User's Manual

32

RX13T Group

User's Manual: Hardware

RENESAS 32-Bit MCU
RX Family／RX100 Series

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.1.10 Mar 2021

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.

© 2021 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.

How to Use This Manual

1. Objective and Target Users

This manual was written to explain the hardware functions and electrical characteristics of this LSI to the target users, i.e. those who will be using this LSI in the design of application systems. Target users are expected to understand the fundamentals of electrical circuits, logic circuits, and microcomputers.
This manual is organized in the following items: an overview of the product, descriptions of the CPU, system control functions, and peripheral functions, electrical characteristics of the device, and usage notes.

When designing an application system that includes this LSI, take all points to note into account.
Points to note are given in their contexts and at the final part of each section, and in the section giving usage notes.

The list of revisions is a summary of major points of revision or addition for earlier versions. It does not cover all revised items. For details on the revised points, see the actual locations in the manual.

The following documents have been prepared for the RX13T Group. Before using any of the documents, please visit our website to verify that you have the most up-to-date available version of the document.

Document Type	Contents	Document Title	Document No.
Datasheet	Overview of hardware and electrical characteristics	RX13T Group Datasheet	R01DS0341EJ
User's Manual: Hardware	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and descriptions of operation	RX13T Group User's Manual: Hardware	This User's manual
User's Manual: Software	Detailed descriptions of the CPU and instruction set	RX Family RXv1 Instruction Set Architecture User's Manual: Software	R01US0032EJ
Application Note	Notes on Printed Circuit Board Patterns	RX Family Hardware Design Guide	R01AN1411EJ
	Examples of register initial setting	RX13T Group Initial Setting Examples	—
	Examples of applications and sample programs	—	—
Renesas Technical Update	Preliminary report on the specifications of a product, document, etc.	—	—

2. Description of Registers

Each register description includes a bit chart, illustrating the arrangement of bits, and a table of bits, describing the meanings of the bit settings. The standard format and notation for bit charts and tables are described below.

X.X.X ... Register

Address(es): xxxx xxxxh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	...[1:0]		...4	—	—	—	...0
Value after reset:	x	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	...0	.....	0: ..... 1: Setting prohibited (3)	R/W (1)
b3 to b1	—	Reserved (2)	These bits are read as 0. The write value should be 0.	R/W
b4	...4	.....	0: 1:	R
b6, b5	...[1:0]	.....	0 0: ..... 0 1: ..... Settings other than above are prohibited. (3)	R/(W)*1
b7	—	Reserved	The read value is undefined. Writing to this bit has no effect.	R

(1) R/W: The bit or field is readable and writable.
R/(W): The bit or field is readable and writable. However, writing to this bit or field has some limitations. For details on the limitations, see the description or notes of respective registers.
R: The bit or field is readable. Writing to this bit or field has no effect.

(2) Reserved.
Use the specified value when writing to this bit or field; otherwise, the correct operation is not guaranteed.

(3) Setting prohibited. The correct operation is not guaranteed if such a setting is performed.

3. List of Abbreviations and Acronyms

Abbreviation	Full Form
ACIA	Asynchronous Communications Interface Adapter
bps	bits per second
CRC	Cyclic Redundancy Check
DMA	Direct Memory Access
DMAC	Direct Memory Access Controller
GSM	Global System for Mobile Communications
Hi-Z	High Impedance
IEBus	Inter Equipment Bus
I/O	Input/Output
IrDA	Infrared Data Association
LSB	Least Significant Bit
MSB	Most Significant Bit
NC	Non-Connect
PLL	Phase Locked Loop
PWM	Pulse Width Modulation
SIM	Subscriber Identity Module
UART	Universal Asynchronous Receiver/Transmitter
VCO	Voltage Controlled Oscillator

All trademarks and registered trademarks are the property of their respective owners.
SuperFlash® is a registered trademark of Silicon Storage Technology, Inc. in several countries including the United States and Japan.

Caution: This product uses SuperFlash® technology licensed from Silicon Storage Technology, Inc.

Contents

Features 32

1. Overview 33

1.1 Outline of Specifications 33

1.2 List of Products 37

1.3 Block Diagram 39

1.4 Pin Functions 40

1.5 Pin Assignments 42

1.5.1 48-Pin LFQFP 42

1.5.2 48-Pin HWQFN 43

1.5.3 32-Pin LQFP 44

1.5.4 32-Pin HWQFN 45

1.6 List of Pins and Pin Functions 46

1.6.1 48-Pin LFQFP/HWQFN 46

1.6.2 32-Pin LQFP/HWQFN 48

2. CPU 49

2.1 Features 49

2.2 Register Set of the CPU 50

2.2.1 General-Purpose Registers (R0 to R15) 51

2.2.2 Control Registers 51

2.2.2.1 Interrupt Stack Pointer (ISP)/User Stack Pointer (USP) 52

2.2.2.2 Interrupt Table Register (INTB) 52

2.2.2.3 Program Counter (PC) 52

2.2.2.4 Processor Status Word (PSW) 53

2.2.2.5 Backup PC (BPC) 54

2.2.2.6 Backup PSW (BPSW) 55

2.2.2.7 Fast Interrupt Vector Register (FINTV) 55

2.2.2.8 Floating-Point Status Word (FPSW) 55

2.2.3 Register Associated with DSP Instructions 58

2.2.3.1 Accumulator (ACC) 58

2.3 Processor Mode 59

2.3.1 Supervisor Mode 59

2.3.2 User Mode 59

2.3.3 Privileged Instruction 59

2.3.4 Switching between Processor Modes 59

2.4 Data Types 60

2.5 Endian 60

2.5.1 Switching the Endian 60

2.5.2 Access to I/O Registers 64

2.5.3 Notes on Access to I/O Registers 64

2.5.4 Data Arrangement 64

2.5.4.1 Data Arrangement in Registers 64

2.5.4.2 Data Arrangement in Memory ... 65
2.5.5 Notes on the Allocation of Instruction Codes ... 65
2.6 Vector Table ... 66
2.6.1 Fixed Vector Table ... 66
2.6.2 Relocatable Vector Table ... 67
2.7 Operation of Instructions ... 68
2.7.1 Data Prefetching by the RMPA Instruction and the String-Manipulation Instructions ... 68
2.8 Pipeline ... 68
2.8.1 Overview ... 68
2.8.2 Instructions and Pipeline Processing ... 70
2.8.2.1 Instructions Converted into Single Micro-Operation and Pipeline Processing ... 70
2.8.2.2 Instructions Converted into Multiple Micro-Operations and Pipeline Processing ... 72
2.8.2.3 Pipeline Basic Operation ... 75
2.8.3 Calculation of the Instruction Processing Time ... 77
2.8.4 Numbers of Cycles for Response to Interrupts ... 78
3. Operating Modes ... 79
3.1 Operating Mode Types and Selection ... 79
3.2 Register Descriptions ... 80
3.2.1 Mode Monitor Register (MDMONR) ... 80
3.2.2 System Control Register 1 (SYSCR1) ... 81
3.3 Details of Operating Modes ... 82
3.3.1 Single-Chip Mode ... 82
3.3.2 Boot Mode ... 82
3.3.2.1 Boot Mode (SCI) ... 82
3.4 Transitions of Operating Modes ... 83
3.4.1 Operating Mode Transitions Determined by the Mode-Setting Pins ... 83
4. Address Space ... 84
4.1 Address Space ... 84
5. I/O Registers ... 86
5.1 I/O Register Addresses (Address Order) ... 88
6. Resets ... 99
6.1 Overview ... 99
6.2 Register Descriptions ... 101
6.2.1 Reset Status Register 0 (RSTSR0) ... 101
6.2.2 Reset Status Register 1 (RSTSR1) ... 102
6.2.3 Reset Status Register 2 (RSTSR2) ... 103
6.2.4 Software Reset Register (SWRR) ... 104
6.3 Operation ... 105
6.3.1 RES# Pin Reset ... 105
6.3.2 Power-On Reset and Voltage Monitoring 0 Reset ... 105
6.3.3 Voltage Monitoring 1 Reset and Voltage Monitoring 2 Reset ... 107

6.3.4 Independent Watchdog Timer Reset 108
6.3.5 Software Reset 108
6.3.6 Determination of Cold/Warm Start 109
6.3.7 Determination of Reset Generation Source 110
7. Option-Setting Memory (OFSM) 111
7.1 Overview 111
7.2 Register Descriptions 112
7.2.1 Option Function Select Register 0 (OFS0) 112
7.2.2 Option Function Select Register 1 (OFS1) 114
7.2.3 Endian Select Register (MDE) 115
7.3 Usage Note 116
7.3.1 Setting Example of Option-Setting Memory 116
8. Voltage Detection Circuit (LVDAb) 117
8.1 Overview 117
8.2 Register Descriptions 120
8.2.1 Voltage Monitoring 1 Circuit Control Register 1 (LVD1CR1) 120
8.2.2 Voltage Monitoring 1 Circuit Status Register (LVD1SR) 121
8.2.3 Voltage Monitoring 2 Circuit Control Register 1 (LVD2CR1) 122
8.2.4 Voltage Monitoring 2 Circuit Status Register (LVD2SR) 123
8.2.5 Voltage Monitoring Circuit Control Register (LVCMPCR) 124
8.2.6 Voltage Detection Level Select Register (LVDLVLR) 125
8.2.7 Voltage Monitoring 1 Circuit Control Register 0 (LVD1CR0) 126
8.2.8 Voltage Monitoring 2 Circuit Control Register 0 (LVD2CR0) 127
8.3 VCC Input Voltage Monitor 128
8.3.1 Monitoring Vdet0 128
8.3.2 Monitoring Vdet1 128
8.3.3 Monitoring Vdet2 128
8.4 Reset from Voltage Monitor 0 129
8.5 Interrupt and Reset from Voltage Monitoring 1 130
8.6 Interrupt and Reset from Voltage Monitoring 2 132
9. Clock Generation Circuit 134
9.1 Overview 134
9.2 Register Descriptions 136
9.2.1 System Clock Control Register (SCKCR) 136
9.2.2 System Clock Control Register 3 (SCKCR3) 138
9.2.3 PLL Control Register (PLLCR) 139
9.2.4 PLL Control Register 2 (PLLCR2) 140
9.2.5 Main Clock Oscillator Control Register (MOSCCR) 141
9.2.6 Low-Speed On-Chip Oscillator Control Register (LOCOCR) 142
9.2.7 IWDT-Dedicated On-Chip Oscillator Control Register (ILOCOCR) 143
9.2.8 High-Speed On-Chip Oscillator Control Register (HOCOCR) 144

9.2.9 Oscillation Stabilization Flag Register (OSCOVFSR) 145
9.2.10 Oscillation Stop Detection Control Register (OSTDCR) 147
9.2.11 Oscillation Stop Detection Status Register (OSTDSR) 148
9.2.12 Main Clock Oscillator Wait Control Register (MOSCWTCR) 149
9.2.13 Main Clock Oscillator Forced Oscillation Control Register (MOFCR) 150
9.2.14 Low-Speed On-Chip Oscillator Trimming Register (LOCOTRR) 151
9.2.15 IWDT-Dedicated On-Chip Oscillator Trimming Register (ILOCOTRR) 151
9.2.16 High-Speed On-Chip Oscillator Trimming Register n (HOCOTRRn) (n = 0) 152
9.3 Main Clock Oscillator 153
9.3.1 Connecting a Crystal 153
9.3.2 External Clock Input 154
9.3.3 Notes on the External Clock Input 154
9.4 Oscillation Stop Detection Function 155
9.4.1 Oscillation Stop Detection and Operation after Detection 155
9.4.2 Oscillation Stop Detection Interrupts 156
9.5 PLL Circuit 157
9.6 Internal Clock 157
9.6.1 System Clock 157
9.6.2 Peripheral Module Clock 157
9.6.3 FlashIF Clock 157
9.6.4 CAC Clock 157
9.6.5 IWDT-Dedicated Clock 157
9.7 Usage Notes 159
9.7.1 Notes on Clock Generation Circuit 159
9.7.2 Note on Rewriting the SCKCR3 Register 159
9.7.3 Notes on Resonator 159
9.7.4 Notes on Board Design 159
9.7.5 Notes on Resonator Connection Pins 161
10. Clock Frequency Accuracy Measurement Circuit (CAC) 162
10.1 Overview 162
10.2 Register Descriptions 164
10.2.1 CAC Control Register 0 (CACR0) 164
10.2.2 CAC Control Register 1 (CACR1) 165
10.2.3 CAC Control Register 2 (CACR2) 166
10.2.4 CAC Interrupt Request Enable Register (CAICR) 167
10.2.5 CAC Status Register (CASTR) 168
10.2.6 CAC Upper-Limit Value Setting Register (CAULVR) 169
10.2.7 CAC Lower-Limit Value Setting Register (CALLVR) 169
10.2.8 CAC Counter Buffer Register (CACNTBR) 169
10.3 Operation 170
10.3.1 Measuring Clock Frequency 170

10.3.2 Digital Filtering of Signals on the CACREF Pin ... 171
10.4 Interrupt Requests ... 171
10.5 Usage Notes ... 172
10.5.1 Module Stop Function Setting ... 172
11. Low Power Consumption ... 173
11.1 Overview ... 173
11.2 Register Descriptions ... 177
11.2.1 Standby Control Register (SBYCR) ... 177
11.2.2 Module Stop Control Register A (MSTPCRA) ... 178
11.2.3 Module Stop Control Register B (MSTPCRB) ... 179
11.2.4 Module Stop Control Register C (MSTPCRC) ... 180
11.2.5 Operating Power Control Register (OPCCR) ... 181
11.3 Reducing Power Consumption by Switching Clock Signals ... 183
11.4 Module Stop Function ... 183
11.5 Function for Lower Operating Power Consumption ... 184
11.5.1 Setting Operating Power Control Mode ... 184
11.6 Low Power Consumption Modes ... 185
11.6.1 Sleep Mode ... 185
11.6.1.1 Entry to Sleep Mode ... 185
11.6.1.2 Exit from Sleep Mode ... 186
11.6.2 Deep Sleep Mode ... 187
11.6.2.1 Entry to Deep Sleep Mode ... 187
11.6.2.2 Exit from Deep Sleep Mode ... 188
11.6.3 Software Standby Mode ... 189
11.6.3.1 Entry to Software Standby Mode ... 189
11.6.3.2 Exit from Software Standby Mode ... 190
11.6.3.3 Example of Software Standby Mode Application ... 191
11.7 Usage Notes ... 192
11.7.1 I/O Port States ... 192
11.7.2 Module Stop State of DTC ... 192
11.7.3 On-Chip Peripheral Module Interrupts ... 192
11.7.4 Write Access to MSTPCRA, MSTPCRB, and MSTPCRC ... 192
11.7.5 Timing of WAIT Instructions ... 192
11.7.6 Rewrite the Register by DTC in Sleep Mode ... 192
12. Register Write Protection Function ... 193
12.1 Register Descriptions ... 194
12.1.1 Protect Register (PRCR) ... 194
13. Exception Handling ... 195
13.1 Exception Events ... 195
13.1.1 Undefined Instruction Exception ... 196
13.1.2 Privileged Instruction Exception ... 196

13.1.3 Floating-Point Exception ... 196
13.1.4 Reset ... 196
13.1.5 Non-Maskable Interrupt ... 196
13.1.6 Interrupts ... 196
13.1.7 Unconditional Trap ... 196
13.2 Exception Handling Procedure ... 197
13.3 Acceptance of Exception Events ... 199
13.3.1 Acceptance Timing and Saved PC Value ... 199
13.3.2 Vector and Site for Saving the Values in the PC and PSW ... 199
13.4 Hardware Processing for Accepting and Returning from Exceptions ... 200
13.5 Hardware Pre-Processing ... 201
13.5.1 Undefined Instruction Exception ... 201
13.5.2 Privileged Instruction Exception ... 201
13.5.3 Floating-Point Exception ... 201
13.5.4 Reset ... 201
13.5.5 Non-Maskable Interrupt ... 201
13.5.6 Interrupt ... 202
13.5.7 Unconditional Trap ... 202
13.6 Return from Exception Handling Routine ... 203
13.7 Priority of Exception Events ... 203
14. Interrupt Controller (ICUb) ... 204
14.1 Overview ... 204
14.2 Register Descriptions ... 206
14.2.1 Interrupt Request Register n (IRn) (n = interrupt vector number) ... 206
14.2.2 Interrupt Request Enable Register m (IERm) (m = 02h to 1Fh) ... 207
14.2.3 Interrupt Source Priority Register n (IPRn) (n = interrupt vector number) ... 208
14.2.4 Fast Interrupt Set Register (FIR) ... 209
14.2.5 Software Interrupt Generation Register (SWINTR) ... 210
14.2.6 DTC Transfer Request Enable Register n (DTCERn) (n = interrupt vector number) ... 211
14.2.7 IRQ Control Register i (IRQCRi) (i = 0 to 5) ... 212
14.2.8 IRQ Pin Digital Filter Enable Register 0 (IRQFLTE0) ... 213
14.2.9 IRQ Pin Digital Filter Setting Register 0 (IRQFLTC0) ... 214
14.2.10 Non-Maskable Interrupt Status Register (NMISR) ... 215
14.2.11 Non-Maskable Interrupt Enable Register (NMIER) ... 217
14.2.12 Non-Maskable Interrupt Status Clear Register (NMICLR) ... 218
14.2.13 NMI Pin Interrupt Control Register (NMICR) ... 219
14.2.14 NMI Pin Digital Filter Enable Register (NMIFLTE) ... 219
14.2.15 NMI Pin Digital Filter Setting Register (NMIFLTC) ... 220
14.3 Vector Table ... 221
14.3.1 Interrupt Vector Table ... 221

14.3.2 Fast Interrupt Vector Table 227
14.3.3 Non-maskable Interrupt Vector Table 227
14.4 Interrupt Operation 228
14.4.1 Detecting Interrupts 228
14.4.1.1 Operation of Status Flags for Edge-Detected Interrupts 228
14.4.1.2 Operation of Status Flags for Level-Detected Interrupts 230
14.4.2 Enabling and Disabling Interrupt Sources 231
14.4.3 Selecting Interrupt Request Destinations 232
14.4.4 Determining Priority 233
14.4.5 Multiple Interrupts 233
14.4.6 Fast Interrupt 233
14.4.7 Digital Filter 234
14.4.8 External Pin Interrupts 234
14.5 Non-maskable Interrupt Operation 235
14.6 Return from Power-Down States 236
14.6.1 Return from Sleep Mode or Deep Sleep Mode 236
14.6.2 Return from Software Standby Mode 236
14.7 Usage Note 237
14.7.1 Note on WAIT Instruction Used with Non-Maskable Interrupt 237
15. Buses 238
15.1 Overview 238
15.2 Description of Buses 240
15.2.1 CPU Buses 240
15.2.2 Memory Buses 240
15.2.3 Internal Main Buses 240
15.2.4 Internal Peripheral Buses 241
15.2.5 Write Buffer Function (Internal Peripheral Bus) 242
15.2.6 Parallel Operation 243
15.2.7 Restrictions 243
15.3 Register Descriptions 244
15.3.1 Bus Error Status Clear Register (BERCLR) 244
15.3.2 Bus Error Monitoring Enable Register (BEREN) 244
15.3.3 Bus Error Status Register 1 (BERSR1) 245
15.3.4 Bus Error Status Register 2 (BERSR2) 245
15.3.5 Bus Priority Control Register (BUSPRI) 246
15.4 Bus Error Monitoring Section 248
15.4.1 Type of Bus Error 248
15.4.1.1 Illegal Address Access 248
15.4.1.2 Timeout 248
15.4.2 Operations When a Bus Error Occurs 248
15.4.3 Conditions Leading to Bus Errors 249

15.5 Interrupt 250
15.5.1 Interrupt Source 250
16. Data Transfer Controller (DTCb) 251
16.1 Overview 251
16.2 Register Descriptions 253
16.2.1 DTC Mode Register A (MRA) 253
16.2.2 DTC Mode Register B (MRB) 255
16.2.3 DTC Mode Register C (MRC) 257
16.2.4 DTC Transfer Source Register (SAR) 258
16.2.5 DTC Transfer Destination Register (DAR) 258
16.2.6 DTC Transfer Count Register A (CRA) 259
16.2.7 DTC Transfer Count Register B (CRB) 260
16.2.8 DTC Control Register (DTCCR) 260
16.2.9 DTC Vector Base Register (DTCVBR) 261
16.2.10 DTC Address Mode Register (DTCADMOD) 261
16.2.11 DTC Module Start Register (DTCST) 262
16.2.12 DTC Status Register (DTCSTS) 263
16.2.13 DTC Index Table Base Register (DTCIBR) 264
16.2.14 DTC Operation Register (DTCOR) 265
16.2.15 DTC Sequence Transfer Enable Register (DTCSQE) 266
16.2.16 DTC Address Displacement Register (DTCDISP) 266
16.3 Request Sources 267
16.3.1 Allocating Transfer Information and DTC Vector Table 267
16.4 Operation 269
16.4.1 Transfer Information Read Skip Function 271
16.4.2 Transfer Information Write-Back Skip Function 272
16.4.2.1 Write-Back Skip by Fixing Addresses 272
16.4.2.2 Write-Back Skip by the MRA.WBDIS Bit 272
16.4.3 Normal Transfer Mode 273
16.4.4 Repeat Transfer Mode 274
16.4.5 Block Transfer Mode 275
16.4.6 Chain Transfer 276
16.4.7 Operation Timing 277
16.4.8 Execution Cycles of the DTC 280
16.4.9 DTC Bus Mastership Release Timing 280
16.4.10 Sequence Transfer 281
16.4.11 DTC Index Table 283
16.4.12 Example of Sequence Transfer 285
16.5 DTC Setting Procedure 291
16.6 Examples of DTC Usage 292
16.6.1 Normal Transfer 292

16.6.2 Chain Transfer When the Counter is 0 293
16.6.3 Sequence Transfer 294
16.7 Interrupt Source 295
16.8 Low Power Consumption Function 296
16.9 Usage Notes 297
16.9.1 Start Address of Transfer Information 297
16.9.2 Allocating Transfer Information 297
16.9.3 Notes on Using the Sequence Transfer 298
17. I/O Ports 299
17.1 Overview 299
17.2 I/O Port Configuration 301
17.3 Register Descriptions 303
17.3.1 Port Direction Register (PDR) 303
17.3.2 Port Output Data Register (PODR) 304
17.3.3 Port Input Data Register (PIDR) 305
17.3.4 Port Mode Register (PMR) 306
17.3.5 Open Drain Control Register 0 (ODR0) 307
17.3.6 Open Drain Control Register 1 (ODR1) 308
17.3.7 Pull-Up Control Register (PCR) 309
17.3.8 Drive Capacity Control Register (DSCR) 310
17.4 Initialization of the Port Direction Register (PDR) 311
17.5 Handling of Unused Pins 312
18. Multi-Function Pin Controller (MPC) 313
18.1 Overview 313
18.2 Register Descriptions 316
18.2.1 Write-Protect Register (PWPR) 316
18.2.2 P1n Pin Function Control Register (P1nPFS) (n = 0, 1) 317
18.2.3 P2n Pin Function Control Register (P2nPFS) (n = 2 to 4) 318
18.2.4 P4n Pin Function Control Register (P4nPFS) (n = 0 to 7) 319
18.2.5 P7n Pin Function Control Register (P7nPFS) (n = 0 to 6) 320
18.2.6 P9n Pin Function Control Register (P9nPFS) (n = 3, 4) 321
18.2.7 PAn Pin Function Control Register (PAnPFS) (n = 2, 3) 322
18.2.8 PBn Pin Function Control Register (PBnPFS) (n = 0 to 7) 323
18.2.9 PDn Pin Function Control Register (PDnPFS) (n = 3 to 6) 324
18.2.10 PE2 Pin Function Control Register (PE2PFS) 325
18.3 Usage Notes 326
18.3.1 Procedure for Specifying Input/Output Pin Function 326
18.3.2 Notes on MPC Register Setting 326
18.3.3 Note on Using Analog Functions 327
18.3.4 Note on PB1 Pin Input Level 327

19. Multi-Function Timer Pulse Unit 3 (MTU3c) 328
19.1 Overview 328
19.2 Register Descriptions 333
19.2.1 Timer Control Register (TCR) 333
19.2.2 Timer Control Register 2 (TCR2) 335
19.2.3 Timer Mode Register 1 (TMDR1) 339
19.2.4 Timer Mode Register 2 (TMDR2A) 341
19.2.5 Timer Mode Register 3 (TMDR3) 342
19.2.6 Timer I/O Control Register (TIOR) 343
19.2.7 Timer Compare Match Clear Register (TCNTCMPCLR) 354
19.2.8 Timer Interrupt Enable Register (TIER) 355
19.2.9 Timer Status Register (TSR) 358
19.2.10 Timer Buffer Operation Transfer Mode Register (TBTM) 359
19.2.11 Timer Input Capture Control Register (TICCR) 360
19.2.12 Timer Counter (TCNT) 361
19.2.13 Timer Longword Counter (TCNTLW) 361
19.2.14 Timer General Register (TGR) 362
19.2.15 Timer Longword General Registers (TGRALW, TGRBLW) 362
19.2.16 Timer Start Registers (TSTRA, TSTR) 363
19.2.17 Timer Synchronous Register (TSYRA) 365
19.2.18 Timer Counter Synchronous Start Register (TCSYSTR) 366
19.2.19 Timer Read/Write Enable Register (TRWERA) 367
19.2.20 Timer Output Master Enable Register (TOERA) 368
19.2.21 Timer Output Control Register 1 (TOCR1A) 369
19.2.22 Timer Output Control Register 2 (TOCR2A) 371
19.2.23 Timer Output Level Buffer Register (TOLBRA) 374
19.2.24 Timer Gate Control Register A (TGCRA) 375
19.2.25 Timer Subcounter (TCNTSA) 376
19.2.26 Timer Period Data Register (TCDRA) 376
19.2.27 Timer Period Buffer Register (TCBRA) 377
19.2.28 Timer Dead Time Data Register (TDDRA) 377
19.2.29 Timer Dead Time Enable Register (TDERA) 378
19.2.30 Timer Buffer Transfer Set Register (TBTERA) 379
19.2.31 Timer Waveform Control Register (TWCRA) 380
19.2.32 Noise Filter Control Register n (NFCRn) (n = 0 to 4, C) 381
19.2.33 Noise Filter Control Register 5 (NFCR5) 383
19.2.34 Timer A/D Conversion Start Request Control Register (TADCR) 384
19.2.35 Timer A/D Conversion Start Request Cycle Set Registers (TADCORA, TADCORB) 386
19.2.36 Timer A/D Conversion Start Request Cycle Set Buffer Registers (TADCOBRA, TADCOBRB) 386
19.2.37 Timer Interrupt Skipping Mode Register (TITMRA) 387

19.2.38 Timer Interrupt Skipping Set Register 1 (TITCR1A) 388
19.2.39 Timer Interrupt Skipping Counter 1 (TITCNT1A) 389
19.2.40 Timer Interrupt Skipping Set Register 2 (TITCR2A) 390
19.2.41 Timer Interrupt Skipping Counter 2 (TITCNT2A) 391
19.2.42 A/D Conversion Start Request Select Register 0 (TADSTRGR0) 392
19.3 Operation 393
19.3.1 Basic Functions 393
19.3.2 Synchronous Operation 399
19.3.3 Buffer Operation 401
19.3.4 Cascaded Operation 405
19.3.5 PWM Modes 410
19.3.6 Phase Counting Mode 414
19.3.6.1 16-Bit Phase Counting Mode 414
19.3.6.2 Cascade Connection 32-Bit Phase Counting Mode 425
19.3.7 Reset-Synchronized PWM Mode 426
19.3.8 Complementary PWM Mode 429
19.3.9 A/D Conversion Start Request Delaying Function 464
19.3.10 Synchronous Operation of MTU0 to MTU4 470
19.3.11 External Pulse Width Measurement 471
19.3.12 Dead Time Compensation 472
19.3.13 TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode 474
19.3.14 Noise Filter Function 475
19.3.15 A/D Conversion Start Request Frame Synchronization Signal 475
19.4 Interrupt Sources 476
19.4.1 Interrupt Sources and Priorities 476
19.4.2 DTC Trigger Sources 477
19.4.3 A/D Converter Trigger Sources 478
19.5 Operation Timing 480
19.5.1 Input/Output Timing 480
19.5.2 Interrupt Signal Timing 486
19.6 Usage Notes 489
19.6.1 Module Stop Function Setting 489
19.6.2 Count Clock Restrictions 489
19.6.3 Note on Period Setting 489
19.6.4 Contention between TCNT Write and Clear Operations 490
19.6.5 Contention between TCNT Write and Increment Operations 490
19.6.6 Contention between TGR Write Operation and Compare Match 491
19.6.7 Contention between Buffer Register Write Operation and Compare Match 491
19.6.8 Contention between Buffer Register Write and TCNT Clear Operations 492
19.6.9 Contention between TGR Read Operation and Input Capture 492

19.6.10 Contention between TGR Write Operation and Input Capture 493
19.6.11 Contention between Buffer Register Write Operation and Input Capture 494
19.6.12 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation 495
19.6.13 Counter Value When Count Operation is Stopped in Complementary PWM Mode 496
19.6.14 Buffer Operation Setting in Complementary PWM Mode 496
19.6.15 Buffer Operation and Compare Match in Reset-Synchronized PWM Mode 497
19.6.16 Overflow in Reset-Synchronized PWM Mode 498
19.6.17 Contention between Overflow/Underflow and Counter Clearing 499
19.6.18 Contention between TCNT Write Operation and Overflow/Underflow 499
19.6.19 Note on Transition from Normal Mode or PWM Mode 1 to Reset-Synchronized PWM Mode 500
19.6.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode 500
19.6.21 Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection 500
19.6.22 Interrupt Skipping Function 2 501
19.6.23 Notes When Complementary PWM Mode Output Protection Function is Not Used 501
19.6.24 Notes Regarding Timer Counter (MTU5.TCNT) and Timer General Register (MTU5.TGR) 501
19.6.25 Notes to Prevent Malfunctions in Synchronous Clearing for Complementary PWM Mode 502
19.6.26 Continuous Output of Interrupt Signal in Response to a Compare Match 504
19.6.27 Usage Notes on A/D Conversion Start Request Delaying Function in Complementary PWM Mode 504
19.7 MTU Output Pin Initialization 506
19.7.1 Operating Modes 506
19.7.2 Operation in Case of Re-Setting Due to Error during Operation 506
19.7.3 Overview of Initialization Procedures and Mode Transitions in Case of Error during Operation 507
20. Port Output Enable 3 (POE3C) 533
20.1 Overview 533
20.2 Register Descriptions 536
20.2.1 Input Level Control/Status Register 1 (ICSR1) 536
20.2.2 Input Level Control/Status Register 3 (ICSR3) 537
20.2.3 Input Level Control/Status Register 4 (ICSR4) 538
20.2.4 Input Level Control/Status Register 6 (ICSR6) 539
20.2.5 Output Level Control/Status Register 1 (OCSR1) 540
20.2.6 Active Level Setting Register 1 (ALR1) 541
20.2.7 Software Port Output Enable Register (SPOER) 543
20.2.8 Port Output Enable Control Register 1 (POECR1) 544
20.2.9 Port Output Enable Control Register 2 (POECR2) 546
20.2.10 Port Output Enable Control Register 4 (POECR4) 547
20.2.11 Port Output Enable Control Register 5 (POECR5) 548
20.2.12 Port Output Enable Comparator Output Detection Flag Register (POECMPFR) 549

20.2.13 Port Output Enable Comparator Request Select Register (POECMPSEL) 550
20.3 Operation 551
20.3.1 Input-Level Detection Operation 557
20.3.2 Output-Level Compare Operation 558
20.3.3 High-Impedance Control Using Registers 559
20.3.4 High-Impedance Control through Detection of Oscillation Stop 559
20.3.5 High-Impedance Control through Detection of the Comparator Output 559
20.3.6 Additional Functions for High-Impedance Control 559
20.3.7 Recover from High-Impedance State 559
20.4 POE Setting Procedure 561
20.5 Interrupts 561
20.6 Usage Notes 562
20.6.1 Transition to Low Power Consumption Mode 562
20.6.2 High-Impedance Control When the MTU is Not Selected 562
20.6.3 When the POE is Not Used 562
21. Compare Match Timer (CMT) 563
21.1 Overview 563
21.2 Register Descriptions 564
21.2.1 Compare Match Timer Start Register 0 (CMSTR0) 564
21.2.2 Compare Match Timer Control Register (CMCR) 564
21.2.3 Compare Match Counter (CMCNT) 565
21.2.4 Compare Match Constant Register (CMCOR) 565
21.3 Operation 566
21.3.1 Periodic Count Operation 566
21.3.2 CMCNT Count Timing 566
21.4 Interrupts 567
21.4.1 Interrupt Sources 567
21.4.2 Timing of Compare Match Interrupt Generation 567
21.5 Usage Notes 568
21.5.1 Setting the Module Stop Function 568
21.5.2 Conflict between CMCNT Counter Writing and Compare Match 568
21.5.3 Conflict between CMCNT Counter Writing and Incrementing 568
22. Independent Watchdog Timer (IWDTa) 569
22.1 Overview 569
22.2 Register Descriptions 571
22.2.1 IWDT Refresh Register (IWDTRR) 571
22.2.2 IWDT Control Register (IWDTCR) 572
22.2.3 IWDT Status Register (IWDTSR) 575
22.2.4 IWDT Reset Control Register (IWDTRCR) 576
22.2.5 IWDT Count Stop Control Register (IWDTCSTPR) 577
22.2.6 Option Function Select Register 0 (OFS0) 577

22.3 Operation 578
22.3.1 Count Operation in Each Start Mode 578
22.3.1.1 Register Start Mode 578
22.3.1.2 Auto-Start Mode 580
22.3.2 Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers 582
22.3.3 Refresh Operation 583
22.3.4 Status Flags 585
22.3.5 Reset Output 585
22.3.6 Interrupt Sources 585
22.3.7 Reading the Counter Value 586
22.3.8 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers 587
22.4 Usage Notes 587
22.4.1 Refresh Operations 587
22.4.2 Clock Divide Ratio Setting 587
23. Serial Communications Interface (SCIg, SCIh) 588
23.1 Overview 588
23.2 Register Descriptions 594
23.2.1 Receive Shift Register (RSR) 594
23.2.2 Receive Data Register (RDR) 594
23.2.3 Receive Data Register H, L, HL (RDRH, RDRL, RDRHL) 595
23.2.4 Transmit Data Register (TDR) 596
23.2.5 Transmit Data Register H, L, HL (TDRH, TDRL, TDRHL) 597
23.2.6 Transmit Shift Register (TSR) 597
23.2.7 Serial Mode Register (SMR) 598
23.2.8 Serial Control Register (SCR) 602
23.2.9 Serial Status Register (SSR) 606
23.2.10 Smart Card Mode Register (SCMR) 611
23.2.11 Bit Rate Register (BRR) 613
23.2.12 Modulation Duty Register (MDDR) 621
23.2.13 Serial Extended Mode Register (SEMR) 622
23.2.14 Noise Filter Setting Register (SNFR) 625
23.2.15 I^2C Mode Register 1 (SIMR1) 626
23.2.16 I^2C Mode Register 2 (SIMR2) 627
23.2.17 I^2C Mode Register 3 (SIMR3) 628
23.2.18 I^2C Status Register (SISR) 630
23.2.19 SPI Mode Register (SPMR) 631
23.2.20 Extended Serial Module Enable Register (ESMER) 632
23.2.21 Control Register 0 (CR0) 633
23.2.22 Control Register 1 (CR1) 633
23.2.23 Control Register 2 (CR2) 634
23.2.24 Control Register 3 (CR3) 635

23.2.25 Port Control Register (PCR) 635
23.2.26 Interrupt Control Register (ICR) 636
23.2.27 Status Register (STR) 637
23.2.28 Status Clear Register (STCR) 638
23.2.29 Control Field 0 Data Register (CF0DR) 638
23.2.30 Control Field 0 Compare Enable Register (CF0CR) 639
23.2.31 Control Field 0 Receive Data Register (CF0RR) 639
23.2.32 Primary Control Field 1 Data Register (PCF1DR) 639
23.2.33 Secondary Control Field 1 Data Register (SCF1DR) 640
23.2.34 Control Field 1 Compare Enable Register (CF1CR) 640
23.2.35 Control Field 1 Receive Data Register (CF1RR) 640
23.2.36 Timer Control Register (TCR) 641
23.2.37 Timer Mode Register (TMR) 641
23.2.38 Timer Prescaler Register (TPRE) 642
23.2.39 Timer Count Register (TCNT) 642
23.3 Operation in Asynchronous Mode 643
23.3.1 Serial Data Transfer Format 643
23.3.2 Receive Data Sampling Timing and Reception Margin in Asynchronous Mode 645
23.3.3 Clock 646
23.3.4 Double-Speed Mode 646
23.3.5 CTS and RTS Functions 647
23.3.6 SCI Initialization (Asynchronous Mode) 648
23.3.7 Serial Data Transmission (Asynchronous Mode) 650
23.3.8 Serial Data Reception (Asynchronous Mode) 654
23.4 Multi-Processor Communications Function 658
23.4.1 Multi-Processor Serial Data Transmission 659
23.4.2 Multi-Processor Serial Data Reception 660
23.5 Operation in Clock Synchronous Mode 663
23.5.1 Clock 663
23.5.2 CTS and RTS Functions 664
23.5.3 SCI Initialization (Clock Synchronous Mode) 665
23.5.4 Serial Data Transmission (Clock Synchronous Mode) 666
23.5.5 Serial Data Reception (Clock Synchronous Mode) 670
23.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode) 673
23.6 Operation in Smart Card Interface Mode 674
23.6.1 Sample Connection 674
23.6.2 Data Format (Except in Block Transfer Mode) 675
23.6.3 Block Transfer Mode 676
23.6.4 Receive Data Sampling Timing and Reception Margin 677
23.6.5 SCI Initialization (Smart Card Interface Mode) 678
23.6.6 Serial Data Transmission (Except in Block Transfer Mode) 680

23.6.7 Serial Data Reception (Except in Block Transfer Mode) 683
23.6.8 Clock Output Control 685
23.7 Operation in Simple I2C Mode 686
23.7.1 Generation of Start, Restart, and Stop Conditions 687
23.7.2 Clock Synchronization 689
23.7.3 SSDA Output Delay 690
23.7.4 SCI Initialization (Simple I2C Mode) 691
23.7.5 Operation in Master Transmission (Simple I2C Mode) 692
23.7.6 Master Reception (Simple I2C Mode) 694
23.7.7 Recovery from Bus Hang-up 696
23.8 Operation in Simple SPI Mode 697
23.8.1 States of Pins in Master and Slave Modes 698
23.8.2 SS Function in Master Mode 698
23.8.3 SS Function in Slave Mode 698
23.8.4 Relationship between Clock and Transmit/Receive Data 699
23.8.5 SCI Initialization (Simple SPI Mode) 700
23.8.6 Transmission and Reception of Serial Data (Simple SPI Mode) 700
23.9 Bit Rate Modulation Function 700
23.10 Extended Serial Mode Control Section: Description of Operation 701
23.10.1 Serial Transfer Protocol 701
23.10.2 Transmitting a Start Frame 702
23.10.3 Receiving a Start Frame 705
23.10.3.1 Priority Interrupt Bit 710
23.10.4 Detection of Bus Collisions 711
23.10.5 Digital Filter for Input on the RXDX12 Pin 712
23.10.6 Bit Rate Measurement 713
23.10.7 Selectable Timing for Sampling Data Received through RXDX12 714
23.10.8 Timer 715
23.11 Noise Cancellation Function 717
23.12 Interrupt Sources 718
23.12.1 Buffer Operations for TXI and RXI Interrupts 718
23.12.2 Interrupts in Asynchronous Mode, Clock Synchronous Mode, and Simple SPI Mode 718
23.12.3 Interrupts in Smart Card Interface Mode 719
23.12.4 Interrupts in Simple I2C Mode 720
23.12.5 Interrupt Requests from the Extended Serial Mode Control Section 721
23.13 Usage Notes 722
23.13.1 Setting the Module Stop Function 722
23.13.2 Break Detection and Processing 722
23.13.3 Mark State and Sending Breaks 722
23.13.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode) 722

23.13.5 Writing Data to the TDR Register ... 722
23.13.6 Restrictions on Clock Synchronous Transmission (Clock Synchronous Mode and Simple SPI Mode) ... 723
23.13.7 Restrictions on Using DTC ... 724
23.13.8 Notes on Starting Transfer ... 724
23.13.9 SCI Operations during Low Power Consumption State ... 724
23.13.10 External Clock Input in Clock Synchronous Mode and Simple SPI Mode ... 727
23.13.11 Limitations on Simple SPI Mode ... 728
23.13.12 Limitation 1 on Usage of the Extended Serial Mode Control Section ... 729
23.13.13 Limitation 2 on Usage of the Extended Serial Mode Control Section ... 729
23.13.14 Note on Transmit Enable Bit (TE Bit) ... 730
23.13.15 Note on Stopping Reception When Using the RTS Function in Asynchronous Mode ... 730
24. I²C-bus Interface (RIICa) ... 731
24.1 Overview ... 731
24.2 Register Descriptions ... 734
24.2.1 I²C-bus Control Register 1 (ICCR1) ... 734
24.2.2 I²C-bus Control Register 2 (ICCR2) ... 736
24.2.3 I²C-bus Mode Register 1 (ICMR1) ... 740
24.2.4 I²C-bus Mode Register 2 (ICMR2) ... 741
24.2.5 I²C-bus Mode Register 3 (ICMR3) ... 743
24.2.6 I²C-bus Function Enable Register (ICFER) ... 745
24.2.7 I²C-bus Status Enable Register (ICSER) ... 747
24.2.8 I²C-bus Interrupt Enable Register (ICIER) ... 749
24.2.9 I²C-bus Status Register 1 (ICSR1) ... 751
24.2.10 I²C-bus Status Register 2 (ICSR2) ... 754
24.2.11 Slave Address Register Ly (SARLy) (y = 0 to 2) ... 757
24.2.12 Slave Address Register Uy (SARUy) (y = 0 to 2) ... 758
24.2.13 I²C-bus Bit Rate Low-Level Register (ICBRL) ... 759
24.2.14 I²C-bus Bit Rate High-Level Register (ICBRH) ... 760
24.2.15 I²C-bus Transmit Data Register (ICDRT) ... 762
24.2.16 I²C-bus Receive Data Register (ICDRR) ... 762
24.2.17 I²C-bus Shift Register (ICDRS) ... 762
24.3 Operation ... 763
24.3.1 Communication Data Format ... 763
24.3.2 Initial Settings ... 764
24.3.3 Master Transmit Operation ... 765
24.3.4 Master Receive Operation ... 768
24.3.5 Slave Transmit Operation ... 774
24.3.6 Slave Receive Operation ... 777
24.4 SCL Synchronization Circuit ... 779
24.5 SDA Output Delay Function ... 780

24.6 Digital Noise Filters 781
24.7 Address Match Detection 782
24.7.1 Slave-Address Match Detection 782
24.7.2 Detection of the General Call Address 784
24.7.3 Device-ID Address Detection 785
24.7.4 Host Address Detection 787
24.8 Automatic Low-Hold Function for SCL 788
24.8.1 Function to Prevent Wrong Transmission of Transmit Data 788
24.8.2 NACK Reception Transfer Suspension Function 789
24.8.3 Function to Prevent Failure to Receive Data 790
24.9 Arbitration-Lost Detection Functions 792
24.9.1 Master Arbitration-Lost Detection (MALE Bit) 792
24.9.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit) 794
24.9.3 Slave Arbitration-Lost Detection (SALE Bit) 795
24.10 Start Condition/Restart Condition/Stop Condition Generating Function 796
24.10.1 Generating a Start Condition 796
24.10.2 Generating a Restart Condition 796
24.10.3 Generating a Stop Condition 797
24.11 Bus Hanging 798
24.11.1 Timeout Function 798
24.11.2 Additional SCL Output Function 800
24.11.3 RIIC Reset and Internal Reset 801
24.12 SMBus Operation 802
24.12.1 SMBus Timeout Measurement 802
24.12.2 Packet Error Code (PEC) 803
24.12.3 SMBus Host Notification Protocol (Notify ARP Master Command) 803
24.13 Interrupt Sources 804
24.13.1 Buffer Operation for TXI and RXI Interrupts 804
24.14 Initialization of Registers and Functions When a Reset is Applied or a Condition is Detected 805
24.15 Usage Notes 806
24.15.1 Setting Module Stop Function 806
24.15.2 Notes on Starting Transfer 806
25. CRC Calculator (CRC) 807
25.1 Overview 807
25.2 Register Descriptions 808
25.2.1 CRC Control Register (CRCCR) 808
25.2.2 CRC Data Input Register (CRCDIR) 808
25.2.3 CRC Data Output Register (CRCDOR) 809
25.3 Operation 810
25.4 Usage Notes 813
25.4.1 Module Stop Function Setting 813

25.4.2 Note on Transmission ... 813

26. 12-Bit A/D Converter (S12ADF) ... 814

26.1 Overview ... 814

26.2 Register Descriptions ... 819

26.2.1 A/D Data Registers y (ADDRy) (y = 0 to 7), A/D Data Duplication Register (ADDBLDR), A/D Data Duplication Register A (ADDBLDRA), A/D Data Duplication Register B (ADDBLDRB), A/D Internal Reference Voltage Data Register (ADOCDR) ... 819

26.2.2 A/D Self-Diagnosis Data Register (ADRD) ... 821

26.2.3 A/D Control Register (ADCSR) ... 822

26.2.4 A/D Channel Select Register A0 (ADANSA0) ... 826

26.2.5 A/D Channel Select Register B0 (ADANSB0) ... 827

26.2.6 A/D Channel Select Register C0 (ADANSC0) ... 828

26.2.7 A/D-Converted Value Addition/Average Function Channel Select Register 0 (ADADS0) ... 829

26.2.8 A/D-Converted Value Addition/Average Count Select Register (ADADC) ... 830

26.2.9 A/D Control Extended Register (ADCER) ... 831

26.2.10 A/D Conversion Start Trigger Select Register (ADSTRGR) ... 833

26.2.11 A/D Conversion Extended Input Control Register (ADEXICR) ... 835

26.2.12 A/D Group C Trigger Select Register (ADGCTRGR) ... 836

26.2.13 A/D Sampling State Register n (ADSSTRn) (n = 0 to 7, O) ... 838

26.2.14 A/D Sample-and-Hold Circuit Control Register (ADSHCR) ... 839

26.2.15 A/D Disconnection Detection Control Register (ADDISCR) ... 840

26.2.16 A/D Group Scan Priority Control Register (ADGSPCR) ... 841

26.2.17 A/D Programmable Gain Amplifier Control Register (ADPGACR) ... 843

26.2.18 A/D Programmable Gain Amplifier Gain Setting Register 0 (ADPGAGS0) ... 844

26.3 Operation ... 845

26.3.1 Scanning Operation ... 845

26.3.2 Single Scan Mode ... 846

26.3.2.1 Basic Operation (Without Channel-Dedicated Sample-and-Hold Circuits) ... 846

26.3.2.2 Basic Operation (With Channel-Dedicated Sample-and-Hold Circuits) ... 847

26.3.2.3 Channel Selection and Self-Diagnosis (Without Channel-Dedicated Sample-and-Hold Circuits) ... 848

26.3.2.4 Channel Selection and Self-Diagnosis (With Channel-Dedicated Sample-and-Hold Circuits) ... 849

26.3.2.5 A/D Conversion of Internal Reference Voltage ... 850

26.3.2.6 A/D Conversion in Double Trigger Mode ... 851

26.3.2.7 A/D Conversion in Extended Double Trigger Mode ... 852

26.3.3 Continuous Scan Mode ... 854

26.3.3.1 Basic Operation (Without Channel-Dedicated Sample and-Hold Circuits) ... 854

26.3.3.2 Basic Operation (With Channel-Dedicated Sample-and-Hold Circuits) ... 855

26.3.3.3 Channel Selection and Self-Diagnosis (Without Channel-Dedicated Sample-and-Hold Circuits) ... 856

26.3.3.4 Channel Selection and Self-Diagnosis (With Channel-Dedicated Sample-and-Hold Circuits) 857
26.3.4 Group Scan Mode 858
26.3.4.1 Basic Operation 858
26.3.4.2 A/D Conversion in Double Trigger Mode 859
26.3.4.3 Operation under Group Priority Control 861
26.3.5 Analog Input Sampling Time and Scan Conversion Time 879
26.3.6 Usage Example of A/D Data Register Automatic Clearing Function 882
26.3.7 A/D-Converted Value Addition/Average Mode 882
26.3.8 Disconnection Detection Assist Function 882
26.3.9 Starting A/D Conversion with Asynchronous Trigger 884
26.3.10 Starting A/D Conversion with Synchronous Trigger from Peripheral Module 884
26.3.11 Programmable Gain Amplifier 884
26.4 Interrupt Sources and DTC Transfer Requests 885
26.4.1 Interrupt Requests 885
26.5 Allowable Impedance of Signal Source 885
26.6 Usage Notes 886
26.6.1 Notes on Reading Data Registers 886
26.6.2 Notes on Stopping A/D Conversion 886
26.6.3 A/D Conversion Restarting Timing and Termination Timing 887
26.6.4 Notes on Scan End Interrupt Handling 887
26.6.5 Module Stop Function Setting 887
26.6.6 Notes on Entering Low Power Consumption States 887
26.6.7 Notes on Canceling Software Standby Mode 887
26.6.8 Error in Absolute Accuracy When Disconnection Detection Assistance is in Use 887
26.6.9 Voltage Range of Analog Power Supply Pins 888
26.6.10 Notes on Board Design 888
26.6.11 Notes on Noise Prevention 889
27. D/A Converter for Generating Comparator C Reference Voltage (DA) 890
27.1 Overview 890
27.2 Register Descriptions 891
27.2.1 D/A Data Register 0 (DADR0) 891
27.2.2 D/A Control Register (DACR) 892
27.2.3 Data Register Format Select Register (DADPR) 892
27.3 Operation 893
27.4 Usage Notes 894
27.4.1 Module Stop Function Setting 894
27.4.2 Operation of the D/A Converter in Module Stop State 894
27.4.3 Operation of the D/A Converter in Software Standby Mode 894
27.4.4 Setting the D/A Converter 894

28. Comparator C (CMPC) 895
28.1 Overview 895
28.2 Register Descriptions 898
28.2.1 Comparator Control Register (CMPCTL) 898
28.2.2 Comparator Input Select Register (CMPSEL0) 899
28.2.3 Comparator Reference Voltage Select Register (CMPSEL1) 900
28.2.4 Comparator Output Monitor Register (CMPMON) 901
28.2.5 Comparator External Output Enable Register (CMPIOC) 901
28.3 Operation 902
28.3.1 Comparator Operation Example 902
28.3.2 Noise Filter 903
28.3.3 Interrupts 904
28.3.4 Comparator Pin Output 904
28.3.5 Comparator Setting Flowchart 905
28.4 Usage Notes 906
28.4.1 Module Stop Function Setting 906
28.4.2 Comparator C Operation in Module Stop State 906
28.4.3 Comparator C Operation in Software Standby Mode 906
28.4.4 Comparator Operation while the 12-Bit A/D Convertor is in the Module-Stop State 906
28.4.5 Setting the D/A Converter for Generating Reference Voltage 906
29. Data Operation Circuit (DOC) 907
29.1 Overview 907
29.2 Register Descriptions 908
29.2.1 DOC Control Register (DOCR) 908
29.2.2 DOC Data Input Register (DODIR) 909
29.2.3 DOC Data Setting Register (DODSR) 909
29.3 Operation 910
29.3.1 Data Comparison Mode 910
29.3.2 Data Addition Mode 911
29.3.3 Data Subtraction Mode 912
29.4 Interrupt Requests 912
29.5 Usage Note 912
29.5.1 Module Stop Function Setting 912
30. RAM 913
30.1 Overview 913
30.2 Operation 913
30.2.1 Low Power Consumption Function 913
31. Flash Memory (FLASH) 914
31.1 Overview 914
31.2 ROM Area and Block Configuration 915
31.3 E2 DataFlash Area and Block Configuration 916

31.4 Register Descriptions 917
31.4.1 E2 DataFlash Control Register (DFLCTL) 917
31.4.2 Flash P/E Mode Entry Register (FENTRYR) 918
31.4.3 Protection Unlock Register (FPR) 919
31.4.4 Protection Unlock Status Register (FPSR) 919
31.4.5 Flash P/E Mode Control Register (FPMCR) 920
31.4.6 Flash Initial Setting Register (FISR) 921
31.4.7 Flash Reset Register (FRESETR) 923
31.4.8 Flash Area Select Register (FASR) 923
31.4.9 Flash Control Register (FCR) 924
31.4.10 Flash Extra Area Control Register (FEXCR) 926
31.4.11 Flash Processing Start Address Register H (FSARH) 927
31.4.12 Flash Processing Start Address Register L (FSARL) 927
31.4.13 Flash Processing End Address Register H (FEARH) 928
31.4.14 Flash Processing End Address Register L (FEARL) 928
31.4.15 Flash Read Buffer Register H (FRBH) 929
31.4.16 Flash Read Buffer Register L (FRBL) 929
31.4.17 Flash Write Buffer Register H (FWBH) 929
31.4.18 Flash Write Buffer Register L (FWBL) 930
31.4.19 Flash Status Register 0 (FSTATR0) 931
31.4.20 Flash Status Register 1 (FSTATR1) 933
31.4.21 Flash Error Address Monitor Register H (FEAMH) 934
31.4.22 Flash Error Address Monitor Register L (FEAML) 934
31.4.23 Flash Start-Up Setting Monitor Register (FSCMR) 935
31.4.24 Flash Access Window Start Address Monitor Register (FAWSMR) 935
31.4.25 Flash Access Window End Address Monitor Register (FAWEMR) 936
31.4.26 Unique ID Register n (UIDRn) (n = 0 to 31) 936
31.5 Start-Up Program Protection 937
31.6 Area Protection 938
31.7 Programming and Erasure 939
31.7.1 Sequencer Modes 939
31.7.1.1 E2 DataFlash Access Disabled Mode 939
31.7.1.2 Read Mode 940
31.7.1.3 P/E Modes 940
31.7.2 Mode Transitions 940
31.7.2.1 Transition from E2 DataFlash Access Disable Mode to Read Mode 940
31.7.2.2 Transition from Read Mode to P/E Mode 941
31.7.2.3 Transition from P/E Mode to Read Mode 943
31.7.3 Software Commands 945
31.7.4 Software Command Usage 946
31.7.4.1 Program 946

31.7.4.2 Block Erase 948
31.7.4.3 Blank Check 950
31.7.4.4 Start-Up Area Information Program/Access Window Information Program 952
31.7.4.5 Unique ID Read 953
31.7.4.6 Forced Stop of Software Commands 954
31.7.5 Interrupt 954
31.8 Boot Mode 955
31.8.1 Boot Mode (SCI Interface) 956
31.8.1.1 Operating Conditions in Boot Mode (SCI Interface) 956
31.8.1.2 Starting Up in Boot Mode (SCI Interface) 957
31.8.2 Boot Mode (FINE Interface) 958
31.8.2.1 Operating Conditions in Boot Mode (FINE Interface) 958
31.9 Flash Memory Protection 959
31.9.1 ID Code Protection 959
31.9.1.1 Boot Mode ID Code Protection 960
31.9.1.2 On-Chip Debugging Emulator ID Code Protection 961
31.10 Communication Protocol 962
31.10.1 State Transition in Boot Mode (SCI Interface) 962
31.10.2 Command and Response Configuration 963
31.10.3 Response to Undefined Commands 963
31.10.4 Boot Mode Status Inquiry 964
31.10.5 Inquiry Commands 965
31.10.5.1 Supported Device Inquiry 965
31.10.5.2 Data Area Availability Inquiry 966
31.10.5.3 User Area Information Inquiry 966
31.10.5.4 Data Area Information Inquiry 967
31.10.5.5 Block Information Inquiry 967
31.10.6 Setting Commands 968
31.10.6.1 Device Select 968
31.10.6.2 Operating Frequency Select 969
31.10.6.3 Program/Erase Host Command Wait State Transition 970
31.10.7 ID Code Authentication Command 971
31.10.7.1 ID Code Check 971
31.10.8 Program/Erase Commands 972
31.10.8.1 User/Data Area Program Preparation 972
31.10.8.2 Program 973
31.10.8.3 Data Area Program 974
31.10.8.4 Erase Preparation 975
31.10.8.5 Block Erase 975
31.10.9 Read-Check Commands 976
31.10.9.1 Memory Read 976

31.10.9.2 User Area Checksum 977
31.10.9.3 Data Area Checksum 978
31.10.9.4 User Area Blank Check 978
31.10.9.5 Data Area Blank Check 979
31.10.9.6 Access Window Information Program 979
31.10.9.7 Access Window Read 980
31.11 Serial Programmer Operation in Boot Mode (SCI Interface) 981
31.11.1 Bit Rate Automatic Adjustment 982
31.11.2 Receive the MCU Information 983
31.11.3 Select the Device and Change the Bit Rate 984
31.11.4 Enter the Program/Erase Host Command Wait State 985
31.11.5 Unlock Boot Mode ID Code Protection 986
31.11.6 Erase the User Area and Data Area 987
31.11.7 Program the User Area and Data Area 988
31.11.8 Check Data in the User Area 989
31.11.9 Check Data in the Data Area 990
31.11.10 Set the Access Window in the User Area 991
31.12 Rewriting by Self-Programming 992
31.12.1 Overview 992
31.13 Usage Notes 993
31.14 Usage Notes in Boot Mode 994
32. Electrical Characteristics 995
32.1 Absolute Maximum Ratings 995
32.2 Recommended operating conditions 995
32.3 DC Characteristics 996
32.3.1 Normal I/O Pin Output Voltage Characteristics 1003
32.4 AC Characteristics 1005
32.4.1 Clock Timing 1005
32.4.2 Reset Timing 1009
32.4.3 Timing of Recovery from Low Power Consumption Modes 1010
32.4.4 Control Signal Timing 1012
32.4.5 Timing of On-Chip Peripheral Modules 1013
32.4.5.1 I/O ports 1013
32.4.5.2 MTU 1014
32.4.5.3 POE 1015
32.4.5.4 SCI 1018
32.4.5.5 A/D converter 1024
32.4.5.6 CAC 1024
32.4.5.7 RIIC 1025
32.5 A/D Conversion Characteristics 1027
32.6 Programmable Gain Amplifier Characteristics 1031

32.7 Comparator Characteristics 1032
32.8 D/A Conversion Characteristics 1033
32.9 Power-On Reset Circuit and Voltage Detection Circuit Characteristics 1034
32.10 Oscillation Stop Detection Timing 1038
32.11 ROM (Code Flash Memory) Characteristics 1039
32.12 E2 DataFlash (Data Flash Memory) Characteristics 1041
32.13 Usage Notes 1042
32.13.1 Connecting VCL Capacitor and Bypass Capacitors 1042
Appendix 1. Port States in Each Processing Mode 1047
Appendix 2. Package Dimensions 1048
REVISION HISTORY 1052

R01UH0822EJ0110
Rev.1.10
Mar 16, 2021

32-MHz 32-bit RX MCUs, built-in FPU, 50 DMIPS, power supply 5 V
12-bit ADC (equipped with 3-channel synchronous S/H circuits, programmable gain amplifier × 3 ch, and comparator)
32-MHz PWM (three-phase complementary output × 1 ch), On-chip data flash memory

Features

■ 32-bit RX CPU core

- Max. operating frequency: 32 MHz
  Capable of 50 DMIPS in operation at 32 MHz
- Accumulator handles 64-bit results (for a single instruction) from 32-bit × 32-bit operations
- Multiplication and division unit handles 32-bit × 32-bit operations (multiplication instructions take one CPU clock cycle)
- Built-in FPU: 32-bit single-precision floating point (compliant to IEEE754)
- Fast interrupt
- CISC Harvard architecture with 5-stage pipeline
- Variable-length instructions, ultra-compact code
- On-chip debugging circuit

■ Low power design and architecture

- Operation from a single 2.7-V to 5.5-V supply
- Three low power consumption modes

■ On-chip code flash memory, no wait states

- 128-/64-Kbyte capacities
- On-board or off-board user programming
- For instructions and operands

■ On-chip data flash memory

- 4 Kbytes (1,000,000 program/erase cycles (typ.))
- BGO (Background Operation)

■ On-chip SRAM, no wait states

- 12 Kbytes of SRAM

■ DMA

- DTC: Five transfer modes

■ Reset and supply management

- Seven types of reset, including the power-on reset (POR)
- Low voltage detection (LVD) with voltage settings

■ Clock functions

- Main clock oscillator frequency: 1 to 20 MHz
- External clock input frequency: Up to 20 MHz
- PLL circuit input: 4 MHz to 8 MHz
- Low-speed on-chip oscillator: 4 MHz
- High-speed on-chip oscillator: 32 MHz ±1%
- IWDT-dedicated on-chip oscillator: 15 kHz
- On-chip clock frequency accuracy measurement circuit (CAC)

■ Independent watchdog timer

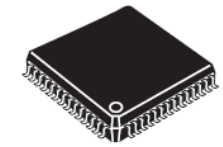
- 15-kHz on-chip oscillator produces a dedicated clock signal to drive IWDT operation.

■ Useful functions for IEC60730 compliance

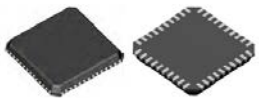
- Self-diagnostic and disconnection-detection assistance functions for the A/D converter, clock frequency accuracy measurement circuit, independent watchdog timer, RAM test assistance functions using the DOC, etc.

■ MPC

- Multiple locations are selectable for I/O pins of peripheral functions



PLQP0048KB-B 7 × 7 mm, 0.5 mm pitch
PLQP0032GB-A 7 × 7 mm, 0.8 mm pitch



PWQN0048KE-A 7 × 7 mm, 0.5 mm pitch
PWQN0032KE-A 5 × 5 mm, 0.5 mm pitch

■ Up to 4 communications channels

- SCI with many useful functions (3 channels)
  Asynchronous mode, clock synchronous mode, smart card interface mode, simplified SPI, simplified I^2C, and extended serial mode.
- I^2C bus interface: Transfer at up to 400 kbps (one channel)

■ Up to 8 extended-function timers

- 16-bit MTU3 (six channels): 32 MHz operation, input capture, output compare, three-phase complementary PWM × 1 channel-output, CPU-efficient complementary PWM, phase counting mode (2 channels)
- 16-bit compare-match timers (2 channels)

■ 12-bit A/D converter: 8 ch

- On-chip sample-and-hold circuit: 12-bit × up to 3 channels
- Sampling time can be set for each channel
- Group scan priority control mode (3 levels)
- Self-diagnostic function and analog input disconnection detection assistance function (compliant to IEC60730)
- Input signal amplitude by the programmable gain amplifier (3 channels)
- ADC: 3-channel simultaneous sample-and-hold circuit (3 shunt method), double data register (1 shunt method), amplifier (3 channels), comparator (3 channels)

■ Register write protection function can protect values in important registers against overwriting.

■ Up to 39 pins for general I/O ports

- 5-V tolerant, open drain, input pull-up

■ Operating temperature range

- –40 to +85°C
- –40 to +105°C

■ Applications

- General industrial and consumer equipment

1. Overview

1.1 Outline of Specifications

Table 1.1 lists the specifications, and Table 1.2 gives a comparison of the functions of the products in different packages.
Table 1.1 is for products with the greatest number of functions, so the number of peripheral modules and channels will differ in accordance with the package type. For details, see Table 1.2, Comparison of Functions for Different Packages.

Table 1.1 Outline of Specifications (1/3)

Classification	Module/Function	Description
CPU	CPU	• Maximum operating frequency: 32 MHz • 32-bit RX CPU • Minimum instruction execution time: One instruction per clock cycle • Address space: 4-Gbyte linear • Register set General purpose: Sixteen 32-bit registers Control: Nine 32-bit registers Accumulator: One 64-bit registers • Basic instructions: 73 Variable-length instruction format • DSP instructions: 9 • Addressing modes: 10 • Data arrangement Instructions: Little endian Data: Selectable as little endian or big endian • On-chip 32-bit multiplier: 32-bit × 32-bit → 64-bit • On-chip divider: 32-bit ÷ 32-bit → 32 bits • Barrel shifter: 32 bits
	FPU	• Single precision (32-bit) floating point • Data types and floating-point exceptions in conformance with the IEEE754 standard
Memory	ROM	• Capacity: 64 K/128 Kbytes • 32 MHz, no-wait memory access • Programming/erasing method: Serial programming (asynchronous serial communication), self-programming
	RAM	• Capacity: 12 Kbytes • 32 MHz, no-wait memory access
	E2 DataFlash	• Capacity: 4 Kbytes • Number of erase/write cycles: 1,000,000 (typ)
MCU operating mode		Single-chip mode
Clock	Clock generation circuit	• Main clock oscillator, low-speed and high-speed on-chip oscillator, PLL frequency synthesizer, and IWDT-dedicated on-chip oscillator • Oscillation stop detection: Available • Clock frequency accuracy measurement circuit (CAC): Available • Independent settings for the system clock (ICLK), peripheral module clock (PCLK), and FlashIF clock (FCLK) The CPU and system sections such as other bus masters run in synchronization with the system clock (ICLK): 32 MHz (at max.) Peripheral modules run in synchronization with the PCLKB: 32 MHz (at max.) The flash peripheral circuit runs in synchronization with the FCLK: 32 MHz (at max.) The ICLK frequency can only be set to FCLK, PCLKB, or PCLKD multiplied by n (n: 1, 2, 4, 8, 16, 32, 64)
Resets		RES# pin reset, power-on reset, voltage monitoring reset, independent watchdog timer reset, and software reset
Voltage detection	Voltage detection circuit (LVDAb)	• When the voltage on VCC falls below the voltage detection level, an internal reset or internal interrupt is generated. Voltage detection circuit 0 is capable of selecting the detection voltage from 3 levels Voltage detection circuit 1 is capable of selecting the detection voltage from 9 levels Voltage detection circuit 2 is capable of selecting the detection voltage from 4 levels
Low power consumption	Low power consumption functions	• Module stop function • Three low power consumption modes Sleep mode, deep sleep mode, and software standby mode
	Function for lower operating power consumption	• Operating power control modes High-speed operating mode and middle-speed operating mode

Table 1.1 Outline of Specifications (2/3)

Classification	Module/Function	Description
Interrupt	Interrupt controller (ICUb)	• Interrupt vectors: 256 • External interrupts: 7 (NMI, IRQ0 to IRQ5 pins) • Non-maskable interrupts: 5 (NMI pin, oscillation stop detection interrupt, voltage monitoring 1 interrupt, voltage monitoring 2 interrupt, and IWDT interrupt) • 16 levels specifiable for the order of priority
DMA	Data transfer controller (DTCb)	• Transfer modes: Normal transfer, repeat transfer, and block transfer • Activation sources: External interrupts and interrupt requests from peripheral functions • Sequence transfer
I/O ports	General I/O ports	48-/32-pin • I/O: 38/22 • Input: 1/1 • Pull-up resistors: 38/22 • Open-drain outputs: 30/18 • 5-V tolerance: 2/2
Multi-function pin controller (MPC)		Capable of selecting the input/output function from multiple pins
Timers	Multi-function timer pulse unit 3 (MTU3c)	• 6 units (16 bis × 6 channels) • Provides up to 16 pulse-input/output lines and three pulse-input lines • Select from among fourteen counter-input clock signals for each channel (PCLK/1, PCLK/2, PCLK/4, PCLK/8, PCLK/16, PCLK/32, PCLK/64, PCLK/256, PCLK/1024, MTCLKA, MTCLKB, MTCLKC, MTCLKD, MTIOC1A) other than channel 1/3/4, for which only eleven signals are available, channel 2 for 12, channel 5 for 10 • 26 output compare/input capture registers • Counter clear operation (with compare match- or input capture-sourced simultaneous counter clear capability) • Simultaneous writing to multiple timer counters (TCNT) • Simultaneous register input/output by synchronous counter operation • Buffer operation • Cascaded operation • 28 interrupt sources • Automatic transfer of register data • Pulse output modes: Toggle/PWM/complementary PWM/reset-synchronized PWM • Complementary PWM output mode 3-phase non-overlapping waveform output for inverter control Automatic dead time setting Adjustable PWM duty cycle: from 0 to 100% A/D conversion request delaying function Interrupt at crest/trough can be skipped Double buffer function • Reset-synchronized PWM mode Outputs three phases each for positive and negative PWM waveforms in user-specified duty cycle • Phase counting modes: 16-bit mode (channel 1 and 2)/32-bit mode (channel 1 and 2) • Dead time compensation counter function • A/D converter start trigger can be generated • A/D converter start triggers can be skipped • Signals from the input capture and external counter clock pins are input via a digital filter
	Port output enable 3 (POE3C)	Controls the high-impedance state of the MTU's waveform output pins
	Compare match timer (CMT)	• (16 bits × 2 channels) × 1 units • Select from among four clock signals (PCLK/8, PCLK/32, PCLK/128, PCLK/512)
	Independent watchdog timer (IWDTa)	• 14 bits × 1 channel • Count clock: Dedicated low-speed on-chip oscillator for the IWDT Frequency divided by 1, 16, 32, 64, 128, or 256

Table 1.1 Outline of Specifications (3/3)

Classification	Module/Function	Description
Communication functions	Serial communications interfaces (SCIg, SCIh)	• 3 channels (channel 1 and 5: SCIg, channel 12: SCIh) • SCIg Serial communications modes: Asynchronous, clock synchronous, and smart-card interface Multi-processor function On-chip baud rate generator allows selection of the desired bit rate Choice of LSB-first or MSB-first transfer Average transfer rate clock can be input from MTU timers Start-bit detection: Level or edge detection is selectable. Simple I^2C Simple SPI 9-bit transfer mode Bit rate modulation • SCIh (The following functions are added to SCIg) Supports the serial communications protocol, which contains the start frame and information frame Supports the LIN format
	I^2C bus interface (RIICa)	• 1 channel • Communications formats: I^2C bus format/SMBus format • Master mode or slave mode selectable • Supports fast mode
12-bit A/D converter (S12ADF)		• 12 bits (8 channels × 1 unit) • 12-bit resolution • Minimum conversion time: 1.4 µs per channel when the ADCLK is operating at 32 MHz • Operating modes Scan mode (single scan mode, continuous scan mode, and 3 group scan mode) Group A priority control (only for 3 group scan mode) • Sampling variable Sampling time can be set up for each channel • Self-diagnostic function • Double trigger mode (A/D conversion data duplicated) • Assist on analog input disconnection detection • A/D conversion start conditions A software trigger, a trigger from a timer (MTU), or an external trigger signal • Sample-and-hold function Sample-and-hold circuit included (3 channels) • Amplification of input signals by a programmable gain amplifier (3 channels) Amplification rate: 2.000 times, 2.500 times, 3.077 times, 5.000 times, 8.000 times, 10.000 times (total of 6 steps)
Comparator C (CMPC)		• 3 channels • Function to compare the reference voltage and the analog input voltage • Reference voltage: Select from among two voltages • Analog input voltage: Select from among four voltages
D/A converter (DA) for generating comparator C reference voltage		• 1 channel • 8-bit resolution • Output voltage: 0 to AVCC0 • Reference voltage generation circuit for comparator C
CRC calculator (CRC)		• CRC code generation for arbitrary amounts of data in 8-bit units • Select any of three generating polynomials: $X^8 + X^2 + X + 1$, $X^{16} + X^{15} + X^2 + 1$, or $X^{16} + X^{12} + X^5 + 1$ • Generation of CRC codes for use with LSB-first or MSB-first communications is selectable.
Data operation circuit (DOC)		Comparison, addition, and subtraction of 16-bit data
Power supply voltages/Operating frequencies		VCC = 2.7 to 5.5V: 32 MHz
Supply current		11 mA at 32 MHz (typ.)
Operating temperature range		D version: –40 to +85°C, G version: –40 to +105°C
Packages		48-pin LFQFP (PLQP0048KB-B) 7 × 7 mm, 0.5 mm pitch 32-pin LQFP (PLQP0032GB-A) 7 × 7 mm, 0.8 mm pitch 48-pin HWQFN (PWQN0048KE-A) 7 × 7 mm, 0.5 mm pitch 32-pin HWQFN (PWQN0032KE-A) 5 × 5 mm, 0.5 mm pitch
Debugging interface		FINE interface

Table 1.2 Comparison of Functions for Different Packages

Module/Functions		RX13T Group	
		48 Pins	32 Pins
Interrupts	External interrupts	NMI, IRQ0 to IRQ5	NMI, IRQ0 to IRQ2, IRQ5
DTC	Data transfer controller	Available	
Timers	Multi-function timer pulse unit 3	6 channels	
	Port output enable 3	POE0#, POE8#, POE10#	POE8#, POE10#
	Compare match timer	2 channels × 1 units	
	Independent watchdog timer	Available	
Communication functions	Serial communications interfaces (SCIg)	2 channels (SCI1, SCI5)	
	Serial communications interfaces (SCIh)	1 channel (SCI12)	
	I^2C bus interface	1 channel	
12-bit A/D converter		8 channels	5 channels
Comparator C		3 channels	
CRC calculator		Available	
Data operation circuit		Available	
Clock frequency accuracy measurement circuit		Available	
Packages		48-pin LFQFP (0.5 mm) 48-pin HWQFN (0.5 mm)	32-pin LQFP (0.8 mm) 32-pin HWQFN (0.5 mm)

1.2 List of Products

Table 1.3 is a list of products, and Figure 1.1 shows how to read the product part no., memory capacity, and package type.

Table 1.3 List of Products

Group	Part No.	Order Part No.	Package	ROM Capacity	RAM Capacity	E2 DataFlash	Operating Frequency (Max.)	Operating Temperature
RX13T (D version)	R5F513T5ADFL	R5F513T5ADFL#30	PLQP0048KB-B	128 Kbytes	12 Kbytes	4 Kbytes	32 MHz	–40 to +85°C
	R5F513T5ADNE	R5F513T5ADNE#20	PWQN0048KE-A					
	R5F513T5ADFJ	R5F513T5ADFJ#30	PLQP0032GB-A					
	R5F513T5ADNH	R5F513T5ADNH#20	PWQN0032KE-A					
	R5F513T3ADFL	R5F513T3ADFL#30	PLQP0048KB-B	64 Kbytes				
	R5F513T3ADNE	R5F513T3ADNE#20	PWQN0048KE-A					
	R5F513T3ADFJ	R5F513T3ADFJ#30	PLQP0032GB-A					
	R5F513T3ADNH	R5F513T3ADNH#20	PWQN0032KE-A					
RX13T (G version)	R5F513T5AGFL	R5F513T5AGFL#30	PLQP0048KB-B	128 Kbytes				–40 to +105°C
	R5F513T5AGNE	R5F513T5AGNE#20	PWQN0048KE-A					
	R5F513T5AGFJ	R5F513T5AGFJ#30	PLQP0032GB-A					
	R5F513T5AGNH	R5F513T5AGNH#20	PWQN0032KE-A					
	R5F513T3AGFL	R5F513T3AGFL#30	PLQP0048KB-B	64 Kbytes				
	R5F513T3AGNE	R5F513T3AGNE#20	PWQN0048KE-A					
	R5F513T3AGFJ	R5F513T3AGFJ#30	PLQP0032GB-A					
	R5F513T3AGNH	R5F513T3AGNH#20	PWQN0032KE-A					

Note: The part numbers for orders above are used for products in mass production or under development when this manual is issued. Refer to the Renesas Electronics Corporation website for the latest part numbers.

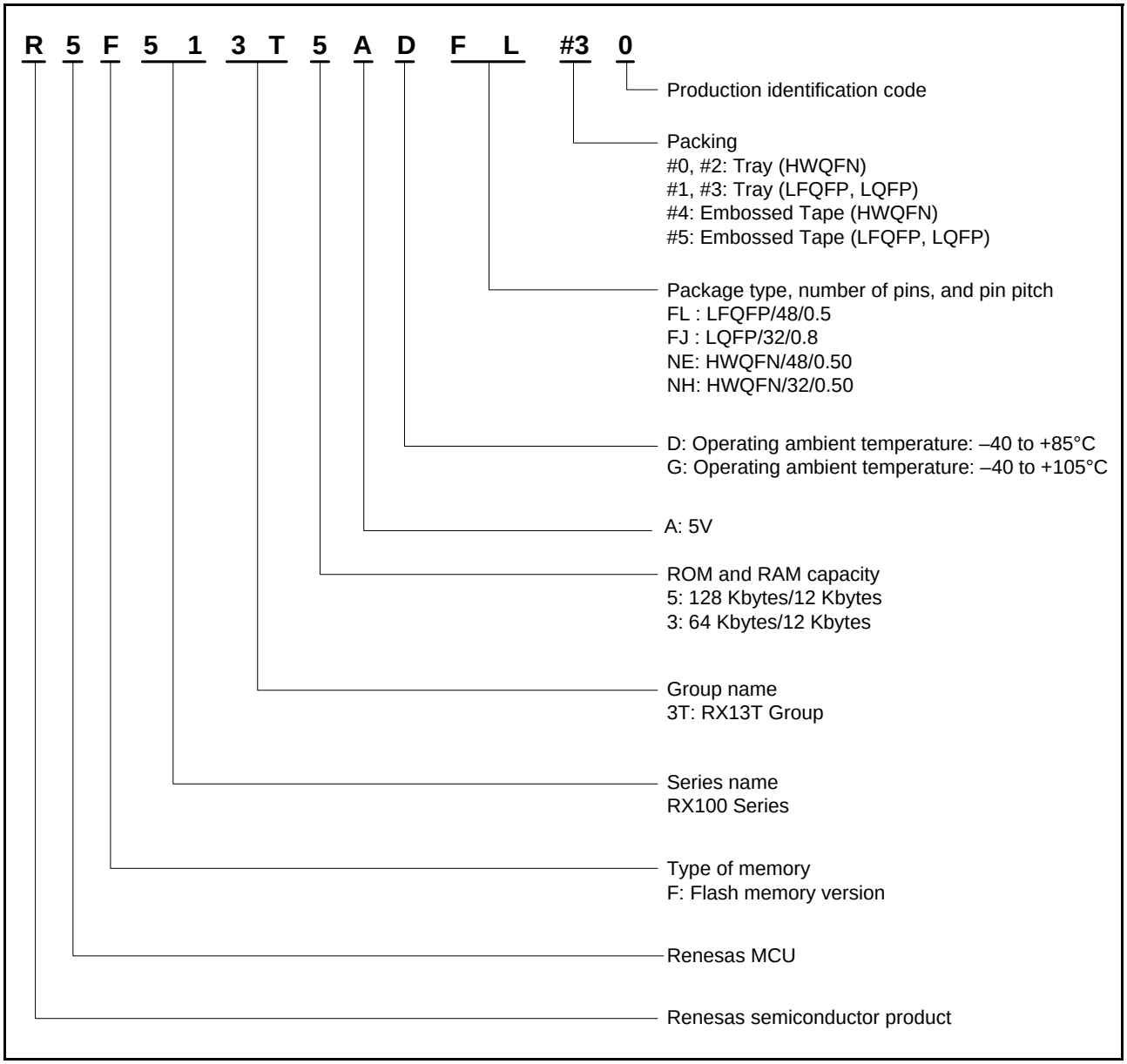


Figure 1.1 How to Read the Product Part Number

1.3 Block Diagram

Figure 1.2 shows a block diagram.

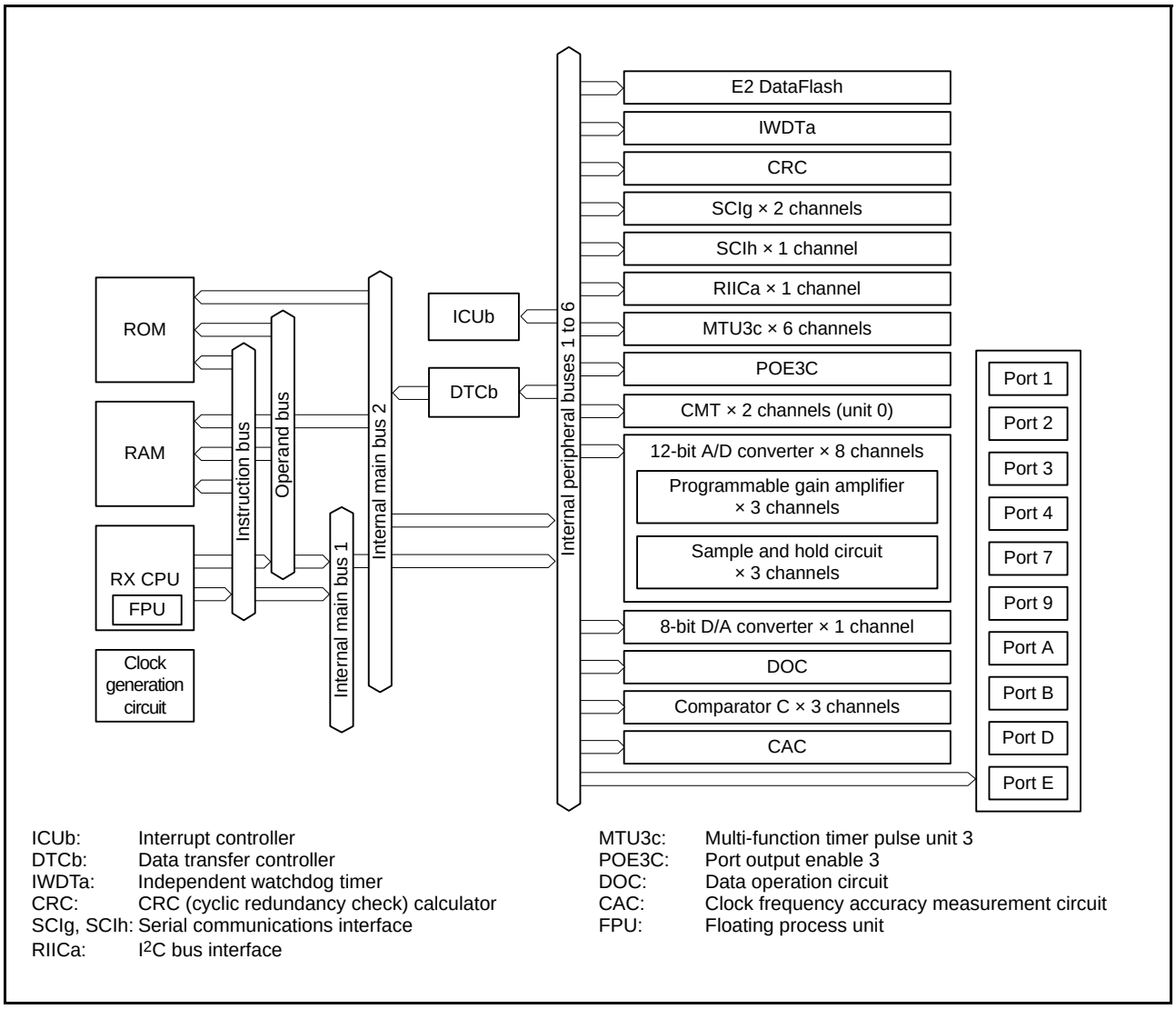


Figure 1.2 Block Diagram

1.4 Pin Functions

Table 1.4 lists the pin functions.

Table 1.4 Pin Functions (1/2)

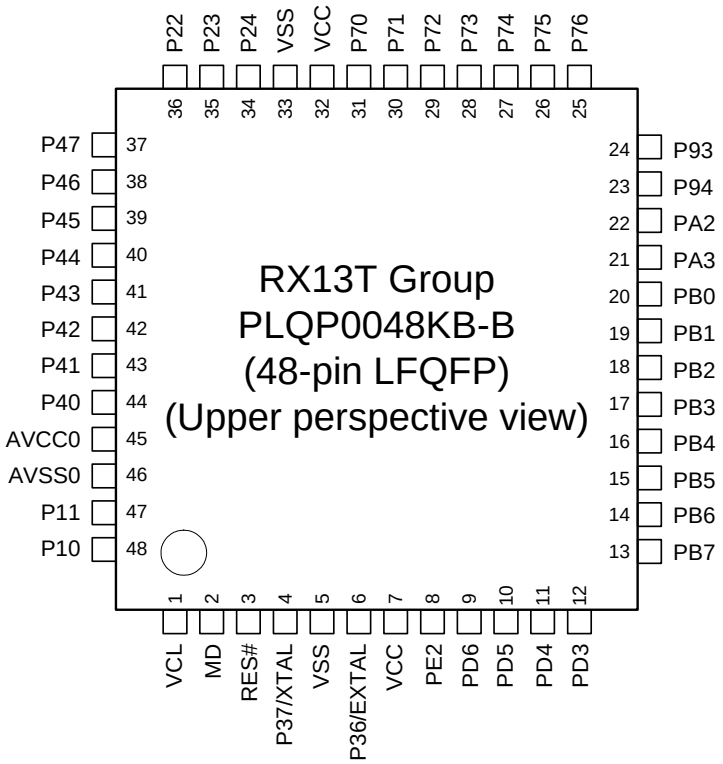
Classifications	Pin Name	I/O	Description
Power supply	VCC	Input	Power supply pin. Connect it to the system power supply.
	VCL	—	Connect this pin to the VSS pin via the 4.7 µF smoothing capacitor used to stabilize the internal power supply. Place the capacitor close to the pin.
	VSS	Input	Ground pin. Connect it to the system power supply (0 V).
Clock	XTAL	Output	Pins for connecting a crystal. An external clock can be input through the EXTAL pin.
	EXTAL	Input	
Operating mode control	MD	Input	Pin for setting the operating mode. The signal levels on this pin must not be changed during operation.
System control	RES#	Input	Reset pin. This MCU enters the reset state when this signal goes low.
CAC	CACREF	Input	Input pin for the clock frequency accuracy measurement circuit.
On-chip emulator	FINED	I/O	FINE interface pin.
Interrupts	NMI	Input	Non-maskable interrupt request pin.
	IRQ0 to IRQ5	Input	Interrupt request pins.
Multi-function timer pulse unit 3	MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D	I/O	The TGRA0 to TGRD0 input capture input/output compare output/PWM output pins.
	MTIOC1A, MTIOC1B	I/O	The TGRA1 and TGRB1 input capture input/output compare output/PWM output pins.
	MTIOC2A, MTIOC2B	I/O	The TGRA2 and TGRB2 input capture input/output compare output/PWM output pins.
	MTIOC3A, MTIOC3B, MTIOC3C, MTIOC3D	I/O	The TGRA3 to TGRD3 input capture input/output compare output/PWM output pins.
	MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D	I/O	The TGRA4 to TGRD4 input capture input/output compare output/PWM output pins.
	MTIC5U, MTIC5V, MTIC5W	Input	The TGRU5, TGRV5, and TGRW5 input capture input/external pulse input pins.
	MTCLKA, MTCLKB, MTCLKC, MTCLKD	Input	Input pins for the external clock.
	ADSM0	Output	A/D trigger output pin.
Port output enable 3	POE0#, POE8#, POE10#	Input	Input pins for request signals to place the MTU pins in the high impedance state.
Serial communications interface (SCIg)	• Asynchronous mode/clock synchronous mode		
	SCK1, SCK5	I/O	Input/output pins for the clock.
	RXD1, RXD5	Input	Input pins for received data.
	TXD1, TXD5	Output	Output pins for transmitted data.
	CTS1#, CTS5#	Input	Input pins for controlling the start of transmission and reception.
	RTS1#, RTS5#	Output	Output pins for controlling the start of transmission and reception.
	• Simple I^2C mode		
	SSCL1, SSCL5	I/O	Input/output pins for the I^2C clock.
	SSDA1, SSDA5	I/O	Input/output pins for the I^2C data.
	• Simple SPI mode		
	SCK1, SCK5	I/O	Input/output pins for the clock.
	SMISO1, SMISO5	I/O	Input/output pins for slave transmit data.
	SMOSI1, SMOSI5	I/O	Input/output pins for master transmit data.
	SS1#, SS5#	Input	Chip-select input pins.

Table 1.4 Pin Functions (2/2)

Classifications	Pin Name	I/O	Description
Serial communications interface (SCIh)	• Asynchronous mode/clock synchronous mode		
	SCK12	I/O	Input/output pin for the clock
	RXD12	Input	Input pin for received data
	TXD12	Output	Output pin for transmitted data
	CTS12#	Input	Input pin for controlling the start of transmission and reception
	RTS12#	Output	Output pin for controlling the start of transmission and reception
	• Simple I^2C mode		
	SSCL12	I/O	Input/output pin for the I^2C clock
	SSDA12	I/O	Input/output pin for the I^2C data
	• Simple SPI mode		
	SCK12	I/O	Input/output pin for the clock
	SMISO12	I/O	Input/output pin for slave transmission of data
	SMOSI12	I/O	Input/output pin for master transmission of data
	SS12#	Input	Chip-select input pin
	• Extended serial mode		
	RXDX12	Input	Input pin for SCIf received data
	TXDX12	Output	Output pin for SCIf transmitted data
	SIOX12	I/O	Input/output pin for SCIf received or transmitted data
I^2C bus interface	SCL0	I/O	Input/output pin for I^2C bus interface clocks. Bus can be directly driven by the N-channel open drain output.
	SDA0	I/O	Input/output pin for I^2C bus interface data. Bus can be directly driven by the N-channel open drain output.
12-bit A/D converter	AN000 to AN007	Input	Input pins for the analog signals to be processed by the A/D converter.
	ADTRG0#	Input	Input pin for the external trigger signals that start the A/D conversion.
	ADST0	Output	Output pin for A/D conversion status.
Comparator C	CMPC00, CMPC02, CMPC03	Input	Analog input pin for CMPC0
	CMPC10, CMPC12, CMPC13	Input	Analog input pin for CMPC1
	CMPC20, CMPC22	Input	Analog input pin for CMPC2
	COMP0 to COMP2	Output	Comparator detection result output pins.
	CVREFC0	Input	Analog reference voltage supply pins for comparator C.
Analog power supply	AVCC0	Input	Analog voltage supply pin for the 12-bit A/D converter, comparator C, and the 8-bit D/A converter for generating comparator C reference voltage. Connect this pin to VCC when these modules are not used.
	AVSS0	Input	Analog ground pin for the 12-bit A/D converter, comparator C, and the 8-bit D/A converter for generating comparator C reference voltage. Connect this pin to VSS when these modules are not used.
I/O ports	P10, P11	I/O	2-bit input/output pins.
	P22 to P24	I/O	3-bit input/output pins.
	P36, P37	I/O	2-bit input/output pins.
	P40 to P47	I/O	8-bit input/output pins.
	P70 to P76	I/O	7-bit input/output pins.
	P93, P94	I/O	2-bit input/output pins.
	PA2, PA3	I/O	2-bit input/output pins.
	PB0 to PB7	I/O	8-bit input/output pins.
	PD3 to PD6	I/O	4-bit input/output pins.
	PE2	Input	1-bit input pin.

1.5 Pin Assignments

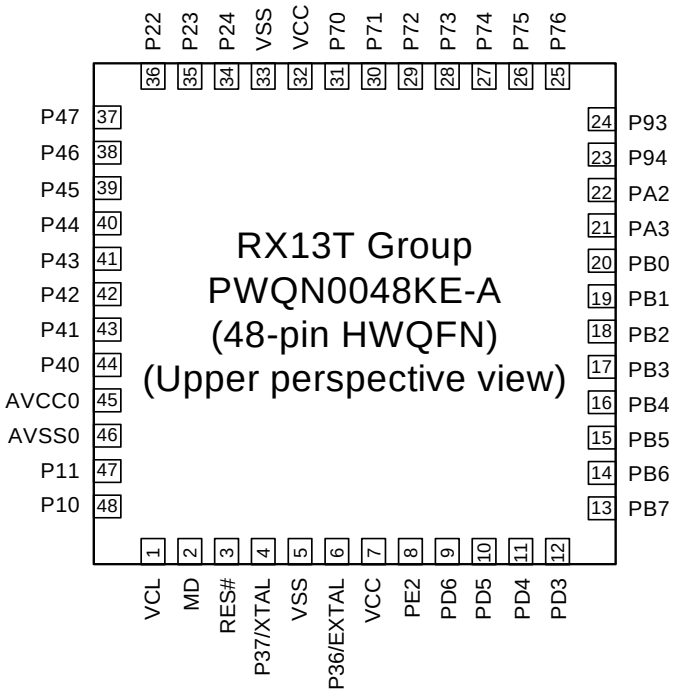
1.5.1 48-Pin LFQFP



Note: This figure indicates the power supply pins and I/O port pins.
For the pin configuration, see the table “List of Pins and Pin Functions (48-Pin LFQFP/HWQFN)”.

Figure 1.3 Pin Assignments of the 48-Pin LFQFP

1.5.2 48-Pin HWQFN

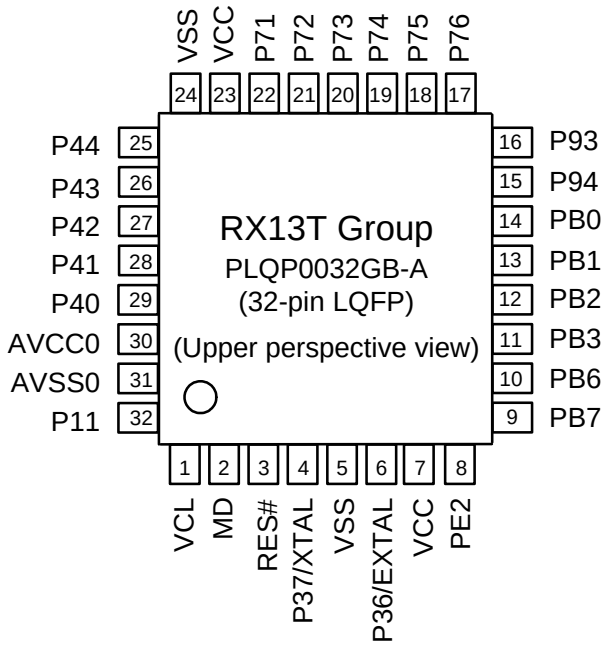


Note: It is recommended to connect an exposed die pad to VSS.

Note: This figure indicates the power supply pins and I/O port pins.
For the pin configuration, see the table “List of Pins and Pin Functions (48-Pin LFQFP/HWQFN)”.

Figure 1.4 Pin Assignments of the 48-Pin HWQFN

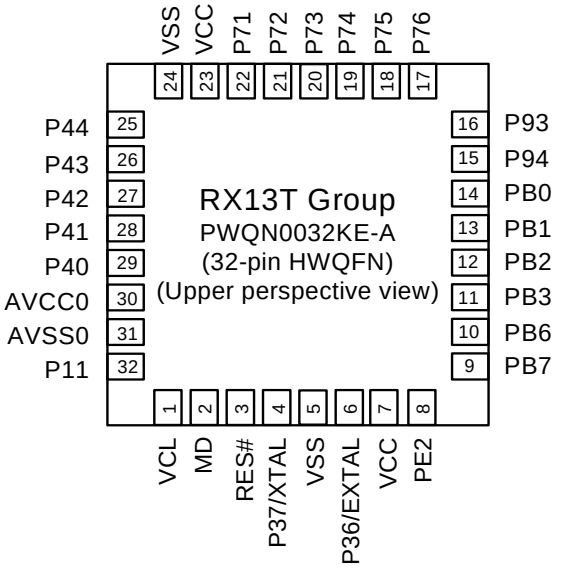
1.5.3 32-Pin LQFP



Note: This figure indicates the power supply pins and I/O port pins.
For the pin configuration, see the table “List of Pins and Pin Functions (32-Pin LQFP/HWQFN)”.

Figure 1.5 Pin Assignments of the 32-Pin LQFP

1.5.4 32-Pin HWQFN



Note: It is recommended to connect an exposed die pad to VSS.

Note: This figure indicates the power supply pins and I/O port pins.
For the pin configuration, see the table “List of Pins and Pin Functions (32-Pin LQFP/HWQFN)”.

Figure 1.6 Pin Assignments of the 32-Pin HWQFN

1.6 List of Pins and Pin Functions

1.6.1 48-Pin LFQFP/HWQFN

Table 1.5 List of Pins and Pin Functions (48-Pin LFQFPLQFP/HWQFN) (1/2)

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, POE, CAC)	Communications (SCI, RIIC)	Others
1	VCL				
2	MD				FINED
3	RES#				
4	XTAL	P37			
5	VSS				
6	EXTAL	P36			
7	VCC				
8		PE2	POE10#		NMI/IRQ0
9		PD6	MTIOC0D	CTS1#/RTS1#/SS1#	IRQ5/ADST0
10		PD5	MTIOC0C	RXD1/SMISO1/SSCL1	IRQ3
11		PD4	MTIOC0B	SCK1	IRQ2
12		PD3	MTIOC0A	TXD1/SMOSI1/SSDA1	
13		PB7	MTIOC3C/MTCLKD	RXD1/SMISO1/SSCL1/RXD5/SMISO5/ SSCL5	IRQ5
14		PB6	MTIOC1B/MTIOC3A	TXD1/SMOSI1/SSDA1/TXD5/SMOSI5/ SSDA5	
15		PB5			ADTRG0#
16		PB4	POE8#		IRQ3
17		PB3	MTIOC0A/CACREF	SCK5/SCK12	
18		PB2	MTIOC0B/MTCLKC/ADSM0	TXD5/SMOSI5/SSDA5/SDA0	
19		PB1	MTIOC0C/MTIC5W/MTCLKA	RXD5/SMISO5/SSCL5/SCL0	IRQ2
20		PB0	MTIOC0D/MTIOC2A/MTCLKB	TXD12/TXDX12/SIOX12/SMOSI12/ SSDA12	
21		PA3	MTIOC1B/MTIOC2A	CTS12#/RTS12#/SS12#	
22		PA2	MTIOC1A/MTIOC2B	CTS5#/RTS5#/SS5#	IRQ4
23		P94	MTIOC2B/MTIC5U/MTCLKA	RXD12/RXDX12/SMISO12/SSCL12	IRQ1
24		P93	MTIOC1A/MTIC5V	SCK5/SCK12	IRQ0/ADTRG0#
25		P76	MTIOC4D		
26		P75	MTIOC4C		
27		P74	MTIOC3D		
28		P73	MTIOC4B		
29		P72	MTIOC4A		
30		P71	MTIOC3B		
31		P70	POE0#		IRQ5
32	VCC				
33	VSS				
34		P24	MTIC5U	RXD5/SMISO5/SSCL5	IRQ3/COMP0
35		P23	MTIC5V/CACREF	TXD5/SMOSI5/SSDA5	IRQ4/COMP1
36		P22	MTIC5W		IRQ2/COMP2
37		P47*1			AN007/CMPC13
38		P46*1			AN006/CMPC03
39		P45*1			AN005/CMPC22
40		P44*1			AN004/CMPC12
41		P43*1			AN003/CMPC02
42		P42*1			AN002/CMPC20
43		P41*1			AN001/CMPC10
44		P40*1			AN000/CMPC00
45	AVCC0				
46	AVSS0				
47		P11	MTIOC3A/MTCLKA/POE8#		IRQ1/CVREFC0

Table 1.5 List of Pins and Pin Functions (48-Pin LFQFPLQFP/HWQFN) (2/2)

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, POE, CAC)	Communications (SCI, RIIC)	Others
48		P10	MTCLKB		IRQ0

Note 1. The power source of the I/O buffer for these pins is AVCC0.

1.6.2 32-Pin LQFP/HWQFN

Table 1.6 List of Pins and Pin Functions (32-Pin LQFP/HWQFN)

Pin No.	Power Supply, Clock, System Control	I/O Port	Timers (MTU, POE, CAC)	Communications (SCI, RIIC)	Others
1	VCL				
2	MD				FINED
3	RES#				
4	XTAL	P37			
5	VSS				
6	EXTAL	P36			
7	VCC				
8		PE2	POE10#		NMI/IRQ0
9		PB7	MTIOC3C/MTCLKD	RXD1/SMISO1/SSCL1/RXD5/SMISO5/ SSCL5	IRQ5
10		PB6	MTIOC1B/MTIOC3A	TXD1/SMOSI1/SSDA1/TXD5/SMOSI5/ SSDA5	
11		PB3	MTIOC0A/CACREF	SCK5/SCK12	
12		PB2	MTIOC0B/MTCLKC/ADSM0	TXD5/SMOSI5/SSDA5/SDA0	
13		PB1	MTIOC0C/MTIC5W/MTCLKA	RXD5/SMISO5/SSCL5/SCL0	IRQ2
14		PB0	MTIOC0D/MTIOC2A/MTCLKB	TXD12/TXDX12/SIOX12/SMOSI12/ SSDA12	
15		P94	MTIOC2B/MTIC5U/MTCLKA	RXD12/RXDX12/SMISO12/SSCL12	IRQ1
16		P93	MTIOC1A/MTIC5V	SCK5/SCK12	IRQ0/ADTRG0#
17		P76	MTIOC4D		
18		P75	MTIOC4C		
19		P74	MTIOC3D		
20		P73	MTIOC4B		
21		P72	MTIOC4A		
22		P71	MTIOC3B		
23	VCC				
24	VSS				
25		P44*1			AN004/CMPC12
26		P43*1			AN003/CMPC02
27		P42*1			AN002/CMPC20
28		P41*1			AN001/CMPC10
29		P40*1			AN000/CMPC00
30	AVCC0				
31	AVSS0				
32		P11	MTIOC3A/MTCLKA/POE8#		IRQ1/CVREFC0

Note 1. The power source of the I/O buffer for these pins is AVCC0.

2. CPU

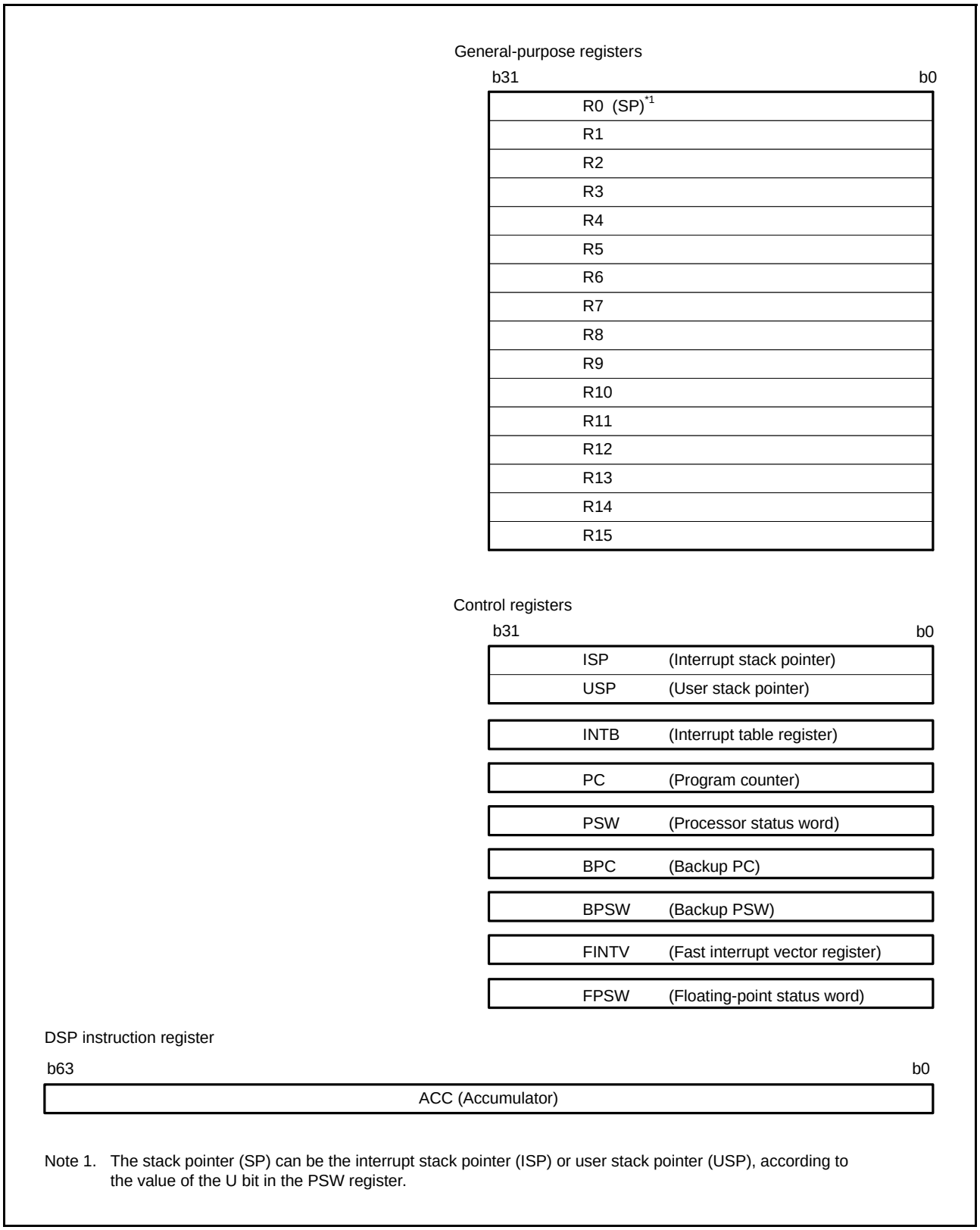
This MCU has the RX CPU as its core.
A variable-length instruction format has been adopted for the RX CPU. Allocating the more frequently used instructions to the shorter instruction lengths facilitates the development of efficient programs that take up less memory.
The CPU has 73 basic instructions, 8 floating-point operation instructions, and nine DSP instructions, for a total of 90 instructions. It has 10 addressing modes and caters to register-to-register operations, register-to-memory operations, immediate-to-register operations, immediate-to-memory operations, memory-to-memory transfer, and bitwise operations. In a single cycle, high-speed calculation is attained for not just register-to-register operations, but also for other types of combined instructions. The CPU includes an internal multiplier and an internal divider for high-speed multiplication and division.
The RX CPU has a five-stage pipeline for processing instructions. The stages are instruction fetching, decoding, execution, memory access, and write-back. In cases where pipeline processing is drawn-out by memory access, subsequent operations may in fact be executed earlier. By adopting an “out-of-order completion” of this kind, instruction execution is controlled to optimize the number of clock cycles.

2.1 Features

- Minimum instruction execution rate: One instruction per clock cycle
- Address space: 4-Gbyte linear
- Register set of the CPU
  General purpose: Sixteen 32-bit registers
  Control: Nine 32-bit registers
  Accumulator: One 64-bit register
- Floating-point operation instructions: 8
- Basic instructions: 73 (arithmetic/logic instructions, data-transfer instructions, branch instructions, bit-manipulation instructions, string-manipulation instructions, and system-manipulation instructions)
  Relative branch instructions to suit branch distances
  Variable-length instruction format (lengths from 1 to 8 bytes)
  Short formats for frequently used instructions
- DSP instructions: 9
  Supports 16-bit × 16-bit multiplication and multiply-and-accumulate operations.
  Rounds the data in the accumulator.
- Addressing modes: 10
- Five-stage pipeline
  Adoption of “out-of-order completion”
- Processor modes
  A supervisor mode and a user mode are supported.
- Floating-point operation unit
  Supports single-precision (32-bit) floating point
  Supports data types and exceptions in conformance with the IEEE754 standard
- Data arrangement
  Selectable as little endian or big endian

2.2 Register Set of the CPU

The RX CPU has 16 general-purpose registers, nine control registers, and one accumulator used for DSP instructions.



Note 1. The stack pointer (SP) can be the interrupt stack pointer (ISP) or user stack pointer (USP), according to the value of the U bit in the PSW register.

Figure 2.1 Register Set of the CPU

2.2.1 General-Purpose Registers (R0 to R15)

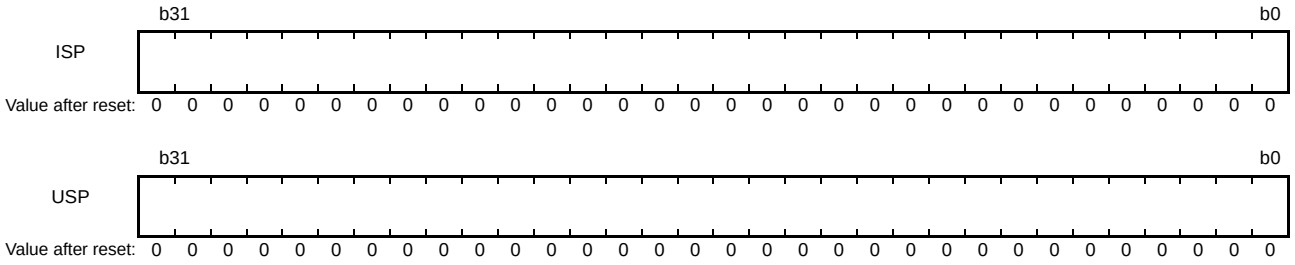
This CPU has 16 general-purpose registers (R0 to R15). R0 to R15 can be used as data registers or address registers. R0, a general-purpose register, also functions as the stack pointer (SP). The stack pointer is switched to operate as the interrupt stack pointer (ISP) or user stack pointer (USP) by the value of the stack pointer select bit (U) in the processor status word (PSW).

2.2.2 Control Registers

This CPU has the following nine control registers.

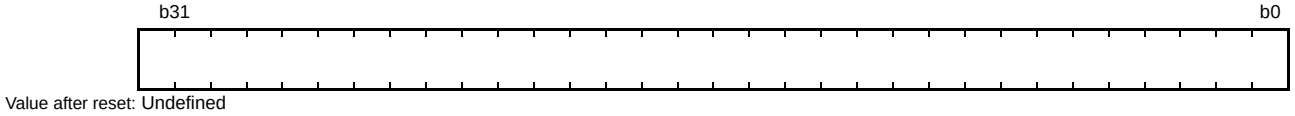
- Interrupt stack pointer (ISP)
- User stack pointer (USP)
- Interrupt table register (INTB)
- Program counter (PC)
- Processor status word (PSW)
- Backup PC (BPC)
- Backup PSW (BPSW)
- Fast interrupt vector register (FINTV)
- Floating-point status word (FPSW)

2.2.2.1 Interrupt Stack Pointer (ISP)/User Stack Pointer (USP)



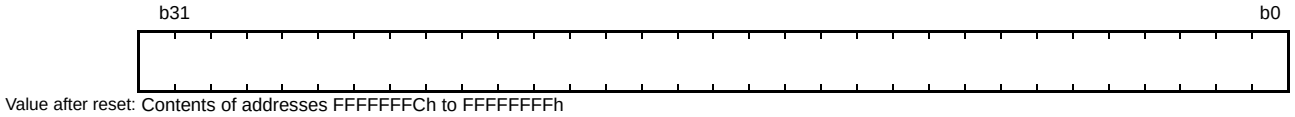
The stack pointer (SP) can be either of two types, the interrupt stack pointer (ISP) or the user stack pointer (USP). Whether the stack pointer operates as the ISP or USP depends on the value of the stack pointer select bit (U) in the processor status word (PSW).
Set the ISP or USP to a multiple of 4, as this reduces the numbers of cycles required to execute interrupt sequences and instructions entailing stack manipulation.

2.2.2.2 Interrupt Table Register (INTB)



The interrupt table register (INTB) specifies the address where the relocatable vector table starts.

2.2.2.3 Program Counter (PC)



The program counter (PC) indicates the address of the instruction being executed.

2.2.2.4 Processor Status Word (PSW)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	IPL[3:0]				—	—	—	PM	—	—	U	I
Value after reset:	0	0	0	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
	—	—	—	—	—	—	—	—	—	—	—	—	O	S	Z	C
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	C	Carry Flag	0: No carry has occurred. 1: A carry has occurred.	R/W
b1	Z	Zero Flag	0: Result is not 0. 1: Result is 0.	R/W
b2	S	Sign Flag	0: Result is a positive value or 0. 1: Result is a negative value.	R/W
b3	O	Overflow Flag	0: No overflow has occurred. 1: An overflow has occurred.	R/W
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	I*1	Interrupt Enable	0: Interrupt disabled. 1: Interrupt enabled.	R/W
b17	U*1	Stack Pointer Select	0: Interrupt stack pointer (ISP) is selected. 1: User stack pointer (USP) is selected.	R/W
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b20	PM*1,*2,*3	Processor Mode Select	0: Supervisor mode is selected. 1: User mode is selected.	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b24	IPL[3:0]*1	Processor Interrupt Priority Level	b27 b24 0 0 0 0: Priority level 0 (lowest) 0 0 0 1: Priority level 1 0 0 1 0: Priority level 2 0 0 1 1: Priority level 3 0 1 0 0: Priority level 4 0 1 0 1: Priority level 5 0 1 1 0: Priority level 6 0 1 1 1: Priority level 7 1 0 0 0: Priority level 8 1 0 0 1: Priority level 9 1 0 1 0: Priority level 10 1 0 1 1: Priority level 11 1 1 0 0: Priority level 12 1 1 0 1: Priority level 13 1 1 1 0: Priority level 14 1 1 1 1: Priority level 15 (highest)	R/W
b31 to b28	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. In user mode, writing to the IPL[3:0], PM, U, and I bits by an MVTC or a POPC instruction is ignored. Writing to the IPL[3:0] bits by an MVTIPL instruction generates a privileged instruction exception.
Note 2. In supervisor mode, writing to the PM bit by an MVTC or a POPC instruction is ignored, but writing to the other bits is possible.
Note 3. Switching from supervisor mode to user mode requires execution of an RTE instruction after having set the PSW.PM bit saved on the stack to 1 or executing an RTFI instruction after having set the BPSW.PM bit to 1.

The processor status word (PSW) indicates the results of instruction execution or the state of the CPU.

C Flag (Carry Flag)

This flag indicates whether a carry, borrow, or shift-out has occurred as the result of an operation.

Z Flag (Zero Flag)

This flag indicates that the result of an operation was 0.

S Flag (Sign Flag)

This flag indicates that the result of an operation was negative.

O Flag (Overflow Flag)

This flag indicates that an overflow occurred during an operation.

I Bit (Interrupt Enable)

This bit enables interrupt requests. When an exception is accepted, this bit becomes 0.

U Bit (Stack Pointer Select)

This bit specifies the stack pointer as either the ISP or USP. When an exception request is accepted, this bit becomes 0. When the processor mode is switched from supervisor mode to user mode, this bit becomes 1.

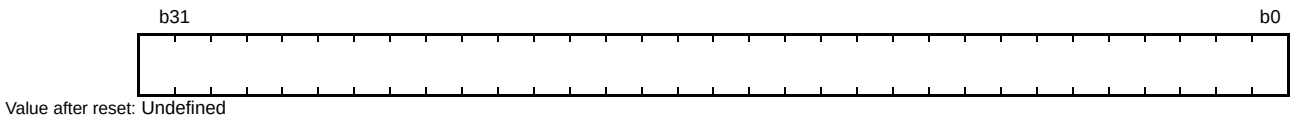
PM Bit (Processor Mode Select)

This bit specifies the processor mode. When an exception is accepted, this bit becomes 0.

IPL[3:0] Bits (Processor Interrupt Priority Level)

The IPL[3:0] bits specify the processor interrupt priority level as one of 16 levels from zero to 15, wherein priority level zero is the lowest and priority level 15 the highest. When the priority level of a requested interrupt is higher than the processor interrupt priority level, the interrupt is enabled. Setting the IPL[3:0] bits to level 15 (Fh) disables all interrupt requests. The IPL[3:0] bits are set to level 15 (Fh) when a non-maskable interrupt is generated. When interrupts are generated, the bits are set to the priority levels of accepted interrupts.

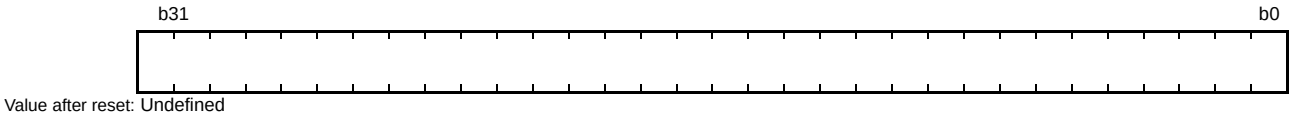
2.2.2.5 Backup PC (BPC)



Value after reset: Undefined

The backup PC (BPC) is provided to speed up response to interrupts.
After a fast interrupt has been generated, the contents of the program counter (PC) are saved in the BPC register.

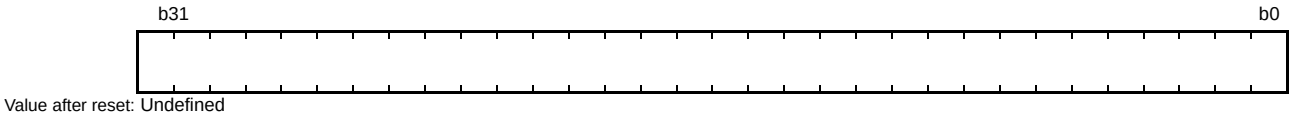
2.2.2.6 Backup PSW (BPSW)



Value after reset: Undefined

The backup PSW (BPSW) is provided to speed up response to interrupts.
After a fast interrupt has been generated, the contents of the processor status word (PSW) are saved in the BPSW. The allocation of bits in the BPSW corresponds to that in the PSW.

2.2.2.7 Fast Interrupt Vector Register (FINTV)



Value after reset: Undefined

The fast interrupt vector register (FINTV) is provided to speed up response to interrupts.
The FINTV register specifies a branch destination address when a fast interrupt has been generated.

2.2.2.8 Floating-Point Status Word (FPSW)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FS	FX	FU	FZ	FO	FV	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	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
	—	EX	EU	EZ	EO	EV	—	DN	CE	CX	CU	CZ	CO	CV	RM[1:0]	
Value after reset:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	RM[1:0]	Floating-Point Rounding-Mode Setting	b1 b0 0 0: Rounding towards the nearest value 0 1: Rounding towards 0 1 0: Rounding towards +∞ 1 1: Rounding towards –∞	R/W
b2	CV	Invalid Operation Cause Flag	0: No invalid operation has been encountered. 1: Invalid operation has been encountered.	R/(W) *1
b3	CO	Overflow Cause Flag	0: No overflow has occurred. 1: Overflow has occurred.	R/(W) *1
b4	CZ	Division-by-Zero Cause Flag	0: No division-by-zero has occurred. 1: Division-by-zero has occurred.	R/(W) *1
b5	CU	Underflow Cause Flag	0: No underflow has occurred. 1: Underflow has occurred.	R/(W) *1
b6	CX	Inexact Cause Flag	0: No inexact exception has been generated. 1: Inexact exception has been generated.	R/(W) *1
b7	CE	Unimplemented Processing Cause Flag	0: No unimplemented processing has been encountered. 1: Unimplemented process has been encountered.	R/(W) *1

Bit	Symbol	Bit Name	Description	R/W
b8	DN	0 Flush Bit of Denormalized Number	0: A denormalized number is handled as a denormalized number. 1: A denormalized number is handled as 0.*2	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	EV	Invalid Operation Exception Enable	0: Invalid operation exception is masked. 1: Invalid operation exception is enabled.	R/W
b11	EO	Overflow Exception Enable	0: Overflow exception is masked. 1: Overflow exception is enabled.	R/W
b12	EZ	Division-by-Zero Exception Enable	0: Division-by-zero exception is masked. 1: Division-by-zero exception is enabled.	R/W
b13	EU	Underflow Exception Enable	0: Underflow exception is masked. 1: Underflow exception is enabled.	R/W
b14	EX	Inexact Exception Enable	0: Inexact exception is masked. 1: Inexact exception is enabled.	R/W
b25 to b15	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b26	FV*3	Invalid Operation Flag	0: No invalid operation has been encountered. 1: Invalid operation has been encountered.*8	R/W
b27	FO*4	Overflow Flag	0: No overflow has occurred. 1: Overflow has occurred.*8	R/W
b28	FZ*5	Division-by-Zero Flag	0: No division-by-zero has occurred. 1: Division-by-zero has occurred.*8	R/W
b29	FU*6	Underflow Flag	0: No underflow has occurred. 1: Underflow has occurred.*8	R/W
b30	FX*7	Inexact Flag	0: No inexact exception has been generated. 1: Inexact exception has been generated.*8	R/W
b31	FS	Floating-Point Error Summary Flag	This bit reflects the logical OR of the FU, FZ, FO, and FV flags.	R

Note 1. Writing 0 to the bit clears it. Writing 1 to the bit does not affect its value.
Note 2. Positive denormalized numbers are treated as +0, negative denormalized numbers as –0.
Note 3. When the EV bit is set to 0, the FV flag is enabled.
Note 4. When the EO bit is set to 0, the FO flag is enabled.
Note 5. When the EZ bit is set to 0, the FZ flag is enabled.
Note 6. When the EU bit is set to 0, the FU flag is enabled.
Note 7. When the EX bit is set to 0, the FX flag is enabled.
Note 8. Once the bit has been set to 1, this value is retained until it is cleared to 0 by software.

The floating-point status word (FPSW) indicates the results of floating-point operations.
When an exception handling enable bit (Ej) enables the exception handling (Ej = 1), the exception cause can be identified by checking the corresponding Cj flag in the exception handling routine. If the exception handling is masked (Ej = 0), the occurrence of exception can be checked by reading the Fj flag at the end of a series of processing. Once the Fj flag has been set to 1, this value is retained until it is cleared to 0 by software (j = X, U, Z, O, or V).

RM[1:0] Bits (Floating-Point Rounding-Mode Setting)

These bits specify the floating-point rounding-mode.

Explanation of Floating-Point Rounding Modes

- Rounding towards the nearest value (the default behavior): An inexact result is rounded to the available value that is closest to the result which would be obtained with an infinite number of digits. If two available values are equally close, rounding is to the even alternative.
- Rounding towards 0: An inexact result is rounded to the smallest available absolute value, i.e. in the direction of zero (simple truncation).
- Rounding towards +∞: An inexact result is rounded to the nearest available value in the direction of positive infinity.

- Rounding towards –∞: An inexact result is rounded to the nearest available value in the direction of negative infinity.

(1) Rounding to the nearest value is specified as the default mode and returns the most accurate value.
(2) Modes such as rounding towards 0, rounding towards +∞, and rounding towards –∞ are used to ensure precision when interval arithmetic is employed.

CV Flag (Invalid Operation Cause Flag), CO Flag (Overflow Cause Flag), CZ Flag (Division-by-Zero Cause Flag), CU Flag (Underflow Cause Flag), CX Flag (Inexact Cause Flag), and CE Flag (Unimplemented Processing Cause Flag)

Floating-point exceptions include the five specified in the IEEE754 standard, namely overflow, underflow, inexact, division-by-zero, and invalid operation. For a further floating-point exception that is generated upon detection of unimplemented processing, the corresponding flag (CE) is set to 1.

- The bit that has been set to 1 is cleared to 0 when the FPU instruction is executed.
- When 0 is written to the bit by the MVTC and POPC instructions, the bit is set to 0; the bit retains the previous value when 1 is written by the instruction.

DN Bit (0 Flush Bit of Denormalized Number)

When this bit is set to 0, a denormalized number is handled as a denormalized number. When this bit is set to 1, a denormalized number is handled as 0.

EV Bit (Invalid Operation Exception Enable), EO Bit (Overflow Exception Enable), EZ Bit (Division-by-Zero Exception Enable), EU Bit (Underflow Exception Enable), and EX Bit (Inexact Exception Enable)

When any of five floating-point exceptions specified in the IEEE754 standard is generated by the floating-point operation instruction, the bit decides whether the CPU will start handling the exception. When the bit is set to 0, the exception handling is masked; when the bit is set to 1, the exception handling is enabled.

FV Flag (Invalid Operation Flag), FO Flag (Overflow Flag), FZ Flag (Division-by-Zero Flag), FU Flag (Underflow Flag), and FX Flag (Inexact Flag)

While the exception handling enable bit (Ej) is 0 (exception handling is masked), if any of five floating-point exceptions specified in the IEEE754 standard is generated, the corresponding bit is set to 1.

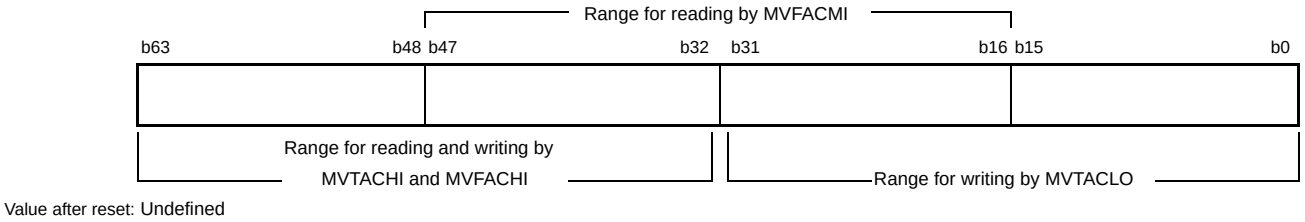
- When Ej is 1 (exception handling is enabled), the value of the flag remains.
- When the corresponding flag is set to 1, it remains 1 until it is cleared to 0 by software. (accumulation flag)

FS Flag (Floating-Point Error Summary Flag)

This bit reflects the logical OR of the FU, FZ, FO, and FV flags.

2.2.3 Register Associated with DSP Instructions

2.2.3.1 Accumulator (ACC)



Value after reset: Undefined

The accumulator (ACC) is a 64-bit register used for DSP instructions. The accumulator is also used for the multiply and multiply-and-accumulate instructions; EMUL, EMULU, FMUL, MUL, and RMPA, in which case the prior value in the accumulator is modified by execution of the instruction.
Use the MVTACHI and MVTACLO instructions for writing to the accumulator. The MVTACHI and MVTACLO instructions write data to the higher-order 32 bits (bits 63 to 32) and the lower-order 32 bits (bits 31 to 0), respectively.
Use the MVFACHI and MVFACMI instructions for reading data from the accumulator. The MVFACHI and MVFACMI instructions read data from the higher-order 32 bits (bits 63 to 32) and the middle 32 bits (bits 47 to 16), respectively.

2.3 Processor Mode

The RX CPU supports two processor modes, supervisor and user. These processor modes enable the realization of a hierarchical CPU resource protection.
Each processor mode imposes a level on rights of access to the CPU resources and the instructions that can be executed. Supervisor mode carries greater rights than those of user mode.
The initial state after a reset is supervisor mode.

2.3.1 Supervisor Mode

In supervisor mode, all CPU resources are accessible and all instructions are available. However, writing to the processor mode select bit (PM) in the processor status word (PSW) by executing an MVTC or a POPC instruction will be ignored. For details on how to write to the PM bit, refer to section 2.2.2.4, Processor Status Word (PSW).

2.3.2 User Mode

In user mode, write access to the CPU resources listed below is restricted. The restriction applies to any instruction capable of write access.

- Some bits (bits IPL[3:0], PM, U, and I) in the processor status word (PSW)
- Interrupt stack pointer (ISP)
- Interrupt table register (INTB)
- Backup PSW (BPSW)
- Backup PC (BPC)
- Fast interrupt vector register (FINTV)

2.3.3 Privileged Instruction

Privileged instructions can only be executed in supervisor mode. Executing a privileged instruction in user mode produces a privileged instruction exception. Privileged instructions include the RTFI, MVTIPL, RTE, and WAIT instructions.

2.3.4 Switching between Processor Modes

Manipulating the processor mode select bit (PM) in the processor status word (PSW) switches the processor mode. However, rewriting to the PM bit by executing an MVTC or a POPC instruction is prohibited. Switch the processor mode by following the procedures described below.

(1) Switching from user mode to supervisor mode

After an exception occurs, the PSW.PM bit is set to 0 and the CPU switches to supervisor mode. The hardware pre-processing is executed in supervisor mode. The state of the processor mode before the exception was generated is retained in the copy of PSW.PM bit is saved on the stack.

(2) Switching from supervisor mode to user mode

Executing an RTE instruction when the value of the copy of the PSW.PM bit that has been preserved on the stack is 1 or an RTFI instruction when the value of the copy of the PSW.PM bit that has been preserved in the backup PSW (BPSW) is 1 causes a transition to user mode. In the transition to user mode, the value of the stack pointer designation bit (the U bit in the PSW) becomes 1.

2.4 Data Types

The RX CPU can handle four types of data: integer, floating-point, bit, and string.
For details, refer to RX Family User's Manual: Software.

2.5 Endian

For the RX CPU, instructions are little endian, but the data arrangement is selectable as little or big endian.

2.5.1 Switching the Endian

As arrangements of bytes, this MCU supports both big endian, where the higher-order byte (MSB) is at location 0, and little endian, where the lower-order byte (LSB) is at location 0.
For details on the endian setting, refer to section 3, Operating Modes.
Operations for access differ according to the endian setting and, depending on the instruction, whether 8-, 16- or 32-bit access has been selected. Operations for access in the various possible cases are described in Table 2.1 to Table 2.12.
In the tables,

LL indicates bits D7 to D0 of the general-purpose register,
LH indicates bits D15 to D8 of the general-purpose register,
HL indicates bits D23 to D16 of the general-purpose register, and
HH indicates bits D31 to D24 of the general-purpose register.

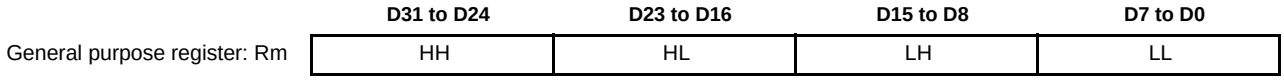


Table 2.1 32-Bit Read Operations when Little Endian has been Selected

Operation / Address of src	Reading a 32-bit unit from address 0	Reading a 32-bit unit from address 1	Reading a 32-bit unit from address 2	Reading a 32-bit unit from address 3	Reading a 32-bit unit from address 4
Address 0	Transfer to LL	—	—	—	—
Address 1	Transfer to LH	Transfer to LL	—	—	—
Address 2	Transfer to HL	Transfer to LH	Transfer to LL	—	—
Address 3	Transfer to HH	Transfer to HL	Transfer to LH	Transfer to LL	—
Address 4	—	Transfer to HH	Transfer to HL	Transfer to LH	Transfer to LL
Address 5	—	—	Transfer to HH	Transfer to HL	Transfer to LH
Address 6	—	—	—	Transfer to HH	Transfer to HL
Address 7	—	—	—	—	Transfer to HH

Table 2.2 32-Bit Read Operations when Big Endian has been Selected

Operation / Address of src	Reading a 32-bit unit from address 0	Reading a 32-bit unit from address 1	Reading a 32-bit unit from address 2	Reading a 32-bit unit from address 3	Reading a 32-bit unit from address 4
Address 0	Transfer to HH	—	—	—	—
Address 1	Transfer to HL	Transfer to HH	—	—	—
Address 2	Transfer to LH	Transfer to HL	Transfer to HH	—	—
Address 3	Transfer to LL	Transfer to LH	Transfer to HL	Transfer to HH	—
Address 4	—	Transfer to LL	Transfer to LH	Transfer to HL	Transfer to HH
Address 5	—	—	Transfer to LL	Transfer to LH	Transfer to HL
Address 6	—	—	—	Transfer to LL	Transfer to LH
Address 7	—	—	—	—	Transfer to LL

Table 2.3 32-Bit Write Operations when Little Endian has been Selected

Operation / Address of dest	Writing a 32-bit unit to address 0	Writing a 32-bit unit to address 1	Writing a 32-bit unit to address 2	Writing a 32-bit unit to address 3	Writing a 32-bit unit to address 4
Address 0	Transfer from LL	—	—	—	—
Address 1	Transfer from LH	Transfer from LL	—	—	—
Address 2	Transfer from HL	Transfer from LH	Transfer from LL	—	—
Address 3	Transfer from HH	Transfer from HL	Transfer from LH	Transfer from LL	—
Address 4	—	Transfer from HH	Transfer from HL	Transfer from LH	Transfer from LL
Address 5	—	—	Transfer from HH	Transfer from HL	Transfer from LH
Address 6	—	—	—	Transfer from HH	Transfer from HL
Address 7	—	—	—	—	Transfer from HH

Table 2.4 32-Bit Write Operations when Big Endian has been Selected

Operation / Address of dest	Writing a 32-bit unit to address 0	Writing a 32-bit unit to address 1	Writing a 32-bit unit to address 2	Writing a 32-bit unit to address 3	Writing a 32-bit unit to address 4
Address 0	Transfer from HH	—	—	—	—
Address 1	Transfer from HL	Transfer from HH	—	—	—
Address 2	Transfer from LH	Transfer from HL	Transfer from HH	—	—
Address 3	Transfer from LL	Transfer from LH	Transfer from HL	Transfer from HH	—
Address 4	—	Transfer from LL	Transfer from LH	Transfer from HL	Transfer from HH
Address 5	—	—	Transfer from LL	Transfer from LH	Transfer from HL
Address 6	—	—	—	Transfer from LL	Transfer from LH
Address 7	—	—	—	—	Transfer from LL

Table 2.5 16-Bit Read Operations when Little Endian has been Selected

Operation / Address of src	Reading a 16-bit unit from address 0	Reading a 16-bit unit from address 1	Reading a 16-bit unit from address 2	Reading a 16-bit unit from address 3	Reading a 16-bit unit from address 4	Reading a 16-bit unit from address 5	Reading a 16-bit unit from address 6
Address 0	Transfer to LL	—	—	—	—	—	—
Address 1	Transfer to LH	Transfer to LL	—	—	—	—	—
Address 2	—	Transfer to LH	Transfer to LL	—	—	—	—
Address 3	—	—	Transfer to LH	Transfer to LL	—	—	—
Address 4	—	—	—	Transfer to LH	Transfer to LL	—	—
Address 5	—	—	—	—	Transfer to LH	Transfer to LL	—
Address 6	—	—	—	—	—	Transfer to LH	Transfer to LL
Address 7	—	—	—	—	—	—	Transfer to LH

Table 2.6 16-Bit Read Operations when Big Endian has been Selected

Operation / Address of src	Reading a 16-bit unit from address 0	Reading a 16-bit unit from address 1	Reading a 16-bit unit from address 2	Reading a 16-bit unit from address 3	Reading a 16-bit unit from address 4	Reading a 16-bit unit from address 5	Reading a 16-bit unit from address 6
Address 0	Transfer to LH	—	—	—	—	—	—
Address 1	Transfer to LL	Transfer to LH	—	—	—	—	—
Address 2	—	Transfer to LL	Transfer to LH	—	—	—	—
Address 3	—	—	Transfer to LL	Transfer to LH	—	—	—
Address 4	—	—	—	Transfer to LL	Transfer to LH	—	—
Address 5	—	—	—	—	Transfer to LL	Transfer to LH	—
Address 6	—	—	—	—	—	Transfer to LL	Transfer to LH
Address 7	—	—	—	—	—	—	Transfer to LL

Table 2.7 16-Bit Write Operations when Little Endian has been Selected

Operation / Address of dest	Writing a 16-bit unit to address 0	Writing a 16-bit unit to address 1	Writing a 16-bit unit to address 2	Writing a 16-bit unit to address 3	Writing a 16-bit unit to address 4	Writing a 16-bit unit to address 5	Writing a 16-bit unit to address 6
Address 0	Transfer from LL	—	—	—	—	—	—
Address 1	Transfer from LH	Transfer from LL	—	—	—	—	—
Address 2	—	Transfer from LH	Transfer from LL	—	—	—	—
Address 3	—	—	Transfer from LH	Transfer from LL	—	—	—
Address 4	—	—	—	Transfer from LH	Transfer from LL	—	—
Address 5	—	—	—	—	Transfer from LH	Transfer from LL	—
Address 6	—	—	—	—	—	Transfer from LH	Transfer from LL
Address 7	—	—	—	—	—	—	Transfer from LH

Table 2.8 16-Bit Write Operations when Big Endian has been Selected

Operation / Address of dest	Writing a 16-bit unit to address 0	Writing a 16-bit unit to address 1	Writing a 16-bit unit to address 2	Writing a 16-bit unit to address 3	Writing a 16-bit unit to address 4	Writing a 16-bit unit to address 5	Writing a 16-bit unit to address 6
Address 0	Transfer from LL	—	—	—	—	—	—
Address 1	Transfer from LH	Transfer from LL	—	—	—	—	—
Address 2	—	Transfer from LH	Transfer from LL	—	—	—	—
Address 3	—	—	Transfer from LH	Transfer from LL	—	—	—
Address 4	—	—	—	Transfer from LH	Transfer from LL	—	—
Address 5	—	—	—	—	Transfer from LH	Transfer from LL	—
Address 6	—	—	—	—	—	Transfer from LH	Transfer from LL
Address 7	—	—	—	—	—	—	Transfer from LH

Table 2.9 8-Bit Read Operations when Little Endian has been Selected

Operation / Address of src	Reading an 8-bit unit from address 0	Reading an 8-bit unit from address 1	Reading an 8-bit unit from address 2	Reading an 8-bit unit from address 3
Address 0	Transfer to LL	—	—	—
Address 1	—	Transfer to LL	—	—
Address 2	—	—	Transfer to LL	—
Address 3	—	—	—	Transfer to LL

Table 2.10 8-Bit Read Operations when Big Endian has been Selected

Operation / Address of src	Reading an 8-bit unit from address 0	Reading an 8-bit unit from address 1	Reading an 8-bit unit from address 2	Reading an 8-bit unit from address 3
Address 0	Transfer to LL	—	—	—
Address 1	—	Transfer to LL	—	—
Address 2	—	—	Transfer to LL	—
Address 3	—	—	—	Transfer to LL

Table 2.11 8-Bit Write Operations when Little Endian has been Selected

Operation / Address of dest	Writing an 8-bit unit to address 0	Writing an 8-bit unit to address 1	Writing an 8-bit unit to address 2	Writing an 8-bit unit to address 3
Address 0	Transfer from LL	—	—	—
Address 1	—	Transfer from LL	—	—
Address 2	—	—	Transfer from LL	—
Address 3	—	—	—	Transfer from LL

Table 2.12 8-Bit Write Operations when Big Endian has been Selected

Operation / Address of dest	Writing an 8-bit unit to address 0	Writing an 8-bit unit to address 1	Writing an 8-bit unit to address 2	Writing an 8-bit unit to address 3
Address 0	Transfer from LL	—	—	—
Address 1	—	Transfer from LL	—	—
Address 2	—	—	Transfer from LL	—
Address 3	—	—	—	Transfer from LL

2.5.2 Access to I/O Registers

The addresses of I/O registers are fixed, and this is regardless of whether the setting is for little endian or big endian. Accordingly, changes to the endian do not affect access to I/O registers. For the arrangements of I/O registers, refer to the descriptions of registers in the relevant sections.

2.5.3 Notes on Access to I/O Registers

Ensure that access to I/O registers is in accord with the following rules.

- With I/O registers for which a bus width of 8 bits is indicated, use instructions having operands of the same width (8 bits). That is, access these registers by using instructions with .B as the size specifier (.size), or with .B or .UB as the size-extension specifier (.memex).
- With I/O registers for which a bus width of 16 bits is indicated, use instructions having operands of the same width (16 bits). That is, access these registers by using instructions with .W as the size specifier (.size), or with .W or .UW as the size-extension specifier (.memex).
- With I/O registers for which a bus width of 32 bits is indicated, use instructions having operands of the same width (32 bits). That is, access these registers by using instructions with .L as the size specifier (.size), or with .L size-extension specifier (.memex).

2.5.4 Data Arrangement

2.5.4.1 Data Arrangement in Registers

Figure 2.2 shows the relation between the sizes of registers and bit numbers.

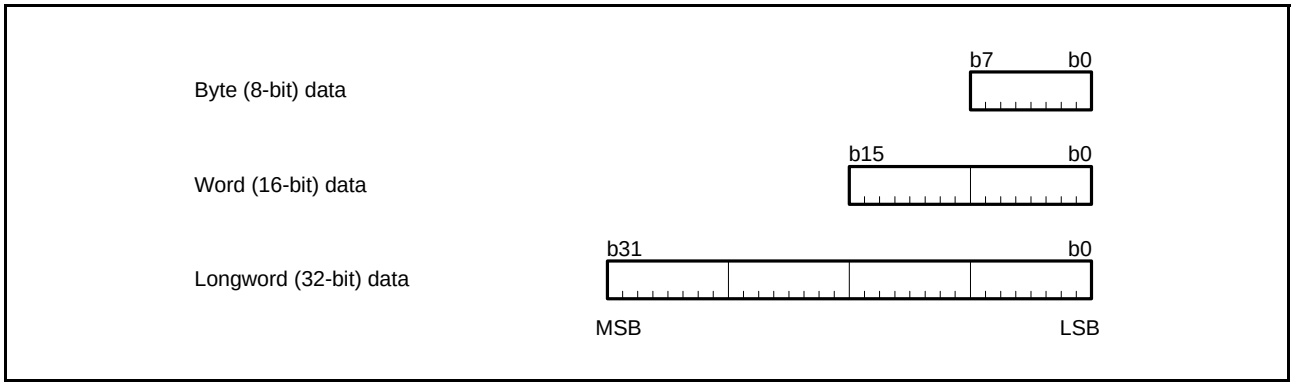


Figure 2.2 Data Arrangement in Registers

2.5.4.2 Data Arrangement in Memory

Data in memory have three sizes: byte (8-bit), word (16-bit), and longword (32-bit). The data arrangement is selectable as little endian or big endian. Figure 2.3 shows the arrangement of data in memory.

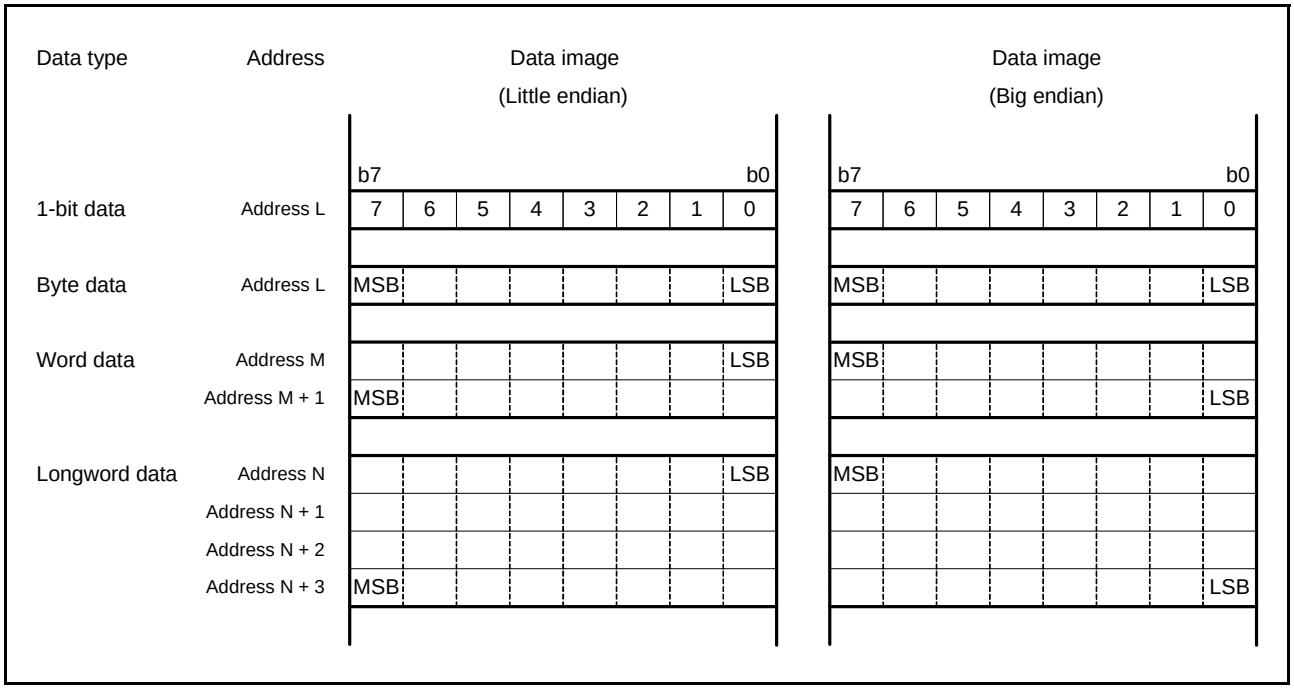


Figure 2.3 Data Arrangement in Memory

2.5.5 Notes on the Allocation of Instruction Codes

The allocation of instruction codes to an external space where the endian differs from that of the chip is prohibited. If the instruction codes are allocated to the external space, they must be allocated to areas where the endian setting is the same as that for the chip.

2.6 Vector Table

There are two types of vector table: fixed and relocatable. Each vector in the vector table consists of 4 bytes and specifies the address where the corresponding exception handling routine starts.

2.6.1 Fixed Vector Table

The fixed vector table is allocated to a fixed address range. The individual vectors for the privileged instruction exception, undefined instruction exception, floating-point exception, non-maskable interrupt, and reset are allocated to addresses in the range from FFFFFF80h to FFFFFFFFh. Figure 2.4 shows the fixed vector table.

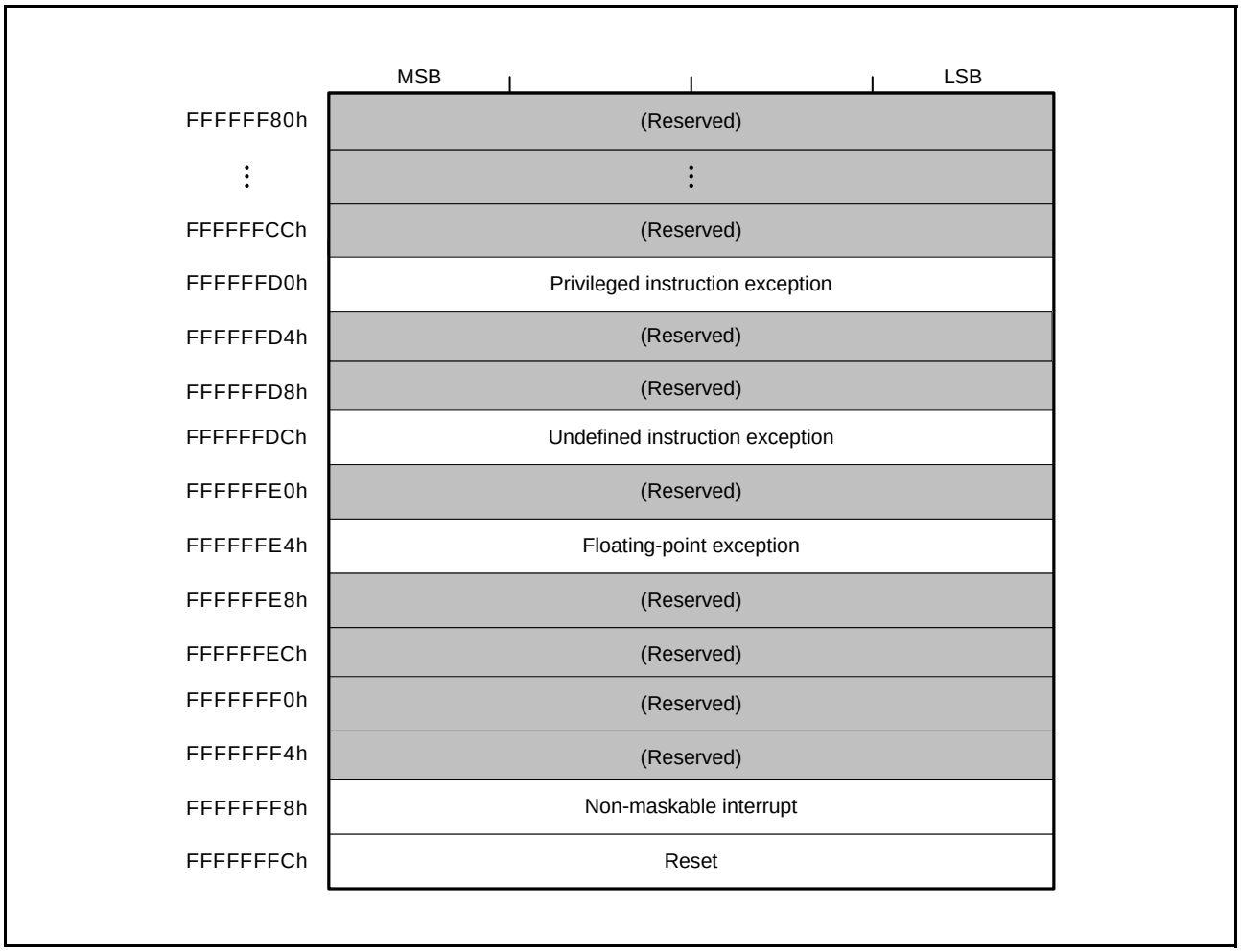


Figure 2.4 Fixed Vector Table

2.6.2 Relocatable Vector Table

The address where the relocatable vector table is placed can be adjusted. The table is a 1,024-byte region that contains all vectors for unconditional traps and interrupts and starts at the address (IntBase) specified in the interrupt table register (INTB). Figure 2.5 shows the relocatable vector table.

Each vector in the relocatable vector table has a vector number from 0 to 255. Each of the INT instructions, which act as the sources of unconditional traps, is allocated to the vector that has the same number as is specified as the operand of the instruction itself (from 0 to 255). The BRK instruction is allocated to the vector with number 0. Furthermore, vector numbers (from 0 to 255) are allocated to interrupt requests in a fixed way for each product. For more on interrupt vector numbers, refer to section 14.3.1, Interrupt Vector Table.

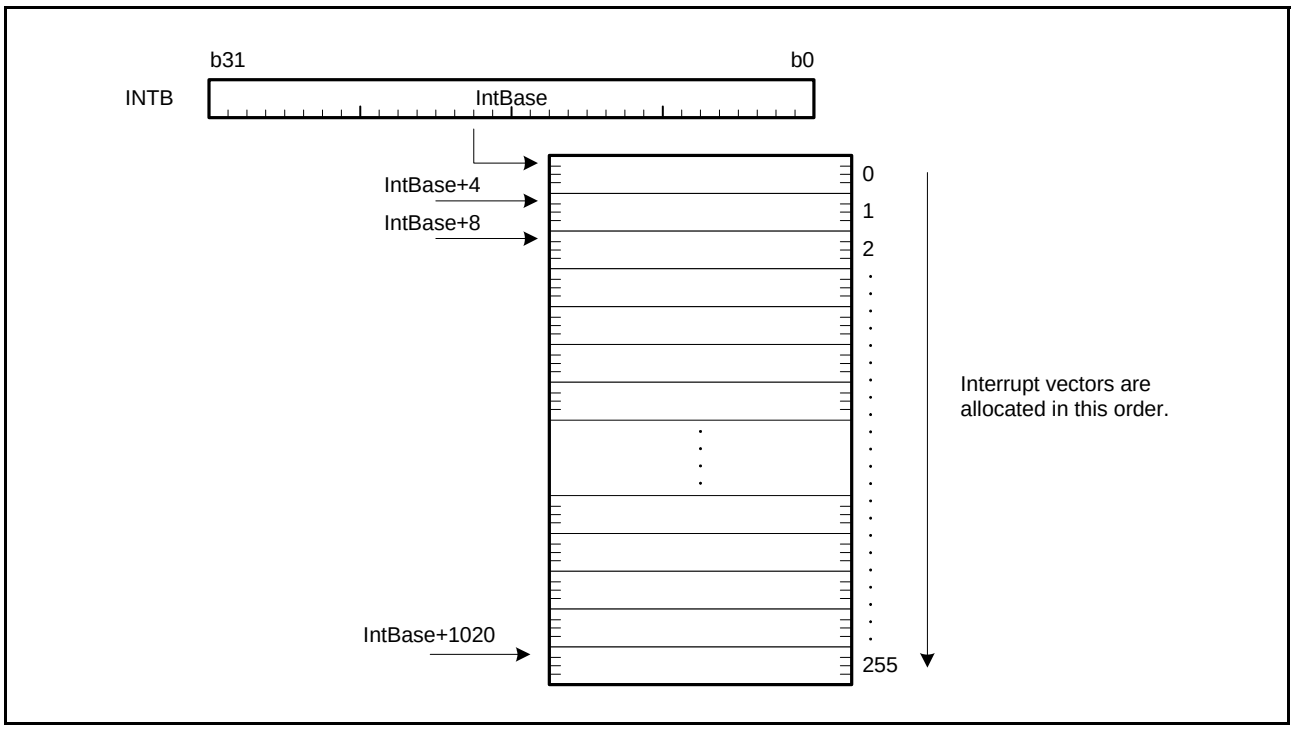


Figure 2.5 Relocatable Vector Table

2.7 Operation of Instructions

2.7.1 Data Prefetching by the RMPA Instruction and the String-Manipulation Instructions

The RMPA instruction and the string-manipulation instructions except the SSTR instruction (that is, SCMPU, SMOVB, SMOVF, SMOVU, SUNTIL, and SWHILE instructions) may prefetch data from the memory to speed up the read processing. Data is prefetched from the prefetching start position with 3 bytes as the upper limit. The prefetching start positions of each operation are shown below.

- RMPA instruction: The multiplicand address specified by R1, and the multiplier address specified by R2
- SCMPU instruction: The source address specified by R1 for comparison, and the destination address specified by R2 for comparison
- SUNTIL and SWHILE instructions: The destination address specified by R1 for comparison
- SMOVB, SMOVF, and SMOVU instructions: The source address specified by R2 for transfer

2.8 Pipeline

2.8.1 Overview

The RX CPU has five-stage pipeline structure. The RX CPU instruction is converted into one or more micro-operations, which are then executed in pipeline processing. In the pipeline stage, the IF stage is executed in the unit of instructions, while the D and subsequent stages are executed in the unit of micro-operations.
The operation of pipeline and respective stages is described below.

(1) IF stage (instruction fetch stage)

In the IF stage, the CPU fetches instructions from the memory. As the RX CPU has four 4-byte instruction queues, it fetches instructions until the instruction queue is full, regardless of the completion of decoding in the D (decoding) stage.

(2) D stage (decoding stage)

The CPU decodes instructions (DEC) in the D stage and converts them into micro-operations. The CPU reads the register information (RF) in this stage and executes a bypass process (BYP) if the result of the preceding instruction will be used in a subsequent instruction. The write of operation result to the register (RW) can be executed with the register reference by using the bypass process.

(3) E stage (execution stage)

Operations and address calculations (OP) are processed in the E stage.

(4) M stage (memory access stage)

Operand memory accesses (OA1, OA2) are processed in the M stage. This stage is used only when the memory is accessed, and is divided into two sub-stages, M1 and M2. The RX CPU enables respective memory accesses for M1 and M2.

- M1 stage (memory-access stage 1)
  Operand memory access (OA1) is processed.
  Store operation: The pipeline processing ends when a write request is received via the bus.
  Load operation: The operation proceeds to the M2 stage when a read request is received via the bus. If a request and load data are received at the same timing (no-wait memory access), the operation proceeds to the WB stage.
- M2 stage (memory-access stage 2)
  Operand memory access (OA2) is processed. The CPU waits for the load data in the M2 stage. When the load data is received, the operation proceeds to the WB stage.

(5) WB stage (write-back stage)

The operation result and the data read from memory are written to the register (RW) in the WB stage. The data read from memory and the other type of data, such as the operation result, can be written to the register in the same clock cycles.

Figure 2.6 shows the pipeline configuration and its operation.

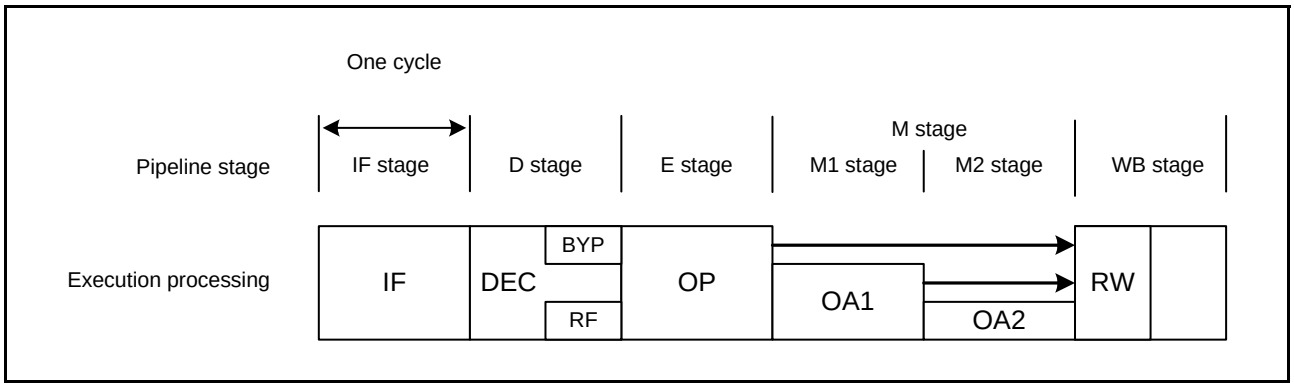


Figure 2.6 Pipeline Configuration and its Operation

2.8.2 Instructions and Pipeline Processing

The operands in the table below indicate the following meaning.

#IMM: Immediate
flag: bit, flag
Rs, Rs2, Rd, Rd2, Ri, Rb: General-purpose register
CR: Control register
dsp: displacement
pcdsp: displacement

2.8.2.1 Instructions Converted into Single Micro-Operation and Pipeline Processing

The table below lists the instructions that are converted into a single micro-operation. The number of cycles in the table indicates the number of cycles during no-wait memory access.

Table 2.13 Instructions that are Converted into a Single Micro-Operation

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Reference Figure	Number of Cycles
Arithmetic/logic instructions (register-register, immediate-register) Except EMUL, EMULU, RMPA, DIV, DIVU and SATR	• {ABS, NEG, NOT} “Rd”/“Rs, Rd” • {ADC, MAX, MIN, ROTL, ROTR, XOR} “#IMM, Rd”/“Rs, Rd” • ADD “#IMM, Rd”/“Rs, Rd”/“#IMM, Rs, Rd”/“Rs, Rs2, Rd” • {AND, MUL, OR, SUB} “#IMM, Rd”/“Rs, Rd”/“Rs, Rs2, Rd” • {CMP, TST} “#IMM, Rs”/“Rs, Rs2” • NOP • {ROLC, RORC, SAT} “Rd” • SBB “Rs, Rd” • {SHAR, SHLL, SHLR} “#IMM, Rd”/“Rs, Rd”/“#IMM, Rs, Rd”	Figure 2.7	1
Arithmetic/logic instructions (division)	• DIV “#IMM, Rd”/“Rs, Rd”	Figure 2.7	3 to 20*1
	• DIVU “#IMM, Rd”/“Rs, Rd”	Figure 2.7	2 to 18*1
Data transfer instructions (register-register, immediate-register)	• MOV “#IMM, Rd”/“Rs, Rd” • {MOVU, REVL, REVW} “Rs, Rd” • SC*Cnd* “Rd” • {STNZ, STZ} “#IMM, Rd”	Figure 2.7	1
Transfer instructions (load operation)	• {MOV, MOVU} “[Rs], Rd”/“dsp[Rs], Rd”/“[Rs+], Rd”/“[-Rs], Rd”/“[Ri, Rb], Rd” • POP “Rd”	Figure 2.8	Throughput: 1 Latency: 2*2
Transfer instructions (store operation)	• MOV “Rs, [Rd]”/“Rs, dsp[Rd]”/“Rs, [Rd+]”/“Rs, [-Rd]”/“Rs, [Ri, Rb]”/“#IMM, dsp[Rd]”/“#IMM, [Rd]” • PUSH “Rs” • PUSHC “CR” • SCCnd “[Rd]”/“dsp[Rd]”	Figure 2.9	1
Bit manipulation instructions (register)	• {BCLR, BNOT, BSET} “#IMM, Rd”/“Rs, Rd” • BM*Cnd* “#IMM, Rd” • BTST “#IMM, Rs”/“Rs, Rs2”	Figure 2.7	1
Branch instructions	• B*Cnd* “pcdsp” • {BRA, BSR} “pcdsp”/“Rs” • {JMP, JSR} “Rs”	Figure 2.18	Branch taken: 3 Branch not taken: 1
Floating-point operation instructions (register-register, immediate-register)	• FCMP “#IMM, Rd”/“Rs, Rs2”	Figure 2.7	1
System manipulation instructions	• {CLRPSW, SETPSW} “flag” • MVTC “#IMM, CR”/“Rs, CR” • MVFC “CR, Rd” • MVTIPL“#IMM”	—	1
DSP instructions	• {MACHI, MACLO, MULHI, MULLO} “Rs, Rs2” • {MVFACHI, MVFACMI} “Rd” • {MVTACHI, MVTACLO} “Rs” • RACW“#IMM”	Figure 2.7	1

Note 1. The number of cycles for the dividing instruction varies according to the divisor and dividend.
Note 2. For the number of cycles for throughput and latency, refer to section 2.8.3, Calculation of the Instruction Processing Time.

Figure 2.7 to Figure 2.9 show the operation of instructions that are converted into a basic single micro-operation.

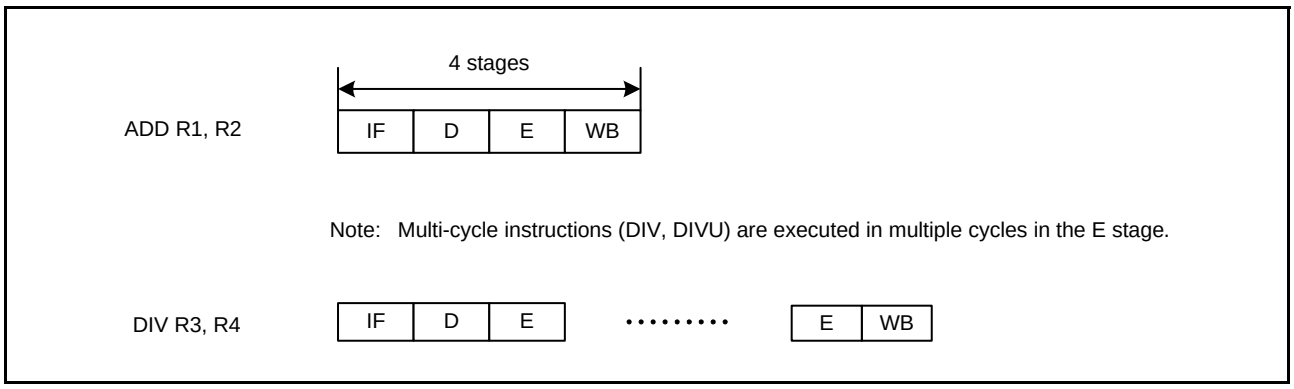


Figure 2.7 Operation for Register-Register, Immediate-Register

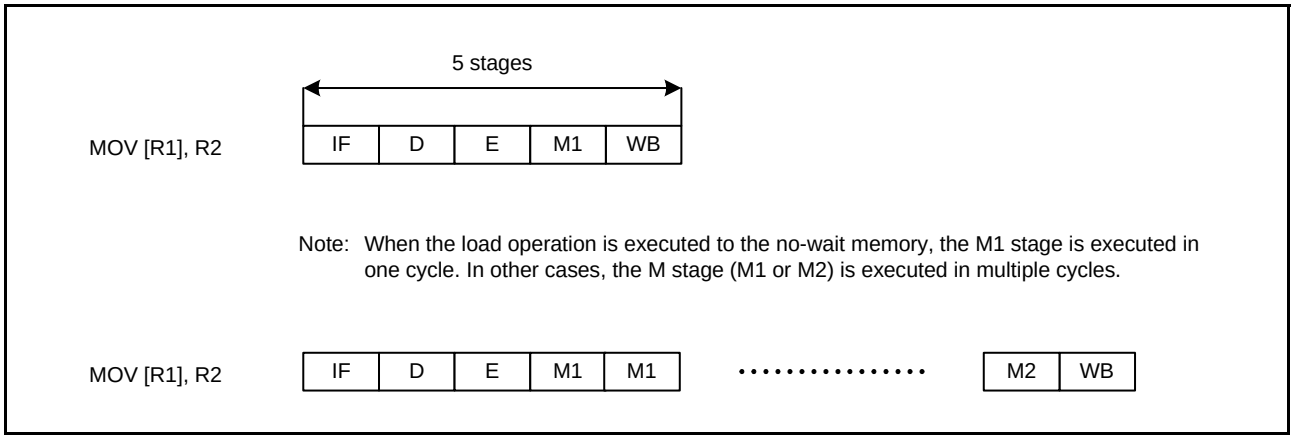


Figure 2.8 Load Operation

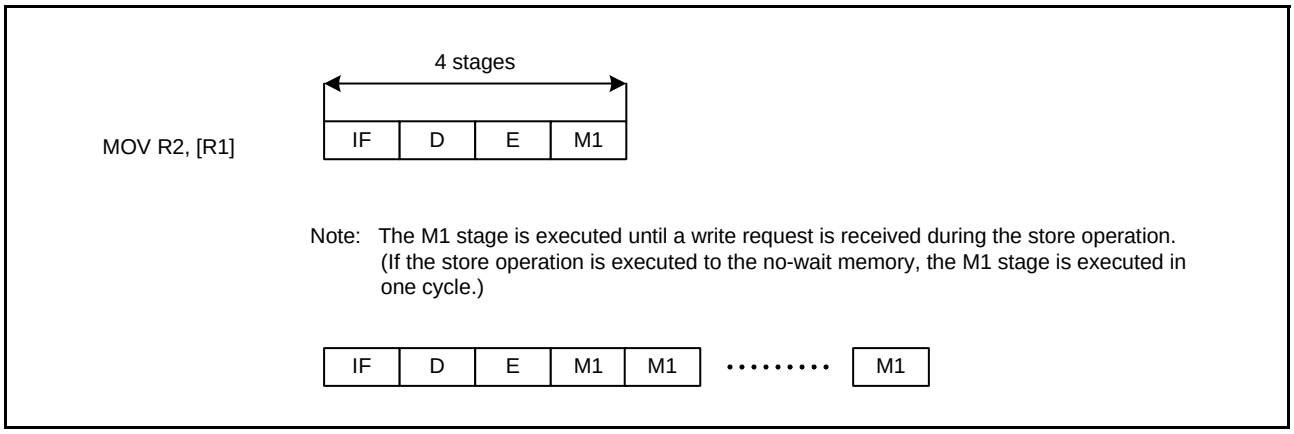


Figure 2.9 Store Operation

2.8.2.2 Instructions Converted into Multiple Micro-Operations and Pipeline Processing

The table below lists the instructions that are converted into multiple micro-operations. The number of cycles in the table indicates the number of cycles during no-wait memory access.

Table 2.14 Instructions that are Converted into Multiple Micro-Operations (1/2)

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Reference Figure	Number of Cycles
Arithmetic/logic instructions (memory source operand)	• {ADC, ADD, AND, MAX, MIN, MUL, OR, SBB, SUB, XOR} “[Rs], Rd”/“dsp[Rs], Rd” • {CMP, TST} “[Rs], Rs2”/“dsp[Rs], Rs2”	Figure 2.10	3
Arithmetic/logic instructions (division)	• DIV “[Rs], Rd / dsp[Rs], Rd”	—	5 to 22
	• DIVU“[Rs], Rd / dsp[Rs], Rd”	—	4 to 20
Arithmetic/logic instruction (multiplier: 32 × 32 → 64 bits) (register-register, register-immediate)	• {EMUL, EMULU} “#IMM, Rd”/“Rs, Rd”	Figure 2.12	2
Arithmetic/logic instruction (multiplier: 32 × 32 → 64 bits) (memory source operand)	• {EMUL, EMULU} “[Rs], Rd”/“dsp[Rs], Rd”	—	4
Arithmetic/logic instructions (multiply-and-accumulate operation)	• RMPA.B	—	6+7×floor(n/4)+4×(n%4) n: Number of processing bytes*1
	• RMPA.W	—	6+5×floor(n/2)+4×(n%2) n: Number of processing words*1
	• RMPA.L	—	6+4n n: Number of processing longwords*1
Arithmetic/logic instruction (64-bit signed saturation processing for the RMPA instruction)	• SATR	—	3
Data transfer instructions (memory-memory transfer)	• MOV “[Rs], [Rd]”/“dsp[Rs], [Rd]”/“[Rs], dsp[Rd]”/“dsp[Rs], dsp[Rd]” • PUSH “[Rs]”/“dsp[Rs]”	Figure 2.11	3
Bit manipulation instructions (memory source operand)	• {BCLR, BNOT, BSET} “#IMM, [Rd]”/“#IMM, dsp[Rd]”/“Rs, [Rd]”/“Rs, dsp[Rd]” • BM*Cnd* “#IMM, [Rd]”/“#IMM, dsp[Rd]” • BTST “#IMM, [Rs]”/“#IMM, dsp[Rs]”/“Rs, [Rs2]”/“Rs, dsp[Rs2]	Figure 2.11	3
Transfer instruction (load operation)	• POPC “CR”	—	Throughput: 3 Latency: 4*2
Transfer instruction (save operation of multiple registers)	• PUSHM “Rs-Rs2”	—	n n: Number of registers*3
Transfer instruction (restore operation of multiple registers)	• POPM “Rs-Rs2”	—	Throughput: n Latency: n+1 n: Number of registers*2,*4
Transfer instruction (register-register)	• XCHG “Rs, Rd”	Figure 2.13	2
Transfer instruction (memory-register)	• XCHG “[Rs], Rd”/“dsp[Rs], Rd”	Figure 2.14	2
Branch instructions	• RTS	—	5
	• RTSD “#IMM”	—	5
	• RTSD “#IMM, Rd-Rd2”	—	Throughput: n<5?5:1+n Latency: n<4?5:2+n n: Number of registers*2

Table 2.14 Instructions that are Converted into Multiple Micro-Operations (2/2)

Instruction	Mnemonic (indicates the common operation when the size is omitted)	Reference Figure	Number of Cycles
String manipulation instructions*5	• SCMPU	—	2+4×floor(n/4)+4×(n%4) n: Number of comparison bytes*1
	• SMOVB	—	n>3? 6+3×floor(n/4)+3×(n%4): 2+3n n: Number of transfer bytes*1
	• SMOVF, SMOVU	—	2+3×floor(n/4)+3×(n%4) n: Number of transfer bytes*1
	• SSTR.B	—	2+floor(n/4)+n%4 n: Number of transfer bytes*1
	• SSTR.W	—	2+floor(n/2)+n%2 n: Number of transfer words*1
	• SSTR.L	—	2+n n: Number of transfer longwords
	• SUNTIL.B, SWHILE.B	—	3+3×floor(n/4)+3×(n%4) n: Number of comparison bytes*1
	• SUNTIL.W, SWHILE.W	—	3+3×floor(n/2)+3×(n%2) n: Number of comparison words*1
	• SUNTIL.L, SWHILE.L	—	3+3×n n: Number of comparison longwords
Floating-point operation instructions (register-register, immediate-register)	• {FADD, FSUB} “#IMM, Rd”/“Rs, Rd”	Figure 2.15	4
	• FMUL “#IMM, Rd”/“Rs, Rd”	—	3
	• FDIV “#IMM, Rd”/“Rs, Rd”	—	16
	• {FTOI, ROUND, ITOF} “Rs, Rd”	—	2
Floating-point operation instructions (memory source operand)	• {FADD, FSUB} “[Rs], Rd”/“dsp[Rs], Rd”	—	6
	• FCMP “[Rs], Rs2”/“dsp[Rs], Rs2”	—	3
	• FMUL “[Rs], Rd”/“dsp[Rs], Rd”	—	5
	• FDIV “[Rs], Rd”/“dsp[Rs], Rd”	—	18
	• {FTOI, ROUND, ITOF} “[Rs], Rd”/ “dsp[Rs], Rd”	—	4
System manipulation instructions	• RTE	—	6
	• RTFI	—	3

?: Conditional operator

Note 1. floor(x): Max. integer that is smaller than x
Note 2. For the number of cycles for throughput and latency, refer to section 2.8.3, Calculation of the Instruction Processing Time.
Note 3. The PUSHM instruction is converted into multiple store operations. The pipeline processing is the same as the one for the store operations of the MOV instruction, where the operation is repeated for the number of specified registers.
Note 4. The POPM instruction is converted into multiple load operations. The pipeline processing is the same as the one for the load operations of the MOV instruction, where the operation is repeated for the number of specified registers.
Note 5. Each of the SCMPU, SMOVU, SWHILE, and SUNTIL instructions ends the execution regardless of the specified cycles, if the end condition is satisfied during execution.

Figure 2.10 to Figure 2.14 show the operation of instructions that are converted into basic multiple micro-operations.
Note: mop: Micro-operation, stall: Pipeline stall

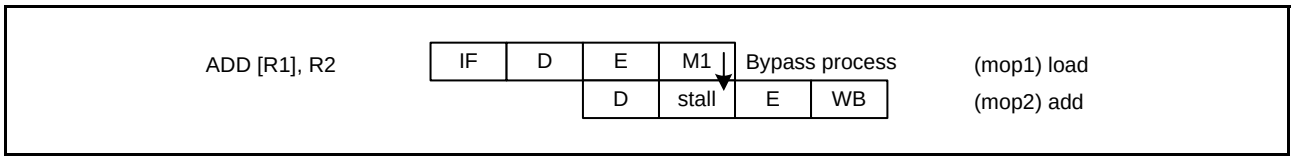


Figure 2.10 Arithmetic/Logic Instruction (Memory Source Operand)

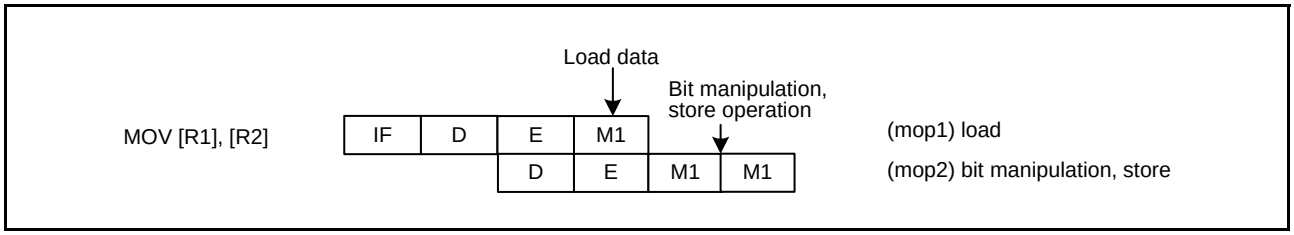


Figure 2.11 MOV Instruction (Memory-Memory), Bit Manipulation Instruction (Memory Source Operand)

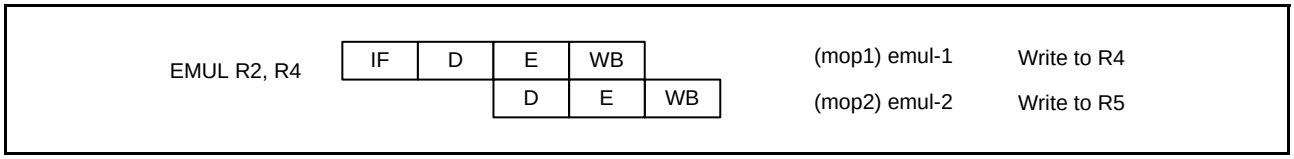


Figure 2.12 EMUL, EMULU Instructions (Register- Register, Register-Immediate)

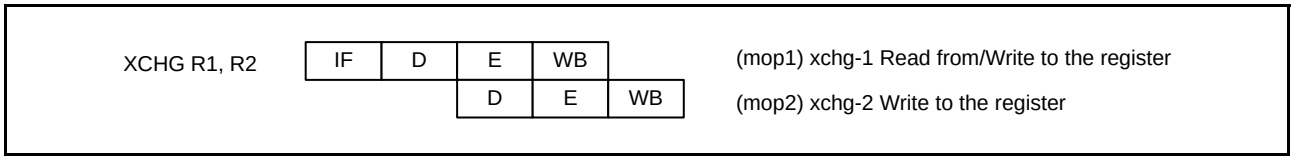


Figure 2.13 XCHG Instruction (Registers)

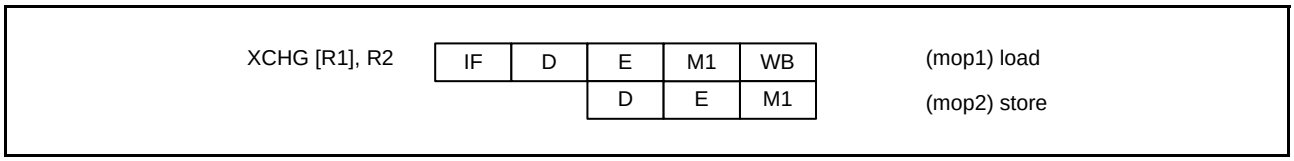


Figure 2.14 XCHG Instruction (Memory Source Operand)

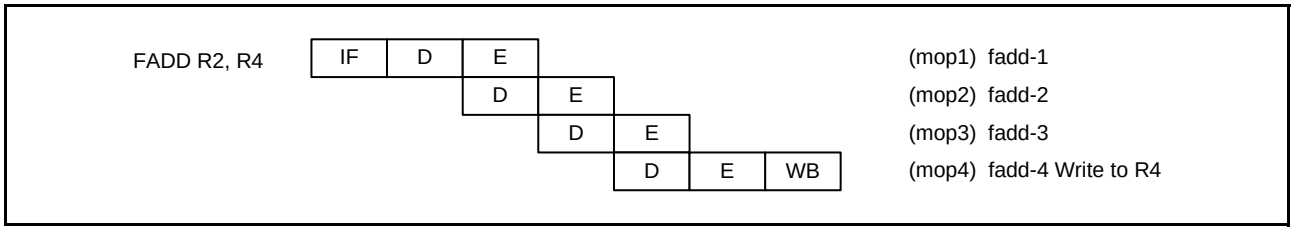


Figure 2.15 Floating-Point Operation Instruction (Register-Register, Immediate-Register)

2.8.2.3 Pipeline Basic Operation

In the ideal pipeline processing, each stage is executed in one cycle, though all instructions may not be pipelined in due to the processing in each stage and the branch execution.
The CPU controls the pipeline stage with the IF stage in the unit of instructions, while the D and subsequent stages in the unit of micro-operations.
The figures below show the pipeline processing of typical cases.
Note: mop: Micro-operation, stall: Pipeline stall

(1) Pipeline Flow with Stalls

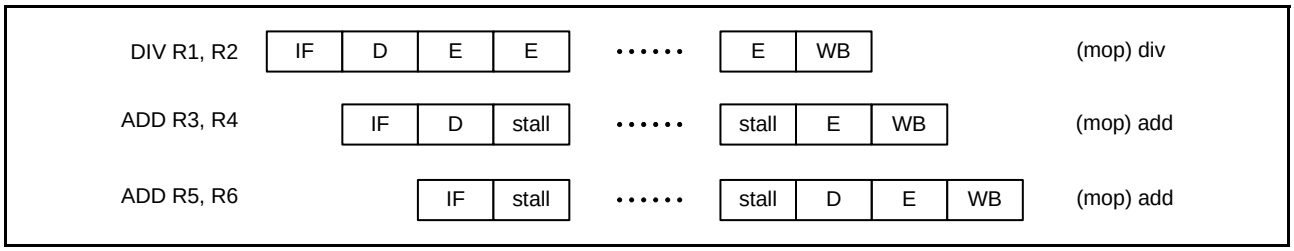


Figure 2.16 When an Instruction which Requires Multiple Cycles is Executed in the E Stage

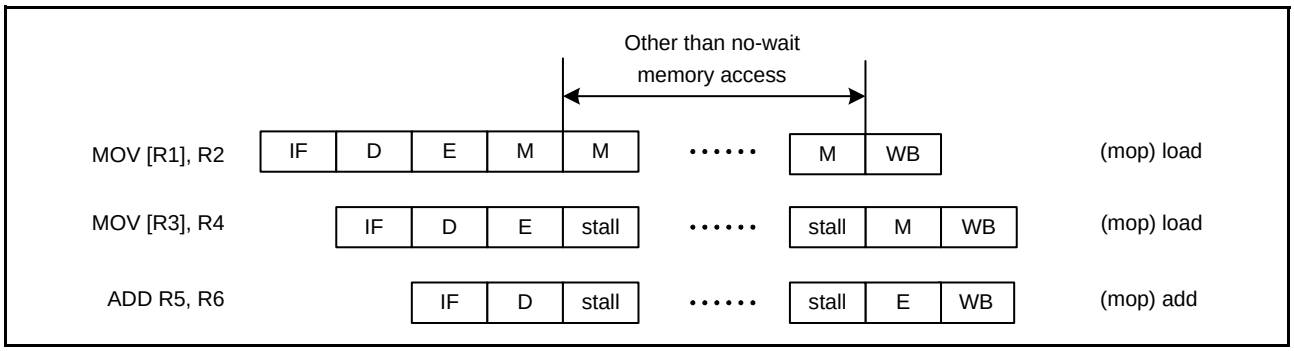


Figure 2.17 When an Instruction which Requires more than One Cycle for its Operand Access is Executed

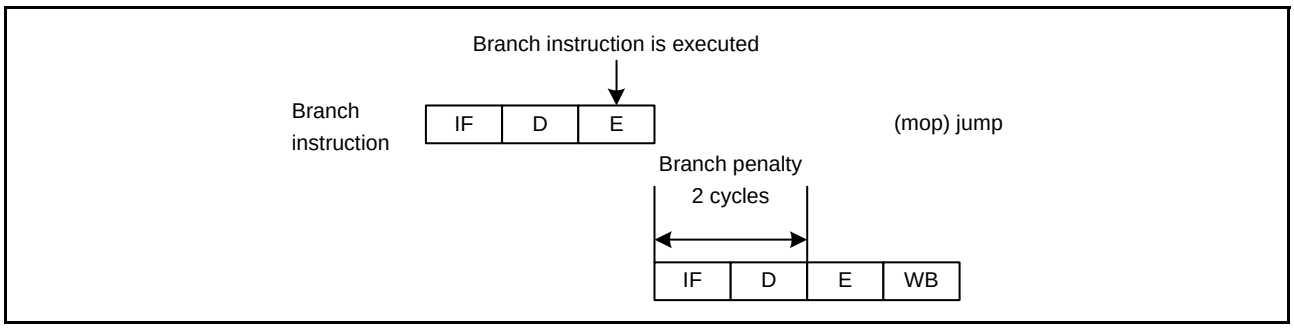


Figure 2.18 When a Branch Instruction is Executed (an Unconditional Branch Instruction is Executed or the Condition is Satisfied for a Conditional Branch Instruction)

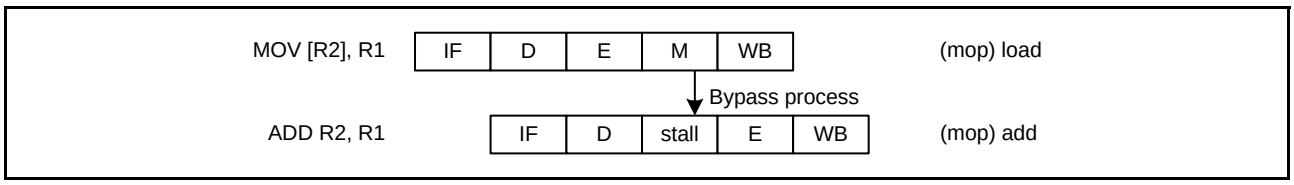


Figure 2.19 When the Subsequent Instruction Uses an Operand Read from the Memory

(2) Pipeline Flow with no Stall

(a) Bypass process

Even when the result of the preceding instruction will be used in a subsequent instruction, the operation processing between registers is pipelined in by the bypass process.

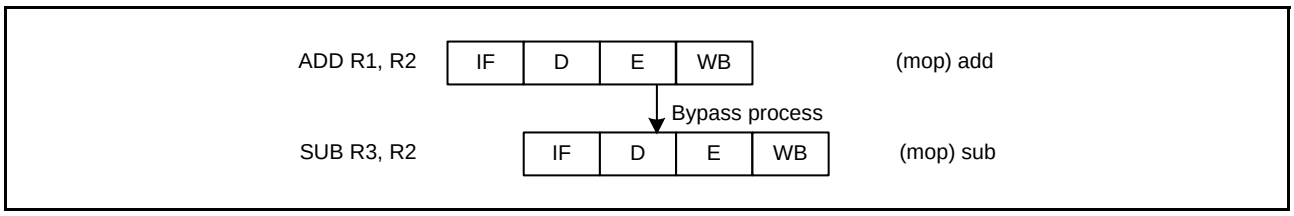


Figure 2.20 Bypass Process

(b) When WB stages for the memory load and for the operation are overlapped

Even when the WB stages for the memory load and for the operation are overlapped, the operation processing is pipelined in, because the load data and the operation result can be written to the register at the same timing.

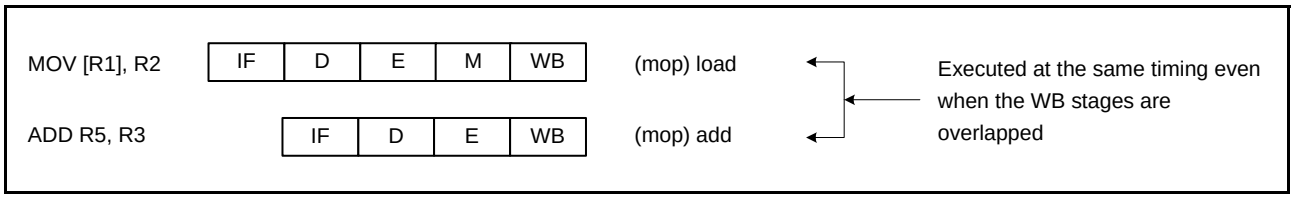


Figure 2.21 When WB Stages for the Memory Load and for the Operation are Overlapped

(c) When subsequent instruction writes to the same register before the end of memory load

Even when the subsequent instruction writes to the same register before the end of memory load, the operation processing is pipelined in, because the WB stage for the memory load is canceled.

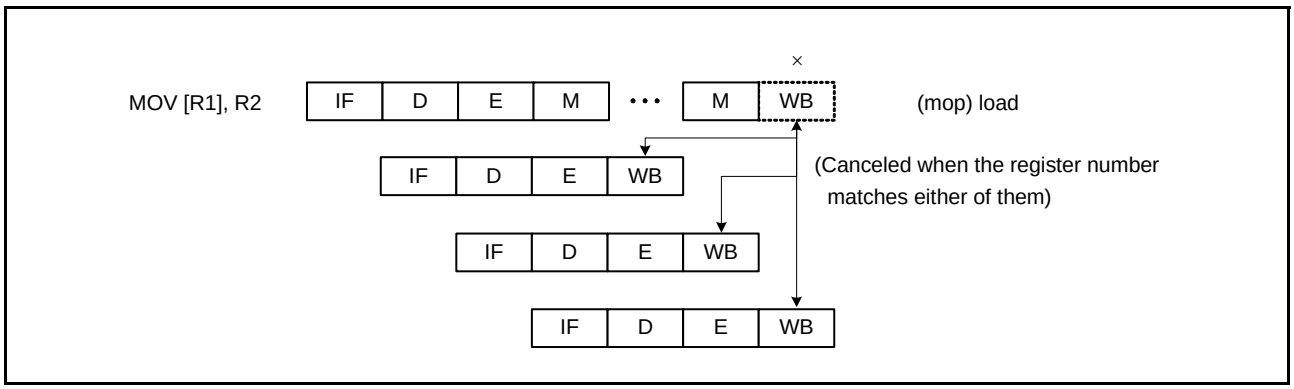


Figure 2.22 When Subsequent Instruction Writes to the Same Register before the End of Memory Load

(d) When the load data is not used by the subsequent instruction

When the load data is not used by the subsequent instruction, the subsequent operations are in fact executed earlier and the operation processing ends (out-of-order completion).

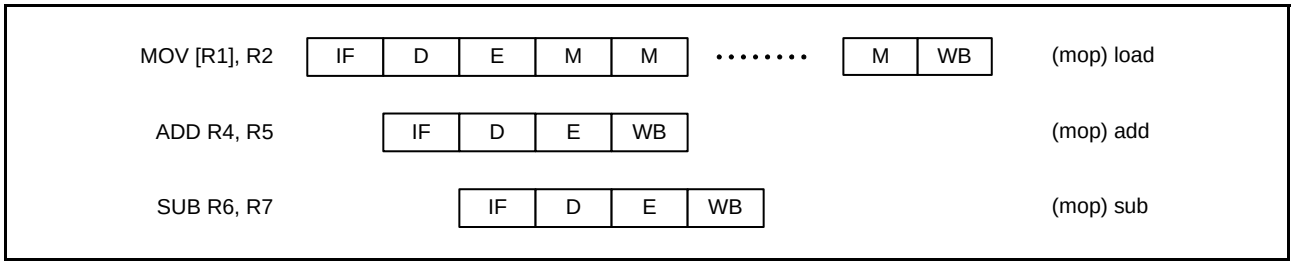


Figure 2.23 When Load Data is not Used by the Subsequent Instruction

2.8.3 Calculation of the Instruction Processing Time

Though the instruction processing time of the CPU varies according to the pipeline processing, the approximate time can be calculated in the following methods.

- Count the number of cycles (see Table 2.13 and Table 2.14)
- When the load data is used by the subsequent instruction, the number of cycles described as “latency” is counted as the number of cycles for the memory load instruction. For the cycles other than the memory load instruction, the number of cycles described as “throughput” is counted.
- If the instruction fetch stall is generated, the number of cycles increments.
- Depending on the system configuration, multiple cycles are required for the memory access.

2.8.4 Numbers of Cycles for Response to Interrupts

Table 2.15 lists numbers of cycles taken by processing for response to interrupts.

Table 2.15 Numbers of Cycles for Response to Interrupts

Type of Interrupt Request/Details of Processing	Fast Interrupt	Other Interrupts
ICU Judgment of priority order	2 cycles	
CPU Number of cycles from notification to acceptance of the interrupt request	N cycles (varies with the instruction being executed at the time the interrupt was received)	
CPU Pre-processing by hardware Saving the current PC and PSW values in RAM (or in control registers in the case of the fast interrupt) Reading of the vector Branching to the start of the exception handling routine	4 cycles	6 cycles

Times calculated from the values in Table 2.15 will be applicable when access to memory from the CPU is processed with no waiting. The ROM and RAM in products of this MCU allow such access. Numbers of cycles for response to interrupts can be minimized by placing program code (and vectors) in ROM and the stack in RAM. Furthermore, place the addresses where the exception handling routine start on 8-byte boundaries.

For information on the number of cycles from notification to acceptance of the interrupt request, indicated by N in the table above, see Table 2.13, Instructions that are Converted into a Single Micro-Operation, and Table 2.14, Instructions that are Converted into Multiple Micro-Operations.

The timing of interrupt acceptance depends on the state of the pipelines. For more information on this, refer to section 13.3.1, Acceptance Timing and Saved PC Value.

3. Operating Modes

3.1 Operating Mode Types and Selection

There are two types of operating-mode selection: one is be selected by the level on pins at the time of release from the reset state, and the other is selected by software after release from the reset state.
Table 3.1 shows the relationship between levels on the mode-setting pins (MD) on release from the reset state and the operating mode selected at that time. For details on each of the operating modes, see section 3.3, Details of Operating Modes.

Table 3.1 Selection of Operating Modes by the Mode-Setting Pins

Mode-Setting Pin	
MD*1	**Operating Mode**
Low	Boot mode (SCI)
High	Single-chip mode

Note 1. Do not change the level on the MD pin while the MCU is operating.

The endian is selectable in single-chip mode. Endian is set by the MDE.MDE[2:0] bits in the option-setting memory. For the correspondence between the setting and endian, see Table 3.2.

Table 3.2 Selection of Endian

MDE.MDE[2:0] Bit Setting	Endian
000b	Big endian
111b	Little endian

3.2 Register Descriptions

3.2.1 Mode Monitor Register (MDMONR)

Address(es): 0008 0000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MD
Value after reset:	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0	0/1*1

Note 1. This affects the level on the MD pin at the time of release from the reset state.

Bit	Symbol	Bit Name	Description	R/W
b0	MD	MD Pin Status Flag	0: The MD pin is low. 1: The MD pin is high.	R
b7 to b1	—	Reserved	These bits are read as 0.	R
b8	—	Reserved	The read value is undefined.	R
b15 to b9	—	Reserved	These bits are read as 0.	R

3.2.2 System Control Register 1 (SYSCR1)

Address(es): 0008 0008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	RAME
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	RAME	RAM Enable	0: The RAM is disabled. 1: The RAM is enabled.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

RAME Bit (RAM Enable)

The RAME bit enables or disables the RAM.

A 0 should not be written to this bit during access to the RAM. When accessing the RAM immediately after changing the RAME bit from 0 (RAM disabled) to 1 (RAM enabled), make sure that the RAME bit is 1 before the access.

3.3 Details of Operating Modes

3.3.1 Single-Chip Mode

In this mode, all I/O ports can be used as general input/output ports, peripheral function input/output, or interrupt input pins.
The chip starts up in single-chip mode if the high level is on the MD pin on release from the reset state.

3.3.2 Boot Mode

In this mode, the on-chip flash memory modifying program (boot program) stored in a dedicated area within the MCU operates. The on-chip flash memory (ROM) can be modified from outside the MCU by using a universal asynchronous receiver/transmitter (SCI1). For details, see section 31, Flash Memory (FLASH).
When a reset is released while the MD pin is low, boot mode is selected.

3.3.2.1 Boot Mode (SCI)

When a reset is released while the MD pin is low, boot mode (SCI) is selected. For details on boot mode (SCI), refer to section 31.8.1, Boot Mode (SCI Interface).

3.4 Transitions of Operating Modes

3.4.1 Operating Mode Transitions Determined by the Mode-Setting Pins

Figure 3.1 shows operating mode transitions determined by the settings of the MD pin.

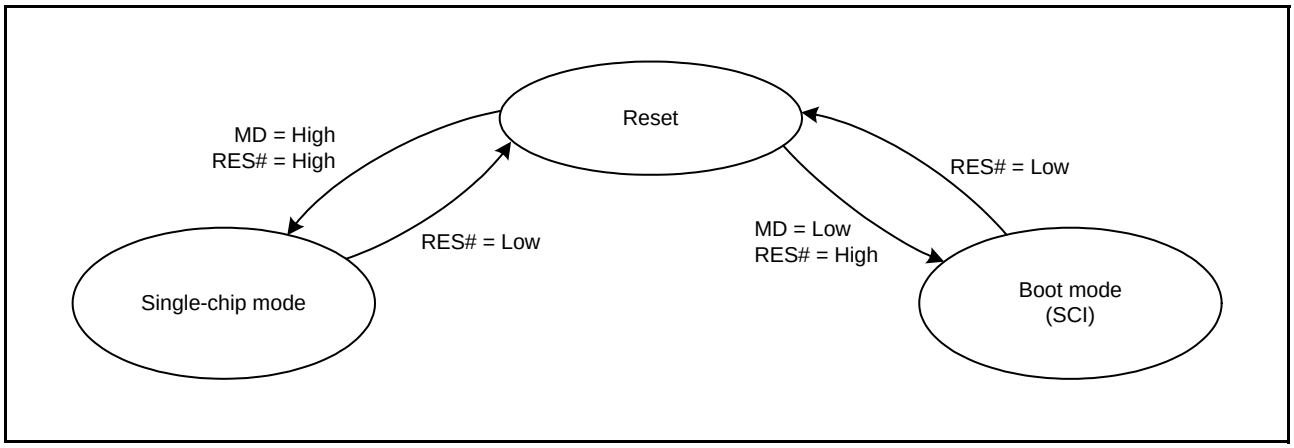
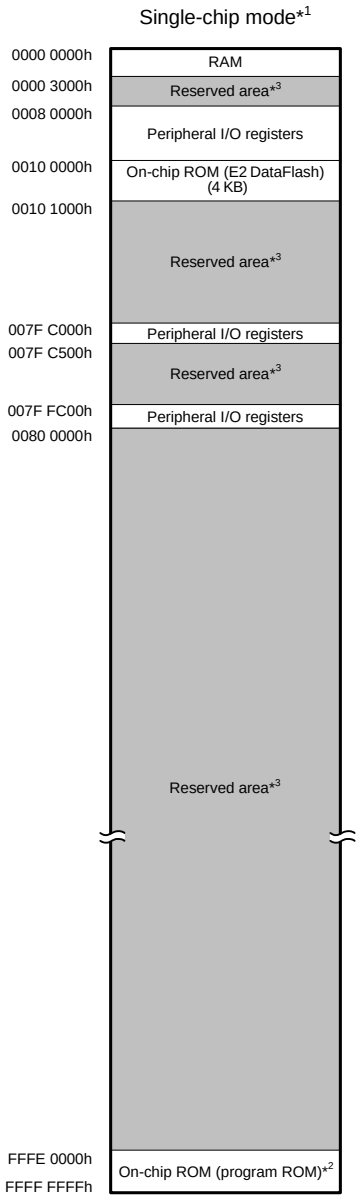


Figure 3.1 Mode-Setting Pin Levels and Operating Modes

4. Address Space

4.1 Address Space

This MCU has a 4-Gbyte address space, consisting of the range of addresses from 0000 0000h to FFFF FFFFh. That is, linear access to an address space of up to 4 Gbytes is possible, and this contains both program and data areas.
Figure 4.1 shows the memory maps.



Note 1. The address space in boot mode is the same as the address space in single-chip mode.
Note 2. The capacity of ROM differs depending on the products.

ROM (bytes)		RAM (bytes)	
Capacity	**Address**	**Capacity**	**Address**
128 K	FFFE 0000h to FFFF FFFFh	12 K	0000 0000h to 0000 2FFFh
64 K	FFFF 0000h to FFFF FFFFh		

Note: See Table 1.3 List of Products, for the product type name.

Note 3. Reserved areas should not be accessed.

Figure 4.1 Memory Map in Each Operating Mode

5. I/O Registers

This section provides information on the on-chip I/O register addresses and bit configuration. The information is given as shown below. Notes on writing to registers are also given below.

(1) I/O register addresses (address order)

- Registers are listed from the lower allocation addresses.
- Registers are classified according to module symbols.
- Numbers of cycles for access indicate numbers of cycles of the given base clock.
- Among the internal I/O register area, addresses not listed in the list of registers are reserved. Reserved addresses must not be accessed. Do not access these addresses; otherwise, the operation when accessing these bits and subsequent operations cannot be guaranteed.

(2) Notes on writing to I/O registers

When writing to an I/O register, the CPU starts executing the subsequent instruction before completing I/O register write. This may cause the subsequent instruction to be executed before the post-update I/O register value is reflected on the operation.
As described in the following examples, special care is required for the cases in which the subsequent instruction must be executed after the post-update I/O register value is actually reflected.

[Examples of cases requiring special care]

- The subsequent instruction must be executed while an interrupt request is disabled with the IENj bit in IERn of the ICU (interrupt request enable bit) cleared to 0.
- A WAIT instruction is executed immediately after the preprocessing for causing a transition to the low power consumption state.

In the above cases, after writing to an I/O register, wait until the write operation is completed using the following procedure and then execute the subsequent instruction.

(a) Write to an I/O register.
(b) Read the value from the I/O register to a general register.
(c) Execute the operation using the value read.
(d) Execute the subsequent instruction.

[Instruction examples]

- Byte-size I/O registers

```
MOV.L  #SFR_ADDR, R1
MOV.B  #SFR_DATA, [R1]
CMP    [R1].UB, R1
;; Next process
```

- Word-size I/O registers

```
MOV.L  #SFR_ADDR, R1
MOV.W  #SFR_DATA, [R1]
CMP    [R1].W, R1
;; Next process
```

- Longword-size I/O registers

```
MOV.L  #SFR_ADDR, R1
MOV.L  #SFR_DATA, [R1]
CMP    [R1].L, R1
;; Next process
```

If multiple registers are written to and a subsequent instruction should be executed after the write operations are entirely completed, only read the I/O register that was last written to and execute the operation using the value; it is not necessary to read or execute operation for all the registers that were written to.

(3) Number of Access Cycles to I/O Registers

For numbers of clock cycles for access to I/O registers, see Table 5.1, List of I/O Registers (Address Order).
The number of access cycles to I/O registers is obtained by following equation.*1

Number of access cycles to I/O registers = Number of bus cycles for internal main bus 1 +
Number of divided clock synchronization cycles +
Number of bus cycles for internal peripheral buses 1 to 3, and 6

The number of bus cycles of internal peripheral buses 1 to 3, and 6 differs according to the register to be accessed.
When the registers for peripheral functions connected to internal peripheral buses 2, 3, and 6 (except for bus error related registers) are accessed, the number of divided clock synchronization cycles is added.
The number of divided clock synchronization cycles differs depending on the frequency ratio between ICLK and PCLK (or FCLK) or bus access timing.
In the peripheral function unit, when the frequency ratio of ICLK is equal to or greater than that of PCLK (or FCLK), the sum of the number of bus cycles for internal main bus 1 and the number of the divided clock synchronization cycles will be one cycle of PCLK (or FCLK) at a maximum. Therefore, one PCLK (or FCLK) has been added to the number of access cycles shown in Table 5.1.
When the frequency ratio of ICLK is lower than that of PCLK (or FCLK), the subsequent bus access is started from the ICLK cycle following the completion of the access to the peripheral functions. Therefore, the access cycles are described on an ICLK basis.

Note 1. This applies to the number of cycles when the access from the CPU does not conflict with the instruction fetching to the external memory or bus access from the different bus master (DTC).

(4) Restrictions in Relation to RMPA and String-Manipulation Instructions

The allocation of data to be handled by RMPA or string-manipulation instructions to I/O registers is prohibited, and operation is not guaranteed if this restriction is not observed.

(5) Notes on Sleep Mode and Mode Transitions

During sleep mode or mode transitions, do not write to the system control related registers (indicated by 'SYSTEM' in the Module Symbol column in Table 5.1, List of I/O Registers (Address Order)).

5.1 I/O Register Addresses (Address Order)

Table 5.1 List of I/O Registers (Address Order) (1 / 11)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles	Reference Section
0008 0000h	SYSTEM	Mode Monitor Register	MDMONR	16	16	3 ICLK	section 3.
0008 0008h	SYSTEM	System Control Register 1	SYSCR1	16	16	3 ICLK	section 3.
0008 000Ch	SYSTEM	Standby Control Register	SBYCR	16	16	3 ICLK	section 11.
0008 0010h	SYSTEM	Module Stop Control Register A	MSTPCRA	32	32	3 ICLK	section 11.
0008 0014h	SYSTEM	Module Stop Control Register B	MSTPCRB	32	32	3 ICLK	section 11.
0008 0018h	SYSTEM	Module Stop Control Register C	MSTPCRC	32	32	3 ICLK	section 11.
0008 0020h	SYSTEM	System Clock Control Register	SCKCR	32	32	3 ICLK	section 9.
0008 0026h	SYSTEM	System Clock Control Register 3	SCKCR3	16	16	3 ICLK	section 9.
0008 0028h	SYSTEM	PLL Control Register	PLLCR	16	16	3 ICLK	section 9.
0008 002Ah	SYSTEM	PLL Control Register 2	PLLCR2	8	8	3 ICLK	section 9.
0008 0032h	SYSTEM	Main Clock Oscillator Control Register	MOSCCR	8	8	3 ICLK	section 9.
0008 0034h	SYSTEM	Low-Speed On-Chip Oscillator Control Register	LOCOCR	8	8	3 ICLK	section 9.
0008 0035h	SYSTEM	IWDT-Dedicated On-Chip Oscillator Control Register	ILOCOCR	8	8	3 ICLK	section 9.
0008 0036h	SYSTEM	High-Speed On-Chip Oscillator Control Register	HOCOCR	8	8	3 ICLK	section 9.
0008 003Ch	SYSTEM	Oscillation Stabilization Flag Register	OSCOVFSR	8	8	3 ICLK	section 9.
0008 0040h	SYSTEM	Oscillation Stop Detection Control Register	OSTDCR	8	8	3 ICLK	section 9.
0008 0041h	SYSTEM	Oscillation Stop Detection Status Register	OSTDSR	8	8	3 ICLK	section 9.
0008 0060h	SYSTEM	Low-Speed On-Chip Oscillator Trimming Register	LOCOTRR	8	8	3 ICLK	section 9.
0008 0064h	SYSTEM	IWDT-Dedicated On-Chip Oscillator Trimming Register	ILOCOTRR	8	8	3 ICLK	section 9.
0008 0068h	SYSTEM	High-Speed On-Chip Oscillator Trimming Register 0	HOCOTRR0	8	8	3 ICLK	section 9.
0008 00A0h	SYSTEM	Operating Power Control Register	OPCCR	8	8	3 ICLK	section 11.
0008 00A2h	SYSTEM	Main Clock Oscillator Wait Control Register	MOSCWTCR	8	8	3 ICLK	section 9.
0008 00C0h	SYSTEM	Reset Status Register 2	RSTSR2	8	8	3 ICLK	section 6.
0008 00C2h	SYSTEM	Software Reset Register	SWRR	16	16	3 ICLK	section 6.
0008 00E0h	SYSTEM	Voltage Monitoring 1 Circuit Control Register 1	LVD1CR1	8	8	3 ICLK	section 8.
0008 00E1h	SYSTEM	Voltage Monitoring 1 Circuit Status Register	LVD1SR	8	8	3 ICLK	section 8.
0008 00E2h	SYSTEM	Voltage Monitoring 2 Circuit Control Register 1	LVD2CR1	8	8	3 ICLK	section 8.
0008 00E3h	SYSTEM	Voltage Monitoring 2 Circuit Status Register	LVD2SR	8	8	3 ICLK	section 8.
0008 03FEh	SYSTEM	Protect Register	PRCR	16	16	3 ICLK	section 12.
0008 1300h	BSC	Bus Error Status Clear Register	BERCLR	8	8	2 ICLK	section 15.
0008 1304h	BSC	Bus Error Monitoring Enable Register	BEREN	8	8	2 ICLK	section 15.
0008 1308h	BSC	Bus Error Status Register 1	BERSR1	8	8	2 ICLK	section 15.
0008 130Ah	BSC	Bus Error Status Register 2	BERSR2	16	16	2 ICLK	section 15.
0008 1310h	BSC	Bus Priority Control Register	BUSPRI	16	16	2 ICLK	section 15.
0008 2400h	DTC	DTC Control Register	DTCCR	8	8	2 ICLK	section 16.
0008 2404h	DTC	DTC Vector Base Register	DTCVBR	32	32	2 ICLK	section 16.
0008 2408h	DTC	DTC Address Mode Register	DTCADMOD	8	8	2 ICLK	section 16.
0008 240Ch	DTC	DTC Module Start Register	DTCST	8	8	2 ICLK	section 16.
0008 240Eh	DTC	DTC Status Register	DTCSTS	16	16	2 ICLK	section 16.
0008 2410h	DTC	DTC Index Table Base Register	DTCIBR	32	32	2 ICLK	section 16.
0008 2414h	DTC	DTC Operation Register	DTCOR	8	8	2 ICLK	section 16.
0008 2416h	DTC	DTC Sequence Transfer Enable Register	DTCSQE	16	16	2 ICLK	section 16.
0008 2418h	DTC	DTC Address Displacement Register	DTCDISP	32	32	2 ICLK	section 16.
0008 7010h to 0008 70FFh	ICU	Interrupt Request Register 016 to Interrupt Request Register 255	IR016 to IR255	8	8	2 ICLK	section 14.
0008 711Bh to 0008 71FFh	ICU	DTC Transfer Request Enable Register 027 to DTC Transfer Request Enable Register 255	DTCER027 to DTCER255	8	8	2 ICLK	section 14.
0008 7202h to 0008 721Fh	ICU	Interrupt Request Enable Register 02 to Interrupt Request Enable Register 1F	IER02 to IER1F	8	8	2 ICLK	section 14.
0008 72E0h	ICU	Software Interrupt Generation Register	SWINTR	8	8	2 ICLK	section 14.
0008 72F0h	ICU	Fast Interrupt Set Register	FIR	16	16	2 ICLK	section 14.

Table 5.1 List of I/O Registers (Address Order) (2 / 11)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles	Reference Section
0008 7300h to 0008 73FFh	ICU	Interrupt Source Priority Register 000 to Interrupt Source Priority Register 255	IPR000 to IPR255	8	8	2 ICLK	section 14.
0008 7500h to 0008 7505h	ICU	IRQ Control Register 0 to IRQ Control Register 5	IRQCR0 to IRQCR5	8	8	2 ICLK	section 14.
0008 7510h	ICU	IRQ Pin Digital Filter Enable Register 0	IRQFLTE0	8	8	2 ICLK	section 14.
0008 7514h	ICU	IRQ Pin Digital Filter Setting Register 0	IRQFLTC0	16	16	2 ICLK	section 14.
0008 7580h	ICU	Non-Maskable Interrupt Status Register	NMISR	8	8	2 ICLK	section 14.
0008 7581h	ICU	Non-Maskable Interrupt Enable Register	NMIER	8	8	2 ICLK	section 14.
0008 7582h	ICU	Non-Maskable Interrupt Status Clear Register	NMICLR	8	8	2 ICLK	section 14.
0008 7583h	ICU	NMI Pin Interrupt Control Register	NMICR	8	8	2 ICLK	section 14.
0008 7590h	ICU	NMI Pin Digital Filter Enable Register	NMIFLTE	8	8	2 ICLK	section 14.
0008 7594h	ICU	NMI Pin Digital Filter Setting Register	NMIFLTC	8	8	2 ICLK	section 14.
0008 8000h	CMT	Compare Match Timer Start Register 0	CMSTR0	16	16	2 or 3 PCLKB	section 21.
0008 8002h	CMT0	Compare Match Timer Control Register	CMCR	16	16	2 or 3 PCLKB	section 21.
0008 8004h	CMT0	Compare Match Counter	CMCNT	16	16	2 or 3 PCLKB	section 21.
0008 8006h	CMT0	Compare Match Constant Register	CMCOR	16	16	2 or 3 PCLKB	section 21.
0008 8008h	CMT1	Compare Match Timer Control Register	CMCR	16	16	2 or 3 PCLKB	section 21.
0008 800Ah	CMT1	Compare Match Counter	CMCNT	16	16	2 or 3 PCLKB	section 21.
0008 800Ch	CMT1	Compare Match Constant Register	CMCOR	16	16	2 or 3 PCLKB	section 21.
0008 8030h	IWDT	IWDT Refresh Register	IWDTRR	8	8	2 or 3 PCLKB	section 22.
0008 8032h	IWDT	IWDT Control Register	IWDTCR	16	16	2 or 3 PCLKB	section 22.
0008 8034h	IWDT	IWDT Status Register	IWDTSR	16	16	2 or 3 PCLKB	section 22.
0008 8036h	IWDT	IWDT Reset Control Register	IWDTRCR	8	8	2 or 3 PCLKB	section 22.
0008 8038h	IWDT	IWDT Count Stop Control Register	IWDTCSTPR	8	8	2 or 3 PCLKB	section 22.
0008 80C0h	DA	D/A Data Register 0	DADR0	16	16	2 or 3 PCLKB	section 27.
0008 80C4h	DA	D/A Control Register	DACR	8	8	2 or 3 PCLKB	section 27.
0008 80C5h	DA	Data Register Format Select Register	DADPR	8	8	2 or 3 PCLKB	section 27.
0008 8280h	CRC	CRC Control Register	CRCCR	8	8	2 or 3 PCLKB	section 25.
0008 8281h	CRC	CRC Data Input Register	CRCDIR	8	8	2 or 3 PCLKB	section 25.
0008 8282h	CRC	CRC Data Output Register	CRCDOR	16	16	2 or 3 PCLKB	section 25.
0008 8300h	RIIC0	I^2C-bus Control Register 1	ICCR1	8	8	2 or 3 PCLKB	section 24.
0008 8301h	RIIC0	I^2C-bus Control Register 2	ICCR2	8	8	2 or 3 PCLKB	section 24.
0008 8302h	RIIC0	I^2C-bus Mode Register 1	ICMR1	8	8	2 or 3 PCLKB	section 24.
0008 8303h	RIIC0	I^2C-bus Mode Register 2	ICMR2	8	8	2 or 3 PCLKB	section 24.
0008 8304h	RIIC0	I^2C-bus Mode Register 3	ICMR3	8	8	2 or 3 PCLKB	section 24.
0008 8305h	RIIC0	I^2C-bus Function Enable Register	ICFER	8	8	2 or 3 PCLKB	section 24.
0008 8306h	RIIC0	I^2C-bus Status Enable Register	ICSER	8	8	2 or 3 PCLKB	section 24.
0008 8307h	RIIC0	I^2C-bus Interrupt Enable Register	ICIER	8	8	2 or 3 PCLKB	section 24.
0008 8308h	RIIC0	I^2C-bus Status Register 1	ICSR1	8	8	2 or 3 PCLKB	section 24.
0008 8309h	RIIC0	I^2C-bus Status Register 2	ICSR2	8	8	2 or 3 PCLKB	section 24.
0008 830Ah	RIIC0	Slave Address Register L0	SARL0	8	8	2 or 3 PCLKB	section 24.
0008 830Bh	RIIC0	Slave Address Register U0	SARU0	8	8	2 or 3 PCLKB	section 24.
0008 830Ch	RIIC0	Slave Address Register L1	SARL1	8	8	2 or 3 PCLKB	section 24.
0008 830Dh	RIIC0	Slave Address Register U1	SARU1	8	8	2 or 3 PCLKB	section 24.
0008 830Eh	RIIC0	Slave Address Register L2	SARL2	8	8	2 or 3 PCLKB	section 24.
0008 830Fh	RIIC0	Slave Address Register U2	SARU2	8	8	2 or 3 PCLKB	section 24.
0008 8310h	RIIC0	I^2C-bus Bit Rate Low-Level Register	ICBRL	8	8	2 or 3 PCLKB	section 24.
0008 8311h	RIIC0	I^2C-bus Bit Rate High-Level Register	ICBRH	8	8	2 or 3 PCLKB	section 24.
0008 8312h	RIIC0	I^2C-bus Transmit Data Register	ICDRT	8	8	2 or 3 PCLKB	section 24.
0008 8313h	RIIC0	I^2C-bus Receive Data Register	ICDRR	8	8	2 or 3 PCLKB	section 24.
0008 9000h	S12AD	A/D Control Register	ADCSR	16	16	2 or 3 PCLKB	section 26.
0008 9004h	S12AD	A/D Channel Select Register A0	ADANSA0	16	16	2 or 3 PCLKB	section 26.
0008 9008h	S12AD	A/D-Converted Value Addition/Average Function Channel Select Register 0	ADADS0	16	16	2 or 3 PCLKB	section 26.

Table 5.1 List of I/O Registers (Address Order) (3 / 11)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles	Reference Section
0008 900Ch	S12AD	A/D-Converted Value Addition/Average Count Select Register	ADADC	8	8	2 or 3 PCLKB	section 26.
0008 900Eh	S12AD	A/D Control Extended Register	ADCER	16	16	2 or 3 PCLKB	section 26.
0008 9010h	S12AD	A/D Conversion Start Trigger Select Register	ADSTRGR	16	16	2 or 3 PCLKB	section 26.
0008 9012h	S12AD	A/D Conversion Extended Input Control Register	ADEXICR	16	16	2 or 3 PCLKB	section 26.
0008 9014h	S12AD	A/D Channel Select Register B0	ADANSB0	16	16	2 or 3 PCLKB	section 26.
0008 9018h	S12AD	A/D Data Duplication Register	ADDBLDR	16	16	2 or 3 PCLKB	section 26.
0008 901Ch	S12AD	A/D Internal Reference Voltage Data Register	ADOCDR	16	16	2 or 3 PCLKB	section 26.
0008 901Eh	S12AD	A/D Self-Diagnosis Data Register	ADRD	16	16	2 or 3 PCLKB	section 26.
0008 9020h	S12AD	A/D Data Register 0	ADDR0	16	16	2 or 3 PCLKB	section 26.
0008 9022h	S12AD	A/D Data Register 1	ADDR1	16	16	2 or 3 PCLKB	section 26.
0008 9024h	S12AD	A/D Data Register 2	ADDR2	16	16	2 or 3 PCLKB	section 26.
0008 9026h	S12AD	A/D Data Register 3	ADDR3	16	16	2 or 3 PCLKB	section 26.
0008 9028h	S12AD	A/D Data Register 4	ADDR4	16	16	2 or 3 PCLKB	section 26.
0008 902Ah	S12AD	A/D Data Register 5	ADDR5	16	16	2 or 3 PCLKB	section 26.
0008 902Ch	S12AD	A/D Data Register 6	ADDR6	16	16	2 or 3 PCLKB	section 26.
0008 902Eh	S12AD	A/D Data Register 7	ADDR7	16	16	2 or 3 PCLKB	section 26.
0008 9066h	S12AD	A/D Sample-and-Hold Circuit Control Register	ADSHCR	16	16	2 or 3 PCLKB	section 26.
0008 907Ah	S12AD	A/D Disconnection Detection Control Register	ADDISCR	8	8	2 or 3 PCLKB	section 26.
0008 9080h	S12AD	A/D Group Scan Priority Control Register	ADGSPCR	16	16	2 or 3 PCLKB	section 26.
0008 9084h	S12AD	A/D Data Duplication Register A	ADDBLDRA	16	16	2 or 3 PCLKB	section 26.
0008 9086h	S12AD	A/D Data Duplication Register B	ADDBLDRB	16	16	2 or 3 PCLKB	section 26.
0008 90D4h	S12AD	A/D Channel Select Register C0	ADANSC0	16	16	2 or 3 PCLKB	section 26.
0008 90D9h	S12AD	A/D Group C Trigger Select Register	ADGCTRGR	8	8	2 or 3 PCLKB	section 26.
0008 90DFh	S12AD	A/D Sampling State Register O	ADSSTRO	8	8	2 or 3 PCLKB	section 26.
0008 90E0h	S12AD	A/D Sampling State Register 0	ADSSTR0	8	8	2 or 3 PCLKB	section 26.
0008 90E1h	S12AD	A/D Sampling State Register 1	ADSSTR1	8	8	2 or 3 PCLKB	section 26.
0008 90E2h	S12AD	A/D Sampling State Register 2	ADSSTR2	8	8	2 or 3 PCLKB	section 26.
0008 90E3h	S12AD	A/D Sampling State Register 3	ADSSTR3	8	8	2 or 3 PCLKB	section 26.
0008 90E4h	S12AD	A/D Sampling State Register 4	ADSSTR4	8	8	2 or 3 PCLKB	section 26.
0008 90E5h	S12AD	A/D Sampling State Register 5	ADSSTR5	8	8	2 or 3 PCLKB	section 26.
0008 90E6h	S12AD	A/D Sampling State Register 6	ADSSTR6	8	8	2 or 3 PCLKB	section 26.
0008 90E7h	S12AD	A/D Sampling State Register 7	ADSSTR7	8	8	2 or 3 PCLKB	section 26.
0008 91A0h	S12AD	A/D Programmable Gain Amplifier Control Register	ADPGACR	16	16	2 or 3 PCLKB	section 26.
0008 91A2h	S12AD	A/D Programmable Gain Amplifier Gain Setting Register 0	ADPGAGS0	16	16	2 or 3 PCLKB	section 26.
0008 A020h	SCI1	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	section 23.
0008 A020h	SMCI1	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	section 23.
0008 A021h	SCI1	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	section 23.
0008 A022h	SCI1	Serial Control Register	SCR	8	8	2 or 3 PCLKB	section 23.
0008 A022h	SMCI1	Serial Control Register	SCR	8	8	2 or 3 PCLKB	section 23.
0008 A023h	SCI1	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	section 23.
0008 A024h	SCI1	Serial Status Register	SSR	8	8	2 or 3 PCLKB	section 23.
0008 A024h	SMCI1	Serial Status Register	SSR	8	8	2 or 3 PCLKB	section 23.
0008 A025h	SCI1	Receive Data Register	RDR	8	8	2 or 3 PCLKB	section 23.
0008 A026h	SCI1	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	section 23.
0008 A026h	SMCI1	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	section 23.
0008 A027h	SCI1	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	section 23.
0008 A028h	SCI1	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	section 23.
0008 A029h	SCI1	I^2C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	section 23.
0008 A02Ah	SCI1	I^2C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	section 23.
0008 A02Bh	SCI1	I^2C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	section 23.
0008 A02Ch	SCI1	I^2C Status Register	SISR	8	8	2 or 3 PCLKB	section 23.
0008 A02Dh	SCI1	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	section 23.

Table 5.1 List of I/O Registers (Address Order) (4 / 11)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles	Reference Section
0008 A02Eh	SCI1	Transmit Data Register HL	TDRHL	16	16	2 or 3 PCLKB	section 23.
0008 A02Eh	SCI1	Transmit Data Register H	TDRH	8	8	2 or 3 PCLKB	section 23.
0008 A02Fh	SCI1	Transmit Data Register L	TDRL	8	8	2 or 3 PCLKB	section 23.
0008 A030h	SCI1	Receive Data Register HL	RDRHL	16	16	2 or 3 PCLKB	section 23.
0008 A030h	SCI1	Receive Data Register H	RDRH	8	8	2 or 3 PCLKB	section 23.
0008 A031h	SCI1	Receive Data Register L	RDRL	8	8	2 or 3 PCLKB	section 23.
0008 A032h	SCI1	Modulation Duty Register	MDDR	8	8	2 or 3 PCLKB	section 23.
0008 A0A0h	SCI5	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	section 23.
0008 A0A0h	SMCI5	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	section 23.
0008 A0A1h	SCI5	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	section 23.
0008 A0A2h	SCI5	Serial Control Register	SCR	8	8	2 or 3 PCLKB	section 23.
0008 A0A2h	SMCI5	Serial Control Register	SCR	8	8	2 or 3 PCLKB	section 23.
0008 A0A3h	SCI5	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	section 23.
0008 A0A4h	SCI5	Serial Status Register	SSR	8	8	2 or 3 PCLKB	section 23.
0008 A0A4h	SMCI5	Serial Status Register	SSR	8	8	2 or 3 PCLKB	section 23.
0008 A0A5h	SCI5	Receive Data Register	RDR	8	8	2 or 3 PCLKB	section 23.
0008 A0A6h	SCI5	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	section 23.
0008 A0A6h	SMCI5	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	section 23.
0008 A0A7h	SCI5	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	section 23.
0008 A0A8h	SCI5	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	section 23.
0008 A0A9h	SCI5	I^2C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	section 23.
0008 A0AAh	SCI5	I^2C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	section 23.
0008 A0ABh	SCI5	I^2C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	section 23.
0008 A0ACh	SCI5	I^2C Status Register	SISR	8	8	2 or 3 PCLKB	section 23.
0008 A0ADh	SCI5	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	section 23.
0008 A0AEh	SCI5	Transmit Data Register HL	TDRHL	16	16	2 or 3 PCLKB	section 23.
0008 A0AEh	SCI5	Transmit Data Register H	TDRH	8	8	2 or 3 PCLKB	section 23.
0008 A0AFh	SCI5	Transmit Data Register L	TDRL	8	8	2 or 3 PCLKB	section 23.
0008 A0B0h	SCI5	Receive Data Register HL	RDRHL	16	16	2 or 3 PCLKB	section 23.
0008 A0B0h	SCI5	Receive Data Register H	RDRH	8	8	2 or 3 PCLKB	section 23.
0008 A0B1h	SCI5	Receive Data Register L	RDRL	8	8	2 or 3 PCLKB	section 23.
0008 A0B2h	SCI5	Modulation Duty Register	MDDR	8	8	2 or 3 PCLKB	section 23.
0008 B000h	CAC	CAC Control Register 0	CACR0	8	8	2 or 3 PCLKB	section 10.
0008 B001h	CAC	CAC Control Register 1	CACR1	8	8	2 or 3 PCLKB	section 10.
0008 B002h	CAC	CAC Control Register 2	CACR2	8	8	2 or 3 PCLKB	section 10.
0008 B003h	CAC	CAC Interrupt Request Enable Register	CAICR	8	8	2 or 3 PCLKB	section 10.
0008 B004h	CAC	CAC Status Register	CASTR	8	8	2 or 3 PCLKB	section 10.
0008 B006h	CAC	CAC Upper-Limit Value Setting Register	CAULVR	16	16	2 or 3 PCLKB	section 10.
0008 B008h	CAC	CAC Lower-Limit Value Setting Register	CALLVR	16	16	2 or 3 PCLKB	section 10.
0008 B00Ah	CAC	CAC Counter Buffer Register	CACNTBR	16	16	2 or 3 PCLKB	section 10.
0008 B080h	DOC	DOC Control Register	DOCR	8	8	2 or 3 PCLKB	section 29.
0008 B082h	DOC	DOC Data Input Register	DODIR	16	16	2 or 3 PCLKB	section 29.
0008 B084h	DOC	DOC Data Setting Register	DODSR	16	16	2 or 3 PCLKB	section 29.
0008 B300h	SCI12	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	section 23.
0008 B300h	SMCI12	Serial Mode Register	SMR	8	8	2 or 3 PCLKB	section 23.
0008 B301h	SCI12	Bit Rate Register	BRR	8	8	2 or 3 PCLKB	section 23.
0008 B302h	SCI12	Serial Control Register	SCR	8	8	2 or 3 PCLKB	section 23.
0008 B302h	SMCI12	Serial Control Register	SCR	8	8	2 or 3 PCLKB	section 23.
0008 B303h	SCI12	Transmit Data Register	TDR	8	8	2 or 3 PCLKB	section 23.
0008 B304h	SCI12	Serial Status Register	SSR	8	8	2 or 3 PCLKB	section 23.
0008 B304h	SMCI12	Serial Status Register	SSR	8	8	2 or 3 PCLKB	section 23.
0008 B305h	SCI12	Receive Data Register	RDR	8	8	2 or 3 PCLKB	section 23.

Table 5.1 List of I/O Registers (Address Order) (5 / 11)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles	Reference Section
0008 B306h	SCI12	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	section 23.
0008 B306h	SMCI12	Smart Card Mode Register	SCMR	8	8	2 or 3 PCLKB	section 23.
0008 B307h	SCI12	Serial Extended Mode Register	SEMR	8	8	2 or 3 PCLKB	section 23.
0008 B308h	SCI12	Noise Filter Setting Register	SNFR	8	8	2 or 3 PCLKB	section 23.
0008 B309h	SCI12	I²C Mode Register 1	SIMR1	8	8	2 or 3 PCLKB	section 23.
0008 B30Ah	SCI12	I²C Mode Register 2	SIMR2	8	8	2 or 3 PCLKB	section 23.
0008 B30Bh	SCI12	I²C Mode Register 3	SIMR3	8	8	2 or 3 PCLKB	section 23.
0008 B30Ch	SCI12	I²C Status Register	SISR	8	8	2 or 3 PCLKB	section 23.
0008 B30Dh	SCI12	SPI Mode Register	SPMR	8	8	2 or 3 PCLKB	section 23.
0008 B30Eh	SCI12	Transmit Data Register HL	TDRHL	16	16	2 or 3 PCLKB	section 23.
0008 B30Eh	SCI12	Transmit Data Register H	TDRH	8	8	2 or 3 PCLKB	section 23.
0008 B30Fh	SCI12	Transmit Data Register L	TDRL	8	8	2 or 3 PCLKB	section 23.
0008 B310h	SCI12	Receive Data Register HL	RDRHL	16	16	2 or 3 PCLKB	section 23.
0008 B310h	SCI12	Receive Data Register H	RDRH	8	8	2 or 3 PCLKB	section 23.
0008 B311h	SCI12	Receive Data Register L	RDRL	8	8	2 or 3 PCLKB	section 23.
0008 B312h	SCI12	Modulation Duty Register	MDDR	8	8	2 or 3 PCLKB	section 23.
0008 B320h	SCI12	Extended Serial Module Enable Register	ESMER	8	8	2 or 3 PCLKB	section 23.
0008 B321h	SCI12	Control Register 0	CR0	8	8	2 or 3 PCLKB	section 23.
0008 B322h	SCI12	Control Register 1	CR1	8	8	2 or 3 PCLKB	section 23.
0008 B323h	SCI12	Control Register 2	CR2	8	8	2 or 3 PCLKB	section 23.
0008 B324h	SCI12	Control Register 3	CR3	8	8	2 or 3 PCLKB	section 23.
0008 B325h	SCI12	Port Control Register	PCR	8	8	2 or 3 PCLKB	section 23.
0008 B326h	SCI12	Interrupt Control Register	ICR	8	8	2 or 3 PCLKB	section 23.
0008 B327h	SCI12	Status Register	STR	8	8	2 or 3 PCLKB	section 23.
0008 B328h	SCI12	Status Clear Register	STCR	8	8	2 or 3 PCLKB	section 23.
0008 B329h	SCI12	Control Field 0 Data Register	CF0DR	8	8	2 or 3 PCLKB	section 23.
0008 B32Ah	SCI12	Control Field 0 Compare Enable Register	CF0CR	8	8	2 or 3 PCLKB	section 23.
0008 B32Bh	SCI12	Control Field 0 Receive Data Register	CF0RR	8	8	2 or 3 PCLKB	section 23.
0008 B32Ch	SCI12	Primary Control Field 1 Data Register	PCF1DR	8	8	2 or 3 PCLKB	section 23.
0008 B32Dh	SCI12	Secondary Control Field 1 Data Register	SCF1DR	8	8	2 or 3 PCLKB	section 23.
0008 B32Eh	SCI12	Control Field 1 Compare Enable Register	CF1CR	8	8	2 or 3 PCLKB	section 23.
0008 B32Fh	SCI12	Control Field 1 Receive Data Register	CF1RR	8	8	2 or 3 PCLKB	section 23.
0008 B330h	SCI12	Timer Control Register	TCR	8	8	2 or 3 PCLKB	section 23.
0008 B331h	SCI12	Timer Mode Register	TMR	8	8	2 or 3 PCLKB	section 23.
0008 B332h	SCI12	Timer Prescaler Register	TPRE	8	8	2 or 3 PCLKB	section 23.
0008 B333h	SCI12	Timer Count Register	TCNT	8	8	2 or 3 PCLKB	section 23.
0008 C001h	PORT1	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 17.
0008 C002h	PORT2	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 17.
0008 C003h	PORT3	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 17.
0008 C004h	PORT4	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 17.
0008 C007h	PORT7	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 17.
0008 C009h	PORT9	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 17.
0008 C00Ah	PORTA	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 17.
0008 C00Bh	PORTB	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 17.
0008 C00Dh	PORTD	Port Direction Register	PDR	8	8	2 or 3 PCLKB	section 17.
0008 C021h	PORT1	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 17.
0008 C022h	PORT2	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 17.
0008 C023h	PORT3	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 17.
0008 C024h	PORT4	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 17.
0008 C027h	PORT7	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 17.
0008 C029h	PORT9	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 17.
0008 C02Ah	PORTA	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 17.

Table 5.1 List of I/O Registers (Address Order) (6 / 11)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles	Reference Section
0008 C02Bh	PORTB	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 17.
0008 C02Dh	PORTD	Port Output Data Register	PODR	8	8	2 or 3 PCLKB	section 17.
0008 C041h	PORT1	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 17.
0008 C042h	PORT2	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 17.
0008 C043h	PORT3	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 17.
0008 C044h	PORT4	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 17.
0008 C047h	PORT7	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 17.
0008 C049h	PORT9	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 17.
0008 C04Ah	PORTA	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 17.
0008 C04Bh	PORTB	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 17.
0008 C04Dh	PORTD	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 17.
0008 C04Eh	PORTE	Port Input Data Register	PIDR	8	8	3 or 4 PCLKB cycles when reading, 2 or 3 PCLKB cycles when writing	section 17.
0008 C061h	PORT1	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 17.
0008 C062h	PORT2	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 17.
0008 C063h	PORT3	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 17.
0008 C067h	PORT7	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 17.
0008 C069h	PORT9	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 17.
0008 C06Ah	PORTA	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 17.
0008 C06Bh	PORTB	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 17.
0008 C06Dh	PORTD	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 17.
0008 C06Eh	PORTE	Port Mode Register	PMR	8	8	2 or 3 PCLKB	section 17.
0008 C082h	PORT1	Open Drain Control Register 0	ODR0	8	8	2 or 3 PCLKB	section 17.
0008 C084h	PORT2	Open Drain Control Register 0	ODR0	8	8	2 or 3 PCLKB	section 17.
0008 C085h	PORT2	Open Drain Control Register 1	ODR1	8	8	2 or 3 PCLKB	section 17.
0008 C087h	PORT3	Open Drain Control Register 1	ODR1	8	8	2 or 3 PCLKB	section 17.
0008 C08Eh	PORT7	Open Drain Control Register 0	ODR0	8	8	2 or 3 PCLKB	section 17.
0008 C08Fh	PORT7	Open Drain Control Register 1	ODR1	8	8	2 or 3 PCLKB	section 17.
0008 C092h	PORT9	Open Drain Control Register 0	ODR0	8	8	2 or 3 PCLKB	section 17.
0008 C093h	PORT9	Open Drain Control Register 1	ODR1	8	8	2 or 3 PCLKB	section 17.
0008 C094h	PORTA	Open Drain Control Register 0	ODR0	8	8	2 or 3 PCLKB	section 17.
0008 C096h	PORTB	Open Drain Control Register 0	ODR0	8	8	2 or 3 PCLKB	section 17.
0008 C097h	PORTB	Open Drain Control Register 1	ODR1	8	8	2 or 3 PCLKB	section 17.
0008 C09Ah	PORTD	Open Drain Control Register 0	ODR0	8	8	2 or 3 PCLKB	section 17.
0008 C09Bh	PORTD	Open Drain Control Register 1	ODR1	8	8	2 or 3 PCLKB	section 17.
0008 C0C1h	PORT1	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 17.
0008 C0C2h	PORT2	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 17.

Table 5.1 List of I/O Registers (Address Order) (7 / 11)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles	Reference Section
0008 C0C3h	PORT3	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 17.
0008 C0C4h	PORT4	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 17.
0008 C0C7h	PORT7	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 17.
0008 C0C9h	PORT9	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 17.
0008 C0CAh	PORTA	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 17.
0008 C0CBh	PORTB	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 17.
0008 C0CDh	PORTD	Pull-Up Control Register	PCR	8	8	2 or 3 PCLKB	section 17.
0008 C0E1h	PORT1	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	section 17.
0008 C0E2h	PORT2	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	section 17.
0008 C0E7h	PORT7	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	section 17.
0008 C0E9h	PORT9	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	section 17.
0008 C0EAh	PORTA	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	section 17.
0008 C0EBh	PORTB	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	section 17.
0008 C0EDh	PORTD	Drive Capacity Control Register	DSCR	8	8	2 or 3 PCLKB	section 17.
0008 C11Fh	MPC	Write-Protect Register	PWPR	8	8	2 or 3 PCLKB	section 18.
0008 C148h	MPC	P10 Pin Function Control Register	P10PFS	8	8	2 or 3 PCLKB	section 18.
0008 C149h	MPC	P11 Pin Function Control Register	P11PFS	8	8	2 or 3 PCLKB	section 18.
0008 C152h	MPC	P22 Pin Function Control Register	P22PFS	8	8	2 or 3 PCLKB	section 18.
0008 C153h	MPC	P23 Pin Function Control Register	P23PFS	8	8	2 or 3 PCLKB	section 18.
0008 C154h	MPC	P24 Pin Function Control Register	P24PFS	8	8	2 or 3 PCLKB	section 18.
0008 C160h	MPC	P40 Pin Function Control Register	P40PFS	8	8	2 or 3 PCLKB	section 18.
0008 C161h	MPC	P41 Pin Function Control Register	P41PFS	8	8	2 or 3 PCLKB	section 18.
0008 C162h	MPC	P42 Pin Function Control Register	P42PFS	8	8	2 or 3 PCLKB	section 18.
0008 C163h	MPC	P43 Pin Function Control Register	P43PFS	8	8	2 or 3 PCLKB	section 18.
0008 C164h	MPC	P44 Pin Function Control Register	P44PFS	8	8	2 or 3 PCLKB	section 18.
0008 C165h	MPC	P45 Pin Function Control Register	P45PFS	8	8	2 or 3 PCLKB	section 18.
0008 C166h	MPC	P46 Pin Function Control Register	P46PFS	8	8	2 or 3 PCLKB	section 18.
0008 C167h	MPC	P47 Pin Function Control Register	P47PFS	8	8	2 or 3 PCLKB	section 18.
0008 C178h	MPC	P70 Pin Function Control Register	P70PFS	8	8	2 or 3 PCLKB	section 18.
0008 C179h	MPC	P71 Pin Function Control Register	P71PFS	8	8	2 or 3 PCLKB	section 18.
0008 C17Ah	MPC	P72 Pin Function Control Register	P72PFS	8	8	2 or 3 PCLKB	section 18.
0008 C17Bh	MPC	P73 Pin Function Control Register	P73PFS	8	8	2 or 3 PCLKB	section 18.
0008 C17Ch	MPC	P74 Pin Function Control Register	P74PFS	8	8	2 or 3 PCLKB	section 18.
0008 C17Dh	MPC	P75 Pin Function Control Register	P75PFS	8	8	2 or 3 PCLKB	section 18.
0008 C17Eh	MPC	P76 Pin Function Control Register	P76PFS	8	8	2 or 3 PCLKB	section 18.
0008 C18Bh	MPC	P93 Pin Function Control Register	P93PFS	8	8	2 or 3 PCLKB	section 18.
0008 C18Ch	MPC	P94 Pin Function Control Register	P94PFS	8	8	2 or 3 PCLKB	section 18.
0008 C192h	MPC	PA2 Pin Function Control Register	PA2PFS	8	8	2 or 3 PCLKB	section 18.
0008 C193h	MPC	PA3 Pin Function Control Register	PA3PFS	8	8	2 or 3 PCLKB	section 18.
0008 C198h	MPC	PB0 Pin Function Control Register	PB0PFS	8	8	2 or 3 PCLKB	section 18.
0008 C199h	MPC	PB1 Pin Function Control Register	PB1PFS	8	8	2 or 3 PCLKB	section 18.
0008 C19Ah	MPC	PB2 Pin Function Control Register	PB2PFS	8	8	2 or 3 PCLKB	section 18.
0008 C19Bh	MPC	PB3 Pin Function Control Register	PB3PFS	8	8	2 or 3 PCLKB	section 18.
0008 C19Ch	MPC	PB4 Pin Function Control Register	PB4PFS	8	8	2 or 3 PCLKB	section 18.
0008 C19Dh	MPC	PB5 Pin Function Control Register	PB5PFS	8	8	2 or 3 PCLKB	section 18.
0008 C19Eh	MPC	PB6 Pin Function Control Register	PB6PFS	8	8	2 or 3 PCLKB	section 18.
0008 C19Fh	MPC	PB7 Pin Function Control Register	PB7PFS	8	8	2 or 3 PCLKB	section 18.
0008 C1ABh	MPC	PD3 Pin Function Control Register	PD3PFS	8	8	2 or 3 PCLKB	section 18.
0008 C1ACh	MPC	PD4 Pin Function Control Register	PD4PFS	8	8	2 or 3 PCLKB	section 18.
0008 C1ADh	MPC	PD5 Pin Function Control Register	PD5PFS	8	8	2 or 3 PCLKB	section 18.
0008 C1AEh	MPC	PD6 Pin Function Control Register	PD6PFS	8	8	2 or 3 PCLKB	section 18.
0008 C1B2h	MPC	PE2 Pin Function Control Register	PE2PFS	8	8	2 or 3 PCLKB	section 18.

Table 5.1 List of I/O Registers (Address Order) (8 / 11)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles	Reference Section
0008 C290h	SYSTEM	Reset Status Register 0	RSTSR0	8	8	4 or 5 PCLKB	section 6.
0008 C291h	SYSTEM	Reset Status Register 1	RSTSR1	8	8	4 or 5 PCLKB	section 6.
0008 C293h	SYSTEM	Main Clock Oscillator Forced Oscillation Control Register	MOFCR	8	8	4 or 5 PCLKB	section 9.
0008 C297h	SYSTEM	Voltage Monitoring Circuit Control Register	LVCMPCR	8	8	4 or 5 PCLKB	section 8.
0008 C298h	SYSTEM	Voltage Detection Level Select Register	LVDLVLR	8	8	4 or 5 PCLKB	section 8.
0008 C29Ah	SYSTEM	Voltage Monitoring 1 Circuit Control Register 0	LVD1CR0	8	8	4 or 5 PCLKB	section 8.
0008 C29Bh	SYSTEM	Voltage Monitoring 2 Circuit Control Register 0	LVD2CR0	8	8	4 or 5 PCLKB	section 8.
0008 C4C0h	POE	Input Level Control/Status Register 1	ICSR1	16	8, 16	2 or 3PCLKB	section 20.
0008 C4C2h	POE	Output Level Control/Status Register 1	OCSR1	16	8, 16	2 or 3PCLKB	section 20.
0008 C4C8h	POE	Input Level Control/Status Register 3	ICSR3	16	8, 16	2 or 3PCLKB	section 20.
0008 C4CAh	POE	Software Port Output Enable Register	SPOER	8	8	2 or 3PCLKB	section 20.
0008 C4CBh	POE	Port Output Enable Control Register 1	POECR1	8	8	2 or 3PCLKB	section 20.
0008 C4CCh	POE	Port Output Enable Control Register 2	POECR2	16	16	2 or 3PCLKB	section 20.
0008 C4D0h	POE	Port Output Enable Control Register 4	POECR4	16	16	2 or 3PCLKB	section 20.
0008 C4D2h	POE	Port Output Enable Control Register 5	POECR5	16	16	2 or 3PCLKB	section 20.
0008 C4D6h	POE	Input Level Control/Status Register 4	ICSR4	16	8, 16	2 or 3PCLKB	section 20.
0008 C4DAh	POE	Active Level Setting Register 1	ALR1	16	8, 16	2 or 3PCLKB	section 20.
0008 C4DCh	POE	Input Level Control/Status Register 6	ICSR6	16	16	2 or 3PCLKB	section 20.
0008 C4E6h	POE	Port Output Enable Comparator Output Detection Flag Register	POECMPFR	16	16	2 or 3PCLKB	section 20.
0008 C4E8h	POE	Port Output Enable Comparator Request Select Register	POECMPSEL	16	16	2 or 3PCLKB	section 20.
0009 5200h	MTU3	Timer Control Register	TCR	8	8, 16, 32	4 or 5PCLKB	section 19.
0009 5201h	MTU4	Timer Control Register	TCR	8	8	4 or 5PCLKB	section 19.
0009 5202h	MTU3	Timer Mode Register 1	TMDR1	8	8, 16	4 or 5PCLKB	section 19.
0009 5203h	MTU4	Timer Mode Register 1	TMDR1	8	8	4 or 5PCLKB	section 19.
0009 5204h	MTU3	Timer I/O Control Register H	TIORH	8	8, 16, 32	4 or 5PCLKB	section 19.
0009 5205h	MTU3	Timer I/O Control Register L	TIORL	8	8	4 or 5PCLKB	section 19.
0009 5206h	MTU4	Timer I/O Control Register H	TIORH	8	8, 16	4 or 5PCLKB	section 19.
0009 5207h	MTU4	Timer I/O Control Register L	TIORL	8	8	4 or 5PCLKB	section 19.
0009 5208h	MTU3	Timer Interrupt Enable Register	TIER	8	8, 16	4 or 5PCLKB	section 19.
0009 5209h	MTU4	Timer Interrupt Enable Register	TIER	8	8	4 or 5PCLKB	section 19.
0009 520Ah	MTU	Timer Output Master Enable Register A	TOERA	8	8	4 or 5PCLKB	section 19.
0009 520Dh	MTU	Timer Gate Control Register A	TGCRA	8	8	4 or 5PCLKB	section 19.
0009 520Eh	MTU	Timer Output Control Register 1A	TOCR1A	8	8, 16	4 or 5PCLKB	section 19.
0009 520Fh	MTU	Timer Output Control Register 2A	TOCR2A	8	8	4 or 5PCLKB	section 19.
0009 5210h	MTU3	Timer Counter	TCNT	16	16, 32	4 or 5PCLKB	section 19.
0009 5212h	MTU4	Timer Counter	TCNT	16	16	4 or 5PCLKB	section 19.
0009 5214h	MTU	Timer Period Data Register A	TCDRA	16	16, 32	4 or 5PCLKB	section 19.
0009 5216h	MTU	Timer Dead Time Data Register A	TDDRA	16	16	4 or 5PCLKB	section 19.
0009 5218h	MTU3	Timer General Register A	TGRA	16	16, 32	4 or 5PCLKB	section 19.
0009 521Ah	MTU3	Timer General Register B	TGRB	16	16	4 or 5PCLKB	section 19.
0009 521Ch	MTU4	Timer General Register A	TGRA	16	16, 32	4 or 5PCLKB	section 19.
0009 521Eh	MTU4	Timer General Register B	TGRB	16	16	4 or 5PCLKB	section 19.
0009 5220h	MTU	Timer Subcounter A	TCNTSA	16	16, 32	4 or 5PCLKB	section 19.
0009 5222h	MTU	Timer Period Buffer Registera A	TCBRA	16	16	4 or 5PCLKB	section 19.
0009 5224h	MTU3	Timer General Register C	TGRC	16	16, 32	4 or 5PCLKB	section 19.
0009 5226h	MTU3	Timer General Register D	TGRD	16	16	4 or 5PCLKB	section 19.
0009 5228h	MTU4	Timer General Register C	TGRC	16	16, 32	4 or 5PCLKB	section 19.
0009 522Ah	MTU4	Timer General Register D	TGRD	16	16	4 or 5PCLKB	section 19.
0009 522Ch	MTU3	Timer Status Register	TSR	8	8, 16	4 or 5PCLKB	section 19.
0009 522Dh	MTU4	Timer Status Register	TSR	8	8	4 or 5PCLKB	section 19.
0009 5230h	MTU	Timer Interrupt Skipping Set Register 1A	TITCR1A	8	8, 16	4 or 5PCLKB	section 19.
0009 5231h	MTU	Timer Interrupt Skipping Counter 1A	TITCNT1A	8	8	4 or 5PCLKB	section 19.

Table 5.1 List of I/O Registers (Address Order) (9 / 11)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles	Reference Section
0009 5232h	MTU	Timer Buffer Transfer Set Register A	TBTERA	8	8	4 or 5PCLKB	section 19.
0009 5234h	MTU	Timer Dead Time Enable Register A	TDERA	8	8	4 or 5PCLKB	section 19.
0009 5236h	MTU	Timer Output Level Buffer Register A	TOLBRA	8	8	4 or 5PCLKB	section 19.
0009 5238h	MTU3	Timer Buffer Operation Transfer Mode Register	TBTM	8	8, 16	4 or 5PCLKB	section 19.
0009 5239h	MTU4	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	4 or 5PCLKB	section 19.
0009 523Ah	MTU	Timer Interrupt Skipping Mode Register A	TITMRA	8	8	4 or 5PCLKB	section 19.
0009 523Bh	MTU	Timer Interrupt Skipping Set Register 2A	TITCR2A	8	8	4 or 5PCLKB	section 19.
0009 523Ch	MTU	Timer Interrupt Skipping Counter 2A	TITCNT2A	8	8	4 or 5PCLKB	section 19.
0009 5240h	MTU4	Timer A/D Conversion Start Request Control Register	TADCR	16	16	4 or 5PCLKB	section 19.
0009 5244h	MTU4	Timer A/D Conversion Start Request Cycle Set Register A	TADCORA	16	16, 32	4 or 5PCLKB	section 19.
0009 5246h	MTU4	Timer A/D Conversion Start Request Cycle Set Register B	TADCORB	16	16	4 or 5PCLKB	section 19.
0009 5248h	MTU4	Timer A/D Conversion Start Request Cycle Set Buffer Register A	TADCOBRA	16	16, 32	4 or 5PCLKB	section 19.
0009 524Ah	MTU4	Timer A/D Conversion Start Request Cycle Set Buffer Register B	TADCOBRB	16	16	4 or 5PCLKB	section 19.
0009 524Ch	MTU3	Timer Control Register 2	TCR2	8	8	4 or 5PCLKB	section 19.
0009 524Dh	MTU4	Timer Control Register 2	TCR2	8	8	4 or 5PCLKB	section 19.
0009 5260h	MTU	Timer Waveform Control Register A	TWCRA	8	8	4 or 5PCLKB	section 19.
0009 5270h	MTU	Timer Mode Register 2A	TMDR2A	8	8	4 or 5PCLKB	section 19.
0009 5272h	MTU3	Timer General Register E	TGRE	16	16	4 or 5PCLKB	section 19.
0009 5274h	MTU4	Timer General Register E	TGRE	16	16	4 or 5PCLKB	section 19.
0009 5276h	MTU4	Timer General Register F	TGRF	16	16	4 or 5PCLKB	section 19.
0009 5280h	MTU	Timer Start Register A	TSTRA	8	8, 16	4 or 5PCLKB	section 19.
0009 5281h	MTU	Timer Synchronous Register A	TSYRA	8	8	4 or 5PCLKB	section 19.
0009 5282h	MTU	Timer Counter Synchronous Start Register	TCSYSTR	8	8	4 or 5PCLKB	section 19.
0009 5284h	MTU	Timer Read/Write Enable Register A	TRWERA	8	8	4 or 5PCLKB	section 19.
0009 5290h	MTU0	Noise Filter Control Register 0	NFCR0	8	8	4 or 5PCLKB	section 19.
0009 5291h	MTU1	Noise Filter Control Register 1	NFCR1	8	8	4 or 5PCLKB	section 19.
0009 5292h	MTU2	Noise Filter Control Register 2	NFCR2	8	8	4 or 5PCLKB	section 19.
0009 5293h	MTU3	Noise Filter Control Register 3	NFCR3	8	8	4 or 5PCLKB	section 19.
0009 5294h	MTU4	Noise Filter Control Register 4	NFCR4	8	8	4 or 5PCLKB	section 19.
0009 5295h	MTU5	Noise Filter Control Register 5	NFCR5	8	8	4 or 5PCLKB	section 19.
0009 5299h	MTU0	Noise Filter Control Register C	NFCRC	8	8	4 or 5PCLKB	section 19.
0009 5300h	MTU0	Timer Control Register	TCR	8	8, 16, 32	4 or 5PCLKB	section 19.
0009 5301h	MTU0	Timer Mode Register 1	TMDR1	8	8	4 or 5PCLKB	section 19.
0009 5302h	MTU0	Timer I/O Control Register H	TIORH	8	8, 16	4 or 5PCLKB	section 19.
0009 5303h	MTU0	Timer I/O Control Register L	TIORL	8	8	4 or 5PCLKB	section 19.
0009 5304h	MTU0	Timer Interrupt Enable Register	TIER	8	8, 16, 32	4 or 5PCLKB	section 19.
0009 5306h	MTU0	Timer Counter	TCNT	16	16	4 or 5PCLKB	section 19.
0009 5308h	MTU0	Timer General Register A	TGRA	16	16, 32	4 or 5PCLKB	section 19.
0009 530Ah	MTU0	Timer General Register B	TGRB	16	16	4 or 5PCLKB	section 19.
0009 530Ch	MTU0	Timer General Register C	TGRC	16	16, 32	4 or 5PCLKB	section 19.
0009 530Eh	MTU0	Timer General Register D	TGRD	16	16	4 or 5PCLKB	section 19.
0009 5320h	MTU0	Timer General Register E	TGRE	16	16, 32	4 or 5PCLKB	section 19.
0009 5322h	MTU0	Timer General Register F	TGRF	16	16	4 or 5PCLKB	section 19.
0009 5324h	MTU0	Timer Interrupt Enable Register 2	TIER2	8	8, 16	4 or 5PCLKB	section 19.
0009 5326h	MTU0	Timer Buffer Operation Transfer Mode Register	TBTM	8	8	4 or 5PCLKB	section 19.
0009 5328h	MTU0	Timer Control Register 2	TCR2	8	8	4 or 5PCLKB	section 19.
0009 5380h	MTU1	Timer Control Register	TCR	8	8, 16	4 or 5PCLKB	section 19.
0009 5381h	MTU1	Timer Mode Register 1	TMDR1	8	8	4 or 5PCLKB	section 19.
0009 5382h	MTU1	Timer I/O Control Register	TIOR	8	8	4 or 5PCLKB	section 19.
0009 5384h	MTU1	Timer Interrupt Enable Register	TIER	8	8, 16, 32	4 or 5PCLKB	section 19.
0009 5385h	MTU1	Timer Status Register	TSR	8	8	4 or 5PCLKB	section 19.

Table 5.1 List of I/O Registers (Address Order) (10 / 11)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles	Reference Section
0009 5386h	MTU1	Timer Counter	TCNT	16	16	4 or 5PCLKB	section 19.
0009 5388h	MTU1	Timer General Register A	TGRA	16	16, 32	4 or 5PCLKB	section 19.
0009 538Ah	MTU1	Timer General Register B	TGRB	16	16	4 or 5PCLKB	section 19.
0009 5390h	MTU1	Timer Input Capture Control Register	TICCR	8	8	4 or 5PCLKB	section 19.
0009 5391h	MTU1	Timer Mode Register 3	TMDR3	8	8	4 or 5PCLKB	section 19.
0009 5394h	MTU1	Timer Control Register 2	TCR2	8	8	4 or 5PCLKB	section 19.
0009 53A0h	MTU1	Timer Longword Counter	TCNTLW	32	32	4 or 5PCLKB	section 19.
0009 53A4h	MTU1	Timer Longword General Register	TGRALW	32	32	4 or 5PCLKB	section 19.
0009 53A8h	MTU1	Timer Longword General Register	TGRBLW	32	32	4 or 5PCLKB	section 19.
0009 5400h	MTU2	Timer Control Register	TCR	8	8, 16	4 or 5PCLKB	section 19.
0009 5401h	MTU2	Timer Mode Register 1	TMDR1	8	8	4 or 5PCLKB	section 19.
0009 5402h	MTU2	Timer I/O Control Register	TIOR	8	8	4 or 5PCLKB	section 19.
0009 5404h	MTU2	Timer Interrupt Enable Register	TIER	8	8, 16, 32	4 or 5PCLKB	section 19.
0009 5405h	MTU2	Timer Status Register	TSR	8	8	4 or 5PCLKB	section 19.
0009 5406h	MTU2	Timer Counter	TCNT	16	16	4 or 5PCLKB	section 19.
0009 5408h	MTU2	Timer General Register A	TGRA	16	16, 32	4 or 5PCLKB	section 19.
0009 540Ah	MTU2	Timer General Register B	TGRB	16	16	4 or 5PCLKB	section 19.
0009 540Ch	MTU2	Timer Control Register 2	TCR2	8	8	4 or 5PCLKB	section 19.
0009 5480h	MTU5	Timer Counter U	TCNTU	16	16, 32	4 or 5PCLKB	section 19.
0009 5482h	MTU5	Timer General Register U	TGRU	16	16	4 or 5PCLKB	section 19.
0009 5484h	MTU5	Timer Control Register U	TCRU	8	8	4 or 5PCLKB	section 19.
0009 5485h	MTU5	Timer Control Register 2U	TCR2U	8	8	4 or 5PCLKB	section 19.
0009 5486h	MTU5	Timer I/O Control Register U	TIORU	8	8	4 or 5PCLKB	section 19.
0009 5490h	MTU5	Timer Counter V	TCNTV	16	16, 32	4 or 5PCLKB	section 19.
0009 5492h	MTU5	Timer General Register V	TGRV	16	16	4 or 5PCLKB	section 19.
0009 5494h	MTU5	Timer Control Register V	TCRV	8	8	4 or 5PCLKB	section 19.
0009 5495h	MTU5	Timer Control Register 2V	TCR2V	8	8	4 or 5PCLKB	section 19.
0009 5496h	MTU5	Timer I/O Control Register V	TIORV	8	8	4 or 5PCLKB	section 19.
0009 54A0h	MTU5	Timer Counter W	TCNTW	16	16, 32	4 or 5PCLKB	section 19.
0009 54A2h	MTU5	Timer General Register W	TGRW	16	16	4 or 5PCLKB	section 19.
0009 54A4h	MTU5	Timer Control Register W	TCRW	8	8	4 or 5PCLKB	section 19.
0009 54A5h	MTU5	Timer Control Register 2W	TCR2W	8	8	4 or 5PCLKB	section 19.
0009 54A6h	MTU5	Timer I/O Control Register W	TIORW	8	8	4 or 5PCLKB	section 19.
0009 54B2h	MTU5	Timer Interrupt Enable Register	TIER	8	8	4 or 5PCLKB	section 19.
0009 54B4h	MTU5	Timer Start Register	TSTR	8	8	4 or 5PCLKB	section 19.
0009 54B6h	MTU5	Timer Compare Match Clear Register	TCNTCMPCLR	8	8	4 or 5PCLKB	section 19.
0009 5D30h	MTU	A/D Conversion Start Request Select Register 0	TADSTRGR0	8	8	4 or 5PCLKB	section 19.
000A 0C80h	CMPC0	Comparator Control Register	CMPCTL	8	8	1 or 2PCLKB	section 28.
000A 0C84h	CMPC0	Comparator Input Select Register	CMPSEL0	8	8	1 or 2PCLKB	section 28.
000A 0C88h	CMPC0	Comparator Reference Voltage Select Register	CMPSEL1	8	8	1 or 2PCLKB	section 28.
000A 0C8Ch	CMPC0	Comparator Output Monitor Register	CMPMON	8	8	1 or 2PCLKB	section 28.
000A 0C90h	CMPC0	Comparator External Output Enable Register	CMPIOC	8	8	1 or 2PCLKB	section 28.
000A 0CA0h	CMPC1	Comparator Control Register	CMPCTL	8	8	1 or 2PCLKB	section 28.
000A 0CA4h	CMPC1	Comparator Input Select Register	CMPSEL0	8	8	1 or 2PCLKB	section 28.
000A 0CA8h	CMPC1	Comparator Reference Voltage Select Register	CMPSEL1	8	8	1 or 2PCLKB	section 28.
000A 0CACh	CMPC1	Comparator Output Monitor Register	CMPMON	8	8	1 or 2PCLKB	section 28.
000A 0CB0h	CMPC1	Comparator External Output Enable Register	CMPIOC	8	8	1 or 2PCLKB	section 28.
000A 0CC0h	CMPC2	Comparator Control Register	CMPCTL	8	8	1 or 2PCLKB	section 28.
000A 0CC4h	CMPC2	Comparator Input Select Register	CMPSEL0	8	8	1 or 2PCLKB	section 28.
000A 0CC8h	CMPC2	Comparator Reference Voltage Select Register	CMPSEL1	8	8	1 or 2PCLKB	section 28.
000A 0CCCh	CMPC2	Comparator Output Monitor Register	CMPMON	8	8	1 or 2PCLKB	section 28.
000A 0CD0h	CMPC2	Comparator External Output Enable Register	CMPIOC	8	8	1 or 2PCLKB	section 28.

Table 5.1 List of I/O Registers (Address Order) (11 / 11)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access Cycles	Reference Section
007F C090h	FLASH	E2 DataFlash Control Register	DFLCTL	8	8	2 or 3 FCLK	section 31.
007F C0B0h	FLASH	Flash Start-Up Setting Monitor Register	FSCMR	16	16	2 or 3 FCLK	section 31.
007F C0B2h	FLASH	Flash Access Window Start Address Monitor Register	FAWSMR	16	16	2 or 3 FCLK	section 31.
007F C0B4h	FLASH	Flash Access Window End Address Monitor Register	FAWEMR	16	16	2 or 3 FCLK	section 31.
007F C0B6h	FLASH	Flash Initial Setting Register	FISR	8	8	2 or 3 FCLK	section 31.
007F C0B7h	FLASH	Flash Extra Area Control Register	FEXCR	8	8	2 or 3 FCLK	section 31.
007F C0B8h	FLASH	Flash Error Address Monitor Register L	FEAML	16	16	2 or 3 FCLK	section 31.
007F C0BAh	FLASH	Flash Error Address Monitor Register H	FEAMH	8	8	2 or 3 FCLK	section 31.
007F C0C0h	FLASH	Protection Unlock Register	FPR	8	8	2 or 3 FCLK	section 31.
007F C0C1h	FLASH	Protection Unlock Status Register	FPSR	8	8	2 or 3 FCLK	section 31.
007F C0C2h	FLASH	Flash Read Buffer Register L	FRBL	16	16	2 or 3 FCLK	section 31.
007F C0C4h	FLASH	Flash Read Buffer Register H	FRBH	16	16	2 or 3 FCLK	section 31.
007F FF80h	FLASH	Flash P/E Mode Control Register	FPMCR	8	8	2 or 3 FCLK	section 31.
007F FF81h	FLASH	Flash Area Select Register	FASR	8	8	2 or 3 FCLK	section 31.
007F FF82h	FLASH	Flash Processing Start Address Register L	FSARL	16	16	2 or 3 FCLK	section 31.
007F FF84h	FLASH	Flash Processing Start Address Register H	FSARH	8	8	2 or 3 FCLK	section 31.
007F FF85h	FLASH	Flash Control Register	FCR	8	8	2 or 3 FCLK	section 31.
007F FF86h	FLASH	Flash Processing End Address Register L	FEARL	16	16	2 or 3 FCLK	section 31.
007F FF88h	FLASH	Flash Processing End Address Register H	FEARH	8	8	2 or 3 FCLK	section 31.
007F FF89h	FLASH	Flash Reset Register	FRESETR	8	8	2 or 3 FCLK	section 31.
007F FF8Ah	FLASH	Flash Status Register 0	FSTATR0	8	8	2 or 3 FCLK	section 31.
007F FF8Bh	FLASH	Flash Status Register 1	FSTATR1	8	8	2 or 3 FCLK	section 31.
007F FF8Ch	FLASH	Flash Write Buffer Register L	FWBL	16	16	2 or 3 FCLK	section 31.
007F FF8Eh	FLASH	Flash Write Buffer Register H	FWBH	16	16	2 or 3 FCLK	section 31.
007F FFB2h	FLASH	Flash P/E Mode Entry Register	FENTRYR	16	16	2 or 3 FCLK	section 31.

6. Resets

6.1 Overview

The following resets are implemented: RES# pin reset, power-on reset, voltage monitoring 0 reset, voltage monitoring 1 reset, voltage monitoring 2 reset, independent watchdog timer reset, and software reset.
Table 6.1 lists the reset names and sources.

Table 6.1 Reset Names and Sources

Reset Name	Source
RES# pin reset	Voltage input to the RES# pin is driven low.
Power-on reset	VCC rises (voltage monitored: VPOR)*1
Voltage monitoring 0 reset	VCC falls (voltage monitored: Vdet0)*1
Voltage monitoring 1 reset	VCC falls (voltage monitored: Vdet1)*1
Voltage monitoring 2 reset	VCC falls (voltage monitored: Vdet2)*1
Independent watchdog timer reset	The independent watchdog timer underflows, or a refresh error occurs.
Software reset	Register setting

Note 1. For the voltages to be monitored (VPOR, Vdet0, Vdet1, and Vdet2), see section 8, Voltage Detection Circuit (LVDAb) and section 32, Electrical Characteristics.

The internal state and pins are initialized by a reset.
Table 6.2 lists the reset targets to be initialized.

Table 6.2 Targets Initialized by Each Reset Source

Target to be Initialized	Reset Source						
	RES# Pin Reset	Power-On Reset	Voltage Monitoring 0 Reset	Independent Watchdog Timer Reset	Voltage Monitoring 1 Reset	Voltage Monitoring 2 Reset	Software Reset
The power-on reset detect flag (RSTSR0.PORF)	○	—	—	—	—	—	—
Register related to the cold start/warm start determination flag (RSTSR1.CWSF)	—*1	○	—	—	—	—	—
Voltage monitoring 0 reset detect flag (RSTSR0.LVD0RF)	○	○	—	—	—	—	—
The independent watchdog timer reset detect flag (RSTSR2.IWDTRF)	○	○	○	—	—	—	—
Registers related to the independent watchdog timer (IWDTRR, IWDTCR, IWDTSR, IWDTRCR, IWDTCSTPR, ILOCOCR)	○	○	○	—	—	—	—
The voltage monitoring 1 reset detect flag (RSTSR0.LVD1RF)	○	○	○	○	—	—	—
Registers related to voltage monitor function 1 (LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL[3:0])	○	○	○	○	—	—	—
(LVD1CR1, LVD1SR)	○	○	○	○	—	—	—
The voltage monitoring 2 reset detect flag (RSTSR0.LVD2RF)	○	○	○	○	○	—	—
Registers related to voltage monitor function 2 (LVD2CR0, LVD2E, LVDLVLR.LVD2LVL[1:0])	○	○	○	○	○	—	—
(LVD2CR1, LVD2SR)	○	○	○	○	○	—	—
The software reset detect flag (RSTSR2.SWRF)	○	○	○	○	○	○	—
Registers other than the above, CPU, and internal state	○	○	○	○	○	○	○

○: Targets to be initialized, —: No change occurs.
Note 1. Initialized at a power-on.

When a reset is canceled, the reset exception handling starts. For the reset exception handling, see section 13, Exception Handling.
Table 6.3 lists the pin related to the reset.

Table 6.3 Pin Related to Reset

Pin Name	I/O	Function
RES#	Input	Reset pin

6.2 Register Descriptions

6.2.1 Reset Status Register 0 (RSTSR0)

Address(es): 0008 C290h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	LVD2RF	LVD1RF	LVD0RF	PORF
Value after reset:	0	0	0	0	0*1	0*1	0*1	0*1

Bit	Symbol	Bit Name	Description	R/W
b0	PORF	Power-On Reset Detect Flag	0: Power-on reset not detected. 1: Power-on reset detected.	R(/W) *2
b1	LVD0RF	Voltage Monitoring 0 Reset Detect Flag	0: Voltage monitoring 0 reset not detected. 1: Voltage monitoring 0 reset detected.	R(/W) *2
b2	LVD1RF	Voltage Monitoring 1 Reset Detect Flag	0: Voltage monitoring 1 reset not detected. 1: Voltage monitoring 1 reset detected.	R(/W) *2
b3	LVD2RF	Voltage Monitoring 2 Reset Detect Flag	0: Voltage monitoring 2 reset not detected. 1: Voltage monitoring 2 reset detected.	R(/W) *2
b7 to b4	—	Reserved	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.
Note 2. Only 0 can be written to clear the flag.

PORF Flag (Power-On Reset Detect Flag)

The PORF flag indicates that a power-on reset has occurred.
[Setting condition]

- When a power-on reset occurs.

[Clearing conditions]

- When resets shown in Table 6.2 occur.
- When PORF is read as 1 and then 0 is written to PORF.

LVD0RF Flag (Voltage Monitoring 0 Reset Detect Flag)

The LVD0RF flag indicates that VCC voltage has fallen below Vdet0.
[Setting condition]

- When Vdet0-level VCC voltage is detected.

[Clearing conditions]

- When resets listed in Table 6.2 occur.
- When LVD0RF is read as 1 and then 0 is written to LVD0RF.

LVD1RF Flag (Voltage Monitoring 1 Reset Detect Flag)

The LVD1RF flag indicates that VCC voltage has fallen below Vdet1.
[Setting condition]

- When Vdet1-level VCC voltage is detected.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When LVD1RF is read as 1 and then 0 is written to LVD1RF.

LVD2RF Flag (Voltage Monitoring 2 Reset Detect Flag)

The LVD2RF flag indicates that VCC voltage has fallen below Vdet2.
[Setting condition]

- When Vdet2-level VCC voltage is detected.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When LVD2RF is read as 1 and then 0 is written to LVD2RF.

6.2.2 Reset Status Register 1 (RSTSR1)

Address(es): 0008 C291h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CWSF
Value after reset:	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Bit Name	Description	R/W
b0	CWSF	Cold/Warm Start Determination Flag	0: Cold start 1: Warm start	R(/W) *2
b7 to b1	—	Reserved	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.
Note 2. Only 1 can be written to set the flag.

RSTSR1 determines whether a power-on reset has caused the reset processing (cold start) or a reset signal input during operation has caused the reset processing (warm start).

CWSF Flag (Cold/Warm Start Determination Flag)

The CWSF flag indicates the type of reset processing: cold start or warm start.
The CWSF flag is initialized at a power-on.
[Setting condition]

- When 1 is written through programming; it is not set to 0 even when 0 is written.

[Clearing condition]

- When a reset listed in Table 6.2 occurs.

6.2.3 Reset Status Register 2 (RSTSR2)

Address(es): 0008 00C0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SWRF	—	IWDTRF
Value after reset:	0	0	0	0	0	0*1	0	0*1

Bit	Symbol	Bit Name	Description	R/W
b0	IWDTRF	Independent Watchdog Timer Reset Detect Flag	0: Independent watchdog timer reset not detected. 1: Independent watchdog timer reset detected.	R(/W) *2
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	SWRF	Software Reset Detect Flag	0: Software reset not detected. 1: Software reset detected.	R(/W) *2
b7 to b3	—	Reserved	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.
Note 2. Only 0 can be written to clear the flag.

IWDTRF Flag (Independent Watchdog Timer Reset Detect Flag)

The IWDTRF flag indicates that an independent watchdog timer reset has occurred.

[Setting condition]

- When an independent watchdog timer reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When IWDTRF is read as 1 and then 0 is written to IWDTRF.

SWRF Flag (Software Reset Detect Flag)

The SWRF flag indicates that a software reset has occurred.

[Setting condition]

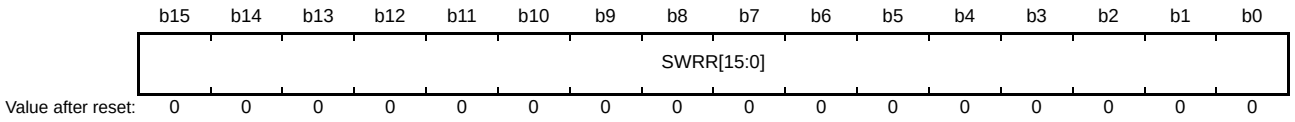
- When a software reset occurs.

[Clearing conditions]

- When a reset listed in Table 6.2 occurs.
- When SWRF is read as 1 and then 0 is written to SWRF.

6.2.4 Software Reset Register (SWRR)

Address(es): 0008 00C2h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	SWRR[15:0]	Software Reset	Writing A501h resets the LSI. These bits are read as 0000h.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

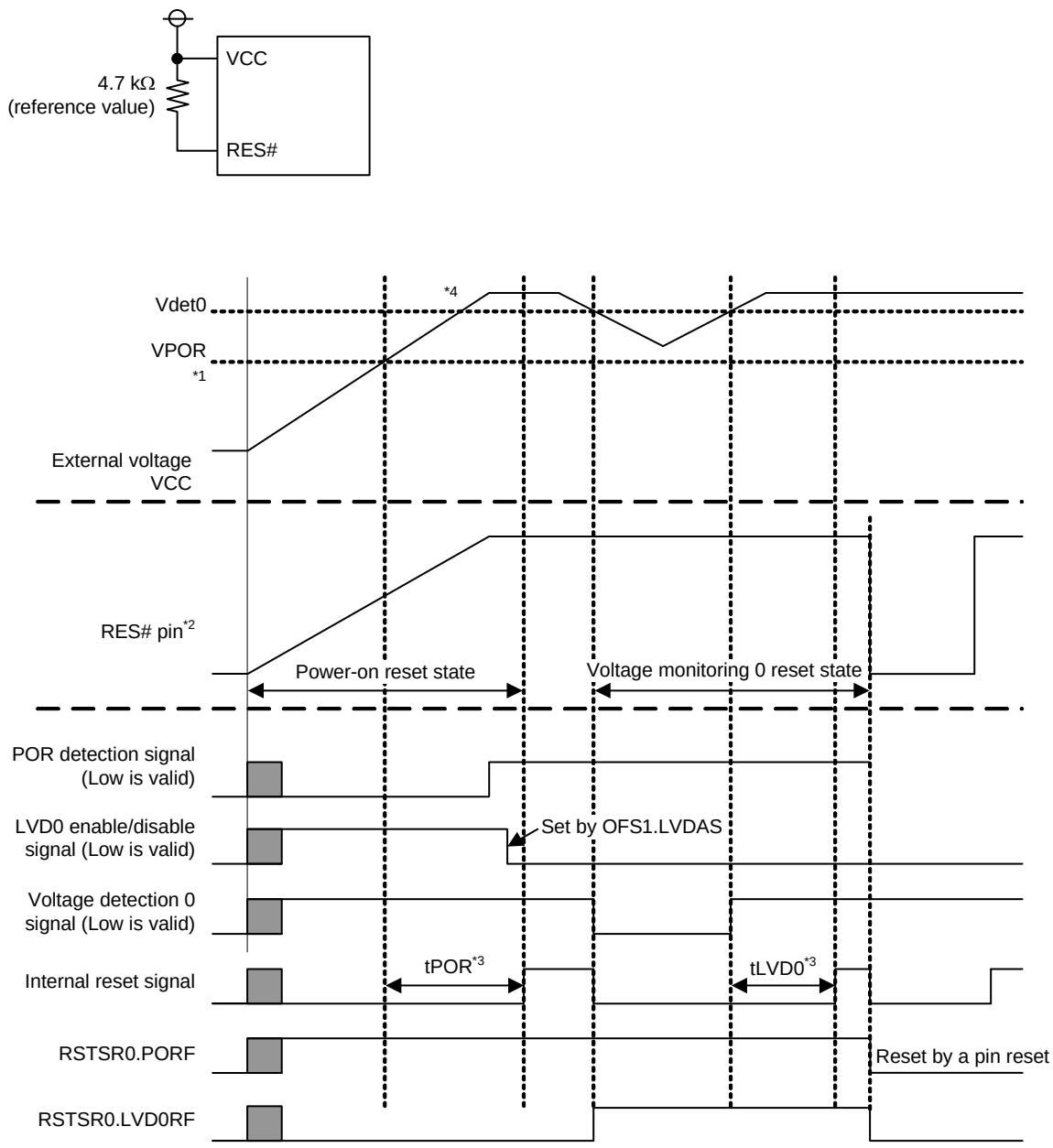
6.3 Operation

6.3.1 RES# Pin Reset

This is a reset generated by the RES# pin.
When the RES# pin is driven low, all the processing in progress is aborted and the LSI enters a reset state.
In order to unfailingly reset the LSI, the RES# pin should be held low for the specified power supply stabilization time at a power-on.
When the RES# pin is driven high from low, the internal reset is canceled after the post-RES# cancellation wait time (tRESWT) has elapsed, and then the CPU starts the reset exception handling.
For details, see section 32, Electrical Characteristics.

6.3.2 Power-On Reset and Voltage Monitoring 0 Reset

The power-on reset is an internal reset generated by the power-on reset circuit. A power-on reset is generated when power is supplied to the RES# pin while it is connected to VCC via a resistor. When connecting a capacitor to the RES# pin, also ensure that the voltage on the RES# pin is always at least VIH. For details on VIH, refer to section 32, Electrical Characteristics. After VCC has exceeded VPOR and the specified period (power-on reset time) has elapsed, the internal reset is canceled and the CPU starts the reset exception handling. The power-on reset time is a stabilization period for the external power supply and the MCU circuit. After a power-on reset has been generated, the PORF flag in RSTSR0 is set to 1. The PORF flag is initialized by RES# pin reset.
The voltage monitoring 0 reset is an internal reset generated by the voltage monitoring circuit. If the voltage detection circuit 0 start bit (LVDAS) in option function select register 1 (OFS1) is 0 (voltage monitoring 0 reset is enabled after a reset) and VCC falls below Vdet0, the RSTSR0.LVD0RF flag becomes 1 and the voltage detection circuit generates voltage monitoring 0 reset. Clear the OFS1.LVDAS bit to 0 if the voltage monitoring 0 reset is to be used.
Release from the voltage monitoring 0 reset state occurs when VCC rises above Vdet0 and the LVD0 reset time (tLVD0) elapses, and then the CPU starts the reset exception handling.
Figure 6.1 shows operations during a power-on reset and voltage monitoring 0 reset.
For details on voltage monitoring 0 reset, refer to section 8, Voltage Detection Circuit (LVDAb).



