the scan of group C is suspended, the scan restarts from the interrupted channel of the channels specified by ADANSC0 after the group A/B scan is completed.
1	0	1	Group priority operation of 3 groups (groups A, B, C). After the scan of group B is interrupted, the scan is restarted from the beginning of the channel specified by ADANSB0 after the scan of group A is completed. Group C continuously performs a single scan without inputting a start trigger. After the group C scan is interrupted, the single scan is restarted from the beginning of the channel specified by ADANSC0 after the group A/B scan is completed.
1	1	1	Group priority operation of 3 groups (groups A, B, C). After the A/D conversion process of group B is interrupted, the interrupted channel is restarted from the channels specified by ADANSB0 after the scan of group A is completed. Group C continuously performs a single scan without inputting a start trigger. After the group C scan is interrupted, the single scan is restarted from the interrupted channel among the channels specified by ADANSC0 after the group A/B scan is completed.

39.3.5 Compare Function (Window A, Window B)

39.3.5.1 Compare Function Window A/B

The compare function is a function that compares the reference value set in the register with the A/D conversion result, and the reference value can be set for each window (A/B). Double trigger mode cannot be used when using the compare function. The major difference between window A and window B is that window B has one selectable channel and different interrupt output signals.

Operation using compare function in combination with continuous scan mode is described below.

1. When the ADCSR.ADST bit becomes 1 (A/D conversion start) by software, synchronous trigger (MTU3, ELC), or asynchronous trigger, A/D conversion of the selected channel is started.
2. When the A/D conversion is completed, the A/D conversion result is stored in the corresponding A/D data register (ADDRy). When ADCMPCR.CMPAE = 1b, if it is set for window A in the ADCMPANSR0 register, it is compared with the ADCMPDR0 register setting value. When ADCMPCR.CMPBE = 1b, if the ADCMPBNSR register is set for window B, it will be compared with the ADWINULB/ADWINLLB register setting value.
3. As a result of comparison, when the condition set in ADCMPLR0 is matched in window A, the flag (ADCMPSR0. CMPSTCHA0n) bit of the compare function window A is set to 1. At this time, if the ADCMPCR.CMPAIE bit is set to 1, an ADC0_CMPAI interrupt request (level) is generated. Similarly, when window B matches the conditions set in ADCMPBNSR.CMPLB, the compare function window B flag (ADCMPBSR.CMPSTB) bit is set to 1. At this time, if the ADCMPCR.CMPBIE bit is set to 1, an ADC0_CMPBI interrupt request (level) is generated.
4. When all the selected A/D conversions and comparisons are completed, scanning is performed again.
5. After accepting the ADC0_CMPAI/ADC0_CMPBI interrupt, set the ADCSR.ADST bit to 0 (A/D conversion stopped) and execute processing for the channel for which the compare flag is set.

6. When all the compare flags in window A are cleared, the ADC0_CMPAI interrupt request is canceled. Similarly, clearing the compare window B flag cancels the ADC0_CMPBI interrupt request. If you want to perform the compare again, start the A/D conversion again.

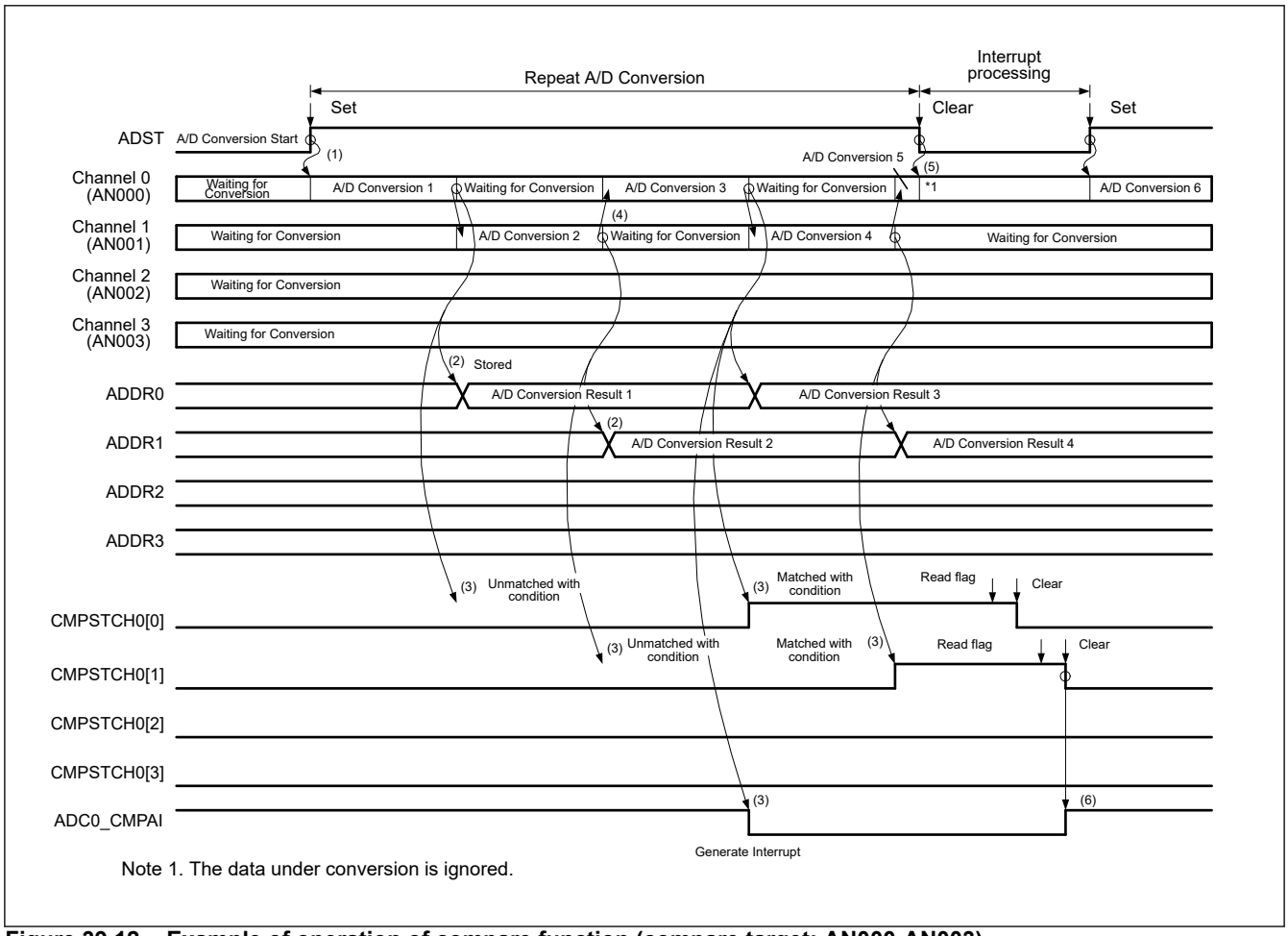


Figure 39.12 Example of operation of compare function (compare target: AN000-AN003)

39.3.5.2 Event output in Compare Function

For the event output of the compare function, specify the upper and lower reference values for each of windows A/B, compare the A/D conversion value of the selected channel with the upper/lower reference values, and then the event (ADCn_WCMPM/ADCn_WCMPUM) is output according to the event condition (A or B, A and B, A exor B) from the result of satisfying/failing the comparison condition of windows A and B. When using this function, perform A/D conversion in single scan mode.

In window A, any channel of ANn00-ANn15 can be selected. When multiple channels are selected in window A and the comparison condition is satisfied in one of the channels, the comparison condition is satisfied in window A. In window B, one channel of ANn00-ANn15 can be selected.

The setting procedure and setting example when using the event output of the compare function are shown below.

1. ADCSR.ADCS bit must be 00b (single scan mode).
2. Select the channel to be used for window A in the ADCMPANSR0 register, set the comparison condition of the compare function window in the ADCMPLR0 register, and set the upper/lower reference value in the ADCMPDR0/1 register.
3. Select the channel to be used for window B and the comparison condition in the ADCMPBNSR register, and set the upper/lower reference values in the ADWINULB/ADWINLLB registers.
4. In the ADCMPCR register, set the window A/B compound condition, window A/B operation permission, and interrupt output permission.

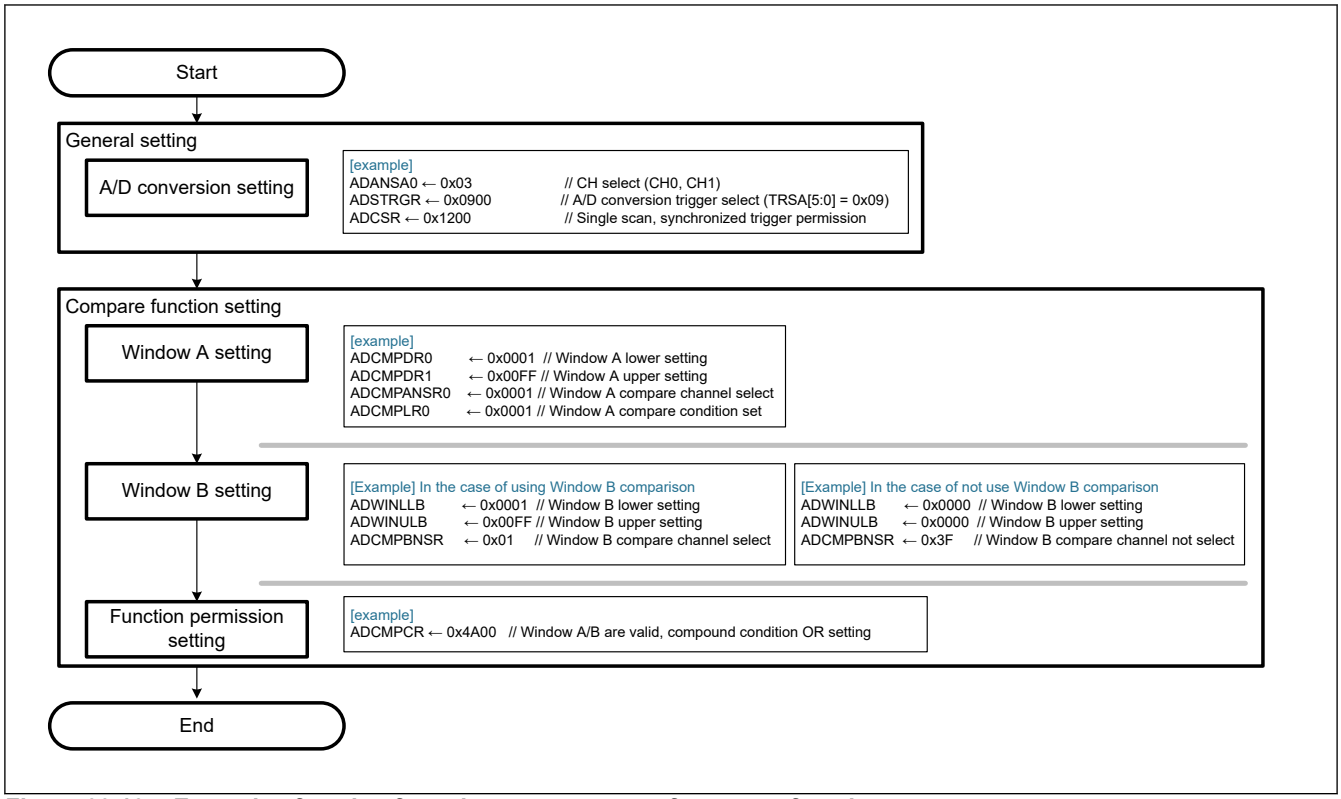


Figure 39.13 Example of setting for using event output of compare function

Precautions when using event output using only window A of the compare function

- Enable both windows A and B. (ADCMPCR.CMPAE = 1, ADCMPCR.CMPBE = 1)
- Set the compound condition setting for windows A and B to "OR condition". (ADCMPCR.CMPAB[1:0] = 00b)
- Set the comparison target channel in window B to "Unselected". (ADCMPBNSR.CMPCHB[5:0] = 111111b).
- Set the comparison condition of window B to 0 < conversion result < 0, which is "always inconsistent". (ADCMPCR.WCMPE = 1, ADWINLLB.CMPLLB[15:0] = ADWINULB.CMPULB[15:0] = 0x0000, ADCMPBNSR.CMPLB = 1)

Figure 39.14 shows example of event output operation of compare function.

The scan end event is output when one single scan ends. Also, depending on the setting of ADCMPCR.CMPAB[1:0], the match/unmatch event (ADCn_WCMPM/ADCn_WCMPUM) is output with a delay of 1 (PCLKH).

Note: Match/unmatch events are exclusive outputs, and both events are not output at the same time.

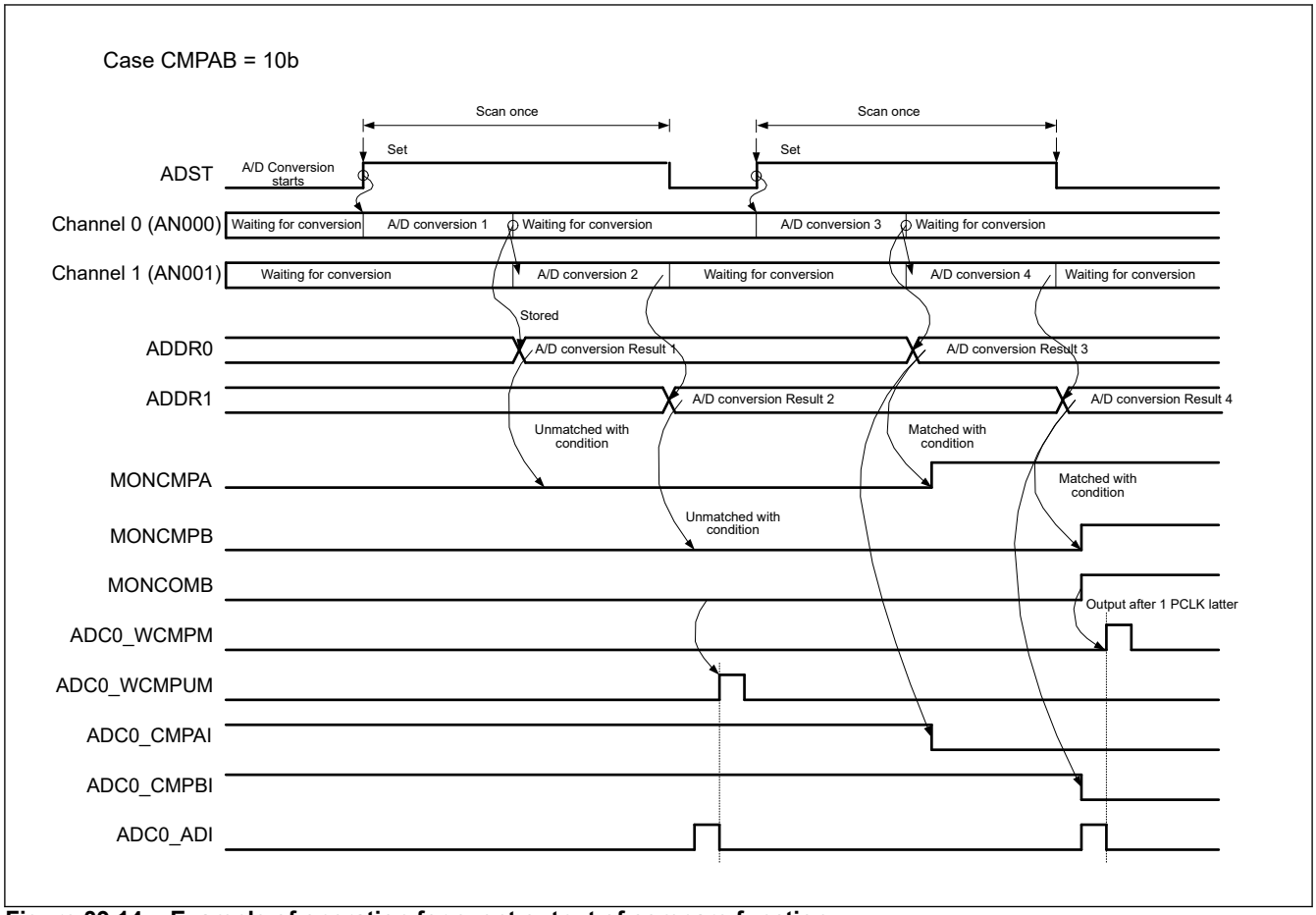


Figure 39.14 Example of operation for event output of compare function

Precautions when using event output of the compare function

The event output of the compare function outputs match/unmatch according to the setting of ADCMPCR.CMPAB[1:0] from the comparison result of window A and the comparison result of window B. The comparison result of window A is the logical sum of the comparison result of the channel to be compared in window A.

The comparison result of window A and the comparison result of window B are updated every A/D conversion of the channel to be compared, and the comparison result is retained even after the single scan is completed. To clear the comparison result (0), set ADCMPCR.CMPAE and ADCMPCR.CMPBE to 0.

39.3.5.3 Compare function constraint

The compare function has the following restrictions.

- Double trigger mode cannot be used together.
  (ADDBLDR, ADDBLDRA, ADDBLDRB are not subject to the compare function.)
- If you want to use match/unmatch event output, set the single scan mode.
- The same CH cannot be set in window A and window B.
- Set so that the upper reference value ≥ the lower reference value.

39.3.6 A/D Data Register Automatic Clearing Function

By setting the ADCER.ACE bit to 1, these are used when the CPU and DMAC read the A/D data registers (ADDRy, ADDBLDR, ADDBLDRA, ADDBLDRB). The register can be cleared automatically to 0x0000.

By using this function, it is possible to detect an un-updated failure of the ADDRy, ADDBLDR, ADDBLDRA, and ADDBLDRB registers. The following is an example when the automatic clear function of the ADDRy register is disabled/enabled.

When the ADCER.ACE bit is 0 (automatic clearing prohibited), the old data (0x0111) becomes the value of the ADDRy register when the A/D conversion result (0x0222) is not written to the ADDRy register for some reason. Furthermore, when the value of this ADDRy register is read to the general-purpose register by using the A/D conversion end interrupt, the old data (0x0111) can be saved in the general-purpose register. However, when checking for un-updated data, it is necessary to check while holding old data in RAM and general-purpose registers one by one.

When the ADCER.ACE bit is 1 (automatic clearing enabled), the ADDRy register is automatically cleared to 0x0000 when ADDRy = 0x0111 is read by the CPU and DMAC. After that, when 0x0222 of the A/D conversion result cannot be transferred to the ADDRy register for some reason, the cleared data (0x0000) remains as the ADDRy register value. If the value of this ADDRy register is read to a general-purpose register by using the A/D conversion end interrupt, 0x0000 is held in the general-purpose register. By simply checking that the read data value is 0x0000, it can be determined that there was an un-updated failure of the ADDRy register.

39.3.7 A/D-converted Value Addition/Average Function

In A/D-converted value addition function, the same channel is A/D-converted two, three, four or sixteen consecutive times and the sum of the converted values is stored in the data register. In A/D-converted value average function, the same channel is A/D-converted two or four consecutive times and the mean of the converted values is stored in the data register. The use of the average of these results can improve the accuracy of A/D conversion, depending on the types of noise components that affect A/D conversion. This function, however, cannot always guarantee an improvement in A/D conversion accuracy.

The A/D-converted value addition/average mode can be specified when A/D conversion of the analog input of the selected channels is selected.

39.3.8 Starting A/D Conversion with Asynchronous Trigger

The A/D conversion can be started by the input of an asynchronous trigger. To start up the A/D converter by an asynchronous trigger, the A/D conversion start trigger select bits (ADSTRGR.TRSA[5:0]) should be set to 000000b and a low-level signal should be input to the asynchronous trigger (ADTRGn#). Both the ADCSR.TRGE and ADCSR.EXTRG bits then should be set to 1. Figure 39.15 shows a timing of the asynchronous trigger input.

Asynchronous triggers cannot be selected for groups B and C used in group scan mode.

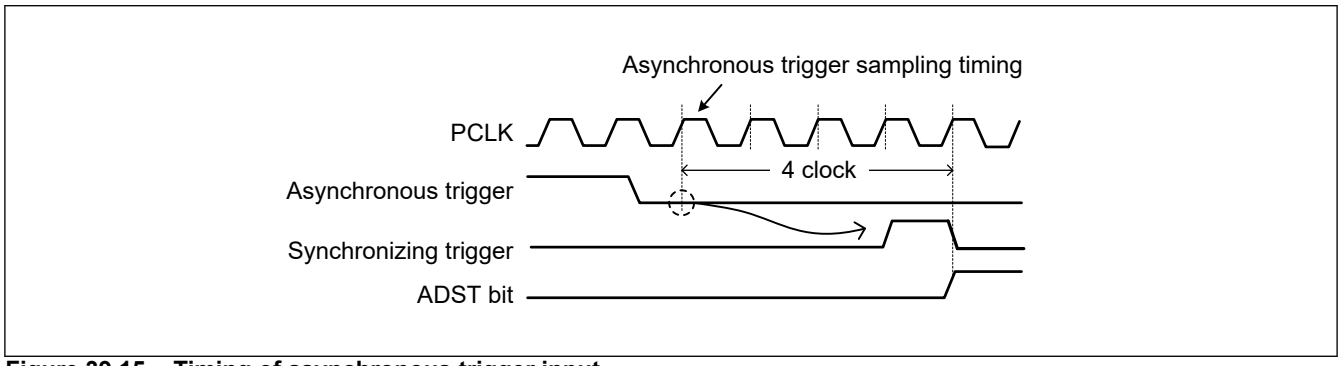


Figure 39.15 Timing of asynchronous trigger input

39.3.9 Starting A/D Conversion with Synchronous Trigger from Peripheral Module

A/D conversion can be started by the synchronization trigger from MTU3 and ELC (1 pulse of "L" in PCLK synchronization). When starting A/D conversion with a synchronous trigger, the ADCSR.TRGE bit is set to 1, the ADCSR.EXTRG bit is set to 0, the ADSTRGR.TRSA[5:0], ADSTRGR.TRSB[5:0], and ADGCTRGR.TRSC[5:0] bits are set to the appropriate trigger factor.

39.3.10 Event Link Function

39.3.10.1 Event output to ELC

ELC can use the ADCn_ADI interrupt request signal as an event signal (ADCn_ADI) to link to a preset module (ADCn_GBADI/ADCn_GCADI interrupt, ADCn_CMPAI/ADCn_CMPBI interrupt cannot be used as an event signal). The event signal can be output regardless of the setting of the corresponding interrupt request permission bit.

The 12-bit A/D converter outputs the scan end event (ADCn_ADI), compare function match event (ADCn_WCMPM), and unmatch event (ADCn_WCMPUM). The scan end event (ADCn_ADI) outputs a "High pulse" for 1 PCLK at the same output timing as the interrupt output (ADCn_ADI) regardless of the setting of ADCSR.ADIE.

The compare match/unmatch event (ADCn_WCMPM/ADCn_WCMPUM) to ELC outputs a "High pulse" for 1 PCLK at a timing delayed by 1 cycle (PCLK) from the interrupt output (ADCn_ADI) regardless of the setting of ADCSR.ADIE. When using the compare match/unmatch event to ELC (ADCn_WCMPM/ADCn_WCMPUM), set it to single scan mode.

39.3.10.2 Starting A/D Conversion with ELC event

The 12-bit A/D converter can start A/D conversion by a preset event by setting ELC_SSEL in ELC.

39.4 Usage Notes

39.4.1 Notes on Reading Data Registers

The A/D data registers, A/D data duplication registers, A/D data duplication register A, A/D data duplication register B, must be read in 16-bit word units, to avoid unmatch from upper value and lower value due to reading by byte units separately.

39.4.2 Notes on Stopping A/D Conversion

39.4.2.1 A/D Conversion Stop Procedure

If using asynchronous or synchronous trigger as the A/D conversion start condition, follow the procedure in the flowchart in Figure 39.16 to stop the A/D conversion.

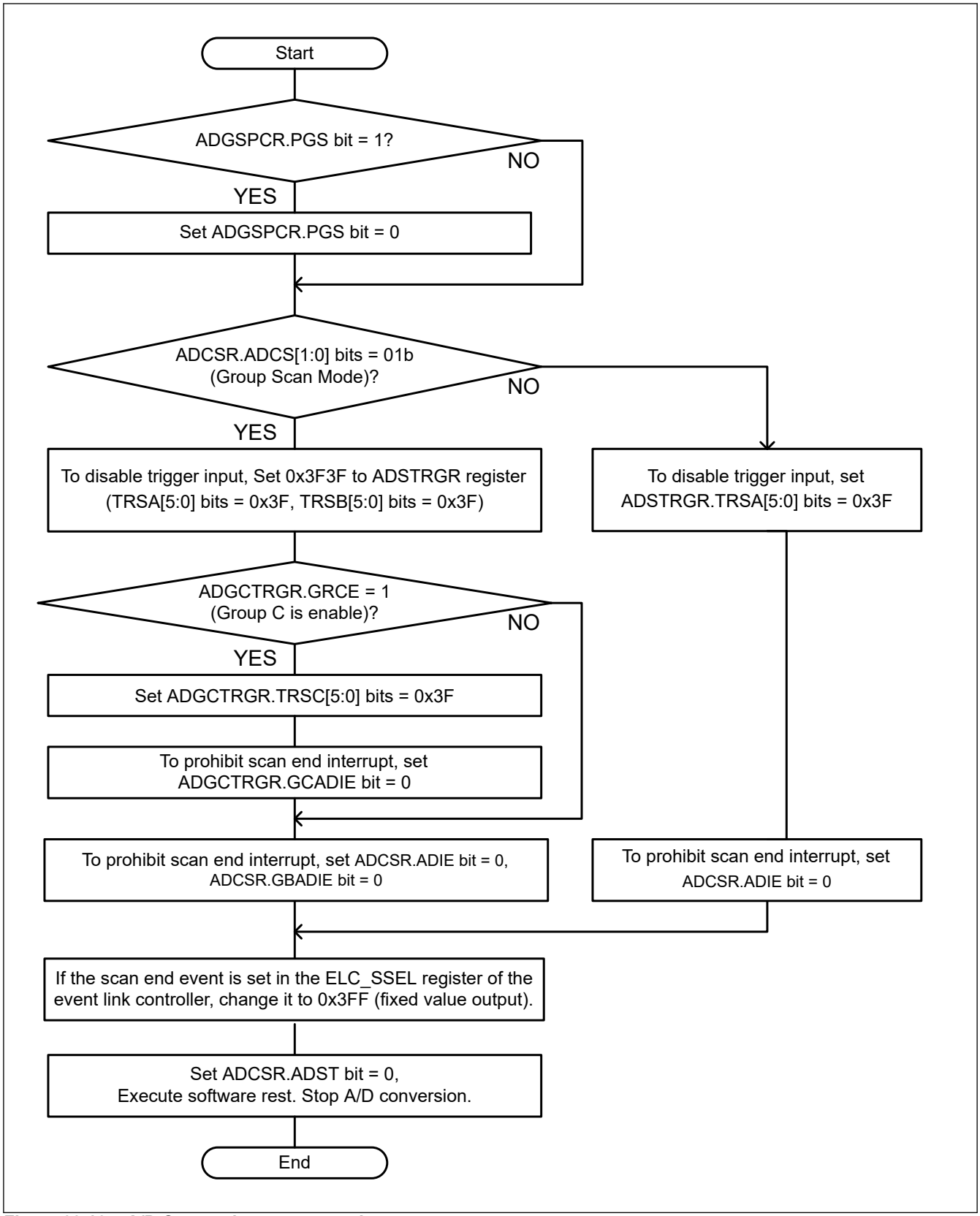


Figure 39.16 A/D Conversion stop procedure

It takes 2 PCLKADC cycles from a software clear until the soft macro stops scanning. If you want to set the following after executing a software clear, enter a wait of 2 PCLKADC cycles or more.

- Scan end interrupt enabled
- Event link controller scan end event valid setting

- Start of A/D conversion by software
- Trigger input valid setting

39.4.2.2 Notes on Mode/Status bits

Initialize or reset the odd/even judgment and monitor bits of the compare function in the double trigger mode as necessary.

- Double trigger mode starts from the first scan operation when ADCSR.DBLE is set from 0 to 1.
- To initialize the monitor bits (MONCMPA, MONCMPB, MONCOMB) of the compare function, set ADCMPCR.CMPAE and ADCMPCR.CMPBE to 0.

39.4.3 A/D conversion forced stop and operation timing at starts

Up to 6 PCLKADCs time is necessary that the A/D converter starts by ADCSR.ADST bit = 1 from stopped status.

When the ADCSR.ADST bit is set to 0 and the A/D conversion is forcibly stopped, the analog part of the 12-bit A/D converter requires a maximum of 2 PCLKADCs to stop operating.

39.4.4 Notes on Scan End Interrupt Processing

When scanning the same analog input twice using any trigger, if the CPU does not complete reading out the A/D-converted data by the time the A/D conversion of the first analog input for the second scan ends after the first scan end interrupt is generated, the first A/D-converted data is overwritten with the second A/D-converted data and an overwrite error occurs.

39.4.5 Module-Stop Function Setting

Operation of the 12-bit A/D converter can be disabled or enabled using the Module Stop Control register. The initial setting is for operation of the 12-bit A/D converter to be halted. Releasing the module-stop state enables access to the registers. After release from the module-stop state, wait for at least 15 us before starting A/D conversion.

39.4.6 Notes on Entering Low Power Consumption States

When transitioning to module stop mode, stop A/D conversion. When stopping the A/D conversion, it is necessary to secure the time until the analog part of the 12-bit A/D converter stops after setting the ADCSR.ADST bit to 0.

40. Temperature Sensor Unit (TSU)

40.1 Overview

The temperature sensor unit (TSU) indicates the junction temperature as a 12-bit binary digital output code. A temperature sensor unit is used to monitor the LSI junction temperature.

Figure 40.1 shows a block diagram of a temperature sensor unit.

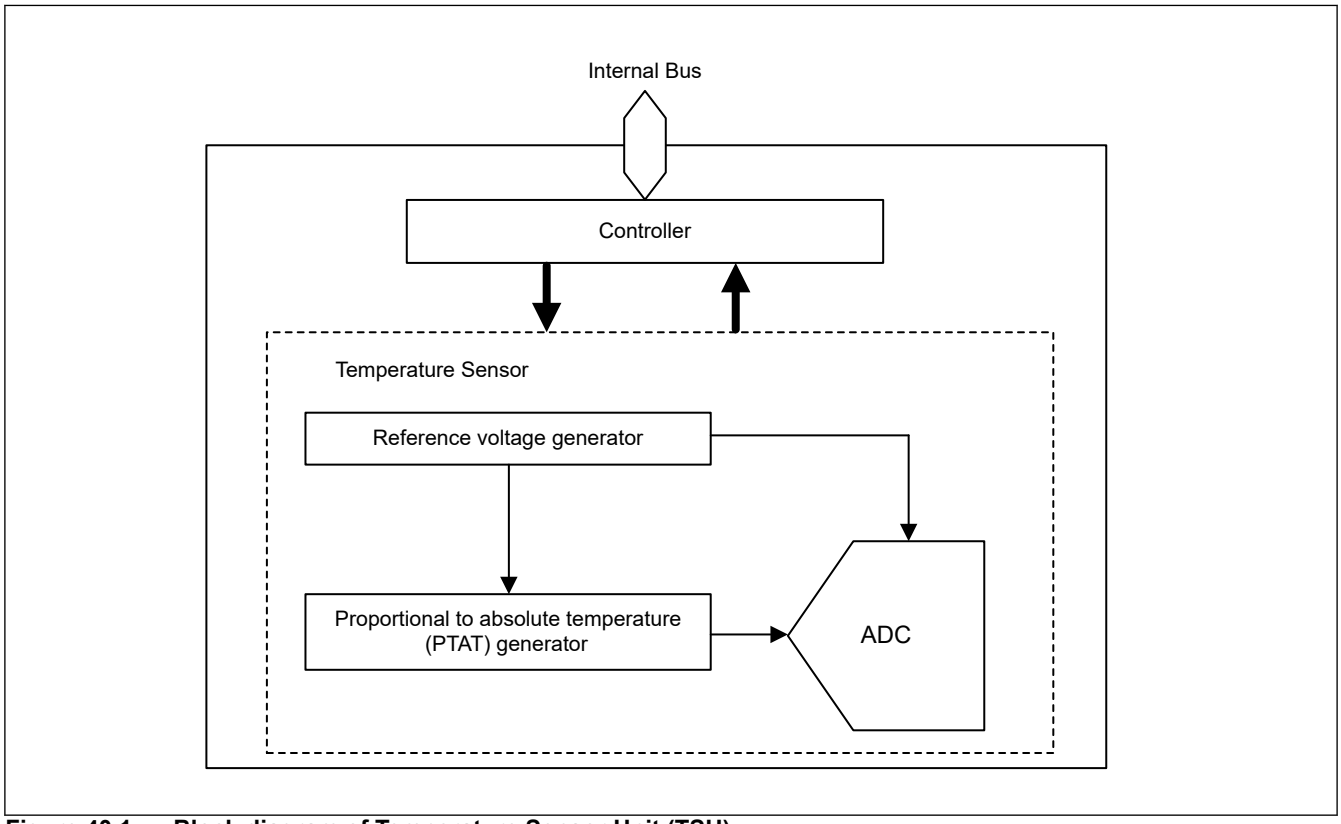


Figure 40.1 Block diagram of Temperature Sensor Unit (TSU)

40.2 Register Descriptions

40.2.1 TSUSM : Sensor Mode Register

Base address: TSU = 0x8004_6000

Offset address: 0x00

Bit position:	31																													1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADCEN	TSEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TSEN	Temperature Sensor Enable 0: Disabled 1: Enabled	R/W
1	ADCEN	ADC Enable 0: Disabled 1: Enabled	R/W
31:2	—	These bits are read as 0. The write value should be 0.	R/W

40.2.2 TSUST : Sensor Trigger Register

Base address: TSU = 0x8004_6000

Offset address: 0x04

Bit position:	31																													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	START
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	START	A/D Conversion Control 0: Stops A/D conversion*1 1: Starts A/D conversion*2	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. The TSUAD value is invalid when START bit is 0. Please read TSUAD before setting START bit to 0.
Note 2. Before the START bit is set to 1, TSUSM register and TSUSCS register settings are required. These registers are set and waits 60 µsec, then the START bit should be set to 1.

The TSUST register controls the temperature sensor unit to turn A/D convert on and off. After A/D conversion started, temperature conversion is executed and conversion data is updated every 960 PCLKL cycles at PCLKL = 50 MHz and every 640 PCLKL cycles at PCLKL = 37.5 MHz.

40.2.3 TSUSCS : Sensor Configuration Setting Register

Base address: TSU = 0x8004_6000

Offset address: 0x08

Bit position:	31																											3		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CKDIV	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	—	This bit is read as 1. The write value should be 1.	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	CKDIV	Divider Value for PCLKL 0: 24 for PCLKL = 50 MHz 1: 16 for PCLKL = 37.5 MHz	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

The TSUSCS register is used for temperature sensor configuration.

40.2.4 TSUSAD : Sensor ADC Data Register

Base address: TSU = 0x8004_6000

Offset address: 0x0C

Bit position:	31																			11										0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DOUT[11:0]										
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
11:0	DOUT[11:0]	Temperature Sensor Data Output	R
31:12	—	These bits are read as 0.	R

The TSUSAD register is used to read data for A/D conversion.

40.2.5 TSUSS : Sensor Status Register

Base address: TSU = 0x8004_6000

Offset address: 0x10

Bit position:	31																													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CONV
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CONV	A/D Conversion Status 0: Stop 1: A/D conversion is running	R
31:1	—	These bits are read as 0.	R

The TSUSS register indicates A/D conversion status.

40.3 Operation

40.3.1 Startup Sequence

This section describes the register operations for power consumption control and initial setting of this module. The startup sequence is as follows:

1. Release the TSU module from stop status.
   (a) Release write protection (PRCRN register).
   (b) Release the TSU module from the stopped state: MSTPCRD07 = 0.
   (c) Set write protection (PRCRN register).
   (d) Set TSUSCS.CKDIV = 0 (PCLKL = 50 MHz) or 1 (PCLKL = 37.5 MHz) depending on PCLK frequency.
2. Change the TSU status to normal operation mode.
   (a) Write to the TSUSM register and set the TSEN and ADCEN bits to 1.
   (b) Wait 60 µsec.
3. Start the conversion.
   (a) Write to the TSUST register and set the START bit to 1.
   A/D converter starts operation and converts the voltage generated by PTAT generator circuit associated temperature to 12-bit data. The CONV bit in the TSUSS register is set to 1.
   - A/D conversion data is stored in a buffer automatically.
   - Update the data every 960 PCLKL cycles (at TSUSCS.CKDIV = 0, PCLKL = 50 MHz : 19.2 µsec.) or 640 PCLKL cycles. (at TSUSCS.CKDIV = 1, PCLKL = 37.5 MHz : 17.1 µsec.)
   - Stored data in the buffer can be read out by TSUSAD register access. Read the data when TSUST.START = 1. When TSUST.START = 0, read data is invalid.

40.3.2 Stop Sequence

1. Stop the conversion.

(a) Clear the START bit in the TSUST register to 0.

(b) After conversion time has passed, A/D converter is stopped and the CONV bit in the TSUSS register is cleared to 0.

(c) Confirm CONV = 0.

2. Disable the TSU.

(a) Write to the TSUSM register and set the TSEN and ADCEN bits to 0. When restarting TSU operation, execute the startup sequence from Step 2 in section 40.3.1. Startup Sequence.

40.3.3 A/D Start and Stop Timing

A/D conversion signal is generated when the START bit in the TSUST register is set to 1.

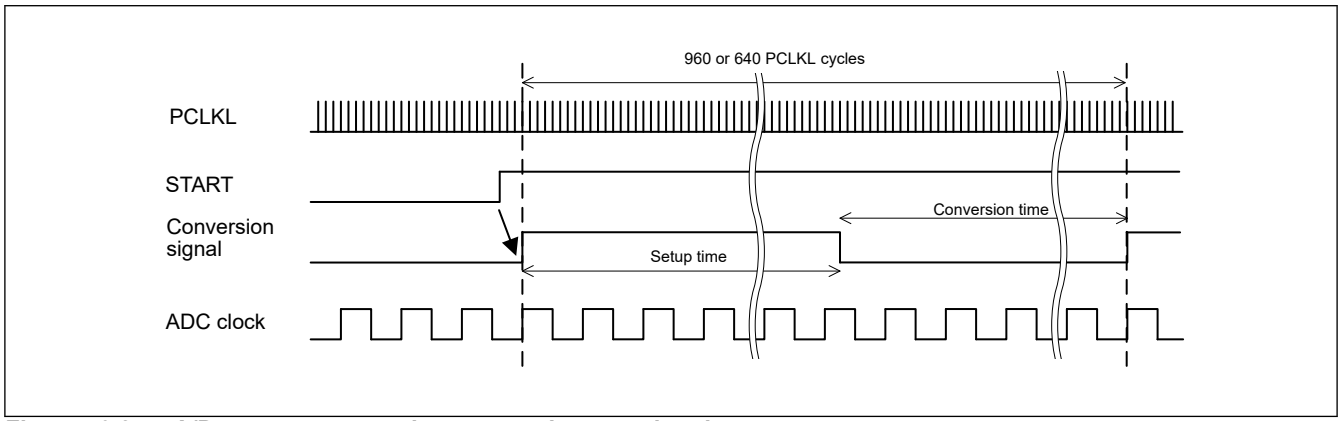


Figure 40.2 A/D converter operation start and conversion time

Even if the START bit in the TSUST register is cleared to 0, A/D converter is still working until the converting cycle is completed. However, do not read the A/D output data after the START bit in the TSUST register is changed to 0.

40.3.4 A/D Output Data Store Timing

When a conversion completion signal is detected, A/D output data is stored in a buffer. Data in the buffer can be read out by read access for TSUSAD register.

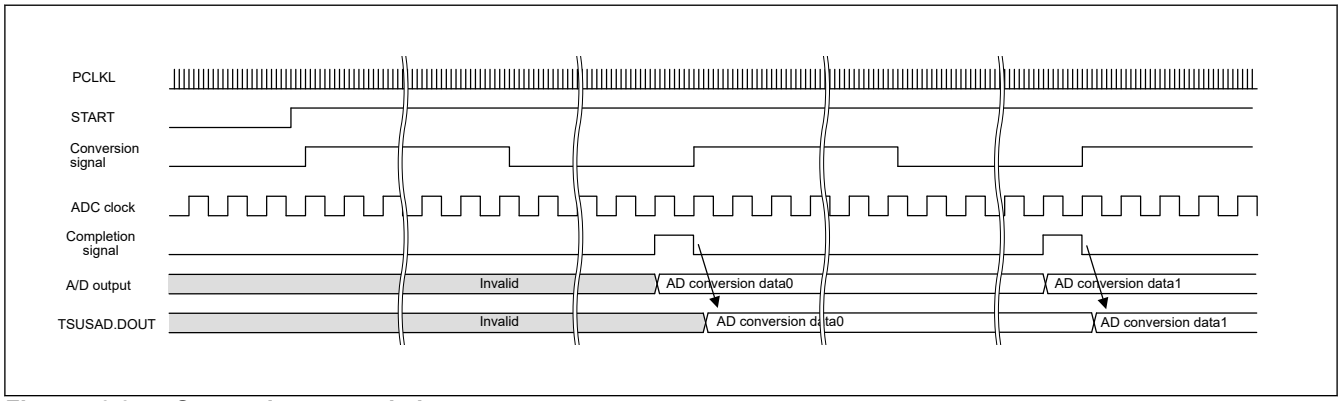


Figure 40.3 Output data store timing

40.3.5 A/D Conversion Status (CONV)

The TSU module enables or disables A/D converter according to the START bit status in the TSUST register. However, conversion completion timing is not the same as the START bit clear point.

In order to confirm the A/D conversion status, TSUSS register is available. When the START bit in TSUST register is set to 1, CONV bit in the TSUSS register is also set to 1. When the START bit in TSUST register is cleared to 0, the CONV bit status in the TSUSS register remains until conversion completion signal is asserted.

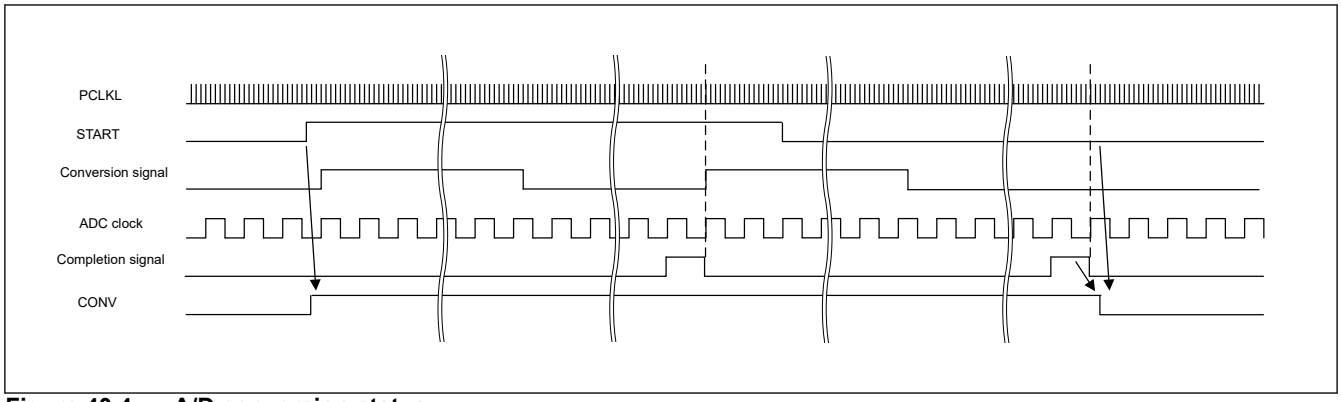


Figure 40.4 A/D conversion status

40.3.6 Using the Temperature Sensor

The temperature sensor outputs 12-bit binary digital output code which varies with the temperature. The user can then obtain the temperature on this LSI by converting the output code value into temperature. To ensure accuracy, 2-point calibration (see section 40.3.6.1. Preparation for Using the Temperature Sensor) and also averaging the data obtained at least 8 times continuously are required.

40.3.6.1 Preparation for Using the Temperature Sensor

The voltage output by the temperature sensor is proportional to temperature, which can be calculated according to the following formula (2-point calibration).

Formula for the temperature characteristic:

$T = (Ds - D1) / Slope + T1$ … (1)

T: Measured temperature (°C)

Ds: Code output (TSUSAD.DOUT) by the temperature sensor at the time of temperature measurement

T1 = 25°C: Temperature experimentally measured at one point (°C)

D1: Code output (TSUSAD.DOUT) by the temperature sensor at the time of measurement of T1

T2 = 85°C: Temperature at the experimental measurement of another point (°C)

D2: Code output (TSUSAD.DOUT) by the temperature sensor at the time of measurement of T2

Slope: Temperature gradient by the temperature sensor (digit/°C)

For temperature more than or equal to 25°C (for example, if Ds ≥ D1)

$Slope = (D2 - D1) / (T2 - T1)$ … (2-1)

For temperature below 25°C (for example, if Ds < D1)

$Slope = (D2 - D1) / (T2 - T1) \times 63 / 65$ … (2-2)

Characteristics vary from sensor to sensor. Therefore, the following measurement at two different temperatures (T1 = 25°C and T2 = 85°C) is recommended.

Measure the code D1 output by the temperature sensor at temperature T1 (25°C).

Again, measure the code D2 output by the temperature sensor at a different temperature T2 (85°C). Obtain the temperature gradient (Slope: the formula (2-1) or (2-2) above) from these results.

Subsequently, obtain temperatures by substituting Slope and code output (Ds) into the formula (1) above.

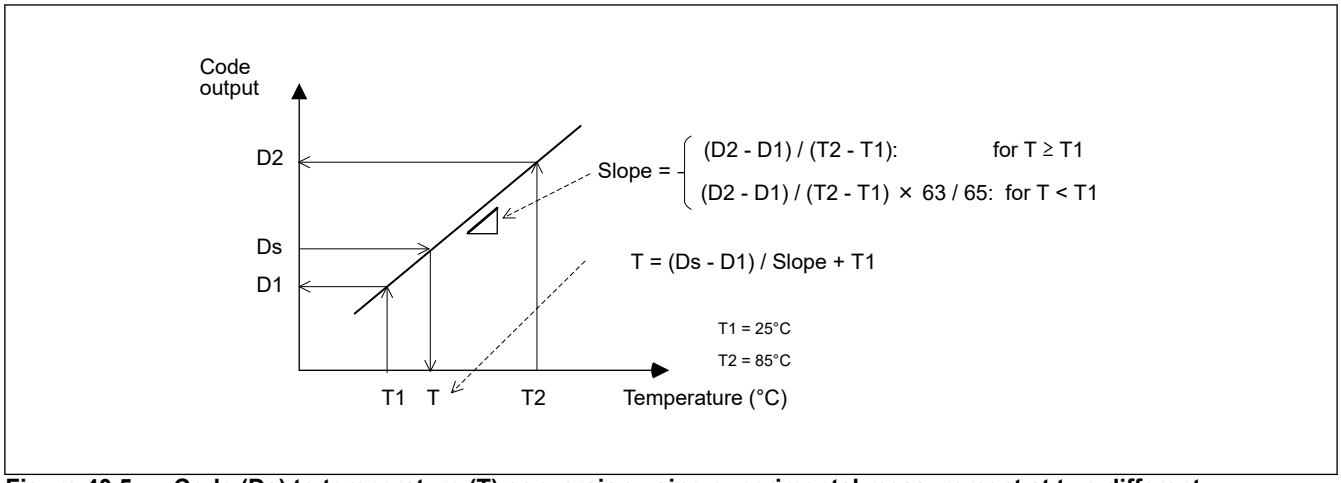


Figure 40.5 Code (Ds) to temperature (T) conversion using experimental measurement at two different temperatures

41. Data Operation Circuit (DOC)

41.1 Overview

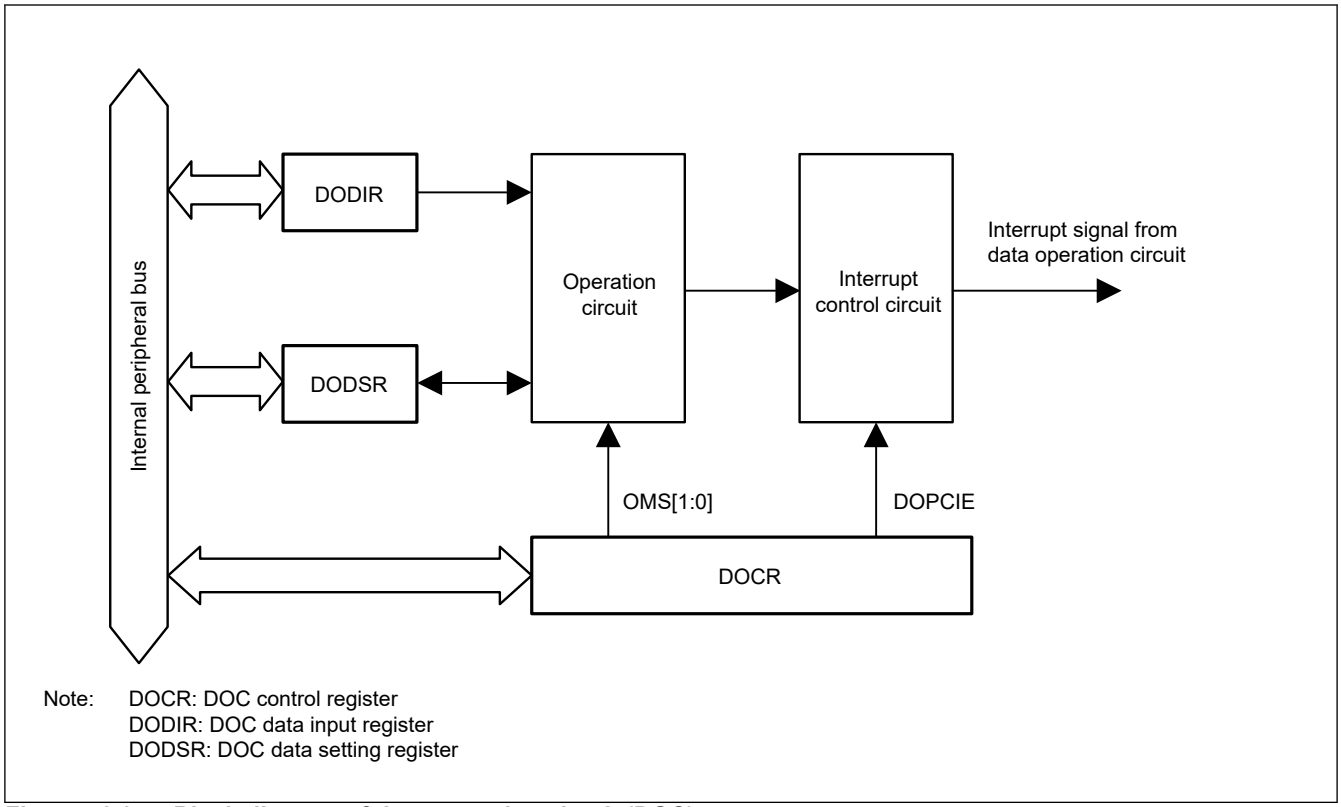
The data operation circuit (DOC) is used to compare, add, and subtract 16-bit data.

Table 41.1 lists the data operation circuit specifications and Figure 41.1 shows a block diagram of the data operation circuit.

- 16-bit data is compared and an interrupt can be generated when a selected condition applies.
- 16-bit data can be added.
- 16-bit data can be subtracted.

Table 41.1 DOC specifications

Parameter	Description
Data operation function	16-bit data comparison, addition, and subtraction
Lower power consumption function	Module-stop state can be set.
Interrupts	An interrupt occurs at the following timings: • The compared values either match or mismatch • The result of data addition is greater than 0xFFFF • The result of data subtraction is less than 0x0000



Note: DOCR: DOC control register
DODIR: DOC data input register
DODSR: DOC data setting register

Figure 41.1 Block diagram of data operation circuit (DOC)

41.2 Register Descriptions

41.2.1 DOCR : DOC Control Register

Base address: DOC = 0x8004_4000

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	DOPCFCL	DOPCF	DOPCIE	—	DCSEL	OMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	OMS[1:0]	Operating Mode Select Selects an operating mode of the data operation circuit. 0 0: Data comparison mode 0 1: Data addition mode 1 0: Data subtraction mode 1 1: Setting prohibited	R/W
2	DCSEL	Detection Condition Select Selects a condition for detecting data comparison result. This bit is valid only in the data comparison mode. 0: Data mismatch is detected. 1: Data match is detected.	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
4	DOPCIE	Data Operation Circuit Interrupt Enable Specifies whether to enable or disable data operation circuit interrupt requests. 0: Disables interrupts from the data operation circuit. 1: Enables interrupts from the data operation circuit.	R/W
5	DOPCF	Data Operation Circuit Flag [Setting conditions] When any of the following is met: ● The condition selected by the DCSEL bit is met. ● A result of data addition is greater than 0xFFFF. ● A result of data subtraction is less than 0x0000. [Clearing condition] Writing 1 to the DOPCFCL bit	R
6	DOPCFCL	DOPCF Flag Clear 0: No effect 1: Clear the DOPCF flag	W
7	—	This bit is read as 0. The write value should be 0.	R/W

41.2.2 DODIR : DOC Data Input Register

Base address: DOC = 0x8004_4000

Offset address: 0x02

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	n/a	The DODIR register is a 16-bit readable/writable register in which 16-bit data for use in operations are stored.	R/W

41.2.3 DODSR : DOC Data Setting Register

Base address: DOC = 0x8004_4000

Offset address: 0x04

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:																
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	n/a	The DODSR register is a 16-bit readable/writable register. This register stores 16-bit data for use as a reference in data comparison mode. This register also stores the results of operations in data addition and data subtraction modes.	R/W

41.3 Operation

41.3.1 Data Comparison Mode

Figure 41.2 shows an example of the steps involved in data comparison mode operation by the data operation circuit. The following is an example of operation when DCSEL is set to 0 (data mismatch is detected as a result of data comparison).

1. Writing 00b to the DOCR.OMS[1:0] bits selects data comparison mode.
2. The 16-bit reference data is set in DODSR.
3. 16-bit data for comparison is written to DODIR.
4. Writing of 16-bit data continues until all data for comparison have been written to DODIR.
5. If a value written to DODIR does not match that in DODSR, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

Note: When the DODIR register has the value after reset (0x0000), the DOCR.DOPCF flag does not change to 1 if a value other than 0x0000 is set in the DODSR register. To perform data comparison, set the DODSR register and write a value to the DODIR register. If the DODSR register is set again after data comparison, the value remaining in the DODIR register will not be compared again with the value set in the DODSR register. That is, comparison proceeds in response to writing to the DODIR register but not in response to writing to the DODSR register.

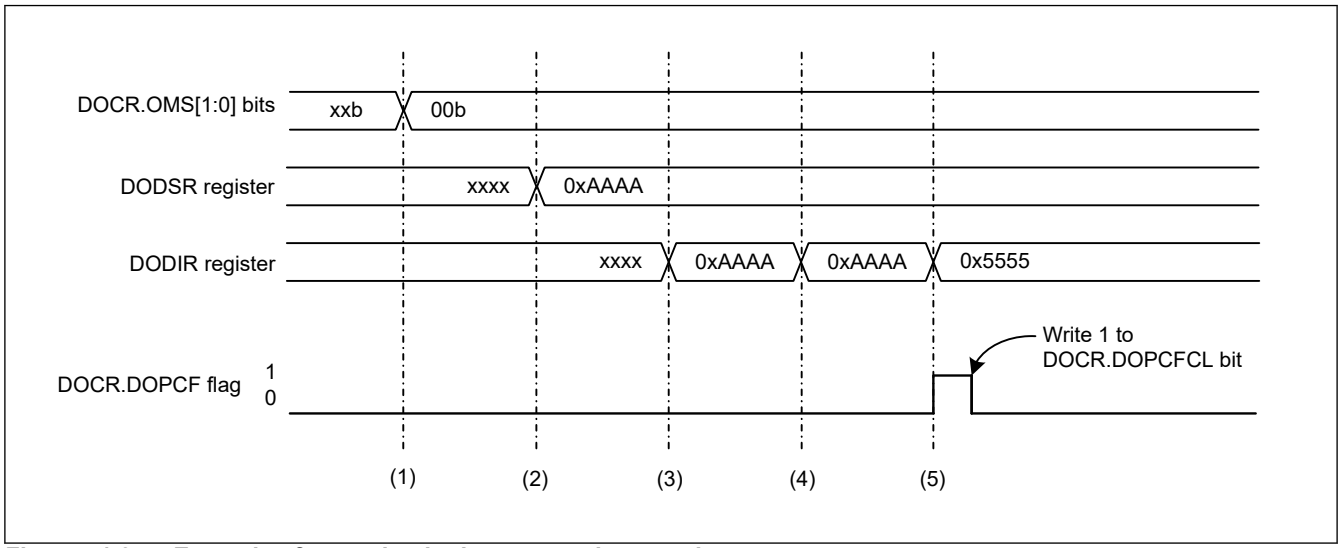


Figure 41.2 Example of operation in data comparison mode

41.3.2 Data Addition Mode

Figure 41.3 shows an example of the steps involved in data addition mode operation by the data operation circuit.

1. Writing 01b to the DOCR.OMS[1:0] bits selects data addition mode.
2. 16-bit data is set in the DODSR register as the value after reset.
3. 16-bit data to be added is written to DODIR. The result of the operation is stored in DODSR.
4. Writing of 16-bit data continues until all data for addition have been written to DODIR.
5. If the result of an operation is greater than 0xFFFF, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

Note: When the DODIR register has the value after reset (0x0000), data addition is not performed if a value other than 0x0000 is set in the DODSR register. To perform data addition, set the DODSR register and write a value to be added into the DODIR register. If the DODSR register is set again after data addition, the value remaining in the DODIR register will not be added again to the value set in the DODSR register. That is, addition proceeds in response to writing to the DODIR register but not in response to writing to the DODSR register.

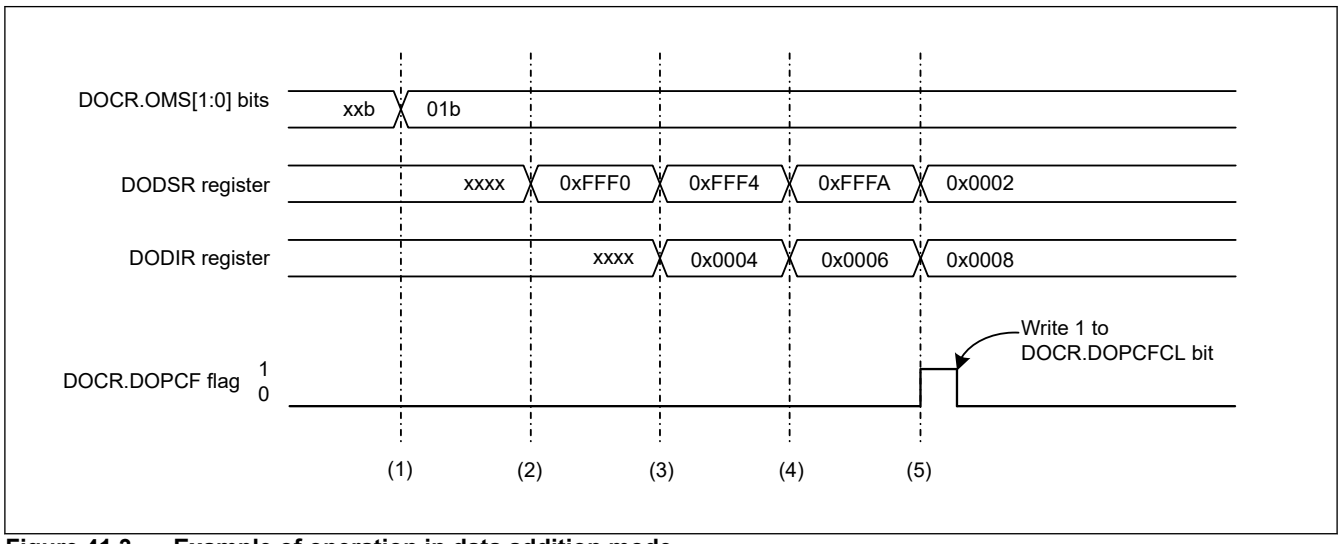


Figure 41.3 Example of operation in data addition mode

41.3.3 Data Subtraction Mode

Figure 41.4 shows an example of the steps involved in data subtraction mode operation by the data operation circuit.

1. Writing 10b to the DOCR.OMS[1:0] bits selects data subtraction mode.
2. 16-bit data is set in the DODSR register as the value after reset.
3. 16-bit data to be subtracted is written to DODIR. The result of the operation is stored in DODSR.
4. Writing of 16-bit data continues until all data for subtraction have been written to DODIR.
5. If the result of an operation is less than 0x0000, the DOCR.DOPCF flag is set to 1. When the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also generated.

Note: When the DODIR register has the value after reset (0x0000), data subtraction is not performed if a value other than 0x0000 is set in the DODSR register. To perform data subtraction, set the DODSR register and write a value to be subtracted into the DODIR register. If the DODSR register is set again after data subtraction, the value remaining in the DODIR register will not be subtracted again from the value set in the DODSR register. That is, subtraction proceeds in response to writing to the DODIR register but not in response to writing to the DODSR register.

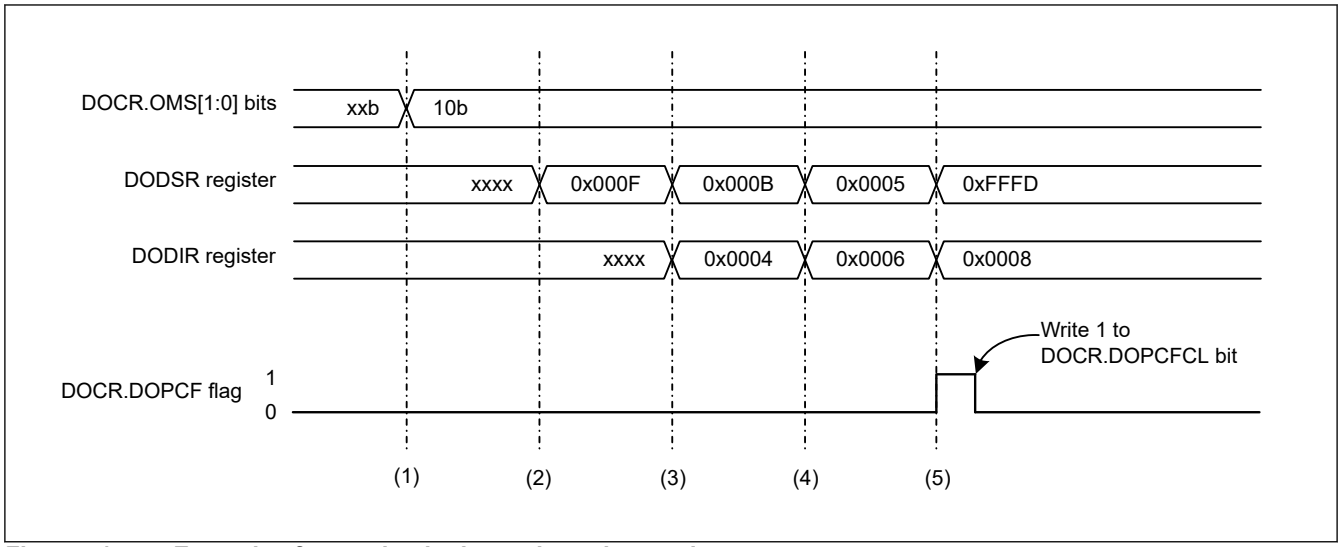


Figure 41.4 Example of operation in data subtraction mode

41.4 Interrupt Requests

The data operation circuit generates the data operation circuit interrupt as an interrupt request. When an interrupt source is generated, the data operation circuit flag (DOCR.DOPCF) corresponding to the interrupt is set to 1. Table 41.2 describes the interrupt request.

Table 41.2 Interrupt request from data operation circuit

Interrupt request	Data operation circuit flag	Interrupt generation timing
Data operation circuit interrupt	DOPCF	• The result of data addition is greater than 0xFFFF • The result of data subtraction is less than 0x0000 • The compared values either match or mismatch

41.5 Usage Note

41.5.1 Module Stop Function Setting

Operation of the data operation circuit can be disabled or enabled using module stop control register D (MSTPCRD). The value after reset indicates that the data operation circuit is in the stop state. Register access is enabled by canceling the module-stop state. For details, see section 8, Low-Power Consumption Function.

42. System SRAM (SYSRAM)

42.1 Overview

The LSI has an on-chip high-speed RAM (with ECC error correction). The RAM, whose capacity is 1.0 MB (512 KB × 2), is installed. Table 42.1 lists the System SRAM (SYSRAM) specifications and Figure 42.1 shows a block diagram of SYSRAM.

Table 42.1 System SRAM specifications

Item	Description
RAM capacity	1.0 MB (512 KB × 2)
Operating frequency	150 MHz or 200 MHz (PCLKH)
RAM address	SYSRAM0: 0x1000_0000 to 0x1007_FFFF SYSRAM1: 0x1008_0000 to 0x100F_FFFF Each RAM module consists of 4 pages of 128 KB
Shortest Latency	• ECC enabled Writing 32/64/96/128-bit transfer: Latency 0 (write to RAM in 1 cycle) 8/16-bit transfer: Latency 2 (write to RAM in 3 cycles) Reading All data transfer: Latency 2 (read from RAM in 3 cycles) • ECC disabled Writing All data transfer: Latency 0 (write to RAM in 1 cycle) Reading All data transfer: Latency 1 (read from RAM in 2 cycles)
Error checking	7-bit ECC data to 32-bit RAM data 1-bit error correction and 2-bit error detection are supported.
Interrupts	3 interrupts are available for each SYSRAMn. They can be used as error events.
Bus error response	SLVERR is returned to the access to the System SRAM. • Write access occurs when write access is disabled by VLWEN bit. • Write or read access occurs when access is disabled by VCEN bit. • Write access occurs when write access is disabled by VRWEN bit to corresponding page. • 2-bit ECC error occurs when read access is done. This response is maskable.
Test function (by error injection)	• Register values can be used as test data to be written to RAM, those in ECEDB[31:0] to the RAM data section and ECERDB[6:0] to the ECC redundancy bit section, respectively. • The ECC redundancy bits included in the RAM data are retained and read as a register value. • The ECC redundancy bits (encoder circuit) generated from the input data and the syndrome code (decoder circuit) can be checked.
Initializing function	Supported

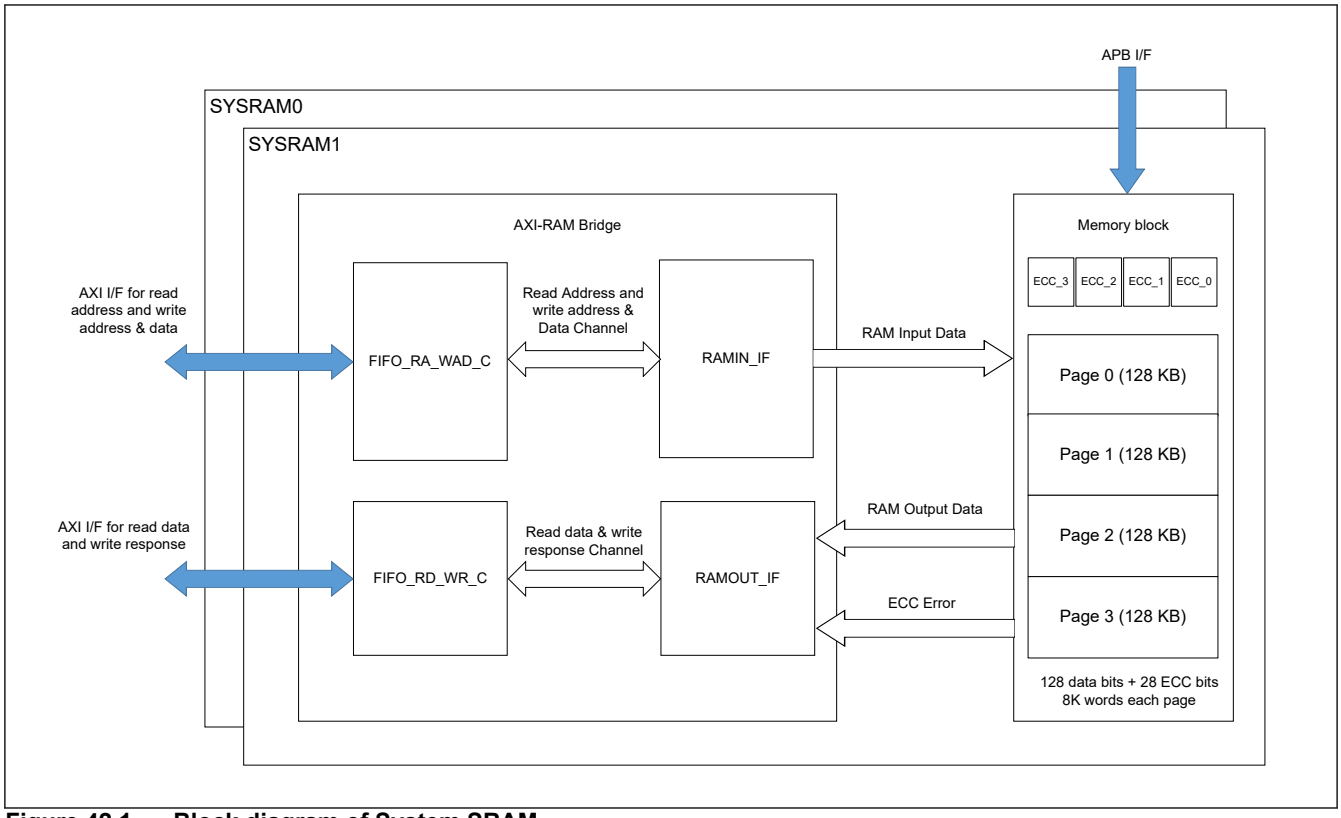


Figure 42.1 Block diagram of System SRAM

42.2 Register Descriptions

42.2.1 SYSRAM_CTRLn : System SRAM Control Register n (n = 0, 1)

Base address: SYSRAM_CTL = 0x8128_1800

Offset address: 0x000 + 0x010 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	MKICCAXIERR	—	—	VLWEN	VCEN	VRWEN[3:0]			
Value after reset:	0	0	0	0	0	0	0	1	0	0	1	1	1	1	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	VECEN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	VECEN	Enables or disables error correction with ECC 0: Disabled 1: Enabled	R/W
15:1	—	These bits are read as 0. The write value should be 0.	R/W
19:16	VRWEN[3:0]	Enables write for each page of RAM The bit n corresponds to the page n (n = 0 to 3). 0: Disabled 1: Enabled	R/W
20	VCEN	Enables access to RAM 0: Disabled 1: Enabled	R/W

Bit	Symbol	Function	R/W
21	VLWEN	Enables write for RAM 0: Disabled 1: Enabled	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
24	MKICCAXIERR	Controls AXI-SLVERR issuance for ECC 2-bit errors 0: Issue AXI-SLEVER 1: Do not issue AXI-SLEVER	R/W
31:25	—	These bits are read as 0. The write value should be 0.	R/W

The SYSRAM_CTRLn register is a 32-bit register that controls ECC of the SYSRAMn.

42.2.2 Wn_EC710CTL : ECC Control Register n (n = 0 to 3)

Base address: SYSRAMi = 0x8104_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x000 + 0x040 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	ECEDF7[1:0]		ECEDF6[1:0]		ECEDF5[1:0]		ECEDF4[1:0]		ECEDF3[1:0]		ECEDF2[1:0]		ECEDF1[1:0]		ECEDF0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	EMCA[1:0]		—	—	ECOVFF	ECER2C	ECER1C	—	ECTHM	ECERVF	EC1ECP	EC2EDIC	EC1EDIC	ECER2F	ECER1F	ECEMF
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bit	Symbol	Function	R/W
0	ECEMF	ECC Error Indicate Flag Indicates that an error is detected in the current read data. This bit is updated for each 32-bit RAM output data. When ECC error determination is disabled (ECERVF = 0) or the ECC function through mode is set (ECTHM = 1), this bit is always 0. 0: No error is detected in the current read data 1: An error is detected in the current read data	R
1	ECER1F	1-Bit ECC Error Detection/Correction Flag Indicates that a 1-bit error is detected and corrected in the 32-bit read data plus 7 ECC bits (total 39 bits) while the ECC error determination is enabled. 0: 1-bit error has not occurred 1: 1-bit error has occurred*1	R
2	ECER2F	2-Bit ECC Error Detection Flag Indicates that a 2-bit error is detected in the 32-bit read data plus 7 ECC bits (total 39 bits) while the ECC error determination is enabled. 0: 2-bit error has not occurred 1: 2-bit error has occurred*2	R
3	EC1EDIC	1-Bit ECC Error Detection Interrupt Control Enables or disables to output an interrupt signal (SRAMn_IE1) on detection of a 1-bit ECC error. 0: Disables to output an interrupt signal 1: Enables to output an interrupt signal	R/W
4	EC2EDIC	2-Bit ECC Error Detection Interrupt Control Enables or disables to output an interrupt signal (SRAMn_IE2) on detection of a 2-bit ECC error. 0: Disables to output an interrupt signal 1: Enables to output an interrupt signal	R/W
5	EC1ECP	1-Bit ECC Error Correction Enable Enables or disables error correction when a 1-bit ECC error is detected. 0: Error correction is performed on detection of a 1-bit ECC error 1: Error correction is not performed on detection of a 1-bit ECC error	R/W

Bit	Symbol	Function	R/W
6	ECERVF	ECC Error Determination Enable To change this bit, it is necessary to write 01b to bits 15 and 14 at the same time to permit writing. 0: The error determination is disabled 1: The error determination is enabled. The ECC function according to EC1EDIC, EC2EDIC, and EC1ECP bits are enabled.	R/W
7	ECTHM	ECC Function Through Mode Enable Enables or disables ECC function Through mode. If Through mode is enabled, no ECC error determination and no error correction is performed. To change this bit, it is necessary to write 01b to bits 15 and 14 at the same time to permit writing. 0: Disables Through mode 1: Enables Through mode. No effect on encoder. Decoder does not determine the error and does not perform error correction.	R/W
8	—	This bit is read as 0. The write value should be 0.	R/W
9	ECER1C	1-Bit ECC Error Detection Clear Clears the 1-bit ECC error detection flag and the error address related flags for the 32-bit read data. Writing 1 to this bit clears the ECER1F, ECOVFF, and ECEDF7[1:0] to ECEDF0[1:0] bits. When a conflict occurs between 1-bit ECC error detection and writing 1 to this bit, writing to this bit is given priority and the 1-bit ECC error detection flag is not set.	R/W
10	ECER2C	2-Bit ECC Error Detection Clear Clears the 2-bit ECC error detection flag and the error address related flags for the 32-bit read data. Writing 1 to this bit clears the ECER2F, ECOVFF, and ECEDF7[1:0] to ECEDF0[1:0] bits. When a conflict occurs between 2-bit ECC error detection and writing 1 to this bit, writing to this bit is given priority and the 2-bit ECC error detection flag is not set.	R/W
11	ECOVFF	ECC Error Address Capture Overflow Flag Indicates that the error address fails to be stored in the ECC error address register when an ECC error occurs. When an ECC error address capture overflow occurs, SRAMn_OVF interrupt is output. If an ECC error address capture overflow occurs again while this flag is 1, SRAMn_OVF interrupt is output again. 0: ECC error address capture overflow has not occurred 1: ECC error address capture overflow has occurred	R
13:12	—	These bits are read as 0. The write value should be 0.	R/W
15:14	EMCA[1:0]	Access Control to ECC Mode Selection Setting these bits to 01b enables writing to bits 7 and 6 of this register. Otherwise, writing to bits 7 and 6 is ignored. These bits are always read as 00b. 0 1: Writing is enabled Others: Writing is disabled	R/W
17:16	ECEDF0[1:0]	ECC Error Address Capture Flag m (m = 0) Indicates the error address register and the error type. When an ECC error occurs and its error type can be identified, the following value is read from these bits. 0 0: ECC error address is not captured in Wn_EC710EADm 0 1: 1-bit ECC error address is captured in Wn_EC710EADm 1 0: 2-bit ECC error address is captured in Wn_EC710EADm	R
19:18	ECEDF1[1:0]	ECC Error Address Capture Flag m (m = 1) This function is the same as ECEDF0.	R
21:20	ECEDF2[1:0]	ECC Error Address Capture Flag m (m = 2) This function is the same as ECEDF0.	R
23:22	ECEDF3[1:0]	ECC Error Address Capture Flag m (m = 3) This function is the same as ECEDF0.	R
25:24	ECEDF4[1:0]	ECC Error Address Capture Flag m (m = 4) This function is the same as ECEDF0.	R
27:26	ECEDF5[1:0]	ECC Error Address Capture Flag m (m = 5) This function is the same as ECEDF0.	R
29:28	ECEDF6[1:0]	ECC Error Address Capture Flag m (m = 6) This function is the same as ECEDF0.	R

Bit	Symbol	Function	R/W
31:30	ECEDF7[1:0]	ECC Error Address Capture Flag m (m = 7) This function is the same as ECEDF0.	R

Note 1. When a 1-bit ECC error is detected while EC1EDIC bit is 1 and ECER1F bit is 0, SRAMn_IE1 interrupt is output. If a 1-bit ECC error is detected while the ECER1F bit is already 1, the SRAMn_IE1 interrupt is not output.
Note 2. When a 2-bit ECC error is detected while EC2EDIC bit is 1 and ECER2F bit is 0, SRAMn_IE2 interrupt is output. If a 2-bit ECC error is detected while the ECER2F bit is already 1, the SRAMn_IE2 interrupt is not output.

The Wn_EC710CTL register indicates operation settings and error status at the time of an ECC error occurrence.

Because RAM block consists of 128-bit data width but ECC is controlled on a 32-bit basis, the highest-order 32 bits are controlled by the W3_EC710CTL register and the lowest-order 32 bits by the W0_EC710CTL register. Do not write to the control bits (bits 7 to 3) of this register while AXI access is in progress.

42.2.3 Wn_EC710TMC : ECC Test Mode Control Register n (n = 0 to 3)

Base address: SYSRAMi = 0x8104_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x004 + 0x040 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ETMA[1:0]		—	—	—	—	—	—	ECTMCE	—	—	ECTRRS	ECREOS	ECENS	ECDCS	ECREIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ECREIS	ECC Redundancy Bit Input Data Select Selects whether the ECERDB[6:0] bits value or the ECC data in RAM is to be input to the decoder. This bit is writable only when the ECTMCE bit is 1, and synchronously cleared when the ECTMCE bit becomes 0. 0: ECC data read from RAM is input to the decoder 1: The value of the ECERDB[6:0] bits is input to the decoder	R/W
1	ECDCS	ECC Decode Input Select Selects whether the Wn_EC710TED register value or the RAM output data is to be input to the decoder. This bit is writable only when the ECTMCE bit is 1, and synchronously cleared when the ECTMCE bit becomes 0. 0: RAM output data is input to the decoder 1: The value of the Wn_EC710TED register is input to the decoder	R/W
2	ECENS	ECC Encode Input Select Selects whether the input data to RAM or the Wn_EC710TED register value is to be input to the encoder. This bit is writable only when the ECTMCE bit is 1, and synchronously cleared when the ECTMCE bit becomes 0. 0: Data input to RAM is selected 1: The value of the Wn_EC710TED register is selected	R/W
3	ECREOS	ECC Redundancy Bit Output Data Select Selects whether the ECC encoder output data or the value of the ECERDB[6:0] bits in the Wn_EC710TRC register is to be output as the ECC output. This bit is writable only when the ECTMCE bit is 1, and synchronously cleared when the ECTMCE bit becomes 0. 0: The encoded result of the encoder is output 1: The value of the ECERDB[6:0] bits in the Wn_EC710TRC register is output	R/W

Bit	Symbol	Function	R/W
4	ECTRRS	RAM Read Test Mode Select If 1 is set, read value of Wn_EC710TED register is 32-bit output data from RAM and read value of the Wn_EC710TRC.ECERDB[6:0] bits is 7 redundancy bits input to RAM. This bit is writable only when the ECTMCE bit is 1, and synchronously cleared when the ECTMCE bit becomes 0. 0: Read value is the written value for the Wn_EC710TED register and the Wn_EC710TRC.ECERDB[6:0] bits 1: Read value of Wn_EC710TED register is 32-bit output data from RAM and read value of the Wn_EC710TRC.ECERDB[6:0] bits is 7 redundancy bits input to RAM	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	ECTMCE	Test Mode Enable Enables or disables access to the test register and the test control register. When writing to this bit, it is necessary to set bits 15 and 14 of this register to 10b. If bits 15 and 14 are not 10b, writing to this bit is ignored. 0: Disables access to the test mode register and its bits 1: Enables access to the test mode register and its bits	R/W
13:8	—	These bits are read as 0. The write value should be 0.	R/W
15:14	ETMA[1:0]	ECTMCE Write Enable When these bits are 10b, writing to the ECTMCE bit is enabled. Any settings other than 10b disables writing to the ECTMCE bit. These bits are always read as 00b. 1 0: Writing is enabled Others: Writing is disabled	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The Wn_EC710TMC register is used to switch over to and control the ECC function test mode. Do not write to this register while AXI access is in progress.

42.2.4 Wn_EC710TRC : ECC Redundancy Bit Data Control Test Register n (n = 0 to 3)

Base address: SYSRAMi = 0x8104_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x008 + 0x040 × n

Bit position:	31	30						24	22						16	14						8	6						0
Bit field:	—	ECSYND[6:0]							—	ECHORD[6:0]							—	ECECRD[6:0]							—	ECERDB[6:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	ECERDB[6:0]	ECC Redundancy Bit Input/Output Substitute Buffer Register Handling of this register differs according to the setting of the associated register bits as follows: • Wn_EC710TMC.ECTMCE = 1: This register holds the substitute data for use as the value for the redundancy bit. • Wn_EC710TMC.ECREOS = 1: The value in this register substitutes for the redundancy bit value from the encoder and is written to the RAM. • Wn_EC710TMC.ECREIS = 1: The value in this register substitutes for the redundancy bit value from the RAM and is input to the decoder. • Wn_EC710TMC.ECTRRS = 1: The value read from this register is the input data to be supplied to the RAM. The value written in this register is not read.	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
14:8	ECECRD[6:0]	ECC Encode Test Register When Wn_EC710TMC.ECTMCE = 1, the redundancy bit value generated in the encoder circuit is read from this register. Note that the read value is that from the encoder as is, for example, the encoded result of the data which was input to the encoder circuit. On the other hand, the value to be written to the RAM is the data output from the encoder circuit with its lower-order 2 bits inverted.	R
15	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
22:16	ECHORD[6:0]	ECC 7-Redundancy-Bit Data Retain Test Register When Wn_EC710TMC.ECTMCE = 1, 7 redundancy bits read from the RAM are stored in this register at the next clock cycle for the RAM read access. In addition, ECC 7-Redundancy-Bit Data is stored in the next clock cycle when Wn_EC710TED register is read if Wn_EC710TMC.ECTRRS = 1.	R
23	—	This bit is read as 0. The write value should be 0.	R/W
30:24	ECSYND[6:0]	ECC Decode Syndrome Register When Wn_EC710TMC.ECTMCE = 1, the syndrome code generated in the decoder circuit is read from this register. When Wn_EC710TMC.ECTMCE = 0, all 0s are read. The value of this register varies according to changes of decoder input value.	R
31	—	This bit is read as 0. The write value should be 0.	R/W

The Wn_EC710TRC register controls ECC redundancy bits. The value read from this register is valid only when Wn_EC710TMC.ECTMCE = 1, and always 0s when Wn_EC710TMC.ECTMCE = 0.

42.2.5 Wn_EC710TED : ECC Encode/Decode Input/Output Switchover Test Register n (n = 0 to 3)

Base address: SYSRAMi = 0x8104_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x00C + 0x040 × n

Bit position:	31																													0
Bit field:	ECEDB[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	ECEDB[31:0]	32-Bit Data Test Register for ECC Encode/Decode In the test mode, the data of this register is used as the input to the encoder/decoder circuit. When Wn_EC710TMC.ECTRRS = 1, the value read from this register is the value of the data output from RAM. This register is readable and writable when Wn_EC710TMC.ECTMCE = 1. When Wn_EC710TMC.ECTMCE = 0, synchronous clearing occurs.	R/W

The Wn_EC710TED register handles ECC redundancy bits.

42.2.6 Wn_EC710EADm : ECC Error Address m Register n (n = 0 to 3, m = 0 to 7)

Base address: SYSRAMi = 0x8104_0000 + 0x1000 × i (i = 0, 1)

Offset address: 0x010 + 0x040 × n + 0x004 × m

Bit position:	31															14														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ECEAD[14:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
14:0	ECEAD[14:0]	Bit Error Address Retain the address where a 1-bit or 2-bit ECC error was detected.	R
31:15	—	These bits are read as 0.	R

The Wn_EC710EADm registers are read-only registers that retain the values of addresses where ECC errors have occurred.

42.3 Operation

42.3.1 Initialization

RAM initialization function initializes the data and redundancy bit area during reset sequence automatically.

42.3.2 Access Cycle to System SRAM with ECC Function Enabled

Read

Read access cycle when the ECC function is enabled is one cycle longer than the case when the ECC function is disabled regardless of access size.

Write

8/16-bit write access cycle when the ECC function is enabled is two cycles longer than the case when the ECC function is disabled. 32/64/96/128-bit write access cycle when the ECC function is enabled is the same as the case when the ECC function is disabled.

42.3.3 ECC Error Address Calculation Method

An ECC error address is stored as a relative address in the ECC Error Address Register (Wn_EC710EADm). To convert a relative address to an absolute address, follow the calculation formula shown in Table 42.2.

Table 42.2 ECC error address calculation formula

Register	ECC error address calculation formula
W0_EC710EAD0 to W0_EC710EAD7	SYSRAMi start address + (register value × 0x10) + 0x0
W1_EC710EAD0 to W1_EC710EAD7	SYSRAMi start address + (register value × 0x10) + 0x4
W2_EC710EAD0 to W2_EC710EAD7	SYSRAMi start address + (register value × 0x10) + 0x8
W3_EC710EAD0 to W3_EC710EAD7	SYSRAMi start address + (register value × 0x10) + 0xC

Note: i = 0, 1

42.3.4 Notes on ECC Error Address Register

When 1-bit or 2-bit errors occur, up to 8 addresses where such errors have occurred are saved in the Wn_EC710EAD0 to Wn_EC710EAD7 registers in this order.

If the retained error address is detected again due to the same error cause, it is not retained this time. When the same address is detected due to a different error cause, it is saved if there is an empty register.

If a 2-bit error is detected when all the error address registers are filled with addresses of 1-bit errors, the Wn_EC710EAD7 register is overwritten with the error address where the 2-bit error occurred. At this time, the status register indicating the error flag information is not overwritten.

When storage of 8 or more error addresses is attempted, an overflow interrupt occurs. Clear the flag when the overflow interrupt occurs or every time a 1-bit error occurs.

42.3.5 Notes on ECC Error Detection

When a 1-bit or 2-bit error is detected, information of up to 8 errors can be stored in the ECC error information storage registers Wn_EC710CTL.ECEDFm[1:0] (m = 0 to 7), Wn_EC710CTL.ECOVFF, Wn_EC710CTL.ECER2F, Wn_EC710CTL.ECER1F, and Wn_EC710EADm (m = 0 to 7) in the ascending order of register numbers.

Note that operation differs depending on the storage state of ECC error information.

Note:
1. The ECC error information storage state is determined by every 32 bits. When an ECC error occurs in the least significant 32 bits, ECC error storage state is determined by the W0_EC710CTL register bits and the W0_EC710EADm register. When an ECC error occurs in the most significant 32 bits, ECC error storage state is determined by the W3_EC710CTL register bits and the W3_EC710EADm register.
2. When a 2-bit error is detected, immediately conduct the error processing and operate the ECER2C and ECER1C registers to clear error flags and the registers related to error address.

3. When storage of 8 or more error addresses is attempted, an overflow interrupt occurs. When an overflow interrupt occurs, immediately conduct the error processing and operate the ECER2C and ECER1C registers to clear error flags and the registers related to error address.

(1) When there are empty ECC error information storage registers

The error type and error address are stored in an empty register with the smallest number. At this time, the ECC error occurred causes the 1-bit ECC error detection/correction flag (ECER1F) or the 2-bit ECC error detection flag (ECER2F) to change from 0 to 1, and an interrupt signal to be output if the corresponding Interrupt Control register (EC1EDIC or EC2EDIC) enables interrupt output.

Error type:	Error type is stored in the ECC Error Address Capture flag (ECEDFm[1:0])
Error address:	ECC error address is stored in the ECC Error Address register (Wn_EC710EADm)
Error detection flag:	When 1-bit ECC error is detected: 1-bit ECC Error Detection/Correction flag (ECER1F) is set to 1. When 2-bit ECC error is detected: 2-bit ECC Error Detection flag (ECER2F) is set to 1.

Note: n = 0 to 3, m = 0 to 7.

(2) When there are no empty ECC error information storage registers (when 1-bit ECC error is detected)

When a 1-bit ECC error is detected, only the ECC Error Address Capture Overflow flag (ECOVFF) is set to 1 in the ECC Error Information Storage register.

At this time, the 1-bit ECC error occurred causes an ECC error address capture overflow interrupt to be generated.

Error type:	Error type is not stored in the ECC Error Address Capture flag (ECEDFm[1:0])
Error address:	ECC error address is not stored in the ECC Error Address register (Wn_EC710EADm)
Error detection flag:	ECC Error Address Capture Overflow flag (ECOVFF) is set to 1

Note: n = 0 to 3, m = 0 to 7.

(3) When there are no empty ECC error information storage registers (when 2-bit ECC error is detected)

When a 2-bit ECC error is detected, operation depends on the state of the 2-bit Error Detection flag (ECER2F) at the detection time.

When ECER2F = 0 on detection of a 2-bit ECC error

If the 2-bit ECC Error Detection flag (ECER2F) is 0 when the error is detected, all ECC error information storage registers are full of 1-bit ECC errors. In this case, only the error address is overwritten in the ECC Error Address register (Wn_EC710EADm).

At this time, the 2-bit ECC error occurred causes the following to occur:

- ECC error capture overflow interrupt
- 2-bit ECC error detection interrupt (EC2EDIC = 1: Interrupt output enabled)

Error type:	Error type is not stored in the ECC Error Address Capture flag (ECEDFm[1:0])
Error address:	2-bit ECC error address is overwritten in the ECC Error Address register (Wn_EC710EADm)
Error detection flag:	2-bit ECC Error Detection flag (ECER2F) is set to 1. ECC Error Address Capture Overflow flag (ECOVFF) is set to 1.

When ECER2F = 1 on detection of a 2-bit ECC error

If the 2-bit Error Detection flag (ECER2F) is 1 when the error is detected, only the ECC Error Address Capture Overflow flag (ECOVFF) is set to 1 in the ECC Error Information Storage register.

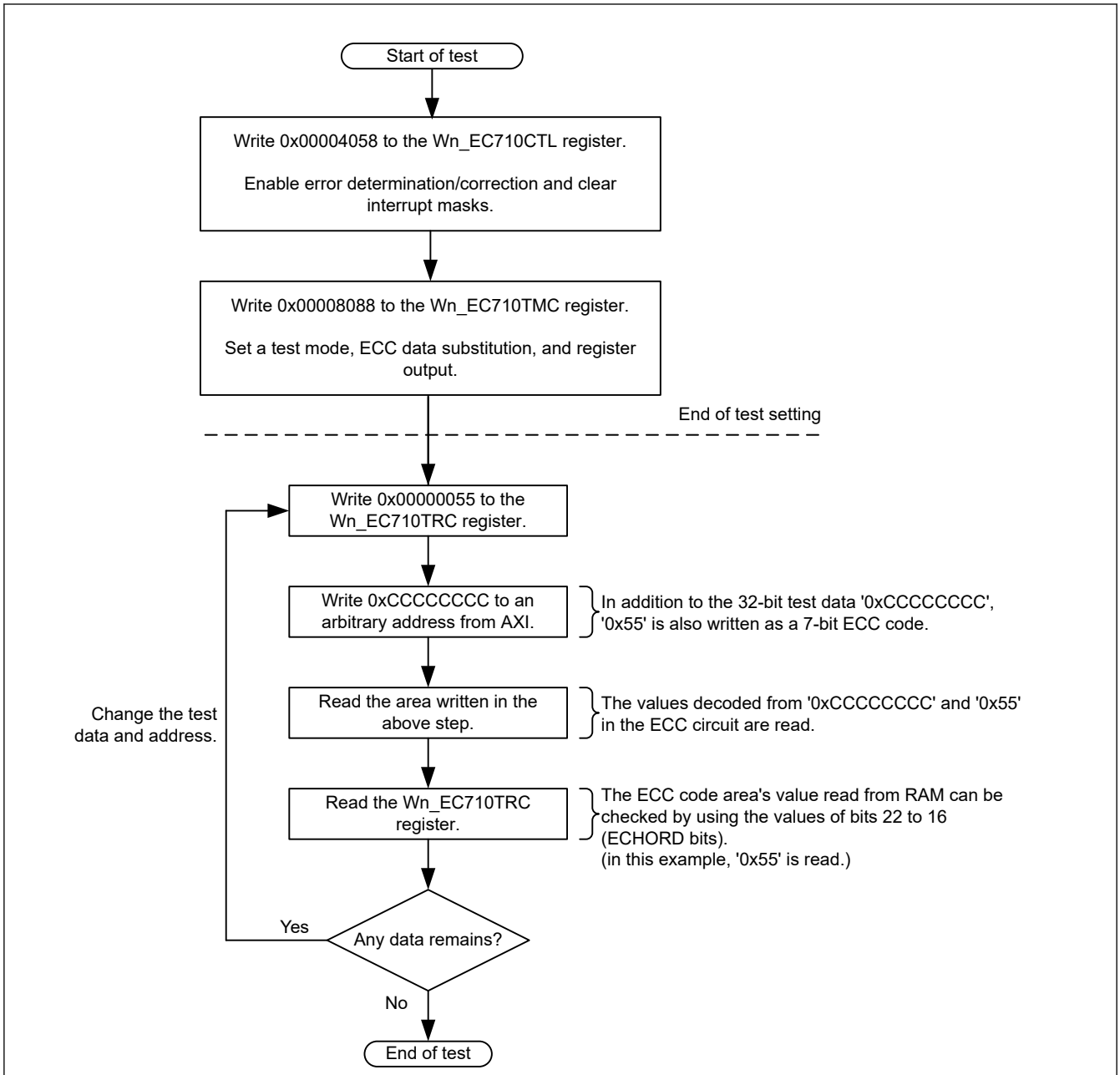
At this time, the 2-bit ECC error occurred causes an ECC error capture overflow interrupt to occur.

Error type:	Error type is not stored in the ECC Error Address Capture flag (ECEDFm[1:0])
Error address:	2-bit ECC error address is not stored in the ECC Error Address register (Wn_EC710EADm)
Error detection flag:	ECC Error Address Capture Overflow flag (ECOVFF) is set to 1

Note: n = 0 to 3, m = 0 to 7.

42.3.6 Test Function (Error Injection)

System SRAM has the error injection function, a facility to test the ECC function. The ECC errors intentionally generated by using the test registers are written to the target RAM area for checking the ECC function without involving the ECC circuit. Figure 42.2 shows an example of test flowchart.



Note: The value of the Wn_EC710CTL register is an example. Use an appropriate value according to the test conditions and purpose.

Figure 42.2 Flowchart example of ECC error injection test

42.3.7 Interrupt

When ECC error occurs, interrupt can be output. Table 42.3 lists the System SRAM interrupt sources. These interrupts are combined to peripheral error interrupts (PERI_ERR0 and PERI_ERR1) as error events with other error interrupts.

Table 42.3 System SRAM interrupt sources

Name	Interrupt sources	CPU0 GIC request	DMAC activation
SRAMn_IE1	System SRAMn ECC 1 bit error	Possible*1	Not possible
SRAMn_IE2	System SRAMn ECC 2 bits error	Possible*1	Not possible
SRAMn_OVF	System SRAMn ECC error address capture overflow	Possible*1	Not possible

Note: n = 0, 1
Note 1. These are not directly connected to CPU GIC but through peripheral error interrupts (PERI_ERR0 and PERI_ERR1).

42.4 Usage Notes

42.4.1 Release of Register Write Protection

SYSRAM_CTRLn registers are write protected. These registers are used to release write protection, use the following steps.

1. Write 0x0000A508 to PRCRS register (release write protection by PRC3 bit).
2. Write value to protection register.
3. Write 0x0000A500 to PRCRS register (set write protection by PRC3 bit).

42.4.2 Note on Setting the Register

All bus masters must not access System SRAM while a register of this module is being set.

42.4.3 Access in Bit Size Units

This module performs ECC determination process on a 32-bit basis. If access is made on an 8/16-bit basis, the module performs the following operations:

8/16-bit reading

1. After reading 32 bits of data, this module performs ECC determination.
2. To the read requesting bus master, this module returns 8/16-bit data.

8/16-bit writing

1. This module reads 32 bits of data from the write-target address and performs ECC determination.
2. On the read-out 32 bits, this module overwrites new 8/16-bit data.
3. After attaching ECC data to the write-target 32-bit data of step 2., this module writes them to the System SRAM.

43. One-Time Programmable Memory (OTP)

43.1 Overview

This LSI has one-time programmable memory (OTP). Various information is stored or can be additionally stored in OTP. Data written in OTP can't be overwritten. OTP is shared with product configuration parameters which cannot be accessed by user. Then, all the areas are not accessible from CPU.

Table 43.1 lists the OTP specifications and Figure 43.1 shows a block diagram of the OTP.

Note: For WS1, One-Time Programmable Memory is not supported.

Table 43.1 OTP Specifications

Item	Description
Number of units	1 unit
Functionality	• Bit width: 16-bit • Word size: 512-word • Register I/F to read or write OTP • WORDLOCK and non WORDLOCK area available – WORDLOCK area protects change of unwritten bit. – Non WORDLOCK area allows additional write to unwritten bit. • Overwrite protection • Redundancy support • Power down mode support
Available information (Non-security support product)	• Unique ID • JTAG authentication ID and setting • SCI/USB boot setting • Anti-rollback counter • Part number • Product Version • SHOSTIF boot mode setting • User area

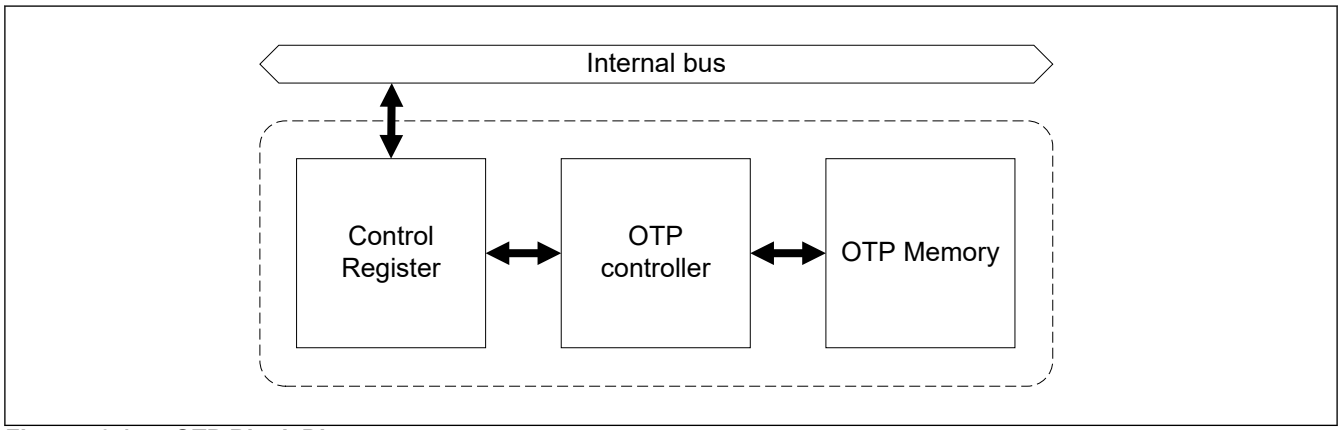


Figure 43.1 OTP Block Diagram

43.2 Register Descriptions

43.2.1 OTPPWR : OTP Power Control Register

Base address: OTP = 0x8102_8000

Offset address: 0x00

Bit position:	31																												4			0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ACCL	—	—	—	PWR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	PWR	OTP power on/off setting 0: Power off 1: Power on	R/W
3:1	—	These bits are read as 0. The write value should be 0.	R/W
4	ACCL	Selects OTP access I/F 0: Not selected 1: Selected	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

OTPPWR register controls the power of OTP and provides a valid access path.

If 0 is written to PWR bit during write access to OTP, OTP powers off after write access is complete. PWR bit cannot be changed during read access to OTP.

Change ACCL bit when OTP access is stopped (CNT_ST_IDLE bit in OTPSTR register is 1).

When OTP access is requested while ACCL bit is 0, the access request is ignored and ERR_RP bit in OTPSTR register is asserted.

43.2.2 OTPSTR : OTP Access Status Register

Base address: OTP = 0x8102_8000

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CNT_ST_IDLE	—	—	—	—	—	ERR_RDY_RD	ERR_RDY_WR	—	—	—	ERR_RP	ERR_WP	ERR_WR[1:0]		CMD_RDY
Value after reset:	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	CMD_RDY	Indicates whether OTP controller is ready to receive command or not. 0: Not ready 1: Ready	R
2:1	ERR_WR[1:0]	OTP write status 0 0: PASS 0 1: Fail due to overwrites 1 0: Fail due to compare mismatch 1 1: Fail due to overwrites and compare mismatch	R
3	ERR_WP	Write protection error This bit is set to 1 when OTP write access is issued to the write prohibited area. Cleared at the start of the next OTP write command.	R
4	ERR_RP	Read protection error This bit is set to 1 when OTP read access is issued to the read prohibited area. Cleared at the start of the next OTP read command.	R
7:5	—	These bits are read as 0. The write value should be 0.	R/W
8	ERR_RDY_WR	OTP write command ready error This bit is set to 1 when OTP write access is issued during CMD_RDY = 0. Cleared by writing 0 on this bit. Writing 1 is ignored.	R/W
9	ERR_RDY_RD	OTP read command ready error This bit is set to 1 when OTP read access is issued during CMD_RDY = 0. Cleared by writing 0 on this bit. Writing 1 is ignored.	R/W

Bit	Symbol	Function	R/W
14:10	—	These bits are read as 0. The write value should be 0.	R/W
15	CNT_ST_IDLE	Indicates status of OTP controller 0: Busy 1: Idle	R
31:16	—	These bits are read as 0. The write value should be 0.	R/W

OTPSTR register indicates status of command ready and error.

43.2.3 OTPSTAWR : OTP Write Command Register

Base address: OTP = 0x8102_8000

Offset address: 0x08

Bit position:	31																													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STAWR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	STAWR	OTP write start By writing 1 to this bit, the OTP write command is issued and writing data to OTP is started. This bit is automatically cleared after the OTP write command is completed. Only 1 can be written to this bit. Writing 0 is ignored.	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

OTPSTAWR register is used to issue the OTP write command.

If OTPSTAWR register is set to 1 while write prohibited area is set in the OTPADRWR register, the STAWR bit does not change to 1 and no OTP write access occurs. In this case, ERR_WP bit in OTPSTR register is set to 1.

43.2.4 OTPADRWR : OTP Write Address Register

Base address: OTP = 0x8102_8000

Offset address: 0x0C

Bit position:	31																							8						0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADRWR[8:0]						
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	ADRWR[8:0]	OTP write address Set the OTP address where data is written. The address to be set is OTP address in a halfword (16-bit) unit.	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

OTPADRWR register is used to set the write address to OTP.

43.2.5 OTPDATAWR : OTP Write Data Register

Base address: OTP = 0x8102_8000

Offset address: 0x10

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DATAWR[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DATAWR[15:0]	OTP write data Set the data to be written to the OTP.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

OTPDATAWR register is used to set the write data to OTP.

43.2.6 OTPADRRD : OTP Read Address Register

Base address: OTP = 0x8102_8000

Offset address: 0x14

Bit position:	31																				8										0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ADRRD[8:0]									
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
8:0	ADRRD[8:0]	OTP read address Set the OTP address where data is read. The address to be set is OTP address in a halfword (16-bit) unit. After reading data from the OTP, the address set in this register is automatically incremented. By reading this register, the incremented address can be read. However, if read error (ERR_RP or ERR_RDY_RD in OTPSTR register) occurs, it will not be incremented. Set a different address.	R/W
31:9	—	These bits are read as 0. The write value should be 0.	R/W

OTPADRRD register is used to set the read addresses to OTP.

43.2.7 OTPDATARD : OTP Read Data Register

Base address: OTP = 0x8102_8000

Offset address: 0x18

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DATARD[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DATARD[15:0]	OTP read data OTP data in address specified by the OTPADRRD register can be read.	R
31:16	—	These bits are read as 0.	R

OTPDATARD register stores read data from OTP.

OTP read access is not performed when the OTPDATARD register is read while ACCL bit in OTPPWR register is 0 or read prohibited area is set in the OTPADRRD register. In this case, the ERR_RP bit of the OTPSTR register is set and the read value from this register is 0.

43.3 OTP Memory Map

Data width of OTP is 16-bit, and its address is not memory mapped. Dedicated address (0x000 to 0x1FF) is defined to access OTP. Table 43.2 shows OTP memory map in non-security support product. Various information is stored or can be stored in OTP. Address not listed in Table 43.2 is reserved and do not access reserved area.

In OTP, unwritten bit is 0 and written bit is 1. There are two kinds of address attribution, WORDLOCK applied address and WORDLOCK not-applied address.

- WORDLOCK applied address

Only one time write of 16-bit data including 0 and 1 is permitted. Once writing to the address, not only written bits but also unwritten bits can't be written again. That means the address is read-only. All 16-bit 0 write is also supported. After writing, write to the address is impossible longer.

- WORDLOCK not-applied address

Write 1 to the unwritten bit is possible even if the write address is the same. Write can be divided to multiple times as long as the write bit is unwritten yet. Set 0 to the bit which is not needed to write.

Table 43.2 OTP memory map in non-security support product

Area	Bit width	Start address	End address	WORDLOCK	Initial property
Unique ID	64	0x089	0x08C	—	R
JTAG authentication mode	3	0x091	0x091	No	R/W
JTAG authentication level 1 plaintext ID	128	0x093	0x09A	Yes	W
JTAG authentication level 2 plaintext ID	128	0x09B	0x0A2	Yes	W
SCI/USB boot mode disable	2	0x18D	0x18D	No	R/W
Anti-rollback counter	320	0x159	0x16C	No	R/W
Part number	16	0x1A7	0x1A7	—	R
Product Version	16	0x1A9	0x1A9	—	R
SHOSTIF boot mode setting	16 × 4	0x1B5	0x1B8	Yes	R/W
User area	368	0x1B9	0x1CF	Yes	R/W

43.3.1 Unique ID (UID)

Indicates 64-bit unique ID. This area is read-only.

Address(es): 0x089 to 0x08C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ID															
Value after reset:	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1

Bit	Symbol	Function	R/W
15:0	ID	64-bit unique ID 0x089: Unique ID [15:0] 0x08A: Unique ID [31:16] 0x08B: Unique ID [47:32] 0x08C: Unique ID [63:48]	R

Note 1. Value after reset depends on the product variant.

43.3.2 JTAG Authentication Mode (AUTHMODEJ)

Specifies how to authenticate JTAG connection.

Address(es): 0x091

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	MODE[2:0]		
Value before write:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	MODE[2:0]	Specifies JTAG authentication mode 0 0 0: No authentication 0 0 1: Authentication level 1 (Both EL2 and EL1/EL0 access permission) 0 1 x: Authentication level 2 (Only EL1/EL0 access permission) 1 x x: Permanent prohibition of JTAG connection	R/W
15:3	—	Reserved	—

43.3.3 JTAG Authentication Level 1 Plaintext ID (JID1P)

Specifies plaintext ID for JTAG authentication level 1. This area is write-only. Read value is 0. ID must include at least one '1' programmed bit each 32-bit. Since WORDLOCK is applied for this area, one time write of 16-bit data is needed each address.

Address(es): 0x093 to 0x09A

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ID															
Value before write:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ID	128-bit JTAG authentication level 1 plaintext ID 0x093: ID [15:0] 0x094: ID [31:16] 0x095: ID [47:32] 0x096: ID [63:48] 0x097: ID [79:64] 0x098: ID [95:80] 0x099: ID [111:96] 0x09A: ID [127:112]	W

43.3.4 JTAG Authentication Level 2 Plaintext ID (JID2P)

Specifies plaintext ID for JTAG authentication level 2. This area is write-only. Read value is 0. ID must include at least one '1' programmed bit each 32-bit. Since WORDLOCK is applied for this area, one time write of 16-bit data is needed each address.

Address(es): 0x09B to 0x0A2

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	ID															
Value before write:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ID	128-bit JTAG authentication level 2 plaintext ID 0x09B: ID [15:0] 0x09C: ID [31:16] 0x09D: ID [47:32] 0x09E: ID [63:48] 0x09F: ID [79:64] 0x0A0: ID [95:80] 0x0A1: ID [111:96] 0x0A2: ID [127:112]	W

43.3.5 SCI/USB Boot Mode Disable (SUDIS)

Specifies whether SCI/USB boot enable or disable.

Address(es): 0x18D

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	DIS	—
Value before write:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	Reserved	—
1	DIS	Specifies SCI/USB boot disable 0: SCI/USB boot enable 1: SCI/USB boot disable. Permanent prohibition of connection.	R/W
15:2	—	Reserved	—

43.3.6 Anti-rollback Counter (ARBC)

This area is used as anti-rollback counter. User can freely use this area to control version of their application. For example, five 64-bit counter for version control of five program modules at firmware update.

Address(es): 0x159 to 0x16C

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	BIT15	BIT14	BIT13	BIT12	BIT11	BIT10	BIT9	BIT8	BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
Value before write:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	BIT15 to BIT0	Uses as version control	R/W

43.3.7 Part Number (PN)

Indicates part number of the product. This area is read-only.

Address(es): 0x1A7

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	PN															
Value after reset:	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1

Bit	Symbol	Function	R/W
15:0	PN	Part number of the product 0x1000 R9A07G074M08GBG : 0x1001 R9A07G074M05GBG : 0x1010 R9A07G074M04GBG : 0x1011: R9A07G074M01GBG	R

Note 1. Depends on part number of the product

43.3.8 Product Version (VER)

Indicates product version. This area is read-only.

Address(es): 0x1A9

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	VER															
Value after reset:	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1	*1

Bit	Symbol	Function	R/W
15:0	VER	Product version	R

Note 1. Depends on version of the product

43.3.9 SHOSTIF Boot Mode Setting (SHBMS)

Specifies values set in SHOSTIF registers (CTRLR0 and SHCFG registers) when SHOSTIF boot mode.is selected. Four addresses (0x1B5 to 0x1B8) are available. If VALID bit of either address is 1, setting of the valid address is applied. If VALID bits of multiple addresses are 1, setting of the highest address in the valid addresses is applied. Since WORDLOCK is applied for this area, one time write of 16-bit data is needed each address.

Address(es): 0x1B5 to 0x1B8

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	VALID	—	—	—	—	—	—	—	—	ADDRESSING	BYTESWAP	SPIMODE[1:0]		—	SCPOL	SCPH
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SCPH	Specifies value of SCPH bit in CTRLR0 register.	R/W
1	SCPOL	Specifies value of SCPOL bit in CTRLR0 register.	R/W
2	—	Reserved	—

Bit	Symbol	Function	R/W
4:3	SPIMODE[1:0]	Specifies value of SPIMODE[1:0] bits in SHCFG register.	R/W
5	BYTESWAP	Specifies value of BYTESWAP bit in SHCFG register.	R/W
6	ADDRESSING	Specifies value of ADDRESSING bit in SHCFG register.	R/W
14:7	—	Reserved	—
15	VALID	Valid flag Writes 1 to indicate setting values of this area is valid.	R/W

43.3.10 User Area (USERAREA)

This area can be used by user freely. Since WORDLOCK is applied for this area, one time write of 16-bit data is needed each address.

Address(es): 0x1B9 to 0x1CF

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AREA[15:0]															
Value before write:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	AREA[15:0]	Free area which user can use.	R/W

43.4 Operation

43.4.1 Write procedure

Table 43.3 Write procedure

Steps	What to do
A	It is assumed that OTP is in a power-down state. Confirm completion of the power-down process by checking that the PWR bit in OTPPWR register is 0 and the CMD_RDY bit in OTPSTR register is 0.
B	Set the PWR and ACCL bits in OTPPWR register to 1. Confirm if CMD_RDY bit in OTPSTR register is 1 after setting.
C	Set the write address to the OTPADRWR register.
D	Set the data to be written to the OTPDATAWR register.
E	Set 1 to the STAWR bit in OTPSTAWR register.
F	OTP controller starts the OTP write process. CMD_RDY bit changes from 1 to 0. If setting the write-prohibited area to OTPADRWR register, the STAWR bit is not set to 1 and the CMD_RDY bit keeps 1. In this case, ERR_WP bit in the OTPSTR register is set to 1.
G	OTP controller writes the value of OTPDATAWR register to OTP. Note) The values of OTPADRWR and OTPDATAWR registers must be retained until this procedure is completed.
H	Poll the STAWR bit until changing to 0 in order to detect the completion of the write command acceptance.
I	Poll the CMD_RDY bit until changing to 1 in order to detect the completion of the write.
J	Once detecting the CMD_RDY bit is 1, confirm that the ERR_WR bit in the OTPSTR register is not set to 1. Note) If set to 1, OTP write error occurs.
K	When continuing OTP write, repeat step C) and later. When issuing OTP read, refer to step C) and later of Table 43.4. When completing OTP access, confirm stop of the OTP access (CMD_RDY = 1). Then set PWR and ACCL bits in the OTPPWR registers to 0, and wait until CMD_RDY bit is set to 0. Note) Be sure to power down by this procedure if no more access is needed.

Note: WORDLOCK is applied for some areas. The OTP write to the WORDLOCK applied area cannot be divided to multiple times. Set the data to be written to all the bits of OTPDATAWR register only once.
For written bits, the bits retain 1 no matter whether they are written with 0 or 1 again.
Reset while writing OTP is prohibited. That operation is not guaranteed.

43.4.2 Read procedure

Table 43.4 Read procedure

Steps	What to do
A	It is assumed that OTP is in a power-down state. Confirm completion of the power-down process by checking that the PWR bit in OTPPWR register is 0 and the CMD_RDY bit in OTPSTR register is 0.
B	Set the PWR and ACCL bits in OTPPWR register to 1. Confirm if CMD_RDY bit in OTPSTR register is 1 after setting.
C	Set the read address to the OTPADRRD register.
D	Read the OTPDATARD register.
E	OTP controller starts the OTP read process. CMD_RDY bit changes from 1 to 0. If setting the read-prohibited area to OTPADRRD register, OTP read process is not performed and the CMD_RDY bit keeps 1. In this case, ERR_RP bit of the OTPSTR register is set to 1.
F	OTP controller reads the data of address set in the OTPADRRD register from OTP. After completion of the process, CPU can receive read data. CMD_RDY bit changes from 0 to 1. Note) Since CPU is waited until completion of the process, polling of CMD_RDY bit is not needed.
G	When continuing OTP read, repeat step C) and later. ● When reading a continuous address (continuous read), repeat step D) and later. The OTPADRRD register is incremented automatically, so no need to change. ● When reading a different address from a continuous address, repeat step C) and later. When issuing OTP write, refer to step C) and later of Table 43.3. When completing OTP access, confirm stop of the OTP access (CMD_RDY = 1). Then set PWR and ACCL bits in the OTPPWR registers to 0, and wait until CMD_RDY bit is set to 0. Note) Be sure to power down by this procedure if no more access is needed.

Note: Continuous read is valid for readable areas. If the read address is changed to the read-prohibited area, OTP read process is not performed and the CMD_RDY bit keeps 1. ERR_RP bit of the OTPSTR register is set to 1.

44. Serial Host Interface (SHOSTIF)

44.1 Overview

This LSI has a serial host interface to connect an external host CPU (SPI master) with up to eight input/output pins. The serial host interface allows an external SPI master to access the internal resources (mainly System SRAM) of this LSI directly. Additionally, the booting of this LSI from the external SPI master is possible.

Figure 44.1 shows a block diagram of the SHOSTIF and Table 44.1 lists the serial host interface specifications.

Table 44.1 SHOSTIF specifications

Parameter	Description
Number of units	1 unit
Functionality	• Serial communication is possible in slave mode • Supported interface: – Motorola Serial Peripheral Interface (4-wire SPI) – Enhanced SPI modes with Dual, Quad or Octal SPI. • Serial clock polarity switching • Serial clock phase switching • Transfer mode: – Single data transfer (SDR) supported – No DDR, data strobe nor data mask supported • All the SPI transfer consists of Instruction phase, Address phase and Data phase: – Command phase: 8 bits — Support read/write, transfer type, wait cycles and data length – Address phase: 24 bits — Aligned to 32-bit data boundary – Data phase: 32 bits × burst length up to 64 • Hardware protocol state machine for transfer management • 32 bits × 64 stage FIFO transmit and receive buffers • Serial clock (sclk_in): Up to 40 MHz
Interrupt source	Maskable interrupt sources. All sources are combined to one interrupt. External output pin is available: • SPI master error • AHB error • Transmit FIFO Empty • Transmit FIFO Underflow • Receive FIFO Full • Receive FIFO Overflow
Accessible area	• System SRAM • Mailbox and Semaphore registers • Serial Host Interface registers
Module-stop function	Module-stop state can be set to reduce power consumption

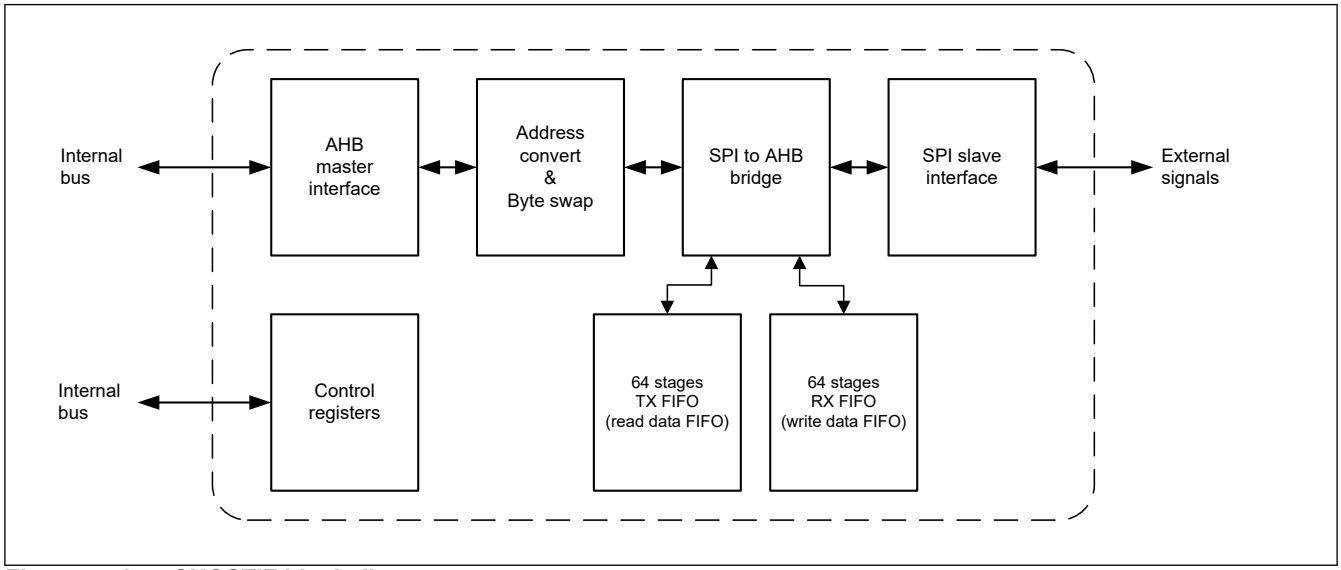


Figure 44.1 SHOSTIF block diagram

Table 44.2 lists the input/output pins of the serial host interface.

Table 44.2 SHOSTIF I/O pins

Name	I/O	Function
HSPI_CK	Input	SHOSTIF Clock input pin
HSPI_CS#	Input	SHOSTIF Chip select input pin
HSPI_IO0	I/O	SHOSTIF Data0 input/output pin
HSPI_IO1	I/O	SHOSTIF Data1 input/output pin
HSPI_IO2	I/O	SHOSTIF Data2 input/output pin
HSPI_IO3	I/O	SHOSTIF Data3 input/output pin
HSPI_IO4	I/O	SHOSTIF Data4 input/output pin
HSPI_IO5	I/O	SHOSTIF Data5 input/output pin
HSPI_IO6	I/O	SHOSTIF Data6 input/output pin
HSPI_IO7	I/O	SHOSTIF Data7 input/output pin
HSPI_INT#	Output	SHOSTIF Interrupt output pin

44.2 Register Descriptions

The register base address is different between external SPI master view and internal memory address view except for SHCFG register.

- External SPI master view (8-bit addressing mode) : SHOSTIF_E08 = 0x18_1000

  Note: When 32-bit addressing mode is used, please shift 8-bit address (base address + offset address) by 2 bits in the right direction.

- Internal memory address view: SHOSTIF = 0x8024_1000
- Internal memory address view for SHCFG: SHOSTIF_CFG = 0x8128_1920

44.2.1 CTRLR0 : Control Register 0

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x0018_1000

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SCPOL	SCPH	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	SCPH	Serial Clock Phase Valid when the frame format (FRF) is set to Motorola SPI. The serial clock phase selects the relationship of the serial clock with the slave select signal. When SCPH = 0, data is captured on the first edge of the serial clock. When SCPH = 1, the serial clock toggles for 1 cycle after the slave select line is activated, and data is captured on the second edge of the serial clock. 0: Serial clock toggles in the middle of the first bit 1: Serial clock toggles at the start of the first bit	R/W

Bit	Symbol	Function	R/W
9	SCPOL	Serial Clock Polarity Valid when the frame format (FRF) is set to Motorola SPI. Used to select the polarity of the inactive serial clock, which is held inactive when the SPI master is not actively transferring data on the serial bus. 0: Inactive state of the serial clock is low 1: Inactive state of the serial clock is high	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

The CTRLR0 register controls the serial data transfer. It is not possible to write to this register when SHOSTIF is enabled (ENR.ENABLE = 1).

44.2.2 ENR : Enable Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x0018_1000

Offset address: 0x008

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ENABLE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENABLE	SHOSTIF Enable Enables and disables all SHOSTIF operations. When disabled, all serial transfers are halted immediately. Transmit and receive FIFO buffers are cleared when the device is disabled. It is not possible to program some of the SHOSTIF control registers when enabled. When disabled, the SLEEP bit in the SHCFG register is set (after a delay) to inform the system that it is safe to remove the internal clock, thus saving power consumption in the system. 0: Disables SHOSTIF 1: Enables SHOSTIF	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The ENR register enables or disables the serial host interface.

44.2.3 RXFBTR : Receive FIFO Burst Threshold Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x0018_1000

Offset address: 0x014

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	RXFBTL[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

Bit	Symbol	Function	R/W
5:0	RXFBTL[5:0]	Receive FIFO Burst Threshold For write operations on SPI slave interface, this register is used to build a burst on AHB master interface. The data is sent in bursts equal to the level specified in this register. If the RXFBTL value is 3 then a burst length of 4 is used for every write transfer. If SHOSTIF receives a burst smaller than the programmed value, then SHOSTIF issues a smaller burst on the AHB interface and completes the transaction. In case of 8-bit addressing mode, do not change from the initial value. In case of 32-bit addressing mode, recommendation value is 0x0F.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

The RXFBTR register controls the burst threshold value for AHB interface from RX FIFO.

44.2.4 TXFTLR : Transmit FIFO Threshold Level Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x0018_1000

Offset address: 0x018

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	TFT[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	TFT[5:0]	Transmit FIFO Threshold Controls the level of entries (or below) at which the transmit FIFO controller triggers an interrupt. The FIFO depth is 64. When the number of transmit FIFO entries is less than or equal to this value, the transmit FIFO empty interrupt is triggered.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

The TXFTLR register controls the threshold value for the transmit FIFO memory.

44.2.5 RXFTLR : Receive FIFO Threshold Level Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x0018_1000

Offset address: 0x01C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	RFT[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	RFT[5:0]	Receive FIFO Threshold Controls the level of entries or above at which the receive FIFO controller triggers an interrupt. The FIFO depth is 64. When the number of receive FIFO entries is greater than or equal to this value + 1, the receive FIFO full interrupt is triggered.	R/W
31:6	—	These bits are read as 0. The write value should be 0.	R/W

The RXFTLR register controls the threshold value for the received FIFO memory.

44.2.6 SR : Status Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x0018_1000

Offset address: 0x028

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	BUSY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	BUSY	Busy Flag When set, indicates that a serial transfer is in progress. When cleared, indicates that the SHOSTIF is idle or disabled. 0: SHOSTIF is idle or disabled 1: SHOSTIF is actively transferring data	R
31:1	—	These bits are read as 0.	R

The SR register is a read-only register that indicates the current transfer status, FIFO status, and any transmission/reception errors that may have occurred. The Status register may be read at any time. None of the bits in this register request an interrupt.

44.2.7 IMR : Interrupt Mask Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x0018_1000

Offset address: 0x02C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SPIMEM	AHBEM	TXUIM	—	—	RXFIM	RXOIM	—	—	TXEIM
Value after reset:	0	0	0	0	0	0	1	1	1	0	0	0	1	0	0	1

Bit	Symbol	Function	R/W
0	TXEIM	Transmit FIFO Empty Interrupt Mask 0: txe_intr interrupt is masked 1: txe_intr interrupt is not masked	R/W

Bit	Symbol	Function	R/W
2:1	—	These bits are read as 0. The write value should be 0.	R/W
3	RXOIM	Receive FIFO Overflow Interrupt Mask 0: rxo_intr interrupt is masked 1: rxo_intr interrupt is not masked	R/W
4	RXFIM	Receive FIFO Full Interrupt Mask 0: rxf_intr interrupt is masked 1: rxf_intr interrupt is not masked	R/W
6:5	—	These bits are read as 0. The write value should be 0.	R/W
7	TXUIM	Transmit FIFO Underflow Mask 0: txu_intr interrupt is masked 1: txu_intr interrupt is not masked	R/W
8	AHBEM	AHB Error Interrupt Mask 0: ahbe_intr interrupt is masked 1: ahbe_intr interrupt is not masked	R/W
9	SPIMEM	SPI Master Error Interrupt Mask 0: spime_intr interrupt is masked 1: spime_intr interrupt is not masked	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

The IMR register masks or enables all interrupts generated by the SHOSTIF. Actual interrupt mask is determined by combination with the SHCFG register.