Note: For details on the electrical characteristics, see the Electrical Characteristics section.
Note 1. Vdet0 indicates the detection level for a voltage monitoring 0 reset and VPOR indicates the detection level for a power-on reset.
Note 2. Ensure that the voltage on the RES# pin is always at least VIH.
Note 3. tPOR indicates the period of a power-on reset and tLVD0 indicates the period of a voltage monitoring 0 reset.
Note 4. At the time the power-supply voltage rises, VCC must rise to at least the minimum guaranteed voltage before release from the POR reset state.

Figure 6.1 Operation Examples During a Power-On Reset and Voltage Monitoring 0 Reset

6.3.3 Voltage Monitoring 1 Reset and Voltage Monitoring 2 Reset

The voltage monitoring 1 reset and voltage monitoring 2 reset are internal resets generated by the voltage monitoring circuit.

When the voltage monitoring 1 interrupt/reset enable bit (LVD1RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the voltage monitoring 1 circuit mode select bit (LVD1RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in the voltage monitoring 1 circuit control register 0 (LVD1CR0), the RSTSR0.LVD1RF flag is set to 1 and the voltage-detection circuit generates a voltage monitoring 1 reset if VCC falls to or below Vdet1.

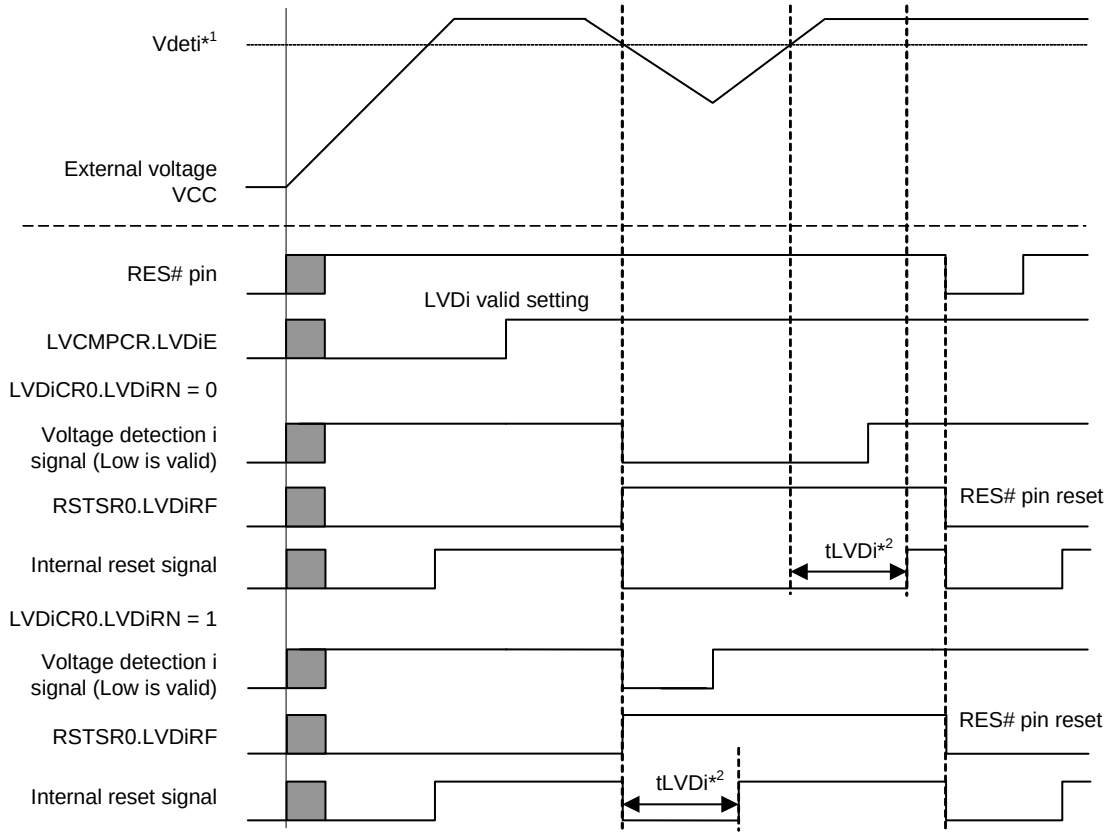
Likewise, when the voltage monitoring 2 interrupt/reset enable bit (LVD2RIE) is set to 1 (enabling generation of a reset or interrupt by the voltage detection circuit) and the voltage monitoring 2 circuit mode select bit (LVD2RI) is set to 1 (selecting generation of a reset in response to detection of a low voltage) in voltage monitoring 2 circuit control register 0 (LVD2CR0), the RSTSR0.LVD2RF flag is set to 1 and the voltage detection circuit generates a voltage monitoring 2 reset if VCC falls to or below Vdet2.

Timing for release from the voltage monitoring 1 reset state is selectable with the voltage monitoring 1 reset negation select bit (LVD1RN) in the LVD1CR0 register. When the LVD1CR0.LVD1RN bit is 0 and VCC has fallen to or below Vdet1, the CPU is released from the internal reset state and starts reset exception handling once the voltage monitoring 1 reset time (tLVD1) has elapsed after VCC has risen above Vdet1. When the LVD1CR0.LVD1RN bit is 1 and VCC has fallen to or below Vdet1, the CPU is released from the internal reset state and starts reset exception handling once the voltage monitoring 1 reset time (tLVD1) has elapsed.

Likewise, timing for release from the voltage monitoring 2 reset state is selectable by setting the voltage monitoring 2 reset negation select bit (LVD2RN) in the LVD2CR0 register. Detection levels Vdet1 and Vdet2 can be changed by settings in the voltage detection level select register (LVDLVLR).

Figure 6.2 shows examples of operations during voltage monitoring 1 and 2 resets.

For details on the voltage monitoring 1 reset and voltage monitoring 2 reset, refer to section 8, Voltage Detection Circuit (LVDAb).



Note: For details on the electrical characteristics, see the Electrical Characteristics section.
Note 1. Vdeti shows a detection level of voltage monitoring 1 reset and voltage monitoring 2 reset. (i = 1, 2)
Note 2. tLVDi shows a time for voltage monitoring 1 reset and voltage monitoring 2 reset. (i = 1, 2)

Figure 6.2 Operation Examples During Voltage Monitoring 1 Reset and Voltage Monitoring 2 Reset

6.3.4 Independent Watchdog Timer Reset

Independent watchdog timer reset is an internal reset generated by the independent watchdog timer.
Output of the independent watchdog timer reset from the independent watchdog timer can be selected by setting the IWDT reset control register (IWDTRCR) and option function select register 0 (OFS0).
When output of the independent watchdog timer reset is selected, an independent watchdog timer reset is generated if the independent watchdog timer underflows, or if data is written outside the refresh-permitted period. When the internal reset time (tRESW2) has elapsed after the independent watchdog timer reset has been generated, the internal reset is canceled and the CPU starts the reset exception handling.
For details on the independent watchdog timer reset, see section 22, Independent Watchdog Timer (IWDTa).

6.3.5 Software Reset

The software reset is an internal reset generated by the software reset circuit.
When A501h is written to SWRR, a software reset is generated. When the internal reset time (tRESW2) has elapsed after the software reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.

6.3.6 Determination of Cold/Warm Start

By reading the CWSF flag in RSTSR1, the type of reset processing caused can be identified; that is, whether a power-on reset has caused the reset processing (cold start) or a reset signal input during operation has caused the reset processing (warm start).

The CWSF flag in RSTSR1 is set to 0 when a power-on reset occurs (cold start); otherwise the flag is not set to 0. The flag is set to 1 when 1 is written to it through programming; it is not set to 0 even when 0 is written.

Figure 6.3 shows an example of cold/warm start determination operation.

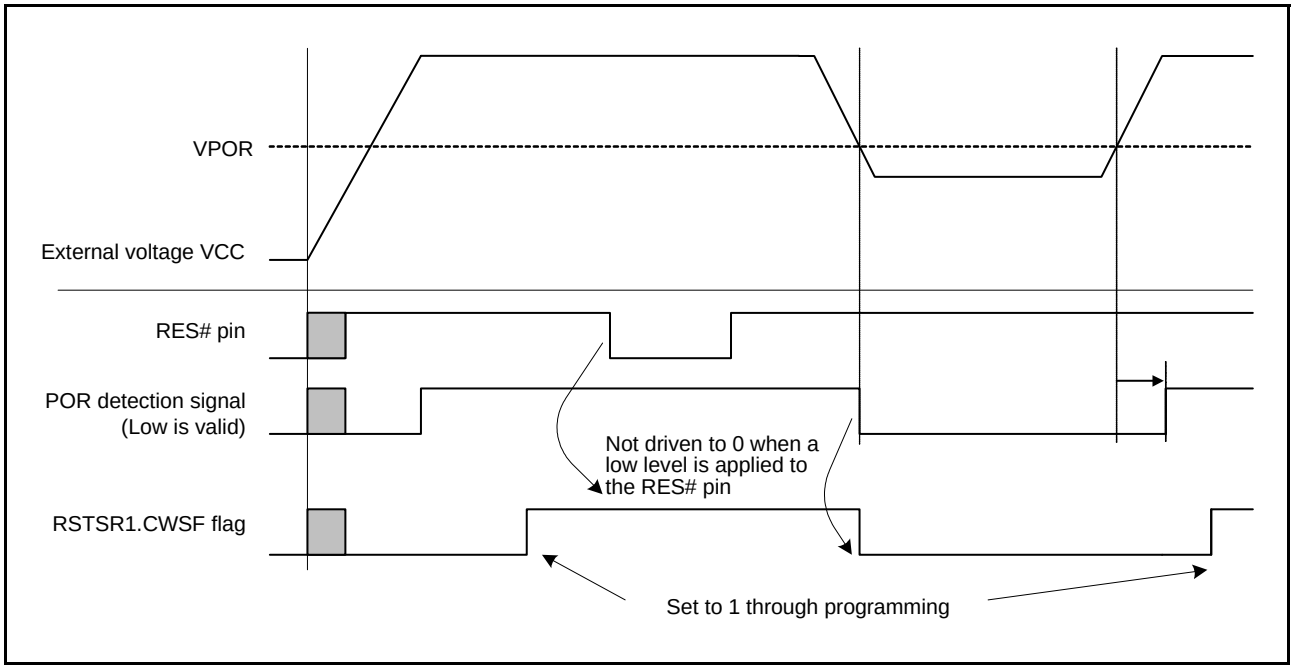


Figure 6.3 Example of Cold/Warm Start Determination Operation

6.3.7 Determination of Reset Generation Source

Reading RSTSR0 and RSTSR2 determines which reset was used to execute the reset exception handling. Figure 6.4 shows an example of the flow to identify a reset generation source.

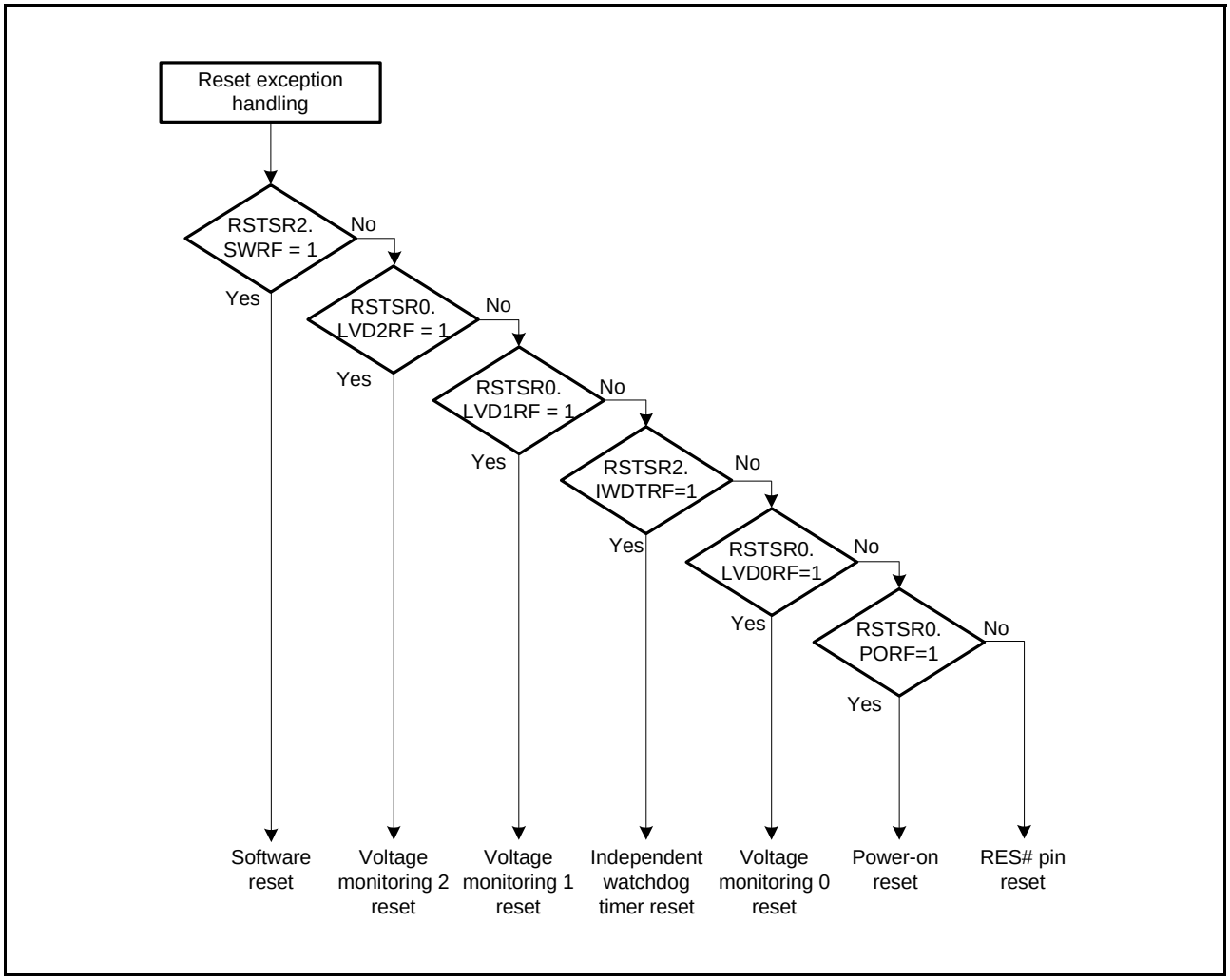


Figure 6.4 Example of Reset Generation Source Determination Flow

7. Option-Setting Memory (OFSM)

7.1 Overview

Option-setting memory (OFSM) refers to a set of registers that are provided for selecting the state of the microcontroller after a reset. The option-setting memory is allocated in the ROM.
Figure 7.1 shows the option-setting memory area.

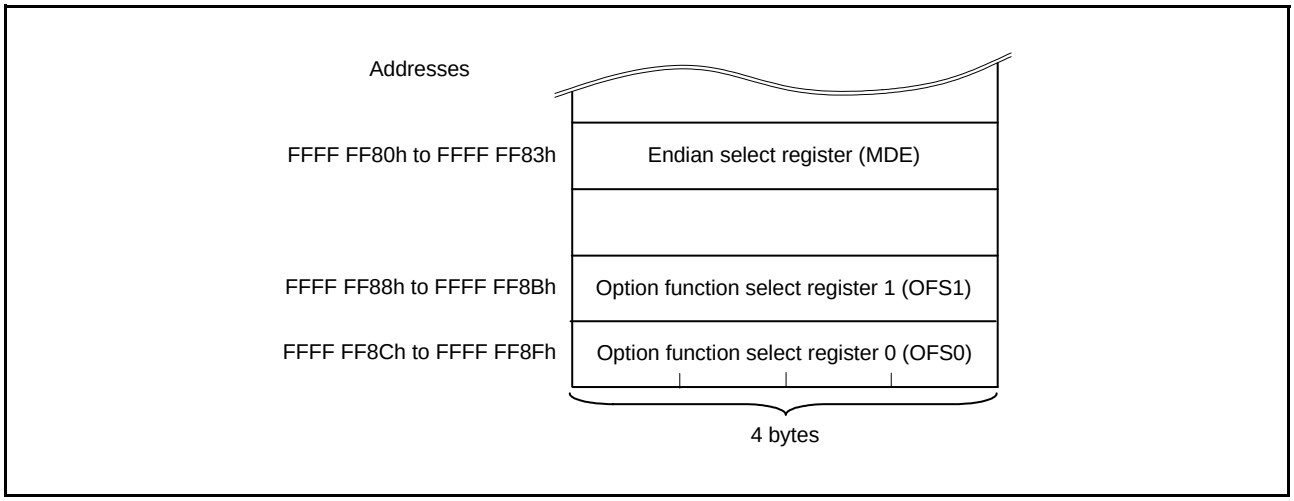


Figure 7.1 Option-Setting Memory Area

7.2 Register Descriptions

7.2.1 Option Function Select Register 0 (OFS0)

Address(es): OFSM.OFS0 FFFF FF8Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	—	IWDTSLCSTP	—	IWDTRSTIRQS	IWDTRPSS[1:0]		IWDTRPES[1:0]		IWDTCKS[3:0]				IWDTTOPS[1:0]		IWDTSTRT	—
	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b1	IWDTSTRT	IWDT Start Mode Select	0: IWDT is automatically activated in auto-start mode after a reset 1: IWDT is halted after a reset	R
b3, b2	IWDTTOPS[1:0]	IWDT Timeout Period Select	b3 b2 0 0: 128 cycles (007Fh) 0 1: 512 cycles (01FFh) 1 0: 1024 cycles (03FFh) 1 1: 2048 cycles (07FFh)	R
b7 to b4	IWDTCKS[3:0]	IWDT Clock Frequency Division Ratio Select	b7 b4 0 0 0 0: No division 0 0 1 0: Divide-by-16 0 0 1 1: Divide-by-32 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 0 1: Divide-by-256 Settings other than above are prohibited.	R
b9, b8	IWDTRPES[1:0]	IWDT Window End Position Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (No window end position setting)	R
b11, b10	IWDTRPSS[1:0]	IWDT Window Start Position Select	b11 b10 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (No window start position setting)	R
b12	IWDTRSTIRQS	IWDT Reset Interrupt Request Select	0: Non-maskable interrupt request is enabled 1: Reset is enabled	R
b13	—	Reserved	When reading, this bit returns the value written by the user. The write value should be 1.	R
b14	IWDTSLCSTP	IWDT Sleep Mode Count Stop Control	0: Counting stop is disabled 1: Counting stop is enabled when entering sleep, software standby, or deep sleep mode	R
b31 to b15	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. This register is set to a specified value after programming of the flash memory with the user program.

The OFS0 register is allocated in the ROM. Set this register at the same time as writing the program. After writing to the OFS0 register once, do not write to it again.

When erasing the block including the OFS0 register, the OFS0 register value becomes FFFF FFFFh.
The setting in the OFS0 register is ignored in boot mode, and this register functions similarly when it is set to FFFF FFFFh.

IWDTSTRT Bit (IWDT Start Mode Select)

This bit selects the mode in which the IWDT is activated after a reset (stopped state or activated in auto-start mode). When activated in auto-start mode, the OFS0 register setting for the IWDT is effective.

IWDTTOPS[1:0] Bits (IWDT Timeout Period Select)

These bits select the timeout period, i.e. the time it takes for the down-counter to underflow, as 128, 512, 1024, or 2048 cycles of the frequency-divided clock set by the IWDTCKS[3:0] bits. The time (number of IWDT-dedicated clock cycles) it takes to underflow after a refresh operation is determined by the combination of the IWDTCKS[3:0] bits and IWDTTOPS[1:0] bits.
For details, see section 22, Independent Watchdog Timer (IWDTa).

IWDTCKS[3:0] Bits (IWDT Clock Frequency Division Ratio Select)

These bits select, from 1/1, 1/16, 1/32, 1/64, 1/128, and 1/256, the division ratio of the prescaler to divide the frequency of the IWDT-dedicated clock. Using the setting of these bits together with the IWDTTOPS[1:0] bit setting, the IWDT counting period can be set from 128 to 524288 IWDT-dedicated clock cycles.
For details, see section 22, Independent Watchdog Timer (IWDTa).

IWDTRPES[1:0] Bits (IWDT Window End Position Select)

These bits select the position of the end of the window for the down-counter as 0%, 25%, 50%, or 75% of the value being counted by the counter. The value of the window end position must be smaller than the value of the window start position (window start position > window end position). If the value for the window end position is greater than the value for the window start position, only the value for the window start position is effective.
The counter values corresponding to the settings for the start and end positions of the window in the IWDTRPSS[1:0] and IWDTRPES[1:0] bits vary with the setting of the IWDTTOPS[1:0] bits.
For details, refer to section 22, Independent Watchdog Timer (IWDTa).

IWDTRPSS[1:0] Bits (IWDT Window Start Position Select)

These bits select the position where the window for the down-counter starts as 25%, 50%, 75%, or 100% of the value being counted (the point at which counting starts is 100% and the point at which an underflow occurs is 0%). The interval between the positions where the window starts and ends becomes the period in which refreshing is possible, and refreshing is not possible outside this period.
For details, refer to section 22, Independent Watchdog Timer (IWDTa).

IWDTRSTIRQS Bit (IWDT Reset Interrupt Request Select)

The setting of this bit selects the operation on an underflow of the down-counter or generation of a refresh error. Either an independent watchdog timer reset or a non-maskable interrupt request is selectable.
For details, refer to section 22, Independent Watchdog Timer (IWDTa).

IWDTSLCSTP Bit (IWDT Sleep Mode Count Stop Control)

This bit selects whether to stop counting when entering sleep, software standby, or deep sleep mode.
For details, see section 22, Independent Watchdog Timer (IWDTa).

7.2.2 Option Function Select Register 1 (OFS1)

Address(es): OFSM.OFS1 FFFF FF88h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	HOCOEN	—	—	—	—	—	LVDAS	VDSEL[1:0]	
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b1, b0	VDSEL[1:0]	Voltage Detection 0 Level Select	b1 b0 0 0: 3.84 V is selected 0 1: 2.82 V is selected 1 0: 2.51 V is selected Settings other than above are prohibited when the voltage detection 0 circuit is used.	R
b2	LVDAS	Voltage Detection 0 Circuit Start	0: Voltage monitoring 0 reset is enabled after a reset 1: Voltage monitoring 0 reset is disabled after a reset	R
b7 to b3	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R
b8	HOCOEN	HOCO Oscillation Enable	0: HOCO oscillation is enabled after a reset 1: HOCO oscillation is disabled after a reset	R
b31 to b9	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. This register is set to a specified value after programming of the flash memory with the user program.

The OFS1 register is allocated in the ROM. Set this register at the same time as writing the program. After writing, do not write additions to this register.
When erasing the block including the OFS1 register, the OFS1 register value becomes FFFF FFFFh.
The setting in the OFS1 register is ignored in boot mode, and this register functions similarly when it is set to FFFF FFFFh.

VDSEL[1:0] Bits (Voltage Detection 0 Level Select)

These bits select the voltage detection level to be monitored by the voltage detection 0 circuit.

LVDAS Bit (Voltage Detection 0 Circuit Start)

This bit selects whether the voltage monitoring 0 reset is enabled or disabled after a reset.
The Vdet0 voltage to be monitored by the voltage detection 0 circuit is selected by the VDSEL[1:0] bits.

HOCOEN Bit (HOCO Oscillation Enable)

This bit selects whether the HOCO oscillation is effective or not after a reset.
Setting the HOCOEN bit to 0 allows the HOCO oscillation to be started before the CPU starts operation, and therefore reduces the wait time for oscillation stabilization.
Note that even if the HOCOEN bit is set to 0, the system clock source is not switched to HOCO. The system clock source is switched to HOCO only by modifying the clock source select bits (SCKCR3.CKSEL[2:0]) from the CPU.
Also, when the HOCOEN bit is set to 0, the HOCO oscillation stabilization time (tHOCO) is secured by hardware, so the clock with the accuracy of the HOCO oscillation frequency (fHOCO) shown in Electrical Characteristics is supplied after release from the CPU reset state.

7.2.3 Endian Select Register (MDE)

Address(es): OFSM.MDE FFFF FF80h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	The value set by the user*1															

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	MDE[2:0]		
Value after reset:	The value set by the user*1															

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	MDE[2:0]	Endian Select	b2 b0 0 0 0: Big endian 1 1 1: Little endian Settings other than above are prohibited.	R
b31 to b3	—	Reserved	When reading, these bits return the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. This register is set to a specified value after programming of the flash memory with the user program.

The MDE register selects the endian for the CPU.MDE is allocated in the ROM. Set the register at the same time as writing the program. After writing to the register once, do not write to it again.
When erasing the block including the MDE register, the MDE register value becomes FFFF FFFFh.

MDE[2:0] Bits (Endian Select)

These bits select little endian or big endian for the CPU.

7.3 Usage Note

7.3.1 Setting Example of Option-Setting Memory

Since the option-setting memory is allocated in the ROM, values cannot be written by executing instructions. Write appropriate values when writing the program. An example of the settings is shown below.

- To set FFFF FFF8h in the OFS0 register

```
.ORG    0FFFFFF8CH
.LWORD  0FFFFFFF8H
```

Note: Programming formats vary depending on the compiler. Refer to the compiler manual for details.

8. Voltage Detection Circuit (LVDAb)

The voltage detection circuit (LVD) monitors the voltage level input to the VCC pin using a program.

8.1 Overview

In voltage detection 0, the detection voltage can be selected from three levels using option function select register 1 (OFS1).
In voltage detection 1, the detection voltage can be selected from nine levels using the voltage detection level select register (LVDLVLR).
In voltage detection 2, the detection voltage can be selected from four levels using the LVDLVLR register.
Voltage monitoring 0 reset, voltage monitoring 1 reset/interrupt, and voltage monitoring 2 reset/interrupt can be used.
Table 8.1 lists the specifications of the voltage detection circuit. Figure 8.1 is a block diagram of the voltage detection circuit. Figure 8.2 is a block diagram of the voltage monitoring 1 interrupt/reset circuit. Figure 8.3 is a block diagram of the voltage monitoring 2 interrupt/reset circuit.

Table 8.1 LVD Specifications

Item		Voltage Monitoring 0	Voltage Monitoring 1	Voltage Monitoring 2
VCC monitoring	Monitored voltage	Vdet0	Vdet1	Vdet2
	Detection target	Voltage drops past Vdet0	When voltage rises above or drops below Vdet1	When voltage rises above or drops below Vdet2
	Detection voltage	Voltage selectable from 3 levels using OFS1	Voltage selectable from 9 levels using the LVDLVLR.LVD1LVL[3:0] bits	Voltage selectable from 4 levels using the LVDLVLR.LVD2LVL[1:0] bits
	Monitoring flag	Not available	LVD1SR.LVD1MON flag: Monitors whether voltage is higher or lower than Vdet1	LVD2SR.LVD2MON flag: Monitors whether voltage is higher or lower than Vdet2
			LVD1SR.LVD1DET flag: Vdet1 passage detection	LVD2SR.LVD2DET flag: Vdet2 passage detection
Process upon voltage detection	Reset	Voltage monitoring 0 reset	Voltage monitoring 1 reset	Voltage monitoring 2 reset
		Reset when Vdet0 > VCC CPU restart after specified time with VCC > Vdet0	Reset when Vdet1 > VCC CPU restart timing selectable: after specified time with VCC > Vdet1 or Vdet1 > VCC	Reset when Vdet2 > VCC CPU restart timing selectable: after specified time with VCC > Vdet2 or after specified time with Vdet2 > VCC
	Interrupt	Not available	Voltage monitoring 1 interrupt	Voltage monitoring 2 interrupt
			Non-maskable or maskable interrupt is selectable	Non-maskable or maskable interrupt is selectable
			Interrupt request issued when Vdet1 > VCC and VCC > Vdet1 or either	Interrupt request issued when Vdet2 > VCC and VCC > Vdet2 or either

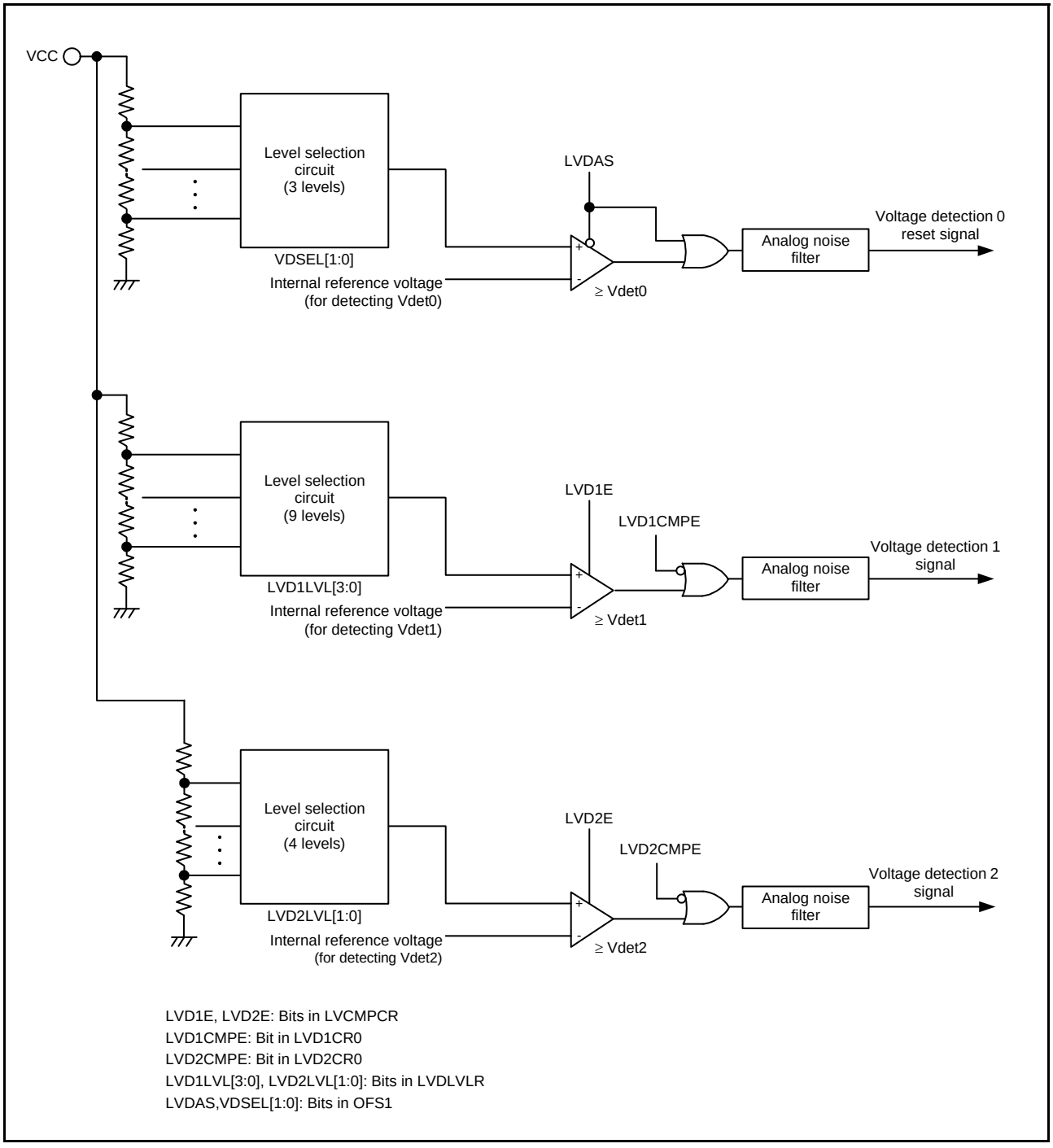


Figure 8.1 Block Diagram of the LVD

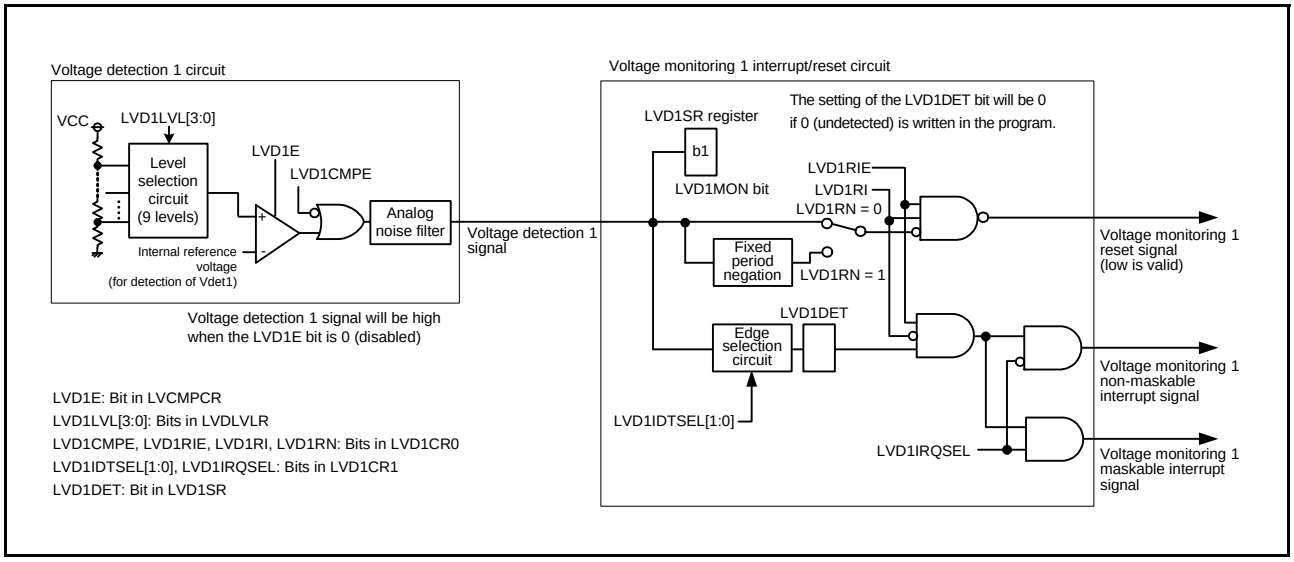


Figure 8.2 Block Diagram of Voltage Monitoring 1 Interrupt/Reset Circuit

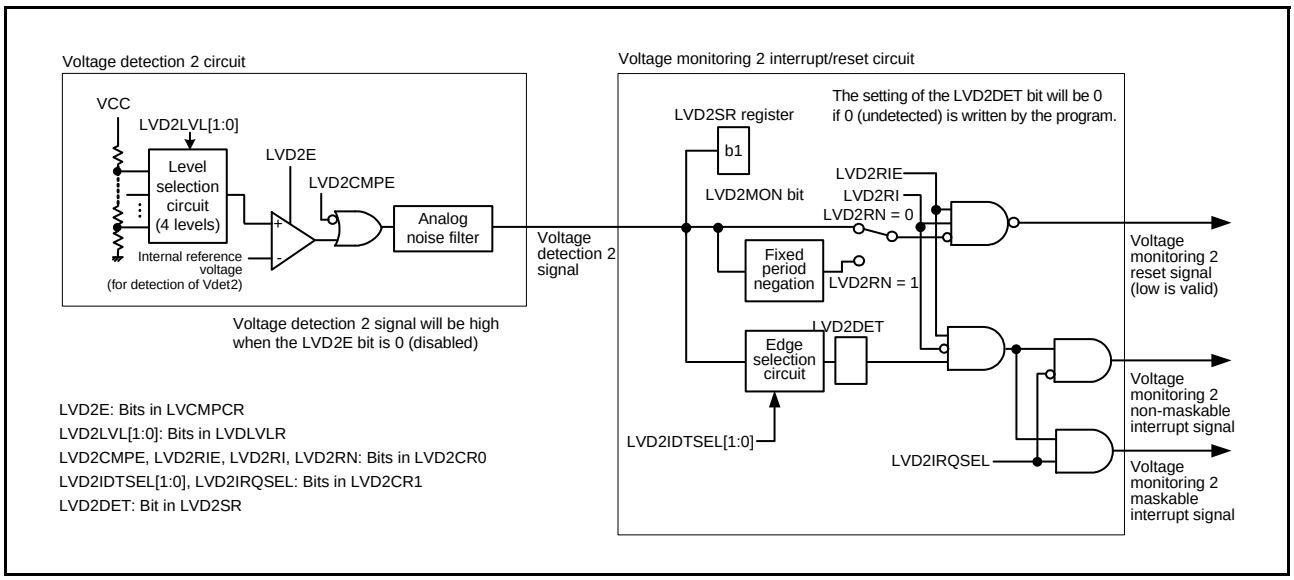


Figure 8.3 Block Diagram of Voltage Monitoring 2 Interrupt/Reset Circuit

8.2 Register Descriptions

8.2.1 Voltage Monitoring 1 Circuit Control Register 1 (LVD1CR1)

Address(es): 0008 00E0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	LVD1IRQSEL	LVD1IDTSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LVD1IDTSEL[1:0]	Voltage Monitoring 1 Interrupt Generation Condition Select	b1 b0 0 0: When VCC ≥ Vdet1 (rise) is detected 0 1: When VCC < Vdet1 (drop) is detected 1 0: When drop and rise are detected 1 1: Setting prohibited	R/W
b2	LVD1IRQSEL	Voltage Monitoring 1 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

8.2.2 Voltage Monitoring 1 Circuit Status Register (LVD1SR)

Address(es): 0008 00E1h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	LVD1MON	LVD1DET
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	LVD1DET	Voltage Monitoring 1 Voltage Change Detection Flag	0: Not detected 1: Vdet1 passage detection	R/(W) *1
b1	LVD1MON	Voltage Monitoring 1 Signal Monitor Flag	0: VCC < Vdet1 1: VCC ≥ Vdet1 or LVD1MON circuit is disabled	R
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to this bit. After writing 0 to this bit, it takes two system clock cycles for the bit to be read as 0.

LVD1DET Flag (Voltage Monitoring 1 Voltage Change Detection Flag)

The LVD1DET flag is enabled when the LVCMPCR.LVD1E bit is 1 (voltage detection 1 circuit enabled) and the LVD1CR0.LVD1CMPE bit is 1 (voltage monitoring 1 circuit comparison result output enabled).
The LVD1DET flag should be set to 0 after LVD1CR0.LVD1RIE is set to 0 (disabled). LVD1CR0.LVD1RIE can be set to 1 (enabled) again after a period of two or more cycles of PCLKB has elapsed.
With read access to an I/O register which access cycle number is defined by PCLKB, two or more cycles of PCLKB may have to be secured as waiting time.

LVD1MON Flag (Voltage Monitoring 1 Signal Monitor Flag)

The LVD1MON flag is enabled when the LVCMPCR.LVD1E bit is 1 (voltage detection 1 circuit enabled) and the LVD1CR0.LVD1CMPE bit is 1 (voltage monitoring 1 circuit comparison result output enabled).

8.2.3 Voltage Monitoring 2 Circuit Control Register 1 (LVD2CR1)

Address(es): 0008 00E2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	LVD2IRQSEL	LVD2IDTSEL [1:0]	
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	LVD2IDTSEL [1:0]	Voltage Monitoring 2 Interrupt Generation Condition Select	b1 b0 0 0:When VCC ≥ Vdet2 (rise) is detected 0 1:When VCC < Vdet2 (drop) is detected 1 0: When drop and rise are detected 1 1: Setting prohibited	R/W
b2	LVD2IRQSEL	Voltage Monitoring 2 Interrupt Type Select	0: Non-maskable interrupt 1: Maskable interrupt	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

8.2.4 Voltage Monitoring 2 Circuit Status Register (LVD2SR)

Address(es): 0008 00E3h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	LVD2MON	LVD2DET
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	LVD2DET	Voltage Monitoring 2 Voltage Change Detection Flag	0: Not detected 1: Vdet2 passage detection	R/(W) *1
b1	LVD2MON	Voltage Monitoring 2 Signal Monitor Flag	0: VCC < Vdet2 1: VCC ≥ Vdet2 or LVD2MON is disabled	R
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.
Note 1. Only 0 can be written to this bit. After writing 0 to this bit, it takes two system clock cycles for the bit to be read as 0.

LVD2DET Flag (Voltage Monitoring 2 Voltage Change Detection Flag)

The LVD2DET flag is enabled when the LVCMPCR.LVD2E bit is 1 (voltage detection 2 circuit enabled) and the LVD2CR0.LVD2CMPE bit is 1 (voltage monitoring 2 circuit comparison result output enabled).
The LVD2DET flag should be set to 0 after LVD2CR0.LVD2RIE is set to 0 (disabled). LVD2CR0.LVD2RIE can be set to 1 (enabled) again after a period of two or more cycles of PCLKB has elapsed.
With read access to an I/O register which access cycle number is defined by PCLKB, two or more cycles of PCLKB may have to be secured as waiting time.

LVD2MON Flag (Voltage Monitoring 2 Signal Monitor Flag)

The LVD2MON flag is enabled when the LVCMPCR.LVD2E bit is 1 (voltage detection 2 circuit enabled) and the LVD2CR0.LVD2CMPE bit is 1 (voltage monitoring 2 circuit comparison result output enabled).

8.2.5 Voltage Monitoring Circuit Control Register (LVCMPCR)

Address(es): 0008 C297h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	LVD2E	LVD1E	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	LVD1E	Voltage Detection 1 Enable	0: Voltage detection 1 circuit disabled 1: Voltage detection 1 circuit enabled	R/W
b6	LVD2E	Voltage Detection 2 Enable	0: Voltage detection 2 circuit disabled 1: Voltage detection 2 circuit enabled	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

LVD1E Bit (Voltage Detection 1 Enable)

When using voltage detection 1 interrupt/reset or the LVD1SR.LVD1MON flag, set the LVD1E bit to 1. The voltage detection 1 circuit starts once $Td_{(E-A)}$ passes after the LVD1E bit value is changed from 0 to 1.

LVD2E Bit (Voltage Detection 2 Enable)

When using voltage detection 2 interrupt/reset or the LVD2SR.LVD2MON flag, set the LVD2E bit to 1. The voltage detection 2 circuit starts once $Td_{(E-A)}$ passes after the LVD2E bit value is changed from 0 to 1.

8.2.6 Voltage Detection Level Select Register (LVDLVLR)

Address(es): 0008 C298h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LVD2LVL[1:0]		LVD1LVL[3:0]			
Value after reset:	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	LVD1LVL[3:0]	Voltage Detection 1 Level Select (Standard voltage during drop in voltage)	b3 b0 0 0 0 0: 4.29 V 0 0 0 1: 4.14 V 0 0 1 0: 4.02 V 0 0 1 1: 3.84 V 0 1 0 0: 3.10 V 0 1 0 1: 3.00 V 0 1 1 0: 2.90 V 0 1 1 1: 2.79 V 1 0 0 0: 2.68 V Settings other than those listed above are prohibited.	R/W
b5, b4	LVD2LVL[1:0]	Voltage Detection 2 Level Select (Standard voltage during drop in voltage)	b5 b4 0 0: 4.29V 0 1: 4.14V 1 0: 4.02V 1 1: 3.84V	R/W
b7, b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

When changing the LVDLVLR register, first set the LVCMPCR.LVD1E and LVCMPCR.LVD2E bits to 0 (voltage detection n circuit disabled) (n = 1, 2).

When a setting is made so that the voltage detection level range set by the LVD1LVL[3:0] bits overlaps with the range set by the LVD2LVL[1:0] bits, it cannot be specified which of LVD1 and LVD2 is used for voltage detection. For details on the voltage detection level range, refer to section 32, Electrical Characteristics.

8.2.7 Voltage Monitoring 1 Circuit Control Register 0 (LVD1CR0)

Address(es): 0008 C29Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	LVD1RN	LVD1RI	—	—	—	LVD1CMPE	—	LVD1RIE
Value after reset:	1	0	0	0	X	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	LVD1RIE	Voltage Monitoring 1 Interrupt/Reset Enable	0: Disabled 1: Enabled	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	LVD1CMPE	Voltage Monitoring 1 Circuit Comparison Result Output Enable	0: Voltage monitoring 1 circuit comparison results output disabled 1: Voltage monitoring 1 circuit comparison results output enabled	R/W
b3	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	LVD1RI	Voltage Monitoring 1 Circuit Mode Select	0: Voltage monitoring 1 interrupt occurs when the voltage passes Vdet1 1: Voltage monitoring 1 reset occurs when the voltage falls below Vdet1	R/W
b7	LVD1RN	Voltage Monitoring 1 Reset Negation Select	0: Negation follows a stabilization time (tLVD1) after VCC > Vdet1 is detected. 1: Negation follows a stabilization time (tLVD1) after assertion of the voltage monitoring 1 reset.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

LVD1RIE Bit (Voltage Monitoring 1 Interrupt/Reset Enable)

The LVD1RIE bit is enabled when the LVCMPCR.LVD1E bit is set to 1 (voltage detection 1 circuit enabled) and the LVD1CMPE bit is set to 1 (voltage monitoring 1 circuit comparison results output enabled).

Ensure that neither a voltage monitoring 1 reset nor a voltage monitoring 1 non-maskable interrupt is generated during programming or erasure of the flash memory.

LVD1RN Bit (Voltage Monitoring 1 Reset Negation Select)

If the LVD1RN bit is to be set to 1 (negation follows a stabilization time after assertion of the voltage monitoring 1 reset), set the LOCOCR.LCSTP bit to 0 (LOCO is operating). Furthermore, if a transition to software standby mode, the only possible value for the LVD1RN bit is 0 (negation follows a stabilization time after VCC > Vdet1 is detected). Do not set the LVD1RN bit to 1 (negation follows a stabilization time after assertion of the voltage monitoring 1 reset).

8.2.8 Voltage Monitoring 2 Circuit Control Register 0 (LVD2CR0)

Address(es): 0008 C29Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	LVD2RN	LVD2RI	—	—	—	LVD2CMPE	—	LVD2RIE
Value after reset:	1	0	0	0	X	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	LVD2RIE	Voltage Monitoring 2 Interrupt/Reset Enable	0: Disabled 1: Enabled	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	LVD2CMPE	Voltage Monitoring 2 Circuit Comparison Result Output Enable	0: Voltage monitoring 2 circuit comparison results output disabled 1: Voltage monitoring 2 circuit comparison results output enabled	R/W
b3	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	LVD2RI	Voltage Monitoring 2 Circuit Mode Select	0: Voltage monitoring 2 interrupt during Vdet2 passage 1: Voltage monitoring 2 reset enabled when the voltage falls to and below Vdet2	R/W
b7	LVD2RN	Voltage Monitoring 2 Reset Negation Select	0: Negation follows a stabilization time (tLVD2) after VCC > Vdet2 is detected. 1: Negation follows a stabilization time (tLVD2) after assertion of the voltage monitoring 2 reset.	R/W

Note: Set the PRCR.PRC3 bit to 1 (write enabled) before rewriting this register.

LVD2RIE Bit (Voltage Monitoring 2 Interrupt/Reset Enable)

The LVD2RIE bit is enabled when the LVCMPCR.LVD2E bit is set to 1 (voltage detection 2 circuit enabled) and the LVD2CMPE bit is set to 1 (voltage monitoring 2 circuit comparison results output enabled).
Ensure that neither a voltage monitoring 2 reset nor a voltage monitoring 2 non-maskable interrupt is generated during programming or erasure of the flash memory.

LVD2RN Bit (Voltage Monitoring 2 Reset Negation Select)

If the LVD2RN bit is to be set to 1 (negation follows a stabilization time after assertion of the voltage monitoring 2 reset), set the LOCOCR.LCSTP bit to 0 (LOCO is operating). Furthermore, if a transition to software standby mode, the only possible value for the LVD2RN bit is 0 (negation follows a stabilization time after VCC > Vdet2 is detected). Do not set the LVD2RN bit to 1 (negation follows a stabilization time after assertion of the voltage monitoring 2 reset).

8.3 VCC Input Voltage Monitor

8.3.1 Monitoring Vdet0

Monitoring Vdet0 is not possible.

8.3.2 Monitoring Vdet1

After making the following settings, the LVD1SR.LVD1MON flag can be used to monitor the results of comparison by voltage monitor 1.

(1) Specify the detection voltage by setting the LVDLVLR.LVD1LVL[3:0] bits (voltage detection 1 level select).
(2) Set the LVCMPCR.LVD1E bit to 1 (voltage detection 1 circuit enabled).
(3) After waiting for $Td_{(E-A)}$, set the LVD1CR0.LVD1CMPE bit to 1 (voltage monitoring 1 circuit comparison results output enabled).

8.3.3 Monitoring Vdet2

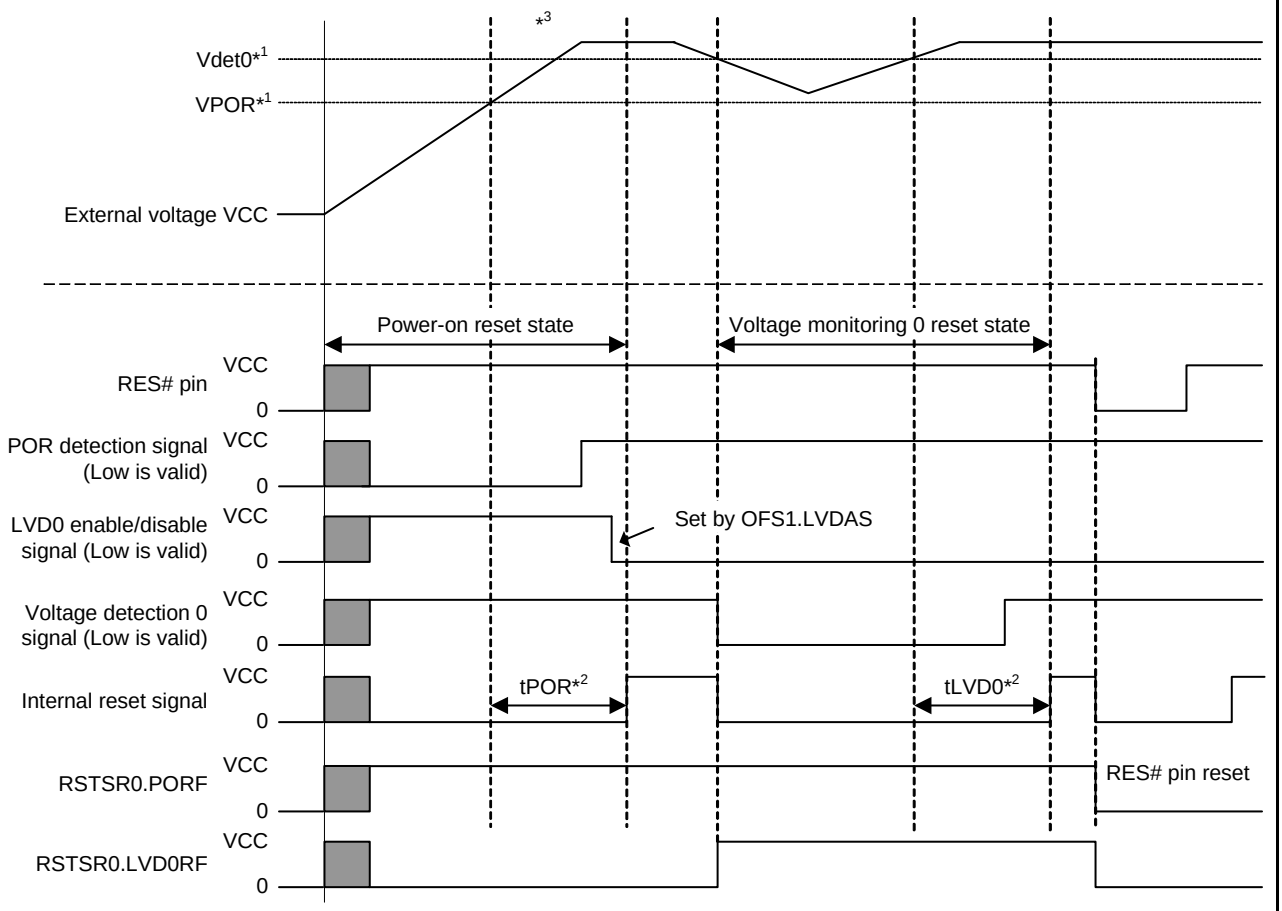
After making the following settings, the LVD2SR.LVD2MON flag can be used to monitor the results of comparison by voltage monitor 2.

(1) Specify the detection voltage by setting the LVDLVLR.LVD2LVL[1:0] bits (voltage detection 2 level select).
(2) Set the LVCMPCR.LVD2E bit to 1 (voltage detection 2 circuit enabled).
(3) After waiting for $Td_{(E-A)}$, set the LVD2CR0.LVD2CMPE bit to 1 (voltage monitoring 2 circuit comparison results output enabled).

8.4 Reset from Voltage Monitor 0

When using the reset from voltage monitor 0, clear the voltage detection 0 circuit start bit (OFS1.LVDAS) to 0 (enabling the voltage monitor 0 reset after a reset).

Figure 8.4 shows an example of operations for a voltage monitoring 0 reset.



Note: For details on the electrical characteristics, see the Electrical Characteristics section.
Note 1. Vdet0 indicates the detection level for a voltage monitoring 0 reset and VPOR indicates the detection level for a power-on reset.
Note 2. tPOR indicates the period of a power-on reset and tLVD0 indicates the period of a voltage monitoring 0 reset.
Note 3. At the time the power-supply voltage rises, VCC must rise to at least the minimum guaranteed voltage before release from the POR reset state.

Figure 8.4 Example of Voltage Monitoring 0 Reset Operation

8.5 Interrupt and Reset from Voltage Monitoring 1

Table 8.2 shows the procedures for setting bits related to the voltage monitoring 1 interrupt and voltage monitoring 1 reset. Table 8.3 shows the procedures for stopping bits related to the voltage monitoring 1 interrupt and voltage monitoring 1 reset. Figure 8.5 shows an example of operations for a voltage monitoring 1 interrupt. For the operation of the voltage monitoring 1 reset, see Figure 6.2 in section 6, Resets.

Table 8.2 Procedures for Setting Bits Related to the Voltage Monitoring 1 Interrupt and Voltage Monitoring 1 Reset

Step	Voltage Monitoring 1 Interrupt	Voltage Monitoring 1 Reset
1*1	Select the detection voltage by setting the LVDLVLR.LVD1LVL[3:0] bits.	
2*1	Set the LVD1CR0.LVD1RI bit to 0 (voltage monitoring 1 interrupt).	Set the LVD1CR0.LVD1RI bit to 1 (voltage monitoring 1 reset). Select the type of reset negation by setting the LVD1CR0.LVD1RN bit.
3	Select the timing of interrupt requests by setting the LVD1CR1.LVD1IDTSEL[1:0] bits. Select the type of interrupt by setting the LVD1CR1.LVD1IRQSEL bit.	—
4	—	Set the LVD1CR0.LVD1RIE bit to 1 (voltage monitoring 1 interrupt/reset enabled).
5*1	Set the LVCMPCR.LVD1E bit to 1 (voltage detection 1 circuit enabled).	
6*1	Wait for at least $Td_{(E-A)}$.	
7	Set the LVD1CR0.LVD1CMPE bit to 1 (voltage monitoring 1 circuit comparison results output enabled).	
8	Wait for at least 2 µs.	—
9	Set the LVD1SR.LVD1DET bit to 0.	—
10	Set the LVD1CR0.LVD1RIE bit to 1 (voltage monitoring 1 interrupt/reset enabled).	—

Note 1. Steps 1, 2, 5, and 6 are not required if operation is with the setting to select the voltage monitoring 1 interrupt (LVD1CR0.LVD1RI = 0) and operation can be restarted by simply changing the settings of the LVD1CR1.LVD1IRQSEL and LVD1IDTSEL[1:0] bits after monitoring is stopped or if restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitoring 1 reset (LVD1CR0.LVD1RI = 1), proceed through all steps from 1 to 10.

Table 8.3 Procedures for Stopping Bits Related to the Voltage Monitoring 1 Interrupt and Voltage Monitoring 1 Reset

Step	Voltage Monitoring 1 Interrupt	Voltage Monitoring 1 Reset
1	Set the LVD1CR0.LVD1RIE bit to 0 (voltage monitoring 1 interrupt/reset disabled).	—
2	Set the LVD1CR0.LVD1CMPE bit to 0 (voltage monitoring 1 circuit comparison results output disabled).	
3*1	Set the LVCMPCR.LVD1E bit to 0 (voltage detection 1 circuit disabled).	
4	—	Set the LVD1CR0.LVD1RIE bit to 0 (voltage monitoring 1 interrupt/reset disabled).
5	Modify settings of bits related to the voltage detection circuit registers other than LVCMPCR.LVD1E, LVD1CR0.LVD1RIE, and LVD1CR0.LVD1CMPE.	

Note 1. Step 3 is not required if operation is with the setting to select the voltage monitoring 1 interrupt (LVD1CR0.LVD1RI = 0) and operation can be restarted by simply changing the settings of the LVD1CR1.LVD1IRQSEL and LVD1IDTSEL[1:0] bits after monitoring is stopped or if restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitoring 1 reset (LVD1CR0.LVD1RI = 1), proceed through all steps from 1 to 5.

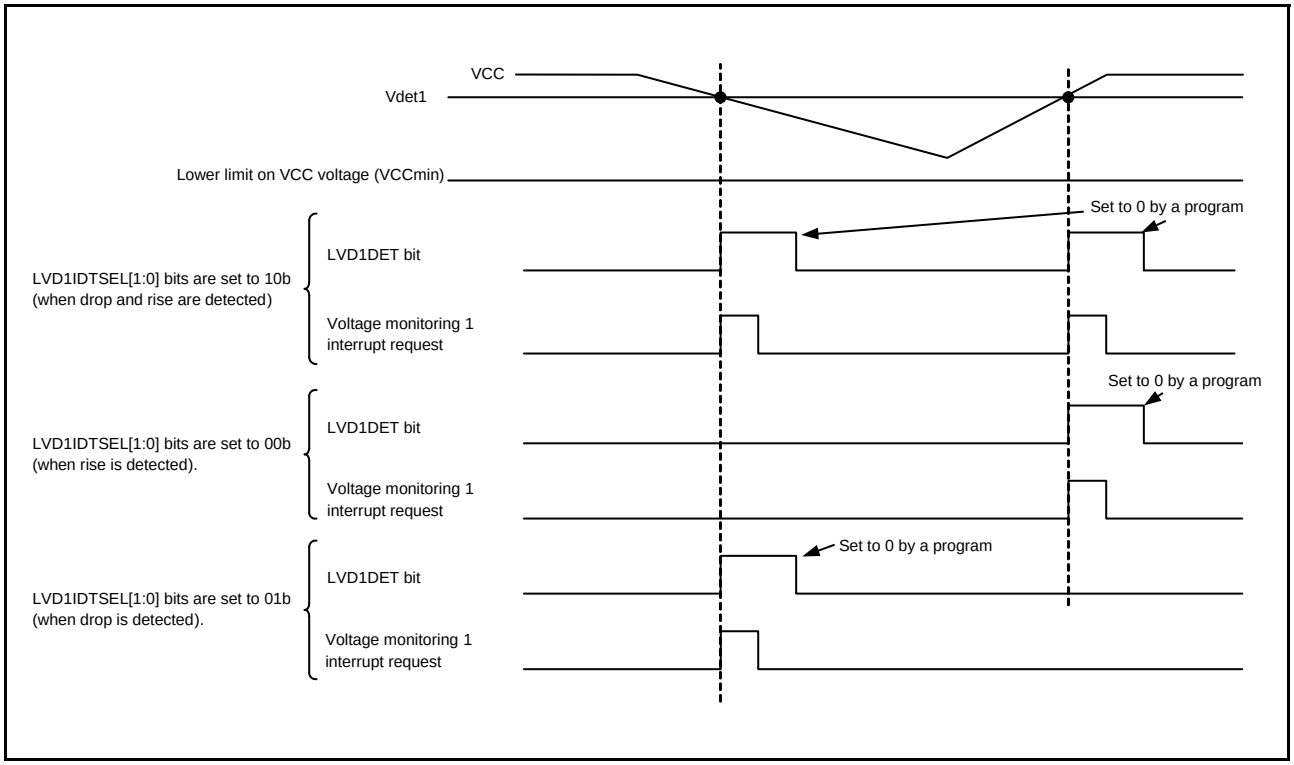


Figure 8.5 Example of Voltage Monitoring 1 Interrupt Operation

8.6 Interrupt and Reset from Voltage Monitoring 2

Table 8.4 shows the procedures for setting bits related to the voltage monitoring 2 interrupt and voltage monitoring 2 reset. Table 8.5 shows the procedure for stopping bits related to the voltage monitoring 2 interrupt and voltage monitoring 2 reset. Figure 8.6 shows an example of operations for a voltage monitoring 2 interrupt. For the operation of the voltage monitoring 2 reset, see Figure 6.2 in section 6, Resets.

Table 8.4 Procedures for Setting Bits Related to the Voltage Monitoring 2 Interrupt and Voltage Monitoring 2 Reset

Step	Voltage Monitoring 2 Interrupt	Voltage Monitoring 2 Reset
1*1	Select the detection voltage by setting the LVDLVLR.LVD2LVL[1:0] bits.	
2*1	Set the LVD2CR0.LVD2RI bit to 0 (voltage monitoring 2 interrupt).	Set the LVD2CR0.LVD2RI bit to 1 (voltage monitoring 2 reset). Select the type of reset negation by setting the LVD2CR0.LVD2RN bit.
3	Select the timing of interrupt requests by setting the LVD2CR1.LVD2IDTSEL[1:0] bits. Select the type of interrupt by setting the LVD2CR1.LVD2IRQSEL bit.	—
4	—	Set the LVD2CR0.LVD2RIE bit to 1 (voltage monitoring 2 interrupt/reset enabled).
5*1	Set the LVCMPCR.LVD2E bit to 1 (voltage detection 2 circuit enabled).	
6*1	Wait for at least $Td_{(E-A)}$.	
7	Set the LVD2CR0.LVD2CMPE bit to 1 (voltage monitoring 2 circuit comparison results output enabled).	
8	Wait for at least 2 µs.	—
9	Set the LVD2SR.LVD2DET bit to 0.	—
10	Set the LVD2CR0.LVD2RIE bit to 1 (voltage monitoring 2 interrupt/reset enabled)	—

Note 1. Steps 1, 2, 5, and 6 are not required if operation is with the setting to select the voltage monitoring 2 interrupt (LVD2CR0.LVD2RI = 0) and operation can be restarted by simply changing the settings of the LVD2CR1.LVD2IRQSEL and LVD2IDTSEL[1:0] bits after monitoring is stopped or if restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitoring 2 reset (LVD2CR0.LVD2RI = 1), proceed through all steps from 1 to 10.

Table 8.5 Procedures for Stopping Bits Related to the Voltage Monitoring 2 Interrupt and Voltage Monitoring 2 Reset

Step	Voltage Monitoring 2 Interrupt	Voltage Monitoring 2 Reset
1	Set the LVD2CR0.LVD2RIE bit to 0 (voltage monitoring 2 interrupt/reset disabled).	—
2	Set the LVD2CR0.LVD2CMPE bit to 0 (voltage monitoring 2 circuit comparison results output disabled).	
3*1	Set the LVCMPCR.LVD2E bit to 0 (voltage monitoring 2 circuit disabled).	
4	—	Set the LVD2CR0.LVD2RIE bit to 0 (voltage monitoring 2 interrupt/reset disabled).
5	Modify settings of bits related to the voltage detection circuit registers other than LVCMPCR.LVD2E, LVD2CR0.LVD2RIE, and LVD2CR0.LVD2CMPE.	

Note 1. Step 3 is not required if operation is with the setting to select the voltage monitoring 2 interrupt (LVD2CR0.LVD2RI = 0) and operation can be restarted by simply changing the settings of the LVD2CR1.LVD2IRQSEL and LVD2IDTSEL[1:0] bits after monitoring is stopped or if restarting is in a case where the settings related to the voltage-detection circuit were not changed after monitoring was stopped. When changes are to be made and operation is with the setting to select the voltage monitoring 2 reset (LVD2CR0.LVD2RI = 1), proceed through all steps from 1 to 5.

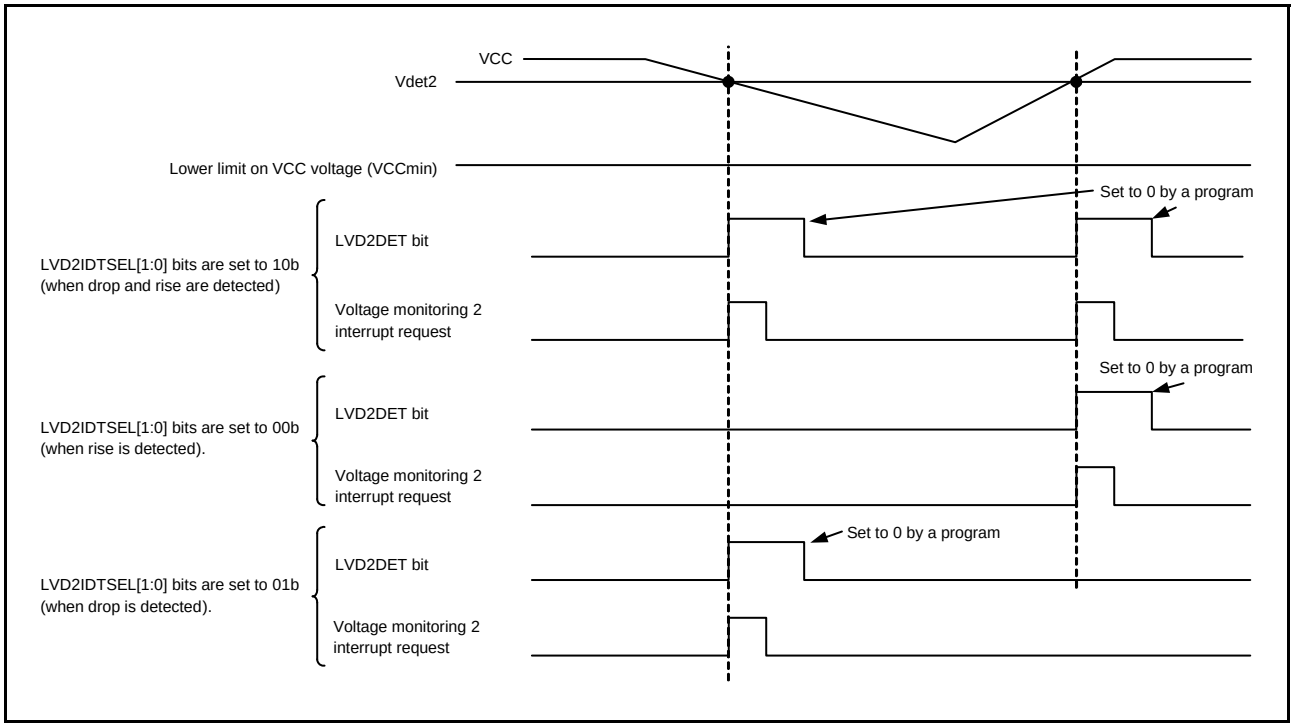


Figure 8.6 Example of Voltage Monitoring 2 Interrupt Operation

9. Clock Generation Circuit

9.1 Overview

This MCU incorporates a clock generation circuit.

Table 9.1 lists the specifications of the clock generation circuit. Figure 9.1 shows a block diagram of the clock generation circuit.

Table 9.1 Specifications of Clock Generation Circuit

Item	Specification
Uses	• Generates the system clock (ICLK) to be supplied to the CPU, DTC, ROM, and RAM. • Generates the peripheral module clocks (PCLKB, and PCLKD) to be supplied to peripheral modules. The peripheral module clock PCLKD is for the S12AD and PCLKB is for other modules. • Generates the FlashIF clock (FCLK) to be supplied to the FlashIF. • Generates the CAC clock (CACCLK) to be supplied to the CAC. • Generates the IWDT-dedicated low-speed clock (IWDTCLK) to be supplied to the IWDT.
Operating frequencies*1	• ICLK: 32 MHz (max)*2 • PCLKB: 32 MHz (max) • PCLKD: 32 MHz (max) • FCLK: 1 to 32 MHz (for programming and erasing the ROM and E2 DataFlash) 32 MHz (max) (for reading from the E2 DataFlash) • CACCLK: Same frequency as each oscillator • IWDTCLK: 15 kHz
Main clock oscillator*3	• Resonator frequency: 1 to 20 MHz • External clock input frequency: 20 MHz (max) • Connectable resonator or additional circuit: ceramic resonator, crystal • Connection pins: EXTAL, XTAL • Oscillation stop detection function: When an oscillation stop is detected with the main clock, the system clock source is switched to LOCO and MTU output can be forcedly driven to the high-impedance. • Drive capacity switching function
PLL circuit	• Input clock source: Main clock • Input pulse frequency division ratio: Selectable from 1, 2, and 4 • Input frequency: 4 to 8 MHz • Frequency multiplication ratio: Selectable from 4 to 8 (increments of 0.5) • VCO oscillation frequency: 24 to 32 MHz
High-speed on-chip oscillator (HOCO)	Oscillation frequency: 32 MHz
Low-speed on-chip oscillator (LOCO)	Oscillation frequency: 4 MHz
IWDT-dedicated on-chip oscillator	Oscillation frequency: 15 kHz

Note 1. The maximum operating frequency in high-speed operating mode. For the maximum operating frequency in the other operating modes, refer to section 11.2.5, Operating Power Control Register (OPCCR).
Note 2. The relationship of frequencies must be set as follows. ICLK: FCLK, PCLKB, and PCLKD = 1: N (N is an integer).
Note 3. When oscillating the PLL at 32 MHz, the frequency of the main clock oscillator is limited to 8 or 16 MHz.

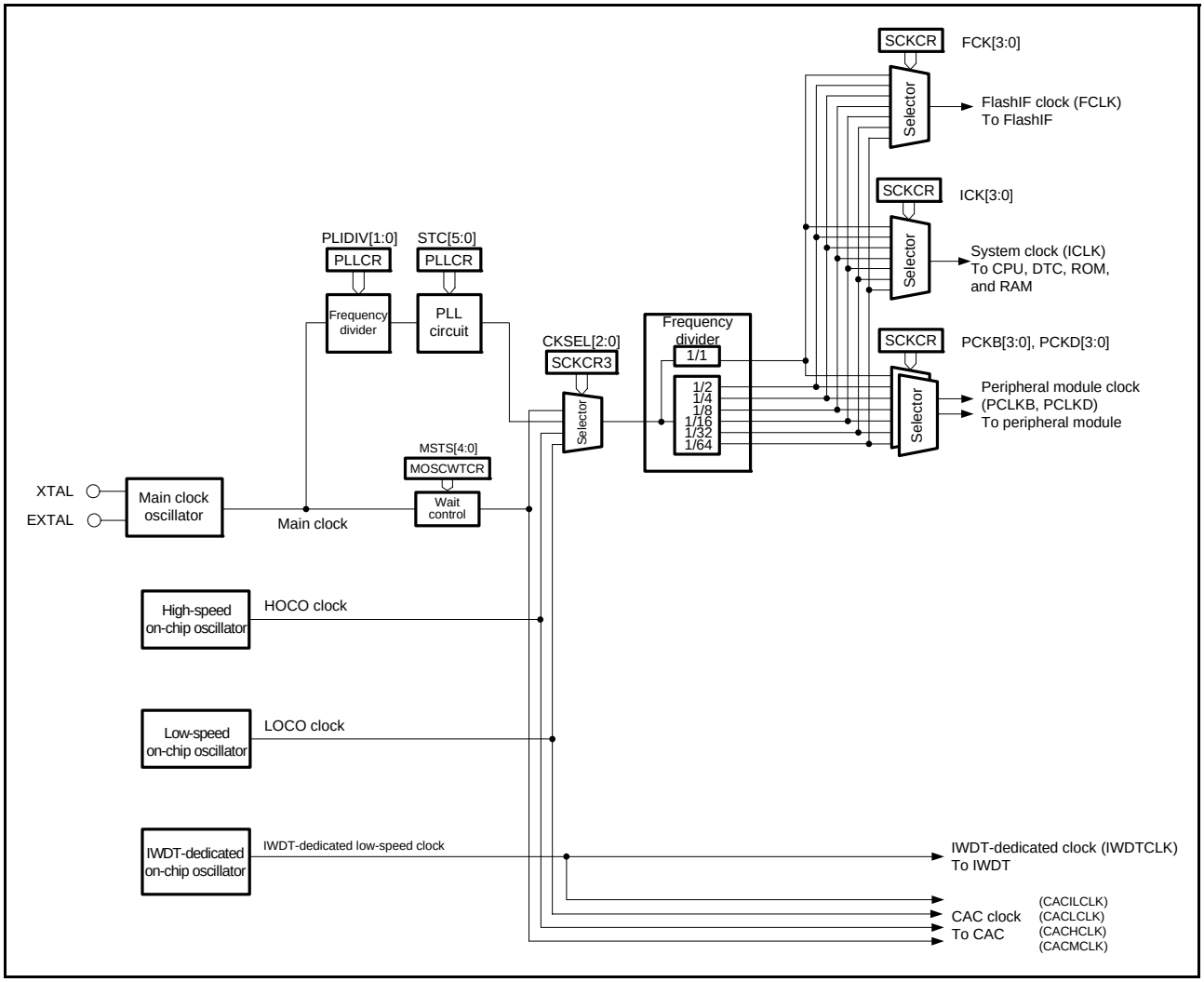


Figure 9.1 Block Diagram of Clock Generation Circuit

Table 9.2 lists the I/O pins of the clock generation circuit.

Table 9.2 I/O Pins of Clock Generation Circuit

Pin Name	I/O	Description
XTAL	Output	These pins are used to connect a crystal. The EXTAL pin can also be used to input an external clock. For details, refer to section 9.3.2, External Clock Input.
EXTAL	Input	

9.2 Register Descriptions

9.2.1 System Clock Control Register (SCKCR)

Address(es): 0008 0020h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	FCK[3:0]				ICK[3:0]				—	—	—	—	—	—	—	—
Value after reset:	0	0	1	1	0	0	1	1	0	0	0	0	0	0	0	0

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	PCKB[3:0]				—	—	—	—	PCKD[3:0]			
Value after reset:	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	PCKD[3:0]	Peripheral Module Clock D (PCLKD) Select	b3 b0 0 0 0 0: ×1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11 to b8	PCKB[3:0]	Peripheral Module Clock B (PCLKB) Select	b11 b8 0 0 0 0: ×1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W
b19 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b27 to b24	ICK[3:0]	System Clock (ICLK) Select	b27 b24 0 0 0 0: ×1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W
b31 to b28	FCK[3:0]	FlashIF Clock (FCLK) Select	b31 b28 0 0 0 0: ×1 0 0 0 1: ×1/2 0 0 1 0: ×1/4 0 0 1 1: ×1/8 0 1 0 0: ×1/16 0 1 0 1: ×1/32 0 1 1 0: ×1/64 Settings other than above are prohibited.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

This register cannot be rewritten while the flash memory is being programmed or erased.
When an instruction for writing to SCKCR or SCKCR3 is to follow writing to the SCKCR register, do so in accord with the procedure below.

1. Write to the SCKCR register.
2. Confirm that the value has actually been written to the SCKCR register.
3. Proceed to the next step.

PCKD[3:0] Bits (Peripheral Module Clock D (PCLKD) Select)

These bits select the frequency of peripheral module clock D (PCLKD).

PCKB[3:0] Bits (Peripheral Module Clock B (PCLKB) Select)

These bits select the frequency of peripheral module clock B (PCLKB).

ICK[3:0] Bits (System Clock (ICLK) Select)

These bits select the frequency of the system clock (ICLK).

FCK[3:0] Bits (FlashIF Clock (FCLK) Select)

These bits select the frequency of the FlashIF clock (FCLK).

9.2.2 System Clock Control Register 3 (SCKCR3)

Address(es): 0008 0026h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CKSEL[2:0]			—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	CKSEL[2:0]	Clock Source Select	b10 b8 0 0 0: LOCO 0 0 1: HOCO 0 1 0: Main clock oscillator 1 0 0: PLL circuit Settings other than above are prohibited.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

This register cannot be rewritten while the flash memory is being programmed or erased.

CKSEL[2:0] Bits (Clock Source Select)

These bits select the source of the system clock (ICLK), peripheral module clock (PCLKB, and PCLKD), FlashIF clock (FCLK), from low-speed on-chip oscillator (LOCO), high-speed on-chip oscillator (HOCO), the main clock oscillator, and the PLL circuit.

Transitions to clock sources which are not in operation are prohibited.

9.2.3 PLL Control Register (PLLCR)

Address(es): 0008 0028h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	STC[5:0]						—	—	—	—	—	—	PLIDIV[1:0]	
Value after reset:	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PLIDIV[1:0]	PLL Input Frequency Division Ratio Select	b1 b0 0 0: ×1 0 1: ×1/2 1 0: ×1/4 1 1: Setting prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	STC[5:0]	Frequency Multiplication Factor Select	b13 b8 0 0 0 1 1 1: ×4 0 0 1 0 0 0: ×4.5 0 0 1 0 0 1: ×5 0 0 1 0 1 0: ×5.5 0 0 1 0 1 1: ×6 0 0 1 1 0 0: ×6.5 0 0 1 1 0 1: ×7 0 0 1 1 1 0: ×7.5 0 0 1 1 1 1: ×8 Settings other than above are prohibited.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Writing to the PLLCR is prohibited when the PLLCR2.PLLEN bit is 0 (PLL is operating).

PLIDIV[1:0] Bits (PLL Input Frequency Division Ratio Select)

These bits select the frequency division ratio of the PLL clock source.
Set these bits so that the frequency of PLL input signal is within the range of 4 MHz to 8 MHz.

STC[5:0] Bits (Frequency Multiplication Factor Select)

These bits select the frequency multiplication factor of the PLL circuit.
Set these bits so that the PLL oscillation frequency is within the range of 24 MHz to 32 MHz.

9.2.4 PLL Control Register 2 (PLLCR2)

Address(es): 0008 002Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PLLEN
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	PLLEN	PLL Stop Control	0: PLL is operating. 1: PLL is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

PLLEN Bit (PLL Stop Control)

This bit runs or stops the PLL circuit.

After setting the PLLEN bit to 0 (PLL is operating), confirm that the OSCOVFSR.PLOVF bit is 1 before switching the system clock to the PLL clock.

That is, a fixed time for stabilization is required after the setting for PLL operation. A fixed time is also required for oscillation to stop after the setting to stop PLL operation. Accordingly, take note of the following limitations when starting and stopping PLL operation.

- After stopping the PLL, confirm that the OSCOVFSR.PLOVF bit is 0 before restarting the PLL.
- Confirm that the PLL is operating and that the OSCOVFSR.PLOVF bit is 1 before stopping the PLL.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.PLOVF bit is 1 before executing a WAIT instruction to place the MCU in software standby mode.
- After stopping the PLL, confirm that the OSCOVFSR.PLOVF bit is 0 and execute a WAIT instruction before entering software standby mode.

When the PLL clock is selected by the SCKCR3.CKSEL[2:0] bits, do not set the PLLEN bit (PLL is stopped) to 1.

9.2.5 Main Clock Oscillator Control Register (MOSCCR)

Address(es): 0008 0032h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	MOSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	MOSTP	Main Clock Oscillator Stop	0: Main clock oscillator is operating. 1: Main clock oscillator is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

Set this register after setting up the main clock oscillator wait control register.

MOSTP Bit (Main Clock Oscillator Stop)

This bit runs or stops the main clock oscillator.

After setting the MOSTP bit to 0 (main clock oscillator is operating), read the OSCOVFSR.MOOVF bit to confirm that is has become 1, and then use the main clock.

For the main clock oscillator, a fixed time is required for oscillation to become stable after the settings for operation have been made. Furthermore, a fixed time is required for oscillation to actually stop after the settings to stop oscillation have been made. Accordingly, take note of the following limitations when starting and stopping operation.

- After stopping the main clock oscillator, confirm that the OSCOVFSR.MOOVF bit is 0 before restarting the main clock oscillator.
- Confirm that the main clock oscillator is operating and that the OSCOVFSR.MOOVF bit is 1 before stopping the main clock oscillator.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.MOOVF bit is 1 and execute a WAIT instruction in order to operate the main clock oscillator and place the MCU in software standby mode.
- After stopping the main clock oscillator, confirm that the OSCOVFSR.MOOVF bit is 0 and execute a WAIT instruction before entering software standby mode.

Do not set the MOSTP bit to 1 when one of the following condition is met.

- When the main clock is selected as the clock source for the system clock (the SCKCR3.CKSEL[2:0] bits are 010b)
- When the PLL clock is selected as the clock source for the system clock (the SCKCR3.CKSEL[2:0] bits are 100b)
- When PLL is operating (the PLLCR2.PLLEN bit is 0)

9.2.6 Low-Speed On-Chip Oscillator Control Register (LOCOCR)

Address(es): 0008 0034h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	LCSTP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	LCSTP	LOCO Stop	0: LOCO is operating. 1: LOCO is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

LCSTP Bit (LOCO Stop)

This bit runs or stops the LOCO.

After the setting of the LCSTP bit has been changed so that the LOCO operates, only start using the LOCO clock after the LOCO clock oscillation stabilization time (t_{LOCO}) has elapsed.

That is, a fixed time for stabilization of oscillation is required after the setting for LOCO operation. A fixed time is also required for oscillation to stop after the setting to stop the oscillator. Accordingly, take note of the following limitations when starting and stopping the oscillator.

- When restarting the LOCO after it has been stopped, allow at least five cycles of the LOCO as an interval over which it is still stopped.
- Ensure that oscillation by the LOCO is stable when making the setting to stop the LOCO.
- Regardless of whether or not it is selected as the system clock, ensure that oscillation by the LOCO is stable before executing a WAIT instruction to place the chip on software standby.
- When a transition to software standby mode is to follow the setting to stop the LOCO, wait for at least three cycles of the LOCO after the setting to stop the LOCO and before executing the WAIT instruction.

While the LOCO is selected by the SCKCR3.CKSEL[2:0] bits, do not set the LCSTP bit to 1 (LOCO is stopped).

9.2.7 IWDT-Dedicated On-Chip Oscillator Control Register (ILOCOCR)

Address(es): 0008 0035h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ILCSTP
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	ILCSTP	IWDT-Dedicated On-Chip Oscillator Stop	0: IWDT-dedicated on-chip oscillator is operating. 1: IWDT-dedicated on-chip oscillator is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

When the IWDT start mode select bit in option function select register 0 (OFS0.IWDTSTRT) is 0 (IWDT is operating), the setting of this register is invalid; it is valid only when the OFS0.IWDTSTRT bit is set to 1 (IWDT is stopped). The ILCSTP bit cannot be changed from 0 (IWDT-dedicated on-chip oscillator is operating) to 1 (IWDT-dedicated on-chip oscillator is stopped) while ILOCOCR is valid.

ILCSTP Bit (IWDT-Dedicated On-Chip Oscillator Stop)

This bit runs or stops the IWDT-dedicated on-chip oscillator.

After the setting of the ILCSTP bit has been changed so that the IWDT-dedicated on-chip oscillator operates, supply of the clock is started the MCU internally after a fixed time corresponding to the IWDT-dedicated clock oscillation stabilization time (t_{ILOCO}) has elapsed.

If the IWDT-dedicated clock is to be used, only start using the oscillator after this wait time has elapsed.

Ensure that oscillation by the IWDT-dedicated on-chip oscillator is stable before executing a WAIT instruction to place the chip on software standby mode.

9.2.8 High-Speed On-Chip Oscillator Control Register (HOCOCR)

Address(es): 0008 0036h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	HCSTP
Value after reset:	0	0	0	0	0	0	0	0/1*1

Bit	Symbol	Bit Name	Description	R/W
b0	HCSTP	HOCO Stop	0: HOCO is operating. 1: HOCO is stopped.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. The HCSTP bit value after a reset is 0 when the HOCO oscillation enable bit in option function select register 1 (OFS1.HOCOEN) is 0. The HCSTP bit value after a reset is 1 when the OFS1.HOCOEN bit is 1.

Set the high-speed on-chip wait control register before setting this register.

HCSTP Bit (HOCO Stop)

This bit runs or stops the HOCO.
When changing the HCSTP bit from 1 to 0 (i.e. changing the HOCO clock from stopped to operating), confirm that the OSCOVFSR.HCOVF flag is 1 before switching the system clock to the HOCO clock.
That is, a fixed time for stabilization of oscillation is required after the setting for HOCO operation. A fixed time is also required for oscillation to stop after the setting to stop the oscillator. Accordingly, take note of the following limitations when starting and stopping the oscillator.

- After stopping the HOCO, confirm that the OSCOVFSR.HCOVF flag is 0 before restarting the HOCO.
- Confirm that the HOCO is operating and that the OSCOVFSR.HCOVF flag is 1 before stopping the HOCO.
- Regardless of whether or not it is selected as the system clock, confirm that the OSCOVFSR.HCOVF flag is 1 before executing a WAIT instruction to place the MCU in software standby mode.
- After stopping the HOCO, confirm that the OSCOVFSR.HCOVF flag is 0 and execute a WAIT instruction before entering software standby mode.

While the HOCO is selected by the SCKCR3.CKSEL[2:0] bits, do not set the HCSTP bit to 1 (HOCO is stopped).

9.2.9 Oscillation Stabilization Flag Register (OSCOVFSR)

Address(es): 0008 003Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	HCOVF	PLOVF	—	MOOVF
Value after reset:	0	0	0	0	0/1*1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MOOVF	Main Clock Oscillation Stabilization Flag	0: Main clock is stopped 1: Oscillation is stable and the clock can be used as the system clock*2	R
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	PLOVF	PLL Clock Oscillation Stabilization Flag	0: PLL is stopped or not stabilized 1: Oscillation is stable and the clock can be used as the system clock	R
b3	HCOVF	HOCO Clock Oscillation Stabilization Flag	0: HOCO is stopped or not stabilized 1: Oscillation is stable and the clock can be used as the system clock*2	R
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The HCOVF value after a reset is 1 when the HOCO oscillation enable bit in option function selection register 1 (OFS1.HOCOEN) is 0. The HCOVF value after a reset is 0 when the OFS1.HOCOEN bit is 1.
Note 2. When an appropriate value is set in the wait control register for each oscillator. If a set value (wait time) is not adequate, clock supply starts before oscillation becomes stable.

The OSCOVFSR register monitors whether oscillation of each oscillator has become stable.
If a wait control register is provided for each oscillator, specify a wait time that is longer than or equal to the stabilization time of the corresponding oscillation circuit.

MOOVF Flag (Main Clock Oscillation Stabilization Flag)

This flag indicates whether oscillation of the main clock is stable.
[Setting condition]

- After the MOSCCR.MOSTP bit is set to 0 (main clock oscillator is operating) when the MOSTP bit is 1 (main clock oscillator is stopped), the corresponding time set in the MOSCWTCR register has elapsed and supply of the main clock is started to the MCU internally.

[Clearing condition]

- After the MOSCCR.MOSTP bit is set to 1, the processing to stop the oscillation of the main clock oscillator is completed.

PLOVF Flag (PLL Clock Oscillation Stabilization Flag)

This flag indicates whether oscillation of the PLL clock is stable.
[Setting condition]
After the PLLCR2.PLLEN is set to 0 (PLL is operating) when the PLLEN bit is 1 (PLL is stopped), the MOOVF flag becomes 1, the PLL clock oscillation stabilization time (t_{PLL}) has elapsed, and supply of the PLL clock is started to the MCU internally.
[Clearing condition]
After the PLLCR2.PLLEN bit is set to 1, the processing to stop the oscillation of the PLL is completed.

HCOVF Flag (HOCO Clock Oscillation Stabilization Flag)

This flag indicates whether oscillation of the HOCO clock is stable.
[Setting condition]

- After the HOCOCR.HCSTP bit is set to 0 (HOCO is operating) when the HCSTP bit is 1 (HOCO is stopped),

supply of the HOCO clock is started to the MCU internally.

[Clearing condition]

- After the HOCOCR.HCSTP bit is set to 1, the processing to stop the oscillation of the HOCO is completed.

9.2.10 Oscillation Stop Detection Control Register (OSTDCR)

Address(es): 0008 0040h

	b7	b6	b5	b4	b3	b2	b1	b0
	OSTDE	—	—	—	—	—	—	OSTDIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OSTDIE	Oscillation Stop Detection Interrupt Enable	0: The oscillation stop detection interrupt is disabled. Oscillation stop detection is not notified to the POE. 1: The oscillation stop detection interrupt is enabled. Oscillation stop detection is notified to the POE.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	OSTDE	Oscillation Stop Detection Function Enable	0: Oscillation stop detection function is disabled. 1: Oscillation stop detection function is enabled.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.

OSTDIE Bit (Oscillation Stop Detection Interrupt Enable)

If the oscillation stop detection flag in the oscillation stop detection status register (OSTDSR.OSTDF) requires clearing, do this after setting the OSTDIE bit to 0. Wait for at least two cycles of PCLKB before again setting the OSTDIE bit to 1. According to the number of cycles for access to read a given I/O register, wait time longer than two cycles of PCLKB may have to be secured.

OSTDE Bit (Oscillation Stop Detection Function Enable)

This bit enables or disables the oscillation stop detection function.

When the OSTDE bit is 1 (oscillation stop detection function enabled), the LOCO stop bit (LOCOCR.LCSTP) is set to 0 and the LOCO operation is started. The LOCO cannot be stopped while the oscillation stop detection function is enabled; writing 1 (LOCO is stopped) to the LOCOCR.LCSTP bit is invalid.

When the oscillation stop detection flag in the oscillation stop detection status register (OSTDSR.OSTDF) is 1 (main clock oscillation stop has been detected), writing 0 to the OSTDE bit is invalid.

When the OSTDE bit is 1, a transition cannot be made to software standby mode. To make a transition to software standby mode, execute the WAIT instruction with the OSTDE bit being 0.

9.2.11 Oscillation Stop Detection Status Register (OSTDSR)

Address(es): 0008 0041h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	OSTDF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OSTDF	Oscillation Stop Detection Flag	0: The main clock oscillation stop has not been detected. 1: The main clock oscillation stop has been detected.	R/(W) *1
b7 to b1	—	Reserved	These bits are read as 0 and cannot be modified.	R

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. This bit can only be set to 0.

OSTDF Flag (Oscillation Stop Detection Flag)

This bit is a flag to indicate the main clock status. When the OSTDF flag is 1, it indicates that the main clock oscillation stop has been detected.

Once the main clock oscillation stop is detected, the OSTDF flag is not set to 0 even though the main clock oscillation is restarted. The OSTDF flag is set to 0 by reading 1 from the bit and then writing 0. At least three ICLK cycles of wait time is necessary between writing 0 to the OSTDF flag and reading the OSTDF flag as 0. If the OSTDF flag is set to 0 while the main clock oscillation is stopped, the OSTDF flag becomes 0 and then returns to 1.

When the main clock oscillator (010b) or PLL (100b) is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]), the OSTDF flag cannot be modified to 0. The OSTDF flag should be set to 0 after switching the clock source to a source other than the main clock oscillator and the PLL.

[Setting condition]

- The main clock oscillation is stopped with the OSTDCR.OSTDE bit being 1 (oscillation stop detection function enabled).

[Clearing condition]

- 1 is read and then 0 is written when the SCKCR3.CKSEL[2:0] bits are neither 010b nor 100b.

9.2.12 Main Clock Oscillator Wait Control Register (MOSCWTCR)

Address(es): 0008 00A2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	MSTS[4:0]				
Value after reset:	0	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	MSTS[4:0]	Main Clock Oscillator Wait Time	b4 b0 0 0 0 0 0: Wait time = 2 cycles (0.5 µs) 0 0 0 0 1: Wait time = 1024 cycles (256 µs) 0 0 0 1 0: Wait time = 2048 cycles (512 µs) 0 0 0 1 1: Wait time = 4096 cycles (1.024 ms) 0 0 1 0 0: Wait time = 8192 cycles (2.048 ms) 0 0 1 0 1: Wait time = 16384 cycles (4.096 ms) 0 0 1 1 0: Wait time = 32768 cycles (8.192 ms) 0 0 1 1 1: Wait time = 65536 cycles (16.384 ms) Settings other than above are prohibited. Wait time when LOCO = 4.0 MHz (0.25 µs, TYP.)	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

MSTS[4:0] Bits (Main Clock Oscillator Wait Time)

Set these bits to select the oscillation stabilization wait time of the main clock oscillator.

Set the main clock oscillation stabilization time to longer than or equal to the stabilization time recommended by the oscillator manufacturer. When the main clock is externally input, set these bits to 00000b because the oscillation stabilization time is not required.

The wait time set by the MSTS[4:0] bits is counted using the LOCO clock. The LOCO automatically oscillates when necessary, regardless of the value of the LOCOCR.LCSTP bit.

After the set wait time has elapsed, supply of the main clock is started to the MCU internally and the OSCOVFSR.MOOVF flag becomes 1. If the set wait time is short, supply of the main clock is started before oscillation of the clock becomes stable.

Only rewrite the MOSCWTCR register when the MOSCCR.MOSTP bit is 1 and the OSCOVFSR.MOOVF flag is 0. Do not rewrite this register under any other conditions.

9.2.13 Main Clock Oscillator Forced Oscillation Control Register (MOFCR)

Address(es): 0008 C293h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MOSEL	MODRV21	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	MODRV21	Main Clock Oscillator Drive Capability Switch	0: 1 MHz or higher and lower than 10 MHz 1: 10 MHz to 20 MHz	R/W
b6	MOSEL	Main Clock Oscillator Switch	0: Resonator 1: External oscillator input	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

MODRV21 Bit (Main Clock Oscillator Drive Capability Switch)

These bits select the drive capability of the main clock oscillator.

MOSEL Bit (Main Clock Oscillator Switch)

This bit selects the oscillation source of the main clock oscillator.

9.2.14 Low-Speed On-Chip Oscillator Trimming Register (LOCOTRR)

Address(es): 0008 0060h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	LOCOTRD[4:0]				
Value after reset:	0	0	0	*1	*1	*1	*1	*1

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	LOCOTRD[4:0]	Low-Speed On-Chip Oscillator Frequency Adjustment	b4 b0 1 0 0 0 0: -16 (Frequency: Low) 1 0 0 0 1: -15 : : 0 1 1 1 0: 14 0 1 1 1 1: 15 (Frequency: High)	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. Chip-specific value

LOCOTRD[4:0] Bits (Low-Speed On-Chip Oscillator Frequency Adjustment)

Set the frequency adjustment value for the low-speed on-chip oscillator.

The setting range is -16 (10h) to 15 (0Fh) by two's complements. The greater the set value is, the higher the frequency is. The frequency is adjusted under certain conditions before shipment, so the value after reset varies with the chip. After a reset, the oscillation frequency returns to the factory default.

9.2.15 IWDT-Dedicated On-Chip Oscillator Trimming Register (ILOCOTRR)

Address(es): 0008 0064h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ILOCOTRD[4:0]				
Value after reset:	0	0	0	*1	*1	*1	*1	*1

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	ILOCOTRD [4:0]	IWDT-Dedicated On-Chip Oscillator Frequency Adjustment	b4 b0 0 0 0 0 0: 0 (Frequency: Low) 0 0 0 0 1: 1 : : 1 1 1 1 0: 30 1 1 1 1 1: 31 (Frequency: High)	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. Chip-specific value

ILOCOTRD[4:0] Bits (IWDT-Dedicated On-Chip Oscillator Frequency Adjustment)

Set the frequency adjustment value for the IWDT-dedicated on-chip oscillator.

The setting range is from 0 (00h) to 31 (1Fh) by binary numbers. The greater the set value is, the higher the frequency is. The frequency is adjusted under certain conditions before shipment, so the value after reset varies with the chip. After a reset, the oscillation frequency returns to the factory default.

9.2.16 High-Speed On-Chip Oscillator Trimming Register n (HOCOTRRn) (n = 0)

Address(es): HOCOTRR0 0008 0068h

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	HOCOTRD[5:0]	High-Speed On-Chip Oscillator Frequency Adjustment	b5 b0 0 0 0 0 0 0: 0 (Frequency: Low) 0 0 0 0 0 1: 1 : : 1 1 1 1 1 0: 62 1 1 1 1 1 1: 63 (Frequency: High)	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC0 bit to 1 (write enabled) before rewriting this register.
Note 1. Chip-specific value

HOCOTRR0 corresponds to 32 MHz.

HOCOTRD[5:0] Bits (High-Speed On-Chip Oscillator Frequency Adjustment)

Set the frequency adjustment value for the high-speed on-chip oscillator.

The setting range is from 0 (00h) to 63 (3Fh) by binary numbers. The greater the set value is, the higher the frequency is. The frequency is adjusted under certain conditions before shipment, so the value after reset varies with the chip. After a reset, the oscillation frequency returns to the factory default.

9.3 Main Clock Oscillator

There are two ways of supplying the clock signal from the main clock oscillator: connecting an oscillator or the input of an external clock signal.

9.3.1 Connecting a Crystal

Figure 9.2 shows an example of connecting a crystal.
A damping resistor (R_d) should be added, if necessary. Since the resistor values vary depending on the resonator and the oscillation drive capability, use values recommended by the resonator manufacturer. If use of an external feedback resistor (R_f) is directed by the resonator manufacturer, insert an R_f between EXTAL and XTAL by following the instruction.
When connecting a resonator to supply the clock, the frequency of the resonator should be in the frequency range of the resonator for the main clock oscillator described in Table 9.1.

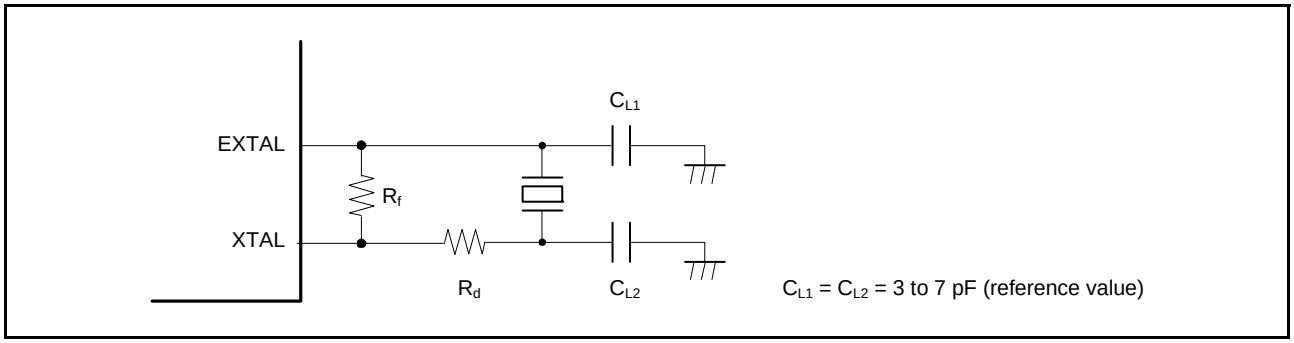


Figure 9.2 Example of Crystal Connection

Table 9.3 Damping Resistance (Reference Values)

Frequency (MHz)	2	8	16	20
R_d (Ω)	0	0	0	0

Figure 9.3 shows an equivalent circuit of the crystal. Use a crystal that has the characteristics shown in Table 9.4 as a reference.

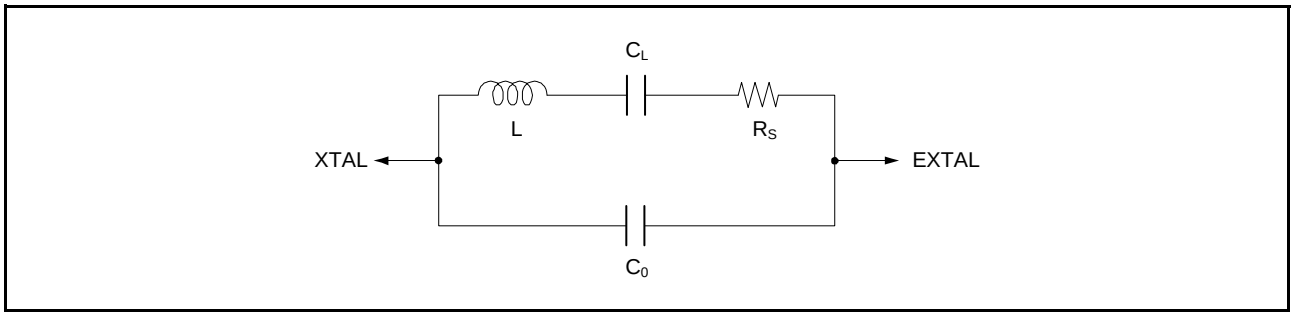


Figure 9.3 Equivalent Circuit of Crystal

Table 9.4 Crystal Characteristics (Reference Values)

Frequency (MHz)	8	12	16
R_S max (Ω)	200	120	56
C_0 max (pF)	1.3	1.3	1.4

9.3.2 External Clock Input

Figure 9.4 shows connection of an external clock. Set the MOFCR.MOSEL bit to 1 and open the XTAL pin to operate the oscillator by inputting an external clock signal.

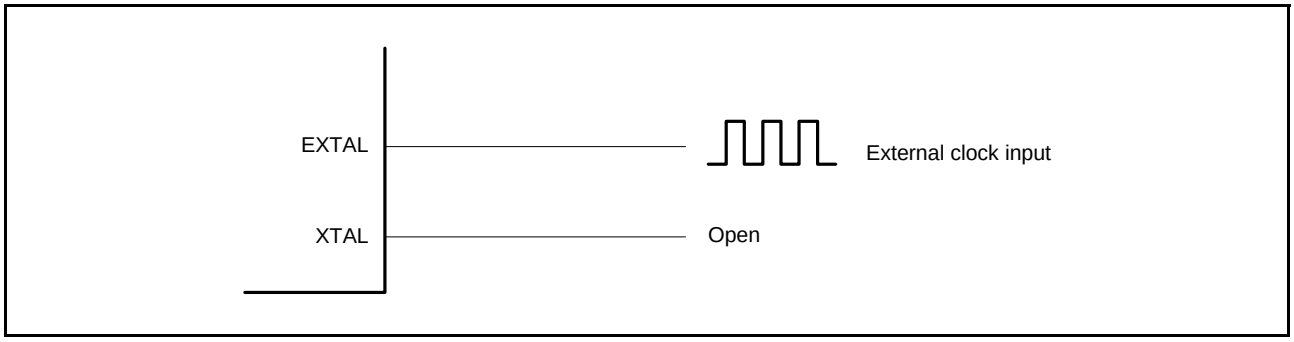


Figure 9.4 Connection Example of External Clock

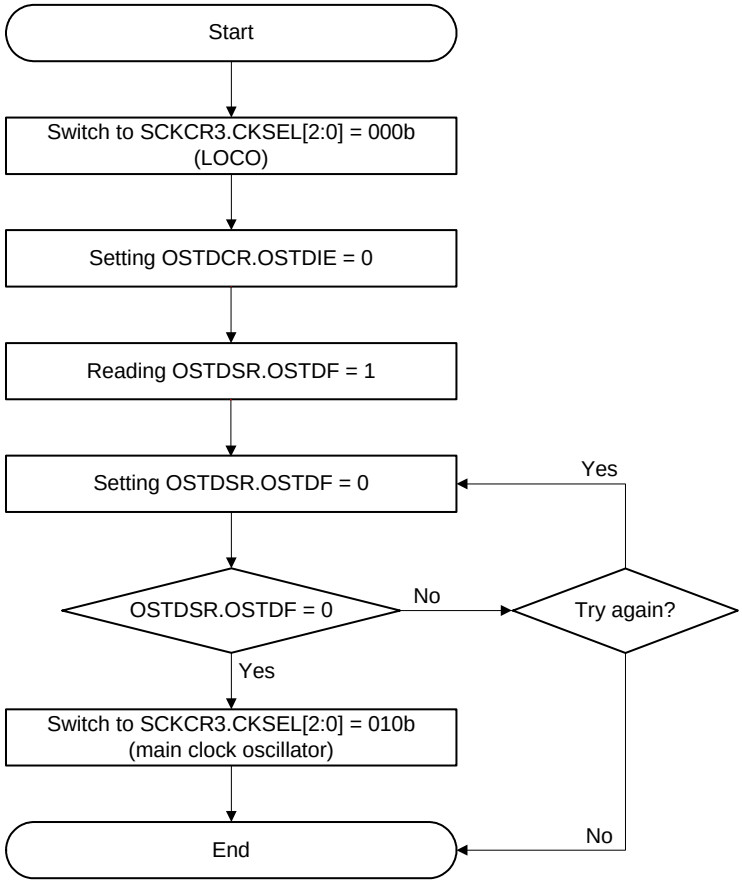
9.3.3 Notes on the External Clock Input

The frequency of the external clock input can only be changed while the main clock oscillator is stopped. Do not change the frequency of the external clock input while the setting of the main clock oscillator stop bit (MOSCCR.MOSTP) is 0 (main clock oscillator is operating).

9.4 Oscillation Stop Detection Function

9.4.1 Oscillation Stop Detection and Operation after Detection

The oscillation stop detection function is used to detect the main clock oscillator stop and to supply LOCO clock pulses from the low-speed on-chip oscillator as the system clock source instead of the main clock.
An oscillation stop detection interrupt request can be generated when an oscillation stop is detected. In addition, the MTU output can be forcedly driven to the high-impedance on the detection. For details, refer to section 19, Multi-Function Timer Pulse Unit 3 (MTU3c) section 20, Port Output Enable 3 (POE3C).
In the MCU, the main clock oscillation stop is detected when the input clock remains to be 0 or 1 for a certain period, for example, due to a malfunction of the main clock oscillator (refer to section 32, Electrical Characteristics).
When an oscillation stop is detected, the main clock selected by the clock source select bits (SCKCR3.CKSEL[2:0]) is switched to the LOCO clock by the corresponding selectors in the former stage. Therefore, if an oscillation stop is detected with the main clock selected as the system clock source, the system clock source is switched to the LOCO clock without a change of CKSEL[2:0].
If an oscillation stop is detected while the PLL clock is selected by the clock source select bits (SCKCR3.CKSEL[2:0]) in system clock control register 3, the SCKCR3.CKSEL[2:0] bit value does not change and the PLL clock remains the system clock source. However, the frequency becomes a free-running oscillation frequency.
Switching between the main clock and LOCO clock is controlled by the oscillation stop detection flag (OSTDSR.OSTDF). The clock source is switched to the LOCO clock when the OSTDF flag is 1, and is switched to the main clock again when the OSTDF flag is set to 0. At this time, if the main clock or PLL clock is selected with the CKSEL[2:0] bits, the OSTDF flag cannot be set to 0. To switch the clock source to the main clock or PLL clock again after the oscillation stop detection, set the CKSEL[2:0] bits to a clock source other than the main clock or PLL clock and set the OSTDF flag to 0. After that, check that the OSTDF flag is not 1, and then set the CKSEL[2:0] bits to the main clock or PLL clock after the specified oscillation stabilization time has elapsed.
After a reset is released, the main clock oscillator is stopped and the oscillation stop detection function is disabled. To enable the oscillation stop detection function, activate the main clock oscillator and write 1 to the oscillation stop detection function enable bit (OSTDCR.OSTDE) after a specified oscillation stabilization time has elapsed.
The oscillation stop detection function is provided against the main clock stop by an external cause. Therefore, the oscillation stop detection function should be disabled before the main clock oscillator is stopped by the software or a transition is made to software standby mode.
When the system clock with the main clock selected as the system clock source and the CAC main clock (CACMCLK) are selected, these clocks are switched to the LOCO clock by the detection of oscillation stop. The system clock (ICLK) frequency during the LOCO clock operation is specified by the LOCO oscillation frequency and the division ratio set by the system clock (ICLK) select bits (SCKCR.ICK[3:0]).
When the PLL clock is selected as the system clock source, these clocks operate at the PLL free-running frequency by the oscillation stop detection.



Note: On return from the oscillation-stopped state, the factor responsible for stopping the main clock oscillation circuit must be removed on the user system to allow the return of oscillation.

Figure 9.5 Flow of Recovery from Detection of Oscillator Stop

9.4.2 Oscillation Stop Detection Interrupts

An oscillator-stop detection interrupt (OSTDI) request is generated if the oscillation-stop detection flag (OSTDSR.OSTDF) becomes 1 while the oscillation-stop detection interrupt enable bit (OSTDCR.OSTDIE) is 1 (oscillation stop detection interrupt enabled). At this time, the stop of the main clock oscillator is notified to the port output enable 3 (POE). On accepting the notification of the oscillation stop, the POE sets the OSTST high-impedance flag in input level control/status register 6 (ICSR6.OSTSTF) to 1. After the oscillation stop is detected, wait for at least 10 cycles of PCLKB before writing to this ICSR6.OSTSTF flag. When the OSTDSR.OSTDF flag requires clearing, do so setting the oscillation stop detection interrupt enable bit (OSTDCR.OSTDIE) to 0. Wait for at least two cycles of PCLKB clock before again setting the OSTDCR.OSTDIE bit to 1. According to the number of cycles for access to read a given I/O register, wait time longer than two cycles of PCLKB may have to be secured.

The oscillation stop detection interrupt is a non-maskable interrupt. Since non-maskable interrupts are disabled in the initial state after a reset release, enable the non-maskable interrupts by the software before using oscillation stop detection interrupts. For details, refer to section 14, Interrupt Controller (ICUb).

When the PLL detects an oscillation stop and is running at its own oscillation frequency, this indicates the occurrence of some system failure. An emergency measure should be taken to handle the failure.

9.5 PLL Circuit

The PLL circuit has a function to multiply the frequency from the oscillator.

9.6 Internal Clock

Clock sources of internal clock signals are the main clock, HOCO clock, LOCO clock, PLL clock, and dedicated low-speed clock for the IWDT. The internal clocks listed below are produced from these sources.

(1) Operating clock of the CPU, DTC, ROM, and RAM: System clock (ICLK)
(2) Operating clock of peripheral modules: Peripheral module clock (PCLKB, and PCLKD)
(3) Operating clock of the FlashIF: FlashIF clock (FCLK)
(4) Operating clock for the CAC: CAC clock (CACCLK)
(5) Operating clock for the IWDT: IWDT-dedicated low-speed clock (IWDTCLK)

Frequencies of the internal clocks are set by the combinations of the division ratios selected by the FCK[3:0], ICK[3:0], PCKB[3:0], and PCKD[3:0] bits in the SCKCR register, the clock source selected by the SCKCR3.CKSEL[2:0] bits, the multiplier and divisor for the frequency of the PLL circuit set by the STC[5:0] and PLIDIV[1:0] bits in the PLLCR register. If the value of any of these bits is changed, subsequent operation will be at the frequency determined by the new value.

9.6.1 System Clock

The system clock (ICLK) is used as the operating clock of the CPU, DTC, ROM, and RAM.
The ICLK frequency is set by using the SCKCR.ICK[3:0] bits, the SCKCR3.CKSEL[2:0] bits, the STC[5:0] and PLIDIV[1:0] bits in the PLLCR register.

9.6.2 Peripheral Module Clock

The peripheral module clocks (PCLKB, and PCLKD) are the operating clocks for use by peripheral modules.
The frequencies of the PCLKB, and PCLKD are set by using the PCKB[3:0] and PCKD[3:0] bits in the SCKCR register, the SCKCR3.CKSEL[2:0] bits, the STC[5:0] and PLIDIV[1:0] bits in the PLLCR register.
The peripheral module clock used as the operating clock is PCLKD for S12AD, and PCLKB is for other modules.

9.6.3 FlashIF Clock

The FlashIF clock (FCLK) is used as the operating clock of the FlashIF.
The FCLK frequency is set by using the SCKCR.FCK[3:0] bits, the SCKCR3.CKSEL[2:0] bits, and the STC[5:0] and PLIDIV[1:0] bits in the PLLCR register.

9.6.4 CAC Clock

The CAC clock (CACCLK) is an operating clock for the CAC module.
The CACCLK clocks include CACMCLK which is generated by the main clock oscillator, CACHCLK which is generated by the high-speed on-chip oscillator, CACLCLK which is generated by the low-speed on-chip oscillator, and CACILCLK which is generated by the IWDT-dedicated on-chip oscillator.

9.6.5 IWDT-Dedicated Clock

The IWDT-dedicated clock (IWDTCLK) is the operating clock for the IWDT.

IWDTCLK is internally generated by the IWDT-dedicated on-chip oscillator.

9.7 Usage Notes

9.7.1 Notes on Clock Generation Circuit

(1) The frequencies of the system clock (ICLK), peripheral module clocks (PCLKB, and PCLKD), and FlashIF clock (FCLK) supplied to each module can be selected by the SCKCR register. Each frequency should meet the following:
Select each frequency that is within the operation guaranteed range of clock cycle time (t_{cyc}) specified in AC characteristics of electrical characteristics.
The frequencies must not exceed the ranges listed in Table 9.1.
The peripheral modules operate on the PCLKB and PCLKD. Note therefore that the operating speed of modules such as the timer and SCI varies before and after the frequency is changed.

(2) The relationship of frequencies of the system clock (ICLK), peripheral module clocks B and D (PCLKB, and PCLKD), and FlashIF clock (FCLK) must be set as follows.
ICLK:FCLK = N:1 (N is an integer)
ICLK:PCLKB, and PCLKD = N:1 (N is an integer)

(3) To secure the processing after the clock frequency is changed, modify the pertinent clock control register to change the frequency, and then read the value from the register, and then perform the subsequent processing.

9.7.2 Note on Rewriting the SCKCR3 Register

When the SCKCR3.CKSEL[2:0] bits are rewritten, the clock output is temporarily stopped to prevent the generation of a clock pulse of short duration (glitch) when the clock source is switched. The interrupt controller may not detect the following signals that were input during this period.

(1) An external pin interrupt or NMI pin interrupt with shorter pulse width than 4 cycles of the PCLKB after the switch while the PCLKB frequency is the 1/1 of the clock source (the SCKCR.PCKB[3:0] bits are 0000b).

(2) An external pin interrupt or NMI pin interrupt with shorter pulse width than 2.5 cycles of the PCLKB after the switch while the PCLKB frequency is the 1/2 of the clock source (the SCKCR.PCKB[3:0] bits are 0001b).

When the external pin interrupt or NMI pin interrupt is in use, input the signal with enough pulse width to exceed the time condition described in (1) and (2).

9.7.3 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.

9.7.4 Notes on Board Design

When using a crystal, 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 9.6 to prevent electromagnetic induction from interfering with correct oscillation.

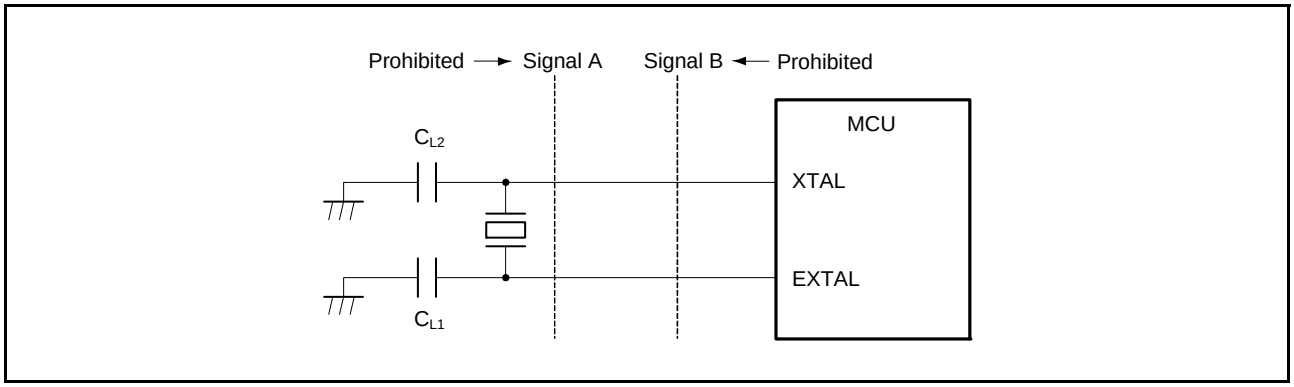


Figure 9.6 Notes on Board Design for Oscillation Circuit

9.7.5 Notes on Resonator Connection Pins

When the main clock is not used, the EXTAL and XTAL pins can be used as general ports P36 and P37. When using these pins as general ports, be sure to stop the main clock (MOSCCR.MOSTP = 1). However, do not use the EXTAL and XTAL pins as general ports P36 and P37 in a system that uses the main clock.
When the main clock is used, do not set P36 and P37 to output.

10. Clock Frequency Accuracy Measurement Circuit (CAC)

10.1 Overview

The clock frequency accuracy measurement circuit (CAC) counts pulses of the clock to be measured (measurement target clock) within the time generated by the clock to be used as a measurement reference (measurement reference clock), and determines the accuracy depending on whether the number of pulses is within the allowable range.
When measurement is completed or the number of pulses within the time generated by the measurement reference clock is not within the allowable range, an interrupt request is generated.
Table 10.1 lists the specifications of the CAC and Figure 10.1 shows a block diagram of the CAC.

Table 10.1 CAC Specifications

Item	Description
Measurement target clocks	The frequency of the following clocks can be measured. • Main clock • HOCO clock • LOCO clock • IWDT-dedicated clock (IWDTCLK) • Peripheral module clock B (PCLKB)
Measurement reference clocks	• External clock input to the CACREF pin • Main clock • HOCO clock • LOCO clock • IWDT-dedicated clock (IWDTCLK) • Peripheral module clock B (PCLKB)
Selectable function	Digital filter function
Interrupt sources	• Measurement end interrupt • Frequency error interrupt • Overflow interrupt
Low power consumption function	Module stop state can be set.

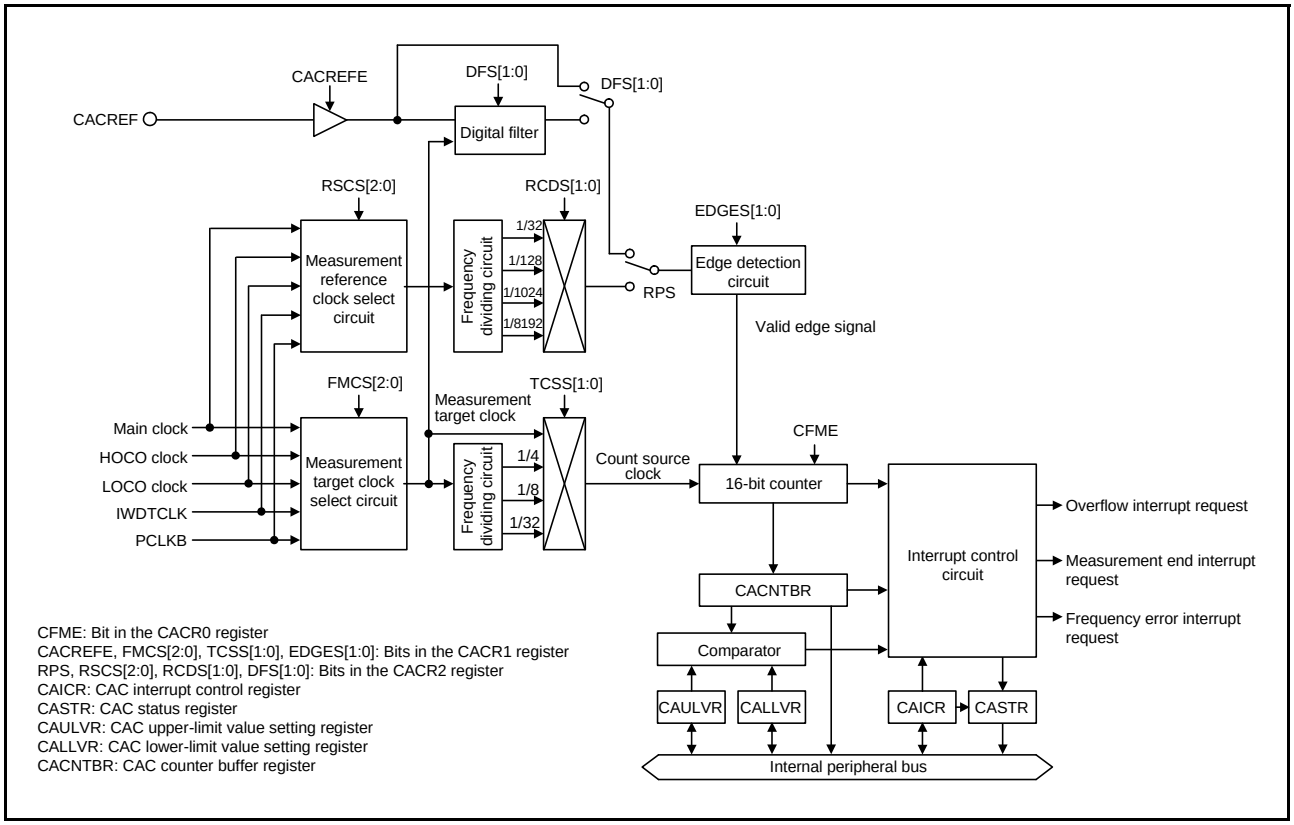


Figure 10.1 CAC Block Diagram

Table 10.2 shows the pin configuration of the CAC.

Table 10.2 Pin Configuration of CAC

Pin Name	I/O	Function
CACREF	Input	Measurement reference clock input pin

10.2 Register Descriptions

10.2.1 CAC Control Register 0 (CACR0)

Address(es): CAC.CACR0 0008 B000h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CFME
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CFME	Clock Frequency Measurement Enable	0: Clock frequency measurement is disabled. 1: Clock frequency measurement is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

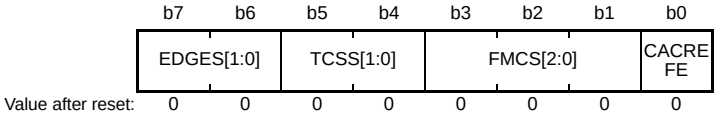
CFME Bit (Clock Frequency Measurement Enable)

This bit specifies whether clock frequency measurement is enabled or disabled.

When rewriting this bit, more time is required than other bits for the new value to be reflected in the register. Further write access to this bit are ignored until the current write access is reflected in the register. Read the bit to confirm that the rewrite has been reflected in the register.

10.2.2 CAC Control Register 1 (CACR1)

Address(es): CAC.CACR1 0008 B001h



Bit	Symbol	Bit Name	Description	R/W
b0	CACREFE	CACREF Pin Input Enable	0: CACREF pin input is disabled. 1: CACREF pin input is enabled.	R/W
b3 to b1	FMCS[2:0]	Measurement Target Clock Select	b3 b1 0 0 0: Main clock 0 1 0: HOCO clock 0 1 1: LOCO clock 1 0 0: IWDT-dedicated clock (IWDTCLK) 1 0 1: Peripheral module clock B (PCLKB) Settings other than above are prohibited.	R/W
b5, b4	TCSS[1:0]	Timer Count Clock Source Select	b5 b4 0 0: No division 0 1: ×1/4 clock 1 0: ×1/8 clock 1 1: ×1/32 clock	R/W
b7, b6	EDGES[1:0]	Valid Edge Select	b7 b6 0 0: Rising edge 0 1: Falling edge 1 0: Both rising and falling edges 1 1: Setting prohibited	R/W

Note: Set the CACR1 register when the CACR0.CFME bit is 0.

CACREFE Bit (CACREF Pin Input Enable)

This bit specifies whether the CACREF pin input is enabled or disabled.

FMCS[2:0]Bits (Measurement Target Clock Select)

These bits select the measurement target clock whose frequency is to be measured.

TCSS[1:0] Bits (Timer Count Clock Source Select)

These bits select the count clock source for the clock frequency accuracy measurement circuit.

EDGES[1:0]Bits (Valid Edge Select)

These bits select the valid edge for the reference signal.

10.2.3 CAC Control Register 2 (CACR2)

Address(es): CAC.CACR2 0008 B002h

	b7	b6	b5	b4	b3	b2	b1	b0
	DFS[1:0]		RCDS[1:0]		RSCS[2:0]			RPS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	RPS	Reference Signal Select	0: CACREF pin input 1: Internal clock (internally generated signal)	R/W
b3 to b1	RSCS[2:0]	Measurement Reference Clock Select	b3 b1 0 0 0: Main clock 0 1 0: HOCO clock 0 1 1: LOCO clock 1 0 0: IWDT-dedicated clock (IWDTCLK) 1 0 1: Peripheral module clock B (PCLKB) Settings other than above are prohibited.	R/W
b5, b4	RCDS[1:0]	Measurement Reference Clock Frequency Division Ration Select	b5 b4 0 0: ×1/32 clock 0 1: ×1/128 clock 1 0: ×1/1024 clock 1 1: ×1/8192 clock	R/W
b7, b6	DFS[1:0]	Digital Filter Select	b7 b6 0 0: Digital filtering is disabled. 0 1: The sampling clock for the digital filter is the measurement target clock. 1 0: The sampling clock for the digital filter is the measurement target clock divided by 4. 1 1: The sampling clock for the digital filter is the measurement target clock divided by 16.	R/W

Note: Set the CACR2 register when the CACR0.CFME bit is 0.

RPS Bit (Reference Signal Select)

This bit selects whether to use the CACREF pin input or an internal clock (internally generated signal) as the reference signal.

RSCS[2:0]Bits (Measurement Reference Clock Select)

These bits select the clock source for generating the measurement reference clock.

RCDS[1:0]Bits (Measurement Reference Clock Frequency Division Ration Select)

These bits select the frequency division ratio of the measurement reference clock.

DFS[1:0]Bits (Digital Filter Select)

The setting of these bits enables or disables the digital filter and selects its sampling clock.

10.2.4 CAC Interrupt Request Enable Register (CAICR)

Address(es): CAC.CAICR 0008 B003h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	OVFFCL	MENDFCL	FERRFCL	—	OVFIE	MENDIE	FERRIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FERRIE	Frequency Error Interrupt Request Enable	0: Frequency error interrupt request is disabled. 1: Frequency error interrupt request is enabled.	R/W
b1	MENDIE	Measurement End Interrupt Request Enable	0: Measurement end interrupt request is disabled. 1: Measurement end interrupt request is enabled.	R/W
b2	OVFIE	Overflow Interrupt Request Enable	0: Overflow interrupt request is disabled. 1: Overflow interrupt request is enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	FERRFCL	FERRF Clear	When 1 is written to this bit, the CASTR.FERRF flag is cleared. This bit is read as 0.	R/W
b5	MENDFCL	MENDF Clear	When 1 is written to this bit, the CASTR.MENDF flag is cleared. This bit is read as 0.	R/W
b6	OVFFCL	OVFF Clear	When 1 is written to this bit, the CASTR.OVFF flag is cleared. This bit is read as 0.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

FERRIE Bit (Frequency Error Interrupt Request Enable)

This bit specifies whether the frequency error interrupt request is enabled or disabled.

MENDIE Bit (Measurement End Interrupt Request Enable)

This bit specifies whether the measurement end interrupt request is enabled or disabled.

OVFIE Bit (Overflow Interrupt Request Enable)

This bit specifies whether the overflow interrupt request is enabled or disabled.

FERRFCL Bit (FERRF Clear)

Setting this bit to 1 clears the CASTR.FERRF flag.

MENDFCL Bit (MENDF Clear)

Setting this bit to 1 clears the CASTR.MENDF flag.

OVFFCL Bit (OVFF Clear)

Setting this bit to 1 clears the CASTR.OVFF flag.

10.2.5 CAC Status Register (CASTR)

Address(es): CAC.CASTR 0008 B004h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	OVFF	MENDF	FERRF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FERRF	Frequency Error Flag	0: The clock frequency is within the range corresponding to the settings. 1: The clock frequency has deviated beyond the range corresponding to the settings (frequency error).	R
b1	MENDF	Measurement End Flag	0: Measurement is in progress. 1: Measurement has ended.	R
b2	OVFF	Overflow Flag	0: The counter has not overflowed. 1: The counter has overflowed.	R
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FERRF Flag (Frequency Error Flag)

This flag indicates deviation of the clock frequency from the set value (frequency error).

[Setting condition]

- The clock frequency is outside of the setting range.

[Clearing condition]

- 1 is written to the CAICR.FERRFCL bit.

MENDF Flag (Measurement End Flag)

This flag indicates the end of measurement.

[Setting condition]

- Measurement has finished.

[Clearing condition]

- 1 is written to the CAICR.MENDFCL bit.

OVFF Flag (Overflow Flag)

This flag indicates that the counter has overflowed.

[Setting condition]

- The counter has overflowed.

[Clearing condition]

- 1 is written to the CAICR.OVFFCL bit.

10.2.6 CAC Upper-Limit Value Setting Register (CAULVR)

Address(es): CAC.CAULVR 0008 B006h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CAULVR register is a 16-bit readable/writable register that specifies the upper-limit value of the counter used for measuring the frequency. When the frequency rises above the value specified in this register, a frequency error is detected.
Write to this register when the CACR0.CFME bit is 0.
The counter value held in the CACNTBR register can vary with the difference between the phases of the digital filter and edge-detection circuit on the one hand and the signal on the CACREF pin on the other, so ensure that this setting allows an adequate margin.

10.2.7 CAC Lower-Limit Value Setting Register (CALLVR)

Address(es): CAC.CALLVR 0008 B008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CALLVR register is a 16-bit readable/writable register that specifies the lower-limit value of the counter used for measuring the frequency. When the frequency falls below the value specified in this register, a frequency error is detected.
Write to this register when the CACR0.CFME bit is 0.
The counter value held in the CACNTBR register can vary with the difference between the phases of the digital filter and edge-detection circuit on the one hand and the signal on the CACREF pin on the other, so ensure that this setting allows an adequate margin.

10.2.8 CAC Counter Buffer Register (CACNTBR)

Address(es): CAC.CACNTBR 0008 B00Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CACNTBR register is a 16-bit read-only register that retains the counter value at the time a valid reference signal edge is input.

10.3 Operation

10.3.1 Measuring Clock Frequency

The clock frequency accuracy measurement circuit measures the clock frequency using the CACREF pin input or the internal clock as a reference. Figure 10.2 shows an operating example of the clock frequency accuracy measurement circuit.

The clock frequency accuracy measurement circuit operates as shown below when measuring the clock frequency.

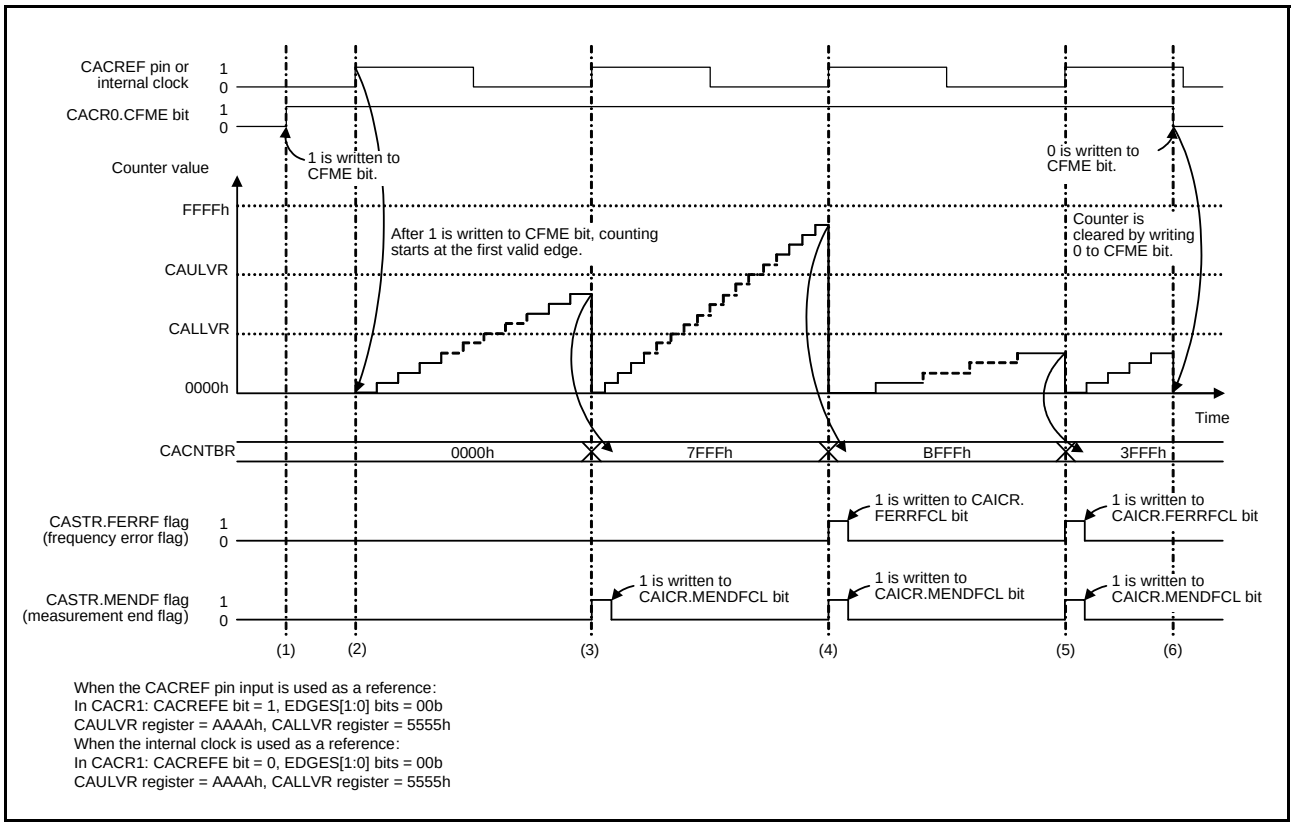


Figure 10.2 Operating Example of Clock Frequency Accuracy Measurement Circuit

(1) When the CACREF pin input is used as a reference (the CACR1.CACREFE bit = 1), clock frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is 0 and the CACR1.CACREFE bit is 1. On the other hand, when the internal clock is used as a reference (the CACR1.CACREFE bit = 0), clock frequency measurement is enabled by writing 1 to the CACR0.CFME bit while the CACR2.RPS bit is 1.

(2) When the CACREF pin input is used as a reference, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits is input from the CACREF pin after 1 is written to the CFME bit. The valid edge is a rising edge (the CACR1.EDGES[1:0] bits = 00b) in Figure 10.2.
When the internal clock is used as a reference, the timer starts up-counting if the valid edge selected by the CACR1.EDGES[1:0] bits is input based on the clock source selected by the CACR2.RSCS[2:0] bits after 1 is written to the CFME bit. The valid edge is a rising edge (the CACR1.EDGES[1:0] bits = 00b) in Figure 10.2.

(3) When the next valid edge is input, the counter value is transferred in the CACNTBR register and compared with the values of the CAULVR and CALLVR registers. If the formula CALLVR ≤ CACNTBR ≤ CAULVR is satisfied, only the CASTR.MENDF flag is set to 1 because the clock frequency is correct. If the CAICR.MENDIE bit is 1, a measurement end interrupt is generated.

(4) When the next valid edge is input, the counter value is transferred in the CACNTBR register and compared with the values of the CAULVR and CALLVR registers. In the case of CACNTBR > CAULVR, the CASTR.FERRF flag is set to 1 because the clock frequency is erroneous. If the CAICR.FERRIE bit is 1, a frequency error interrupt is

generated. Also, the CASTR.MENDF flag is set to 1. If the CAICR.MENDIE bit is 1, a measurement end interrupt is generated.

(5) When the next valid edge is input, the counter value is transferred in the CACNTBR register and compared with the values of the CAULVR and CALLVR registers. In the case of CACNTBR < CALLVR, the CASTR.FERRF flag is set to 1 because the clock frequency is erroneous. If the CAICR.FERRIE bit is 1, a frequency error interrupt is generated. Also, the CASTR.MENDF flag is set to 1. If the CAICR.MENDIE bit is 1, a measurement end interrupt is generated.

(6) While the CACR0.CFME bit is 1, the counter value is transferred in the CACNTBR register and compared with the values of the CAULVR and CALLVR registers every time a valid edge is input. Writing 0 to the CACR0.CFME bit clears the counter and stops up-counting.

10.3.2 Digital Filtering of Signals on the CACREF Pin

The CACREF pin has a digital filter. Levels on the target pin for sampling are conveyed to the internal circuitry after matching three consecutive times at the selected sampling interval and the same level continues to be conveyed internally until the level on the pin again matches three consecutive times.

Enabling and disabling of the digital filter and its sampling clock are selectable.

The counter value transferred in the CACNTBR register may be in error by up to one cycle of the sampling clock due to the difference between the phases of the digital filter and the signal input to the CACREF pin.

When a frequency dividing clock is selected as a count source clock, the counter value error is obtained by the following formula:

Counter value error = (One cycle of the count source clock) / (One cycle of the sampling clock)

10.4 Interrupt Requests

The CAC generates three types of interrupt request: frequency error interrupt, measurement end interrupt, and overflow interrupt. When an interrupt source is generated, the corresponding status flag becomes 1. Table 10.3 lists details on the interrupt requests of the clock frequency accuracy measurement circuit.

Table 10.3 Interrupt Requests of Clock Frequency Accuracy Measurement Circuit

Interrupt Request	Interrupt Enable Bit	Status Flag	Interrupt Source
Frequency error interrupt	CAICR.FERRIE	CASTR.FERRF	The result of comparing CACNTBR to CAULVR and CALLVR is either CACNTBR > CAULVR or CACNTBR < CALLVR.
Measurement end interrupt	CAICR.MENDIE	CASTR.MENDF	A valid edge is input from the CACREF pin. Note however that a measurement end interrupt does not occur at the first valid edge after writing 1 to the CACR0.CFME bit.
Overflow interrupt	CAICR.OVFIE	CASTR.OVFF	The counter has overflowed.

10.5 Usage Notes

10.5.1 Module Stop Function Setting

CAC operation can be disabled or enabled using module stop control register C (MSTPCRC). The initial setting is for the CAC to be halted. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

11. Low Power Consumption

11.1 Overview

This MCU has several functions for reducing power consumption, by setting clock dividers, stopping modules, changing to low power consumption mode in normal operation, and changing to operating power control mode.
Table 11.1 lists the specifications of low power consumption functions, and Table 11.2 lists the conditions to change to low power consumption modes, states of the CPU and peripheral modules, and the method for exiting each mode.
After a reset, this MCU returns to normal mode, but modules except the DTC, and RAM are stopped.

Table 11.1 Specifications of Low Power Consumption Functions

Item	Specification
Clock divider functions	The frequency division ratio can be set independently for the system clock (ICLK), peripheral module clock (PCLKB), S12AD clock (PCLKD), and FlashIF clock (FCLK).*1
Module stop function	Each peripheral module can be stopped independently by the module stop control register.
Function for transition to low power consumption mode	Transition to a low power consumption mode in which the CPU, peripheral modules, or oscillators are stopped is enabled.
Low power consumption modes	• Sleep mode • Deep sleep mode • Software standby mode
Operating power control modes	• Power consumption can be reduced in normal operation, sleep mode, and deep sleep mode by selecting an appropriate operating power control mode according to the operating frequency and operating voltage. • Two operating power control modes are available High-speed operating mode Middle-speed operating mode

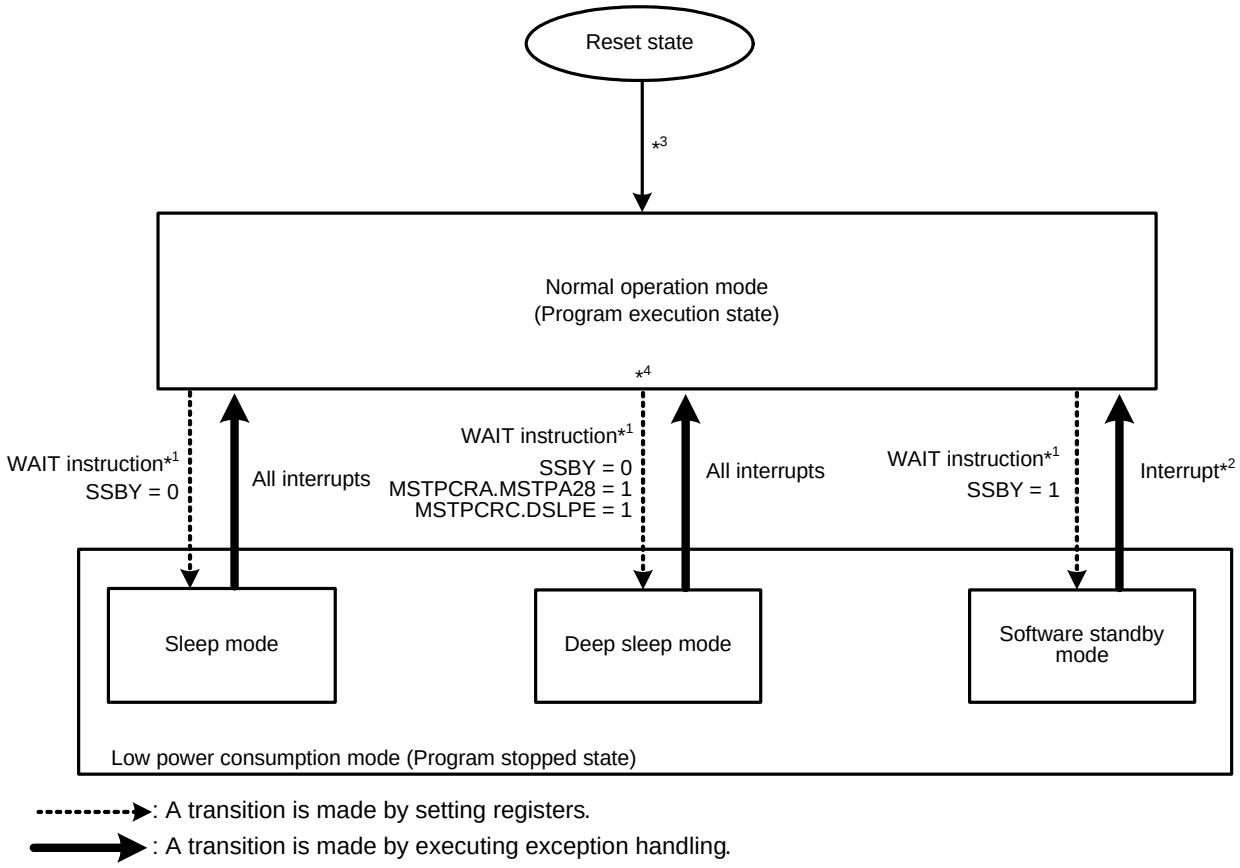
Note 1. For details, refer to section 9, Clock Generation Circuit.

Table 11.2 Operating Conditions of Each Power Consumption Mode

	Sleep Mode	Deep Sleep Mode	Software Standby Mode
Entry trigger	Control register + instruction	Control register + instruction	Control register + instruction
Exit trigger	Interrupt	Interrupt	Interrupt*1
After exiting from each mode, CPU begins from*2	Interrupt handling	Interrupt handling	Interrupt handling
Main clock oscillator	Operating possible	Operating possible	Stopped
High-speed on-chip oscillator	Operating possible	Operating possible	Stopped
Low-speed on-chip oscillator	Operating possible	Operating possible	Stopped
IWDT-dedicated on-chip oscillator	Operating possible*3	Operating possible*3	Operating possible*3
PLL	Operating possible	Operating possible	Stopped
CPU	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)
RAM0 (0000 0000h to 0000 2FFFh)	Operating possible (Retained)	Stopped (Retained)	Stopped (Retained)
DTC	Operating possible*5	Stopped (Retained)	Stopped (Retained)
Flash memory	Operating	Stopped (Retained)	Stopped (Retained)
Independent watchdog timer (IWDT)	Operating possible*3	Operating possible*3	Operating possible*3
Voltage detection circuit (LVD)	Operating possible	Operating possible	Operating possible
Power-on reset circuit	Operating	Operating	Operating
Peripheral modules	Operating possible	Operating possible	Stopped (Retained)*4
I/O ports	Operating	Operating	Retained
Comparator C	Operating possible	Operating possible	Operating possible*6

“Operating possible” means that operating or stopped can be controlled by the register setting.
“Stopped (Retained)” means that internal register values are retained and internal operations are suspended.

Note 1. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ5) or any of peripheral interrupts (the IWDT, and voltage monitoring interrupts).
Note 2. This does not include a RES# pin reset, power-on reset, voltage monitoring reset, or independent watchdog-timer reset. One of these reset sources initiate transition to reset state.
Note 3. Operating or stopping is selected by setting the IWDT sleep mode count stop control bit (IWDTSLCSTP) in option function select register 0 (OFS0) in IWDT auto-start mode. In any mode other than IWDT auto-start mode, operating or stopping is selected by the setting of the sleep mode count stop control bit (SLCSTP) in the IWDT count stop control register (IWDTCSTPR).
Note 4. The peripheral logic states are retained.
Note 5. During sleep mode, do not write to the system control related registers (indicated by ‘SYSTEM’ in the Module Symbol column in Table 5.1, List of I/O Registers (Address Order).
Note 6. Using the digital filter function is prohibited. Operation for outputting the comparison result to the COMPn pin is possible.



Note 1. If an interrupt is accepted while entering low power consumption mode, the transition to the program stop state is exited and the interrupt exception handling is executed.
Note 2. “Interrupts” here indicates an external pin interrupt (the NMI or IRQ0 to IRQ5) or any of peripheral interrupts (the IWDT and voltage monitoring interrupts).
Note 3. The LOCO is the clock source following a transition from the reset state to normal mode.
Note 4. Makes a transition from sleep mode, deep sleep mode, or software standby mode to normal operating mode by an interrupt.
For deep sleep mode and software standby mode, the clock source after exiting is the same as that of the clock before entering software standby mode.

Transition to the reset state is made at any state instantly after a RES# pin reset, power on reset, voltage monitoring reset, IWDT reset, or software reset is generated.

Figure 11.1 Mode Transitions

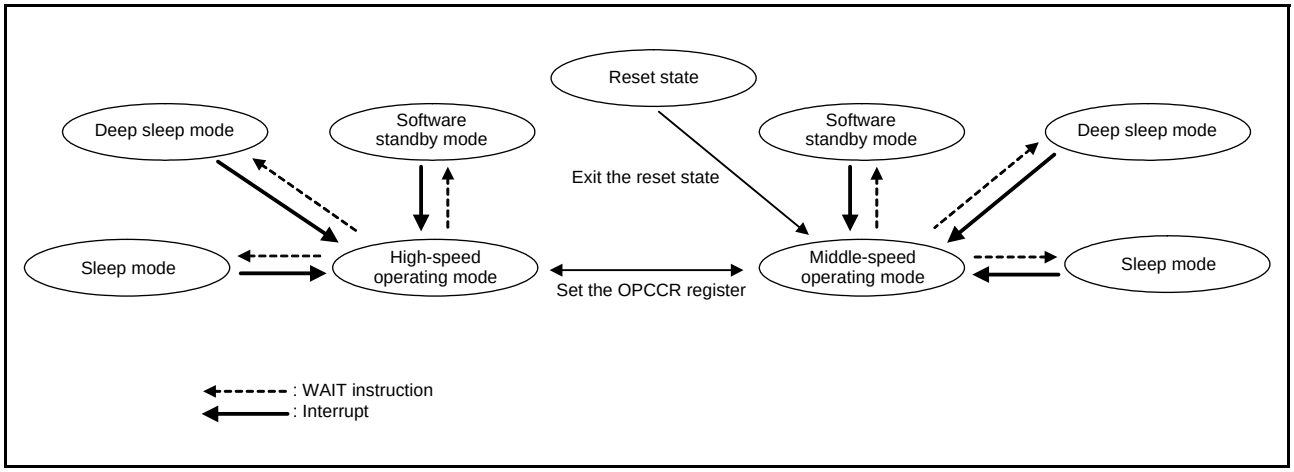


Figure 11.2 Operating Modes

- It is possible to return from sleep mode to the previous operating state used before entering sleep mode.
- After exiting the reset state, operation starts in middle-speed operating mode.

11.2 Register Descriptions

11.2.1 Standby Control Register (SBYCR)

Address(es): 0008 000Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	SSBY	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b14 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	SSBY	Software Standby	0: Set entry to sleep mode or deep sleep mode after the WAIT instruction is executed 1: Set entry to software standby mode after the WAIT instruction is executed	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

SSBY Bit (Software Standby)

The SSBY bit specifies the transition destination after the WAIT instruction is executed.
When the SSBY bit is set to 1, the MCU enters software standby mode after execution of the WAIT instruction. When the MCU returns to normal mode after an interrupt has triggered and exits from software standby mode, the SSBY bit remains 1. The SSBY bit can be cleared by writing 0 to the SSBY bit.

When the oscillation stop detection function enable bit (OSTDCR.OSTDE) in the oscillation stop detection control register is 1, the set value of the SSBY bit is invalid. Even if the SSBY bit is 1, the MCU will enter sleep mode or deep sleep mode after execution of the WAIT instruction.

11.2.2 Module Stop Control Register A (MSTPCRA)

Address(es): 0008 0010h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	MSTPA28	—	—	—	—	—	—	—	—	MSTPA19	—	MSTPA17	—
Value after reset:	1	1	1	0	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MSTPA15	—	—	—	—	—	MSTPA9	—	—	—	—	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b9	MSTPA9	Multifunction Timer Pulse Unit 3 Module Stop	Target module: MTU (MTU0 to MTU5) 0: This module clock is enabled 1: This module clock is disabled	R/W
b14 to b10	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b15	MSTPA15	Compare Match Timer (Unit 0) Module Stop	Target module: CMT unit 0 (CMT0, CMT1) 0: This module clock is enabled 1: This module clock is disabled	R/W
b16	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b17	MSTPA17	12-Bit A/D Converter Module Stop	Target module: S12AD 0: This module clock is enabled 1: This module clock is disabled	R/W
b18	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b19	MSTPA19	D/A Converter for generating comparator C reference voltage Module Stop	Target module: DA 0: This module clock is enabled 1: This module clock is disabled	R/W
b27 to b20	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b28	MSTPA28	Data Transfer Controller Module Stop	Target module: DTC 0: This module clock is enabled 1: This module clock is disabled	R/W
b31 to b29	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

11.2.3 Module Stop Control Register B (MSTPCRB)

Address(es): 0008 0014h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	MSTPB30	—	—	—	MSTPB26	—	—	MSTPB23	—	MSTPB21	—	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	MSTPB10	—	—	—	MSTPB6	—	MSTPB4	—	—	—	—
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b4	MSTPB4	Serial Communication Interface SCIh Module Stop	Target module: SCIh (SCI12) 0: This module clock is enabled 1: This module clock is disabled	R/W
b5	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b6	MSTPB6	DOC Module Stop	Target module: DOC 0: This module clock is enabled 1: This module clock is disabled	R/W
b9 to b7	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b10	MSTPB10	Comparator C Module Stop	Target module: CMPC 0: This module clock is enabled 1: This module clock is disabled	R/W
b20 to b11	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b21	MSTPB21	I^2C Bus Interface 0 Module Stop	Target module: RIIC0 0: This module clock is enabled 1: This module clock is disabled	R/W
b22	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b23	MSTPB23	CRC Calculator Module Stop	Target module: CRC 0: This module clock is enabled 1: This module clock is disabled	R/W
b25, b24	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b26	MSTPB26	Serial Communication Interface 5 Module Stop	Target module: SCI5 0: This module clock is enabled 1: This module clock is disabled	R/W
b29 to b27	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b30	MSTPB30	Serial Communication Interface 1 Module Stop	Target module: SCI1 0: This module clock is enabled 1: This module clock is disabled	R/W
b31	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

11.2.4 Module Stop Control Register C (MSTPCRC)

Address(es): 0008 0018h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	DSLPE	—	—	—	—	—	—	—	—	—	—	—	MSTPC19	—	—	—
Value after reset:	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTPC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MSTPC0	RAM0 Module Stop*1	Target module: RAM0 (0000 0000h to 0000 2FFFh) 0: RAM0 operating 1: RAM0 stopped	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b19	MSTPC19	Clock Frequency Accuracy Measurement Circuit Module Stop*2	Target module: CAC 0: This module clock is enabled 1: This module clock is disabled	R/W
b30 to b20	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b31	DSLPE	Deep Sleep Mode Enable	0: Deep sleep mode is disabled 1: Deep sleep mode is enabled	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.
Note 1. The corresponding MSTPC0 bit should not be set to 1 during access to the RAM. The corresponding RAM should not be accessed while the MSTPC0 bit is 1.
Note 2. The MSTPC19 bit should be rewritten while the oscillation of the clock to be controlled by this bit is stable. For entering software standby mode after rewriting this bit, wait for two cycles of the slowest clock among the clocks output by the oscillators actually oscillating and execute the WAIT instruction.

DSLPE Bit (Deep Sleep Mode Enable)

The DSLPE bit enables or disables a transition to deep sleep mode.

When the CPU executes the WAIT instruction with the DSLPE bit set to 1 and the SBYCR.SSBY and MSTPCRA.MSTPA28 bits meet specified conditions, the MCU enters deep sleep mode. For details, refer to section 11.6.2, Deep Sleep Mode.

11.2.5 Operating Power Control Register (OPCCR)

Address(es): 0008 00A0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	OPCMTSF	—	OPCM[2:0]		
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	OPCM[2:0]	Operating Power Control Mode Select	b2 b0 0 0 0: High-speed operating mode 0 1 0: Middle-speed operating mode Settings other than above are prohibited.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	OPCMTSF	Operating Power Control Mode Transition Status Flag	0: Transition completed 1: During transition	R
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: Set the PRCR.PRC1 bit to 1 (write enabled) before rewriting this register.

The OPCCR register is used to reduce power consumption in normal operating mode, sleep mode, and deep sleep mode. Power consumption can be reduced according to the operating frequency and operating voltage to be used by the OPCCR setting.

The OPCCR register cannot be rewritten under the following conditions:

- When the OPCCR.OPCMTSF flag is 1 (during transition)
- Time period from WAIT instruction execution for a sleep mode transition, until exit from sleep mode to normal operation
- Time period from WAIT instruction execution for a deep sleep mode transition, until exit from deep sleep mode to normal operation

The OPCCR register cannot be rewritten while the flash memory is being programmed or erased (P/E).
For the procedures of changing operating power control modes, refer to Function in section 11.5, Function for Lower Operating Power Consumption.
During a transition to an operating power control mode (while the OPCCR.OPCMTSF flag is 1), a correct value cannot be read from the E2 DataFlash. If a setting is made so that the E2 DataFlash is read using a DTC transfer, stop the DTC module before rewriting the OPCCR.OPCM[2:0] bits.
During sleep mode or mode transitions, do not write to the registers related to system control (indicated by 'SYSTEM' in the Module Symbol column in Table 5.1, List of I/O Registers (Address Order)).

OPCM[2:0] Bits (Operating Power Control Mode Select)

The OPCM[2:0] bits select operating power control mode in normal operating mode, sleep mode, and deep sleep mode. Table 11.3 shows the relationship between operating power control modes, the OPCM[2:0] bit settings, and the operating frequency and voltage ranges.

OPCMTSF Flag (Operating Power Control Mode Transition Status Flag)

This flag indicates the switching control state during and after operating power mode transition.
This flag becomes 1 when the value of the OPCM[2:0] bits is rewritten, and 0 when mode transition is completed. Read this flag and confirm that it is 0 before proceeding to the next processing. Only rewrite the OPCM[2:0] bits when this

flag is 0.

Table 11.3 Operating Frequency and Voltage Ranges in Operating Power Control Modes

Operating Power Control Mode	OPCM [2:0] Bits	Operating Voltage Range	Operating Frequency Range				
			Flash Memory Read Frequency				Flash Memory Programming/ Erasure Frequency
			ICLK	FCLK	PCLKD	PCLKB	FCLK
High-speed operating mode	000b	2.7 to 5.5 V	Up to 32 MHz	Up to 32 MHz	Up to 32 MHz	Up to 32 MHz	1 to 32 MHz
Middle-speed operating mode	010b	2.7 to 5.5 V	Up to 12 MHz	Up to 12 MHz	Up to 12 MHz	Up to 12 MHz	1 to 12 MHz

Note: When using the FCLK at lower than 4 MHz during programming or erasing the flash memory, the frequency can be set to 1, 2, or 3 MHz.

Each operating power control mode is described below.

- High-Speed Operating Mode
  The maximum operating frequency during FLASH read is 32 MHz for ICLK, FCLK, PCLKB, and PCLKD.
  During FLASH programming/erasure, the operating frequency range is 1 to 32 MHz.

Note: When using the FCLK at lower than 4 MHz during programming or erasing the flash memory, the frequency can be set to 1, 2, or 3 MHz.

- Middle-Speed Operating Mode
  As compared to high-speed operating mode, this mode reduces power consumption for low-speed operation.
  The maximum operating frequency during FLASH read is 12 MHz for ICLK, FCLK, PCLKB, and PCLKD.
  During FLASH programming/erasure, the operating frequency range is 1 to 12 MHz.
  The power consumption of this mode is lower than that of high speed mode under the same conditions.
  After a reset is canceled, operation is started from this mode.

Note: When using the FCLK at lower than 4 MHz during programming or erasing the flash memory, the frequency can be set to 1, 2, or 3 MHz.

11.3 Reducing Power Consumption by Switching Clock Signals

The clock frequency can change by setting the SCKCR.FCK[3:0], ICK[3:0], PCKB[3:0], and PCKD[3:0] bits. The CPU, DTC, ROM, and RAM clocks can be set by the ICK[3:0] bits. The peripheral module clocks can be set by the PCKB[3:0] and PCKD[3:0] bits.
The flash memory clock can be set by the FCK[3:0] bits.
For details, refer to section 9, Clock Generation Circuit.

11.4 Module Stop Function

The module stop function can be set for each on-chip peripheral module.
When the MSTPmi bit (m = A to C; i = 0 to 31) in MSTPCRA to MSTPCRC is set to 1, the specified module stops operating and enters the module stop state, but the CPU continues to operate independently. When the corresponding MSTPmi bit is set to 0, the module exits the module state and restarts operating at the end of the bus cycle. The internal states of modules are retained in the module stop state.
After a reset is canceled, all modules other than the DTC, and on-chip RAM are in the module stop state. Basically the registers in the module stop state cannot be read or written. However, note that data may be written to these registers if write access is made immediately after the setting of the module stop state. To avoid this, always write to the module stop registers after confirming that the last register setting is done.

11.5 Function for Lower Operating Power Consumption

By selecting an appropriate operating power control mode according to the operating frequency and operating voltage, power consumption can be reduced in normal mode, sleep mode, and deep sleep mode.

11.5.1 Setting Operating Power Control Mode

Examples of the procedures for switching operating power control modes are shown below:

(1) Switching from Normal Power Consumption Mode to Low Power Consumption Mode

- From high-speed operating mode to middle-speed operating mode

(High-speed operation in high-speed operating mode)
↓
Set the frequency of each clock to lower than the maximum operating frequency for middle-speed operating mode
↓
Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)
↓
Set the OPCCR.OPCM[2:0] bits to 010b (middle-speed operating mode)
↓
Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)
↓
(Middle-speed operation in middle-speed operating mode)

(2) Switching from Low Power Consumption Mode to Normal Power Consumption Mode

- From middle-speed operating mode to high-speed operating mode

Middle-speed operation in middle-speed operating mode
↓
Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)
↓
Set the OPCCR.OPCM[2:0] bit to 0 (high-speed operating mode)
↓
Confirm that the OPCCR.OPCMTSF flag is 0 (transition completed)
↓
Set the frequency of each clock to lower than the maximum operating frequency for high-speed operating mode
↓
High-speed operation in high-speed operating mode

11.6 Low Power Consumption Modes

11.6.1 Sleep Mode

11.6.1.1 Entry to Sleep Mode

When the WAIT instruction is executed while the SBYCR.SSBY bit is 0, the CPU enters sleep mode. In sleep mode, the CPU stops operating but the contents of its internal registers are retained. Other peripheral functions do not stop.
Counting by the IWDT stops if a transition to sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1. In the same way, counting by the IWDT stops if a transition to sleep mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 1.
Furthermore, counting by the IWDT continues if a transition to sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low power consumption modes). In the same way, counting by the IWDT continues if a transition to sleep mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 0.
To use sleep mode, make the following settings and then execute a WAIT instruction.

(1) Set the PSW.I bit*1 of the CPU to 0.
(2) Set the interrupt request destination*2 to be used for exit from sleep mode.
(3) Set the priority*3 of the interrupt to be used for exit from sleep mode to a level higher than the setting of the PSW.IPL[3:0] bits*1 of the CPU.
(4) Set the IERm.IENj bit*3 to 1 for the interrupt.
(5) Read the I/O register that is written last and confirm that the written value has been reflected.
(6) Execute the WAIT instruction (this automatically sets the I bit*1 in the PSW of the CPU to 1).

Note 1. For details, refer to section 2, CPU.
Note 2. For details, refer to section 14.4.3, Selecting Interrupt Request Destinations.
Note 3. For details, refer to section 14, Interrupt Controller (ICUb).

11.6.1.2 Exit from Sleep Mode

Exit from sleep mode is initiated by any interrupt, a RES# pin reset, a power-on reset, a voltage monitoring reset, or a reset caused by an IWDT underflow.

- Initiated by an interrupt
  An interrupt initiates exit from sleep mode and the interrupt exception handling starts. If a maskable interrupt has been masked by the CPU (the priority level*1 of the interrupt has been set to a value lower than that of the PSW.IPL[3:0] bits*2 of the CPU), sleep mode is not exited.
- Initiated by a RES# pin reset
  When the RES# pin is driven low, the MCU enters the reset state. When the RES# pin is driven high after the reset signal is input for a predetermined time period, the CPU starts the reset exception handling.
- Initiated by a power-on reset
  A power-on reset asserts a reset to the MCU.
  When a power-on reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by a voltage monitoring reset
  A voltage monitoring reset asserts a reset to the MCU.
  When a voltage monitoring reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by an independent watchdog timer reset
  An internal reset generated by an IWDT underflow asserts a reset to the MCU. However, when IWDT counting is stopped in sleep mode by setting OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1, the IWDT is stopped in sleep mode and sleep mode is not exited by the independent watchdog timer reset.

Note 1. For details, refer to section 14, Interrupt Controller (ICUb).
Note 2. For details, refer to section 2, CPU.

11.6.2 Deep Sleep Mode

11.6.2.1 Entry to Deep Sleep Mode

When a WAIT instruction is executed with the MSTPCRC.DSLPE bit set to 1, the MSTPCRA.MSTPA28 bit set to 1, and the SBYCR.SSBY bit cleared to 0, a transition to deep sleep mode is made.*1
In deep sleep mode, the CPU and the DTC, ROM, and RAM clocks stop. Peripheral functions do not stop.
Counting by the IWDT stops if a transition to deep sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1. In the same way, counting by the IWDT stops if a transition to deep sleep mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 1.
Furthermore, counting by the IWDT continues if a transition to deep sleep mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low power consumption modes). In the same way, counting by the IWDT continues if a transition to deep sleep mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 0.

To use deep sleep mode, make the following settings and then execute a WAIT instruction.

(1) Set the PSW.I bit*2 of the CPU to 0.
(2) Set the interrupt request destination*3 to be used for exit from deep sleep mode.
(3) Set the priority*4 of the interrupt to be used for exit from deep sleep mode to a level higher than the setting of the PSW.IPL[3:0] bits*2 of the CPU.
(4) Set the IERm.IENj bit*4 to 1 for the interrupt.
(5) Read the I/O register that is written last and confirm that the written value has been reflected.
(6) Execute a WAIT instruction (executing a WAIT instruction causes automatic setting of the PSW.I bit*2 of the CPU to 1).

Note 1. Transition to deep sleep mode might not be possible, depending on the operating state of the DTC. Before setting the MSTPCRA.MSTPA28 bit to 1, set the DTCST.DTCST bit of the DTC to 0 to avoid activating the DTC.
Note 2. For details, refer to section 2, CPU.
Note 3. For details, refer to section 14.4.3, Selecting Interrupt Request Destinations.
Note 4. For details, refer to section 14, Interrupt Controller (ICUb).

11.6.2.2 Exit from Deep Sleep Mode

Exit from deep sleep mode is initiated by any interrupt, a RES# pin reset, a power-on reset, a voltage monitoring reset, or a reset caused by an IWDT underflow.

- Initiated by an interrupt
  An interrupt initiates exit from deep sleep mode and the interrupt exception handling starts. If a maskable interrupt has been masked by the CPU (the priority level*1 of the interrupt has been set to a value lower than that of the PSW.IPL[3:0] bits*2 of the CPU), deep sleep mode is not exited.
- Initiated by the RES# pin reset
  When the RES# pin is driven low, the MCU enters the reset state. When the RES# pin is driven high after the reset signal is input for a predetermined time period, the CPU starts the reset exception handling.
- Initiated by a power-on reset
  A power-on reset asserts a reset to the MCU.
  When a power-on reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by a voltage monitoring reset
  A voltage monitoring reset asserts a reset to the MCU.
  When a voltage monitoring reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by the independent watchdog timer
  An internal reset generated by an IWDT underflow asserts a reset to the MCU. However, when IWDT counting is stopped in deep sleep mode by setting OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1, the IWDT is stopped in deep sleep mode and deep sleep mode is not exited by the independent watchdog timer reset.

Note 1. For details, refer to section 14, Interrupt Controller (ICUb).
Note 2. For details, refer to section 2, CPU.

11.6.3 Software Standby Mode

11.6.3.1 Entry to Software Standby Mode

When a WAIT instruction is executed with the SBYCR.SSBY bit set to 1, a transition to software standby mode is made. In this mode, the CPU, on-chip peripheral functions stop. However, the contents of the CPU internal registers, RAM data, the states of on-chip peripheral functions, the I/O ports are retained. Software standby mode allows significant reduction in power consumption because the oscillator stops in this mode.
Set the DTCST.DTCST bit of the DTC to 0 before executing the WAIT instruction.
Counting by the IWDT stops if a transition to software standby mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 1. In the same way, counting by the IWDT stops if a transition to software standby mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 1.

Furthermore, counting by the IWDT continues if a transition to software standby mode is made while the IWDT is being used in auto-start mode and the OFS0.IWDTSLCSTP bit is 0 (counting by the IWDT continues through transitions to low power consumption modes). In the same way, counting by the IWDT continues if a transition to software standby mode is made while the IWDT is being used in register start mode and the IWDTCSTPR.SLCSTP bit is 0.
To use software standby mode, make the following settings and then execute a WAIT instruction.

(1) Set the PSW.I bit*1 of the CPU to 0.
(2) Set the interrupt request destination*2 to be used for recovery from software standby mode to the CPU.
(3) Set the priority*3 of the interrupt to be used for recovery from software standby mode to a level higher than the setting of the PSW.IPL[3:0] bits*1 of the CPU.
(4) Set the IERm.IENj bit*3 to 1 for the interrupt.
(5) Read the I/O register that is written last and confirm that the written value has been reflected.
(6) Execute a WAIT instruction (executing a WAIT instruction causes automatic setting of the PSW.I bit*1 of the CPU to 1).

Note 1. For details, refer to section 2, CPU.
Note 2. For details, refer to section 14.4.3, Selecting Interrupt Request Destinations.
Note 3. For details, refer to section 14, Interrupt Controller (ICUb).

11.6.3.2 Exit from Software Standby Mode

Exit from software standby mode is initiated by an external pin interrupt (the NMI or IRQ0 to IRQ5), peripheral function interrupts (the IWDT, and voltage monitoring), a RES# pin reset, a power-on reset, a voltage monitoring reset, or an independent watchdog timer reset. When any trigger which initiates exit from software standby mode is asserted, the oscillators which were operating before entry to software standby mode restart operation. After the oscillation of all these oscillators has been stabilized, operation returns from software standby mode.

- Initiated by an interrupt
  When an interrupt request from among the NMI, IRQ0 to IRQ5, IWDT, voltage monitoring interrupts is generated, each of the oscillators which was operating before the transition to software standby mode resumes oscillation. After the oscillation stabilization wait time of each oscillator set by the MOSCWTCR.MSTS[4:0] bits has elapsed, the MCU exits software standby mode and interrupt exception processing starts.
- Initiated by a RES# pin reset
  Clock oscillation starts when the low level is applied to the RES# pin. Clock supply for the MCU starts at the same time. Keep the level on the RES# pin low over the time required for oscillation of the clocks to become stable. Reset exception processing starts when the high level is applied to the RES# pin.
- Initiated by a power-on reset
  A power-on reset asserts a reset to the MCU.
  When a power-on reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by a voltage monitoring reset
  A voltage monitoring reset asserts a reset to the MCU.
  When a voltage monitoring reset is negated by a rise in the supply voltage, the CPU starts the reset exception handling.
- Initiated by an independent watchdog timer reset
  An internal reset generated by an IWDT underflow asserts a reset to the MCU.
  Note that the independent watchdog timer is stopped in software standby mode due to the register settings (OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1) in software standby mode. In that case, exit from software standby mode by the independent watchdog timer reset cannot be done.

11.6.3.3 Example of Software Standby Mode Application

Figure 11.3 shows an example of entry to software standby mode by the falling edge of the IRQn pin, and exit from software standby mode by the rising edge of the IRQn pin.

In this example, an IRQn interrupt is accepted with the IRQCRi.IRQMD[1:0] bits of the ICU set to 01b (falling edge), and then the IRQCRi.IRQMD[1:0] bits are set to 10b (rising edge). After that, the SBYCR.SSBY bit is set to 1 and the WAIT instruction is executed. Thus entry to software standby mode is completed. After that, exit from software standby mode is initiated by the rising edge of the IRQn pin.

To exit software standby mode, settings of the interrupt controller (ICU) are also necessary. For details, refer to section 14, Interrupt Controller (ICUb).

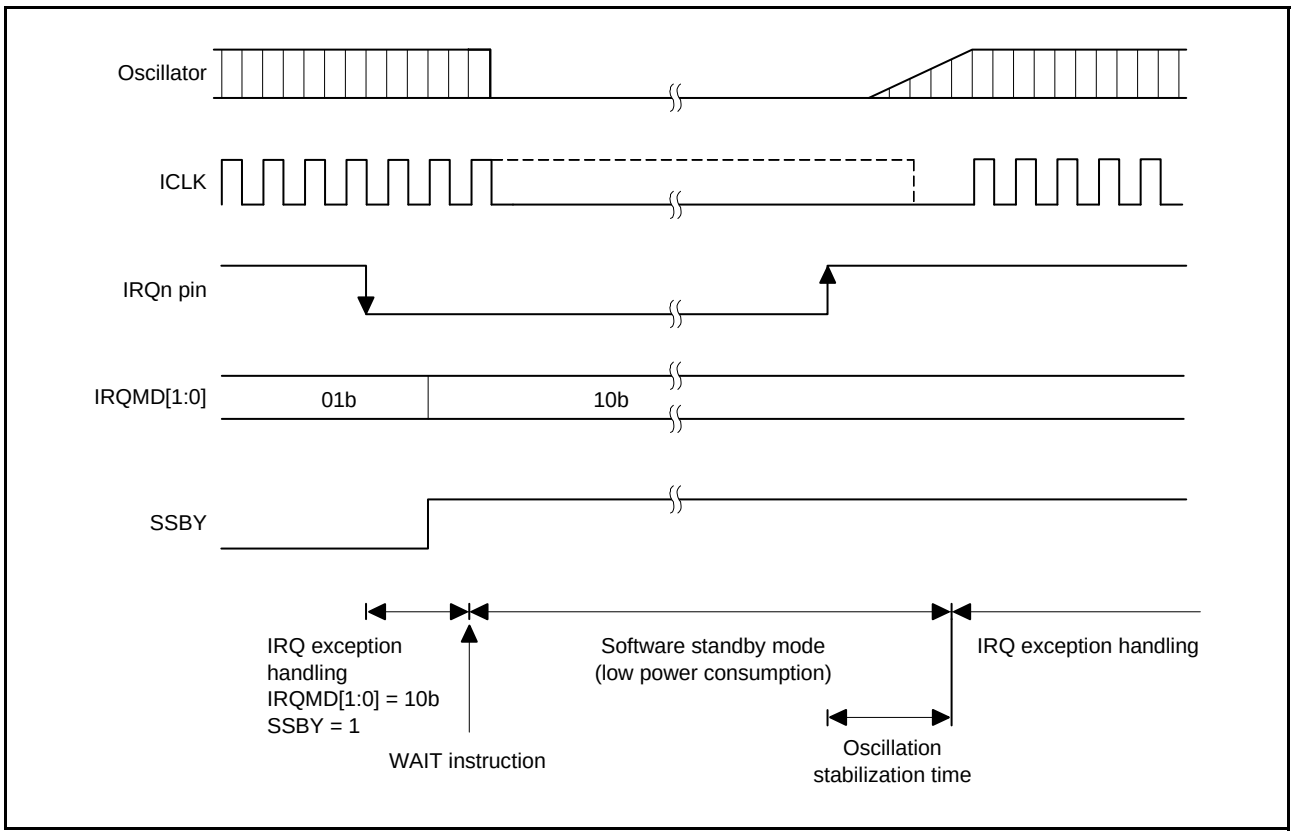


Figure 11.3 Example of Software Standby Mode Application

11.7 Usage Notes

11.7.1 I/O Port States

I/O port states are retained in software standby mode. Therefore, the supply current is not reduced if output signals are high level.

11.7.2 Module Stop State of DTC

Before setting the MSTPCRA.MSTPA28 bit to 1, set the DTCST.DTCST bit of the DTC to 0 to avoid activating the DTC.
For details, refer to section 16, Data Transfer Controller (DTCb).

11.7.3 On-Chip Peripheral Module Interrupts

Interrupts do not operate in the module stop state. Therefore, if the module stop state is made after an interrupt request is generated, a CPU interrupt source or a DTC startup source cannot be cleared. For this reason, disable interrupts before entering the module stop state.

11.7.4 Write Access to MSTPCRA, MSTPCRB, and MSTPCRC

Write accesses to MSTPCRA, MSTPCRB, and MSTPCRC should be made only by the CPU.

11.7.5 Timing of WAIT Instructions

The WAIT instruction is executed before completion of the preceding register write. The WAIT instruction being executed before the register setting is modified may cause unintended operation. To avoid this, always execute the WAIT instruction after confirming that the last register setting is done.

11.7.6 Rewrite the Register by DTC in Sleep Mode

Depending on the settings of the OFS0.IWDTSLCSTP bit and IWDTCSTPR.SLCSTP bit, the IWDT may also stop in sleep mode. To avoid this, do not set up the DTC to rewrite any registers related to the IWDT in sleep mode.

12. Register Write Protection Function

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

Table 12.1 lists the association between the PRCR bits and the registers to be protected.

Table 12.1 Association between PRCR Bits and Registers to be Protected

PRCR Bit	Register to be Protected
PRC0	• Registers related to the clock generation circuit: SCKCR, SCKCR3, PLLCR, PLLCR2, MOSCCR, LOCOCR, ILOCOCR, HOCOCR, OSTDCR, OSTDSR, LOCOTRR, ILOCOTRR, HOCOTRR0
PRC1	• Register related to the operating modes: SYSCR1 • Registers related to low power consumption functions: SBYCR, MSTPCRA, MSTPCRB, MSTPCRC, OPCCR • Registers related to the clock generation circuit: MOFCR, MOSCWTCR • Software reset register: SWRR
PRC3	• Registers related to the LVD: LVCMPCR, LVDLVLR, LVD1CR0, LVD1CR1, LVD1SR, LVD2CR0, LVD2CR1, LVD2SR

12.1 Register Descriptions

12.1.1 Protect Register (PRCR)

Address(es): 0008 03FEh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	PRKEY[7:0]								—	—	—	—	PRC3	—	PRC1	PRC0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PRC0	Protect Bit 0	Enables writing to the registers related to the clock generation circuit. 0: Write disabled 1: Write enabled	R/W
b1	PRC1	Protect Bit 1	Enables writing to the registers related to operating modes, low power consumption functions, the clock generation circuit, and software reset. 0: Write disabled 1: Write enabled	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	PRC3	Protect Bit 3	Enables writing to the registers related to the LVD. 0: Write disabled 1: Write enabled	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	PRKEY[7:0]	PRC Key Code	These bits control permission and prohibition of writing to the PRCR register. To modify the PRCR register, write A5h to the 8 higher-order bits and the desired value to the 8 lower-order bits as a 16-bit unit.	R/W*1

Note 1. Write data is not retained.

PRCi Bits (Protect Bit i) (i = 0, 1, 3)

These bits enable or disable writing to the corresponding registers to be protected.

Setting the PRCi bits to 1 and 0 enable and disable writing to the corresponding registers to be protected, respectively.

13. Exception Handling

13.1 Exception Events

During execution of a program by the CPU, the occurrence of a certain event may cause execution of that program to be suspended and execution of another program to be started. Such kinds of events are called exception events.
The RX CPU supports seven types of exceptions. The types of exception events are shown in Figure 13.1.
The occurrence of an exception causes the processor mode to switch to supervisor mode.

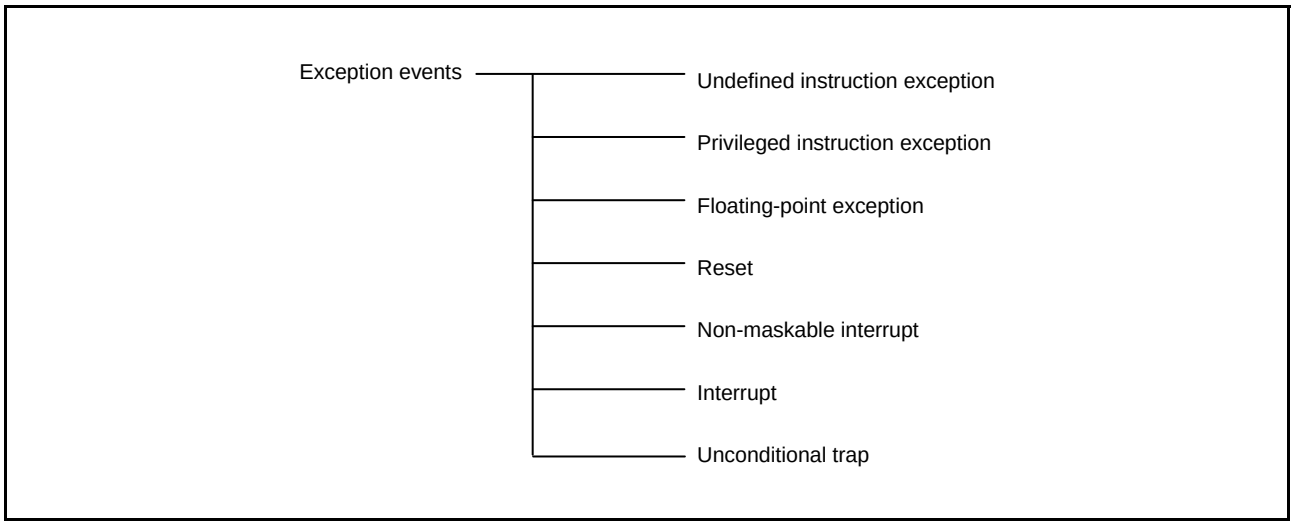


Figure 13.1 Types of Exception Events

13.1.1 Undefined Instruction Exception

An undefined instruction exception occurs when execution of an undefined instruction (an instruction not implemented) is detected.

13.1.2 Privileged Instruction Exception

A privileged instruction exception occurs when execution of a privileged instruction is detected in user mode. Privileged instructions can be executed only in supervisor mode.

13.1.3 Floating-Point Exception

Floating-point exceptions include the five exception events (overflow, underflow, inexact, division-by-zero, and invalid operation) specified in the IEEE754 standard and another floating-point exception that is generated on detection of unimplemented processing. The exception handling of floating-point exceptions is prohibited when the EX, EU, EZ, EO, or EV bit in FPSW is 0.

13.1.4 Reset

A reset is generated by input of a reset signal to the CPU. This has the highest priority of any exception and is always accepted.

13.1.5 Non-Maskable Interrupt

A non-maskable interrupt is generated by input of a non-maskable interrupt signal to the CPU and is only used when a fatal fault is considered to have occurred in the system. Never use the non-maskable interrupt with an attempt to return to the program that was being executed at the time of interrupt generation after the exception handling routine is ended.

13.1.6 Interrupts

Interrupts are generated by the input of interrupt signals to the CPU. A fast interrupt can be selected as the interrupt with the highest priority. In the case of the fast interrupt, hardware pre-processing and hardware post-processing are handled fast. The priority level of the fast interrupt is 15 (the highest). The exception handling of interrupts is masked when the I bit in PSW is 0.

13.1.7 Unconditional Trap

An unconditional trap is generated when the INT or BRK instruction is executed.

13.2 Exception Handling Procedure

In the exception handling, part of the processing is handled automatically by hardware and part of it is handled by a program (exception handling routine) that has been written by the user. Figure 13.2 shows the processing procedure when an exception other than a reset is accepted.

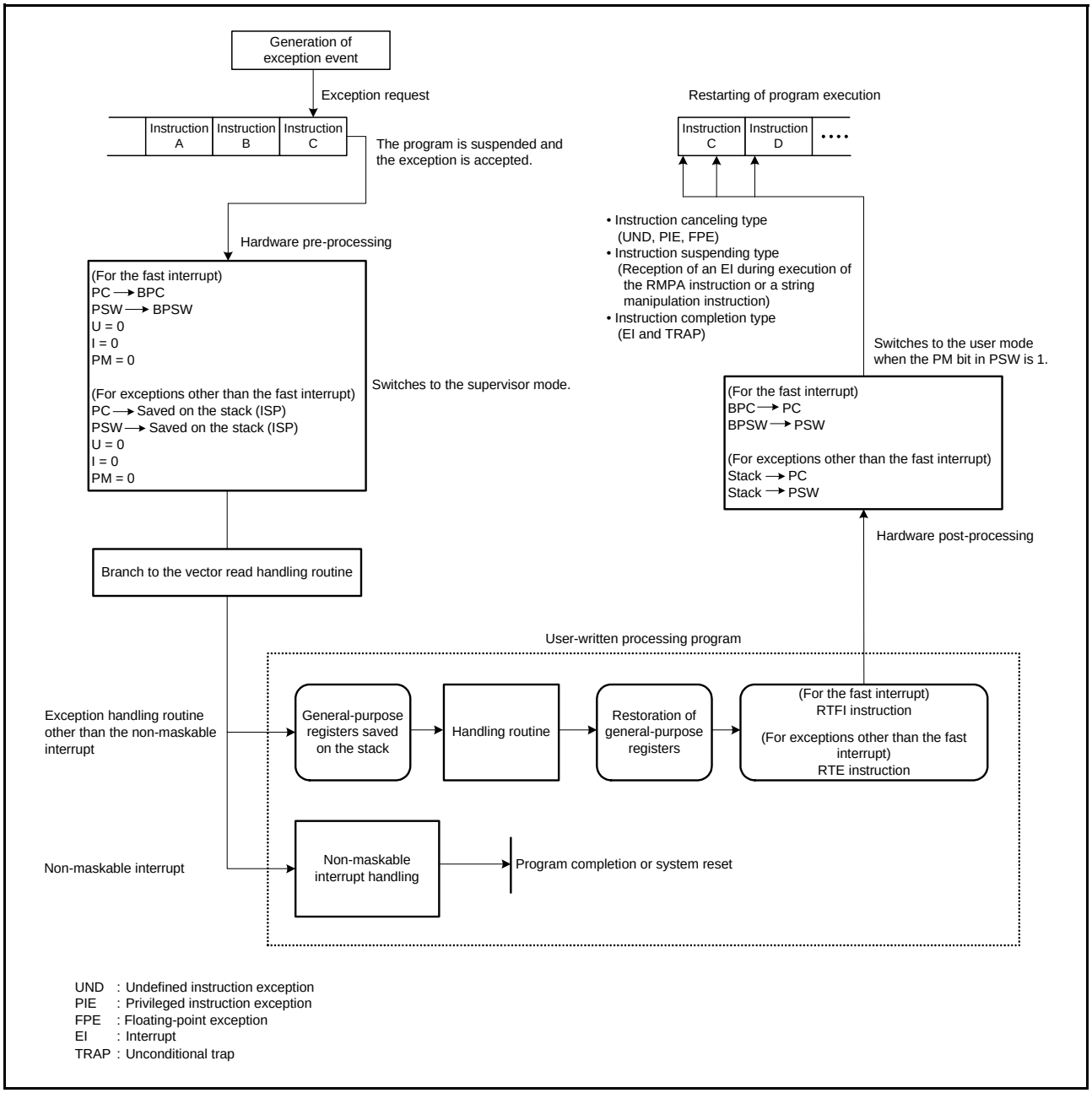


Figure 13.2 Outline of Exception Handling Procedure

When an exception is accepted, hardware processing by the RX CPU is followed by access to the vector to acquire the address of the branch destination. In the vector, a vector address is allocated to each exception, and the branch destination address of the exception handling routine is written to each vector address.
Hardware pre-processing by the RX CPU handles saving of the contents of the program counter (PC) and processor status word (PSW). In the case of a fast interrupt, the contents are saved in the backup PC (BPC) and the backup PSW (BPSW), respectively. In the case of exceptions other than a fast interrupt, the contents are saved in the stack area.
General purpose registers and control registers other than the PC and PSW that are to be used within the exception handling routine must be saved on the stack by a user program at the start of the exception handling routine.
On completion of processing by an exception handling routine, execution is restored from the exception handling routine to the original program by saving the registers saved on the stack and executing the RTE instruction. For return from a fast interrupt, the RTFI instruction is used instead. In the case of a non-maskable interrupt, however, finish the program or reset the system without returning to the original program.
Hardware post-processing by the RX CPU handles restoration of the contents of PC and PSW. In the case of a fast interrupt, the values of BPC and BPSW are restored to PC and PSW, respectively. In the case of exceptions other than a fast interrupt, the values are restored from the stack to PC and PSW.

13.3 Acceptance of Exception Events

When an exception occurs, the CPU suspends the execution of the program and processing branches to the exception handling routine.

13.3.1 Acceptance Timing and Saved PC Value

Table 13.1 lists the timing of acceptance and the program counter (PC) value to be saved for each exception event.

Table 13.1 Acceptance Timing and Saved PC Value

Exception Event		Type of Handling	Acceptance Timing	Value Saved in BPC or on the Stack
Undefined instruction exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Privileged instruction exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Floating-point exception		Instruction canceling type	During instruction execution	PC value of the instruction that generated the exception
Reset		Instruction abandonment type	Any machine cycle	None
Non-maskable interrupt	During execution of the RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions	Instruction suspending type	During instruction execution	PC value of the instruction being executed
	Other than above	Instruction completion type	At the next break between instructions	PC value of the next instruction
Interrupt	During execution of the RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, and SWHILE instructions	Instruction suspending type	During instruction execution	PC value of the instruction being executed
	Other than above	Instruction completion type	At the next break between instructions	PC value of the next instruction
Unconditional trap		Instruction completion type	At the next break between instructions	PC value of the next instruction

13.3.2 Vector and Site for Saving the Values in the PC and PSW

The vector for each type of exception and the site for saving the values of the program counter (PC) and processor status word (PSW) are listed in Table 13.2.

Table 13.2 Vector and Site for Saving the Values in the PC and PSW

Exception		Vector	Site for Saving the Values in the PC and PSW
Undefined instruction exception		Fixed vector table	Stack
Privileged instruction exception		Fixed vector table	Stack
Floating-point exception		Fixed vector table	Stack
Reset		Fixed vector table	Nowhere
Non-maskable interrupt		Fixed vector table	Stack
Interrupt	Fast interrupt	FINTV	BPC and BPSW
	Other than above	Relocatable vector table (INTB)	Stack
Unconditional trap		Relocatable vector table (INTB)	Stack

13.4 Hardware Processing for Accepting and Returning from Exceptions

This section describes the hardware processing for accepting and returning from exceptions other than a reset.

(1) Hardware Pre-Processing for Accepting an Exception

(a) Saving PSW

- For a fast interrupt
  PSW → BPSW
- For exceptions other than a fast interrupt
  PSW → Stack

Note: The values in FPSW are not saved by hardware pre-processing. Therefore, if floating-point instructions are to be used within an exception handling routine, the user must save these values on the stack within the exception handling routine.

(b) Updating PM, U, and I Bits in PSW

I: Set to 0
U: Set to 0
PM: Set to 0

(c) Saving PC

- For a fast interrupt
  PC → BPC
- For exceptions other than a fast interrupt
  PC → Stack

(d) Setting Branch Destination Address of Exception Handling Routine in PC

Processing is shifted to the exception handling routine by acquiring the vector corresponding to the exception and then branching accordingly.

(2) Hardware Post-Processing for Execution of RTE and RTFI Instructions

(a) Restoring PSW

- For a fast interrupt
  BPSW → PSW
- For exceptions other than a fast interrupt
  Stack → PSW

(b) Restoring PC

- For a fast interrupt
  BPC → PC
- For exceptions other than a fast interrupt
  Stack → PC

13.5 Hardware Pre-Processing

The hardware pre-processing from reception of each exception request to execution of the associated exception handling routine are explained below.

13.5.1 Undefined Instruction Exception

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from address FFFF FFDCh.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.5.2 Privileged Instruction Exception

1. The value in the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from address FFFF FFD0h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.5.3 Floating-Point Exception

1. The value in the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are cleared to 0.
3. The value of the program counter (PC) is saved on the stack (ISP).
4. The vector is fetched from address FFFF FFE4h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.5.4 Reset

1. The control registers are initialized.
2. The vector is fetched from address FFFF FFFCh.
3. The fetched vector is set to the PC.

13.5.5 Non-Maskable Interrupt

1. The value of the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved on the stack (ISP). For other instructions, the PC value of the next instruction is saved.
4. The processor interrupt priority level bits (IPL[3:0]) in PSW are set to Fh.
5. The vector is fetched from address FFFF FFF8h.
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.5.6 Interrupt

1. The value of the processor status word (PSW) is saved on the stack (ISP) or, for the fast interrupt, in the backup PSW (BPSW).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. If the interrupt was generated during the execution of an RMPA, SCMPU, SMOVB, SMOVF, SMOVU, SSTR, SUNTIL, or SWHILE instruction, the value of the program counter (PC) for that instruction is saved. For other instructions, the PC value of the next instruction is saved. Saving of the PC is in the backup PC (BPC) for fast interrupts.
4. The processor interrupt priority level bits (IPL[3:0]) in PSW indicate the interrupt priority level of the interrupt.
5. The vector for an interrupt source other than the fast interrupt is fetched from the relocatable vector table. For the fast interrupt, the address is fetched from the fast interrupt vector register (FINTV).
6. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.5.7 Unconditional Trap

1. The value in the processor status word (PSW) is saved on the stack (ISP).
2. The processor mode select bit (PM), the stack pointer select bit (U), and the interrupt enable bit (I) in PSW are set to 0.
3. The value of the program counter (PC) for the next instruction is saved on the stack (ISP).
4. For the INT instruction, the value at the vector corresponding to the INT instruction number is fetched from the relocatable vector table.
   For the BRK instruction, the value at the vector from the start address is fetched from the relocatable vector table.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

13.6 Return from Exception Handling Routine

Executing the instruction listed in Table 13.3 at the end of the corresponding exception handling routine restores the values of the program counter (PC) and processor status word (PSW) that were saved on the stack or in the control registers (BPC and BPSW) immediately before the exception handling sequence.

Table 13.3 Return from Exception Handling Routine

Exception		Instruction for Return
Undefined instruction exception		RTE
Privileged instruction exception		RTE
Floating-point exception		RTE
Reset		Return is impossible
Non-maskable interrupt		Return is impossible
Interrupt	Fast interrupt	RTFI
	Other than above	RTE
Unconditional trap		RTE

13.7 Priority of Exception Events

The priority of exception events is listed in Table 13.4. When multiple exceptions are generated at the same time, the exception with the highest priority is accepted first.

Table 13.4 Priority of Exception Events

Priority		Exception Event
High	1	Reset
↑	2	Non-maskable interrupt
	3	Interrupt
	4	Undefined instruction exception Privileged instruction exception
	5	Unconditional trap
Low	6	Floating-point exception

14. Interrupt Controller (ICUb)

14.1 Overview

The interrupt controller receives interrupt requests from peripheral modules and external pins, and generates an interrupt request to the CPU and a transfer request to the DTC.

Table 14.1 lists the specifications of the interrupt controller, and Figure 14.1 shows a block diagram of the interrupt controller.

Table 14.1 Specifications of Interrupt Controller

Item		Description
Interrupts	Peripheral function interrupts	• Interrupts from peripheral modules • Interrupt detection: Edge detection/level detection Edge detection or level detection is fixed for each source of connected peripheral modules.
	External pin interrupts	• Interrupts from pins IRQ0 to IRQ5 • Number of sources: 6 • Interrupt detection: Low level/falling edge/rising edge/rising and falling edges One of these detection methods can be set for each source. • Digital filter function: Supported
	Software interrupt	• Interrupt generated by writing to a register • One interrupt source
	Interrupt priority	Specified by registers.
	Fast interrupt function	Faster interrupt processing of the CPU can be set only for a single interrupt source.
	DTC control	Interrupt sources can be used to start the DTC.*1
Non-maskable interrupts	NMI pin interrupt	• Interrupt from the NMI pin • Interrupt detection: Falling edge/rising edge • Digital filter function: Supported
	Oscillation stop detection interrupt	Interrupt on detection of oscillation having stopped
	IWDT underflow/ refresh error	Interrupt on an underflow of the down counter or occurrence of a refresh error
	Voltage monitoring 1 interrupt	Voltage monitoring interrupt of voltage monitoring circuit 1 (LVD1)
	Voltage monitoring 2 interrupt	Voltage monitoring interrupt of voltage monitoring circuit 2 (LVD2)
Return from power-down modes		• Sleep mode, deep sleep mode: Return is initiated by non-maskable interrupts or any other interrupt source. • Software standby mode: Return is initiated by non-maskable interrupts, IRQ0 to IRQ5 interrupts.

Note 1. For the DTC trigger, refer to Table 14.3, Interrupt Vector Table.

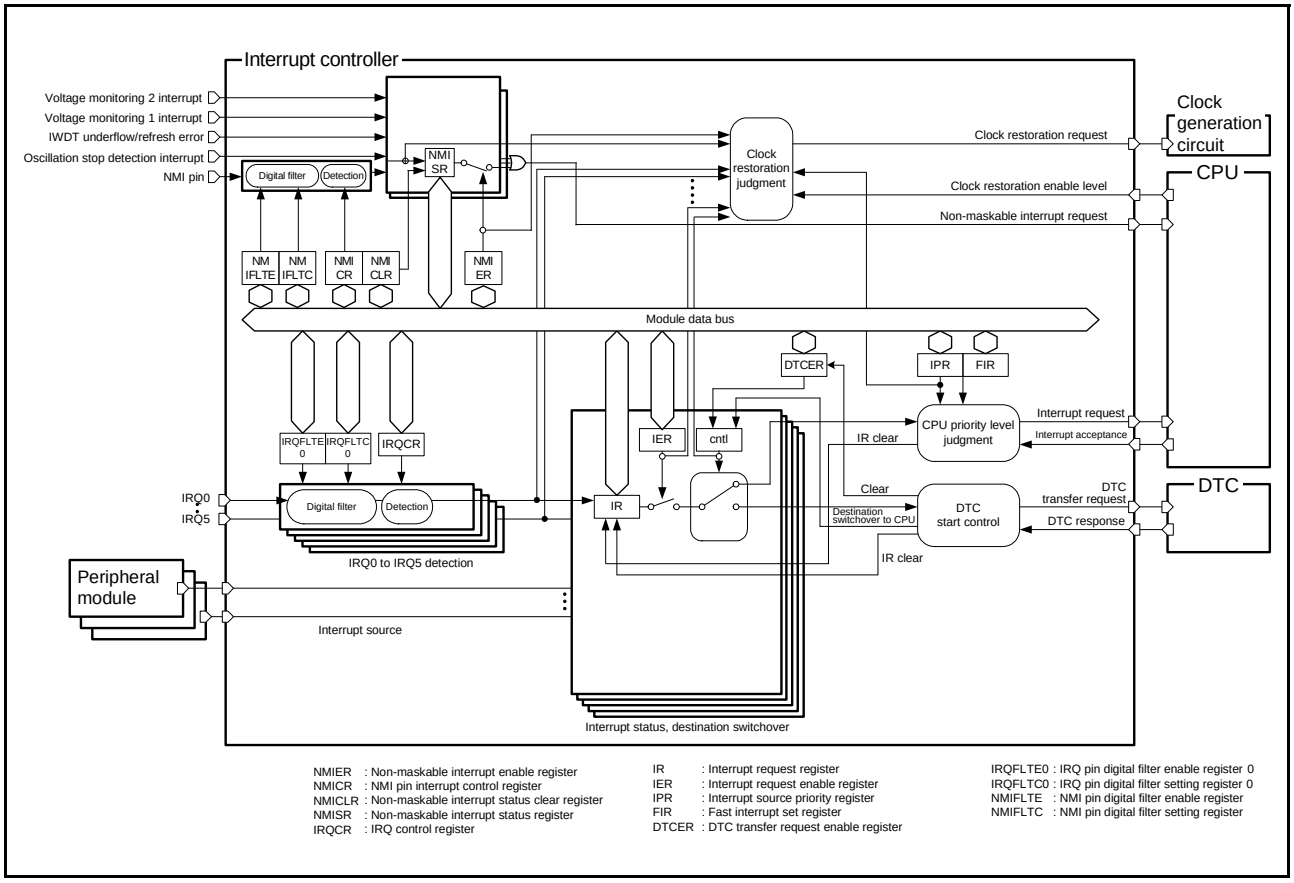


Figure 14.1 Block Diagram of Interrupt Controller

Table 14.2 lists the input/output pins of the interrupt controller.

Table 14.2 Pin Configuration of Interrupt Controller

Pin Name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQ0 to IRQ5	Input	External interrupt request pins

14.2 Register Descriptions

14.2.1 Interrupt Request Register n (IRn) (n = interrupt vector number)

Address(es): ICU.IR016 0008 7010h to ICU.IR255 0008 70FFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IR	Interrupt Status Flag	0: No interrupt request is generated 1: An interrupt request is generated	R/(W) *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. For an edge detection interrupt, only 0 can be written to this bit; do not write 1.
For a level detection interrupt, neither 0 nor 1 can be written.

IRn is provided for each interrupt source, where “n” indicates the interrupt vector number.
For the correspondence between interrupt sources and interrupt vector numbers, see Table 14.3, Interrupt Vector Table.

IR Flag (Interrupt Status Flag)

This flag is the status flag of an individual interrupt request. This flag is set to 1 when the corresponding interrupt request is generated. To detect an interrupt request, the interrupt request output should be enabled by the corresponding peripheral module interrupt enable bit.
There are two interrupt request detection methods: edge detection and level detection. For interrupts from peripheral modules, either edge detection or level detection is determined per interrupt source. For interrupts from IRQi (i = 0 to 5) pins, edge detection or level detection is selected by setting the corresponding IRQCRi.IRQMD[1:0] bits. For detection of the various interrupt sources, see Table 14.3, Interrupt Vector Table.

(1) Edge detection

[Setting condition]

- The flag is set to 1 in response to the generation of an interrupt request from the corresponding peripheral module or IRQi pin. For interrupt generation by the various peripheral modules, refer to the sections describing the modules.

[Clearing conditions]

- The flag is cleared to 0 when the interrupt request destination accepts the interrupt request.
- The IR flag is cleared to 0 by writing 0 to it. Note, however, that writing 0 to the IR flag is prohibited if the destination of the interrupt request is the DTC.

(2) Level detection

[Setting condition]

- The flag remains set to 1 while an interrupt request is being sent from the corresponding peripheral module or IRQi pin. For interrupt generation by the various peripheral modules, refer to the sections describing the modules.

[Clearing condition]

- The flag is cleared to 0 when the source of the interrupt request is cleared (it is not cleared when the interrupt request destination accepts the interrupt request). For clearing interrupts from the various peripheral modules, refer to the sections describing the modules.

When level detection has been selected for an IRQi pin, the interrupt request is withdrawn by driving the IRQi pin high.
Do not write 0 or 1 to the IR flag while level detection is selected.

14.2.2 Interrupt Request Enable Register m (IERm) (m = 02h to 1Fh)

Address(es): ICU.IER02 0008 7202h to ICU.IER1F 0008 721Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	IEN7	IEN6	IEN5	IEN4	IEN3	IEN2	IEN1	IEN0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IEN0	Interrupt Request Enable 0	0: Interrupt request is disabled 1: Interrupt request is enabled	R/W
b1	IEN1	Interrupt Request Enable 1		R/W
b2	IEN2	Interrupt Request Enable 2		R/W
b3	IEN3	Interrupt Request Enable 3		R/W
b4	IEN4	Interrupt Request Enable 4		R/W
b5	IEN5	Interrupt Request Enable 5		R/W
b6	IEN6	Interrupt Request Enable 6		R/W
b7	IEN7	Interrupt Request Enable 7		R/W

Note: Write 0 to the bit that corresponds to the vector number for reservation. These bits are read as 0.

IENj Bit (Interrupt Request Enable j) (j = 0 to 7)

When an IENj bit is 1, the corresponding interrupt request will be output to the destination selected for the request.
When an IENj bit is 0, the corresponding interrupt request will not be output to the destination selected for the request.
The setting of an IENj bit does not affect the IRn.IR flag (n = interrupt vector number). Even if the corresponding IENj bit is 0, the IR flag value changes according to the descriptions in section 14.2.1, Interrupt Request Register n (IRn) (n = interrupt vector number).
The IERm.IENj bit is set for each request source (vector number).
For the correspondence between interrupt sources and IERm.IENj bits, see Table 14.3, Interrupt Vector Table.
For the procedure for setting IERm.IENj bits during the selection of destinations for interrupt requests, refer to section 14.4.3, Selecting Interrupt Request Destinations.

14.2.3 Interrupt Source Priority Register n (IPRn) (n = interrupt vector number)

Address(es): ICU.IPR000 0008 7300h to ICU.IPR255 0008 73FFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	IPR[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IPR[3:0]	Interrupt Priority Level Select	b3 b0 0 0 0 0: Level 0 (interrupt disabled)*1 0 0 0 1: Level 1 0 0 1 0: Level 2 0 0 1 1: Level 3 0 1 0 0: Level 4 0 1 0 1: Level 5 0 1 1 0: Level 6 0 1 1 1: Level 7 1 0 0 0: Level 8 1 0 0 1: Level 9 1 0 1 0: Level 10 1 0 1 1: Level 11 1 1 0 0: Level 12 1 1 0 1: Level 13 1 1 1 0: Level 14 1 1 1 1: Level 15 (highest)	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the interrupt is specified as a fast interrupt, it can be issued even if the priority level is level 0.

For the correspondence between interrupt sources and IPRn registers, see Table 14.3, Interrupt Vector Table.

IPR[3:0] Bits (Interrupt Priority Level Select)

These bits specify the priority level of the corresponding interrupt source.

Priority levels specified by the IPR[3:0] bits are used only to determine the priority of interrupt requests to be transferred to the CPU, and do not affect transfer requests to the DTC.

The CPU accepts only interrupt requests higher than the priority level specified by the IPL[3:0] bits in PSW, and handles accepted interrupts.

If two or more interrupt requests are generated at the same time, their priority levels are compared with the value of the IPR[3:0] bits. If interrupt requests of the same priority level are generated at the same time, an interrupt source with a smaller vector number takes precedence.

These bits should be written to while an interrupt request is disabled (IERm.IENj bit = 0 (m = 02h to 1Fh, j = 0 to 7)).

14.2.4 Fast Interrupt Set Register (FIR)

Address(es): ICU.FIR 0008 72F0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FIEN	—	—	—	—	—	—	—	FVCT[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	FVCT[7:0]	Fast Interrupt Vector Number	Specify the vector number of an interrupt source to be a fast interrupt.	R/W
b14 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	FIEN	Fast Interrupt Enable	0: Fast interrupt is disabled 1: Fast interrupt is enabled	R/W

The fast interrupt function based on the FIR register setting is applicable only to interrupts to the CPU. It will not affect any transfer request to the DTC.
Before writing to this register, be sure to disable interrupt requests (IERm.IENj bit = 0 (m = 02h to 1Fh, j = 0 to 7)).

FVCT[7:0] Bits (Fast Interrupt Vector Number)

The FVCT[7:0] bits specify the vector number of an interrupt source that uses the fast interrupt function.

FIEN Bit (Fast Interrupt Enable)

This bit enables the fast interrupt.
Setting this bit to 1 makes the interrupt request of the vector number specified by the FVCT[7:0] bits a fast interrupt. When an interrupt request of the vector number specified by the FVCT[7:0] bits is generated and the interrupt request destination is the CPU while the FIEN bit is 1, the interrupt request is output to the CPU as a fast interrupt regardless of the setting of the IPRn register (n = interrupt vector number). When using the fast interrupt for returning from the software standby mode, see section 14.6.2, Return from Software Standby Mode.
If the setting of the IERm.IENj bit has disabled interrupt requests from the interrupt source with the vector number in this register, fast interrupt requests are not output to the CPU.
For settable vector numbers, see Table 14.3, Interrupt Vector Table.
Do not write any reserved vector numbers to the FVCT[7:0] bits.

For details on the fast interrupt, see section 13, Exception Handling, and section 14.4.6, Fast Interrupt.

14.2.5 Software Interrupt Generation Register (SWINTR)

Address(es): ICU.SWINTR 0008 72E0h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SWINT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SWINT	Software Interrupt Generation	This bit is read as 0. Writing 1 issues a software interrupt request. Writing 0 to this bit has no effect.	R/(W) *1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 1 can be written.

SWINT Bit (Software Interrupt Generation)

When 1 is written to the SWINT bit, the interrupt request register 027 (IR027) is set to 1.

If 1 is written to the SWINT bit when the DTC transfer request enable register 027 (DTCER027) is set to 0, an interrupt to the CPU is generated.

If 1 is written to the SWINT bit when the DTC transfer request enable register 027 (DTCER027) is set to 1, a DTC transfer request is issued.

14.2.6 DTC Transfer Request Enable Register n (DTCERn) (n = interrupt vector number)

Address(es): ICU.DTCER027 0008 711Bh to ICU.DTCER255 0008 71FFh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTCE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DTCE	DTC Transfer Request Enable	0: The corresponding interrupt source is not selected as the DTC trigger. 1: The corresponding interrupt source is selected as the DTC trigger.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

See Table 14.3, Interrupt Vector Table, for the interrupt sources that are selectable as the DTC trigger.

DTCE Bit (DTC Transfer Request Enable)

When the DTCE bit is set to 1, the corresponding interrupt source is selected as the DTC trigger.

[Setting condition]

- When 1 is written to the DTCE bit

[Clearing conditions]

- When the specified number of transfers is completed (for the chain transfer, the number of transfers for the last chain transfer is completed)
- When 0 is written to the DTCE bit

14.2.7 IRQ Control Register i (IRQCRi) (i = 0 to 5)

Address(es): ICU.IRQCR0 0008 7500h to ICU.IRQCR5 0008 7505h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	IRQMD[1:0]		—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3, b2	IRQMD[1:0]	IRQ Detection Sense Select	b3 b2 0 0: Low level 0 1: Falling edge 1 0: Rising edge 1 1: Rising and falling edges	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Only change the settings of this register while the corresponding interrupt request enable bit is prohibiting the interrupt request (IENj bit in IERm (m = 02h to 1Fh, j = 0 to 7) is 0). After changing the setting, clear the IR flag in IRn before setting the interrupt enable bit. However, when the change is to the low level, the IR flag does not require clearing.

IRQMD[1:0] Bits (IRQ Detection Sense Select)

These bits select the interrupt detection sensing method of IRQi pin.

For the external pin interrupt detection setting, see section 14.4.8, External Pin Interrupts.

14.2.8 IRQ Pin Digital Filter Enable Register 0 (IRQFLTE0)

Address(es): ICU.IRQFLTE0 0008 7510h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	FLTEN5	FLTEN4	FLTEN3	FLTEN2	FLTEN1	FLTEN0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FLTEN0	IRQ0 Digital Filter Enable	0: Digital filter is disabled 1: Digital filter is enabled	R/W
b1	FLTEN1	IRQ1 Digital Filter Enable		R/W
b2	FLTEN2	IRQ2 Digital Filter Enable		R/W
b3	FLTEN3	IRQ3 Digital Filter Enable		R/W
b4	FLTEN4	IRQ4 Digital Filter Enable		R/W
b5	FLTEN5	IRQ5 Digital Filter Enable		R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FLTENi Bit (IRQi Digital Filter Enable) (i = 0 to 5)

This bit enables the digital filter used for the IRQi pin.

The digital filter is enabled when the FLTENi bit is 1, and disabled when the FLTENi bit is 0.

The IRQi pin level is sampled at the sampling clock cycle specified with the IRQFLTC0.FCLKSELi[1:0] bits. When the sampled level matches three times, the output level from the digital filter changes.

For details of the digital filter, see section 14.4.7, Digital Filter.

14.2.9 IRQ Pin Digital Filter Setting Register 0 (IRQFLTC0)

Address(es): ICU.IRQFLTC0 0008 7514h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	FCLKSEL5[1:0]		FCLKSEL4[1:0]		FCLKSEL3[1:0]		FCLKSEL2[1:0]		FCLKSEL1[1:0]		FCLKSEL0[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	FCLKSEL0[1:0]	IRQ0 Digital Filter Sampling Clock	0 0: PCLK 0 1: PCLK/8 1 0: PCLK/32 1 1: PCLK/64	R/W
b3, b2	FCLKSEL1[1:0]	IRQ1 Digital Filter Sampling Clock		R/W
b5, b4	FCLKSEL2[1:0]	IRQ2 Digital Filter Sampling Clock		R/W
b7, b6	FCLKSEL3[1:0]	IRQ3 Digital Filter Sampling Clock		R/W
b9, b8	FCLKSEL4[1:0]	IRQ4 Digital Filter Sampling Clock		R/W
b11, b10	FCLKSEL5[1:0]	IRQ5 Digital Filter Sampling Clock		R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FCLKSELi[1:0] Bits (IRQi Digital Filter Sampling Clock) (i = 0 to 5)

These bits select the cycle of the digital filter sampling clock for the IRQi pin.

The sampling clock cycle can be selected from among the PCLK (every cycle), PCLK/8 (once every eight cycles), PCLK/32 (once every 32 cycles), and PCLK/64 (once every 64 cycles).

For details of the digital filter, see section 14.4.7, Digital Filter.

14.2.10 Non-Maskable Interrupt Status Register (NMISR)

Address(es): ICU.NMISR 0008 7580h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LVD2ST	LVD1ST	IWDTST	—	OSTST	NMIST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NMIST	NMI Status Flag	0: NMI pin interrupt is not requested 1: NMI pin interrupt is requested	R
b1	OSTST	Oscillation Stop Detection Interrupt Status Flag	0: Oscillation stop detection interrupt is not requested 1: Oscillation stop detection interrupt is requested	R
b2	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R
b3	IWDTST	IWDT Underflow/Refresh Error Status Flag	0: IWDT underflow/refresh error interrupt is not requested 1: IWDT underflow/refresh error interrupt is requested	R
b4	LVD1ST	Voltage Monitoring 1 Interrupt Status Flag	0: Voltage monitoring 1 interrupt is not requested 1: Voltage monitoring 1 interrupt is requested	R
b5	LVD2ST	Voltage Monitoring 2 Interrupt Status Flag	0: Voltage monitoring 2 interrupt is not requested 1: Voltage monitoring 2 interrupt is requested	R
b7, b6	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

The NMISR register monitors the status of a non-maskable interrupt source. Writing to the NMISR register is ignored. The setting in the non-maskable interrupt enable register (NMIER) does not affect the status flags in NMISR.
Before the end of the non-maskable interrupt handler, read the NMISR register and confirm the generation status of other non-maskable interrupts. Be sure to confirm that all of the bits in the NMISR register are set to 0 before the end of the handler.

NMIST Flag (NMI Status Flag)

This flag indicates the NMI pin interrupt request.
The NMIST flag is read-only, and cleared by the NMICLR.NMICLR bit.
[Setting condition]

- When an edge specified by the NMICR.NMIMD bit is input to the NMI pin

[Clearing condition]

- When 1 is written to the NMICLR.NMICLR bit

OSTST Flag (Oscillation Stop Detection Interrupt Status Flag)

This flag indicates the oscillation stop detection interrupt request.
The OSTST flag is read-only, and cleared by the NMICLR.OSTCLR bit.
[Setting condition]

- When the oscillation stop detection interrupt is generated

[Clearing condition]

- When 1 is written to the NMICLR.OSTCLR bit

IWDTST Flag (IWDT Underflow/Refresh Error Status Flag)

This flag indicates the IWDT underflow/refresh error interrupt request.
The IWDTST flag is read-only, and cleared by the NMICLR.IWDTCLR bit.
[Setting condition]

- When the IWDT underflow/refresh error interrupt is generated while this interrupt is enabled at its source.

[Clearing condition]

- When 1 is written to the NMICLR.IWDTCLR bit

LVD1ST Flag (Voltage Monitoring 1 Interrupt Status Flag)

This flag indicates the request for voltage monitoring 1 interrupt.
The LVD1ST flag is read-only, and cleared by the NMICLR.LVD1CLR bit.
[Setting condition]

- When the voltage monitoring 1 interrupt is generated while this interrupt is enabled at its source.

[Clearing condition]

- When 1 is written to the NMICLR.LVD1CLR bit

LVD2ST Flag (Voltage Monitoring 2 Interrupt Status Flag)

This flag indicates the request for voltage monitoring 2 interrupt.
The LVD2ST flag is read-only, and cleared by the NMICLR.LVD2CLR bit.
[Setting condition]

- When the voltage monitoring 2 interrupt is generated while this interrupt is enabled at its source.

[Clearing condition]

- When 1 is written to the NMICLR.LVD2CLR bit

14.2.11 Non-Maskable Interrupt Enable Register (NMIER)

Address(es): ICU.NMIER 0008 7581h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LVD2EN	LVD1EN	IWDTEN	—	OSTEN	NMIEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NMIEN	NMI Pin Interrupt Enable	0: NMI pin interrupt is disabled 1: NMI pin interrupt is enabled	R/(W) *1
b1	OSTEN	Oscillation Stop Detection Interrupt Enable	0: Oscillation stop detection interrupt is disabled 1: Oscillation stop detection interrupt is enabled	R/(W) *1
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	IWDTEN	IWDT Underflow/Refresh Error Enable	0: IWDT underflow/refresh error interrupt is disabled 1: IWDT underflow/refresh error interrupt is enabled	R/(W) *1
b4	LVD1EN	Voltage Monitoring 1 Interrupt Enable	0: Voltage monitoring 1 interrupt is disabled 1: Voltage monitoring 1 interrupt is enabled	R/(W) *1
b5	LVD2EN	Voltage Monitoring 2 Interrupt Enable	0: Voltage monitoring 2 interrupt is disabled 1: Voltage monitoring 2 interrupt is enabled	R/(W) *1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

NMIEN Bit (NMI Pin Interrupt Enable)

This bit enables the NMI pin interrupt.

A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

Writing 0 to this bit is disabled.

OSTEN Bit (Oscillation Stop Detection Interrupt Enable)

This bit enables the oscillation stop detection interrupt.

A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

Writing 0 to this bit is disabled.

IWDTEN Bit (IWDT Underflow/Refresh Error Enable)

This bit enables the IWDT underflow/refresh error interrupt.

A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

Writing 0 to this bit is disabled.

LVD1EN Bit (Voltage Monitoring 1 Interrupt Enable)

This bit enables the voltage monitoring 1 interrupt.

A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

Writing 0 to this bit is disabled.

LVD2EN Bit (Voltage Monitoring 2 Interrupt Enable)

This bit enables the voltage monitoring 2 interrupt.

A 1 can be written to this bit only once, and subsequent write accesses are no longer enabled.

Writing 0 to this bit is disabled.

14.2.12 Non-Maskable Interrupt Status Clear Register (NMICLR)

Address(es): ICU.NMICLR 0008 7582h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LVD2CLR	LVD1CLR	IWDTCLR	—	OSTCLR	NMICLR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NMICLR	NMI Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.NMIST flag. Writing 0 to this bit has no effect.	R/(W) *1
b1	OSTCLR	OST Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.OSTST flag. Writing 0 to this bit has no effect.	R/(W) *1
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	IWDTCLR	IWDT Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.IWDTST flag. Writing 0 to this bit has no effect.	R/(W) *1
b4	LVD1CLR	LVD1 Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.LVD1ST flag. Writing 0 to this bit has no effect.	R/(W) *1
b5	LVD2CLR	LVD2 Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.LVD2ST flag. Writing 0 to this bit has no effect.	R/(W) *1
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 1 can be written to this bit.

NMICLR Bit (NMI Clear)

Writing 1 to the NMICLR bit clears the NMISR.NMIST flag. This bit is read as 0.

OSTCLR Bit (OST Clear)

Writing 1 to the OSTCLR bit clears the NMISR.OSTST flag. This bit is read as 0.

IWDTCLR Bit (IWDT Clear)

Writing 1 to the IWDTCLR bit clears the NMISR.IWDTST flag. This bit is read as 0.

LVD1CLR Bit (LVD1 Clear)

Writing 1 to the LVD1CLR bit clears the NMISR.LVD1ST flag. This bit is read as 0.

LVD2CLR Bit (LVD2 Clear)

Writing 1 to the LVD2CLR bit clears the NMISR.LVD2ST flag. This bit is read as 0.

14.2.13 NMI Pin Interrupt Control Register (NMICR)

Address(es): ICU.NMICR 0008 7583h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	NMIMD	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	NMIMD	NMI Detection Set	0: Falling edge 1: Rising edge	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Change the setting of the NMICR register before the NMI pin interrupt is enabled (before setting the NMIER.NMIEN bit to 1).

NMIMD Bit (NMI Detection Set)

This bit specifies the detection edge of the NMI pin interrupt.

14.2.14 NMI Pin Digital Filter Enable Register (NMIFLTE)

Address(es): ICU.NMIFLTE 0008 7590h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	NFLTEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	NFLTEN	NMI Digital Filter Enable	0: Digital filter is disabled 1: Digital filter is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

NFLTEN Bit (NMI Digital Filter Enable)

This bit enables the digital filter used for the NMI pin interrupt.

The digital filter is enabled when the NFLTEN bit is 1, and disabled when the NFLTEN bit is 0.

The NMI pin level is sampled at the sampling clock cycle specified with the NMIFLTC.NFCLKSEL[1:0] bits. When the sampled level matches three times, the output level from the digital filter changes.

For details of the digital filter, see section 14.4.7, Digital Filter.

14.2.15 NMI Pin Digital Filter Setting Register (NMIFLTC)

Address(es): ICU.NMIFLTC 0008 7594h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	NFCLKSEL[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	NFCLKSEL[1:0]	NMI Digital Filter Sampling Clock	b1 b0 0 0: PCLK 0 1: PCLK/8 1 0: PCLK/32 1 1: PCLK/64	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

NFCLKSEL[1:0] Bits (NMI Digital Filter Sampling Clock)

These bits select the cycle of the digital filter sampling clock for the NMI pin interrupt.

The sampling clock cycle can be selected from among the PCLK (every cycle), PCLK/8 (once every eight cycles), PCLK/32 (once every 32 cycles), and PCLK/64 (once every 64 cycles).

For details of the digital filter, see section 14.4.7, Digital Filter.

14.3 Vector Table

There are two types of interrupts detected by the interrupt controller: maskable interrupts and non-maskable interrupts. When the CPU accepts an interrupt or non-maskable interrupt, it acquires a 4-byte vector address from the vector table.

14.3.1 Interrupt Vector Table

The interrupt vector table is placed in the 1024-byte range (4 bytes × 256 sources) beginning at the address specified in the interrupt table register (INTB) of the CPU. Write a value to the INTB register before enabling interrupts. The value written to the INTB register should be a multiple of 4.
Executing an INT instruction or BRK instruction leads to the generation of an unconditional trap. The same range of memory as shown in Table 14.3, Interrupt Vector Table, is used for the vectors for unconditional traps. The vector for BRK instructions is vector 0 while the vector numbers for INT instructions are specifiable as numbers in the range from 0 to 255.
Table 14.3 lists details of the interrupt vectors. Details of the headings in Table 14.3 are listed below.

Item	Description
Source of interrupt request generation	Name of the source for generation of the interrupt request
Name	Name of the interrupt
Vector no.	Vector number for the interrupt
Vector address offset	Value of the offset from the base address for the vector table
Interrupt detection method	“Edge” or “level” as the method for detection of the interrupt
CPU interrupt	“✓” in this column indicates usability as a CPU interrupt.
Start DTC	“✓” in this column indicates usability as a request for DTC transfer.
SSBY return	“✓” in this column indicates usability as a request for return from software-standby mode.
IER	Name of the IER register and bit corresponding to the vector number
IPR	Name of the IPR register corresponding to the interrupt source
DTCER	Name of the DTCER register corresponding to the DTC trigger

Table 14.3 Interrupt Vector Table (1/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start DTC	SSBY Return	IER	IPR	DTCER
—	For an unconditional trap	0	0000h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	1	0004h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	2	0008h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	3	000Ch	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	4	0010h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	5	0014h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	6	0018h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	7	001Ch	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	8	0020h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	9	0024h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	10	0028h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	11	002Ch	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	12	0030h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	13	0034h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	14	0038h	—	N/A	N/A	N/A	—	—	—
—	For an unconditional trap	15	003Ch	—	N/A	N/A	N/A	—	—	—
BSC	BUSERR	16	0040h	Level	✓	N/A	N/A	IER02.IEN0	IPR000	—
—	Reserved	17	0044h	—	N/A	N/A	N/A	—	—	—
—	Reserved	18	0048h	—	N/A	N/A	N/A	—	—	—
—	Reserved	19	004Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	20	0050h	—	N/A	N/A	N/A	—	—	—
—	Reserved	21	0054h	—	N/A	N/A	N/A	—	—	—
—	Reserved	22	0058h	—	N/A	N/A	N/A	—	—	—
FCU	FRDYI	23	005Ch	Edge	✓	N/A	N/A	IER02.IEN7	IPR002	—
—	Reserved	24	0060h	—	N/A	N/A	N/A	—	—	—
—	Reserved	25	0064h	—	N/A	N/A	N/A	—	—	—
—	Reserved	26	0068h	—	N/A	N/A	N/A	—	—	—
ICU	SWINT	27	006Ch	Edge	✓	✓	N/A	IER03.IEN3	IPR003	DTCER027
CMT0	CMI0	28	0070h	Edge	✓	✓	N/A	IER03.IEN4	IPR004	DTCER028
CMT1	CMI1	29	0074h	Edge	✓	✓	N/A	IER03.IEN5	IPR005	DTCER029
—	Reserved	30	0078h	—	N/A	N/A	N/A	—	—	—
—	Reserved	31	007Ch	—	N/A	N/A	N/A	—	—	—
CAC	FERRF	32	0080h	Level	✓	N/A	N/A	IER04.IEN0	IPR032	—
	MENDF	33	0084h	Level	✓	N/A	N/A	IER04.IEN1	IPR033	—
	OVFF	34	0088h	Level	✓	N/A	N/A	IER04.IEN2	IPR034	—
—	Reserved	35	008Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	36	0090h	—	N/A	N/A	N/A	—	—	—
—	Reserved	37	0094h	—	N/A	N/A	N/A	—	—	—
—	Reserved	38	0098h	—	N/A	N/A	N/A	—	—	—
—	Reserved	39	009Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	40	00A0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	41	00A4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	42	00A8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	43	00ACh	—	N/A	N/A	N/A	—	—	—
—	Reserved	44	00B0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	45	00B4h	—	N/A	N/A	N/A	—	—	—

Table 14.3 Interrupt Vector Table (2/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start DTC	SSBY Return	IER	IPR	DTCER
—	Reserved	46	00B8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	47	00BCh	—	N/A	N/A	N/A	—	—	—
—	Reserved	48	00C0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	49	00C4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	50	00C8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	51	00CCh	—	N/A	N/A	N/A	—	—	—
—	Reserved	52	00D0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	53	00D4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	54	00D8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	55	00DCh	—	N/A	N/A	N/A	—	—	—
—	Reserved	56	00E0h	—	N/A	N/A	N/A	—	—	—
DOC	DOPCF	57	00E4h	Level	✓	N/A	N/A	IER07.IEN1	IPR057	—
—	Reserved	58	00E8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	59	00ECh	—	N/A	N/A	N/A	—	—	—
—	Reserved	60	00F0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	61	00F4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	62	00F8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	63	00FCh	—	N/A	N/A	N/A	—	—	—
ICU	IRQ0	64	0100h	Edge/Level	✓	✓	✓	IER08.IEN0	IPR064	DTCER064
	IRQ1	65	0104h	Edge/Level	✓	✓	✓	IER08.IEN1	IPR065	DTCER065
	IRQ2	66	0108h	Edge/Level	✓	✓	✓	IER08.IEN2	IPR066	DTCER066
	IRQ3	67	010Ch	Edge/Level	✓	✓	✓	IER08.IEN3	IPR067	DTCER067
	IRQ4	68	0110h	Edge/Level	✓	✓	✓	IER08.IEN4	IPR068	DTCER068
	IRQ5	69	0114h	Edge/Level	✓	✓	✓	IER08.IEN5	IPR069	DTCER069
—	Reserved	70	0118h	—	N/A	N/A	N/A	—	—	—
—	Reserved	71	011Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	72	0120h	—	N/A	N/A	N/A	—	—	—
—	Reserved	73	0124h	—	N/A	N/A	N/A	—	—	—
—	Reserved	74	0128h	—	N/A	N/A	N/A	—	—	—
—	Reserved	75	012Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	76	0130h	—	N/A	N/A	N/A	—	—	—
—	Reserved	77	0134h	—	N/A	N/A	N/A	—	—	—
—	Reserved	78	0138h	—	N/A	N/A	N/A	—	—	—
—	Reserved	79	013Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	80	0140h	—	N/A	N/A	N/A	—	—	—
—	Reserved	81	0144h	—	N/A	N/A	N/A	—	—	—
—	Reserved	82	0148h	—	N/A	N/A	N/A	—	—	—
—	Reserved	83	014Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	84	0150h	—	N/A	N/A	N/A	—	—	—
—	Reserved	85	0154h	—	N/A	N/A	N/A	—	—	—
—	Reserved	86	0158h	—	N/A	N/A	N/A	—	—	—
—	Reserved	87	015Ch	—	N/A	N/A	N/A	—	—	—
LVD	LVD1	88	0160h	Edge	✓	N/A	✓	IER0B.IEN0	IPR088	—
	LVD2	89	0164h	Edge	✓	N/A	✓	IER0B.IEN1	IPR089	—
—	Reserved	90	0168h	—	N/A	N/A	N/A	—	—	—
—	Reserved	91	016Ch	—	N/A	N/A	N/A	—	—	—

Table 14.3 Interrupt Vector Table (3/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start DTC	SSBY Return	IER	IPR	DTCER
—	Reserved	92	0170h	—	N/A	N/A	N/A	—	—	—
—	Reserved	93	0174h	—	N/A	N/A	N/A	—	—	—
—	Reserved	94	0178h	—	N/A	N/A	N/A	—	—	—
—	Reserved	95	017Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	96	0180h	—	N/A	N/A	N/A	—	—	—
—	Reserved	97	0184h	—	N/A	N/A	N/A	—	—	—
—	Reserved	98	0188h	—	N/A	N/A	N/A	—	—	—
—	Reserved	99	018Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	100	0190h	—	N/A	N/A	N/A	—	—	—
—	Reserved	101	0194h	—	N/A	N/A	N/A	—	—	—
S12AD	S12ADI	102	0198h	Edge	✓	✓	N/A	IER0C.IEN6	IPR102	DTCER102
	GBADI	103	019Ch	Edge	✓	✓	N/A	IER0C.IEN7	IPR103	DTCER103
	GCADI	104	01A0h	Edge	✓	✓	N/A	IER0D.IEN0	IPR104	DTCER104
—	Reserved	105	01A4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	106	01A8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	107	01ACh	—	N/A	N/A	N/A	—	—	—
CMPC0	CMPC0	108	01B0h	Edge	✓	✓	N/A	IER0D.IEN4	IPR108	DTCER108
CMPC1	CMPC1	109	01B4h	Edge	✓	✓	N/A	IER0D.IEN5	IPR109	DTCER109
CMPC2	CMPC2	110	01B8h	Edge	✓	✓	N/A	IER0D.IEN6	IPR110	DTCER110
—	Reserved	111	01BCh	—	N/A	N/A	N/A	—	—	—
—	Reserved	112	01C0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	113	01C4h	—	N/A	N/A	N/A	—	—	—
MTU0	TGIA0	114	01C8h	Edge	✓	✓	N/A	IER0E.IEN2	IPR114	DTCER114
	TGIB0	115	01CCh	Edge	✓	✓	N/A	IER0E.IEN3		DTCER115
	TGIC0	116	01D0h	Edge	✓	✓	N/A	IER0E.IEN4		DTCER116
	TGID0	117	01D4h	Edge	✓	✓	N/A	IER0E.IEN5		DTCER117
	TCIV0	118	01D8h	Edge	✓	N/A	N/A	IER0E.IEN6	IPR118	—
	TGIE0	119	01DCh	Edge	✓	N/A	N/A	IER0E.IEN7		—
	TGIF0	120	01E0h	Edge	✓	N/A	N/A	IER0F.IEN0		—
MTU1	TGIA1	121	01E4h	Edge	✓	✓	N/A	IER0F.IEN1	IPR121	DTCER121
	TGIB1	122	01E8h	Edge	✓	✓	N/A	IER0F.IEN2		DTCER122
	TCIV1	123	01ECh	Edge	✓	N/A	N/A	IER0F.IEN3	IPR123	—
	TCIU1	124	01F0h	Edge	✓	N/A	N/A	IER0F.IEN4		—
MTU2	TGIA2	125	01F4h	Edge	✓	✓	N/A	IER0F.IEN5	IPR125	DTCER125
	TGIB2	126	01F8h	Edge	✓	✓	N/A	IER0F.IEN6		DTCER126
	TCIV2	127	01FCh	Edge	✓	N/A	N/A	IER0F.IEN7	IPR127	—
	TCIU2	128	0200h	Edge	✓	N/A	N/A	IER10.IEN0		—
MTU3	TGIA3	129	0204h	Edge	✓	✓	N/A	IER10.IEN1	IPR129	DTCER129
	TGIB3	130	0208h	Edge	✓	✓	N/A	IER10.IEN2		DTCER130
	TGIC3	131	020Ch	Edge	✓	✓	N/A	IER10.IEN3		DTCER131
	TGID3	132	0210h	Edge	✓	✓	N/A	IER10.IEN4		DTCER132
	TCIV3	133	0214h	Edge	✓	N/A	N/A	IER10.IEN5	IPR133	—

Table 14.3 Interrupt Vector Table (4/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start DTC	SSBY Return	IER	IPR	DTCER
MTU4	TGIA4	134	0218h	Edge	✓	✓	N/A	IER10.IEN6	IPR134	DTCER134
	TGIB4	135	021Ch	Edge	✓	✓	N/A	IER10.IEN7		DTCER135
	TGIC4	136	0220h	Edge	✓	✓	N/A	IER11.IEN0		DTCER136
	TGID4	137	0224h	Edge	✓	✓	N/A	IER11.IEN1		DTCER137
	TCIV4	138	0228h	Edge	✓	✓	N/A	IER11.IEN2	IPR138	DTCER138
MTU5	TGIU5	139	022Ch	Edge	✓	✓	N/A	IER11.IEN3	IPR139	DTCER139
	TGIV5	140	0230h	Edge	✓	✓	N/A	IER11.IEN4		DTCER140
	TGIW5	141	0234h	Edge	✓	✓	N/A	IER11.IEN5		DTCER141
—	Reserved	142	0238h	—	N/A	N/A	N/A	—	—	—
—	Reserved	143	023Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	144	0240h	—	N/A	N/A	N/A	—	—	—
—	Reserved	145	0244h	—	N/A	N/A	N/A	—	—	—
—	Reserved	146	0248h	—	N/A	N/A	N/A	—	—	—
—	Reserved	147	024Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	148	0250h	—	N/A	N/A	N/A	—	—	—
—	Reserved	149	0254h	—	N/A	N/A	N/A	—	—	—
—	Reserved	150	0258h	—	N/A	N/A	N/A	—	—	—
—	Reserved	151	025Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	152	0260h	—	N/A	N/A	N/A	—	—	—
—	Reserved	153	0264h	—	N/A	N/A	N/A	—	—	—
—	Reserved	154	0268h	—	N/A	N/A	N/A	—	—	—
—	Reserved	155	026Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	156	0270h	—	N/A	N/A	N/A	—	—	—
—	Reserved	157	0274h	—	N/A	N/A	N/A	—	—	—
—	Reserved	158	0278h	—	N/A	N/A	N/A	—	—	—
—	Reserved	159	027Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	160	0280h	—	N/A	N/A	N/A	—	—	—
—	Reserved	161	0284h	—	N/A	N/A	N/A	—	—	—
—	Reserved	162	0288h	—	N/A	N/A	N/A	—	—	—
—	Reserved	163	028Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	164	0290h	—	N/A	N/A	N/A	—	—	—
—	Reserved	165	0294h	—	N/A	N/A	N/A	—	—	—
—	Reserved	166	0298h	—	N/A	N/A	N/A	—	—	—
—	Reserved	167	029Ch	—	N/A	N/A	N/A	—	—	—
POE	OEI1	168	02A0h	Level	✓	N/A	N/A	IER15.IEN0	IPR168	—
—	Reserved	169	02A4h	—	N/A	N/A	N/A	—	—	—
POE	OEI3	170	02A8h	Level	✓	N/A	N/A	IER15.IEN2	IPR168	—
	OEI4	171	02ACh	Level	✓	N/A	N/A	IER15.IEN3		—
—	Reserved	172	02B0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	173	02B4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	174	02B8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	175	02BCh	—	N/A	N/A	N/A	—	—	—
—	Reserved	176	02C0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	177	02C4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	178	02C8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	179	02CCh	—	N/A	N/A	N/A	—	—	—

Table 14.3 Interrupt Vector Table (5/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start DTC	SSBY Return	IER	IPR	DTCER
—	Reserved	180	02D0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	181	02D4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	182	02D8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	183	02DCh	—	N/A	N/A	N/A	—	—	—
—	Reserved	184	02E0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	185	02E4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	186	02E8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	187	02ECh	—	N/A	N/A	N/A	—	—	—
—	Reserved	188	02F0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	189	02F4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	190	02F8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	191	02FCh	—	N/A	N/A	N/A	—	—	—
—	Reserved	192	0300h	—	N/A	N/A	N/A	—	—	—
—	Reserved	193	0304h	—	N/A	N/A	N/A	—	—	—
—	Reserved	194	0308h	—	N/A	N/A	N/A	—	—	—
—	Reserved	195	030Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	196	0310h	—	N/A	N/A	N/A	—	—	—
—	Reserved	197	0314h	—	N/A	N/A	N/A	—	—	—
—	Reserved	198	0318h	—	N/A	N/A	N/A	—	—	—
—	Reserved	199	031Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	200	0320h	—	N/A	N/A	N/A	—	—	—
—	Reserved	201	0324h	—	N/A	N/A	N/A	—	—	—
—	Reserved	202	0328h	—	N/A	N/A	N/A	—	—	—
—	Reserved	203	032Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	204	0330h	—	N/A	N/A	N/A	—	—	—
—	Reserved	205	0334h	—	N/A	N/A	N/A	—	—	—
—	Reserved	206	0338h	—	N/A	N/A	N/A	—	—	—
—	Reserved	207	033Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	208	0340h	—	N/A	N/A	N/A	—	—	—
—	Reserved	209	0344h	—	N/A	N/A	N/A	—	—	—
—	Reserved	210	0348h	—	N/A	N/A	N/A	—	—	—
—	Reserved	211	034Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	212	0350h	—	N/A	N/A	N/A	—	—	—
—	Reserved	213	0354h	—	N/A	N/A	N/A	—	—	—
—	Reserved	214	0358h	—	N/A	N/A	N/A	—	—	—
—	Reserved	215	035Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	216	0360h	—	N/A	N/A	N/A	—	—	—
—	Reserved	217	0364h	—	N/A	N/A	N/A	—	—	—
SCI1	ERI1	218	0368h	Level	✓	N/A	N/A	IER1B.IEN2	IPR218	—
	RXI1	219	036Ch	Edge	✓	✓	N/A	IER1B.IEN3		DTCER219
	TXI1	220	0370h	Edge	✓	✓	N/A	IER1B.IEN4		DTCER220
	TEI1	221	0374h	Level	✓	N/A	N/A	IER1B.IEN5		—
SCI5	ERI5	222	0378h	Level	✓	N/A	N/A	IER1B.IEN6	IPR222	—
	RXI5	223	037Ch	Edge	✓	✓	N/A	IER1B.IEN7		DTCER223
	TXI5	224	0380h	Edge	✓	✓	N/A	IER1C.IEN0		DTCER224
	TEI5	225	0384h	Level	✓	N/A	N/A	IER1C.IEN1		—

Table 14.3 Interrupt Vector Table (6/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Interrupt Detection Method	CPU Interrupt	Start DTC	SSBY Return	IER	IPR	DTCER
—	Reserved	226	0388h	—	N/A	N/A	N/A	—	—	—
—	Reserved	227	038Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	228	0390h	—	N/A	N/A	N/A	—	—	—
—	Reserved	229	0394h	—	N/A	N/A	N/A	—	—	—
—	Reserved	230	0398h	—	N/A	N/A	N/A	—	—	—
—	Reserved	231	039Ch	—	N/A	N/A	N/A	—	—	—
—	Reserved	232	03A0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	233	03A4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	234	03A8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	235	03ACh	—	N/A	N/A	N/A	—	—	—
—	Reserved	236	03B0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	237	03B4h	—	N/A	N/A	N/A	—	—	—
SCI12	ERI12	238	03B8h	Level	✓	N/A	N/A	IER1D.IEN6	IPR238	—
	RXI12	239	03BCh	Edge	✓	✓	N/A	IER1D.IEN7		DTCER239
	TXI12	240	03C0h	Edge	✓	✓	N/A	IER1E.IEN0		DTCER240
	TEI12	241	03C4h	Level	✓	N/A	N/A	IER1E.IEN1		—
	SCIX0	242	03C8h	Level	✓	N/A	N/A	IER1E.IEN2	IPR242	—
	SCIX1	243	03CCh	Level	✓	N/A	N/A	IER1E.IEN3	IPR243	—
	SCIX2	244	03D0h	Level	✓	N/A	N/A	IER1E.IEN4	IPR244	—
	SCIX3	245	03D4h	Level	✓	N/A	N/A	IER1E.IEN5	IPR245	—
RIIC0	EEI0	246	03D8h	Level	✓	N/A	N/A	IER1E.IEN6	IPR246	—
	RXI0	247	03DCh	Edge	✓	✓	N/A	IER1E.IEN7	IPR247	DTCER247
	TXI0	248	03E0h	Edge	✓	✓	N/A	IER1F.IEN0	IPR248	DTCER248
	TEI0	249	03E4h	Level	✓	N/A	N/A	IER1F.IEN1	IPR249	—
—	Reserved	250	03E8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	251	03ECh	—	N/A	N/A	N/A	—	—	—
—	Reserved	252	03F0h	—	N/A	N/A	N/A	—	—	—
—	Reserved	253	03F4h	—	N/A	N/A	N/A	—	—	—
—	Reserved	254	03F8h	—	N/A	N/A	N/A	—	—	—
—	Reserved	255	03FCh	—	N/A	N/A	N/A	—	—	—

Note 1. An interrupt source with a smaller vector number takes precedence.

14.3.2 Fast Interrupt Vector Table

The address of the entry in the interrupt vector table that corresponds to the vector number of the fast interrupt is placed in the fast interrupt vector register (FINTV) of the CPU.

14.3.3 Non-maskable Interrupt Vector Table

The non-maskable interrupt vector table is at FFFF FFF8h.

14.4 Interrupt Operation

The interrupt controller performs the following processing.

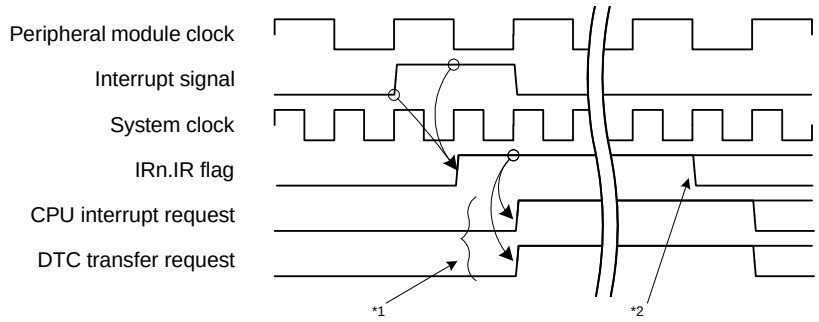
- Detecting interrupts
- Enabling and disabling interrupts
- Selecting interrupt request destinations (CPU interrupt or DTC trigger)
- Determining priority

14.4.1 Detecting Interrupts

Interrupt requests are detected in either of two ways: the detection of edges of the interrupt signal or the detection of a level of the interrupt signal.
Edge detection or level detection is selected for the IRQi pins (i = 0 to 5) as external interrupt requests by the setting of the IRQMD[1:0] bits in IRQCRi.
For interrupts from peripheral modules, either edge detection or level detection is determined per interrupt source.
For the correspondence between interrupt sources and methods of detection, see Table 14.3, Interrupt Vector Table.

14.4.1.1 Operation of Status Flags for Edge-Detected Interrupts

Figure 14.2 shows the operation of the IR flag in IRn (n = interrupt vector number) in the case of edge detection of an interrupt from a peripheral module or on an external pin.
The IR flag in IRn is set to 1 immediately after the transition of the interrupt signal due to generation of the interrupt. If the CPU is the request destination for the interrupt, the IR flag is automatically cleared to 0 on acceptance of the interrupt. If the DTC is the request destination for the interrupt, the IRn.IR flag operation differs according to the DTC transfer settings and transfer count. For details, see Table 14.4, Operation When Starting the DTC.



Note 1. One of the following requests is issued: CPU interrupt request, DTC transfer request. For details of the setting, see section 14.4.3, Selecting Interrupt Request Destinations.
Note 2. When the CPU interrupt request is specified, this flag is set to 0 on acceptance of a CPU interrupt. For the timing of 0 setting at the DTC transfer request, see Table 14.4, Operation When Starting the DTC.

Figure 14.2 IRn.IR Flag Operation for Edge Detection Interrupts

Figure 14.3 to Figure 14.5 show the interrupt signals of the interrupt controller. Note that the timings of the interrupts with interrupt vector numbers 64 to 95 are different from those of other interrupts. For the IRQ pin interrupts with interrupt vector numbers 64 to 79, “internal delay + 2 PCLK cycles” of delay is added after the IRQ pin input. For the interrupts with interrupt vector numbers 80 to 95, “2 PCLK cycles” of delay is added.

If an interrupt signal is generated every clock cycle, the subsequent interrupts cannot be detected; secure two or more clock cycles of the system clock between issuance of continuous interrupt requests.

While the IRn.IR flag is 1 after an interrupt request is generated, the interrupt request that is generated again will be ignored.*1
Figure 14.3 shows the timing for IRn.IR flag re-setting.

Note 1. When the transmission or reception interrupt of the SCI or RIIC is generated with the IRn.IR flag being 1, the interrupt request is retained. After the IRn.IR flag is cleared to 0, the IRn.IR flag is set to 1 again by the retained request. For details, see descriptions of the interrupts in section 23, Serial Communications Interface (SCIg, SCIh) and section 24, I2C-bus Interface (RIICa).

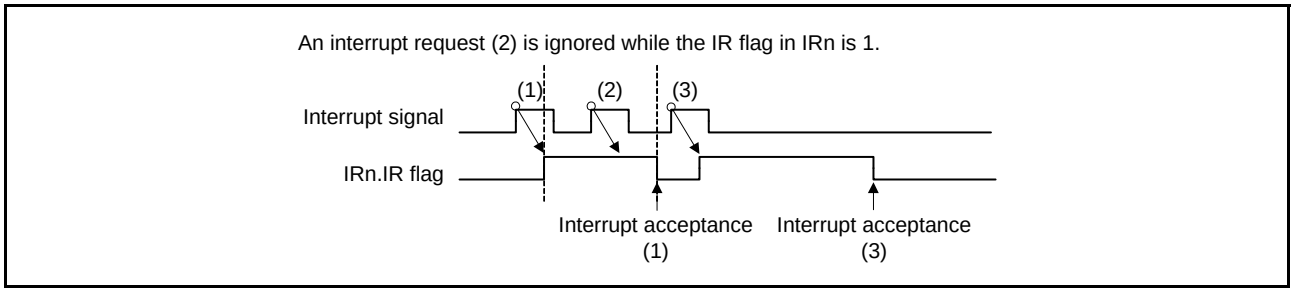


Figure 14.3 Timing for IRn.IR Flag Re-Setting

If an interrupt is disabled after the IRn.IR flag is set to 1 (output of the interrupt request is disabled by the interrupt enable bit of the relevant peripheral module), the IRn.IR flag is not affected but retains its state. Figure 14.4 shows operation when the interrupt is disabled.

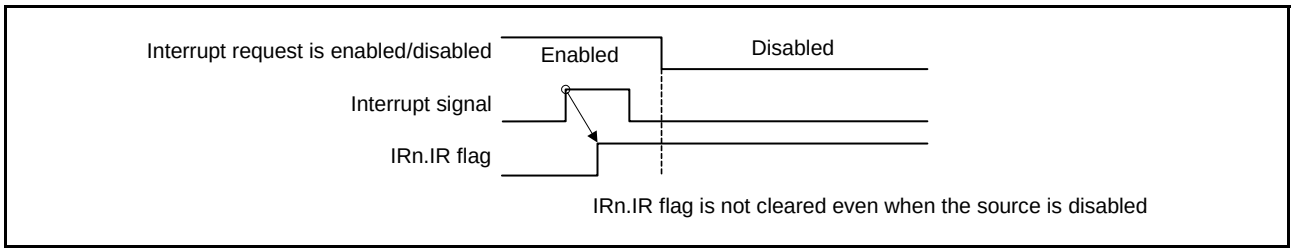


Figure 14.4 Relationship between IRn.IR Flag Operation and Disabling of Interrupt Request

14.4.1.2 Operation of Status Flags for Level-Detected Interrupts

Figure 14.5 shows the operation of the interrupt status flag (IR flag) in IRn (n = interrupt vector number) in the case of level detection of an interrupt from a peripheral module or an external pin.
The IR flag in IRn remains set to 1 as long as the interrupt signal is asserted. To clear the IRn.IR flag to 0, clear the interrupt request in the source generating the interrupt. Confirm that the interrupt request flag in the source generating the interrupt has been cleared to 0 and that the IRn.IR flag has been cleared to 0, and then complete the interrupt handling.

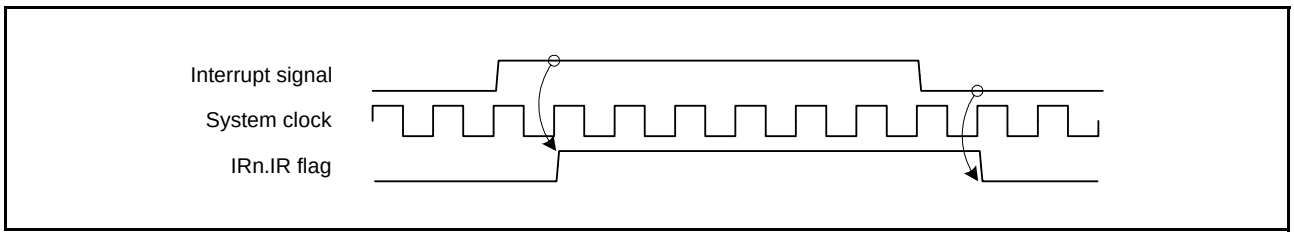


Figure 14.5 IRn.IR Flag Operation for Level Detection Interrupts

Figure 14.6 shows the procedure for handling level detection interrupts.

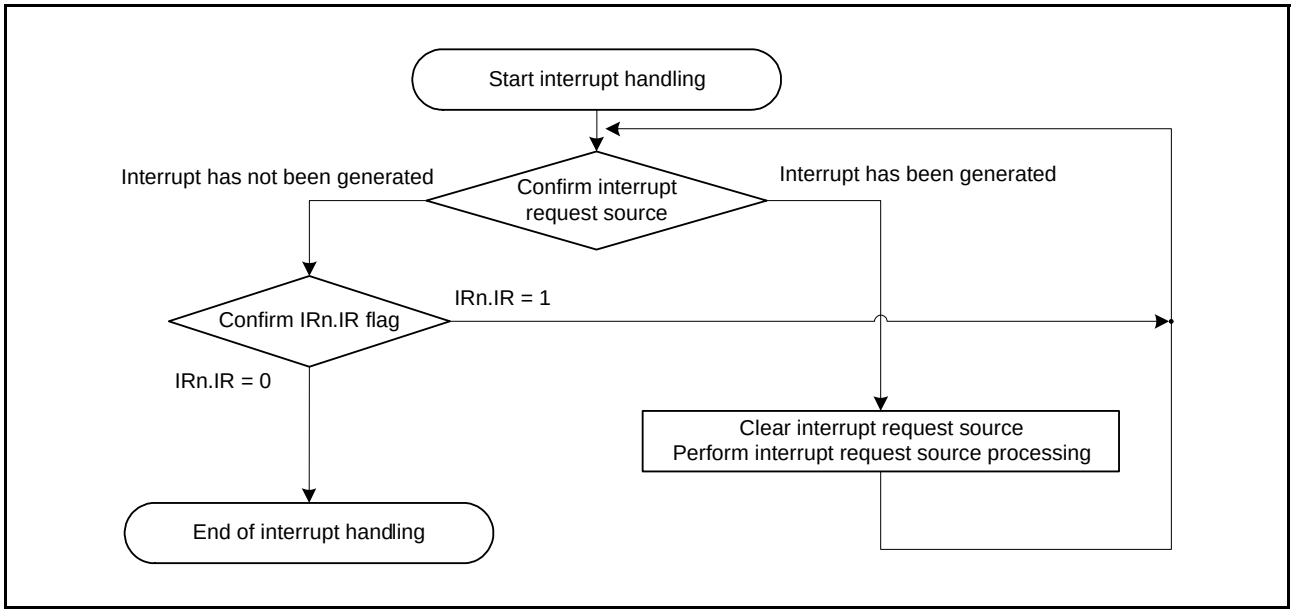


Figure 14.6 Procedure for Handling Level Detection Interrupts

14.4.2 Enabling and Disabling Interrupt Sources

Enabling requests from a given interrupt source requires the following settings.

1. In the case of interrupt requests from peripheral modules, setting the interrupt enable bit for the peripheral module to permit the output of interrupt requests from the source
2. Enabling of the interrupt by the IERm.IENj bit (m = 02h to 1Fh, j = 0 to 7)

When an interrupt request that is enabled at the corresponding source is generated, the corresponding IRn.IR flag (n = interrupt vector number) is set to 1.
Setting the IERm.IENj bit to enable an interrupt request allows the interrupt request for which the corresponding IRn.IR is 1 to be output to the interrupt request destination. Setting the IERm.IENj bit to disable an interrupt request suspends the output of the interrupt request for which the corresponding IRn.IR is 1.
The IRn.IR flag is not affected by the IERm.IENj bit.

Use the following procedure to disable interrupt requests.

1. Set the IERm.IENj bit to disable interrupt requests.
2. Set the peripheral module interrupt output enable bit to disable the output. Read the last written register and confirm that writing is completed.
3. Check the IRn.IR flag, and clear the IRn.IR flag if necessary.*1

Note 1. To disable the transmission or reception interrupt of the SCI or RIIC from the enabled state, clear the IRn.IR flag to 0 using the above procedure. For details, see descriptions of the interrupts in section 23, Serial Communications Interface (SCIg, SCIh) and section 24, I^2C-bus Interface (RIICa).

14.4.3 Selecting Interrupt Request Destinations

Possible settings for the request destination of each interrupt are fixed. That is, settings for request destination other than those indicated in Table 14.3, Interrupt Vector Table, are not possible. Do not make an interrupt request destination setting that is not indicated by a "✓" in Table 14.3.
If the DTC is selected as the destination for requests from an IRQi pin (i = 0 to 5), be sure to set the IRQMD[1:0] bits in IRQCRi for that interrupt to select edge detection.
The following describes how to specify the destinations of interrupt requests.

(1) DTC Trigger

Make the following settings for each source while the IERm.IENj bit (m = 02h to 1Fh, j = 0 to 7) is 0.

1. Set the DTC transfer request enable bit in the DTC transfer request enable register (DTCERn.DTCE (n = interrupt vector number)) for the pertinent source to 1.

After making the above settings, set the IERm.IENj bit to 1.
In addition, set the DTC module start bit (DTCST.DTCST) to 1. The order of making settings for each interrupt and enabling the DTC module start bit does not matter.
For the DTC setting procedure, refer to section 16.5, DTC Setting Procedure, in section 16, Data Transfer Controller (DTCb).

(2) CPU Interrupt Request

If the interrupt request destination is the DTC, the interrupt request is sent to the CPU. Set the IERm.IENj bit (m = 02h to 1Fh, j = 0 to 7) to 1 while the DTC trigger settings described above are in place.

Table 14.4 shows operation when the DTC is the request destination.

Table 14.4 Operation When Starting the DTC

Interrupt Request Destination	DISEL *1	Remaining Number of Transfer Operations	Operation per Request	IR*2	Interrupt Request Destination after Transfer
DTC*3	1	≠ 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	DTC
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	The DTCERn.DTCE bit is cleared and the CPU becomes the destination.
	0	≠ 0	DTC transfer	Cleared at the start of DTC data transfer after reading DTC transfer information	DTC
		= 0	DTC transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	The DTCERn.DTCE bit is cleared and the CPU becomes the destination.

Note 1. DISEL for the DTC is set by the DTC.MRB.DISEL bit.
Note 2. When the IRn.IR flag is 1, an interrupt request (DTC transfer request) that is generated again will be ignored.
Note 3. For chain transfer, DTC transfer continues until the last chain transfer ends. Whether a CPU interrupt is generated at the end of chain transfer, the IRn.IR flag clear timing, and the interrupt request destination after transfer are determined by the state of DISEL and the remaining transfer count at the end of chain transfer. For the chain transfer, see Table 16.4, Chain Transfer Conditions in section 16, Data Transfer Controller (DTCb).

The request destination for an interrupt should be changed while the IERm.IENj bit is 0.

When a source is to be changed to an interrupt request or the DTC transfer information is to be changed while a transfer is not complete (i.e. while the DTCERn.DTCE bit (n = interrupt vector number) has not been cleared) after the settings described under (1) DTC Trigger have been made, follow the procedure below.

1. For both the source to be withdrawn and the source that will have a new trigger, clear the IENj bits in IERm to 0.
2. Check the state of transfer by the DTC. If transfer is in progress, wait for its completion.
3. Make the settings described under (1) DTC Trigger.

14.4.4 Determining Priority

Interrupt priority is determined for each interrupt request destination.
The priority for each interrupt request destination is determined as follows.

(1) Determining Priority when the CPU is the Request Destination of the Interrupt

A source selected for the fast interrupt has the highest priority. After that, an interrupt source with a larger value of the interrupt priority level select bits (IPR[3:0]) in IPRn takes priority. If interrupts with the same priority level are generated by multiple sources, the source with the smallest vector number takes precedence.

(2) Determining Priority when the DTC is the Request Destination of the Interrupt

The IPR[3:0] bits in IPRn (n = interrupt vector number) have no effect. An interrupt source with a smaller vector number takes precedence.

14.4.5 Multiple Interrupts

To enable multiple interrupts of the CPU, set the PSW.I bit to 1 (interrupt enabled) in the handling routine of accepted interrupts.
The PSW.IPL[3:0] bits immediately after processing branches to the interrupt handling routine are set to the same value as the interrupt priority level of the accepted interrupt request. If an interrupt request which has an interrupt level higher than that of the PSW.IPL[3:0] bits is generated at this time, this interrupt request (for multiple interrupts) is accepted.
If the interrupt priority level of the accepted interrupt request is 15 (fast interrupt or interrupt when IPR[3:0] are set to 1111b), multiple interrupts are not generated.

14.4.6 Fast Interrupt

The fast interrupt is an interrupt for executing a faster interrupt response by the CPU, so only one of the interrupt sources can be assigned.
The interrupt priority level of the fast interrupt is 15 (highest) regardless of the setting of the IPR[3:0] bits in IPRn (n = interrupt vector number). In addition, the fast interrupt is accepted with precedence over other interrupt sources with level 15. However, when the value of the PSW.IPL[3:0] bits are 1111b (priority level 15), even the fast interrupt cannot be accepted.
To assign an interrupt source to the fast interrupt, specify the vector number of the source in the FIR.FVCT[7:0] bits, and set the FIR.FIEN bit to 1 (fast interrupt is enabled).
For details on the fast interrupt, see section 2, CPU and section 13, Exception Handling.

14.4.7 Digital Filter

The digital filter function is provided for the external interrupt request IRQi pins (i = 0 to 5) and NMI pin interrupt.
The digital filter samples input signals at the filter sampling clock (PCLK) and removes the pulses of which length is less than three sampling cycles.
To use the digital filter for the IRQi pin, set the sampling clock cycle (PCLK, PCLK/8, PCLK/32, or PCLK/64) with the IRQFLTC0.FCLKSELi[1:0] bits and set the IRQFLTE0.FLTENi bit to 1 (digital filter enabled).
To use the digital filter for the NMI pin, set the sampling clock cycle (PCLK, PCLK/8, PCLK/32, or PCLK/64) with the NMIFLTC.NFCLKSEL[1:0] bits and set the NMIFLTE.NFLTEN bit to 1 (digital filter enabled).
Figure 14.7 shows an example of digital filter operation.

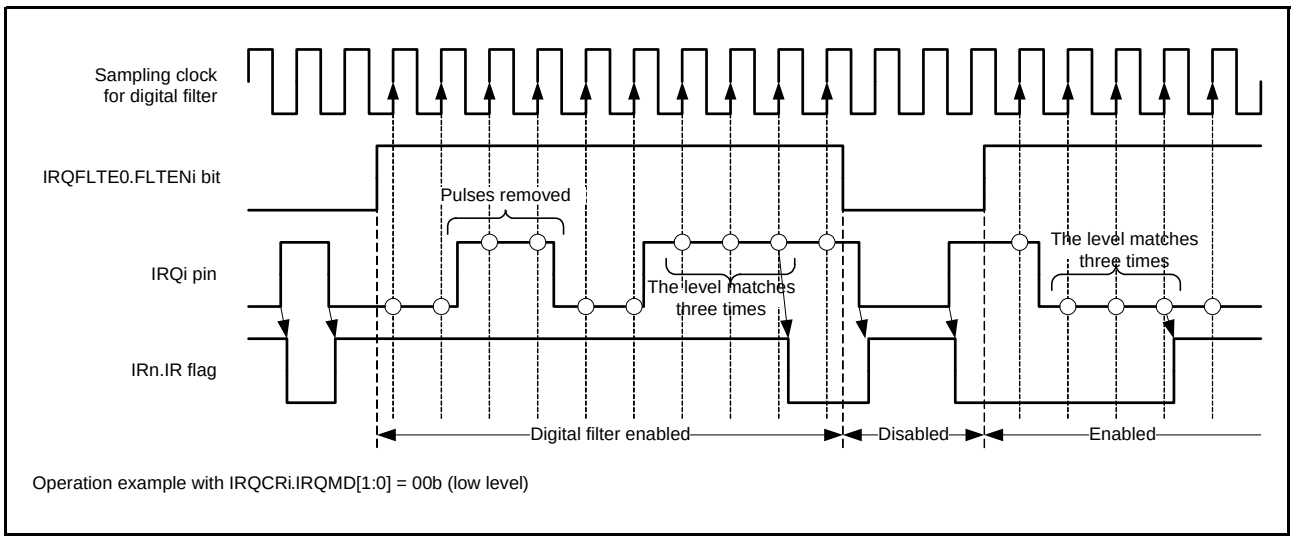


Figure 14.7 Digital Filter Operation Example

Before software standby mode is entered, set the IRQFLTE0.FLTENi and NMIFLTE.NFLTEN bits to 0 (digital filter disabled). To use the digital filter again after return from software standby mode, set the IRQFLTE0.FLTENi or NMIFLTE.NFLTEN bit to 1 (digital filter enabled).

14.4.8 External Pin Interrupts

The procedure for using the signal on an external pin as an interrupt is as follows.

1. Clear the IERm.IENj bit (m = 02h to 1Fh, j = 0 to 7) to 0 (interrupt request disabled).
2. Clear the IRQFLTE0.FLTENi bit (i = 0 to 5) to 0 (digital filter disabled).*1
3. Set the digital filter sampling clock with the IRQFLTC0.FCLKSELi[1:0] bits.*1
4. Make or confirm the I/O port settings.
5. Set the method of detection for the interrupt in the IRQCRi.IRQMD[1:0] bits.
6. Clear the corresponding IRn.IR flag (n = interrupt vector number) to 0 (if edge detection is in use).
7. Set the IRQFLTE0.FLTENi bit to 1 (digital filter enabled).*1
8. If the interrupt is to be used for DTC trigger, set the DTCERn.DTCE bit. The interrupt will be a CPU interrupt if the setting not is made.
9. Set the IERm.IENj bit to 1 (interrupt request enabled).

Note 1. To use the digital filter function, settings must be made beforehand.

14.5 Non-maskable Interrupt Operation

There are six types of non-maskable interrupt: the NMI pin interrupt, oscillation stop detection interrupt, IWDT underflow/refresh error, voltage monitoring 1 interrupt, and voltage monitoring 2 interrupt. Non-maskable interrupts are only usable as interrupts for the CPU; that is, they are not capable of DTC trigger. Non-maskable interrupts take precedence over all interrupts, including the fast interrupt.
Non-maskable interrupt requests are accepted regardless of the states of the I (interrupt enable) bit and IPL[3:0] (processor interrupt priority level) bits in the PSW of the CPU. The current states of the non-maskable interrupts can be checked in the non-maskable interrupt status register (NMISR).
Confirm that all bits in the NMISR have returned to 0 from within the handler for the non-maskable interrupt, before ending the handler.
Non-maskable interrupts are disabled by default. If a system is to use non-maskable interrupts, the following procedure must be followed at the beginning of program processing.

Non-maskable interrupt usage procedure:

1. Set the stack pointer (SP).
2. To use the NMI pin, clear the NMIFLTE.NFLTEN bit to 0 (digital filter disabled).*1
3. To use the NMI pin, set the digital filter sampling clock with the NMIFLTC.NFCLKSEL[1:0] bits.*1
4. To use the NMI pin, set the NMI pin detection sense with the NMICR.NMIMD bit.
5. To use the NMI pin, write 1 to the NMICLR.NMICLR bit to clear the NMISR.NMIST flag to 0.
6. To use the NMI pin, set the NMIFLTE.NFLTEN bit to 1 (digital filter enabled).*1
7. Enable the non-maskable interrupt by writing 1 to the corresponding bit in the non-maskable interrupt enable register (NMIER).

Note 1. To use the digital filter function, settings must be made beforehand.

After 1 is written to the NMIER register, subsequent write access to the NMIEN bit in NMIER is ignored. The NMI interrupt cannot be disabled. It can be disabled only by a reset.
For the flow of non-maskable interrupt processing, see section 13, Exception Handling.
Writing 1 to the NMICLR.NMICLR bit clears the NMI status flag (NMISR.NMIST) to 0.
Writing 1 to the NMICLR.OSTCLR bit clears the oscillation stop detection interrupt status flag (NMISR.OSTST) to 0.
Writing 1 to the NMICLR.IWDTCLR bit clears the IWDT underflow/refresh error status flag (NMISR.IWDTST) to 0.
Writing 1 to the NMICLR.LVD1CLR bit clears the voltage monitoring 1 interrupt status flag (NMISR.LVD1ST) to 0.
Writing 1 to the NMICLR.LVD2CLR bit clears the voltage monitoring 2 interrupt status flag (NMISR.LVD2ST) to 0.

14.6 Return from Power-Down States

The interrupt sources that can be used to return operation from sleep mode, deep sleep mode, or software standby mode are listed in Table 14.3, Interrupt Vector Table.
For details, refer to section 11, Low Power Consumption. The following describes how to use an interrupt to return operation from each low power consumption mode.

14.6.1 Return from Sleep Mode or Deep Sleep Mode

If the interrupt controller is to return operation from sleep mode in response to an interrupt or non-maskable interrupt, make the following settings for the interrupt.

- Interrupts

1. Select the CPU as the interrupt request destination.
2. Use the IENj bit in IERm (m = 02h to 1Fh, j = 0 to 7) to enable the given interrupt request.
3. Set a priority level higher than that set by the IPL[3:0] bits in the PSW of CPU.

- Non-maskable interrupts

Use the NMIER register to enable the given interrupt request.

14.6.2 Return from Software Standby Mode

The interrupt controller can return operation from a non-maskable interrupt or an interrupt that enables the return from the software standby mode.
The conditions for the return are listed below.

- Interrupts

1. Select the interrupt source that enables the return from the software standby mode.
2. Select the CPU as the interrupt request destination.
3. Use the IENj bit in IERm (m = 02h to 1Fh, j = 0 to 7) to enable the given interrupt request.
4. Set a priority level higher than that set by the IPL[3:0] bits in the PSW of CPU.
   (For the interrupt source specified as a fast interrupt, as well as setting the fast interrupt set register (FIR), the interrupt priority level (IPRn (n = interrupt vector number)) should be set above the level set by IPL in the PSW of the CPU.)

Interrupt requests through the IRQ pins that do not satisfy the above conditions are not detected while the clock is stopped in software standby mode.

- Non-maskable interrupts

Use the NMIER register to enable the given interrupt request.

- Procedure to make a transition to/from software standby mode

1. Before software standby mode is entered, disable the digital filter for the interrupt source as a return target (IRQFLTE0.FLTENi = 0, NMIFLTE.NFLTEN = 0).
2. To use the digital filter again after return from software standby mode, enable the digital filter (IRQFLTE0.FLTENi = 1, NMIFLTE.NFLTEN = 1).

14.7 Usage Note

14.7.1 Note on WAIT Instruction Used with Non-Maskable Interrupt

Before executing the WAIT instruction, check to see that all the status flags in NMISR are 0.

15. Buses

15.1 Overview

Table 15.1 lists the bus specifications, Figure 15.1 shows the bus configuration, and Table 15.2 lists the addresses assigned to each bus.

Table 15.1 Bus Specifications

Bus Type		Description
CPU bus	Instruction bus	• Connected to the CPU for instructions • Connected to on-chip memory (RAM, ROM) • Operates in synchronization with the system clock (ICLK)
	Operand bus	• Connected to the CPU (for operands) • Connected to on-chip memory (RAM, ROM) • Operates in synchronization with the system clock (ICLK)
Memory bus	Memory bus 1	• Connected to RAM
	Memory bus 2	• Connected to ROM
Internal main buses	Internal main bus 1	• Connected to the CPU • Operates in synchronization with the system clock (ICLK)
	Internal main bus 2	• Connected to the DTC • Connected to on-chip memory (RAM, ROM) • Operates in synchronization with the system clock (ICLK)
Internal peripheral buses	Internal peripheral bus 1	• Connected to peripheral modules (DTC, interrupt controller, and bus error monitoring section) • Operates in synchronization with the system clock (ICLK)
	Internal peripheral bus 2	• Connected to peripheral modules • Operates in synchronization with the peripheral module clock (PCLKB, PCLKD)
	Internal peripheral bus 3	• Connected to peripheral modules (CMPC) • Operates in synchronization with the peripheral module clock (PCLKB)
	Internal peripheral bus 6	• Connected to ROM (P/E) and E2 DataFlash • Operates in synchronization with the FlashIF clock (FCLK)

P/E: Programming/Erasure

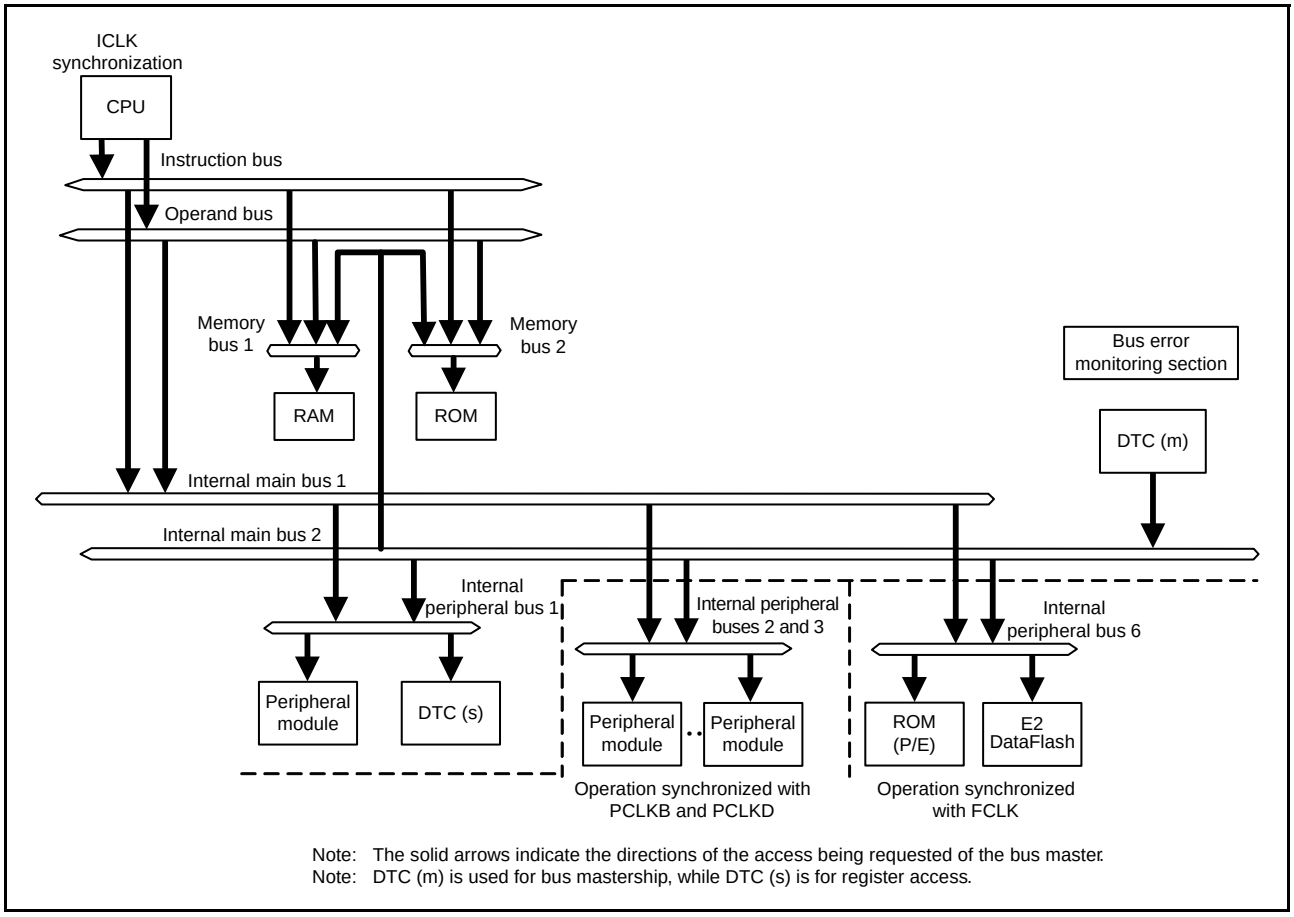


Figure 15.1 Bus Configuration

Table 15.2 Addresses Assigned for Each Bus

Address	Bus	Area
0000 0000h to 0000 FFFFh	Memory bus 1	RAM
0008 0000h to 0008 7FFFh	Internal peripheral bus 1	Peripheral I/O registers
0008 8000h to 0009 FFFFh	Internal peripheral bus 2	
000A 0000h to 000B FFFFh	Internal peripheral bus 3	
0010 0000h to 00FF FFFFh	Internal peripheral bus 6	E2 DataFlash memory and ROM (for programming/erasure)
8000 0000h to FEFF FFFFh	Memory bus 2	ROM (for reading only)
FF00 0000h to FFFF FFFFh		

15.2 Description of Buses

15.2.1 CPU Buses

The CPU buses consist of the instruction and operand buses, which are connected to internal main bus 1. As the names suggest, the instruction bus is used to fetch instructions for the CPU, while the operand bus is used for operand access. Connection of the instruction and operand buses to RAM and ROM provides the CPU with direct access to these areas, i.e. access is not via internal main bus 1. However, only reading is possible in direct access to ROM by the CPU; programming and erasure are handled via an internal peripheral bus.
Bus requests for instruction fetching and operand access are arbitrated through internal main bus 1. The order of priority is operand access then instruction fetching.
If instruction fetching and operand access are requested for different buses (memory bus 1, memory bus 2, and internal main bus 1), the bus-access operations can proceed simultaneously. For example, parallel access to ROM and RAM is possible.

15.2.2 Memory Buses

The memory buses consist of memory bus 1 and memory bus 2. RAM is connected to memory bus 1 and ROM is connected to memory bus 2. Requests for bus mastership from the CPU buses (instruction fetching and operand) and internal main bus 2 are arbitrated through memory buses 1 and 2.
The priority order of CPU bus and internal main bus 2 can be set using the memory bus 1 (RAM) priority control bits (BPRA[1:0]) and memory bus 2 (ROM) priority control bits (BPRO[1:0]) in the bus priority control register (BUSPRI) for the corresponding memory buses. When the priority order is fixed, internal main bus 2 has priority over the CPU bus (operand over instruction fetching). When the priority order is toggled, the bus for which a request has been accepted has lower priority.

15.2.3 Internal Main Buses

The internal main buses consist of a bus for use by the CPU (internal main bus 1) and a bus for use by the other bus-master modules, i.e. the DTC (internal main bus 2).
Bus requests for instruction fetching and operand access are arbitrated through internal main bus 1. The order of priority is operand access then instruction fetching.
Requests for bus mastership from the DTC is arbitrated by internal main bus 2. The order of priority is as shown in Table 15.3.
If the CPU and another bus master are requesting access to different buses (on-chip memory, internal peripheral buses 1 to 3 and 6), the respective bus-access operations can proceed simultaneously.
However, when the CPU executes the XCHG instruction, requests for bus access from masters other than the CPU are not accepted until data transfer for the XCHG instruction is completed regardless of the bus priority control register (BUSPRI) setting. Furthermore, requests for bus access from masters other than the DTC are not accepted during reading and writing-back of transfer control information for the DTC.

Table 15.3 Order of Priority for Bus Masters

Priority	Bus Master
High ↑ Low	DTC
	CPU

15.2.4 Internal Peripheral Buses

Connection of peripheral modules to the internal peripheral buses is as described in Table 15.4.

Table 15.4 Connection of Peripheral Modules to the Internal Peripheral Buses

Type of Bus	Peripheral Modules
Internal peripheral bus 1	DTC, interrupt controller, and bus error monitoring section
Internal peripheral bus 2	Peripheral modules other than those connected to internal peripheral buses 1 and 3
Internal peripheral bus 3	CMPC
Internal peripheral bus 6	ROM (P/E)/E2 DataFlash

Requests for bus mastership from the CPU (internal main bus 1) and other bus masters (internal main bus 2) are arbitrated through internal peripheral buses 1 to 3 and 6.
The priority order of the two internal main buses can be set using the bus priority control register (BUSPRI). The priority order can be set with the internal peripheral bus 1 priority control bits (BUSPRI.BPIB[1:0]), internal peripheral buses 2 and 3 priority control bits (BUSPRI.BPGB[1:0]), and internal peripheral bus 6 priority control bits (BUSPRI.BPFB[1:0]) for the corresponding internal peripheral buses. When the priority order is fixed, internal main bus 2 has priority over internal main bus 1. When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.
The order of accepting requests may change depending on the BUSPRI setting (see Figure 15.2).