44.2.8 ISR : Interrupt Status Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x0018_1000

Offset address: 0x030

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SPIMES	AHBES	TXUIS	—	—	RXFIS	RXOIS	—	—	TXEIS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*1

Bit	Symbol	Function	R/W
0	TXEIS	Transmit FIFO Empty Interrupt Status 0: txe_intr interrupt is not active after masking 1: txe_intr interrupt is active after masking	R
2:1	—	These bits are read as 0.	R
3	RXOIS	Receive FIFO Overflow Interrupt Status 0: rxo_intr interrupt is not active after masking 1: rxo_intr interrupt is active after masking	R
4	RXFIS	Receive FIFO Full Interrupt Status 0: rxf_intr interrupt is not active after masking 1: rxf_intr interrupt is active after masking	R
6:5	—	These bits are read as 0.	R
7	TXUIS	Transmit FIFO Underflow Status 0: txu_intr interrupt is not active after masking 1: txu_intr interrupt is active after masking	R

Bit	Symbol	Function	R/W
8	AHBES	AHB Error Interrupt Status 0: ahbe_intr interrupt is not active after masking 1: ahbe_intr interrupt is active after masking	R
9	SPIMES	SPI Master Error Interrupt Status 0: spime_intr interrupt is not active after masking 1: spime_intr interrupt is active after masking	R
31:10	—	These bits are read as 0.	R

Note 1. Initial value depends on the setting of ENR register. The value is 0 when ENR.ENABLE = 0 and 1 when ENR.ENABLE = 1.

The ISR register reports the status of the SHOSTIF interrupts after they are masked.

44.2.9 RISR : Raw Interrupt Status Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x0018_1000

Offset address: 0x034

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	SPIMER	AHBER	TXUIR	—	—	RXFIR	RXOIR	—	—	TXEIR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	*1

Bit	Symbol	Function	R/W
0	TXEIR	Transmit FIFO Empty Raw Interrupt Status 0: txe_intr interrupt is not active prior to masking 1: txe_intr interrupt is active prior to masking	R
2:1	—	These bits are read as 0.	R
3	RXOIR	Receive FIFO Overflow Raw Interrupt Status 0: rxo_intr interrupt is not active prior to masking 1: rxo_intr interrupt is active prior to masking	R
4	RXFIR	Receive FIFO Full Raw Interrupt Status 0: rxf_intr interrupt is not active prior to masking 1: rxf_intr interrupt is active prior to masking	R
6:5	—	These bits are read as 0.	R
7	TXUIR	Transmit FIFO Underflow Raw Interrupt Status 0: txu_intr interrupt is not active prior to masking 1: txu_intr interrupt is active prior to masking	R
8	AHBER	AHB Error Raw Interrupt Status This bit indicates whether AHB master interface received an error response. This bit is cleared when the status register is read. 0: ahbe_intr interrupt is not active prior to masking 1: ahbe_intr interrupt is active prior to masking	R
9	SPIMER	SPI Master Error Raw Interrupt Status This bit is set to 1 under the following conditions: ● If SPI slave receives a command which does not fall into the command set ● If SPI slave receives an out-of-order read command (for example read data before read request in polling mode) ● If SPI master writes more data than specified in the instruction phase This bit is cleared when the status register is read. 0: spime_intr interrupt is not active prior to masking 1: spime_intr interrupt is active prior to masking	R
31:10	—	These bits are read as 0.	R

Note 1. Initial value depends on the setting of the ENR register. The value is 0 when ENR.ENABLE = 0 and 1 when ENR.ENABLE = 1.

The RISR register reports the status of the SHOSTIF interrupts before they are masked.

44.2.10 TXUICR : Transmit FIFO Underflow Interrupt Clear Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x0018_1000

Offset address: 0x038

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TXUICR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TXUICR	Clear Transmit FIFO Underflow Interrupt This register reflects the status of the interrupt. A read from this register clears the txu_intr interrupt. Writing to this register has no effect. Read value is the same as TXUIS bit in the ISR register.	R
31:1	—	These bits are read as 0.	R

The TXUICR register is used to clear the txu_intr interrupts.

44.2.11 RXOICR : Receive FIFO Overflow Interrupt Clear Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x0018_1000

Offset address: 0x03C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RXOICR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	RXOICR	Clear Receive FIFO Overflow Interrupt This register reflects the status of the interrupt. A read from this register clears the rxo_intr interrupt. Writing has no effect. Read value is the same as RXOIS bit in the ISR register.	R
31:1	—	These bits are read as 0.	R

The RXOICR register is used to clear the rxo_intr interrupts.

44.2.12 SPIMECR : SPI Master Interrupt Clear Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x0018_1000

Offset address: 0x040

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SPIMECR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SPIMECR	Clear SPI Master Error interrupt This register reflects the status of the interrupt. A read from this register clears the spime_intr interrupt. Writing to this register has no effect. Read value is the same as the SPIMES bit in the ISR register.	R
31:1	—	These bits are read as 0.	R

The SPIMECR register is used to clear the spime_intr interrupts.

44.2.13 AHBECR : AHB Error Clear Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x0018_1000

Offset address: 0x044

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	AHBECR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AHBECR	Clear AHB Error Interrupt This register reflects the status of the interrupt. A read from this register clears the ahbe_intr interrupt. Writing to this register has no effect. Read value is the same as AHBES bit in the ISR register.	R
31:1	—	These bits are read as 0.	R

The AHBECR register is used to clear the ahbe_intr interrupts.

44.2.14 ICR : Interrupt Clear Register

Base address: SHOSTIF = 0x8024_1000
SHOSTIF_E08 = 0x0018_1000

Offset address: 0x048

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ICR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ICR	Clear Interrupts This register is set if any of the specified interrupts are active. A read clears the txo_intr, rxu_intr, rxo_intr, ahbe_intr, and the mst_intr interrupts. Writing to this register has no effect. Read value is the same as logical OR of the RXOIS, TXUIS, AHBES, and SPIMES bits in the ISR register.	R
31:1	—	These bits are read as 0.	R

The ICR register is used to clear the txo_intr, rxu_intr, rxo_intr, ahbe_intr, and the mst_intr interrupts.

44.2.15 SHCFG : SHOSTIF Configuration Register

Base address: SHOSTIF_CFG = 0x8128_1920

Offset address: 0x000

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	INTMASKE[5:0]						—	—	INTMASKI[5:0]					
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	SLEEP	ADDRESSING	BYTESWAP	SPIMODE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	SPIMODE[1:0]	SPI Frame Format Select Indicates the mode in which the SPI transfer occurs. 0 0: Standard SPI mode 0 1: SPI Dual mode 1 0: SPI Quad mode 1 1: SPI Octal mode	R/W
2	BYTESWAP	Byte Swap Mode Swap data by byte order on AHB bus. 0: No byte swap. Use this setting if byte order of data input by SPI master is [31:24], [23:16], [15:8], or [7:0]. 1: Byte swap. Use this setting if byte order of data input by SPI master is [7:0], [15:8], [23:16], or [31:24].	R/W
3	ADDRESSING	Addressing Mode Specifies unit of the address input by SPI master. 0: 32-bit 1: 8-bit	R/W

Bit	Symbol	Function	R/W
4	SLEEP	SHOSTIF Enable Flag Monitor Indicates whether the internal clock can be stopped. Note that when SHOSTIF is reset, this flag does not indicate correct state. 0: Not safe to stop the internal clock 1: Safe to stop the internal clock	R
15:5	—	These bits are read as 0. The write value should be 0.	R/W
21:16	INTMASKI[5:0]	Interrupt Mask Enable for Internal Interrupt (SHOST_INT) Bits correspond to the interrupts: • INTMASKI[0]: Transmit FIFO Empty Interrupt • INTMASKI[1]: Receive FIFO Overflow Interrupt • INTMASKI[2]: Receive FIFO Full Interrupt • INTMASKI[3]: Transmit FIFO underflow Interrupt • INTMASKI[4]: AHB Master interface error Interrupt • INTMASKI[5]: SPI Master interface error Interrupt 0: No masked 1: Masked	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
29:24	INTMASKE[5:0]	Interrupt Mask Enable for External Interrupt (HSPI_INT# signal) Bits correspond to the interrupts: • INTMASKE[0]: Transmit FIFO Empty Interrupt • INTMASKE[1]: Receive FIFO Overflow Interrupt • INTMASKE[2]: Receive FIFO Full Interrupt • INTMASKE[3]: Transmit FIFO Underflow Interrupt • INTMASKE[4]: AHB Master Interface Error Interrupt • INTMASKE[5]: SPI Master Interface Error Interrupt 0: No masked 1: Masked	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

The SHCFG register sets the operation mode and monitors the status of SHOSTIF. This register is not accessible from the external SPI master.

44.3 Operations

The SHOSTIF converts all transfers from SPI to AHB transactions. The external SPI master sends address to SHOSTIF. SHOSTIF takes the address and converts it into an AHB transfer. This enables the external master to access the internal memory directly without any CPU intervention. SHOSTIF can be enabled through boot mode on the reset and load boot program from external master.

All SPI transfers have the following phases:

- Instruction
- Address
- Data

44.3.1 Instruction Phase

Each transfer from the SPI master starts with an 8-bit instruction, which determines the transfer type and process for the rest of the transfer. Table 44.3 shows the instruction field decode.

Table 44.3 Instruction field decode (1 of 2)

Bit	Bit name	Description
7	Read/Write	Defines whether the transfer is read or write. 0: Read 1: Write
6:5	Transfer Type	Defines the transfer type. 00: Read/write data 01: Read request 10: Read/write status 11: Read data with dummy

Table 44.3 Instruction field decode (2 of 2)

Bit	Bit name	Description
4:3	Wait Cycles	Defines the wait cycles for read transactions. Selects the number of wait cycles for a read transfer. This field is only applicable for read data transfers. When the transfer type is 00, the field decode is as follows: 00: 0 Wait cycles 01: 1 Wait cycle 10: Reserved 11: Reserved When the transfer type is 11, the field decode is as follows: 00: 8 Wait cycles 01: 16 Wait cycles 10: 24 Wait cycles 11: 32 Wait cycles Note: Wait cycles are not applicable for write data operation.
2:0	Data Length	Defines the data to be transmitted in the current transfer. Unit is 32 bits (4 bytes). 000: 1 001: 4 010: 8 011: 16 100: 32 101: 64 110: Prohibited 111: Prohibited

44.3.2 Address Phase

The address phase determines the address from which the data should be read or written on the AHB master interface. Address could be received in the standard, dual, quad, or octal frame format, which is defined by the SPIMODE[1:0] bits in SHCFG register. AHB interface address width is fixed to 32 bits. SPI interface expects to receive 24-bit address, which is sent to the AHB interface. Because the data is always read/written in frame of 32 bits, the address is always word-aligned. The 32-bit output address on the SPI to AHB bridge is dependent on the setting of addressing mode as shown in Table 44.4. Addressing mode is determined by the ADDRESSING bit in SHCFG register.

Table 44.4 Addressing mode

Addressing mode	Data length	Permitted SPI_ADDR*1	Permitted address granularity	SPI to AHB bridge output address
32-bit*2	1/4/8/16/32/64	0x0, 0x1, 0x2, …, 0xFF_FFFF	0x1	{000000b, {SPI_ADDR}, 00b}
8-bit	1	0x0, 0x4, 0x8, …, 0xFF_FFFC	0x4	{00000000b, {SPI_ADDR}}
	4	0x0, 0x10, 0x20, …, 0xFF_FFF0	0x10	
	8	0x0, 0x20, 0x40, …, 0xFF_FFE0	0x20	
	16	0x0, 0x40, 0x80, …, 0xFF_FFC0	0x40	
	32	0x0, 0x80, 0x100, …, 0xFF_FF80	0x80	
	64	0x0, 0x100, 0x200, …, 0xFF_FF00	0x100	

Note 1. SPI_ADDR = 24 bits address received on SPI Interface.
Note 2. Burst transfer across 1 Kbyte boundary is prohibited.

44.3.2.1 Address Conversion

The SPI master cannot access all the memory area in this LSI because the address size is 24-bit address on SPI frame. In this LSI, System SRAM, Mailbox and Semaphore, and SHOSTIF registers are accessible from the external SPI master. Address conversion circuit converts the SPI to AHB bridge output address to the internal address of System SRAM, Mailbox and Semaphore, and SHOSTIF registers. Table 44.5 shows address mapping between SPI address on SPI frame and internal address. Access to reserved area causes an AHB error interrupt.

Table 44.5 Address mapping

Module	SPI address*1	Internal address	Size
System SRAM0 (SYSRAM0)	0x00_0000 to 0x01_FFFF (32-bit) 0x00_0000 to 0x07_FFFC (8-bit)	0x1000_0000 to 0x1007_FFFC	512 KB
System SRAM1 (SYSRAM1)	0x02_0000 to 0x03_FFFF (32-bit) 0x08_0000 to 0x0F_FFFC (8-bit)	0x1008_0000 to 0x100F_FFFC	512 KB
Mailbox and Semaphore (MBXSEM)	0x06_0000 to 0x06_03FF (32-bit) 0x18_0000 to 0x18_0FFC (8-bit)	0x8024_0000 to 0x8024_0FFC	4 KB
SHOSTIF registers	0x06_0400 to 0x06_07FF (32-bit) 0x18_1000 to 0x18_1FFC (8-bit)	0x8024_1000 to 0x8024_1FFC	4 KB
Reserved	0x06_0800 to 0xFF_FFFC (32-bit) 0x18_2000 to 0xFF_FFFF (8-bit)	Error response	—

Note 1. Depends on the addressing mode.

44.3.3 Data Phase

Data is received or transmitted in frames of 32 bits. The amount of data to be transmitted or received is defined by the Data Length field of Instruction phase.

Data frame is assumed MSB first, which means the byte order is [31:24], [23:16], [15:8], or [7:0]. When the BYTESWAP bit in the SHCFG register is set to 1, the byte order of Data frame can be swapped to [7:0], [15:8], [23:16], or [31:24].

44.3.4 Bridge Operation

The SHOSTIF performs as a bridge between the SPI master and AHB subsystem. The transfers from SPI master are directly converted in AHB read/write operations. The 8-bit instruction defines the characteristics of the transfer and direction of data flow, and SPIMODE[1:0] bits in the SHCFG register define the SPI operating mode. The following sections describe the read and write operations.

44.3.5 Write Operation

44.3.5.1 Write Instruction

The Instruction phase determines the amount of data to be transmitted during write operation. The SPI frame format is derived from SPIMODE[1:0] bits in the SHCFG register. SPI slave supports the instruction opcodes as shown in Table 44.6.

Table 44.6 Instruction set for write data operations

Instruction	Instruction name	Description
0x8i*1 i = 0 to 5	WRITEn n = 1, 4, 8, 16, 32, 64	Write operation
0xC0	WRITESTATUS0	Write Data Status instruction
0xC8	WRITESTATUS1	Write Data Status instruction with 1 wait cycle

Note 1. The lower 3 bits of the instruction determine the amount of data to be fetched by AHB master. For field decode, see Table 44.3.

44.3.5.2 Write Operation

Figure 44.2 shows how SHOSTIF performs write operation.

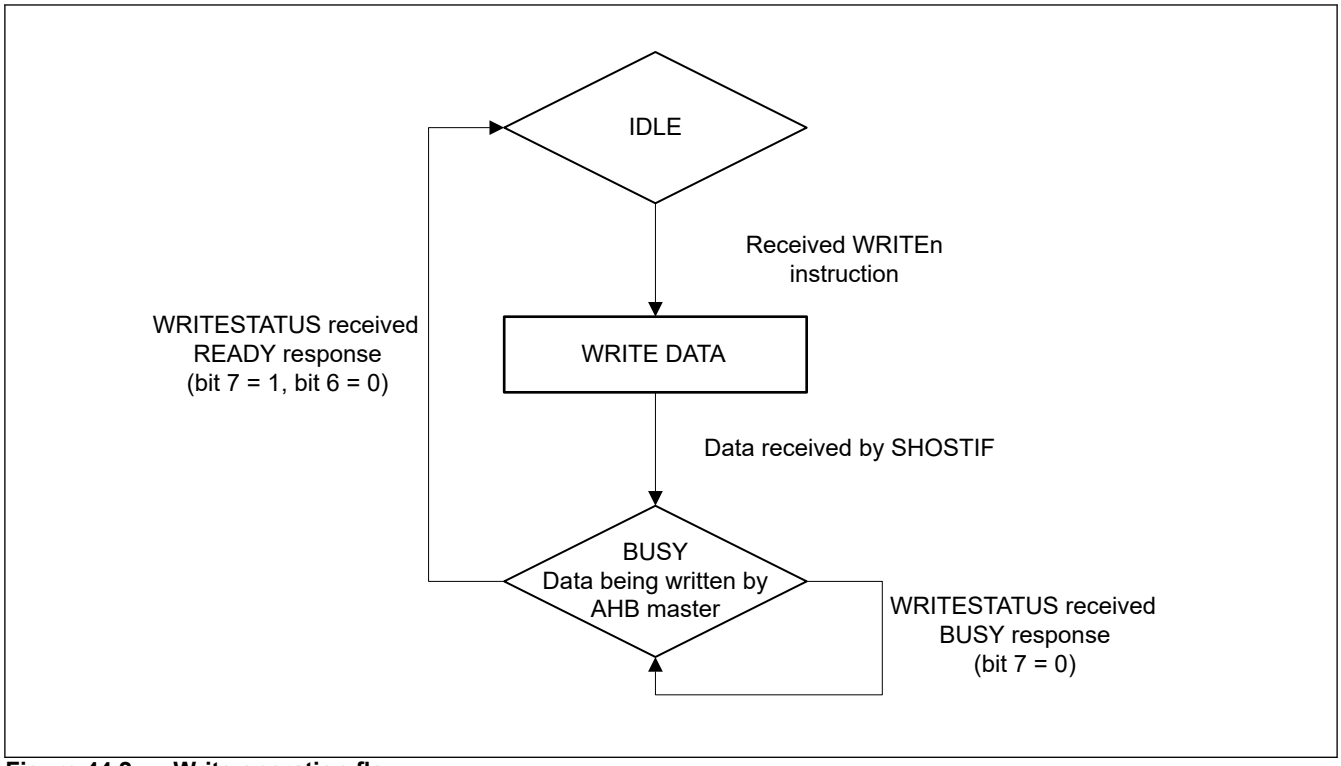


Figure 44.2 Write operation flow

(1) WRITEn (Write Data Instruction)

Data write transaction is always performed in the format specified by SPIMODE[1:0] bits in SHCFG register. The instruction, data, and address phases are sent in single, dual, quad, or octal format. Write operation from the AHB master interface to the internal slave module starts after the data of length specified in the WRITEn instruction is received in the receive FIFO.

Figure 44.3 shows the write data operation using WRITEn instruction.

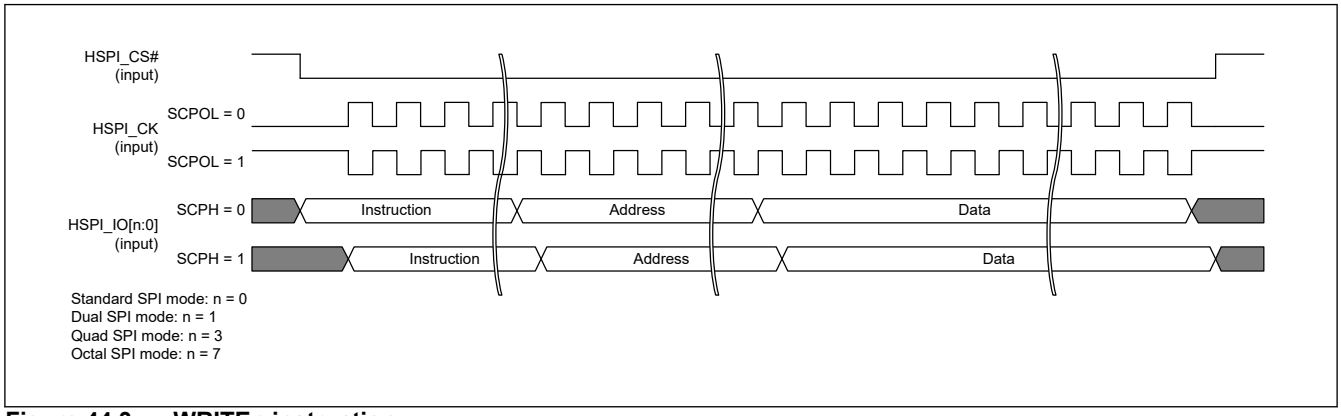


Figure 44.3 WRITEn instruction

(2) WRITESTATUSm (Write Status Instruction)

After the write operation is completed by the master, it should poll for status of the write operation using instruction opcode 0xC0 or 0xC8. Only when the master receives a READY response, it should start a new transfer.

Figure 44.4 shows the write status operation using WRITESTATUSm instruction.

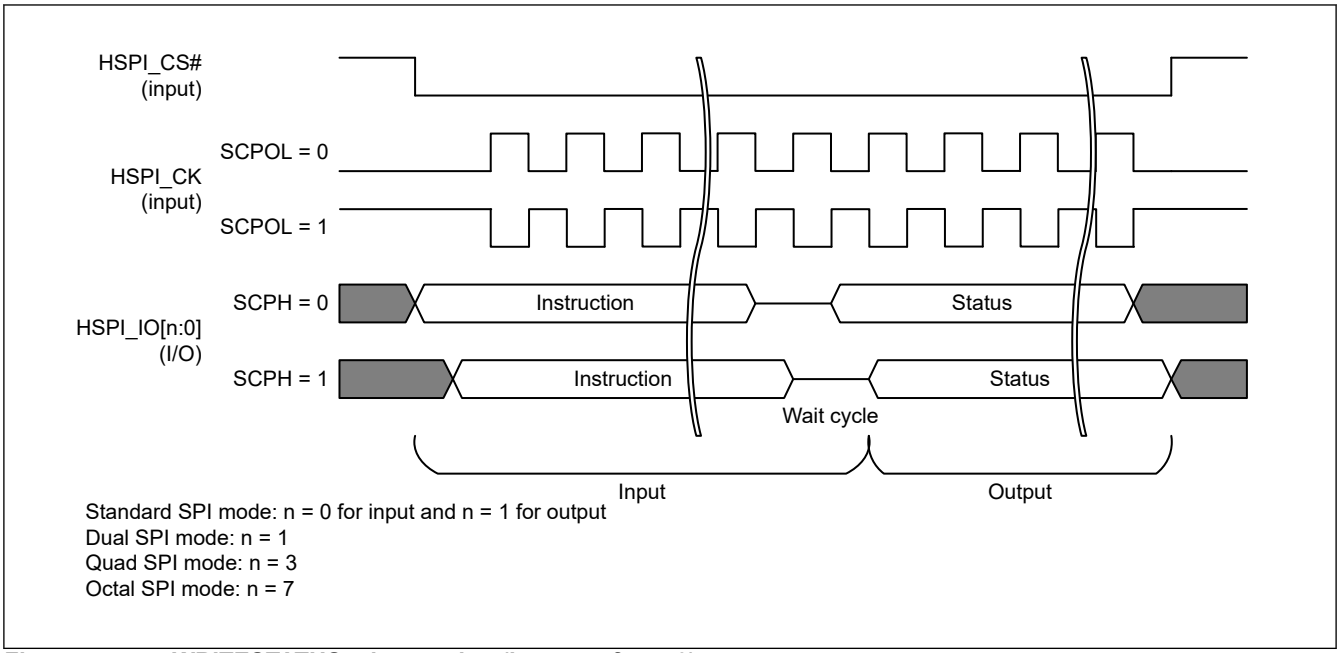


Figure 44.4 WRITESTATUSm instruction (in case of m = 0)

After the SPI master receives a READY response from SHOSTIF, it can move on to the next operation. The AHB interface may get an ERROR response for the write transfer. If the AHB interface gets an ERROR response, the WRITESTATUSm instruction indicates the same by sending a response of 0xC0. The SPI master can then issue a new write transfer or retry the same operation.

Table 44.7 Response field decode

Bit	Name	Description
7	READY	Indicates whether next operation is ready to accept or not. 0: Not ready 1: Ready
6	ERROR	Indicates whether AHB transaction ended with an ERROR response. 0: No error 1: Error
5:0	Reserved	Reserved

44.3.6 Read Operation

44.3.6.1 Read Instruction

In enhanced SPI (Dual/Quad/Octal) modes, all signals are bidirectional. Therefore, the master and slave know when to sample or drive the data on the line. The read instruction set defines the protocol under which the read transactions occur. There are two ways in which the SPI master can read data through SHOSTIF.

(1) Direct Read

In the Direct Read Instruction method, SPI Master initiates a read command and expects the data to be returned in the same command. This way the data is read in the same instruction. Be sure that the data is read on the AHB master interface within the specified cycles (DUMMY cycles).

(2) Read Data in Three Stages (this requires polling)

In some cases, the delay involved during the read operation is such that the read data availability cannot be guaranteed in terms of HSPI_CK cycles due to bus access conflict. To make sure that the data is read in the expected way during the read operation, SHOSTIF provides an alternate approach to divide the read operation into three parts:

- Read request phase
- Wait state

- Read data phase

SPI master requests the read data from SPI slave by sending only the instruction and address during the read request phase. The length of data (1, 4, 8, 16, 32, or 64) to be fetched is also encoded into the instruction. This is converted into an AHB transfer, and data is fetched and stored in the transmit FIFO. After the Read request phase, SPI master checks for availability of data by polling through the READSTATUS instruction. If SPI master receives a positive response for that instruction, then it reads the data using the READn instruction. SHOSTIF then sends all the data received from transmit FIFO to SPI master. Table 44.8 shows all the instructions applicable for read operation.

Table 44.8 Instruction set for read operations

Method	Instruction	Instruction name	Description
Direct Read	0x6i*1 i = 0 to 5	READ8n n = 1, 4, 8, 16, 32, 64	Read Data instruction with 8 wait cycles.
	0x6i*1 i = 8, 9, A, B, C, D	READ16n n = 1, 4, 8, 16, 32, 64	Read Data instruction with 16 wait cycles.
	0x7i*1 i = 0 to 5	READ24n n = 1, 4, 8, 16, 32, 64	Read Data instruction with 24 wait cycles.
	0x7i*1 i = 8, 9, A, B, C, D	READ32n n = 1, 4, 8, 16, 32, 64	Read Data instruction with 32 wait cycles.
3-Stage Read	0x2i*1 i = 0 to 5	READREQn n = 1, 4, 8, 16, 32, 64	Read request.
	0x40	READSTATUS0	Read Data ready instruction without wait cycles.
	0x48	READSTATUS1	Read Data ready instruction with 1 wait cycle.
	0x0i*1 i = 0 to 5	READ0n n = 1, 4, 8, 16, 32, 64	Read Data instruction without wait cycles.
	0x0i*1 i = 8, 9, A, B, C, D	READ1n n = 1, 4, 8, 16, 32, 64	Read Data instruction with 1 wait cycle.

Note 1. The lower 3 bits of the instruction determine the amount of data to be fetched by AHB master. For field decode, see Table 44.3.

44.3.6.2 Direct Read Operation

(1) READmn (Direct Read Data Instruction, m = 8, 16, 24, or 32)

In the direct read approach, only a single instruction is used to fetch the data using AHB interface. After SHOSTIF receives a command with more than or equal to 8 wait cycles (see Table 44.8 for details), SHOSTIF initiates a command on AHB interface to fetch the required amount of data. After data is received from AHB interface, it is transmitted to the SPI interface after the required amount of wait cycles. Figure 44.5 shows the waveform of the direct read transfer.

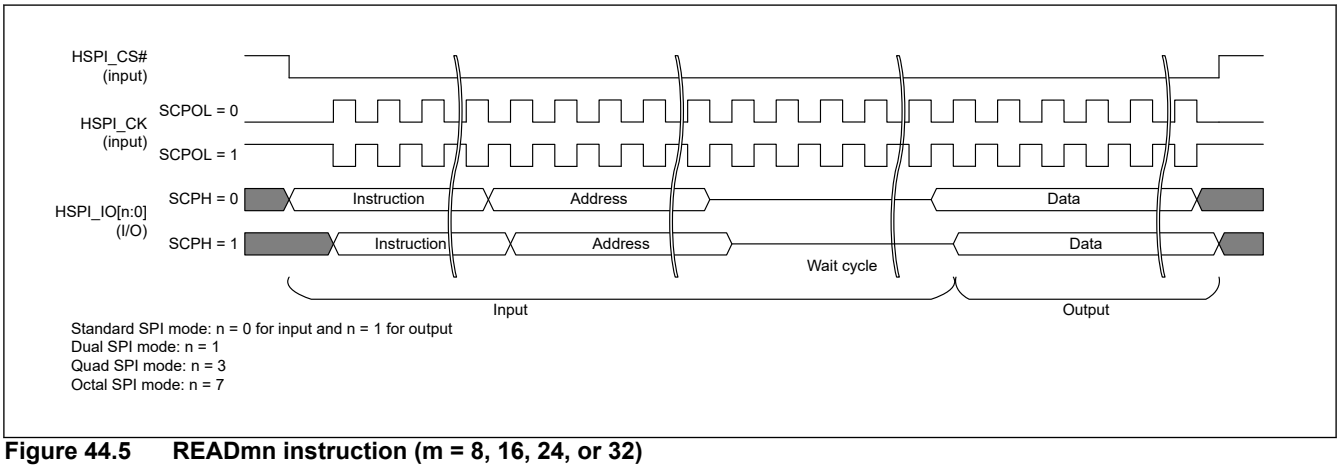


Figure 44.5 READmn instruction (m = 8, 16, 24, or 32)

If SHOSTIF is unable to fetch the data during specific wait cycles, then SHOSTIF indicates this by setting TXE bit in the Status Register (SR) and the Transmit FIFO Underflow Interrupt is asserted.

44.3.6.3 Three Stage Read Operation

When SHOSTIF is idle (the receive/transmit FIFO is empty), and a READREQn instruction is received, it is treated as a read request, and it starts fetching the data from AHB master interface. After the data is available, the READmn instruction returns the read data to SPI master. The SPI master ensures that the data is available before issuing another READmn instruction by polling for data availability. The AHB interface may receive an ERROR response for the read transfer and the READSTATUS instruction indicates this in the response while the SPI master performs the required actions.

Figure 44.6 shows the flow for read operation.

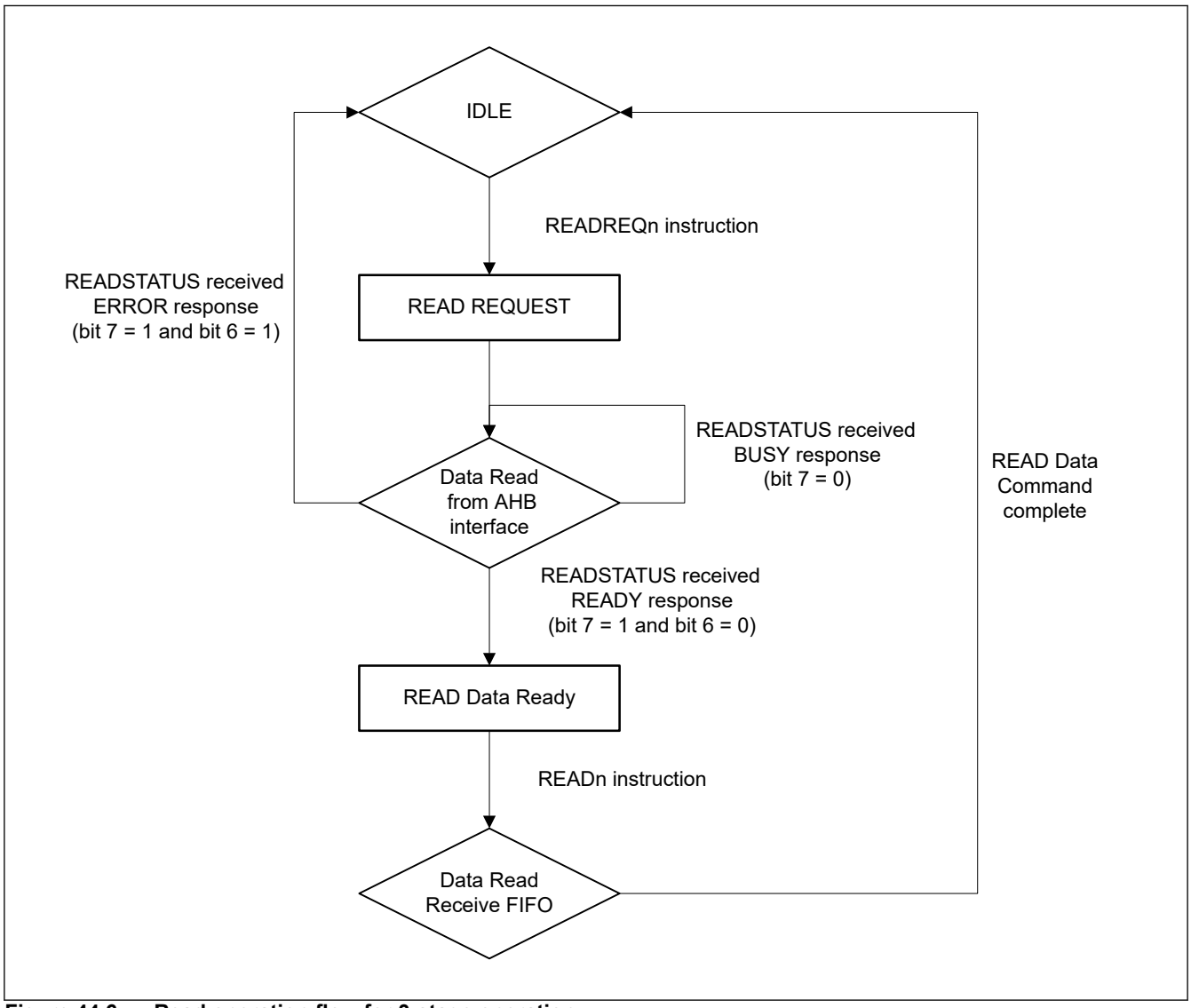


Figure 44.6 Read operation flow for 3-stage operation

(1) READREQn (Read Request Instruction)

Read transaction is always performed in the format specified by SPIMODE[1:0] bits in SHCFG register. This instruction does not require any data transfer and SHOSTIF expects only instruction and address.

Figure 44.7 shows the read request operation using READREQn instruction.

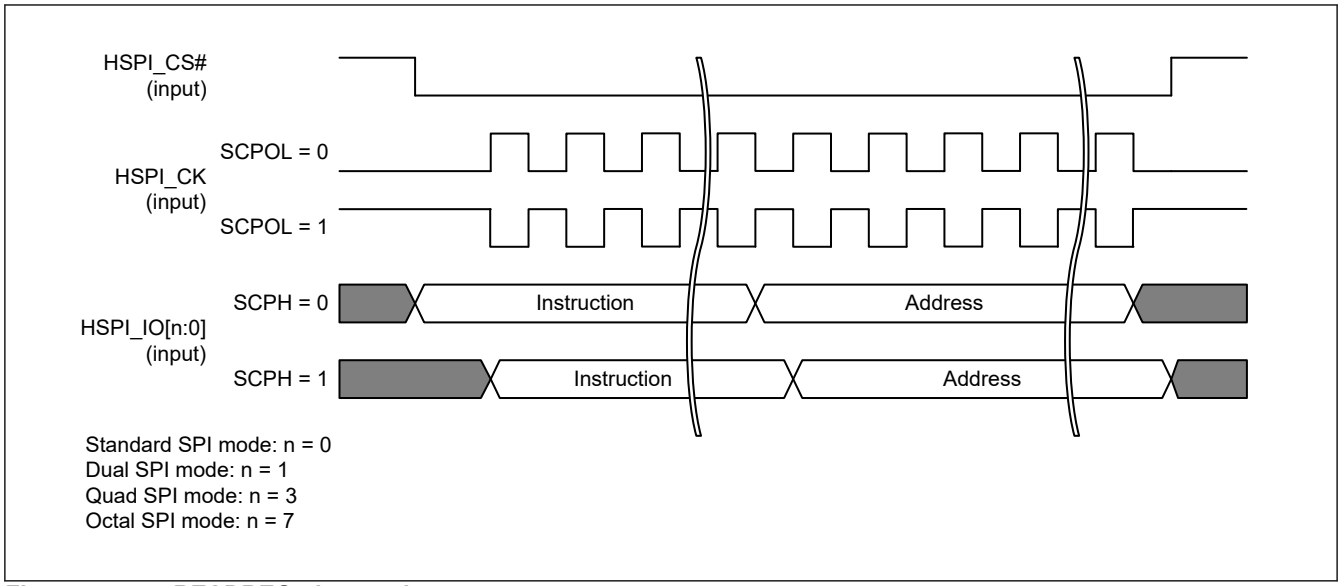


Figure 44.7 READREQn instruction

(2) READSTATUSm (Read Status Instruction, m = 0 or 1)

Read status instruction is used while SHOSTIF fetches the data from AHB interface. The SPI master always polls for data before performing a read data operation, to ensure that data is available for the read operation. SHOSTIF responds to the READSTATUSm command with 8-bit status. The MSB of the status bit indicates whether data is ready or not.

- Response bit 7 = 0 indicates that SHOSTIF is not ready with the data
- Response bit 7 = 1 and bit 6 = 0 indicates that SHOSTIF is ready with the data

Figure 44.8 shows the read status operation using READSTATUSm instruction.

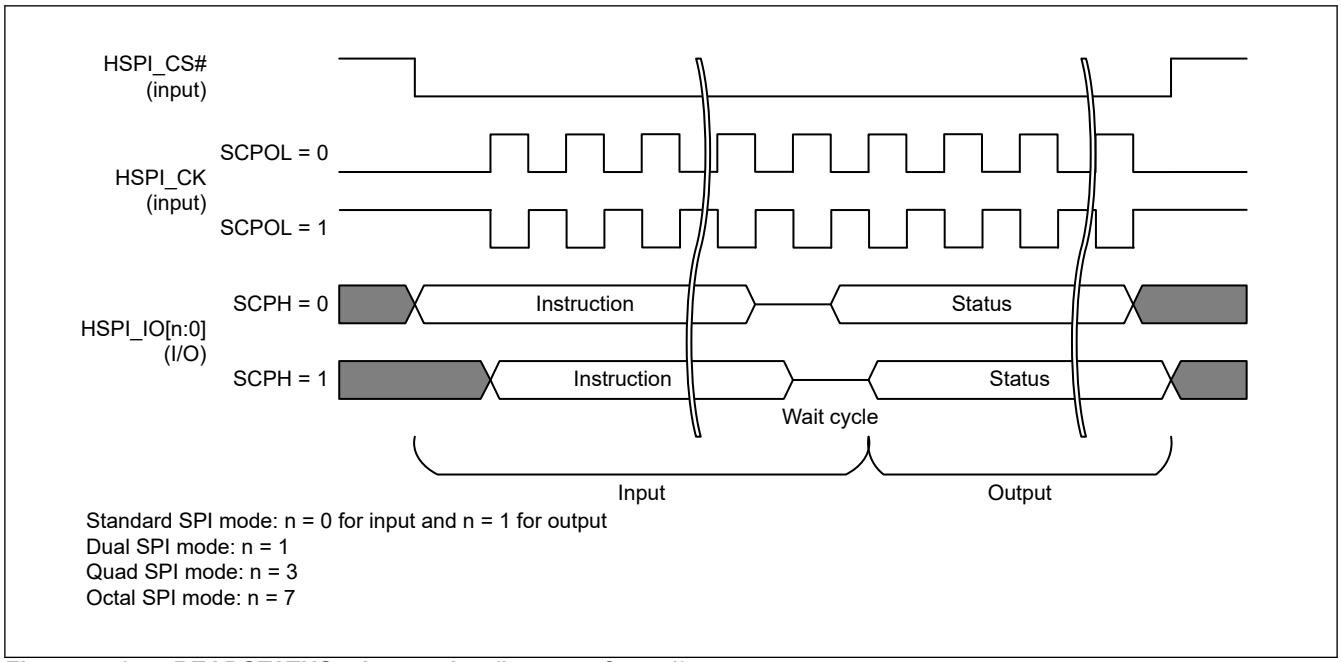


Figure 44.8 READSTATUSm instruction (in case of m = 1)

After the SPI master receives a READY response from SHOSTIF, it performs the read data operation to read the data from SHOSTIF. The AHB interface may get an ERROR response for the read transfer. In that situation, the READSTATUS instruction indicates the same status by sending a response of bit 7 =1 and bit 6 = 1. SPI master can then issue a new read transfer or retry the same operation.

Table 44.9 Response field decode

Bit	Name	Description
7	READY	Indicates whether data is ready to be fetched from SHOSTIF. 0: Not ready 1: Ready
6	ERROR	Indicates whether AHB transaction ended with an ERROR response or not. 0: No error 1: Error
5:0	Reserved	Reserved

(3) READmn (Read Data Instruction, m = 0 or 1)

After the data is available, READmn instruction is used to read back data from SHOSTIF. This time SHOSTIF does not expect any address to be transmitted by the SPI master. After the instruction is received, the inputs/outputs change direction and revert the data after some wait cycles. Wait cycles can be sent by SHOSTIF depending on the instruction (Table 44.3).

Figure 44.9 shows the read data operation using READmn instruction.

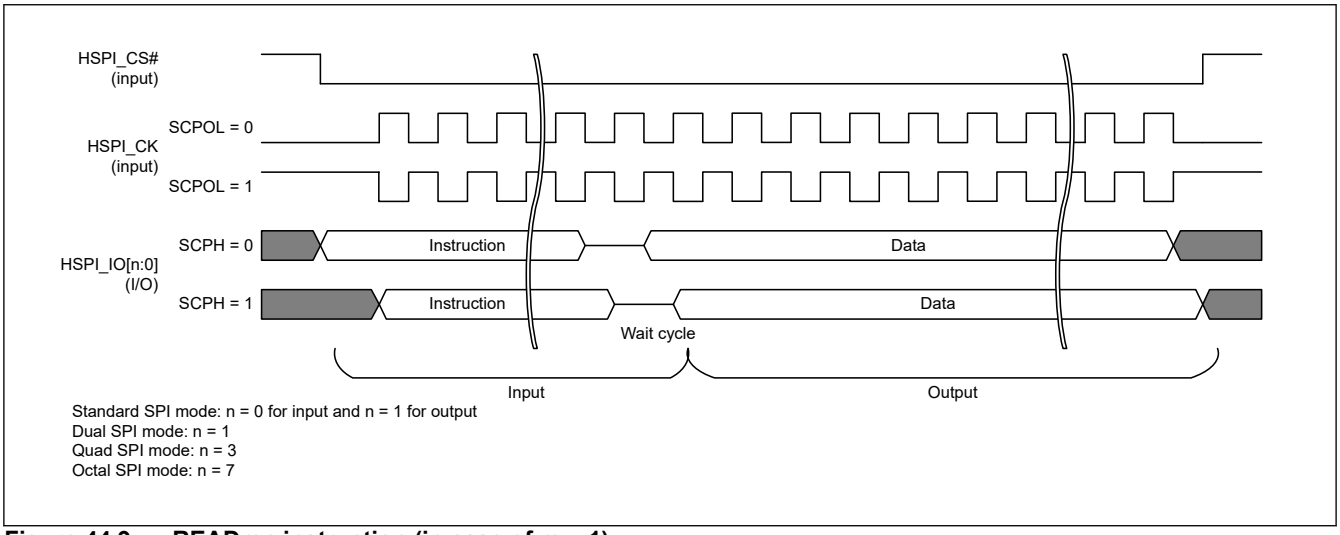


Figure 44.9 READmn instruction (in case of m = 1)

44.3.7 External SPI Master Expectations

The external SPI master must adhere to the flow specified for write and read operations. Any out-of-order operations by SPI master can move SHOSTIF into an undefined state. If SPI master behaves abnormally, then interrupt is sent by SHOSTIF. In such situations, reset SHOSTIF and restart the new transfer.

The following conditions may lead to an error state and the SPIMER bit in RISR register is set for notification.

- SPI master requests for read/write transfer when SHOSTIF is busy with AHB transfer. For example:
  - A read is requested by the SPI master and is followed by a write data transfer while read transfer is still ongoing on the AHB interface.
  - A write transfer is pending on the AHB interface but another read/write data transfer is received by SHOSTIF.
- SPI master reads less amount of data then requested during the Read request phase.
- If the SPI master requested for 16 data frames but it reads only 12 data frames, then the SPIMER bit in RISR register is set and the rest of the data is flushed from the FIFO by performing a soft reset.
- SPI master writes less amount of data than specified in the transfer length.
- If the SPI master requested for 16 data frames but it writes only 12 data frames and deselects the slave early, then the SPIMER bit in the RISR register is set and any of the data left in the FIFO is flushed by performing a soft reset.
- SPI master sends more data for a write, or SPI master read more data then requested.

- If AHB read transfer receives an error response from the slave, then no further transfer is performed for the current request, and data in the FIFO is flushed immediately. If read request transfer receives an error response on AHB interface, SHOSTIF does not expect read data transfer after that because there is no read data available in FIFO. If AHB master still performs a read data transfer, this results in an SPI master error.
- Successful read request transfer is followed by another read request from SPI master.
- A successful read request transfer (which does not receive an error response on the AHB interface) should be followed by read data or read status transfer. If a successful read request transfer is followed by another read request from SPI master, this results in an SPI master error.

44.3.8 AHB Master Interface

The AHB master interface is used as a secondary interface to read or write data for every SPI transfer. The AHB interface address and data width is fixed to 32. The following sections describe how SPI transfers are converted to AHB transactions by SHOSTIF.

44.3.8.1 Write Operation

When SHOSTIF receives a write request (WRITEn command) from SPI master, SHOSTIF saves the address per the rules specified in section 44.3.2. Address Phase. After data is received, it is transmitted as a burst transfer defined by RXFBTR register on AHB interface. This continues until the AHB slave is deselected by the AHB master. If AHB write transfer receives and ERROR response, it is indicated by ahbe_intr interrupt of the ISR and RISR registers. As soon as the error is received on the AHB interface, the rest of the data from SPI interface is discarded and no further transfers are performed for the current operation.

44.3.8.2 Read Operation

When SHOSTIF receives a read request (READmn command) from SPI master, SHOSTIF saves the address per the rules specified in section 44.3.2. Address Phase. SHOSTIF then waits for a burst to complete, and the data to be fully received. The data length is specified in the Data length field of the instruction. For data length of 1, 4, 8, or 16, the corresponding AHB burst type is used to fetch the data. For the burst length of 32 or more, INCR burst type is used to fetch the data.

If any read transfer receives an ERROR response, SHOSTIF aborts the rest of the transfer and indicates the same status in the ISR and RISR registers. When the SPI interface performs a READSTATUS instruction for data ready status, this is indicated to the SPI master with an error response of bit 6 = 1.

44.3.9 Interrupt

Interrupt signal is available in the internal CPU and external host CPU (SPI master). Multiple individual interrupt factors are combined to one signal for internal CPU and external host CPU. Individual interrupt factor can be masked commonly and separately. Figure 44.10 shows a diagram of the interrupt.

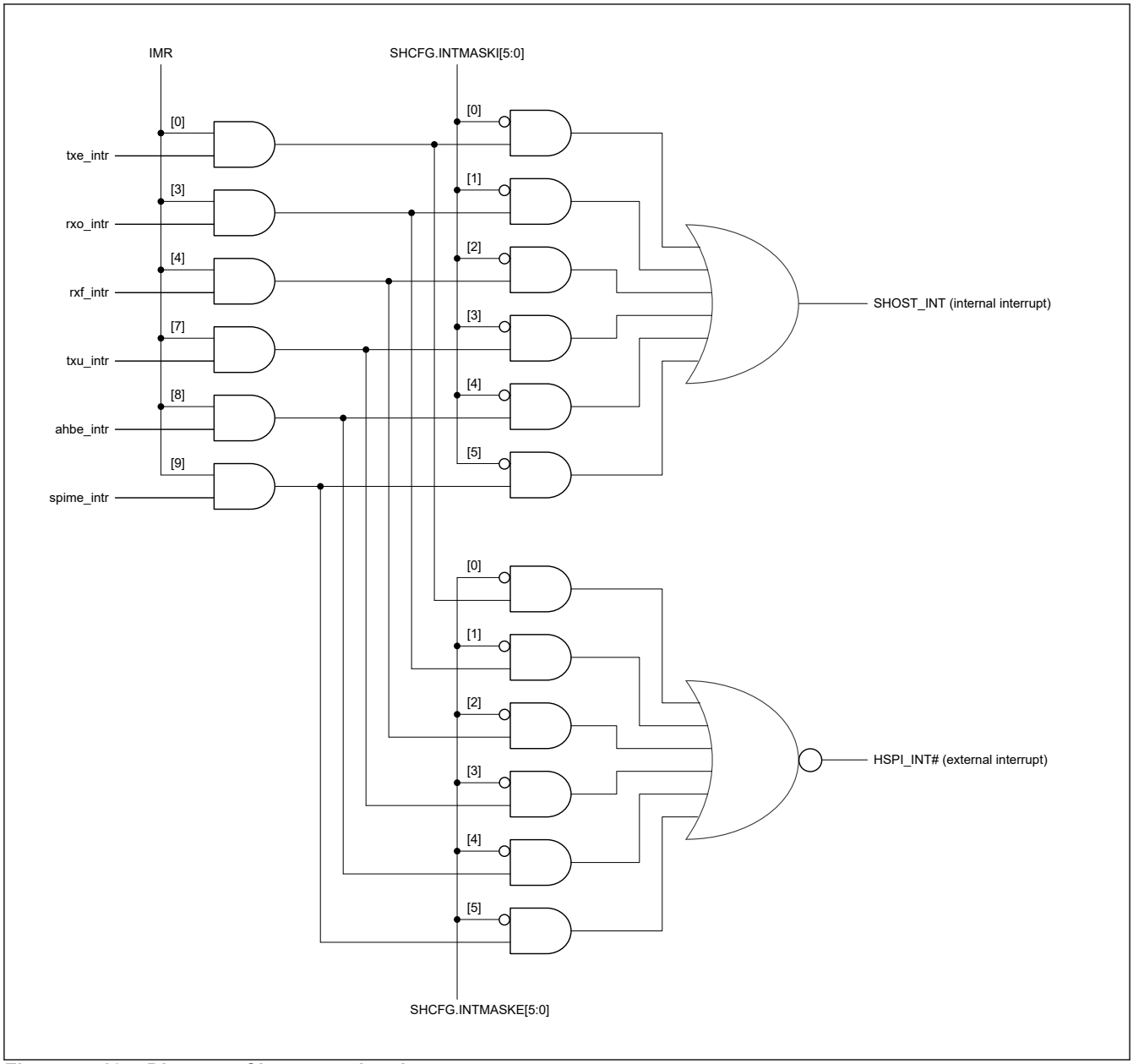


Figure 44.10 Diagram of interrupt signal

Table 44.10 SHOSTIF interrupt source (internal)

Name	Interrupt sources	CPU GIC request	DMAC activation	ELC activation
SHOST_INT	SHOSTIF interrupt (logical OR of all individual interrupts)	Possible	Not possible	Not possible

Table 44.11 SHOSTIF individual interrupt factor (1 of 2)

Factor name	Description	Interrupt clear	Interrupt mask setting		
			Common	Internal CPU	External host CPU
			IMR register	SHCFG register	
				INTMASKI[5:0] bits	INTMASKE[5:0] bits
txe_intr	Transmit FIFO (read data FIFO) Empty interrupt. When the number of transmit FIFO entries is less than or equal to transmit FIFO threshold level, the interrupt is triggered.	Transmit FIFO entries is greater than threshold	b0	b0	b0

Table 44.11 SHOSTIF individual interrupt factor (2 of 2)

Factor name	Description	Interrupt clear	Interrupt mask setting		
			Common	Internal CPU	External host CPU
			IMR register	SHCFG register	
				INTMASKI[5:0] bits	INTMASKE[5:0] bits
rxo_intr	Receive FIFO (write data FIFO) Overflow interrupt. This interrupt is not triggered if module operates normally.	RXOICR or ICR register read	b3	b1	b1
rxf_intr	Receive FIFO (write data FIFO) Full interrupt. When the number of receive FIFO entries is greater than or equal to receive FIFO threshold level + 1, the interrupt is triggered.	Receive FIFO entries is less than threshold	b4	b2	b2
txu_intr	Transmit FIFO (read data FIFO) Underflow interrupt. The interrupt is triggered in any of the following conditions: ● Data length in READmn is greater than that in READREQn in the 3-stage approach ● Transmission of AHB master interface is not completed within wait cycles specified in READmn in direct read approach	TXUICR or ICR register read	b7	b3	b3
ahbe_intr	AHB Master Interface Error interrupt. When AHB master interface receives an error response, the interrupt is triggered.	AHBECR or ICR register read	b8	b4	b4
spime_intr	SPI Master Interface Error interrupt. The interrupt is triggered in any of the following conditions: ● Illegal instruction code ● Next read or write transaction is issued before the previous transfer is completed	SPEMECR or ICR register read	b9	b5	b5

Note: If Transmit FIFO Underflow interrupt (txu_intr) happens, read access from external bus master may be impossible. Set ENR.ENABLE to 0 once.

44.4 Usage Notes

- When changing the SPI data width, disable SHOSTIF first by setting ENR.ENABLE = 0 during no transmission. Then, change SHCFG.SPIMODE. After changing the SPI data width, enable SHOSTIF again and start transmission.
- During ENR.ENABLE = 0, register access from internal CPU is available but access from external SPI master is ignored. No interrupts occur and read data is undefined.
- Do not change Receive FIFO Burst Threshold Register (RXFBTR) from 0x3F (initial value) in case of 8-bit addressing mode. In case of 32-bit addressing mode, recommendation value is 0x0F.

45. Mailbox and Semaphore (MBXSEM)

45.1 Overview

Mailbox and semaphore are available for controlling exclusive access between internal CPU (Cortex-R52) and external host CPU. Table 45.1 shows specification of mailbox and semaphore registers. These registers are accessible from external host CPU via either SHOSTIF or PHOSTIF as well as internal CPU. Concurrent use from both host interface modules is not supported.

Table 45.1 Specification of MBXSEM

Semaphore/ Mailbox	Function	Bit size	Number of register	Interrupt generation		Access property		
				Cortex-R52	Host CPU	Cortex-R52	Host CPU	Access size
Semaphore	Semaphore	1	8	No	No	R(C)/W	R(C)/W	8, 16, 32
	Read clear enable*1	8	1	No	No	R/W	R/W	8, 16, 32
Mailbox (External Host CPU to Cortex-R52)	Mailbox	32	4	No	No	R	R/W	8, 16, 32
	Interrupt Set*2 *3	4	1	Yes (each bit)	No	R	R/W	8, 16, 32
	Interrupt Clear*2 *4	4	1	Cleared (each bit)	No	R/W	R	8, 16, 32
Mailbox (Cortex-R52 to External Host CPU)	Mailbox	32	4	No	No	R/W	R	8, 16, 32
	Interrupt Set*2 *3	4	1	No	Yes (combined)	R/W	R	8, 16, 32
	Interrupt Clear*2 *4	4	1	No	Cleared (when all bits become 0)	R	R/W	8, 16, 32

Note 1. Each bit corresponds to each semaphore register.
Note 2. Each bit represents the interrupt status. When read, same interrupt status can be read through interrupt set and clear registers.
Note 3. When 1 is written to a bit, the corresponding bit is set.
Note 4. When 1 is written to a bit, the corresponding bit is cleared.

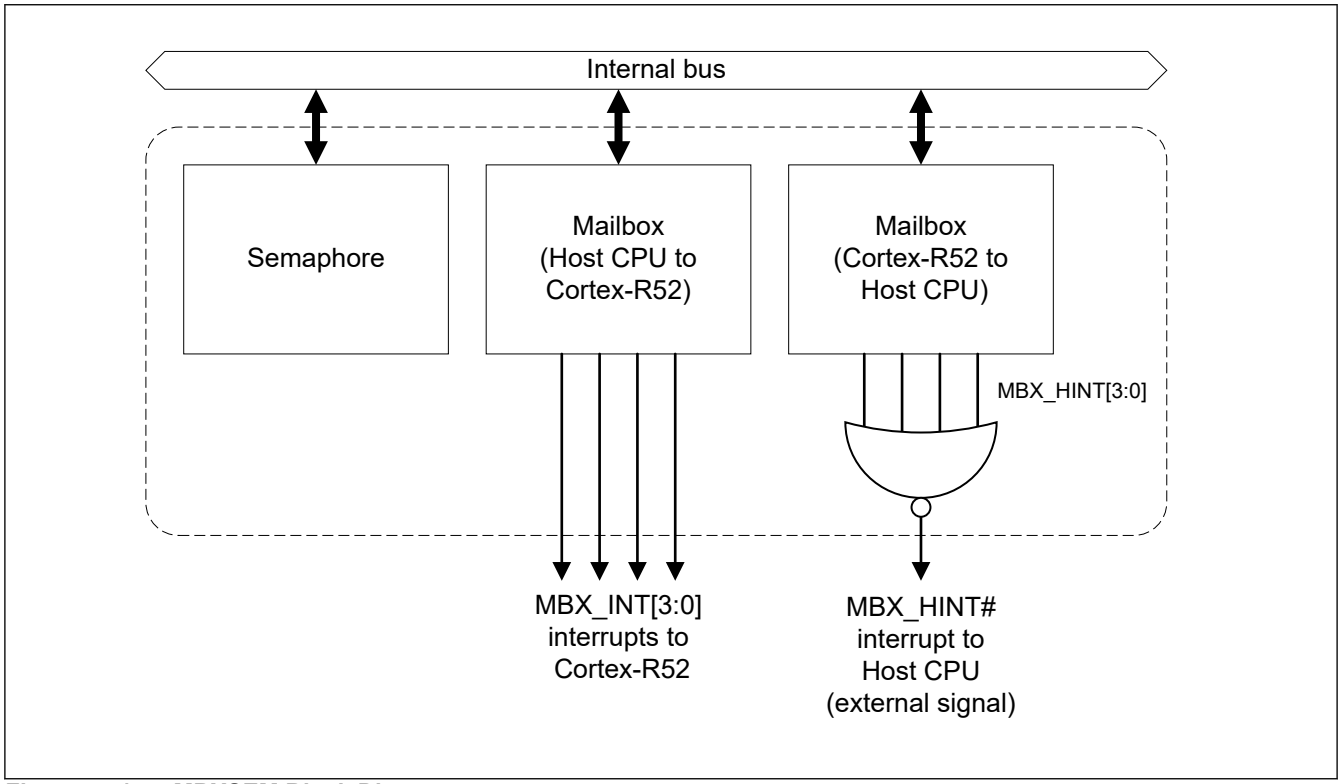


Figure 45.1 MBXSEM Block Diagram

Table 45.2 lists the output pin of the MBXSEM.

Table 45.2 MBXSEM I/O pin

Name	I/O	Function
MBX_HINT#	Output	Mailbox (Cortex-R52 to Host CPU) interrupt output pin

45.2 Register Descriptions

Register base address is different between external Host CPU view and internal memory address view except.

- External host CPU view : BASE = 0x18_0000

  Note: When 32-bit addressing mode in SHOSTIF is used, please shift the address (base address + offset address) by 2 bits in the right direction.

- Internal CPU view: BASE = 0x8024_0000

Register offset address is described in byte address. Convert it to word address by dividing by 4 if 32-bit addressing mode is used in SHOSTIF.

45.2.1 SEMn : Semaphore Register n (n = 0 to 7)

Base address: MBXSEM = 0x8024_0000
HOSTIF_E = 0x0018_0000

Offset address: 0x00 + 0x04 × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SEM
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SEM	Semaphore bit 0: Resource being used 1: Resource not being used	R/W*1
31:1	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. When read clear function is disabled (SEMRCEN.SEMRCENn = 0), register value is retained after reading the register. When read clear function is enabled (SEMRCEN.SEMRCENn = 1), register is cleared to 0 after reading the register.

SEMn register is used to perform semaphore for synchronization and exclusive control between internal CPU (Cortex-R52) and external host CPU. Resource status can be controlled when the resource is shared between CPUs. Semaphore bit can be cleared when reading this register if SEMRCEN register is enabled.

45.2.2 SEMRCEN : Semaphore Read Clear Enable Register

Base address: MBXSEM = 0x8024_0000
HOSTIF_E = 0x0018_0000

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	SEMRCEN7	SEMRCEN6	SEMRCEN5	SEMRCEN4	SEMRCEN3	SEMRCEN2	SEMRCEN1	SEMRCEN0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	SEMRCEN0	Enables or disables SEM0 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
1	SEMRCEN1	Enables or disables SEM1 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
2	SEMRCEN2	Enables or disables SEM2 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
3	SEMRCEN3	Enables or disables SEM3 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
4	SEMRCEN4	Enables or disables SEM4 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
5	SEMRCEN5	Enables or disables SEM5 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
6	SEMRCEN6	Enables or disables SEM6 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
7	SEMRCEN7	Enables or disables SEM7 register read clear function. 0: Disables read clear function 1: Enables read clear function	R/W
31:8	—	These bits are read as 0. The write value should be 0.	R/W

SEMRCEN register enables and disables read clear function of SEMn register.

45.2.3 MBXH2Cn : Host to CR52 Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000
HOSTIF_E = 0x0018_0000

Offset address: 0x40 + 0x04 × n

Bit position:	31	0
Bit field:	MBX[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from external host CPU to internal Cortex-R52	R

Note: Only read is available from internal Cortex-R52, and read and write are available from external host CPU.

MBXH2Cn register is used as mailbox from external host CPU to internal Cortex-R52.

45.2.4 MBXISETH2C : Host to CR52 Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000
HOSTIF_E = 0x0018_0000

Offset address: 0x50

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INT3S	MBX_INT2S	MBX_INT1S	MBX_INT0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INT0S	Generates or indicates MBX_INT0 interrupt of mailbox from external host CPU to internal Cortex-R52. Write from external host CPU: 0: No operation. MBX_INT0 interrupt status is not changed. 1: MBX_INT0 interrupt is asserted. Read: Indicates MBX_INT0 interrupt status	R
1	MBX_INT1S	Generates or indicates MBX_INT1 interrupt of mailbox from external host CPU to internal Cortex-R52. Write from external host CPU: 0: No operation. MBX_INT1 interrupt status is not changed. 1: MBX_INT1 interrupt is asserted. Read: Indicates MBX_INT1 interrupt status	R
2	MBX_INT2S	Generates or indicates MBX_INT2 interrupt of mailbox from external host CPU to internal Cortex-R52. Write from external host CPU: 0: No operation. MBX_INT2 interrupt status is not changed. 1: MBX_INT2 interrupt is asserted. Read: Indicates MBX_INT2 interrupt status	R
3	MBX_INT3S	Generates or indicates MBX_INT3 interrupt of mailbox from external host CPU to internal Cortex-R52. Write from external host CPU: 0: No operation. MBX_INT3 interrupt status is not changed. 1: MBX_INT3 interrupt is asserted. Read: Indicates MBX_INT3 interrupt status	R
31:4	—	These bits are read as 0.	R

Note: Only read is available from internal Cortex-R52, and read and write are available from external host CPU.

MBXISETH2C register is used to generate mailbox interrupt from external host CPU to internal Cortex-R52. Maximum 4 interrupts can be generated.

45.2.5 MBXICLRH2C : Host to CR52 Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000
HOSTIF_E = 0x0018_0000

Offset address: 0x54

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_INT3C	MBX_INT2C	MBX_INT1C	MBX_INT0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_INT0C	Clears or indicates MBX_INT0 interrupt of mailbox from external host CPU to internal Cortex-R52. Write from internal Cortex-R52: 0: No operation. MBX_INT0 interrupt status is not changed. 1: MBX_INT0 interrupt is cleared. Read: Indicates MBX_INT0 interrupt status	R/W
1	MBX_INT1C	Clears or indicates MBX_INT1 interrupt of mailbox from external host CPU to internal Cortex-R52. Write from internal Cortex-R52: 0: No operation. MBX_INT1 interrupt status is not changed. 1: MBX_INT1 interrupt is cleared. Read: Indicates MBX_INT1 interrupt status	R/W
2	MBX_INT2C	Clears or indicates MBX_INT2 interrupt of mailbox from external host CPU to internal Cortex-R52. Write from internal Cortex-R52: 0: No operation. MBX_INT2 interrupt status is not changed. 1: MBX_INT2 interrupt is cleared. Read: Indicates MBX_INT2 interrupt status	R/W
3	MBX_INT3C	Clears or indicates MBX_INT3 interrupt of mailbox from external host CPU to internal Cortex-R52. Write from internal Cortex-R52: 0: No operation. MBX_INT3 interrupt status is not changed. 1: MBX_INT3 interrupt is cleared. Read: Indicates MBX_INT3 interrupt status	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: Read and write are available from internal Cortex-R52, and only read is available from external host CPU.

MBXICLRH2C register is used to clear mailbox interrupt from external host CPU to internal Cortex-R52.

45.2.6 MBXC2Hn : CR52 to Host Mailbox Register n (n = 0 to 3)

Base address: MBXSEM = 0x8024_0000
HOSTIF_E = 0x0018_0000

Offset address: 0x80 + 0x04 × n

Bit position:	31	0
Bit field:	MBX[31:0]	
Value after reset:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	

Bit	Symbol	Function	R/W
31:0	MBX[31:0]	Mailbox from internal Cortex-R52 to external host CPU	R/W

Note: Read and write are available from internal Cortex-R52, and only read is available from external host CPU.

MBXC2Hn register is used as mailbox from internal Cortex-R52 to external host CPU.

45.2.7 MBXISETC2H : CR52 to Host Mailbox Interrupt Set Register

Base address: MBXSEM = 0x8024_0000
HOSTIF_E = 0x0018_0000

Offset address: 0x90

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_HINT3S	MBX_HINT2S	MBX_HINT1S	MBX_HINT0S
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_HINT0S	Generates or indicates MBX_HINT0 interrupt of mailbox from internal Cortex-R52 to external host CPU. Write from internal Cortex-R52: 0: No operation. MBX_INT0 interrupt status is not changed. 1: MBX_HINT0 interrupt is asserted. Read: Indicates MBX_HINT0 interrupt status	R/W
1	MBX_HINT1S	Generates or indicates MBX_HINT1 interrupt of mailbox from internal Cortex-R52 to external host CPU. Write from internal Cortex-R52: 0: No operation. MBX_INT1 interrupt status is not changed. 1: MBX_HINT1 interrupt is asserted. Read: Indicates MBX_HINT1 interrupt status	R/W
2	MBX_HINT2S	Generates or indicates MBX_HINT2 interrupt of mailbox from internal Cortex-R52 to external host CPU. Write from internal Cortex-R52: 0: No operation. MBX_INT2 interrupt status is not changed. 1: MBX_HINT2 interrupt is asserted. Read: Indicates MBX_HINT2 interrupt status	R/W
3	MBX_HINT3S	Generates or indicates MBX_HINT3 interrupt of mailbox from internal Cortex-R52 to external host CPU. Write from internal Cortex-R52: 0: No operation. MBX_INT3 interrupt status is not changed. 1: MBX_HINT3 interrupt is asserted. Read: Indicates MBX_HINT3 interrupt status	R/W
31:4	—	These bits are read as 0. The write value should be 0.	R/W

Note: Read and write are available from internal Cortex-R52, and only read is available from external host CPU.

MBXISETC2H register is used to generate mailbox interrupt from internal Cortex-R52 to external host CPU. Maximum 4 interrupts can be generated but all the interrupts are combined to one output interrupt signal (MBX_HINT#) by logical NOR.

45.2.8 MBXICLRC2H : CR52 to Host Mailbox Interrupt Clear Register

Base address: MBXSEM = 0x8024_0000
HOSTIF_E = 0x0018_0000

Offset address: 0x94

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	MBX_HINT3C	MBX_HINT2C	MBX_HINT1C	MBX_HINT0C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MBX_HINT0C	Clears or indicates MBX_HINT0 interrupt of mailbox from internal Cortex-R52 to external host CPU. Write from external host CPU: 0: No operation. MBX_HINT0 interrupt status is not changed. 1: MBX_HINT0 interrupt is cleared. Read: Indicates MBX_HINT0 interrupt status	R
1	MBX_HINT1C	Clears or indicates MBX_HINT1 interrupt of mailbox from internal Cortex-R52 to external host CPU. Write from external host CPU: 0: No operation. MBX_HINT1 interrupt status is not changed. 1: MBX_HINT1 interrupt is cleared. Read: Indicates MBX_HINT1 interrupt status	R
2	MBX_HINT2C	Clears or indicates MBX_HINT2 interrupt of mailbox from internal Cortex-R52 to external host CPU. Write from external host CPU: 0: No operation. MBX_HINT2 interrupt status is not changed. 1: MBX_HINT2 interrupt is cleared. Read: Indicates MBX_HINT2 interrupt status	R
3	MBX_HINT3C	Clears or indicates MBX_HINT3 interrupt of mailbox from internal Cortex-R52 to external host CPU. Write from external host CPU: 0: No operation. MBX_HINT3 interrupt status is not changed. 1: MBX_HINT3 interrupt is cleared. Read: Indicates MBX_HINT3 interrupt status	R
31:4	—	These bits are read as 0.	R

Note: Only read is available from internal Cortex-R52, and read and write are available from external host CPU.

MBXICLRC2H register is used to clear mailbox interrupt from internal Cortex-R52 to external host CPU.

45.3 Operation

45.3.1 Semaphore

SEMn registers are used to perform up to 8 semaphores for synchronization and exclusive control between internal CPU (Cortex-R52) and external host CPU.

SEM bit in SEMn register is used to set the status of the resource shared between CPUs to indicate whether the resource is being used or not. Writing 1 to this bit maintains its value. When the read clear function is enabled (SEMRCEN.SEMRCENn bit = 1), reading SEM bit of corresponding SEM register clears its value to 0.

When the SEM0 register is used for semaphore control for a resource A, for example, the following procedure takes place, where CPU0 is Cortex-R52 and CPU1 is external host CPU or vice versa:

1. Use the initialization routine of CPU0 to set the SEMRCEN.SEMRCEN0 bit to 1 (this enables the read clear function for SEM0 register).
2. Set the SEM bit to 1 (resource not being used) in the SEM0 register.
3. For example, when CPU1 attempts to use the resource A, CPU1 reads the SEM0 register repeatedly until 1 is read.
4. When SEM0.SEM = 1 is read, CPU1 confirms that the resource is not being used.
   This operation clears the SEM0.SEM bit to 0.
5. Then CPU1 uses the resource A.
   If CPU0 attempts to use the resource A while CPU1 is using it, 0 is read from the SEM0.SEM bit (resource A being used by CPU1). CPU0 repeats read operation and waits until the resource is released.
6. When CPU1 completes its use of the resource A, 1 is written in the SEM0.SEM bit to indicate that the resource A is released.
7. If CPU0 attempts to use the resource A after the above step (6), it confirms whether the resource is used or not by reading the SEM0.SEM bit. When the SEM0.SEM bit is 1, CPU0 starts to use the resource A. This operation clears the SEM0.SEM bit to 0.

45.3.2 Mailbox

MBXH2Cn and MBXC2Hn register are used to perform up to 4 32-bit mailboxes for communication between internal CPU (Cortex-R52) and external host CPU. MBXH2Cn registers are mailboxes from external host CPU to Coretex-R52 and MBXC2Hn registers are vice versa.

45.3.3 Interrupt

Interrupt signals are available to internal CPU and external host CPU. One CPU can generate up to 4 interrupts to the other CPU each other. They can be related to 4 mailboxes and be used independent of mailbox. MBXISETH2C and MBXISETC2H registers are used to generate interrupt by writing 1 to corresponding bit. Writing 0 to corresponding bit is to retain the current interrupt status of the bit. MBXICLRH2C and MBXICLRC2H registers are used to clear interrupt by writing 1 to corresponding bit. Current interrupt status can be confirmed by reading these registers.

Table 45.3 MBXSEM interrupt source (to Cortex-R52)

Name	Interrupt sources	CPU GIC request	DMAC activation	ELC activation
MBX_INT0	Mailbox (Host CPU to Cortex-R52) interrupt 0	Possible	Possible	Possible
MBX_INT1	Mailbox (Host CPU to Cortex-R52) interrupt 1	Possible	Possible	Possible
MBX_INT2	Mailbox (Host CPU to Cortex-R52) interrupt 2	Possible	Possible	Possible
MBX_INT3	Mailbox (Host CPU to Cortex-R52) interrupt 3	Possible	Possible	Possible

Table 45.4 MBXSEM interrupt source (to external Host CPU)

Name	Interrupt sources	CPU GIC request	DMAC activation	ELC activation
MBX_HINT0	Mailbox (Cortex-R52 to Host CPU) interrupt 0	Not Possible	Not Possible	Not Possible
MBX_HINT1	Mailbox (Cortex-R52 to Host CPU) interrupt 1	Not Possible	Not Possible	Not Possible
MBX_HINT2	Mailbox (Cortex-R52 to Host CPU) interrupt 2	Not Possible	Not Possible	Not Possible
MBX_HINT3	Mailbox (Cortex-R52 to Host CPU) interrupt 3	Not Possible	Not Possible	Not Possible

Note: MBX_HINT[3:0] are combined to one output interrupt signal (MBX_HINT#) by logical NOR.

46. Encoder Interface Subsystem

46.1 Overview

Encoder interface subsystem consists of EnDat 2.2 (ENDAT), BiSS-C (BISS), HIPERFACE DSL (HDSL), and A-format. The encoder interface I/O pins are shared with these modules, and one module is statically selected for each channel. Table 46.1 describes the encoder interface subsystem specifications and Figure 46.1 shows a block diagram of the encoder interface subsystem.

Table 46.1 Encoder interface subsystem specifications

Item	Description
Number of units	2 units
Supported encoder interface	• EnDat 2.2 (ENDAT) • BiSS-C (BISS) • HIPERFACE DSL • A-format (AFMT)
Event link	Trigger input
Module-stop function	Module-stop state can be set to reduce power consumption for each module

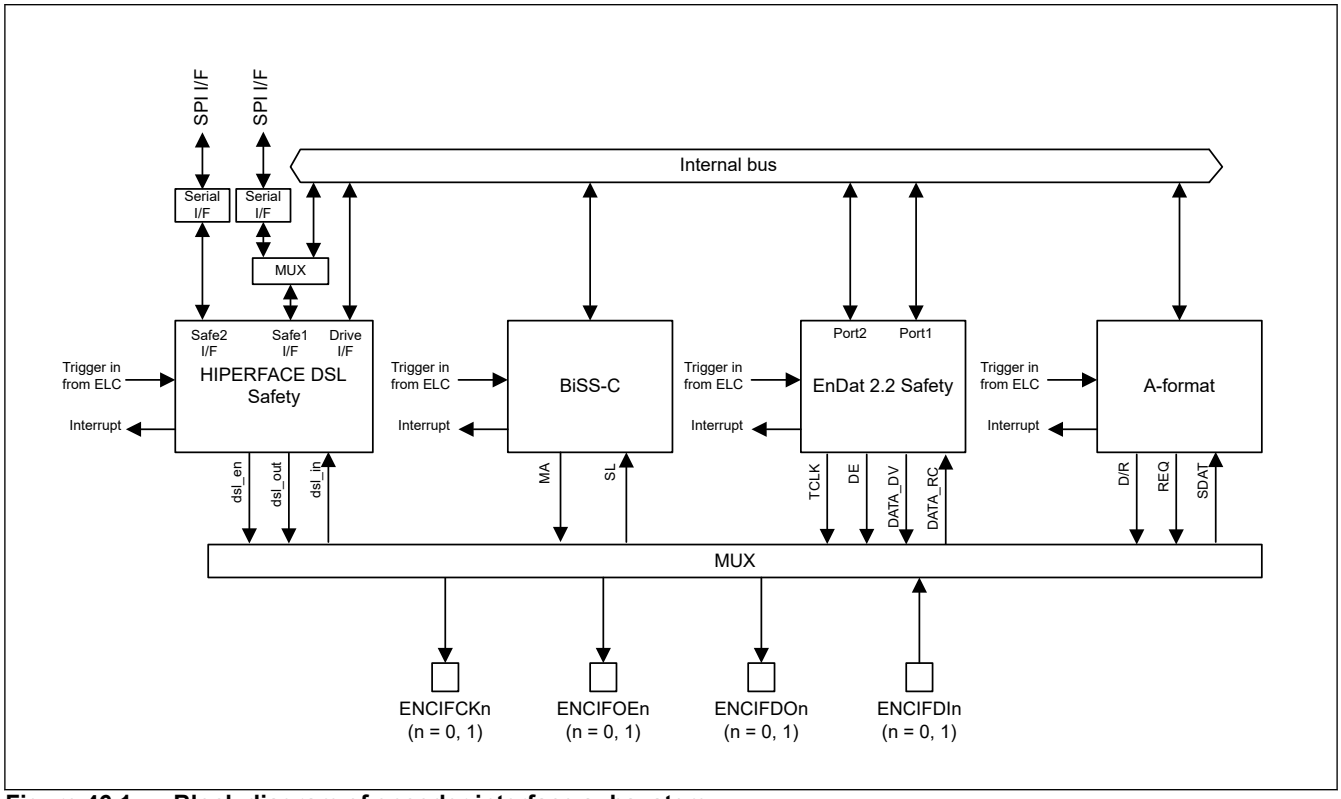


Figure 46.1 Block diagram of encoder interface subsystem

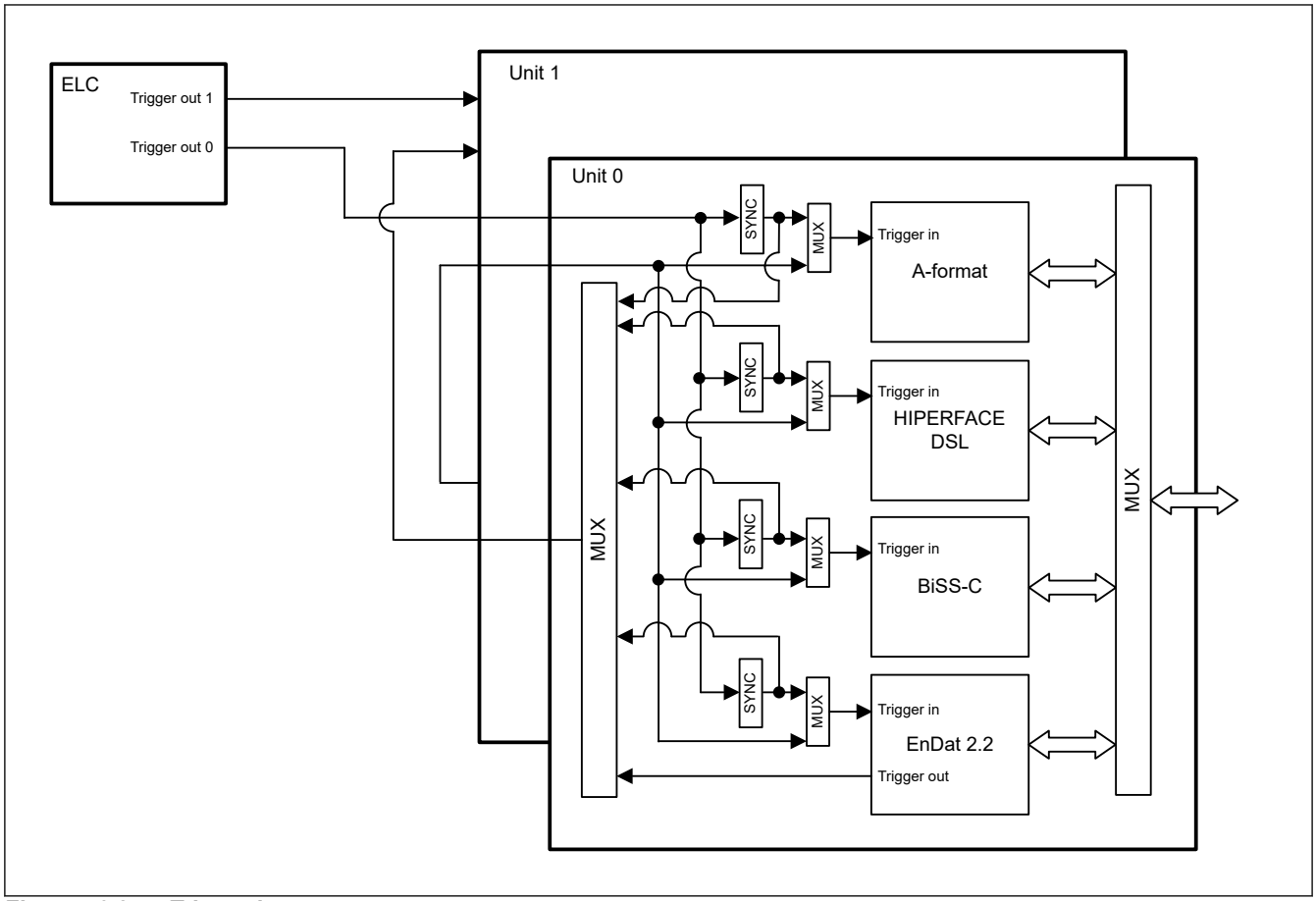


Figure 46.2 Trigger input

Table 46.2 lists the input/output pins of the encoder interface subsystem.

Table 46.2 Pin configuration of the encoder interface subsystem

Channel	Pin name	I/O	Function
CH0	ENCIFCK0	Output	Encoder interface clock output pin
	ENCIFOE0	Output	Encoder interface data output enable pin
	ENCIFDO0	Output	Encoder interface data output pin
	ENCIFDI0	Input	Encoder interface data input pin
CH1	ENCIFCK1	Output	Encoder interface clock output pin
	ENCIFOE1	Output	Encoder interface data output enable pin
	ENCIFDO1	Output	Encoder interface data output pin
	ENCIFDI1	Input	Encoder interface data input pin

Table 46.3 Encoder interface pin function of each module

Pin name	EnDat 2.2	BiSS-C	HIPERFACE DSL	A-format
ENCIFCKn	TCLK	MA	(N/A)	(N/A)
ENCIFOEn	DE	(N/A)	dsl_en	D/R
ENCIFDOn	DATA_DV	(N/A)	dsl_out	REQ
ENCIFDIn	DATA_RC	SL	dsl_in	SDAT

Note: n = 0, 1

46.2 Register Descriptions

46.2.1 ENCODER_CFG : Encoder IF Configuration Register

Base address: ENCSS = 0x8028_1400

Offset address: 0x00

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	TRGSEL1	SS1SEL[1:0]		EVTSRC1	HF1SAFE1SEL	HF1ENDIAN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	TRGSEL0	SS0SEL[1:0]		EVTSRC0	HF0SAFE1SEL	HF0ENDIAN
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	HF0ENDIAN	HIPERFACE DSL Unit 0 Endian Mode Byte sequence selection for register access 0: Little endian 1: Big endian	R/W
1	HF0SAFE1SEL	HIPERFACE DSL Unit 0 Safe 1 Interface Select Select the interface mode for safe 1 interface 0: Internal bus interface 1: SPI interface	R/W
2	EVTSRC0	Event Source Select Select the event trigger source for encoder interface subsystem unit 0 0: Trigger from ELC 1: Trigger from encoder interface subsystem unit 1	R/W
4:3	SS0SEL[1:0]	Encoder Interface Port Select Select the encoder interface for unit 0 0 0: HIPERFACE DSL 0 1: BiSS-C 1 0: EnDat 2.2 1 1: A-format	R/W
5	TRGSEL0	Trigger Source Select for Encoder Interface Subsystem Unit 1 Select the event trigger for encoder interface subsystem unit 1 0: Trigger from ELC 1: Timer of EnDat 2.2 unit 0	R/W
15:6	—	These bits are read as 0. The write value should be 0.	R/W
16	HF1ENDIAN	HIPERFACE DSL Unit 1 Endian Mode Byte sequence selection for register access 0: Little endian 1: Big endian	R/W
17	HF1SAFE1SEL	HIPERFACE DSL Unit 1 Safe 1 Interface Select Select the interface mode for safe 1 interface 0: Internal bus interface 1: SPI interface	R/W
18	EVTSRC1	Event Source Select Select the event trigger source for unit 1 0: Trigger from ELC 1: Trigger from encoder interface subsystem unit 0	R/W

Bit	Symbol	Function	R/W
20:19	SS1SEL[1:0]	Encoder Interface Port Select Select the encoder interface for unit 1 0 0: HIPERFACE DSL 0 1: BiSS-C 1 0: EnDat 2.2 1 1: A-format	R/W
21	TRGSEL1	Trigger Source Select for Encoder Interface Unit 0 Select the event trigger for encoder interface unit 0 0: Trigger from ELC 1: Timer of EnDat 2.2 unit 1	R/W
31:22	—	These bits are read as 0. The write value should be 0.	R/W

The ENCODER_CFG register is used to set the encoder interface operating mode. This register should be set before releasing the module stop for each encoder interface module.

46.2.2 BISS_STAT : BiSS-C Status Register

Base address: ENCSS = 0x8028_1400

Offset address: 0x04

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	BISS1_EOT	—	—	—	BISS1_ERR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	BISS0_EOT	—	—	—	BISS0_ERR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1

Bit	Symbol	Function	R/W
0	BISS0_ERR	BiSS-C Unit 0 Error 0: External or internal error occured 1: No error occurred	R
3:1	—	These bits are read as 0.	R
4	BISS0_EOT	End-of-Transmission from BiSS-C Unit 0 0: Data transmission active 1: Data transmission inactive	R
15:5	—	These bits are read as 0.	R
16	BISS1_ERR	BiSS-C Unit 1 Error 0: External or internal error occured 1: No error occurred	R
19:17	—	These bits are read as 0.	R
20	BISS1_EOT	End-of-Transmission from BiSS-C Unit 1 0: Data transmission active 1: Data transmission inactive	R
31:21	—	These bits are read as 0.	R

The BISS_STAT register indicates the status of BiSS-C.

46.2.3 HDSL_STAT : HIPERFACE DSL Status Register

Base address: ENCSS = 0x8028_1400

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	HDSL1_ENCERR	HDSL1_ACTHERR	HDSL1_ACCERR	HDSL1_EDTERR	HDSL1_SPERR	HDSL1_SCHERR	HDSL1_ESTON	—	—	HDSL1_SYNCLCKD	HDSL1_LINK
Value after reset:	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	HDSL0_ENCERR	HDSL0_ACTHERR	HDSL0_ACCERR	HDSL0_EDTERR	HDSL0_SPERR	HDSL0_SCHERR	HDSL0_ESTON	—	—	HDSL0_SYNCLCKD	HDSL0_LINK
Value after reset:	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0

Bit	Symbol	Function	R/W
0	HDSL0_LINK	HIPERFACE-DSL Link Connection indication from HIPERFACE-DSL unit 0 0: Disconnect 1: Connect	R
1	HDSL0_SYNCLCKD	HIPERFACE-DSL Sync Locked Position sampling resolution locked from HIPERFACE-DSL unit 0 0: Unlock 1: Locked	R
3:2	—	These bits are read as 0.	R
4	HDSL0_ESTON	HIPERFACE-DSL Estimator ON Position estimator activated from HIPERFACE-DSL unit 0 0: Inactive 1: Active	R
5	HDSL0_SCHERR	HIPERFACE-DSL Safe Channel Error Transmission error in safe channel 1 from HIPERFACE-DSL unit 0 0: No error 1: Error	R
6	HDSL0_SPERR	HIPERFACE-DSL Safe Position Error Safe position not valid from HIPERFACE-DSL unit 0 0: No error 1: Error	R
7	HDSL0_EDTERR	HIPERFACE-DSL Estimator Deviation Threshold Error Estimator deviation threshold reached from HIPERFACE-DSL unit 0 0: No error 1: Error	R
8	HDSL0_ACCERR	HIPERFACE-DSL Acceleration Error Fast channel/position error from HIPERFACE-DSL unit 0 0: No error 1: Error	R
9	HDSL0_ACTHERR	HIPERFACE-DSL Acceleration Threshold Error Fast channel/position threshold error from HIPERFACE-DSL unit 0 0: No error 1: Error	R
10	HDSL0_ENCERR	HIPERFACE-DSL Encoding Error 8B/10B encoding error in DSL frame transmission from HIPERFACE-DSL unit 0 0: No error 1: Error	R
15:11	—	These bits are read as 0.	R

Bit	Symbol	Function	R/W
16	HDSL1_LINK	HIPERFACE-DSL Link Connection indication from HIPERFACE-DSL unit 1 0: Disconnect 1: Connect	R
17	HDSL1_SYNCLCKD	HIPERFACE-DSL Sync Locked Position sampling resolution locked from HIPERFACE-DSL unit 1 0: Unlock 1: Locked	R
19:18	—	These bits are read as 0.	R
20	HDSL1_ESTON	HIPERFACE-DSL Estimator ON Position estimator activated from HIPERFACE-DSL unit 1 0: Inactive 1: Active	R
21	HDSL1_SCHERR	HIPERFACE-DSL Safe Channel Error Transmission error in safe channel 1 from HIPERFACE-DSL unit 1 0: No error 1: Error	R
22	HDSL1_SPERR	HIPERFACE-DSL Safe Position Error Safe position not valid from HIPERFACE-DSL unit 1 0: No error 1: Error	R
23	HDSL1_EDTERR	HIPERFACE-DSL Estimator Deviation Threshold Error Estimator deviation threshold reached from HIPERFACE-DSL unit 1 0: No error 1: Error	R
24	HDSL1_ACCERR	HIPERFACE-DSL Acceleration Error Fast channel/position error from HIPERFACE-DSL unit 1 0: No error 1: Error	R
25	HDSL1_ACTHERR	HIPERFACE-DSL Acceleration Threshold Error Fast channel/position threshold error from HIPERFACE-DSL unit 1 0: No error 1: Error	R
26	HDSL1_ENCERR	HIPERFACE-DSL Encoding Error 8B/10B encoding error in DSL frame transmission from HIPERFACE-DSL unit 1 0: No error 1: Error	R
31:27	—	These bits are read as 0.	R

The HDSL_STAT register indicates the status of HIPERFACE-DSL.

46.2.4 ENDAT_CFG : EnDat Configuration Register

Base address: ENCSS = 0x8028_1400

Offset address: 0x0C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	ENDAT1_AXADD[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	ENDAT0_AXADD[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	ENDAT0_AXADD[4:0]	EnDat Unit 0 Axis Address The component address for EnDat 2.2 unit 0	R/W
15:5	—	These bits are read as 0. The write value should be 0.	R/W
20:16	ENDAT1_AXADD[4:0]	EnDat Unit 1 Axis Address The component address for EnDat 2.2 unit 1	R/W
31:21	—	These bits are read as 0. The write value should be 0.	R/W

The ENDAT_CFG register is used to set the EnDat 2.2.

46.2.5 AFMT_STAT : A-Format Status Register

Base address: ENCSS = 0x8028_1400

Offset address: 0x10

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	AFMT1_TMOUT	AFMT1_EOT	AFMT1_BUSY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	AFMT0_TMOUT	AFMT0_EOT	AFMT0_BUSY
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

Bit	Symbol	Function	R/W
0	AFMT0_BUSY	A-format Busy Transmission busy from A-format unit 0 0: Busy 1: Idle	R
1	AFMT0_EOT	A-format End-Of-Frame End-of-transmission from A-format unit 0 0: Data transfer end (cleared by data read) 1: Idle or in-progress	R
2	AFMT0_TMOUT	A-format Timeout Timeout error from A-format unit 0 0: No error 1: Time out error	R
15:3	—	These bits are read as 0.	R
16	AFMT1_BUSY	A-format Busy Transmission busy from A-format unit 1 0: Busy 1: Idle	R
17	AFMT1_EOT	A-format End-of-Frame End-of-transmission from A-format unit 1 0: Data transfer end (cleared by data read) 1: Idle or in-progress	R
18	AFMT1_TMOUT	A-format Timeout Timeout error from A-format unit 1 0: No error 1: Time out error	R
31:19	—	These bits are read as 0.	R

The AFMT_STAT register indicates the status of A-format.

46.2.6 HDSLn_OS_D : HIPERFACE_DSL (SSn) Online Status (Drive) Register (n = 0, 1)

Base address: ENCSS = 0x8028_1400

Offset address: 0x14 + 0xC × n

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	INT	SUM	—	—	POS	—	DTE	PRST	POSTX[1:0]		MIN	ANS	—	QMLW	FREL	—
Value after reset:	0	0	0	1	1	0	1	1	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0.	R
1	FREL	Channel Status for long message 0: The channel for the long message is in use. 1: The channel for the long message is free.	R
2	QMLW	Quality monitoring at Low level 0: Current value of quality monitoring greater than or equal to 14 1: Current value of quality monitoring less than 14	R
3	—	This bit is read as 0.	R
4	ANS	Incorrect Answer Detected 0: No error detected in the last answer to a long message 1: The last answer to a long message was damaged	R
5	MIN	Acknowledgment of Message Initialization 0: Parameter Channel not functioning 1: The DSL encoder sends a figure by which the initialization of the Parameter Channel is acknowledged	R
7:6	POSTX[1:0]	Position Transmission Status 0 0: Position request is transmitted to the DSL encoder 0 1: Safe position 1 was received 1 0: Fast position was received or position newly updated by estimator 1 1: Safe position 2 was received	R
8	PRST	Protocol Reset 0: HDSLn running 1: HDSLn has restarted the protocol	R
9	DTE	Deviation Threshold Error 0: Current value of deviation smaller than the specified maximum 1: Current value of deviation greater than the specified maximum	R
10	—	This bit is read as 0.	R
11	POS	Estimator Turned On 0: No fast position error 1: A source of an error in the fast position was identified or an alignment procedure is currently being carried out. It is probable that the last fast position is invalid. Be aware that the fast position read through drive interface is provided by the estimator.	R
12	—	This bit is read as 1.	R
13	—	This bit is read as 0.	R
14	SUM	Summary Byte 0: The last valid value from MIR_SUM was zero. 1: The last valid value from MIR_SUM was not zero. The importance of this flag depends on the particular error source that leads to a set MIR_SUM.	R

Bit	Symbol	Function	R/W
15	INT	Status of the Interrupt Output 0: Interrupt output on the low level 1: Interrupt output on the high level	R
31:16	—	These bits are read as 0.	R

The HDSLn_OS_D register is used to indicate a status of HIPERFACE_DSL. (n = 0, 1)

46.2.7 HDSL0_OS_1 : HIPERFACE_DSL (SS0) Online Status (Safe1) Register

Base address: ENCSS = 0x8028_1400

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SINT	SSUM	SCE	—	—	VPOS	—	PRST	POSTX[1:0]		MIN	—	—	QMLW	—	FRES
Value after reset:	0	0	1	1	0	1	0	1	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
0	FRES	Channel Status for the short message 0: The channel for the short message is in use. 1: The channel for the short message is free.	R
1	—	This bit is read as 0.	R
2	QMLW	Quality Monitoring at Low level 0: Current value of quality monitoring greater than or equal to 14 1: Current value of quality monitoring less than 14	R
4:3	—	These bits are read as 0.	R
5	MIN	Acknowledgment of Message Initialization 0: Parameter channel not functioning 1: The DSL encoder sends a figure by which the initialization of the parameter channel is acknowledged.	R
7:6	POSTX[1:0]	Position Transmission Status 0 0: Position request is transmitted to the DSL encoder. 0 1: Safe position 1 was received. 1 0: Fast position was received or position newly updated by estimator. 1 1: Safe position 2 was received.	R
8	PRST	Protocol Reset 0: HDSL0 running 1: HDSL0 has restarted the protocol	R
9	—	This bit is read as 0.	R
10	VPOS	Safe Position Invalid 0: The last safe position received was correct. 1: An error in the safe position was identified. It is expected that the safe position transmitted from the encoder is invalid.	R
11	—	This bit is read as 0.	R
12	—	This bit is read as 1.	R
13	SCE	CRC Error on the Safe Channel 0: The last Safe Channel CRC received was correct. 1: The last Safe Channel CRC received was wrong. It is expected that the last safe position 2 transmitted is invalid.	R

Bit	Symbol	Function	R/W
14	SSUM	Safe Summary Bit 0: The last valid value from SAFE_SUM was zero. 1: The last valid value from SAFE_SUM was not zero. The importance of this flag depends on the particular error source that leads to a set SAFE_SUM.	R
15	SINT	Status of the Interrupt Output 0: Interrupt output on the low level 1: Interrupt output on the high level	R
31:16	—	These bits are read as 0.	R

The HDSL0_OS_1 register is used to indicate a status of HIPERFACE_DSL.

46.2.8 HDSL0_OS_2 : HIPERFACE_DSL (SS0) Online Status (Safe2) Register

Base address: ENCSS = 0x8028_1400

Offset address: 0x1C

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	SSUM2	SCE2	—	—	VPOS2	—	PRST	POSTX[1:0]		—	—	—	QMLW	—	—
Value after reset:	0	0	1	1	0	0	0	1	0	0	0	0	0	1	0	0

Bit	Symbol	Function	R/W
1:0	—	This bit is read as 0.	R
2	QMLW	Quality Monitoring at Low level 0: Current value of quality monitoring greater than or equal to 14 1: Current value of quality monitoring less than 14	R
5:3	—	These bits are read as 0.	R
7:6	POSTX[1:0]	Position Transmission Status 0 0: Position request is transmitted to the DSL encoder. 0 1: Safe position 1 was received. 1 0: Fast position was received or position newly updated by estimator. 1 1: Safe position 2 was received.	R
8	PRST	Protocol Reset 0: HDSL0 running 1: HDSL0 has restarted the protocol	R
9	—	This bit is read as 0.	R
10	VPOS2	Safe Position 2 Invalid 0: The last safe position received in channel 2 was correct. 1: A source of an error in the safe position in channel 2 was identified. It is probable that the safe position transmitted from Channel 2 is invalid. Suitable measures must be installed in the user application.	R
11	—	This bit is read as 0.	R
12	—	This bit is read as 1.	R
13	SCE2	Transmission Error Channel 2 0: The last data received in channel 2 was correct. 1: The last safe channel 2 CRC received was wrong. It is expected that the last safe position 2 transmitted is invalid. Suitable measures must be installed in the user application.	R

Bit	Symbol	Function	R/W
14	SSUM2	Summary Bit Channel 2 0: Neither TEST2 nor ERR2 is set. 1: One of the indications TEST2 or ERR2 is set. The error reaction to the flag depends on the meaning of the bit.	R
31:15	—	These bits are read as 0.	R

The HDSL0_OS_2 register is used to indicate a status of HIPERFACE_DSL.

46.3 Usage Notes

46.3.1 Release of Register Write Protection

ENCODER_CFG and ENDAT_CFG registers are write protected. To release write protection, use the following steps:

1. Write 0x0000A508 to PRCRN register (release the write protection by PRC3 bit).
2. Write value to ENCODER_CFG or ENDAT_CFG registers.
3. Write 0x0000A500 to PRCRN register (set the write protection by PRC3 bit).

47. EnDat 2.2 (ENDAT)

47.1 Overview

The MPU includes EnDat 2.2 (ENDAT) master to enable communication between the encoder and EnDat 2.2 interface. Table 47.1 describes the ENDAT specifications and Figure 47.1 shows a block diagram of the ENDAT.

Table 47.1 EnDat 2.2 (ENDAT) specifications

Item	Description
Number of units	2 units
Protocol	EnDat 2.2
Functions	• EnDat 2.2 Basic • Host interface: 32 bits APB • Basic functions of EnDat data transfer: – Clock generator – Data word length – Continuous clock – Error handling – Encoder status – CRC – Cable preparation time – Recovery time II – Recovery time III • Advanced functions of EnDat data transfer – Watchdog – Reset with selected additional information • Strobe source: Hardware, timer and software • Digital filter for data input
Interrupt sources	• Basic interrupt: 1 – Watchdog, receive register updated
Event link	STR input for HW trigger provided from ELC
Module-stop function	Module-stop state can be set to reduce power consumption

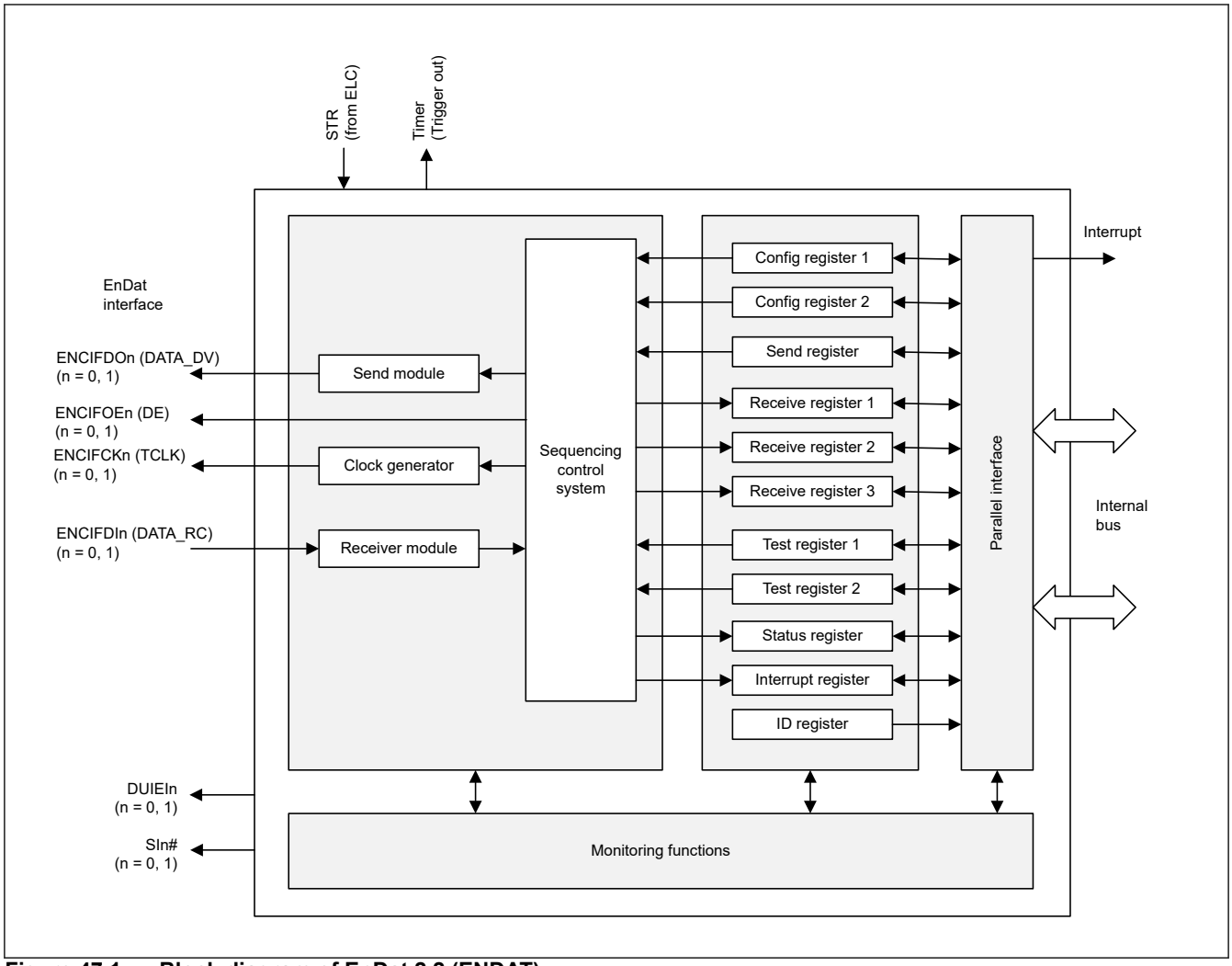


Figure 47.1 Block diagram of EnDat 2.2 (ENDAT)

Table 47.2 lists the input/output pins of the ENDAT.

Table 47.2 Pin configuration of the ENDAT

Unit	Pin name	I/O	Function
ENDAT0	ENCIFOE0 (DE)	Output	EnDat 2.2 unit 0 drive enable
	ENCIFDO0 (DATA_DV)	Output	EnDat 2.2 unit 0 data output (drive)
	ENCIFCK0 (TCLK)	Output	EnDat 2.2 unit 0 clock generator output
	ENCIFDI0 (DATA_RC)	Input	EnDat 2.2 unit 0 data input (receive)
	DUEI0	Output	EnDat 2.2 unit 0 data transfer
	TST_OUT0	Output	EnDat 2.2 unit 0 data input after internal synchronization
	SI0#	Output	EnDat 2.2 unit 0 start pulse
ENDAT1	ENCIFOE1 (DE)	Output	EnDat 2.2 unit 1 drive enable
	ENCIFDO1 (DATA_DV)	Output	EnDat 2.2 unit 1 data output (drive)
	ENCIFCK1 (TCLK)	Output	EnDat 2.2 unit 1 clock generator output
	ENCIFDI1 (DATA_RC)	Input	EnDat 2.2 unit 1 data input (receive)
	DUEI1	Output	EnDat 2.2 unit 1 data transfer
	TST_OUT1	Output	EnDat 2.2 unit 1 data input after internal synchronization
	SI1#	Output	EnDat 2.2 unit 1 start pulse

Table 47.3 ENDAT interrupt sources

Name	Interrupt sources	CPU GIC request	DMAC activation	ELC activation
ENDATn_INT1	Interrupt output 1	Possible	Possible	Not possible

Note: n = 0, 1

47.2 Register Descriptions

All register access operation should be executed with 32-bit access. It is not possible to write to or read from a single byte or word of a register. Unused bits are filled with zeros.

Table 47.4 ENDAT register map

Offset address	Register symbol	Register name
0x00	SEND	Send Register
0x04	EMPFR1_L	Receive Register 1, Low Double Word
0x08	EMPFR1_H	Receive Register 1, High Double Word
0x0C	EMPFR2	Receive Register 2
0x10	EMPFR3	Receive Register 3
0x14	KONFR1	Configuration Register 1
0x18	KONFR2	Configuration Register 2
0x20	STATR	Status Register
0x24	INTMR	Interrupt Mask Register
0x38	SWSTRBR	SW Strobe Register
0x3C	IDR	ID Register

47.2.1 SEND : Send Register

Base address: ENDATm = 0x9004_3000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x00

Bit position:	31		29			24	23			16	15																		0
Bit field:	—	—	M[5:0]				A[7:0]				D[15:0]																		
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	D[15:0]	Parameters, Instructions See Table 47.5.	R/W
23:16	A[7:0]	Memory Range Select (MRS) code, Address, Port address See Table 47.5.	R/W
29:24	M[5:0]	Mode command See Table 47.5.	R/W
31:30	—	These bits are read as 0. The write value should be 0.	R/W

The SEND register contains data to be transmitted to the EnDat encoder.

Table 47.5 shows an overview of the supported mode commands with data to be entered into the send register.

Table 47.5 is list of the EnDat master side, however, some commands may not be supported depending on the EnDat encoder that is the EnDat slave side.

Table 47.5 Mode commands

Mode command	M[5:0]	A[7:0]	D[15:0]
Encoder transmit position values	0x07	(Reserved)	(Reserved)
Encoder transmit position values with additional information	0x38	(Reserved)	(Reserved)
Selection of memory area	0x0E	MRS code	(Reserved)
Encoder receive parameters	0x1C	Address	Parameters
Encoder transmit parameters	0x23	Address	(Reserved)
Encoder receive reset	0x2A	Address	(Reserved)
Encoder transmit test values	0x15	(Reserved)	(Reserved)
Encoder receive communication command	0x12	Address	Instruction
Encoder receive test command	0x31	Port address	(Reserved)
Encoder transmit position values with additional information and selection of the memory area	0x09	MRS code	(Reserved)
Encoder transmit position values with additional information and receive parameters	0x1B	Address	Parameters
Encoder transmit position values with additional information and parameters	0x24	Address	(Reserved)
Encoder transmit position values with additional information and receive error reset	0x2D	Address	(Reserved)
Encoder transmit position values with additional information and receive test command	0x36	Port address	(Reserved)

Table 47.6 MRS code (1 of 2)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Request
Additional information 1								
0	1	0	0	0	0	0	0	Send additional info 1 without data contents (NOP)
0	1	0	0	0	0	0	1	Send diagnostic value
0	1	0	0	0	0	1	0	Send position value 2 word 1 LSB
0	1	0	0	0	0	1	1	Send position value 2 word 2
0	1	0	0	0	1	0	0	Send position value 2 word 3 MSB
0	1	0	0	0	1	0	1	Acknowledge memory content LSB
0	1	0	0	0	1	1	0	Acknowledge memory content MSB
0	1	0	0	0	1	1	1	Acknowledge MRS code
0	1	0	0	1	0	0	0	Acknowledge test command
0	1	0	0	1	0	0	1	Send test values word 1 LSB
0	1	0	0	1	0	1	0	Send test values word 2
0	1	0	0	1	0	1	1	Send test values word 3 MSB
0	1	0	0	1	1	0	0	Send temperature 1
0	1	0	0	1	1	0	1	Send temperature 2
0	1	0	0	1	1	1	0	Additional sensors
0	1	0	0	1	1	1	1	Stop sending additional information 1
Additional information 2								
0	1	0	1	0	0	0	0	Send additional information 2 without data contents
0	1	0	1	0	0	0	1	Send commutation
0	1	0	1	0	0	1	0	Send acceleration
0	1	0	1	0	0	1	1	Send commutation and acceleration
0	1	0	1	0	1	0	0	Send limit position signals
0	1	0	1	0	1	0	1	Send limit position signals & acceleration

Table 47.6 MRS code (2 of 2)

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Request
0	1	0	1	0	1	1	0	Asynchronous position value word 1 LSB
0	1	0	1	0	1	1	1	Asynchronous position value word 2
0	1	0	1	1	0	0	0	Asynchronous position value word 3 MSB
0	1	0	1	1	0	0	1	Operating status error sources
0	1	0	1	—	—	—	—	Not assigned
0	1	0	1	1	1	1	1	Stop sending additional information 2

Note: Data transfer from the send register to the send module occurs only after a completed write cycle and completed transmission cycle to/from encoder.

Only values for the mode commands defined above are entered into the send register. Only EnDat2.2 mode commands are entered in transmission mode 1 (safety mode). This ensures that the most recently entered valid mode command is active if an attempt is made to enter an invalid mode command into the send register.

47.2.2 Receive Registers

The transfer of data from the receive module to the corresponding receive registers occurs at the same time as related status information is updated, directly after the serial-to-parallel conversion.

47.2.2.1 EMPFR1_L : Receive Register 1, Low Double Word

Base address: ENDATm = 0x9004_3000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x04

Bit position:	31																													0
Bit field:	RDATA_L[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RDATA_L[31:0]	Received data, bit 31 to bit 0 See Table 47.7.	R

47.2.2.2 EMPFR1_H : Receive Register 1, High Double Word

Base address: ENDATm = 0x9004_3000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x08

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	POS1	IF2	F1	CRC[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	RDATA_H[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	RDATA_H[15:0]	Received data, bit 47 to bit 32 See Table 47.7.	R
20:16	CRC[4:0]	Received CRC	R

Bit	Symbol	Function	R/W
21	F1	Safety-relevant error bit F1 0: No error occurred 1: Error occurred or if forced dynamic sampling is active	R
22	IF2	Safety-relevant error bit F2 inversion 0: Error occurred or if forced dynamic sampling is active 1: No error occurred	R
23	POS1	Absolute position value 1 This bit marks Pos1, to which the following additional information with Pos2 belongs. 0: No Pos1 1: Pos1	R
31:24	—	These bits are read as 0.	R

Depending on the transmitted type 2.1 mode command, EMPFR1_L/EMPFR1_H registers contain different data. With EnDat type 2.2 mode commands, the position value is always entered into FMPFR1_L/EMPFR1_H.

Table 47.7 Received data

Entry	RDATA_H[15:8]	RDATA_H[7:0]	RDATA_L[31:24]	RDATA_L[23:16]	RDATA_L[15:8]	RDATA_L[7:0]
EnDat position value	MSB*3	…	…	…	…	D[7:0]
MRS code*1	(Invalid)	(Invalid)	(Invalid)	MRS code	D[15:8]	D[7:0]
Address and parameter*1	(Invalid)	(Invalid)	(Invalid)	Address	D[15:8]	D[7:0]
Port address and data*1	(Invalid)	(Invalid)	(Invalid)	Port Address	D[15:8]	D[7:0]
Test values*1	(Invalid)	DV[39:32]	DV[31:24]	DV[23:16]	DV[15:8]	DV[7:0]
Address and instruction*2	(Invalid)	(Invalid)	(Invalid)	Address	D[15:8]	D[7:0]

Note 1. Only for type 2.1 EnDat mode commands
Note 2. Only for EnDat mode command 010010
Note 3. The position of the MSB shifts depending on the length of the position value. The bits to left of the MSB are not part of the position value.

47.2.2.3 EMPFR2 : Receive Register 2

Base address: ENDATm = 0x9004_3000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0C

Bit position:	31																													0
Bit field:	RDATA2[31:0]																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	RDATA2[31:0]	Received data 2 Additional information 2 See Table 47.8.	R

If a type 2.2 mode command was sent, EMPFR2 register will contain the contents of additional information 2 and its CRC.

Please note that the Additional information 2 is supported is difference depending on the EnDat encoder.

Table 47.8 Received data 2

Entry	RDATA2[31:24]	RDATA2[23:16]	RDATA2[15:8]	RDATA2[7:0]
Additional information 2*1	CRC*2	MRS code*3	D[15:8]	D[7:0]

Note 1. Only with type 2.2 mode commands; otherwise the register will not be updated.

Note 2. CRC: bit 28 to bit 24
Note 3. If the EnDat encoder does not support the Additional information 2, the MRS code may be 0x5F.

47.2.2.4 EMPFR3 : Receive Register 3

Base address: ENDATm = 0x9004_3000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x10

Bit position:	31			28				24	23							16	15													0
Bit field:	—	—	—	CRC[4:0]					C[7:0]								D[15:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	D[15:0]	Data in the additional information 1*1	R
23:16	C[7:0]	Memory Range Select (MRS) code in the additional information 1*1 *2	R
28:24	CRC[4:0]	CRC for additional information 1*1	R
31:29	—	These bits are read as 0.	R

Note 1. Only with type 2.2 mode commands; otherwise, the register will not be updated.
Note 2. If the EnDat encoder does not support the Additional information 1, the MRS code may be 0x4F.

If a type 2.2 mode command was sent, EMPFR3 register will contain the contents of additional information 1 and its CRC.

Please note that the Additional information 1 is supported is difference depending on the EnDat encoder.

47.2.3 Configuration Registers

Setting required for operation and for handling of the corresponding protocols must be made in the configuration registers.

All of the configuration registers can only be written to if no EnDat transmission is active. The bit 30 of STATR register can be evaluated for this purpose.

47.2.3.1 KONFR1 : Configuration Register 1

Base address: ENDATm = 0x9004_3000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x14

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	SET[1:0]		ICRST	FSYS[2:0]			—	DELAYCMP	CPTIME[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	AUTORST	RSTWD	DWLG[5:0]						FTCLK[3:0]				—	DT	DU	HWSTRB
Value after reset:	0	0	0	0	1	1	0	1	1	1	1	1	0	0	0	0

Bit	Symbol	Function	R/W
0	HWSTRB	Hardware strobe Enable external / STR signal as strobe signal. 0: HW strobe not enabled 1: HW strobe enabled	R/W
1	DU	DU Defines the unconditional data transfer to receive registers 1, 2, 3, and 4 on completion of a data transmission process, despite a flag being set in the status register. For safety applications, DU = 1 must be set. 0: Data transfer according to flag set in the status register 1: Data transfer despite the flag in the status register	R/W

Bit	Symbol	Function	R/W
2	DT	DT Select the EnDat continuous clock mode. 0: Continuous clock off 1: Continuous clock on	R/W
3	—	This bit is read as 0. The write value should be 0.	R/W
7:4	FTCLK[3:0]	f_{TCLK} Setting (4 bits) of transmission rate for EnDat from 100 kHz to 16 MHz (EnDat) 0x0: 16.67 MHz 0x1: 16.67 MHz 0x2: 16.67 MHz 0x3: 16.67 MHz 0x6: 8.33 MHz 0xB: 4.16 MHz 0xC: 2 MHz 0xD: 1 MHz 0xE: 200 kHz 0xF: 100 kHz Others: Not permitted	R/W
13:8	DWLG[5:0]	Data word length*1 Data word length is set binary with 6 bits for EnDat, is the sum of MT (multi-turn bits) and ST (single-turn bits). The permissible setting range for EnDat is from 8 bits to 48 bits. 0x00-0x07: Not permitted 0x08: Data word length = 8 bits ⋮ 0x0D: Data word length = 13 bits ⋮ 0x30: Data word length = 48 bits 0x31-0x3F: Not permitted	R/W
14	RSTWD	Reset window The set bit allows resetting of the status and error register only within a defined time period. For safety applications: Reset window = 1 0: Resetting of the registers mentioned above can be performed at any time. (i.e. without considering malfunctions). 1: Resetting of the registers mentioned above must be performed within a defined time period for acceptance by the protocol engine.	R/W
15	AUTORST	Automatic reset If this bit is set, resetting of the status register and error register is performed automatically. For safety applications: Automatic reset = 0 0: Resetting of the above-mentioned registers must be performed by the application. 1: Resetting of the above-mentioned registers is done automatically. However, this resetting only occurs in the next EnDat transmission with the start of data reception.	R/W
23:16	CPTIME[7:0]	Cable propagation time The cable propagation time determined is stored. The application may change this value. If that is the case the status register's propagation time measurement (LZM) bit will automatically be reset. The binary value has a step width of one system clock. At a system clock of 100 MHz, this corresponds to a setting range from 0 µs to 2.55 µs in steps of 10.0 ns.	R/W
24	DELAYCMP	Delay compensation This bit switches propagation delay compensation on. When this bit is set, propagation time measurement is performed with the next data transmission to the EnDat encoder. The ENDAT determines the cable propagation time and saves this in KONFR1 register. This value is used to determine propagation delay compensation. To measure the propagation time again, the delay compensation bit must be reset and set again. For 16-bit access, it must be considered that the measured cable propagation time value is overwritten with 0x00. 0: Delay compensation off 1: Delay compensation on	R/W
25	—	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
28:26	FSYS[2:0]	System clock frequency (f_{sys}) The system frequency actually used must be selected. 1 1 0: 100 MHz Others: Not permitted	R/W
29	ICRST	IC reset Setting of this bit has the effect that the entire ENDAT is reset to its initial state. 0: IC reset inactive 1: IC reset active	R/W
31:30	SET[1:0]	EnDat set These two bits set the EnDat transmission mode. 0 0: Not permitted 0 1: Not permitted 1 0: EnDat 1 1: Not permitted	R/W

Note 1. The data word length has to set to 40/d bit while using mode command "encoder transmit test values".

The KONFR1 register specifies the basic settings and EnDat functions.

47.2.3.2 KONFR2 : Configuration Register 2

Base address: ENDATm = 0x9004_3000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x18

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	HWSTRBDLY[7:0]								—	—	FILTER[2:0]			RCVTIME[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WTDOG[7:0]								TMSMPLRATE[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	TMSMPLRATE[7:0]	Timer for sampling rate 256 different sampling rates can be set. For details, see section 47.3.4. Timer for Sampling Rate. In the default setting 0x00 or 0x80, the timer is off.	R/W
15:8	WTDOG[7:0]	Watchdog 256 different watchdog time values can be set. For details, see section 47.3.5. Watchdog Timer. In the default setting 0x00 or 0x80, the watchdog is off.	R/W
18:16	RCVTIME[2:0]	Recovery time III t_{ST} This time is to be set in accordance with EnDat specification. The set time has an accuracy of ±0.1 µs. 0 0 0: 1/2 TCLK 0 0 1: 0.5 µs 0 1 0: 1 µs 0 1 1: 1.5 µs 1 0 0: 2 µs 1 0 1: 4 µs 1 1 0: 8 µs 1 1 1: 10 µs	R/W

Bit	Symbol	Function	R/W
21:19	FILTER[2:0]	Filter This digital filter for the Data_RC data input can be adjusted in eight steps (3 bits) as below. The filter setting value corresponds to system clock cycles.*1 0 0 0: Off 0 0 1: 3 cycles 0 1 0: 4 cycles 0 1 1: 5 cycles 1 0 0: 6 cycles 1 0 1: 10 cycles 1 1 0: 20 cycles 1 1 1: 40 cycles	R/W
23:22	—	These bits are read as 0. The write value should be 0.	R/W
31:24	HWSTRBDLY[7:0]	Hardware strobe delay*2 The application can enter a value for the HW strobe delay. The binary value has a step width of one system clock. 0x00: Off 0x01: Not permitted 0x02: Not permitted 0x03: 3 cycles 0x04: 4 cycles 0x05: 5 cycles ⋮ 0xFF: 255 cycles	R/W

Note 1. The filter must be set according to the transmission rate of the serial interface to the encoder.
Example: f_{TCLK} = 1 MHz (corresponds to 100 system clock cycles with f_{CLK} = 100 MHz)
For the filter, 1/10 of the f_{CLK} must be set. That means 10 system clock cycles → setting: 101b.

Note 2. At a system clock of 100 MHz, this corresponds to a value range from 30 ns to 2.55 µs in steps of 10 ns.

The KONFR2 register specifies protocol-independent settings.

47.2.4 STATR : Status Register

Base address: ENDATm = 0x9004_3000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x20

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RDY	RDY4 STRB	—	—	—	—	—	—	LZM	LZK	—	—	—	FTYPII I	WTDG	SPIKE
Value after reset:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WRN	RM	BUSY	CRCZI 2	CRCZI 1	IERR2	EMPF R3	EMPF R2	—	—	MRS_ ADR	FTYPII	FTYPI	CRC_ PRTY	ERR1	EMPF R1
Value after reset:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EMPFR1	EMPFR1 register update This status flag indicates that the data in EMPFR1 register has been updated. This flag is ignored if the DU bit in KONFR1 register is enabled. 0: EMPFR1 register not updated 1: EMPFR1 register updated	R/W
1	ERR1	Error 1 The status bit "error 1" from the EnDat protocol is entered. 0: Error1 did not occur 1: Error1 occurred	R/W
2	CRC_PRTY	CRC-PW / parity With EnDat transmission it represents the result of the CRC check of the received value (position value, parameter or test value). 0: CRC check okay 1: CRC check faulty	R/W

Bit	Symbol	Function	R/W
3	FTYPI	F Type I Shows type I error handling in accordance with the EnDat specification at Annex A2. 0: A type I error did not occur 1: A type I error occurred	R/W
4	FTYPII	F Type II Shows type II error handling in accordance with the EnDat specification at Annex A2. 0: A Type II error did not occur 1: A Type II error occurred	R/W
5	MRS_ADR	MRS/Adr The occurrence of an addressing or acknowledgment error is shown. The errors (F type I / II) are special cases of MRS/address errors, i.e. they are a sub-quality of these. Accordingly, whenever a type I or type II error is identified, the MRS/Adr bit is set. For example, if an MRS/address bit is recognized incorrectly due to a disturbance, only the MRS/Adr status bit will be set, not the F TYP I/II bits. 0: No acknowledgment or addressing error has occurred 1: An acknowledgment or addressing error has occurred	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	EMPFR2	EMPFR2 register update This status flag indicates that the data in EMPFR2 register has been updated. It must be cleared after EMPFR2 register has been read to allow the ENDAT to rewrite data. 0: EMPFR2 not updated 1: EMPFR2 updated	R/W
9	EMPFR3	EMPFR3 register update This status flag indicates that the data in EMPFR3 register has been updated. It must be cleared after EMPFR3 register has been read to allow the ENDAT to rewrite data. 0: EMPFR3 not updated 1: EMPFR3 updated	R/W
10	IERR2	/Error2 Contains the "Error 2" status bit from the EnDat protocol (only with EnDat2.2 commands). 0: /Error2 occurred 1: /Error2 did not occur	R/W
11	CRCZI1	CRC-ZI1 During EnDat transmissions, this bit indicates the result of the CRC checking of additional information (ZI) 1. 0: CRC check of ZI1 okay 1: CRC check of ZI1 faulty	R/W
12	CRCZI2	CRC-ZI2 During EnDat transmissions, this bit indicates the result of the CRC checking of additional information (ZI) 2. 0: CRC check of ZI2 okay 1: CRC check of ZI2 faulty	R/W
13	BUSY	Busy Contains the "Busy" status bit as transmitted in the EnDat protocol. 0: Busy = 0 1: Busy = 1	R/W
14	RM	RM Contains the "RM" status bit as transmitted in the EnDat protocol. 0: RM = 0 1: RM = 1	R/W
15	WRN	WRN Contains the "WRN" status bit as transmitted in the EnDat protocol. 0: WRN = 0 1: WRN = 1	R/W
16	SPIKE	Spike Reports that a "Spike" was detected at the data input port. Condition: filter in the KONFR1 register is set. 0: No spike 1: Spike occurred	R/W

Bit	Symbol	Function	R/W
17	WTDG	Watchdog Reports triggering of the watchdog. Condition: watchdog in KONFR2 register is set. 0: Watchdog not triggered 1: Watchdog triggered	R/W
18	FTYPIII	F Type III Type II error (transmission layer) triggers F type III. Error recognition occurs in the EnDat master. 0: The error did not occur 1: The error occurred	R/W
21:19	—	These bits are read as 0. The write value should be 0.	R/W
22	LZK*1	Delay compensation This bit reports if propagation delay compensation is active. If propagation delay compensation in KONFR1 register is switched off, this bit and propagation time measurement will automatically be reset. 0: LZK inactive 1: LZK active	R/W
23	LZM*1	Propagation time measurement This bit reports that propagation time measurement was successfully completed. Condition: propagation delay compensation (LZK) in KONFR1 register is set. 0: LZM incomplete 1: LZM complete	R/W
29:24	—	These bits are read as 0. The write value should be 0.	R/W
30	RDY4STRB	Ready for strobe This bit reports that data transmission has ended and that the EnDat protocol automation machine is ready for the next transmission. The time values "Recovery time 1" (t_m) and "Recovery time 2" (t_R) are completed. This bit cannot be reset by writing a "1" as this is a status display of the current conditions of internal automation engines. The bit cannot cause an interrupt. 0: No ready 1: Ready	R/W
31	RDY	Ready If the ready bit is set, the status register is completely updated. All checks have been performed. Data transmission is not yet completed, however, meaning that the EnDat protocol automation machine is not yet ready again. 0: No ready 1: Ready	R/W

Note 1. Neither the LZM nor the LZK bit can be reset by writing a "1" to the respective bit as this is a status display of the current conditions of internal automation engines. Neither of the two bits can cause an interrupt.

The status bits are created by the sequencing controller, as required. Status information remains set until it is reset by the application.

The application can selectively reset status information with a write command. This occurs by writing "1" to the selected bits. In the event of concurrent access, the internal sequencing controller has priority. This ensures that status information is not 'lost'.

The bit 15 to bit 11 in STATR register are only valid when additional information 1 or 2 has been received.

Note: The status register should be read after each data transmission. It provides information about validity of the data contained in the receive registers. The status bits must be reset in order that the internal sequencing controller can recognize a renewed setting of the status bit.