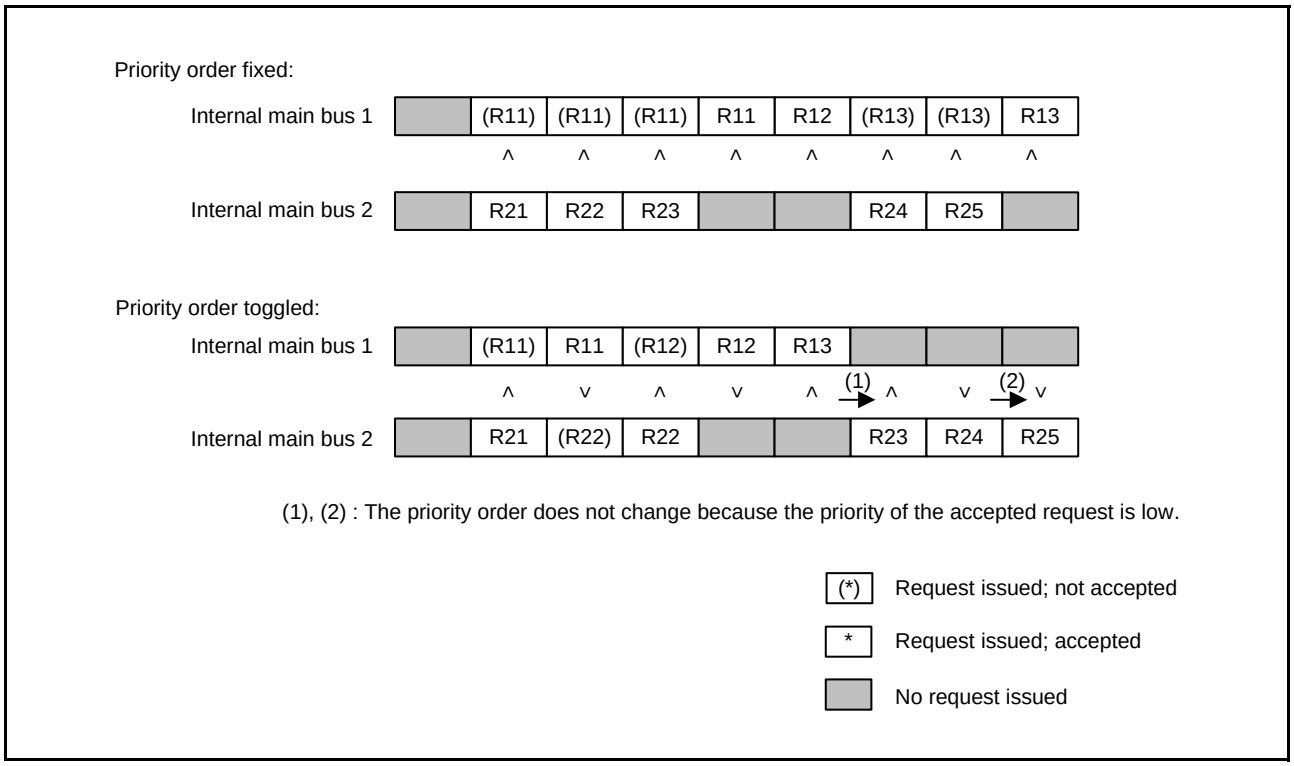


Figure 15.2 Priority Order Between Internal Peripheral Bus Accesses

15.2.5 Write Buffer Function (Internal Peripheral Bus)

The internal peripheral bus has the write buffer function, which allows the next round of bus access to start, before the current write access is completed, in write access. However, if the following round of bus access is from the same bus master but to the different internal peripheral bus, it is suspended until the bus operations already in progress are completed. When read access to the internal memory is scheduled after the write access to the internal peripheral bus from the CPU, the following round of bus access can be started before the current bus operation is completed and thus the order of accesses may be changed (see Figure 15.3).

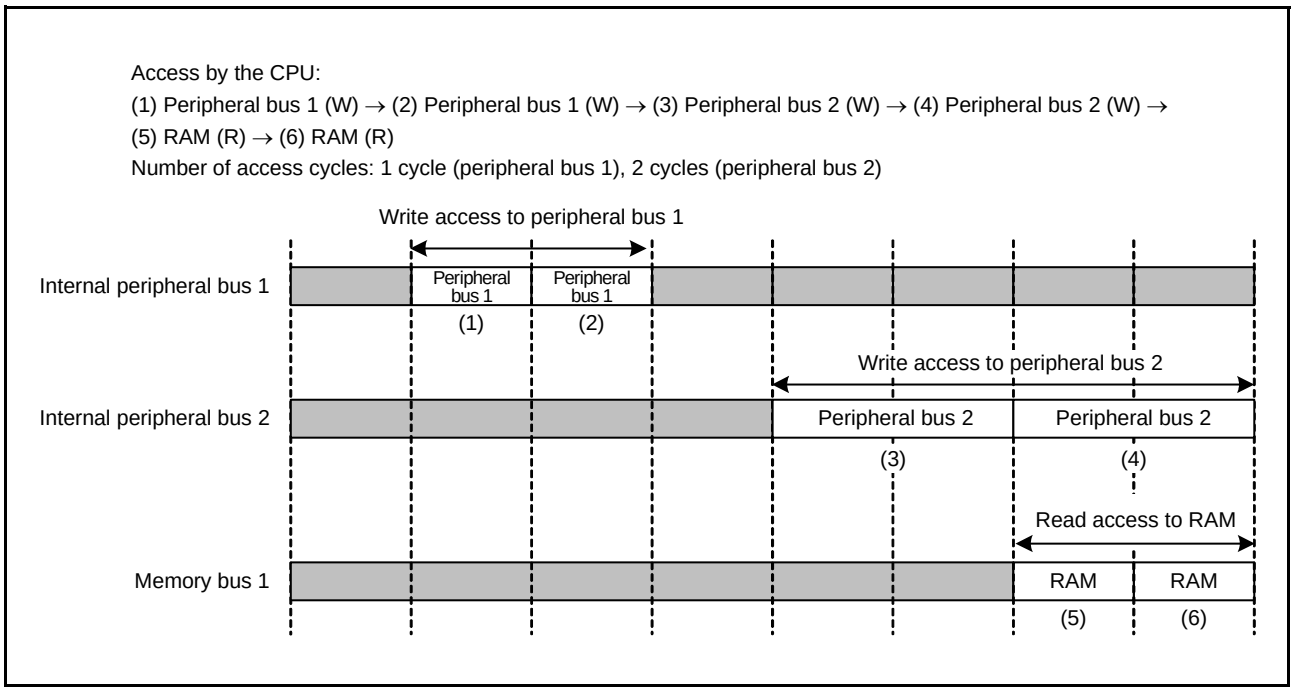


Figure 15.3 Write Buffer Function

15.2.6 Parallel Operation

Parallel operation is possible when different bus-master modules are requesting access to different slave modules. For example, if the CPU is fetching an instruction from ROM and an operand from RAM, the DTC is able to handle transfer between a peripheral bus and peripheral bus at the same time.
An example of parallel operations is shown in Figure 15.4. In this example, the CPU is able to employ the instruction and operand buses for simultaneous access to ROM and RAM, respectively. Furthermore, the DTC simultaneously employs internal main bus 2 for access to a peripheral bus during access to RAM and ROM by the CPU.

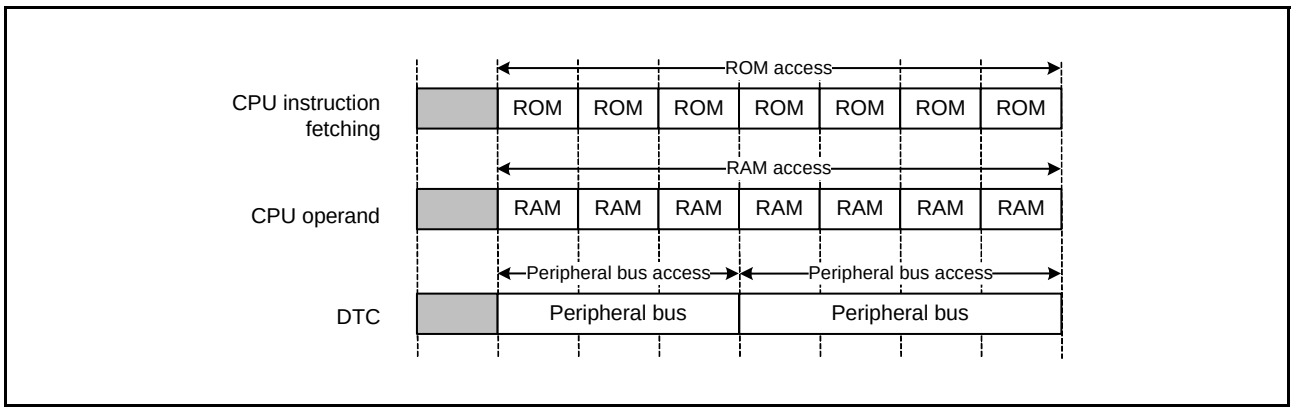


Figure 15.4 Example of Parallel Operations

15.2.7 Restrictions

(1) Prohibition of Access that Spans Multiple Areas of Address Space

Single access that spans two areas of the address space is prohibited, and operation of such an access is not guaranteed. Ensure that a single word or longword access does not span across two areas by crossing address space area boundaries.

(2) Restrictions on RMPA and String-Manipulation Instructions

(a) The allocation of data to be handled by RMPA or string-manipulation instructions to I/O registers is prohibited, and operation is not guaranteed if this restriction is not observed.

15.3 Register Descriptions

15.3.1 Bus Error Status Clear Register (BERCLR)

Address(es): 0008 1300h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	STSCLR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STSCLR	Status Clear	0: Invalid 1: Bus error status register cleared	(W)*1
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only writing 1 is effective; i.e. writing 0 has no effect.

STSCLR Bit (Status Clear)

Writing 1 to this bit clears the bus error status registers 1 and 2 (BERSR1 and BERSR2).
Writing 0 has no effect. It is read as 0.

15.3.2 Bus Error Monitoring Enable Register (BEREN)

Address(es): 0008 1304h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TOEN	IGAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IGAEN	Illegal Address Access Detection Enable	0: Illegal address access detection is disabled. 1: Illegal address access detection is enabled.	R/W
b1	TOEN	Timeout Detection Enable*1, *2	0: Bus timeout detection is disabled. 1: Bus timeout detection is enabled.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When detection is disabled (the TOEN bit is cleared to 0), bus access can cause the bus to freeze.
Note 2. Do not set the TOEN bit to 0 (bus timeout detection disabled) while timeout errors are being detected.

15.3.3 Bus Error Status Register 1 (BERSR1)

Address(es): 0008 1308h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	MST[2:0]			—	—	TO	IA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IA	Illegal Address Access	0: Illegal address access not made 1: Illegal address access made	R
b1	TO	Timeout	0: Timeout not generated 1: Timeout generated	R
b3, b2	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b6 to b4	MST[2:0]	Bus Master Code	b6 b4 0 0 0: CPU 0 0 1: Reserved 0 1 0: Reserved 0 1 1: DTC 1 0 0: Reserved 1 0 1: Reserved 1 1 0: Reserved 1 1 1: Reserved	R
b7	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

MST[2:0] Bits (Bus Master Code)

These bits indicate the bus master that accessed a bus when a bus error occurred.

15.3.4 Bus Error Status Register 2 (BERSR2)

Address(es): 0008 130Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADDR[12:0]													—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b15 to b3	ADDR[12:0]	Bus Error Occurrence Address	The upper 13 bits of an address that was accessed when a bus error occurred (in units of 512 Kbytes).	R

15.3.5 Bus Priority Control Register (BUSPRI)

Address(es): 0008 1310h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	BPFB[1:0]		—	—	BPGB[1:0]		BPIB[1:0]		BPRO[1:0]		BPRA[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	BPRA[1:0]	Memory Bus 1 (RAM) Priority Control	b1 b0 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R/(W) *1
b3, b2	BPRO[1:0]	Memory Bus 2 (ROM) Priority Control	b3 b2 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R/(W) *1
b5, b4	BPIB[1:0]	Internal Peripheral Bus 1 Priority Control	b5 b4 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R/(W) *1
b7, b6	BPGB[1:0]	Internal Peripheral Buses 2 and 3 Priority Control	b7 b6 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R/(W) *1
b9, b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b11, b10	BPFB[1:0]	Internal Peripheral Bus 6 Priority Control	b11 b10 0 0: The order of priority is fixed. 0 1: The order of priority is toggled. 1 0: Setting prohibited 1 1: Setting prohibited	R/(W) *1
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits can be written to only once while the DTC is stopped. When they are written to more than one time, the operation is not guaranteed.

BPRA[1:0] Bits (Memory Bus 1 (RAM) Priority Control)

These bits specify the priority order for memory bus 1 (RAM).

When the priority order is fixed, internal main bus 2 has priority over the CPU bus.

When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPRO[1:0] Bits (Memory Bus 2 (ROM) Priority Control)

These bits specify the priority order for memory bus 2 (ROM).

When the priority order is fixed, internal main bus 2 has priority over the CPU bus.

When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPIB[1:0] Bits (Internal Peripheral Bus 1 Priority Control)

These bits specify the priority order for internal peripheral bus 1.

When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.

When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPGB[1:0] Bits (Internal Peripheral Buses 2 and 3 Priority Control)

These bits specify the priority order for internal peripheral buses 2 and 3.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

BPFB[1:0] Bits (Internal Peripheral Bus 6 Priority Control)

These bits specify the priority order for internal peripheral bus 6.
When the priority order is fixed, internal main bus 2 has priority over internal main bus 1.
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted.

15.4 Bus Error Monitoring Section

The bus error monitoring section monitors the individual areas for bus errors, and when a bus error occurs, the error is indicated to the bus master.

15.4.1 Type of Bus Error

There are two types of bus error: illegal address access and timeout.
Illegal address access is the detection of illegal access to an area, and time-out is the detection of a bus-access operation not being completed within 768 cycles.

15.4.1.1 Illegal Address Access

When the illegal address access detection enable bit (IGAEN) in the bus error monitoring enable register (BEREN) is set to 1, access to an illegal address area leads to illegal address access errors.
The address ranges where access will lead to illegal address access errors are listed in Table 15.5.

15.4.1.2 Timeout

When the timeout detection enable bit in the bus error monitoring enable register is enabled (BEREN.TOEN = 1), bus access that is not completed within 768 cycles leads to a timeout error.

- Internal peripheral buses (2 and 3): Bus access is not completed within 768 peripheral module clock (PCLKB) cycles from the start of the access.
  If a timeout error occurs, accesses from the bus master are not accepted for 256 PCLKB cycles.
- Internal peripheral bus (6): Bus access is not completed within 768 FlashIF clock (FCLK) cycles from the start of the access.
  If a timeout error occurs, accesses from the bus master are not accepted for 256 FCLK cycles.

15.4.2 Operations When a Bus Error Occurs

When a bus error occurs, the error is indicated to the CPU. Operation is not guaranteed when a bus error occurs.

- Bus error indication to the CPU
  An interrupt is generated. The ICU.IERn register can specify whether to generate an interrupt in the case of a bus error.

15.4.3 Conditions Leading to Bus Errors

Table 15.5 lists the type of bus errors for each area in the respective address space.
If an illegal address access error or timeout is detected when no bus error has occurred (bus error status register n (BERSRn; n = 1 or 2) is cleared), the detected error is reflected on the BERSRn. Once a bus error occurs, no subsequent bus errors are reflected in the register unless the register is cleared.
If bus errors are simultaneously caused by two or more bus masters, error information of only one bus master is reflected. Once a bus error occurs, the status is retained until the BERSRn register is cleared.

Table 15.5 Type of Bus Errors

Address	Type of Area	Type of Error	
		Illegal Address Access	Timeout
0000 0000h to 0007 FFFFh	Memory bus 1	—	—
0008 0000h to 0008 7FFFh	Internal peripheral bus 1	—	—
0008 8000h to 0009 FFFFh	Internal peripheral bus 2	Δ	Δ
000A 0000h to 000B FFFFh	Internal peripheral bus 3	Δ	—
000C 0000h to 000F FFFFh	Reserved area	○	—
0010 0000h to 00FF FFFFh	Internal peripheral bus 6	Δ	—
0100 0000h to 07FF FFFFh	Reserved area	○	—
0800 0000h to 0FFF FFFFh	Reserved area	—	—
1000 0000h to 7FFF FFFFh	Reserved area	○	—
8000 0000h to FFFF FFFFh	Memory bus 2	—	—

—: A bus error does not result.
Δ: A bus error may or may not result.
○: A bus error results.
Note: The capacity of the RAM, data flash, and ROM differs depending on the product. For details, refer to section 30, RAM, and section 31, Flash Memory (FLASH).

15.5 Interrupt

15.5.1 Interrupt Source

An illegal address access error or detection of a timeout leads to a bus error signal for the interrupt controller.

Table 15.6 Interrupt Source

Name	Interrupt Source	DTC Activation
BUSERR	Illegal address access error or timeout	Not possible

16. Data Transfer Controller (DTCb)

This MCU incorporates a data transfer controller (DTC).
The DTC is triggered by an interrupt request to perform data transfers.
In addition to the conventional methods of DTC transfer (normal, repeat, block, and chain), DTCb supports sequential transfer, in which it handles a series of transfers made up of a combination of the other methods. In sequential transfer, the data that is initially transferred selects one from possible 256 sequences for execution. The DTCb can divide one sequence into several transfers depending on how the parts of the sequence are combined.

16.1 Overview

Table 16.1 lists the specifications of the DTC, and Figure 16.1 shows a block diagram of the DTC.

Table 16.1 DTC Specifications

Item	Description
Number of transfer channels	• The same number as all interrupt sources that can start the DTC transfer.
Transfer modes	• Normal transfer mode A single transfer request leads to a single data transfer. • Repeat transfer mode A single transfer request leads to a single data transfer. The transfer address is returned to the transfer start address after the number of data transfers corresponding to “repeat size”. The maximum number of repeat transfers is 256, and the maximum data transfer size is 256 × 32 bits, 1024 bytes. • Block transfer mode A single transfer request leads to the transfer of a single block. The maximum block size is 256 × 32 bits = 1024 bytes.
Chain transfer	• Multiple types of data transfers can sequentially be executed in response to a single request. • Either “performed only when the transfer counter becomes 0” or “every time” can be selected.
Sequence transfer	A series of complicated transfers can be registered as a sequence. Any sequence can be selected by the transfer data and executed. • Only one trigger source can be set at a time. • Up to 256 sequences for a single trigger source • The data that is initially transferred in response to a transfer request determines a sequence • The whole sequence can be executed on a single request, or be suspended in the middle of the sequence and resumed on the next transfer request (division of sequence).
Transfer space	• In short-address mode: 16 Mbytes (Areas from 0000 0000h to 007F FFFFh and FF80 0000h to FFFF FFFFh except reserved areas) • In full-address mode: 4 Gbytes (Area from 0000 0000h to FFFF FFFFh except reserved areas)
Data transfer units	• Single data: 1 byte (8 bits), 1 word (16 bits), 1 longword (32 bits) • Single block size: 1 to 256 data
CPU interrupt source	• An interrupt request can be generated to the CPU on a request source for a data transfer. • An interrupt request can be generated to the CPU after a single data transfer. • An interrupt request can be generated to the CPU after data transfer of specified volume.
Read skip	Reading of the transfer information can be skipped when the same transfer is repeated.
Write-back skip	Write-back of the transferred data that is not updated can be skipped when the address of the transfer source or destination is fixed.
Write-back disable	Allows disabling the write-back of transfer information.
Displacement addition	The displacement value can be added to the transfer source address (for each transfer information)
Low power consumption function	Module stop state can be set.

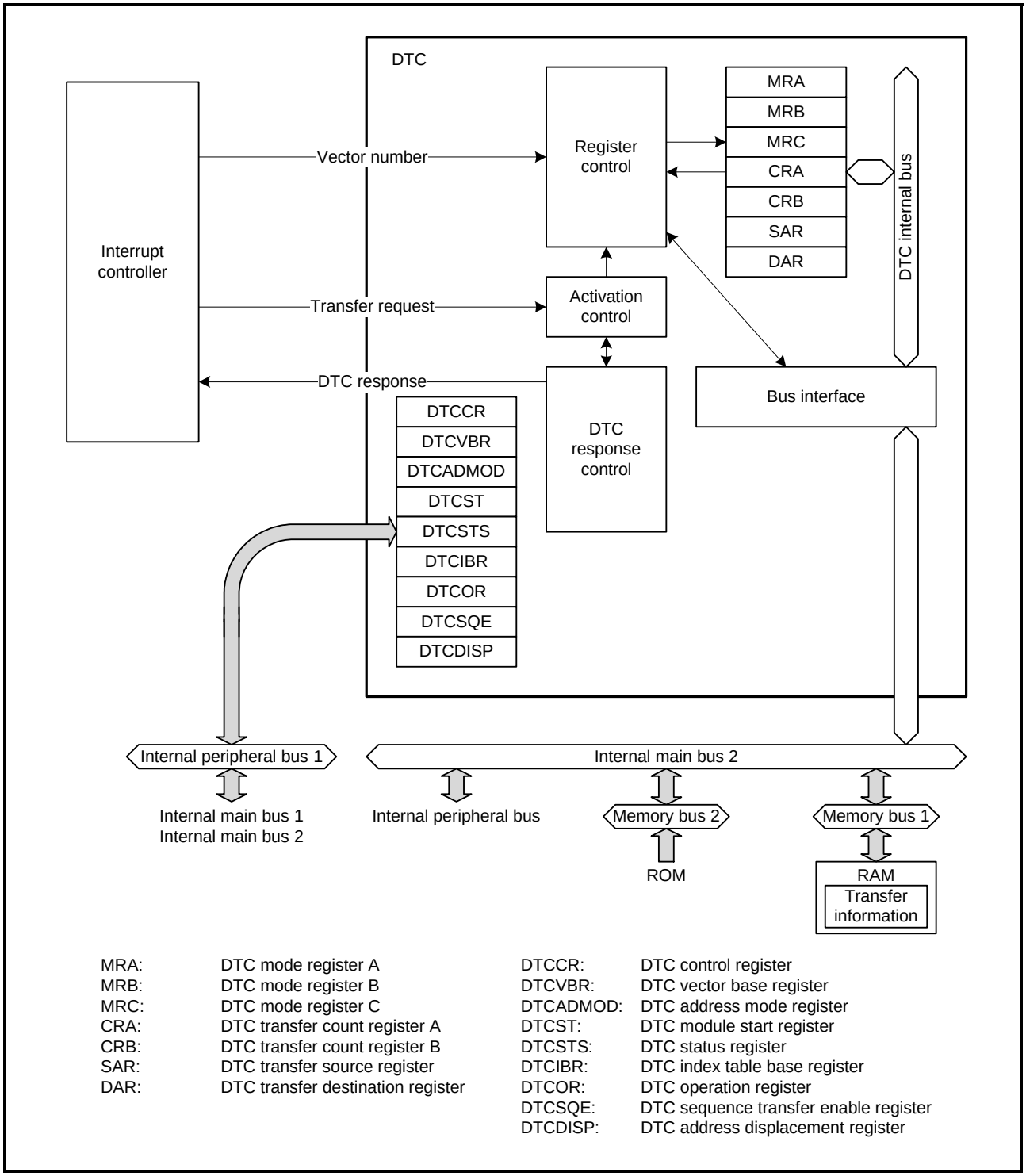


Figure 16.1 DTC Block Diagram

16.2 Register Descriptions

Registers MRA, MRB, MRC, SAR, DAR, CRA, and CRB are DTC internal registers, which cannot be directly accessed from the CPU. Values to be set in these DTC internal registers are placed in the RAM area as transfer information. When accepting a transfer request, the DTC reads the transfer information from the RAM area and sets it in the internal registers. After the data transfer ends, the values of the updated internal register are written back to the RAM area as transfer information.

16.2.1 DTC Mode Register A (MRA)

Address(es): (inaccessible directly from the CPU)

	b7	b6	b5	b4	b3	b2	b1	b0
	MD[1:0]		SZ[1:0]		SM[1:0]		—	WBDIS
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	WBDIS	Write-back Disable	0: Writes back the transfer information on completion of the data transfer 1: Does not write back the transfer information on completion of the data transfer	—
b1	—	Reserved	Set this bit to 0.	—
b3, b2	SM[1:0]	Transfer Source Address Addressing Mode	b3 b2 0 0: The address in the SAR register is fixed. (write-back to SAR is skipped.) 0 1: The address in the SAR register is fixed. (write-back to SAR is skipped.) 1 0: The SAR value is incremented after a data transfer. (+1 when the SZ[1:0] bits are 00b, +2 when 01b, +4 when 10b) 1 1: The SAR value is decremented after a data transfer. (–1 when the SZ[1:0] bits are 00b, –2 when 01b, –4 when 10b)	—
b5, b4	SZ[1:0]	DTC Data Transfer Size	b5 b4 0 0: Byte (8-bit) transfer 0 1: Word (16-bit) transfer 1 0: Longword (32-bit) transfer 1 1: Setting prohibited	—
b7, b6	MD[1:0]	DTC Transfer Mode Select	b7 b6 0 0: Normal transfer mode 0 1: Repeat transfer mode 1 0: Block transfer mode 1 1: Setting prohibited	—

MRA register is used to select the DTC operating mode and cannot be accessed directly from the CPU.

WBDIS Bit (Write-back Disable)

The WBDIS bit selects whether to write back the transfer information.

When the bit is 0, updated transfer information is written back.

When the bit is 1, updated transfer information is not written back even with the setting of that address is incremented after a transfer, and the same data transfer is executed every time for each transfer request. The transfer information can be stored in ROM because the transfer information is not written back.

While the WBDIS bit is 1, operation for each transfer mode is as follows:

(1) Normal transfer and repeat transfer modes
1-byte, 1-word, or 1-longword of data is transferred on a single transfer request. The transfer address and transfer count are not updated so that the same transfer is repeated on each transfer request. When the transfer count is 1, the ICU.DTCERn.DTCE bit is not set to 0, and data transfer continues in response to the next transfer request.

(2) Block transfer mode
1-block of data is transferred on a single transfer request. The transfer address and transfer count are not updated so that the same block transfer is repeated on each transfer request. When the block transfer count is 1, the ICU.DTCERn.DTCE bit is not set to 0, and data transfer continues in response to the next transfer request.

When setting the DISPE bit to 1, set the MRA.WBDIS bit to 1 (does not write back the transfer information). If the value of the WBDIS bit in any transfer information is 1, set the DTCCR.RRS bit to 0 (so that reading of the transfer information is not skipped).

16.2.2 DTC Mode Register B (MRB)

Address(es): (inaccessible directly from the CPU)

	b7	b6	b5	b4	b3	b2	b1	b0
	CHNE	CHNS	DISEL	DTS	DM[1:0]		INDX	SQEND
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	SQEND	Sequence Transfer End	0: Continue the sequence transfer 1: End the sequence transfer	—
b1	INDX	Index Table Reference	0: Does not refer to the index table 1: Refers the index table based on the transferred data*1	—
b3, b2	DM[1:0]	Transfer Destination Address Addressing Mode	b3 b2 0 0: The address in the DAR register is fixed. (Write-back to DAR is skipped.) 0 1: The address in the DAR register is fixed. (Write-back to DAR is skipped.) 1 0: The DAR value is incremented after data transfer. (+1 when the MRA.SZ[1:0] bits are 00b, +2 when 01b, +4 when 10b) 1 1: The DAR value is decremented after data transfer. (–1 when the MRA.SZ[1:0] bits are 00b, –2 when 01b, –4 when 10b)	—
b4	DTS	DTC Transfer Mode Select	0: Transfer destination side is repeat area or block area. 1: Transfer source side is repeat area or block area.	—
b5	DISEL	DTC Interrupt Select	0: An interrupt request to the CPU is generated on completion of the specified number of data transfers. 1: An interrupt request to the CPU is generated for each data transfer.	—
b6	CHNS	DTC Chain Transfer Select	0: Chain transfer is performed on completion of each transfer. 1: Chain transfer is performed only when the transfer counter is changed from 1 to 0 or 1 to CRAH.	—
b7	CHNE	DTC Chain Transfer Enable	0: Chain transfer is disabled. 1: Chain transfer is enabled.	—

Note 1. Set the MRA.MD[1:0] bits to 00b (normal transfer mode) when setting the INDX bit to 1.

MRB register is used to select the DTC operating mode and cannot be accessed directly from the CPU.

SQEND Bit (Sequence Transfer End)

The SQEND bit selects whether to continue or end sequence transfer. Refer to Table 16.2 for details.
This bit can only be set to 1 for transfer information referred to by the DTC index table. Set this bit to 0 for transfer information referred to by the DTC vector table.

INDX Bit (Index Table Reference)

When the value of the INDX bit in transfer information that is read is 1, a sequence transfer proceeds. Refer to Table 16.2 for details.
Set this bit to 0 for transfer information which is not associated with sequence transfer or is not intended to start sequence transfer. Do not allow transfer requests to be generated by the sources different from that specified in the DTCSQE register but having the INDX bit set to 1.

Table 16.2 Values of Bits CHNE, SQEND, and INDX in the Sequence Transfer and DTC Operation

CHNE Bit	SQEND Bit	INDX Bit	Operation	Usage
0	0	1	Start sequence transfer	Use this setting for the transfer information that is first read in response to a transfer request from the source specified in the DTCSQE register.
1	0	0	Continue sequence transfer	Use this setting for the first or intermediate transfer information in a sequence.
0	0	0	Suspend sequence transfer	Use this setting for the first or intermediate transfer information in a sequence.
0	1	0	End sequence transfer	Use this setting with the last transfer information in a sequence.
0	1	1	End current sequence transfer and start new sequence transfer	Use this setting with the last transfer information in a sequence.

Note: Do not set the values other than listed above.

DTS Bit (DTC Transfer Mode Select)

The DTS bit specifies the side (transfer source or destination) to be a repeat area or block area in repeat transfer mode or block transfer mode.

CHNS Bit (DTC Chain Transfer Select)

The CHNS bit selects the chain transfer condition.

When the CHNE bit is 0, setting of the CHNS bit is ignored. For details on the conditions to select the chain transfer, refer to Table 16.4, Chain Transfer Conditions.

When the next transfer is chain transfer, completion of the specified number of transfers is not determined, the interrupt status flag for the request source is not cleared, and an interrupt request to the CPU is not generated.

CHNE Bit (DTC Chain Transfer Enable)

The CHNE bit enables or disables chain transfer.

The chain transfer condition is selected by the CHNS bit.

For details of chain transfer, refer to section 16.4.6, Chain Transfer.

Refer to Table 16.2 for the setting value to be used in the sequence transfer.

16.2.3 DTC Mode Register C (MRC)

Address(es): (inaccessible directly from the CPU)

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DISPE
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	DISPE	Displacement Addition	0: The displacement value is not added to the transfer source address. 1: The displacement value is added to the transfer source address.	—
b7 to b1	—	Reserved	Set these bits to 0.	—

The MRC register is used to select DTC operating mode and cannot be accessed directly from the CPU.
This register can only be used in full-address mode, but not in short-address mode. Therefore, set the DTCADMOD.SHORT bit to 0 (full-address mode) when using the displacement addition function.

DISPE Bit (Displacement Addition)

This bit specifies whether to use the SAR + DTCDISP value as the transfer source address.
When setting the DISPE bit to 1, set the MRA.WBDIS bit to 1 (does not write back the transfer information) and set the DTCCR.RRS bit to 0 (transfer information read is not skipped).

16.2.4 DTC Transfer Source Register (SAR)

Address(es): (inaccessible directly from the CPU)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

SAR register is used to set the transfer source start address.
In full-address mode, 32 bits are valid.
In short-address mode, lower 24 bits are valid and upper 8 bits (b31 to b24) are ignored. The address of this register is extended by the value specified by b23.
SAR register cannot be accessed directly from the CPU.

16.2.5 DTC Transfer Destination Register (DAR)

Address(es): (inaccessible directly from the CPU)

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

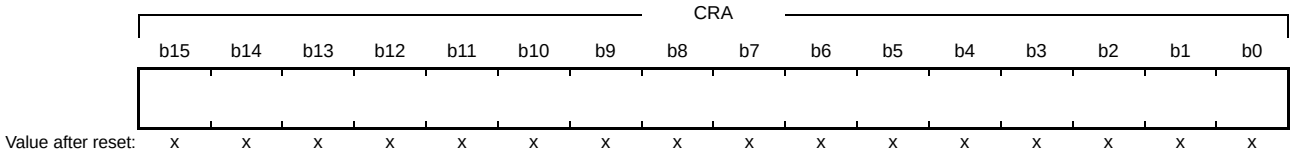
x: Undefined

DAR register is used to set the transfer destination start address.
In full-address mode, 32 bits are valid.
In short-address mode, lower 24 bits are valid and upper 8 bits (b31 to b24) are ignored. The address of this register is extended by the value specified by b23.
DAR register cannot be accessed directly from the CPU.

16.2.6 DTC Transfer Count Register A (CRA)

- Normal transfer mode

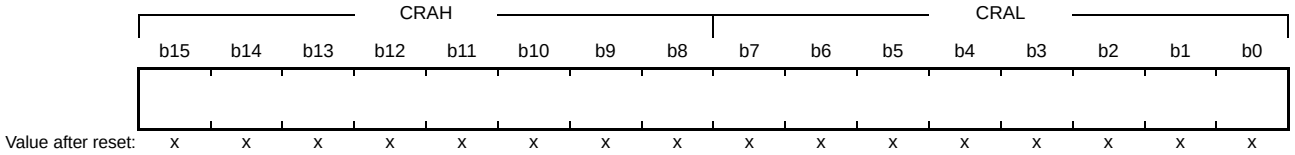
Address(es): (inaccessible directly from the CPU)



x: Undefined

- Repeat transfer mode/block transfer mode

Address(es): (inaccessible directly from the CPU)



x: Undefined

Symbol	Register Name	Description	R/W
CRAL	Transfer Counter A Lower Register	Set transfer count. This register functions as a transfer counter during data transfer.	—
CRAH	Transfer Counter A Upper Register	Set transfer count. This register functions as a reload register during data transfer.	—

Note: The function depends on transfer mode.
Note: Set CRAH and CRAL to the same value in repeat transfer mode and block transfer mode.

This register is for counting the number of transfers and cannot be accessed directly from the CPU.

(1) Normal transfer mode (MRA.MD[1:0] bits = 00b)

CRA register functions as a 16-bit transfer counter in normal transfer mode.
The transfer count is 1, 65535, and 65536 when the set value is 0001h, FFFFh, and 0000h, respectively.
The CRA value is decremented (–1) at each data transfer.

(2) Repeat transfer mode (MRA.MD[1:0] bits = 01b)

The CRAH register retains the transfer count and the CRAL register functions as an 8-bit transfer counter.
The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively.
The CRAL value is decremented (–1) at each data transfer. When it reaches 00h, the CRAH value is reloaded to the CRAL register.

(3) Block transfer mode (MRA.MD[1:0] bits = 10b)

The CRAH register retains the block size and the CRAL register functions as an 8-bit block size counter.
The transfer count is 1, 255, and 256 when the set value is 01h, FFh, and 00h, respectively.
The CRAL value is decremented (–1) at each data transfer. When it reaches 00h, the CRAH value is reloaded to the CRAL register.

16.2.7 DTC Transfer Count Register B (CRB)

Address(es): (inaccessible directly from the CPU)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

CRB register is used to set the block transfer count for block transfer mode and cannot be accessed directly from the CPU.
The transfer count is 1, 65535, and 65536 when the set value is 0001h, FFFFh, and 0000h, respectively.
The CRB value is decremented (–1) when the final data of a single block size is transferred.
When normal transfer mode or repeat transfer mode is selected, this register is not used and the set value is ignored.

16.2.8 DTC Control Register (DTCCR)

Address(es): DTC.DTCCR 0008 2400h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	RRS	—	—	—	—
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b4	RRS	DTC Transfer Information Read Skip Enable*1	0: Transfer information read is not skipped. 1: Transfer information read is skipped when vector numbers match.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Set this bit to 0 when using the sequence transfer.

DTCCR register is used to control the DTC operation.

RRS Bit (DTC Transfer Information Read Skip Enable)

The DTC vector number is compared with the vector number in the previous data transfer.
When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transferred information. However, when the previous transfer was chain transfer, the transferred information is read regardless of the value of the RRS bit.
Furthermore, when the transfer counter (CRA register) became 0 during the previous normal transfer and when the transfer counter (CRB register) became 0 during the previous block transfer, the transferred information is read regardless of the RRS bit value.
If the value of the MRA.WBDIS bit in any transfer information is 1, set the RRS bit to 0. Note that the MRA.WBDIS bit should be set to 1 when the MRC.DISPE bit is set to 1.
Like chain transfer, sequence transfer handles sequences of multiple types of data transfer. When sequence transfer is to be used, set the RRS bit to 0 so that the previous data transfer will not be repeated.

16.2.9 DTC Vector Base Register (DTCVBR)

Address(es): DTC.DTCVBR 0008 2404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	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
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DTCVBR register is used to set the base address for calculating the address to which the DTC vector is allocated. Values for the upper 4 bits (b31 to b28) cannot be written but reflect the value written to b27. The lower 10 bits are reserved and the values are fixed to 0. Write 0 to the lower 10 bits if necessary.
It can be set in the range of 0000 0000h to 07FF FC00h and F800 0000h to FFFF FC00h in 1-Kbyte units.

16.2.10 DTC Address Mode Register (DTCADMOD)

Address(es): DTC.DTCADMOD 0008 2408h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SHORT
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SHORT	Short-Address Mode Set*1	0: Full-address mode 1: Short-address mode	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Set this bit to 0 (full-address mode) when using the sequence transfer.

DTCADMOD register is used to specify the area accessible by the DTC.

SHORT Bit (Short-Address Mode Set)

This bit is used to select address mode of registers SAR and DAR.
Full-address mode allows the DTC to access to a 4-Gbyte space (0000 0000h to FFFF FFFFh).
Short-address mode allows the DTC to access to a 16-Mbyte space (0000 0000h to 007F FFFFh and FF80 0000h to FFFF FFFFh).

16.2.11 DTC Module Start Register (DTCST)

Address(es): DTC.DTCST 0008 240Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DTCST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DTCST	DTC Module Start	0: DTC module stop 1: DTC module start	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DTCST Bit (DTC Module Start)

Set the DTCST bit to 1 to enable the DTC to accept transfer requests. When this bit is set to 0, transfer requests are no longer accepted.

If this bit is set to 0 during data transfer, the accepted transfer request is active until the processing is completed.

Set the DTCST bit to 0 before making a transition to the module stop state, deep sleep mode, or software standby mode.

Set the DTCST bit to 1 to resume the data transfer after returning from the module stop state, deep sleep mode, or software standby mode.

For details on transitions to the module stop state, deep sleep mode, and software standby mode, refer to section 16.8, Low Power Consumption Function, and section 11, Low Power Consumption.

16.2.12 DTC Status Register (DTCSTS)

Address(es): DTC.DTCSTS 0008 240Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ACT	—	—	—	—	—	—	—	VECN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	VECN[7:0]	DTC Active Vector Number Monitoring Flag	These bits indicate the vector number for the request source when data transfer is in progress. The value is only valid if data transfer is in progress (the value of the ACT flag is 1).	R
b14 to b8	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b15	ACT	DTC Active Flag	0: Data transfer is not in progress. 1: Data transfer is in progress.	R

VECN[7:0] Flags (DTC Active Vector Number Monitoring Flag)

While data transfer is in progress, these bits indicate the vector number corresponding to the request source for the transfer.

When the DTCSTS register is read, the value of the VECN[7:0] flags is valid if the value of the ACT flag was 1 (data transfer is in progress) and invalid if the value of the ACT flag was 0 (data transfer is not in progress).

For the correspondence between the DTC request sources and the vector addresses, refer to section 14.3.1, Interrupt Vector Table in section 14, Interrupt Controller (ICUb).

ACT Flag (DTC Active Flag)

This flag indicates the state of data transfer operation.

[Setting condition]

- When the data transfer is started by a transfer request.
- When the sequence transfer is resumed.

[Clearing condition]

- When the data transfer is completed in response to a transfer request.
- When the sequence transfer is suspended.

16.2.13 DTC Index Table Base Register (DTCIBR)

Address(es): DTC.DTCIBR 0008 2410h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	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
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DTCIBR register is used to set the base address for calculating the address to which the DTC index is allocated.
Values for the upper 4 bits (b31 to b28) cannot be written but reflect the value written to b27.
The lower 10 bits (b9 to b0) are reserved bits and fixed to 0. When writing this register, set these bits to 0.
It can be set in the range of 0000 0000h to 07FF FC00h and F800 0000h to FFFF FC00h in 1-Kbyte units.

16.2.14 DTC Operation Register (DTCOR)

Address(es): DTC.DTCOR 0008 2414h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SQTFRL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SQTFRL	Sequence Transfer Terminate	Writing 1 to this bit terminates the sequence transfer in progress. This bit is read as 0.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R

The DTCOR register sets the operation of the DTC module.

SQTFRL Bit (Sequence Transfer Terminate)

Setting the SQTFRL bit to 1 terminates the sequence transfer in progress.

When the DTCSQE.ESPSEL bit is 1 (Sequence transfer is enabled), follow the procedure shown in Figure 16.2 to terminate the sequence transfer.

Writing 1 to the bit, while no sequence transfer is performed, have no effect.

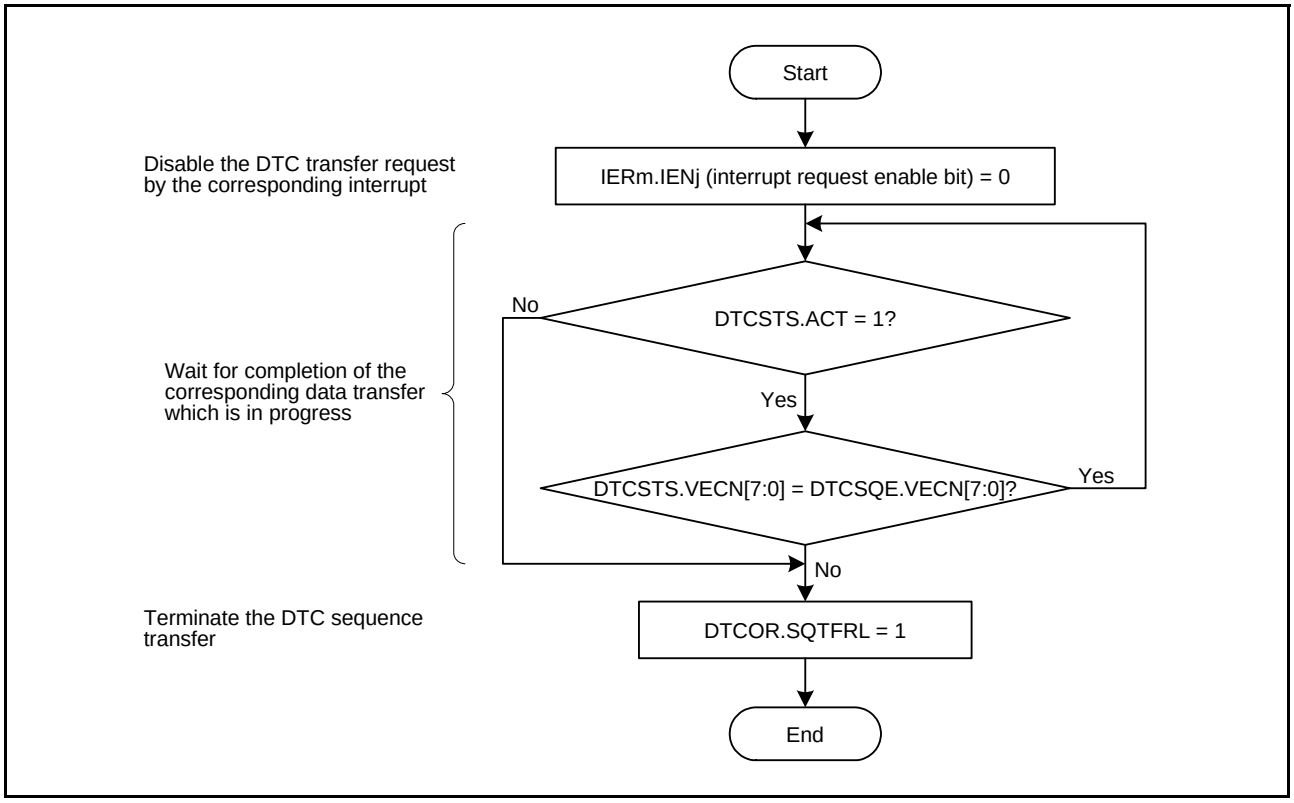


Figure 16.2 Procedure to Terminate Sequence Transfer

16.2.15 DTC Sequence Transfer Enable Register (DTCSQE)

Address(es): DTC.DTCSQE 0008 2416h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ESPSEL	—	—	—	—	—	—	—	VECN[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	VECN[7:0]	Sequence Transfer Vector Number Setting	Specify the vector number by which a sequence transfer is enabled. The value is only valid when the ESPSEL bit is 1.	R/W
b14 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R
b15	ESPSEL	Sequence Transfer Enable	0: Sequence transfer is disabled. 1: Sequence transfer is enabled.	R/W

The DTCSQE register is used to specify sequence transfer. Follow Figure 16.24 for details on the setting procedure.

VECN[7:0] Bit (Sequence Transfer Vector Number Setting)

This bit is used to specify for which vector number to perform sequence transfer. Sequence transfer can occur only for this trigger source.

section 14.3.1, Interrupt Vector Table in section 14, Interrupt Controller (ICUb) shows the relationship between the trigger source and the vector number.

ESPSEL Bit (Sequence Transfer Enable)

The ESPSEL bit specifies whether sequence transfer is used.

Set the DTCADMOD.SHORT bit to 0 (full address mode), when setting the ESPSEL bit to 1.

16.2.16 DTC Address Displacement Register (DTCDISP)

Address(es): DTC.DTCDISP 0008 2418h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	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
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DTCDISP register is used to specify the displacement value to add to the DTC transfer source address.

If MRC.DISPE bit is 1, the value SAR + DTCDISP is used as the transfer source address.

16.3 Request Sources

The DTC data transfer is triggered by an interrupt request. Setting the ICU.DTCERn.DTCE bit (n = interrupt vector number) to 1 selects the corresponding interrupt request as a request source for the DTC.
For the correspondence between the DTC request sources and the vector addresses, refer to section 14.3.1, Interrupt Vector Table in section 14, Interrupt Controller (ICUb). For request by software, refer to section 14.2.5, Software Interrupt Generation Register (SWINTR) in section 14, Interrupt Controller (ICUb).
Once the DTC has accepted a transfer request, it does not accept another transfer request until transfer for that single request is completed, regardless of the priority of the requests.
When multiple transfer requests are generated during data transfer by the DTC, the request with the highest priority on completion of the current transfer is accepted. When multiple transfer requests are generated while the DTCST.DTCST bit is 0 (DTC module stop), the request with the highest priority at the moment when the bit is subsequently set to 1 (DTC module start) is accepted.

The DTC performs the following operations at the start of a single data transfer (or the last of the consecutive transfers in the case of a chain transfer).

- On completion of a specified number of data transfer, the ICU.DTCERn.DTCE bit is set to 0 and an interrupt is requested to the CPU.
- If the MRB.DISEL bit is 1, an interrupt is requested to the CPU on completion of data transfer.
- For the other transfers, the interrupt status flag of the request source is set to 0 at the start of data transfer.

16.3.1 Allocating Transfer Information and DTC Vector Table

The DTC reads the start address of the transfer information corresponding to each request source from the vector table and reads the transfer information starting at that address.
The vector table should be located so that the lower 10 bits of the base address (start address) are 0. Use the DTC vector base register (DTCVBR) to set the base address of the DTC vector table.
Transfer information is allocated in the RAM area. Transfer information can be allocated in the ROM area when the MRA.WBDIS bit is set to 1. The start address of the transfer information n with vector number n should be allocated at DTCVBR + 4n.
Transfer information should be aligned on a 4-byte boundary. The size of a transfer information is 12 bytes in short-address mode or 16 bytes in full-address mode. Use the DTCADMOD.SHORT bit to select short-address mode (SHORT bit = 1) or full-address mode (SHORT bit = 0).
Figure 16.3 shows the relationship between the DTC vector table and transfer information.
Figure 16.4 shows the allocation of transfer information in the RAM area. The lower addresses vary according to the endian of the corresponding allocation area. For details, refer to section 16.9.2, Allocating Transfer Information.

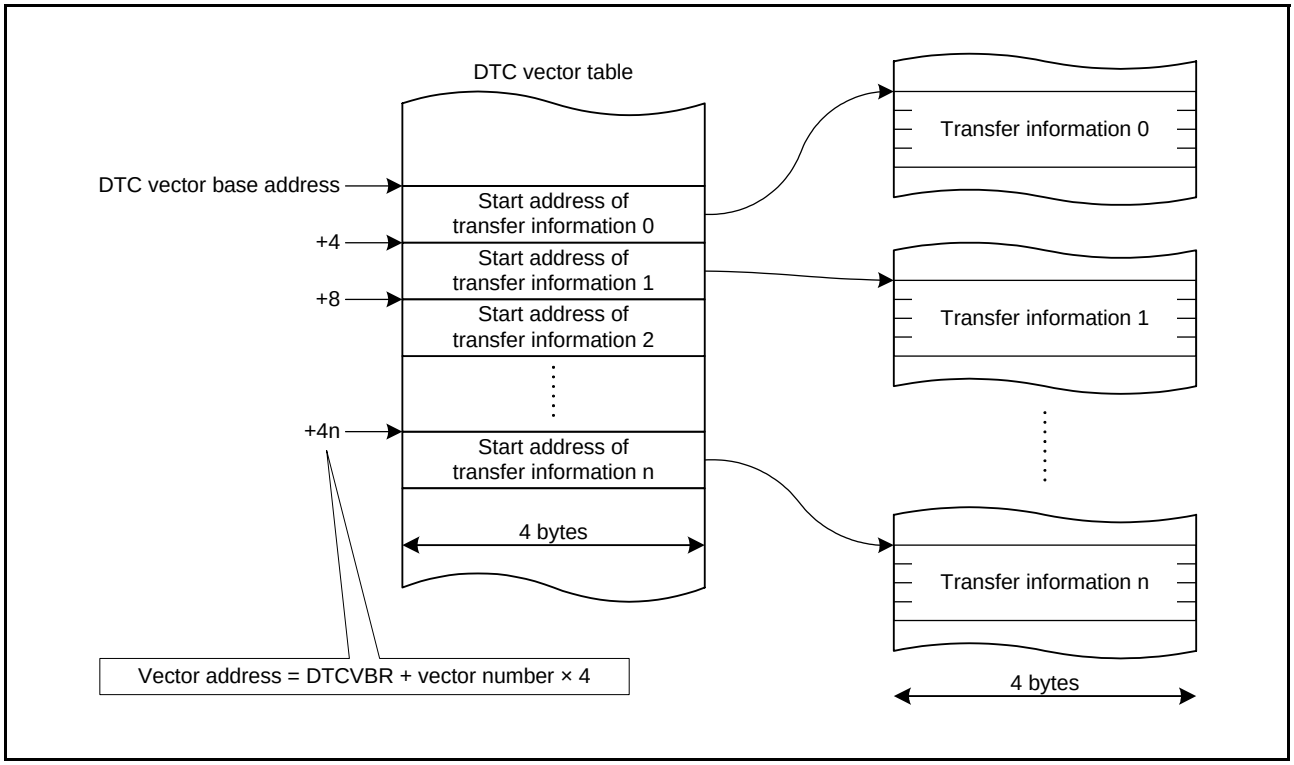


Figure 16.3 DTC Vector Table and Transfer Information

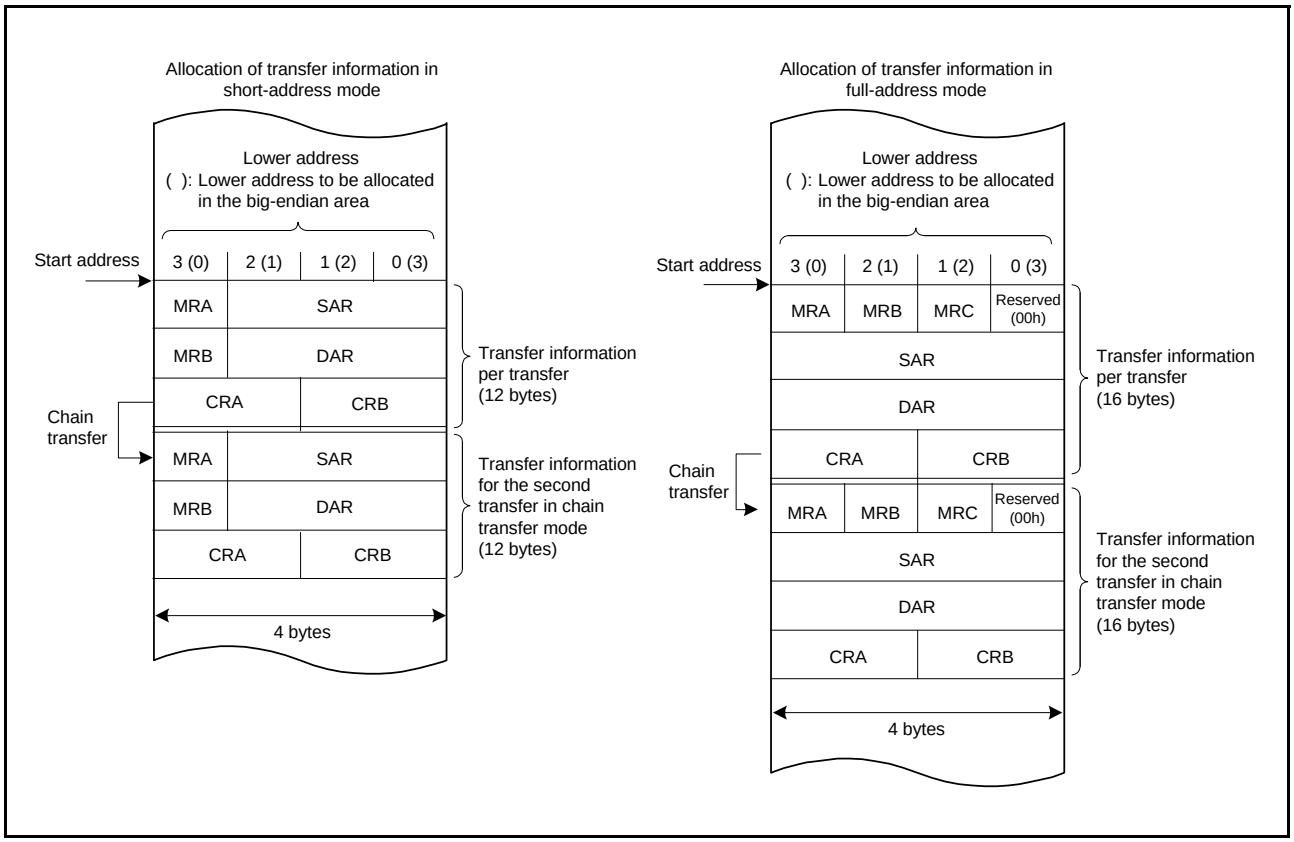


Figure 16.4 Allocation of Transfer Information in the RAM Area

16.4 Operation

The DTC transfers data in accordance with the transfer information. Storage of the transfer information in the RAM area is required before DTC operation.

When the DTC accepts a transfer request, it reads the DTC vector corresponding to the vector number. Next, the DTC reads transfer information from the address pointed by the DTC vector, transfers data, and then writes back the transfer information after the data transfer. Allocating transfer information in the RAM area allows data transfer of arbitrary number of channels.

There are three transfer modes: normal transfer mode, repeat transfer mode, and block transfer mode.

Set a transfer source address in the SAR register and a transfer destination address in the DAR register. The SAR and DAR registers are updated after the transfer according to the respective settings (increment, decrement, or fixed).

Table 16.3 lists transfer modes of the DTC.

Table 16.3 Transfer Modes of the DTC

Transfer Mode	Data Size Transferred on Single Transfer Request	Increment/Decrement of Memory Address	Settable Transfer Count
Normal transfer mode	1 byte/1 word/1 longword	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 65536
Repeat transfer mode*1	1 byte/1 word/1 longword	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 256*3
Block transfer mode*2	Block size specified in CRAH (1 to 256 bytes/1 to 256 words/1 to 256 longwords)	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 65536

Note 1. Set transfer source or transfer destination in the repeat area.
Note 2. Set transfer source or transfer destination in the block area.
Note 3. After data transfer of the specified count, the initial state is restored and the operation is continued (repeated).

Setting the MRB.CHNE bit to 1 allows multiple transfers (chain transfer) on a single transfer request. The setting in combination with the MRB.CHNS bit enables a chain transfer when the specified number of data transfers is completed. Figure 16.5 shows the operation flowchart of the DTC. Table 16.4 lists chain transfer conditions.

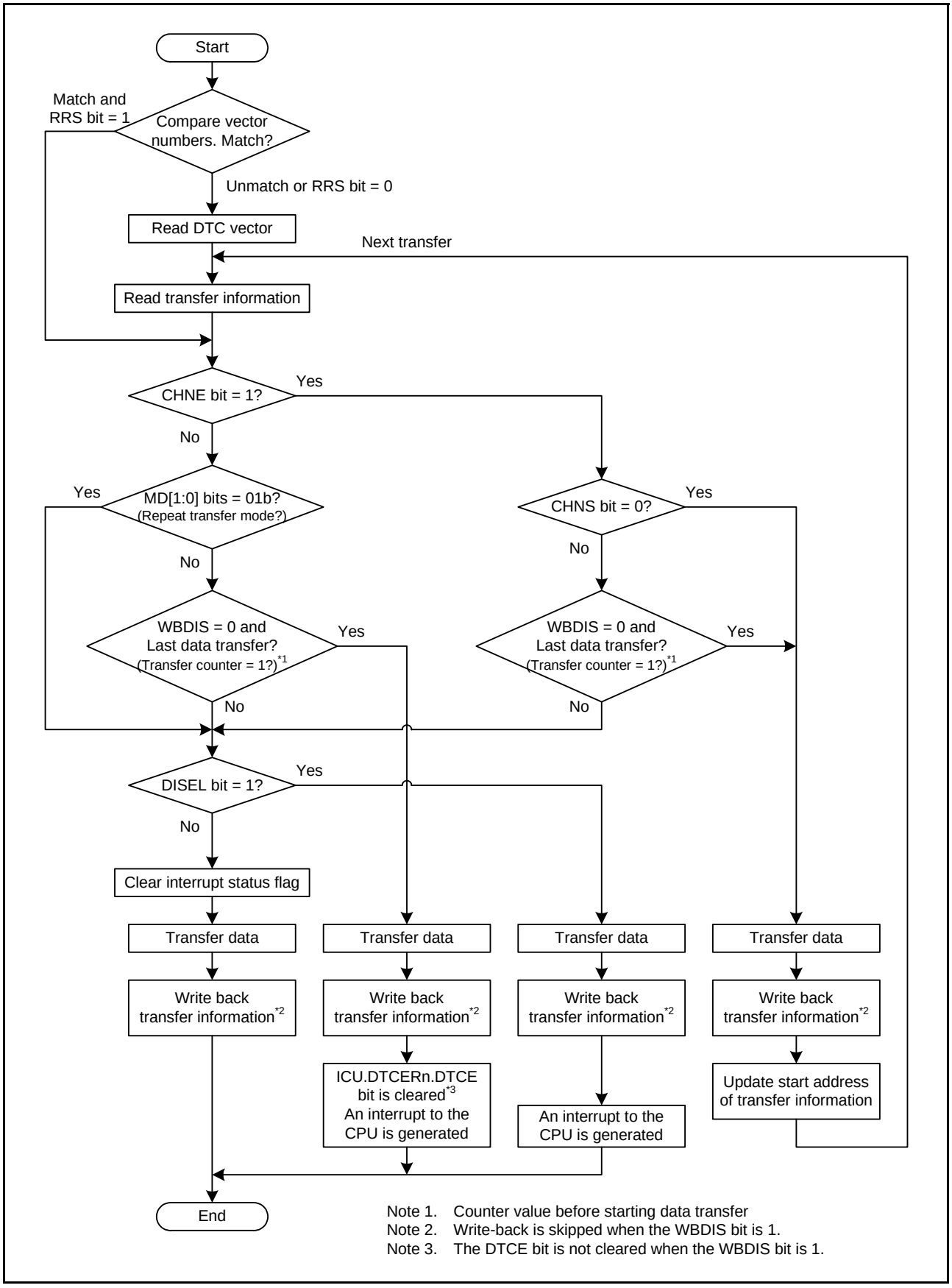


Figure 16.5 Operation Flowchart of the DTC

Table 16.4 Chain Transfer Conditions

First Transfer				Second Transfer*3				
CHNE Bit	CHNS Bit	DISEL Bit	Transfer Counter*1,*2	CHNE Bit	CHNS Bit	DISEL Bit	Transfer Counter*1,*2	Data Transfer
0	—	0	Other than (1 → 0)	—	—	—	—	Ends after the first transfer
0	—	0	(1 → 0)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU
0	—	1	—	—	—	—	—	
1	0	—	—	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	0	Other than (1 → ∗)	—	—	—	—	Ends after the first transfer
1	1	—	(1 → ∗)	0	—	0	Other than (1 → 0)	Ends after the second transfer
				0	—	0	(1 → 0)	Ends after the second transfer with an interrupt request to the CPU
				0	—	1	—	
1	1	1	Other than (1 → ∗)	—	—	—	—	Ends after the first transfer with an interrupt request to the CPU

Note 1. The transfer counters used depend on transfer modes as follows:
Normal transfer mode: CRA register
Repeat transfer mode: CRAL register
Block transfer mode: CRB register

Note 2. On completion of data transfer, the counters operate as follows:
1 → 0 in normal and block transfer modes
1 → CRAH in repeat transfer mode
(1 → ∗) in the table indicates both of the two operations above.

Note 3. Chain transfer can be selected for the second or subsequent transfers. The condition combination of “second transfer and the CHNE bit is 1” is omitted.

16.4.1 Transfer Information Read Skip Function

Reading of DTC vector and transfer information can be skipped by the setting of the DTCCR.RRS bit.

When a DTC transfer request is accepted, the current DTC vector number is compared with the DTC vector number in the previous data transfer. When these vector numbers match and the RRS bit is 1, the DTC does not read the DTC vector and transfer information, and transfers data according to the transfer information remained in the DTC.

However, when the previous transfer was chain transfer, the DTC vector and transfer information are read. Furthermore, when the transfer counter (CRA register) became 0 during the previous normal transfer and when the transfer counter (CRB register) became 0 during the previous block transfer, transfer information is read regardless of the value of the RRS bit. Figure 16.14 shows an example of transfer information read skip.

When updating the vector table and transfer information, set the RRS bit to 0, update the vector table and transfer information, and then set the RRS bit to 1. Setting the RRS bit to 0 discards the vector numbers retained in the DTC. The updated DTC vector table and transfer information are read in the next data transfer.

16.4.2 Transfer Information Write-Back Skip Function

16.4.2.1 Write-Back Skip by Fixing Addresses

When the MRA.SM[1:0] bits or the MRB.DM[1:0] bits are set to “address is fixed” (00b or 01b), a part of transfer information is not written back. This function is performed independently of the setting of short-address mode or full-address mode.

Table 16.5 lists transfer information write-back skip conditions and applicable registers. The CRA and CRB registers are written back independently of the setting of short-address mode or full-address mode.

Furthermore, in full-address mode, write-back of registers MRA, MRB, and MRC is skipped.

Table 16.5 Transfer Information Write-Back Skip Conditions and Applicable Registers

MRA.SM[1:0] Bits		MRB.DM[1:0] Bits			
b3	b2	b3	b2	**SAR Register**	**DAR Register**
0	0	0	0	Skip	Skip
0	0	0	1		
0	1	0	0		
0	1	0	1		
0	0	1	0	Skip	Write-back
0	0	1	1		
0	1	1	0		
0	1	1	1		
1	0	0	0	Write-back	Skip
1	0	0	1		
1	1	0	0		
1	1	0	1		
1	0	1	0	Write-back	Write-back
1	0	1	1		
1	1	1	0		
1	1	1	1		

16.4.2.2 Write-Back Skip by the MRA.WBDIS Bit

When the MRA.WBDIS bit is 1, the transfer information (SAR, DAR, CRA, and CRB) is not written back regardless of the settings of the transfer information.

The transfer information on the memory is not updated, data can be transferred by the DTC without copying the transfer information from ROM to RAM. Skipping a write-back reduces time for post-processing of the data transfer.

16.4.3 Normal Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single transfer request. The transfer count can be set to 1 to 65536.

Transfer source addresses and transfer destination addresses can be set to increment, decrement, or fixed independently.

This mode enables an interrupt request to the CPU to be generated at the end of specified-count transfer.

Table 16.6 lists register functions in normal transfer mode, and Figure 16.6 shows the memory map of normal transfer mode.

Table 16.6 Register Functions in Normal Transfer Mode

Register	Description	Value Written Back by Writing Transfer Information*1
SAR	Transfer source address	Increment/decrement/fixed*2
DAR	Transfer destination address	Increment/decrement/fixed*2
CRA	Transfer counter A	CRA – 1
CRB	Transfer counter B	Not updated

Note 1. Write-back operation is skipped when the MRA.WBDIS bit is 1.
Note 2. Write-back operation is skipped when address is fixed.

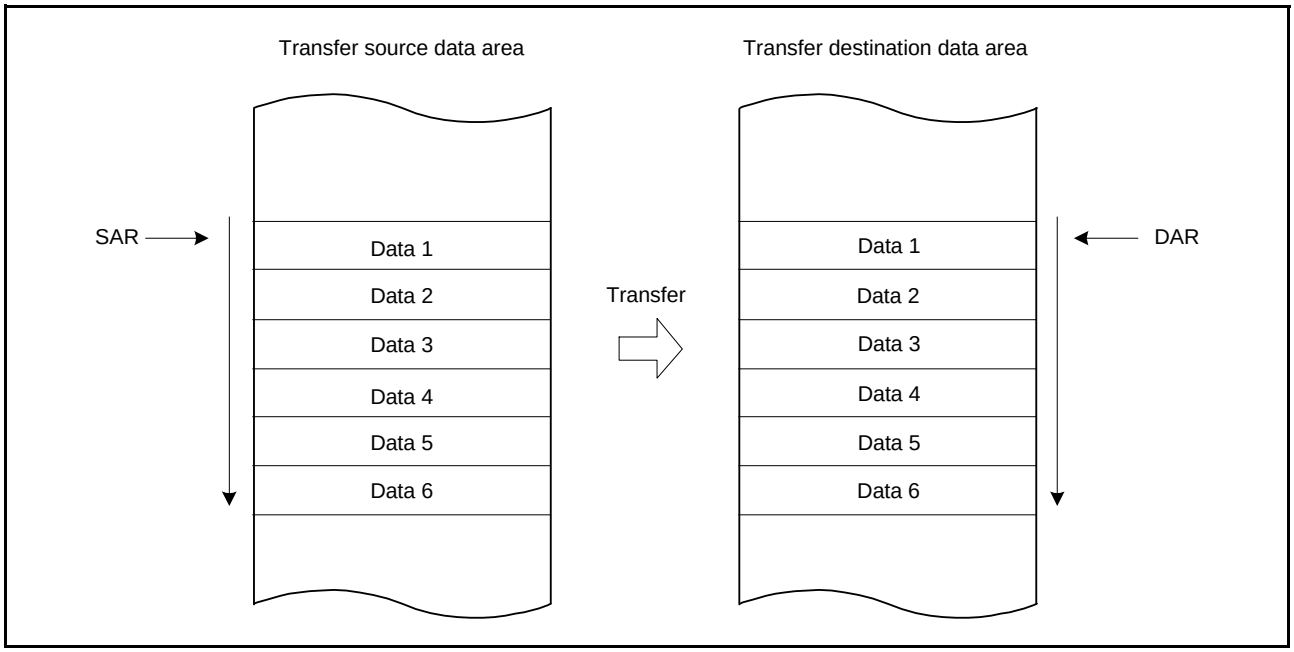


Figure 16.6 Memory Map of Normal Transfer Mode

16.4.4 Repeat Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single transfer request.
Specify either transfer source or transfer destination for the repeat area by the MRB.DTS bit. The transfer count can be set to 1 to 256. When the specified-count transfer is completed, the initial value of the address register specified in the transfer counter and the repeat area is restored and transfer is repeated. The other address register is incremented or decremented continuously or remains unchanged.
When the transfer counter CRAL is decreased to 00h in repeat transfer mode, the CRAL value is updated to the value set in the CRAH register. Thus the transfer counter does not become 00h, which disables an interrupt request to be generated to the CPU when the MRB.DISEL bit is set to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers).
Table 16.7 lists the register functions in repeat transfer mode, and Figure 16.7 shows the memory map of repeat transfer mode.

Table 16.7 Register Functions in Repeat Transfer Mode

		Value Written Back by Writing Transfer Information*1		
			When CRAL = 1	
Register	**Description**	**When CRAL ≠ 1**	**When the MRB.DTS Bit is 0**	**When the MRB.DTS Bit is 1**
SAR	Transfer source address	Increment/decrement/fixed*2	Increment/decrement/fixed*2	SAR register initial value
DAR	Transfer destination address	Increment/decrement/fixed*2	DAR register initial value	Increment/decrement/fixed*2
CRAH	Retains initial value of transfer counter	CRAH	CRAH	
CRAL	Transfer counter A	CRAL – 1	CRAH	
CRB	Transfer counter B	Not updated	Not updated	

Note 1. Write-back operation is skipped when the MRA.WBDIS bit is 1.
Note 2. Write-back operation is skipped when address is fixed.

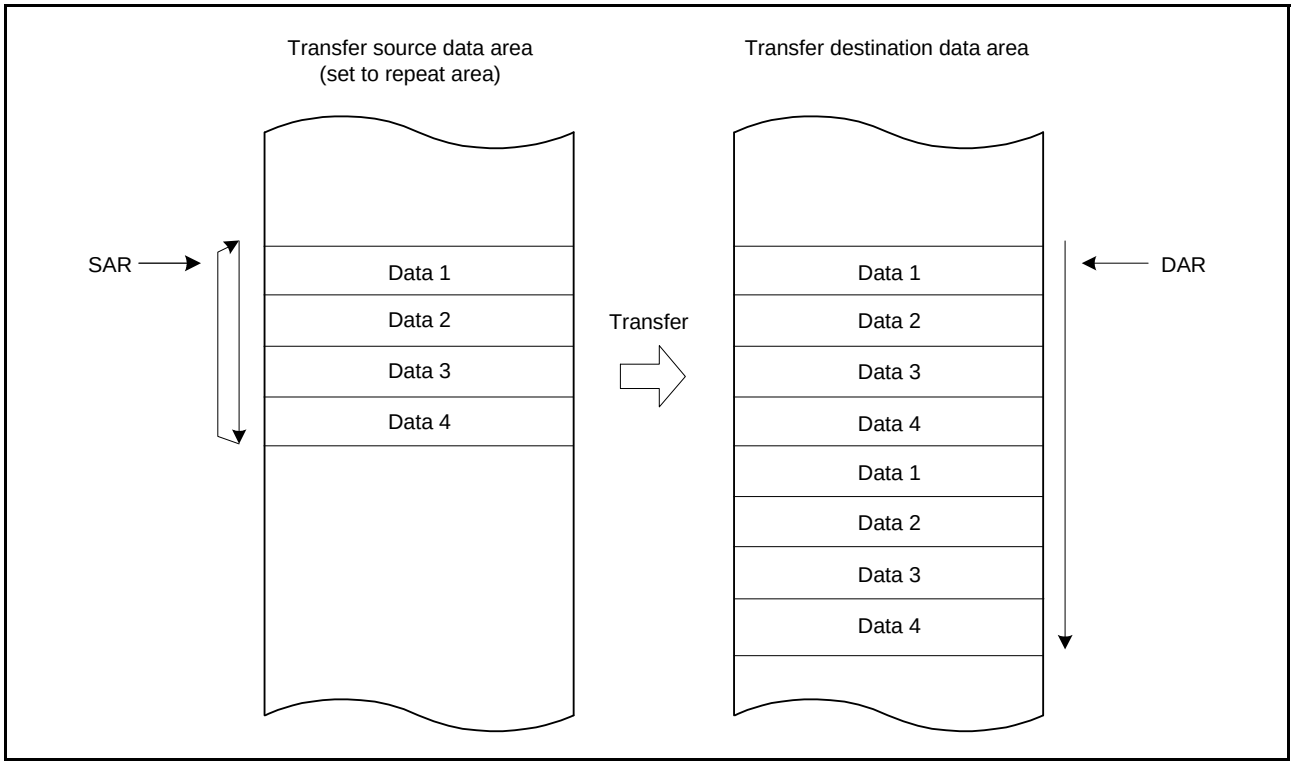


Figure 16.7 Memory Map of Repeat Transfer Mode (Transfer Source: Repeat Area)

16.4.5 Block Transfer Mode

This mode allows single-block data transfer on a single transfer request.
Specify either transfer source or transfer destination for the block area by the MRB.DTS bit. The block size can be set to 1 to 256 bytes, 1 to 256 words, or 1 to 256 longwords.
When transfer of the specified one block is completed, the initial values of the block size counter CRAL and the address register (the SAR register when the MRB.DTS bit is 1 or the DAR register when the DTS bit is 0) specified in the block area are restored. The other address register is incremented or decremented continuously or remains unchanged.
The transfer count (block count) can be set to 1 to 65536. This mode enables an interrupt request to the CPU to be generated at the end of specified-count block transfer.
Table 16.8 lists register functions in block transfer mode, and Figure 16.8 shows the memory map of block transfer mode.

Table 16.8 Register Functions in Block Transfer Mode

		Value Written Back by Writing Transfer Information*1	
Register	**Description**	**When MRB.DTS Bit is 0**	**When MRB.DTS Bit is 1**
SAR	Transfer source address	Increment/decrement/fixed*2	SAR register initial value
DAR	Transfer destination address	DAR register initial value	Increment/decrement/fixed*2
CRAH	Retains initial value of block size	CRAH	
CRAL	Block size counter	CRAH	
CRB	Block transfer counter	CRB – 1	

Note 1. Write-back operation is skipped when the MRA.WBDIS bit is 1.
Note 2. Write-back operation is skipped when address is fixed.

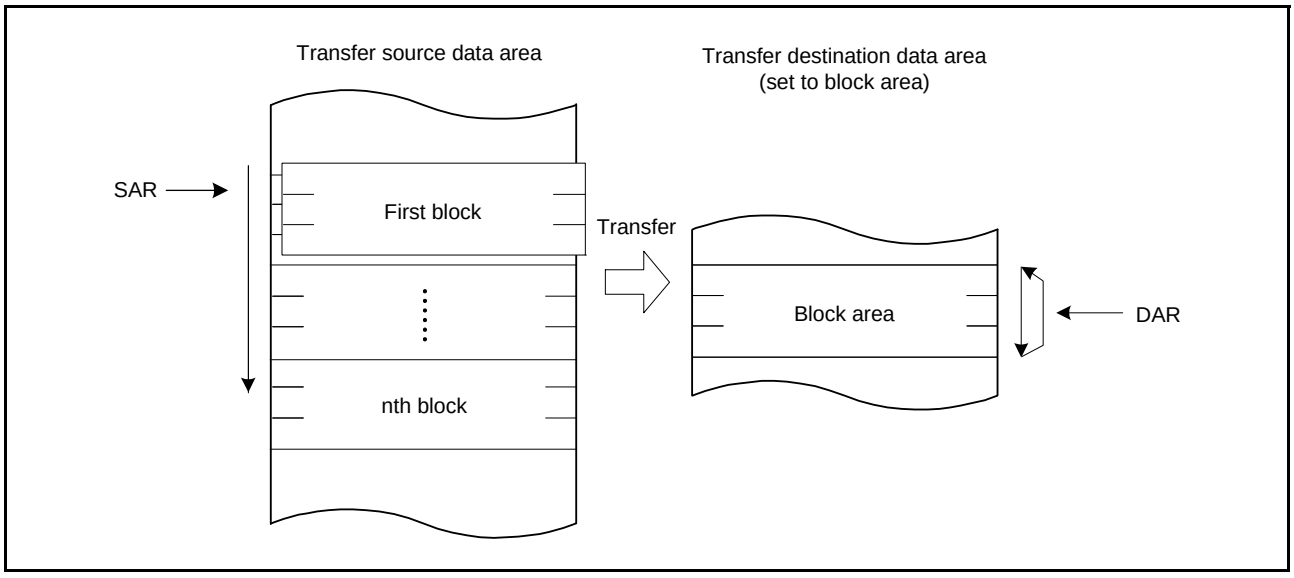


Figure 16.8 Memory Map of Block Transfer Mode (Transfer Destination: Block Area)

16.4.6 Chain Transfer

Setting the MRB.CHNE bit to 1 allows chain transfer to be performed continuously on a single transfer request.
If the MRB.CHNE bit is 1 and the MRB.CHNS bit is 0, an interrupt request to the CPU is not generated when the specified number of data transfers is completed, or while the MRB.DISEL bit is 1 (an interrupt request to the CPU is generated for every data transfer). Data transfer has no effect on the interrupt status flag, which is the request source.
The transfer information (SAR, DAR, CRA, CRB, MRA, MRB, and MRC) that define a data transfer can be specified independently of each other. Figure 16.9 shows chain transfer operation.

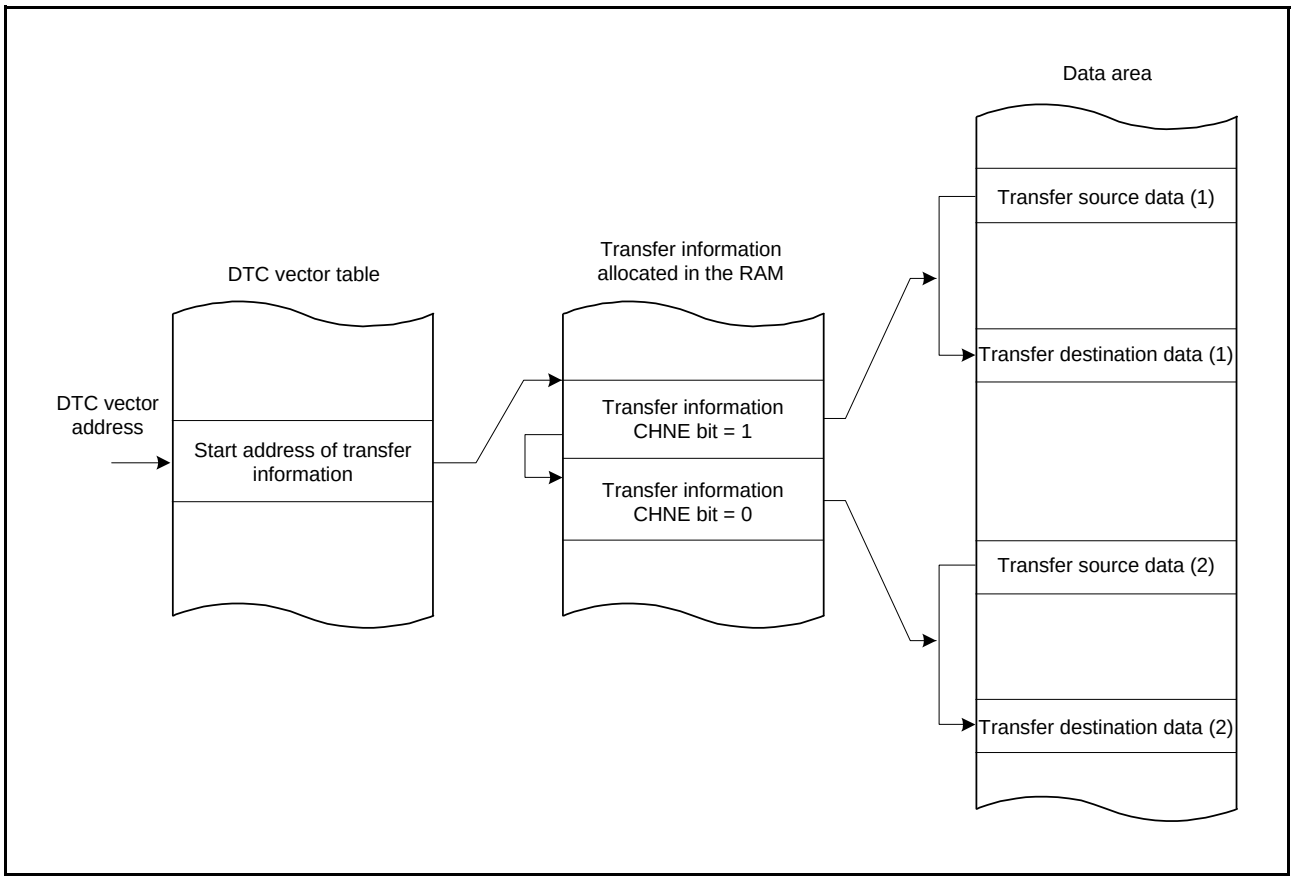


Figure 16.9 Chain Transfer Operation

If the MRB.CHNE bit is 1 and the CHNS bit is 1, chain transfer is performed only after completion of specified number of data transfers. In repeat transfer mode, chain transfer is performed after completion of specified number of data transfers.
For details on chain transfer conditions, refer to Table 16.4, Chain Transfer Conditions.

16.4.7 Operation Timing

Figure 16.10 to Figure 16.14 show examples of DTC operation timing.

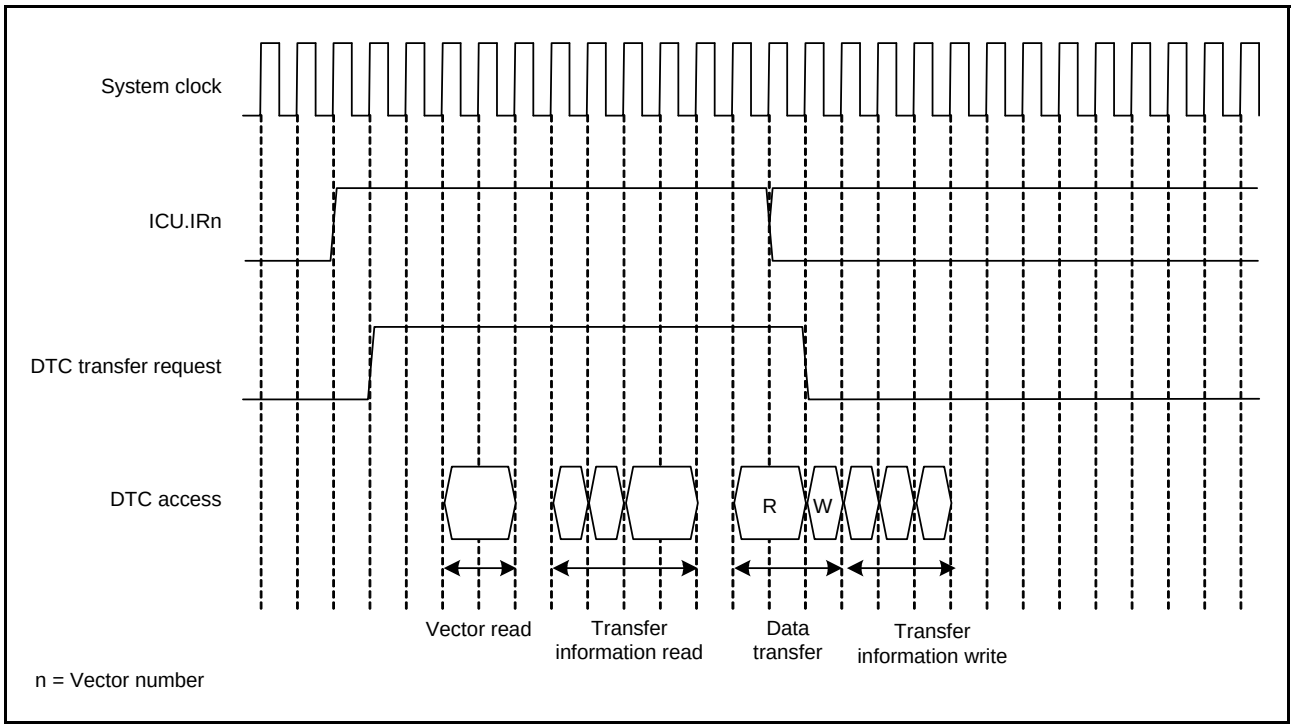


Figure 16.10 Example (1) of DTC Operation Timing (Short-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)

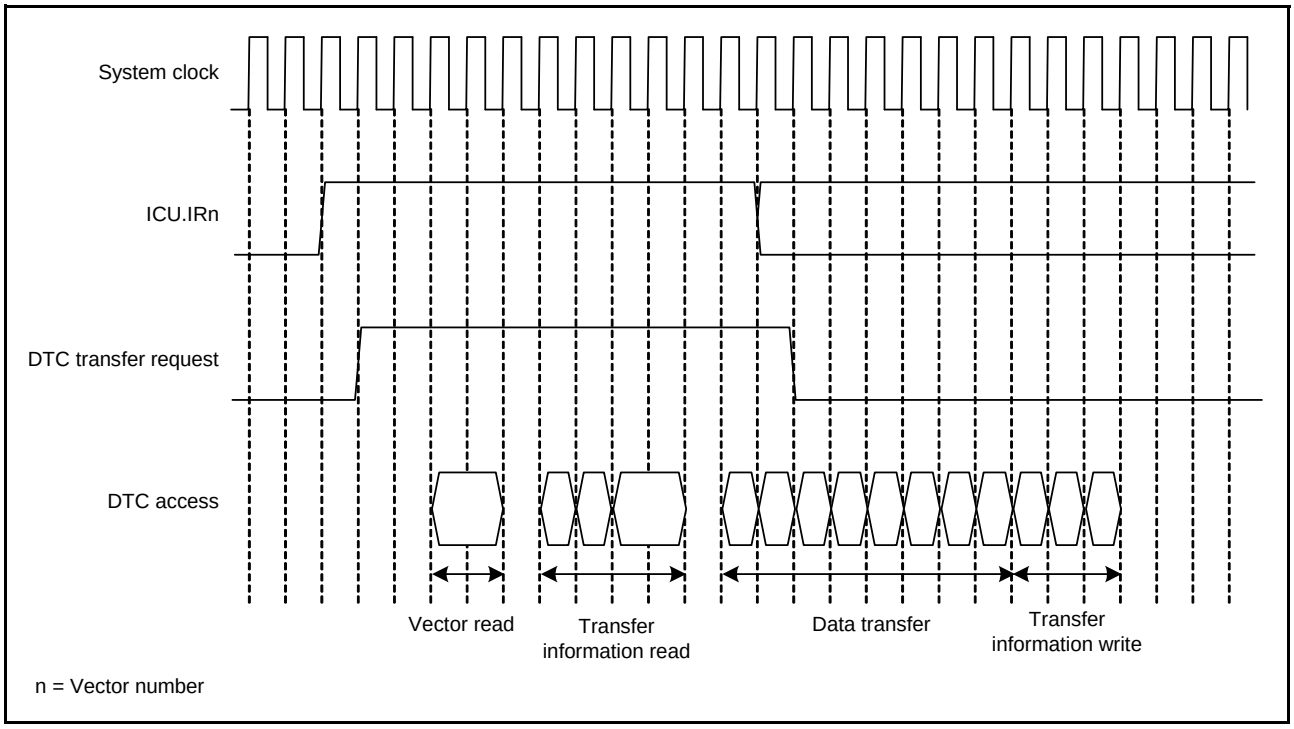


Figure 16.11 Example (2) of DTC Operation Timing (Short-Address Mode, Block Transfer Mode, Block Size = 4)

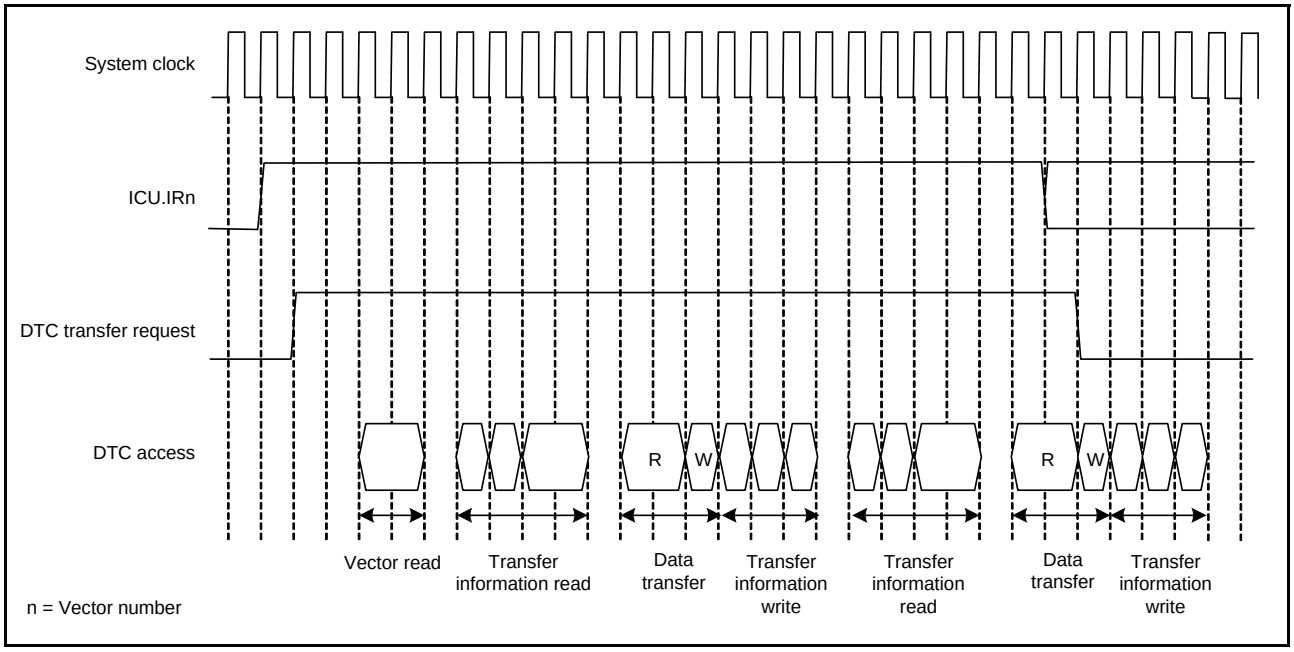


Figure 16.12 Example (3) of DTC Operation Timing (Short-Address Mode, Chain Transfer)

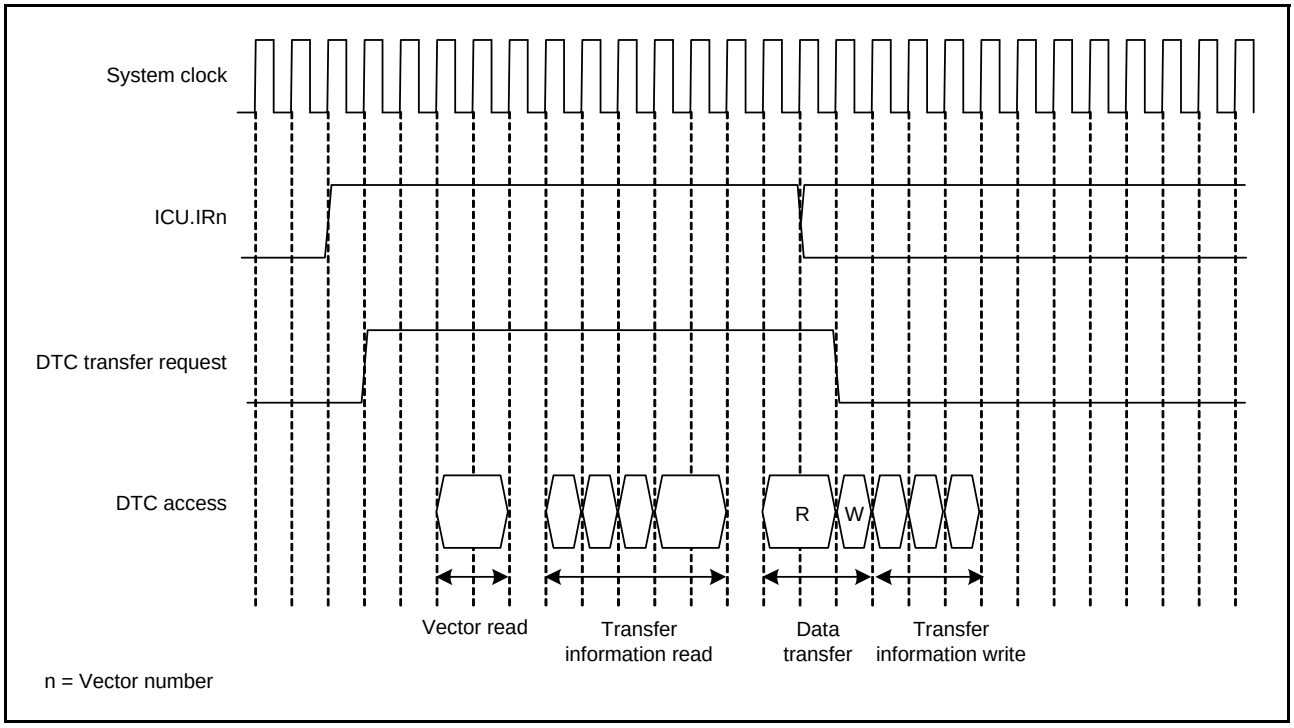
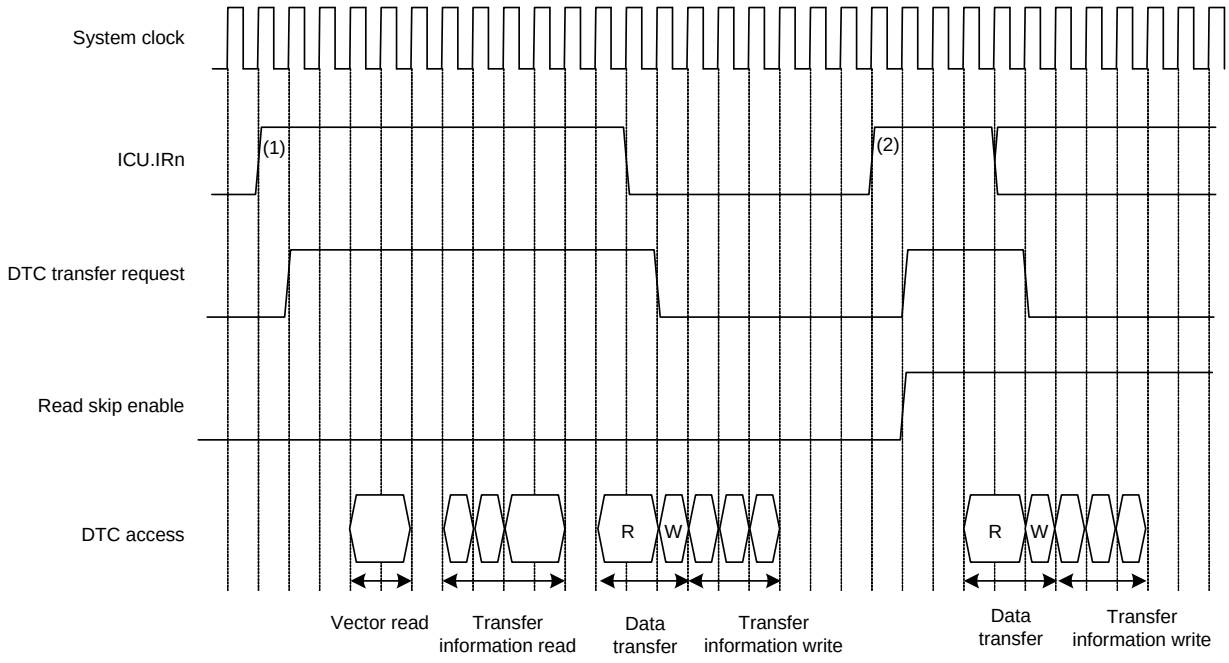


Figure 16.13 Example (4) of DTC Operation Timing (Full-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)



n = Vector number

Note: When request sources (vector numbers) of (1) and (2) are the same and the DTCCR.RRS bit is 1, the transfer information read for request (2) is skipped.

Figure 16.14 Example of Operation When Transfer Information Read Skip is Executed (Vector, Transfer Information, and Transfer Destination Data on the RAM, and Transfer Source Data on the Peripheral Module)

16.4.8 Execution Cycles of the DTC

Table 16.9 lists the execution cycles of single data transfer of the DTC.
For the order of the execution states, refer to section 16.4.7, Operation Timing.

Table 16.9 Execution Cycles of the DTC

Transfer Mode	Vector Read		Transfer Information Read			Transfer Information Write			Data Transfer		Internal Operation	
									Read	Write		
Normal	Cv + 1	0*1	4 × Ci + 1*2	3 × Ci + 1*3	0*1	3 × Ci*4	2 × Ci*5	Ci*6	Cr + 1	Cw	2	0*1
Repeat									Cr + 1	Cw		
Block*7									P × Cr	P × Cw		

Note 1. When transfer information read is skipped
Note 2. In full-address mode
Note 3. In short-address mode
Note 4. When neither SAR nor DAR is set to address-fixed
Note 5. When SAR or DAR is set to address-fixed
Note 6. When SAR and DAR are set to address-fixed
Note 7. When the block size is 2 or more. If the block size is 1, the cycle number for normal transfer is applied.

P: Block size (initial settings of CRAH and CRAL)
Cv: Cycles for access to vector transfer information storage destination
Ci: Cycles for access to transfer information storage destination address
Cr: Cycles for access to data read destination
Cw: Cycles for access to data write destination
(The unit is system clocks (ICLK) for “+ 1” in the Vector Read, Transfer Information Read, and Data Transfer Read columns and “2” in the Internal Operation column.)
(Cv, Ci, Cr, and Cw vary depending on the corresponding access destination. For the number of cycles for respective access destinations, refer to section 30, RAM, section 31, Flash Memory (FLASH), and section 5, I/O Registers.)

16.4.9 DTC Bus Mastership Release Timing

The DTC does not release the bus mastership during transfer information read and transfer information write. While transfer information is not read or written, bus arbitration is made according to the priority determined by the bus master arbitrator.
For bus arbitration, refer to section 15, Buses.

16.4.10 Sequence Transfer

A sequence transfer can be executed on the request source that is specified in the DTCSQE register. A sequence transfer is started by setting the MRB.INDX bit to 1 and ended by setting the MRB.SQEND bit to 1. Setting the DTCOR.SQTFRL bit to 1, even during a sequence transfer, forcibly ends the transfer and the next DTC transfer request starts new sequence transfer with reference to the index table.
A sequence transfer includes the following processing.

(1) The first data transfer is executed by referring to the DTC vector table in response to a DTC transfer request from the source specified in the DTCSQE register.
(2) The DTC index table is referred based on the value of the lower 8 bits of the first data transferred data in (1) (sequence number).
(3) The transfer information is read from the address obtained from the DTC index table.
(4) A data transfer is executed in accordance with the transfer information. On completion of the data transfer, either one of the following operations is performed by using the values of bits MRB.CHNE and MRB.SQEND.
- A chain transfer is executed when the CHNE bit is 1. The next transfer information is read. Go to (4).
- The sequence transfer is suspended when the CHNE bit is 0 and the SQEND bit is 0. Go to (5).
- The sequence transfer ends when the CHNE bit is 0 and the SQEND bit is 1.

(5) When a DTC transfer request from the source specified in the DTCSQE register is accepted, the suspended sequence transfer is resumed and the next transfer information is read. Go to (4).

Note 1. When the ICU.DTCERn.DTCE bit becomes 0 based on the result of the data transfer, a DTC transfer request is not generated. Set the DTCE bit to 1 to resume sequence transfer. Refer to Figure 16.5 or section 14, Interrupt Controller (ICUb) for the conditions where the DTCE bit becomes 1.

Figure 16.15 and Figure 16.16 shows a basic sequence transfer operation.

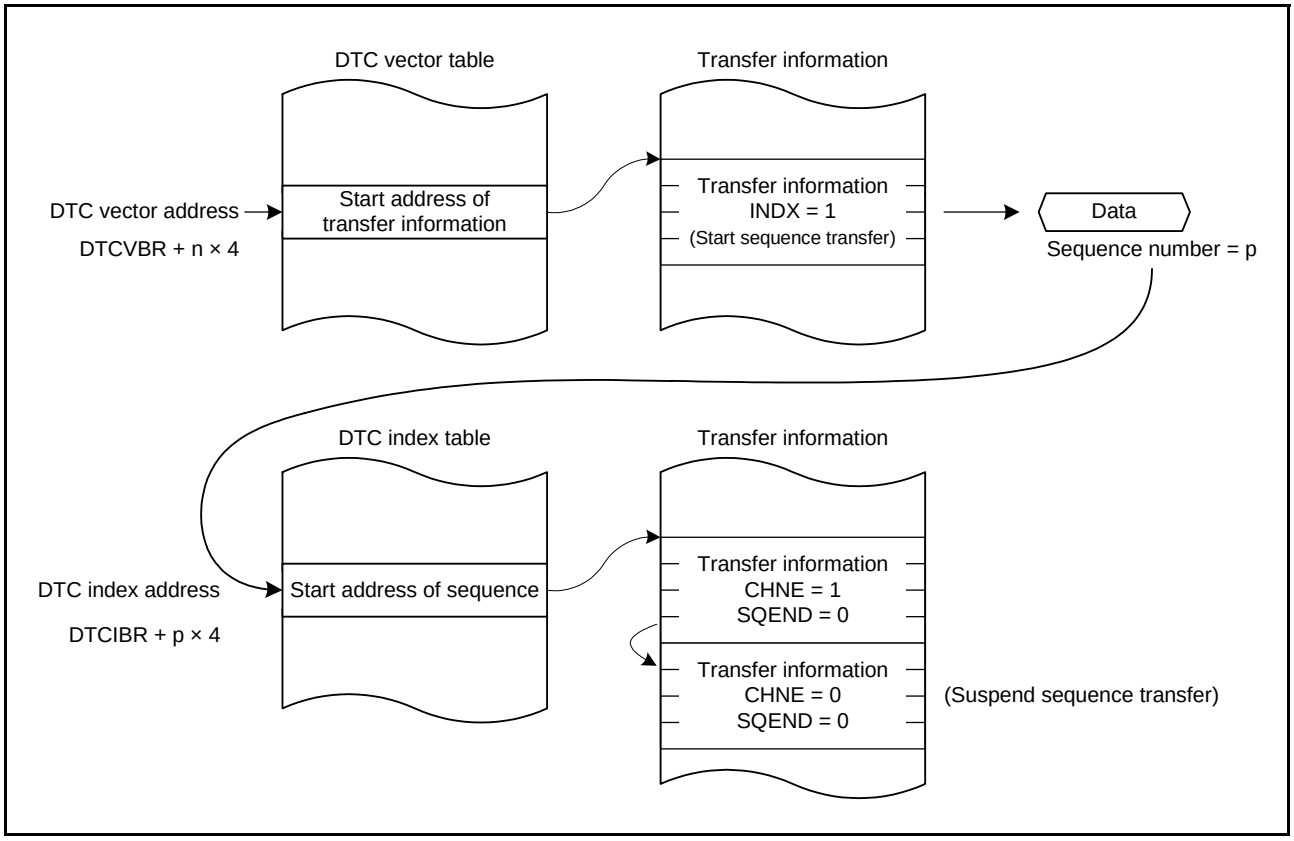


Figure 16.15 Start and Suspension of Sequence Transfer

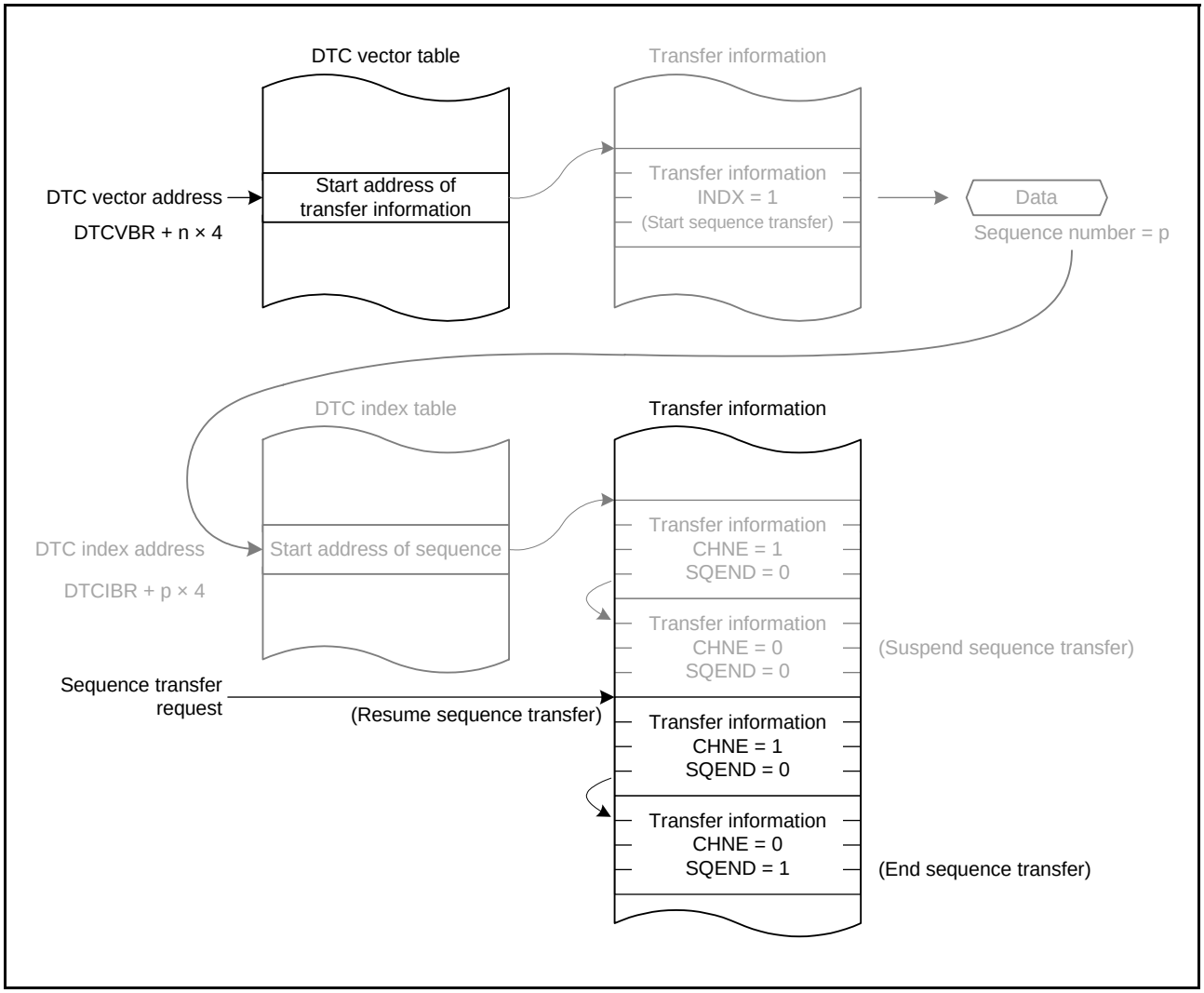


Figure 16.16 Resumption and End of Sequence Transfer

Table 16.10 lists the settings of bits CHNE, SQEND, and INDX during the sequence transfer.

Table 16.10 Sequence Transfer Process and Values of Bits CHNE, SQEND, and INDX

DTC Operations	CHNE Bit	SQEND Bit	INDX Bit
Start sequence transfer	0	0	1*1
Continue sequence transfer	1	0	0
Suspend sequence transfer*2	0	0	0
End sequence transfer	0	1	0
End current sequence transfer and Obtain new sequence number	0	1	1*1
Some other transfer (not sequence transfer)	—	0	0

Note: Do not set the values other than listed above.
Note 1. Set MRA.MD[1:0] bits to 00b (normal transfer mode) when setting the INDX bit to 1.
Note 2. When a sequence transfer is suspended, the ICU.DTCERn.DTCE bit may become 0. Set the DTCE bit to 1 to resume sequence transfer.

Even when a sequence transfer is suspended, a new sequence transfer cannot start until the suspended sequence transfer is eventually completed. When a sequence transfer request is received during suspension of the sequence transfer, the suspended sequence transfer is resumed.

16.4.11 DTC Index Table

The DTC index table is allocated to the area where its start address is configured in the DTCIBR register.
Store the start address of transfer information table p for sequence number p in the address of DTCIBR + p × 4.
The upper 30 bits of the start address is set to the upper 30 bits of the DTC index. Set the CPUSEL bit to select either of reading the transfer information and starting the sequence, or output an interrupt request to the CPU without starting the sequence. For a complicated sequence that the DTC cannot handle, set the CPUSEL bit to 1 to allow the CPU to handle such a sequence.

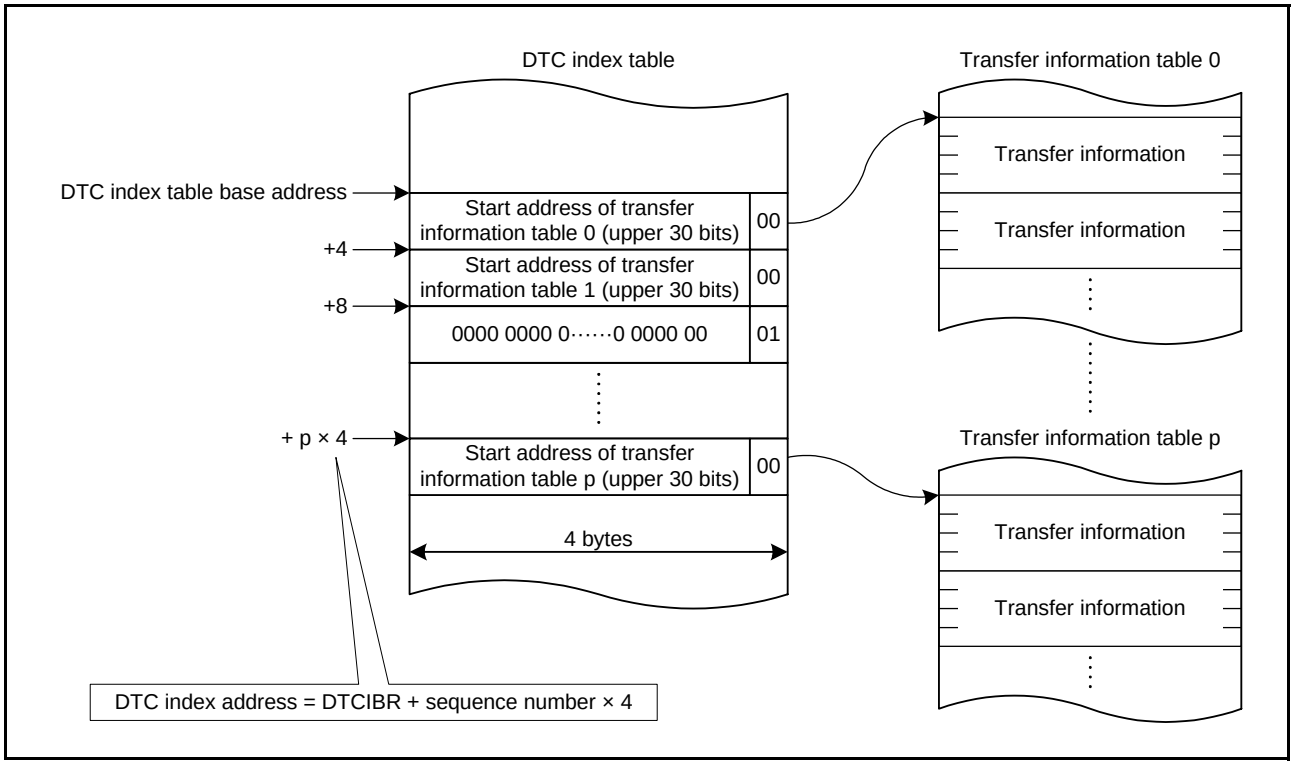
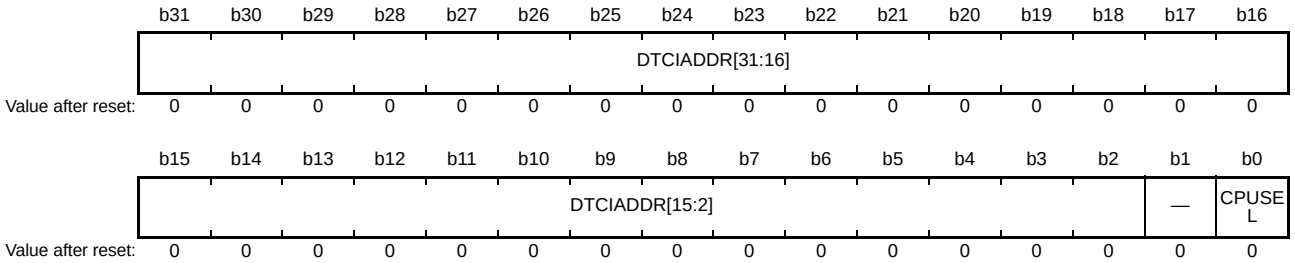


Figure 16.17 DTC Index Table

- DTC Index

Address(es): DTCIBR + p × 4



Bit	Symbol	Bit Name	Description	R/W
b0	CPUSEL	Sequence Transfer/CPU Interrupt Select	0: Continues the sequence transfer (starts the sequence) 1: Ends the sequence transfer and outputs an interrupt request to the CPU	—
b1	—	Reserved	Set this bit to 0.	—
b31 to b2	DTCIADDR[31:2]	Transfer Information Table Address	Set the upper 30 bits of the start address of the transfer information table to these bits. Writing to the upper 4 bits (b31 to b28) is ignored and the values in b31 to b28 become the same value as b27.	—

When the CPUSEL bit in the DTC index that the obtained sequence number indicates is 1, an interrupt request to the CPU is generated. At this time, the ICU.DTCERn.DTCE bit becomes 0. From this point, the interrupt request signal from the request source that is specified in the DTCSQE register is sent to the CPU, but not DTC. After completion of CPU interrupt processing, set the ICU.DTCERn.DTCE bit to 1 to enable DTC transfer request for starting the next sequence transfer.

16.4.12 Example of Sequence Transfer

Figure 16.18 shows a typical examples of a sequence transfer and Figure 16.19 to Figure 16.23 show configurations of the transfer information for the examples of the transfers in the figure.
In these examples, the interrupt source of vector number n is set as the source of the sequence transfer (DTCSQE.VECN[7:0] = n).
Once the DTC transfer request due to the interrupt source of vector number n (hereinafter referred to as “transfer request n”) is input, the DTC refers to the DTC vector table and reads the corresponding transfer information. The lower 8 bits that have been transferred based on the transfer information become a sequence number, selecting one sequence among possible 256 sequences.

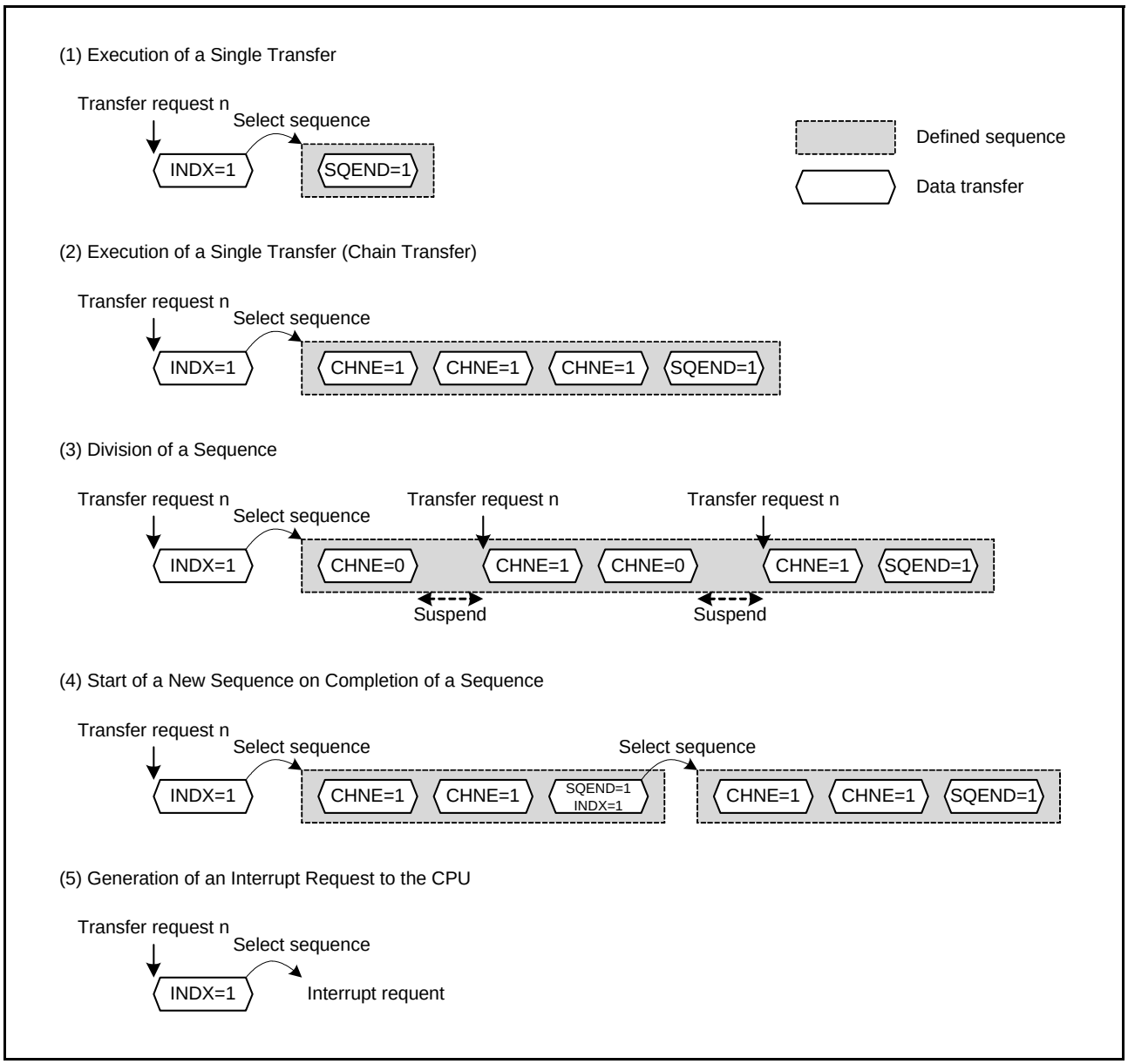


Figure 16.18 Examples of Sequence Transfers

(1) When Executing a Single Transfer

Figure 16.19 shows an example of a single transfer (normal, repeat, or block).
The DTC refers to the DTC index table, and reads the transfer information corresponding to the obtained sequence number p.
Since the values of the CHNE, INDX, and SQEND bits are 0, 0, and 1 respectively, the sequence ends after the specified transfer is executed.

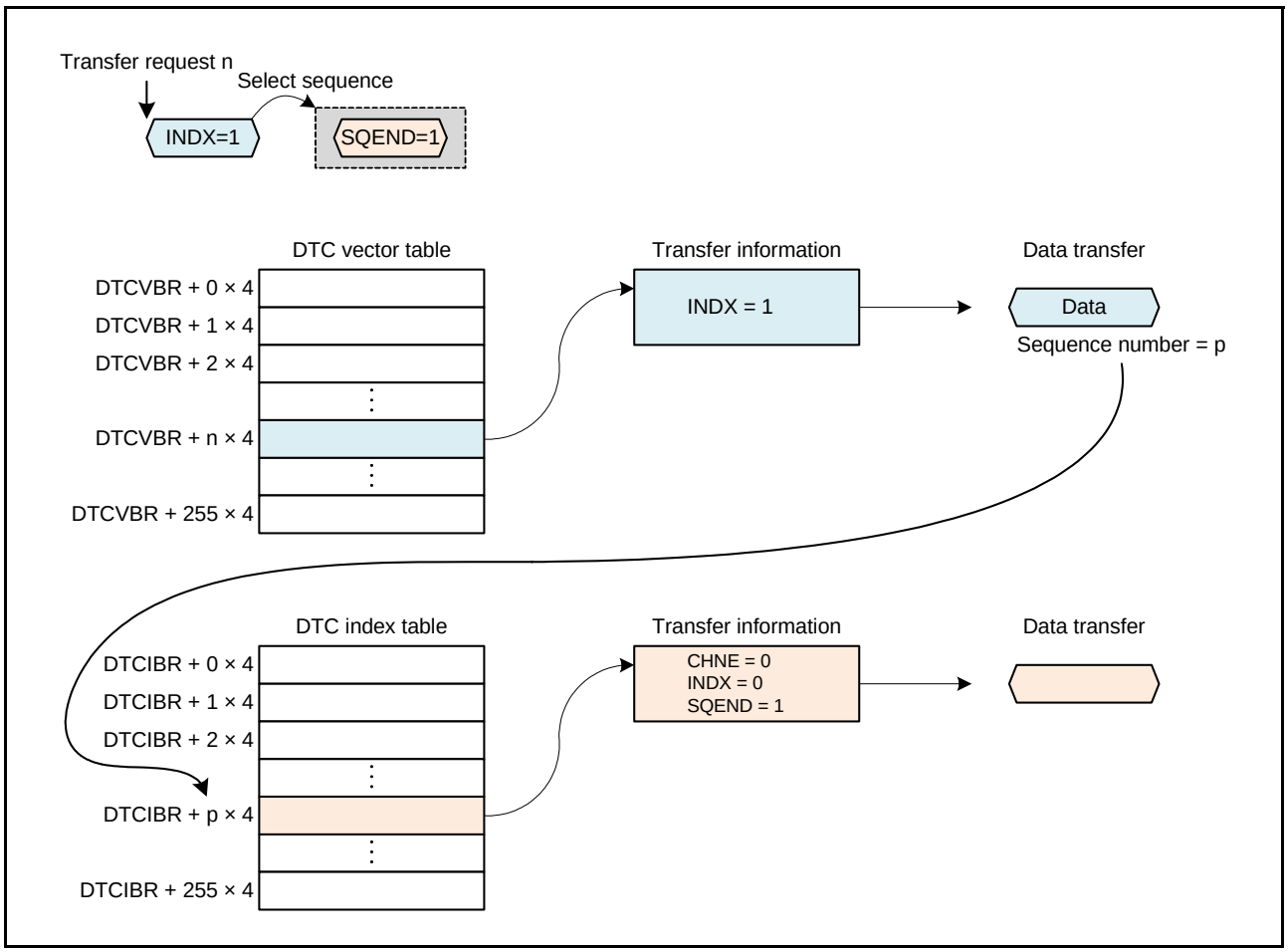


Figure 16.19 Example of a Sequence of Single Transfer

(2) When Executing a Single Chain Transfer

Figure 16.20 is an example of a sequence for a single chain transfer.

The DTC refers to the DTC index table, and reads the transfer information corresponding to the obtained sequence number q.

While the values of the CHNE, INDX, and SQEND bits are 1, 0, and 0 respectively, the specified chain transfer is executed. When the transfer information in which the CHNE, INDX, and SQEND bits are 0, 0, and 1 respectively is read, the sequence ends after the specified transfer is executed.

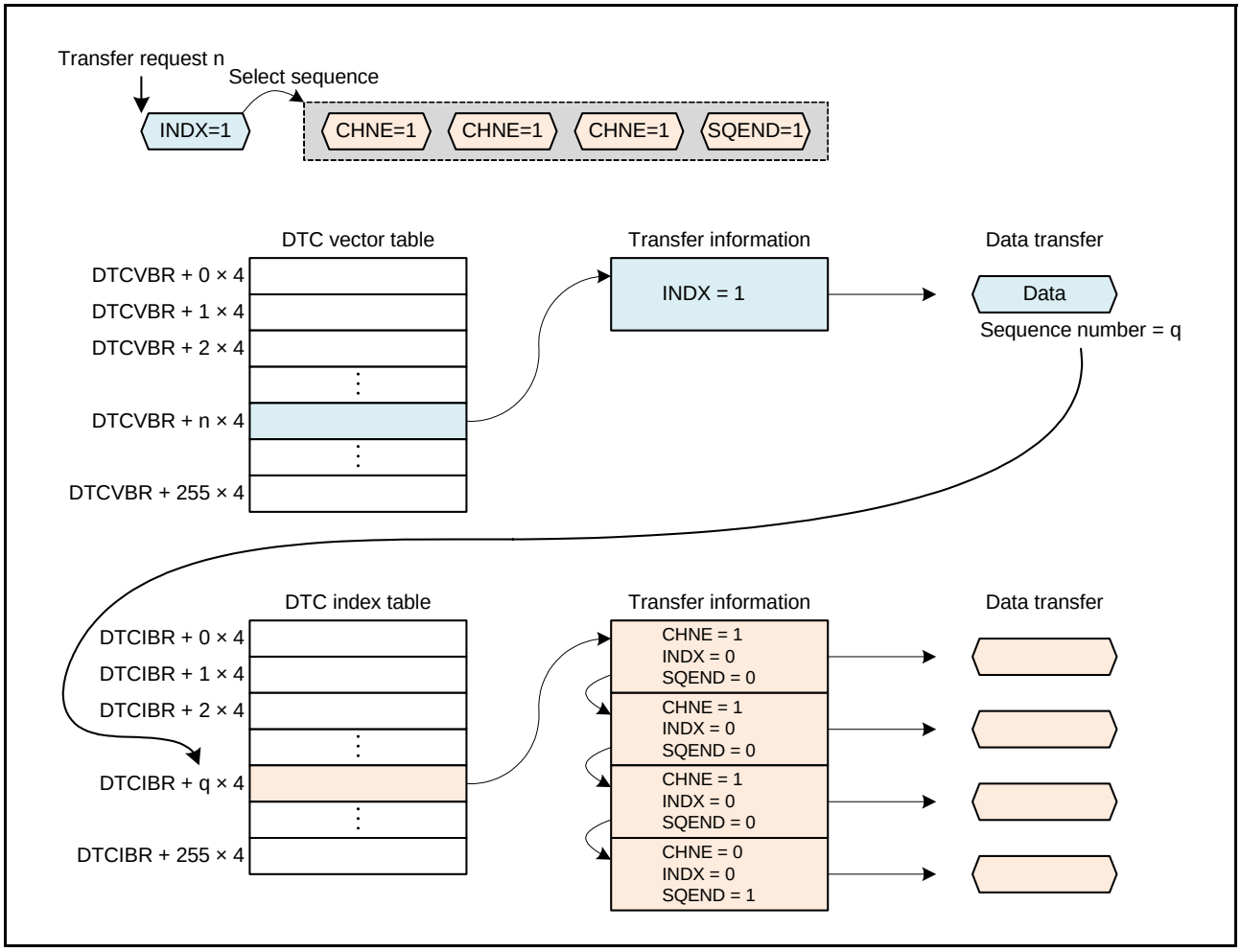


Figure 16.20 Example of Sequence of a Single Chain Transfer

(3) When Dividing a Sequence

Figure 16.21 is an example of the sequence that is divided into 3 parts.
The DTC refers to the DTC index table, and reads the transfer information corresponding to the obtained sequence number r.
Since the values of the CHNE, INDX, and SQEND bits in the transfer information are 0, 0, and 0 respectively, the sequence is suspended after the specified transfer is executed and the DTC waits for the next transfer request n.
When the transfer request n is input during a sequence transfer, the DTC vector table is not referred and the suspended sequence is resumed.
When the transfer information in which the CHNE, INDX, and SQEND bits are 0, 0, and 1 respectively is read, the sequence ends after the specified transfer is executed.

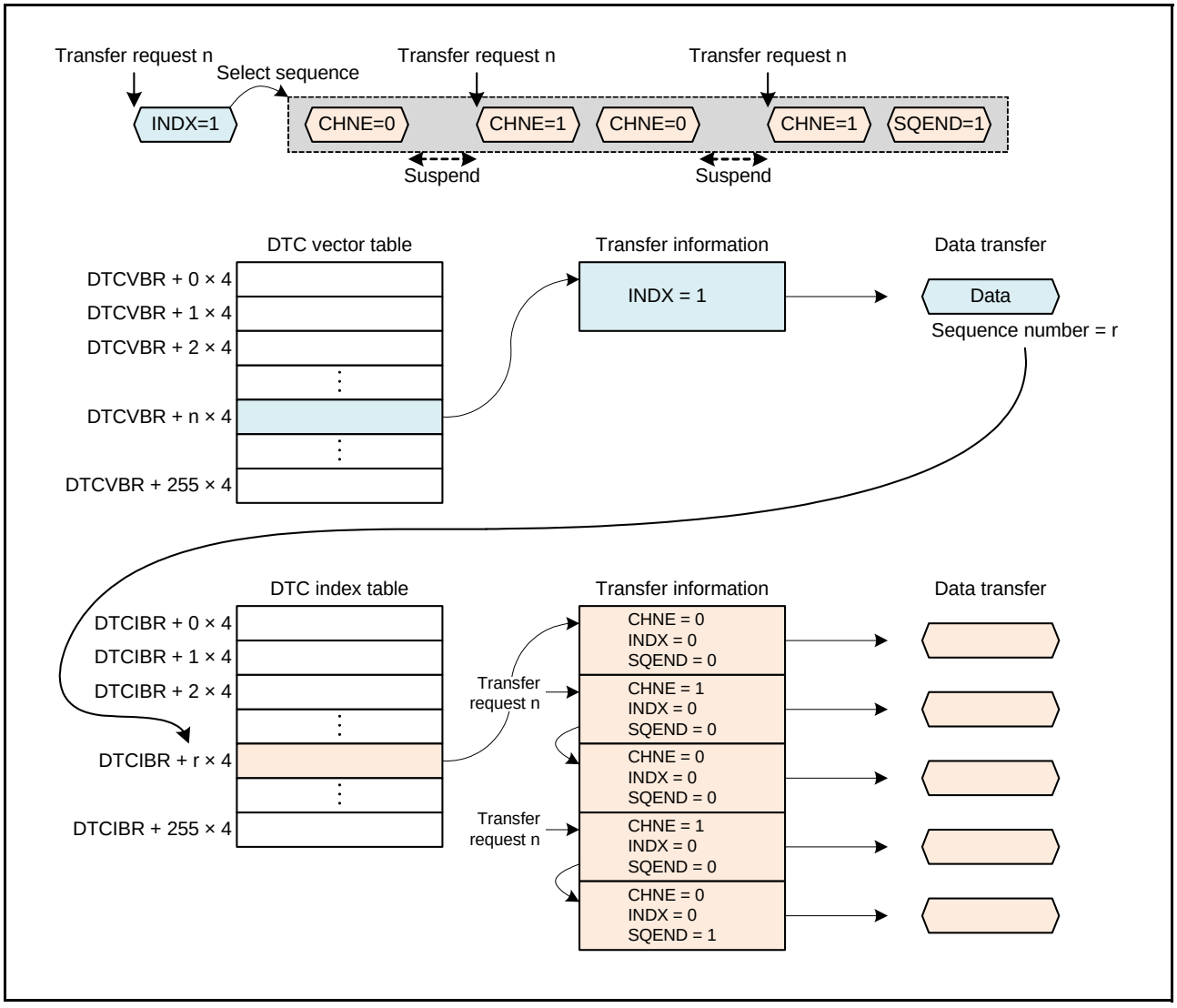


Figure 16.21 Example of Divided Sequence

(4) When Starting a New Sequence on completion of a Sequence

Figure 16.22 is an example for starting the next and new sequence on completion of the first sequence.
The DTC refers to the DTC index table, and reads the transfer information corresponding to the obtained sequence number s.
When the transfer information in which the CHNE, INDX, and SQEND bits are 0, 1, and 1 respectively is read, the specified transfer is executed, then a new sequence number is obtained from the lower 8 bits of the transferred data.
The DTC again refers to the DTC index table, and reads the transfer information corresponding to the obtained sequence number k and then starts a new sequence.
When the transfer information in which the CHNE, INDX, and SQEND bits are 0, 0, and 1 respectively is read, the sequence ends after the specified transfer is executed.

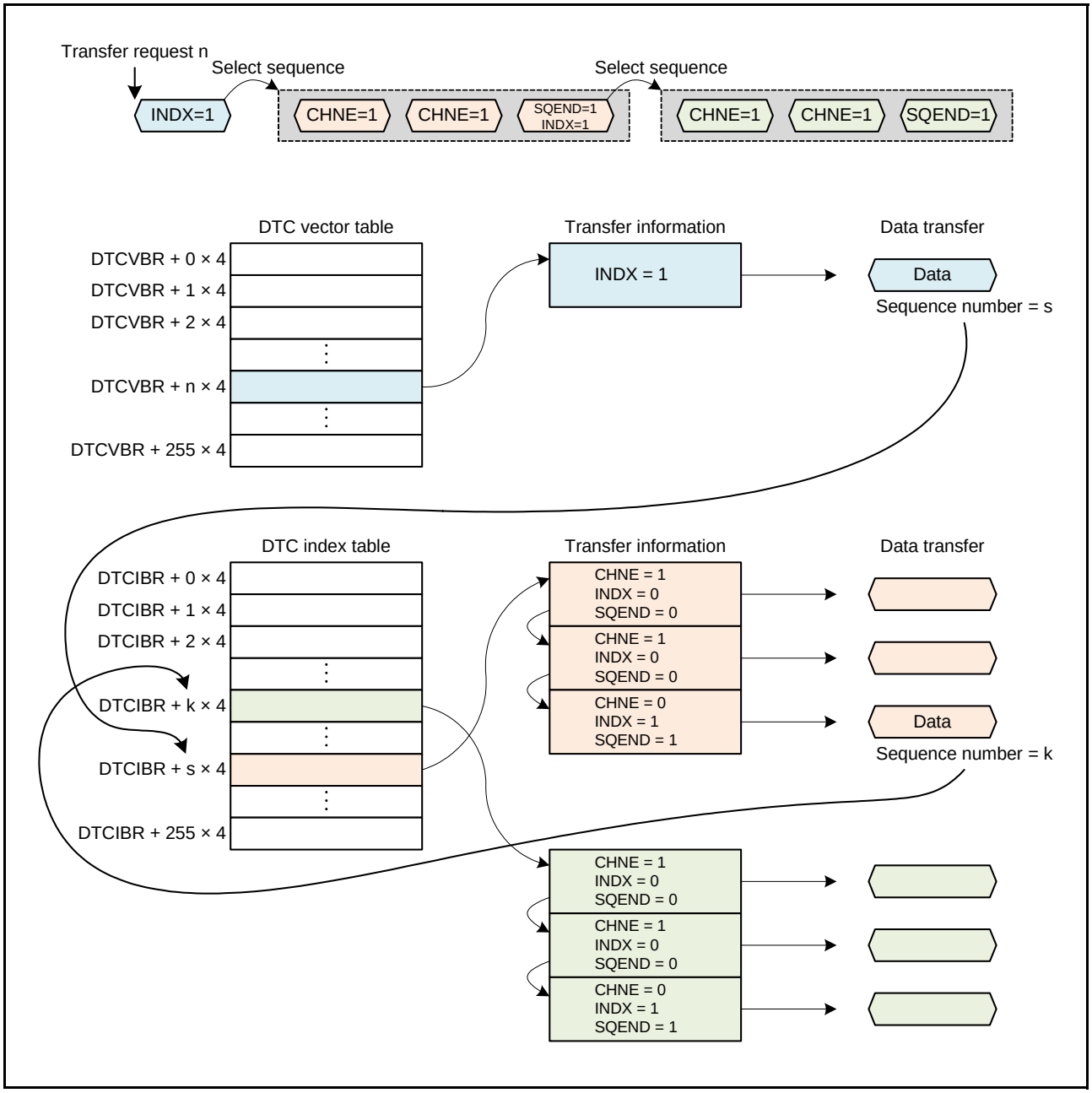


Figure 16.22 Example When Starting a New Sequence on Completion of a Sequence

(5) When Generating an Interrupt Request to the CPU

Figure 16.23 is an example of that an interrupt request is output to the CPU without starting of sequence.
The DTC obtains a DTC index that corresponds to the obtained sequence number t.
When the CPUSEL bit of the obtained DTC index is 1, the DTC ends the sequence transfer without starting the sequence, and then outputs an interrupt request to the CPU.

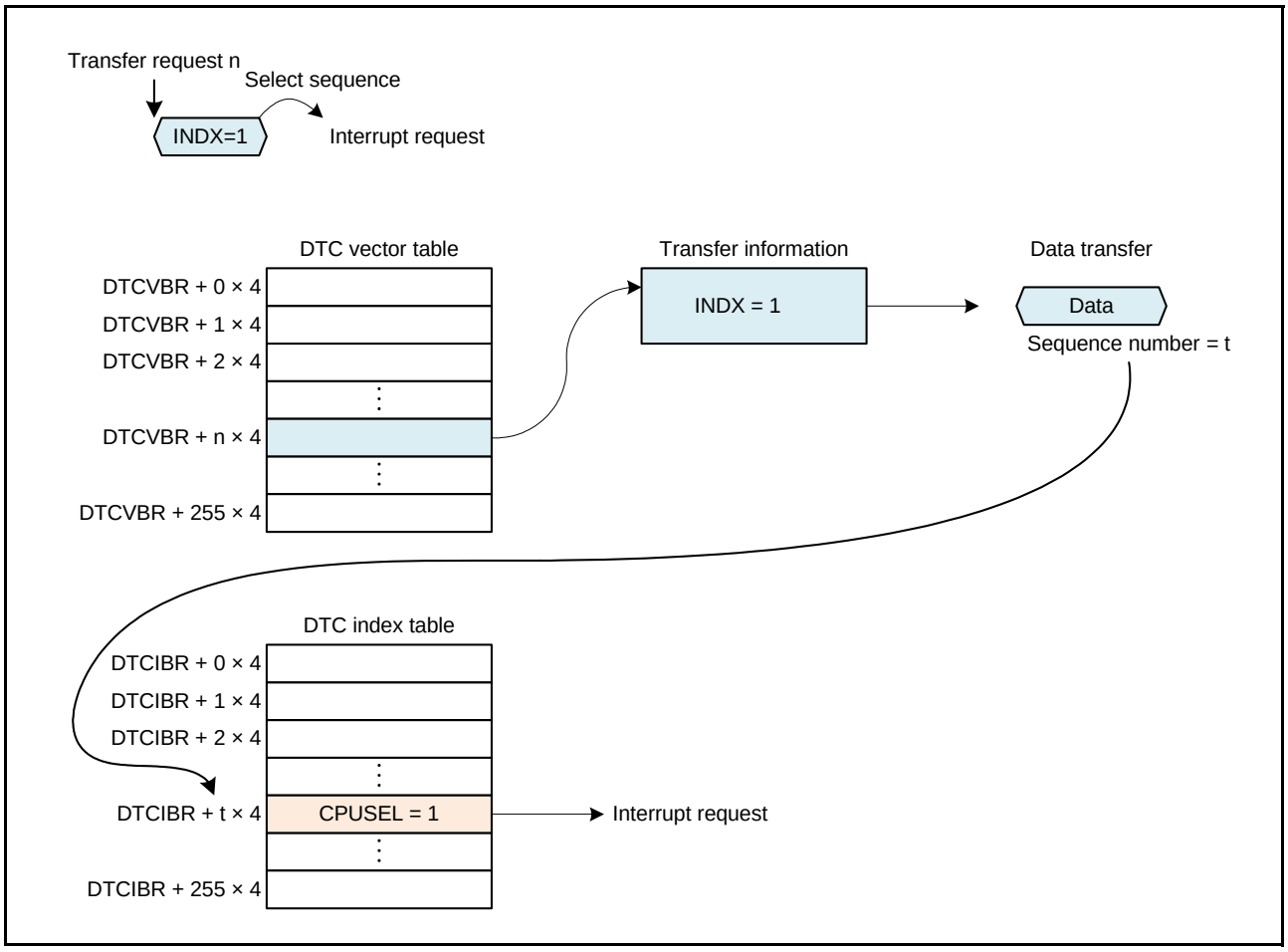
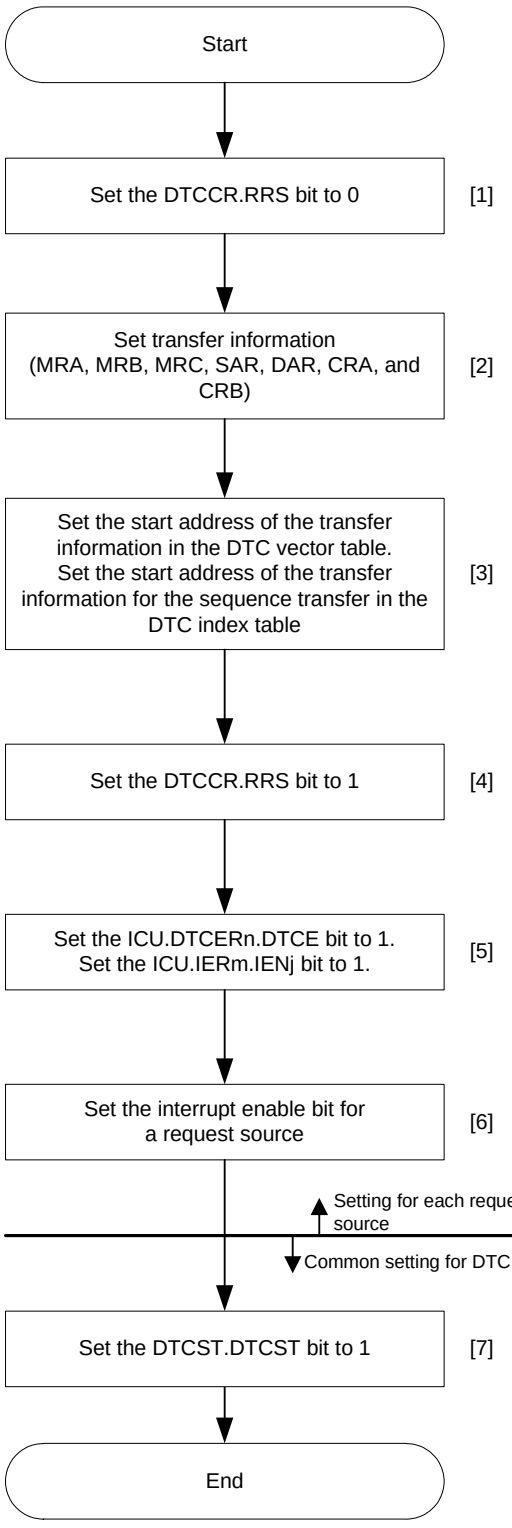


Figure 16.23 Example of Output of an Interrupt Request to the CPU

16.5 DTC Setting Procedure

Before using the DTC, set the DTC vector base register (DTCVBR). When using sequence transfer, also set the DTC index table base register (DTCIBR).

Figure 16.24 shows the procedure to set the DTC.



Set the ICU.IERm.IENj bit corresponding to the request source interrupt to 0 and provide the following settings.

[1] Setting the DTCCR.RRS bit to 0 resets the transfer information read skip flag. After that, transfer information read is not skipped during the data transfer. When transfer information is updated, be sure to make this setting.

[2] Allocate transfer information (MRA, MRB, MRC, SAR, DAR, CRA, and CRB) in the data area. For setting transfer information, refer to section 16.2, Register Descriptions. For how to allocate transfer information, refer to section 16.3.1, Allocating Transfer Information and DTC Vector Table.

[3] Set the start address of the transfer information in the DTC vector table. For how to set the DTC vector table, refer to section 16.3.1, Allocating Transfer Information and DTC Vector Table. When using sequence transfer, set the DTCADMOD.SHORT bit to 0, set registers DTCIBR, DTCSQE, and DTCDISP, and allocate the DTC index table.

[4] Setting the DTCCR.RRS bit to 1 enables skipping of the second and the subsequent transfer information read cycles for continuous data transfer due to the same interrupt source. The RRS bit can be set to 1 at any time, but this setting during data transfer becomes valid from the next transfer. Leave the value of the RRS bit as 0 if the value of the MRA.WBDIS bit or the MRC.DISPE bit in any transfer information is 1 or when sequence transfer is to be used.

[5] Set the ICU.DTCERn.DTCE bit (n = interrupt vector number) that corresponds to the DTC request source to 1. Set the ICU.IERm.IENj bit to 1. For the correspondence between interrupt sources and DTCERn register, refer to section 14.3.1, Interrupt Vector Table in section 14, Interrupt Controller (ICUb).

[6] Set the interrupt enable bit of the peripheral module, which is request source, to 1. When a source interrupt is generated, the DTC starts operation. For the setting of the interrupt enable bit, refer to the setting of the peripheral module that is to be an request source.

[7] Set the DTCST.DTCST bit to 1 (DTC module start)*1.

Note 1. The DTCST.DTCST bit can be set even if the setting for each request source is not completed.

Figure 16.24 Procedure to Set the DTC

16.6 Examples of DTC Usage

16.6.1 Normal Transfer

As an example of DTC usage, its employment in the reception of 128 bytes of data by an SCI is described below.

(1) Transfer Information Setting

Set the MRA.MD[1:0] bits to 00b (normal transfer mode), the MRA.SZ[1:0] bits to 00b (byte transfer), and the MRA.SM[1:0] bits to 00b (source address is fixed). Set the MRB.CHNE bit to 0 (chain transfer is disabled), the MRB.DISEL bit to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers), and the MRB.DM[1:0] bits to 10b (DAR is incremented after data transfer). The MRB.DTS bit can be set to any value. Set the RDR register address of the SCI in the SAR register, the start address of the RAM area for data storage in the DAR register, and 128 (0080h) in the CRA register. The CRB register can be set to any value.

(2) DTC Vector Table Setting

The start address of the transfer information for the RXI interrupt is set in the vector table for the DTC.

(3) ICU Setting and DTC Module Activation

Set the corresponding ICU.DTCERn.DTCE bit to 1 and the ICU.IERm.IENj bit to 1.
Set the DTCST.DTCST bit to 1.

(4) SCI Setting

Enable the RXI interrupt by setting the SCR.RIE bit in the SCI to 1. If a reception error occurs during the SCI receive operation, further reception is not performed. Accordingly, make settings so that the CPU can accept receive error interrupts.

(5) DTC Transfer

Every time the reception of 1 byte by the SCI is completed, an RXI interrupt is generated to start the data transfer. The DTC transfers the received byte from the RDR of the SCI to RAM, after which the DAR register is incremented and the CRA register is decremented.

(6) Interrupt Handling

After 128 times of data transfers have been completed and the value in the CRA register becomes 0, an RXI interrupt request is output to the CPU. Complete the process in the handling routine for this interrupt.

16.6.2 Chain Transfer When the Counter is 0

The second data transfer is performed only when the transfer counter is set to 0 in the first data transfer, and the first data transfer information is repeatedly changed in the second data transfer. Repeating this chain transfer enables transfers to be repeated more than 256 times.
The following shows an example of configuring a 128-Kbyte input buffer to addresses 20 0000h to 21 FFFFh (where the input buffer is set so that its lower address starts with 0000h). Figure 16.25 shows a chain transfer when the counter is 0.

(1) Set normal transfer mode for input data for the first data transfer. Set the following:
Transfer source address: Fixed, the CRA register is 0000h (65,536 times), the MRB.CHNE bit is 1 (chain transfer is enabled), the MRB.CHNS bit is 1 (chain transfer is performed only when the transfer counter becomes 0), and the MRB.DISEL bit is 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers).
(2) Prepare the upper 8 bits (in this case, 21h and 20h) of the start address at every 65,536 times of the transfer destination address for the first data transfer in another area (such as ROM).
(3) For the second data transfer, set repeat transfer mode (source is repeat area) for rewriting the transfer destination address of the first data transfer. The transfer destination is the address where the upper 8 bits of the DAR register in the first transfer information is allocated. In this case, set the MRB.CHNE bit to 0 (chain transfer is disabled) and the MRB.DISEL bit to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers). In this case, set the transfer counter to 2.
(4) When a transfer request is accepted, the first data transfer is executed. When transfer is executed 65,536 times and the transfer counter of the first data transfer becomes 0, the second data transfer is started and the upper 8 bits of the transfer destination address of the first data transfer is set to 21h. At this time, the lower 16 bits of the transfer destination address and the transfer counter of the first data transfer have become 0000h.
(5) In succession, when another transfer request is accepted, the first data transfer is executed. When transfer is executed 65,536 times and the transfer counter of the first data transfer becomes 0, the second data transfer is started and the upper 8 bits of the transfer destination address of the first data transfer is set to 20h. At this time, the lower 16 bits of the transfer destination address and the transfer counter of the first data transfer have become 0000h.
(6) Steps (4) and (5) above are repeated infinitely. Because the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated.

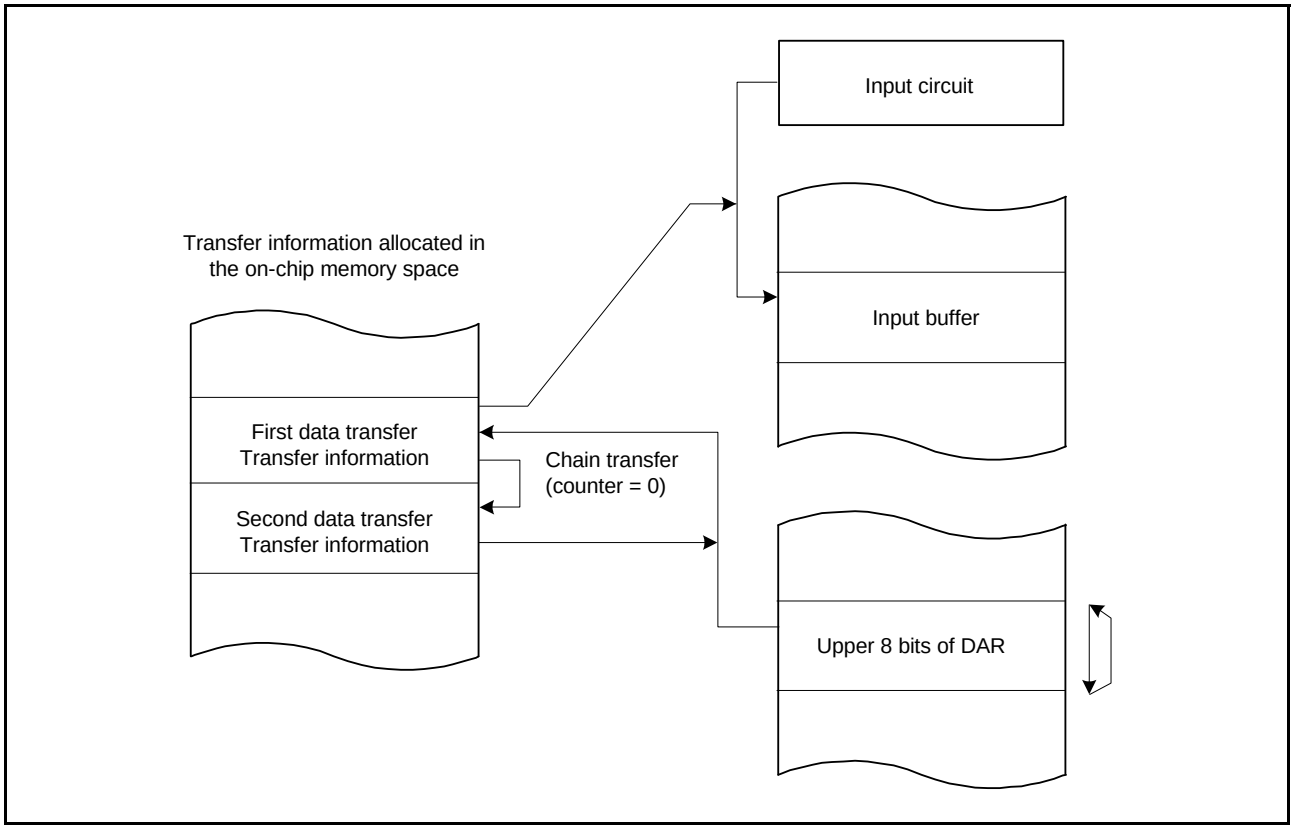


Figure 16.25 Chain Transfer When the Counter is 0

16.6.3 Sequence Transfer

The following is an example of using the SCI receive interrupt as a request source of sequence transfer.

(1) Transfer Information Settings

Set the MRA.MD[1:0] bits to 00b (normal transfer mode), the MRA.SZ[1:0] bits to 00b (byte transfer), and the MRA.SM[1:0] bits to 00b (source address is fixed). Set the MRB.CHNE bit to 0 (chain transfer is disabled), the MRB.DISEL bit to 0 (an interrupt request to the CPU is generated on completion of the specified number of data transfers), the MRB.DM[1:0] bits to 10b (DAR is incremented after data transfer), the MRB.INDX bit to 1 (start sequence transfer), and the MRB.SQEND bit to 0 (continue the sequence transfer). The MRB.DTS bit can be set to any value. Set the address of the SCIk.RDR register in the SAR register and set the start address of the RAM area which stores the data in the DAR register.
When the MRA.WBDIS bit is set to 1 (Does not write back the transfer information), the values of registers CRA and CRB are ignored.

(2) DTC Vector Table Setting

Set the start address of the transfer information for the corresponding receive data full interrupt (RXI) in the DTC vector table.

(3) DTC Index Table Setting

Set the start address of the transfer information for each sequence in the DTC index table.

(4) ICU Setting and DTC Module Activation

Set the corresponding ICU.DTCERn.DTCE bit to 1 and the ICU.IERm.IENj bit to 1. Set the DTCST.DTCST bit to 1.

(5) SCI Setting

Set the SCIk.SCR.RIE bit to 1 to enable the RXI interrupt. If a reception error occurs during the SCI receive operation, subsequent receptions are not performed. Accordingly, make settings so that the CPU can accept receive error interrupts.

(6) Start of the Sequence Transfer

On completion of reception of 1-byte data by the SCI, an RXI interrupt is generated to start the DTC. The DTC transfers the received data from the SCIk.RDR register to the RAM. The DTC looks up the DTC index table by using the value from the received data (sequence number) and continues to transfer data corresponding to the that number.
When the CPUSEL bit in the DTC index is 1, the DTC does not read the transfer information and sets the ICU.DTCERn.DTCE bit to 0. Then the DTC outputs an interrupt request to the CPU and ends the sequence transfer.

(7) During Suspension of the Sequence Transfer

Set the ICU.DTCERn.DTCE bit to 1 if the bit is 0. The DTC continues to transfer the data for every generation of the DTC transfer request in response to the corresponding RXI interrupt.

(8) End of the Sequence Transfer

Set the MRB.SQEND bit in the last transfer information of the sequence transfer to 1. After execution of this data transfer, the DTC ends the sequence transfer. The DTC starts to refer to the DTC vector table when a DTC transfer request is generated due to the next corresponding RXI interrupt.

16.7 Interrupt Source

When the DTC has finished data transfer of specified count or when data transfer with the MRB.DISEL bit set to 1 (an interrupt request to the CPU is generated each time the data transfer is performed) has been completed, an interrupt to the CPU is generated by the DTC trigger source. Such interrupts to the CPU are controlled according to the PSW.I bit (interrupt enable) of the CPU, the PSW.IPL[3:0] bits (processor interrupt priority level), and the priority level of the interrupt controller.

16.8 Low Power Consumption Function

Before making a transition to the module stop state, deep sleep mode, or software standby mode, set the DTCST.DTCST bit to 0 (DTC module stop), and then perform the following.

(1) Module Stop Function

Writing 1 (transition to the module-stop state is made) to the MSTPCRA.MSTPA28 bit enables the module stop function of the DTC. If data transfer is in progress at the time 1 is written to the MSTPCRA.MSTPA28 bit, the transition to the module stop state proceeds after data transfer has ended. While the MSTPCRA.MSTPA28 bit is 1, accessing the DTC registers is prohibited.
Writing 0 (release from the module-stop state) to the MSTPCRA.MSTPA28 bit releases the DTC from the module stop state.

(2) Deep Sleep Mode

Make settings according to the procedure under section 11.6.2.1, Entry to Deep Sleep Mode, in section 11, Low Power Consumption.
If any data transfer is in progress at the time the WAIT instruction is executed, the transition to deep sleep mode follows the completion of the data transfer.
The DTC is released from the module stop state by writing 0 to the MSTPCRA.MSTPA28 bit following recovery from deep sleep mode.

(3) Software Standby Mode

Make settings according to the procedure under section 11.6.3.1, Entry to Software Standby Mode, in section 11, Low Power Consumption.
If any data transfer is in progress at the time the WAIT instruction is executed, the transition to software standby mode follows the completion of the data transfer.

(4) Notes on Low Power Consumption Function

For the WAIT instruction and the register setting procedure, refer to section 11.7.5, Timing of WAIT Instructions in section 11, Low Power Consumption.
To perform data transfer after returning from a low power consumption mode, set the DTCST.DTCST bit to 1 again.
To use a request that is generated in deep sleep mode or software standby mode as an interrupt request to the CPU but not as a DTC transfer request, specify the CPU as the interrupt request destination according to the description in section 14.4.3, Selecting Interrupt Request Destinations in section 14, Interrupt Controller (ICUb), and then execute the WAIT instruction.

16.9 Usage Notes

16.9.1 Start Address of Transfer Information

Set multiples of 4 for the start addresses of the transfer information to be specified in the DTC vector table. If any value other than a multiple of 4 is specified, access still proceeds with the lower 2 bits of the address regarded as 00b.

16.9.2 Allocating Transfer Information

Allocate transfer information in the memory area according to the endian of the area as shown in Figure 16.26. For example, when writing CRA and CRB settings in 16-bit units in big endian, write the CRA setting to the address plus 8h (Ch) and the CRB setting to the address plus Ah (Eh). In little endian, write the CRB setting to the address plus 8h (Ch) and the CRA setting to the address plus Ah (Eh). When writing CRA and CRB settings in 32-bit units, allocate the CRA setting at the MSB side of the 32 bits and the CRB setting at the LSB side, and write the settings to the address plus 8h (Ch), regardless of endian.

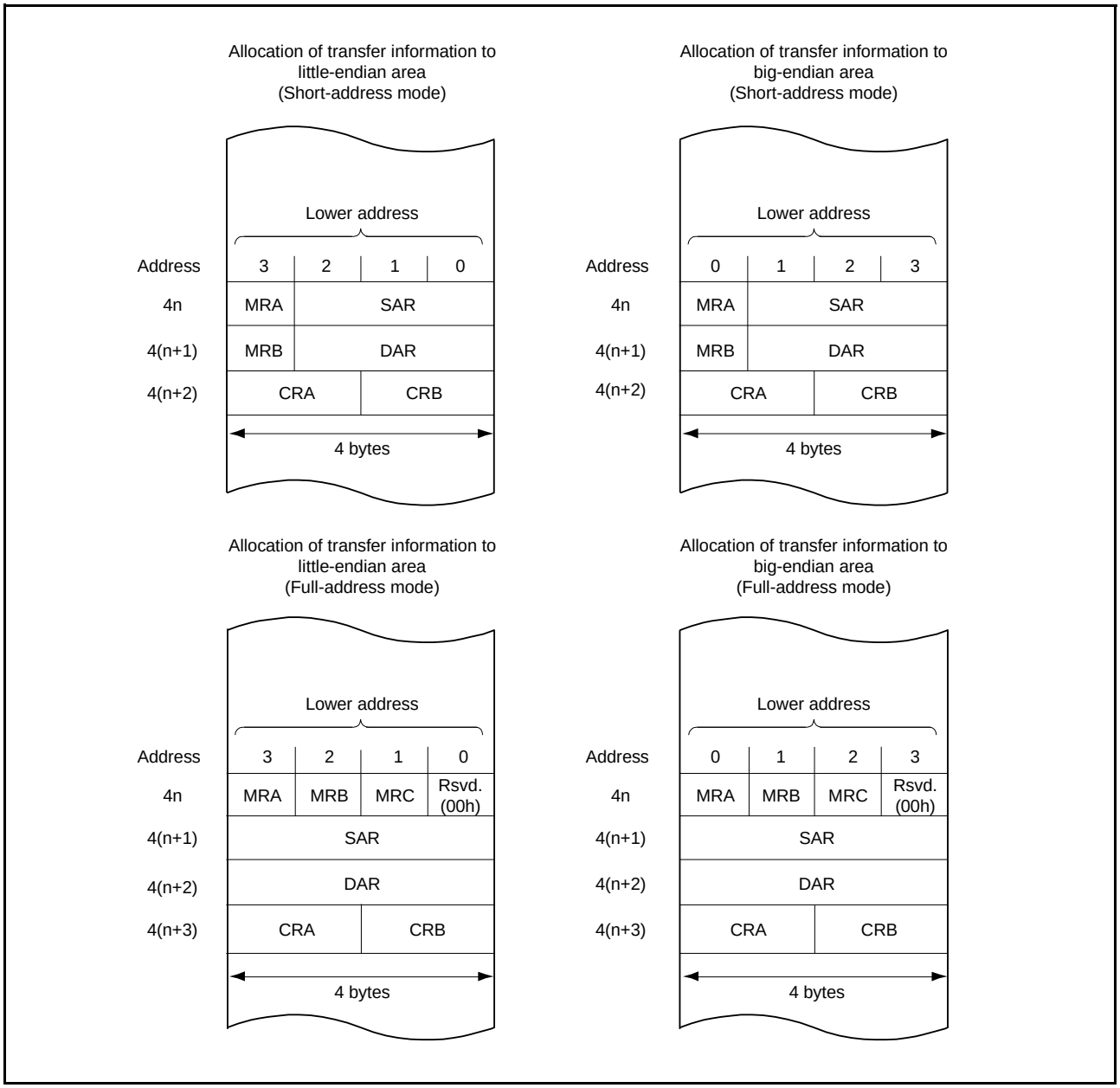


Figure 16.26 Allocation of Transfer Information

16.9.3 Notes on Using the Sequence Transfer

When sequence transfer is to be used, make sure that the DTCADMOD.SHORT bit is 0 (full-address mode) and the DTCCR.RRS bit is also 0 (transfer information read is not skipped).
In addition, set the MRB.CHNE bit to 0 (chain transfer is disabled) when setting the MRB.INDX bit to 1 (start sequence transfer and refer the index table) or the MRB.SQEND bit to 1 (end the sequence transfer).

17. I/O Ports

17.1 Overview

The I/O ports function as a general I/O port, an I/O pin of a peripheral module, or an input pin for an interrupt.
Some of the pins are also configurable as an I/O pin of a peripheral module or an input pin for an interrupt. All pins function as input pins immediately after a reset, and pin functions are switched by register settings. The setting of each pin is specified by the registers for the corresponding I/O port and on-chip peripheral modules.
Each port has the port direction register (PDR) that selects input or output direction, the port output data register (PODR) that holds data for output, the port input data register (PIDR) that indicates the pin states, the open drain control register y (ODRy, y = 0, 1) that selects the output type of each pin, the pull-up control register (PCR) that controls on/off of the input pull-up MOS, the drive capacity control register (DSCR) that selects the drive capacity, and the port mode register (PMR) that specifies the pin function of each port.
For details on the PMR register, see section 18, Multi-Function Pin Controller (MPC).
The configuration of the I/O ports differs depending on the package. Table 17.1 lists the specifications of I/O ports, and Table 17.2 list the port functions.

Table 17.1 Specifications of I/O Ports

Port	Package		Package	
	48 Pins	Number of Pin	32 Pins	Number of Pin
PORT1	P10, P11	2	P11	1
PORT2	P22 to P24	3	—	—
PORT3	P36, P37	2	P36, P37	2
PORT4	P40 to P47	8	P40 to P44	5
PORT7	P70 to P76	7	P71 to P76	6
PORT9	P93, P94	2	P93, P94	2
PORTA	PA2, PA3	2	—	—
PORTB	PB0 to PB7	8	PB0 to PB3, PB6, PB7	6
PORTD	PD3 to PD6	4	—	—
PORTE	PE2	1	PE2	1
	Total of Pins	39	Total of Pins	23

Table 17.2 Port Functions

Port	Pin	Input Pull-up	Open Drain Output	Drive Capacity Switching	High Current Pin	5-V Tolerant
PORT1	P10, P11	○	○	○	—	—
PORT2	P22	○	○	○	—	—
	P23, P24	○	○	○	—	—
PORT3	P36, P37	○	○	—	—	—
PORT4	P40 to P47	○	—	Fixed to normal output	—	—
PORT7	P70	○	○	○	—	—
	P71 to P76	○	○	Fixed to high drive output	○	—
PORT9	P93, P94	○	○	○	—	—
PORTA	PA2, PA3	○	○	○	—	—
PORTB	PB0, PB3	○	○	○	—	—
	PB1, PB2	○	○	Fixed to high drive output	—	○
	PB6	○	○	Fixed to high drive output	○	—
	PB4, PB5, PB7	○	○	○	—	—
PORTD	PD3 to PD6	○	○	○	—	—
PORTE	PE2	—	—	—	—	—

Specifying input pull-up, open-drain output, switching of drive capacity, or 5-V tolerance is available for other signals on pins that also function as general I/O pins.

17.2 I/O Port Configuration

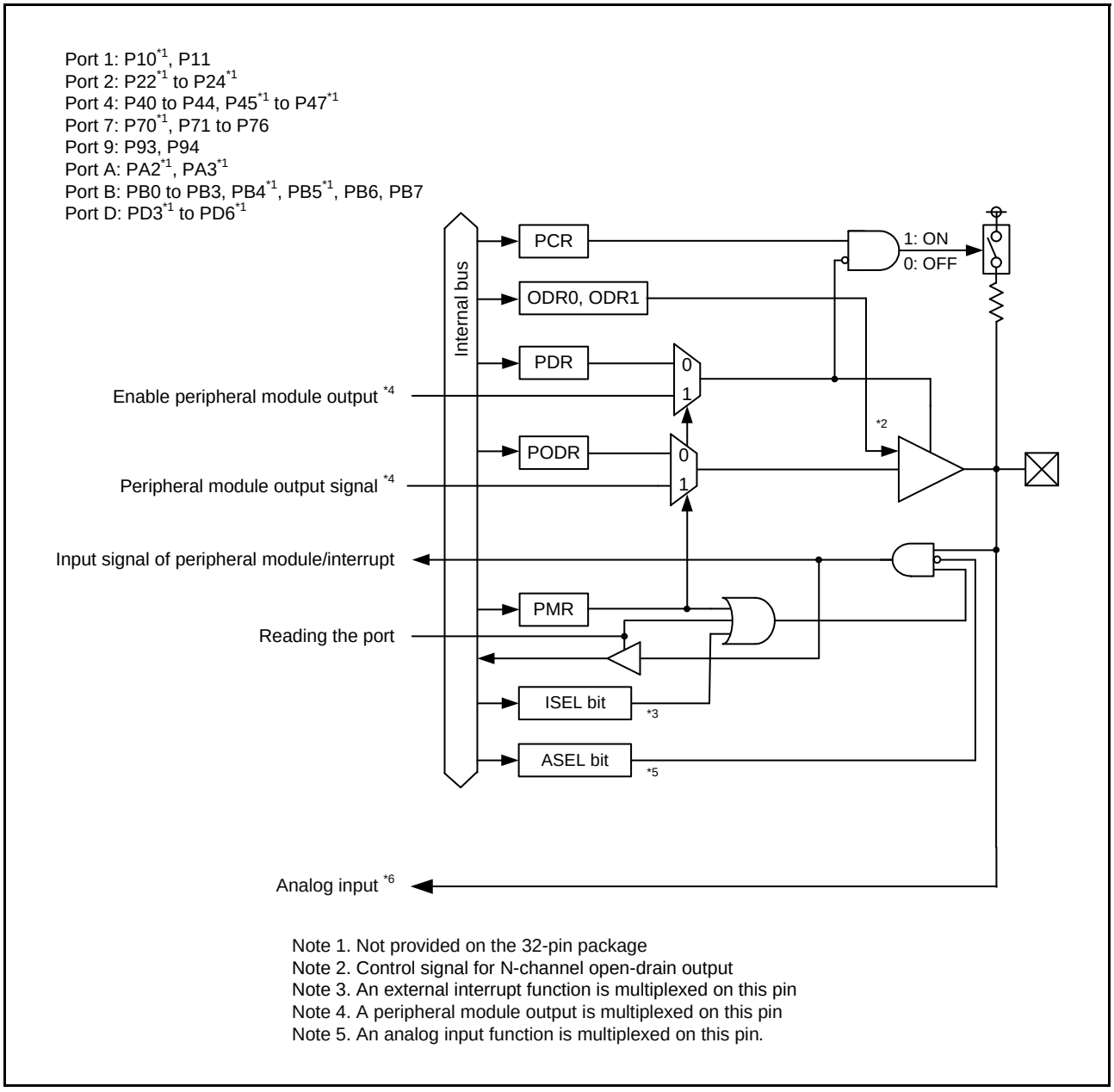


Figure 17.1 I/O Port Configuration (1)

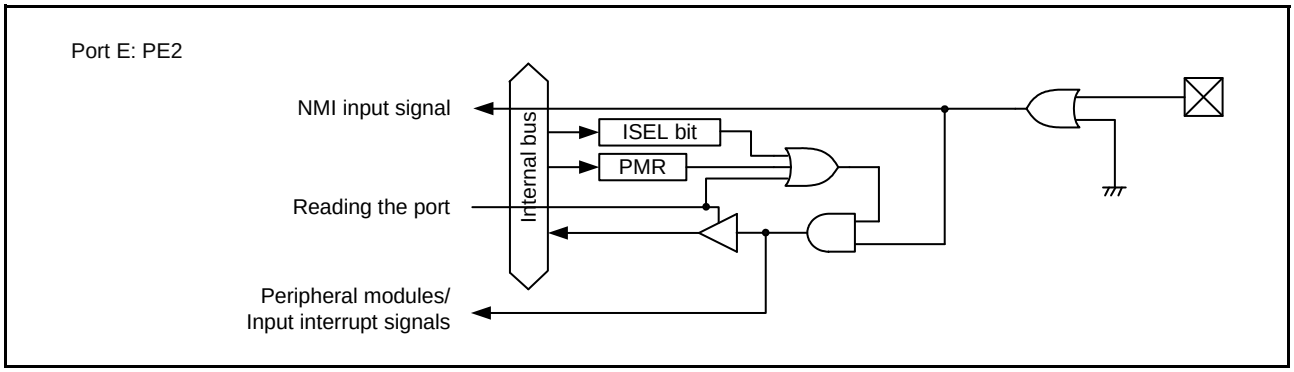
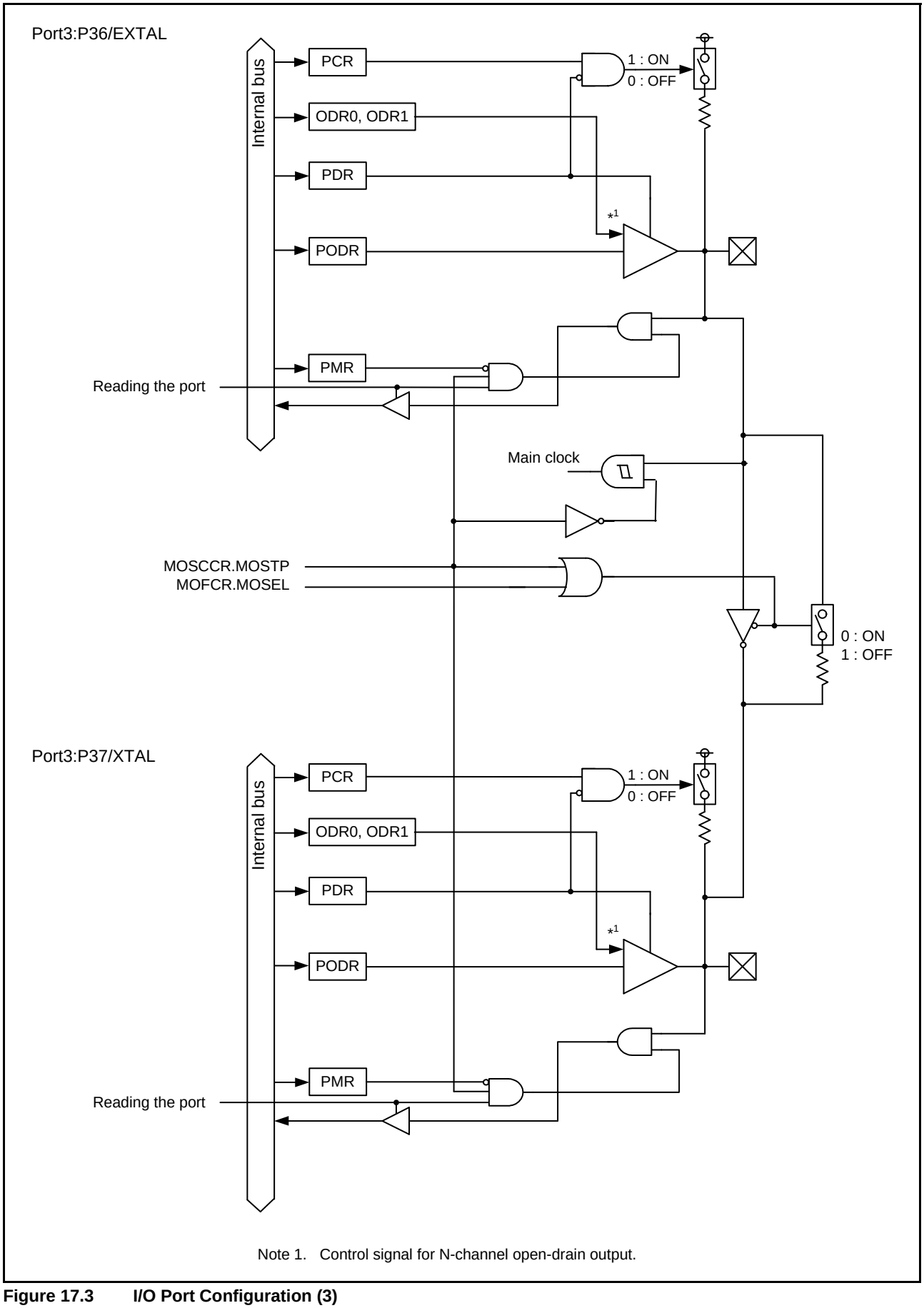


Figure 17.2 I/O Port Configuration (2)



Note 1. Control signal for N-channel open-drain output.

Figure 17.3 I/O Port Configuration (3)

17.3 Register Descriptions

17.3.1 Port Direction Register (PDR)

Address(es): PORT1.PDR 0008 C001h, PORT2.PDR 0008 C002h, PORT3.PDR 0008 C003h, PORT4.PDR 0008 C004h, PORT7.PDR 0008 C007h, PORT9.PDR 0008 C009h, PORTA.PDR 0008 C00Ah, PORTB.PDR 0008 C00Bh, PORTD.PDR 0008 C00Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 I/O Select	0: Input (Functions as an input pin.) 1: Output (Functions as an output pin.)	R/W
b1	B1	Pm1 I/O Select		R/W
b2	B2	Pm2 I/O Select		R/W
b3	B3	Pm3 I/O Select		R/W
b4	B4	Pm4 I/O Select		R/W
b5	B5	Pm5 I/O Select		R/W
b6	B6	Pm6 I/O Select		R/W
b7	B7	Pm7 I/O Select		R/W

m = 1 to 4, 7, 9, A, B, D

PDR is used to select the input or output direction for individual pins of the corresponding port m when the pins are configured as the general I/O pins.

Each bit of PORTm.PDR corresponds to each pin of port m; I/O direction can be specified in 1-bit units.

Write 1 (output) to each bit of PDR corresponding to port m that does not exist.

Each bit of PDR corresponding to port m that does not exist is reserved. Make settings according to the description in section 17.4, Initialization of the Port Direction Register (PDR).

17.3.2 Port Output Data Register (PODR)

Address(es): PORT1.PODR 0008 C021h, PORT2.PODR 0008 C022h, PORT3.PODR 0008 C023h, PORT4.PODR 0008 C024h, PORT7.PODR 0008 C027h, PORT9.PODR 0008 C029h, PORTA.PODR 0008 C02Ah, PORTB.PODR 0008 C02Bh, PORTD.PODR 0008 C02Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Output Data Store	Holds output data.	R/W
b1	B1	Pm1 Output Data Store		R/W
b2	B2	Pm2 Output Data Store		R/W
b3	B3	Pm3 Output Data Store		R/W
b4	B4	Pm4 Output Data Store		R/W
b5	B5	Pm5 Output Data Store		R/W
b6	B6	Pm6 Output Data Store		R/W
b7	B7	Pm7 Output Data Store		R/W

m = 1 to 4, 7, 9, A, B, D

PODR holds the data to be output from the pins used for general output ports.

17.3.3 Port Input Data Register (PIDR)

Address(es): PORT1.PIDR 0008 C041h, PORT2.PIDR 0008 C042h, PORT3.PIDR 0008 C043h, PORT4.PIDR 0008 C044h, PORT7.PIDR 0008 C047h, PORT9.PIDR 0008 C049h, PORTA.PIDR 0008 C04Ah, PORTB.PIDR 0008 C04Bh, PORTD.PIDR 0008 C04Dh, PORTE.PIDR 0008 C04Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0	Indicates individual pin states of the corresponding port.	R
b1	B1	Pm1		R
b2	B2	Pm2		R
b3	B3	Pm3		R
b4	B4	Pm4		R
b5	B5	Pm5		R
b6	B6	Pm6		R
b7	B7	Pm7		R

m = 1 to 4, 7, 9, A, B, D, E

PIDR indicates individual pin states of port m.

The pin states of port m can be read with the PORTm.PIDR, regardless of the values of PORTm.PDR and PORTm.PMR.

The NMI pin state is reflected in the PE2 bit.

The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as undefined, and cannot be modified.

Note: When using P36 and P37 as general I/O ports, set the MOSCCR.MOSTP bit to 1 (main clock oscillator is stopped) and the P36 and P37 control bits in the PORT3.PMR register to 0 (use pin as general I/O port).

17.3.4 Port Mode Register (PMR)

Address(es): PORT1.PMR 0008 C061h, PORT2.PMR 0008 C062h, PORT3.PMR 0008 C063h, PORT7.PMR 0008 C067h, PORT9.PMR 0008 C069h, PORTA.PMR 0008 C06Ah, PORTB.PMR 0008 C06Bh, PORTD.PMR 0008 C06Dh, PORTE.PMR 0008 C06Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Pin Mode Control	0: Use pin as general I/O port. 1: Use pin as I/O port for peripheral functions.	R/W
b1	B1	Pm1 Pin Mode Control		R/W
b2	B2	Pm2 Pin Mode Control		R/W
b3	B3	Pm3 Pin Mode Control		R/W
b4	B4	Pm4 Pin Mode Control		R/W
b5	B5	Pm5 Pin Mode Control		R/W
b6	B6	Pm6 Pin Mode Control		R/W
b7	B7	Pm7 Pin Mode Control		R/W

m = 1 to 3, 7, 9, A, B, D, E

Each bit of PORTm.PMR corresponds to each pin of port m; pin function can be specified in 1-bit units.
Bits corresponding to port m on the 48 pin-product but which do not exist on a product with fewer than 48 pins are reserved. Write 0 to these bits.
The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as 0. The write value should be 0.

17.3.5 Open Drain Control Register 0 (ODR0)

Address(es): PORT1.ODR0 0008 C082h, PORT2.ODR0 0008 C084h, PORT7.ODR0 0008 C08Eh, PORT9.ODR0 0008 C092h, PORTA.ODR0 0008 C094h, PORTB.ODR0 0008 C096h, PORTD.ODR0 0008 C09Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	B6	—	B4	—	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Output Type Select	• P10, P70 b0 0: CMOS output 1: N-channel open-drain b1 This bit is read as 0. The write value should be 0. • PB0 b1 b0 0 0: CMOS output 0 1: N-channel open-drain 1 0: P-channel open-drain 1 0: Hi-Z	R/W
b1	B1			R/W
b2	B2	Pm1 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	B4	Pm2 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	B6	Pm3 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

m = 1, 2, 7, 9, A, B, D

Bits corresponding to port m on the 48 pin-product but which do not exist on a product with fewer than 48 pins are reserved. Write 0 to these bits.

The bits corresponding to a pin that does not exist or pins with no open-drain output allocation are reserved. A reserved bit is read as 0. The write value should be 0.

17.3.6 Open Drain Control Register 1 (ODR1)

Address(es): PORT2.ODR1 0008 C085h, PORT3.ODR1 0008 C087h, PORT7.ODR1 0008 C08Fh, PORT9.ODR1 0008 C093h, PORTB.ODR1 0008 C097h, PORTD.ODR1 0008 C09Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	B6	—	B4	—	B2	—	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm4 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	B2	Pm5 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	B4	Pm6 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	B6	Pm7 Output Type Select	0: CMOS output 1: N-channel open-drain	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

m = 2, 3, 7, 9, B, D

Bits corresponding to port m on the 48 pin-product but which do not exist on a product with fewer than 48 pins are reserved. Write 0 to these bits.

The bits corresponding to a pin that does not exist or pins with no open-drain output allocation are reserved. A reserved bit is read as 0. The write value should be 0.

17.3.7 Pull-Up Control Register (PCR)

Address(es): PORT1.PCR 0008 C0C1h, PORT2.PCR 0008 C0C2h, PORT3.PCR 0008 C0C3h, PORT4.PCR 0008 C0C4h, PORT7.PCR 0008 C0C7h, PORT9.PCR 0008 C0C9h, PORTA.PCR 0008 C0CAh, PORTB.PCR 0008 C0CBh, PORTD.PCR 0008 C0CDh

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Input Pull-Up Resistor Control	0: Disables an input pull-up resistor. 1: Enables an input pull-up resistor.	R/W
b1	B1	Pm1 Input Pull-Up Resistor Control		R/W
b2	B2	Pm2 Input Pull-Up Resistor Control		R/W
b3	B3	Pm3 Input Pull-Up Resistor Control		R/W
b4	B4	Pm4 Input Pull-Up Resistor Control		R/W
b5	B5	Pm5 Input Pull-Up Resistor Control		R/W
b6	B6	Pm6 Input Pull-Up Resistor Control		R/W
b7	B7	Pm7 Input Pull-Up Resistor Control		R/W

m = 1 to 4, 7, 9, A, B, D

While a pin is in the input state with the corresponding bit in PORTm.PCR set to 1, the pull-up resistor connected to the pin is enabled.

When a pin is used as a general port output pin, or a peripheral function output pin, the pull-up resistor for the pin is disabled regardless of the settings of the PCR register.

The pull-up resistor is also disabled in the reset state.

The bit corresponding to a pin that does not exist is also reserved. A reserved bit is read as 0. The write value should be 0.

17.3.8 Drive Capacity Control Register (DSCR)

Address(es): PORT1.DSCR 0008 C0E1h, PORT2.DSCR 0008 C0E2h, PORT7.DSCR 0008 C0E7h, PORT9.DSCR 0008 C0E9h, PORTA.DSCR 0008 C0EAh, PORTB.DSCR 0008 C0EBh, PORTD.DSCR 0008 C0EDh

	b7	b6	b5	b4	b3	b2	b1	b0
	B7	B6	B5	B4	B3	B2	B1	B0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0 Drive Capacity Control	0: Normal drive output 1: High-drive output	R/W
b1	B1	Pm1 Drive Capacity Control		R/W
b2	B2	Pm2 Drive Capacity Control		R/W
b3	B3	Pm3 Drive Capacity Control		R/W
b4	B4	Pm4 Drive Capacity Control		R/W
b5	B5	Pm5 Drive Capacity Control		R/W
b6	B6	Pm6 Drive Capacity Control		R/W
b7	B7	Pm7 Drive Capacity Control		R/W

m = 1, 2, 7, 9, A, B, D

The bit corresponding to a pin with the fixed drive capacity can be read from or written to. However, the drive capacity cannot be changed.
When high-drive output is selected, switching noise increases compared to when normal output is selected. Carefully evaluate the effect of noise on the MCU caused by adjacent pins before selecting high-drive output.
The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as 0. The write value should be 0.

17.4 Initialization of the Port Direction Register (PDR)

Initialize reserved bits in the PDR register according to Table 17.3 and Table 17.4.

- The blank columns in Table 17.3 and Table 17.4 indicate the bits corresponding to the pins listed in Table 17.1, Specifications of I/O Ports.
  The corresponding bits should be set to 1 (output) or 0 (input) depending on the user system.
- The columns other than the blank columns in Table 17.3 and Table 17.4 indicate reserved bits.
  A reserved bit should be set to 0 (input) or 1 (output) according to Table 17.3 and Table 17.4.
  When setting a value to a reserved bit, access in byte units.

Table 17.3 PDR Register Settings in 48-Pin Packages

	PDR Register							
Port Symbol	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
PORT1	0	0	0	0	0	0		
PORT2	0	0	0				0	0
PORT3			0	0	0	0	0	0
PORT4								
PORT7	0							
PORT9	0	0	0			0	0	0
PORTA	0	0	0	0			0	0
PORTB								
PORTD	0					0	0	0

Table 17.4 PDR Register Settings in 32-Pin Packages

	PDR Register							
Port Symbol	**b7**	**b6**	**b5**	**b4**	**b3**	**b2**	**b1**	**b0**
PORT1	0	0	0	0	0	0		1
PORT2	0	0	0	1	1	1	0	0
PORT3			0	0	0	0	0	0
PORT4	1	1	1					
PORT7	0							1
PORT9	0	0	0			0	0	0
PORTA	0	0	0	0	0	0	0	0
PORTB			1	1				
PORTD	0	1	1	1	1	0	0	0

17.5 Handling of Unused Pins

The configuration of unused pins is listed in Table 17.5.

Table 17.5 Unused Pin Configuration

Pin Name	Description
MD	(Always used as mode pins)
RES#	Connect this pin to VCC via a pull-up resistor.
PE2/NMI	Connect this pin to VCC via a pull-up resistor.
P36/EXTAL	When the main clock is not used, set the MOSCCR.MOSTP bit to 1 (general port P36). When this pin is not used as port P36 either, it is configured in the same way as port 1, 2, 7, 9, A, B, D.
P37/XTAL	When the main clock is not used, set the MOSCCR.MOSTP bit to 1 (general port P37). When this pin is not used as port P37 either, it is configured in the same way as port 1, 2, 7, 9, A, B, D. When the external clock is input to the EXTAL pin, leave this pin open.
Ports 1, 2, 7, 9, Ports A, B, D	• If the direction setting is for input (PORTn.PDR = 0), the corresponding pin is connected to VCC (pulled up) via a resistor or to VSS (pulled down) via a resistor.*1 • If the direction setting is for output (PORTn.PDR = 1), the pin is released.*1, *2
Port 4	• If the direction setting is for input (PORTn.PDR = 0), the corresponding pin is connected to AVCC0 (pulled up) via a resistor or to AVSS0 (pulled down) via a resistor. • If the direction setting is for output (PORTn.PDR = 1), the pin is released.*1, *2

Note 1. Clear the PORTn.PMR bit, the PmnPFS.ISEL bit and the PmnPFS.ASEL bit to 0.
Note 2. In the case of release when the setting is for output, the port is an input over the period from release from the reset state to the pin becoming an output. Since the voltage on the pin is undefined while it is an input, this may lead to an increase in the current drawn.

18. Multi-Function Pin Controller (MPC)

18.1 Overview

The multi-function pin controller (MPC) is used to allocate input and output signals for peripheral modules and input interrupt signals to pins from among multiple ports.

Table 18.1 shows the allocation of pin functions to multiple pins. The symbols ○ and × in the table indicate whether the pins are or are not present on the given package. Allocating the same function to more than one pin is prohibited.

Table 18.1 Allocation of Pin Functions to Multiple Pins (1/3)

Module/Function	Channel	Pin Functions	Allocation Port	Package	
				48-pin	32-pin
Interrupt	NMI	NMI (input)	PE2	○	○
Interrupt	IRQ0	IRQ0 (input)	P10	○	×
			P93	○	○
			PE2	○	○
	IRQ1	IRQ1 (input)	P11	○	○
			P94	○	○
	IRQ2	IRQ2 (input)	P22	○	×
			PB1	○	○
			PD4	○	×
	IRQ3	IRQ3 (input)	P24	○	×
			PB4	○	×
			PD5	○	×
	IRQ4	IRQ4 (input)	P23	○	×
			PA2	○	×
	IRQ5	IRQ5 (input)	P70	○	×
			PB7	○	○
			PD6	○	×
Multi-function timer unit 3	MTU0	MTIOC0A (input/output)	PB3	○	○
			PD3	○	×
		MTIOC0B (input/output)	PB2	○	○
			PD4	○	×
		MTIOC0C (input/output)	PB1	○	○
			PD5	○	×
		MTIOC0D (input/output)	PB0	○	○
			PD6	○	×
	MTU1	MTIOC1A (input/output)	P93	○	○
			PA2	○	×
		MTIOC1B (input/output)	PA3	○	×
			PB6	○	○
	MTU2	MTIOC2A (input/output)	PA3	○	×
			PB0	○	○
		MTIOC2B (input/output)	PA2	○	×
			P94	○	○

Table 18.1 Allocation of Pin Functions to Multiple Pins (2/3)

Module/Function	Channel	Pin Functions	Allocation Port	Package	
				48-pin	32-pin
Multi-function timer unit 3	MTU3	MTIOC3A (input/output)	P11	○	○
			PB6	○	○
		MTIOC3B (input/output)	P71	○	○
		MTIOC3C (input/output)	PB7	○	○
		MTIOC3D (input/output)	P74	○	○
	MTU4	MTIOC4A (input/output)	P72	○	○
		MTIOC4B (input/output)	P73	○	○
		MTIOC4C (input/output)	P75	○	○
		MTIOC4D (input/output)	P76	○	○
	MTU5	MTIC5U (input)	P24	○	×
			P94	○	○
		MTIC5V (input)	P23	○	×
			P93	○	○
		MTIC5W (input)	P22	○	×
			PB1	○	○
	MTU	MTCLKA (input)	P11	○	○
			P94	○	○
			PB1	○	○
		MTCLKB (input)	P10	○	×
			PB0	○	○
		MTCLKC (input)	PB2	○	○
		MTCLKD (input)	PB7	○	○
		ADSM0 (output)	PB2	○	○
Port output enable 3	POE0	POE0# (input)	P70	○	×
	POE8	POE8# (input)	PB4	○	×
			P11	○	○
	POE10	POE10# (input)	PE2	○	○
Serial communications interface	SCI1	RXD1 (input) /SMISO1 (input/output) /SSCL1 (input/output)	PD5	○	×
			PB7	○	○
		TXD1 (output) /SMOSI1 (input/output) /SSDA1 (input/output)	PD3	○	×
			PB6	○	○
		SCK1 (input/output)	PD4	○	×
		CTS1# (input) /RTS1# (output) / SS1# (input)	PD6	○	×
	SCI5	RXD5 (input) /SMISO5 (input/output) /SSCL5 (input/output)	PB1	○	○
			PB7	○	○
			P24	○	×
		TXD5 (output) /SMOSI5 (input/output) /SSDA5 (input/output)	PB2	○	○
			PB6	○	○
			P23	○	×
		SCK5 (input/output)	P93	○	○
			PB3	○	○
		CTS5# (input) /RTS5# (output) / SS5# (input)	PA2	○	×

Table 18.1 Allocation of Pin Functions to Multiple Pins (3/3)

Module/Function	Channel	Pin Functions	Allocation Port	Package	
				48-pin	32-pin
Serial communications interface	SCI12	RXD12 (input) /SMISO12 (input/output) /SSCL12 (input/output) / RXDX12 (input)	P94	○	○
		TXD12 (output) /SMOSI12 (input/output) /SSDA12 (input/output) / TXDX12 (output) /SIOX12 (input/output)	PB0	○	○
		SCK12 (input/output)	PB3	○	○
			P93	○	○
		CTS12# (input) /RTS12# (output) / SS12# (input)	PA3	○	×
I^2C bus interface		SCL0 (input/output)	PB1	○	○
		SDA0 (input/output)	PB2	○	○
12-bit A/D converter		AN000 (input)	P40	○	○
		AN001 (input)	P41	○	○
		AN002 (input)	P42	○	○
		AN003 (input)	P43	○	○
		AN004 (input)	P44	○	○
		AN005 (input)	P45	○	×
		AN006 (input)	P46	○	×
		AN007 (input)	P47	○	×
		ADTRG0# (input)	P93	○	○
			PB5	○	×
		ADST0 (output)	PD6	○	×
Clock frequency accuracy measurement circuit		CACREF (input)	P23	○	×
			PB3	○	○
Comparator		CMPC00 (input)	P40	○	○
		CMPC02 (input)	P43	○	○
		CMPC03 (input)	P46	○	×
		CMPC10 (input)	P41	○	○
		CMPC12 (input)	P44	○	○
		CMPC13 (input)	P47	○	×
		CMPC20 (input)	P42	○	○
		CMPC22 (input)	P45	○	×
		COMP0 (output)	P24	○	×
		COMP1 (output)	P23	○	×
		COMP2 (output)	P22	○	×
		CVREFC0 (input)	P11	○	○

18.2 Register Descriptions

Registers and bits for pins that are not present due to differences according to the package are reserved. Write the value after a reset when writing to such bits.

18.2.1 Write-Protect Register (PWPR)

Address(es): 0008 C11Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	B0WI	PFSWE	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	PFSWE	PFS Register Write Enable	0: Writing to the PFS register is disabled 1: Writing to the PFS register is enabled	R/W
b7	B0WI	PFSWE Bit Write Disable	0: Writing to the PFSWE bit is enabled 1: Writing to the PFSWE bit is disabled	R/W

PFSWE Bit (PFS Register Write Enable)

Writing to PmnPFS register is enabled only when the PFSWE bit is set to 1.
To set the PFSWE bit to 1, write 1 to the PFSWE bit after writing 0 to the B0WI bit.

B0WI Bit (PFSWE Bit Write Disable)

Writing to the PFSWE bit is enabled only when the B0WI bit is set to 0.

18.2.2 P1n Pin Function Control Register (P1nPFS) (n = 0, 1)

Address(es): P10PFS 0008 C148h, P11PFS 0008 C149h

	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 18.2.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P10: IRQ0 (48-pin) P11: IRQ1 (48-/32-pin)	R/W
b7	ASEL	Analog Input Function Select	0: Used other than as analog pin 1: Used as analog pin P11: CVREFC0 (48-/32-pin)	R/W

The Pmn pin function control register (PmnPFS) selects the pin function.

Bits PSEL[4:0] select the peripheral function assigned to each port pin.

The ISEL bit is set when a pin is used as an IRQ input pin. This setting can be used with the combination of the peripheral function, though IRQn (external pin interrupt) of the same number should not be enabled by two or more pins. The ISEL bit to which IRQn is not specified is reserved.

Table 18.2 Register Settings for Input/Output Pin Function in 48-/32-pin

	Register/Pin	
	P10PFS	P11PFS
PSEL[4:0] Settings	P10	P11
00000b (Initial value)	Hi-Z	
00001b	—	MTIOC3A
00010b	MTCLKB	MTCLKA
00111b	—	POE8#

—: Do not specify this value.

18.2.3 P2n Pin Function Control Register (P2nPFS) (n = 2 to 4)

Address(es): P22PFS 0008 C152h, P23PFS 0008 C153h, P24PFS 0008 C154h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 18.3.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P22: IRQ2 (48-pin) P23: IRQ4 (48-pin) P24: IRQ3 (48-pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 18.3 Register Settings for Input/Output Pin Function in 48-pin

	Register/Pin		
	P22PFS	P23PFS	P24PFS
PSEL[4:0] Settings	P22	P23	P24
00000b (Initial value)	Hi-Z		
00001b	MTIC5W	MTIC5V	MTIC5U
00111b	—	CACREF	—
01010b	—	TXD5 SMOSI5 SSDA5	RXD5 SMISO5 SSCL5
11110b	COMP2	COMP1	COMP0

—: Do not specify this value.

18.2.4 P4n Pin Function Control Register (P4nPFS) (n = 0 to 7)

Address(es): P40PFS 0008 C160h, P41PFS 0008 C161h, P42PFS 0008 C162h, P43PFS 0008 C163h, P44PFS 0008 C164h, P45PFS 0008 C165h, P46PFS 0008 C166h, P47PFS 0008 C167h

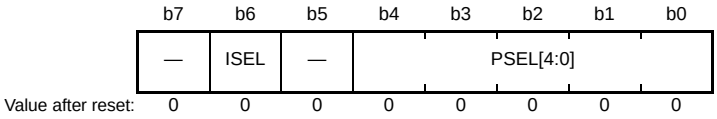
	b7	b6	b5	b4	b3	b2	b1	b0
	ASEL	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ASEL	Analog Input Function Select	0: Not used as an analog pin 1: Used as an analog pin P40: AN000/CMPC00 (48-/32-pin) P41: AN001/CMPC10 (48-/32-pin) P42: AN002/CMPC20 (48-/32-pin) P43: AN003/CMPC02 (48-/32-pin) P44: AN004/CMPC12 (48-/32-pin) P45: AN005/CMPC22 (48-pin) P46: AN006/CMPC03 (48-pin) P47: AN007/CMPC13 (48-pin)	R/W

The ASEL bit is set when a pin is used as an analog pin. The pin state cannot be read at this point.

18.2.5 P7n Pin Function Control Register (P7nPFS) (n = 0 to 6)

Address(es): P70PFS 0008 C178h, P71PFS 0008 C179h, P72PFS 0008 C17Ah, P73PFS 0008 C17Bh, P74PFS 0008 C17Ch, P75PFS 0008 C17Dh, P76PFS 0008 C17Eh



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 18.4.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P70: IRQ5 (48-pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 18.4 Register Settings for Input/Output Pin Function in 48-/32-pin

	Register/Pin						
	P70PFS	P71PFS	P72PFS	P73PFS	P74PFS	P75PFS	P76PFS
PSEL[4:0] Settings	P70	P71	P72	P73	P74	P75	P76
00000b (Initial value)	Hi-Z						
00001b	—	MTIOC3B	MTIOC4A	MTIOC4B	MTIOC3D	MTIOC4C	MTIOC4D
00111b	POE0#	—	—	—	—	—	—

—: Do not specify this value.

18.2.6 P9n Pin Function Control Register (P9nPFS) (n = 3, 4)

Address(es): P93PFS 0008 C18Bh, P94PFS 0008 C18Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 18.5.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin P93: IRQ0 (48-/32-pin) P94: IRQ1 (48-/32-pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 18.5 Register Settings for Input/Output Pin Function in 48-/32-pin

	Register/Pin	
	P93PFS	P94PFS
PSEL[4:0] Settings	P93	P94
00000b (initial value)	Hi-Z	
00001b	MTIOC1A	MTIOC2B
00010b	—	MTCLKA
00011b	MTIC5V	MTIC5U
01001b	ADTRG0#	—
01010b	SCK5	—
01100b	SCK12	RXD12 SMISO12 SSCL12 RXDX12

—: Do not specify this value.

18.2.7 PAn Pin Function Control Register (PAnPFS) (n = 2, 3)

Address(es): PA2PFS 0008 C192h, PA3PFS 0008 C193h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 18.6.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PA2: IRQ4 input switch (48-pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 18.6 Register Settings for Input/Output Pin Function in 48-pin

	Register/Pin	
	PA2PFS	PA3PFS
PSEL[4:0] Settings	PA2	PA3
00000b (initial value)	Hi-Z	
00001b	MTIOC1A	MTIOC1B
00011b	MTIOC2B	MTIOC2A
01010b	CTS5# RTS5# SS5#	—
01100b	—	CTS12# RTS12# SS12#

—: Do not specify this value.

18.2.8 PBn Pin Function Control Register (PBnPFS) (n = 0 to 7)

Address(es): PB0PFS 0008 C198h, PB1PFS 0008 C199h, PB2PFS 0008 C19Ah, PB3PFS 0008 C19Bh, PB4PFS 0008 C19Ch, PB5PFS 0008 C19Dh, PB6PFS 0008 C19Eh, PB7PFS 0008 C19Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 18.7.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PB1: IRQ2 (48-/32-pin) PB4: IRQ3 (48-pin) PB7: IRQ5 (48-/32-pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

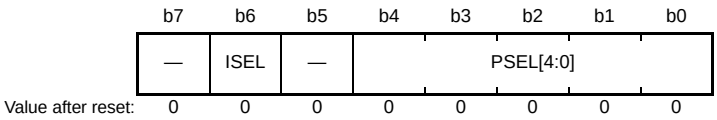
Table 18.7 Register Settings for Input/Output Pin Function in 48-/32-pin

	Register/Pin							
	PB0PFS	PB1PFS	PB2PFS	PB3PFS	PB4PFS	PB5PFS	PB6PFS	PB7PFS
PSEL[4:0] Settings	PB0	PB1	PB2	PB3	PB4	PB5	PB6	PB7
00000b (Initial value)	Hi-Z							
00001b	MTIOC0D	MTIOC0C	MTIOC0B	MTIOC0A	—	—	MTIOC1B	MTIOC3C
00010b	MTCLKB	MTCLKA	MTCLKC	—	—	—	—	MTCLKD
00011b	MTIOC2A	MTIC5W	—	—	—	—	MTIOC3A	—
00111b	—	—	—	CACREF	POE8#	—	—	—
01001b	—	—	ADSM0	—	—	ADTRG0#	—	—
01010b	—	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	SCK5	—	—	TXD5 SMOSI5 SSDA5	RXD5 SMISO5 SSCL5
01011b	—	—	—	—	—	—	TXD1 SMOSI1 SSDA1	RXD1 SMISO1 SSCL1
01100b	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	—	—	SCK12	—	—	—	—
01111b	—	SCL0	SDA0	—	—	—	—	—

—: Do not specify this value.

18.2.9 PDn Pin Function Control Register (PDnPFS) (n = 3 to 6)

Address(es): PD3PFS 0008 C1ABh, PD4PFS 0008 C1ACh, PD5PFS 0008 C1ADh, PD6PFS 0008 C1AEh



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 18.8.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PD4:IRQ2 (48-pin) PD5:IRQ3 (48-pin) PD6:IRQ5 (48-pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 18.8 Register Settings for Input/Output Pin Function in 48-pin

	Register/Pin			
	PD3PFS	PD4PFS	PD5PFS	PD6PFS
PSEL[4:0] Settings	PD3	PD4	PD5	PD6
00000b (initial value)	Hi-Z			
00001b	MTIOC0A	MTIOC0B	MTIOC0C	MTIOC0D
01001b	—	—	—	ADST0
01010b	TXD1 SMOSI1 SSDA1	SCK1	RXD1 SMISO1 SSCL1	CTS1#/ RTS1#/ SS1#

—: Do not specify this value.

18.2.10 PE2 Pin Function Control Register (PE2PFS)

Address(es): PE2PFS 0008 C1B2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	ISEL	—	PSEL[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 18.9.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin PE2:IRQ0 (48-/32-pin)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 18.9 Register Settings for Input/Output Pin Function in 48-/32-pin

	Register/Pin
	PE2PFS
PSEL[4:0] Settings	PE2
00000b (Initial value)	Hi-Z
00111b	POE10#

Note: The priority is given to NMI pin interrupt operation when the NMIER.NMIEN = 1

18.3 Usage Notes

18.3.1 Procedure for Specifying Input/Output Pin Function

Use the following procedure to specify the input/output pin functions.

(1) Clear the port mode register (PMR) to 0 to select the general I/O port function.
(2) Specify the assignments of input/output signals for peripheral functions to the desired pins.
(3) Enable writing to the Pmn pin function control register (PmnPFS) through the write-protect register (PWPR) setting. (m = 1, 2, 7, 9, A, B, D, E; n = 0 to 7)
(4) Specify the input/output function for the pin through the PSEL[4:0] bit settings in the PmnPFS register.
(5) Clear the PFSWE bit in the PWPR register to 0 to disable writing to the PmnPFS register.
(6) Set the PMR to 1 as necessary to switch to the selected input/output function for the pin.

18.3.2 Notes on MPC Register Setting

(1) Settings of the Pmn pin function control register (PmnPFS) should be made only while the PMR register for the target pin is cleared to 0. If a Pmn pin function control register is set while the PMR register of corresponding pin is 1, unexpected edges may be input through the input pin or unexpected pulses are output through the output pin.
(2) Only the allowed values (functions) should be specified in the Pmn pin function control registers. 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 MPC settings.
(4) Analog input functions for the A/D converter are multiplexed with input pins of port 4. If a pin is to be used as an analog input, avoid loss of resolution by setting the given bits of the port mode register (PMR) and of the port direction register (PDR) to 0, i.e. configuring the pin as a general-purpose input, and setting the PmnPFS.ASEL bit to 1.
(5) Points to note regarding the port mode register (PMR), port direction register (PDR), and Pmn pin function control register (PmnPFS) settings for pins that have multiplexed pin functions are listed in Table 18.10.
The pin state is readable when the PmnPFS.ASEL bit is 0.
If the value of the PmnPFS.PSEL[4:0] bits is to be changed, do so while the PMR.Bj bit is 0.

Table 18.10 Register Settings

Item	PMR.Bn	PDR.Bn	PmnPFS			Point to Note
			ASEL	ISEL	PSEL[4:0]	
After a reset	0	0	0	0	00000b	Pins function as general input port pins after release from the reset state.
General input ports	0	0	0	0/1	x	Set the ISEL bit to 1 if these are multiplexed with interrupt inputs.
General output ports	0	1	0	0	x	
Peripheral functions	1	x	0	0/1	Peripheral functions (see Table 18.2 to Table 18.9)	Set the ISEL bit to 1 if these are multiplexed with interrupt inputs.
Interrupt inputs	0	0	0	1	x	
NMI	x	x	x	x	x	Register settings are not required.
Analog inputs	0*2	0	1	x*1	x	Set these as general input port pins so that the output buffers are turned off.

x: Setting not required.
0/1: Setting the PmnPFS.ISEL bit to 0 makes the pin incapable of functioning as an IRQ pin.
Setting the PmnPFS.ISEL bit to 1 makes the pin capable of functioning as an IRQ pin (if the IRQ is selected from the multiplexed functions).
Note: - The pin state is readable when the PmnPFS.ASEL bit is 0.
- If the value of the PmnPFS.PSEL[4:0] bits is to be changed, do so while the PMR.Bn bit is 0.
- If an RIIC function is assigned to a port pin, clear the PCR.Bn (to 0); pulling up is automatically turned off for outputs from peripheral modules other than the RIIC.
Note 1. The pin does not function as the IRQn input pin even if the PmnPFS.ISEL bit is set to 1.
Note 2. Setting PORT4 is not required.

18.3.3 Note on Using Analog Functions

When an analog function is in use, configure the pin as a general-purpose input by setting the given bits of the port mode register (PMR) and of the port direction register (PDR) to 0, and then set the ASEL bit in the Pmn pin function control register (PmnPFS) to 1.

18.3.4 Note on PB1 Pin Input Level

PB1 input level is specified to TTL when SCL is selected in the PB1PFS.PSEL bit and SMBus is selected in the ICMR3.SMBS bit in RIIC. At this time, input levels of the PB1 port read and the IRQ2 also become TTL.

19. Multi-Function Timer Pulse Unit 3 (MTU3c)

19.1 Overview

This MCU has an on-chip multi-function timer pulse unit 3 (MTU3c), consisting of six 16-bit timer channels. Table 19.1 shows the specifications of the MTU and Table 19.2 lists the functions of the MTU. Figure 19.1 shows a block diagram of the MTU.

Table 19.1 MTU Specifications

Item	Description
Pulse input/output	16 lines max.
Pulse input	3 lines
Count clock	11 clocks for each channel (14 clocks for MTU0, 12 clocks for MTU2, 10 clocks for MTU5, and four clocks for MTU1 & MTU2 (LWA = 1))
Available operations	[MTU0 to MTU4] • Waveform output on compare match • Input capture function (noise filter setting available) • Counter-clearing operation • Simultaneous writing to multiple timer counters (TCNT) • Simultaneous clearing on compare match or input capture • Simultaneous input and output to registers in synchronization with counter operations • Up to 12-phase PWM output in combination with synchronous operation
	[MTU0, MTU3, MTU4] • 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] • Through interlocked operation of MTU3/4, the positive and negative signals in six phases 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 or troughs of the timer-counter values or writing to the buffer registers (MTU4.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-synchronized 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
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	28 sources
Buffer operation	Automatic transfer of register data (transfer from the buffer register to the timer register)
Trigger generation	A/D conversion start triggers can be generated
	A/D conversion 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 19.2 MTU Functions (1/2)

Item		MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4	MTU5
Count clock		PCLKB/1 PCLKB/2 PCLKB/4 PCLKB/8 PCLKB/16 PCLKB/32 PCLKB/64 PCLKB/256 PCLKB/1024 MTCLKA MTCLKB MTCLKC MTCLKD MTIOC1A	PCLKB/1 PCLKB/2 PCLKB/4 PCLKB/8 PCLKB/16 PCLKB/32 PCLKB/64 PCLKB/256 PCLKB/1024 MTCLKA MTCLKB	PCLKB/1 PCLKB/2 PCLKB/4 PCLKB/8 PCLKB/16 PCLKB/32 PCLKB/64 PCLKB/256 PCLKB/1024 MTCLKA MTCLKB MTCLKC	MTCLKA MTCLKB MTCLKC MTCLKD	PCLKB/1 PCLKB/2 PCLKB/4 PCLKB/8 PCLKB/16 PCLKB/32 PCLKB/64 PCLKB/256 PCLKB/1024 MTCLKA MTCLKB	PCLKB/1 PCLKB/2 PCLKB/4 PCLKB/8 PCLKB/16 PCLKB/32 PCLKB/64 PCLKB/256 PCLKB/1024 MTCLKA MTCLKB	PCLKB/1 PCLKB/2 PCLKB/4 PCLKB/8 PCLKB/16 PCLKB/32 PCLKB/64 PCLKB/256 PCLKB/1024 MTIOC1A
External clock for phase counting mode		—	MTCLKA MTCLKB	MTCLKA MTCLKB MTCLKC MTCLKD	MTCLKA MTCLKB MTCLKC MTCLKD	—	—	—
General registers (TGR)		TGRA TGRB TGRE	TGRA TGRB	TGRA TGRB	TRGALW TRGBLW	TGRA TGRB	TGRA TGRB	TGRU TGRV TGRW
General registers/ buffer registers		TGRC TGRD TGRF	—	—	—	TGRC TGRD TGRE	TGRC TGRD TGRE TGRF	—
I/O pins		MTIOC0A MTIOC0B MTIOC0C MTIOC0D	MTIOC1A MTIOC1B	MTIOC2A MTIOC2B	MTIOC1A MTIOC1B	MTIOC3A MTIOC3B MTIOC3C MTIOC3D	MTIOC4A MTIOC4B MTIOC4C MTIOC4D	MTIC5U MTIC5V MTIC5W
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGRALW/TGRBLW 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	✓	✓	✓
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		—	—	—	—	—	—	✓
DTC trigger sources		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGRALW/TGRBLW input capture	TGR compare match or input capture	TGR compare match or input capture and TCNT overflow/ underflow*2	TGR compare match or input capture
A/D conversion 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	—

Table 19.2 MTU Functions (2/2)

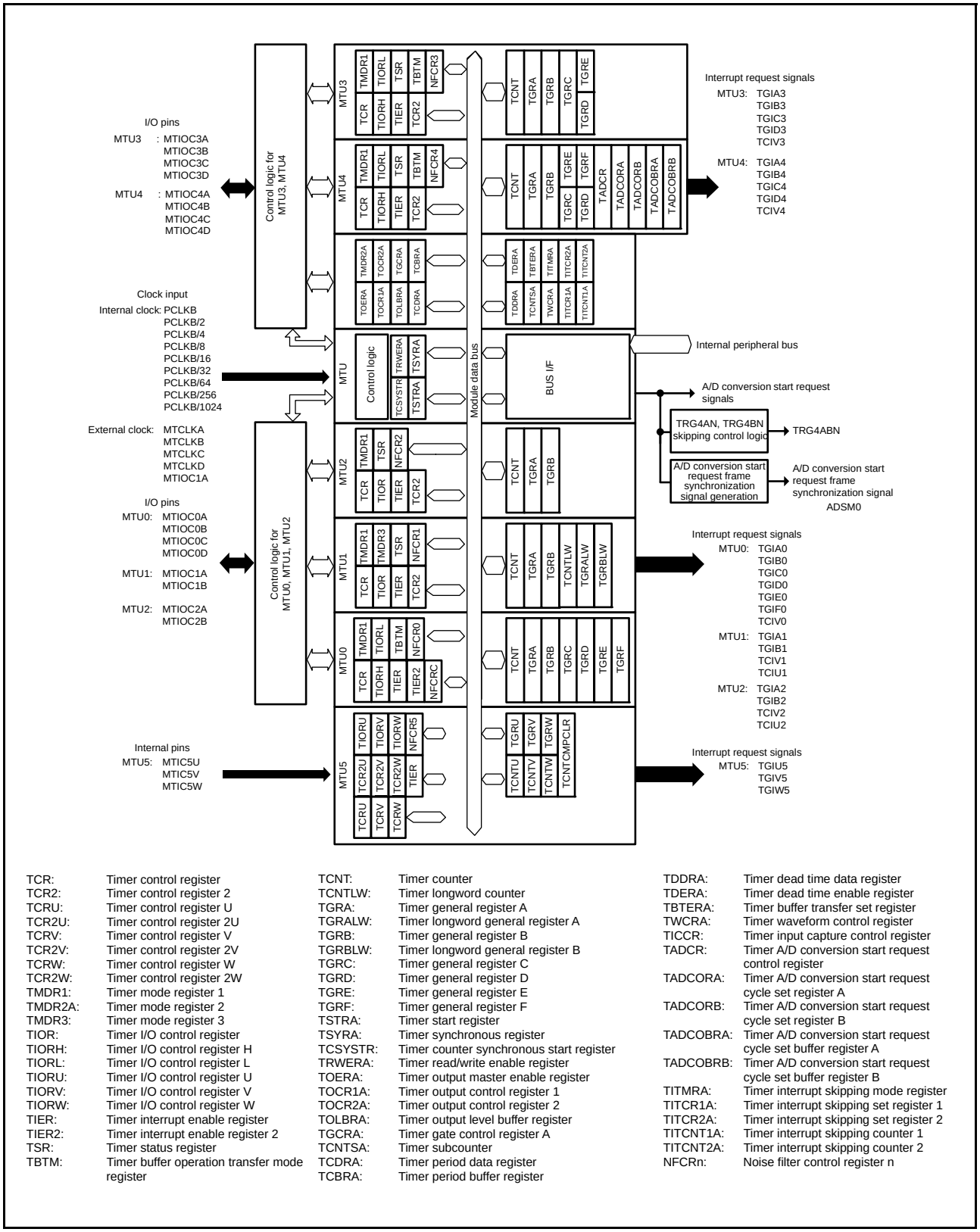
Item	MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4	MTU5
Interrupt sources	Seven sources • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Compare match 0E • Compare match 0F • Overflow	Four sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	Four sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	Four sources • Input capture 1A • Input capture 1B • Overflow • Underflow	Five sources • Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	Five sources • Compare match or input capture 4A • Compare match or input capture 4B • Compare match or input capture 4C • Compare match or input capture 4D • Overflow or underflow*2	Three sources • Compare match or input capture 5U • Compare match or input capture 5V • Compare match or input capture 5W
A/D conversion start request delaying function	—	—	—	—	—	A/D conversion start request at a match between TADCORA and TCNT or A/D conversion start request at a match between TADCORB and TCNT	—
Interrupt skipping 1	—	—	—	—	Skips TGRA compare match interrupts	Skips TCIV interrupts	—
Interrupt skipping 2	—	—	—	—	—	Skipping in compare count between TADCORA and TCNT, and TADCORB and TCNT	—
Module stop function	MSTPCRA.MSTPA9*3						

✓: Possible —: Not possible

Note 1. When LWA is 1, the TGRALW capture source can be selected from either of the following: an input from MTIOC1A or MTU0.TGRA compare match/input capture event.
The TGRBLW capture source can be selected from any of the following: an input from MTIOC1B, MTU0.TGRC compare match/input capture event.

Note 2. Underflow is available only in complementary PWM mode.

Note 3. For details on the module stop function, refer to section 11, Low Power Consumption.



TCR:	Timer control register	TCNT:	Timer counter	TDDRA:	Timer dead time data register
TCR2:	Timer control register 2	TCNTLW:	Timer longword counter	TDERA:	Timer dead time enable register
TCRU:	Timer control register U	TGRA:	Timer general register A	TBTERA:	Timer buffer transfer set register
TCR2U:	Timer control register 2U	TGRALW:	Timer longword general register A	TWCRA:	Timer waveform control register
TCRV:	Timer control register V	TGRB:	Timer general register B	TICCR:	Timer input capture control register
TCR2V:	Timer control register 2V	TGRBLW:	Timer longword general register B	TADCR:	Timer A/D conversion start request control register
TCRW:	Timer control register W	TGRC:	Timer general register C	TADCORA:	Timer A/D conversion start request cycle set register A
TCR2W:	Timer control register 2W	TGRD:	Timer general register D	TADCORB:	Timer A/D conversion start request cycle set register B
TMDR1:	Timer mode register 1	TGRE:	Timer general register E	TADCOBRA:	Timer A/D conversion start request cycle set buffer register A
TMDR2A:	Timer mode register 2	TGRF:	Timer general register F	TADCOBRB:	Timer A/D conversion start request cycle set buffer register B
TMDR3:	Timer mode register 3	TSTRA:	Timer start register	TITMRA:	Timer interrupt skipping mode register
TIOR:	Timer I/O control register	TSYRA:	Timer synchronous register	TITCR1A:	Timer interrupt skipping set register 1
TIORH:	Timer I/O control register H	TCSYSTR:	Timer counter synchronous start register	TITCR2A:	Timer interrupt skipping set register 2
TIORL:	Timer I/O control register L	TRWERA:	Timer read/write enable register	TITCNT1A:	Timer interrupt skipping counter 1
TIORU:	Timer I/O control register U	TOERA:	Timer output master enable register	TITCNT2A:	Timer interrupt skipping counter 2
TIORV:	Timer I/O control register V	TOCR1A:	Timer output control register 1	NFCRn:	Noise filter control register n
TIORW:	Timer I/O control register W	TOCR2A:	Timer output control register 2		
TIER:	Timer interrupt enable register	TOLBRA:	Timer output level buffer register		
TIER2:	Timer interrupt enable register 2	TGCRA:	Timer gate control register A		
TSR:	Timer status register	TCNTSA:	Timer subcounter		
TBTM:	Timer buffer operation transfer mode register	TCDRA:	Timer period data register		
		TCBRA:	Timer period buffer register		

Figure 19.1 Block Diagram of MTU (MTU0 to MTU5)

Table 19.3 shows the configuration of pins for the MTU.

Table 19.3 Pin Configuration of the MTU

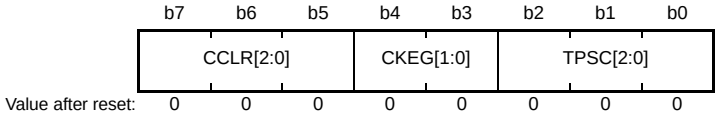
Channel	Pin Name	I/O	Function
MTU	MTCLKA	Input	External clock A input pin (MTU1 and MTU2 phase counting mode A phase input)
	MTCLKB	Input	External clock B input pin (MTU1 and MTU2 phase counting mode B phase input)
	MTCLKC	Input	External clock C input pin (MTU2 phase counting mode A phase input)
	MTCLKD	Input	External clock D input pin (MTU2 phase counting mode B phase input)
	ADSM0	Output	A/D conversion start request frame synchronization signal 0 output pin
MTU0	MTIOC0A	I/O	MTU0 TGRA input capture input/output compare output/PWM output pin
	MTIOC0B	I/O	MTU0 TGRB input capture input/output compare output/PWM output pin
	MTIOC0C	I/O	MTU0 TGRC input capture input/output compare output/PWM output pin
	MTIOC0D	I/O	MTU0 TGRD input capture input/output compare output/PWM output pin
MTU1	MTIOC1A	I/O	MTU1 TGRA input capture input/output compare output/PWM output pin
	MTIOC1B	I/O	MTU1 TGRB input capture input/output compare output/PWM output pin
MTU2	MTIOC2A	I/O	MTU2 TGRA input capture input/output compare output/PWM output pin
	MTIOC2B	I/O	MTU2 TGRB input capture input/output compare output/PWM output pin
MTU3	MTIOC3A	I/O	MTU3 TGRA input capture input/output compare output/PWM output pin
	MTIOC3B	I/O	MTU3 TGRB input capture input/output compare output/PWM output pin
	MTIOC3C	I/O	MTU3 TGRC input capture input/output compare output/PWM output pin
	MTIOC3D	I/O	MTU3 TGRD input capture input/output compare output/PWM output pin
MTU4	MTIOC4A	I/O	MTU4 TGRA input capture input/output compare output/PWM output pin
	MTIOC4B	I/O	MTU4 TGRB input capture input/output compare output/PWM output pin
	MTIOC4C	I/O	MTU4 TGRC input capture input/output compare output/PWM output pin
	MTIOC4D	I/O	MTU4 TGRD 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

19.2 Register Descriptions

19.2.1 Timer Control Register (TCR)

- MTU0.TCR, MTU1.TCR, MTU2.TCR, MTU3.TCR, MTU4.TCR

Address(es): MTU0.TCR 0009 5300h, MTU1.TCR 0009 5380h, MTU2.TCR 0009 5400h, MTU3.TCR 0009 5200h, MTU4.TCR 0009 5201h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC[2:0]	Time Prescaler Select	Refer to Table 19.6 to Table 19.9.	R/W
b4, b3	CKEG[1:0]	Clock Edge Select	b4 b3 0 0: Count at rising edge 0 1: Count at falling edge 1 x: Count at both edges	R/W
b7 to b5	CCLR[2:0]	Counter Clear Source Select	Refer to Table 19.4 and Table 19.5.	R/W

x: Don’t care

The TCR register controls the TCNT operation for each channel in combination with the TCR2 register. The MTU has a total of eight TCR registers, one each for MTU0 to MTU4 and three (TCRU, TCRV, and TCRW) for MTU5. TCR values should be specified only while TCNT operation is stopped.

TPSC[2:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock source. The count clock source can be selected independently for each channel. Refer to Table 19.6 to Table 19.9 for details.

CKEG[1:0] Bits (Clock Edge Select)

These bits select the clock edge, including the MTIOC1A pin. When the internal clock is counted at both edges, the count clock period is halved (e.g. PCLKB/4 at both edges = PCLKB/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 count clock source is PCLKB/2 or slower. When PCLKB/1 or the overflow/underflow in another channel is selected for the count clock source, 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. Refer to Table 19.4 and Table 19.5 for details.

Table 19.4 CCLR[2:0] (MTU0, MTU3, MTU4)

Channel	Bit 7 CCLR[2]	Bit 6 CCLR[1]	Bit 5 CCLR[0]	Description
MTU0 MTU3 MTU4	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 bit to 1
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 19.5 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 (when LWA = 0) TCNTLW cleared by TGRALW input capture (when LWA = 1)
	0	1	0	TCNT cleared by TGRB compare match/input capture (when LWA = 0) TCNTLW cleared by TGRBLW input capture (when LWA = 1)
	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 bit to 1.
Note 2. Bit 7 is reserved in MTU1 and MTU2. It is read as 0. The write value is ignored.

- MTU5.TCRU, MTU5.TCRV, MTU5.TCRW

Address(es): MTU5.TCRU 0009 5484h, MTU5.TCRV 0009 5494h, MTU5.TCRW 0009 54A4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TPSC[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TPSC[1:0]	Time Prescaler Select	Refer to Table 19.10.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TPSC[1:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock source. Refer to Table 19.10 for details.

19.2.2 Timer Control Register 2 (TCR2)

- MTU0.TCR2, MTU3.TCR2, MTU4.TCR2

Address(es): MTU0.TCR2 0009 5328h, MTU3.TCR2 0009 524Ch, MTU4.TCR2 0009 524Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

- MTU1.TCR2, MTU2.TCR2

Address(es): MTU1.TCR2 0009 5394h, MTU2.TCR2 0009 540Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	PCB[1:0]		TPSC2[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC2[2:0]	Time Prescaler Select	Refer to Table 19.6 to Table 19.9.	R/W
b4, b3	PCB[1:0]	Phase Counting Mode Function Expansion Control	Functional Expansion Control for Phase Counting Modes 2, 3, and 5	R/W
b7 to b5	—	Reserved	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. The MTU has a total of eight TCR2 registers, one each for MTU0 to MTU4 and three (TCR2U, TCR2V, and TCR2W) for MTU5. TCR2 values should be specified only while TCNT operation is stopped.

TPSC2[2:0] Bits (Time Prescaler Select)

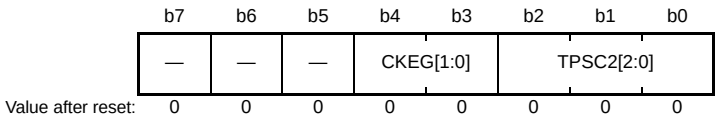
These bits select the TCNT count clock source. The count clock source can be selected independently for each channel. Refer to Table 19.6 to Table 19.9 for details.

PCB[1:0] Bits (Phase Counting Mode Function Expansion Control)

These bits control extended functions for phase counting mode 2, 3, and 5 in MTU1 and MTU2. Refer to section 19.3.6, Phase Counting Mode.

- MTU5.TCR2U, MTU5.TCR2V, MTU5.TCR2W

Address(es): MTU5.TCR2U 0009 5485h, MTU5.TCR2V 0009 5495h, MTU5.TCR2W 0009 54A5h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC2[2:0]	Time Prescaler Select	Refer to Table 19.10.	R/W
b4, b3	CKEG[1:0]	Clock Edge Select	b4 b3 0 0: Counts at the rising edge. 0 1: Counts at the falling edge. 1 x: Counts at both edges.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

x: Don't care

TPSC2[2:0] Bits (Time Prescaler Select)

These bits select the TCNT count clock source. Refer to Table 19.10 for details.

CKEG[1:0] Bits (Clock Edge Select)

These bits select the edge of the count clock signal input from the MTIOC1A pin.

Table 19.6 TPSC[2:0], TPSC2[2:0] (MTU0)

	TCR2 register			TCR register			
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	Description
MTU0	0	0	0	0	0	0	Internal clock: counts on PCLKB/1
	0	0	0	0	0	1	Internal clock: counts on PCLKB/4
	0	0	0	0	1	0	Internal clock: counts on PCLKB/16
	0	0	0	0	1	1	Internal clock: counts on PCLKB/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 PCLKB/2
	0	1	0	x	x	x	Internal clock: counts on PCLKB/8
	0	1	1	x	x	x	Internal clock: counts on PCLKB/32
	1	0	0	x	x	x	Internal clock: counts on PCLKB/256
	1	0	1	x	x	x	Internal clock: counts on PCLKB/1024
	1	1	0	x	x	x	Setting prohibited
	1	1	1	x	x	x	External clock: counts on MTIOC1A pin input

x: Don't care

Table 19.7 TPSC[2:0], TPSC2[2:0] (MTU1)

	TCR2 register			TCR register			
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	Description
MTU1	0	0	0	0	0	0	Internal clock: counts on PCLKB/1
	0	0	0	0	0	1	Internal clock: counts on PCLKB/4
	0	0	0	0	1	0	Internal clock: counts on PCLKB/16
	0	0	0	0	1	1	Internal clock: counts on PCLKB/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 PCLKB/256
	0	0	0	1	1	1	Overflow/underflow of MTU2.TCNT
	0	0	1	x	x	x	Internal clock: counts on PCLKB/2
	0	1	0	x	x	x	Internal clock: counts on PCLKB/8
	0	1	1	x	x	x	Internal clock: counts on PCLKB/32
	1	0	0	x	x	x	Internal clock: counts on PCLKB/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

x: Don't care

Note: This setting has no effect when MTU1 is in phase counting mode.

Table 19.8 TPSC[2:0], TPSC2[2:0] (MTU2)

	TCR2 register			TCR register			
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	Description
MTU2	0	0	0	0	0	0	Internal clock: counts on PCLKB/1
	0	0	0	0	0	1	Internal clock: counts on PCLKB/4
	0	0	0	0	1	0	Internal clock: counts on PCLKB/16
	0	0	0	0	1	1	Internal clock: counts on PCLKB/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 PCLKB/1024
	0	0	1	x	x	x	Internal clock: counts on PCLKB/2
	0	1	0	x	x	x	Internal clock: counts on PCLKB/8
	0	1	1	x	x	x	Internal clock: counts on PCLKB/32
	1	0	0	x	x	x	Internal clock: counts on PCLKB/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

x: Don't care

Note: This setting has no effect when the MTU2 is in phase counting mode.

Table 19.9 TPSC[2:0], TPSC2[2:0] (MTU3, MTU4)

	TCR2 register			TCR register			
	Bit 2	Bit 1	Bit 0	Bit 2	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[2]	TPSC[1]	TPSC[0]	Description
MTU3 MTU4	0	0	0	0	0	0	Internal clock: counts on PCLKB/1
	0	0	0	0	0	1	Internal clock: counts on PCLKB/4
	0	0	0	0	1	0	Internal clock: counts on PCLKB/16
	0	0	0	0	1	1	Internal clock: counts on PCLKB/64
	0	0	0	1	0	0	Internal clock: counts on PCLKB/256
	0	0	0	1	0	1	Internal clock: counts on PCLKB/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 PCLKB/2
	0	1	0	x	x	x	Internal clock: counts on PCLKB/8
	0	1	1	x	x	x	Internal clock: counts on PCLKB/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

x: Don't care

Table 19.10 TPSC[1:0], TPSC2[2:0] (MTU5)

	TCR2 register			TCR register		
	Bit 2	Bit 1	Bit 0	Bit 1	Bit 0	
Channel	TPSC2[2]	TPSC2[1]	TPSC2[0]	TPSC[1]	TPSC[0]	Description
MTU5	0	0	0	0	0	Internal clock: counts on PCLKB/1
	0	0	0	0	1	Internal clock: counts on PCLKB/4
	0	0	0	1	0	Internal clock: counts on PCLKB/16
	0	0	0	1	1	Internal clock: counts on PCLKB/64
	0	0	1	x	x	Internal clock: counts on PCLKB/2
	0	1	0	x	x	Internal clock: counts on PCLKB/8
	0	1	1	x	x	Internal clock: counts on PCLKB/32
	1	0	0	x	x	Internal clock: counts on PCLKB/256
	1	0	1	x	x	Internal clock: counts on PCLKB/1024
	1	1	0	x	x	Setting prohibited
	1	1	1	x	x	Internal clock: counts on MTIOC1A pin input

x: Don't care

19.2.3 Timer Mode Register 1 (TMDR1)

- MTU0.TMDR1

Address(es): MTU0.TMDR1 0009 5301h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	BFE	BFB	BFA	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

- MTU1.TMDR1, MTU2.TMDR1

Address(es): MTU1.TMDR1 0009 5381h, MTU2.TMDR1 0009 5401h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

- MTU3.TMDR1, MTU4.TMDR1

Address(es): MTU3.TMDR1 0009 5202h, MTU4.TMDR1 0009 5203h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	BFB	BFA	MD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MD[3:0]	Mode Select	These bits specify the timer operating mode. Refer to Table 19.11 for details.	R/W
b4	BFA	Buffer Operation A	0: TGRA and TGRC operate normally 1: TGRA and TGRC used together for buffer operation	R/W
b5	BFB	Buffer Operation B	0: TGRB and TGRD operate normally 1: TGRB and TGRD used together for buffer operation	R/W
b6	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
b7	—	Reserved	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 five TMDR1 registers, one each for MTU0 to MTU4. TMDR1 register values should be specified only while TCNT operation is stopped.

Table 19.11 Operating Mode Setting by MD[3:0] Bits (MTU0 to MTU4)

Bit 3	Bit 2	Bit 1	Bit 0								
MD[3]	MD[2]	MD[1]	MD[0]	Description	MTU0	MTU1	MTU2	MTU1 & MTU2 (LWA = 1)	MTU3	MTU4
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	x	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					✓	

x: Don't care

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.
When MTU3 is set to reset-synchronized PWM mode or complementary PWM mode, the MTU4 settings become ineffective and automatically conform to the MTU3 setting, respectively. MTU4 should be set to the initial values (normal mode).

BFA Bit (Buffer Operation A)

This 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 modes other than complementary PWM mode, but compare match with TGRC occurs in complementary PWM mode.

If a compare match occurs on MTU4 in the Tb interval in complementary PWM mode, the TGIEC bit in timer interrupt enable register (MTU4.TIER) should be set to 0.

In reset-synchronized PWM mode or complementary PWM mode, buffer operation in MTU3 and MTU4 depends on the settings in the BFA bit of MTU3.TMDR1. The BFA bit of MTU4.TMDR1 should be set 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.

Refer to Figure 19.47 for an illustration of the Tb interval in complementary PWM mode.

BFB Bit (Buffer Operation B)

This 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 take place in modes other than complementary PWM mode, but compare match with TGRD occurs in complementary PWM mode.

If a compare match occurs on MTU4 in the Tb interval in complementary PWM mode, the TGIED bit in timer interrupt enable register (MTU4.TIER) should be set to 0.

In reset-synchronized PWM mode or complementary PWM mode, buffer operation in MTU3 and MTU4 depends on the settings in the BFB bit of MTU3.TMDR1. The BFB bit of MTU4.TMDR1 should be set 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.

Refer to Figure 19.47 for an illustration of the Tb interval in complementary PWM mode.

BFE Bit (Buffer Operation E)

This bit specifies whether to operate MTU0.TGRE and MTU0.TGRF in the normal way or to use them together for buffer operation. Compare match with TGRF occurs even when TGRF is used as a buffer register.
In MTU0 to MTU4, this bit is reserved. It is read as 0. The write value should be 0.

19.2.4 Timer Mode Register 2 (TMDR2A)

Address(es): MTU.TMDR2A 0009 5270h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DRS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DRS	Double Buffer Select	0: Double buffer function is disabled 1: Double buffer function is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TMDR2A specifies the double buffer function in complementary PWM mode 3 (transfer at the crest and trough of the counter value). TMDR2A value 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.

19.2.5 Timer Mode Register 3 (TMDR3)

Address(es): MTU1.TMDR3 0009 5391h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PHCKSEL	LWA
Value after reset:	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	LWA	MTU1/MTU2 Combination Longword Access Control	0: 16-bit access is enabled. 1: 32-bit access is enabled.	R/W
b1	PHCKSEL	External Input Phase Clock Select	0: MTCLKA and MTCLKB are selected for the external phase clock. 1: MTCLKC and MTCLKD are selected for the external phase clock.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TMDR3 register controls longword access to a 32-bit register or counter in a combination of MTU1 and MTU2. There is only one TMDR3 register in MTU1. The counter (TCNTLW), general register A (TGRALW), and general register B (TGRBLW) of MTU1 and MTU2 are accessed in the combinations listed in Table 19.12.

LWA Bit (MTU1/MTU2 Combination Longword Access Control)

This bit selects a 32-bit access in a combination of MTU1 and MTU2.

When LWA is set to 0, the MTU1 and MTU2 independently operate as a 16-bit timer. therefore registers TCNTLW, TGRALW, and TGRBLW cannot be accessed.

When LWA is set to 1, MTU1 and MTU2 operate as a 32-bit cascaded timer and the timer is controlled by registers MTU1.TCR, MTU1.TCR2, MTU1.TIOR, and MTU1.TMDR1. The settings of registers MTU2.TCR, MTU2.TCR2, MTU2.TIOR, and MTU2.TMDR1 are disabled and the 16-bit registers (TCNT, TGRA, and TGRB) in MTU1 and MTU2 cannot be accessed. Furthermore, MTU2 input capture and compare match are also disabled.

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, PWM1 mode, or PWM2 mode. Select phase counting mode when setting the LWA bit to 1.

Initialize the registers TCNT, TGRA, and TGRB in MTU1 and MTU2 in advance before setting the LWA bit to 1.

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 either the A- or B-phase signal from the external clock. Refer to Table 19.50, Clock Input Pins in Phase Counting Mode for details.

Table 19.12 Setting and Combination of the TMDR3 Register

	TMDR3.LWA = 0		TMDR3.LWA = 1	
Register	**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.

19.2.6 Timer I/O Control Register (TIOR)

- MTU0.TIORH, MTU1.TIOR, MTU2.TIOR, MTU3.TIORH, MTU4.TIORH

Address(es): MTU0.TIORH 0009 5302h, MTU1.TIOR 0009 5382h, MTU2.TIOR 0009 5402h, MTU3.TIORH 0009 5204h, MTU4.TIORH 0009 5206h

	b7	b6	b5	b4	b3	b2	b1	b0
	IOB[3:0]				IOA[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOA[3:0]	I/O Control A*1	Refer to the following tables. MTU0.TIORH: Table 19.21 MTU1.TIOR: Table 19.23 MTU2.TIOR: Table 19.24 MTU3.TIORH: Table 19.25 MTU4.TIORH: Table 19.27	R/W
b7 to b4	IOB[3:0]	I/O Control B*1	Refer to the following tables. MTU0.TIORH: Table 19.13 MTU1.TIOR: Table 19.15 MTU2.TIOR: Table 19.16 MTU3.TIORH: Table 19.17 MTU4.TIORH: Table 19.19	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.

- MTU0.TIORL, MTU3.TIORL, MTU4.TIORL

Address(es): MTU0.TIORL 0009 5303h, MTU3.TIORL 0009 5205h, MTU4.TIORL 0009 5207h

	b7	b6	b5	b4	b3	b2	b1	b0
	IOD[3:0]				IOC[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOC[3:0]	I/O Control C*1	Refer to the following tables. MTU0.TIORL: Table 19.22 MTU3.TIORL: Table 19.26 MTU4.TIORL: Table 19.28	R/W
b7 to b4	IOD[3:0]	I/O Control D*1	Refer to the following tables. MTU0.TIORL: Table 19.14 MTU3.TIORL: Table 19.18 MTU4.TIORL: Table 19.20	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.

- MTU5.TIORU, MTU5.TIORV, MTU5.TIORW

Address(es): MTU5.TIORU 0009 5486h, MTU5.TIORV 0009 5496h, MTU5.TIORW 0009 54A6h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	IOC[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	IOC[4:0]	I/O Control C	Refer to the following table. MTU5.TIORU, MTU5.TIORV, MTU5.TIORW: Table 19.29	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TIOR register controls the TGR register. The MTU has a total of 11 TIOR registers, two each for MTU0, MTU3, and MTU4, one each for MTU1 and MTU2, and three (MTU5.TIORU/TIORV/TIORW) for MTU5. The TIOR register should be set when the TMDR register setting is normal mode, PWM mode, or phase counting mode.
Note that TIOR is affected by the TMDR1 setting.
The initial output specified by TIOR is valid when the counter is stopped (the CSTn bit in TSTRA is set to 0). Note also that, in PWM mode 2, the output at the point at which the counter becomes 0000h is specified.
When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

Table 19.13 TIORH (MTU0)

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

x: Don't care

Note 1. When PCLKB/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKB/1 as the count clock for MTU1.

Table 19.14 TIORL (MTU0)

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

x: Don't care

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 PCLKB/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKB/1 as the count clock for MTU1.

Table 19.15 TIOR (MTU1)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB[3]	IOB[2]	IOB[1]	IOB[0]	TGRB/TGRBLW Register Function	MTIOC1B Pin Function
0	0	0	0	Output compare register (only available when LWA = 0)	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 occurrence of compare match or input capture in the MTU0.TGRC register

x: Don't care

Table 19.16 TIOR (MTU2)

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.
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.

x: Don't care

Table 19.17 TIORH (MTU3)

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.

x: Don't care

Table 19.18 TIORL (MTU3)

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.

x: Don't care

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.

Table 19.19 TIORH (MTU4)

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.

x: Don't care

Table 19.20 TIORL (MTU4)

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.

x: Don't care

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.

Table 19.21 TIORH (MTU0)

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	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

x: Don't care

Note 1. When PCLKB/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKB/1 as the count clock for MTU1.

Table 19.22 TIORL (MTU0)

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

x: Don't care

Note 1. When the MTU0.TMDR1.BFB 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 PCLKB/1 is selected as the count clock for MTU1, MTU0 input capture is not generated. Do not select PCLKB/1 as the count clock for MTU1.

Table 19.23 TIOR (MTU1)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA[3]	IOA[2]	IOA[1]	IOA[0]	TGRA/TGRALW Register Function	MTIOC1A Pin Function
0	0	0	0	Output compare register (only available when LWA = 0)	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.

x: Don't care

Table 19.24 TIOR (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.

x: Don't care

Table 19.25 TIORH (MTU3)

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.

x: Don't care

Table 19.26 TIORL (MTU3)

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.

x: Don't care

Note 1. When the MTU3.TMDR1.BFB 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 19.27 TIORH (MTU4)

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.

x: Don't care

Table 19.28 TIORL (MTU4)

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.

x: Don't care

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 19.29 TIORU, TIORV, and TIORW (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 Registers 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*1	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		Setting prohibited
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.

x: Don't care

Note 1. Set the IOC[4:0] bits to 19h, 1Ah, 1Bh, 1Dh, 1Eh, or 1Fh only when using external pulse width measurement or only when using dead time compensation linked with MTU3 and MTU4. For details, refer to section 19.3.11, External Pulse Width Measurement and section 19.3.12, Dead Time Compensation.

19.2.7 Timer Compare Match Clear Register (TCNTCMPCLR)

Address(es): MTU5.TCNTCMPCLR 0009 54B6h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CMPCLR5U	CMPCLR5V	CMPCLR5W
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPCLR5W	TCNT Compare Clear 5W	0: Disables MTU5.TCNTW to be cleared to 0000h at MTU5.TCNTW and MTU5.TGRW compare match or input capture 1: Enables MTU5.TCNTW to be cleared to 0000h at MTU5.TCNTW and MTU5.TGRW compare match or input capture	R/W
b1	CMPCLR5V	TCNT Compare Clear 5V	0: Disables MTU5.TCNTV to be cleared to 0000h at MTU5.TCNTV and MTU5.TGRV compare match or input capture 1: Enables MTU5.TCNTV to be cleared to 0000h at MTU5.TCNTV and MTU5.TGRV compare match or input capture	R/W
b2	CMPCLR5U	TCNT Compare Clear 5U	0: Disables MTU5.TCNTU to be cleared to 0000h at MTU5.TCNTU and MTU5.TGRU compare match or input capture 1: Enables MTU5.TCNTU to be cleared to 0000h at MTU5.TCNTU and MTU5.TGRU compare match or input capture	R/W
b7 to b3	—	Reserved	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).

19.2.8 Timer Interrupt Enable Register (TIER)

- MTU1.TIER, MTU2.TIER

Address(es): MTU1.TIER 0009 5384h, MTU2.TIER 0009 5404h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

- MTU0.TIER, MTU3.TIER

Address(es): MTU0.TIER 0009 5304h, MTU3.TIER 0009 5208h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

- MTU4.TIER

Address(es): MTU4.TIER 0009 5209h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE	TTGE2	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIEA	TGR Interrupt Enable A	0: Interrupt requests (TGIA) disabled 1: Interrupt requests (TGIA) enabled	R/W
b1	TGIEB	TGR Interrupt Enable B	0: Interrupt requests (TGIB) disabled 1: Interrupt requests (TGIB) enabled	R/W
b2	TGIEC	TGR Interrupt Enable C	0: Interrupt requests (TGIC) disabled 1: Interrupt requests (TGIC) enabled	R/W
b3	TGIED	TGR Interrupt Enable D	0: Interrupt requests (TGID) disabled 1: Interrupt requests (TGID) enabled	R/W
b4	TCIEV	Overflow Interrupt Enable	0: Interrupt requests (TCIV) disabled 1: Interrupt requests (TCIV) enabled	R/W
b5	TCIEU	Underflow Interrupt Enable	0: Interrupt requests (TCIU) disabled 1: Interrupt requests (TCIU) enabled	R/W
b6	TTGE2	A/D Conversion Start Request Enable 2	0: A/D conversion start request generation by MTUn.TCNT underflow (trough) disabled 1: A/D conversion start request generation by MTUn.TCNT underflow (trough) enabled	R/W
b7	TTGE	A/D Conversion Start Request Enable	0: A/D conversion start request generation disabled 1: A/D conversion start request generation enabled	R/W

n = 4

The TIER register enables or disables interrupt requests from each channel. The MTU has a total of seven TIER registers, two for MTU0 and one each for MTU1 to MTU5.

TGIEA and TGIEB Bits (TGR Interrupt Enable A and B)

Each bit enables or disables interrupt requests (TGIn) (n = A, B).

TGIEC and TGIED Bits (TGR Interrupt Enable C and D)

Each bit enables or disables an interrupt request (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, and MTU4, this bit is reserved. It is read as 0. The write value should be 0.

TTGE2 Bit (A/D Conversion Start Request Enable 2)

This bit enables or disables generation of A/D conversion start requests by MTUn.TCNT underflow (trough) in complementary PWM mode (n = 4).
In MTU0 to MTU3, this bit is reserved. It is read as 0. The write value should be 0.

TTGE Bit (A/D Conversion Start Request Enable)

This bit enables or disables generation of A/D conversion start requests by TGRA input capture/compare match.

- MTU0.TIER2

Address(es): MTU0.TIER2 0009 5324h

	b7	b6	b5	b4	b3	b2	b1	b0
	TTGE2	—	—	—	—	—	TGIEF	TGIEE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIEE	TGR Interrupt Enable E	0: Interrupt requests (TGIE) disabled 1: Interrupt requests (TGIE) enabled	R/W
b1	TGIEF	TGR Interrupt Enable F	0: Interrupt requests (TGIF) disabled 1: Interrupt requests (TGIF) enabled	R/W
b6 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TTGE2	A/D Conversion Start Request Enable 2	0: A/D conversion start request generation by compare match between MTU0.TCNT and MTU0.TGRE disabled 1: A/D conversion start request generation by compare match between MTU0.TCNT and MTU0.TGRE enabled	R/W

TGIEE and TGIEF Bits (TGR Interrupt Enable E and F)

Each bit enables or disables interrupt requests by compare match between MTU0.TCNT and MTU0.TGRn (n = E, F).

TTGE2 Bit (A/D Conversion Start Request Enable 2)

Each bit enables or disables A/D conversion start requests by compare match between MTU0.TCNT and MTU0.TGRE.

- MTU5.TIER

Address(es): MTU5.TIER 0009 54B2h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TGIE5U	TGIE5V	TGIE5W
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIE5W	TGR Interrupt Enable 5W	0: Interrupt requests TGIW5 disabled 1: Interrupt requests TGIW5 enabled	R/W
b1	TGIE5V	TGR Interrupt Enable 5V	0: Interrupt requests TGIV5 disabled 1: Interrupt requests TGIV5 enabled	R/W
b2	TGIE5U	TGR Interrupt Enable 5U	0: Interrupt requests TGIU5 disabled 1: Interrupt requests TGIU5 enabled	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TGIE5n Bits (TGR Interrupt Enable 5n)

Each bit enables or disables interrupt requests (TGIn5) (n = U, V, W).

19.2.9 Timer Status Register (TSR)

- MTU1.TSR, MTU2.TSR

Address(es): MTU1.TSR 0009 5385h, MTU2.TSR 0009 5405h

	b7	b6	b5	b4	b3	b2	b1	b0
	TCFD	—	—	—	—	—	—	—
Value after reset:	1	1	0	0	0	0	0	0

- MTU3.TSR, MTU4.TSR

Address(es): MTU3.TSR 0009 522Ch, MTU4.TSR 0009 522Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	TCFD	—	—	—	—	—	—	—
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	The read value is undefined.The write value should be 1.	R/W
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7	TCFD	Count Direction Flag	0: TCNT counts down 1: TCNT counts up	R

TSR indicates the states of each of the channels. The MTU has a total of four TSR registers, one each for MTU1 to MTU4.

TCFD Flag (Count Direction Flag)

Status flag that indicates the direction in which TCNT is counting in MTU1 to MTU4.

19.2.10 Timer Buffer Operation Transfer Mode Register (TBTM)

- MTU0.TBTM

Address(es): MTU0.TBTM 0009 5326h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TTSE	TTSB	TTSA
Value after reset:	0	0	0	0	0	0	0	0

- MTU3.TBTM, MTU4.TBTM

Address(es): MTU3.TBTM 0009 5238h, MTU4.TBTM 0009 5239h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	TTSB	TTSA
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	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
b1	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
b2	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, data is transferred from MTU0.TGRF to MTU0.TGRE	R/W
b7 to b3	—	Reserved	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 three TBTM registers, one each for MTU0, MTU3, and MTU4.

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 and MTU4, 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.

19.2.11 Timer Input Capture Control Register (TICCR)

Address(es): MTU1.TICCR 0009 5390h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	I2BE	I2AE	I1BE	I1AE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	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
b1	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
b2	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
b3	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
b7 to b4	—	Reserved	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.

19.2.12 Timer Counter (TCNT)

Address(es): MTU0.TCNT 0009 5306h, MTU1.TCNT 0009 5386h, MTU2.TCNT 0009 5406h, MTU3.TCNT 0009 5210h, MTU4.TCNT 0009 5212h, MTU5.TCNTU 0009 5480h, MTU5.TCNTV 0009 5490h, MTU5.TCNTW 0009 54A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: TCNT must not be accessed in 8 bits; it should be accessed in 16 bits.

TCNT is a 16-bit readable/writable counter. The MTU has a total of eight TCNT counters, one each for MTU0 to MTU4 and three (MTU5.TCNTU, TCNTV, and TCNTW) for MTU5. The TCNT counters in MTU0 to MTU4 are initialized to 0000h by a reset. MTU5.TCNTU, MTU5.TCNTV, and MTU5.TCNTW are initialized to 0000h by a reset.
In MTU0 to MTU4, the TCNT counters must not be accessed in 8-bit units; they should be accessed in 16-bit units.
The MTU1.TCNT and MTU2.TCNT counters are read as 0000h when TMDR3.LWA is 1. Refer to section 19.2.5, Timer Mode Register 3 (TMDR3) for details.

19.2.13 Timer Longword Counter (TCNTLW)

Address(es): MTU1.TCNTLW 0009 53A0h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	0	0	0	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
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: TCNTLW must not be accessed in 8 or 16 bits; it should be accessed in 32 bits.

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 is 1. The TCNTLW counter is initialized to 0000 0000h by a reset. This counter is read as 0000 0000h when TMDR3.LWA is 0. Refer to section 19.2.5, Timer Mode Register 3 (TMDR3) for details. This register can only be used in 32-bit phase counting mode.

19.2.14 Timer General Register (TGR)

Address(es): MTU0.TGRA 0009 5308h, MTU0.TGRB 0009 530Ah, MTU0.TGRC 0009 530Ch, MTU0.TGRD 0009 530Eh, MTU0.TGRE 0009 5320h, MTU0.TGRF 0009 5322h,
MTU1.TGRA 0009 5388h, MTU1.TGRB 0009 538Ah,
MTU2.TGRA 0009 5408h, MTU2.TGRB 0009 540Ah,
MTU3.TGRA 0009 5218h, MTU3.TGRB 0009 521Ah, MTU3.TGRC 0009 5224h, MTU3.TGRD 0009 5226h, MTU3.TGRE 0009 5272h,
MTU4.TGRA 0009 521Ch, MTU4.TGRB 0009 521Eh, MTU4.TGRC 0009 5228h, MTU4.TGRD 0009 522Ah, MTU4.TGRE 0009 5274h, MTU4.TGRF 0009 5276h,
MTU5.TGRU 0009 5482h, MTU5.TGRV 0009 5492h, MTU5.TGRW 0009 54A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: TGR must not be accessed in 8 bits; it should be accessed in 16 bits. The initial value of TGR is FFFFh.

The TGR register is 16-bit readable/writable register. The MTU has a total of 24 TGR registers, six for MTU0, two each for MTU1 and MTU2, five for MTU3, six for MTU4, and three for MTU5.
The TGRA, TGRB, TGRC, and TGRD registers function as either output compare or input capture registers. The TGRC and TGRD registers for MTU0, MTU3, and MTU4 can also be designated for operation as buffer registers. TGR buffer register combinations are TGRA and TGRC, and TGRB and TGRD.
MTU0.TGRE and MTU0.TGRF function as compare registers. When the MTU0.TCNT count matches the MTU0.TGRE value, an A/D conversion 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. MTU5.TGRU, MTU5.TGRV, and MTU5.TGRW function as compare match, input capture, or external pulse width measurement registers.
The MTU1.TGRA, MTU2.TGRA, MTU1.TGRB, and MTU2.TGRB registers are read as 0000h when TMDR3.LWA is 1. Refer to section 19.2.5, Timer Mode Register 3 (TMDR3) for details.

19.2.15 Timer Longword General Registers (TGRALW, TGRBLW)

Address(es): MTU1.TGRALW 0009 53A4h, MTU1.TGRBLW 0009 53A8h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: TGRALW and TGRBLW must not be accessed in 8 or 16 bits; it should be accessed in 32 bits.

The TGRnLW register (n = A, B) 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 is 1. The TGRnLW register is initialized to FFFF FFFFh by a reset, but it is read as 0000 0000h when TMDR3.LWA is 0. Refer to section 19.2.5, Timer Mode Register 3 (TMDR3) for details.
The TGRALW and TGRBLW registers function as input capture registers which can only be used in 32-bit phase counting mode.

19.2.16 Timer Start Registers (TSTRA, TSTR)

- MTU.TSTRA (for MTU0, MTU1, MTU2, MTU3, and MTU4)

Address(es): MTU.TSTRA 0009 5280h

	b7	b6	b5	b4	b3	b2	b1	b0
	CST4	CST3	—	—	—	CST2	CST1	CST0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CST0	Counter Start 0	0: MTU0.TCNT counting is stopped 1: MTU0.TCNT performs count operation	R/W
b1	CST1	Counter Start 1	0: MTU1.TCNT counting is stopped 1: MTU1.TCNT performs count operation	R/W
b2	CST2	Counter Start 2	0: MTU2.TCNT counting is stopped 1: MTU2.TCNT performs count operation	R/W
b5 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CST3	Counter Start 3	0: MTU3.TCNT counting is stopped 1: MTU3.TCNT performs count operation	R/W
b7	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.

The TSTRA register starts or stops TCNT operation in MTU0 to MTU4.
TSTR starts or stops TCNT operation in MTU5.
Before setting the operating mode in TMDR1 or setting the TCNT count clock in TCR, be sure to stop the TCNT counter.

CSTn Bits (Counter Start n) (n = 0, 1, 2, 3, 4)

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, initial output level specified in the TOCR1A or TOCR2A register is output from the MTIOC pin in complementary PWM mode or reset-synchronized PWM mode. In any mode other than complementary PWM mode and reset synchronous PWM mode, the output compare signal level from the MTIOC pin is retained. If TIOR is written to while the CSTn bit is 0, the pin output level will be changed to the specified initial output value.

- MTU5.TSTR

Address(es): MTU5.TSTR 0009 54B4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	CSTU5	CSTV5	CSTW5
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CSTW5	Counter Start W5	0: MTU5.TCNTW counting is stopped 1: MTU5.TCNTW performs count operation	R/W
b1	CSTV5	Counter Start V5	0: MTU5.TCNTV counting is stopped 1: MTU5.TCNTV performs count operation	R/W
b2	CSTU5	Counter Start U5	0: MTU5.TCNTU counting is stopped 1: MTU5.TCNTU performs count operation	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

19.2.17 Timer Synchronous Register (TSYRA)

Address(es): MTU.TSYRA 0009 5281h

	b7	b6	b5	b4	b3	b2	b1	b0
	SYNC4	SYNC3	—	—	—	SYNC2	SYNC1	SYNC0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SYNC0	Timer Synchronous Operation 0	0: MTU0.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU0.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b1	SYNC1	Timer Synchronous Operation 1	0: MTU1.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU1.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b2	SYNC2	Timer Synchronous Operation 2	0: MTU2.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU2.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b5 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	SYNC3	Timer Synchronous Operation 3	0: MTU3.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU3.TCNT performs synchronous operation. (TCNT synchronous setting/synchronous clearing is enabled.)	R/W
b7	SYNC4	Timer Synchronous Operation 4	0: MTU4.TCNT operates independently (TCNT setting/clearing is not related to other channels). 1: MTU4.TCNT performs synchronous operation. (TCNT synchronous setting/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 TSYR is set to 1.

SYNCn Bits (Timer Synchronous Operation n) (n = 0, 1, 2, 3, 4)

Each bit selects whether operation is independent of or synchronized with other channels.
When synchronous operation is selected, the TCNT synchronous setting 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.

19.2.18 Timer Counter Synchronous Start Register (TCSYSTR)

Address(es): MTU.TCSYSTR 0009 5282h

	b7	b6	b5	b4	b3	b2	b1	b0
	SCH0	SCH1	SCH2	SCH3	SCH4	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R
b3	SCH4	Synchronous Start 4	0: Does not specify synchronous start for MTU4.TCNT 1: Specifies synchronous start for MTU4.TCNT	R/(W)*1
b4	SCH3	Synchronous Start 3	0: Does not specify synchronous start for MTU3.TCNT 1: Specifies synchronous start for MTU3.TCNT	R/(W)*1
b5	SCH2	Synchronous Start 2	0: Does not specify synchronous start for MTU2.TCNT 1: Specifies synchronous start for MTU2.TCNT	R/(W)*1
b6	SCH1	Synchronous Start 1	0: Does not specify synchronous start for MTU1.TCNT 1: Specifies synchronous start for MTU1.TCNT	R/(W)*1
b7	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. This bit is automatically cleared when the corresponding counter starts.

TCSYSTR specifies synchronous start of the counters.

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

19.2.19 Timer Read/Write Enable Register (TRWERA)

Address(es): MTU.TRWERA 0009 5284h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	RWE
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	RWE	Read/Write Enable	0: Read/write access to the registers is disabled 1: Read/write access to the registers is enabled	R/W
b7 to b1	—	Reserved	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.

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

- 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)

19.2.20 Timer Output Master Enable Register (TOERA)

Address(es): MTU.TOERA 0009 520Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	OE4D	OE4C	OE3D	OE4B	OE4A	OE3B
Value after reset:	1	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OE3B	Master Enable MTIOC3B	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b1	OE4A	Master Enable MTIOC4A	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b2	OE4B	Master Enable MTIOC4B	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b3	OE3D	Master Enable MTIOC3D	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b4	OE4C	Master Enable MTIOC4C	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b5	OE4D	Master Enable MTIOC4D	0: MTU output is disabled*1 1: MTU output is enabled	R/W
b7, b6	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Note 1. To output the inactive level from each pin when the MTU output is set to disabled, first set the data direction register (PDR) and port output data register (PODR) of I/O ports to output the inactive level from general I/O ports, and then set the port mode register (PMR) to use general I/O ports. For details, refer to section 17, I/O Ports.

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 (refer to Figure 19.42 and Figure 19.45).

19.2.21 Timer Output Control Register 1 (TOCR1A)

Address(es): MTU.TOCR1A 0009 520Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	PSYE	—	—	TOCL	TOCS	OLSN	OLSP
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLSP	Output Level Select P*1, *3	Refer to Table 19.30.	R/W
b1	OLSN	Output Level Select N*1, *3	Refer to Table 19.31.	R/W
b2	TOCS	TOC Select	0: TOCR1j setting is selected (j = A) 1: TOCR2j setting is selected	R/W
b3	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
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	PSYE	PWM Synchronous Output Enable	0: Toggle output is disabled 1: Toggle output is enabled	R/W
b7	—	Reserved	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 enables or disables 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) 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).

PSYE Bit (PWM Synchronous Output Enable)

This bit enables or disables toggle output synchronized with the PWM period.

Table 19.30 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 19.31 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 19.2 shows an example of output in complementary PWM mode (one phase) when OLSN = 1 and OLSP = 1.

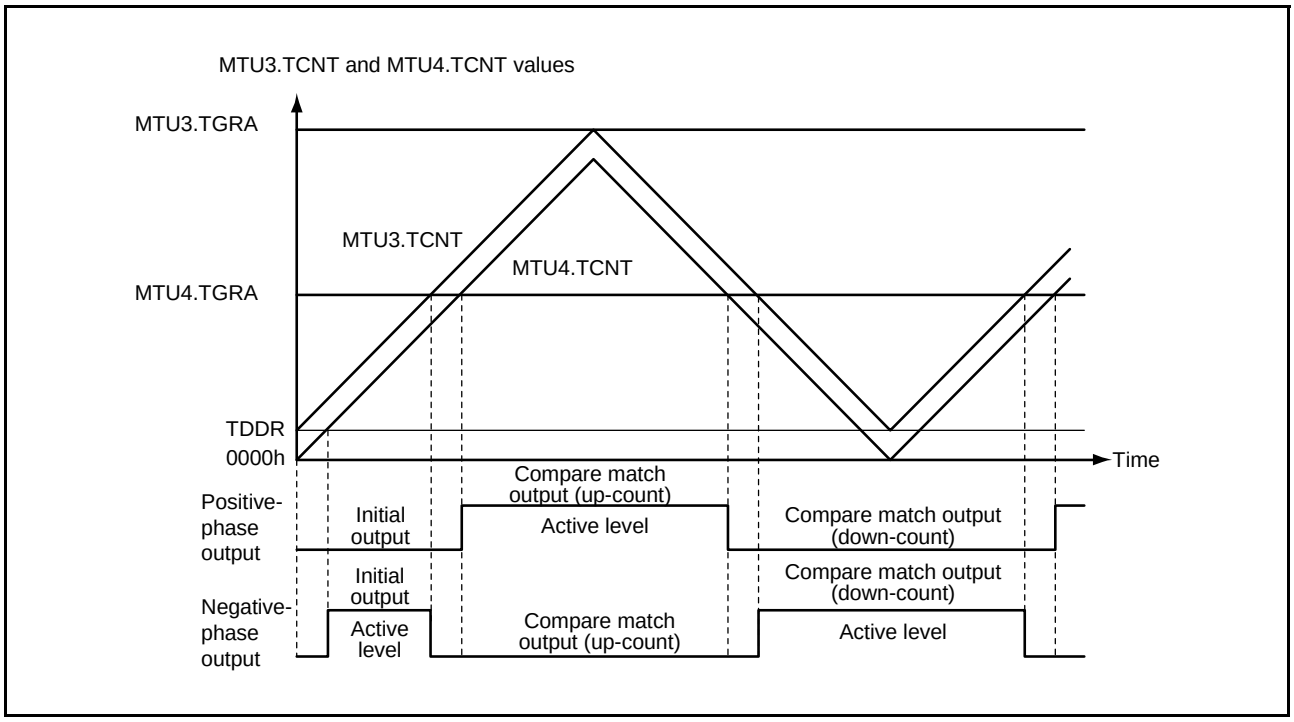


Figure 19.2 Example of Output in Complementary PWM Mode

19.2.22 Timer Output Control Register 2 (TOCR2A)

Address(es): MTU.TOCR2A 0009 520Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	BF[1:0]		OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLS1P	Output Level Select 1P*1, *2	This bit selects the output level on MTIOC3B in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 19.32.	R/W
b1	OLS1N	Output Level Select 1N*1, *2	This bit selects the output level on MTIOC3D in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 19.33.	R/W
b2	OLS2P	Output Level Select 2P*1, *2	This bit selects the output level on MTIOC4A in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 19.34.	R/W
b3	OLS2N	Output Level Select 2N*1, *2	This bit selects the output level on MTIOC4C in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 19.35.	R/W
b4	OLS3P	Output Level Select 3P*1, *2	This bit selects the output level on MTIOC4B in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 19.36.	R/W
b5	OLS3N	Output Level Select 3N*1, *2	This bit selects the output level on MTIOC4D in reset-synchronized PWM mode and complementary PWM mode. Refer to Table 19.37.	R/W
b7, b6	BF[1:0]	TOLBR Buffer Transfer Timing Select	These bits select the timing for transferring data from TOLBRj to TOCR2j. Refer to Table 19.38 for details.	R/W

j = A

Note 1. Setting the TOCR1j.TOCS bit to 1 makes this bit setting valid.

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 controls 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 19.32 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

m = 3

Table 19.33 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

m = 3

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 19.34 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

m = 4

Table 19.35 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

m = 4

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 19.36 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

m = 4

Table 19.37 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

m = 4

Note: The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

Table 19.38 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

n = 4; m = 3; j = A

19.2.23 Timer Output Level Buffer Register (TOLBRA)

Address(es): MTU.TOLBRA 0009 5236h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	OLS3N	OLS3P	OLS2N	OLS2P	OLS1N	OLS1P
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	OLS1P	Output Level Select 1P	Specify the buffer value to be transferred to the OLS1P bit in TOCR2j.	R/W
b1	OLS1N	Output Level Select 1N	Specify the buffer value to be transferred to the OLS1N bit in TOCR2j.	R/W
b2	OLS2P	Output Level Select 2P	Specify the buffer value to be transferred to the OLS2P bit in TOCR2j.	R/W
b3	OLS2N	Output Level Select 2N	Specify the buffer value to be transferred to the OLS2N bit in TOCR2j.	R/W
b4	OLS3P	Output Level Select 3P	Specify the buffer value to be transferred to the OLS3P bit in TOCR2j.	R/W
b5	OLS3N	Output Level Select 3N	Specify the buffer value to be transferred to the OLS3N bit in TOCR2j.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

j = A

TOLBRA is buffer register for TOCR2A and specify the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Figure 19.3 shows an example of the PWM output level setting procedure in buffer operation.

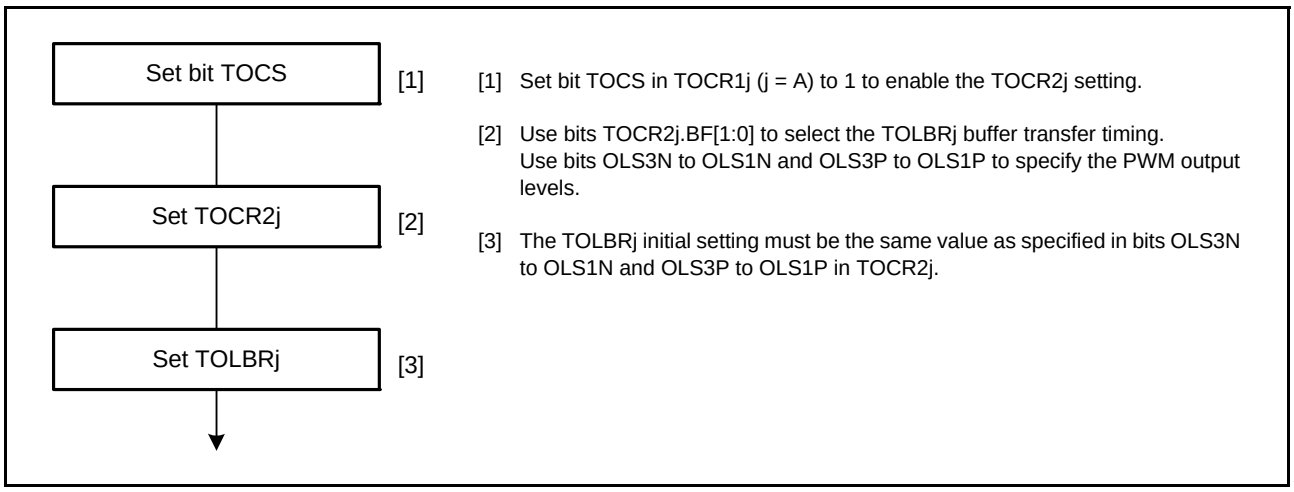


Figure 19.3 Example of PWM Output Level Setting Procedure in Buffer Operation

19.2.24 Timer Gate Control Register A (TGCRA)

Address(es): MTU.TGCRA 0009 520Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	BDC	N	P	FB	WF	VF	UF
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	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 the FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. Refer to Table 19.39.	R/W
b1	VF			R/W
b2	WF			R/W
b3	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 (TGCRA's UF, VF, and WF settings)	R/W
b4	P	Positive-Phase Output (P) Control	0: Level output 1: Reset-synchronized PWM or complementary PWM output	R/W
b5	N	Negative-Phase Output (N) Control	0: Level output 1: Reset-synchronized PWM or complementary PWM output	R/W
b6	BDC	Brushless DC Motor	0: Ordinary output 1: Functions of this register are made effective	R/W
b7	—	Reserved	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 the FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. Refer to Table 19.39 for details.

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.

Table 19.39 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

19.2.25 Timer Subcounter (TCNTSA)

Address(es): MTU.TCNTSA 0009 5220h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Note: TCNTSA must not be accessed in 8 bits; it should be accessed in 16 bits.

TCNTSA is a 16-bit read-only counter used only in complementary PWM mode.
The initial value of TCNTSA after a reset is 0000h.

19.2.26 Timer Period Data Register (TCDRA)

Address(es): MTU.TCDRA 0009 5214h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: TCDRA must not be accessed in 8 bits; it should be accessed in 16 bits.

TCDRA is a 16-bit readable/writable register used only in complementary PWM mode. Set half the PWM carrier period as the TCDRA value. The TCDRA register is constantly compared with the TCNTSA counter in complementary PWM mode, respectively. When a match occurs, the TCNTSA counter switches the count direction (down-count to up-count).
The initial value of TCDRA after a reset is FFFFh.

19.2.27 Timer Period Buffer Register (TCBRA)

Address(es): MTU.TCBRA 0009 5222h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: TCBRA must not be accessed in 8 bits; it should be accessed in 16 bits.

TCBRA is a 16-bit readable/writable register, used only in complementary PWM mode, that function as buffer register for TCDRA. The TCBRA value is transferred to TCDRA with the transfer timing set in TMDR1. The initial value of TCBRA after a reset is FFFFh.

19.2.28 Timer Dead Time Data Register (TDDRA)

Address(es): MTU.TDDRA 0009 5216h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: TDDRA must not be accessed in 8 bits; it should be accessed in 16 bits.

TDDRA is a 16-bit readable/writable register, used only in complementary PWM mode, that specify the MTU3.TCNT and MTU4.TCNT counter offset value. In complementary PWM mode, when the MTU3.TCNT and MTU4.TCNT counters are cleared and then restarted, the TDDRA value is loaded into the MTU3.TCNT counter and the count operation starts. The initial value of TDDRA after a reset is FFFFh.

19.2.29 Timer Dead Time Enable Register (TDERA)

Address(es): MTU.TDERA 0009 5234h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TDER
Value after reset:	0	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	TDER	Dead Time Enable	0: No dead time is generated 1: Dead time is generated*1	R/(W)
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. TDDRA must be set to 1 or a larger value.

TDERA controls dead time generation in complementary PWM mode. The MTU has one TDER for MTU3. TDERA 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

19.2.30 Timer Buffer Transfer Set Register (TBTERA)

Address(es): MTU.TBTERA 0009 5232h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	BTE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	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, refer to Table 19.40.	R/W
b7 to b2	—	Reserved	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

TBTERA enables or disables 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 19.40 Setting of TBTERA.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, refer to section 19.3.8, Complementary PWM Mode.

Note 2. When interrupt skipping is disabled the T3AEN and T4VEN bits are set to 0 in the timer interrupt skipping set register (TITCR1A) or the skipping count set bits (T3ACOR and T4VCOR) in TITCR1A 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) to 0). If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed.

19.2.31 Timer Waveform Control Register (TWCRA)

Address(es): MTU.TWCRA 0009 5260h

	b7	b6	b5	b4	b3	b2	b1	b0
	CCE	—	—	—	—	—	—	WRE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WRE	Waveform Retain Enable	0: Initial values specified in TOCR1A and TOCR2A are output 1: Initial output is inhibited	R/(W)
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	CCE	Compare Match Clear Enable *1	0: Counters are not cleared at MTU3.TGRA compare match 1: Counters are cleared at MTU3.TGRA compare match	R/(W)

Note 1. Do not set to 1 when complementary PWM mode 1 is not selected.

TWCRA controls the output waveform when synchronous counter clearing occurs in MTU3.TCNT and MTU4.TCNT in complementary PWM mode and specifies whether to clear the counters at MTU3.TGRA compare match.
The CCE bit and WRE bit in TWCRA 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, refer to Figure 19.47.
[Setting condition]

- When 1 is written to the WRE bit after reading WRE = 0

CCE Bit (Compare Match Clear Enable)

This bit specifies whether to clear counters at MTU3.TGRA compare match in complementary PWM mode.
[Setting condition]

- When 1 is written to CCE after reading CCE = 0

19.2.32 Noise Filter Control Register n (NFCRn) (n = 0 to 4, C)

- MTU0.NFCR0, MTU1.NFCR1, MTU2.NFCR2, MTU3.NFCR3, MTU4.NFCR4

Address(es): MTU0.NFCR0 0009 5290h, MTU1.NFCR1 0009 5291h, MTU2.NFCR2 0009 5292h, MTU3.NFCR3 0009 5293h, MTU4.NFCR4 0009 5294h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	NFCS[1:0]		NFDEN	NFCEN	NFBEN	NFAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	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
b1	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
b2	NFCEN	Noise Filter C Enable*1	0: The noise filter for the MTIOCnC pin is disabled. 1: The noise filter for the MTIOCnC pin is enabled.	R/W
b3	NFDEN	Noise Filter D Enable*1	0: The noise filter for the MTIOCnD pin is disabled. 1: The noise filter for the MTIOCnD pin is enabled.	R/W
b5, b4	NFCS[1:0]	Noise Filter Clock Select	b5 b4 0 0: PCLKB/1 0 1: PCLKB/8 1 0: PCLKB/32 1 1: Clock source for counting	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. These bits are reserved in MTU1 and MTU2. These bits are read as 0 and writing to them has no effect.

The NFCRn register (n = 0 to 4) sets the noise filter function of input capture pins for the corresponding channel.

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 TMDR.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 TMDR.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 TMDR.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 TMDR.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.

- MTU0.NFCRC

Address(es): MTU0.NFCRC 0009 5299h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	NFCS[1:0]		NFDEN	NFCEN	NFBEN	NFAEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	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
b1	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
b2	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
b3	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
b5, b4	NFCS[1:0]	Noise Filter Clock Select	b5 b4 0 0: PCLKB/1 0 1: PCLKB/2 1 0: PCLKB/8 1 1: PCLKB/32	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The NFCRC register sets the noise filter function of external clock pins common to each channel.

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 input capture function.

19.2.33 Noise Filter Control Register 5 (NFCR5)

Address(es): MTU5.NFCR5 0009 5295h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	NFCS[1:0]		—	NFWEN	NFVEN	NFUEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	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
b1	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
b2	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
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	NFCS[1:0]	Noise Filter Clock Select	b5 b4 0 0: PCLKB/1 0 1: PCLKB/8 1 0: PCLKB/32 1 1: Clock source for counting	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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.

19.2.34 Timer A/D Conversion Start Request Control Register (TADCR)

Address(es): MTU4.TADCR 0009 5240h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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	Bit Name	Description	R/W
b0	ITB4VE	TCIV4 Interrupt Skipping Link Enable*1, *2, *3	0: A/D conversion start request TRG4BN and TCIV4 interrupt skipping 1 are not linked 1: A/D conversion start request TRG4BN and TCIV4 interrupt skipping 1 are linked	R/W
b1	ITB3AE	TGIA3 Interrupt Skipping Link Enable*1, *2, *3	0: A/D conversion start request TRG4BN and TGIA3 interrupt skipping 1 are not linked 1: A/D conversion start request TRG4BN and TGIA3 interrupt skipping 1 are linked	R/W
b2	ITA4VE	TCIV4 Interrupt Skipping Link Enable*1, *2, *3	0: A/D conversion start request TRG4AN and TCIV4 interrupt skipping 1 are not linked 1: A/D conversion start request TRG4AN and TCIV4 interrupt skipping 1 are linked	R/W
b3	ITA3AE	TGIA3 Interrupt Skipping Link Enable*1, *2, *3	0: A/D conversion start request TRG4AN and TGIA3 interrupt skipping 1 are not linked 1: A/D conversion start request TRG4AN and TGIA3 interrupt skipping 1 are linked	R/W
b4	DT4BE	Down-Count TRG4BN Enable*3	0: A/D conversion start requests (TRG4BN) disabled during MTU4.TCNT down-count operation 1: A/D conversion start requests (TRG4BN) enabled during MTU4.TCNT down-count operation	R/W
b5	UT4BE	Up-Count TRG4BN Enable	0: A/D conversion start requests (TRG4BN) disabled during MTU4.TCNT up-count operation 1: A/D conversion start requests (TRG4BN) enabled during MTU4.TCNT up-count operation	R/W
b6	DT4AE	Down-Count TRG4AN Enable*3	0: A/D conversion start requests (TRG4AN) disabled during MTU4.TCNT down-count operation 1: A/D conversion start requests (TRG4AN) enabled during MTU4.TCNT down-count operation	R/W
b7	UT4AE	Up-Count TRG4AN Enable	0: A/D conversion start requests (TRG4AN) disabled during MTU4.TCNT up-count operation 1: A/D conversion start requests (TRG4AN) enabled during MTU4.TCNT up-count operation	R/W
b13 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	BF[1:0]	MTU4.TADCOBRA/TADCOBRB Transfer Timing Select	Refer to Table 19.41 for details. These bits specify the transfer timing from MTU4.TADCOBRA and MTU4.TADCOBRB to MTU4.TADCORA and MTU4.TADCORB.	R/W

Note: MTU4.TADCR must not be accessed in 8 bits; it should be accessed in 16 bits.
Note 1. Set to 0 when interrupt skipping is disabled (the T3AEN and T4VEN bits in TITCR1A are set to 0 or the T3ACOR and T4VCOR bits in TITCR1A are set to 0).
Note 2. If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D conversion start requests will not be issued.
Note 3. Set to 0 when complementary PWM mode is not selected.

TADCR enables or disables A/D conversion start requests and specifies whether to link A/D conversion start requests with interrupt skipping function.

Table 19.41 Setting of Transfer Timing by TADCR.BF[1:0] Bits (MTU4)

Bit 15	Bit 14	Description			
BF[1]	BF[0]	In Complementary PWM Mode	In Reset-Synchronized PWM Mode	In PWM Mode 1	In Normal Mode
0	0	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).	Data is not transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB).	Data is not transferred 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) at the crest of the MTU4.TCNT.	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) when a compare match occurs between MTU3.TCNT and MTU3.TGRA.	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) when a compare match occurs between MTU4.TCNT and MTU4.TGRA.	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) when a compare match occurs between MTU4.TCNT and MTU4.TGRA.
1	0	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) at the trough of the MTU4.TCNT.	Setting prohibited	Setting prohibited	Setting prohibited
1	1	Data is transferred from the cycle set buffer register (MTU4.TADCOBRA, MTU4.TADCOBRB) to the cycle set register (MTU4.TADCORA, MTU4.TADCORB) at the crest and trough of the MTU4.TCNT.	Setting prohibited	Setting prohibited	Setting prohibited

19.2.35 Timer A/D Conversion Start Request Cycle Set Registers (TADCORA, TADCORB)

Address(es): MTU4.TADCORA 0009 5244h, MTU4.TADCORB 0009 5246h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: TADCORA and TADCORB must not be accessed in 8 bits; it should be accessed in 16 bits.

Note 1. When the A/D conversion start request delaying function linked with skipping function 1 (for details, refer to section 19.3.9 (5), A/D Conversion Start Request Delaying Function Linked with Interrupt Skipping Function 1) is used, the value of this register should be 0002h to TCDRA setting – 2 in MTU4.

Note 2. 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 conversion 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 PCLKB 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 PCLKB 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 PCLKB cycles (the TADCORB update value should be the previous value + 2 or greater, or previous value – 2 or smaller)

TADCORA and TADCORB are 16-bit readable/writable registers that issue a corresponding A/D conversion start request when the MTUn.TCNT (n = 4) count reaches the value in TADCORA or TADCORB.
MTUn.TADCORA and TADCORB are initialized to FFFFh by a reset.

19.2.36 Timer A/D Conversion Start Request Cycle Set Buffer Registers (TADCOBRA, TADCOBRB)

Address(es): MTU4.TADCOBRA 0009 5248h, MTU4.TADCOBRB 0009 524Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Note: TADCOBRA and TADCOBRB must not be accessed in 8 bits; it should be accessed in 16 bits.

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.
TADCOBRA and TADCOBRB are initialized to FFFFh by a reset.

19.2.37 Timer Interrupt Skipping Mode Register (TITMRA)

Address(es): MTU.TITMRA 0009 523Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TITM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TITM	Interrupt Skipping Function Select	Selects one of the two types of interrupt skipping functions. 0: Selects interrupt skipping function 1*1 1: Selects interrupt skipping function 2*2	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Setting the TITCR1A register enables interrupt skipping function 1.
Note 2. Setting the TITCR2A register enables interrupt skipping function 2.

TITMRA is used to select either of two skipping functions for the TITMRA register.

19.2.38 Timer Interrupt Skipping Set Register 1 (TITCR1A)

Address(es): MTU.TITCR1A 0009 5230h

	b7	b6	b5	b4	b3	b2	b1	b0
	T3AEN	T3ACOR[2:0]			T4VEN	T4VCOR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T4VCOR[2:0]	TCIV4 Interrupt Skipping Count Setting	These bits specify the TCIV4 interrupt skipping count within the range from 0 to 7.*1 For details, refer to Table 19.42.	R/W
b3	T4VEN	T4VEN	0: TCIV4 interrupt skipping disabled 1: TCIV4 interrupt skipping enabled	R/W
b6 to b4	T3ACOR[2:0]	TGIA3 Interrupt Skipping Count Setting	These bits specify the TGIA3 interrupt skipping count within the range from 0 to 7.*1 For details, refer to Table 19.43.	R/W
b7	T3AEN	T3AEN	0: TGIA3 interrupt skipping disabled 1: TGIA3 interrupt skipping 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 specify 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.

Table 19.42 Setting of Interrupt Skipping Count by T4VCOR[2:0] Bits

Bit 2	Bit 1	Bit 0	
T4VCOR[2]	T4VCOR[1]	T4VCOR[0]	Description
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.

Table 19.43 Setting of Interrupt Skipping Count by T3ACOR[2:0] Bits

Bit 6	Bit 5	Bit 4	
T3ACOR[2]	T3ACOR[1]	T3ACOR[0]	Description
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.

19.2.39 Timer Interrupt Skipping Counter 1 (TITCNT1A)

Address(es): MTU.TITCNT1A 0009 5231h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	T3ACNT[2:0]			—	T4VCNT[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	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
b3	—	Reserved	This bit is read as 0.	R
b6 to b4	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
b7	—	Reserved	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

19.2.40 Timer Interrupt Skipping Set Register 2 (TITCR2A)

Address(es): MTU.TITCR2A 0009 523Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	TRG4COR[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	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. For details, refer to Table 19.44.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

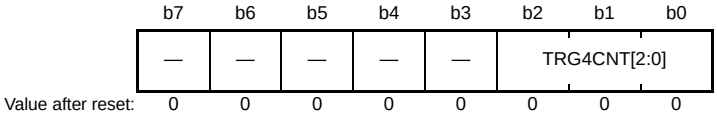
TITCR2A specifies the interrupt skipping count for TRG4AN and TRG4BN. This setting is valid only while TITMRA is set to 1.

Table 19.44 Setting of Interrupt Skipping Count by TRG4COR[2:0] Bits

Bit 2	Bit 1	Bit 0	
TRG4COR[2]	TRG4COR[1]	TRG4COR[0]	Description
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.

19.2.41 Timer Interrupt Skipping Counter 2 (TITCNT2A)

Address(es): MTU.TITCNT2A 0009 523Ch



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	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
b7 to b3	—	Reserved	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 (TITCNT2A) is generated 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

19.2.42 A/D Conversion Start Request Select Register 0 (TADSTRGR0)

Address(es): MTU.TADSTRGR0 0009 5D30h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	TADSTRS0[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	TADSTRS0[4:0]	A/D Conversion Start Request Select for ADSM0 Pin Output Frame Synchronization Signal Generation	These bits select the A/D conversion start request for generating the frame synchronization signal to be output from the ADSM0 pin. Refer to Table 19.45 for the relationship between the A/D conversion start request and settings. Settings other than those listed in Table 19.45 are prohibited.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The TADSTRGR0 register selects the A/D conversion start request for generating the A/D conversion start request frame synchronization signal to be output from the ADSM0 pin.

Table 19.45 Settings of A/D Conversion Start Request for Generating Frame Synchronization Signal

TADSTRS0[4:0]					Source	Descriptions
[4]	[3]	[2]	[1]	[0]		
0	0	0	0	0	—	Source not selected
0	0	0	0	1	TRGA0N	Compare match/input capture in MTU0.TGRA
0	0	0	1	0	TRGA1N	Compare match/input capture in MTU1.TGRA
0	0	0	1	1	TRGA2N	Compare match/input capture in MTU2.TGRA
0	0	1	0	0	TRGA3N	Compare match/input capture in MTU3.TGRA
0	0	1	0	1	TRGA4N	Compare match/input capture in MTU4.TGRA or MTU4.TCNT underflow (trough) in complementary PWM mode
0	1	0	0	0	TRG0N	Compare match in MTU0.TGRE
0	1	0	0	1	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT
0	1	0	1	0	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT
0	1	1	0	0	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT or compare match between MTU4.TADCORB and MTU4.TCNT (Interrupt skipping function 2 used)

19.3 Operation

19.3.1 Basic Functions

Each channel has TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting.
Each TGR register can be used as an input capture register or an output compare register.

(1) Counter Operation

When one of bits CST0 to CST4 in the TSTRA 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.

(a) Example of Count Operation Setting Procedure

Figure 19.4 shows an example of the count operation setting procedure.

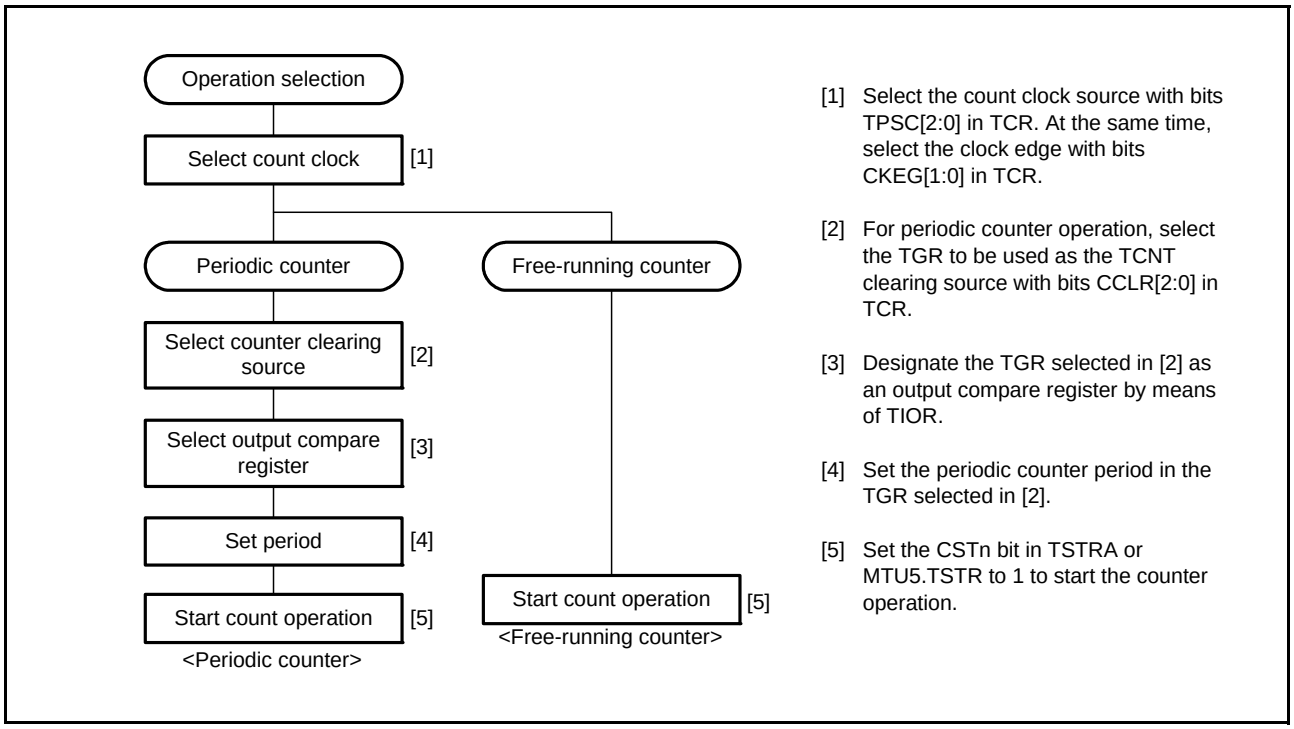


Figure 19.4 Example of Count Operation Setting Procedure

(b) Free-Running Count Operation and Periodic Count Operation

Immediately after a reset, the TCNT counters are all designated as free-running counters. When the CSTn bit in TSTRA or MTU5.TSTR is set to 1, the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from FFFFh to 0000h), an interrupt request is issued to the CPU if the corresponding TIER.TCIEV bit is 1. After an overflow, TCNT starts counting up again from 0000h.

Figure 19.5 illustrates free-running counter operation.

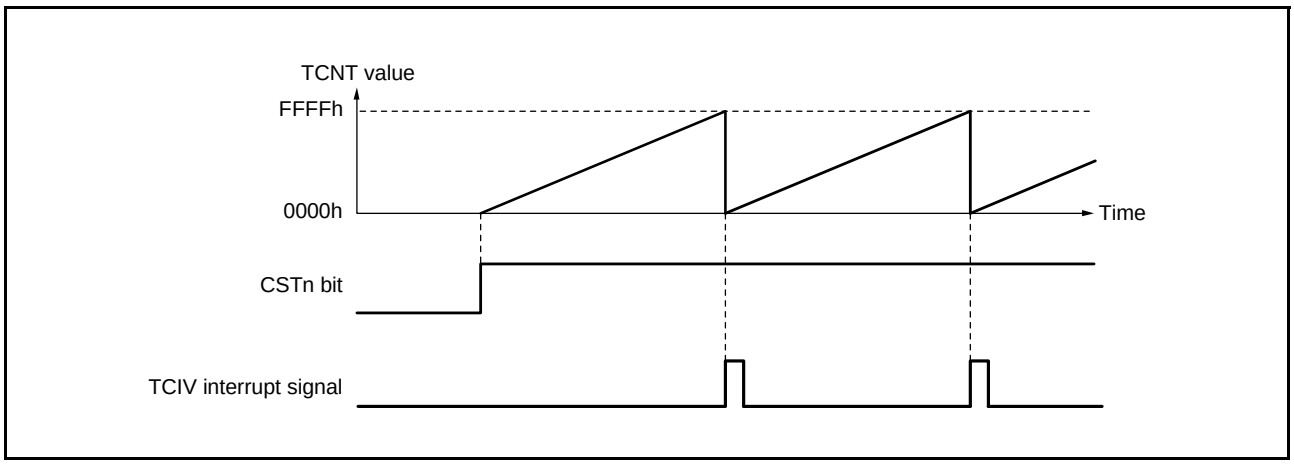


Figure 19.5 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, TCNT for the relevant channel performs periodic count operation. TGR for setting the period 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 CSTn bit in TSTRA or MTU5.TSTR is set to 1. When the count matches the value in TGR, TCNT becomes 0000h.

If the value of the corresponding TIER.TGIE bit is 1 at this point, an interrupt request is issued to the CPU. After a compare match, TCNT starts counting up again from 0000h.

Figure 19.6 illustrates periodic counter operation.

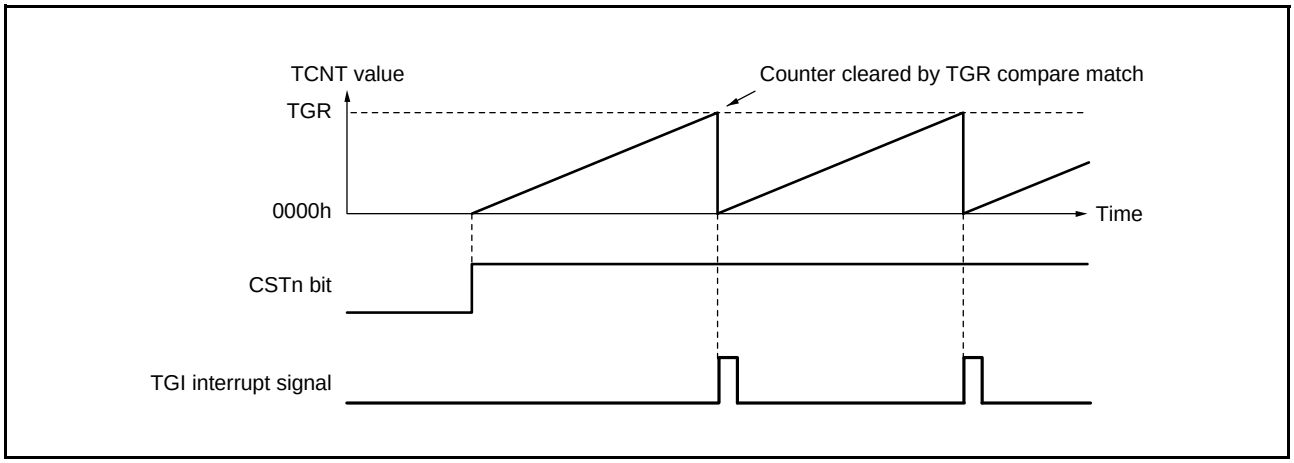


Figure 19.6 Periodic Counter Operation

(2) Waveform Output by Compare Match

Upon compare match, low, high, or toggle output from the corresponding pin can be performed.

(a) Example of Procedure for Setting Waveform Output by Compare Match

Figure 19.7 shows an example of the procedure for setting waveform output by compare match

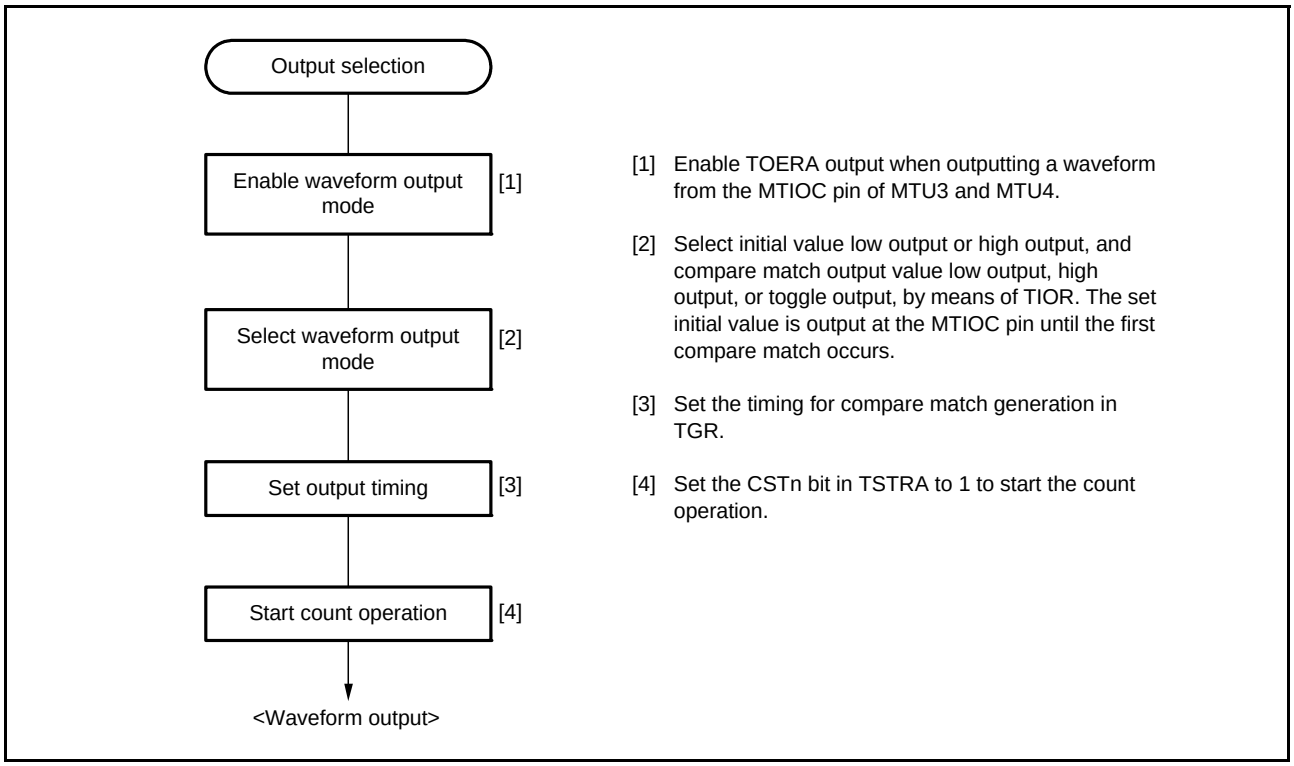


Figure 19.7 Example of Procedure for Setting Waveform Output by Compare Match

(b) Examples of Waveform Output Operation

Figure 19.8 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.

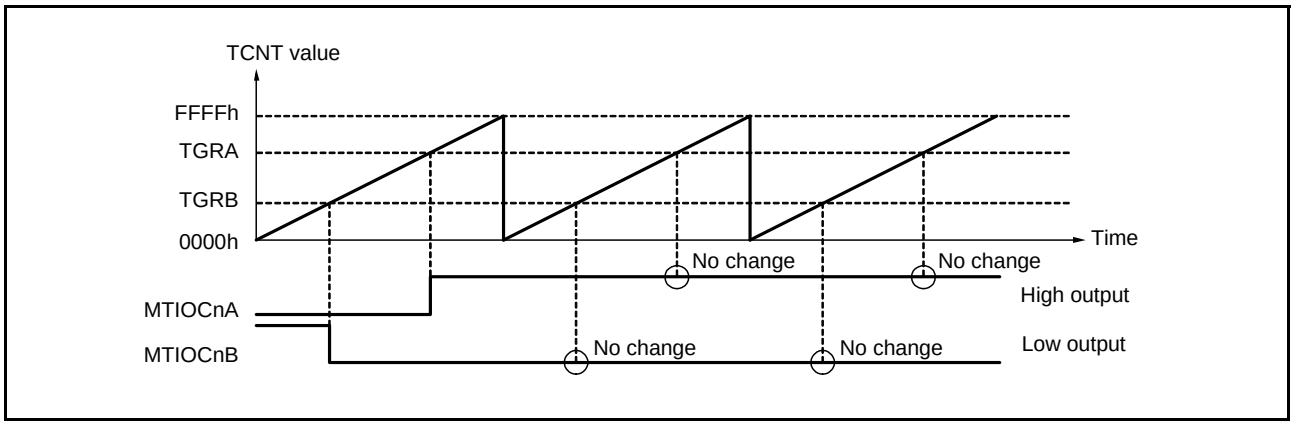


Figure 19.8 Example of low output and high output Operation (n = 0 to 4)

Figure 19.9 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.

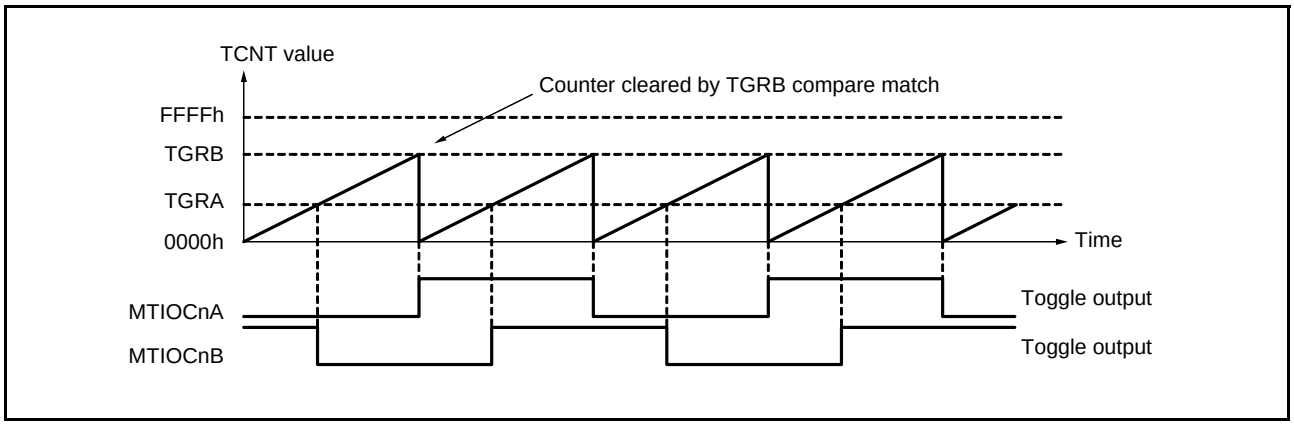


Figure 19.9 Example of Toggle Output Operation (n = 0 to 4)

(3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the MTIOCnm pin (n = 0 to 4; m = A to D) or MTIC5m pin (m = U, V, W) input edge.
The rising edge, falling edge, or both edges can be selected as the detection edge. For MTU0 and MTU1, another channel's count clock or compare match signal can also be specified as the input capture source.

Note: When another channel's count clock is used as the input capture input for MTU0 and MTU1, PCLKB/1 should not be selected as the count clock used for input capture input. Input capture will not be generated if PCLKB/1 is selected.

(a) Example of Input Capture Operation Setting Procedure

Figure 19.10 shows an example of the input capture operation setting procedure.

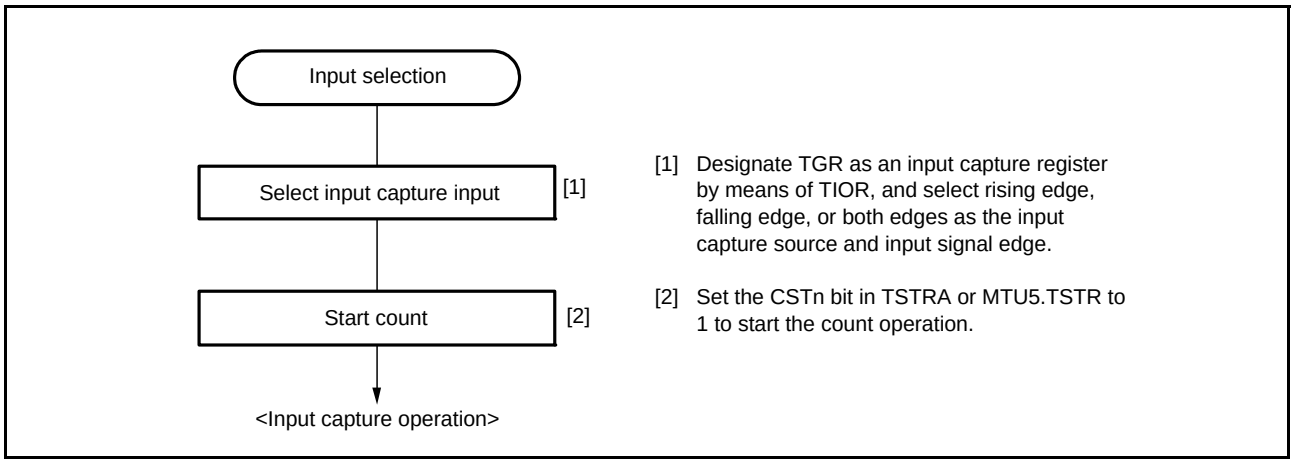


Figure 19.10 Example of Input Capture Operation Setting Procedure

(b) Example of Input Capture Operation

Figure 19.11 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)

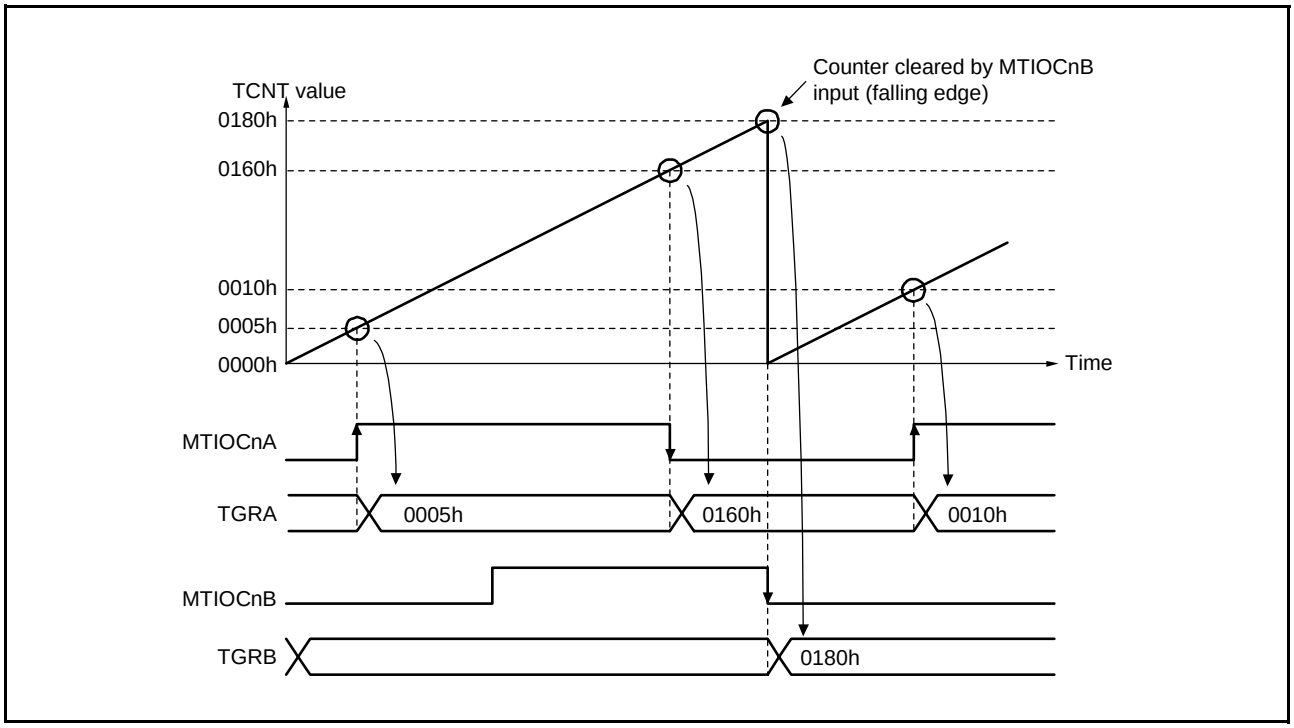


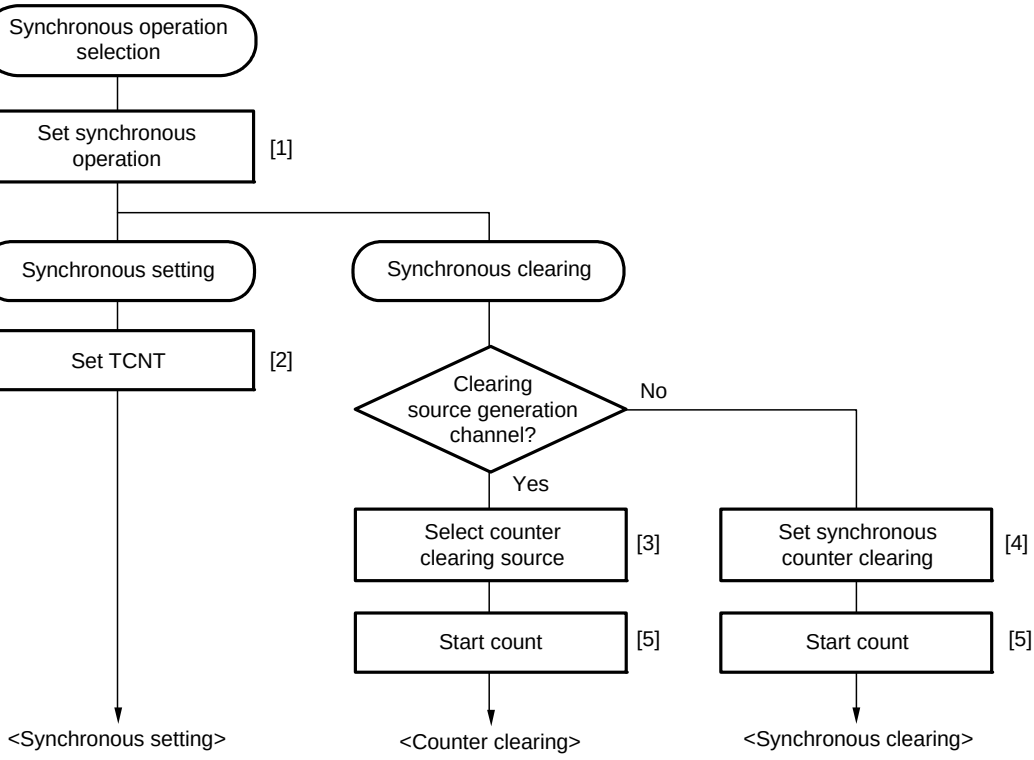
Figure 19.11 Example of Input Capture Operation (n = 0 to 4)

19.3.2 Synchronous Operation

In synchronous operation, the values in multiple TCNT counters can be modified simultaneously (synchronous setting). In addition, multiple TCNT counters can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TCR.
Synchronous operation can increase the number of TGR registers assigned to the same time base.
MTU0 to MTU4 can all be designated for synchronous operation. MTU5 cannot be used for synchronous operation.

(1) Example of Synchronous Operation Setting Procedure

Figure 19.12 shows an example of the synchronous operation setting procedure.



[1] Set to 1 in the SYNC bits in TSYRA corresponding to the channels to be designated for synchronous operation.
[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 the CCLR[2:0] bits in TCR to specify TCNT clearing by input capture/output compare, etc.
[4] Use the CCLR[2:0] bits in TCR to designate synchronous clearing for the counter clearing source.
[5] Set to 1 in the CSTn bits in TSTRA for the relevant channels, to start the count operation.

Figure 19.12 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 19.13 shows an example of synchronous operation.
In this example, synchronous operation and PWM mode 1 have been designated for MTU0 to MTU 2, 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 setting 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 period.
For details of PWM modes, refer to section 19.3.5, PWM Modes.

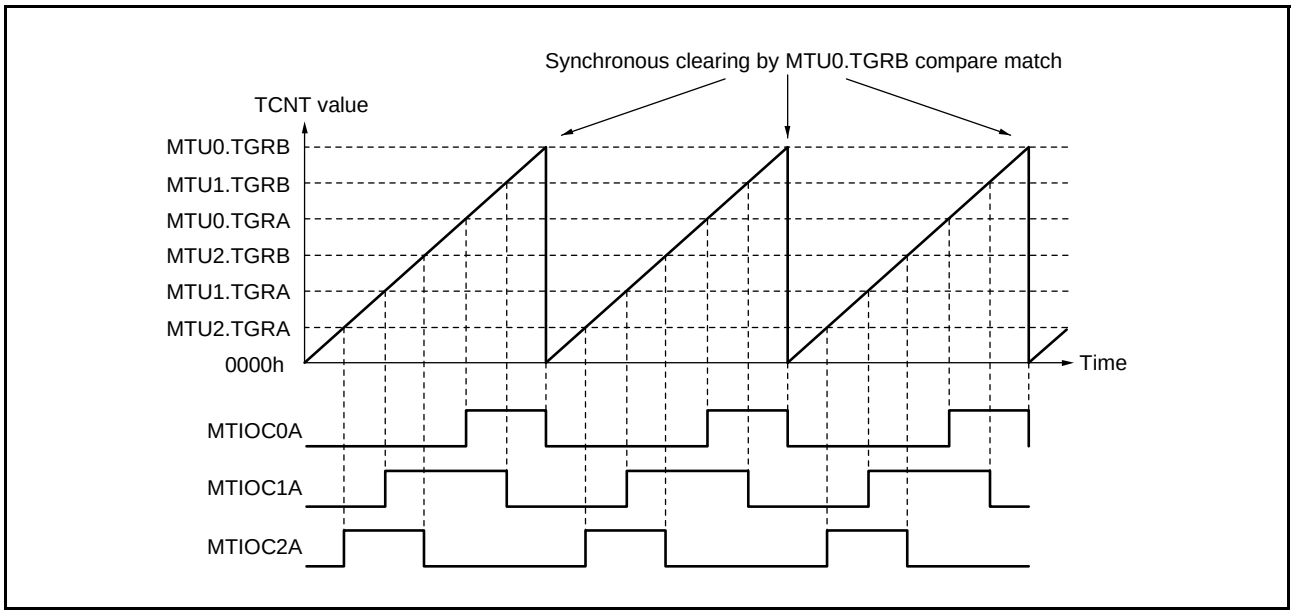


Figure 19.13 Example of Synchronous Operation

19.3.3 Buffer Operation

Buffer operation, provided for MTU0, MTU3, and MTU4, 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 19.46 shows the register combinations used in buffer operation.

Table 19.46 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

- 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 19.14.

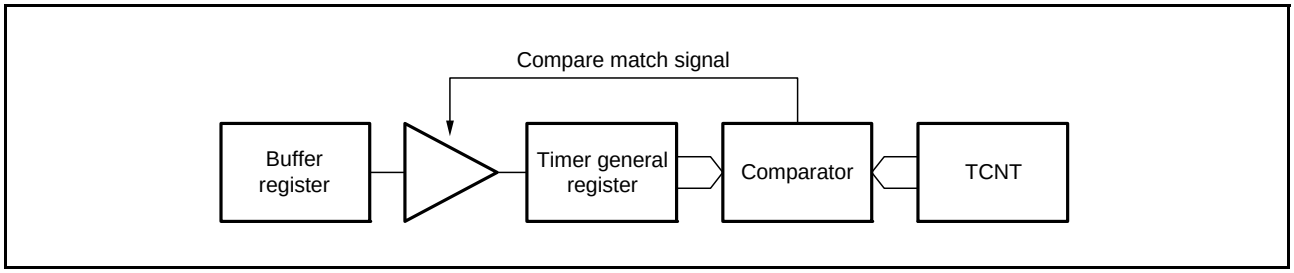


Figure 19.14 Compare Match Buffer Operation

- 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 19.15.

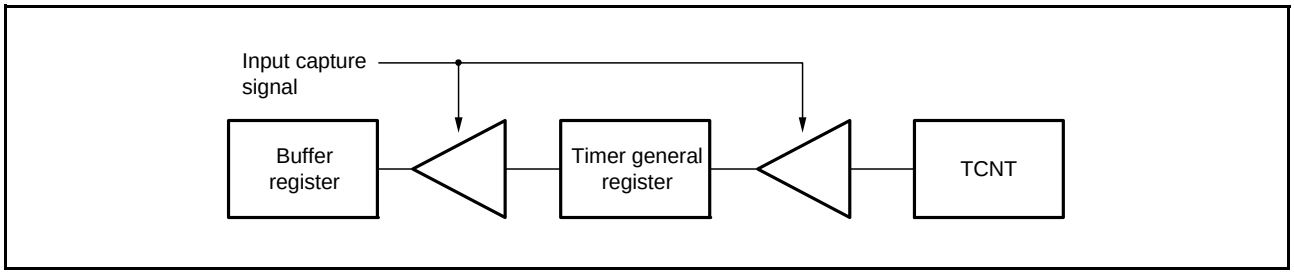


Figure 19.15 Input Capture Buffer Operation

(1) Example of Buffer Operation Setting Procedure

Figure 19.16 shows an example of the buffer operation setting procedure.

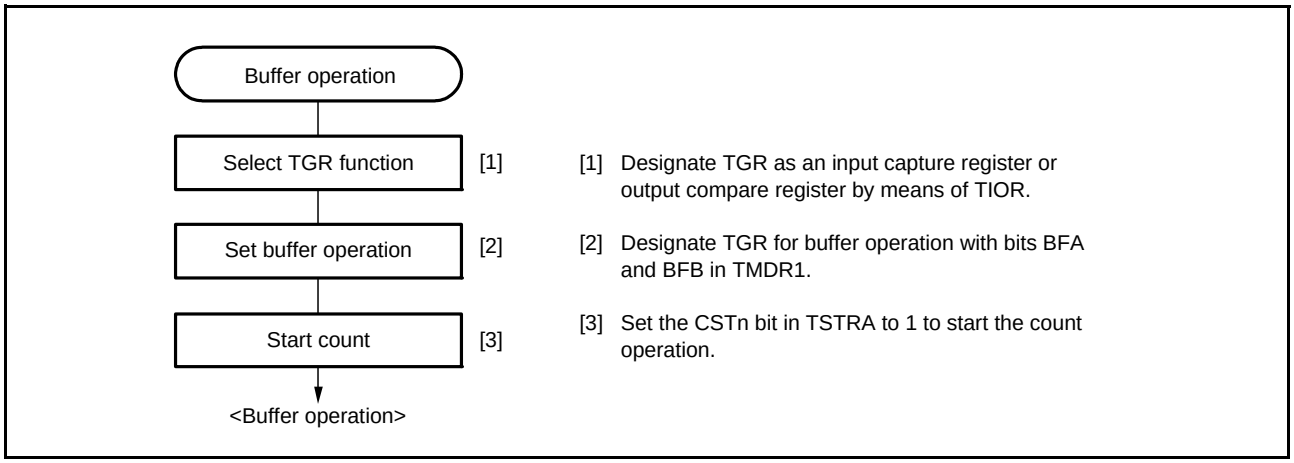


Figure 19.16 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an Output Compare Register

Figure 19.17 shows an operation example in which PWM mode 1 has been designated for MTU0, and buffer operation has been designated for TGRA and TGRC. 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, refer to section 19.3.5, PWM Modes.

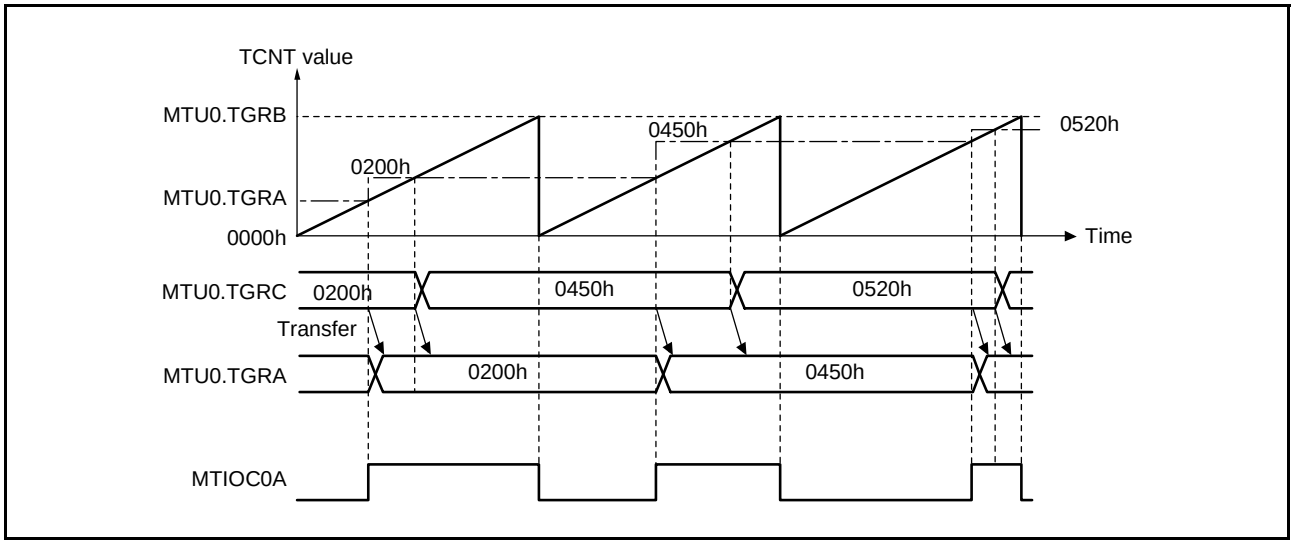


Figure 19.17 Example of Buffer Operation (1)

(b) When TGR is an Input Capture Register

Figure 19.18 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC.
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)
As buffer operation has been set, when the TCNT value is transferred to TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

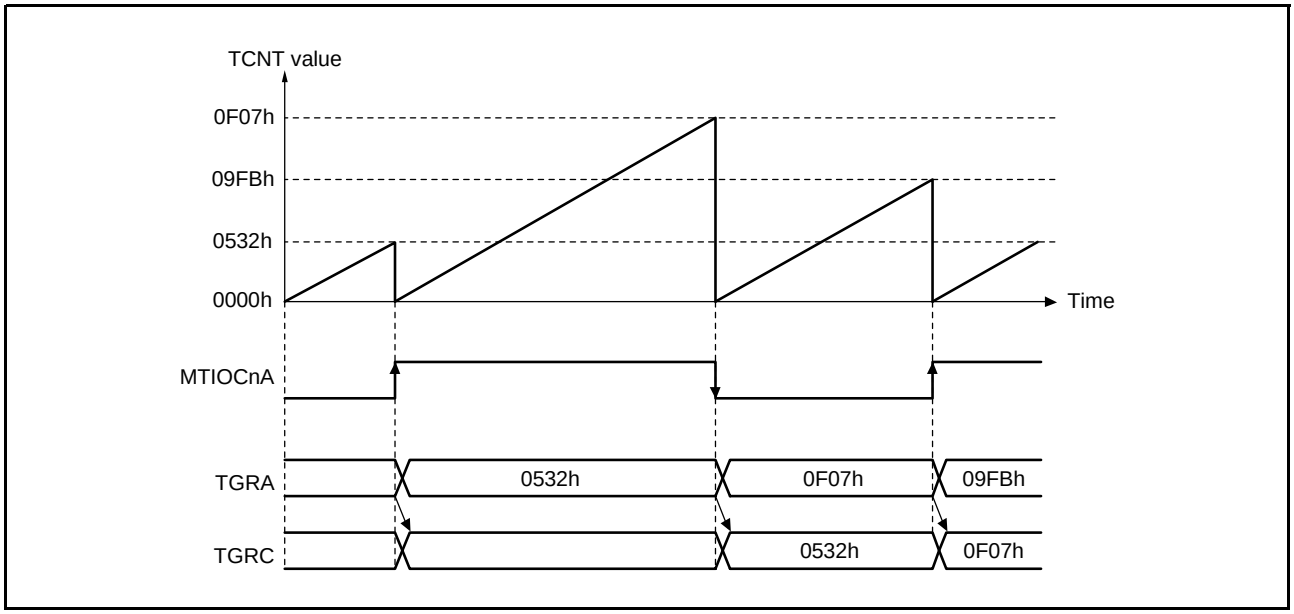


Figure 19.18 Example of Buffer Operation (2) (n = 0 to 4,)

(3) Selecting Timing for Transfer from Buffer Registers to Timer General Registers in Buffer Operation

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 and MTU4 by setting the buffer operation transfer mode registers (MTUn.TBTM (n = 0, 3, 4)). 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 (FFFFh to 0000h)
- When 0000h is written to TCNT during counting
- When TCNT becomes 0000h under the condition specified in the CCLR[2:0] bits in TCR

Note: TBTM must be modified only while TCNT stops.

Figure 19.19 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.

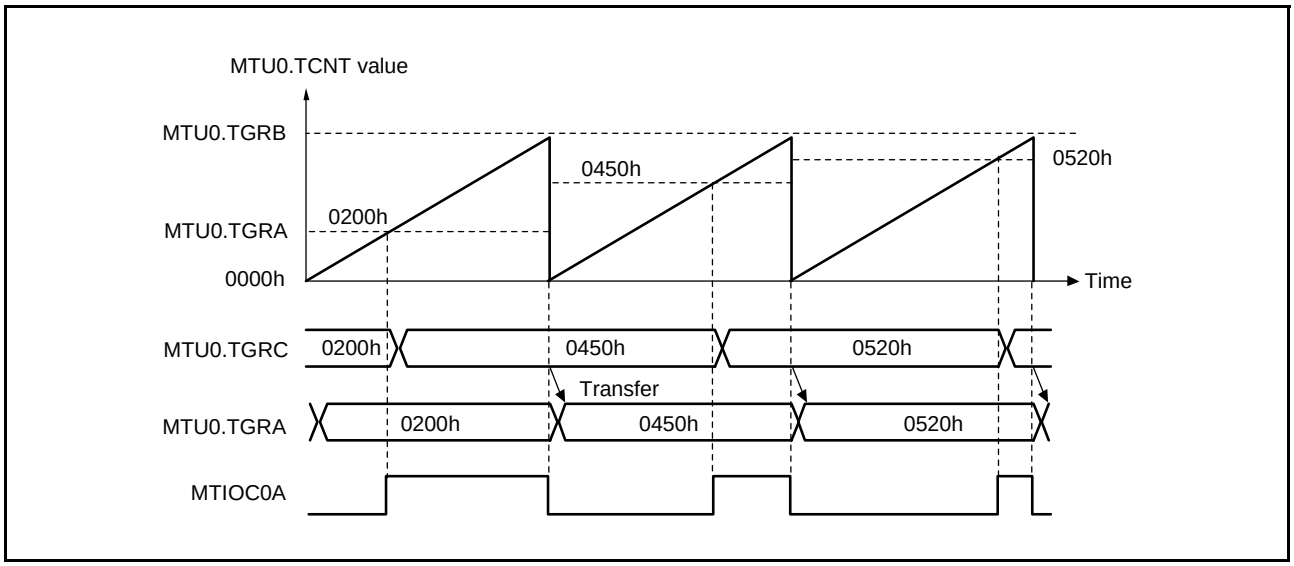


Figure 19.19 Example of Buffer Operation When MTU0.TCNT Clearing is Selected for MTU0.TGRC to MTU0.TGRA Transfer Timing

19.3.4 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, refer to section 19.3.6.2, 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 19.47 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for MTU1, the count clock setting is invalid and the counters operate independently in phase counting mode.

Table 19.47 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, refer to (4), Cascaded Operation Example (c). For input capture in cascade connection, refer to section 19.6.21, Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection.
Table 19.48 shows the TICCR setting and input capture input pins.

Table 19.48 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

(1) Example of Cascaded Operation Setting Procedure

Figure 19.20 shows an example of the cascaded operation setting procedure.

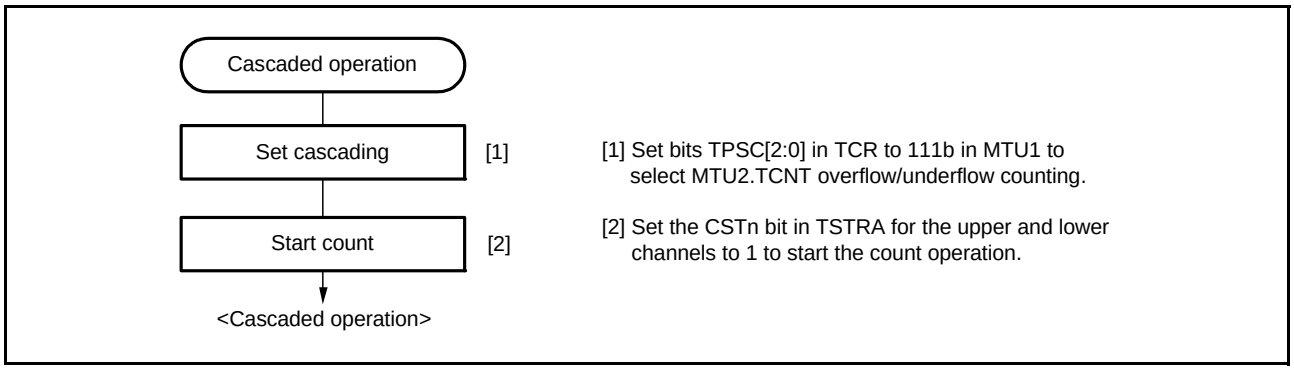


Figure 19.20 Cascaded Operation Setting Procedure

(2) Cascaded Operation Example (a)

Figure 19.21 shows the operation when MTU1.TCNT is set for counting at MTU2.TCNT overflow/underflow and MTU2 is set for phase counting mode 1 while MTU1.TCNT and MTU2.TCNT are cascaded.
MTU1.TCNT is incremented by MTU2.TCNT overflow and decremented by MTU2.TCNT underflow.

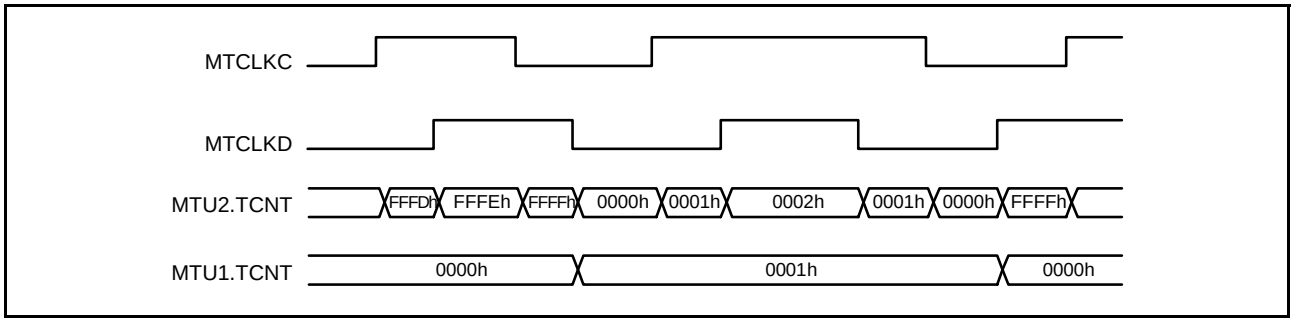


Figure 19.21 Cascaded Operation Example (a)

(3) Cascaded Operation Example (b)

Figure 19.22 illustrates 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.

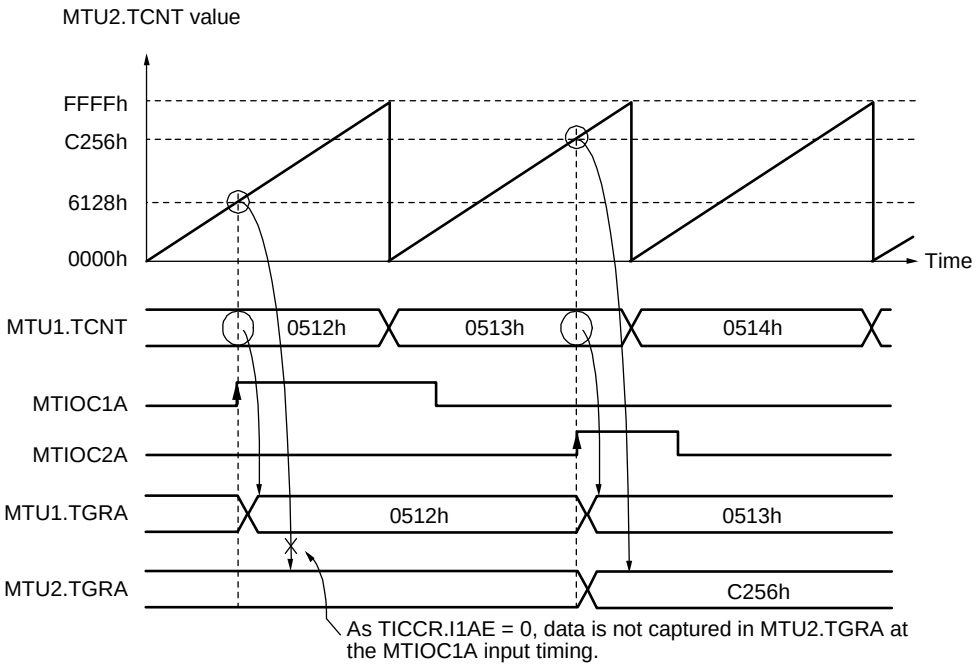
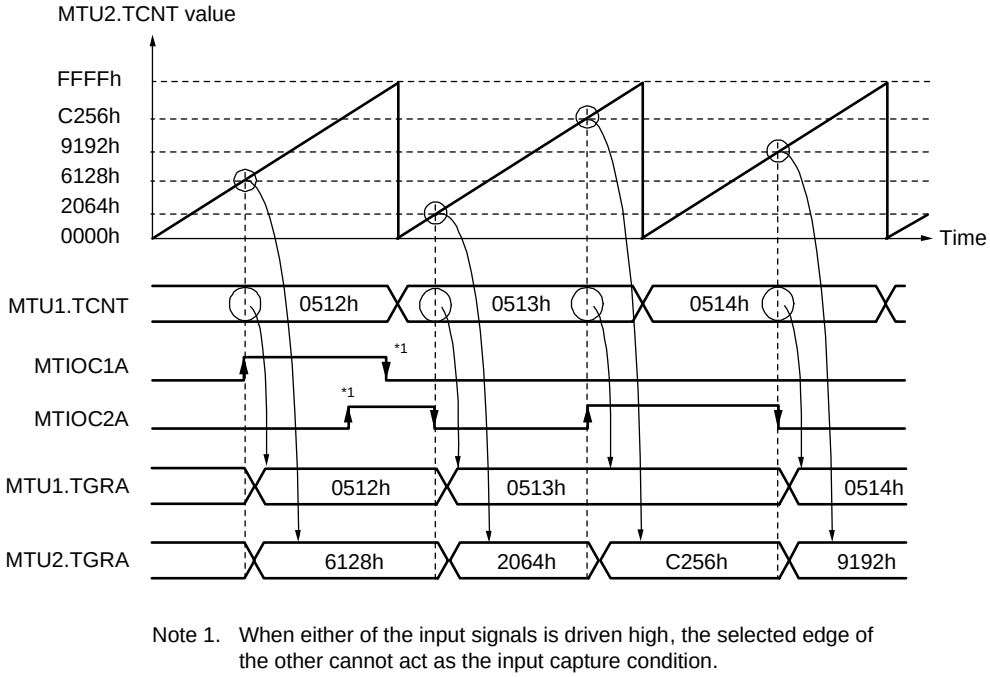


Figure 19.22 Cascaded Operation Example (b)

(4) Cascaded Operation Example (c)

Figure 19.23 illustrates 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 19.23 Cascaded Operation Example (c)

(5) Cascaded Operation Example (d)

Figure 19.24 illustrates 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.

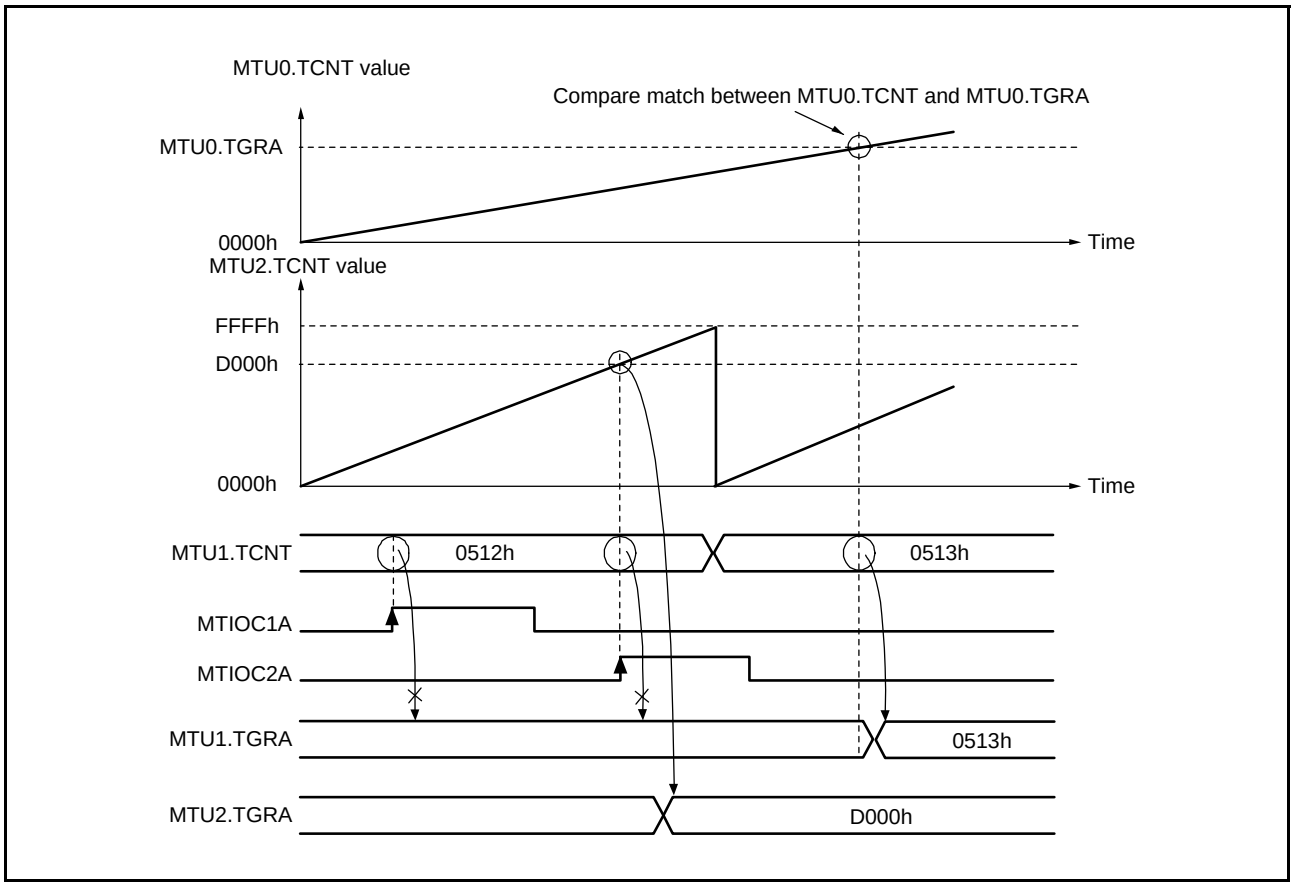


Figure 19.24 Cascaded Operation Example (d)

19.3.5 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 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.

(a) 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). 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 eight phases can be output.

(b) 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 19.49.

Table 19.49 PWM Output Registers and Output Pins

		Output Pins	
Channel	**Register**	**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		

Note: In PWM mode 2, PWM waveform output is not possible for the TGR register in which the PWM period is set.

(1) Example of PWM Mode Setting Procedure

Figure 19.25 shows an example of the PWM mode setting procedure.

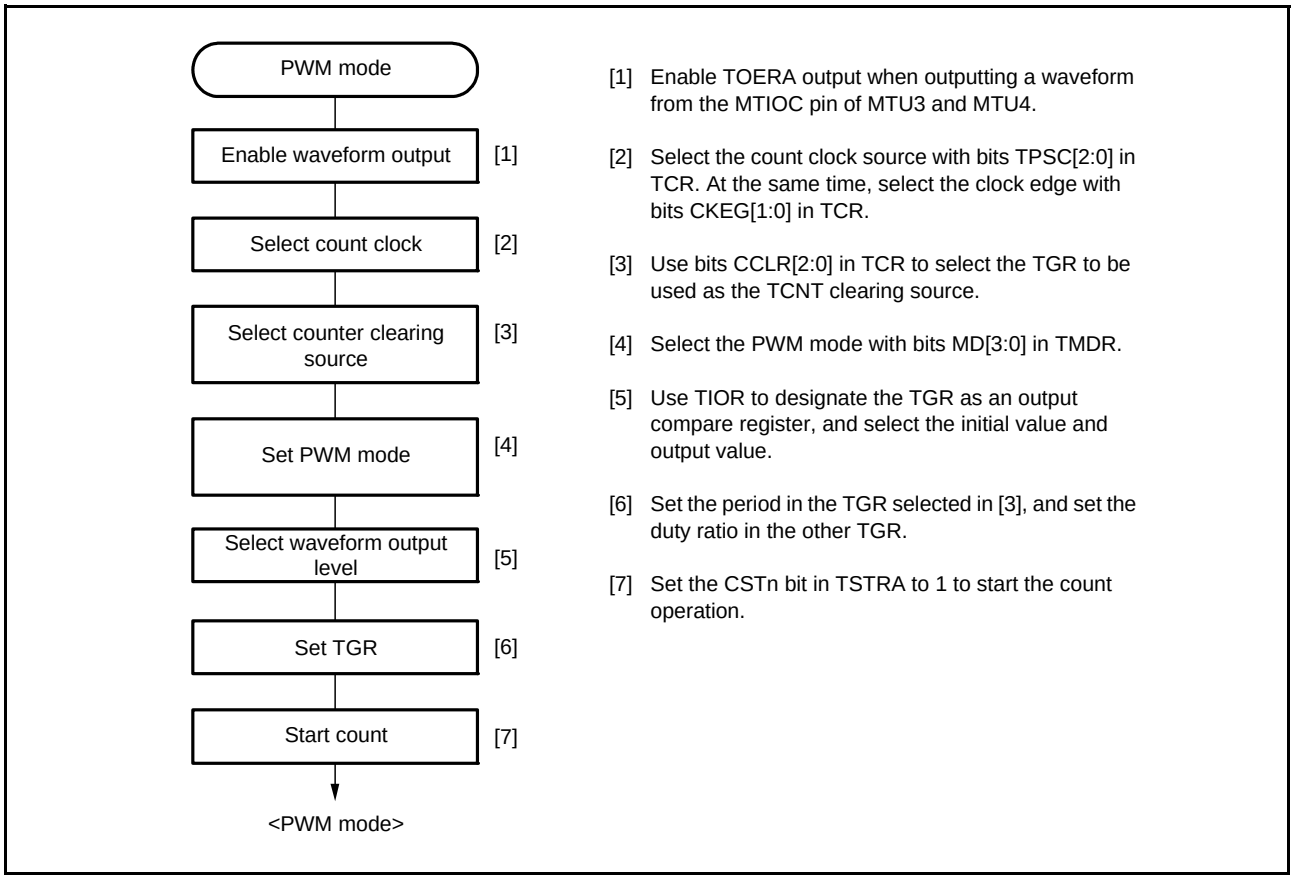


Figure 19.25 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 19.26 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.

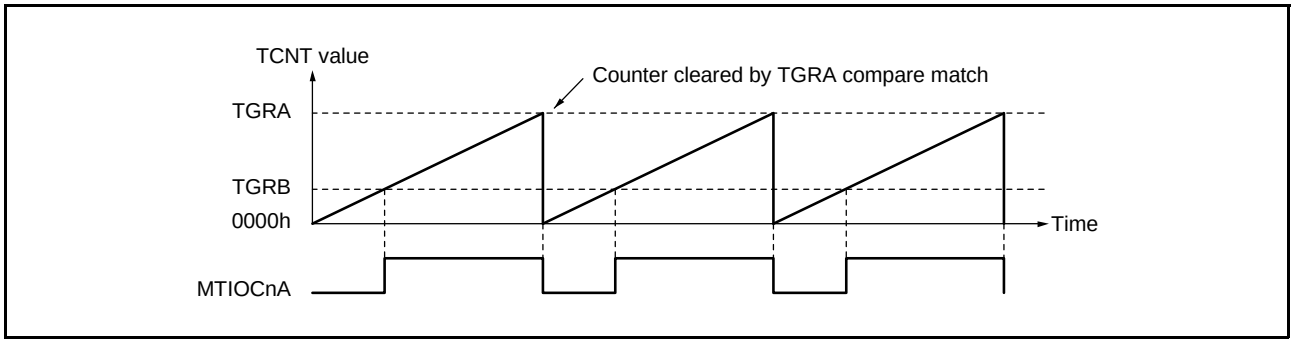


Figure 19.26 Example of PWM Mode 1 Operation (n = 0 to 4)

Figure 19.27 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.

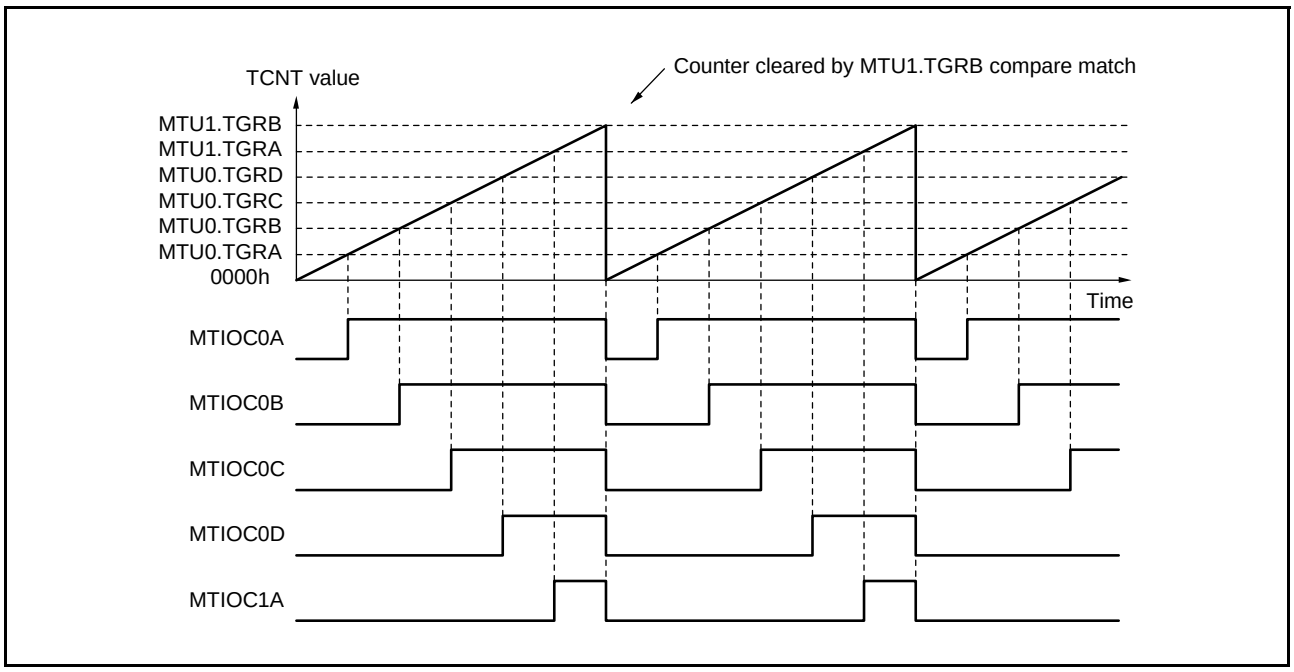


Figure 19.27 Example of PWM Mode 2 Operation

Figure 19.28 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode 1.
In this example, TGRA compare match is set as the TCNT clearing source, a low level is set as the initial output value and output value for TGRA, and a high level is set as the output value for TGRB.

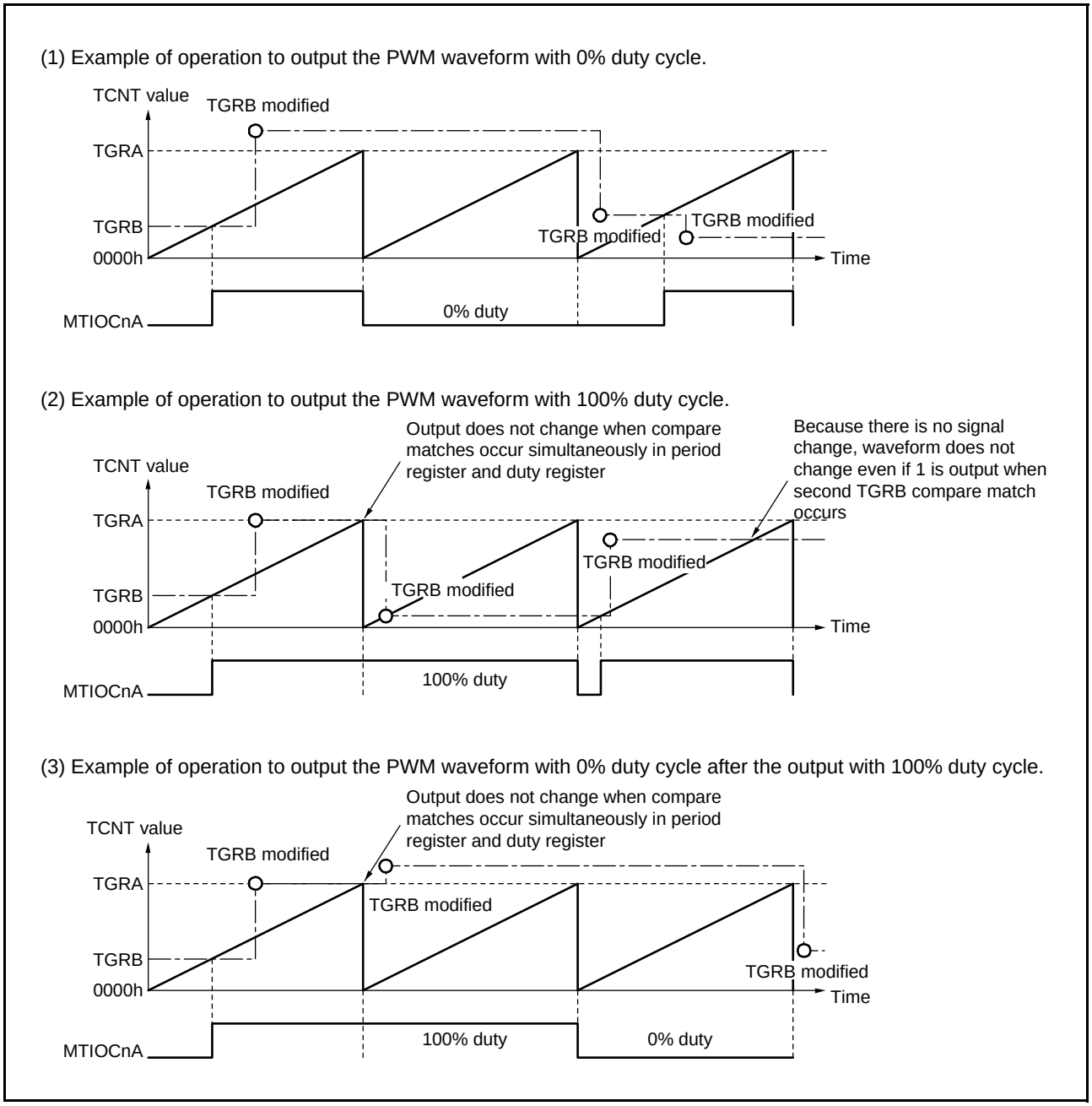


Figure 19.28 Examples of PWM Mode Operation (PWM Waveform Output with 0% Duty and 100% Duty) (n = 0 to 4)

19.3.6 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 19.50 lists the external clock input pins to be connected in each phase counting mode.

Table 19.50 Clock Input Pins in Phase Counting Mode

		External Clock Input Pins	
Phase Counting Mode	**TMDR3.PHCKSEL bit**	**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

19.3.6.1 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.

(1) Example of 16-Bit Phase Counting Mode Setting Procedure

Figure 19.29 shows an example of the phase counting mode setting procedure.

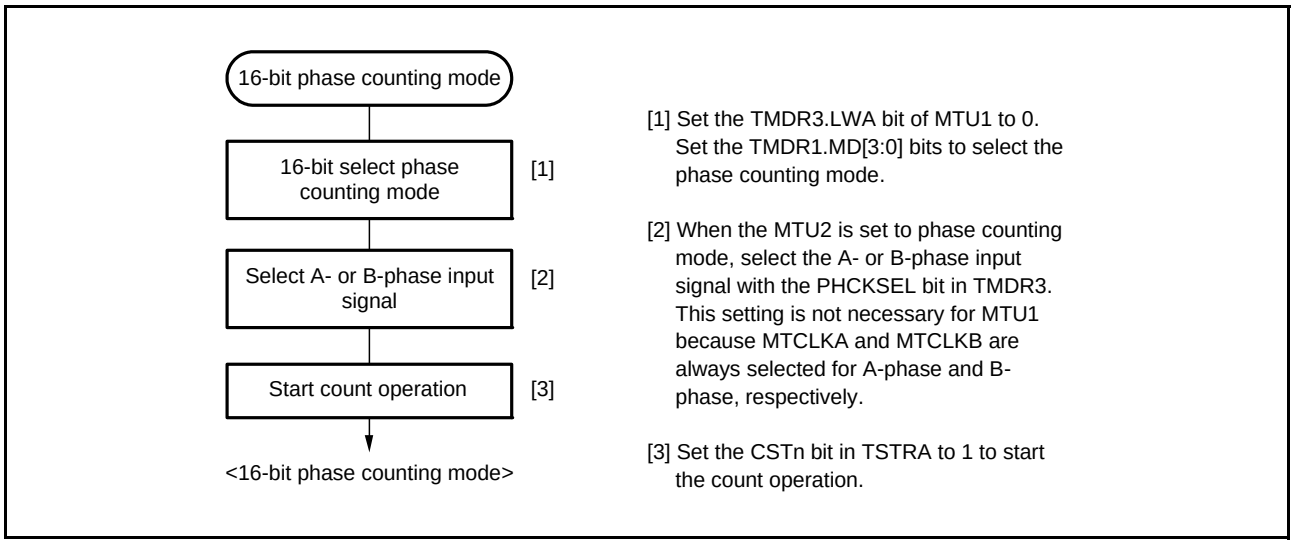


Figure 19.29 Example of 16-Bit Phase Counting Mode Setting Procedure

(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.

(a) Phase Counting Mode 1

Figure 19.30 shows an example of operation in phase counting mode 1, and Table 19.51 summarizes the TCNT up-counting and down-counting conditions.

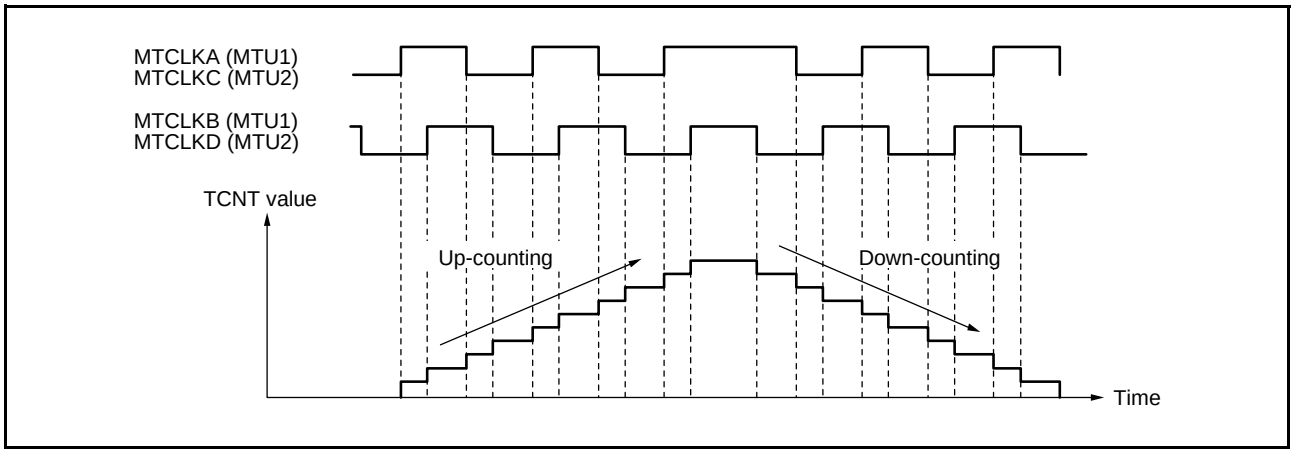


Figure 19.30 Example of Operation in Phase Counting Mode 1

Table 19.51 Up-Counting and Down-Counting Conditions in Phase Counting Mode 1

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High	↑	Up-counting
Low	↓	
↑	Low	
↓	High	
High	↓	Down-counting
Low	↑	
↑	High	
↓	Low	

↑ : Rising edge

↓ : Falling edge

(b) Phase Counting Mode 2

Figure 19.31 to Figure 19.33 show the examples of operation in phase counting mode 2 and Table 19.52 summarizes the TCNT up-counting and down-counting conditions.

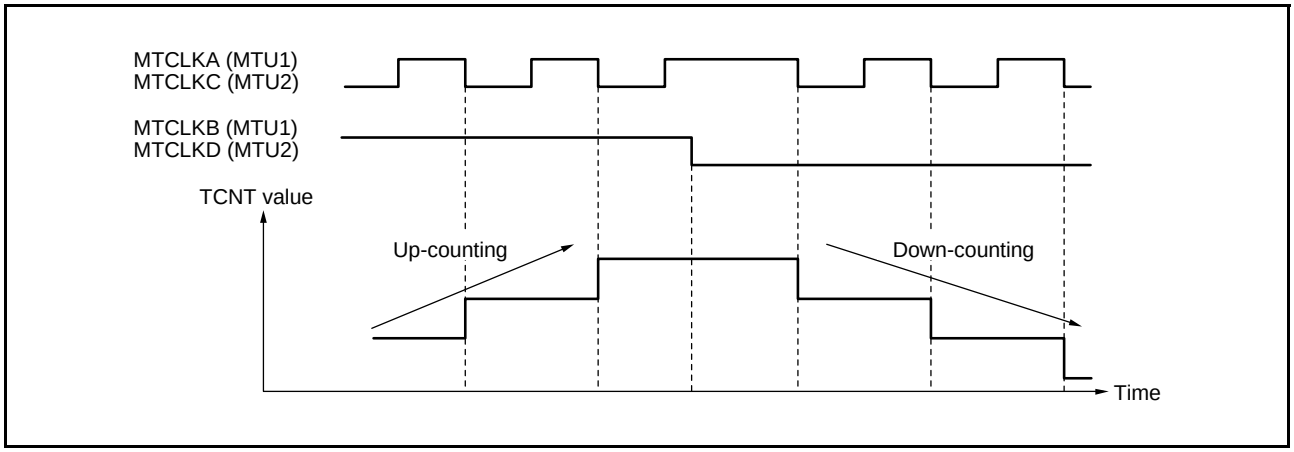


Figure 19.31 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] = 00b (n = 1, 2))

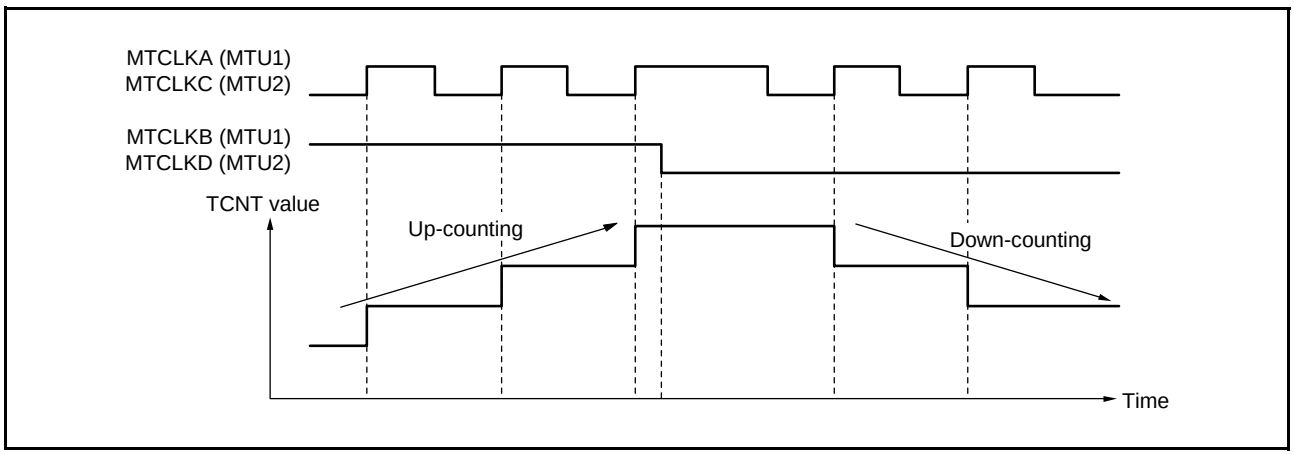


Figure 19.32 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] = 01b (n = 1, 2))

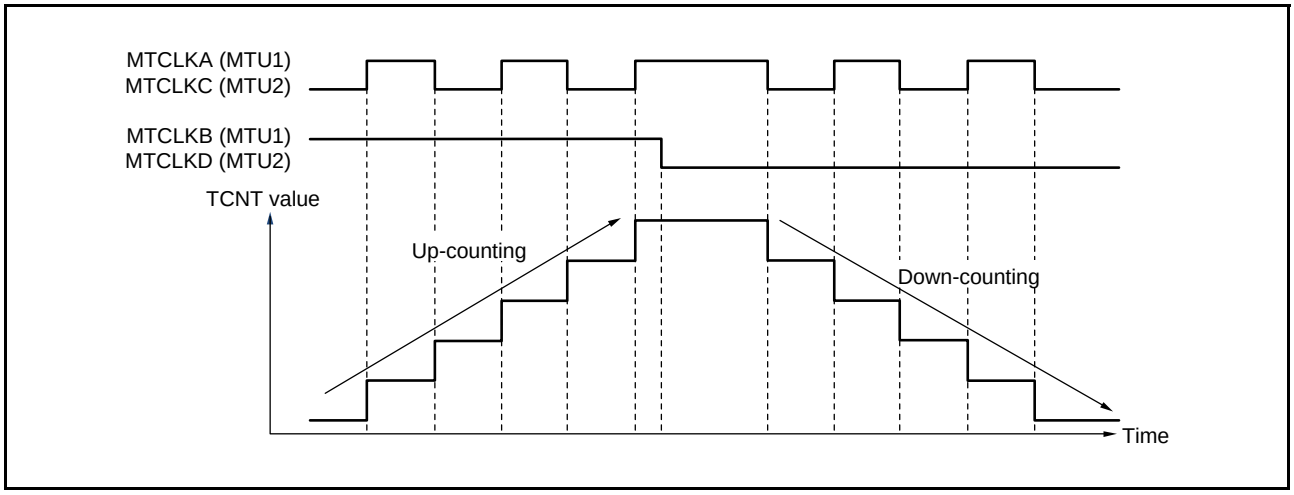


Figure 19.33 Example of Operation in Phase Counting Mode 2 (When MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 19.52 Up-Counting and Down-Counting Conditions in Phase Counting Mode 2

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

↑ : Rising edge

↓ : Falling edge

(c) Phase Counting Mode 3

Figure 19.34 to Figure 19.36 show the examples of operation in phase counting mode 3 and Table 19.53 summarizes the TCNT up-counting and down-counting conditions.

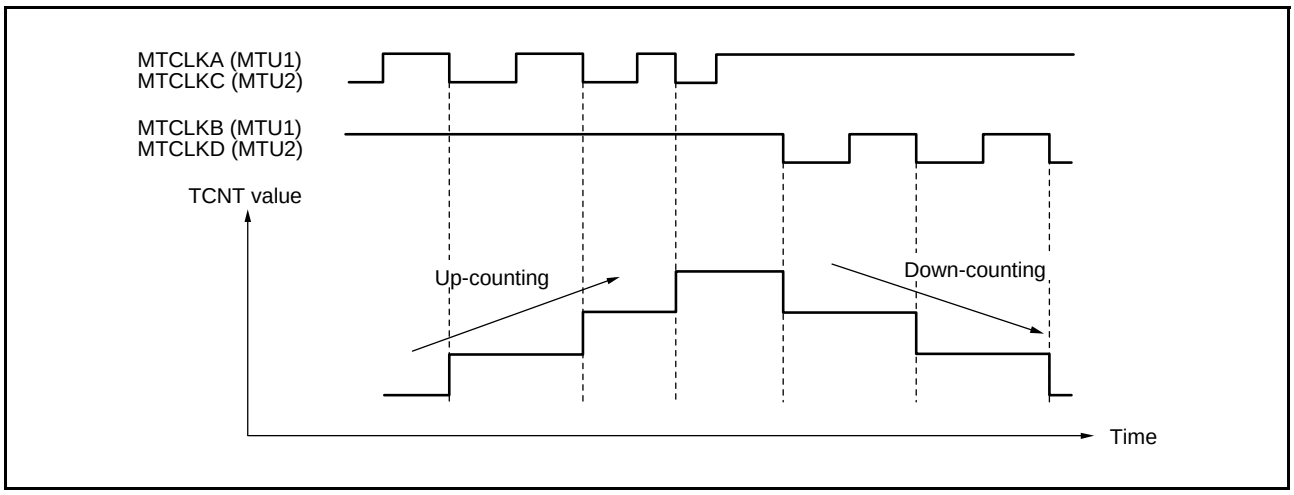


Figure 19.34 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] = 00b (n = 1, 2))

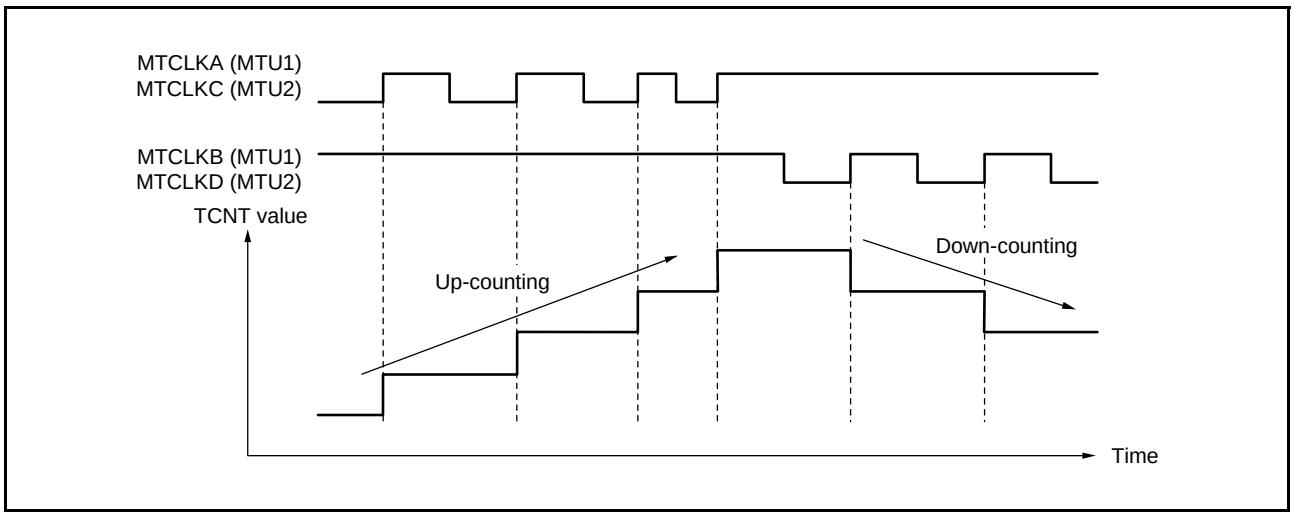


Figure 19.35 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] = 01b (n = 1, 2))

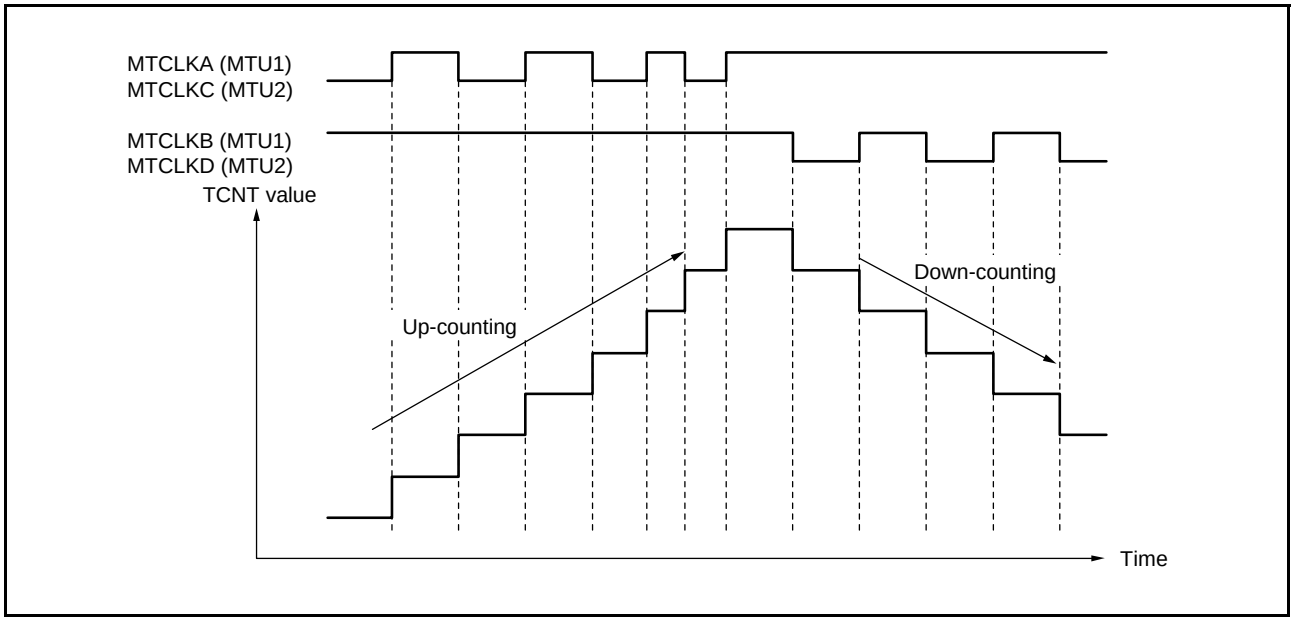


Figure 19.36 Example of Operation in Phase Counting Mode 3 (When MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 19.53 Up-Counting and Down-Counting Conditions in Phase Counting Mode 3

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)

↑ : Rising edge

↓ : Falling edge

(d) Phase Counting Mode 4

Figure 19.37 shows an example of operation in phase counting mode 4, and Table 19.54 summarizes the TCNT up-counting and down-counting conditions.

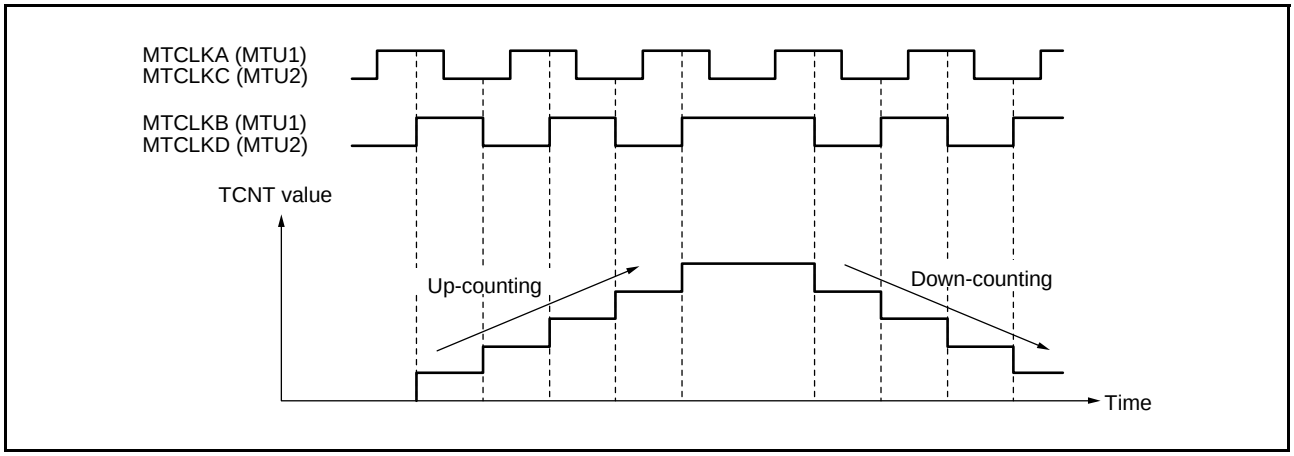


Figure 19.37 Example of Operation in Phase Counting Mode 4

Table 19.54 Up-Counting and Down-Counting Conditions in Phase Counting Mode 4

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	

↑ : Rising edge

↓ : Falling edge

(e) Phase Counting Mode 5

Figure 19.38 and Figure 19.39 show the examples of operation in phase counting mode 5 and Table 19.55 summarizes the TCNT up-counting and down-counting conditions.

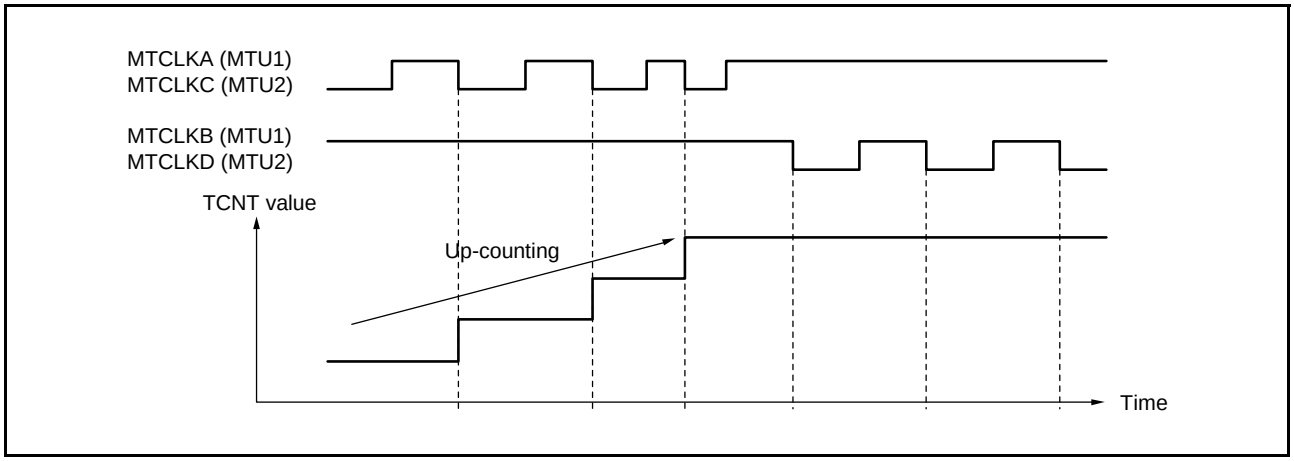


Figure 19.38 Example of Operation in Phase Counting Mode 5 (When MTUn.TCR2.PCB[1:0] = 0xb (n = 1, 2))

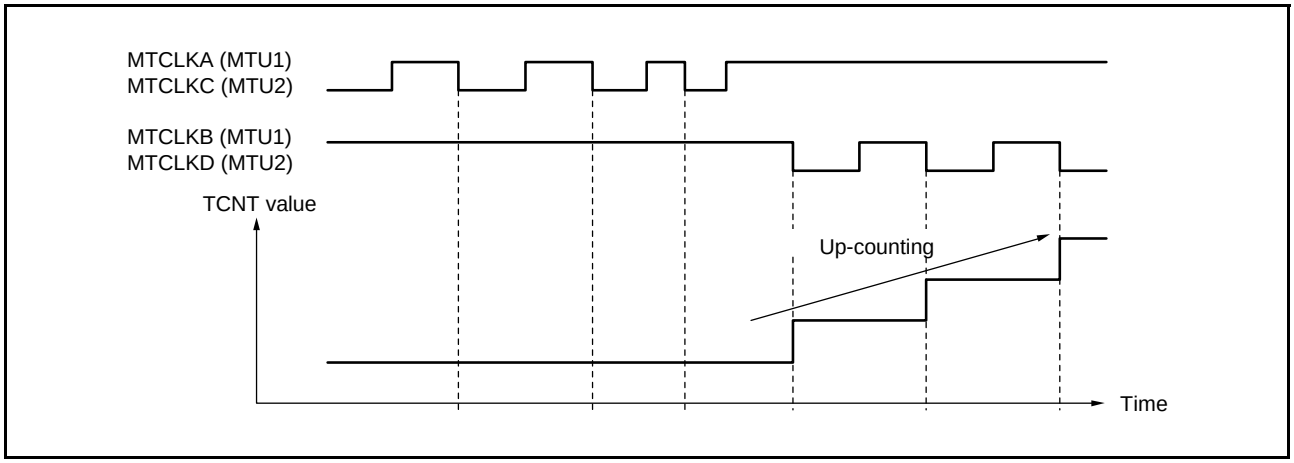


Figure 19.39 Example of Operation in Phase Counting Mode 5 (When MTUn.TCR2.PCB[1:0] = 1xb (n = 1, 2))

Table 19.55 Up-Counting and Down-Counting Conditions in Phase Counting Mode 5

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
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	

↑ : Rising edge

↓ : Falling edge

(3) 16-Bit Phase Counting Mode Application Example

Figure 19.40 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.

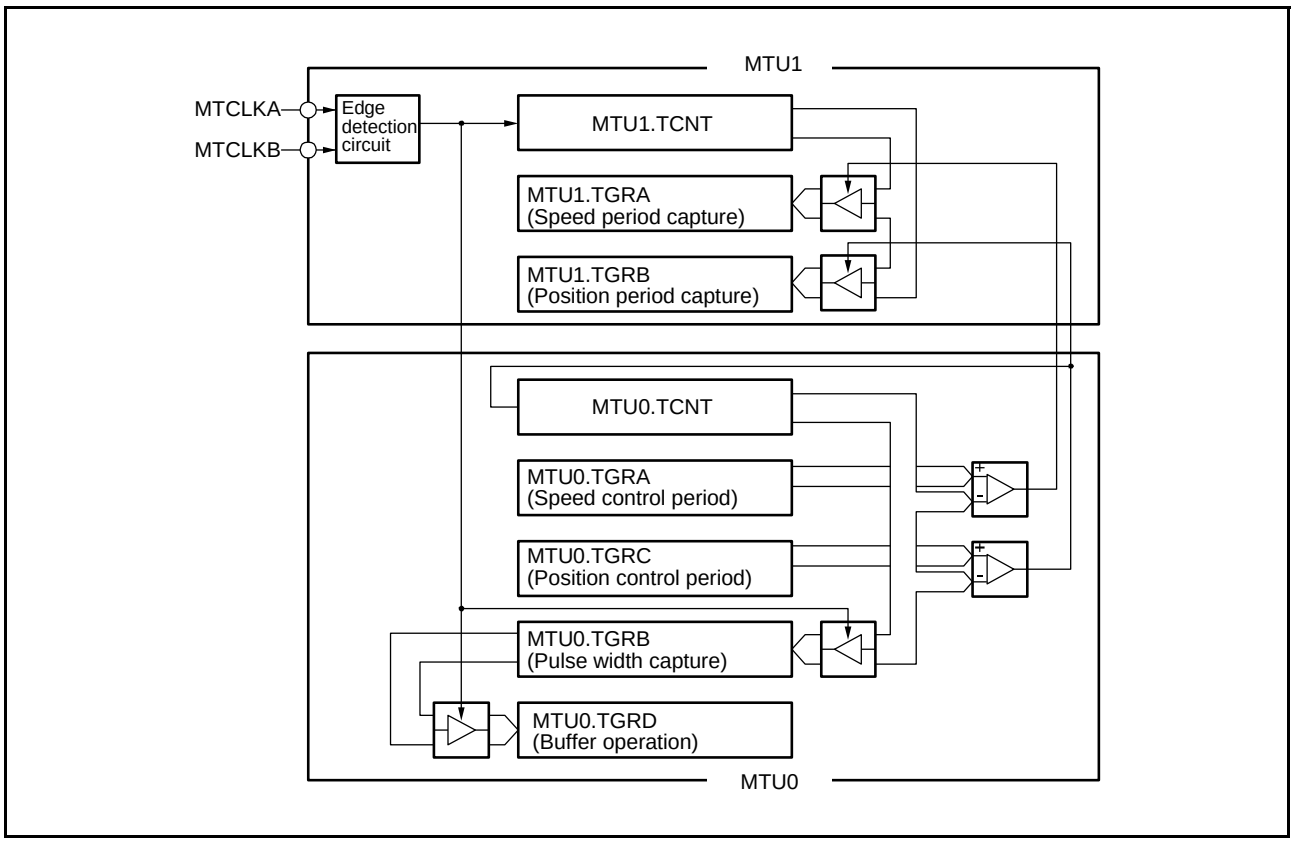


Figure 19.40 16-Bit Phase Counting Mode Application Example

19.3.6.2 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. 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. Refer to Figure 19.41 for the procedure for setting cascade connection 32-bit phase counting mode.
Refer to section 19.3.4, 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.

(1) Example of Setting Cascade Connection 32-Bit Phase Counting Mode

Figure 19.41 shows an example of the procedure for setting cascade connection 32-bit phase counting mode.

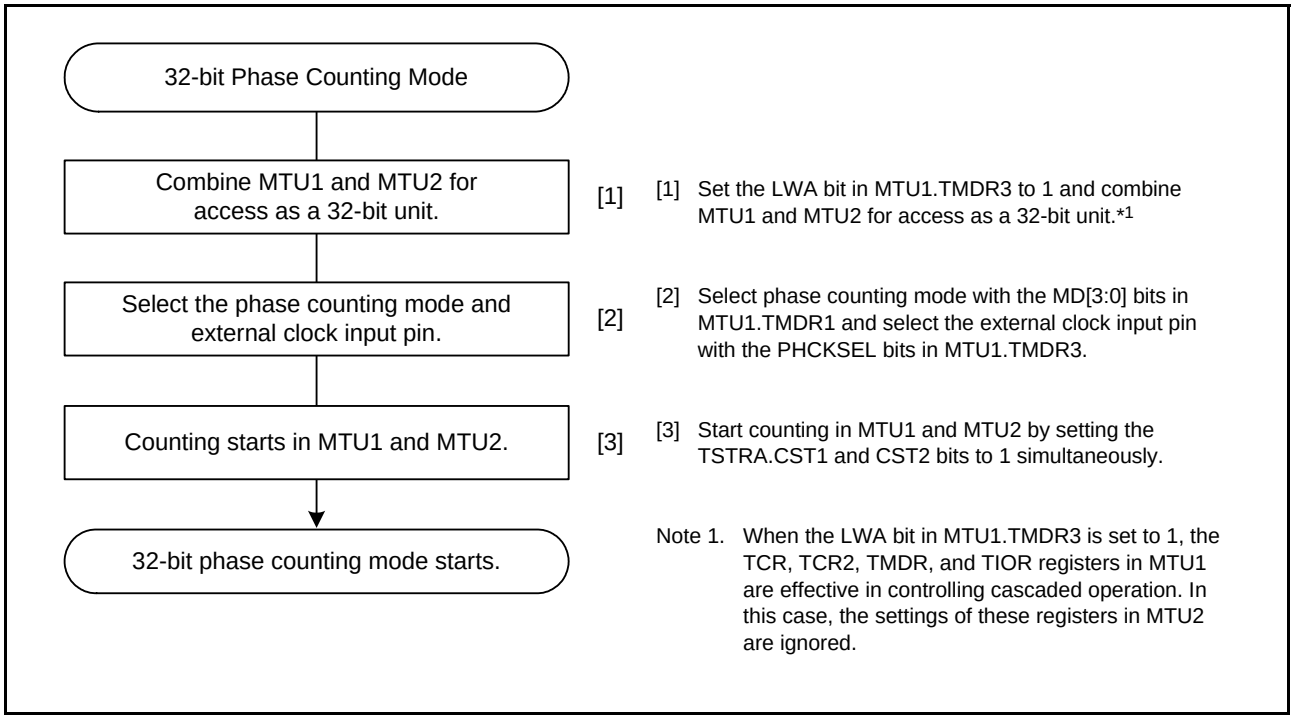


Figure 19.41 Procedure for Setting Cascade Connection 32-Bit Phase Counting Mode

19.3.7 Reset-Synchronized PWM Mode

In the reset-synchronized PWM mode, three phases of positive and negative PWM waveforms (six phases in total) that share a common wave transition point can be output by combining MTU3 and MTU4.
When set for reset-synchronized PWM mode, the MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, and MTIOC4D pins function as PWM output pins and timer counter 3 (MTU3.TCNT) functions as an up-counter.
Table 19.56 shows the PWM output pins used. Table 19.57 shows the settings of the registers.

Table 19.56 Output Pins for Reset-Synchronized PWM Mode

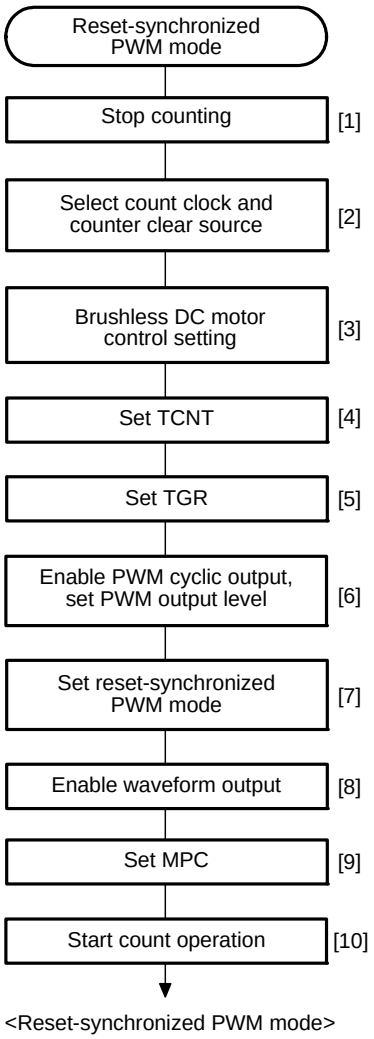
Channel	Output Pin	Description
MTU3	MTIOC3B	PWM output pin 1
	MTIOC3D	PWM output pin 1' (negative-phase waveform of PWM output 1)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (negative-phase waveform of PWM output 2)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (negative-phase waveform of PWM output 3)

Table 19.57 Register Settings for Reset-Synchronized PWM Mode

Register	Setting
MTU3.TCNT	Initial setting (0000h)
MTU4.TCNT	Initial setting (0000h)
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

(1) Example of Procedure for Setting Reset-Synchronized PWM Mode

Figure 19.42 shows an example of procedure for setting the reset-synchronized PWM mode.



[1] Set the TSTRA.CST3 and TSTRA.CST4 bits to 0 to stop the TCNT count operation.
Specify the reset-synchronized PWM mode while MTU3.TCNT and MTU4.TCNT are stopped.

[2] Set the TCR.TPSC[2:0] and TCR.CKEG[1:0] bits of MTU3 to select the count clock source and clock edge for MTU3.
Set the TCR.CCLR[2:0] bits to select TGRA compare match as the counter clear source.

[3] To control a brushless DC motor, set the TGCRA.BDC bit and set the feedback signal input source and output chopping or gate signal direct output (only when MTU3 and MTU4 are used).

[4] Set MTU3.TCNT and MTU4.TCNT to 0000h.

[5] MTU3.TGRA is the period register. Set the PWM period in MTU3.TGRA.
Set the PWM output transition timing in MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB.
Note that the setting values should be within the range of compare match with MTU3.TCNT.

X ≤ MTU3.TGRA (X: Set value)

[6] Set the PSYE bit in the TOCR1A register to enable or disable toggle output synchronized with the PWM period, and set the OLSP and OLSN bits to select the PWM output level.
When specifying the PWM output level using the TOCR2A and TOLBRA in buffer operation, refer to Timer Output Level Buffer Registers (TOLBRA).

[7] Set the MD[3:0] bits in MTU3.TMDR1 register to 1000b to select the reset-synchronized PWM mode. Specify the buffer operation (BFA = 1) of the MTU3.TGRA register and the buffer operation (BFB = 1) of the MTU3.TGRB register when necessary. Set the MTU4.TMDR1 register to the initial value (normal operation; BFA = 0 and BFB = 0).

[8] Set the TOERA register to enable or disable output from the PWM output pin. Then, set the MTU3.TIORH, MTU3.TIORL, MTU4.TIORH, and MTU4.TIORL registers to 00h.

[9] Set the MPU and the port mode register (PMR) of the I/O ports.

[10] Set the TSTRA.CST3 bit to 1 to start count operation.

Note: The output waveform starts to toggle at the point of MTU3.TCNT = MTU3.TGRA = X by setting X = MTU3.TGRA, i.e., period = duty ratio.

Figure 19.42 Procedure for Selecting Reset-Synchronized PWM Mode

(2) Example of Reset-Synchronized PWM Mode Operation

Figure 19.43 shows an example of operation in the reset-synchronized PWM mode.
MTU3.TCNT and MTU4.TCNT operate as up-counters. The counters are cleared when a compare match occurs between MTU3.TCNT and MTU3.TGRA, and then begin incrementing from 0000h. The output from the PWM pins toggles every time a compare match occurs in MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB and the counters are cleared.

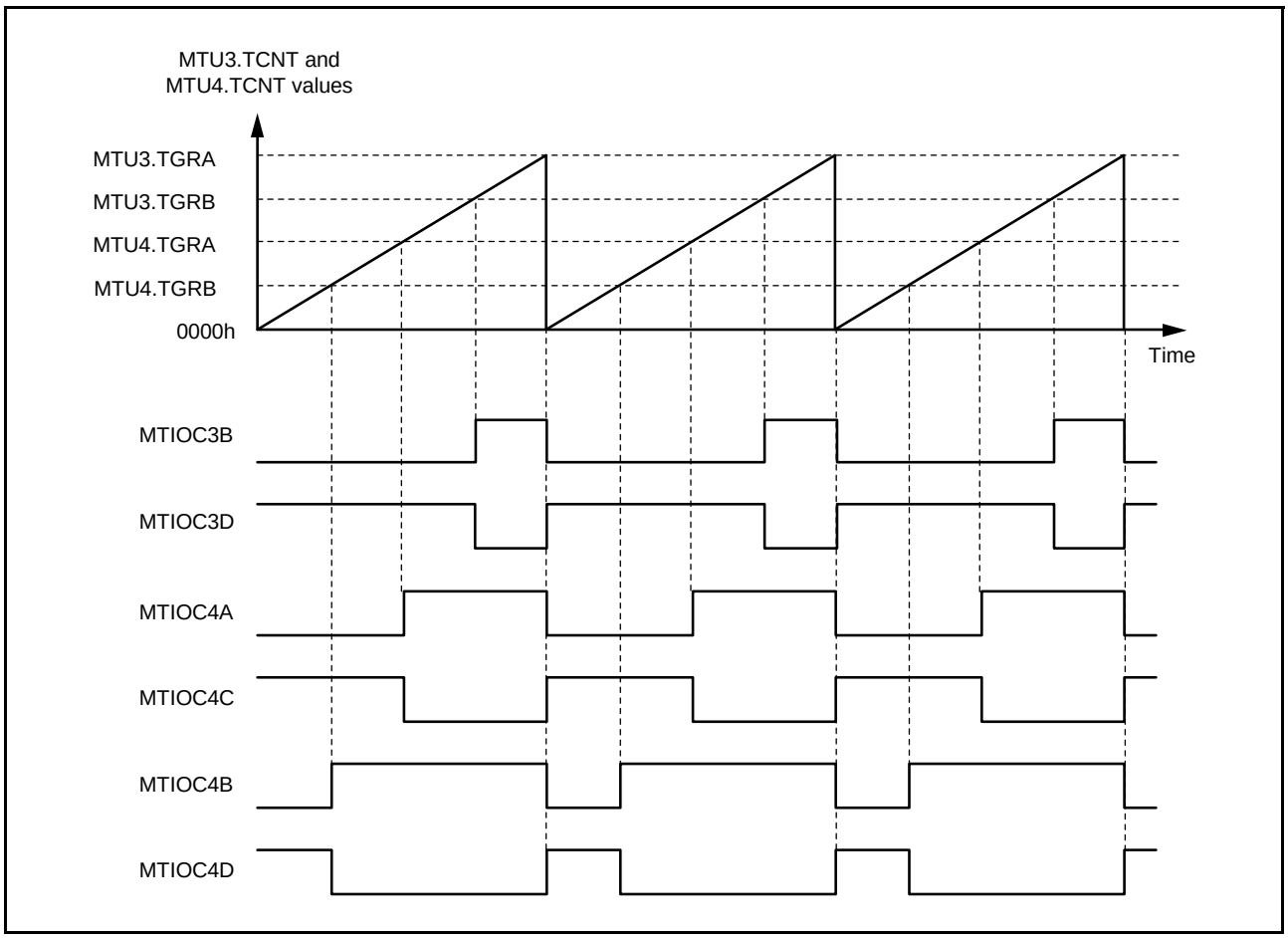


Figure 19.43 Example of Reset-Synchronized PWM Mode Operation (When OLSN = 1 and OLSP = 1 in MTU3.TOCR1 and MTU4.TOCR1)

19.3.8 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 three negative-phase PWM waveforms (six phases in total) with dead time can be output by combining MTU3/ MTU4. PWM waveforms without dead time can also be output.

In complementary PWM mode, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D pins function as PWM output pins, and the MTIOC3A pin can be set for toggle output synchronized with the PWM period. MTU3.TCNT and MTU4.TCNT function as up/down-counters.

Table 19.58 shows the PWM output pins used. Table 19.59 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 19.58 Output Pins for Complementary PWM Mode

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' (negative-phase waveform output of PWM output 1)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (negative-phase waveform output of PWM output 1)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (negative-phase waveform output of PWM output 1)

Note 1. Avoid setting the MTIOC3C pin as timer I/O pins in complementary PWM mode.

Table 19.59 Register Settings for Complementary PWM Mode (1/2)

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 0000h	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

Note 1. Access can be enabled or disabled according to the setting in TRWERA (timer read/write enable register A).

Table 19.60 Register Settings for Complementary PWM Mode (2/2)

Channel	Counter/ Register	Description	Read/Write from CPU
Timer dead time data register A (TDDRA)		Set MTU4.TCNT and MTU3.TCNT offset value (dead time value)	Maskable by TRWERA setting*1
Timer period data register A (TCDRA)		Set MTU4.TCNT upper limit value (1/2 carrier period)	Maskable by TRWERA setting*1
Timer period buffer register A (TCBRA)		TCDRA buffer register	Readable/writable
Subcounter A (TCNTSA)		Subcounter A 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

Note 1. Access can be enabled or disabled according to the setting in TRWERA (timer read/write enable register A).

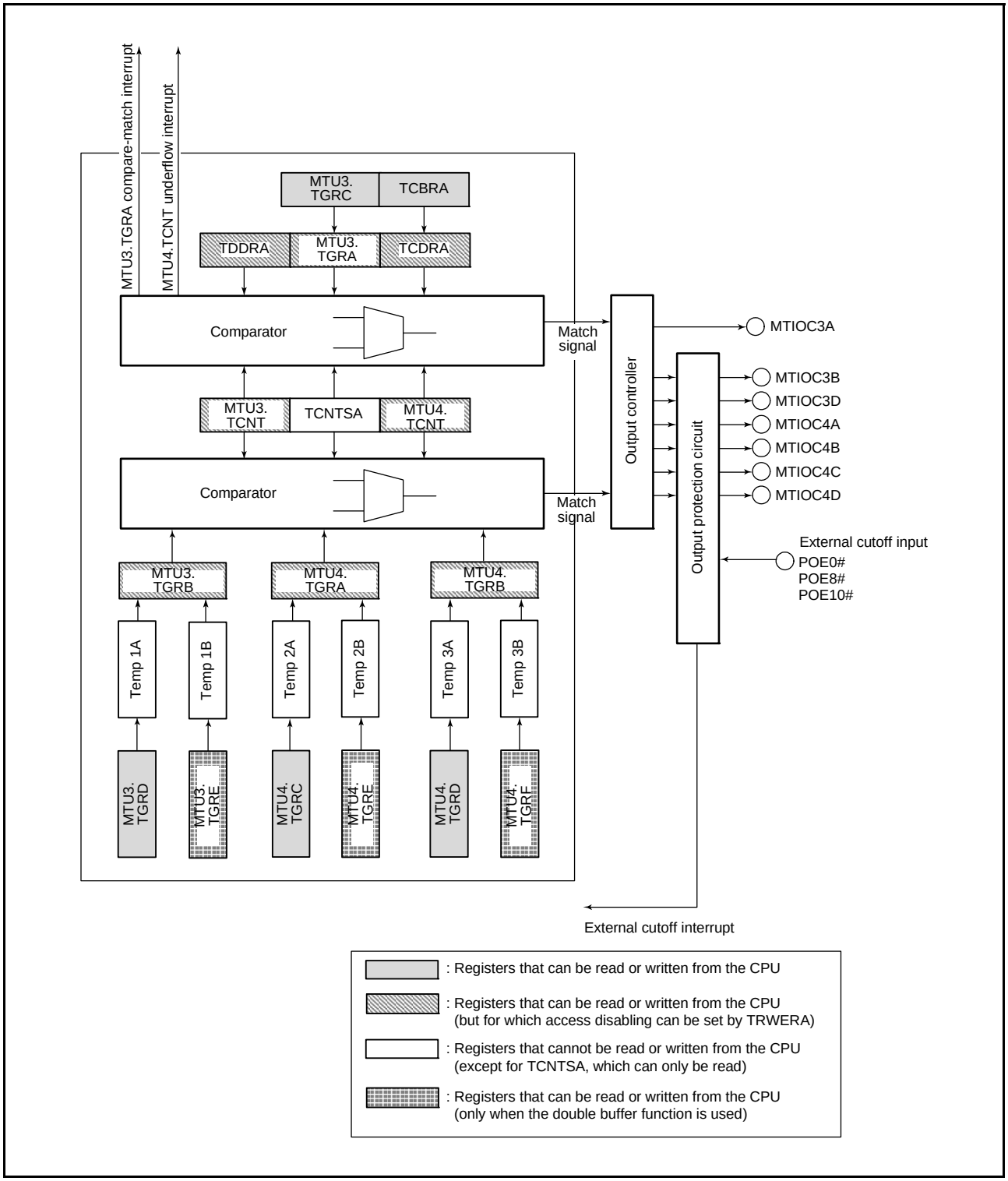
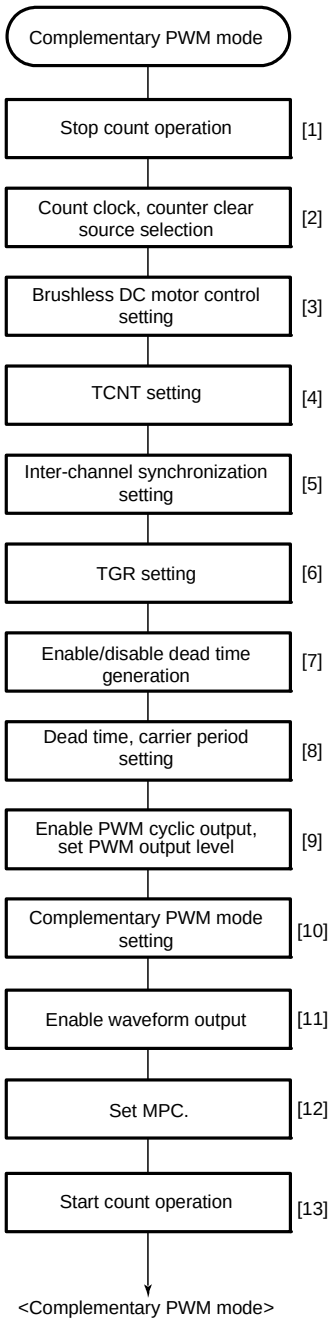


Figure 19.44 Block Diagram of MTU3 and MTU4 in Complementary PWM Mode

(1) Example of Complementary PWM Mode Setting Procedure

Figure 19.45 shows an example of the complementary PWM mode setting procedure.



[1] Set the TSTRA.CST3 and TSTRA.CST4 bits to 0 to stop the TCNT count operation. Specify complementary PWM mode while MTU3.TCNT and MTU4.TCNT are stopped.

[2] Set the TCR.TPSC[2:0] and TCR.CKEG[1:0] bits of MTU3 and MTU4 to select the same count clock source and clock edge for MTU3 and MTU4.
Set the TCR.CCLR[2:0] bits only when restarting by synchronous clearing with another channel during complementary PWM operation.

[3] To control a brushless DC motor, set the TGCRA.BDC bit and set the feedback signal input source and output chopping or gate signal direct output (only when MTU3 and MTU4 are used).

[4] Set the dead time in MTU3.TCNT. Set MTU4.TCNT to 0000h.

[5] Make this setting only when restarting by synchronous clearing with another channel during complementary PWM mode operation. In this case, set the TSYRA register to synchronize the channel generating the synchronous clear with MTU3 and MTU4.

[6] When MTU3 is used, set the PWM output duty ratio 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 the duty registers (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD) and buffer registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB).
Only when the double buffer function is used, set the buffer registers (MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF).

[7] This setting is necessary only when no dead time should be generated. Make appropriate settings in the TDERA register so that no dead time generated.

[8] Set the dead time in the TDDRA register, 1/2 the PWM period in the TCDRA register and TCBRA register, and 1/2 the PWM period plus the dead time in MTU3.TGRA and MTU3.TGRC.
When no dead time generation is selected, set 1 in the TDDRA register and 1/2 the PWM period + 1 in MTU3.TGRA and MTU3.TGRC.

[9] Set the PSYE bit in the TOCR1A register to enable or disable toggle output synchronized with the PWM period, and set the OLSP and OLSN bits to select the PWM output level.
When specifying the PWM output level using the TOCR2A and TOLBRA in buffer operation, refer to Timer Output Level Buffer Registers (TOLBRA).

[10] Select complementary PWM mode in MTU3.TMDR1. Specify the buffer operation (BFA = 1) of MTU3.TGRA and the buffer operation (BFB = 1) of MTU3.TGRB.
Only when the double buffer function is used, set the TMDR2A register to enable the double buffer function.

[11] Set the TOERA register to enable or disable output from the PWM output pin. Then, set the MTU3.TIORH, MTU3.TIORL, MTU4.TIORH, and MTU4.TIORL to 00h.

[12] Set the MPU and the port mode register (PMR) of the I/O ports.

[13] Set the TSTRA.CST3 and TSTRA.CST4 bits to 1 simultaneously to start count operation.

Figure 19.45 Example of Complementary PWM Mode Setting Procedure

(2) Outline of Complementary PWM Mode Operation

In complementary PWM mode, six phases (three positive and three negative) PWM waveforms can be output. Figure 19.46 illustrates counter operation in complementary PWM mode (MTU3 and MTU4), and Figure 19.47 shows an example of operation in complementary PWM mode.

(a) Counter Operation

In complementary PWM mode, three counters—MTU3.TCNT, MTU4.TCNT, and TCNTSA —in each unit perform up-/down-count operations.

MTU3.TCNT is automatically initialized to the value set in TDDRA when complementary PWM mode is selected and the CST3 bit in TSTRA is 0. When the CST3 bit is set to 1, MTU3.TCNT counts up to the value set in MTU3.TGRA, then switches to down-counting when it matches MTU3.TGRA. When the MTU4.TCNT value matches 0000h, MTU3.TCNT switches to up-counting, and the operation is repeated in this way.

MTU4.TCNT should be initialized to 0000h after a reset. When the CST4 bit is set to 1, MTU4.TCNT counts up in synchronization with MTU3.TCNT, and switches to down-counting when MTU3.TCNT matches MTU3.TGRA. On reaching 0000h, MTU4.TCNT switches to up-counting, and the operation is repeated in this way. TCNTSA is a read-only counter. It does not need to be initialized after a reset.

In counting up by MTU3.TCNT and MTU4.TCNT, MTU3.TCNT starts counting up when it matches TCDRA and switches to counting down when it matches MTU3.TGRA. Furthermore, when MTU4.TCNT matches TDDRA, TCNTSA is set to the value in MTU3.TGRA and counting is stopped.

When MTU4.TCNT matches TDDRA during down-counting of MTU3.TCNT and MTU4.TCNT, TCNTSA starts up-counting, and when MTU4.TCNT matches 0000h, the operation switches to down-counting. When MTU3.TCNT matches TCDRA, TCNTSA becomes 0000h and stops counting.

TCNTSA is compared with the compare register and temporary register, in which the PWM duty is specified, only during the count operation.

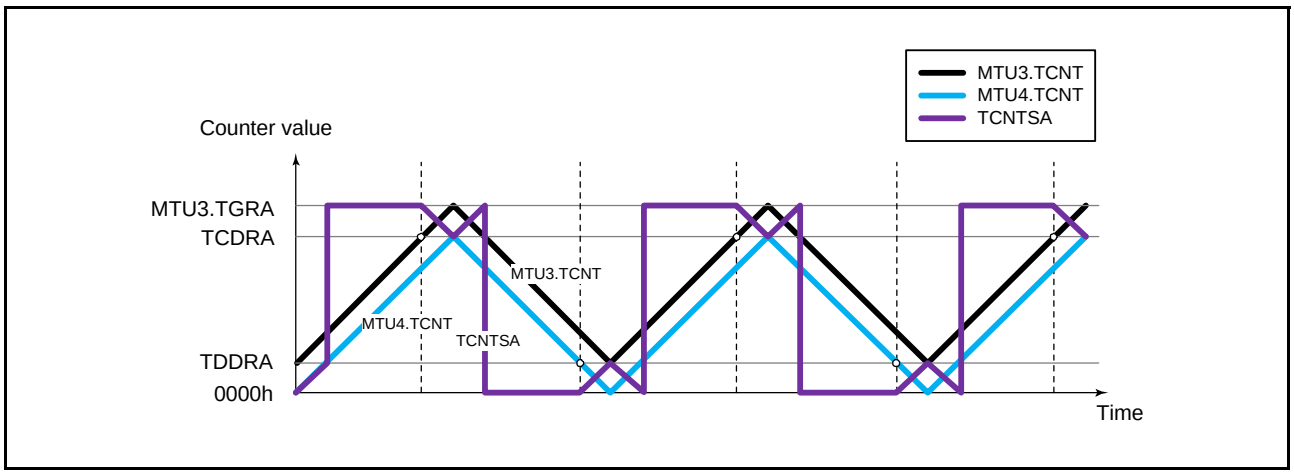


Figure 19.46 Count Operation in Complementary PWM Mode (MTU3 and MTU4)

(b) Register Operation

In complementary PWM mode, nine registers (compare registers, buffer registers, and temporary registers) are used to control the duty ratio for the PWM output. Figure 19.47 shows an example of operation in complementary PWM mode (MTU3 and MTU4).

MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB are constantly 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 register (TOCR1A) is output from the PWM output pin.

MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD are buffer registers for these compare registers.

When the double buffer function is used, MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF are also used as buffer registers B. For details of double buffer operation, refer to section 19.3.8 (2) (r), 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 last and enable data transfer from the buffer register to a temporary register. At this time, transfer from the TCBRA register and MTU3.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 at the end of the Tb1 interval (when matches MTU3.TGRA while TCNTSA is counting up), or at the end of the Tb2 interval (when matches 0000h while TCNTSA is 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 19.47 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 19.47), 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 and two registers (compare register and temporary register) are compared, and PWM output is controlled accordingly.

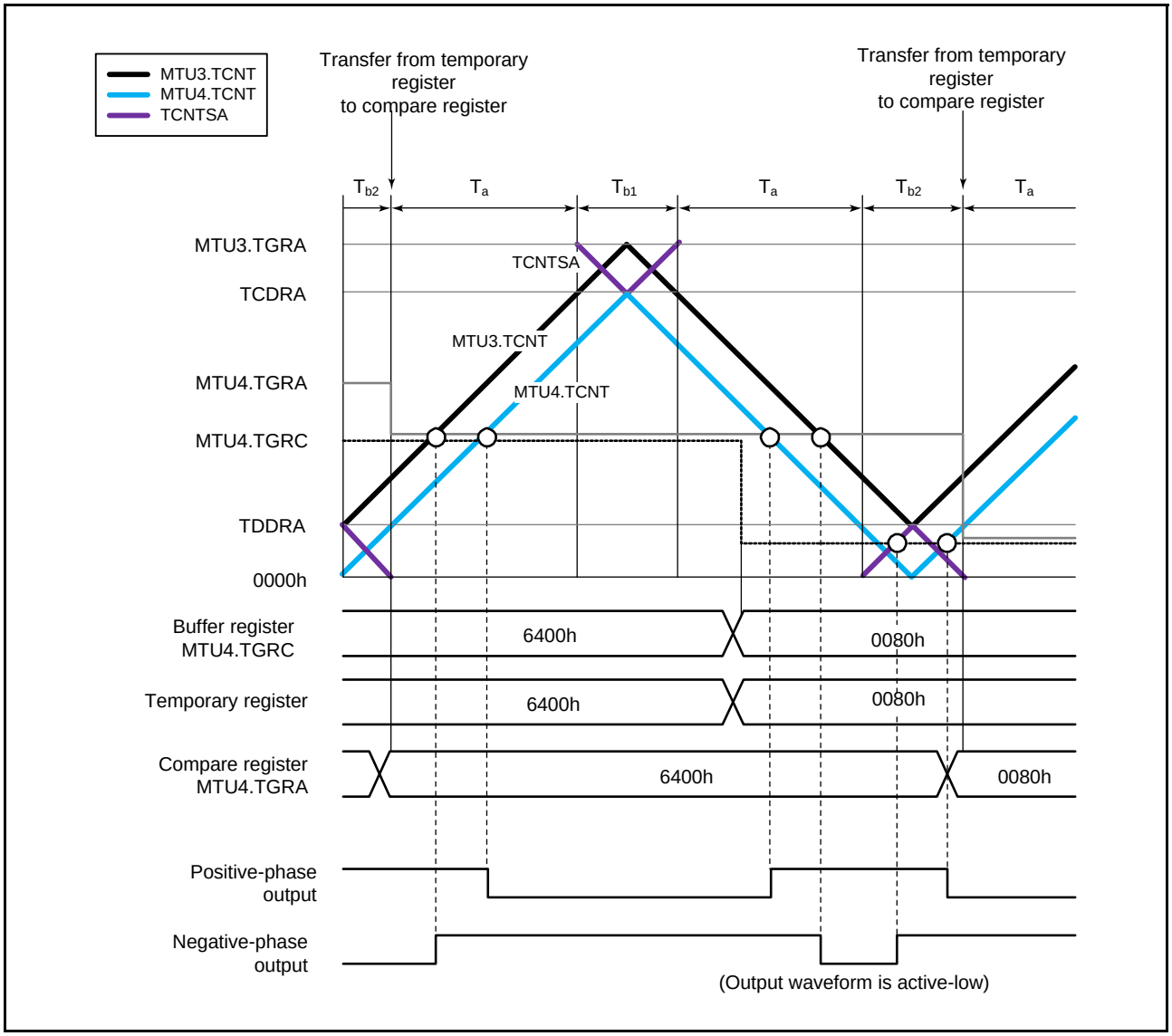


Figure 19.47 Example of Operation in Complementary PWM Mode (MTU3 and MTU4)

(c) 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] bits, initial values should be set in the following registers.
The TOCR1A and TOCR2A registers are used to set the PWM output level. MTU3.TGRC operates as the buffer register for MTU3.TGRA, and should be set with 1/2 the PWM period + dead time Td. The timer period buffer register (TCBRA) operates as the buffer register for the timer period data register (TCDRA), and should be set with 1/2 the PWM period. Set dead time Td in the timer dead time data register (TDDRA).
When dead time is not needed, the TDER bit in the timer dead time enable register (TDERA) should be set to 0, MTU3.TGRC and MTU3.TGRA should be set to 1/2 the PWM carrier period + 1, and TDDRA should be set to 1.
Set the respective initial PWM duty values in three buffer registers A (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD). Set three buffer registers B (MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF) only when the double buffer function is used.
The values set in the five buffer registers excluding TDDRA are transferred to the corresponding compare registers as soon as complementary PWM mode is set.
Set MTU4.TCNT to 0000h before setting complementary PWM mode.

Table 19.61 Registers and Counters Requiring Initial Setting

Register and Counter	Setting
TOCR1A, TOCR2A	PWM output level
MTU3.TGRC	1/2 PWM period + dead time Td (1/2 PWM period + 1 when dead time generation is disabled by TDERA)
TDDRA	Dead time Td (1 when dead time generation is disabled by TDERA)
TCBRA	1/2 PWM period
MTU3.TGRD, MTU4.TGRC, MTU4.TGRD	Initial PWM duty ratio value for each phase
MTU3.TGRE, MTU4.TGRE, MTU4.TGRF	Initial PWM duty ratio value for each phase (only when double buffer function is used)
MTU4.TCNT	0000h

Note: The value set in MTU3.TGRC should be the sum of 1/2 the PWM period set in TCBRA and dead time Td set in TDDRA. When dead time generation is disabled by TDERA, TGRC should be set to 1/2 the PWM period + 1.

(d) 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 bits OLS1P to OLS3P and OLS1N to OLS3N in timer output control register 2 (TOCR2A).
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.

(e) 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). The value set in TDDRA is used as the MTU3.TCNT counter start value and creates a non-overlapping interval between MTU3.TCNT and MTU4.TCNT. Complementary PWM mode should be cleared before changing the contents of TDDRA.

(f) Dead Time Suppressing

Dead time generation is suppressed by setting the TDER bit in the timer dead time enable register (TDERA) to 0. TDERA can be set to 0 only when 0 is written to it after reading TDER = 1.
MTU3.TGRA and MTU3.TGRC should be set to 1/2 PWM period + 1 and the timer dead time data register (TDDRA) should be set to 1.
By the above settings, PWM waveforms without dead time can be obtained. Figure 19.48 shows an example of operation without dead time (MTU3 and MTU4).

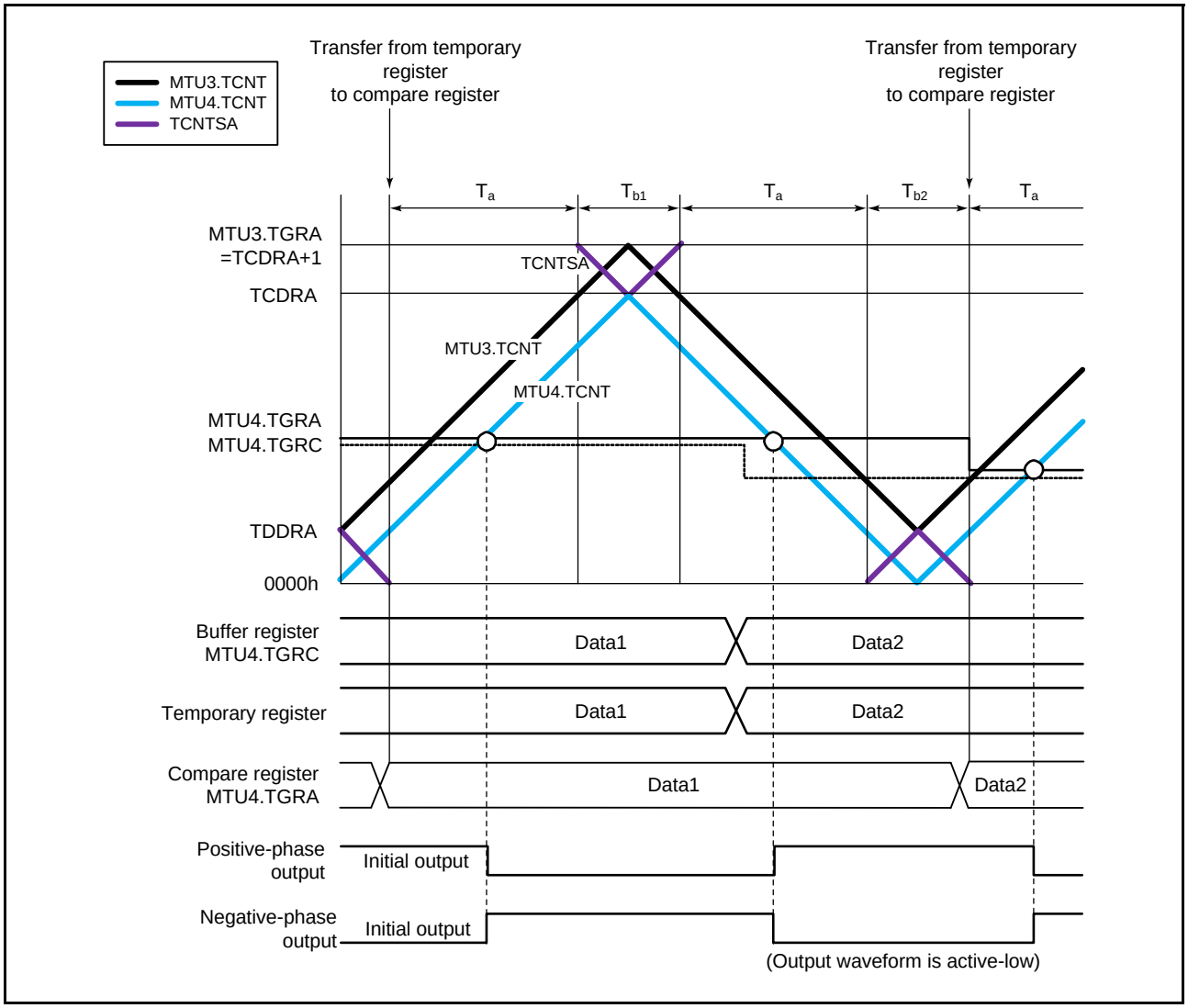


Figure 19.48 Example of Operation without Dead Time (MTU3 and MTU4)

(g) PWM Period Setting

In complementary PWM mode, the PWM period is set in two registers—MTU3.TGRA, in which the MTU3.TCNT upper limit value is set, and TCDRA, in which the MTU4.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 setting = TCDRA setting + TDDRA setting
Without dead time: MTU3.TGRA setting = TCDRA setting + 1

In addition, the settings should be made so as to achieve the following relationship between the TCDRA register and the TDDRA register:

TCDRA setting > TDDRA setting × 2 + 2

The MTU3.TGRA and TCDRA settings are made by setting values in buffer registers MTU3.TGRC and TCBRA. When data is written to MTU4.TGRD to enable transfers, the values set in MTU3.TGRC and TCBRA are transferred simultaneously to the MTU3.TGRA and TCDRA with the transfer timing selected with the MTU3.TMDR1.MD[3:0] bits.
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 19.49 illustrates the operation when the PWM period is updated at the crest.
Refer to the following section, (h), Register Data Updating, for the method of updating the data in each buffer register.

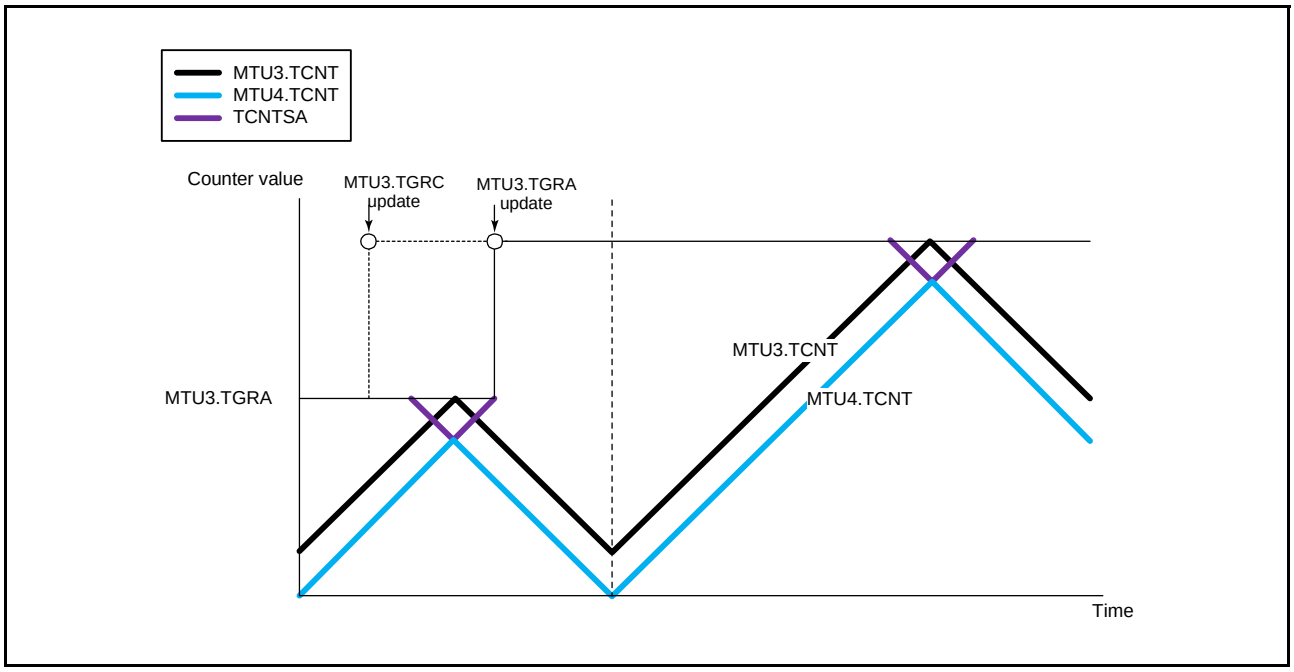


Figure 19.49 Example of PWM Period Updating (MTU3 and MTU4)

(h) Register Data Updating

The buffer registers are used to update the data in five compare registers for the PWM duty and PWM period in complementary PWM mode. The update data can be written to the buffer register at any time.

There is a temporary register between each of these registers and its buffer register. While subcounter TCNTSA 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 is counting; in this case, the value written to a buffer register is transferred after TCNTSA halts.

The temporary register value is transferred to the compare register at the data update timing set with MTU3.TMDR1.MD[3:0] bits. Figure 19.50 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 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.

Even when not updating all five registers or when not updating the MTU4.TGRD data, be sure to write to MTU4.TGRD after writing data to the registers to be updated. In this case, the data written to MTU4.TGRD should be the same as the data prior to the write operation.

Refer to section 19.3.8 (2) (r), Double Buffer Function in Complementary PWM Mode, for data updating when the double buffer function is used.

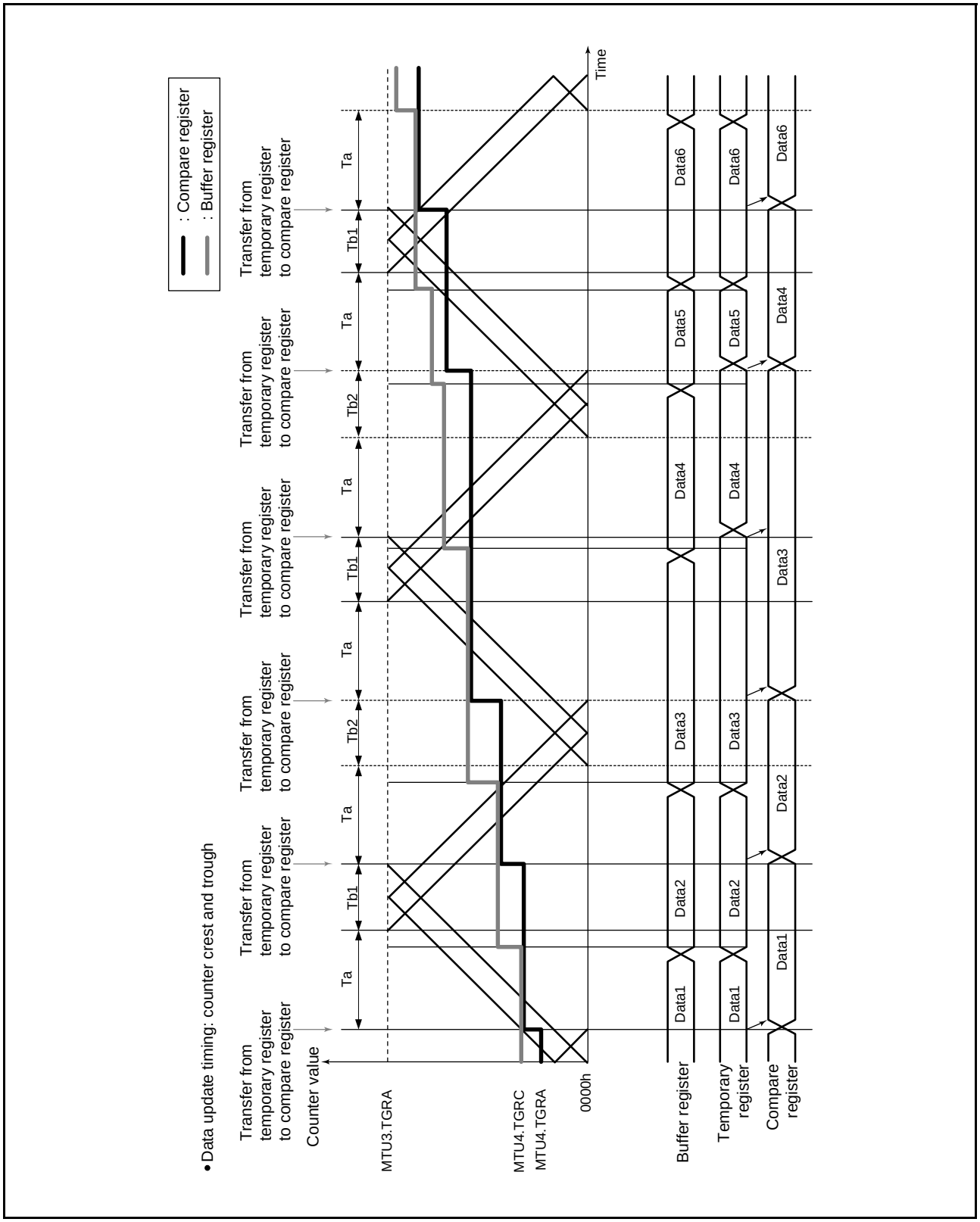


Figure 19.50 Example of Data Updating in Complementary PWM Mode (MTU3 and MTU4)

(i) Initial Output in Complementary PWM Mode

In complementary PWM mode, the initial output is determined by the setting of the OLSN and OLSP bits in the TOCR1A register or the OLS1N to OLS3N and OLS1P to OLS3P bits in the TOCR2A register.
This initial output is the non-active level of the PWM output and continues from when complementary PWM mode is set with the MTU3.TMDR1 until MTU4.TCNT exceeds the value set in the TDDRA register. Figure 19.51 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 19.52.

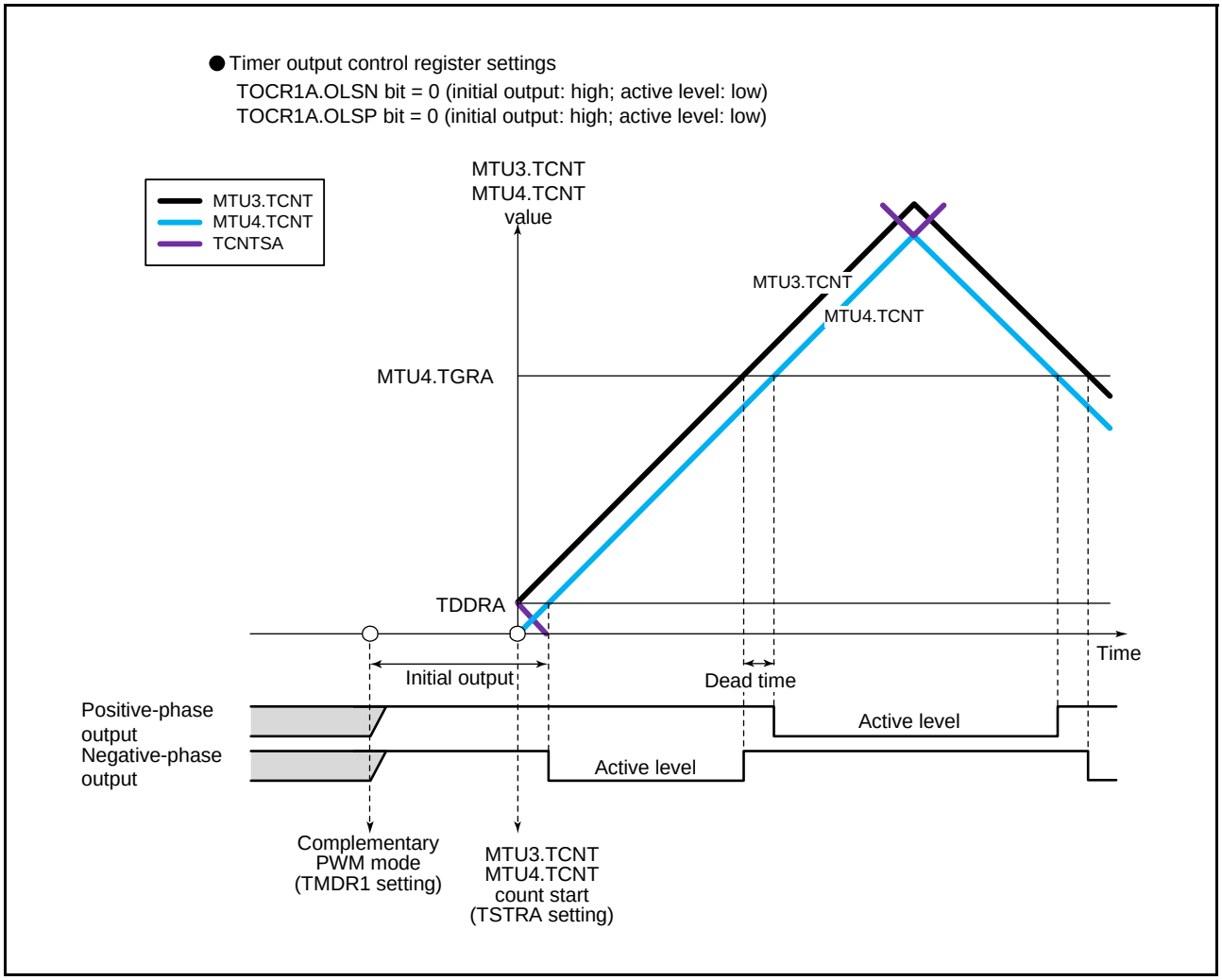


Figure 19.51 Example of Initial Output in Complementary PWM Mode (MTU3 and MTU4) (1))

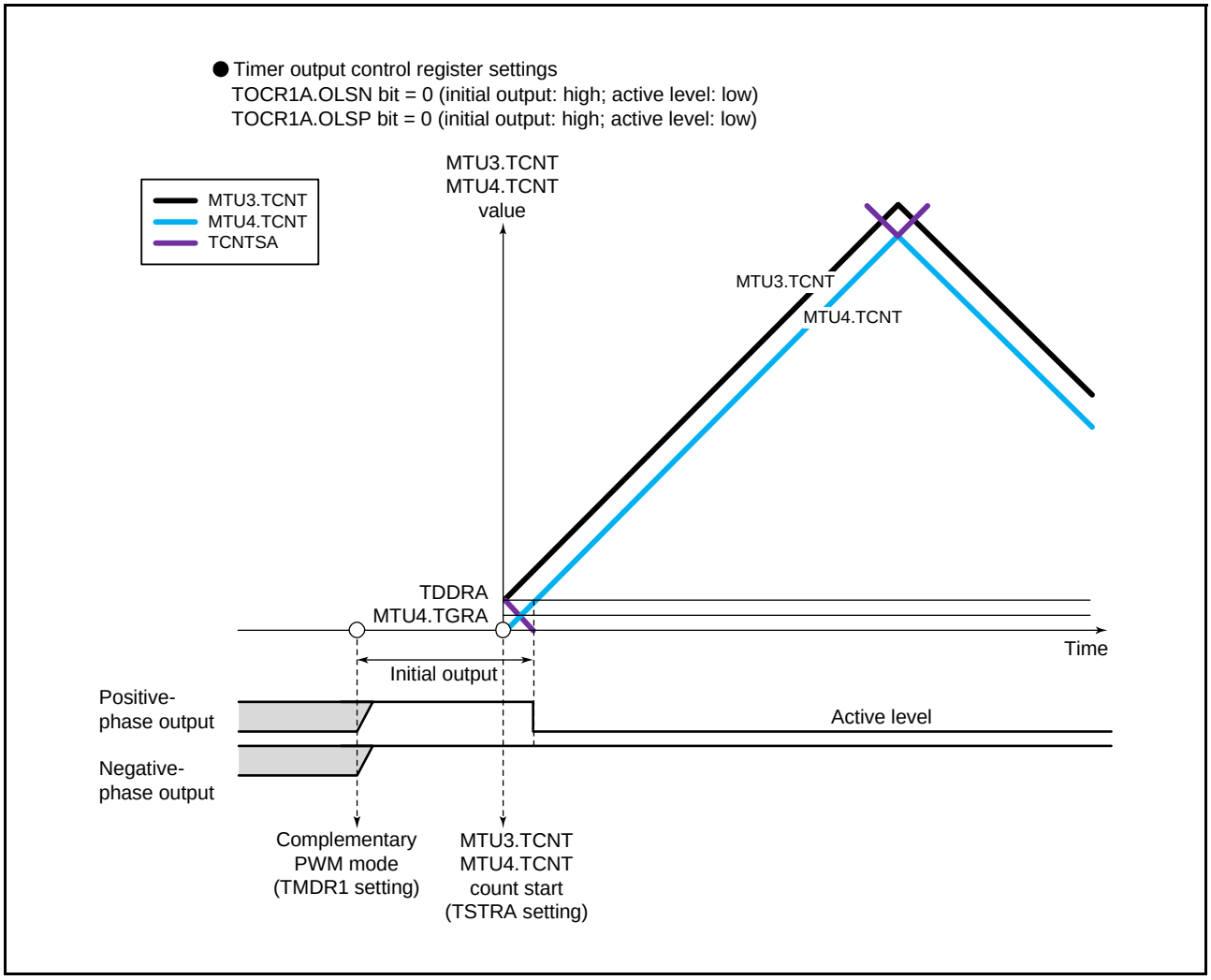


Figure 19.52 Example of Initial Output in Complementary PWM Mode (MTU3 and MTU4) (2)

(j) Method for Generating PWM Output in Complementary PWM Mode

In complementary PWM mode, six phases (three positive and three negative) 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 is counting, the compare register and temporary register values are simultaneously compared to create consecutive PWM output 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 19.53 to Figure 19.55 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 19.53.

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 19.54, 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 19.55, turning off the negative phase has priority due to the occurrence of compare match a' (negative-phase off timing) before compare match d (negative-phase on timing). 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.

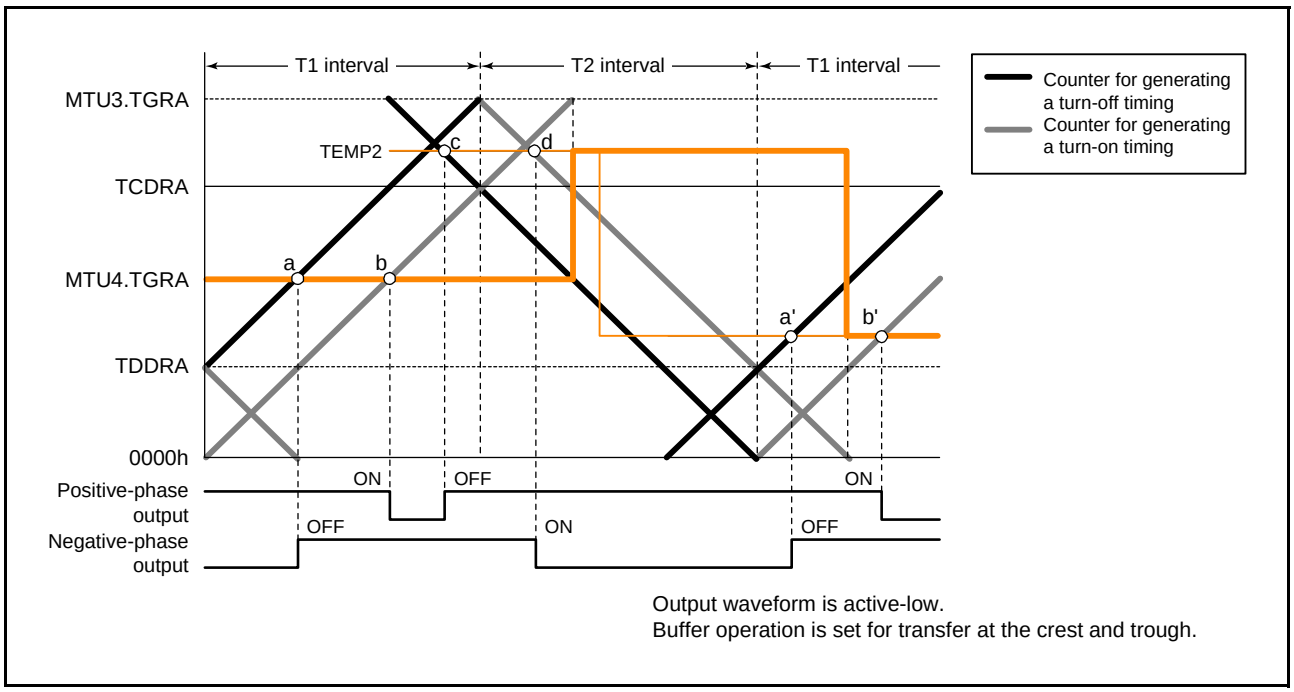


Figure 19.53 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (1)

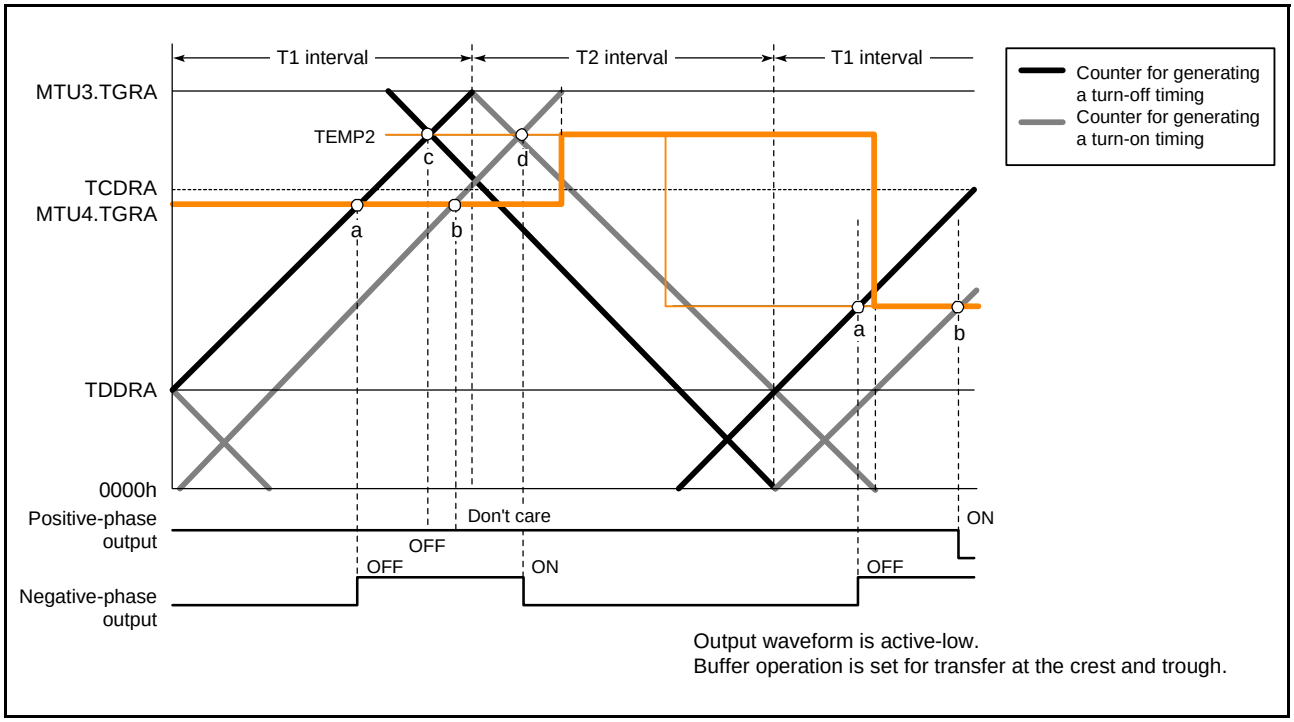


Figure 19.54 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (2)

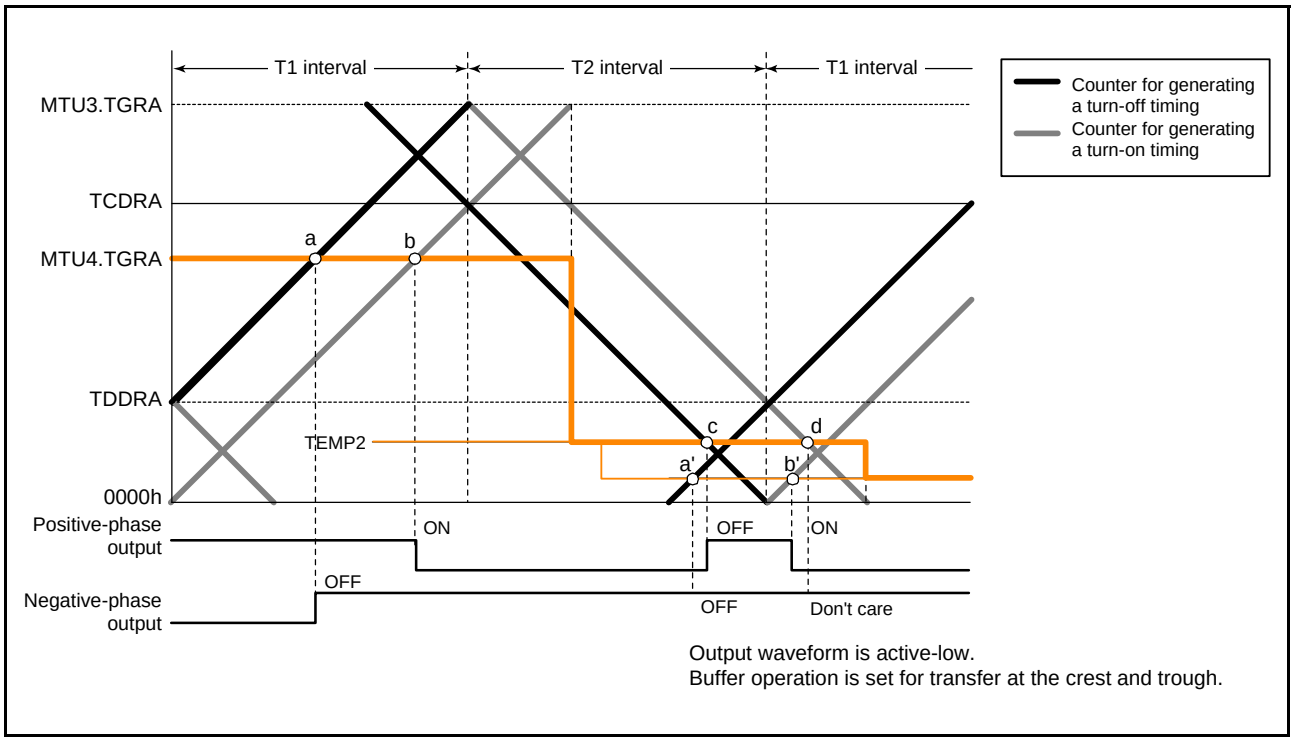


Figure 19.55 Example of Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (3)

(k) 0% and 100% Duty Ratio Output in Complementary PWM Mode

In complementary PWM mode, 0% and 100% duty PWM output can be output as required. Figure 19.56 to Figure 19.60 show output examples.

A 100% duty waveform is output when the compare register value is set to 0000h. 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. 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.

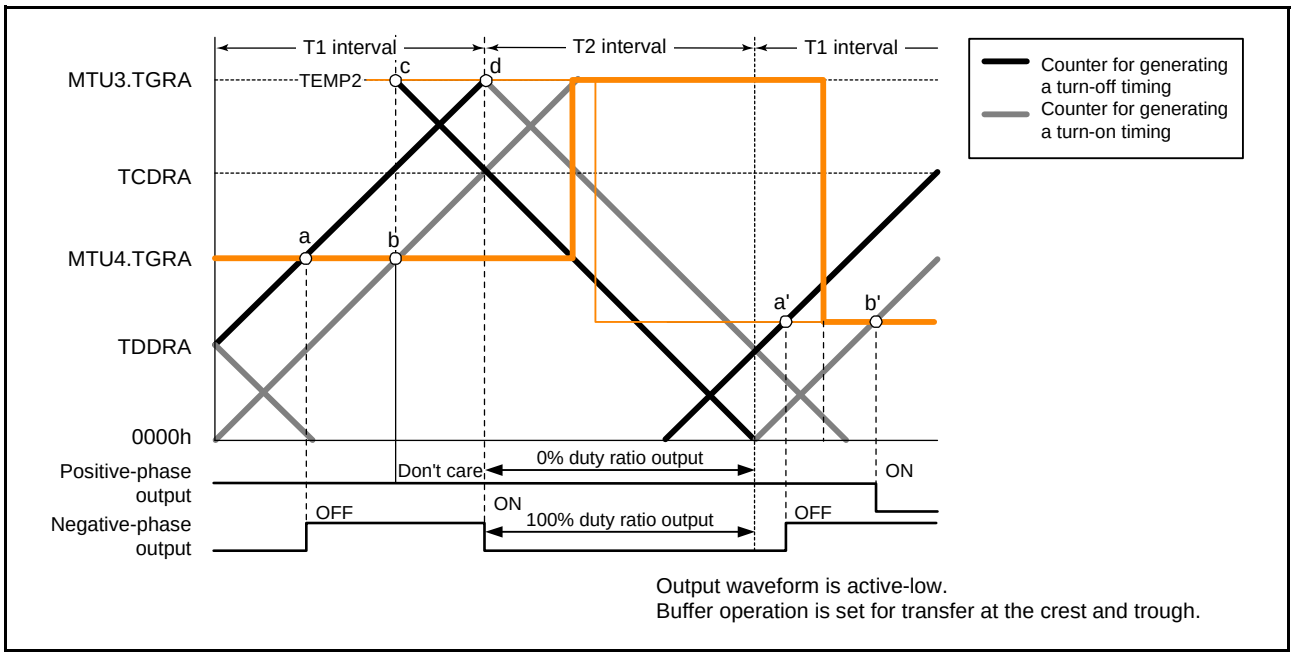


Figure 19.56 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (1)

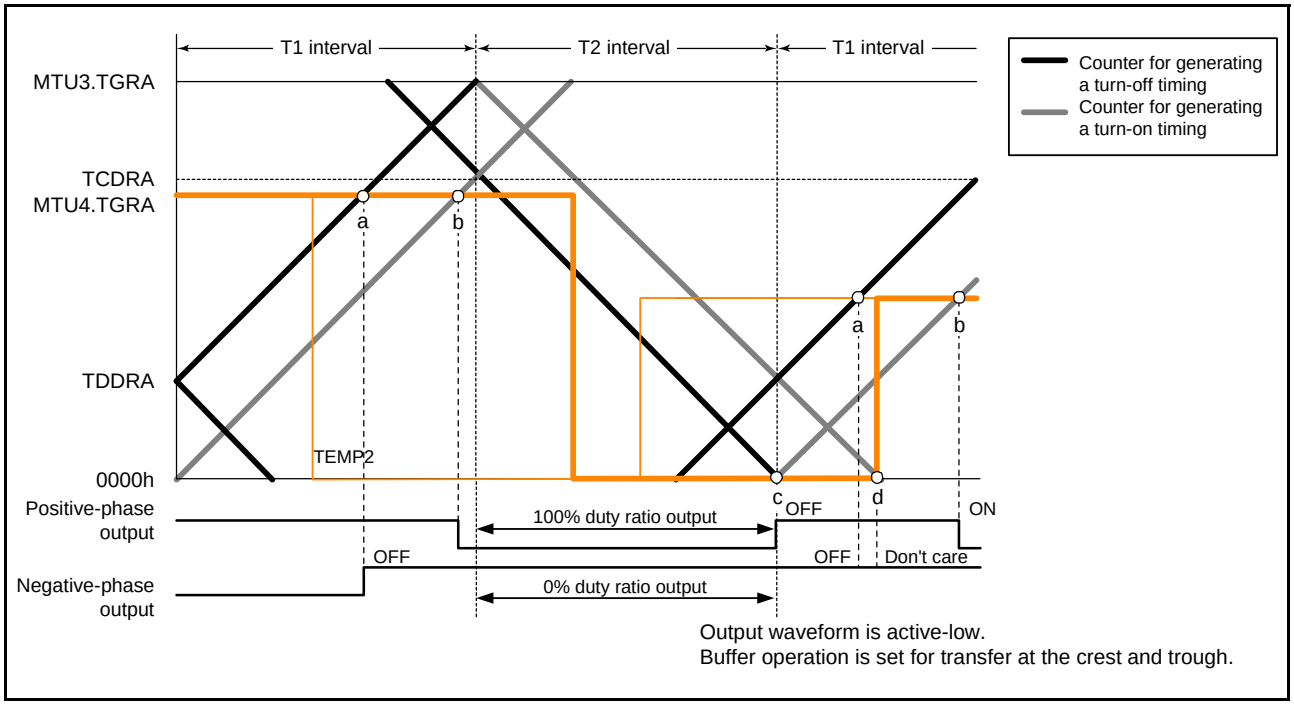


Figure 19.57 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (2)

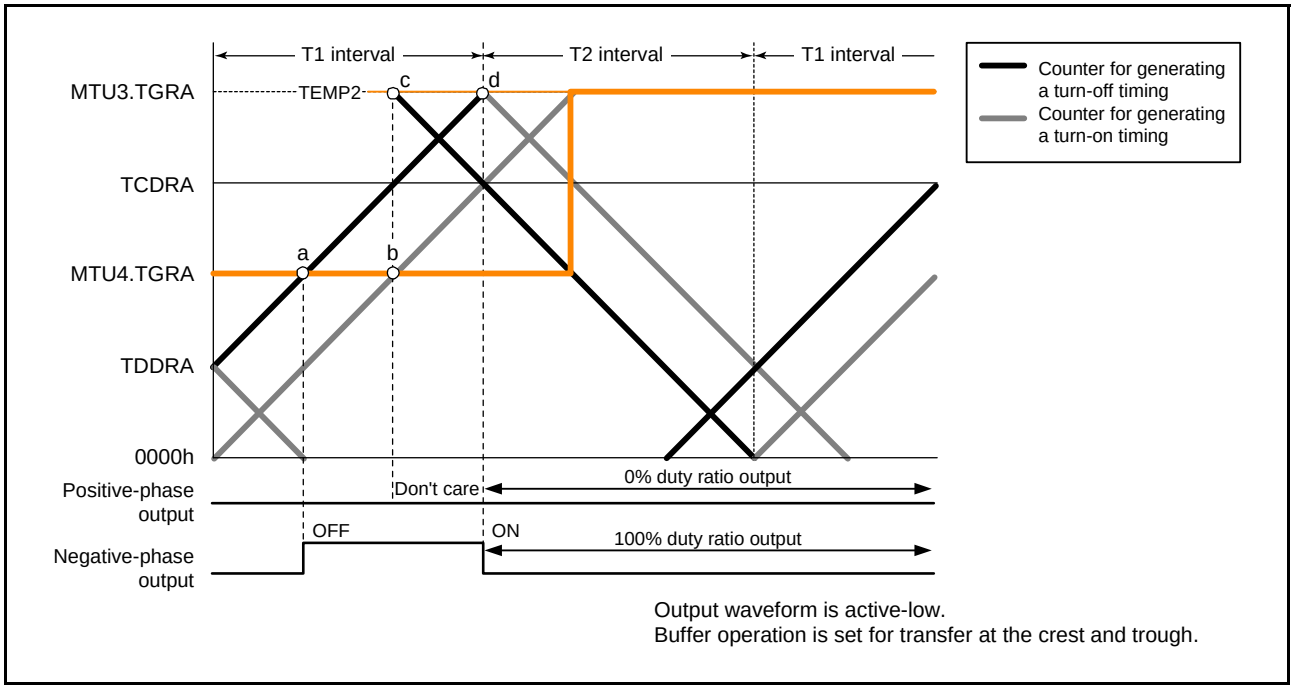


Figure 19.58 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (3)

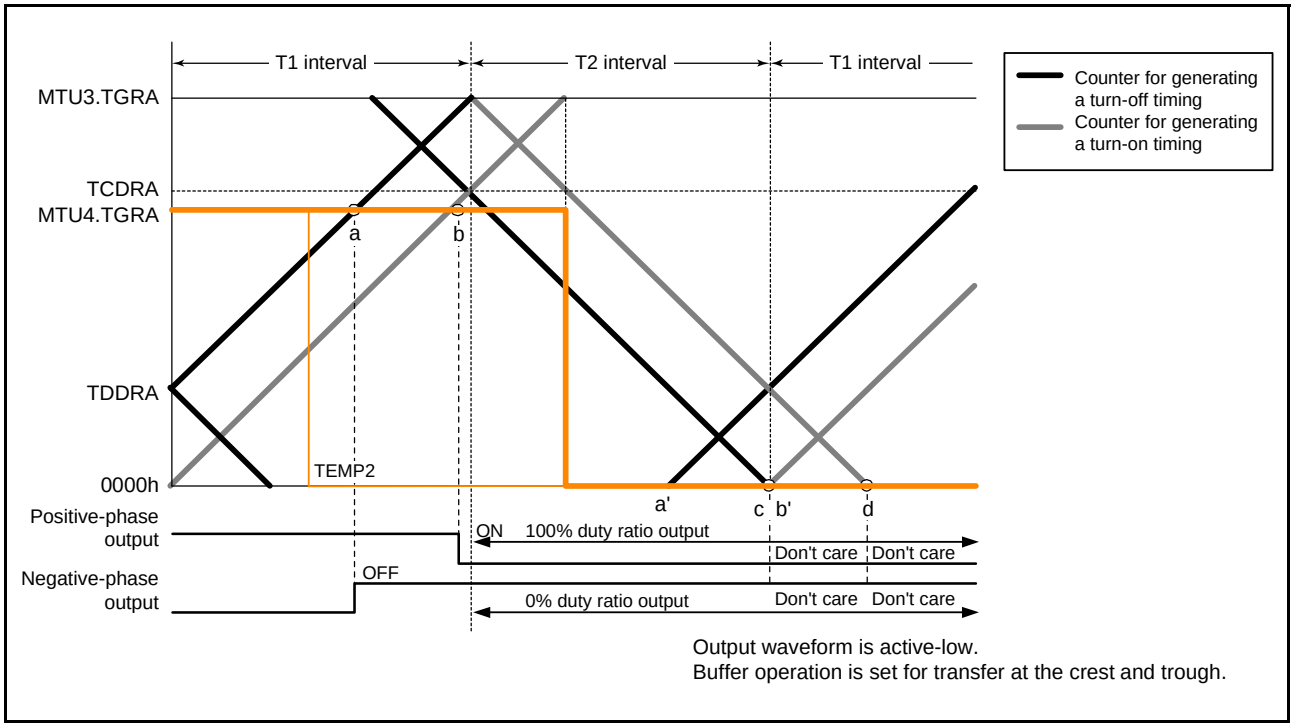


Figure 19.59 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (4)

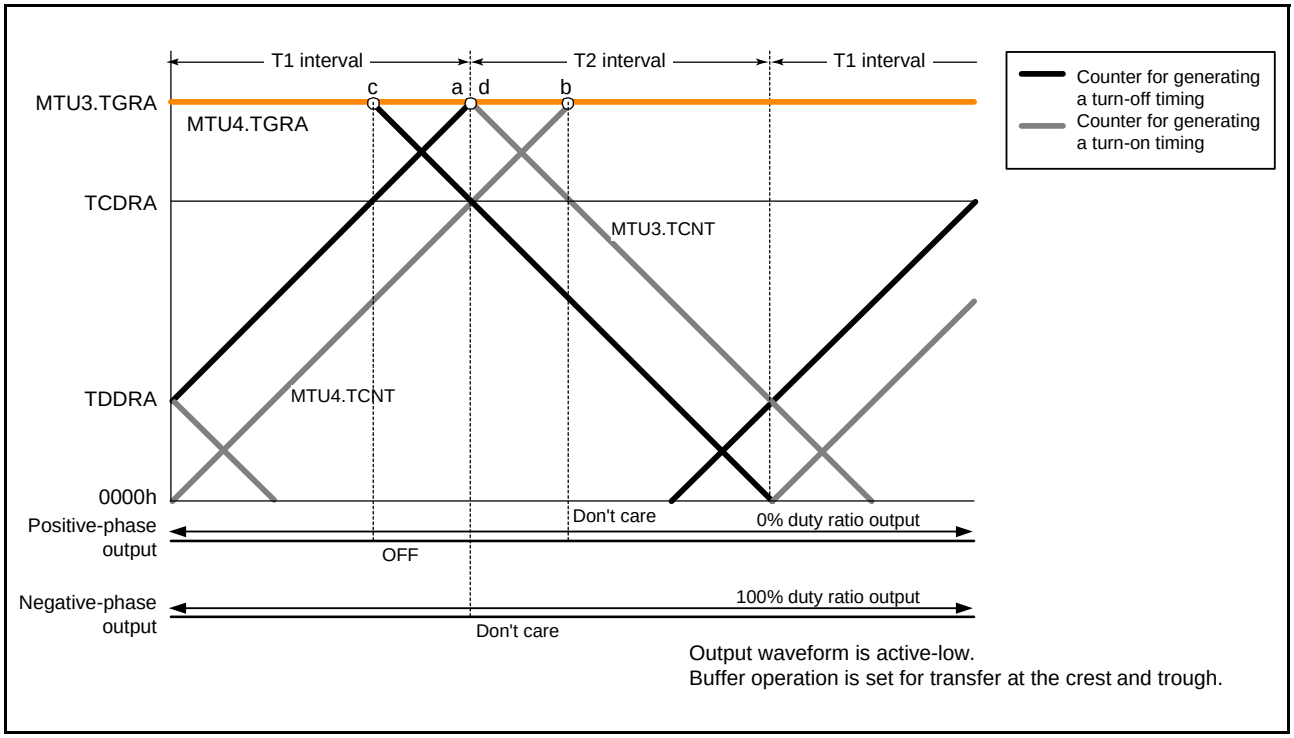


Figure 19.60 Example of 0% and 100% Waveform Output in Complementary PWM Mode (MTU3 and MTU4) (5)

(l) Toggle Output Synchronized with PWM Period

In complementary PWM mode, toggle output from the PWM output pin in synchronization with the PWM period can be enabled by setting the PSYE bit in the TOCR1A register to 1. An example of a toggle output waveform is shown in Figure 19.61.

This output is toggled by a compare match between MTU3.TCNT and MTU3.TGRAand a compare match between MTU4.TCNT and 0000h.

pin is assigned for this toggle output. The initial output is high-level output.

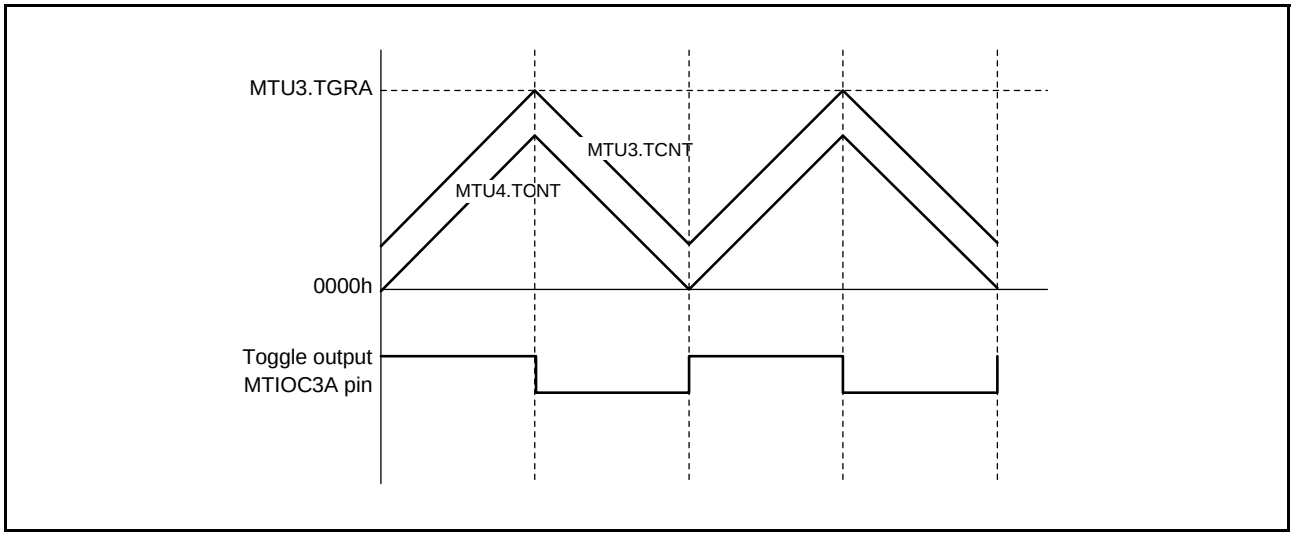


Figure 19.61 Example of Toggle Output Waveform Synchronized with PWM Output (MTU3 and MTU4)

(m) Counter Clearing by Another Channel

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTSA can be cleared by another channel source when a mode for synchronization with another channel is specified through the TSYRA register and synchronous clearing is selected with MTU3.TCR.CCLR[2:0] bits.

Figure 19.62 illustrates an example of this operation.

Use of this function enables a counter to be cleared and restarted through an external signal.

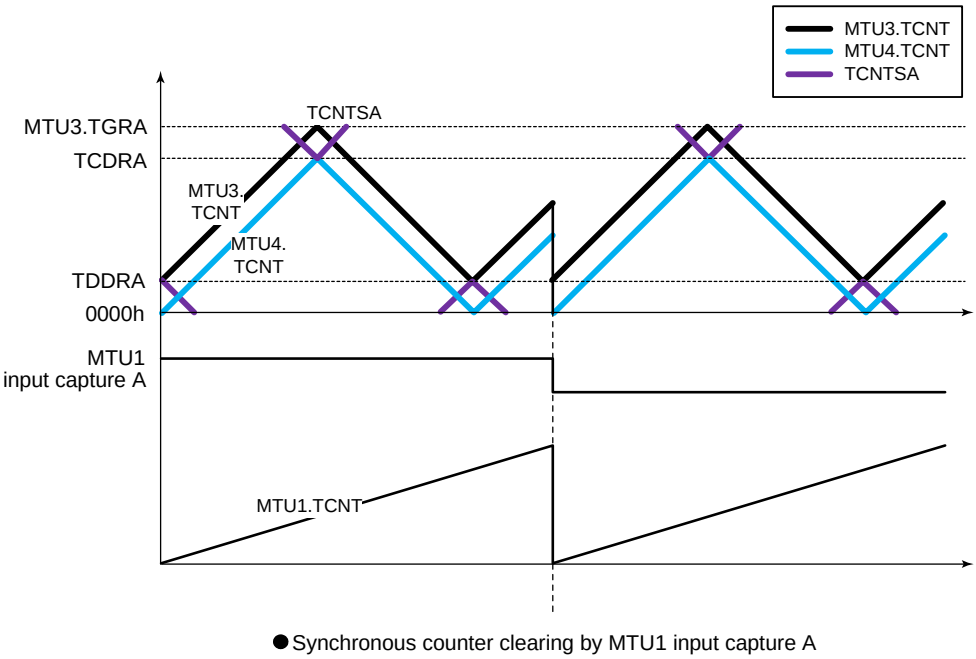


Figure 19.62 Counter Clearing Synchronized with Another Channel (MTU3 and MTU4)

(n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the WRE bit in TWCRA to 1 suppresses initial output when synchronous counter clearing occurs in the Tb interval (Tb2 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 Tb2 interval as indicated by (10) or (11) in Figure 19.63. When synchronous clearing occurs outside that interval, the initial value specified by the OLSN and OLSP bits in TOCR1A is output. Even in the Tb2 interval, if synchronous clearing occurs in the initial value output period (indicated by (1) in Figure 19.63) 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. In MTU3 and MTU4, synchronous clearing in any of MTU0 to MTU2 can cause counter clearing.

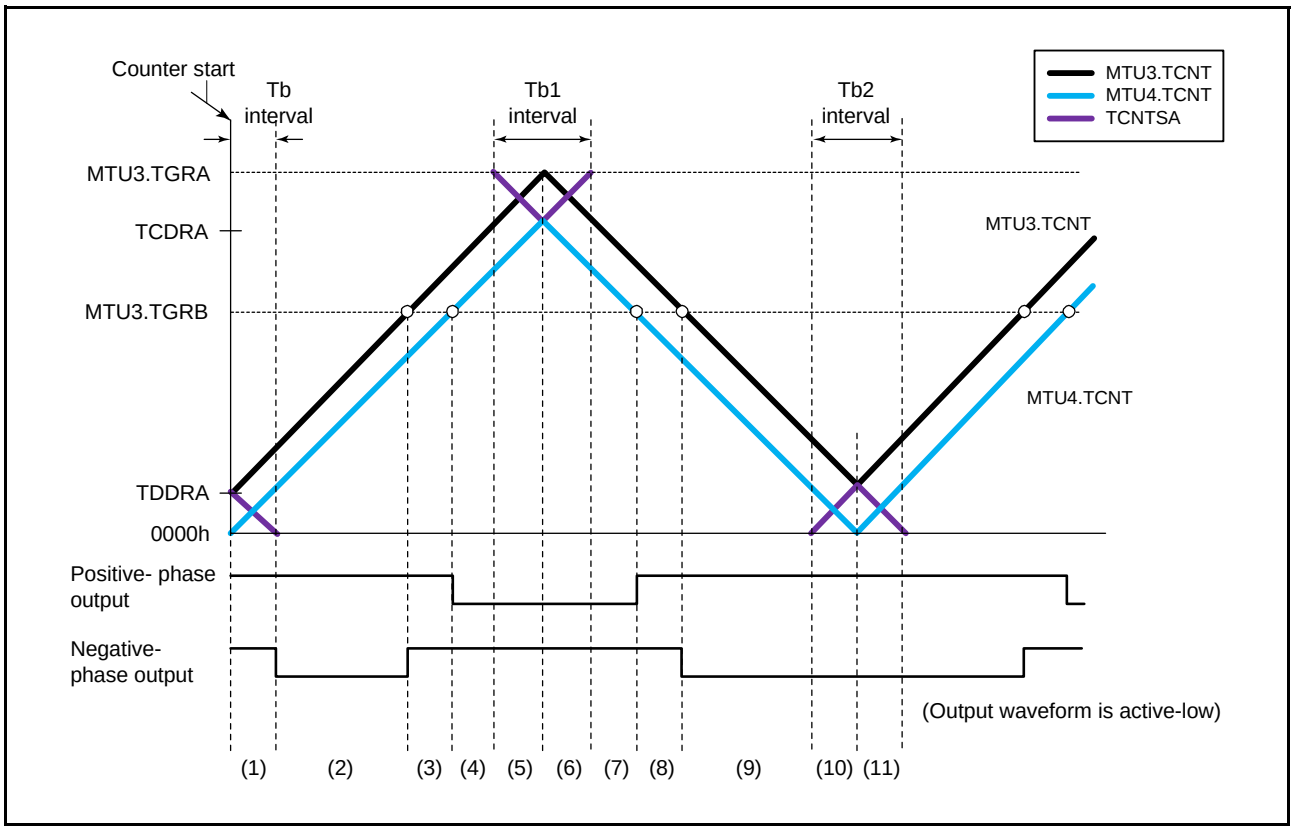


Figure 19.63 Timing for Synchronous Counter Clearing (MTU3 and MTU4)

- 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 19.64.

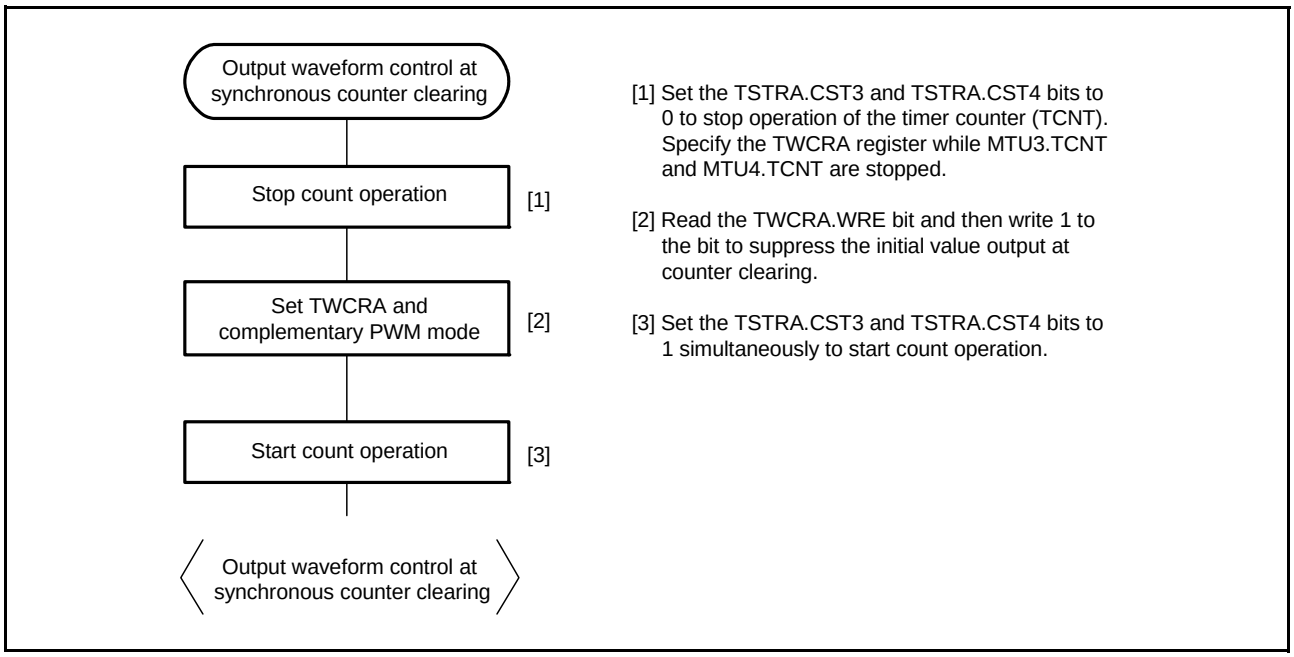


Figure 19.64 Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode (MTU3 and MTU4)

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Figure 19.65 to Figure 19.68 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 19.65 to Figure 19.68, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 19.63, respectively.

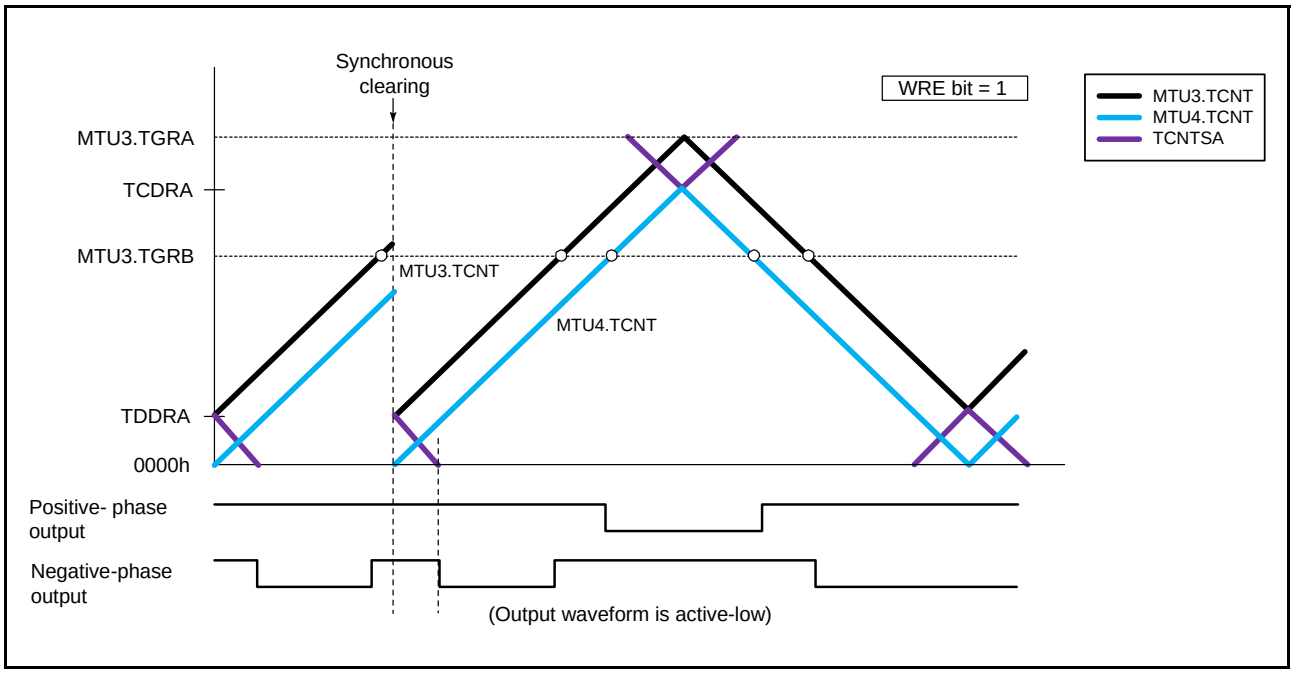


Figure 19.65 Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 19.63; TWCRA.WRE Bit is 1)

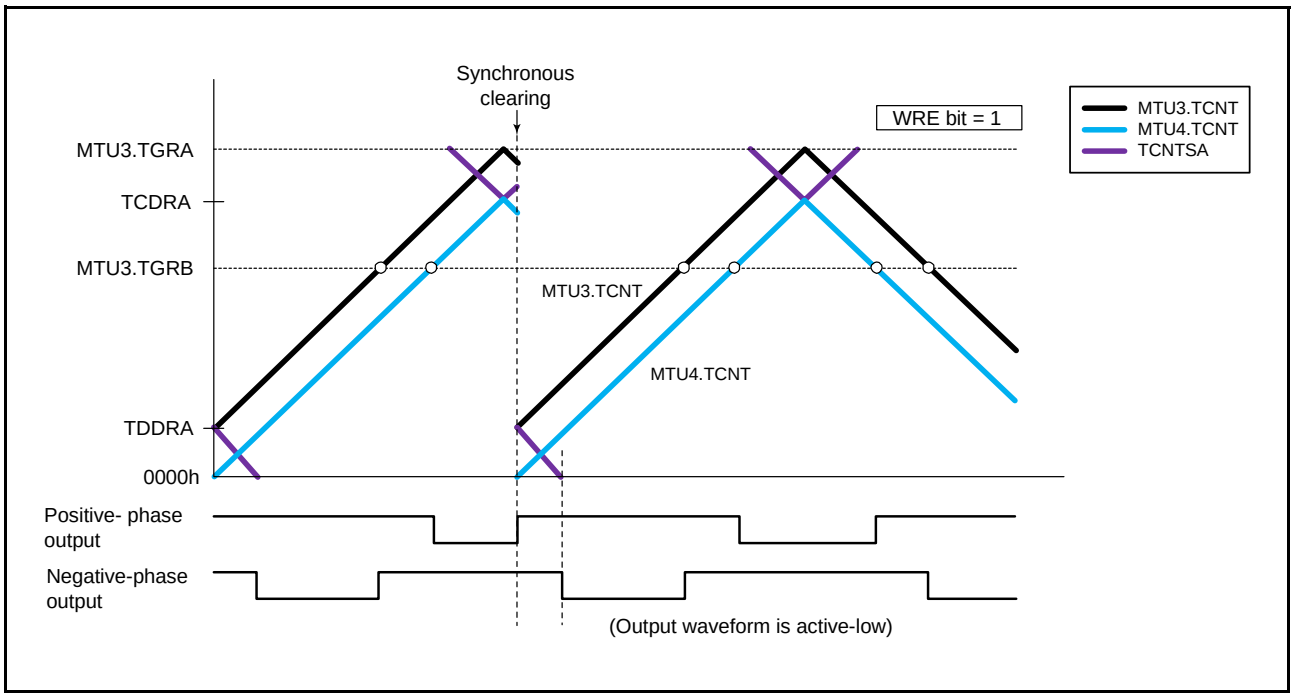


Figure 19.66 Example of Synchronous Clearing in Tb1 interval (Timing (6) in Figure 19.63; TWCRA.WRE Bit is 1)

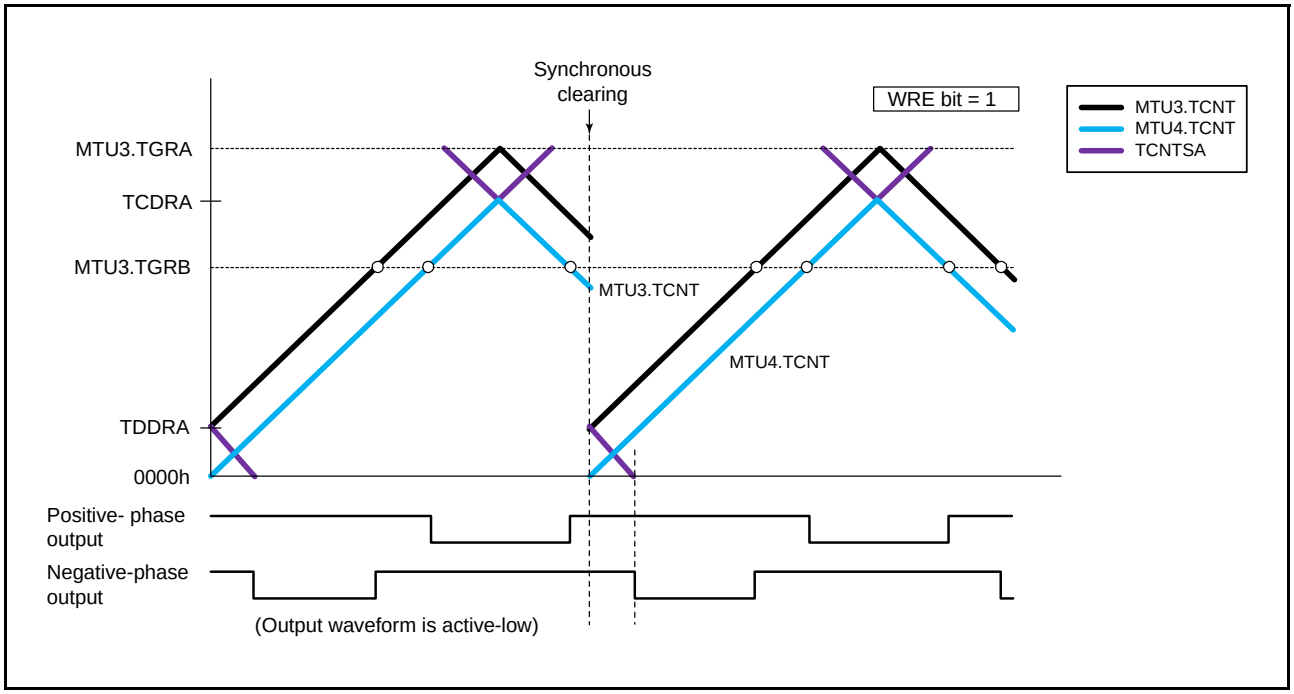


Figure 19.67 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 19.63; TWCRA.WRE Bit is 1)

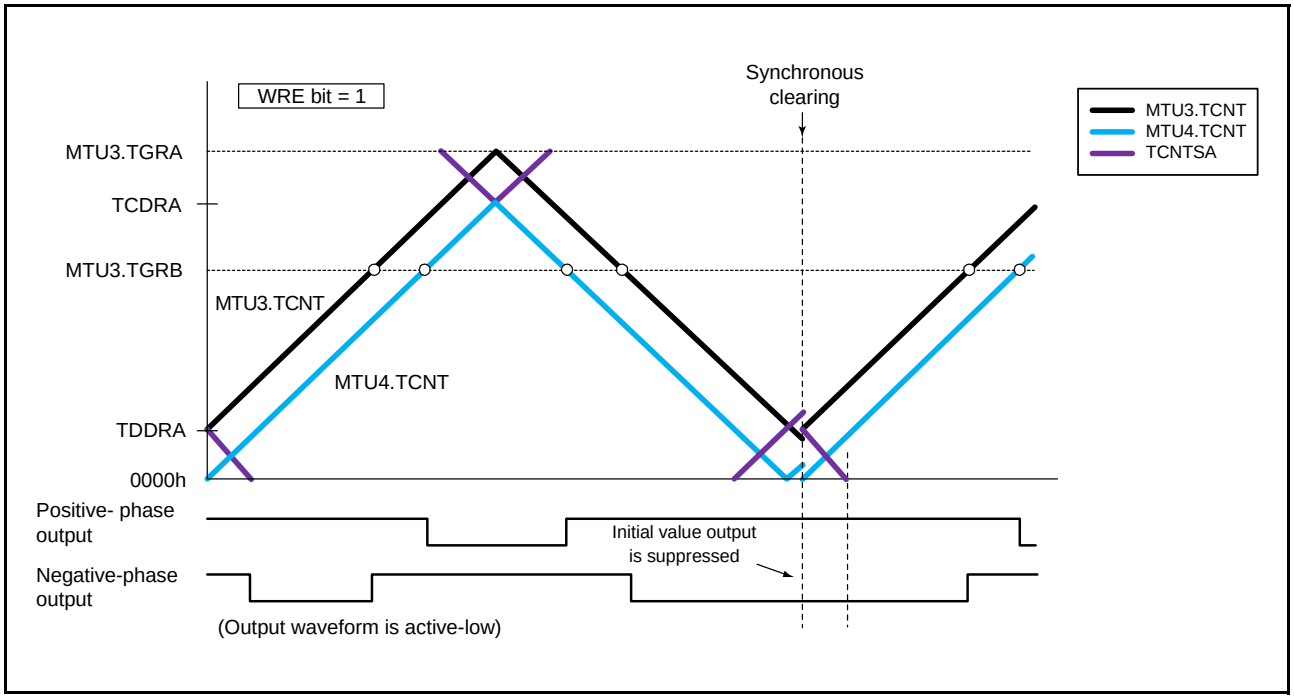


Figure 19.68 Example of Synchronous Clearing in Tb2 interval (Timing (11) in Figure 19.63; TWCRA.WRE Bit is 1)

(o) Counter Clearing by MTU3.TGRA Compare Match

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTSA can be cleared by MTU3.TGRA compare match when the TWCRA.CCE bit.
Figure 19.69 illustrates an operation example.

Note 1. Use this function only in complementary PWM mode 1 (transfer at crest).
Note 2. Do not specify synchronous clearing by another channel (do not set 1 in the SYNC0 to SYNC4 bits in the timer synchronous register (TSYRA)).
Note 3. Do not set the PWM duty value to 0000h.
Note 4. Do not set the PSYE bit in timer output control register 1 (TOCR1A) to 1.

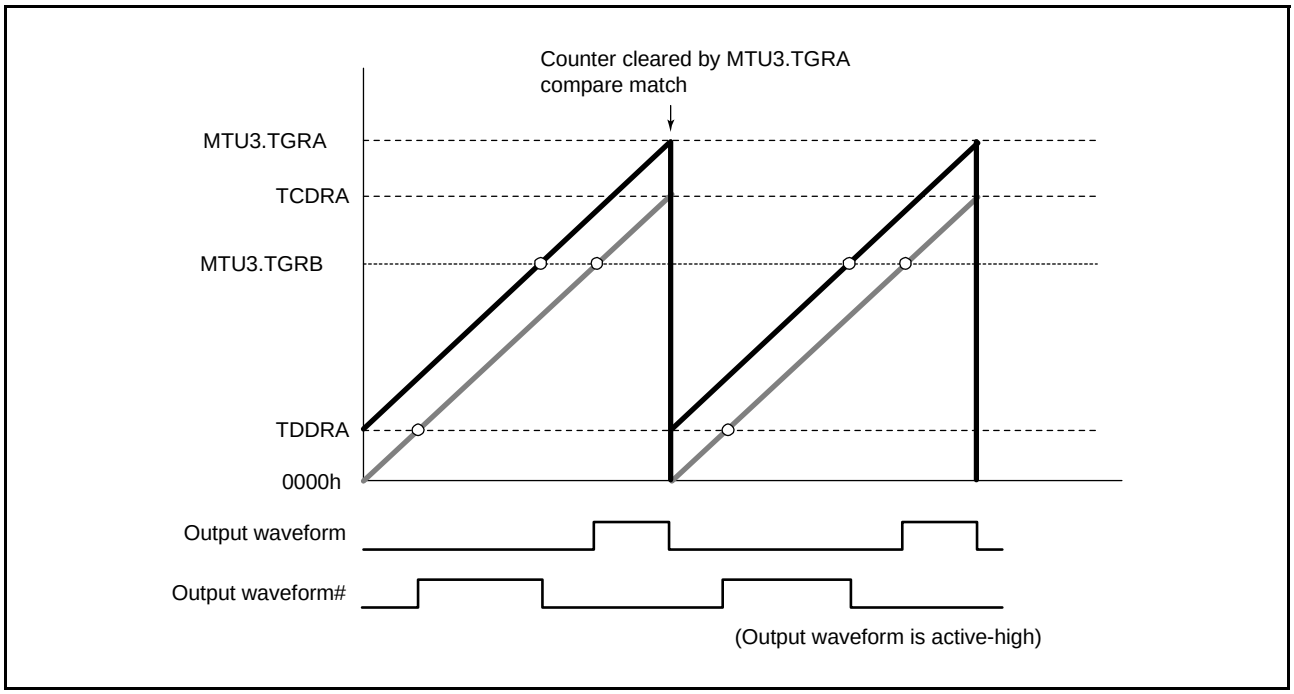


Figure 19.69 Example of Counter Clearing Operation by MTU3.TGRA Compare Match

(p) Example of Waveform Output for Driving AC Synchronous Motor (Brushless DC Motor)

In complementary PWM mode when MTU3 and MTU4 are used, a brushless DC motor can easily be controlled using the TGCRA register. Figure 19.70 to Figure 19.73 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 (make appropriate settings with the MPC and port mode registers (PMR) 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 PWM 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.

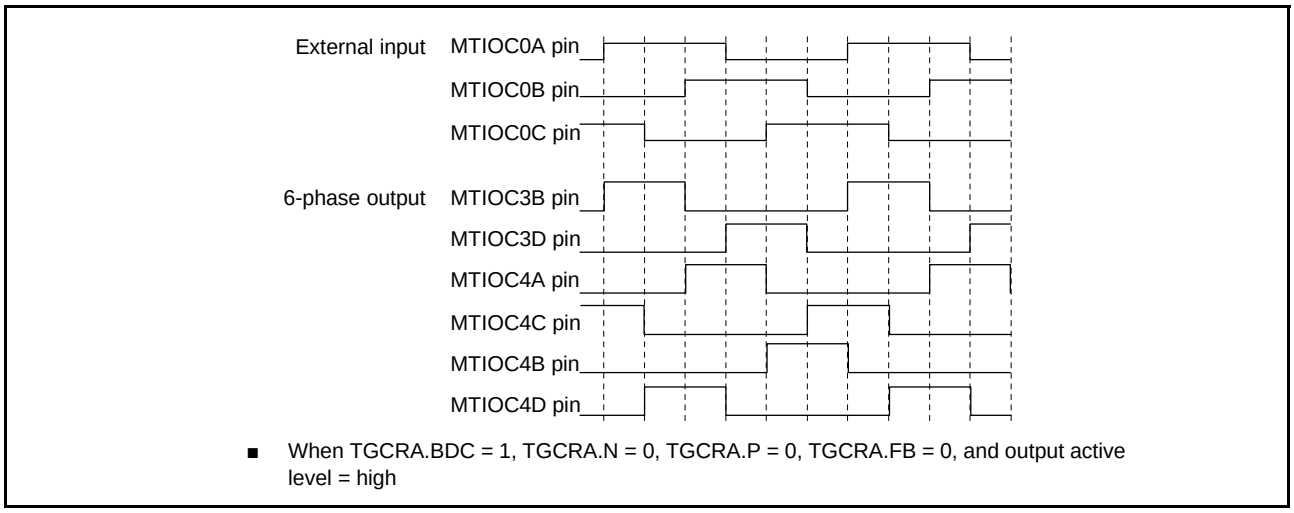


Figure 19.70 Example of Output Phase Switching by External Input (1)

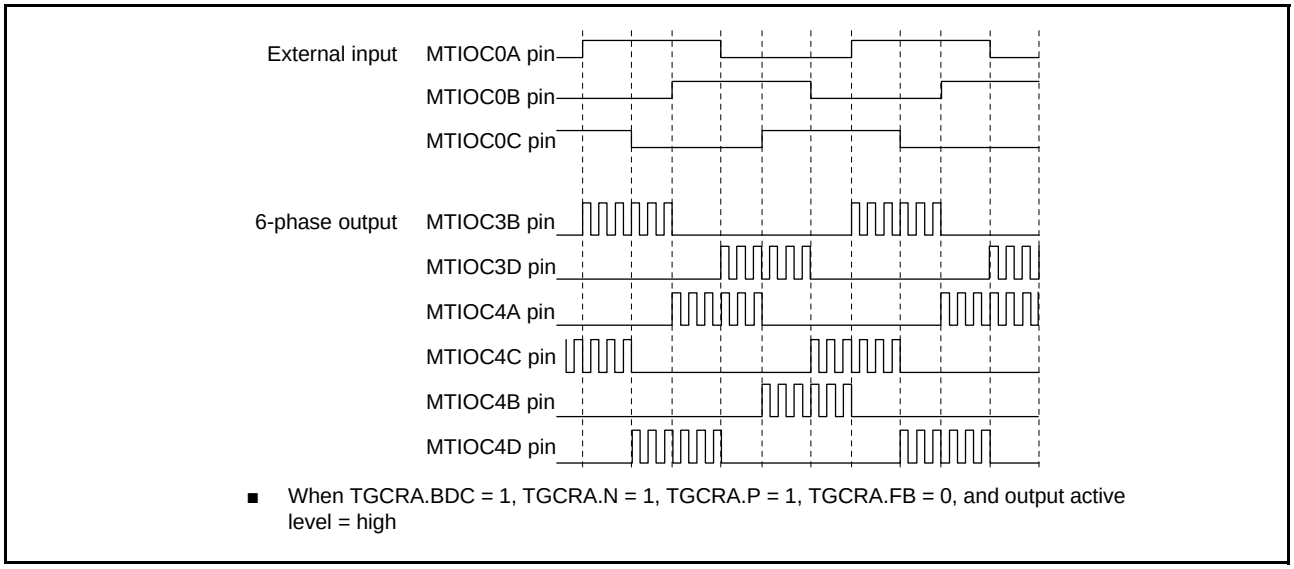
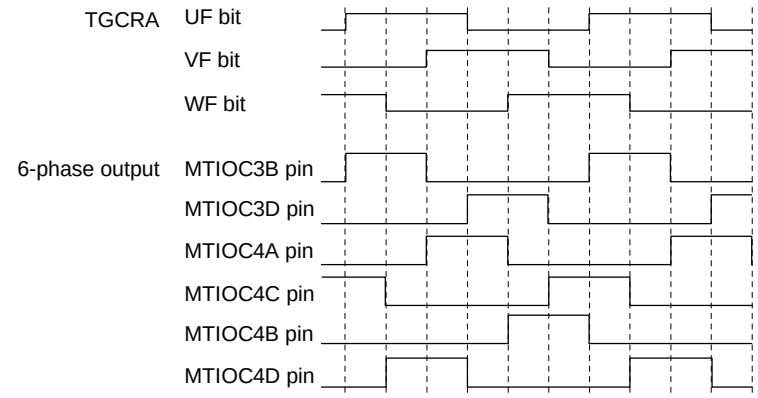
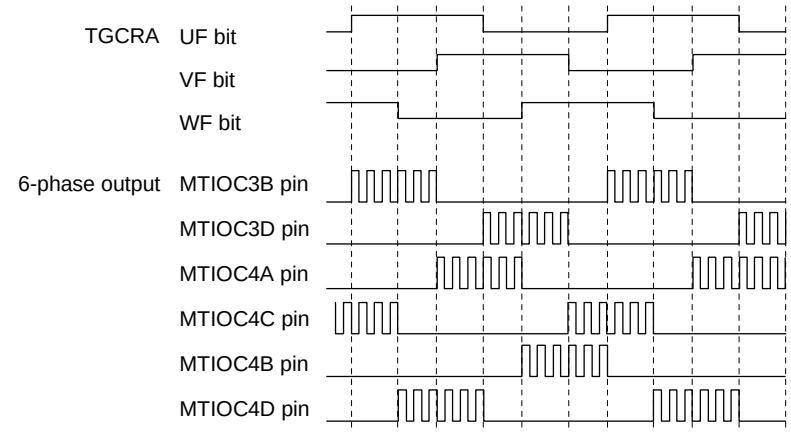


Figure 19.71 Example of Output Phase Switching by External Input (2)



- When TGCRA.BDC = 1, TGCRA.N = 0, TGCRA.P = 0, TGCRA.FB = 1, and output active level = high

Figure 19.72 Example of Output Phase Switching through UF, VF, and WF Bit Settings (1)



- When TGCRA.BDC = 1, TGCRA.N = 1, TGCRA.P = 1, TGCRA.FB = 1, and output active level = high

Figure 19.73 Example of Output Phase Switching through UF, VF, and WF Bit Settings (2)

(q) A/D Conversion Start Request Setting

In complementary PWM mode, an A/D conversion start request can be issued using MTU3.TGRA compare match, MTU4.TCNT underflow (trough), or compare match on a channel other than MTU3 and MTU4.
When start requests using MTU3.TGRA compare match are specified, A/D conversion can be started at the crest of the MTU3.TCNT count.
A/D conversion start requests can be specified by setting the TIER.TTGE bit. To issue an A/D conversion start request at an MTU4.TCNT underflow (trough), set the MTU4.TIER.TTGE2bit to 1.

(r) 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 bit to 1.
When setting buffer registers A (MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD), set also buffer registers B (MTU3.TGRE, MTU4.TGRE, and MTU4.TGRF) at the same time. For details of the setting procedure, refer to section 19.3.8 (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 not set to the buffer register A value, asymmetric PWM waveforms are output.

Figure 19.74 shows an example of double buffer operation.
Each register data is transferred as follows.

- After MTU4.TGRD (buffer A) is written to, data is transferred from MTU4.TGRD (buffer A) to Temp3A (temporary A) and from MTU4.TGRF (buffer B) to Temp3B (temporary B).
- With timing (1) in the figure, data is transferred from Temp3A (temporary A) to MTU4.TGRB (compare).
- With timing (2) in the figure, data is transferred from Temp3B (temporary B) to MTU4.TGRB (compare).

In the crest interval (Tb1 interval), the compare register and temporary register A are valid; in the trough interval (Tb2 interval), the compare register and temporary register B are valid.

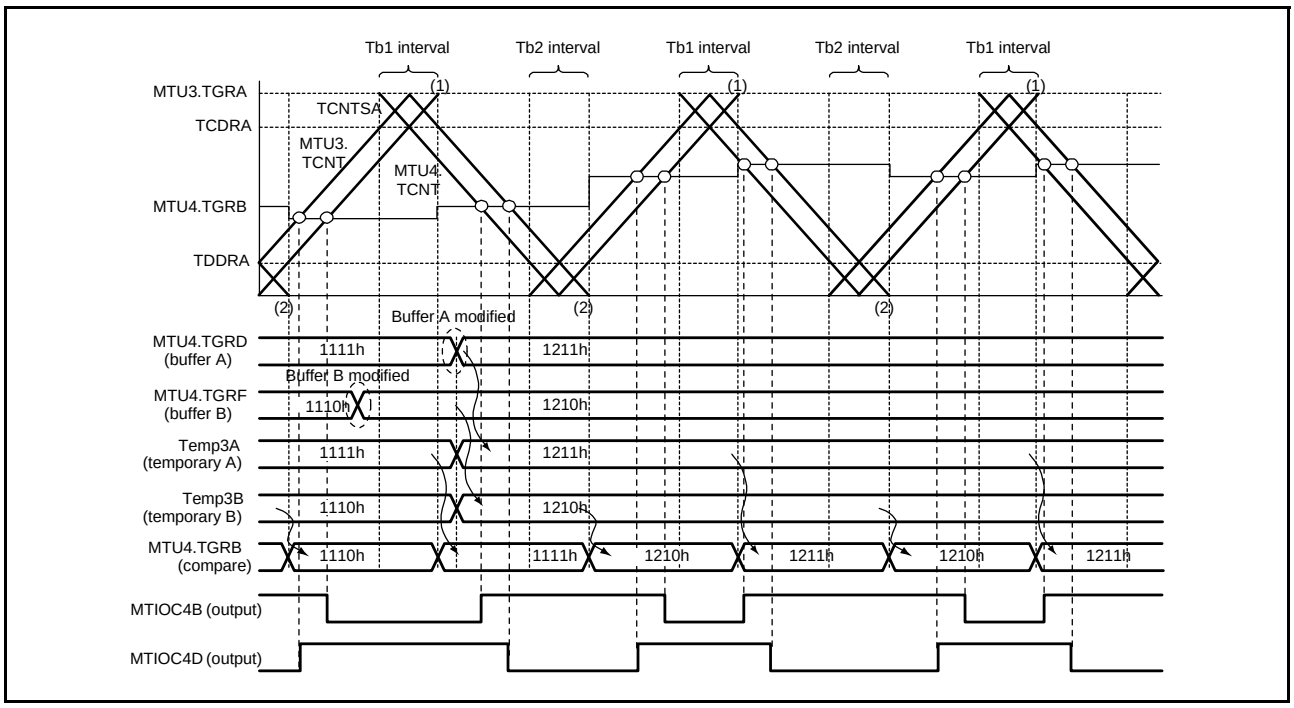


Figure 19.74 Example of Double Buffer Operation

Figure 19.75 shows an example when the buffer write value is smaller than the TDDRA value, and Figure 19.76 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.

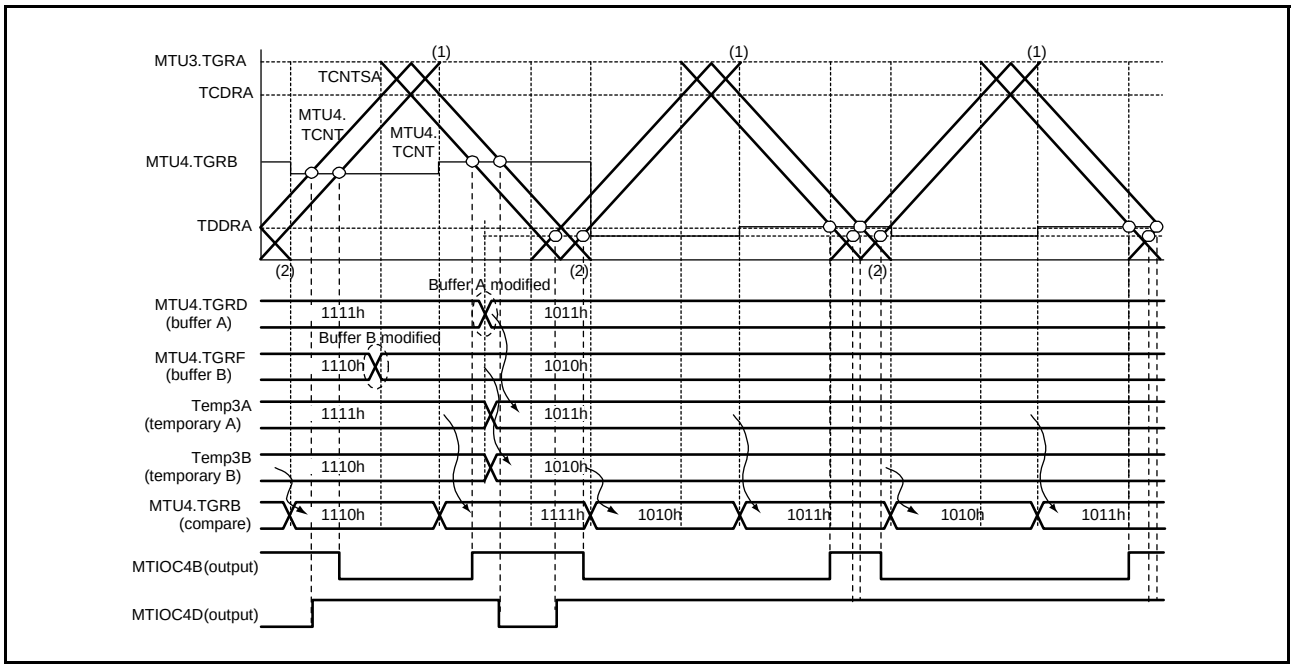


Figure 19.75 Example of Double Buffer Operation (Buffer Write Value is Smaller than TDDRA)

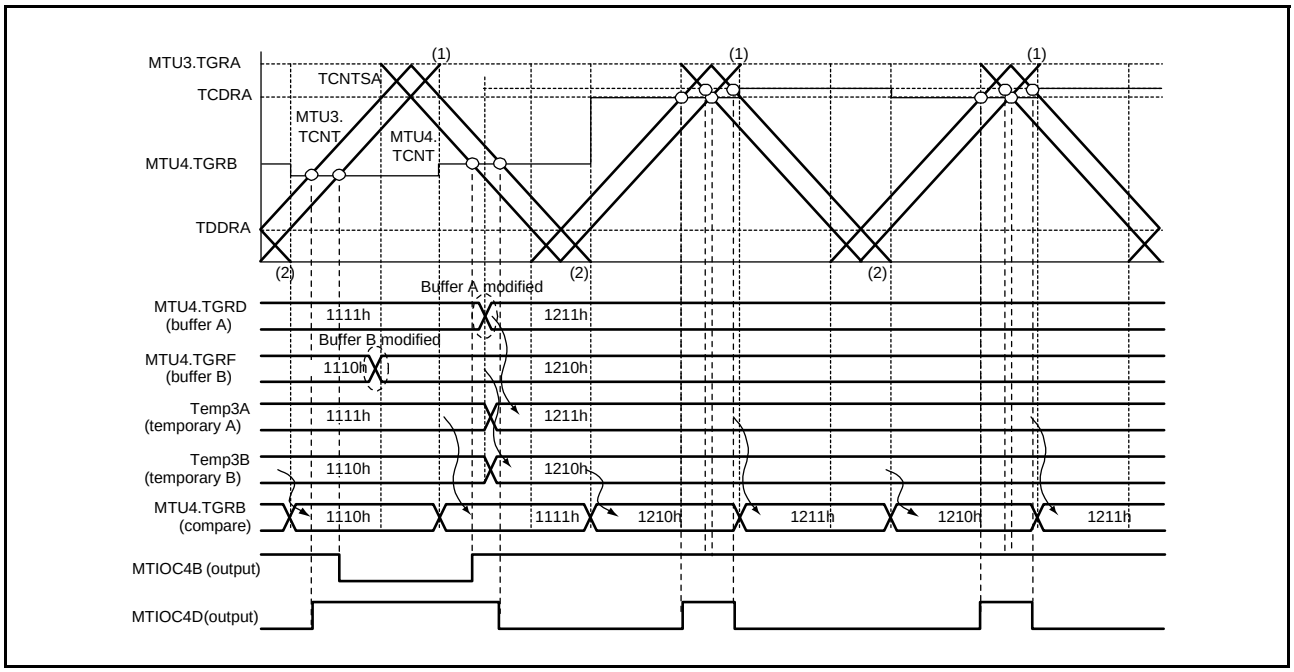


Figure 19.76 Example of Double Buffer Operation (Buffer Write Value is Greater than TCDRA)

(3) Interrupt Skipping Function 1 in Complementary PWM Mode

Interrupts TGIA3 (at the crest) and TCIV4 (at the trough) in MTU3 and MTU4 can be skipped up to seven times by making settings in the TITCR1A 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 register. For the linkage with buffer registers, refer to description (c), Buffer Transfer Control Linked with Interrupt Skipping, below.

A/D conversion start requests generated by the A/D conversion start request delaying function can also be skipped in coordination with interrupt skipping by making settings in the MTU4.TADCR register. For the linkage with the A/D conversion start request delaying function, refer to section 19.3.9, A/D Conversion Start Request Delaying Function.

The TITCR1A register should be set while interrupt skipping function 1 is selected by setting the TITM bit in the timer interrupt skipping mode register (TITMRA) to 0, TGIA3 interrupt requests are disabled by setting the MTU3.TIER register, TCIV4 interrupt requests are disabled by setting the MTU4.TIER register, and a compare match is not generated. Before changing the skipping count, be sure to set the T3AEN and T4VENbits to 0 to clear the skipping counter.

(a) Example of Interrupt Skipping Function 1 Setting Procedure

Figure 19.77 shows an example of the interrupt skipping function 1 setting procedure. Figure 19.78 shows the periods during which interrupt skipping count can be changed.

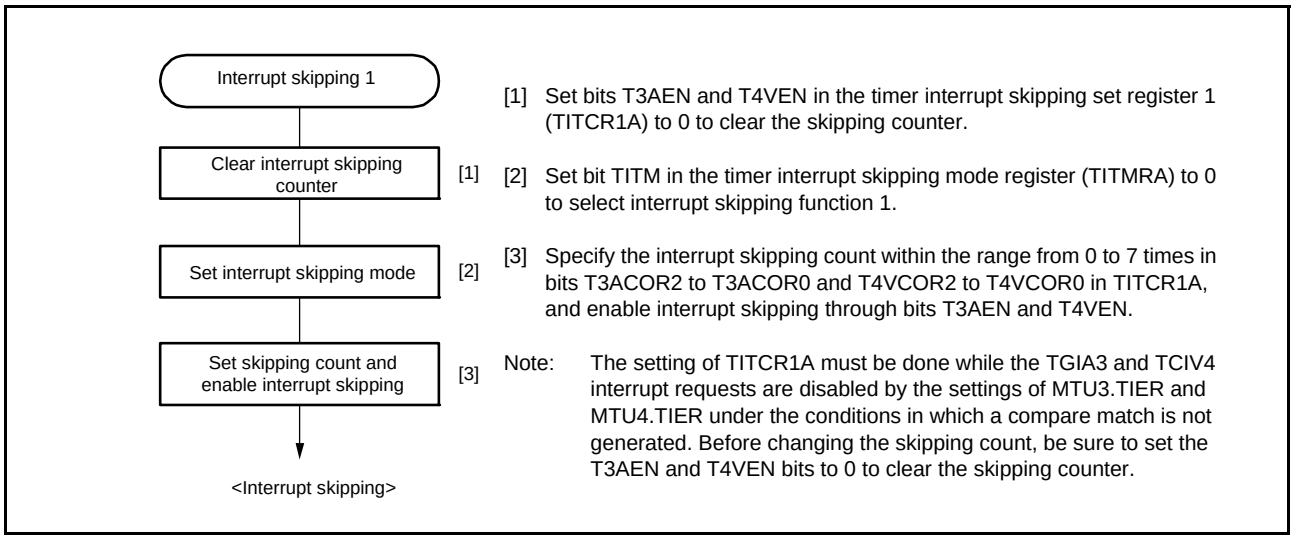


Figure 19.77 Example of Interrupt Skipping Function 1 Setting Procedure

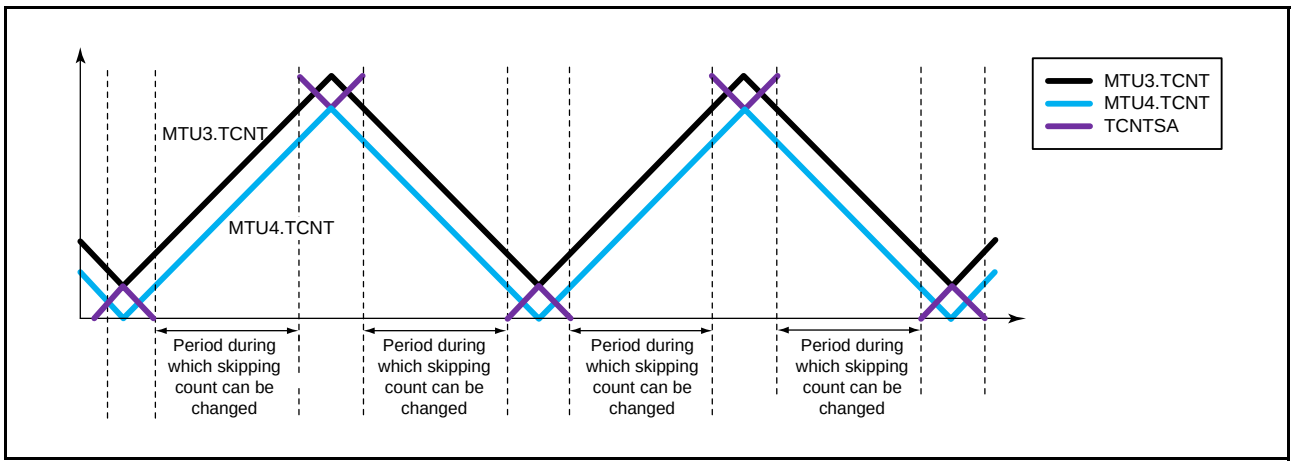


Figure 19.78 Periods during which Interrupt Skipping Count can be Changed

(b) Example of Interrupt Skipping Function 1

Figure 19.79 shows an example of TGIA3 interrupt skipping in which the interrupt skipping count is set to three by the T3ACOR bits and the T3AEN bit is set to 1 in the TITCR1A register.

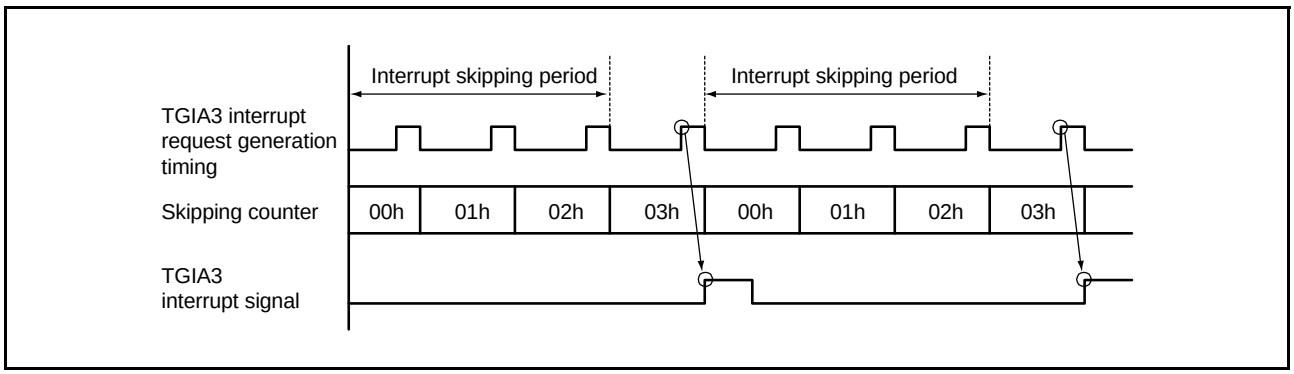
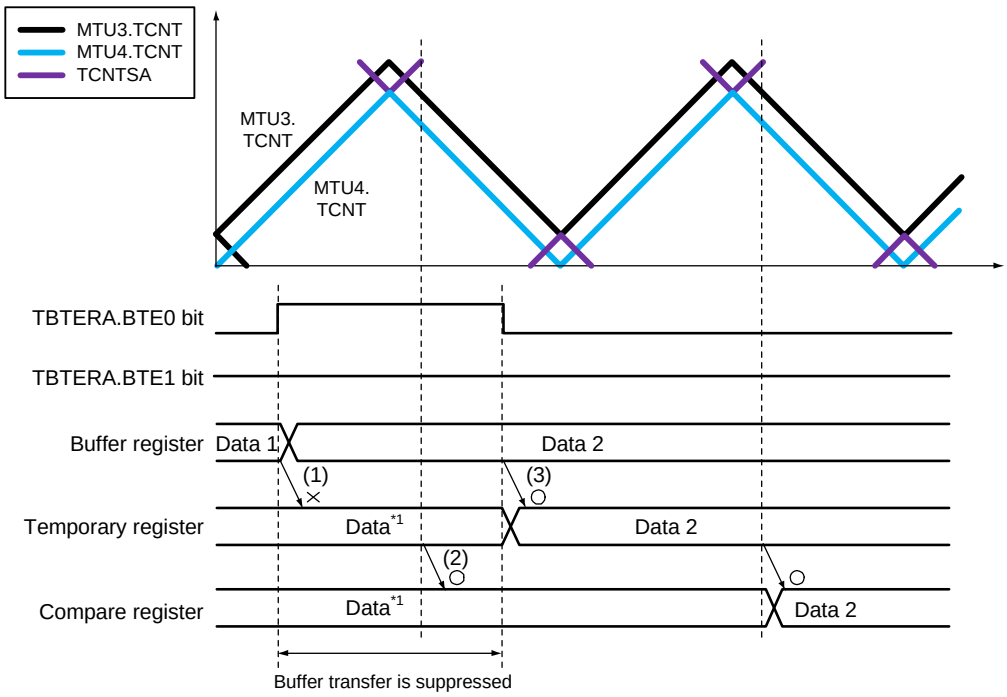


Figure 19.79 Example of Interrupt Skipping Function 1

(c) 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 register.
Figure 19.80 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 19.81 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 differs depending on whether only the T3AEN bit in the TITCR1A register is set to 1, only the T4VEN bit in the TITCR1A register is set to 1, or both the T3AEN and T4VEN bits are set to 1. Figure 19.82 shows the relationship between the T3AEN and T4VEN bit settings in TITCR1A 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 bits in the timer interrupt skipping set register 1 (TITCR1A) are set to 0 or the skipping count set bits (T3ACOR and T4VCOR) in TITCR1A are set to 0), make sure that buffer transfer is not linked with interrupt skipping (set the BTE1 bit in TBTERA to 0).
If buffer transfer is linked with interrupt skipping while interrupt skipping is disabled, buffer transfer is never performed.



(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 19.80 Example of Operation When Buffer Transfer is Disabled (BTE[1:0] = 01b)

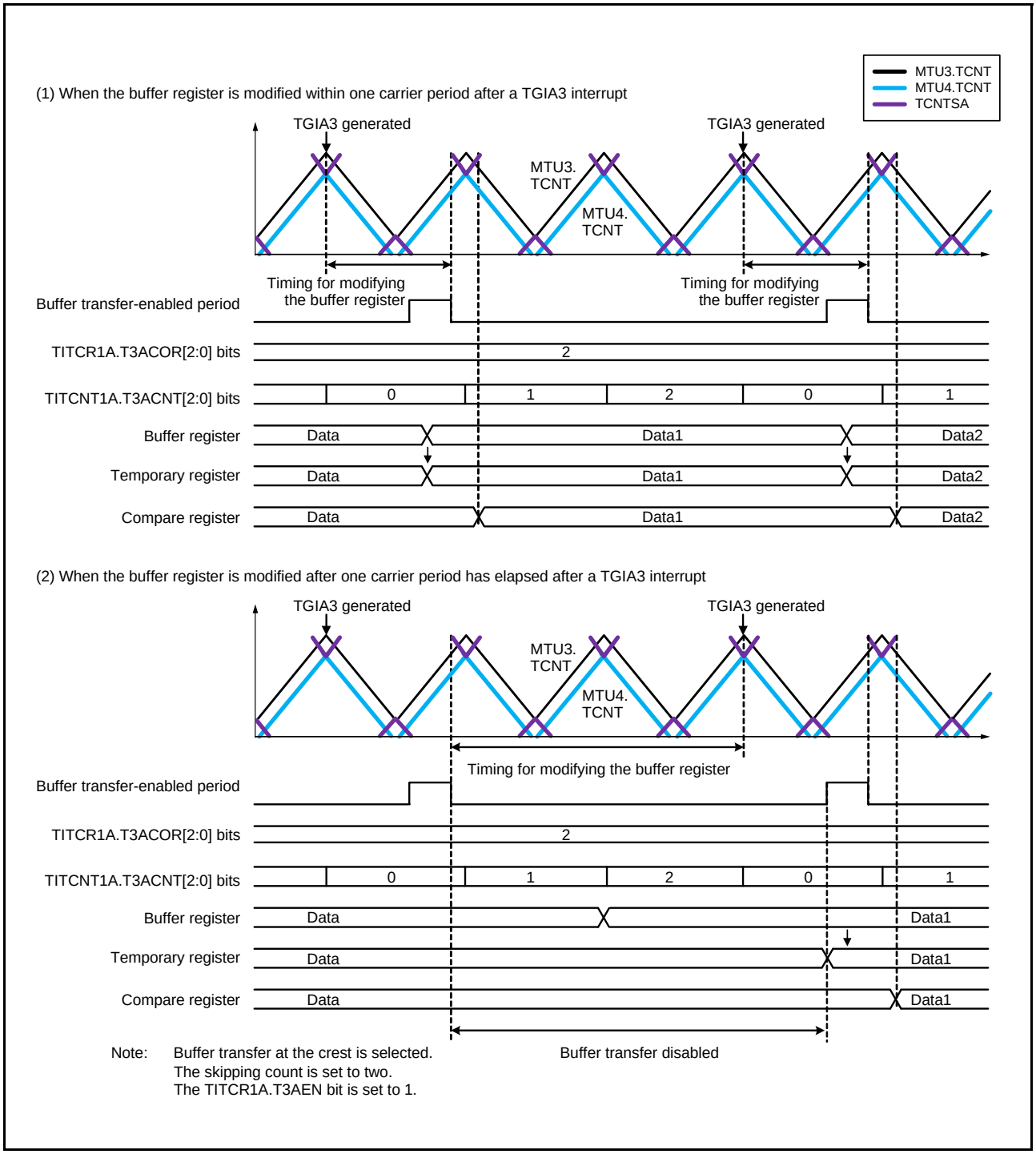


Figure 19.81 Example of Operation When Buffer Transfer is Linked with Interrupt Skipping (BTE[1:0] = 10b)

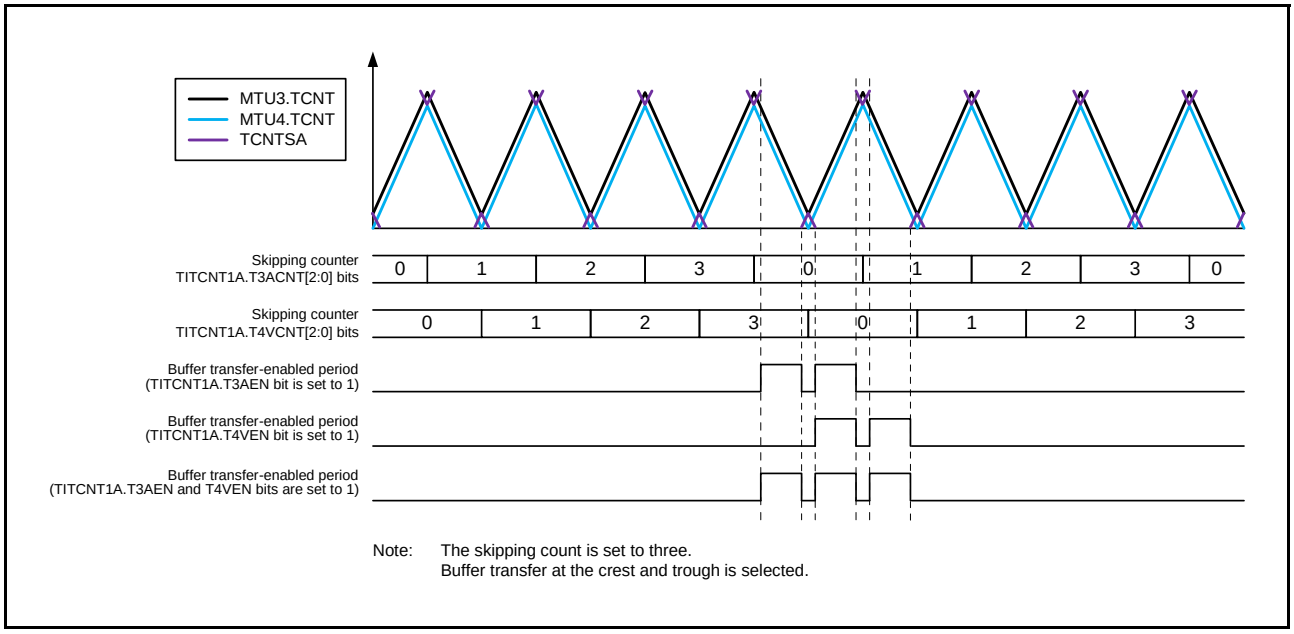


Figure 19.82 Relationship between Bits T3AEN and T4VEN in TITCR1A and Buffer Transfer-Enabled Period

(4) Complementary PWM Mode Output Protection Functions

The following output protection functions are provided for complementary PWM mode.

(a) Register and Counter Miswrite Prevention Function

Access from the CPU to the mode registers, control registers, compare registers, and counters can be enabled or disabled by setting the RWE bit in the TRWERA register. The applicable registers are some of the registers in MTU3 and MTU4 shown below:
24 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.TOERA, MTU.TOCR1A, MTU.TOCR2A, MTU.TGCRA, MTU.TCDRA, and MTU.TDDRA
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.

(b) Halting of PWM Output by External Signal

The PWM output pins of MTU0, MTU3, and MTU4 can be set to the high-impedance state automatically.
Refer to section 20, Port Output Enable 3 (POE3C), for details.

19.3.9 A/D Conversion Start Request Delaying Function

A/D conversion start requests can be issued in MTU4 by making settings in the timer A/D conversion start request control register (MTU4.TADCR), timer A/D conversion start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB), and timer A/D conversion start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB).
The A/D conversion start request delaying function compares MTU4.TCNT with MTU4.TADCORA or MTU4.TADCORB, and when their values match, the function issues a respective A/D conversion start request (TRG4AN or TRG4BN).
A/D conversion start requests (TRG4AN and TRG4BN) can be skipped in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in MTU4.TADCR.

(1) Example of Procedure for Specifying A/D Conversion Start Request Delaying Function

Figure 19.83 shows an example of procedure for specifying the A/D conversion start request delaying function.

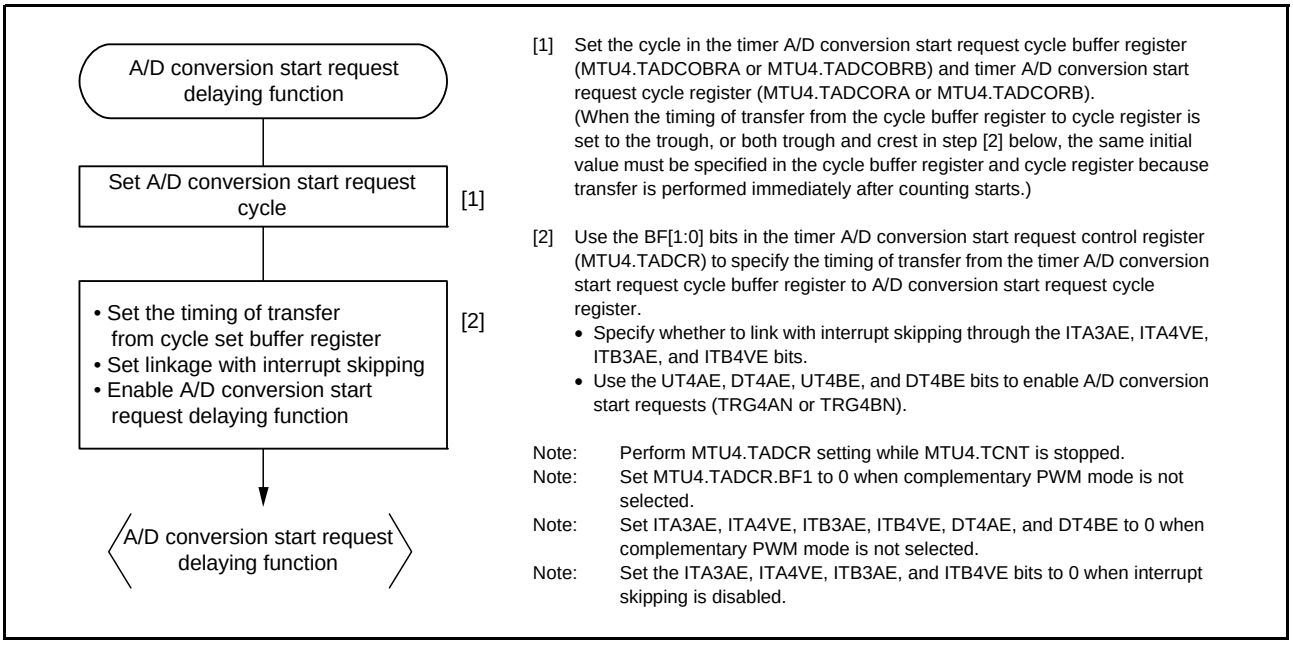
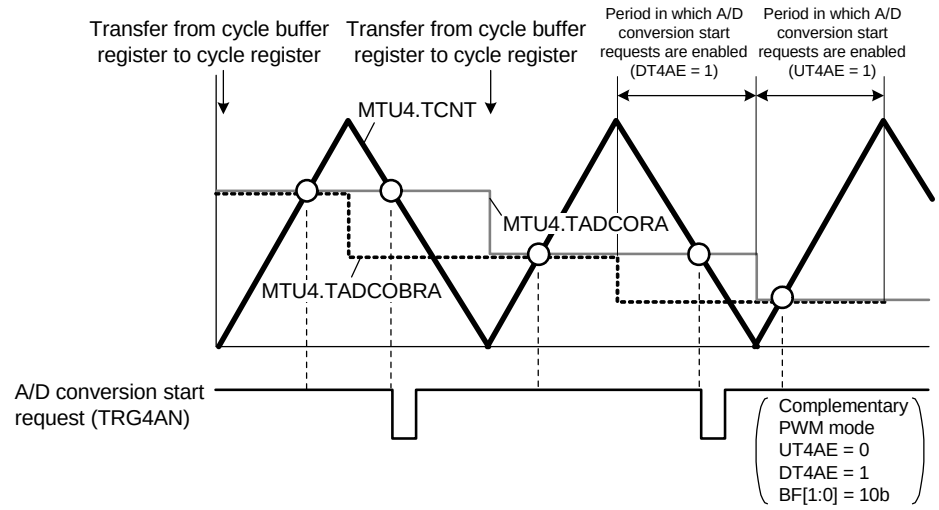


Figure 19.83 Example of Procedure for Specifying A/D Conversion Start Request Delaying Function (MTU3 and MTU4)

(2) Basic Example of A/D Conversion Start Request Delaying Function Operation

Figure 19.84 shows a basic example of A/D conversion start request signal (TRG4AN) operation when the trough of MTU4.TCNT is specified for the buffer transfer timing and an A/D conversion start request signal is output during MTU4.TCNT down-counting.



Period in which A/D conversion start requests are enabled (UT4AE = 1): 0 ≤ MTU4.TCNT ≤ TCDR – 1
Period in which A/D conversion start requests are enabled (DT4AE = 1): TCDR ≥ MTU4.TCNT ≥ 1

UT4AE, DT4AE, BF[1:0]: Bits in TADCR

Figure 19.84 Basic Example of A/D Conversion Start Request Signal (TRG4AN) Operation

(3) Period in Which A/D Conversion Start Requests are Enabled

When the MTU4.TCNT counter and the MTU4.TADCORA or MTU4.TADCORB register matches within the period enabled by the UT4AE and UT4BE bits, the corresponding A/D conversion start request (TRG4AN or TRG4BN) is issued.

When the UT4AE and UT4BE bits in the MTU4.TADCR register are set to 1 in complementary PWM mode, A/D conversion start requests are enabled during the MTU4.TCNT up-counting (0 ≤ MTU4.TCNT ≤ TCDR – 1). When the DT4AE and DT4BE bits in the MTU4.TADCR register are set to 1, A/D conversion start requests are enabled during MTU4.TCNT down-counting (TCDR ≥ MTU4.TCNT ≥ 1). Refer to Figure 19.84.

(4) Buffer Transfer

The data in the timer A/D conversion start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB) is updated by writing data to the timer A/D conversion start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.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 register.

In complementary PWM mode, data is also transferred from the timer A/D conversion start request cycle set buffer registers to the timer A/D conversion start request cycle set registers when MTU4.TGRD register is updated.

There are notes on the timing for transferring data when using buffer transfer in complementary PWM mode.

For details, section 19.6.27, Usage Notes on A/D Conversion Start Request Delaying Function in Complementary PWM Mode.

In modes other than complementary PWM mode, set the BF1 bit in the MTU4.TADCR register to 0.

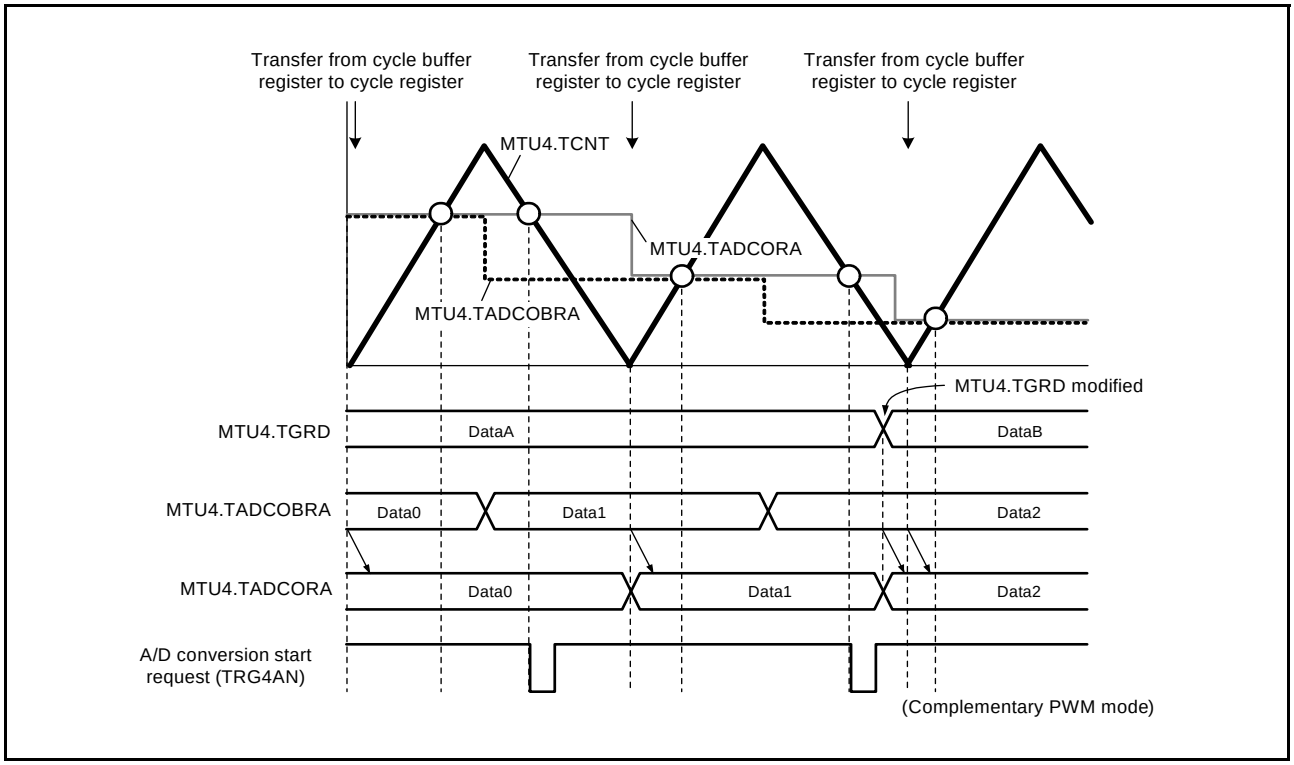


Figure 19.85 Example of A/D Conversion Start Request Signal (TRG4AN) and Buffer Transfer Operation

(5) A/D Conversion Start Request Delaying Function Linked with Interrupt Skipping Function 1

In complementary PWM mode, A/D conversion start requests (TRG4AN and TRG4BN) can be issued in coordination with interrupt skipping 1 by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the MTU4.TADCR register.

Figure 19.86 shows an example of A/D conversion start request signal (TRG4AN) operation when TRG4AN output is enabled during MTU4.TCNT up-counting and down-counting and A/D conversion start requests are linked with interrupt skipping 1.

Figure 19.87 shows another example of A/D conversion start request signal (TRG4AN) operation when TRG4AN output is enabled during MTU4.TCNT up-counting and A/D conversion start requests are linked with interrupt skipping 1.

In modes other than complementary PWM mode, do not use the A/D conversion start request delaying function linked with the interrupt skipping function 1.

Set the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the MTU4.TADCR register to 0.

Note: This function should be used in combination with interrupt skipping 1.
When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR1A) are set to 0 or the skipping count set bits (T3ACOR and T4VCOR) in TITCR1A are set to 0), make sure that A/D conversion start requests are not linked with interrupt skipping 1 (set the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D conversion start request control register (MTU4.TADCR) to 0).
When this function is used, MTU4.TADCORA and MTU4.TADCORB should be set with the value ranging 0002h to the value set in TCDRA minus 2.

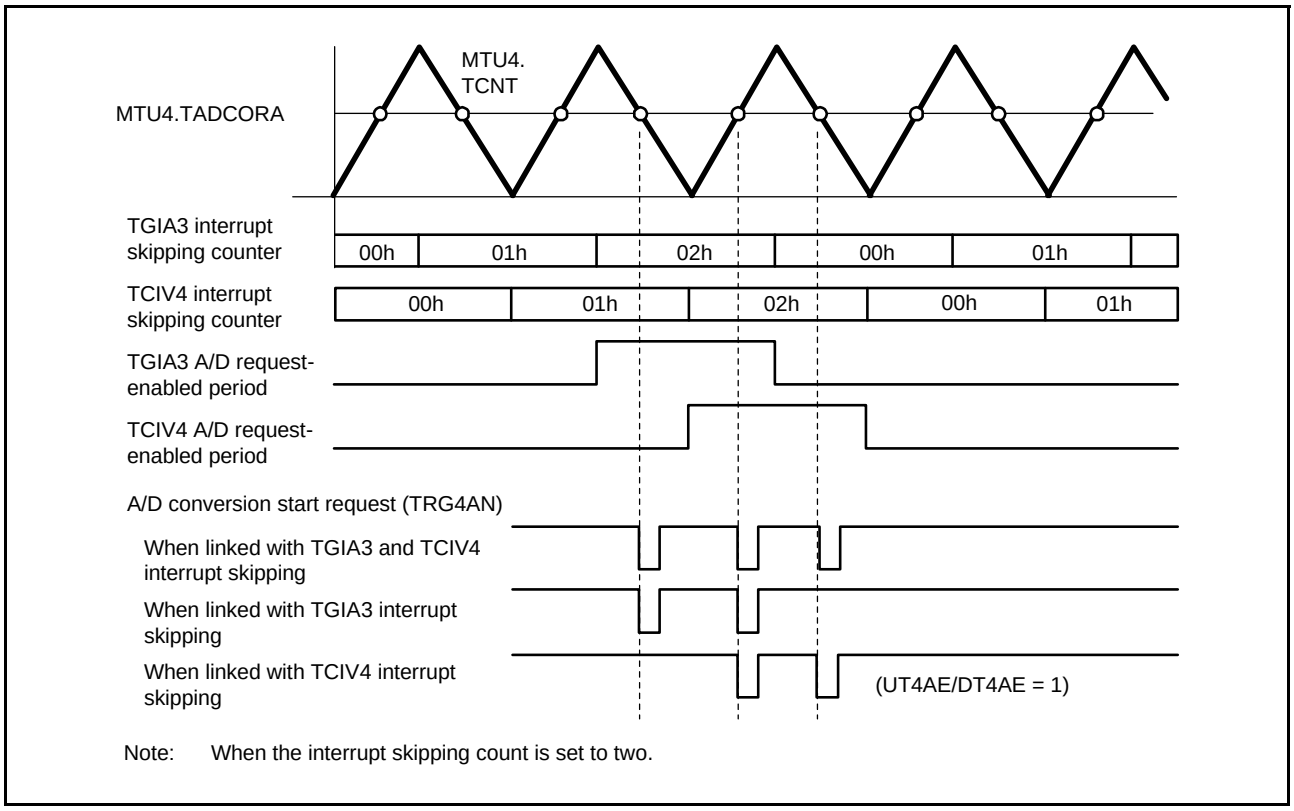


Figure 19.86 Example of A/D Conversion Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping Function 1 (UT4AE and DT4AE = 1)

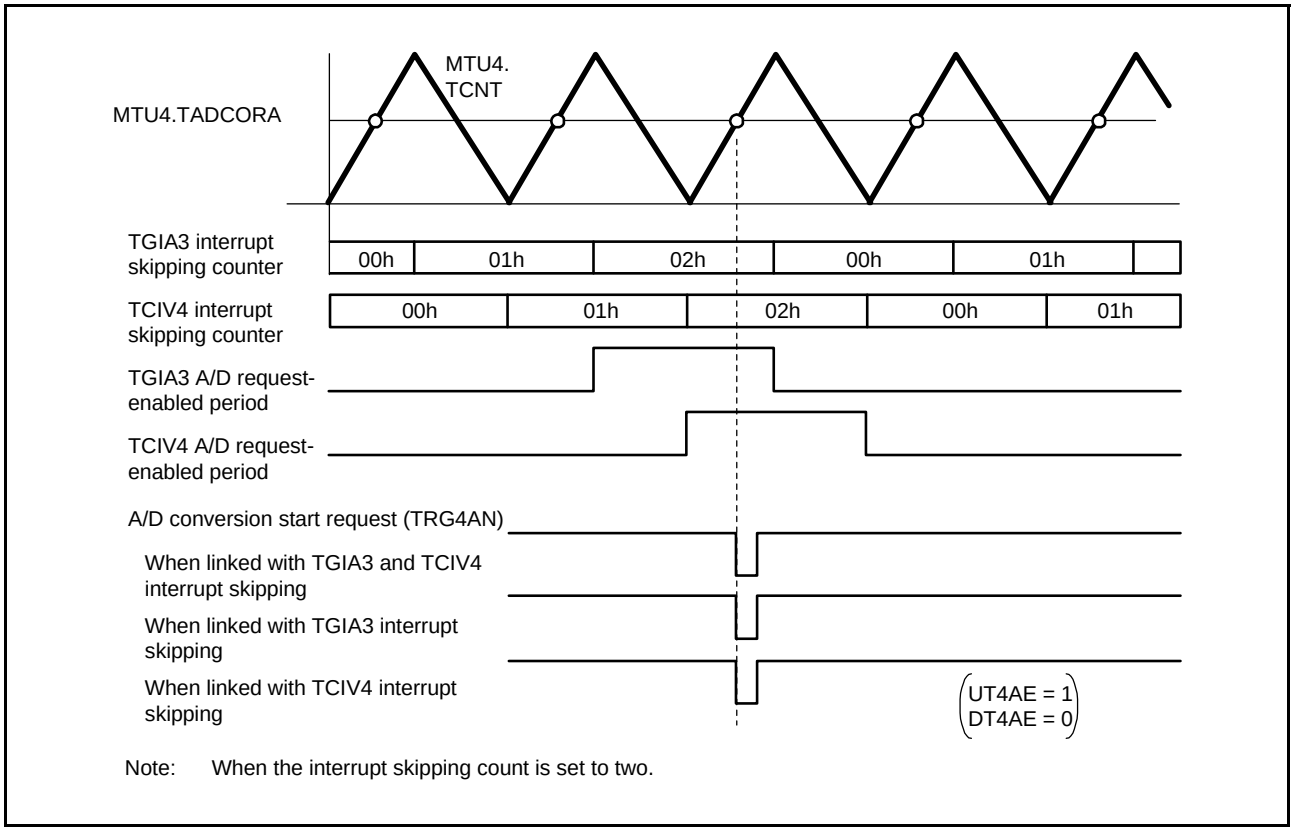


Figure 19.87 Example of A/D Conversion Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping Function 1 (UT4AE = 1, DT4AE = 0)

(6) A/D Conversion Start Request Delaying Function Linked with Interrupt Skipping Function 2

By setting the TITM bit to 1 in the TITMRA register, the counter starts down-counting from the value (0 to 7) set in the TRG4COR[2:0] bits in TITCR2A register every time an A/D conversion start trigger (TRG4AN or TRG4BN) is generated. When the counter value reaches 0 and is reloaded, the TRG4AN and TRG4BN interrupts become valid and an A/D conversion start request signal (TRG4ABN) is output.
This function is valid only when the A/D conversion start request delaying function is enabled.

(a) Example of Procedure for Setting Interrupt Skipping Function 2

Figure 19.88 shows an example of procedure for setting interrupt skipping function 2.

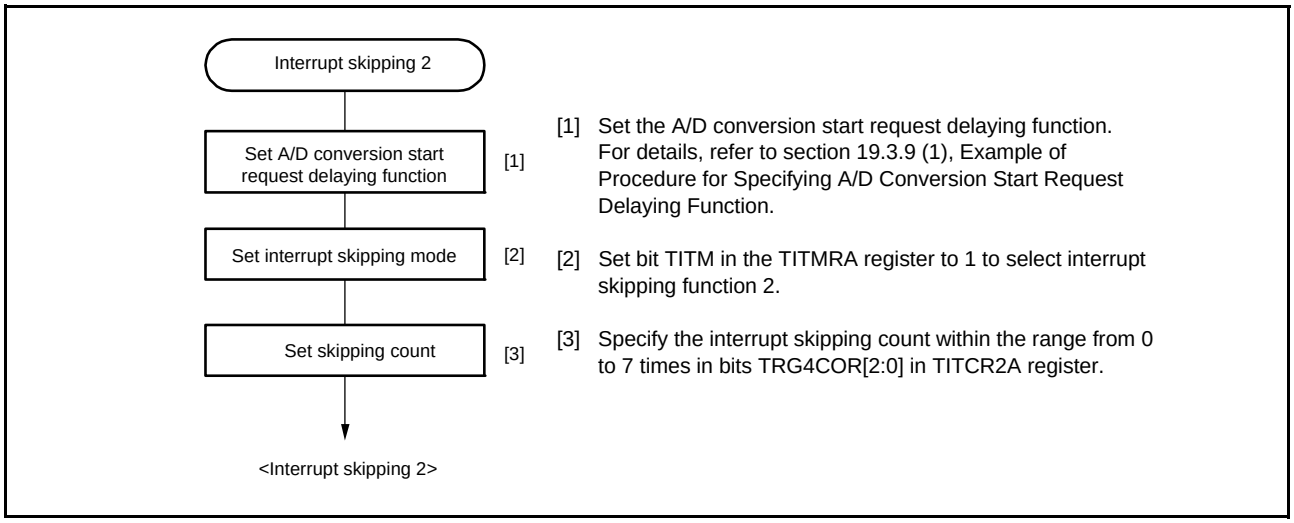


Figure 19.88 Example of Procedure for Setting Interrupt Skipping Function 2

(b) Example of Interrupt Skipping Function 2 Operation

Figure 19.89 shows an example of interrupt skipping function 2 operation.

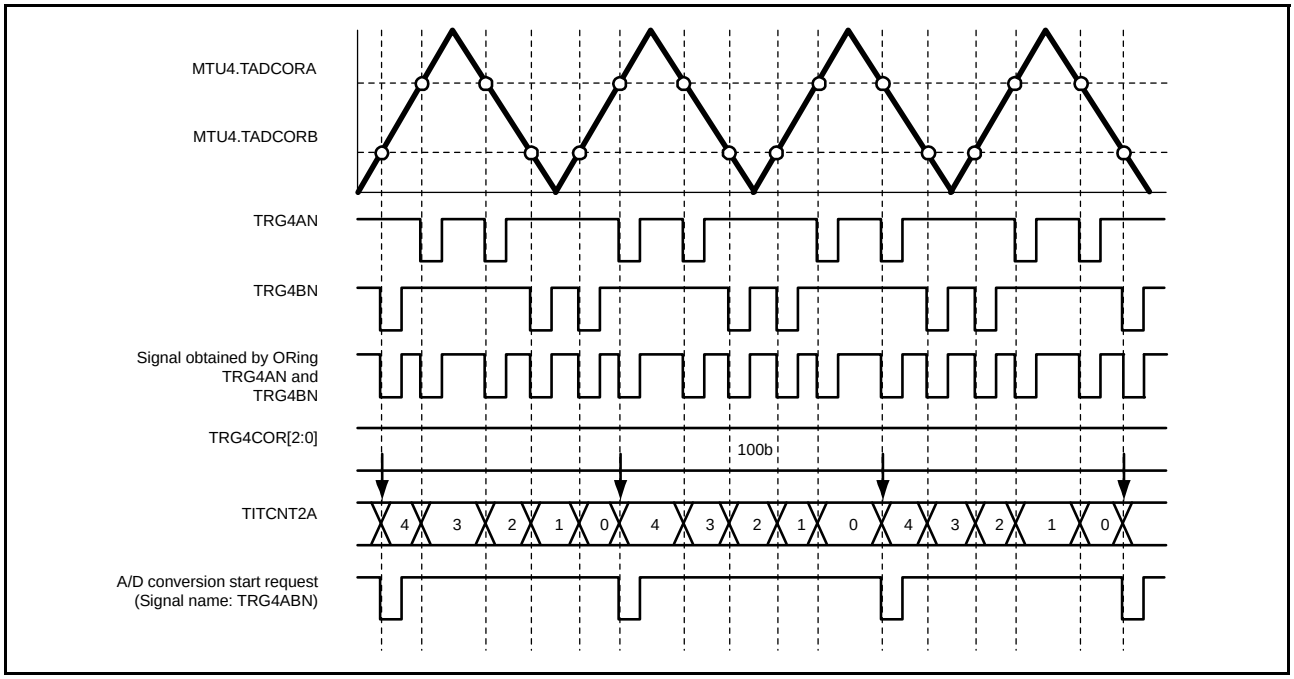


Figure 19.89 Example of Interrupt Skipping Function 2 Operation (Skipping Count is Set to Four)

19.3.10 Synchronous Operation of MTU0 to MTU4

(1) Synchronous Counter Start for MTU0 to MTU4

The counters in MTU0 to MTU4 can be started synchronously by making the TCSYSTR settings.

(a) Example of Procedure for Setting Synchronous Counter Start for MTU0 to MTU4

Figure 19.90 shows an example of procedure for setting synchronous counter start for MTU0 to MTU4.

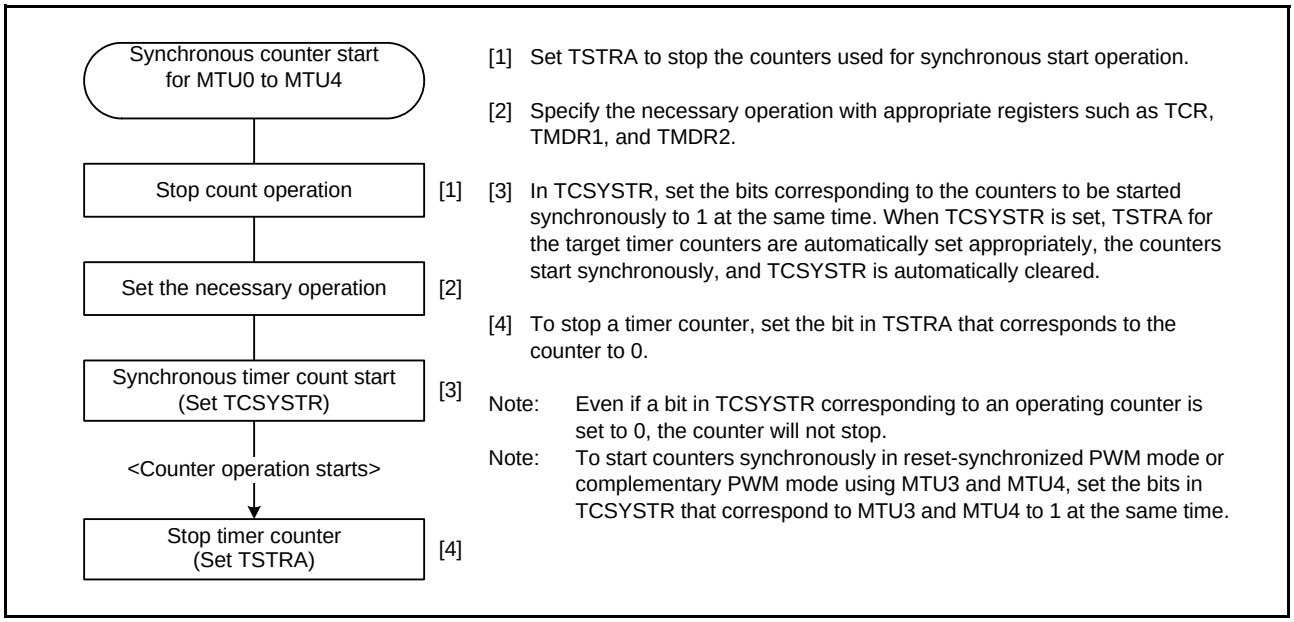


Figure 19.90 Example of Procedure for Setting Synchronous Counter Start for MTU0 to MTU4

(b) Examples of Synchronous Counter Start Operation

Figure 19.91 shows an examples of synchronous counter start operation for MTU0 to MTU4.

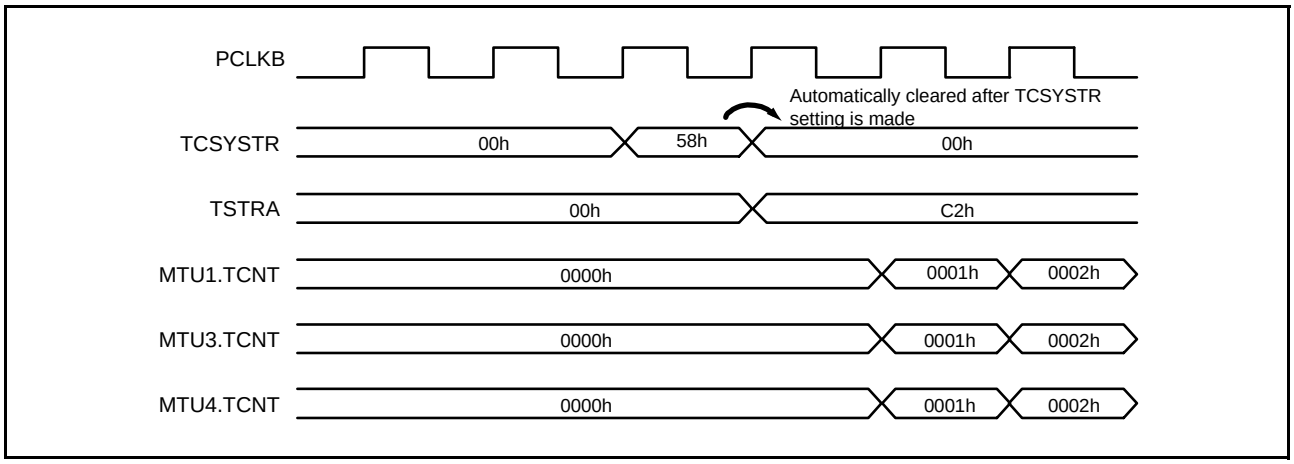


Figure 19.91 Examples of Synchronous Counter Start Operation for MTU0 to MTU4

19.3.11 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 19.92 shows an example of setting external pulse width measurement, and Figure 19.93 an example of external pulse width measurement.