Note: Each bit (except for LZM, LZK, RDY4STRB) can trigger an interrupt. Masking is performed with the INTMR register. If a bit that has been set (and thus has triggered an interrupt) is reset and if no other bit has triggered an interrupt, the interrupt is deasserted.

47.2.5 INTMR : Interrupt Mask Register

Base address: ENDATm = 0x9004_3000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x24

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	RDY	—	—	—	—	—	—	—	—	—	—	—	—	FTYPII I	WTDG	SPIKE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	WRN	RM	BUSY	CRCZI 2	CRCZI 1	IERR2	EMPF R3	EMPF R2	—	—	MRS_ ADR	FTYPII	FTYPI	CRC_ PRTY	ERR1	EMPF R1
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EMPFR1	Interrupt mask for EMPFR1 register update 0: Masked 1: Not masked	R/W
1	ERR1	Interrupt mask for Error 1 0: Masked 1: Not masked	R/W
2	CRC_PRTY	Interrupt mask for CRC-PW / parity 0: Masked 1: Not masked	R/W
3	FTYPI	Interrupt mask for F Type I 0: Masked 1: Not masked	R/W
4	FTYPII	Interrupt mask for F Type II 0: Masked 1: Not masked	R/W
5	MRS_ADR	Interrupt mask for MRS/Adr 0: Masked 1: Not masked	R/W
7:6	—	These bits are read as 0. The write value should be 0.	R/W
8	EMPFR2	Interrupt mask for EMPFR2 register update 0: Masked 1: Not masked	R/W
9	EMPFR3	Interrupt mask for EMPFR3 register update 0: Masked 1: Not masked	R/W
10	IERR2	Interrupt mask for /Error2 0: Masked 1: Not masked	R/W
11	CRCZI1	Interrupt mask for CRC-ZI1 0: Masked 1: Not masked	R/W
12	CRCZI2	Interrupt mask for CRC-ZI2 0: Masked 1: Not masked	R/W
13	BUSY	Interrupt mask for Busy 0: Masked 1: Not masked	R/W
14	RM	Interrupt mask for RM 0: Masked 1: Not masked	R/W

Bit	Symbol	Function	R/W
15	WRN	Interrupt mask for WRN 0: Masked 1: Not masked	R/W
16	SPIKE	Interrupt mask for Spike 0: Masked 1: Not masked	R/W
17	WTDG	Interrupt mask for Watchdog 0: Masked 1: Not masked	R/W
18	FTYPIII	Interrupt mask for F Type III 0: Masked 1: Not masked	R/W
29:19	—	These bits are read as 0. The write value should be 0.	R/W
30	—	This bit is read as 0. The write value should be 0.	R/W
31	RDY	Interrupt mask for Ready 0: Masked 1: Not masked	R/W

The INTMR register is for the masking of the STATR register's interrupt sources. All bits shown in the status register (except for LZM, LZK, RDY4STRB) can generate an interrupt. The bit assignments of the interrupt mask register are identical to those of the status register. An interrupt is allowed by setting the corresponding bit to 1.

47.2.6 SWSTRBR : SW Strobe Register

Base address: ENDATm = 0x9004_3000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x38

Bit position:	31																												0
Bit field:																													
Value after reset:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	SW strobe The application triggers the data transfer through a write command (set this register to 1). This option is used on applications that are not time-critical.	W

47.2.7 IDR : ID Register

Base address: ENDATm = 0x9004_3000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x3C

Bit position:	31																												0
Bit field:																													
Value after reset:	1	1	1	0	0	0	1	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1

Bit	Symbol	Function	R/W
31:0	n/a	EnDat specification ID	R

47.3 Operation

47.3.1 Control Sequence

Figure 47.2 shows the basic operating sequence of ENDAT. The flow charts are valid for EnDat transmission.

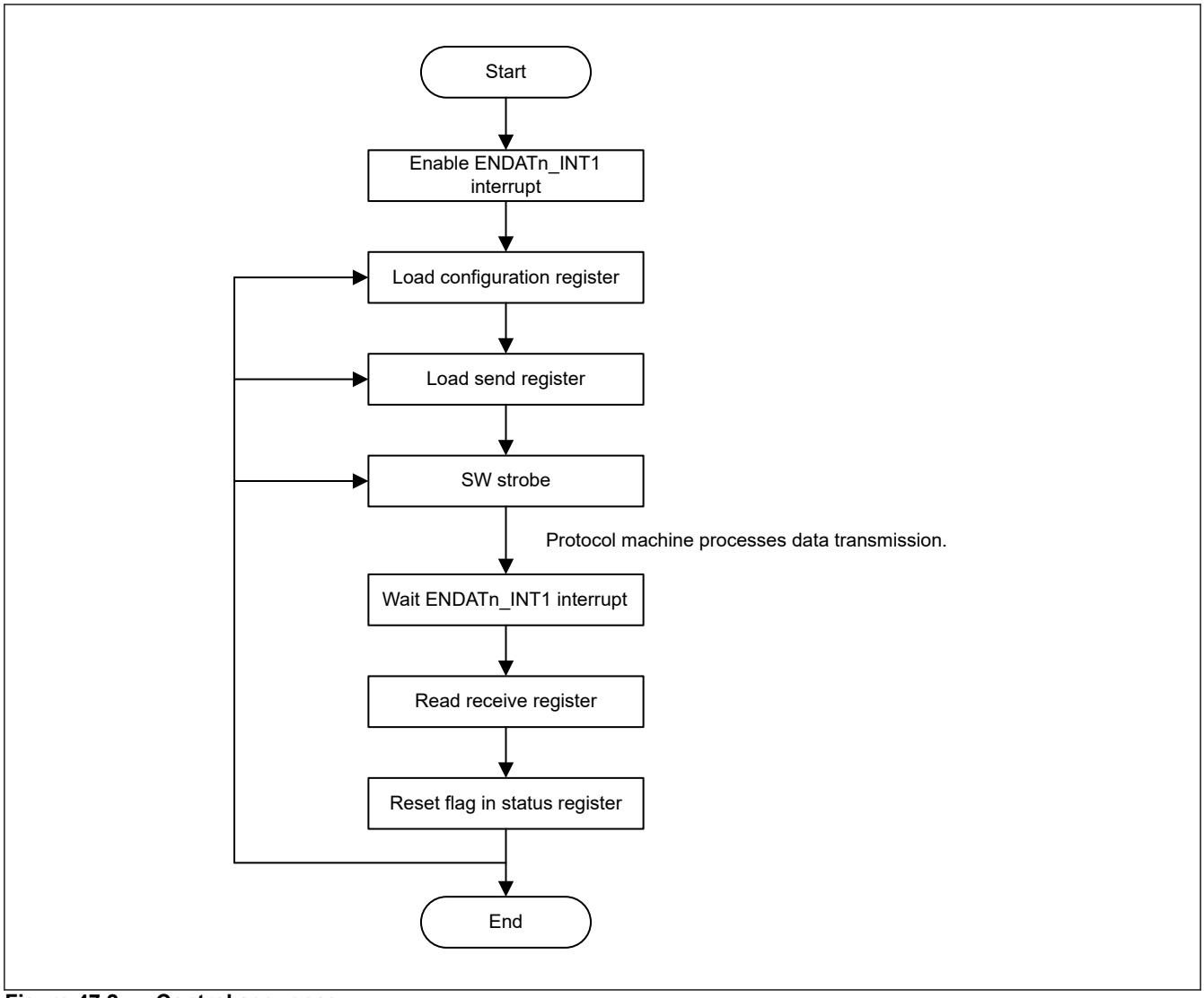


Figure 47.2 Control sequence

47.3.2 EnDat Data Transfer

EnDat is a bidirectional serial interface. The ENDAT starts transmission by sending a clock frequency with a defined length. Synchronously with the clock, a mode command (and other data, if required) is transmitted to the encoder. The encoder responds by sending the requested data to the subsequent electronics. The transmission direction can change multiple times during data transfer (half-duplex).

The EnDat protocol must be selected in the KONFR1 register.

47.3.2.1 Operating Sequence

Figure 47.3 shows the operating sequence of EnDat data transfer. The mode commands can be divided into three groups.

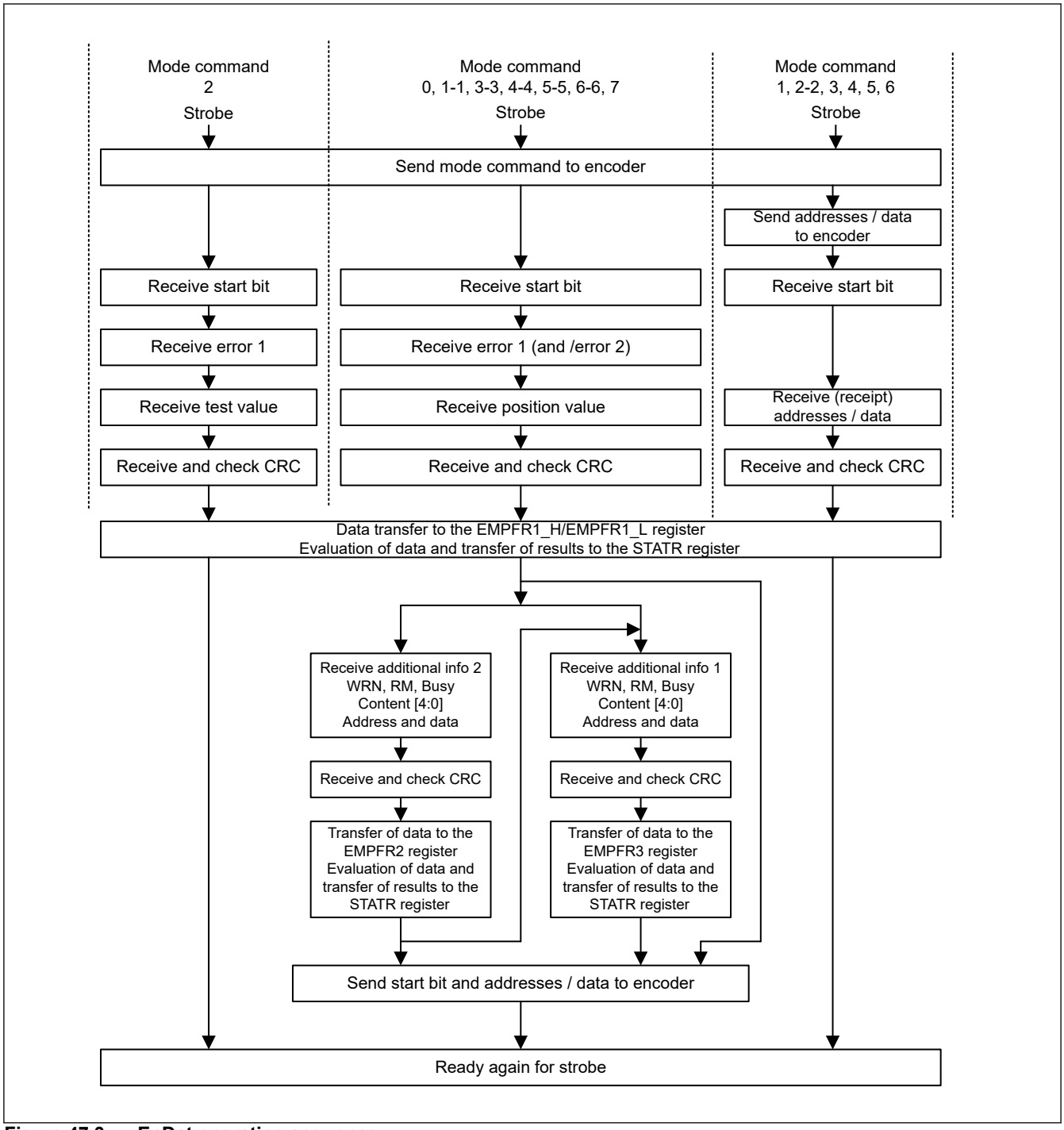


Figure 47.3 EnDat operating sequence

On mode commands for transfer of additional information (center block), the additional information previously selected by mode command 001001b and MRS code for selection (and deselection) of additional information determines the further progress of the protocol, i.e. whether and how many clocks will be sent to receive additional information.

47.3.2.2 Basic Functions

(1) Clock Generator

The clock generator creates the transmission clock required for the synchronous serial interface. The transmission frequency f_{TCLK} is set in the KONFR1 register. The pulse/break ratio specified for EnDat is complied with.

The clock generator also ensures that the number of clocks required for the respective mode commands is output. This is ensured during operation with and without delay compensation.

The number of clocks depends on the data word length and the mode command used as per EnDat specification.

(2) Data Word Length

The number of bits to be transferred for the position value must be set in the KONFR1 register.

(3) Continuous Clock

The ENDAT supports the possibility of executing a position value transfer with continuous clock using mode command 000111b. The settings are made in the KONFR1 register. (Default setting: Continuous clock = off)

Note: The updating rate can be very high. The EMPFR1_L/EMPFR1_H registers are not updated if the corresponding flag in the STATR register is not reset by the application. It is therefore recommended that the data transfer bit, DU, be set to 1 in the KONFR1 register.

(4) Error Handling

The ENDAT executes type I and type II error handling. Type III error handling is executed only for EnDat2.1 mode commands.

Note: This function is supported in the FR register. The results of type I and type II error handling are shown in the STATR register.

(5) Encoder Status

The status messaged from the encoder transported in the EnDat protocol are reflected to ERR1, IERR2, WRN, RM, and BUSY bits in the STATR register after every data transfer.

(6) Cyclical Redundancy Check (CRC)

The CRC bits received are compared with the CRC bits generated from the received data flow.

The CRC result of the position value and the data and addresses (with EnDat 2.1 commands) is saved in the STATR register under CRC-PW.

The CRC result of additional information 1 (2) is shown in the STATR register under CRC-ZI1 (2).

(7) Cable Propagation Time

At high transmission rates and with long cable lengths, the cable propagation time must be compensated for technical reasons.

The cable propagation time is measured during the next EnDat transmission[*1] if a corresponding entry is made in the KONFR1 register (cable propagation time = on). The propagation time measurement is carried out at a reduced transmission frequency (independent of the selected setting).

The measured value "cable propagation time" is saved in the KONFR1 register and is taken into account for subsequent transmissions.

After completion of the propagation time measurement, the bit "propagation time measurement completed" (LZM) in the STATR register is set to 1 and the bit "Delay compensation" (LZK) in the STATR register is set to 1.

If the measurement is to be repeated, the delay compensation bit (DELAYCMP) must be reset and then set again in the KONFR1 register. (For 16-bit access, it must be considered that the measured cable propagation time value in the KONFR1 register is overwritten with 0x00.)

In additional to that, the application can correct the (measured) cable propagation time by overwriting the propagation value in the KONFR1 register. In this case the propagation time measurement (LZM) display in the STATR register disappears (LZM to 0).

This function is supported with certified cables only.

Note 1. Use mode command 0 "Encoder transmit position values"

(8) Recovery Time II (Max. 500 ns)

During recover time the EnDat master does not permit starting an EnDat transmission. If delay compensation is enabled and the propagation time value entered in the control register is higher than or equals 500 ns, the master does not implement it additionally. In this case, the master is ready for a new transmission at 5 to 6 system clocks after the H/L edge on

DATA_RC. The bit 30, RDY4STRB, in the STATR register signals to the user that the EnDat system is ready for another transmission.

(9) Recovery Time III

At high transmission rates and with long cable lengths, recovery time III t_{ST} must, for technical reasons, be set in the KONFR2 register. (Default setting: 2 µs)

Note: The application can change the value for recovery time III t_{ST} if required. The time (t_{ST}) in the KONFR2 register should be greater than the measured propagation delay time plus 20 percent. (t_{ST} > 1.2 × propagation delay time)

47.3.2.3 Advanced Functions

(1) Watchdog

The watchdog allows to set a maximum value for the transmission duration. If this value is exceeded (e.g. due to an error), the transmission will be aborted and the EnDat sequencing controller is redefined to the outset. Register will not be reset, however. If a propagation time measurement or delay compensation was carried out before, it is retained and remains active. (This means that the delay compensation is not reset, either.)

The watchdog timer can be set in the KONFR2 register. If it is enabled, it starts when the EnDat transmission begins and stops when the EnDat transmission ends (EnDat engine idle). The status of the watchdog is shown in the STATR register.

The reaction of the watchdog on the interrupt output can be made visible by means of the interrupt mask. (Default setting = off)

The watchdog is ineffective with the following functions:

- Continuous clock
- Mode command 5 ("Encoder receive reset")
- Mode command 3 and 4 ("Encoder receive/transmit parameters")

(2) Reset with Selected Additional Information

If additional information was selected before a reset, it must be taken into consideration that this information is lost due to the reset. However, the measuring instrument keeps this information and still expects clock pulses for the additional information. For this reason, the EnDat2.1-Reset command must be given after the reset of the EnDat Master. (This EnDat command then deselects the additional information in the measuring instrument too.)

47.3.3 Interrupt

The ENDATn_INT1 interrupt output triggers status information at the output in accordance with INTMR register settings. (n = 0, 1)

47.3.4 Timer for Sampling Rate

Timer values for the sampling rate are set in the KONFR2 register.

Table 47.9 shows the programming values of sampling rate.

Two ranges are available for selection (microseconds and milliseconds).

Note: If the timer and the HW strobe delay are active at the same time, KONFR2[31:24], the timer output (Trigger signal), / timer, stays active, but transmission is only triggered with an H/L edge at the /STR input from ELC.

Table 47.9 Programming values of sampling rate (1 of 2)

µs: KONFR2[7] = 0		ms: KONFR2[7] = 1	
KONFR2[6:0]	**Sampling rate (µs)**	**KONFR2[6:0]**	**Sampling rate (ms)**
0x00	Stop	0x00	Stop
0x01	2	0x01	0.2
0x02	4	0x02	0.4

Table 47.9 Programming values of sampling rate (2 of 2)

µs: KONFR2[7] = 0		ms: KONFR2[7] = 1	
KONFR2[6:0]	**Sampling rate (µs)**	**KONFR2[6:0]**	**Sampling rate (ms)**
0x03	6	0x03	0.6
⋮	⋮	⋮	⋮
0x0A	20	0x0A	2.0
⋮	⋮	⋮	⋮
0x7F	254	0x7F	25.4

47.3.5 Watchdog Timer

Setting for the Watchdog is entered in the KONFR2 register.

Table 47.10 shows the programming values of Watchdog timer.

Two ranges are available for selection (microseconds and milliseconds).

Note: Accuracy: +2 µs or 0.2 ms. Depending on the range setting.

Table 47.10 Programming values of timer

µs: KONFR2[15] = 0		ms: KONFR2[15] = 1	
KONFR2[14:8]	**Timer (µs)**	**KONFR2[14:8]**	**Timer (ms)**
0x00	Stop	0x00	Stop
0x01	2	0x01	0.2
0x02	4	0x02	0.4
0x03	6	0x03	0.6
⋮	⋮	⋮	⋮
0x0A	20	0x0A	2.0
⋮	⋮	⋮	⋮
0x7F	254	0x7F	25.4

48. BiSS-C (BISS)

48.1 Overview

The MPU includes BiSS (BISS) master to enable communication between the encoder and BiSS-C interface. Table 48.1 describes the BiSS-C specifications and Figure 48.1 shows a block diagram of the BISS.

Table 48.1 BiSS-C (BISS) specifications

Item	Description
Number of units	2 units
Protocol	BiSS-C
Functions	• Bidirectional BiSS communication with up to 8 slaves • 1 physical BiSS channel each unit • Synchronous sensor data acquisition with cycle data transfers • Slave register operations during cyclic data transfers • Automatic compensation of line delays and conversion times • Data lengths of up to 64 bits for sensor data, configurable for each slave • Data verification by CRC polynomials of up to 16 bits per slave • Separate memory banks enable free controller access during BiSS sensor data transfers • 64 bytes memory for bidirectional register communication • Parallel interface with 32-bit data/address bus • Point-to-point connection (no bus connection: No MO pin)
Interrupt sources	• EOT: End-of-transmission • NER: Error output
Event link	GETSENS input for HW trigger provided from ELC
Module-stop function	Module-stop state can be set to reduce power consumption

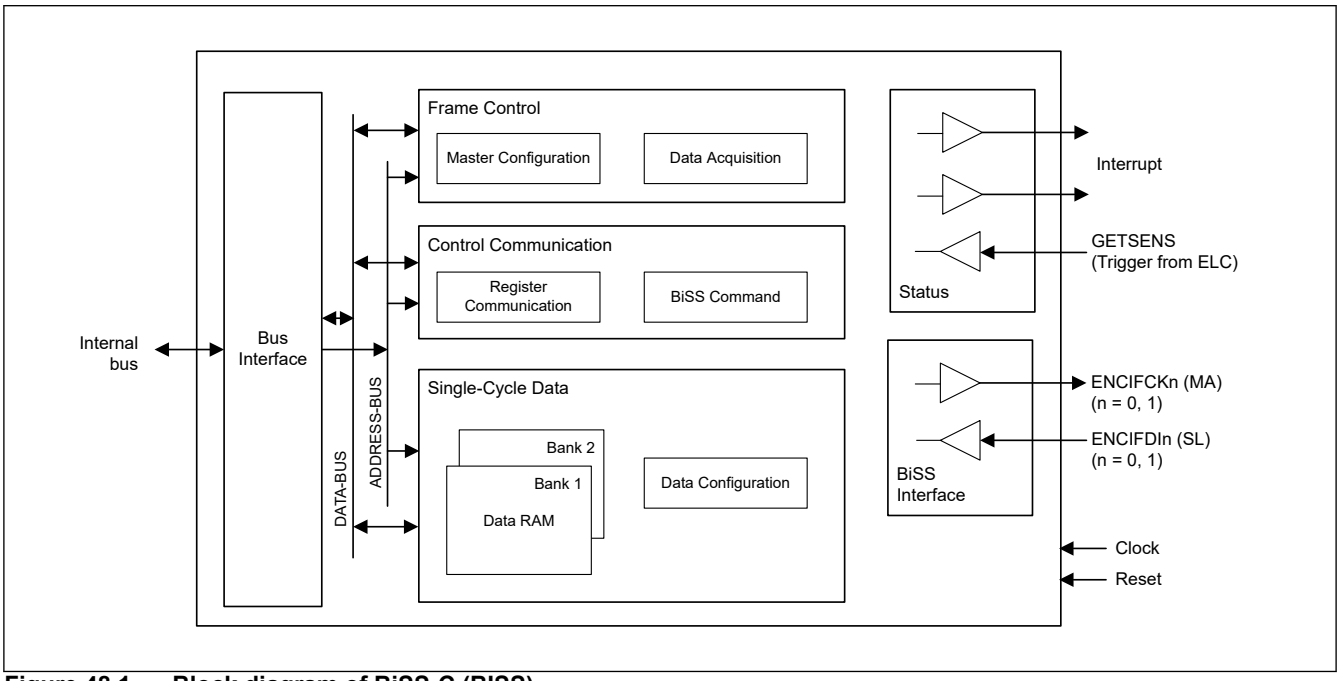


Figure 48.1 Block diagram of BiSS-C (BISS)

Table 48.2 PIN configuration of the BISS (1 of 2)

Unit	Pin name	I/O	Function
BISS0	ENCIFCK0 (MA)	Output	BiSS-C unit 0 clock output
	ENCIFDI0 (SL)	Input	BiSS-C unit 0 data input

Table 48.2 PIN configuration of the BISS (2 of 2)

Unit	Pin name	I/O	Function
BISS1	ENCIFCK1 (MA)	Output	BiSS-C unit 1 clock output
	ENCIFDI1 (SL)	Input	BiSS-C unit 1 data input

Table 48.3 BISS interrupt sources

Name	Interrupt sources	CPU GIC request	DMAC activation	ELC activation
BISSn_EOT	End-of-transmission interrupt	Possible	Possible	Not possible
BISSn_NER	Error interrupt	Possible*1	Not possible	Not possible

Note: n = 0, 1
Note 1. Some errors from other peripherals are ORed and connected to the GIC.

48.2 Register Descriptions

Table 48.4 BISS register map

Offset address	Register symbol	Register name
0x00 + 0x08 × (n - 1)	SCDATALn	Sensor Data L (n = 1 to 8)
0x04 + 0x08 × (n - 1)	SCDATAHn	Sensor Data H (n = 1 to 8)
0x80 + 0x04 × (n - 1)	RDATAn	Register Data n (n = 1 to 16)
0xC0 + 0x04 × (n - 1)	CNFSLVn	Configuration Slave n (n = 1 to 8)
0xE0	REGACC	Register Access
0xE4	CTRLCOMM	Control Communication
0xE8	MSTRCNF	Master Configuration
0xEC	CHCNF	Channel Configuration
0xF0	STINF	Status Information
0xF4	INSTREG	Instruction Register
0xF8	STINF2	Status Information 2

48.2.1 SCDATALn : Sensor Data L (n = 1 to 8)

Base address: BISSm = 0x9004_2000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x00 + 0x08 × (n - 1)

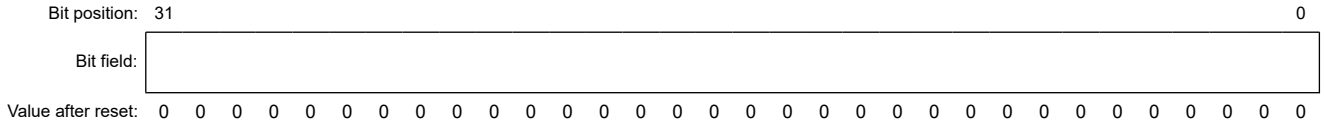
Bit position:	31																												0
Bit field:																													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	n/a	Sensor data lower for slave n This register stores the lower four bytes (SCDATAn[31:0]). For details, see section 48.3.3. Sensor Data.	R/W

48.2.2 SCDATAHn : Sensor Data H (n = 1 to 8)

Base address: BISSm = 0x9004_2000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x04 + 0x08 × (n - 1)

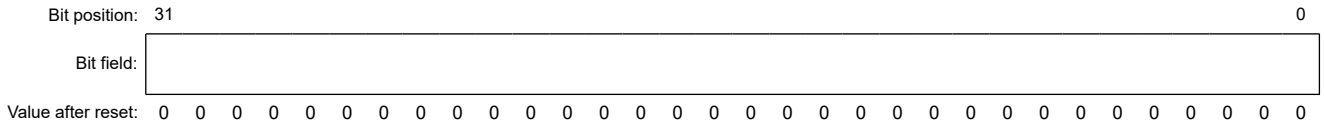


Bit	Symbol	Function	R/W
31:0	n/a	Sensor data upper for slave n This register stores the upper four bytes (SCDATAn[63:32]). For details, see section 48.3.3. Sensor Data.	R/W

48.2.3 RDATAn : Register Data n (n = 1 to 16)

Base address: BISSm = 0x9004_2000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x80 + 0x04 × (n - 1)

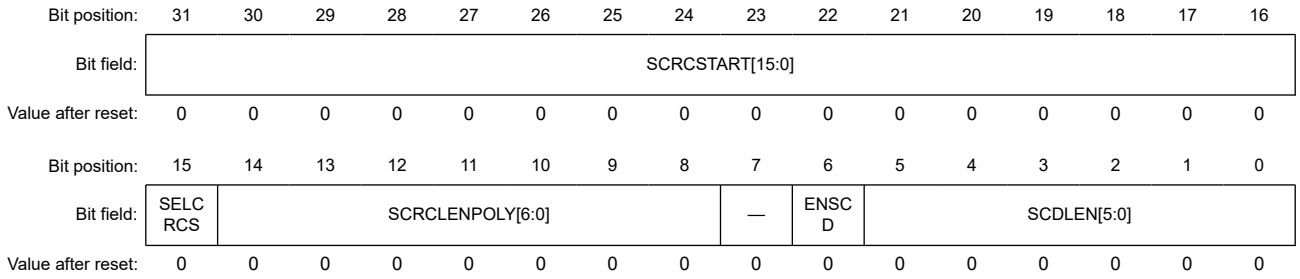


Bit	Symbol	Function	R/W
31:0	n/a	● Register communication data n (RDATAn) Using register communication in control communication (CTRLCOMM.CTS = 1). There is an individual 64 bytes storage area used to store the data of automatic register communication. In case of a register write access to a BiSS slave, RDATAn must be written with register data before starting the BiSS communication with INSTREG.INSTR. In case of a register read access to a BiSS slave, the received register data is stored in RDATAn after successful BiSS communication. In contradiction to SCDATAn there is only one memory bank for RDATAn. Therefore, after performing register communication, RDATAn may only be accessed, if no control communication is currently active (STINF.REGEND = 1). ● Slave addressing for BiSS commands (IDS[7:0]) Using BiSS commands in control communication (CTRLCOMM.CTS = 0). When 0x00: All slaves addressed (broadcast). The bits correspond to N. (For details, see section 48.3.4.1. BiSS Commands.) 0: Slave with ID (7 – N) is not addressed 1: Slave with ID (7 – N) is addressed	R/W

48.2.4 CNFSLVn : Configuration Slave n (n = 1 to 8)

Base address: BISSm = 0x9004_2000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0xC0 + 0x04 × (n - 1)



Bit	Symbol	Function	R/W
5:0	SCDLEN[5:0]	Single-cycle data length of slave n For more information about the length, refer to section 48.4.1. The value to be set to SCDLEN in the CNFSVn register and section 48.4.2. The storage of received CRC when SCDLEN = 33 bits or more. 0x00: Single-cycle data length 1 0x01: Single-cycle data length 2 ⋮ 0x3F: Single-cycle data length 64	R/W
6	ENSCD	Enable single-cycle data for slave n 0: Single-cycle data not available 1: Single-cycle data available	R/W
7	—	This bit is read as 0. The write value should be 0.	R/W
14:8	SCRCLENPOLY[6:0]	● CRC length with predefined CRC polynomial for slave n (SCRCLEN[6:0]) 0x00: CRC verification deactivated 0x03: 1011b = 0xB 0x04: 1_0011b = 0x13 0x05: 10_0101b = 0x25 0x06: 100_0011b = 0x43 0x07: 1000_1001b = 0x89 0x08: 1_0010_1111b = 0x12F 0x10: 1_1001_0000_1101_1001b = 0x190D9 Others: codes are not permitted ● CRC polynomial for slave n (SCRCPOLY[7:1]) Since the least significant bit of the CRC polynomial is always 1, there is no need to store this information in the CRC polynomial. Therefore, a maximum polynomial length of 8 bit is possible. If the maximum 8 bits polynomial length is not required, the used polynomial (without the LSB bit) is stored right aligned and leading bits are filled with 0. Example: The CRC polynomial 0x43 = 100 0011b is stored as 010 0001b (value 0x21). 0: CRC verification deactivated n (other than 0): n & 1	R/W
15	SELCRCS	CRC polynomial selection for slave n BiSS uses CRC polynomials depending on the data length predefined in the Electronic Data Sheet (EDS) and Profile Definitions. This bit chooses between predefined polynomials selected according to the CRC length with SCRCLEN and individual CRC polynomials SCRCPOLY. If SCRCLEN = 0, the CRC verification is deactivated. 0: Apply CRC polynomial as predefined in SCRCLEN 1: Apply CRC polynomial as individually defined in SCRCPOLY	R/W
31:16	SCRCSTART[15:0]	CRC calculation start value for slave n The CRC start values can be used to safely differentiate between single-cycle data (SCD) data words e.g., two position words provided by two different sensors. The same CRC start value that is used in the sensor needs to be applied in the master to validate the received single-cycle data. The CRC start value has a length of 16 bits. A typical default slave CRC start value is 0x0000.	R/W

48.2.5 REGACC : Register Access

Base address: BISSm = 0x9004_2000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0xE0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	REGNUM[5:0]						WNR	REGADR[6:0]							—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	—	The read values are undefined. The write value should be 0.	R/W
22:16	REGADR[6:0]	Register communication start address	R/W

Bit	Symbol	Function	R/W
23	WNR	Register communication direction 0: Read register data 1: Write register data	R/W
29:24	REGNUM[5:0]	Number of consecutive registers to access If REGNUM > 1, a sequential register access is performed. (See BiSS-C Protocol Description) 0x00: Register count 1 0x01: Register count 2 ⋮ 0x3F: Register count 64	R/W
31:30	—	The read values are undefined. The write value should be 0.	R/W

This register is used to configure the BiSS-C control communication for register communication. For details, see section 48.3.4.2. Register Access.

48.2.6 CTRLCOMM : Control Communication

Base address: BISSm = 0x9004_2000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0xE4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	NOCRC	SINGLEBANK	FREQR[2:0]			FREQS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	CTS	REGVERS	SID_CMD_IDT[2:0]			—	—	HOLDCDM	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	—	These bits are read as 0. The write value should be 0.	R/W
8	HOLDCDM	Behavior of MA signal at the end of frame In general operation, the BiSS master is configured according to the data of the connected BiSS slave(s) as described in CNFSLVn registers. A timeout period follows after the data transmission. And after the timeout period expires, the end of the data transmission is signaled to the BiSS master by a rising edge of SL. As soon as the timeout is detected by the BiSS master, it can start a new BiSS frame. If SL signal is not digital low at the end of the frame, e.g., because the BiSS-C frame has not been clocked out completely, MA signal can be programmed with HOLDCDM = 1 to be constant until the next frame begins. Thereby it is ensured that the slave can recognize CDM and take part in control communication, even if the master is not configured according to the data of the connected BiSS slave(s). 0: MA changes to digital high after detecting the slave's timeout termination at SL 1: MA remains constant until the next frame begins	R/W
10:9	—	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Function	R/W
13:11	SID_CMD_IDT[2:0]	● Slave addressing for register communication (SLAVEID[2:0]) Using register communication in control communication (CTS = 1). ● BiSS Command of addressed slave (CMD[1:0]) (Corresponds to SID_CMD_IDT[2:1] bits) Using BiSS commands in control communication (CTS = 0). CMD determines which BiSS command is sent. Some BiSS commands are predefined in the BiSS-C Protocol Description, others can be used for device-specific functionality of the slave. ● BiSS command execution control (IDA_TEST) (Corresponds to SID_CMD_IDT[0] bit) Using BiSS commands in control communication (CTS = 0). In contrast to the broadcast BiSS commands, the reception of an addressed BiSS command is confirmed during IDA (ID Acknowledge) by the addressed slave(s). This bit defines if the master triggers the command execution immediately or after checking the IDA bits (see Figure 48.7). The BiSS command is only executed if the check is successful. 0: Immediate execution 1: The slave's feedback (IDA) is tested before execution. The BiSS command is not executed if test fails.	R/W
14	REGVERS	Type of protocol for register communication Supports autonomous register communication with BiSS-C. REGVERS is used to select the protocol type. 0: Not permitted 1: Register communication BiSS-C	R/W
15	CTS	Type of control communication 0: BiSS Commands 1: Register communication	R/W
20:16	FREQS[4:0]	Single-cycle data clock frequency at MA (fMA) The master clock at MA signal is generated for BiSS from the system clock (fCLK) and depends on the frequency division by FREQS. With a system clock frequency of 20 MHz, the clock frequency at MA ranges from 10 MHz down to 62.5 kHz. Single-cycle data clock frequency (fMA): (N: Decimal number) 0x00: fCLK/2/(N + 1) ⋮ 0x0F: fCLK/2/(N + 1) 0x10: Not permitted 0x11: fCLK/20/(N - 15) ⋮ 0x1F: fCLK/20/(N - 15)	R/W
23:21	FREQR[2:0]	BiSS register data frequency*1 *2 BiSS slave device recognizes an idle bus at the end of a transmission frame via timeout detection. Thus, the choice of possible clock frequencies is limited as the duration of each logic level may not exceed the shortest timeout of all the connected slaves. 0 0 0: $fMA/2^{(N + 1)}$ ⋮ 1 1 1: $fMA/2^{(N + 1)}$	R/W
24	SINGLEBANK	Usage of one/two RAM bank(s) for SCDATA With SINGLEBANK = 0, two RAM banks are used for buffering SCDATA. Thus, it is ensured that one RAM bank can be written with single-cycle data received at the BiSS interface while the single-cycle data content of the other RAM bank is read. A downgrade is possible by setting SINGLEBANK = 1. 0: Two RAM banks are used for SCDATA 1: One RAM bank is used for SCDATA	R/W
25	NOCRC	Storage of received CRC in SCDATA When NOCRC = 1, the CRC is not updated to the SCDATA register, but it is not cleared to 0. Therefore, switching from NOCRC = 0 to NOCRC = 1 leaves the last data in the register. When clearing the CRC to 0, the SCDDATA register must be cleared according to the number of the RAM banks used. 0: CRC of SCD is stored in SCDATA 1: CRC of SCD is not stored in SCDATA	R/W
31:26	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. FREQR = 0x05 should be set, if CHCNF.CFGCH1 = 0x01.

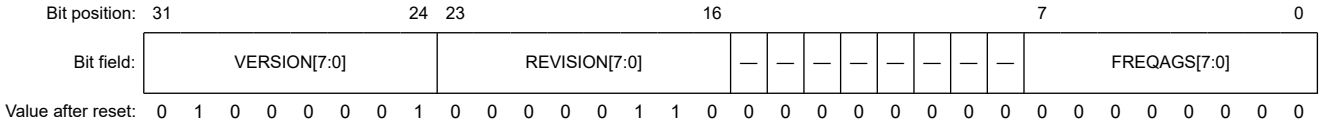
Note 2. BiSS-C devices generally permit a lower clock frequency. BiSS-C devices do not use a MA clock duty cycle (PWM signal) and can be operated down to 80 kHz.

The control communication enables protected and confirmed reading and writing of the registers of a slave (register communication). It is also used for protected and confirmed sending of BiSS commands to specified slaves or to all slaves. The control data is compounded of 1 bit per single-cycle data frame in each direction. The control data bit sent as part of the single-cycle data frame from master to the slaves is called CDM (Control Data Master) and the control data bit sent from a slave to the master is called CDS (Control Data Slave). For more information about the control communication refer to the BiSS-C Protocol Description.

48.2.7 MSTRCNF : Master Configuration

Base address: BISSm = 0x9004_2000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0xE8



Bit	Symbol	Function	R/W
7:0	FREQAGS[7:0]	Frame repetition rate (AGS frequency divider)*1 *2 FREQAGS controls the automatic data transmission (INSTREG.AGS). With FREQAGS the frame repetition rate can be set to a dedicated ratio of the system clock frequency (fCLK). With a system clock frequency of 20 MHz frame repetition rates from 1 µs to 4 ms are possible. FREQAGS must be set in a way that the time in between two frames is greater than a complete cycle. One cycle consists of the transmission of a request, an acknowledge signal (including line delay), a start bit (including processing time), a control data bit (BiSS-C), the sensor data and CRC bits of each slave and the maximum BiSS timeout of all connected slaves. FREQAGS is activated with INSTREG.AGS. (N: Decimal number, fCLK: system clock frequency) 0x00: fCLK/20/(N + 1) ⋮ 0x7B: fCLK/20/(N + 1) 0x7C: AGSMIN 0x7D: AGSINFINITE ⋮ 0x7F: AGSINFINITE 0x80: fCLK/625/(N - 127) ⋮ 0xFF: fCLK/625/(N - 127)	R/W
15:8	—	These bits are read as 0. The write value should be 0.	R/W
23:16	REVISION[7:0]	Design ID	R
31:24	VERSION[7:0]	Version	R

Note 1. AGSMIN: The master automatically restarts the next frame after the previous frame is finished. The master automatically generates the highest frame rate possible. AGSMIN requires complete BiSS frames to ensure a low level at SL at the end of each frame. The rate depends on the configured sensor data clock frequency (CTRLCOMM.FREQS), the slave configurations (e.g., CNFSLVn.SCDLEN), the maximum processing time of the slaves and the total system line delay.

Note 2. AGSINFINITE: The master does not automatically restart the next cycle. AGSINFINITE requires a trigger event to start the next frame. The next frame is started by a digital pulse at GETSENS signal or by setting INSTREG.INSTR.

48.2.8 CHCNF : Channel Configuration

Base address: BISSm = 0x9004_2000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0xEC

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	CFGCH1[1:0]		—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Function	R/W
0	—	This bit is read as 1. The write value should be reset value.	R/W
7:1	—	These bits are read as 0. The write value should be reset value.	R/W
9:8	CFGCH1[1:0]	Channel configuration 0 0: Not permitted 0 1: BiSS-C 1 0: Not permitted 1 1: Channel is not used (no device connected)	R/W
31:10	—	These bits are read as 0. The write value should be 0.	R/W

48.2.9 STINF : Status Information

Base address: BISSm = 0x9004_2000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0xF0

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	CDMTIMEOUT	CDSSEL	REGBYTES[5:0]						SVALID[15:8]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	SVALID[7:0]								nERR	nAGSERR	nDELAYERR	nSCDERR	nREGERR	REGEND	—	EOT
Value after reset:	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	1

Bit	Symbol	Function	R/W
0	EOT	End of transmission EOT flag is connected to EOT signal and signals a running frame (see section 48.3.2. BiSS-C). 0: Data transmission active 1: Data transmission not active	R
1	—	This bit is read as 0. The write value should be 0.	R/W
2	REGEND	End of Control Communication REGEND indicates a finished control communication. It is also set if the control communication was not successful. REGEND flag is reset in the following three patterns: ● After power on ● Control communication has not started after initialization by INSTREG.INIT ● Control communication running 0: Control communication running or not started since power on/INIT 1: Control communication completed/not running	R

Bit	Symbol	Function	R/W
3	nREGERR	Control communication error nREGERR flag is set after power on and after executing INSTREG.INIT. If a register data error is detected, the number of bytes transmitted correctly before the error occurred is provided by the register message REGBYTES. In case of an error, the transmission of data is terminated. The following issues can cause error indication: ● Read register access: CRC error or inverted R-bit ● Write register access error in repeated CDS data ● BiSS Commands: error in IDA 0: Error in last control communication 1: No error in last control communication	R
4	nSCDERR	Single-cycle data (SCD) transmission error An error in the single-cycle data detected by checksum verification (CRC) is shown with nSCDERR. If nSCDERR = 0, the faulty sensor can be identified by reading SVALID bits. 0: Error in last single-cycle data transmission 1: No error in last single-cycle data transmission	R
5	nDELAYERR	Start bit in register communication*1 nDELAYERR flag is set after power on and after executing INSTREG.INIT. nDELAYERR notifies about a missing start bit in last register communication. The addressed slave register might not be implemented. 0: Missing start bit in last register data transmission. Register address is not implemented or register access processing time too long (> 20 ms, if CTRLCOMM.FREQR = 0x05). 1: No missing start bit in last register data transmission	R
6	nAGSERR	Automatic Get Sensor (AGS) error When automatic transmission of sensor data is enabled with instruction bit INSTREG.AGS, an AGS watchdog error nAGSERR is indicated, if no new cycle could be initiated. If the last BiSS frame has not been finished in time, the next BiSS frame will be omitted. The following BiSS frame will be executed if possible. nAGSERR flag is also indicated, if a write access to INSTREG register is performed (e.g., by writing INSTREG.BREAK). 0: At least one BiSS frame has been omitted 1: No missing BiSS frames	R
7	nERR	State of the error interrupt (NER) nERR flag indicates the state of NER signal. Any of the errors is indicated. 0: Internal error occurred 1: No error occurred	R
23:8	SVALID[15:0]	SCDATAn validity indication (n: slave, n = 1 to 8)*2 Two bits are assigned to each slave, the upper bit is valid (SVALID[1]: SVALID1 for Slave 1, SVALID[3]: SVALID2 for Slave 2, SVALID[5]: SVALID3 for Slave 3, and so on). The lower bit is invalid, and the write value should be 0. The CRC verification result of the received single-cycle sensor data of every BiSS frame is written to the validity SVALID for each slave separately. If the CRC is disabled in the slave configuration, the correspondent SVALID flag is set after reading of sensor data has been completed. The validity flags are reset by writing to SVALID. 0: Single-cycle data invalid 1: Single-cycle data valid	R/W
29:24	REGBYTES[5:0]	Number of valid register bytes In case of an error in a sequential register communication (nREGERR = 0), the number of faultless transmitted register values are stored in REGBYTES. These bits are reset if an initialization sequence (INSTREG.INIT) is transmitted or a new control communication started.	R
30	CDSSEL	State of Control Data Slave (CDS)*3 0: CDS value 0 1: CDS value 1	R
31	CDMTIMEOUT	Control Data Master (CDM) timeout reached For BiSS-C control communication, a minimum number of SCD cycles with exclusively CDM = 0 must be sent before starting a new control communication frame. For a manual control communication by software, CDMTIMEOUT bit indicates that ≥ 14 SCD cycles with exclusively CDM = 0 have already been sent. 0: Timeout of BiSS-C control communication not reached 1: Timeout of BiSS-C control communication reached	R

Note 1. This controller can abort register communication after tbusy_r (as defined in BiSS-C Protocol Description).
Note 2. After reading sensor data, it is recommended to reset SVALID flags in order to recognize updated sensor data.

Note 3. There are two RAM banks for SVALID, CDSSEL, and STINF2.CDS1 that switch simultaneously to SCDATA RAM banks.

The status information is combined in a set of registers. It indicates all device and communication states of the master.

48.2.10 INSTREG : Instruction Register

Base address: BISSm = 0x9004_2000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0xF4

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	MAVS	MAFS	—	—	—	—	BREAK	HOLD BANK	SWBANK	INIT	INSTR[2:0]			AGS
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	AGS	Automatic Get Sensor Data (AGS) The automatic data transmission is enabled with AGS, and the frame rate is set with MSTRCNF.FREQAGS bits. For details, see section 48.3.5. Data Acquisition. FREQAGS = AGSMIN: Start of data transmission after slave's timeout FREQAGS = AGSINFINITE: Start of data transmission triggered by GETSENS FREQAGS = Other: Start of data transmission equally spaced at a constant rate 0: No automatic data transmission 1: Automatic data transmission	R/W
3:1	INSTR[2:0]	Single-Cycle Data (SCD) Control Instruction INSTR can be used to trigger data transmission by program. For details, see section 48.3.5. Data Acquisition. When AGS = 0: 0 0 0: Reset state. No INSTR performed. 0 0 1: Start one frame with CDM = 1. INSTR automatically resets. 0 1 0: Start one frame with CDM = 0. INSTR automatically resets. 0 1 1: Start one frame with CDM = not(CDSSEL). INSTR automatically resets. 1 0 0: Start one frame with control communication. INSTR automatically resets. 1 1 1: Start one reduced frame with control communication. INSTR automatically resets. When AGS = 1: 0 0 1: Upcoming frames with CDM = 1. 0 1 0: Upcoming frames with CDM = 0. 0 1 1: Upcoming frames with CDM = not(CDSSEL). 1 0 0: Upcoming frames with control communication. INSTR will be reset at the end of the control communication. 1 1 1: Upcoming reduced frames with control communication. INSTR will be reset at the end of the control communication.	R/W
4	INIT	Start initialization sequence*1 Typically, the BiSS data line SL is set to digital high by the last slave after timeout. After power on or after a failure, the last slave may not be defined, and the SL line is digital low. An initialization sequence started by INIT can be used to initialize the slave chain. Additionally, an INIT sets the status bits nSCDERR, nREGERR, nDELAYERR and nAGSERR in STINF register to 1 and resets REGEND to 0. CTRLCOMM.REGVERS is used to select the protocol type. 0: No operation 1: Initialize (REGVERS = 1: BiSS-C)	R/W
5	SWBANK	RAM Bank Switching By inverting SWBANK value a switch of the RAM banks is forced. This can be used, for instance, to initialize the content of both RAM banks. 0: No operation when Previous value = 0. RAM banks are switched when Previous value = 1. 1: RAM banks are switched when Previous value = 0. No operation when Previous value = 1.	R/W

Bit	Symbol	Function	R/W
6	HOLDBANK	RAM Bank Control When reading more than one Data RAM registers by the controller, the RAM banks in the master could be switched once a sensor data transmission via BiSS is completed. To avoid unexpected bank switching, the controller can set HOLDBANK before reading Data RAM registers and releases it afterwards. If bank switching was prevented with HOLDBANK = 1, the bank is switched after resetting HOLDBANK = 0. If a new BiSS frame is scheduled before HOLDBANK is reset (e.g., when using INSTREG.AGS with a high MSTRCNF.FREQAGS), no bank switching is performed but STINF.nAGSERR will be set. 0: Automatic bank switching is permitted 1: Automatic bank switching is prevented	R/W
7	BREAK	Start BREAK Sequence*2 All current actions can be aborted using the BREAK command so that the controller enters a defined state if one of the sensors does not respond correctly. 0: No operation 1: Abort data transmission	R/W
11:8	—	These bits are read as 0. The write value should be 0.	R/W
12	MAFS	Control of the MA line 0: MA line is not controlled 1: MA line is forced with MAVS	R/W
13	MAVS	MA line force level 0: Low 1: High	R/W
31:14	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. With REGVERS = 1 (BiSS-C) and an INIT, the master stores the measured line delay in SCDATA1[7:0] register. The unit of these values is 1/4 of the configured single-cycle data clock frequency configured with CTRLCOMM.FREQS.
tLine Delay = SCDATA1[7:0]/(4 × fMA)
For the INIT sequence, the maximum line delay is 255. If this limit is exceeded, the INIT sequence is aborted and STINF.nAGSERR is activated (nAGSERR = 0). The INIT sequence does not test the SL state. A constant SL = 0 state is ignored and does not set any error state.

Note 2. An INIT should be executed after BREAK to ensure resetting status bits.

48.2.11 STINF2 : Status Information 2

Base address: BISSm = 0x9004_2000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0xF8

Bit position:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit field:	—	—	—	—	—	—	—	SWBANKFAIL	—	—	—	—	IDL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit position:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CDS1	SL1
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
0	SL1	SL input line state The actual state of the BiSS data SL input. 0: SL line level low 1: SL line level high	R
1	CDS1	Control Data Slave (CDS) bit value This bit has the same contents as STINF.CDSSEL. 0: CDS value 0 1: CDS value 1	R
15:2	—	The read values are undefined.	R

Bit	Symbol	Function	R/W
19:16	IDL[3:0]	Number of set ID lock bits during control communication*1 At the beginning of each control communication, slave IDs are automatically assigned to the connected slaves by considering the physical order within the daisy chain. IDL indicates the total number of IDL bits that have been set by the BiSS slaves. It corresponds to the number of slaves that take part in control communication. IDL = 9 indicates that there are more than eight BiSS slaves connected that can take part in control communication. Since only 8 BiSS slaves are addressable via CTRLCOMM.SLAVEID, the first slave(s) cannot be addressed unless other slaves have been disabled via predefined BiSS commands as defined in BiSS-C Protocol Description. 0: No IDL bits received 1 to 8: 1 to 8 IDL bits (corresponds to 1…8 slaves) 9: There are more than 8 slaves available. Note, only the first 8 slave IDs are addressable with CTRLCOMM.SLAVEID.	R
23:20	—	The read values are undefined.	R
24	SWBANKFAIL	Bank switching status Switching SCDATA RAM bank—which is executed at the end of a BiSS frame—also switches the corresponding STINF.SVALID flags. It fails if it disables bank switching via INSTREG.HOLDBANK. 0: Bank switching (SCDATA) successful 1: Bank switching (SCDATA) not successful	R
31:25	—	These bits are read as 0.	R

Note 1. A control communication must be performed before the correct number of slaves is indicated with IDL.

48.3 Operation

48.3.1 BiSS Function

In point-to-point configuration, BiSS devices including one or several sensors are connected to the clock line MA and the data return line SL. RS422 transceivers are available on-chip for robust differential communication in industrial environments. Two types of BiSS communication are supported:

- Single-Cycle Data (SCD):

  Sensor data with up to 64 bits for each slave (e.g., measurement data, error, and warning flag, Sign of Life counter, …).

- Control Communication:

  E.g., register communication for reading/writing the sensor's registers (e.g., for reading an Electronic Datasheet or calibrating sensors).

This controller provides two RAM memory banks for each slave enabling simultaneous access by CPU while new sensor data is being transmitted via BiSS interface. A 64-byte memory supports register transfers. Sensor data acquisition is started by a software instruction, via GETSENS signal or automatically at a configurable frame repetition rate. The end of the sensor data transmission is indicated at BISSn_EOT (EOT: n = 0, 1) interrupt. An error during transmission is flagged by BISSn_NER (NER: n = 0, 1) interrupt. Error details can be accessed in the status registers. This controller generates a clock signal for sensor communication using 20 MHz system clock.

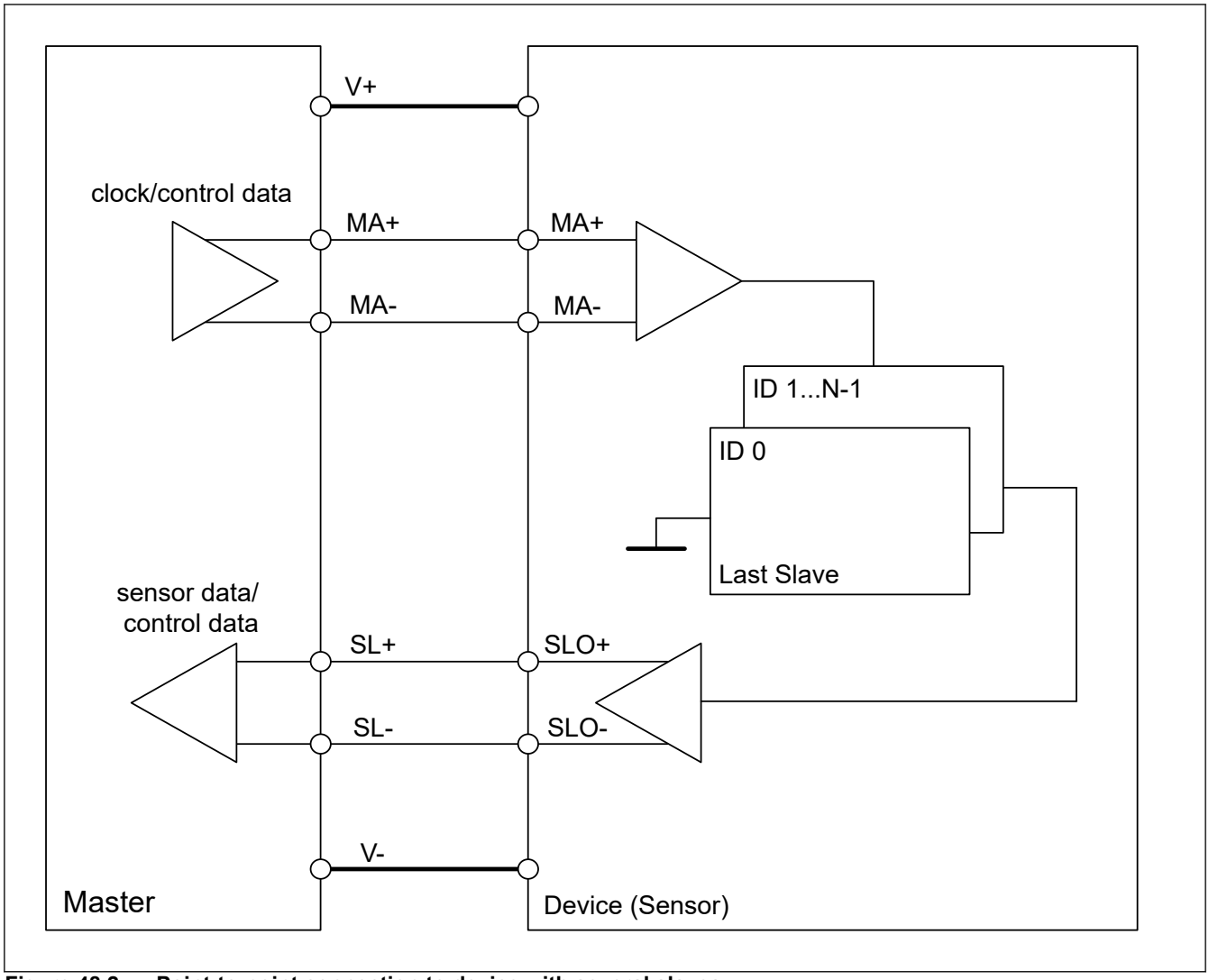


Figure 48.2 Point-to-point connection to device with several slaves

The open-source BiSS Interface Protocol implements a real-time interface for digital, serial, and secure communication between drive, sensor and actuator. In the point-to-point configuration, BiSS uses one clock line MA from the master to the sensor and one data line SL from the sensor to the master. A device may contain multiple slaves. The data input signal SL of the last slave is set to digital low. The slaves are daisy-chained and the data output signal SLO of the first slave is directly connected to the master. The BiSS protocol describes cyclic data frames and differentiates between process data and control data. Process data are completely transmitted in each frame (SCD) used as sensor data, and control data are transmitted one bit per frame (CD) used for commands and register access.

48.3.2 BiSS-C

(1) BiSS-C Frame

The BiSS-C Frame starts with a digital high on the clock line MA and is notified with a falling edge at EOT signal. On the first falling edge of MA, all slaves check the SL signal for a digital low determining the last slave. With the first rising edge at MA, all sensors start calculating their sensor data. The second rising edge of MA forces all slaves to acknowledge the BiSS-C frame with a falling edge at SL. The master uses the acknowledge to measure the line delay. When the sensor data calculation is finished, the last slave in the chain generates the start bit which will be passed synchronously through all slaves to the master. Subsequent to the start bit follows one control data bit (CDS) for all slaves which is set according to the rules of the control frame. After the CDS bit, the process data including sensor data is sent with the most significant bit (MSB) first. At the end, the master sends its control data bit (CDM) inverted on the clock line MA to conclude the BiSS-C Frame.

During processing the frame, all slaves observe the MA clock line and change into the timeout state if MA is stable for a specific time defined within each slave. In the timeout state, only the last slave forces its SL signal to digital high. The other

slaves in the chain indicate to the master that all slaves are in the timeout state. After detecting the slave's timeout with SL = 1, the master may change the MA clock line to digital high or keeps the clock line constant until the next frame begins.

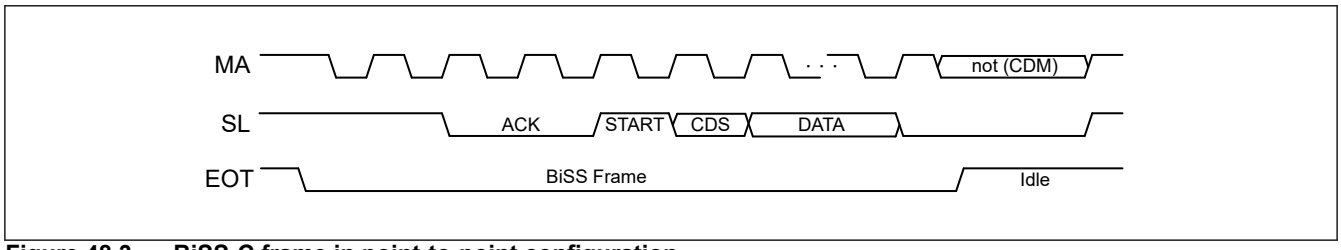


Figure 48.3 BiSS-C frame in point-to-point configuration

(2) BiSS-C Register Communication

The master is able to transmit a BiSS-C register write access to the slave without the slave's CDS feedback. The master cannot verify that the BiSS-C register write access to the slave did succeed. At the end of the SCD frame, the master sends the CDM bit inverted on the MA clock line.

(3) BiSS-C Initialization Sequence

An initialization (INIT) sequence is necessary if the last slave in the point-to-point configuration is not defined and the data line SL is digital low. In the initialization sequence, two digital low pulses are generated at MA. The slave(s) should start communication at the second falling edge of the MA pulse and answer the end of communications by generating a rising edge on SL after the BiSS timeout detection. The time between the second rising-edge at MA and the falling-edge at SL is measured as line delay and stored in the single-cycle data RAM, see INSTREG.INIT bit function.

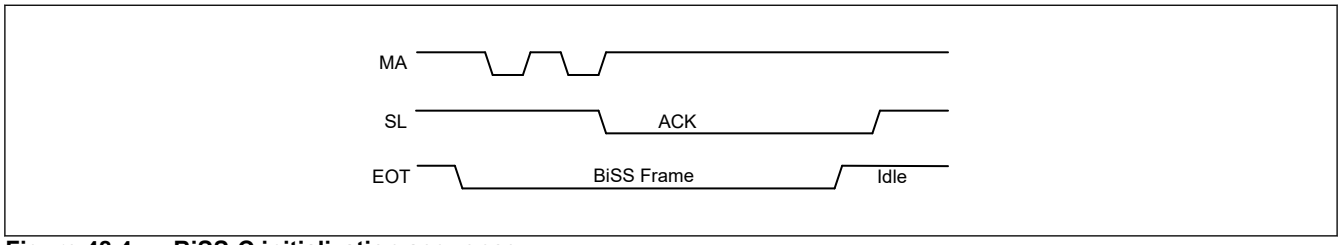


Figure 48.4 BiSS-C initialization sequence

48.3.3 Sensor Data

For sensor data, an 8x8 byte single-cycle data memory SCDATA is provided. Eight slaves with up to 64 bits of single-cycle data each are supported.

Table 48.5 Address mapping of single-cycle data (SCDATA)

Offset address	Register	Content
0x00 + 0x08 × (n - 1)	SCDATALn[31:0]	SCDATAn[31:0]
0x04 + 0x08 × (n - 1)	SCDATAHn[31:0]	SCDATAn[63:32]

Note: Slave: n = 1 to 8

The sensor data is arranged in the memory area with the least significant bit (LSB) at the lower address at bit position 0. The memory is written byte-by-byte and unused bits are set to zero. Unused bytes remain unchanged. In BiSS the CRC bits are inverted before transmission. The received CRC bits are reinverted and then stored at the most significant SCDATA byte(s). This can be disabled with CTRLCOMM.NOCRC = 1.

- Example: BiSS Sensor Bus with 3 Slaves
  The following example shows the address mapping of the sensor data and CRC bits generated by three slaves.
  - Slave 1: 19 + 2 bits of sensor data, 6 bits of CRC: Total length of 27 bits
  - Slave 2: 12 + 2 bits of sensor data, 5 bits of CRC: Total length of 19 bits
  - Slave 3: 24 bits of sensor data, 16 bits of CRC: Total length of 40 bits

Table 48.6 Example address mapping of sensor data

Offset address	SCDATA[31:0]
SCDATAL1[31:0]	Not changed [7:0], 000b, Sensor data 1 [20:0]
SCDATAH1[31:0]	00b, CRC 1 [5:0], Not changed [23:0]
SCDATAL2[31:0]	Not changed [15:0], 00b, Sensor data 2 [13:0]
SCDATAH2[31:0]	000b, CRC 2 [4:0], Not changed [23:0]
SCDATAL3[31:0]	Not changed [7:0], Sensor data 3 [23:0]
SCDATAH3[31:0]	CRC 3 [15:0], Not changed [15:0]
SCDATALn[31:0] (n = 4 to 8)	Not changed [31:0]
SCDATAHn[31:0] (n = 4 to 8)	Not changed [31:0]

In order to import new sensor data during access, this controller has two memory banks for sensor data. While sensor data is being read and written into the first RAM bank during processing of the BiSS Frame, the second RAM bank section containing sensor data of the previous frame can be read. The relevant sensor data memory banks are swapped at the end of the BiSS Frame. This can be prevented by setting INSTREG.HOLDBANK. Simultaneously the validity (STINF.SVALID) that corresponds to the RAM bank is also swapped.

As an example, the basic procedure of sensor data is below:

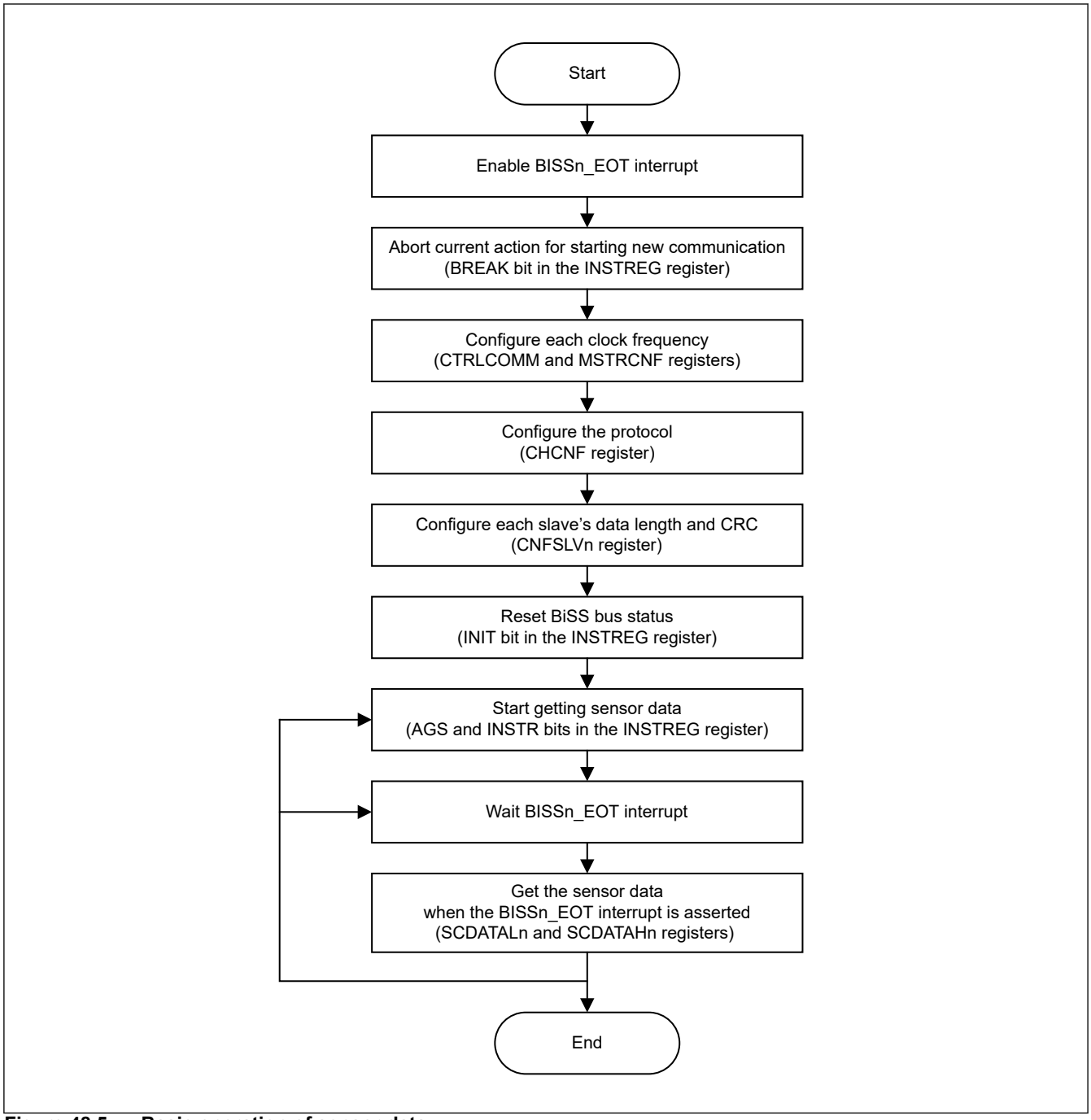


Figure 48.5 Basic operation of sensor data

48.3.4 Control Frame

48.3.4.1 BiSS Commands

The BiSS-C Protocol Description defines BiSS commands as a subset of the control communication. A BiSS command can address one or several slaves by setting the appropriate bits in IDS. To address all slaves via broadcast command, IDS has to be reset to zero. IDS[7:0] bits correspond to the lower 8 bits of Register Data 1 (RDATA1) register.

Slave addressing for BiSS commands:

- IDS[7:0] = 0x00: All slaves addressed (broadcast)
- IDS[N]:
  0: Slave with ID (7 – N) is not addressed (e.g., IDS[0]: ID 7, IDS[1]: ID 6, IDS[7]: ID 0)
  1: Slave with ID (7 – N) is addressed

CMD (corresponding to CTRLCOMM.SID_CMD_IDT[2:1] bits) determines which BiSS command is sent.

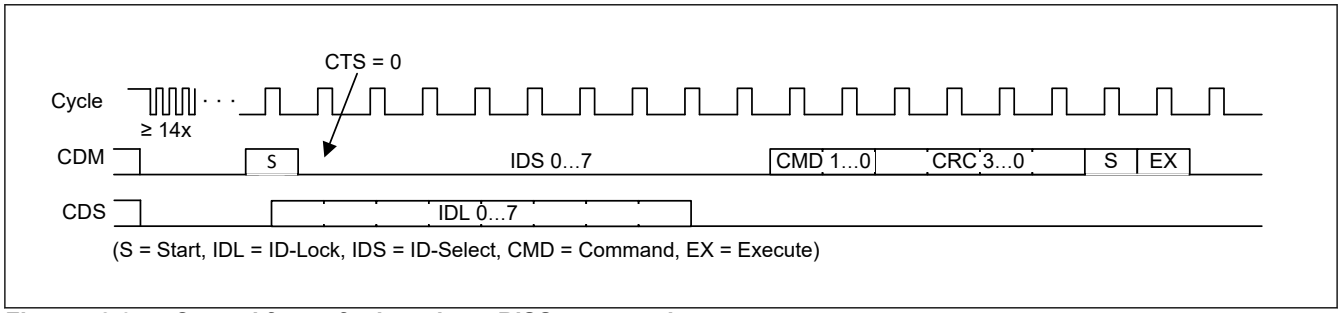


Figure 48.6 Control frame for broadcast BiSS command

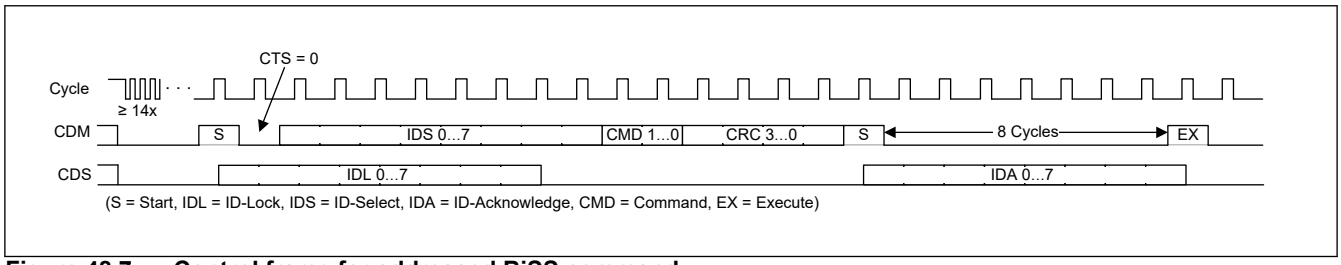


Figure 48.7 Control frame for addressed BiSS command

48.3.4.2 Register Access

The register start address (REGADR), register access direction (WNR), number of bytes (REGNUM) in REGACC register, and slave address (SLAVEID) in CTRLCOMM register are used to configure the BiSS-C control communication for register communication. If a byte count of 0 is entered for REGNUM, it means the transmission of a single register value.

Note: All configuration parameters should be stable during the control communication frame.

The following figure exemplarily shows a control frame including a register read access.

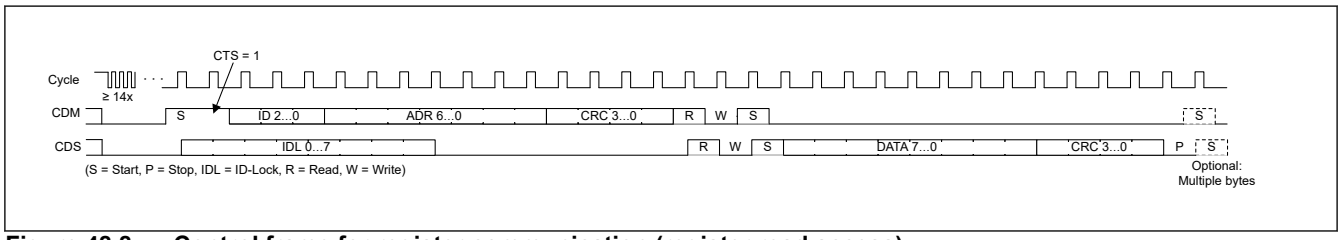


Figure 48.8 Control frame for register communication (register read access)

48.3.5 Data Acquisition

Trigger data transmissions:

- Automatically (MSTRCNF.FREQAGS ≠ AGSINFINITE)
- By GETSENS signal (MSTRCNF.FREQAGS = AGSINFINITE)
- By selecting a program with INSTREG.INSTR

(1) Automatic Sensor Data Request

With INSTREG.AGS = 1 and MSTRCNF.FREQAGS = AGSMIN, data transmissions are started automatically as soon as the slave's timeout is detected. With FREQAGS configured to a constant frame repetition rate, data transmissions are started automatically at the configured frame repetition rate.

(2) Sensor Data Request by GETSENS Signal

GETSENS signal can be used to start a new BiSS frame if enabled by INSTREG.AGS = 1 and MSTRCNF.FREQAGS = AGSINFINITE. The delay between the trigger event at GETSENS and the falling edge at MA is defined by the system clock fCLK and the single-cycle clock frequency fMA (configured with CTRLCOMM.FREQS).

tGETSENS to falling edge at MA = 2/fCLK + 1/(2 × fMA)

(3) Program Selection with Instructions

INSTREG.INSTR is used to trigger data transmissions by program. With INSTREG.AGS = 0, the master starts the data transmission after finishing writing INSTREG register and resets INSTR after executing the frame automatically. STINF.nAGSERR error will be generated if the SL line is low at the start of the frame. The error can be suppressed by setting MSTRCNF.FREQAGS = AGSMIN due to waiting for a high state at SL. A BiSS-C control communication (refer to CTRLCOMM.REGVERS) can be triggered via INSTR as well. It is also possible to enable reduced BiSS frames for control communication with BiSS commands. Within a running control communication, the CDM bit is generated automatically. Otherwise, the generation of CDM depends on INSTR.

48.4 Usage Notes

48.4.1 The value to be set to SCDLEN in the CNFSVn register

For the data length set in SCDLEN, the following sum is required.

MT (multi-turn) bits + ST (single-turn) bits + 2 bits (Error bit and Warning bit between data and CRC in SCD frame)

48.4.2 The storage of received CRC when SCDLEN = 33 bits or more

The CRC of SCD is normally stored at the end of SCDATA, but if SCDLEN is set to 33 bits or more, it is not stored in SCDATA. However, CRC check is executed every time by HW, the result is informed by the status flag (nSCDERR).

When need to read the CRC data, it is required to set the data length including CRC to SCDLEN (ex: set 49 (43 + 6) bits when data = 43 bits and CRC = 6 bits). In this case, the CRC check is required to be executed by SW.

49. HIPERFACE DSL (HDSL)

49.1 Overview

The MPU includes HIPERFACE DSL (HDSL) master to enable communication between the encoder and HIPERFACE DSL interface. Table 49.1 describes the HDSL specifications and Figure 49.1 shows a block diagram of the HDSL.

Table 49.1 HIPERFACE DSL (HDSL) specifications

Item	Description
Number of units	2 units
Protocol	HIPERFACE DSL
Transfer rate	Up to 9.375 Mbaud
Functions	• HIPERFACE DSL with safety • Safe channel 1 interface is selectable from internal bus access or external MCU access through SPI. This safe channel 1 interface enables safety application and it can be implemented in this MPU together with normal application. • Safe channel 2 interface is SPI I/F which is accessed from external MCU • SensorHub SPI PIPE interface is not supported • System diagnostics function • Fast position function – Estimator – Free running mode – SYNC mode • Status and error messages – Event register – Online status – Status summary of the motor feedback system – Motor feedback system error messages • Safety implementation
Interrupt sources	• Configurable interrupt: 5 – Normal related – Safety related – Fast position received – Safety position received • Error related interrupt: 7 – Position estimator activated – Transmission error in safe channel 1 – Safe position not valid – Fast channel/position error – Fast channel/position threshold error – DSL message encoding error – Estimator deviation threshold reached
Event link	SYNC input for HW trigger provided from ELC
Module-stop function	Module-stop state can be set to reduce power consumption

This HIPERFACE DSL® MASTER is licensed from SICK AG. For details, see the following SICK document.

- HIPERFACE DSL® MASTER Safety Integration Manual

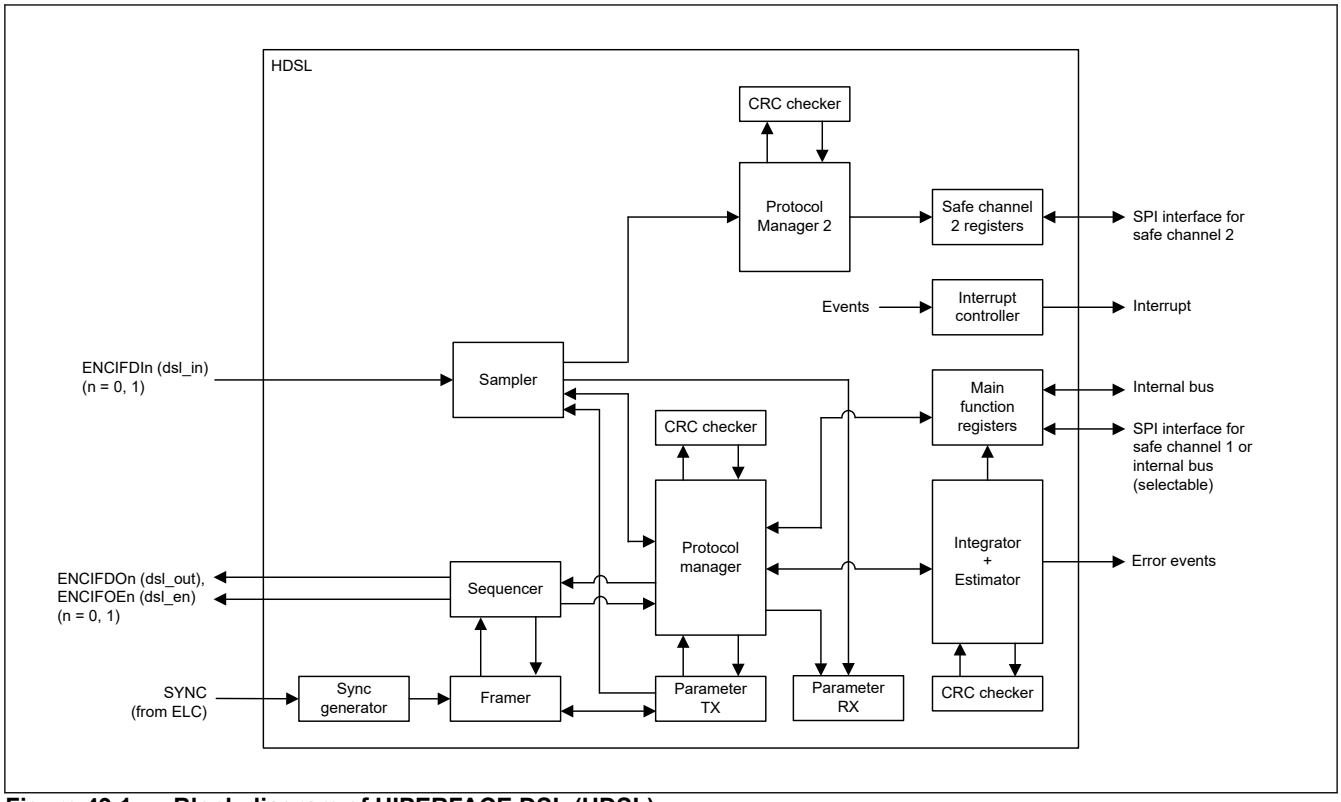


Figure 49.1 Block diagram of HIPERFACE DSL (HDSL)

Table 49.2 lists the input/output pins of the HIPERFACE DSL.

Table 49.2 Pin configuration of the HDSL

Unit	Pin name	I/O	Function
HDSLn (n = 0, 1)	ENCIFOEn (dsl_en)	Output	HDSL unit n DSL cable transceiver, activation
	ENCIFDOn (dsl_out)	Output	HDSL unit n DSL cable, output data
	ENCIFDIn (dsl_in)	Input	HDSL unit n DSL cable, input data
	HDSLn_LINK	Output	HDSL unit n LINK
	HDSLn_SMPL	Output	HDSL unit n Test signal line sampler
	HDSLn_CLK1	Input	HDSL unit n SPI clock safe channel 1
	HDSLn_SEL1	Input	HDSL unit n SPI selection safe channel 1
	HDSLn_MISO1	Output	HDSL unit n data output safe channel 1
	HDSLn_MOSI1	Input	HDSL unit n data input safe channel 1
	HDSLn_CLK2	Input	HDSL unit n SPI clock safe channel 2
	HDSLn_SEL2	Input	HDSL unit n SPI selection safe channel 2
	HDSLn_MISO2	Output	HDSL unit n data output safe channel 2
	HDSLn_MOSI2	Input	HDSL unit n data input safe channel 2

Table 49.3 HDSL interrupt sources (1 of 2)

Name	Interrupt sources	CPU GIC request	DMAC activation	ELC activation
HDSLn INT	Interrupt output	Possible	Possible	Not possible
HDSLn INTS	Interrupt output (safety related)	Possible	Possible	Not possible
HDSLn_FP	Fast position received interrupt	Possible	Possible	Not possible
HDSLn_SP	Safety position received interrupt	Possible	Possible	Possible

Table 49.3 HDSL interrupt sources (2 of 2)

Name	Interrupt sources	CPU GIC request	DMAC activation	ELC activation
HDSLn estimator on	Position estimator activated	Possible*1	Not possible	Not possible
HDSLn safe channel error	Transmission error in safe channel 1	Possible*1	Not possible	Not possible
HDSLn safe position error	Safe position not valid	Possible*1	Not possible	Not possible
HDSLn acceleration error	Fast channel/position error	Possible*1	Not possible	Not possible
HDSLn acceleration threshold error	Fast channel/position threshold	Possible*1	Not possible	Not possible
HDSLn encoding error	DSL message encoding error	Possible*1	Not possible	Not possible
HDSLn deviation threshold error	Estimator deviation threshold reached	Possible*1	Not possible	Not possible

Note: n = 0, 1
Note 1. Some errors from other peripherals are ORed and connected to the GIC.

49.2 Register Descriptions

HDSL has main function registers for the drive interface, safe channel 1 interface, and safe channel 2 interface. The drive interface registers and safe channel 1 interface registers can be accessed through the internal bus. The safe channel 1 interface register and safe channel 2 interface register can be accessed through each SPI interface. In this case, register address should be referred to offset address only.

And register access operation can be executed with 8-bit/16-bit/32-bit access. However, when 16-bit/32-bit access with unappropriate R/W attribution for the registers is executed, HDSL operation cannot be guaranteed.

49.2.1 Drive Interface Register

Table 49.4 Drive interface register map (1 of 2)

Offset address	Register symbol	Register name
0x00	SYS_CTRL	System Control Register
0x01	SYNC_CTRL	Synchronization Control Register
0x03	MASTER_QM	Quality Monitoring Register
0x04	EVENT_H	High Byte Event Register
0x05	EVENT_L	Low Byte Event Register
0x06	MASK_H	High Byte Event Mask Register
0x07	MASK_L	Low Byte Event Mask Register
0x08	MASK_SUM	Summary Mask Register
0x09	EDGES	Edge Register
0x0A	DELAY	Run Time Delay Register
0x0B	VERSION	Version Register
0x0D	ENC_ID2	Encoder ID 2 Register
0x0E	ENC_ID1	Encoder ID 1 Register
0x0F	ENC_ID0	Encoder ID 0 Register
0x10	POS4	Fast Position Byte 4 Register
0x11	POS3	Fast Position Byte 3 Register
0x12	POS2	Fast Position Byte 2 Register
0x13	POS1	Fast Position Byte 1 Register
0x14	POS0	Fast Position Byte 0 Register
0x15	VEL2	Speed Byte 2 Register
0x16	VEL1	Speed Byte 1 Register

Table 49.4 Drive interface register map (2 of 2)

Offset address	Register symbol	Register name
0x17	VEL0	Speed Byte 0 Register
0x18	MIR_SUM	Mirror Status Summary Register
0x20	PC_BUF0	Parameter Channel Buffer 0 Register
0x21	PC_BUF1	Parameter Channel Buffer 1 Register
0x22	PC_BUF2	Parameter Channel Buffer 2 Register
0x23	PC_BUF3	Parameter Channel Buffer 3 Register
0x24	PC_BUF4	Parameter Channel Buffer 4 Register
0x25	PC_BUF5	Parameter Channel Buffer 5 Register
0x26	PC_BUF6	Parameter Channel Buffer 6 Register
0x27	PC_BUF7	Parameter Channel Buffer 7 Register
0x28	PC_ADD_H	High Byte Long Message Address Register
0x29	PC_ADD_L	Low Byte Long Message Address Register
0x2A	PC_OFF_H	High Byte Long Message Address Offset Register
0x2B	PC_OFF_L	Low Byte Long Message Address Offset Register
0x2C	PC_CTRL	Parameter Channel Control Register
0x2D	PIPE_S	The SensorHub Channel Status Register
0x2E	PIPE_D	SensorHub FIFO Register
0x2F	PC_DATA	Parameter Channel Short Message Mirror Register
0x38	ACC_ERR_CNT	Fast Position Error Counter Register
0x39	MAXACC	Fast Position Acceleration Boundary Register
0x3A	MAXDEV_H	Fast Position Estimator Deviation High Byte Register
0x3B	MAXDEV_L	Fast Position Estimator Deviation Low Byte Register
0x3F	DUMMY	Dummy Register
0x60 + 0x1 × n	MIR_STn	Mirror Status n Register (n = 0 to 7)

49.2.1.1 SYS_CTRL : System Control Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x00

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PRST	MRST	FRST	LOOP	—	—	SPOL	OEN
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	OEN	Activation of the output 0: The impedance of the DSL cable is high. 1: The DSL cables are activated for output to the DSL Slave.	R/W
1	SPOL	Polarity of the synchronization pulse 0: The SYNC trailing edge is used. 1: The SYNC leading edge is used.	R/W
3:2	—	These bits are read as 0. The write value should be 0.	R/W
4	LOOP	Test drive interface Value for the read-back test for drive interface. This value has no other purpose.	R/W

Bit	Symbol	Function	R/W
5	FRST	Pipeline FIFO, reset 0: Normal FIFO access 1: The FIFO is reset. Data is not stored and cannot be read.	R/W
6	MRST	Messages reset 0: Normal parameters channel action 1: The parameters channel is reset. Current short and long messages are discarded.	R/W
7	PRST	Protocol reset 0: Normal protocol action 1: A forced reset of the protocol status will be initiated. If the bit is cleared, a restart of the connection is triggered.	R/W

The SYS_CTRL register contains the main control bits of the DSL Master.

49.2.1.2 SYNC_CTRL : Synchronization Control Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x01

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ES[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	ES[7:0]	External synchronization 0x00: Position sampling during free running at the shortest cycle time. Others: Position sampling with the SYNC input synchronized. The value from ES determines the number of position samplings carried out in one SYNC cycle. The user must match the number of samplings per cycle to the shortest frame length.	W

The synchronization control register for control of the synchronization contains the bit with which the synchronization source for position sampling is controlled.

49.2.1.3 MASTER_QM : Quality Monitoring Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x03

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LINK	—	—	—	QM[3:0]			
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Function	R/W
3:0	QM[3:0]	Quality monitoring bits Quality monitoring value. Higher values indicate a better connection. If the quality monitoring reaches the value 0x0, a forced reset of the protocol is carried out.	R
6:4	—	These bits are read as 0.	R
7	LINK	DSL protocol connection status It should be noted that LINK is also represented at the HDSLn_LINK output. (n = 0, 1). 0: No connection present or connection error due to a communications error. 1: Protocol connection between DSL Master and Slave was established.	R

The MASTER_QM register contains the quality monitoring value for the data connection.

49.2.1.4 EVENT_H : High Byte Event Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x04

Bit position:	7	6	5	4	3	2	1	0
Bit field:	INT	SUM	—	—	POS	—	DTE	PRST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PRST	Protocol reset warning 0: Normal protocol action 1: The forced protocol reset was triggered.	R/W*1
1	DTE	Estimator deviation threshold error 0: Current value of deviation smaller than the specified maximum. 1: Current value of deviation greater than the specified maximum.	R/W*1
2	—	This bit is read as 0.	R
3	POS	Estimator turned on This error usually indicates a transmission error on the DSL connection. If this error occurs frequently, the wiring of the DSL connection should be checked. If this error occurs continuously, there is probably an error in the motor feedback system. 0: The data for the fast position was correctly transmitted. 1: Fast position data consistency error. The fast position read through drive interface is supplied by the estimator.	R/W*1
5:4	—	These bits are read as 0.	R
6	SUM	Remote event monitoring When the SUM bit is set, an error or a warning has been transmitted from the DSL Slave. The frequency inverter application must check the SUMMARY register to obtain a detailed description. This bit is level sensitive. 0: All DSL Slave events are cleared. 1: The DSL Slave has signaled an event and the summary mask is set accordingly.	R/W*1
7	INT	Interrupt status 0: No interrupt in DSL Master 1: DSL Master interrupt occurred	R

Note 1. Write 0 only to the bit to be cleared.

49.2.1.5 EVENT_L : Low Byte Event Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x05

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	MIN	ANS	—	QMLW	FREL	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	This bit is read as 0. The write value should be 0.	R/W
1	FREL	Channel free for "long message" If the bit is set, the frequency inverter application can trigger a "long message". Provided no answer has been received from the DSL Slave, this bit remains deleted. As the processing duration of a "long message" in the motor feedback system is not specified, a user time limit condition should be installed via the frequency inverter application. When a time limit is exceeded, the MRST bit in the SYS_CTRL register is set, which causes the Parameters Channel to be reset. 0: No "long message" can be sent. 1: A "long message" can be sent on the Parameters Channel.	R/W*1

Bit	Symbol	Function	R/W
2	QMLW	Quality monitoring low-value warning This warning indicates that a transmission error occurred at bit level for one of the CRC values. If this error occurs frequently, the wiring of the DSL connection should be checked. 0: Quality monitoring value greater than or equal to "14" 1: Quality monitoring value below "14"	R/W*1
3	—	This bit is read as 0. The write value should be 0.	R/W
4	ANS	Erroneous answer to "long message" This error indicates that the transmission of an answer from the DSL Slave to the last "long message" failed. The frequency inverter application must send the "long message" again. 0: The last answers to "long messages" were error-free. 1: An error occurred during the answer to a long message. The effectiveness of the previous transaction is not known.	R/W*1
5	MIN	Message initialization When this warning is displayed, the Parameters Channel is still in the initialization status and no "short message" or "long message" can be triggered. 0: No acknowledgment for the initialization received. 1: An acknowledgment was received from the Slave for the initialization of a message.	R/W*1
7:6	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Write 0 only to the bit to be cleared.

The EVENT_H/EVENT_L register contains the messaging bits for all warning and error modes of the DSL system.

All messaging bits are set by the DSL Master if a corresponding status is determined.

An event bit that has been set is not reset by the DSL Master. The frequency inverter application must clear bits that have been set.

49.2.1.6 MASK_H : High Byte Event Mask Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x06

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	MSUM	—	—	MPOS	—	MDTE	MPRST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MPRST	Mask for protocol reset warning 0: A protocol reset does not set the interrupt signal. 1: A protocol reset sets the interrupt signal.	W
1	MDTE	Mask for estimator deviation threshold error warning 0: A high estimator deviation threshold error value does not set the interrupt signal. 1: A high estimator deviation threshold error value sets the interrupt signal.	W
2	—	The write value should be 0.	W
3	MPOS	Mask for fast position error 0: An error in the fast position does not set the interrupt signal. 1: An error in the fast position sets the interrupt signal.	W
5:4	—	The write value should be 0.	W
6	MSUM	Mask for remote event monitoring 0: DSL Slave events that are masked in the SUMMARY register do not set the interrupt signal. 1: DSL Slave events that are masked in the SUMMARY register set the interrupt signal.	W
7	—	The write value should be 0.	W

49.2.1.7 MASK_L : Low Byte Event Mask Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x07

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	MMIN	MANS	—	MQMLW	MFREL	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	—	The write value should be 0.	W
1	MFREL	Mask for "channel free for "long message"" 0: If a "long message" can be sent on the Parameters Channel, the interrupt signal is not set. 1: If a "long message" can be sent on the Parameters Channel, the interrupt signal is set.	W
2	MQMLW	Mask for low-quality monitoring value warning 0: A low-quality monitoring value does not set the interrupt signal. 1: A low-quality monitoring value sets the interrupt signal.	W
3	—	The write value should be 0.	W
4	MANS	Mask for erroneous answer to "long message" 0: A transmission error during the answer to a long message does not set the interrupt signal. 1: A transmission error during the answer to a long message sets the interrupt signal.	W
5	MMIN	Mask for message initialization confirmation 0: The acknowledgment for the initialization of a DSL Slave message does not set the interrupt signal. 1: The acknowledgment for the initialization of a DSL Slave message sets the interrupt signal.	W
7:6	—	The write value should be 0.	W

In the MASK_H/MASK_L registers, the events are set with which the interrupt signal is set.

49.2.1.8 MASK_SUM : Summary Mask Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x08

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MSUM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	MSUM[7:0]	Mask for status summary bits 0: In the set status, the corresponding status summary bit does not set the SUM event monitoring and the interrupt signal. 1: In the set status, the corresponding status summary bit sets the SUM event monitoring and the interrupt signal.	W

In the MASK_SUM register, the DSL Slave collective events are determined with which the SUM event monitoring in the EVENT_H register, as well as the interrupt signal, are set.

Several events can be masked to trigger an interrupt. In addition, events from the DSL Master can be combined with these events.

49.2.1.9 EDGES : Edge Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)
HDSLS1m = 0x9004_1000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x09

Bit position:	7	6	5	4	3	2	1	0
Bit field:	EDGES							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	EDGES	Identification of edges in the cable signal 0: No edge was detected in the time period of the corresponding bit. 1: An edge was detected in the time period of the corresponding bit.	R

The EDGES register contains the time control for the DSL cable bit sampling and can be used to monitor the connection quality.

Each individual EDGES register bit is set if, at system start-up, an edge of the test signal is detected during the time period of the corresponding bit. An edge is defined as a change in cable value between successive detections. The sampling is carried out eight times as fast as the cable bit rate.

The register is write-protected. The contents of this register do not change after the start-up phase. A new bit sampling pattern is only generated after a forced reset of the protocol.

49.2.1.10 DELAY : Run Time Delay Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)
HDSLS1m = 0x9004_1000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RSSI[3:0]				CBLDLY[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	CBLDLY[3:0]	4-bit value for cable delay This value gives the cable signal round trip delay of cable and transceivers in bits. This value enables a rough estimate of cable length to be made.	R
7:4	RSSI[3:0]	Indication of the received signal strength 4-bit value for the cable signal strength, from "0" to "12". Higher values indicate better connection quality. Low connection (< 2) quality affects QM. RSSI is continuously updated during operation and used for signal monitoring during run time.	R

The DELAY register stores information about the run-time delay of the system cable and the signal strength. The register can be used to monitor the connection quality.

The register is write-protected.

The value for Line Delay does not change after the start-up phase. A fresh value for Line Delay is only measured after a forced reset of the protocol.

Table 49.5 shows the relationship between the value in Line Delay and the cable length of the DSL connection.

Table 49.5 Cable delay (1 of 2)

Cable delay (CBLDLY[3:0])	DSL connection cable delay	Cable length DSL connection
0	< 100 ns	< 10 m

Table 49.5 Cable delay (2 of 2)

Cable delay (CBLDLY[3:0])	DSL connection cable delay	Cable length DSL connection
1	100 to 200 ns	10 to 20 m
2	200 to 300 ns	20 to 30 m
3	300 to 400 ns	30 to 40 m
4	400 to 500 ns	40 to 50 m
5	500 to 600 ns	50 to 60 m
6	600 to 700 ns	60 to 70 m
7	700 to 800 ns	70 to 80 m
8	800 to 900 ns	80 to 90 m
9	900 to 1000 ns	90 to 100 m

49.2.1.11 VERSION : Version Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)
HDSLS1m = 0x9004_1000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CODE[1:0]		MAJOR[1:0]		MINOR[3:0]			
Value after reset:	0	1	0	1	0	1	1	1

Bit	Symbol	Function	R/W
3:0	MINOR[3:0]	Minor release number The current version is 7.	R
5:4	MAJOR[1:0]	Major release number The current version is 1.	R
7:6	CODE[1:0]	Type of module 0 1: DSL Master Others: Reserved	R

The VERSION register contains the release version of the DSL Master module. The register is write-protected.

49.2.1.12 ENC_ID2 : Encoder ID 2 Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)
HDSLS1m = 0x9004_1000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	SCI[2:0]			ENCID[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
3:0	ENCID[3:0]	Encoder designation code bit 19 to bit 16	R
6:4	SCI[2:0]	Indication of special characters 3-bit special character for later enhancements of the encoder designation code. Not allocated.	R
7	—	This bit is read as 0.	R

The ENC_ID2/1/0 registers contain the designation code of the motor feedback system connected to the DSL Master. In the current protocol specification, the designation code is 20 bits long. With later enhancements, the free bits in the encoder ID registers are used to indicate special characters.

These registers are write-protected.

The individual ENC_ID2/1/0 register bits are allocated as follows:

Table 49.6 Encoder designation code

Bit	Symbol	Function
19	ENC_ID2.ENCID[3]	Continue In High status, Continue indicates that ENDID is longer than 20 bits (for future use).
18:16	ENC_ID2.ENCID[2:0]	Reserved These bits are read as 0.
15:12	ENC_ID1.ENCID[7:4]	User-defined encoder index 4-bit value (0 to 15) for user-defined encoder index
11	ENC_ID1.ENCID[3]	Reserved This bit is read as 0.
10	ENC_ID1.ENCID[2]	Sign In High status, Sign indicates that the position value is signed, in Low status, Sign indicates that the position value is not signed.
9:8	ENC_ID1.ENCID[1:0]	#Pos – #Acc Length of position information (standard value: 40 bits) minus length of the acceleration value transmitted (standard value: 11 bits).
7:4	ENC_ID0.ENCID[7:4]	
3:0	ENC_ID0.ENCID[3:0]	#Acc – 8 Length of the acceleration value transmitted (standard value: 11 bits) minus 8.

49.2.1.13 ENC_IDn : Encoder ID n Register (n = 0, 1)

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)
HDSLS1m = 0x9004_1000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x0E (ENC_ID1)
0x0F (ENC_ID0)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENCID[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	ENCID[7:0]	Encoder designation code, byte n	R

49.2.1.14 POSn : Fast Position Byte n Register (n = 0 to 4)

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x10 (POS4)
0x11 (POS3)
0x12 (POS2)
0x13 (POS1)
0x14 (POS0)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	POS[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	POS[7:0]	Fast position, byte n Position value of the motor feedback system (length: 40 bits), incrementally generated.	R

The POSn registers for the fast position contain the value of the motor feedback system connected. This position is generated incrementally from the safe position at start-up and is updated with every protocol frame.

After every eight protocol frames, the fast position is checked against the safe position (VPOSn register).

The position sampling point is determined by the ES bits in SYNC_CTRL register.

Only those POS bits are activated that lie within the range that the motor feedback system has actually measured. All other higher value bits are read as "0". The number of measurable bits can be taken from ENCID bits 9 to 0 in the ENC_ID0 to 2 registers.

If Sign is set in the ENC_IDn register, the value of the fast position is given signed in the two's complement.

The units of the position value are (steps). These registers are write-protected.

49.2.1.15 VELn : Speed Byte n Register (n = 0 to 2)

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x15 (VEL2)
0x16 (VEL1)
0x17 (VEL0)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	VEL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	VEL[7:0]	Speed, byte n Speed of the motor feedback system (length: 24 bits)	R

The VELn registers contain the speed values of the connected motor feedback system. This value is calculated as a Δ position from the acceleration value (ΔΔposition) transmitted on the process data channel and the currently updated protocol frame.

The speed sampling point is determined by the ES bits of the SYNC_CTRL register. The units of the speed value are steps/ frame cycle time.

These registers are write-protected.

49.2.1.16 MIR_SUM : Mirror Status Summary Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x18

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SUM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SUM[7:0]	Status summary bit bit [0] (interface): 0: The DSL Slave protocol has not triggered an error, warning, or event. 1: An error, a warning, or an event associated with the DSL Slave protocol interface was triggered. bit [7:1] (external resource): 0: The corresponding error, warning, or event is not active. 1: An error, a warning, or an event associated with the DSL Slave external resources was triggered.	R

The MIR_SUM register contains the same information for the drive application. The MIR_SUM register is based on the encoder mirror registers MIR_ST7 to MIR_ST0 in the DSL Slave and therefore delayed when compared to the SAFE_SUM register. The bits in the MIR_SUM register can be read by the drive application only.

49.2.1.17 PC_BUFn : Parameter Channel Buffer n Register (n = 0 to 7)

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x20 + 0x1 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PCBUF[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PCBUF[7:0]	Parameter Channel, byte n Eight bytes for the answer to a long message (read operation) or for a "long message" write operation. PCBUF[0]: bit 7 to bit 0 for Parameters Channel PCBUF[1]: bit 15 to bit 8 for Parameters Channel PCBUF[2]: bit 23 to bit 16 for Parameters Channel PCBUF[3]: bit 31 to bit 24 for Parameters Channel PCBUF[4]: bit 39 to bit 32 for Parameters Channel PCBUF[5]: bit 47 to bit 40 for Parameters Channel PCBUF[6]: bit 55 to bit 48 for Parameters Channel PCBUF[7]: bit 63 to bit 56 for Parameters Channel	R/W

The eight PC_BUFn registers of the Parameters Channel buffer contain the answer to the last "long message" request or the data for a "long message" write operation.

Depending on the length of the "long message" answer, the registers are used as follows:

Table 49.7 Data length of the "long message"

Length of the "long message"	Register used (Offset address)
8 bytes	0x20 to 0x27
4 bytes	0x20 to 0x23
2 bytes	0x20 to 0x21
0 bytes	None

These registers are also for the reporting of error conditions arising from a "long message" operation. If, when accessing a resource, an error due to a "long message" arises (e.g. invalid data, error in the A/D conversion), after the answering message has been received the LOFF bit in the PC_ADD_H register is set. In this case, the Parameters Channel buffer bytes 0 and 1 contain an error code.

The meaning of the error code depends on the particular HIPERFACE DSL® encoder.

49.2.1.18 PC_ADD_H : High Byte Long Message Address Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x28

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	LRW	LOFF	LIND	LLEN[1:0]		LADDH[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
1:0	LADDH[1:0]	Long message address, bit 9 to bit 8 Database entry with 10-bit address for a "long message" operation.	W
3:2	LLEN[1:0]	Data length of the "long message" 0 0: No data bytes 0 1: 2 data bytes 1 0: 4 data bytes 1 1: 8 data bytes	W
4	LIND	Indirect addressing of long messages 0: Direct addressing of "long messages". The operation affects the database entry given in the current address. 1: Indirect addressing of "long messages". During this operation, the stored address content in the given database entry is evaluated.	W
5	LOFF	Long message addressing mode/long message error 0: **Write access:** Addressing of "long message" without offset. The offset value from the PCOFFn registers is not used. **Read access:** The last "long message" was correctly processed. 1: **Write access:** Offset addressing of "long message". The offset value from the PCOFFn registers is used in the resource of the selected database entry as a sub-address. **Read access:** The last "long message" caused an error.	R/W
6	LRW	Long message, read/write mode 0: "long message" write operation 1: "long message" read operation	W
7	—	This bit is read as 0. The write value should be 0.	R/W

49.2.1.19 PC_ADD_L : Low Byte Long Message Address Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x29

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LADDL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	LADDL[7:0]	Long message address, bit 7 to bit 0 Database entry with 10-bit address for a "long message" operation	W

The addresses and the addressing mode for "long messages" sent over the Parameters Channel are determined in the PC_ADD_H/PC_ADD_L registers.

In addition, the PCA_DD_H register contains indications of errors arising from "long message" operations. For this sort of error, the Parameters Channel buffer contains the error code in bytes 0 and 1 associated with this status.

49.2.1.20 PC_OFF_H : High Byte Long Message Address Offset Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x2A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LID	LOFFADDH[6:0]						
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
6:0	LOFFADDH[6:0]	Long message offset value, bit 14 to bit 8 The 15-bit offset value of the "long message" address offset is stored in these bits.	W
7	LID	Long message identification	R

49.2.1.21 PC_OFF_L : Low Byte Long Message Address Offset Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x2B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	LOFFADDL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	LOFFADDL[7:0]	Long message offset value, bit 7 to bit 0 The 15-bit offset value of the "long message" address offset is stored in these bits.	W

The PC_OFF_H/PC_OFF_L registers for long messages are used in "long message" operations if LOFF is set in the PC_ADD_H register. In this case, the LOFFADD value from these registers is used to communicate with the sub-address of a multiple-byte encoder resource.

Only write access is possible for these registers.

49.2.1.22 PC_CTRL : Parameter Channel Control Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x2C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	—	—	LSTA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	LSTA	Control of the long message start 0: No effect 1: A "long message" transaction is started with the values currently stored in the "long message" registers.	W
7:1	—	The write value should be 0.	W

The PC_CTRL register for the Parameters Channel handles the start of "long message" transactions. After setting all "long message" registers (registers PC_BUF0 to 7, PC_ADD_H/PC_ADD_L, and PC_OFF_H/PC_OFF_L), the "long message" is transmitted to the DSL Slave by setting the LSTA bit.

49.2.1.23 PIPE_S : SensorHub Channel Status Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x2D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	POVR	PEMP	PERR	PSCI
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	PSCI	Indication for special characters in the SensorHub Channel This bit is stored together with the pipeline data byte in question in the FIFO buffer. 0: Indication for "no special character". 1: A special character was received in the SensorHub Channel.	R
1	PERR	Coding error of the bits in the SensorHub Channel This bit is stored together with the pipeline data byte in question in the FIFO buffer. 0: No error in bit coding. 1: The bit level coding of the data currently in the SensorHub Channel is erroneous.	R
2	PEMP	The SensorHub channel buffer is empty This bit is updated after every access to the FIFO buffer. 0: No "buffer empty" error. 1: A read request was issued, but the FIFO buffer for SensorHub Channel data is empty. In this case, PIPE_D contains the value 0x00.	R
3	POVR	SensorHub Channel overflow This bit is deleted after the read process. Special characters are normally used as data separators or to indicate special events. To obtain information about which special character was received, the PIPE_D register must be read. All 8b10b special characters can be used on the SensorHub channel. An exception is the "K30.7" symbol that is used in HIPERFACE DSL® to indicate "no data" and is not stored in the FIFO buffer. Below table contains the supported 8b10b special characters.	R

Special characters	Coding in register PIPE_D
K28.0	0x1C
K28.1	0x3C
K28.2	0x5C
K28.3	0x7C
K28.4	0x9C
K28.5	0xBC
K28.6	0xDC
K28.7	0xFC
K23.7	0xF7
K27.7	0xFB
K29.7	0xFD

0: The capacity of the FIFO buffer for SensorHub Channel data is not yet exhausted.
1: The capacity of the 8-byte FIFO buffer for SensorHub Channel data was exhausted and since the last read process, values have been discarded.

7:4	—	These bits are read as 0.	R

The PIPE_S register provides information about the current status of the SensorHub Channel. When this register is read, the current data from the FIFO buffer is read and stored in an intermediate register so that a subsequent read process in the PIPE_D register can be considered to be completed at the same time as the PIPE_S register is read. Using this mechanism, any deviation between status and data information in this instance will prevent new data entering the SensorHub Channel during access to the FIFO buffer.

49.2.1.24 PIPE_D : SensorHub Channel Data Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x2E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SCDATA[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SCDATA[7:0]	SensorHub Channel data 8-bit value of the FIFO buffer for SensorHub Channel data	R

The PIPE_D register contains the SensorHub Channel data that is stored in an 8 byte FIFO buffer. If new data arrives at the buffer when it is full, before PIPE_D is read, the oldest value is discarded and the POVR bit in PIPE_S is set. If a read request is issued when the buffer is empty, the PEMP bit in PIPE_S is set and the value 0x00 is transmitted. At the moment that the PIPE_S register is accessed, the corresponding PIPE_D value is frozen to guarantee synchronization between status and data information.

49.2.1.25 PC_DATA : Parameter Channel Short Message Mirror Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x2F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PCDATA[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PCDATA[7:0]	"Short message" Mirror register data 8-bit value of the requested mirror register	R

When reading the mirror registers MIR_ST7 to MIR_ST0 instead of the real remote registers, the answer is available immediately in the PC_DATA register.

49.2.1.26 ACC_ERR_CNT : Fast Position Error Counter Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x38

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	CNT[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	CNT[4:0]	Position error count/threshold **Read:** 5-bit value of count of transmitted fast position values with consecutive transmission errors **Write:** 5-bit value for threshold	R/W
7:5	—	These bits are read as 0. The write value should be 0.	R/W

The ACC_ERR_CNT register returns the count of transmitted fast position values with consecutive transmission errors. The value is clamped to a maximum of 31 (0x1F).

With a writing access the error threshold can be set. This value is also clamped to a maximum of 31 (0x1F). If the count of transmitted fast position values with consecutive transmission errors exceeds this threshold, the error threshold will be set to 1.

49.2.1.27 MAXACC : Fast Position Acceleration Boundary Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x39

Bit position:	7	6	5	4	3	2	1	0
Bit field:	RES[1:0]		MNT[5:0]					
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	MNT[5:0]	Mantissa of fast position acceleration boundary	W
7:6	RES[1:0]	Resolution of fast position acceleration boundary	W

The MAXACC register allows setting an acceleration threshold for a given application. This threshold is used by the fast position estimator to clamp the acceleration of the estimated position during communication or sensor failures of the fast position channel.

49.2.1.28 MAXDEV_H : Fast Position Estimator Deviation High Byte Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x3A

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DEVH[7:0]							
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	DEVH[7:0]	Fast position estimator deviation high byte **Read:** 16-bit value of position deviation **Write:** 16-bit value for deviation threshold	R/W

49.2.1.29 MAXDEV_L : Fast Position Estimator Deviation Low Byte Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x3B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DEVL[7:0]							
Value after reset:	1	1	1	1	1	1	1	1

Bit	Symbol	Function	R/W
7:0	DEVL[7:0]	Fast position estimator deviation low byte **Read:** 16-bit value of position deviation **Write:** 16-bit value for deviation threshold	R/W

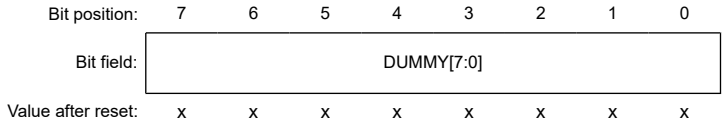
The MAXDEV_H/MAXDEV_L registers return the maximum absolute position deviation while the position estimator is active. The returned 16-bit value has the same format (resolution) as the fast position channel and is clamped to a maximum of 65535 steps (0xFFFF). The registers are set to the maximum value 0xFFFF at reset.

These registers also allow setting a deviation threshold value for triggering the error event. The threshold value can be written with the same format as the deviation (unsigned 16-bit, same resolution as the fast position channel).

49.2.1.30 DUMMY : Dummy Register

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)
HDSLS1m = 0x9004_1000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x3F



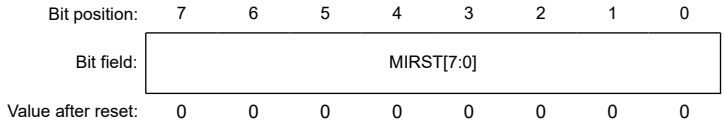
Bit	Symbol	Function	R/W
7:0	DUMMY[7:0]	Due to the transmission of the online-status, Online Status read transactions need less time for transmission. Therefore, dummy read transactions must be inserted to avoid unwanted extra transactions. A read access to this register 0x3F has no effect and must be used for this purpose.	R

When accessing through SPI interface, the dummy read by the DUMMY register in Safe 1 interface is required. The dummy read by the DUMMY register in Drive interface is not required.

49.2.1.31 MIR_STn : Mirror Status n Register (n = 0 to 7)

Base address: HDSLDm = 0x9004_0000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x60 + 0x1 × n



Bit	Symbol	Function	R/W
7:0	MIRST[7:0]	Mirror status, byte n The individual bits indicate different errors, events and warnings. The meaning of each individual bit depends on the particular DSL Slave specification. 0: Encoder in normal status 1: Error, event, or warning status	R/W*1

Note 1. Write 0 only to the bit to be cleared.

The MIR_ST encoder status registers contain all slave system errors, events, and warnings from the DSL encoder. This way, the information can be accessed via drive interface without blocking the short message channel.

The allocation between the individual bits and the slave system statuses depends on the DSL Slave specification. MIR_STn registers (n = 0 to 7) are updated only when a read Short Message is initiated on the corresponding remote status register ENC_STn and the response is read from S_PC_DATA. Trigger for updating the MIR_STn, is reading S_PC_DATA.

Background: initiating a Short Message saves the index of the register, but reading S_PC_DATA triggers the actual copy of register content to the corresponding MIR_STn.

Example:

1. Read Short Message on ENC_ST0
2. Read S_PC_DATA
3. Read access on MIR_ST0 and read PC_DATA content – ENC_ST0 content is mirrored to Drive Interface successfully.

Table 49.8 Relationship between mirror status and SAFE_SUM register

Encoder status	SAFE_SUM bit
MIRST[0]	SSUM0
MIRST[1]	SSUM1
MIRST[2]	SSUM2
MIRST[3]	SSUM3
MIRST[4]	SSUM4
MIRST[5]	SSUM5
MIRST[6]	SSUM6
MIRST[7]	SSUM7

49.2.2 Safe Channel 1 Interface Registers

Table 49.9 Safe channel 1 interface register map

Offset address	Register symbol	Register name
0x09	EDGES*1	Edge Register
0x0A	DELAY*1	Run Time Delay Register
0x0B	VERSION*1	Version Register
0x0D	ENC_ID2*1	Encoder ID 2 Register
0x0E	ENC_ID1*1	Encoder ID 1 Register
0x0F	ENC_ID0*1	Encoder ID 0 Register
0x19	VPOS4	Safe Position 4 Register
0x1A	VPOS3	Safe Position 3 Register
0x1B	VPOS2	Safe Position 2 Register
0x1C	VPOS1	Safe Position 1 Register
0x1D	VPOS0	Safe Position 0 Register
0x1E	VPOSCRC_H	Safe Position CRC High Byte Register
0x1F	VPOSCRC_L	Safe Position CRC Low Byte Register
0x35	SAFE_CTRL	Safe System Control Register
0x36	SAFE_SUM	Safe Status Summary Register
0x37	S_PC_DATA	Parameter Channel Short Message Register
0x3D	EVENT_S	Safe Event Register
0x3E	MASK_S	Safe Event Mask Register
0x3F	DUMMY*1	Dummy Register
0x40 + 0x1 × n	ENC_STn	Encoder Status n Register (n = 0 to 7)
0x7C	SRSSI	Slave RSSI Register
0x7E	MAIL	Slave Mail Register
0x7F	PING	Slave Ping Register

Note 1. These registers are shared with Drive Interface.

49.2.2.1 VPOSn : Safe Position n Register (n = 0 to 4)

Base address: HDSLS1m = 0x9004_1000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x19 (VPOS4)
0x1A (VPOS3)
0x1B (VPOS2)
0x1C (VPOS1)
0x1D (VPOS0)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	VPOS[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	VPOS[7:0]	Safe position, byte n Position value transmitted through safe channel 1 (length: 40 bits), absolute value.	R

The VPOSn registers for the safe position contain the position value from the primary channel of the motor feedback system connected. This safe position is transmitted in every eighth protocol frame.

The safe position is not synchronized with the sync signal.

Only those vpos bits are activated that lie within the range that the motor feedback system has actually measured. All other higher value bits are read as 0. The number of measurable bits can be taken from ENCID bits 9 to 0 in the ENC_ID0 to ENC_ID2 registers.

If Sign is set in the ENC_ID register, the value of the safe position is given signed in the two's complement.

The units of the position value are [steps]. These registers are write-protected.

49.2.2.2 VPOSCRC_H : Safe Position CRC High Byte Register

Base address: HDSLS1m = 0x9004_1000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	VPOSCRCH[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	VPOSCRCH[7:0]	CRC of the safe position, bit 15 to bit 8 16-bit CRC checksum (CRC 16) of the safe position and status summary in Safe Channel 1.	R

49.2.2.3 VPOSCRC_L : Safe Position CRC Low Byte Register

Base address: HDSLS1m = 0x9004_1000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x1F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	VPOSCRCL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	VPOSCRCL[7:0]	CRC of the safe position, bit 7 to bit 0 16-bit CRC checksum (CRC 16) of the safe position and status summary in Safe Channel 1.	R

The VPOSCRC_H/VPOSCRC_L registers contain the CRC checksum of the VPOSn registers and the SAFE_SUM register.

The CRC is checked in the DSL Master. In order to guarantee, in a safety-related application, that the CRC machine in the module is functioning, these registers can be checked with an external cross-check in the diagnostics test interval.

The CRC is generated with the following CRC parameters:

Table 49.10 VPOSCRC parameter

Parameter	Value
CRC sequence	16 bits
CRC polynomial	$X^{16} + X^{15} + X^{12} + X^7 + X^6 + X^4 + X^3 + 1$ Normal representation: 0x90D9
Starting value	0x0000
Closing XOR value	0x00FF
Reverse data bytes	No
Reverse CRC before closing XOR	No

The sequence of the bytes to calculate the CRC is shown in the following figure:

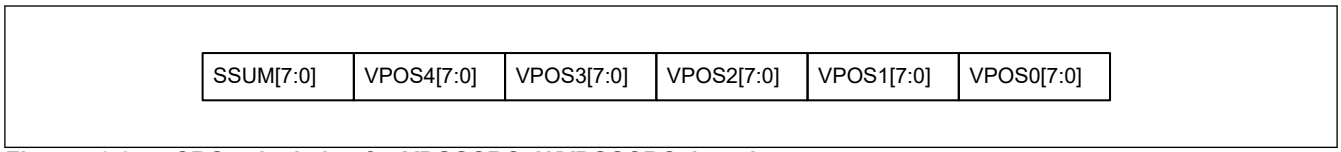


Figure 49.2 CRC calculation for VPOSCRC_H/VPOSCRC_L registers

49.2.2.4 SAFE_CTRL : Safe System Control Register

Base address: HDSLS1m = 0x9004_1000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x35

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PRST	MRST	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
5:0	—	These bits are read as 0. The write value should be 0.	R/W
6	MRST	Message reset 0: Normal Parameters Channel action 1: The Parameters Channel is reset. Current short and long messages are discarded.	R/W
7	PRST	Protocol reset 0: Normal protocol action 1: A forced reset of the protocol status will be initiated. If the bit is deleted, a restart of the connection is triggered.	R/W

The SAFE_CTRL register contains two control bits of the DSL Master available for the safety-related application.

49.2.2.5 SAFE_SUM : Safety Status Summary Register

Base address: HDSLS1m = 0x9004_1000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x36

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SSUM[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SSUM[7:0]	Status summary bit bit [0] (interface): 0: The DSL Slave protocol has not triggered an error, a warning, or an event. 1: An error, a warning, or an event associated with the DSL Slave protocol interface was triggered. bit [7:1] (external resource): 0: The corresponding error, warning, or event is not active. 1: An error, a warning, or an event associated with DSL Slave external resources was triggered.	R

The SAFE_SUM register contains the summarized DSL Slave status information for the safety-related application. It is based on the encoder status ENC_STn (n = 0 to 7). Each status summary bit contains the summarized information from 8 error, warning, and event modes of the DSL Slave. The bits in the status summary register can be read by the safety-related application only.

49.2.2.6 S_PC_DATA : Parameter Channel Short Message Register

Base address: HDSLS1m = 0x9004_1000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x37

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SPCDATA[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	SPCDATA[7:0]	"Short message" Parameter Channel data 8-bit value of the requested remote register.	R

The S_PC_DATA register for the Parameters Channel short message contains the results of "short message" transactions.

"Short message" transactions are generated if operations are carried out with remote registers (DSL Slave). Generally, FRES (in the EVENT_S register) must be set after a transaction is started. Only then will SPCDATA contain valid information.

49.2.2.7 EVENT_S : Safe Event Register

Base address: HDSLS1m = 0x9004_1000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x3D

Bit position:	7	6	5	4	3	2	1	0
Bit field:	SINT	SSUM	SCE	VPOS	QMLW	PRST	MIN	FRES
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	FRES	Channel free for "short message" If the bit is set, the frequency inverter application can trigger a "short message". Provided no answer has been received from the DSL Slave, this bit remains 0. As the processing duration of a "long message" in the motor feedback system is not specified, a time limit condition is installed in the DSL Master. If the time limit is exceeded, attempts are made again automatically. 0: No "short message" can be sent. 1: A "short message" can be sent on the Parameters Channel.	R/W*1
1	MIN	Message init When this warning is displayed, the Parameters Channel is still in the initialization status and no "short message" or "long message" can be triggered. 0: No acknowledgment for the initialization received. 1: An acknowledgment was received from the Slave for the initialization of a message.	R/W*1

Bit	Symbol	Function	R/W
2	PRST	Protocol reset warning This error message indicates that the protocol connection to the DSL Slave has been re-initialized. This error message can be caused by a frequency inverter application request (PRST bit in SYS_CTRL register), a safety-related application request (PRST bit in SAFE_CTRL register), generated by the DSL Master itself, or activated via the reset input. The DSL Master causes a protocol reset if too many transmission errors indicate a connection problem. A protocol reset causes a re-synchronization with the DSL Slave that can improve the connection quality. 0: Normal protocol action 1: The forced protocol reset was triggered.	R/W*1
3	QMLW	Quality monitoring low-value warning This warning indicates that a transmission error occurred. If this error occurs frequently, the wiring of the DSL connection should be checked. 0: Quality monitoring value, QM[3:0] in the MASTERQM register, greater than or equal to "14" 1: Quality monitoring value below "14"	R/W*1
4	VPOS	Safe position error This error usually indicates an encoder sensor error. If this error occurs continuously, there is probably an error in the motor feedback system. 0: The safe position is correct. 1: Sensor error	R/W*1
5	SCE	Error on the Safe Channel This error usually indicates a transmission error on the DSL connection. If this error occurs frequently, the wiring of the DSL connection should be checked. If this error occurs continuously, there is probably an error in the motor feedback system. This error affects quality monitoring and produces the QMLW warning or a protocol reset. 0: Safe Channel data was correctly transmitted. 1: Data consistency error on the Safe Channel.	R/W*1
6	SSUM	Remote event monitoring When the bit is set, an error or a warning has been transmitted from the DSL Slave. The safety-related application must check the SAFE_SUM register to obtain a detailed description. This bit is level sensitive. 0: All DSL Slave events are cleared. 1: The DSL Slave has signaled an event.	R/W*1
7	SINT	Safe interrupt status This bit reflects the status of the safety-related interrupt.	R

Note 1. Write 0 only to the bit to be cleared.

The EVENT_S register contains the message Quality monitoring bits for all warning and error modes of the DSL system.

All messaging bits are set by the DSL Master if a corresponding status is determined.

The following bit description lists the effects of warning and error conditions as well as the reactions to errors that must be installed in the safety-related application.

An event bit that has been set is not reset by the DSL Master. The safety-related application must delete bits that have been set.

Both edge and level-sensitive flags are present in the EVENT_S register. Edge-sensitive bits are set when the corresponding status arises. They are only set again if the corresponding status disappears and then arises once more. This is the standard action. The level-sensitive bits set a bit as long as the corresponding status exists.

49.2.2.8 MASK_S : Safe Event Mask Register

Base address: HDSLS1m = 0x9004_1000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x3E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	MSSUM	MSCE	MVPOS	MQMLW	MPRST	MMIN	MFRES
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	MFRES	Mask for "channel free for "short message"" 0: If a "short message" can be sent on the Parameters Channel, the safety-related interrupt signal is not set. 1: If a "short message" can be sent on the Parameters Channel, the safety-related interrupt signal is set.	W
1	MMIN	Mask for message initialization confirmation 0: The acknowledgment for the initialization of a DSL Slave message does not set the safety-related interrupt signal. 1: The acknowledgment for the initialization of a DSL Slave message sets the safety-related interrupt signal.	W
2	MPRST	Mask for protocol reset warning 0: A protocol reset does not set the safety-related interrupt signal. 1: A protocol reset sets the safety-related interrupt signal.	W
3	MQMLW	Mask for low-quality monitoring value warning 0: A low-quality monitoring value does not set the safety-related interrupt signal. 1: A low-quality monitoring value sets the safety-related interrupt signal.	W
4	MVPOS	Mask for safe position error 0: An error in the safe position does not set the safety-related interrupt signal. 1: An error in the safe position sets the safety-related interrupt signal.	W
5	MSCE	Mask for transmission errors on the Safe Channel 0: A transmission error on the Safe Channel does not set the safety-related interrupt signal. 1: A transmission error on the Safe Channel sets the safety-related interrupt signal.	W
6	MSSUM	Mask for remote event monitoring 0: DSL Slave events that are set in the SAFE_SUM register do not set the safety-related interrupt signal. 1: DSL Slave events that are set in the SAFE_SUM register sets the safety-related interrupt signal.	W
7	—	The write value should be 0.	W

In the MASK_S register, events are chosen that can trigger the safety-related interrupt signal.

Several events can be masked to trigger an interrupt. In addition, events from the DSL Slave summary can be combined with these events.

49.2.2.9 ENC_STn : Encoder Status n Register (n = 0 to 7)

Base address: HDSLS1m = 0x9004_1000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x40 + 0x1 × n

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENCST[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	ENCST[7:0]	Encoder status, byte n The individual bits indicate different errors, events, and warnings. The meaning of each individual bit is determined by the particular DSL Slave installation. ENCST[0]: bit 7 to bit 0 for Parameters Channel ENCST[1]: bit 15 to bit 8 for Parameters Channel ENCST[2]: bit 23 to bit 16 for Parameters Channel ENCST[3]: bit 31 to bit 24 for Parameters Channel ENCST[4]: bit 39 to bit 32 for Parameters Channel ENCST[5]: bit 47 to bit 40 for Parameters Channel ENCST[6]: bit 55 to bit 48 for Parameters Channel ENCST[7]: bit 63 to bit 56 for Parameters Channel 0: Encoder in normal status 1: Error, event, or warning status	R/W*1

Note 1. Write 0 only to the bit to be cleared.

The ENC_STn encoder status registers contain all slave system errors, events, and warnings from the DSL encoder.

The allocation between the individual bits and the slave system statuses depends on DSL Slave specification.

Table 49.11 Relationship between encoder status and SAFE_SUM register

Encoder status	SAFE_SUM bit
ENCST[0]	SSUM0
ENCST[1]	SSUM1
ENCST[2]	SSUM2
ENCST[3]	SSUM3
ENCST[4]	SSUM4
ENCST[5]	SSUM5
ENCST[6]	SSUM6
ENCST[7]	SSUM7

49.2.2.10 SRSSI : Slave RSSI Register

Base address: HDSLS1m = 0x9004_1000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x7C

Bit position:	7	6	5	4	3	2	1	0
Bit field:	—	—	—	—	—	SRSSI[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	SRSSI[2:0]	Value of the Slave RSSI The values for the Slave RSSI range from "0" (poorest signal strength) to "7" (best signal strength).	R
7:3	—	These bits are read as 0.	R

The SRSSI register for indicating the received signal strength at the slave (Slave Received Signal Strength Indicator, RSSI) provides an indication of the strength of the signal arriving at the slave.