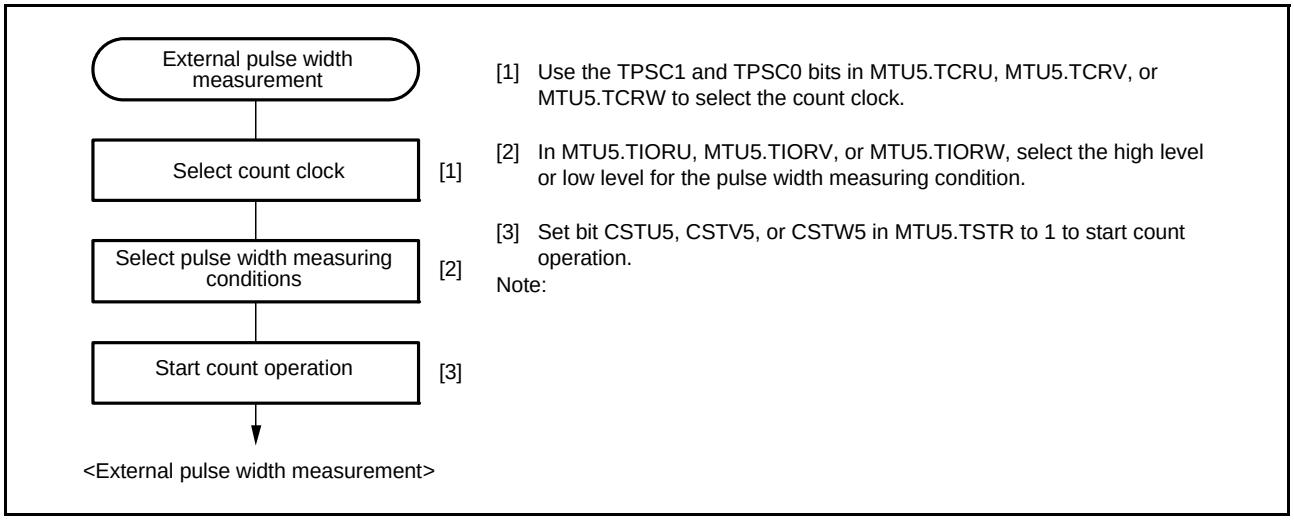


Figure 19.92 Example of External Pulse Width Measurement Setting Procedure

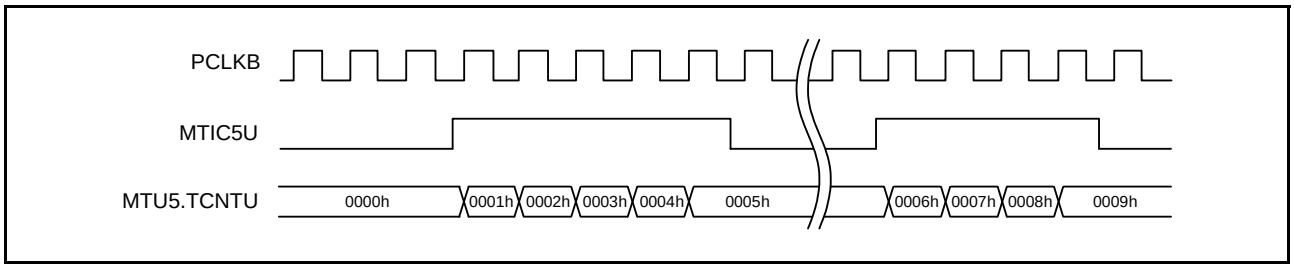


Figure 19.93 Example of External Pulse Width Measurement (Measuring High Pulse Width)

19.3.12 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 MTU3 and MTU4. Figure 19.94 shows an example of the motor control circuit compensating a dead time delay by combining MTU5 with MTU3 and MTU4. A dead time for the PWM output waveform during complementary PWM operation using MTU3 and MTU4 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 19.95). Figure 19.96 shows the procedure for setting dead time compensation using MTU3 to MTU5. For details on MTU5 operation at this time, refer to section 19.3.13, TCNTU, TCNTV, and TCNTW Capture at Crest and/or Trough in Complementary PWM Mode.

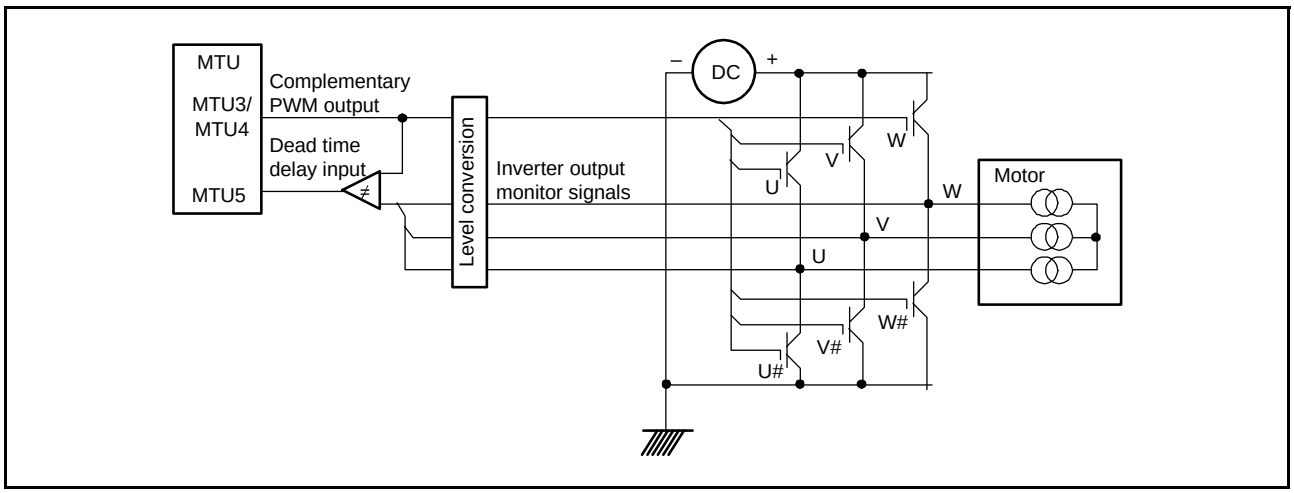


Figure 19.94 Motor Control Circuit Example

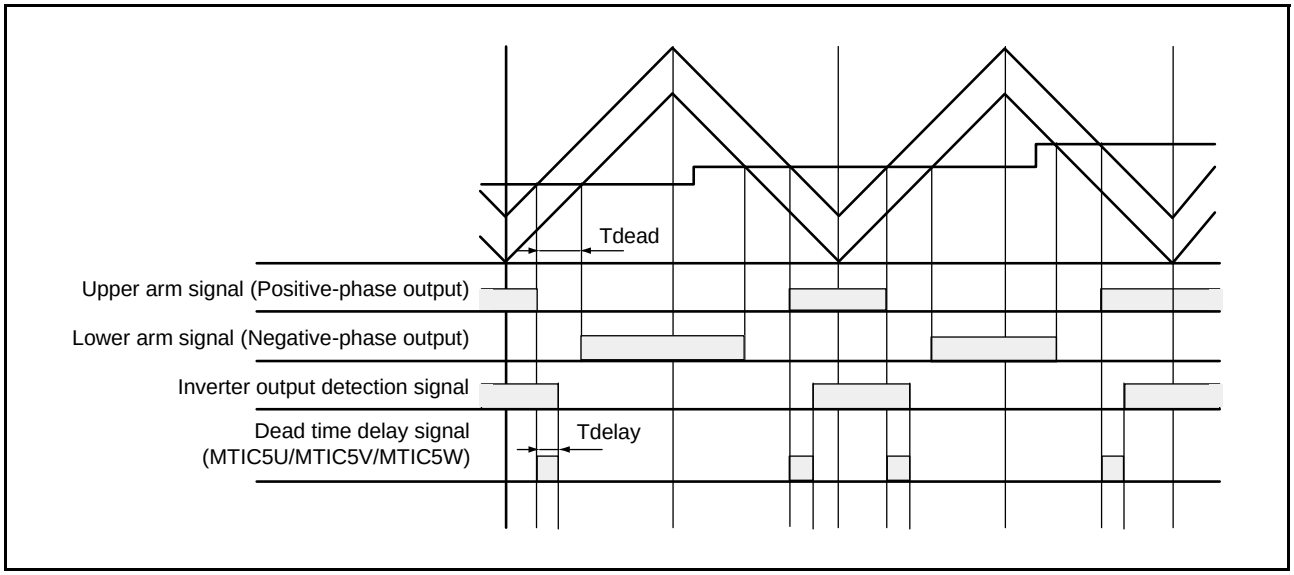
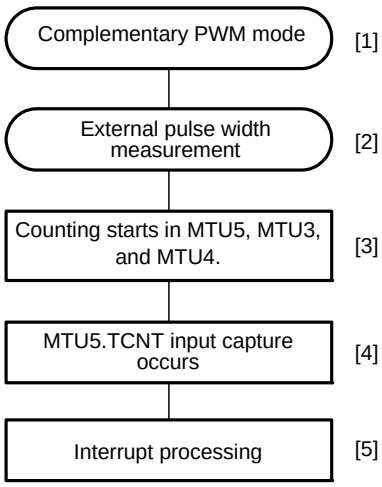


Figure 19.95 Delay in Dead Time in Complementary PWM Operation

(1) Example of Dead Time Compensation Setting Procedure

Figure 19.96 shows an example of dead time compensation setting procedure by using three counters in MTU5.



[1] Set MTU3 and MTU4 to complementary PWM mode. For details, refer to section 19.3.8, Complementary PWM Mode.

[2] Specify the external pulse width measurement function for the target TIOR in MTU5. For details, refer to section 19.3.11, External Pulse Width Measurement.

[3] Set bits CST3 and CST4 in TSTRA and bits CSTU5, CSTV5, and CSTW5 in MTU5.TSTR to 1 to start count operation.

[4] When the capture condition specified in TIOR is satisfied, the MTU5.TCNT value is captured in MTU5.TGR.
To prevent the generation of the MTU5.TGIm5 interrupt (m = U, V, W) request under the capture condition specified in TIOR, do not enable interrupt requests in MTU5.TIER.

[5] For U-phase dead time compensation, when an interrupt is generated at the crest (TGIA3) or trough (TCIV4) in complementary PWM mode, read the MTU5.TGRU value, calculate the difference in time between MTU3.TGRB and MTU5.TGRU, and write the corrected value to MTU3.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 MTU4.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 MPC and port mode registers (PMR) of the I/O ports should be set up in advance.
As the interrupt flag is set under the capture condition specified in TIOR, do not enable interrupt requests in MTU5.TIER.

Figure 19.96 Example of Dead Time Compensation Setting Procedure

19.3.13 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 MTU3 and MTU4 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 0000h at the transfer timing for TGRU, TGRV, and TGRW.

Figure 19.97 shows an operation example in which TCNTU is used as a free-running counter without being cleared, and the value is captured in TGRU at the crest and trough in complementary PWM mode.

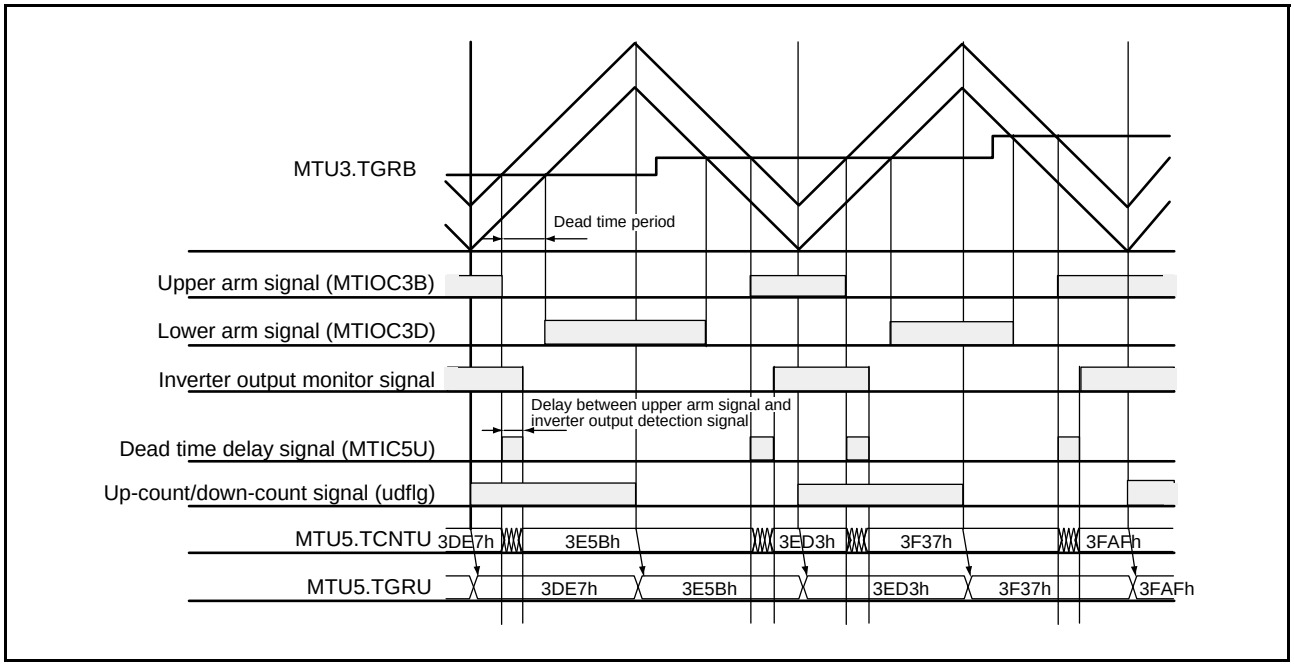


Figure 19.97 TCNTU Capture at Crest and Trough in Complementary PWM Operation

19.3.14 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 5, 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 19.98 shows the timing of noise filtering.

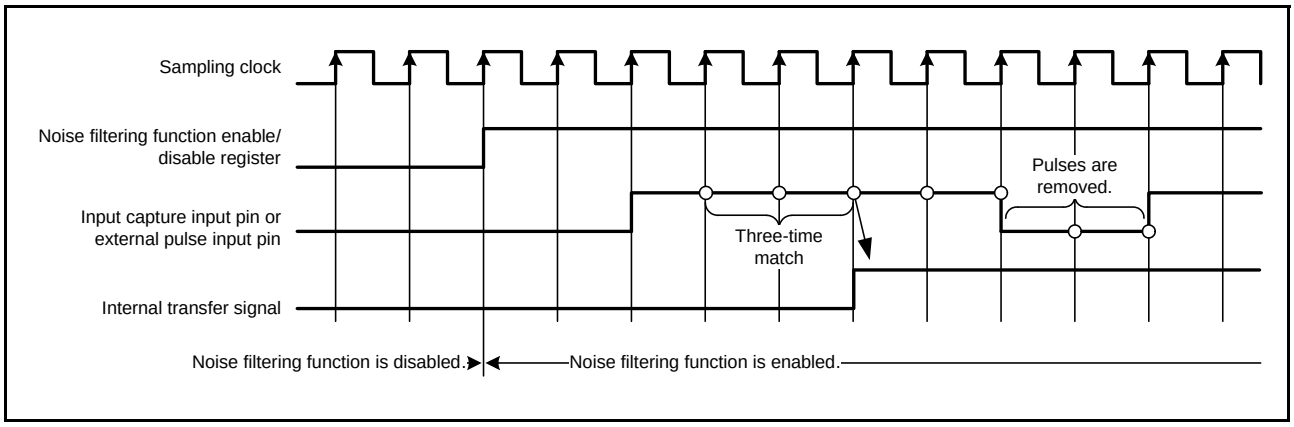


Figure 19.98 Timing of Noise Filtering

19.3.15 A/D Conversion Start Request Frame Synchronization Signal

This function can be used to monitor the generation timing of the A/D conversion start request signal using an external pin.
When the A/D conversion start request signal to be monitored is selected by the TADSTRGR0 register, a pulse signal is output from the ADSM0 pin that is at the high level when the A/D conversion start request signal is generated, and at the low level in the timer cycle used to generate the A/D conversion start request signal.
Figure 19.99 shows an example of outputting the A/D conversion start request frame synchronization signal.

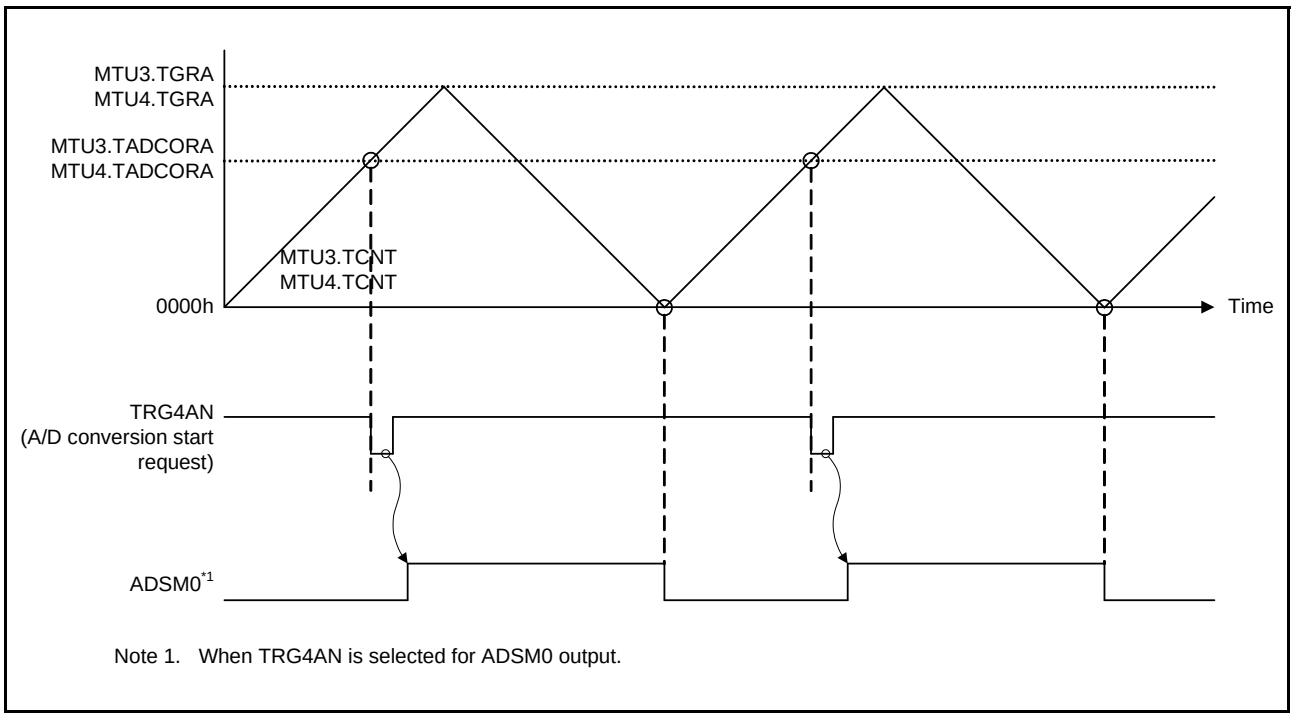


Figure 19.99 Example of Outputting A/D Conversion Start Request Frame Synchronization Signal

19.4 Interrupt Sources

19.4.1 Interrupt Sources and Priorities

There are three kinds of interrupt source; TGR input capture/compare match, TCNT overflow, and 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; however the priority within a channel is fixed. For details, refer to section 14, Interrupt Controller (ICUb). Table 19.62 lists the MTU interrupt sources.

Table 19.62 MTU Interrupt Sources

Channel	Name	Interrupt Source	DTC Activation	Priority
MTU0	TGIA0	MTU0.TGRA input capture/compare match	Possible	High
	TGIB0	MTU0.TGRB input capture/compare match	Possible	
	TGIC0	MTU0.TGRC input capture/compare match	Possible	
	TGID0	MTU0.TGRD input capture/compare match	Possible	
	TCIV0	MTU0.TCNT overflow	Not possible	
	TGIE0	MTU0.TGRE compare match	Not possible	
	TGIF0	MTU0.TGRF compare match	Not possible	
MTU1	TGIA1	MTU1.TGRA input capture/compare match	Possible	
	TGIB1	MTU1.TGRB input capture/compare match	Possible	
	TCIV1	MTU1.TCNT overflow	Not possible	
	TCIU1	MTU1.TCNT underflow	Not possible	
MTU2	TGIA2	MTU2.TGRA input capture/compare match	Possible	
	TGIB2	MTU2.TGRB input capture/compare match	Possible	
	TCIV2	MTU2.TCNT overflow	Not possible	
	TCIU2	MTU2.TCNT underflow	Not possible	
MTU3	TGIA3	MTU3.TGRA input capture/compare match	Possible	
	TGIB3	MTU3.TGRB input capture/compare match	Possible	
	TGIC3	MTU3.TGRC input capture/compare match	Possible	
	TGID3	MTU3.TGRD input capture/compare match	Possible	
	TCIV3	MTU3.TCNT overflow	Not possible	
MTU4	TGIA4	MTU4.TGRA input capture/compare match	Possible	
	TGIB4	MTU4.TGRB input capture/compare match	Possible	
	TGIC4	MTU4.TGRC input capture/compare match	Possible	
	TGID4	MTU4.TGRD input capture/compare match	Possible	
	TCIV4	MTU4.TCNT overflow/underflow*1	Possible	
MTU5	TGIU5	MTU5.TGRU input capture/compare match	Possible	
	TGIV5	MTU5.TGRV input capture/compare match	Possible	
	TGIW5	MTU5.TGRW input capture/compare match	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. Underflow is available 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 21 input capture/compare match interrupts (six for MTU0, four each for MTU3 and MTU4, 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 five overflow interrupts (one for each channel except MTU5).
Note that an overflow interrupt is generated also when an underflow of the MTU4.TCNT occurs while operating in complementary PWM mode.

(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).

19.4.2 DTC Trigger Sources

(1) DTC Trigger Sources

The DTC can be triggered by the TGR input capture/compare match interrupt in each channel or the overflow interrupt in MTU4. For details, refer to section 16, Data Transfer Controller (DTCb).
The MTU provides a total of 20 input capture/compare match interrupts and overflow interrupts that can be used as DTC trigger sources: four each for MTU0 and MTU3, two each for MTU1 and MTU2, five for MTU4, and three for MTU5.

19.4.3 A/D Converter Trigger Sources

The A/D converter can be triggered by one of the following three methods in the MTU. Table 19.63 shows the relationship between interrupt sources and A/D conversion start request signals.

(1) A/D Conversion Start by TGRA Input Capture/Compare Match or at Trough of MTU4.TCNT in Complementary PWM Mode

The A/D converter can be triggered by the occurrence of a TGRA input capture/compare match in each channel. 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 MTU4.TCNT count (MTU4.TCNT = 0000h).
A/D conversion start request TRGAnN is issued to the A/D converter under either of the following conditions (n = 0 to 4).

- When a TGRA input capture/compare match occurs on a channel while the TIER.TTGE bit is set to 1
- When the MTU4.TCNT count reaches the trough (MTU4.TCNT = 0000h) during complementary PWM operation while the TTGE2 bit in MTU4.TIER is set to 1

When either condition is satisfied, if A/D conversion start request 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 conversion start request 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 conversion start request TRG0N is issued to the A/D converter. If A/D conversion start request 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 Conversion Start Request Delaying Function

The A/D converter can be triggered by generating A/D conversion start request signal TRG4AN or TRG4BN when the MTU4.TCNT count matches the MTU4.TADCORA or MTU4.TADCORB value if the UT4AE, DT4AE, UT4BE, or DT4BE bit in the A/D conversion start request control register (MTU4.TADCR) is set to 1. For details, refer to section 19.3.9, A/D Conversion Start Request Delaying Function.
A/D conversion will start when TRG4AN is generated if A/D conversion start request signal TRG4AN from the MTU is selected as the trigger in the A/D converter, when TRG4BN is generated if TRG4BN from the MTU is selected as the trigger in the A/D converter, or when TRG4ABN is generated if TRG4ABN from the MTU is selected as the trigger in the A/D converter.

Table 19.63 Interrupt Sources and A/D Conversion Start Request Signals

Target Registers	Interrupt Source	A/D Conversion 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	
MTU0.TGRE and MTU0.TCNT	Compare match	TRG0N
MTU4.TADCORA and MTU4.TCNT		TRG4AN
MTU4.TADCORB and MTU4.TCNT		TRG4BN
MTU4.TADCORA and MTU4.TCNT, MTU4.TADCORB and MTU4.TCNT	Compare match (interrupt skipping function 2)	TRG4ABN

Note 1. Since PWM waveforms are generated in complementary PWM mode, MTU4.TGRA compare match not only with MTU4.TCNT but also with MTU3.TCNT and TCNTSA is detected. Accordingly, when compare match with MTU3.TCNT and TCNTSA occurs, TRGA4N is also generated.
When MTU3 and MTU 4 are made to operate in complementary PWM mode for generating an A/D conversion start request, use the A/D conversion start request by compare match between MTU4.TCNT and MTU4.TADCORA/TADCORB.

19.5 Operation Timing

19.5.1 Input/Output Timing

(1) TCNT Count Timing

Figure 19.100 and Figure 19.101 show the TCNT count timing in internal clock operation, Figure 19.102 shows the TCNT count timing in external clock operation (normal mode), and Figure 19.103 shows the TCNT count timing in external clock operation (phase counting mode).

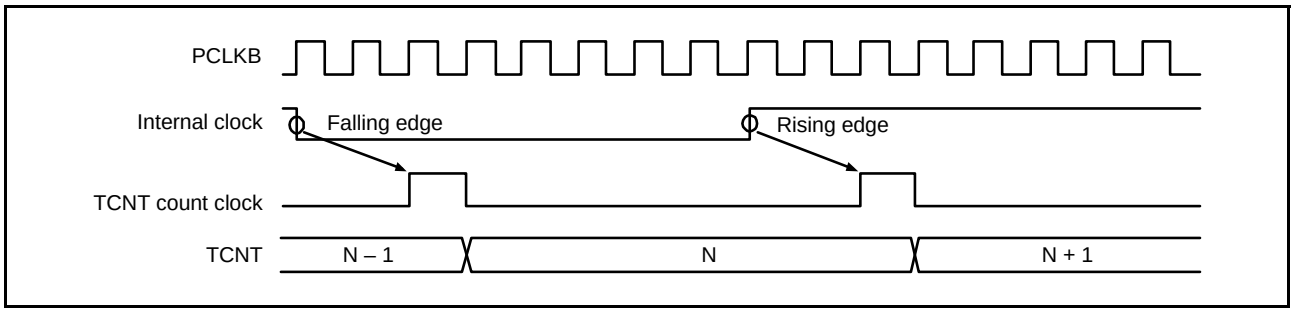


Figure 19.100 Count Timing in Internal Clock Operation (MTU0 to MTU4)

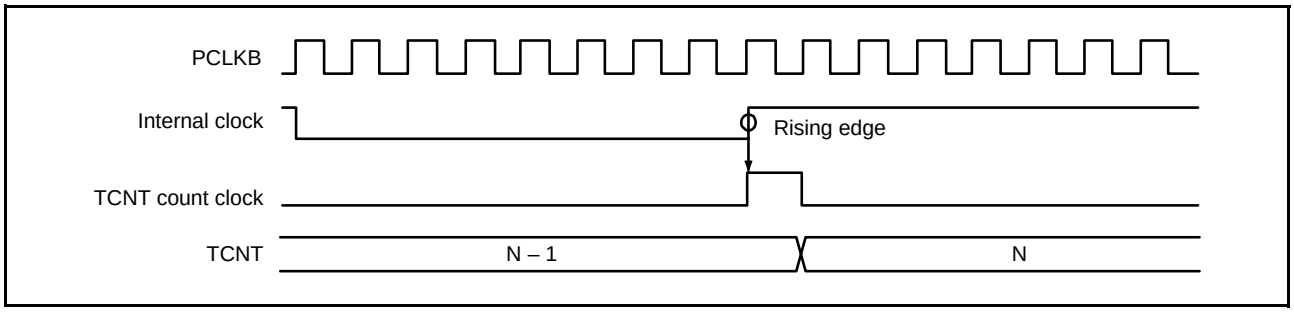


Figure 19.101 Count Timing in Internal Clock Operation (MTU5)

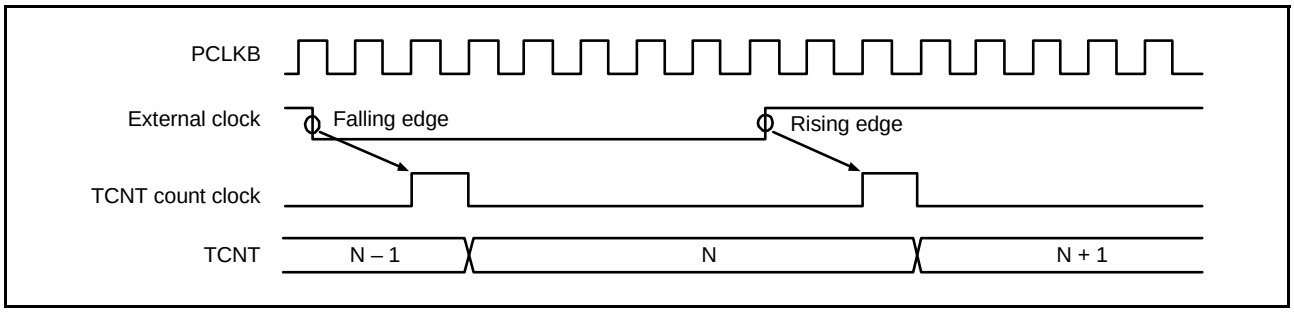


Figure 19.102 Count Timing in External Clock Operation (MTU0 to MTU4)

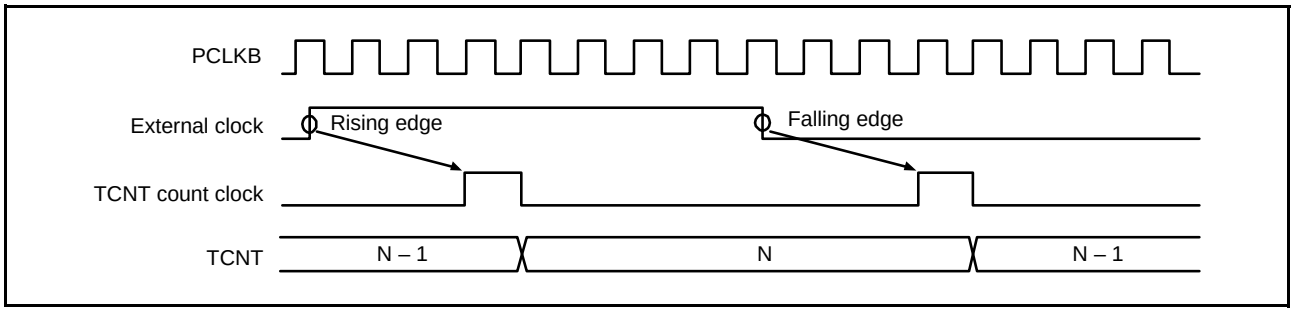


Figure 19.103 Count Timing in External Clock Operation (Phase Counting Mode)

(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; m = A to D). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT count clock is generated.

Figure 19.104 shows the output compare output timing (normal mode or PWM mode) and Figure 19.105 shows the output compare output timing (complementary PWM mode or reset-synchronized PWM mode).

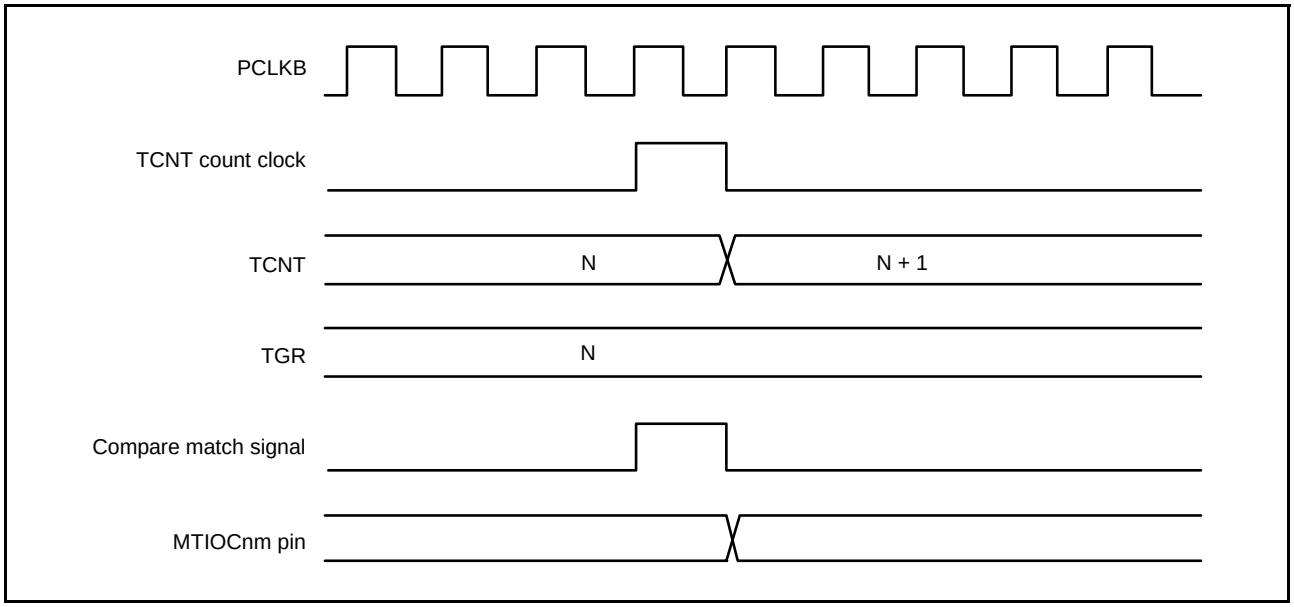


Figure 19.104 Output Compare Output Timing (Normal Mode or PWM Mode) (n = 0 to 4; m = A to D)

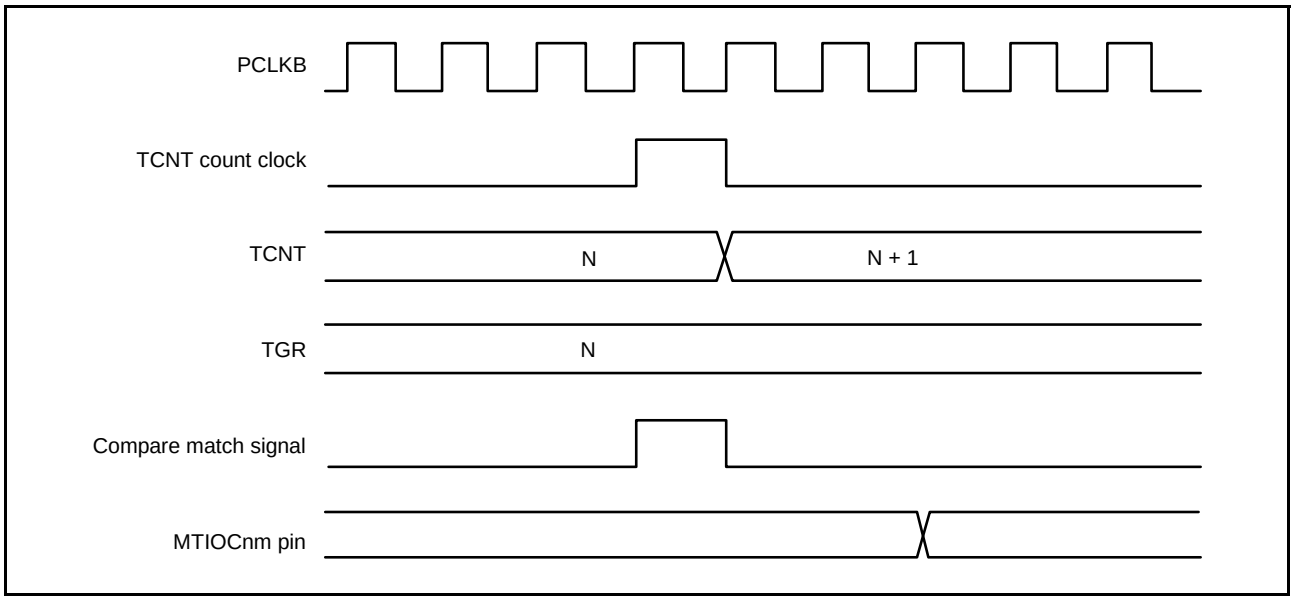


Figure 19.105 Output Compare Output Timing (Complementary PWM Mode or Reset-Synchronized PWM Mode) (n = 0 to 4; m = A to D)

(3) Input Capture Signal Timing

Figure 19.106 shows the input capture signal timing.

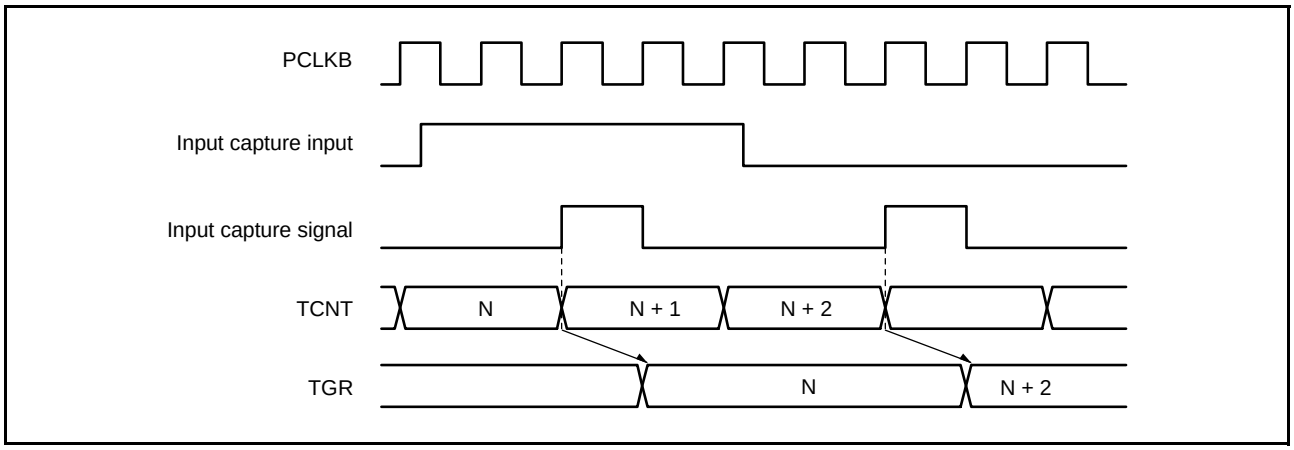


Figure 19.106 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

FFigure 19.107 and Figure 19.108 show the timing when counter clearing on compare match is specified, and Figure 19.109 shows the timing when counter clearing on input capture is specified.

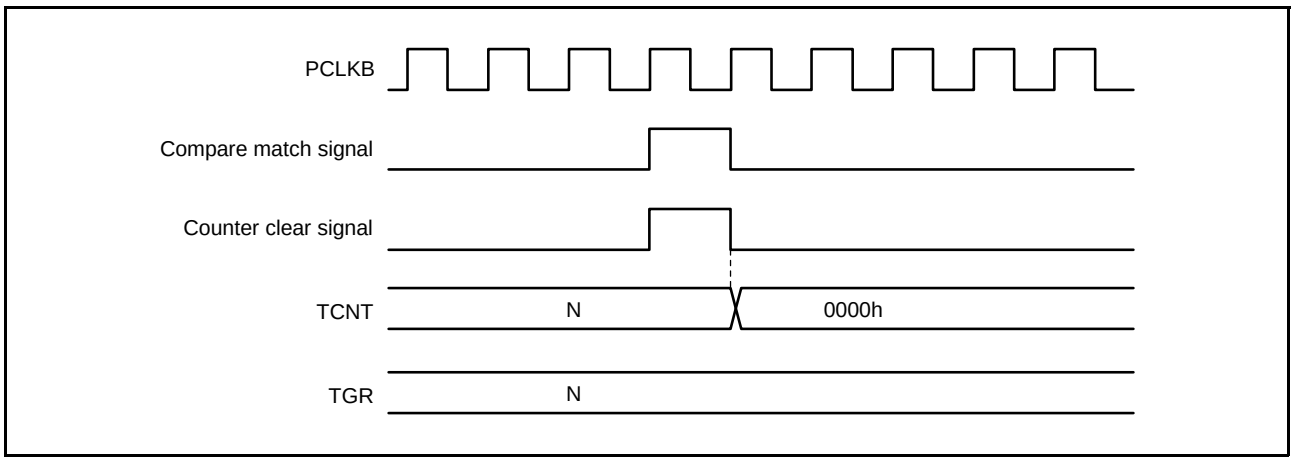


Figure 19.107 Counter Clear Timing (Compare Match) (MTU0 to MTU4)

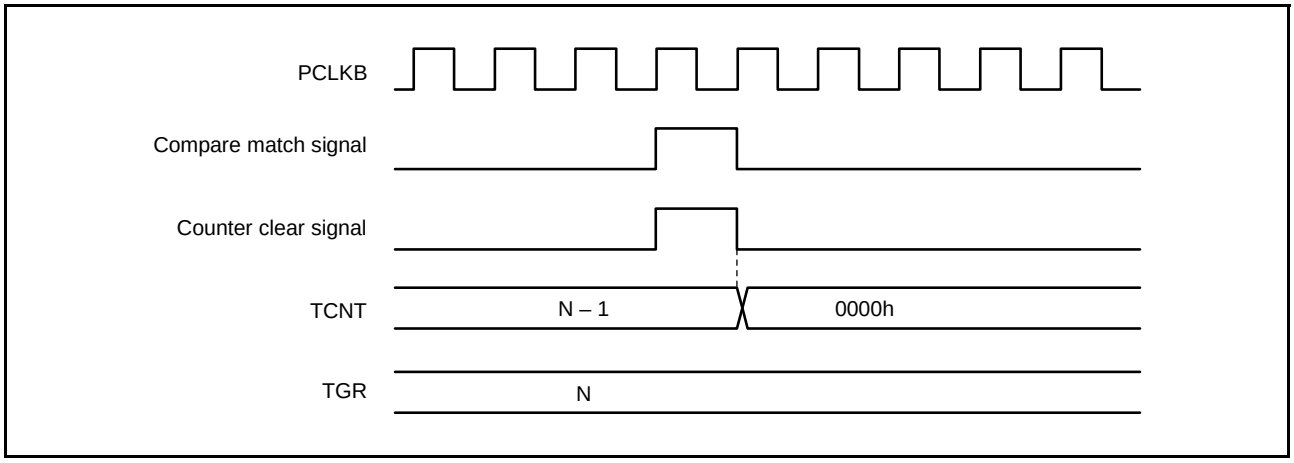


Figure 19.108 Counter Clear Timing (Compare Match) (MTU5)

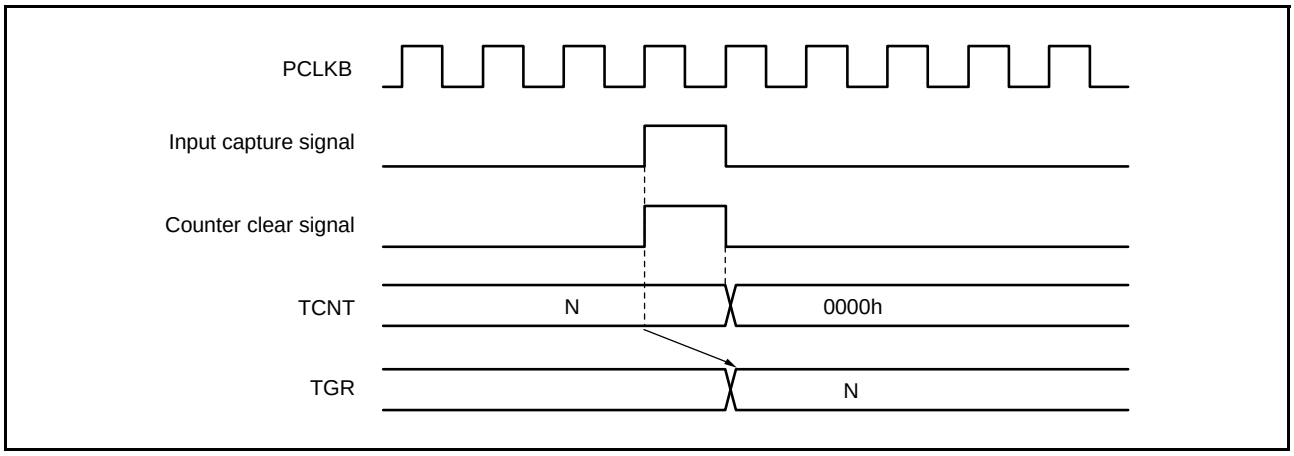


Figure 19.109 Counter Clear Timing (Input Capture) (MTU0 to MTU5)

(5) Buffer Operation Timing

Figure 19.110 to Figure 19.112 show the timing in buffer operation.

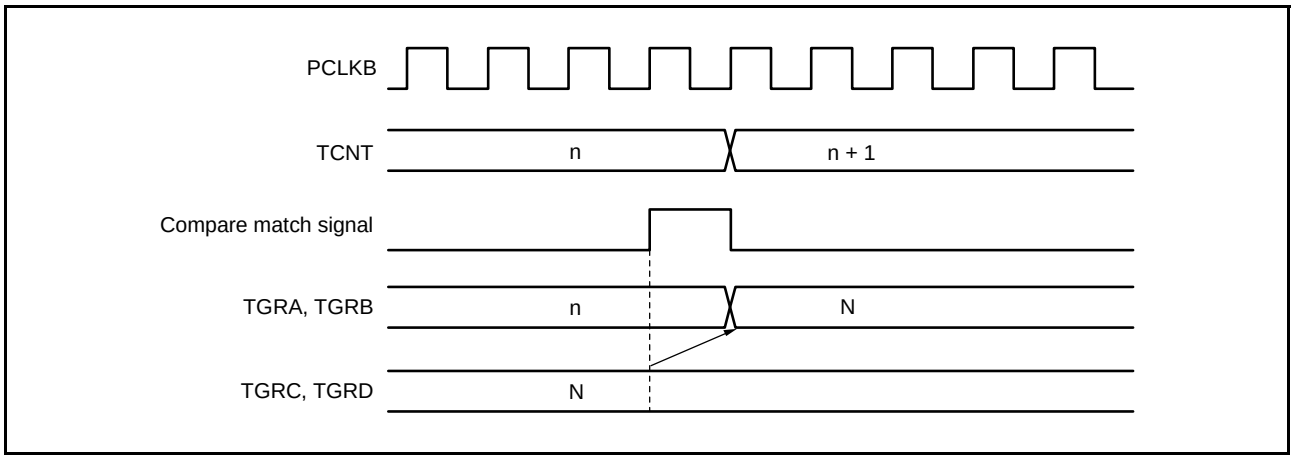


Figure 19.110 Buffer Operation Timing (Compare Match)

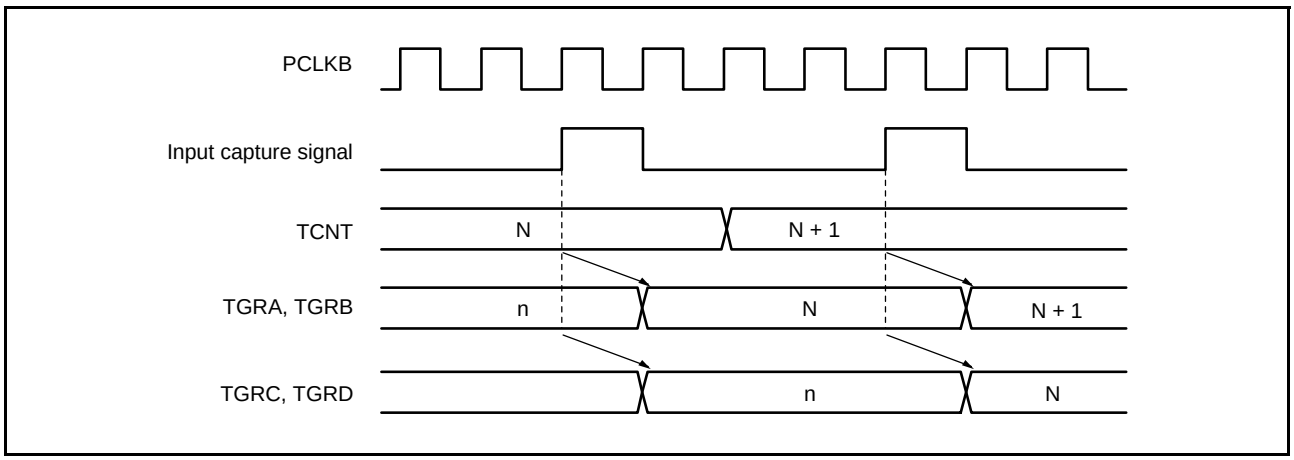


Figure 19.111 Buffer Operation Timing (Input Capture)

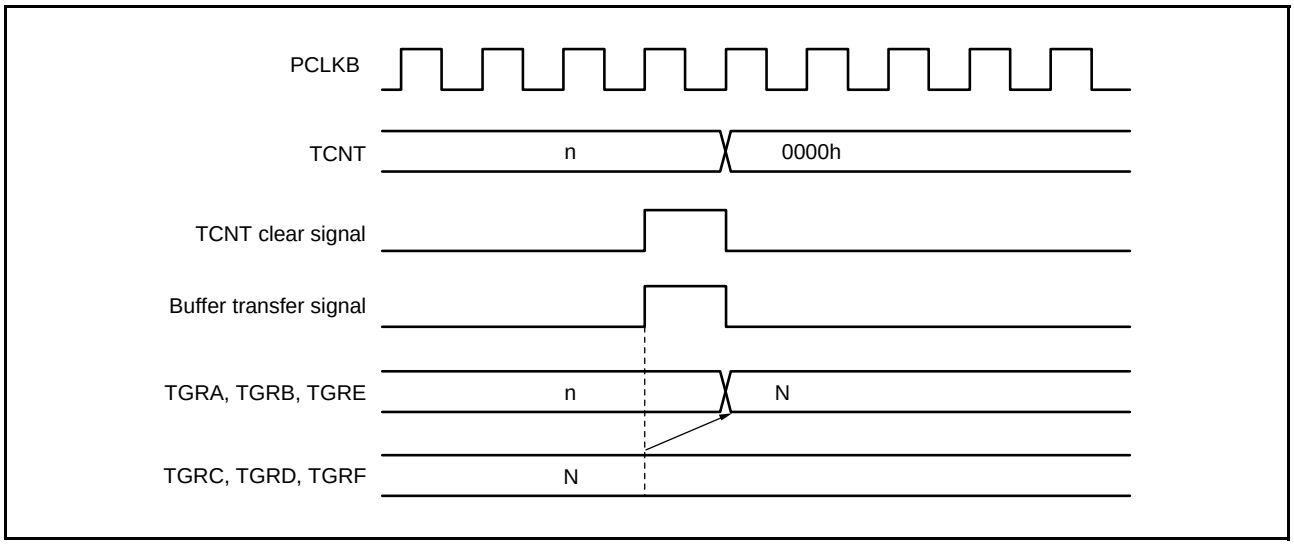


Figure 19.112 Buffer Operation Timing (When TCNT Cleared)

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figure 19.113 to Figure 19.115 show the buffer transfer timing in complementary PWM mode.

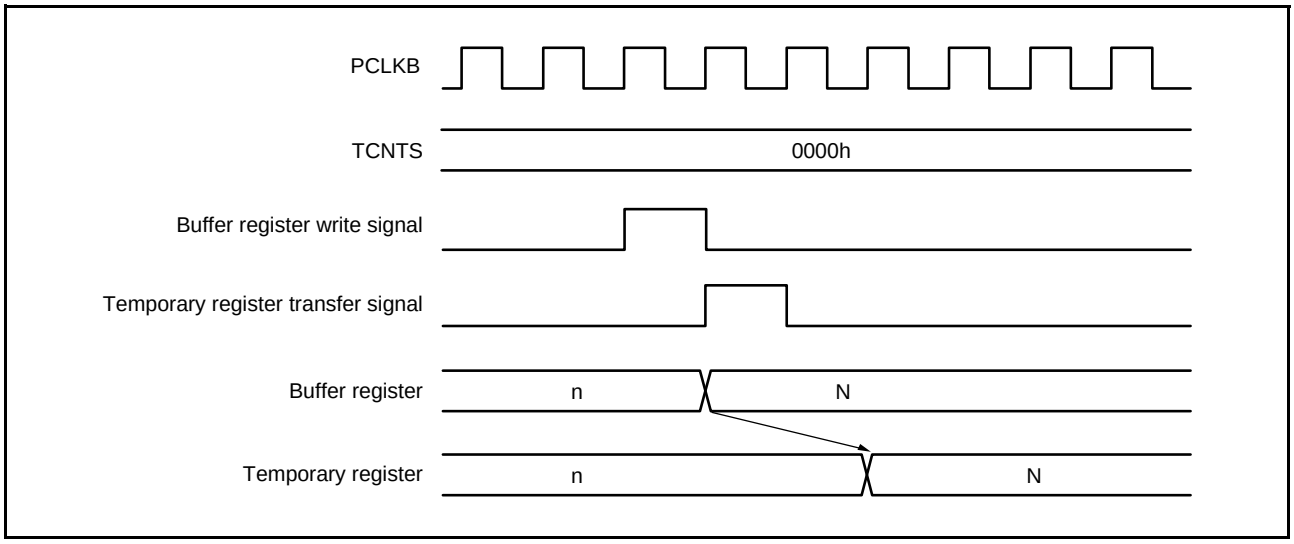


Figure 19.113 Transfer Timing from Buffer Register to Temporary Register (TCNTSA Stopped)

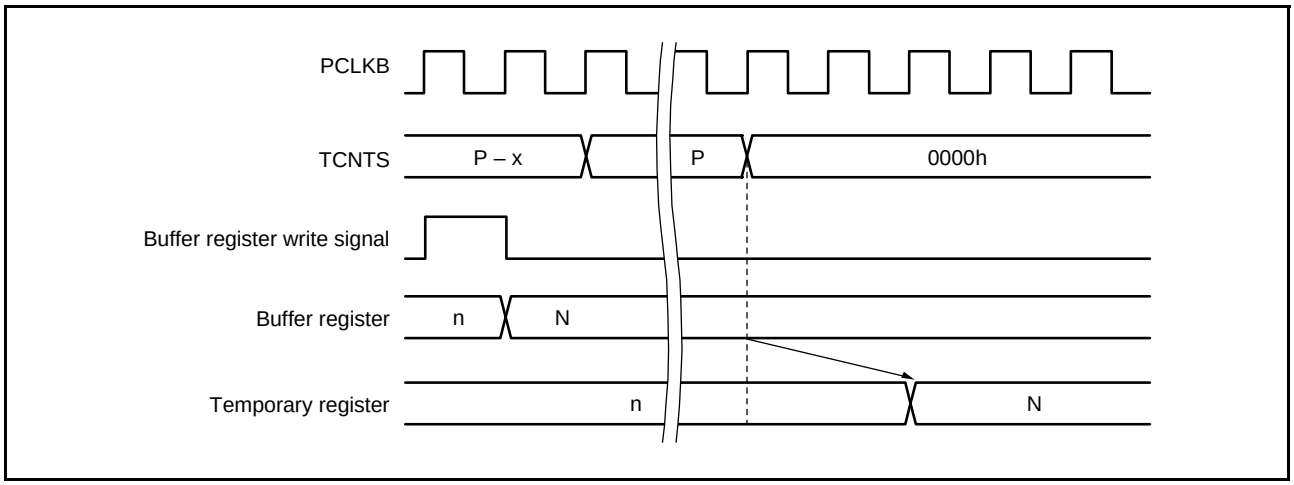


Figure 19.114 Transfer Timing from Buffer Register to Temporary Register (TCNTSA Operating)

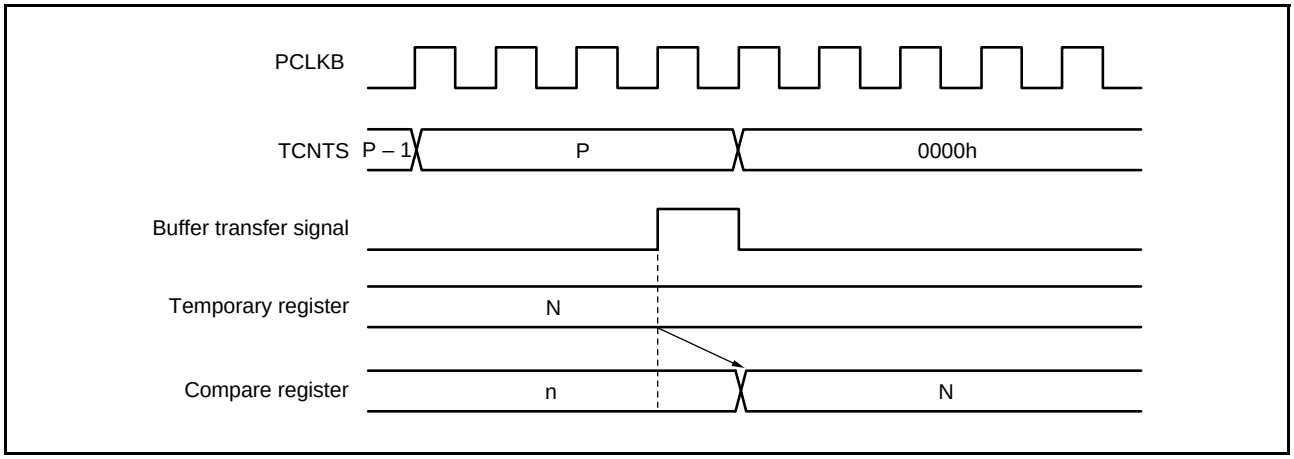


Figure 19.115 Transfer Timing from Temporary Register to Compare Register

19.5.2 Interrupt Signal Timing

(1) TGI Interrupt Timing by Compare Match

Figure 19.116 and Figure 19.117 show the TGI interrupt request signal timing when a compare match occurs.

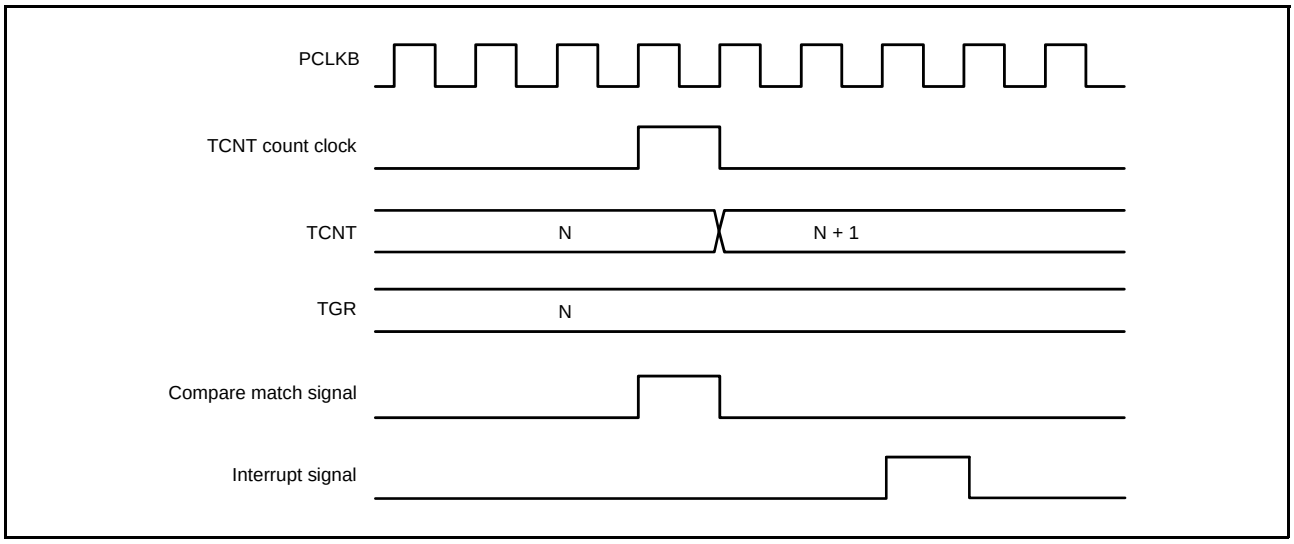


Figure 19.116 TGI Interrupt Timing (Compare Match) (MTU0 to MTU4)

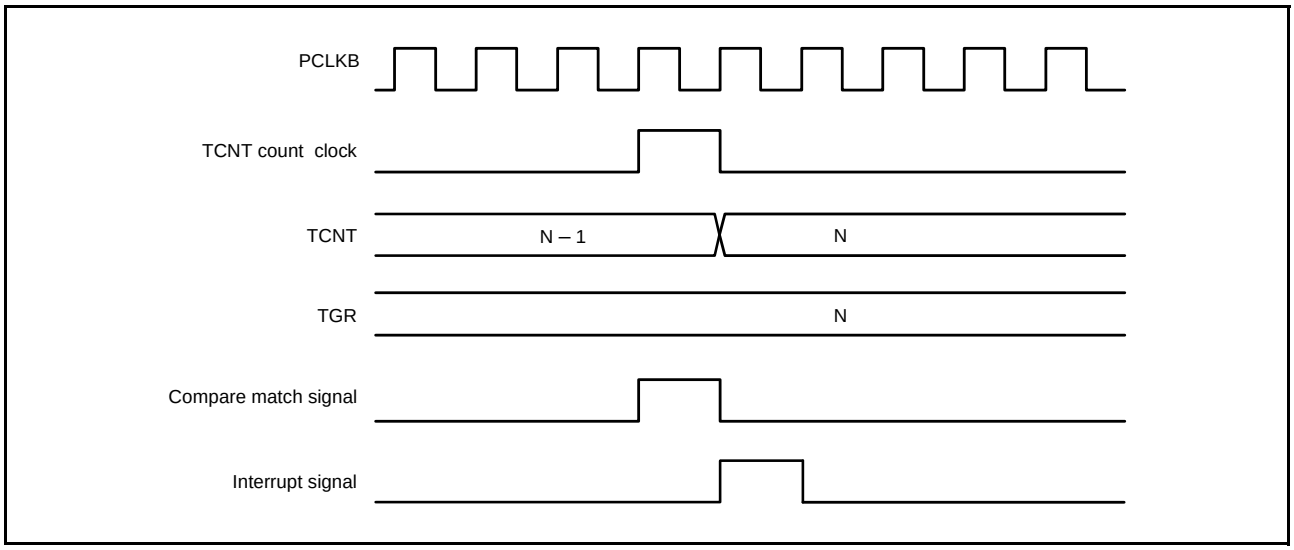


Figure 19.117 TGI Interrupt Timing (Compare Match) (MTU5)

(2) TGI Interrupt Timing by Input Capture

Figure 19.118 and Figure 19.119 show the TGI interrupt request signal timing when an input capture occurs.

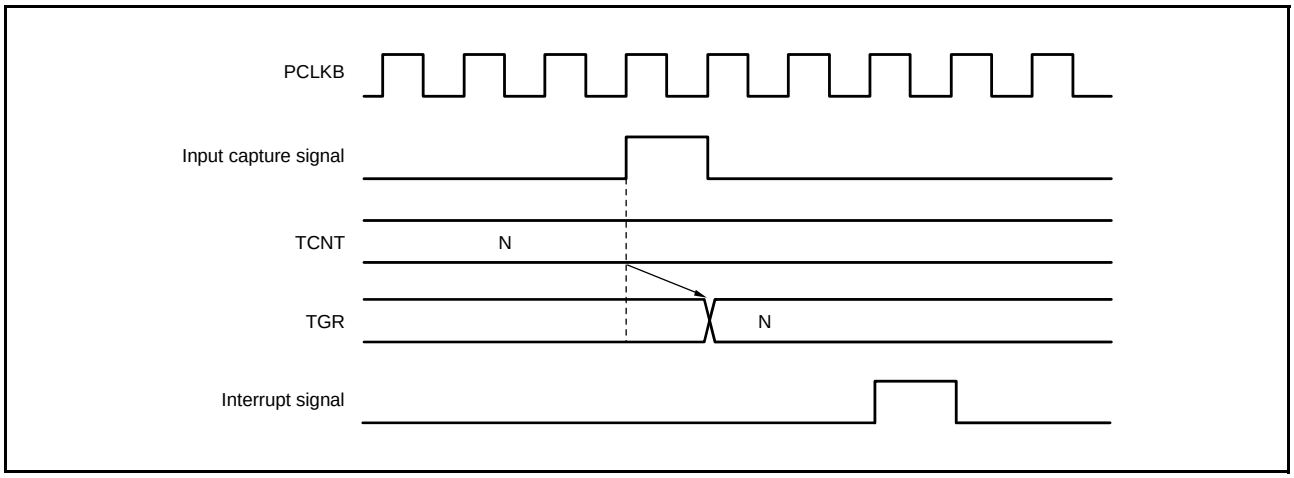


Figure 19.118 TGI Interrupt Timing (Input Capture) (MTU0 to MTU4)

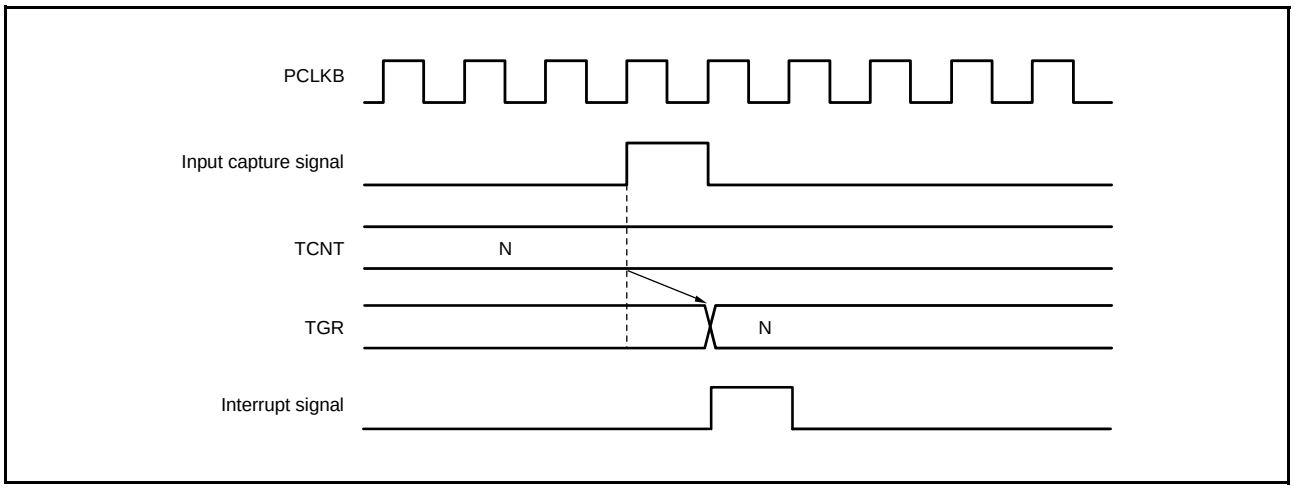


Figure 19.119 TGI Interrupt Timing (Input Capture) (MTU5)

(3) TCIV and TCIU Interrupt Timing

Figure 19.120 shows the TCIV interrupt request signal timing when an overflow is generated.
Figure 19.121 shows the TCIU interrupt request signal timing when an underflow is generated.

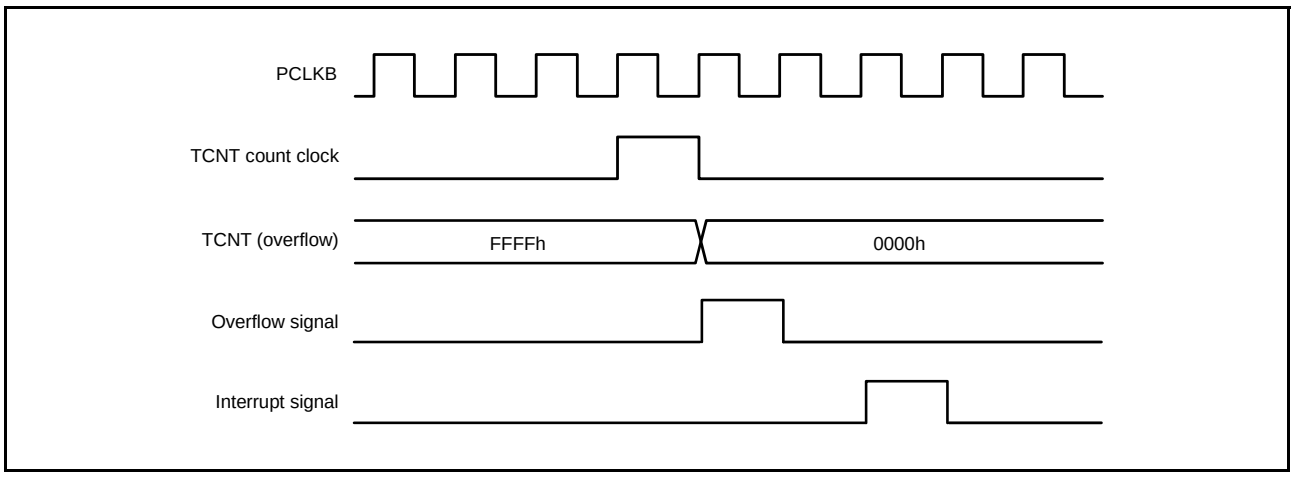


Figure 19.120 TCIV Interrupt Timing

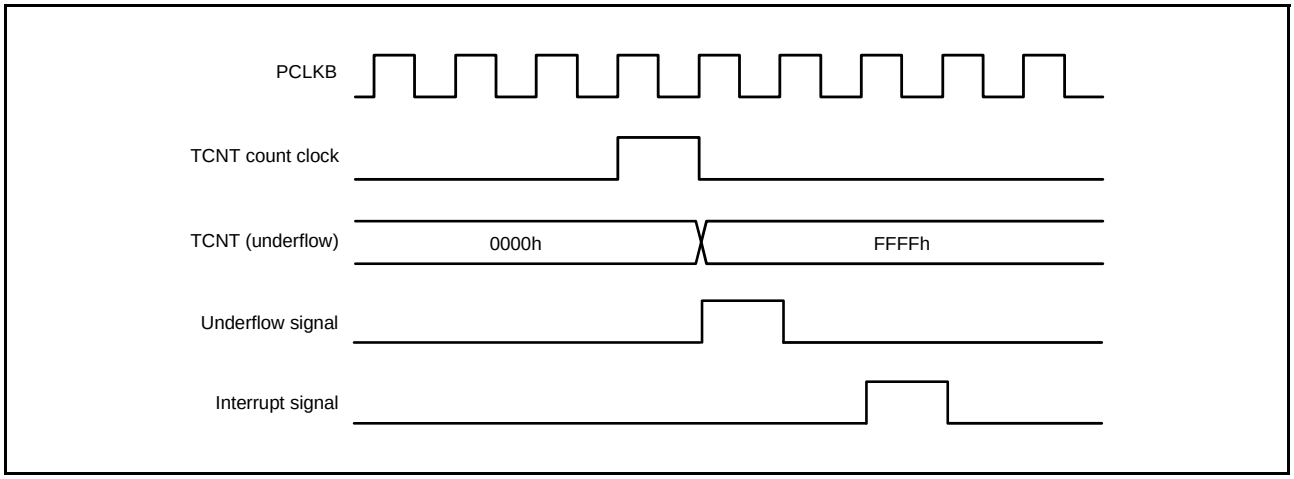


Figure 19.121 TCIU Interrupt Timing

19.6 Usage Notes

19.6.1 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, refer to section 11, Low Power Consumption.

19.6.2 Count Clock Restrictions

The count clock source pulse width must be at least three PCLKB cycles for single-edge detection, and at least five PCLKB 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 PCLKB cycles, and the pulse width must be at least five PCLKB cycles. Figure 19.122 shows the input clock conditions in phase counting mode.

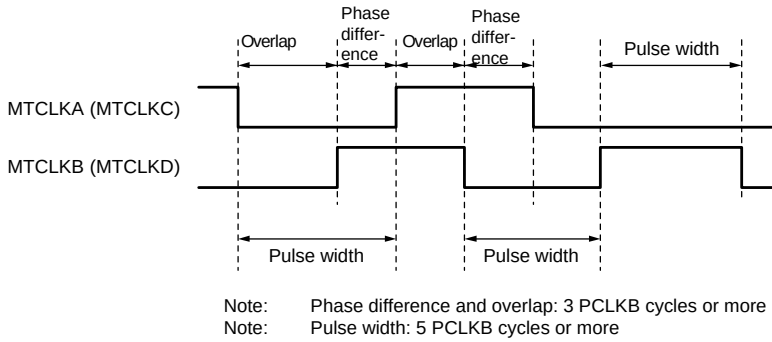


Figure 19.122 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

19.6.3 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

$$f = \frac{CNTCLK}{N + 1}$$

- MTU5

$$f = \frac{CNTCLK}{N}$$

f: Counter frequency
CNTCLK: The count clock frequency set by TCR.TPSC[2:0] and TCR2.TPSC2[2:0]
N: TGR setting

19.6.4 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 19.123 shows the timing in this case.

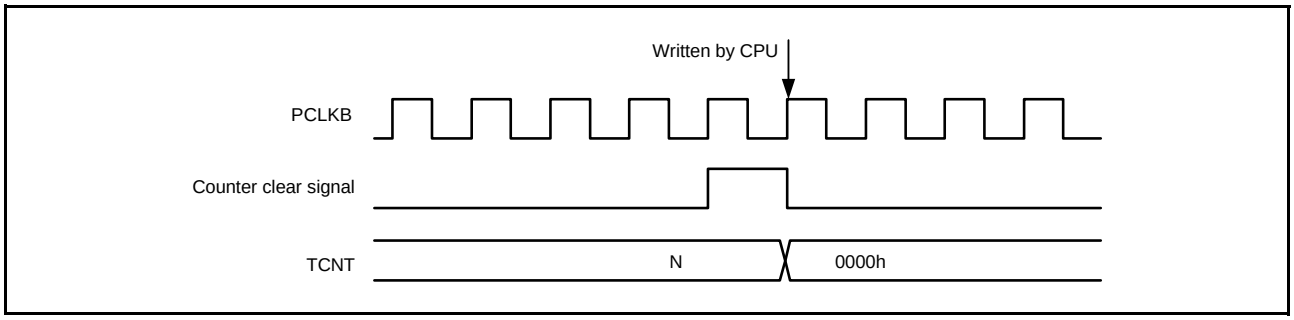


Figure 19.123 Contention between TCNT Write and Clear Operations

19.6.5 Contention between TCNT Write and Increment Operations

If incrementing occurs in a TCNT write cycle, TCNT write operation takes precedence and TCNT is not incremented.
Figure 19.124 shows the timing in this case.

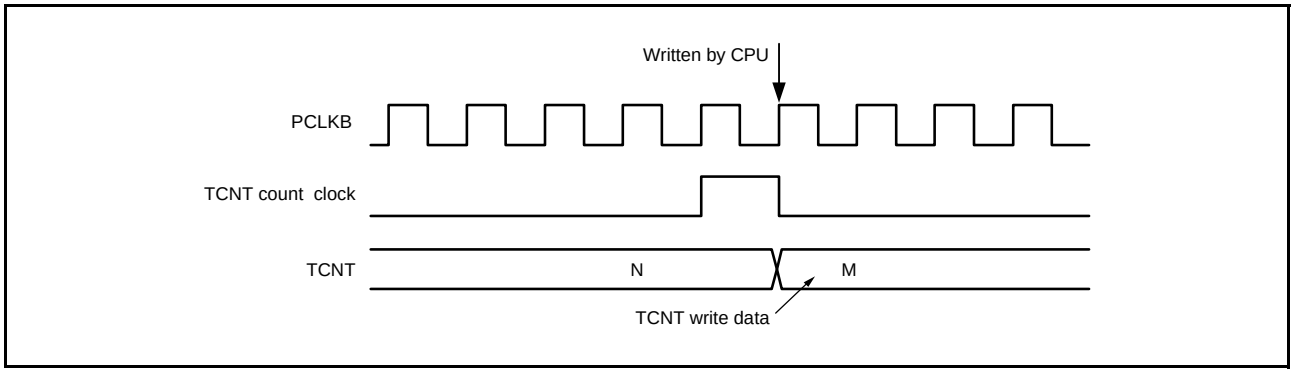


Figure 19.124 Contention between TCNT Write and Increment Operations

19.6.6 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 19.125 shows the timing in this case.

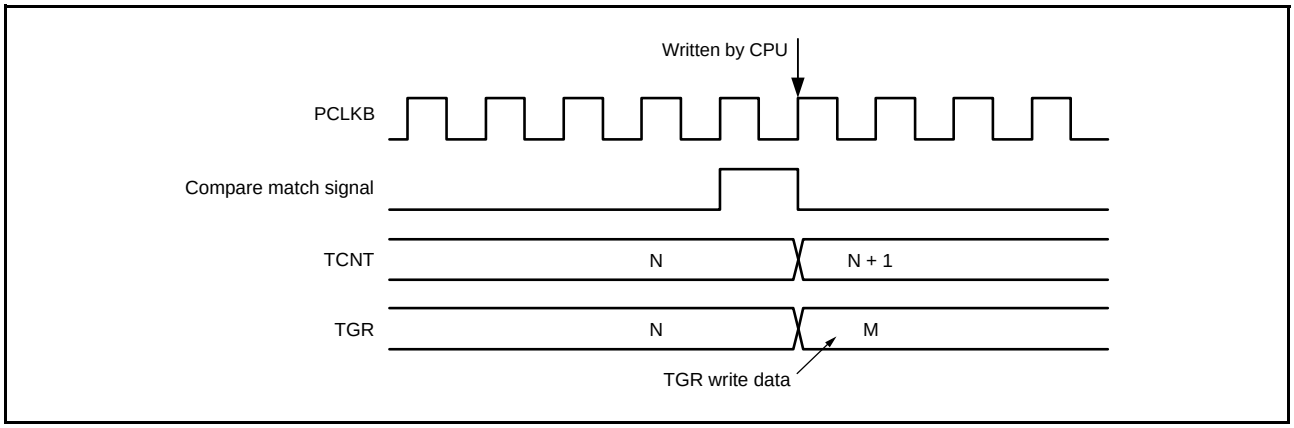


Figure 19.125 Contention between TGR Write Operation and Compare Match

19.6.7 Contention between Buffer Register Write Operation and Compare Match

If a compare match occurs in the T2 state in a TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.
Figure 19.126 shows the timing in this case.

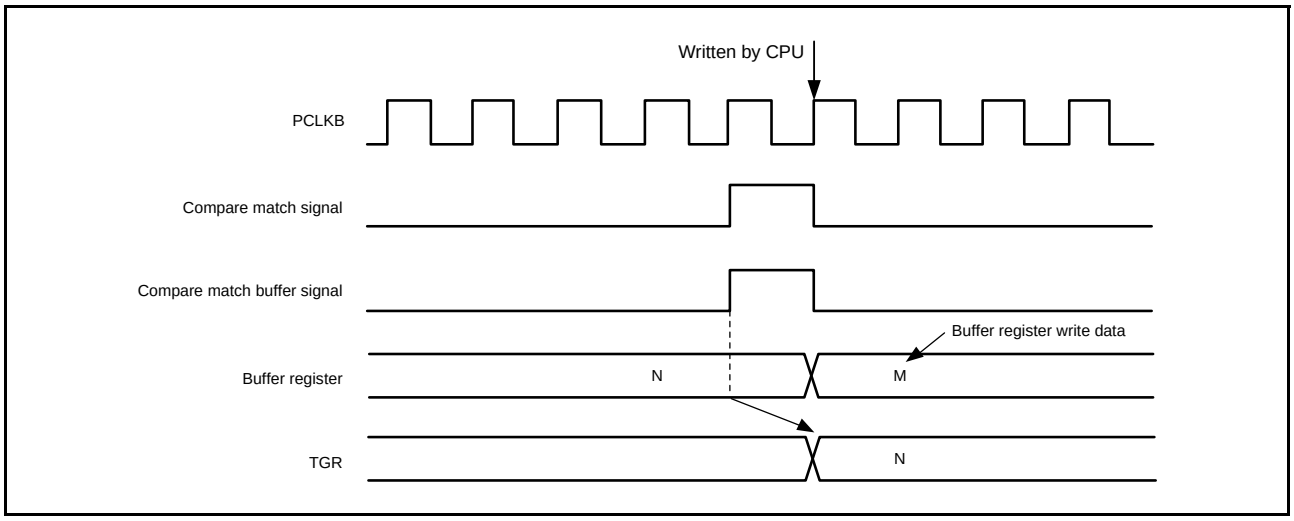


Figure 19.126 Contention between Buffer Register Write Operation and Compare Match

19.6.8 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 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 19.127 shows the timing in this case.

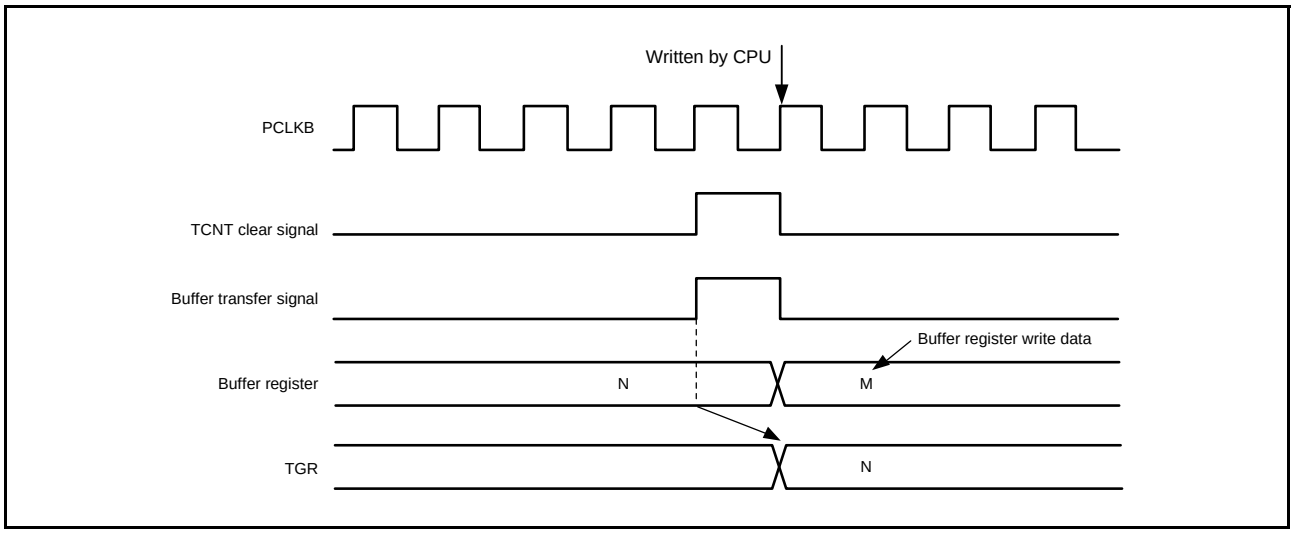


Figure 19.127 Contention between Buffer Register Write and TCNT Clear Operations

19.6.9 Contention between TGR Read Operation and Input Capture

If an input capture signal is generated in a TGR read cycle, the data before input capture transfer is read.
Figure 19.128 shows the timing in this case.

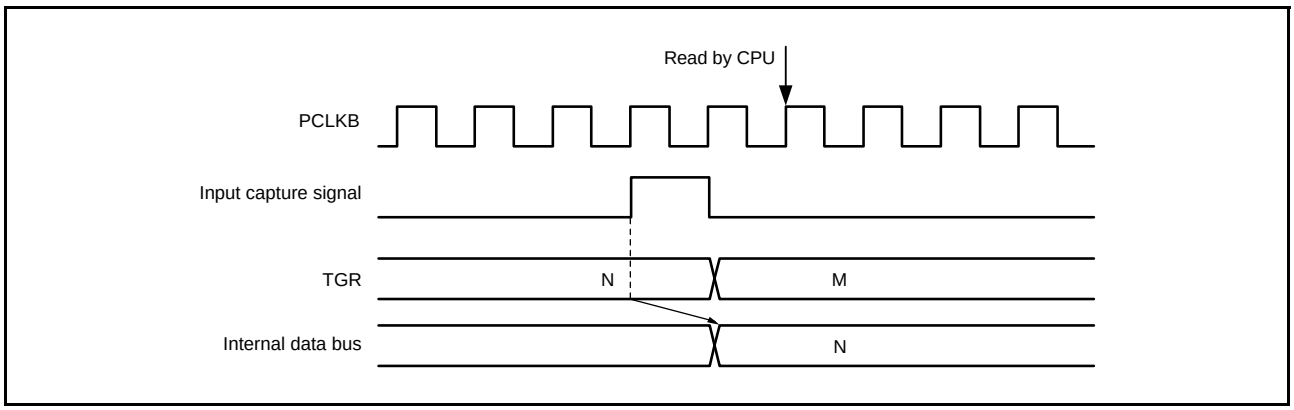


Figure 19.128 Contention between TGR Read Operation and Input Capture (MTU0 to MTU5)

19.6.10 Contention between TGR Write Operation and Input Capture

If an input capture signal is generated in the TGR write cycle, the input capture operation takes precedence and the TGR write operation is not performed in MTU0 to MTU4. In MTU5, the TGR write operation is performed and the input capture signal is generated.

Figure 19.129 and Figure 19.130 show the timing in this case.

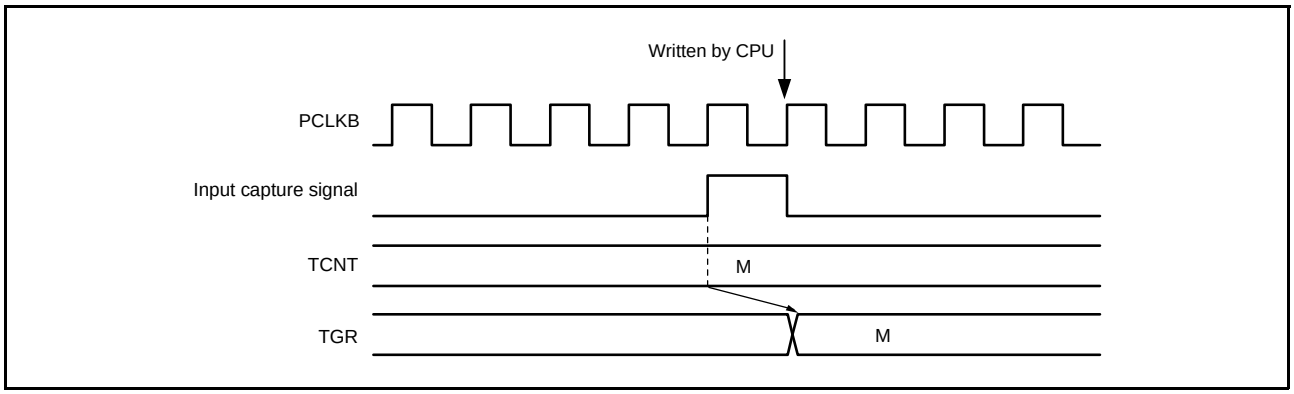


Figure 19.129 Contention between TGR Write Operation and Input Capture (MTU0 to MTU4)

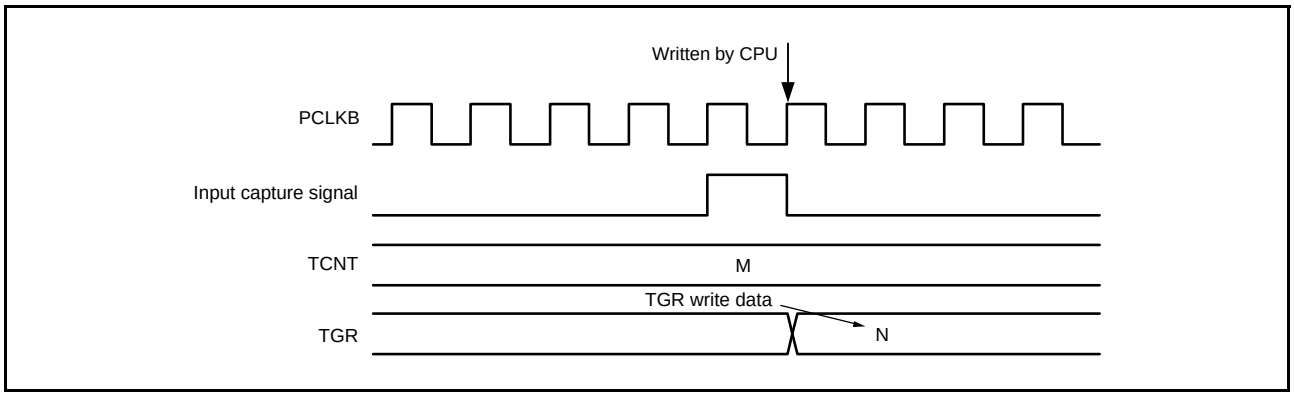


Figure 19.130 Contention between TGR Write Operation and Input Capture (MTU5)

19.6.11 Contention between Buffer Register Write Operation and Input Capture

If an input capture signal is generated in the buffer register write cycle, the buffer operation takes precedence and the buffer register write operation is not performed.
Figure 19.131 shows the timing in this case.

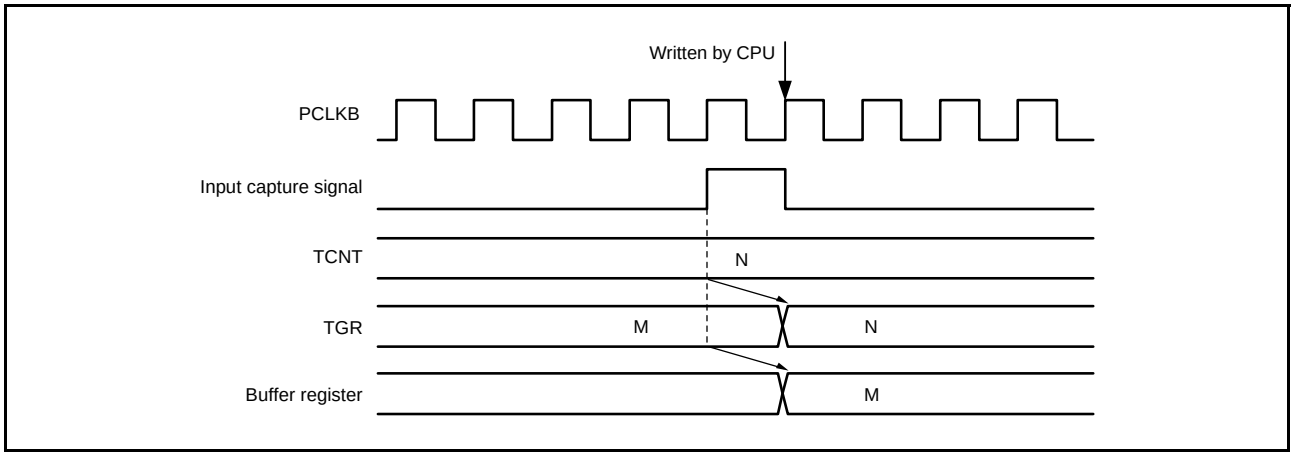


Figure 19.131 Contention between Buffer Register Write Operation and Input Capture

19.6.12 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 19.132 shows the timing in this case.

When setting the TCNT clearing function in cascaded operation, be sure to synchronize MTU1 and MTU2.

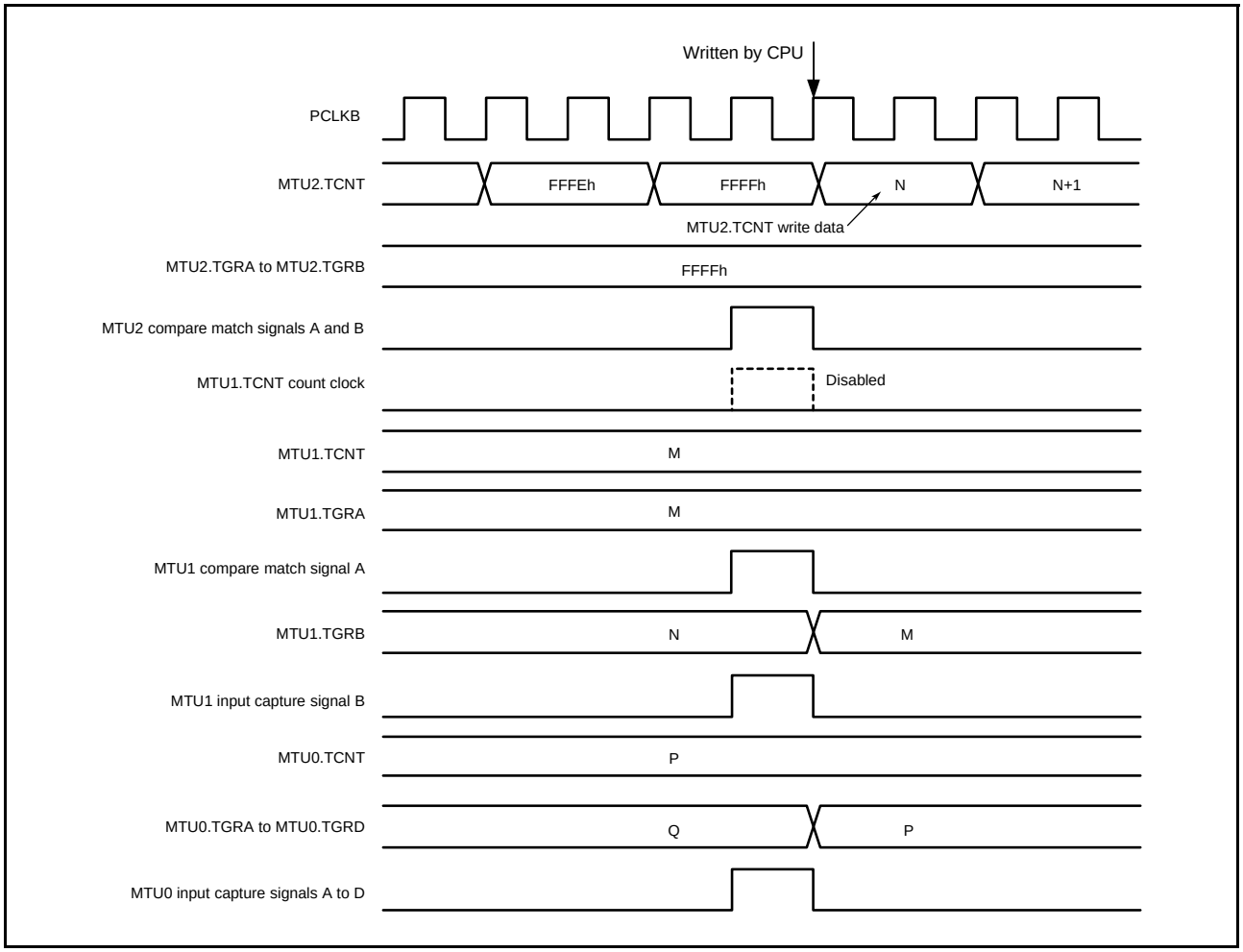


Figure 19.132 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

19.6.13 Counter Value When Count Operation is Stopped in Complementary PWM Mode

When counting operation in MTU3.TCNT and MTU4.TCNT is stopped in complementary PWM mode, the MTU3.TCNT value is set to the timer dead time register (TDDRA) value and MTU4.TCNT is set to 0000h.
When operation is restarted in complementary PWM mode, counting begins automatically from the initial setting state. Figure 19.133 shows this operation.
When counting begins in another operating mode, be sure to make initial settings in MTU3.TCNT and MTU4.TCNT.

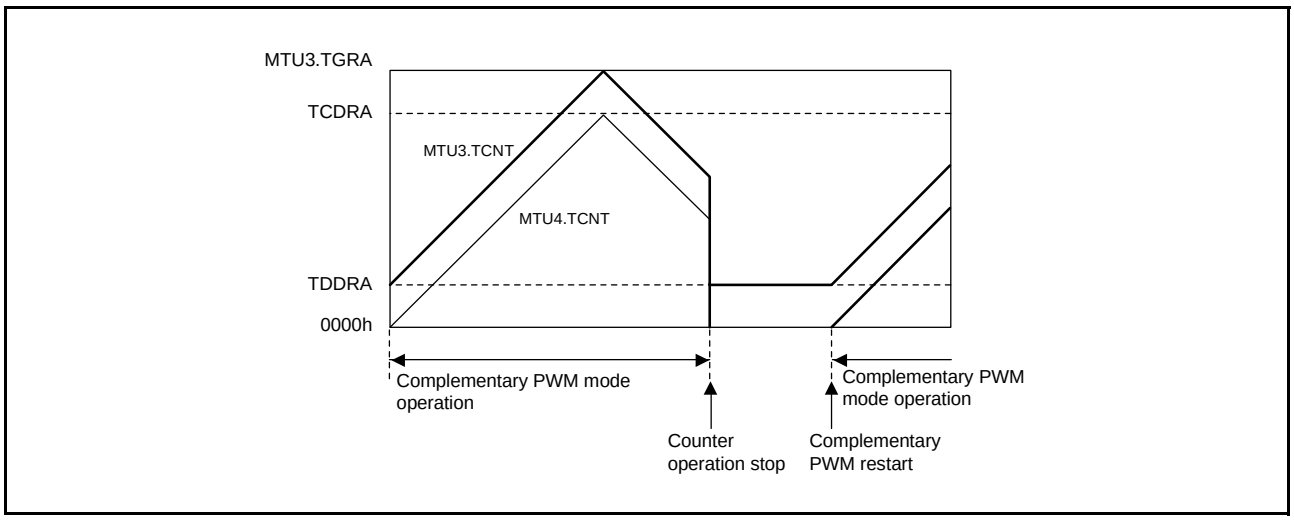


Figure 19.133 Counter Value When Stopped in Complementary PWM Mode

19.6.14 Buffer Operation Setting in Complementary PWM Mode

When modifying the PWM period set register (MTU3.TGRA), timer period data register (TCDRA), and duty set registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) in complementary PWM mode, be sure to use buffer operation. In addition, set the BFA and BFB bits in MTU4.TMDR1 to 0. The MTIOC4C pin cannot output waveforms if the BFA bit in MTU4.TMDR1 is set to 1. Likewise, the MTIOC4D pin cannot output waveforms if the BFB bit in MTU4.TMDR1 is set to 1.
In complementary PWM mode, buffer operation in MTU3 and MTU4 depends on the settings in bits BFA and BFB of MTU3.TMDR1. When the BFA bit in MTU3.TMDR1 is set to 1, MTU3.TGRC functions as a buffer register for MTU3.TGRA. At the same time, MTU4.TGRC functions as a buffer register for MTU4.TGRA, and TCBRA functions as a buffer register for TCDRA.

19.6.15 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 to 0. The MTIOC4C pin cannot output waveforms if the BFA bit in MTU4.TMDR1 is set to 1. Likewise, the MTIOC4D pin cannot output waveforms if the BFB bit in MTU4.TMDR1 is set to 1.

In reset-synchronized PWM mode, buffer operation in MTU3 and MTU4 depends on the settings in the BFA and BFB bits of MTU3.TMDR1. For example, if the BFA bit in MTU3.TMDR1 is set to 1, MTU3.TGRC functions as a buffer register for MTU3.TGRA. At the same time, MTU4.TGRC functions as a buffer register for MTU4.TGRA.

While the MTU3.TGRC and MTU3.TGRD are operating as buffer registers, a TGImn interrupt (m = C, D; n = 3, 4) is not generated.

Figure 19.134 shows an example of MTU3.TGR, MTU4.TGR, MTIOC3, and MTIOC4 operation with the BFA and BFB bits in MTU3.TMDR1 set to 1 and the BFA and BFB bits in MTU4.TMDR1 set to 0.

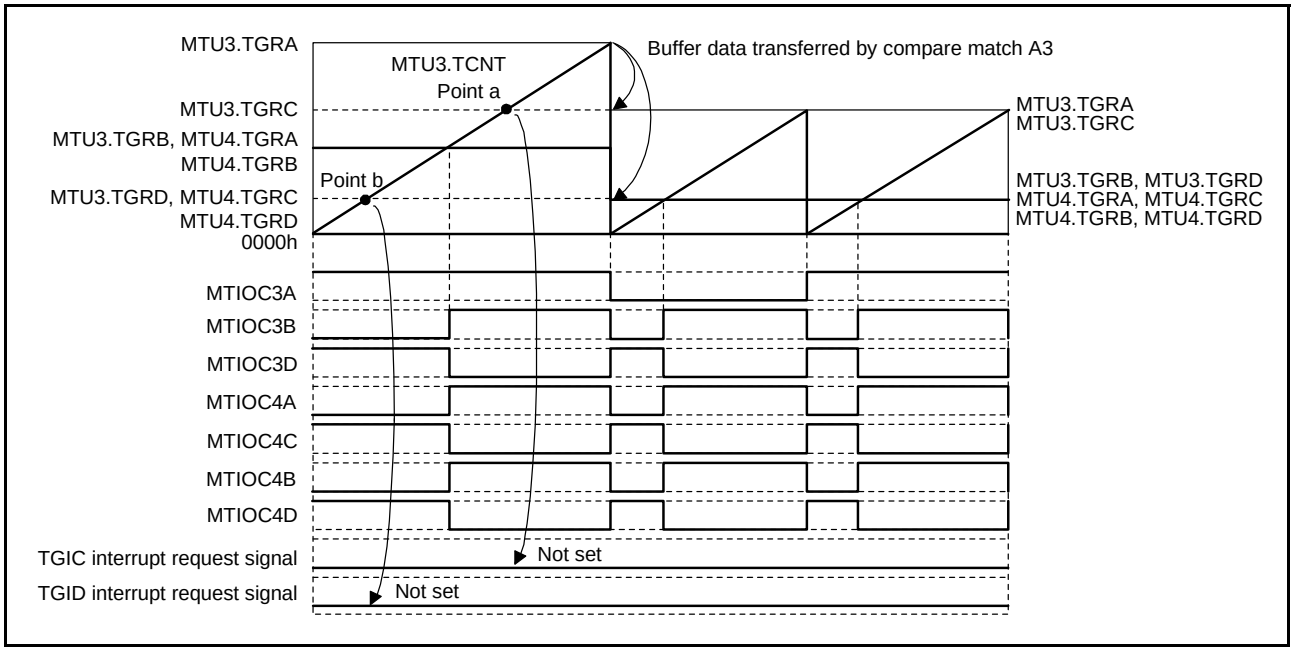


Figure 19.134 Buffer Operation and Compare Match in Reset-Synchronized PWM Mode

19.6.16 Overflow in Reset-Synchronized PWM Mode

After reset-synchronized PWM mode is selected, MTU3.TCNT and MTU4.TCNT start counting when the CST3 bit of TSTRA is set to 1. In this state, the MTU4.TCNT count clock source and count edge are determined by the MTU3.TCR setting.

In reset-synchronized PWM mode, with period register MTU3.TGRA set to FFFFh and the MTU3.TGRA compare match selected as the counter clearing source, MTU3.TCNT and MTU4.TCNT count up to FFFFh, then a compare match occurs with MTU3.TGRA, and MTU3.TCNT and MTU4.TCNT are both cleared. In this case, a TCIVn interrupt (n = 3, 4) is not generated.

Figure 19.135 shows an example of operation in reset-synchronized PWM mode with period register MTU3.TGRA set to FFFFh and the MTU3.TGRA compare match specified for the counter clearing source.

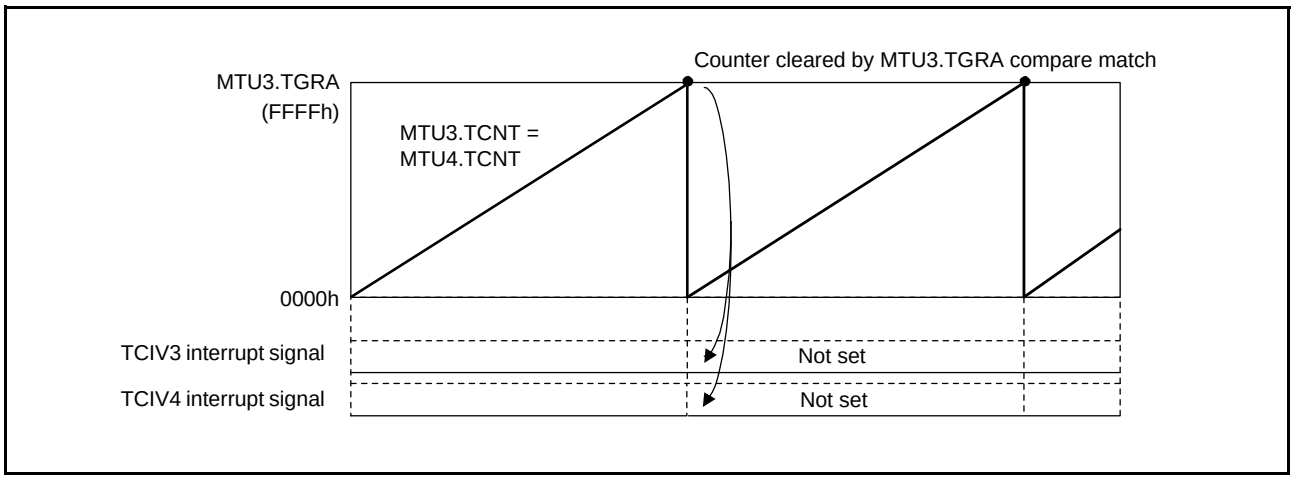


Figure 19.135 Overflow in Reset-Synchronized PWM Mode

19.6.17 Contention between Overflow/Underflow and Counter Clearing

If an overflow/underflow and counter clearing occur simultaneously, a TCIVn interrupt (n = 0 to 4) nor a TCIUn interrupt (n = 1, 2) is not generated and TCNT clearing takes precedence.
Figure 19.136 shows the operation timing when a TGR compare match is specified as the clearing source and TGR is set to FFFFh.

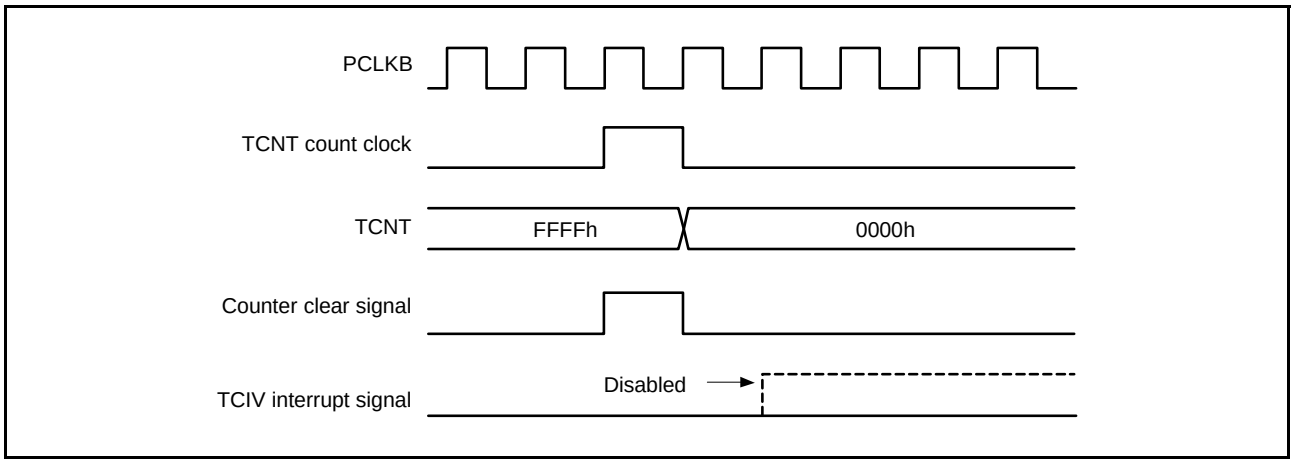


Figure 19.136 Contention between Overflow and Counter Clearing

19.6.18 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) nor a TCIUn interrupt (n = 1, 2) is not generated.
Figure 19.137 shows the operation timing when there is contention between TCNT write operation and overflow.

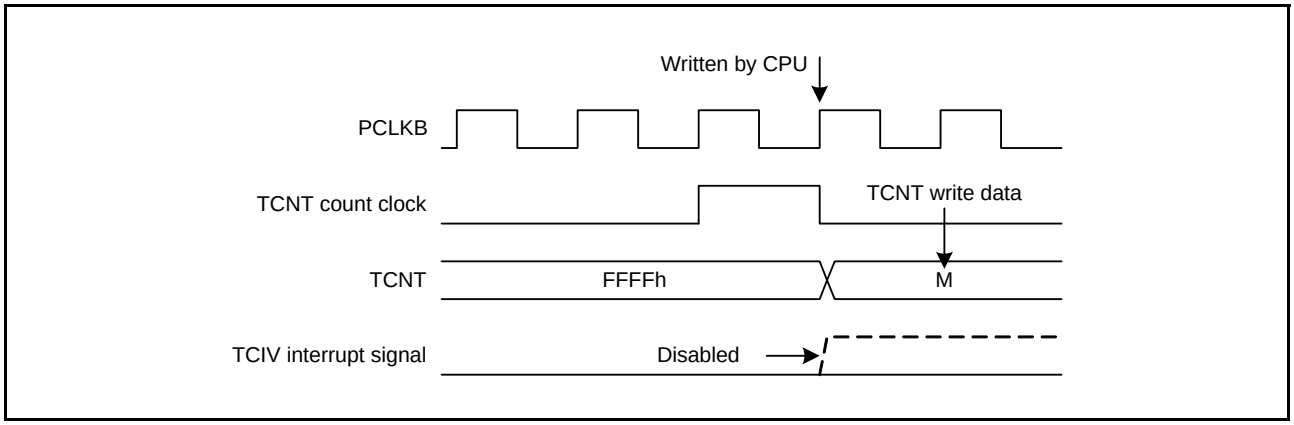


Figure 19.137 Contention between TCNT Write Operation and Overflow

19.6.19 Note on Transition from Normal Mode or PWM Mode 1 to Reset-Synchronized PWM Mode

When making a transition from normal mode or PWM mode 1 to reset-synchronized PWM mode in MTU3 and MTU4, if the counter is stopped while the output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B and MTIOC4D) 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 11h to MTU3.TIORH, MTU3.TIORL, MTU4.TIORH, and MTU4.TIORL to initialize the output pin state to a low level, then set the registers to the initial value (00h) 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 (00h) before making the transition to reset-synchronized PWM mode.

19.6.20 Output Level in Complementary PWM Mode and Reset-Synchronized PWM Mode

When MTU3 and MTU4 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). In complementary PWM mode or reset-synchronized PWM mode, TIOR should be set to 00h.
The output level in negative phase when the TDERA.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 bit. It is equivalent to the inverted level of positive phase output based on the setting of the TOCR1A.OLSP bit.

19.6.21 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 = FFF1h and MTU2.TCNT = 0000h should be transferred to MTU1.TGRA and MTU2.TGRA or to MTU1.TGRB and MTU2.TGRB, but the values of MTU1.TCNT = FFF0h and MTU2.TCNT = 0000h are erroneously transferred.
The MTU has a function that allows simultaneous capture of MTU1.TCNT and MTU2.TCNT with a single input capture input. 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, refer to section 19.2.11, Timer Input Capture Control Register (TICCR).

19.6.22 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.

(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 PCLKB (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 PCLKB (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 PCLKB (the updated value of MTU4.TADCORB is set to the previous value plus or minus at least two).

19.6.23 Notes When Complementary PWM Mode Output Protection Function is Not Used

The complementary PWM mode output protection function is initially enabled. For details, refer to section 20, Port Output Enable 3 (POE3C).

19.6.24 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 0000h when the compare-match is generated, regardless of whether interrupts from the MTU5.TCNTj (j = U, V, W) are enabled or disabled.

19.6.25 Notes to Prevent Malfunctions in Synchronous Clearing for Complementary PWM Mode

If control of the output waveform is enabled (TWCRA.WRE bit = 1) 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 19.138, 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 19.139, synchronous clearing occurs when any condition from among MTU3.TGRB ≤ TDDRA, MTU4.TGRA ≤ TDDRA, or MTU4.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, MTU4.TGRA, and MTU4.TGRB) set to at least double the value of the TDDRA register.

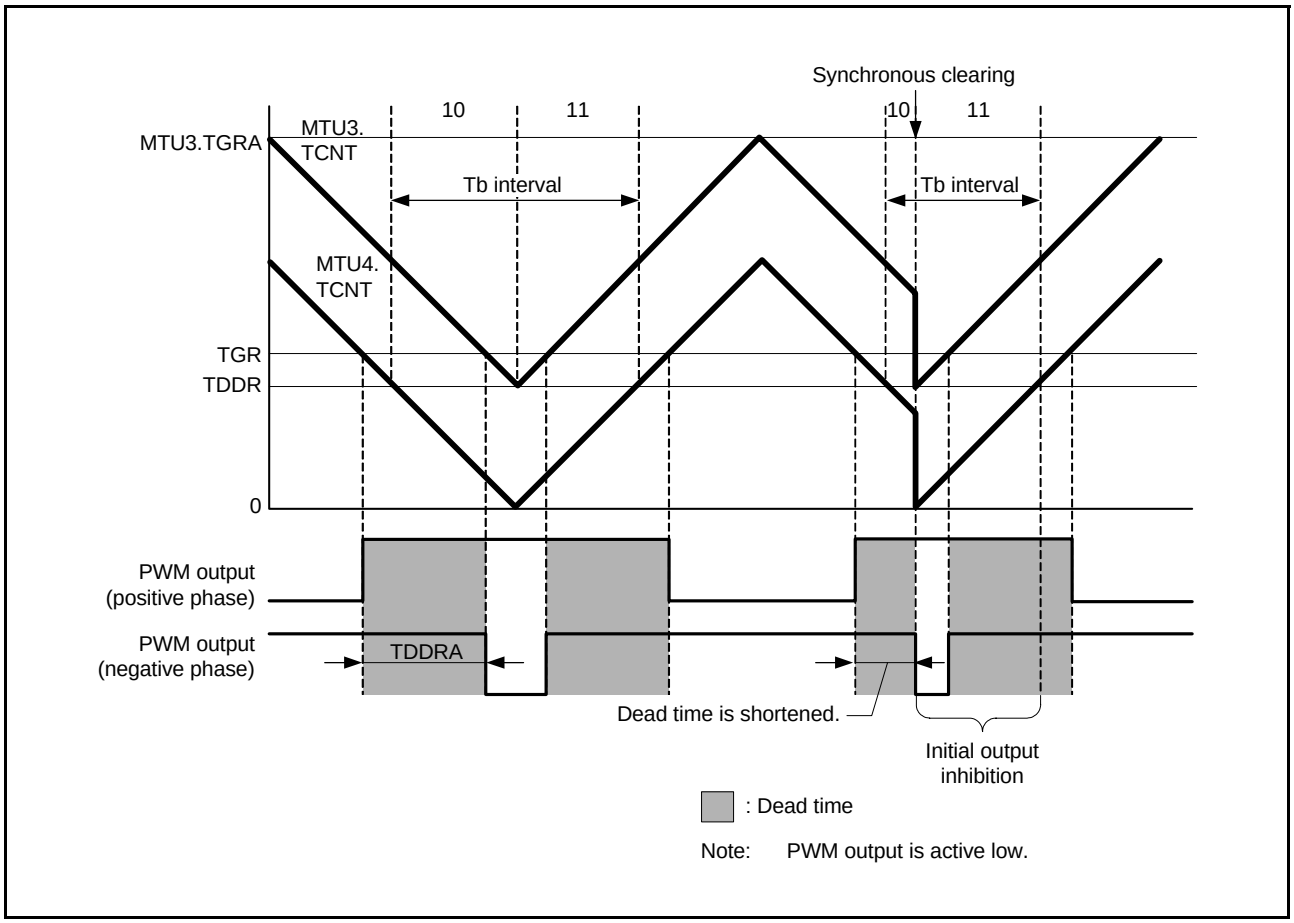


Figure 19.138 Example of Synchronous Clearing (When Condition 1 Applies)

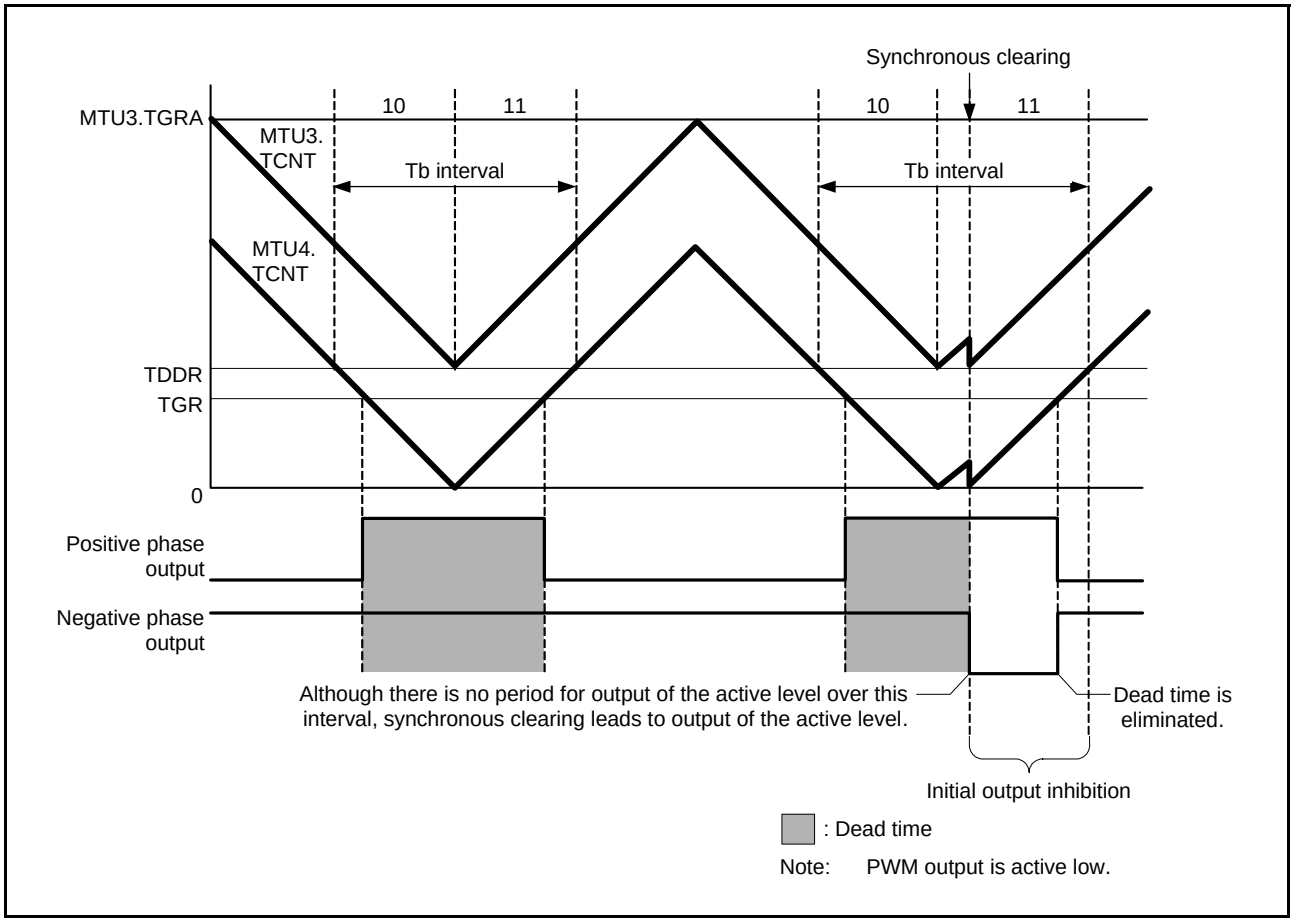


Figure 19.139 Example of Synchronous Clearing (When Condition 2 Applies)

19.6.26 Continuous Output of Interrupt Signal in Response to a Compare Match

When the TGR register is set to 0000h, the PCLKB/1 is set as the count clock, and compare match is set as the trigger for clearing of the count clock, the value of the TCNT counter remains 0000h, and the interrupt signal will be output continuously (i.e. its level will be flat) rather than output over a single cycle. Consequently, interrupts will not be detected in response to second and subsequent compare matches.
Figure 19.140 shows the timing for continuous output of the interrupt signal in response to a compare match.

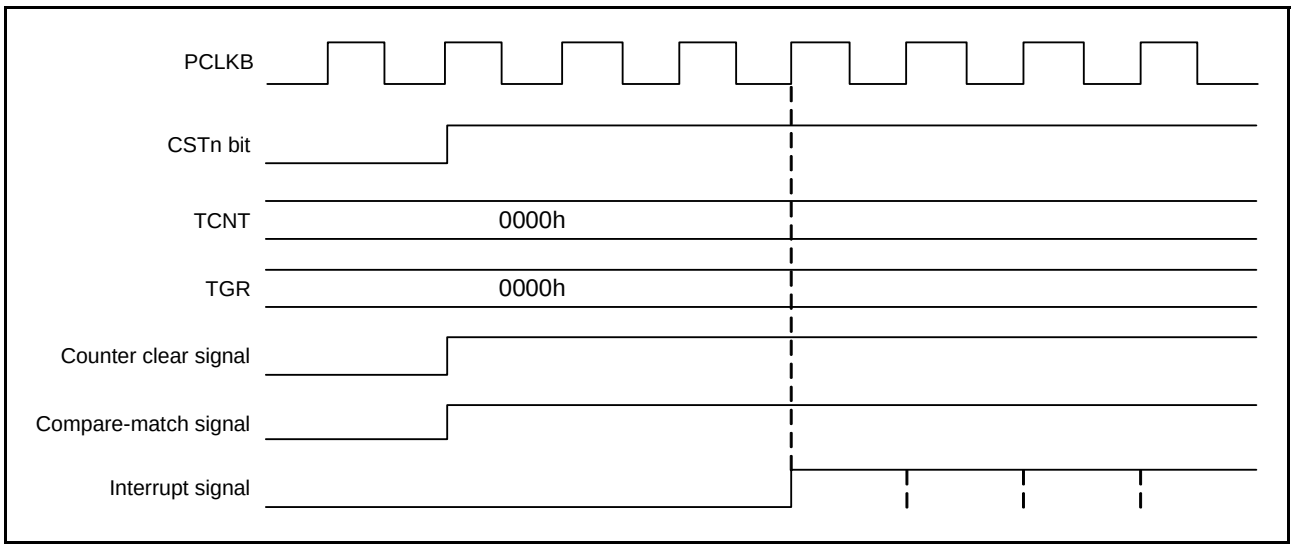


Figure 19.140 Continuous Output of Interrupt Signal in Response to a Compare Match

19.6.27 Usage Notes on A/D Conversion Start Request Delaying Function in Complementary PWM Mode

- When data is transferred from a buffer register at the trough of the MTU4.TCNT counter while the MTU4.TADCOBRA and MTU4.TADCOBRB registers are set to 0 and the UT4AE and UT4BE bits in the MTU4.TADCR register are set to 1, no A/D conversion start request is issued during up-counting immediately after transfer. Refer to Figure 19.141.
- When data is transferred from a buffer register at the crest of the MTU4.TCNT counter while the MTU4.TADCOBRA and MTU4.TADCOBRB registers are set to the same value as the TCDR and the UT4AE and UT4BE bits in the MTU4.TADCR register are set to 1, no A/D conversion start request is issued during down-counting immediately after transfer. Refer to Figure 19.142.
- To issue an A/D conversion start request linked with the interrupt skipping function, set the MTU4.TADCORA and MTU4.TADCORB registers so that $2 \leq$ MTUn.TADCORA/TADCORB $\leq$ TCDR $- 2$ is satisfied (n = 4).

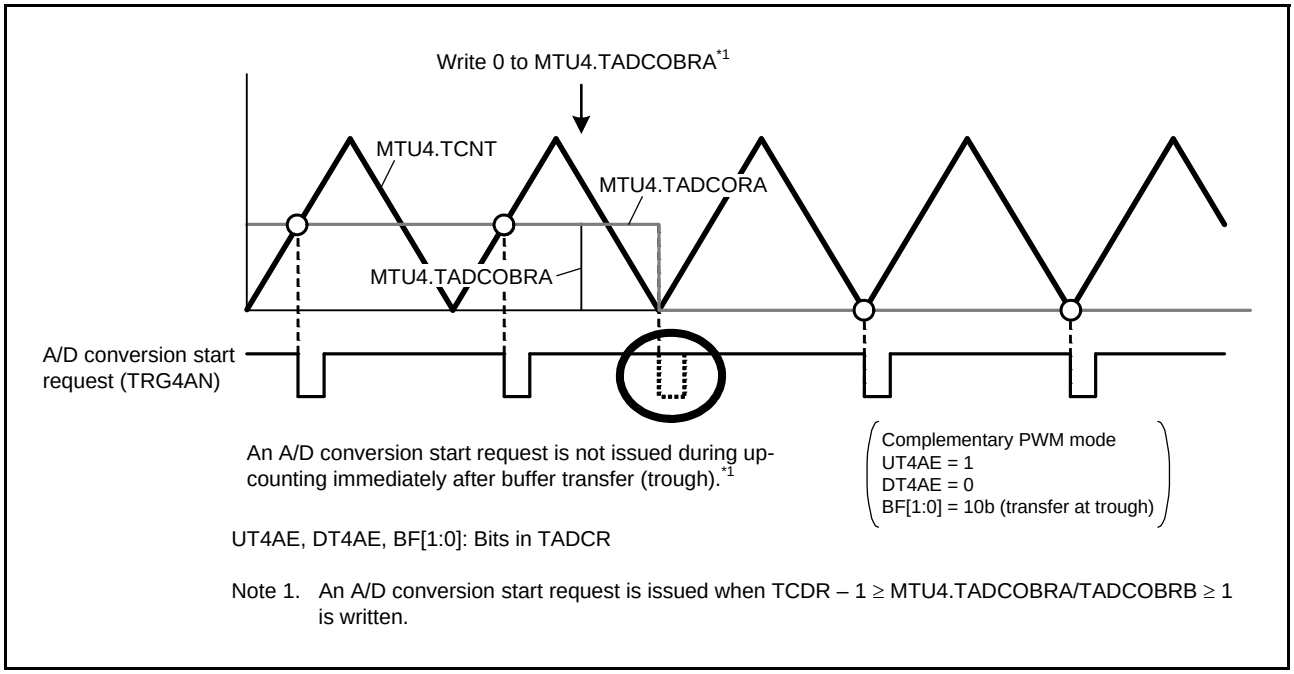


Figure 19.141 A/D Conversion Start Request When 0 is Written to MTU4.TADCOBRA

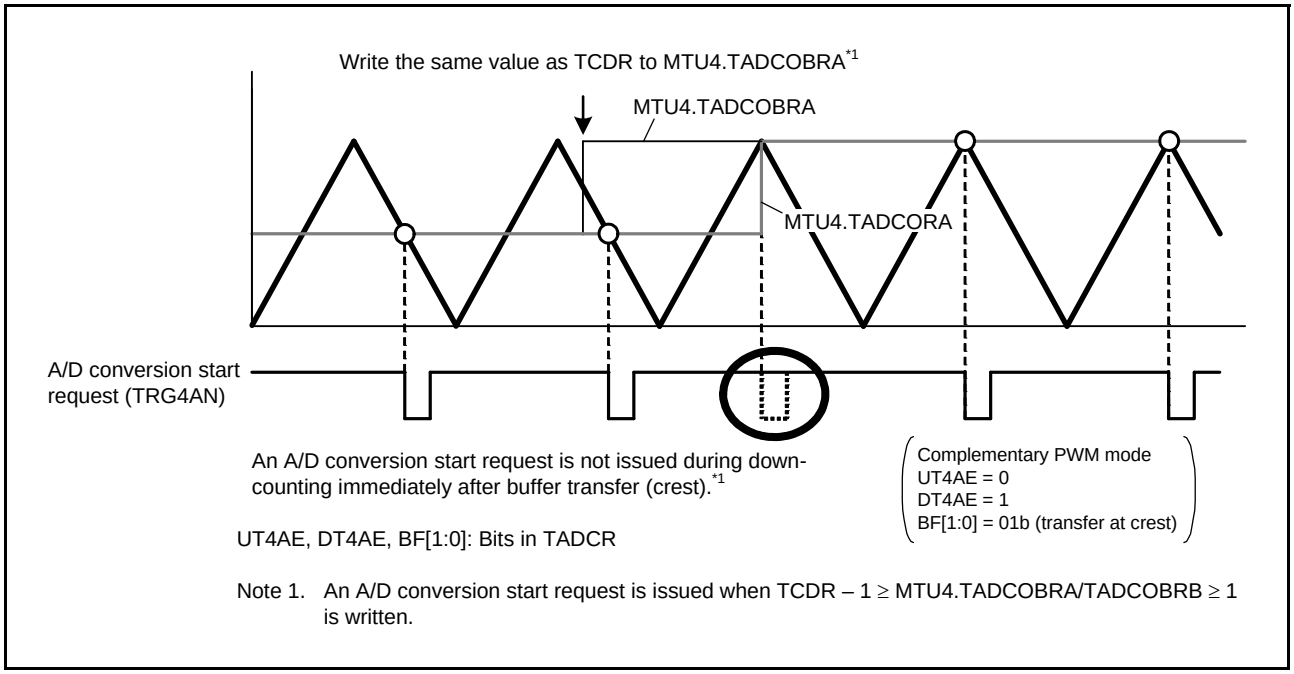


Figure 19.142 A/D Conversion Start Request When the Same Value as TCDR is Written to MTU4.TADCOBRA

19.7 MTU Output Pin Initialization

19.7.1 Operating Modes

The MTU has the following six operating modes. Waveforms can be output in any of these modes.

- Normal mode (MTU0 to MTU4)
- PWM mode 1 (MTU0 to MTU4)
- PWM mode 2 (MTU0 to MTU2)
- Phase counting modes 1 to 5 (MTU1 and MTU2)
- Complementary PWM mode (MTU3 and MTU4)
- Reset-synchronized PWM mode (MTU3 and MTU4)

This section describes how to initialize the MTU output pins in each of these modes.

19.7.2 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 direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports. MTU output can be disabled through TIOR settings. Complementary PWM output (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C and MTIOC4D) should be specified through TOERA setting. 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 19.64.

Table 19.64 Mode Transition Combinations

	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
CPWM	(21)	(22)	Not available	Not available	(23) (24)	(25)
RPWM	(26)	(27)	Not available	Not available	(28)	(29)

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

19.7.3 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) pins. When a pin is configured for MTIOCnB or MTIOCnD, it enters high-impedance state. To output a specified level, set the pin to general output port.
- In PWM mode 2, a waveform is not output to the pin that corresponds to the TGR register used as a period register. When a pin is configured for MTIOCnm (n = 0 to 2; m = A to D), it enters high-impedance state. To output a specified level, set the pin to general output port.
- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, waveforms are not output to the corresponding pins (MTIOCnC or MTIOCnD (n = 0, 3, 4)). When a pin is configured for MTIOCnC or MTIOCnD, it enters high-impedance state. To output a specified level, set the pin to general output port.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, waveforms are not output to the corresponding pins (MTIOCnC or MTIOCnD (n = 0, 3, 4)). When a pin is configured for MTIOCnC or MTIOCnD, it enters high-impedance state. To output a specified level, set the pin to general output port.
- 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 or TOCR2A) setting, temporarily disable output in MTU3 and MTU4 with the timer output master enable register (TOERA). At this time, when a pin is configured for MTIOCnm (n = 3, 4; m = A to D), it enters high-impedance state. To output a specified level, set the pin to general output port. 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).

Note: Channel number is substituted for “n” indicated in this section.

Pin initialization procedures are described below for the numbered combinations in Table 19.64. The active level is assumed to be low.

(1) Operation When Error Occurs in Normal Mode and Operation is Restarted in Normal Mode

Figure 19.143 shows a case in which an error occurs in normal mode and operation is restarted in normal mode after re-setting.

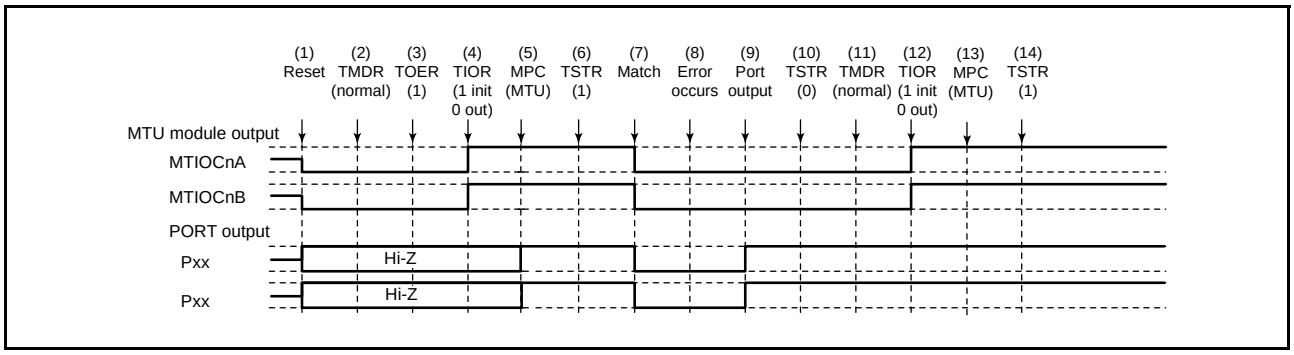


Figure 19.143 Error Occurrence in Normal Mode, Recovery in Normal Mode

(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, enable output with the TOERA 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 MPC and port mode registers (PMR) corresponding to the I/O ports.
(6) Start count operation by setting the TSTRA 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 direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(10) Stop count operation by setting the TSTRA 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 MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

(2) Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM Mode 1

Figure 19.144 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

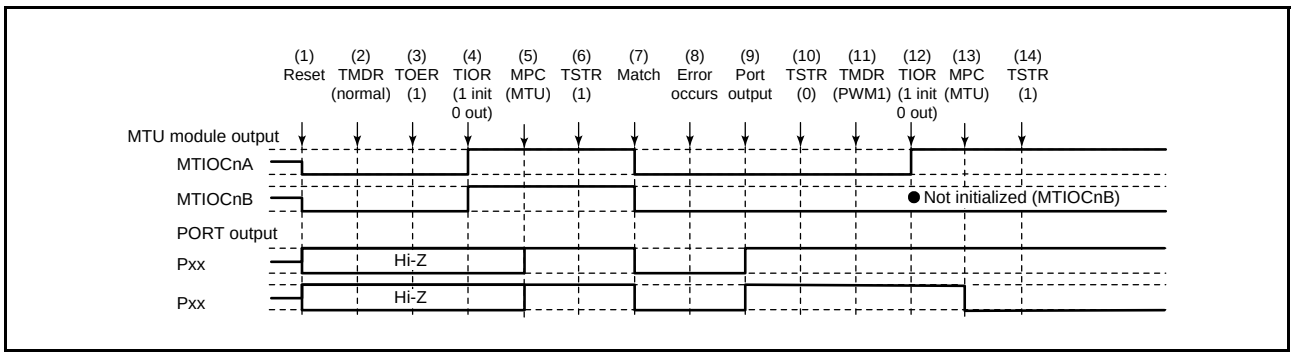


Figure 19.144 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 19.143.
(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 direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

(3) Operation When Error Occurs in Normal Mode and Operation is Restarted in PWM mode 2

Figure 19.145 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

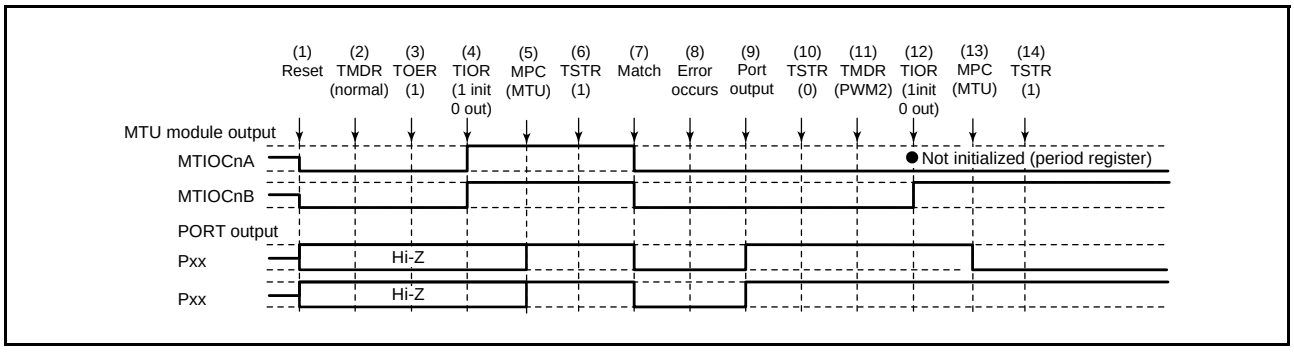


Figure 19.145 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

(1) to (10) are the same as in Figure 19.143.
(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 direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) 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.

(4) Operation When Error Occurs in Normal Mode and Operation is Restarted in Phase Counting Mode

Figure 19.146 shows a case in which an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

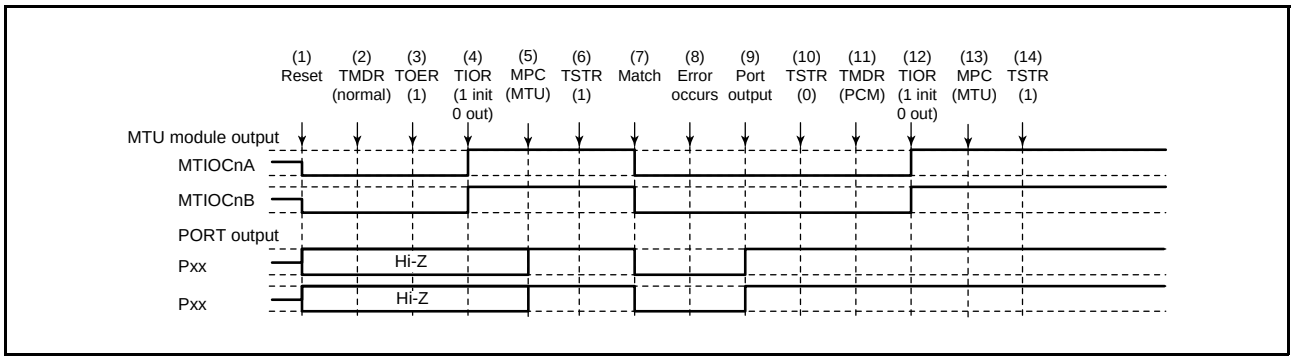


Figure 19.146 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

(1) to (10) are the same as in Figure 19.143.
(11) Set the phase counting mode.
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) 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.

(5) Operation When Error Occurs in Normal Mode and Operation is Restarted in Complementary PWM Mode

Figure 19.147 shows a case in which an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.

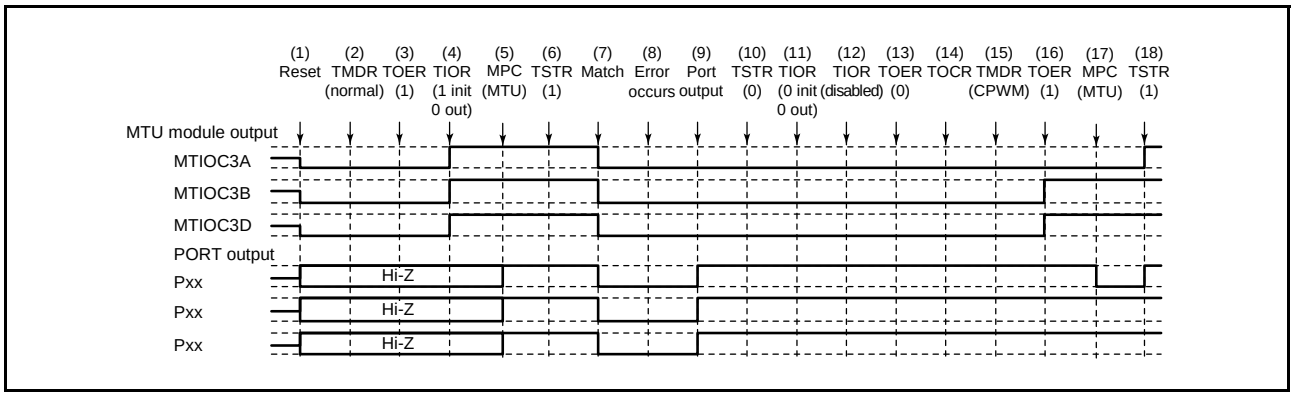


Figure 19.147 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 19.143.
(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 with the TOERA register.
(14) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A.
(15) Set complementary PWM mode.
(16) Enable output in MTU3 and MTU4 with the TOERA register.
(17) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(18) Restart operation by setting the TSTRA register.

(6) Operation When Error Occurs in Normal Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 19.148 shows a case in which an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.

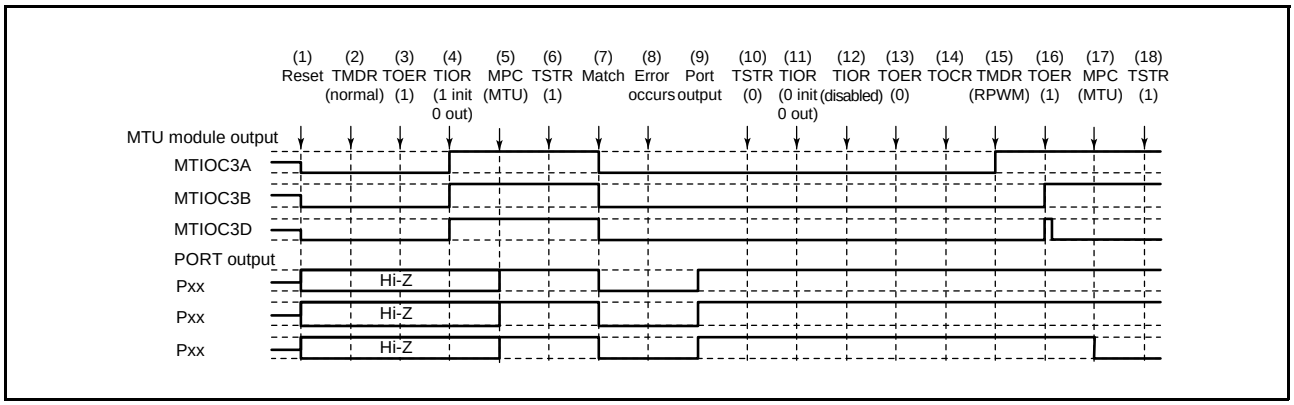


Figure 19.148 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (13) are the same as in Figure 19.147.

(14) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A.

(15) Set reset-synchronized PWM mode.

(16) Enable output in MTU3 and MTU4 with the TOERA register.

(17) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(18) Restart operation by setting the TSTRA register.

(7) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Normal Mode

Figure 19.149 shows a case in which an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.

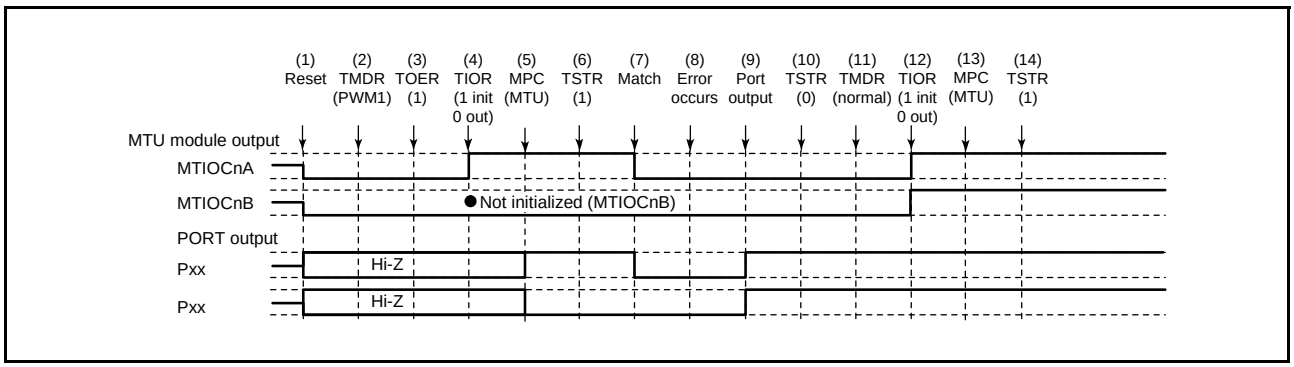


Figure 19.149 Error Occurrence in PWM Mode 1, Recovery in Normal Mode

(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, enable output with the TOERA 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 MPC and port mode registers (PMR) corresponding to the I/O ports.
(6) Start count operation by setting the TSTRA 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 direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(10) Stop count operation by setting the TSTRA register.
(11) Set normal mode.
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

(8) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 1

Figure 19.150 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

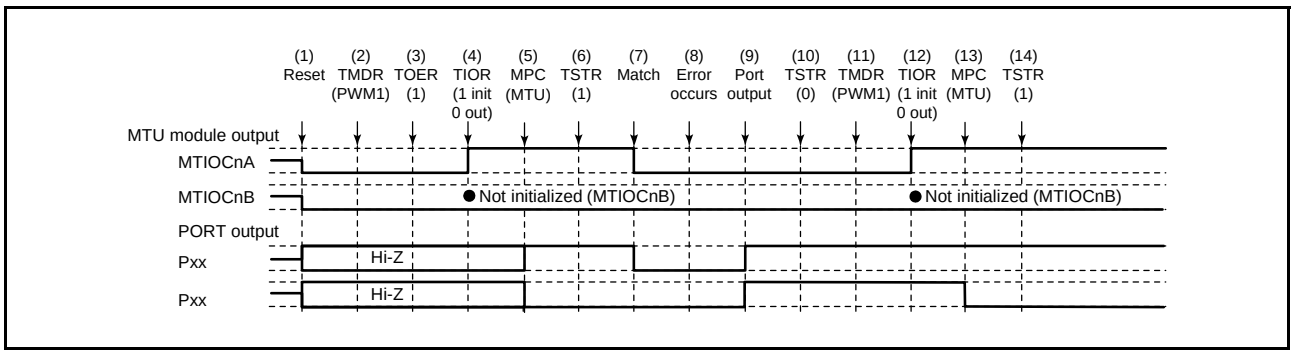


Figure 19.150 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 19.149.
(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 direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

(9) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in PWM mode 2

Figure 19.151 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

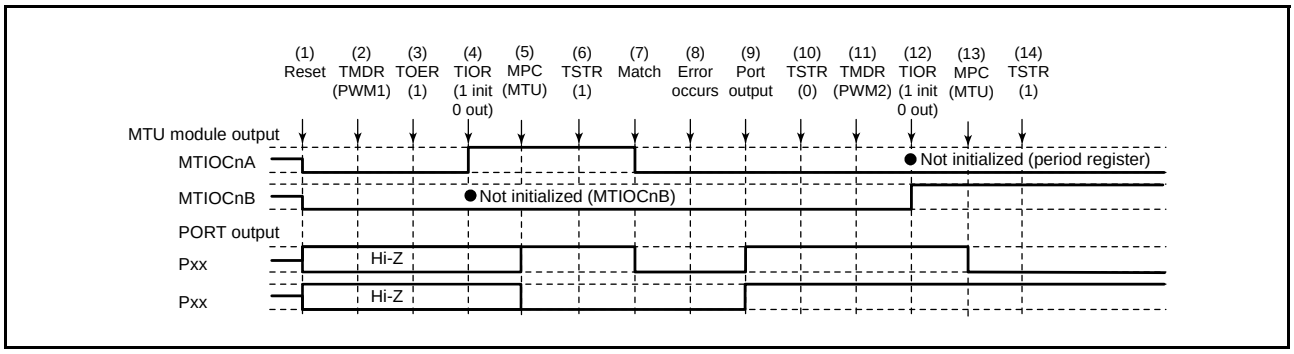


Figure 19.151 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

(1) to (10) are the same as in Figure 19.149.
(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 direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) 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.

(10) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Phase Counting Mode

Figure 19.152 shows a case in which an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

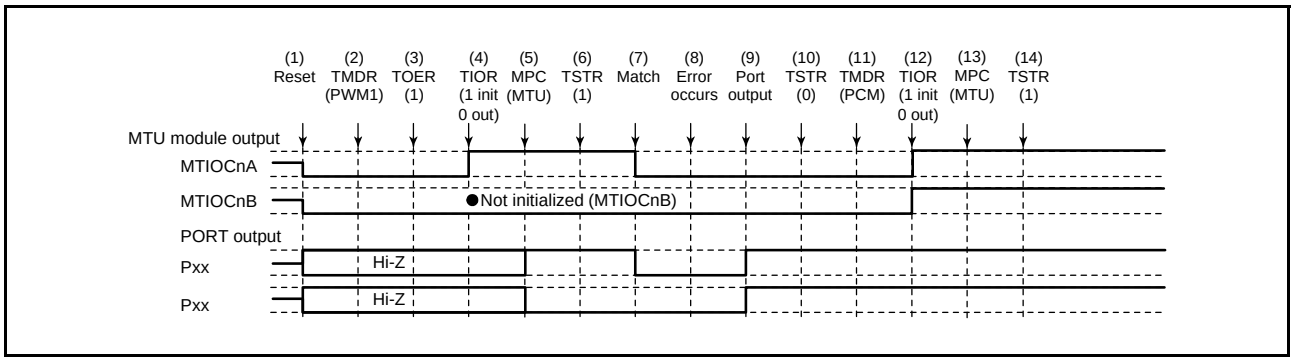


Figure 19.152 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

(1) to (10) are the same as in Figure 19.149.
(11) Set the phase counting mode.
(12) Initialize the pins with the TIOR register.
(13) Set MTU output using the MPC and port mode registers (PMR) 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.

(11) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Complementary PWM Mode

Figure 19.153 shows a case in which an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

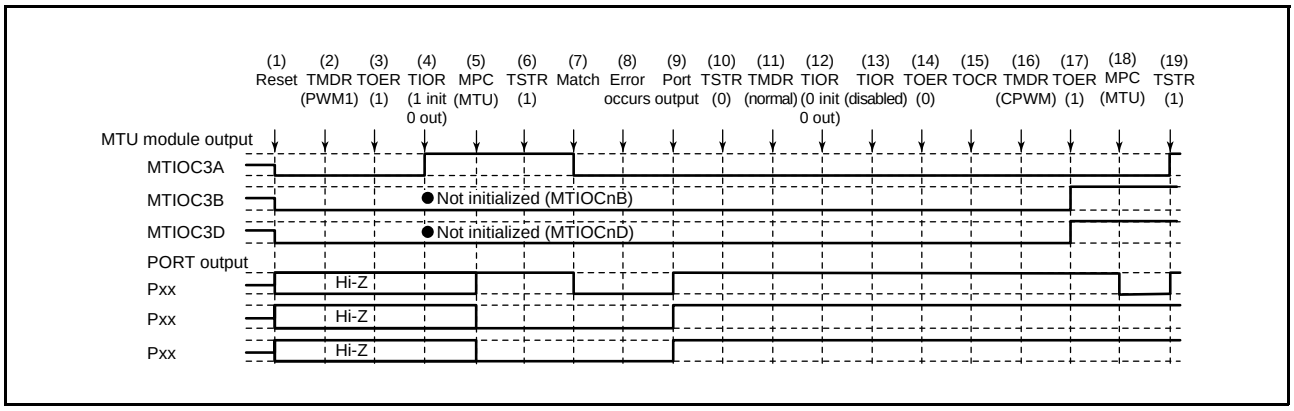


Figure 19.153 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 19.149.
(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 with the TOERA register.
(15) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A.
(16) Set complementary PWM mode.
(17) Enable output in MTU3 and MTU4 with the TOERA register.
(18) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(19) Restart operation by setting the TSTRA register.

(12) Operation When Error Occurs in PWM Mode 1 and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 19.154 shows a case in which an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

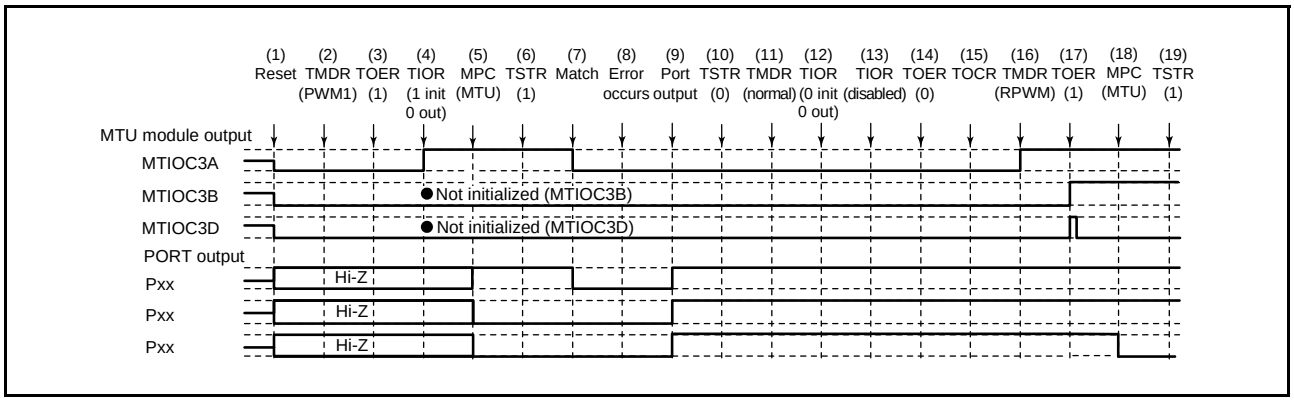


Figure 19.154 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

(1) to (14) are the same as in Figure 19.153.

(15) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A.

(16) Set reset-synchronized PWM mode.

(17) Enable output in MTU3 and MTU4 with the TOERA register.

(18) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(19) Restart operation by setting the TSTRA register.

(13) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Normal Mode

Figure 19.155 shows a case in which an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.

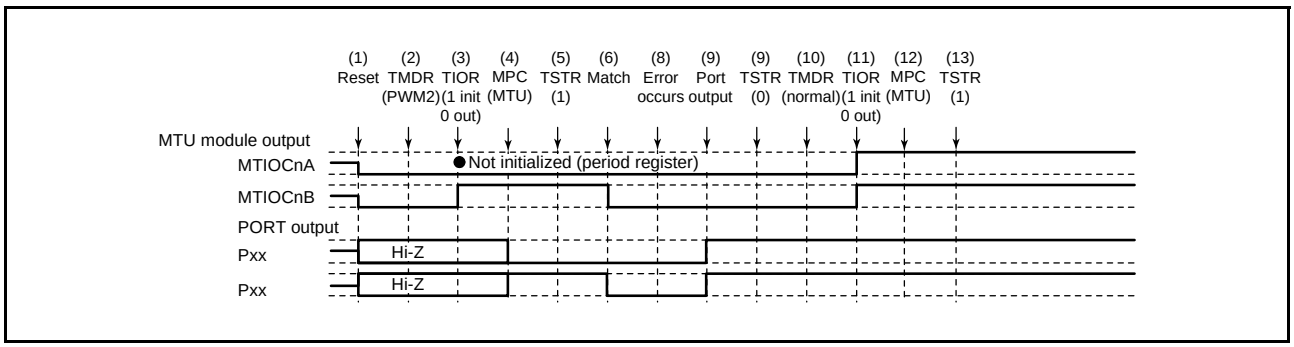


Figure 19.155 Error Occurrence in PWM Mode 2, Recovery in Normal Mode

(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 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 2, the pin that corresponds to the TGR register used as a period register is not initialized. In the example, the MTUn.TGRA register is used as a period register.)
(4) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(5) Start count operation by setting the TSTRA 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 direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(9) Stop count operation by setting the TSTRA register.
(10) Set normal mode.
(11) Initialize the pins with the TIOR register.
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(14) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 1

Figure 19.156 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

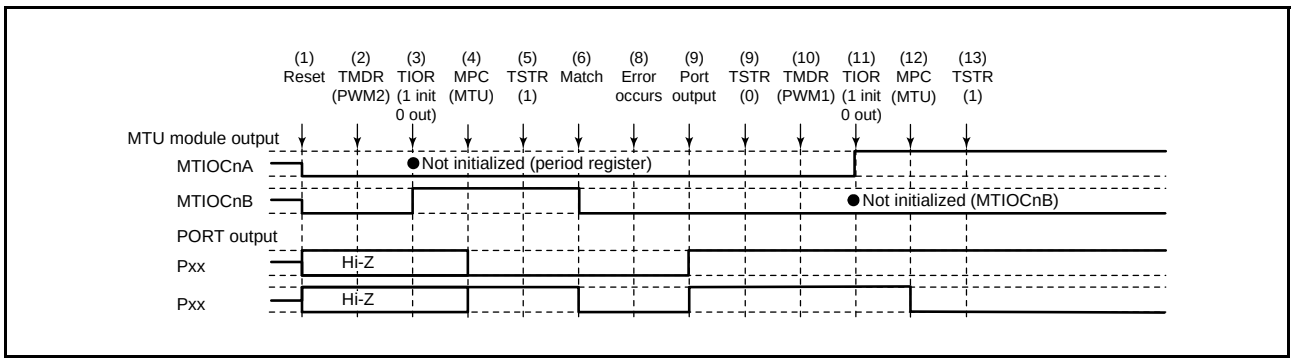


Figure 19.156 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 19.155.
(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 direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(15) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 2

Figure 19.157 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

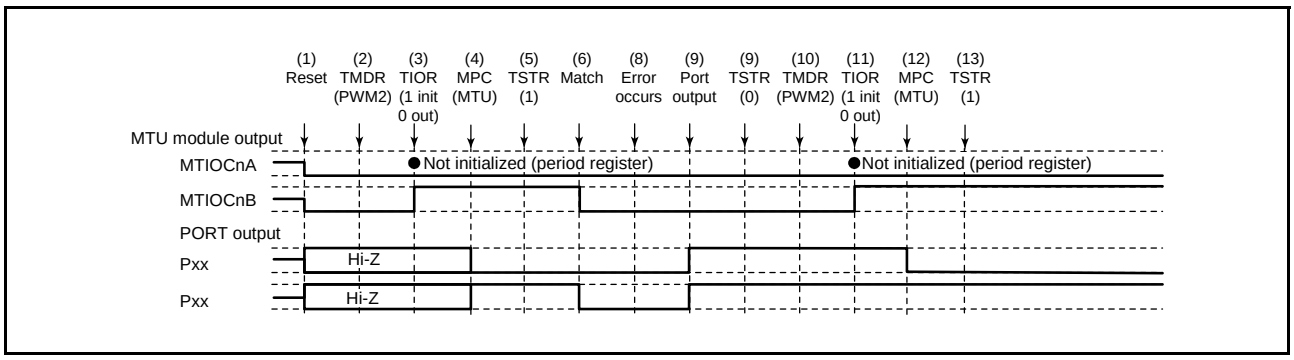


Figure 19.157 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

(1) to (9) are the same as in Figure 19.155.
(10) This step is not necessary when restarting in PWM mode 2.
(11) 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 direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(16) Operation When Error Occurs in PWM Mode 2 and Operation is Restarted in Phase Counting Mode

Figure 19.158 shows a case in which an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

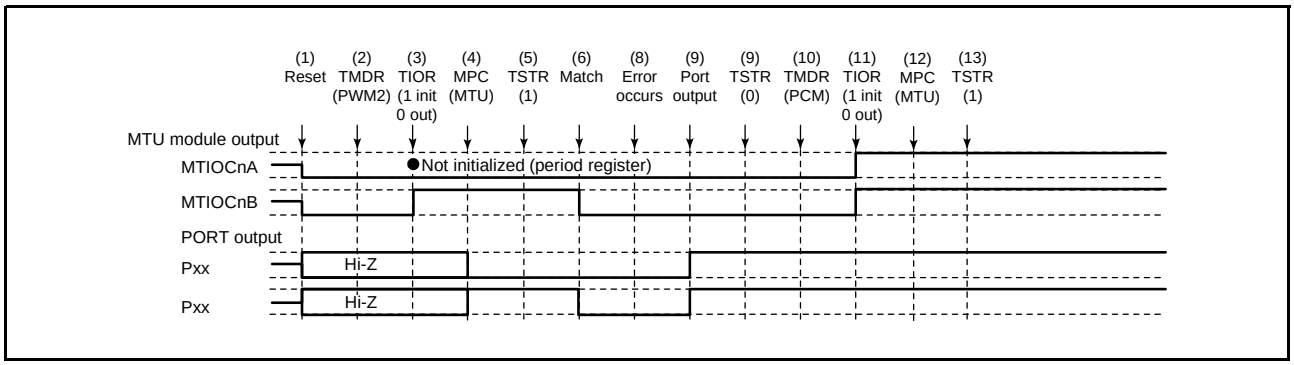


Figure 19.158 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

(1) to (9) are the same as in Figure 19.155.
(10) Set the phase counting mode.
(11) Initialize the pins with the TIOR register.
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(17) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Normal Mode

Figure 19.159 shows a case in which an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.

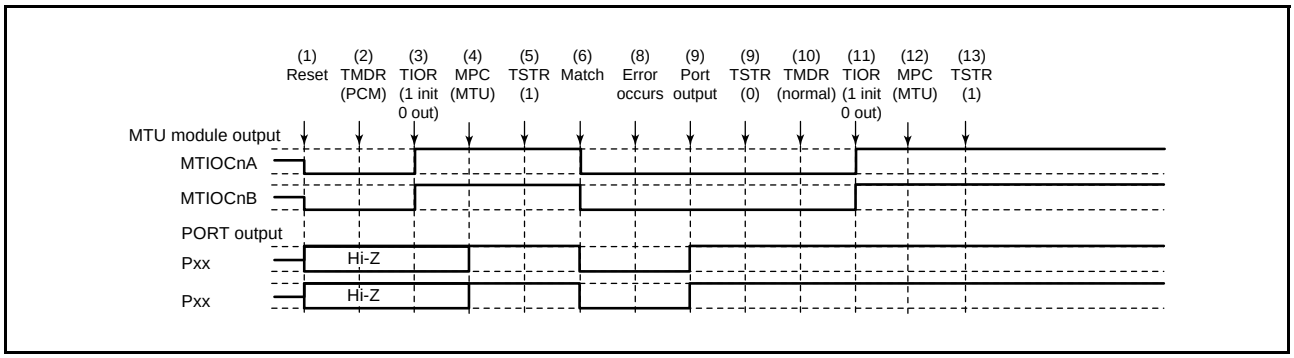


Figure 19.159 Error Occurrence in Phase Counting Mode, Recovery in Normal Mode

(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 MPC and port mode registers (PMR) corresponding to the I/O ports.
(5) Start count operation by setting the TSTRA 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 direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(9) Stop count operation by setting the TSTRA register.
(10) Set normal mode.
(11) Initialize the pins with the TIOR register.
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(18) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 1

Figure 19.160 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

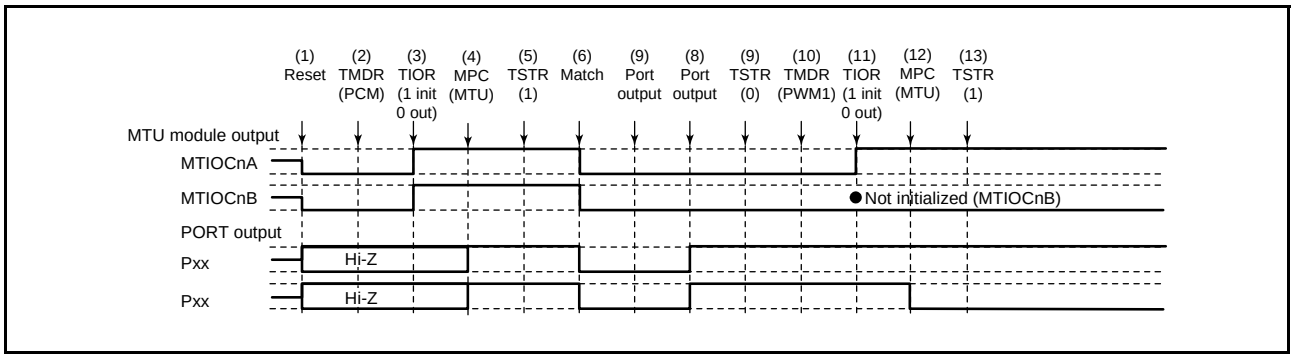


Figure 19.160 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

(1) to (9) are the same as in Figure 19.159.
(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 direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(19) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 2

Figure 19.161 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

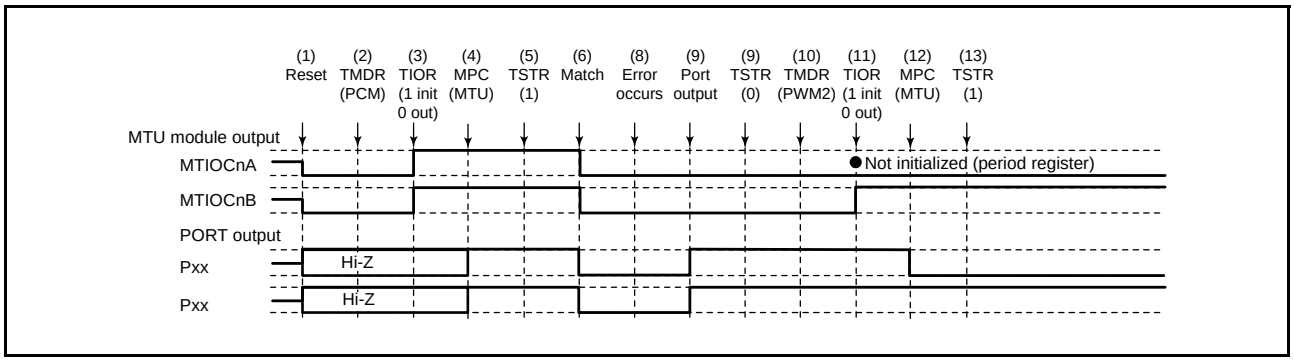


Figure 19.161 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

(1) to (9) are the same as in Figure 19.159.
(10) Set PWM mode 2.
(11) 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 direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(12) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(20) Operation When Error Occurs in Phase Counting Mode and Operation is Restarted in Phase Counting Mode

Figure 19.162 shows a case in which an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

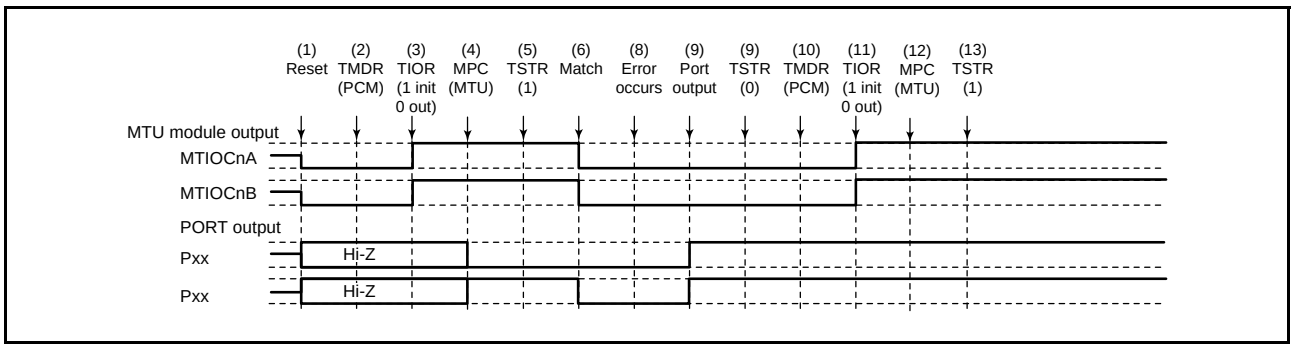


Figure 19.162 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

(1) to (9) are the same as in Figure 19.159.
(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 MPC and port mode registers (PMR) corresponding to the I/O ports.
(13) Restart operation by setting the TSTRA register.

(21) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Normal Mode

Figure 19.163 shows a case in which an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.

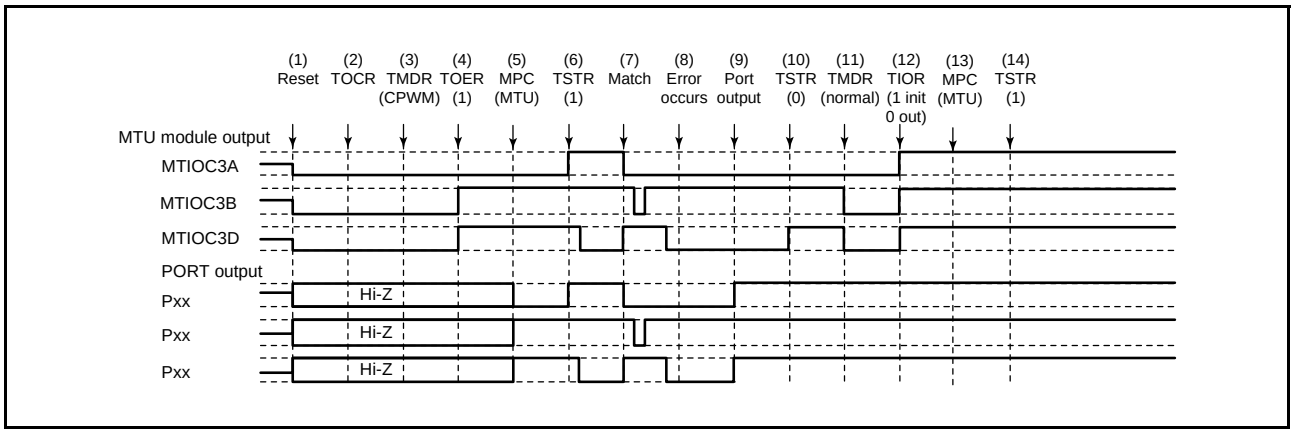


Figure 19.163 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode

(1) After a reset, the MTU3 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.
(3) Set complementary PWM mode.
(4) Enable output in MTU3 and MTU4 with the TOERA register.
(5) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(6) Start count operation by setting the TSTRA 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 direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(10) Stop count operation by setting the TSTRA 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 MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

(22) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in PWM Mode 1

Figure 19.164 shows a case in which an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

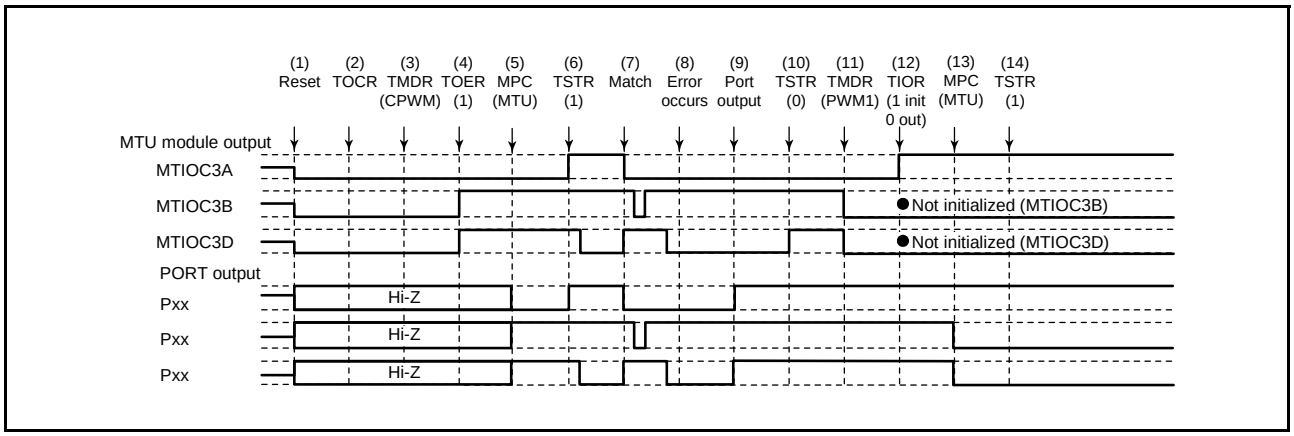


Figure 19.164 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 19.163.
(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 direction registers (PDR) and port output data registers (PODR) of the I/O ports.)
(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

(23) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 19.165 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).

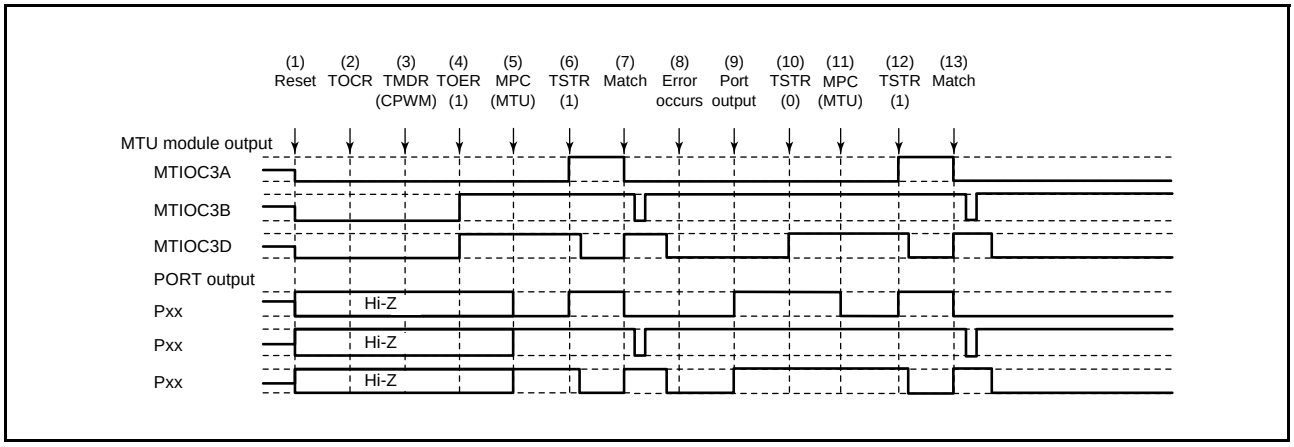


Figure 19.165 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode (1)

(1) to (10) are the same as in Figure 19.163.
(11) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(12) Restart operation by setting the TSTRA register.
(13) The complementary PWM waveform is output on compare match occurrence.

(24) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode with New Settings

Figure 19.166 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).

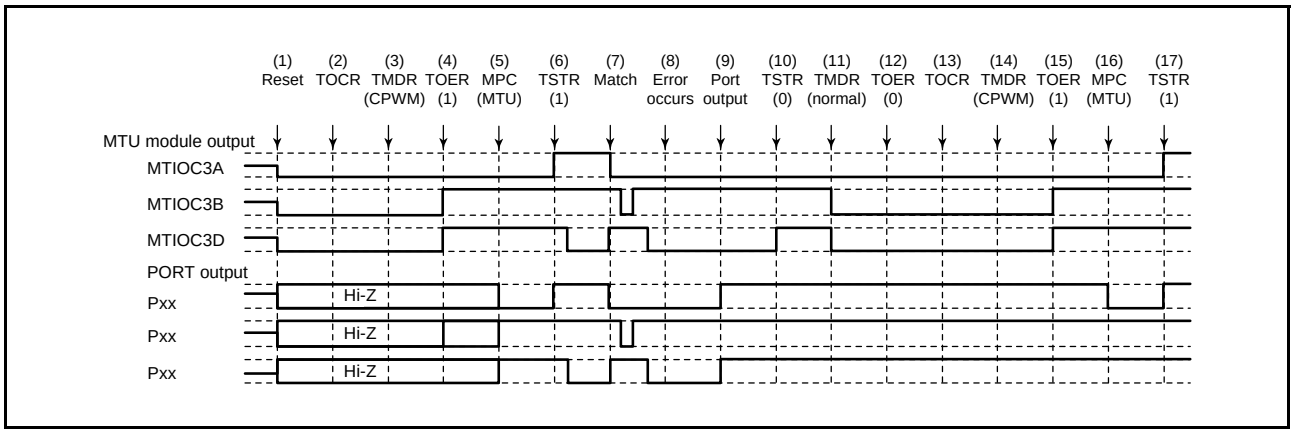


Figure 19.166 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode (2)

(1) to (10) are the same as in Figure 19.163.
(11) Set normal mode and make new settings (MTU output goes low).
(12) Disable output in MTU3 and MTU4 with the TOERA register.
(13) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A.
(14) Set complementary PWM mode.
(15) Enable output in MTU3 and MTU4 with the TOERA register.
(16) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(17) Restart operation by setting the TSTRA register.

(25) Operation When Error Occurs in Complementary PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 19.167 shows a case in which an error occurs in complementary PWM mode and operation is restarted in reset-synchronized PWM mode after re-setting.

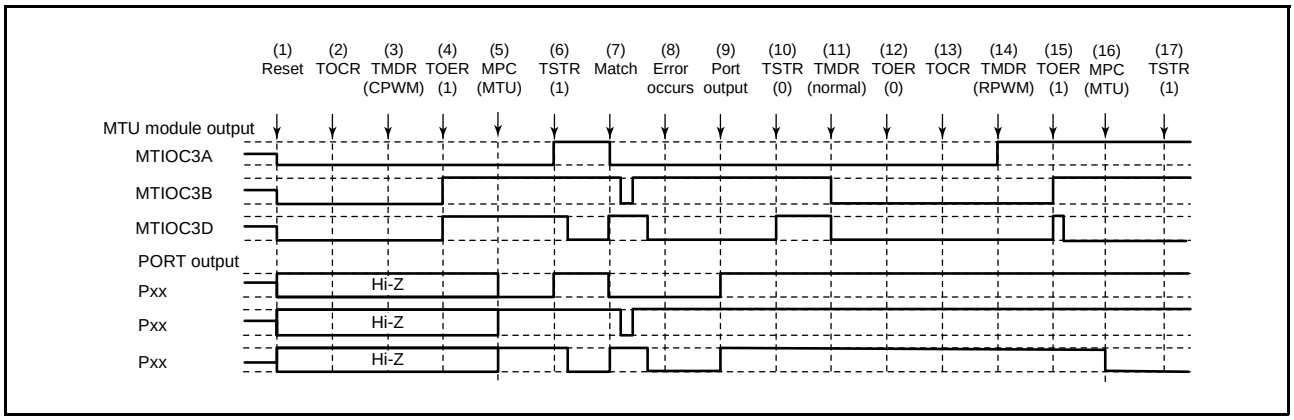


Figure 19.167 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (10) are the same as in Figure 19.163.
(11) Set normal mode (MTU output goes low).
(12) Disable output in MTU3 and MTU4 with the TOERA register.
(13) Select the reset-synchronized PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A.
(14) Set reset-synchronized PWM mode.
(15) Enable output in MTU3 and MTU4 with the TOERA register.
(16) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(17) Restart operation by setting the TSTRA register.

(26) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Normal Mode

Figure 19.168 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.

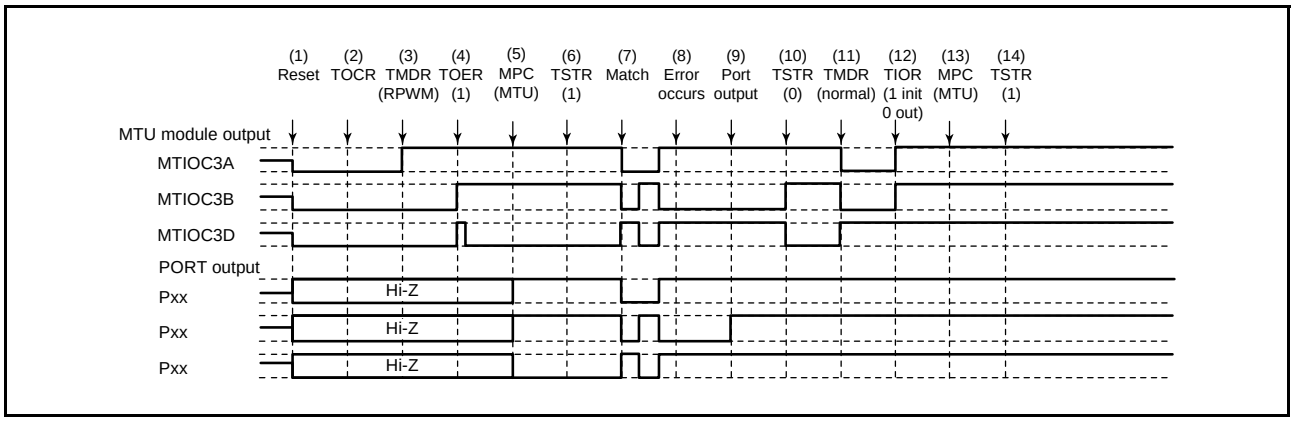


Figure 19.168 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode

(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.
(3) Set reset-synchronized PWM mode.
(4) Enable output in MTU3 and MTU4 with the TOERA register.
(5) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(6) Start count operation by setting the TSTRA 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 direction registers (PDR), port output data registers (PODR), and port mode registers (PMR) of the I/O ports.
(10) Stop count operation by setting the TSTRA 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 MPC and port mode registers (PMR) corresponding to the I/O ports.
(14) Restart operation by setting the TSTRA register.

(27) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in PWM Mode 1

Figure 19.169 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in PWM mode 1 after re-setting.

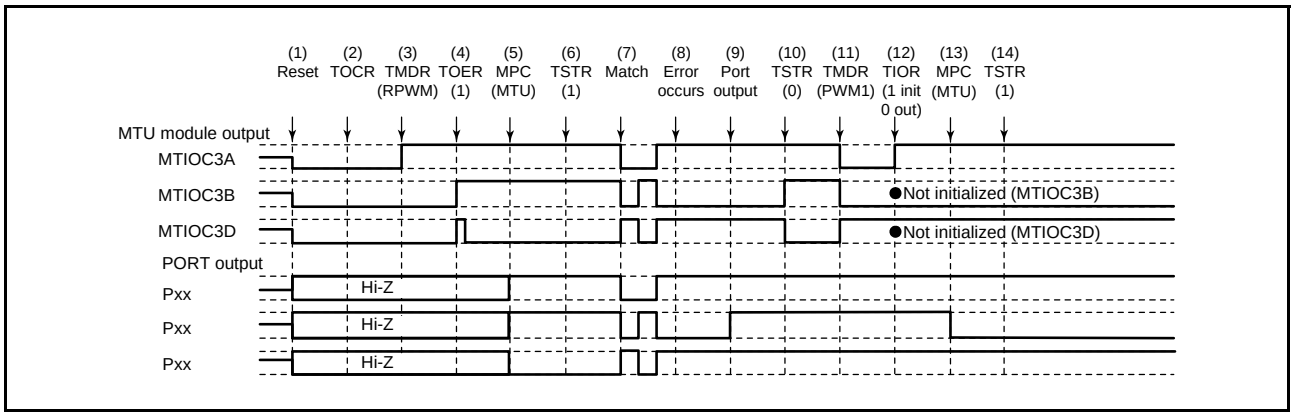


Figure 19.169 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1

(1) to (10) are the same as in Figure 19.168.

(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 direction registers (PDR) and port output data registers (PODR) of the I/O ports.)

(13) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(14) Restart operation by setting the TSTRA register.

(28) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 19.170 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in complementary PWM mode after re-setting.

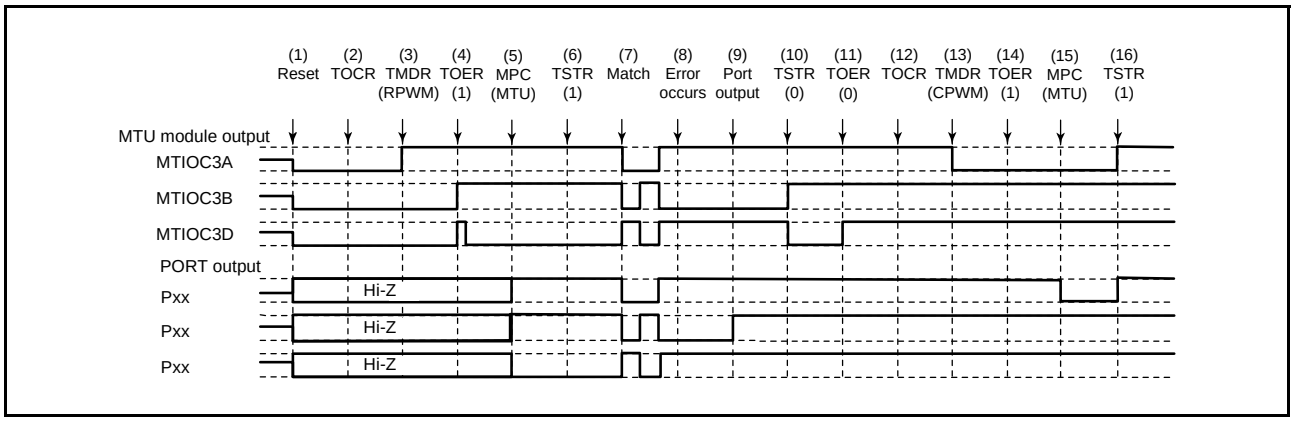


Figure 19.170 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode

(1) to (10) are the same as in Figure 19.168.
(11) Disable output in MTU3 and MTU4 with the TOERA register.
(12) Select the complementary PWM output level and enable or disable cyclic output with registers TOCR1A and TOCR2A.
(13) Set complementary PWM mode (MTU cyclic output pin goes low).
(14) Enable output in MTU3 and MTU4 with the TOERA register.
(15) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.
(16) Restart operation by setting the TSTRA register.

(29) Operation When Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 19.171 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.

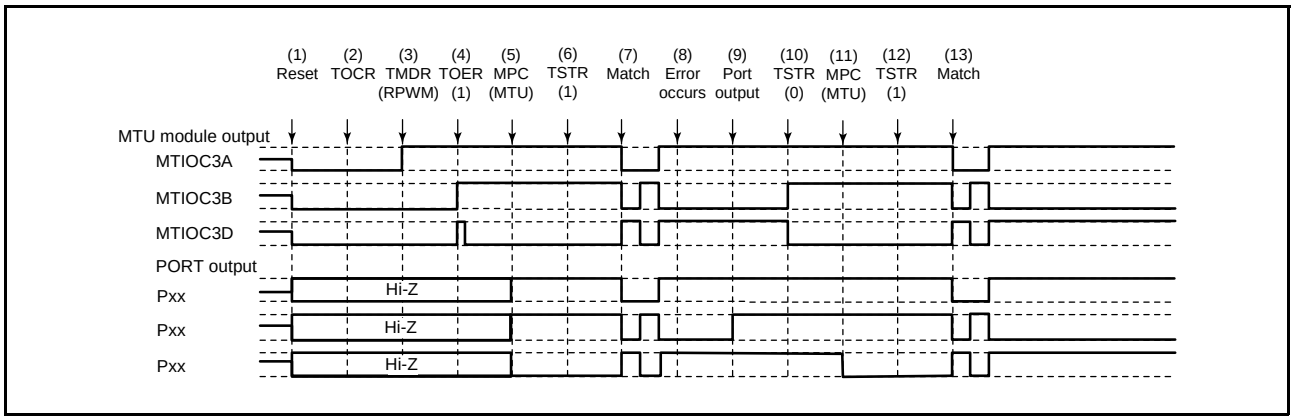


Figure 19.171 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode

(1) to (10) are the same as in Figure 19.168.

(11) Set MTU output using the MPC and port mode registers (PMR) corresponding to the I/O ports.

(12) Restart operation by setting the TSTRA register.

(13) The reset-synchronized PWM waveform is output on compare match occurrence.

20. Port Output Enable 3 (POE3C)

This MCU incorporates a port output enable 3 (POE3C) which can be used to, under various conditions, disable output signals for the MTU. Every output signal is put in the high-impedance state when the output is disabled.
In this section, “PCLK” is used to refer to PCLKB.

20.1 Overview

Table 20.1 lists the specifications of the POE, and Figure 20.1 shows a block diagram of the POE.

Table 20.1 POE Specifications

Item	Description
Pin status while output is disabled	• High-impedance
Target pins for switching to high-impedance state	• MTU output pins MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D) MTU3 pins (MTIOC3B, MTIOC3D) MTU4 pins (MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D)
Generating conditions of request for switching to high-impedance state	• Input signal detection: Detection of the POE0#, POE8#, and POE10# signal level. • Simultaneous conduction between output pins: A match (simultaneous conduction) between the output signal levels at the active level over one or more cycles on the following combination of pins • The SPOER register setting • Detection that the main clock oscillator had stopped oscillating • Detection of the comparator C (CMPC) outputs
Function	• Each of the POE0#, POE8#, and POE10# input pins can be set for falling edge, PCLK/8 × 16, PCLK/16 × 16, or PCLK/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#, POE8#, or POE10# 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 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 in response to comparator C (CMPC) output detection. • The outputs of the target pins can be in the high-impedance state by modifying the settings of the POE registers. • Interrupts can be generated by input-level sampling or output-level comparison results.

	MTU Complementary PWM Output Pins
1	MTIOC3B and MTIOC3D
2	MTIOC4A and MTIOC4C
3	MTIOC4B and MTIOC4D

The POE has input-level detection circuits, output-level comparison circuits, and a high-impedance request/interrupt request generating circuit as shown in Figure 20.1.

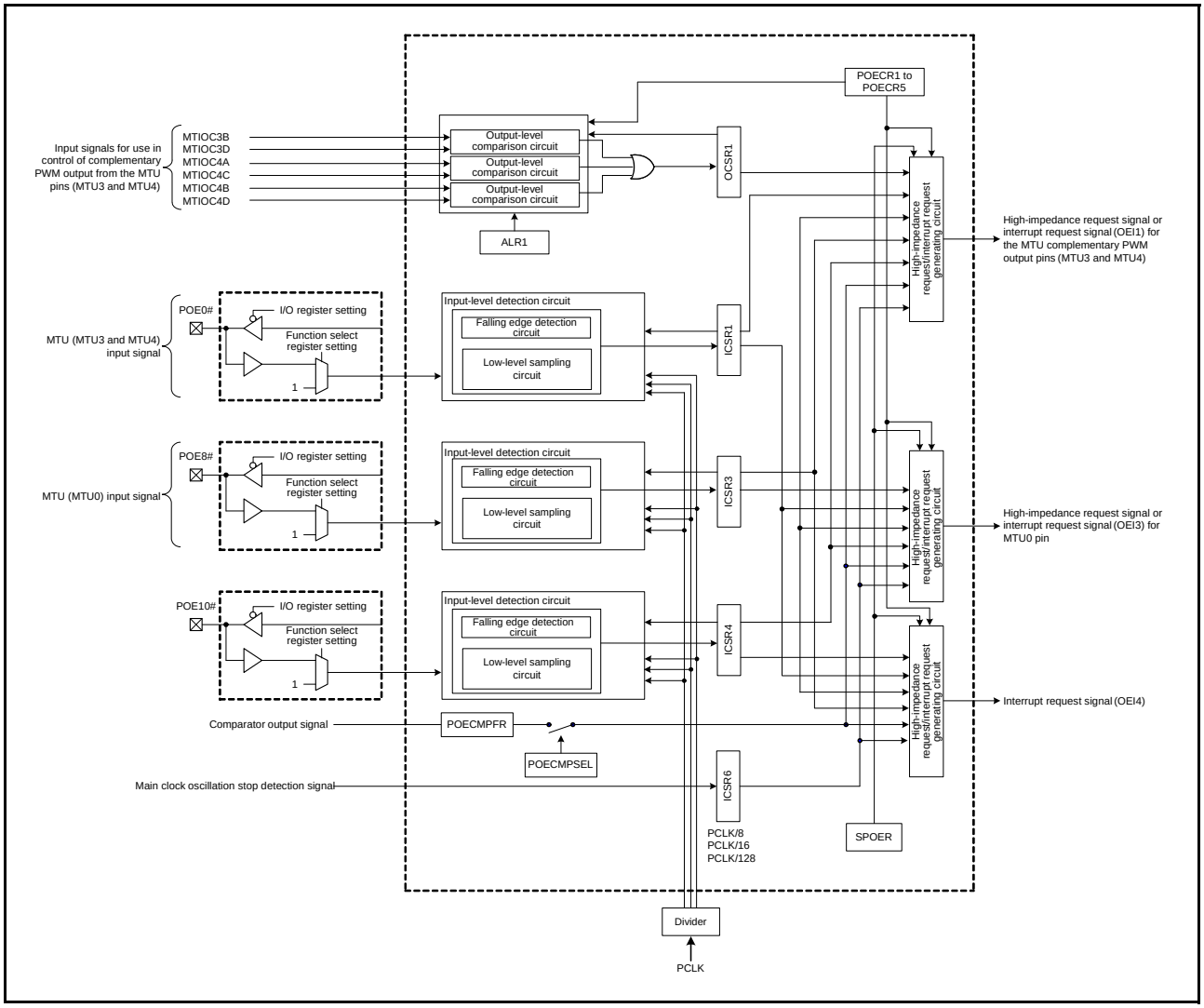


Figure 20.1 POE Block Diagram

Table 20.2 shows I/O pins to be used by the POE.

Table 20.2 POE I/O Pins

Pin Name	I/O	Description
POE0#	Input	Request signal to put the outputs of the MTU complementary PWM output pins (MTU3, MTU4 pins) in the high-impedance state, and is also capable of controlling the other target pins by register settings.
POE8#	Input	Request signal to put the output of the MTU0 pins in the high-impedance state, and is also capable of controlling the other target pins by register settings.
POE10#	Input	Is capable of controlling every target pins by register settings.

Table 20.3 shows output-level comparisons with pin combinations.

Table 20.3 Pin Combinations

Pin Combination	I/O	Description
MTIOC3B and MTIOC3D	Output	The MTU complementary PWM output pins (MTU3 and MTU4 pins) are in the high-impedance state when two pins of the set simultaneously output the active level (low level when the MTU.TOCR1A.OLSP bit is 0 or high level when the OLSP bit is 1 while the OLSEN bit in the ALR1 register is 0 and the MTU.TOCR1A.TOCS bit is 0, low level when the OLS1P, OLS1N, OLS2P, OLS2N, OLS3P, and OLS3N bits in the MTU.TOCR2A register are 0 or high level when these bits are 1 while the OLSEN bit in the ALR1 register is 0 and the MTU.TOCR1A.TOCS bit is 1, or low level when the OLSG0A, OLSG0B, OLSG1A, OLSG1B, OLSG2A, and OLSG2B bits in the ALR1 register are 0 and high level when these bits are 1 while the OLSEN bit in the ALR1 register is 1) for at least one cycle of the peripheral module clock (PCLK). Pin combinations for output comparison can be selected by registers of POE.
MTIOC4A and MTIOC4C	Output	
MTIOC4B and MTIOC4D	Output	

20.2 Register Descriptions

The POE registers are initialized by a reset.

20.2.1 Input Level Control/Status Register 1 (ICSR1)

Address(es): POE.ICSR1 0008 C4C0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	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	Bit Name	Description	R/W
b1, b0	POE0M[1:0]	POE0 Mode Select	b1 b0 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 PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE0# pin input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE0# pin input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PIE1	Port Interrupt Enable 1	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	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
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR1 register selects the POE0# pin input modes, 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 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
  When low-level sampling is set by the POE0M[1:0] bits, the high level needs to be input to the POE0# pin to write 0 to this flag.
  For details, refer to section 20.3.7, Recover from High-Impedance State.

20.2.2 Input Level Control/Status Register 3 (ICSR3)

Address(es): POE.ICSR3 0008 C4C8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	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	Bit Name	Description	R/W
b1, b0	POE8M[1:0]	POE8 Mode Select	b1 b0 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 PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE8# pin input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE8# pin input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PIE3	Port Interrupt Enable 3	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	POE8E	POE8 High-Impedance Enable	0: Does not put the output in the high-impedance state by POE8# signal. 1: Put the output in the high-impedance state by POE8# signal.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	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
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR3 register selects the POE8# pin input mode, 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 is set to 1.

POE8E Bit (POE8 High-Impedance Enable)

This bit specifies whether to put the output of 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
  When low-level sampling is set by the POE8M[1:0] bits, the high level needs to be input to the POE8# pin to write 0 to this flag.

For details, refer to section 20.3.7, Recover from High-Impedance State.

20.2.3 Input Level Control/Status Register 4 (ICSR4)

Address(es): POE.ICSR4 0008 C4D6h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	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	Bit Name	Description	R/W
b1, b0	POE10M[1:0]	POE10 Mode Select	b1 b0 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 PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE10# pin input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE10# pin input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PIE4	Port Interrupt Enable 4	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	POE10E	POE10 High-Impedance Enable	0: Does not put the output in the high-impedance state by POE10# signal. 1: Put the output in the high-impedance state by POE10# signal.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	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
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR4 register selects the POE10# pin input mode, 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 is set to 1.

POE10E Bit (POE10 High-Impedance Enable)

This bit specifies whether to put the output of 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
  When low-level sampling is set by the POE10M[1:0] bits, the high level needs to be input to the POE10# pin to write 0 to this flag.
  For details, refer to section 20.3.7, Recover from High-Impedance State.

20.2.4 Input Level Control/Status Register 6 (ICSR6)

Address(es): POE.ICSR6 0008 C4DCh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	OSTSTF	—	—	OSTSTE	—	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	OSTSTE	Oscillation Stop High-Impedance Enable	0: Does not put the output in the high-impedance state when the oscillation stop is detected. 1: Put the output in the high-impedance state when the oscillation stop is detected.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	OSTSTF	Oscillation Stop Detection Flag	0: Indicates that a high-impedance request by oscillation stop has not been generated. 1: Indicates that a high-impedance request by oscillation stop has been generated.	R/W*2
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The ICSR6 register controls the oscillation stop high-impedance and indicates status.

OSTSTE Bit (Oscillation Stop High-Impedance Enable)

This bit specifies whether to put the output of the target pin in the high-impedance state when oscillation stop is detected.

OSTSTF Flag (Oscillation Stop Detection Flag)

This flag indicates that a high-impedance request by the oscillation stop has been generated.
When the main clock oscillation stops, this flag is set to 1. To clear this flag, wait for at least 10 cycles of PCLK after this flag becomes 1 and write 0 to this flag while the OSTDSR.OSTDF flag is 0. Writing 0 to this flag while the OSTDSR.OSTDF flag is 1 cannot clear this flag. After clearing this flag, confirm that the flag has actually been modified to 0.

[Setting condition]

- When oscillation stop is detected

[Clearing condition]

- By writing 0 to the OSTSTF flag after reading OSTSTF = 1

20.2.5 Output Level Control/Status Register 1 (OCSR1)

Address(es): POE.OCSR1 0008 C4C2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	OSF1	—	—	—	—	—	OCE1	OIE1	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	OIE1	Simultaneous Conduction Interrupt Enable 1	0: Interrupt requests disabled 1: Interrupt requests enabled	R/W
b9	OCE1	Simultaneous Conduction High-Impedance Enable 1	0: Does not put the outputs in the high-impedance state when they simultaneously go to an active level. 1: Put the outputs in the high-impedance state when they simultaneously go to an active level.	R/W*1
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	OSF1	Simultaneous Conduction 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.
Note 2. Writing 0 to this bit after reading it as 1 clears the flag and is the only allowed way.

The OCSR1 register controls the enable/disable of output-level comparison and interrupts, and indicates status.

OIE1 Bit (Simultaneous Conduction Interrupt Enable 1)

This bit enables or disables interrupt requests when the OSF1 flag is set to 1.

OCE1 Bit (Simultaneous Conduction High-Impedance Enable 1)

This bit specifies whether to put the output of the target pin in the high-impedance state when the OSF1 flag is set to 1.

OSF1 Flag (Simultaneous Conduction Flag 1)

This flag indicates that at least one of the three pairs of two-phase output pins for MTU complementary PWM output (MTU3 and MTU4) has simultaneously become at the active level. If the high-impedance control for the corresponding pins is not enabled, this flag does not become 1.
For setting the active level, refer to section 20.2.6, Active Level Setting Register 1 (ALR1).
[Setting condition]

- When the MTIOC3B and MTIOC3D pins simultaneously go to the active level*1 for at least one cycle of PCLK while the POECR2.MTU3BDZE bit is 1.
- When the MTIOC4A and MTIOC4C pins simultaneously go to the active level*1 for at least one cycle of PCLK while the POECR2.MTU4ACZE bit is 1.
- When the MTIOC4B and MTIOC4D pins simultaneously go to the active level*1 for at least one cycle of PCLK while the POECR2.MTU4BDZE bit is 1.

Note 1. The setting condition is judged only by the level of the pin regardless the setting of the MPC.PmnPFS register.

[Clearing condition]

- By writing 0 to the OSF1 flag after reading OSF1 = 1

To write 0 to this flag, the inactive level needs to be output from MTU complementary PWM output pins. For details,

refer to section 20.3.7, Recover from High-Impedance State.

20.2.6 Active Level Setting Register 1 (ALR1)

Address(es): POE.ALR1 0008 C4DAh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	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	Bit Name	Description	R/W
b0	OLSG0A	MTIOC3B Pin Active Level Setting	0: Active low 1: Active high	R/W*1
b1	OLSG0B	MTIOC3D Pin Active Level Setting	0: Active low 1: Active high	R/W*1
b2	OLSG1A	MTIOC4A Pin Active Level Setting	0: Active low 1: Active high	R/W*1
b3	OLSG1B	MTIOC4C Pin Active Level Setting	0: Active low 1: Active high	R/W*1
b4	OLSG2A	MTIOC4B Pin Active Level Setting	0: Active low 1: Active high	R/W*1
b5	OLSG2B	MTIOC4D Pin Active Level Setting	0: Active low 1: Active high	R/W*1
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	OLSEN	Active Level Setting Enable	0: Disabled 1: Enabled	R/W*1
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The ALR1 register specifies the active levels of the MTU outputs for detection of simultaneous conduction of those outputs as reflected in the OCSR1 register.

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 simultaneous conduction.

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 simultaneous conduction.

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 simultaneous conduction.

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 simultaneous conduction.

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 simultaneous conduction.

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 simultaneous conduction.

OLSEN Bit (Active Level Setting Enable)

This bit enables or disables of 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.

20.2.7 Software Port Output Enable Register (SPOER)

Address(es): POE.SPOER 0008 C4CAh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	MTUCH0HIZ	—	MTUCH34HIZ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MTUCH34HIZ	MTU3 and MTU4 Pin High-Impedance Enable	0: Does not put the outputs in the high-impedance state. 1: Put the outputs in the high-impedance state.	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	MTUCH0HIZ	MTU0 Pin High-Impedance Enable	0: Does not put the outputs in the high-impedance state. 1: Put the outputs in the high-impedance state.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The SPOER register is used to put the outputs of the corresponding pins in the high-impedance state.

MTUCH34HIZ Bit (MTU3 and MTU4 Pin High-Impedance Enable)

This bit specifies whether to put the outputs of the MTU complementary PWM output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D) in the 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

MTUCH0HIZ Bit (MTU0 Pin High-Impedance Enable)

This bit specifies whether to put the outputs of 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

20.2.8 Port Output Enable Control Register 1 (POECR1)

Address(es): POE.POECR1 0008 C4CBh

	b7	b6	b5	b4	b3	b2	b1	b0
	MTU0D1ZE	MTU0C1ZE	MTU0B1ZE	MTU0A1ZE	MTU0DZE	MTU0CZE	MTU0BZE	MTU0AZE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MTU0AZE	MTIOC0A (PB3) Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b1	MTU0BZE	MTIOC0B (PB2) Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b2	MTU0CZE	MTIOC0C (PB1) Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b3	MTU0DZE	MTIOC0D (PB0) Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b4	MTU0A1ZE	MTIOC0A (PD3) Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b5	MTU0B1ZE	MTIOC0B (PD4) Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b6	MTU0C1ZE	MTIOC0C (PD5) Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1
b7	MTU0D1ZE	MTIOC0D (PD6) Pin High-Impedance Enable	0: Does not switch the pin to high-impedance state. 1: Switch the pin to high-impedance state.	R/W*1

Note 1. Can be modified only once after a reset.

The POECR1 register controls high-impedance state of the MTU0 pins.

MTU0AZE Bit (MTIOC0A (PB3) Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC0A output of PB3 to 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, 4; m = 0, 10), or POECMPFR.CnFLAG flag (n = 0 to 2), is set to 1.

MTU0BZE Bit (MTIOC0B (PB2) Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC0B output of PB2 to 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, 4; m = 0, 10), or POECMPFR.CnFLAG flag (n = 0 to 2), is set to 1.

MTU0CZE Bit (MTIOC0C (PB1) Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC0C output of PB1 to 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, 4; m = 0, 10), or POECMPFR.CnFLAG flag (n = 0 to 2), is set to 1.

MTU0DZE Bit (MTIOC0D (PB0) Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC0D output of PB0 to 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, 4; m = 0, 10), or POECMPFR.CnFLAG flag (n = 0 to 2), is set to 1.

MTU0A1ZE Bit (MTIOC0A (PD3) Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC0A output of PD3 to 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, 4; m = 0, 10), or POECMPFR.CnFLAG flag (n = 0 to 2), is set to 1.

MTU0B1ZE Bit (MTIOC0B (PD4) Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC0B output of PD4 to 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, 4; m = 0, 10), or POECMPFR.CnFLAG flag (n = 0 to 2), is set to 1.

MTU0C1ZE Bit (MTIOC0C (PD5) Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC0C output of PD5 to 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, 4; m = 0, 10), or POECMPFR.CnFLAG flag (n = 0 to 2), is set to 1.

MTU0D1ZE Bit (MTIOC0D (PD6) Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC0D output of PD6 to 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, 4; m = 0, 10), or POECMPFR.CnFLAG flag (n = 0 to 2), is set to 1.

20.2.9 Port Output Enable Control Register 2 (POECR2)

Address(es): POE.POECR2 0008 C4CCh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	MTU3BDZE	MTU4ACZE	MTU4BDZE	—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 1.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	MTU4BDZE	MTIOC4B/MTIOC4D Pin High-Impedance Enable	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b9	MTU4ACZE	MTIOC4A/MTIOC4C Pin High-Impedance Enable	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b10	MTU3BDZE	MTIOC3B/MTIOC3D Pin High-Impedance Enable	0: Does not switch the pins to high-impedance state. 1: Switch the pins to high-impedance state.	R/W*1
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR2 register controls high-impedance state of the MTU complementary PWM output pins (MTU3 and MTU4 pins).

MTU4BDZE Bit (MTIOC4B/MTIOC4D Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC4B output and MTIOC4D output to the high-impedance state when at least 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 = 3, 4; m = 8, 10), or POECMPFR.CnFLAG flag (n = 0 to 2) is set to 1.

MTU4ACZE Bit (MTIOC4A/MTIOC4C Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC4A output and MTIOC4C output to the high-impedance state when at least 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 = 3, 4; m = 8, 10), or POECMPFR.CnFLAG flag (n = 0 to 2) is set to 1.

MTU3BDZE Bit (MTIOC3B/MTIOC3D Pin High-Impedance Enable)

This bit specifies whether to switch the MTIOC3B output and MTIOC3D output to the high-impedance state when at least 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 = 3, 4; m = 8, 10), or POECMPFR.CnFLAG flag (n = 0 to 2) is set to 1.

20.2.10 Port Output Enable Control Register 4 (POECR4)

Address(es): POE.POECR4 0008 C4D0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	IC4ADDMT34ZE	IC3ADDMT34ZE	—	—	CMADDMT34ZE
Value after reset:	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMADDMT34ZE	MTU3 and MTU4 High-Impedance Condition CFLAG Add	0: Does not add the flags to the conditions to put the output in the high-impedance state. 1: Adds the flags to the conditions to put the output in the high-impedance state.	R/W*1
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	IC3ADDMT34ZE	MTU3 and MTU4 High-Impedance Condition POE8F Add	0: Does not add the flag to the conditions to put the output in the high-impedance state. 1: Adds the flag to the conditions to put the output in the high-impedance state.	R/W*1
b4	IC4ADDMT34ZE	MTU3 and MTU4 High-Impedance Condition POE10F Add	0: Does not add the flag to the conditions to put the output in the high-impedance state. 1: Adds the flag to the conditions to put the output in the high-impedance state.	R/W*1
b9 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR4 register is used to extend the control conditions to put the output of the MTU complementary PWM output pins (MTU3 and MTU4) in the high-impedance state.

CMADDMT34ZE Bit (MTU3 and MTU4 High-Impedance Condition CFLAG Add)

Adds the POECMPFR.CnFLAG flag (n = 0 to 2) to the high-impedance control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).
However, when the pins are in the high-impedance state by the flag, an OEIn interrupt (n = 1, 3, 4) will not be generated.

IC3ADDMT34ZE Bit (MTU3 and MTU4 High-Impedance Condition POE8F Add)

Adds the ICSR3.POE8F flag 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 Condition POE10F Add)

Adds the ICSR4.POE10F flag to the high-impedance control conditions for the MTU3 and MTU4 pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4C, MTIOC4B, MTIOC4D).

20.2.11 Port Output Enable Control Register 5 (POECR5)

Address(es): POE.POECR5 0008 C4D2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	IC4ADDMT0ZE	—	—	IC1ADDMT0ZE	CMADDMT0ZE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMADDMT0ZE	MTU0 High-Impedance Condition CFLAG Add	0: Does not add the flags to the conditions to put the output in the high-impedance state. 1: Adds the flags to the conditions to put the output in the high-impedance state.	R/W*1
b1	IC1ADDMT0ZE	MTU0 High-Impedance Condition POE0F Add	0: Does not add the flag to the conditions to put the output in the high-impedance state. 1: Adds the flag to the conditions to put the output in the high-impedance state.	R/W*1
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b4	IC4ADDMT0ZE	MTU0 High-Impedance Condition POE10F Add	0: Does not add the flag to the conditions to put the output in the high-impedance state. 1: Adds the flag to the conditions to put the output in the high-impedance state.	R/W*1
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECR5 register is used to extend the control conditions to put the output of the MTU0 pins in the high-impedance state.

CMADDMT0ZE Bit (MTU0 High-Impedance Condition CFLAG Add)

Adds the POECMPFR.CnFLAG flag (n = 0 to 2) to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).
However, when the pins are in the high-impedance state by the flag, an OEIn interrupt (n = 1, 3, 4) will not be generated.

IC1ADDMT0ZE Bit (MTU0 High-Impedance Condition POE0F Add)

Adds the ICSR1.POE0F flag to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

IC4ADDMT0ZE Bit (MTU0 High-Impedance Condition POE10F Add)

Adds the ICSR4.POE10F flag to the high-impedance control conditions for the MTU0 pin (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D).

20.2.12 Port Output Enable Comparator Output Detection Flag Register (POECMPFR)

Address(es): POE.POECMPFR 0008 C4E6h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	C2FLAG	C1FLAG	C0FLAG
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	C0FLAG	Comparator Channel 0 Output Detection Flag	0: Comparator output not detected 1: Comparator output detected	R/(W)*1
b1	C1FLAG	Comparator Channel 1 Output Detection Flag	0: Comparator output not detected 1: Comparator output detected	R/(W)*1
b2	C2FLAG	Comparator Channel 2 Output Detection Flag	0: Comparator output not detected 1: Comparator output detected	R/(W)*1
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The flag can only be set to 0 by writing 0 after reading 1.

CnFLAG Flag (Comparator Channel n Output Detection Flag) (n = 0 to 2)

This flag indicates whether each comparator output is detected or not detected.

[Setting condition]

- A change from low level to high level in the comparator output is detected.
  - When the comparator is set to non-inverted output, the input voltage changes from lower to higher than the reference voltage
  - When the comparator is set to inverted output, the input voltage changes from higher to lower than the reference voltage

[Clearing condition]

- By writing 0 to the CnFLAG flag after reading CnFLAG = 1

20.2.13 Port Output Enable Comparator Request Select Register (POECMPSEL)

Address(es): POE.POECMPSEL 0008 C4E8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	POEREQ2	POEREQ1	POEREQ0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	POEREQ0	Comparator Channel 0 High-Impedance Request Enable	0: Disables high-impedance request generation upon comparator output detection. 1: Enables high-impedance request generation upon comparator output detection.	R/W*1
b1	POEREQ1	Comparator Channel 1 High-Impedance Request Enable	0: Disables high-impedance request generation upon comparator output detection. 1: Enables high-impedance request generation upon comparator output detection.	R/W*1
b2	POEREQ2	Comparator Channel 2 High-Impedance Request Enable	0: Disables high-impedance request generation upon comparator output detection. 1: Enables high-impedance request generation upon comparator output detection.	R/W*1
b15 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

The POECMPSEL register sets a comparator output detection flag to use as a control condition to put the outputs in the high-impedance state.

POEREQn Bit (Comparator Channel n High-Impedance Request Enable) (n = 0 to 2)

This bit disables or enables high-impedance request generation in response to each comparator output detection. A high-impedance request is generated when one of the comparator outputs is detected.

20.3 Operation

The following shows the target pins and conditions for high-impedance control.

(1) MTU3 pins (MTIOC3B, MTIOC3D)

When one of the following conditions is satisfied while the POECR2.MTU3BDZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE0# input level
  When the ICSR1.POE0F flag becomes 1.
- Operation for comparison of the output levels on the MTIOC3B and MTIOC3D pins
  When the OCSR1.OSF1 flag becomes 1 while the OCSR1.OCE1 bit is 1.
- SPOER setting
  When the SPOER.MTUCH34HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT34ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT34ZE bit and the ICSR4.POE10E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPSEL.POEREQ2 bit is 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(2) MTU4 pins (MTIOC4A, MTIOC4C)

When one of the following conditions is satisfied while the POECR2.MTU4ACZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE0# input level
  When the ICSR1.POE0F flag becomes 1.
- Operation for comparison of the output levels on the MTIOC4A and MTIOC4C pins
  When the OCSR1.OSF1 flag becomes 1 while the OCSR1.OCE1 bit is 1.
- SPOER setting
  When the SPOER.MTUCH34HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT34ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT34ZE bit and the ICSR4.POE10E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPSEL.POEREQ2 bit is 1.
- Detection of oscillation stop

When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(3) MTU4 pins (MTIOC4B, MTIOC4D)

When one of the following conditions is satisfied while the POECR2.MTU4BDZE bit is 1, the pins become high-impedance.

- Operation for detection of the POE0# input level
  When the ICSR1.POE0F flag becomes 1.
- Operation for comparison of the output levels on the MTIOC4B and MTIOC4D pins
  When the OCSR1.OSF1 flag becomes 1 while the OCSR1.OCE1 bit is 1.
- SPOER setting
  When the SPOER.MTUCH34HIZ bit is set to 1.
- Conditions added by POECR4
  When the ICSR3.POE8F flag becomes 1 while the POECR4.IC3ADDMT34ZE bit and the ICSR3.POE8E bit are 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR4.IC4ADDMT34ZE bit and the ICSR4.POE10E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR4.CMADDMT34ZE bit is 1 and the POECMPSEL.POEREQ2 bit is 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(4) MTU0 pin PB3 (MTIOC0A)

When one of the following conditions is satisfied while the POECR1.MTU0AZE bit is 1, the pin becomes high-impedance.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ2 bit is 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(5) MTU0 pin PD3 (MTIOC0A)

When one of the following conditions is satisfied while the POECR1.MTU0A1ZE bit is 1, the pin becomes high-

impedance.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ2 bit is 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(6) MTU0 pin PB2 (MTIOC0B)

When one of the following conditions is satisfied while the POECR1.MTU0BZE bit is 1, the pin becomes high-impedance.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ2 bit is 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(7) MTU0 pin PD4 (MTIOC0B)

When one of the following conditions is satisfied while the POECR1.MTU0B1ZE bit is 1, the pin becomes high-impedance.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.

- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ2 bit is 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(8) MTU0 pin PD5 (MTIOC0C)

When one of the following conditions is satisfied while the POECR1.MTU0C1ZE bit is 1, the pin becomes high-impedance.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ2 bit is 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(9) MTU0 pin PB1 (MTIOC0C)

When one of the following conditions is satisfied while the POECR1.MTU0CZE bit is 1, the pin becomes high-impedance.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the

POECMPSEL.POEREQ0 bit is 1.
When the POECMPFR.C1FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ1 bit is 1.
When the POECMPFR.C2FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ2 bit is 1.

- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(10) MTU0 pin PD6 (MTIOC0D)

When one of the following conditions is satisfied while the POECR1.MTU0D1ZE bit is 1, the pin becomes high-impedance.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ2 bit is 1.
- Detection of oscillation stop
  When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

(11) MTU0 pin PB0 (MTIOC0D)

When one of the following conditions is satisfied while the POECR1.MTU0DZE bit is 1, the pin becomes high-impedance.

- Operation for detection of the POE8# input level
  When the ICSR3.POE8F flag becomes 1 while the ICSR3.POE8E bit is 1.
- SPOER setting
  When the SPOER.MTUCH0HIZ bit is set to 1.
- Conditions added by POECR5
  When the ICSR1.POE0F flag becomes 1 while the POECR5.IC1ADDMT0ZE bit is 1.
  When the ICSR4.POE10F flag becomes 1 while the POECR5.IC4ADDMT0ZE bit and the ICSR4.POE10E bit are 1.
- Comparator output detection
  When the POECMPFR.C0FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ0 bit is 1.
  When the POECMPFR.C1FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ1 bit is 1.
  When the POECMPFR.C2FLAG flag becomes 1 while the POECR5.CMADDMT0ZE bit is 1 and the POECMPSEL.POEREQ2 bit is 1.
- Detection of oscillation stop

When the ICSR6.OSTSTF flag becomes 1 while the ICSR6.OSTSTE bit is 1.

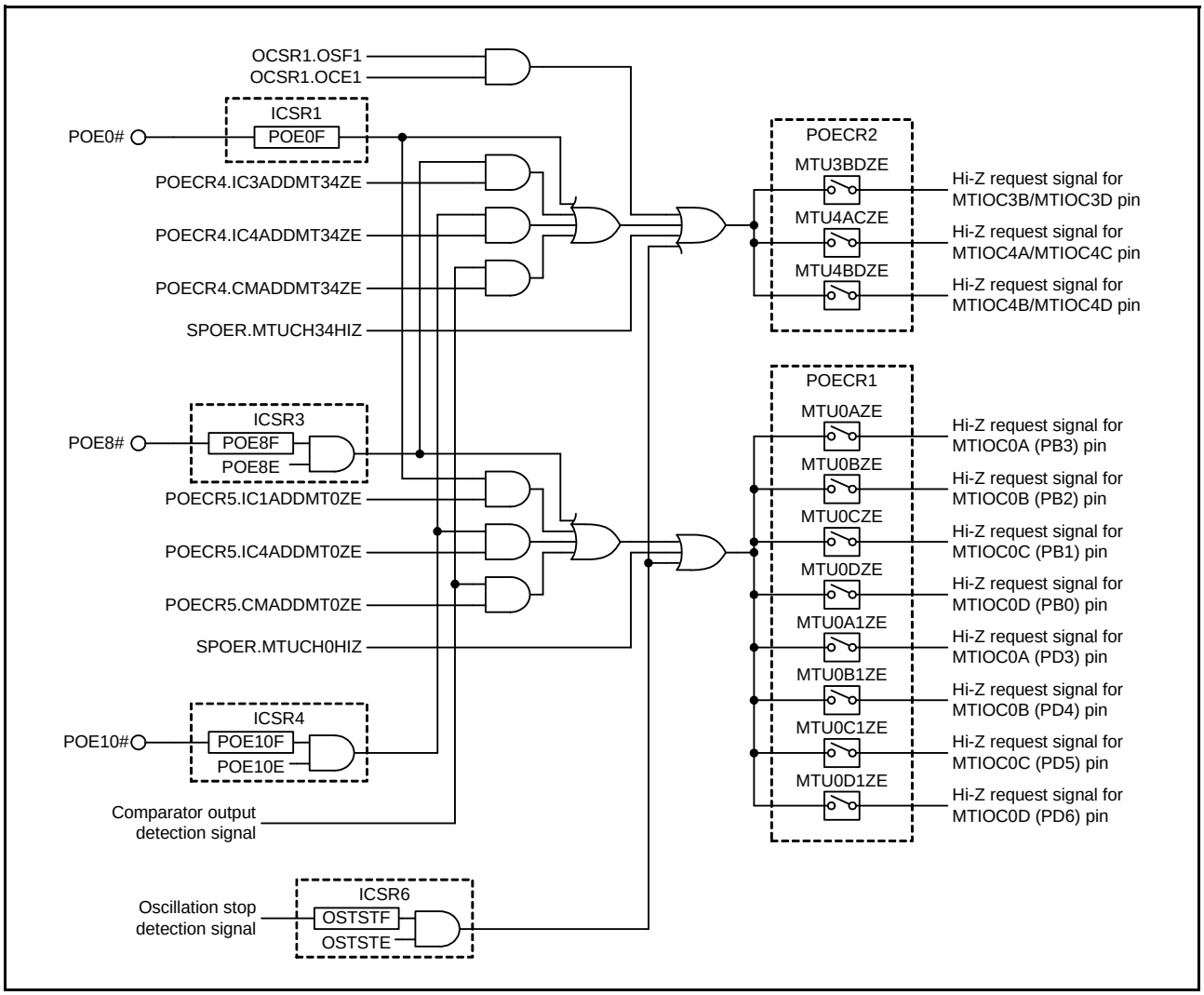


Figure 20.2 Target Pins and Conditions for High-Impedance Control

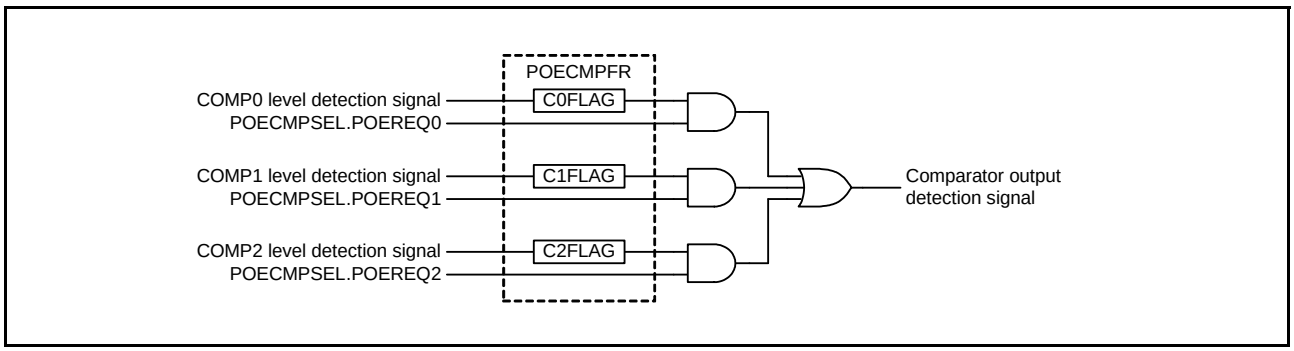


Figure 20.3 Comparator Output Detection Signal Generator Block Diagram

20.3.1 Input-Level Detection Operation

If the input conditions set by ICSR1 to ICSR4 occur on the POE0#, POE8#, and POE10# pins, the outputs of the MTU complementary PWM output pins (MTU3 and MTU4) and MTU0 pins are in the high-impedance state. Note however, that these outputs are still in the high-impedance state even when the MTU functions are not selected for the pins.

(1) Falling Edge Detection

When a change from a high to low level is input to the POE0#, POE8#, and POE10# pins, the outputs of the pins multiplexed with MTU complementary PWM output pins and MTU0 pins are in the high-impedance state.
The falling edge is detected after the level is sampled with PCLK. Input a low level for at least one PCLK clock to the POE0#, POE8#, and POE10# pins.
Figure 20.4 shows a sample timing after the level changes in input to the POE0#, POE8#, and POE10# pins until the respective pins become high-impedance.

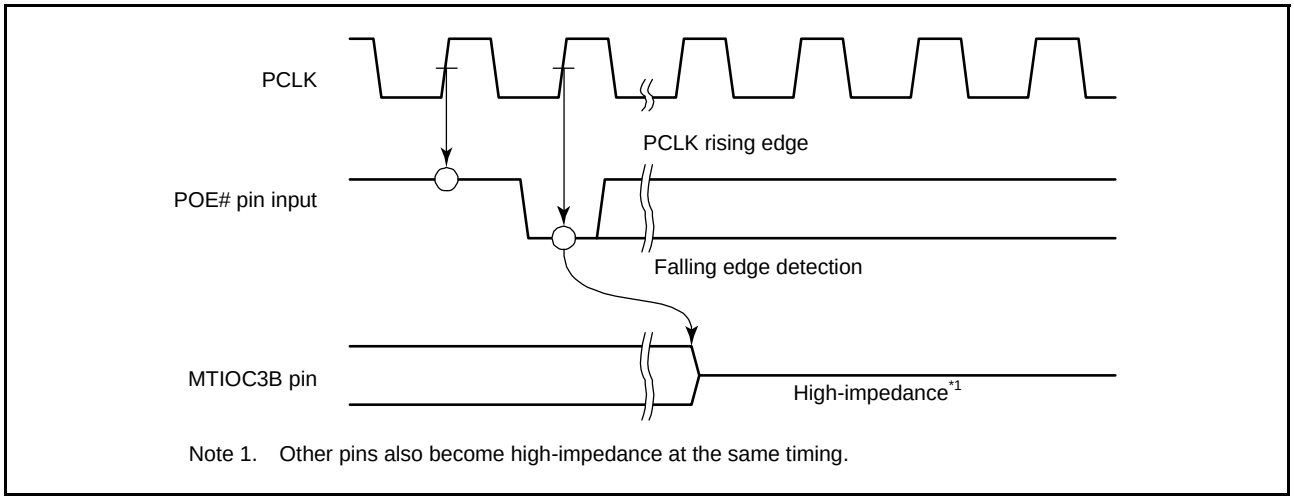
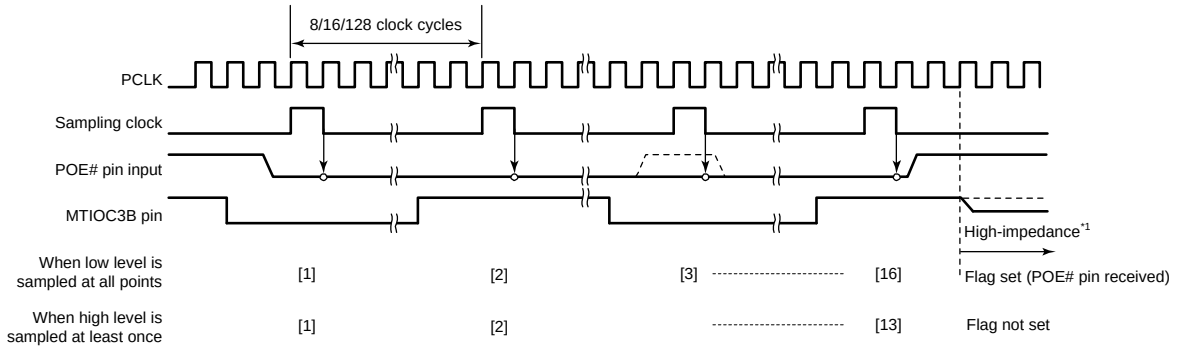


Figure 20.4 Operation when A Falling Edge Detection is Selected

(2) Low-Level Detection

Figure 20.5 shows an example of operation when a pin is placed in the high-impedance state in response to low-level detection. When 16 continuous low levels are sampled with the sampling clock selected by the ICSR1 to ICSR4 registers, the low level is recognized and the outputs of the MTU complementary PWM output pins and MTU0 pins are in the high-impedance state. If even one high level is detected during this interval, the low level is not recognized.
The timing when the outputs of the MTU complementary PWM output pins and MTU0 pins are in the high-impedance state after the sampling clock is input is the same in both falling-edge detection and in low-level detection.

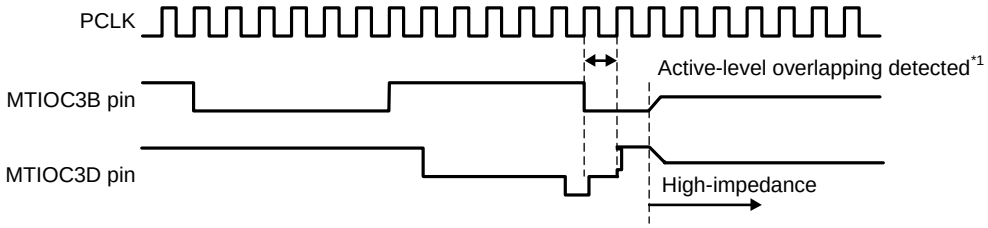


Note 1. Other pins also become high-impedance at the same timing.

Figure 20.5 Operation when A Low-Level Detection is Selected

20.3.2 Output-Level Compare Operation

Figure 20.6 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.



Note 1. When the active level of both MTIOC3B and MTIOC3D pins is set to low.

Figure 20.6 Output-Level Compare Operation

20.3.3 High-Impedance Control Using Registers

The high-impedance request of the MTU pins (MTU0, MTU3, and MTU4) can be directly controlled by using the SPOER register.
For instance, setting the SPOER.MTUCH34HIZ bit to 1 switches the MTU3 and MTU4 pins specified by the POECR2 register to the high-impedance state.
The high-impedance request of other pins can also be controlled by setting the appropriate bits in the SPOER register.

20.3.4 High-Impedance Control through Detection of Oscillation Stop

When oscillation stop is detected by the oscillation stop detection function of the clock generator while the ICSR6.OSTSTE bit is 1, the MTU complementary PWM output pins specified by the POECR2 register and the MTU0 pins specified by the POECR1 register are switched to the high-impedance state.

20.3.5 High-Impedance Control through Detection of the Comparator Output

The outputs of the MTU complementary PWM output pins and MTU0 pins can be in the high-impedance state in response to detection of the output from the comparator.
For instance, when the POECMPFR.CnFLAG flag (n = 0 to 2) is added to the high-impedance control conditions for the MTU3 and MTU4 pins by setting the POECR4.CMADDMT34ZE bit to 1, the MTU3 and MTU4 pins specified by the POECR2 register become high-impedance on comparator output detection.
The high-impedance control of other pins can be controlled by the POECR1 to POECR5 registers.

20.3.6 Additional Functions for High-Impedance Control

High-impedance control conditions for the MTU complementary PWM output pins and MTU0 pins can be added by setting the POECR4 and POECR5 registers.
For instance, the settings listed below can be added as high-impedance control conditions for the MTU3 and MTU4 pins.

- Setting the POECR4.CMADDMT34ZE bit to 1 adds comparator output detection
- 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

The high-impedance control of other pins can also be controlled by setting the appropriate bits in the POECR4 and POECR5 registers.

20.3.7 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, ICSR3.POE8F, and ICSR4.POE10F flags. However, note that when low-level sampling is selected with the ICSR1.POE0M[1:0], ICSR3.POE8M[1:0], and ICSR4.POE10M[1:0] bits, just writing 0 to a flag is ignored (the flag is not set to 0); flags can be cleared by writing 0 to it only after a high level is input to the POE0#, POE8#, and POE10# 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 to 0. However, note that just writing 0 to a flag is ignored (the flag is not set to 0); the flags 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 comparator output detection can be recovered from the

state either by returning them to their initial state with a reset or by setting the POECMPFR.CnFLAG flag (n = 0 to 2) to 0.

When setting the POECMPFR.CnFLAG flag to 0, be sure to confirm that the analog input signal that triggered comparator output detection has returned to a normal value by performing A/D conversion and so on.

Note that the above POECMPFR.CnFLAG flag is not set to 1 again in the following cases:

- This flag is cleared without confirmation that the analog input signal has returned to a normal value, and
  1. the analog input signal remains above the reference voltage when the comparator is set to non-inverted output, or
  2. the analog input signal remains below the reference voltage when the comparator is set to inverted output.

The outputs which have been in the high-impedance state due to oscillation stop detection can be recovered from the state either by returning them to their initial state with a reset or by setting the SYSTEM.OSTDSR.OSTDF flag to 0 to set the ICSR6.OSTSTF flag to 0.

20.4 POE Setting Procedure

Figure 20.7 shows the procedure for setting the POE. It illustrates an example of high-impedance control in response to comparison of the output levels on the MTU3 pins (MTIOC3B/MTIOC3D). In the figure, P71 is used as the MTIOC3B pin and P74 is used as the MTIOC3D pin.

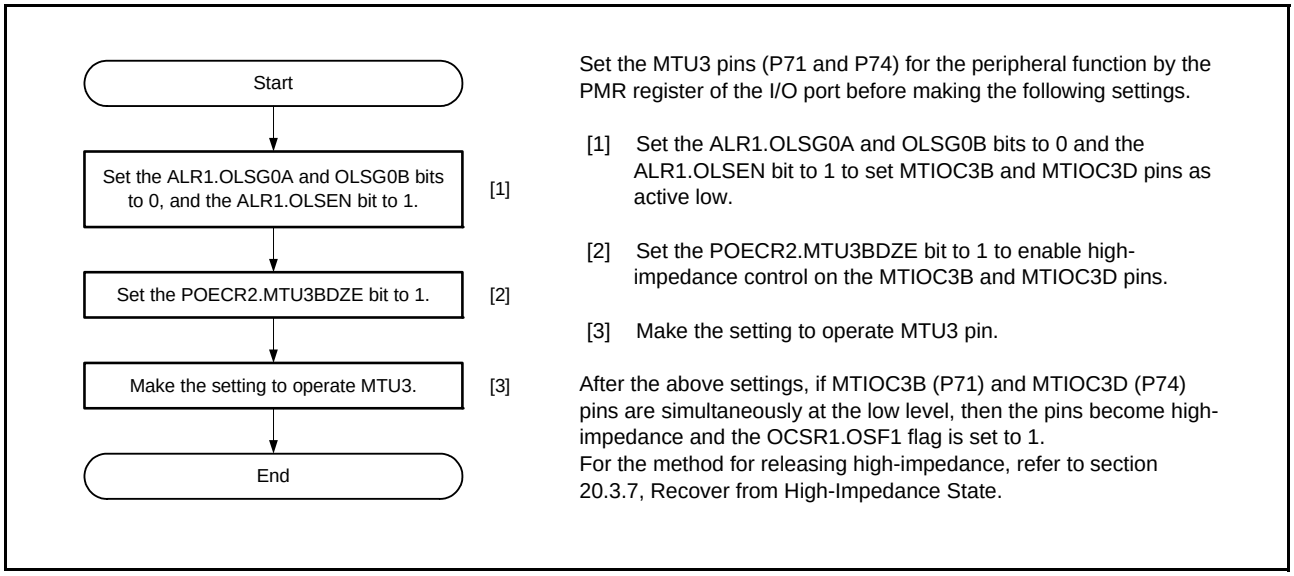


Figure 20.7 Procedure for Setting the POE

20.5 Interrupts

The POE issues a request to generate an interrupt when the specified condition is satisfied during input-level detection or output-level comparison. Table 20.4 shows the interrupt sources and their conditions.

Table 20.4 Interrupt Sources and Conditions

Name	Interrupt Source	Interrupt Flag	Condition
OEI1	Output enable interrupt 1	POE0F OSF1	When the ICSR1.POE0F flag becomes 1 while the ICSR1.PIE1 bit is 1 or when the OCSR1.OSF1 flag becomes 1 while the OCSR1.OIE1 bit is 1
OEI3	Output enable interrupt 3	POE8F	When the ICSR3.POE8F flag becomes 1 while the ICSR3.PIE3 bit is 1
OEI4	Output enable interrupt 4	POE10F	When the ICSR4.POE10F flag is set to 1 while the ICSR4.PIE4 bit is 1

20.6 Usage Notes

20.6.1 Transition to Low Power Consumption Mode

When the POE is used, do not make a transition to software standby mode. In this mode, the POE stops and thus the high-impedance control of pins cannot operate.

20.6.2 High-Impedance Control When the MTU is Not Selected

If high-impedance control for a pin having a multiplexed MTU pin function is enabled by setting the POECR1 and POECR2 registers and the high-impedance control condition is satisfied, the output is to be in the high-impedance state even if the MTU function is not selected for the pin on which it is multiplexed.

20.6.3 When the POE is Not Used

The high-impedance control of some pins can be enabled using the POE after a reset. When the POE is not used, write 0 to the target bits in the POECR1 and POECR2 registers.

21. Compare Match Timer (CMT)

This MCU has an on-chip compare match timer (CMT) unit (unit 0) consisting of a two-channel 16-bit timer (i.e., a total of two channels). The CMT has a 16-bit counter, and can generate interrupts at set intervals.
In this section, “PCLK” is used to refer to PCLKB.

21.1 Overview

Table 21.1 lists the specifications for the CMT.
Figure 21.1 shows a block diagram of the CMT (unit 0). A two-channel CMT constitutes a unit.

Table 21.1 CMT Specifications

Item	Description
Count clocks	• Four frequency dividing clocks One clock from PCLK/8, PCLK/32, PCLK/128, and PCLK/512 can be selected for each channel.
Interrupt	A compare match interrupt can be requested for each channel.
Low power consumption function	Module stop state can be set.

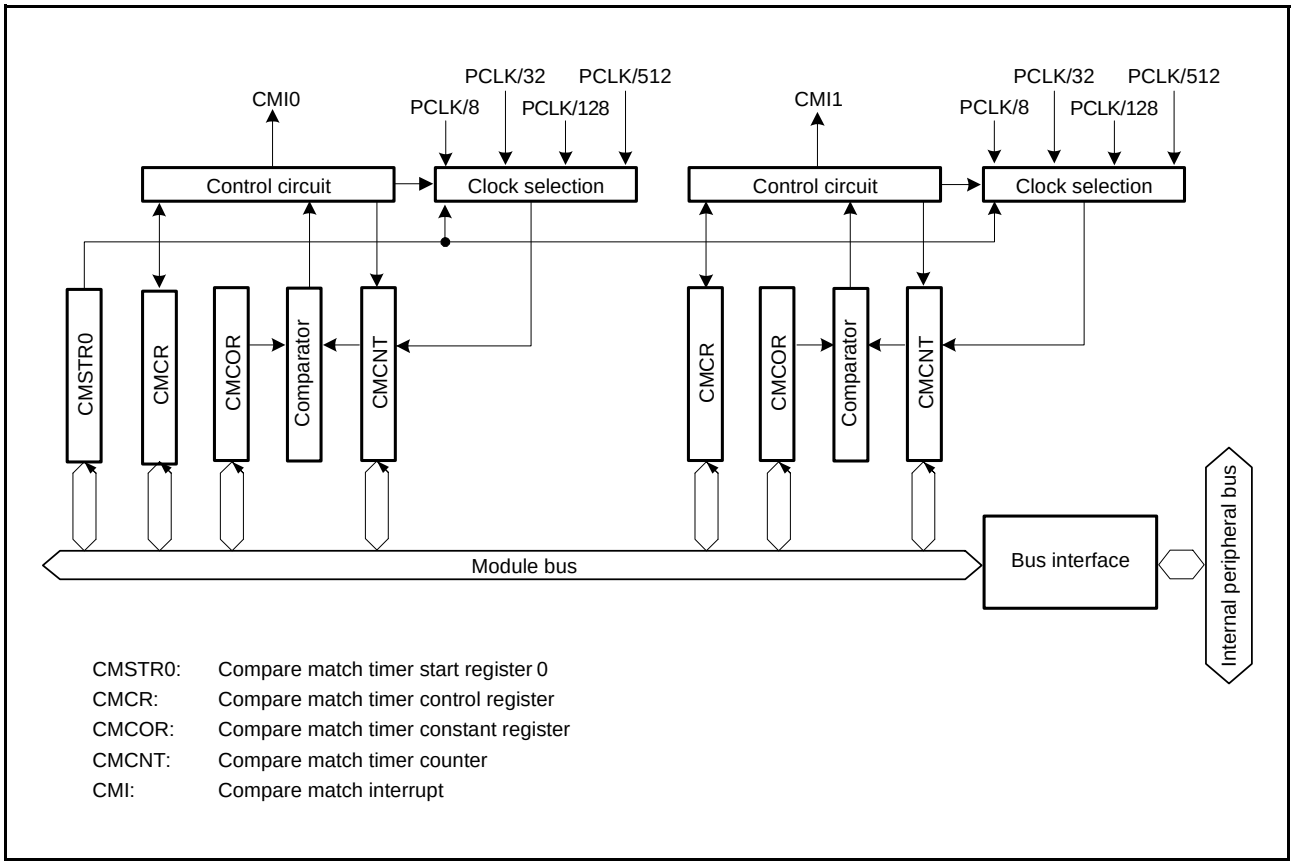


Figure 21.1 CMT (Unit 0) Block Diagram

21.2 Register Descriptions

21.2.1 Compare Match Timer Start Register 0 (CMSTR0)

Address(es): 0008 8000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	STR1	STR0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	STR0	Count Start 0	0: CMT0.CMCNT count is stopped. 1: CMT0.CMCNT count is started.	R/W
b1	STR1	Count Start 1	0: CMT1.CMCNT count is stopped. 1: CMT1.CMCNT count is started.	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

21.2.2 Compare Match Timer Control Register (CMCR)

Address(es): CMT0.CMCR 0008 8002h, CMT1.CMCR 0008 8008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	CMIE	—	—	—	—	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK/8 0 1: PCLK/32 1 0: PCLK/128 1 1: PCLK/512	R/W
b5 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	CMIE	Compare Match Interrupt Enable	0: Compare match interrupt (CMIn) disabled 1: Compare match interrupt (CMIn) enabled	R/W
b7	—	Reserved	This bit is read as undefined. The write value should be 1.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

CKS[1:0] Bits (Clock Select)

These bits select the count source from four frequency dividing clocks obtained by dividing the peripheral module clock (PCLK).

When the CMSTR0.STRn (n = 0, 1) bit is set to 1, the CMCNT counter starts counting up on the clock selected with the CKS[1:0] bits.

CMIE Bit (Compare Match Interrupt Enable)

The CMIE bit enables or disables compare match interrupt (CMIn) (n = 0, 1) generation when the CMCNT counter and the CMCOR register values match.

21.2.3 Compare Match Counter (CMCNT)

Address(es): CMT0.CMCNT 0008 8004h, CMT1.CMCNT 0008 800Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The CMCNT counter is a readable/writable up-counter.
When an frequency dividing clock is selected by the CMCR.CKS[1:0] bits and the CMSTR0.STRn (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 set to 0000h. At the same time, a compare match interrupt (CMIn) (n = 0, 1) is generated.

21.2.4 Compare Match Constant Register (CMCOR)

Address(es): CMT0.CMCOR 0008 8006h, CMT1.CMCOR 0008 800Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The CMCOR register is a readable/writable register to set a value for compare match with the CMCNT counter.

21.3 Operation

21.3.1 Periodic Count Operation

When an frequency dividing clock is selected by the CMCR.CKS[1:0] bits and the CMSTR0.STRn (n = 0, 1) bit is set to 1, the CMCNT counter starts counting up using the selected clock.
When the value in the counter and the value in the register match, a compare match interrupt (CMIn) (n = 0,1) is generated. The CMCNT counter then starts counting up again from 0000h. Figure 21.2 shows the operation of the CMCNT counter.

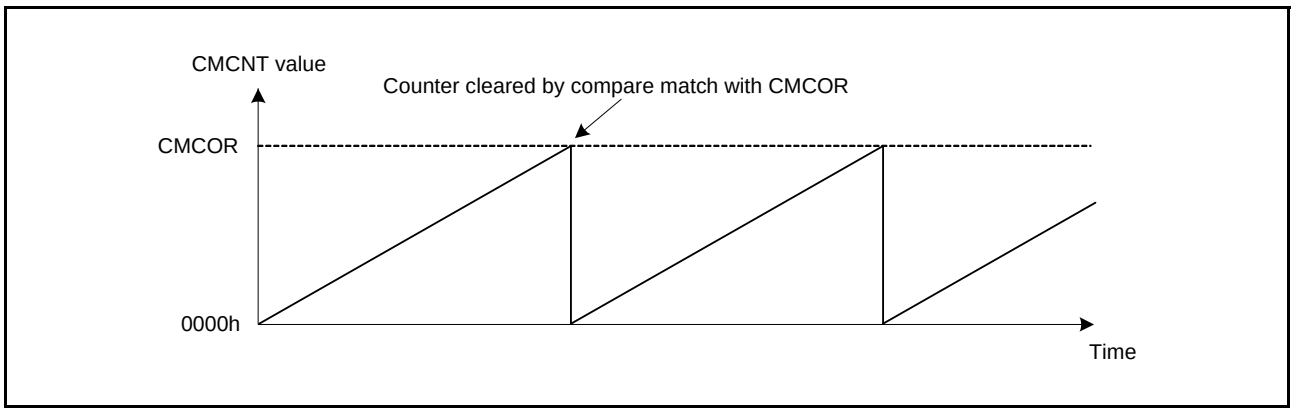


Figure 21.2 CMCNT Counter Operation

21.3.2 CMCNT Count Timing

As the count clock to be input to the CMCNT counter, one of four frequency dividing clocks (PCLK/8, PCLK/32, PCLK/128, and PCLK/512) obtained by dividing the peripheral module clock (PCLK) can be selected with the CMCR.CKS[1:0] bits. Figure 21.3 shows the timing of the CMCNT counter.

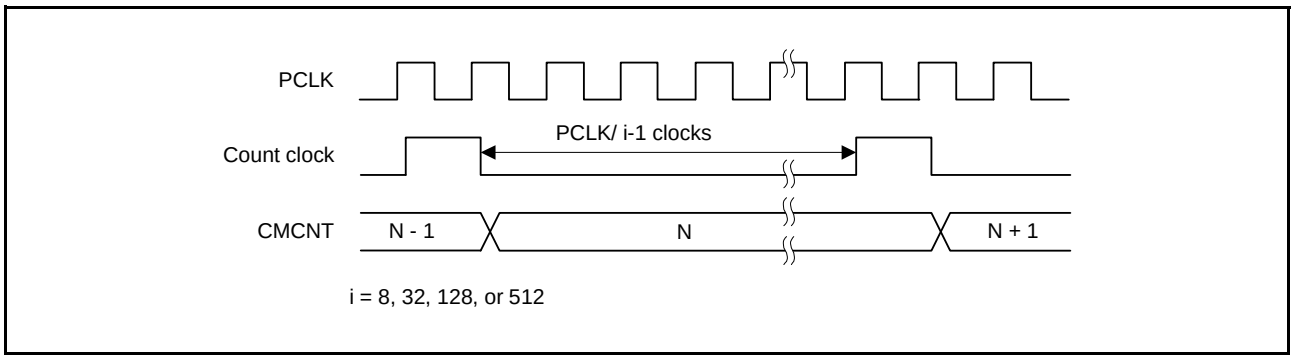


Figure 21.3 CMCNT Count Timing

21.4 Interrupts

21.4.1 Interrupt Sources

The CMT has channels and each of them to which a different vector address is allocated has a compare match interrupt (CMIn) (n = 0, 1). When a compare match interrupt occurs, the corresponding interrupt request is output.
When the interrupt request is used to generate a CPU interrupt, the priority of channels can be changed by the interrupt controller settings. For details, see section 14, Interrupt Controller (ICUb).

Table 21.2 CMT Interrupt Sources

Name	Interrupt Sources	DTC Activation
CMI0	Compare match in CMT0	Possible
CMI1	Compare match in CMT1	Possible

21.4.2 Timing of Compare Match Interrupt Generation

When the CMCNT counter and the CMCOR register match, a compare match interrupt (CMIn) (n = 0, 1) 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 next the CMCNT counter input clock.
Figure 21.4 shows the timing of a compare match interrupt.

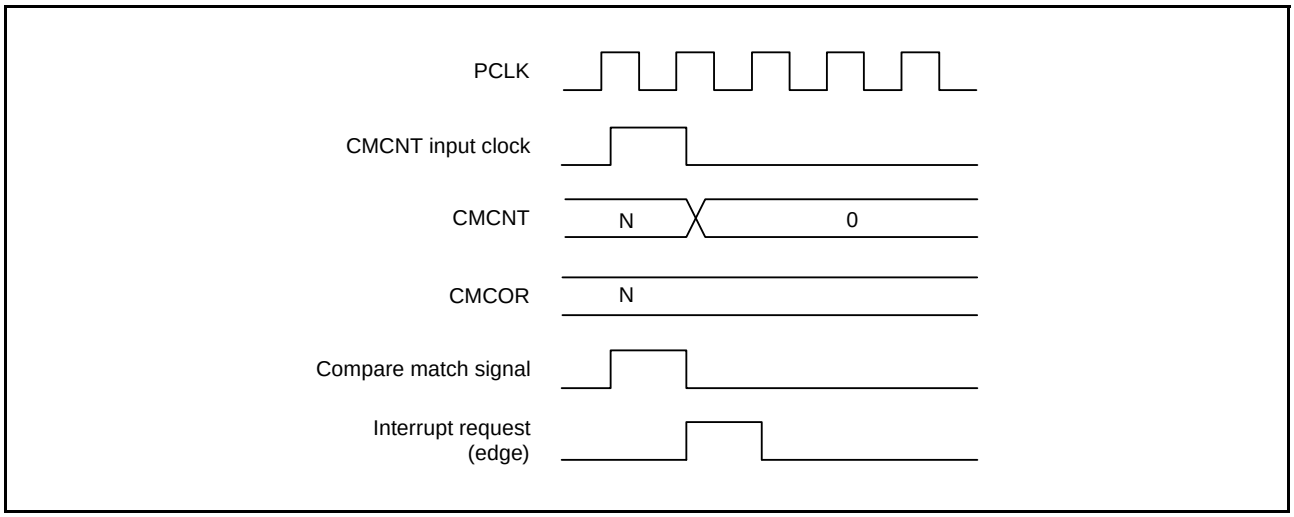


Figure 21.4 Timing of a Compare Match Interrupt

21.5 Usage Notes

21.5.1 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 releasing the module stop state. For details, refer to section 11, Low Power Consumption.

21.5.2 Conflict between CMCNT Counter Writing and Compare Match

When the compare match signal is generated while writing to the CMCNT counter, clearing the CMCNT counter has priority over writing to it. In this case, the CMCNT counter is not written to. Figure 21.5 shows the timing to clear the CMCNT counter.

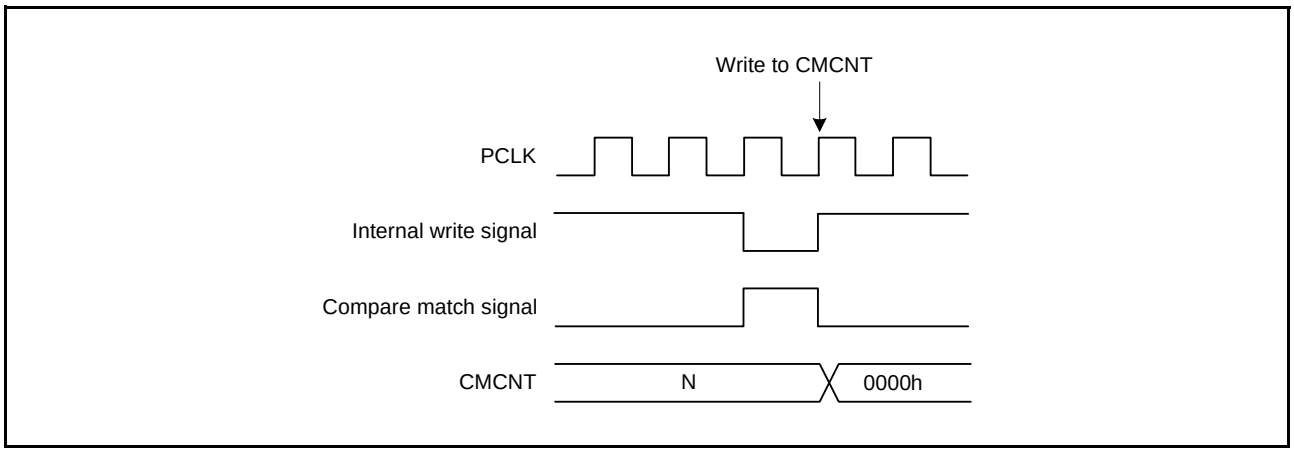


Figure 21.5 Conflict between CMCNT Counter Writing and Compare Match

21.5.3 Conflict between CMCNT Counter Writing and Incrementing

If writing to the counter and the incrementing conflict, the writing has priority over the incrementing. Figure 21.6 shows the timing to write the CMCNT counter.

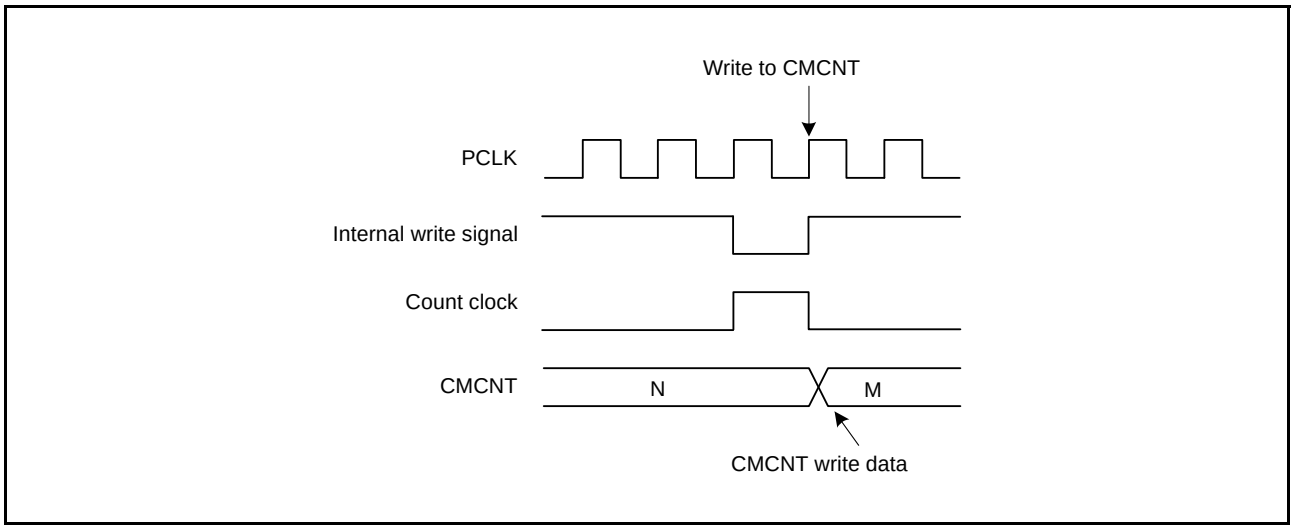


Figure 21.6 Conflict between CMCNT Counter Writing and Incrementing

22. Independent Watchdog Timer (IWDTa)

In this section, “PCLK” is used to refer to PCLKB.

22.1 Overview

The independent watchdog timer (IWDT) can be used to detect programs being out of control.
The user can detect when a program runs out of control if an underflow occurs, by creating a program that refreshes the IWDT counter before it underflows.
The functions of the IWDT are different from those of the WDT in the following respects.

- The divided IWDT-dedicated clock (IWDTCLK) is used as the count source (not affected by the PCLK).
- When making a transition to sleep mode, software standby mode, or deep sleep mode, the IWDTCSTPR.SLCSTP bit or the OFS0.IWDTSLCSTP bit can be used to select whether to stop the counter or not.

Table 22.1 lists the specifications of the IWDT and Figure 22.1 shows a block diagram of the IWDT.

Table 22.1 IWDT Specifications

Item	Description
Count source*1	IWDT-dedicated clock (IWDTCLK)
Clock divide ratio	Divide by 1, 16, 32, 64, 128, or 256
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	• Auto-start mode: Counting automatically starts after a reset is released • Register start mode: Counting is started by refresh operation (writing 00h and then FFh to the IWDTRR register).
Conditions for stopping the counter	• Reset (the down-counter and other registers return to their initial values) • In low power consumption states (depends on the register setting*2) • A counter underflows or a refresh error occurs (only in register start mode)
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Reset output sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Non-maskable interrupt sources	• Down-counter underflows • Refreshing outside the refresh-permitted period (refresh error)
Reading the counter value	The down-counter value can be read by the IWDTSR register.
Output signal (internal signal)	• Reset output • Interrupt request output • Sleep mode count stop control output
Auto-start mode (controlled by option function select register 0 (OFS0))	• Selecting the clock frequency divide ratio after a reset (OFS0.IWDTCKS[3:0] bits) • Selecting the timeout period of the independent watchdog timer (OFS0.IWDTTOPS[1:0] bits) • Selecting the window start position in the independent watchdog timer (OFS0.IWDTRPSS[1:0] bits) • Selecting the window end position in the independent watchdog timer (OFS0.IWDTRPES[1:0] bits) • Selecting the reset output or interrupt request output (OFS0.IWDTRSTIRQS bit) • Selecting the down-count stop function at transition to sleep mode, software standby mode, or deep sleep mode (OFS0.IWDTSLCSTP bit)
Register start mode (controlled by the IWDT registers)	• Selecting the clock frequency divide ratio after refreshing (IWDTCR.CKS[3:0] bits) • Selecting the timeout period of the independent watchdog timer (IWDTCR.TOPS[1:0] bits) • Selecting the window start position in the independent watchdog timer (IWDTCR.RPSS[1:0] bits) • Selecting the window end position in the independent watchdog timer (IWDTCR.RPES[1:0] bits) • Selecting the reset output or interrupt request output (IWDTRCR.RSTIRQS bit) • Selecting the down-count stop function at transition to sleep mode, software standby mode, or deep sleep mode (IWDTCSTPR.SLCSTP bit)

Note 1. Satisfy the frequency of the peripheral module clock (PCLK) ≥ 4 × (the frequency of the count source after divide).
Note 2. When the OFS0.IWDTSLCSTP bit is 1 in auto-start mode, and when the IWDTCSTPR.SLCSTP bit is 1 in register start mode.

To use the IWDT, the IWDT-dedicated clock (IWDTCLK) should be supplied so that the IWDT operates even if the peripheral module clock (PCLK) stops. The bus interface and registers operate with PCLK, and the 14-bit counter and control circuits operate with IWDTCLK.
Figure 22.1 is a block diagram of the IWDT.

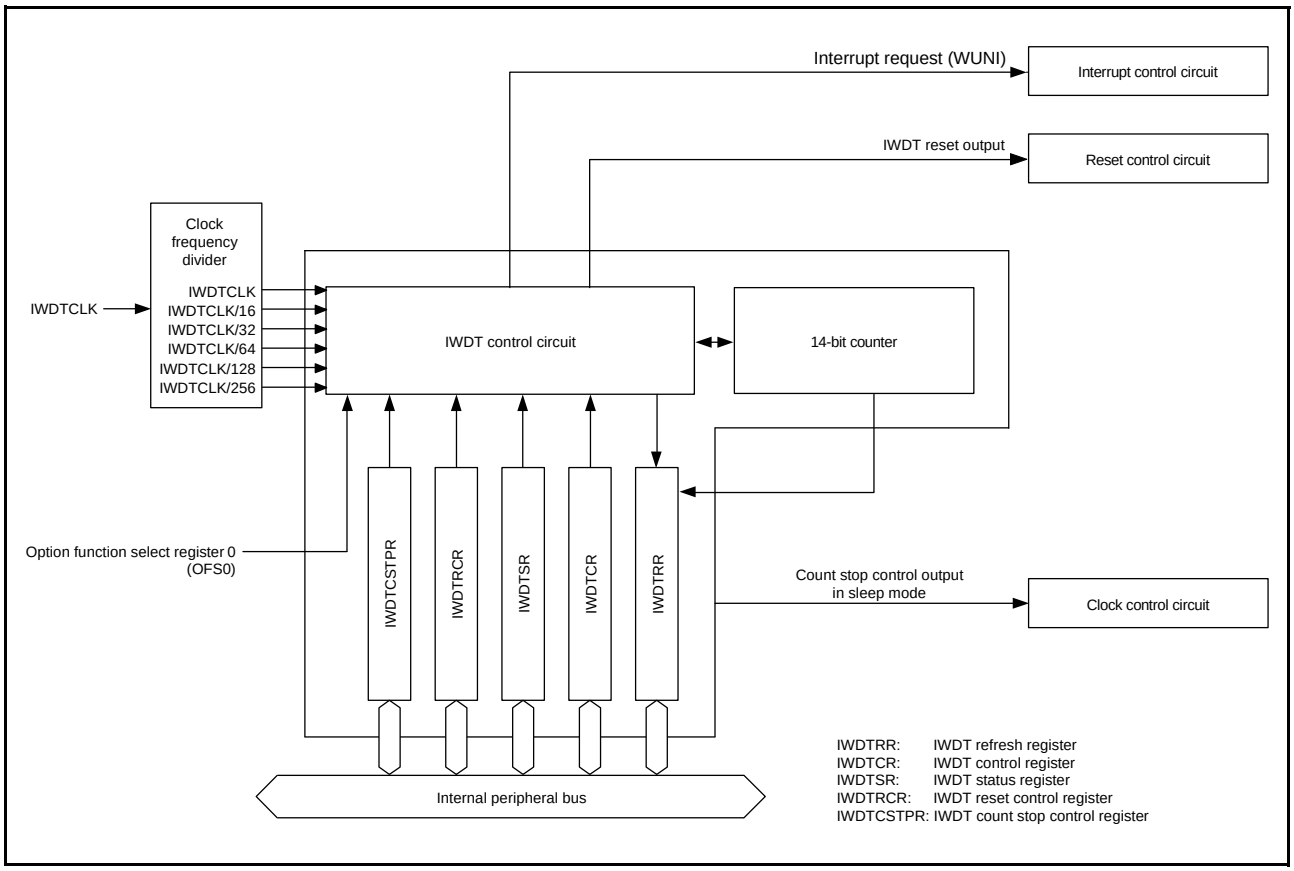
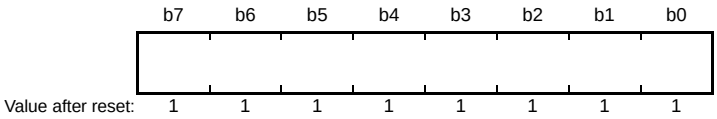


Figure 22.1 IWDT Block Diagram

22.2 Register Descriptions

22.2.1 IWDT Refresh Register (IWDTRR)

Address(es): IWDT.IWDTRR 0008 8030h



Bit	Description	R/W
b7 to b0	The counter is refreshed by writing 00h and then writing FFh to this register.	R/W

The IWDTRR register refreshes the counter of the IWDT.
The counter of the IWDT is refreshed by writing 00h and then writing FFh to the IWDTRR register (refresh operation) within the refresh-permitted period.
After the counter has been refreshed, it starts counting down from the value selected by the IWDTTOPS[1:0] bits in option function select register 0 (OFS0) in auto-start mode. In register start mode, counting down starts from the value selected by setting the IWDTCR.TOPS[1:0] bits in the first refresh operation after a reset is released.
When 00h is written, the read value is 00h. When a value other than 00h is written, the read value is FFh.
For details of the refresh operation, refer to section 22.3.3, Refresh Operation.

22.2.2 IWDT Control Register (IWDTCR)

Address(es): IWDT.IWDTCR 0008 8032h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	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	Bit Name	Description	R/W
b1, b0	TOPS[1:0]	Timeout Period Select	b1 b0 0 0: 128 cycles (007Fh) 0 1: 512 cycles (01FFh) 1 0: 1024 cycles (03FFh) 1 1: 2048 cycles (07FFh)	R/W
b3, b2	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7 to b4	CKS[3:0]	Clock Divide Ratio Select	b7 b4 0 0 0 0: No division 0 0 1 0: Divide-by-16 0 0 1 1: Divide-by-32 0 1 0 0: Divide-by-64 1 1 1 1: Divide-by-128 0 1 0 1: Divide-by-256 Other settings are prohibited.	R/W
b9, b8	RPES[1:0]	Window End Position Select	b9 b8 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (window end position is not specified.)	R/W
b11, b10	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b13, b12	RPSS[1:0]	Window Start Position Select	b13 b12 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (window start position is not specified.)	R/W
b15, b14	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

There are some restrictions on writing to the IWDTCR register. For details, refer to section 22.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the settings in the IWDTCR register are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting made to the IWDTCR register can also be made in the OFS0 register. For details, refer to section 22.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

TOPS[1:0] Bits (Timeout Period Select)

These bits select the timeout period (period until the counter underflows) from among 128, 512, 1024, or 2048 cycles, taking the divided clock specified by the CKS[3:0] bits as one cycle.
After the counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of IWDTCLK cycles) until the counter underflows.
Relations between the CKS[3:0] and TOPS[1:0] bit setting, the timeout period, and the number of IWDTCLK cycles are listed in Table 22.2.

Table 22.2 Settings and Timeout Periods

CKS[3:0] Bits				TOPS[1:0] Bits				
b7	b6	b5	b4	b1	b0	Clock Divide Ratio	Timeout Period (Number of Cycles)	Cycles of IWDTCLK
0	0	0	0	0	0	No division	128	128
				0	1		512	512
				1	0		1024	1024
				1	1		2048	2048
0	0	1	0	0	0	Divide-by-16	128	2048
				0	1		512	8192
				1	0		1024	16384
				1	1		2048	32768
0	0	1	1	0	0	Divide-by-32	128	4096
				0	1		512	16384
				1	0		1024	32768
				1	1		2048	65536
0	1	0	0	0	0	Divide-by-64	128	8192
				0	1		512	32768
				1	0		1024	65536
				1	1		2048	131072
1	1	1	1	0	0	Divide-by-128	128	16384
				0	1		512	65536
				1	0		1024	131072
				1	1		2048	262144
0	1	0	1	0	0	Divide-by-256	128	32768
				0	1		512	131072
				1	0		1024	262144
				1	1		2048	524288

CKS[3:0] Bits (Clock Divide Ratio Select)

These bits select the IWDTCLK clock divide ratio from among divide-by 1, 16, 32, 64, 128, and 256. Combination with the TOPS[1:0] bit setting, a count period between 128 and 524288 cycles of the IWDTCLK clock can be selected for the IWDT.

RPES[1:0] Bits (Window End Position Select)

These bits select 75%, 50%, 25% or 0% of the count period for the window end position of the counter. The 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 enabled.
The counter values for the window start and end positions selected by setting the RPSS[1:0] and RPES[1:0] bits change depending on the TOPS[1:0] bit setting.
Table 22.3 lists the counter values for the window start and end positions corresponding to TOPS[1:0] bit values.

Table 22.3 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	128	007Fh	007Fh	005Fh	003Fh	001Fh
0	1	512	01FFh	01FFh	017Fh	00FFh	007Fh
1	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
1	1	2048	07FFh	07FFh	05FFh	03FFh	01FFh

RPSS[1:0] Bits (Window Start Position Select)

These bits select a counter window start position from 100%, 75%, 50%, or 25% of the count period (100% when the count starts and 0% when the counter underflows). The interval between the window start position and window end position is the refresh-permitted period and the other periods are refresh-prohibited periods.
Figure 22.2 shows the relationship between of the RPSS[1:0] and RPES[1:0] bit setting and the refresh-permitted and refresh-prohibited periods.

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 is greater than or equal to window start setting, the window end setting is set to 0%.

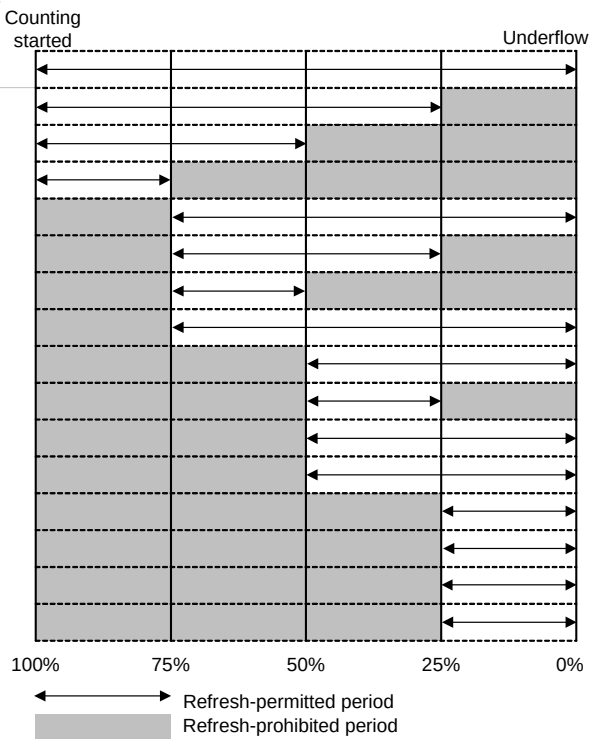


Figure 22.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period

22.2.3 IWDT Status Register (IWDTSR)

Address(es): IWDT.IWDTSR 0008 8034h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	REFEF	UNDFF	CNTVAL[13:0]													
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	CNTVAL[13:0]	Counter Value	Value counted by the counter	R
b14	UNDFF	Underflow Flag	0: No underflow occurred 1: Underflow occurred	R/(W) *1
b15	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 IWDTSR register is initialized by the reset source of the IWDT. The IWDTSR register is not initialized by other reset sources.

CNTVAL[13:0] Bits (Counter Value)

These bits are used to confirm the counter value of the counter, but note that the read value may differ from the actual count by a value of one count.

UNDFF Flag (Underflow Flag)

This bit is used to confirm whether or not an underflow has occurred in the counter.
The value 1 indicates that the counter has underflowed. The value 0 indicates that the counter has not underflowed.
Write 0 to the UNDFF flag to set the value to 0. Writing 1 has no effect.

REFEF Flag (Refresh Error Flag)

This bit is used to confirm whether or not a refresh error (performing a refresh operation during a refresh-prohibited period).
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.

22.2.4 IWDT Reset Control Register (IWDTRCR)

Address(es): IWDT.IWDTRCR 0008 8036h

	b7	b6	b5	b4	b3	b2	b1	b0
	RSTIRQS	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	RSTIRQS	Reset Interrupt Request Select	0: Non-maskable interrupt request output is enabled. 1: Reset output is enabled.	R/W

There are some restrictions on writing to the IWDTRCR register. For details, refer to section 22.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the IWDTRCR register setting are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting mode to the IWDTRCR register can also be made in the OFS0 register. For details, refer to section 22.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

22.2.5 IWDT Count Stop Control Register (IWDTCSTPR)

Address(es): IWDT.IWDTCSTPR 0008 8038h

	b7	b6	b5	b4	b3	b2	b1	b0
	SLCSTP	—	—	—	—	—	—	—
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	SLCSTP	Sleep Mode Count Stop Control	0: Count stop is disabled. 1: Count is stopped at a transition to sleep mode, software standby mode, or deep sleep mode.	R/W

The IWDTCSTPR register controls whether to stop the IWDT counter in a low power consumption state. There are some restrictions on writing to the IWDTCSTPR register. For details, refer to section 22.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.
In auto-start mode, the IWDTCSTPR register setting are disabled, and the settings in option function select register 0 (OFS0) are enabled. The bit setting mode to the IWDTCSTPR register can also be made in the OFS0 register. For details, refer to section 22.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

SLCSTP Bit (Sleep Mode Count Stop Control)

This bit selects whether to stop counting at a transition to sleep mode, software standby mode, or deep sleep mode.

22.2.6 Option Function Select Register 0 (OFS0)

For option function select register 0 (OFS0), refer to section 22.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

22.3 Operation

22.3.1 Count Operation in Each Start Mode

Select the IWDT start mode by setting the IWDTSTRT bit in option function select register 0 (OFS0).
When the OFS0.IWDTSTRT bit is 1 (register start mode), the IWDTCR, IWDTRCR, and IWDTCSTPR registers are enabled, and counting is started by refresh operation (writing) to the IWDTRR register. When the OFS0.IWDTSTRT bit is 0 (auto-start mode), the setting of the OFS0 register is enabled, and counting automatically starts after reset.

22.3.1.1 Register Start Mode

When the OFS0.IWDTSTRT bit in option function select register 0 is 1, register start mode is selected, and the IWDTCR, IWDTRCR, and IWDTCSTPR registers are enabled.
After a reset is released, set the clock divide ratio, window start and end positions, and timeout period in the IWDTCR register, the reset output or interrupt request output in the IWDTRCR register, and the counter stop control at transitions to low power consumption states in the IWDTCSTPR register. Then refresh the counter to start counting down from the value selected by setting the IWDTCR.TOPS[1:0] bits.
Thereafter, as long as the program continues normal operation and the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The IWDT does not output the reset signal as long as this continues. However, if the counter underflows because the counter cannot be refreshed due to a program runaway, or if a refresh error occurs because the counter was refreshed outside the refresh-permitted period, the IWDT outputs a reset signal or a non-maskable interrupt request (WUNI). Set the IWDTRCR.RSTIRQS bit to select either reset output or interrupt request output.

Figure 22.3 shows an example of operation under the following conditions.

- Register start mode (OFS0.IWDTSTRT = 1)
- Reset output is enabled (IWDTRCR.RSTIRQS = 1)
- The window start position is 75% (IWDTCR.RPSS[1:0] = 10b)
- The window end position is 25% (IWDTCR.RPES[1:0] = 10b)

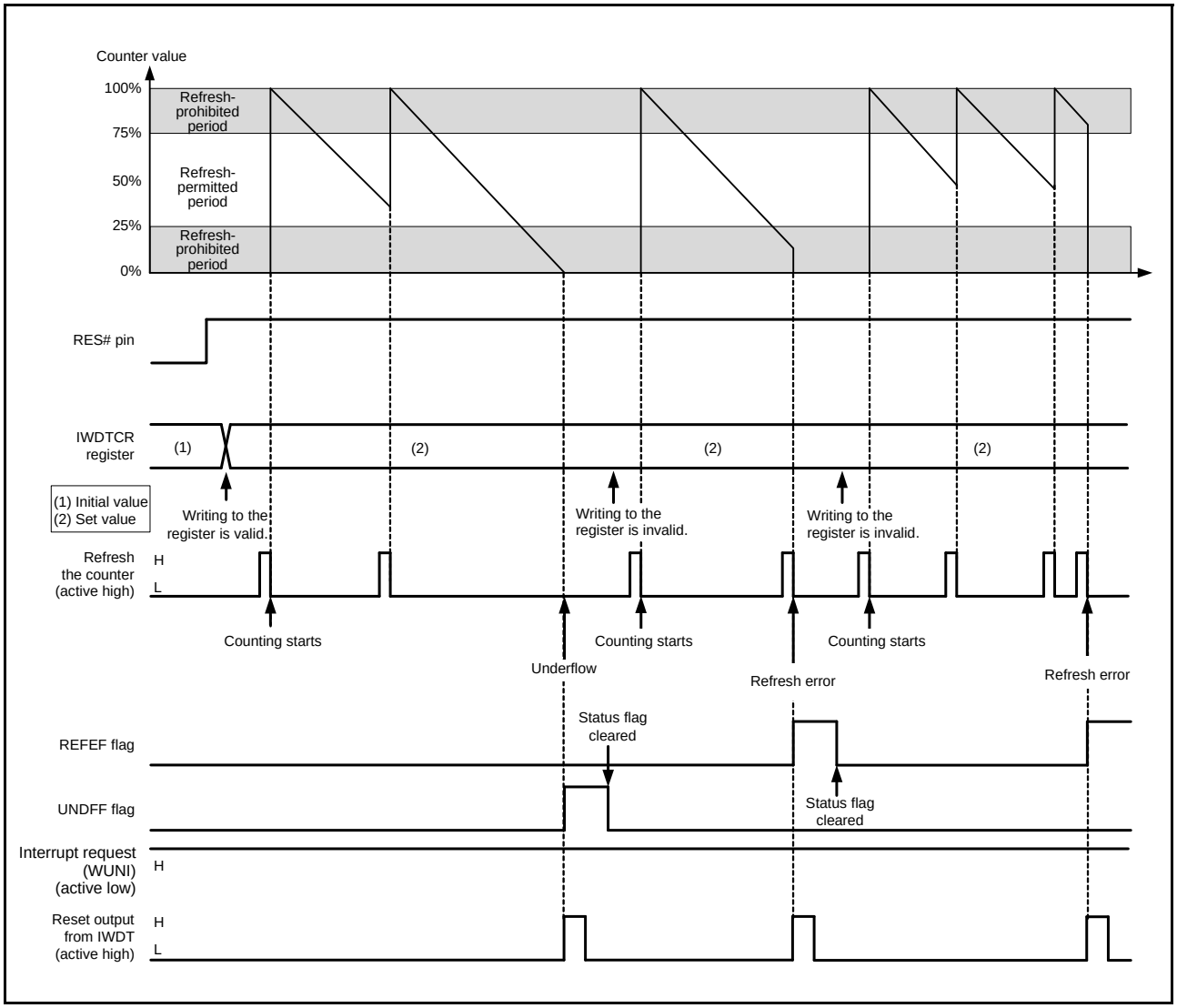


Figure 22.3 Operation Example in Register Start Mode

22.3.1.2 Auto-Start Mode

When the IWDTSTRT bit in option function select register 0 (OFS0) is 0, auto-start mode is selected, and the IWDTCR, IWDTRCR, and IWDTCSTPR registers are disabled.
Within the reset state, the clock divide ratio, window start and end positions, timeout period, reset output or interrupt request output, and counter stop control at transitions to low power consumption states are set using the values specified in the OFS0 register. When the reset is released, the counter automatically starts counting down from the value selected by the OFS0.IWDTTOPS[1:0] bits.
After that, as long as the program continues normal operation and the counter is refreshed in the refresh-permitted period, the value in the counter is re-set each time the counter is refreshed and counting down continues. The IWDT does not output the reset signal as long as this continues. However, if the counter underflows because refreshing of the counter is not possible due to the program having entered crashed execution or if a refresh error occurs due to refreshing outside the refresh-permitted period, the IWDT outputs the reset signal or non-maskable interrupt request (WUNI). After the reset signal or non-maskable interrupt request (WUNI) is generated, the counter reloads the timeout period after counting for one cycle, and restarts counting. Set the OFS0.IWDTRSTIRQS bit to select either reset output or interrupt request output.

Figure 22.4 shows an example of operation under the following conditions.

- Auto-start mode (OFS0.IWDTSTRT = 0)
- Non-maskable interrupt request output is enabled (OFS0.IWDTRSTIRQS = 0)
- The window start position is 75% (OFS0.IWDTRPSS[1:0] = 10b)
- The window end position is 25% (OFS0.IWDTRPES[1:0] = 10b)

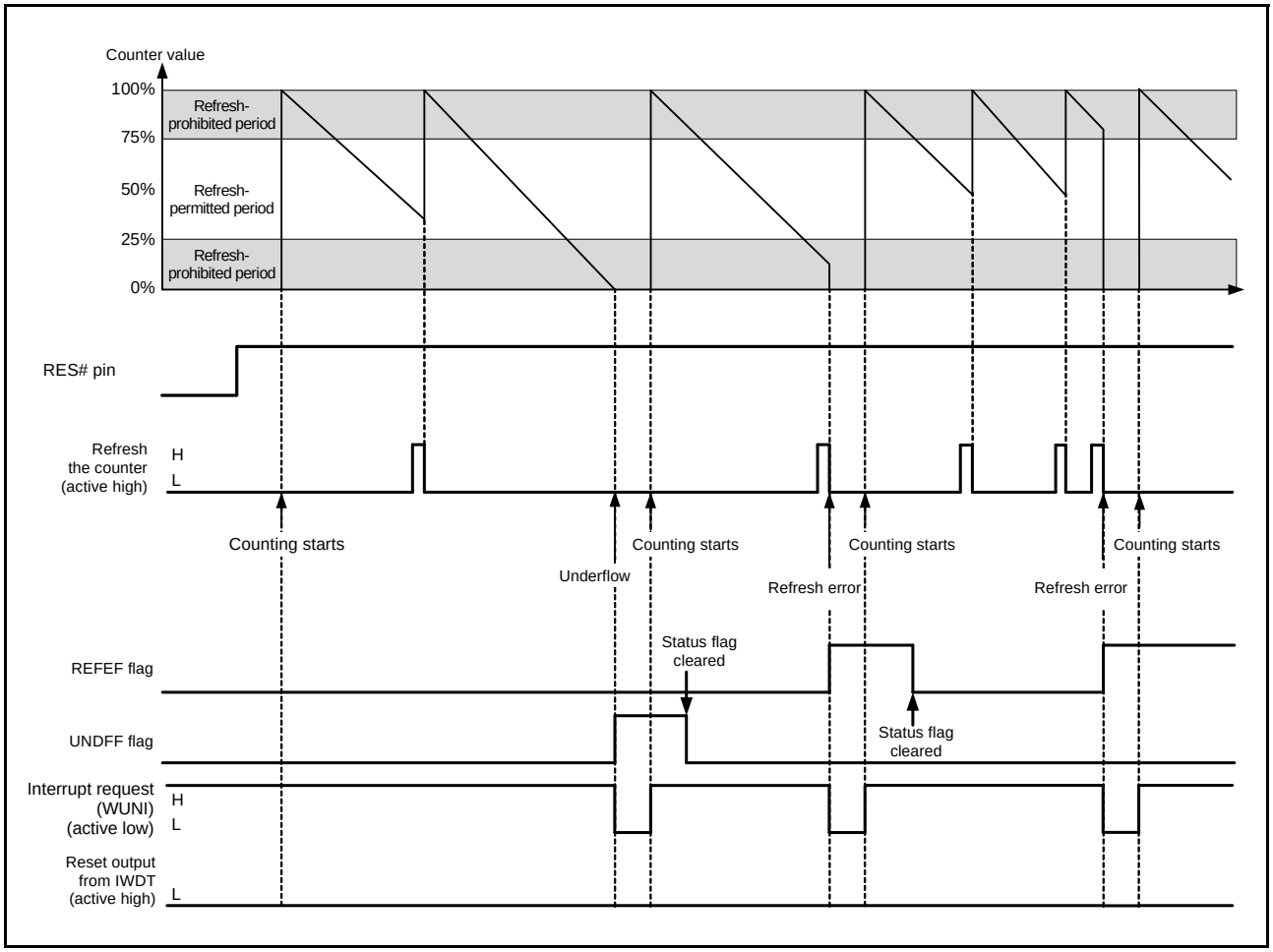


Figure 22.4 Operation Example in Auto-Start Mode

22.3.2 Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers

Writing to the IWDTCR, IWDTRCR, or IWDTCSTPR register is only possible once between the release from the reset state and the first refresh operation.

After a refresh operation (counting starts) or the IWDTCR, IWDTRCR, or IWDTCSTPR register is written to, the protection signal in the IWDT becomes 1 to protect registers IWDTCR, IWDTRCR, and IWDTCSTPR against subsequent attempts at writing.

This protection is released by the reset source of the IWDT. With other reset sources, the protection is not released.

Figure 22.5 shows control waveforms produced in response to writing to the IWDTCR register.

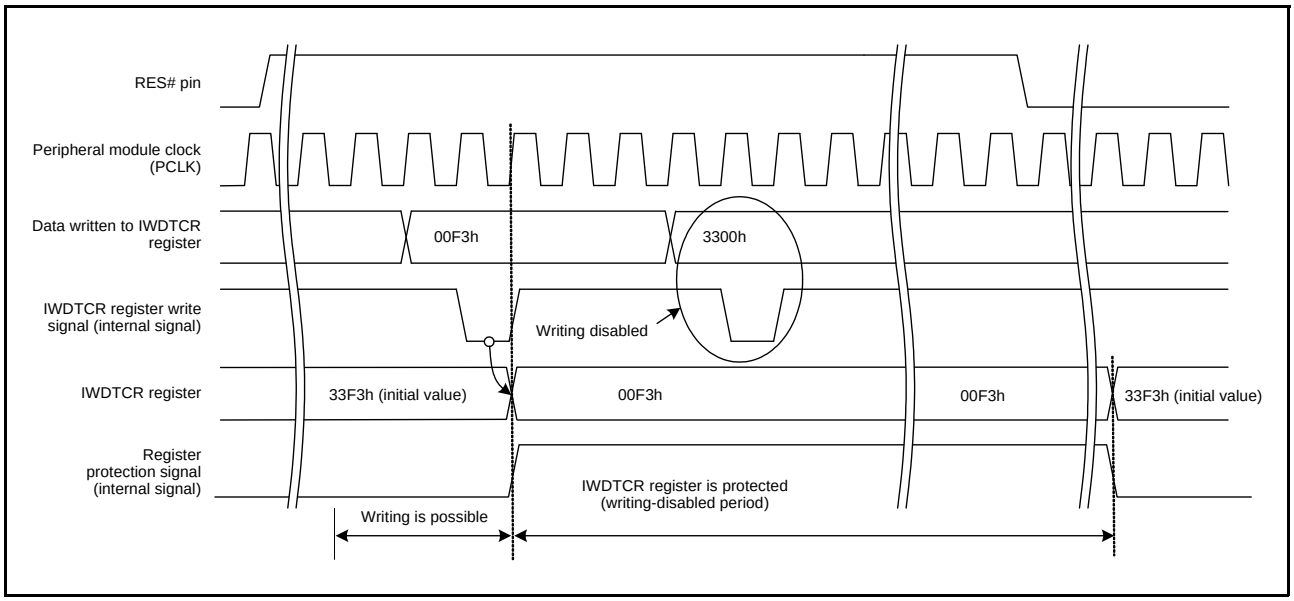


Figure 22.5 Control Waveforms Produced in Response to Writing to the IWDTCR Register

22.3.3 Refresh Operation

The counter is refreshed and starts operation (counting is started by refreshing) by writing the values 00h and then FFh to the IWDTRR register. If a value other than FFh is written after 00h, the counter is not refreshed. After such invalid writing, correct refreshing is performed by again writing 00h and then FFh to the IWDTRR register.
When writing is done in the order of 00h (first time) → 00h (second time), and if FFh is written after that, the writing order 00h → FFh is satisfied; writing 00h (n–1-th time) → 00h (nth time) → FFh is valid and correct refreshing will be done. Even when the first value written before 00h is not 00h, correct refreshing will be done if the operation contains the set of writing 00h → FFh. Moreover, even if a register other than the IWDTRR register is accessed or the IWDTRR register is read between writing 00h and writing FFh to the IWDTRR register, correct refreshing will be done.

[Sample sequences of writing that are valid for refreshing the counter]

- 00h → FFh
- 00h (n–1-th time) → 00h (nth time) → FFh
- 00h → access to another register or read from the IWDTRR register → FFh

[Sample sequences of writing that are not valid for refreshing the counter]

- 23h (a value other than 00h) → FFh
- 00h → 54h (a value other than FFh)
- 00h → AAh (00h and a value other than FFh) → FFh

Even when 00h is written to the IWDTRR register outside the refresh-permitted period, if FFh is written to the IWDTRR register in the refresh-permitted period, the writing sequence is valid and refreshing will be done.
After FFh is written to the IWDTRR register, refreshing the counter requires up to four cycles of the signal for counting (the IWDTCR.CKS[3:0] bits determine how many cycles of the IWDT-dedicated clock (IWDTCLK) make up one cycle for counting). Therefore, writing FFh to the IWDTRR register should be completed four-count cycles before the end position of the refresh-permitted period or a counter underflow. The value of the counter can be checked by the IWDTSR.CNTVAL[13:0] bits.

[Sample refreshing timings]

- When the window start position is set to 03FFh, even if 00h is written to the IWDTRR register before 03FFh is reached (0402h, for example), refreshing is done if FFh is written to the IWDTRR register after the value of the IWDTSR.CNTVAL[13:0] bits has reached 03FFh.
- When the window end position is set to 03FFh, refreshing is done if 0403h (four-count cycles before 03FFh) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to the IWDTRR register.
- When the refresh-permitted period continues until count 0000h, refreshing can be done immediately before an underflow. In this case, if 0003h (four-count cycles before an underflow) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to the IWDTRR register, no underflow occurs and refreshing is done.

Figure 22.6 shows the IWDT refresh-operation waveforms when PCLK > IWDTCLK and clock divide ratio = IWDTCLK.

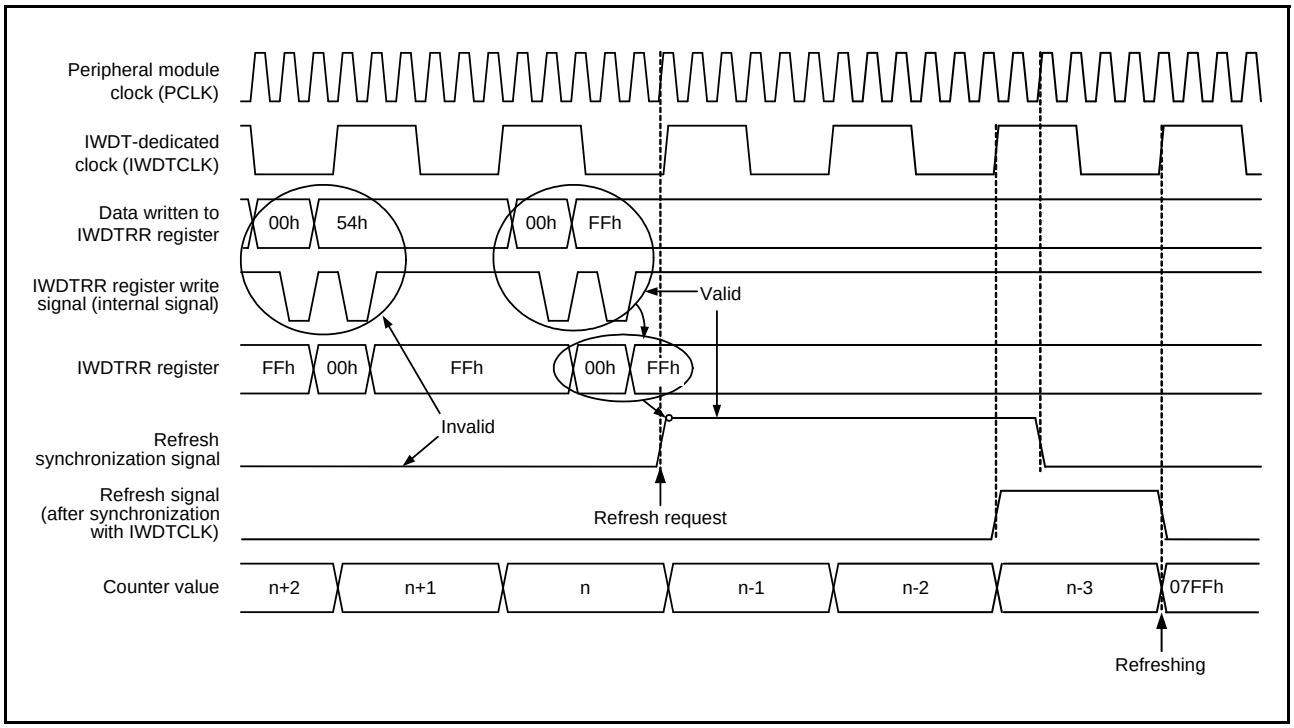


Figure 22.6 IWDT Refresh Operation Waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

22.3.4 Status Flags

The IWDTSR.REFEF and IWDTSR.UNDFF flags retain the source of the reset signal output from the IWDT or the source of the interrupt request from the IWDT.
Thus, after release from the reset state or interrupt request generation, read the IWDTSR.REFEF and IWDTSR.UNDFF flags to check for the reset or interrupt source.
For each flag, writing 0 clears the bit and writing 1 has no effect.
Leaving the status flags unchanged does not affect operation. If the flags are not cleared, at the time of the next reset or interrupt request from the IWDT, the earlier reset or interrupt source is cleared and the new reset or interrupt source is written.
After 0 is written to each flag, up to three IWDTCLK cycles and two PCLK cycles are required before the value is reflected.

22.3.5 Reset Output

When the IWDTRCR.RSTIRQS bit is set to 1 in register start mode or when the IWDTRSTIRQS bit in option function select register 0 (OFS0) is set to 1 in auto-start mode, a reset signal is output when an underflow in the counter or a refresh error occurs.
In register start mode, the counter is initialized (0000h) and kept in that state after assertion of the reset signal. After the reset is released and the program is restarted, the counter is set up again and counting down is started by refreshing.
In auto-start mode, counting down automatically starts after the reset output.

22.3.6 Interrupt Sources

When the IWDTRCR.RSTIRQS bit is set to 0 in register start mode or when the OFS0.IWDTRSTIRQS bit is set to 0 in auto-start mode, an interrupt (WUNI) signal is output when an underflow in the counter or a refresh error occurs. This interrupt can be used as a non-maskable interrupt. For details, refer to section 14, Interrupt Controller (ICUb).

Table 22.4 IWDT Interrupt Source

Name	Interrupt Source	DTC Activation
WUNI	Counter underflow Refresh error	Not possible

22.3.7 Reading the Counter Value

As the counter in IWDT-dedicated clock (IWDTCLK), the counter value cannot be read directly. The IWDT synchronizes the counter value with the peripheral module clock (PCLK) and stores it in the IWDTSR.CNTVAL[13:0] bits. Thus, the counter value can be checked indirectly through the IWDTSR.CNTVAL[13:0] bits.
Reading the counter value requires multiple PCLK clock cycles (up to four clock cycles), and the read counter value may differ from the actual counter value by a value of one count.
Figure 22.7 shows the processing for reading the IWDT counter value when PCLK > IWDTCLK and clock divide ratio = IWDTCLK.

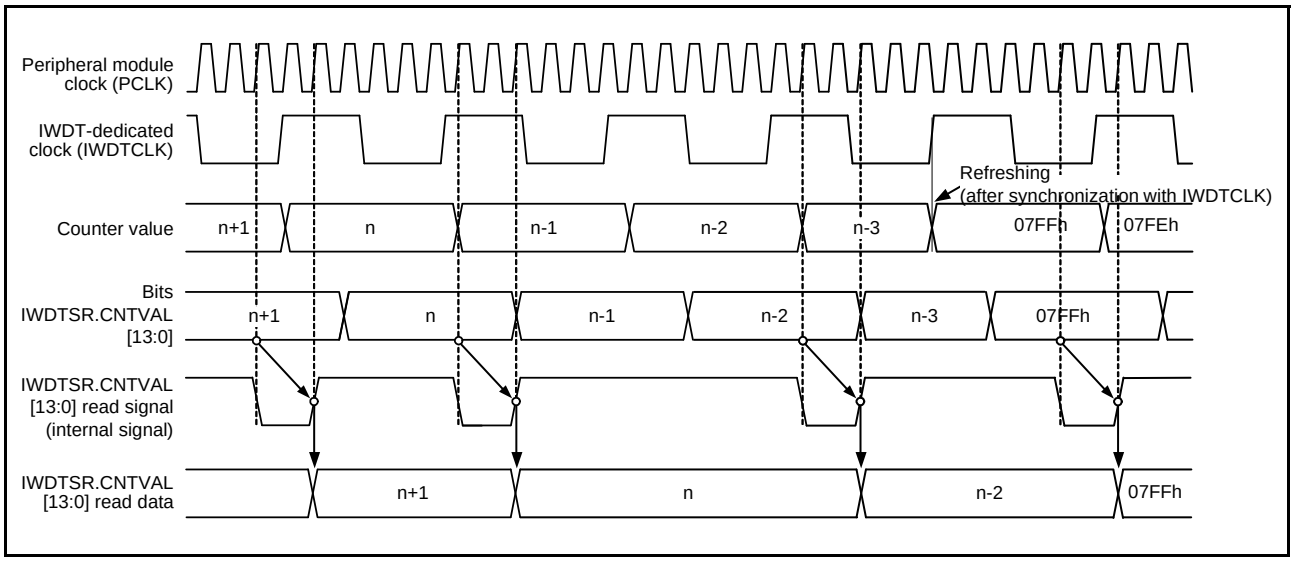


Figure 22.7 Processing for Reading IWDT Counter Value (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)

22.3.8 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Table 22.5 lists the correspondence between option function select register 0 (OFS0) used in auto-start mode and the registers used in register start mode.
Do not change the OFS0 register setting during IWDT operation.
For details on the OFS0 register, refer to section 7.2.1, Option Function Select Register 0 (OFS0).

Table 22.5 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Target of Control	Function	OFS0 Register (Enabled in Auto-Start Mode) OFS0.IWDTSTRT = 0	IWDT Registers (Enabled in Register Start Mode) OFS0.IWDTSTRT = 1
Counter	Timeout period selection	OFS0.IWDTTOPS[1:0]	IWDTCR.TOPS[1:0]
	Clock frequency divide ratio selection	OFS0.IWDTCKS[3:0]	IWDTCR.CKS[3:0]
	Window start position selection	OFS0.IWDTRPSS[1:0]	IWDTCR.RPSS[1:0]
	Window end position selection	OFS0.IWDTRPES[1:0]	IWDTCR.RPES[1:0]
Reset output or interrupt request output	Reset output or interrupt request output selection	OFS0.IWDTRSTIRQS	IWDTRCR.RSTIRQS
Count stop	Sleep mode count stop control	OFS0.IWDTSLCSTP	IWDTCSTPR.SLCSTP

22.4 Usage Notes

22.4.1 Refresh Operations

When making the settings to control the timing of refreshing, consider variations in the range of errors due to the accuracy of the PCLK and IWDTCLK and set values which ensure that refreshing is possible.

22.4.2 Clock Divide Ratio Setting

Satisfy the frequency of the peripheral module clock (PCLK) ≥ 4 × (the frequency of the count source after divide).

23. Serial Communications Interface (SCIg, SCIh)

This MCU has three independent serial communications interface (SCI) channels. The SCI consists of the SCIg module (SCI1 and SCI5) and the SCIh module (SCI12).
The SCIg module (SCI1 and SCI5) can handle both asynchronous and clock synchronous serial communications. Asynchronous serial data communications can be carried out with standard asynchronous communications chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communications Interface Adapter (ACIA). As an extended function in asynchronous communications mode, the SCI also supports smart card (IC card) interfaces conforming to ISO/IEC 7816-3 (standard for Identification Cards). The SCI is also supports simple SPI interfaces, and simple I^{2}C-bus interfaces when configured for single-master systems.
The SCIh module includes the functions of the SCIg module, and supports an extended serial communication protocol formed of Start Frames and Information Frames.
In this section, “PCLK” is used to refer to PCLKB.

23.1 Overview

Table 23.1 lists the specifications of the SCIg module, Table 23.2 lists the specifications of the SCIh module, and Table 23.3 lists the specifications of the individual SCI channels.
Figure 23.1 shows the block diagram of SCI1 and SCI5 and Figure 23.2 shows the block diagram of SCI12 (SCIh).

Table 23.1 SCIg Specifications (1/2)

Item		Description
Serial communication modes		• Asynchronous • Clock synchronous • Smart card interface • Simple I^{2}C-bus • Simple SPI bus
Transfer speed		Bit rate specifiable with the on-chip baud rate generator.
Full-duplex communications		Transmitter: Continuous transmission possible using double-buffer structure. Receiver: Continuous reception possible using double-buffer structure.
I/O pins		Refer to Table 23.4 to Table 23.6.
Data transfer		Selectable as LSB first or MSB first transfer*1
Interrupt sources		Transmit end, transmit data empty, receive data full, and receive error Completion of generation of a start condition, restart condition, or stop condition (for simple I^{2}C mode)
Low power consumption function		Module stop state can be set for each channel.
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Start-bit detection	Low level or falling edge is selectable.
	Break detection	When a framing error occurs, a break can be detected by reading the RXDn pin level directly.
	Clock source	An internal or external clock can be selected. Transfer rate clock input from the MTU can be used. (SCI1 and SCI5)
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.

Table 23.1 SCIg Specifications (2/2)

Item		Description
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
Smart card interface mode	Error processing	An error signal can be automatically transmitted when detecting a parity error during reception
		Data can be automatically retransmitted when receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.
Simple I^2C mode	Transfer format	I^2C-bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Fast mode is supported (refer to section 23.2.11, Bit Rate Register (BRR) to set the transfer rate).
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun error
	SS input pin function	Applying the high level to the SSn# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable.
Bit rate modulation function		Correction of outputs from the on-chip baud rate generator can reduce errors.

Note 1. In simple I^2C mode, only MSB first is available.

Table 23.2 SCIh Specifications (1/2)

Item	Description
Serial communication modes	• Asynchronous • Clock synchronous • Smart card interface • Simple I^2C-bus • Simple SPI bus
Transfer speed	Bit rate specifiable with the on-chip baud rate generator.
Full-duplex communications	Transmitter: Continuous transmission possible using double-buffer structure. Receiver: Continuous reception possible using double-buffer structure.
I/O pins	Refer to Table 23.4 to Table 23.7.
Data transfer	Selectable as LSB first or MSB first transfer*1
Interrupt sources	Transmit end, transmit data empty, receive data full, and receive error Completion of generation of a start condition, restart condition, or stop condition (for simple I^2C mode)
Low power consumption function	Module stop state can be set.

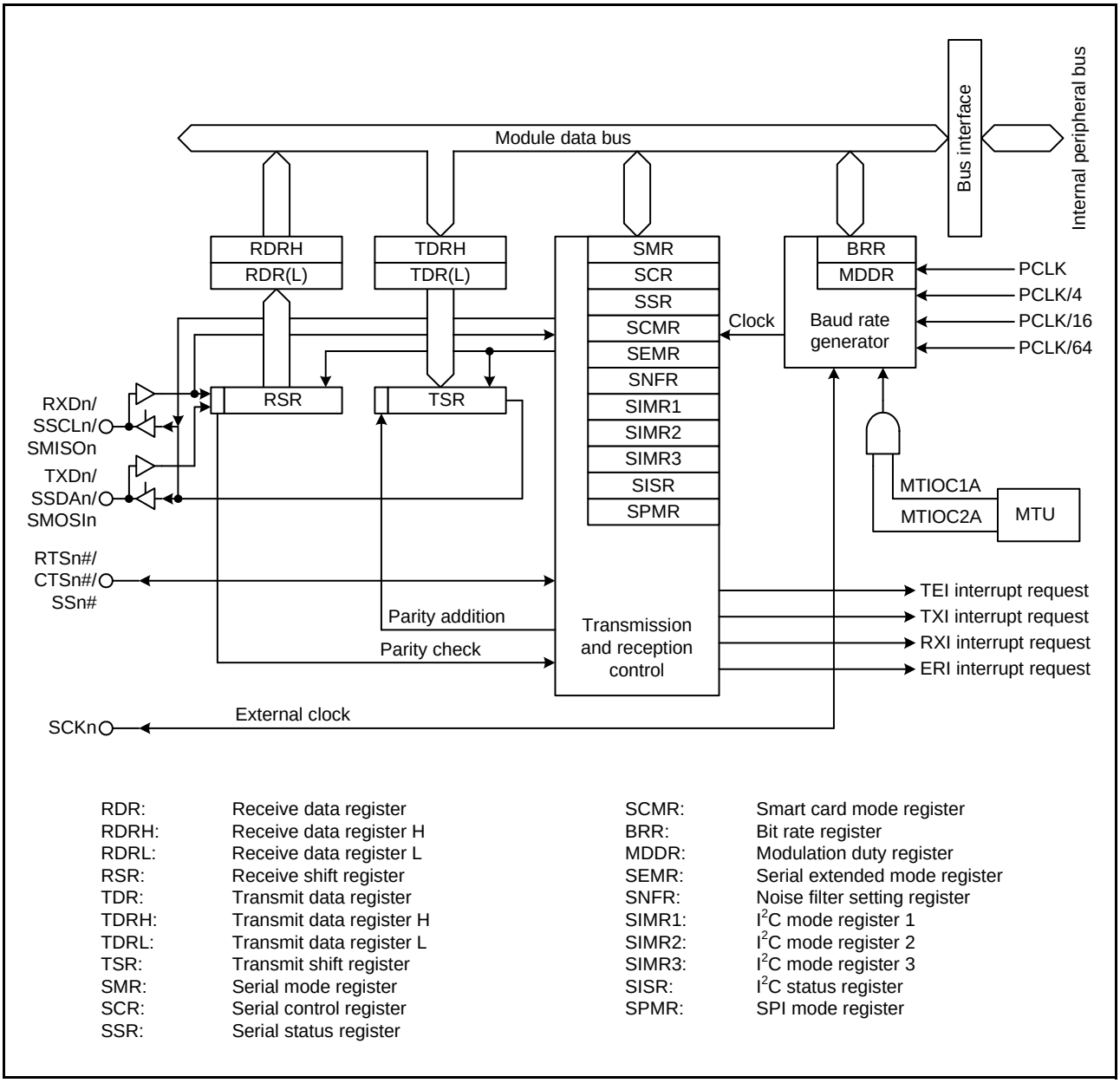
Table 23.2 SCIh Specifications (2/2)

Item		Description
Asynchronous mode	Data length	7, 8, or 9 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even parity, odd parity, or no parity
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
	Start-bit detection	Low level or falling edge is selectable.
	Break detection	When a framing error occurs, a break can be detected by reading the RXDn pin level directly.
	Clock source	An internal or external clock can be selected. Transfer rate clock input from the MTU can be used.
	Double-speed mode	Baud rate generator double-speed mode is selectable.
	Multi-processor communications function	Serial communication among multiple processors
	Noise cancellation	The signal paths from input on the RXDn pins incorporate digital noise filters.
Clock synchronous mode	Data length	8 bits
	Receive error detection	Overrun error
	Hardware flow control	CTSn# and RTSn# pins can be used in controlling transmission/reception.
Smart card interface mode	Error processing	An error signal can be automatically transmitted when detecting a parity error during reception
		Data can be automatically retransmitted when receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.
Simple I^2C mode	Transfer format	I^2C-bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Fast mode is supported (refer to section 23.2.11, Bit Rate Register (BRR) to set the transfer rate).
	Noise cancellation	The signal paths from input on the SSCLn and SSDAn pins incorporate digital noise filters, and the interval for noise cancellation is adjustable.
Simple SPI bus	Data length	8 bits
	Detection of errors	Overrun error
	SS input pin function	Applying the high level to the SSn# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock polarity are selectable.
Extended serial mode	Start Frame transmission	• Output of a low level as the Break Field over a specified width and generation of interrupts on completion • Detection of bus collisions and the generation of interrupts on detection
	Start Frame reception	• Detection of the Break Field low width and generation of an interrupt on detection • Comparison of Control Fields 0 and 1 and generation of an interrupt when the two match • Two kinds of data for comparison (primary and secondary) can be set in Control Field 1. • A priority interrupt bit can be set in Control Field 1. • Handling of Start Frames that do not include a Break Field • Handling of Start Frames that do not include a Control Field 0 • Function for measuring bit rates
	I/O control function	• Selectable polarity for TXDX12 and RXDX12 signals • Selection of a digital filter for the RXDX12 signal • Half-duplex operation employing RXDX12 and TXDX12 signals multiplexed on the same pin • Selectable timing for the sampling of data received through RXDX12
	Timer function	• Usable as a reload timer
Bit rate modulation function		Correction of outputs from the on-chip baud rate generator can reduce errors.

Note 1. In simple I^2C mode, only MSB first is available.

Table 23.3 Functions of SCI Channels

Item	SCI1	SCI5	SCI12
Asynchronous mode	Available	Available	Available
Clock synchronous mode	Available	Available	Available
Smart card interface mode	Available	Available	Available
Simple I²C mode	Available	Available	Available
Simple SPI mode	Available	Available	Available
Extended serial mode	Not available	Not available	Available
MTU clock input	Available	Available	Available



RDR:	Receive data register	SCMR:	Smart card mode register
RDRH:	Receive data register H	BRR:	Bit rate register
RDRL:	Receive data register L	MDDR:	Modulation duty register
RSR:	Receive shift register	SEMR:	Serial extended mode register
TDR:	Transmit data register	SNFR:	Noise filter setting register
TDRH:	Transmit data register H	SIMR1:	I²C mode register 1
TDRL:	Transmit data register L	SIMR2:	I²C mode register 2
TSR:	Transmit shift register	SIMR3:	I²C mode register 3
SMR:	Serial mode register	SISR:	I²C status register
SCR:	Serial control register	SPMR:	SPI mode register
SSR:	Serial status register		

Figure 23.1 Block Diagram of SCIg (SCI1 and SCI5)

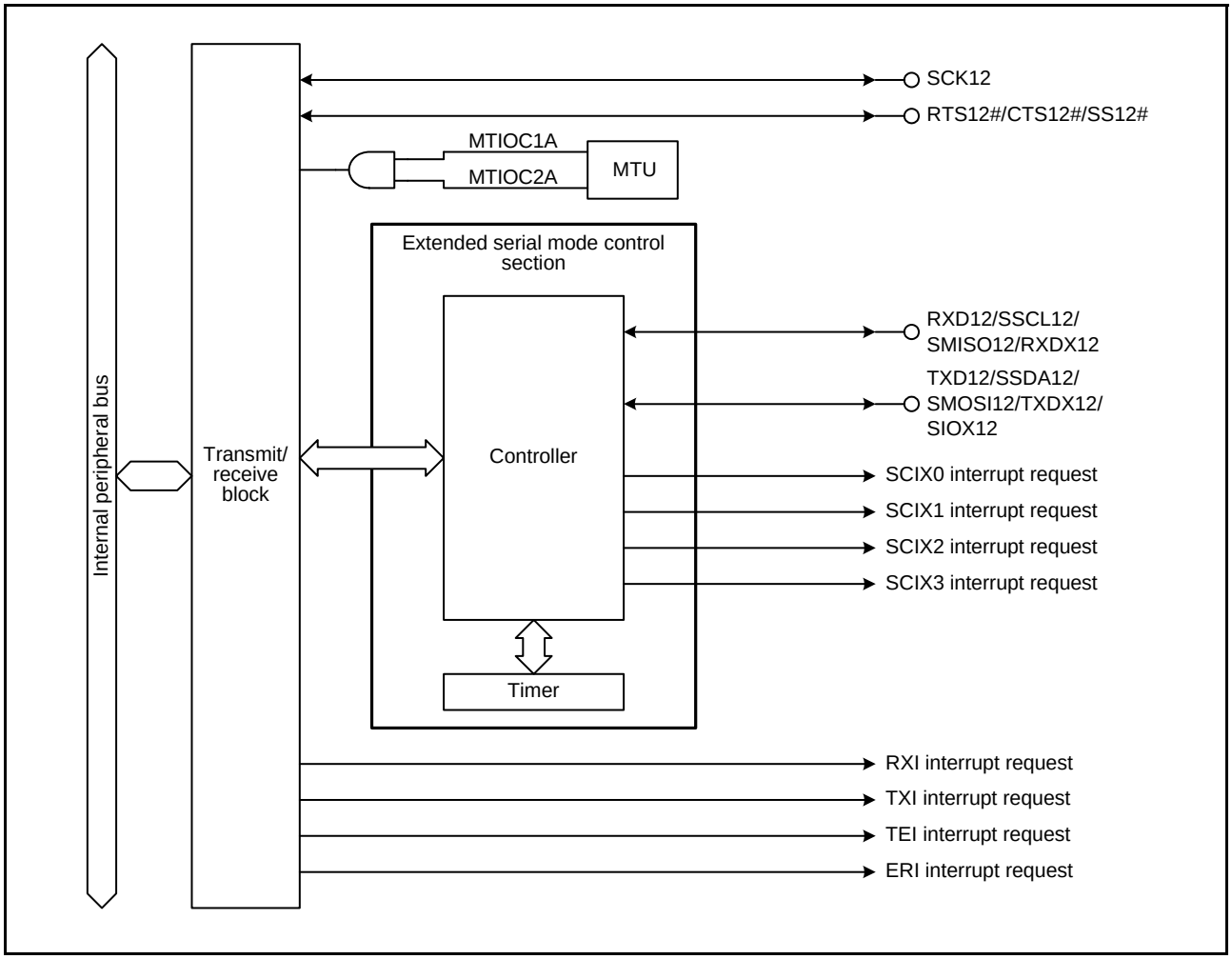


Figure 23.2 Block Diagram of SCIh (SCI12)

Table 23.4 to Table 23.7 list the pin configuration of the SCIs for the individual modes.

Table 23.4 SCI Pin Configuration in Asynchronous Mode and Clock Synchronous Mode

Channel	Pin Name	I/O	Function
SCI1	SCK1	I/O	SCI1 clock input/output
	RXD1	Input	SCI1 receive data input
	TXD1	Output	SCI1 transmit data output
	CTS1#/RTS1#	I/O	SCI1 transfer start control input/output
SCI5	SCK5	I/O	SCI5 clock input/output
	RXD5	Input	SCI5 receive data input
	TXD5	Output	SCI5 transmit data output
	CTS5#/RTS5#	I/O	SCI5 transfer start control input/output
SCI12	SCK12	I/O	SCI12 clock input/output
	RXD12	Input	SCI12 receive data input
	TXD12	Output	SCI12 transmit data output
	CTS12#/RTS12#	I/O	SCI12 transfer start control input/output

Table 23.5 SCI Pin Configuration in Simple I^2C Mode

Channel	Pin Name	I/O	Function
SCI1	SSCL1	I/O	SCI1 I^2C clock input/output
	SSDA1	I/O	SCI1 I^2C data input/output
SCI5	SSCL5	I/O	SCI5 I^2C clock input/output
	SSDA5	I/O	SCI5 I^2C data input/output
SCI12	SSCL12	I/O	SCI12 I^2C clock input/output
	SSDA12	I/O	SCI12 I^2C data input/output

Table 23.6 SCI Pin Configuration in Simple SPI Mode

Channel	Pin Name	I/O	Function
SCI1	SCK1	I/O	SCI1 clock input/output
	SMISO1	I/O	SCI1 slave transmit data input/output
	SMOSI1	I/O	SCI1 master transmit data input/output
	SS1#	Input	SCI1 chip select input
SCI5	SCK5	I/O	SCI5 clock input/output
	SMISO5	I/O	SCI5 slave transmit data input/output
	SMOSI5	I/O	SCI5 master transmit data input/output
	SS5#	Input	SCI5 chip select input
SCI12	SCK12	I/O	SCI12 clock input/output
	SMISO12	I/O	SCI12 slave transmit data input/output
	SMOSI12	I/O	SCI12 master transmit data input/output
	SS12#	Input	SCI12 chip select input

Table 23.7 SCI Pin Configuration in Extended Serial Mode

Channel	Pin Name	I/O	Function
SCI12	RXDX12	Input	SCI12 receive data input
	TXDX12	Output	SCI12 transmit data output
	SIOX12	I/O	SCI12 transfer data input/output

23.2 Register Descriptions

23.2.1 Receive Shift Register (RSR)

The RSR register 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.

23.2.2 Receive Data Register (RDR)

Address(es): SCI1.RDR 0008 A025h, SCI5.RDR 0008 A0A5h, SCI12.RDR 0008 B305h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

The RDR register is an 8-bit register that stores receive data.
When one frame of serial data has been received, the received serial data is transferred from the RSR register to the RDR register. Then the RSR register can receive the next data.
Since the RSR and RDR registers function as a double buffer in this way, continuous receive operations can be performed.
Read the RDR register only once after a receive data full interrupt (RXI) has occurred. Note that if next one frame of data is received before reading receive data from the RDR register, an overrun error occurs.
The RDR register cannot be written to by the CPU.

23.2.3 Receive Data Register H, L, HL (RDRH, RDRL, RDRHL)

- Receive Data Register H (RDRH)

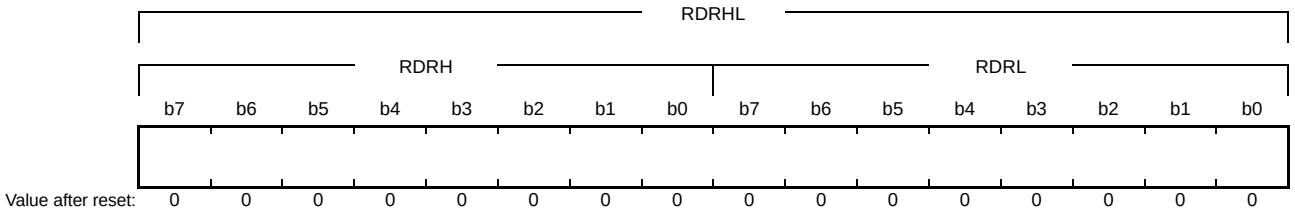
Address(es): SCI1.RDRH 0008 A030h, SCI5.RDRH 0008 A0B0h, SCI12.RDRH 0008 B310h

- Receive Data Register L (RDRL)

Address(es): SCI1.RDRL 0008 A031h, SCI5.RDRL 0008 A0B1h, SCI12.RDRL 0008 B311h

- Receive Data Register HL (RDRHL)

Address(es): SCI1.RDRHL 0008 A030h, SCI5.RDRHL 0008 A0B0h, SCI12.RDRHL 0008 B310h



The RDRH and RDRL registers are 8-bit registers that store receive data. Use these registers when asynchronous mode and 9-bit data length are selected.

The RDRL register is the shadow register of the RDR register; i.e. access to the RDRL register is equivalent to access to the RDR register.

After one frame of data is received, the received data is transferred from the RSR register to these registers, thus allowing the RSR register to receive the next data.

The RSR, RDRH and RDRL registers have a double-buffered construction to enable continuous reception.

Read the RDRH and RDRL registers should be performed only once in the order from the RDRH register to the RDRL register when a receive data full interrupt (RXI) request is issued. Note that an overrun error occurs when the next frame of data is received before the received data has been read from the RDRL register.

The CPU cannot write to the RDRH and RDRL registers. Bits 0 to 7 in the RDRH register are fixed to 0. These bits are read as 0.

The RDRHL register can be accessed in 16-bit units.

23.2.4 Transmit Data Register (TDR)

Address(es): SCI1.TDR 0008 A023h, SCI5.TDR 0008 A0A3h, SCI12.TDR 0008 B303h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

The TDR register is an 8-bit register that stores transmit data.
When the SCI detects that the TSR register is empty, it transfers the transmit data written in the TDR register to the TSR register and starts transmission.
The double-buffered structures of the TDR register and the TSR register enable continuous serial transmission. If the next transmit data has already been written to the TDR register when one frame of data is transmitted, the SCI transfers the written data to the TSR register to continue transmission.
The CPU is able to read from or write to the TDR register at any time. Only write transmit data to the TDR register once after each instance of the transmit data empty interrupt (TXI).

23.2.5 Transmit Data Register H, L, HL (TDRH, TDRL, TDRHL)

- Transmit Data Register H (TDRH)

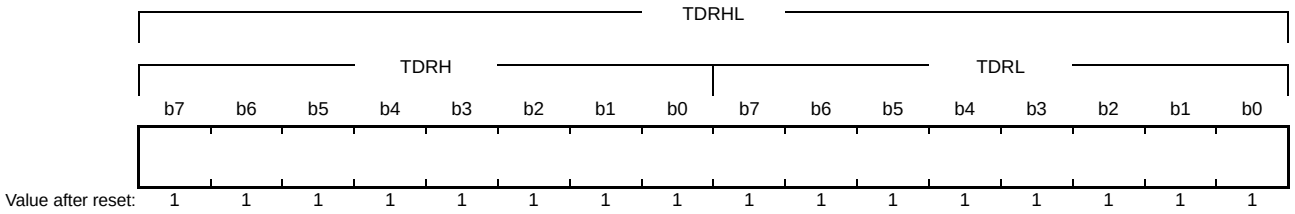
Address(es): SCI1.TDRH 0008 A02Eh, SCI5.TDRH 0008 A0AEh, SCI12.TDRH 0008 B30Eh

- Transmit Data Register L (TDRL)

Address(es): SCI1.TDRL 0008 A02Fh, SCI5.TDRL 0008 A0AFh, SCI12.TDRL 0008 B30Fh

- Transmit Data Register HL (TDRHL)

Address(es): SCI1.TDRHL 0008 A02Eh, SCI5.TDRHL 0008 A0AEh, SCI12.TDRHL 0008 B30Eh



The TDRH and TDRL registers are 8-bit registers that store transmit data. Use these registers when asynchronous mode and 9-bit data length are selected.
The TDRL register is the shadow register of the TDR register; i.e. access to the TDRL register is equivalent to access to the TDR register.
When empty space is detected in the TSR register, the transmit data stored in the TDRH and TDRL registers is transferred to the TSR register; i.e., transmitting is started.
The TSR, TDRH and TDRL registers have a double-buffered construction to realize continuous reception. When the next data to be transmitted is stored in the TDRL register after one frame of data has been transmitted, the transmitting operation is continued by transfer to the TSR register.
The CPU can read and write to the TDRH and TDRL registers. Bits 0 to 7 in the RDRH register are fixed to 1. These bits are read as 1. The write value should be 1.
Writing transmit data to the TDRH and TDRL registers should be performed only once in the order from the TDRH register to the TDRL register when a transmit data empty interrupt (TXI) request is issued.
The TDRHL register can be accessed in 16-bit units.

23.2.6 Transmit Shift Register (TSR)

The TSR register is a shift register that transmits serial data.
To perform serial data transmission, the SCI first automatically transfers transmit data from the TDR register to the TSR register, and then sends the data to the TXDn pin.
The TSR register cannot be directly accessed by the CPU.

23.2.7 Serial Mode Register (SMR)

Some bits in the SMR register have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI1.SMR 0008 A020h, SCI5.SMR 0008 A0A0h, SCI12.SMR 0008 B300h

	b7	b6	b5	b4	b3	b2	b1	b0
	CM	CHR	PE	PM	STOP	MP	CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK (n = 0)*1 0 1: PCLK/4 (n = 1)*1 1 0: PCLK/16 (n = 2)*1 1 1: PCLK/64 (n = 3)*1	R/W*4
b2	MP	Multi-Processor Mode	(Valid only in asynchronous mode) 0: Multi-processor communications function is disabled 1: Multi-processor communications function is enabled	R/W*4
b3	STOP	Stop Bit Length	(Valid only in asynchronous mode) 0: 1 stop bit 1: 2 stop bits	R/W*4
b4	PM	Parity Mode	(Valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity	R/W*4
b5	PE	Parity Enable	(Valid only in asynchronous mode) • When transmitting 0: Parity bit addition is not performed 1: The parity bit is added • When receiving 0: Parity bit checking is not performed 1: The parity bit is checked	R/W*4
b6	CHR	Character Length	(Valid only in asynchronous mode*2) Selects in combination with the SCMR.CHR1 bit. CHR1 CHR 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*3	R/W*4
b7	CM	Communications Mode	0: Asynchronous mode or simple I2C mode 1: Clock synchronous mode or simple SPI mode	R/W*4

Note 1. n is the decimal notation of the value of n in the BRR register (refer to section 23.2.11, Bit Rate Register (BRR)).
Note 2. In other than asynchronous mode, this bit setting is invalid and a fixed data length of 8 bits is used.
Note 3. LSB first is fixed and the MSB (bit 7) in the TDR register is not transmitted in transmission.
Note 4. Writable only when the SCR.TE bit is 0 and the SCR.RE bit is 0 (both serial transmission and reception are disabled).

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relation between the settings of these bits and the baud rate, refer to section 23.2.11, Bit Rate Register (BRR).

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.

STOP 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.

PM Bit (Parity Mode)

Selects the parity mode (even or odd) for transmission and reception.
The setting of the PM bit is invalid in multi-processor mode.

PE Bit (Parity Enable)

When this bit is set to 1, the parity bit is added to transmit data, and the parity bit is checked in reception.
Irrespective of the setting of the PE bit, the parity bit is not added or checked in multi-processor format.

CHR Bit (Character Length)

Selects the data length for transmission and reception.
Selects in combination with the SCMR.CHR1 bit.
In other than asynchronous mode, a fixed data length of 8 bits is used.

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMCI1.SMR 0008 A020h, SMCI5.SMR 0008 A0A0h, SMCI12.SMR 0008 B300h

	b7	b6	b5	b4	b3	b2	b1	b0
	GM	BLK	PE	PM	BCP[1:0]		CKS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKS[1:0]	Clock Select	b1 b0 0 0: PCLK (n = 0)*1 0 1: PCLK/4 (n = 1)*1 1 0: PCLK/16 (n = 2)*1 1 1: PCLK/64 (n = 3)*1	R/W*2
b3, b2	BCP[1:0]	Base Clock Pulse	Selects the number of base clock cycles in combination with the SCMR.BCP2 bit. Table 23.8 lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits.	R/W*2
b4	PM	Parity Mode	(Valid only when the PE bit is 1) 0: Selects even parity 1: Selects odd parity	R/W*2
b5	PE	Parity Enable	When this bit is set to 1, a parity bit is added to transmit data, and the parity of received data is checked. Set this bit to 1 in smart card interface mode.	R/W*2
b6	BLK	Block Transfer Mode	0: Non-block transfer mode operation 1: Block transfer mode operation	R/W*2
b7	GM	GSM Mode	0: Non-GSM mode operation 1: GSM mode operation	R/W*2

Note 1. n is the decimal notation of the value of n in the BRR register (refer to section 23.2.11, Bit Rate Register (BRR)).
Note 2. Writable only when the SCR.TE bit is 0 and the SCR.RE bit is 0 (both serial transmission and reception are disabled).

CKS[1:0] Bits (Clock Select)

These bits select the clock source for the on-chip baud rate generator.

For the relationship between the settings of these bits and the baud rate, refer to section 23.2.11, Bit Rate Register (BRR).

BCP[1:0] Bits (Base Clock Pulse)

These bits select the number of base clock cycles in a 1-bit data transfer time in smart card interface mode.

Set these bits in combination with the SCMR.BCP2 bit.

For details, refer to section 23.6.4, Receive Data Sampling Timing and Reception Margin.

Table 23.8 Combinations of the SCMR.BCP2 Bit and SMR.BCP[1:0] Bits

SCMR.BCP2 Bit	SMR.BCP[1:0] Bits		Number of Base Clock Cycles for 1-Bit Transfer Period
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

Note 1. S is the value of S in the BRR register (refer to section 23.2.11, Bit Rate Register (BRR)).

PM Bit (Parity Mode)

Selects the parity mode for transmission and reception (even or odd).
For details on the usage of this bit in smart card interface mode, refer to section 23.6.2, Data Format (Except in Block Transfer Mode).

PE Bit (Parity Enable)

Set the PE bit to 1.
The parity bit is added to transmit data before transmission, and the parity bit is checked in reception.

BLK Bit (Block Transfer Mode)

Setting this bit to 1 allows block transfer mode operation.
For details, refer to section 23.6.3, Block Transfer Mode.

GM Bit (GSM Mode)

Setting this bit to 1 allows GSM mode operation.
In GSM mode, the SSR.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, refer to section 23.6.6, Serial Data Transmission (Except in Block Transfer Mode) and section 23.6.8, Clock Output Control.

23.2.8 Serial Control Register (SCR)

Some bits in the SCR register have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI1.SCR 0008 A022h, SCI5.SCR 0008 A0A2h, SCI12.SCR 0008 B302h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKE[1:0]	Clock Enable	(Asynchronous mode) b1 b0 0 0: On-chip baud rate generator The SCKn pin becomes high-impedance. 0 1: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin. 1 x: External clock or MTU clock • The clock with a frequency 16 times the bit rate should be input from the SCKn pin. Input a clock signal with a frequency eight times the bit rate when the SEMR.ABCS bit is 1. • The SCKn pin becomes high-impedance when the MTU clock is used. (Clock synchronous mode) b1 b0 0 x: Internal clock The SCKn pin functions as the clock output pin. 1 x: External clock The SCKn pin functions as the clock input pin.	R/W*1
b2	TEIE	Transmit End Interrupt Enable	0: A TEI interrupt request is disabled 1: A TEI interrupt request is enabled	R/W
b3	MPIE	Multi-Processor Interrupt Enable	(Valid in asynchronous mode when the SMR.MP bit is 1) 0: Normal reception 1: When the data with the multi-processor bit set to 0 is received, the data is not read, and setting the status flags RDRF, ORER, and FER in the SSR register to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and normal reception is resumed.	R/W
b4	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W*2
b5	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*2
b6	RIE	Receive Interrupt Enable	0: RXI and ERI interrupt requests are disabled 1: RXI and ERI interrupt requests are enabled	R/W
b7	TIE	Transmit Interrupt Enable	0: A TXI interrupt request is disabled 1: A TXI interrupt request is enabled	R/W

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. 1 can be written only when TE = 0 and RE = 0, while the SMR.CM bit is 1. After setting TE or RE to 1, only 0 can be written to TE and RE. While the SMR.CM bit is 0 and the SIMR1.IICM bit is 0, writing is enabled under any condition.

CKE[1:0] Bits (Clock Enable)

These bits select the clock source and SCKn pin function.
The combination of the settings of these bits and of the SEMR.ACS0 bit sets the internal MTU clock.

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 I^2C 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.

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 RDRF, ORER, and FER in the SSR register to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE bit is automatically cleared to 0, and normal reception is resumed. For details, refer to section 23.4, Multi-Processor Communications Function.
When the data with the multi-processor bit set to 0 is received, the receive data is not transferred from the RSR register to the RDR register, a receive error is not detected, and setting the flags RDRF, ORER, and FER to 1 is disabled.
When the data with the multi-processor bit set to 1 is received, the MPB bit is set to 1, the MPIE bit is automatically cleared to 0, the RXI and ERI interrupt requests are enabled (if the SCR.RIE bit is set to 1), and setting the flags RDRF, ORER, and FER to 1 is enabled.
Set the MPIE bit to 0 if multi-processor communications function is not to be used.

RE Bit (Receive Enable)

Enables or disables serial reception.
When this bit is set to 1, serial reception is started by detecting the start bit in asynchronous mode or the synchronous clock input in clock synchronous mode. Note that the SMR register should be set prior to setting the RE bit to 1 in order to designate the reception format.
Even if reception is halted by setting the RE bit to 0, the ORER, FER, PER, and RDRF flags in the SSR register are not affected and the previous value is retained.

TE Bit (Transmit Enable)

Enables or disables serial transmission.
When this bit is set to 1, serial transmission is started by writing transmit data to the TDR register. Note that the SMR register should be set prior to setting the TE bit to 1 in order to designate the transmission format.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.
An RXI interrupt request is disabled by setting the RIE bit to 0.
An ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in the SSR register and then setting the flag to 0, or setting the RIE bit to 0.

TIE Bit (Transmit Interrupt Enable)

Enables or disables TXI interrupt request.
A TXI interrupt request is disabled by setting the TIE bit to 0.

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMCI1.SCR 0008 A022h, SMCI5.SCR 0008 A0A2h, SMCI12.SCR 0008 B302h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CKE[1:0]	Clock Enable	• When SMR.GM = 0 b1 b0 0 0: Output disabled The SCKn pin becomes high-impedance. 0 1: Clock output 1 x: Setting prohibited • When SMR.GM = 1 b1 b0 0 0: Output fixed low x 1: Clock output 1 0: Output fixed high	R/W*1
b2	TEIE	Transmit End Interrupt Enable	This bit should be 0 in smart card interface mode.	R/W
b3	MPIE	Multi-Processor Interrupt Enable	This bit should be 0 in smart card interface mode.	R/W
b4	RE	Receive Enable	0: Serial reception is disabled 1: Serial reception is enabled	R/W*2
b5	TE	Transmit Enable	0: Serial transmission is disabled 1: Serial transmission is enabled	R/W*2
b6	RIE	Receive Interrupt Enable	0: RXI and ERI interrupt requests are disabled 1: RXI and ERI interrupt requests are enabled	R/W
b7	TIE	Transmit Interrupt Enable	0: A TXI interrupt request is disabled 1: A TXI interrupt request is enabled	R/W

x: Don't care

Note 1. Writable only when TE = 0 and RE = 0.

Note 2. 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.

For details on interrupt requests, refer to section 23.12, Interrupt Sources.

CKE[1:0] Bits (Clock Enable)

These bits control the clock output from the SCKn pin.

In GSM mode, clock output can be dynamically switched. For details, refer to section 23.6.8, Clock Output Control.

TEIE Bit (Transmit End Interrupt Enable)

This bit should be 0 in smart card interface mode.

MPIE Bit (Multi-Processor Interrupt Enable)

This bit should be 0 in smart card interface mode.

RE Bit (Receive Enable)

Enables or disables serial reception.

When this bit is set to 1, serial reception is started by detecting the start bit. Note that the SMR register should be set prior to setting the RE bit to 1 in order to designate the reception format.

Even if reception is halted by setting the RE bit to 0, the ORER, FER, and PER flags in the SSR register are not affected and the previous value is retained.

TE Bit (Transmit Enable)

Enables or disables serial transmission.

When this bit is set to 1, serial transmission is started by writing transmit data to the TDR register. Note that the SMR register should be set prior to setting the TE bit to 1 in order to designate the transmission format.

RIE Bit (Receive Interrupt Enable)

Enables or disables RXI and ERI interrupt requests.

An RXI interrupt request is disabled by setting the RIE bit to 0.

An ERI interrupt request can be canceled by reading 1 from the ORER, FER, or PER flag in the SSR register and then setting the flag to 0, or setting the RIE bit to 0.

TIE Bit (Transmit Interrupt Enable)

Enables or disables TXI interrupt request.

A TXI interrupt request is disabled by setting the TIE bit to 0.

23.2.9 Serial Status Register (SSR)

Some bits in the SSR register have different functions in smart card interface mode and non-smart card interface mode.

(1) Non-Smart Card Interface Mode (SCMR.SMIF = 0)

Address(es): SCI1.SSR 0008 A024h, SCI5.SSR 0008 A0A4h, SCI12.SSR 0008 B304h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	Sets the multi-processor bit for adding to the transmission frame 0: Data transmission cycles 1: ID transmission cycles	R/W
b1	MPB	Multi-Processor	Value of the multi-processor bit in the reception frame 0: Data transmission cycles 1: ID transmission cycles	R
b2	TEND	Transmit End Flag	0: A character is being transmitted. 1: Character transfer has been completed.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R/(W) *1
b4	FER	Framing Error Flag	0: No framing error occurred 1: A framing error has occurred	R/(W) *1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R/(W) *1
b6	RDRF	Receive Data Full Flag	0: No valid data is held in the RDR register 1: Received data is held in the RDR register	R/(W) *2
b7	TDRE	Transmit Data Empty Flag	0: Data to be transmitted is held in the TDR register 1: No data is held in the TDR register	R/(W) *2

Note 1. Only 0 can be written to this bit, to clear the flag. To clear this flag, confirm that the flag is 1 and then set it to 0.
Note 2. Write 1 when writing is necessary.

MPB Bit (Multi-Processor)

Holds the value of the multi-processor bit in the reception frame. This bit does not change when the SCR.RE bit is 0.

TEND Flag (Transmit End Flag)

Indicates completion of transmission.

[Setting conditions]

- When the SCR.TE bit is set to 0 (serial transmission is disabled)
  When the SCR.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 the time of transmission of the tail-end bit of a character being transmitted

[Clearing condition]

- When transmit data are written to the TDR register while the SCR.TE bit is 1
  When setting the TEND flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

PER Flag (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception
  Although receive data when the parity error occurs is transferred to the RDR register, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to the PER flag after reading PER = 1
  When setting the PER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

FER Flag (Framing Error Flag)

Indicates that a framing error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When the stop bit is 0
  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 the RDR register, no RXI interrupt request occurs. In addition, when the FER flag is being set to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to the FER flag after reading FER = 1
  When setting the FER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the SCR.RE bit is set to 0, the FER flag is not affected and retains its previous value.

ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from the RDR register
  In the RDR register, 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, subsequent serial reception cannot be performed. Note that, in clock synchronous mode, serial transmission also cannot continue.

[Clearing condition]

- When 0 is written to the ORER flag after reading ORER = 1
  When setting the ORER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.

RDRF Flag (Receive Data Full Flag)

Indicates whether the RDR register has received data.

[Setting condition]

- When data has been received normally, and transferred from the RSR register to the RDR register

[Clearing condition]

- When data is read from the RDR register

TDRE Flag (Transmit Data Empty Flag)

Indicates whether the TDR register has data to be transmitted.

[Setting condition]

- When data is transferred from the TDR register to the TSR register

[Clearing condition]

- When data is written to the TDR register

(2) Smart Card Interface Mode (SCMR.SMIF = 1)

Address(es): SMCI1.SSR 0008 A024h, SMCI5.SSR 0008 A0A4h, SMCI12.SSR 0008 B304h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
Value after reset:	1	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MPBT	Multi-Processor Bit Transfer	This bit should be set to 0 in smart card interface mode.	R/W
b1	MPB	Multi-Processor	This bit is not used in smart card interface mode. It should be set to 0.	R
b2	TEND	Transmit End Flag	0: A character is being transmitted. 1: Character transfer has been completed.	R
b3	PER	Parity Error Flag	0: No parity error occurred 1: A parity error has occurred	R/(W) *1
b4	ERS	Error Signal Status Flag	0: Low error signal not responded 1: Low error signal responded	R/(W) *1
b5	ORER	Overrun Error Flag	0: No overrun error occurred 1: An overrun error has occurred	R/(W) *1
b6	RDRF	Receive Data Full Flag	0: No valid data is held in the RDR register 1: Received data is held in the RDR register	R/(W) *2
b7	TDRE	Transmit Data Empty Flag	0: Data to be transmitted is held in the TDR register 1: No data is held in the TDR register	R/(W) *2

Note 1. Only 0 can be written to this bit, to clear the flag. To clear this flag, confirm that the flag is 1 and then set it to 0.
Note 2. Write 1 when writing is necessary.

TEND Flag (Transmit End Flag)

With no error signal from the receiving side, this bit is set to 1 when further data for transfer is ready to be transferred to the TDR register.

[Setting conditions]

- When the SCR.TE bit is 0 (serial transmission is disabled)
  When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- When a specified period has elapsed after the latest transmission of 1 byte, the ERS flag is 0, and the TDR register is not updated
  The set timing is determined by register settings as listed below.
  When SMR.GM = 0 and SMR.BLK = 0, 12.5 etu after the start of transmission
  When SMR.GM = 0 and SMR.BLK = 1, 11.5 etu after the start of transmission
  When SMR.GM = 1 and SMR.BLK = 0, 11.0 etu after the start of transmission
  When SMR.GM = 1 and SMR.BLK = 1, 11.0 etu after the start of transmission

[Clearing condition]

- When transmit data are written to the TDR register while the SCR.TE bit is 1
  When setting the TEND flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.

PER Flag (Parity Error Flag)

Indicates that a parity error has occurred during reception in asynchronous mode and the reception ends abnormally.

[Setting condition]

- When a parity error is detected during reception
  Although receive data when the parity error occurs is transferred to the RDR register, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to the RDR register.

[Clearing condition]

- When 0 is written to the PER flag after reading PER = 1
  When setting the PER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the SCR.RE bit is set to 0 (serial reception is disabled), the PER flag is not affected and retains its previous value.

ERS Flag (Error Signal Status Flag)

[Setting condition]

- When a low error signal is sampled

[Clearing condition]

- When 0 is written to the ERS flag after reading ERS = 1
  When setting the ERS flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the SCR.RE bit is set to 0, the ERS flag is not affected and retains its previous value.

ORER Flag (Overrun Error Flag)

Indicates that an overrun error has occurred during reception and the reception ends abnormally.

[Setting condition]

- When the next data is received before receive data is read from the RDR register
  In the RDR register, the receive data prior to an overrun error occurrence is retained, but data received following the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed.

[Clearing condition]

- When 0 is written to the ORER flag after reading ORER = 1
  When setting the ORER flag to 0 to complete the interrupt handling, refer to section 14.4.1.2, Operation of Status Flags for Level-Detected Interrupts.
  Even when the SCR.RE bit is set to 0, the ORER flag is not affected and retains its previous value.

RDRF Flag (Receive Data Full Flag)

Indicates whether the RDR register has received data.

[Setting condition]

- When data has been received normally, and transferred from the RSR register to the RDR register

[Clearing condition]

- When data is read from the RDR register

TDRE Flag (Transmit Data Empty Flag)

Indicates whether the TDR register has data to be transmitted.

[Setting condition]

- When data is transferred from the TDR register to the TSR register

[Clearing condition]

- When data is written to the TDR register

23.2.10 Smart Card Mode Register (SCMR)

Address(es): SCI1.SCMR 0008 A026h, SCI5.SCMR 0008 A0A6h, SCI12.SCMR 0008 B306h, SMCI1.SCMR 0008 A026h, SMCI5.SCMR 0008 A0A6h, SMCI12.SCMR 0008 B306h

	b7	b6	b5	b4	b3	b2	b1	b0
	BCP2	—	—	CHR1	SDIR	SINV	—	SMIF
Value after reset:	1	1	1	1	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	SMIF	Smart Card Interface Mode Select	0: Non-smart card interface mode (Asynchronous mode, clock synchronous mode, simple SPI mode, or simple I2C mode) 1: Smart card interface mode	R/W*1
b1	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b2	SINV	Transmitted/Received Data Invert*2, *3	0: Data bits in the TDR register are transferred to the TSR register as they are. Data bits in the RSR register are transferred to the RDR register as they are. 1: Data bits in the TDR register are transferred to the TSR register with inverting. Data bits in the RSR register are transferred to the RDR register with inverting.	R/W*1
b3	SDIR	Transmitted/Received Data Transfer Direction*2, *4	0: Transfer with LSB first 1: Transfer with MSB first	R/W*1
b4	CHR1	Character Length 1*5	Selects in combination with the SMR.CHR bit. CHR1 CHR 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*6	R/W*1
b6, b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b7	BCP2	Base Clock Pulse 2	Selects the number of base clock cycles in combination with the SMR.BCP[1:0] bits. Table 23.9 lists the combinations of the SCMR.BCP2 bit and SMR.BCP[1:0] bits.	R/W*1

Note 1. Writable only when the SCR.TE bit is 0 and the SCR.RE bit is 0 (both serial transmission and reception are disabled).
Note 2. This bit can be used in the smart card interface mode, asynchronous mode (multi-processor mode), clock synchronous mode, and simple SPI mode.
Note 3. Set this bit to 0 if operation is to be in simple I2C mode.
Note 4. Set this bit to 1 if operation is to be in simple I2C mode.
Note 5. This bit is only valid in asynchronous mode. The setting is invalid and a fixed data length of 8 bits is used in modes other than asynchronous mode.
Note 6. LSB first should be selected and the value of MSB (b7) in the TDR register cannot be transmitted.

SMIF Bit (Smart Card Interface Mode Select)

When this bit is set to 1, smart card interface mode is selected.

When this bit is set to 0, non-smart card interface mode, i.e., asynchronous mode (including multi-processor mode), clock synchronous mode, simple SPI mode, or simple I2C mode is selected.

SINV Bit (Transmitted/Received Data Invert)

This bit is used to invert the logic level of the data bits when the data is transferred between data register and shift register. This bit does not affect the logic level of the parity bit. To invert the parity bit, invert the SMR.PM bit.

CHR1 Bit (Character Length 1)

Selects the data length of transmit/receive data.

Selects in combination with the SMR.CHR bit.

A fixed data length of 8 bits is used in modes other than asynchronous mode.

BCP2 Bit (Base Clock Pulse 2)

Selects the number of base clock cycles in a 1-bit data transfer time in smart card interface mode. Set this bit in combination with the SMR.BCP[1:0] bits.

Table 23.9 Combinations of the SCMR.BCP2 Bit and SMR.BCP[1:0] Bits

SCMR.BCP2 Bit	SMR.BCP[1:0] Bits		Number of Base Clock Cycles for 1-Bit Transfer Period
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

Note 1. S is the value of S in the BRR register (refer to section 23.2.11, Bit Rate Register (BRR)).

23.2.11 Bit Rate Register (BRR)

Address(es): SCI1.BRR 0008 A021h, SCI5.BRR 0008 A0A1h, SCI12.BRR 0008 B301h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

The BRR register is an 8-bit register that adjusts the bit rate.
As each SCI channel has independent baud rate generator control, different bit rates can be set for each. Table 23.10 shows the relationship between the setting (N) in the BRR register and the bit rate (B) for normal asynchronous mode, multi-processor communication, clock synchronous mode, smart card interface mode, simple SPI mode, and simple I²C mode.
The BRR register is writable only when the TE and RE bits in the SCR register are 0.

Table 23.10 Relationship between N Setting in the BRR Register and Bit Rate B

Mode	SEMR Settings		BRR Setting	Error (%)
	BGDM bit	ABCS bit		
Asynchronous, multi-processor communication	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	0	1	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	1	0		
	1	1	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI			$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times B} - 1$	
Smart card interface			$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times (N+1)} - 1 \right\} \times 100$
Simple I²C*1			$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	

B: Bit rate (bps)
N: BRR setting for on-chip baud rate generator ($0 \le N \le 255$)
PCLK: Operating frequency (MHz)
n and S: Determined by the settings of the SMR and SCMR registers as listed in Table 23.12 and Table 23.13.
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-bus standard.

Table 23.11 Calculating Widths at High and Low Level for SCL

Mode	SCL	Formula (Result in Seconds)
I²C	High period (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{PCLK \times 10^6}$
	Low period (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{PCLK \times 10^6}$

Table 23.12 Clock Source Settings

SMR.CKS[1:0] Bit Setting	Clock Source	n
0 0	PCLK	0
0 1	PCLK/4	1
1 0	PCLK/16	2
1 1	PCLK/64	3

Table 23.13 Base Clock Settings in Smart Card Interface Mode

SCMR.BCP2 Bit Setting	SMR.BCP[1:0] Bit Setting	Base Clock Cycles for 1-bit Period	S
0	0 0	93 clock cycles	93
0	0 1	128 clock cycles	128
0	1 0	186 clock cycles	186
0	1 1	512 clock cycles	512
1	0 0	32 clock cycles	32
1	0 1	64 clock cycles	64
1	1 0	372 clock cycles	372
1	1 1	256 clock cycles	256

Table 23.14 lists examples of N settings in the BRR register in normal asynchronous mode. Table 23.15 lists 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 23.18. Examples of BRR (N) settings in smart card interface mode are listed in Table 23.20. Examples of BRR (N) settings in simple I^2C mode are listed in Table 23.22. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, refer to section 23.6.4, Receive Data Sampling Timing and Reception Margin. Table 23.16 and Table 23.19 list the maximum bit rates with external clock input.

When either the SEMR.ABCS or BGDM bit is set to 1 in asynchronous mode, the bit rate becomes twice that listed in Table 23.14. When both of those bits are set to 1, the bit rate becomes four times the listed value.

Table 23.14 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	8			9.8304			10			12			12.288		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	141	0.03	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	25	0.16	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
31250	0	7	0.00	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	—	—	—	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	14			16			17.2032			18			19.6608		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	248	–0.17	3	70	0.03	3	75	0.48	3	79	–0.12	3	86	0.31
150	2	181	0.16	2	207	0.16	2	223	0.00	2	233	0.16	2	255	0.00
300	2	90	0.16	2	103	0.16	2	111	0.00	2	116	0.16	2	127	0.00
600	1	181	0.16	1	207	0.16	1	223	0.00	1	233	0.16	1	255	0.00
1200	1	90	0.16	1	103	0.16	1	111	0.00	1	116	0.16	1	127	0.00
2400	0	181	0.16	0	207	0.16	0	223	0.00	0	233	0.16	0	255	0.00
4800	0	90	0.16	0	103	0.16	0	111	0.00	0	116	0.16	0	127	0.00
9600	0	45	–0.93	0	51	0.16	0	55	0.00	0	58	–0.69	0	63	0.00
19200	0	22	–0.93	0	25	0.16	0	27	0.00	0	28	1.02	0	31	0.00
31250	0	13	0.00	0	15	0.00	0	16	1.20	0	17	0.00	0	19	–1.70
38400	—	—	—	0	12	0.16	0	13	0.00	0	14	–2.34	0	15	0.00

Bit Rate (bps)	Operating Frequency PCLK (MHz)								
	20			25			30		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	88	–0.25	3	110	–0.02	3	132	0.13
150	3	64	0.16	3	80	0.47	3	97	–0.35
300	2	129	0.16	2	162	–0.15	2	194	0.16
600	2	64	0.16	2	80	0.47	2	97	–0.35
1200	1	129	0.16	1	162	–0.15	1	194	0.16
2400	1	64	0.16	1	80	0.47	1	97	–0.35
4800	0	129	0.16	0	162	–0.15	0	194	0.16
9600	0	64	0.16	0	80	0.47	0	97	–0.35
19200	0	32	–1.36	0	40	–0.76	0	48	–0.35
31250	0	19	0.00	0	24	0.00	0	29	0.00
38400	0	15	1.73	0	19	1.73	0	23	1.73

Note: This is an example when the ABCS and BGDM bits in the SEMR register are 0.
When either the ABCS bit or BGDM bit is set to 1, the bit rate doubles.
When both ABCS and BGDM bits in the SEMR register are set to 1, the bit rate increases four times.

Table 23.15 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode)

	SEMR Settings						SEMR Settings				
PCLK (MHz)	BGDM Bit	ABCS Bit	n	N	Maximum Bit Rate (bps)	PCLK (MHz)	BGDM Bit	ABCS Bit	n	N	Maximum Bit Rate (bps)
8	0	0	0	0	250000	17.2032	0	0	0	0	537600
		1	0	0	500000			1	0	0	1075200
	1	0	0	0			1	0	0	0	
		1	0	0	1000000			1	0	0	2150400
9.8304	0	0	0	0	307200	18	0	0	0	0	562500
		1	0	0	614400			1	0	0	1125000
	1	0	0	0			1	0	0	0	
		1	0	0	1228800			1	0	0	2250000
10	0	0	0	0	312500	19.6608	0	0	0	0	614400
		1	0	0	625000			1	0	0	1228800
	1	0	0	0			1	0	0	0	
		1	0	0	1250000			1	0	0	2457600
12	0	0	0	0	375000	20	0	0	0	0	625000
		1	0	0	750000			1	0	0	1250000
	1	0	0	0			1	0	0	0	
		1	0	0	1500000			1	0	0	2500000
12.288	0	0	0	0	384000	25	0	0	0	0	781250
		1	0	0	768000			1	0	0	1562500
	1	0	0	0			1	0	0	0	
		1	0	0	1536000			1	0	0	3125000
14	0	0	0	0	437500	30	0	0	0	0	937500
		1	0	0	875000			1	0	0	1875000
	1	0	0	0			1	0	0	0	
		1	0	0	1750000			1	0	0	3750000
16	0	0	0	0	500000						
		1	0	0	1000000						
	1	0	0	0							
		1	0	0	2000000						

Table 23.16 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

		Maximum Bit Rate (bps)	
PCLK (MHz)	External Input Clock (MHz)	SEMR.ABCS Bit = 0	SEMR.ABCS Bit = 1
8	2.0000	125000	250000
9.8304	2.4576	153600	307200
10	2.5000	156250	312500
12	3.0000	187500	375000
12.288	3.0720	192000	384000
14	3.5000	218750	437500
16	4.0000	250000	500000
17.2032	4.3008	268800	537600
18	4.5000	281250	562500
19.6608	4.9152	307200	614400
20	5.0000	312500	625000
25	6.2500	390625	781250
30	7.5000	468750	937500

Table 23.17 Maximum Bit Rate with MTU Clock Input (Asynchronous Mode)

		Maximum Bit Rate (bps)	
PCLK (MHz)	MTU Clock (MHz)	SEMR.ABCS Bit = 0	SEMR.ABCS Bit = 1
8	4	250000	500000
9.8304	4.9152	307200	614400
10	5	312500	625000
12	6	375000	750000
12.288	6.144	384000	768000
14	7	437500	875000
16	8	500000	1000000
17.2032	8.6016	537600	1075200
18	9	562500	1125000
19.6608	9.8304	614400	1228800
20	10	625000	1250000
25	12.5	781250	1562500
30	15	937500	1875000

Table 23.18 BRR Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode)

	Operating Frequency PCLK (MHz)											
	8		10		16		20		25		30	
Bit Rate (bps)	n	N	n	N	n	N	n	N	n	N	n	N
110												
250	3	124	3	155	3	249						
500	2	249	3	77	3	124	3	155	3	194	3	233
1 k	2	124	2	155	2	249	3	77	3	97	3	116
2.5 k	1	199	1	249	2	99	2	124	2	155	2	187
5 k	1	99	1	124	1	199	1	249	2	77	2	93
10 k	0	199	0	249	1	99	1	124	1	155	1	187
25 k	0	79	0	99	0	159	0	199	0	249	1	74
50 k	0	39	0	49	0	79	0	99	0	124	0	149
100 k	0	19	0	24	0	39	0	49	0	62	0	74
250 k	0	7	0	9	0	15	0	19	0	24	0	29
500 k	0	3	0	4	0	7	0	9	—	—	0	14
1 M	0	1			0	3	0	4	—	—		
2 M	0	0*1			0	1			—	—		
2.5 M			0	0*1			0	1			0	2
4 M					0	0*1						
5 M							0	0*1				
6.25 M									0	0*1		
7.5 M											0	0*1

Blank cell: Cannot be set since the bit rate error exceeds 5%.

—: Can be set, but a bit rate error of 1 to 5% will occur.

Note 1. Continuous transmission or reception is not possible. After transmitting/receiving one frame of data, there is an interval of a 1-bit period before starting transmitting/receiving the next frame of data. The output of the synchronization clock is stopped for a 1-bit period. For this reason, it takes 9 bits worth of time to transfer one frame (8 bits) of data, and the average transfer rate is 8/9 times the bit rate.

Table 23.19 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode)

PCLK (MHz)	External Input Clock (MHz)	Maximum Bit Rate (Mbps)
8	1.3333	1.3333
10	1.6667	1.6667
12	2.0000	2.0000
14	2.3333	2.3333
16	2.6667	2.6667
18	3.0000	3.0000
20	3.3333	3.3333
25	4.1667	4.1667
30	5.0000	5.0000

Table 23.20 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372)

Bit Rate (bps)	PCLK (MHz)	n	N	Error (%)
9600	7.1424	0	0	0.00
	10.00	0	1	–30.00
	10.7136	0	1	–25.00
	13.00	0	1	–8.99
	14.2848	0	1	0.00
	16.00	0	1	12.01
	18.00	0	2	–15.99
	20.00	0	2	–6.66
	25.00	0	3	–12.49
	30.00	0	3	5.01

Table 23.21 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 32)

PCLK (MHz)	Maximum Bit Rate (bps)	n	N
10.00	156250	0	0
10.7136	167400	0	0
13.00	203125	0	0
16.00	250000	0	0
18.00	281250	0	0
20.00	312500	0	0
25.00	390625	0	0
30.00	468750	0	0

Table 23.22 BRR Settings for Various Bit Rates (Simple I²C Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)														
	8			10			16			20			25		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	0	24	0.0	0	31	–2.3	1	12	–3.8	1	15	–2.3	1	19	–2.3
25 k	0	9	0.0	0	12	–3.8	1	4	0.0	1	6	–10.7	1	7	–2.3
50 k	0	4	0.0	0	6	–10.7	1	2	–16.7	1	3	–21.9	1	3	–2.3
100 k	0	2	–16.7	0	3	–21.9	0	4	0.0	0	6	–10.7	1	1	–2.3
250 k	0	0	0.0	0	1	–37.5	0	1	0.0	0	2	–16.7	0	3	–21.9
350 k										0	1	–10.7	0	2	–25.6

Bit Rate (bps)	Operating Frequency PCLK (MHz)		
	30		
	n	N	Error (%)
10 k	1	23	–2.3
25 k	1	9	–6.3
50 k	1	4	–6.3
100 k	1	2	–21.9
250 k	0	3	–6.3
350 k	0	2	–10.7

Table 23.23 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple I²C Mode)

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	8			10			16			20		
	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)
10 k	0	24	43.75/50.00	0	31	44.80/51.20	1	12	45.50/52.00	1	15	44.80/51.20
25 k	0	9	17.50/20.00	0	12	18.20/20.80	1	4	17.50/20.00	1	6	19.60/22.40
50 k	0	4	8.75/10.00	0	6	9.80/11.20	1	2	10.50/12.00	1	3	11.20/12.80
100 k	0	2	5.25/6.00	0	3	5.60/6.40	0	4	4.37/5.00	0	6	4.90/5.60
250 k	0	0	1.75/2.00	0	1	2.80/3.20	0	1	1.75/2.00	0	2	2.10/2.40
350 k										0	1	1.40/1.60

Bit Rate (bps)	Operating Frequency PCLK (MHz)					
	25			30		
	n	N	Min. Widths at High/Low Level for SCL (µs)	n	N	Min. Widths at High/Low Level for SCL (µs)
10 k	1	19	44.80/51.20	1	23	44.80/51.20
25 k	1	7	17.92/20.48	1	9	18.66/21.33
50 k	1	3	8.96/10.24	1	4	9.33/10.66
100 k	1	1	4.48/5.12	1	2	5.60/6.40
250 k	0	3	2.24/2.56	0	3	1.86/2.13
350 k	0	2	1.68/1.92	0	2	1.40/1.60

23.2.12 Modulation Duty Register (MDDR)

Address(es): SCI1.MDDR 0008 A032h, SCI5.MDDR 0008 A0B2h, SCI12.MDDR 0008 B312h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

The MDDR register corrects the bit rate adjusted by the BRR register.

When the SEMR.BRME bit is set to 1, the bit rate generated by the on-chip baud rate generator is evenly corrected according to the settings of the MDDR register (M/256). The relationship between the MDDR register setting (M) and the bit rate (B) is given in Table 23.24.

The range of the value that can be set in the MDDR register is from 80h to FFh. A value other than these cannot be set.

The MDDR register is writable only when the TE and RE bits in the SCR register are 0.

Table 23.24 Relationship between MDDR Setting (M) and Bit Rate (B) When Bit Rate Modulation Function is Used

Mode	SEMR Settings		BRR Setting	Error (%)
	BGDM Bit	ABCS Bit		
Asynchronous, multi-processor communication	0	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	1	0	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
	0	1		
	1	1	$N = \frac{PCLK \times 10^6}{16 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times 16 \times 2^{2n-1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
Clock synchronous, simple SPI*1			$N = \frac{PCLK \times 10^6}{8 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	
Smart card interface			$N = \frac{PCLK \times 10^6}{S \times 2^{2n+1} \times \frac{256}{M} \times B} - 1$	$Error = \left\{ \frac{PCLK \times 10^6}{B \times S \times 2^{2n+1} \times \frac{256}{M} \times (N+1)} - 1 \right\} \times 100$
Simple I²C*2			$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times \frac{256}{M} \times B} - 1$	

B: Bit rate (bps)
M: MDDR setting (128 ≤ MDDR ≤ 256)
N: BRR setting for baud rate generator (0 ≤ N ≤ 255)
PCLK: Operating frequency (MHz)
n and S: Determined by the settings of the SMR and SCMR registers as listed in Table 23.12 and Table 23.13, section 23.2.11, Bit Rate Register (BRR).

Note 1. Do not use this function in clock synchronous mode and in the highest speed settings in simple SPI mode (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0).

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-bus standard.

Smaller settings of the SMR.CKS[1:0] bits and larger settings of the BRR register reduce difference in the length of the 1-bit period.

23.2.13 Serial Extended Mode Register (SEMR)

Address(es): SCI1.SEMR 0008 A027h, SCI5.SEMR 0008 A0A7h, SCI12.SEMR 0008 B307h

	b7	b6	b5	b4	b3	b2	b1	b0
	RXDESEL	BGDM	NFEN	ABCS	—	BRME	—	ACS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ACS0	Asynchronous Mode Clock Source Select	(Valid only in asynchronous mode) 0: External clock input 1: Logical AND of two compare matches output from MTU	R/W*1
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	BRME	Bit Rate Modulation Enable	0: Bit rate modulation function is disabled. 1: Bit rate modulation function is enabled.	R/W*1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	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*1
b5	NFEN	Digital Noise Filter Function Enable	(In asynchronous mode) 0: Noise cancellation function for the RXDn input signal is disabled. 1: Noise cancellation function for the RXDn input signal is enabled. (in simple I^2C mode) 0: Noise cancellation function for the SSCLn and SSDAn input signals is disabled. 1: Noise cancellation function for the SSCLn and SSDAn input signals is enabled. The NFEN bit should be 0 in any mode other than above.	R/W*1
b6	BGDM	Baud Rate Generator Double-Speed Mode Select	(Only valid the SCR.CKE[1] bit is 0 in asynchronous mode) 0: Baud rate generator outputs the clock with normal frequency. 1: Baud rate generator outputs the clock with doubled frequency.	R/W*1
b7	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*1

Note 1. Writable only when the SCR.TE bit is 0 and the SCR.RE bit is 0 (both serial transmission and reception are disabled).

The SEMR register is used to select a clock source for 1-bit period in asynchronous mode or a detection method of the start bit.

ACS0 Bit (Asynchronous Mode Clock Source Select)

Selects the clock source in the asynchronous mode.

The ACS0 bit is valid in asynchronous mode (SMR.CM bit = 0) and when an external clock input is selected (SCR.CKE[1:0] bits = 10b or 11b). This bit is used to select an external clock input or the logical AND of compare matches output from the internal MTU.

Set the ACS0 bit to 0 in other than asynchronous mode.

The MTIOC1A and MTIOC2A output of the MTU can be set as the base clock source. Refer to Table 23.25 for details.

Table 23.25 Correspondence between SCI Channels and Compare Match Outputs

SCI	MTU	Compare Match Output
SCI1	MTU1, MTU2	MTIOC1A, MTIOC2A
SCI5	MTU1, MTU2	MTIOC1A, MTIOC2A
SCI12	MTU1, MTU2	MTIOC1A, MTIOC2A

Figure 23.3 shows a setting example of when MTIOC1A and MTIOC2A in the MTU are selected for output.

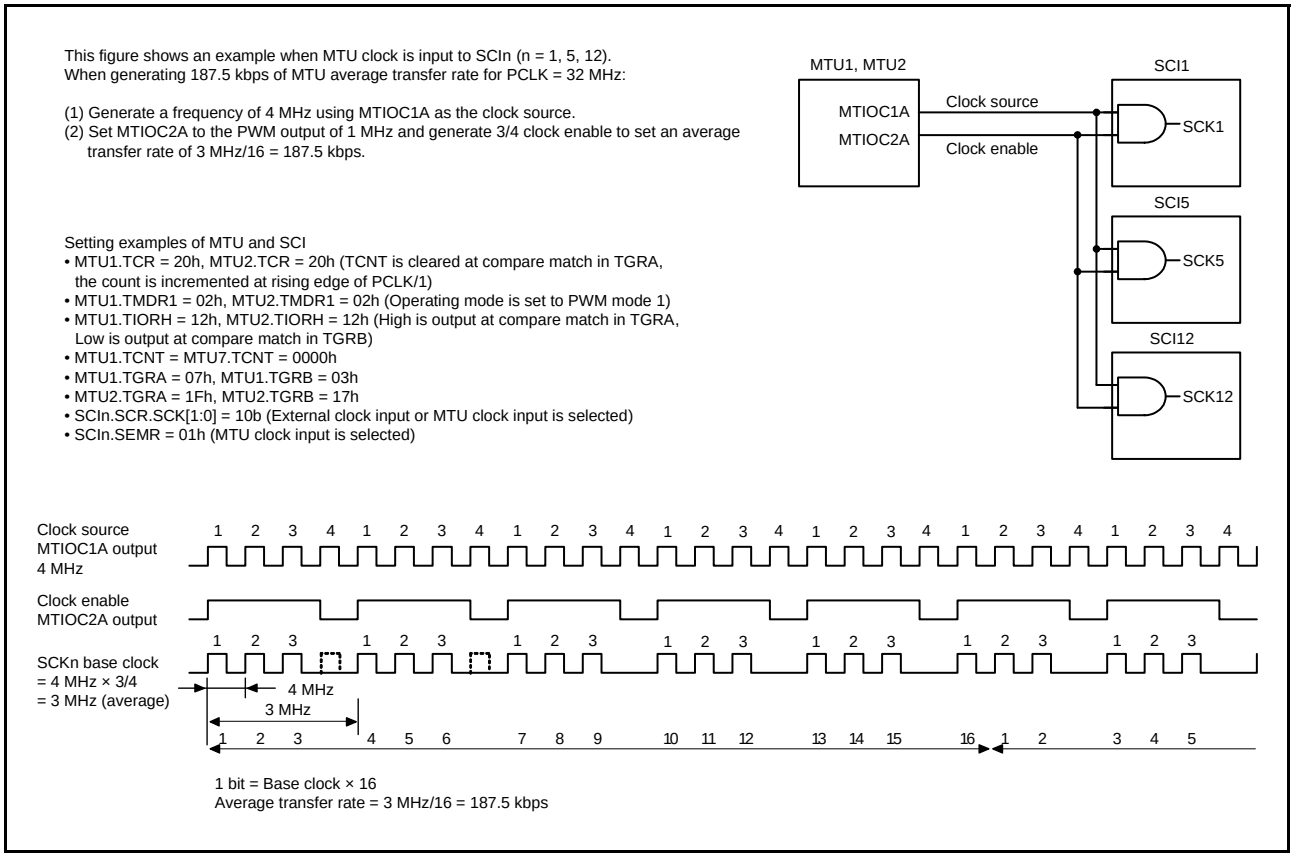


Figure 23.3 Example of Average Transfer Rate Setting When MTU Clock is Input

BRME Bit (Bit Rate Modulation Enable)

This bit enables and disables the bit rate modulation function. The bit rate generated by on-chip baud rate generator is evenly corrected when this function is enabled.

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 signal in asynchronous mode, and noise cancellation is applied to the SSDAn and SSCLn input signals in simple I^2C mode.
In any mode other than above, set the NFEN bit to 0 to disable the digital noise filter function.
When the function is disabled, input signals are transferred as is, as internal signals.

BGDM Bit (Baud Rate Generator Double-Speed Mode Select)

Selects the cycle of output clock for the baud rate generator.
This bit is valid when the on-chip baud rate generator is selected as the clock source (SCR.CKE[1] = 0) in asynchronous mode (SMR.CM = 0). For the clock output from the baud rate generator, either normal 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.

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 other than asynchronous mode.

23.2.14 Noise Filter Setting Register (SNFR)

Address(es): SCI1.SNFR 0008 A028h, SCI5.SNFR 0008 A0A8h, SCI12.SNFR 0008 B308h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	NFCS[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	NFCS[2:0]	Noise Filter Clock Select	In asynchronous mode, the standard setting for the base clock is as follows. b2 b0 0 0 0: The clock signal divided by 1 is used with the noise filter. In simple I2C mode, the standard settings for the clock source of the on-chip baud rate generator selected by the SMR.CKS[1:0] bits are given below. b2 b0 0 0 1: The clock signal divided by 1 is used with the noise filter. 0 1 0: The clock signal divided by 2 is used with the noise filter. 0 1 1: The clock signal divided by 4 is used with the noise filter. 1 0 0: The clock signal divided by 8 is used with the noise filter. Settings other than above are prohibited.	R/W*1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (serial reception and transmission disabled).

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 to 000b. In simple I2C mode, set the bits to a value in the range from 001b to 100b.

23.2.15 I²C Mode Register 1 (SIMR1)

Address(es): SCI1.SIMR1 0008 A029h, SCI5.SIMR1 0008 A0A9h, SCI12.SIMR1 0008 B309h

	b7	b6	b5	b4	b3	b2	b1	b0
	IICDL[4:0]					—	—	IICM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IICM	Simple I²C Mode Select	SMIF IICM 0 0: Asynchronous mode, Multi-processor mode, Clock synchronous mode (in asynchronous mode, synchronous, or simple SPI mode) 0 1: Simple I²C mode 1 0: Smart card interface mode 1 1: Setting prohibited.	R/W*1
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7 to b3	IICDL[4:0]	SSDA Output Delay Select	(Cycles below are of the clock signal from the on-chip baud rate generator.) b7 b3 0 0 0 0 0: No output delay 0 0 0 0 1: 0 to 1 cycle 0 0 0 1 0: 1 to 2 cycles 0 0 0 1 1: 2 to 3 cycles 0 0 1 0 0: 3 to 4 cycles 0 0 1 0 1: 4 to 5 cycles : : 1 1 1 1 0: 29 to 30 cycles 1 1 1 1 1: 30 to 31 cycles	R/W*1

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (both serial transmission and reception are disabled).

SIMR1 is used to select simple I²C mode and the number of delay stages for the SSDA output.

IICM Bit (Simple I²C Mode Select)

In conjunction with the SCMR.SMIF bit, this bit selects the operating mode.

IICDL[4:0] Bits (SSDA Output Delay Select)

These bits are used to set a delay for output on the SSDAn pin relative to the falling edge of the output on the SSCLn pin. The available delay settings range from no delay to 31 cycles, with the clock signal from the on-chip baud rate generator as the base. The signal obtained by frequency-dividing PCLK by the divisor set in the SMR.CKS[1:0] bits is supplied as the clock signal from the on-chip baud rate generator. Set these bits to 00000b unless operation is in simple I²C mode. In simple I²C mode, set the bits to a value in the range from 00001b to 11111b.

23.2.16 I²C Mode Register 2 (SIMR2)

Address(es): SCI1.SIMR2 0008 A02Ah, SCI5.SIMR2 0008 A0AAh, SCI12.SIMR2 0008 B30Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	IICACKT	—	—	—	IICCSC	IICINTM
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IICINTM	I²C Interrupt Mode Select	0: Use ACK/NACK interrupts. 1: Use reception and transmission interrupts.	R/W*1
b1	IICCSC	Clock Synchronization	0: No synchronization with the clock signal 1: Synchronization with the clock signal	R/W*1
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	IICACKT	ACK Transmission Data	0: ACK transmission 1: NACK transmission and reception of ACK/NACK	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (serial reception and transmission disabled).

SIMR2 is used to select how reception and transmission are controlled in simple I²C mode.

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 SSCLn clock signal is to be synchronized when the SSCLn pin has been placed at the low level in the case of a wait inserted by the other device, etc.

The SSCLn clock signal is not synchronized if the IICCSC bit is 0. The SSCLn clock signal is generated in accord with the rate selected in the BRR regardless of the level being input on the SSCLn 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.

23.2.17 I2C Mode Register 3 (SIMR3)

Address(es): SCI1.SIMR3 0008 A02Bh, SCI5.SIMR3 0008 A0ABh, SCI12.SIMR3 0008 B30Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	IICSCLS[1:0]		IICSDAS[1:0]		IICSTIF	IICSTPREQ	IICRSTAREQ	IICSTAREQ
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	IICSTAREQ	Start Condition Generation	0: A start condition is not generated. 1: A start condition is generated.*1, *3, *4, *5	R/W
b1	IICRSTAREQ	Restart Condition Generation	0: A restart condition is not generated. 1: A restart condition is generated.*2, *3, *4, *5	R/W
b2	IICSTPREQ	Stop Condition Generation	0: A stop condition is not generated. 1: A stop condition is generated.*2, *3, *4, *5	R/W
b3	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/W
b5, b4	IICSDAS[1:0]	SSDA Output Select	b5 b4 0 0: Serial data output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SSDAn pin. 1 1: Place the SSDAn pin in the high-impedance state.	R/W
b7, b6	IICSCLS[1:0]	SSCL Output Select	b7 b6 0 0: Serial clock output 0 1: Generate a start, restart, or stop condition. 1 0: Output the low level on the SSCLn pin. 1 1: Place the SSCLn pin in the high-impedance state.	R/W

Note 1. Generate a start condition only when the SSCLn and SSDAn pins are both high (the corresponding bits in the corresponding PIDR registers are 1).
Note 2. Generate a restart or stop condition only when the SSCLn pin is low (the corresponding bit in the PIDR register is 0).
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.

SIMR3 is used to control the simple I2C mode start and stop conditions, and to hold the SSDAn and SSCLn pins at fixed levels.

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.

[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 generation of the stop condition

IICSTIF Flag (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 bit to cause generation of a condition, do so after setting the IICSTIF flag to 0.
When the IICSTIF flag is 1 while an interrupt request is enabled by setting the SCR.TEIE bit, an STI request is output.
[Setting condition]

- Completion of the generation of a start, restart, or stop condition (however, in cases where this conflicts with any of the conditions for the flag becoming 0 listed below, the other condition takes precedence)

[Clearing conditions]

- Writing 0 to the bit (confirm that the IICSTIF flag is 0 before doing so)
- Writing 0 to the SIMR1.IICM bit (when operation is not in simple I^2C mode)
- Writing 0 to the SCR.TE bit

IICSDAS[1:0] Bits (SSDA Output Select)

These bits control output from the SSDAn pin.
Set the IICSDAS[1:0] and IICSCLS[1:0] bits to the same value during normal operations.

IICSCLS[1:0] Bits (SSCL Output Select)

These bits control output from the SSCLn pin.
Set the IICSCLS[1:0] and IICSDAS[1:0] bits to the same value during normal operations.

23.2.18 I²C Status Register (SISR)

Address(es): SCI1.SISR 0008 A02Ch, SCI5.SISR 0008 A0ACh, SCI12.SISR 0008 B30Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	IICACKR
Value after reset:	0	0	x	x	0	x	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	IICACKR	ACK Reception Data Flag	0: ACK received 1: NACK received	R/W*1
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b2	—	Reserved	The read value is undefined.	R
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	—	Reserved	The read value is undefined.	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 0 can be written to this bit, to clear the flag.

SISR is used to monitor state in relation to simple I²C mode.

IICACKR Flag (ACK Reception Data Flag)

Received ACK and NACK bits can be read from this bit.

The IICACKR flag is updated at the rising of SSCLn clock for the ACK/NACK receiving bit.

23.2.19 SPI Mode Register (SPMR)

Address(es): SCI1.SPMR 0008 A02Dh, SCI5.SPMR 0008 A0ADh, SCI12.SPMR 0008 B30Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	CKPH	CKPOL	—	MFF	—	MSS	CTSE	SSE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SSE	SSn# Pin Function Enable	0: SSn# pin function is disabled. 1: SSn# pin function is enabled.	R/W*1
b1	CTSE	CTS Enable	0: CTS function is disabled (RTS output function is enabled). 1: CTS function is enabled.	R/W*1
b2	MSS	Master Slave Select	0: Transmission is through the SMOSIn pin and reception is through the SMISOn pin (master mode). 1: Reception is through the SMOSIn pin and transmission is through the SMISOn pin (slave mode).	R/W*1
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	MFF	Mode Fault Flag	0: No mode fault error 1: Mode fault error	R/W*2
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	CKPOL	Clock Polarity Select	0: Clock polarity is not inverted. 1: Clock polarity is inverted.	R/W*1
b7	CKPH	Clock Phase Select	0: Clock is not delayed. 1: Clock is delayed.	R/W*1

Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR register are 0 (both serial transmission and reception are disabled).
Note 2. Only 0 can be written to these bits, which clears the flag.

SPMR is used to select the extension settings in asynchronous and clock synchronous modes.

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. Furthermore, even for usage in simple SPI mode, the SSn# pin on the master side is not required to control reception and transmission when master mode (SCR.CKE[1:0] = 00b and MSS = 0) is selected and there is a single master, so the setting for the SSE bit is 0. Do not set both the SSE and CTSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

CTSE Bit (CTS Enable)

Set this bit to 1 if the SSn# pin is to be used for inputting of the CTS control signal to control of transmission and reception. 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 I^2C mode. Do not set both the CTSE and SSE bits to enabled (even if this setting is made, operation is the same as that when these bits are set to 0).

MSS Bit (Master Slave Select)

This bit selects between master and slave operation in simple SPI mode. When the MSS bit is set to 1, data is received through the SMOSIn pin and transmitted through the SMISOn pin.
Set this bit to 0 in modes other than simple SPI mode.

MFF Flag (Mode Fault Flag)

This bit indicates mode fault errors.
In a multi-master configuration, 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 in simple SPI mode (SSE bit = 1 and MSS bit = 0)

[Clearing condition]

- Writing 0 to the bit after it was read as 1

CKPOL Bit (Clock Polarity Select)

This bit selects the polarity of the clock signal output through the SCKn pin. Refer to Figure 23.57 for details.
Set the bit to 0 in other than simple SPI mode and clock synchronous mode.

CKPH Bit (Clock Phase Select)

This bit selects the phase of the clock signal output through the SCKn pin. Refer to Figure 23.57 for details.
Set the bit to 0 in other than simple SPI mode and clock synchronous mode.

23.2.20 Extended Serial Module Enable Register (ESMER)

Address(es): SCI12.ESMER 0008 B320h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	ESME
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ESME	Extended Serial Mode Enable	0: The extended serial mode is disabled. 1: The extended serial mode is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ESME Bit (Extended Serial Mode Enable)

When the ESME bit is 1, the facilities of the extended serial mode control section are enabled.
When the ESME bit is 0, the extended serial mode control section is initialized.

Table 23.26 Settings of the ESME Bit and Timer Operation Mode

ESME Bit	Timer Mode	Break Field Low Width Determination Mode	Break Field Low Width Output Mode
0	Available*1	Not available	Not available
1	Available	Available	Available

Note 1. Operation is only possible with PCLK selected.

23.2.21 Control Register 0 (CR0)

Address(es): SCI12.CR0 0008 B321h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	BRME	RXDSF	SFSF	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	SFSF	Start Frame Status Flag	0: Start Frame detection function is disabled. 1: Start Frame detection function is enabled.	R
b2	RXDSF	RXDX12 Input Status Flag	0: RXDX12 input is enabled. 1: RXDX12 input is disabled.	R
b3	BRME	Bit Rate Measurement Enable	0: Measurement of bit rate is disabled. 1: Measurement of bit rate is enabled.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

23.2.22 Control Register 1 (CR1)

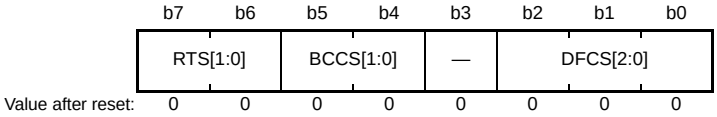
Address(es): SCI12.CR1 0008 B322h

	b7	b6	b5	b4	b3	b2	b1	b0
	PIBS[2:0]			PIBE	CF1DS[1:0]		CF0RE	BFE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BFE	Break Field Enable	0: Break Field detection is disabled. 1: Break Field detection is enabled.	R/W
b1	CF0RE	Control Field 0 Reception Enable	0: Reception of Control Field 0 is disabled. 1: Reception of Control Field 0 is enabled.	R/W
b3, b2	CF1DS[1:0]	Control Field 1 Data Register Select	b3 b2 0 0: Selects comparison with the value in the PCF1DR register. 0 1: Selects comparison with the value in the SCF1DR register. 1 0: Selects comparison with the values in the PCF1DR and SCF1DR registers. 1 1: Setting prohibited.	R/W
b4	PIBE	Priority Interrupt Bit Enable	0: The priority interrupt bit is disabled. 1: The priority interrupt bit is enabled.	R/W
b7 to b5	PIBS[2:0]	Priority Interrupt Bit Select	b7 b5 0 0 0: 0th bit of Control Field 1 0 0 1: 1st bit of Control Field 1 0 1 0: 2nd bit of Control Field 1 0 1 1: 3rd bit of Control Field 1 1 0 0: 4th bit of Control Field 1 1 0 1: 5th bit of Control Field 1 1 1 0: 6th bit of Control Field 1 1 1 1: 7th bit of Control Field 1	R/W

23.2.23 Control Register 2 (CR2)

Address(es): SCI12.CR2 0008 B323h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DFCS[2:0]	RXDX12 Signal Digital Filter Clock Select	b2 b0 0 0 0: Filter is disabled. 0 0 1: Filter clock is base clock*1, *2 0 1 0: Filter clock is PCLK/8 0 1 1: Filter clock is PCLK/16 1 0 0: Filter clock is PCLK/32 1 0 1: Filter clock is PCLK/64 1 1 0: Filter clock is PCLK/128 1 1 1: Setting prohibited	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	BCCS[1:0]	Bus Collision Detection Clock Select	• When SEMR.BGDM = 0 or SEMR.BGDM = 1 and SMR.CKS[1:0] = a value other than 00b b5 b4 0 0: Base clock 0 1: Base clock frequency divided by 2 1 0: Base clock frequency divided by 4 1 1: Setting prohibited • When SEMR.BGDM = 1 and SMR.CKS[1:0] = 00b b5 b4 0 0: Base clock frequency divided by 2 0 1: Base clock frequency divided by 4 1 0: Setting prohibited 1 1: Setting prohibited	R/W
b7, b6	RTS[1:0]	RXDX12 Reception Sampling Timing Select	• When SCI12.SEMR.ABCS = 0 b7 b6 0 0: Rising edge of the 8th cycle of base clock 0 1: Rising edge of the 10th cycle of base clock 1 0: Rising edge of the 12th cycle of base clock 1 1: Rising edge of the 14th cycle of base clock • When SCI12.SEMR.ABCS = 1 b7 b6 0 0: Rising edge of the 4th cycle of base clock 0 1: Rising edge of the 5th cycle of base clock 1 0: Rising edge of the 6th cycle of base clock 1 1: Rising edge of the 7th cycle of base clock	R/W

Note: The period of the base clock is 1/16 of a single bit period when the SCI12.SEMR.ABCS is 0, and 1/8 of a single bit period when the SCI12.SEMR.ABCS is 1.
Note 1. To use the base clock, set the SCI12.SCR.TE bit to 1.
Note 2. The base clock divided by 2 is the filter clock when the SEMR.BGDM bit is 1 and the SMR.CKS[1:0] bits are 00b.

23.2.24 Control Register 3 (CR3)

Address(es): SCI12.CR3 0008 B324h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SDST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SDST	Start Frame Detection Start	0: Detection of Start Frame is not performed. 1: Detection of Start Frame is performed.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SDST Bit (Start Frame Detection Start)

Detection of a Start Frame begins when this bit is set to 1. The bit is read as 0.

23.2.25 Port Control Register (PCR)

Address(es): SCI12.PCR 0008 B325h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	SHARPS	—	—	RXDXPS	TXDXPS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXDXPS	TXDX12 Signal Polarity Select	0: The polarity of TXDX12 signal is not inverted for output. 1: The polarity of TXDX12 signal is inverted for output.	R/W
b1	RXDXPS	RXDX12 Signal Polarity Select	0: The polarity of RXDX12 signal is not inverted for input. 1: The polarity of RXDX12 signal is inverted for input.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SHARPS	TXDX12/RXDX12 Pin Multiplexing Select	0: The TXDX12 and RXDX12 pins are independent. 1: The TXDX12 and RXDX12 signals are multiplexed on the same pin.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SHARPS Bit (TXDX12/RXDX12 Pin Multiplexing Select)

When this bit is set to 1, the TXDX12 and RXDX12 signals are multiplexed on the same pin so that half-duplex communications become possible.

23.2.26 Interrupt Control Register (ICR)

Address(es): SCI12.ICR 0008 B326h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AEDIE	BCDIE	PIBDIE	CF1MIE	CF0MIE	BFDIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BFDIE	Break Field Low Width Detected Interrupt Enable	0: Interrupts on detection of the low width for a Break Field are disabled. 1: Interrupts on detection of the low width for a Break Field are enabled.	R/W
b1	CF0MIE	Control Field 0 Match Detected Interrupt Enable	0: Interrupts on detection of a match with Control Field 0 are disabled. 1: Interrupts on detection of a match with Control Field 0 are enabled.	R/W
b2	CF1MIE	Control Field 1 Match Detected Interrupt Enable	0: Interrupts on detection of a match with Control Field 1 are disabled. 1: Interrupts on detection of a match with Control Field 1 are enabled.	R/W
b3	PIBDIE	Priority Interrupt Bit Detected Interrupt Enable	0: Interrupts on detection of the priority interrupt bit are disabled. 1: Interrupts on detection of the priority interrupt bit are enabled.	R/W
b4	BCDIE	Bus Collision Detected Interrupt Enable	0: Interrupts on detection of a bus collision are disabled. 1: Interrupts on detection of a bus collision are enabled.	R/W
b5	AEDIE	Valid Edge Detected Interrupt Enable	0: Interrupts on detection of a valid edge are disabled. 1: Interrupts on detection of a valid edge are enabled.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

23.2.27 Status Register (STR)

Address(es): SCI12.STR 0008 B327h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AEDF	BCDF	PIBDF	CF1MF	CF0MF	BFDF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BFDF	Break Field Low Width Detection Flag	[Setting conditions] • Detection of the low width for a Break Field • Completion of the output of the low width for a Break Field • Underflow of the timer [Clearing condition] • Writing 1 to the STCR.BFDCL bit	R
b1	CF0MF	Control Field 0 Match Flag	[Setting condition] • A match between the value received in Control Field 0 and the set value. [Clearing condition] • Writing 1 to the STCR.CF0MCL bit	R
b2	CF1MF	Control Field 1 Match Flag	[Setting condition] • A match between the data received in Control Field 1 and the set values. [Clearing condition] • Writing 1 to the STCR.CF1MCL bit	R
b3	PIBDF	Priority Interrupt Bit Detection Flag	[Setting condition] • Detection of the priority interrupt bit [Clearing condition] • Writing 1 to the STCR.PIBDCL bit	R
b4	BCDF	Bus Collision Detected Flag	[Setting condition] • Detection of the bus collision [Clearing condition] • Writing 1 to the STCR.BCDCL bit	R
b5	AEDF	Valid Edge Detection Flag	[Setting condition] • Detection of a valid edge [Clearing condition] • Writing 1 to the STCR.AEDCL bit	R
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R

23.2.28 Status Clear Register (STCR)

Address(es): SCI12.STCR 0008 B328h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	AEDCL	BCDCL	PIBDCL	CF1MCL	CF0MCL	BFDCL
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BFDCL	BFDF Clear	Setting this bit to 1 clears the STR.BFDF flag. This bit is read as 0.	R/W
b1	CF0MCL	CF0MF Clear	Setting this bit to 1 clears the STR.CF0MF flag. This bit is read as 0.	R/W
b2	CF1MCL	CF1MF Clear	Setting this bit to 1 clears the STR.CF1MF flag. This bit is read as 0.	R/W
b3	PIBDCL	PIBDF Clear	Setting this bit to 1 clears the STR.PIBDF flag. This bit is read as 0.	R/W
b4	BCDCL	BCDF Clear	Setting this bit to 1 clears the STR.BCDF flag. This bit is read as 0.	R/W
b5	AEDCL	AEDF Clear	Setting this bit to 1 clears the STR.AEDF flag. This bit is read as 0.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

23.2.29 Control Field 0 Data Register (CF0DR)

Address(es): SCI12.CF0DR 0008 B329h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

The CF0DR register is an 8-bit readable and writable register that holds a value for comparison with Control Field 0.

23.2.30 Control Field 0 Compare Enable Register (CF0CR)

Address(es): SCI12.CF0CR 0008 B32Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	CF0CE7	CF0CE6	CF0CE5	CF0CE4	CF0CE3	CF0CE2	CF0CE1	CF0CE0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CF0CE0	Control Field 0 Bit 0 Compare Enable	0: Comparison with bit 0 of Control Field 0 is disabled. 1: Comparison with bit 0 of Control Field 0 is enabled.	R/W
b1	CF0CE1	Control Field 0 Bit 1 Compare Enable	0: Comparison with bit 1 of Control Field 0 is disabled. 1: Comparison with bit 1 of Control Field 0 is enabled.	R/W
b2	CF0CE2	Control Field 0 Bit 2 Compare Enable	0: Comparison with bit 2 of Control Field 0 is disabled. 1: Comparison with bit 2 of Control Field 0 is enabled.	R/W
b3	CF0CE3	Control Field 0 Bit 3 Compare Enable	0: Comparison with bit 3 of Control Field 0 is disabled. 1: Comparison with bit 3 of Control Field 0 is enabled.	R/W
b4	CF0CE4	Control Field 0 Bit 4 Compare Enable	0: Comparison with bit 4 of Control Field 0 is disabled. 1: Comparison with bit 4 of Control Field 0 is enabled.	R/W
b5	CF0CE5	Control Field 0 Bit 5 Compare Enable	0: Comparison with bit 5 of Control Field 0 is disabled. 1: Comparison with bit 5 of Control Field 0 is enabled.	R/W
b6	CF0CE6	Control Field 0 Bit 6 Compare Enable	0: Comparison with bit 6 of Control Field 0 is disabled. 1: Comparison with bit 6 of Control Field 0 is enabled.	R/W
b7	CF0CE7	Control Field 0 Bit 7 Compare Enable	0: Comparison with bit 7 of Control Field 0 is disabled. 1: Comparison with bit 7 of Control Field 0 is enabled.	R/W

23.2.31 Control Field 0 Receive Data Register (CF0RR)

Address(es): SCI12.CF0RR 0008 B32Bh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

CF0RR is a readable register that holds the value received in Control Field 0.

23.2.32 Primary Control Field 1 Data Register (PCF1DR)

Address(es): SCI12.PCF1DR 0008 B32Ch

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

PCF1DR is an 8-bit readable and writable register that holds the 8-bit primary value for comparison with Control Field 1.

23.2.33 Secondary Control Field 1 Data Register (SCF1DR)

Address(es): SCI12.SCF1DR 0008 B32Dh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

PCF1DR is an 8-bit readable and writable register that holds the 8-bit secondary value for comparison with Control Field 1.

23.2.34 Control Field 1 Compare Enable Register (CF1CR)

Address(es): SCI12.CF1CR 0008 B32Eh

	b7	b6	b5	b4	b3	b2	b1	b0
	CF1CE7	CF1CE6	CF1CE5	CF1CE4	CF1CE3	CF1CE2	CF1CE1	CF1CE0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CF1CE0	Control Field 1 Bit 0 Compare Enable	0: Comparison with bit 0 of Control Field 1 is disabled. 1: Comparison with bit 0 of Control Field 1 is enabled.	R/W
b1	CF1CE1	Control Field 1 Bit 1 Compare Enable	0: Comparison with bit 1 of Control Field 1 is disabled. 1: Comparison with bit 1 of Control Field 1 is enabled.	R/W
b2	CF1CE2	Control Field 1 Bit 2 Compare Enable	0: Comparison with bit 2 of Control Field 1 is disabled. 1: Comparison with bit 2 of Control Field 1 is enabled.	R/W
b3	CF1CE3	Control Field 1 Bit 3 Compare Enable	0: Comparison with bit 3 of Control Field 1 is disabled. 1: Comparison with bit 3 of Control Field 1 is enabled.	R/W
b4	CF1CE4	Control Field 1 Bit 4 Compare Enable	0: Comparison with bit 4 of Control Field 1 is disabled. 1: Comparison with bit 4 of Control Field 1 is enabled.	R/W
b5	CF1CE5	Control Field 1 Bit 5 Compare Enable	0: Comparison with bit 5 of Control Field 1 is disabled. 1: Comparison with bit 5 of Control Field 1 is enabled.	R/W
b6	CF1CE6	Control Field 1 Bit 6 Compare Enable	0: Comparison with bit 6 of Control Field 1 is disabled. 1: Comparison with bit 6 of Control Field 1 is enabled.	R/W
b7	CF1CE7	Control Field 1 Bit 7 Compare Enable	0: Comparison with bit 7 of Control Field 1 is disabled. 1: Comparison with bit 7 of Control Field 1 is enabled.	R/W

23.2.35 Control Field 1 Receive Data Register (CF1RR)

Address(es): SCI12.CF1RR 0008 B32Fh

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

CF1RR is a readable register that holds the value received in Control Field 1.

23.2.36 Timer Control Register (TCR)

Address(es): SCI12.TCR 0008 B330h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	TCST
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TCST	Timer Count Start	0: Stops the timer counting 1: Starts the timer counting	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

23.2.37 Timer Mode Register (TMR)

Address(es): SCI12.TMR 0008 B331h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	TCSS[2:0]			TWRC	—	TOMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOMS[1:0]	Timer Operating Mode Select*1	b1 b0 0 0: Timer mode 0 1: Break Field low width determination mode 1 0: Break Field low width output mode 1 1: Setting prohibited	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	TWRC	Counter Write Control	0: Data is written to the reload register and counter 1: Data is written to the reload register only	R/W
b6 to b4	TCSS[2:0]	Timer Count Clock Source Select*1	b6 b4 0 0 0: PCLK 0 0 1: PCLK/2 0 1 0: PCLK/4 0 1 1: PCLK/8 1 0 0: PCLK/16 1 0 1: PCLK/32 1 1 0: PCLK/64 1 1 1: PCLK/128	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Rewrite the TOMS[1:0] and TCSS[2:0] bits only when the timer is stopped (TCST = 0).

TWRC Bit (Counter Write Control)

This bit determines whether a value written to the TPRE or TCNT register is written to the reload register only or is written to both the reload register and the counter.

23.2.38 Timer Prescaler Register (TPRE)

Address(es): SCI12.TPRE 0008 B332h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

TPRE consists of an 8-bit reload register, a read buffer, and a counter, each of which has FFh as its initial value. The counter counts down in synchronization with the counter clock selected by the TMR.TCSS[2:0] bits, and is reloaded with the value in the reload register when it underflows. Underflows of this register provide the clock source to drive counting by the TCNT register. The reload register and read buffer are allocated to the same address. Data is written to the reload register in writing, and the counter value that has been transferred to the read buffer is returned in reading.
It takes one PCLK cycle to load a value from the reload register to the counter.

23.2.39 Timer Count Register (TCNT)

Address(es): SCI12.TCNT 0008 B333h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

TCNT consists of an 8-bit reload register, a read buffer, and a counter, each of which has FFh as its initial value. This down-counter counts underflows of the TPRE register until the TCNT register underflows, and is then reloaded with the value from the reload register. The reload register and read buffer are allocated to the same address. Data is written to the reload register in writing, and the counter value that has been transferred to the read buffer is returned in reading.
It takes one PCLK cycle to load a value from the reload register to the counter.

23.3 Operation in Asynchronous Mode

Figure 23.4 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 usually held in the mark state (high level).
The SCI monitors the communications line. When the SCI detects a low, it regards that as a start bit and starts serial communication.
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, so that data can be read or written during transmission or reception, enabling continuous data transmission and reception.

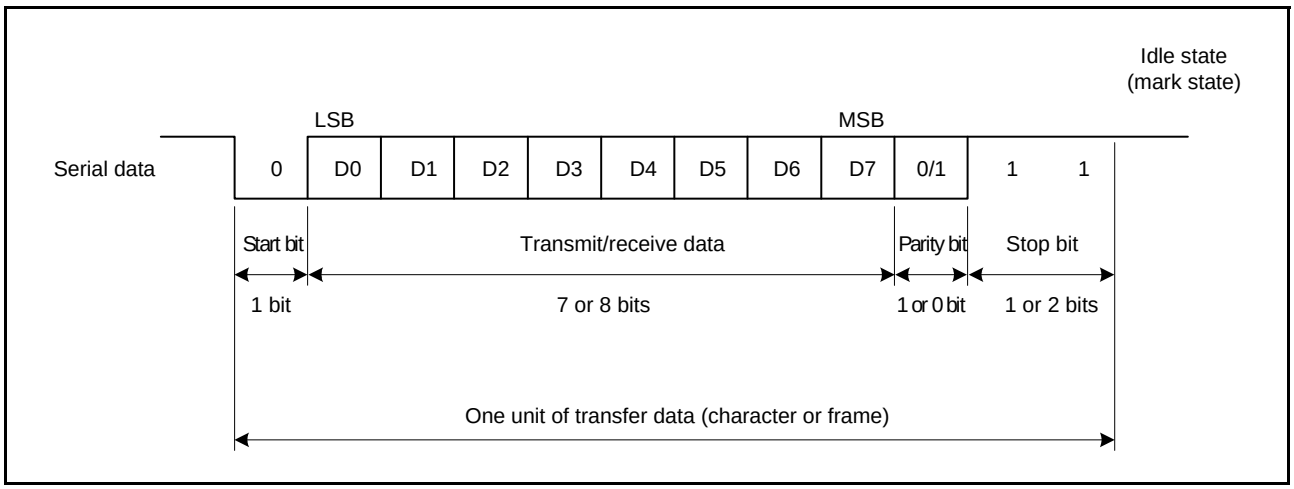


Figure 23.4 Data Format in Asynchronous Serial Communications (Example with 8-Bit Data, Parity, 2 Stop Bits)

23.3.1 Serial Data Transfer Format

Table 23.27 lists the serial data transfer formats that can be used in asynchronous mode.
Any of 18 transfer formats can be selected according to the SMR and SCMR setting. For details of multi-processor function, refer to section 23.4, Multi-Processor Communications Function.

Table 23.27 Serial Transfer Formats (Asynchronous Mode)

SCMR Setting	SMR Setting				Serial Transfer Format and Frame Length
CHR1	CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12 13
0	0	0	0	0	S 9-bit data STOP
0	0	0	0	1	S 9-bit data STOP STOP
0	0	1	0	0	S 9-bit data P STOP
0	0	1	0	1	S 9-bit data P STOP STOP
1	0	0	0	0	S 8-bit data STOP
1	0	0	0	1	S 8-bit data STOP STOP
1	0	1	0	0	S 8-bit data P STOP
1	0	1	0	1	S 8-bit data P STOP STOP
1	1	0	0	0	S 7-bit data STOP
1	1	0	0	1	S 7-bit data STOP STOP
1	1	1	0	0	S 7-bit data P STOP
1	1	1	0	1	S 7-bit data P STOP STOP
0	0	—	1	0	S 9-bit data MPB STOP
0	0	—	1	1	S 9-bit data MPB STOP STOP
1	0	—	1	0	S 8-bit data MPB STOP
1	0	—	1	1	S 8-bit data MPB STOP STOP
1	1	—	1	0	S 7-bit data MPB STOP
1	1	—	1	1	S 7-bit data MPB STOP STOP

S: Start bit
STOP: Stop bit
P: Parity bit
MPB: Multi-processor bit

23.3.2 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. Since 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 23.5. Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ (\%) \quad \cdots \text{Formula (1)}$$

M: Reception margin
N: Ratio of bit rate to clock
(N = 16 when SEMR.ABCS = 0, N = 8 when SEMR.ABCS = 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 F = 0 and D = 0.5 in formula (1), the reception margin is determined by the formula below.

M = {0.5 – 1/(2 × 16)} × 100 (%) = 46.875 (%)

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

Note 1. These are values when the SEMR.ABCS bit is 0. When the ABCS bit is 1, 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.

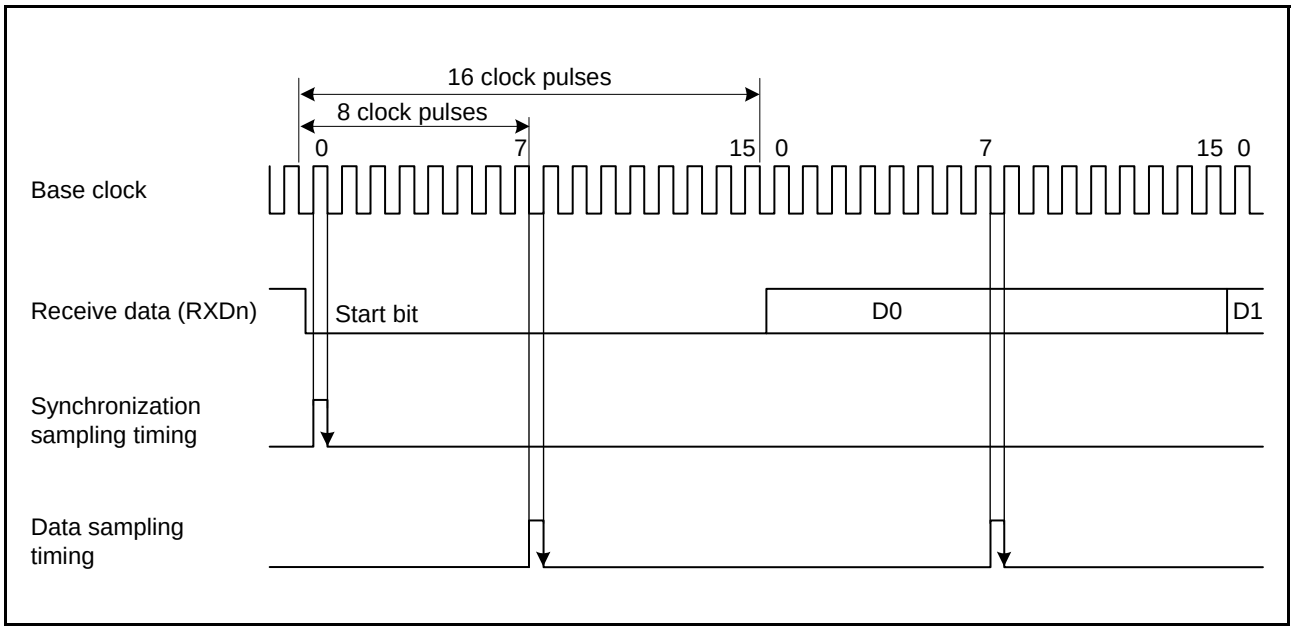


Figure 23.5 Receive Data Sampling Timing in Asynchronous Mode

23.3.3 Clock

Either an internal clock generated by the on-chip 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 the SMR.CM bit and the SCR.CKE[1:0] bits.
When an external clock is input to the SCKn pin, the clock frequency should be 16 times the bit rate (when SEMR.ABCS bit = 0) and 8 times the bit rate (when SEMR.ABCS bit = 1). In addition, when an external clock is specified, the base clock of MTIOC1A and MTIOC2A can be selected by the SCIn.SEMR.ACS0 bit (n = 1, 5, 12).
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 23.6.

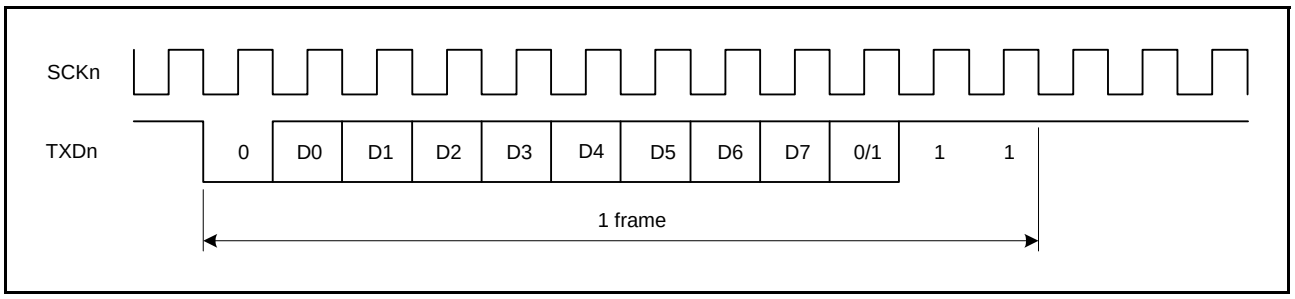


Figure 23.6 Phase Relationship between Output Clock and Transmit Data (Asynchronous Mode: SMR.CHR = 0, PE = 1, MP = 0, STOP = 1)

23.3.4 Double-Speed Mode

The output clock frequency of the on-chip baud rate generator is doubled by setting the SEMR.BGDM bit to 1, enabling high-speed communication at a doubled bit rate. If the SEMR.ABCS bit is set to 1 under the above condition, the number of base clock cycles changes from 16 to 8, so the bit rate becomes four times faster than the initial state.
As shown by Formula (1) in section 23.3.2, Receive Data Sampling Timing and Reception Margin in Asynchronous Mode, setting the SEMR.ABCS bit to 1 changes the number of cycles to 8, and the sampling interval becomes longer. This causes the reception margin to decrease. Therefore, setting the SEMR.BGDM bit to 1 and the SEMR.ABCS bit to 0 is recommended instead of setting the SEMR.BGDM bit to 0 and the SEMR.ABCS bit to 1 for high-speed operation at a doubled bit rate.

23.3.5 CTS and RTS Functions

The CTS function is the use of input on the CTSn# pin in transmission control. Setting the SPMR.CTSE bit to 1 enables the CTS function. When the CTS function is enabled, placing the low level on the CTSn# pin causes transmission to start.
Applying the high level to the CTS# pin while transmission is in progress does not affect transmission of the current frame, which continues.
In the RTS function, by using the function of output on the RTSn# pin, a low level is output when reception becomes possible. Conditions for output of the low and high level are shown below.

[Conditions for low-level output]
When the following conditions are all satisfied.

- The SCR.RE bit is 1.
- Reception is not in progress.
- There are no received data yet to be read.
- The ORER, FER, and PER flags in the SSR register are all 0.

[Condition for high-level output]
When the conditions for low-level output are not satisfied.

Note that either one of CTS and RTS can be selected.

23.3.6 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, start by writing the initial value 00h to the SCR register and then continue through the procedure for SCI given in Figure 23.7. Whenever the operating mode or transfer format is changed, the SCR register 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 SCR.RE bit to 0 initializes neither the ORER, FER, PER, and RDRF flags in the SSR register nor registers RDR, RDRH, and RDRL.
In addition, note that setting bits TIE, TE, and TEIE in the SCR register to 1 simultaneously leads to the generation of a transmit end interrupt (TEI) request before the generation of a transmit data empty interrupt (TXI) request.

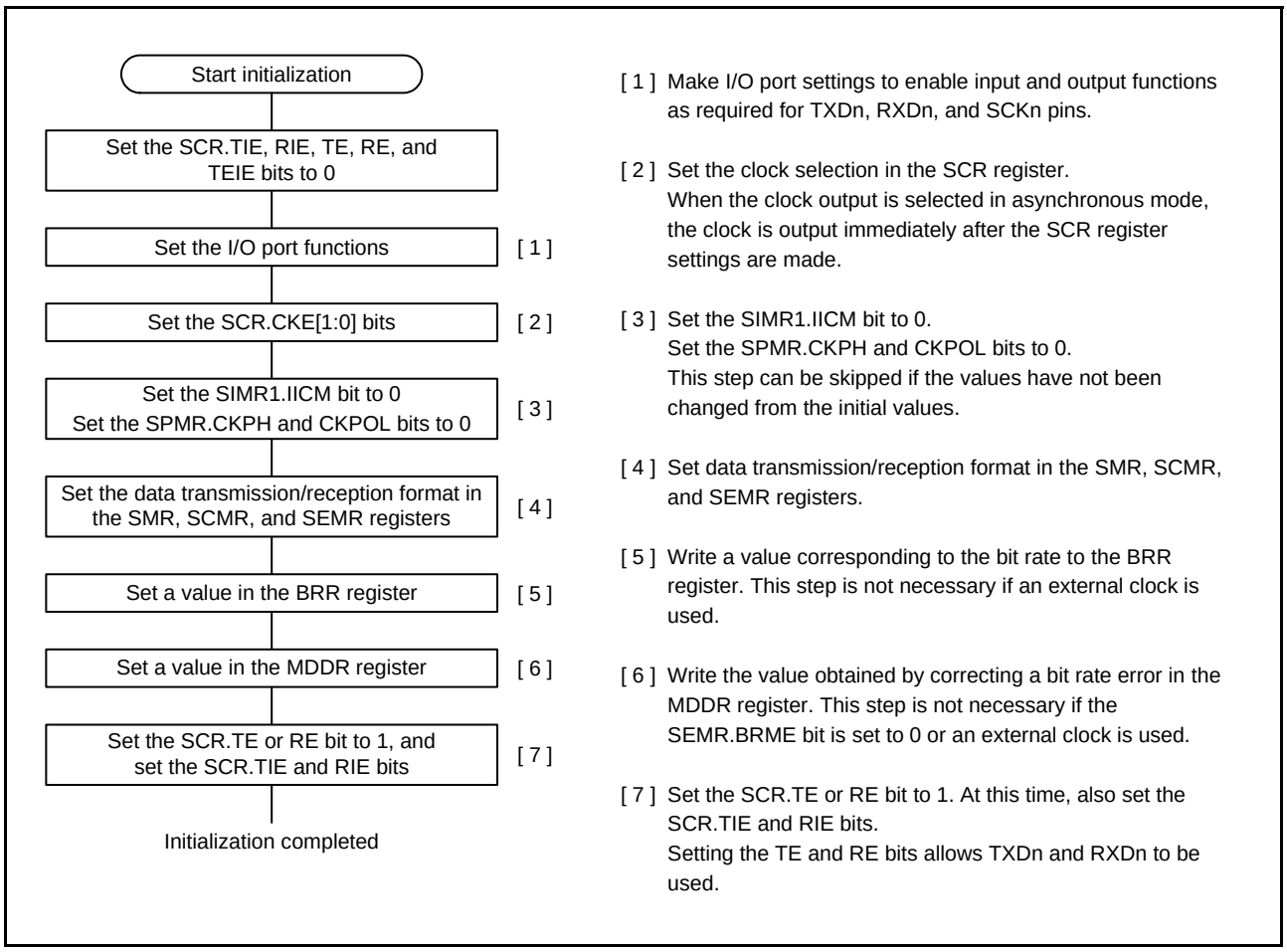


Figure 23.7 Sample SCI Initialization Flowchart (Asynchronous Mode)

Figure 23.8 shows an example of data transmission when the SCI is set to asynchronous mode according to the flow described in Figure 23.7 after a reset. When the pin function is set to the TXD pin, it is still high-impedance because the SCR.TE bit is 0. When the transmit data is written after setting the TE bit to 1, a data transmission starts. After the TE bit is set to 1, one frame of high is output from TXD pin (internal wait time) and then the data transmission starts.

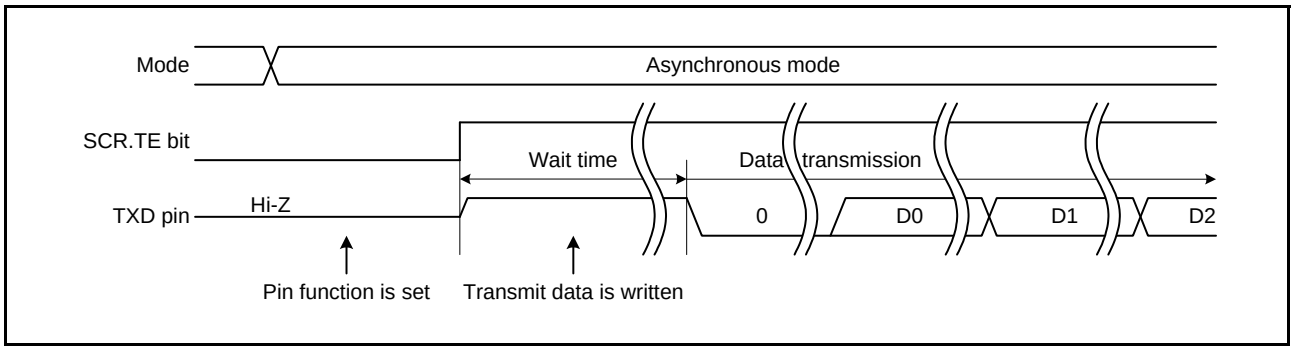


Figure 23.8 Example of Data Transmission Timing in Asynchronous Mode

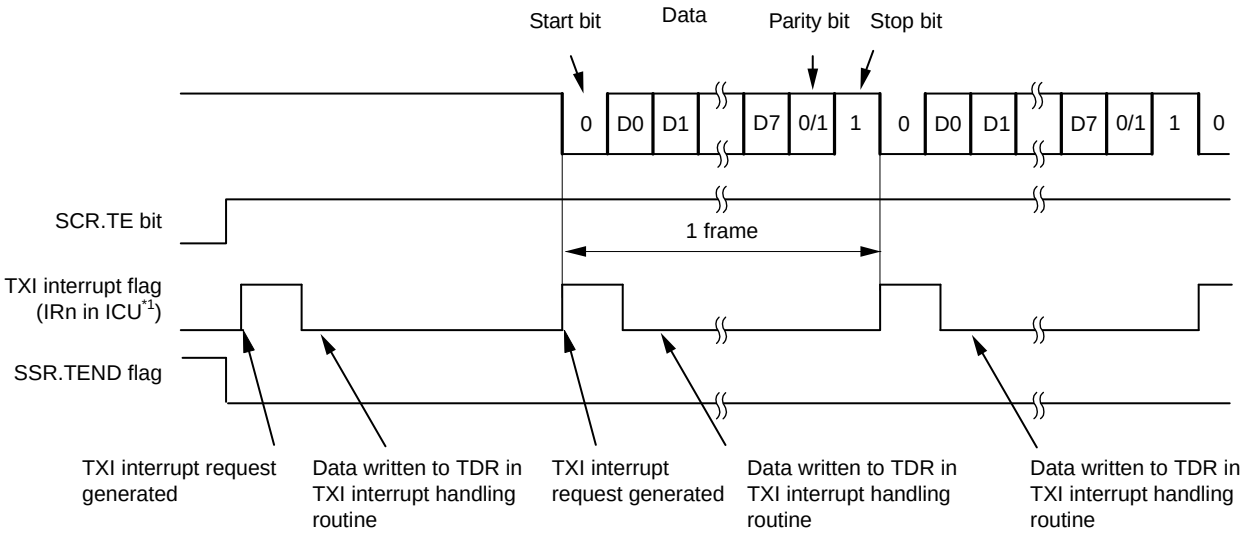
23.3.7 Serial Data Transmission (Asynchronous Mode)

Figure 23.9 to Figure 23.11 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 the TDR register*1 to the TSR register when data is written to the TDR register*1 in the TXI interrupt handling routine. The TXI interrupt request at the beginning of transmission is generated when the SCR.TE bit is set to 1 after the SCR.TIE bit is set to 1 or when these 2 bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the SPMR.CTSE bit is set to 0 (CTS function is disabled) and a low level on the CTSn# pin causes data transfer from the TDR register*1 to the TSR register. If the SCR.TIE bit is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next transmit data to the TDR register*1 in the TXI interrupt handling routine before transmission of the current transmit data is completed. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR.TEIE bit to 1 (a TEI interrupt request is enabled) after the last of the data to be transmitted are written to the TDR register*1, *2 from the handling routine for TXI requests.
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) the TDR register*3 at the time of stop bit output.
5. When the TDR register*3 is updated, setting of the SPMR.CTSE bit 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 the TDR register*1 to the TSR register and sending of the stop bit, after which serial transmission of the next frame starts.
6. If the TDR register*3 is not updated, the SCI sets the SSR.TEND flag to 1, sends the stop bit, and then outputs high to put the line in the mark state. If the SCR.TEIE bit is 1 at this time, the SSR.TEND flag is set to 1 and a TEI interrupt request is generated.

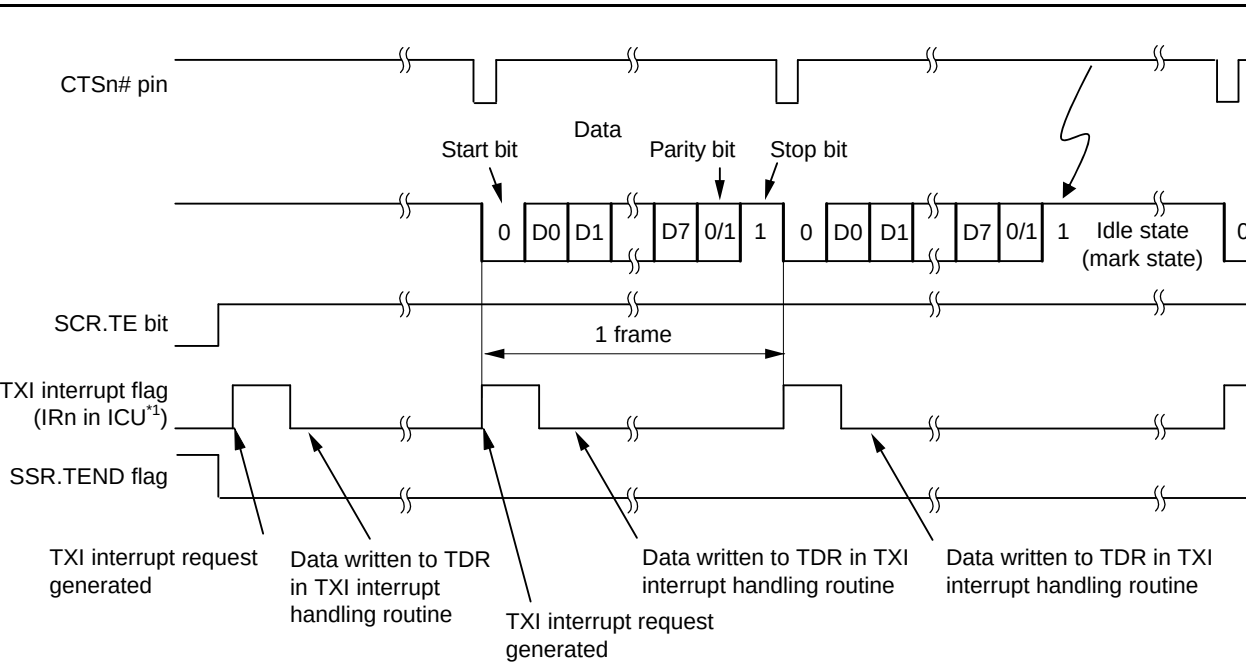
Note 1. Write data not to the TDR register but to the TDRH and TDRL registers when 9-bit data length is selected.
Note 2. Write data in the order from the TDRH register to the TDRL register when 9-bit data length is selected.
Note 3. The SCI checks for updating of the TDRL register only and does not check for updating of the TDRH register when 9-bit data length is selected.

Figure 23.12 shows a sample flowchart for serial transmission in asynchronous mode.



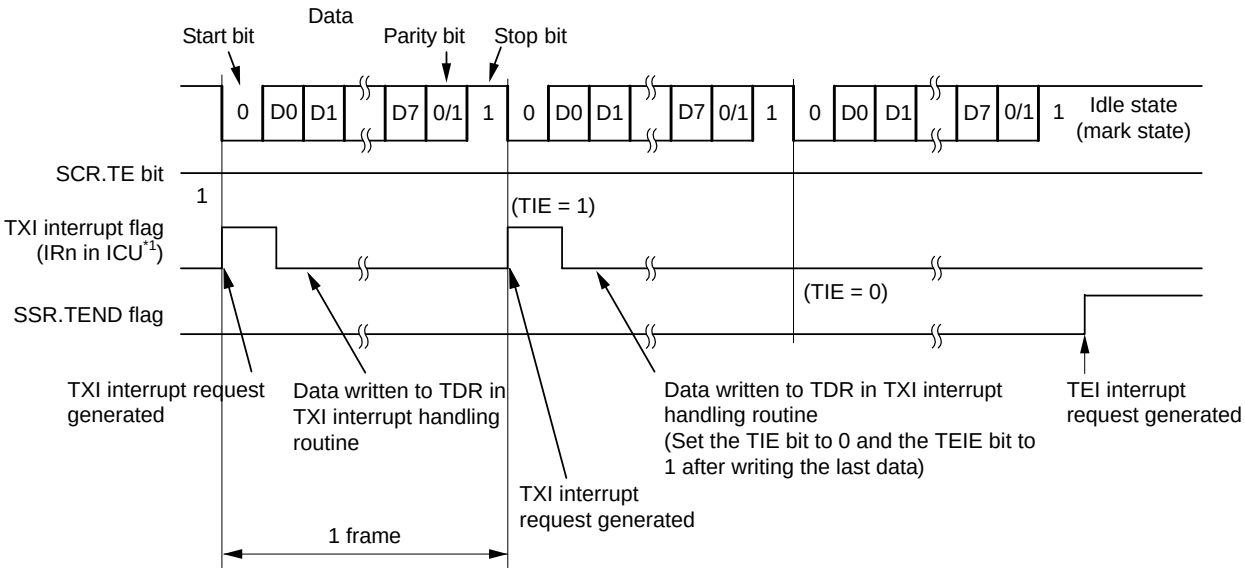
Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 23.9 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)



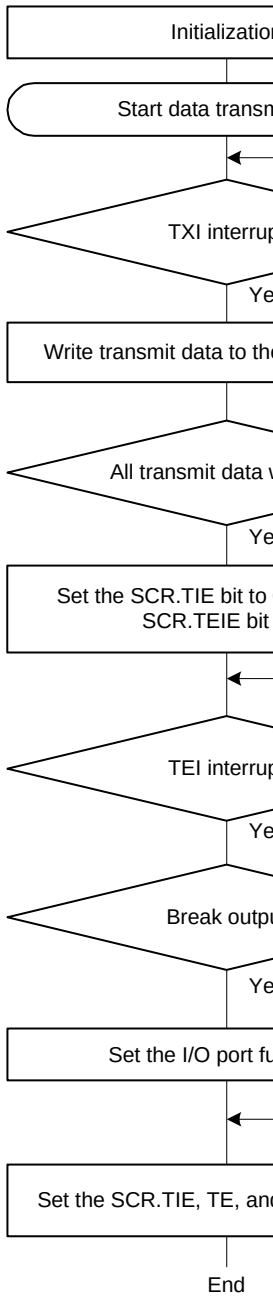
Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 23.10 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. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 23.11 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)



[1]

[2]

[3]

[4]

[1] SCI initialization:
Set data transmission.
After the SCR.TE bit is set to 1, high is output for a frame, and transmission is enabled.

[2] Transmit data write to the TDR register by a TXI interrupt request:
When transmit data is transferred from the TDR register to the TSR register, a transmit data empty interrupt (TXI) request is generated.
Write transmit data to the TDR register once in the TXI interrupt handling routine.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to the TDR register once using a TXI interrupt request.
Transmit data can also be written to the TDR register by activating the DTC.
When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR register.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the I/O port function corresponding to the TXDn pin (corresponding to output of the low level), and after switching the TXDn pin to the general I/O port function, set the SCR.TE bit to 0.

Note: The TDR register becomes the TDRH and TDRL registers when 9-bit data length is selected. Write data in the order from the TDRH register to the TDRL register.

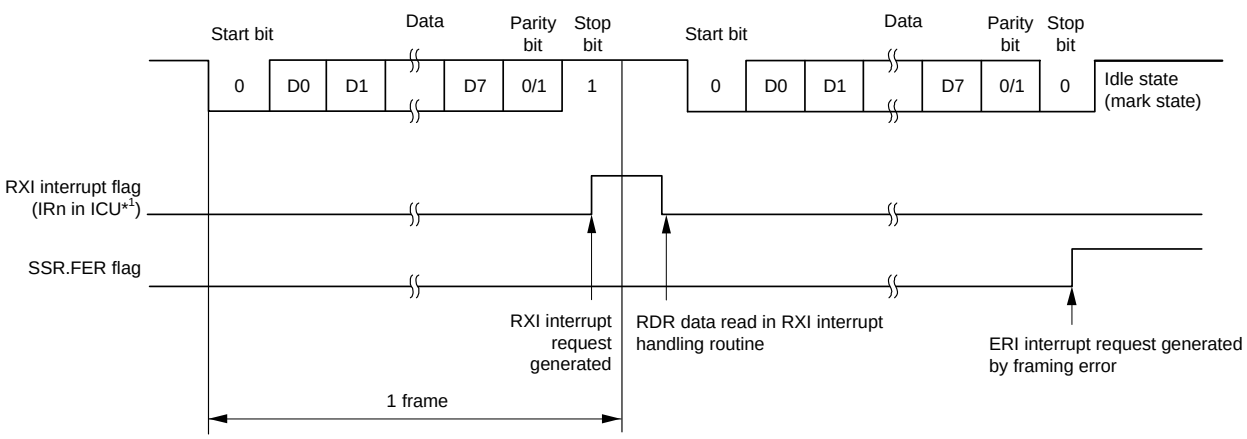
Figure 23.12 Example of Serial Transmission Flowchart in Asynchronous Mode

23.3.8 Serial Data Reception (Asynchronous Mode)

Figure 23.13 and Figure 23.14 show an example of the operation for serial data reception in asynchronous mode. In serial data reception, the SCI operates as described below.

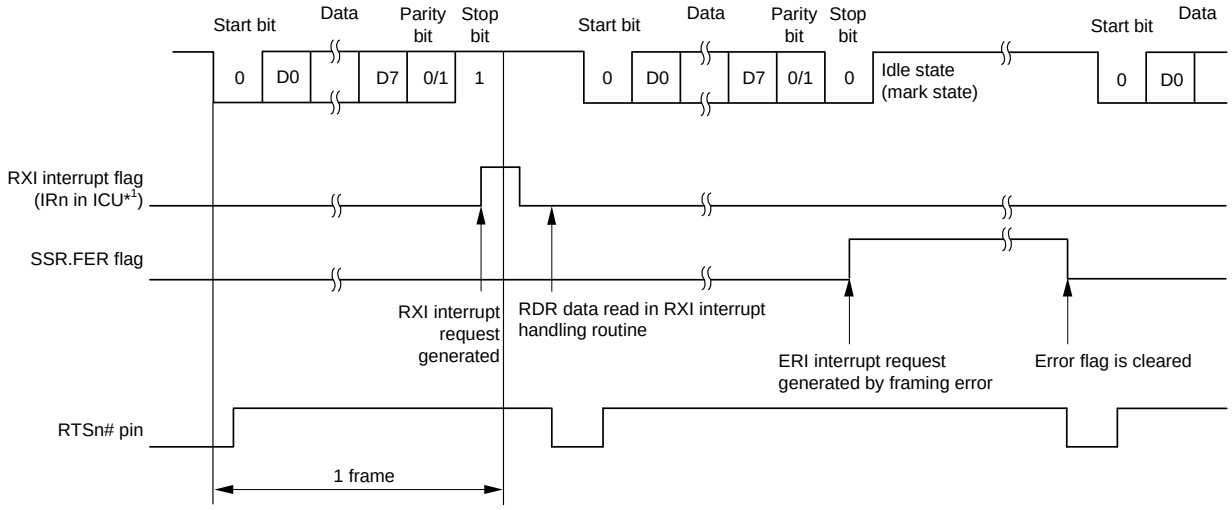
1. When the value of the SCR.RE bit becomes 1, the output signal on the RTSn# pin goes to the low level.
2. When the SCI monitors the communications line and detects a start bit, it performs internal synchronization, stores receive data in the RSR register, and checks the parity bit and stop bit.
3. If an overrun error occurs, the SSR.ORER flag is set to 1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to the RDR register*1.
4. If a parity error is detected, the SSR.PER flag is set to 1 and receive data is transferred to the RDR register*1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated.
5. If a framing error (when the stop bit is 0) is detected, the SSR.FER flag is set to 1 and receive data is transferred to the RDR register*1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to the RDR register*1. If the SCR.RIE bit is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register*1 in this RXI interrupt handling routine before reception of the next receive data is completed. Reading the received data that have been transferred to the RDR register*1 causes the RTSn# pin to output the low level.

Note 1. Read data not in the RDR register but in the RDRH and RDRL registers when 9-bit data length is selected.
Note 2. The SCI checks for reading of the RDRL register only and does not check for reading of the RDRH register when 9-bit data length is selected.



Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 23.13 Example of SCI Operation for Serial Reception in Asynchronous Mode (1) (When RTS Function is Not Used) (Example with 8-Bit Data, Parity, 1 Stop Bit)



Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 23.14 Example of SCI Operation for Serial Reception in Asynchronous Mode (2) (When RTS Function is Used) (Example with 8-Bit Data, Parity, 1 Stop Bit)

Table 23.28 lists the states of the status flags in the SSR register and receive data handling when a receive error is detected.

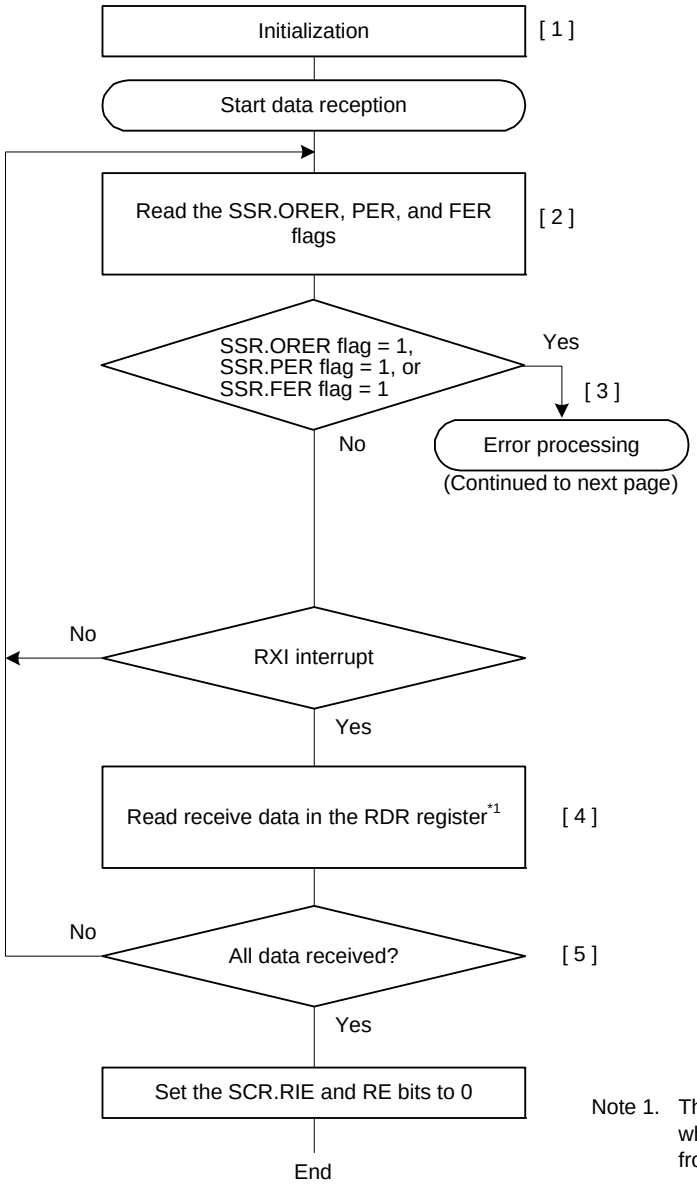
If a receive error is detected, an ERI interrupt request is generated but an RXI interrupt request is not generated. Data reception cannot be resumed while the receive error flag is 1. Accordingly, set the ORER, FER, and PER flags to 0 before resuming reception. Moreover, be sure to read the RDR (or the RDRL) register during overrun error processing. When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR (or the RDRL) register because received data which has not yet been read may be left in the RDR (or the RDRL) register.

Figure 23.15 and Figure 23.16 show samples of flowcharts for serial data reception.

Table 23.28 Status Flags in the SSR Register and Receive Data Handling

Status Flags in the SSR Register				
ORER	**FER**	**PER**	**Receive Data**	**Receive Error Type**
1	0	0	Lost	Overrun error
0	1	0	Transferred to the RDR register*1	Framing error
0	0	1	Transferred to the RDR register*1	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to the RDR register*1	Framing error + parity error
1	1	1	Lost	Overrun error + framing error + parity error

Note 1. Read data not in the RDR register but in the RDRH and RDRL registers when 9-bit data length is selected.



[1] SCI initialization:
Set data reception.

[2] [3] Receive error processing and break detection:
If a receive error occurs, an ERI interrupt is generated. An error is identified by reading the ORER, PER, and FER flags in the SSR register. 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.

[4] Read the receive data in the RDR register once in the RXI interrupt handling routine.

[5] Serial reception continuation procedure:
To continue serial reception, before the stop bit of the current frame is received, read data from the RDR register in the RXI interrupt handling routine.
The RDR data can also be read by activating the DTC.

Note 1. The RDR register becomes the RDRH and RDRL registers when 9-bit data length is selected. Read data in the order from the RDRH register to the RDRL register.

Figure 23.15 Example Flowchart of Serial Reception in Asynchronous Mode (1)

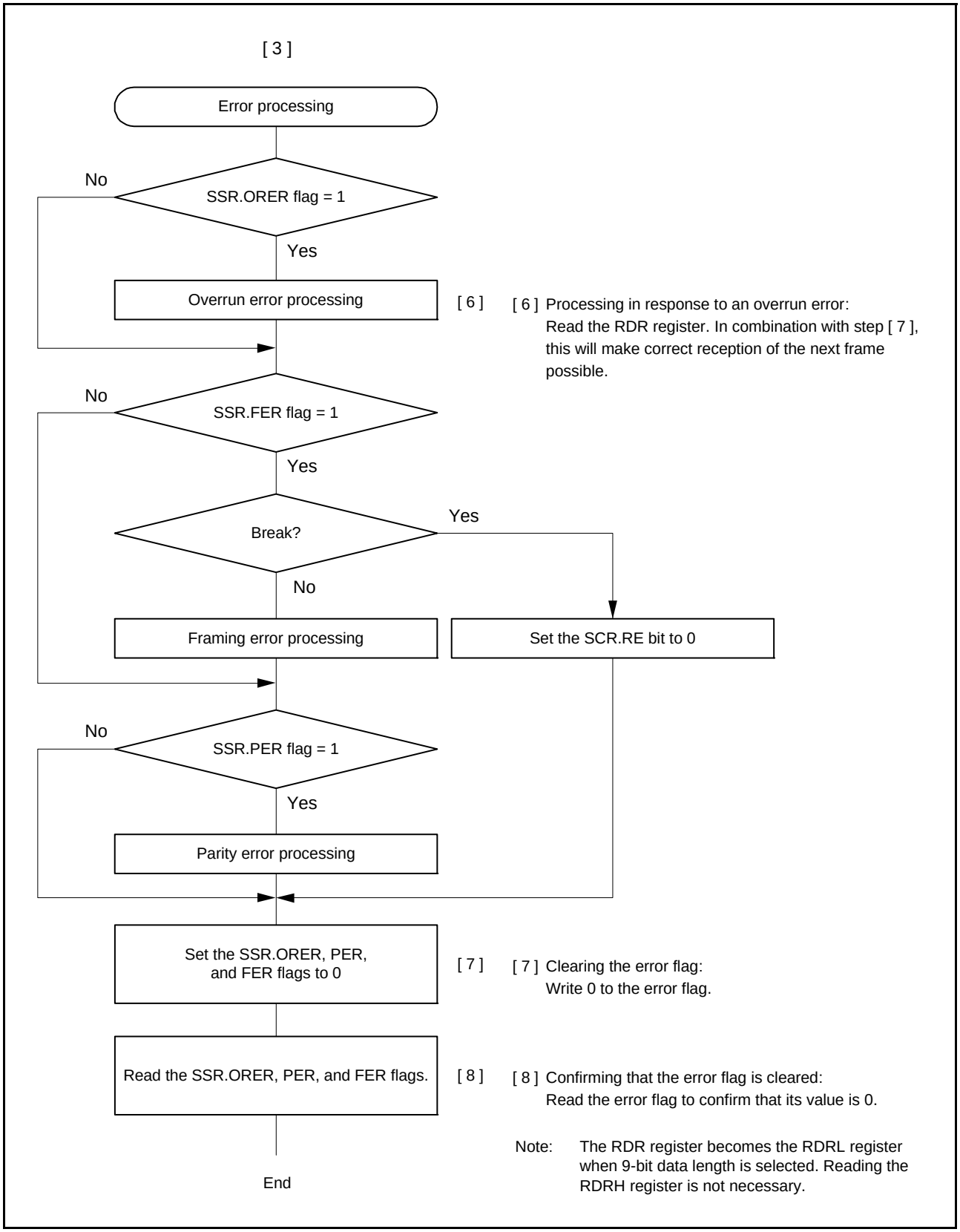


Figure 23.16 Example Flowchart of Serial Reception in Asynchronous Mode (2)

23.4 Multi-Processor Communications 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 multi-processor bit is set to 0, it indicates the data transmission cycle. Figure 23.17 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. Upon 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 match, 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.

For supporting this function, the SCI provides the SCR.MPIE bit. When the MPIE bit is set to 1, transfer of receive data from the RSR register to the RDR register (the RDRH and RDRL registers when 9-bit data length is selected), detection of a receive error, and setting the respective status flags RDRF, ORER, and FER in the SSR register are disabled until reception of data in which the multi-processor bit is set to 1. Upon receiving a reception character in which the multi-processor bit is set to 1, the SSR.MPB bit is set to 1 and the SCR.MPIE bit is automatically cleared, thus returning to a normal reception operation. During this time, an RXI interrupt is generated if the SCR.RIE bit is 1.

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 normal asynchronous mode. A clock which is used for the multi-processor communication is also the same as the clock used in the normal asynchronous mode.

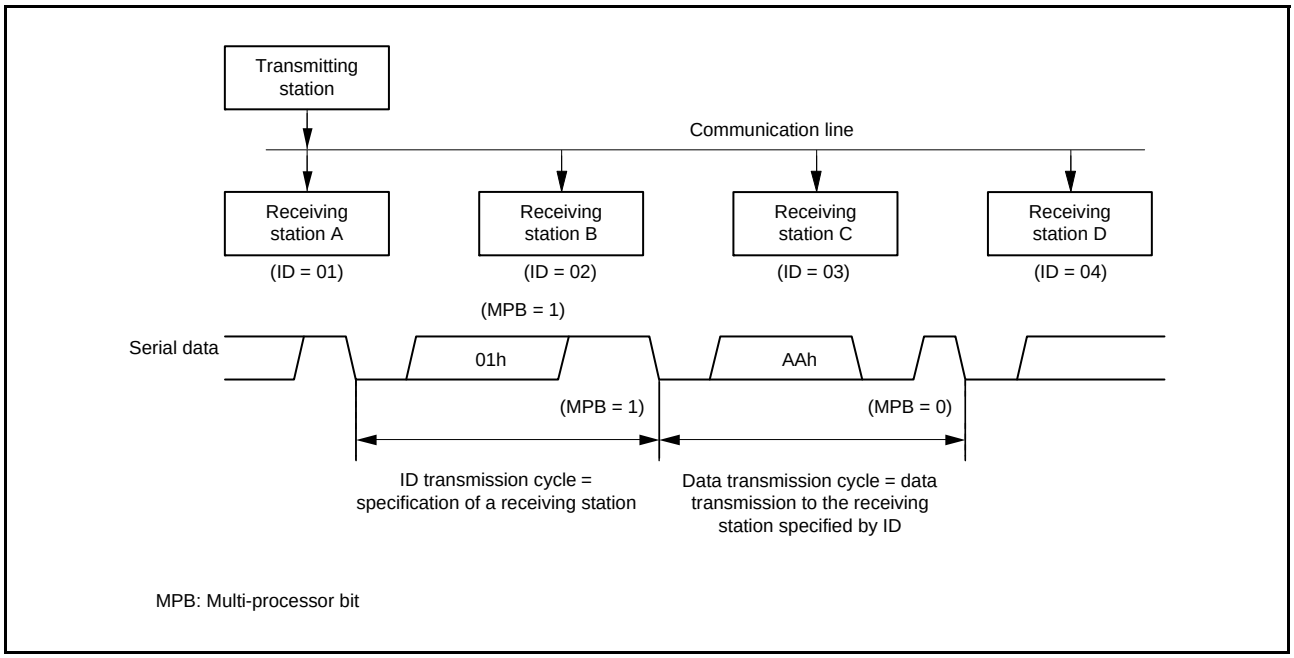


Figure 23.17 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)

23.4.1 Multi-Processor Serial Data Transmission

Figure 23.18 is a sample flowchart of multi-processor data transmission. In the ID transmission cycle, the ID should be transmitted with the SSR.MPBT bit 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.

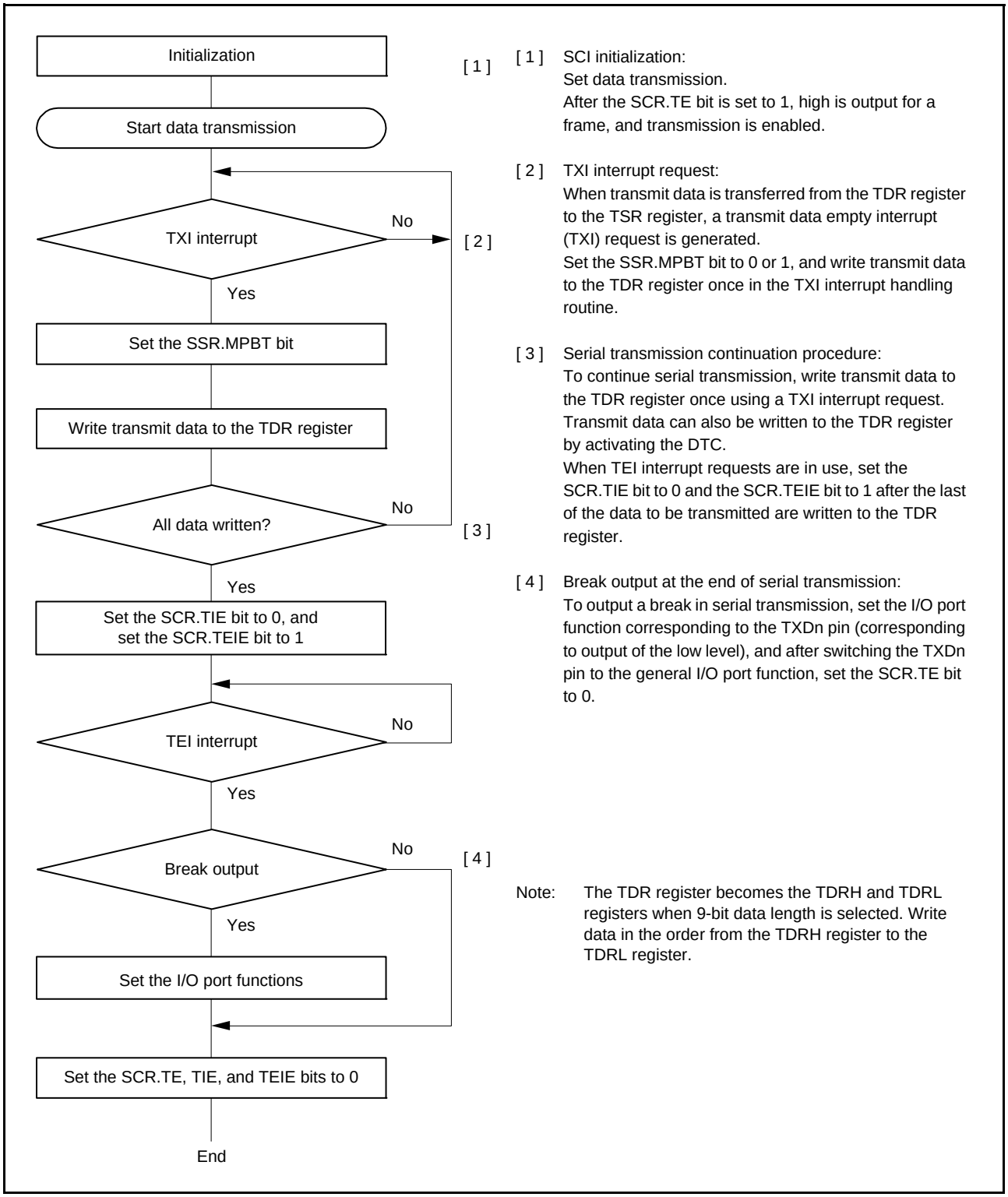
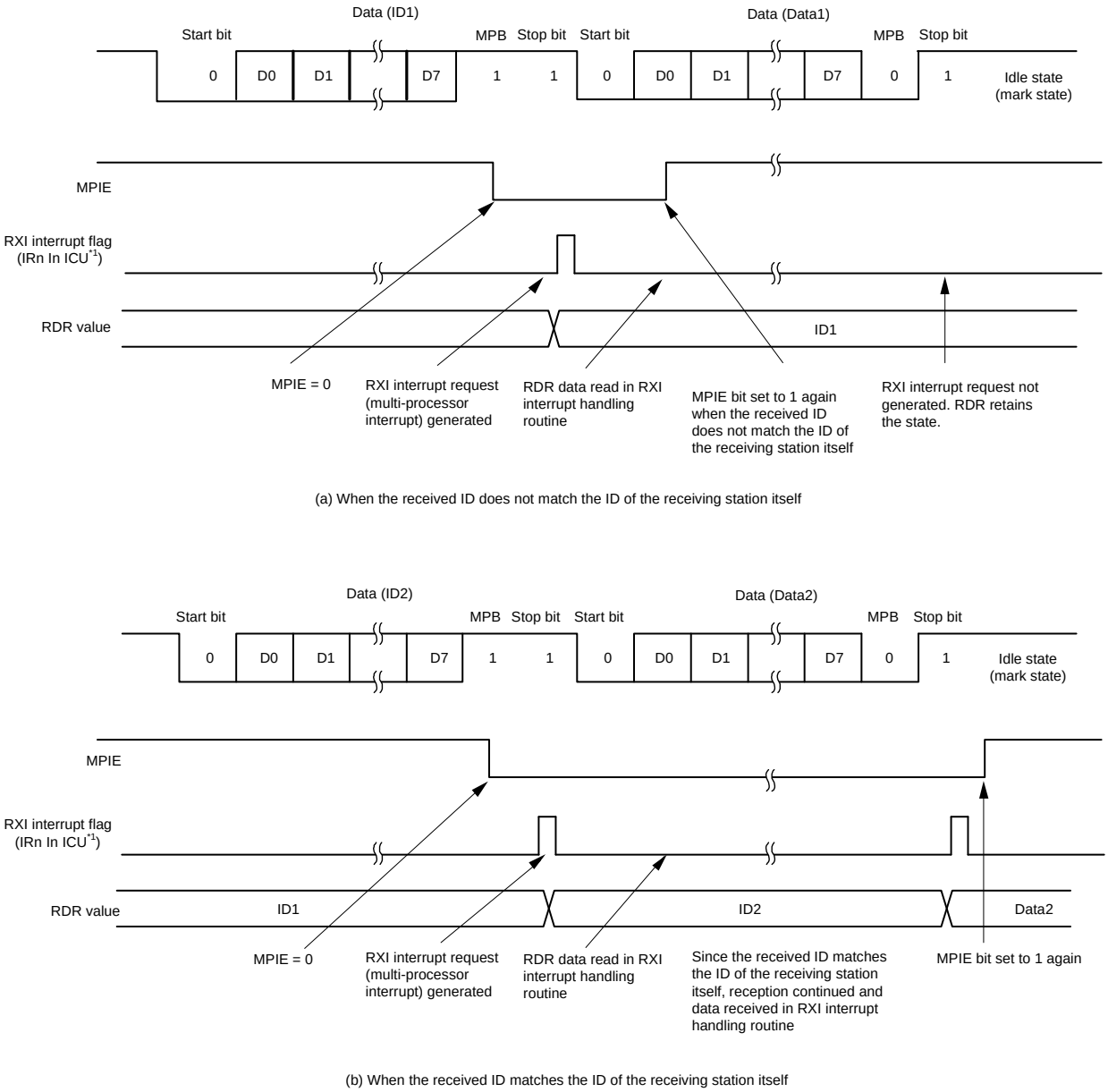


Figure 23.18 Example of Multi-Processor Serial Transmission Flowchart

23.4.2 Multi-Processor Serial Data Reception

Figure 23.20 and Figure 23.21 are sample flowcharts of multi-processor data reception. When the SCR.MPIE bit 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 the RDR register (the RDRH and RDRL registers when 9-bit data length is selected). During this time, the RXI interrupt request is generated. The other operations are the same as the operations in asynchronous mode. Figure 23.19 is the example of operation for reception.



Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 23.19 Example of SCI Reception (8-Bit Data/Multi-Processor Bit/1 Stop Bit)

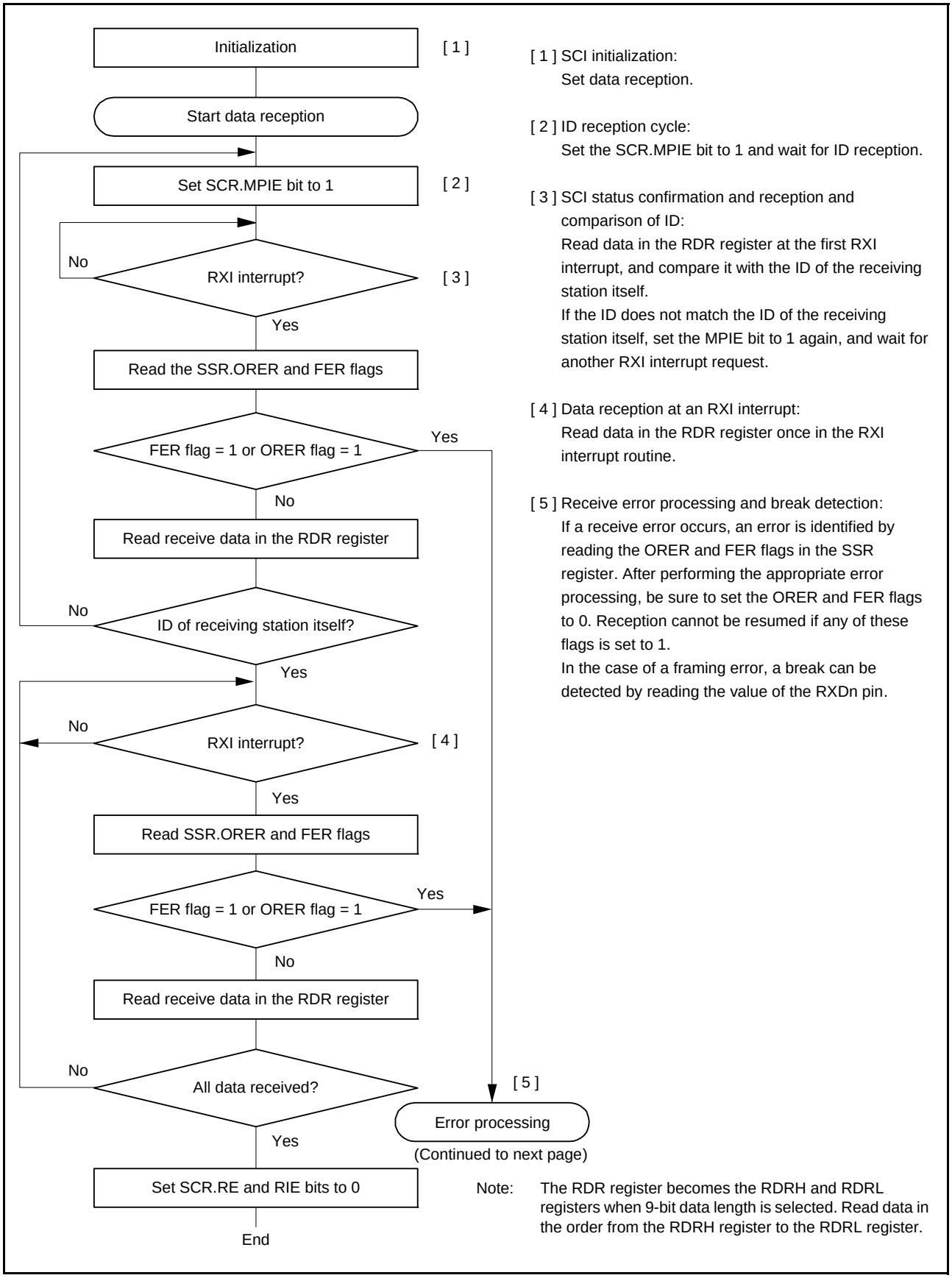


Figure 23.20 Example of Multi-Processor Serial Reception Flowchart (1)

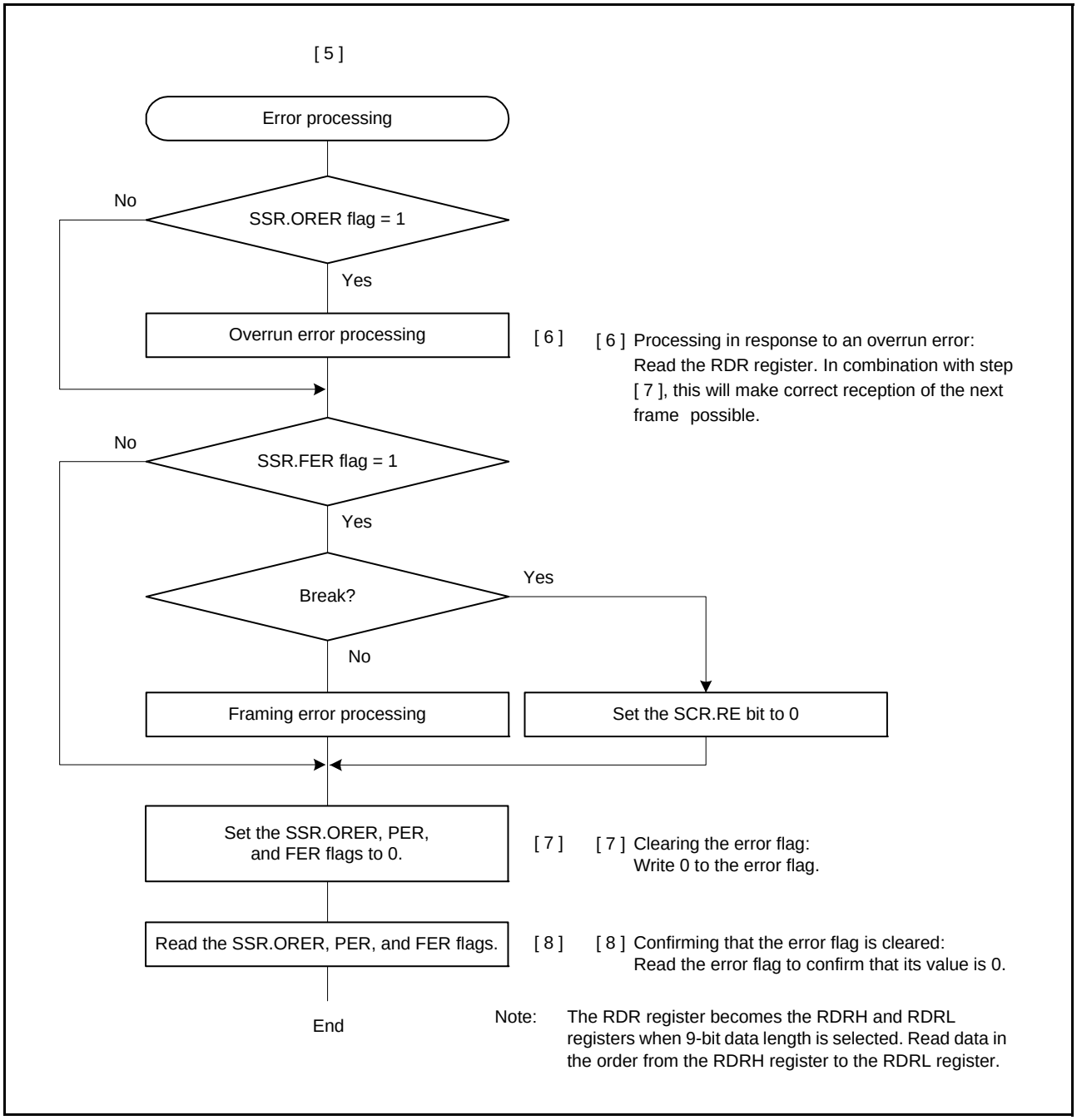


Figure 23.21 Example of Multi-Processor Serial Reception Flowchart (2)

23.5 Operation in Clock Synchronous Mode

Figure 23.22 shows the data format for clock synchronous serial data communications.
In clock synchronous mode, data is transmitted or received in synchronization with clock pulses. One character in transfer data consists of 8-bit data. In clock synchronous mode, no parity bit can be added.
In data transmission, the SCI outputs data from one falling edge of the synchronization clock to the next. In data reception, the SCI receives data in synchronization with the rising edge of the synchronization clock. After 8-bit data is output, the communication line holds the last bit output state.
Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communications by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that the next transmit data can be written during transmission or the previous receive data can be read during reception, enabling continuous data transfer.

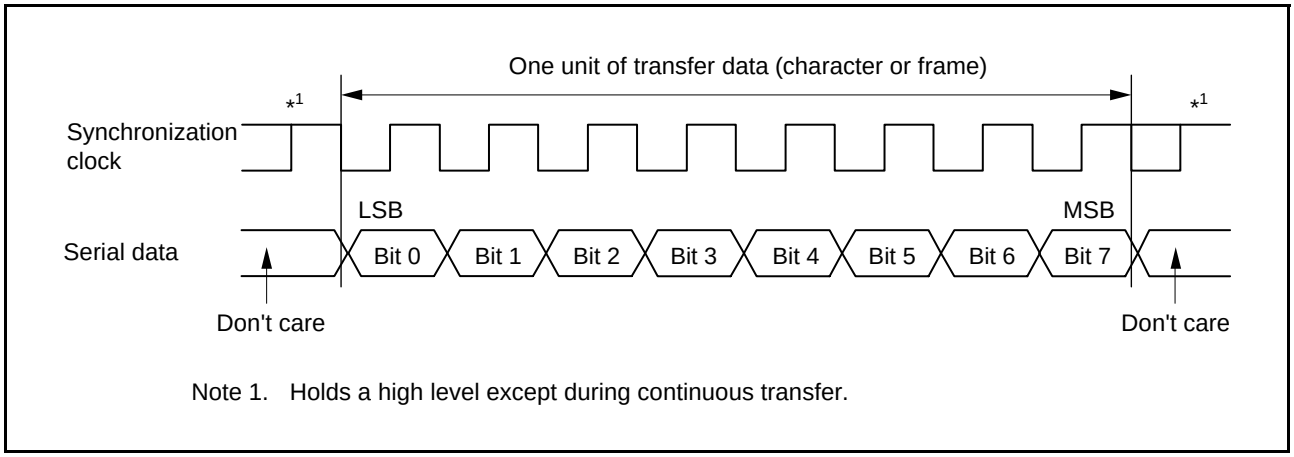


Figure 23.22 Data Format in Clock Synchronous Serial Communications (LSB First)

23.5.1 Clock

Either an internal clock generated by the on-chip baud rate generator or an external synchronization clock input at the SCKn pin can be selected, according to the setting of the SCR.CKE[1:0] bits.
When the SCI is operated on an internal clock, the synchronization clock is output from the SCKn pin. Eight synchronization clock pulses are output in the transfer of one character, and when no transfer is performed the clock is held high. However, when only data reception is performed while the CTS function is disabled, the synchronization clock output is started at the same time when the SCR.RE bit set to 1. The synchronization clock is stopped at the high level when an overrun error occurs or the SCR.RE bit is set to 0.
When only data reception is performed and the CTS function is enabled, the clock output is not started even when the SCR.RE bit set to 1 if the CTSn# pin input is high when the SCR.RE bit is 0. The synchronization clock output is started when the SCR.RE bit is set to 1 and the CTSn# pin input is low. After that, if the CTSn# pin input is high on completion of the frame reception, the synchronization clock output is stopped at the high level. If the CTSn# pin input continues to be low, the synchronization clock is stopped at the high level when an overrun error occurs or the SCR.RE bit is set to 0.

23.5.2 CTS and RTS Functions

In the CTS function, CTSn# pin input is used to control reception/transmission start when the clock source is the internal clock. Setting the SPMR.CTSE bit to 1 enables the CTS function.
When the CTS function is enabled, placing the low level on the CTSn# pin causes reception/transmission to start. Applying the high level to the CTS# pin while reception/transmission are in progress does not affect reception/transmission of the current frame, which continues.
In the RTS function, RTSn# pin output is used to request reception/transmission start when the clock source is an external synchronizing clock. A low level is output when serial communications become possible. Conditions for output of the low and high level are shown below.

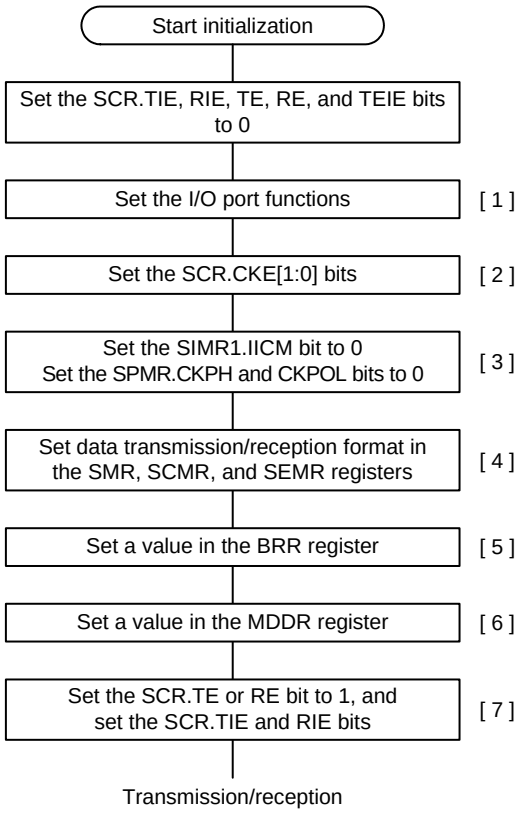
[Conditions for low-level output]
When the following conditions are all satisfied.

- The SCR.RE or SCR.TE bit is 1.
- Transmission or reception is not in progress.
- There are no received data yet to be read (when the SCR.RE bit is 1).
- Untransmitted data is present (when the SCR.TE bit is 1).
- The SSR.ORER flag is 0.

[Condition for high-level output]
The conditions for low-level output have not been satisfied.

23.5.3 SCI Initialization (Clock Synchronous Mode)

Before transmitting and receiving data, start by writing the initial value 00h to the SCR register and then continue through the procedure for SCI given in Figure 23.23. Whenever the operating mode or transfer format is changed, the SCR register must be initialized before the change is made.
Note that setting the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in the SSR register nor the RDR register.



[1] Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.

[2] Set the clock selection in the SCR register.
When an internal clock is selected, the SCK pin functions as the clock output pin.

[3] Set the SIMR1.IICM bit to 0.
Set the SPMR.CKPH and CKPOL bits to 0.
Step [3] can be skipped if the values have not been changed from the initial values.

[4] Set the data transmission/reception format in the SMR, SCMR, and SEMR registers.

[5] Write a value corresponding to the bit rate to the BRR register. This step is not necessary if an external clock is used.

[6] Write the value obtained by correcting a bit rate error in the MDDR register. This step is not necessary if the SEMR.BRME bit is set to 0 or an external clock is used.

[7] Set the SCR.TE or RE bit to 1. At this time, also set the SCR.TIE and RIE bits.
Setting the TE and RE bits allows TXDn and RXDn to be used.

Note: In simultaneous transmit and receive operations, the SCR.TE and RE bits should both be set to 0 or 1 simultaneously.

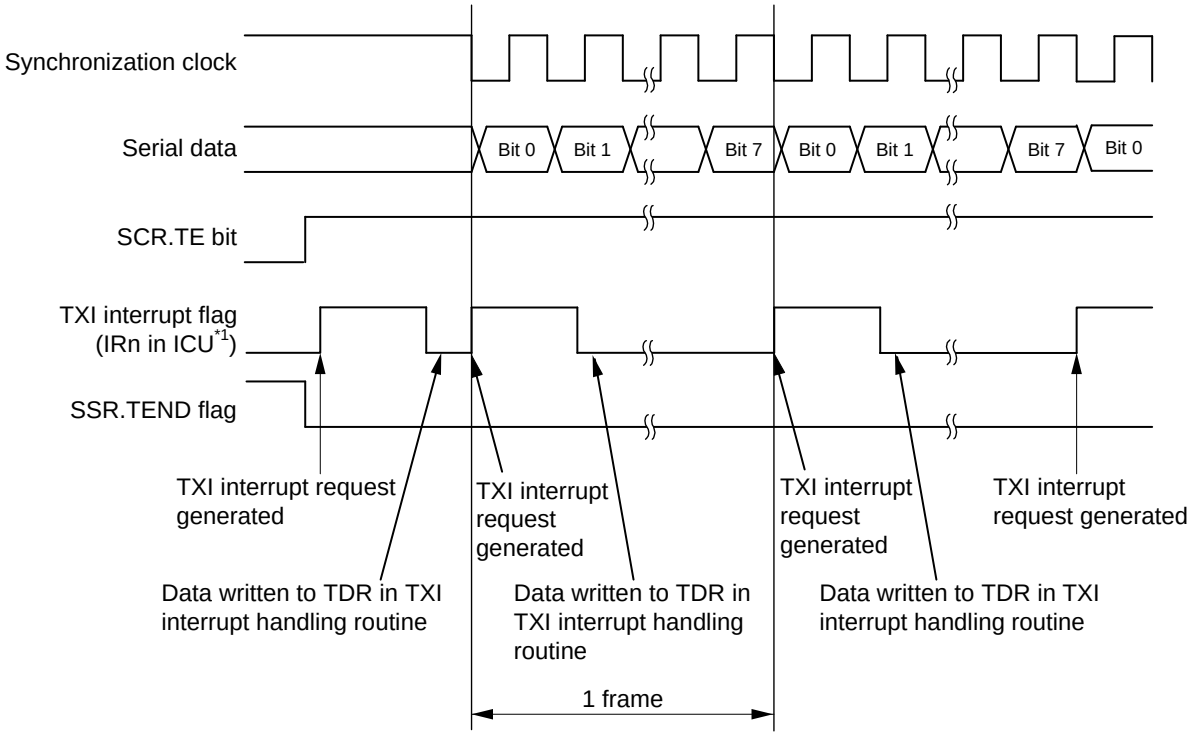
Figure 23.23 Example of SCI Initialization Flowchart (Clock Synchronous Mode)

23.5.4 Serial Data Transmission (Clock Synchronous Mode)

Figure 23.24, Figure 23.25, and Figure 23.26 show an example of the operation for serial transmission in clock synchronous mode.
In serial data transmission, the SCI operates as described below.

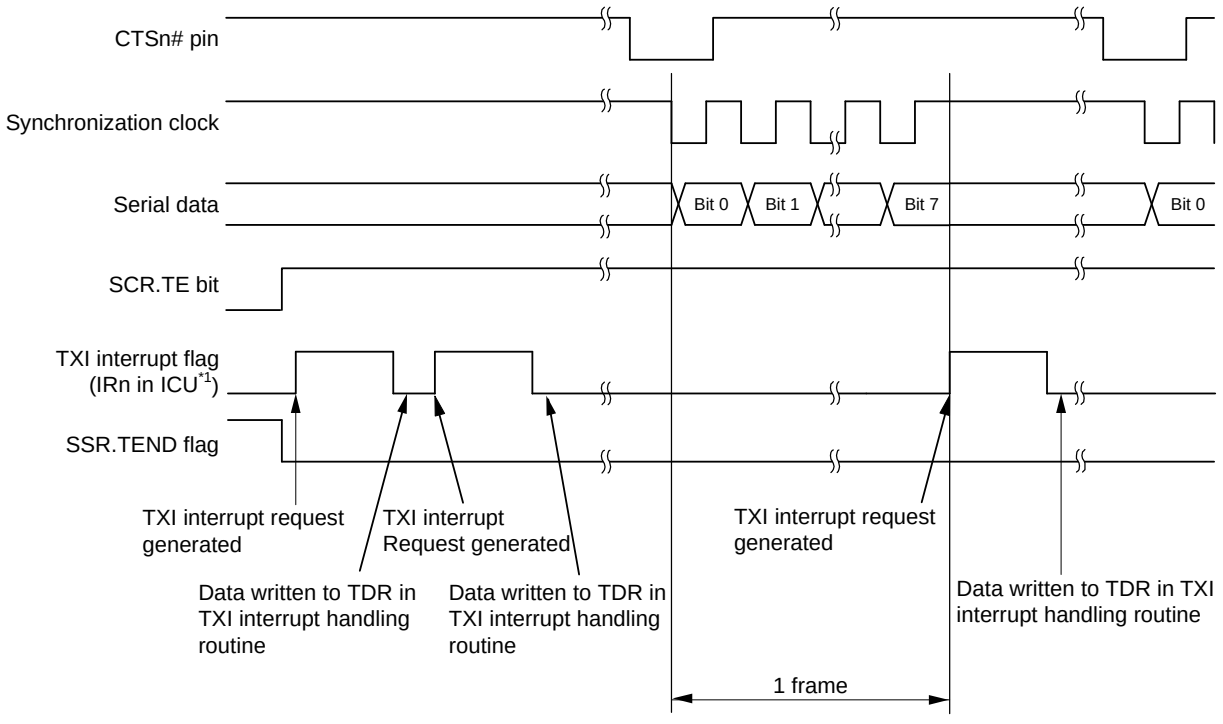
1. The SCI transfers data from the TDR register to the TSR register when data is written to the TDR register in the TXI interrupt handling routine. The TXI interrupt request at the beginning of transmission is generated when the SCR.TE bit is set to 1 after the SCR.TIE bit is set to 1 or when these 2 bits are set to 1 simultaneously by a single instruction.
2. After transferring data from the TDR register to the TSR register, the SCI starts transmission. When the SCR.TIE bit is set to 1 at this time, a TXI interrupt request is generated. Continuous transmission is enabled by writing the next transmit data to the TDR register in this TXI interrupt handling routine before transmission of the current transmit data has finished. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (a TXI interrupt request is disabled) and the SCR.TEIE bit to 1 (a TEI interrupt request is enabled) after the last of the data to be transmitted are written to the TDR register from the handling routine for TXI requests.
3. 8-bit data is sent from the TXDn pin in synchronization with the output clock when clock output mode has been specified and in synchronization with the input clock when use of an external clock has been specified. Output of the clock signal is suspended until the input CTS signal is at the low level while the SPMR.CTSE bit is 1 (CTS function is enabled).
4. The SCI checks for updating of (writing to) the TDR register at the time of the last bit output.
5. When TDR is updated, the next transmit data is transferred from the TDR register to the TSR register, and serial transmission of the next frame is started.
6. If the TDR register is not updated, set the SSR.TEND flag to 1 and the TXDn pin retains the output state of the last bit. If the SCR.TEIE bit is 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Figure 23.27 shows a sample flowchart of serial data transmission.
Transmission will not start while a receive error flag (ORER, FER, or PER in the SSR register) is set to 1. Be sure to set the receive error flags to 0 before starting transmission. Note that setting the SCR.RE bit to 0 does not clear the receive error flags.



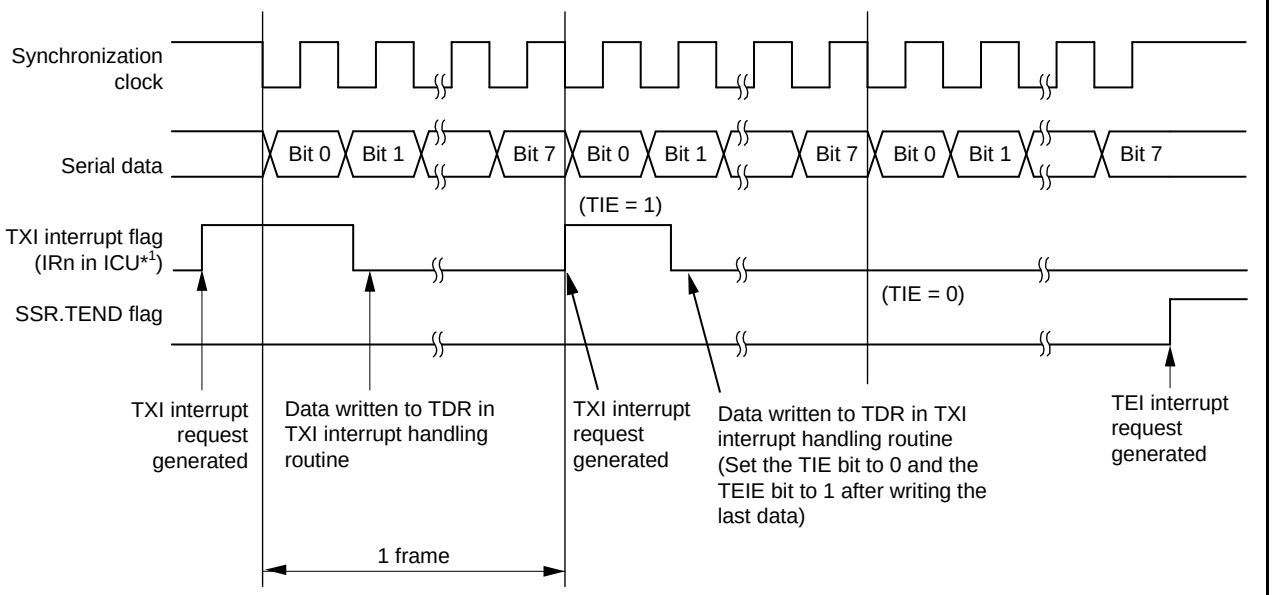
Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 23.24 Example of Serial Data Transmission in Clock Synchronous Mode When the CTS Function is Not Used at the Beginning of Transmission



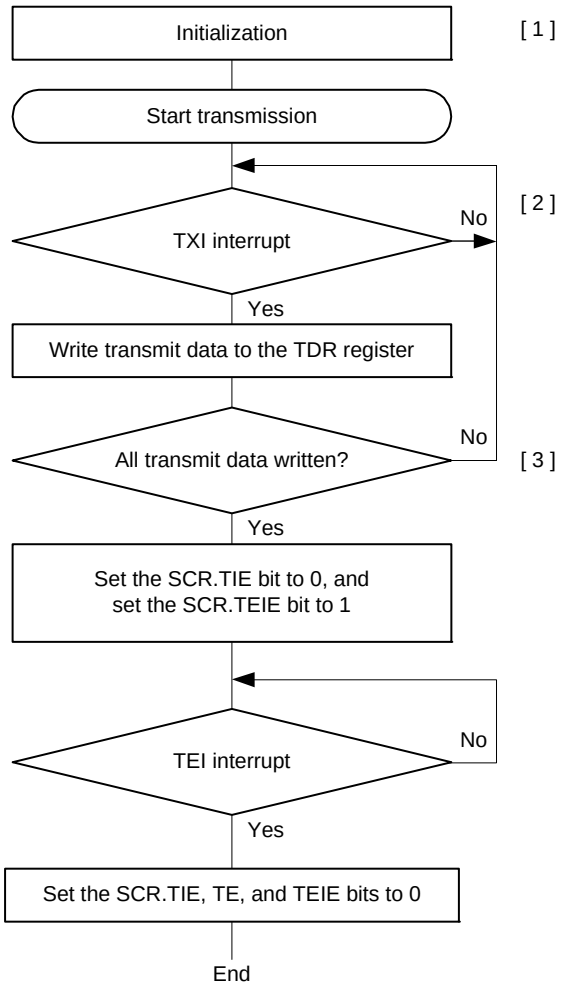
Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 23.25 Example of Serial Data Transmission in Clock Synchronous Mode When the CTS Function is Used at the Beginning of Transmission



Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 23.26 Example of Serial Data Transmission in Clock Synchronous Mode from the Middle of Transmission until Transmission Completion



[1] SCI initialization:
Set data transmission.

[2] Writing transmit data write to the TDR register by a TXI interrupt request:
When transmit data is transferred from the TDR register to the TSR register, a transmit data empty interrupt (TXI) request is generated.
Transmit data is written to the TDR register once from the handling routine for TXI requests.

[3] Serial transmission continuation procedure:
To continue serial transmission, write transmit data to the TDR register upon accepting a transmit data empty interrupt (TXI) request. Transmit data can also be written to the TDR register by activating the DTC by the TXI interrupt request.
When TEI interrupt requests are in use, set the SCR.TIE bit to 0 and the SCR.TEIE bit to 1 after the last of the data to be transmitted are written to the TDR register.

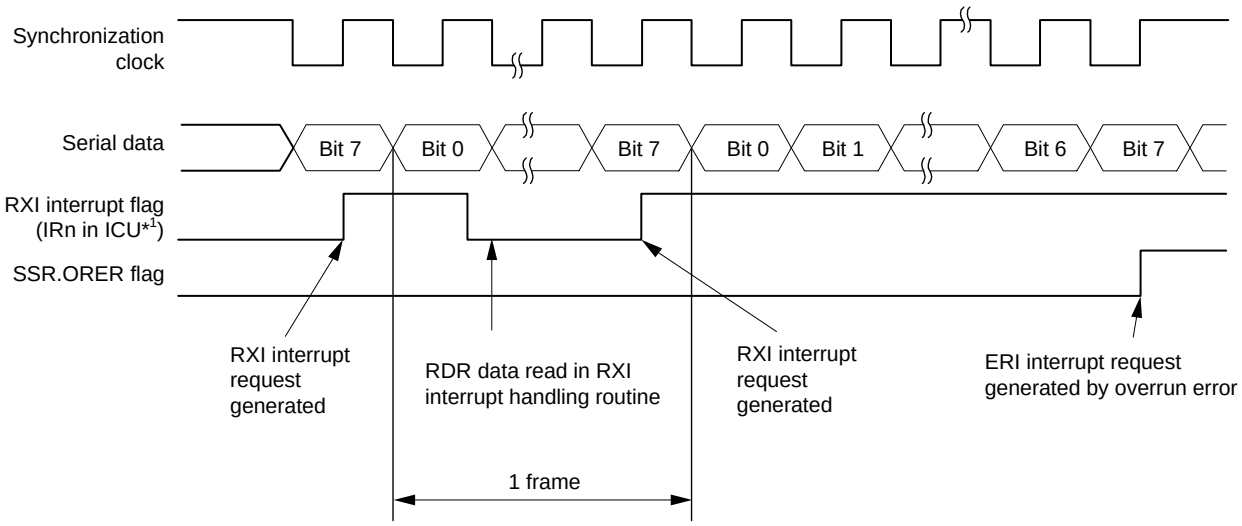
Note: When the external clock is in use (the value of the SCR.CKE[1:0] bits is 10b or 11b), the rising edge on the SCK pin for the last bit sets the SSR.TEND flag to 1. Setting the SCR.TE bit to 0 immediately after this may lead to insufficient received-data hold time on the receiver side.

Figure 23.27 Example Flowchart of Serial Transmission in Clock Synchronous Mode

23.5.5 Serial Data Reception (Clock Synchronous Mode)

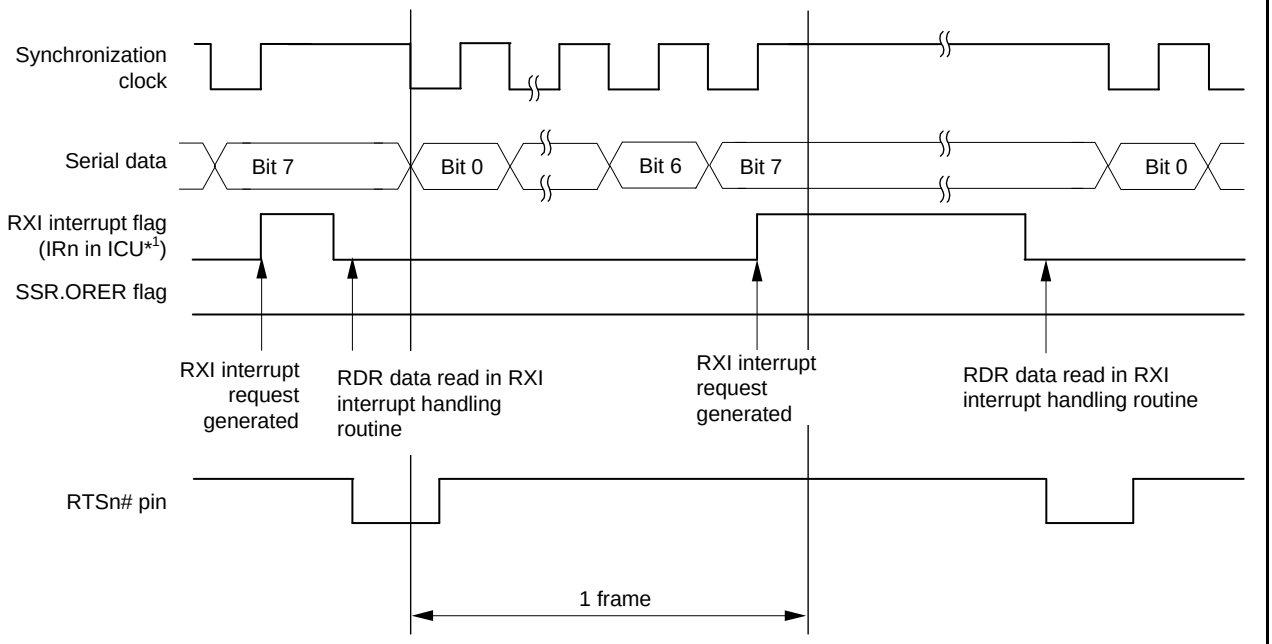
Figure 23.28 and Figure 23.29 show an example of SCI operation for serial reception in clock synchronous mode. In serial data reception, the SCI operates as described below.

1. The value of the SCR.RE bit becoming 1 places the signal output on the RTSn# pin at the low level (when the RTS function is in use).
2. The SCI performs internal initialization and starts receiving data in synchronization with a synchronization clock input or output, and stores the receive data in the RSR register.
3. If an overrun error occurs, the SSR.ORER flag is set to 1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to the RDR register.
4. When reception finishes successfully, receive data is transferred to the RDR register. If the SCR.RIE bit is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to the RDR register in this RXI interrupt handling routine before reception of the next receive data is completed. Reading out the received data that have been transferred to the RDR register causes the RTSn# pin to output the low level (when the RTS function is in use).



Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 23.28 Example of Operation for Serial Reception in Clock Synchronous Mode (1) (When RTS Function is Not Used)

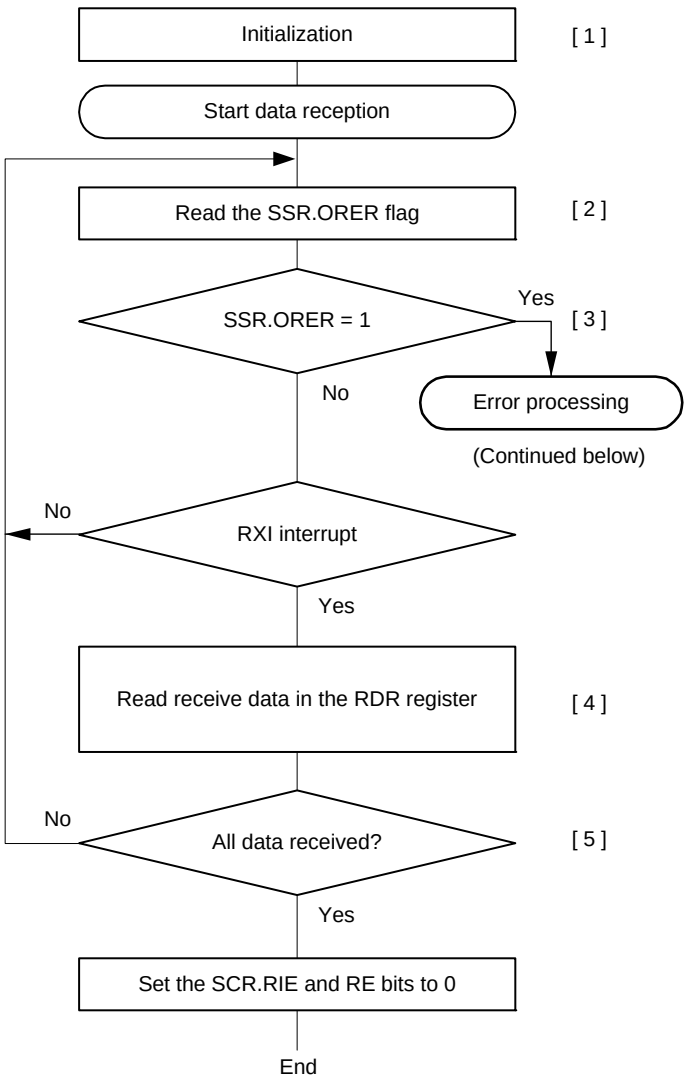


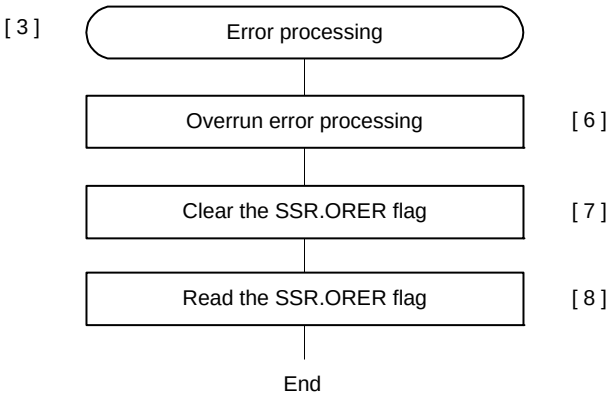
Note 1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 23.29 Example of Operation for Serial Reception in Clock Synchronous Mode (2) (When RTS Function is Used)

Data transfer cannot be resumed while a receive error flag is 1. Accordingly, clear the ORER, FER, and PER flags in the SSR register before resuming reception. Moreover, be sure to read the RDR register during overrun error processing. When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR register because received data which has not yet been read may be left in the RDR register.

Figure 23.30 shows a sample flowchart for serial data reception.





[1] SCI initialization:
Make input port-pin settings for pins to be used as RXDn pins.

[2] [3] Receive error processing:
If a receive error occurs, read the SSR.ORER flag, perform the relevant error processing, and then set the ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[4] Read the receive data in the RDR register once in the receive data full interrupt (RXI) request handling routine.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, finish reading the receive data in the RDR register. The RDR data can also be read by activating the DTC by an RXI interrupt request.

[6] Processing in response to an overrun error:
Read the RDR register. In combination with step [7], this will make correct reception of the next frame possible.

[7] Clearing the error flag:
Write 0 to the error flag.

[8] Confirming that the error flag is cleared:
Read the error flag to confirm that its value is 0.

Figure 23.30 Example Flowchart of Serial Reception in Clock Synchronous Mode

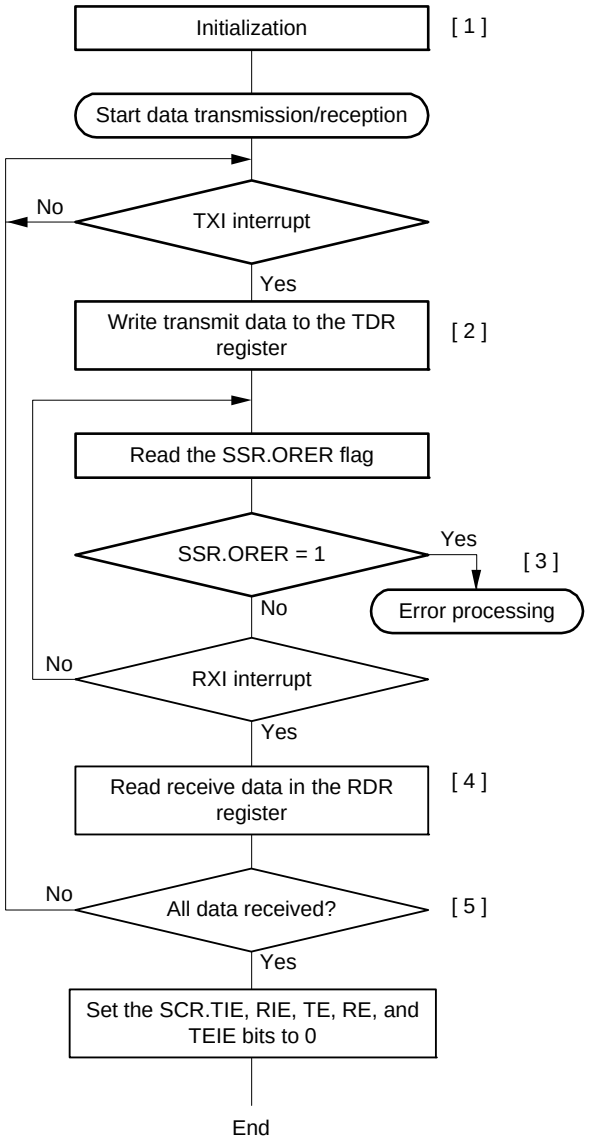
23.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

Figure 23.31 shows a sample flowchart for simultaneous serial transmit and receive operations in clock synchronous mode.

After initializing the SCI, the following procedure should be used for simultaneous serial data transmit and receive operations.

To switch from transmit mode to simultaneous transmit and receive mode, check that the SCI has finished transmission by reading that the SSR.TEND flag is 1, and then initialize the SCR register. Then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.

To switch from receive mode to simultaneous transmit and receive mode, check that the SCI has finished reception, and then set the RIE and RE bits to 0. Then check that the receive error flags (ORER, FER, and PER in the SSR register) are 0, and then set the TIE, RIE, TE, and RE bits in the SCR register to 1 simultaneously by a single instruction.



[1] SCI initialization:
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 the TDR register once in the TXI interrupt handling routine.

[3] Receive error processing:
If a receive error occurs, read the SSR.ORER flag, perform the relevant error processing, and then set the ORER flag to 0. Data reception cannot be resumed while the ORER flag is 1.

[4] Reading receive data:
Read the receive data in the RDR register once in the RXI interrupt request handling routine.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission and reception, before the MSB (bit 7) of the current frame is received, finish reading the receive data in the RDR register by the RXI interrupt. Also, before the MSB (bit 7) of the current frame is transmitted, write data to the TDR register by the TXI interrupt. Transmit data can also be written to the TDR register by activating the DTC by a transmit data empty interrupt (TXI) request. Similarly, the RDR data can also be read by activating the DTC by a receive data full interrupt (RXI) request.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations first set the TIE, RIE, TE, RE, and TEIE bits in the SCR register to 0, and then set TIE, RIE, TE, and RE bits to 1 simultaneously.

Figure 23.31 Example Flowchart of Simultaneous Serial Transmission and Reception in Clock Synchronous Mode

23.6 Operation in 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 of the SCI.
Smart card interface mode can be selected using the appropriate register.

23.6.1 Sample Connection

Figure 23.32 shows a sample connection between a smart card (IC card) and this MCU.
As in the figure, since this MCU 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 the SCR register 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.
The output port of the this MCU can be used to output a reset signal.

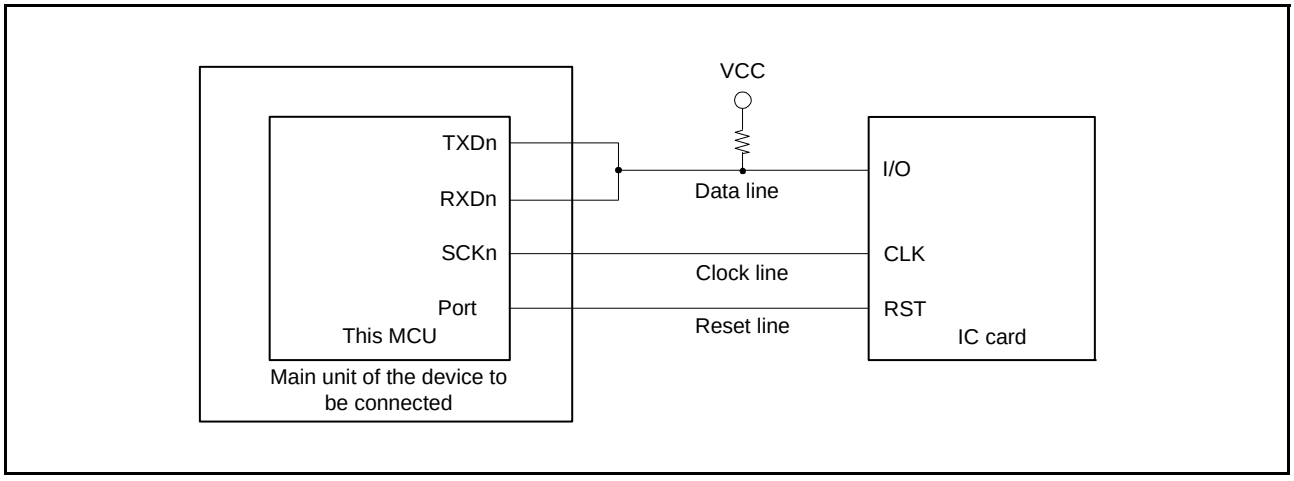


Figure 23.32 Sample Connection with a Smart Card (IC Card)

23.6.2 Data Format (Except in Block Transfer Mode)

Figure 23.33 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.

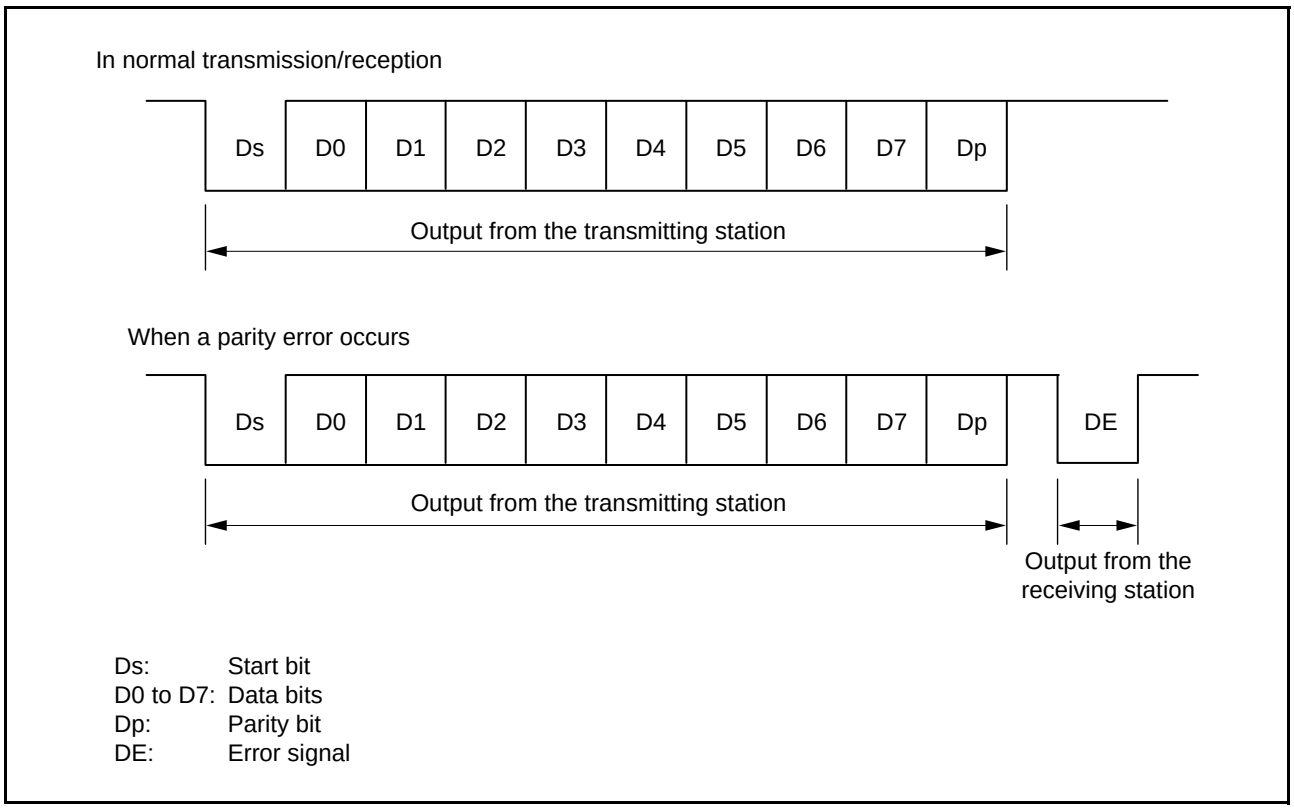


Figure 23.33 Data Formats in Smart Card Interface Mode

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 23.34. Therefore, data in the start character in the figure is 3Bh. When using the direct convention type, write 0 to both the SDIR and SINV bits in the SCMR register. Write 0 to the SMR.PM bit in order to use even parity, which is prescribed by the smart card standard.

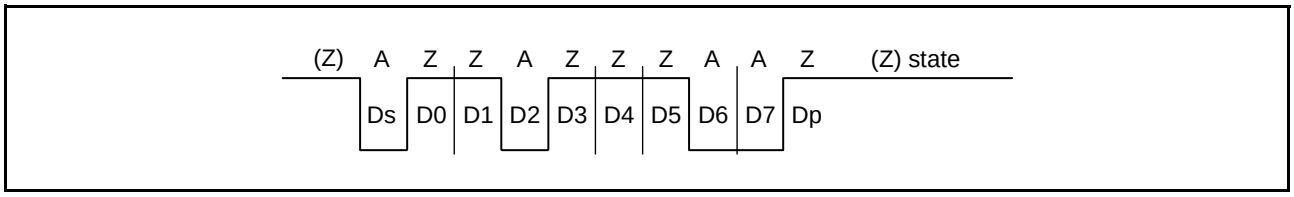


Figure 23.34 Direct Convention (SCMR.SDIR bit = 0, SCMR.SINV bit = 0, SMR.PM bit = 0)

(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 23.35. Therefore, data in the start character in the figure is 3Fh. When using the inverse convention type, write 1 to both the SDIR and SINV bits in the SCMR register. 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 the this MCU only inverts data bits D7 to D0, write 1 to the SMR.PM bit to invert the parity bit for both transmission and reception.

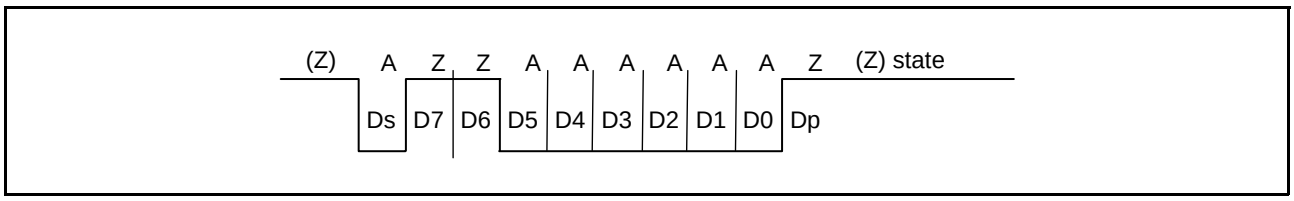


Figure 23.35 Inverse Convention (SCMR.SDIR bit = 1, SCMR.SINV bit = 1, SMR.PM bit = 1)

23.6.3 Block Transfer Mode

Block transfer mode is different from non-block transfer mode in the following respects.

- Even if a parity error is detected during reception, no error signal is output. Since the SSR.PER flag is set by error detection, clear the PER flag 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, the SSR.TEND flag is set 11.5 etu after transmission start.
- In block transfer mode, the SSR.ERS flag indicates the error signal status as in non-block transfer mode, but the flag is read as 0 because no error signal is transferred.

23.6.4 Receive Data Sampling Timing and Reception Margin

Only the base clock generated by the on-chip baud rate generator can be used as a transmit/receive 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 the SCMR.BCP2 bit and the SMR.BCP[1:0] bits.

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 23.36. The reception margin here is determined by the following formula.

$$M = \left| \left(0.5 - \frac{1}{2N}\right) - (L - 0.5)F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\ (\%)$$

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 × 372)} × 100 (%) = 49.866 (%)

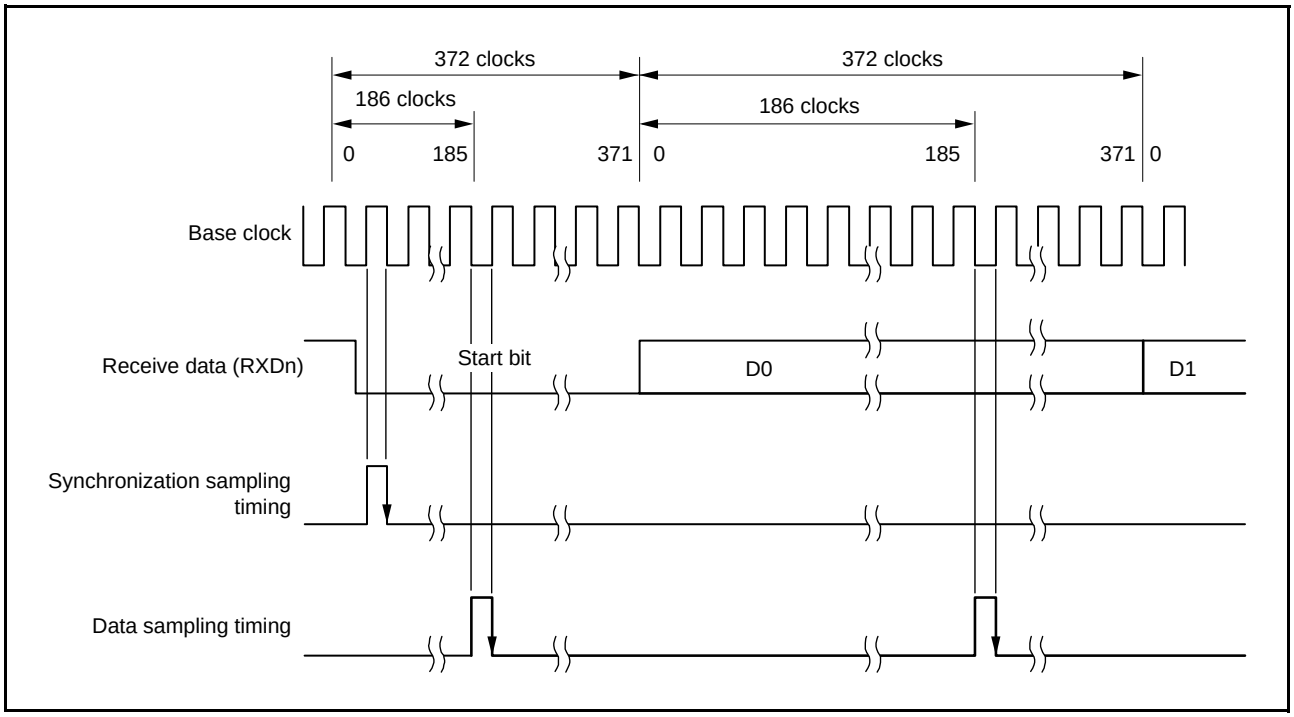


Figure 23.36 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)

23.6.5 SCI Initialization (Smart Card Interface Mode)

Initialize the SCI following the example of flowchart shown in Figure 23.37.
Initialize the SCR and SSR registers before switching from transmit mode to receive mode and vice versa. When not changing the bit rate, it is not necessary to set the CKE[1:0] bits to 00b. Even if the RE bit is set to 0, the RDR register is not initialized.

To change receive mode to transmit mode, first check that reception has completed, and then execute steps [1] and [3] in Figure 23.37. Set TE = 1 and RE = 0 at step [11]. Reception completion can be verified by reading the RXI request, SSR.ORER, or SSR.PER flag.
To change transmit mode to receive mode, first check that transmission has completed, and then execute steps [1] and [3] in Figure 23.37. Set TE = 0 and RE = 1 at step [11]. Transmission completion can be verified by reading the SSR.TEND flag.

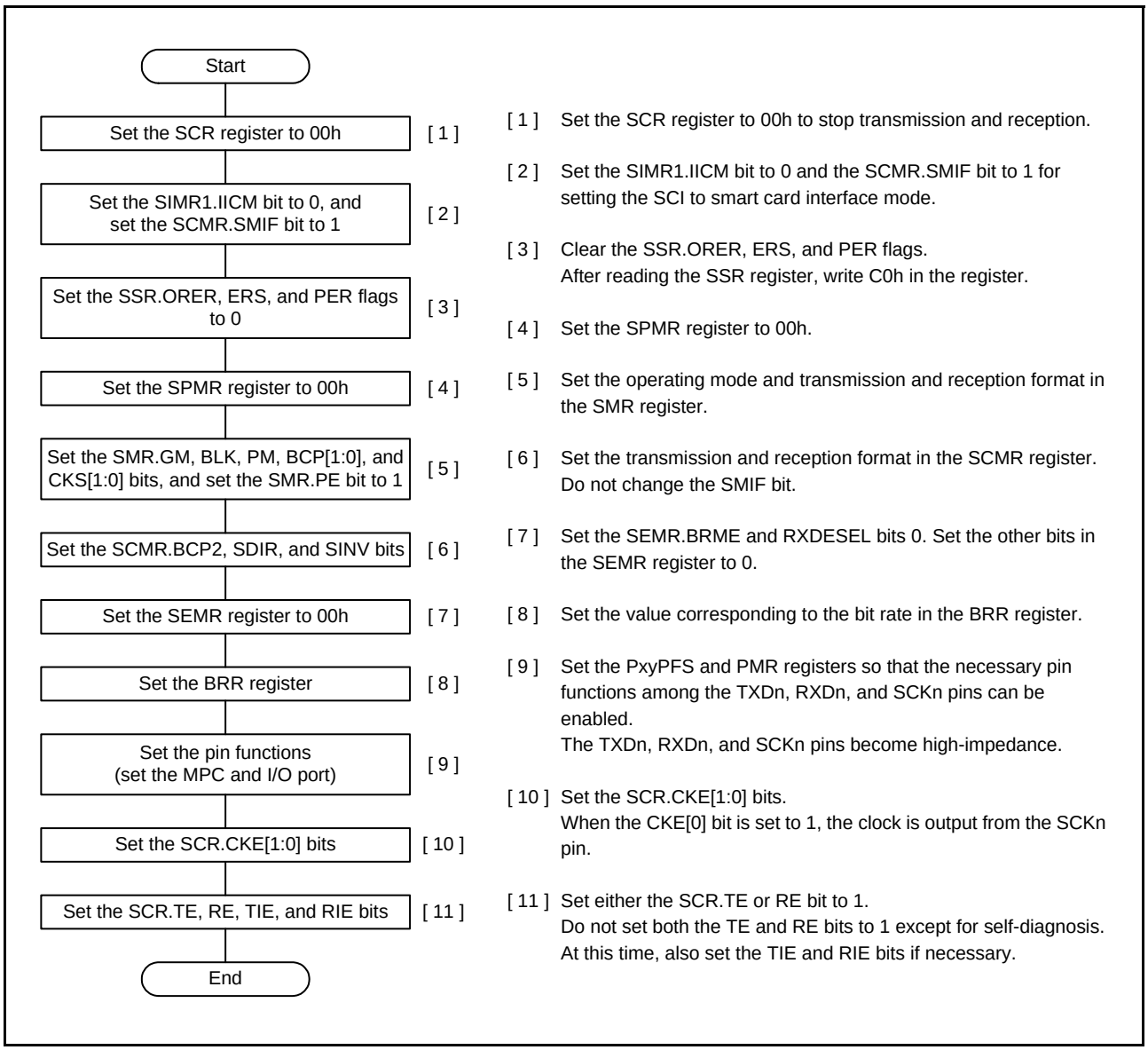


Figure 23.37 Example of SCI Initialization Flowchart (Smart Card Interface Mode)

Figure 23.38 shows an example of data transmission when the SCI is set to smart card interface mode according to the flow described in Figure 23.37 after a reset. When the pin functions are set to the SCK and TXD pins, they are still high-impedance because the SCR.CKE[0] and SCR.TE bits are 0. When the CKE[0] bit is set to 1, clock is output from the SCK pin. When the transmit data is written after setting the TE bit to 1, a data transmission starts. After the TE bit is set to 1, one frame of high-impedance is output from TXD pin (internal wait time) and then the data transmission starts. In smart card interface mode, the clock is continuously output while the CKE[0] bit is set to 1 (clock output) even if both the TE and RE bits are set to 0.

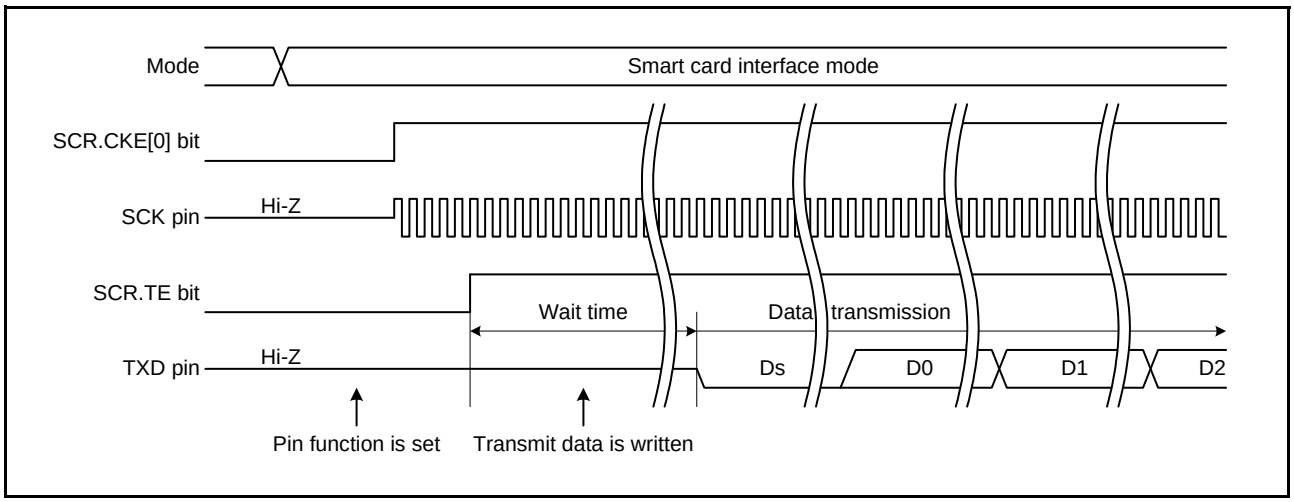


Figure 23.38 Example of Data Transmission Timing in Smart Card Interface Mode

23.6.6 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 23.39 shows the data retransmit operation during transmission.

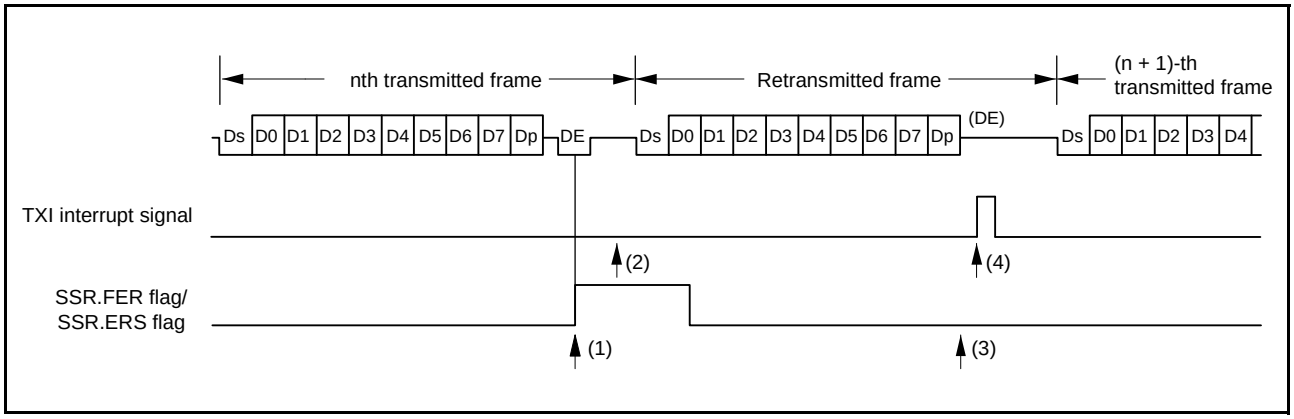


Figure 23.39 Data Retransmit Operation in SCI Transmit Mode

(1) When an error signal from the receiver end is sampled after one-frame data has been transmitted, the SSR.ERS flag is set to 1. If the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated. Clear the ERS flag before the next parity bit is sampled.

(2) For a frame in which an error signal is received, the SSR.TEND flag is not set. Data is retransferred from the TDR register to the TSR register allowing automatic data retransmission.

(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 retransmission) has been completed, and the TEND flag is set. If the SCR.TIE bit is 1 at this time, a TXI interrupt request is generated. Writing transmit data to the TDR register starts transmission of the next data.

Figure 23.40 shows a sample flowchart of serial transmission.

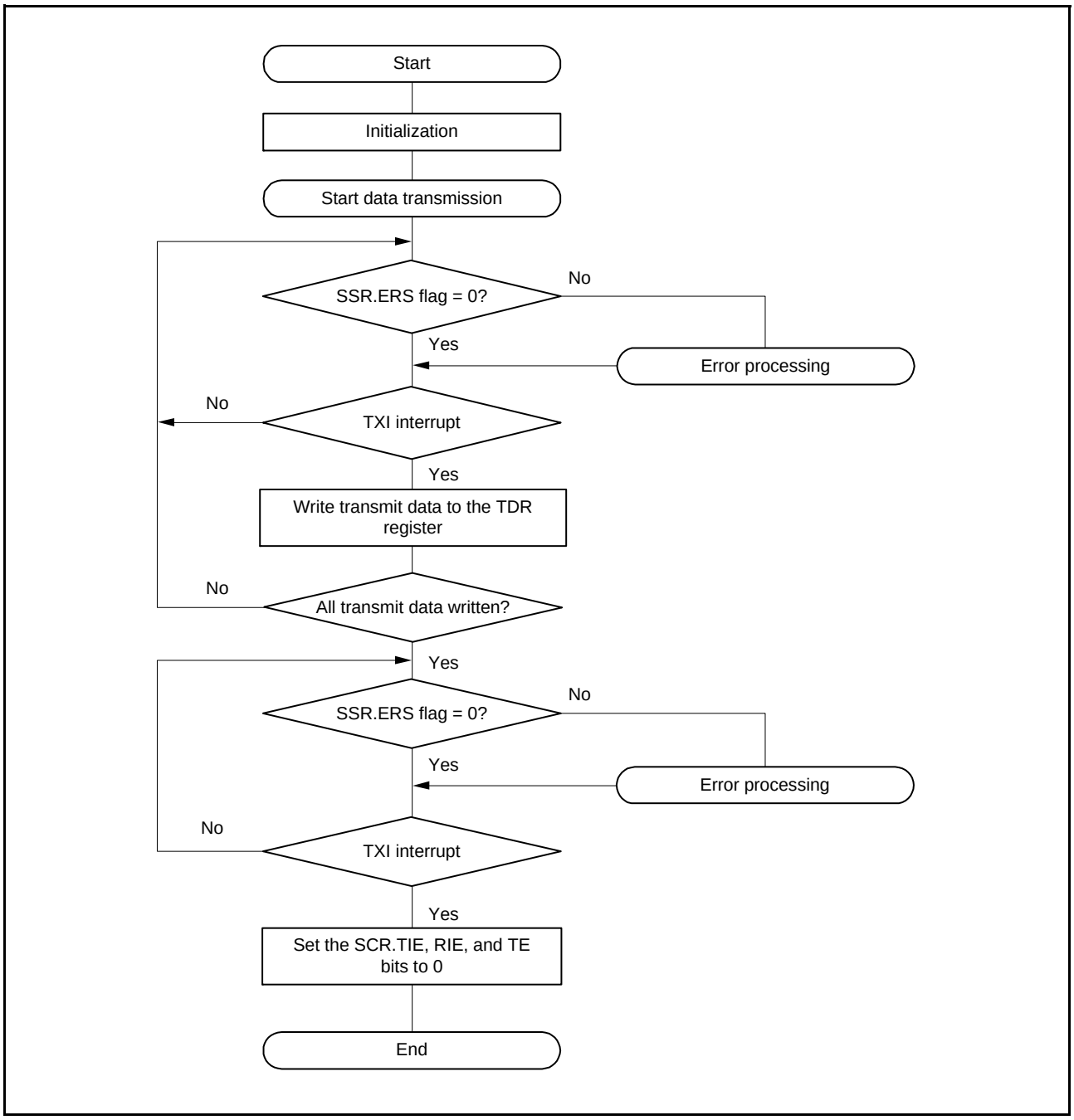


Figure 23.40 Sample Smart Card Interface Transmission Flowchart

All the processing steps are automatically performed using a TXI interrupt request to activate the DTC.
When the SSR.TEND flag is set to 1 in transmission, if the SCR.TIE bit is 1, a TXI interrupt request is generated. The DTC is activated by a TXI interrupt request if the TXI interrupt request is specified as a source of DTC activation beforehand, allowing transfer of transmit data. The TEND flag is automatically set to 0 when the DTC 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 DTC is not activated. Therefore, the SCI and DTC 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.

When transmitting/receiving data using the DTC, be sure to make settings to enable the DTC before making SCI settings. For DTC settings, refer to section 16, Data Transfer Controller (DTCb).

Note that the SSR.TEND flag is set in different timings depending on the SMR.GM bit setting. Figure 23.41 shows the TEND flag generation timing.

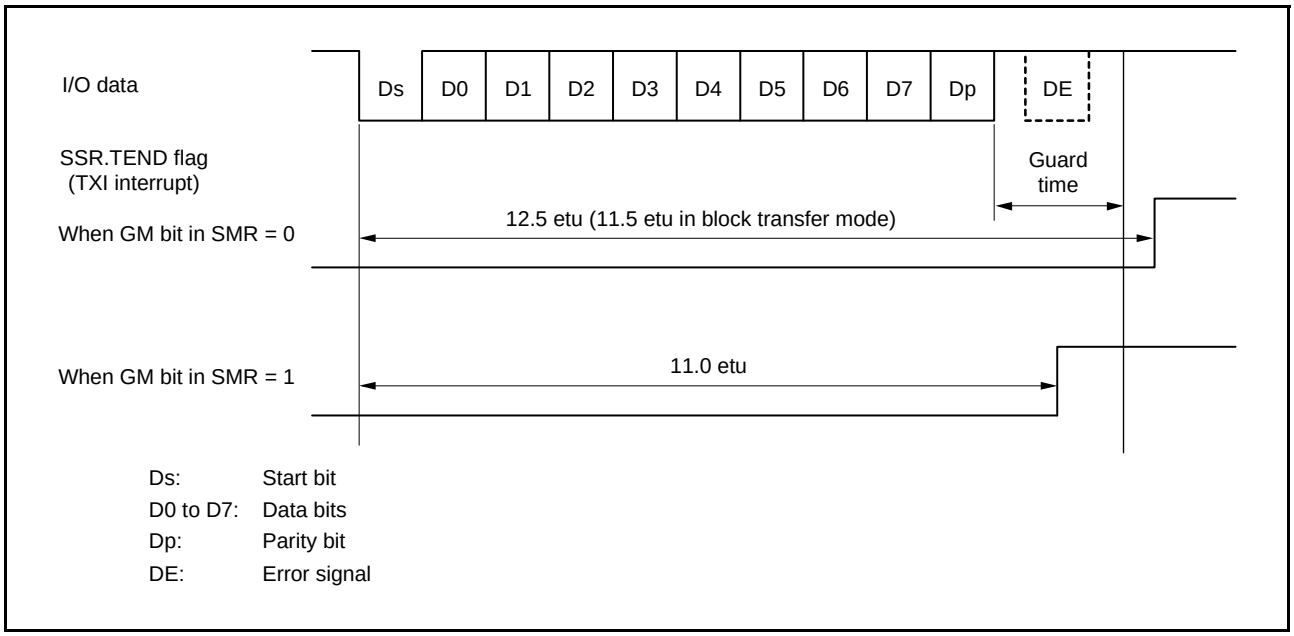


Figure 23.41 SSR.TEND Flag Generation Timing during Transmission

23.6.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in non-smart card interface mode. Figure 23.42 shows the data retransmit operation in receive mode.

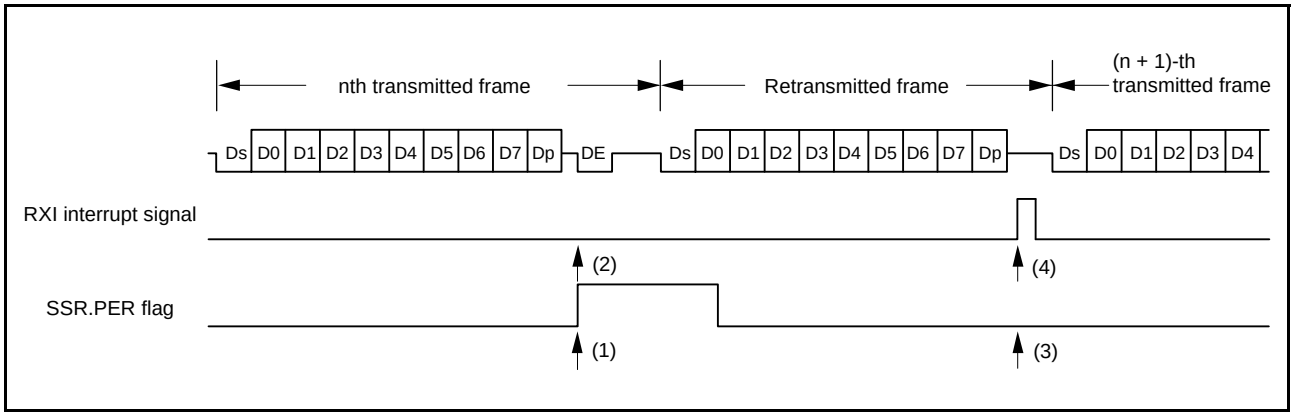


Figure 23.42 Data Retransmit Operation in SCI Receive Mode (Data Retransmit Operation during Reception)

(1) If a parity error is detected in receive data, the SSR.PER flag is set to 1. When the SCR.RIE bit is 1 at this time, an ERI interrupt request is generated. Clear the PER flag 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 SSR.PER flag is not set to 1.
(4) In this case, data is determined to have been received successfully. When the SCR.RIE bit is 1, an RXI interrupt request is generated.

Figure 23.43 shows a sample flowchart for serial data reception.

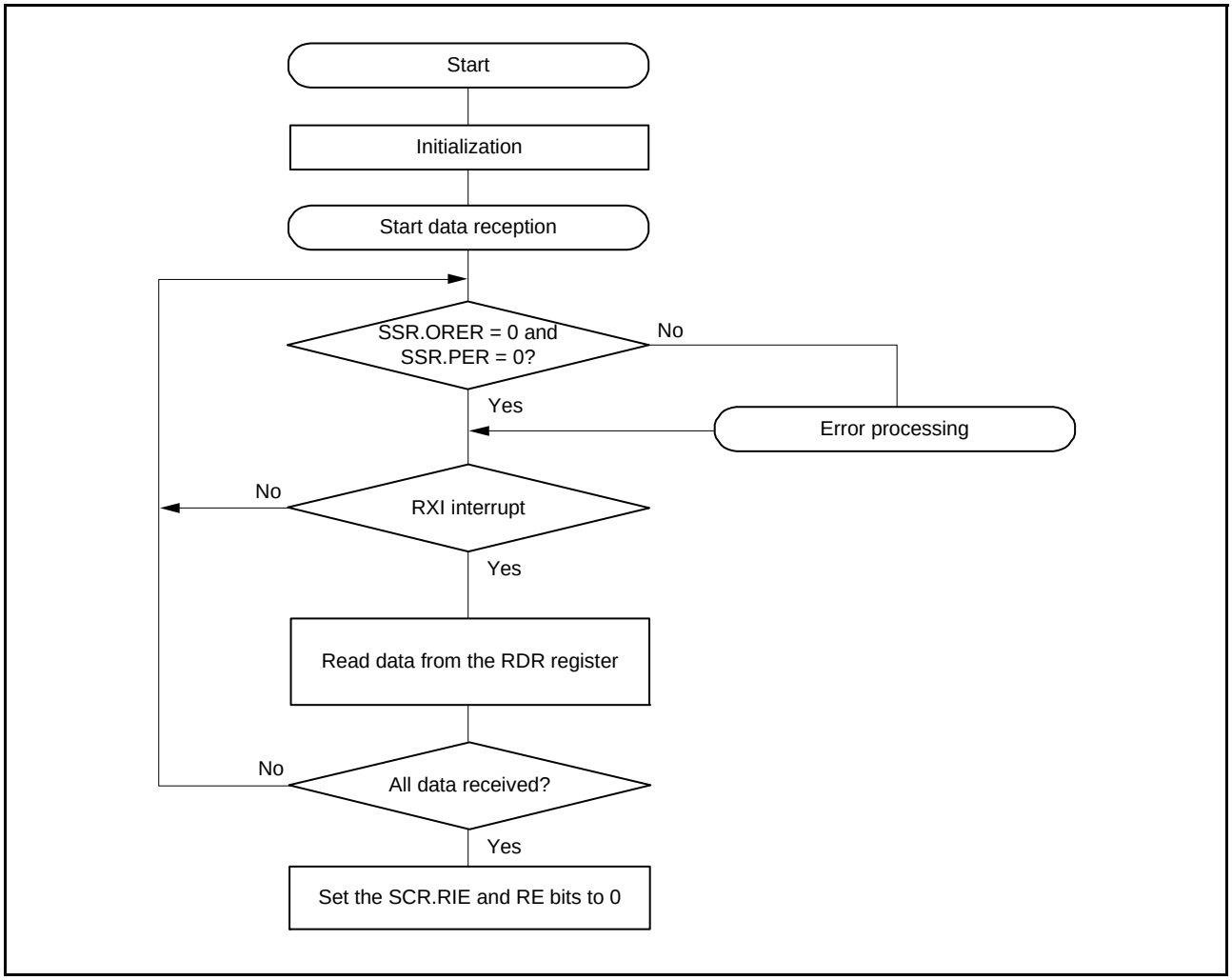


Figure 23.43 Sample Smart Card Interface Reception Flowchart

All the processing steps are automatically performed using an RXI interrupt request to activate the DTC.
In reception, setting the RIE bit to 1 allows an RXI interrupt request to be generated. The DTC is activated by an RXI interrupt request if the RXI interrupt request is specified as a source of DTC activation beforehand, allowing transfer of receive data.
If an error occurs during reception and either the ORER or PER flag in the SSR register 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 DTC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DTC is transferred.
Even if a parity error occurs and the PER flag is set to 1 during reception, receive data is transferred to the RDR register, thus allowing the data to be read.
When a reception is forcibly terminated by setting the SCR.RE bit to 0 during operation, read the RDR register because the received data which has not yet been read may be left in the RDR register.

Note 1. For operations in block transfer mode, refer to section 23.3, Operation in Asynchronous Mode.

23.6.8 Clock Output Control

Clock output can be fixed to high or low using the SCR.CKE[1:0] bits when the SMR.GM bit is 1. When the CKE[1:0] bits are set to 01b (clock output), the base clock is output from the SCK pin. For the settings of the base clock frequency (bit rate), refer to section 23.2.11, Bit Rate Register (BRR). When the CKE[1:0] bits are set to 00b (output fixed low) or 10b (output fixed to high), the SCK pin can be fixed to low or high.

Figure 23.44 shows a timing chart when the clock output is controlled.

If changing the CKE[1:0] bits while the SMR.GM bit is 0 (non-GSM mode), a pulse of unexpected width may output from SCK pin because the result is immediately reflected to the SCK pin.

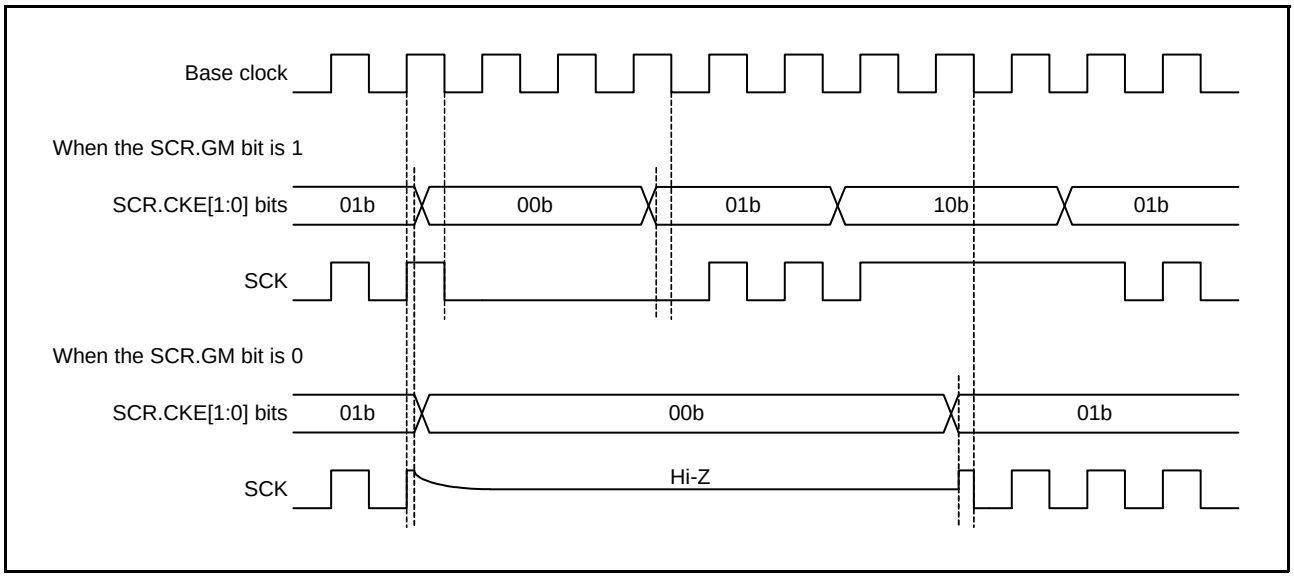


Figure 23.44 Clock Output Control

23.7 Operation in Simple I2C Mode

Simple I2C-bus format is composed of 8 data bits and an acknowledge bit. By continuing into a slave-address frame after a start condition or restart condition, a master device is able to specify a slave device as the partner for communications. The currently specified slave device remains valid until a new slave device is specified or a stop condition is satisfied.
The 8 data bits in all frames are transmitted in order from the MSB.
The I2C-bus format and timing of the I2C-bus are shown in Figure 23.45 and Figure 23.46.

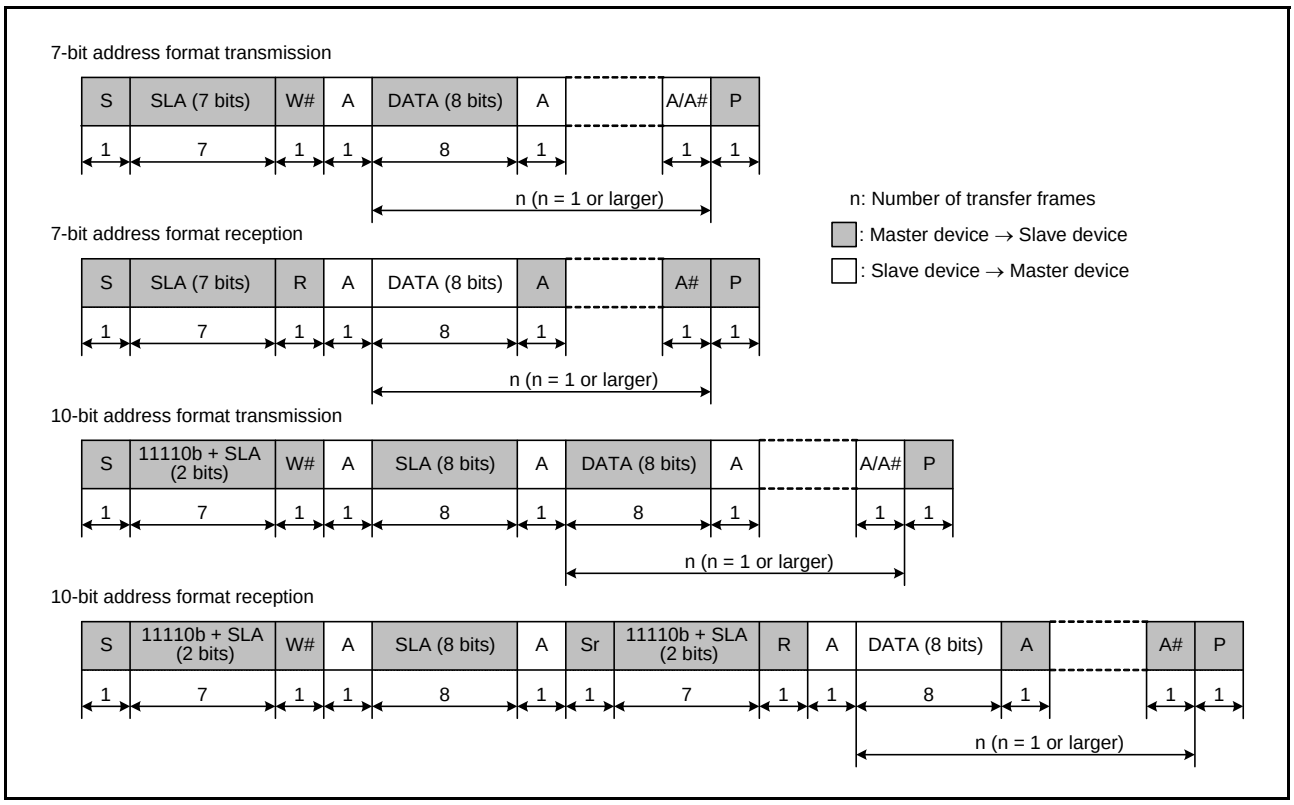


Figure 23.45 I2C-bus Format

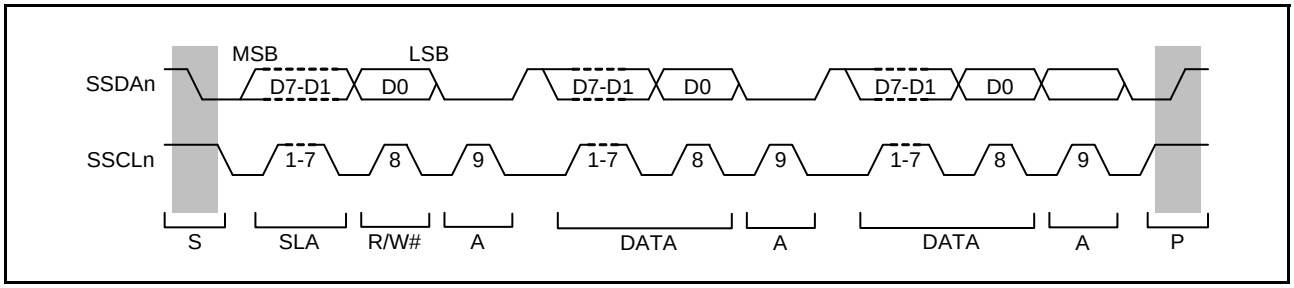


Figure 23.46 I2C-bus Timing (When SLA is 7 Bits)

S: Indicates a start condition, i.e. the master device changing the level on the SSDAn line from the high to the low level while the SSCLn line is at the high level.
SLA: Indicates a slave address, by which the master device selects a slave device.
R/W#: Indicates the direction of transfer (reception or transmission). The value 1 corresponds to transfer from the slave device to the master device and 0 corresponds to transfer from the master device to the slave device.
A/A#: Indicates an acknowledge bit. This is returned by the slave device for master transmission and by the master device for master reception. Return of the low level indicates ACK and return of the high level indicates NACK.
Sr: Indicates a restart condition, i.e. the master device changing the level on the SSDAn line from the high to the low level while the SSCLn line is at the high level and after the setup time has elapsed.
DATA: Indicates the data being received or transmitted.
P: Indicates a stop condition, i.e. the master device changing the level on the SSDAn line from the low to the high level while the SSCLn line is at the high level.

23.7.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to the SIMR3.IICSTAREQ bit causes the generation of a start condition. The generation of a start condition proceeds through the following operations.

- The level on the SSDAn line falls (from the high level to the low level) and the SSCLn 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 BRR.
- The level on the SSCLn line falls (from the high level to the low level), the SIMR3.IICSTAREQ bit is set (to 0), and a start-condition generated interrupt is output.

Writing 1 to the SIMR3.IICRSTAREQ bit causes the generation of a start condition. The generation of a start condition proceeds through the following operations.

- The SSDAn line is released and the SSCLn line is kept at the low level.
- The period at low level for the SSCLn line is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The SSCLn line is released (transition from the low to the high level).
- Once the high level on the SSCLn 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 BRR.
- The level on the SSDAn 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 BRR.
- The level on the SSCLn line falls (from the high level to the low level), the SIMR3.IICRSTAREQ bit is set (to 0), and a restart-condition generated interrupt is output.

Writing 1 to the SIMR3.IICSTPREQ bit causes the generation of a stop condition. The generation of a stop condition proceeds through the following operations.

- The level on the SSDAn line falls (from the high level to the low level) and the SSCLn line is kept at the low level.
- The period at low level for the SSCLn line is secured as half of a bit period at the bit rate determined by the setting of the BRR.
- The SSCLn line is released (transition from the low to the high level).
- Once the high level on the SSCLn 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 BRR.
- The SSDAn is released (transition from the low to the high level), the SIMR3.IICSTPREQ bit is set (to 0), and a stop-condition generated interrupt is output.

Figure 23.47 shows the timing of operations in the generation of start, restart, and stop conditions.

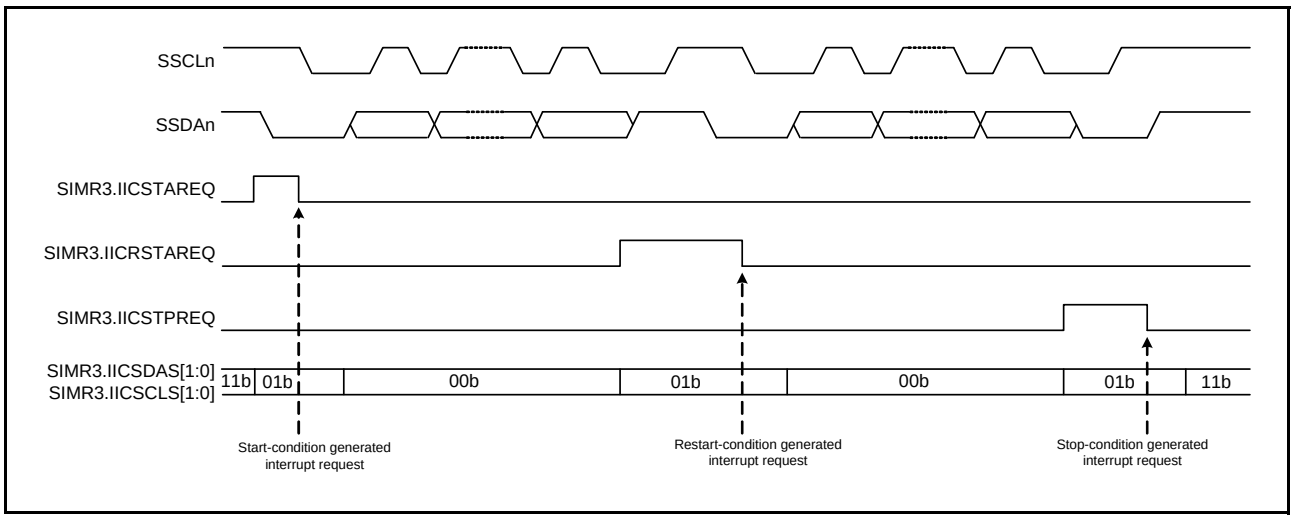


Figure 23.47 Timing of Operations in the Generation of Start, Restart, and Stop Conditions

23.7.2 Clock Synchronization

The SSCLn line may be placed at the low level in the case of a wait inserted by a slave device as the other side of transfer. Setting the SIMR2.IICCSC bit to 1 applies control to obtain synchronization when the levels of the internal SSCLn clock signal and the level being input on the SSCLn pin differ.

When the SIMR2.IICCSC bit is set to 1, the level of the internal SSCLn 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 SSCLn pin, and counting to determine the period at high level starts after the transition of the input on the SSCLn pin to the high level. The interval from this time until counting to determine the period at high level starts on the transition of the SSCLn pin to the high level is the total of the delay of SSCLn output, delay for noise filtering of the input on the SSCLn pin (2 or 3 cycles of sampling clock for the noise filter), and delay for internal processing (1 or 2 cycles of PCLK). The period at high level of the internal SSCLn clock is extended even if other devices are not placing the low level on the SSCLn line. If the SIMR2.IICCSC bit is 1, synchronization is obtained for the transmission and reception of data by taking the logical AND of the input on the SSCLn pin and the internal SSCLn clock. If the SIMR2.IICCSC bit is 0, synchronization with the internal SSCLn 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 SSCLn 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 SSCLn 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 23.48 shows an example of operations to synchronize the clocks.

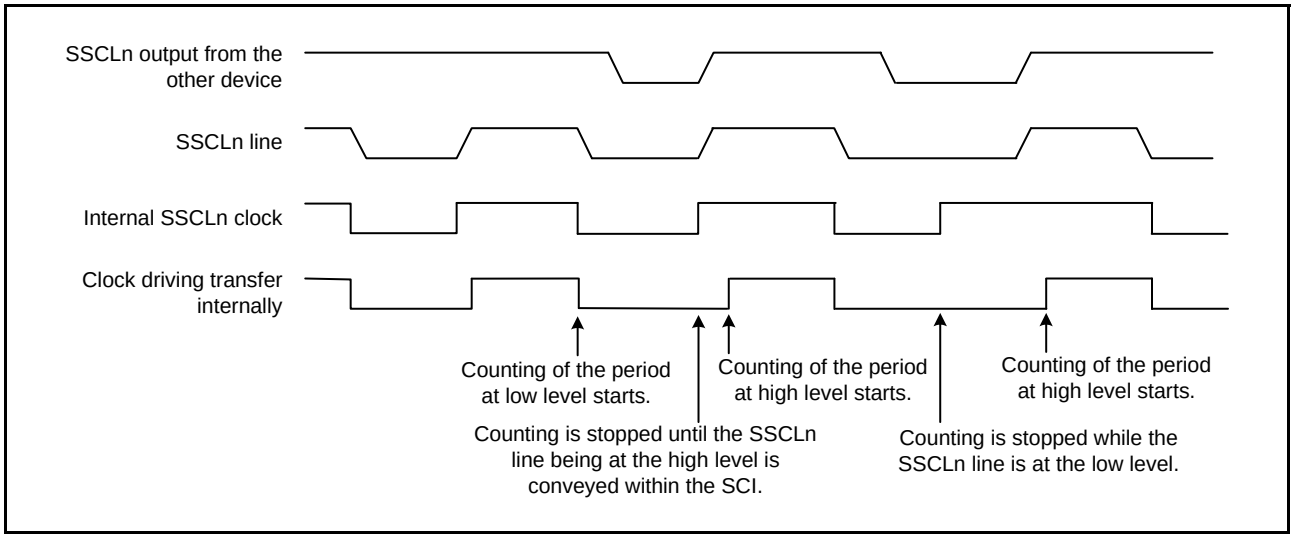


Figure 23.48 Example of Operations for Clock Synchronization

23.7.3 SSDA Output Delay

The SIMR1.IICDL[4:0] bits can be used to set a delay for output on the SSDAn pin relative to falling edges of output on the SSCLn pin. Delay-time settings from 0 to 31 are selectable, representing periods of the corresponding numbers of cycles of the clock signal from the on-chip baud rate generator (derived by frequency-dividing the base clock, PCLK, by the divisor selected by the SMR.CKS[1:0] bits). A delay for output on the SSDAn pin is for the start condition/restart condition/stop condition signal, 8-bit transmit data, and an acknowledge bit.

If the SSDA output delay is shorter than the time for the level on the SSCLn pin to fall, the change of the output on the SSDAn pin will start while the output level on the SSCLn pin is falling, creating a possibility of erroneous operation for slave devices. Ensure that settings for the delay of output on the SSDAn pin are for times greater than the time output on the SSCLn pin takes to fall (300 ns for I^2C-bus in normal mode and fast mode).

Figure 23.49 shows the timing of delays in SSDA output.

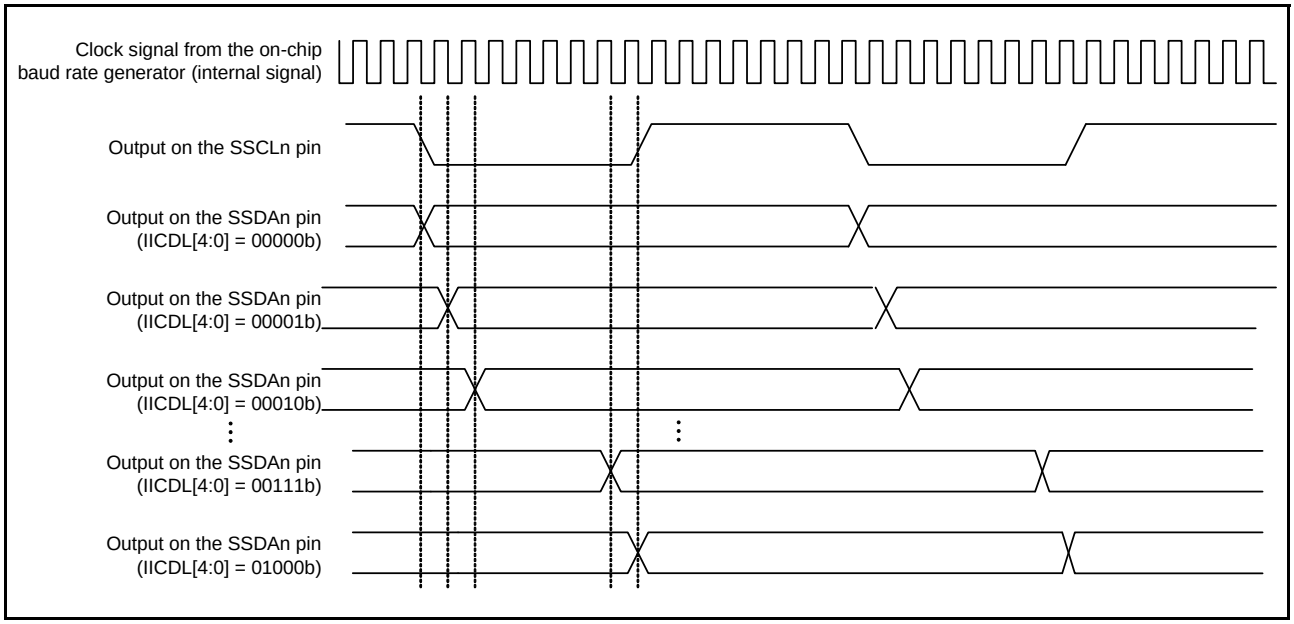
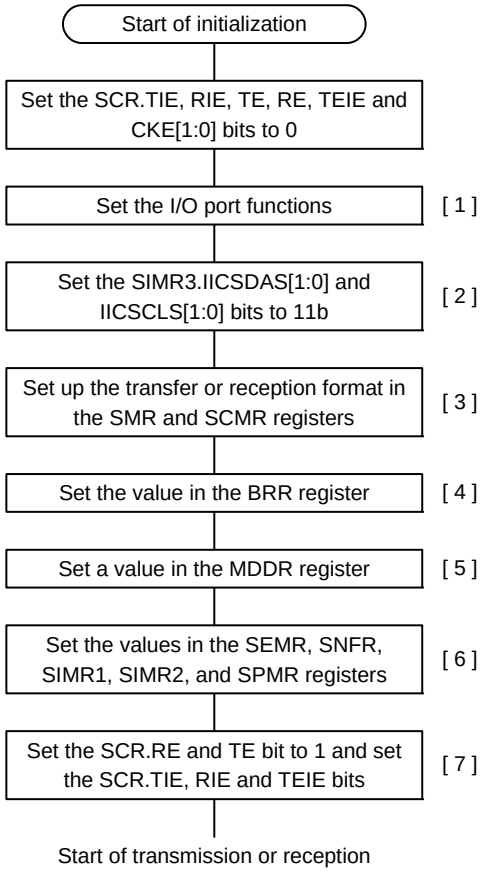


Figure 23.49 Timing of Delays in SSDA Output

23.7.4 SCI Initialization (Simple I2C Mode)

Before transferring data, write the initial value (00h) to the SCR register and initialize the interface following the example shown in Figure 23.50.
When changing the operating mode, transfer format, and so on, be sure to set the SCR register to its initial value before proceeding with the changes.
In simple I2C mode, the open-drain setting for the communication ports should be made on the port side.



[1] Make I/O port settings that allow use (on N-channel open-drain output pins) of the SSCLn and SSDAn pin functions.

[2] Place the SSCLn and SSDAn pins in the high-impedance state until a start condition is to be generated.

[3] Set the format for transmission and reception in the SMR and SCMR registers.
In the SMR register, set the CKS[1:0] bits to the desired value and set the other bits to 0.
In the SCMR register, set the SDIR bit to 1 and the SINV and SMIF bits to 0.

[4] Write the value for the desired bit rate to the BRR register.

[5] Write the value obtained by correcting a bit rate error in the MDDR register. This step is not necessary if the SEMR.BRME bit is set to 0.

[6] Set the values in the SEMR, SNFR, SIMR1, SIMR2, and SPMR registers.
In the SEMR register, set the NFEN and BRME bits.
In the SNFR register, set the NFCS[2:0] bits.
In the SIMR1 register, set the IICM bit to 1 and the IICDL[4:0] bits as required.
In the SIMR2 register, set the IICACKT and IICCSC bits to 1 and the IICINTM bits as required.
In the SPMR register, set all the bits to 0.

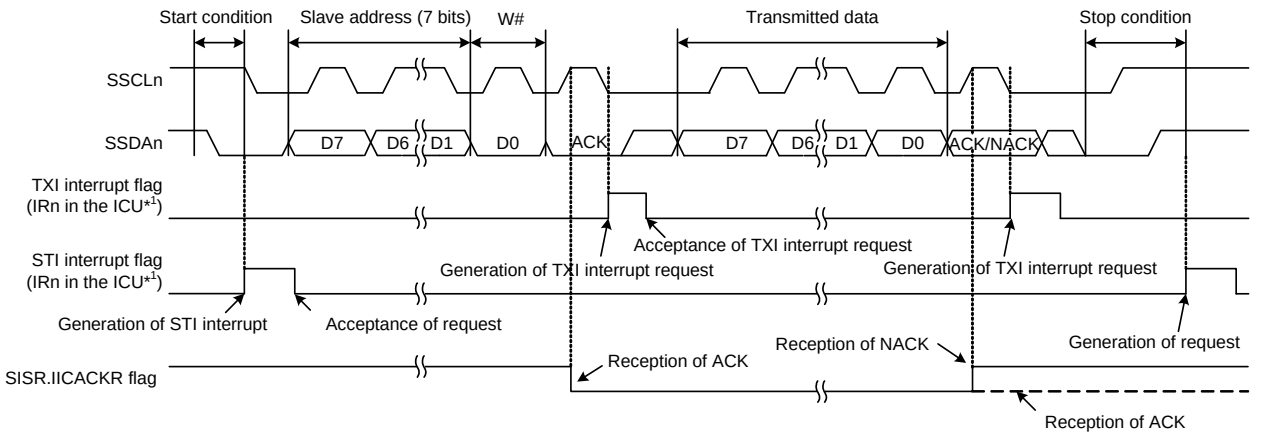
[7] Set the SCR.RE and TE bits to 1. At this time, also set the SCR.TIE, RIE, and TEIE bits (for transmission and when the SIMR2.IICINTM bit is 1, set the RIE bit to 0).
Setting the TE and RE bits to 1 makes the SSCLn and SSDAn pin functions available.

Note: Set the SCR.RE and TE bits to 0 or 1 at the same time.

Figure 23.50 Example of the Flowchart of SCI Initialization (for Simple I2C Mode)

23.7.5 Operation in Master Transmission (Simple I2C Mode)

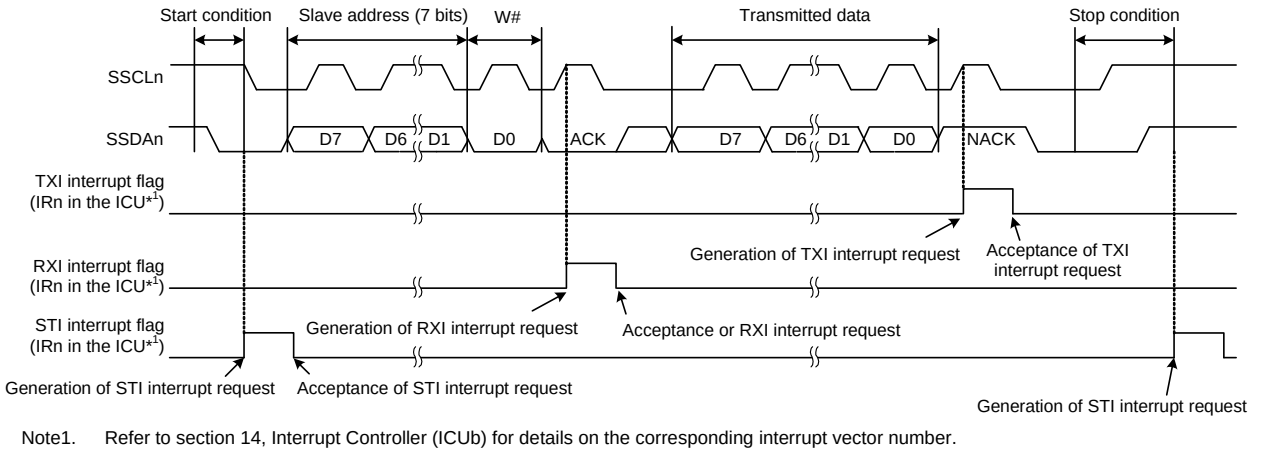
Figure 23.51 and Figure 23.52 show examples of operations in master transmission and Figure 23.53 is a flowchart showing the procedure for data transmission. Refer to Table 23.33 for more information on the STI interrupt.
When 10-bit slave addresses are in use, steps [3] and [4] in Figure 23.53 are repeated twice.
In simple I2C mode, the transmit data empty interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock synchronous transmission.