The value of the register is only updated from frame to frame if the measurement result deteriorates. After a read access to this register, the register is reset to the value "7" (maximum signal strength).

The register is write-protected.

49.2.2.11 MAIL : Slave Mail Register

Base address: HDSLS1m = 0x9004_1000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x7E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	MAIL[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	MAIL[7:0]	Slave-Mail 8-bit slave mail data for multiple applications	W

The MAIL multi-purpose register of the slave is used for fast communication with the DSL motor feedback system processor. The content of the slave mail register is transmitted to the encoder processor by the most rapid route possible.

The slave mail register is used for the transmission of test messages to the DSL motor feedback system.

49.2.2.12 PING : Slave Ping Register

Base address: HDSLS1m = 0x9004_1000 + 0x1_0000 × m (m = 0, 1)

Offset address: 0x7F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	PING[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	PING[7:0]	Slave-Ping 8-bit hardware version of the DSL encoder at start-up. On first reading, the Ping value of the slave can be used as a connection test on behalf of the slave.	R/W

The PING register of the slave is used to carry out connection tests on behalf of the DSL Slave. The register can be written to and read externally, without this affecting the DSL interface.

On start-up, the register is initialized with the DSL Slave interface.

49.2.3 Safe Channel 2 Interface Registers

Table 49.12 Safe channel 2 interface register map

Address	Register symbol	Register name
0x0B	VERSION2	Version in Safe Channel 2 Register
0x0F	ENC2_ID	Encoder ID in Safe Channel 2 Register
0x18	STATUS2	Safe Channel 2 Status Register
0x19	VPOS24	Safe Position Chanel 2 Byte 4 Register
0x1A	VPOS23	Safe Position Chanel 2 Byte 3 Register
0x1B	VPOS22	Safe Position Chanel 2 Byte 2 Register
0x1C	VPOS21	Safe Position Chanel 2 Byte 1 Register
0x1D	VPOS20	Safe Position Chanel 2 Byte 0 Register
0x1E	VPOSCRC2_H	Safe Position CRC 2 High Byte Register
0x1F	VPOSCRC2_L	Safe Position CRC 2 Low Byte Register
0x3F	DUMMY2	Dummy in Safe Channel 2 Register

49.2.3.1 VERSION2 : Version in Safe Channel 2 Register

Base address: —

Offset address: 0x0B

Bit position:	7	6	5	4	3	2	1	0
Bit field:	CODE[1:0]		MAJOR[1:0]		MINOR[3:0]			
Value after reset:	0	1	0	1	0	1	1	1

Bit	Symbol	Function	R/W
3:0	MINOR[3:0]	Minor release number The current version is 7.	R
5:4	MAJOR[1:0]	Major release number The current version is 1.	R

Bit	Symbol	Function	R/W
7:6	CODE[1:0]	Type of module 0 1: DSL Master Others: Reserved	R

The VERSION2 register in safe channel 2 is identical to the VERSION register in the drive and safe channel 1 interfaces.

49.2.3.2 ENC2_ID : Encoder ID in Safe Channel 2 Register

Base address: —

Offset address: 0x0F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	ENC2ID[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	ENC2ID[7:0]	Designation of the safe channel 2 channel **0x02/0x22:** SIL2 arrangement. The same position data is transmitted to the safe channel 2 channel as to the safe channel 1 channel. **0x03/0x33:** SIL3 arrangement. Position data is transmitted to the safe channel 2 channel that originates from the secondary sensor in the HIPERFACE DSL® motor feedback system.	R

The ENC2_ID register for the ID of the encoder in safe channel 2 contains the designation code for the safe channel 2 channel of the DSL encoder.

The contents of ENC2_ID are defined during the actual installation of the motor feedback system and can be found in the corresponding data sheet.

The register is write-protected.

49.2.3.3 STATUS2 : Safe Channel 2 Status Register

Base address: —

Offset address: 0x18

Bit position:	7	6	5	4	3	2	1	0
Bit field:	TOG2	TEST2	ERR2	FIX2[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
4:0	FIX2[4:0]	Safe Channel 2 Fixed Bit Pattern FIX2 is a fixed bit pattern for each SPI transaction that indicates errors arising in the SPI interface due to unchanged bit values. The standard value of the fixed bit pattern is 11100b. All other values indicate an error on safe channel 2 of the DSL system. Suitable measures must be installed in the user application.	R
5	ERR2	Safe Channel 2 Position Error 0: The last safe position received in safe channel 2 is correct. 1: The last safe position received in safe channel 2 is invalid. Suitable measures must be installed in the user application.	R
6	TEST2	Safe Channel 2 Test TEST2 bit is set to 1 if a test message is executed via the MAIL register while the safe channel 2 status and position data are valid. TEST2 can only be valid if the user application has previously requested a test. Corresponding error indications for TEST2 are either the ERR2 bit or a discrepancy between the position and the CRC of safe channel 2.	R

Bit	Symbol	Function	R/W
7	TOG2	Safe Channel 2 Toggle For successive position transmissions on safe channel 2, TOG2 must always toggle between 0 and 1. The starting value for TOG2 is 0. If the toggle bit does not change its value, it is probable that a transmission error occurred and the transmitted absolute value for safe channel 2 is invalid. Suitable measures must be installed in the user application.	R

The STATUS2 status register contains the status information for safe channel 2 of the HIPERFACE DSL® motor feedback system.

49.2.3.4 VPOS2n : Safe Position Channel 2 Byte n Register

Base address: —

Offset address: 0x19 (VPOS24)
0x1A (VPOS23)
0x1B (VPOS22)
0x1C (VPOS21)
0x1D (VPOS20)

Bit position:	7	6	5	4	3	2	1	0
Bit field:	VPOS2[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	VPOS2[7:0]	Safe position, Channel 2, byte n Position value at motor feedback system Safe Channel 2 (length: 40 bits), transmitted as an absolute value complement.	R

The VPOS2n registers contain the position value from the secondary channel of the motor feedback system connected. This safe position is transmitted in every eighth protocol frame if the validity of the data transfer has been checked.

The safe position is not synchronized with the most recent sync signal but to the (user or internally generated) sync signal of one of the last 8 protocol frames. Only those VPOS2 bits are relevant that lie within the range that the motor feedback system has actually measured. Also, typically channel 2 has a lower resolution than channel 1. The operating manual and datasheet of the product specify the resolution of each channel.

The units of the position value are [steps].

These registers are write-protected.

49.2.3.5 VPOSCRC2_H : Safe Position CRC 2 High Byte Register

Base address: —

Offset address: 0x1E

Bit position:	7	6	5	4	3	2	1	0
Bit field:	VPOSCRC2H[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	VPOSCRC2H[7:0]	CRC of the safe position 2, bit 15 to bit 8 16-bit CRC checksum (CRC 16) of the safe position and status summary in Safe Channel 2.	R

49.2.3.6 VPOSCRC2_L : Safe Position CRC 2 Low Byte Register

Base address: —

Offset address: 0x1F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	VPOSCRC2L[7:0]							
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
7:0	VPOSCRC2L[7:0]	CRC of the safe position 2, bit 7 to bit 0 16-bit CRC checksum (CRC 16) of the safe position and status summary in safe channel 2.	R

The VPOSCRC2_H/VPOSCRC2_L registers for the position checksum in safe channel 2 contain the CRC checksum for the safe position VPOS2n.

The CRC is checked in the DSL MasterIP Core. In order to guarantee, in a safety-related application, that the CRC machine is functioning, these registers can be checked with an external cross-check in the diagnostics test interval.

The CRC is generated with the following CRC parameters:

Table 49.13 VPOSCRC2 parameter

Parameter	Value
CRC sequence	16 bits
CRC polynomial	$X^{16} + X^{15} + X^{12} + X^7 + X^6 + X^4 + X^3 + 1$ Normal representation: 0x90D9
Starting value	0x0000
Closing XOR value	0x00FF
Reverse data bytes	No
Reverse CRC before closing XOR	No

The sequence of the bytes to calculate the CRC is shown in the following figure:

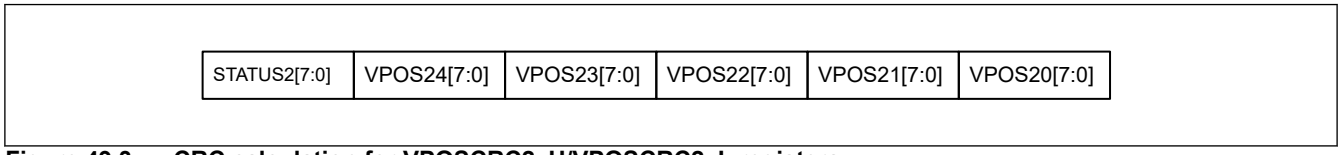


Figure 49.3 CRC calculation for VPOSCRC2_H/VPOSCRC2_L registers

49.2.3.7 DUMMY2 : Dummy in Safe Channel 2 Register

Base address: —

Offset address: 0x3F

Bit position:	7	6	5	4	3	2	1	0
Bit field:	DUMMY[7:0]							
Value after reset:	x	x	x	x	x	x	x	x

Bit	Symbol	Function	R/W
7:0	DUMMY[7:0]	Due to the transmission of the online-status, Online status read transactions need less time for transmission. Therefore, dummy read transactions must be inserted to avoid unwanted extra transactions. A read access to this register 0x3F has no effect and must be used for this purpose.	R

When accessing through SPI interface, the dummy read by the DUMMY2 register in the safe channel 2 is required.

49.3 Operation

49.3.1 System Start

As soon as the motor feedback system is supplied with power, a forced reset ensures that a defined system start status is produced in the DSL Master.

Figure 49.4 shows the status table for system start.

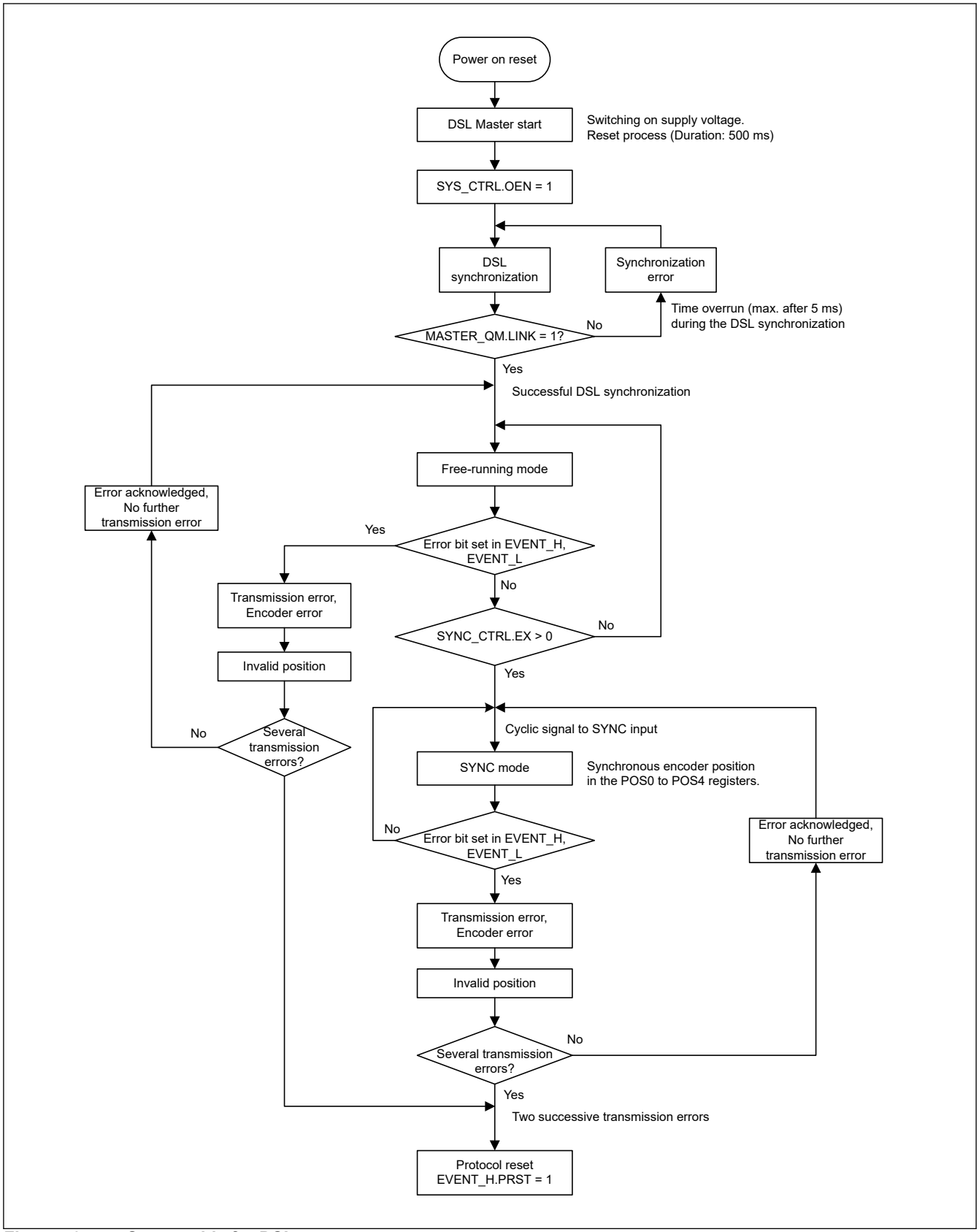


Figure 49.4 Status table for DSL system start

49.3.2 Quality Monitoring

The MASTERQM register contains the quality monitoring value for the data connection.

As soon as the DSL Master detects events that indicate an improvement or degradation of the quality of the data connection, these events are indicated as values higher or lower than the quality monitoring value.

Table 49.14 Quality monitoring events

Protocol event	Value change in quality monitoring
Wrong synchronization in Safe Channel (last byte)	-4
Wrong synchronization in Safe Channel (1st … 7th bytes)	-6
RSSI < 2	-4
Wrong encoding in parameter	-1
Wrong encoding in process data channel	-2
Unknown special characters in the protocol package	-2
Any identified error in Safe Channel 1	-6
Any identified error in Safe Channel 2	-8
Correct synchronization in Safe Channel	+1
Correct CRC value in Safe Channel	+1

Quality monitoring is initiated with the value "8". The maximum quality monitoring value is "15". This is the standard value during operation.

Note: Particular attention must be paid to the quality monitoring value during the development of a DSL drive controller. If a value lower than "15" is indicated, the cause may be a problem with the connection circuit, particularly if the value is continuously displayed.

If the quality monitoring value falls below "14", QMLW information is indicated in the EVENT_L register.

If the quality monitoring value falls to "0", a forced reset of the protocol is carried out. This is indicated by the PRST error bit in the EVENT_H register.

49.3.3 Interrupt

The interrupt signal is set to 1 if an interrupt condition has been fulfilled in the DSL Master. The interrupt conditions are set using the registers MASKH, MASKL, and MASKSUM. The interrupt conditions for safety are set using register MASKS.

Figure 49.5 and Figure 49.6 show an interrupt condition as a diagram.

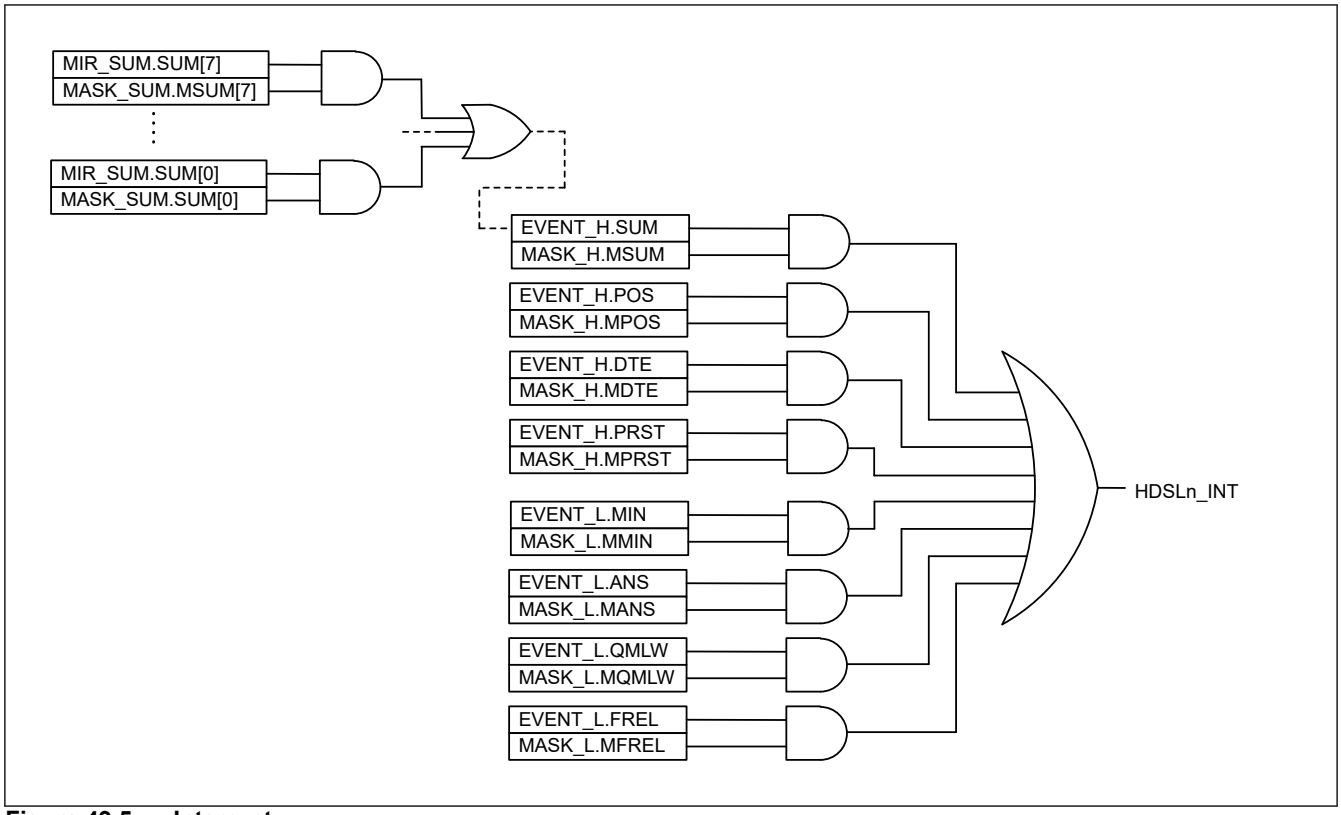


Figure 49.5 Interrupt

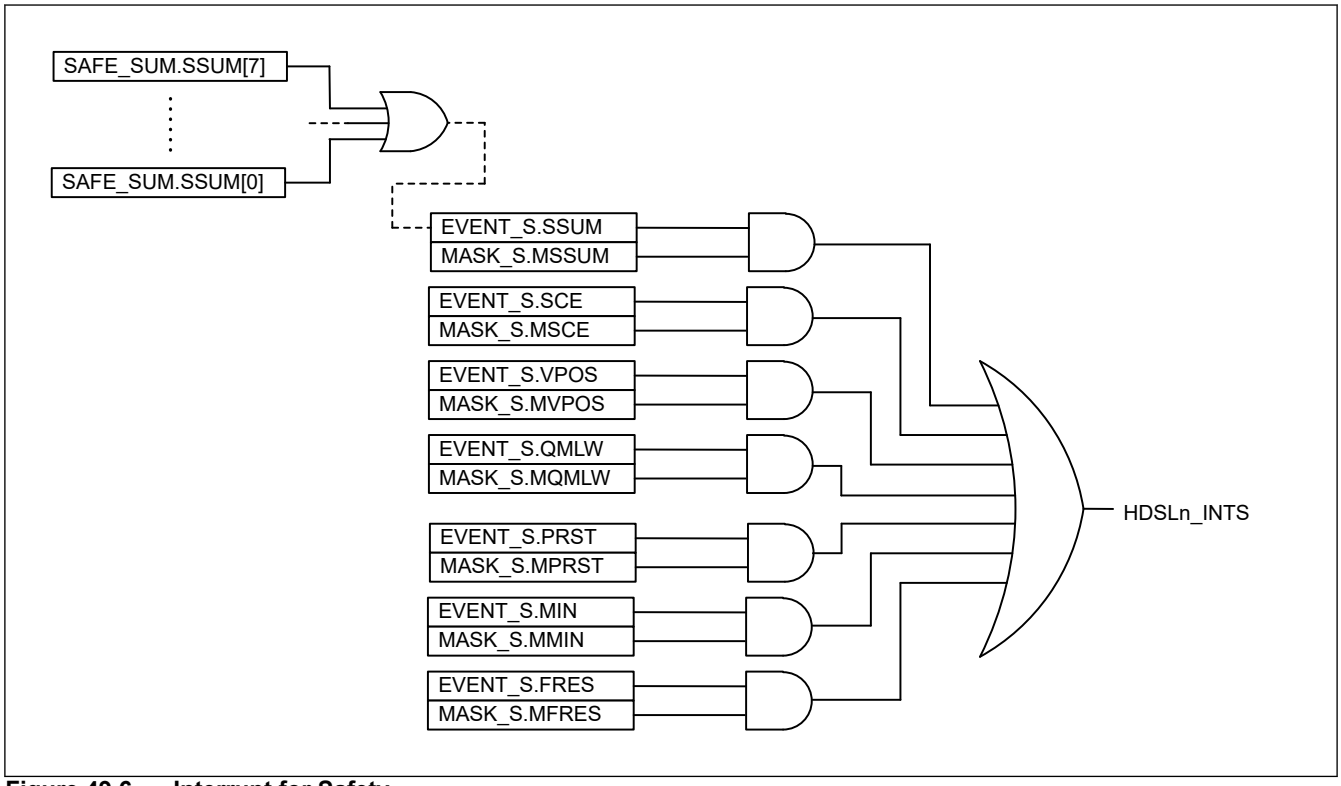


Figure 49.6 Interrupt for Safety

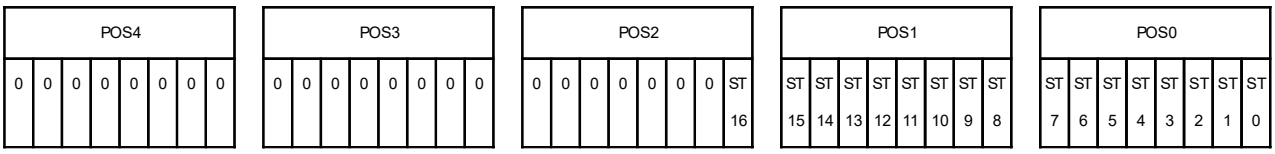
49.3.4 Fast Position

The fast position and the rotation speed of the encoder shaft are transmitted on the DSL motor feedback system process data channel. These values are the main process values for the drive application control circuit.

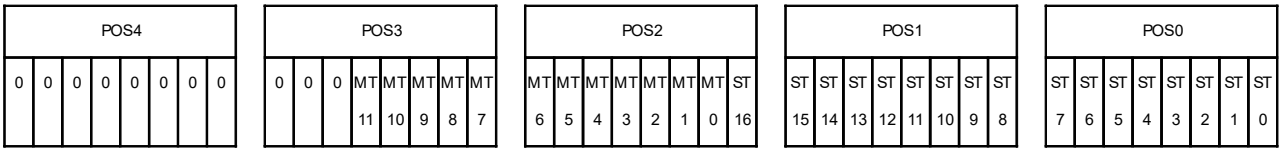
HIPERFACE DSL stores the fast position in the POS0 to POS4 HDSL Master registers and the rotation speed in the VEL0 to VEL2 registers.

The position is given as a 40 bit value that includes the angle setting ("single-turn" value) and the number of rotations ("multi-turn" value). Only the position bits actually measured by the motor feedback system are accessible and are stored in the registers as a right-justified value. The other (higher value) bits are constantly set at "0" (see examples "a" to "c" below).

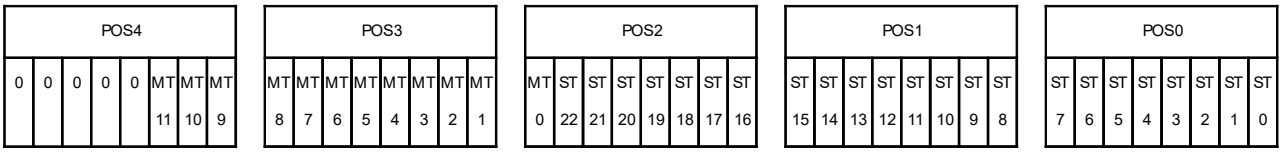
a. Single-turn 17 bit



b. Single-turn 17 bit, Multi-turn 12 bit



c. Single-turn 23 bit, Multi-turn 12 bit



The motor feedback system fast position is sampled and transmitted if the DSL Master receives a SYNC signal.

This SYNC signal can be created in two different ways (see section 49.3.4.1. Free running mode and section 49.3.4.2. SYNC mode).

49.3.4.1 Free running mode

In free-running mode, the SYNC signal is automatically created by the DSL Master, for which the maximum frame transmission frequency is used. The free-running mode is the standard DSL Master operating mode at start-up. This operating mode can also be selected manually, by setting the ES value in the SYNC_CTRL register to 0.

49.3.4.2 SYNC mode

In SYNC mode, the DSL Master depends on a prepared cyclic control signal. This control signal triggers position measurements and enables polling of position and rotation speed values synchronously with the control signal. The control signal must be applied to the SYNC input and have the characteristics prescribed for the DSL Master.

The position is available after a set delay in relation to the leading edge of the control signal.

When SYNC mode is used the following points must be noted:

1. A correct control signal must be applied at the sync input. The signal must correspond with the specifications for pulse width and cycle time.
2. Setting or deleting the SPOL bit in the SYS_CTRL register determines whether the position measurements are to be triggered by the leading or the trailing edge of the control signal. The set latency of the DSL system is measured from this edge.
3. The correct ES divider must be set in the SYNC_CTRL register. This divider determines how many position samplings and transmissions will be undertaken for each control signal. Do not change setting of ES divider during start-up or operation.

50. A-Format (AFMT)

50.1 Overview

The MPU includes A-format (AFMT) master to enable communication between the encoder with A-format interface. Table 50.1 describes the A-format specifications and Figure 50.1 shows a block diagram of the AFMT.

Table 50.1 A-format (AFMT) specifications

Item	Description
Number of units	2 units
Protocol	A-format Ver.2.0
Functions	• Communication with absolute encoder supporting A-format • Up to 8 slaves bus connection each unit • Compliant to A-format Version 2.0 and A-Safety-format Version 1.0 • Trigger input – CPU control (SW trigger) – HW trigger • Encoder status • Communication alarm • Status and error messages • Time out
Interrupt sources	• Two interrupts – Communication completion – Time out error
Event link	nEXT_STRG input for HW trigger provided from ELC
Module-stop function	Module-stop state can be set to reduce power consumption

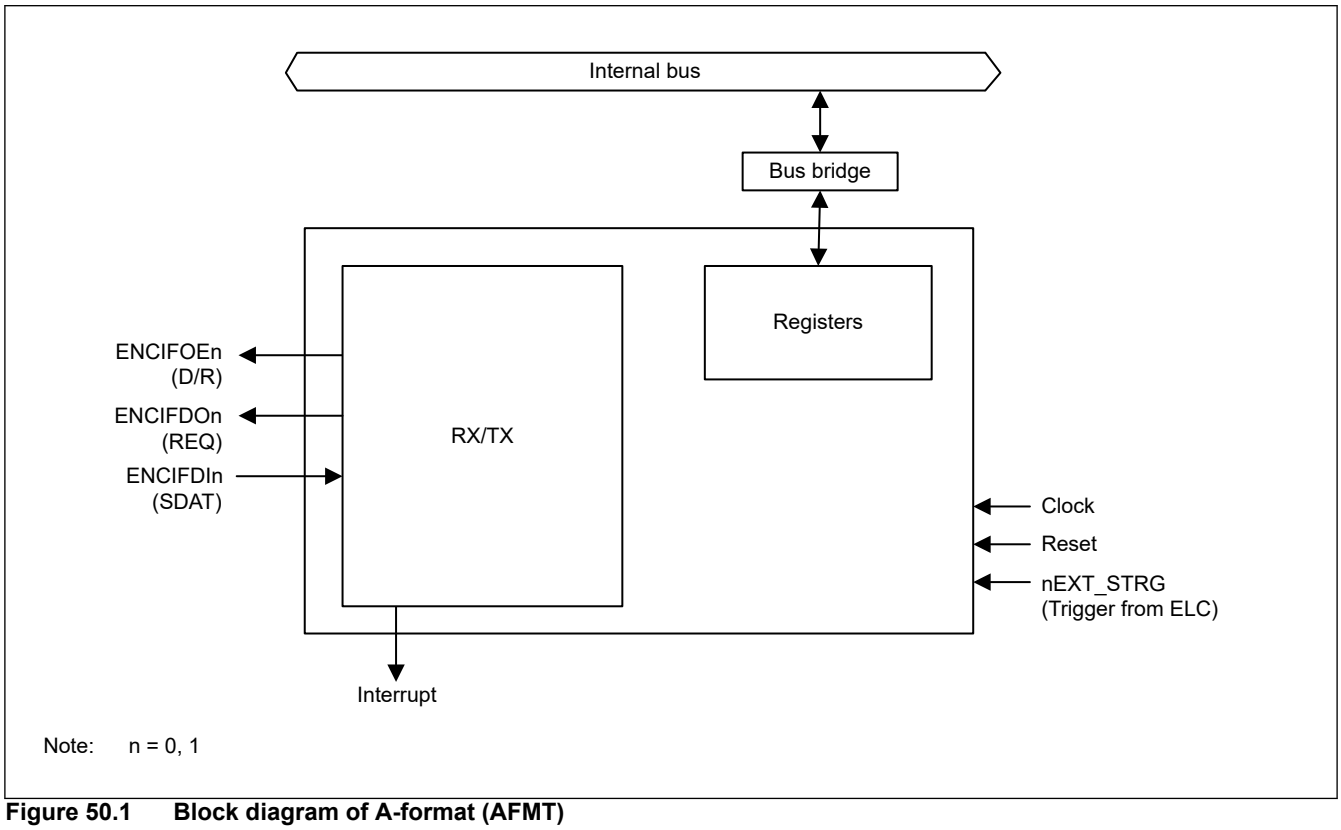


Figure 50.1 Block diagram of A-format (AFMT)

Table 50.2 lists the input/output pins of this module.

Table 50.2 PIN configuration of the AFMT

Unit	Pin name	I/O	Function
AFMT0	ENCIFOE0 (D/R)	Output	A-format unit 0 line transceiver control signal
	ENCIFDO0 (REQ)	Output	A-format unit 0 serial data output
	ENCIFDI0 (SDAT)	Input	A-format unit 0 serial data input
AFMT1	ENCIFOE1 (D/R)	Output	A-format unit 1 line transceiver control signal
	ENCIFDO1 (REQ)	Output	A-format unit 1 serial data output
	ENCIFDI1 (SDAT)	Input	A-format unit 1 serial data input

Table 50.3 AFMT interrupt sources

Name	Interrupt sources	CPU GIC request	DMAC activation	ELC activation
AFMTi_EOF	Communication completion	Possible	Possible	Not possible
AFMTi_TMOUT	Time out error	Possible*1	Not possible	Not possible

Note: i = 0, 1
Note 1. Some errors from other peripherals are ORed and connected to the GIC.

50.2 Register Descriptions

AFMT register map is below. A part of the offset address of the register is shared between writing and reading.

Table 50.4 AFMT register map (for writing)

Offset address	Register symbol	Register name
0x000	COMMAND	Command Code Encoder Address Register
0x004	ROMDATA	EEPROM Address and Write Data Register
0x008	ID	ID Code Register
0x100	TRGSEL	Trigger Select Register
0x104	TXTRG	Transmission Trigger Register
0x200	BRSEL	Baud Rate Setting Register

Table 50.5 AFMT register map (for reading)

Offset address	Register symbol	Register name
0x004 × m + 0x020 × (n – 1)	ENCnRXDATAm	ENCn Receive Data m Register (n = 1 to 8, m = 0 to 7)

50.2.1 COMMAND : Command Code Encoder Address Register

Base address: AFMTi = 0x9004_5000 + 0x1_0000 × i (i = 0, 1)

Offset address: 0x000

Bit position:	31																			12			8	7			3	2		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	SID[4:0]				CC[4:0]				XEA[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	XEA[2:0]	Encoder address Please see Table 50.6.	W
7:3	CC[4:0]	Command code Please see Table 50.7.	W
12:8	SID[4:0]	Subcommand ID Please see Table 50.8.	W

Bit	Symbol	Function	R/W
31:13	—	The write value should be 0.	W

Table 50.6 Encoder address

XEA2 / EA2	XEA1 / EA1	XEA0 / EA0	Encoders
0	0	0	ENC1
0	0	1	ENC2
0	1	0	ENC3
0	1	1	ENC4
1	0	0	ENC5
1	0	1	ENC6
1	1	0	ENC7
1	1	1	ENC8

Table 50.7 Command code (1 of 2)

Frame name	CC[4:0]	Classification	Remarks
CDF0	00000b	Data request (Individual mode)	ABS full 40-bit data request
CDF1	00001b		ABS lower 24-bit data request
CDF2	00010b		ABS upper 24-bit data request
CDF3	00011b		Encoder status request
CDF4	00100b	Data request (multiple mode)	ABS full 40-bit data request
CDF5	00101b		ABS lower 24-bit data request
CDF6	00110b		ABS upper 24-bit data request
CDF7	00111b		Encoder status request
CDF8	01000b	Operating request (Individual mode)	Status flag clear request (8 times in a row)
CDF9	01001b		Multi-rotation data clear request (8 times in a row)
CDF10	01010b		Status and multi-rotation data clear requests (8 times in a row)
CDF11	01011b		Encoder address setting I (8 times in a row)
CDF12	01100b		One-rotation data zero preset (8 times in a row)
CDF13	01101b	Memory access request (Individual mode)	Memory read request (2 frames)
CDF14	01110b		Memory write request (4 frames)
CDF15	01111b	Data request (Individual mode)	10-bit temperature data request
CDF16	10000b	ID code read	ID code read I (Individual)
CDF17	10001b		ID code read II (peer-to-peer connection)
CDF18	10010b	ID code write	ID code write I (Individual, 4 frames)
CDF19	10011b		ID code write II (peer-to-peer connection, 4 frames)
CDF20	10100b	Reserved	This command is not supported.
CDF21	10101b	Data request (Individual)	ABS lower 17-bit data request
CDF22	10110b	Data request (Multiple)	ABS lower 17-bit data request
CDF27	11011b	Data request (Individual)	ABS lower 24-bit data and status requests
CDF28	11100b	Data request (Multiple)	ABS lower 24-bit data and status requests
CDF29	11101b	Data request (Individual)	ABS lower 24-bit data and temperature data requests
CDF30	11110b	Data request (Multiple)	ABS lower 24-bit data and temperature data requests
CDF31	11111b	Operating request (Individual)	A-Safety Format

Table 50.7 Command code (2 of 2)

Frame name	CC[4:0]	Classification	Remarks
CDF23 - 26	Others	No definition	

Table 50.8 Subcommand ID

Frame name	SID[4:0]	Classification	Remarks
SID0	00000b	Data request (Individual transmission)	Pos1(39:0), Pos2(31:0), Diag_p(15:0), Diag_n(15:0), DB(7:0)
SID1	00001b	Data request (Multiple transmission)	
SID2	00010b	Data request (Individual transmission)	Pos1(39:0), Pos2(31:0), Diag_p(15:0), Diag_n(15:0)
SID3	00011b	Data request (Multiple transmission)	
SID4	00100b	Data request (Individual transmission)	Pos1(23:0), Pos2(31:0), Diag_p(15:0), Diag_n(15:0), DB(7:0)
SID5	00101b	Data request (Multiple transmission)	
SID6	00110b	Data request (Individual transmission)	Pos1(23:0), Pos2(31:0), Diag_p(15:0), Diag_n(15:0)
SID7	00111b	Data request (Multiple transmission)	
SID24 - 30	Command for power-on test of Safety encoder		
	Others	No definition	

50.2.2 ROMDATA : EEPROM Address and Write Data Register

Base address: AFMTi = 0x9004_5000 + 0x1_0000 × i (i = 0, 1)

Offset address: 0x004

Bit position:	31								23								16	15														0
Bit field:	—	—	—	—	—	—	—	—	ADDR[7:0]									DATA[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	DATA[15:0]	16-bit EEPROM data When the command code is a memory writing request (CDF14), 16-bit write data value should be written in the ROMDATA register with writing in the ROMADDR register.	W
23:16	ADDR[7:0]	8-bit EEPROM address When the command code is a memory reading request (CDF13), 16-bit read address should be written in the ROMADDR register. When command code is a memory writing request (CDF14), 16-bit read address should be written as well.	W
31:24	—	The write value should be 0.	W

50.2.3 ID : ID Code Register

Base address: AFMTi = 0x9004_5000 + 0x1_0000 × i (i = 0, 1)

Offset address: 0x008

Bit position:	31								23								16	15														0
Bit field:	—	—	—	—	—	—	—	—	ID_H[7:0]									ID_L[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	ID_L[15:0]	24-bit ID code, low word When the command code is ID code writing request (CDF18 or 19), 24-bit ID code should be written in the ID_H and ID_L bits.	W

Bit	Symbol	Function	R/W
23:16	ID_H[7:0]	24-bit ID code, high byte When the command code is ID code writing request (CDF18 or 19), 24-bit ID code should be written in ID_H and ID_L bits.	W
31:24	—	The write value should be 0.	W

50.2.4 TRGSEL : Trigger Select Register

Base address: AFMTi = 0x9004_5000 + 0x1_0000 × i (i = 0, 1)

Offset address: 0x100

Bit position:	31																													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EXT
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	EXT	Trigger select 0: Set TRG bit in ENCnTXSEL register 1: Trigger from ELC	W
31:1	—	The write value should be 0.	W

50.2.5 TXTRG : Transmission Trigger Register

Base address: AFMTi = 0x9004_5000 + 0x1_0000 × i (i = 0, 1)

Offset address: 0x104

Bit position:	31																													0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TRG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	TRG	Command data transmission trigger generation When EXT bit in ECNnTRGSEL is set to 0, transmission trigger is generated by this bit setting and command data transmission to specified Encoder starts. When EXT = 1, this bit setting is ignored. 0: Do nothing. 1: Transmission trigger is generated.	W
31:1	—	The write value should be 0.	W

50.2.6 BRSEL : Baud Rate Setting Register

Base address: AFMTi = 0x9004_5000 + 0x1_0000 × i (i = 0, 1)

Offset address: 0x200

Bit position:	31																			11									4	3	2		0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	TM[7:0]										AS	BR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
2:0	BR[2:0]	Baud rate setting The baud rate is required a value depending on the encoder used. 0 0 0: 2.5 Mbps 0 0 1: 4 Mbps 0 1 0: 6.67 Mbps 0 1 1: 8 Mbps 1 0 0: 16 Mbps Others: Setting prohibited	W
3	AS	A-Safety enable 0: Disable 1: Enable	W
11:4	TM[7:0]	t2 time setting N t2 time is set to 1 µs + N × 0.25 µs	W
31:12	—	The write value should be 0.	W

50.2.7 ENCnRXDATAm : ENCn Receive Data m Register (n = 1 to 8, m = 0 to 7)

Base address: AFMTi = 0x9004_5000 + 0x1_0000 × i (i = 0, 1)

Offset address: 0x004 × m + 0x020 × (n - 1)

Bit position:	31																												0
Bit field:	DATA[31:0]																												
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
31:0	DATA[31:0]	Encoder read data For details, see Table 50.9 and Table 50.10.	R

The ENCnRXDATAm registers are used to receive data which encoder outputs according to the transmitted command.

Table 50.9 Receive data (except CDF31) (1 of 2)

Command code	ENCnRXDATAm	DATA[31:0]																															
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDF0, CDF4	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[3:0]				ES[3:0]				CC[4:0]					EA[2:0]		
	ENCnRXDATA1	ABS 40-bit full data (ABS[31:0])																															
	ENCnRXDATA2																	0	0	0	0	0	0	0	0	ABS 40-bit full data (ABS[39:32])							
CDF1, CDF5	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[3:0]				ES[3:0]				CC[4:0]					EA[2:0]		
	ENCnRXDATA1	0	0	0	0	0	0	0	0	ABS lower 24-bit data (ABS[23:0])																							
CDF2, CDF6	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[3:0]				ES[3:0]				CC[4:0]					EA[2:0]		
	ENCnRXDATA1									ABS upper 24-bit data (ABS[39:16])																							
CDF3, CDF7 to CDF12	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[3:0]				ES[3:0]				CC[4:0]					EA[2:0]		
	ENCnRXDATA1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Status code (ALM[15:0])															
CDF13, CDF14	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[3:0]				ES[3:0]				CC[4:0]					EA[2:0]		
	ENCnRXDATA1	0	0	0	0	0	0	0	0	8-bit EEPROM address								16-bit EEPROM data															
CDF15	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[3:0]				ES[3:0]				CC[4:0]					EA[2:0]		
	ENCnRXDATA1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10-bit temperature data (DB[9:0])									

Table 50.9 Receive data (except CDF31) (2 of 2)

Command code	ENCnRXDATAm	DATA[31:0]																															
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CDF16 to CDF19	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[3:0]				ES[3:0]				CC[4:0]					EA[2:0]		
	ENCnRXDATA1	0	0	0	0	0	0	0	0	24-bit ID code [23:0]																							
CDF21, CDF22	ENCnRXDATA0	CA[3:1]			0	ES	EA[2:0]			0	0	0	0	0	0	0	ABS lower 17-bit data (ABS[16:0])																
CDF27, CDF28	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[3:0]				ES[3:0]				CC[4:0]					EA[2:0]		
	ENCnRXDATA1	0	0	0	0	0	0	0	0	ABS lower 24-bit data (ABS[23:0])																							
	ENCnRXDATA2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Status code (ALM[15:0])															
CDF29, CDF30	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[3:0]				ES[3:0]				CC[4:0]					EA[2:0]		
	ENCnRXDATA1	0	0	0	0	0	0	0	0	ABS lower 24-bit data (ABS[23:0])																							
	ENCnRXDATA2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10-bit temperature data (DB[9:0])									

Note: CA[3:0]: Communication alarm. For details, see Table 50.11.
ES[3:0]: Encoder status. For details, see Table 50.12.
CC[4:0]: Command code. For details, see Table 50.7.
EA[2:0]: Encoder address. For details, see Table 50.6.
ALM[15:0]: Status code. For details, see Table 50.13.

Table 50.10 Receive data (CDF31) (1 of 2)

Command code	ENCnRXDATAm	DATA[31:0]																															
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID0, SID1	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[4:0]					SQD[2:0]			SID[4:0]					EA[2:0]		
	ENCnRXDATA1	Pos1[31:0]																															
	ENCnRXDATA2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DTB[7:0]								Pos1[39:32]							
	ENCnRXDATA3	Pos2[31:0]																															
	ENCnRXDATA4	Diag_N[15:0]																Diag_P[15:0]															
SID2, SID3	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[4:0]					SQD[2:0]			SID[4:0]					EA[2:0]		
	ENCnRXDATA1	Pos1[31:0]																															
	ENCnRXDATA2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	Pos1[39:32]							
	ENCnRXDATA3	Pos2[31:0]																															
	ENCnRXDATA4	Diag_N[15:0]																Diag_P[15:0]															
SID4, SID5	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[4:0]					SQD[2:0]			SID[4:0]					EA[2:0]		
	ENCnRXDATA1	DTB[7:0]								Pos1[23:0]																							
	ENCnRXDATA2	Pos2[31:0]																															
	ENCnRXDATA3	Diag_N[15:0]																Diag_P[15:0]															
SID6, SID7	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[4:0]					SQD[2:0]			SID[4:0]					EA[2:0]		
	ENCnRXDATA1	0	0	0	0	0	0	0	0	Pos1[23:0]																							
	ENCnRXDATA2	Pos2[31:0]																															
	ENCnRXDATA3	Diag_N[15:0]																Diag_P[15:0]															

Table 50.10 Receive data (CDF31) (2 of 2)

Command code	ENCnRXDATAm	DATA[31:0]																													
		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SID24 to SID30	ENCnRXDATA0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	CA[4:0]					SQD[2:0]			SID[4:0]					EA[2:0]		
	ENCnRXDATA1	Pos1[31:0]																															
	ENCnRXDATA2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	DTB[7:0]								Pos1[39:32]							
	ENCnRXDATA3	Pos2[31:0]																															
	ENCnRXDATA5	0	0	0	0	0	0	0	0	ALM[23:0]																							

Table 50.11 Communication alarm (CA[4:0])

Bit	Symbol	Function
0	CA[0]	CMD status flag When this bit is set to 1, indicates transmitted command is not matched with received command.
1	CA[1]	FORM status flag When this bit is set to 1, indicate start bit and stop bit mismatch for the field or a lack of receive data length.
2	CA[2]	SYNC status flag When this bit is set to 1, indicates receive data does not come within the designated time after command code transmission.
3	CA[3]	CRC status flag When this bit is set to 1, indicates transmitted CRC is not matched with CRC calculated from received data.
4	CA[4]	SQN status flag When this bit is set to 1, indicates transmitted sequence number is not matched with received sequence number. (CDF31 only)

Table 50.12 Encoder status (ES[3:0] and ES)

Bit	Symbol	Function
0	ES[0]	BUSY + MEMBUSY
1	ES[1]	BATT
2	ES[2]	OVSPD + MEMERR
3	ES[3]	STERR + PSERR + MTERR
—	ES	ORed ES[3:0]. (CDF21 and CDF22 only)

Note: Confirm the specification document of the encoder to be used.

Table 50.13 Status code (ALM[15:0]) (1 of 2)

Bit	Symbol	Function
0	ALM[0]	BATT
1	ALM[1]	MTERR
2	ALM[2]	0 fixed
3	ALM[3]	OVSPD
4	ALM[4]	MEMERR
5	ALM[5]	STERR
6	ALM[6]	PSERR
7	ALM[7]	BUSY
8	ALM[8]	MEMBUSY
9	ALM[9]	OVTEMP
10	ALM[10]	0 fixed
11	ALM[11]	0 fixed

Table 50.13 Status code (ALM[15:0]) (2 of 2)

Bit	Symbol	Function
12	ALM[12]	0 fixed
13	ALM[13]	0 fixed
14	ALM[14]	0 fixed
15	ALM[15]	0 fixed

Note: Confirm the specification document of the encoder to be used.

The status when the main power is turned on (250 min^{-1} or less) when the external battery is not connected and the internal backup capacitor is completely discharged is as follows:

BUSY, STERR, MEMBUSY, MEMERR, OVTEMP = 0

MTERR, BATT = 1

OVSPD, PSERR = x (indefinite value)

50.3 Operation

50.3.1 Basic Operation

After writing a command and memory data such as read address and write data, command data transmission starts by register-write or trigger input according to the TRGSEL register setting.

(1) Startup by register access

When the EXT bit in the TRGSEL register is set to 0, the register trigger is selected.

In this mode, when the TRG bit in the TXTRG register is set to 1, command data transmission starts.

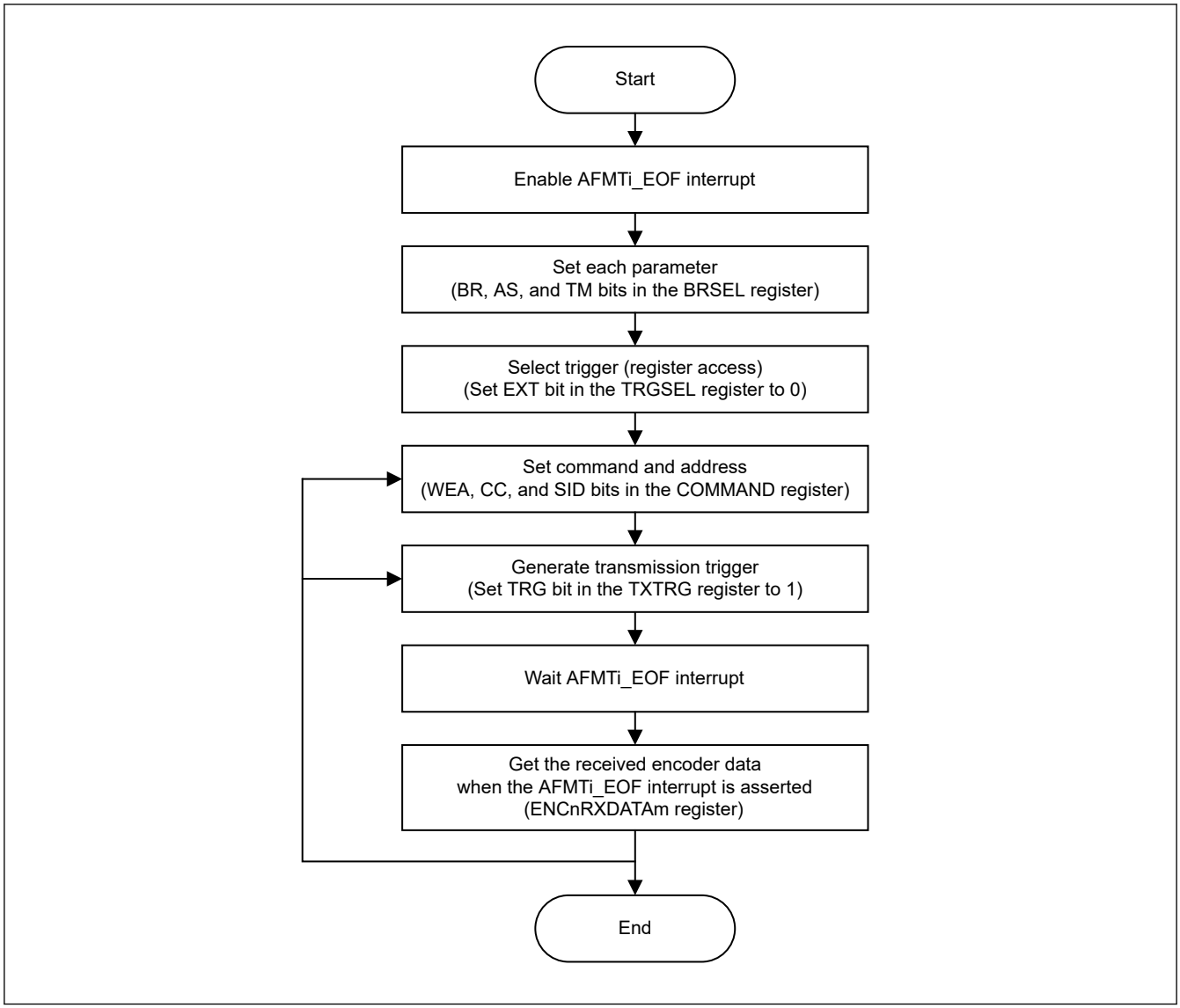


Figure 50.2 Basic operation (startup by register access)

(2) Startup by trigger input from ELC

When the EXT bit in the TRGSEL register is set to 1, the trigger from ELC is selected.

In this mode, when the trigger input is received from ELC, command data transmission starts.

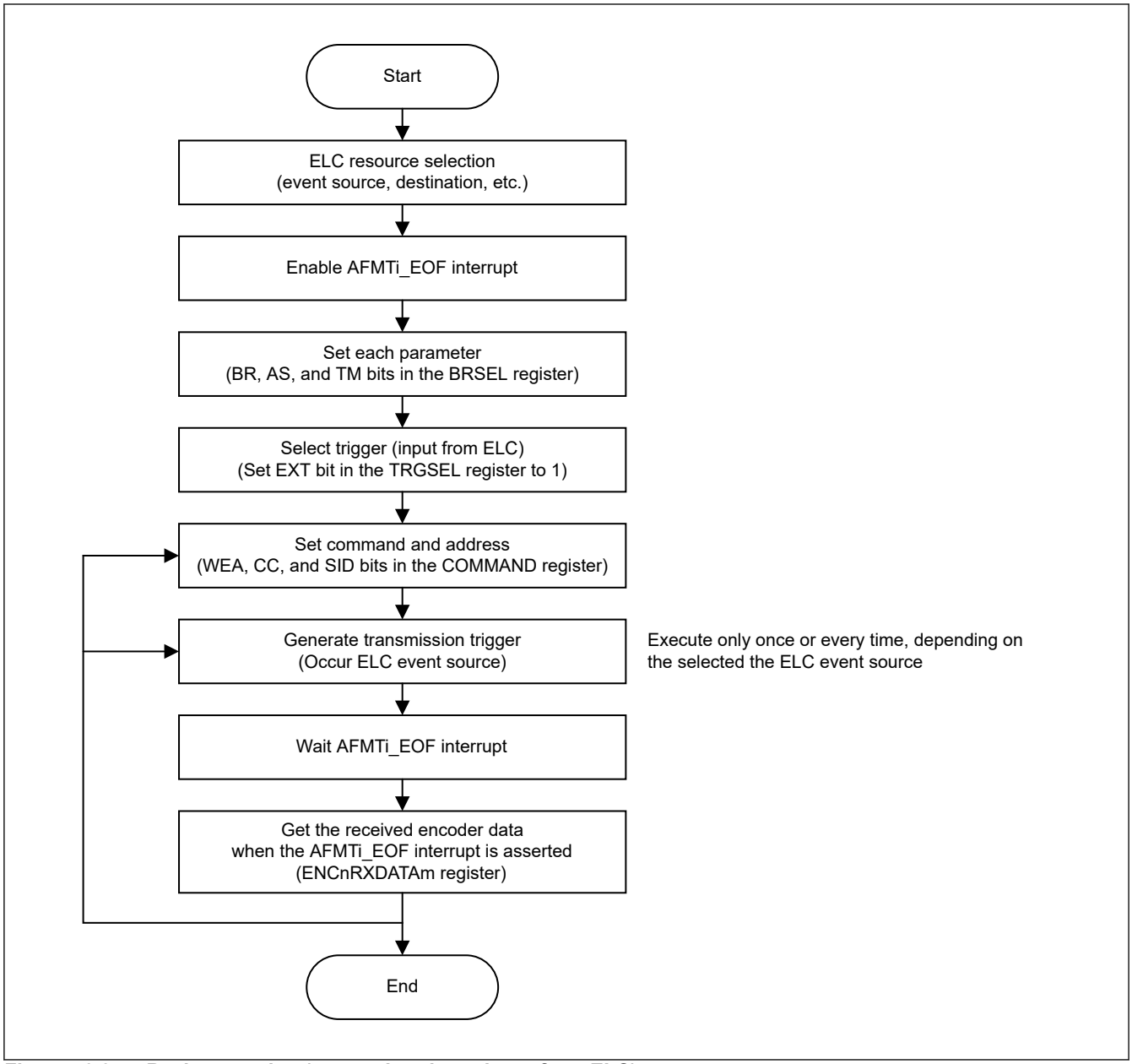


Figure 50.3 Basic operation (startup by trigger input from ELC)

A received encoder data is stored to the ENCnRXDATAm register and the AFMTi_EOF interrupt is asserted when communication is completed. The AFMTi_EOF interrupt is canceled when the first output data is read out.

The baud rate is required a value depending on the encoder used.

50.3.2 Command Configuration

(1) One frame command (CDF0 to CDF12, CDF15 to CDF17, CDF21, CDF22, and CDF27 to CDF31)
Position data request / various clear request / EC address setting / one-rotate data zero preset

Command data

(2) Two frame command (CDF13)
EEPROM data read

Command data	Read address (8 bits)

(2) Four frame command (CDF14, and CDF18 to CDF19)
EEPROM data write

Command data	Write data (lower 8 bits)	Write data (upper 8 bits)	Read address (8 bits)

Figure 50.4 Command configuration

50.3.3 Interrupts

50.3.3.1 Communication Completion Interrupt

When a received encoder data is set to the ENCnRXDATAm (n = 1 to 8, m = 0 to 7) registers and the sequence of receiving and sending operations is completed, the communication completion interrupt, AFMTi_EOF (i = 0, 1), is asserted. When the first output data readout is executed, the communication completion interrupt is canceled.

The TRGSEL register is reset when the communication completion interrupt happens.

50.3.3.2 Time Out Error Interrupt

When data is not returned within the predetermined time for a request, the time out error interrupt, AFMTi_TMOUT (i = 0, 1), is asserted. When next communication start, the time out error interrupt is canceled.

50.3.4 Individual Transmission Mode

Figure 50.5 shows the serial data transmission/reception timing for the individual mode.

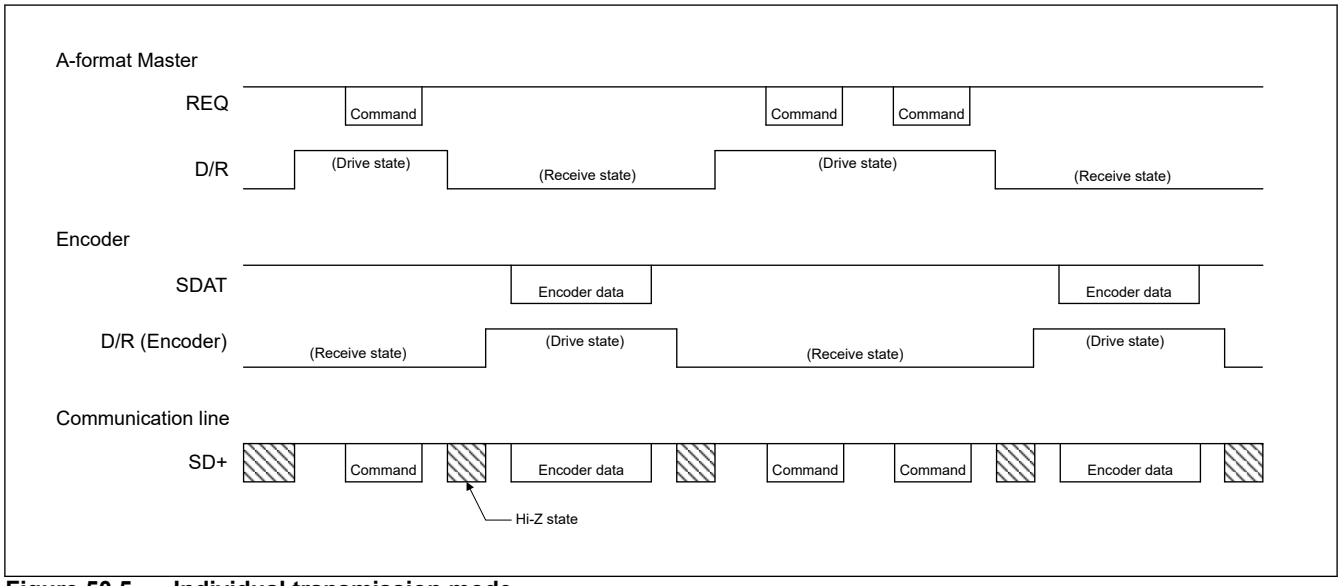


Figure 50.5 Individual transmission mode

50.3.5 Multiple transmission Mode

Figure 50.6 shows the serial data transmission/reception timing for the multiple transmission mode.

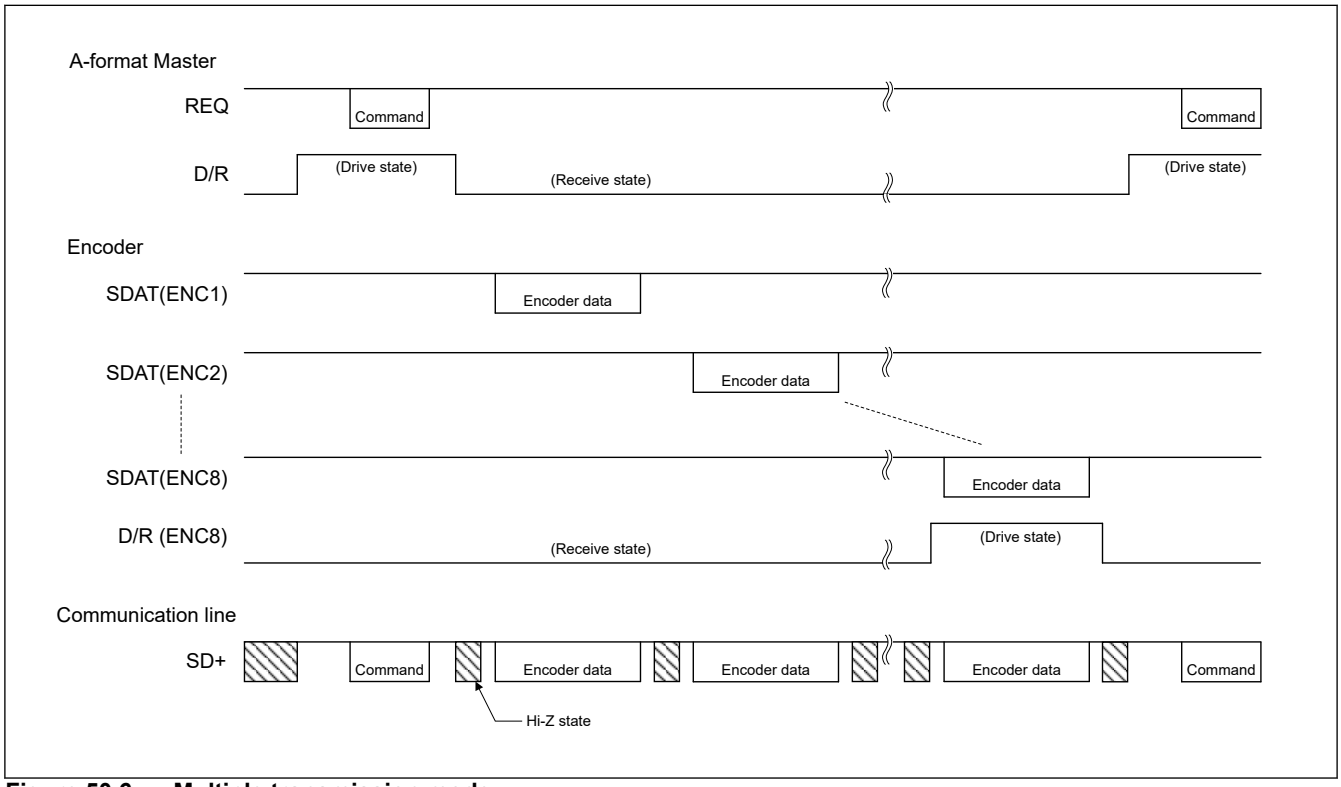


Figure 50.6 Multiple transmission mode

50.4 Usage Note

50.4.1 CDF20 Command Code

This LSI does not support the CDF20 command code. The encoder address should be set by CDF11 command code for each encoder.

51. Encoder Divided-Output (ENCOUT) Module

51.1 Overview

The encoder divided-output (ENCOUT) module of this MPU generates Phase A, Phase B, and Phase Z (index) output signals. The waveforms are similar to the output signals from an incremental encoder, and reflect the positional information set in registers. Especially, by setting the results of reading positional information from an incremental or absolute encoder in the registers of ENCOUT, these signals can be converted to the desired Phase A, Phase B, and Phase Z output signals.

Table 51.1 describes the ENCOUT specifications and Figure 51.1 shows a block diagram of the ENCOUT module.

Table 51.1 ENCOUT specifications

Item	Description
Number of units	1 unit
Functions	• Number of edges per rotation: 16 to 65,536 (a multiple of 4) • Carrier period: 350 ns to 3,276.75 µs • Changing the active sense of Phase B • Changing the phase and width of Phase Z
Event link	Support for the input of carrier period event signals from ELC
Module-stop function	Module-stop state can be set to reduce power consumption

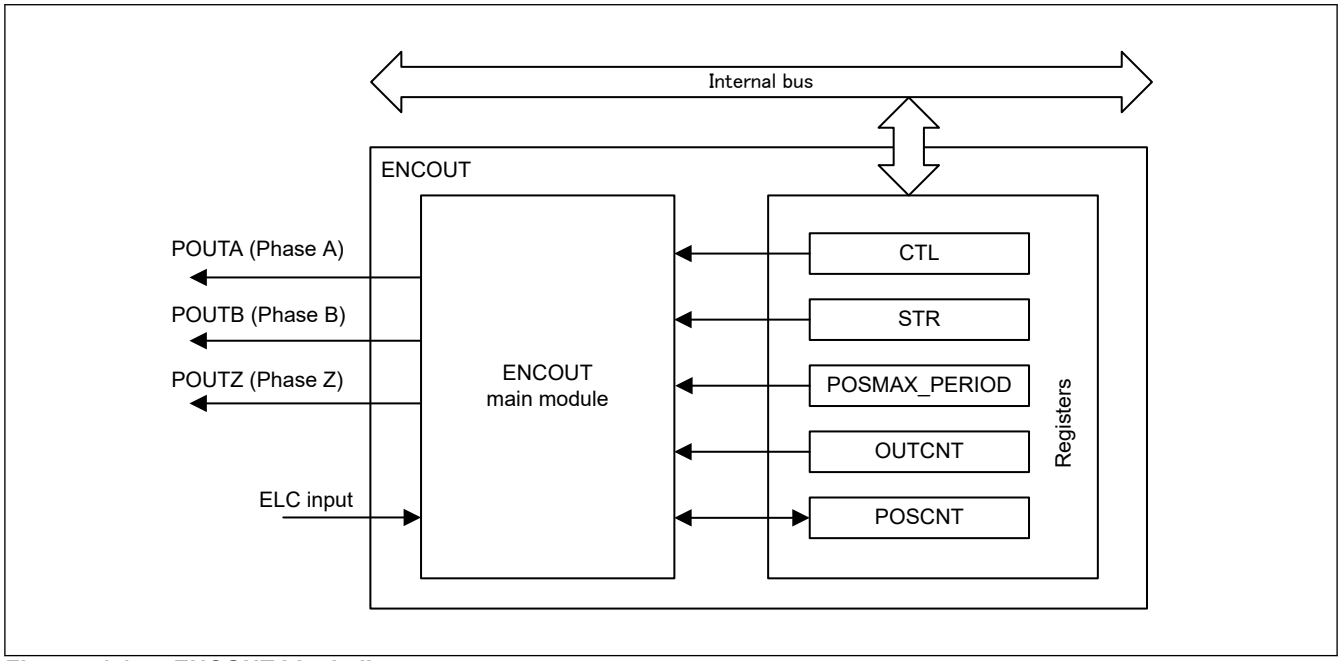
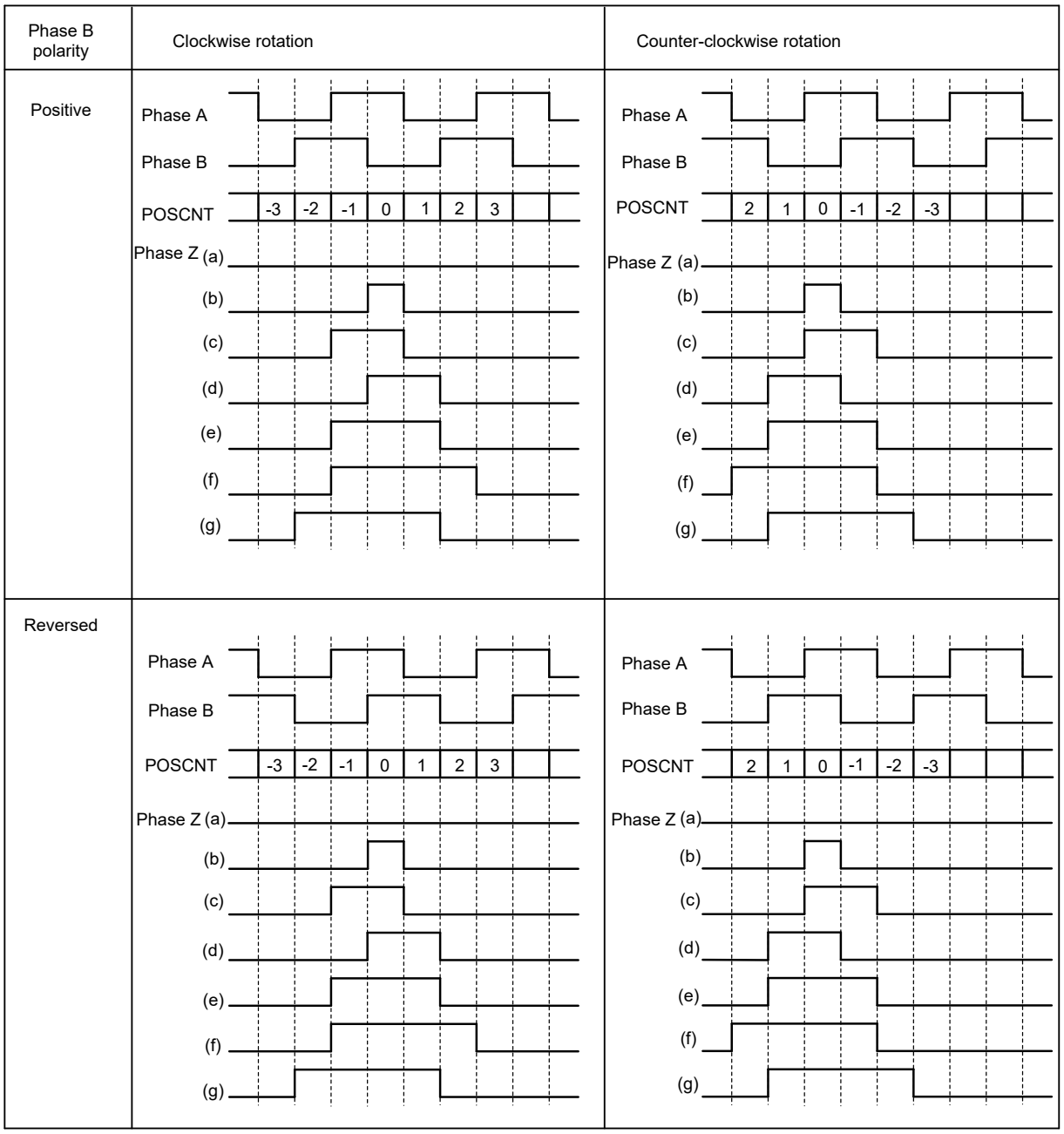


Figure 51.1 ENCOUT block diagram

Table 51.2 lists the input/output pins of the ENCOUT module.

Table 51.2 ENCOUT pin configuration

Pin name	I/O	Function
POUTA	Output	Phase A output pin
POUTB	Output	Phase B output pin
POUTZ	Output	Phase Z output pin



Note: Each output waveform of Phase Z is based on the following settings:
(a) Disable Phase Z output; CTL.ZW = 0
(b) Pulse output when current position, POSCNT, is 0; CTL.ZW = 1
(c) Pulse output when current position, POSCNT, is −1 and 0; CTL.ZW = 2 and CTL.ZS = 0
(d) Pulse output when current position, POSCNT, is 0 and 1; CTL.ZW = 2 and CTL.ZS = 1
(e) Pulse output when current position, POSCNT, is −1, 0, and 1; CTL.ZW = 3
(f) Pulse output when current position, POSCNT, is −1, 0, 1, and 2; CTL.ZW = 4 and CTL.ZS = 0
(g) Pulse output when current position, POSCNT, is −2, −1, 0, and 1; CTL.ZW = 4 and CTL.ZS = 1

Note: Negative values for current position (POSCNT).
POSCNT = −1: POSMAX setting value
POSCNT = −2: POSMAX setting value − 1

Figure 51.2 Phase A, Phase B, and Phase Z output waveforms

51.2 Register Descriptions

51.2.1 CTL : Control Register

Base address: ENCOUT = 0x9006_0000

Offset address: 0x000

Bit position:	31																											4	3	1	0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ZS	ZW[2:0]		POL
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	POL	Polarity of Phase B Set this bit when STR.ENCE = 0. The bit setting is immediately reflected to Phase B output. 0: Positive phase 1: Reversed phase	R/W
3:1	ZW[2:0]	Pulse Width of Phase Z*1 0x0: Disable Phase Z output 0x1: Pulse output when current position, POSCNT, is 0 0x2: Pulse output when current position, POSCNT, is -1 and 0 (sync with Phase A) Pulse output when current position, POSCNT, is 0 and 1 (sync with Phase B) 0x3: Pulse output when current position, POSCNT, is -1, 0 and 1 0x4: Pulse output when current position, POSCNT, is -1, 0, 1 and 2 (sync with Phase A) Pulse output when current position, POSCNT, is -2, -1, 0 and 1 (sync with Phase B) Others: Disable Phase Z output	R/W
4	ZS	Synchronization phase of Phase Z 0: Phase A 1: Phase B	R/W
31:5	—	These bits are read as 0. The write value should be 0.	R/W

Note 1. Negative values for current position (POSCNT).
POSCNT = −1: POSMAX setting value
POSCNT = −2: POSMAX setting value − 1

The CTL register is used to set ENCOUT operation. This register should be set when STR.ENCE = 0. When STR.ENCE = 1, the write operation is disregarded.

51.2.2 STR : Start Register

Base address: ENCOUT = 0x9006_0000

Offset address: 0x004

Bit position:	31																														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ENCE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
0	ENCE	ENCOUT Operation Start When the operation is stopped, the outputs of Phase A and Phase B correspond to POSCNT register, and Phase Z output is 0. 0: ENCOUT operation stop 1: ENCOUT operation start	R/W
31:1	—	These bits are read as 0. The write value should be 0.	R/W

The STR register is used to start/stop ENCOUT operation.

51.2.3 POSMAX_PERIOD : Maximum Position and Carrier Period Register

Base address: ENCOUT = 0x9006_0000

Offset address: 0x00C

Bit position:	31															16	15														0
Bit field:	POSMAX[15:0]																PERIOD[15:0]														
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	PERIOD[15:0]	Carrier Period Setting Carrier period length should be set in units of 50 ns according to the event input period from ELC. Settable value is from 7 (0x0007) to 65535 (0xFFFF). If event period from ELC cannot be divided evenly by 50 ns, the value should be set to ("event period from ELC" / 50) + 1 (round up the decimal point).	R/W
31:16	POSMAX[15:0]	Maximum Position Setting Maximum position value (number of edges) for one revolution. The maximum position value is set a multiple of 4. If maximum position value is 4 × N, this value should be set to 4 × N – 1. Settable value is 15 (0x000F), 19 (0x0013), 23 (0x0017), 27 (0x001B), 31 (0x001F), …, 65531 (0xFFFB), 65535 (0xFFFF).	R/W

The POSMAX_PERIOD register is used to set the maximum position value and carrier period. This register should be set when STR.ENCE = 0. When STR.ENCE = 1, the write operation is ignored.

51.2.4 OUTCNT : Output Count Register

Base address: ENCOUT = 0x9006_0000

Offset address: 0x010

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	EDGCNT[15:0]														
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	EDGCNT[15:0]	Edge Count Setting Set the number of edges for Phase A and Phase B in the next carrier period. Set in two's complement representation, negative values indicate counter-clockwise rotation. Settable value is from -32768 (0x8000) to 32767 (0x7FFF). This value should be set to less than or equal to the setting value in PERIOD[15:0]. The update interval of edge count can be calculated from (PERIOD[15:0] / EDGCNT[15:0]) × 50 (ns). If PERIOD[15:0] / EDGCNT[15:0] cannot be divided evenly, the EDGCNT[15:0] is updated to disperse the error over the carrier period. See section 51.3.6. Uniformization of Waveform Output Accumulation Error.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The OUTCNT register is used to set the number of edges for Phase A and Phase B. When STR.ENCE = 1, the value should be set before an event occurs for the next carrier period.

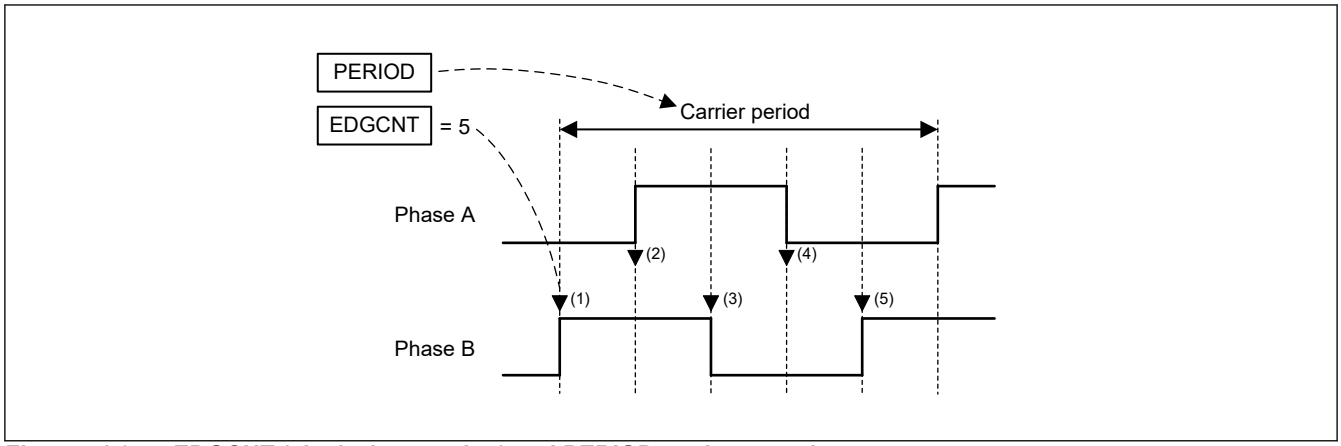


Figure 51.3 EDGCNT (clockwise rotation) and PERIOD setting sample

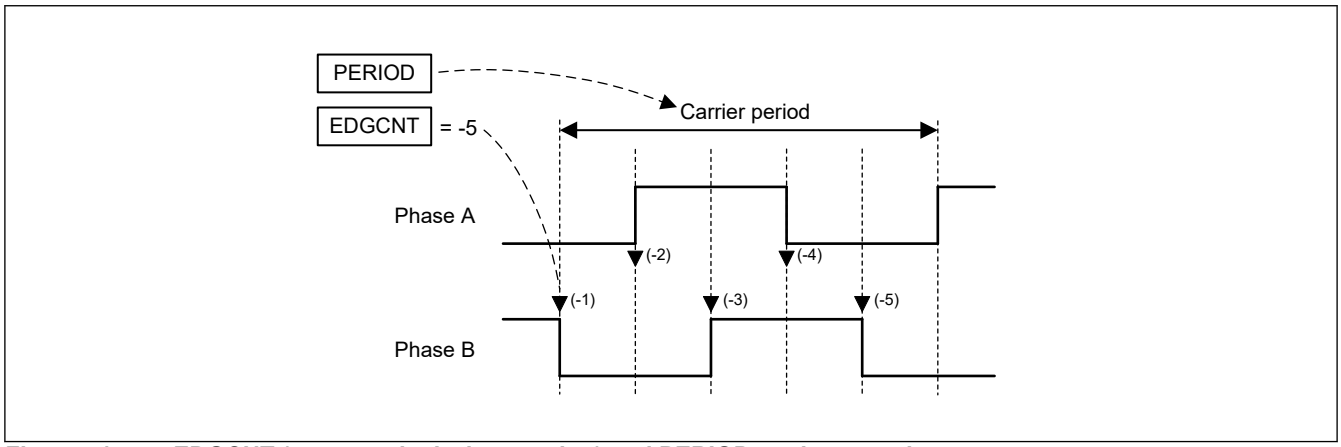


Figure 51.4 EDGCNT (counter-clockwise rotation) and PERIOD setting sample

51.2.5 POSCNT : Position Count Register

Base address: ENCOUT = 0x9006_0000

Offset address: 0x014

Bit position:	31																15														0
Bit field:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	POSCNT[15:0]															
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Function	R/W
15:0	POSCNT[15:0]	Current Position Set a current position value. Settable value is from 0 to value in POSMAX[15:0]. After starting ENCOUT operation, the value is updated according to the position.	R/W
31:16	—	These bits are read as 0. The write value should be 0.	R/W

The POSCNT register is used to set a current position value and check the current position. This register should be written when STR.ENCE = 0. When STR.ENCE = 1, the value is updated according to the position associated with Phase A and Phase B output.

51.3 Operation

ENCOUT generates Phase A/B and Phase Z outputs based on the register settings. The carrier period of Phase A/B and Phase Z outputs are generated by input from ELC.

Phase A/B and Phase Z output formats are shown in Figure 51.2. And Phase Z output is generated by matching POSCNT register value according to Table 51.4.

51.3.1 Phase A and Phase B Output Format

Phase A and Phase B output are represented by the four patterns shown in the table below according to the value of POSCNT[1:0]. The patterns are shifted right with the value of POSCNT[1:0] as Phase A pattern = 1001b (0x9) and Phase B pattern = 1100b (0xC), the first bit will be the output value of Phase A and Phase B. When reverse phase output (CTL.POL = 1), Phase B output pattern is reversed.

Table 51.3 Phase A and Phase B output pattern

POSCNT[1:0] value	3	2	1	0
Phase A output	1	0	0	1
Phase B output (CTL.POL = 0: Positive)	1	1	0	0
Phase B output (CTL.POL = 1: Reversed)	0	0	1	1

51.3.2 Phase Z Output Format

Phase Z output is generated by matching POSCNT register value to the table values below.

Table 51.4 Phase Z output pattern

Phase Z output width (CTL.ZW[2:0])	POSCNT[15:0] value when Phase Z output is 1
0	(Disable Phase Z output)
1	0
2 (Synchronize with Phase A)	-1, 0
2 (Synchronize with Phase B)	0, 1
3	-1, 0, 1
4 (Synchronize with Phase A)	-1, 0, 1, 2
4 (Synchronize with Phase B)	-2, -1, 0, 1

Note: Negative values for current position (POSCNT)
POSCNT = −1: POSMAX setting value
POSCNT = −2: POSMAX setting value − 1

51.3.3 Carrier Period Generation by Events from ELC

Event inputs from ELC are recognized as a carrier period. Generate a carrier period with a timer, etc., and input it using ELC.

51.3.4 PERIOD Setting Value 1

If the ELC event period is not divisible by the number of operating clocks in POSMAX_PERIOD.PERIOD configuration, set the integer value + 1 to POSMAX_PERIOD.PERIOD.

The figure below is an operation example. The operation clock period is 50 ns (20 MHz), the carrier period is 1000 ns (PERIOD = 20), ELC event period is 980 ns. ELC event is checked on every clock, and if the cumulative error between the carrier period and ELC event period exceeds 50 ns, an edge update is performed by 50 ns shorter than the 1000 ns carrier period and the carrier period counter is reset. Therefore, the cumulative error is less than 50 ns (one clock) as shown in the following table.

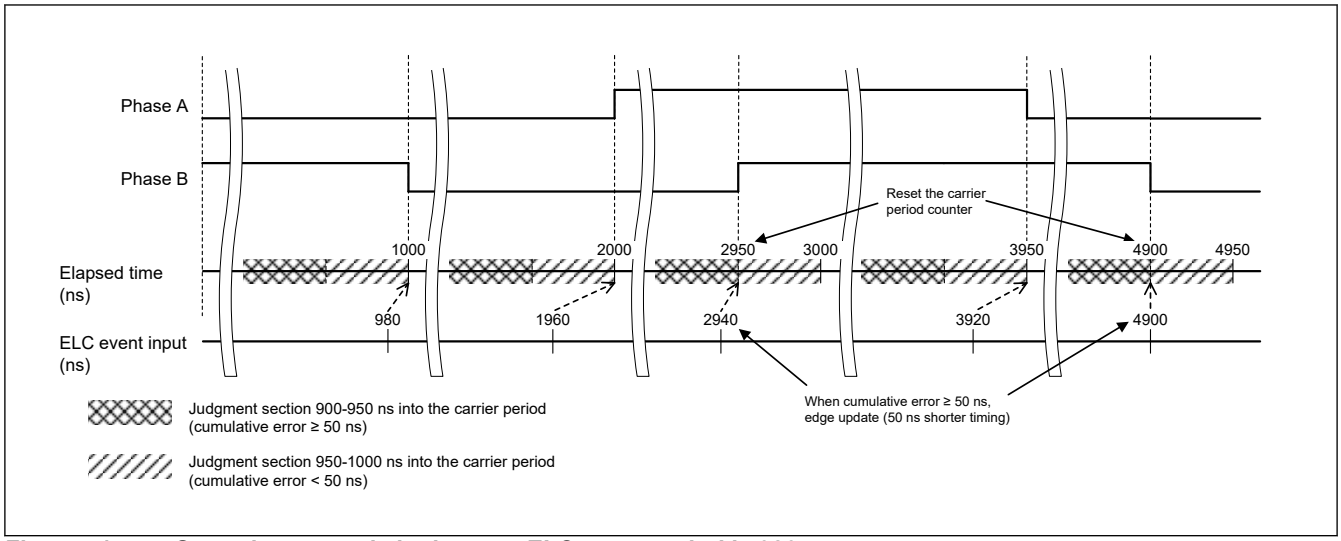


Figure 51.5 Operation example in the case ELC event period is 980 ns

Table 51.5 Actual carrier period in the case ELC event period is 980 ns

ELC event period	Actual carrier period	Remarks
980	1000	—
1960	2000	—
2940	2950	If the cumulative error is ≥ 50 ns, the edge is updated by 50 ns earlier.
3920	3950	—
4900	4900	If the cumulative error is ≥ 50 ns, the edge is updated by 50 ns earlier.
5880	5900	—
6860	6900	—
7840	7850	If the cumulative error is ≥ 50 ns, the edge is updated by 50 ns earlier.
8820	8850	—
9800	9800	If the cumulative error is ≥ 50 ns, the edge is updated by 50 ns earlier.

51.3.5 PERIOD Setting Value 2

The minimum setting for POSMAX_PERIOD.PERIOD is 7. The carrier period takes the edge of the ELC signal asynchronously at the operation clock (20 MHz). The next ELC input signal cannot be accepted from the edge detection to the completion of operation clock synchronization. Therefore, the minimum setting value of PERIOD is 7 or more, so that the ELC signal can be reliably detected. Operation cannot be guaranteed when the setting value is set to 6 or less.

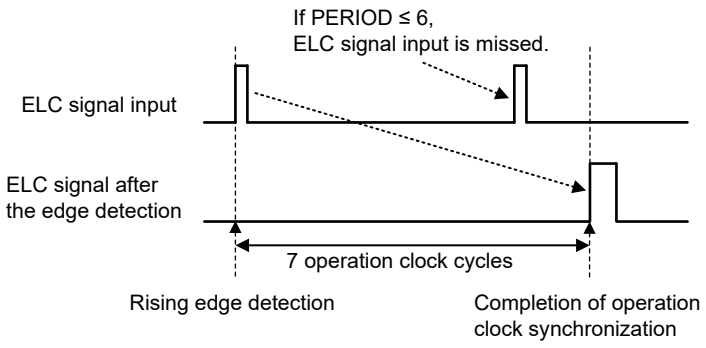


Figure 51.6 Example of the operation when PERIOD value is 6 or less

51.3.6 Uniformization of Waveform Output Accumulation Error

The edge-to-edge operation cycle truncates the decimal point of the calculated value, resulting in an error at the end of the carrier signal.

Counter for waveform update = carrier period / operation clock period / number of edges (OUTCNT.EDGCNT)

Example of the error accumulation with waveform output

The following figure is an operation example. The carrier period = 5000 ns, the operation clock period = 20 MHz (50 ns), the number of edges = 7. In this case, the error occurs finally because the decimal point is truncated by the edge interval calculation as follows.

Edge interval = 5000 / 50 / 7 = 14.2857 (the decimal point is truncated, and it results in 14)

Error = 5000 / 50 – 14 × 7 = 2

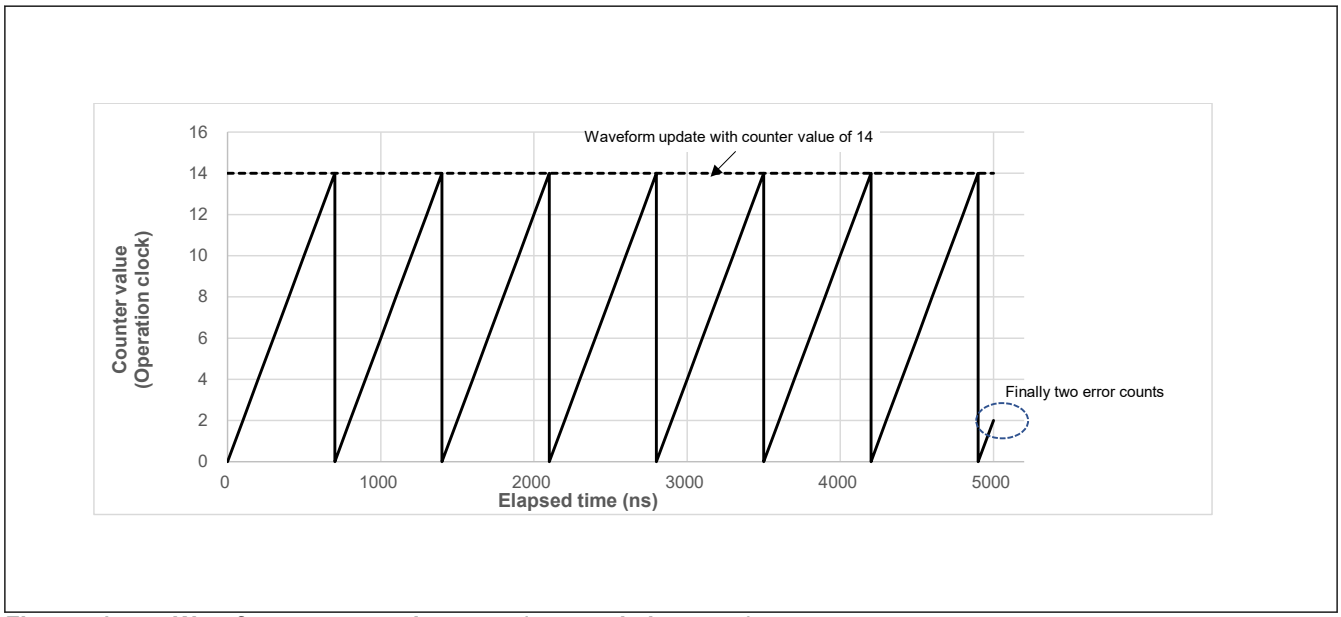


Figure 51.7 Waveform output and counter (accumulation error)

Therefore, the settings of the waveform output are changed as follows, and the error of the output waveform is uniformed throughout the carrier period.

- Add the "number of edges" at each operating clock
- The total counts of the carrier period = the number of edges × the number of operating clocks during the carrier period
- Update the waveform when the counter value of the operating clock exceeds the "number of clocks during the carrier period"

Example of the waveform output with the error accumulation measure

The following figure is an operation example. The carrier period = 5000 ns, the operation clock period = 20 MHz (50 ns), the number of edges = 7. The number of operating clocks during the carrier period is 100, counting by 7 per clock cycle. When the counter exceeds 100, it is subtracted by 100 and the waveform is updated. The counter increases by 700 in one carrier period, and the waveform changes exactly seven times in one carrier period.

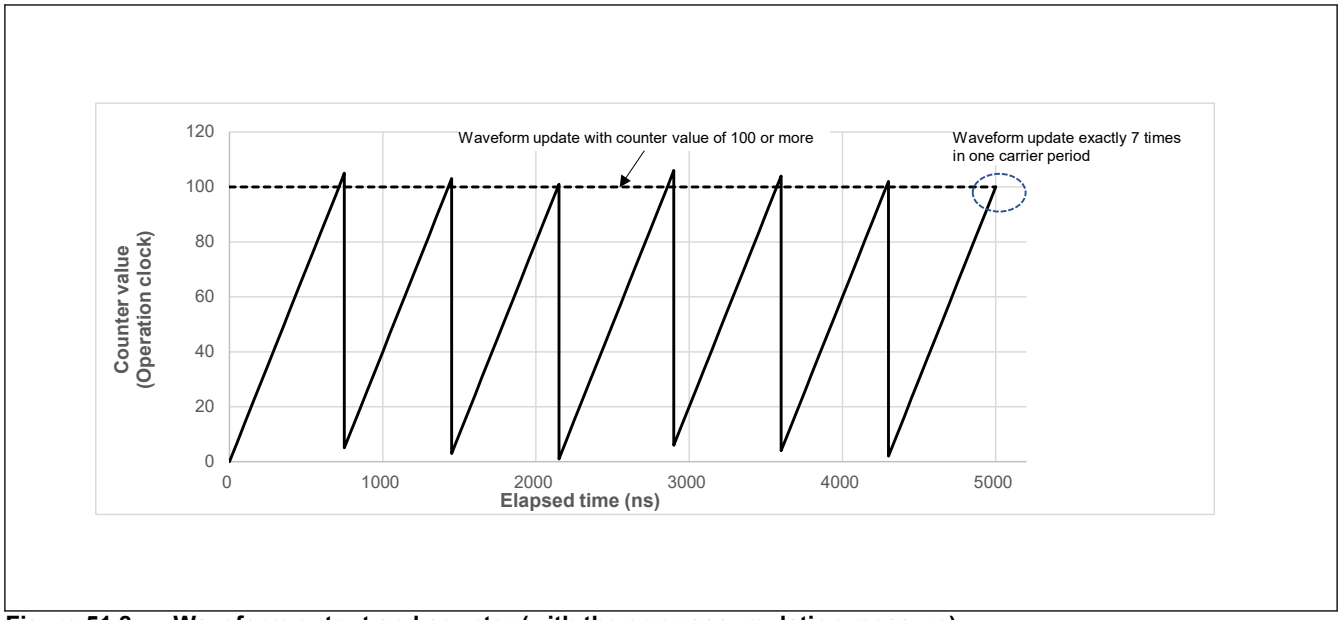


Figure 51.8 Waveform output and counter (with the error accumulation measure)

In the case of this setting, once the "number of clocks during the carrier period" is initially set, the waveform will be output by simply updating the "number of edges" setting during operation. In addition, since the assumption of the carrier period is 50000 to 3276000 (ns), the "number of clocks for carrier period" can be handled within the range of 2 bytes data.

(Carrier period: 3276000 ns) / (Operation clock period: 50 ns) = 65520 (0xFFF0)

Maximum carrier period setting: (POSMAX_PERIOD.PERIOD setting: 65535 (0xFFFF)) × 50 = 3276750 [ns]

51.3.7 Setting Procedure

51.3.7.1 System Configuration

The following figure shows a system configuration example that uses ENCOUT to convert the position information of an incremental encoder or an absolute encoder into Phase A/B and Phase Z outputs. The CPU obtains position information from the encoder for each period of the current loop and position control loop notified by the timer module. Then by updating ENCOUT setting periodically due to the difference from the previous position information, Phase A/B and Phase Z output is generated. The timer module events that generate periods of the current loop or position control loop should be input to ENCOUT through the ELC.

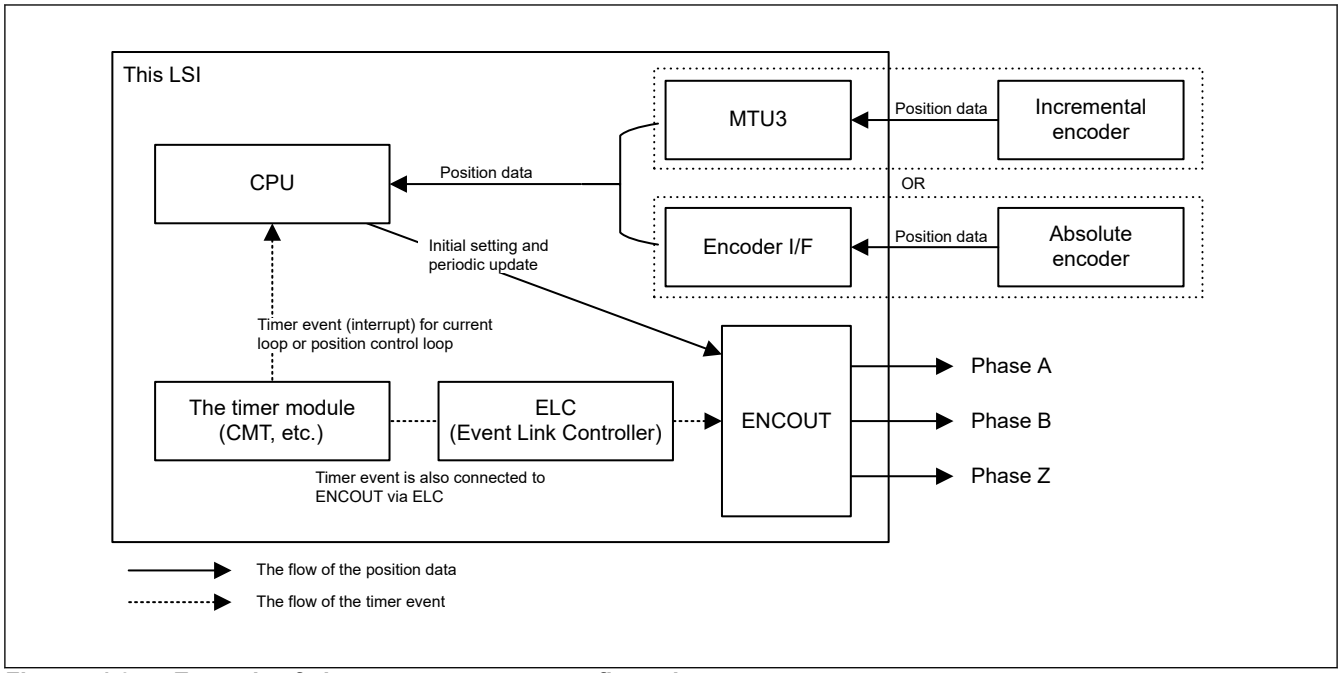


Figure 51.9 Example of phase output system configuration

51.3.7.2 Initialization

The following figure shows the initialization flow of ENCOUT.

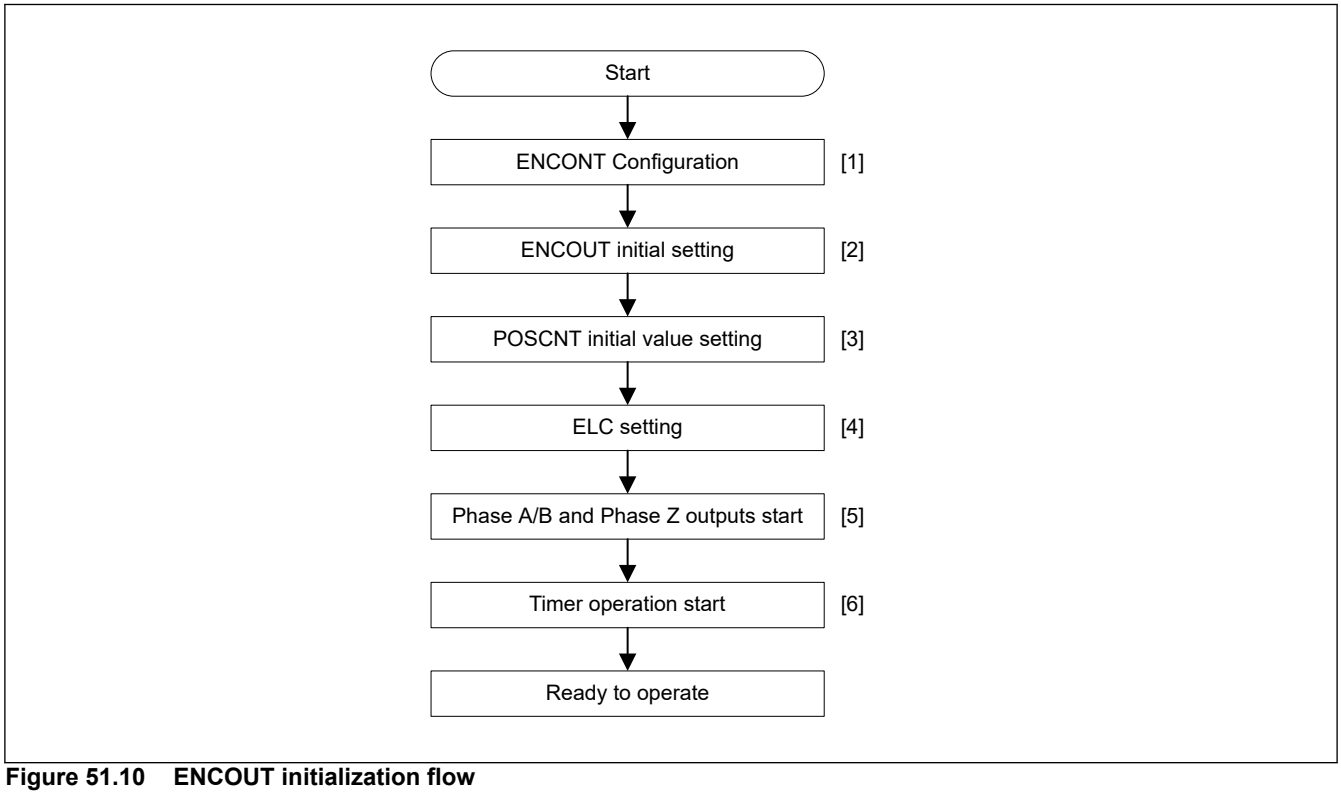


Figure 51.10 ENCOUT initialization flow

[1] ENCOUT configuration

Set the output pin assignment using the I/O port registers. Release from the module-stop state by Module Stop Control Register and release from reset by Module Reset Control Register.

[2] ENCOUT initial setting

Initialize CTL, POSMAX_PERIOD, and OUTCNT registers. Set OUTCNT register to 0. Set the value of POSMAX_PERIOD.PERIOD to match the event input period from the ELC. If the event period from ELC is not divisible by operation clock period (50 ns), the value should be set to ["event period from ELC" / 50 + 1]. (Round up the decimal point)

Example 1: The number of edges per revolution is 100, Phase B output is positive, and the carrier period is 500 (ns).
POSMAX_PERIOD.POSMAX = 99 (4 × 25 − 1)
POSMAX_PERIOD.PERIOD = 10 (500 / 50)
CTL = 0, OUTCNT = 0

Example 2: The number of edges per revolution is 100, Phase B output is positive, and the carrier period is 520 (ns).
POSMAX_PERIOD.POSMAX = 99 (4 × 25 − 1)
POSMAX_PERIOD.PERIOD = 11 (520 / 50 + 1)
CTL = 0, OUTCNT = 0

[3] POSCNT initial value setting

Set the POSCNT register to the initial position in the range from 0 to POSMAX_PERIOD.POSMAX of an absolute encoder or an incremental encoder.

The position values = [Encoder position value] × [Number of edges per rotation] / [Encoder position value resolution] (truncate the decimal point).

Example: The number of edges per revolution is 100, encoder position value resolution is 20 bits (1048576), and initial position value is 943719.
POSCNT = 90 (943719 × 100 / 1048576)

[4] ELC setting

The event from the timer is input to ENCOUT by the setting of ELC registers.

[5] Phase A/B and Phase Z outputs start

Set the STR.ENCE bit to 1 to start Phase A/B and Phase Z output.

[6] Timer operation start

Operate a timer to generate a carrier period event.

51.3.7.3 Main Processing

The following figure shows the main processing flow of ENCOUT. This flow should be performed for each carrier period.

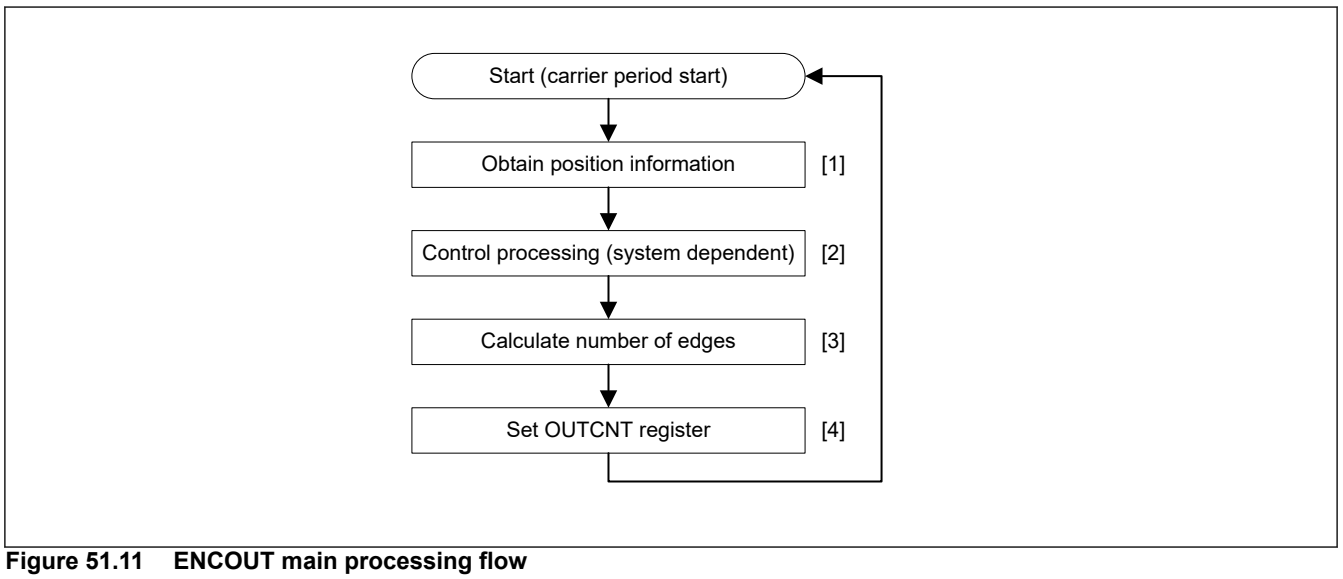


Figure 51.11 ENCOUT main processing flow

[1] Obtain position information

Obtain position information from the absolute or incremental encoders.

[2] Control processing (system dependent)

Performs motor control processing, etc. based on position information. This process is system dependent.

[3] Calculate number of edges

Calculates the setting value for OUTCNT.EDGCNT.

Current position value with the maximum position as POSMAX = [Position value obtained in step [1]] × [Number of edges per rotation] / [Encoder position value resolution] (truncate the decimal point)

Setting value of OUTCNT.EDGCNT = current position − previous position (the maximum position as POSMAX)

However, if the absolute value of above is greater than half of the number of edges per revolution:

- When setting value > 0, setting value − [Number of edges per rotation]
- When setting value < 0, setting value + [Number of edges per rotation]

Example: In the following cases:
Encoder position value = 20972, encoder position value resolution = 20 bits (1048576), number of edges per rotation = 100, previous position (the maximum position as POSMAX) = 95, carrier period = 100 µs
Current position (the maximum position as POSMAX) = 20972 × 100 / 1048576 = 2
Setting value of OUTCNT.EDGCNT = 2 − 95 = −93
This absolute value is greater than half of the number of edges per revolution (93 > 50).
Setting value is (−93) < 0, setting value + [Number of edges per rotation] = −93 + 100 = 7.

[4] Set OUTCNT register

Set OUTCNT.EDGCNT to the value calculated in step [3].

51.3.8 Setting and Output Examples

The following figure shows an example of Phase A/B and Phase Z output waveforms that correspond to the ENCOUT settings.

This is an example when the carrier period is 100 µs, the number of edges per revolution is 100, Phase B is the positive phase, and initial position value is 90 (the maximum position as POSMAX). This is an example of a case the position transitions to 95, 2, and 98.

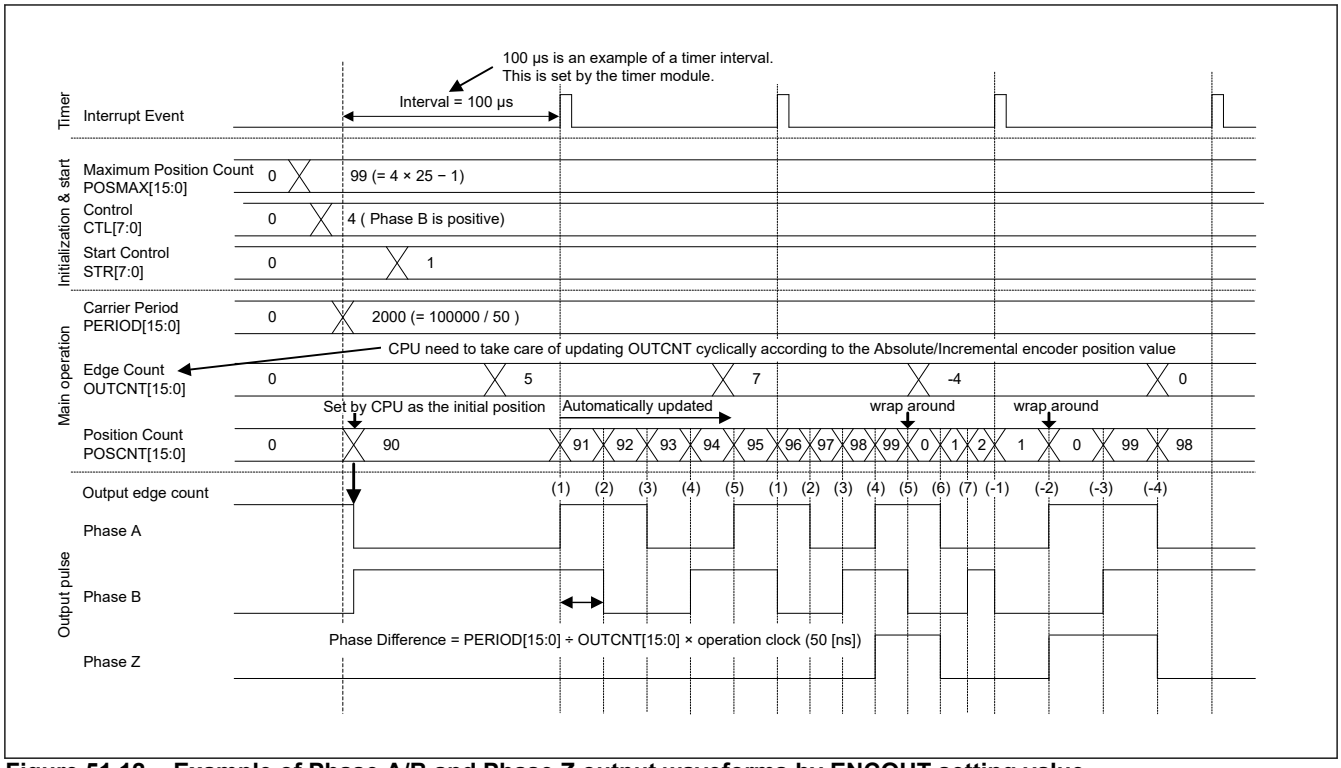


Figure 51.12 Example of Phase A/B and Phase Z output waveforms by ENCOUT setting value

52. Electrical Characteristics

Electrical characteristics of this LSI is defined with the following conditions unless otherwise described.

Conditions:

VDD = 1.05 to 1.15 V

VCC18 = VCC1833_n (1.8-V mode) = VCC18_PLL0 = VCC18_PLL1 = VCC18_USB = AVCC18_USB = VCC18_ADC0 = VCC18_ADC1 = AVCC18_TSU = VREFH0 = VREFH1= 1.70 to 1.95 V

VCC33 = VCC1833_n (3.3-V mode) = VCC33_USB = 3.135 to 3.465 V

VSS = VSS_USB = 0 V

Tj = −40 to 125ºC

52.1 Absolute Maximum Ratings

Table 52.1 Absolute maximum ratings

Parameter	Symbol	Value	Unit
Power supply voltage (3.3-V mode)	VCC33, VCC1833_2, VCC1833_3	−0.3 to +3.8	V
Power supply voltage (1.8-V mode)	(VCC18) VCC1833_2, VCC1833_3	−0.3 to +2.5	V
Power supply voltage	VDD	−0.3 to +1.5	V
Input voltage	Vin (3.3-V logic)	−0.3 to VCC33 + 0.3	V
Input voltage	Vin (1.8-V logic)	−0.3 to VCC18 + 0.3	V
Analog power supply voltage	VCC18_PLL0, VCC18_PLL1, VCC18_USB, AVCC18_USB, VCC18_ADC0, VCC18_ADC1, AVCC18_TSU*1	−0.3 to smaller value of VCC18*2 + 0.3 or 2.5	V
	VCC33_USB	−0.3 to smaller value of VCC33 + 0.3 or 3.8	V
Voltage difference between power supply pins	VCC33-VCC18	-2.5 to + 2.1	V
Analog input voltage	VAN	−0.3 to smaller value of VCC18_ADC0/1 + 0.3 or 2.5	V
Reference voltage	VREFH0, VREFH1	−0.3 to smaller value of VCC18_ADC0/1 + 0.3 or 2.5	V
Crystal oscillator pins input voltage	XTAL, EXTAL	−0.3 to +1.5	V
Operating temperature (Junction temperature)	T_j	−40 to +125	ºC
Storage temperature	T_{stg}	−55 to +125	ºC

Note 1. Connect Analog power supply pins to VCC18 when Analog block(s) are not used. Do not leave these pins open.
Note 2. For convention, "VCC18" virtually represents any 1.8-V power supplies of the chip such as VCC1833_n in 1.8-V mode.

Caution: Permanent damage to the LSI might result if absolute maximum ratings are exceeded.

52.2 Power Supply

Table 52.2 Power supply

Parameter	Symbol	Value	Min.	Typ.	Max.	Unit
Power supply voltages	VCC33		3.135	—	3.465	V
	VDD		1.05	1.1	1.15	V
	VSS		—	0	—	V
Power supply voltages supporting multi voltage mode	VCC1833_2, VCC1833_3	3.3-V mode	3.135	3.3	3.465	V
		1.8-V mode (VCC18)	1.70	1.8	1.95	V
Analog power supply voltages	VCC18_PLL0		—	VCC18	—	V
	VCC18_PLL1		—	VCC18	—	V
	VCC33_USB		—	VCC33	—	V
	VCC18_USB		—	VCC18	—	V
	AVCC18_USB		—	VCC18	—	V
	VCC18_ADC0		—	VCC18	—	V
	VCC18_ADC1		—	VCC18	—	V
	AVCC18_TSU		—	VCC18	—	V
	VSS_USB		—	0	—	V

52.3 Power On/Off Sequence

Power on/off sequence and timing are shown in the figure and table below.

For power-up, 1.1-V and 1.8-V power (i.e. VDD, VCC18, and AVCC) must be supplied first, then 3.3-V power (i.e. VCC33) must be supplied. The power-up sequence must be completed within 100 ms. Reset signal (i.e. RES#) must be held to Low level during the power-up.

For Power-down, 3.3-V power (i.e. VCC33) must go down first and then 1.1-V and 1.8-V power (i.e. VDD, VCC18, and AVCC). The power-down sequence must be completed within 100 ms.

Rise and fall time of each power supply for the power-up and the power-down must be larger than 10 µs.

Power supply voltages and reset signal must be applied with monotonic increase.

Do not apply a negative voltage to power supply voltages.

Stable clock must be supplied to EXTAL/XTAL or EXTCLKIN pin when reset signal (i.e. RES#) is driven high.

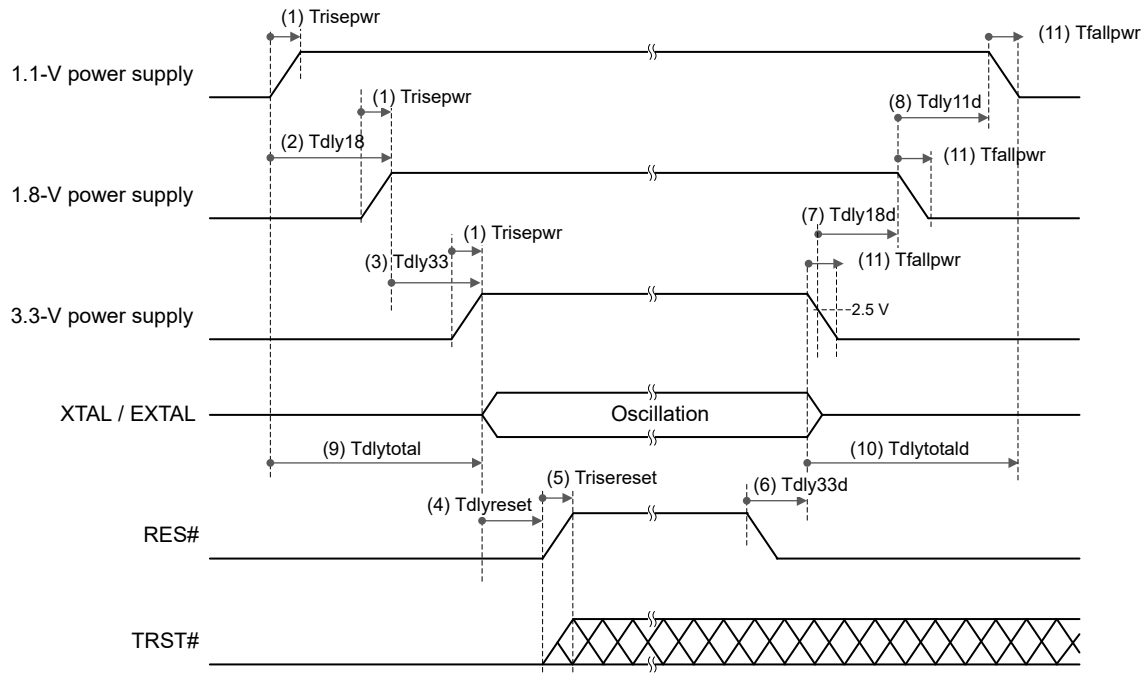


Figure 52.1 Power on/off sequence

Table 52.3 Power on/off sequence timing

No.	Symbol	Description	Value		
			Min.	Typ.	Max.
(1)	Trisepwr	Rising time of the power supply voltage	10 µs	—	30 ms
(2)	Tdly18	Delay time from start of rising of the 1.1-V power supply voltage to completion of rising of the 1.8-V power supply voltage	0	—	100 ms
(3)	Tdly33	Delay time from completion of rising of the 1.8-V power supply voltage to completion of rising of the 3.3-V power supply voltage	0	—	100 ms
(4)	Tdlyreset	Delay time from completion of rising of the 3.3-V power supply voltage to start of rising of RES# when XTAL/EXTAL is used.	10 ms	—	—
		Delay time from completion of rising of the 3.3-V power supply voltage to start of rising of RES# when EXTCLKIN is used.	1 ms	—	—
(5)	Trisereset	Rising time of RES#	—	—	150 µs
(6)	Tdly33d	Delay time from start of falling of RES# to start of falling of the 3.3-V power supply voltage	10 µs	—	—
(7)	Tdly18d	Delay time from start of falling time of the 3.3-V power supply voltage to start of falling of the 1.8-V power supply voltage	0	—	100 ms
(8)	Tdly11d	Delay time from start of falling of the 1.8-V power supply voltage to start of falling of the 1.1-V power supply voltage	0	—	100 ms
(9)	Tdlytotal	Startup time of all power supply voltage	—	—	100 ms
(10)	Tdlytotald	Shut down time of all power supply voltage	—	—	100 ms
(11)	Tfallpwr	Falling time of the power supply voltage	10 µs	—	30 ms

52.4 DC Characteristics

Table 52.4 DC Characteristics (3.3-V mode)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input High-level voltage	V_{IH33}	3.3-V mode, Schmitt Trigger Control Disabled, Except P07_4 pin	2.0	—	VCC33 + 0.3	V
Input Low-level voltage	V_{IL33}		−0.3	—	0.8	V
Positive trigger voltage	V_{T33+}	3.3-V mode, Schmitt Trigger Control Enabled, Except P07_4 pin	0.9	—	2.1	V
Negative trigger voltage	V_{T33-}		0.7	—	1.9	V
Hysteresis voltage	ΔV_{T33}		0.2	—	—	V
Input High-level voltage 2	V_{IH33_2}	3.3-V mode, Schmitt Trigger Control Disabled, P07_4 pin ONLY	VCC33 × 0.7	—	VCC33 + 0.3	V
Input Low-level voltage 2	V_{IL33_2}		-0.3	—	VCC33 × 0.3	V
Positive trigger voltage 2	V_{T33+_2}	3.3-V mode, Schmitt Trigger Control Enabled, P07_4 pin ONLY	—	—	VCC33 × 0.72	V
Negative trigger voltage 2	V_{T33-_2}		VCC33 × 0.3	—	—	V
Hysteresis voltage 2	ΔV_{T33_2}		VCC33 × 0.1	—	—	V
Output High-level voltage	V_{OH33}	Low, IOH = −2 mA	VCC33 − 0.4	—	—	V
	V_{OH33}	Middle, IOH = −4 mA	VCC33 − 0.4	—	—	V
	V_{OH33}	High, IOH = −8 mA	VCC33 − 0.4	—	—	V
	V_{OH33}	Ultra High, IOH = −12 mA	VCC33 − 0.4	—	—	V
Output Low-level voltage	V_{OL33}	Low, IOL = 2 mA	—	—	0.4	V
	V_{OL33}	Middle, IOL = 4 mA	—	—	0.4	V
	V_{OL33}	High, IOL = 8 mA	—	—	0.4	V
	V_{OL33}	Ultra High, IOL = 12 mA	—	—	0.4	V
Input leakage current	|Iin|	Vin = 0 V, Vin = VCC33	—	—	10	µA
Three-State leakage current (off state)	$\|I_{TSI}\|$	Vin = 0 V, Vin = VCC33	—	—	10	µA
Input Pull-up resistors resistance	Rpu	Vin = 0 V	15	—	300	kΩ
Input Pull-up resistors current	Ipu	Vin = 0 V	-220	—	-11	µA
Input Pull-down resistors resistance	Rpd	Vin = VCC33	15	—	300	kΩ
Input Pull-down resistors current	Ipd	Vin = VCC33	11	—	220	µA
Input Capacitance	Cin	All input/output and input pins	—	—	10	pF

Table 52.5 DC Characteristics (1.8-V mode) (1 of 2)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input High-level voltage	V_{IH18}	1.8-V mode, Schmitt Trigger Control Disabled	VCC18 × 0.65	—	VCC18 + 0.3	V
Input Low-level voltage	V_{IL18}		−0.3	—	VCC18 × 0.35	V
Positive trigger voltage	V_{T18+}	1.8-V mode, Schmitt Trigger Control Enabled	VCC18 × 0.4	—	VCC18 × 0.7	V
Negative trigger voltage	V_{T18-}		VCC18 × 0.3	—	VCC18 × 0.6	V
Hysteresis voltage	ΔV_{T18}		VCC18 × 0.1	—	—	V
Output High-level voltage	V_{OH18}	Low, IOH = −2 mA	VCC18 − 0.45	—	—	V
	V_{OH18}	Middle, IOH = −4 mA	VCC18 − 0.45	—	—	V
	V_{OH18}	High, IOH = −8 mA	VCC18 − 0.45	—	—	V
	V_{OH18}	Ultra High, IOH = −12 mA	VCC18 − 0.45	—	—	V

Table 52.5 DC Characteristics (1.8-V mode) (2 of 2)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Output Low-level voltage	V_{OL18}	Low, IOL = 2 mA	—	—	0.45	V
	V_{OL18}	Middle, IOL = 4 mA	—	—	0.45	V
	V_{OL18}	High, IOL = 8 mA	—	—	0.45	V
	V_{OL18}	Ultra High, IOL = 12 mA	—	—	0.45	V
Input leakage current	|Iin|	Vin = 0 V, Vin = VCC18	—	—	10	µA
Three-State leakage current (off state)	$\|I_{TSI}\|$	Vin = 0 V, Vin = VCC18	—	—	10	µA
Input Pull-up resistors resistance	Rpu	Vin = 0 V	15	—	300	kΩ
Input Pull-up resistors current	Ipu	Vin = 0 V	-120	—	-6	µA
Input Pull-down resistors resistance	Rpd	Vin = VCC18	15	—	300	kΩ
Input Pull-down resistors current	Ipd	Vin = VCC18	6	—	120	µA
Input Capacitance	Cin	All input/output and input pins	—	—	10	pF

Table 52.6 USB2.0 USB_RREF Pin

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Reference resistor*1	R_{REF}	—	1.8 (±1%)			kΩ

Note 1. The reference resistor connected to the USB_RREF pin is for external connection to this LSI.

Table 52.7 USB2.0 Pull-Up/Pull-Down Resistors

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
USB_DP pull-up resistor (when the function controller is selected)	R_{PU}	Idle	0.900	—	1.575	kΩ
		Transmission/reception	1.425	—	3.090	kΩ
USB_DP/USB_DM pull-down resistors (when the host controller is selected)	R_{PD}	—	14.25	—	24.80	kΩ

Table 52.8 USB2.0 Host/Function-Related Pins (Low/Full Speed)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input high level voltage	V_{FSIH}	—	2.0	—	—	V
Input low level voltage	V_{FSIL}	—	—	—	0.8	V
Differential input sensitivity	V_{FSDI}	|(USB_DP) − (USB_DM)|	0.2	—	—	V
Differential common mode range	V_{FSCM}	—	0.8	—	2.5	V
Output low level voltage	V_{FSOL}	IFSOL = 2 mA	0.0	—	0.3	V
Output high level voltage	V_{FSOH}	IFSOH = −200 µA	2.8	—	3.6	V
Output signal crossover voltage	V_{FSCRS}	—	1.3	—	2.0	V

Table 52.9 USB2.0 Host/Function-Related Pins (High Speed) (1 of 2)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Squelch detection threshold voltage (differential voltage)	V_{HSSQ}	—	100	—	150	mV
Disconnection detection threshold voltage (differential voltage)	V_{HSDSC}	—	525	—	625	mV
Common mode voltage range	V_{HSCM}	—	−50	—	500	mV
Idle state	V_{HSOI}	—	−10.0	—	10.0	mV
Output high level voltage	V_{HSOH}	—	360	—	440	mV

Table 52.9 USB2.0 Host/Function-Related Pins (High Speed) (2 of 2)

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Output low level voltage	V_{HSOL}	—	−10.0	—	10.0	mV
Chirp J output voltage (differential)	V_{CHIRPJ}	—	700	—	1100	mV
Chirp K output voltage (differential)	V_{CHIRPK}	—	−900	—	−500	mV

Table 52.10 Supply Current

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Normal operation	I_{vdd}	ICLK = 200 MHz, CPU0CLK = 800 MHz, Tj ≤ 110°C	—	—	400	mA
	I_{VCC33}	*1	—	60	—	mA
	$I_{VCC1833_2}$	*1	—	9	—	mA
	$I_{VCC1833_3}$	*1	—	9	—	mA
	I_{VCC18_PLL0}	—	—	—	6	mA
	I_{VCC18_PLL1}	—	—	—	6	mA
	I_{VCC18_USB}	—	—	—	39	mA
	I_{VCC18_ADC0}	—	—	—	3	mA
	I_{VCC18_ADC1}	—	—	—	3	mA
	I_{VCC18_TSU}		—	—	2	mA
	I_{VCC33_USB}	—	—	—	6	mA
Low power consumption mode*2	I_{vdd}	All modules inactive	—	12	—	mA
	I_{VCC33}	—	—	9	—	mA
	$I_{VCC1833_2}$	—	—	2	—	mA
	$I_{VCC1833_3}$	—	—	2	—	mA
	I_{VCC18_PLL0}	—	—	3.5	—	mA
	I_{VCC18_PLL1}	—	—	0.1	—	mA
	I_{VCC18_USB}	—	—	0.5	—	mA
	I_{VCC18_ADC0}	—	—	0.2	—	mA
	I_{VCC18_ADC1}	—	—	0.2	—	mA
	I_{VCC18_TSU}	—	—	0.1	—	mA
	I_{VCC33_USB}	—	—	0.3	—	mA

Note: These values are reference values. The actual operating current greatly depends on the system (such as unsharpened waveforms due to I/O load and toggle frequency). Be sure to measure these current values in the system.