Note1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

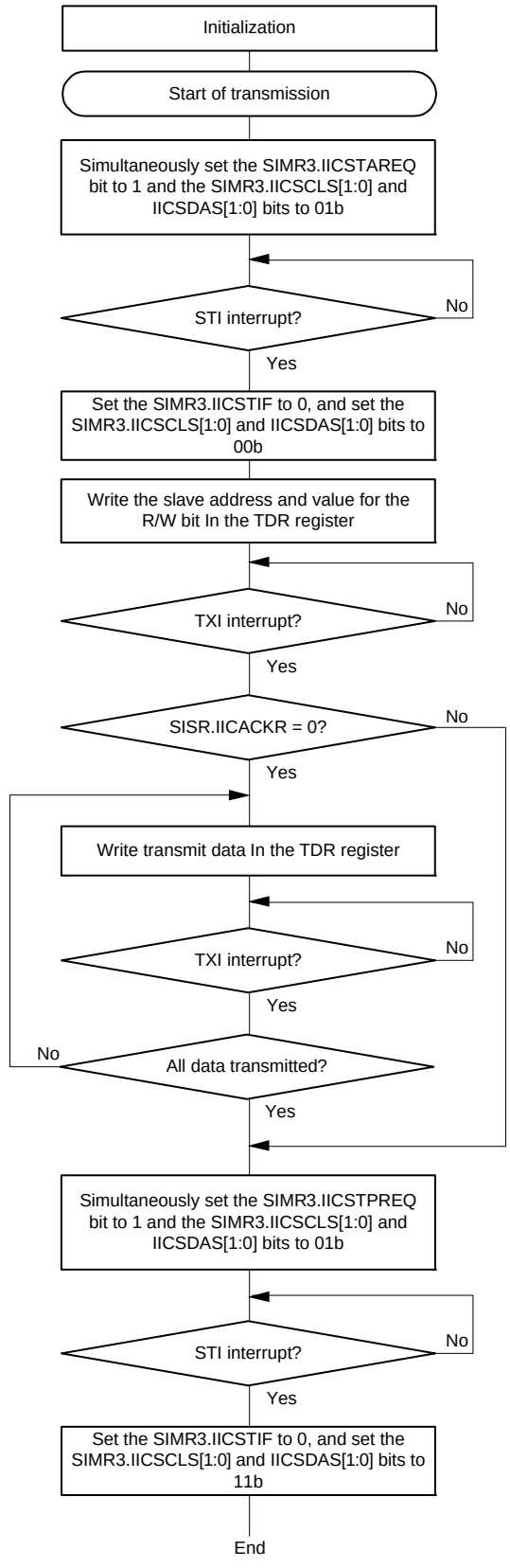
Figure 23.51 Example 1 of Operations for Master Transmission in Simple I2C-bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)

When the SIMR2.IICINTM bit is set to 0 (use ACK/NACK interrupts) during master transmission, the DTC is activated by the ACK interrupt as the trigger 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 interrupt as the trigger.



Note1. Refer to section 14, Interrupt Controller (ICUb) for details on the corresponding interrupt vector number.

Figure 23.52 Example 2 of Operations for Master Transmission in Simple I2C-bus Mode (with 7-Bit Slave Addresses, ACK Interrupts, and NACK Interrupts in Use)



[1] Initialization for simple I²C mode
For transmission, set the SCR.RIE bit to 0 (RXI and ERI interrupts requests are disabled)

[2] Generate a start condition.

[3] Writing to the TDR register:
Writing the slave address and value for the R/W bit to the TDR register.

[4] Confirming ACK response from the slave address:
Check the SISR.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 the TDR register.
Except for the first data to be transmitted, a TXI request can activate the DTC to handle writing of data to the TDR register. In this case, ACK/NACK cannot be confirmed.

[6] Generation of a stop condition.

[3] to [4] If 10-bit slave addresses are in use, processing of [3] and [4] is repeated twice.

Note: In simple I²C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock synchronous transmission.

Figure 23.53 Example of the Procedure for Master Transmission Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)

23.7.6 Master Reception (Simple I2C Mode)

Figure 23.54 shows an example of operations in simple I2C mode master reception and Figure 23.55 is a flowchart showing the procedure for master reception.
The value of the SIMR2.IICINTM bit is assumed to be 1 (use reception and transmission interrupts).
In simple I2C mode, the transmit data empty interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock synchronous transmission.

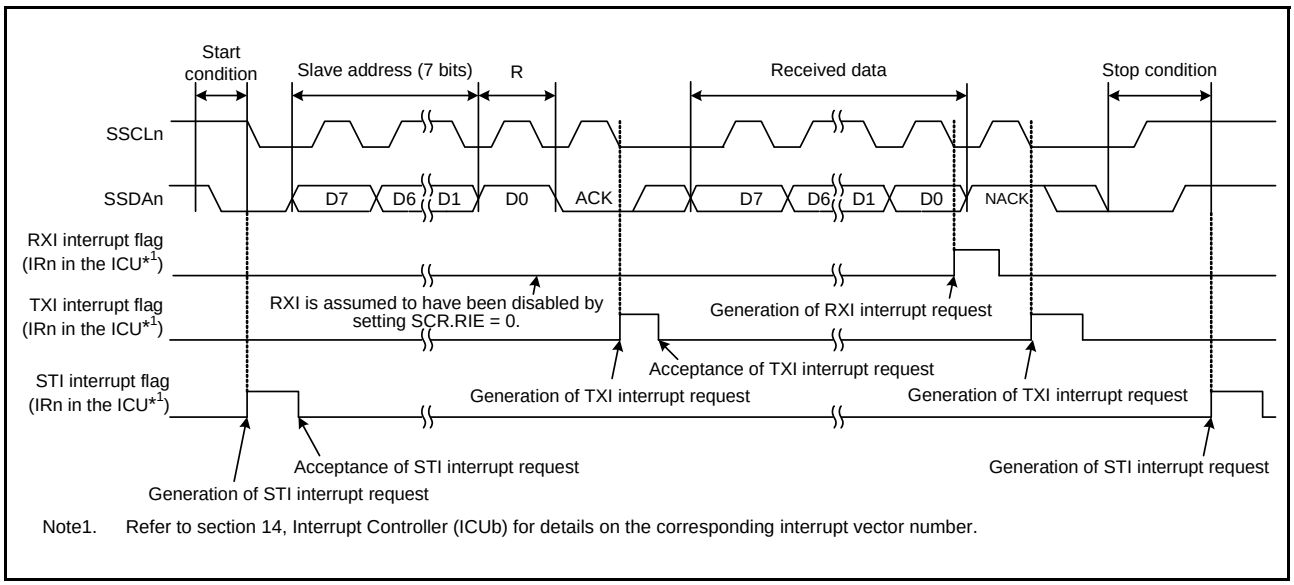
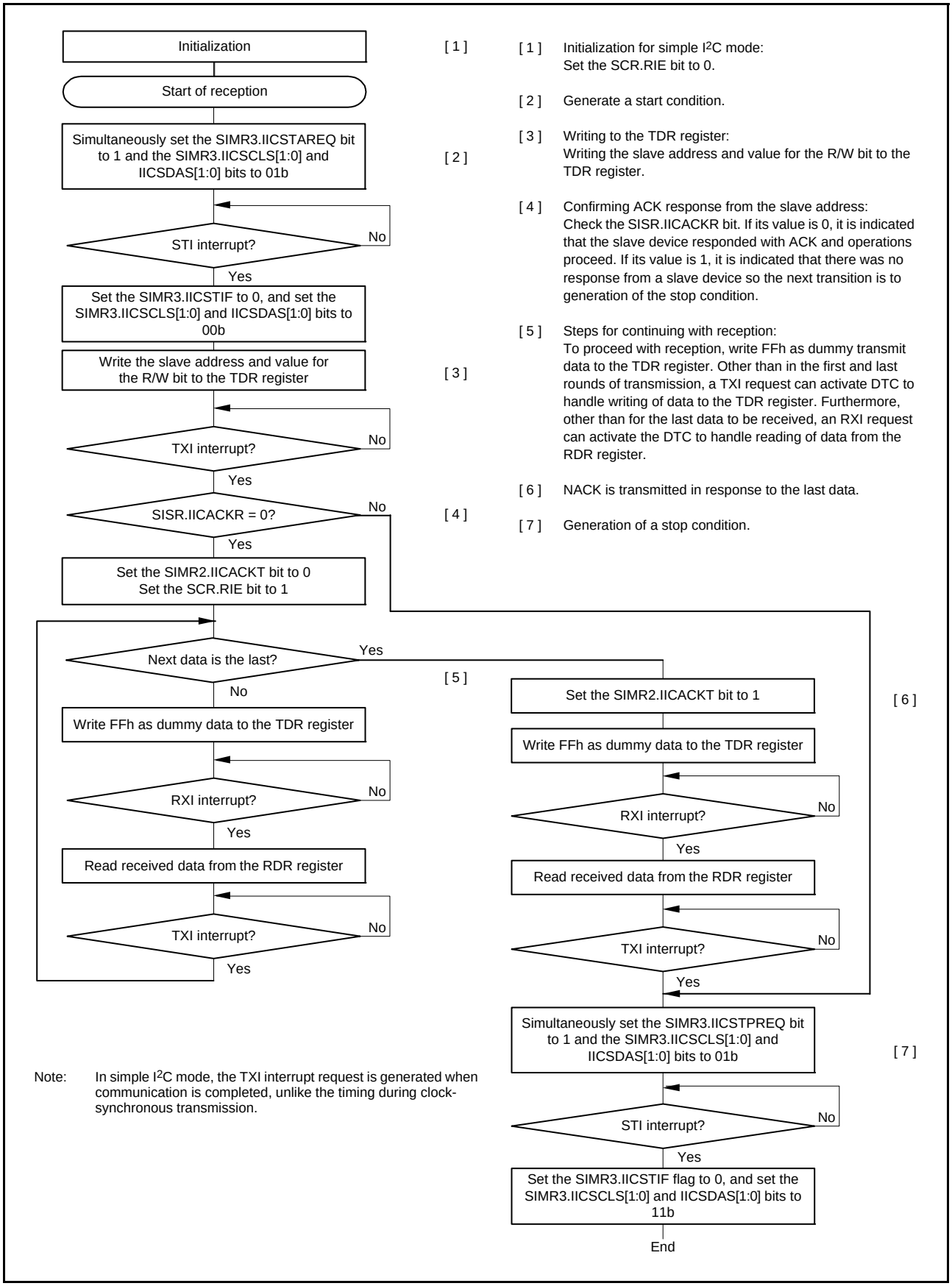


Figure 23.54 Example of Operations for Master Reception in Simple I2C-bus Mode (with 7-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)



[1] Initialization for simple I²C mode:
Set the SCR.RIE bit to 0.

[2] Generate a start condition.

[3] Writing to the TDR register:
Writing the slave address and value for the R/W bit to the TDR register.

[4] Confirming ACK response from the slave address:
Check the SISR.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 reception:
To proceed with reception, write FFh as dummy transmit data to the TDR register. Other than in the first and last rounds of transmission, a TXI request can activate DTC to handle writing of data to the TDR register. Furthermore, other than for the last data to be received, an RXI request can activate the DTC to handle reading of data from the RDR register.

[6] NACK is transmitted in response to the last data.

[7] Generation of a stop condition.

Note: In simple I²C mode, the TXI interrupt request is generated when communication is completed, unlike the timing during clock-synchronous transmission.

Figure 23.55 Example of the Procedure for Master Reception Operations in Simple I²C Mode (with Transmission Interrupts and Reception Interrupts in Use)

23.7.7 Recovery from Bus Hang-up

If the bus is stuck by an abnormal state in SCI because of the communication error, reset the SCI according to the following steps and release the bus.

(1) Set the SCR.TE and RE bit to 0 at the same time to reset SCI.
(2) Set the SIMR3 register to F0h to release the bus.
(3) If the SSR.RDRF flag is 1, dummy-read the RDR register to clear the flag.
(4) Set the SCR.TE and RE bit to 1 at the same time.

23.8 Operation in Simple SPI Mode

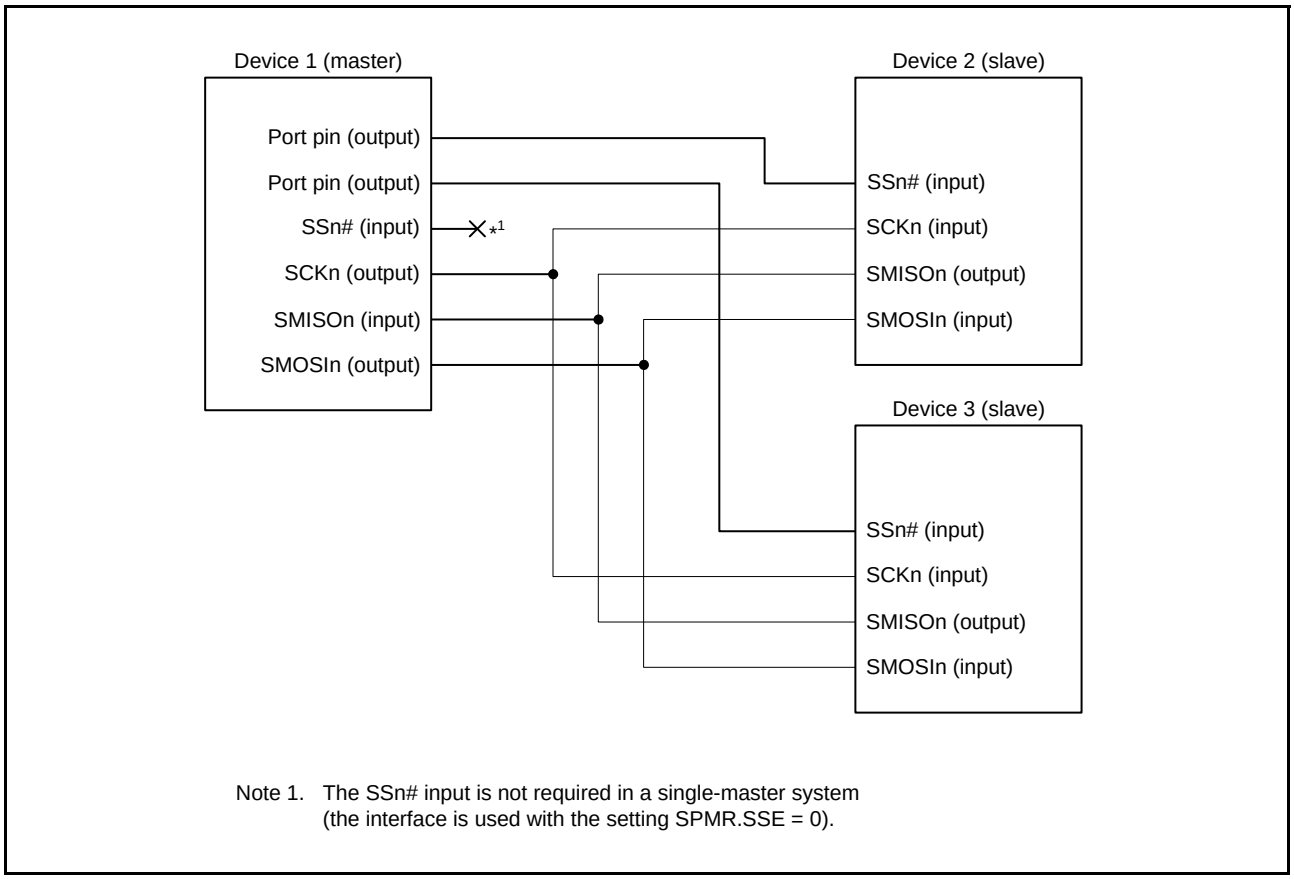
As an extended function, the SCI supports a simple SPI mode that handles transfer among one or multiple master devices and multiple slave devices.

Making the settings for clock synchronous mode (SCMR.SMIF = 0, SIMR1.IICM = 0, SMR.CM = 1) plus setting the SPMR.SSE bit to 1 places the SCI in simple SPI mode. However, the SS pin function on the master side is unnecessary for connection of the device used as the master in simple SPI mode when the configuration only has a single master, so set the SPMR.SSE bit to 0 in such cases.

Figure 23.56 shows an example of connections for simple SPI mode. Control a general port pin to produce the SS output signal from the master.

In simple SPI mode, data are transferred in synchronization with clock pulses in the same way as in clock synchronous mode. One character of data for transfer consists of 8 bits of data, and parity bits cannot be appended to this. The data can be inverted by setting the SCMR.SINV bit to 1.

Since the receiver and transmitter are independent of each other within the SCI module, full-duplex communications are possible, with a common clock signal. Furthermore, since both the transmitter and receiver have a double-buffered structure, writing of further transmit data while transmission is in progress and reading of previously received data while reception is in progress are both possible. Continuous transfer is thus possible.



Note 1. The SSn# input is not required in a single-master system (the interface is used with the setting SPMR.SSE = 0).

Figure 23.56 Example of Connections via a Simple SPI Mode (In Single Master Mode, SPMR.SSE Bit = 0)

23.8.1 States of Pins in Master and Slave Modes

The direction (input or output) of pins for the simple SPI mode interface differs according to whether the device is a master (SCR.CKE[1:0] = 00b or 01b and SPMR.MSS = 0) or slave (SCR.CKE[1:0] = 10b or 11b and SPMR.MSS = 1). Table 23.29 lists the states of pins according to the mode and the level on the SSn# pin.

Table 23.29 States of Pins by Mode and Input Level on the SSn# Pin

Mode	Input on SSn# Pin	State of SMOSIn Pin	State of SMISOn Pin	State of SCKn Pin
Master mode*1	High level (transfer can proceed)	Output for data transmission*2	Input for received data	Clock output*3
	Low level (transfer cannot proceed)	High-impedance	Input for received data (but disabled)	High-impedance
Slave mode	High level (transfer cannot proceed)	Input for received data (but disabled)	High-impedance	Clock input (but disabled)
	Low level (transfer can proceed)	Input for received data	Output for data transmission	Clock input

Note 1. When there is only a single master (SPMR.SSE = 0), transfer is possible regardless of the input level on the SSn# pin (this is equivalent to input of a high level on the SSn# pin). Since the SSn# pin function is not required, the pin is available for other purposes.
Note 2. The SMOSIn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE bit = 0).
Note 3. The SCKn pin output is in the high-impedance state when serial transmission is disabled (SCR.TE and RE bits = 00b) in a multi-master configuration (SPMR.SSE = 1).

23.8.2 SS Function in Master Mode

Setting the SCR.CKE[1:0] bits to 00b and the SPMR.MSS bit to 0 selects master operation. The SSn# pin is not used in single-master configurations (SPMR.SSE = 0), so transmission or reception can proceed regardless of the value of the SSn# pin.

When the level on the SSn# pin is high in a multi-master configuration (SPMR.SSE = 1), a master device outputs clock signals from the SCKn pin before starting transmission or reception to indicate that there are no other masters or another master is performing reception or transmission. When the level on the SSn# pin is low in a multi-master configuration (SPMR.SSE = 1), there are other masters, and this indicates that transmission or reception is in progress. At this time the SMOSIn output and SCKn pins will be placed in the high-impedance state and starting transmission or reception will not be possible. Furthermore, the value of the SPMR.MFF bit will be 1, indicating a mode fault error. In a multi-master configuration, start error processing by reading SPMR.MFF flag. Also, even if a mode fault error occurs while transmission or reception is in progress, transmission or reception will not be stopped, but the SMOSIn and SCKn pin output will be placed in the high-impedance state after the completion of the transfer.

Control a general port pin to produce the SS output signal from the master.

23.8.3 SS Function in Slave Mode

Setting the SCR.CKE[1:0] bits to 10b and the SPMR.MSS bit to 1 selects slave operation. When the level on the SSn# pin is high, the SMISOn output pin will be in the high-impedance state and clock input through the SCKn pin will be ignored. When the level on the SSn# pin is low, clock input through the SCKn pin will be effective and transmission or reception can proceed.

If the input on the SSn# pin changes from low to high level during transmission or reception, the SMISOn output pin will be placed in the high-impedance state. Meanwhile, the internal processing for transmission or reception will continue at the rate of the clock input through the SCKn pin until processing for the character currently being transmitted or received is completed, after which it stops. At that time, an interrupt (the appropriate one from among TXI, RXI, and TEI) will be generated.

23.8.4 Relationship between Clock and Transmit/Receive Data

The CKPOL and CKPH bits in the SPMR register can be used to set up the clock for use in transmission and reception in four different ways. The relation between the clock signal and the transmission and reception of data is shown in Figure 23.57. The relation is the same for both master and slave operation.

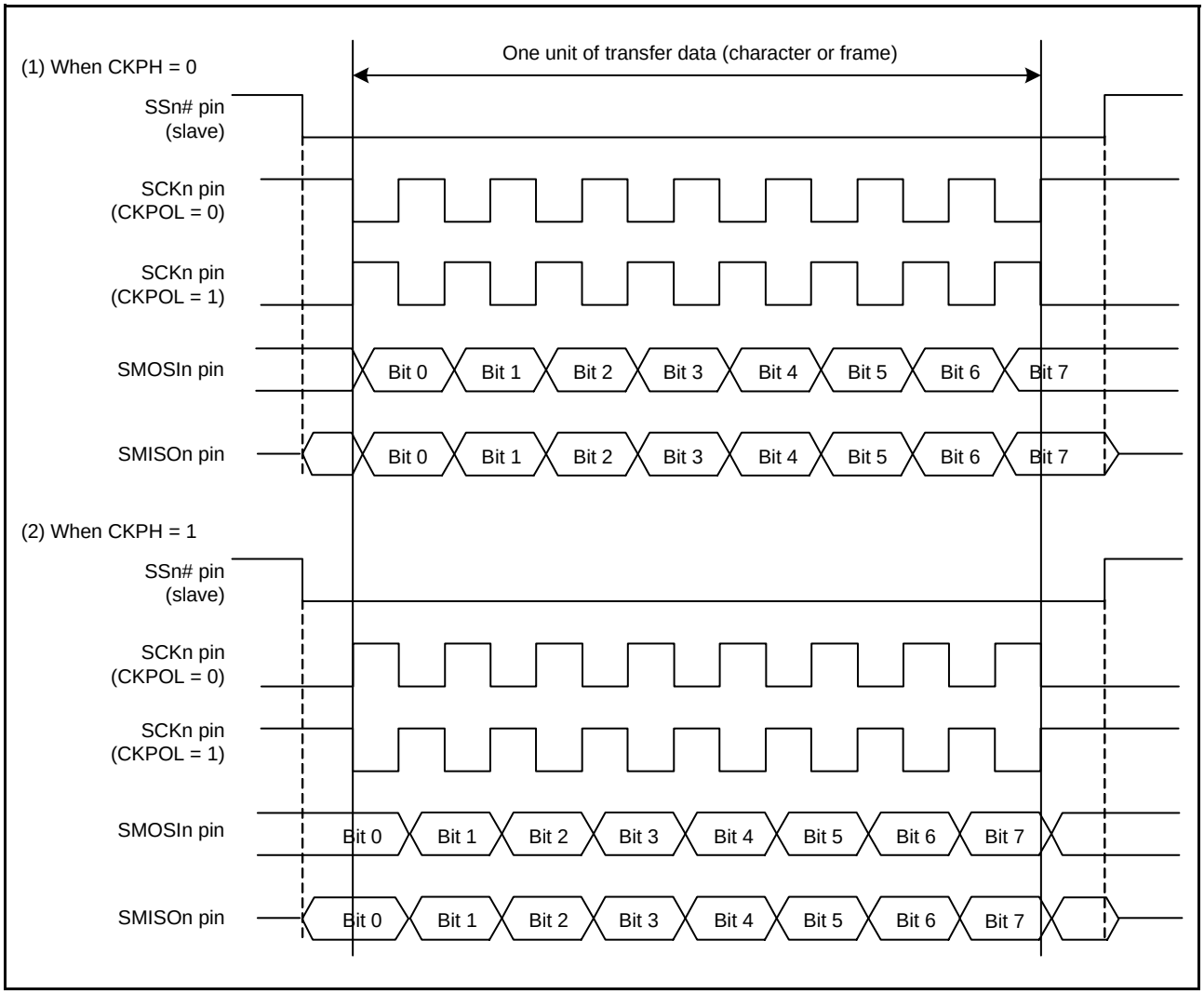


Figure 23.57 Relation between Clock Signal and Transmit/Receive Data in Simple SPI Mode

23.8.5 SCI Initialization (Simple SPI Mode)

The procedure is the same as for initialization in clock synchronous mode Figure 23.23, Sample SCI Initialization Flowchart. The CKPOL and CKPH bits in the SPMR register must be set to ensure that the kind of clock signal they select is suitable for both master and slave devices.
For initialization, changes to the operating mode, changes to the transfer format, and so on, initialize the SCR register before proceeding with changes.
As well as setting the RE bit to 0, note that the SSR.ORER, FER, and PER flags, as well as the RDR register, are not initialized.

23.8.6 Transmission and Reception of Serial Data (Simple SPI Mode)

In master operation, ensure that the SSn# pin of the slave device on the other side of the transfer is at the low level before starting the transfer and at the high level on completion of the transfer. Otherwise, the procedures are the same as in clock synchronous mode.

23.9 Bit Rate Modulation Function

The bit rate modulation function corrects the bit rate by thinning out the specified amount of clocks from those input to the baud rate generator.
When the SEMR.BRME bit is 1, the baud rate generator validates and counts the average interval of the number of clocks set in the MDDR register out of the total 256 clocks input.
Figure 23.58 assumes the SCI is in asynchronous mode, bits SMR.CKS[1:0] are 00b, the BRR register is 00h, and the MDDR register is 160. In this example, the cycle of the base clock is evenly corrected to 256/160, and the bit rate is corrected to 160/256. Note that there is an imbalance in thinning out the internal clock, and expansion and contraction occur in the pulse width of the base clock.

Note: Do not use this function in clock synchronous mode and in the highest speed settings in simple SPI mode (SMR.CKS[1:0] = 00b, SCR.CKE[1] = 0, and BRR = 0).

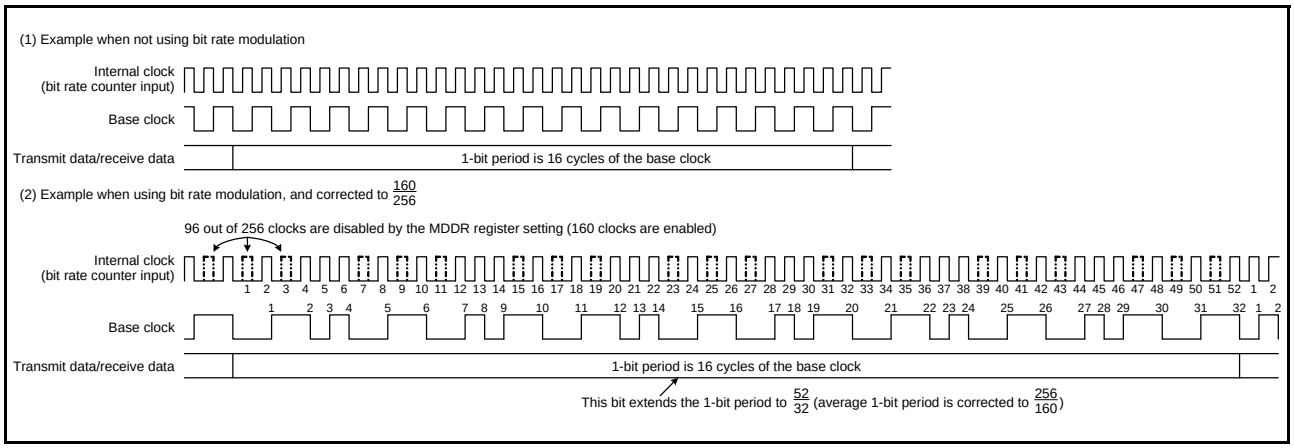


Figure 23.58 Example of the Base Clock When the Bit Rate Modulation Function is Used

The input of a clock signal with a shorter period to the baud rate generator reduces difference in the generated base clock period and, since the division ratio of the baud rate generator also becomes larger, reduces difference in the length of the 1-bit period.

23.10 Extended Serial Mode Control Section: Description of Operation

23.10.1 Serial Transfer Protocol

The extended serial mode control section of the SCI12 can realize the serial transfer protocol composed of Start Frames and Information Frames that is shown in Figure 23.59.
A Start Frame is composed of a Break Field, Control Field 0, and Control Field 1. An Information Frame is composed of a number of Data Fields, a CRC16 Upper Field, and a CRC16 Lower Field.

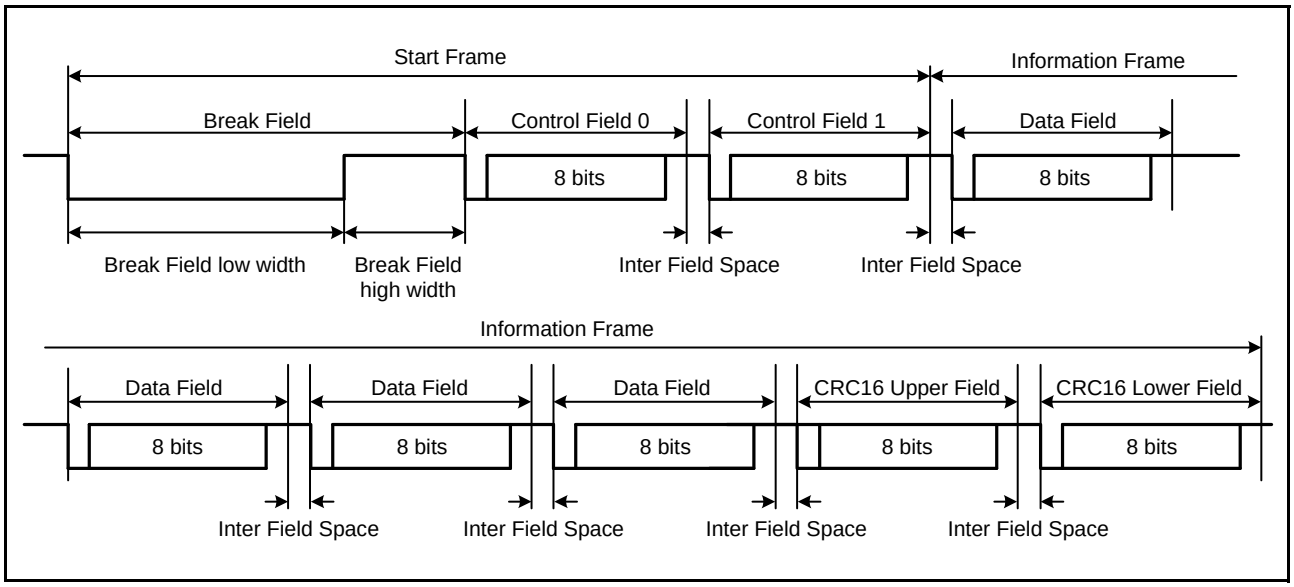


Figure 23.59 Protocol for Serial Transfer by the Extended Serial Mode Control Section

23.10.2 Transmitting a Start Frame

Figure 23.60 shows an example of operations to transmit a Start Frame, which is composed of the Break Field low width, Control Field 0, and Control Field 1. Figure 23.61 and Figure 23.62 are flowcharts for the transmission of a Start Frame.
Operations when the extended serial mode control section is to be used to transmit a Start Frame are as listed below. Be sure to use the SCI12 in asynchronous mode.

(1) With Break Field low width output mode as the operating mode for the timer, writing 1 to the TCR.TCST bit starts counting by the timer, and the low level will be output from the TXDX12 pin over the period corresponding to registers TCNT and TPRE settings.
(2) The output on the TXDX12 pin is inverted when the timer counter underflows, and the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1.
(3) Write 0 to the TCR.TCST bit to stop counting by the timer, and send the data for Control Field 0. After the Break Field low width output, stop counting before the next underflow occurs.
(4) When the data for Control Field 0 have been transmitted, the data for Control Field 1 is transmitted.
(5) When the data for Control Field 1 have been transmitted, an Information Frame is transmitted.

Omit the Break Field and Control Field 0 to suit the structure of the Start Frame.

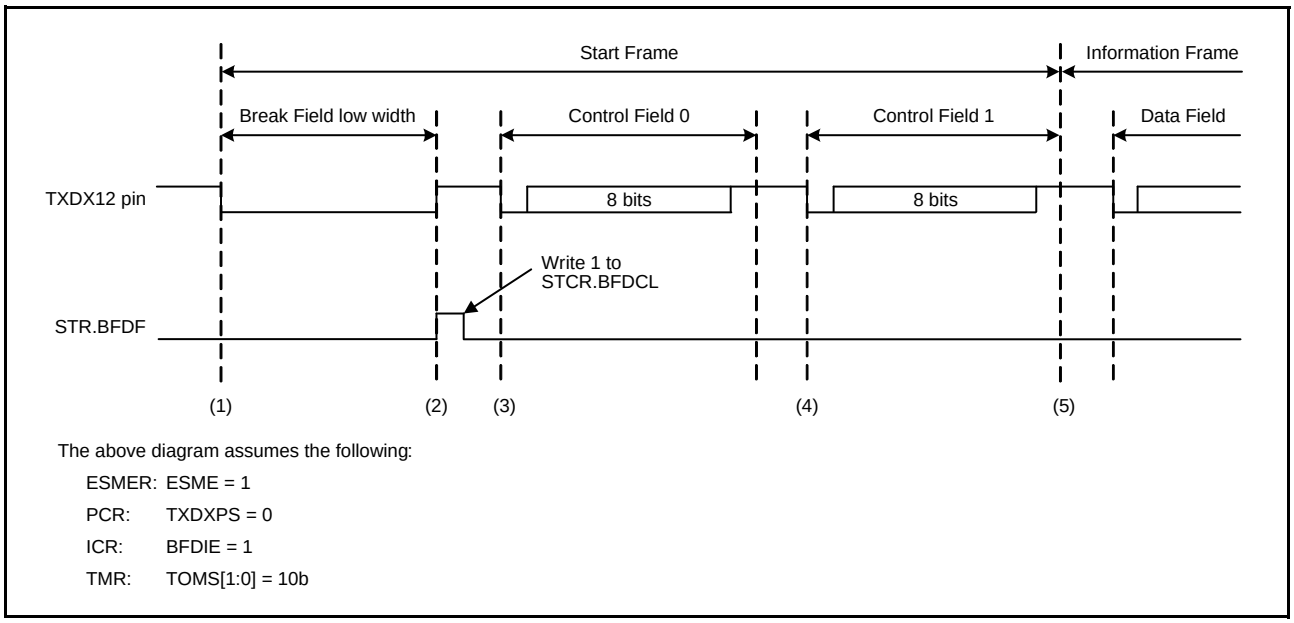


Figure 23.60 Example of Operations When Transmitting a Start Frame

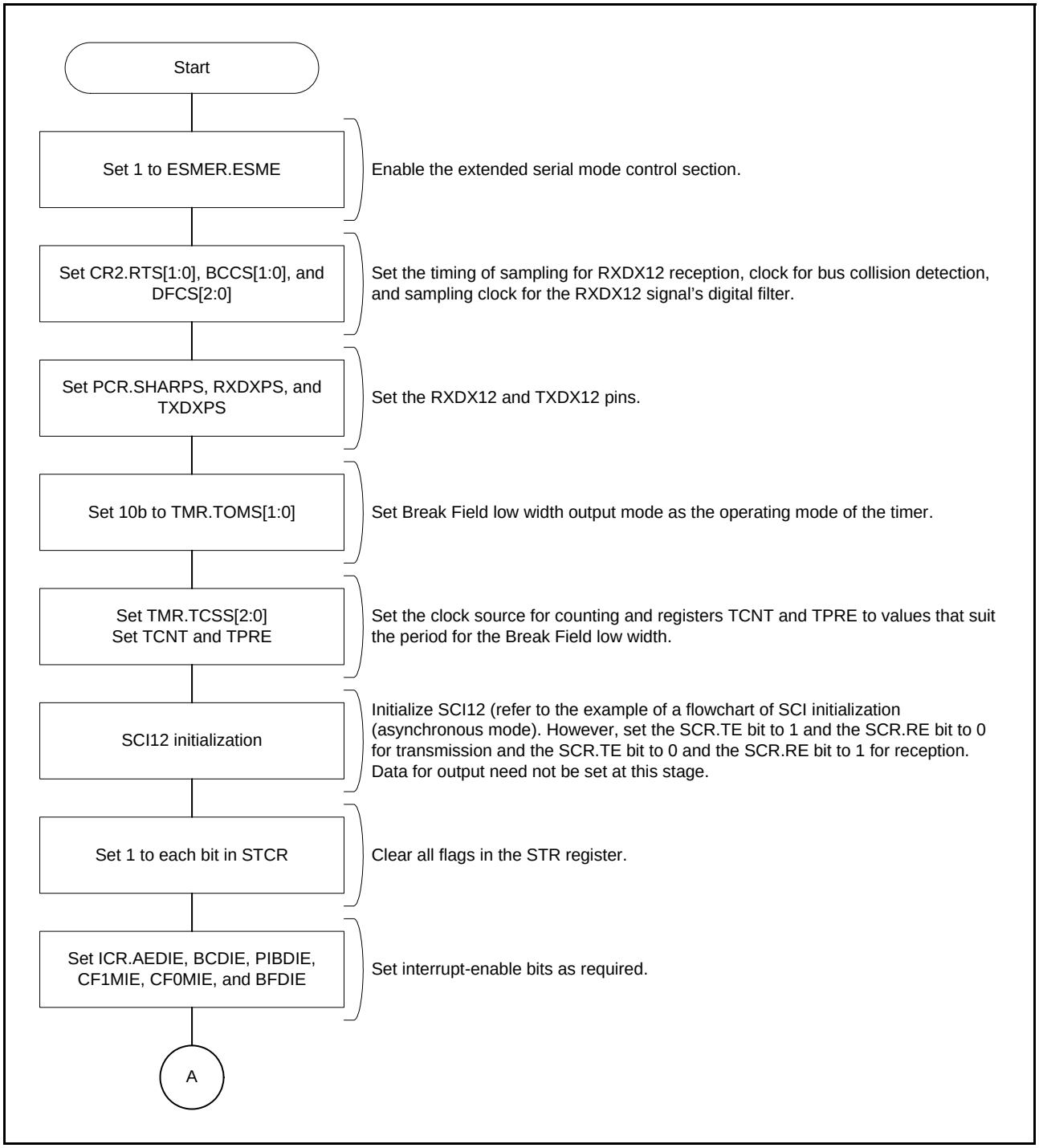


Figure 23.61 Example of Start Frame Transmission (1/2)

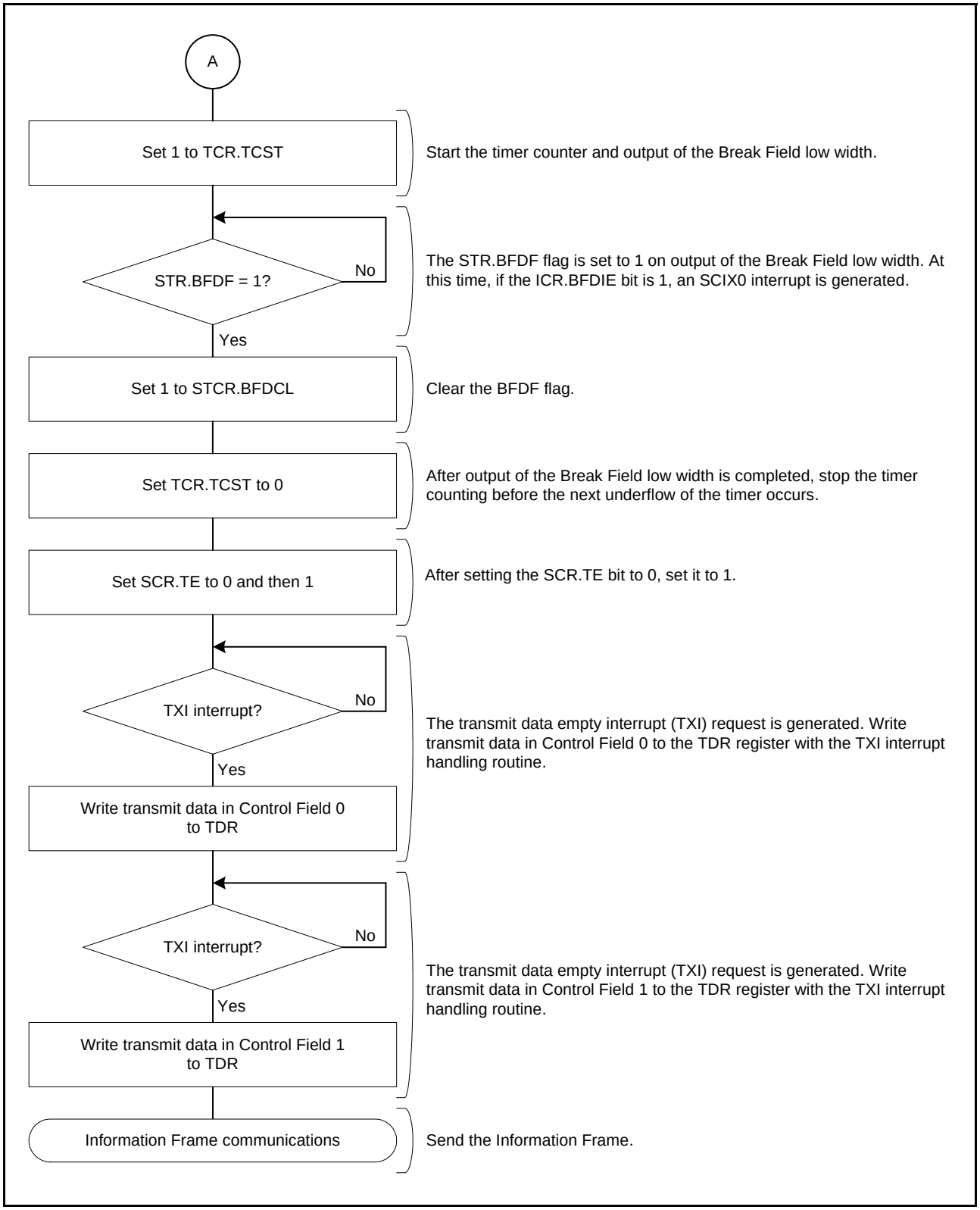


Figure 23.62 Example of Start Frame Transmission (2/2)

23.10.3 Receiving a Start Frame

The extended serial mode control section is capable of receiving Start Frames with the structures listed in Table 23.30.

Table 23.30 Structures of Start Frames

Bit Setting		Structures of Start Frames
BFE	CF0RE	
0	0	Control Field 1 (8 bits) → Information Frame
0	1	Control Field 0 (8 bits) → Control Field 1 (8 bits) → Information Frame
1	0	Break Field (low width) → Control Field 1 (8 bits) → Information Frame
1	1	Break Field (low width) → Control Field 0 (8 bits) → Control Field 1 (8 bits) → Information Frame

Figure 23.63 shows an example of operations to receive a Start Frame, which is composed of the Break Field low width, Control Field 0, and Control Field 1. Figure 23.64 and Figure 23.65 are flowcharts for the reception of a Start Frame, and Figure 23.66 is a state transition diagram when receiving a Start Frame.

Operations when the extended serial mode control section is to be used to receive a Start Frame are as listed below. Be sure to use the SCI12 in asynchronous mode.

(1) With Break Field low width detection mode as the operating mode for the timer, writing 1 to the CR3.SDST bit enables detection of the Break Field low width.

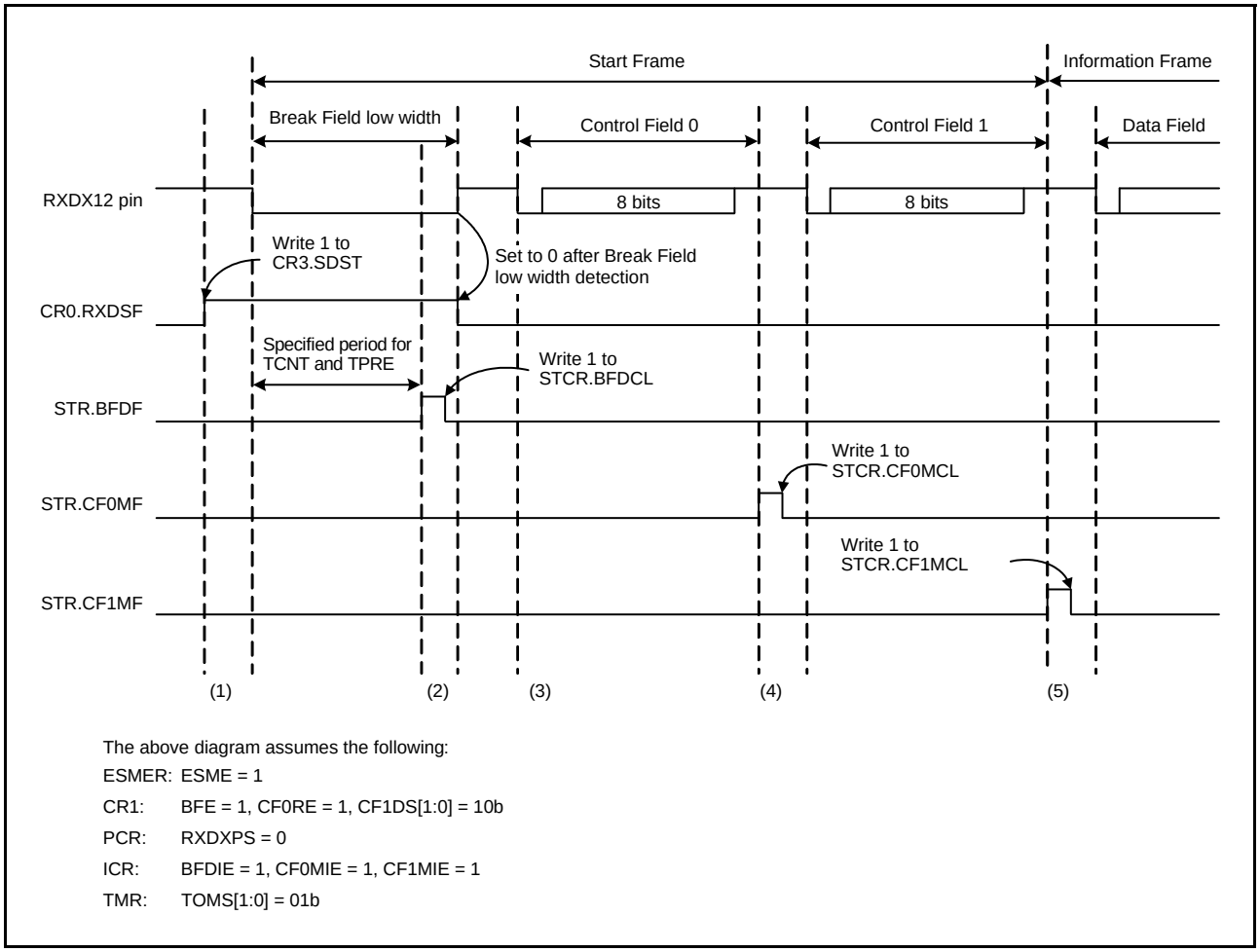
(2) Low-level input on the RXDX12 pin continuing over a period longer than that corresponding to the settings of registers TCNT and TPRE is detected as the Break Field low width. At this time, the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1.

(3) When the input from the RXDX12 pin goes high after the Break Field low width, the CR0.RXDSF flag becomes 0 and reception of Control Field 0 starts.

(4) If the data received in Control Field 0 match the data set in the CF0DR register, the STR.CF0MF flag is set to 1. An SCIX1 interrupt is also generated if the value of the ICR.CF0MIE bit is 1. Reception of Control Field 1 starts after that. If the data received in Control Field 0 do not match the data set in the CF0DR register, a transition to the state prior to Break Field low width detection proceeds.

(5) If the data received in Control Field 1 match the data set in registers PCF1DR and SCF1DR, the STR.CF1MF flag is set to 1. An SCIX1 interrupt is also generated if the value of the ICR.CF1MIE bit is 1. Transfer of the Information Frame starts after that. If the data received in Control Field 1 do not match the data set in either or both of registers PCF1DR and SCF1DR, a transition to the state prior to Break Field low width detection proceeds.

Omit the Break Field and Control Field 0 to suit the structure of the Start Frame.



The above diagram assumes the following:

ESMER: ESME = 1

CR1: BFE = 1, CF0RE = 1, CF1DS[1:0] = 10b

PCR: RXDXPS = 0

ICR: BFDIE = 1, CF0MIE = 1, CF1MIE = 1

TMR: TOMS[1:0] = 01b

Figure 23.63 Example of Operations at the Time of Start Frame Reception

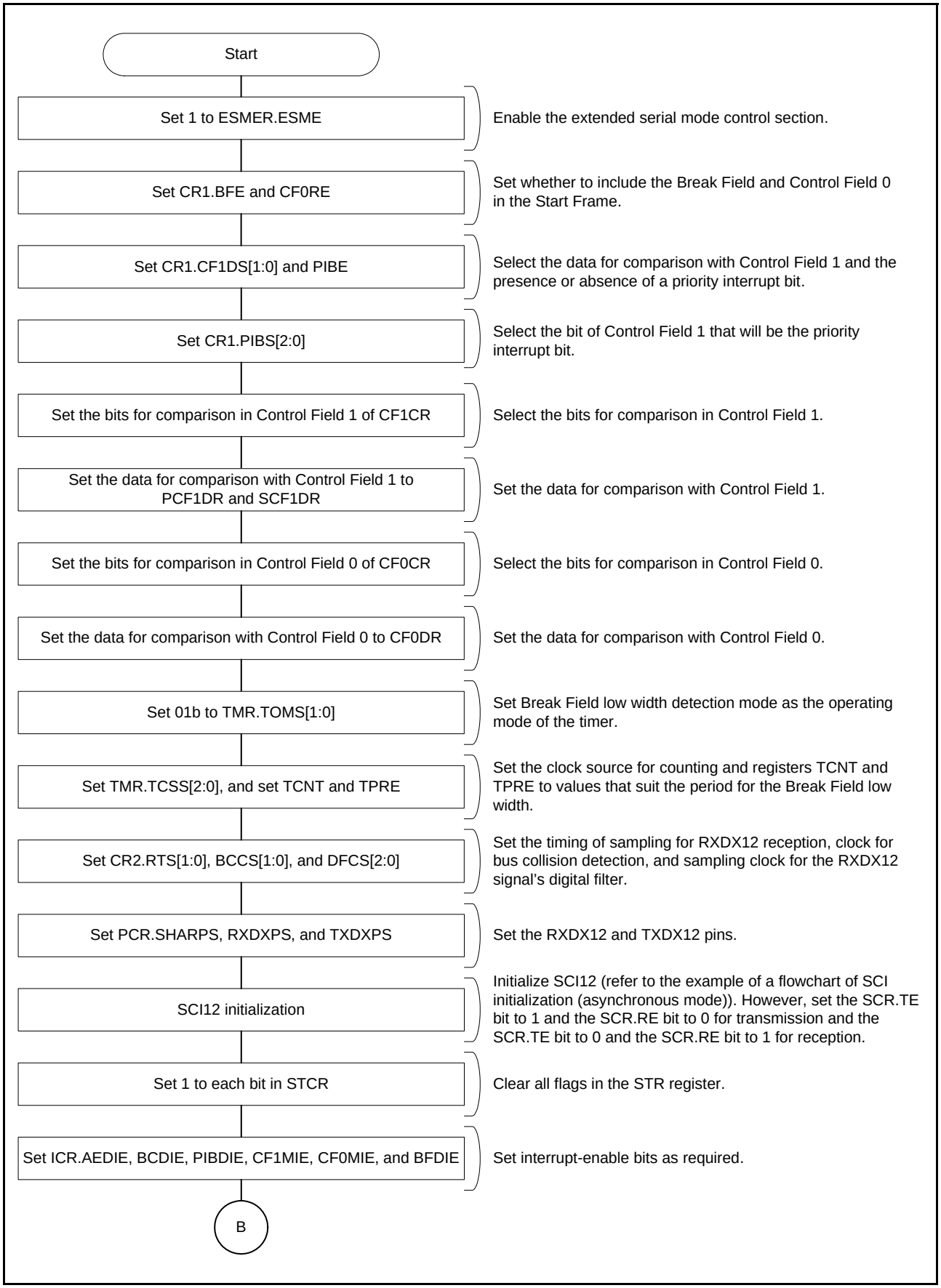


Figure 23.64 Sample Flowchart for Reception of a Start Frame (1)

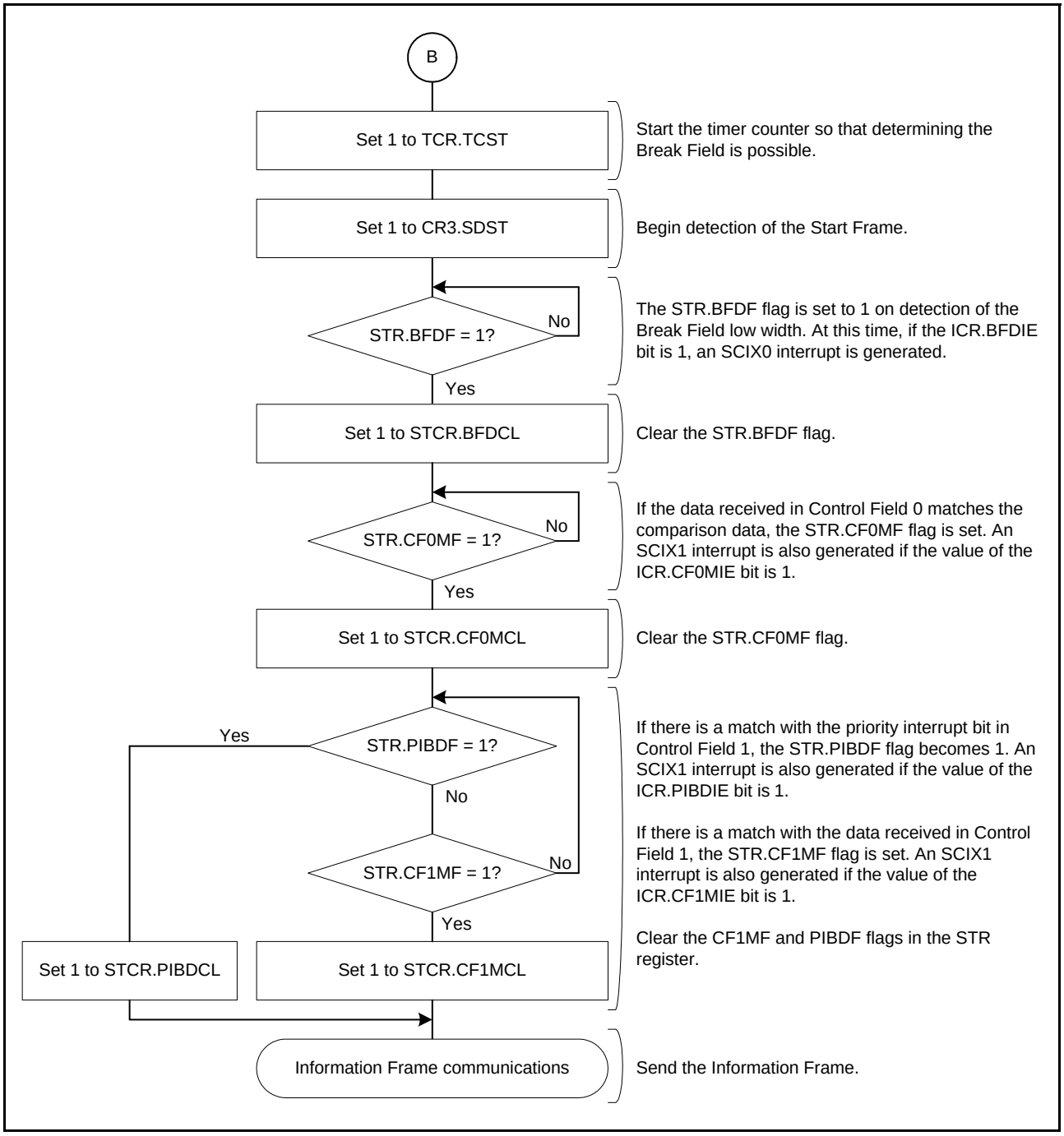


Figure 23.65 Sample Flowchart for Reception of a Start Frame (2)

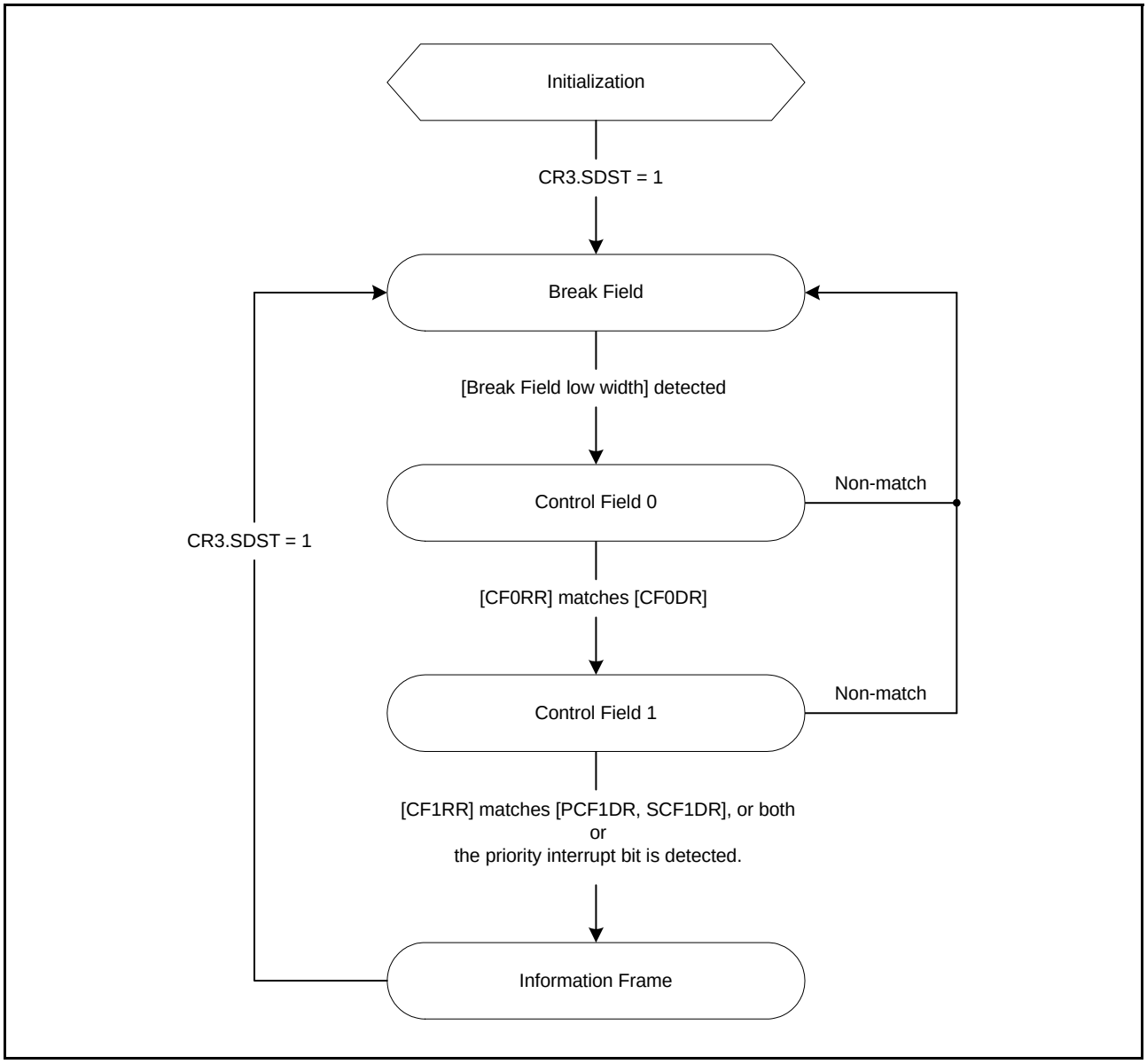


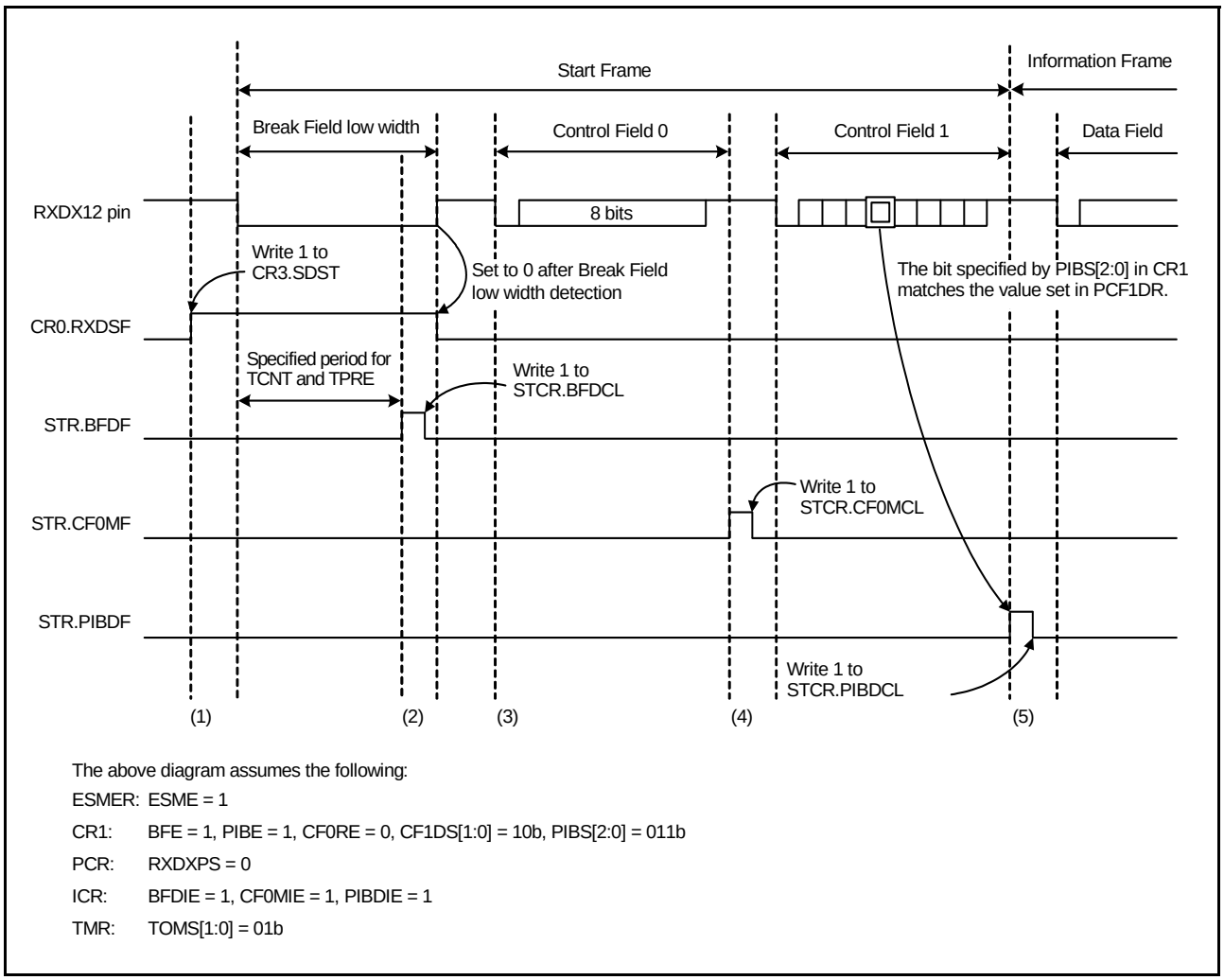
Figure 23.66 State Transitions When Receiving a Start Frame

23.10.3.1 Priority Interrupt Bit

Figure 23.67 shows an example of operation in Start Frame reception where a priority interrupt bit is in use. Setting the CR1.PIBE bit to 1 enables the use of a priority interrupt bit.
Operations of the extended serial mode control section in start Frame reception where a priority interrupt bit is in use are as described below.

Steps (1) to (4) are the same as in Figure 23.63, for Start Frame reception.

(5) If the value of the bit selected by the CR1.PIBS[2:0] bits matches the corresponding bit in the PCF1DR register, the STR.PIBDF flag is set to 1. An SCIX1 interrupt is also generated if the value of the ICR.PIBDIE bit is 1. Transfer of the Information Frame starts after that. If the data received in Control Field 1 do not match the data set in either or both of registers PCF1DR and SCF1DR and the priority interrupt bit is not detected, a transition to the state prior to Break Field low width detection proceeds.



The above diagram assumes the following:

ESMER: ESME = 1
CR1: BFE = 1, PIBE = 1, CF0RE = 0, CF1DS[1:0] = 10b, PIBS[2:0] = 011b
PCR: RXDXPS = 0
ICR: BFDIE = 1, CF0MIE = 1, PIBDIE = 1
TMR: TOMS[1:0] = 01b

Figure 23.67 Example of Operations When Receiving a Start Frame While the CR1.PIBE Bit is 1

23.10.4 Detection of Bus Collisions

Detection of bus collisions operate for cases where output of the Break Field low width and transmission of data are in progress when the ESMER.ESME bit and the SCI.TE bit are set to 1.

Figure 23.68 shows an example of operations with bus collision detection. Signals output through TXDX12 and input through RXDX12 are sampled with the bus collision detection clock set with the CR2.BCCS[1:0] bits as the sampling clock, and the STR.BCDF flag is set to 1 if the signals fail to match three times in a row. An SCIX2 interrupt is also generated if the value of the ICR.BCDIE bit is 1.

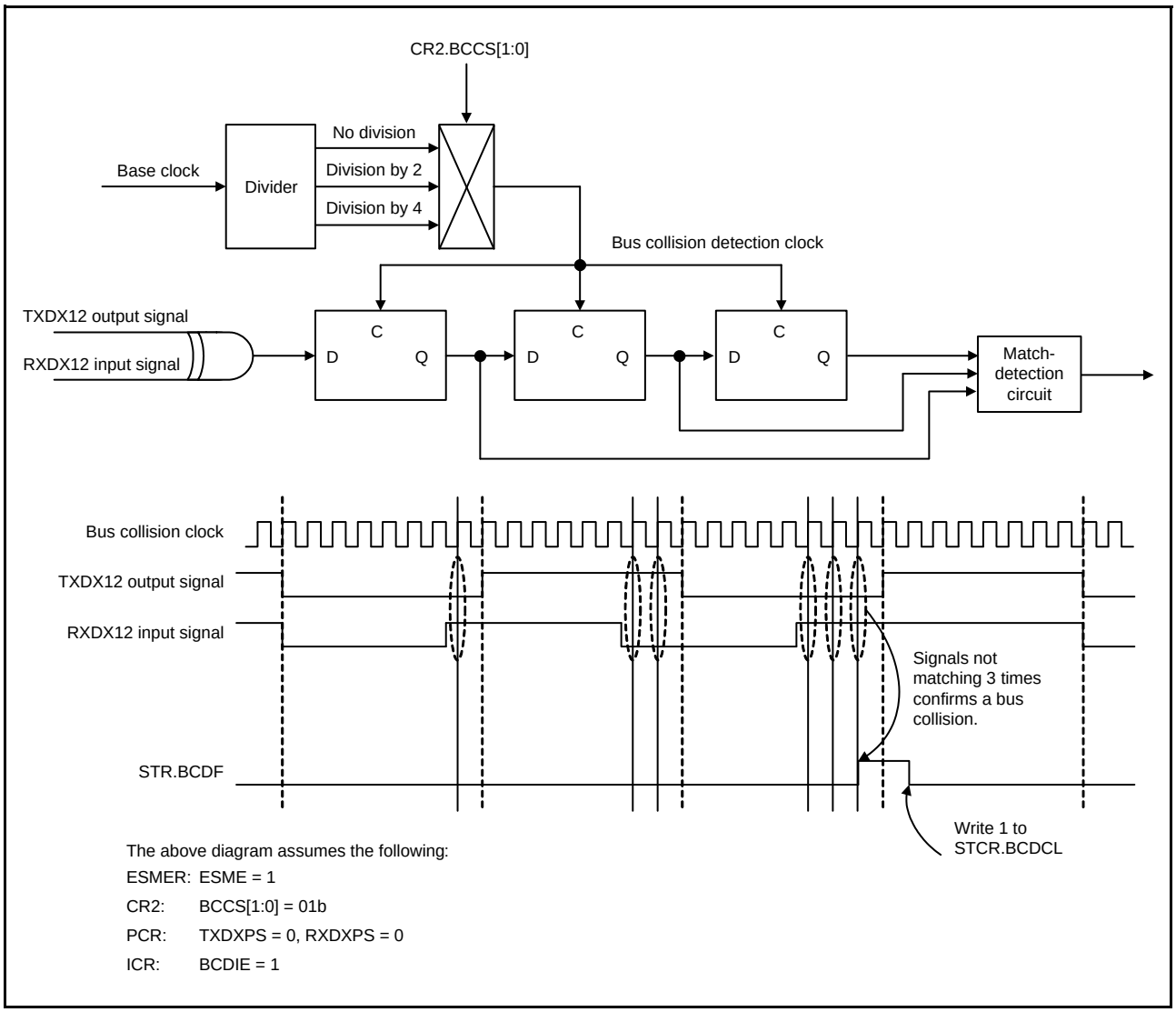


Figure 23.68 Example of Operations with Bus Collision Detection

23.10.5 Digital Filter for Input on the RXDX12 Pin

Signals input through the RXDX12 pin can be passed through a digital filter before they are conveyed to the internal circuits. The digital filter consists of three flip-flop circuit stages connected in series and a match-detecting circuit. The CR2.DFCS[2:0] bits select the sampling clock for the RXDX12 pin input signals. If the outputs of all three latches match, the given level is conveyed to subsequent circuits. If the levels do not match, the previous value is retained. In other words, levels are confirmed as being the signal if they are retained for at least three cycles of the sampling clock but judged to be noise rather than changes in the signal level if they change within three cycles of the sampling clock. Figure 23.69 shows an example of operations with the digital filter.

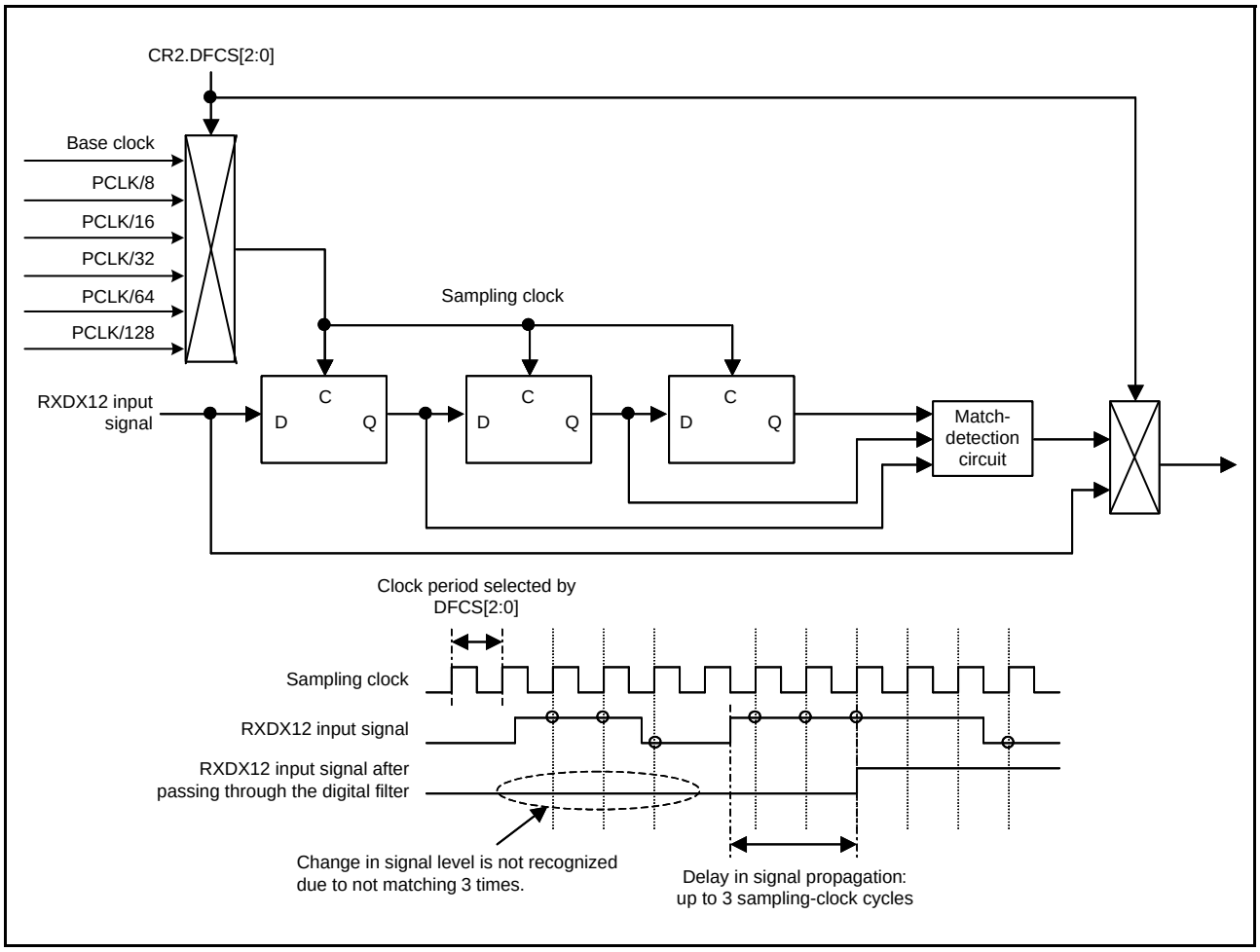
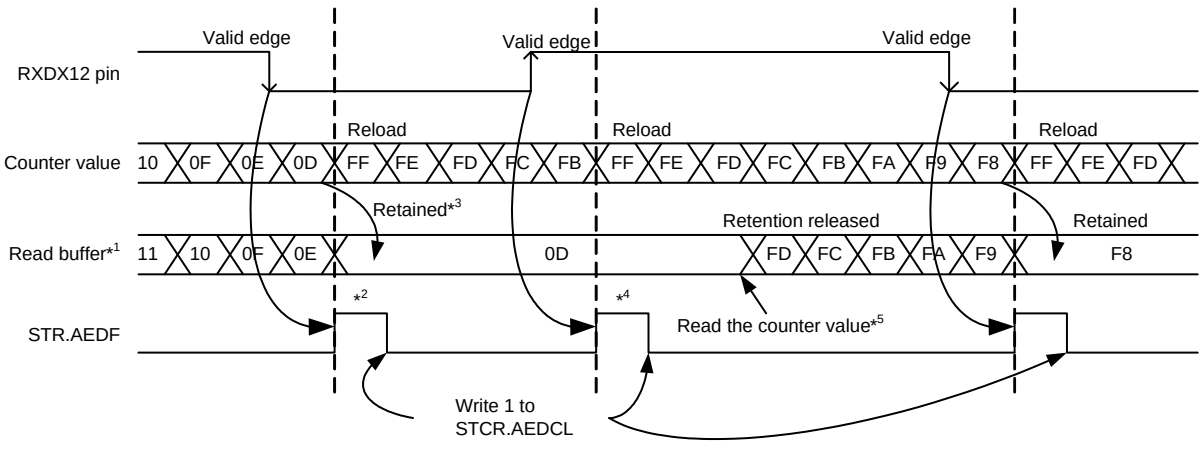


Figure 23.69 Example of Operations with the Digital Filter

23.10.6 Bit Rate Measurement

The bit rate measurement function measures the intervals between rising and falling edges and between falling and rising edges of the signal input from the RXDX12 pin. Figure 23.70 shows an example of operations for bit rate measurement.

(1) Writing 1 to the CR0.BRME bit enables bit rate measurement. Only set the BRME bit to 1 when you wish to proceed with bit rate measurement. Furthermore, bit rate measurement will not proceed during a Break Field, even if the BRME bit is set to 1.
(2) After detection of the Break Field low width, bit rate measurement starts when the level input on the RXDX12 pin becomes high.
(3) Once bit rate measurement has started, counter values from the timer are retained in the read buffers on the input of valid edges from the RXDX12 pin (rising and falling edges) and the counter is reloaded. An SCIX3 interrupt is also generated if the value of the ICR.AEDIE bit is 1. Retention by registers TCNT and TPRE is released by reading these registers.
(4) The bit rate as calculated from the values counted during intervals between valid edges can be used for adjusting the rate by changing the settings of the BRR register. To disable the bit rate measurement after a match with Control Field 1, write 0 to the CR0.BRME bit.



Note 1. The TCNT or TPRE register indicates the contents of the read buffers.
Note 2. The STR.AEDF flag becomes 1 on detection of a valid edge.
Note 3. When an effective edge is detected, the current value of the counter is stored in the read buffer.
Note 4. The value retained in the read buffers due to one valid edge is not updated to the new value even if a further valid edge is detected.
Note 5. Retention of a value in a read buffer due to a valid edge is released by reading the TCNT or TPRE register.

Figure 23.70 Example of Operations for Bit Rate Measurement

23.10.7 Selectable Timing for Sampling Data Received through RXDX12

The extended serial mode control section provides a way of adjusting the timing for the sampling of data received through the RXDX12 pin by setting the CR2.RTS[1:0] bits to select the rising edges of 8th, 10th, 12th, or 14th cycle of the base clock. If the value of the SEMR.ABCS bit is 1, the bits select the rising edges of 4th, 5th, 6th, or 7th cycle of the base clock. Figure 23.71 shows timing for the sampling of data received through RXDX12.

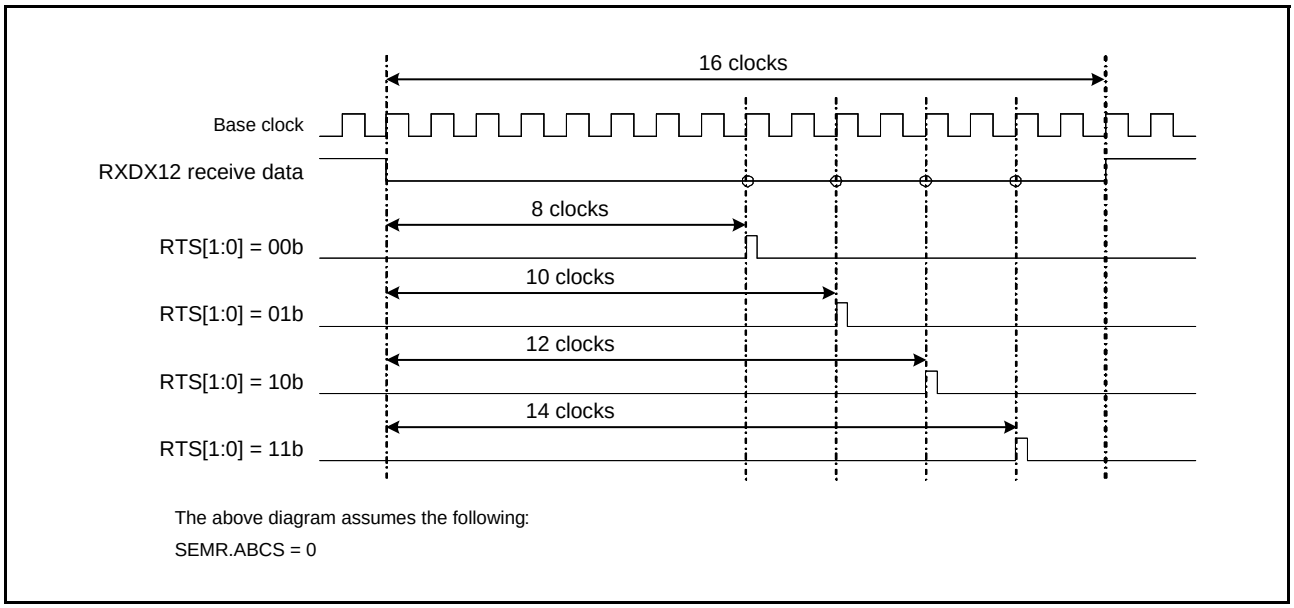


Figure 23.71 Timing for Sampling of Data Received through RXDX12

23.10.8 Timer

The timer has the following operating modes.

(1) Break Field Low Width Output Mode

This mode is for output through the TXDX12 pin of the low level over the Break Field low width at the transmission of a Start Frame. Setting the TMR.TOMS[1:0] bits to 10b switches operation to Break Field low width output mode. The TMR.TCSS[2:0] bits select the clock source for the counter. When the TCR.TCST bit is set to 1, the output on the TXDX12 pin goes to the low level and counting starts. When the timer underflows, the output on the TXDX12 pin goes to the high level and the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1. When 0 is written to the TCR.TCST bit, counting stops after reloading of registers TPRE and TCNT. After output of the Break Field low width is completed, stop the timer before it underflows again. Figure 23.72 shows an example of operations in Break Field low width output mode.

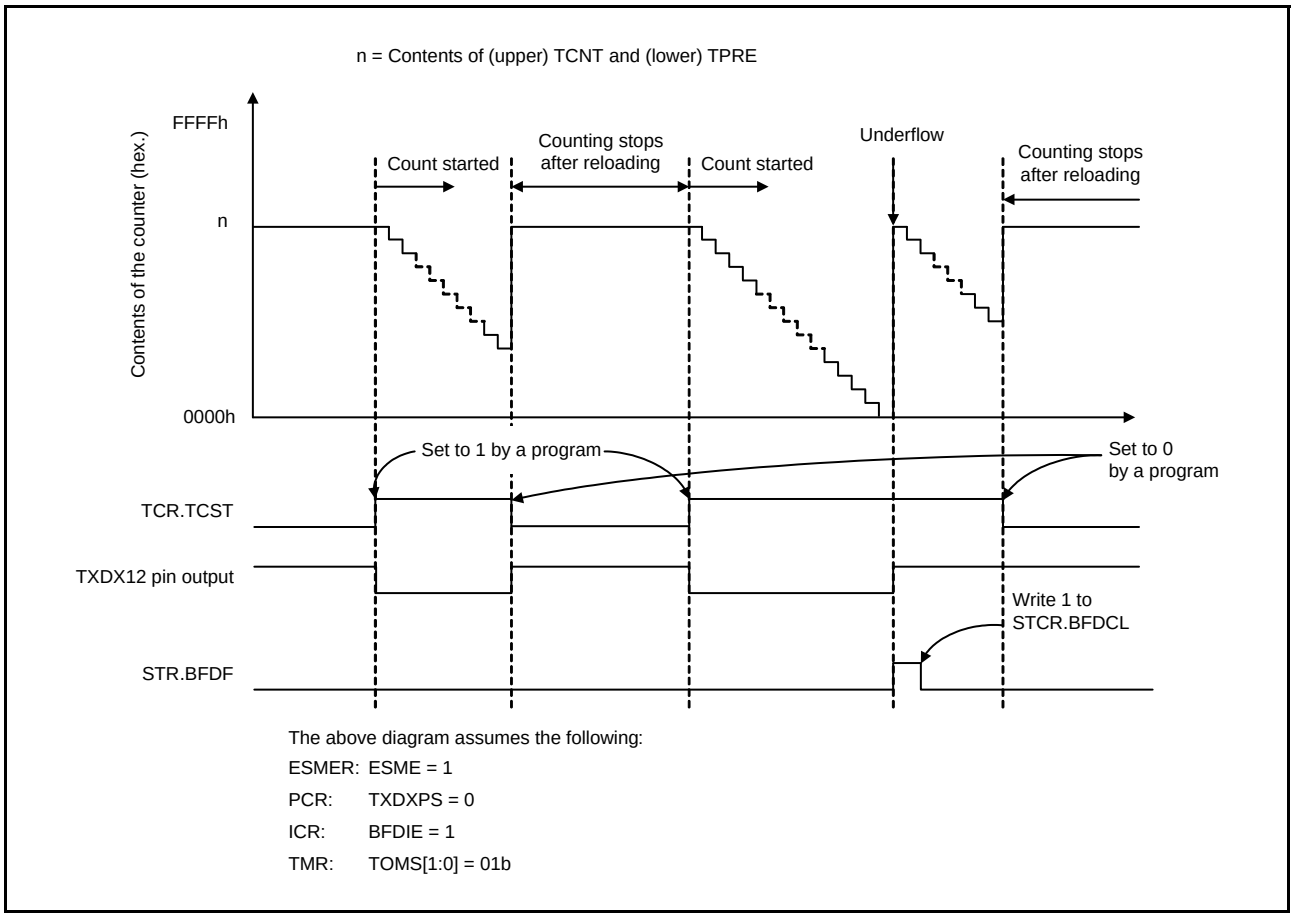


Figure 23.72 Example of Operations in Break Field Low Width Output Mode

(2) Break Field Low Width Determination Mode

This mode is for determining the Break Field low width in the input signal on the RXDX12 pin at the reception of a Start Frame. Setting the TMR.TOMS[1:0] bits to 01b switches operation to Break Field low width determination mode. The TMR.TCSS[2:0] bits select the clock source for the counter. When the TCR.TCST bit is set to 1, the interface enters the Break Field low width determinable state. Determination starts when a low level is input from the RXDX12 pin. When a high level is then input on the RXDX12 pin, registers TPRE and TCNT are reloaded and the interface enters the Break Field low width determinable state. When the timer underflows during Break Field low width determination, the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1. If an underflow of the timer during data transfer cause a problem in the form of interrupt generation, stop the timer after Break Field low width determination. Figure 23.73 shows an example of operations in Break Field low width output mode.

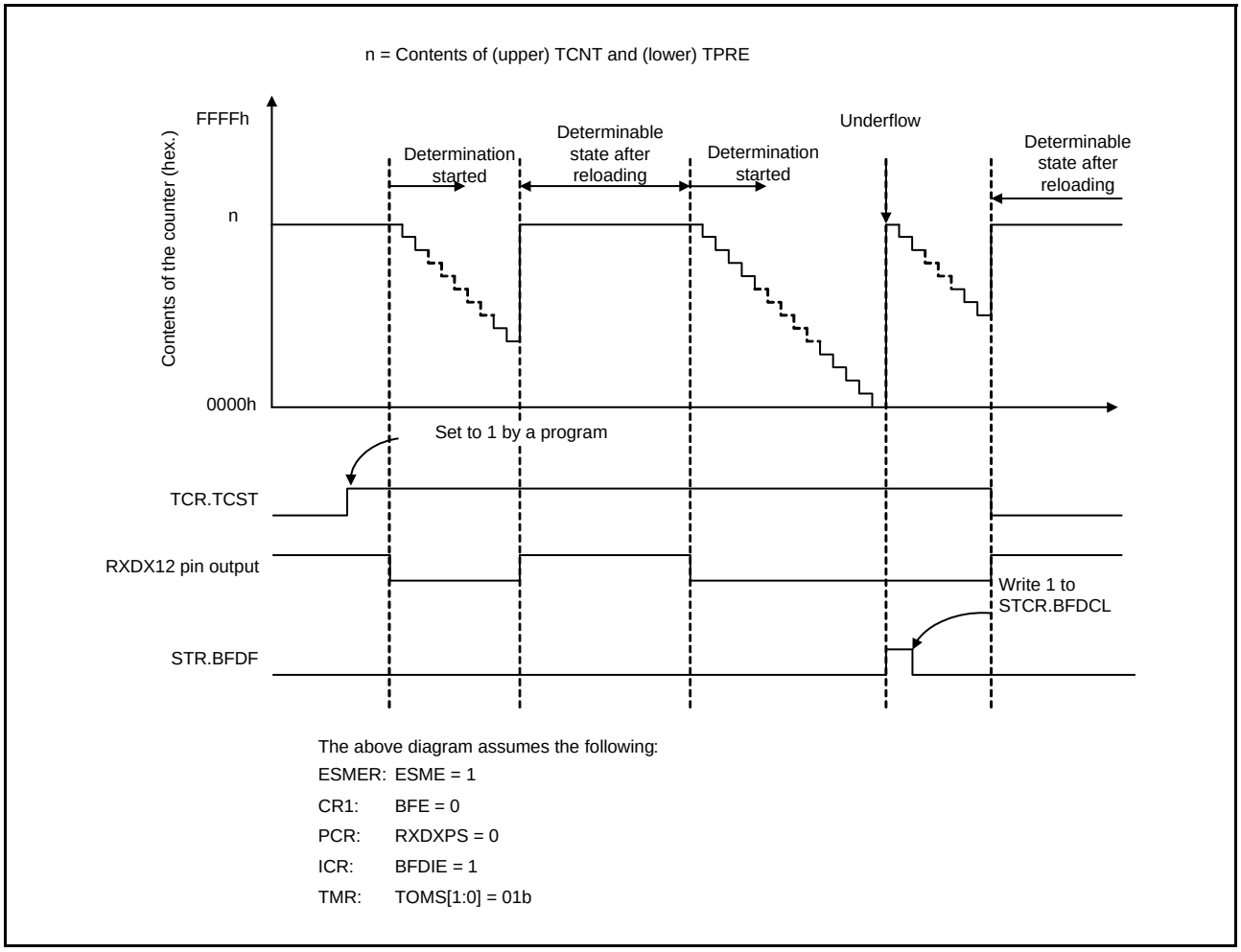


Figure 23.73 Example of Operations in Break Field Low Width Determination Mode

(3) Timer Mode

This mode is for counting cycles of the internal clock as the clock source. Setting the TMR.TOMS[1:0] bits to 00b switches operation to timer mode. The TMR.TCSS[2:0] bits select the clock source for the counter. Counting starts when 1 is written to the TCR.TCST bit and stops when 0 is written to the TCST bit. Registers TPRE and TCNT both count down. The TPRE register counts cycles of the clock source for counting, and underflows of the TPRE register provide the clock source for counting by the TCNT register. When the timer underflows, the STR.BFDF flag is set to 1. An SCIX0 interrupt is also generated if the value of the ICR.BFDIE bit is 1.

23.11 Noise Cancellation Function

Figure 23.74 shows the configuration of the noise filter used for noise cancellation. The noise filter consists of two stages of flip-flop circuits and a match-detection circuit. When the level on the pin matches in three consecutive samples taken at the set sampling interval, the matching level continues to be conveyed internally until the level on the pin again matches in three consecutive samples.

In asynchronous mode, the noise cancellation function can be applied on the RXDn input signal. The period of the base clock (1/16th of a bit-period when SEMR.ABCS = 0 and 1/8th of a bit-period when SEMR.ABCS = 1) is the sampling interval.

In simple I^2C mode, the noise cancellation function can be applied on the SSDAn and SSCLn input signals. The sampling clock is the clock signal produced by frequency-dividing the signal from the clock source for the internal baud-rate generator by one, two, four, or eight as selected by the setting of the SNFR.NFCS[2:0] bits.

If the base clock is stopped with the noise filter enabled and then the clock input is started again, the noise filter operation resumes from where the clock was stopped. If SCR.TE and SCR.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 to the 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 samples.

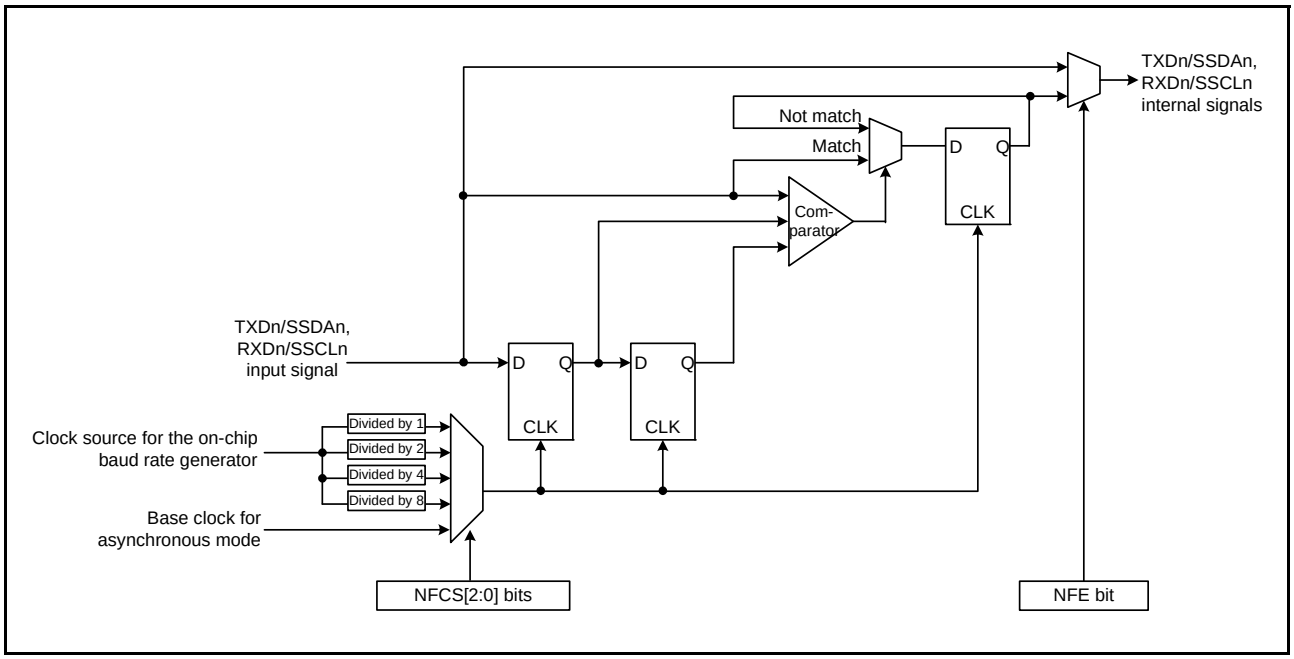


Figure 23.74 Block Diagram of Digital Noise Filter

23.12 Interrupt Sources

23.12.1 Buffer Operations for TXI and RXI Interrupts

If the conditions for a TXI and RXI interrupt are satisfied while the interrupt status flag in the interrupt controller is 1, the SCI does not output the interrupt request but retains it internally (with a capacity for retention of one request per source). When the value of the interrupt status flag in the interrupt controller becomes 0, the interrupt request retained within the SCI is output. The internally retained interrupt request is automatically discarded once the actual interrupt is output. Clearing of the corresponding interrupt enable bit (the TIE or RIE bit in the SCR register) can also be used to discard an internally retained interrupt request.

23.12.2 Interrupts in Asynchronous Mode, Clock Synchronous Mode, and Simple SPI Mode

Table 23.31 lists interrupt sources in asynchronous mode, clock synchronous mode, and simple SPI mode. Individual interrupt sources can be enabled or disabled with the enable bits in the SCR register.

If the SCR.TIE bit is 1, a TXI interrupt request is generated when transmit data is transferred from the TDR or TDRL register*1 to the TSR register. A TXI interrupt request can also be generated by setting the SCR.TE bit to 1 after setting the SCR.TIE bit to 1 or by using a single instruction to set the SCR.TE and SCR.TIE bit to 1 at the same time. A TXI interrupt request can activate the DTC to handle data transfer.

A TXI interrupt request is not generated by setting the SCR.TE bit to 1 while the setting of the SCR.TIE bit is 0 or by setting the SCR.TIE bit to 1 while the setting of the SCR.TE bit is 1.*2

Note that setting the SCR.TE bit to 0 while the SCR.TIE bit is 1 leads to the generation of a TXI interrupt.

When new data is not written by the time of transmission of the last bit of the current transmit data and the setting of the SCR.TEIE bit is 1, the SSR.TEND flag becomes 1 and a TEI interrupt request is generated. Furthermore, when the setting of the SCR.TE bit is 1, the SSR.TEND flag retains the value 1 until further transmit data are written to the TDR or TDRL register*1, and setting the SCR.TEIE bit to 1 leads to the generation of a TEI interrupt request.

Writing data to the TDR or TDRL register*1 leads to clearing of the SSR.TEND flag and, after a certain time, discarding of the TEI interrupt request.

If the SCR.RIE bit is 1, an RXI interrupt request is generated when received data is stored in the RDR register. An RXI interrupt request can activate the DTC to handle data transfer.

Setting of any from among the ORER, FER, and PER flags in the SSR register to 1 while the SCR.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, and PER) leads to discarding of the ERI interrupt request.

Note 1. In the case where asynchronous mode and 9-bit data length are selected

Note 2. To temporarily disable 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 transmission-completed interrupt, control disabling and enabling of the interrupt by using the interrupt request enable bit in the interrupt controller rather than using the SCR.TIE bit. This can prevent the suppression of TXI interrupt requests in the transfer of new data.

Table 23.31 Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DTC Activation
ERI	Receive error	ORER, FER, or PER	Not possible
RXI	Receive data full	RDRF	Possible
TXI	Transmit data empty	TDRE	Possible
TEI	Transmit end	TEND	Not possible

23.12.3 Interrupts in Smart Card Interface Mode

Table 23.32 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

Table 23.32 SCI Interrupt Sources

Name	Interrupt Source	Interrupt Flag	DTC Activation
ERI	Receive error or error signal detection	ORER, PER, or ERS	Not possible
RXI	Receive data full	—	Possible
TXI	Transmit data empty	TEND	Possible

Data transmission/reception using the DTC is also possible in smart card interface mode, similar to in the normal SCI mode. In transmission, when the SSR.TEND flag is set to 1, a TXI interrupt request is generated. This TXI interrupt request activates the DTC allowing transfer of transmit data if the TXI request is specified beforehand as a source of DTC activation. The TEND flag is automatically set to 0 when the DTC 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 DTC is not activated. Therefore, the SCI and DTC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, the SSR.ERS flag is not automatically cleared to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the SCR.RIE bit to 1 to enable an ERI interrupt request to be generated at error occurrence.
When transmitting/receiving data using the DTC, be sure to make settings to enable the DTC before making SCI settings. For DTC settings, refer to section 16, Data Transfer Controller (DTCb).
In reception, an RXI interrupt request is generated when receive data is set to the RDR register. This RXI interrupt request activates the DTC allowing transfer of receive data if the RXI request is specified beforehand as a source of DTC activation. If an error occurs, the error flag is set. Therefore, the DTC is not activated and an ERI interrupt request is issued to the CPU instead; the error flag must be cleared.

23.12.4 Interrupts in Simple I2C Mode

The interrupt sources in simple I2C mode are listed in Table 23.33. The STI interrupt is allocated to the transmit end interrupt (TEI) request. The receive error interrupt (ERI) request cannot be used.

The DTC can also be used to handle transfer in simple I2C mode.

When the value of the SIMR2.IICINTM bit is 1, an RXI request will be generated on the falling edge of the SSCLn signal for the eighth bit. If the RXI has been set up as an activating request for the DTC beforehand, the RXI request will activate the DTC to handle transfer of the received data. Furthermore, a TXI request is generated on the falling edge of the SSCLn signal for the ninth bit (acknowledge bit). If the TXI has been set up as an activating request for the DTC beforehand, the TXI request will activate the DTC to handle transfer of the transmit data.

When the value of the SIMR2.IICINTM bit is 0, an RXI request (ACK detection) if the input on the SSDAn pin is at the low level or a TXI request (NACK detection) if the input on the SSDAn pin is at the high level will be generated on the rising edge of the SSCLn signal for the ninth bit (acknowledge bit). If the RXI has been set up as an activating request for the DTC beforehand, the RXI request will activate the DTC to handle transfer of the received data.

Also, if the DTC is used for data transfer in reception or transmission, be sure to set up and enable the DTC before setting up the SCI.

When the IICSTAREQ, IICRSTAREQ, and IICSTPREQ bits in the SIMR3 register are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

Table 23.33 SCI Interrupt Sources

Name	Interrupt Source		Interrupt Flag	DTC Activation
	IICINTM bit = 0	IICINTM bit = 1		
RXI	ACK detection	Reception	—	Possible
TXI	NACK detection	Transmission	—	Possible*1
STI	Completion of generation of a start, restart, or stop condition		IICSTIF	Not possible

Note 1. Activation of the DTC is only possible when the SIMR2.IICINTM bit is 1 (use reception and transmission interrupts).

23.12.5 Interrupt Requests from the Extended Serial Mode Control Section

The extended serial mode control section has a total of six types of interrupt request for generating the SCIX0 interrupt (Break Field low width detected), SCIX1 interrupt (Control Field 0 match, Control Field 1 match, priority interrupt bit detected), SCIX2 interrupt (bus collision detected), and SCIX3 interrupt (valid edge detected). When any of the interrupt factors is generated, the corresponding status flag is set to 1. Details of all of the interrupt requests are listed in Table 23.34.

Table 23.34 Interrupt Sources of the Extended Serial Mode Control Section

Interrupt Request	Status Flag	Interrupt Factors
SCIX0 interrupt (Break Field low width detected)	BFDF	• Detection of a Break Field low width longer than the interval corresponding to the timer setting • Completion of the output of a Break Field low width over the interval corresponding to the timer setting • Underflow of the timer
SCIX1 interrupt (Control Field 0 match)	CF0MF	The data received in Control Field 0 matching the value set in the CF0DR register
SCIX1 interrupt (Control Field 1 match)	CF1MF	The data received in Control Field 1 matching the value set in the PCF1DR or SCF1DR register
SCIX1 interrupt (priority interrupt bit detected)	PIBDF	The value of the bit specified as the priority interrupt bit matching the value set in the PCF1DR register
SCIX2 interrupt (bus collision detected)	BCDF	The output level on the TXDX12 pin and the input level on the RXDX12 pin not matching on three consecutive cycles of the bus collision detection clock
SCIX3 interrupt (valid edge detected)	AEDF	Detection of a valid edge during bit rate measurement

23.13 Usage Notes

23.13.1 Setting the Module Stop Function

Module stop control register B (MSTPCRB) is used to stop and start SCI operations. With the value after a reset, SCI operations are stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

23.13.2 Break Detection and Processing

When a framing error is detected, a break can be detected by reading the RXDn pin value directly. In a break, the input from the RXDn pin becomes all 0s, and so the SSR.FER flag is set to 1 (framing error has occurred), and the SSR.PER flag may also be set to 1 (parity error has occurred). When the SEMR.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 SEMR.RXDESEL bit is 1, the SCI sets the SSR.FER flag to 1 and stops receiving operation until a start bit of the next data frame is detected. If the SSR.FER flag is set to 0 at this time, the SSR.FER flag retains 0 during the break. When the RXDn pin becomes high 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.

23.13.3 Mark State and Sending Breaks

When the SCR.TE bit is 0 (serial transmission is disabled), the TXDn pin becomes high-impedance. To forcibly set the TXDn pin to mark or space state while the TE bit is 0, set the I/O port associated registers and switch the TXDn pin to general output port.
For holding the communication line in the mark (“1”) state until the TE bit is set to 1 (serial transmission is enabled), set the corresponding bit in the PODR register to 1 for high output from general output port. To start communications, set the TE bit to 1 and then the corresponding bit in the PMR register to 1.
To send a break (the space state for longer than a certain period of time) while data transmission, set the corresponding bit in the PODR register to 0 (low output), and set the corresponding bit in the PMR register to 0 (general I/O port). Then set the TE bit to 0 if necessary. When the TE bit is set to 0, the transmitter is initialized regardless of the current transmit status.

23.13.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode and Simple SPI Mode)

Transmission cannot be started when a receive error flag (SSR.ORER) is set to 1, even if data is written to the TDR register. Be sure to set the receive error flags to 0 before starting transmission. Note also that the receive error flags cannot be set to 0 even if the SCR.RE bit is set to 0 (serial reception is disabled).

23.13.5 Writing Data to the TDR Register

Data can be written to registers TDR, TDRH, and TDRL. However, if new data is written to registers TDR, TDRH, and TDRL when transmit data is remaining in registers TDR, TDRH, and TDRL, the previous data in registers TDR, TDRH, and TDRL is lost because it has not been transferred to the TSR register yet. Be sure to write transmit data to registers TDR, TDRH, and TDRL in the TXI interrupt request handling routine.

23.13.6 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 the TDR register by the CPU or DTC and wait for at least five PCLK cycles before allowing the transmit clock to be input (refer to Figure 23.75).

(2) Continuous transmission

(a) Write the next transmit data to the TDR or TDRL register before the falling edge of the transmit clock (bit 7) (refer to Figure 23.75).

(b) When updating the TDR register after bit 7 has started to transmit, update the TDR register while the synchronization clock is in the low-level period, and set the high-level width of the transmit clock (bit 7) to four PCLK cycles or longer (refer to Figure 23.75).

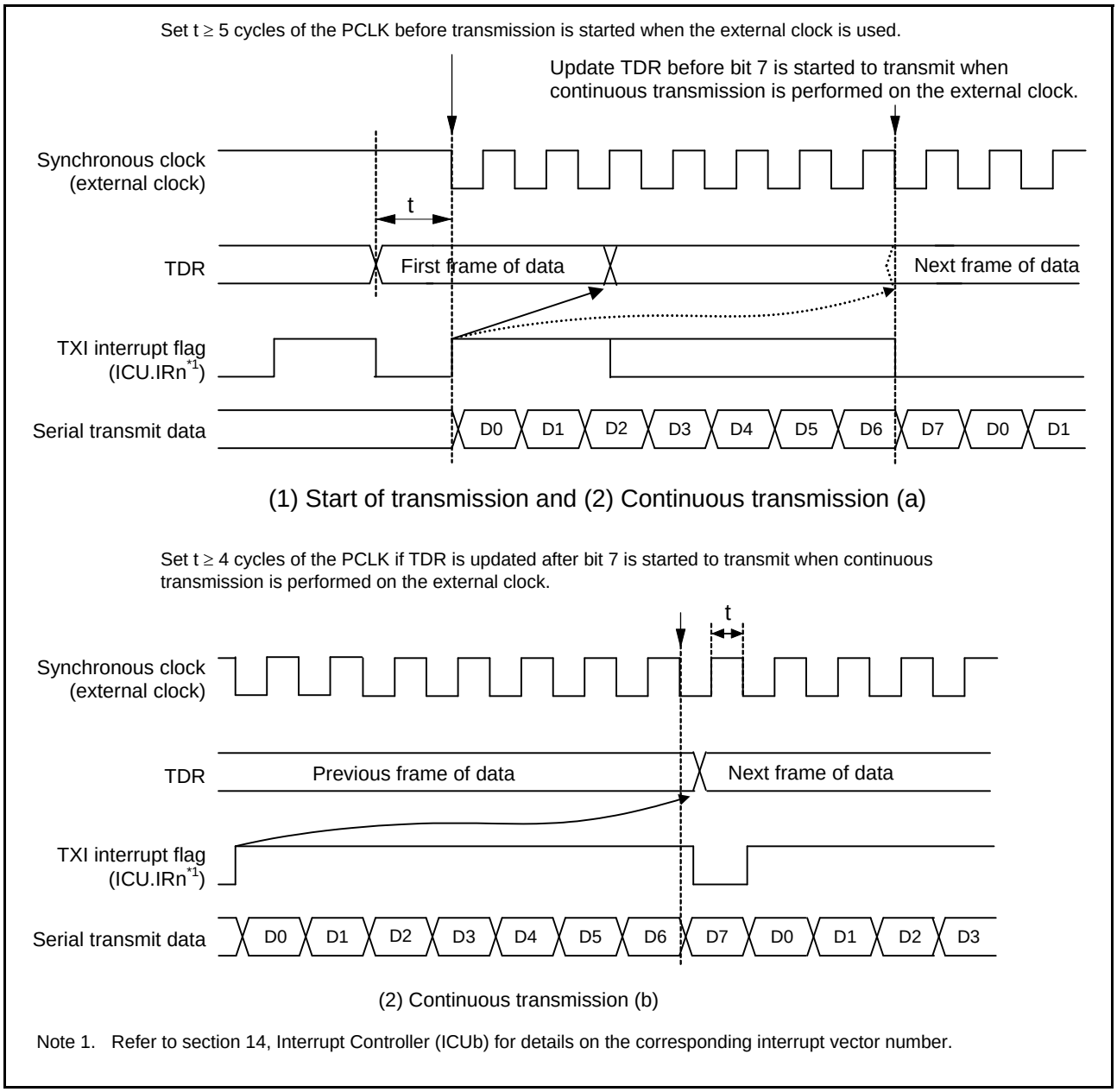


Figure 23.75 Restrictions on Use of External Clock in Clock Synchronous Transmission

23.13.7 Restrictions on Using DTC

When using the DTC to read the RDR, RDRH, and RDRL registers, be sure to set the receive data full interrupt (RXI) as the activation source of the relevant SCI.

23.13.8 Notes on Starting Transfer

At the point where transfer starts when the interrupt status flag (IRn.IR flag) in the interrupt controller is 1, follow the procedure below to clear interrupt requests before permitting operations (by setting the SCR.TE or SCR.RE bit to 1). For details on the interrupt status flag, refer to section 14, Interrupt Controller (ICUb).

- Confirm that transfer has stopped (the setting of the SCR.TE or SCR.RE bits is 0).
- Set the corresponding interrupt enable bit (SCR.TIE or SCR.RIE) to 0.
- Read the corresponding interrupt enable bit (SCR.TIE or SCR.RIE bit) to check that it has become 0.
- Set the interrupt status flag (IRn.IR flag) in the interrupt controller to 0.

23.13.9 SCI Operations during Low Power Consumption State

(1) Transmission

When making settings for the module stopped state or in transitions to software standby, stop operations (by setting the TIE, TE, and TEIE bits in the SCR register to 0) after switching the TXDn pin to the general I/O port pin function. Setting the TE bit to 0 resets the TSR register and the SSR.TEND flag. Depending on the port settings, output pins may output the level before a transition to the low power consumption state is made after release from the module stopped state or software standby mode. When transitions to these states are made during transmission, the data being transmitted become indeterminate.

To transmit data in the same transmit mode after cancellation of the low power consumption state, set the TE bit to 1, read the SSR register, and write data to the TDR register sequentially to start data transmission. To transmit data with a different transmit mode, initialize the SCI first.

Figure 23.76 shows a sample flowchart for transition to software standby mode during transmission. Figure 23.77 and Figure 23.78 show the port pin states during transition to software standby mode.

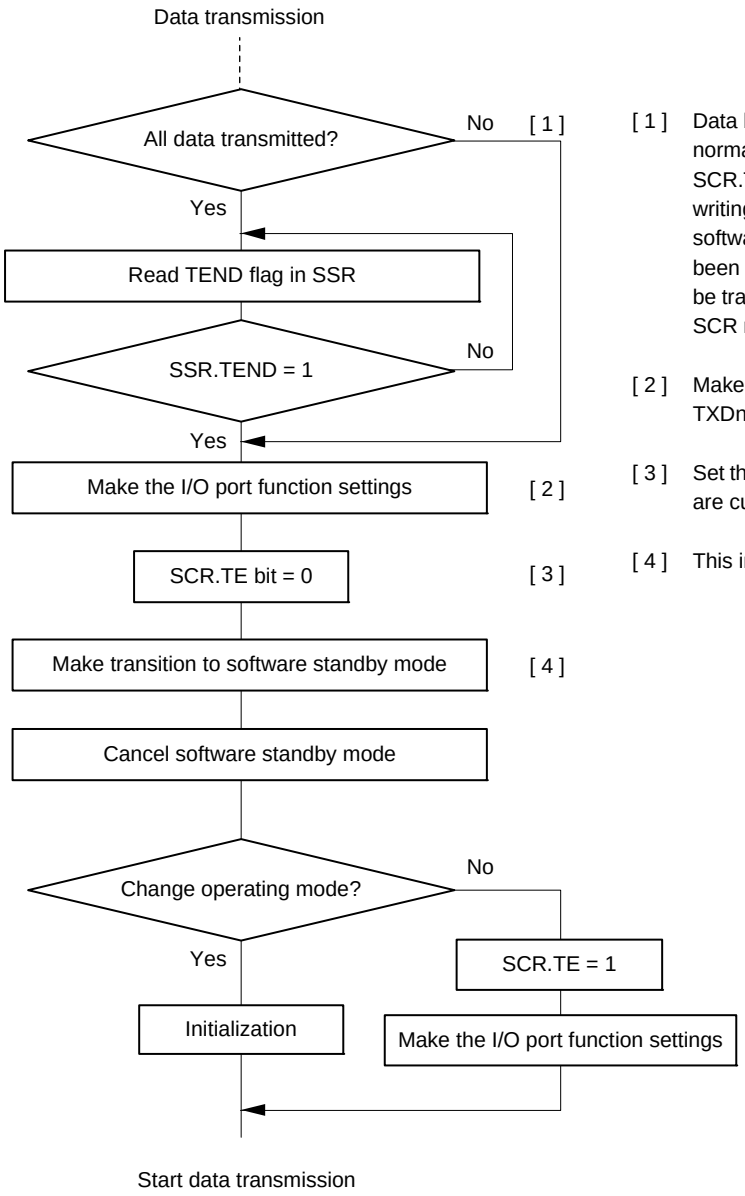
Before specifying the module stop state or making a transition to software standby mode from the transmit mode using DTC transfer, stop the transmit operations (TE = 0). To start transmission after cancellation using the DTC, set the TE and TIE bits to 1. The TXI interrupt flag is set to 1 and transmission starts using the DTC.

(2) Reception

Before specifying the module stop state or making a transition to software standby mode, stop the receive operations (SCR.RE = 0). If transition is made during data reception, the data being received will be invalid.

To receive data in the same receive mode after cancellation of the low power consumption state, set the RE bit to 1, and then start reception. To receive data in a different receive mode, initialize the SCI first.

Figure 23.79 shows a sample flowchart for transition to software standby mode during reception.



[1] Data being transmitted is lost halfway. Data can be normally transmitted from the CPU by setting the SCR.TE bit to 1, reading the SSR register, and writing data to the TDR register after canceling software standby mode. However, if the DTC has been activated, the data remaining in the DTC will be transmitted when both the TE and TIE bits in the SCR register are set to 1.

[2] Make the I/O port function settings to switch the TXDn pin to operate as a general I/O port.

[3] Set the TIE and TEIE bits in the SCR register if they are currently set to 1.

[4] This includes the setting for the module stop state.

Figure 23.76 Example of Flowchart for Transition to Software Standby Mode during Transmission

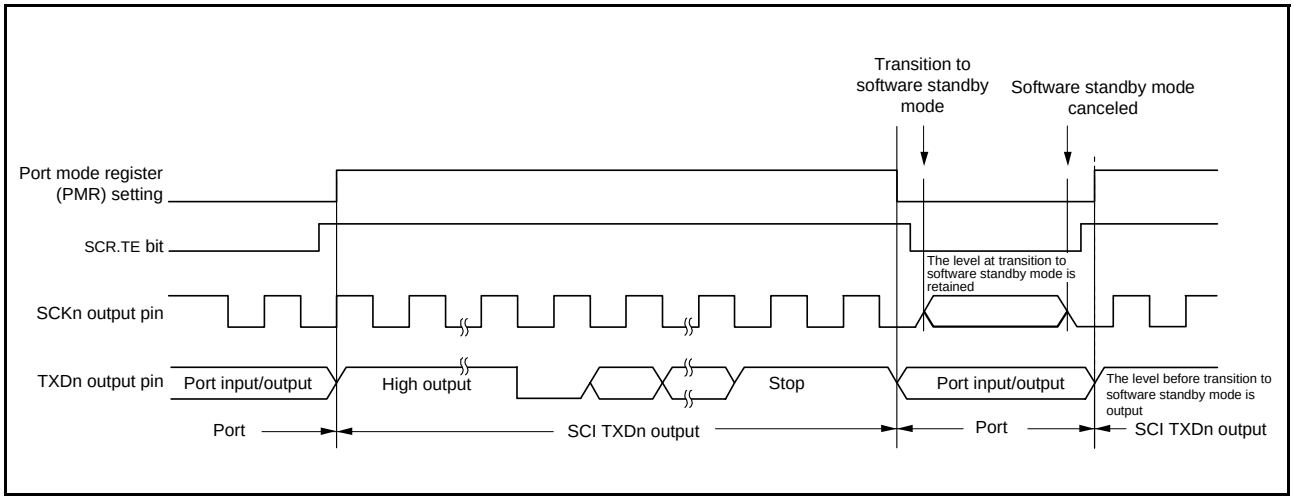


Figure 23.77 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)

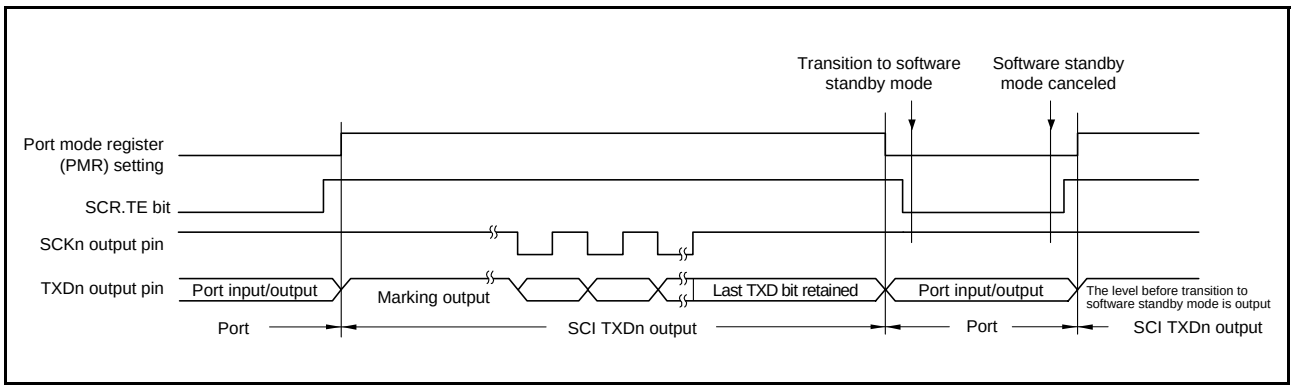


Figure 23.78 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)

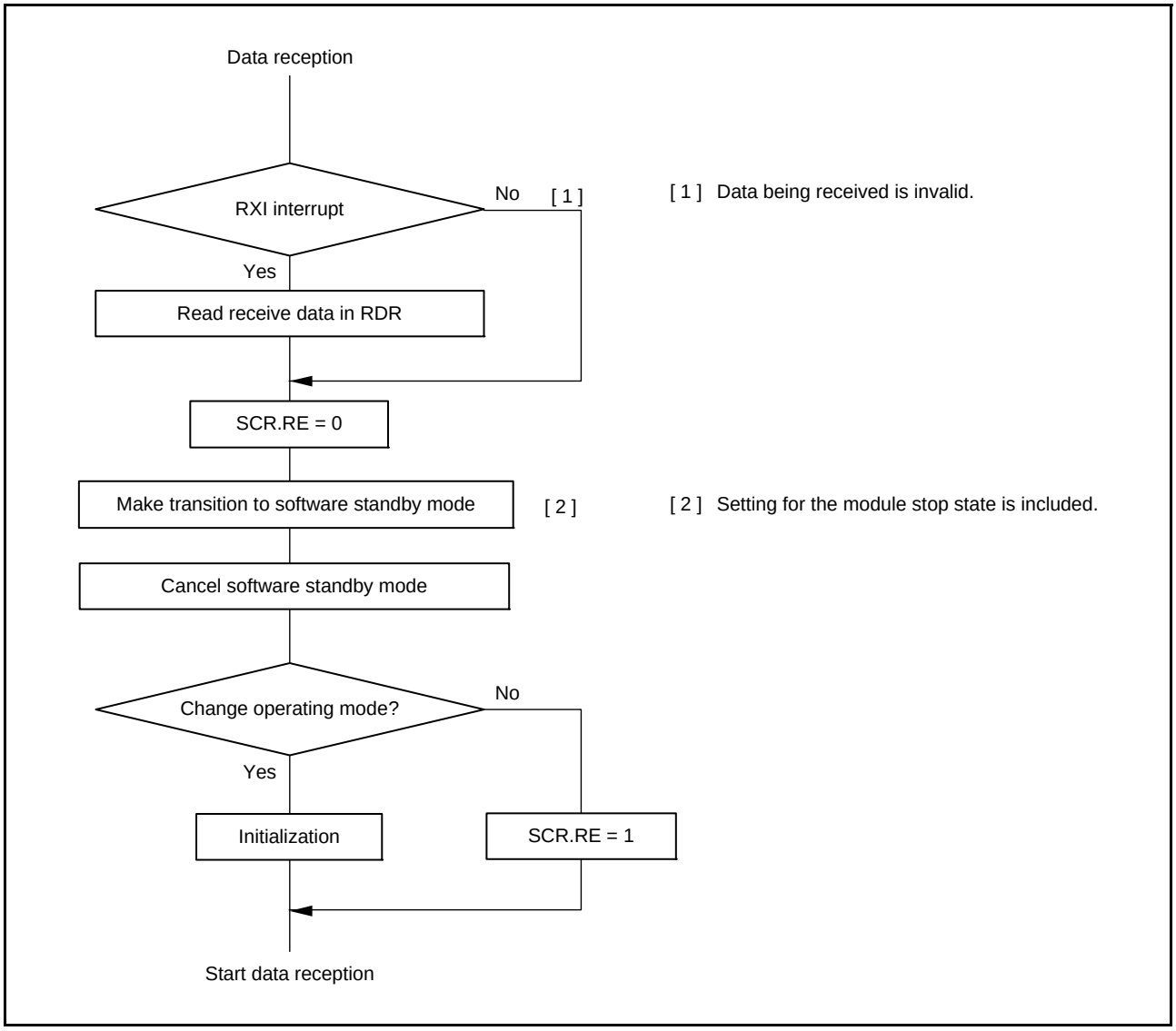


Figure 23.79 Example of Flowchart for Transition to Software Standby Mode during Reception

23.13.10 External Clock Input in Clock Synchronous Mode and Simple SPI Mode

In clock synchronous mode and simple SPI mode, the external clock SCKn must be input as follows:
High-pulse period, low-pulse period = 2 PCLK cycles or more, period = 6 PCLK cycles or more

23.13.11 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 SPMR.CKPH and CKPOL bits when the SPMR.SSE bit is 1.
  This prevents the clock line from being placed in the high-impedance state when the SCR.TE bit is set to 0 or unexpected edges from being generated on the clock line when the SCR.TE bit is changed from 0 to 1. When the SPMR.SSE bit is 0 in single master mode, pulling up or pulling down the clock line is not necessary because the clock line is not placed in the high-impedance state even when the SCR.TE bit is set to 0.
- In the case of the setting for clock delay (SPMR.CKPH bit is 1), the receive data full interrupt (RXI) is generated before the final clock edge on the SCKn pin as indicated in Figure 23.80. If the TE and RE bits in the SCR register 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 configuration, 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, 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.

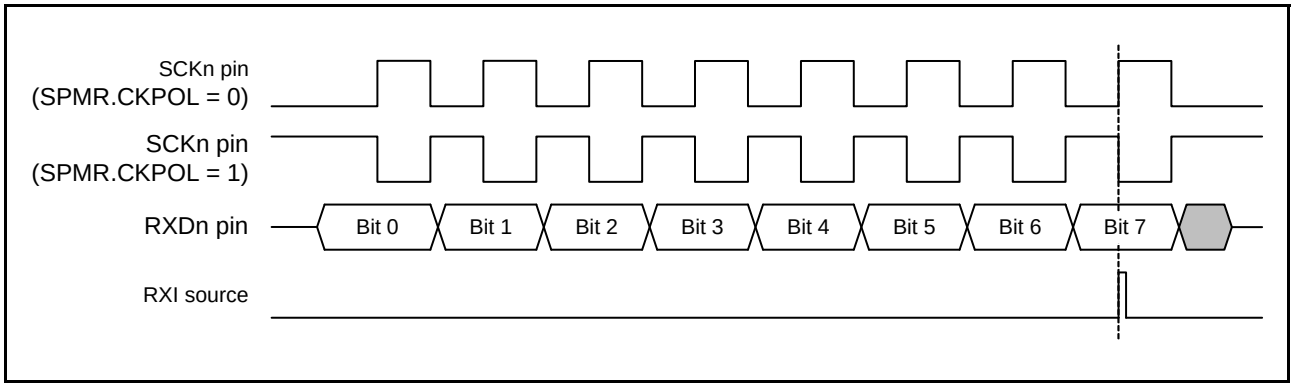


Figure 23.80 Timing of the RXI Interrupt in Simple SPI Mode (with Clock Delay)

(2) Slave Mode

- Secure at least five cycles of the PCLK from writing transmit data in the TDR register to start of the external clock input. Also secure at least five cycles of the PCLK from input of low level on the SSn# pin to start of the external clock input.
- Provide an external clock signal from the master the same as the transmit/receive data length.
- 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, set the TE and RE bits in the SCR register to 0 and, after remaking the settings, restart transfer of the first byte.

23.13.12 Limitation 1 on Usage of the Extended Serial Mode Control Section

When the PCR.SHARPS bit is set to 1, output on the TXDX12/RXDX12 pin is only possible when the following conditions apply.

- The timer is in Break Field low width output mode and the value of the TCR.TCST bit is 1 (when the TCST bit is set to 1, the high level continues to be output for up to one cycle of the clock source for counting by the timer counter before output of the low level)
- The value of the SCR.TE bit is 1.

23.13.13 Limitation 2 on Usage of the Extended Serial Mode Control Section

The TXI, RXI, ERI, and TEI interrupt requests are generated even if the extended serial mode is enabled. However, the RXI interrupt should not be enabled during reception of a Start Frame because the extended serial mode control section uses the receive data full signal.
To use the RXI interrupts during a reception of the Information Frame, use it in accordance with one of the following procedures. When a receive error is detected, clear the receive error flag and initialize the extended serial mode control section.

(1) Set the SCR.RIE bit to 0 to disable the output of interrupt requests. Check the error flags in the SSR register on completion of the reception of a Start Frame, because an ERI interrupt is not generated if a receive error occurs. After reception of the Start Frame is completed, set the SCR.RIE bit to 1 by the time the first byte of the Information Frame is received.

(2) Set the SCR.RIE bit to 1 to disable RXI interrupts and enable ERI interrupts for ICU.
Clear the IRn.IR flag to enable the acceptance of RXI interrupts by ICU by the time the first byte of the Information Frame is received after the completion of Start Frame reception.

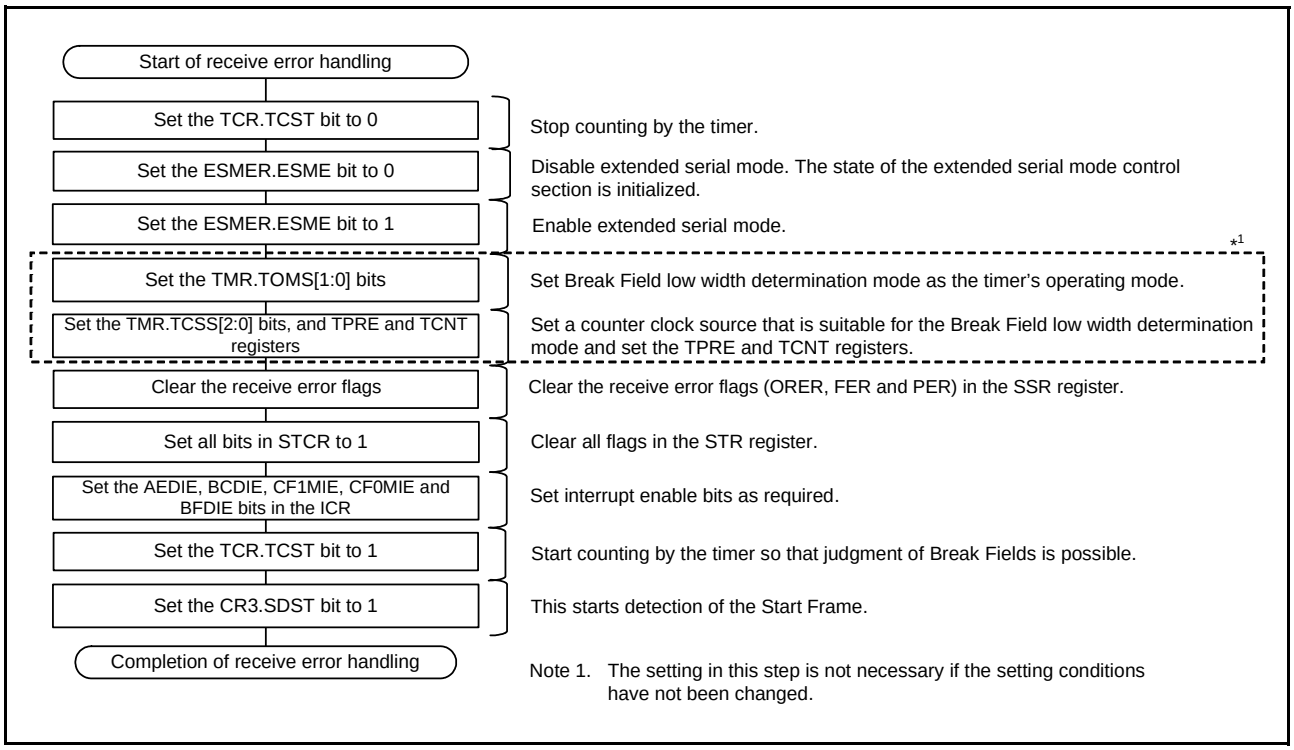


Figure 23.81 Example of Flowchart for Receive Error Handling (during Reception of the Start Frame)

23.13.14 Note on Transmit Enable Bit (TE Bit)

When setting the pin function to “TXDn” while the SCR.TE bit is 0 (serial transmission is disabled) or setting the TE bit to 0 while the pin function is “TXDn”, output of the TXDn pin becomes high-impedance.

Prevent the TXDn line from becoming high-impedance by any of the following ways:

(1) Connect a pull-up resistor to the TXDn line.

(2) Set the TE bit to 1*1 before changing the pin function to “TXDn”. Change the pin function to “general-purpose I/O port, output” before setting the TE bit to 0.

Note 1. An interrupt is generated when the TE bit is set to 1 while the TXI interrupt is enabled. If this creates a problem, change the pin function to “TXDn” first, and then set the corresponding ICU.IERm.IENj bit to 1.

23.13.15 Note on Stopping Reception When Using the RTS Function in Asynchronous Mode

One clock cycle of PCLK is required for the time from setting the SCR.RE bit to 0 to stopping the RTS signal generator in asynchronous mode.

When reading the RDR (or RDRL) register after setting the SCR.RE bit to 0, confirm that the RE bit has been set to 0 before reading the RDR (or RDRL) register to prevent these two processes from being performed consecutively.

24. I2C-bus Interface (RIICa)

This MCU has a single-channel I2C-bus interface (RIIC).
The RIIC module conforms with the NXP I2C-bus (Inter-IC bus) interface and provides a subset of its functions.
In this section, “PCLK” is used to refer to PCLKB.

24.1 Overview

Table 24.1 lists the specifications of the RIIC, Figure 24.1 shows a block diagram of the RIIC, and Figure 24.2 shows an example of I/O pin connections to external circuits (I2C-bus configuration example). Table 24.2 lists the I/O pins of the RIIC.

Table 24.1 RIIC Specifications (1/2)

Item	Description
Communications format	• I2C-bus format or SMBus format • Master mode or slave mode selectable • Automatic securing of the various setup times, hold times, and bus-free times for the transfer rate
Transfer rate	Fast-mode is supported (up to 400 kbps)
Serial clock (SCL)	For master operation, the duty cycle of the SCL is selectable in the range from 4 to 96%.
Generating and detecting conditions	Start, restart, and stop conditions are automatically generated. Start conditions (including restart conditions) and stop conditions are detectable.
Slave address	• Up to three different slave addresses can be set. • 7-bit and 10-bit address formats are supported (along with the use of both at once). • General call addresses, device ID addresses, and SMBus host addresses are detectable.
Acknowledgment	• For transmission, the acknowledgment bit is automatically loaded. Transfer of the next data for transmission can be automatically suspended on detection of a not-acknowledge signal. • For reception, the acknowledgment bit is automatically transmitted. If a wait between the eighth and ninth clock pulses has been selected, software control of the acknowledgment in response to the received data is possible.
Wait function	• During reception, cycles of waiting by holding the SCL line low can be inserted at the following two types of timing: Waiting between the eighth and ninth clock pulses Waiting between the ninth clock pulse and the first clock pulse of the next byte
SDA output delay function	Changes in the timing of the output of data bits for transmission, and of the acknowledgment bit, can be delayed relative to the falling edge of SCL.
Arbitration	• For multi-master operation Clock synchronization for the SCL line in cases of conflict with the SCL signal from another master is possible. When generating the start condition would create conflict on the bus, loss in arbitration is detected by testing for non-matching between the internal data level and the actual level on the SDA line. During master operation, loss in arbitration is detected by testing for non-matching between the actual level on the SDA line and the internal data level. • Loss in arbitration due to detection of the start condition while the bus is busy is detectable (to prevent the generating of double start conditions). • Loss in arbitration in transfer of a not-acknowledge signal due to the internal signal (NACK) and the actual level on the SDA line not matching is detectable. • Loss in arbitration due to non-matching of internal data level and the actual level on the SDA line is detectable in slave transmission.
Timeout function	The internal timeout function is capable of detecting long-interval stop of the SCL.
Noise cancellation	The interface incorporates digital noise filters for both the SCL and SDA signals, and the width for noise cancellation by the filters is adjustable by software.
Interrupt sources	Four sources: • Error in transfer or occurrence of events Detection of arbitration-lost, NACK, timeout, a start condition including a restart condition, or a stop condition • Receive data full (including matching with a slave address) • Transmit data empty (including matching with a slave address) • Transmit end

Table 24.1 RIIC Specifications (2/2)

Item	Description
Low power consumption function	Module stop state can be set.
RIIC operating modes	• Four Master transmit mode, master receive mode, slave transmit mode, and slave receive mode

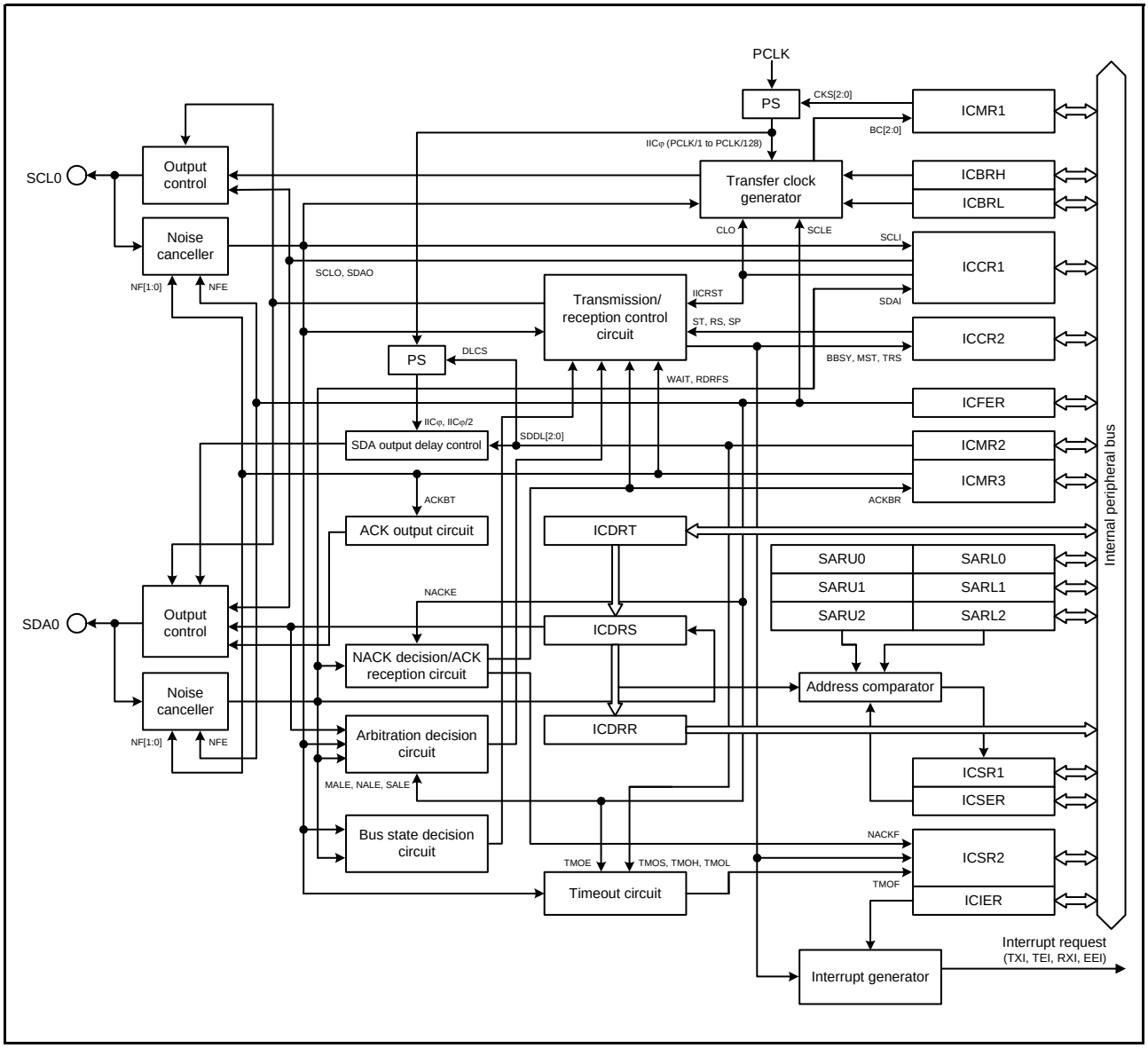


Figure 24.1 RIIC Block Diagram

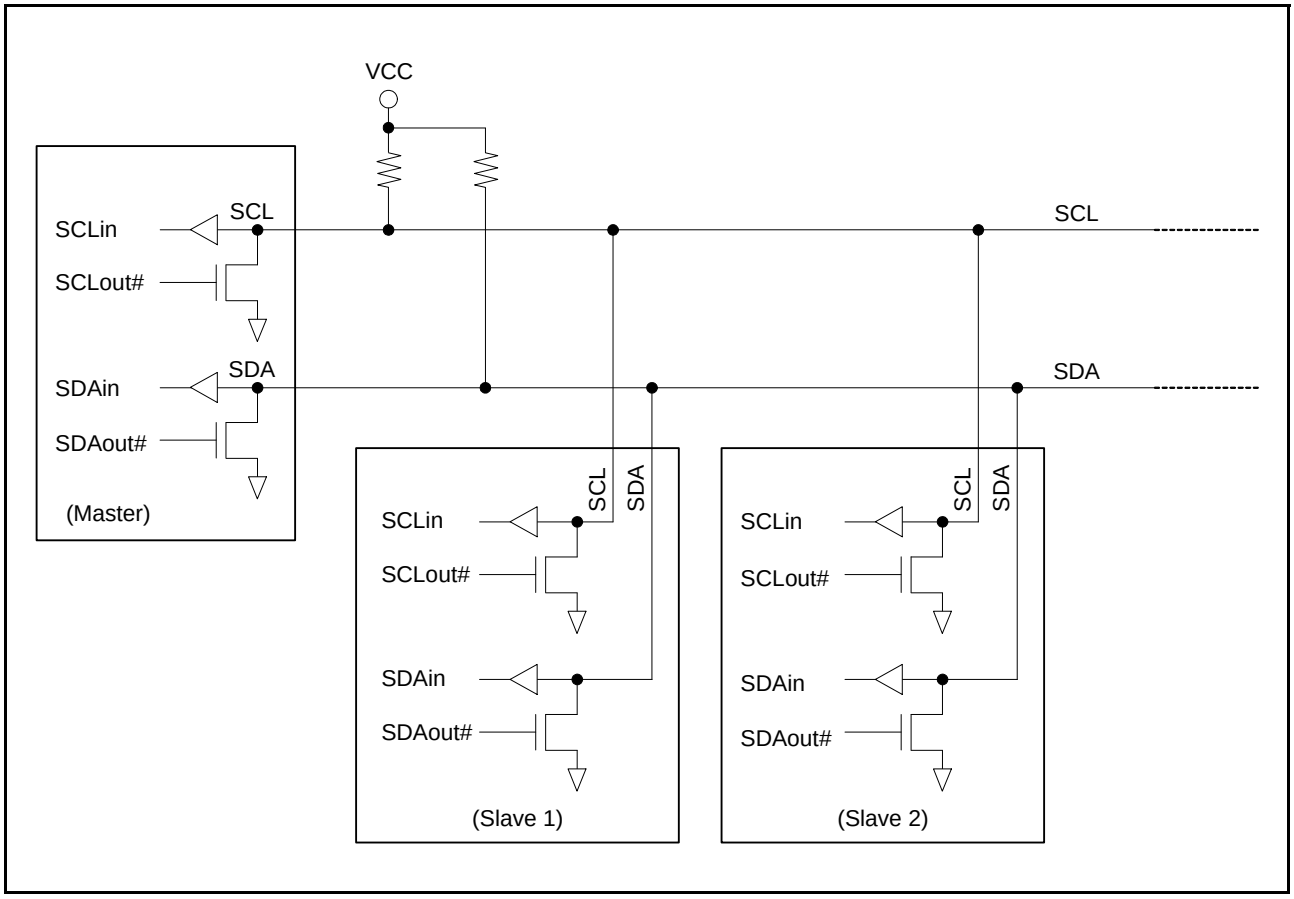


Figure 24.2 I/O Pin Connection to the External Circuit (I2C-bus Configuration Example)

The logic levels of the input signals for RIIC are CMOS when the I2C-bus is selected (ICMR3.SMBS bit is 0), or TTL when the SMBus is selected (ICMR3.SMBS bit is 1).

Table 24.2 RIIC Pin Configuration

Channel	Pin Name	I/O	Function
RIIC0	SCL0	I/O	RIIC0 serial clock I/O pin
	SDA0	I/O	RIIC0 serial data I/O pin

24.2 Register Descriptions

24.2.1 I2C-bus Control Register 1 (ICCR1)

Address(es): RIIC0.ICCR1 0008 8300h

	b7	b6	b5	b4	b3	b2	b1	b0
	ICE	IICRST	CLO	SOWP	SCLO	SDAO	SCLI	SDAI
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	SDAI	SDA Line Monitor	0: SDA0 line is low. 1: SDA0 line is high.	R
b1	SCLI	SCL Line Monitor	0: SCL0 line is low. 1: SCL0 line is high.	R
b2	SDAO	SDA Output Control/Monitor	• Read: 0: The RIIC has driven the SDA0 pin low. 1: The RIIC has released the SDA0 pin. • Write: 0: The RIIC drives the SDA0 pin low. 1: The RIIC releases the SDA0 pin. (High level output is achieved through an external pull-up resistor.)	R/W
b3	SCLO	SCL Output Control/Monitor	• Read: 0: The RIIC has driven the SCL0 pin low. 1: The RIIC has released the SCL0 pin. • Write: 0: The RIIC drives the SCL0 pin low. 1: The RIIC releases the SCL0 pin. (High level output is achieved through an external pull-up resistor.)	R/W
b4	SOWP	SCLO/SDAO Write Protect	0: SCLO and SDAO bits can be written. 1: SCLO and SDAO bits are protected. (This bit is read as 1.)	R/W
b5	CLO	Additional SCL Output	0: Does not output an additional SCL (default). 1: Outputs an additional SCL. (The CLO bit is cleared automatically after one clock pulse is output.)	R/W
b6	IICRST	I2C-bus Interface Internal Reset	0: Releases the RIIC reset or internal reset. 1: Initiates the RIIC reset or internal reset. (Clears the bit counter and the SCL0/SDA0 output latch)	R/W
b7	ICE	I2C-bus Interface Enable	0: Disable (SCL0 and SDA0 pins in inactive state) 1: Enable (SCL0 and SDA0 pins in active state) (Combined with the IICRST bit to select either RIIC or internal reset.)	R/W

SDAO Bit (SDA Output Control/Monitor) and SCLO Bit (SCL Output Control/Monitor)

These bits are used to directly control the SDA0 and SCL0 signals output from the RIIC.

When writing to these bits, also write 0 to the SOWP bit.

The result of setting these bits is input to the RIIC via the input buffer. When slave mode is selected, a start condition may be detected and the bus may 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 above conditions is not guaranteed.

When reading these bits, the state of signals output from the RIIC can be read.

CLO Bit (Additional SCL Output)

This bit is used to output an additional SCL for debugging or error processing.
Normally, set the bit to 0. Setting the bit to 1 in a normal communication state causes a communication error.
For details on this function, refer to section 24.11.2, Additional SCL Output Function.

IICRST Bit (I2C-bus Interface Internal Reset)

This bit is used to reset the internal states of the RIIC.
Setting this bit to 1 initiates an RIIC reset or internal reset.
Whether an RIIC reset or internal reset is initiated is determined according to the combination with the ICE bit. Table 24.3 lists the resets of the RIIC.
The RIIC reset initializes all registers and internal states of the RIIC, and the internal reset initializes the bit counter (ICMR1.BC[2:0] bits), the I2C-bus shift register (ICDRS), and the I2C-bus status registers (ICSR1 and ICSR2) as well as the internal states of the RIIC. For the reset conditions for each register, refer to section 24.14, Initialization of Registers and Functions When a Reset is Applied or a Condition is Detected.
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 RIIC without initializing the port settings and the control and setting registers of the RIIC when the bus or RIIC hangs up due to a communication error.
If the RIIC hangs up in a low level output state, resetting the internal states cancels the low level output state and releases the bus with the SCL0 pin and SDA0 pin at a high impedance.

Note: If an internal reset is initiated using the IICRST bit for a bus hang-up occurred during communication with the master device in slave mode, the states may become different between the slave device and the master device (due to the difference in the bit counter information). For this reason, do not initiate an internal reset in slave mode, but initiate restoration processing from the master device. If an internal reset is necessary because the RIIC has hung with the SCL0 line in a low level output state in slave mode, initiate an internal reset and then generate a restart condition from the master device or resume communications from the start condition after having generated a stop condition. If communication is restarted by initiating a reset solely in the slave device without generating a start condition or restart condition from the master device, synchronization will be lost because the master and slave devices operate asynchronously.

Table 24.3 RIIC Resets

IICRST	ICE	State	Specifications
1	0	RIIC reset	Resets all registers and internal states of the RIIC.
	1	Internal reset	Resets the ICMR1.BC[2:0] bits, registers ICSR1, ICSR2, and ICDRS, and the internal states of the RIIC.

ICE Bit (I2C-bus Interface Enable)

This bit selects the active or inactive state of the SCL0 and SDA0 pins. It can also be combined with the IICRST bit to initiate two types of resets. See Table 24.3, RIIC Resets, for the types of resets.
Set the ICE bit to 1 when using the RIIC. The SCL0 and SDA0 pins are placed in the active state when the ICE bit is set to 1.
Set the ICE bit to 0 when the RIIC is not to be used. The SCL0 and SDA0 pins are placed in the inactive state when the ICE bit is set to 0. Do not assign the SCL0 or SDA0 pin to the RIIC when setting up the multi-function pin controller (MPC). Note that the slave address comparison operation is carried out if the pins are assigned to the RIIC.

24.2.2 I2C-bus Control Register 2 (ICCR2)

Address(es): RIIC0.ICCR2 0008 8301h

	b7	b6	b5	b4	b3	b2	b1	b0
	BBSY	MST	TRS	—	SP	RS	ST	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	ST	Start Condition Generation Request	0: Does not request to generate a start condition. 1: Requests to generate a start condition.	R/W
b2	RS	Restart Condition Generation Request	0: Does not request to generate a restart condition. 1: Requests to generate a restart condition.	R/W
b3	SP	Stop Condition Generation Request	0: Does not request to generate a stop condition. 1: Requests to generate a stop condition.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	TRS	Transmit/Receive Mode	0: Receive mode 1: Transmit mode	R/W*1
b6	MST	Master/Slave Mode	0: Slave mode 1: Master mode	R/W*1
b7	BBSY	Bus Busy Detection Flag	0: The I2C-bus is released (bus free state). 1: The I2C-bus is occupied (bus busy state).	R

Note 1. When the ICMR1.MTWP bit is set to 1, bits MST and TRS can be written to.

ST Bit (Start Condition Generation Request)

This bit is used to request transition to master mode and generation of a start condition.

When this bit is set to 1 to request to generate a start condition, a start condition is generated when the BBSY flag is set to 0 (bus free state).

For details on the start condition generation, refer to section 24.10, Start Condition/Restart Condition/Stop Condition Generating 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 has been generated (a start condition is detected)
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: Set the ST bit to 1 (requests to generate a start condition) when the BBSY flag is set to 0 (bus free state). Note that arbitration may be lost due to a start condition generation error if the ST bit is set to 1 (requests to generate a start condition) when the BBSY flag is set to 1 (bus busy state).

RS Bit (Restart Condition Generation Request)

This bit is used to request that a restart condition be generated in master mode.
When this bit is set to 1 to request to generate a restart condition, a restart condition is generated when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode).
For details on the restart condition generation, refer to section 24.10, Start Condition/Restart Condition/Stop Condition Generating Function.
[Setting condition]

- When 1 is written to the RS bit with the ICCR2.BBSY flag set to 1

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition has been generated (a start condition is detected)
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: Do not set the RS bit to 1 while generating a stop condition.
Note: If 1 (requests to generate a restart condition) is written to the RS bit in slave mode, the restart condition is not generated but the RS bit remains set to 1. If the operating mode changes to master mode with the bit not being cleared, note that the restart condition may be generated.

SP Bit (Stop Condition Generation Request)

This bit is used to request that a stop condition be generated in master mode.
When this bit is set to 1 to request to generate a stop condition, a stop condition is generated when the BBSY flag is set to 1 (bus busy state) and the MST bit is set to 1 (master mode).
For details on the stop condition generation, refer to section 24.10, Start Condition/Restart Condition/Stop Condition Generating Function.
[Setting condition]

- When 1 is written to the SP bit with both the BBSY flag and the ICCR2.MST bit set to 1

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition has been generated (a stop condition is detected)
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: Writing to the SP bit is not possible while the setting of the BBSY flag is 0 (bus free state).
Note: Do not set the SP bit to 1 while a restart condition is being generated.

TRS Bit (Transmit/Receive Mode)

This bit indicates transmit or receive mode.
The RIIC is in receive mode when the TRS bit is set to 0 and is in transmit mode when the bit is set to 1. Combination of this bit and the MST bit indicates the operating mode of the RIIC.
The value of the TRS bit is automatically changed to 1 for transmission or 0 for reception in response to the generation or detection of a start condition and setting of the R/W# bit. Although writing to the TRS bit is possible when the ICMR1.MTWP bit is set to 1, writing to this bit is not necessary during normal usage.
[Setting conditions]

- When a start condition is generated normally according to the start condition generation request (when a start condition is detected with the ST bit set to 1)
- When a restart condition is generated normally according to the restart condition generation 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 the ICSER register, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the ICMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- The ICSR2.AL (arbitration-lost) flag being set to 1
- In master mode, reception of a slave address to which an R/W# bit with the value 1 is appended
- In slave mode, a match between the received address and the address enabled in the ICSER register when the value of the received R/W# bit is 0 (including cases where the received address is the general call address)
- In slave mode, a restart condition is detected (that is, a start condition is detected while the ICCR2.BBSY flag is 1 and the ICCR2.MST bit is 0)
- When 0 is written to the TRS bit with the ICMR1.MTWP bit set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

MST Bit (Master/Slave Mode)

This bit indicates master or slave mode.
The RIIC is in slave mode when the MST bit is set to 0 and is in master mode when the bit is set to 1. Combination of this bit and the TRS bit indicates the operating mode of the RIIC.
The value of the MST bit is automatically changed to 1 for master mode or 0 for slave mode by generating of a start condition and generating or detection of a stop condition, etc. Although writing to the MST bit is possible when the ICMR1.MTWP bit is set to 1, writing to this bit is not necessary during normal usage.
[Setting conditions]

- When a start condition is generated normally according to the start condition generation request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the ICMR1.MTWP bit set to 1

[Clearing conditions]

- When a stop condition is detected
- When the ICSR2.AL (arbitration-lost) flag is set to 1
- When 0 is written to the MST bit with the ICMR1.MTWP bit set to 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

BBSY Flag (Bus Busy Detection Flag)

The BBSY flag indicates whether the I2C-bus is occupied (bus busy state) or released (bus free state).
This bit is set to 1 when the SDA0 line changes from high to low under the condition of SCL0 line = high, assuming that a start condition has been generated.
The RIIC recognizes the SDA0 line changing from low to high while the SCL0 line is high as generation of the stop condition. After that, this flag becomes 0 if the RIIC does not detect a start condition during the bus free time (the period set in the ICBRL register).
[Setting condition]

- When a start condition is detected

[Clearing conditions]

- When the bus free time (specified in the ICBRL register) start condition is not detected after detecting a stop condition
- When 1 is written to the ICCR1.IICRST bit with the ICCR1.ICE bit set to 0 (RIIC reset)

24.2.3 I2C-bus Mode Register 1 (ICMR1)

Address(es): RIIC0.ICMR1 0008 8302h

	b7	b6	b5	b4	b3	b2	b1	b0
	MTWP	CKS[2:0]			BCWP	BC[2:0]		
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	BC[2:0]	Bit Counter	b2 b0 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
b3	BCWP	BC Write Protect	0: Enables a value to be written in the BC[2:0] bits. (This bit is read as 1.)	R/W*1
b6 to b4	CKS[2:0]	Internal Reference Clock Select	Select the internal reference clock (IICφ) source for the RIIC. b6 b4 0 0 0: PCLK/1 0 0 1: PCLK/2 0 1 0: PCLK/4 0 1 1: PCLK/8 1 0 0: PCLK/16 1 0 1: PCLK/32 1 1 0: PCLK/64 1 1 1: PCLK/128	R/W
b7	MTWP	MST/TRS Write Protect	0: Disables writing to the ICCR2.MST and TRS bits. 1: Enables writing to the ICCR2.MST and TRS bits.	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)

These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a rising edge on the SCL0 line. Although these bits are writable and readable, it is not necessary to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one (data is transferred with an additional acknowledgment bit) between transferred bytes when the SCL0 line is low.

The values of the BC[2:0] bits return to 000b at the end of a data transfer including the acknowledgment bit or when a start condition including a restart condition is detected.

24.2.4 I2C-bus Mode Register 2 (ICMR2)

Address(es): RIIC0.ICMR2 0008 8303h

	b7	b6	b5	b4	b3	b2	b1	b0
	DLCS	SDDL[2:0]			—	TMOH	TMOL	TMOS
Value after reset:	0	0	0	0	0	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOS	Timeout Detection Time Select	0: Long mode is selected. 1: Short mode is selected.	R/W
b1	TMOL	Timeout L Count Control	0: Count-up is disabled while the SCL0 line is low. 1: Count-up is enabled while the SCL0 line is low.	R/W
b2	TMOH	Timeout H Count Control	0: Count-up is disabled while the SCL0 line is high. 1: Count-up is enabled while the SCL0 line is high.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	SDDL[2:0]	SDA Output Delay Counter	• When ICMR2.DLCS bit is 0 (IICφ) b6 b4 0 0: No output delay 0 1: 1 IICφ cycle 1 0: 2 IICφ cycles 1 1: 3 IICφ cycles 0 0: 4 IICφ cycles 0 1: 5 IICφ cycles 1 0: 6 IICφ cycles 1 1: 7 IICφ cycles • When ICMR2.DLCS bit is 1 (IICφ/2) b6 b4 0 0: No output delay 0 1: 1 or 2 IICφ cycles 1 0: 3 or 4 IICφ cycles 1 1: 5 or 6 IICφ cycles 0 0: 7 or 8 IICφ cycles 0 1: 9 or 10 IICφ cycles 1 0: 11 or 12 IICφ cycles 1 1: 13 or 14 IICφ cycles	R/W
b7	DLCS	SDA Output Delay Clock Source Select	0: The internal reference clock (IICφ) is selected as the clock source of the SDA output delay counter. 1: The internal reference clock divided by 2 (IICφ/2) is selected as the clock source of the SDA output delay counter.*1	R/W

Note 1. The DLCS bit setting of 1 (IICφ/2) only becomes valid when SCL pin is low. When SCL pin is high, the DLCS bit setting of 1 becomes invalid and the clock source becomes the internal reference clock (IICφ).

TMOS Bit (Timeout Detection Time Select)

This bit is used to select long mode or short mode for the timeout detection time when the timeout function is enabled (ICFER.TMOE bit is 1). When this bit is set to 0, long mode is selected. When this bit 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 SCL0 line is in the state that enables this counter as specified by bits TMOH and TMOL, the counter counts up in synchronization with the internal reference clock (IICφ) as a count source.

For details on the timeout function, refer to section 24.11.1, Timeout Function.

TMOL Bit (Timeout L Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL0 line is held low when the timeout function is enabled (ICFER.TMOE bit is 1).

TMOH Bit (Timeout H Count Control)

This bit is used to enable or disable the internal counter of the timeout function to count up while the SCL0 line is held high when the timeout function is enabled (ICFER.TMOE bit is 1).

SDDL[2:0] Bits (SDA Output Delay Counter)

The SDA output can be delayed by the SDDL[2:0] setting. This counter works with the clock source selected by the DLCS bit. The setting of this function can be used for all types of SDA output, including the transmission of the acknowledgment bit.

Set the SDA output delay time to meet the I2C-bus specification (within the data valid time/data valid acknowledge time*1) or the SMBus specification (more than the data hold time (300 ns) and less than “clock low period – data setup time (250 ns)”). Note that, if a value outside the specification is set, communication with communication devices may malfunction or it may seemingly become a start condition or stop condition depending on the bus state.

For details on this function, refer to section 24.5, SDA Output Delay Function.

Note 1. Data valid time/data valid acknowledge time
3,450 ns (up to 100 kbps: Standard-mode (Sm))
900 ns (up to 400 kbps: Fast-mode (Fm))

24.2.5 I2C-bus Mode Register 3 (ICMR3)

Address(es): RIIC0.ICMR3 0008 8304h

	b7	b6	b5	b4	b3	b2	b1	b0
	SMBS	WAIT	RDRFS	ACKWP	ACKBT	ACKBR	NF[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	NF[1:0]	Noise Filter Stage Select	b1 b0 0 0: Noise of up to one IICφ cycle is filtered out (single-stage filter). 0 1: Noise of up to two IICφ cycles is filtered out (2-stage filter). 1 0: Noise of up to three IICφ cycles is filtered out (3-stage filter). 1 1: Noise of up to four IICφ cycles is filtered out (4-stage filter).	R/W
b2	ACKBR	Received Acknowledge	0: 0 is received as the acknowledgment bit (ACK reception). 1: 1 is received as the acknowledgment bit (NACK reception).	R
b3	ACKBT	Transmit Acknowledge	0: 0 is to be sent as the acknowledgment bit (ACK transmission). 1: 1 is to be sent as the acknowledgment bit (NACK transmission).	R/W*1
b4	ACKWP	ACKBT Write Protect	0: Modification of the ACKBT bit is disabled. 1: Modification of the ACKBT bit is enabled.	R/W*1
b5	RDRFS	RDRF Flag Set Timing Select	0: The RDRF flag is set at the rising edge of the ninth SCL. (The SCL0 line is not held low at the falling edge of the eighth clock pulse.) 1: The RDRF flag is set at the rising edge of the eighth SCL. (The SCL0 line is held low at the falling edge of the eighth clock pulse.) Low-hold is released by writing a value to the ACKBT bit.	R/W*2
b6	WAIT	WAIT	0: No WAIT (The period between ninth clock pulse and first clock pulse is not held low.) 1: WAIT (The period between ninth clock pulse and first clock pulse is held low.) Low-hold is released by reading the ICDRR register.	R/W*2
b7	SMBS	SMBus/I2C-bus Select	0: The I2C-bus is selected. 1: The SMBus is selected.	R/W

Note 1. Write to the ACKBT bit only while the ACKWP bit is already 1. If it is attempted to write 1 to both the ACKWP and ACKBT bits at the same time, the ACKBT bit will not be set to 1.

Note 2. The WAIT and RDRFS bits are valid only in receive mode (invalid in transmit mode).

NF[1:0] Bits (Noise Filter Stage Select)

These bits are used to select the number of stages in the digital noise filter.

For details on the digital noise filter function, refer to section 24.6, Digital Noise Filters.

Note: Set the noise range to be filtered out by the noise filter within a range less than the SCL0 line high period or low period. If the noise filter width is set to a value of [the shorter one of either SCL high width or SCL low width] – {1.5 × t_{IICcyc} (cycle time of internal reference clock (IICφ)) + 120 ns (pulse width suppressed by the analog noise filter, a reference value)} or a greater value, the serial clock is regarded as noise by the noise filter function of the RIIC, which may prevent the RIIC from operating normally.

ACKBR Bit (Received Acknowledge)

This bit is used to store the value of the acknowledgment bit received from the receiver in transmit mode.
[Setting condition]

- When 1 is received as the acknowledgment bit with the ICCR2.TRS bit set to 1

[Clearing conditions]

- When 0 is received as the acknowledgment bit with the ICCR2.TRS bit set to 1
- When 1 is written to the ICCR1.IICRST bit while the ICCR1.ICE bit is 0 (RIIC reset)

ACKBT Bit (Transmit Acknowledge)

This bit is used to set the bit to be sent at the acknowledgment timing 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 generation is detected (when a stop condition is detected with the ICCR2.SP bit set to 1)
- When 1 is written to the ICCR1.IICRST bit while the ICCR1.ICE bit is 0 (RIIC reset)

ACKWP Bit (ACKBT Write Protect)

This bit is used to control the modification of the ACKBT bit.

RDRFS Bit (RDRF Flag Set Timing Select)

This bit is used to select the RDRF flag set timing in receive mode and also to select whether to hold the SCL0 line low at the falling edge of the eighth SCL.
When the RDRFS bit is 0, the SCL0 line is not held low at the falling edge of the eighth SCL, and the RDRF flag is set to 1 at the rising edge of the ninth SCL.
When the RDRFS bit is 1, the RDRF flag is set to 1 at the rising edge of the eighth SCL and the SCL0 line is held low at the falling edge of the eighth SCL. The low-hold of the SCL0 line is released by writing a value to the ACKBT bit.
After data is received with this setting, the SCL0 line is automatically held low before the acknowledgment bit is sent. This enables processing to send ACK (ACKBT bit is 0) or NACK (ACKBT bit is 1) according to receive data.

WAIT Bit (WAIT)

This bit is used to control whether to hold the period between the ninth SCL and the first SCL low until the I2C-bus receive data register (ICDRR) is completely read each time 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 ninth and the first SCL 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 SCL0 line is held low from the falling edge of the ninth SCL until the ICDRR register value is read each time single-byte data is received. This enables receive operation in byte units.

Note: When the value of the WAIT bit is to be read, be sure to read the ICDRR register beforehand.

SMBS Bit (SMBus/I2C-bus Select)

Setting this bit to 1 selects the SMBus and enables the ICSER.HOAE bit.

24.2.6 I2C-bus Function Enable Register (ICFER)

Address(es): RIIC0.ICFER 0008 8305h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	SCLE	NFE	NACKE	SALE	NALE	MALE	TMOE
Value after reset:	0	1	1	1	0	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOE	Timeout Function Enable	0: The timeout function is disabled. 1: The timeout function is enabled.	R/W
b1	MALE	Master Arbitration-Lost Detection Enable	0: Master arbitration-lost detection is disabled. (Disables the arbitration-lost detection function and does not clear the ICCR2.MST and TRS bits automatically when arbitration is lost.) 1: Master arbitration-lost detection is enabled. (Enables the arbitration-lost detection function and clears the ICCR2.MST and TRS bits automatically when arbitration is lost.)	R/W
b2	NALE	NACK Transmission Arbitration-Lost Detection Enable	0: NACK transmission arbitration-lost detection is disabled. 1: NACK transmission arbitration-lost detection is enabled.	R/W
b3	SALE	Slave Arbitration-Lost Detection Enable	0: Slave arbitration-lost detection is disabled. 1: Slave arbitration-lost detection is enabled.	R/W
b4	NACKE	NACK Reception Transfer Suspension Enable	0: Transfer operation is not suspended during NACK reception (transfer suspension disabled). 1: Transfer operation is suspended during NACK reception (transfer suspension enabled).	R/W
b5	NFE	Digital Noise Filter Enable	0: Digital noise filters are not used. 1: Digital noise filters are used.	R/W
b6	SCLE	SCL Synchronization Enable	0: SCL synchronization is disabled. 1: SCL synchronization is enabled.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

TMOE Bit (Timeout Function Enable)

This bit is used to enable or disable the timeout function.

For details on the timeout function, refer to section 24.11.1, Timeout Function.

MALE Bit (Master Arbitration-Lost Detection Enable)

This bit is used to specify whether to use the arbitration-lost detection function in master mode. Normally, set this bit to 1.

NALE Bit (NACK Transmission Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when ACK is detected during transmission of NACK in receive mode (such as when slaves with the same address exist on the bus or when two or more masters select the same slave device simultaneously with different number of receive bytes).

SALE Bit (Slave Arbitration-Lost Detection Enable)

This bit is used to specify whether to cause arbitration to be lost when a value different from the value being transmitted is detected on the bus in slave transmit mode (such as when slaves with the same address exist on the bus or when a mismatch with the transmit data occurs due to noise).

NACKE Bit (NACK Reception Transfer Suspension Enable)

This bit is used to specify whether to continue or discontinue the data transfer when NACK is received in transmit mode. Normally, 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 value of the received acknowledgment bit.

For details on the NACK reception transfer suspension function, refer to section 24.8.2, NACK Reception Transfer Suspension Function.

SCLE Bit (SCL Synchronization Enable)

This bit is used to specify whether to the SCL output is to be synchronized with the SCL input. Normally, set this bit to 1.

When the SCLE bit is set to 0 (SCL synchronization is disabled), the RIIC does not synchronize the SCL output with the SCL input. In this setting, the RIIC outputs the clock with the transfer rate set in registers ICBRH and ICBRL regardless of the SCL0 line state. For this reason, if the load of the I2C-bus line is much larger than the specification value or if the SCL output overlaps in multiple masters, the short-cycle SCL that does not meet the specification may be output. When the SCL synchronization is not used, it also affects the generation of a start condition, restart condition, and stop condition, and the continuous output of additional SCL.

This bit must not be set to 0 except for checking the output of the set transfer rate.

24.2.7 I2C-bus Status Enable Register (ICSER)

Address(es): RIIC0.ICSER 0008 8306h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOAE	—	DIDE	—	GCAE	SAR2E	SAR1E	SAR0E
Value after reset:	0	0	0	0	1	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	SAR0E	Slave Address Register 0 Enable	0: Slave address in registers SARL0 and SARU0 is disabled. 1: Slave address in registers SARL0 and SARU0 is enabled.	R/W
b1	SAR1E	Slave Address Register 1 Enable	0: Slave address in registers SARL1 and SARU1 is disabled. 1: Slave address in registers SARL1 and SARU1 is enabled.	R/W
b2	SAR2E	Slave Address Register 2 Enable	0: Slave address in registers SARL2 and SARU2 is disabled. 1: Slave address in registers SARL2 and SARU2 is enabled.	R/W
b3	GCAE	General Call Address Enable	0: General call address detection is disabled. 1: General call address detection is enabled.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DIDE	Device-ID Address Detection Enable	0: Device-ID address detection is disabled. 1: Device-ID address detection is enabled.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	HOAE	Host Address Enable	0: Host address detection is disabled. 1: Host address detection is enabled.	R/W

SARyE Bit (Slave Address Register y Enable) (y = 0 to 2)

This bit is used to enable or disable the slave address set in registers SARLy and SARUy.

When this bit is set to 1, the slave address set in registers 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 registers SARLy and SARUy is disabled and is ignored even if it matches the received slave address.

GCAE Bit (General Call Address Enable)

This bit is used to specify whether to ignore the general call address (0000 000b + 0 (write): All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the RIIC recognizes the received slave address as the general call address independently of the slave addresses set in registers SARLy and SARUy (y = 0 to 2) and performs 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)

This bit is used to specify whether to recognize and execute the device-ID address when a device ID (1111 100b) is received in the first byte after a start condition or restart condition is detected.

When this bit is set to 1, if the received first byte matches the device ID, the RIIC recognizes that the device-ID address has been received. When the following R/W# bit is 0 (write), the RIIC recognizes the second and the following bytes as slave addresses and continues the receive operation.

When this bit is set to 0, the RIIC ignores the received first byte even if it matches the device ID address and recognizes the first byte as a normal slave address.

For details on the device-ID address detection, refer to section 24.7.3, Device-ID Address Detection.

HOAE Bit (Host Address Enable)

This bit is used to specify whether to ignore received host address (0001 000b) when the ICMR3.SMBS bit is 1.

When this bit is set to 1 while the ICMR3.SMBS bit is 1, if the received slave address matches the host address, the RIIC recognizes the received slave address as the host address independently of the slave addresses set in registers SARLy and SARUy (y = 0 to 2) and performs the receive operation.

When the ICMR3.SMBS bit or the HOAE bit is set to 0, the received slave address is ignored even if it matches the host address.

24.2.8 I2C-bus Interrupt Enable Register (ICIER)

Address(es): RIIC0.ICIER 0008 8307h

	b7	b6	b5	b4	b3	b2	b1	b0
	TIE	TEIE	RIE	NAKIE	SPIE	STIE	ALIE	TMOIE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOIE	Timeout Interrupt Request Enable	0: Timeout interrupt (TMOI) request is disabled. 1: Timeout interrupt (TMOI) request is enabled.	R/W
b1	ALIE	Arbitration-Lost Interrupt Request Enable	0: Arbitration-lost interrupt (ALI) request is disabled. 1: Arbitration-lost interrupt (ALI) request is enabled.	R/W
b2	STIE	Start Condition Detection Interrupt Request Enable	0: Start condition detection interrupt (STI) request is disabled. 1: Start condition detection interrupt (STI) request is enabled.	R/W
b3	SPIE	Stop Condition Detection Interrupt Request Enable	0: Stop condition detection interrupt (SPI) request is disabled. 1: Stop condition detection interrupt (SPI) request is enabled.	R/W
b4	NAKIE	NACK Reception Interrupt Request Enable	0: NACK reception interrupt (NAKI) request is disabled. 1: NACK reception interrupt (NAKI) request is enabled.	R/W
b5	RIE	Receive Data Full Interrupt Request Enable	0: Receive data full interrupt (RXI) request is disabled. 1: Receive data full interrupt (RXI) request is enabled.	R/W
b6	TEIE	Transmit End Interrupt Request Enable	0: Transmit end interrupt (TEI) request is disabled. 1: Transmit end interrupt (TEI) request is enabled.	R/W
b7	TIE	Transmit Data Empty Interrupt Request Enable	0: Transmit data empty interrupt (TXI) request is disabled. 1: Transmit data empty interrupt (TXI) request is enabled.	R/W

TMOIE Bit (Timeout Interrupt Request Enable)

This bit is used to enable or disable timeout interrupt (TMOI) requests when the ICSR2.TMOF flag is set to 1. A TMOI interrupt request is canceled by setting the TMOF flag or the TMOIE bit to 0.

ALIE Bit (Arbitration-Lost Interrupt Request Enable)

This bit is used to enable or disable arbitration-lost interrupt (ALI) requests when the ICSR2.AL flag is set to 1. An ALI interrupt request is canceled by setting the AL flag or the ALIE bit to 0.

STIE Bit (Start Condition Detection Interrupt Request Enable)

This bit is used to enable or disable start condition detection interrupt (STI) requests when the ICSR2.START flag is set to 1. An STI interrupt request is canceled by setting the START flag or the STIE bit to 0.

SPIE Bit (Stop Condition Detection Interrupt Request Enable)

This bit is used to enable or disable stop condition detection interrupt (SPI) requests when the ICSR2.STOP flag is set to 1. An SPI interrupt request is canceled by setting the STOP flag or the SPIE bit to 0.

NAKIE Bit (NACK Reception Interrupt Request Enable)

This bit is used to enable or disable NACK reception interrupt (NAKI) requests when the ICSR2.NACKF flag is set to 1. An NAKI interrupt request is canceled by setting the NACKF flag or the NAKIE bit to 0.

RIE Bit (Receive Data Full Interrupt Request Enable)

This bit is used to enable or disable receive data full interrupt (RXI) requests when the ICSR2.RDRF flag is set to 1.

TEIE Bit (Transmit End Interrupt Request Enable)

This bit is used to enable or disable transmit end interrupt (TEI) requests when the ICSR2.TEND flag is set to 1. An TEI interrupt request is canceled by setting the TEND flag or the TEIE bit to 0.

TIE Bit (Transmit Data Empty Interrupt Request Enable)

This bit is used to enable or disable transmit data empty interrupt (TXI) requests when the ICSR2.TDRE flag is set to 1.

24.2.9 I2C-bus Status Register 1 (ICSR1)

Address(es): RIIC0.ICSR1 0008 8308h

	b7	b6	b5	b4	b3	b2	b1	b0
	HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	AAS0	Slave Address 0 Detection Flag	0: Slave address 0 is not detected. 1: Slave address 0 is detected.	R/(W) *1
b1	AAS1	Slave Address 1 Detection Flag	0: Slave address 1 is not detected. 1: Slave address 1 is detected.	R/(W) *1
b2	AAS2	Slave Address 2 Detection Flag	0: Slave address 2 is not detected. 1: Slave address 2 is detected.	R/(W) *1
b3	GCA	General Call Address Detection Flag	0: General call address is not detected. 1: General call address is detected.	R/(W) *1
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DID	Device-ID Address Detection Flag	0: Device-ID command is not detected. 1: Device-ID command is detected. • This bit is set to 1 when the first byte received immediately after a start condition is detected matches a value of (device ID (1111 100b) + 0 (write)).	R/(W) *1
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	HOA	Host Address Detection Flag	0: Host address is not detected. 1: Host address is detected. • This bit is set to 1 when the received slave address matches the host address (0001 000b).	R/(W) *1

Note 1. Only 0 can be written to clear the flag.

AASy Flag (Slave Address y Detection Flag) (y = 0 to 2)

[Setting conditions]

For 7-bit address format: SARUy.FS bit = 0

- When the received slave address matches the SARLy.SVA[6:0] bits value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL in the first byte.

For 10-bit address format: SARUy.FS bit = 1

- When the received slave address matches a value of (11110b + SARUy.SVA[1:0] bits) and the following address matches the SARLy value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 1 at the rising edge of the ninth SCL in the second byte.

[Clearing conditions]

- When 0 is written to the AASy flag after reading the AASy flag to be 1
- When a stop condition is detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

For 7-bit address format: SARUy.FS bit = 0

- When the received slave address does not match the SARLy.SVA[6:0] bits value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the first byte.

For 10-bit address format: SARUy.FS bit = 1

- When the received slave address does not match a value of (11110b + SARUy.SVA[1:0] bits) with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the first byte.
- When the received slave address matches a value of (11110b + SARUy.SVA[1:0] bits) and the following address does not match the SARLy value with the ICSER.SARyE bit set to 1 (slave address y detection enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the second byte.

GCA Flag (General Call Address Detection Flag)

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 (write)) with the ICSER.GCAE bit set to 1 (general call address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL in the first byte.

[Clearing conditions]

- When 0 is written to the GCA flag after reading GCA flag to be 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000b + 0 (write)) with the ICSER.GCAE bit set to 1 (general call address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the first byte.
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

DID Flag (Device-ID Address Detection Flag)

[Setting condition]

- When the first byte received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 (write)) with the ICSER.DIDE bit set to 1 (device-ID address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL in the first byte.

[Clearing conditions]

- When 0 is written to the DID flag after reading DID flag to be 1
- When a stop condition is detected
- When the first byte received immediately after a start condition or restart condition is detected does not match a value of (device ID (1111 100b)) with the ICSER.DIDE bit set to 1 (device-ID address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the first byte.
- When the first byte received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 (write)) and the second byte does not match any of slave addresses 0 to 2 with the ICSER.DIDE bit set to 1 (device-ID address detection is enabled)
  This flag is set to 0 at the rising edge of the ninth SCL in the second byte.
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

HOA Flag (Host Address Detection Flag)

[Setting condition]

- When the received slave address matches the host address (0001 000b) with the ICSER.HOAE bit set to 1 (host address detection is enabled)
  This flag is set to 1 at the rising edge of the ninth SCL in the first byte.

[Clearing conditions]

- When 0 is written to the HOA flag after reading HOA flag to be 1
- When a stop condition is detected
- When the received slave address does not match the host address (0001 000b) with the ICSER.HOAE bit set to 1

(host address detection is enabled)
This flag is set to 0 at the rising edge of the ninth SCL in the first byte.

- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

24.2.10 I2C-bus Status Register 2 (ICSR2)

Address(es): RIIC0.ICSR2 0008 8309h

	b7	b6	b5	b4	b3	b2	b1	b0
	TDRE	TEND	RDRF	NACKF	STOP	START	AL	TMOF
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TMOF	Timeout Detection Flag	0: Timeout is not detected. 1: Timeout is detected.	R/(W) *1
b1	AL	Arbitration-Lost Flag	0: Arbitration is not lost. 1: Arbitration is lost.	R/(W) *1
b2	START	Start Condition Detection Flag	0: Start condition is not detected. 1: Start condition is detected.	R/(W) *1
b3	STOP	Stop Condition Detection Flag	0: Stop condition is not detected. 1: Stop condition is detected.	R/(W) *1
b4	NACKF	NACK Detection Flag	0: NACK is not detected. 1: NACK is detected.	R/(W) *1
b5	RDRF	Receive Data Full Flag	0: The ICDRR register contains no receive data. 1: The ICDRR register contains receive data.	R/(W) *1
b6	TEND	Transmit End Flag	0: Data is being transmitted. 1: Data has been transmitted.	R/(W) *1
b7	TDRE	Transmit Data Empty Flag	0: The ICDRT register contains transmit data. 1: The ICDRT register contains no transmit data.	R

Note 1. Only 0 can be written to clear the flag.

TMOF Flag (Timeout Detection Flag)

This flag is set to 1 when the RIIC recognizes timeout after the SCL0 line state remains unchanged for a certain period.
[Setting condition]

- When the SCL0 line state remains unchanged for the period specified by bits ICMR2.TMOH, TMOL, and TMOS while the ICFER.TMOE bit is 1 (the timeout function is enabled) in master mode or in slave mode and the received slave address matches.

[Clearing conditions]

- When 0 is written to the TMOF bit after reading TMOF = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

AL Flag (Arbitration-Lost Flag)

This flag shows that bus mastership has been lost (loss in arbitration) due to a bus conflict or some other reason when a start condition is generated or an address and data are transmitted. The RIIC monitors the level on the SDA0 line during transmission and, if the level on the line does not match the value of the bit being output, sets the value of the AL flag to 1 to indicate that the bus is occupied by another device.
The RIIC can also detect loss of arbitration during NACK transmission in receive 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 SDA0 line level at the rising edge of SCL except for the ACK period during data (including slave address) transmission in master transmit mode (when the SDA0 line is driven low while the internal SDA output is high (the SDA0 pin is in the high-impedance state))
- When a start condition is detected while the ICCR2.ST bit is 1 (requests to generate a start condition) or the internal SDA output state does not match the SDA0 line level
- When the ICCR2.ST bit is set to 1 (requests to generate a start condition) with the ICCR2.BBSY flag set to 1.

When NACK arbitration-lost detection is enabled: ICFER.NALE = 1

- When the internal SDA output state does not match the SDA0 line level at the rising edge of SCL 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 SDA0 line level at the rising edge of SCL 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 ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Table 24.4 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 generation error	When internal SDA output state does not match SDA0 line level when a start condition is detected while the ICCR2.ST bit is 1
					When ICCR2.ST bit is set to 1 with ICCR2.BBSY flag set to 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 receive mode or slave receive mode
×	×	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

×: Don't care

START Flag (Start Condition Detection Flag)

[Setting condition]

- When a start condition (or a restart condition) is detected

[Clearing conditions]

- When 0 is written to the START bit after reading START = 1
- When a stop condition is detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

STOP Flag (Stop Condition Detection Flag)

[Setting condition]

- When a stop condition is detected

[Clearing conditions]

- When 0 is written to the STOP bit after reading STOP = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

NACKF Flag (NACK Detection Flag)

[Setting condition]

- When ACK is not received (NACK is received) from the receiver in transmit mode with the ICFER.NACKE bit set to 1 (transfer suspension enabled)

[Clearing conditions]

- When 0 is written to the NACKF bit after reading NACKF = 1
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: When the NACKF flag is set to 1, the RIIC suspends data transmission/reception. Writing to the ICDRT register in transmit mode or reading from the ICDRR register in receive mode with the NACKF flag set to 1 does not enable data transmit/receive operation. To restart data transmission/reception, set the NACKF flag to 0.

RDRF Flag (Receive Data Full Flag)

[Setting conditions]

- When receive data has been transferred from the ICDRS register to the ICDRR register
  This flag is set to 1 at the rising edge of the eighth or ninth SCL (selected by the ICMR3.RDRFS bit)
- When the received slave address matches after a start condition (or a restart condition) is detected with the ICCR2.TRS bit set to 0

[Clearing conditions]

- When 0 is written to the RDRF bit after reading RDRF = 1
- When data is read from the ICDRR register
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

TEND Flag (Transmit End Flag)

[Setting condition]

- At the rising edge of the ninth SCL while the TDRE flag is 1

[Clearing conditions]

- When 0 is written to the TEND bit after reading TEND = 1
- When data is written to the ICDRT register
- When a stop condition is detected
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

TDRE Flag (Transmit Data Empty Flag)

[Setting conditions]

- When data has been transferred from the ICDRT register to the ICDRS register and the ICDRT register becomes empty
- When the ICCR2.TRS bit is set to 1
- When the received slave address matches while the TRS bit is 1

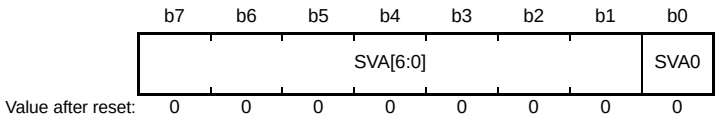
[Clearing conditions]

- When data is written to the ICDRT register
- When the ICCR2.TRS bit is set to 0
- When 1 is written to the ICCR1.IICRST bit to apply an RIIC reset or an internal reset

Note: The NACKF flag becoming 1 while the ICFER.NACKE bit is 1 suspends data transmission and reception by the RIIC. Even if the next data for transmission has already been written to the ICDRT register (the TDRE flag is 0), the data in the ICDRT register is retained but not transferred to the ICDRS register. At this point, the TDRE flag does not become 1.

24.2.11 Slave Address Register Ly (SARLy) (y = 0 to 2)

Address(es): RIIC0.SARL0 0008 830Ah, RIIC0.SARL1 0008 830Ch, RIIC0.SARL2 0008 830Eh



Bit	Symbol	Bit Name	Description	R/W
b0	SVA0	10-Bit Address LSB	Set a slave address	R/W
b7 to b1	SVA[6:0]	7-Bit Address/10-Bit Address Lower Bits	Set a slave address	R/W

SVA0 Bit (10-Bit Address LSB)

When the 10-bit address format is selected (SARUy.FS bit is 1), this bit functions as the LSB of a 10-bit address and forms the lower 8 bits of a 10-bit address in combination with the SVA[6:0] bits.
When the ICSER.SARyE bit is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, this bit is valid. While the SARUy.FS bit or SARyE bit is 0, the setting of 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 bit is 0), these bits function as a 7-bit address. When the 10-bit address format is selected (SARUy.FS bit is 1), these bits function as the lower 8 bits of a 10-bit address in combination with the SVA0 bit.
While the ICSER.SARyE bit is 0, the setting of these bits is ignored.

24.2.12 Slave Address Register Uy (SARUy) (y = 0 to 2)

Address(es): RIIC0.SARU0 0008 830Bh, RIIC0.SARU1 0008 830Dh, RIIC0.SARU2 0008 830Fh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SVA[1:0]		FS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FS	7-Bit/10-Bit Address Format Select	0: The 7-bit address format is selected. 1: The 10-bit address format is selected.	R/W
b2, b1	SVA[1:0]	10-Bit Address Upper Bits	Set a slave address	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FS Bit (7-Bit/10-Bit Address Format Select)

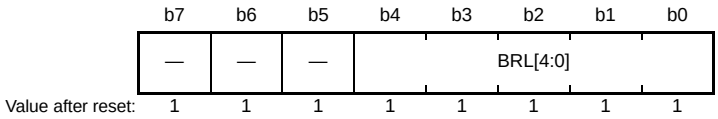
This bit is used to select 7-bit address or 10-bit address for slave address y (in registers SARLy and SARUy).
When the ICSER.SARyE bit is set to 1 (registers SARLy and SARUy enabled) and the SARUy.FS bit is 0, the 7-bit address format is selected for slave address y, the SARLy.SVA[6:0] bits setting is valid, and the settings of the SVA[1:0] bits and the SARLy.SVA0 bit are ignored.
When the ICSER.SARyE bit is set to 1 (registers SARLy and SARUy enabled) and the SARUy.FS bit is 1, the 10-bit address format is selected for slave address y and the settings of the SVA[1:0] bits and SARLy are valid.
While the ICSER.SARyE bit is 0 (registers SARLy and SARUy disabled), the setting of the SARUy.FS bit is invalid.

SVA[1:0] Bits (10-Bit Address Upper Bits)

When the 10-bit address format is selected (FS = 1), these bits function as the upper 2 bits of a 10-bit address.
When the ICSER.SARyE bit is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, these bits are valid.
While the SARUy.FS bit or SARyE bit is 0, the setting of these bits is ignored.

24.2.13 I2C-bus Bit Rate Low-Level Register (ICBRL)

Address(es): RIIC0.ICBRL 0008 8310h



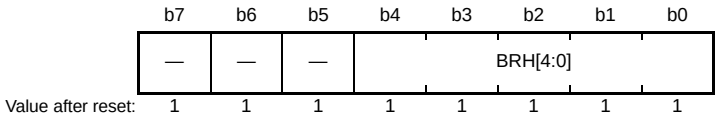
Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BRL[4:0]	Bit Rate Low Period	Low period of SCL	R/W
b7 to b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

ICBRL is a 5-bit register to set the low period of SCL.
It also works to generate the data setup time for automatic SCL low-hold operation (refer to section 24.8, Automatic Low-Hold Function for SCL); when the RIIC is used only in slave mode, this register needs to be set to a value longer than the data setup time*1.
ICBRL counts the low period with the internal reference clock (IICφ) specified by the ICMR1.CKS[2:0] bits.
If the digital noise filter is enabled (the ICFER.NFE bit is 1), set the ICBRL register to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICMR3.NF[1:0] bits.

Note 1. Data setup time (tSU: DAT)
250 ns (up to 100 kbps: Standard-mode (Sm))
100 ns (up to 400 kbps: Fast-mode (Fm))

24.2.14 I2C-bus Bit Rate High-Level Register (ICBRH)

Address(es): RIIC0.ICBRH 0008 8311h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BRH[4:0]	Bit Rate High Period	High period of SCL	R/W
b7 to b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

ICBRH is a 5-bit register to set the high period of SCL. ICBRH is valid in master mode. If the RIIC is used only in slave mode, this register need not to set the high period.
ICBRH counts the high period with the internal reference clock (IICφ) specified by the ICMR1.CKS[2:0] bits.
If the digital noise filter is enabled (the ICFER.NFE bit is 1), set the ICBRH register to a value at least one greater than the number of stages in the noise filter. Regarding the number of stages in the noise filter, see the description of the ICMR3.NF[1:0] bits.

The I2C transfer rate and the SCL duty are calculated using the following expression.
Transfer rate = 1 / {[(ICBRH + 1) + (ICBRL + 1)] / IICφ*1 + SCL0 line rise time [tr] + SCL0 line fall time [tf]}
Duty cycle = {SCL0 line rise time [tr]*2 + (ICBRH + 1) / IICφ} / {SCL0 line fall time [tf]*2 + (ICBRL + 1) / IICφ}

Note 1. IICφ = PCLK × Division ratio
Note 2. The SCL0 line rise time [tr] and SCL0 line fall time [tf] depend on the total bus line capacitance [Cb] and the pull-up resistor [Rp]. For details, refer to the I2C-bus specification from NXP Semiconductors.

Table 24.5 lists examples of ICBRH/ICBRL settings.

Table 24.5 Examples of ICBRH/ICBRL Settings for Transfer Rate

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	8			10			12.5		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	100b	22 (F6h)	25 (F9h)	101b	13 (EDh)	15 (EFh)	101b	16 (F0h)	20 (F4h)
50	010b	16 (F0h)	19 (F3h)	010b	21 (F5h)	24 (F8h)	011b	12 (ECh)	15 (EFh)
100	001b	15 (EFh)	18 (F2h)	001b	19 (F3h)	23 (F7h)	001b	24 (F8h)	29 (FDh)
400	000b	4 (E4h)	10 (EAh)	000b	5 (E5h)	12 (ECh)	000b	7 (E7h)	16 (F0h)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	16			20			25		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	101b	22 (F6h)	25 (F9h)	110b	13 (EDh)	15 (EFh)	110b	16 (F0h)	20 (F4h)
50	011b	16 (F0h)	19 (F3h)	011b	21 (F5h)	24 (F8h)	100b	12 (ECh)	15 (EFh)
100	010b	15 (EFh)	18 (F2h)	010b	19 (F3h)	23 (F7h)	010b	24 (F8h)	29 (FDh)
400	000b	9 (E9h)	20 (F4h)	000b	11 (EBh)	25 (F9h)	001b	7 (E7h)	16 (F0h)

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)					
	30			32		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	110b	20 (F4h)	24 (F8h)	110b	22 (F6h)	25 (F9h)
50	100b	15 (EFh)	18 (F2h)	100b	16 (F0h)	19 (F3h)
100	011b	14 (EEh)	17 (F1h)	011b	15 (EFh)	18 (F2h)
400	001b	8 (E8h)	19 (F3h)	001b	9 (E9h)	20 (F4h)

Note: ICBRH/ICBRL settings in these tables are calculated using the following values:
SCL0 line rise time (tr): 100 kbps or less (Sm): 1000 ns, 400 kbps or less (Fm): 300 ns
SCL0 line fall time (tf): 400 kbps or less (Sm/Fm): 300 ns
For the specified values of rise time (tr) and fall time (tf) of the SCL0 signal, refer to the I2C-bus specification from NXP Semiconductors.

24.2.15 I2C-bus Transmit Data Register (ICDRT)

Address(es): RIIC0.ICDRT 0008 8312h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	1	1	1	1	1	1	1	1

When the ICDRT register detects a space in the I2C-bus shift register (ICDRS), it transfers the transmit data that has been written to the ICDRT register to the ICDRS register and starts transmitting data in transmit mode.
The double-buffer structure of the ICDRT register and the ICDRS register allows continuous transmit operation if the next transmit data has been written to the ICDRT register while the ICDRS register data is being transmitted.
The ICDRT register can always be read and written. Write transmit data to the ICDRT register once when a transmit data empty interrupt (TXI) request is generated.

24.2.16 I2C-bus Receive Data Register (ICDRR)

Address(es): RIIC0.ICDRR 0008 8313h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

When 1 byte of data has been received, the received data is transferred from the I2C-bus shift register (ICDRS) to the ICDRR register to enable the next data to be received.
The double-buffer structure of the ICDRS register and the ICDRR register allows continuous receive operation if the received data has been read from the ICDRR register while the ICDRS register is receiving data.
The ICDRR register cannot be written. Read data from the ICDRR register once when a receive data full interrupt (RXI) request is generated.
If the ICDRR register receives the next receive data before the current data is read from the ICDRR register (while the ICSR2.RDRF flag is 1), the RIIC automatically holds the SCL line low one cycle before the RDRF flag is set to 1 next.

24.2.17 I2C-bus Shift Register (ICDRS)

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	—	—	—	—	—	—	—	—

The ICDRS register is an 8-bit shift register to transmit and receive data.
During transmission, transmit data is transferred from the ICDRT register to the ICDRS register and is sent from the SDA0 pin. During reception, data is transferred from the ICDRS register to the ICDRR register after 1 byte of data has been received.
The ICDRS register cannot be accessed directly.

24.3 Operation

24.3.1 Communication Data Format

The I2C-bus format consists of 8-bit data and 1-bit acknowledgment. The first byte following a start condition or restart condition is an address byte used to specify 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 generated.
Figure 24.3 shows the I2C-bus format, and Figure 24.4 shows the I2C-bus timing.

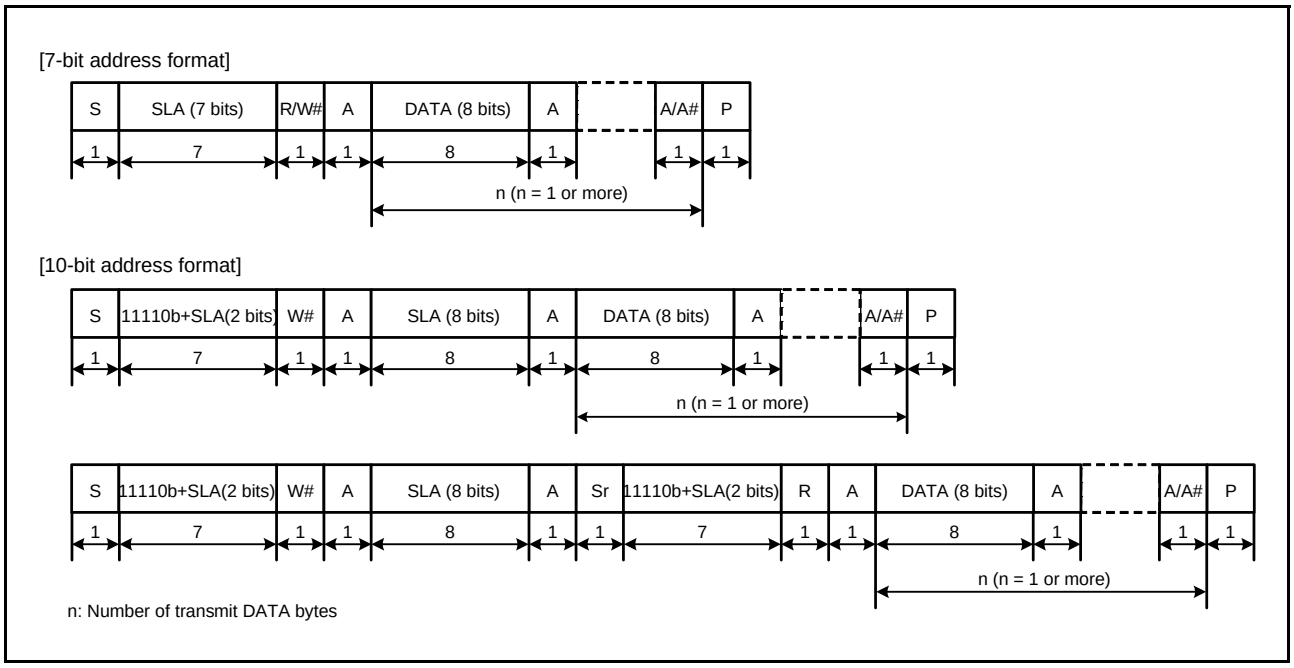


Figure 24.3 I2C-bus Format

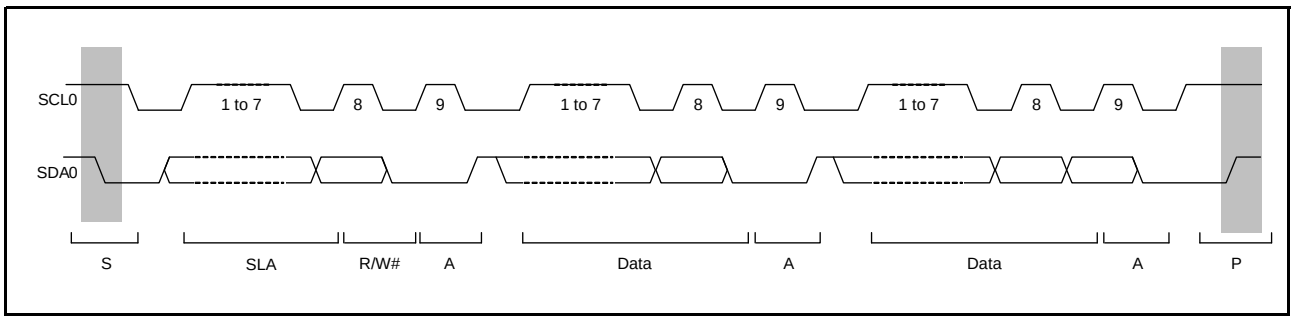


Figure 24.4 I2C-bus Timing (SLA = 7 Bits)

S: Start condition. The master device drives the SDA0 line low from high while the SCL0 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 receiver drives the SDA0 line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)

A#: Not Acknowledge. The receiver drives the SDA0 line high.

Sr: Restart condition. The master device drives the SDA0 line low from high after the setup time has elapsed with the SCL0 line high.

DATA: Transmitted or received data

P: Stop condition. The master device drives the SDA0 line high from low while the SCL0 line is high.

24.3.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in Figure 24.5.
Set the ICCR1.ICE bit to 1 (internal reset) after setting the ICCR1.IICRST bit to 1 (RIIC reset) with the ICCR1.ICE bit set to 0 (SCL0 and SDA0 pins in inactive state). This initializes the various flags and internal state of the ICSR1 register. After that, set registers SARLy, SARUy, ICSER, ICMR1, ICBRH, and ICBRL (y = 0 to 2), and set the other registers as necessary (for initial settings of the RIIC, see Figure 24.5). When the necessary register settings have been completed, set the ICCR1.IICRST bit to 0 (releases the RIIC reset). This step is not necessary if initialization of the RIIC has already been completed.

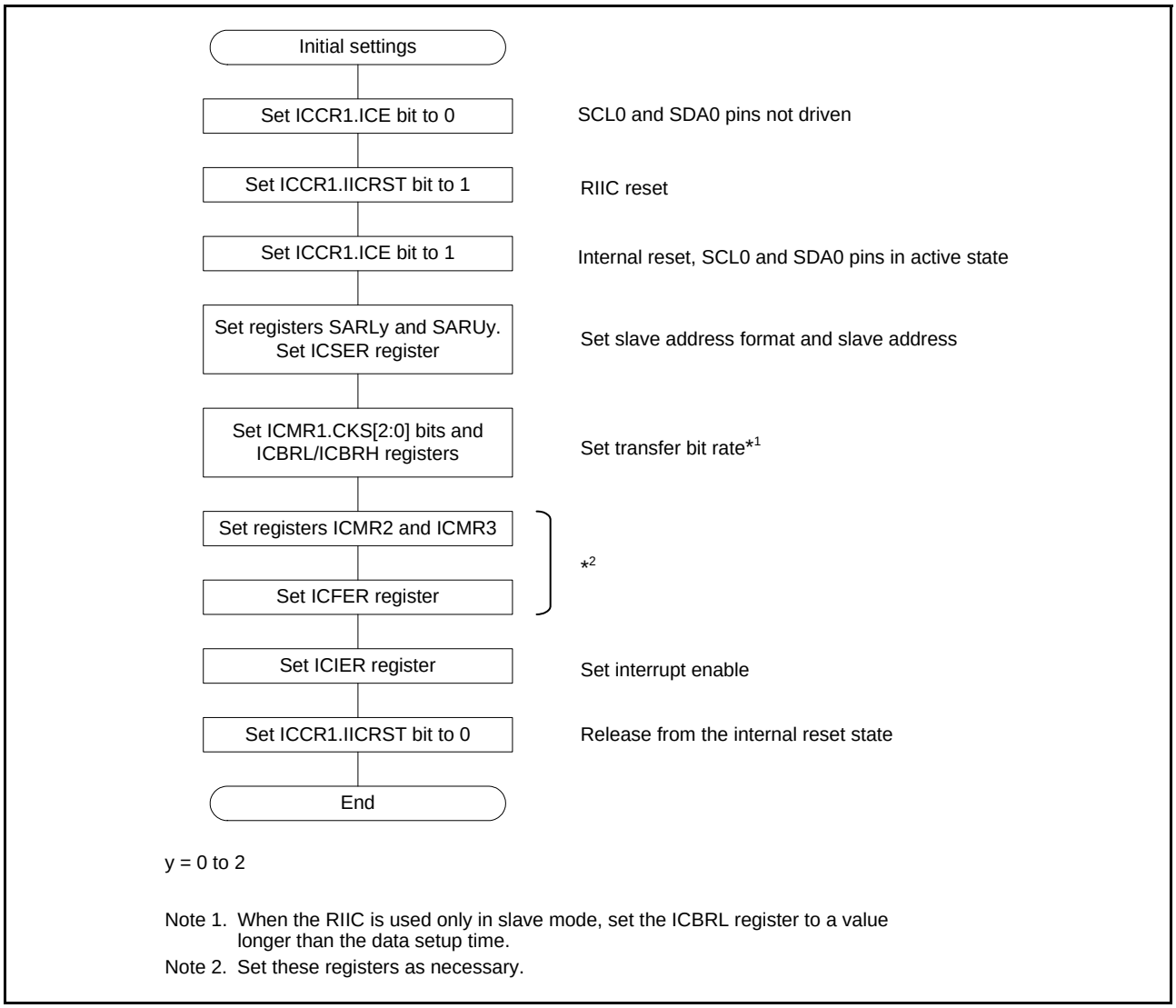


Figure 24.5 Example of RIIC Initialization Flowchart

24.3.3 Master Transmit Operation

In master transmit operation, the RIIC generates clock signals and sends data as the master device, and the slave device returns acknowledgments. Figure 24.6 shows an example of usage of master transmission and Figure 24.7 to Figure 24.9 show the timing of operations in master transmission.
The following describes the procedure and operations for master transmission.

(1) Initial settings. For details, refer to section 24.3.2, Initial Settings.
(2) Read the ICCR2.BBSY flag to check that the bus is open, and then set the ICCR2.ST bit to 1 (requests to generate a start condition). Upon receiving the request, the RIIC generates a start condition. At the same time, the BBSY flag and the ICSR2.START flag are automatically set to 1 and the ST bit is automatically set to 0. At this time, if the start condition is detected and the internal levels for the SDA output state and the levels on the SDA0 line have matched while the ST bit is 1, the RIIC recognizes that generating of the start condition as requested by the ST bit has been successfully completed, and bits MST and TRS in the ICCR2 register are automatically set to 1, placing the RIIC in master transmit mode. The ICSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.
(3) Check that the ICSR2.TDRE flag is 1, and then write the value for transmission (the slave address and the R/W# bit) to the ICDRT register. Once the data for transmission are written to the ICDRT register, the TDRE flag is automatically set to 0, the data are transferred from the ICDRT register to the ICDRS register, and the TDRE flag is again set 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 in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 0, the RIIC continues in master transmit mode.
Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to generate a stop condition.
For data transmission with an address in the 10-bit format, start by writing 1111 0b, the 2 higher-order bits of the slave address, and W to the ICDRT register as the first address transmission. Then, as the second address transmission, write the 8 lower-order bits of the slave address to the ICDRT register.
(4) After confirming that the ICSR2.TDRE flag is 1, write the data for transmission to the ICDRT register. The RIIC automatically holds the SCL0 line low until the data for transmission are ready or a stop condition is generated.
(5) After all bytes of data for transmission have been written to the ICDRT register, wait until the value of the ICSR2.TEND flag returns to 1, and then set the ICCR2.SP bit to 1 (requests to generate a stop condition). Upon receiving a stop condition generation request, the RIIC generates the stop condition.
(6) Upon detecting the stop condition, the RIIC automatically sets bits MST and TRS in the ICCR2 register to 00b and enters slave receive mode. Furthermore, it automatically sets the TDRE and TEND flags to 0, and sets the ICSR2.STOP flag 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.

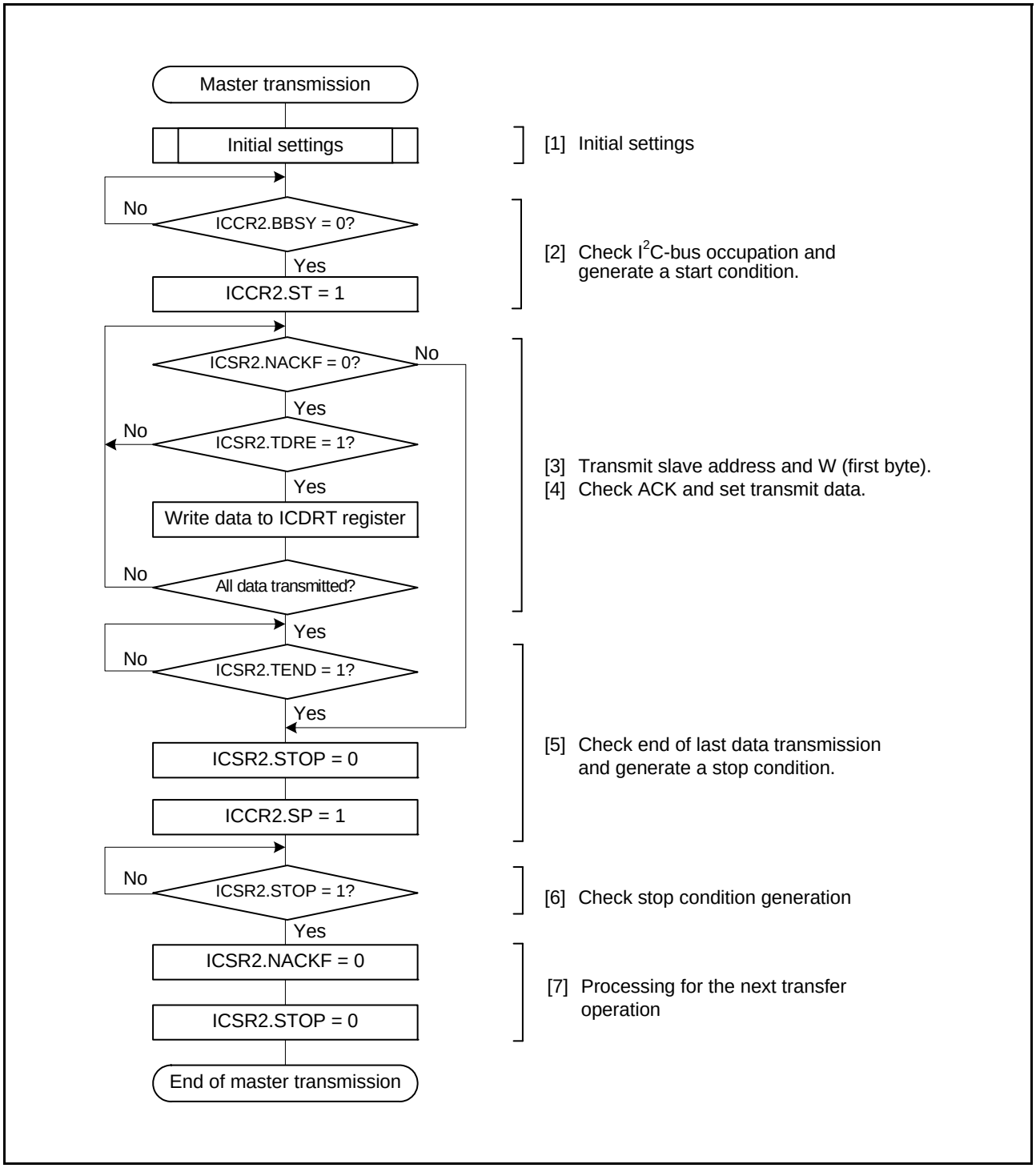


Figure 24.6 Example of Master Transmission Flowchart

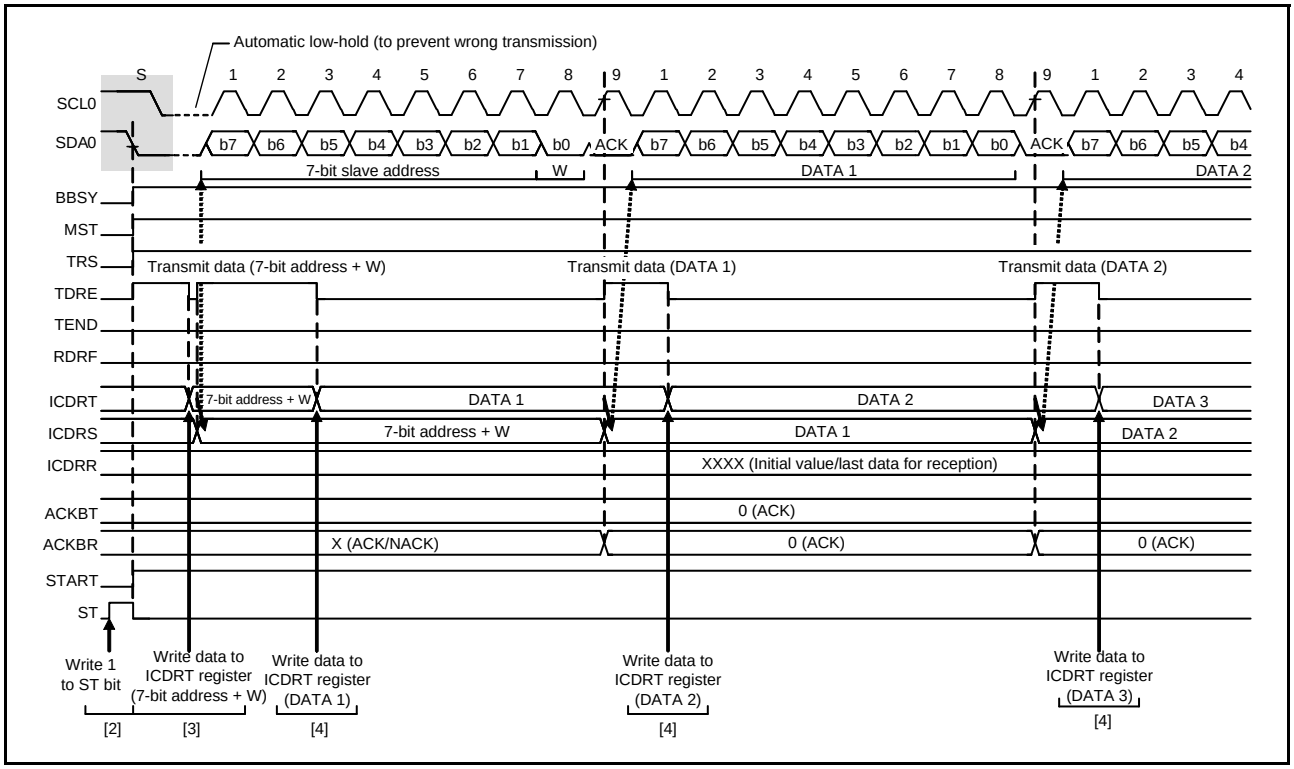


Figure 24.7 Master Transmit Operation Timing (1) (7-Bit Address Format)

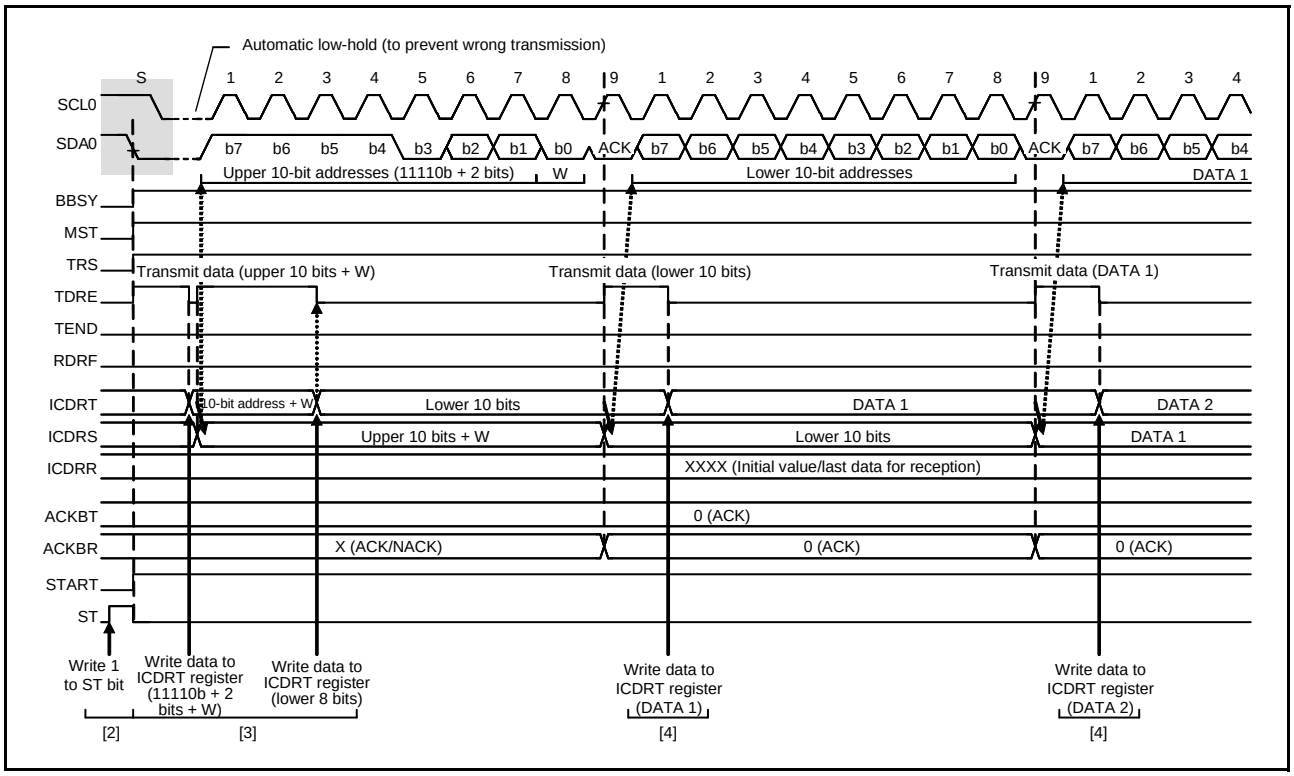


Figure 24.8 Master Transmit Operation Timing (2) (10-Bit Address Format)

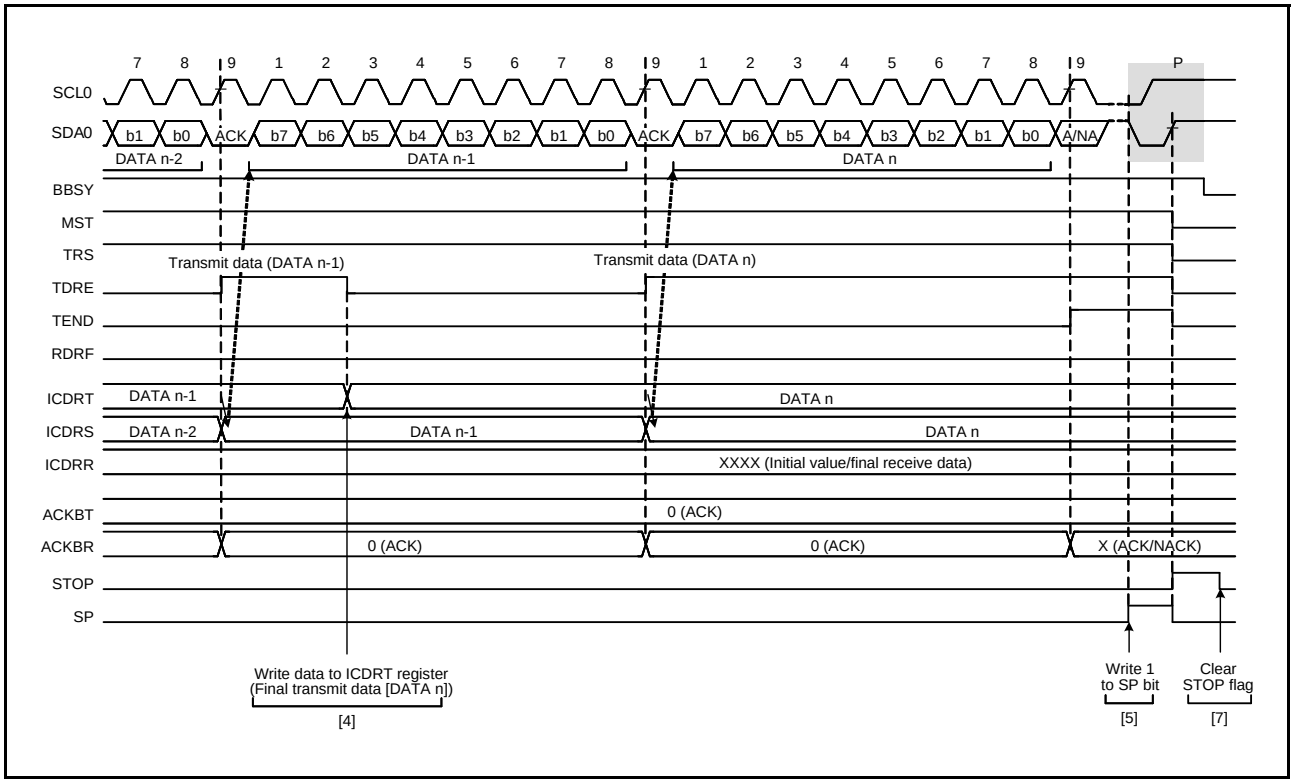


Figure 24.9 Master Transmit Operation Timing (3)

24.3.4 Master Receive Operation

In master receive operation, the RIIC as a master device generates clock signals, receives data from the slave device, and returns acknowledgments. Because the RIIC must start by sending a slave address to the corresponding slave device, this part of the procedure is performed in master transmit mode, but the subsequent steps are in master receive mode.
Figure 24.10 and Figure 24.11 show examples of usage of master reception (7-bit address format) and Figure 24.12 to Figure 24.14 show the timing of operations in master reception.
The following describes the procedure and operations for master reception.

(1) Initial settings. For details, refer to section 24.3.2, Initial Settings.

(2) Read the ICCR2.BBSY flag to check that the bus is open, and then set the ICCR2.ST bit to 1 (requests to generate a start condition). Upon receiving the request, the RIIC generates a start condition. When the RIIC detects the start condition, the BBSY flag and the ICSR2.START flag are automatically set to 1 and the ST bit is automatically set to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDA0 line have matched while the ST bit is 1, the RIIC recognizes that generating of the start condition as requested by the ST bit has been successfully completed, and bits MST and TRS in the ICCR2 register are automatically set to 1, placing the RIIC in master transmit mode. The ICSR2.TDRE flag is also automatically set to 1 in response to setting of the TRS bit to 1.

(3) Check that the ICSR2.TDRE flag is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to the ICDRT register. Once the data for transmission are written to the ICDRT register, the TDRE flag is automatically set to 0, the data are transferred from the ICDRT register to the ICDRS register, and the TDRE flag is again set to 1. Once the byte containing the slave address and R/W# bit has been transmitted, the value of the ICCR2.TRS bit is automatically updated to select transmit or receive mode in accord with the value of the transmitted R/W# bit. If the value of the R/W# bit was 1, the TRS bit is set to 0 on the rising edge of the ninth SCL, placing the RIIC in master receive mode. At this time, the TDRE flag is set to 0 and the ICSR2.RDRF flag is automatically set to 1.

Because the ICSR2.NACKF flag being 1 at this time indicates that no slave device recognized the address or there was an error in communications, write 1 to the ICCR2.SP bit to generate a stop condition.
For master reception from a device with a 10-bit address, start by using master transmission to send the 10-bit address, and then generate a restart condition. After that, transmitting 1111 0b, the two higher-order bits of the slave address, and the R bit places the RIIC in master receive mode.

(4) Dummy read the ICDRR register after confirming that the ICSR2.RDRF flag is 1; this makes the RIIC start output of the SCL and start data reception.

(5) After 1 byte of data has been received, the ICSR2.RDRF flag is set to 1 on the rising edge of the eighth or ninth SCL (the clock signal) as selected by the ICMR3.RDRFS bit. Reading the ICDRR register at this time will produce the received data, and the RDRF flag is automatically set to 0 at the same time. Furthermore, the value of the acknowledgment bit received during the ninth SCL is returned as the value set in the ICMR3.ACKBT bit. Furthermore, if the next byte to be received is the next to last byte, set the ICMR3.WAIT bit to 1 (for wait insertion) before reading the ICDRR register (containing the second byte from last). As well as enabling NACK output even in the case of delays in processing to set the ICMR3.ACKBT bit to 1 (NACK) in step (6), due to other interrupts, etc., this fixes the SCL0 line to low on the falling edge of the ninth clock pulse in reception of the last byte, so the state is such that generating a stop condition is possible.

(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 (final) byte, set the ICMR3.ACKBT bit to 1 (NACK).

(7) After reading the byte before last from the ICDRR register, if the value of the ICSR2.RDRF flag is confirmed to be 1, write 1 to the ICCR2.SP bit (requests to generate a stop condition) and then read the last byte from the ICDRR register. When the ICDRR register is read, the RIIC is released from the wait state and generates the stop condition after low-level output in the ninth clock pulse is completed or the SCL0 line is released from the low-hold state.

(8) Upon detecting the stop condition, the RIIC automatically sets bits MST and TRS in the ICCR2 register to 00b and enters slave receive mode. Furthermore, detection of the stop condition leads to setting of the ICSR2.STOP flag to 1.

(9) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

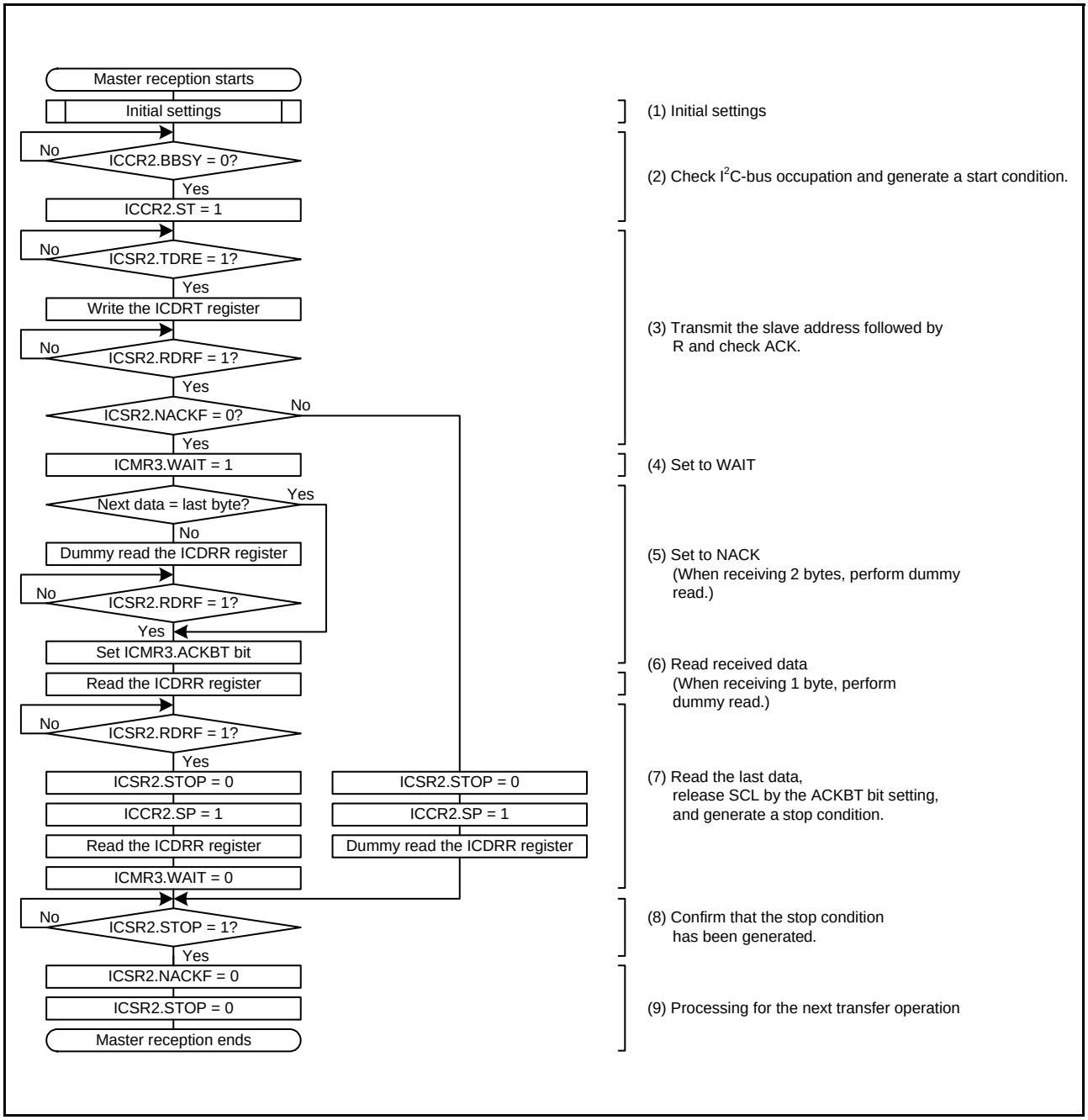


Figure 24.10 Example of Master Reception (7-Bit Address Format, 1 or 2 bytes)

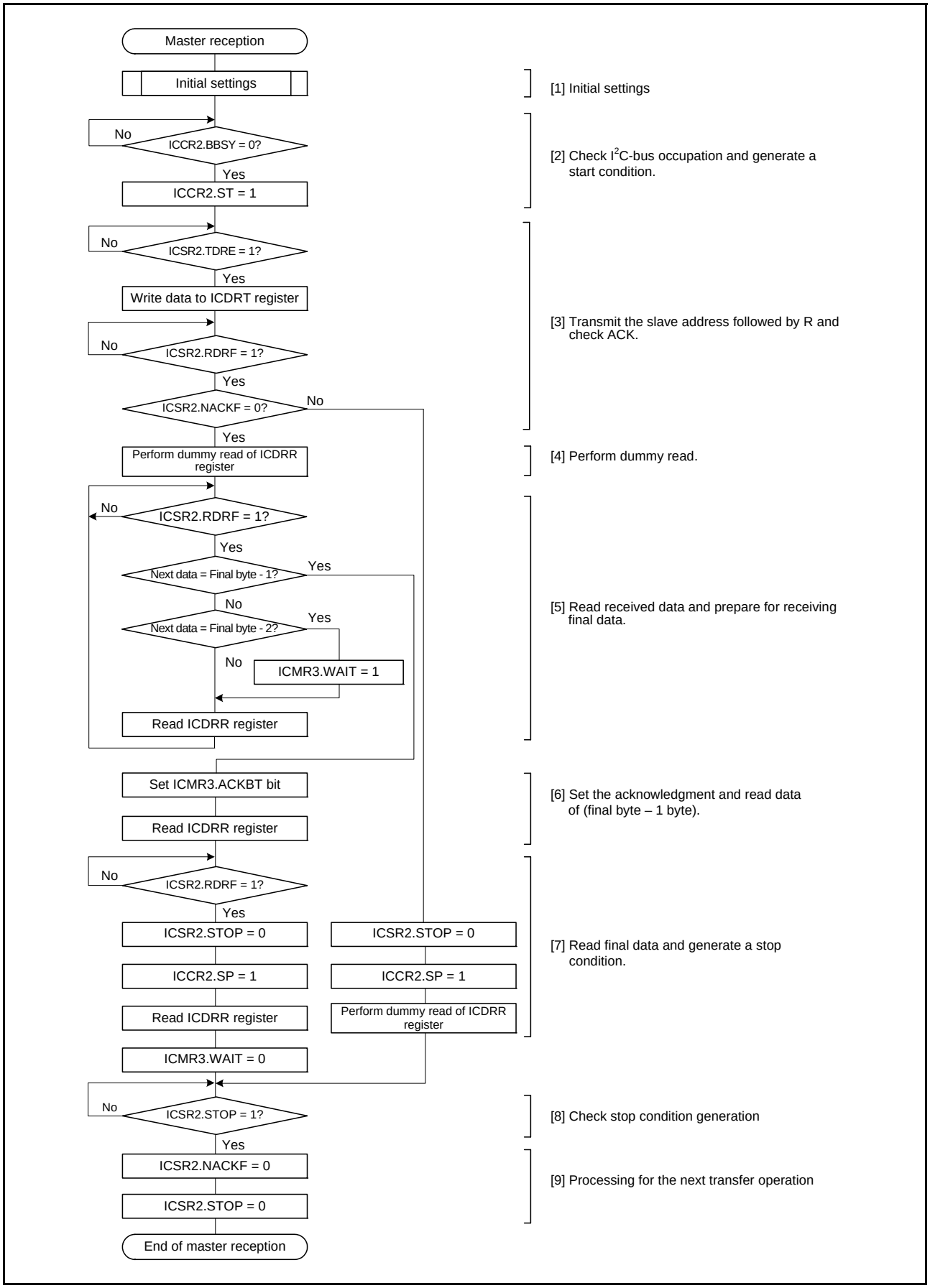


Figure 24.11 Example of Master Reception (7-Bit Address Format, 3 Bytes or More)

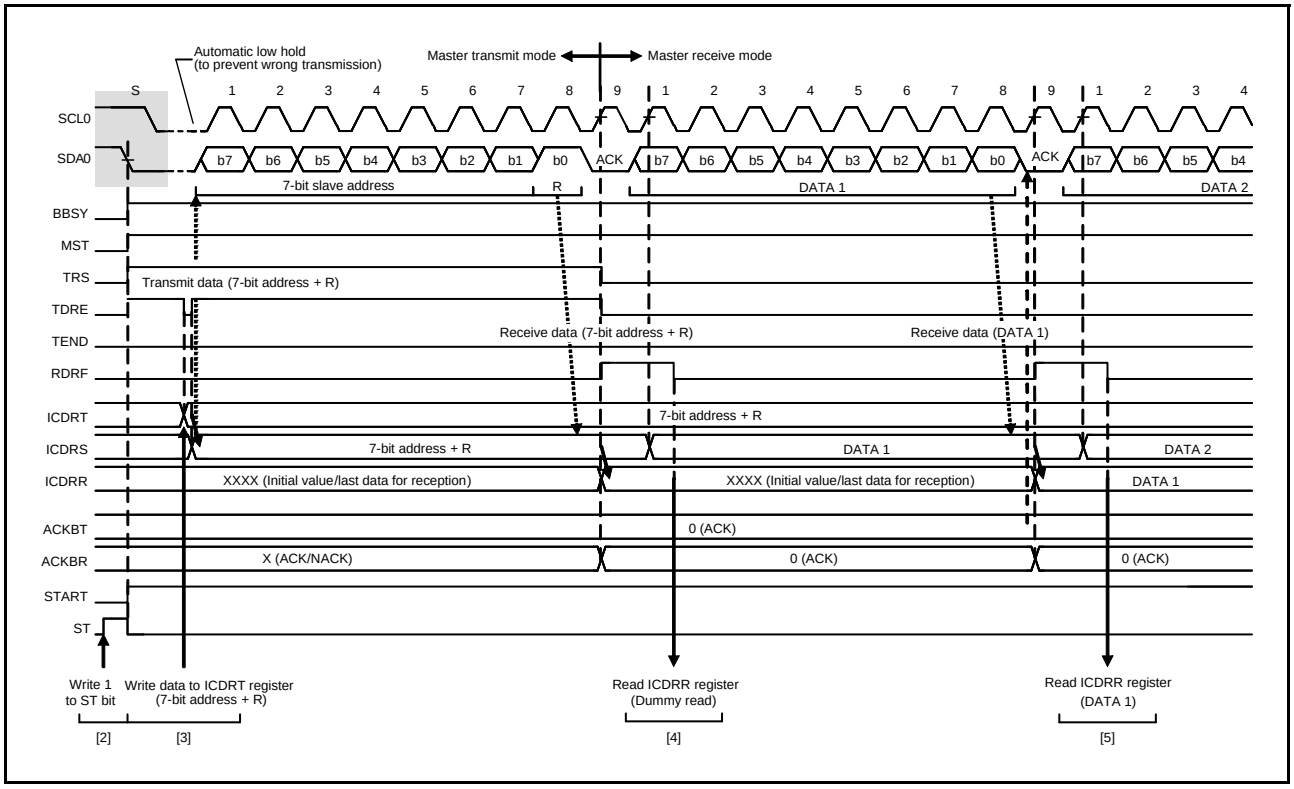


Figure 24.12 Master Receive Operation Timing (1) (7-Bit Address Format, When RDRFS bit is 0)

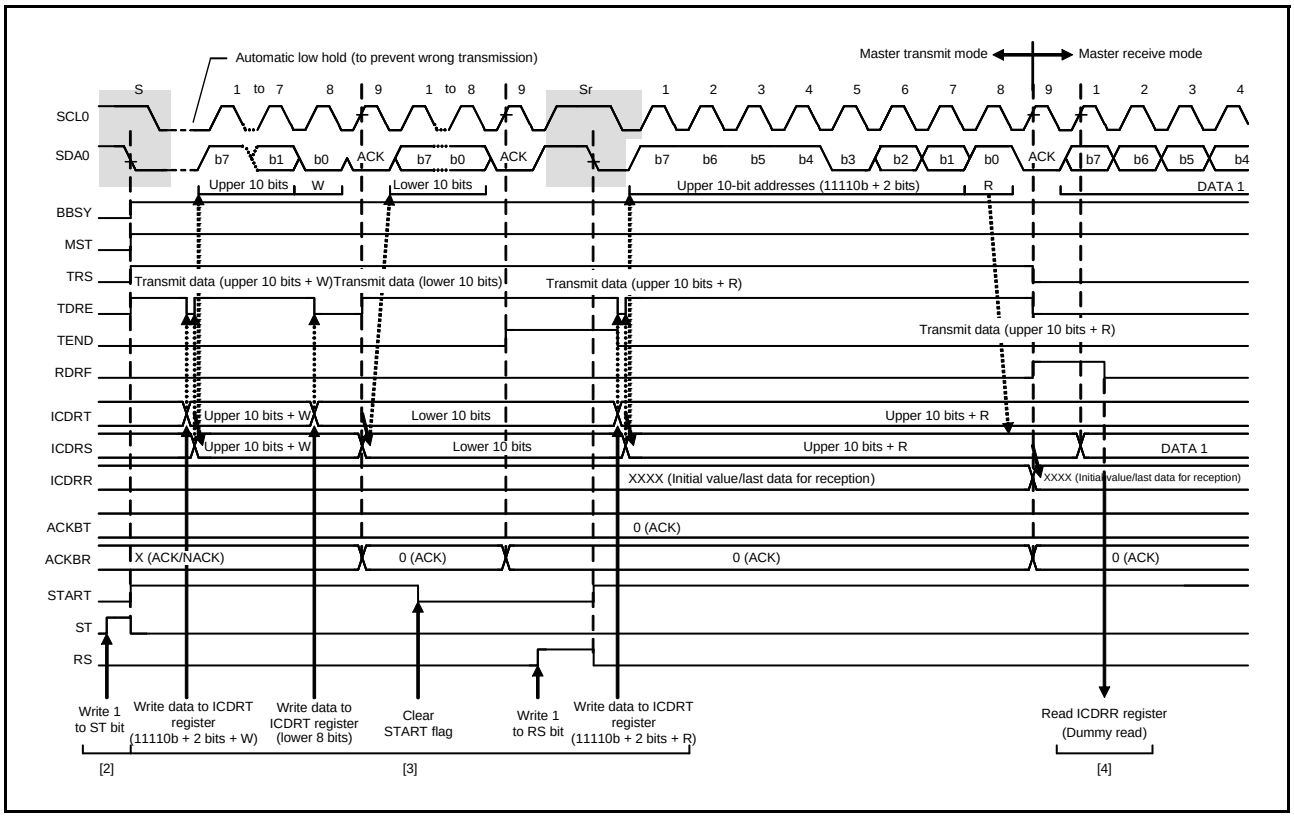


Figure 24.13 Master Receive Operation Timing (2) (10-Bit Address Format, When RDRFS bit is 0)

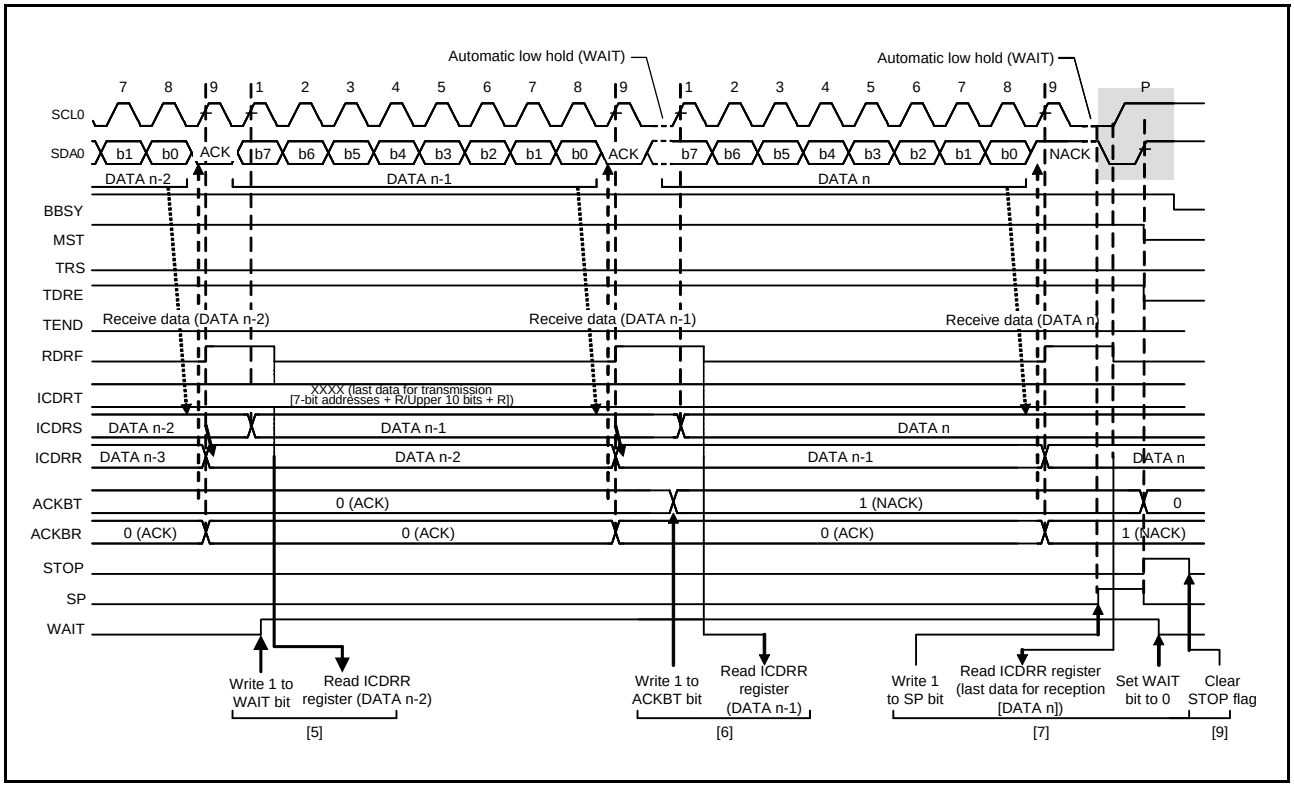


Figure 24.14 Master Receive Operation Timing (3) (When RDRFS bit is 0)

24.3.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL, the RIIC transmits data as a slave device, and the master device returns acknowledgments.

Figure 24.15 shows an example of usage of slave transmission and Figure 24.16 and Figure 24.17 show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

(1) Initial settings. For details, refer to section 24.3.2, Initial Settings.
After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth SCL (the clock signal) and outputs the value set in the ICMR3.ACKBT bit to the acknowledgment bit on the ninth SCL. If the value of the R/W# bit that was also received at this time is 1, the RIIC automatically places itself in slave transmit mode by setting both the ICCR2.TRS bit and the ICSR2.TDRE flag to 1.

(3) After the ICSR2.TDRE flag is confirmed to be 1, write the data for transmission to the ICDRT register. At this time, if the RIIC does not receive ACK from the master device (receives an NACK signal) while the ICFER.NACKE bit is 1, the RIIC 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 RIIC drives the SCL0 line low on the falling edge of ninth SCL.

(5) When the ICSR2.NACKF flag or the ICSR2.TEND flag is 1, dummy read the ICDRR register to complete the processing. This releases the SCL0 line.

(6) Upon detecting the stop condition, the RIIC automatically sets bits ICSR1.HOA, GCA, and AASy (y = 0 to 2), flags ICSR2.TDRE and TEND, and the ICCR2.TRS bit to 0, and enters slave receive mode.

(7) After checking that the ICSR2.STOP flag is 1, set the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

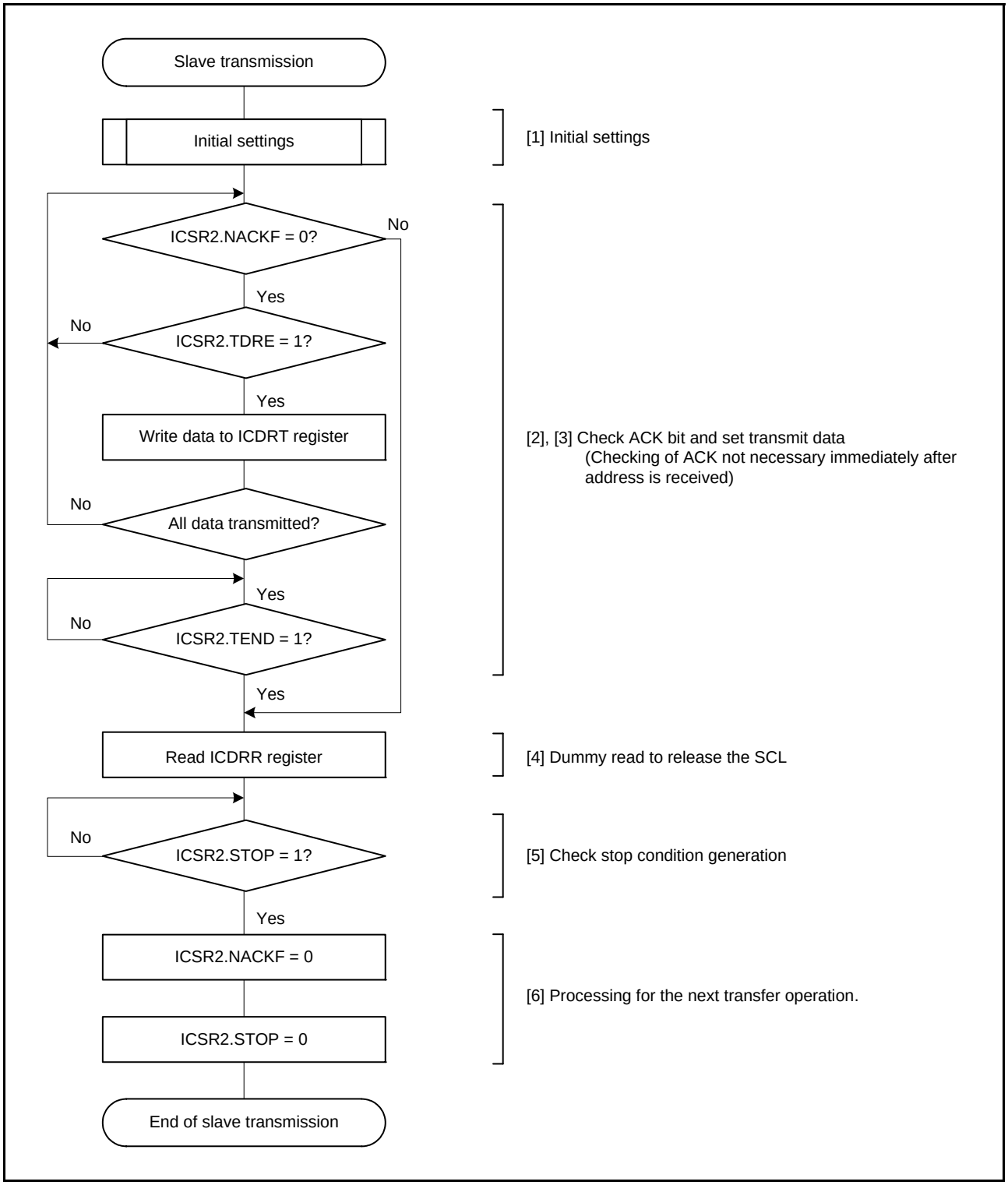


Figure 24.15 Example of Slave Transmission Flowchart

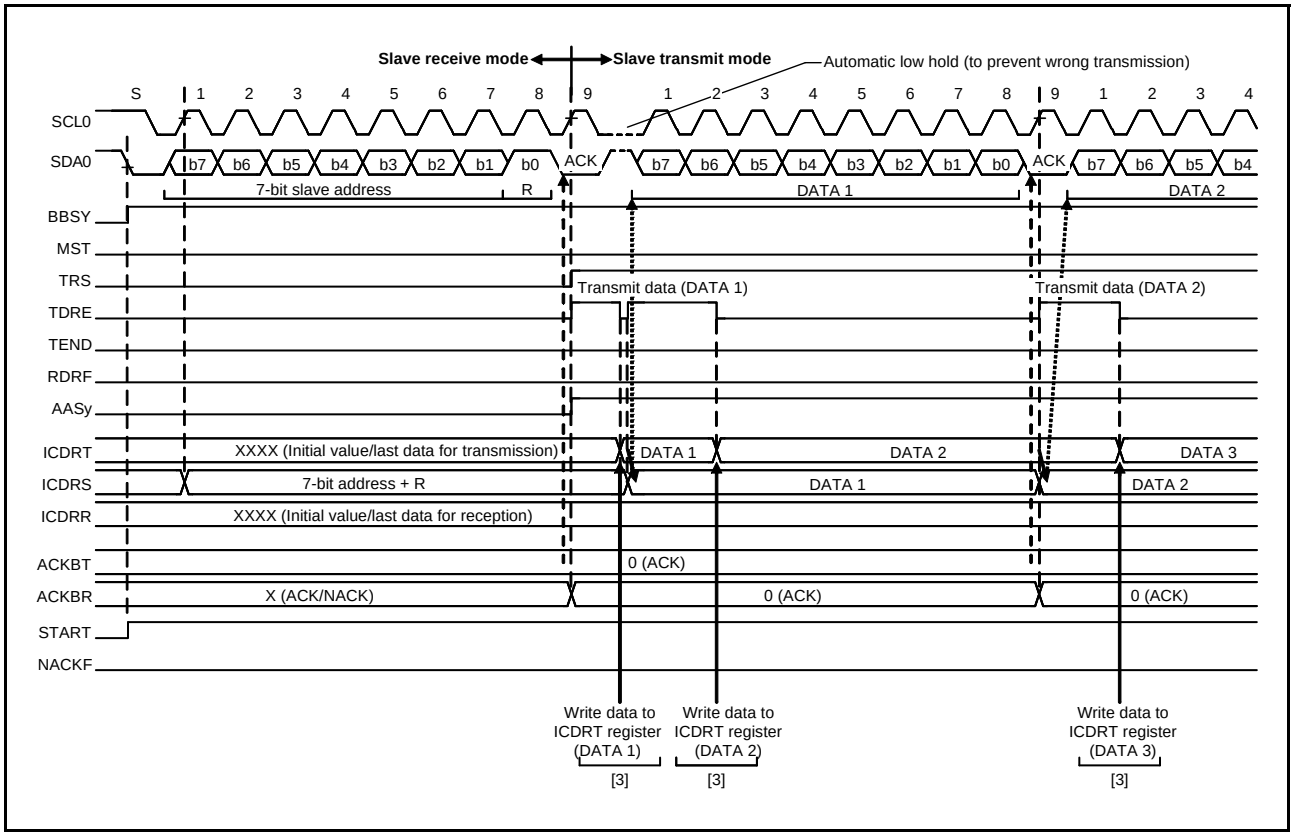


Figure 24.16 Slave Transmit Operation Timing (1) (7-Bit Address Format)

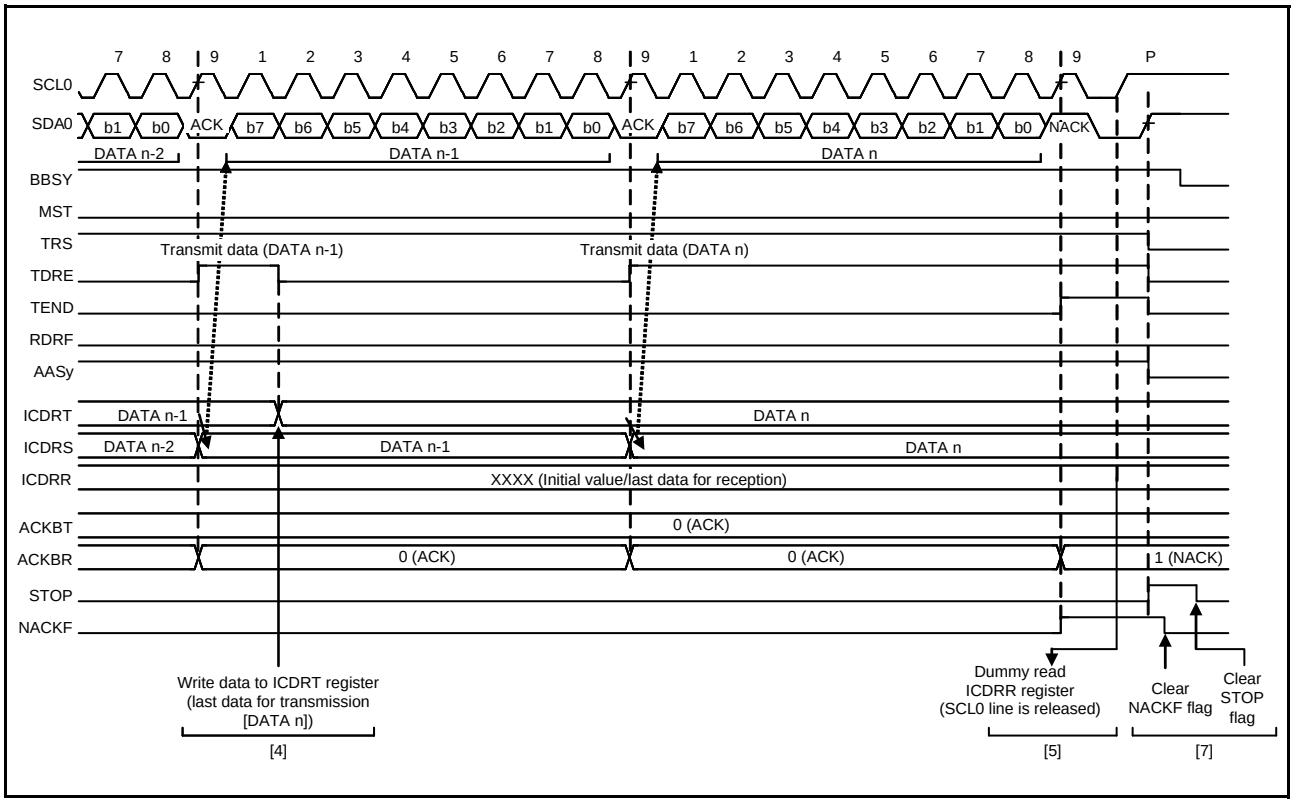


Figure 24.17 Slave Transmit Operation Timing (2)

24.3.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL and transmit data, and the RIIC returns acknowledgments as a slave device.

Figure 24.18 shows an example of usage of slave reception and Figure 24.19 and Figure 24.20 show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

(1) Initial settings. For details, refer to section 24.3.2, Initial Settings.
After initial settings, the RIIC will stay in the standby state until it receives a slave address that it matches.

(2) After receiving a matching slave address, the RIIC sets one of the corresponding bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 1 on the rising edge of the ninth SCL (the clock signal) and outputs the value set in the ICMR3.ACKBT bit to the acknowledgment bit on the ninth SCL. If the value of the R/W# bit that was also received at this time is 0, the RIIC continues to place itself in slave receive mode and sets the ICSR2.RDRF flag to 1.

(3) After the ICSR2.STOP flag is confirmed to be 0 and the ICSR2.RDRF flag to be 1, 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 the ICDRR register is read, the RIIC automatically sets the ICSR2.RDRF flag to 0. If reading of the ICDRR register is delayed and a next byte is received while the RDRF flag is still set to 1, the RIIC holds the SCL0 line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading the ICDRR register releases the SCL0 line from being held low.
When the ICSR2.STOP flag is 1 and the ICSR2.RDRF flag is also 1, read the ICDRR register until all the data is completely received.

(5) Upon detecting the stop condition, the RIIC automatically clears bits ICSR1.HOA, GCA, and AASy (y = 0 to 2) to 0.

(6) After checking that the ICSR2.STOP flag is 1, set the ICSR2.STOP flag to 0 for the next transfer operation.

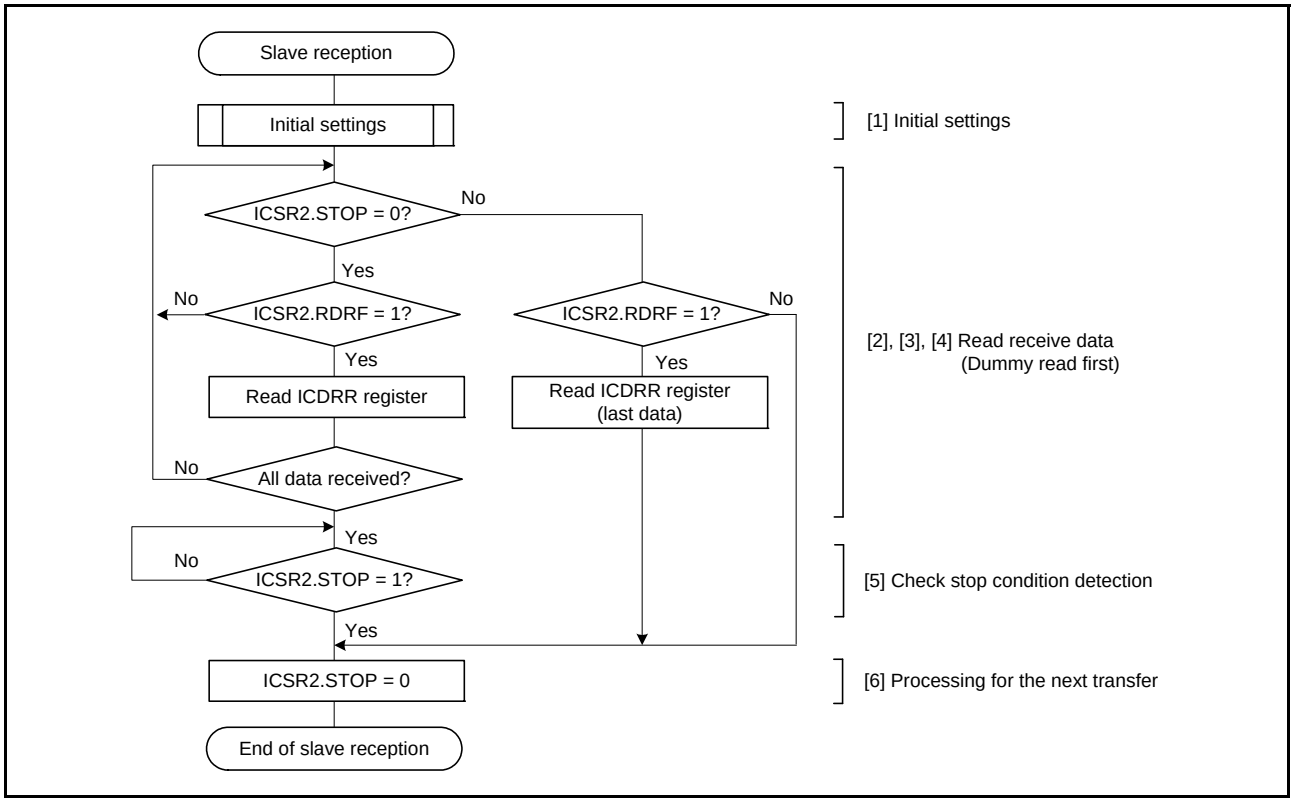


Figure 24.18 Example of Slave Reception Flowchart

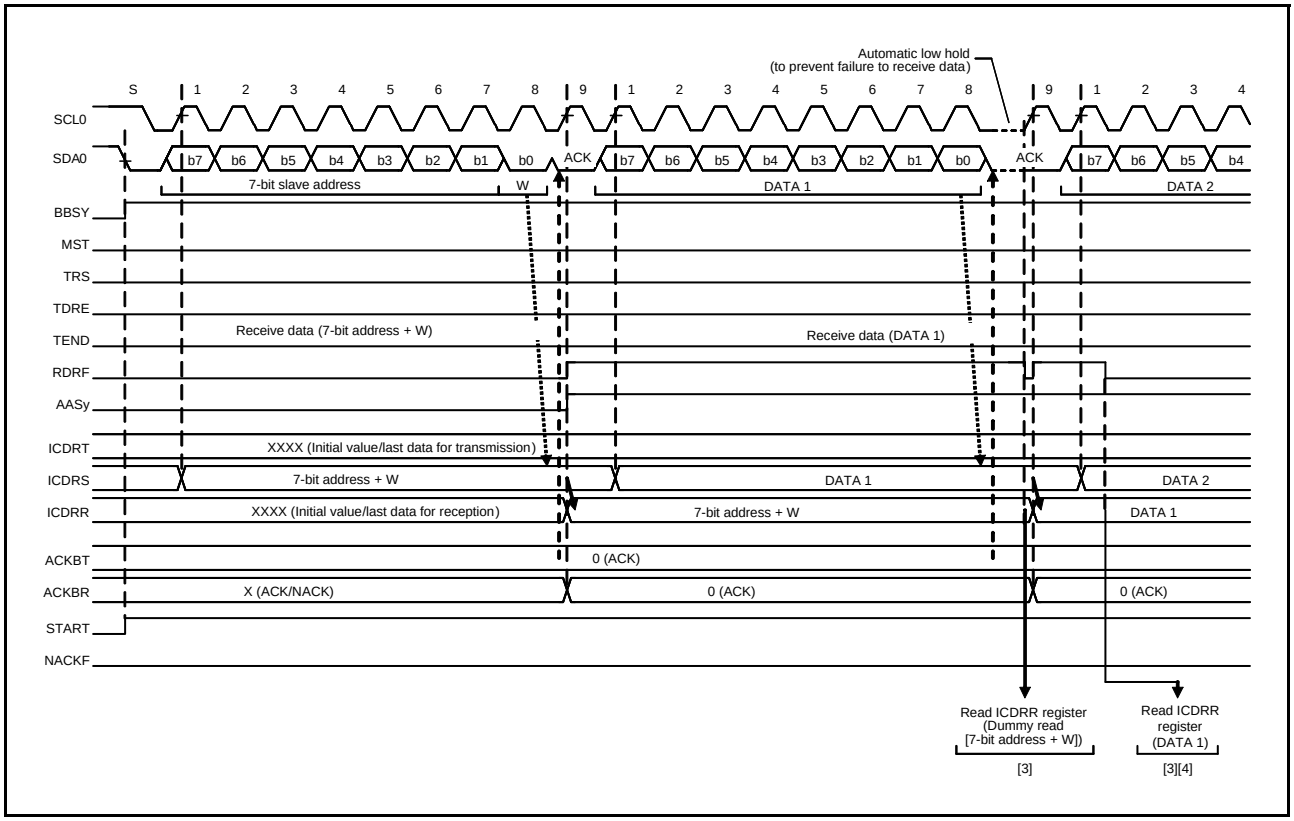


Figure 24.19 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS bit is 0)

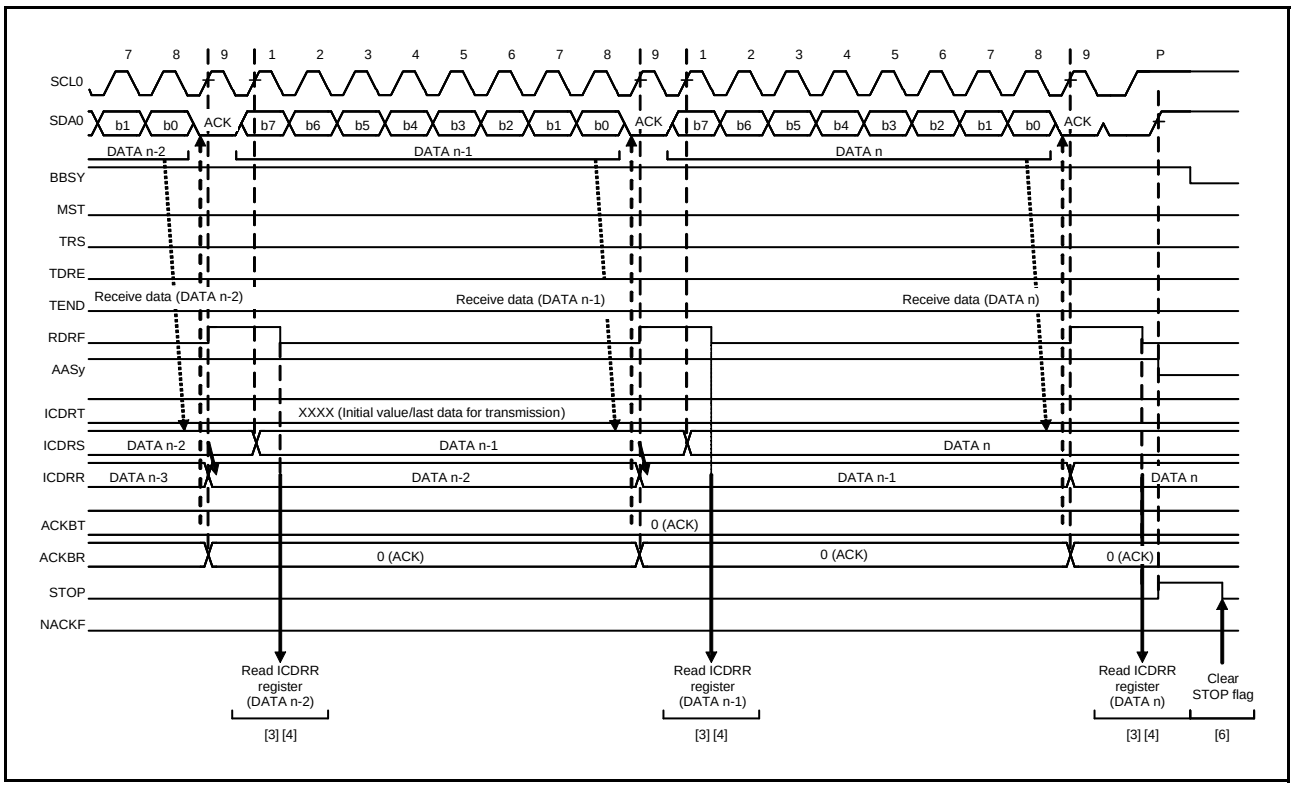


Figure 24.20 Slave Receive Operation Timing (2) (when RDRFS bit is 0)

24.4 SCL Synchronization Circuit

In generation of the SCL, the RIIC starts counting out the value for width at high level specified in the ICBRH register when it detects a rising edge on the SCL0 line and drives the SCL0 line low once counting of the width at high level is complete. When the RIIC detects the falling edge of the SCL0 line, it starts counting out the width at low period specified in the ICBRL register, and then stops driving the SCL0 line (releases the line) once counting of the width at low level is complete. The SCL is thus generated.

If multiple master devices are connected to the I2C-bus, a collision of SCL signals may arise due to contention with another master device. In such cases, the master devices have to synchronize their SCL signals. Since synchronization of the SCL signals must be handled bit by bit, the RIIC is equipped with a facility (the SCL synchronization circuit) for obtaining bit-by-bit synchronization of the SCL signals by monitoring the SCL0 line while in master mode.

When the RIIC has detected a rising edge on the SCL0 line and thus started counting out the width at high level specified in the ICBRH register, and the level on the SCL0 line falls because an SCL signal is being generated by another master device, the RIIC stops counting when it detects the falling edge, drives the level on the SCL0 line low, and starts counting out the width at low level specified in the ICBRL register. When the RIIC finishes counting out the width at low level, it stops driving the SCL0 line low (i.e. releases the line). At this time, if the width at low level of the SCL signal from the other master device is longer than the width at low level set in the RIIC, the width at low level of the SCL signal will be extended. Once the width at low level for the other master device has ended, the SCL signal rises because the SCL0 line has been released. When the RIIC finishes outputting the low period of the SCL, the SCL0 line is released and the SCL rises. That is, in cases of contention of SCL signals from more than one master, the width at high level of the SCL signal is synchronized with that of the clock having the narrower width, and the width at low level of the SCL signal is synchronized with that of the clock having the broader width. However, such synchronization of the SCL signal is only enabled when the ICFER.SCLE bit is set to 1.

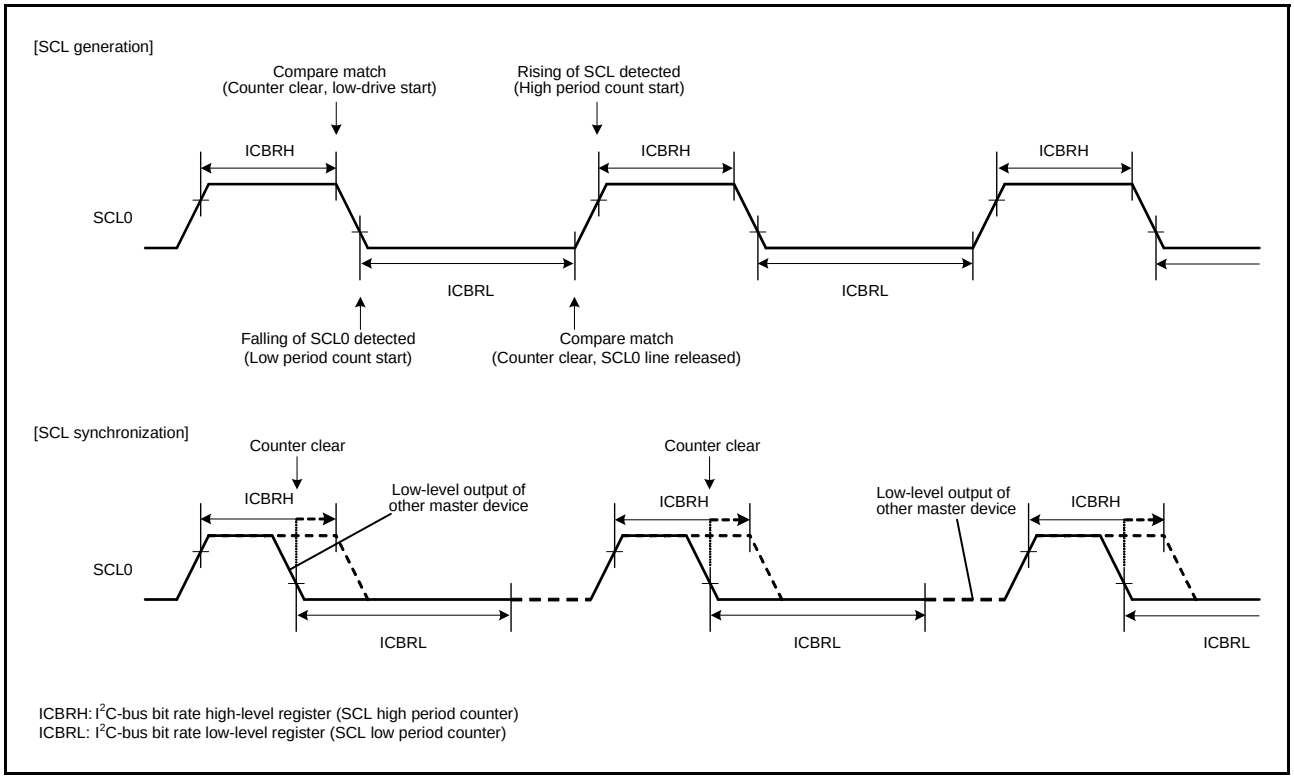


Figure 24.21 Generation and Synchronization of the SCL Signal from the RIIC

24.5 SDA Output Delay Function

The RIIC module incorporates a function for delaying output on the SDA line. The delay can be applied to all output (generation of the start, restart, and stop conditions, data, and the ACK and NACK signals) on the SDA line.

The SDA output delay function is used to delay the SDA output timing relative to falling edges of SCL to ensure that the SDA signal changes while the SCL is low and can be used to prevent erroneous operation of communications devices. This function is also used to satisfy the 300 ns (min.) data hold time prescribed by the SMBus specification.

The output delay function is enabled by setting the ICMR2.SDDL[2:0] bits to any value other than 000b, and disabled by setting the same bits to 000b.

When the SDA output delay function is enabled (the SDDL[2:0] bits are not “000b”), the SDA output delay counter counts the number of cycles set in the SDDL[2:0] bits of the count source selected by the ICMR2.DLCS bit (the internal reference clock (IICφ) or internal reference clock divided by 2 (IICφ/2)). On completion of counting of cycles of delay, the RIIC changes the bit being output as the SDA signal (generation of the start, restart, or stop condition, a new bit, or an ACK or NACK signal).

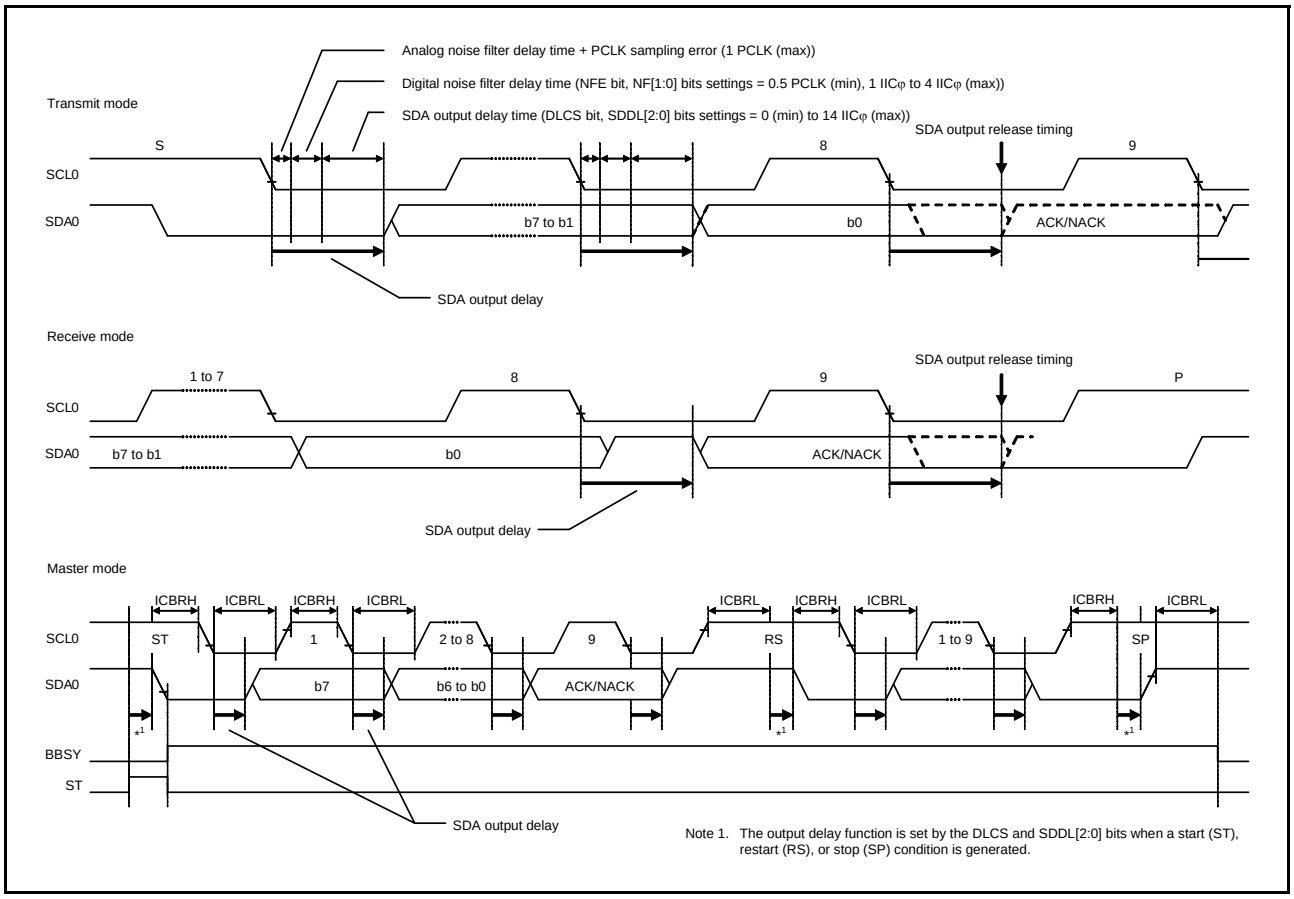


Figure 24.22 SDA Output Delay Function

24.6 Digital Noise Filters

The states of the SCL0 and SDA0 pins are conveyed to the internal circuitry through analog noise filters and digital noise filters. Figure 24.23 is a block diagram of the digital noise filter.

The on-chip digital noise filter of each RIIC consists of four flip-flop circuit stages connected in series and a match detection circuit.

The number of effective stages in the digital noise filter is selected by the ICMR3.NF[1:0] bits. The selected number of effective stages determines the noise-filtering capability as a period from one to four cycles of IICφ.

The input signal to the SCL0 pin (or SDA0 pin) is sampled on falling edges of the IICφ signal. When the input signal level matches the output level for the number of effective flip-flop circuit stages selected by the ICMR3.NF[1:0] bits, the signal level is conveyed to the subsequent stages. If the signal levels do not match, the previous value is retained.

If the ratio between the frequency of the internal operating clock (PCLK) and the transfer rate is relatively small (e.g. data transfer at 400 kbps with PCLK = 4 MHz), the characteristics of the digital noise filter may lead to the elimination of needed signals as noise. In such cases, it is possible to disable the digital noise filter (by setting the ICFER.NFE bit to 0) and use only the analog noise filter.

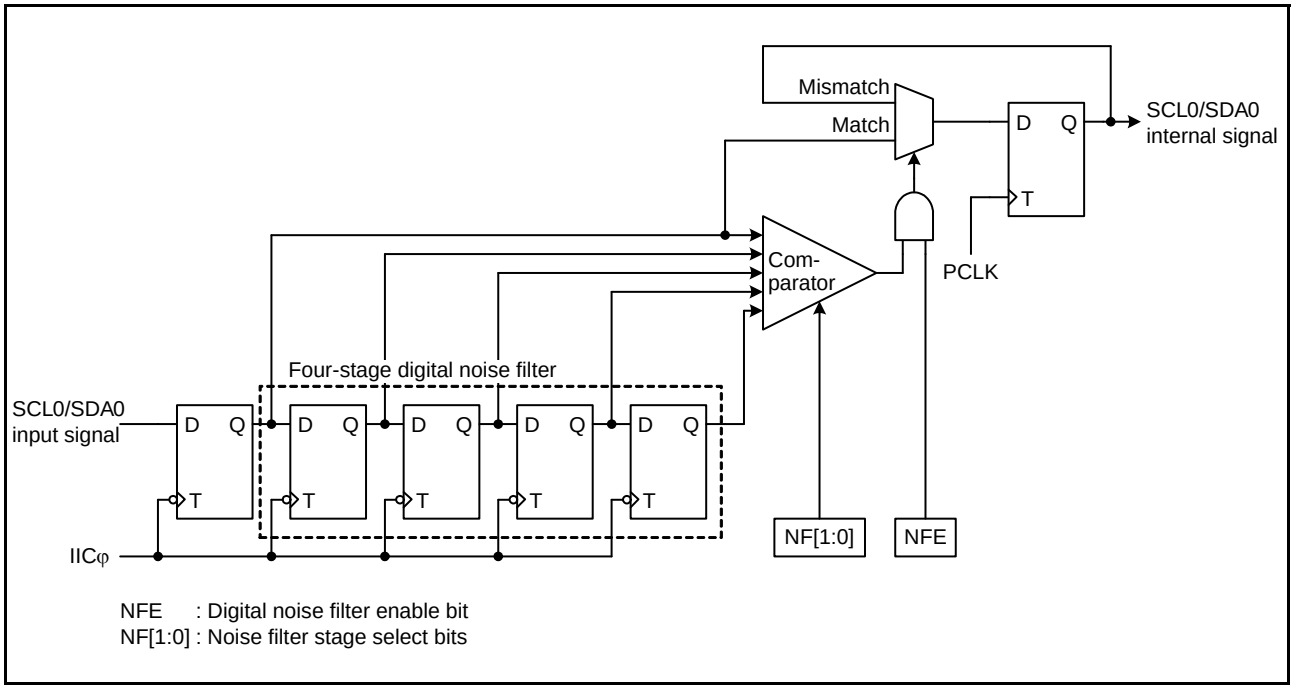


Figure 24.23 Block Diagram of the Digital Noise Filter

24.7 Address Match Detection

The RIIC can set three unique slave addresses in addition to the general call address and host address, and also can set 7-bit or 10-bit slave addresses.

24.7.1 Slave-Address Match Detection

The RIIC can set three unique slave addresses, and has a slave address detection function for each unique slave address. When the ICSER.SARyE bit (y = 0 to 2) is set to 1, the slave addresses set in registers SARUy and SARLy (y = 0 to 2) can be detected.