Note 1. IO supply current (I_{VCC33}, $I_{VCC1833_n\ (n = 0\ to\ 4)}$) should be 80 mA or less. (ΣIOH in Table 52.11)

Note 2. All applicable modules are stopped or standby mode with the lowest clock frequency setting, no pull-up/down or operation for all I/O ports, and room temperature.

Table 52.11 Permissible Output Currents

Item	Symbol	Conditions	Min.	Typ.	Max.	Unit
Permissible output low current (max. value per pin)	IOL	All output pins	—	—	12	mA
Permissible output low current (total)	ΣIOL	Sum of all output pins	—	—	80	mA
Permissible output high current (max. value per pin)	IOH	All output pins	—	—	-12	mA
Permissible output high current (total)	ΣIOH	Sum of all output pins	—	—	-80	mA

Note: All output current values shall be within the values in this table to ensure the reliability of this LSI.

Table 52.12 Thermal Resistance value (Reference)

Item	Symbol	Package	Max.	Unit
Thermal Resistance	Θja	196 pin, FBGA 12 × 12 mm, 0.8-mm pitch	25.4	°C/W
	Ψjt	196 pin, FBGA 12 × 12 mm, 0.8-mm pitch	0.38	°C/W

Note: Package thermal resistance values above are based on EIA/JESD51-9 (2s2p) condition and reference only.

52.5 AC Characteristics

Table 52.13 Operating frequency

Parameter		Symbol	Min.	Max.	Unit
Operating frequency	CPU clock (CPU0CLK)	f	150 200	600 800	MHz
	System clock (ICLK)		150	200	
	Peripheral module clock H (PCLKH)		150	200	
	Peripheral module clock M (PCLKM)		75	100	
	Peripheral module clock L (PCLKL)		37.5	50	
	Peripheral module clock for ADC (PCLKADC)		18.75	25	
	Peripheral module clock for SCIn (PCLKSCIn, n = 0 to 5)		75	100	
	Peripheral module clock for SPIn (PCLKSPIn, n = 0 to 3)		75	100	
	External bus clock output (CKIO)		18.75	100	
	Ethernet PHY reference clock (ETHn_REFCLK, n = 0 to 2)		25		
	Ethernet PHY reference clock (RMIIn_REFCLK, n = 0 to 2)		50		

AC Characteristics are defined in condition of the IO setting (DRCTLm register setting) show in Table 52.14.

Table 52.14 IO setting (DRCTLm register setting) condition (1 of 2)

Module	Signal		IO type	Voltage	DRCTLm register		
					DRVn	SRn	SMTn
Bus	CKIO	SDRAM and High drive	—	3.3 V	High	Fast	—
		Other than the above	—	3.3 V	Middle	Fast	—
	Other than the above		Type A	3.3 V	Middle	Slow	Disable
			Type B	3.3 V	Low	Slow	Disable
DMAC, MTU3, IIC, CANFD, DSMIF	All signals		Type A	3.3 V	Middle	Slow	Disable
			Type B	3.3 V	Low	Slow	Disable
GPT (n = 0 to 17) (m = 0, 1)	GTIOCnA, GTIOCnB		Type A	3.3 V	Middle	Slow	Disable
			Type B	3.3 V	Low	Slow	Disable
	GTADSMLm, GTADSMPm		—	3.3 V	Low	Slow	Disable
SCI, SPI	All signals		—	3.3 V	High	Fast	Disable
xSPI (n = 0, 1; m = 0, 1)	XSPIn_CKP, XSPI0_CKN, XSPIn_IO[7:0], XSPIn_CSm#, XSPIn_DS		—	1.8 V*1	High	Fast	Disable
			—	3.3 V	High	Fast	Enable
	Other than the above		—	—	Low	Slow	Disable

Table 52.14 IO setting (DRCTLm register setting) condition (2 of 2)

Module	Signal	IO type	Voltage	DRCTLm register		
				DRVn	SRn	SMTn
Ethernet Interface (n = 0 to 2)	ETHn_TXCLK, ETHn_TXD[3:0]	—	1.8 V*1/3.3 V	High	Fast	Disable
	ETH2_TXER	—	3.3 V	Middle	Fast	—
	ETHn_RXCLK, ETHn_RXD[3:0]	—	1.8 V*1/3.3 V	—	—	Disable
	ETHn_RXER, ETH2_COL, ETH2_CRS	—	3.3 V	—	—	Disable
	ETHn_REFCLK, RMIIn_REFCLK	—	3.3 V	High	Fast	—
	Other than the above	—	—	Low	Slow	Disable
SHOSTIF	HSPI_CK, HSPI_CS#, HSPI_IO[7:0]	—	3.3 V	High	Fast	Disable
	HSPI_INT#	—	3.3 V	Low	Slow	Disable
Debug Interface	TRACECLK, TRACECTL, TRACEDATA[7:0], TDO, TMS	—	3.3 V	High	Fast	Disable
	Other than the above	—	—	Low	Slow	Disable
Other than the above		—	—	Low	Slow	Disable

Note 1. 1.8 V is only supported for xSPI0 and Ethernet port 2 (ETH2_).

52.5.1 Clock Timing

52.5.1.1 CKIO Pin Output Timing

Table 52.15 CKIO pin output timing

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
CKIO pin output cycle time	t_{CKcyc}	Figure 52.2	10	—	53.4	ns
CKIO pin output high level pulse width	t_{CKH}		t_{CKcyc} / 2 − t_{CKr}	—	—	ns
CKIO pin output low level pulse width	t_{CKL}		t_{CKcyc} / 2 − t_{CKf}	—	—	ns
CKIO pin output rising time 1	t_{CKr}	CKIO: High drive output Setting	—	—	3.8	ns
CKIO pin output falling time 1	t_{CKf}		—	—	3.8	ns
CKIO pin output rising time 2	t_{CKr}	CKIO: Normal output setting	—	—	9	ns
CKIO pin output falling time 2	t_{CKf}		—	—	9	ns

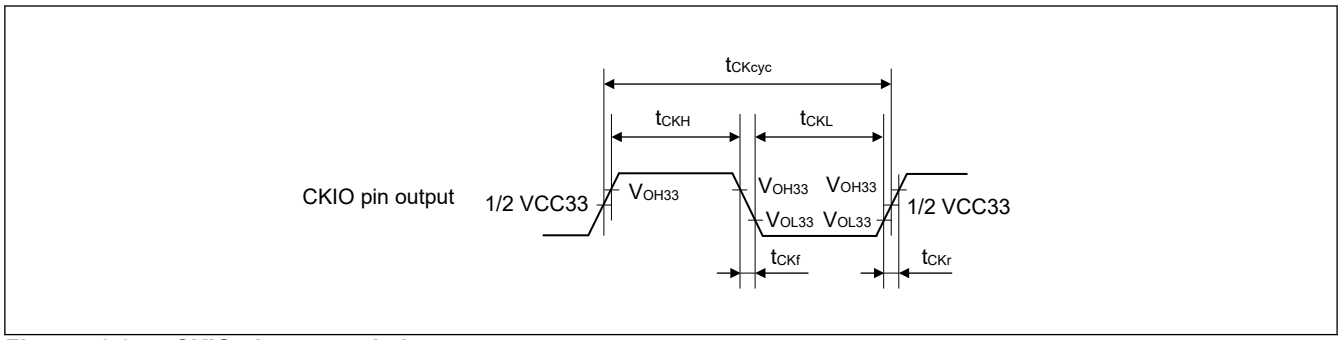


Figure 52.2 CKIO pin output timing

52.5.1.2 Ethernet PHY Reference Clock Output Timing

Conditions:

C = 30 pF (ETHn_REFCLK)

C = 20 pF (RMIIn_REFCLK)

Table 52.16 Ethernet PHY reference clock output timing

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
ETHn_REFCLK cycle time	t_{CK}	Figure 52.3	40	—	—	ns
ETHn_REFCLK frequency	—		25.00 ± 50 ppm			MHz
ETHn_REFCLK duty	—		45	—	55	%
ETHn_REFCLK rising/falling time	t_{CKr} / t_{CKf}		0.5	—	4.0	ns
RMIIn_REFCLK cycle time	t_{CK}		20	—	—	ns
RMIIn_REFCLK frequency	—		50.00 ± 50 ppm			MHz
RMIIn_REFCLK duty	—		45	—	55	%
RMIIn_REFCLK rising/falling time	t_{CKr} / t_{CKf}		0.5	—	3.5	ns

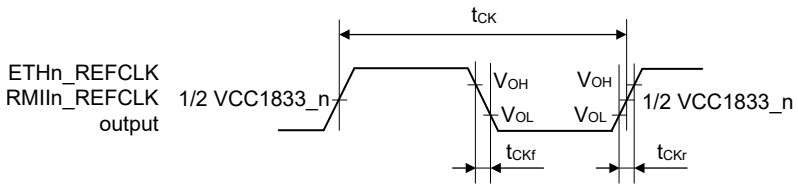


Figure 52.3 Ethernet PHY reference clock output timing

52.5.1.3 EXTCLKIN External Clock Input

Table 52.17 EXTCLKIN clock timing

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
EXTCLKIN external clock frequency	$f_{EXTCLKIN}$	—	25.00 ± 50 ppm			MHz
		EtherCAT in use	25.00 ± 25 ppm			MHz
EXTCLKIN duty	$r_{EXTCLKIN}$	—	±5%			—
EXTCLKIN rising time	$t_{rEXTCLKIN}$	—	0	—	5	ns
EXTCLKIN falling time	$t_{fEXTCLKIN}$	—	0	—	5	ns

Note: When using crystal resonator (i.e. EXTA/XTAL clock is used), EXTCLKIN should be driven low.

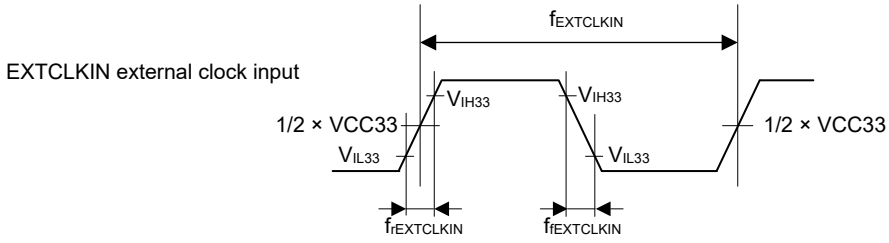


Figure 52.4 EXTCLKIN external clock input timing

52.5.1.4 EXTAL/XTAL Clock Timing

Table 52.18 EXTAL/XTAL clock timing

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
EXTAL/XTAL clock frequency*1	f_{XTAL}	—	25.00 ± 50 ppm			MHz
		EtherCAT in use	25.00 ± 25 ppm			MHz

Note: When using an external oscillator, be sure to leave XTAL open-circuit and make sure that EXTAL is driven low.
Note 1. When using the EXTAL/XTAL clock (i.e. crystal resonator), ask the oscillator manufacturer to evaluate oscillation of the oscillator. For the oscillation stabilization time, see the evaluation result provided by the oscillator manufacturer.

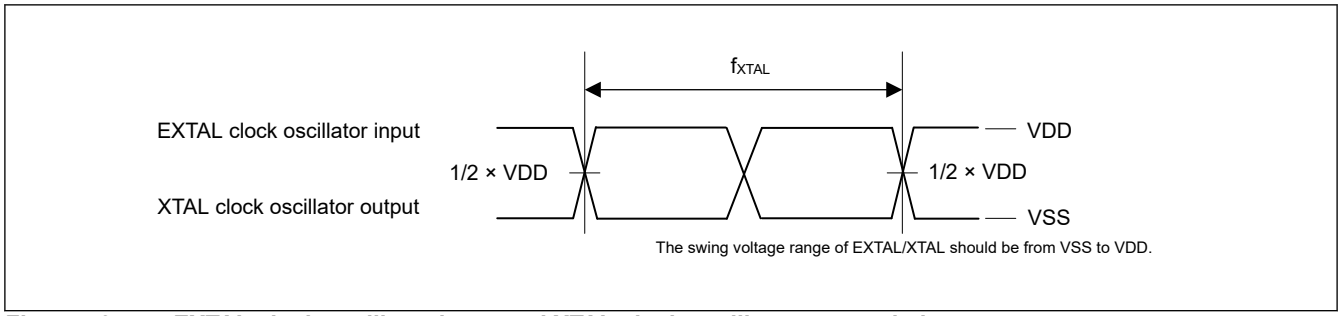


Figure 52.5 EXTAL clock oscillator input and XTAL clock oscillator output timing

52.5.1.5 LOCO Clock Timing

Table 52.19 LOCO clock timing

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
LOCO clock cycle time	t_{Lcyc}	—	4.62	4.17	3.79	µs
LOCO clock oscillation frequency	t_{LOCO}	—	216	240	264	kHz
LOCO clock oscillation stabilization wait time	t_{LOCOWT}	—	—	—	40	µs

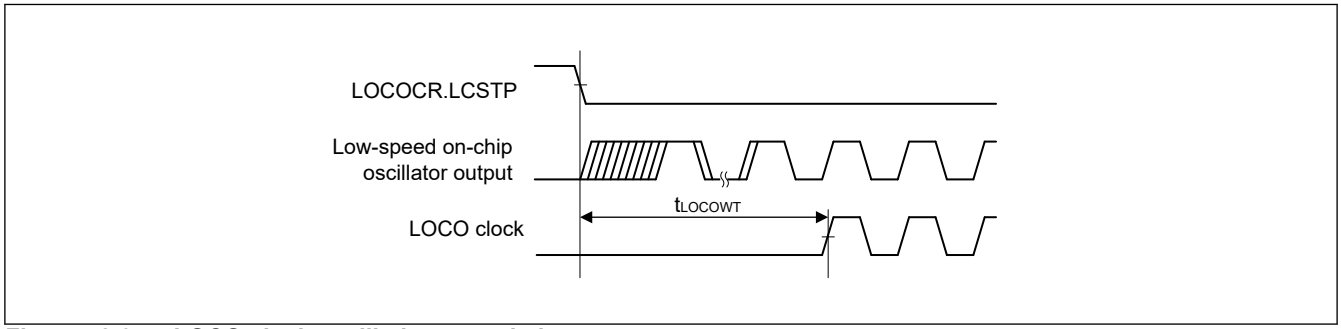


Figure 52.6 LOCO clock oscillation start timing

52.5.2 Reset, Interrtup, and Mode Timing

Table 52.20 Reset, interrupt, and mode timing (1 of 2)

Parameter		Symbol	Conditions	Min.*1	Typ.	Max.	Unit
RES# pulse width	At power on	$t_{dlyreset}$	Figure 52.7	10	—	—	ms
	Other than above	$t_{dlyreset2}$		1	—	—	ms
RES# rising time		$t_{risereset}$		—	—	150	µs
TRST# pulse width	At power on	$t_{dlyreset}$		10	—	—	ms
	Other than above	$t_{dlyreset2}$		1	—	—	ms
TRST# rising time		$t_{risereset}$		—	—	150	µs

Table 52.20 Reset, interrupt, and mode timing (2 of 2)

Parameter		Symbol	Conditions	Min.*1	Typ.	Max.	Unit
NMI pulse width		t_{NMIW}	Figure 52.8	t_{Icyc} × 2	—	—	ns
IRQ pulse width		t_{IRQW}	Figure 52.9	t_{Icyc} × 2	—	—	ns
Mode hold time (to RES#)	At power on	t_{MDH}	Figure 52.10	250	—	—	ns

Note 1. t_{Icyc}: ICLK cycle

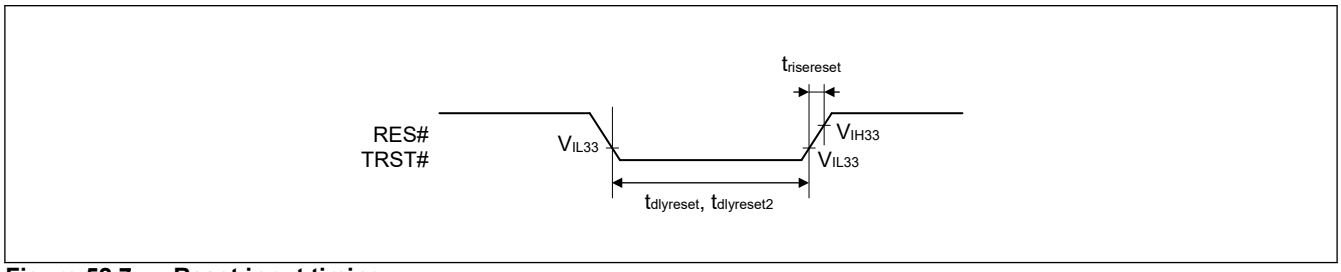


Figure 52.7 Reset input timing

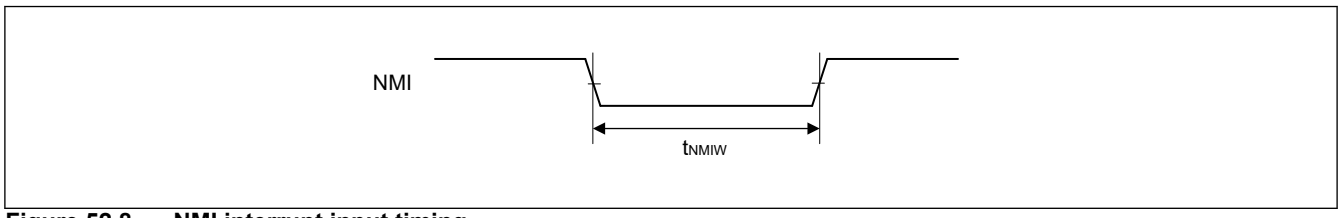


Figure 52.8 NMI interrupt input timing

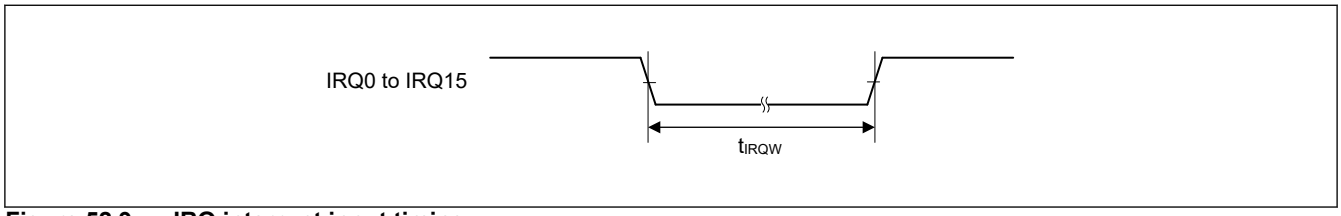


Figure 52.9 IRQ interrupt input timing

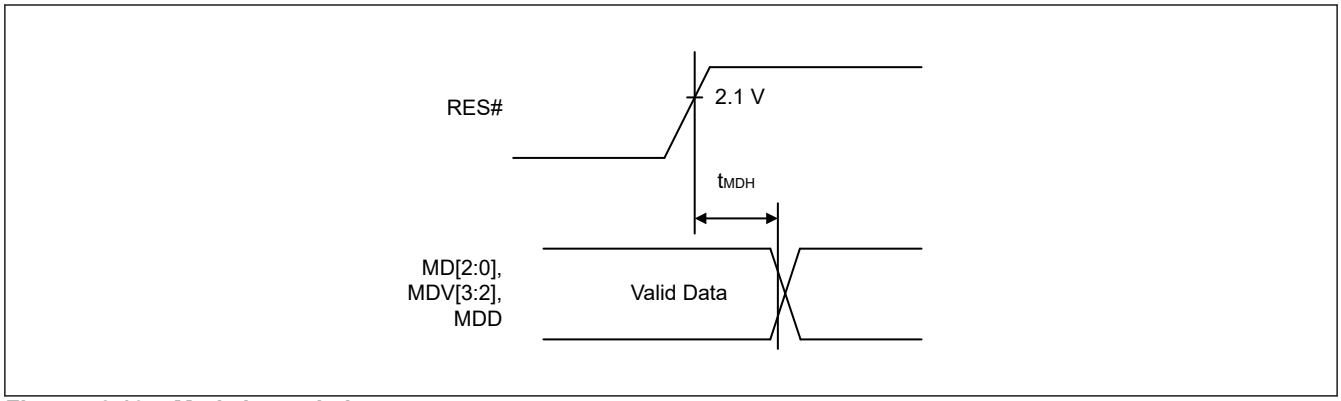


Figure 52.10 Mode input timing

52.5.3 Bus Timing

Table 52.21 Bus timing (1 of 2)

Conditions: V_{OH} = VCC33 × 0.5, V_{OL} = VCC33 × 0.5, C = 15 pF (CKIO), 30 pF (others), $T_j$min = −40°C

Parameter		Symbol	CKIO = $1/t_{CKcyc}$ *1 (Max 66 MHz)		Unit	Reference Figure
			Min.	Max.		
Address delay time 1	SDRAM*2	t_{AD1}	2	11	ns	Figure 52.11 to Figure 52.35
	Other than the above		0	10	ns	
Address delay time 2		t_{AD2}	$1/2t_{CKcyc}$	$1/2t_{CKcyc}$ + 10	ns	Figure 52.18
Address setup time		t_{AS}	0	—	ns	Figure 52.11 to Figure 52.14, Figure 52.18
Chip enable setup time		t_{CS}	0	—	ns	Figure 52.11 to Figure 52.14, Figure 52.18
Address hold time		t_{AH}	0	—	ns	Figure 52.11 to Figure 52.14
BS delay time		t_{BSD}	—	11	ns	Figure 52.11 to Figure 52.32
CSn# delay time 1	SDRAM*2	t_{CSD1}	2	11	ns	Figure 52.11 to Figure 52.35
	Other than the above		0	10	ns	
Read/write delay time 1	SDRAM*2	t_{RWD1}	2	11	ns	Figure 52.11 to Figure 52.35
	Other than the above		0	10	ns	
Read strobe delay time		t_{RSD}	$1/2t_{CKcyc}$	$1/2t_{CKcyc}$ + 10	ns	Figure 52.11 to Figure 52.18
Read data setup time 1*3	High-drive output	t_{RDS1}	$1/2t_{CKcyc}$ + 4	—	ns	Figure 52.11 to Figure 52.17
	Normal output		$1/2t_{CKcyc}$ + 7	—	ns	
Read data setup time 2*3	High-drive output	t_{RDS2}	6.6	—	ns	Figure 52.19 to Figure 52.22, Figure 52.27 to Figure 52.29
Read data setup time 3*3	High-drive output	t_{RDS3}	$1/2t_{CKcyc}$ + 4	—	ns	Figure 52.18
	Normal output		$1/2t_{CKcyc}$ + 7	—	ns	
Read data hold time 1		t_{RDH1}	0	—	ns	Figure 52.11 to Figure 52.17
Read data hold time 2		t_{RDH2}	2.5	—	ns	Figure 52.19 to Figure 52.22, Figure 52.27 to Figure 52.29
Read data hold time 3		t_{RDH3}	0	—	ns	Figure 52.18
Write enable delay time 1		t_{WED1}	$1/2t_{CKcyc}$	$1/2t_{CKcyc}$ + 10	ns	Figure 52.11 to Figure 52.16
Write enable delay time 2		t_{WED2}	—	11	ns	Figure 52.17
Write data delay time 1		t_{WDD1}	—	11	ns	Figure 52.11 to Figure 52.17
Write data delay time 2		t_{WDD2}	—	11	ns	Figure 52.23 to Figure 52.26, Figure 52.30 to Figure 52.32
Write data hold time 1		t_{WDH1}	1	—	ns	Figure 52.11 to Figure 52.17
Write data hold time 2		t_{WDH2}	2	—	ns	Figure 52.23 to Figure 52.26, Figure 52.30 to Figure 52.32
Write data hold time 4		t_{WDH4}	0	—	ns	Figure 52.11 to Figure 52.15
WAIT# setup time*3	High-drive output	t_{WTS}	$1/2t_{CKcyc}$ + 4.5	—	ns	Figure 52.12 to Figure 52.18
	Normal output		$1/2t_{CKcyc}$ + 8	—	ns	
WAIT# hold time		t_{WTH}	$1/2t_{CKcyc}$ + 3.5	—	ns	Figure 52.12 to Figure 52.18
RAS# delay time 1		t_{RASD1}	2	11	ns	Figure 52.19 to Figure 52.35
CAS# delay time 1		t_{CASD1}	2	11	ns	Figure 52.19 to Figure 52.35

Table 52.21 Bus timing (2 of 2)

Conditions: V_{OH} = VCC33 × 0.5, V_{OL} = VCC33 × 0.5, C = 15 pF (CKIO), 30 pF (others), $T_j$min = −40°C

Parameter	Symbol	CKIO = $1/t_{CKcyc}$ *1 (Max 66 MHz)		Unit	Reference Figure
		Min.	Max.		
DQM delay time 1	t_{DQMD1}	2	11	ns	Figure 52.19 to Figure 52.32
CKE delay time 1	t_{CKED1}	2	11	ns	Figure 52.34
AH# delay time	t_{AHD}	$1/2t_{CKcyc}$	$1/2t_{CKcyc}$ + 10	ns	Figure 52.15
Multiplex address delay time	t_{MAD}	—	10	ns	Figure 52.15
Multiplex address hold time	t_{MAH}	1	—	ns	Figure 52.15
Address setup time to AH#	t_{AVVH}	$1/2t_{CKcyc}$ − 2	—	ns	Figure 52.15
DACK/TEND delay time	t_{DACD}	See section 52.5.4. DMAC Timing		ns	Figure 52.11 to Figure 52.32

Note: Notation of $1/2t_{CKcyc}$ in the delay time, setup time, and hold time shows 1/2 cycles from the clock rising edge, that is, the reference of clock falling.

Note 1. Take the number of cycles of waiting that suits the system configuration into consideration with regard to the fmax value for CKIO (the external bus clock). When CKIO is running at 50 MHz or a higher frequency, select high drive output.

Note 2. These are values when SDRAM (CSnBCR.TYPE[2:0] = 100b) is selected in the CSn space bus control register (CSnBCR) and high-drive output is selected for CKIO.

Note 3. These are values when high-drive output and normal output are respectively selected for CKIO.

Table 52.22 Bus timing (1 of 2)

Conditions: V_{OH} = VCC33 × 0.5, V_{OL} = VCC33 × 0.5, C = 12 pF (CKIO), 12 pF (others), $T_j$min = −20°C

Parameter		Symbol	CKIO = $1/t_{CKcyc}$ *1 (Max 100 MHz)		Unit	Reference Figure
			Min.	Max.		
Address delay time 1	SDRAM*2	t_{AD1}	1.3	8	ns	Figure 52.11 to Figure 52.35
	Other than the above		0	8	ns	
Address delay time 2		t_{AD2}	$1/2t_{CKcyc}$ - 0.5	$1/2t_{CKcyc}$ + 8	ns	Figure 52.18
Address setup time		t_{AS}	0	—	ns	Figure 52.11 to Figure 52.14, Figure 52.18
Chip enable setup time		t_{CS}	0	—	ns	Figure 52.11 to Figure 52.14, Figure 52.18
Address hold time		t_{AH}	0	—	ns	Figure 52.11 to Figure 52.14
BS delay time		t_{BSD}	—	8	ns	Figure 52.11 to Figure 52.32
CSn# delay time 1	SDRAM*2	t_{CSD1}	1.3	8	ns	Figure 52.11 to Figure 52.35
	Other than the above		0	8	ns	
Read/write delay time 1	SDRAM*2	t_{RWD1}	1.3	8	ns	Figure 52.11 to Figure 52.35
	Other than the above		0	8	ns	
Read strobe delay time		t_{RSD}	$1/2t_{CKcyc}$	$1/2t_{CKcyc}$ + 8	ns	Figure 52.11 to Figure 52.18
Read data setup time 1*3	High-drive output	t_{RDS1}	$1/2t_{CKcyc}$ + 4.5	—	ns	Figure 52.11 to Figure 52.17
	Normal output		$1/2t_{CKcyc}$ + 7	—	ns	
Read data setup time 2*3	High-drive output	t_{RDS2}	3.5	—	ns	Figure 52.19 to Figure 52.22, Figure 52.27 to Figure 52.29
Read data setup time 3*3	High-drive output	t_{RDS3}	$1/2t_{CKcyc}$ + 4.5	—	ns	Figure 52.18
	Normal output		$1/2t_{CKcyc}$ + 7	—	ns	
Read data hold time 1		t_{RDH1}	0	—	ns	Figure 52.11 to Figure 52.17

Table 52.22 Bus timing (2 of 2)

Conditions: V_{OH} = VCC33 × 0.5, V_{OL} = VCC33 × 0.5, C = 12 pF (CKIO), 12 pF (others), $T_j$min = −20°C

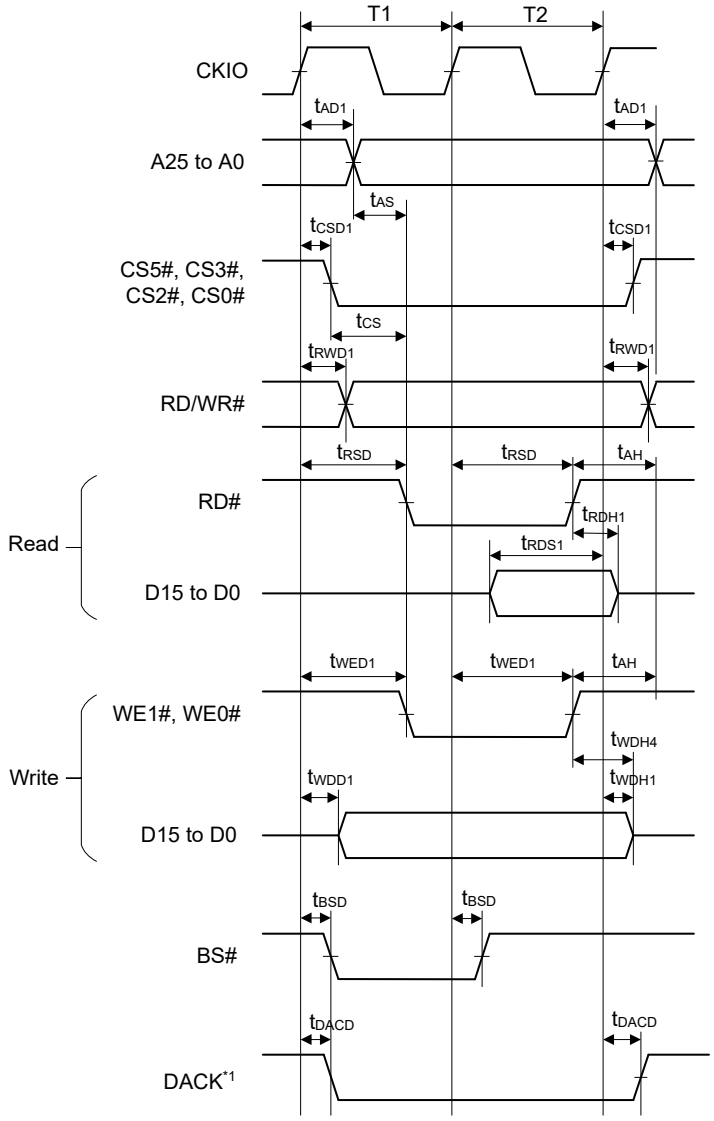
Parameter		Symbol	CKIO = $1/t_{CKcyc}$ *1 (Max 100 MHz) Min.	Max.	Unit	Reference Figure
Read data hold time 2		t_{RDH2}	2.5	—	ns	Figure 52.19 to Figure 52.22, Figure 52.27 to Figure 52.29
Read data hold time 3		t_{RDH3}	0	—	ns	Figure 52.18
Write enable delay time 1		t_{WED1}	$1/2t_{CKcyc}$	$1/2t_{CKcyc}$ + 8	ns	Figure 52.11 to Figure 52.16
Write enable delay time 2		t_{WED2}	—	9	ns	Figure 52.17
Write data delay time 1		t_{WDD1}	—	8	ns	Figure 52.11 to Figure 52.17
Write data delay time 2		t_{WDD2}	—	8	ns	Figure 52.23 to Figure 52.26, Figure 52.30 to Figure 52.32
Write data hold time 1		t_{WDH1}	1	—	ns	Figure 52.11 to Figure 52.17
Write data hold time 2		t_{WDH2}	1.3	—	ns	Figure 52.23 to Figure 52.26, Figure 52.30 to Figure 52.32
Write data hold time 4		t_{WDH4}	0	—	ns	Figure 52.11 to Figure 52.15
WAIT# setup time*3	High-drive output	t_{WTS}	$1/2t_{CKcyc}$ + 4.5	—	ns	Figure 52.12 to Figure 52.18
	Normal output		$1/2t_{CKcyc}$ + 8	—	ns	
WAIT# hold time		t_{WTH}	$1/2t_{CKcyc}$ + 3.5	—	ns	Figure 52.12 to Figure 52.18
RAS# delay time 1		t_{RASD1}	1.3	8	ns	Figure 52.19 to Figure 52.35
CAS# delay time 1		t_{CASD1}	1.3	8	ns	Figure 52.19 to Figure 52.35
DQM delay time 1		t_{DQMD1}	1.3	8	ns	Figure 52.19 to Figure 52.32
CKE delay time 1		t_{CKED1}	1.3	8	ns	Figure 52.34
AH# delay time		t_{AHD}	$1/2t_{CKcyc}$	$1/2t_{CKcyc}$ + 8	ns	Figure 52.15
Multiplex address delay time		t_{MAD}	—	8	ns	Figure 52.15
Multiplex address hold time		t_{MAH}	1	—	ns	Figure 52.15
Address setup time to AH#		t_{AVVH}	$1/2t_{CKcyc}$ − 2	—	ns	Figure 52.15
DACK/TEND delay time		t_{DACD}	See section 52.5.4. DMAC Timing		ns	Figure 52.11 to Figure 52.32

Note: Notation of $1/2t_{CKcyc}$ in the delay time, setup time, and hold time shows 1/2 cycles from the clock rising edge, that is, the reference of clock falling.

Note 1. Take the number of cycles of waiting that suits the system configuration into consideration with regard to the fmax value for CKIO (the external bus clock). When CKIO is running at 50 MHz or a higher frequency, select high drive output.

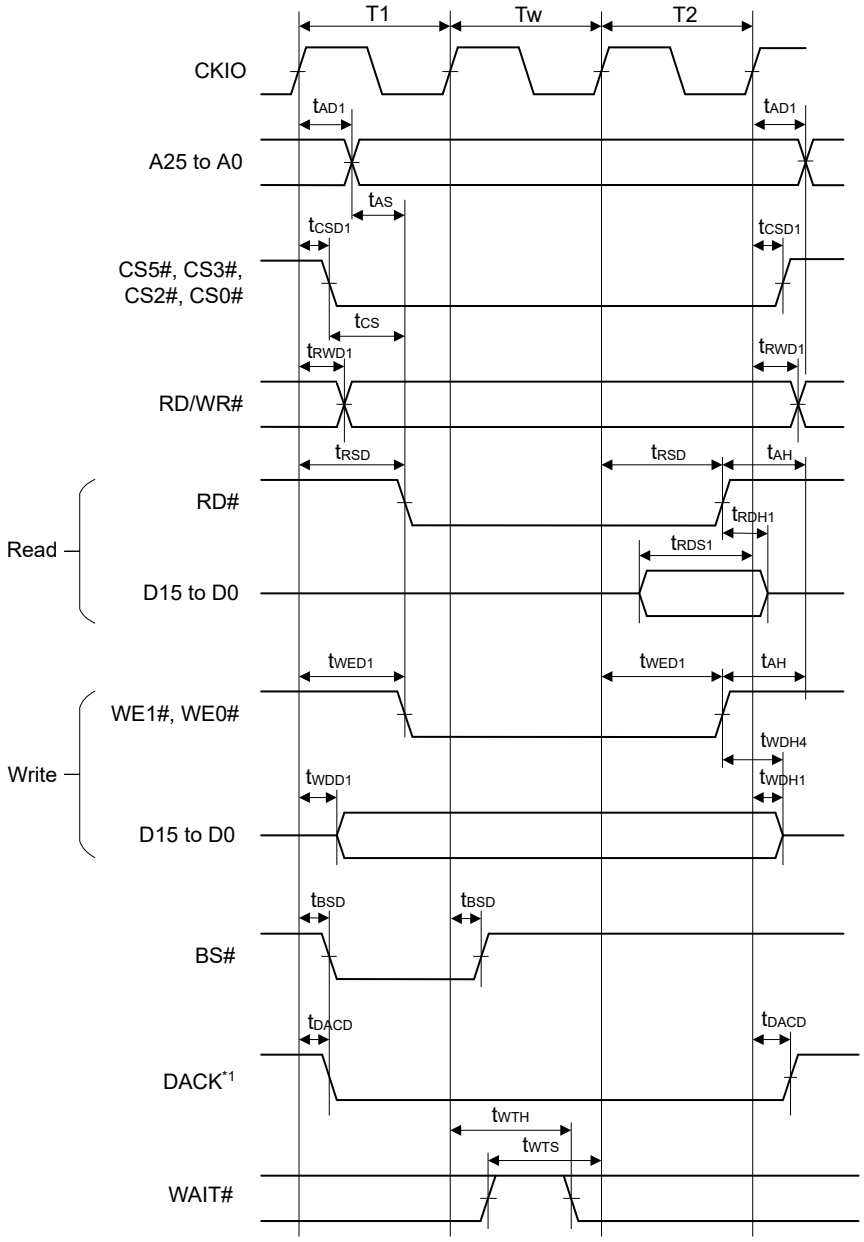
Note 2. These are values when SDRAM (CSnBCR.TYPE[2:0] = 100b) is selected in the CSn space bus control register (CSnBCR) and high-drive output is selected for CKIO.

Note 3. These are values when high-drive output and normal output are respectively selected for CKIO.



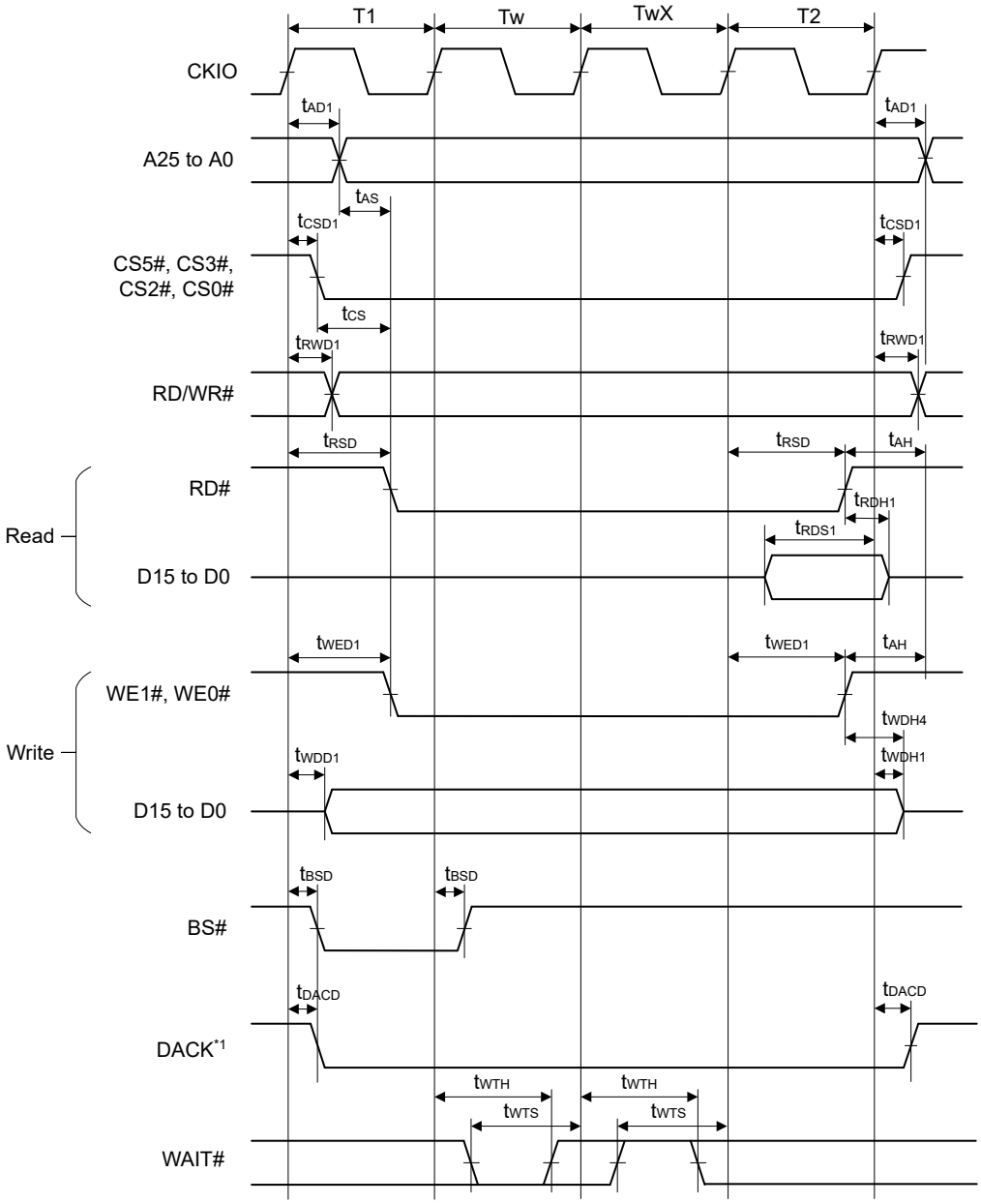
Note 1. DACK is waveform when active-low is specified.

Figure 52.11 SRAM interface basic bus cycle (no wait)



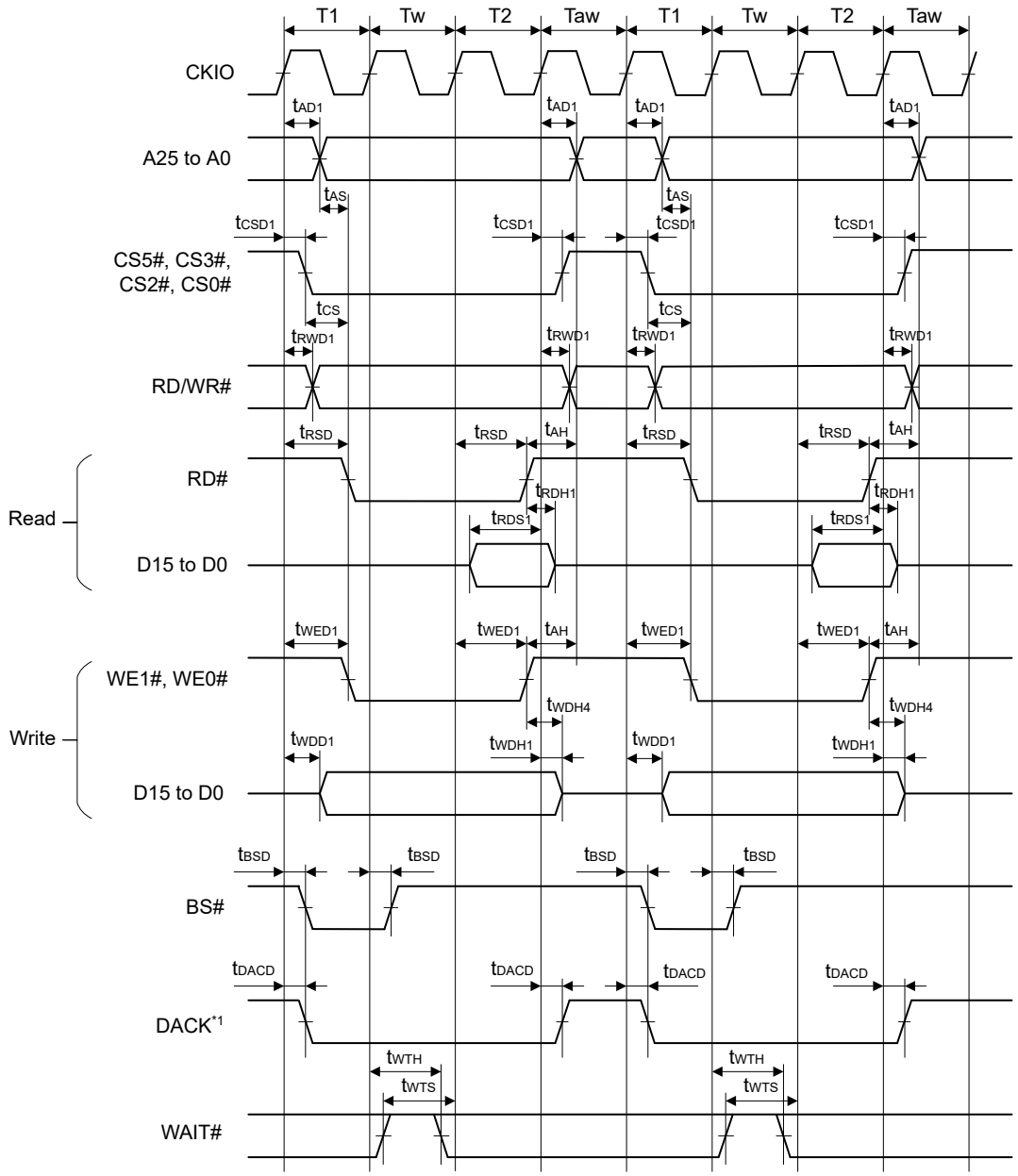
Note 1. DACK is waveform when active-low is specified.

Figure 52.12 SRAM interface basic bus cycle (software wait 1)



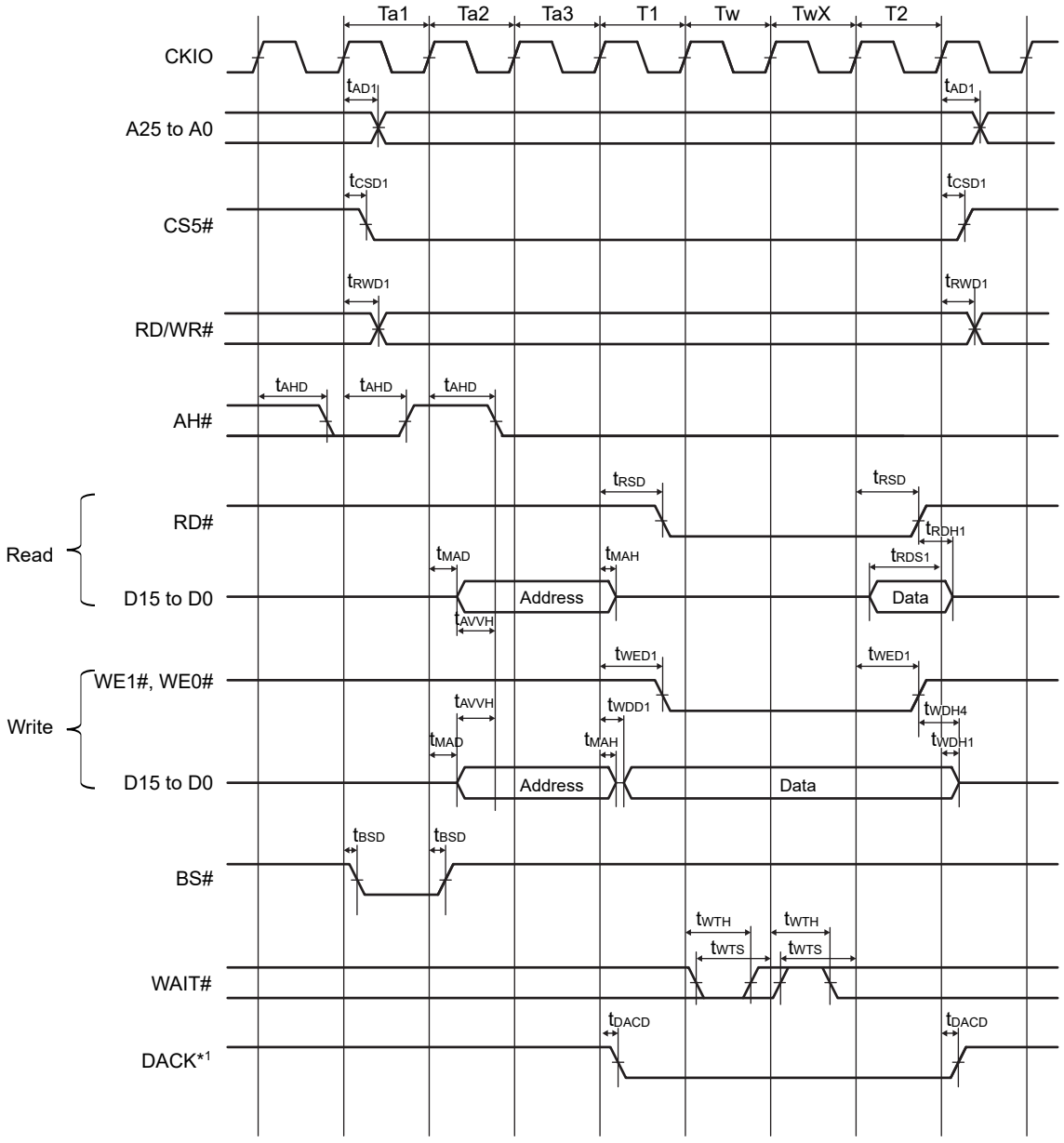
Note 1. DACK is waveform when active-low is specified.

Figure 52.13 SRAM interface basic bus cycle (software wait 1, external wait 1 inserted)



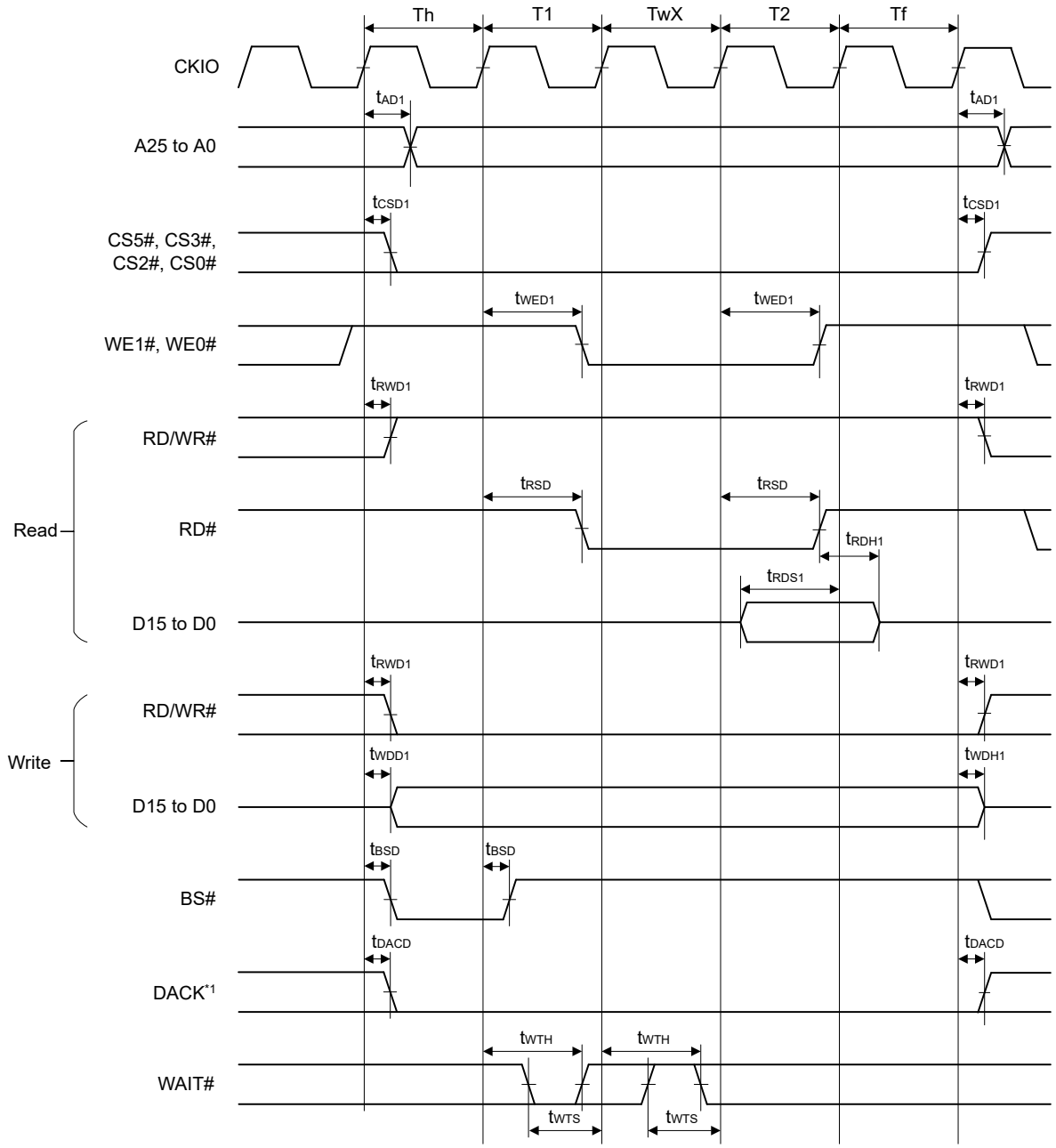
Note 1. DACK is waveform when active-low is specified.

Figure 52.14 SRAM interface basic bus cycle (software wait 1, external wait enabled (WM = 0), no idle cycle)



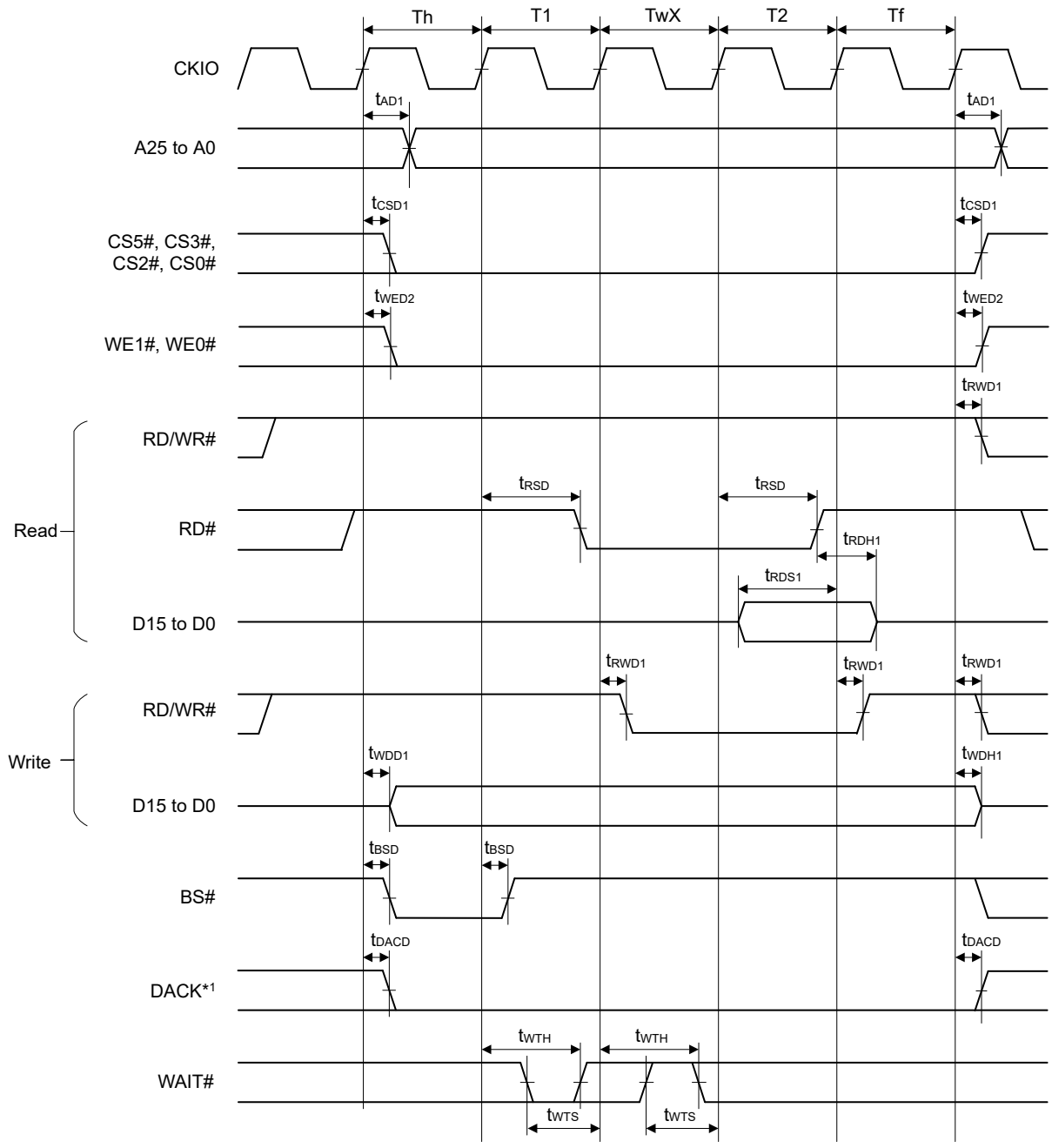
Note 1. DACK is waveform when active-low is specified.

Figure 52.15 MPX-I/O interface bus cycle (address cycle 3, software wait 1, external wait 1 inserted)



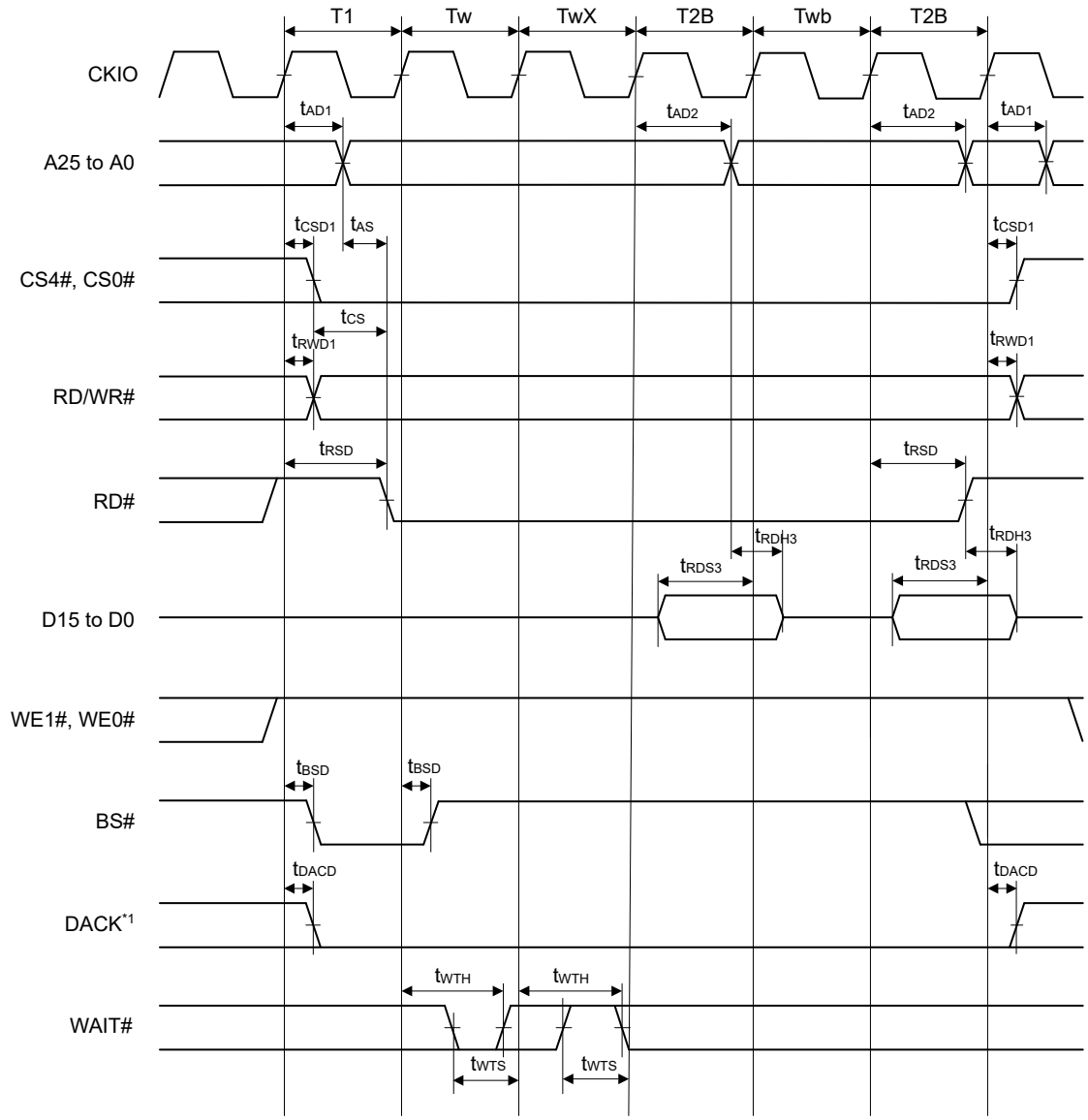
Note 1. DACK is waveform when active-low is specified.

Figure 52.16 SRAM bus cycle with byte selection (SW = 1 cycle, HW = 1 cycle, asynchronous external wait 1 inserted, BAS = 0 (write cycle UB/LB control))



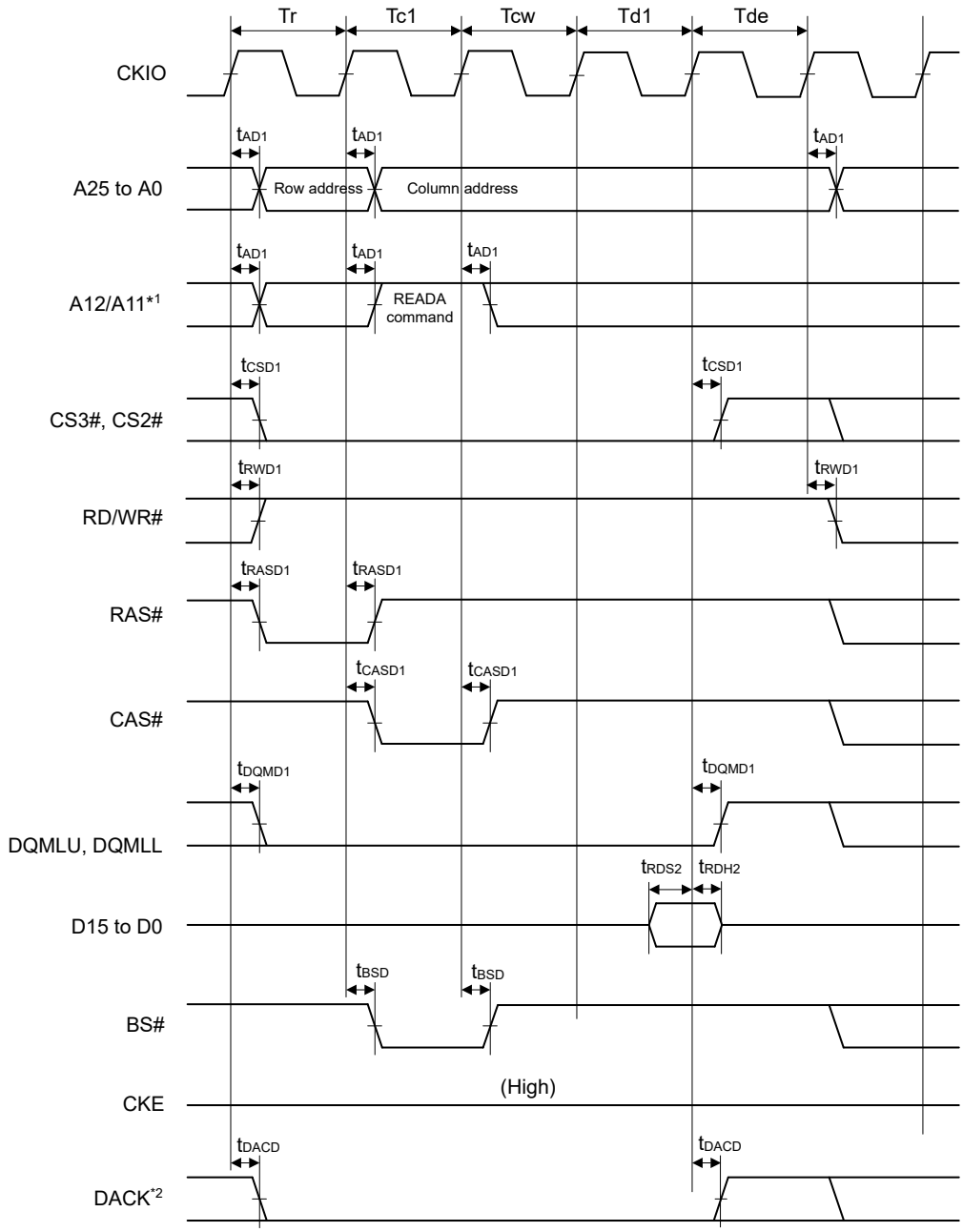
Note 1. DACK is waveform when active-low is specified.

Figure 52.17 SRAM bus cycle with byte selection (SW = 1 cycle, HW = 1 cycle, asynchronous external wait 1 inserted, BAS = 1 (write cycle WE control))



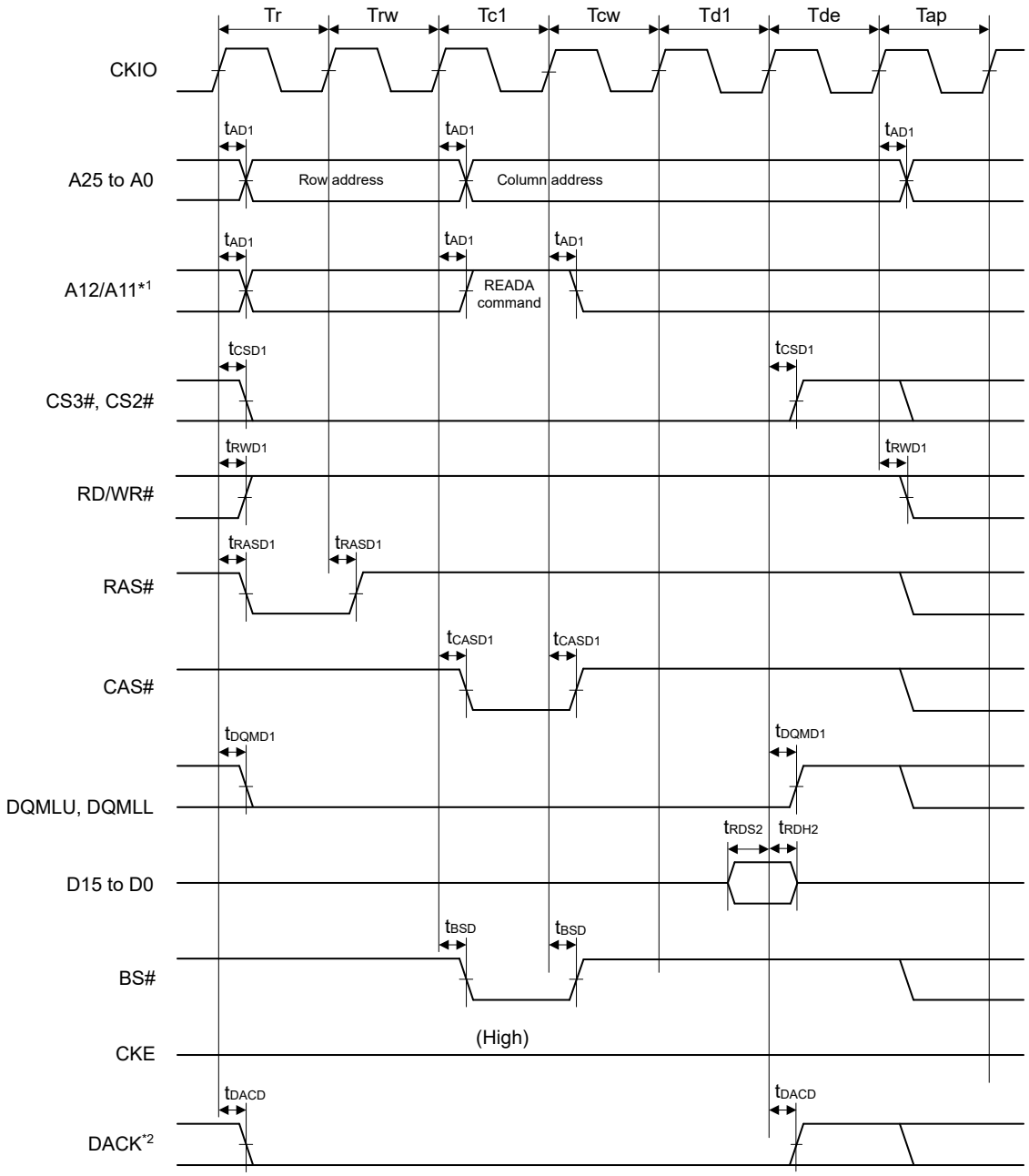
Note 1. DACK is waveform when active-low is specified.

Figure 52.18 Burst ROM read cycle (software wait 1, asynchronous external wait 1 inserted, burst wait 1, 2)



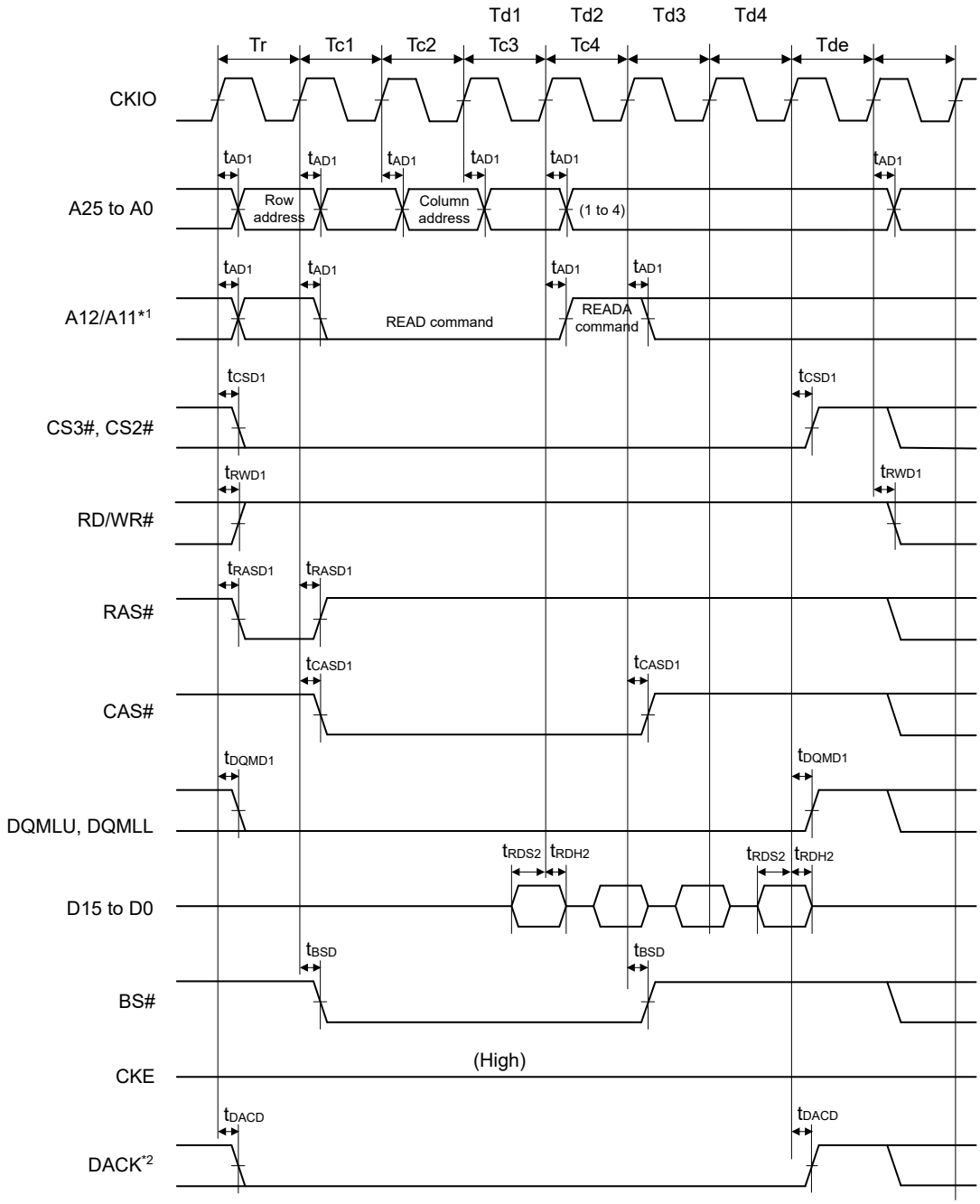
Note 1. Address pin to be connected to A10 of SDRAM.
Note 2. DACK is waveform when active-low is specified.

Figure 52.19 Synchronous DRAM single-read bus cycle (with auto precharge, CAS latency 2, WTRCD = 0 cycles, WTRP = 0 cycles)



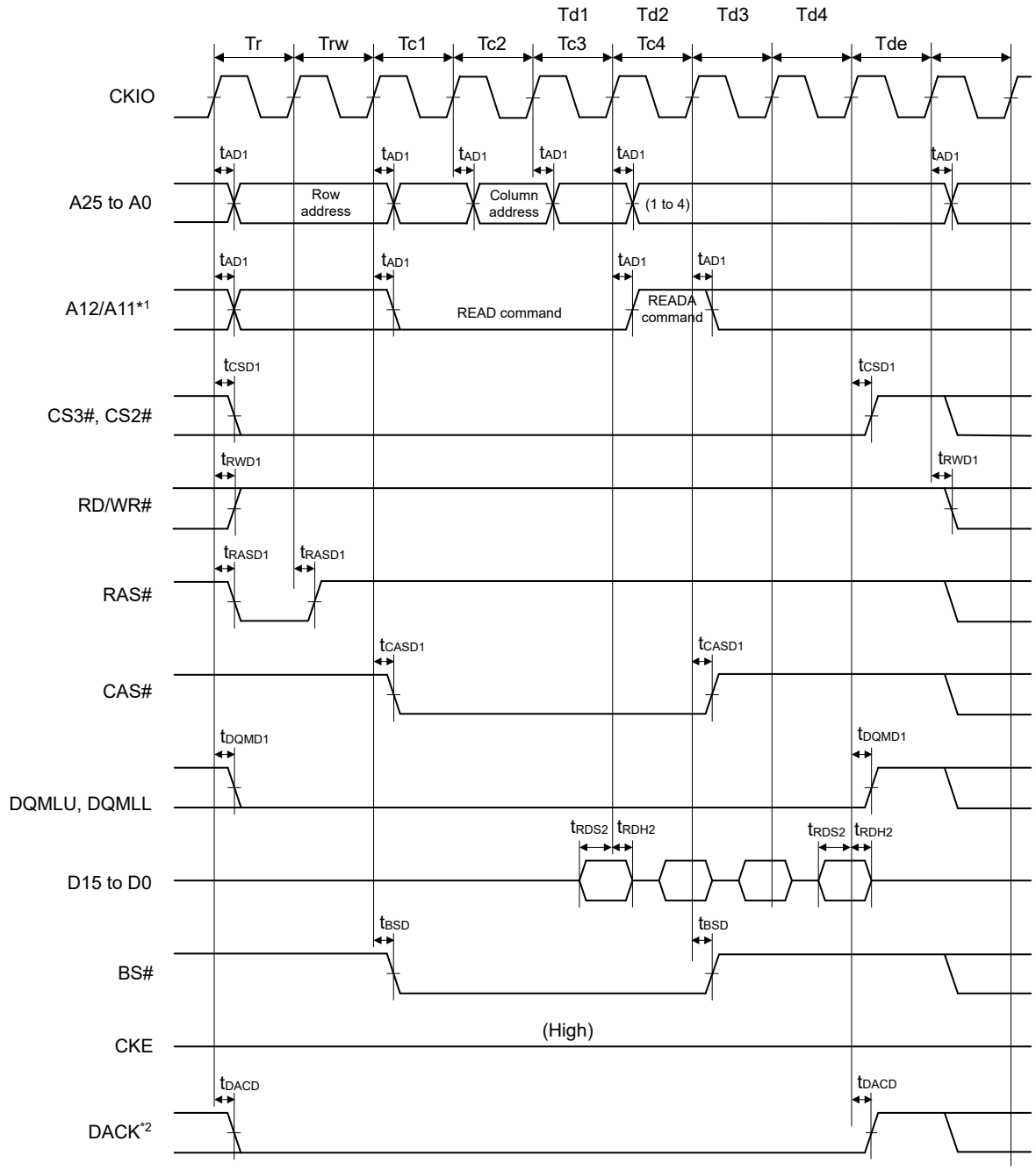
Note 1. Address pin to be connected to A10 of SDRAM.
Note 2. DACK is waveform when active-low is specified.

Figure 52.20 Synchronous DRAM single-read bus cycle (with auto precharge, CAS latency 2, WTRCD = 1 cycle, WTRP = 1 cycle)



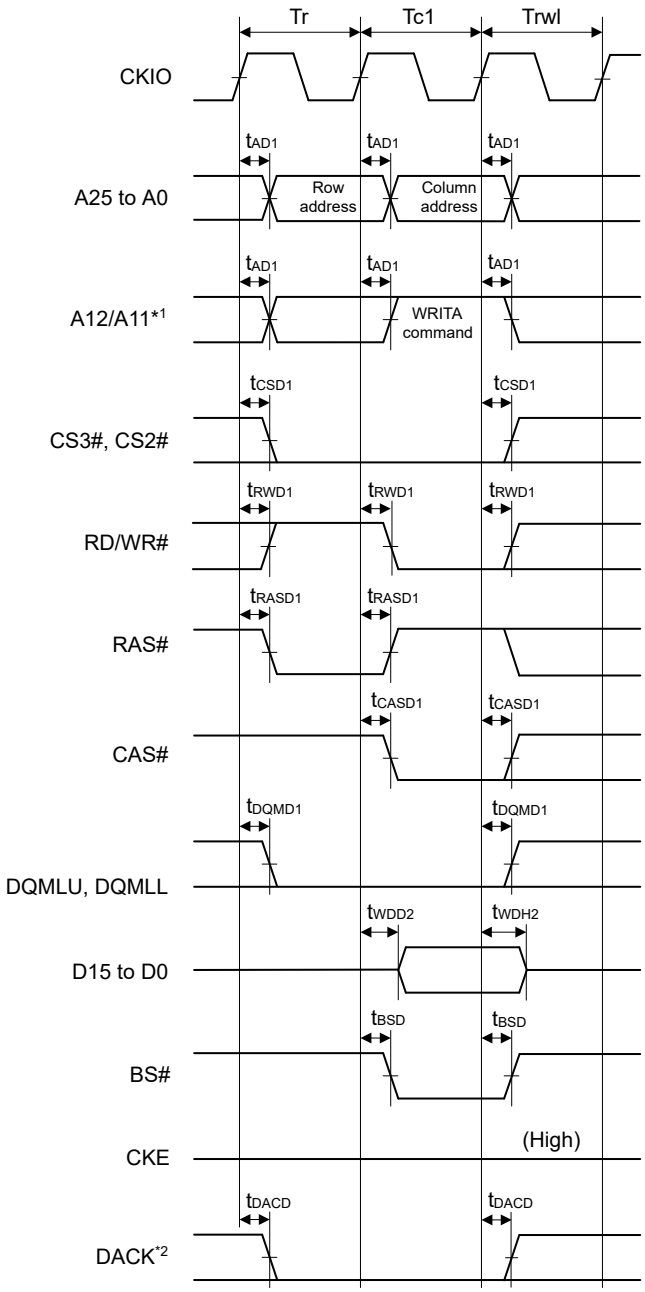
Note 1. Address pin to be connected to A10 of SDRAM.
Note 2. DACK is waveform when active-low is specified.

Figure 52.21 Synchronous DRAM burst-read bus cycle (read for 4 cycles) (with auto precharge, CAS latency 2, WTRCD = 0 cycles, WTRP = 1 cycle)



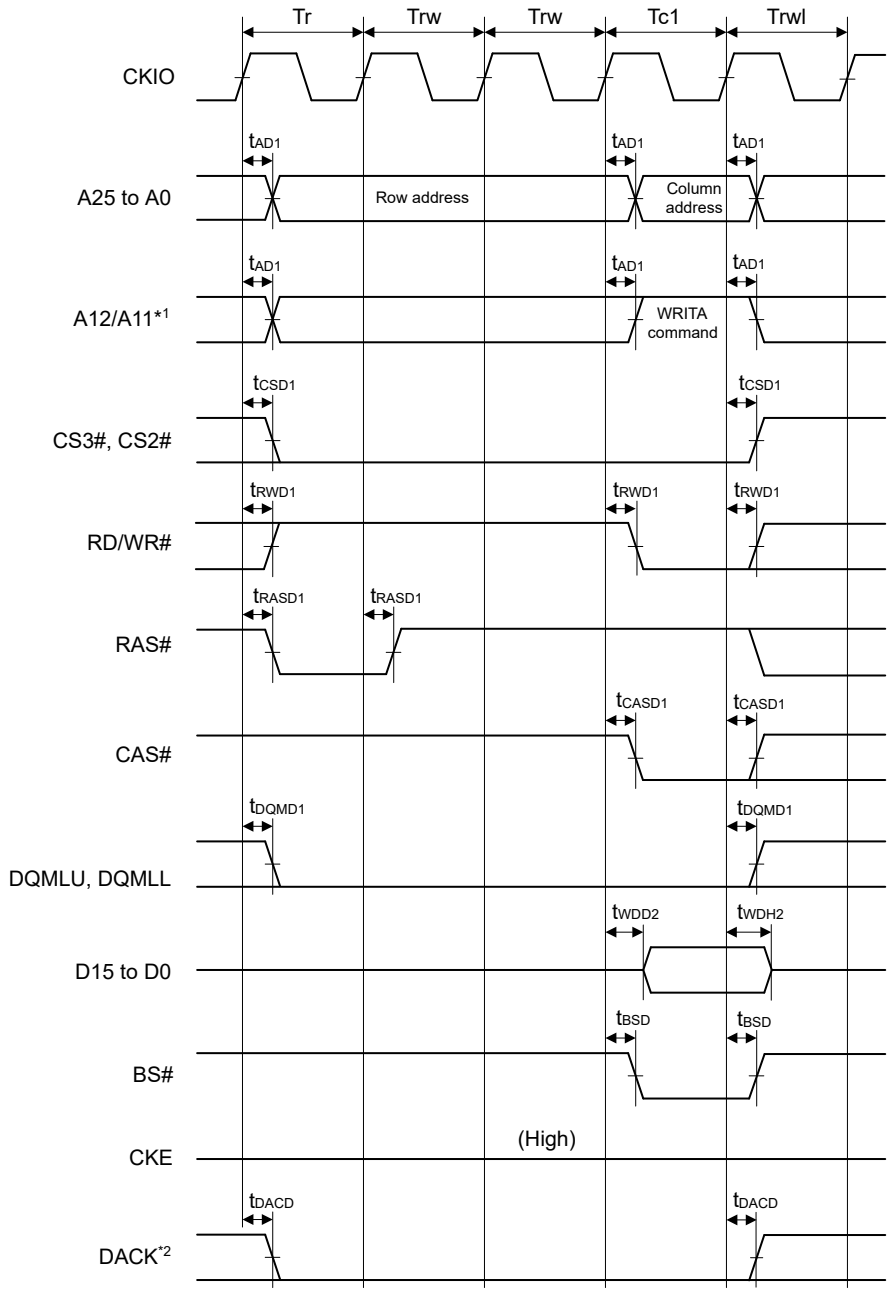
Note 1. Address pin to be connected to A10 of SDRAM.
Note 2. DACK is waveform when active-low is specified.

Figure 52.22 Synchronous DRAM burst-read bus cycle (read for 4 cycles) (with auto precharge, CAS latency 2, WTRCD = 1 cycle, WTRP = 0 cycles)



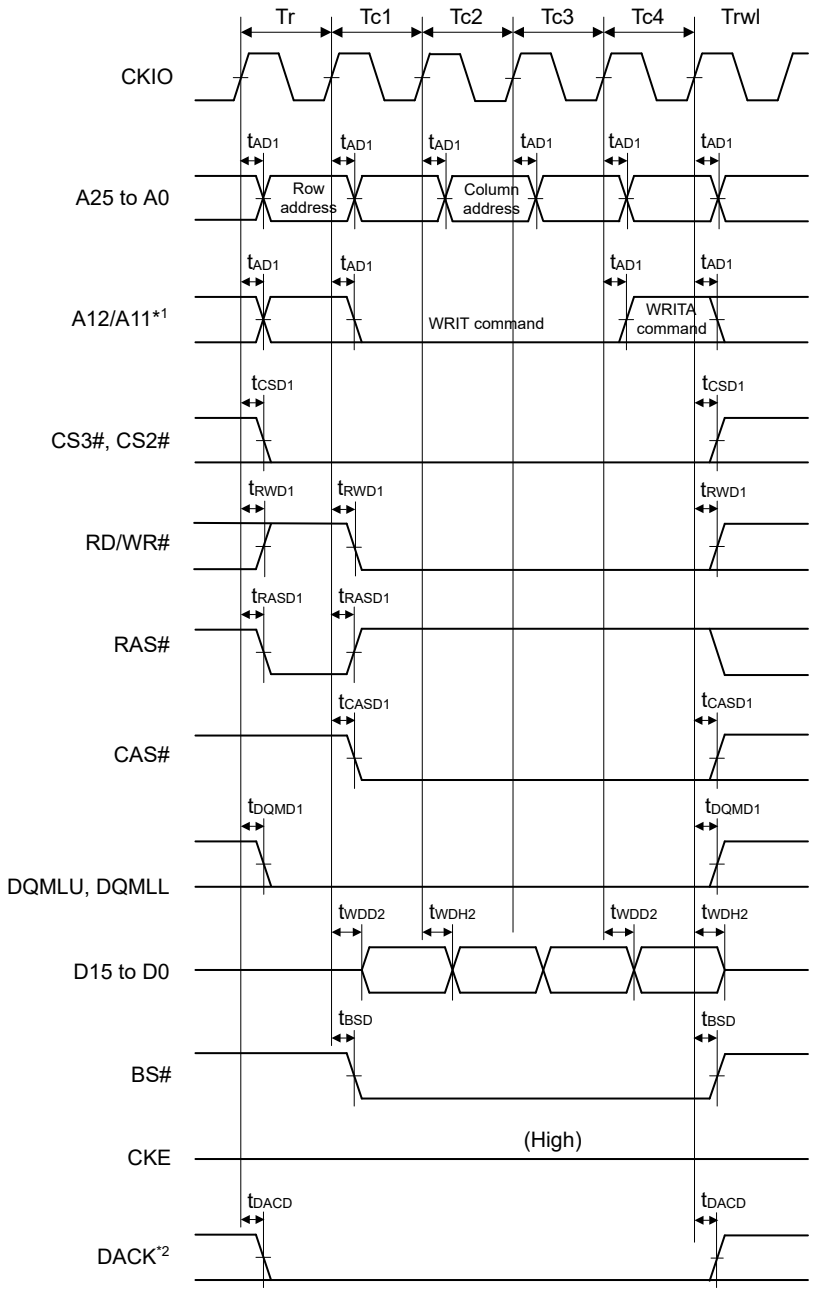
Note 1. Address pin to be connected to A10 of SDRAM.
Note 2. DACK is waveform when active-low is specified.

Figure 52.23 Synchronous DRAM single-write bus cycle (with auto precharge, TRWL = 1 cycle)



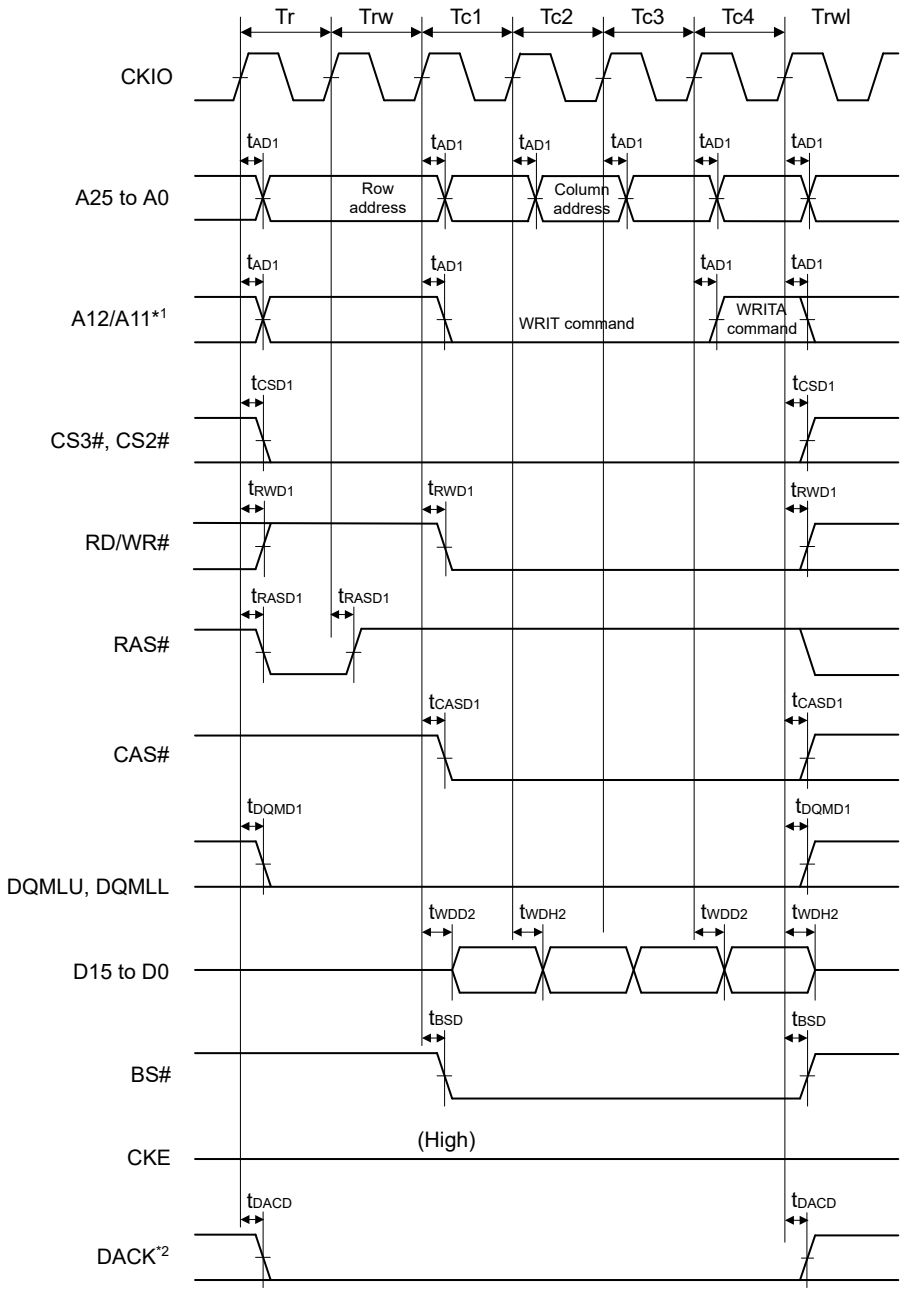
Note 1. Address pin to be connected to A10 of SDRAM.
Note 2. DACK is waveform when active-low is specified.

Figure 52.24 Synchronous DRAM single-write bus cycle (with auto precharge, WTRCD = 2 cycles, TRWL = 1 cycle)



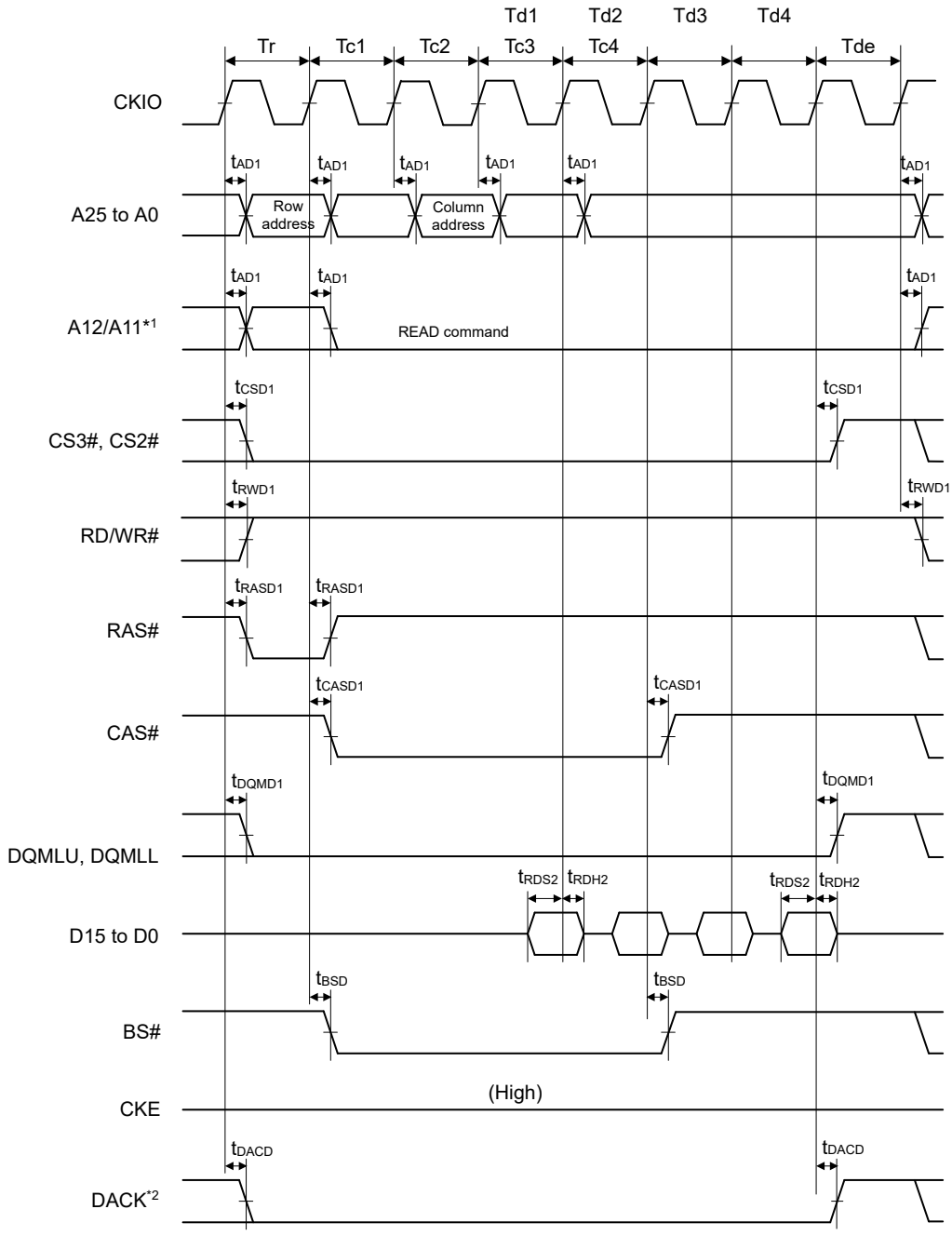
Note 1. Address pin to be connected to A10 of SDRAM.
Note 2. DACK is waveform when active-low is specified.

Figure 52.25 Synchronous DRAM burst-write bus cycle (write for 4 cycles) (with auto precharge, WTRCD = 0 cycles, TRWL = 1 cycle)



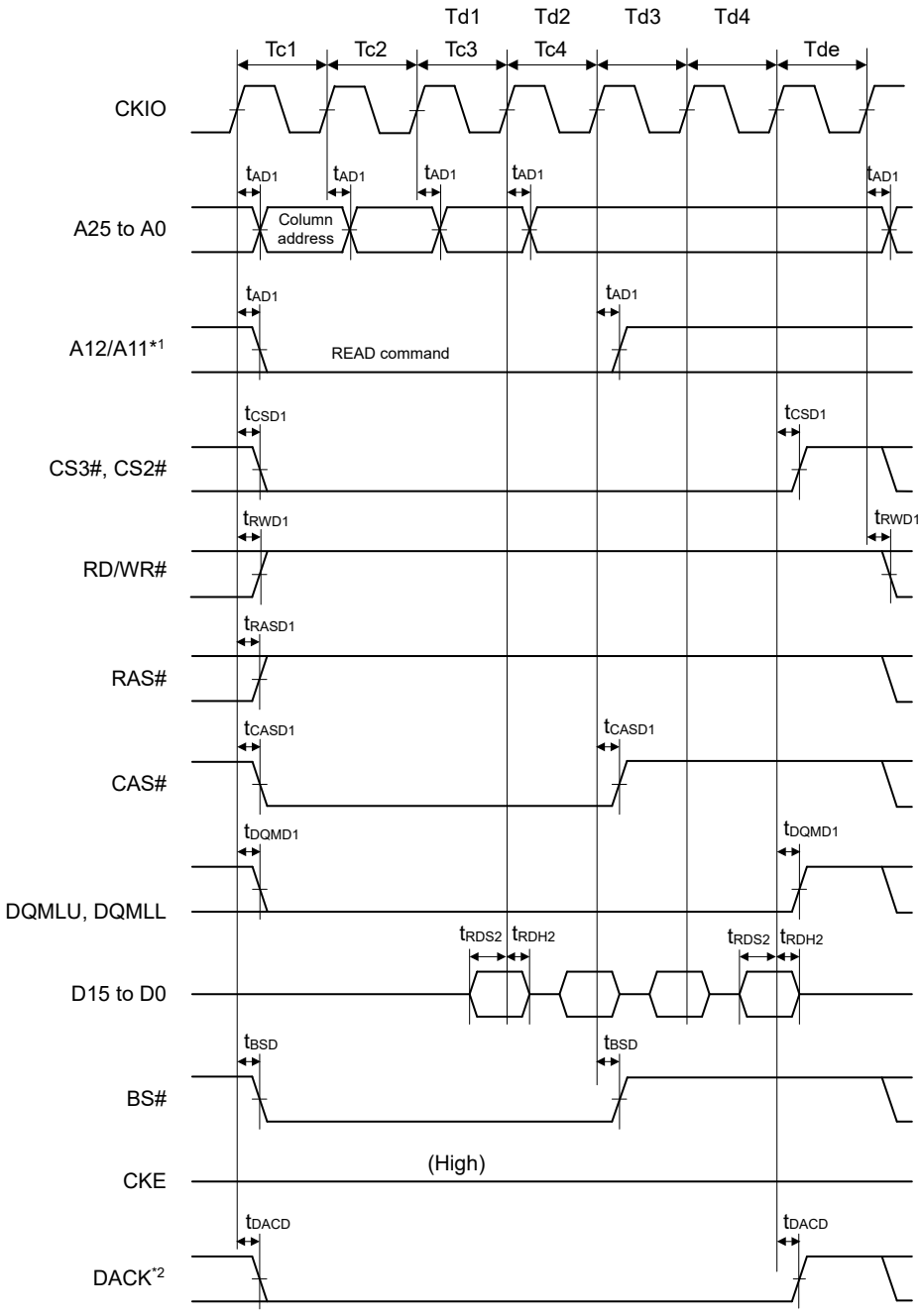
Note 1. Address pin to be connected to A10 of SDRAM.
Note 2. DACK is waveform when active-low is specified.

Figure 52.26 Synchronous DRAM burst-write bus cycle (write for 4 cycles) (with auto precharge, WTRCD = 1 cycle, TRWL = 1 cycle)



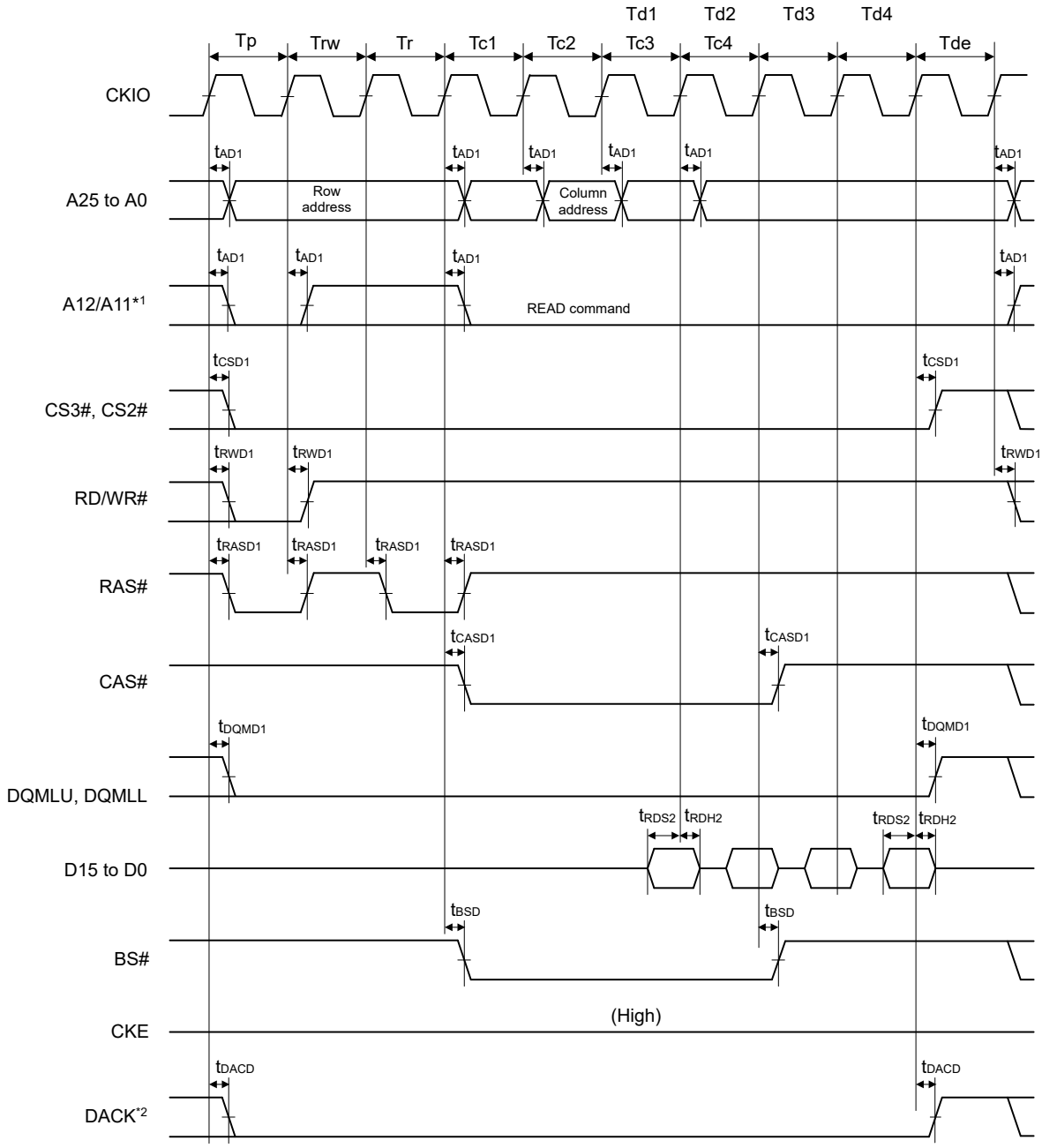
Note 1. Address pin to be connected to A10 of SDRAM.
Note 2. DACK is waveform when active-low is specified.

Figure 52.27 Synchronous DRAM burst-read bus cycle (read for 4 cycles) (bank active mode: ACT + READ command, CAS latency 2, WTRCD = 0 cycles)



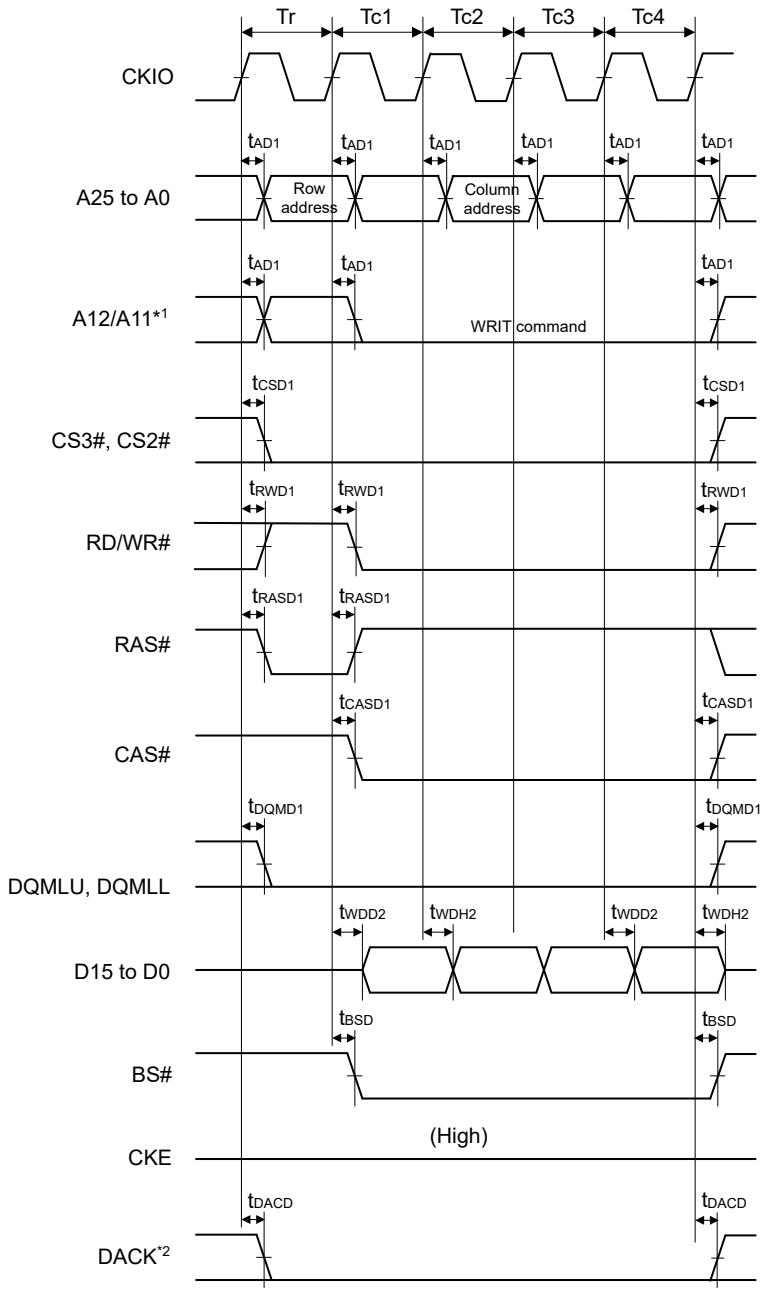
Note 1. Address pin to be connected to A10 of SDRAM.
Note 2. DACK is waveform when active-low is specified.

Figure 52.28 Synchronous DRAM burst-read bus cycle (read for 4 cycles) (bank active mode: READ command, same row address, CAS latency 2, WTRCD = 0 cycles)



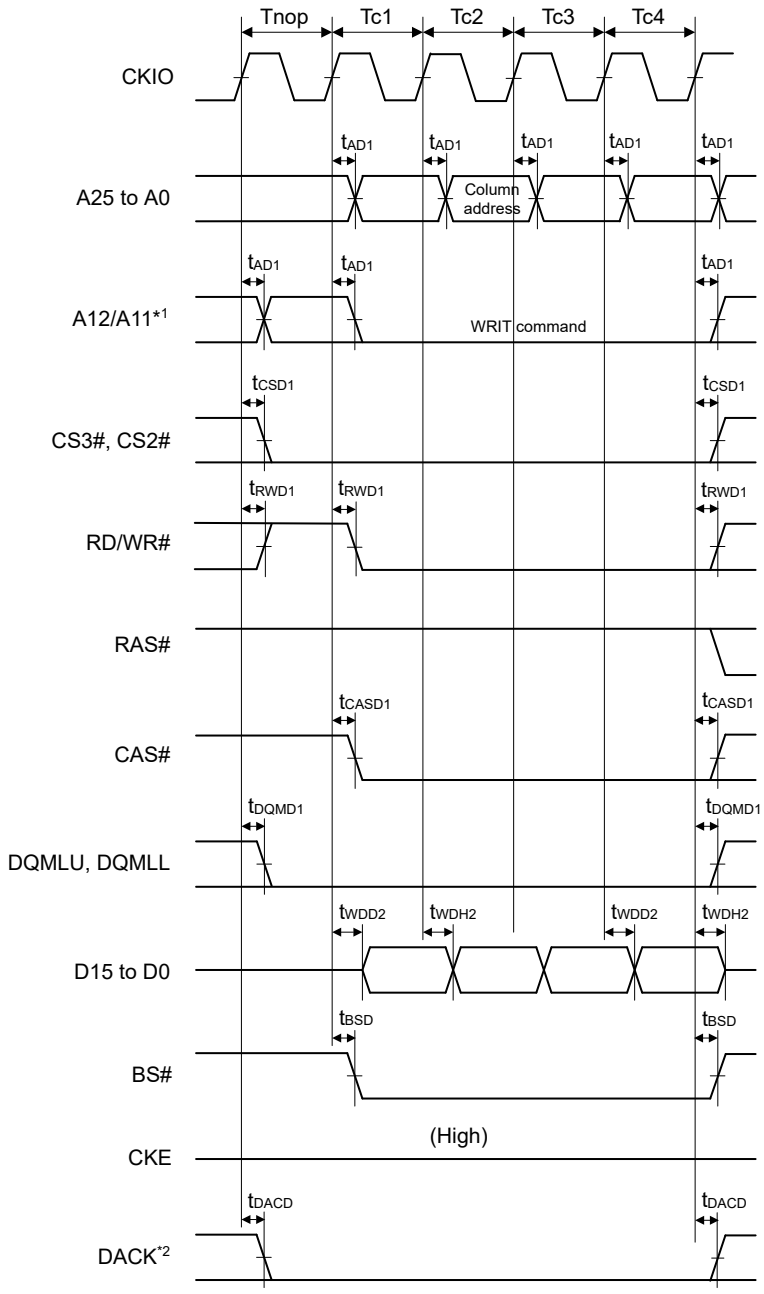
Note 1. Address pin to be connected to A10 of SDRAM.
Note 2. DACK is waveform when active-low is specified.

Figure 52.29 Synchronous DRAM burst-read bus Cycle (read for 4 cycles) (bank active mode: PRE + ACT + READ command, different row address, CAS latency 2, WTRCD = 0 cycles)



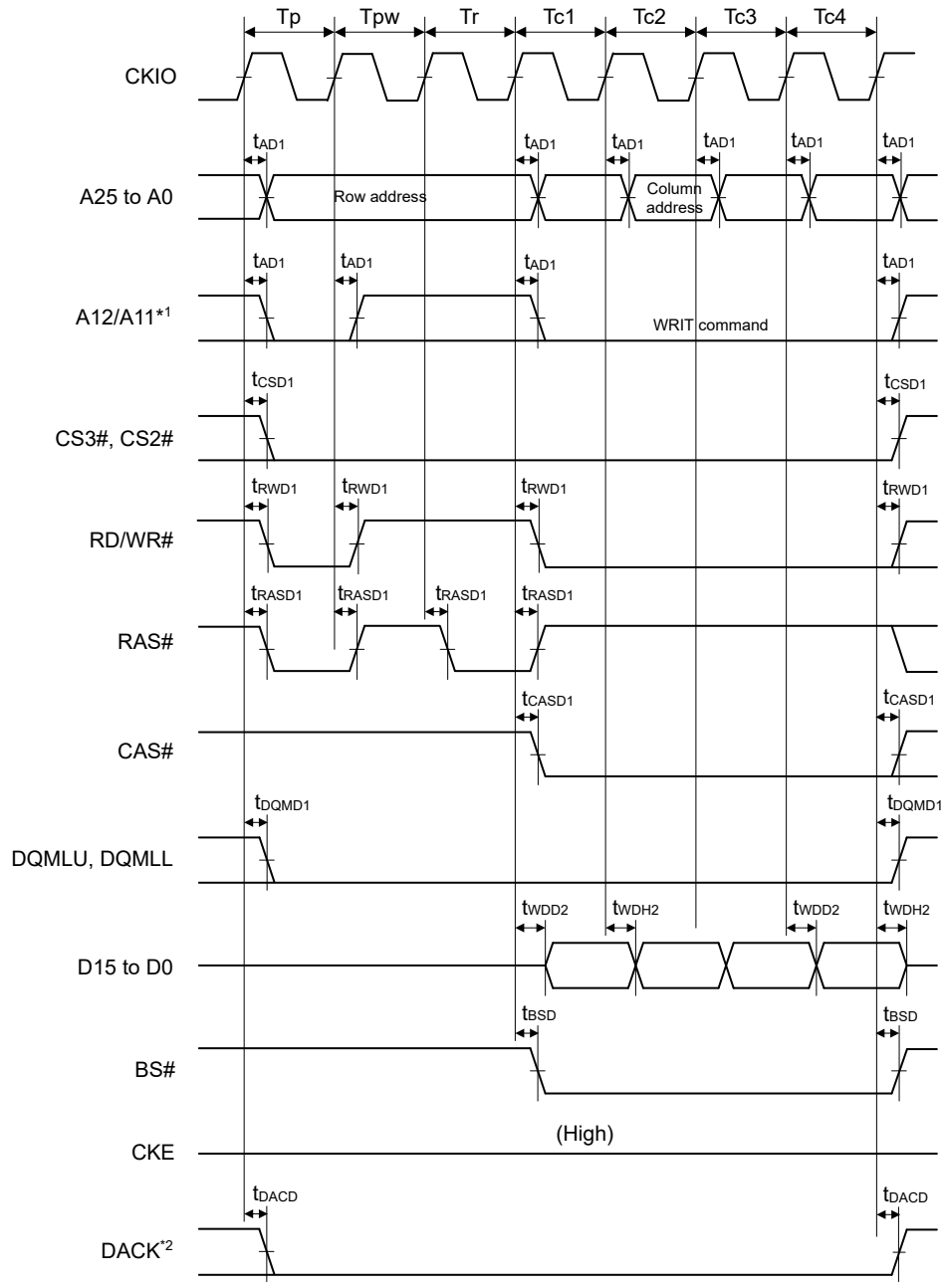
Note 1. Address pin to be connected to A10 of SDRAM.
Note 2. DACK is waveform when active-low is specified.

Figure 52.30 Synchronous DRAM burst-write bus cycle (write for 4 cycles) (bank active mode: ACT + WRITE command, WTRCD = 0 cycles, TRWL = 0 cycles)



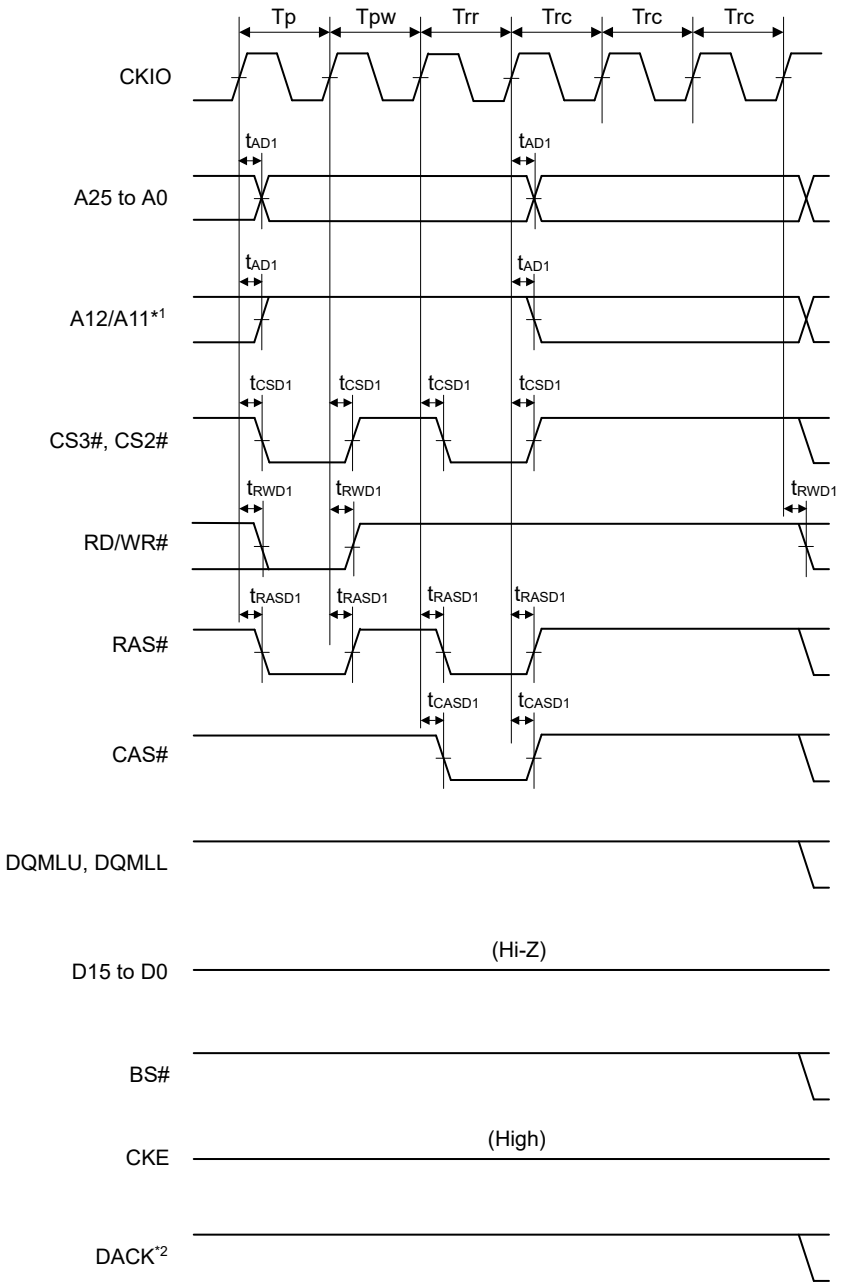
Note 1. Address pin to be connected to A10 of SDRAM.
Note 2. DACK is waveform when active-low is specified.

Figure 52.31 Synchronous DRAM burst-write bus cycle (write for 4 cycles) (bank active mode: WRITE command, same row address, WTRCD = 0 cycles, TRWL = 0 cycles)



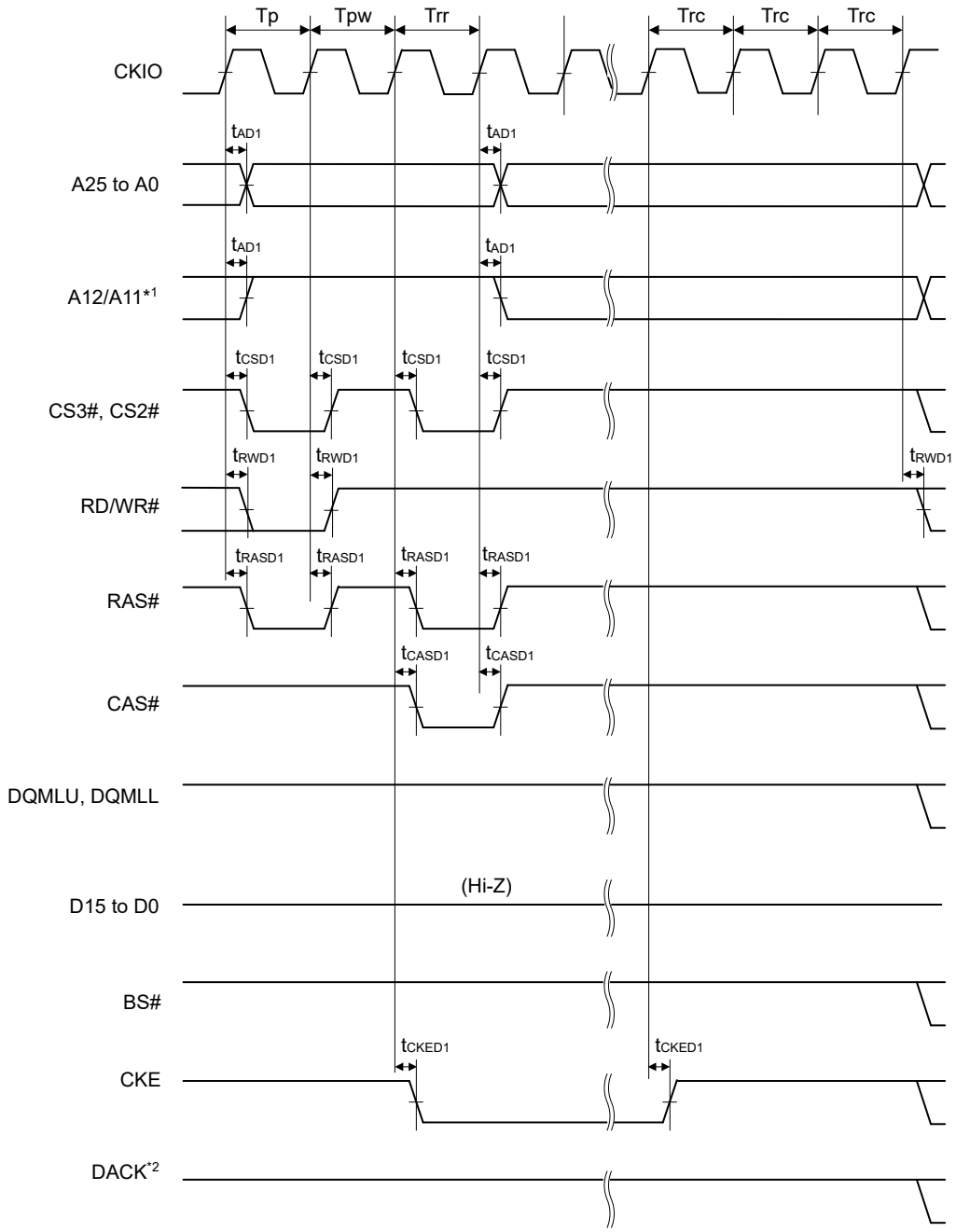
Note 1. Address pin to be connected to A10 of SDRAM.
Note 2. DACK is waveform when active-low is specified.

Figure 52.32 Synchronous DRAM burst-write bus cycle (write for 4 cycles) (bank active mode: PRE + ACT + WRITE command, different row address, WTRCD = 0 cycles, TRWL = 0 cycles)



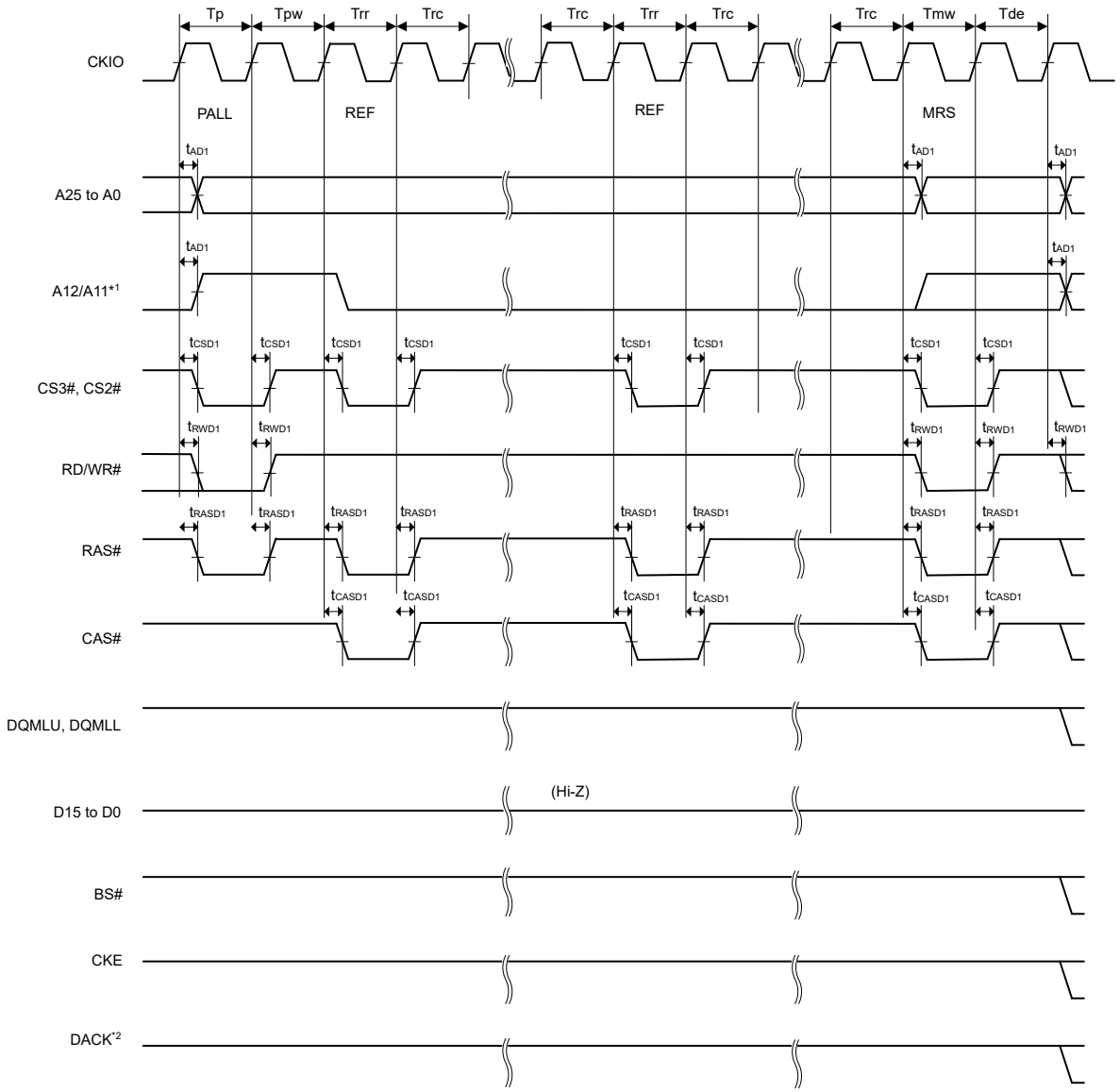
Note 1. Address pin to be connected to A10 of SDRAM.
Note 2. DACK is waveform when active-low is specified.

Figure 52.33 Synchronous DRAM auto-refresh timing (WTRP = 1 cycle, WTRC = 3 cycles)



Note 1. Address pin to be connected to A10 of SDRAM.
Note 2. DACK is waveform when active-low is specified.

Figure 52.34 Synchronous DRAM self-refresh timing (WTRP = 1 cycle)



Note 1. Address pin to be connected to A10 of SDRAM.
Note 2. DACK is waveform when active-low is specified.

Figure 52.35 Synchronous DRAM mode register set timing (WTRP = 1 cycle)

52.5.4 DMAC Timing

Table 52.23 DMAC timing

Conditions: V_{OH} = VCC33 × 0.5, V_{OL} = VCC33 × 0.5, C = 15 pF (CKIO), 30 pF (others), $T_j$min = −40°C

Parameter		Symbol	Min.*1	Max.	Unit	Reference figure
DMAC	DREQ pulse width	t_{DRQW}	t_{PLcyc} × 2	—	ns	Figure 52.36
	DACK and TEND delay time	t_{DACD}	0	10	ns	Figure 52.37

Note 1. t_{PLcyc}: PCLKL cycle

Table 52.24 DMAC timing

Conditions: V_{OH} = VCC33 × 0.5, V_{OL} = VCC33 × 0.5, C = 12 pF (CKIO), 12 pF (others), $T_j$min = −20°C

Parameter		Symbol	Min.*1	Max.	Unit	Reference figure
DMAC	DREQ pulse width	t_{DRQW}	t_{PLcyc} × 2	—	ns	Figure 52.36
	DACK and TEND delay time	t_{DACD}	-0.5	8	ns	Figure 52.37

Note 1. t_{PLcyc}: PCLKL cycle

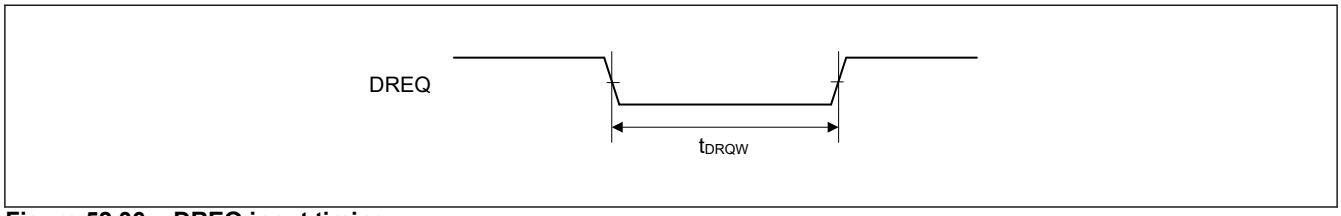


Figure 52.36 DREQ input timing

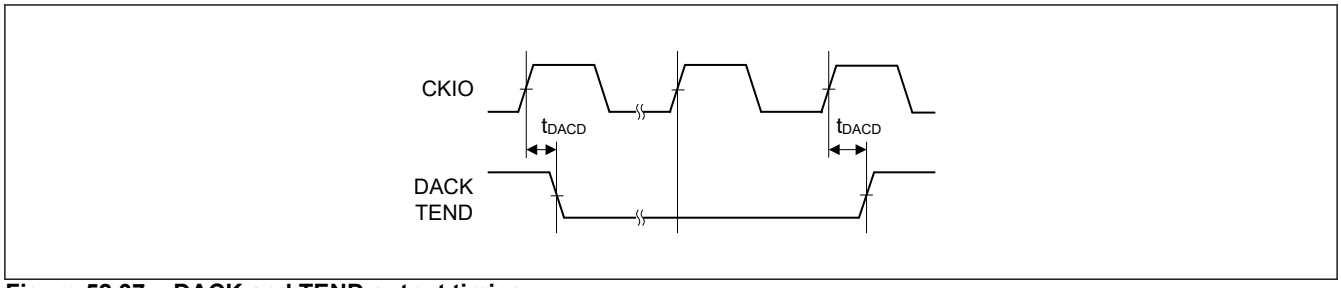


Figure 52.37 DACK and TEND output timing

52.5.5 On-Chip Peripheral Module Timing

52.5.5.1 I/O Port Timing

Table 52.25 I/O port timing

Parameter		Symbol	Min.	Max.	Unit*1	Reference figure
I/O port	Input data pulse width	t_{PRW}	1.5	—	t_{PLcyc}	Figure 52.38

Note 1. t_{PLcyc}: PCLKL cycle

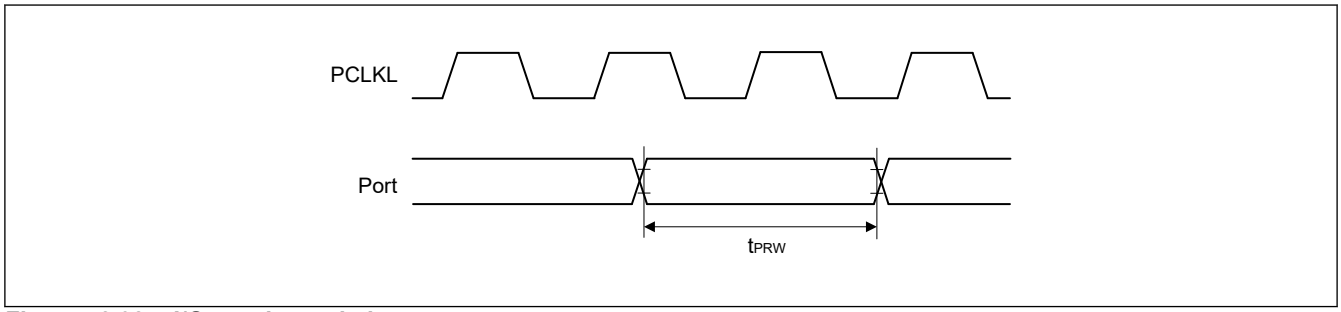


Figure 52.38 I/O port input timing

52.5.5.2 CMTW Timing

Table 52.26 CMTW timing

Parameter			Symbol	Min.	Max.	Unit*1	Reference figure
CMTW	Input capture input pulse width	Single-edge setting	$t_{CMTWICW}$	1.5	—	t_{PLcyc}	Figure 52.39
		Both-edge setting		2.5	—		

Note 1. t_{PLcyc}: PCLKL cycle

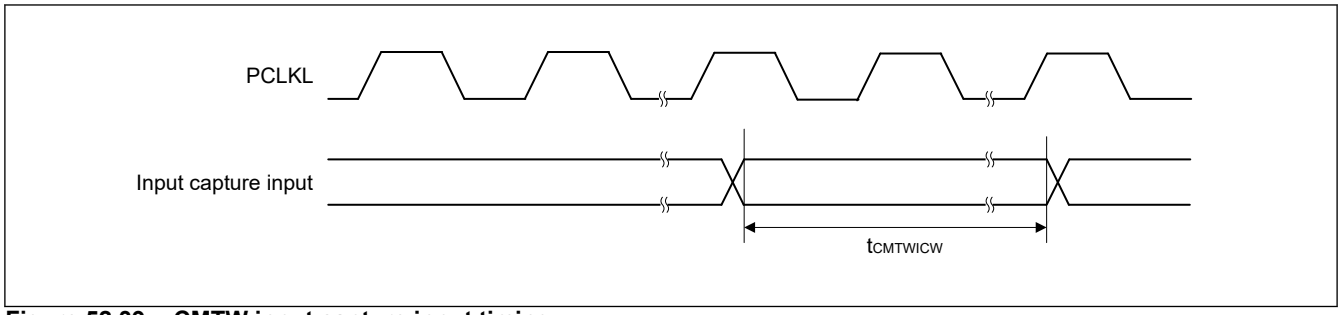


Figure 52.39 CMTW input capture input timing

52.5.5.3 MTU3 Timing

Table 52.27 MTU3 timing

Parameter			Symbol	Min.	Max.	Unit*1	Reference figure
MTU3	Input capture input pulse width	Single-edge setting	t_{MTICW}	2.5	—	t_{PHcyc}	Figure 52.40
		Both-edge setting		3.5	—		
	Timer clock pulse width	Single-edge setting	t_{MTCKWH}, t_{MTCKWL}	2.5	—	t_{PHcyc}	Figure 52.41
		Both-edge setting		3.5	—		
		Phase counting mode		3.5	—		

Note 1. t_{PHcyc}: PCLKH cycle

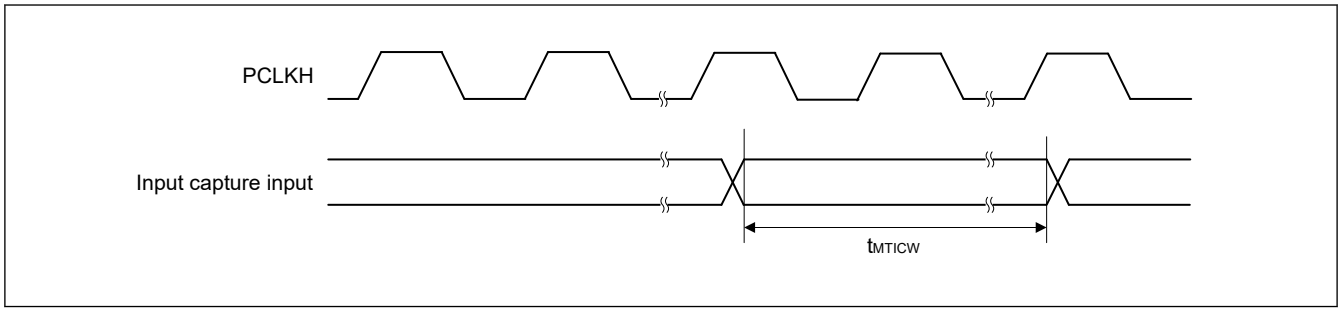


Figure 52.40 MTU3 input capture input timing

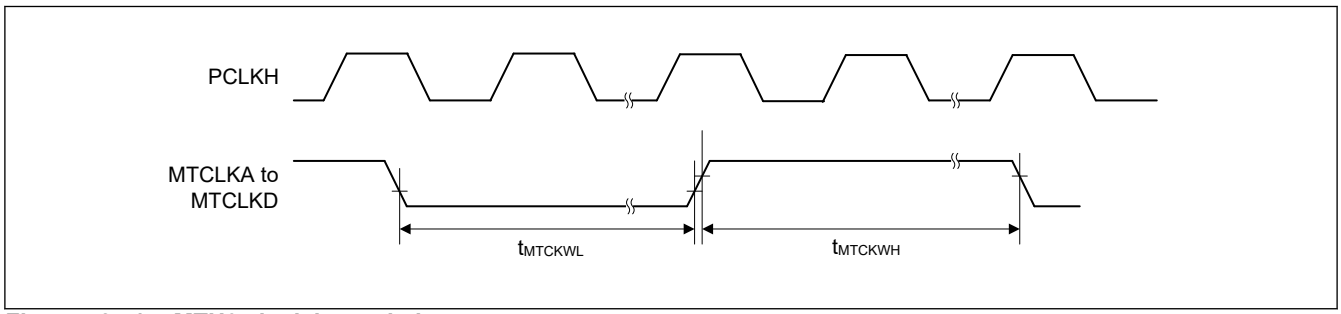


Figure 52.41 MTU3 clock input timing

52.5.5.4 POE3 Timing

Table 52.28 POE3 timing

Parameter			Symbol	Min.	Max.	Unit*1	Reference figure
POE3	POEn# input pulse width		t_{POEW}	2.5	—	t_{PHcyc}	Figure 52.42
	Output disable time	Transition of the POEn# signal level	t_{POEDI}	—	5 × PCLKH + 0.1	μs	Figure 52.43
		Simultaneous conduction of output pins	t_{POEDO}	—	3 × PCLKH + 0.1	μs	Figure 52.44
		Register setting	t_{POEDS}	—	PCLKH + 0.1	μs	Figure 52.45
		Oscillation stop detection	t_{POEDOS}	—	74	μs	Figure 52.46

Note 1. t_{PHcyc}: PCLKH cycle

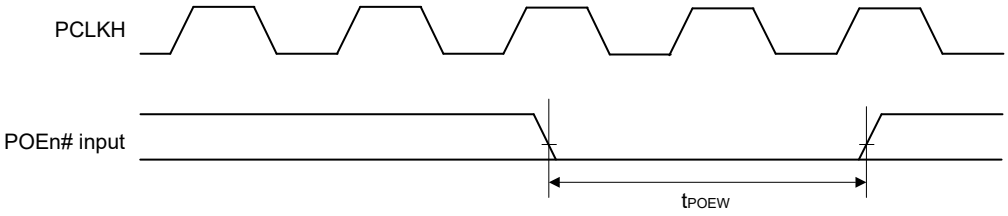


Figure 52.42 POEn# input pulse timing

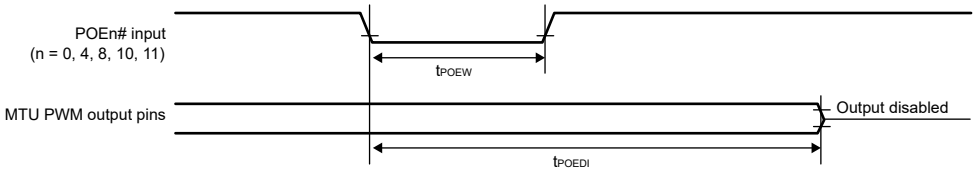
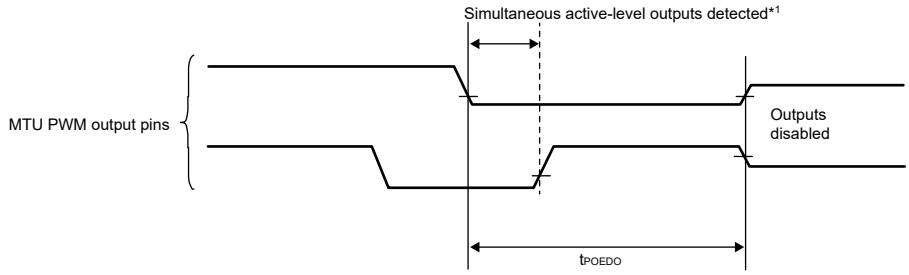


Figure 52.43 Output disable time for POE in response to transition of the POEn# signal level



Note 1. When the active level is set to low.

Figure 52.44 Output disable time for POE in response to the simultaneous conduction of output pins

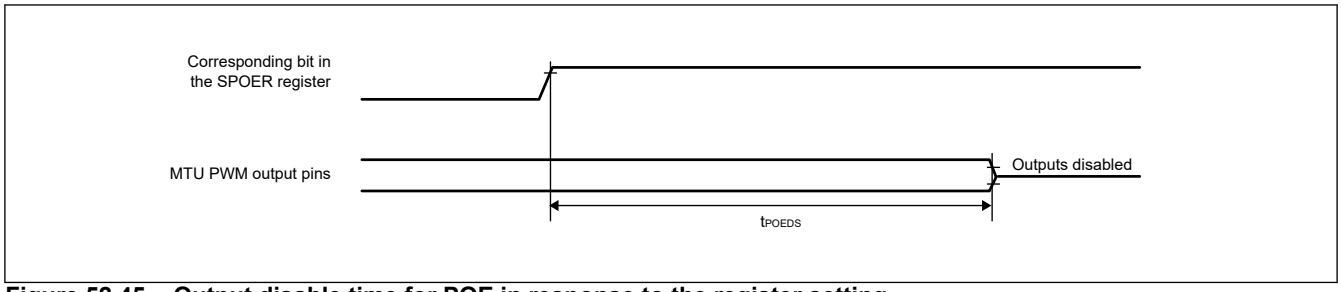


Figure 52.45 Output disable time for POE in response to the register setting

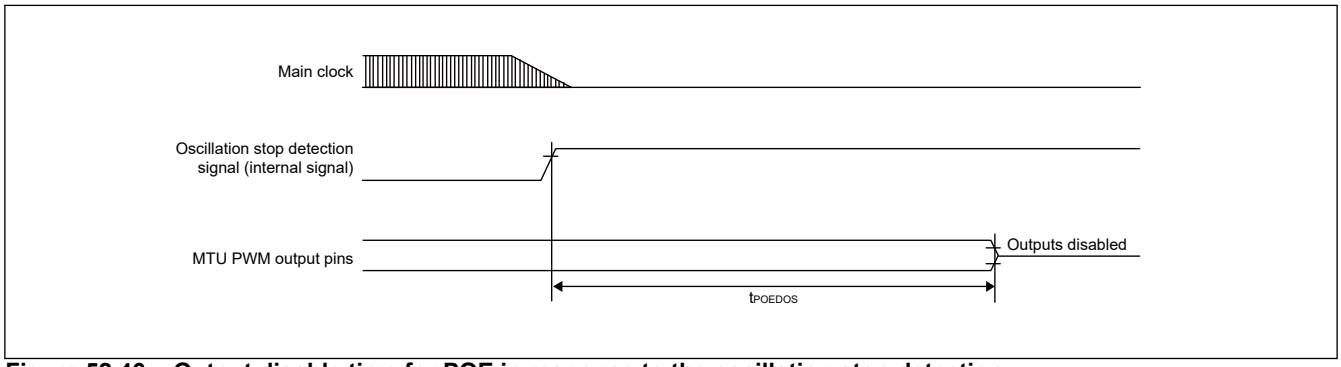


Figure 52.46 Output disable time for POE in response to the oscillation stop detection

52.5.5.5 GPT Timing

Table 52.29 GPT timing

Parameter			Symbol	Min.	Max.	Unit*1	Reference figure
GPT	Input capture input pulse width	Single-edge setting	t_{GTICW}	2.5	—	t_{PHcyc}	Figure 52.47
		Both-edge setting		3.5	—		
	External trigger input pulse width	Single-edge setting	t_{GTEW}	2.5	—	t_{PHcyc}	Figure 52.48
		Both-edge setting		3.5	—		

Note 1. t_{PHcyc}: PCLKH cycle (LLPP channels), PCLKM cycle (Other channels)

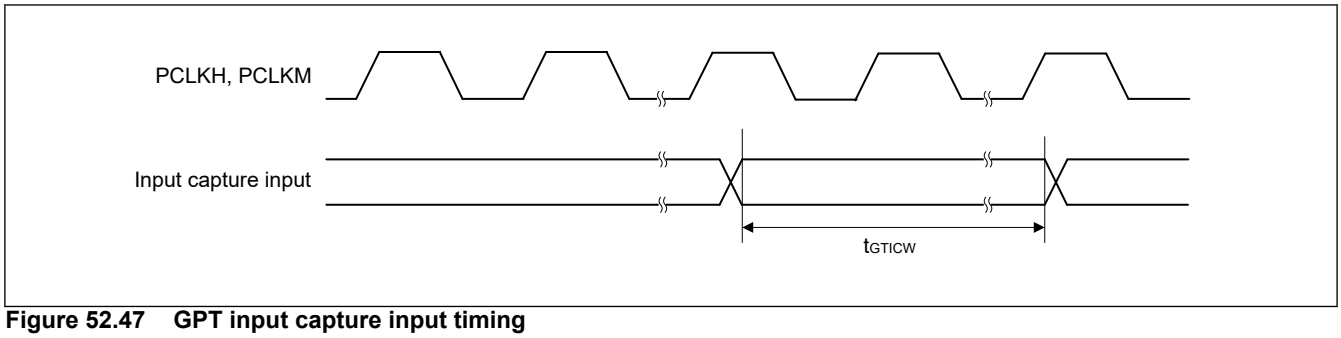


Figure 52.47 GPT input capture input timing

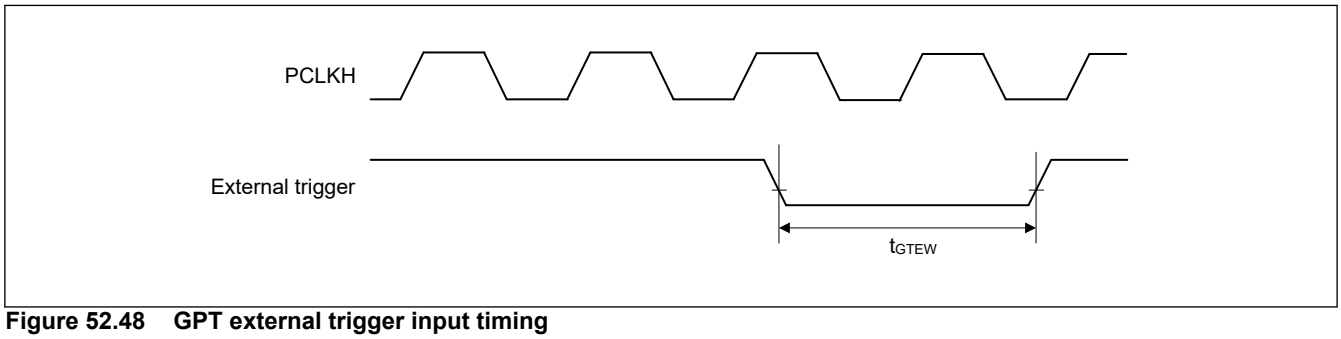


Figure 52.48 GPT external trigger input timing

52.5.5.6 POEG Timing

Table 52.30 POEG timing

Parameter			Symbol	Min.	Max.	Unit*1	Reference figure
POEG	GTETRGn input pulse width (n = A to D)		t_{POEGW}	2.5	—	t_{PHcyc}	Figure 52.49
	Output disable time	Input level detection of the GTETRGn pin (via flag)	t_{POEGDI}	—	3 × PCLKH + 0.1	μs	Figure 52.50
		Detection of the output stopping signal from GPT (dead time error, simultaneous high output, or simultaneous low output)	t_{POEGDO}	—	0.1	μs	Figure 52.51
		Register setting	t_{POEGDS}	—	PCLKH + 0.1	μs	Figure 52.52
		Oscillation stop detection	$t_{POEGDOS}$	—	74	μs	Figure 52.53

Note 1. t_{PHcyc}: PCLKH cycle (LLPP channels), PCLKL cycle (Other channels)

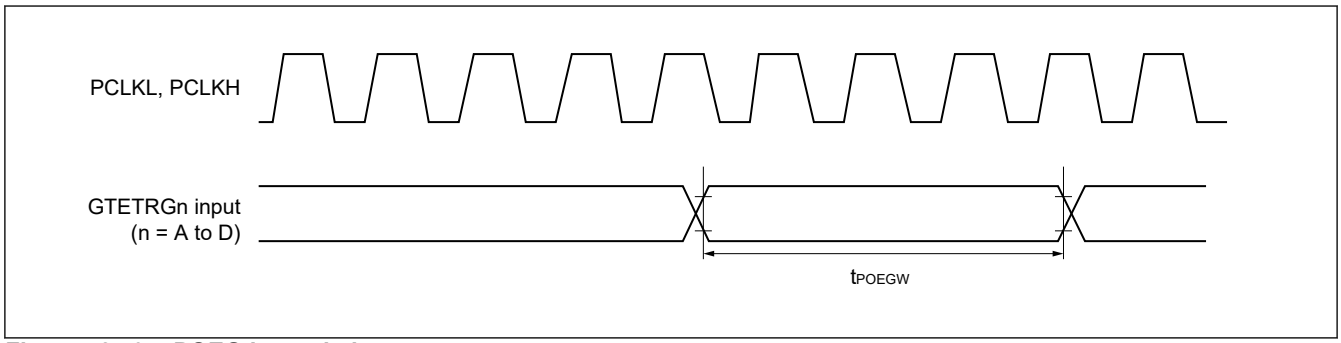


Figure 52.49 POEG input timing

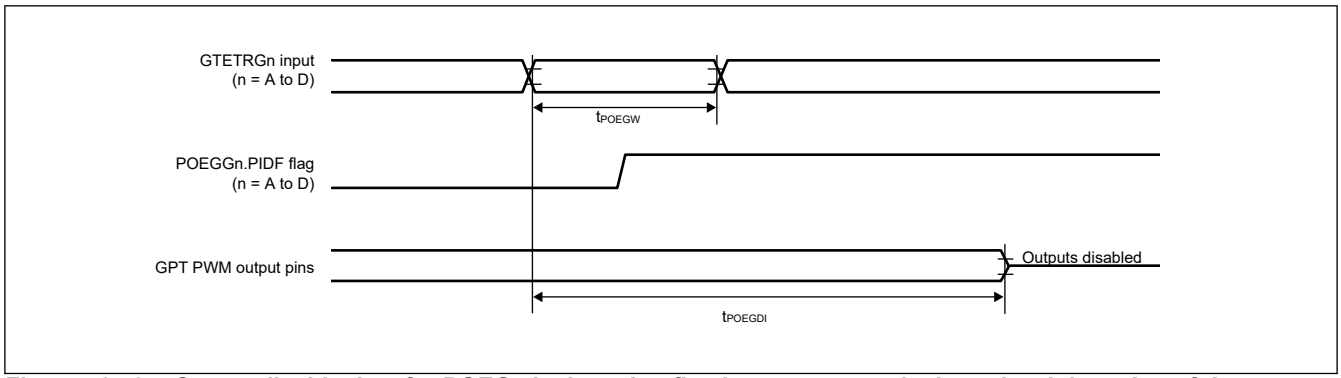
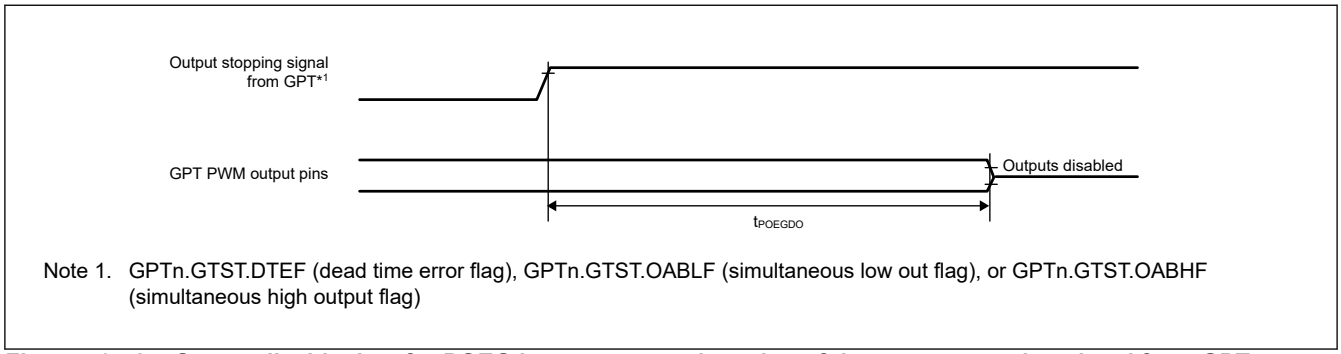


Figure 52.50 Output disable time for POEG via detection flag in response to the input level detection of the GTETRGn pin



Note 1. GPTn.GTST.DTEF (dead time error flag), GPTn.GTST.OABLF (simultaneous low out flag), or GPTn.GTST.OABHF (simultaneous high output flag)

Figure 52.51 Output disable time for POEG in response to detection of the output stopping signal from GPT

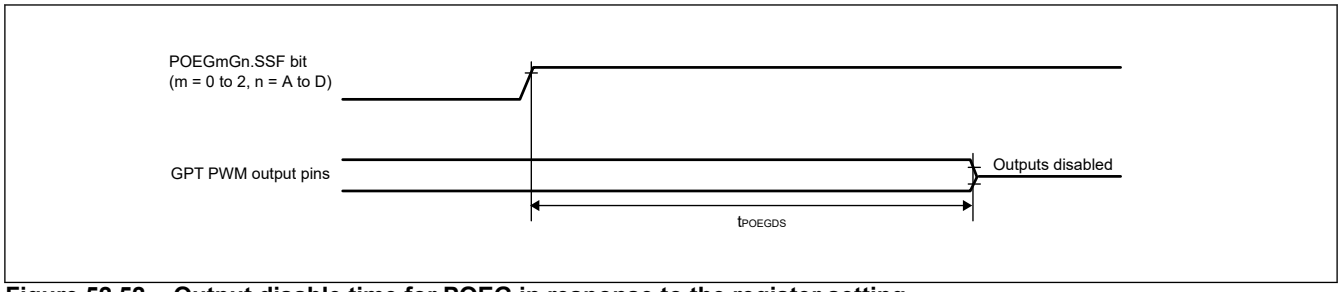


Figure 52.52 Output disable time for POEG in response to the register setting

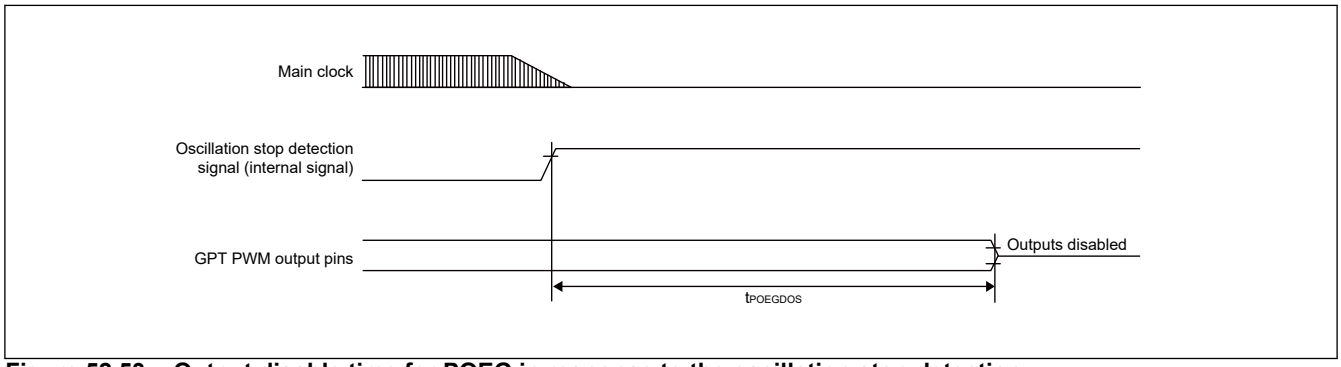


Figure 52.53 Output disable time for POEG in response to the oscillation stop detection

52.5.5.7 A/D Converter Trigger Timing

Table 52.31 A/D converter trigger timing

Parameter			Symbol	Min.	Max.	Unit*1	Reference figure
A/D converter	A/D converter trigger input pulse width	ADTRG0#, ADTRG1#	t_{TRGW}	1.5	—	$t_{PADCcyc}$	Figure 52.54

Note 1. $t_{PADCcyc}$: PCLKADC cycle

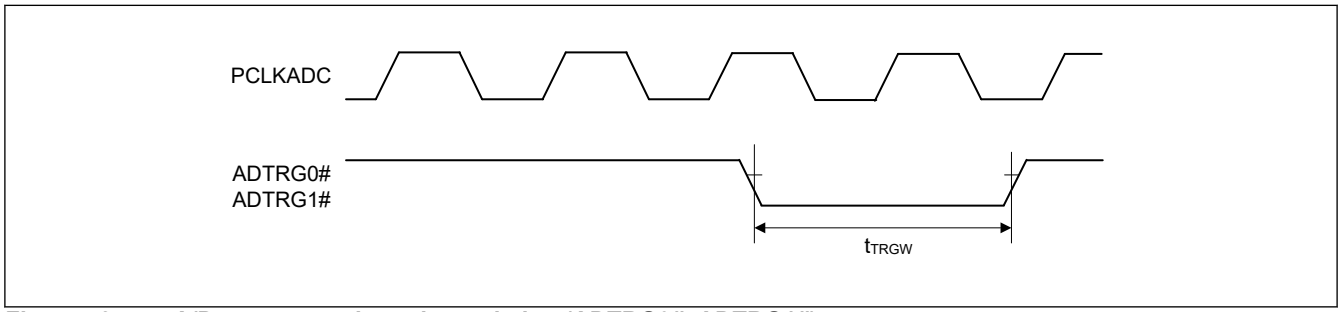


Figure 52.54 A/D converter trigger input timing (ADTRG0#, ADTRG1#)

52.5.5.8 SCI Timing

Conditions: V_{OH} = VCC33 × 0.5, V_{OL} = VCC33 × 0.5, C = 30 pF (except Simple I2C)

Table 52.32 SCI timing (1 of 2)

Parameter		Symbol	Min.	Max.	Unit	Reference figure
SCI (Asynchronous)	Input clock cycle	t_{Scyc}	4	—	$t_{PSCIcyc}$	Figure 52.55
	Input clock pulse width	t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time	t_{SCKr}	—	3	ns	
	Input clock fall time	t_{SCKf}	—	3	ns	
	Output clock cycle	t_{Scyc}	6	—	$t_{PSCIcyc}$	
	Output clock pulse width	t_{SCKW}	0.4	0.6	t_{Scyc}	
	Output clock rise time	t_{SCKr}	—	3	ns	
	Output clock fall time	t_{SCKf}	—	3	ns	
SCI (Simple I2C, Standard mode)	SDA input rise time	t_{Sr}	—	1000	ns	Figure 52.56
	SDA input fall time	t_{Sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	$2 \times NF_{cyc}$ *1	ns	
	Data input setup time	t_{SDAS}	250	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	400	pF	
SCI (Simple I2C, Fast mode)	SDA input rise time	t_{Sr}	—	300	ns	Figure 52.56
	SDA input fall time	t_{Sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	$2 \times NF_{cyc}$ *1	ns	
	Data input setup time	t_{SDAS}	100	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	400	pF	

Table 52.32 SCI timing (2 of 2)

Parameter			Symbol	Min.	Max.	Unit	Reference figure
SCI (Clock sync, Simple SPI)	SCK output clock cycle (master)		t_{SPcyc}	2	65536	$t_{PSCIcyc}$	Figure 52.57 to Figure 52.62
	SCK input clock cycle (slave)			2	65536		
	SCK clock high pulse width		t_{SPCKWH}	0.4	0.6	t_{SPcyc}	
	SCK clock low pulse width		t_{SPCKWL}	0.4	0.6	t_{SPcyc}	
	SCK clock rise/fall time		t_{SPCKR}, t_{SPCKF}	—	3	ns	
	Data input setup time	Internal clock	t_{SU}	7	—	ns	
		External clock		3	—		
	Data input hold time	Internal clock	t_H	3	—	ns	
		External clock		3	—		
	Data output delay time	Internal clock	t_{OD}	—	3	ns	
		External clock		—	12		
	Data output hold time	Internal clock	t_{OH}	0	—	ns	
		External clock		0	—		
	Data rise/fall time		t_{DR}, t_{DF}	—	3	ns	
	Slave access time	Internal clock	t_{SA}	—	$3 \times t_{PSCIcyc} + 12$	ns	
		External clock		—	$3 \times t_{PSCIcyc} + 12$		
	Slave output release time	Internal clock	t_{REL}	—	$3 \times t_{PSCIcyc} + 12$	ns	
		External clock		—	$3 \times t_{PSCIcyc} + 12$		
SCI (Simple SPI)	SS input setup time		t_{LEAD}	1	—	t_{SPcyc}	
	SS input hold time		t_{LAG}	1	—	t_{SPcyc}	
	SS input rise/fall time		t_{SSR}, t_{SSF}	—	3	ns	

Note: $t_{PSCIcyc}$: PCLKSCIn cycle

Note 1. $N_{Fcyc} = 4^n \times 2^{m-1} \times t_{PSCIcyc}$
n: CCR2.CKS[1:0] (n = 0, 1, 2, 3)
m: CCR1.NFCS[2:0] (m = 1, 2, 3, 4)

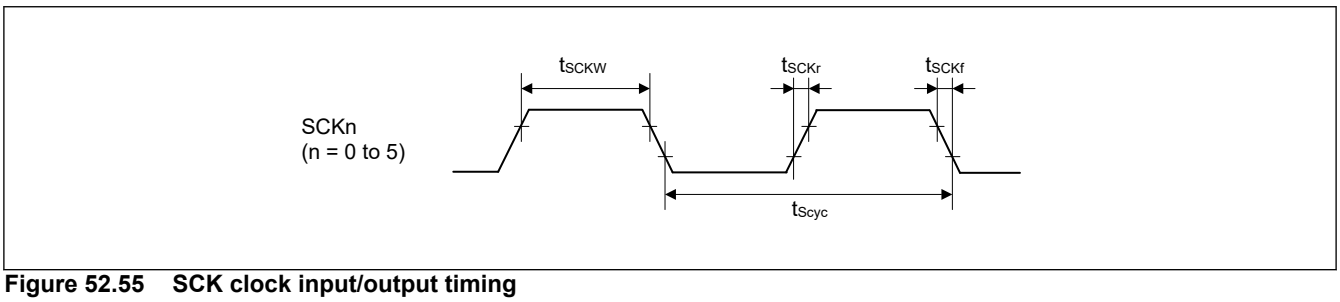
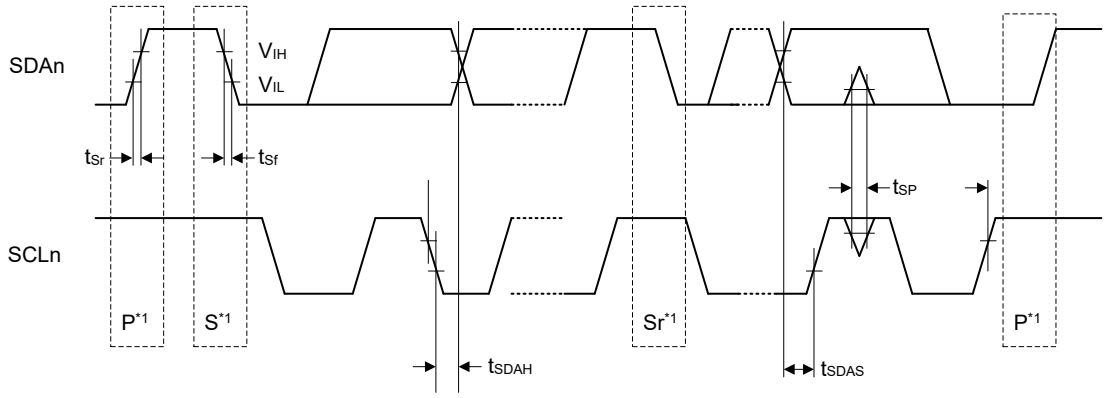


Figure 52.55 SCK clock input/output timing



Note 1. S, P, and Sr indicate the following conditions.
S: Start condition
P: Stop condition
Sr: Restart condition

Figure 52.56 SCI simple I2C mode timing

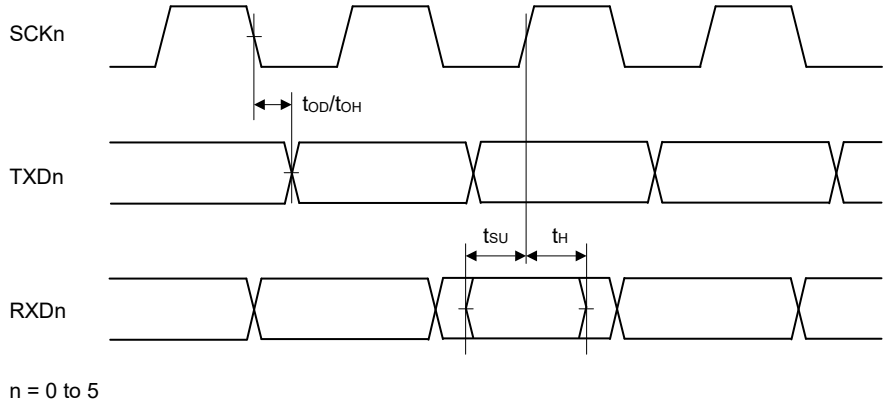
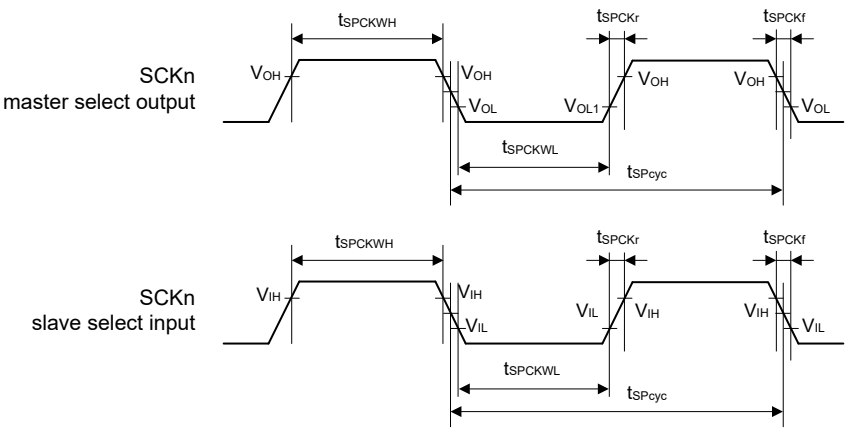


Figure 52.57 SCI input/output timing in clock synchronous mode



VOH = 0.7 × VCC33, VOL = 0.3 × VCC33, VIH = 0.7 × VCC33, VIL = 0.3 × VCC33

Figure 52.58 SCI simple SPI mode clock timing

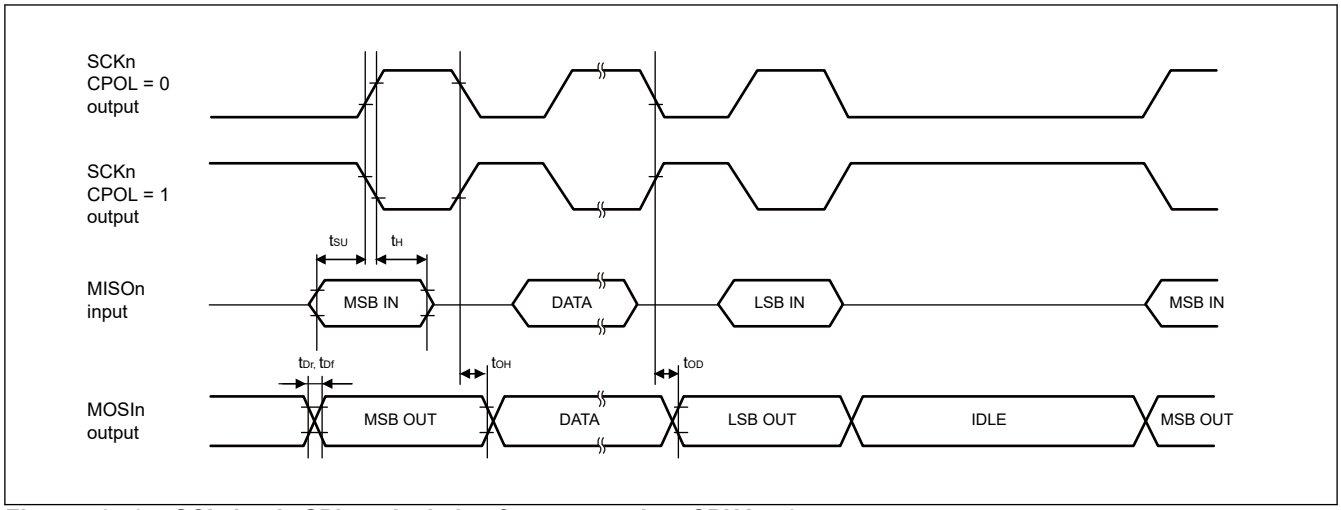


Figure 52.59 SCI simple SPI mode timing for master when CPHA = 0

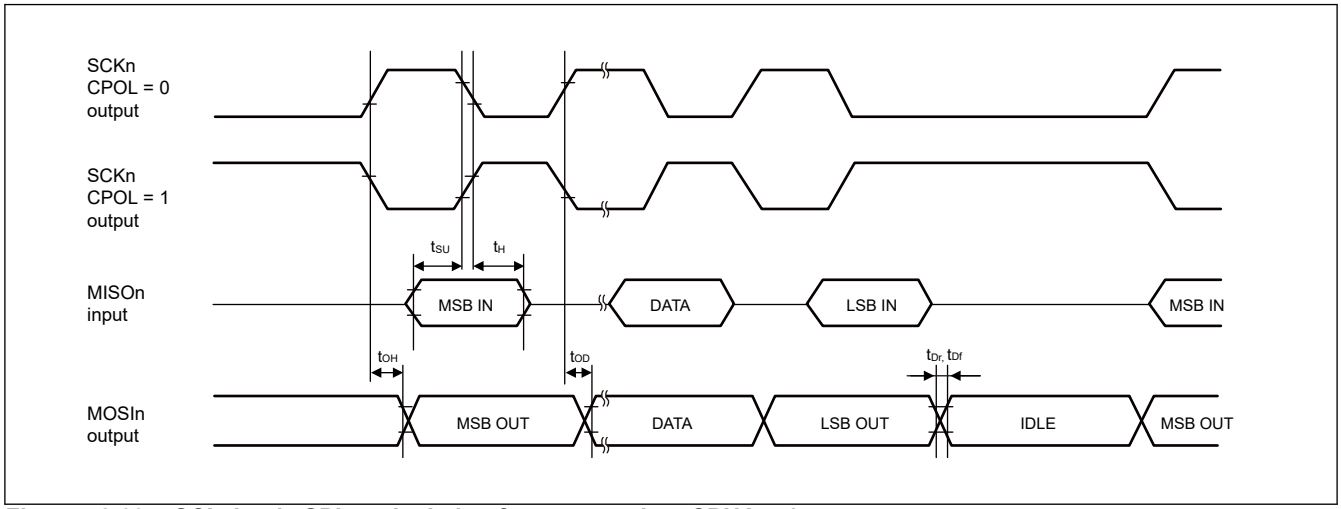


Figure 52.60 SCI simple SPI mode timing for master when CPHA = 1

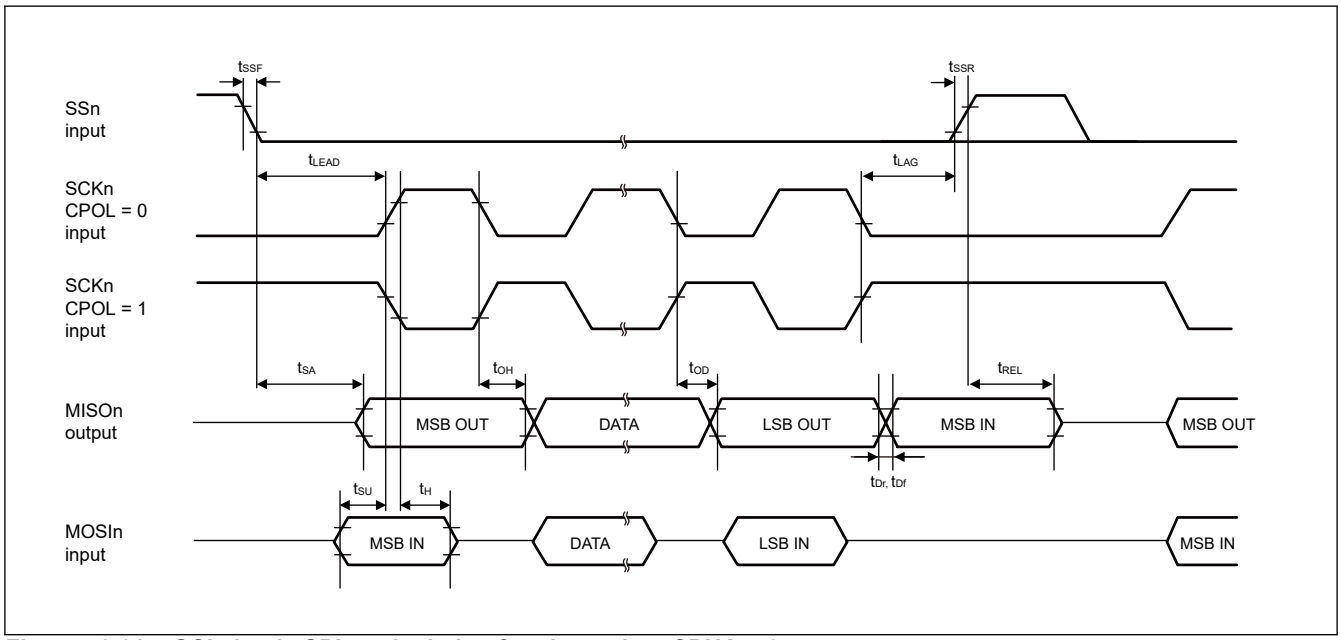


Figure 52.61 SCI simple SPI mode timing for slave when CPHA = 0

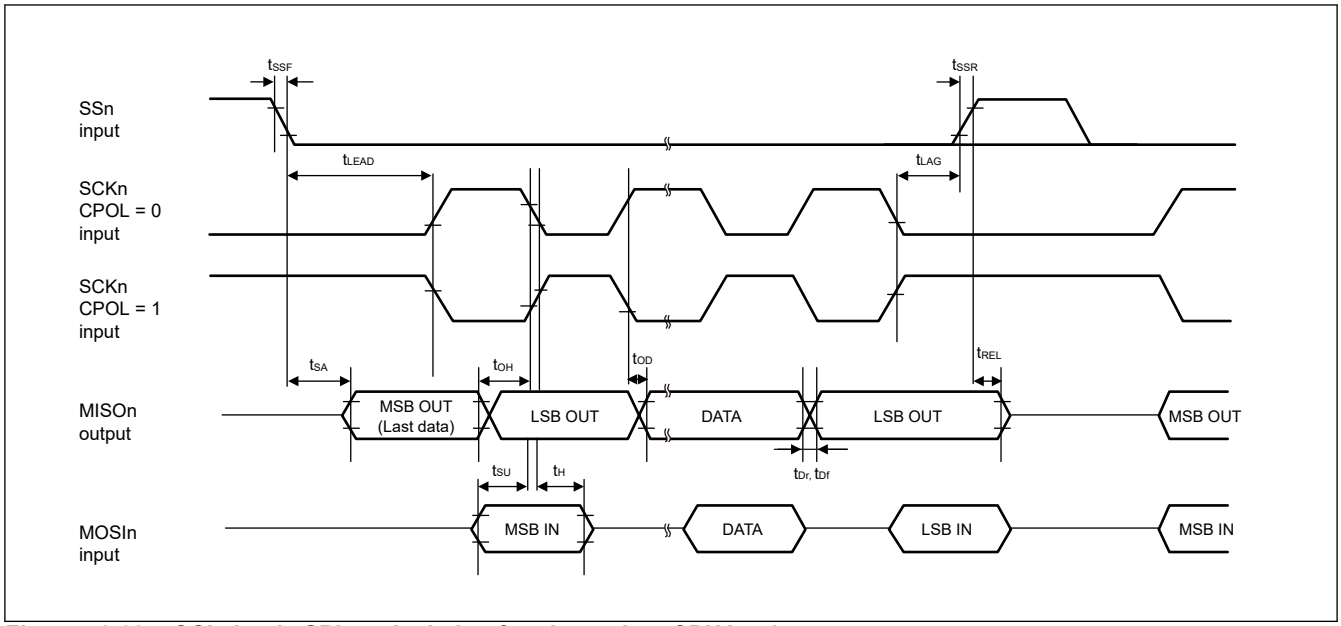


Figure 52.62 SCI simple SPI mode timing for slave when CPHA = 1

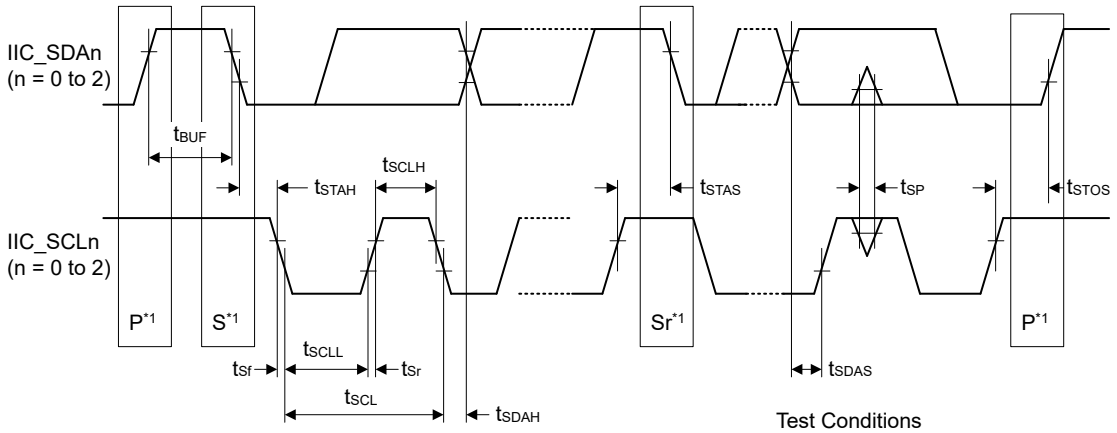
52.5.5.9 IIC Timing

Conditions: V_{OL} = 0.4 V, I_{OL} = 4 mA

Table 52.33 IIC timing

Parameter		Symbol	Min.*1 *2	Max.*1 *2	Unit	Reference figure
IIC (Standard-mode)	SCL input cycle time	t_{SCL}	6(12) × t_{IICcyc} + 1300	—	ns	Figure 52.63
	SCL input high pulse width	t_{SCLH}	3(6) × t_{IICcyc} + 300	—	ns	
	SCL input low pulse width	t_{SCLL}	3(6) × t_{IICcyc} + 300	—	ns	
	SCL, SDA input rising time	t_{sr}	—	1000	ns	
	SCL, SDA input falling time	t_{sf}	—	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1(4) × t_{IICcyc}	ns	
	SDA input bus free time	t_{BUF}	3(6) × t_{IICcyc} + 300	—	ns	
	Start condition input hold time	t_{STAH}	t_{IICcyc} + 300	—	ns	
	Restart condition input setup time	t_{STAS}	1000	—	ns	
	Stop condition input setup time	t_{STOS}	1000	—	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load	C_b	—	400	pF	
IIC (Fast-mode)	SCL input cycle time	t_{SCL}	6(12) × t_{IICcyc} + 600	—	ns	
	SCL input high pulse width	t_{SCLH}	3(6) × t_{IICcyc} + 300	—	ns	
	SCL input low pulse width	t_{SCLL}	3(6) × t_{IICcyc} + 300	—	ns	
	SCL, SDA input rising time	t_{sr}	—*4	300	ns	
	SCL, SDA input falling time	t_{sf}	—*4	300	ns	
	SCL, SDA input spike pulse removal time	t_{SP}	0	1(4) × t_{IICcyc}	ns	
	SDA input bus free time	t_{BUF}	3(6) × t_{IICcyc} + 300	—	ns	
	Start condition input hold time	t_{STAH}	t_{IICcyc} + 300	—	ns	
	Restart condition input setup time	t_{STAS}	300	—	ns	
	Stop condition input setup time	t_{STOS}	300	—	ns	
	Data input setup time	t_{SDAS}	t_{IICcyc} + 50	—	ns	
	Data input hold time	t_{SDAH}	0	—	ns	
	SCL, SDA capacitive load*3	C_b	—	400	pF	

Note 1. t_{IICcyc}: IIC internal reference clock (IICΦ) cycle
Note 2. The value out of parentheses is applicable when the value of the ICMR3.NF[1:0] bits is 00b while the digital filter is enabled by setting ICFER.NFE = 1. The value within parentheses is applicable when the value of the ICMR3.NF[1:0] bits is 11b while the digital filter is enabled by setting ICFER.NFE = 1.
Note 3. Cb is the total capacitance of the bus lines.
Note 4. The minimum values are not specified for t_{sr} and t_{sf} in Fast-mode.



Note 1. S, P, and Sr indicate the following conditions.
S: Start condition
P: Stop condition
Sr: Restart condition

Figure 52.63 IIC bus interface input/output timing

52.5.5.10 CANFD Timing

Table 52.34 CANFD timing

Parameter		Symbol	CAN		CANFD		Unit	Reference figure
			Min.	Max.	Min.	Max.		
CANFD	Internal delay time	t_{node}	—	100	—	50	ns	Figure 52.64
	Transmission rate	—	—	1	—	8	Mbps	

Note: Internal delay time (t_{node}) = Internal transmission delay time (t_{output}) + Internal reception delay time (t_{input})

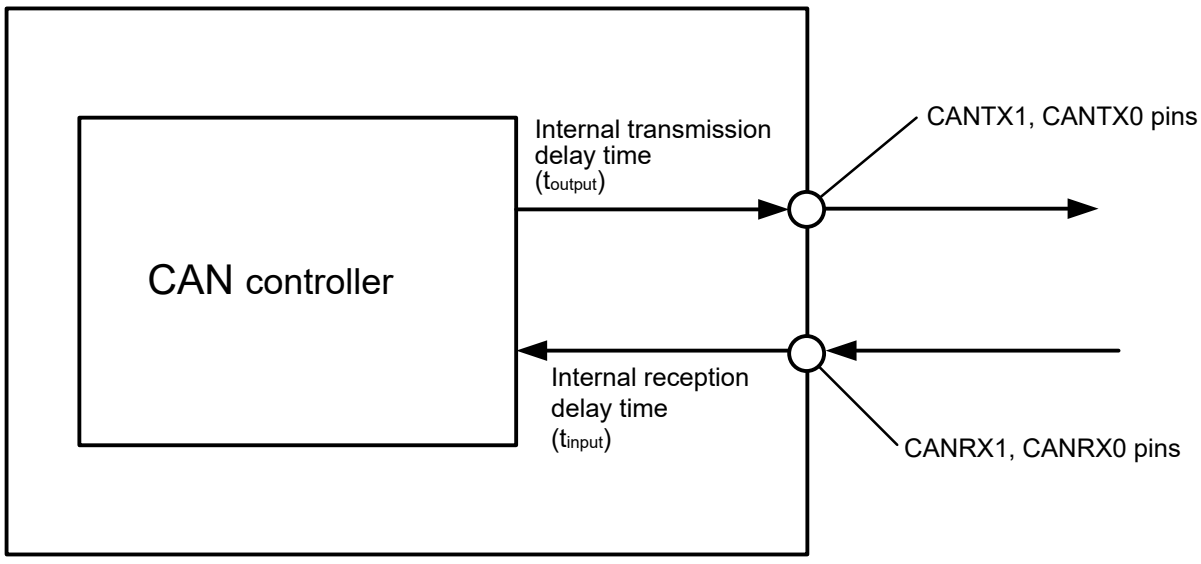


Figure 52.64 CAN interface condition

52.5.5.11 SPI Timing

Table 52.35 SPI timing (1 of 2)

Conditions: V_{OH} = VCC33 × 0.5, V_{OL} = VCC33 × 0.5, C = 30 pF

Parameter		Symbol	Min.*1	Max.*1	Unit*1	Reference figure
RSPCK clock cycle	Master	t_{SPcyc}	2	4096	t_{SPIcyc}	Figure 52.65
	Slave		2	4096		
RSPCK clock high level pulse width	Master	t_{SPCKWH}	(t_{SPcyc} − t_{SPCKr} − t_{SPCKf}) / 2 − 2.5	—	ns	
	Slave		1	—	t_{SPIcyc}	
RSPCK clock low level pulse width	Master	t_{SPCKWL}	(t_{SPcyc} − t_{SPCKr} − t_{SPCKf}) / 2 − 2.5	—	ns	
	Slave		1	—	t_{SPIcyc}	
RSPCK clock rising/falling time	Output	t_{SPCKr}, t_{SPCKf}	—	3	ns	
	Input		—	1	ns	
Data input setup time	Master	t_{SU}	5	—	ns	Figure 52.66 to Figure 52.72
	Slave		3	—		
Data input hold time	Master	t_H	3	—	ns	
	Slave		3	—		
SSL setup time	Master	t_{LEAD}	N × t_{SPcyc} − 3*2	N × t_{SPcyc} + 3*2	ns	Figure 52.66 to Figure 52.69
	Slave		4	—	t_{SPIcyc}	
SSL hold time	Master	t_{LAG}	N × t_{SPcyc} − 3*3	N × t_{SPcyc} + 3*3	ns	
	Slave		4	—	t_{SPIcyc}	
Continuous transmission delay	Master	t_{TD}	t_{SPcyc} + 2 × t_{SPIcyc}	8 × t_{SPcyc} + 2 × t_{SPIcyc}	ns	
	Slave		t_{SPcyc} + 5 × t_{SPIcyc}	—		
TI-SSP SS input setup time		t_{TISS}	3	—	ns	Figure 52.70 to Figure 52.72
TI-SSP SS input hold time		t_{TISH}	3	—	ns	
TI-SSP next access time		t_{TIND}	M*4	—	t_{SPIcyc}	
TI-SSP Master SS output delay		t_{TISSOD}	-3	3	ns	
TI-SSP Master OE delay 1		$t_{TIMOED1}$	—	2	ns	
TI-SSP Master OE delay 2		$t_{TIMOED2}$	—	2	ns	
TI-SSP Slave OE delay 1		$t_{TISOED1}$	—	12	ns	
TI-SSP Slave OE delay 2		$t_{TISOED2}$	—	8	ns	
Data output delay time	Master	t_{OD}	—	3	ns	Figure 52.66 to Figure 52.72
	Slave		—	12	ns	
Data output hold time	Master	t_{OH}	-3	—	ns	
	Slave		3	—		
MOSI, MISO rising/falling time	Output	t_{Dr}, t_{Df}	—	3	ns	
	Input		—	1	µs	
SSL rising/falling time	Output	t_{SSLr}, t_{SSLf}	—	3	ns	Figure 52.66, Figure 52.67
	Input		—	1	µs	

Table 52.35 SPI timing (2 of 2)

Conditions: V_{OH} = VCC33 × 0.5, V_{OL} = VCC33 × 0.5, C = 30 pF

Parameter	Symbol	Min.*1	Max.*1	Unit*1	Reference figure
Slave access time	t_{SA}	—	12	ns	Figure 52.68, Figure 52.69
Slave output release time	t_{REL}	—	12	ns	

Note 1. t_{SPIcyc}: PCLKSPIn cycle
Note 2. SPCKD set value + 1 (1 to 8)
Note 3. SSLND set value + 1 (1 to 8)
Note 4. SSLND set value + 2 (2 to 9)

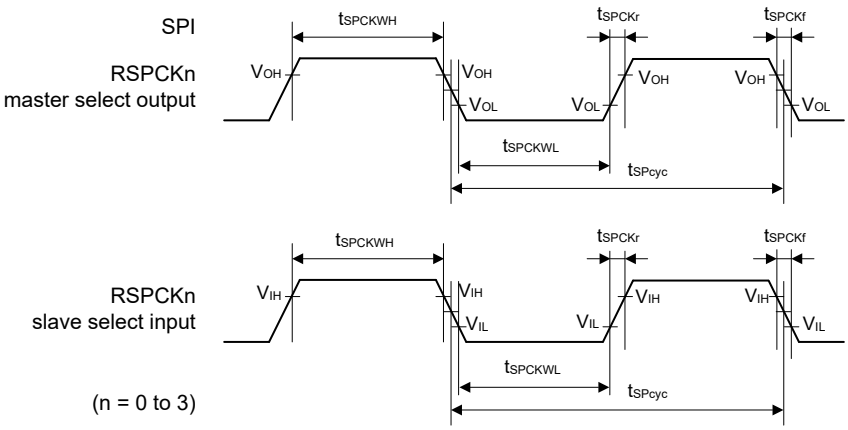


Figure 52.65 SPI clock timing

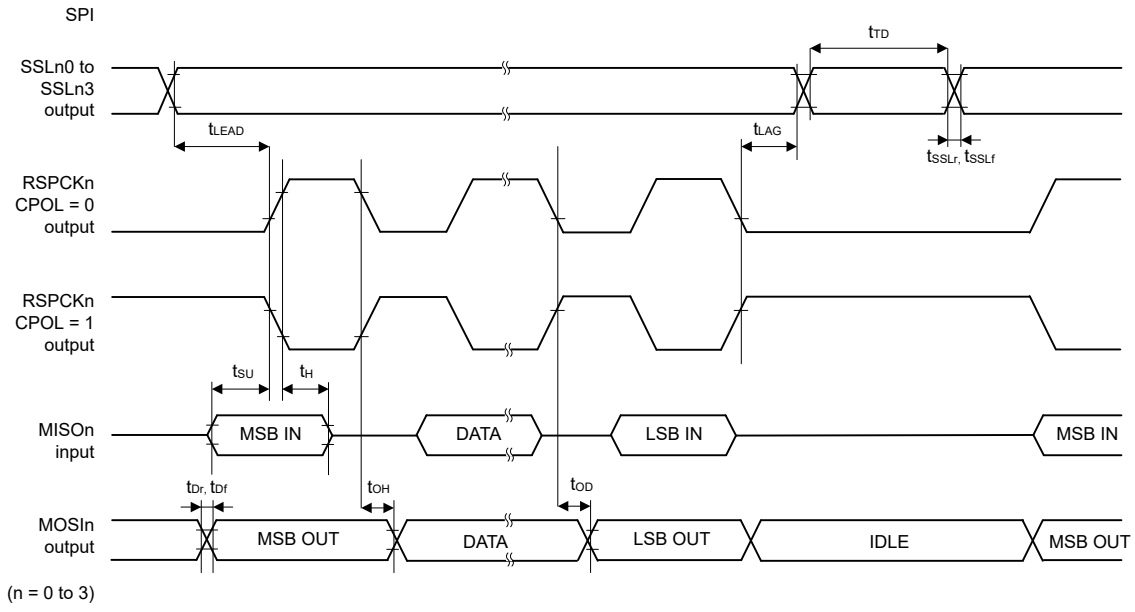


Figure 52.66 SPI timing (Master, Motorola SPI, CPHA = 0)

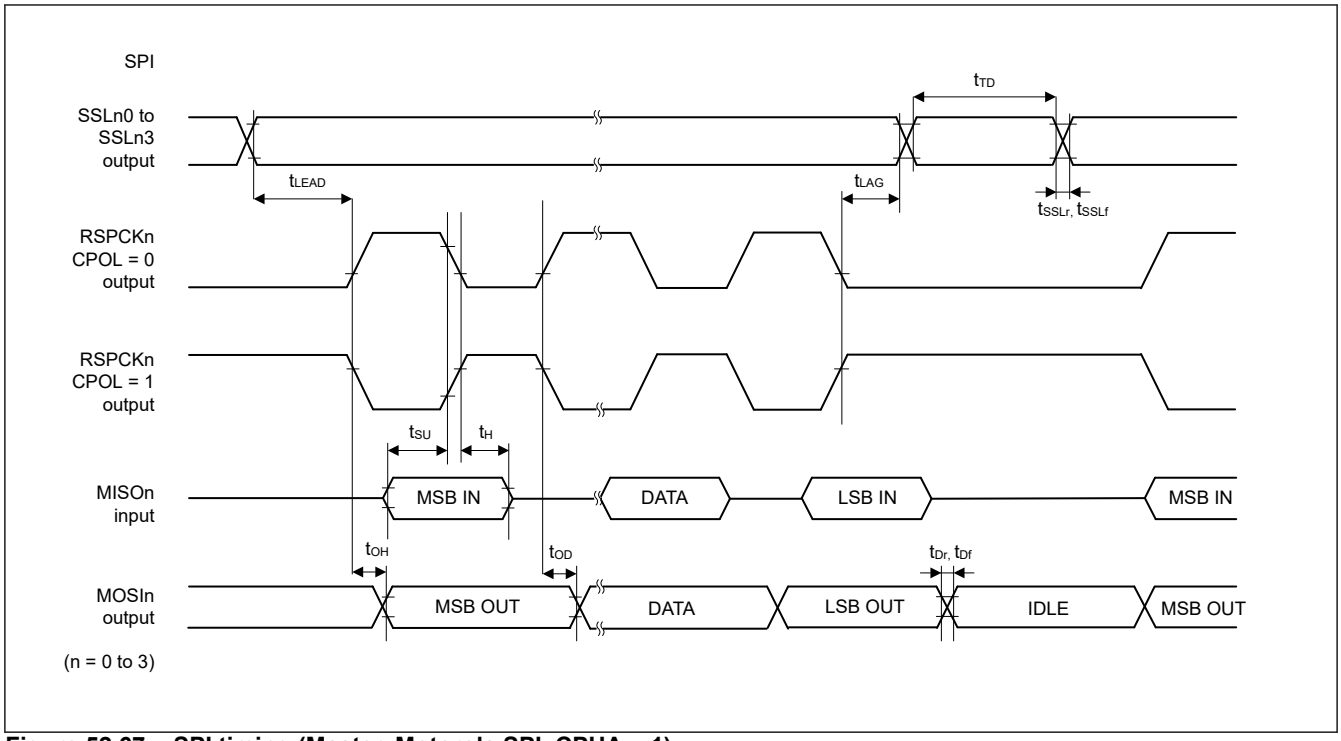


Figure 52.67 SPI timing (Master, Motorola SPI, CPHA = 1)

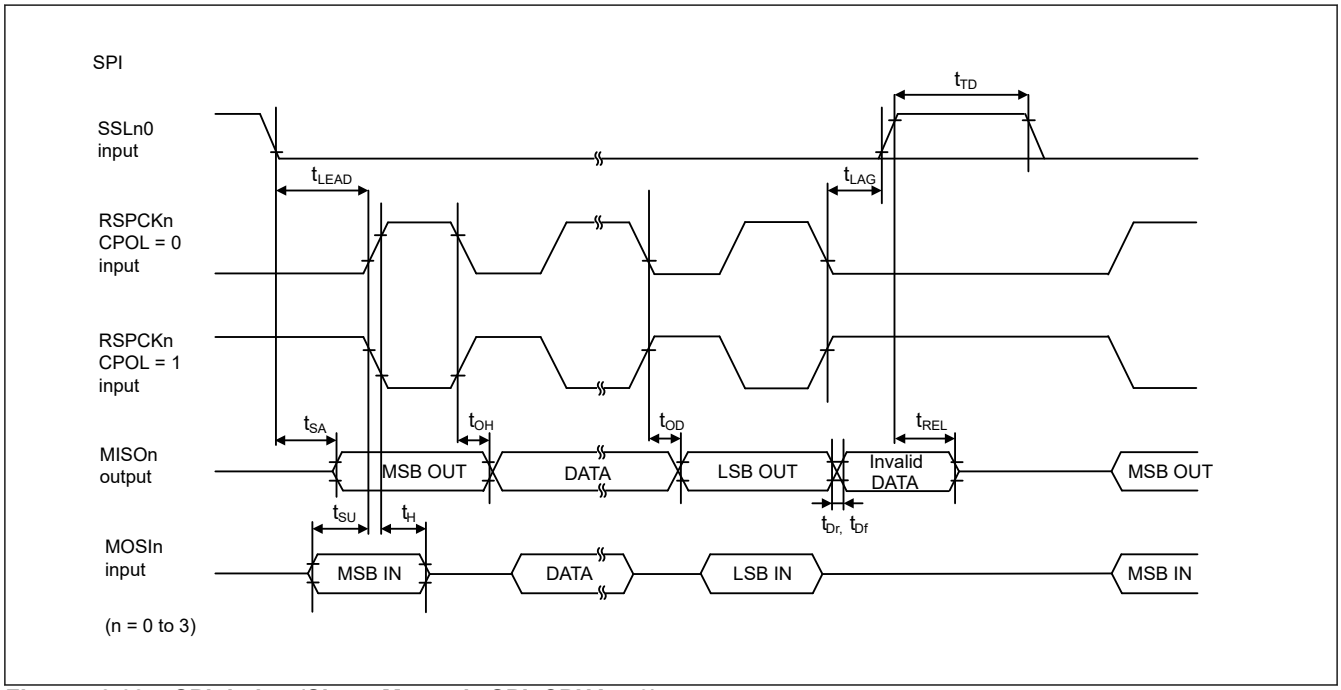


Figure 52.68 SPI timing (Slave, Motorola SPI, CPHA = 0)

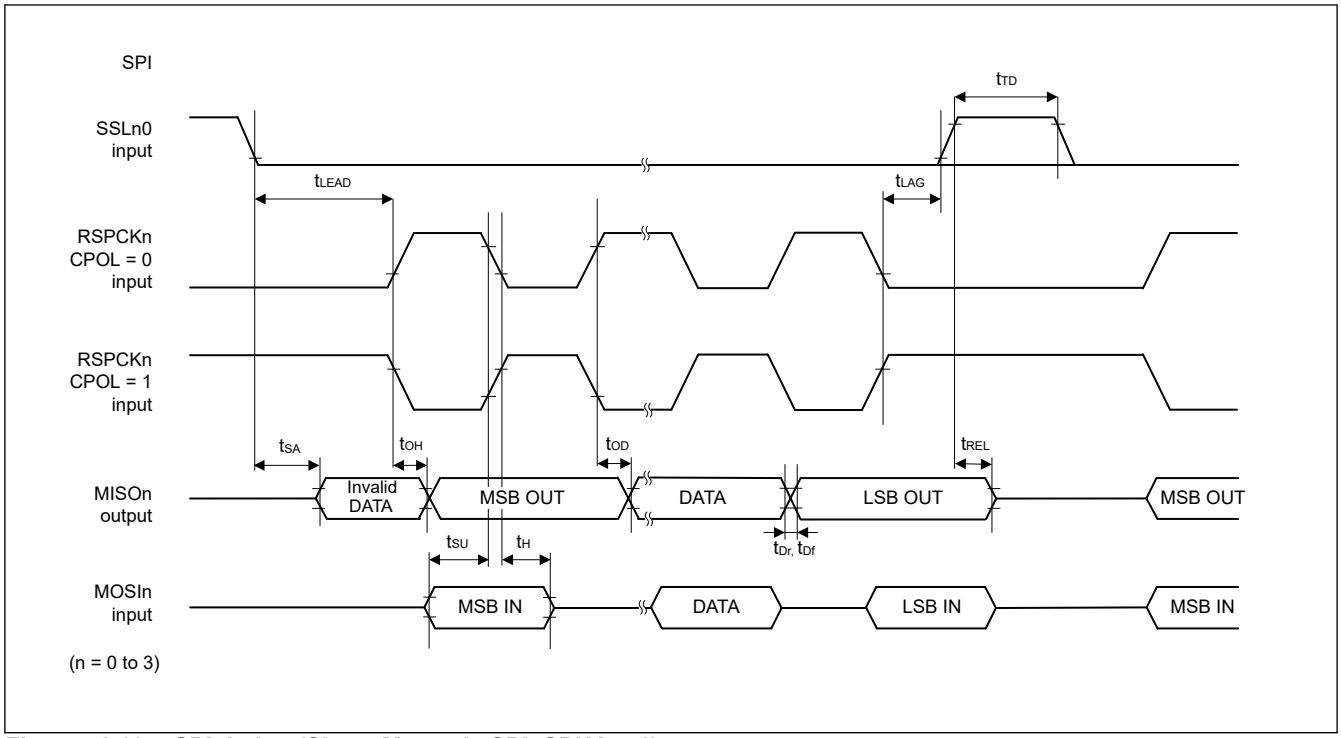


Figure 52.69 SPI timing (Slave, Motorola SPI, CPHA = 1)

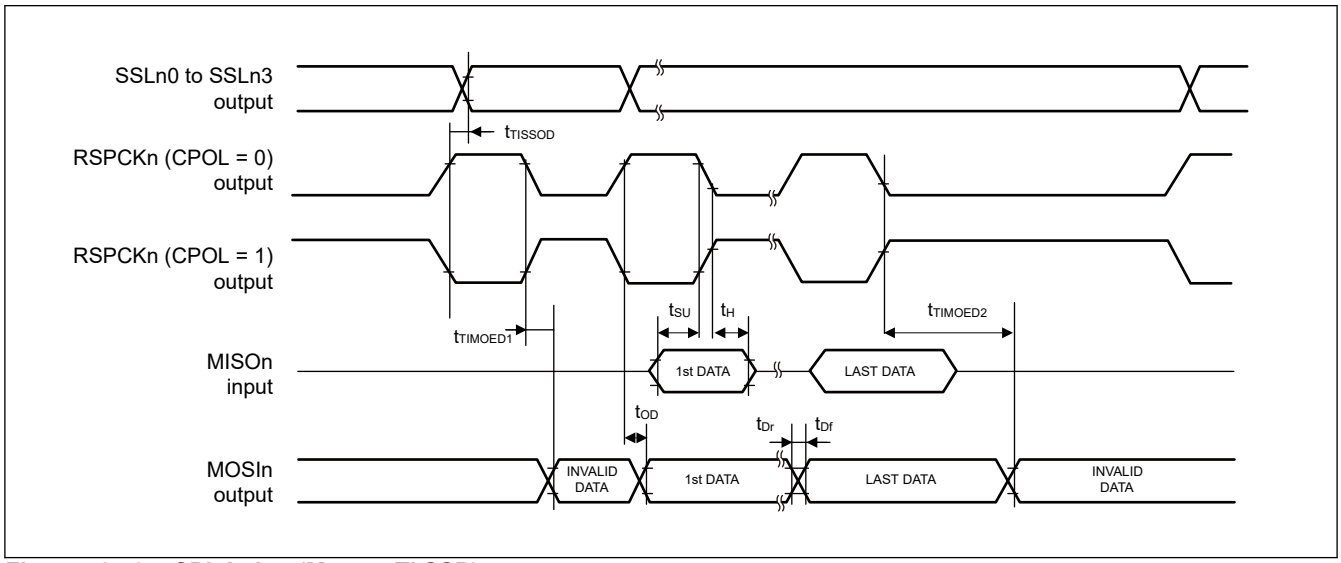


Figure 52.70 SPI timing (Master, TI SSP)

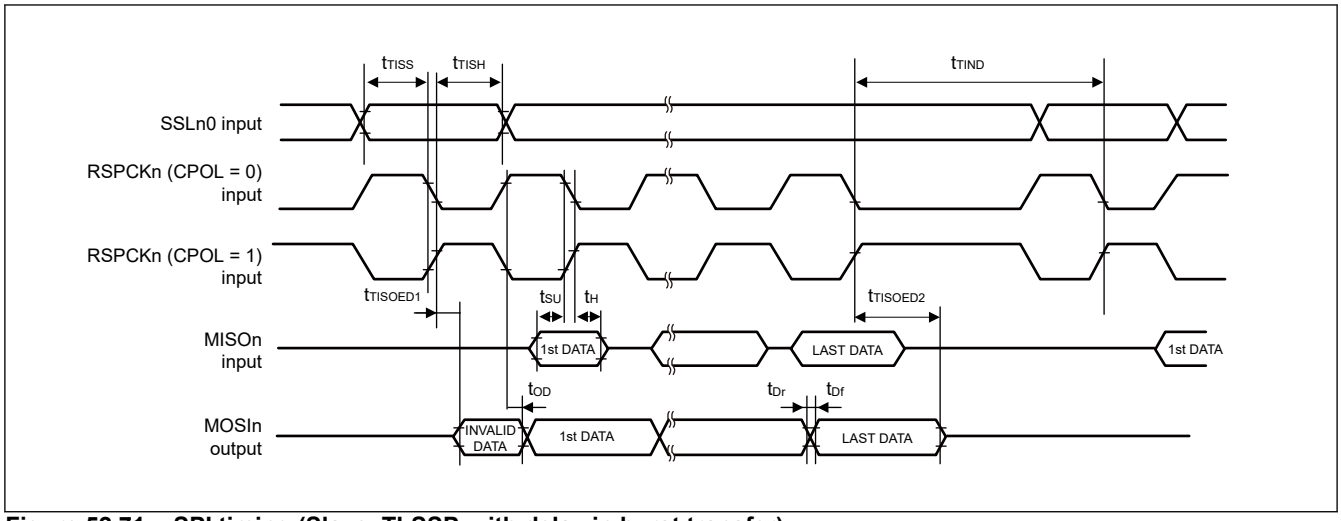


Figure 52.71 SPI timing (Slave, TI-SSP, with delay in burst transfer)

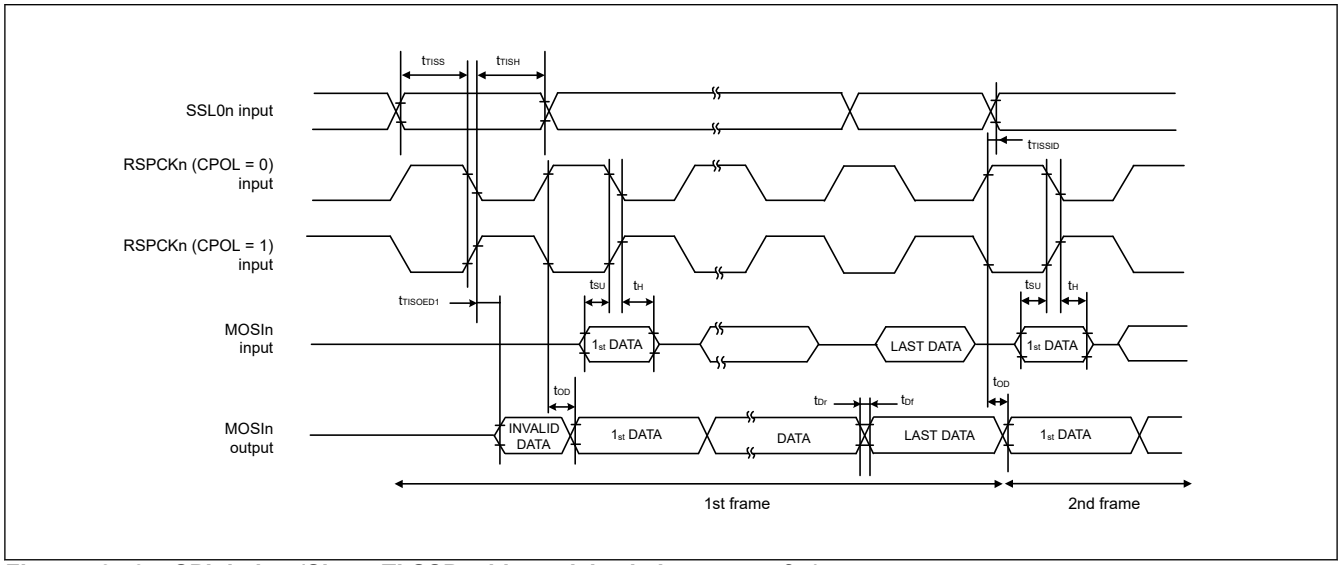


Figure 52.72 SPI timing (Slave, TI-SSP, without delay in burst transfer)

52.5.5.12 xSPI Timing

Conditions:

Single end Clock

V_{OH} = VCC18 × 0.5, V_{OL} = VCC18 × 0.5, C = 15 pF (1.8 V)

V_{OH} = VCC33 × 0.5, V_{OL} = VCC33 × 0.5, C = 15 pF (3.3 V)

Data

V_{OH} = VCC18 × 0.5, V_{OL} = VCC18 × 0.5, C = 15 pF (1.8 V)

V_{OH} = VCC33 × 0.5, V_{OL} = VCC33 × 0.5, C = 15 pF (3.3 V)

Table 52.36 xSPI timing

Parameter		Symbol	1.8 V		3.3 V		Unit	Reference figure
			Min.	Max.	Min.	Max.		
Cycle time	SDR	t_{PERIOD}	7.5	—	13.3	—	ns	Figure 52.73
	DDR		10.0	—	13.3	—	ns	
Clock output slew rate		t_{SRck}	0.75/0.56*2	—	0.56	—	V/ns	
Clock Duty cycle distortion		t_{CKDCD}	0.0	t_{PERIOD} × 0.05	0.0	t_{PERIOD} × 0.05	ns	
Clock Minimum Pulse width		t_{CKMPW}	t_{PERIOD} × 0.45	—	t_{PERIOD} × 0.45	—	ns	
Differential clock crossing voltage		$V_{OX(AC)}$	0.4 × VCC18	0.6 × VCC18	—	—	V	
DS Duty cycle distortion		t_{DSDCD}	0.0	t_{PERIOD} × 0.04	0.0	t_{PERIOD} × 0.04	ns	
DS Minimum Pulse width		t_{DSMPW}	t_{PERIOD} × 0.41	—	t_{PERIOD} × 0.41	—	ns	
Data input/output slew rate		t_{SR}	0.75/0.56*2	—	0.56	—	V/ns	
Data input setup time (to CK)	SDR	t_{SU}	2.0	—	2.4	—	ns	Figure 52.74
Data input hold time (to CK)		t_{H}	1.0	—	1.0	—	ns	
Data output delay time		t_{OD}	—	1.0*3	—	1.4*3	ns	
Data output hold time		t_{OH}	-1.0	—	-2.3	—	ns	
Data output buffer off time		t_{BOFF}	-1.0	—	-2.3	—	ns	
Data input setup time (to DS)	DDR*1 *3	t_{SU}	-0.8	—	-0.8	—	ns	Figure 52.75, Figure 52.76
Data input hold time (to DS)		t_{H}	t_{PERIOD} × 0.41 - 0.8	—	t_{PERIOD} × 0.41 - 0.8	—	ns	
Data output setup time (to CK)		t_{SUO}	1.0	—	1.0	—	ns	
Data output hold time (to CK)		t_{HO}	1.0	—	1.0	—	ns	
CS Low to Clock High		t_{CSLCKH}	6.0/8.0*2 *4	—	8.0*4	—	ns	Figure 52.74 to Figure 52.76
Clock Low to CS High		t_{CKLCSH}	6.0/8.0*2	—	8.0	—	ns	
CS High time		t_{CSTD}	1	16	1	16	t_{PERIOD}	
DS Low to CS High		t_{DSLCSH}	6.0/8.0*2	—	10.6	—	ns	Figure 52.77
CS High to DS Tri-State		t_{CSHDST}	0.0	t_{PERIOD}	0.0	t_{PERIOD}	ns	
CS Low to DS Low		t_{CSLDSL}	0.0	—	0.0	—	ns	
DS Tri-State to CS Low		t_{DSTCSL}	0.0	—	0.0	—	ns	

Note 1. The DS shift setting (WRAPCFG.DSSFTCSx[4:0]) is 01000b for xSPI200.
Note 2. Specification at 133 MHz / Specification at 100 MHz
Note 3. These are values when the OEN assertion is extended in the Output Enable Asserting extension bit (COMCFG.OEASTEX = 1).
Note 4. These are the values when the CS assertion is extended in the CS asserting extension bit (LIOCFGCSn.CSASTEX = 1).

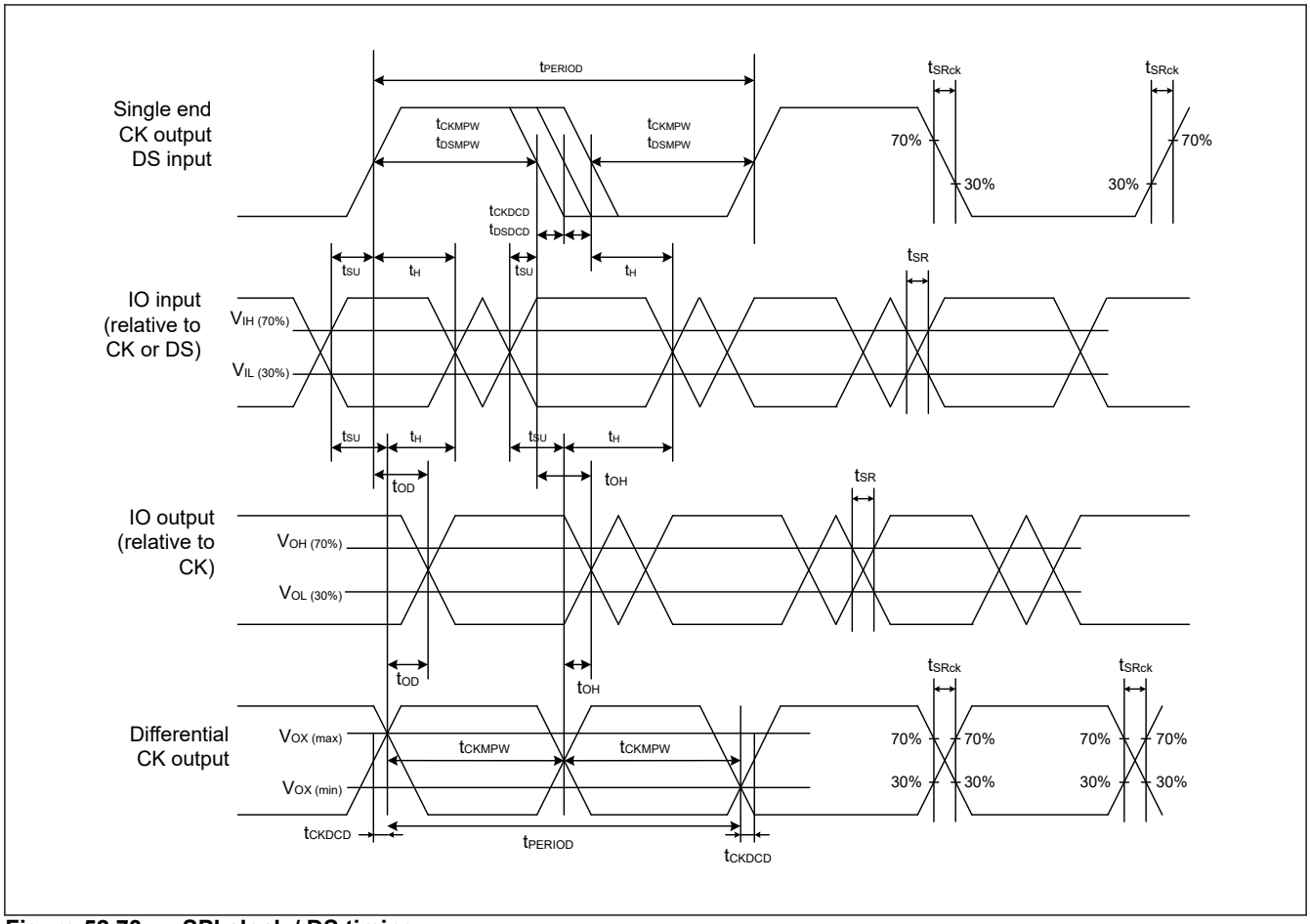


Figure 52.73 xSPI clock / DS timing

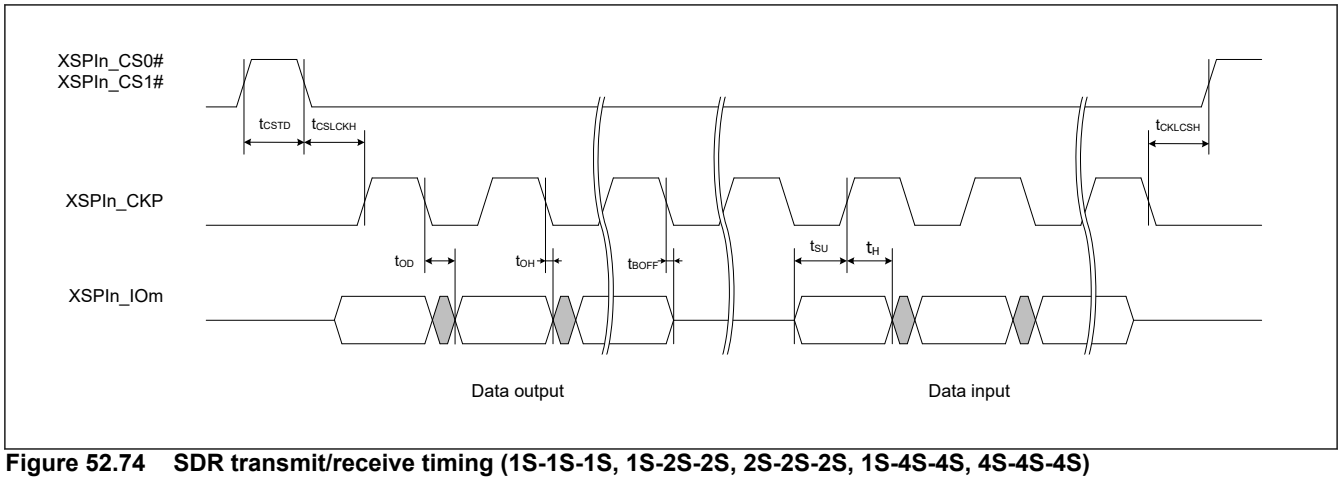


Figure 52.74 SDR transmit/receive timing (1S-1S-1S, 1S-2S-2S, 2S-2S-2S, 1S-4S-4S, 4S-4S-4S)

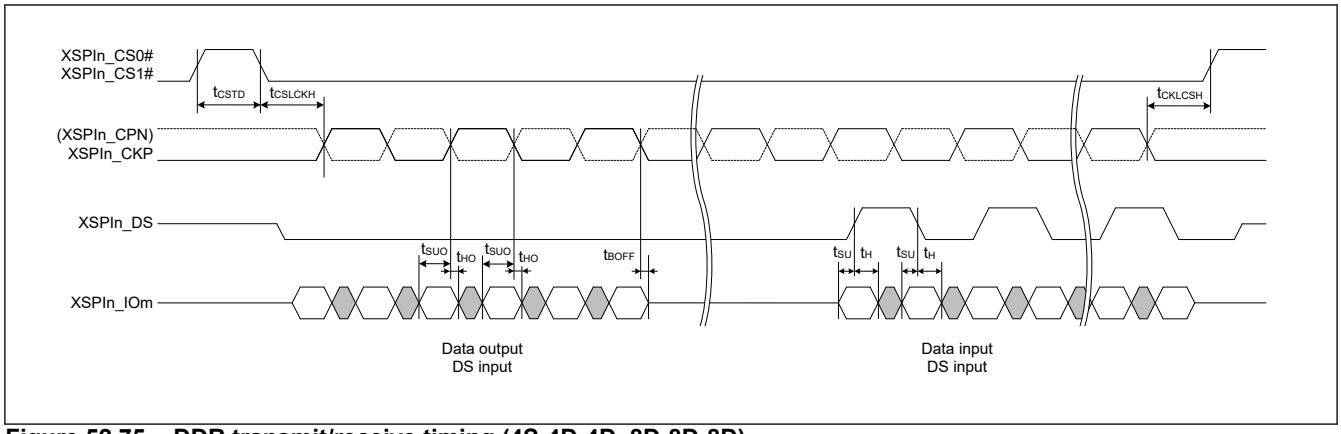


Figure 52.75 DDR transmit/receive timing (4S-4D-4D, 8D-8D-8D)

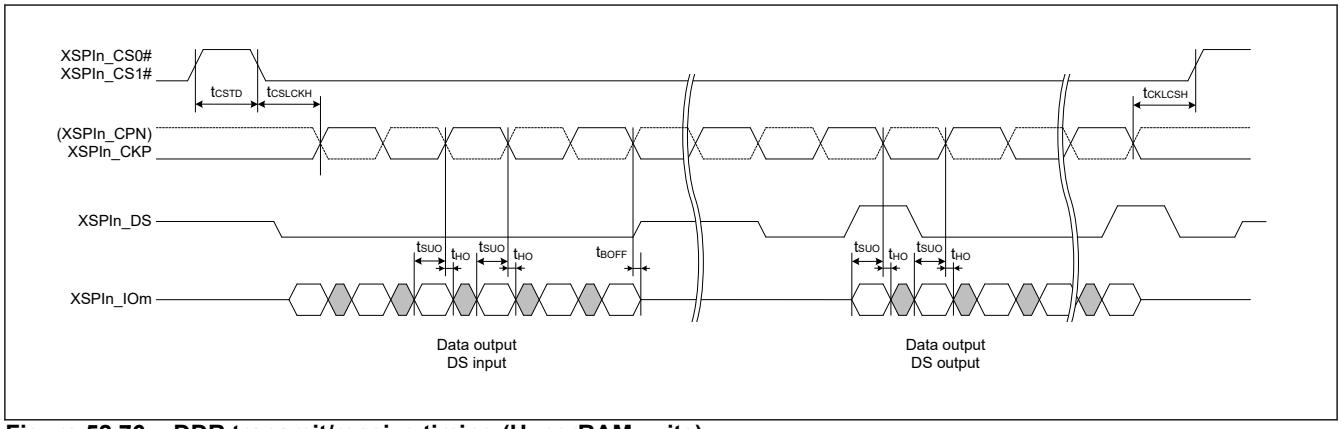


Figure 52.76 DDR transmit/receive timing (HyperRAM write)

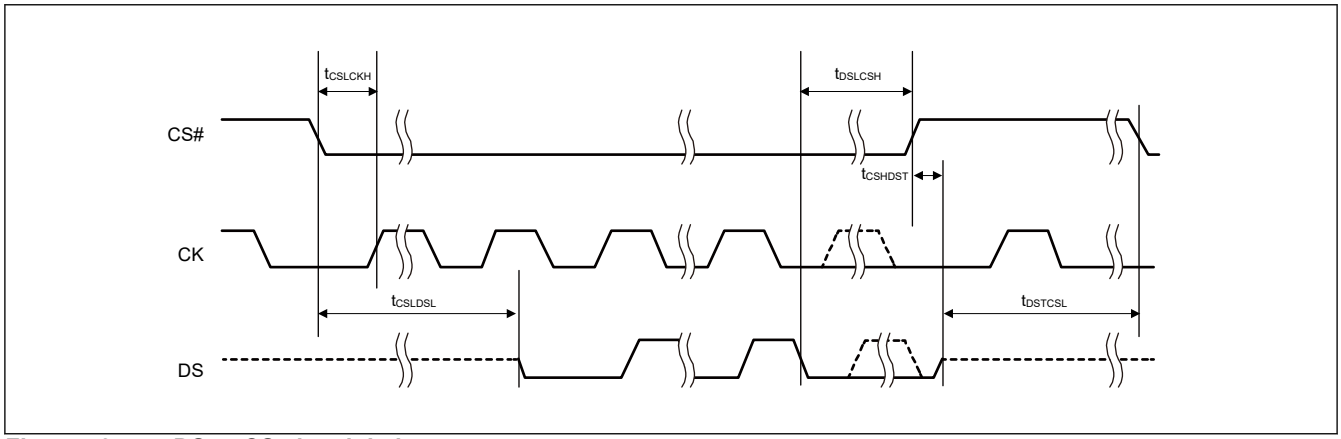


Figure 52.77 DS to CS signal timing

52.5.5.13 Delta-Sigma Interface Timing

Conditions: V_{OH} = VCC33 × 0.5, V_{OL} = VCC33 × 0.5, C = 30 pF

Table 52.37 ΔΣ interface timing

Parameter			Symbol	Min.	Max.	Unit	Reference figure
DSMIF	Clock cycle	Master	t_{DScyc}	40	200	ns	Figure 52.78
		Slave		40	200		
	Clock high level	Master	t_{DSCKWH}	16	—	ns	
		Slave		16	—		
	Clock low level	Master	t_{DSCKWL}	16	—	ns	
		Slave		16	—		
	Setup time	Master	t_{SU}	15	—	ns	Figure 52.79, Figure 52.80
		Slave		10	—		
	Hold time	Master	t_H	0	—	ns	
		Slave		10	—		

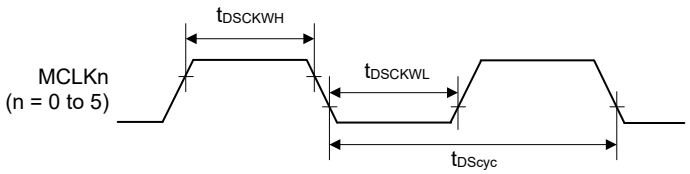


Figure 52.78 Clock input/output timing

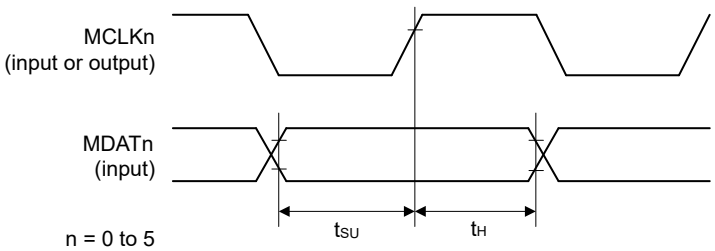


Figure 52.79 Reception timing (MCLKn rising synchronous)

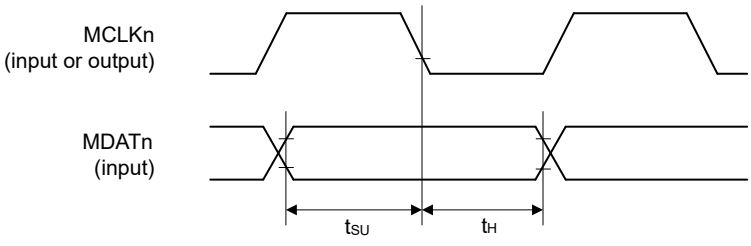


Figure 52.80 Reception timing (MCLKn falling synchronous)

52.5.5.14 Ethernet Interface Timing

Conditions:

V_{OH} = VCC18 × 0.5, V_{OL} = VCC18 × 0.5, C = 15 pF (RGMII)

V_{OH} = VCC33 × 0.5, V_{OL} = VCC33 × 0.5, C = 25 pF (RMII)

V_{OH} = VCC33 × 0.5, V_{OL} = VCC33 × 0.5, C = 30 pF (MII)

Table 52.38 Ethernet interface timing

Parameter			Symbol	Min.	Max.	Unit	Reference figure
Ethernet (RGMII)	ETHn_TXCLK, ETHn_RXCLK cycle time duration	1 Gbps	$t_{RGMIIck}$	7.2	8.8	ns	Figure 52.81
		100 Mbps		36	44		
		10 Mbps		360	440		
	ETHn_TXCLK, ETHn_RXCLK frequency	1 Gbps	—	125 − 50 ppm	125 + 50 ppm	MHz	
		100 Mbps		25 − 50 ppm	25 + 50 ppm		
		10 Mbps		2.5 − 50 ppm	2.5 + 50 ppm		
	ETHn_TXCLK, ETHn_RXCLK duty clcle	1 Gbps	—	45	55	%	
		100 Mbps 10 Mbps		40	60		
	ETHn_TXCLK, ETHn_TXD0 to ETHn_TXD3, ETHn_TXEN (TX_CTL), ETHn_RXCLK, ETHn_RXD0 to ETHn_RXD3, ETHn_RXDV (RX_CTL) rise/fall time		t_{RGMIIr}, t_{RGMIIf}	—	0.75	ns	
	ETHn_TXD0 to ETHn_TXD3, ETHn_TXEN (TX_CTL) to ETHn_TXCLK output skew		$t_{RGMIIos}$	−0.5	0.5	ns	
	ETHn_RXD0 to ETHn_RXD3, ETHn_RXDV (RX_CTL) setup time		t_{RGMIIs}	1	—	ns	
	ETHn_RXD0 to ETHn_RXD3, ETHn_RXDV (RX_CTL) hold time		t_{RGMIIh}	1	—	ns	
Ethernet (RMII)	ETHn_RXCLK cycle time		t_{RMIIck}	20	—	ns	Figure 52.82
	ETHn_RXCLK frequency Typ. 50 MHz		—	50 − 50 ppm	50 + 50 ppm	MHz	
	ETHn_RXCLK duty		—	35	65	%	
	ETHn_RXCLK rise/fall time		$t_{RMIIckr}$, $t_{RMIIckf}$	0.5	3.5	ns	
	ETHn_TXD0, ETHn_TXD1, ETHn_TXEN output delay time		t_{RMIId}	2.5	12	ns	
	ETHn_RXD0, ETHn_RXD1, ETHn_RXER, ETHn_RXDV (CRS_DV) setup time		t_{RMIIs}	4	—	ns	
	ETHn_RXD0, ETHn_RXD1, ETHn_RXER, ETHn_RXDV (CRS_DV) hold time		t_{RMIIh}	2	—	ns	
	ETHn_TXD0, ETHn_TXD1, ETHn_TXEN, ETHn_RXD0, ETHn_RXD1, ETHn_RXER, ETHn_RXDV (CRS_DV) rise/fall time		t_{RMIIr}, t_{RMIIf}	0.5	4	ns	
Ethernet (MII)	ETHn_TXCLK, ETHn_RXCLK cycle time	100 Mbps	t_{MIIck}	40	—	ns	Figure 52.83
		10 Mbps		400	—		
	ETHn_TXCLK, ETHn_RXCLK frequency	100 Mbps	—	25 − 50 ppm	25 + 50 ppm	MHz	
		10 Mbps		2.5 − 50 ppm	2.5 + 50 ppm		
	ETHn_TXD0 to ETHn_TXD3, ETHn_TXEN, ETHn_TXER output delay time		t_{MIId}	1	20	ns	
	ETHn_RXD0 to ETHn_RXD3, ETHn_RXDV, ETHn_RXER setup time		t_{MIIs}	10	—	ns	
	ETHn_RXD0 to ETHn_RXD3, ETHn_RXDV, ETHn_RXER hold time		t_{MIIh}	10	—	ns	

Table 52.39 Ethernet interface timing (MAC to MAC connection mode)

Parameter		Symbol	Min.	Max.	Unit	Reference figure
Ethernet (RGMII)	ETH2_TXCLK, ETH2_RXCLK cycle time	$t_{MRGMIIck}$	7.2	8.8	ns	Figure 52.84
	ETH2_TXCLK, ETH2_RXCLK frequency Typ. 125 MHz	—	125 − 50 ppm	125 + 50 ppm	MHz	
	ETH2_TXCLK, ETH2_RXCLK duty clcle	—	45	55	%	
	ETH2_TXCLK, ETH2_TXD0 to ETH2_TXD3, ETH2_TXEN (TX_CTL), ETH2_RXCLK, ETH2_RXD0 to ETH2_RXD3, ETH2_RXDV (RX_CTL) rise/fall time	$t_{MRGMIIr}$, $t_{MRGMIIf}$	—	0.75	ns	
	ETH2_TXD0 to ETHn_TXD3, ETHn_TXEN (TX_CTL) output skew	$t_{MRGMIIos}$	—	0.6	ns	
	ETH2_TXD0 to ETH2_TXD3, ETH2_TXEN (TX_CTL) output setup time	$t_{MRGMIIso}$	1.1	—	ns	
	ETH2_TXD0 to ETH2_TXD3, ETH2_TXEN (TX_CTL) output hold time	$t_{MRGMIIho}$	1.1	—	ns	
	ETH2_RXD0 to ETH2_RXD3, ETH2_RXDV (RX_CTL) input setup time	$t_{MRGMIIsi}$	-0.7	—	ns	
	ETH2_RXD0 to ETH2_RXD3, ETH2_RXDV (RX_CTL) input hold time	$t_{MRGMIIhi}$	2.9	—	ns	
Ethernet (MII)	ETH2_REFCLK cycle time	t_{MMIIck}	40	—	ns	Figure 52.85
	ETH2_REFCLK frequency Typ. 25 MHz	—	25 − 50 ppm	25 + 50 ppm	MHz	
	ETH2_TXD0 to ETH2_TXD3, ETH2_TXEN, ETH2_TXER output delay time	t_{MMIId}	11	25	ns	
	ETH2_RXD0 to ETH2_RXD3, ETH2_RXDV, ETH2_RXER setup time	t_{MMIIs}	10	—	ns	
	ETH2_RXD0 to ETH2_RXD3, ETH2_RXDV, ETH2_RXER hold time	t_{MMIIh}	0	—	ns	

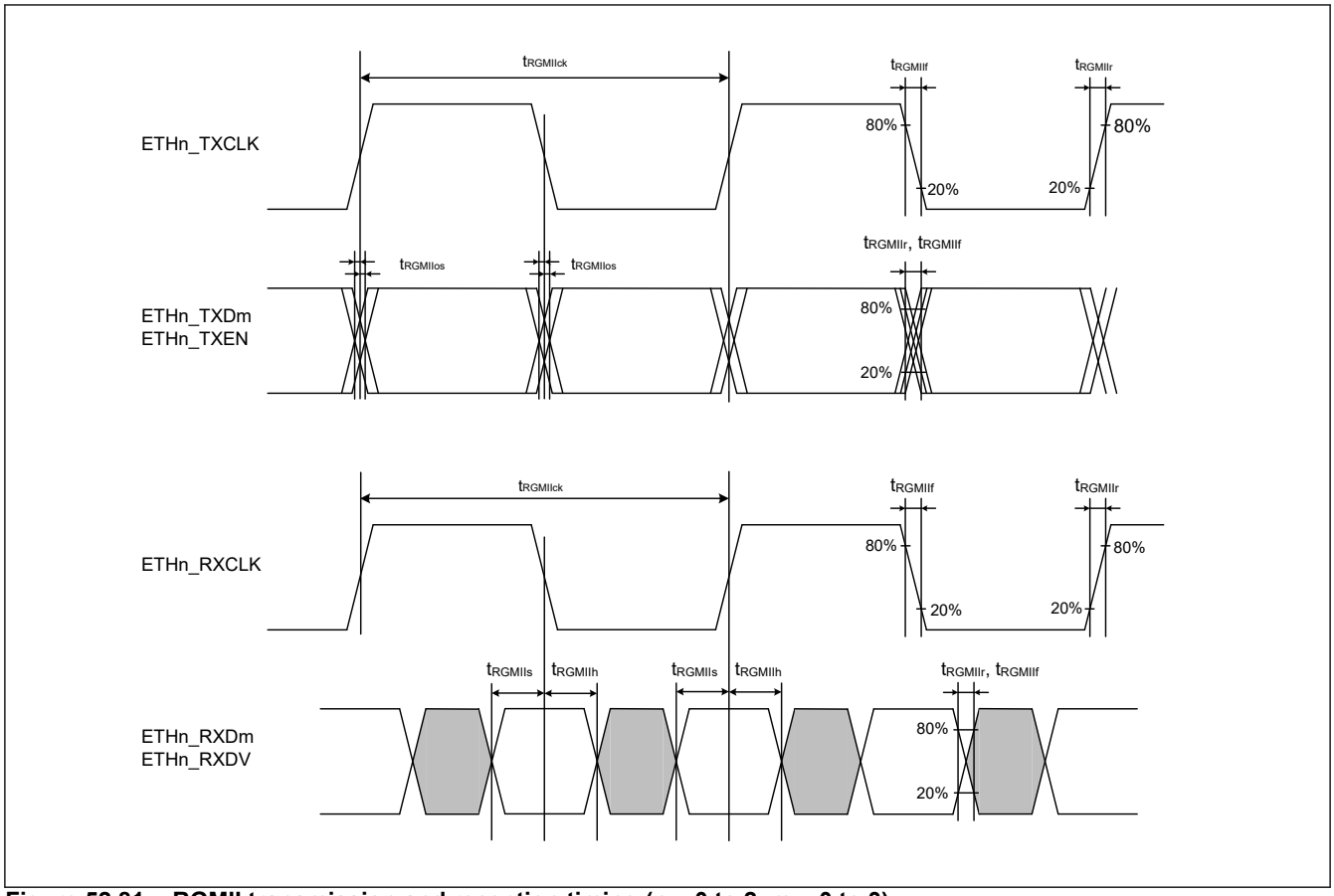


Figure 52.81 RGMII transmission and reception timing (n = 0 to 2, m = 0 to 3)

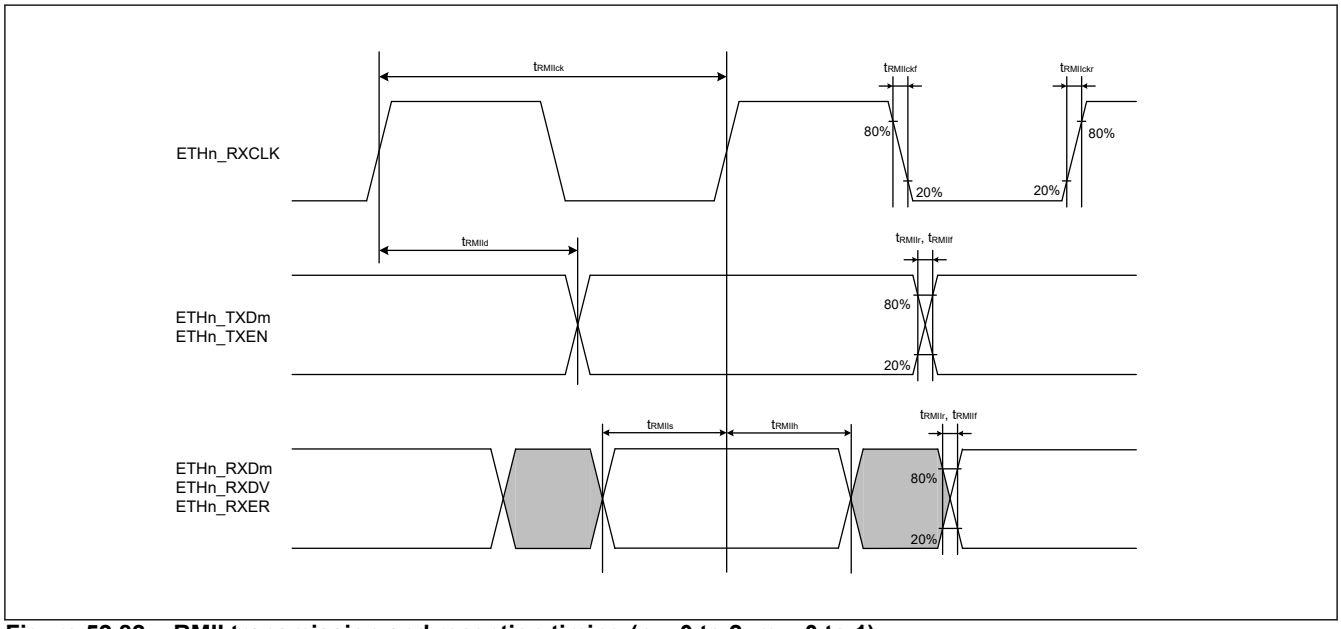


Figure 52.82 RMII transmission and reception timing (n = 0 to 2, m = 0 to 1)

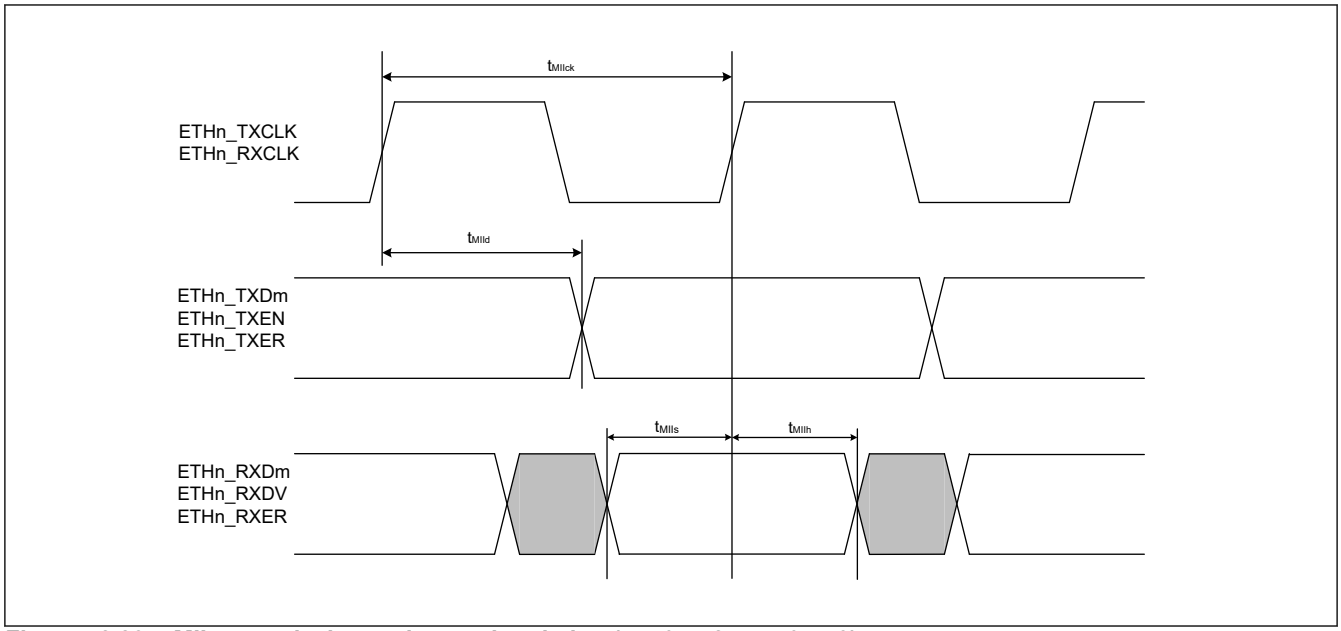


Figure 52.83 MII transmission and reception timing (n = 0 to 2, m = 0 to 3)

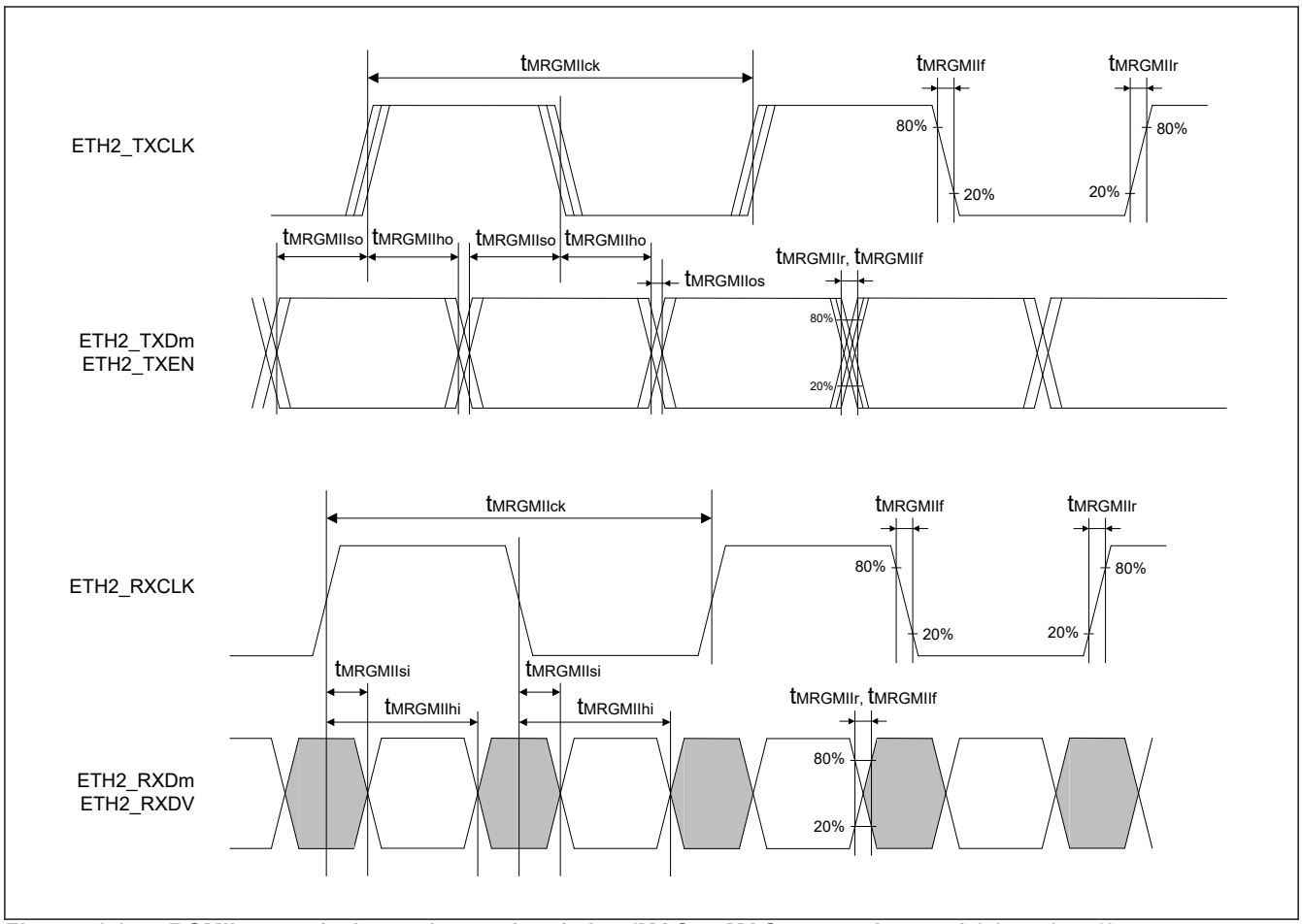


Figure 52.84 RGMII transmission and reception timing (MAC to MAC connection mode) (m = 0 to 3)

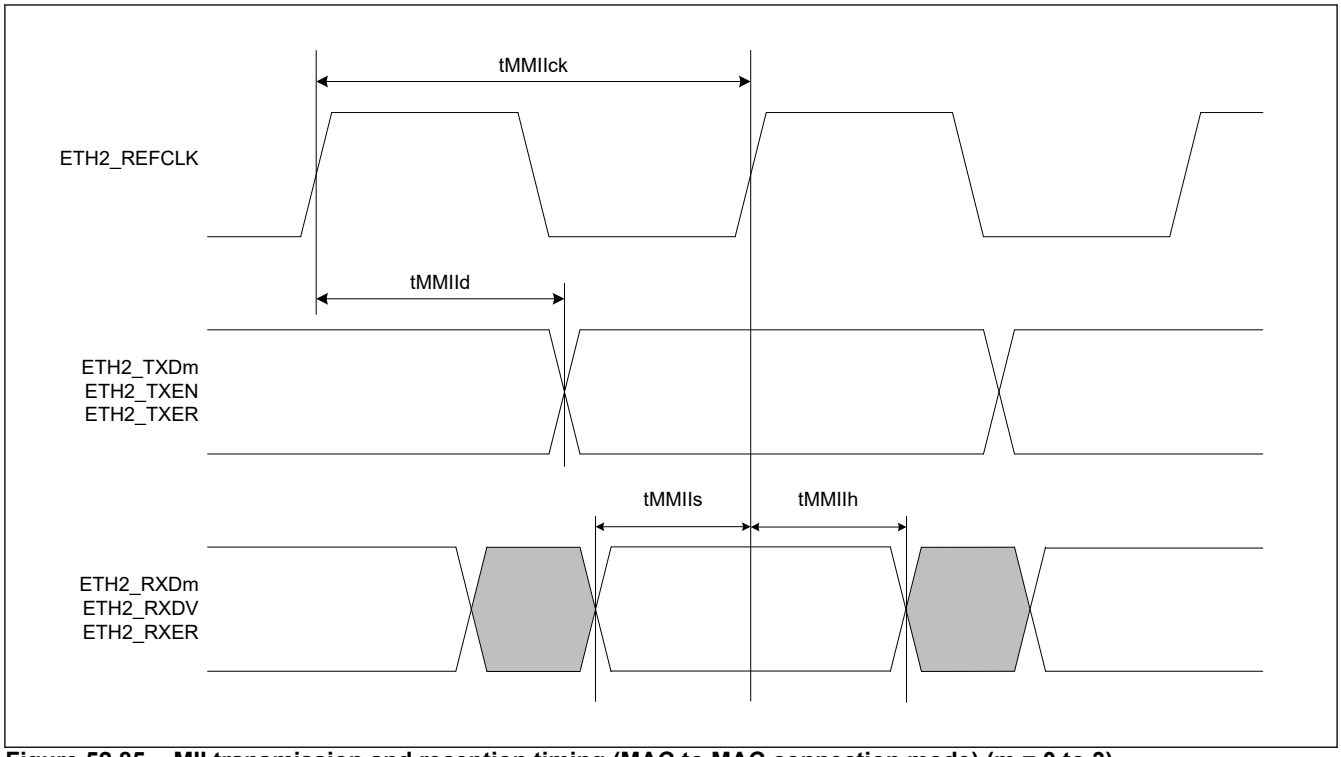


Figure 52.85 MII transmission and reception timing (MAC to MAC connection mode) (m = 0 to 3)

52.5.5.15 Serial Management Interface Timing

Conditions:

V_{OH} = VCC18 × 0.5, V_{OL} = VCC18 × 0.5, C = 30 pF (1.8 V)

V_{OH} = VCC33 × 0.5, V_{OL} = VCC33 × 0.5, C = 30 pF (3.3 V)

Table 52.40 Serial management interface timing

Parameter			Symbol	Min.	Max.	Unit	Reference figure
MDIO	MDC output cycle time	GMAC_MDC, ETHSW_MDC	T_{MDCck}	80	—	ns	Figure 52.86
		ESC_MDC		400	—	ns	
	MDIO output delay time (for MDC fall)*1		T_{MDIOd}	—	20	ns	
	MDIO input setup time (for MDC rise)	GMAC_MDC, ETHSW_MDC	T_{MDIOs}	18	—	ns	
		ESC_MDC		70	—	ns	
	MDIO input hold time (for MDC rise)		T_{MDIOh}	0	—	ns	

Note 1. The output timing from ETHSW is based on the rising edge of MDC, and the output delay can be set in the register.

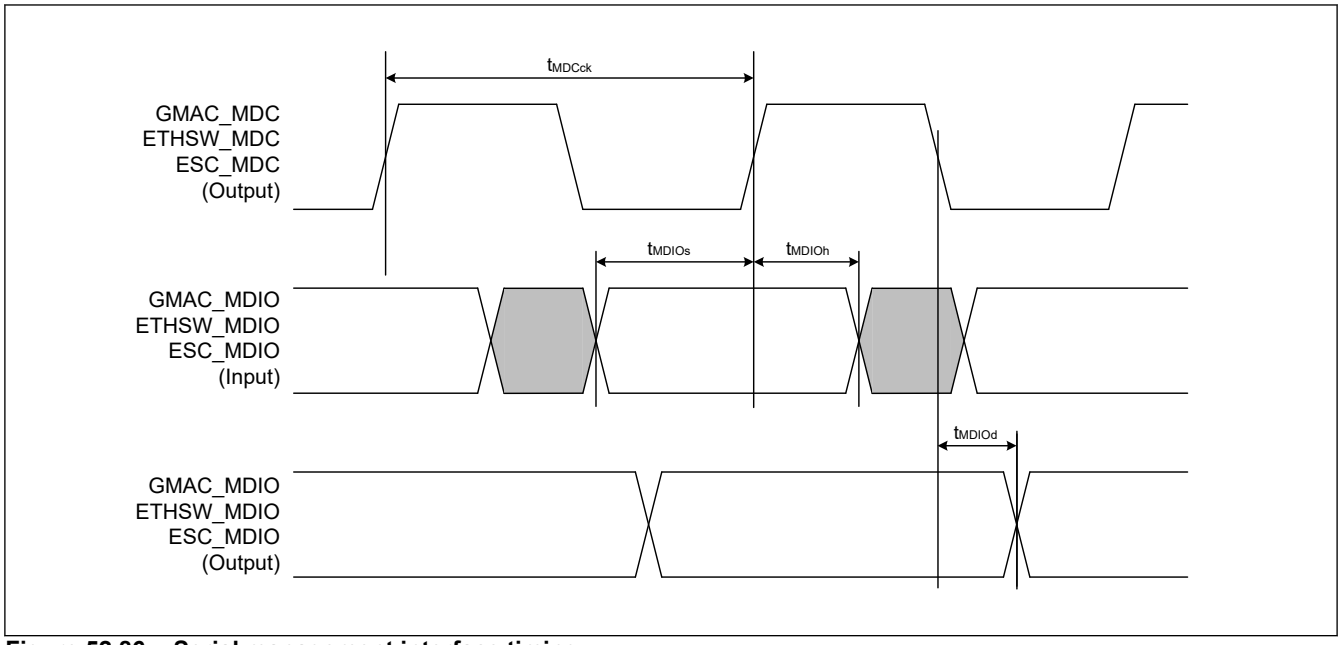


Figure 52.86 Serial management interface timing

52.5.5.16 SHOSTIF Timing

Conditions:

V_{OH} = VCC18 × 0.5, V_{OL} = VCC18 × 0.5, C = 30 pF (1.8 V)

V_{OH} = VCC33 × 0.5, V_{OL} = VCC33 × 0.5, C = 30 pF (3.3 V)

Table 52.41 SHOSTIF timing

Parameter		Symbol	Min.	Max.	Unit	Reference figure
SHOSTIF	Clock cycle time	t_{SHck}	25	—	ns	Figure 52.87
	Clock high time	t_{CH}	0.45	0.55	t_{SHck}	
	Clock low time	t_{CL}	0.45	0.55	t_{SHck}	
	Clock rise slew rate	t_{CRT}	0.1	—	V/ns	
	Clock fall slew rate	t_{CFT}	0.1	—	V/ns	
	CS# high time	t_{CS}	2	—	t_{SHck}	Figure 52.88, Figure 52.89
	CS# active setup time	t_{CSS}	15	—	ns	
	CS# active hold time	t_{CSH}	15	—	ns	
	Data input setup time	t_{SU}	3	—	ns	
	Data input hold time	t_{HD}	10.5	—	ns	
	Clock low to output valid	t_{V}	—	15.5	ns	
	Data output hold time	t_{HO}	6	—	ns	
	Data output disable time	t_{DIS}	—	18	ns	

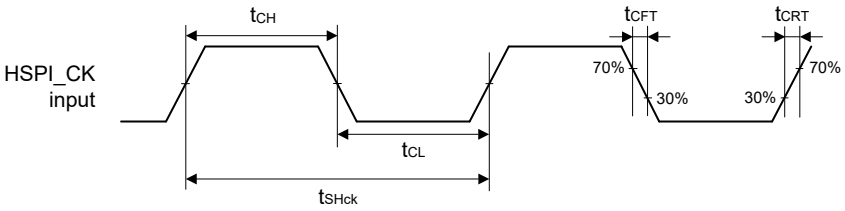
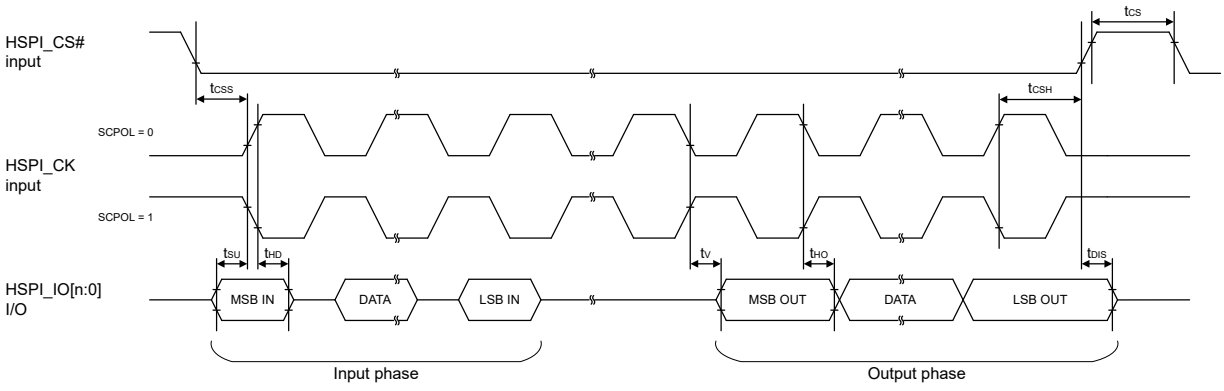
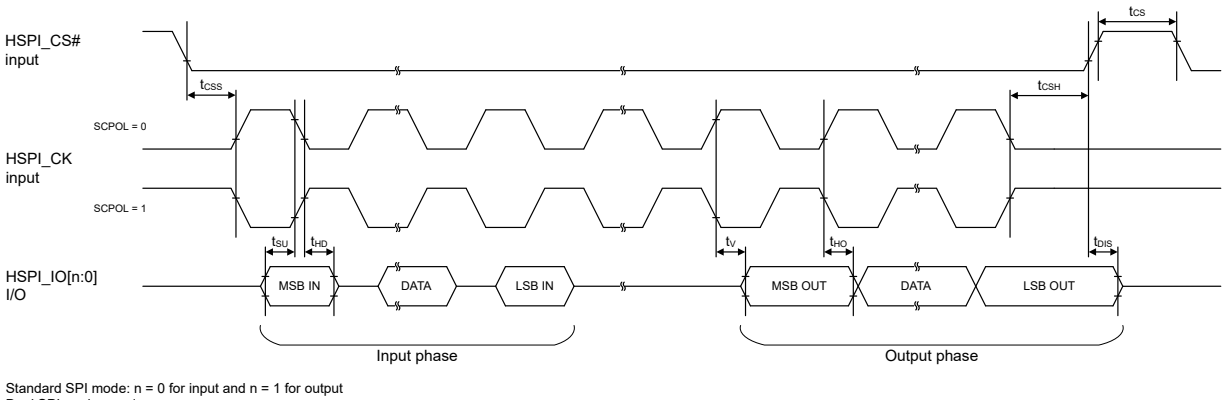


Figure 52.87 SHOSTIF clock timing



Standard SPI mode: n = 0 for input and n = 1 for output
Dual SPI mode: n = 1
Quad SPI mode: n = 3
Octal SPI mode: n = 7

Figure 52.88 SHOSTIF timing (SCPH = 0)



Standard SPI mode: n = 0 for input and n = 1 for output
Dual SPI mode: n = 1
Quad SPI mode: n = 3
Octal SPI mode: n = 7

Figure 52.89 SHOSTIF timing (SCPH = 1)

52.6 USB Characteristics

Table 52.42 On-chip USB low-speed (host only) characteristics

Parameter	Symbol	Min.	Typ.	Max.	Unit	Reference figure
Rising time	t_{LR}	75	—	300	ns	Figure 52.90, Figure 52.91
Falling time	t_{LF}	75	—	300	ns	
Rising/falling time ratio	t_{LR}/t_{LF}	80	—	125	%	

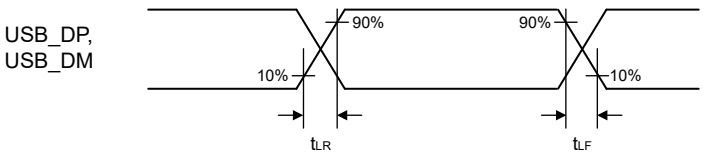


Figure 52.90 USB_DP, USB_DM output timing (low-speed/host only)

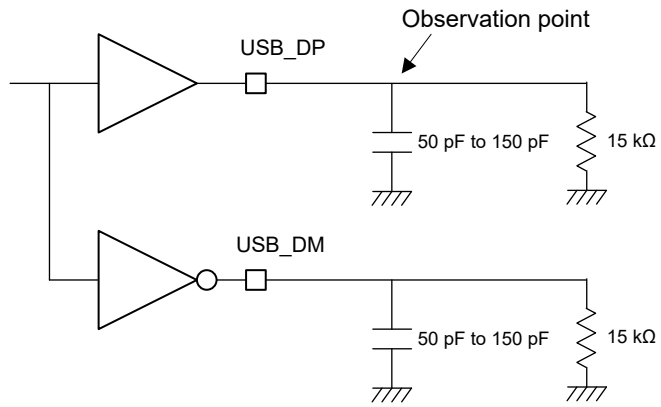


Figure 52.91 Measurement circuit (low-speed/host only)

Table 52.43 On-chip USB full-speed characteristics

Parameter	Symbol	Min.	Typ.	Max.	Unit	Reference figure
Rising time	t_{FR}	4	—	20	ns	Figure 52.92, Figure 52.93
Falling time	t_{FF}	4	—	20	ns	
Rising/falling time ratio	t_{FR}/t_{FF}	90	—	111.11	%	

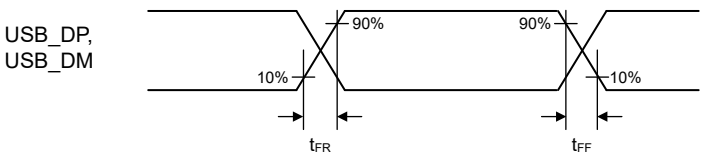


Figure 52.92 USB_DP, USB_DM output timing (full speed)

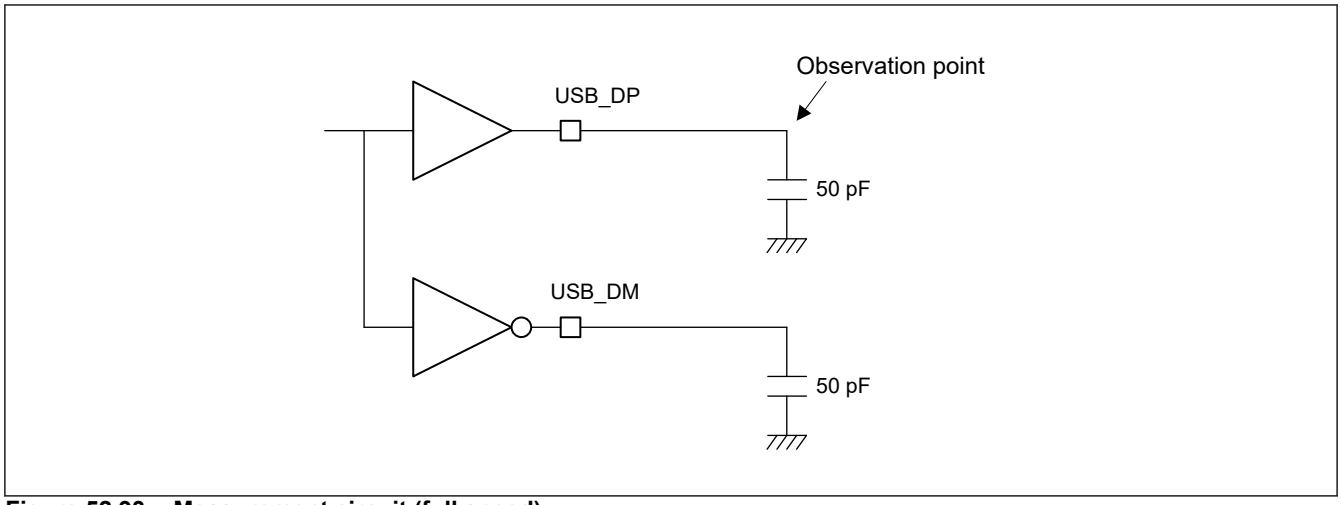


Figure 52.93 Measurement circuit (full speed)

Table 52.44 On-chip USB high-speed characteristics

Parameter	Symbol	Min.	Typ.	Max.	Unit	Reference figure
Rising time	t_{HSR}	—	—	2.133	V/µs	Figure 52.94, Figure 52.95
Falling time	t_{HSF}	—	—	2.133	V/µs	
Output resistors (also used as high-speed terminating resistor)	Z_{HSDRV}	40.5	—	49.5	Ω	—

Note: The output resistors (ZHSDRV) for connection to the USB_DP and USB_DM pins are within the LSI.

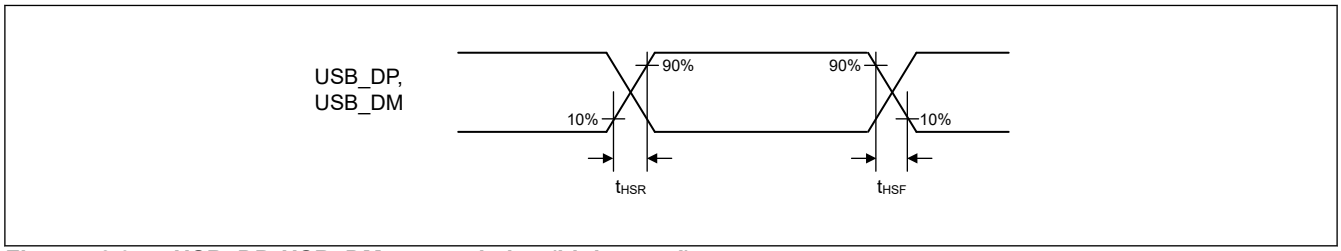


Figure 52.94 USB_DP, USB_DM output timing (high speed)

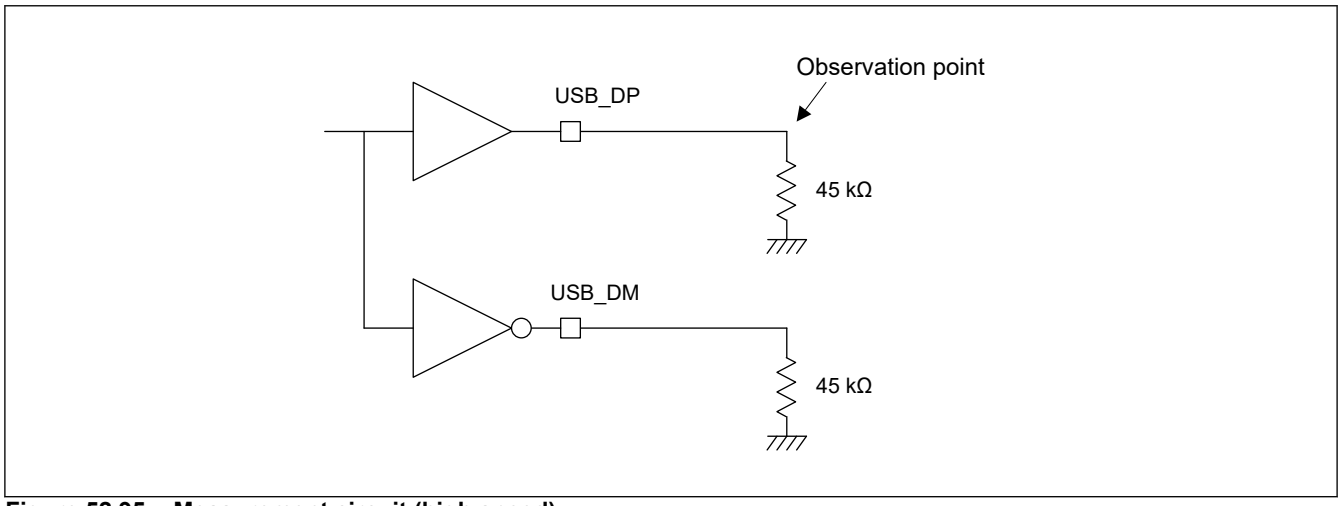


Figure 52.95 Measurement circuit (high speed)

52.7 A/D Conversion Characteristics

Table 52.45 12-Bit A/D conversion characteristics

Parameter		Min.	Typ.	Max.	Unit	Reference figure
Resolution		12			bits	—
Analog input capacitance		—	—	13	pF	—
Channel-dedicated sample-and-hold circuits in use (AN000 to AN002, AN100 to AN102)	Conversion time*1 Permissible signal source impedance Max. = 1.0 kΩ	1.52	—	—	µs	—
	Offset error	—	—	±13	LSB	—
	Full-scale error	—	—	±13	LSB	—
	Quantization error	—	±0.5	—	LSB	—
	Absolute accuracy	—	—	±14	LSB	—
	DNL differential non-linearity error	—	—	±3	LSB	—
	INL integral non-linearity error	—	—	±4	LSB	—
	Holding characteristics of sample-and-hold circuits	—	—	2.67	µs	—
	Dynamic range	0.15	—	VREFH0 - 0.15	V	—
Channel-dedicated sample-and-hold circuits not in use (AN000 to AN003, AN100 to AN103)	Conversion time*1 Permissible signal source impedance Max. = 1.0 kΩ	0.84	—	—	µs	—
	Offset error	—	—	±11	LSB	—
	Full-scale error	—	—	±11	LSB	—
	Quantization error	—	±0.5	—	LSB	—
	Absolute accuracy	—	—	±12	LSB	—
	DNL differential non-linearity error	—	—	±3	LSB	—
	INL integral non-linearity error	—	—	±4	LSB	—

Note: The specified values in the table apply when there is no access to the external bus during A/D conversion. If access proceeds during A/D conversion, values may not fall within the specified ranges.

Note 1. The conversion time is the total of the sampling time and the comparison time.

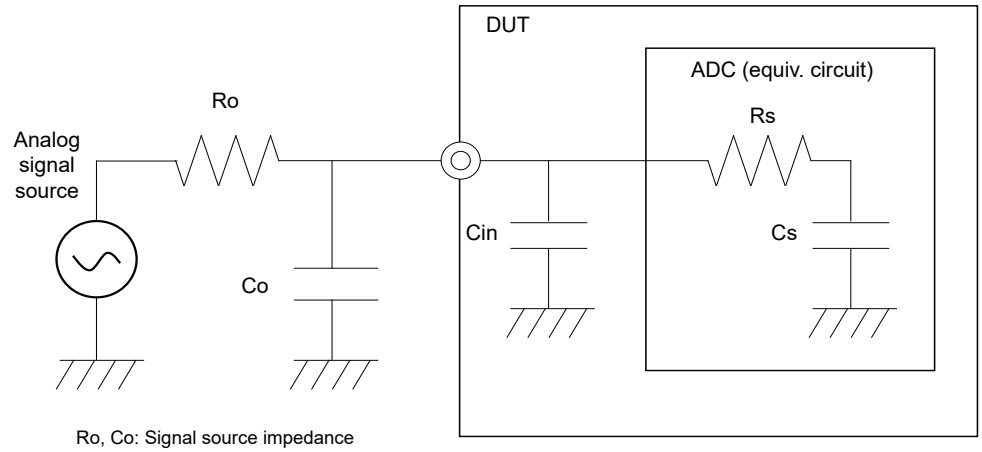


Figure 52.96 A/D converter equivalent circuit and peripheral configuration diagram

52.8 Temperature Sensor Characteristics

Table 52.46 Temperature sensor characteristics

Parameter	Min.	Typ.	Max.	Unit	Test Conditions
Relative accuracy	—	±1	—	°C	*1
Temperature slope	—	0.0625	—	°C/LSB	—
Output code (at 25°C)	—	1545 (decimal)	—	—	TSUSAD register

Note 1. 2-point calibration (Tj = 25°C and Tj = 85°C) and 8 times averaging. For details of 2-point calibration, see section 40.3.6.1. Preparation for Using the Temperature Sensor.

52.9 Debug Interface Timing

Condition: V_{OH} = VCC33 × 0.5, V_{OL} = VCC33 × 0.5

Table 52.47 Debug interface timing (1 of 2)

Parameter		Symbol	Min.	Max.	Unit	Reference figure
TCK cycle time	With an ICE connected	t_{TCKcyc}	30*1	—	ns	Figure 52.97
	For use in BSCAN		80	—		
TCK high pulse width		t_{TCKH}	0.4	0.6	t_{TCKcyc}	
TCK low pulse width		t_{TCKL}	0.4	0.6	t_{TCKcyc}	
TDI setup time		t_{TDIS}	5	—	ns	Figure 52.98 Output load: 30 pF
TDI hold time		t_{TDIH}	5	—	ns	
TMS/SWDIO setup time		t_{TMSS}	5	—	ns	
TMS/SWDIO hold time		t_{TMSH}	5	—	ns	
SWDIO delay time		t_{SWDO}	—	15	ns	
TDO delay time	With an ICE connected	t_{TDOD}	—	15	ns	
	For use in BSCAN		—	22		

Table 52.47 Debug interface timing (2 of 2)

Parameter	Symbol	Min.	Max.	Unit	Reference figure
Capture register setup time	t_{CAPTS}	5	—	ns	Figure 52.99
Capture register hold time	t_{CAPTH}	5	—	ns	
Update register delay time	$t_{UPDATED}$	—	15	ns	
Trace clock cycle	t_{TCYC}	20	—	ns	Figure 52.100 Output load: 15 pF
Trace data delay time	t_{TDT}	1.3	8.7	ns	

Note 1. This value is the minimum cycle time for the normal operation of internal circuits.
The actual cycle time should be determined in consideration of the TCK capture edge timing and cable length of the connected ICE.

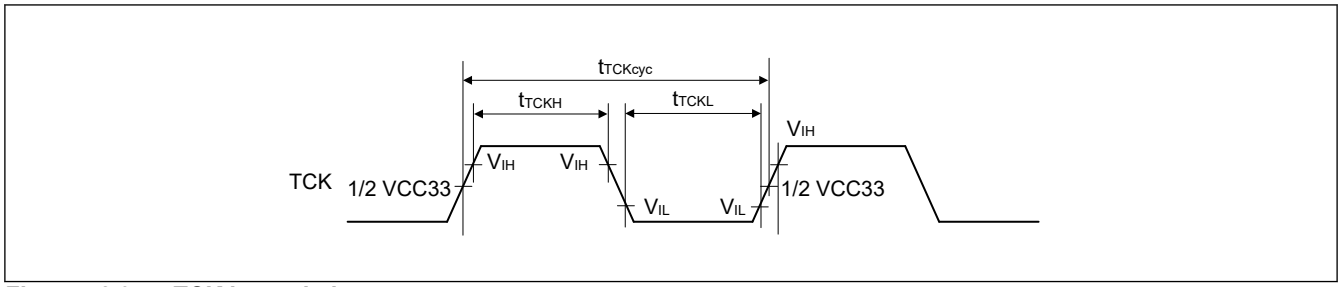


Figure 52.97 TCK input timing

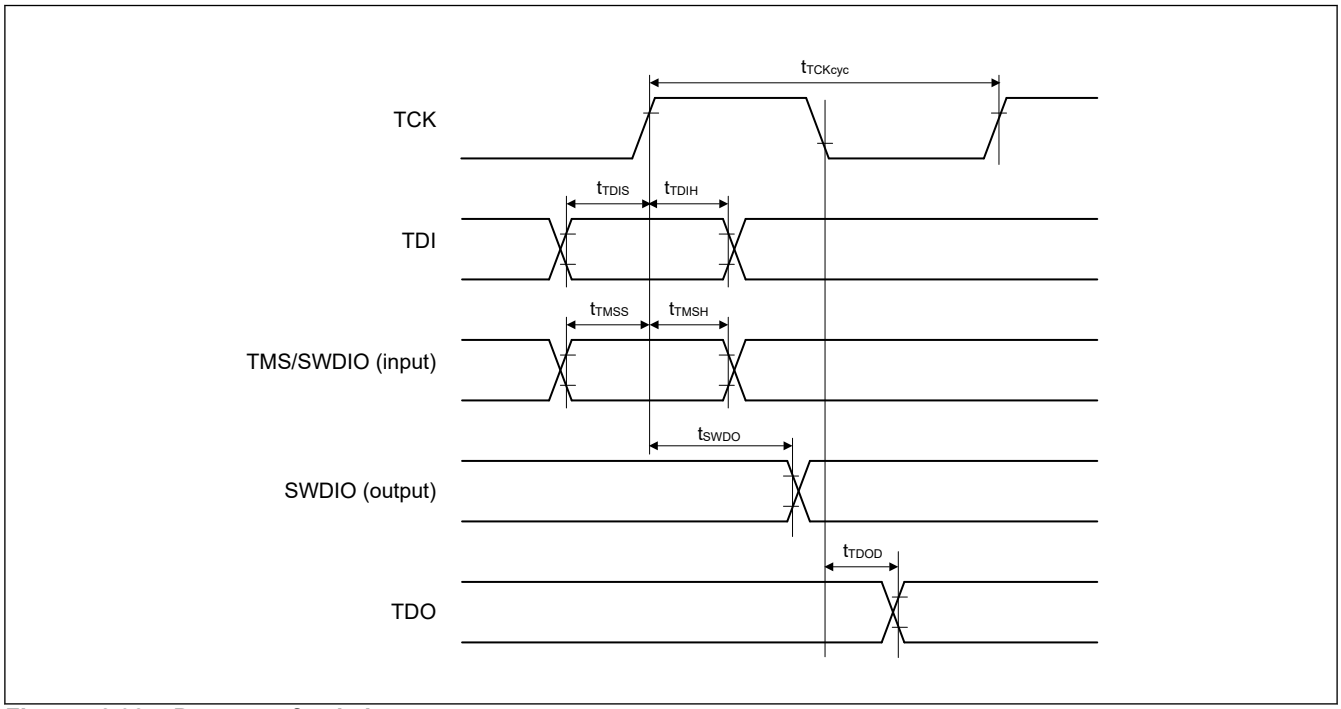


Figure 52.98 Data transfer timing

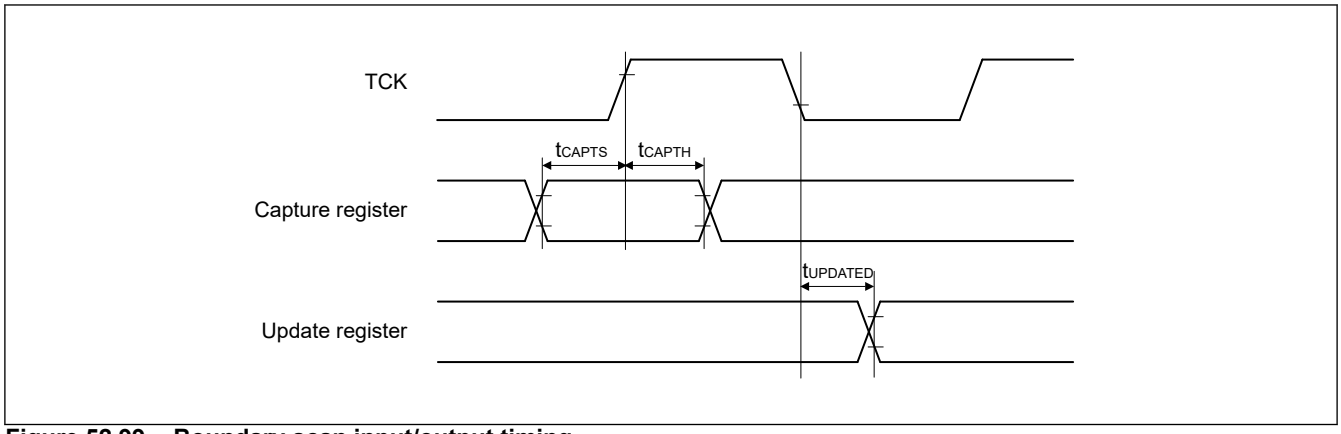


Figure 52.99 Boundary scan input/output timing

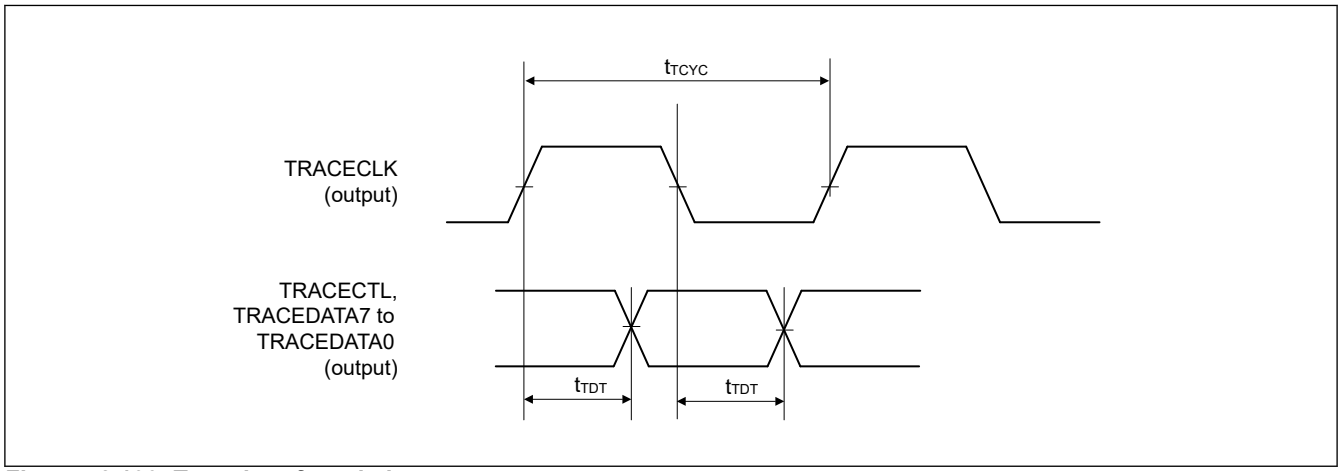


Figure 52.100 Trace interface timing

Appendix 1. Peripheral I/O Registers

This appendix describes pheripheral I/O register addresses, access type, and reset values by function.

1.1 Peripheral Base Addresses

This section provides the base addresses for peripherals described in this manual.

Table 1.1 shows the name, description, and the base address of each peripheral.

Table 1.1 Peripheral base address (1 of 4)

Name	Description	Base address
GPT7	General PWM Timer 7	0x80000000
GPT8	General PWM Timer 8	0x80000100
GPT9	General PWM Timer 9	0x80000200
GPT10	General PWM Timer 10	0x80000300
GPT11	General PWM Timer 11	0x80000400
GPT12	General PWM Timer 12	0x80000500
GPT13	General PWM Timer 13	0x80000600
SCI0	Serial Communication Interface 0	0x80001000
SCI1	Serial Communication Interface 1	0x80001400
SCI2	Serial Communication Interface 2	0x80001800
SCI3	Serial Communication Interface 3	0x80001C00
SCI4	Serial Communication Interface 4	0x80002000
SPI0	Serial Peripheral Interface 0	0x80003000
SPI1	Serial Peripheral Interface 1	0x80003400
SPI2	Serial Peripheral Interface 2	0x80003800
CRC0	CRC Unit 0	0x80004000
CANFD	CAN-FD	0x80020000
CMTC	Compare Match Timer Control	0x80040000
CMT0	Compare Match Timer 0	0x80040002
CMT1	Compare Match Timer 1	0x80040008
CMT2	Compare Match Timer 2	0x80040402
CMT3	Compare Match Timer 3	0x80040408
CMT4	Compare Match Timer 4	0x80040802
CMT5	Compare Match Timer 5	0x80040808
CMTW0	Compare Match Timer W	0x80041000
CMTW1	Compare Match Timer W	0x80041400
WDT0	Watchdog Timer 0	0x80042000
IIC0	I2C Bus Interface 0	0x80043000
IIC1	I2C Bus Interface 1	0x80043400
DOC	Data Operation Circuit	0x80044000
TSU	Temperature Sensor Unit	0x80046000
POEG1	GPT Port Output Enable 1	0x80047000
DMAC0	DMA Controller 0	0x80080000
DMAC1	DMA Controller 1	0x80081000
ICU_NS	Interrupt Controller in Non Safety Domain	0x80090000

Table 1.1 Peripheral base address (2 of 4)

Name	Description	Base address
ELC	Evnet Link Controller	0x80090010
DMA	DMAC Configuration	0x80090060
PORT_NSR	I/O Ports (Non safety region)	0x800A0000
GMAC	Ethernet MAC	0x80100000
ETHSS	Ethernet Subsystem	0x80110000
ESC_INI	Initial Configuration 1 for EtherCAT Slave Controller	0x80110200
GMAC_PTP	GMAC for PTP	0x80110400
ESC	EtherCAT Slave Controller	0x80130000
USBHC	USB 2.0 HS Host Module	0x80200000
USBF	USB 2.0 Host and Function Module	0x80201000
BSC	Bus State Controller	0x80210000
XSPI0	xSPI	0x80220000
XSPI1	xSPI	0x80221000
MBXSEM	Mailbox and Semaphore	0x80240000
SHOSTIF	Serial Host Interface	0x80241000
SYSC_NS	System Control for Non-safety region	0x80280000
ELO	Evnet Link Option Setting	0x80281200
ENCSS	Encoder Interface Subsystem	0x80281400
RWP_NS	Register Write Protection for Non-safety Area	0x80281A10
GPT14	General PWM Timer 14	0x81000000
GPT15	General PWM Timer 15	0x81000100
GPT16	General PWM Timer 16	0x81000200
GPT17	General PWM Timer 17	0x81000300
SCI5	Serial Communication Interface 5	0x81001000
SPI3	Serial Peripheral Interface 3	0x81002000
CRC1	CRC Unit 1	0x81003000
IIC2	I2C Bus Interface 2	0x81008000
RTC	Real Time Clock	0x81009000
POEG2	GPT Port Output Enable 2	0x8100A000
OTP	One-Time Programmable Memory	0x81028000
PORT_SR	I/O Ports (Safety region)	0x81030000
PTADR	Port Address Selection	0x81030C00
SYSRAM0	System SRAM 0	0x81040000
SYSRAM1	System SRAM 1	0x81041000
ICU	Interrupt Controller	0x81048000
SYSC_S	Register Write Protection for Safety Area	0x81280000
CLMA0	Clock Monitor Circuit 0	0x81280800
CLMA1	Clock Monitor Circuit 1	0x81280820
CLMA2	Clock Monitor Circuit 2	0x81280840
CLMA3	Clock Monitor Circuit 3	0x81280860
MPU0	Master MPU 0	0x81281100
MPU1	Master MPU 1	0x81281200

Table 1.1 Peripheral base address (3 of 4)

Name	Description	Base address
MPU2	Master MPU 2	0x81281300
MPU3	Master MPU 3	0x81281400
MPU4	Master MPU 4	0x81281500
MPU6	Master MPU 6	0x81281700
SYSRAM_CTL	System SRAM Control	0x81281800
TCMAW	TCM access wait state	0x81281900
SHOSTIF_CFG	Serial Host Interface Configuration	0x81281920
RWP_S	Register Write Protection for Safety Area	0x81281A00
MPU7	Master MPU 7	0x81281C00
MTU	Multi-Function Timer Pulse Unit	0x90001000
MTU3	Multi-Function Timer Pulse Unit Channel 3	0x90001100
MTU4	Multi-Function Timer Pulse Unit Channel 4	0x90001200
MTU_NF	Multi-Function Timer Pulse Unit Noise Filter	0x90001290
MTU0	Multi-Function Timer Pulse Unit Channel 0	0x90001300
MTU1	Multi-Function Timer Pulse Unit Channel 1	0x90001380
MTU2	Multi-Function Timer Pulse Unit Channel 2	0x90001400
MTU8	Multi-Function Timer Pulse Unit Channel 8	0x90001600
MTU6	Multi-Function Timer Pulse Unit Channel 6	0x90001900
MTU7	Multi-Function Timer Pulse Unit Channel 7	0x90001A00
MTU5	Multi-Function Timer Pulse Unit Channel 5	0x90001C00
GPT0	General PWM Timer 0	0x90002000
GPT1	General PWM Timer 1	0x90002100
GPT2	General PWM Timer 2	0x90002200
GPT3	General PWM Timer 3	0x90002300
GPT4	General PWM Timer 4	0x90002400
GPT5	General PWM Timer 5	0x90002500
GPT6	General PWM Timer 6	0x90002600
TFU	Trigonometric Function Unit	0x90003000
ADC120	12-Bit A/D converter	0x90004000
ADC121	12-Bit A/D converter	0x90004800
POE3	Port Output Enable 3	0x90005000
POEG0	GPT Port Output Enable 0	0x90006000
DSMIF0	Delta-sigma Interface 0	0x90008000
DSMIF1	Delta-sigma Interface 1	0x90008800
HDSLD0	HIPERFACE DSL for Drive Interface Unit 0	0x90040000
HDSLS10	HIPERFACE DSL for Safe Channel 1 Unit 0	0x90041000
BISS0	BiSS-C Unit 0	0x90042000
ENDAT0	EnDat2.2 Unit 0	0x90043000
AFMT0	A-format Unit 0	0x90045000
HDSLD1	HIPERFACE DSL for Drive Interface Unit 1	0x90050000
HDSLS11	HIPERFACE DSL for Safe Channel 1 Unit 1	0x90051000
BISS1	BiSS-C Unit 1	0x90052000

Table 1.1 Peripheral base address (4 of 4)

Name	Description	Base address
ENDAT1	EnDat2.2 Unit 1	0x90053000
AFMT1	A-format Unit 1	0x90055000
ENCOUT	Encoder Divided-Output Module	0x90060000
GSC	Global System Counter	0xC0060000

Note: Name = Peripheral name
Description = Peripheral functionality
Base address = Lowest reserved address or address used by the peripheral

1.2 Register Descriptions

This section provides information associated with registers described in this manual.

Table 1.2 shows a list of registers including address offsets, address sizes, access rights, and reset values.

Table 1.2 Register description (1 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
GPT7-13	—	—	—	GTWP	General PWM Timer Write-Protection Register	0x00	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTSTR	General PWM Timer Software Start Register	0x04	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTSTP	General PWM Timer Software Stop Register	0x08	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT7-13	—	—	—	GTCLR	General PWM Timer Software Clear Register	0x0C	32	W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTSSR	General PWM Timer Start Source Select Register	0x10	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTPSR	General PWM Timer Stop Source Select Register	0x14	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTCSR	General PWM Timer Clear Source Select Register	0x18	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTUPSR	General PWM Timer Count-Up Source Select Register	0x1C	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTDNSR	General PWM Timer Count-Down Source Select Register	0x20	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTICASR	General PWM Timer Input Capture Source Select Register A	0x24	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTICBSR	General PWM Timer Input Capture Source Select Register B	0x28	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTCR	General PWM Timer Control Register	0x2C	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTUDDTYC	General PWM Timer Count Direction and Duty Setting Register	0x30	32	R/W	0x00000001	0xFFFFFFFF
GPT7-13	—	—	—	GTIOR	General PWM Timer I/O Control Register	0x34	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTINTAD	General PWM Timer Interrupt Output Setting Register	0x38	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTST	General PWM Timer Status Register	0x3C	32	R/W	0x00008000	0xFFFFFFFF
GPT7-13	—	—	—	GTBER	General PWM Timer Buffer Enable Register	0x40	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTITC	General PWM Timer Interrupt and A/D Converter Start Request Skipping Setting Register	0x44	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTCNT	General PWM Timer Counter	0x48	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTCCRA	General PWM Timer Compare Capture Register A (m = A to F)	0x4C	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT7-13	—	—	—	GTCCRB	General PWM Timer Compare Capture Register B (m = A to F)	0x50	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT7-13	—	—	—	GTCCRC	General PWM Timer Compare Capture Register C (m = A to F)	0x54	32	R/W	0xFFFFFFFF	0xFFFFFFFF

Table 1.2 Register description (2 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
GPT7-13	—	—	—	GTCCRE	General PWM Timer Compare Capture Register E (m = A to F)	0x58	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT7-13	—	—	—	GTCCRD	General PWM Timer Compare Capture Register D (m = A to F)	0x5C	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT7-13	—	—	—	GTCCRF	General PWM Timer Compare Capture Register F (m = A to F)	0x60	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT7-13	—	—	—	GTPR	General PWM Timer Cycle Setting Register	0x64	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT7-13	—	—	—	GTPBR	General PWM Timer Cycle Setting Buffer Register	0x68	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT7-13	—	—	—	GTPDBR	General PWM Timer Cycle Setting Double-Buffer Register	0x6C	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT7-13	—	—	—	GTADTRA	A/D Converter Start Request Timing Register A (m = A, B)	0x70	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT7-13	—	—	—	GTADTBRA	A/D Converter Start Request Timing Buffer Register A (m = A, B)	0x74	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT7-13	—	—	—	GTADTDBRA	A/D Converter Start Request Timing Double-Buffer Register A	0x78	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT7-13	—	—	—	GTADTRB	A/D Converter Start Request Timing Register B (m = A, B)	0x7C	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT7-13	—	—	—	GTADTBRB	A/D Converter Start Request Timing Buffer Register B (m = A, B)	0x80	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT7-13	—	—	—	GTADTDBRB	A/D Converter Start Request Timing Double-Buffer Register B	0x84	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT7-13	—	—	—	GTDTCR	General PWM Timer Dead Time Control Register	0x88	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTDVU	General PWM Timer Dead Time Value Register U (m = U, D)	0x8C	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT7-13	—	—	—	GTDVD	General PWM Timer Dead Time Value Register D (m = U, D)	0x90	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT7-13	—	—	—	GTDBU	General PWM Timer Dead Time Value Buffer Register U (m = U, D)	0x94	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT7-13	—	—	—	GTDBD	General PWM Timer Dead Time Value Buffer Register D (m = U, D)	0x98	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT7-13	—	—	—	GTSOS	General PWM Timer Output Protection Function Status Register	0x9C	32	R	0x00000000	0xFFFFFCFF
GPT7-13	—	—	—	GTSOTR	General PWM Timer Output Protection Function Temporary Release Register	0xA0	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTADSMR	General PWM Timer A/D Conversion Start Request Signal Monitoring Register	0xA4	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTEITC	General PWM Timer Extended Interrupt Skipping Counter Control Register	0xA8	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTEITLI1	General PWM Timer Extended Interrupt Skipping Setting Register 1	0xAC	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTEITLI2	General PWM Timer Extended Interrupt Skipping Setting Register 2	0xB0	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTEITLB	General PWM Timer Extended Buffer Transfer Skipping Setting Register	0xB4	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTSECSR	General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register	0xD0	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTSECR	General PWM Timer Operation Enable Bit Simultaneous Control Register	0xD4	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTSWSR	General PWM Timer Switch Source Select Register	0xD8	32	R/W	0x00000000	0xFFFFFFFF
GPT7-13	—	—	—	GTSWOS	General PWM Timer Switch Offset Setting Register	0xDC	32	R/W	0x00000000	0xFFFFFFFF
SCI0-4	—	—	—	RDR	Receive Data Register	0x00	32	R	0x00000000	0xFFFFFFFF
SCI0-4	—	—	—	TDR	Transmit Data Register	0x04	32	R/W	0xFFFFFFFF	0xFFFFFFFF

Table 1.2 Register description (3 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
SCI0-4	—	—	—	CCR0	Common Control Register 0	0x08	32	R/W	0x00000000	0xFFFFFFFF
SCI0-4	—	—	—	CCR1	Common Control Register 1	0x0C	32	R/W	0x00000010	0xFFFFFFFF
SCI0-4	—	—	—	CCR2	Common Control Register 2	0x10	32	R/W	0xFF00FF04	0xFFFFFFFF
SCI0-4	—	—	—	CCR3	Common Control Register 3	0x14	32	R/W	0x00001203	0xFFFFFFFF
SCI0-4	—	—	—	CCR4	Common Control Register 4	0x18	32	R/W	0x00000000	0xFFFFFFFF
SCI0-4	—	—	—	ICR	Simple I2C Control Register	0x20	32	R/W	0x00000000	0xFFFFFFFF
SCI0-4	—	—	—	FCR	FIFO Control Register	0x24	32	R/W	0x1F1F0000	0xFFFFFFFF
SCI0-4	—	—	—	DCR	Driver Control Register	0x30	32	R/W	0x00000000	0xFFFFFFFF
SCI0-4	—	—	—	CSR	Common Status Register	0x48	32	R	0x60008000	0xFFFFFFFF
SCI0-4	—	—	—	ISR	Simple I2C Status Register	0x4C	32	R	0x00000000	0xFFFFFFCB
SCI0-4	—	—	—	FRSR	FIFO Receive Status Register	0x50	32	R	0x00000000	0xFFFFFFFF
SCI0-4	—	—	—	FTSR	FIFO Transmit Status Register	0x54	32	R	0x00000000	0xFFFFFFFF
SCI0-4	—	—	—	CFCLR	Common Flag Clear Register	0x68	32	W	0x00000000	0xFFFFFFFF
SCI0-4	—	—	—	ICFCLR	Simple I2C Flag Clear Register	0x6C	32	W	0x00000000	0xFFFFFFFF
SCI0-4	—	—	—	FFCLR	FIFO Flag Clear Register	0x70	32	W	0x00000000	0xFFFFFFFF
SPI0-2	—	—	—	SPDR	SPI Data Register	0x00	32	R/W	0x00000000	0xFFFFFFFF
SPI0-2	—	—	—	SPDR_HA	SPI Data Register	0x00	16	R/W	0x0000	0xFFFF
SPI0-2	—	—	—	SPDR_BY	SPI Data Register	0x00	8	R/W	0x00	0xFF
SPI0-2	—	—	—	SPCKD	SPI Clock Delay Register	0x04	8	R/W	0x00	0xFF
SPI0-2	—	—	—	SSLND	SPI Slave Select Negation Delay Register	0x05	8	R/W	0x00	0xFF
SPI0-2	—	—	—	SPND	SPI Next-Access Delay Register	0x06	8	R/W	0x00	0xFF
SPI0-2	—	—	—	MRCKD	SPI ClocK Digital control Register for Master Receive	0x07	8	R/W	0x00	0xFF
SPI0-2	—	—	—	SPCR	SPI Control Register	0x08	32	R/W	0x00000000	0xFFFFFFFF
SPI0-2	—	—	—	SPCRRM	SPI Control Register for Master Receive only	0x0C	8	R/W	0x00	0xFF
SPI0-2	—	—	—	SPDRCR	SPI Control Register for Received Data Ready Detection	0x0D	8	R/W	0x00	0xFF
SPI0-2	—	—	—	SPPCR	SPI Pin Control Register	0x0E	8	R/W	0x00	0xFF
SPI0-2	—	—	—	SPCR2	SPI Control Register 2	0x0F	8	R/W	0x00	0xFF
SPI0-2	—	—	—	SSLP	SPI Slave Select Polarity Register	0x10	8	R/W	0x00	0xFF
SPI0-2	—	—	—	SPBR	SPI Bit Rate Register	0x11	8	R/W	0xFF	0xFF
SPI0-2	—	—	—	SPSCR	SPI Sequence Control Register	0x13	8	R/W	0x00	0xFF
SPI0-2	8	0x04	0-7	SPCMD%s	SPI Command Register %s (m = 0 to 7)	0x14	32	R/W	0x00070000	0xFFFFFFFF
SPI0-2	—	—	—	SPDCR	SPI Data Control Register	0x40	16	R/W	0x0000	0xFFFF
SPI0-2	—	—	—	SPDCR2	SPI Data Control Register 2	0x44	16	R/W	0x0000	0xFFFF
SPI0-2	—	—	—	SPSSR	SPI Sequence Status Register	0x51	8	R	0x00	0xFF
SPI0-2	—	—	—	SPSR	SPI Status Register	0x52	16	R	0x2000	0xFFFF
SPI0-2	—	—	—	SPTFSR	SPI Transfer FIFO Status Register	0x58	8	R	0x04	0xFF
SPI0-2	—	—	—	SPRFSR	SPI Receive FIFO Status Register	0x5C	8	R	0x00	0xFF
SPI0-2	—	—	—	SPPSR	SPI Poling Register	0x60	32	R	0x00000000	0xFFFFFFFF
SPI0-2	—	—	—	SPSRC	SPI Status Clear Register	0x6A	16	R/W	0x0000	0xFFFF
SPI0-2	—	—	—	SPFCR	SPI FIFO Clear Register	0x6C	8	W	0x00	0xFF
CRC0	—	—	—	CRCCR0	CRC Control Register 0	0x0	8	R/W	0x00	0xFF
CRC0	—	—	—	CRCDIR	CRC Data Input Register	0x4	32	R/W	0x00000000	0xFFFFFFFF
CRC0	—	—	—	CRCDIR_BY	CRC Data Input Register	0x4	8	R/W	0x00	0xFF
CRC0	—	—	—	CRCDOR	CRC Data Output Register	0x8	32	R/W	0x00000000	0xFFFFFFFF

Table 1.2 Register description (4 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
CRC0	—	—	—	CRCDOR_HA	CRC Data Output Register	0x8	16	R/W	0x0000	0xFFFF
CRC0	—	—	—	CRCDOR_BY	CRC Data Output Register	0x8	8	R/W	0x00	0xFF
CANFD	2	0x10	0-1	CFDC%sNCFG	Channel %s Nominal Bit Rate Configuration Register	0x0000	32	R/W	0x00000000	0xFFFFFFFF
CANFD	2	0x10	0-1	CFDC%sCTR	Channel %s Control Register	0x0004	32	R/W	0x00000005	0xFFFFFFFF
CANFD	2	0x10	0-1	CFDC%sSTS	Channel %s Status Register	0x0008	32	R/W	0x00000005	0xFFFFFFFF
CANFD	2	0x10	0-1	CFDC%sERFL	Channel %s Error Flag Register	0x000C	32	R/W	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDGIPV	Global IP Version Register	0x0080	32	R	0x31240042	0xFFFFFEFF
CANFD	—	—	—	CFDGCFG	Global Configuration Register	0x0084	32	R/W	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDGCTR	Global Control Register	0x0088	32	R/W	0x00000005	0xFFFFFFFF
CANFD	—	—	—	CFDGSTS	Global Status Register	0x008C	32	R	0x0000000D	0xFFFFFFFF
CANFD	—	—	—	CFDGERFL	Global Error Flag Register	0x0090	32	R/W	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDGTSC	Global Timestamp Counter Register	0x0094	32	R	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDGAFLECTR	Global Acceptance Filter List Entry Control Register	0x0098	32	R/W	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDGAFLCFG0	Global Acceptance Filter List Configuration Register 0	0x009C	32	R/W	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDRMNB	RX Message Buffer Number Register	0x00AC	32	R/W	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDRMND0	RX Message Buffer New Data Register 0	0x00B0	32	R/W	0x00000000	0xFFFFFFFF
CANFD	8	0x04	0-7	CFDRFCC%s	RX FIFO Configuration/Control Register %s	0x00C0	32	R/W	0x00000000	0xFFFFFFFF
CANFD	8	0x04	0-7	CFDRFSTS%s	RX FIFO Status Register %s	0x00E0	32	R/W	0x00000001	0xFFFFFFFF
CANFD	8	0x04	0-7	CFDRFPCTR%s	RX FIFO Pointer Control Register %s	0x0100	32	R/W	0x00000000	0xFFFFFFFF
CANFD	6	0x04	0-5	CFDCFCC%s	Common FIFO Configuration/Control Register %s	0x0120	32	R/W	0x00000000	0xFFFFFFFF
CANFD	6	0x04	0-5	CFDCFCCE%s	Common FIFO Configuration/Control Enhancement Register %s	0x0180	32	R/W	0x00000000	0xFFFFFFFF
CANFD	6	0x04	0-5	CFDCFSTS%s	Common FIFO Status Register %s	0x01E0	32	R/W	0x00000001	0xFFFFFFFF
CANFD	6	0x04	0-5	CFDCFPCTR%s	Common FIFO Pointer Control Register %s	0x0240	32	W	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDFESTS	FIFO Empty Status Register	0x02A0	32	R	0x00003FFF	0xFFFFFFFF
CANFD	—	—	—	CFDFFSTS	FIFO Full Status Register	0x02A4	32	R	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDFMSTS	FIFO Message Lost Status Register	0x02A8	32	R	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDRFISTS	RX FIFO Interrupt Flag Status Register	0x02AC	32	R	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDCFRISTS	Common FIFO RX Interrupt Flag Status Register	0x02B0	32	R	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDCFTISTS	Common FIFO TX Interrupt Flag Status Register	0x02B4	32	R	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDCFOFRISTS	Common FIFO One Frame RX Interrupt Flag Status Register	0x02B8	32	R	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDCFOFTISTS	Common FIFO One Frame TX Interrupt Flag Status Register	0x02BC	32	R	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDCFMOWSTS	Common FIFO Message Overwrite Status Register	0x02C0	32	R	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDFFFSTS	FIFO FDC Full Status Register	0x02C4	32	R	0x00000000	0xFFFFFFFF
CANFD	128	0x01	0-127	CFDTMC%s	TX Message Buffer Control Register %s	0x02D0	8	R/W	0x00	0xFF
CANFD	128	0x01	0-127	CFDTMSTS%s	TX Message Buffer Status Register %s	0x07D0	8	R/W	0x00	0xFF
CANFD	4	0x04	0-3	CFDTMTRSTS%s	TX Message Buffer Transmission Request Status Register %s	0x0CD0	32	R	0x00000000	0xFFFFFFFF
CANFD	4	0x04	0-3	CFDTMTARSTS%s	TX Message Buffer Transmission Abort Request Status Register %s	0x0D70	32	R	0x00000000	0xFFFFFFFF
CANFD	4	0x04	0-3	CFDTMTCSTS%s	TX Message Buffer Transmission Completion Status Register %s	0x0E10	32	R	0x00000000	0xFFFFFFFF

Table 1.2 Register description (5 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
CANFD	4	0x04	0-3	CFDTMTASTS%s	TX Message Buffer Transmission Abort Status Register %s	0x0EB0	32	R	0x00000000	0xFFFFFFFF
CANFD	4	0x04	0-3	CFDTMIEC%s	TX Message Buffer Transmission Interrupt Enable Register %s	0x0F50	32	R/W	0x00000000	0xFFFFFFFF
CANFD	2	0x04	0-1	CFDTXQCC0%s	TX Queue Configuration/Control Register 0%s	0x1000	32	R/W	0x00000000	0xFFFFFFFF
CANFD	2	0x04	0-1	CFDTXQSTS0%s	TX Queue Status Register 0%s	0x1020	32	R/W	0x00000001	0xFFFFFFFF
CANFD	2	0x04	0-1	CFDTXQPCTR0%s	TX Queue Pointer Control Register 0%s	0x1040	32	W	0x00000000	0xFFFFFFFF
CANFD	2	0x04	0-1	CFDTXQCC1%s	TX Queue Configuration/Control Register 1%s	0x1060	32	R/W	0x00000000	0xFFFFFFFF
CANFD	2	0x04	0-1	CFDTXQSTS1%s	TX Queue Status Register 1%s	0x1080	32	R/W	0x00000001	0xFFFFFFFF
CANFD	2	0x04	0-1	CFDTXQPCTR1%s	TX Queue Pointer Control Register 1%s	0x10A0	32	W	0x00000000	0xFFFFFFFF
CANFD	2	0x04	0-1	CFDTXQCC2%s	TX Queue Configuration/Control Register 2%s	0x10C0	32	R/W	0x00000000	0xFFFFFFFF
CANFD	2	0x04	0-1	CFDTXQSTS2%s	TX Queue Status Register 2%s	0x10E0	32	R/W	0x00000001	0xFFFFFFFF
CANFD	2	0x04	0-1	CFDTXQPCTR2%s	TX Queue Pointer Control Register 2%s	0x1100	32	W	0x00000000	0xFFFFFFFF
CANFD	2	0x04	0-1	CFDTXQCC3%s	TX Queue Configuration/Control Register 3%s	0x1120	32	R/W	0x00000000	0xFFFFFFFF
CANFD	2	0x04	0-1	CFDTXQSTS3%s	TX Queue Status Register 3%s	0x1140	32	R/W	0x00000001	0xFFFFFFFF
CANFD	2	0x04	0-1	CFDTXQPCTR3%s	TX Queue Pointer Control Register 3%s	0x1160	32	W	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDTXQESTS	TX Queue Empty Status Register	0x1180	32	R	0x000000FF	0xFFFFFFFF
CANFD	—	—	—	CFDTXQFISTS	TX Queue Full Interrupt Status Register	0x1184	32	R	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDTXQMSTS	TX Queue Message Lost Status Register	0x1188	32	R	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDTXQISTS	TX Queue Interrupt Status Register	0x1190	32	R	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDTXQOFTISTS	TX Queue One Frame TX Interrupt Status Register	0x1194	32	R	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDTXQOFRISTS	TX Queue One Frame RX Interrupt Status Register	0x1198	32	R	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDTXQFSTS	TX Queue Full Status Register	0x119C	32	R	0x00000000	0xFFFFFFFF
CANFD	2	0x04	0-1	CFDTHLCC%s	TX History List Configuration/Control Register %s	0x1200	32	R/W	0x00000000	0xFFFFFFFF
CANFD	2	0x04	0-1	CFDTHLSTS%s	TX History List Status Register %s	0x1220	32	R/W	0x00000001	0xFFFFFFFF
CANFD	2	0x04	0-1	CFDTHLPCTR%s	TX History List Pointer Control Register %s	0x1240	32	W	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDGTINTSTS0	Global TX Interrupt Status Register 0	0x1300	32	R	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDGTSTCFG	Global Test Configuration Register	0x1308	32	R/W	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDGTSTCTR	Global Test Control Register	0x130C	32	R/W	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDGFDCFG	Global FD Configuration Register	0x1314	32	R/W	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDGLOCKK	Global Lock Key Register	0x131C	32	W	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDCDTCT	DMA Transfer Control Register	0x1330	32	R/W	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDCDTSTS	DMA Transfer Status Register	0x1334	32	R	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDCDTTCT	DMA TX Transfer Control Register	0x1340	32	R/W	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDCDTTSTS	DMA TX Transfer Status Register	0x1344	32	R	0x00000000	0xFFFFFFFF
CANFD	2	0x04	0-1	CFDGRINTSTS%s	Global RX Interrupt Status Register %s	0x1350	32	R	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDGRSTC	Global Reset Control Register	0x1380	32	R/W	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDGFCMC	Global Flexible CAN Mode Configuration Register	0x1384	32	R/W	0x00000000	0xFFFFFFFF
CANFD	—	—	—	CFDGFTBAC	Global Flexible Transmission Buffer Assignment Configuration Register	0x138C	32	R/W	0x00000000	0xFFFFFFFF
CANFD	2	0x20	0-1	CFDC%sDCFG	Channel %s Data Bit Rate Configuration Register	0x1400	32	R/W	0x00000000	0xFFFFFFFF
CANFD	2	0x20	0-1	CFDC%sFDCFG	Channel %s CAN-FD Configuration Register	0x1404	32	R/W	0x00000000	0xFFFFFFFF

Table 1.2 Register description (6 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
CANFD	2	0x20	0-1	CFDC%sFDCTR	Channel %s CAN-FD Control Register	0x1408	32	R/W	0x00000000	0xFFFFFFFF
CANFD	2	0x20	0-1	CFDC%sFDSTS	Channel %s CAN-FD Status Register	0x140C	32	R/W	0x00000000	0xFFFFFFFF
CANFD	2	0x20	0-1	CFDC%sFDCRC	Channel %s CAN-FD CRC Register	0x1410	32	R	0x00000000	0xFFFFFFFF
CANFD	2	0x20	0-1	CFDC%sBLCT	Channel %s Bus Load Control Register	0x1418	32	R/W	0x00000000	0xFFFFFFFF
CANFD	2	0x20	0-1	CFDC%sBLSTS	Channel %s Bus Load Status Register	0x141C	32	R	0x00000000	0xFFFFFFFF
CANFD	16	0x10	1-16	CFDGAFLID%s	Global Acceptance Filter List ID Register %s	0x1800	32	R/W	0x00000000	0xFFFFFFFF
CANFD	16	0x10	1-16	CFDGAFLM%s	Global Acceptance Filter List Mask Register %s	0x1804	32	R/W	0x00000000	0xFFFFFFFF
CANFD	16	0x10	1-16	CFDGAFLP0%s	Global Acceptance Filter List Pointer 0 Register %s	0x1808	32	R/W	0x00000000	0xFFFFFFFF
CANFD	16	0x10	1-16	CFDGAFLP1%s	Global Acceptance Filter List Pointer 1 Register %s	0x180C	32	R/W	0x00000000	0xFFFFFFFF
CANFD	32	0x080	0-31	CFDRMID%s	RX Message Buffer ID Register %s	0x2000	32	R	0x00000000	0xFFFFFFFF
CANFD	32	0x080	0-31	CFDRMPTR%s	RX Message Buffer Pointer Register	0x2004	32	R/W	0x00000000	0xFFFFFFFF
CANFD	32	0x080	0-31	CFDRMFDSTS%s	RX Message Buffer CAN-FD Status Register	0x2008	32	R/W	0x00000000	0xFFFFFFFF
CANFD	32	0x080	0-31	CFDRMDF0_%s	RX Message Buffer Data Field 0 Register %s	0x200C	32	R	0x00000000	0xFFFFFFFF
CANFD	32	0x080	0-31	CFDRMDF1_%s	RX Message Buffer Data Field 1 Register %s	0x2010	32	R	0x00000000	0xFFFFFFFF
CANFD	32	0x080	0-31	CFDRMDF2_%s	RX Message Buffer Data Field 2 Register %s	0x2014	32	R	0x00000000	0xFFFFFFFF
CANFD	32	0x080	0-31	CFDRMDF3_%s	RX Message Buffer Data Field 3 Register %s	0x2018	32	R	0x00000000	0xFFFFFFFF
CANFD	32	0x080	0-31	CFDRMDF4_%s	RX Message Buffer Data Field 4 Register %s	0x201C	32	R	0x00000000	0xFFFFFFFF
CANFD	32	0x080	0-31	CFDRMDF5_%s	RX Message Buffer Data Field 5 Register %s	0x2020	32	R	0x00000000	0xFFFFFFFF
CANFD	32	0x080	0-31	CFDRMDF6_%s	RX Message Buffer Data Field 6 Register %s	0x2024	32	R	0x00000000	0xFFFFFFFF
CANFD	32	0x080	0-31	CFDRMDF7_%s	RX Message Buffer Data Field 7 Register %s	0x2028	32	R	0x00000000	0xFFFFFFFF
CANFD	32	0x080	0-31	CFDRMDF8_%s	RX Message Buffer Data Field 8 Register %s	0x202C	32	R	0x00000000	0xFFFFFFFF
CANFD	32	0x080	0-31	CFDRMDF9_%s	RX Message Buffer Data Field 9 Register %s	0x2030	32	R	0x00000000	0xFFFFFFFF
CANFD	32	0x080	0-31	CFDRMDF10_%s	RX Message Buffer Data Field 10 Register %s	0x2034	32	R	0x00000000	0xFFFFFFFF
CANFD	32	0x080	0-31	CFDRMDF11_%s	RX Message Buffer Data Field 11 Register %s	0x2038	32	R	0x00000000	0xFFFFFFFF
CANFD	32	0x080	0-31	CFDRMDF12_%s	RX Message Buffer Data Field 12 Register %s	0x203C	32	R	0x00000000	0xFFFFFFFF
CANFD	32	0x080	0-31	CFDRMDF13_%s	RX Message Buffer Data Field 13 Register %s	0x2040	32	R	0x00000000	0xFFFFFFFF
CANFD	32	0x080	0-31	CFDRMDF14_%s	RX Message Buffer Data Field 14 Register %s	0x2044	32	R	0x00000000	0xFFFFFFFF
CANFD	32	0x080	0-31	CFDRMDF15_%s	RX Message Buffer Data Field 15 Register %s	0x2048	32	R	0x00000000	0xFFFFFFFF
CANFD	8	0x080	0-7	CFDRFID%s	RX FIFO Access ID Register %s	0x6000	32	R	0x00000000	0xFFFFFFFF
CANFD	8	0x080	0-7	CFDRFPTR%s	RX FIFO Access Pointer Register %s	0x6004	32	R	0x00000000	0xFFFFFFFF
CANFD	8	0x080	0-7	CFDRFFDSTS%s	RX FIFO Access CAN-FD Status Register %s	0x6008	32	R	0x00000000	0xFFFFFFFF
CANFD	8	0x080	0-7	CFDRFDF0%s	RX FIFO Access Data Field 0 Register %s	0x600C	32	R	0x00000000	0xFFFFFFFF
CANFD	8	0x080	0-7	CFDRFDF6%s	RX FIFO Access Data Field 6 Register %s	0x600C	32	R	0x00000000	0xFFFFFFFF

Table 1.2 Register description (7 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
CANFD	8	0x080	0-7	CFDRFDF7%s	RX FIFO Access Data Field 7 Register %s	0x600C	32	R	0x00000000	0xFFFFFFFF
CANFD	8	0x080	0-7	CFDRFDF1%s	RX FIFO Access Data Field 1 Register %s	0x6010	32	R	0x00000000	0xFFFFFFFF
CANFD	8	0x080	0-7	CFDRFDF2%s	RX FIFO Access Data Field 2 Register %s	0x6014	32	R	0x00000000	0xFFFFFFFF
CANFD	8	0x080	0-7	CFDRFDF3%s	RX FIFO Access Data Field 3 Register %s	0x6018	32	R	0x00000000	0xFFFFFFFF
CANFD	8	0x080	0-7	CFDRFDF4%s	RX FIFO Access Data Field 4 Register %s	0x601C	32	R	0x00000000	0xFFFFFFFF
CANFD	8	0x080	0-7	CFDRFDF5%s	RX FIFO Access Data Field 5 Register %s	0x6020	32	R	0x00000000	0xFFFFFFFF
CANFD	8	0x080	0-7	CFDRFDF8%s	RX FIFO Access Data Field 8 Register %s	0x602C	32	R	0x00000000	0xFFFFFFFF
CANFD	8	0x080	0-7	CFDRFDF9%s	RX FIFO Access Data Field 9 Register %s	0x6030	32	R	0x00000000	0xFFFFFFFF
CANFD	8	0x080	0-7	CFDRFDF10%s	RX FIFO Access Data Field 10 Register %s	0x6034	32	R	0x00000000	0xFFFFFFFF
CANFD	8	0x080	0-7	CFDRFDF11%s	RX FIFO Access Data Field 11 Register 11	0x6038	32	R	0x00000000	0xFFFFFFFF
CANFD	8	0x080	0-7	CFDRFDF12%s	RX FIFO Access Data Field 12 Register %s	0x603C	32	R	0x00000000	0xFFFFFFFF
CANFD	8	0x080	0-7	CFDRFDF13%s	RX FIFO Access Data Field 13 Register %s	0x6040	32	R	0x00000000	0xFFFFFFFF
CANFD	8	0x080	0-7	CFDRFDF14%s	RX FIFO Access Data Field 14 Register %s	0x6044	32	R	0x00000000	0xFFFFFFFF
CANFD	8	0x080	0-7	CFDRFDF15%s	RX FIFO Access Data Field 15 Register %s	0x6048	32	R	0x00000000	0xFFFFFFFF
CANFD	6	0x080	0-5	CFDCFID%s	Common FIFO Access ID Register %s	0x6400	32	R/W	0x00000000	0xFFFFFFFF
CANFD	6	0x080	0-5	CFDCFPTR%s	Common FIFO Access Pointer Register %s	0x6404	32	R/W	0x00000000	0xFFFFFFFF
CANFD	6	0x080	0-5	CFDCFFDCSTS%s	Common FIFO Access CAN-FD Control/ Status Register %s	0x6408	32	R/W	0x00000000	0xFFFFFFFF
CANFD	6	0x080	0-5	CFDCFDF0%s	Common FIFO Access Data Field 0 Register %s	0x640C	32	R/W	0x00000000	0xFFFFFFFF
CANFD	6	0x080	0-5	CFDCFDF1%s	Common FIFO Access Data Field 1 Register %s	0x6410	32	R/W	0x00000000	0xFFFFFFFF
CANFD	6	0x080	0-5	CFDCFDF2%s	Common FIFO Access Data Field 2 Register %s	0x6414	32	R/W	0x00000000	0xFFFFFFFF
CANFD	6	0x080	0-5	CFDCFDF3%s	Common FIFO Access Data Field 3 Register %s	0x6418	32	R/W	0x00000000	0xFFFFFFFF
CANFD	6	0x080	0-5	CFDCFDF4%s	Common FIFO Access Data Field 4 Register %s	0x641C	32	R/W	0x00000000	0xFFFFFFFF
CANFD	6	0x080	0-5	CFDCFDF5%s	Common FIFO Access Data Field 5 Register %s	0x6420	32	R/W	0x00000000	0xFFFFFFFF
CANFD	6	0x080	0-5	CFDCFDF6%s	Common FIFO Access Data Field 6 Register %s	0x6424	32	R/W	0x00000000	0xFFFFFFFF
CANFD	6	0x080	0-5	CFDCFDF7%s	Common FIFO Access Data Field 7 Register %s	0x6428	32	R/W	0x00000000	0xFFFFFFFF
CANFD	6	0x080	0-5	CFDCFDF8%s	Common FIFO Access Data Field 8 Register %s	0x642C	32	R/W	0x00000000	0xFFFFFFFF
CANFD	6	0x080	0-5	CFDCFDF9%s	Common FIFO Access Data Field 9 Register %s	0x6430	32	R/W	0x00000000	0xFFFFFFFF
CANFD	6	0x080	0-5	CFDCFDF10%s	Common FIFO Access Data Field 10 Register %s	0x6434	32	R/W	0x00000000	0xFFFFFFFF
CANFD	6	0x080	0-5	CFDCFDF11%s	Common FIFO Access Data Field 11 Register %s	0x6438	32	R/W	0x00000000	0xFFFFFFFF
CANFD	6	0x080	0-5	CFDCFDF12%s	Common FIFO Access Data Field 12 Register %s	0x643C	32	R/W	0x00000000	0xFFFFFFFF

Table 1.2 Register description (8 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
CANFD	6	0x080	0-5	CFDCFDF13%s	Common FIFO Access Data Field 13 Register %s	0x6440	32	R/W	0x00000000	0xFFFFFFFF
CANFD	6	0x080	0-5	CFDCFDF14%s	Common FIFO Access Data Field 14 Register %s	0x6444	32	R/W	0x00000000	0xFFFFFFFF
CANFD	6	0x080	0-5	CFDCFDF15%s	Common FIFO Access Data Field 15 Register %s	0x6448	32	R/W	0x00000000	0xFFFFFFFF
CANFD	2	0x08	0-1	CFDTHLACC0%s	Channel %s TX History List Access Register 0	0x8000	32	R	0x00000000	0xFFFFFFFF
CANFD	2	0x08	0-1	CFDTHLACC1%s	Channel %s TX History List Access Register 1	0x8004	32	R	0x00000000	0xFFFFFFFF
CANFD	64	0x04	0-63	CFDRPGACC%s	RAM Test Page Access Register %s	0x8400	32	R/W	0x00000000	0xFFFFFFFF
CANFD	128	0x080	0-127	CFDTMID%s	TX Message Buffer ID Register %s	0x10000	32	R/W	0x00000000	0xFFFFFFFF
CANFD	128	0x080	0-127	CFDTMPTR%s	TX Message Buffer Pointer Register %s	0x10004	32	R/W	0x00000000	0xFFFFFFFF
CANFD	128	0x080	0-127	CFDTMFDCTR%s	TX Message Buffer CAN-FD Control Register %s	0x10008	32	R/W	0x00000000	0xFFFFFFFF
CANFD	128	0x080	0-127	CFDTMDF0_%s	TX Message Buffer Data Field 0 Register %s	0x1000C	32	R/W	0x00000000	0xFFFFFFFF
CANFD	128	0x080	0-127	CFDTMDF1_%s	TX Message Buffer Data Field 1 Register %s	0x10010	32	R/W	0x00000000	0xFFFFFFFF
CANFD	128	0x080	0-127	CFDTMDF2_%s	TX Message Buffer Data Field 2 Register %s	0x10014	32	R/W	0x00000000	0xFFFFFFFF
CANFD	128	0x080	0-127	CFDTMDF3_%s	TX Message Buffer Data Field 3 Register %s	0x10018	32	R/W	0x00000000	0xFFFFFFFF
CANFD	128	0x080	0-127	CFDTMDF4_%s	TX Message Buffer Data Field 4 Register %s	0x1001C	32	R/W	0x00000000	0xFFFFFFFF
CANFD	128	0x080	0-127	CFDTMDF5_%s	TX Message Buffer Data Field 5 Register %s	0x10020	32	R/W	0x00000000	0xFFFFFFFF
CANFD	128	0x080	0-127	CFDTMDF6_%s	TX Message Buffer Data Field 6 Register %s	0x10024	32	R/W	0x00000000	0xFFFFFFFF
CANFD	128	0x080	0-127	CFDTMDF7_%s	TX Message Buffer Data Field 7 Register %s	0x10028	32	R/W	0x00000000	0xFFFFFFFF
CANFD	128	0x080	0-127	CFDTMDF8_%s	TX Message Buffer Data Field 8 Register %s	0x1002C	32	R/W	0x00000000	0xFFFFFFFF
CANFD	128	0x080	0-127	CFDTMDF9_%s	TX Message Buffer Data Field 9 Register %s	0x10030	32	R/W	0x00000000	0xFFFFFFFF
CANFD	128	0x080	0-127	CFDTMDF10_%s	TX Message Buffer Data Field 10 Register %s	0x10034	32	R/W	0x00000000	0xFFFFFFFF
CANFD	128	0x080	0-127	CFDTMDF11_%s	TX Message Buffer Data Field 11 Register %s	0x10038	32	R/W	0x00000000	0xFFFFFFFF
CANFD	128	0x080	0-127	CFDTMDF12_%s	TX Message Buffer Data Field 12 Register %s	0x1003C	32	R/W	0x00000000	0xFFFFFFFF
CANFD	128	0x080	0-127	CFDTMDF13_%s	TX Message Buffer Data Field 13 Register %s	0x10040	32	R/W	0x00000000	0xFFFFFFFF
CANFD	128	0x080	0-127	CFDTMDF14_%s	TX Message Buffer Data Field 14 Register %s	0x10044	32	R/W	0x00000000	0xFFFFFFFF
CANFD	128	0x080	0-127	CFDTMDF15_%s	TX Message Buffer Data Field 15 Register %s	0x10048	32	R/W	0x00000000	0xFFFFFFFF
CMTC	—	—	—	CMSTR0	Compare Match Timer Start Register	0x000	16	R/W	0x0000	0xFFFF
CMTC	—	—	—	CMSTR2	Compare Match Timer Start Register	0x400	16	R/W	0x0000	0xFFFF
CMTC	—	—	—	CMSTR4	Compare Match Timer Start Register	0x800	16	R/W	0x0000	0xFFFF
CMT0	—	—	—	CMCR	Compare Match Timer Control Register	0x000	16	R/W	0x0000	0xFF7F
CMT0	—	—	—	CMCNT	Compare Match Timer Counter Register	0x002	16	R/W	0x0000	0xFFFF
CMT0	—	—	—	CMCOR	Compare Match Timer Constant Register	0x004	16	R/W	0x0000	0xFFFF
CMT1	—	—	—	CMCR	Compare Match Timer Control Register	0x000	16	R/W	0x0000	0xFF7F
CMT1	—	—	—	CMCNT	Compare Match Timer Counter Register	0x002	16	R/W	0x0000	0xFFFF
CMT1	—	—	—	CMCOR	Compare Match Timer Constant Register	0x004	16	R/W	0x0000	0xFFFF

Table 1.2 Register description (9 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
CMT2	—	—	—	CMCR	Compare Match Timer Control Register	0x000	16	R/W	0x0000	0xFF7F
CMT2	—	—	—	CMCNT	Compare Match Timer Counter Register	0x002	16	R/W	0x0000	0xFFFF
CMT2	—	—	—	CMCOR	Compare Match Timer Constant Register	0x004	16	R/W	0x0000	0xFFFF
CMT3	—	—	—	CMCR	Compare Match Timer Control Register	0x000	16	R/W	0x0000	0xFF7F
CMT3	—	—	—	CMCNT	Compare Match Timer Counter Register	0x002	16	R/W	0x0000	0xFFFF
CMT3	—	—	—	CMCOR	Compare Match Timer Constant Register	0x004	16	R/W	0x0000	0xFFFF
CMT4	—	—	—	CMCR	Compare Match Timer Control Register	0x000	16	R/W	0x0000	0xFF7F
CMT4	—	—	—	CMCNT	Compare Match Timer Counter Register	0x002	16	R/W	0x0000	0xFFFF
CMT4	—	—	—	CMCOR	Compare Match Timer Constant Register	0x004	16	R/W	0x0000	0xFFFF
CMT5	—	—	—	CMCR	Compare Match Timer Control Register	0x000	16	R/W	0x0000	0xFF7F
CMT5	—	—	—	CMCNT	Compare Match Timer Counter Register	0x002	16	R/W	0x0000	0xFFFF
CMT5	—	—	—	CMCOR	Compare Match Timer Constant Register	0x004	16	R/W	0x0000	0xFFFF
CMTW0-1	—	—	—	CMWSTR	Timer Start Register	0x000	16	R/W	0x0000	0xFFFF
CMTW0-1	—	—	—	CMWCR	Timer Control Register	0x004	16	R/W	0x0000	0xFFFF
CMTW0-1	—	—	—	CMWIOR	Timer I/O Control Register	0x008	16	R/W	0x0000	0xFFFF
CMTW0-1	—	—	—	CMWCNT	Timer Counter	0x010	32	R/W	0x00000000	0xFFFFFFFF
CMTW0-1	—	—	—	CMWCOR	Compare Match Constant Register	0x014	32	R/W	0xFFFFFFFF	0xFFFFFFFF
CMTW0-1	2	0x004	0-1	CMWICR%s	Input Capture Registers	0x018	32	R	0x00000000	0xFFFFFFFF
CMTW0-1	2	0x004	0-1	CMWOCR%s	Output Compare Registers	0x020	32	R/W	0xFFFFFFFF	0xFFFFFFFF
WDT0	—	—	—	WDTRR	WDT Refresh Register	0x00	8	R/W	0xFF	0xFF
WDT0	—	—	—	WDTCR	WDT Control Register	0x02	16	R/W	0x33F3	0xFFFF
WDT0	—	—	—	WDTSR	WDT Status Register	0x04	16	R/W	0x0000	0xFFFF
WDT0	—	—	—	WDTRCR	WDT Reset Control Register	0x06	8	R/W	0x80	0xFF
IIC0-1	—	—	—	ICCR1	I2C Bus Control Register 1	0x00	8	R/W	0x1F	0xFF
IIC0-1	—	—	—	ICCR2	I2C Bus Control Register 2	0x01	8	R/W	0x00	0xFF
IIC0-1	—	—	—	ICMR1	I2C Bus Mode Register 1	0x02	8	R/W	0x08	0xFF
IIC0-1	—	—	—	ICMR2	I2C Bus Mode Register 2	0x03	8	R/W	0x06	0xFF
IIC0-1	—	—	—	ICMR3	I2C Bus Mode Register 3	0x04	8	R/W	0x00	0xFF
IIC0-1	—	—	—	ICFER	I2C Bus Function Enable Register	0x05	8	R/W	0x72	0xFF
IIC0-1	—	—	—	ICSER	I2C Bus Status Enable Register	0x06	8	R/W	0x09	0xFF
IIC0-1	—	—	—	ICIER	I2C Bus Interrupt Enable Register	0x07	8	R/W	0x00	0xFF
IIC0-1	—	—	—	ICSR1	I2C Bus Status Register 1	0x08	8	R/W	0x00	0xFF
IIC0-1	—	—	—	ICSR2	I2C Bus Status Register 2	0x09	8	R/W	0x00	0xFF
IIC0-1	3	0x02	0-2	SARL%s	Slave Address Register L %s (y = 0 to 2)	0x0A	8	R/W	0x00	0xFF
IIC0-1	3	0x02	0-2	SARU%s	Slave Address Register U %s (y = 0 to 2)	0x0B	8	R/W	0x00	0xFF
IIC0-1	—	—	—	ICBRL	I2C Bus Bit Rate Low-Level Register	0x10	8	R/W	0xFF	0xFF
IIC0-1	—	—	—	ICBRH	I2C Bus Bit Rate High-Level Register	0x11	8	R/W	0xFF	0xFF
IIC0-1	—	—	—	ICDRT	I2C Bus Transmit Data Register	0x12	8	R/W	0xFF	0xFF
IIC0-1	—	—	—	ICDRR	I2C Bus Receive Data Register	0x13	8	R	0x00	0xFF
DOC	—	—	—	DOCR	DOC Control Register	0x00	8	R/W	0x00	0xFF
DOC	—	—	—	DODIR	DOC Data Input Register	0x02	16	R/W	0x0000	0xFFFF
DOC	—	—	—	DODSR	DOC Data Setting Register	0x04	16	R/W	0x0000	0xFFFF
TSU	—	—	—	TSUSM	Sensor Mode Register	0x00	32	R/W	0x00000000	0xFFFFFFFF
TSU	—	—	—	TSUST	Sensor Trigger Register	0x04	32	R/W	0x00000000	0xFFFFFFFF
TSU	—	—	—	TSUSCS	Sensor Configuration Setting Register	0x08	32	R/W	0x00000001	0xFFFFFFFF
TSU	—	—	—	TSUSAD	Sensor ADC Data Register	0x0C	32	R	0x00000000	0xFFFFFFFF
TSU	—	—	—	TSUSS	Sensor Status Register	0x10	32	R	0x00000000	0xFFFFFFFF

Table 1.2 Register description (10 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
POEG1	—	—	—	POEG1GA	POEG1 Group A Setting Register	0x000	32	R/W	0x00000000	0xFFFFFFFF
POEG1	—	—	—	POEG1GB	POEG1 Group B Setting Register	0x400	32	R/W	0x00000000	0xFFFFFFFF
POEG1	—	—	—	POEG1GC	POEG1 Group C Setting Register	0x800	32	R/W	0x00000000	0xFFFFFFFF
POEG1	—	—	—	POEG1GD	POEG1 Group D Setting Register	0xC00	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	0-7	N0SA_%s	Next0 Source Address Register %s	0x000	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	0-7	N0DA_%s	Next0 Destination Address Register %s	0x004	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	0-7	N0TB_%s	Next0 Transaction Byte Register %s	0x008	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	0-7	N1SA_%s	Next1 Source Address Register %s	0x00C	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	0-7	N1DA_%s	Next1 Destination Address Register %s	0x010	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	0-7	N1TB_%s	Next1 Transaction Byte Register %s	0x014	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	0-7	CRSA_%s	Current Source Address Register %s	0x018	32	R	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	0-7	CRDA_%s	Current Destination Address Register %s	0x01C	32	R	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	0-7	CRTB_%s	Current Transaction Byte Register %s	0x020	32	R	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	0-7	CHSTAT_%s	Channel Status Register %s	0x024	32	R	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	0-7	CHCTRL_%s	Channel Control Register %s	0x028	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	0-7	CHCFG_%s	Channel Configuration Register %s	0x02C	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	0-7	CHITVL_%s	Channel Interval Register %s	0x030	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	0-7	CHEXT_%s	Channel Extension Register %s	0x034	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	0-7	NXLA_%s	Next Link Address Register %s	0x038	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	0-7	CRLA_%s	Current Link Address Register %s	0x03C	32	R	0x00000000	0xFFFFFFFF
DMAC0-1	—	—	—	DCTRL_A	DMA Control Register A	0x300	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	—	—	—	DSTAT_EN_A	DMA Status EN Register A	0x310	32	R	0x00000000	0xFFFFFFFF
DMAC0-1	—	—	—	DSTAT_ER_A	DMA Status ER Register A	0x314	32	R	0x00000000	0xFFFFFFFF
DMAC0-1	—	—	—	DSTAT_END_A	DMA Status END Register A	0x318	32	R	0x00000000	0xFFFFFFFF
DMAC0-1	—	—	—	DSTAT_SUS_A	DMA Status SUS Register A	0x320	32	R	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	8-15	N0SA_%s	Next0 Source Address Register %s	0x400	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	8-15	N0DA_%s	Next0 Destination Address Register %s	0x404	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	8-15	N0TB_%s	Next0 Transaction Byte Register %s	0x408	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	8-15	N1SA_%s	Next1 Source Address Register %s	0x40C	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	8-15	N1DA_%s	Next1 Destination Address Register %s	0x410	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	8-15	N1TB_%s	Next1 Transaction Byte Register %s	0x414	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	8-15	CRSA_%s	Current Source Address Register %s	0x418	32	R	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	8-15	CRDA_%s	Current Destination Address Register %s	0x41C	32	R	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	8-15	CRTB_%s	Current Transaction Byte Register %s	0x420	32	R	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	8-15	CHSTAT_%s	Channel Status Register %s	0x424	32	R	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	8-15	CHCTRL_%s	Channel Control Register %s	0x428	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	8-15	CHCFG_%s	Channel Configuration Register %s	0x42C	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	8-15	CHITVL_%s	Channel Interval Register %s	0x430	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	8-15	CHEXT_%s	Channel Extension Register %s	0x434	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	8-15	NXLA_%s	Next Link Address Register %s	0x438	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	8	0x40	8-15	CRLA_%s	Current Link Address Register %s	0x43C	32	R	0x00000000	0xFFFFFFFF
DMAC0-1	—	—	—	DCTRL_B	DMA Control Register B	0x700	32	R/W	0x00000000	0xFFFFFFFF
DMAC0-1	—	—	—	DSTAT_EN_B	DMA Status EN Register B	0x710	32	R	0x00000000	0xFFFFFFFF
DMAC0-1	—	—	—	DSTAT_ER_B	DMA Status ER Register B	0x714	32	R	0x00000000	0xFFFFFFFF
DMAC0-1	—	—	—	DSTAT_END_B	DMA Status END Register B	0x718	32	R	0x00000000	0xFFFFFFFF
DMAC0-1	—	—	—	DSTAT_SUS_B	DMA Status SUS Register B	0x720	32	R	0x00000000	0xFFFFFFFF
ICU_NS	—	—	—	NS_SWINT	Software Interrupt Register	0x00	32	W	0x00000000	0xFFFFFFFF

Table 1.2 Register description (11 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
ICU_NS	—	—	—	NS_PORTNF_FLTSEL	Interrupt Noise Filter Enable Register	0x04	32	R/W	0x00000000	0xFFFFFFFF
ICU_NS	—	—	—	NS_PORTNF_CLKSEL	Interrupt Noise Filter Setting Register	0x08	32	R/W	0x00000000	0xFFFFFFFF
ICU_NS	—	—	—	NS_PORTNF_MD	Interrupt Edge Detection Setting Register	0x0C	32	R/W	0x15555555	0xFFFFFFFF
ELC	19	0x04	0-18	ELC_SSEL%s	ELC Event Source Select Register %s	0x00	32	R/W	0x3FFFFFFF	0xFFFFFFFF
DMA	5	0x04	0-4	DMAC0_RSSEL%s	DMAC Unit 0 Resource Select Register %s	0x0C	32	R/W	0x3FFFFFFF	0xFFFFFFFF
DMA	—	—	—	DMAC0_RSSEL5	DMAC Unit 0 Resource Select Register 5	0x20	32	R/W	0x000003FF	0xFFFFFFFF
DMA	5	0x04	0-4	DMAC1_RSSEL%s	DMAC Unit 1 Resource Select Register %s	0x24	32	R/W	0x3FFFFFFF	0xFFFFFFFF
DMA	—	—	—	DMAC1_RSSEL5	DMAC Unit 1 Resource Select Register 5	0x38	32	R/W	0x000003FF	0xFFFFFFFF
PORT_NSR	10	0x001	0-9	P0%s	Port %s Register	0x000	8	R/W	0x00	0xFF
PORT_NSR	15	0x001	10-24	P%s	Port %s Register	0x00A	8	R/W	0x00	0xFF
PORT_NSR	10	0x002	0-9	PM0%s	Port %s Mode Register	0x200	16	R/W	0x0000	0xFFFF
PORT_NSR	15	0x002	10-24	PM%s	Port %s Mode Register	0x214	16	R/W	0x0000	0xFFFF
PORT_NSR	10	0x001	0-9	PMC0%s	Port %s Mode Control Register	0x400	8	R/W	0x00	0xFF
PORT_NSR	15	0x001	10-24	PMC%s	Port %s Mode Control Register	0x40A	8	R/W	0x00	0xFF
PORT_NSR	10	0x004	0-9	PFC0%s	Port %s Function Control Register	0x600	32	R/W	0x00000000	0xFFFFFFFF
PORT_NSR	15	0x004	10-24	PFC%s	Port %s Function Control Register	0x628	32	R/W	0x00000000	0xFFFFFFFF
PORT_NSR	10	0x001	0-9	PIN0%s	Port %s Input Register	0x800	8	R	0x00	0xFF
PORT_NSR	15	0x001	10-24	PIN%s	Port %s Input Register	0x80A	8	R	0x00	0xFF
PORT_NSR	10	0x008	0-9	DRCTL0%s_L	I/O Buffer %s Function Switching Register	0xA00	32	R/W	0x00000000	0xFFFFFFFF
PORT_NSR	10	0x008	0-9	DRCTL0%s_H	I/O Buffer %s Function Switching Register	0xA04	32	R/W	0x00000000	0xFFFFFFFF
PORT_NSR	15	0x008	10-24	DRCTL%s_L	I/O Buffer %s Function Switching Register	0xA50	32	R/W	0x00000000	0xFFFFFFFF
PORT_NSR	15	0x008	10-24	DRCTL%s_H	I/O Buffer %s Function Switching Register	0xA54	32	R/W	0x00000000	0xFFFFFFFF
PORT_NSR	2	0x01	1-2	ELC_PGR%s	ELC Port Group Setting Register %s	0xE00	8	R/W	0x00	0xFF
PORT_NSR	2	0x01	1-2	ELC_PGC%s	ELC Port Group Control Register %s	0xE02	8	R/W	0x88	0xFF
PORT_NSR	2	0x04	1-2	ELC_PDBF%s	ELC Port Buffer Register %s	0xE04	8	R/W	0x00	0xFF
PORT_NSR	4	0x01	0-3	ELC_PEL%s	ELC Port Setting Register %s	0xE0C	8	R/W	0x80	0xFF
PORT_NSR	—	—	—	ELC_DPTC	ELC Edge Detection Control Register	0xE10	8	R/W	0x00	0xFF
PORT_NSR	—	—	—	ELC_ELSR2	ELC Port Event Control Register	0xE11	8	R/W	0x00	0xFF
GMAC	—	—	—	MAC_Configuration	MAC Configuration Register	0x0000	32	R/W	0x00000000	0x0AFFFEFF
GMAC	—	—	—	MAC_Frame_Filter	MAC Frame Filter Register	0x0004	32	R/W	0x00000000	0x800107FF
GMAC	—	—	—	GMII_Address	GMII Address Register	0x0010	32	R/W	0x00000000	0x0000FFFF
GMAC	—	—	—	GMII_Data	GMII Data Register	0x0014	32	R/W	0x00000000	0x0000FE00
GMAC	—	—	—	Flow_Control	Flow Control Register	0x0018	32	R/W	0x00000000	0xFFFF00BF
GMAC	—	—	—	VLAN_Tag	VLAN Tag Register	0x001C	32	R/W	0x00000000	0x800FFFFF
GMAC	—	—	—	Version	Version Register	0x0020	32	R	0x00003137	0x0000FFFF
GMAC	—	—	—	Debug	Debug Register	0x0024	32	R	0x00000000	0x037F0377
GMAC	—	—	—	Remote_Wake_Up_Frame_Filter	Remote Wake-Up Frame Filter Register	0x0028	32	R/W	0x00000000	0xFFFFFFFF
GMAC	—	—	—	PMT_Control_Status	PMT Control and Status Register	0x002C	32	R/W	0x00000000	0x87000667
GMAC	—	—	—	LPI_Control_Status	LPI Control and Status Register	0x0030	32	R/W	0x00000000	0x000B030F
GMAC	—	—	—	LPI_Timers_Control	LPI Timers Control Register	0x0034	32	R/W	0x03E80000	0x03FFFFFF

Table 1.2 Register description (12 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
GMAC	—	—	—	Interrupt_Status	Interrupt Status Register	0x0038	32	R	0x00000000	0x000006F8
GMAC	—	—	—	Interrupt_Mask	Interrupt Mask Register	0x003C	32	R/W	0x00000000	0x00000000
GMAC	—	—	—	MAR0_H	MAC Address 0 High Register	0x0040	32	R/W	0x8000FFFF	0x8000FFFF
GMAC	—	—	—	MAR0_L	MAC Address 0 Low Register	0x0044	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GMAC	15	0x8	1-15	MAR%s_H	MAC ADDRESS High Register	0x0048	32	R/W	0x0000FFFF	0xFF00FFFF
GMAC	15	0x8	1-15	MAR%s_L	MAC ADDRESS Low Register	0x004C	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GMAC	—	—	—	WDog_Timeout	Watchdog Timeout Register	0x00DC	32	R/W	0x00000000	0x00013FFF
GMAC	—	—	—	MMC_Control	MMC Control Register	0x0100	32	R/W	0x00000000	0x0000013F
GMAC	—	—	—	MMC_Receive_Interrupt	MMC Receive Interrupt Register	0x0104	32	R	0x00000000	0x07FFFFFF
GMAC	—	—	—	MMC_Transmit_Interrupt	MMC Transmit Interrupt Register	0x0108	32	R	0x00000000	0x03FFFFFF
GMAC	—	—	—	MMC_Receive_Interrupt_Mask	MMC Receive Interrupt Mask Register	0x010C	32	R/W	0x00000000	0x03FFFFFF
GMAC	—	—	—	MMC_Transmit_Interrupt_Mask	MMC Transmit Interrupt Mask Register	0x0110	32	R/W	0x00000000	0x03FFFFFF
GMAC	—	—	—	Tx_Octet_Count_Good_Bad	Transmit Octet Count for Good and Bad Frames	0x0114	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Tx_Frame_Count_God_Bad	Transmit Frame Count for Good and Bad Frames	0x0118	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Tx_Broadcast_Frames_Good	Transmit Frame Count for Good Broadcast Frames	0x011C	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Tx_Multicast_Frames_Good	Transmit Frame Count for Good Multicast Frames	0x0120	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Tx_64Octets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 64 Byte Frames	0x0124	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Tx_65To127Octets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 65 to 127 Bytes Frames	0x0128	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Tx_128To255Octets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 128 to 255 Bytes Frames	0x012C	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Tx_256To511Octets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 256 to 511 Bytes Frames	0x0130	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Tx_512To1023Octets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 512 to 1023 Bytes Frames	0x0134	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Tx_1024ToMaxOctets_Frames_Good_Bad	Transmit Octet Count for Good and Bad 1024 to Maxsize Bytes Frames	0x0138	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Tx_Unicast_Frames_Good_Bad	Transmit Frame Count for Good and Bad Unicast Frames	0x013C	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Tx_Multicast_Frames_Good_Bad	Transmit Frame Count for Good and Bad Multicast Frames	0x0140	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Tx_Broadcast_Frames_Good_Bad	Transmit Frame Count for Good and Bad Broadcast Frames	0x0144	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Tx_Underflow_Error_Frames	Transmit Frame Count for Underflow Error Frames	0x0148	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Tx_Single_Collision_Good_Frames	Transmit Frame Count for Frames Transmitted after Single Collision	0x014C	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Tx_Multiple_Collision_Good_Frames	Transmit Frame Count for Frames Transmitted after Multiple Collision	0x0150	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Tx_Deferred_Frames	Transmit Frame Count for Deferred Frames	0x0154	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Tx_Late_Collision_Frames	Transmit Frame Count for Late Collision Error Frames	0x0158	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Tx_Excessive_Collision_Frames	Transmit Frame Count for Excessive Collision Error Frames	0x015C	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Tx_Carrier_Error_Frames	Transmit Frame Count for Carrier Sense Error Frames	0x0160	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Tx_Octet_Count_Good	Transmit Octet Count for Good Frames	0x0164	32	R	0x00000000	0xFFFFFFFF

Table 1.2 Register description (13 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
GMAC	—	—	—	Tx_Frame_Count_Good	Transmit Frame Count for Good Frames	0x0168	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Tx_Excessive_Deferral_Error	Transmit Frame Count for Excessive Deferral Error Frames	0x016C	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Tx_Pause_Frames	Transmit Frame Count for Good PAUSE Frames	0x0170	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Tx_VLAN_Frames_Good	Transmit Frame Count for Good VLAN Frames	0x0174	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Tx_OSize_Frames_Good	Transmit Frame Count for Good Oversize Frames	0x0178	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Rx_Frames_Count_Good_Bad	Receive Frame Count for Good and Bad Frames	0x0180	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Rx_Octet_Count_Good_Bad	Receive Octet Count for Good and Bad Frames	0x0184	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Rx_Octet_Count_Good	Receive Octet Count for Good Frames	0x0188	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Rx_Broadcast_Frames_Good	Receive Frame Count for Good Broadcast Frames	0x018C	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Rx_Multicast_Frames_Good	Receive Frame Count for Good Multicast Frames	0x0190	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Rx_CRC_Error_Frames	Receive Frame Count for CRC Error Frames	0x0194	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Rx_Alignment_Error_Frames	Receive Frame Count for Alignment Error Frames	0x0198	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Rx_Runt_Error_Frames	Receive Frame Count for Runt Error Frames	0x019C	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Rx_Jabber_Error_Frames	Receive Frame Count for Jabber Error Frames	0x01A0	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Rx_Undersize_Frames_Good	Receive Frame Count for Undersize Frames	0x01A4	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Rx_Oversize_Frames_Good	Receive Frame Count for Oversize Frames	0x01A8	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Rx_64Octets_Frames_Good_Bad	Receive Frame Count for Good and Bad 64 Byte Frames	0x01AC	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Rx_65To127Octets_Frames_Good_Bad	Receive Frame Count for Good and Bad 65 to 127 Bytes Frames	0x01B0	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Rx_128To255Octets_Frames_Good_Bad	Receive Frame Count for Good and Bad 128 to 255 Bytes Frames	0x01B4	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Rx_256To511Octets_Frames_Good_Bad	Receive Frame Count for Good and Bad 256 to 511 Bytes Frames	0x01B8	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Rx_512To1023Octets_Frames_Good_Bad	Receive Frame Count for Good and Bad 512 to 1,023 Bytes Frames	0x01BC	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Rx_1024ToMaxOctets_Frames_Good_Bad	Receive Frame Count for Good and Bad 1,024 to Maxsize Bytes Frames	0x01C0	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Rx_Unicast_Frames_Good	Receive Frame Count for Good Unicast Frames	0x01C4	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Rx_Length_Error_Frames	Receive Frame Count for Length Error Frames	0x01C8	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Rx_Out_Of_Range_Type_Frames	Receive Frame Count for Out of Range Frames	0x01CC	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Rx_Pause_Frames	Receive Frame Count for PAUSE Frames	0x01D0	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Rx_FIFO_Overflow_Frames	Receive Frame Count for FIFO Overflow Frames	0x01D4	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Rx_VLAN_Frames_Good_Bad	Receive Frame Count for Good and Bad VLAN Frames	0x01D8	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Rx_Watchdog_Error_Frames	Receive Frame Count for Watchdog Error Frames	0x01DC	32	R	0x00000000	0x0000FFFF
GMAC	—	—	—	Rx_Receive_Error_Frames	Receive Frame Count for Receive Error Frames	0x01E0	32	R	0x00000000	0x0000FFFF

Table 1.2 Register description (14 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
GMAC	—	—	—	Rx_Control_Frames_Good	Receive Frame Count for Good Control Frames	0x01E4	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	GMACTRGSEL	GMAC PTP Trigger Select Register	0x0400	32	R/W	0x00000000	0xFFFFFFFF
GMAC	8	0x4	0-7	HASH_TABLE_REG %s	Hash Table Register %s (n = 0 to 7)	0x0500	32	R/W	0x00000000	0xFFFFFFFF
GMAC	—	—	—	VLAN_Hash_Table_Reg	VLAN Hash Table Register	0x0588	32	R/W	0x00000000	0x0000FFFF
GMAC	—	—	—	Timestamp_Control	Timestamp Control Register	0x0700	32	R/W	0x00002000	0x0707FF3F
GMAC	—	—	—	Sub_Second_Increment	Sub-Second Increment Register	0x0704	32	R/W	0x00000000	0x000000FF
GMAC	—	—	—	System_Time_Seconds	System Time - Seconds Register	0x0708	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	System_Time_Nanoseconds	System Time - Nanoseconds Register	0x070C	32	R	0x00000000	0x7FFFFFFF
GMAC	—	—	—	System_Time_Seconds_Update	System Time - Seconds Update Register	0x0710	32	R/W	0x00000000	0xFFFFFFFF
GMAC	—	—	—	System_Time_Nanoseconds_Update	System Time - Nanoseconds Update Register	0x0714	32	R/W	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Timestamp_Addend	Timestamp Addend Register	0x0718	32	R/W	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Target_Time_Seconds	Target Time Seconds Register	0x071C	32	R/W	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Target_Time_Nanoseconds	Target Time Nanoseconds Register	0x0720	32	R/W	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Timestamp_Status	Timestamp Status Register	0x0728	32	R	0x00000000	0x3F0F0004
GMAC	—	—	—	Auxiliary_Timestamp_Nanoseconds	Auxiliary Timestamp – Nanoseconds Register	0x0730	32	R	0x00000000	0x7FFFFFFF
GMAC	—	—	—	Auxiliary_Timestamp_Seconds	Auxiliary Timestamp - Seconds Register	0x0734	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	MAR16_H	MAC ADDRESS High Register	0x0800	32	R/W	0x0000FFFF	0xFF00FFFF
GMAC	—	—	—	MAR16_L	MAC ADDRESS Low Register	0x0804	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GMAC	—	—	—	MAR17_H	MAC ADDRESS High Register	0x0808	32	R/W	0x0000FFFF	0xFF00FFFF
GMAC	—	—	—	MAR17_L	MAC ADDRESS Low Register	0x080C	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GMAC	—	—	—	Bus_Mode	Bus Mode Register	0x1000	32	R/W	0x00020100	0xBFFFFFFF
GMAC	—	—	—	Transmit_Poll_Demand	Transmit Poll Demand Register	0x1004	32	R/W	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Receive_Poll_Demand	Receive Poll Demand Register	0x1008	32	R/W	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Receive_Descriptor_List_Address	Receive Descriptor List Address Register	0x100C	32	R/W	0x00000000	0xFFFFFFFC
GMAC	—	—	—	Transmit_Descriptor_List_Address	Transmit Descriptor List Address Register	0x1010	32	R/W	0x00000000	0xFFFFFFFC
GMAC	—	—	—	Status	Status Register	0x1014	32	R/W	0x00000000	0x7BFFE7FF
GMAC	—	—	—	Operation_Mode	Operation Mode Register	0x1018	32	R/W	0x00000000	0x0671FFFE
GMAC	—	—	—	Interrupt_Enable	Interrupt Enable Register	0x101C	32	R/W	0x00000000	0x0001E7FF
GMAC	—	—	—	Missed_Frame_And_Buffer_Overflow_Counter	Missed Frame and Buffer Overflow Counter Register	0x1020	32	R	0x00000000	0x1FFFFFFF
GMAC	—	—	—	Receive_Interrupt_Watchdog_Timer	Receive Interrupt Watchdog Timer Register	0x1024	32	R/W	0x00000000	0x000000FF
GMAC	—	—	—	AXI_Bus_Mode	AXI Bus Mode Register	0x1028	32	R/W	0x00110001	0xC033300F
GMAC	—	—	—	AXI_Status	AXI Status Register	0x102C	32	R	0x00000000	0x00000003
GMAC	—	—	—	Current_Host_Transmit_Descriptor	Current Host Transmit Descriptor Register	0x1048	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Current_Host_Receive_Descriptor	Current Host Receive Descriptor Register	0x104C	32	R	0x00000000	0xFFFFFFFF

Table 1.2 Register description (15 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
GMAC	—	—	—	Current_Host_Transmit_Buffer_Address	Current Host Transmit Buffer Address Register	0x1050	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	Current_Host_Receive_Buffer_Address	Current Host Receive Buffer Address Register	0x1054	32	R	0x00000000	0xFFFFFFFF
GMAC	—	—	—	HW_Feature	HW Feature Register	0x1058	32	R	0x030D6F3F	0x7FFFFFFF
ETHSS	—	—	—	PRCMD	Ethernet Protect Register	0x000	32	R/W	0x00000000	0xFFFFFFFF
ETHSS	—	—	—	MODCTRL	Mode Control Register	0x008	32	R/W	0x00000000	0xFFFFFFFF
ETHSS	—	—	—	PTPMCTRL	PTP Mode Control Register	0x00C	32	R/W	0x00000000	0xFFFFFFFF
ETHSS	—	—	—	PHYLNK	Ethernet PHY Link Mode Register	0x014	32	R/W	0x00000070	0xFFFFFFFF
ETHSS	3	0x004	0-2	CONVCTRL%s	RGMII/RMII Converter %s Control Register	0x100	32	R/W	0x01000500	0xFFFFFFFF
ETHSS	—	—	—	CONVRST	RGMII/RMII Converter Reset Control Register	0x114	32	R/W	0x00000000	0xFFFFFFFF
ESC_INI	—	—	—	ECATOFFADR	EtherCAT PHY Offset Address Setting Register	0x00000	32	R/W	0x00000000	0xFFFFFFFF
ESC_INI	—	—	—	ECATOPMOD	EtherCAT Operation Mode Register	0x00004	32	R/W	0x00000000	0xFFFFFFFF
ESC_INI	—	—	—	ECATDBGC	EtherCAT Debug Control Register	0x00008	32	R/W	0x00000000	0xFFFFFFFF
ESC_INI	—	—	—	ECATTRGSEL	EtherCAT DC Latch Trigger Select Register	0x0000C	32	R/W	0x00000000	0xFFFFFFFF
GMAC_PTP	4	0x100	0-3	SWTMEN%s	PTP Timer Pulse Output Enable %s Register	0x400	32	R/W	0x00000000	0xFFFFFFFF
GMAC_PTP	4	0x100	0-3	SWTMSTSEC%s	PTP Timer Pulse Start Second %s Register	0x404	32	R/W	0x00000000	0xFFFFFFFF
GMAC_PTP	4	0x100	0-3	SWTMSTNS%s	PTP Timer Pulse Start Nanosecond %s Register	0x408	32	R/W	0x00000000	0xFFFFFFFF
GMAC_PTP	4	0x100	0-3	SWTMPSEC%s	PTP Timer Pulse Period Second %s Register	0x40C	32	R/W	0x00000000	0xFFFFFFFF
GMAC_PTP	4	0x100	0-3	SWTMPNS%s	PTP Timer Pulse Period Nanosecond %s Register	0x410	32	R/W	0x3B9ACA00	0xFFFFFFFF
GMAC_PTP	4	0x100	0-3	SWTMWTH%s	PTP Timer Pulse Width %s Register	0x414	32	R/W	0x00000003	0xFFFFFFFF
GMAC_PTP	4	0x100	0-3	SWTMMAXP%s	PTP Timer Pulse Max Nanosecond %s Register	0x418	32	R/W	0x3B9ACA00	0xFFFFFFFF
GMAC_PTP	4	0x100	0-3	SWTMLATSEC%s	PTP Timer Pulse Latch Second %s Register	0x41C	32	R/W	0x00000000	0xFFFFFFFF
GMAC_PTP	4	0x100	0-3	SWTMLATNS%s	PTP Timer Pulse Latch Nanosecond %s Register	0x420	32	R/W	0x00000000	0xFFFFFFFF
ESC	—	—	—	TYPE	Type Register	0x0000	8	R	0xA0	0xFF
ESC	—	—	—	REVISION	Revision Register	0x0001	8	R	0x02	0xFF
ESC	—	—	—	BUILD	Build Register	0x0002	8	R	0x00	0xFF
ESC	—	—	—	FMMU_NUM	FMMU Supported Register	0x0004	8	R	0x08	0xFF
ESC	—	—	—	SYNC_MANAGER	SyncManager Supported Register	0x0005	8	R	0x08	0xFF
ESC	—	—	—	RAM_SIZE	RAM Size Register	0x0006	8	R	0x08	0xFF
ESC	—	—	—	PORT_DESC	Port Descriptor Register	0x0007	8	R	0x3F	0xFF
ESC	—	—	—	FEATURE	ESC Features Supported Register	0x0008	16	R	0x01CC	0x0FCD
ESC	—	—	—	STATION_ADR	Configured Station Address Register	0x0010	16	R	0x0000	0xFFFF
ESC	—	—	—	STATION_ALIAS	Configured Station Alias Register	0x0012	16	R/W	0x0000	0xFFFF
ESC	—	—	—	WR_REG_ENABLE	Write Register Enable Register	0x0020	8	R	0x00	0x01
ESC	—	—	—	WR_REG_PROTECT	Write Register Protection Register	0x0021	8	R	0x00	0x01
ESC	—	—	—	ESC_WR_ENABLE	ESC Write Enable Register	0x0030	8	R	0x00	0x01
ESC	—	—	—	ESC_WR_PROTECT	ESC Write Protection Register	0x0031	8	R	0x00	0x01
ESC	—	—	—	ESC_RESET_ECAT_R	ESC Reset ECAT Register for read	0x0040	8	R	0x00	0x03
ESC	—	—	—	ESC_RESET_ECAT_W	ESC Reset ECAT Register for write	0x0040	8	R	0x00	0xFF

Table 1.2 Register description (16 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
ESC	—	—	—	ESC_RESET_PDI_R	ESC Reset PDI Register for read	0x0041	8	R/W	0x00	0x03
ESC	—	—	—	ESC_RESET_PDI_W	ESC Reset PDI Register for write	0x0041	8	R/W	0x00	0xFF
ESC	—	—	—	ESC_DL_CONTROL	ESC DL Control Register	0x0100	32	R	0x0007C001	0x0107FF0F
ESC	—	—	—	PHYSICAL_RW_OFFSET	Physical Read/Write Offset Register	0x0108	16	R	0x0000	0xFFFF
ESC	—	—	—	ESC_DL_STATUS	ESC DL Status Register	0x0110	16	R	0x0004	0xFFF7
ESC	—	—	—	AL_CONTROL	AL Control Register	0x0120	16	R	0x0001	0x003F
ESC	—	—	—	AL_STATUS	AL Status Register	0x0130	16	R/W	0x0001	0x003F
ESC	—	—	—	AL_STATUS_CODE	AL Status Code Register	0x0134	16	R/W	0x0000	0xFFFF
ESC	—	—	—	RUN_LED_OVERRIDE	RUN LED Override Register	0x0138	8	R/W	0x00	0x1F
ESC	—	—	—	ERR_LED_OVERRIDE	ERR LED Override Register	0x0139	8	R/W	0x00	0x1F
ESC	—	—	—	PDI_CONTROL	PDI Control Register	0x0140	8	R	0x80	0xFF
ESC	—	—	—	ESC_CONFIG	ESC Configuration Register	0x0141	8	R	0x0C	0xFF
ESC	—	—	—	PDI_CONFIG	PDI Configuration Register	0x0150	8	R	0x44	0xFF
ESC	—	—	—	SYNC_LATCH_CONFIG	SYNC/LATCH PDI Configuration Register	0x0151	8	R	0xEE	0xFF
ESC	—	—	—	EXT_PDI_CONFIG	Extended PDI Configuration Register	0x0152	16	R	0x0000	0x0003
ESC	—	—	—	ECAT_EVENT_MASK	ECAT Event Mask Register	0x0200	16	R	0x0000	0xFFFF
ESC	—	—	—	AL_EVENT_MASK	AL Event Mask Register	0x0204	32	R/W	0x00FFFF0F	0xFFFFFFFF
ESC	—	—	—	ECAT_EVENT_REQ	ECAT Event Request Register	0x0210	16	R	0x0000	0x0FFD
ESC	—	—	—	AL_EVENT_REQ	AL Event Request Register	0x0220	32	R	0x00000000	0x0000FF5F
ESC	3	0x2	0-2	RX_ERR_COUNT%s	RX Error Counter %s Register (n = 0 to 2)	0x0300	16	R	0x0000	0xFFFF
ESC	3	0x1	0-2	FWD_RX_ERR_COUNT%s	Forwarded RX Error Counter %s Register (n = 0 to 2)	0x0308	8	R	0x00	0xFF
ESC	—	—	—	ECAT_PROC_ERR_COUNT	ECAT Processing Unit Error Counter Register	0x030C	8	R	0x00	0xFF
ESC	—	—	—	PDI_ERR_COUNT	PDI Error Counter Register	0x030D	8	R	0x00	0xFF
ESC	3	0x1	0-2	LOST_LINK_COUNT%s	Lost Link Counter %s Register (n = 0 to 2)	0x0310	8	R	0x00	0xFF
ESC	—	—	—	WD_DIVIDE	Watchdog Divider Register	0x0400	16	R	0x09C2	0xFFFF
ESC	—	—	—	WDT_PDI	Watchdog Time PDI Register	0x0410	16	R	0x03E8	0xFFFF
ESC	—	—	—	WDT_DATA	Watchdog Time Process Data Register	0x0420	16	R	0x03E8	0xFFFF
ESC	—	—	—	WDS_DATA	Watchdog Status Process Data Register	0x0440	16	R	0x0000	0x0001
ESC	—	—	—	WDC_DATA	Watchdog Counter Process Data Register	0x0442	8	R	0x00	0xFF
ESC	—	—	—	WDC_PDI	Watchdog Counter PDI Register	0x0443	8	R	0x00	0xFF
ESC	—	—	—	EEP_CONF	EEPROM Configuration Register	0x0500	8	R	0x00	0x03
ESC	—	—	—	EEP_STATE	EEPROM PDI Access State Register	0x0501	8	R/W	0x00	0x01
ESC	—	—	—	EEP_CONT_STAT	EEPROM Control/Status Register	0x0502	16	R/W	0x0000	0xFF41
ESC	—	—	—	EEP_ADR	EEPROM Address Register	0x0504	32	R/W	0x00000000	0xFFFFFFFF
ESC	—	—	—	EEP_DATA	EEPROM Data Register	0x0508	32	R/W	0x00000000	0xFFFFFFFF
ESC	—	—	—	MII_CONT_STAT	MII Management Control/Status Register	0x0510	16	R/W	0x0002	0xE3FF
ESC	—	—	—	PHY_ADR	PHY Address Register	0x0512	8	R/W	0x00	0x1F
ESC	—	—	—	PHY_REG_ADR	PHY Register Address Register	0x0513	8	R/W	0x00	0x1F
ESC	—	—	—	PHY_DATA	PHY Data Register	0x0514	16	R/W	0x0000	0xFFFF
ESC	—	—	—	MII_ECAT_ACS_STAT	MII Management ECAT Access State Register	0x0516	8	R	0x00	0x01

Table 1.2 Register description (17 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
ESC	—	—	—	MII_PDI_ACS_STAT	MII Management PDI Access State Register	0x0517	8	R/W	0x00	0x03
ESC	8	0x10	0-7	FMMU %s_L_START_ADR	FMMU Logical Start Address %s Register (n = 0 to 7)	0x0600	32	R	0x00000000	0xFFFFFFFF
ESC	8	0x10	0-7	FMMU%s_LEN	FMMU Length %s Register (n = 0 to 7)	0x0604	16	R	0x0000	0xFFFF
ESC	8	0x10	0-7	FMMU %s_L_START_BIT	FMMU Logical Start Bit %s Register (n = 0 to 7)	0x0606	8	R	0x00	0x07
ESC	8	0x10	0-7	FMMU %s_L_STOP_BIT	FMMU Logical Stop Bit %s Register (n = 0 to 7)	0x0607	8	R	0x00	0x07
ESC	8	0x10	0-7	FMMU %s_P_START_ADR	FMMU Physical Start Address %s Register (n = 0 to 7)	0x0608	16	R	0x0000	0xFFFF
ESC	8	0x10	0-7	FMMU %s_P_START_BIT	FMMU Physical Start Bit %s Register (n = 0 to 7)	0x060A	8	R	0x00	0x07
ESC	8	0x10	0-7	FMMU%s_TYPE	FMMU Type %s Register (n = 0 to 7)	0x060B	8	R	0x00	0x03
ESC	8	0x10	0-7	FMMU%s_ACT	FMMU Activate %s Register (n = 0 to 7)	0x060C	8	R	0x00	0x01
ESC	8	0x8	0-7	SM %s_P_START_ADR	SyncManager Physical Start Address %s Register (n = 0 to 7)	0x0800	16	R	0x0000	0xFFFF
ESC	8	0x8	0-7	SM%s_LEN	SyncManager Length %s Register (n = 0 to 7)	0x0802	16	R	0x0000	0xFFFF
ESC	8	0x8	0-7	SM%s_CONTROL	SyncManager Control %s Register (n = 0 to 7)	0x0804	8	R	0x00	0x7F
ESC	8	0x8	0-7	SM%s_STATUS	SyncManager Status %s Register (n = 0 to 7)	0x0805	8	R	0x30	0xFB
ESC	8	0x8	0-7	SM%s_ACT	SyncManager Activate %s Register (n = 0 to 7)	0x0806	8	R	0x00	0xC3
ESC	8	0x8	0-7	SM%s_PDI_CONT	SyncManager PDI Control %s Register (n = 0 to 7)	0x0807	8	R/W	0x00	0x03
ESC	—	—	—	DC_RCV_TIME_PORT0	Receive Time Port 0 Register	0x0900	32	R	0x00000000	0x00000000
ESC	—	—	—	DC_RCV_TIME_PORT1	Receive Time Port 1 Register	0x0904	32	R	0x00000000	0x00000000
ESC	—	—	—	DC_RCV_TIME_PORT2	Receive Time Port 2 Register	0x0908	32	R	0x00000000	0x00000000
ESC	—	—	—	DC_SYS_TIME_L	System Time Register L	0x0910	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_SYS_TIME_H	System Time Register H	0x0914	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_RCV_TIME_UNIT_L	Receive Time ECAT Processing Unit Register L	0x0918	32	R	0x00000000	0x00000000
ESC	—	—	—	DC_RCV_TIME_UNIT_H	Receive Time ECAT Processing Unit Register H	0x091C	32	R	0x00000000	0x00000000
ESC	—	—	—	DC_SYS_TIME_OFFSET_L	System Time Offset Register L	0x0920	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_SYS_TIME_OFFSET_H	System Time Offset Register H	0x0924	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_SYS_TIME_DELAY	System Time Delay Register	0x0928	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_SYS_TIME_DIFF	System Time Difference Register	0x092C	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_SPEED_COUNT_START	Speed Counter Start Register	0x0930	16	R	0x1000	0x7FFF
ESC	—	—	—	DC_SPEED_COUNT_DIFF	Speed Counter Difference Register	0x0932	16	R	0x0000	0xFFFF
ESC	—	—	—	DC_SYS_TIME_DIFF_FIL_DEPTH	System Time Difference Filter Depth Register	0x0934	8	R	0x04	0x0F
ESC	—	—	—	DC_SPEED_COUNT_FIL_DEPTH	Speed Counter Filter Depth Register	0x0935	8	R	0x0C	0x0F
ESC	—	—	—	DC_CYC_CONT	Cyclic Unit Control Register	0x0980	8	R	0x00	0x31
ESC	—	—	—	DC_ACT	Activation Register	0x0981	8	R/W	0x00	0xFF
ESC	—	—	—	DC_PULSE_LEN	SYNC Signal Pulse Length Register	0x0982	16	R	0x0000	0xFFFF

Table 1.2 Register description (18 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
ESC	—	—	—	DC_ACT_STAT	Activation Status Register	0x0984	8	R	0x00	0x07
ESC	—	—	—	DC_SYNC0_STAT	SYNC0 Status Register	0x098E	8	R	0x00	0x01
ESC	—	—	—	DC_SYNC1_STAT	SYNC1 Status Register	0x098F	8	R	0x00	0x01
ESC	—	—	—	DC_CYC_START_TIME_L	Start Time Cyclic Operation/Next SYNC0 Pulse Register L	0x0990	32	R/W	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_CYC_START_TIME_H	Start Time Cyclic Operation/Next SYNC0 Pulse Register H	0x0994	32	R/W	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_NEXT_SYNC1_PULSE_L	Next SYNC1 Pulse Register L	0x0998	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_NEXT_SYNC1_PULSE_H	Next SYNC1 Pulse Register H	0x099C	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_SYNC0_CYC_TIME	SYNC0 Cycle Time Register	0x09A0	32	R/W	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_SYNC1_CYC_TIME	SYNC1 Cycle Time Register	0x09A4	32	R/W	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_LATCH0_CONT	Latch 0 Control Register	0x09A8	8	R/W	0x00	0x03
ESC	—	—	—	DC_LATCH1_CONT	Latch 1 Control Register	0x09A9	8	R/W	0x00	0x03
ESC	—	—	—	DC_LATCH0_STAT	Latch 0 Status Register	0x09AE	8	R	0x00	0x07
ESC	—	—	—	DC_LATCH1_STAT	Latch 1 Status Register	0x09AF	8	R	0x00	0x07
ESC	—	—	—	DC_LATCH0_TIME_POS_L	Latch 0 Time Positive Edge Register L	0x09B0	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_LATCH0_TIME_POS_H	Latch 0 Time Positive Edge Register H	0x09B4	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_LATCH0_TIME_NEG_L	Latch 0 Time Negative Edge Register L	0x09B8	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_LATCH0_TIME_NEG_H	Latch 0 Time Negative Edge Register H	0x09BC	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_LATCH1_TIME_POS_L	Latch 1 Time Positive Edge Register L	0x09C0	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_LATCH1_TIME_POS_H	Latch 1 Time Positive Edge Register H	0x09C4	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_LATCH1_TIME_NEG_L	Latch 1 Time Negative Edge Register L	0x09C8	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_LATCH1_TIME_NEG_H	Latch 1 Time Negative Edge Register H	0x09CC	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_ECAT_CNG_EV_TIME	Buffer Change Event Time Register	0x09F0	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_PDI_START_EV_TIME	PDI Buffer Start Event Time Register	0x09F8	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	DC_PDI_CNG_EV_TIME	PDI Buffer Change Event Time Register	0x09FC	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	PRODUCT_ID_L	Product ID Register L	0x0E00	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	PRODUCT_ID_H	Product ID Register H	0x0E04	32	R	0x00000000	0xFFFFFFFF
ESC	—	—	—	VENDOR_ID_L	Vendor ID Register L	0x0E08	32	R	0x00000000	0xFFFFFFFF
USBHC	—	—	—	HCREVISION	HcRevision Register	0x000	32	R	0x00000010	0xFFFFFFFF
USBHC	—	—	—	HCCONTROL	HcControl Register	0x004	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	HCCOMMANDSTATUS	HcCommandStatus Register	0x008	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	HCINTERRUPTSTATUS	HcInterruptStatus Register	0x00C	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	HCINTERRUPTENABLE	HcInterruptEnable Register	0x010	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	HCINTERRUPTDISABLE	HcInterruptDisable Register	0x014	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	HCHCCA	HcHCCA Register	0x018	32	R/W	0x00000000	0xFFFFFFFF

Table 1.2 Register description (19 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
USBHC	—	—	—	HCPERIODCCURRENTIED	HcPeriodicCurrentED Register	0x01C	32	R	0x00000000	0xFFFFFFFF
USBHC	—	—	—	HCCONTROLHEADED	HcControlHeadED Register	0x020	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	HCCONTROLCURRENTED	HcControlCurrentED Register	0x024	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	HCBULKHEADED	HcBulkHeadED Register	0x028	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	HCBULKCURRENTED	HcBulkCurrentED Register	0x02C	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	HCDONEHEAD	HcDoneHead Register	0x030	32	R	0x00000000	0xFFFFFFFF
USBHC	—	—	—	HCFMINTERVAL	HcFmInterval Register	0x034	32	R/W	0x00002EDF	0xFFFFFFFF
USBHC	—	—	—	HCFNREMAINING	HcFmRemaining Register	0x038	32	R	0x00002EDF	0xFFFFFFFF
USBHC	—	—	—	HCFMNUMBER	HcFmNumber Register	0x03C	32	R	0x00000000	0xFFFFFFFF
USBHC	—	—	—	HCPERIODSTART	HcPeriodicStart Register	0x040	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	HCLSTHRESHOLD	HcLSThreshold Register	0x044	32	R/W	0x00000628	0xFFFFFFFF
USBHC	—	—	—	HCRHDESCRIPTORA	HcRhDescriptorA Register	0x048	32	R/W	0x0F000901	0xFFFFFFFF
USBHC	—	—	—	HCRHDESCRIPTORB	HcRhDescriptorB Register	0x04C	32	R/W	0x00020000	0xFFFFFFFF
USBHC	—	—	—	HCRHSTATUS	HcRhStatus Register	0x050	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	HCRHPORTSTATUS1	HcRhPortStatus1 Register	0x054	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	CAPL_VERSION	Capability Registers Length and EHCI Version Number Register	0x100	32	R	0x01100020	0xFFFFFFFF
USBHC	—	—	—	HCSPARAMS	Structural Parameters Register	0x104	32	R	0x00001191	0xFFFFFFFF
USBHC	—	—	—	HCCPARAMS	Capability Parameters Register	0x108	32	R	0x000E0006	0xFFFFFFFF
USBHC	—	—	—	HCSP_PORTROUTE	Companion Port Route Description Register	0x10C	32	R	0x00000000	0xFFFFFFFF
USBHC	—	—	—	USBCMD	USB Command Register	0x120	32	R/W	0x00080B00	0xFFFFFFFF
USBHC	—	—	—	USBSTS	USB Status Register	0x124	32	R/W	0x00001000	0xFFFFFFFF
USBHC	—	—	—	USBINTR	USB Interrupt Enable Register	0x128	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	FRINDEX	USB Frame Index Register	0x12C	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	CTRLDSSEGMENT	Control Data Structure Segment Register	0x130	32	R	0x00000000	0xFFFFFFFF
USBHC	—	—	—	PERIODICLISTBASE	Periodic Frame List Base Address Register	0x134	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	ASYNCLISTADDR	Next Asynchronous List Address Register	0x138	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	CONFIGFLAG	Configure Flag Register	0x160	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	PORTSC1	Port 1 Status and Control Register	0x164	32	R/W	0x00002000	0xFFFFFFFF
USBHC	—	—	—	INTENABLE	INT_ENABLE Register	0x200	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	INTSTATUS	INT_STATUS Register	0x204	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	AHBBUSCTR	AHB_BUS_CTR Register	0x208	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	USBCTR	USBCTR Register	0x20C	32	R/W	0x00000002	0xFFFFFFFF
USBHC	—	—	—	REVID	Revision and Core ID Register	0x300	32	R	0x00000301	0xFFFFFFFF
USBHC	—	—	—	OCSLPTIMSET	Overcurrent Detection/Sleep Timer Setting Register	0x310	32	R/W	0x0C830D40	0xFFFFFFFF
USBHC	—	—	—	COMMCTRL	Common Control Register	0x800	32	R/W	0x80000000	0xFFFFFFFF
USBHC	—	—	—	OBINTSTA	OTG-BC Interrupt Status Register	0x804	32	R/W	0x00000001	0xFFFFFFFF
USBHC	—	—	—	OBINTEN	OTG-BC Interrupt Enable Register	0x808	32	R/W	0x00000000	0xFFFFFFFF
USBHC	—	—	—	VBCTRL	VBUS Control Register	0x80C	32	R/W	0x00010000	0xFFFFFFFF
USBHC	—	—	—	LINECTRL1	Line Control Port 1 Register	0x810	32	R/W	0x00000000	0xFFFFFFFE
USBF	—	—	—	SYSCFG0	System Configuration Control Register 0	0x00	16	R/W	0x0020	0xFFFF

Table 1.2 Register description (20 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
USBF	—	—	—	SYSCFG1	System Configuration Control Register 1	0x02	16	R/W	0x0F0F	0xFFFF
USBF	—	—	—	SYSSTS0	System Configuration Status Register	0x04	16	R	0x0000	0x3FFF
USBF	—	—	—	DVSTCTR0	Device State Control Register 0	0x08	16	R/W	0x0000	0xFFFF
USBF	—	—	—	TESTMODE	USB Test Mode Register	0x0C	16	R/W	0x0000	0xFFFF
USBF	—	—	—	CFIFO	FIFO Port Register	0x014	32	R/W	0x00000000	0xFFFFFFFF
USBF	—	—	—	CFIFOL	FIFO Port Register	0x014	16	R/W	0x0000	0xFFFF
USBF	—	—	—	CFIFOLL	FIFO Port Register	0x014	8	R/W	0x00	0xFF
USBF	—	—	—	CFIFOH	FIFO Port Register	0x016	16	R/W	0x0000	0xFFFF
USBF	—	—	—	CFIFOHH	FIFO Port Register	0x017	8	R/W	0x00	0xFF
USBF	—	—	—	D0FIFO	FIFO Port Register	0x018	32	R/W	0x00000000	0xFFFFFFFF
USBF	—	—	—	D0FIFOL	FIFO Port Register	0x018	16	R/W	0x0000	0xFFFF
USBF	—	—	—	D0FIFOLL	FIFO Port Register	0x018	8	R/W	0x00	0xFF
USBF	—	—	—	D0FIFOH	FIFO Port Register	0x01A	16	R/W	0x0000	0xFFFF
USBF	—	—	—	D0FIFOHH	FIFO Port Register	0x01B	8	R/W	0x00	0xFF
USBF	—	—	—	D1FIFO	FIFO Port Register	0x01C	32	R/W	0x00000000	0xFFFFFFFF
USBF	—	—	—	D1FIFOL	FIFO Port Register	0x01C	16	R/W	0x0000	0xFFFF
USBF	—	—	—	D1FIFOLL	FIFO Port Register	0x01C	8	R/W	0x00	0xFF
USBF	—	—	—	D1FIFOH	FIFO Port Register	0x01E	16	R/W	0x0000	0xFFFF
USBF	—	—	—	D1FIFOHH	FIFO Port Register	0x01F	8	R/W	0x00	0xFF
USBF	—	—	—	CFIFOSEL	CFIFO Port Select Register	0x20	16	R/W	0x0000	0xFFFF
USBF	—	—	—	CFIFOCTR	FIFO Port Control Register	0x022	16	R/W	0x0000	0xFFFF
USBF	2	0x4	0-1	D%sFIFOSEL	D%sFIFO Port Select Register	0x028	16	R/W	0x0000	0xFFFF
USBF	2	0x4	0-1	D%sFIFOCTR	FIFO Port Control Register	0x02A	16	R/W	0x0000	0xFFFF
USBF	—	—	—	INTENB0	Interrupt Enable Register 0	0x30	16	R/W	0x0000	0xFFFF
USBF	—	—	—	INTENB1	Interrupt Enable Register 1	0x32	16	R/W	0x0000	0xFFFF
USBF	—	—	—	BRDYENB	BRDY Interrupt Enable Register	0x36	16	R/W	0x0000	0x03FF
USBF	—	—	—	NRDYENB	NRDY Interrupt Enable Register	0x38	16	R/W	0x0000	0x03FF
USBF	—	—	—	BEMPENB	BEMP Interrupt Enable Register	0x3A	16	R/W	0x0000	0x03FF
USBF	—	—	—	SOFCFG	SOF Pin Configuration Register	0x3C	16	R/W	0x0000	0xFFFF
USBF	—	—	—	INTSTS0	Interrupt Status Register 0	0x40	16	R/W	0x0000	0xFF7F
USBF	—	—	—	INTSTS1	Interrupt Status Register 1	0x42	16	R/W	0x0000	0xFFFF
USBF	—	—	—	BRDYSTS	BRDY Interrupt Status Register	0x46	16	R/W	0x0000	0x03FF
USBF	—	—	—	NRDYSTS	NRDY Interrupt Status Register	0x48	16	R/W	0x0000	0x03FF
USBF	—	—	—	BEMPSTS	BEMP Interrupt Status Register	0x4A	16	R/W	0x0000	0x03FF
USBF	—	—	—	FRMNUM	Frame Number Register	0x4C	16	R/W	0x0000	0xFFFF
USBF	—	—	—	UFRMNUM	Frame Number Register	0x4E	16	R	0x0000	0xFFFF
USBF	—	—	—	USBADDR	USB Address Register	0x50	16	R	0x0000	0xFFFF
USBF	—	—	—	USBREQ	USB Request Type Register	0x54	16	R	0x0000	0xFFFF
USBF	—	—	—	USBVAL	USB Request Value Register	0x56	16	R	0x0000	0xFFFF
USBF	—	—	—	USBINDX	USB Request Index Register	0x58	16	R	0x0000	0xFFFF
USBF	—	—	—	USBLENG	USB Request Length Register	0x5A	16	R	0x0000	0xFFFF
USBF	—	—	—	DCPCFG	DCP Configuration Register	0x5C	16	R/W	0x0000	0xFFFF
USBF	—	—	—	DCPMAXP	DCP Maximum Packet Size Register	0x5E	16	R/W	0x0040	0xFFFF
USBF	—	—	—	DCPCTR	DCP Control Register	0x60	16	R/W	0x0040	0xFFFF
USBF	—	—	—	PIPESEL	Pipe Window Select Register	0x64	16	R/W	0x0000	0xFFFF
USBF	—	—	—	PIPECFG	Pipe Configuration Register	0x68	16	R/W	0x0000	0xFFFF
USBF	—	—	—	PIPEBUF	Pipe Buffer Specification Register	0x6A	16	R/W	0x0000	0xFFFF

Table 1.2 Register description (21 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
USBF	—	—	—	PIPEMAXP	Pipe Maximum Packet Size Register	0x6C	16	R/W	0x0000	0xFFFF
USBF	—	—	—	PIPEPERI	Pipe Timing Control Register	0x6E	16	R/W	0x0000	0xFFFF
USBF	5	0x2	1-5	PIPE%sCTR	PIPE%s Control Register	0x070	16	R/W	0x0000	0xFFFF
USBF	4	0x2	6-9	PIPE%sCTR	PIPE%s Control Register	0x07A	16	R/W	0x0000	0xFFFF
USBF	5	0x4	1-5	PIPE%sTRE	PIPE%s Transaction Counter Enable Register	0x090	16	R/W	0x0000	0xFFFF
USBF	5	0x4	1-5	PIPE%sTRN	PIPE%s Transaction Counter Register	0x092	16	R/W	0x0000	0xFFFF
USBF	—	—	—	LPSTS	Low Power Status Register	0x102	16	R/W	0x0000	0xFFFF
USBF	2	0x40	0-1	N0SA_%s	Next Source Address Register %s	0x400	32	R/W	0x00000000	0xFFFFFFFF
USBF	2	0x40	0-1	N0DA_%s	Next Destination Address Register %s	0x404	32	R/W	0x00000000	0xFFFFFFFF
USBF	2	0x40	0-1	N0TB_%s	Next Transaction Byte Register %s	0x408	32	R/W	0x00000000	0xFFFFFFFF
USBF	2	0x40	0-1	N1SA_%s	Next Source Address Register %s	0x40C	32	R/W	0x00000000	0xFFFFFFFF
USBF	2	0x40	0-1	N1DA_%s	Next Destination Address Register %s	0x410	32	R/W	0x00000000	0xFFFFFFFF
USBF	2	0x40	0-1	N1TB_%s	Next Transaction Byte Register %s	0x414	32	R/W	0x00000000	0xFFFFFFFF
USBF	2	0x40	0-1	CRSA_%s	Current Source Address Register %s	0x418	32	R	0x00000000	0xFFFFFFFF
USBF	2	0x40	0-1	CRDA_%s	Current Destination Address Register %s	0x41C	32	R	0x00000000	0xFFFFFFFF
USBF	2	0x40	0-1	CRTB_%s	Current Transaction Byte Register %s	0x420	32	R	0x00000000	0xFFFFFFFF
USBF	2	0x40	0-1	CHSTAT_%s	Channel Status Register %s	0x424	32	R	0x00000000	0xFFFFFFFF
USBF	2	0x40	0-1	CHCTRL_%s	Channel Control Register %s	0x428	32	W	0x00000000	0xFFFFFFFF
USBF	2	0x40	0-1	CHCFG_%s	Channel Configuration Register %s	0x42C	32	R/W	0x00000000	0xFFFFFFFF
USBF	2	0x40	0-1	CHITVL_%s	Channel Interval Register %s	0x430	32	R/W	0x00000000	0xFFFFFFFF
USBF	2	0x40	0-1	CHEXT_%s	Channel Extension Register %s	0x434	32	R/W	0x00000000	0xFFFFFFFF
USBF	2	0x40	0-1	NXLA_%s	Next Link Address Register %s	0x438	32	R/W	0x00000000	0xFFFFFFFF
USBF	2	0x40	0-1	CRLA_%s	Current Link Address Register %s	0x43C	32	R	0x00000000	0xFFFFFFFF
USBF	2	0x20	0-1	SCNT_%s	Source Continuous Register %s	0x600	32	R/W	0x00000000	0xFFFFFFFF
USBF	2	0x20	0-1	SSKP_%s	Source Skip Register %s	0x604	32	R/W	0x00000000	0xFFFFFFFF
USBF	2	0x20	0-1	DCNT_%s	Destination Continuous Register %s	0x608	32	R/W	0x00000000	0xFFFFFFFF
USBF	2	0x20	0-1	DSKP_%s	Destination Skip Register %s	0x60C	32	R/W	0x00000000	0xFFFFFFFF
USBF	—	—	—	DCTRL	DMA Control Register	0x700	32	R/W	0x00000000	0xFFFFFFFF
USBF	—	—	—	DSCITVL	Descriptor Interval Register	0x704	32	R/W	0x00000000	0xFFFFFFFF
USBF	—	—	—	DSTAT_EN	DMA Status EN Register	0x710	32	R	0x00000000	0xFFFFFFFF
USBF	—	—	—	DSTAT_ER	DMA Status ER Register	0x714	32	R	0x00000000	0xFFFFFFFF
USBF	—	—	—	DSTAT_END	DMA Status END Register	0x718	32	R	0x00000000	0xFFFFFFFF
USBF	—	—	—	DSTAT_TC	DMA Status TC Register	0x71C	32	R	0x00000000	0xFFFFFFFF
USBF	—	—	—	DSTAT_SUS	DMA Status SUS Register	0x720	32	R	0x00000000	0xFFFFFFFF
BSC	—	—	—	CMNCR	Common Control Register	0x00	32	R/W	0x00001018	0xFFFFFFFF
BSC	—	—	—	CS0BCR	CS0 Space Bus Control Register	0x004	32	R/W	0x36DB0C00	0xFFFFFFFF
BSC	—	—	—	CS2BCR	CS2 Space Bus Control Register	0x00C	32	R/W	0x36DB0E00	0xFFFFFFFF
BSC	—	—	—	CS3BCR	CS3 Space Bus Control Register	0x010	32	R/W	0x36DB0E00	0xFFFFFFFF
BSC	—	—	—	CS5BCR	CS5 Space Bus Control Register	0x018	32	R/W	0x36DB0E00	0xFFFFFFFF
BSC	—	—	—	CS0WCR_0	CS0 Space Wait Control Register for Normal Space, SRAM with Byte Selection	0x028	32	R/W	0x00000500	0xFFFFFFFF
BSC	—	—	—	CS0WCR_1	CS0 Space Wait Control Register for Burst ROM with Clocked Asynchronous	0x028	32	R/W	0x00000500	0xFFFFFFFF
BSC	—	—	—	CS0WCR_2	CS0 Space Wait Control Register for Burst ROM with Clocked Synchronous	0x028	32	R/W	0x00000500	0xFFFFFFFF
BSC	—	—	—	CS2WCR_0	CS2 Space Wait Control Register for Normal Space, SRAM with Byte Selection	0x030	32	R/W	0x00000500	0xFFFFFFFF

Table 1.2 Register description (22 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
BSC	—	—	—	CS2WCR_1	CS2 Space Wait Control Register for SDRAM	0x030	32	R/W	0x00000500	0xFFFFFFFF
BSC	—	—	—	CS3WCR_0	CS3 Space Wait Control Register for Normal Space, SRAM with Byte Selection	0x034	32	R/W	0x00000500	0xFFFFFFFF
BSC	—	—	—	CS3WCR_1	CS3 Space Wait Control Register for SDRAM	0x034	32	R/W	0x00000500	0xFFFFFFFF
BSC	—	—	—	CS5WCR	CS5 Space Wait Control Register for Normal Space, SRAM with Byte Selection, and MPX-I/O	0x03C	32	R/W	0x00000500	0xFFFFFFFF
BSC	—	—	—	SDCR	SDRAM Control Register	0x04C	32	R/W	0x00000000	0xFFFFFFFF
BSC	—	—	—	RTCSR	Refresh Timer Control/Status Register	0x050	32	R/W	0x00000000	0xFFFFFFFF
BSC	—	—	—	RTCNT	Refresh Timer Counter	0x054	32	R/W	0x00000000	0xFFFFFFFF
BSC	—	—	—	RTCOR	Refresh Time Constant Register	0x058	32	R/W	0x00000000	0xFFFFFFFF
BSC	—	—	—	TOSCOR0	Timeout Cycle Constant Register 0	0x060	32	R/W	0x00000000	0xFFFFFFFF
BSC	—	—	—	TOSCOR2	Timeout Cycle Constant Register 2	0x068	32	R/W	0x00000000	0xFFFFFFFF
BSC	—	—	—	TOSCOR3	Timeout Cycle Constant Register 3	0x06C	32	R/W	0x00000000	0xFFFFFFFF
BSC	—	—	—	TOSCOR5	Timeout Cycle Constant Register 5	0x074	32	R/W	0x00000000	0xFFFFFFFF
BSC	—	—	—	TOSTR	Timeout Status Register	0x080	32	R/W	0x00000000	0xFFFFFFFF
BSC	—	—	—	TOENR	Timeout Enable Register	0x084	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	—	—	—	WRAPCFG	xSPI Wrapper Configuration Register	0x000	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	—	—	—	COMCFG	xSPI Common Configuration Register	0x004	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	—	—	—	BMCFG	xSPI Bridge Map Configuration Register	0x008	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	2	0x010	0-1	CMCFG0CS%s	xSPI Command Map Configuration Register 0 CSn	0x010	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	2	0x010	0-1	CMCFG1CS%s	xSPI Command Map Configuration Register 1 CSn	0x014	32	R/W	0x00080000	0xFFFFFFFF
XSPI0-1	2	0x010	0-1	CMCFG2CS%s	xSPI Command Map Configuration Register 2 CSn	0x018	32	R/W	0x00080000	0xFFFFFFFF
XSPI0-1	2	0x004	0-1	LIOCFGCS%s	xSPI Link I/O Configuration Register CSn	0x050	32	R/W	0x00070000	0xFFFFFFFF
XSPI0-1	—	—	—	BMCTL0	xSPI Bridge Map Control Register 0	0x060	32	R/W	0x000000FF	0xFFFFFFFF
XSPI0-1	—	—	—	BMCTL1	xSPI Bridge Map Control Register 1	0x064	32	W	0x00000000	0xFFFFFFFF
XSPI0-1	—	—	—	CMCTL	xSPI Command Map Control Register	0x068	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	—	—	—	CSSCTL	xSPI CS Size Control Register	0x06C	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	—	—	—	CDCTL0	xSPI Command Manual Control Register 0	0x070	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	—	—	—	CDCTL1	xSPI Command Manual Control Register 1	0x074	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	—	—	—	CDCTL2	xSPI Command Manual Control Register 2	0x078	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	4	0x010	0-3	CDTBUF%s	xSPI Command Manual Type Buf %s	0x080	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	4	0x10	0-3	CDABUF%s	xSPI Command Manual Address Buf %s	0x084	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	4	0x10	0-3	CDD0BUF%s	xSPI Command Manual Data 0 Buf %s	0x088	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	4	0x10	0-3	CDD1BUF%s	xSPI Command Manual Data 1 Buf %s	0x08C	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	—	—	—	LPCTL0	xSPI Link Pattern Control Register 0	0x100	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	—	—	—	LPCTL1	xSPI Link Pattern Control Register 1	0x104	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	—	—	—	LIOCTL	xSPI Link I/O Control Register	0x108	32	R/W	0x00030003	0xFFFFFFFF
XSPI0-1	2	0x020	0-1	CCCTL0CS%s	xSPI Command Calibration Control Register 0 CSn	0x130	32	R/W	0x1F000000	0xFFFFFFFF
XSPI0-1	2	0x020	0-1	CCCTL1CS%s	xSPI Command Calibration Control Register 1 CSn	0x134	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	2	0x020	0-1	CCCTL2CS%s	xSPI Command Calibration Control Register 2 CSn	0x138	32	R/W	0x00000000	0xFFFFFFFF

Table 1.2 Register description (23 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
XSPI0-1	2	0x020	0-1	CCCTL3CS%s	xSPI Command Calibration Control Register 3 CSn	0x13C	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	2	0x020	0-1	CCCTL4CS%s	xSPI Command Calibration Control Register 4 CSn	0x140	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	2	0x020	0-1	CCCTL5CS%s	xSPI Command Calibration Control Register 5 CSn	0x144	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	2	0x020	0-1	CCCTL6CS%s	xSPI Command Calibration Control Register 6 CSn	0x148	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	2	0x020	0-1	CCCTL7CS%s	xSPI Command Calibration Control Register 7 CSn	0x14C	32	R/W	0x00000000	0xFFFFFFFF
XSPI0-1	—	—	—	VERSTT	xSPI Version Register	0x180	32	R	0x00000000	0xFFFFFFFF
XSPI0-1	—	—	—	COMSTT	xSPI Common Status Register	0x184	32	R	0x00000000	0xFF8FFFFF
XSPI0-1	2	0x004	0-1	CASTTCS%s	xSPI Calibration Status Register CSn	0x188	32	R	0x00000000	0xFFFFFFFF
XSPI0-1	—	—	—	INTS	xSPI Interrupt Status Register	0x190	32	R	0x00000000	0xFFFFFFFF
XSPI0-1	—	—	—	INTC	xSPI Interrupt Clear Register	0x194	32	W	0x00000000	0xFFFFFFFF
XSPI0-1	—	—	—	INTE	xSPI Interrupt Enable Register	0x198	32	R/W	0x00000000	0xFFFFFFFF
MBXSEM	8	0x04	0-7	SEM%s	Semaphore Register %s	0x00	32	R/W	0x00000000	0xFFFFFFFF
MBXSEM	—	—	—	SEMRCEN	Semaphore Read Clear Enable Register	0x20	32	R/W	0x00000000	0xFFFFFFFF
MBXSEM	4	0x04	0-3	MBXH2C%s	Host to CR52 Mailbox Register %s	0x40	32	R	0x00000000	0xFFFFFFFF
MBXSEM	—	—	—	MBXISETH2C	Host to CR52 Mailbox Interrupt Set Register	0x50	32	R	0x00000000	0xFFFFFFFF
MBXSEM	—	—	—	MBXICLRH2C	Host to CR52 Mailbox Interrupt Clear Register	0x54	32	R/W	0x00000000	0xFFFFFFFF
MBXSEM	4	0x04	0-3	MBXC2H%s	CR52 to Host Mailbox Register %s	0x80	32	R/W	0x00000000	0xFFFFFFFF
MBXSEM	—	—	—	MBXISETC2H	CR52 to Host Mailbox Interrupt Set Register	0x90	32	R/W	0x00000000	0xFFFFFFFF
MBXSEM	—	—	—	MBXICLRC2H	CR52 to Host Mailbox Interrupt Clear Register	0x94	32	R	0x00000000	0xFFFFFFFF
SHOSTIF	—	—	—	CTRLR0	Control Register 0	0x000	32	R/W	0x00000000	0xFFFFFFFF
SHOSTIF	—	—	—	ENR	Enable Register	0x008	32	R/W	0x00000000	0xFFFFFFFF
SHOSTIF	—	—	—	RXFBTR	Receive FIFO Burst Threshold Register	0x014	32	R/W	0x0000003F	0xFFFFFFFF
SHOSTIF	—	—	—	TXFTLR	Transmit FIFO Threshold Level Register	0x018	32	R/W	0x00000000	0xFFFFFFFF
SHOSTIF	—	—	—	RXFTLR	Receive FIFO Threshold Level Register	0x01C	32	R/W	0x00000000	0xFFFFFFFF
SHOSTIF	—	—	—	SR	Status Register	0x028	32	R	0x00000000	0xFFFFFFFF
SHOSTIF	—	—	—	IMR	Interrupt Mask Register	0x02C	32	R/W	0x00000389	0xFFFFFFFF
SHOSTIF	—	—	—	ISR	Interrupt Status Register	0x030	32	R	0x00000000	0xFFFFFFFE
SHOSTIF	—	—	—	RISR	Raw Interrupt Status Register	0x034	32	R	0x00000000	0xFFFFFFFE
SHOSTIF	—	—	—	TXUICR	Transmit FIFO Underflow Interrupt Clear Register	0x038	32	R	0x00000000	0xFFFFFFFF
SHOSTIF	—	—	—	RXOICR	Receive FIFO Overflow Interrupt Clear Register	0x03C	32	R	0x00000000	0xFFFFFFFF
SHOSTIF	—	—	—	SPIMECR	SPI Master Interrupt Clear Register	0x040	32	R	0x00000000	0xFFFFFFFF
SHOSTIF	—	—	—	AHBECR	AHB Error Clear Register	0x044	32	R	0x00000000	0xFFFFFFFF
SHOSTIF	—	—	—	ICR	Interrupt Clear Register	0x048	32	R	0x00000000	0xFFFFFFFF
SYSC_NS	—	—	—	SCKCR	System Clock Control Register	0x00	32	R/W	0x00720606	0xFFFFFFFF
SYSC_NS	—	—	—	RSTSR0	Reset Status Register 0	0x200	32	R/W	0x00000000	0xFFFFFFE1
SYSC_NS	—	—	—	MRCTLA	Module Reset Control Register A	0x240	32	R/W	0x00000000	0xFFFFFFFF
SYSC_NS	—	—	—	MRCTLE	Module Reset Control Register E	0x250	32	R/W	0x0000007B	0xFFFFFFFF
SYSC_NS	—	—	—	MRCTLJ	Module Reset Control Register J	0x264	32	R/W	0x00000000	0xFFFFFFFF
SYSC_NS	—	—	—	MRCTLK	Module Reset Control Register K	0x268	32	R/W	0x00000000	0xFFFFFFFF
SYSC_NS	—	—	—	MSTPCRA	Module Stop Control Register A	0x0300	32	R/W	0x00001E00	0xFFFFFECE

Table 1.2 Register description (24 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
SYSC_NS	—	—	—	MSTPCRB	Module Stop Control Register B	0x0304	32	R/W	0x00000073	0xFFFFFFFF
SYSC_NS	—	—	—	MSTPCRC	Module Stop Control Register C	0x308	32	R/W	0x000000E7	0xFFFFFFFF
SYSC_NS	—	—	—	MSTPCRD	Module Stop Control Register D	0x30C	32	R/W	0x00000FFF	0xFFFFFFFF
SYSC_NS	—	—	—	MSTPCRE	Module Stop Control Register E	0x310	32	R/W	0x0000010D	0xFFFFFFFF
SYSC_NS	—	—	—	MSTPCRJ	Module Stop Control Register J	0x324	32	R/W	0x0000000F	0xFFFFFFFF
SYSC_NS	—	—	—	MSTPCRK	Module Stop Control Register K	0x328	32	R/W	0x0000000F	0xFFFFFFFF
SYSC_NS	—	—	—	MSTPCRL	Module Stop Control Register L	0x32C	32	R/W	0x00000001	0xFFFFFFFF
SYSC_NS	—	—	—	MD_MON	Operating Mode Monitor Register	0x1100	32	R	0x00000000	0xFFE08FEE
ELO	—	—	—	ELOPA	Event Link Option Setting Register A	0x00	32	R/W	0x000000FF	0xFFFFFFFF
ELO	—	—	—	ELOPB	Event Link Option Setting Register B	0x04	32	R/W	0x000000FF	0xFFFFFFFF
ENCSS	—	—	—	ENCODER_CFG	Encoder IF Configuration Register	0x00	32	R/W	0x00000000	0xFFFFFFFF
ENCSS	—	—	—	BISS_STAT	BiSS-C Status Register	0x04	32	R	0x00110011	0xFFFFFFFF
ENCSS	—	—	—	HDSL_STAT	HIPERFACE DSL Status Register	0x08	32	R	0x01F001F0	0xFFFFFFFF
ENCSS	—	—	—	ENDAT_CFG	EnDat Configuration Register	0x0C	32	R/W	0x00010000	0xFFFFFFFF
ENCSS	—	—	—	AFMT_STAT	A-Format Status Register	0x10	32	R	0x00030003	0xFFFFFFFF
ENCSS	2	0xC	0-1	HDSL%s_OS_D	HIPERFACE_DSL (SSn) Online Status (Drive) Register	0x14	32	R	0x00001B04	0xFFFFFFFF
ENCSS	—	—	—	HDSL0_OS_1	HIPERFACE_DSL (SS0) Online Status (Safe1) Register	0x18	32	R	0x00003504	0xFFFFFFFF
ENCSS	—	—	—	HDSL0_OS_2	HIPERFACE_DSL (SS0) Online Status (Safe2) Register	0x1C	32	R	0x00003104	0xFFFFFFFF
RWP_NS	—	—	—	PRCRN	Non_Safety Area Protect Register	0x00	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTWP	General PWM Timer Write-Protection Register	0x00	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTSTR	General PWM Timer Software Start Register	0x04	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTSTP	General PWM Timer Software Stop Register	0x08	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT14-17	—	—	—	GTCLR	General PWM Timer Software Clear Register	0x0C	32	W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTSSR	General PWM Timer Start Source Select Register	0x10	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTPSR	General PWM Timer Stop Source Select Register	0x14	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTCSR	General PWM Timer Clear Source Select Register	0x18	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTUPSR	General PWM Timer Count-Up Source Select Register	0x1C	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTDNSR	General PWM Timer Count-Down Source Select Register	0x20	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTICASR	General PWM Timer Input Capture Source Select Register A	0x24	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTICBSR	General PWM Timer Input Capture Source Select Register B	0x28	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTCR	General PWM Timer Control Register	0x2C	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTUDDTYC	General PWM Timer Count Direction and Duty Setting Register	0x30	32	R/W	0x00000001	0xFFFFFFFF
GPT14-17	—	—	—	GTIOR	General PWM Timer I/O Control Register	0x34	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTINTAD	General PWM Timer Interrupt Output Setting Register	0x38	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTST	General PWM Timer Status Register	0x3C	32	R/W	0x00008000	0xFFFFFFFF
GPT14-17	—	—	—	GTBER	General PWM Timer Buffer Enable Register	0x40	32	R/W	0x00000000	0xFFFFFFFF

Table 1.2 Register description (25 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
GPT14-17	—	—	—	GTITC	General PWM Timer Interrupt and A/D Converter Start Request Skipping Setting Register	0x44	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTCNT	General PWM Timer Counter	0x48	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTCCRA	General PWM Timer Compare Capture Register A (m = A to F)	0x4C	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT14-17	—	—	—	GTCCRB	General PWM Timer Compare Capture Register B (m = A to F)	0x50	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT14-17	—	—	—	GTCCRC	General PWM Timer Compare Capture Register C (m = A to F)	0x54	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT14-17	—	—	—	GTCCRE	General PWM Timer Compare Capture Register E (m = A to F)	0x58	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT14-17	—	—	—	GTCCRD	General PWM Timer Compare Capture Register D (m = A to F)	0x5C	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT14-17	—	—	—	GTCCRF	General PWM Timer Compare Capture Register F (m = A to F)	0x60	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT14-17	—	—	—	GTPR	General PWM Timer Cycle Setting Register	0x64	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT14-17	—	—	—	GTPBR	General PWM Timer Cycle Setting Buffer Register	0x68	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT14-17	—	—	—	GTPDBR	General PWM Timer Cycle Setting Double-Buffer Register	0x6C	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT14-17	—	—	—	GTDTCR	General PWM Timer Dead Time Control Register	0x88	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTDVU	General PWM Timer Dead Time Value Register U (m = U, D)	0x8C	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT14-17	—	—	—	GTDVD	General PWM Timer Dead Time Value Register D (m = U, D)	0x90	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT14-17	—	—	—	GTDBU	General PWM Timer Dead Time Value Buffer Register U (m = U, D)	0x94	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT14-17	—	—	—	GTDBD	General PWM Timer Dead Time Value Buffer Register D (m = U, D)	0x98	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT14-17	—	—	—	GTSOS	General PWM Timer Output Protection Function Status Register	0x9C	32	R	0x00000000	0xFFFFFCFF
GPT14-17	—	—	—	GTSOTR	General PWM Timer Output Protection Function Temporary Release Register	0xA0	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTEITC	General PWM Timer Extended Interrupt Skipping Counter Control Register	0xA8	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTEITLI1	General PWM Timer Extended Interrupt Skipping Setting Register 1	0xAC	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTEITLI2	General PWM Timer Extended Interrupt Skipping Setting Register 2	0xB0	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTEITLB	General PWM Timer Extended Buffer Transfer Skipping Setting Register	0xB4	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTSECSR	General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register	0xD0	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTSECR	General PWM Timer Operation Enable Bit Simultaneous Control Register	0xD4	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTSWSR	General PWM Timer Switch Source Select Register	0xD8	32	R/W	0x00000000	0xFFFFFFFF
GPT14-17	—	—	—	GTSWOS	General PWM Timer Switch Offset Setting Register	0xDC	32	R/W	0x00000000	0xFFFFFFFF
SCI5	—	—	—	RDR	Receive Data Register	0x00	32	R	0x00000000	0xFFFFFFFF
SCI5	—	—	—	TDR	Transmit Data Register	0x04	32	R/W	0xFFFFFFFF	0xFFFFFFFF
SCI5	—	—	—	CCR0	Common Control Register 0	0x08	32	R/W	0x00000000	0xFFFFFFFF
SCI5	—	—	—	CCR1	Common Control Register 1	0x0C	32	R/W	0x00000010	0xFFFFFFFF
SCI5	—	—	—	CCR2	Common Control Register 2	0x10	32	R/W	0xFF00FF04	0xFFFFFFFF
SCI5	—	—	—	CCR3	Common Control Register 3	0x14	32	R/W	0x00001203	0xFFFFFFFF

Table 1.2 Register description (26 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
SCI5	—	—	—	CCR4	Common Control Register 4	0x18	32	R/W	0x00000000	0xFFFFFFFF
SCI5	—	—	—	ICR	Simple I2C Control Register	0x20	32	R/W	0x00000000	0xFFFFFFFF
SCI5	—	—	—	FCR	FIFO Control Register	0x24	32	R/W	0x1F1F0000	0xFFFFFFFF
SCI5	—	—	—	DCR	Driver Control Register	0x30	32	R/W	0x00000000	0xFFFFFFFF
SCI5	—	—	—	CSR	Common Status Register	0x48	32	R	0x60008000	0xFFFFFFFF
SCI5	—	—	—	ISR	Simple I2C Status Register	0x4C	32	R	0x00000000	0xFFFFFFCB
SCI5	—	—	—	FRSR	FIFO Receive Status Register	0x50	32	R	0x00000000	0xFFFFFFFF
SCI5	—	—	—	FTSR	FIFO Transmit Status Register	0x54	32	R	0x00000000	0xFFFFFFFF
SCI5	—	—	—	CFCLR	Common Flag Clear Register	0x68	32	W	0x00000000	0xFFFFFFFF
SCI5	—	—	—	ICFCLR	Simple I2C Flag Clear Register	0x6C	32	W	0x00000000	0xFFFFFFFF
SCI5	—	—	—	FFCLR	FIFO Flag Clear Register	0x70	32	W	0x00000000	0xFFFFFFFF
SPI3	—	—	—	SPDR	SPI Data Register	0x00	32	R/W	0x00000000	0xFFFFFFFF
SPI3	—	—	—	SPDR_HA	SPI Data Register	0x00	16	R/W	0x0000	0xFFFF
SPI3	—	—	—	SPDR_BY	SPI Data Register	0x00	8	R/W	0x00	0xFF
SPI3	—	—	—	SPCKD	SPI Clock Delay Register	0x04	8	R/W	0x00	0xFF
SPI3	—	—	—	SSLND	SPI Slave Select Negation Delay Register	0x05	8	R/W	0x00	0xFF
SPI3	—	—	—	SPND	SPI Next-Access Delay Register	0x06	8	R/W	0x00	0xFF
SPI3	—	—	—	MRCKD	SPI ClocK Digital control Register for Master Receive	0x07	8	R/W	0x00	0xFF
SPI3	—	—	—	SPCR	SPI Control Register	0x08	32	R/W	0x00000000	0xFFFFFFFF
SPI3	—	—	—	SPCRRM	SPI Control Register for Master Receive only	0x0C	8	R/W	0x00	0xFF
SPI3	—	—	—	SPDRCR	SPI Control Register for Received Data Ready Detection	0x0D	8	R/W	0x00	0xFF
SPI3	—	—	—	SPPCR	SPI Pin Control Register	0x0E	8	R/W	0x00	0xFF
SPI3	—	—	—	SPCR2	SPI Control Register 2	0x0F	8	R/W	0x00	0xFF
SPI3	—	—	—	SSLP	SPI Slave Select Polarity Register	0x10	8	R/W	0x00	0xFF
SPI3	—	—	—	SPBR	SPI Bit Rate Register	0x11	8	R/W	0xFF	0xFF
SPI3	—	—	—	SPSCR	SPI Sequence Control Register	0x13	8	R/W	0x00	0xFF
SPI3	8	0x04	0-7	SPCMD%s	SPI Command Register %s (m = 0 to 7)	0x14	32	R/W	0x00070000	0xFFFFFFFF
SPI3	—	—	—	SPDCR	SPI Data Control Register	0x40	16	R/W	0x0000	0xFFFF
SPI3	—	—	—	SPDCR2	SPI Data Control Register 2	0x44	16	R/W	0x0000	0xFFFF
SPI3	—	—	—	SPSSR	SPI Sequence Status Register	0x51	8	R	0x00	0xFF
SPI3	—	—	—	SPSR	SPI Status Register	0x52	16	R	0x2000	0xFFFF
SPI3	—	—	—	SPTFSR	SPI Transfer FIFO Status Register	0x58	8	R	0x04	0xFF
SPI3	—	—	—	SPRFSR	SPI Receive FIFO Status Register	0x5C	8	R	0x00	0xFF
SPI3	—	—	—	SPPSR	SPI Poling Register	0x60	32	R	0x00000000	0xFFFFFFFF
SPI3	—	—	—	SPSRC	SPI Status Clear Register	0x6A	16	R/W	0x0000	0xFFFF
SPI3	—	—	—	SPFCR	SPI FIFO Clear Register	0x6C	8	W	0x00	0xFF
CRC1	—	—	—	CRCCR0	CRC Control Register 0	0x0	8	R/W	0x00	0xFF
CRC1	—	—	—	CRCDIR	CRC Data Input Register	0x4	32	R/W	0x00000000	0xFFFFFFFF
CRC1	—	—	—	CRCDIR_BY	CRC Data Input Register	0x4	8	R/W	0x00	0xFF
CRC1	—	—	—	CRCDOR	CRC Data Output Register	0x8	32	R/W	0x00000000	0xFFFFFFFF
CRC1	—	—	—	CRCDOR_HA	CRC Data Output Register	0x8	16	R/W	0x0000	0xFFFF
CRC1	—	—	—	CRCDOR_BY	CRC Data Output Register	0x8	8	R/W	0x00	0xFF
IIC2	—	—	—	ICCR1	I2C Bus Control Register 1	0x00	8	R/W	0x1F	0xFF
IIC2	—	—	—	ICCR2	I2C Bus Control Register 2	0x01	8	R/W	0x00	0xFF

Table 1.2 Register description (27 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
IIC2	—	—	—	ICMR1	I2C Bus Mode Register 1	0x02	8	R/W	0x08	0xFF
IIC2	—	—	—	ICMR2	I2C Bus Mode Register 2	0x03	8	R/W	0x06	0xFF
IIC2	—	—	—	ICMR3	I2C Bus Mode Register 3	0x04	8	R/W	0x00	0xFF
IIC2	—	—	—	ICFER	I2C Bus Function Enable Register	0x05	8	R/W	0x72	0xFF
IIC2	—	—	—	ICSER	I2C Bus Status Enable Register	0x06	8	R/W	0x09	0xFF
IIC2	—	—	—	ICIER	I2C Bus Interrupt Enable Register	0x07	8	R/W	0x00	0xFF
IIC2	—	—	—	ICSR1	I2C Bus Status Register 1	0x08	8	R/W	0x00	0xFF
IIC2	—	—	—	ICSR2	I2C Bus Status Register 2	0x09	8	R/W	0x00	0xFF
IIC2	3	0x02	0-2	SARL%s	Slave Address Register L %s (y = 0 to 2)	0x0A	8	R/W	0x00	0xFF
IIC2	3	0x02	0-2	SARU%s	Slave Address Register U %s (y = 0 to 2)	0x0B	8	R/W	0x00	0xFF
IIC2	—	—	—	ICBRL	I2C Bus Bit Rate Low-Level Register	0x10	8	R/W	0xFF	0xFF
IIC2	—	—	—	ICBRH	I2C Bus Bit Rate High-Level Register	0x11	8	R/W	0xFF	0xFF
IIC2	—	—	—	ICDRT	I2C Bus Transmit Data Register	0x12	8	R/W	0xFF	0xFF
IIC2	—	—	—	ICDRR	I2C Bus Receive Data Register	0x13	8	R	0x00	0xFF
RTC	—	—	—	RTCA0CTL0	RTC Control Register 0	0x00	32	R/W	0x00000000	0xFFFFFFFF
RTC	—	—	—	RTCA0CTL1	RTC Control Register 1	0x04	32	R/W	0x00000000	0xFFFFFFFF
RTC	—	—	—	RTCA0CTL2	RTC Control Register 2	0x08	32	R/W	0x00000000	0xFFFFFFFF
RTC	—	—	—	RTCA0SUBC	RTC Sub Count Register	0x0C	32	R	0x00000000	0xFFFFFFFF
RTC	—	—	—	RTCA0SRBU	RTC Sub Count Register Read Buffer	0x10	32	R	0x00000000	0xFFFFFFFF
RTC	—	—	—	RTCA0SEC	RTC Sec Count Buffer Register	0x14	32	R/W	0x00000000	0xFFFFFFFF
RTC	—	—	—	RTCA0MIN	RTC Min Count Buffer Register	0x18	32	R/W	0x00000000	0xFFFFFFFF
RTC	—	—	—	RTCA0HOUR	RTC Hour Count Buffer Register	0x1C	32	R/W	0x00000012	0xFFFFFFFF
RTC	—	—	—	RTCA0WEEK	RTC Week Count Buffer Register	0x20	32	R/W	0x00000000	0xFFFFFFFF
RTC	—	—	—	RTCA0DAY	RTC Day Count Buffer Register	0x24	32	R/W	0x00000001	0xFFFFFFFF
RTC	—	—	—	RTCA0MONTH	RTC Month Count Buffer Register	0x28	32	R/W	0x00000001	0xFFFFFFFF
RTC	—	—	—	RTCA0YEAR	RTC Year Count Buffer Register	0x2C	32	R/W	0x00000000	0xFFFFFFFF
RTC	—	—	—	RTCA0TIME	RTC Time Set Register	0x30	32	R/W	0x00120000	0xFFFFFFFF
RTC	—	—	—	RTCA0CAL	RTC Calendar Set Register	0x34	32	R/W	0x00010100	0xFFFFFFFF
RTC	—	—	—	RTCA0SCMP	RTC Sub Count Compare Register	0x3C	32	R/W	0x00000000	0xFFFFFFFF
RTC	—	—	—	RTCA0ALM	RTC Alarm Min Set Register	0x40	32	R/W	0x00000000	0xFFFFFFFF
RTC	—	—	—	RTCA0ALH	RTC Alarm Hour Set Register	0x44	32	R/W	0x00000012	0xFFFFFFFF
RTC	—	—	—	RTCA0ALW	RTC Alarm Week Set Register	0x48	32	R/W	0x00000000	0xFFFFFFFF
RTC	—	—	—	RTCA0SECC	RTC Second Count Register	0x4C	32	R	0x00000000	0xFFFFFFFF
RTC	—	—	—	RTCA0MINC	RTC Minute Count Register	0x50	32	R	0x00000000	0xFFFFFFFF
RTC	—	—	—	RTCA0HOURC	RTC Hour Count Register	0x54	32	R	0x00000012	0xFFFFFFFF
RTC	—	—	—	RTCA0WEEKC	RTC Week Count Register	0x58	32	R	0x00000000	0xFFFFFFFF
RTC	—	—	—	RTCA0DAYC	RTC Day Count Register	0x5C	32	R	0x00000001	0xFFFFFFFF
RTC	—	—	—	RTCA0MONC	RTC Month Count Register	0x60	32	R	0x00000001	0xFFFFFFFF
RTC	—	—	—	RTCA0YEARC	RTC Year Count Register	0x64	32	R	0x00000000	0xFFFFFFFF
RTC	—	—	—	RTCA0TIMEC	RTC Time Count Register	0x68	32	R	0x00120000	0xFFFFFFFF
RTC	—	—	—	RTCA0CALC	RTC Calendar Count Register	0x6C	32	R	0x00010100	0xFFFFFFFF
POEG2	—	—	—	POEG2GA	POEG2 Group A Setting Register	0x000	32	R/W	0x00000000	0xFFFFFFFF
POEG2	—	—	—	POEG2GB	POEG2 Group B Setting Register	0x400	32	R/W	0x00000000	0xFFFFFFFF
POEG2	—	—	—	POEG2GC	POEG2 Group C Setting Register	0x800	32	R/W	0x00000000	0xFFFFFFFF
POEG2	—	—	—	POEG2GD	POEG2 Group D Setting Register	0xC00	32	R/W	0x00000000	0xFFFFFFFF
OTP	—	—	—	OTPPWR	OTP Power Control Register	0x00	32	R/W	0x00000001	0xFFFFFFFF
OTP	—	—	—	OTPSTR	OTP Access Status Register	0x04	32	R/W	0x00008000	0xFFFFFFFF

Table 1.2 Register description (28 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
OTP	—	—	—	OTPSTAWR	OTP Write Command Register	0x08	32	R/W	0x00000000	0xFFFFFFFF
OTP	—	—	—	OTPADRWR	OTP Write Address Register	0x0C	32	R/W	0x00000000	0xFFFFFFFF
OTP	—	—	—	OTPDATAWR	OTP Write Data Register	0x10	32	R/W	0x00000000	0xFFFFFFFF
OTP	—	—	—	OTPADRRD	OTP Read Address Register	0x14	32	R/W	0x00000000	0xFFFFFFFF
OTP	—	—	—	OTPDATARD	OTP Read Data Register	0x18	32	R	0x00000000	0xFFFFFFFF
PORT_SR	10	0x001	0-9	P0%s	Port %s Register	0x000	8	R/W	0x00	0xFF
PORT_SR	15	0x001	10-24	P%s	Port %s Register	0x00A	8	R/W	0x00	0xFF
PORT_SR	10	0x002	0-9	PM0%s	Port %s Mode Register	0x200	16	R/W	0x0000	0xFFFF
PORT_SR	15	0x002	10-24	PM%s	Port %s Mode Register	0x214	16	R/W	0x0000	0xFFFF
PORT_SR	10	0x001	0-9	PMC0%s	Port %s Mode Control Register	0x400	8	R/W	0x00	0xFF
PORT_SR	15	0x001	10-24	PMC%s	Port %s Mode Control Register	0x40A	8	R/W	0x00	0xFF
PORT_SR	10	0x004	0-9	PFC0%s	Port %s Function Control Register	0x600	32	R/W	0x00000000	0xFFFFFFFF
PORT_SR	15	0x004	10-24	PFC%s	Port %s Function Control Register	0x628	32	R/W	0x00000000	0xFFFFFFFF
PORT_SR	10	0x001	0-9	PIN0%s	Port %s Input Register	0x800	8	R	0x00	0xFF
PORT_SR	15	0x001	10-24	PIN%s	Port %s Input Register	0x80A	8	R	0x00	0xFF
PORT_SR	10	0x008	0-9	DRCTL0%s_L	I/O Buffer %s Function Switching Register	0xA00	32	R/W	0x00000000	0xFFFFFFFF
PORT_SR	10	0x008	0-9	DRCTL0%s_H	I/O Buffer %s Function Switching Register	0xA04	32	R/W	0x00000000	0xFFFFFFFF
PORT_SR	15	0x008	10-24	DRCTL%s_L	I/O Buffer %s Function Switching Register	0xA50	32	R/W	0x00000000	0xFFFFFFFF
PORT_SR	15	0x008	10-24	DRCTL%s_H	I/O Buffer %s Function Switching Register	0xA54	32	R/W	0x00000000	0xFFFFFFFF
PORT_SR	2	0x01	1-2	ELC_PGR%s	ELC Port Group Setting Register %s	0xE00	8	R/W	0x00	0xFF
PORT_SR	2	0x01	1-2	ELC_PGC%s	ELC Port Group Control Register %s	0xE02	8	R/W	0x88	0xFF
PORT_SR	2	0x04	1-2	ELC_PDBF%s	ELC Port Buffer Register %s	0xE04	8	R/W	0x00	0xFF
PORT_SR	4	0x01	0-3	ELC_PEL%s	ELC Port Setting Register %s	0xE0C	8	R/W	0x80	0xFF
PORT_SR	—	—	—	ELC_DPTC	ELC Edge Detection Control Register	0xE10	8	R/W	0x00	0xFF
PORT_SR	—	—	—	ELC_ELSR2	ELC Port Event Control Register	0xE11	8	R/W	0x00	0xFF
PTADR	10	0x001	0-9	RSELP0%s	Port %s Region Select Register	0x000	8	R/W	0x00	0xFF
PTADR	15	0x001	10-24	RSELP%s	Port %s Region Select Register	0x00A	8	R/W	0x00	0xFF
SYSRAM0-1	4	0x040	0-3	W%s_EC710CTL	ECC Control Register %s	0x000	32	R/W	0x00000010	0xFFFFFFFF
SYSRAM0-1	4	0x040	0-3	W%s_EC710TMC	ECC Test Mode Control Register %s	0x004	32	R/W	0x00000000	0xFFFFFFFF
SYSRAM0-1	4	0x040	0-3	W%s_EC710TRC	ECC Redundancy Bit Data Control Test Register %s	0x008	32	R/W	0x00000000	0xFFFFFFFF
SYSRAM0-1	4	0x040	0-3	W%s_EC710TED	ECC Encode/Decode Input/Output Switchover Test Register %s	0x00C	32	R/W	0x00000000	0xFFFFFFFF
SYSRAM0-1	8	0x004	0-7	W0_EC710EAD%s	ECC Error Address %s Register 0	0x010	32	R	0x00000000	0xFFFFFFFF
SYSRAM0-1	8	0x004	0-7	W1_EC710EAD%s	ECC Error Address %s Register 1	0x050	32	R	0x00000000	0xFFFFFFFF
SYSRAM0-1	8	0x004	0-7	W2_EC710EAD%s	ECC Error Address %s Register 2	0x090	32	R	0x00000000	0xFFFFFFFF
SYSRAM0-1	8	0x004	0-7	W3_EC710EAD%s	ECC Error Address %s Register 3	0x0D0	32	R	0x00000000	0xFFFFFFFF
ICU	—	—	—	S_SWINT	Software Interrupt Register for Safety Register	0x000	32	W	0x00000000	0xFFFFFFFF
ICU	—	—	—	S_PORTNF_FLTSEL	Interrupt Noise Filter Enable Register for Safety Register	0x004	32	R/W	0x00000000	0xFFFFFFFF
ICU	—	—	—	S_PORTNF_CLKSEL	Interrupt Noise Filter Setting Register for Safety Register	0x008	32	R/W	0x00000000	0xFFFFFFFF
ICU	—	—	—	S_PORTNF_MD	Interrupt Edge Detection Setting Register for Safety Register	0x00C	32	R/W	0x00000015	0xFFFFFFFF
ICU	—	—	—	CPU0ERR_STAT	CPU0 Error Event Status Register	0x060	32	R	0x00000000	0xFFFFFFFF
ICU	—	—	—	PERIERR_STAT0	Peripheral Error Event Status Register 0	0x068	32	R	0x00000000	0xFFFFFFFF

Table 1.2 Register description (29 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
ICU	—	—	—	PERIERR_STAT1	Peripheral Error Event Status Register 1	0x06C	32	R	0x00000000	0xFFFFFFFF
ICU	—	—	—	CPU0ERR_CLR	CPU0 Error Event Status Clear Register	0x070	32	W	0x00000000	0xFFFFFFFF
ICU	—	—	—	PERIERR_CLR0	Peripheral Error Event Status Clear Register 0	0x078	32	W	0x00000000	0xFFFFFFFF
ICU	—	—	—	PERIERR_CLR1	Peripheral Error Event Status Clear Register 1	0x07C	32	W	0x00000000	0xFFFFFFFF
ICU	—	—	—	CPU0ERR_RSTMSK	CPU0 Error Event Reset Mask Register	0x080	32	R/W	0x07FFFFFF	0xFFFFFFFF
ICU	—	—	—	PERIERR_RSTMSK0	Peripheral Error Event Reset Mask Register 0	0x088	32	R/W	0xFFFFFEFF	0xFFFFFFFF
ICU	—	—	—	PERIERR_RSTMSK1	Peripheral Error Event Reset Mask Register 1	0x08C	32	R/W	0xF9FEA07F	0xFFFFFFFF
ICU	—	—	—	CPU0ERR_E0MSK	CPU0 E0 Error Event Mask Register	0x090	32	R/W	0x07FFFFFF	0xFFFFFFFF
ICU	—	—	—	PERIERR_E0MSK0	Peripheral E0 Error Event Mask Register 0	0x098	32	R/W	0xFFFFFEFF	0xFFFFFFFF
ICU	—	—	—	PERIERR_E0MSK1	Peripheral E0 Error Event Mask Register 1	0x09C	32	R/W	0x19FEA07F	0xFFFFFFFF
ICU	—	—	—	CPU0ERR_E1MSK	CPU0 E1 Error Event Mask Register	0x100	32	R/W	0x07FFFFFF	0xFFFFFFFF
ICU	—	—	—	PERIERR_E1MSK0	Peripheral E1 Error Event Mask Register 0	0x108	32	R/W	0xFFFFFEFF	0xFFFFFFFF
ICU	—	—	—	PERIERR_E1MSK1	Peripheral E1 Error Event Mask Register 1	0x10C	32	R/W	0x19FEA07F	0xFFFFFFFF
ICU	—	—	—	PERIERR_STAT2	Peripheral Error Event Status Register 2	0x120	32	R	0x00000000	0xFFFFFFFF
ICU	—	—	—	PERIERR_STAT3	Peripheral Error Event Status Register 3	0x124	32	R	0x13E7E000	0xFFFFFFFF
ICU	—	—	—	PERIERR_CLR2	Peripheral Error Event Status Clear Register 2	0x130	32	W	0x00000000	0xFFFFFFFF
ICU	—	—	—	PERIERR_CLR3	Peripheral Error Event Status Clear Register 3	0x134	32	W	0x00000000	0xFFFFFFFF
ICU	—	—	—	PERIERR_RSTMSK2	Peripheral Error Event Reset Mask Register 2	0x140	32	R/W	0xFFFFFFFF	0xFFFFFFFF
ICU	—	—	—	PERIERR_RSTMSK3	Peripheral Error Event Reset Mask Register 3	0x144	32	R/W	0x7FFFFFFF	0xFFFFFFFF
ICU	—	—	—	PERIERR_E0MSK2	Peripheral E0 Error Event Mask Register 2	0x150	32	R/W	0xFFFFFFFF	0xFFFFFFFF
ICU	—	—	—	PERIERR_E0MSK3	Peripheral E0 Error Event Mask Register 3	0x154	32	R/W	0x7FFFFFFF	0xFFFFFFFF
ICU	—	—	—	PERIERR_E1MSK2	Peripheral E1 Error Event Mask Register 2	0x1C0	32	R/W	0xFFFFFFFF	0xFFFFFFFF
ICU	—	—	—	PERIERR_E1MSK3	Peripheral E1 Error Event Mask Register 3	0x1C4	32	R/W	0x7FFFFFFF	0xFFFFFFFF
SYSC_S	—	—	—	SCKCR2	System Clock Control Register 2	0x04	32	R/W	0x00000012	0xFFFFFFFF
SYSC_S	—	—	—	PLL0MON	PLL0 Monitor Register	0x20	32	R	0x00000001	0xFFFFFFFF
SYSC_S	—	—	—	PLL1MON	PLL1 Monitor Register	0x40	32	R	0x00000000	0xFFFFFFFE
SYSC_S	—	—	—	PLL1EN	PLL1 Enable Register	0x50	32	R/W	0x00000001	0xFFFFFFFF
SYSC_S	—	—	—	LOCOCR	Low-Speed On-Chip Oscillator Control Register	0x70	32	R/W	0x00000000	0xFFFFFFFF
SYSC_S	—	—	—	HIZCTRLEN	High-Impedance Control Enable Register	0x80	32	R/W	0x00000003	0xFFFFFFFF
SYSC_S	—	—	—	SWRSYS	System Software Reset Register	0x210	32	W	0x00000000	0xFFFFFFFF
SYSC_S	—	—	—	SWRCPU0	CPU0 Software Reset Register	0x220	32	R/W	0x00000000	0xFFFFFFFF
SYSC_S	—	—	—	MRCTLI	Module Reset Control Register I	0x0260	32	R/W	0x0000000E	0xFFFFFFFF
SYSC_S	—	—	—	MSTPCRF	Module Stop Control Register F	0x314	32	R/W	0x00000000	0xFFFFFFFF
SYSC_S	—	—	—	MSTPCRG	Module Stop Control Register G	0x318	32	R/W	0x00000F3F	0xFFFFFFFF
SYSC_S	—	—	—	MSTPCRI	Module Stop Control Register I	0x320	32	R/W	0x00000002	0xFFFFFFFF
CLMA0-3	—	—	—	CTL0	CLMA Control Register 0	0x00	8	R/W	0x00	0xFF
CLMA0-3	—	—	—	CMPL	CLMA Compare Register L	0x08	16	R/W	0x0001	0xFFFF

Table 1.2 Register description (30 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
CLMA0-3	—	—	—	CMPH	CLMA Compare Register H	0x0C	16	R/W	0x03FF	0xFFFF
CLMA0-3	—	—	—	PCMD	CLMA Command Register	0x10	8	W	0xFF	0xFF
CLMA0-3	—	—	—	PROTSR	CLMA Protection Status Register	0x14	8	R	0x00	0xFF
MPU0	8	0x10	0-7	STADD%s	Master MPU Safety Region Start Address Register %s	0x00	32	R/W	0x00000003	0xFFFFFFFF
MPU0	8	0x10	0-7	ENDADD%s	Master MPU Safety Region End Address Register %s	0x04	32	R/W	0xFFFFFC00	0xFFFFFFFF
MPU0	—	—	—	ERRINF_R	Master MPU Error Information Register for AXI type	0x80	32	R/W	0x00000000	0xFFFFFFFF
MPU0	—	—	—	ERRINF_W	Master MPU Error Information Register for AXI type	0x84	32	R/W	0x00000000	0xFFFFFFFF
MPU1-2,6	8	0x10	0-7	STADD%s	Master MPU Safety Region Start Address Register %s	0x00	32	R/W	0x00000000	0xFFFFFFFC
MPU1-2,6	8	0x10	0-7	ENDADD%s	Master MPU Safety Region End Address Register %s	0x04	32	R/W	0x00000000	0x000003FF
MPU1-2,6	—	—	—	ERRINF_R	Master MPU Error Information Register for AXI type	0x80	32	R/W	0x00000000	0xFFFFFFFF
MPU1-2,6	—	—	—	ERRINF_W	Master MPU Error Information Register for AXI type	0x84	32	R/W	0x00000000	0xFFFFFFFF
MPU3-4	8	0x10	0-7	STADD%s	Master MPU Safety Region Start Address Register %s	0x00	32	R/W	0x00000000	0xFFFFFFFC
MPU3-4	8	0x10	0-7	ENDADD%s	Master MPU Safety Region End Address Register %s	0x04	32	R/W	0x00000000	0x000003FF
MPU3-4	—	—	—	ERRINF	Master MPU Error Information Register for AHB type	0x80	32	R/W	0x00000000	0xFFFFFFFF
SYSRAM_CTL	2	0x010	0-1	SYSRAM_CTRL%s	System SRAM Control Register %s	0x000	32	R/W	0x013F0001	0xFFFFFFFF
TCMAW	—	—	—	WAIT	ATCM Wait State Register	0x10	32	R	0x00000000	0xFFFFFFFE
SHOSTIF_CFG	—	—	—	SHCFG	SHOSTIF Configuration Register	0x000	32	R/W	0x00000000	0xFFFFFFFF
RWP_S	—	—	—	PRCRS	Safety Area Protect Register	0x00	32	R/W	0x00000000	0xFFFFFFFF
MPU7	8	0x10	0-7	STADD%s	Master MPU Safety Region Start Address Register %s	0x00	32	R/W	0x00000003	0xFFFFFFFF
MPU7	8	0x10	0-7	ENDADD%s	Master MPU Safety Region End Address Register %s	0x04	32	R/W	0xFFFFFC00	0xFFFFFFFF
MPU7	—	—	—	ERRINF	Master MPU Error Information Register for AHB type	0x80	32	R/W	0x00000000	0xFFFFFFFF
MTU	—	—	—	TOERA	Timer Output Master Enable Register A	0x20A	8	R/W	0xC0	0xFF
MTU	—	—	—	TGCRA	Timer Gate Control Register A	0x20D	8	R/W	0x80	0xFF
MTU	—	—	—	TOCR1A	Timer Output Control Register 1A	0x20E	8	R/W	0x00	0xFF
MTU	—	—	—	TOCR2A	Timer Output Control Register 2A	0x20F	8	R/W	0x00	0xFF
MTU	—	—	—	TCDRA	Timer Cycle Data Register A	0x214	16	R/W	0xFFFF	0xFFFF
MTU	—	—	—	TDDRA	Timer Dead Time Data Register A	0x216	16	R/W	0xFFFF	0xFFFF
MTU	—	—	—	TCNTSA	Timer Subcounter A	0x220	16	R	0x0000	0xFFFF
MTU	—	—	—	TCBRA	Timer Cycle Buffer Register A	0x222	16	R/W	0xFFFF	0xFFFF
MTU	—	—	—	TITCR1A	Timer Interrupt Skipping Set Register 1A	0x230	8	R/W	0x00	0xFF
MTU	—	—	—	TITCNT1A	Timer Interrupt Skipping Counter 1A	0x231	8	R	0x00	0xFF
MTU	—	—	—	TBTERA	Timer Buffer Transfer Set Register A	0x232	8	R/W	0x00	0xFF
MTU	—	—	—	TDERA	Timer Dead Time Enable Register A	0x234	8	R/W	0x01	0xFF
MTU	—	—	—	TOLBRA	Timer Output Level Buffer Register A	0x236	8	R/W	0x00	0xFF
MTU	—	—	—	TITMRA	Timer Interrupt Skipping Mode Register A	0x23A	8	R/W	0x00	0xFF
MTU	—	—	—	TITCR2A	Timer Interrupt Skipping Set Register 2A	0x23B	8	R/W	0x00	0xFF
MTU	—	—	—	TITCNT2A	Timer Interrupt Skipping Counter 2A	0x23C	8	R	0x00	0xFF
MTU	—	—	—	TWCRA	Timer Waveform Control Register A	0x260	8	R/W	0x00	0xFF

Table 1.2 Register description (31 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
MTU	—	—	—	TMDR2A	Timer Mode Register 2A	0x270	8	R/W	0x00	0xFF
MTU	—	—	—	TSTRA	Timer Start Register A	0x280	8	R/W	0x00	0xFF
MTU	—	—	—	TSYRA	Timer Synchronous Register A	0x281	8	R/W	0x00	0xFF
MTU	—	—	—	TCSYSTR	Timer Counter Synchronous Start Register	0x282	8	R/W	0x00	0xFF
MTU	—	—	—	TRWERA	Timer Read/Write Enable Register	0x284	8	R/W	0x01	0xFF
MTU	—	—	—	TOERB	Timer Output Master Enable Register B	0xA0A	8	R/W	0xC0	0xFF
MTU	—	—	—	TOCR1B	Timer Output Control Register 1B	0xA0E	8	R/W	0x00	0xFF
MTU	—	—	—	TOCR2B	Timer Output Control Register 2B	0xA0F	8	R/W	0x00	0xFF
MTU	—	—	—	TCDRB	Timer Cycle Data Register B	0xA14	16	R/W	0xFFFF	0xFFFF
MTU	—	—	—	TDDRB	Timer Dead Time Data Register B	0xA16	16	R/W	0xFFFF	0xFFFF
MTU	—	—	—	TCNTSB	Timer Subcounter B	0xA20	16	R	0x0000	0xFFFF
MTU	—	—	—	TCBRB	Timer Cycle Buffer Register B	0xA22	16	R/W	0xFFFF	0xFFFF
MTU	—	—	—	TITCR1B	Timer Interrupt Skipping Set Register 1B	0xA30	8	R/W	0x00	0xFF
MTU	—	—	—	TITCNT1B	Timer Interrupt Skipping Counter 1B	0xA31	8	R	0x00	0xFF
MTU	—	—	—	TBTERB	Timer Buffer Transfer Set Register B	0xA32	8	R/W	0x00	0xFF
MTU	—	—	—	TDERB	Timer Dead Time Enable Register B	0xA34	8	R/W	0x01	0xFF
MTU	—	—	—	TOLBRB	Timer Output Level Buffer Register B	0xA36	8	R/W	0x00	0xFF
MTU	—	—	—	TITMRB	Timer Interrupt Skipping Mode Register B	0xA3A	8	R/W	0x00	0xFF
MTU	—	—	—	TITCR2B	Timer Interrupt Skipping Set Register 2B	0xA3B	8	R/W	0x00	0xFF
MTU	—	—	—	TITCNT2B	Timer Interrupt Skipping Counter 2B	0xA3C	8	R	0x00	0xFF
MTU	—	—	—	TWCRB	Timer Waveform Control Register B	0xA60	8	R/W	0x00	0xFF
MTU	—	—	—	TMDR2B	Timer Mode Register 2B	0xA70	8	R/W	0x00	0xFF
MTU	—	—	—	TSTRB	Timer Start Register B	0xA80	8	R/W	0x00	0xFF
MTU	—	—	—	TSYRB	Timer Synchronous Register B	0xA81	8	R/W	0x00	0xFF
MTU	—	—	—	TRWERB	Timer Read/Write Enable Register	0xA84	8	R/W	0x01	0xFF
MTU3	—	—	—	TCR	Timer Control Register	0x100	8	R/W	0x00	0xFF
MTU3	—	—	—	TMDR1	Timer Mode Register 1	0x102	8	R/W	0x00	0xFF
MTU3	—	—	—	TIORH	Timer I/O Control Register H	0x104	8	R/W	0x00	0xFF
MTU3	—	—	—	TIORL	Timer I/O Control Register L	0x105	8	R/W	0x00	0xFF
MTU3	—	—	—	TIER	Timer Interrupt Enable Register	0x108	8	R/W	0x00	0xFF
MTU3	—	—	—	TCNT	Timer Counter	0x110	16	R/W	0x0000	0xFFFF
MTU3	—	—	—	TGRA	Timer General Register A	0x118	16	R/W	0xFFFF	0xFFFF
MTU3	—	—	—	TGRB	Timer General Register B	0x11A	16	R/W	0xFFFF	0xFFFF
MTU3	—	—	—	TGRC	Timer General Register C	0x124	16	R/W	0xFFFF	0xFFFF
MTU3	—	—	—	TGRD	Timer General Register D	0x126	16	R/W	0xFFFF	0xFFFF
MTU3	—	—	—	TSR	Timer Status Register	0x12C	8	R/W	0xC0	0xFF
MTU3	—	—	—	TBTM	Timer Buffer Operation Transfer Mode Register	0x138	8	R/W	0x00	0xFF
MTU3	—	—	—	TCR2	Timer Control Register 2	0x14C	8	R/W	0x00	0xFF
MTU3	—	—	—	TGRE	Timer General Register E	0x172	16	R/W	0xFFFF	0xFFFF
MTU4	—	—	—	TCR	Timer Control Register	0x001	8	R/W	0x00	0xFF
MTU4	—	—	—	TMDR1	Timer Mode Register 1	0x003	8	R/W	0x00	0xFF
MTU4	—	—	—	TIORH	Timer I/O Control Register H	0x006	8	R/W	0x00	0xFF
MTU4	—	—	—	TIORL	Timer I/O Control Register L	0x007	8	R/W	0x00	0xFF
MTU4	—	—	—	TIER	Timer Interrupt Enable Register	0x009	8	R/W	0x00	0xFF
MTU4	—	—	—	TCNT	Timer Counter	0x012	16	R/W	0x0000	0xFFFF

Table 1.2 Register description (32 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
MTU4	—	—	—	TGRA	Timer General Register A	0x01C	16	R/W	0xFFFF	0xFFFF
MTU4	—	—	—	TGRB	Timer General Register B	0x01E	16	R/W	0xFFFF	0xFFFF
MTU4	—	—	—	TGRC	Timer General Register C	0x028	16	R/W	0xFFFF	0xFFFF
MTU4	—	—	—	TGRD	Timer General Register D	0x02A	16	R/W	0xFFFF	0xFFFF
MTU4	—	—	—	TSR	Timer Status Register	0x02D	8	R/W	0xC0	0xFF
MTU4	—	—	—	TBTM	Timer Buffer Operation Transfer Mode Register	0x039	8	R/W	0x00	0xFF
MTU4	—	—	—	TADCR	Timer A/D Converter Start Request Control Register	0x040	16	R/W	0x0000	0xFFFF
MTU4	—	—	—	TADCORA	Timer A/D Converter Start Request Cycle Set Register A	0x044	16	R/W	0xFFFF	0xFFFF
MTU4	—	—	—	TADCORB	Timer A/D Converter Start Request Cycle Set Register B	0x046	16	R/W	0xFFFF	0xFFFF
MTU4	—	—	—	TADCOBRA	Timer A/D Converter Start Request Cycle Set Buffer Register A	0x048	16	R/W	0xFFFF	0xFFFF
MTU4	—	—	—	TADCOBRB	Timer A/D Converter Start Request Cycle Set Buffer Register B	0x04A	16	R/W	0xFFFF	0xFFFF
MTU4	—	—	—	TCR2	Timer Control Register 2	0x4D	8	R/W	0x00	0xFF
MTU4	—	—	—	TGRE	Timer General Register E	0x074	16	R/W	0xFFFF	0xFFFF
MTU4	—	—	—	TGRF	Timer General Register F	0x076	16	R/W	0xFFFF	0xFFFF
MTU_NF	—	—	—	NFCR0	Noise Filter Control Register 0	0x000	8	R/W	0x00	0xFF
MTU_NF	—	—	—	NFCR1	Noise Filter Control Register 1	0x001	8	R/W	0x00	0xFF
MTU_NF	—	—	—	NFCR2	Noise Filter Control Register 2	0x002	8	R/W	0x00	0xFF
MTU_NF	—	—	—	NFCR3	Noise Filter Control Register 3	0x003	8	R/W	0x00	0xFF
MTU_NF	—	—	—	NFCR4	Noise Filter Control Register 4	0x004	8	R/W	0x00	0xFF
MTU_NF	—	—	—	NFCR8	Noise Filter Control Register 8	0x008	8	R/W	0x00	0xFF
MTU_NF	—	—	—	NFCRC	Noise Filter Control Register C	0x009	8	R/W	0x00	0xFF
MTU_NF	—	—	—	NFCR6	Noise Filter Control Register 6	0x803	8	R/W	0x00	0xFF
MTU_NF	—	—	—	NFCR7	Noise Filter Control Register 7	0x804	8	R/W	0x00	0xFF
MTU_NF	—	—	—	NFCR5	Noise Filter Control Register 5	0x805	8	R/W	0x00	0xFF
MTU0	—	—	—	TCR	Timer Control Register	0x000	8	R/W	0x00	0xFF
MTU0	—	—	—	TMDR1	Timer Mode Register 1	0x001	8	R/W	0x00	0xFF
MTU0	—	—	—	TIORH	Timer I/O Control Register H	0x002	8	R/W	0x00	0xFF
MTU0	—	—	—	TIORL	Timer I/O Control Register L	0x003	8	R/W	0x00	0xFF
MTU0	—	—	—	TIER	Timer Interrupt Enable Register	0x004	8	R/W	0x00	0xFF
MTU0	—	—	—	TCNT	Timer Counter	0x006	16	R/W	0x0000	0xFFFF
MTU0	—	—	—	TGRA	Timer General Register A	0x008	16	R/W	0xFFFF	0xFFFF
MTU0	—	—	—	TGRB	Timer General Register B	0x00A	16	R/W	0xFFFF	0xFFFF
MTU0	—	—	—	TGRC	Timer General Register C	0x00C	16	R/W	0xFFFF	0xFFFF
MTU0	—	—	—	TGRD	Timer General Register D	0x00E	16	R/W	0xFFFF	0xFFFF
MTU0	—	—	—	TGRE	Timer General Register E	0x020	16	R/W	0xFFFF	0xFFFF
MTU0	—	—	—	TGRF	Timer General Register F	0x022	16	R/W	0xFFFF	0xFFFF
MTU0	—	—	—	TIER2	Timer Interrupt Enable Register 2	0x024	8	R/W	0x00	0xFF
MTU0	—	—	—	TBTM	Timer Buffer Operation Transfer Mode Register	0x026	8	R/W	0x00	0xFF
MTU0	—	—	—	TCR2	Timer Control Register 2	0x28	8	R/W	0x00	0xFF
MTU1	—	—	—	TCR	Timer Control Register	0x000	8	R/W	0x00	0xFF
MTU1	—	—	—	TMDR1	Timer Mode Register 1	0x001	8	R/W	0x00	0xFF
MTU1	—	—	—	TIOR	Timer I/O Control Register	0x002	8	R/W	0x00	0xFF
MTU1	—	—	—	TIER	Timer Interrupt Enable Register	0x004	8	R/W	0x00	0xFF

Table 1.2 Register description (33 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
MTU1	—	—	—	TSR	Timer Status Register	0x005	8	R/W	0xC0	0xFF
MTU1	—	—	—	TCNT	Timer Counter	0x006	16	R/W	0x0000	0xFFFF
MTU1	—	—	—	TGRA	Timer General Register A	0x008	16	R/W	0xFFFF	0xFFFF
MTU1	—	—	—	TGRB	Timer General Register B	0x00A	16	R/W	0xFFFF	0xFFFF
MTU1	—	—	—	TICCR	Timer Input Capture Control Register	0x010	8	R/W	0x00	0xFF
MTU1	—	—	—	TMDR3	Timer Mode Register 3	0x011	8	R/W	0x02	0xFF
MTU1	—	—	—	TCR2	Timer Control Register 2	0x14	8	R/W	0x00	0xFF
MTU1	—	—	—	TCNTLW	Timer Longword Counter	0x020	32	R/W	0x00000000	0xFFFFFFFF
MTU1	—	—	—	TGRALW	Timer Longword General Register A	0x024	32	R/W	0xFFFFFFFF	0xFFFFFFFF
MTU1	—	—	—	TGRBLW	Timer Longword General Register B	0x028	32	R/W	0xFFFFFFFF	0xFFFFFFFF
MTU2	—	—	—	TCR	Timer Control Register	0x000	8	R/W	0x00	0xFF
MTU2	—	—	—	TMDR1	Timer Mode Register 1	0x001	8	R/W	0x00	0xFF
MTU2	—	—	—	TIOR	Timer I/O Control Register	0x002	8	R/W	0x00	0xFF
MTU2	—	—	—	TIER	Timer Interrupt Enable Register	0x004	8	R/W	0x00	0xFF
MTU2	—	—	—	TSR	Timer Status Register	0x005	8	R/W	0xC0	0xFF
MTU2	—	—	—	TCNT	Timer Counter	0x006	16	R/W	0x0000	0xFFFF
MTU2	—	—	—	TGRA	Timer General Register A	0x008	16	R/W	0xFFFF	0xFFFF
MTU2	—	—	—	TGRB	Timer General Register B	0x00A	16	R/W	0xFFFF	0xFFFF
MTU2	—	—	—	TCR2	Timer Control Register 2	0x0C	8	R/W	0x00	0xFF
MTU8	—	—	—	TCR	Timer Control Register	0x000	8	R/W	0x00	0xFF
MTU8	—	—	—	TMDR1	Timer Mode Register 1	0x001	8	R/W	0x00	0xFF
MTU8	—	—	—	TIORH	Timer I/O Control Register H	0x002	8	R/W	0x00	0xFF
MTU8	—	—	—	TIORL	Timer I/O Control Register L	0x003	8	R/W	0x00	0xFF
MTU8	—	—	—	TIER	Timer Interrupt Enable Register	0x004	8	R/W	0x00	0xFF
MTU8	—	—	—	TCR2	Timer Control Register 2	0x06	8	R/W	0x00	0xFF
MTU8	—	—	—	TCNT	Timer Counter	0x008	32	R/W	0x00000000	0xFFFFFFFF
MTU8	—	—	—	TGRA	Timer General Register A	0x00C	32	R/W	0xFFFFFFFF	0xFFFFFFFF
MTU8	—	—	—	TGRB	Timer General Register B	0x010	32	R/W	0xFFFFFFFF	0xFFFFFFFF
MTU8	—	—	—	TGRC	Timer General Register C	0x014	32	R/W	0xFFFFFFFF	0xFFFFFFFF
MTU8	—	—	—	TGRD	Timer General Register D	0x018	32	R/W	0xFFFFFFFF	0xFFFFFFFF
MTU6	—	—	—	TCR	Timer Control Register	0x100	8	R/W	0x00	0xFF
MTU6	—	—	—	TMDR1	Timer Mode Register 1	0x102	8	R/W	0x00	0xFF
MTU6	—	—	—	TIORH	Timer I/O Control Register H	0x104	8	R/W	0x00	0xFF
MTU6	—	—	—	TIORL	Timer I/O Control Register L	0x105	8	R/W	0x00	0xFF
MTU6	—	—	—	TIER	Timer Interrupt Enable Register	0x108	8	R/W	0x00	0xFF
MTU6	—	—	—	TCNT	Timer Counter	0x110	16	R/W	0x0000	0xFFFF
MTU6	—	—	—	TGRA	Timer General Register A	0x118	16	R/W	0xFFFF	0xFFFF
MTU6	—	—	—	TGRB	Timer General Register B	0x11A	16	R/W	0xFFFF	0xFFFF
MTU6	—	—	—	TGRC	Timer General Register C	0x124	16	R/W	0xFFFF	0xFFFF
MTU6	—	—	—	TGRD	Timer General Register D	0x126	16	R/W	0xFFFF	0xFFFF
MTU6	—	—	—	TSR	Timer Status Register	0x12C	8	R/W	0xC0	0xFF
MTU6	—	—	—	TBTM	Timer Buffer Operation Transfer Mode Register	0x138	8	R/W	0x00	0xFF
MTU6	—	—	—	TCR2	Timer Control Register 2	0x14C	8	R/W	0x00	0xFF
MTU6	—	—	—	TSYCR	Timer Synchronous Clear Register	0x150	8	R/W	0x00	0xFF
MTU6	—	—	—	TGRE	Timer General Register E	0x172	16	R/W	0xFFFF	0xFFFF
MTU7	—	—	—	TCR	Timer Control Register	0x001	8	R/W	0x00	0xFF

Table 1.2 Register description (34 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
MTU7	—	—	—	TMDR1	Timer Mode Register 1	0x003	8	R/W	0x00	0xFF
MTU7	—	—	—	TIORH	Timer I/O Control Register H	0x006	8	R/W	0x00	0xFF
MTU7	—	—	—	TIORL	Timer I/O Control Register L	0x007	8	R/W	0x00	0xFF
MTU7	—	—	—	TIER	Timer Interrupt Enable Register	0x009	8	R/W	0x00	0xFF
MTU7	—	—	—	TCNT	Timer Counter	0x012	16	R/W	0x0000	0xFFFF
MTU7	—	—	—	TGRA	Timer General Register A	0x01C	16	R/W	0xFFFF	0xFFFF
MTU7	—	—	—	TGRB	Timer General Register B	0x01E	16	R/W	0xFFFF	0xFFFF
MTU7	—	—	—	TGRC	Timer General Register C	0x028	16	R/W	0xFFFF	0xFFFF
MTU7	—	—	—	TGRD	Timer General Register D	0x02A	16	R/W	0xFFFF	0xFFFF
MTU7	—	—	—	TSR	Timer Status Register	0x02D	8	R/W	0xC0	0xFF
MTU7	—	—	—	TBTM	Timer Buffer Operation Transfer Mode Register	0x039	8	R/W	0x00	0xFF
MTU7	—	—	—	TADCR	Timer A/D Converter Start Request Control Register	0x040	16	R/W	0x0000	0xFFFF
MTU7	—	—	—	TADCORA	Timer A/D Converter Start Request Cycle Set Register A	0x044	16	R/W	0xFFFF	0xFFFF
MTU7	—	—	—	TADCORB	Timer A/D Converter Start Request Cycle Set Register B	0x046	16	R/W	0xFFFF	0xFFFF
MTU7	—	—	—	TADCOBRA	Timer A/D Converter Start Request Cycle Set Buffer Register A	0x048	16	R/W	0xFFFF	0xFFFF
MTU7	—	—	—	TADCOBRB	Timer A/D Converter Start Request Cycle Set Buffer Register B	0x04A	16	R/W	0xFFFF	0xFFFF
MTU7	—	—	—	TCR2	Timer Control Register 2	0x4D	8	R/W	0x00	0xFF
MTU7	—	—	—	TGRE	Timer General Register E	0x074	16	R/W	0xFFFF	0xFFFF
MTU7	—	—	—	TGRF	Timer General Register F	0x076	16	R/W	0xFFFF	0xFFFF
MTU5	—	—	—	TCNTU	Timer Counter U	0x080	16	R/W	0x0000	0xFFFF
MTU5	—	—	—	TGRU	Timer General Register U	0x082	16	R/W	0xFFFF	0xFFFF
MTU5	—	—	—	TCRU	Timer Control Register U	0x084	8	R/W	0x00	0xFF
MTU5	—	—	—	TCR2U	Timer Control Register 2U	0x85	8	R/W	0x00	0xFF
MTU5	—	—	—	TIORU	Timer I/O Control Register U	0x086	8	R/W	0x00	0xFF
MTU5	—	—	—	TCNTV	Timer Counter V	0x090	16	R/W	0x0000	0xFFFF
MTU5	—	—	—	TGRV	Timer General Register V	0x092	16	R/W	0xFFFF	0xFFFF
MTU5	—	—	—	TCRV	Timer Control Register V	0x094	8	R/W	0x00	0xFF
MTU5	—	—	—	TCR2V	Timer Control Register 2V	0x95	8	R/W	0x00	0xFF
MTU5	—	—	—	TIORV	Timer I/O Control Register V	0x096	8	R/W	0x00	0xFF
MTU5	—	—	—	TCNTW	Timer Counter W	0x0A0	16	R/W	0x0000	0xFFFF
MTU5	—	—	—	TGRW	Timer General Register W	0x0A2	16	R/W	0xFFFF	0xFFFF
MTU5	—	—	—	TCRW	Timer Control Register W	0x0A4	8	R/W	0x00	0xFF
MTU5	—	—	—	TCR2W	Timer Control Register 2W	0xA5	8	R/W	0x00	0xFF
MTU5	—	—	—	TIORW	Timer I/O Control Register W	0x0A6	8	R/W	0x00	0xFF
MTU5	—	—	—	TIER	Timer Interrupt Enable Register	0x0B2	8	R/W	0x01	0xFF
MTU5	—	—	—	TSTR	Timer Start Register	0x0B4	8	R/W	0x00	0xFF
MTU5	—	—	—	TCNTCMPCLR	Timer Compare Match Clear Register	0x0B6	8	R/W	0x00	0xFF
GPT0-6	—	—	—	GTWP	General PWM Timer Write-Protection Register	0x00	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTSTR	General PWM Timer Software Start Register	0x04	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTSTP	General PWM Timer Software Stop Register	0x08	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT0-6	—	—	—	GTCLR	General PWM Timer Software Clear Register	0x0C	32	W	0x00000000	0xFFFFFFFF

Table 1.2 Register description (35 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
GPT0-6	—	—	—	GTSSR	General PWM Timer Start Source Select Register	0x10	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTPSR	General PWM Timer Stop Source Select Register	0x14	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTCSR	General PWM Timer Clear Source Select Register	0x18	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTUPSR	General PWM Timer Count-Up Source Select Register	0x1C	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTDNSR	General PWM Timer Count-Down Source Select Register	0x20	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTICASR	General PWM Timer Input Capture Source Select Register A	0x24	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTICBSR	General PWM Timer Input Capture Source Select Register B	0x28	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTCR	General PWM Timer Control Register	0x2C	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTUDDTYC	General PWM Timer Count Direction and Duty Setting Register	0x30	32	R/W	0x00000001	0xFFFFFFFF
GPT0-6	—	—	—	GTIOR	General PWM Timer I/O Control Register	0x34	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTINTAD	General PWM Timer Interrupt Output Setting Register	0x38	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTST	General PWM Timer Status Register	0x3C	32	R/W	0x00008000	0xFFFFFFFF
GPT0-6	—	—	—	GTBER	General PWM Timer Buffer Enable Register	0x40	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTITC	General PWM Timer Interrupt and A/D Converter Start Request Skipping Setting Register	0x44	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTCNT	General PWM Timer Counter	0x48	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTCCRA	General PWM Timer Compare Capture Register A (m = A to F)	0x4C	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT0-6	—	—	—	GTCCRB	General PWM Timer Compare Capture Register B (m = A to F)	0x50	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT0-6	—	—	—	GTCCRC	General PWM Timer Compare Capture Register C (m = A to F)	0x54	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT0-6	—	—	—	GTCCRE	General PWM Timer Compare Capture Register E (m = A to F)	0x58	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT0-6	—	—	—	GTCCRD	General PWM Timer Compare Capture Register D (m = A to F)	0x5C	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT0-6	—	—	—	GTCCRF	General PWM Timer Compare Capture Register F (m = A to F)	0x60	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT0-6	—	—	—	GTPR	General PWM Timer Cycle Setting Register	0x64	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT0-6	—	—	—	GTPBR	General PWM Timer Cycle Setting Buffer Register	0x68	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT0-6	—	—	—	GTPDBR	General PWM Timer Cycle Setting Double-Buffer Register	0x6C	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT0-6	—	—	—	GTADTRA	A/D Converter Start Request Timing Register A (m = A, B)	0x70	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT0-6	—	—	—	GTADTBRA	A/D Converter Start Request Timing Buffer Register A (m = A, B)	0x74	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT0-6	—	—	—	GTADTDBRA	A/D Converter Start Request Timing Double-Buffer Register A	0x78	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT0-6	—	—	—	GTADTRB	A/D Converter Start Request Timing Register B (m = A, B)	0x7C	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT0-6	—	—	—	GTADTBRB	A/D Converter Start Request Timing Buffer Register B (m = A, B)	0x80	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT0-6	—	—	—	GTADTDBRB	A/D Converter Start Request Timing Double-Buffer Register B	0x84	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT0-6	—	—	—	GTDTCR	General PWM Timer Dead Time Control Register	0x88	32	R/W	0x00000000	0xFFFFFFFF

Table 1.2 Register description (36 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
GPT0-6	—	—	—	GTDVU	General PWM Timer Dead Time Value Register U (m = U, D)	0x8C	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT0-6	—	—	—	GTDVD	General PWM Timer Dead Time Value Register D (m = U, D)	0x90	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT0-6	—	—	—	GTDBU	General PWM Timer Dead Time Value Buffer Register U (m = U, D)	0x94	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT0-6	—	—	—	GTDBD	General PWM Timer Dead Time Value Buffer Register D (m = U, D)	0x98	32	R/W	0xFFFFFFFF	0xFFFFFFFF
GPT0-6	—	—	—	GTSOS	General PWM Timer Output Protection Function Status Register	0x9C	32	R	0x00000000	0xFFFFFCFF
GPT0-6	—	—	—	GTSOTR	General PWM Timer Output Protection Function Temporary Release Register	0xA0	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTADSMR	General PWM Timer A/D Conversion Start Request Signal Monitoring Register	0xA4	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTEITC	General PWM Timer Extended Interrupt Skipping Counter Control Register	0xA8	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTEITLI1	General PWM Timer Extended Interrupt Skipping Setting Register 1	0xAC	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTEITLI2	General PWM Timer Extended Interrupt Skipping Setting Register 2	0xB0	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTEITLB	General PWM Timer Extended Buffer Transfer Skipping Setting Register	0xB4	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTSECSR	General PWM Timer Operation Enable Bit Simultaneous Control Channel Select Register	0xD0	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTSECR	General PWM Timer Operation Enable Bit Simultaneous Control Register	0xD4	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTSWSR	General PWM Timer Switch Source Select Register	0xD8	32	R/W	0x00000000	0xFFFFFFFF
GPT0-6	—	—	—	GTSWOS	General PWM Timer Switch Offset Setting Register	0xDC	32	R/W	0x00000000	0xFFFFFFFF
TFU	—	—	—	FXSCIOC	Fixed-point Sine Cosine Input/Output Setting Register	0x04	8	R/W	0x00	0xFF
TFU	—	—	—	FXATIOC	Fixed-point Arctangent hypot_k Input/ Output Setting Register	0x05	8	R/W	0x00	0xFF
TFU	—	—	—	TRGSTS	Trigonometric Status Register	0x08	8	R/W	0x00	0xFF
TFU	—	—	—	FPSCDT0	Floating-point Sine Cosine Data Register 0	0x10	32	R/W	0x00000000	0xFFFFFFFF
TFU	—	—	—	FPSCDT1	Floating-point Sine Cosine Data Register 1	0x14	32	R/W	0x00000000	0xFFFFFFFF
TFU	—	—	—	FPATDT0	Floating-point Arctangent hypot_k Data Register 0	0x18	32	R/W	0x00000000	0xFFFFFFFF
TFU	—	—	—	FPATDT1	Floating-point Arctangent hypot_k Data Register 1	0x1C	32	R/W	0x00000000	0xFFFFFFFF
TFU	—	—	—	FXSCDT0	Fixed-point Sine Cosine Data Register	0x20	32	R/W	0x00000000	0xFFFFFFFF
TFU	—	—	—	FXSCDT1	Fixed-point Sine Cosine Data Register 1	0x24	32	R/W	0x00000000	0xFFFFFFFF
TFU	—	—	—	FXATDT0	Fixed-point Arctangent hypot_k Data Register 0	0x28	32	R/W	0x00000000	0xFFFFFFFF
TFU	—	—	—	FXATDT1	Fixed-point Arctangent hypot_k Data Register 1	0x2C	32	R/W	0x00000000	0xFFFFFFFF
TFU	—	—	—	DTSR0	Data Save Restore Register 0	0x30	32	R/W	0x00000000	0xFFFFFFFF
TFU	—	—	—	DTSR1	Data Save Restore Register 1	0x34	32	R/W	0x00000000	0xFFFFFFFF
ADC120	—	—	—	ADCSR	A/D Control Register	0x00	16	R/W	0x0000	0xFFFF
ADC120	—	—	—	ADANSA0	A/D Channel Select Register A0	0x04	16	R/W	0x0000	0xFFFF
ADC120	—	—	—	ADADS0	A/D-Converted Value Addition/Average Function Channel Select Register 0	0x08	16	R/W	0x0000	0xFFFF
ADC120	—	—	—	ADADC	A/D-Converted Value Addition/Average Count Select Register	0x0C	8	R/W	0x00	0xFF

Table 1.2 Register description (37 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
ADC120	—	—	—	ADCER	A/D Control Extended Register	0x0E	16	R/W	0x0000	0xFFFF
ADC120	—	—	—	ADSTRGR	A/D Conversion Start Trigger Select Register	0x10	16	R/W	0x0000	0xFFFF
ADC120	—	—	—	ADANSB0	A/D Channel Select Register B0	0x14	16	R/W	0x0000	0xFFFF
ADC120	—	—	—	ADDBLDR	A/D Data Duplication Register	0x18	16	R	0x0000	0xFFFF
ADC120	4	0x002	0-3	ADDR%s	A/D Data Register %s	0x020	16	R	0x0000	0xFFFF
ADC120	—	—	—	ADSHCR	A/D Sample and Hold Control Register	0x66	16	R/W	0x0018	0xFFFF
ADC120	—	—	—	ADELCCR	A/D Event Link Control Register	0x7D	8	R/W	0x00	0xFF
ADC120	—	—	—	ADGSPCR	A/D Group Scan Priority Control Register	0x80	16	R/W	0x0000	0xFFFF
ADC120	—	—	—	ADDBLDRA	A/D Data Duplication Register A	0x84	16	R	0x0000	0xFFFF
ADC120	—	—	—	ADDBLDRB	A/D Data Duplication Register B	0x86	16	R	0x0000	0xFFFF
ADC120	—	—	—	ADWINMON	A/D Compare Function Window A/B Status Monitoring Register	0x8C	8	R	0x00	0xFF
ADC120	—	—	—	ADCMPCR	A/D Compare Function Control Register	0x90	16	R/W	0x0000	0xFFFF
ADC120	—	—	—	ADCMPANSR0	A/D Compare Function Window A Channel Select Register 0	0x94	16	R/W	0x0000	0xFFFF
ADC120	—	—	—	ADCMPLR0	A/D Compare Function Window A Comparison Condition Setting Register 0	0x98	16	R/W	0x0000	0xFFFF
ADC120	—	—	—	ADCMPDR0	A/D Comparison Function Window A Lower Level Setting Register	0x9C	16	R/W	0x0000	0xFFFF
ADC120	—	—	—	ADCMPDR1	AD Comparison Function Window A Upper Level Setting Register	0x9E	16	R/W	0x0000	0xFFFF
ADC120	—	—	—	ADCMPSR0	A/D Comparison Function Window A Channel Status Register 0	0xA0	16	R/W	0x0000	0xFFFF
ADC120	—	—	—	ADCMPBNSR	A/D Compare Function Window B Channel Select Register	0xA6	8	R/W	0x00	0xFF
ADC120	—	—	—	ADWINLLB	A/D Compare Function Window B Lower-Side Level Setting Register	0xA8	16	R/W	0x0000	0xFFFF
ADC120	—	—	—	ADWINULB	A/D Compare Function Window B Upper-Side Level Setting Register	0xAA	16	R/W	0x0000	0xFFFF
ADC120	—	—	—	ADCMPBSR	A/D Compare Function Window B Status Register	0xAC	8	R/W	0x00	0xFF
ADC120	—	—	—	ADANSC0	A/D Channel Select Register C0	0xD4	16	R/W	0x0000	0xFFFF
ADC120	—	—	—	ADGCTRGR	A/D Group C Trigger Select Register	0xD9	8	R/W	0x00	0xFF
ADC120	4	0x01	0-3	ADSSTR%s	A/D Sampling State Register %s	0xE0	8	R/W	0x0B	0xFF
ADC121	—	—	—	ADCSR	A/D Control Register	0x00	16	R/W	0x0000	0xFFFF
ADC121	—	—	—	ADANSA0	A/D Channel Select Register A0	0x04	16	R/W	0x0000	0xFFFF
ADC121	—	—	—	ADADS0	A/D-Converted Value Addition/Average Function Channel Select Register 0	0x08	16	R/W	0x0000	0xFFFF
ADC121	—	—	—	ADADC	A/D-Converted Value Addition/Average Count Select Register	0x0C	8	R/W	0x00	0xFF
ADC121	—	—	—	ADCER	A/D Control Extended Register	0x0E	16	R/W	0x0000	0xFFFF
ADC121	—	—	—	ADSTRGR	A/D Conversion Start Trigger Select Register	0x10	16	R/W	0x0000	0xFFFF
ADC121	—	—	—	ADANSB0	A/D Channel Select Register B0	0x14	16	R/W	0x0000	0xFFFF
ADC121	—	—	—	ADDBLDR	A/D Data Duplication Register	0x18	16	R	0x0000	0xFFFF
ADC121	4	0x002	0-3	ADDR%s	A/D Data Register %s	0x020	16	R	0x0000	0xFFFF
ADC121	—	—	—	ADSHCR	A/D Sample and Hold Control Register	0x66	16	R/W	0x0018	0xFFFF
ADC121	—	—	—	ADELCCR	A/D Event Link Control Register	0x7D	8	R/W	0x00	0xFF
ADC121	—	—	—	ADGSPCR	A/D Group Scan Priority Control Register	0x80	16	R/W	0x0000	0xFFFF
ADC121	—	—	—	ADDBLDRA	A/D Data Duplication Register A	0x84	16	R	0x0000	0xFFFF
ADC121	—	—	—	ADDBLDRB	A/D Data Duplication Register B	0x86	16	R	0x0000	0xFFFF

Table 1.2 Register description (38 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
ADC121	—	—	—	ADWINMON	A/D Compare Function Window A/B Status Monitoring Register	0x8C	8	R	0x00	0xFF
ADC121	—	—	—	ADCMPCR	A/D Compare Function Control Register	0x90	16	R/W	0x0000	0xFFFF
ADC121	—	—	—	ADCMPANSR0	A/D Compare Function Window A Channel Select Register 0	0x94	16	R/W	0x0000	0xFFFF
ADC121	—	—	—	ADCMPLR0	A/D Compare Function Window A Comparison Condition Setting Register 0	0x98	16	R/W	0x0000	0xFFFF
ADC121	—	—	—	ADCMPDR0	A/D Comparison Function Window A Lower Level Setting Register	0x9C	16	R/W	0x0000	0xFFFF
ADC121	—	—	—	ADCMPDR1	AD Comparison Function Window A Upper Level Setting Register	0x9E	16	R/W	0x0000	0xFFFF
ADC121	—	—	—	ADCMPSR0	A/D Comparison Function Window A Channel Status Register 0	0xA0	16	R/W	0x0000	0xFFFF
ADC121	—	—	—	ADCMPBNSR	A/D Compare Function Window B Channel Select Register	0xA6	8	R/W	0x00	0xFF
ADC121	—	—	—	ADWINLLB	A/D Compare Function Window B Lower-Side Level Setting Register	0xA8	16	R/W	0x0000	0xFFFF
ADC121	—	—	—	ADWINULB	A/D Compare Function Window B Upper-Side Level Setting Register	0xAA	16	R/W	0x0000	0xFFFF
ADC121	—	—	—	ADCMPBSR	A/D Compare Function Window B Status Register	0xAC	8	R/W	0x00	0xFF
ADC121	—	—	—	ADANSC0	A/D Channel Select Register C0	0xD4	16	R/W	0x0000	0xFFFF
ADC121	—	—	—	ADGCTRGR	A/D Group C Trigger Select Register	0xD9	8	R/W	0x00	0xFF
ADC121	4	0x01	0-3	ADSSTR%s	A/D Sampling State Register %s	0xE0	8	R/W	0x0B	0xFF
POE3	—	—	—	ICSR1	Input Level Control/Status Register 1	0x00	16	R/W	0x0000	0xFFFF
POE3	—	—	—	OCSR1	Output Level Control/Status Register 1	0x02	16	R/W	0x0000	0xFFFF
POE3	—	—	—	ICSR2	Input Level Control/Status Register 2	0x04	16	R/W	0x0000	0xFFFF
POE3	—	—	—	OCSR2	Output Level Control/Status Register 2	0x06	16	R/W	0x0000	0xFFFF
POE3	—	—	—	ICSR3	Input Level Control/Status Register 3	0x08	16	R/W	0x0000	0xFFFF
POE3	—	—	—	SPOER	Software Port Output Enable Register	0x0A	8	R/W	0x00	0xFF
POE3	—	—	—	POECR1	Port Output Enable Control Register 1	0x0B	8	R/W	0x00	0xFF
POE3	—	—	—	POECR2	Port Output Enable Control Register 2	0x0C	16	R/W	0x0707	0xFFFF
POE3	—	—	—	POECR4	Port Output Enable Control Register 4	0x10	16	R/W	0x0000	0xFFFF
POE3	—	—	—	POECR5	Port Output Enable Control Register 5	0x12	16	R/W	0x0000	0xFFFF
POE3	—	—	—	ICSR4	Input Level Control/Status Register 4	0x16	16	R/W	0x0000	0xFFFF
POE3	—	—	—	ICSR5	Input Level Control/Status Register 5	0x18	16	R/W	0x0000	0xFFFF
POE3	—	—	—	ALR1	Active Level Setting Register 1	0x1A	16	R/W	0x0000	0xFFFF
POE3	—	—	—	ICSR6	Input Level Control/Status Register 6	0x1C	16	R/W	0x0000	0xFFFF
POE3	—	—	—	ICSR7	Input Level Control/Status Register 7	0x1E	16	R/W	0x0000	0xFFFF
POE3	—	—	—	M0SELR1	MTU0 Pin Select Register 1	0x24	8	R/W	0x00	0xFF
POE3	—	—	—	M0SELR2	MTU0 Pin Select Register 2	0x25	8	R/W	0x00	0xFF
POE3	—	—	—	M3SELR	MTU3 Pin Select Register	0x26	8	R/W	0x00	0xFF
POE3	—	—	—	M4SELR1	MTU4 Pin Select Register 1	0x27	8	R/W	0x00	0xFF
POE3	—	—	—	M4SELR2	MTU4 Pin Select Register 2	0x28	8	R/W	0x00	0xFF
POE3	—	—	—	M6SELR	MTU6 Pin Select Register	0x2A	8	R/W	0x00	0xFF
POE3	—	—	—	M7SELR1	MTU7 Pin Select Register 1	0x2B	8	R/W	0x00	0xFF
POE3	—	—	—	M7SELR2	MTU7 Pin Select Register 2	0x2C	8	R/W	0x00	0xFF
POEG0	—	—	—	POEG0GA	POEG0 Group A Setting Register	0x000	32	R/W	0x00000000	0xFFFFFFFF
POEG0	—	—	—	POEG0GB	POEG0 Group B Setting Register	0x400	32	R/W	0x00000000	0xFFFFFFFF
POEG0	—	—	—	POEG0GC	POEG0 Group C Setting Register	0x800	32	R/W	0x00000000	0xFFFFFFFF
POEG0	—	—	—	POEG0GD	POEG0 Group D Setting Register	0xC00	32	R/W	0x00000000	0xFFFFFFFF

Table 1.2 Register description (39 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
DSMIF0-1	—	—	—	DSSEICR	Overcurrent Sum Error Detect Interrupt Control Register	0x080	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	—	—	—	DSSECSR	Overcurrent Sum Error Detect Channel Setting Register	0x084	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	—	—	—	DSSELTR	Overcurrent Sum Error Detect Low Threshold Register	0x088	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	—	—	—	DSSEHTR	Overcurrent Sum Error Detect High Threshold Register	0x08C	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	—	—	—	DSSECR	Overcurrent Sum Error Detect Control Register	0x090	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x04	0-2	DSSECDR%s	Overcurrent Sum Error Detect Capture Data Register %s	0x0A0	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	—	—	—	DSCMSR	Common Mode Setting Register	0x0C0	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	—	—	—	DSCICR	Common Interrupt Control Register	0x0C4	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	—	—	—	DSCSTRTR	Channel Software Start Trigger Register	0x200	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	—	—	—	DSCSTPTR	Channel Software Stop Trigger Register	0x204	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	—	—	—	DSCESR	Channel Error Status Register	0x220	32	R	0x00000000	0xFFFFFFFF
DSMIF0-1	—	—	—	DSCOCESR	Channel Overcurrent Error Status Register	0x224	32	R	0x00000000	0xFFFFFFFF
DSMIF0-1	—	—	—	DSCOCNSR	Channel Overcurrent Notification Status Register	0x228	32	R	0x00000000	0xFFFFFFFF
DSMIF0-1	—	—	—	DSCOCRMR	Channel Overcurrent Comparator Result Monitor Register	0x22C	32	R	0x00000000	0xFFFFFFFF
DSMIF0-1	—	—	—	DSCSR	Channel Status Register	0x240	32	R	0x00000000	0xFFFFFFFF
DSMIF0-1	—	—	—	DSCSSR	Channel State Status Register	0x244	32	R	0x00000000	0xFFFFFFFF
DSMIF0-1	—	—	—	DSCESCR	Channel Error Status Clear Register	0x260	32	W	0x00000000	0xFFFFFFFF
DSMIF0-1	—	—	—	DSCOCESCR	Channel Overcurrent Error Status Clear Register	0x264	32	W	0x00000000	0xFFFFFFFF
DSMIF0-1	—	—	—	DSCOCNSCR	Channel Overcurrent Notification Status Clear Register	0x268	32	W	0x00000000	0xFFFFFFFF
DSMIF0-1	—	—	—	DSCSCR	Channel Status Clear Register	0x280	32	W	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSICRCH%s	Interrupt Control Register Channel %s	0x300	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSCMCCRCH%s	Current Measurement Clock Control Register Channel %s	0x304	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSCMFCRCH%s	Current Measurement Filter Control Register Channel %s	0x308	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSCMCTCRCH%s	Current Measurement Capture Trigger Control Register Channel %s	0x30C	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSEDCRCH%s	Error Detect Control Register Channel %s	0x320	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSSCTSRCH%s	Short Circuit Threshold Setting Register Channel %s	0x324	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSOCFCRCH%s	Overcurrent Detect Filter Control Register Channel %s	0x330	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSODCRCH%s	Overcurrent Detect Control Register Channel %s	0x334	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSODWCRCH%s	Overcurrent Detect Window Control Register Channel %s	0x338	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSOCLTR0CH%s	Overcurrent Low Threshold Register 0 Channel %s	0x3A0	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSOCHTR0CH%s	Overcurrent High Threshold Register 0 Channel %s	0x3A4	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSOCLTR1CH%s	Overcurrent Low Threshold Register 1 Channel %s	0x3A8	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSOCHTR1CH%s	Overcurrent High Threshold Register 1 Channel %s	0x3AC	32	R/W	0x00000000	0xFFFFFFFF

Table 1.2 Register description (40 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
DSMIF0-1	3	0x140	0-2	DSOCLTR2CH%s	Overcurrent Low Threshold Register 2 Channel %s	0x3B0	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSOCHTR2CH%s	Overcurrent High Threshold Register 2 Channel %s	0x3B4	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSCSTRTRCH%s	Software Start Trigger Register Channel %s	0x3E0	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSCSTPTRCH%s	Software Stop Trigger Register Channel %s	0x3E4	32	R/W	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSCDRCH%s	Current Data Register Channel %s	0x3F0	32	R	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSCCDRACH%s	Capture Current Data Register A Channel %s	0x3F4	32	R	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSCCDRBCH%s	Capture Current Data Register B Channel %s	0x3F8	32	R	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSOCDRCH%s	Overcurrent Data Register Channel %s	0x3FC	32	R	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSCOCDR0CH%s	Capture Overcurrent Data Register 0 Channel %s	0x400	32	R	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSCOCDR1CH%s	Capture Overcurrent Data Register 1 Channel %s	0x404	32	R	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSCOCDR2CH%s	Capture Overcurrent Data Register 2 Channel %s	0x408	32	R	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSCSRCH%s	Status Register Channel %s	0x420	32	R	0x00000000	0xFFFFFFFF
DSMIF0-1	3	0x140	0-2	DSCSCRCH%s	Status Clear Register Channel %s	0x430	32	W	0x00000000	0xFFFFFFFF
HDSLD0-1	—	—	—	SYS_CTRL	System Control Register	0x00	8	R/W	0x80	0xFF
HDSLD0-1	—	—	—	SYNC_CTRL	Synchronization Control Register	0x01	8	W	0x00	0xFF
HDSLD0-1	—	—	—	MASTER_QM	Quality Monitoring Register	0x03	8	R	0x08	0xFF
HDSLD0-1	—	—	—	EVENT_H	High Byte Event Register	0x04	8	R/W	0x00	0xFF
HDSLD0-1	—	—	—	EVENT_L	Low Byte Event Register	0x05	8	R/W	0x00	0xFF
HDSLD0-1	—	—	—	MASK_H	High Byte Event Mask Register	0x06	8	W	0x00	0xFF
HDSLD0-1	—	—	—	MASK_L	Low Byte Event Mask Register	0x07	8	W	0x00	0xFF
HDSLD0-1	—	—	—	MASK_SUM	Summary Mask Register	0x08	8	W	0x00	0xFF
HDSLD0-1	—	—	—	EDGES	Edge Register	0x09	8	R	0x00	0xFF
HDSLD0-1	—	—	—	DELAY	Run Time Delay Register	0x0A	8	R	0x00	0xFF
HDSLD0-1	—	—	—	VERSION	Version Register	0x0B	8	R	0x57	0xFF
HDSLD0-1	—	—	—	ENC_ID2	Encoder ID 2 Register	0x0D	8	R	0x00	0xFF
HDSLD0-1	—	—	—	ENC_ID1	Encoder ID 1 Register	0x0E	8	R	0x00	0xFF
HDSLD0-1	—	—	—	ENC_ID0	Encoder ID 0 Register	0x0F	8	R	0x00	0xFF
HDSLD0-1	—	—	—	POS4	Fast Position Byte 4 Register	0x10	8	R	0x00	0xFF
HDSLD0-1	—	—	—	POS3	Fast Position Byte 3 Register	0x11	8	R	0x00	0xFF
HDSLD0-1	—	—	—	POS2	Fast Position Byte 2 Register	0x12	8	R	0x00	0xFF
HDSLD0-1	—	—	—	POS1	Fast Position Byte 1 Register	0x13	8	R	0x00	0xFF
HDSLD0-1	—	—	—	POS0	Fast Position Byte 0 Register	0x14	8	R	0x00	0xFF
HDSLD0-1	—	—	—	VEL2	Speed Byte 2 Register	0x15	8	R	0x00	0xFF
HDSLD0-1	—	—	—	VEL1	Speed Byte 1 Register	0x16	8	R	0x00	0xFF
HDSLD0-1	—	—	—	VEL0	Speed Byte 0 Register	0x17	8	R	0x00	0xFF
HDSLD0-1	—	—	—	MIR_SUM	Mirror Status Summary Register	0x18	8	R	0x00	0xFF
HDSLD0-1	8	0x1	0-7	PC_BUF%s	Parameter Channel Buffer %s Register	0x20	8	R/W	0x00	0xFF
HDSLD0-1	—	—	—	PC_ADD_H	High Byte Long Message Address Register	0x28	8	R/W	0x00	0xFF
HDSLD0-1	—	—	—	PC_ADD_L	Low Byte Long Message Address Register	0x29	8	W	0x00	0xFF
HDSLD0-1	—	—	—	PC_OFF_H	High Byte Long Message Address Offset Register	0x2A	8	R/W	0x00	0xFF

Table 1.2 Register description (41 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
HDSLD0-1	—	—	—	PC_OFF_L	Low Byte Long Message Address Offset Register	0x2B	8	W	0x00	0xFF
HDSLD0-1	—	—	—	PC_CTRL	Parameter Channel Control Register	0x2C	8	W	0x00	0xFF
HDSLD0-1	—	—	—	PIPE_S	SensorHub Channel Status Register	0x2D	8	R	0x00	0xFF
HDSLD0-1	—	—	—	PIPE_D	SensorHub Channel Data Register	0x2E	8	R	0x00	0xFF
HDSLD0-1	—	—	—	PC_DATA	Parameter Channel Short Message Mirror Register	0x2F	8	R	0x00	0xFF
HDSLD0-1	—	—	—	ACC_ERR_CNT	Fast Position Error Counter Register	0x38	8	R/W	0x00	0xFF
HDSLD0-1	—	—	—	MAXACC	Fast Position Acceleration Boundary Register	0x39	8	W	0x00	0xFF
HDSLD0-1	—	—	—	MAXDEV_H	Fast Position Estimator Deviation High Byte Register	0x3A	8	R/W	0xFF	0xFF
HDSLD0-1	—	—	—	MAXDEV_L	Fast Position Estimator Deviation Low Byte Register	0x3B	8	R/W	0xFF	0xFF
HDSLD0-1	—	—	—	DUMMY	Dummy Register	0x3F	8	R	0x00	0x00
HDSLD0-1	8	0x1	0-7	MIR_ST%s	Mirror Status %s Register	0x60	8	R/W	0x00	0xFF
HDSLS10-1	—	—	—	EDGES	Edge Register	0x09	8	R	0x00	0xFF
HDSLS10-1	—	—	—	DELAY	Run Time Delay Register	0x0A	8	R	0x00	0xFF
HDSLS10-1	—	—	—	VERSION	Version Register	0x0B	8	R	0x57	0xFF
HDSLS10-1	—	—	—	ENC_ID2	Encoder ID 2 Register	0x0D	8	R	0x00	0xFF
HDSLS10-1	—	—	—	ENC_ID1	Encoder ID 1 Register	0x0E	8	R	0x00	0xFF
HDSLS10-1	—	—	—	ENC_ID0	Encoder ID 0 Register	0x0F	8	R	0x00	0xFF
HDSLS10-1	—	—	—	VPOS4	Safe Position 4 Register	0x19	8	R	0x00	0xFF
HDSLS10-1	—	—	—	VPOS3	Safe Position 3 Register	0x1A	8	R	0x00	0xFF
HDSLS10-1	—	—	—	VPOS2	Safe Position 2 Register	0x1B	8	R	0x00	0xFF
HDSLS10-1	—	—	—	VPOS1	Safe Position 1 Register	0x1C	8	R	0x00	0xFF
HDSLS10-1	—	—	—	VPOS0	Safe Position 0 Register	0x1D	8	R	0x00	0xFF
HDSLS10-1	—	—	—	VPOSCRC_H	Safe Position CRC High Byte Register	0x1E	8	R	0x00	0xFF
HDSLS10-1	—	—	—	VPOSCRC_L	Safe Position CRC Low Byte Register	0x1F	8	R	0x00	0xFF
HDSLS10-1	—	—	—	SAFE_CTRL	Safe System Control Register	0x35	8	R/W	0x00	0xFF
HDSLS10-1	—	—	—	SAFE_SUM	Safety Status Summary Register	0x36	8	R	0x00	0xFF
HDSLS10-1	—	—	—	S_PC_DATA	Parameter Channel Short Message Register	0x37	8	R	0x00	0xFF
HDSLS10-1	—	—	—	EVENT_S	Safe Event Register	0x3D	8	R/W	0x00	0xFF
HDSLS10-1	—	—	—	MASK_S	Safe Event Mask Register	0x3E	8	W	0x00	0xFF
HDSLS10-1	—	—	—	DUMMY	Dummy Register	0x3F	8	R	0x00	0x00
HDSLS10-1	8	0x1	0-7	ENC_ST%s	Encoder Status %s Register	0x40	8	R/W	0x00	0xFF
HDSLS10-1	—	—	—	SRSI	Slave RSSI Register	0x7C	8	R	0x00	0xFF
HDSLS10-1	—	—	—	MAIL	Slave Mail Register	0x7E	8	W	0x00	0xFF
HDSLS10-1	—	—	—	PING	Slave Ping Register	0x7F	8	R/W	0x00	0xFF
BISS0-1	8	0x08	1-8	SCDATAL%s	Sensor Data L	0x00	32	R/W	0x00000000	0xFFFFFFFF
BISS0-1	8	0x08	1-8	SCDATAH%s	Sensor Data H	0x04	32	R/W	0x00000000	0xFFFFFFFF
BISS0-1	16	0x04	1-16	RDATA%s	Register Data %s	0x80	32	R/W	0x00000000	0xFFFFFFFF
BISS0-1	8	0x04	1-8	CNFSLV%s	Configuration Slave %s	0xC0	32	R/W	0x00000000	0xFFFFFFFF
BISS0-1	—	—	—	REGACC	Register Access	0xE0	32	R/W	0x00000000	0xFFFFFFFF
BISS0-1	—	—	—	CTRLCOMM	Control Communication	0xE4	32	R/W	0x00000000	0xFFFFFFFF
BISS0-1	—	—	—	MSTRCNF	Master Configuration	0xE8	32	R/W	0x41060000	0xFFFFFFFF
BISS0-1	—	—	—	CHCNF	Channel Configuration	0xEC	32	R/W	0x00000001	0xFFFFFFFF
BISS0-1	—	—	—	STINF	Status Information	0xF0	32	R/W	0x000000F9	0xFFFFFFFF

Table 1.2 Register description (42 of 42)

Peripheral name	Dim	Dim inc.	Dim index	Register name	Description	Address offset	Size	R/W	Reset value	Reset mask
BISS0-1	—	—	—	INSTREG	Instruction Register	0xF4	32	R/W	0x00000000	0xFFFFFFFF
BISS0-1	—	—	—	STINF2	Status Information 2	0xF8	32	R	0x00000000	0xFFFF0000
ENDAT0-1	—	—	—	SEND	Send Register	0x00	32	R/W	0x07000000	0xFFFFFFFF
ENDAT0-1	—	—	—	EMPFR1_L	Receive Register 1, Low Double Word	0x04	32	R	0x00000000	0xFFFFFFFF
ENDAT0-1	—	—	—	EMPFR1_H	Receive Register 1, High Double Word	0x08	32	R	0x00400000	0xFFFFFFFF
ENDAT0-1	—	—	—	EMPFR2	Receive Register 2	0x0C	32	R	0x00000000	0xFFFFFFFF
ENDAT0-1	—	—	—	EMPFR3	Receive Register 3	0x10	32	R	0x00000000	0xFFFFFFFF
ENDAT0-1	—	—	—	KONFR1	Configuration Register 1	0x14	32	R/W	0x00000DF0	0xFFFFFFFF
ENDAT0-1	—	—	—	KONFR2	Configuration Register 2	0x18	32	R/W	0x00040000	0xFFFFFFFF
ENDAT0-1	—	—	—	STATR	Status Register	0x20	32	R/W	0x40000400	0xFFFFFFFF
ENDAT0-1	—	—	—	INTMR	Interrupt Mask Register	0x24	32	R/W	0x00000000	0xFFFFFFFF
ENDAT0-1	—	—	—	SWSTRBR	SW Strobe Register	0x38	32	W	0x40000400	0xFFFFFFFF
ENDAT0-1	—	—	—	IDR	ID Register	0x3C	32	R	0xE2260013	0xFFFFFFFF
AFMT0-1	—	—	—	COMMAND	Command Code Encoder Address Register	0x000	32	W	0x00000000	0xFFFFFFFF
AFMT0-1	8	0x020	1-8	ENC%sRXDATA0	ENC%s Receive Data 0 Register	0x000	32	R	0x00000000	0xFFFFFFFF
AFMT0-1	8	0x020	1-8	ENC%sRXDATA1	ENC%s Receive Data 1 Register	0x004	32	R	0x00000000	0xFFFFFFFF
AFMT0-1	—	—	—	ROMDATA	EEPROM Address and Write Data Register	0x004	32	W	0x00000000	0xFFFFFFFF
AFMT0-1	8	0x020	1-8	ENC%sRXDATA2	ENC%s Receive Data 2 Register	0x008	32	R	0x00000000	0xFFFFFFFF
AFMT0-1	—	—	—	ID	ID Code Register	0x008	32	W	0x00000000	0xFFFFFFFF
AFMT0-1	8	0x020	1-8	ENC%sRXDATA3	ENC%s Receive Data 3 Register	0x00C	32	R	0x00000000	0xFFFFFFFF
AFMT0-1	8	0x020	1-8	ENC%sRXDATA4	ENC%s Receive Data 4 Register	0x010	32	R	0x00000000	0xFFFFFFFF
AFMT0-1	8	0x020	1-8	ENC%sRXDATA5	ENC%s Receive Data 5 Register	0x014	32	R	0x00000000	0xFFFFFFFF
AFMT0-1	8	0x020	1-8	ENC%sRXDATA6	ENC%s Receive Data 6 Register	0x018	32	R	0x00000000	0xFFFFFFFF
AFMT0-1	8	0x020	1-8	ENC%sRXDATA7	ENC%s Receive Data 7 Register	0x01C	32	R	0x00000000	0xFFFFFFFF
AFMT0-1	—	—	—	TRGSEL	Trigger Select Register	0x100	32	W	0x00000000	0xFFFFFFFF
AFMT0-1	—	—	—	TXTRG	Transmission Trigger Register	0x104	32	W	0x00000000	0xFFFFFFFF
AFMT0-1	—	—	—	BRSEL	Baud Rate Setting Register	0x200	32	W	0x00000000	0xFFFFFFFF
ENCOUT	—	—	—	CTL	Control Register	0x000	32	R/W	0x00000000	0xFFFFFFFF
ENCOUT	—	—	—	STR	Start Register	0x004	32	R/W	0x00000000	0xFFFFFFFF
ENCOUT	—	—	—	POSMAX_PERIOD	Maximum Position and Carrier Period Register	0x00C	32	R/W	0xFFFF0000	0xFFFFFFFF
ENCOUT	—	—	—	OUTCNT	Output Count Register	0x010	32	R/W	0x00000000	0xFFFFFFFF
ENCOUT	—	—	—	POSCNT	Position Count Register	0x014	32	R/W	0x00000000	0xFFFFFFFF
GSC	—	—	—	CNTCR	Global System Counter Control Register	0x00	32	R/W	0x00000000	0xFFFFFFFF
GSC	—	—	—	CNTSR	Global System Counter Status Register	0x04	32	R	0x00000000	0xFFFFFFFF
GSC	—	—	—	CNTCVL	Global System Counter Current Value Lower Register	0x08	32	R/W	0x00000000	0xFFFFFFFF
GSC	—	—	—	CNTCVU	Global System Counter Current Value Upper Register	0x0C	32	R/W	0x00000000	0xFFFFFFFF
GSC	—	—	—	CNTFID0	Global System Counter Base Frequency ID Register	0x20	32	R/W	0x00000000	0xFFFFFFFF

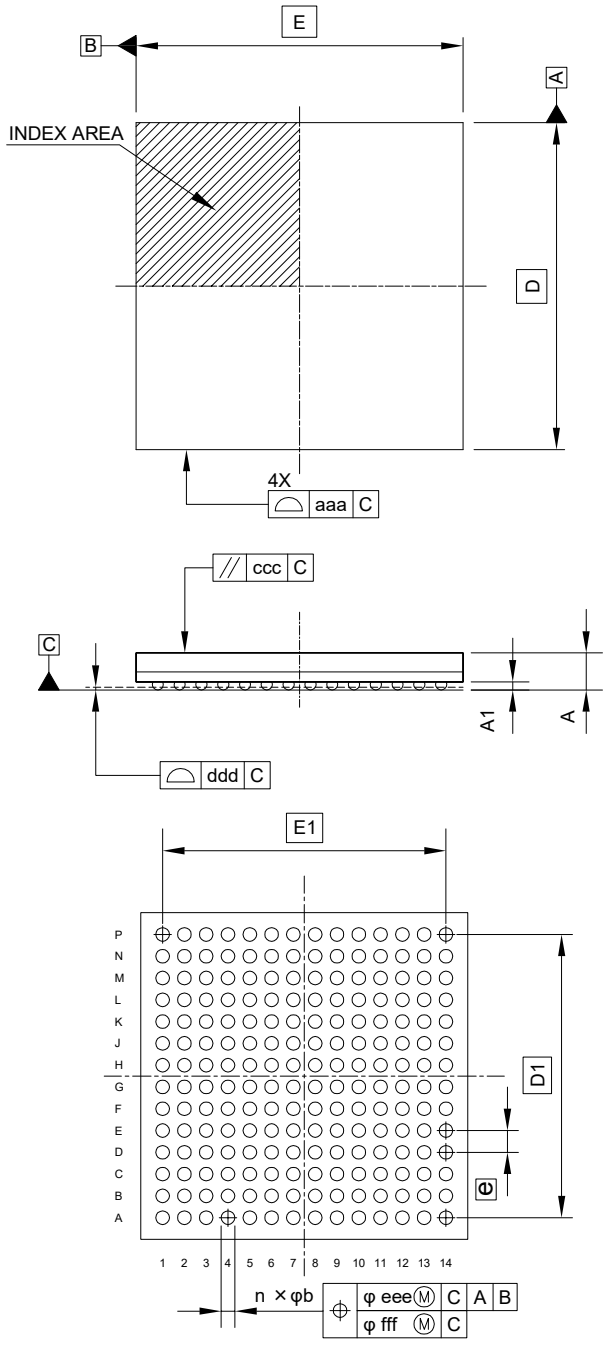
Note: Peripheral name = Name of peripheral
Dim = Number of elements in an array of registers
Dim inc. = Address increment between two simultaneous registers of a register array in the address map
Dim index = Sub string that replaces the %s placeholder within the register name
Register name = Name of register
Description = Register description
Address offset = Address of the register relative to the base address defined by the peripheral of the register
Size = Bit width of the register
Reset value = Default reset value of a register

Reset mask = Identifies which register bits have a defined reset value

Appendix 2. Package Dimensions

Information on the latest version of the package dimensions or mountings is displayed in "Packages" on the Renesas Electronics Corporation website.

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-LFBGA196-12x12-0.80	PLBG0196GC-A	0.38



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
D	—	12.00	—
E	—	12.00	—
D1	—	10.40	—
E1	—	10.40	—
A	—	—	1.43
A1	0.25	—	—
b	0.36	0.41	0.46
e	—	0.80	—
aaa	—	—	0.15
ccc	—	—	0.10
ddd	—	—	0.10
eee	—	—	0.15
fff	—	—	0.08
n	—	196	—

Figure 2.1 196-pin FBGA

Revision History

Revision 1.00 — November 30, 2022

- Initial release

Revision 1.10 — February 28, 2023

3. Operating Modes:
- Corrected 3.4.1 MD_MON : Operating Mode Monitor Register.

5. Reset:
- Corrected 5.2.8 MRCTLK : Module Reset Control Register K.

6. Clock Generation Circuit:
- Corrected Figure 6.1 Block diagram of clock generation circuit.
- Corrected 6.2.2 SCKCR2 : System Clock Control Register 2.

9. Debugging Interface:
- Corrected Table 9.4 CoreSight address map (Debug-APB).

11. Interrupt Controller (ICU):
- Corrected 11.2.1.6 PERIERR_STAT0 : Peripheral Error Event Status Register 0.
- Corrected 11.2.1.21 PERIERR_STAT3 : Peripheral Error Event Status Register 3.
- Corrected Table 11.2 Event table.
- Corrected 11.3.4 Noise Filter.
- Added Figure 11.2 Digital noise filter operation example.

14. DMA Controller (DMAC):
- Corrected 14.5.3 Setting Example 3 (Register Mode Continuous Execution).

15. Event Link Controller (ELC):
- Corrected 15.2.1 ELC_SSELn : ELC Event Source Select Register n (n = 0 to 18).
- Corrected Table 15.2 ELC Event Source.
- Corrected 15.3.1 Event Linkage.
- Corrected 15.3.6 Example of Procedure for Linking Events.

17. Multi-Function Timer Pulse Unit 3 (MTU3):
- Corrected 17.2.1 TCR : Timer Control Register.

19. General PWM Timer (GPT):
- Corrected 19.2.13 GTUDDTYC : General PWM Timer Count Direction and Duty Setting Register.
- Corrected 19.3.6 Function of Output Duty 0% and 100%.

21. Arithmetic Unit for Trigonometric Functions (TFU):
- Corrected 21.2.3 TRGSTS : Trigonometric Status Register.
- Changed title 21.2.8 FXSCDT0 : Fixed-point Sine Cosine Data Register 0.

26. Ethernet Subsystem:
- Corrected 26.2.5 CONVCTRLn : RGMII/RMII Converter n Control Register (n = 0 to 2).
- Added Table 26.5 Ethernet interface support mode.

28. EtherCAT Slave Controller (ESC):
- Added 28.21 Usage Notes.

29. USB 2.0 Host Module (USBHC):
- Corrected Figure 29.7 Initial setting sequence.

30. USB 2.0 HS Function Module (USBf):
- Corrected Figure 30.6 Startup sequence.

31. Serial Communications Interface (SCI):
- Corrected 31.2.2 RDR : Receive Data Register.

33. CAN-FD Interface (CANFD):
- Added 33.11 Flexible CAN Mode.
- Added Figure 33.59 Diagram of the Flexible CAN.
- Added 33.12 Flexible Transmission Buffer Assignment.
- Added Figure 33.60 The priority of message buffer number priority mode (Example of CH0: 96 buffers, CH1: 32 buffers).
- Added Figure 33.61 Flexible transmission buffer assignment (0, 32, 28 TXMB rental).
- Added Figure 33.62 Flexible transmission buffer assignment (24, 20, 16 TXMB rental).
- Added Figure 33.63 Flexible transmission buffer assignment (12, 8, 4 TXMB rental).

35. Expanded Serial Peripheral Interface (xSPI):
- Corrected Figure 35.25 Flow of memory-mapping stop.

Revision 1.10 — February 28, 2023

38. ΔΣ Interface (DSMIF):
- Added 38.3.16 Common Interrupt.
- Added Table 38.12 Data update common interrupt.
- Added Table 38.13 Mode of common interrupt.
- Added Figure 38.11 Common interrupt output timing (Three channel synchronous mode).
- Added Figure 38.12 Common interrupt output timing (Two channel synchronous mode).

39. 12-Bit A/D converter (ADC12):
- Corrected Figure 39.16 A/D Conversion stop procedure.

44. Serial Host Interface (SHOSTIF):
- Corrected Table 44.5 Address mapping.

47. EnDat 2.2 (ENDAT):
- Corrected 47.2.4 STATR : Status Register.
- Corrected 47.2.5 INTMR : Interrupt Mask Register.

49. HIPERFACE DSL (HDSL):
- Corrected Table 49.9 Safe channel 1 interface register map.
- Changed title 49.2.2.10 SRSSI : Slave RSSI Register.

50. A-format (AFMT):
- Corrected Table 50.1 A-format (AFMT) specifications.
- Corrected Figure 50.1 Block diagram of A-format (AFMT).
- Corrected Table 50.12 Encoder status (ES[3:0] and ES).
- Corrected Table 50.13 Status code (ALM[15:0]).

51. Encoder Divided-Output Module (ENCOUT):
- Corrected Table 51.1 ENCOUT specifications.

52. Electrical Characteristics:
- Corrected Table 52.35 SPI timing.

Appx 2. Package Dimensions:
- Corrected Figure 2.1 196-pin FBGA.

Rvision 1.20 — November 30, 2023

1. Overview:
- Updated Table 1.7 Timers.

3. Operating Modes:
- Added 3.5.13.2 Bus Conflict on 16-Bit and 32-Bit Bus Boot Modes
- Added 3.5.13.3 Serial Flash Reset on xSPI Boot Modes (X1 Boot Serial Flash)

9. Debugging Interface:
- Added 9.4.3 Debug Reset

11. Interrupt Controller (ICU):
- Updated 11.2.1.4 S_PORTNF_MD : Interrupt Edge Detection Setting Register for Safety Register.
- Updated 11.2.2.4 NS_PORTNF_MD : Interrupt Edge Detection Setting Register.

26. Ethernet Subsystem:
- Updated 26.3.1 Ethernet Initializing Operation.

34. CAN-FD Interface (CANFD):
- Updated Table 34.1 CANFD specifications.
- Updated 34.5.1 Overview.

36. Expanded Serial Peripheral Interface (xSPI):
- Updated 35.2.3.2 COMSTT : xSPI Common Status Register
- Updated 35.2.4.1 INTS : xSPI Interrupt Status Register
- Updated 35.2.4.2 INTC : xSPI Interrupt Clear Register
- Updated 35.2.4.3 INTE : xSPI Interrupt Enable Register

45. Electrical Characteristics:
- Updated Table 45.40 Serial management interface timnig.

RZ/T2L Group User’s Manual: Hardware

Publication Date: Rev.1.20 Oct. 31, 2023
Rev.1.10 Feb. 28, 2023

Published by: Renesas Electronics Corporation

RZ/T Series for Real-Time Control
RZ/T2L Group

Renesas Electronics Corporation

R01UH0985EJ0120