When the RIIC detects a match with its set slave address, the corresponding ICSR1.AASy flag (y = 0 to 2) is set to 1 on the rising edge of the ninth SCL, and the ICSR2.RDRF flag or the ICSR2.TDRE flag is set to 1 according to the level of the R/W# bit. This causes a receive data full interrupt (RXI) or transmit data empty interrupt (TXI) to be generated. The AASy flag is used to identify which slave address has been specified.

Figure 24.24 to Figure 24.26 show the AASy flag set timing in three cases.

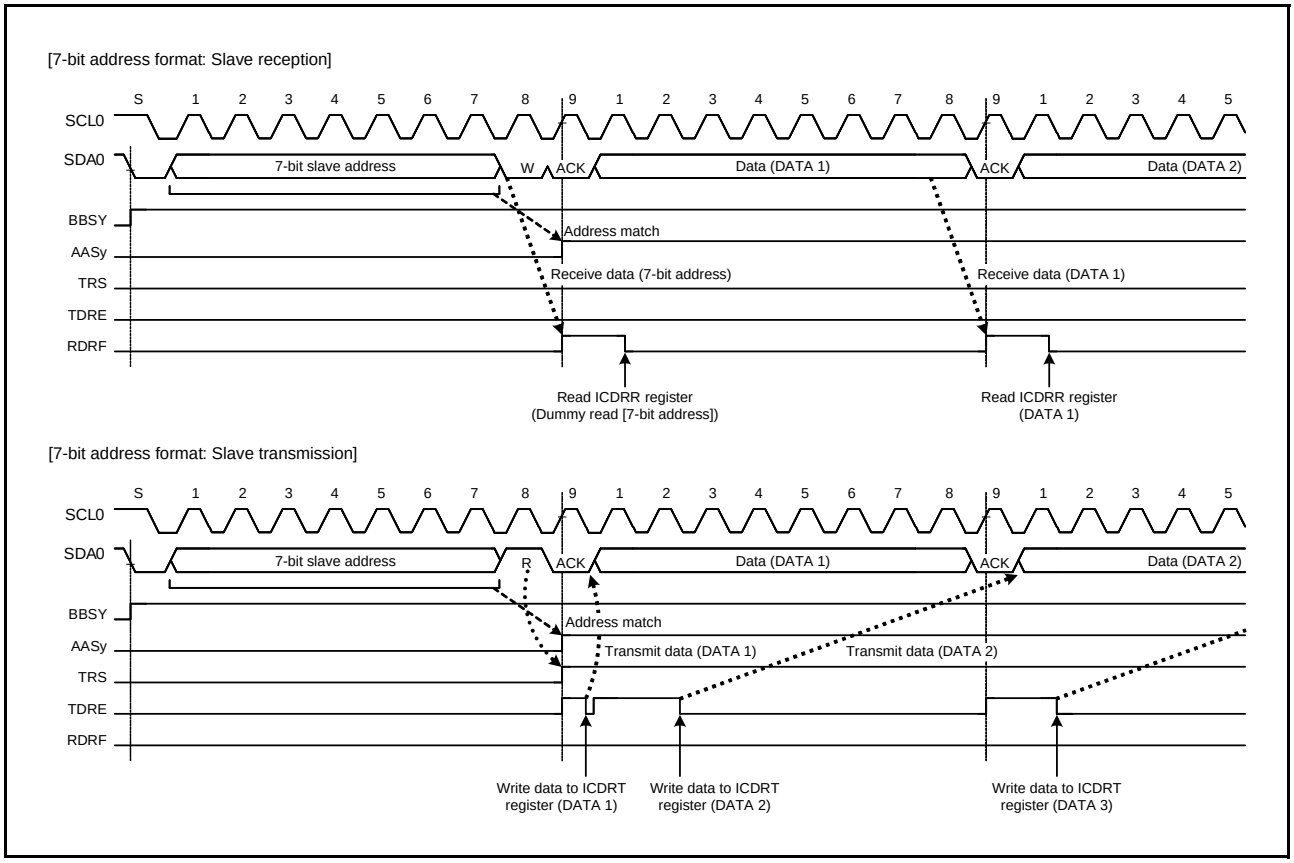


Figure 24.24 AASy Flag Set Timing with 7-Bit Address Format Selected

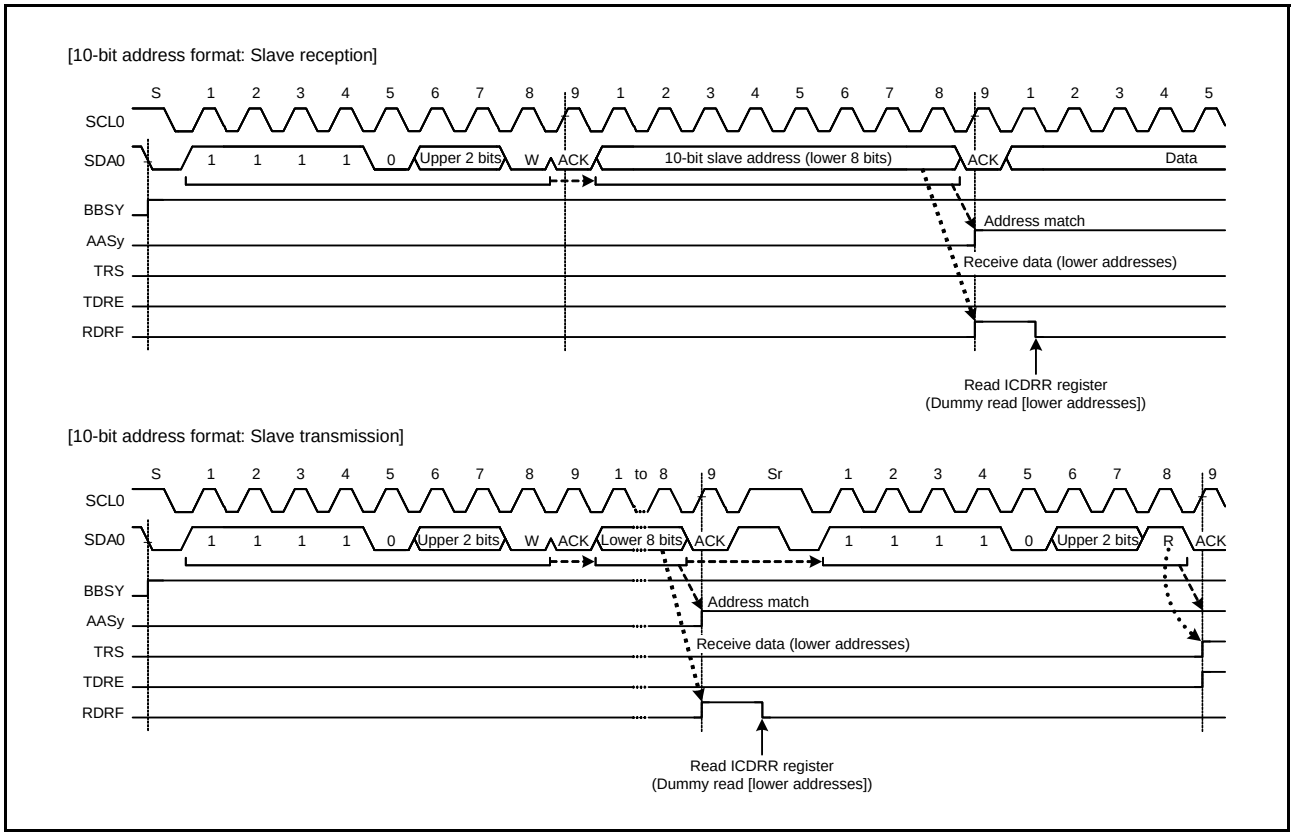


Figure 24.25 AASy Flag Set Timing with 10-Bit Address Format Selected

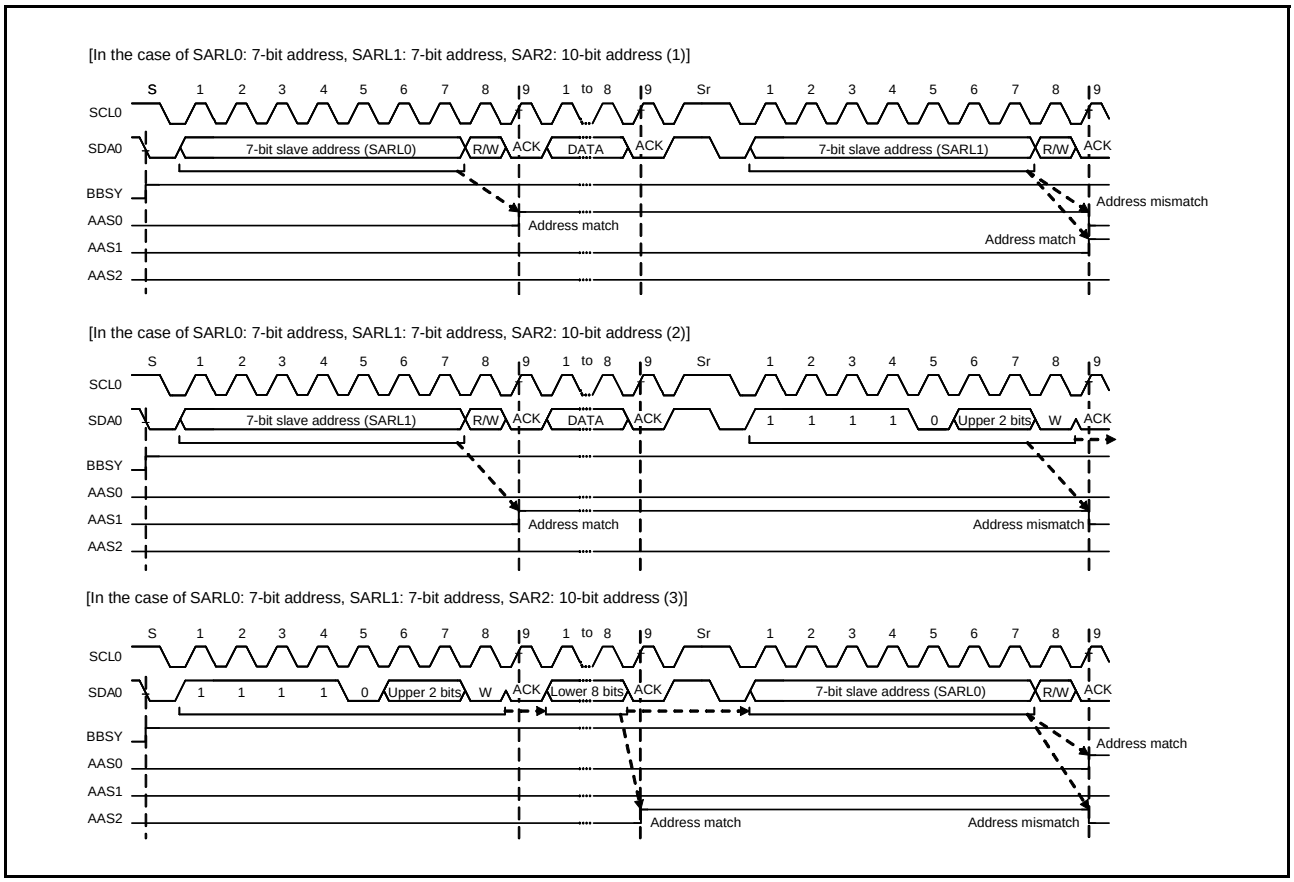


Figure 24.26 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

24.7.2 Detection of the General Call Address

The RIIC also has a facility for detecting the general call address (0000 000b + 0 (write)). This is enabled by setting the ICSER.GCAE bit to 1.

If the address following a start or restart condition is 0000 000b + 1 (read) (start byte), the RIIC recognizes this as the address of a slave device with an “all-zero” address but not as the general call address.

When the RIIC detects the general call address, both the ICSR1.GCA flag and the ICSR2.RDRF flag are set to 1 on the rising edge of the ninth SCL. This leads to the generation of a receive data full interrupt (RXI). The value of the GCA flag can be confirmed to recognize that the general call address has been transmitted.

Operation after detection of the general call address is the same as normal slave receive operation.

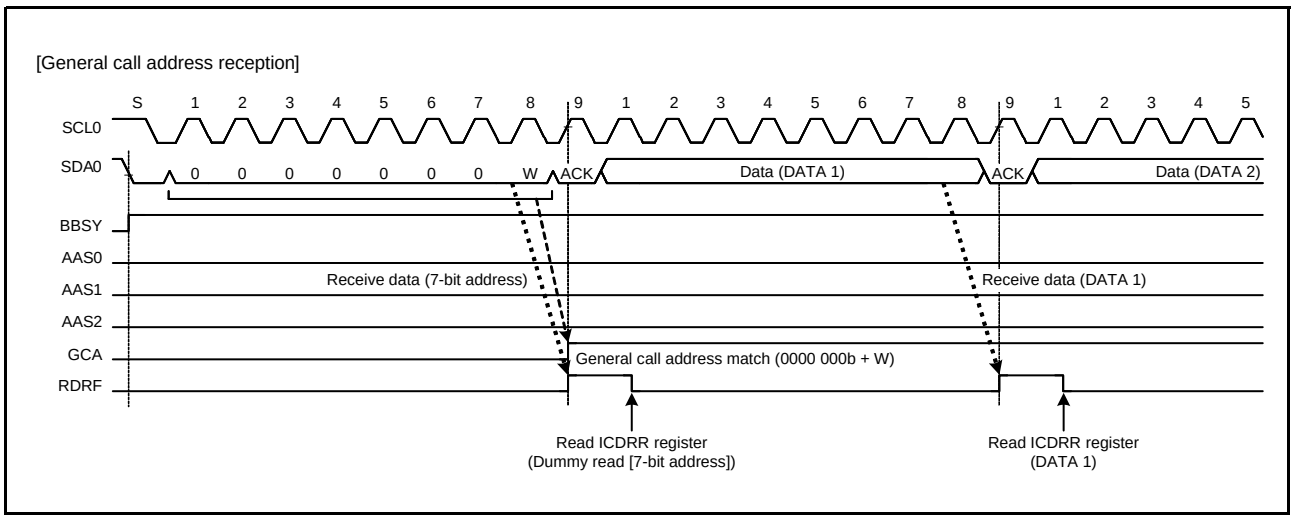


Figure 24.27 Timing of GCA Flag Setting during Reception of General Call Address

24.7.3 Device-ID Address Detection

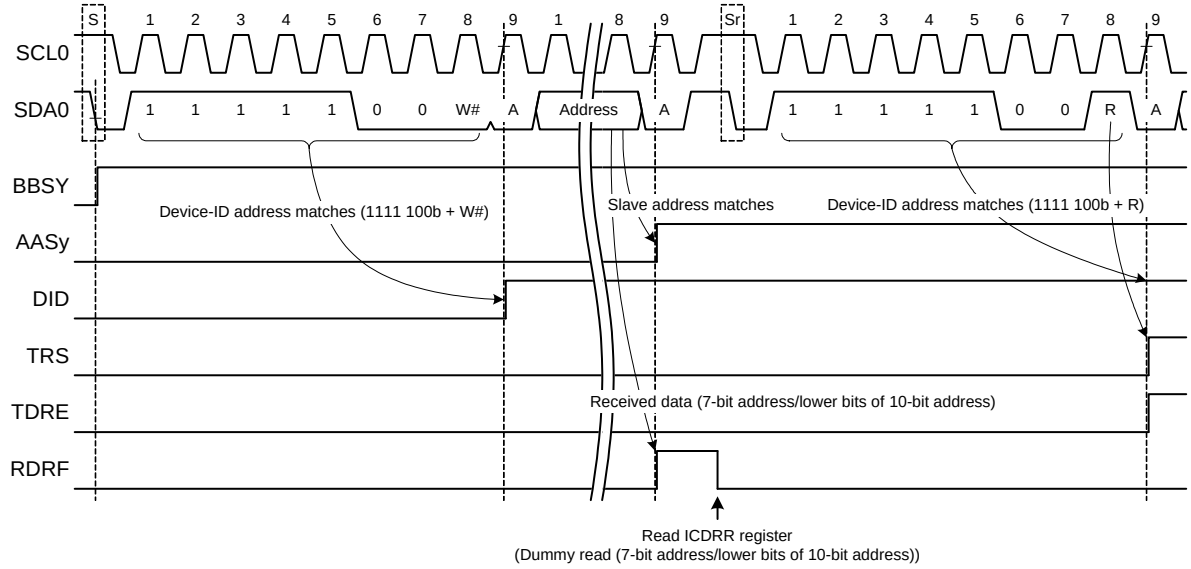
The RIIC module has a function to detect device-ID addresses complying with the I2C-bus specification. When the RIIC receives 1111 100b as the first seven bits of the first byte following a start condition or a restart condition while the ICSER.DIDE bit set to 1, the RIIC recognizes the address as a device-ID address, sets the ICSR1.DID flag to 1 on the rising edge of the ninth SCL when the following R/W# bit is 0, and then compares the second and following bytes with its own slave address. If the received address matches the value in the slave address register, the RIIC sets the corresponding ICSR1.AASy flag (y = 0 to 2) to 1.

After that, when the first byte received after a start or restart condition is generated matches the device ID address (1111 100b) again and the following R/W# bit is 1, the RIIC does not compare the second and subsequent bytes and sets the ICSR2.TDRE flag to 1.

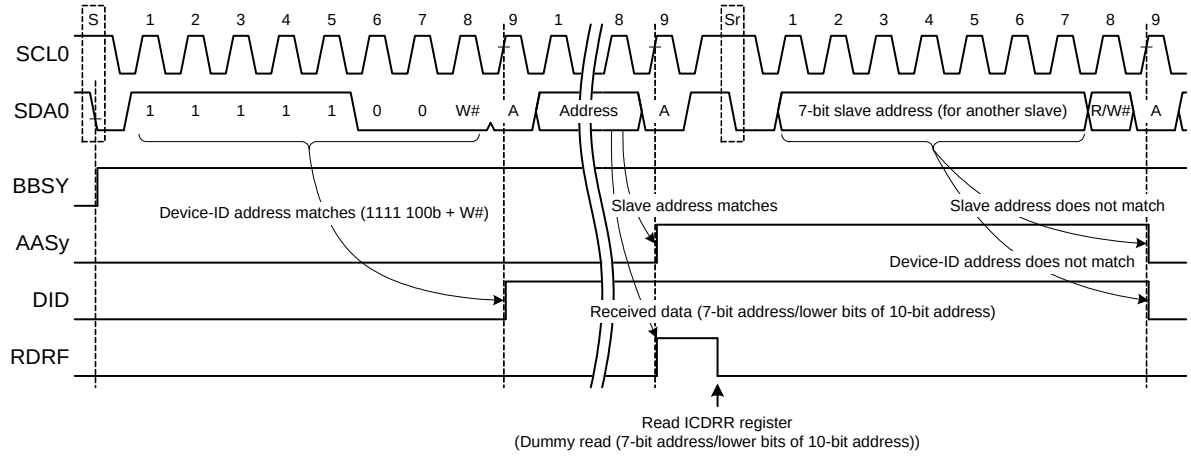
In the device-ID address detection function, the RIIC sets the DID flag to 0 if a match with the RIIC's own slave address is not obtained or a match with the device ID address is not obtained after a match with the RIIC's own slave address and the detection of a restart condition. If the first byte after detection of a start or restart condition matches the device ID address (1111 100b) and the R/W# bit is 0, the RIIC sets the DID flag to 1 and compares the second and subsequent bytes with the RIIC's slave address. If the R/W# bit is 1, the DID flag holds the previous value and the RIIC does not compare the second and subsequent bytes. Therefore, the reception of a device-ID address can be checked by reading the DID flag after confirming that TDRE flag is 1.

Furthermore, 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 data for transmission. For details of the information that must be included in device-ID fields, contact NXP Semiconductors.

[Device-ID reception]



[When the address following a restart condition is not a device-ID address]



[When the address before the device-ID address + R had not matched the slave address]

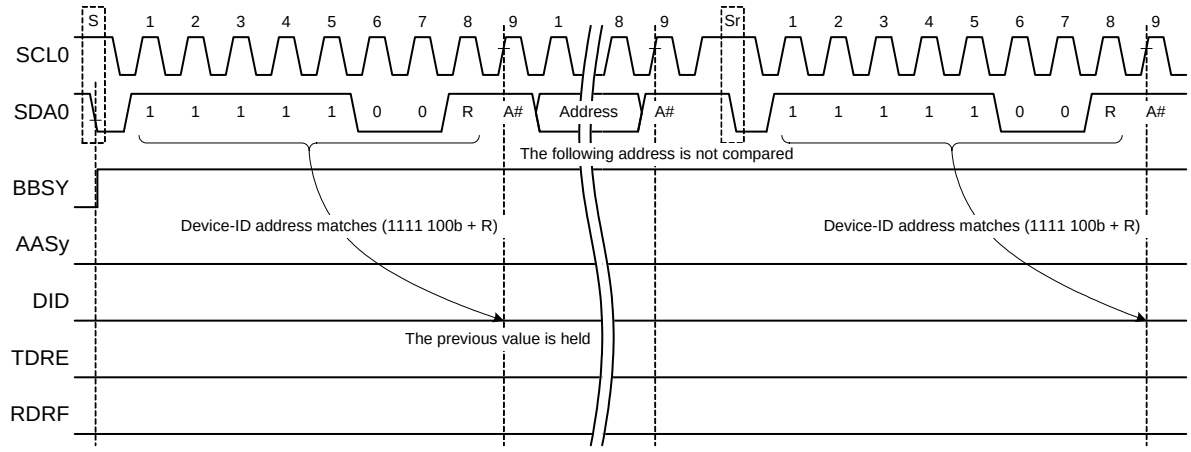


Figure 24.28 Set/Clear Timing of the AASy and DID Flags during Reception of Device-ID Address

24.7.4 Host Address Detection

The RIIC has a function to detect the host address while the SMBus is operating. When the ICSER.HOAE bit is set to 1 while the ICMR3.SMBS bit is 1, the RIIC can detect the host address (0001 000b) in slave receive mode (bits MST and TRS in the ICCR2 register are 00b).

When the RIIC detects the host address, the ICSR1.HOA flag is set to 1 at the rising edge of the ninth SCL, and at the same time, the ICSR2.RDRF flag is set to 1 when the R/W# bit is 0 (Wr bit). This causes a receive data full interrupt (RXI) to be generated. The HOA flag is used to recognize that the host address was sent from the smart battery or other devices.

If the bit following the host address (0001 000b) is an Rd bit (R/W# bit is 1), the RIIC can also detect the host address. After the host address is detected, the RIIC operates in the same manner as normal slave operation.

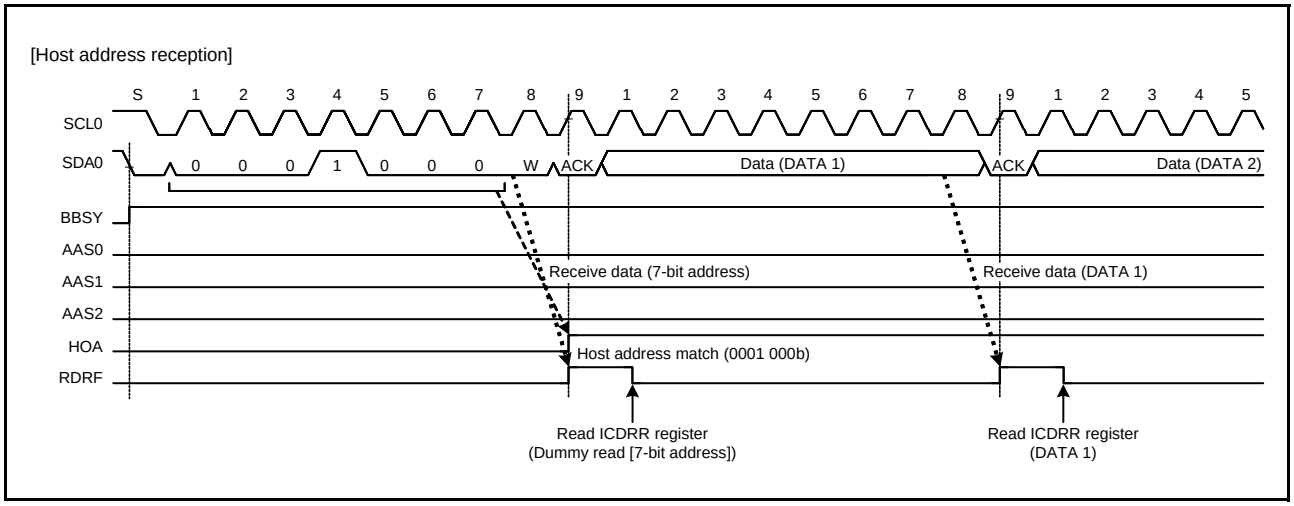


Figure 24.29 HOA Flag Set Timing during Reception of Host Address

24.8 Automatic Low-Hold Function for SCL

24.8.1 Function to Prevent Wrong Transmission of Transmit Data

If the shift register (ICDRS) is empty when data have not been written to the I2C-bus transmit data register (ICDRT) with the RIIC in transmission mode (ICCR2.TRS bit is 1), the SCL0 line is automatically held low over the intervals shown below. This low period is extended until data for transmission have been written, which prevents the unintended transmission of erroneous data.

Master transmit mode

- Low period after a start condition or restart condition is generated
- Low period between the ninth clock pulse of one transfer and the first clock pulse of the next

Slave transmit mode

- Low period between the ninth clock pulse of one transfer and the first clock pulse of the next

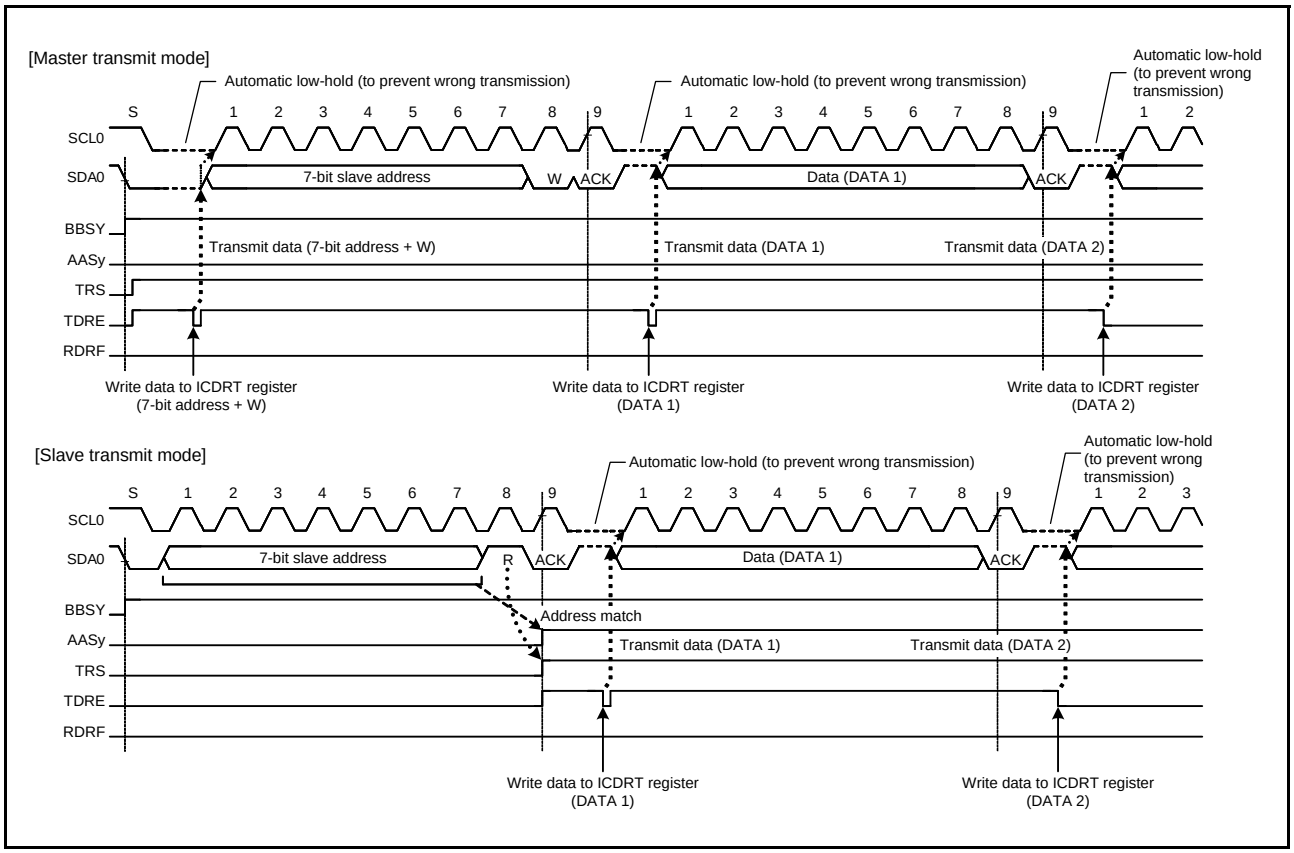


Figure 24.30 Automatic Low-Hold Operation in Transmit Mode

24.8.2 NACK Reception Transfer Suspension Function

The RIIC has a function to suspend transfer operation when NACK is received in transmit mode (ICCR2.TRS bit is 1). This function is enabled when the ICFER.NACKE bit is set to 1 (transfer suspension enabled). If the next transmit data has already been written (ICSR2.TDRE flag is 0) when NACK is received, next data transmission at the falling edge of the ninth SCL is automatically suspended. This prevents the SDA0 line output level from being held low when the MSB of the next transmit data is 0.

If the data transmission is suspended (ICSR2.NACKF flag is 1) by this function, the following data transmission and data reception are not started. To resume data transfer, set the NACKF flag to 0. In master transmit mode, restart data transfer by setting the NACKF flag to 0 after generating a restart condition, or restart data transfer from a start condition after generating a stop condition then setting the NACKF flag to 0.

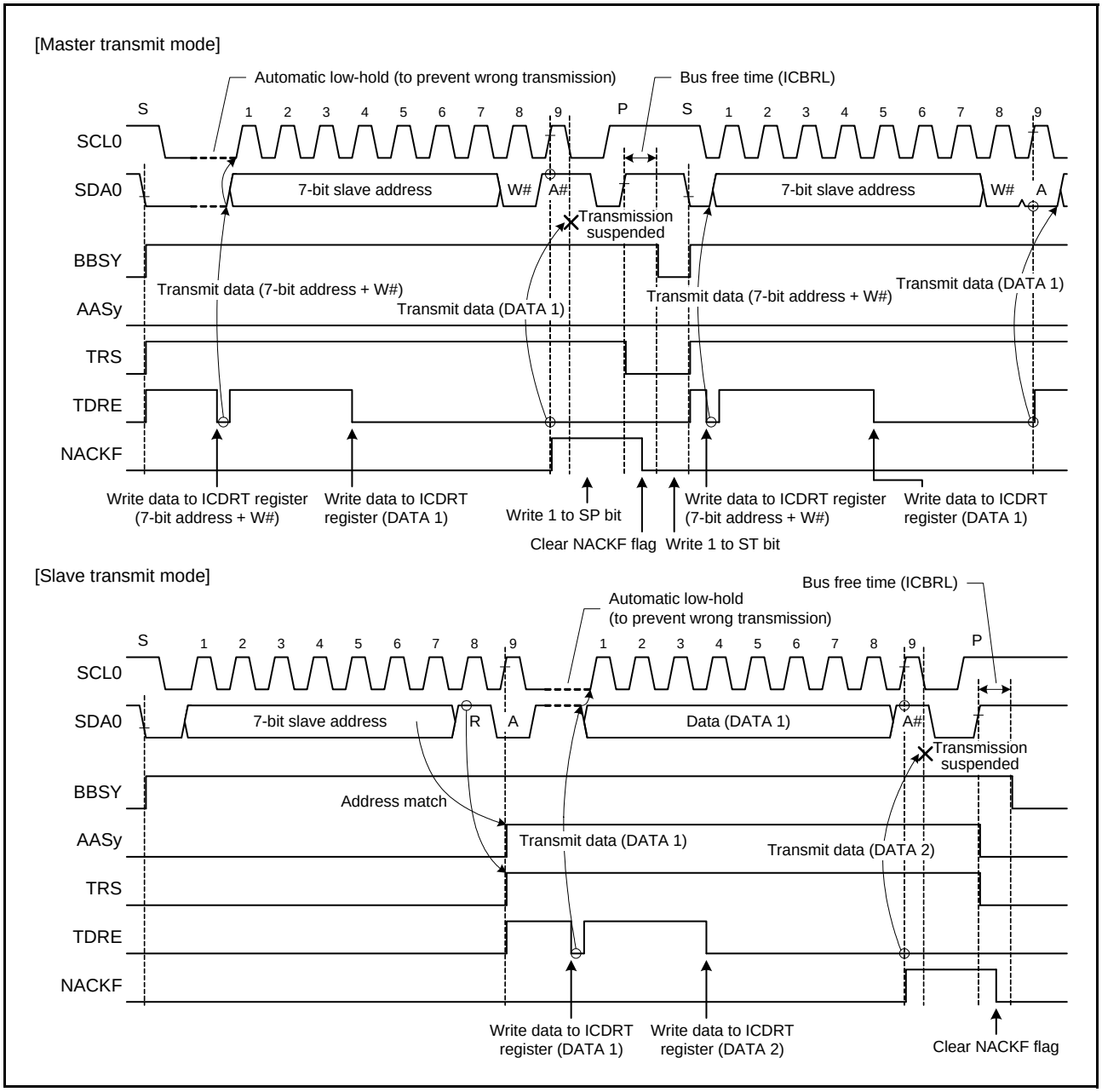


Figure 24.31 Suspension of Data Transmission When NACK is Received (NACKE = 1)

24.8.3 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 byte or more with receive data full (ICSR2.RDRF flag is 1) in receive mode (ICCR2.TRS bit is 0), the RIIC holds the SCL0 line low automatically immediately before the next data is received to prevent failure to receive data.
This function to prevent failure to receive data using the automatic low-hold function is also enabled even if the read processing of the final receive data is delayed and, in the meantime, the RIIC's own slave address is designated after a stop condition is generated. This function does not disturb other communication because the RIIC does not hold the SCL0 line low when a mismatch with its own slave address occurs after a stop condition is generated.
Sections in which the SCL0 line is held low can be selected with a combination of the WAIT and RDRFS bits in the ICMR3 register.

(1) 1-Byte Receive Operation and Automatic Low-Hold Function Using the WAIT Bit

When the ICMR3.WAIT bit is set to 1, the RIIC performs 1-byte receive operation using the WAIT bit function. Furthermore, when the ICMR3.RDRFS bit is 0, the RIIC automatically sends the ICMR3.ACKBT bit value for the acknowledgment bit in the period from the falling edge of the eighth SCL to the falling edge of the ninth SCL, and automatically holds the SCL0 line low at the falling edge of the ninth SCL using the WAIT bit function. This low-hold is released by reading data from the ICDRR register, which enables bytewise receive operation.
The WAIT bit function is enabled for receive bytes after a match with the RIIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

(2) 1-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the RDRFS Bit

When the ICMR3.RDRFS bit is set to 1, the RIIC performs 1-byte receive operation using the RDRFS bit function. When the RDRFS bit is set to 1, the ICSR2.RDRF flag (receive data full) is set to 1 at the rising edge of the eighth SCL, and the SCL0 line is automatically held low at the falling edge of the eighth SCL. This low-hold is released by writing a value to the ICMR3.ACKBT bit, but cannot be released by reading data from the ICDRR register, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units.
The RDRFS bit function is enabled for receive bytes after a match with the RIIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

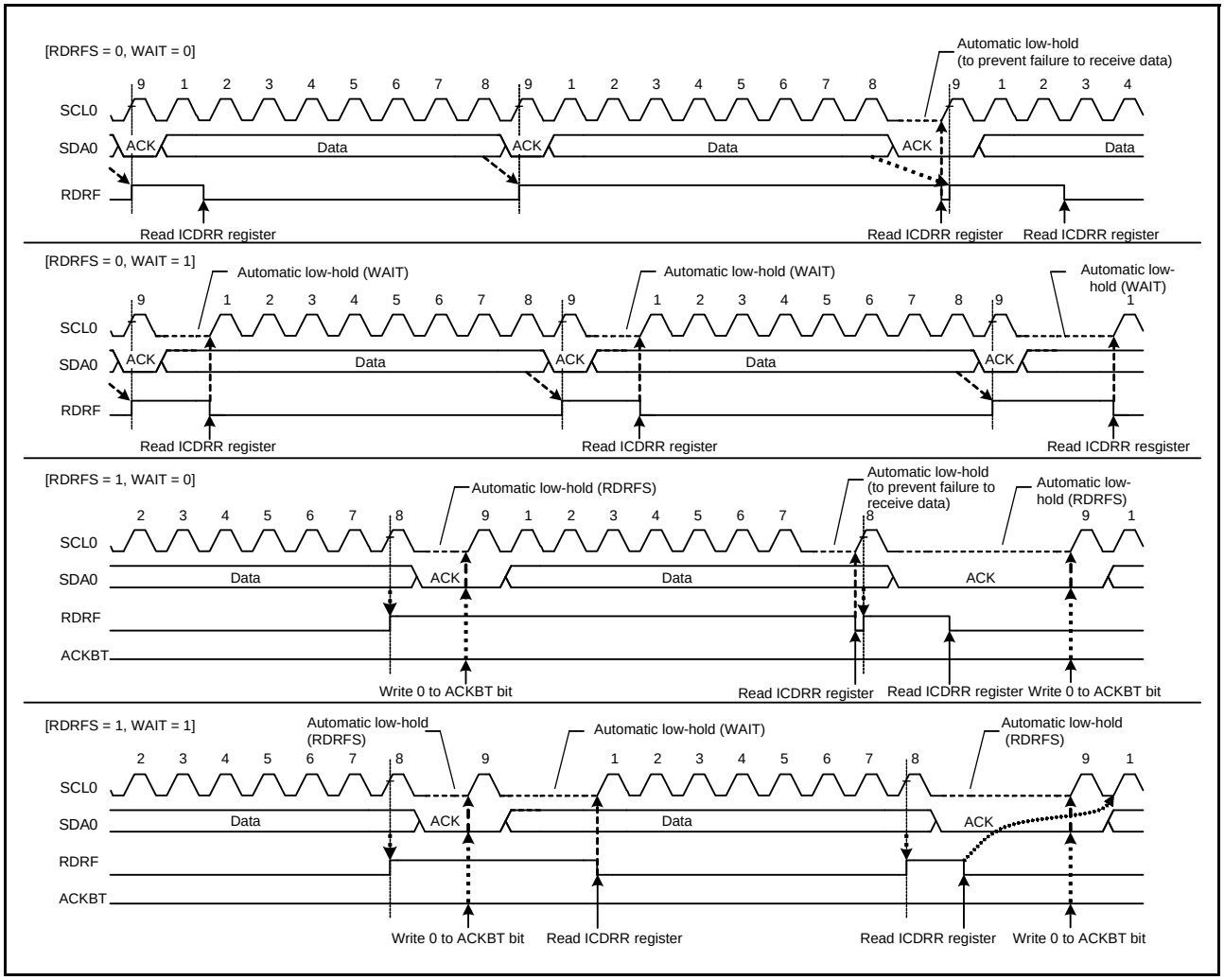


Figure 24.32 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)

24.9 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I2C-bus specification, the RIIC has functions to prevent double-generation of a start condition, to detect arbitration-lost during transmission of NACK, and to detect arbitration-lost in slave transmit mode.

24.9.1 Master Arbitration-Lost Detection (MALE Bit)

The RIIC drives the SDA0 line low to generate a start condition. However, if the SDA0 line has already been driven low by another master device generating a start condition, the RIIC causes arbitration to be lost, so priority is given to transfer by the other master device. Similarly, if the ICCR2.ST bit is set to 1 while the ICCR2.BBSY flag is 1 (bus busy state), arbitration is lost, so priority is given to transfer by the other master device. No start condition is generated in this case.
When a start condition is generated successfully, if the data for transmission including the address bits (i.e. the internal SDA output level) and the level on the SDA0 line do not match (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state) and the low is detected on the SDA0 line, the RIIC loses in arbitration.
After a loss in arbitration of mastership, the RIIC immediately enters slave receive mode. If a slave address (including the general call address) matches its own address at this time, the RIIC continues in slave operation.
A loss in arbitration of mastership is detected when the following conditions are met while the ICFER.MALE bit is 1 (master arbitration-lost detection enabled).

Conditions for master arbitration-lost

- Non-matching of the internal level for output on SDA and the level on the SDA0 line after a start condition was generated by setting the ICCR2.ST bit to 1 while the ICCR2.BBSY flag was set to 0 (erroneous generation of a start condition)
- Setting of the ICCR2.ST bit to 1 while the BBSY flag is set to 1 (start condition double-generation error)
- When the transmit data excluding acknowledgment bit (internal SDA output level) does not match the level on the SDA0 line in master transmit mode (bits MST and TRS in the ICCR2 register = 11b)

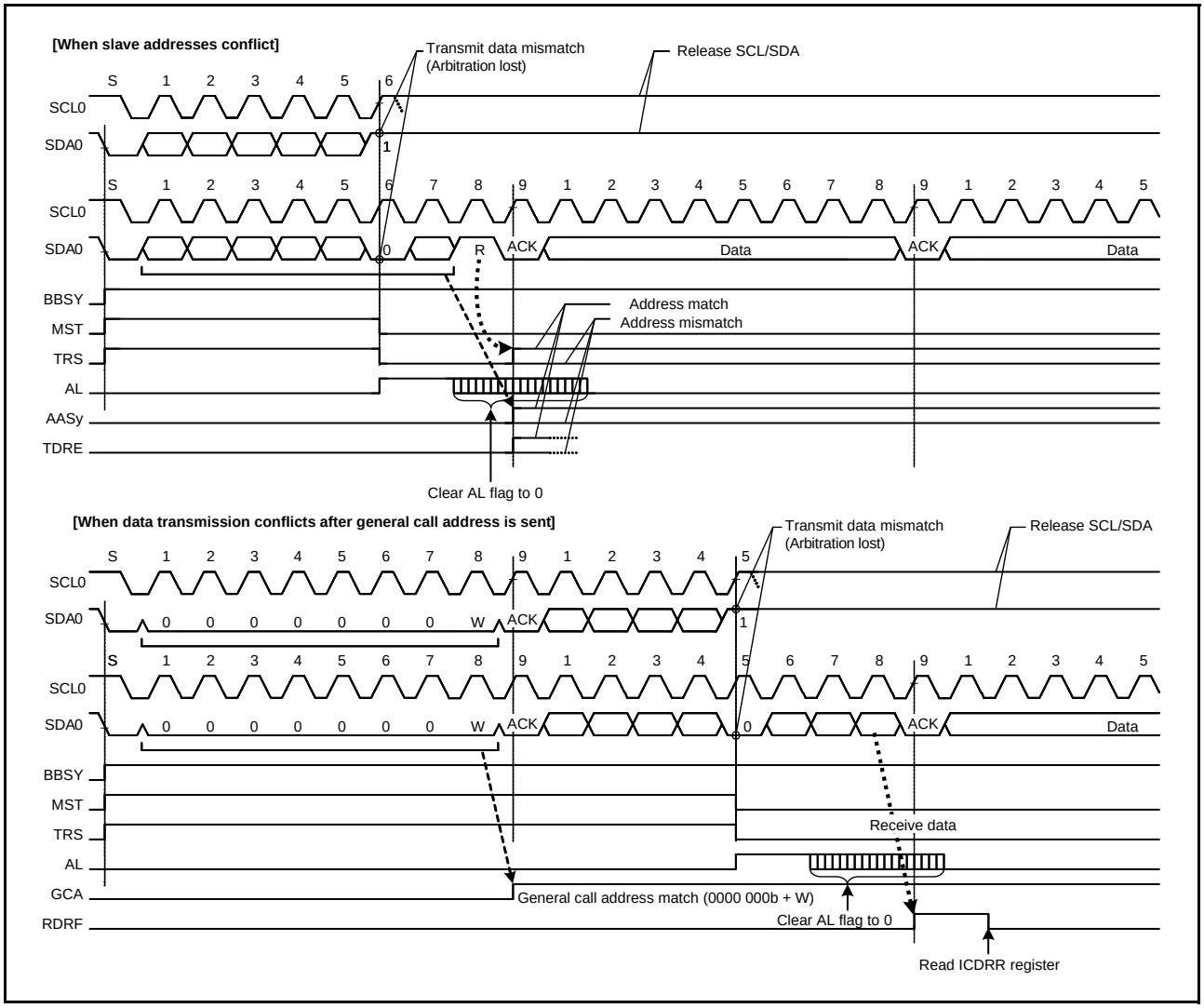


Figure 24.33 Examples of Master Arbitration-Lost Detection (MALE = 1)

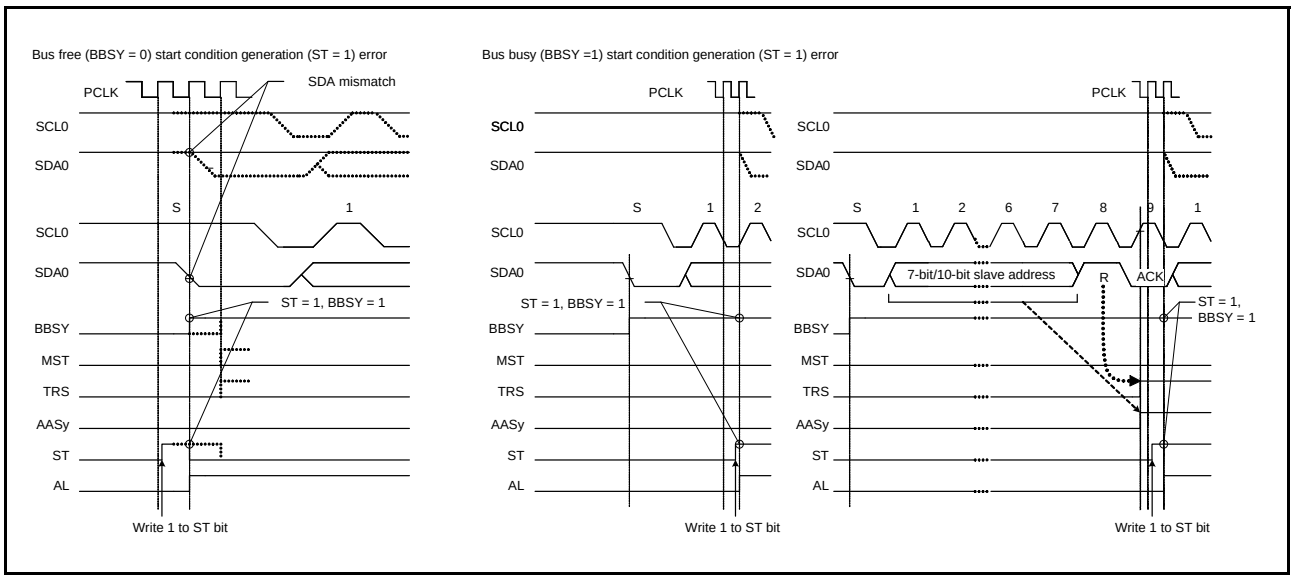


Figure 24.34 Arbitration-Lost When a Start Condition is Generated (MALE = 1)

24.9.2 Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit)

The RIIC has a function to cause arbitration to be lost if the internal SDA output level does not match the level on the SDA0 line (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state) and the low is detected on the SDA0 line during transmission of NACK in receive mode. Arbitration is lost due to a conflict of NACK transmission and ACK transmission 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/receive the same information through a single slave device. Figure 24.35 shows an example of arbitration-lost detection during transmission of NACK.

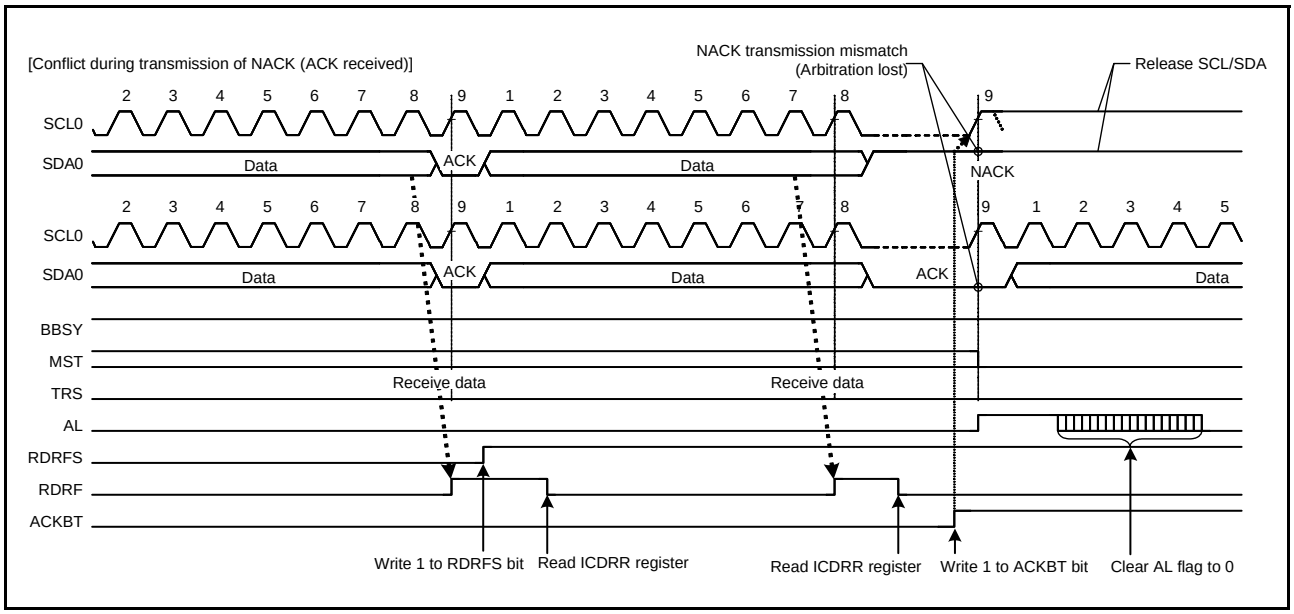


Figure 24.35 Example of Arbitration-Lost Detection during Transmission of NACK (NALE = 1)

The following explains arbitration-lost detection using an example where 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 both master A and master 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. Here, master A sends NACK when it has received 2 final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received necessary 4 bytes of data. At this time, 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 ACK transmitted by master B and generates a stop condition. Therefore, the generation of the stop condition conflicts with the SCL output of master B, which disturbs communication.

When the RIIC 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, the RIIC immediately cancels the slave match condition and enters slave receive mode. This prevents a stop condition from being generated, 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 processing (such as FFh transmission processing) necessary if the UDID (Unique Device Identifier) of assign address does not match in the Get UDID (general) processing after the Assign address command.

The RIIC detects arbitration-lost during transmission of NACK when the following condition is met with the ICFER.NALE bit 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 SDA0 line (ACK is received) during transmission of NACK (ICMR3.ACKBT bit = 1)

24.9.3 Slave Arbitration-Lost Detection (SALE Bit)

The RIIC has a function to cause arbitration to be lost if the data for transmission (i.e. the internal SDA output level) and the level on the SDA0 line do not match (the high output as the internal SDA output; i.e. the SDA0 pin is in the high-impedance state and the low is detected on the SDA0 line in slave transmit mode). This arbitration-lost detection function is mainly used when transmitting a UDID (Unique Device Identifier) over an SMBus.
When it loses slave arbitration, the RIIC 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 eliminate subsequent redundant processing (processing for the transmission of FFh).
The RIIC detects slave arbitration-lost when the following condition is met with the ICFER.SALE bit set to 1 (slave arbitration-lost detection enabled).

Condition for slave arbitration-lost

- When transmit data excluding acknowledgment bit (internal SDA output level) does not match the SDA0 line in slave transmit mode (bits MST and TRS in the ICCR2 register are 01b)

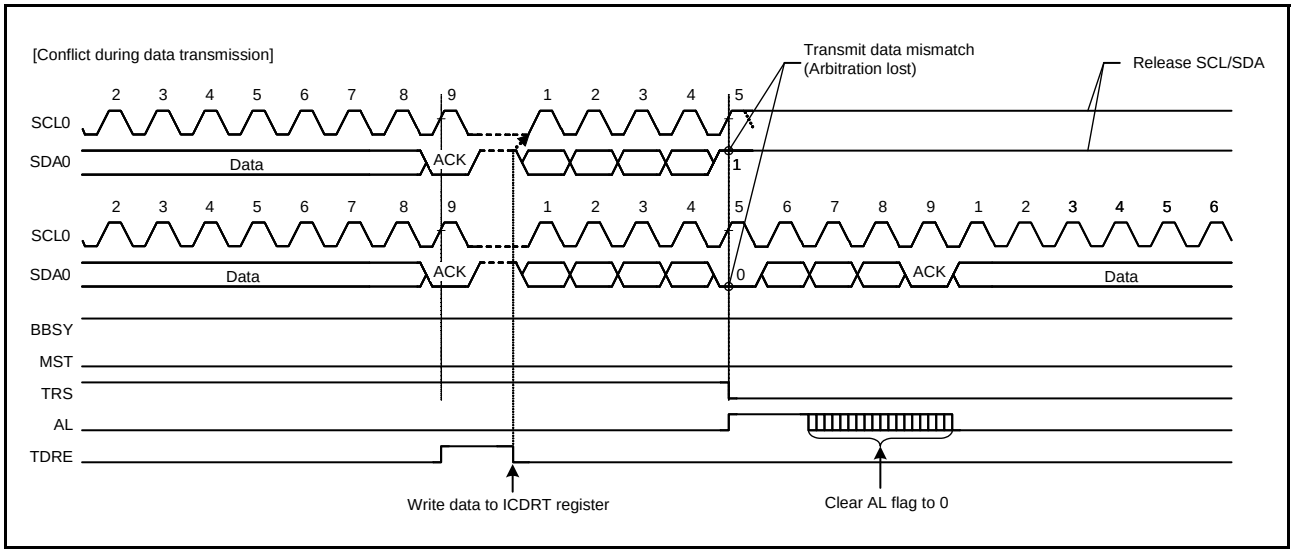


Figure 24.36 Example of Slave Arbitration-Lost Detection (SALE = 1)

24.10 Start Condition/Restart Condition/Stop Condition Generating Function

24.10.1 Generating a Start Condition

The RIIC generates a start condition when the ICCR2.ST bit is set to 1.
When the ST bit is set to 1, a start condition generation request is made and the RIIC generates a start condition when the ICCR2.BBSY flag is 0 (bus free state). When a start condition is generated normally, the RIIC automatically shifts to the master transmit mode.
A start condition is generated in the following sequence.

Start condition generation

(1) Drive the SDA0 line low (high to low).
(2) Secure the time set in the ICBRH register and the start condition hold time.
(3) Drive the SCL0 line low (high to low).
(4) Detect the low level on the SCL0 line and secure the low period of the signal on the SCL0 line set in the ICBRL register.

24.10.2 Generating a Restart Condition

The RIIC generates a restart condition when the ICCR2.RS bit is set to 1.
When the RS bit is set to 1, a restart condition generation request is made and the RIIC generates a restart condition when the ICCR2.BBSY flag is 1 (bus busy state) and the ICCR2.MST bit is 1 (master mode).
A restart condition is generated in the following sequence.

Restart condition generation

(1) Release the SDA0 line.
(2) Secure the low period of the signal on the SCL0 line set in the ICBRL register.
(3) Release the SCL0 line (low to high).
(4) Detect the high level on the SCL0 line and secure the time set in the ICBRL register and the restart condition setup time.
(5) Drive the SDA0 line low (high to low).
(6) Secure the time set in the ICBRH register and the restart condition hold time.
(7) Drive the SCL0 line low (high to low).
(8) Detect the low level on the SCL0 line and secure the low period of the signal on the SCL0 line set in the ICBRL register.

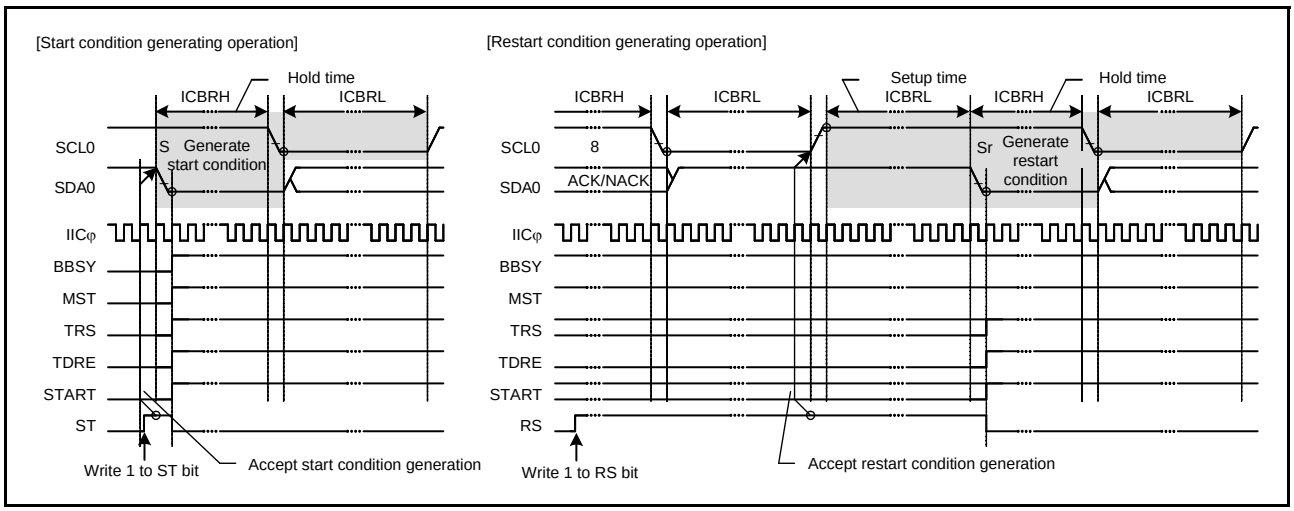


Figure 24.37 Start Condition/Restart Condition Generation Timing (ST and RS Bits)

24.10.3 Generating a Stop Condition

The RIIC generates a stop condition when the ICCR2.SP bit is set to 1.
When the SP bit is set to 1, a stop condition generation request is made and the RIIC generates a stop condition when the ICCR2.BBSY flag is 1 (bus busy state) and the ICCR2.MST bit is 1 (master mode).
A stop condition is generated in the following sequence.

Stop condition generation

(1) Drive the SDA0 line low (high to low).
(2) Secure the low period of the signal on the SCL0 line set in the ICBRL register.
(3) Release the SCL0 line (low to high).
(4) Detect the high level on the SCL0 line and secure the time set in the ICBRH register and the stop condition setup time.
(5) Release the SDA0 line (low to high).
(6) Secure the time set in the ICBRL register and the bus free time.
(7) Set the BBSY flag to 0 (to release the bus mastership).

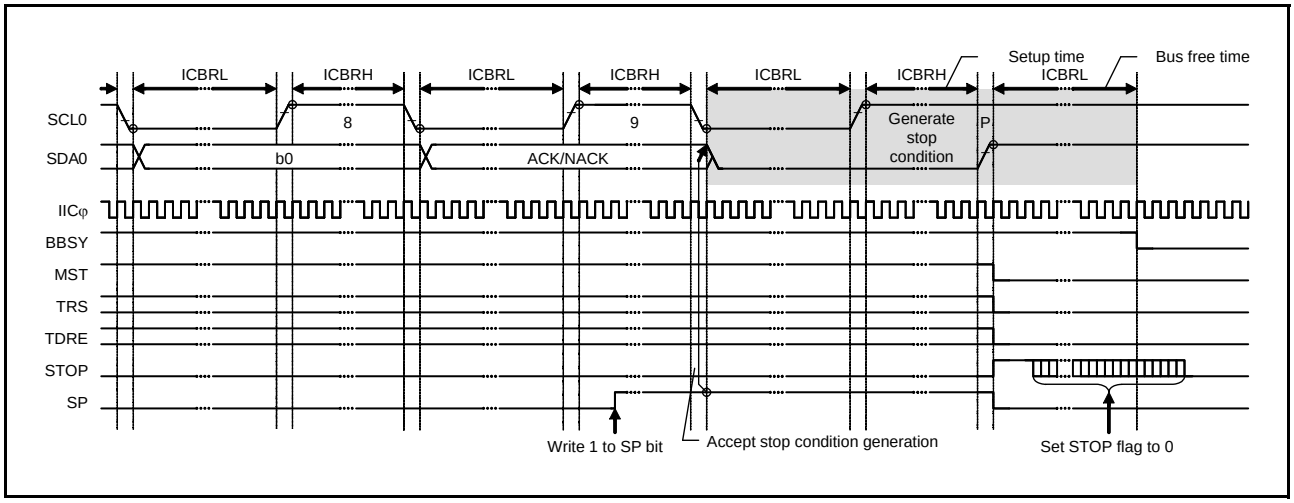


Figure 24.38 Stop Condition Generation Timing (SP Bit)

24.11 Bus Hanging

If the clock signals from the master and slave devices go out of synchronization due to noise or other factors, the I2C-bus might hang with a fixed level on the SCL0 line and/or SDA0 line.
As measures against the bus hanging, the RIIC has a timeout function to detect hanging by monitoring the SCL0 line, a function for the output of an additional SCL to release the bus from a hung state due to clock signals being out of synchronization, the RIIC reset function, and internal reset function.
By checking bits SCLO, SDAO, SCLI, and SDAI in the ICCR1 register, it is possible to see whether the RIIC or its partner in communications is placing the low level on the SCL0 or SDA0 lines.

24.11.1 Timeout Function

The RIIC includes a timeout function for detecting when the SCL0 line has been stuck longer than the predetermined time. The RIIC can detect an abnormal bus state by monitoring that the SCL0 line is stuck low or high for a predetermined time.
The timeout function monitors the SCL0 line state and counts the low period or high period using the internal counter. The timeout function resets the internal counter each time the SCL0 line changes (rising or falling), but continues to count unless the SCL0 line changes. If the internal counter overflows due to no SCL0 line change, the RIIC 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 that the SCL0 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).
- The RIIC's own slave address is detected (ICSR1 register is not 00h) 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 generation of a start condition is requested (ICCR2.ST bit is 1).

The internal counter of the timeout function works using the internal reference clock (IICφ) set by the ICMR1.CKS[2:0] bits as a count source. It functions as a 16-bit counter when long mode is selected (ICMR2.TMOS bit is 0) or a 14-bit counter when short mode is selected (TMOS bit is 1).
The SCL0 line level (low/high or both levels) during which this counter is activated can be selected by the setting of bits TMOH and TMOL in the ICMR2 register. If both bits TMOL and TMOH are set to 0, the internal counter does not work.

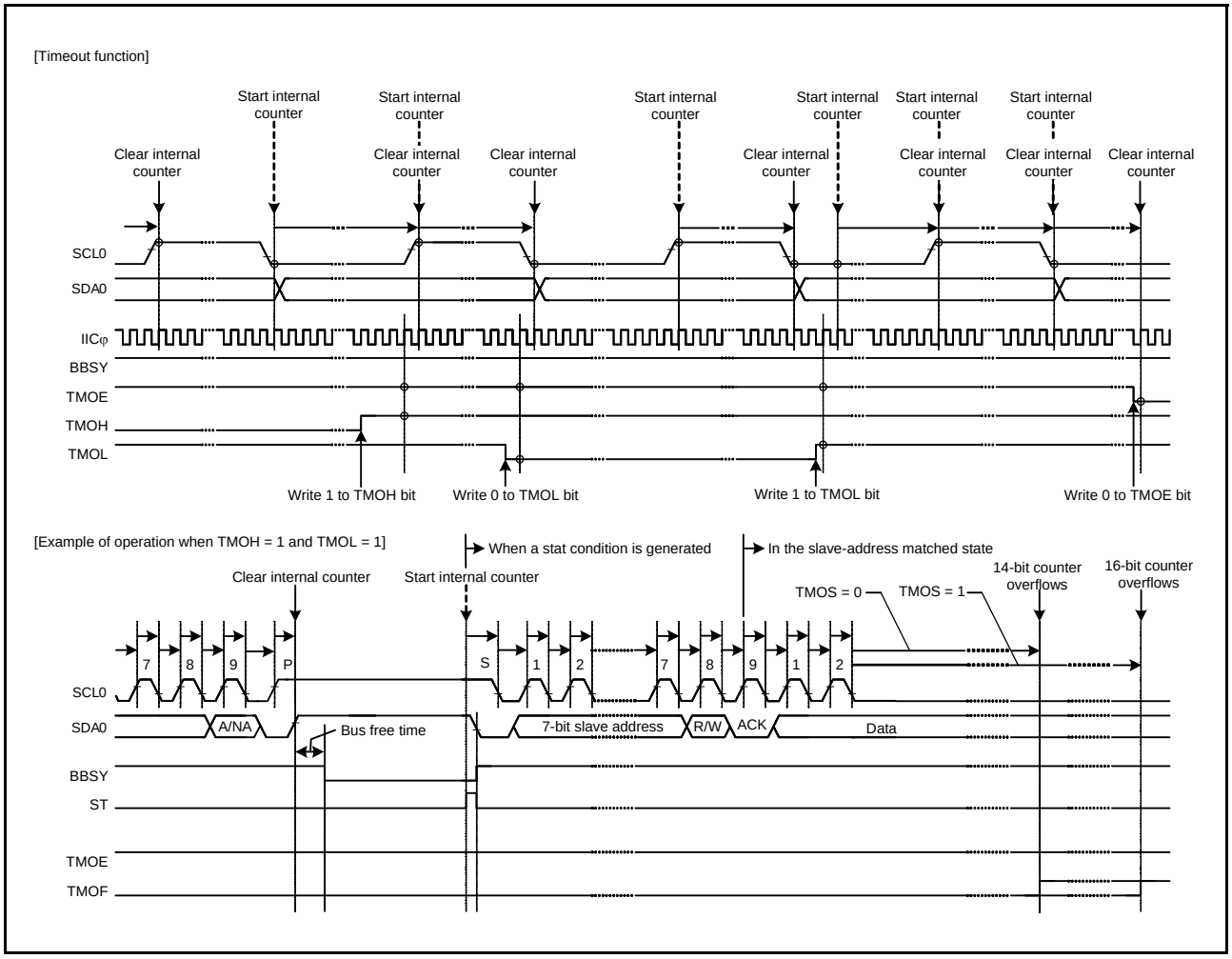


Figure 24.39 Timeout Function

24.11.2 Additional SCL Output Function

In master mode, the RIIC module has a facility for the output of additional SCL to release the SDA0 line from being held low by the slave device due to the master being out of synchronization with the slave device.
This function is mainly used in master mode to release the SDA0 line from the state of being stuck low by including additional SCL output from the RIIC with single cycles of the SCL as the unit if the RIIC cannot generate a stop condition because the slave device is holding the SDA0 line low. Do not use this function in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.
When the ICCR1.CLO bit is set to 1, an additional clock pulse at the frequency set by the ICMR1.CKS[2:0] bits and the ICBRH and ICBRL registers is output from the SCL0 pin. After output of this clock pulse, the CLO bit automatically becomes 0. The SCL0 pin is held low when the ICCR2.BBSY flag is 1 and held high when the BBSY flag is 0.
Consecutive additional clock pulses can be output by writing 1 to the CLO bit after confirming the CLO bit to be 0.
When the RIIC module is in master mode and the slave device is holding the SDA0 line low because synchronization with the slave device has been lost due to the effects of noise, etc., the output of a stop condition is not possible. The additional SCL output function can be used to output additional clock pulses one by one to make the slave device release the SDA0 line from being held low, thus recovering the bus from an unusable state. Release of the SDA0 line by the slave device can be monitored by reading the ICCR1.SDAI bit. After confirming release of the SDA0 line by the slave device, complete communications by regenerating a stop condition.
Use this function with the ICFER.MALE bit set to 0 (master arbitration-lost detection disabled).

Conditions for using the ICCR1.CLO bit

- When the bus is free (ICCR2.BBSY flag is 0) or in master mode (ICCR2.MST bit is 1 and ICCR2.BBSY flag is 1)
- When the communication device does not hold the SCL0 line low

Figure 24.40 shows the operation timing of the additional SCL output function (CLO bit).

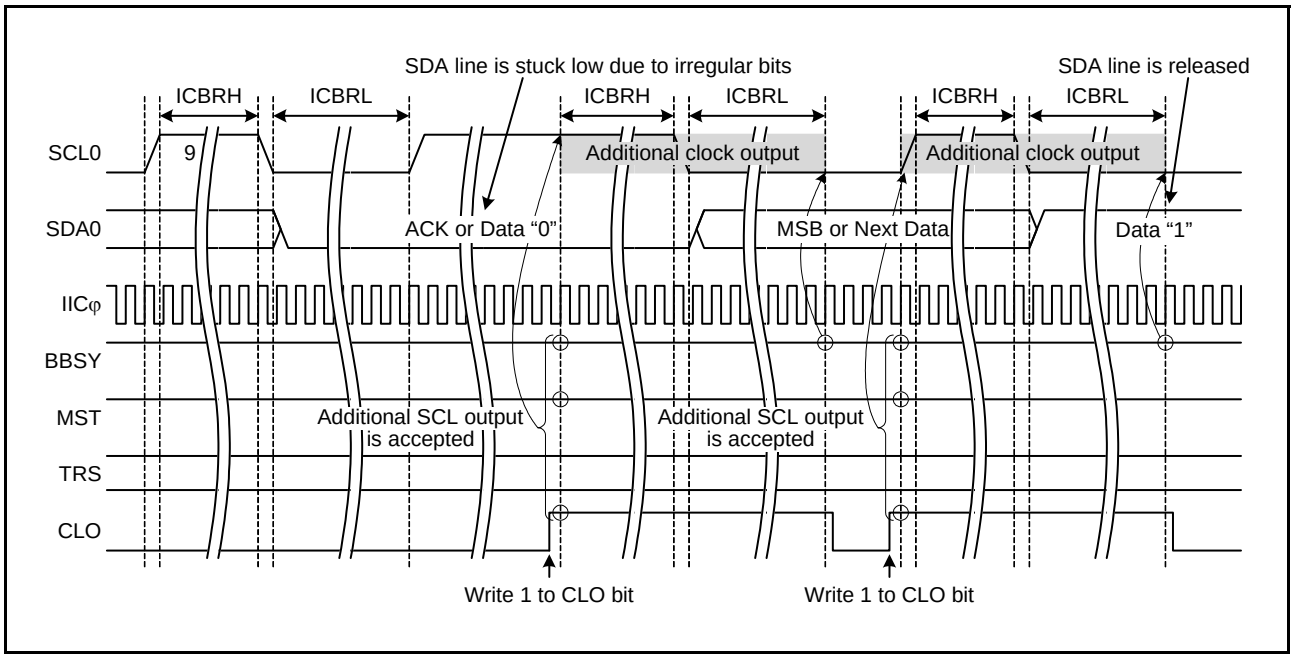


Figure 24.40 Additional SCL Output Function (CLO Bit)

24.11.3 RIIC Reset and Internal Reset

The RIIC module incorporates a function for resetting itself. There are two types of reset. One is referred to as an RIIC reset; this initializes all registers including the ICCR2.BBSY flag. The other is referred to as an internal reset; this releases the RIIC from the slave-address matched state and initializes the internal counter while retaining other settings. After applying a reset, be sure to set the ICCR1.IICRST bit to 0.

Both types of reset are effective for release from bus-hung states because both restore the output state of the SCL0 and SDA0 pins to the high-impedance state.

Applying a reset during slave operation may lead to a loss of synchronization between the master device clock and the slave device clock, so avoided this where possible. Note that monitoring of the bus state, such as for the presence of a start condition, is not possible during an RIIC reset (bits ICE and IICRST in the ICCR1 register are 01b).

For a detailed description of the RIIC and internal resets, refer to section 24.14, Initialization of Registers and Functions When a Reset is Applied or a Condition is Detected.

24.12 SMBus Operation

The RIIC is available for data communication conforming to the SMBus (Version 2.0). To perform SMBus communication, set the ICMR3.SMBS bit to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus specification, set the ICMR1.CKS[2:0] bits, the ICBRH register, and the ICBRL register. In addition, determine the values of the ICMR2.DLCS bit and the ICMR2.SDDL[2:0] bits to meet the data hold time (300 ns (min.)). If the RIIC is used only as a slave device, the transfer rate setting is not necessary, whereas the ICBRL register needs to 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 (registers SARL0, SARL1, and SARL2), and set the corresponding SARUy.FS bit (7-bit/10-bit address format select) (y = 0 to 2) to 0 (7-bit address format).
When transmitting the UDID (Unique Device Identifier), set the ICFER.SALE bit to 1 to enable the slave arbitration-lost detection function.

24.12.1 SMBus Timeout Measurement

(1) Measuring timeout of slave device

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 MTU timer using a start condition detection interrupt (STI) and stop condition detection interrupt (SPI) of the RIIC. The measured timeout period must be within the cumulative clock low extend time (slave device) ($T_{LOW:SEXT}$: 25 ms (max.)) of the SMBus specification.
If the time measured with the MTU exceeds the detect clock low timeout ($T_{TIMEOUT}$: 25 ms (min.)) of the SMBus specification, the slave device must release the bus by writing 1 to the ICCR1.IICRST bit to apply an internal reset of the RIIC. When an internal reset is applied to the RIIC, it stops driving the SCL0 and SDA0 pins of the bus and makes the SCL0/SDA0 pin outputs high-impedance, thus releasing the bus.

(2) Measuring timeout of master device

The following periods (timeout interval: $T_{LOW:MEXT}$) must be measured for master devices in SMBus communication.

- From start condition to acknowledgment bit
- Between acknowledgment bits
- From acknowledgment bit to stop condition

To measure timeout for master devices, measure these periods with the MTU timer using a start condition detection interrupt (STI), stop condition detection interrupt (SPI), and transmit end interrupt (TEI) or receive data full interrupt (RXI) of the RIIC. The measured timeout period must be within the cumulative clock low extend time (master device) ($T_{LOW:MEXT}$: 10 ms (max.)) of the SMBus specification, and the total of all $T_{LOW:MEXT}$ from start condition to stop condition must be within $T_{LOW:SEXT}$: 25 ms (max.).
For the ACK receive timing (rising edge of the ninth SCL), monitor the ICSR2.TEND flag in master transmit mode (master transmitter) and the ICSR2.RDRF flag in master receive mode (master receiver). For this reason, perform bytewise transmit operation in master transmit mode, and hold the ICMR3.RDRFS bit 0 until the byte just before reception of the final byte in master receive mode. While the RDRFS bit is 0, the RDRF flag is set to 1 at the rising edge of the ninth SCL.
If the period measured with the MTU exceeds the cumulative clock low extend time (master device) ($T_{LOW:MEXT}$: 10 ms (max.)) of the SMBus specification or the total of measured periods exceeds the detect clock low timeout ($T_{TIMEOUT}$: 25 ms (min.)) of the SMBus specification, the master device must stop the transaction by generating a stop condition. In master transmit mode, immediately stop the transmit operation (writing data to the ICDRT register).

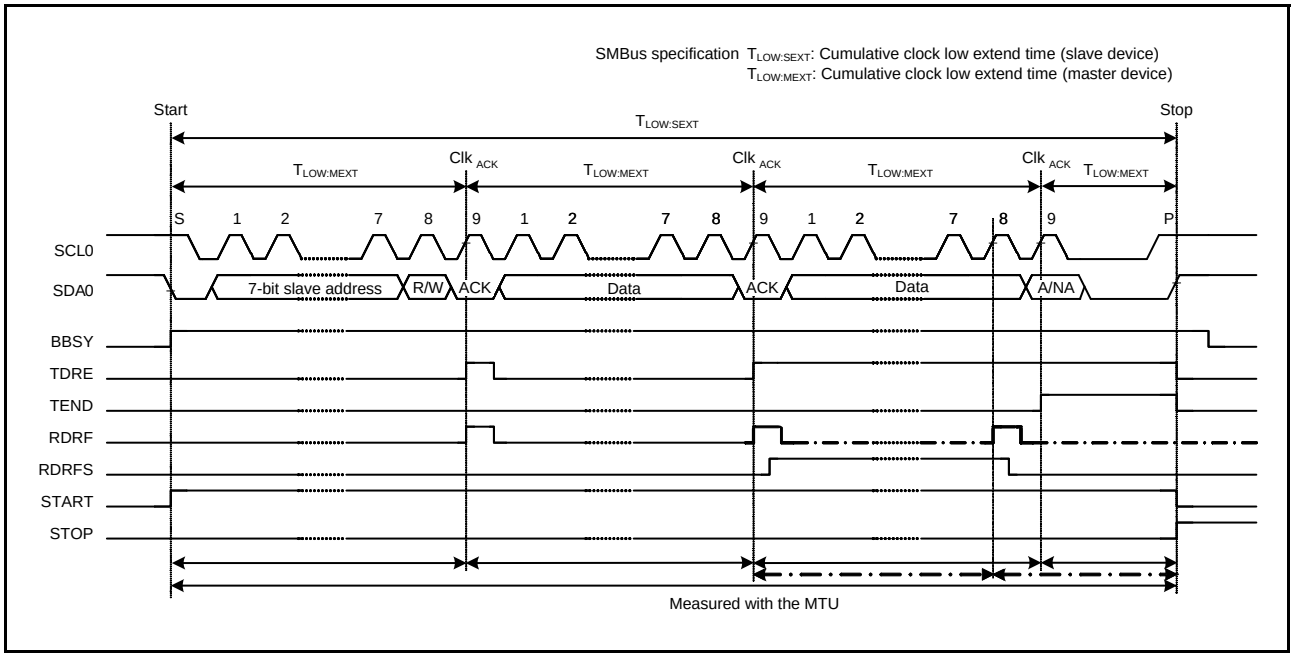


Figure 24.41 SMBus Timeout Measurement

24.12.2 Packet Error Code (PEC)

This MCU incorporates a CRC calculator. The CRC calculator enables transmission of a packet error code (PEC) or checking the received data of the SMBus in data communication of the RIIC. For the CRC generating polynomials of the CRC calculator, refer to section 25, CRC Calculator (CRC).

The PEC data in master transmit mode can be generated by writing all transmit data to the CRC data input register (CRCDIR) in the CRC calculator.

The PEC data in master receive mode 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 according to the match or mismatch result when the final byte is received as a result of the PEC code check, set the ICMR3.RDRFS bit to 1 before the rising edge of the eighth SCL during reception of the final byte, and hold the SCL0 line low at the falling edge of the eighth clock pulse.

24.12.3 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 of this MCU 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, so the RIIC has a function for detecting the host address. To detect the host address as a slave address, set the ICMR3.SMBS bit and the ICSER.HOAE bit to 1. Operation after the host address has been detected is the same as normal slave operation.

24.13 Interrupt Sources

The RIIC generates four types of interrupt request: transfer error or event generation (arbitration-lost, NACK detection, timeout detection, start condition detection, and stop condition detection), receive data full, transmit data empty, and transmit end.

Table 24.6 lists details of the several interrupt requests. The receive data full and transmit data empty are both capable of activating data transfer by the DTC.

Table 24.6 Interrupt Sources

Symbol	Interrupt Source	Interrupt Flag	DTC Activation	Interrupt Condition
EEI	Transfer error/ event generation	AL	Not possible	AL = 1 and ALIE = 1
		NACKF		NACKF = 1 and NAKIE = 1
		TMOF		TMOF = 1 and TMOIE = 1
		START		START = 1 and STIE = 1
		STOP		STOP = 1 and SPIE = 1
RXI*2	Receive data full	RDRF	Possible	RDRF = 1 and RIE = 1
TXI*1	Transmit data empty	TDRE	Possible	TDRE = 1 and TIE = 1
TEI*3	Transmit end	TEND	Not possible	TEND = 1 and TEIE = 1

Note: There is a delay time between the execution of a write instruction for a peripheral module by the CPU and actual writing to the module. Thus, when an interrupt flag has been cleared or an interrupt request has been masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt handling. Returning from interrupt handling without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.

Note 1. Because TXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICSR2.TDRE flag (a condition for TXI) is automatically set to 0 when data for transmission are written to the ICDRT register or a stop condition is detected (ICSR2.STOP flag is 1).

Note 2. Because RXI is an edge-detected interrupted, it does not require clearing. Furthermore, the ICSR2.RDRF flag (a condition for RXI) is automatically set to 0 when data are read from the ICDRR register.

Note 3. When using the TEI interrupt, clear the ICSR2.TEND flag in the TEI interrupt handling.
Note that the ICSR2.TEND flag is automatically set to 0 when data for transmission are written to the ICDRT register or a stop condition is detected (ICSR2.STOP flag is 1).

Clear the each flag or mask the interrupt request during interrupt handling.

24.13.1 Buffer Operation for TXI and RXI Interrupts

If the conditions for generating a TXI and RXI interrupt are satisfied while the corresponding IR flag is 1, the interrupt request is not output for the ICU but retained internally (the capacity for internal retention is one request per source). An interrupt request that was being retained internally is output to the ICU when the value of the ICU.IRn.IR flag becomes 0. Internally retained interrupt requests are automatically cleared under normal conditions of usage. Internally retained interrupt requests can also be cleared by writing 0 to the corresponding interrupt enable bit in the ICIER register.

24.14 Initialization of Registers and Functions When a Reset is Applied or a Condition is Detected

The RIIC can be reset by MCU reset, RIIC reset, and internal reset functions. Table 24.7 lists the reset states of registers and functions when a reset is applied or a condition is detected.

Table 24.7 Reset States of Registers and Functions When a Reset is Applied or a Condition is Detected

		MCU Reset	RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start Condition/ Restart Condition Detection	Stop Condition Detection
ICCR1	SDAO, SCLO	To be reset	To be reset	To be reset	Retained	Retained
	IICRST, ICE		Retained	Retained		
	Others		To be reset			
ICCR2	ST, RS	To be reset	To be reset	To be reset	To be reset	Retained
	SP					To be reset
	TRS				See note 1	
	MST				See note 1	
	BBSY			Retained	Becomes 1	
ICMR1	BC[2:0]	To be reset	To be reset	To be reset	To be reset	Retained
	Others			Retained	Retained	
ICMR2		To be reset	To be reset	Retained	Retained	Retained
ICMR3	ACKBT	To be reset	To be reset	Retained	Retained	To be reset
	Others					Retained
ICFER		To be reset	To be reset	Retained	Retained	Retained
ICSER		To be reset	To be reset	Retained	Retained	Retained
ICIER		To be reset	To be reset	Retained	Retained	Retained
ICSR1		To be reset	To be reset	To be reset	Retained	To be reset
ICSR2	START	To be reset	To be reset	To be reset	Becomes 1	To be reset
	STOP				Retained	Becomes 1
	TEND					To be reset
	TDRE				See note 1	
	Others				Retained	Retained
SARL0, SARL1, SARL2, SARU0, SARU1, SARU2		To be reset	To be reset	Retained	Retained	Retained
ICBRH, ICBRL		To be reset	To be reset	Retained	Retained	Retained
ICDRT		To be reset	To be reset	Retained	Retained	Retained
ICDRR		To be reset	To be reset	Retained	Retained	Retained
ICDRS		To be reset	To be reset	To be reset	Retained	Retained
Timeout function		To be reset	To be reset	To be reset	Operation	Operation
Bus free time measurement		To be reset	To be reset	Operation	Operation	Operation

Note 1. This bit is not reset. This bit becomes 0 or 1 in accordance with the conditions.

24.15 Usage Notes

24.15.1 Setting Module Stop Function

Module stop state can be entered or released using module stop control register B (MSTPCRB). The initial setting is for operation of the RIIC to be stopped. RIIC register access is enabled by releasing the module stop state.
For details on module stop control register B, refer to section 11, Low Power Consumption.

24.15.2 Notes on Starting Transfer

If the IR flag corresponding to the RIIC interrupt is 1 when transfer is started (ICCR1.ICE bit is 1), follow the procedure below to clear interrupts before enabling operations. Starting transfer with the IR flag set to 1 while the ICCR1.ICE bit is 1 leads to an interrupt request being internally retained after transfer starts, and this can lead to unanticipated behavior of the IR flag.

1. Confirm that the ICCR1.ICE bit is 0.
2. Set the relevant interrupt enable bits (ICIER.TIE, etc.) on the peripheral function side to 0.
3. Read the relevant interrupt enable bits (ICIER.TIE, etc.) on the peripheral function side and confirm that its value is 0.
4. Set the IR flag to 0.

25. CRC Calculator (CRC)

The CRC (Cyclic Redundancy Check) calculator generates CRC codes.

25.1 Overview

Table 25.1 lists the specifications of the CRC calculator, and Figure 25.1 shows a block diagram of the CRC calculator.

Table 25.1 CRC Specifications

Item	Description
Data for CRC calculation*1	CRC codes are generated for any desired data in 8n-bit units (where n is a whole number)
CRC processor unit	8-bit parallel processing
CRC generating polynomial	One of three generating polynomials is selectable • 8-bit CRC $X^8 + X^2 + X + 1$ • 16-bit CRC $X^{16} + X^{15} + X^2 + 1$ $X^{16} + X^{12} + X^5 + 1$
CRC calculation switching	The bit order of CRC calculation results can be switched for LSB first or MSB first communication
Low power consumption function	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 units.

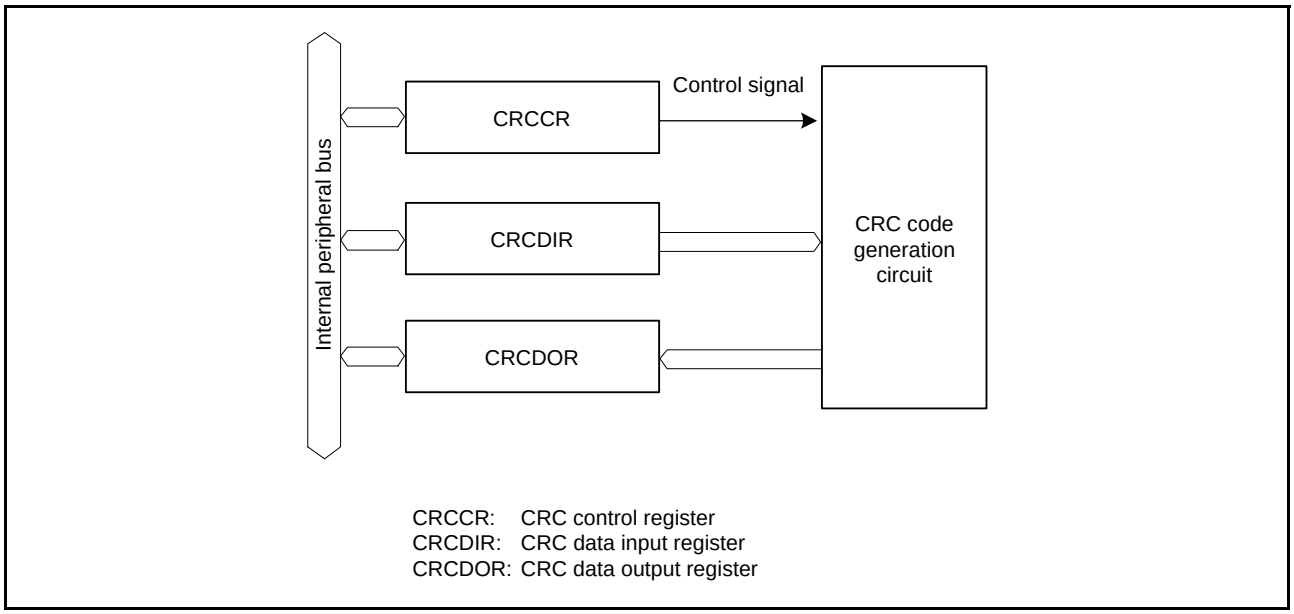


Figure 25.1 CRC Block Diagram

25.2 Register Descriptions

25.2.1 CRC Control Register (CRCCR)

Address(es): 0008 8280h

	b7	b6	b5	b4	b3	b2	b1	b0
	DORCLR	—	—	—	—	LMS	GPS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	GPS[1:0]	CRC Generating Polynomial Switching	b1 b0 0 0: No calculation is executed. 0 1: 8-bit CRC ($X^8 + X^2 + X + 1$) 1 0: 16-bit CRC ($X^{16} + X^{15} + X^2 + 1$) 1 1: 16-bit CRC ($X^{16} + X^{12} + X^5 + 1$)	R/W
b2	LMS	CRC Calculation Switching	0: Generates CRC for LSB first communication. 1: Generates CRC for MSB first communication.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DORCLR	CRCDOR Register Clear	1: Clears the CRCDOR register. This bit is read as 0.	R/W*1

Note 1. Only 1 can be written.

LMS Bit (CRC Calculation Switching)

This bit selects the bit order of generated 16-bit CRC code. Transmit the lower-order byte (b7 to b0) of the CRC code first for LSB first communication and the higher-order byte (b15 to b8) first for MSB first communication. For details on the transmission and reception of CRC codes, refer to section 25.3, Operation.

DORCLR Bit (CRCDOR Register Clear)

Write 1 to this bit so that the CRCDOR register is set to 0000h.
This bit is read as 0. Only 1 can be written.

25.2.2 CRC Data Input Register (CRCDIR)

Address(es): 0008 8281h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0

CRCDIR is a readable and writable register. Write data for CRC calculation to this register.

25.2.3 CRC Data Output Register (CRCDOR)

Address(es): 0008 8282h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CRCDOR is a readable and writable register.
Since its initial value is 0000h, rewrite the CRCDOR register to perform calculation using a value other than the initial value.
Data written to the CRCDIR register is CRC calculated and the result is stored in the CRCDOR register. If the CRC code is calculated following the transferred data and the result is 0000h, there is no CRC error.
When an 8-bit CRC ($X^8 + X^2 + X + 1$ polynomial) is in use, the valid CRC code is obtained in the low-order byte (b7 to b0). The high-order byte (b15 to b8) is not updated.

25.3 Operation

The CRC calculator generates CRC codes for use in LSB first or MSB first transfer.
The following shows examples of generating the CRC code for input data (F0h) using the 16-bit CRC generating polynomial ($X^{16} + X^{12} + X^5 + 1$). In these examples, the value of the CRC data output register (CRCDOR) is cleared before CRC calculation.
When an 8-bit CRC (with the polynomial $X^8 + X^2 + X + 1$) is in use, the valid bits of the CRC code are obtained in the lower-order byte of the CRCDOR register.

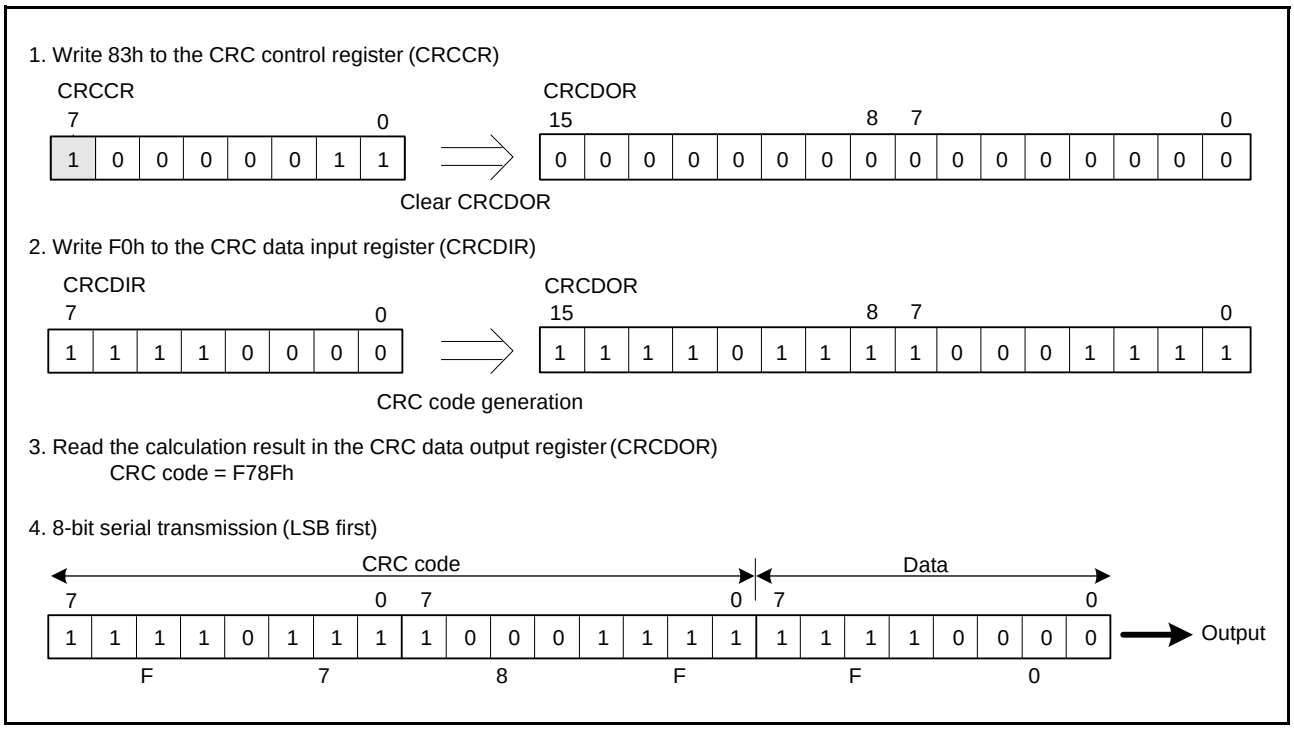


Figure 25.2 LSB First Data Transmission

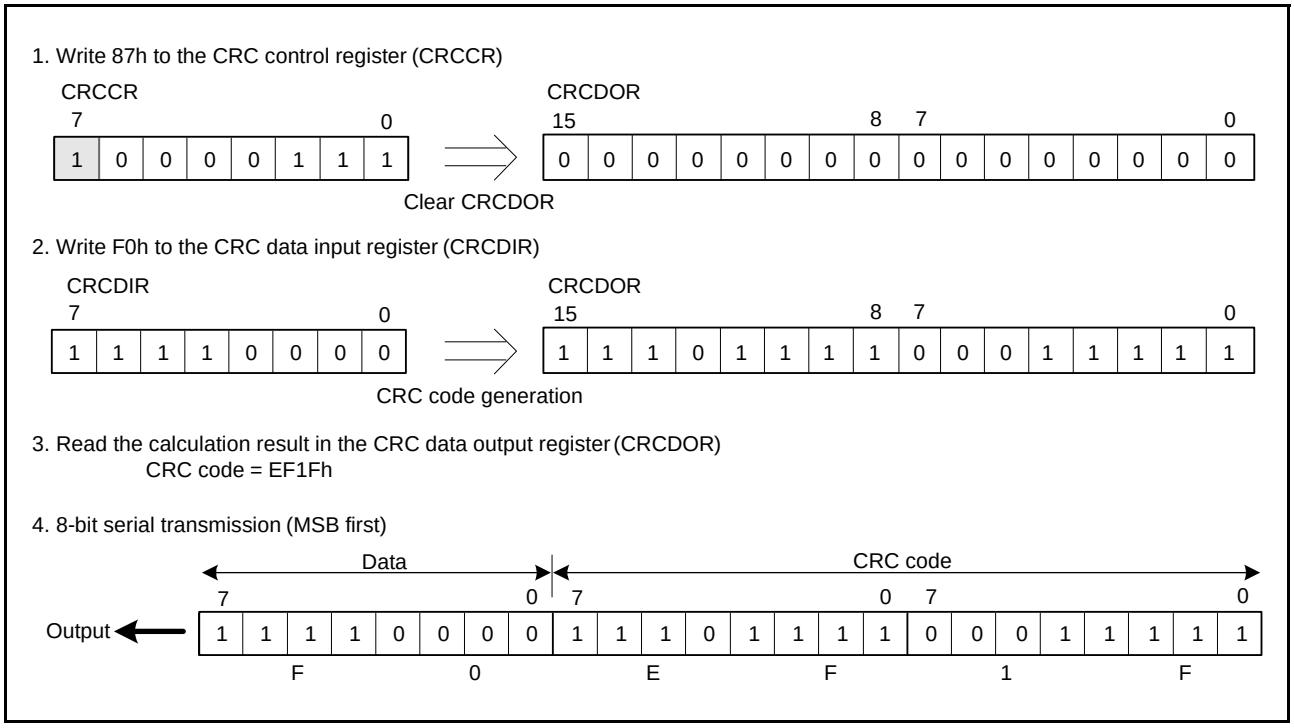


Figure 25.3 MSB First Data Transmission

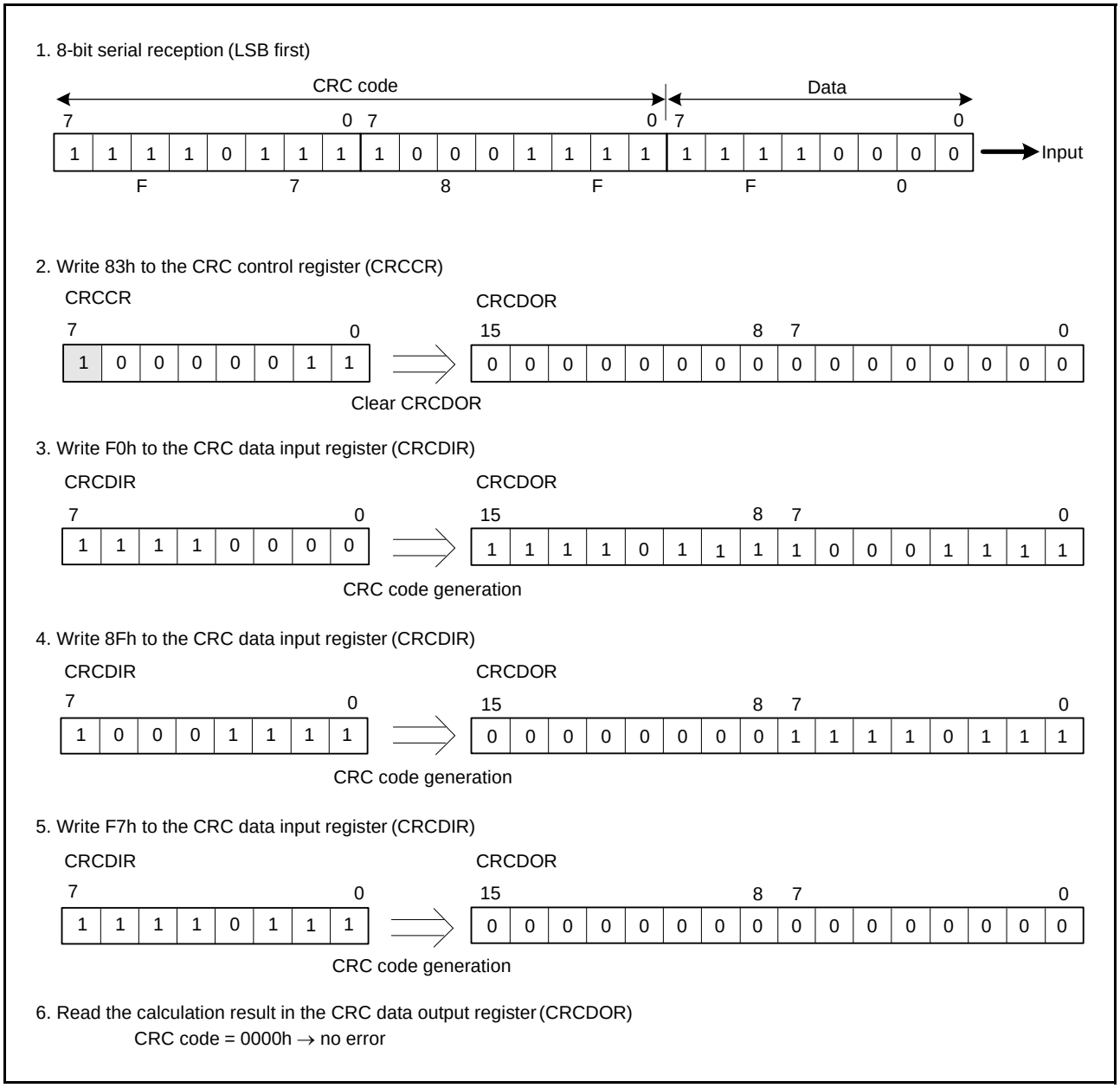


Figure 25.4 LSB First Data Reception

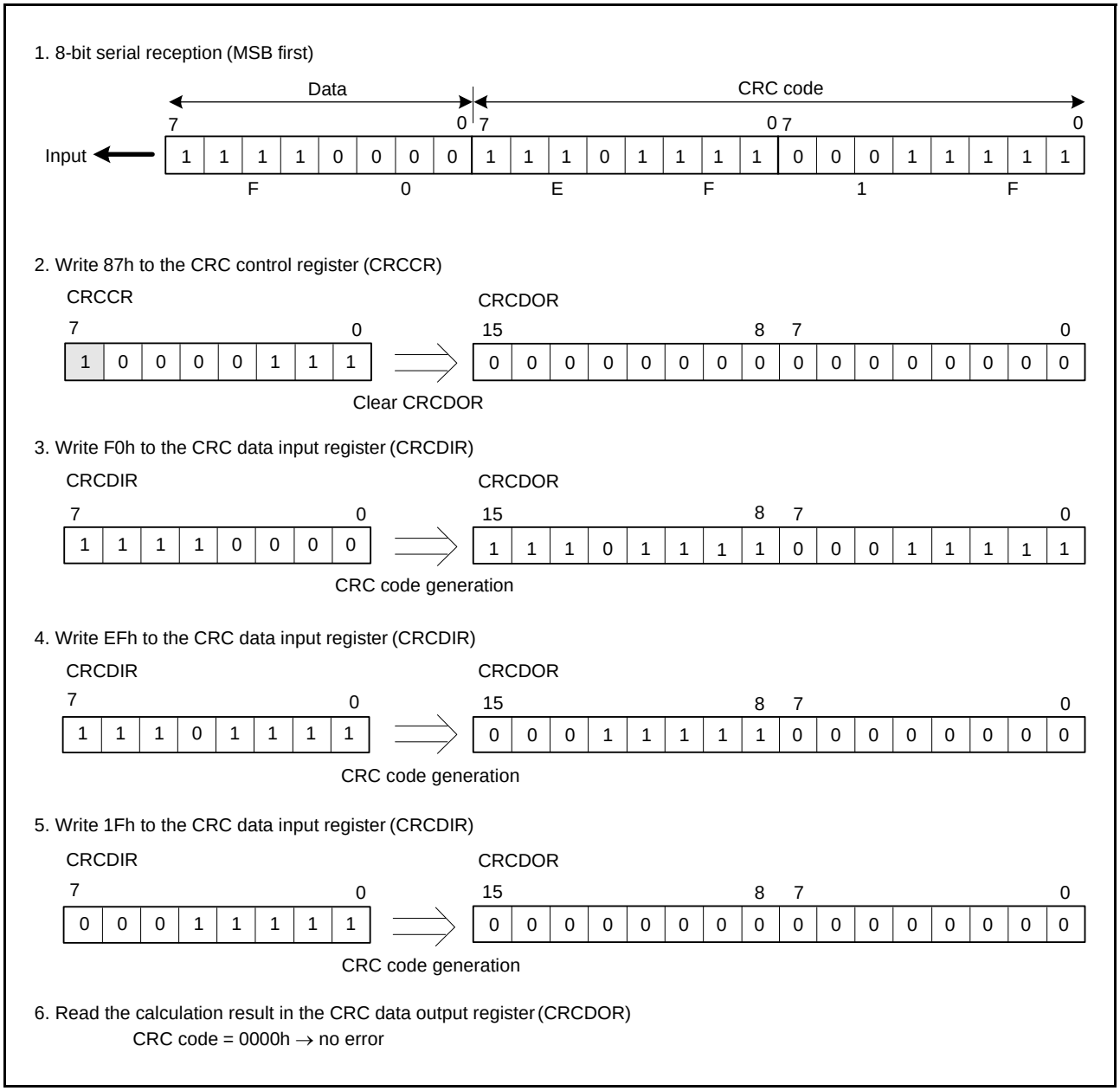


Figure 25.5 MSB First Data Reception

25.4 Usage Notes

25.4.1 Module Stop Function Setting

Operation of the CRC calculator can be disabled or enabled using the module stop control register B (MSTPCRB). After a reset, the CRC is in the module stop state. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

25.4.2 Note on Transmission

Note that the sequence of transmission for the CRC code differs according to whether transmission is LSB first or MSB first.

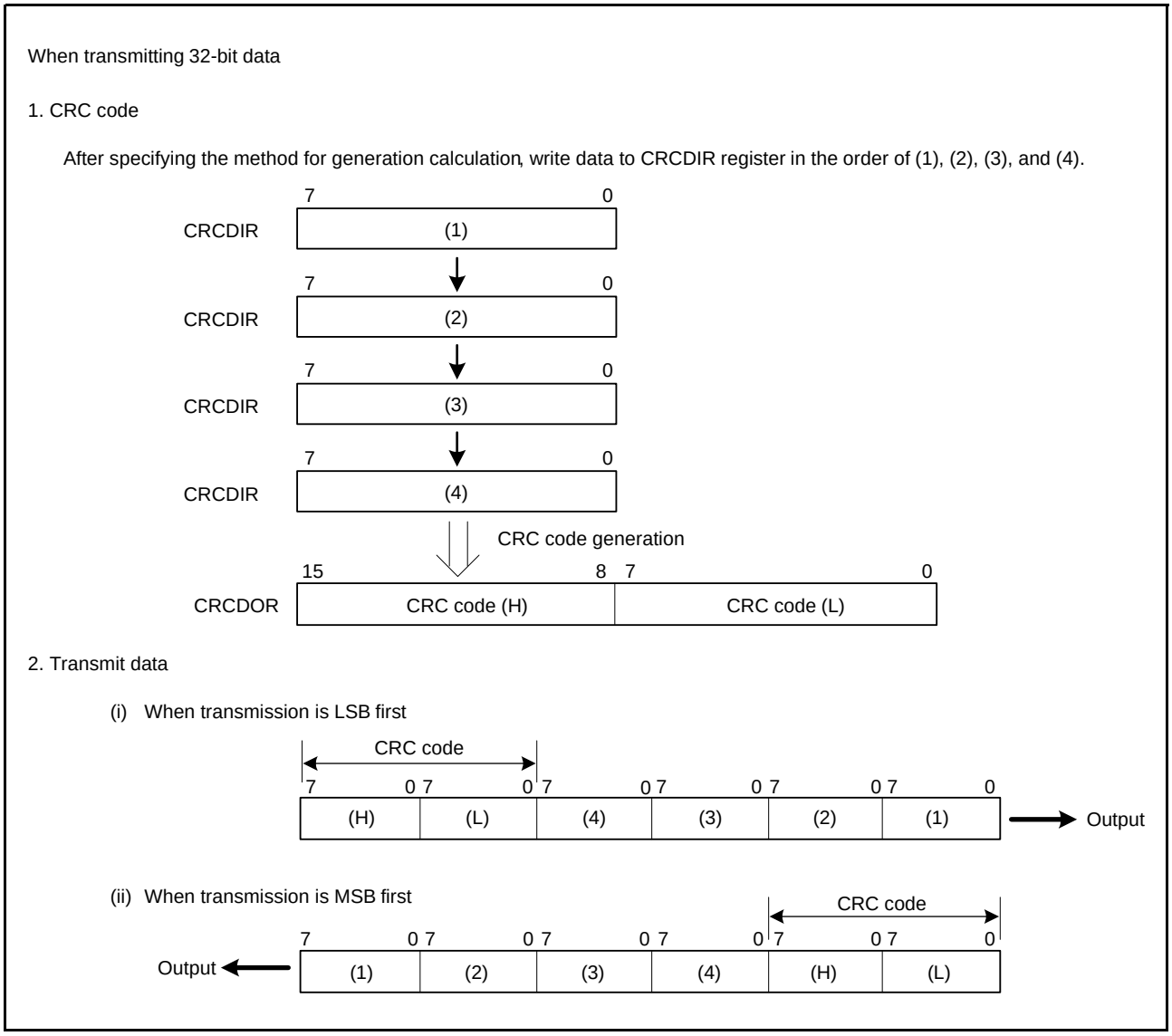


Figure 25.6 LSB First and MSB First Data Transmission

26. 12-Bit A/D Converter (S12ADF)

In this section, “PCLK” is used to refer to PCLKB.

26.1 Overview

This MCU incorporates one unit of a 12-bit successive approximation A/D converter. The one A/D converter unit (unit 0) provides eight channels for use, and the analog inputs and internal reference voltage for the converter are selectable.
The 12-bit A/D converter converts a maximum of eight selected channels of analog inputs and internal reference voltage into a 12-bit digital value through successive approximation.
The A/D converter has three operating modes: single scan mode in which the analog inputs of eight channels are converted in ascending channel order; continuous scan mode in which the analog inputs of eight channels are continuously converted in ascending channel order; and group scan mode in which eight channels are arbitrarily divided into two groups (groups A and B) or three groups (groups A, B, and C) and converted in ascending channel order in each group.
In group scan mode, either two groups (groups A and B) or three groups (groups A, B, and C) is selected.
The conditions for scanning start of each group (A and B or A, B, and C) (synchronous trigger) can be independently selected, thus allowing A/D conversion of each group to be started independently.
During group priority operation, in addition to the above-mentioned operation, a trigger to start scanning for the priority group is accepted during scan for the low-priority group, and scan for the priority group is started after scan for the low-priority group was discontinued. The priority order is group A > group B > group C. Accordingly, as priority operation, when a trigger to start scanning for group B is accepted during scan for group C, group C scan is discontinued, and scan for group B is started. Likewise, when a trigger to start scanning for group A is accepted during scan for group C, group C scan is discontinued and scan for group A is started. In the same way, when a trigger to start scanning for group A is accepted during scan for group B, group B scan is discontinued and scan for group A is started.
The discontinued scan operation can be restarted after the scanning of the priority group is completed.
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).
Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages internally generated in the 12-bit A/D converter is converted.
Table 26.1 lists the specifications of the 12-bit A/D converter and Table 26.2 lists the functions of the 12-bit A/D converter. Figure 26.1 shows block diagrams of the 12-bit A/D converter.

Table 26.1 Specifications of 12-Bit A/D Converter (1/2)

Item	Description
Number of units	One unit (S12AD)
Input channels	Eight channels for S12AD
Extended analog function	Internal reference voltage
A/D conversion method	Successive approximation method
Resolution	12 bits
Conversion time	1.4 µs per channel (when A/D conversion clock ADCLK = 32 MHz)
A/D conversion clock	Peripheral module clock PCLK*1 and A/D conversion clock ADCLK*1 can be set so that the frequency ratio should be one of the following. PCLK to ADCLK frequency ratio = 1:1, 2:1, 4:1, 8:1 ADCLK is set using the clock generation circuit.
Data registers	• Eight registers for analog input, 1 for A/D-converted data duplication in double trigger mode, and 2 for A/D-converted data duplication during extended operation in double trigger mode. • One register for internal reference • One register for self-diagnosis • The results of A/D conversion are stored in 12-bit A/D data registers. • 12-bit accuracy output for the results of A/D conversion • The value obtained by adding up A/D-converted results is stored as a value in the number of bit for conversion accuracy + 2 bits/4 bits*2 in the A/D data registers in A/D-converted value addition mode. • Double trigger mode (selectable in single scan and group scan modes): The first piece of A/D-converted analog-input data on one selected channel is stored in the data register for the channel, and the second piece is stored in the duplication register. • Extended operation in double trigger mode (available for specific triggers): A/D-converted analog-input data on one selected channel is stored in the duplication register that is prepared for each type of trigger.
Operating modes	• Single scan mode: A/D conversion is performed only once on the analog inputs arbitrarily selected. A/D conversion is performed only once on the internal reference voltage. • Continuous scan mode: A/D conversion is performed repeatedly on the analog inputs arbitrarily selected. • 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 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 priority control selected): 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 trigger Trigger by the multi-function timer pulse unit (MTU) • Asynchronous trigger A/D conversion can be triggered by the external trigger ADTRG0# pin.
Functions	• Channel-dedicated sample-and-hold function (three channels) • Input signal amplification function of the programmable gain amplifier (3 channels) • Variable sampling state count (settable for each channel) • Self-diagnosis of 12-bit A/D converter • Selectable A/D-converted value addition mode or average mode • Analog input disconnection detection assist function (discharge function/precharge function) • Double trigger mode (duplication of A/D conversion data) • Automatic clear function of A/D data registers

Table 26.1 Specifications of 12-Bit A/D Converter (2/2)

Item	Description
Interrupt sources	• In the modes except double trigger mode and group scan mode, a scan end interrupt request (S12ADI) can be generated on completion of single scan. • In double trigger mode, a scan end interrupt request (S12ADI) can be generated on completion of double scan. • In group scan mode, a scan end interrupt request (S12ADI) can be generated on completion of group A scan, whereas a scan end interrupt request (GBADI) for group B can be generated on completion of group B scan, and a group C scan end interrupt request (GCADI) can be generated on completion of group C scan. • When double trigger mode is selected in group scan mode, an A/D scan end interrupt request (S12ADI) can be generated on completion of double scan of group A, and the corresponding scan end interrupt request (GBADI/GCADI) can be generated on completion of group B and group C scan. • The S12ADI, GBADI, and GCADI interrupts can activate the data transfer controller (DTC).
Low power consumption function	• Module stop state can be set.*3, *4

Note 1. The frequency of PCLK, the peripheral module clock, becomes that set in the SCKCR.PCKB[3:0] bits, and that of ADCLK, the A/D conversion clock, becomes the frequency set in the SCKCR.PCKD[3:0] bits.
Note 2. The number of extended bits during addition differs depending on the addition count.
2-bit extension: 1-time to 4-time conversion (add zero to three times)
4-bit extension: 16-time conversion (add 15 times)
Note 3. See section 11, Low Power Consumption for details.
Note 4. Wait for 1 µs or longer to start A/D conversion after release from the module stop state.

Table 26.2 Functions of 12-Bit A/D Converter

Item			Pin Name, Abbreviation
Analog input channels			AN000 to AN007, internal reference voltage
Conditions for A/D conversion start	Software	Software trigger	Enabled
	Asynchronous trigger	Trigger input pin	ADTRG0#
	Synchronous trigger	Compare match/input capture from MTU0.TGRA	TRGA0N
		Compare match/input capture from MTU1.TGRA	TRGA1N
		Compare match/input capture from MTU2.TGRA	TRGA2N
		Compare match/input capture from MTU3.TGRA	TRGA3N
		Compare match/input capture from MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	TRGA4N
		Compare match from MTU0.TGRE	TRG0N
		Compare match between MTU4.TADCORA and MTU4.TCNT	TRG4AN
		Compare match between MTU4.TADCORB and MTU4.TCNT	TRG4BN
		Compare match between MTU4.TADCORA and MTU4.TCNT, or compare match between MTU4.TADCORB and MTU4.TCNT	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
Channel-dedicated sample-and-hold function		Target channels	AN000 to AN002
PGA		Target channels	AN000 to AN002
Interrupts			S12ADI, GBADI, GCADI interrupt
Setting of module stop function*1, *2			MSTPCRA.MSTPA17 bit

Note: When setting an A/D conversion start trigger to ADTRG0#, set the pin mode control bit in the port mode register for the corresponding pin to 1 (peripheral functions), and set the pin function select bit in the pin function control register to ADTRG0#. See section 17, I/O Ports for details.
Note 1. See section 11, Low Power Consumption for details.
Note 2. Wait for 1 µs or longer to start A/D conversion after release from the module stop state.

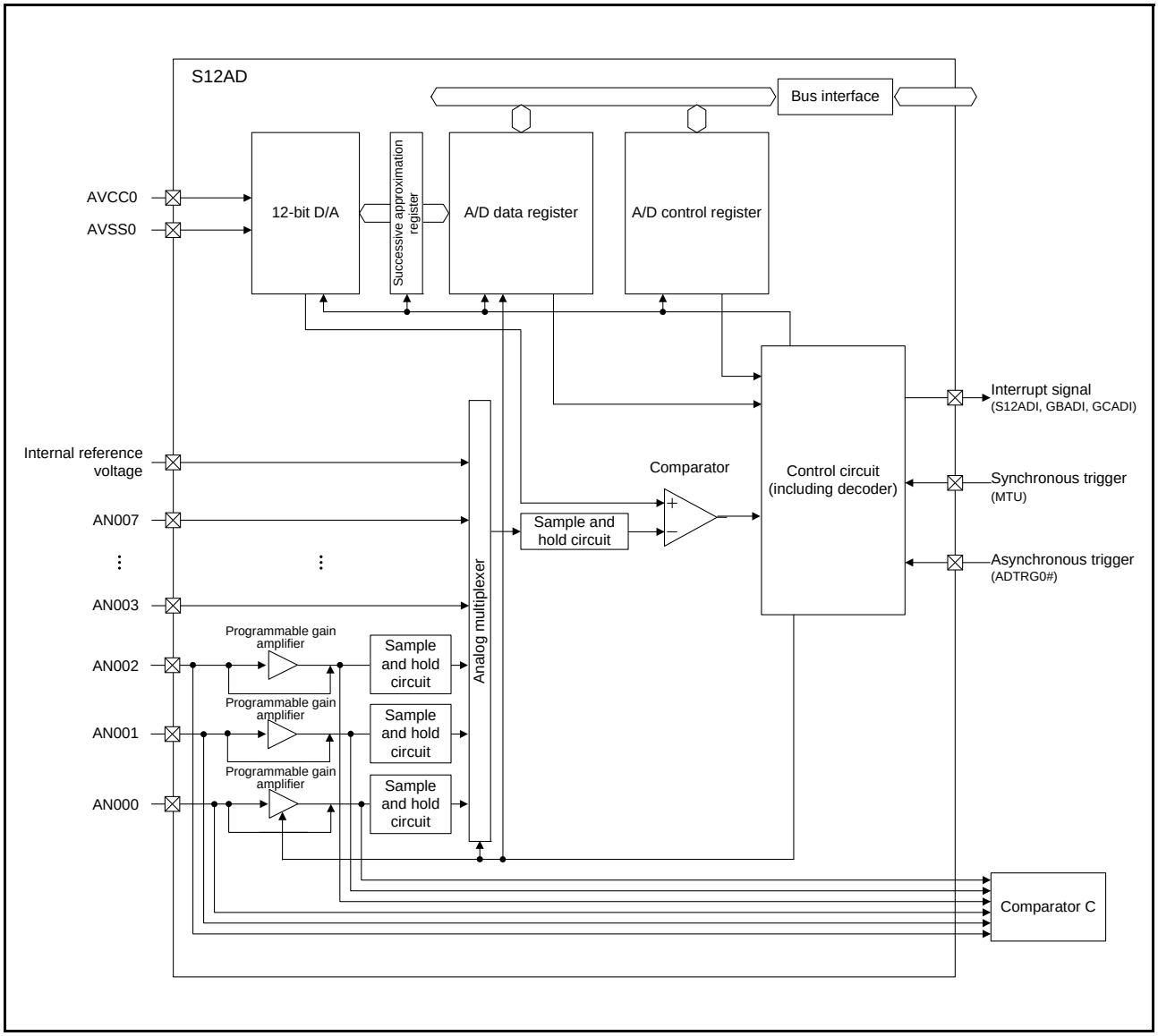


Figure 26.1 Block Diagram of 12-Bit A/D Converter (Unit 0)

Table 26.3 lists the input/output pins of the 12-bit A/D converter.

These units can be operated independently. The input channels of S12AD can be divided into three groups for operation. A programmable gain amplifier (PGA) is included in AN000 to AN002.

Table 26.3 Input/Output Pins of 12-Bit A/D Converter

Unit	Pin Name	I/O	Function	PGA	Internal Sample & Hold Circuit for the Pin
Unit 0 (S12AD)	AN000	Input	Analog input pin	Incorporated	Incorporated
	AN001	Input	Analog input pin	Incorporated	Incorporated
	AN002	Input	Analog input pin	Incorporated	Incorporated
	AN003	Input	Analog input pin	—	—
	AN004	Input	Analog input pin	—	—
	AN005	Input	Analog input pin	—	—
	AN006	Input	Analog input pin	—	—
	AN007	Input	Analog input pin	—	—
	ADTRG0#	Input	External trigger input pin for starting A/D conversion	—	—
	ADST0	Output	ADST bit state output pin	—	—
	AVCC0	—	Analog block power supply pin	—	—
	AVSS0	—	Analog block ground pin	—	—

26.2 Register Descriptions

26.2.1 A/D Data Registers y (ADDRy) (y = 0 to 7), A/D Data Duplication Register (ADDBLDR), A/D Data Duplication Register A (ADDBLDRA), A/D Data Duplication Register B (ADDBLDRB), A/D Internal Reference Voltage Data Register (ADOCDR)

Address(es): S12AD.ADDR0 0008 9020h, S12AD.ADDR1 0008 9022h, S12AD.ADDR2 0008 9024h, S12AD.ADDR3 0008 9026h, S12AD.ADDR4 0008 9028h, S12AD.ADDR5 0008 902Ah, S12AD.ADDR6 0008 902Ch, S12AD.ADDR7 0008 902Eh, S12AD.ADDBLDR 0008 9018h, S12AD.ADDBLDRA 0008 9084h, S12AD.ADDBLDRB 0008 9086h, S12AD.ADOCDR 0008 901Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The ADDRy registers (y = 0 to 7) are 16-bit read-only registers which store the A/D conversion results.
The ADDBLDR register is a 16-bit read-only register used in double trigger mode. The ADDBLDR register stores the results of A/D conversion when the conversion is started by the second trigger.
The ADDBLDRA and ADDBLDRB registers are 16-bit read-only registers that store the A/D conversion results in response to the respective triggers during extended operation in double trigger mode.
The ADOCDR register is a 16-bit read-only register that stores the A/D conversion results of the internal reference voltage.
The format of each register differs depending on the conditions below.

- Settings of the A/D data register format select bit (ADCER.ADRFMT) (flush-right or flush-left)
- Settings of the addition count select bits (ADADC.ADC[2:0]) (2-, 3-, 4-, or 16-time conversion)
- Settings of the average mode enable bit (ADADC.AVEE) (add or average)

The data formats for each given condition are shown below.

(1) When A/D-Converted Value Addition/Average Mode is Not Selected

- Flush-right format
  The A/D-converted value is stored in bits 11 to 0. Bits 15 to 12 are read as 0.
- Flush-left format
  The A/D-converted value is stored in bits 15 to 4. Bits 3 to 0 are read as 0.

(2) When A/D-Converted Average Mode is Selected

- Flush-right format
  The mean value of the A/D-converted results of the same channel is stored in bits 11 to 0.
  Bits 15 to 12 are read as 0.
- Flush-left format
  The mean value of the A/D-converted results of the same channel is stored in bits 15 to 4.
  Bits 3 to 0 are read as 0.

A/D-converted value average mode can be set only when 2- or 4-time conversion is selected in A/D-converted value addition mode.

(3) When A/D-Converted Value Addition Mode is Selected

- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  The value added by the A/D-converted value of the same channel is stored in bits 13 to 0.
  Bits 15 and 14 are read as 0.

- Flush-right format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 1 to 4 times)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 2.
  Bits 1 and 0 are read as 0.
- Flush-left format (in A/D-converted value addition mode and when number of conversions is selected for 16 times)
  The value added by the A/D-converted value of the same channel is stored in bits 15 to 0.

When A/D-converted addition mode is selected, the value added by the A/D-converted value of the same channel is indicated. The number of A/D conversions can be set to 1, 2, 3, 4, or 16 times. If A/D-converted addition mode is selected, when the conversion count is set to 1 to 4 times, the value added by the A/D conversion result is retained in the A/D data register as 2-bit extended data of the conversion accuracy bits; when the conversion count is set to 16 times, the value added by the A/D conversion result is retained in the A/D data register as 4-bit extended data of the conversion accuracy bits. Even if A/D-converted value addition mode is selected, the value is stored in the A/D data register according to the settings of the A/D data register format select bits.

26.2.2 A/D Self-Diagnosis Data Register (ADRD)

Address(es): S12AD.ADRD 0008 901Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ADRD is a 16-bit read-only register that stores the A/D conversion results based on the 12-bit A/D converter's self-diagnosis. In addition to the A/D-converted value, the self-diagnosis status is included in. In the ADRD register, the different formats are used depending on the conditions below.

- Settings of the A/D data register format select bit (ADCER.ADRFMT) (flush-right or flush-left)

The A/D-converted value addition mode and A/D-converted value average mode cannot be applied to the A/D self-diagnosis function. For details of self-diagnosis, see section 26.2.9, A/D Control Extended Register (ADCER).

The data formats for each given condition are shown below.

- Flush-right format
  The A/D-converted value is stored in bits 11 to 0. The self-diagnosis status is stored in bits 15 and 14.
  Bits 13 and 12 are read as 0.
- Flush-left format
  The A/D-converted value is stored in bits 15 to 4. The self-diagnosis status is stored in bits 1 and 0.
  Bits 3 and 2 are read as 0.

Table 26.4 Self-Diagnosis Status Description

Bits 15 and 14 for flush-right format setting Bits 1 and 0 for flush-left format setting	Self-diagnosis status
00b	Self-diagnosis has never been executed since power-on.
01b	Self-diagnosis using the voltage of 0 V has been executed.
10b	Self-diagnosis using the voltage of reference power supply × 1/2 has been executed.
11b	Self-diagnosis using the voltage of reference power supply has been executed.

Note: For details of self-diagnosis, see section 26.2.9, A/D Control Extended Register (ADCER).

26.2.3 A/D Control Register (ADCSR)

Address(es): S12AD.ADCSR 0008 9000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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	Bit Name	Description	R/W
b4 to b0	DBLANS[4:0]	Double Trigger Channel Select	These bits select one analog input channel for double triggered operation. The setting is only effective while double trigger mode is selected.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	GBADIE	Group B Scan End Interrupt Enable	0: Disables interrupt generation upon group B scan completion. 1: Enables interrupt generation upon group B scan completion.	R/W
b7	DBLE	Double Trigger Mode Select	0: Deselects double trigger mode. 1: Selects double trigger mode.	R/W
b8	EXTRG	Trigger Select*1	0: A/D conversion is started by synchronous trigger. 1: A/D conversion is started by asynchronous trigger.	R/W
b9	TRGE	Trigger Start Enable	0: Disables A/D conversion to be started by synchronous or asynchronous trigger. 1: Enables A/D conversion to be started by synchronous or asynchronous trigger.	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	ADIE	Scan End Interrupt Enable	0: Disables interrupt generation upon scan completion. 1: Enables interrupt generation upon scan completion.	R/W
b14, b13	ADCS[1:0]	Scan Mode Select	b14 b13 0 0: Single scan mode 0 1: Group scan mode 1 0: Continuous scan mode 1 1: Setting prohibited	R/W
b15	ADST	A/D Conversion Start	0: Stops A/D conversion process. 1: Starts A/D conversion process.	R/W

Note 1. Starting A/D conversion using an external pin (asynchronous trigger)
Set the ADCSR.TRGE and EXTRG bits to 1 while a high-level signal is input to the external pin (ADTRG0#). Then, if the ADTRG0# signal is changed to low, the falling edge is detected and the scan process is started. In this case, the pulse width of the low-level input must be at least 1.5 clock cycles of PCLK.

ADCSR sets double trigger mode, A/D conversion start trigger; enables/disables scan end interrupt; selects the scan mode; and starts or stops A/D conversion.

DBLANS[4:0] Bits (Double Trigger Channel Select)

The DBLANS[4:0] bits select one of the channels for A/D conversion data duplication in double trigger mode. The A/D conversion results of the analog input of the channel selected by the DBLANS[4:0] bits are stored into the A/D data register y when conversion is started by the first trigger, and into the A/D data duplication register when started by the second trigger. Table 26.5 shows selection of the channel for double triggered operation.

When double trigger mode is selected, the channels selected by the ADANSA0 register are invalid, and the channel selected by the DBLANS[4:0] bits is subjected to A/D conversion instead.

When double trigger mode is selected in group scan mode, double trigger mode operation is performed for group A only and not performed for group B or C. Also, in double trigger mode, the analog inputs of multiple internal reference voltage cannot be selected for group A, but can be selected for groups B and C.

The DBLANS[4:0] bits should be set while the ADST bit is 0. They should not be set simultaneously when 1 is written to the ADST bit.

Table 26.5 Relationship between DBLANS[4:0] Bits Settings and Double Trigger Enabled Channels

DBLANS[4:0]	Duplication Channel
00000b	AN000
00001b	AN001
00010b	AN002
00011b	AN003
00100b	AN004
00101b	AN005
00110b	AN006
00111b	AN007

Note: Duplication cannot be selected for the A/D conversion data of self-diagnosis and internal reference voltage.

GBADIE Bit (Group B Scan End Interrupt Enable)

The GBADIE bit enables or disables group B scan end interrupt in group scan mode.

DBLE Bit (Double Trigger Mode Select)

Double trigger mode has a function to store the resulting data of A/D conversion started by the first and second synchronous triggers into separate registers.

When double trigger mode is selected, the channels specified in the ADANSA0 register are invalid and the channel selected by the DBLANS[4:0] bits is effective instead. Double trigger mode can be only operated by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits. Do not generate an asynchronous or software trigger. The A/D conversion results started by the first trigger are stored into the A/D data register y and those started by the second trigger are stored into the A/D data duplication register. In this case, if the ADIE bit is set to 1, the interrupt is generated not upon completion of the first conversion but upon completion of the second conversion.

In continuous scan mode, double trigger mode should not be selected. In addition, double trigger mode should not be used for self-diagnosis or conversion of the internal reference voltage. When using double trigger mode in group scan mode, A/D conversion of the internal reference voltage should not be selected for group A.

The DBLE bit should be set after the ADST bit has been set to 0.

EXTRG Bit (Trigger Select)

The EXTRG bit selects the synchronous trigger or the asynchronous trigger as the trigger for starting A/D conversion. In group scan mode, the setting of this bit is valid for the selected trigger of group A.

For groups B and C, A/D conversion is started by the selected synchronous trigger regardless of this bit setting.

TRGE Bit (Trigger Start Enable)

The TRGE bit enables or disables A/D conversion by the synchronous trigger and the asynchronous trigger.

This bit should be set to 1 in group scan mode.

ADIE Bit (Scan End Interrupt Enable)

The ADIE bit enables or disables the A/D scan end interrupt (S12ADI) in scans except for groups B and C scan in group scan mode.

With double trigger mode deselected, the A/D scan conversion end is generated after the first scan is completed if the ADIE bit is set to 1.

With double trigger mode selected, the A/D scan conversion end is generated after the second scan is completed if the ADIE bit is set to 1 as long as the scan is started by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits.

When scan is started by a software trigger, even with double trigger mode selected, the A/D scan conversion end interrupt is generated if the ADIE bit is set to 1 when the scan is completed.

ADCS[1:0] Bits (Scan Mode Select)

The ADCS[1:0] bits select the scan mode.
In single scan mode, A/D conversion is performed for the analog inputs selected with the ADANSA0 register in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, the scan conversion is stopped.
In continuous scan mode, while the ADCSR.ADST bit is 1, A/D conversion is performed for the analog inputs selected with the ADANSA0 register in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is repeated from the first channel. If the ADCSR.ADST bit is set to 0 during continuous scan, A/D conversion is stopped even if scanning is in progress.
In group scan mode, A/D conversion is performed for the analog inputs (group A) selected with the ADANSA0 register in the ascending order of the channel number after scanning is started by the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits, and when one cycle of A/D conversion is completed for all the selected channels, A/D conversion is stopped. A/D conversion is also performed for the analog inputs (group B or C) selected with the ADANSB0 register and the ADANSC0 register in the ascending order of the channel number after A/D conversion is started by the synchronous trigger selected by the ADSTRGR.TRSB[5:0] bits and ADGCTRGR.TRSC[5:0] bits, and when A/D conversion is completed for all the selected channels, A/D conversion is stopped.
When selecting group scan mode, different channels and triggers should be selected for groups A, B, and C.
When using two groups while group scan mode is set, use groups A and B (ADGCTRGR.GRCE bit = 0). When using three groups, use groups A, B, and C (ADGCTRGR.GRCE bit = 1).
When selecting the internal reference voltage, select single scan mode, and deselect all the channels selected with the ADANSA0 register before performing A/D conversion. When A/D conversion of the selected internal reference voltage is completed, A/D conversion is stopped.
The ADCS[1:0] bits should be set while the ADST bit is 0. They should not be set simultaneously when 1 is written to the ADST bit.

ADST Bit (A/D Conversion Start)

The ADST bit starts or stops A/D conversion process.
Before the ADST bit is set to 1, set the A/D conversion clock, the conversion mode, and conversion target analog input.
[Setting conditions]

- 1 is written by software.
- The synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits is detected with ADCSR.EXTRG and ADCSR.TRGE bits being set to 0 and 1, respectively.
- The synchronous trigger selected by the ADSTRGR.TRSB[5:0] bits is detected with the ADCSR.TRGE bit being set to 1 in group scan mode.
- The asynchronous trigger is detected with the ADCSR.TRGE and ADCSR.EXTRG bits being set to 1 and the ADSTRGR.TRSA[5:0] bits being set to 000000b.
- With group priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), a group B or C trigger is detected and A/D conversion of group B or C is started.
- With group priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRSCN bit is set to 1 and A/D conversion of group B or C is restarted.
- With group priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRP bit is set to 1 and A/D conversion of the lowest-priority group is started.

[Clearing conditions]

- 0 is written by software.
- The A/D conversion of all the selected channels or the internal reference voltage is completed in single scan mode.
- Group A scan is completed in group scan mode.
- Group B scan is completed in group scan mode.
- Group C scan is completed in group scan mode.

- With group priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADGSPCR.GBRSCN bit is set to 1 and the scanning of the low-priority group started by a trigger is stopped.

Note: When group priority control operation mode has been enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), do not set the ADST bit to 1.

Note: When group priority control operation mode has been enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1) and ADGSPCR.GBRP = 1, do not set the ADST bit to 0. When forcibly terminating A/D conversion, follow the procedure for clearing the ADST bit.

Note: When the single-scan continuous function is used (ADGSPCR.GBRP bit = 1) with group priority control operation mode enabled (ADCSR.ADCS[1:0] bits = 01b and ADGSPCR.PGS bit = 1), the ADST bit remains as 1.

26.2.4 A/D Channel Select Register A0 (ADANSA0)

(1) S12AD.ADANSA0

Address(es): 0008 9004h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ANSA007	ANSA006	ANSA005	ANSA004	ANSA003	ANSA002	ANSA001	ANSA000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSA000	A/D Conversion Channel Select	0: AN000 to AN007 are not subjected to conversion. 1: AN000 to AN007 are subjected to conversion.	R/W
b1	ANSA001			R/W
b2	ANSA002			R/W
b3	ANSA003			R/W
b4	ANSA004			R/W
b5	ANSA005			R/W
b6	ANSA006			R/W
b7	ANSA007			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

S12AD.ADANSA0 selects analog input channels for A/D conversion from among AN000 to AN007. In group scan mode, this register selects group A channels.

ANSA0n Bit (n = 00 to 07) (A/D Conversion Channel Select)

The ANSA0n bit select analog input channels for A/D conversion from among AN000 to AN007. The channels to be selected and the number of channels can be arbitrarily set. The ANSA000 bit corresponds to AN000 and the ANSA007 bit corresponds to AN007.

When double trigger mode is selected, the channel selected by S12AD.ADCSR.DBLANS[4:0] bits is selected in group A, and the ANSA0n bit setting is invalid.

The ANSA0n bit should be set while the S12AD.ADCSR.ADST bit is 0.

26.2.5 A/D Channel Select Register B0 (ADANSB0)

(1) S12AD.ADANSB0

Address(es): 0008 9014h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ANSB007	ANSB006	ANSB005	ANSB004	ANSB003	ANSB002	ANSB001	ANSB000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSB000	A/D Conversion Channel Select	0: AN000 to AN007 are not subjected to conversion. 1: AN000 to AN007 are subjected to conversion.	R/W
b1	ANSB001			R/W
b2	ANSB002			R/W
b3	ANSB003			R/W
b4	ANSB004			R/W
b5	ANSB005			R/W
b6	ANSB006			R/W
b7	ANSB007			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

S12AD.ADANSB0 selects analog input channels for A/D conversion from among AN000 to AN007 in group B when group scan mode is selected. The S12AD.ADANSB0 register is not used in any scan mode other than group scan mode.

ANSB0n Bit (n = 00 to 07) (A/D Conversion Channel Select)

The ANSB0n bit select analog input channels for A/D conversion from among AN000 to AN007 in group B when group scan mode is selected. The S12AD.ADANSB0 register is used for group scan mode only; not used for any other modes. The channels specified in group A (the channels corresponding to group A, selected with the S12AD.ADANSA0 register and the S12AD.ADCSR.DBLANS[4:0] bits in double trigger mode) should be excluded as the channels to be selected and the number of channels to be set.

The ANSB000 bit corresponds to AN000 and the ANSB007 bit corresponds to AN007.

The ANSB0n bit should be set while the S12AD.ADCSR.ADST bit is 0.

26.2.6 A/D Channel Select Register C0 (ADANSC0)

(1) S12AD.ADANSC0

Address(es): 0008 90D4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ANSC007	ANSC006	ANSC005	ANSC004	ANSC003	ANSC002	ANSC001	ANSC000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ANSC000	A/D Conversion Channel Select	0: AN000 to AN007 are not subjected to conversion. 1: AN000 to AN007 are subjected to conversion.	R/W
b1	ANSC001			R/W
b2	ANSC002			R/W
b3	ANSC003			R/W
b4	ANSC004			R/W
b5	ANSC005			R/W
b6	ANSC006			R/W
b7	ANSC007			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

S12AD.ADANSC0 selects analog input channels for A/D conversion from among AN000 to AN007 in group C when group scan mode is selected. The S12AD.ADANSC0 register is not used in any scan mode other than group scan mode.

ANSC0n Bit (n = 00 to 07) (A/D Conversion Channel Select)

The ANSC0n bit select analog input channels for A/D conversion from among AN000 to AN007 in group C when group scan mode is selected. The S12AD.ADANSC0 register is used for group scan mode only; not used for any other modes. The ANSC000 bit corresponds to AN000 and the ANSC007 bit corresponds to AN007.
The ANSC0n bit should be set while the S12AD.ADCSR.ADST bit is 0.

26.2.7 A/D-Converted Value Addition/Average Function Channel Select Register 0 (ADADS0)

(1) S12AD.ADADS0

Address(es): 0008 9008h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	ADS007	ADS006	ADS005	ADS004	ADS003	ADS002	ADS001	ADS000
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ADS000	A/D-Converted Value Addition/ Average Channel Select	0: A/D-converted value addition/average mode for AN000 to AN007 is not selected. 1: A/D-converted value addition/average mode for AN000 to AN007 is selected.	R/W
b1	ADS001			R/W
b2	ADS002			R/W
b3	ADS003			R/W
b4	ADS004			R/W
b5	ADS005			R/W
b6	ADS006			R/W
b7	ADS007			R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

S12AD.ADADS0 selects channels AN000 to AN007 on which A/D conversion is performed successively 2, 3, 4, or 16 times and then converted values are added (integrated) or averaged.

ADS0n Bit (n = 00 to 07) (A/D-Converted Value Addition/Average Channel Select)

When the ADS0n bit of the number that is the same as that of A/D-converted channel selected by the S12AD.ADANSA0.ANSA0n bit (n = 00 to 07) or S12AD.ADCSR.DBLANS[4:0] bits and S12AD.ADANSB0.ANSB0n bit is set to 1, A/D conversion of analog input of the selected channels is performed successively 2, 3, 4, or 16 times that is set with the S12AD.ADADC.ADC[2:0] bits.

When the S12AD.ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D data register.

When the S12AD.ADADC.AVEE bit is 1, the mean value of the results obtained by addition (integration) is stored in the A/D data register.

As for the channel on which the A/D conversion is performed and addition/average mode is not selected, a normal one-time conversion is executed and the conversion result is stored to the A/D data register.

The ADS0n bit should be set while the S12AD.ADCSR.ADST bit is 0.

26.2.8 A/D-Converted Value Addition/Average Count Select Register (ADADC)

Address(es): S12AD.ADADC 0008 900Ch

	b7	b6	b5	b4	b3	b2	b1	b0
	AVEE	—	—	—	—	ADC[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	ADC[2:0]	Addition Count Select	b2 b0 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 Settings other than above are prohibited.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	AVEE	Average Mode Enable	0: Addition mode is selected. 1: Average mode is selected.	R/W

Note 1. The AVEE bit is enabled only when 2-time or 4-time conversion is selected. When average mode is selected (ADADC.AVEE bit = 1), do not set 3-time conversion (ADADC.ADC[2:0] = 010b) nor 16-time conversion (ADADC.ADC[2:0] = 101b).

ADADC sets the addition count for A/D conversion of the channel, and internal reference voltage for which A/D-converted value addition/average mode is selected, and selects either addition or average mode.

ADC[2:0] Bits (Addition Count Select)

The ADC[2:0] bits set the addition count common to the channels for which A/D conversion and A/D-converted value addition/average mode is selected, including the channels selected in double trigger mode (by ADCSR.DBLANS[4:0] bits), and to A/D conversion of internal reference voltage.
When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the addition count to one time (ADADC.ADC[2:0] = 000b), three times (ADADC.ADC[2:0] = 010b), or 16 times (ADADC.ADC[2:0] = 101b).
The ADC[2:0] bits should be set while the ADCSR.ADST bit is 0.

AVEE Bit (Average Mode Enable)

The AVEE bit selects addition or average mode for A/D conversion of the channel for which A/D conversion and A/D-converted value addition/average mode is selected, including the channels selected in double trigger mode (by ADCSR.DBLANS[4:0] bits) and internal reference voltage.
When average mode is selected by setting the ADADC.AVEE bit to 1, do not set the addition count to one time (ADADC.ADC[2:0] = 000b), three times (ADADC.ADC[2:0] = 010b), or 16 times (ADADC.ADC[2:0] = 101b). The mean value of 1-time, 3-time, and 16-time conversion cannot be obtained.
The AVEE bit should be set while the ADCSR.ADST bit is 0.

26.2.9 A/D Control Extended Register (ADCER)

Address(es): S12AD.ADCER 0008 900Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADRFMT	—	—	—	DIAGM	DIAGLD	DIAGVAL[1:0]		—	—	ACE	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	ACE	A/D Data Register Automatic Clearing Enable	0: Disables automatic clearing. 1: Enables automatic clearing.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	DIAGVAL[1:0]	Self-Diagnosis Conversion Voltage Select	b9 b8 0 0: Setting prohibited in self-diagnosis voltage fixed mode 0 1: Uses the voltage of 0 V for self-diagnosis. 1 0: Uses the voltage of reference power supply × 1/2 for self-diagnosis. 1 1: Uses the voltage of reference power supply for self-diagnosis.	R/W
b10	DIAGLD	Self-Diagnosis Mode Select	0: Rotation mode for self-diagnosis voltage 1: Fixed mode for self-diagnosis voltage	R/W
b11	DIAGM	Self-Diagnosis Enable	0: Disables self-diagnosis of 12-bit A/D converter. 1: Enables self-diagnosis of 12-bit A/D converter.	R/W
b14 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	ADRFMT	A/D Data Register Format Select	0: Flush-right is selected for the A/D data register format. 1: Flush-left is selected for the A/D data register format.	R/W

ADCER sets self-diagnosis mode, format of A/D data registers y (ADDRy), and automatic clearing of A/D data registers.

ACE Bit (A/D Data Register Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing (all 0) of ADDRy, ADRD, ADDBLDR, ADDBLDRA, ADDBLDRB, or ADOCDR after any of these registers have been read by the CPU and DTC. Automatic clearing of the A/D data register is enabled to detect a failure which has not been updated in the A/D data register.

DIAGVAL[1:0] Bits (Self-Diagnosis Conversion Voltage Select)

The DIAGVAL[1:0] bits select the voltage value used in self-diagnosis voltage fixed mode. For details, refer to the descriptions of the ADCER.DIAGLD bit.

Self-diagnosis should not be executed by setting the ADCER.DIAGLD bit to 1 when the ADCER.DIAGVAL[1:0] bits are set to 00b.

DIAGLD Bit (Self-Diagnosis Mode Select)

The DIAGLD bit selects whether the three voltage values are rotated or the fixed voltage is used in self-diagnosis. Setting this bit (ADCER.DIAGLD) to 0 allows conversion of the voltages in rotation mode where 0, the reference power supply × 1/2, and the reference power supply are converted in this order. When self-diagnosis rotation mode is selected after a reset, self-diagnosis is performed from 0 V. When self-diagnosis voltage fixed mode is selected, the fixed voltage specified by the ADCER.DIAGVAL[1:0] bits is converted. In self-diagnosis voltage rotation mode, the self-diagnosis voltage value does not return to 0 when scan conversion is completed. When scan conversion is restarted, therefore, rotation starts at the voltage value following the previous value. If fixed mode is switched to rotation mode, rotation

starts at the fixed voltage value.
The DIAGLD bit should be set while the ADCSR.ADST bit is 0.

DIAGM Bit (Self-Diagnosis Enable)

The DIAGM bit enables or disables self-diagnosis.
Self-diagnosis is used to detect a failure of the 12-bit A/D converter. Specifically, one of the internally generated voltage values 0, the reference power supply × 1/2, and the reference power supply is converted. When conversion is completed, information on the converted voltage and the conversion result is stored into the self-diagnosis data register (ADRD). ADRD can then be read out by software to determine whether the conversion result falls within the normal range (normal) or not (abnormal). Self-diagnosis is executed once at the beginning of each scan, and one of the three voltages is converted. When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed in groups A, B, and C.
The DIAGM bit should be set while the ADCSR.ADST bit is 0.

ADRFMT Bit (A/D Data Register Format Select)

The ADRFMT bit specifies flush-right or flush-left for the data to be stored in ADDRy, ADDBLDR, ADDBLDRA, ADDBLDRB, ADOCDR, or ADRD.
The ADRFMT bit should be set while the ADCSR.ADST bit is 0.

For details on the format of each data register, see section 26.2.1, A/D Data Registers y (ADDRy) (y = 0 to 7), A/D Data Duplication Register (ADDBLDR), A/D Data Duplication Register A (ADDBLDRA), A/D Data Duplication Register B (ADDBLDRB), A/D Internal Reference Voltage Data Register (ADOCDR), and section 26.2.2, A/D Self-Diagnosis Data Register (ADRD).

26.2.10 A/D Conversion Start Trigger Select Register (ADSTRGR)

Address(es): S12AD.ADSTRGR 0008 9010h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	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	Bit Name	Description	R/W
b5 to b0	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.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13 to b8	TRSA[5:0]	A/D Conversion Start Trigger Select	Select the A/D conversion start trigger in single scan mode and continuous scan mode. In group scan mode, the A/D conversion start trigger for group A is selected.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADSTRGR selects the A/D conversion start trigger.

TRSB[5:0] Bits (A/D Conversion Start Trigger Select for Group B)

The TRSB[5:0] bits select the trigger to start scanning of the analog input selected in group B. The TRSB[5:0] bits require to be set only in group scan mode and are not used in any other scan mode. For the scan conversion start trigger for group B, setting a software trigger or an asynchronous trigger is prohibited. Therefore, the TRSB[5:0] bits should be set to the value other than 000000b and the ADCSR.TRGE bit should be set to 1 in group scan mode.

When two groups are selected (ADGCTRGR.GRCE = 0) during group priority operation in group scan mode, setting the ADGSPCR.GBRP bit to 1 allows group B to continuously operate in single scan mode. When setting the ADGSPCR.GBRP bit to 1, set the TRSB[5:0] bits to 3Fh. Note that the issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance period is less than t_{SCAN}, A/D conversion by the trigger may have no effect.

When the trigger from the module (MTU) operated in PCLKB is selected as an A/D conversion start trigger, a delay of the period for synchronization processing occurs. See section 26.3.5, Analog Input Sampling Time and Scan Conversion Time for details.

Table 26.6 lists the A/D conversion startup sources selected by the TRSB[5:0] bits.

TRSA[5:0] Bits (A/D Conversion Start Trigger Select)

The TRSA[5:0] bits select the trigger to start A/D conversion in single scan mode and continuous scan mode. In group scan mode, the trigger to start scanning of the analog input selected in group A is selected. When scanning is executed in group scan mode or double trigger mode, set the ADCSR.TRGE bit to 1.

- When using the A/D conversion startup source of a synchronous trigger, set the ADCSR.TRGE bit to 1 and set the ADCSR.EXTRG bit to 0.
- When using the asynchronous trigger, set the ADCSR.TRGE bit to 1 and set the ADCSR.EXTRG bit to 1.
- Software trigger (ADCSR.ADST) is enabled regardless of the settings of the ADCSR.TRGE bit, the ADCSR.EXTRG bit, and the TRSA[5:0] bits.

Note that the issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance period is less than t_{SCAN}, A/D conversion by a trigger may have no effect. When the trigger from the module (MTU) operated in PCLKB is selected as an A/D conversion start trigger, a delay of the period for synchronization processing occurs. See section 26.3.5, Analog Input Sampling Time and Scan Conversion Time

for details.

Table 26.7 lists the selection of A/D conversion start sources selected by the TRSA[5:0] bits.

Table 26.6 Selection of A/D Activation Sources by the TRSB[5:0] Bits

Module	Source	Remarks	TRSB[5]	TRSB[4]	TRSB[3]	TRSB[2]	TRSB[1]	TRSB[0]
Trigger source deselection state			1	1	1	1	1	1
MTU	TRGA0N	Compare match/input capture from MTU0.TGRA	0	0	0	0	0	1
	TRGA1N	Compare match/input capture from MTU1.TGRA	0	0	0	0	1	0
	TRGA2N	Compare match/input capture from MTU2.TGRA	0	0	0	0	1	1
	TRGA3N	Compare match/input capture from MTU3.TGRA	0	0	0	1	0	0
	TRGA4N	Compare match/input capture from MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	0	0	0	1	0	1
	TRG0N	Compare match from MTU0.TGRE	0	0	1	0	0	0
	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT	0	0	1	0	0	1
	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	0
	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
	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT, and compare match between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is used)	0	0	1	1	0	0

Table 26.7 Selection of A/D Activation Sources by the TRSA[5:0] Bits

Module	Source	Remarks	TRSA[5]	TRSA[4]	TRSA[3]	TRSA[2]	TRSA[1]	TRSA[0]
Trigger source deselection state			1	1	1	1	1	1
External pin	ADTRG0#	Trigger input pin	0	0	0	0	0	0
MTU	TRGA0N	Compare match/input capture from MTU0.TGRA	0	0	0	0	0	1
	TRGA1N	Compare match/input capture from MTU1.TGRA	0	0	0	0	1	0
	TRGA2N	Compare match/input capture from MTU2.TGRA	0	0	0	0	1	1
	TRGA3N	Compare match/input capture from MTU3.TGRA	0	0	0	1	0	0
	TRGA4N	Compare match/input capture from MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	0	0	0	1	0	1
	TRG0N	Compare match from MTU0.TGRE	0	0	1	0	0	0
	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT	0	0	1	0	0	1
	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	0
	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
	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT, and compare match between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is used)	0	0	1	1	0	0

26.2.11 A/D Conversion Extended Input Control Register (ADEXICR)

Address(es): S12AD.ADEXICR 0008 9012h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	OCSA	—	—	—	—	—	—	—	OCSAD	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	OCSAD	Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select	0: Internal reference voltage A/D-converted value addition/average mode is not selected. 1: Internal reference voltage A/D-converted value addition/average mode is selected.	R/W
b8 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9	OCSA	Internal Reference Voltage A/D Conversion Select	0: A/D conversion of internal reference voltage is not performed. 1: A/D conversion of internal reference voltage is performed.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADEXICR specifies the settings of A/D conversion of the internal reference voltage.

OCSAD Bit (Internal Reference Voltage A/D-Converted Value Addition/Average Mode Select)

When the OCSAD bit is set to 1, A/D conversion of the internal reference voltage is selected and performed successively 2, 3, 4, or 16 times that is set with the ADADC.ADC[2:0] bits. When the ADADC.AVEE bit is 0, the value obtained by addition (integration) is stored in the A/D internal reference voltage data register (ADOCDR). When the ADADC.AVEE bit is 1, the mean value is stored in ADOCDR.

The OCSAD bit should be set while the ADCSR.ADST bit is 0.

OCSA Bit (Internal Reference Voltage A/D Conversion Select)

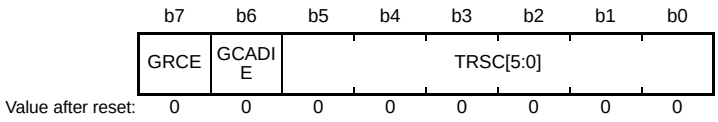
This bit selects A/D conversion of the internal reference voltage in single scan mode. When A/D conversion of the internal reference voltage is to be performed, set all the bits in the ADANSA0 and ADANSB0 registers and the ADCSR.DBLE bit should be set to all 0 in single scan mode.

The OCSA bit should be set while the ADCSR.ADST bit is 0. For A/D conversion of the internal reference voltage, the ADDISCR.ADNDIS[4:0] bits should be automatically set to 0Fh to discharge the A/D converter before sampling. The sampling time should be 5 µs or longer.

Sampling starts after discharging is completed during A/D conversion of the internal reference voltage, so an auto-discharging period of 15 ADCLK cycles is inserted before sampling.

26.2.12 A/D Group C Trigger Select Register (ADGCTRGR)

Address(es): S12AD.ADGCTRGR 0008 90D9h



Bit	Symbol	Bit Name	Description	R/W
b5 to b0	TRSC[5:0]	Group C A/D Conversion Start Trigger Select	Select the A/D conversion start trigger for group C in group scan mode.	R/W
b6	GCADIE	Group C Scan End Interrupt Enable	0: Disables interrupt generation after completion of group C scan 1: Enables interrupt generation after completion of group C scan	R/W
b7	GRCE	Group C A/D Conversion Operation Enable	Enables A/D conversion operation for group C. 0: Group C is not used 1: Group C is used	R/W

ADGCTRGR enables operation for group C and selects the A/D conversion start trigger. For details on group priority operation, see Table 26.11 and Table 26.12.

TRSC[5:0] Bits (Group C A/D Conversion Start Trigger Select)

These bits select the trigger to start scanning of the analog input selected in group C. These bits are used for group scan mode only; not used for any other modes. Software trigger or asynchronous trigger cannot be set as the scan conversion trigger for group C. When using group C in group scan mode, set the TRSC[5:0] bits to a value other than 000000b, set the ADCSR.TRGE bit to 1, and set the GRCE bit to 1.

When group C is used during group priority control in group scan mode and the ADGSPCR.GBRP bit is set to 1, group C can be continuously operated in single scan mode. When continuously operating group C in single scan mode, set the TRSC[5:0] bits to 3Fh and disable trigger selection.

Note that the issuance period of trigger for A/D conversion must be more than or equal to the actual scan conversion time (t_{SCAN}). If the issuance period is less than tSCAN, A/D conversion by a trigger may have no effect.

When the trigger from the module (MTU) operated in PCLKB is selected as an A/D conversion start trigger, a delay of the period for synchronization processing occurs. See section 26.3.5, Analog Input Sampling Time and Scan Conversion Time for details.

Table 26.8 lists the selection of A/D conversion start sources selected by the TRSA[5:0] bits for group C.

Table 26.8 Selection of A/D Activation Sources by the TRSC[5:0] Bits

Module	Source	Remarks	TRSC[5]	TRSC[4]	TRSC[3]	TRSC[2]	TRSC[1]	TRSC[0]
Trigger source deselection state			1	1	1	1	1	1
MTU	TRGA0N	Compare match/input capture from MTU0.TGRA	0	0	0	0	0	1
	TRGA1N	Compare match/input capture from MTU1.TGRA	0	0	0	0	1	0
	TRGA2N	Compare match/input capture from MTU2.TGRA	0	0	0	0	1	1
	TRGA3N	Compare match/input capture from MTU3.TGRA	0	0	0	1	0	0
	TRGA4N	Compare match/input capture from MTU4.TGRA or underflow (trough) of MTU4.TCNT in complementary PWM mode	0	0	0	1	0	1
	TRG0N	Compare match from MTU0.TGRE	0	0	1	0	0	0
	TRG4AN	Compare match between MTU4.TADCORA and MTU4.TCNT	0	0	1	0	0	1
	TRG4BN	Compare match between MTU4.TADCORB and MTU4.TCNT	0	0	1	0	1	0
	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
	TRG4ABN	Compare match between MTU4.TADCORA and MTU4.TCNT, and compare match between MTU4.TADCORB and MTU4.TCNT (when interrupt skipping function 2 is used)	0	0	1	1	0	0

GCADIE Bit (Group C Scan End Interrupt Enable)

This bit enables or disables scan end interrupt generation for group C.

GRCE Bit (Group C A/D Conversion Operation Enable)

When using group C in group scan mode, set the GRCE bit to 1.

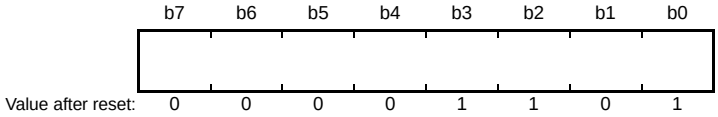
When the GRCE bit is 0, trigger input for group C is disabled.

During group priority operation (the ADGSPCR.PGS bit is 1) with group C used, when the ADGSPCR.GBRP bit is set to 1, single scan for group C is continuously operated. When the GRCE bit is set to 1, single scan for group B is not continuously operated.

The GRCE bit should be set while the ADCSR.ADST bit is 0.

26.2.13 A/D Sampling State Register n (ADSSTRn) (n = 0 to 7, O)

Address(es): S12AD.ADSSTR0 0008 90E0h, S12AD.ADSSTR1 0008 90E1h, S12AD.ADSSTR2 0008 90E2h, S12AD.ADSSTR3 0008 90E3h, S12AD.ADSSTR4 0008 90E4h, S12AD.ADSSTR5 0008 90E5h, S12AD.ADSSTR6 0008 90E6h, S12AD.ADSSTR7 0008 90E7h, S12AD.ADSSTRO 0008 90DFh



The ADSSTRn register sets the sampling time for analog input.
If one state is one ADCLK (A/D conversion clock) cycle and the ADCLK clock is 32 MHz, one state is 31.25 ns. The initial value is 13 states. If the impedance of analog input signal source is too high to secure sufficient sampling time or if the ADCLK clock is slow, the sampling time can be adjusted. The ADSSTRn register should be set while the ADCSR.ADST bit is 0. The lower-limit value for sampling time differs depending on the PCLK to ADCLK frequency ratio.
Set a value that is 5 states or more when PCLK to ADCLK frequency ratio = 1:1, 2:1, 4:1, or 8:1.

Table 26.9 shows the relationship between the A/D sampling state register and the relevant channels.
For details, refer to section 26.3.5, Analog Input Sampling Time and Scan Conversion Time.

Table 26.9 Relationship between A/D Sampling State Register and Relevant Channels

Unit	Register Name	Relevant Channels
S12AD	ADSSTR0 register	AN000, Self-Diagnosis
	ADSSTR1 register	AN001
	ADSSTR2 register	AN002
	ADSSTR3 register	AN003
	ADSSTR4 register	AN004
	ADSSTR5 register	AN005
	ADSSTR6 register	AN006
	ADSSTR7 register	AN007
	ADSSTRO register	Internal reference voltage*1

Note 1. When performing A/D conversion of the internal reference voltage, the sampling time should be 5 µs or longer.

26.2.14 A/D Sample-and-Hold Circuit Control Register (ADSHCR)

Address(es): S12AD.ADSHCR 0008 9066h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SHANS[2:0]			SSTSH[7:0]							
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	SSTSH[7:0]	Channel-Dedicated Sample-and-Hold Circuit Sampling Time Setting	Set the sampling time (4 to 255 states).	R/W
b10 to b8	SHANS[2:0]	Channel-Dedicated Sample-and-Hold Circuit Bypass Select	Select whether to use or not use (bypass) AN000 to AN002 channel-dedicated sample-and-hold circuits. 0: Bypass the channel-dedicated sample-and-hold circuits. 1: Use the channel-dedicated sample-and-hold circuits.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADSHCR sets the parameters related to channel-dedicated sample-and-hold circuits.

SSTSH[7:0] Bits (Channel-Dedicated Sample-and-Hold Circuit Sampling Time Setting)

These bits set the sampling time for the channel-dedicated sample-and-hold circuits. If one state is one ADCLK (A/D conversion clock) cycle and the ADCLK clock is 32 MHz, one state is 31.25 ns. The initial value is 26 states. If the impedance of analog input signal source is too high to secure sufficient sampling time or if the ADCLK clock is slow, the sampling time can be adjusted. The SSTSH[7:0] bits should be set while the ADCSR.ADST bit is 0. The sampling time must be set to a value that is 4 states or more and is 255 or less. Also, the sampling state setting value should be at least the specified test condition in section 32.5, A/D Conversion Characteristics.

SHANS[2:0] Bits (Channel-Dedicated Sample-and-Hold Circuit Bypass Select)

These bits select whether to use or not use (bypass) AN000 to AN002 channel-dedicated sample-and-hold circuits. The SHANS[0] bit selects AN000, SHANS[1] bit selects AN001, and SHANS[2] bit selects AN002. The SHANS[2:0] bits should be set while the ADCSR.ADST bit is 0.

If any channel from among AN000 to AN002 is selected for group B or C while operation is in group scan mode under group priority control, make the setting to bypass the channel-dedicated sample-and-hold circuit.

26.2.15 A/D Disconnection Detection Control Register (ADDISCR)

Address(es): S12AD.ADDISCR 0008 907Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	ADNDIS[4:0]				
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	ADNDIS[4:0]	A/D Disconnection Detection Assist Setting	b4 ADNDIS[4]: Discharge/precharge selected 0: Discharge 1: Precharge b3 to b0 ADNDIS[3:0]: Discharge/precharge period	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

ADDISCR sets the disconnection detection assist function.

ADNDIS[4:0] Bits (A/D Disconnection Detection Assist Setting)

These bits select either precharge or discharge and the period of precharge/discharge for the A/D disconnection detection assist function. Setting the ADNDIS[4] bit = 1 allows to select precharge and setting the ADNDIS[4] bit = 0 allows to select discharge. The period of precharge/discharge can be set with the ADNDIS[3:0] bits. When the ADNDIS[3:0] bits = 0000b, the disconnection detection assist function is not effective. Setting of the ADNDIS[3:0] bits to 0001b is prohibited. Except for the case of ADNDIS[3:0] = 0000b or 0001b, the specified value indicates the number of states for the period of precharge/discharge. The ADNDIS[4:0] bits should be set when the ADCSR.ADST bit is 0. When ADNDIS[3:0] are set to any value other than 0000b and the disconnection detection assist function is enabled, the channel-dedicated disconnection detection assist function is also enabled. Be sure to secure the wait time for the sample-and-hold circuit when using the channel-dedicated disconnection detection assist function.
When the ADEXICR.OCSA bit is set to 1 to perform A/D conversion of the internal reference voltage, ADNDIS[4:0] are automatically fixed to 0Fh, and discharging is executed prior to A/D conversion (auto-discharging). An auto-discharge period of 15 ADCLK cycles is inserted before sampling each time the internal reference voltage is A/D-converted.

26.2.16 A/D Group Scan Priority Control Register (ADGSPCR)

Address(es): S12AD.ADGSPCR 0008 9080h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	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	Bit Name	Description	R/W
b0	PGS	Group Priority Control Setting*1	0: Operation is without group priority control 1: Operation is with group priority control	R/W
b1	GBRSCN	Low-Priority Group Restart Setting*2	(Enabled only when PGS = 1. Reserved when PGS = 0.) 0: Scanning for the group is not restarted after having been discontinued due to group priority control. 1: Scanning for the group is restarted after having been discontinued due to group priority control.	R/W
b13 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	LGRRS	Restart Channel Select	(Enabled when PGS = 1 and GBRSCN = 1. Reserved when PGS = 0 or GBRSCN = 0.) 0: Scanning is restarted from the scan start channel. 1: Scanning is restarted from the channel on which A/D conversion is not completed.	R/W
b15	GBRP	Single Scan Continuous Start*3	(Enabled only when PGS = 1. Reserved when PGS = 0.) 0: Single scan is not continuously activated. 1: Single scan for the lowest-priority group 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 is not guaranteed.
Note 2. When the GBRSCN bit is to be set to 1, the frequency ratio of peripheral module clock PCLK to A/D conversion clock ADCLK should be set to 1:1.
Note 3. When the GBRP bit has been set to 1, single scan is performed continuously for the lowest-priority group regardless of the setting of the GBRSCN bit.

ADGSPCR is used to discontinue scanning of the low-priority group and make settings for priority control of scanning for the priority group in group scan mode.
For the settings on group priority operation, see Table 26.11 and Table 26.12.

PGS Bit (Group Priority Control Setting)

This bit sets the priority of operation in group scan mode. Set this bit to 1 when giving priority to operation on the group. When the PGS bit is to be set to 1, the ADCSR.ADCS[1:0] bits must be set to 01b (group scan mode).
During group priority operation, a trigger to start scanning for the priority group is accepted during scan for the low-priority group, and scan for the priority group is started after scan for the low-priority group was discontinued. The priority order is group A > group B > group C.
When a trigger to start scanning for group B is accepted during scan for group C, group C scan is discontinued, and scan for group B is started. When a trigger to start scanning for group A is accepted during scan for group C, group C scan is discontinued, and scan for group A is started.
Likewise, when a trigger to start scanning for group A is accepted during scan for group B, group B scan is discontinued and scan for group A is started.
When setting the PGS bit to 0, clearing should be performed by software according to section 26.6.2, Notes on Stopping A/D Conversion. When setting the PGS bit to 1, follow the procedure described in section 26.3.4.3, Operation under Group Priority Control.

GBRSCN Bit (Low-Priority Group Restart Setting)

This bit controls the restarting of scan operation during group priority control.
If a scan operation on the low-priority group has been stopped by a priority group trigger input with the GBRSCN bit set to 1, the scan operation is restarted after the scanning of the priority group is completed. Also, if a low-priority trigger is input during scan for the priority group, the scan operation on the low-priority group is restarted after the scan for the priority group is completed.
If the GBRSCN bit has been set to 0, triggers that are input during A/D conversion are ignored. Also, the ADCSR.ADST bit must be 0 when the GBRSCN bit is to be set.
The setting of the GBRSCN bit is enabled when the PGS bit is set to 1.

LGRRS Bit (Restart Channel Select)

This bit sets the channel on which scan is restarted during group priority control. The setting of the LGRRS bit is enabled when the PGS and GBRSCN bits are set to 1.
If a scan operation on the low-priority group has been stopped due to group priority operation with the LGRRS bit set to 0, the scan operation is restarted from the start channel after the scan for the priority group is completed.
If a scan operation on the low-priority group has been stopped due to group priority operation with the LGRRS bit set to 1, the scan operation is restarted*1 on the channel on which A/D conversion is not competed after the scan for the priority group is completed.
The LGRRS bit should be set while the ADCSR.ADST bit is 0.

Note 1. If A/D conversion on the addition set channel is not completed for the set number of times when scanning is stopped, A/D conversion on the channel is restarted for the set number of times when scanning is restarted.

GBRP Bit (Single Scan Continuous Start)

This bit is set when the lowest-priority group is continuously operated in single scan mode while group priority operation is set. The lowest-priority group is group C when groups A, B, and C are used; group B when groups A and B are used.
Setting the GBRP bit to 1 starts a single scan on the lowest-priority group. On completion of the scan, another single scan on the lowest-priority group is automatically started.
If scanning has been stopped due to group priority operation, single scan on the lowest-priority group is automatically restarted on completion of the A/D conversion on the priority group.
Disable the trigger input for the lowest-priority group before setting the GBRP bit to 1. When the GBRP bit is set to 1, only the lowest-priority group is scanned again even if the GBRSCN bit is 0.
The ADCSR.ADST bit must be 0 when the GBRP bit is to be set.
The setting of the GBRP bit is enabled when the PGS bit is 1.

26.2.17 A/D Programmable Gain Amplifier Control Register (ADPGACR)

(1) S12AD.ADPGACR

Address(es): 0008 91A0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	P002ENAMP	P002SEL1	—	—	P001ENAMP	P001SEL1	—	—	P000ENAMP	P000SEL1	—
Value after reset:	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b1	P000SEL1	PGA P000 Amplifier Pass-Through Enable	0: Does not pass through the amplifier in the PGA. 1: Passes through the amplifier in the PGA.	R/W
b2	P000ENAMP	PGA P000 Amplifier Enable	0: Does not use the amplifier in the PGA. 1: Uses the amplifier in the PGA.	R/W
b4, b3	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b5	P001SEL1	PGA P001 Amplifier Pass-Through Enable	0: Does not pass through the amplifier in the PGA. 1: Passes through the amplifier in the PGA.	R/W
b6	P001ENAMP	PGA P001 Amplifier Enable	0: Does not use the amplifier in the PGA. 1: Users the amplifier in the PGA.	R/W
b8, b7	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b9	P002SEL1	PGA P002 Amplifier Pass-Through Enable	0: Does not pass through the amplifier in the PGA. 1: Passes through the amplifier in the PGA.	R/W
b10	P002ENAMP	PGA P002 Amplifier Enable	0: Does not use the amplifier in the PGA. 1: Users the amplifier in the PGA.	R/W
b12, b11	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b14, b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

The S12AD.ADPGACR register is used to set whether or not to use programmable gain amplifier P000 to P002 of unit 0.

P000SEL1 Bit (PGA P000 Amplifier Pass-Through Enable)

This bit is used to select whether or not to pass through the amplifier in programmable gain amplifier P000 of unit 0.

P000ENAMP Bit (PGA P000 Amplifier Enable)

This bit is used to select whether or not to use the amplifier in programmable gain amplifier P000 of unit 0.

P001SEL1 Bit (PGA P001 Amplifier Pass-Through Enable)

This bit is used to select whether or not to pass through the amplifier in programmable gain amplifier P001 of unit 0.

P001ENAMP Bit (PGA P001 Amplifier Enable)

This bit is used to select whether or not to use the amplifier in programmable gain amplifier P001 of unit 0.

P002SEL1 Bit (PGA P002 Amplifier Pass-Through Enable)

This bit is used to select whether or not to pass through the amplifier in programmable gain amplifier P002 of unit 0.

P002ENAMP Bit (PGA P002 Amplifier Enable)

This bit is used to select whether or not to use the amplifier in programmable gain amplifier P002 of unit 0.

26.2.18 A/D Programmable Gain Amplifier Gain Setting Register 0 (ADPGAGS0)

(1) S12AD.ADPGAGS0

Address(es): 0008 91A2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	P002GAIN[3:0]				P001GAIN[3:0]				P000GAIN[3:0]			
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	P000GAIN[3:0]	PGA P000 Gain Setting	The relationship between each setting and the gain is as follows: 0 0 0 0: × 2.000 0 0 0 1: × 2.500 0 1 0 0: × 3.077 1 0 0 1: × 5.000 1 1 0 0: × 8.000 1 1 0 1: × 10.000 Settings other than above are prohibited.	R/W
b7 to b4	P001GAIN[3:0]	PGA P001 Gain Setting		R/W
b11 to b8	P002GAIN[3:0]	PGA P002 Gain Setting		R/W
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The S12AD.ADPGAGS0 register is used to set the gain of programmable gain amplifier P000 to P002 of unit 0.

P000GAIN[3:0] Bits (PGA P000 Gain Setting)

These bits are used to set the gain of programmable gain amplifier P000 of unit 0.

P001GAIN[3:0] Bits (PGA P001 Gain Setting)

These bits are used to set the gain of programmable gain amplifier P001 of unit 0.

P002GAIN[3:0] Bits (PGA P002 Gain Setting)

These bits are used to set the gain of programmable gain amplifier P002 of unit 0.

26.3 Operation

26.3.1 Scanning Operation

In scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.
A scan conversion is performed in 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 by software. In group scan mode, the selected channels of groups A, B, and C are scanned once after starting to be scanned according to the respective synchronous trigger.
In single scan mode and continuous scan mode, A/D conversion is performed for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n. In group scan mode, A/D conversion is performed for ANn channels of groups A, B, and C selected by the ADANSA0 register, ADANSB0 register, and ADANSC0 register, respectively, starting from the channel with the smallest number n.
When self-diagnosis is selected, it is executed once at the beginning of each scan and one of the three voltages internally generated in the 12-bit A/D converter is converted.
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 the synchronous trigger selected by the ADSTRGR.TRSA[5:0] bits. In group scan mode, the double trigger function can be used only for group A.
Extended double trigger mode indicates a state when the following synchronous trigger (two synchronous trigger sources enabled) is selected by the TRSA[5:0] bits in the A/D conversion start trigger select register (ADSTRGR) in double trigger mode.

- TRG4AN or TRG4BN (the ADSTRGR.TRSA[5:0] bits are set to 001011b)

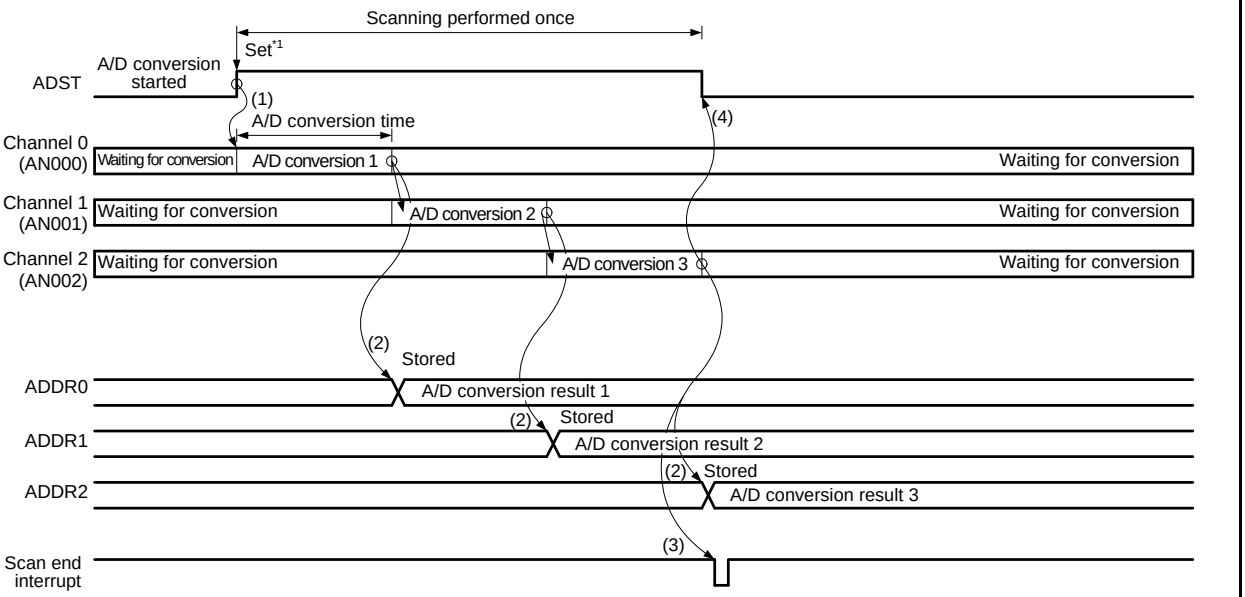
In extended double trigger mode, in addition to normal operations in double trigger mode, A/D conversion data is stored in A/D data duplication register A (ADDBLDRA) or A/D data duplication register B (ADDBLDRB) depending on the trigger type. If two types of triggers have occurred simultaneously in this mode, A/D conversion data is not sorted by the trigger sources and is stored in data duplication register B (ADDBLDRB).
Note that if a new trigger is input during A/D conversion caused by another trigger, the new trigger is ignored.
When any of AN000 to AN002 channels is set as a channel-dedicated sample-and-hold circuit by the S12AD.ADSHCR.SHANS[2:0] bits, the target analog input is sampled and held before the first A/D conversion of each scan.
The ADST0 output is used to output the S12AD.ADCSR.ADST bit state

26.3.2 Single Scan Mode

26.3.2.1 Basic Operation (Without Channel-Dedicated Sample-and-Hold 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 is set to 1 (A/D conversion start) by software, or synchronous or asynchronous trigger input, A/D conversion is performed for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(4) The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state.



Note 1. ↓ indicates the instruction is executed by software.

Figure 26.2 Example of Operation in Single Scan Mode (Basic Operation: AN000, AN001, AN002 Selected)

26.3.2.2 Basic Operation (With Channel-Dedicated Sample-and-Hold Circuits)

When a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, and this is followed by A/D conversion once of the analog inputs on all selected channels. The ADSHCR.SHANS[2:0] bits are used to select the channels for which the channel-dedicated sample-and-hold circuits are to be used.

(1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, or synchronous or asynchronous trigger input.
(2) After sample-and-hold operation, A/D conversion is performed for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(3) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).
(4) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(5) The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a waiting state.

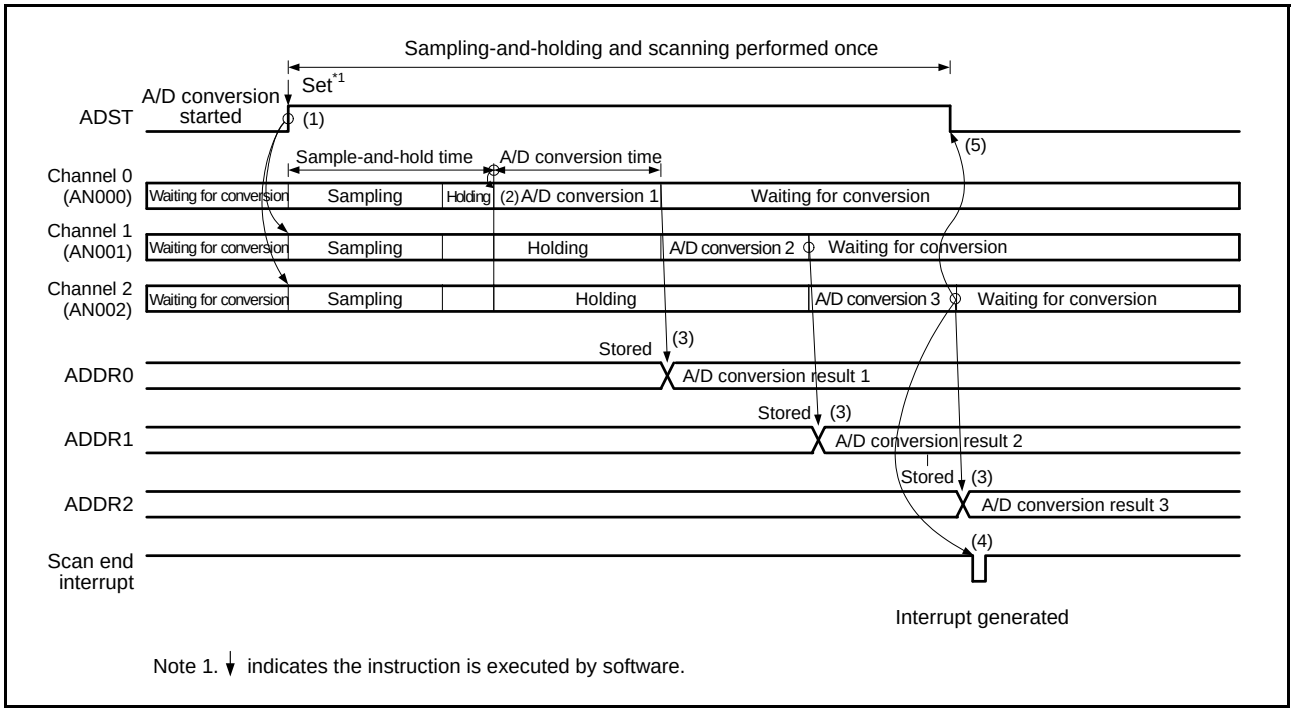


Figure 26.3 Example of Operation in Single Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used; AN000, AN001, AN002 Selected)

26.3.2.3 Channel Selection and Self-Diagnosis (Without Channel-Dedicated Sample-and-Hold Circuits)

When channels and self-diagnosis are selected, A/D conversion is performed once for the reference voltage supplied to the 12-bit A/D converter as below. After that, A/D conversion is performed only once on the analog input of the selected channels.

(1) A/D conversion for self-diagnosis is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, or synchronous or asynchronous trigger input.
(2) When A/D conversion for self-diagnosis is completed, A/D conversion result is stored into the A/D self-diagnosis data register (ADRD), and A/D conversion is performed for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(3) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(4) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(5) The ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state.

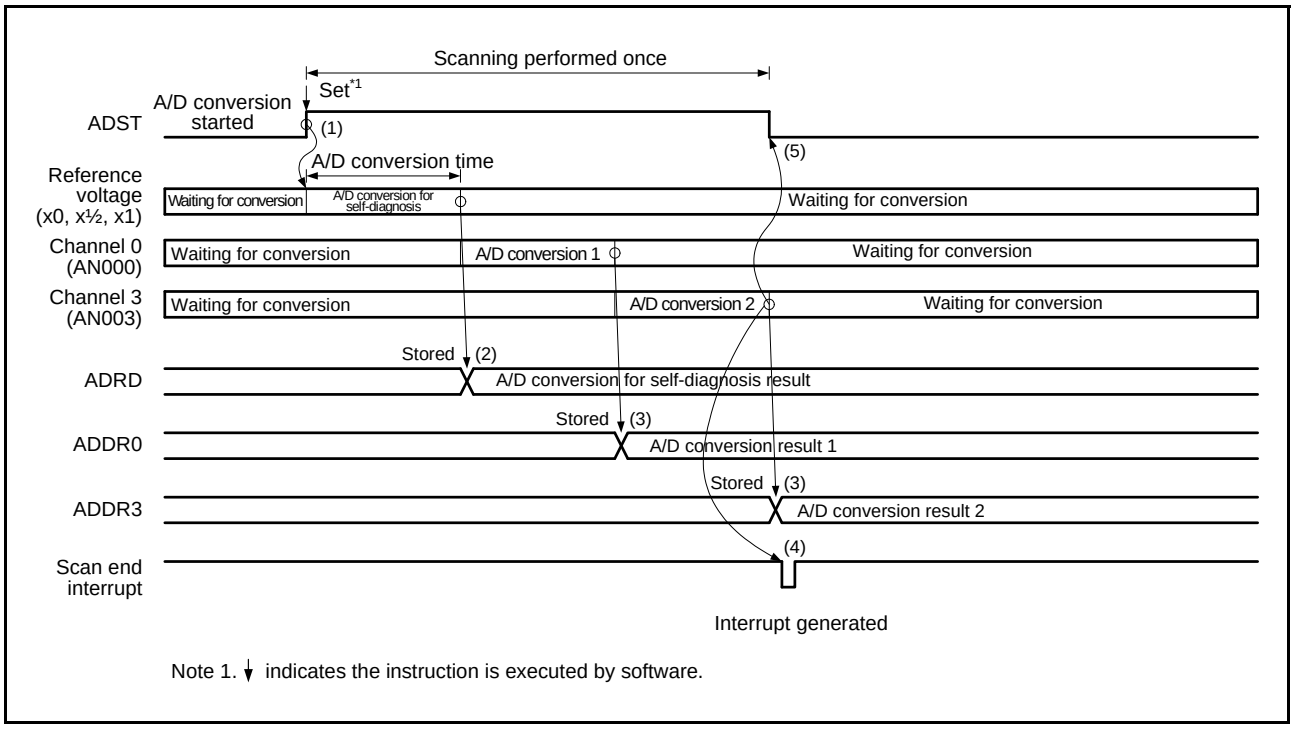


Figure 26.4 Example of Operation in Single Scan Mode (Basic Operation: AN000, AN003 Selected + Self-Diagnosis)

26.3.2.4 Channel Selection and Self-Diagnosis (With Channel-Dedicated Sample-and-Hold Circuits)

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, and A/D conversion is performed once for the reference voltage supplied to the 12-bit A/D converter as below. After that, A/D conversion is performed only once on the analog input of the selected channels.

(1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, or synchronous or asynchronous trigger input.
(2) A/D conversion for self-diagnosis is started after completion of sampling and holding.
(3) When A/D conversion for self-diagnosis is completed, A/D conversion result is stored into the A/D self-diagnosis data register (ADRD), and A/D conversion is performed for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(4) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(5) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt upon scanning completion enabled).
(6) The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed. Then the 12-bit A/D converter enters a wait state.

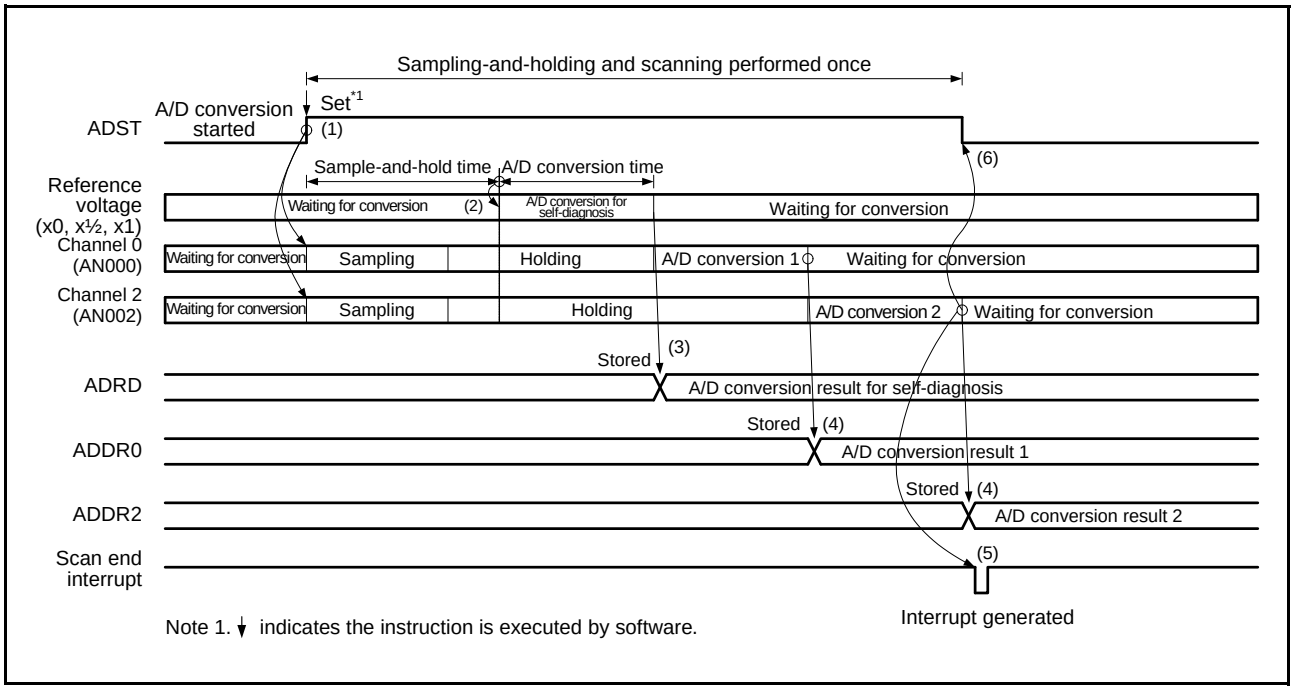


Figure 26.5 Example of Operation in Single Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used: AN000, AN002 Selected + Self-Diagnosis)

26.3.2.5 A/D Conversion of Internal Reference Voltage

A/D conversion of the internal reference voltage is performed in single scan mode as below.
All channels should be deselected (by setting the ADANSA0 register bits to all 0 and the ADCSR.DBLE bit to 0).

(1) Set the sampling time to 5 µs or longer.
(2) After switching to A/D conversion of the internal reference voltage, start A/D conversion by setting the ADST bit to 1.
(3) When A/D conversion is completed, the conversion result is stored into the A/D internal reference voltage data register (ADOCDR). If the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled), a scan end interrupt request is generated.
(4) The ADST bit remains 1 during A/D conversion, and is automatically cleared to 0 upon completion of A/D conversion. Then the 12-bit A/D converter enters a wait state.

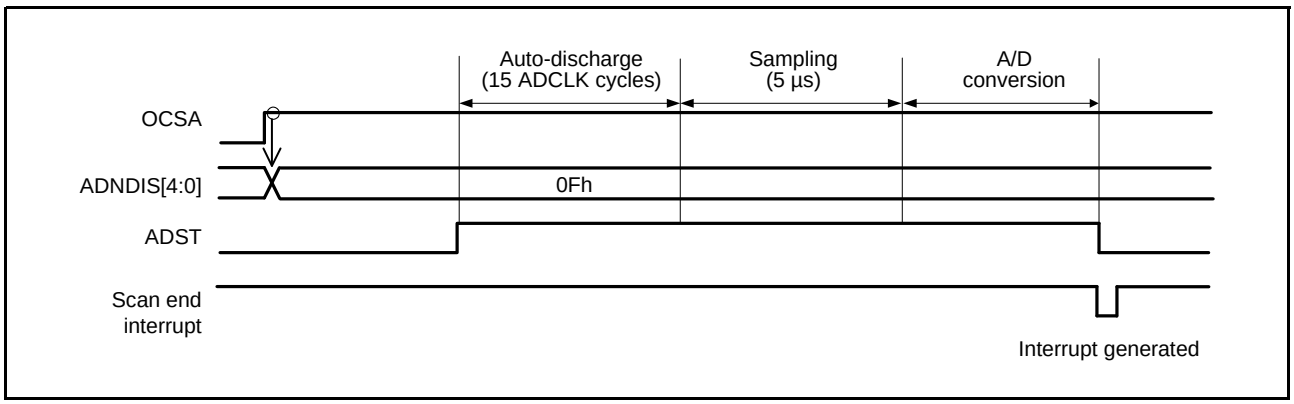


Figure 26.6 Example of Operation in Single Scan Mode (Internal Reference Voltage Selected)

26.3.2.6 A/D Conversion in Double Trigger Mode

In single scan mode with double trigger mode, single scan operation started by synchronous trigger is performed twice as below.
Self-diagnosis should be deselected, and the internal reference voltage A/D conversion select bit (S12AD.ADEXICR.OCSA) should be set to 0.
Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1. When the ADCSR.DBLE bit is set to 1, channel selection using the ADANSA0 register is invalid. In double trigger mode, synchronous triggers should be selected using the ADSTRGR.TRSA[5:0] bits, the ADCSR.EXTRG bit should be set to 0, and the ADCSR.TRGE bit should be set to 1. Software trigger should not be used.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by synchronous trigger input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.
(2) When A/D conversion is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) The ADCSR.ADST bit is automatically cleared to 0 and the 12-bit A/D converter enters a wait state. Here, a scan end interrupt request is not generated irrespective of the ADCSR.ADIE bit setting (interrupt generation upon scanning completion enabled).
(4) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by the second trigger input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.
(5) When A/D conversion is completed, the A/D conversion result is stored into the A/D data duplication register (ADDBLDR), which is exclusively used in double trigger mode.
(6) If the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled), a scan end interrupt request is generated.
(7) The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion is completed. Then the 12-bit A/D converter enters a wait state.

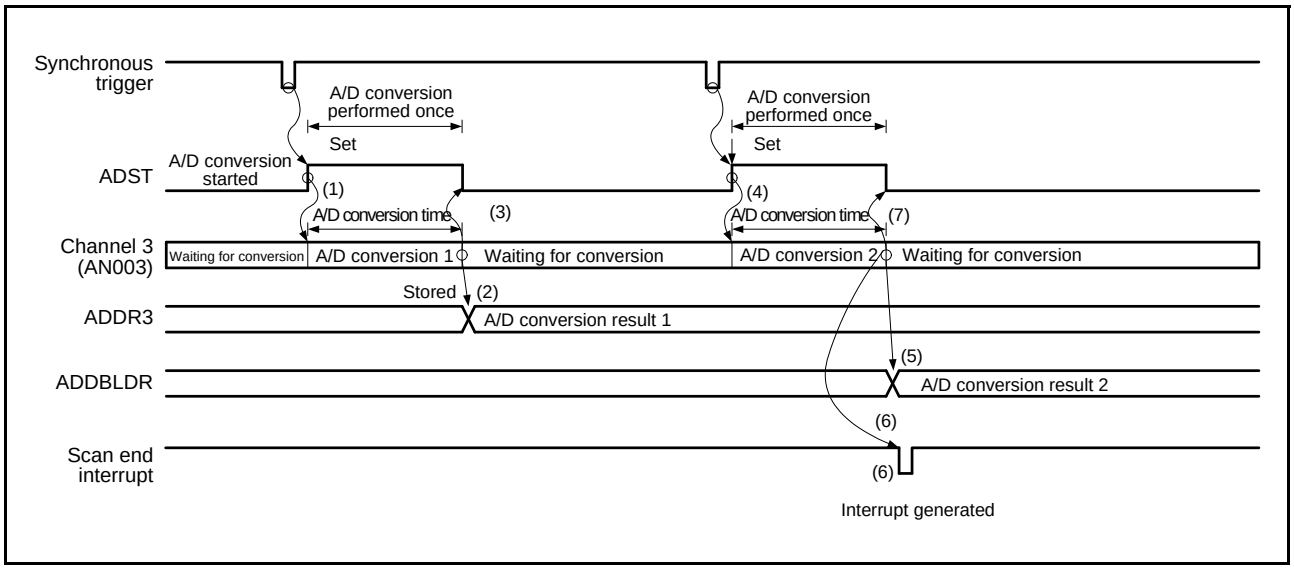


Figure 26.7 Example of Operation in Single Scan Mode (Double Trigger Mode Selected; AN003 Duplicated)

26.3.2.7 A/D Conversion in Extended Double Trigger Mode

When the double-trigger mode is selected in single-scanning mode, with TRG4AN or TRG4BN selected in the TRSA[5:0] bits of the A/D conversion start trigger select register (ADSTRGR), proceed with single scanning twice as follows.
Self-diagnosis should be deselected, and the internal reference voltage A/D conversion select bit (S12AD.ADEXICR.OCSA) should be set to 0.
Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1. When the ADCSR.DBLE bit is set to 1, channel selection using the ADANSA0 register is invalid. In extended double trigger mode, the ADCSR.EXTRG bit should be set to 0, and the ADCSR.TRGE bit should be set to 1. Software trigger should not be used.

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by TRG4AN input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.
(2) When A/D conversion is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy) and A/D data-duplication register A (ADDBLDRA).
(3) The ADCSR.ADST bit is automatically cleared to 0 and the 12-bit A/D converter enters a wait state. Here, a scan end interrupt request is not generated irrespective of the ADCSR.ADIE bit setting (interrupt generation upon scanning completion enabled).
(4) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by TRG4BN input, A/D conversion is started on the single channel selected by the ADCSR.DBLANS[4:0] bits.
(5) When A/D conversion is completed, the A/D conversion result is stored into A/D data duplication register (ADDBLDR) and A/D data duplication register B (ADDBLDRB).
(6) If the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled), a scan end interrupt request is generated.
(7) The ADCSR.ADST bit remains 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion is completed. Then the 12-bit A/D converter enters a wait state.

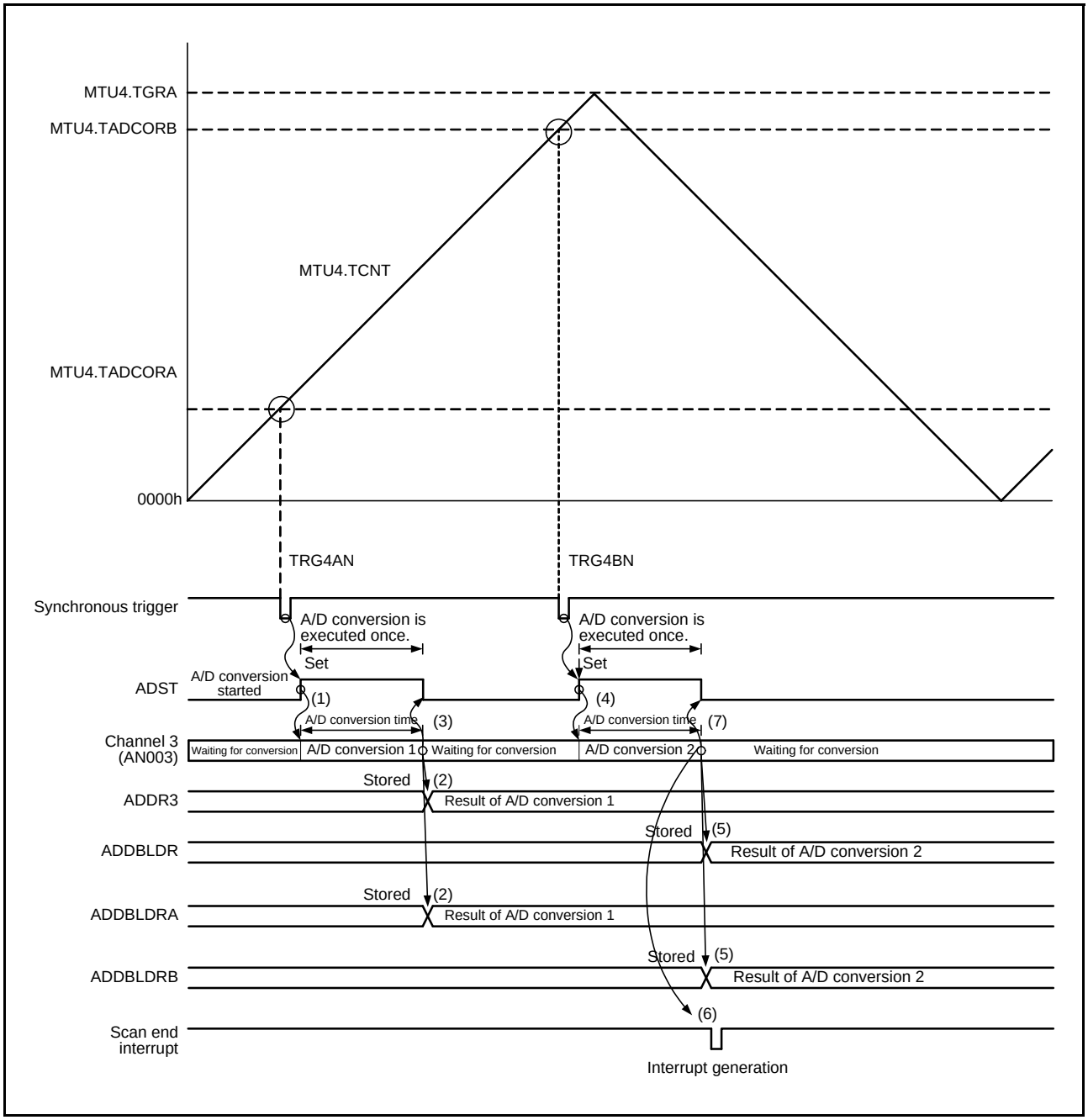


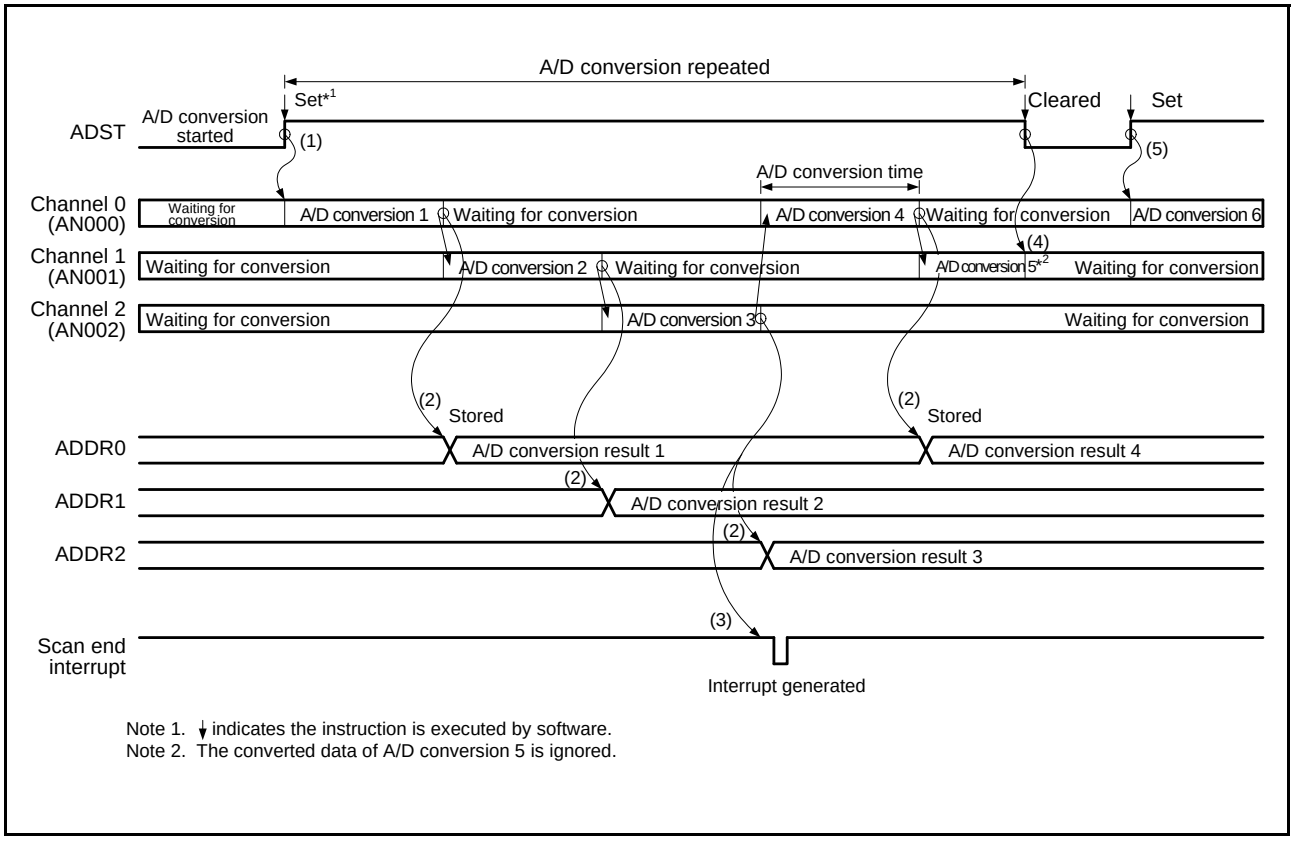
Figure 26.8 Example of Extended Operation in Double Trigger Mode (1)
(Duplication Selected for AN003, TRG4AN or TRG4BN Selected, First Trigger is TRG4AN)

26.3.3 Continuous Scan Mode

26.3.3.1 Basic Operation (Without Channel-Dedicated Sample and-Hold Circuits)

In basic operation of continuous scan mode, A/D conversion is performed repeatedly on the analog input of the specified channels as below.
In continuous scan mode, the internal reference voltage A/D conversion select bit (S12AD.ADEXICR.OCSA) should be set to 0 (deselected).

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, or synchronous or asynchronous trigger input, A/D conversion is performed for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(2) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(3) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
The 12-bit A/D converter sequentially starts A/D conversion for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(4) The ADCSR.ADST bit is not automatically cleared to 0 and steps 2 and 3 are repeated as long as the bit remains 1 (A/D conversion start). When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.
(5) When the ADST bit is later set to 1 (A/D conversion start), A/D conversion is started again for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.



Note 1. ↓ indicates the instruction is executed by software.
Note 2. The converted data of A/D conversion 5 is ignored.

Figure 26.9 Example of Operation in Continuous Scan Mode (Basic Operation: AN000 to AN002 Selected)

26.3.3.2 Basic Operation (With Channel-Dedicated Sample-and-Hold Circuits)

When a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, after which the analog inputs on all selected channels are A/D converted as below. The channels for which the channel-dedicated sample-and-hold circuits are to be used can be selected by the ADSHCR.SHANS[2:0] bits.
In continuous scan mode, the internal reference voltage A/D conversion select bit (S12AD.ADEXICR.OCSA) should be set to 0 (deselected).

(1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, synchronous trigger input, or asynchronous trigger input.
(2) After sample-and-hold operation, A/D conversion is performed for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(3) Each time A/D conversion of a single channel is completed, the result of A/D conversion is stored in the corresponding A/D data register (ADDRy).
(4) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled). At the same time, analog input sampling is started for all the channels for which the channel-dedicated sample-and-hold circuits are to be used.
(5) The ADCSR.ADST bit is not automatically cleared and steps 2 to 4 are repeated as long as the bit remains 1. When the ADCSR.ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.
(6) When the ADCSR.ADST bit is then set to 1 (A/D conversion start), analog input sampling is started again for all the channels for which the channel-dedicated sample-and-hold circuits are to be used.

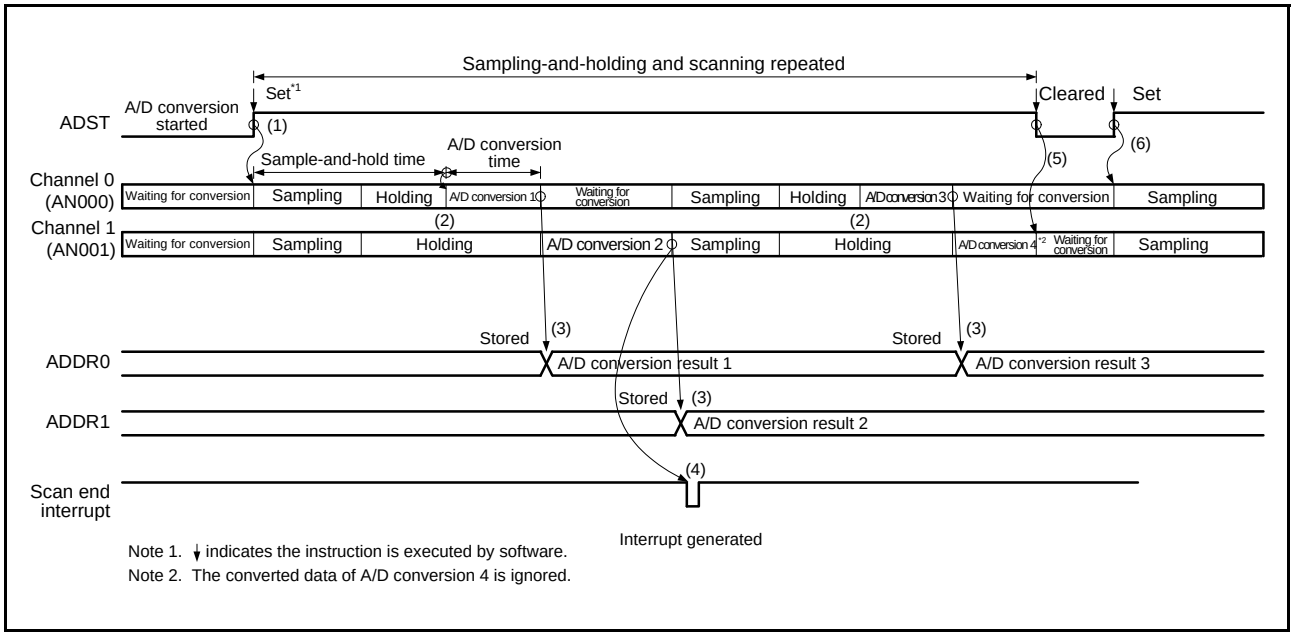
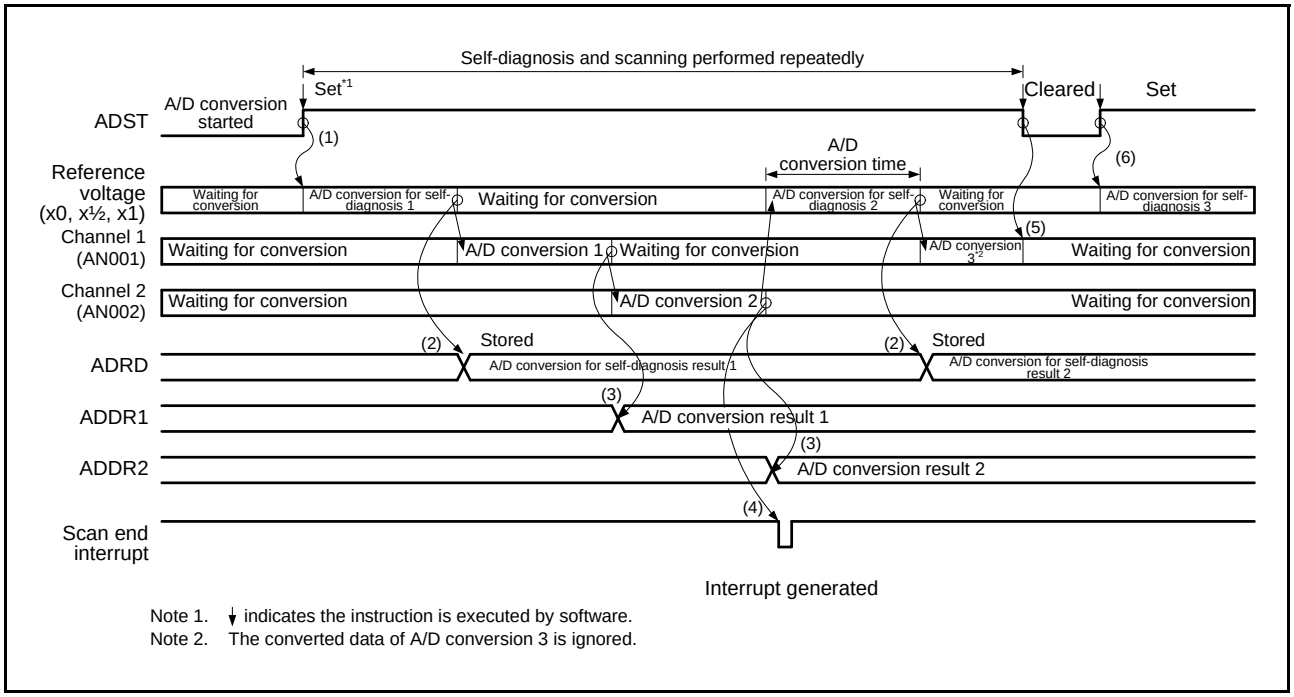


Figure 26.10 Example of Operation in Continuous Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used)

26.3.3.3 Channel Selection and Self-Diagnosis (Without Channel-Dedicated Sample-and-Hold Circuits)

When channels and self-diagnosis are selected at the same time, A/D conversion is first performed for the reference voltage supplied to the 12-bit A/D converter, and then A/D conversion is performed on the analog input of the selected channels, which sequence is repeated as below. In continuous scan mode, the internal reference voltage A/D conversion select bit (S12AD.ADEXICR.OCSA) should be set to 0 (deselected).

(1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, or synchronous or asynchronous trigger input, A/D conversion for self-diagnosis is started first.
(2) When A/D conversion for self-diagnosis is completed, the A/D conversion result is stored into the A/D self-diagnosis data register (ADRD). A/D conversion is then performed for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(3) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(4) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled). At the same time, the 12-bit A/D converter starts A/D conversion for self-diagnosis and then starts A/D conversion on ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(5) The ADCSR.ADST bit is not automatically cleared and steps 2 to 4 are repeated as long as the bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.
(6) When the ADCSR.ADST bit is later set to 1 (A/D conversion start), the A/D conversion for self-diagnosis is started again.



Note 1. ↓ indicates the instruction is executed by software.
Note 2. The converted data of A/D conversion 3 is ignored.

Figure 26.11 Example of Operation in Continuous Scan Mode (Basic Operation; AN001 and AN002 Selected + Self-Diagnosis)

26.3.3.4 Channel Selection and Self-Diagnosis (With Channel-Dedicated Sample-and-Hold Circuits)

When channels and self-diagnosis are selected and a channel-dedicated sample-and-hold circuit is used, sample-and-hold operations are performed first, and A/D conversion is performed for the reference voltage supplied to the 12-bit A/D converter as below. After that, A/D conversion is performed repeatedly on the analog input of the selected channels.In continuous scan mode, the internal reference voltage A/D conversion select bit (S12AD.ADEXICR.OCSA) should be set to 0 (deselected).

(1) Analog input sampling of all channels for which the channel-dedicated sample-and-hold circuits are to be used is started when the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, or synchronous or asynchronous trigger input.
(2) A/D conversion for self-diagnosis is started after completion of sampling and holding.
(3) When A/D conversion for self-diagnosis is completed, A/D conversion result is stored into the A/D self-diagnosis data register (ADRD), and A/D conversion is performed for ANn channels selected by the ADANSA0 register, starting from the channel with the smallest number n.
(4) Each time A/D conversion of a single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy).
(5) When A/D conversion of all the selected channels is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt upon scanning completion enabled). At the same time, analog input sampling is started for all the channels for which the channel-dedicated sample-and-hold circuits are to be used.
(6) The ADCSR.ADST bit is not automatically cleared and steps 2 to 5 are repeated as long as the ADCSR.ADST bit remains 1. When the ADST bit is set to 0 (A/D conversion stop), A/D conversion stops and the 12-bit A/D converter enters a wait state.
(7) When the ADST bit is later set to 1 (A/D conversion start), the A/D conversion for self-diagnosis is started again.

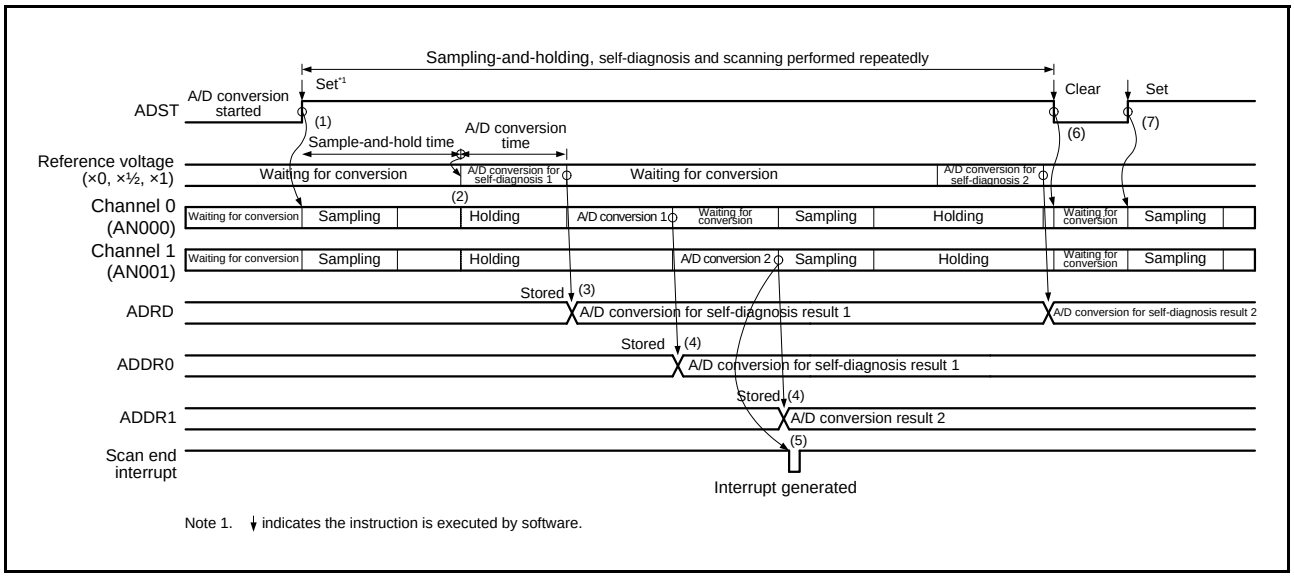


Figure 26.12 Example of Operation in Continuous Scan Mode (Channel-Dedicated Sample-and-Hold Circuits Used: AN000 and AN001 Selected + Self-Diagnosis)

26.3.4 Group Scan Mode

26.3.4.1 Basic Operation

Either two (groups A and B) or three (groups A, B, and C) can be selected as the number of the groups to be used in group scan mode.
In basic operation of group scan mode, A/D conversion is performed once on the analog inputs of all the specified channels in groups A and B, or groups A, B, and C after scan is started by a synchronous trigger as below. Scan operation of each group is similar to the scan operation in single scan mode.
The synchronous triggers of groups A and B, or groups A, B, and C can be selected using the TRSA[5:0], TRSB[5:0], and TRSC[5:0] bits in ADSTRGR, respectively. Different triggers should be used for each group A, B, and C so that scanning of groups A, B, and C does not occur simultaneously. Software trigger should not be used.
The channels to be scanned are selected using register ADANSA0 for group A, register ADANSB0 for group B, and register ADANSC0 for group C.
In group scan mode, the internal reference voltage A/D conversion select bit (S12AD.ADEXICR.OCSA) should be set to 0 (deselected).
When self-diagnosis is selected in group scan mode, self-diagnosis is separately executed for groups A and B, or groups A, B, and C.
The following describes operation in group scan mode using a trigger from the MTU. The TRG4AN, TRG4BN, and TRG4ABN triggers from the MTU are assumed to be used to start conversion of groups A, B and C, respectively.

(1) Scanning of group A is started by the TRG4AN trigger from the MTU.
(2) When group A scanning is completed, a scan end interrupt is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(3) Scanning of group B is started by the TRG4BN trigger from the MTU.
(4) When group B scanning is completed, a group B scan end interrupt is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group B scan completion enabled).
(5) Scanning of group C is started by the TRG4ABN trigger from the MTU.
(6) When group C scanning is completed, a group C scan end interrupt is generated if the ADGCTRGR.GCADIE bit is 1 (interrupt generation upon group C scan completion enabled).

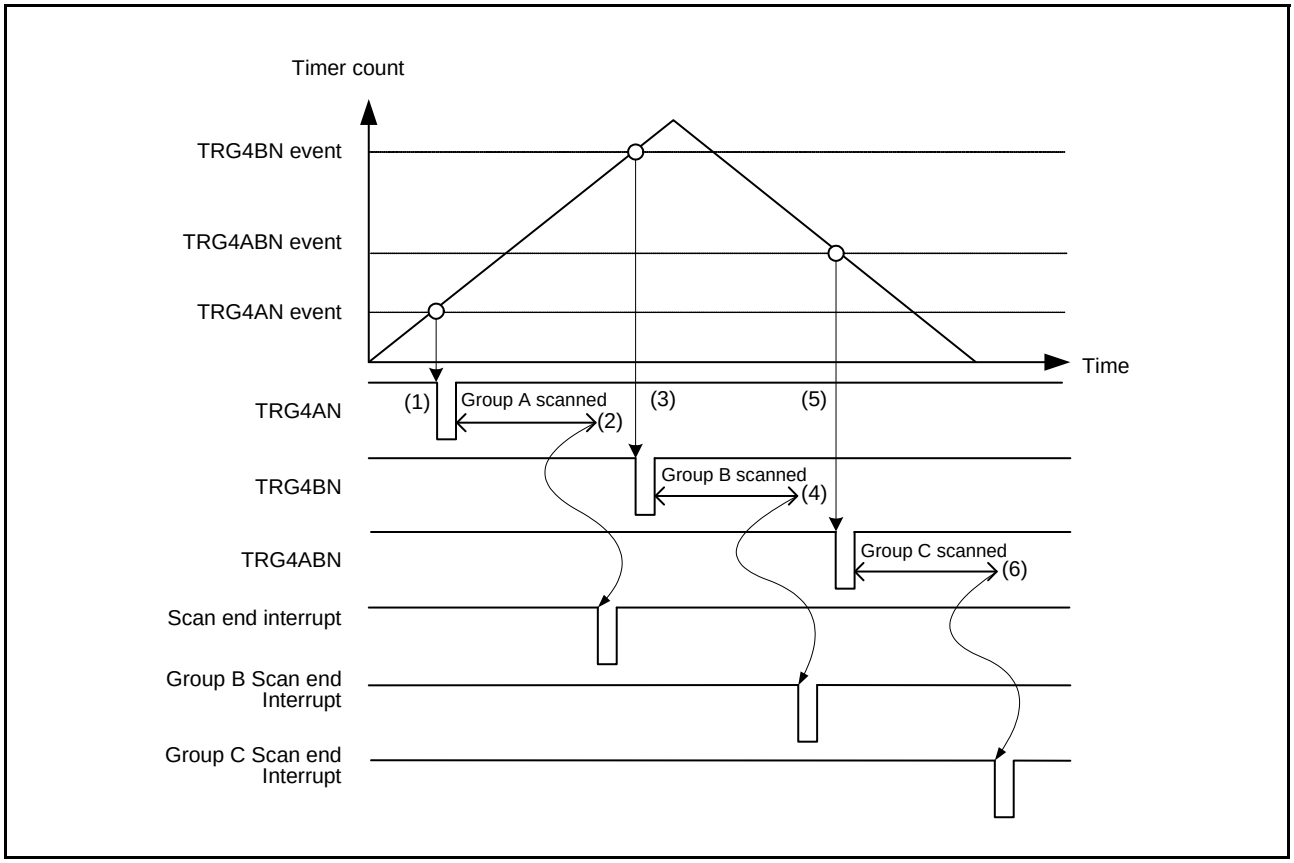


Figure 26.13 Example of Operation in Group Scan Mode (Basic Operation: Synchronous Triggers from MTU Used)

26.3.4.2 A/D Conversion in Double Trigger Mode

When double trigger mode is selected in group scan mode, two rounds of single scan operation started by a synchronous trigger are performed as a sequence for group A. For groups B and C, single scan operation started by a synchronous trigger is performed once.

In group scan mode, the synchronous triggers of groups A and B, or groups A, B, and C can be selected using the TRSA[5:0], TRSB[5:0], and TRSC[5:0] bits in ADSTRGR, respectively. Different triggers should be used for each group A, B, and C so that scanning of groups A, B, and C does not occur simultaneously. Software trigger and asynchronous trigger should not be used. When the TRG4AN or TRG4BN is selected in the ADSTRGR.TRSA[5:0] bits as the synchronous trigger of the group A, operation is in the extended double trigger mode.

The channels to be scanned are selected using bits ADCSR.DBLANS[4:0] for group A, register ADANSB0 for group B, and register ADANSC0 for group C.

In group scan mode, the internal reference voltage A/D conversion select bit (S12AD.ADEXICR.OCSA) should be set to 0 (deselected).

When double trigger mode is selected in group scan mode, self-diagnosis cannot be selected.

Duplication of A/D conversion data is enabled by setting the channel numbers to be duplicated to the ADCSR.DBLANS[4:0] bits and setting the ADCSR.DBLE bit to 1.

The following describes operation in group scan mode with double trigger mode using a synchronous trigger from the MTU. The TRG4ABN, TRGA0N, and TRGA1N triggers from the MTU are assumed to be used to start conversion of groups A, B, and C, respectively.

(1) Scanning of group C is started by the TRGA1N trigger from the MTU.
(2) When group C scanning is completed, a group C scan interrupt is generated if the ADGCTRGR.GCADIE bit is 1 (interrupt generation upon group C scan completion enabled).
(3) Scanning of group B is started by the TRGA0N trigger from the MTU.
(4) When group B scanning is completed, a group B scan end interrupt is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group B scan enabled).
(5) The first scanning of group A is started by the first TRG4ABN trigger from the MTU.
(6) When the first scanning of group A is completed, the conversion result is stored into the corresponding A/D data register (ADDRy); a scan end interrupt request is not generated irrespective of the ADIE bit setting in ADCSR.
(7) The second scanning of group A is started by the second TRG4ABN trigger from the MTU.
(8) When the second scanning of group A is completed, the conversion result is stored into ADDBLDR. a scan end interrupt is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(9) The second scanning of group B is started by the second TRGA0N trigger from the MTU.
(10) When the second scanning of group B is completed, a group B scan interrupt is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group B scan completion enabled).
(11) The second scanning of group C is started by the second TRGA1N trigger from the MTU.
(12) When the second scanning of group C is completed, a group C scan interrupt is generated if the ADGCTRGR.GCADIE bit is 1 (interrupt generation upon group C scan completion enabled).

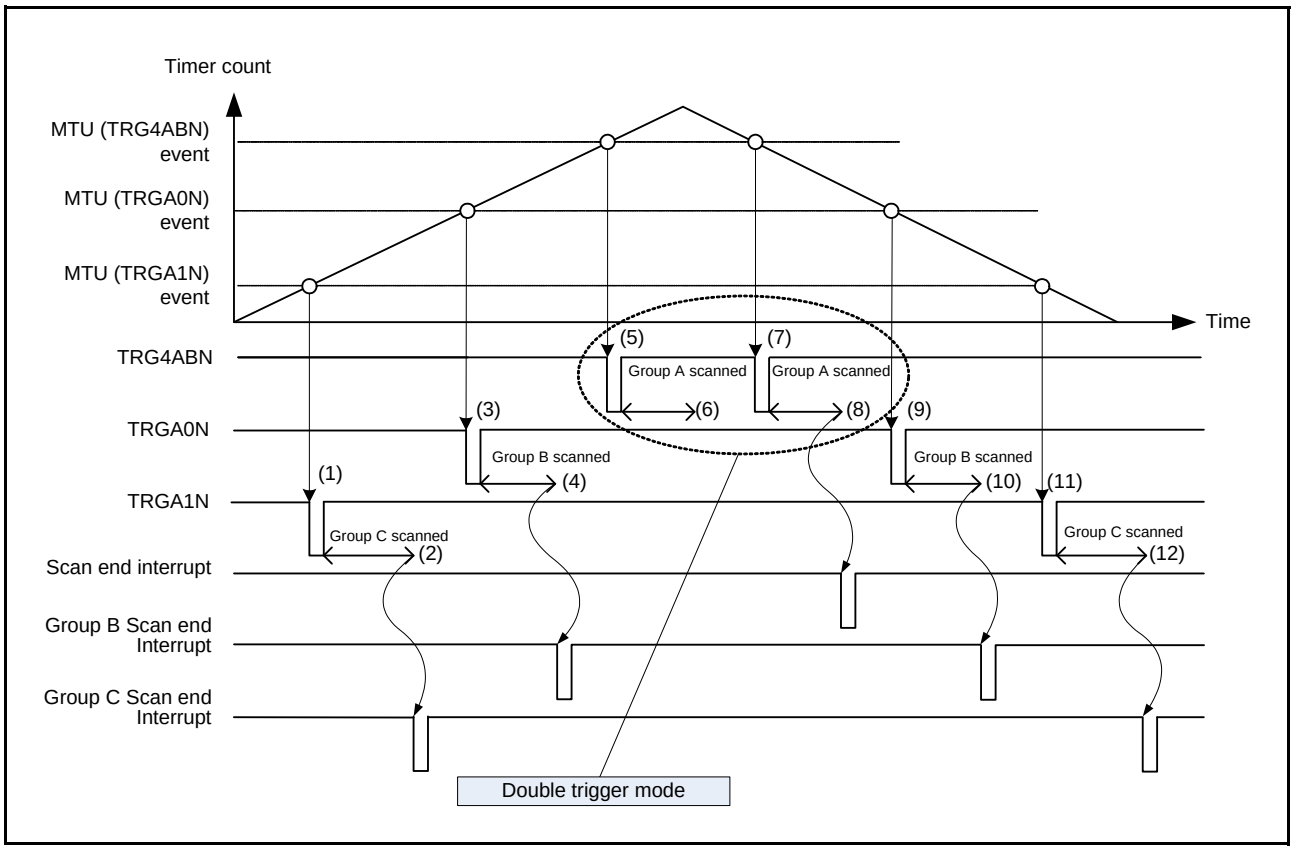


Figure 26.14 Example of Operation in Group Scan Mode with Double Trigger Mode (Basic Operation: Synchronous Triggers from MTU Used)

26.3.4.3 Operation under Group Priority Control

Setting the ADGSPCR.PGS bit to 1 in group scan mode makes operation proceed under group priority control. The group priority order is group A > group B > group C. Either two (groups A and B) or three (groups A, B, and C) can be selected as the number of the groups to be used in group scan mode. When setting the ADGSPCR.PGS bit to 1, follow the procedure described in Figure 26.15. If the procedure is not followed, proper scanning operation and data storage are not guaranteed.

In basic operation of group scan mode, if group A, B, or C is scanning, all other trigger inputs are ignored. Under group priority control, if a priority group trigger is input during A/D conversion for the low-priority group, A/D conversion for the low-priority group is discontinued and A/D conversion for the priority group proceeds.

If the ADGSPCR.GBRSCN bit is 0, the converter enters a wait state for the low-priority group on completion of the A/D conversion for the priority group.

The trigger input for the low-priority group during scanning is ignored.

If the ADGSPCR.GBRSCN bit is 1, the converter automatically restarts scanning for the low-priority group from the head of the group after A/D conversion for the priority group.

Also, the trigger input for the low-priority group generated during scanning for the priority group is enabled, and the converter automatically restarts scanning for the low-priority group after scanning for the priority group.

When the ADGSPCR.LGRRS bit is 0 while the ADGSPCR.GBRSCN bit is 1, the converter restarts scanning for the low-priority group from the head of the group. When the ADGSPCR.LGRRS bit is 1, the converter restarts scanning for the low-priority group from the channel on which scanning is discontinued.

However, when self-diagnosis is used, the converter restarts scanning from the channel on which scanning is discontinued after self-diagnosis is completed.

Table 26.10 summarizes operations in response to the input of a trigger during scanning with the settings of the ADGSPCR.GBRSCN bit.

When the ADGSPCR.GBRP bit is set to 1, scan operations in the lowest-priority group are continuously performed.

For the trigger settings in group scan mode, select a synchronous trigger for group A using the ADSTRGR.TRSA[5:0] bits, select a synchronous trigger different from that of group A for group B using the ADSTRGR.TRSB[5:0] bits, and select a synchronous trigger different from those of groups A and B for group C using the ADGCTRGR.TRSC[5:0] bits.

Set the ADSTRGR.TRSB[5:0] bits to 3Fh when setting group scan mode for two groups (the ADGCTRGR.GRCE bit to 0) and setting the ADGSPCR.GBRP bit to 1.

Set the ADGCTRGR.TRSC[5:0] bits to 3Fh when setting group scan mode for three groups (the ADGCTRGR.GRCE bit to 1) and setting the ADGSPCR.GBRP bit to 1.

Furthermore, as targets for scanning, select channels for group A using the ADANSA0 register, select channels for group B using the ADANSB0 register, and select channels for group C using the ADANSC0 register.

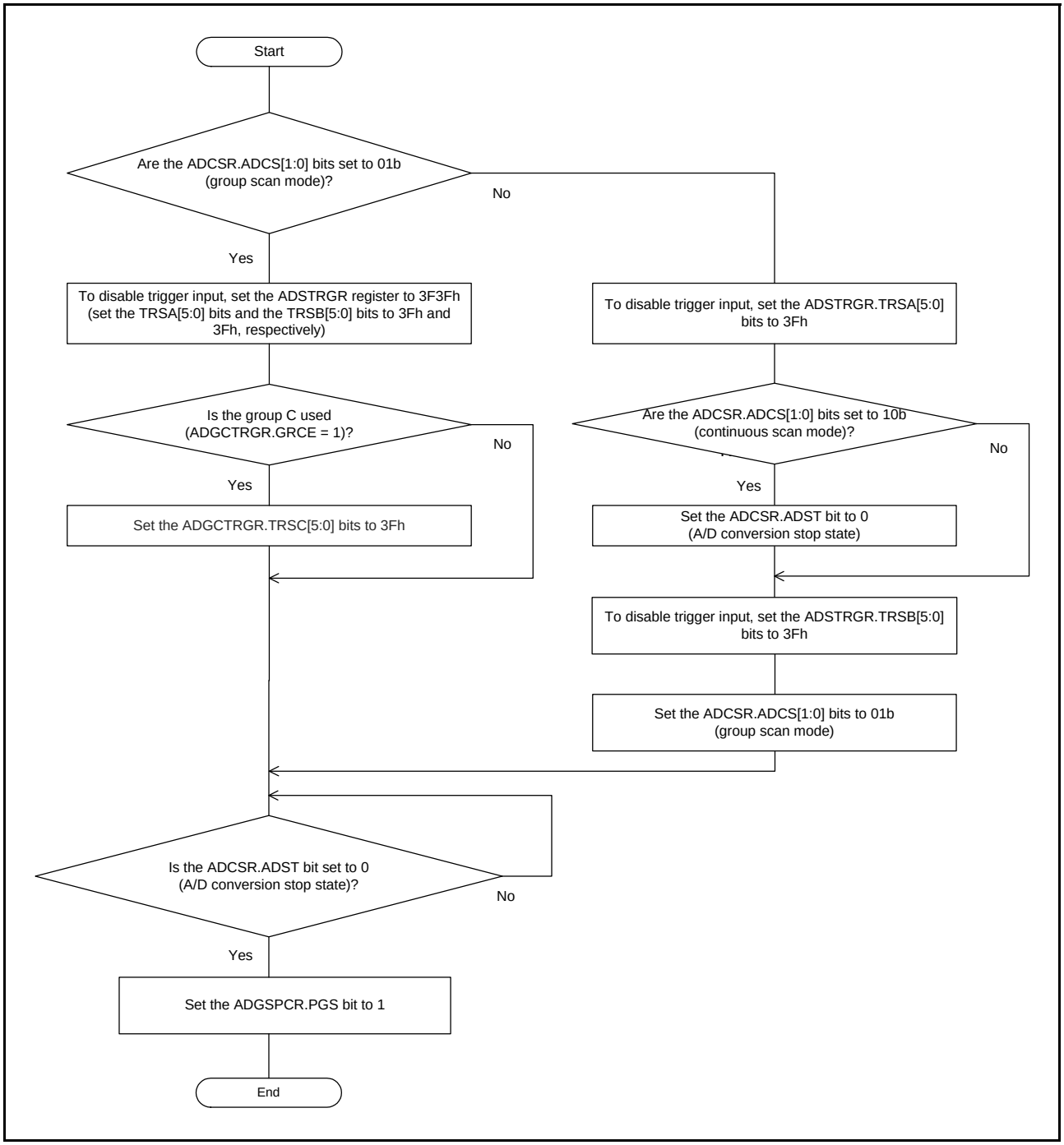


Figure 26.15 Flow of Setting the ADGSPCR.PGS Bit

Table 26.10 Control of Scanning Operations According to the Settings of the ADGSPCR.GBRSCN Bit

Scanning 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.
When A/D conversion for group B is in progress	Input of trigger for group A	A/D conversion in progress for group B discontinued and conversion for group A starts.	• A/D conversion in progress for group B is discontinued and conversion for group A starts. • A/D 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	A/D conversion in progress for group C discontinued and conversion for group A starts.	• A/D conversion in progress for group C is discontinued and conversion for group A starts. • A/D conversion for group C starts after conversion for group A is completed.
	Input of trigger for group B	A/D conversion in progress for group C is discontinued and conversion for group B starts.	• A/D conversion in progress for group C is discontinued and conversion for group B starts. • A/D 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.

When using group priority operation mode, refer to the following tables to select the desirable operating mode and set the registers.

Table 26.11 Group Priority Operation Setting and Operating Mode for Two Groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 0)

ADGSPCR			
GBRSCN	LGRRS	GBRP	Operation
0	x	0	Group priority operation for two groups (groups A and B) • When a group A trigger is input, group B scan is completed (not restarted)
1	0	0	Group priority operation for two groups (groups A and B) • After group B scan is discontinued, group B scan is restarted from the head of the channel selected with the ADANSB0 register after group A scan is completed.
1	1	0	Group priority operation for two groups (groups A and B) • After group B scan is discontinued, group B scan is restarted from the channel on which scan was discontinued*1, among the channels selected with the ADANSB0 register after group A scan is completed.
x	0	1	Group priority operation for two groups (groups A and B) • Single scan for group B is started continuously without start trigger input. After group B scan is discontinued, single scan is restarted from the head of the channel selected with the ADANSB0 register after scan for group A is completed.
1	1	1	Group priority operation for two groups (groups A and B) • Single scan for group B is started continuously without start trigger input. After group B scan was discontinued, single scan is restarted from the channel on which scan was discontinued*1, among the channels selected with the ADANSB0 register after group A scan is completed.

x = Don't care

Note 1. When self-diagnosis is used (ADCER.DIAGM = 1), A/D conversion of the discontinued channel is started after self-diagnosis.

Table 26.12 Group Priority Operation Setting and Operating Mode for Three Groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 1)

ADGSPCR			
GBRSCN	**LGRRS**	**GBRP**	**Operation**
0	x	0	Group priority operation for three groups (groups A, B, and C) • When a group A trigger is input, group B scan is completed (not restarted) • When a trigger for group A or B is input, group C scan is not completed (not restarted).
0	x	1	Group priority operation for three groups (groups A, B, and C) • When a group A trigger is input, group B scan is completed (not restarted) • Single scan for group C is started continuously without start trigger input. After group C scan is discontinued, scan is restarted from the head of the channel selected with the ADANSC0 register after scan for groups A and B is completed.
1	0	0	Group priority operation for three groups (groups A, B, and C) • After group B scan is discontinued, scan is restarted from the head of the channel selected with the ADANSB0 register after group A scan is completed. • After group C scan is discontinued, scan is restarted from the head of the channel selected with the ADANSC0 register after scan for groups A and B is completed.
1	1	0	Group priority operation for three groups (groups A, B, and C) • After group B scan is discontinued, scan is restarted from the head of the channel selected with the ADANSB0 register after group A scan is completed. • After group C scan is discontinued, scan is restarted from the channel on which scan was discontinued*1, among the channels selected with the ADANSC0 register after scan for groups A and B is completed.
1	0	1	Group priority operation for three groups (groups A, B, and C) • After group B scan is discontinued, scan is restarted from the head of the channel selected with the ADANSB0 register after group A scan is completed. • Single scan for group C is started continuously without start trigger input. After group C scan is discontinued, single scan is restarted from the head of the channel selected with the ADANSC0 register after scan for groups A and B is completed.
1	1	1	Group priority operation for three groups (groups A, B, and C) • After scanning for group B is discontinued, once scanning for group A completes, scanning is resumed starting from the channel on which scanning was discontinued*1, among those channels selected with the ADANSB0 register. • Single scan for group C is started continuously without start trigger input. After group C scan is discontinued, single scan is restarted from the channel on which scan was discontinued*1, among the channels selected with the ADANSC0 register after scan for groups A and B is completed.

x = Don't care

Note 1. When self-diagnosis is used (ADCER.DIAGM = 1), A/D conversion of the discontinued channel is started after self-diagnosis.

(1) Group Priority Operation for Two Groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 0)

The following examples 1 to 5 show group priority operation in group scan mode (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0) when channel 0 is selected for group A and channels 1 to 3 are selected for group B.

Operation example 1: Group A trigger input during group B scan, with rescan setting

(1) When a group B trigger input sets the ADCSR.ADST bit to 1 (A/D conversion start), scan for the ANn channels selected in the ADANSB0 register, starts from the channel with the smallest number n.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(3) If a group A trigger is input during scan for group B, group B scan is discontinued while the ADCSR.ADST bit remains 1, and scan for the ANn channels of group A selected in the ADANSA0 register, starts from the channel with the smallest number n.
If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADDRy).
(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(5) A scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(6) If the ADGSPCR.GBRSCN bit is 1 (scan for the group is restarted after having been discontinued due to group priority operation), scan for the ANn channels of group B selected in the ADANSB0 register, restarts from the channel with the smallest number n while the ADCSR.ADST bit remains 1.
(7) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(8) A group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group B scan completion enabled).
(9) The ADCSR.ADST bit is automatically cleared to 0 when scan of all selected channels is completed, and the 12-bit A/D converter enters a wait state.

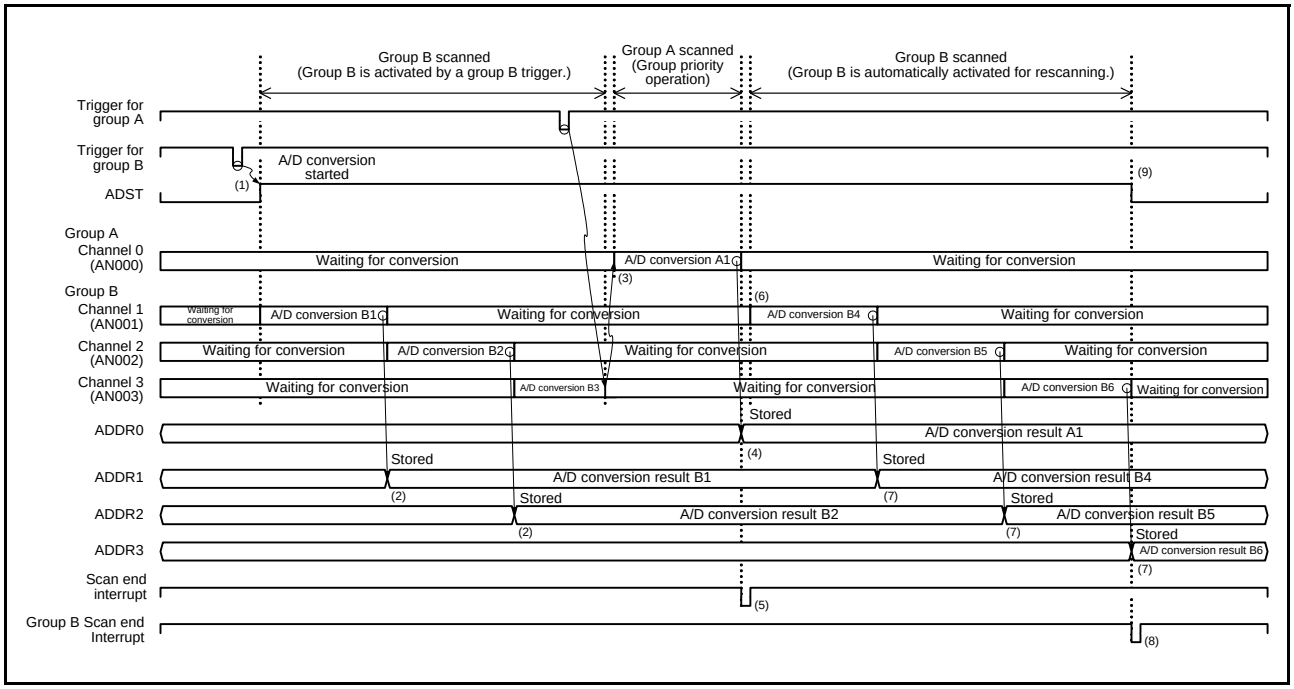


Figure 26.16 Example 1 of Group Priority Operation: Group A Trigger Input during Group B Scan, With Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0)

Operation example 2: Group A trigger input during group B rescan, with rescan setting

Figure 26.17 shows an example when a group A trigger is input during rescan operation on group B.

If a group A trigger is input, scan for group A starts even while rescan operation is in progress. Scan for group B starts after scan for group A is completed.

Operations of the ADCSR.ADST bit, storing to the A/D conversion result in the A/D data register (ADDRy), and interrupt requests are the same as example 1.

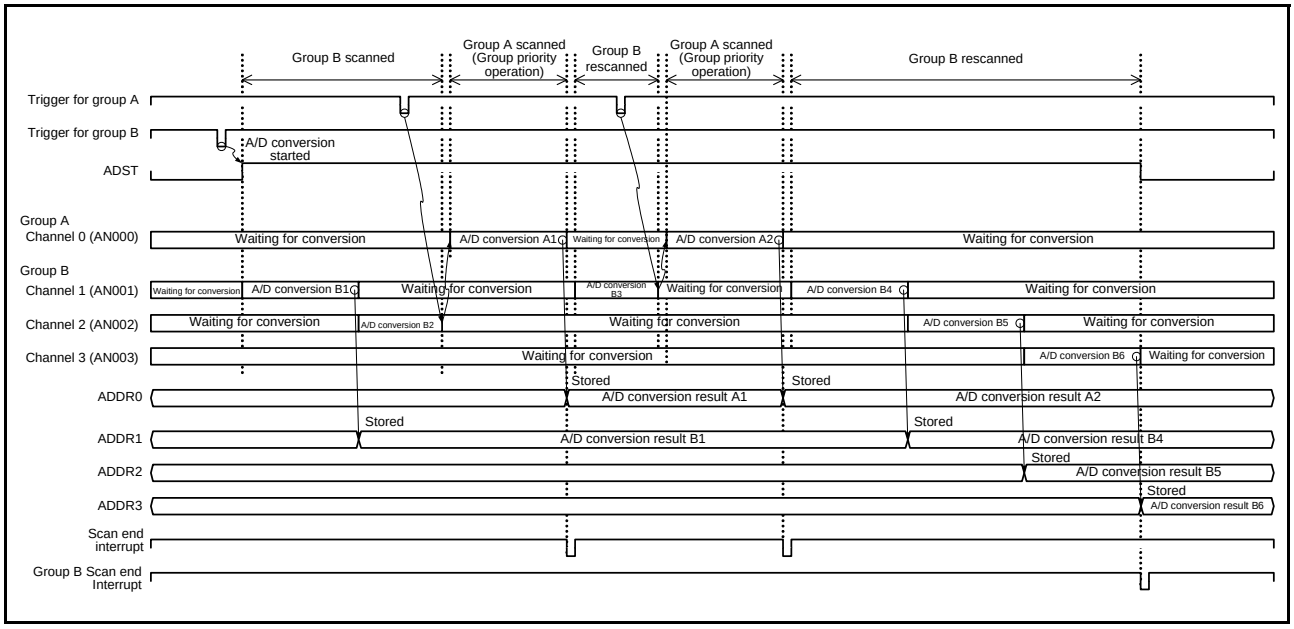


Figure 26.17 Example 2 of Group Priority Operation: Group A Trigger Input during Group B Rescan, With Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0)

Operation example 3: Group B trigger input during group A scan, with rescan setting

The following describes an example when a group B trigger is input during scan operation on group A when the ADGPSCR.GBRSCN bit is 1 (scan for the group is restarted after having been discontinued due to group priority operation).
If the ADGPSCR.GBRSCN bit is 0, all group B triggers that are input during scan operation on group A are disabled.

(1) When a group A trigger input sets the ADCSR.ADST bit to 1 (A/D conversion start), scan for the ANn channels of group A selected in the ADANSA0 register, starts from the channel with the smallest number n.
(2) If a group B trigger is input during scan for group A, scan for group B can be started.
(3) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(4) After scan for group A is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(5) After scan for group A is completed, scan for the ANn channels of group B selected in the ADANSB0 register, starts from the channel with the smallest number n while the ADCSR.ADST bit remains 1. When a group A trigger is input during scan for group B, group A scan starts as in example 1, and group B scan starts after group A scan is completed.
(6) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(7) After scan for group B is completed, a group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group B scan completion enabled).
(8) The ADCSR.ADST bit is automatically cleared to 0 when scan of all selected channels is completed, and the 12-bit A/D converter enters a wait state.

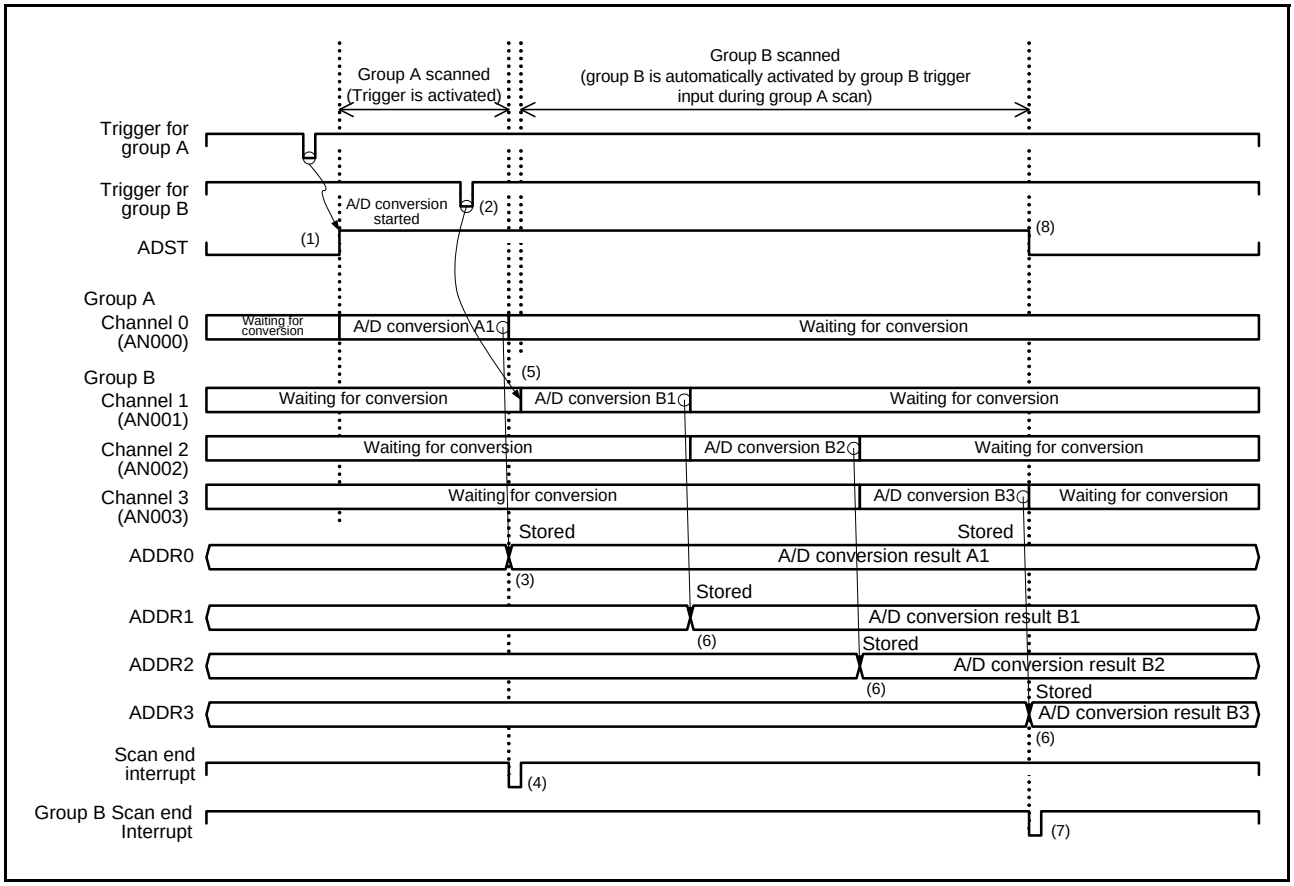


Figure 26.18 Example 3 of Group Priority Operation: Group B Trigger Input during Group A Scan, With Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0)

Operation example 4 shows group priority operation in group scan mode (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0) when channel 0 is selected for group A and channels 1 to 3 are selected for group B.

Operation example 4: Group A trigger input during group B scan, without rescan setting

(1) When a group B trigger input sets the ADCSR.ADST bit to 1 (A/D conversion start), scan for the ANn channels of group B selected in the ADANSB0 register, starts from the channel with the smallest number n.

(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).

(3) If a group A trigger is input during scan for group B, group B scan is discontinued while the ADCSR.ADST bit remains 1, and scan for the ANn channels of group A selected in the ADANSA0 register, starts from the channel with the smallest number n.
If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADDRy).

(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).

(5) After scan for group A is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).

(6) The ADST bit is automatically cleared to 0 when scan for group A is completed, and the 12-bit A/D converter enters a wait state. Scan for group B is not started until the next group B trigger is input.

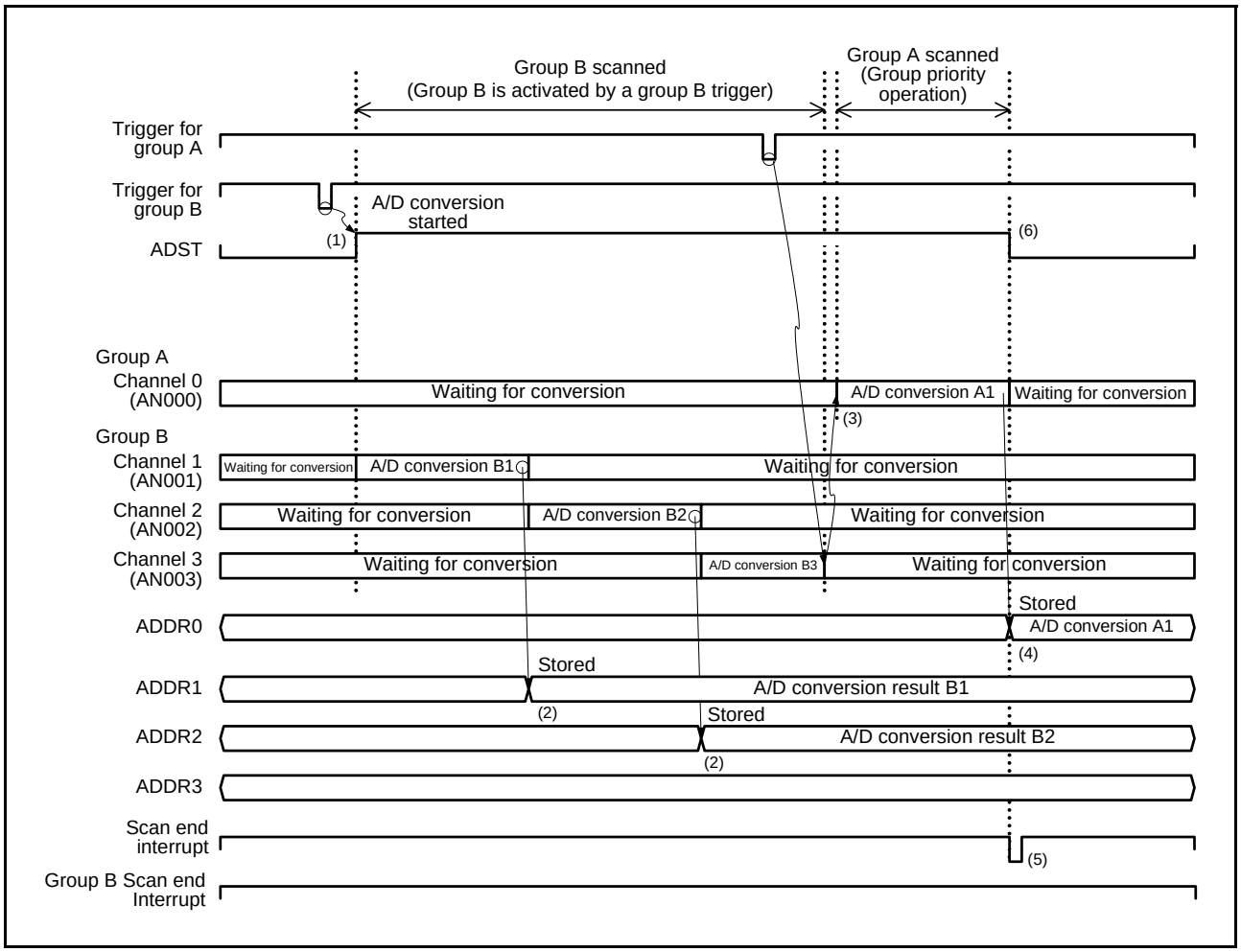


Figure 26.19 Example 4 of Group Priority Operation: Group A Trigger Input during Group B Scan, Without Rescan Setting (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0)

Operation example 5 shows group priority operation in group scan mode (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 1, ADGSPCR.LGRRS = 0) when channel 0 is selected for group A and channels 1 and 2 are selected for group B. When the ADGCTRGR.GRCE bit is set to 1, single scan mode is continuously operated on group C and scan for group B is started by trigger input.

Operation example 5: Continuous single scan operation on group B

(1) When setting ADGSPCR.GBRP to 1 sets the ADCSR.ADST bit to 1 (A/D conversion start), scan for the ANn channels selected in the ADANSB0 register, starts from the channel with the smallest number n.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(3) If a group A trigger is input during scan for group B, group B scan is discontinued while the ADCSR.ADST bit remains 1, and scan for the ANn channels of group A selected in the ADANSA0 register, starts from the channel with the smallest number n. If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADDRy).
(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(5) After scan for group A is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(6) If the ADGSPCR.GBRP bit is set to 1 (single scan is continuously operated), scan for the ANn channels of group B selected in the ADANSB0 register, restarts from the channel with the smallest number n while the ADCSR.ADST bit remains 1.
(7) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy).
(8) A group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group B scan completion enabled).
(9) If the ADGSPCR.GBRP bit is set to 1 (single scan is continuously operated), scan for the ANn channels of group B selected in the ADANSB0 register, restarts from the channel with the smallest number n while the ADCSR.ADST bit remains 1.

To continuously operate single scan for group B, disable trigger input for group B.
Steps 6 to 9 are repeated as long as the ADGSPCR.GBRP bit remains 1. Clearing of the ADCSR.ADST bit to 0 is prohibited while the ADGSPCR.GBRP bit is set to 1. To forcibly stop scanning when ADGSPCR.GBRP = 1, follow the procedures for clear operation by software through the ADCSR.ADST bit shown in section 26.6.2, Notes on Stopping A/D Conversion.

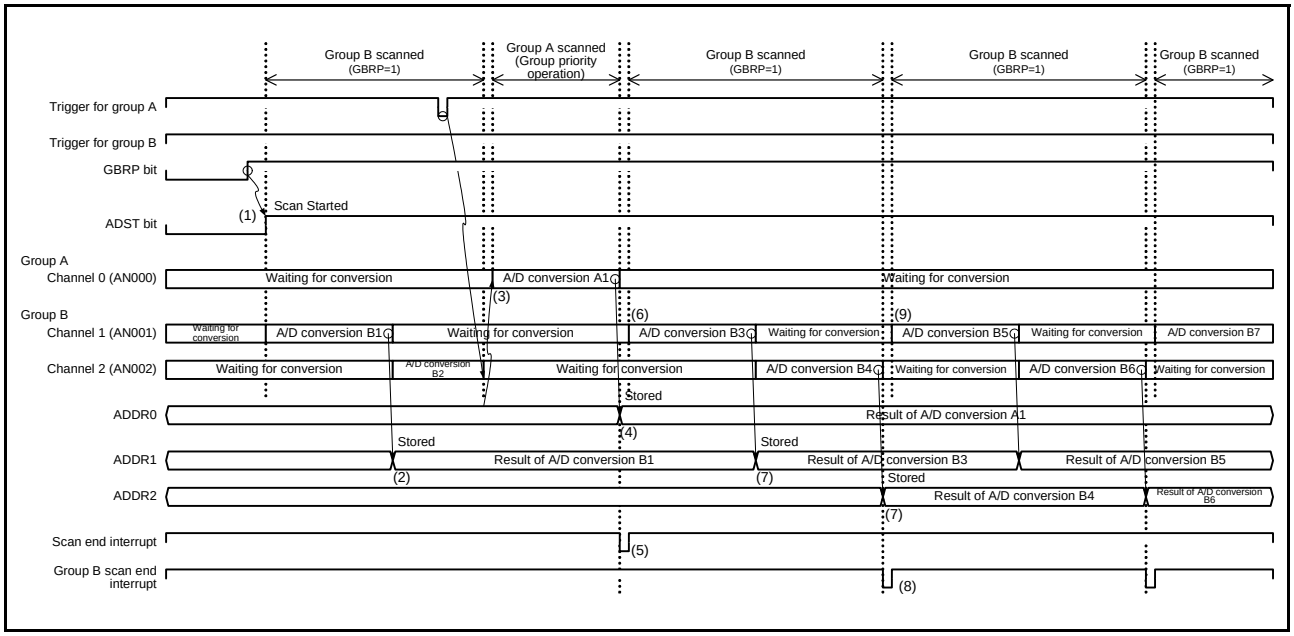


Figure 26.20 Example 5 for Group Priority Operation: Continuous Single Scan Operation on Group B (ADGSPCR.GBRP = 1, ADGSPCR.LGRRS = 1, ADGCTRGR.GRCE = 0)

(2) Group Priority Operation for Three Groups (ADGSPCR.PGS = 1, ADGCTRGR.GRCE = 1)

The following examples 1 to 5 show group priority operation in group scan mode (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 1) when channel 0 is selected for group A, channels 1 and 2 are selected for group B, and channels 3 and 4 are selected for group C.
The priority groups mean groups A and B for group C and group A for group B.

Operation example 1: Priority group trigger input during low-priority group scan, with rescan setting

(1) When a group C trigger input sets the ADCSR.ADST bit to 1 (A/D conversion start), scan for the ANn channels selected in the ADANSC0 register, starts from the channel with the smallest number n.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(3) If a group B trigger is input during scan for group C, group C scan is discontinued while the ADCSR.ADST bit remains 1, and scan for the ANn channels of group B selected in the ADANSB0 register, starts from the channel with the smallest number n. On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(5) If a group A trigger is input during scan for group B, group B scan is discontinued while the ADCSR.ADST bit remains 1, and scan for the ANn channels of group A selected in the ADANSA0 register, starts from the channel with the smallest number n. If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADDRy).
(6) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(7) A scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(8) If the ADGSPCR.GBRSCN bit is 1 (scan for the group is restarted after having been discontinued due to group priority operation), scan for the ANn channels of group B selected in the ADANSB0 register, restarts from the channel with the smallest number n while the ADCSR.ADST bit remains 1. If the ADGSPCR.LGRRS bit is set to 1 at this time, scan for group B starts from the channel on which A/D conversion was discontinued.
(9) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(10) A group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group B scan completion enabled).
(11) If the ADGSPCR.GBRSCN bit is 1, scan for the ANn channels of group C selected in the ADANSC0 register, restarts from the channel with the smallest number n while the ADCSR.ADST bit remains 1. If the ADGSPCR.LGRRS bit is set to 1 at this time, scan for group C starts from the channel on which A/D conversion was discontinued.
(12) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(13) A group C scan end interrupt is generated if the ADGCTRGR.GCADIE bit is 1 (interrupt generation upon group C scan completion enabled).
(14) The ADST bit is automatically cleared to 0 when scan for group A is completed, and the 12-bit A/D converter enters a wait state.

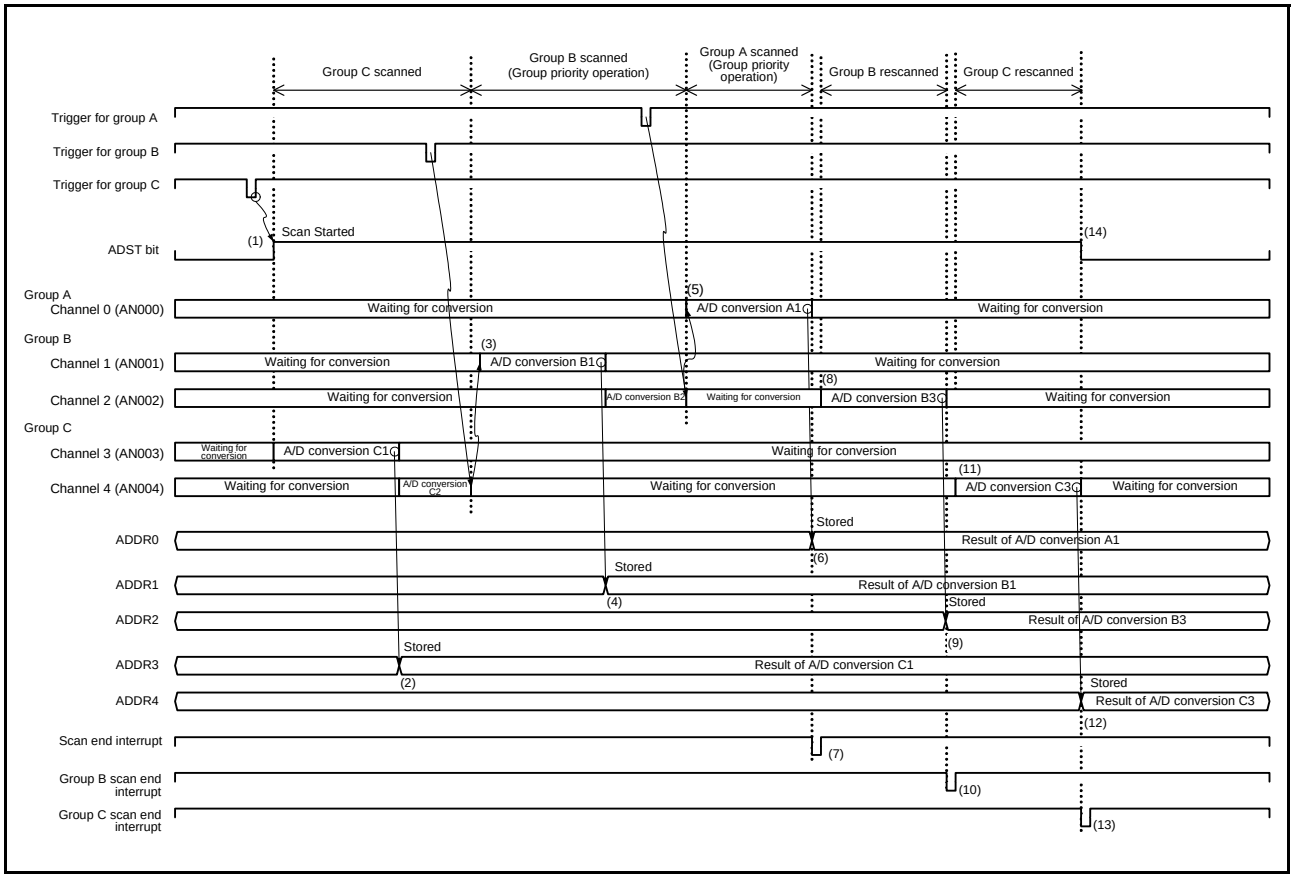


Figure 26.21 Example 1 of Group Priority Operation: Priority Group Trigger Input during Low-Priority Group Scan, With Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 1)

Operation example 2: Priority group trigger input during low-priority group rescan, with rescan setting

Figure 26.22 shows an example when a group A trigger is input during rescan operation on group B.

If a trigger for the priority groups (groups A and B for group C and group A for group B) is input, scan for the priority group starts even while rescan operation on the low-priority group is in progress. After scan for the priority group is completed, scan for the low-priority group is restarted after having been discontinued.

Operations of the ADCSR.ADST bit, storing to the A/D conversion result in the A/D data register (ADDRy), and interrupt requests are the same as example 1.

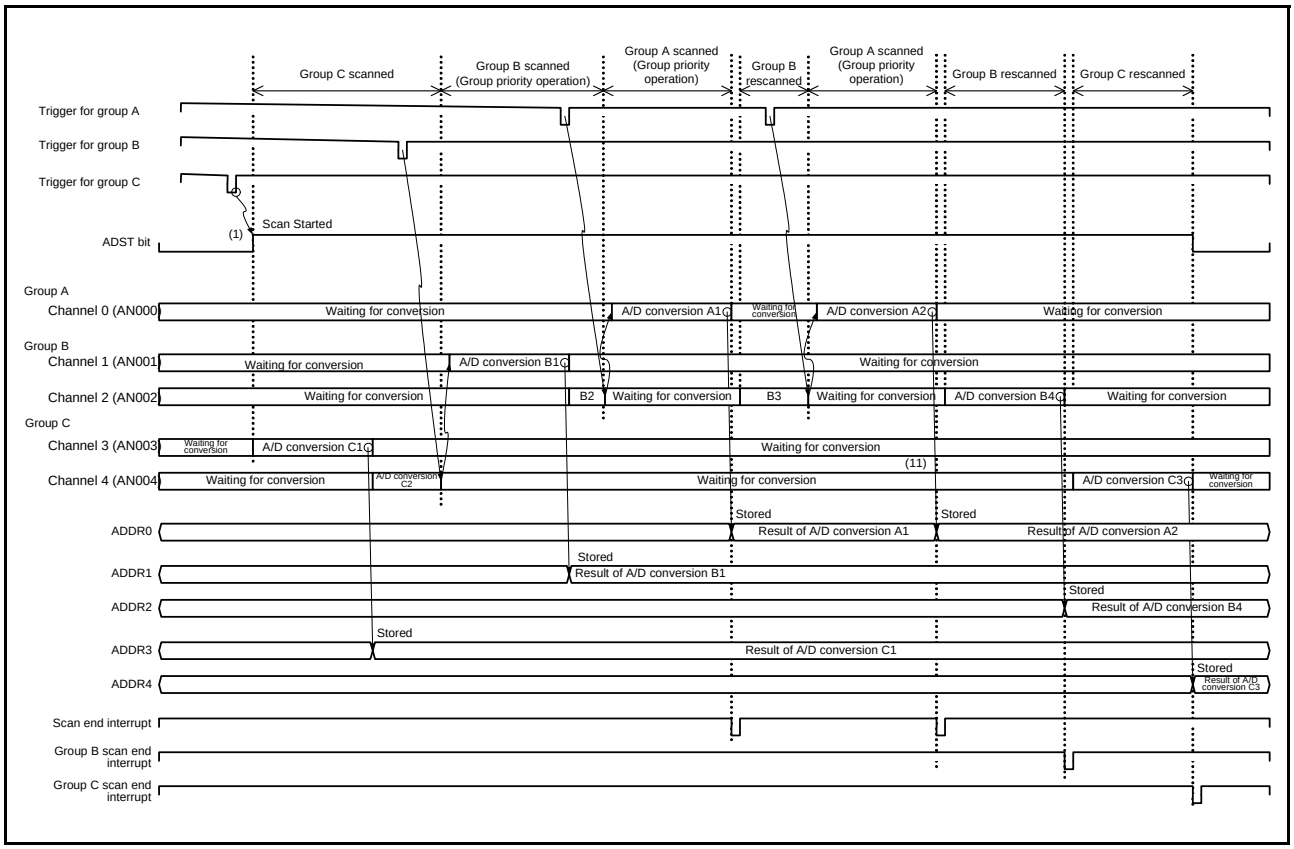


Figure 26.22 Example 2 of Group Priority Operation: Priority Group Trigger Input during Low-Priority Group Rescan, With Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 1)

Operation example 3: Low-priority group trigger input during priority group scan, with rescan setting

The following describes an example when a trigger for the low-priority group is input during scan operation on the priority group when the ADGPSCR.GBRSCN bit is 1 (scan for the group is restarted after having been discontinued due to group priority operation). If the ADGPSCR.GBRSCN bit is 0, all triggers for the low-priority group that are input during scan operation on the priority group are disabled.

(1) When a group A trigger input sets the ADCSR.ADST bit to 1 (A/D conversion start), scan for the ANn channels of group A selected in the ADANSA0 register, starts from the channel with the smallest number n.
(2) If a group B trigger is input during scan for group A, scan for group B can be started.
(3) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(4) After scan for group A is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(5) After scan for group A is completed, scan for the ANn channels of group B selected in the ADANSB0 register, starts from the channel with the smallest number n while the ADCSR.ADST bit remains 1. If the ADGSPCR.LGRRS bit is set to 1 at this time, scan for group B starts from the channel on which A/D conversion was discontinued.
When a group A trigger is input during scan for group B, group A scan starts as in example 1, and group B scan starts after group A scan is completed.
(6) If a group C trigger is input during scan for group B, scan for group C can be started.
(7) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(8) After scan for group B is completed, a group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group B scan completion enabled).
(9) After scan for group B is completed, scan for the ANn channels of group C selected in the ADANSC0 register, starts from the channel with the smallest number n while the ADCSR.ADST bit remains 1. If the ADGSPCR.LGRRS bit is set to 1 at this time, scan for group C starts from the channel on which A/D conversion was discontinued.
When a group A or B trigger is input during scan for group C, group A or B scan starts as in example 1, and group C scan starts after group A or B scan is completed.
(10) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(11) After scan for group C is completed, a group C scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group C scan completion enabled).
(12) The ADCSR.ADST bit is automatically cleared to 0 when scan of all selected channels is completed, and the 12-bit A/D converter enters a wait state.

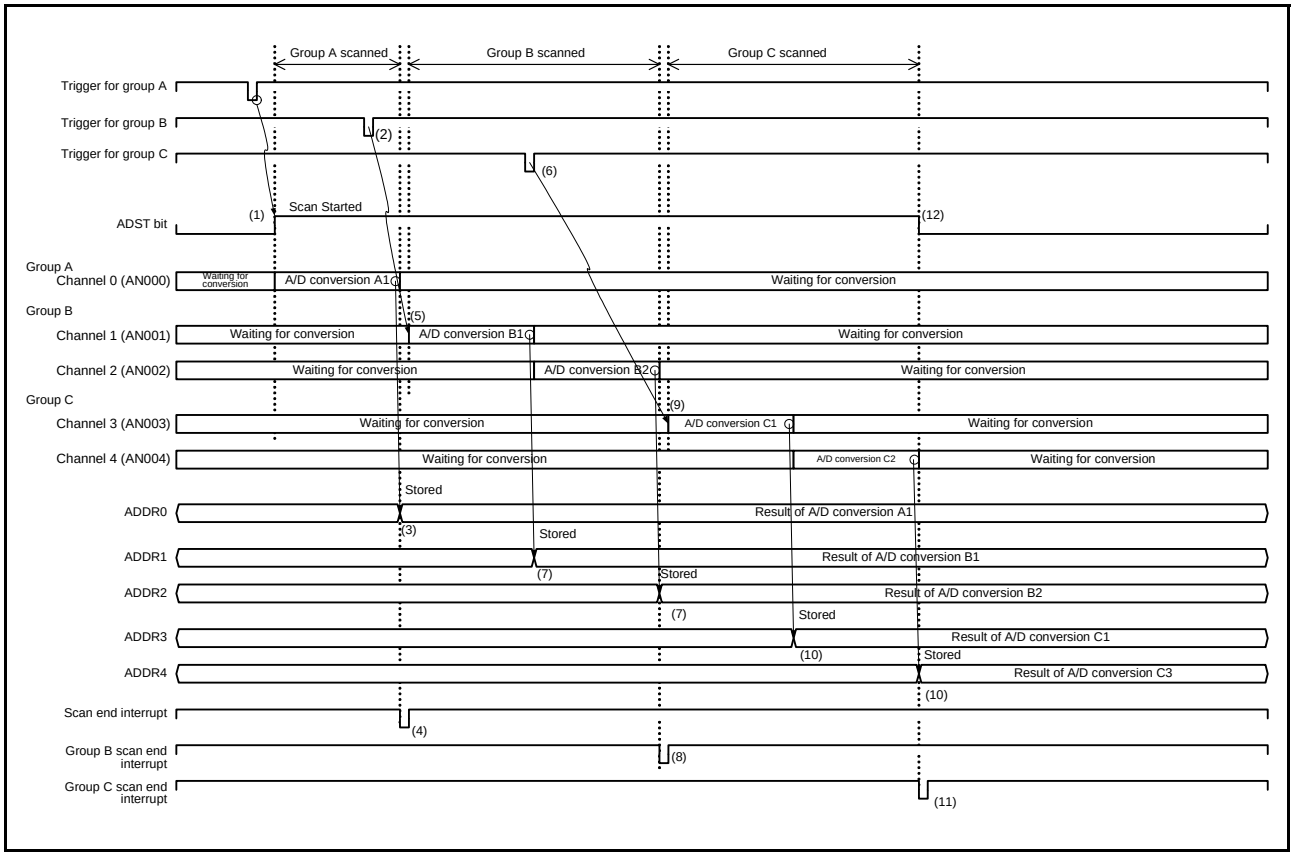


Figure 26.23 Example 3 of Group Priority Operation: Low-Priority Group Trigger Input during Priority Group Scan, With Rescan Setting (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 1, ADGCTRGR.GRCE = 1)

Operation example 4 shows group priority operation in group scan mode (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 0) when channel 0 is selected for group A, channels 1 and 2 are selected for group B, and channels 3 and 4 are selected for group C.

Operation example 4: Priority group trigger input during low-priority group scan, without rescan setting

(1) When a group C trigger input sets the ADCSR.ADST bit to 1 (A/D conversion start), scan for the ANn channels selected in the ADANSC0 register, starts from the channel with the smallest number n.

(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)

(3) If a group B trigger is input during scan for group C, group C scan is discontinued while the ADCSR.ADST bit remains 1, and scan for the ANn channels of group B selected in the ADANSB0 register, starts from the channel with the smallest number n. If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADDRy).

(4) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)

(5) If a group A trigger is input during scan for group B, group B scan is discontinued while the ADCSR.ADST bit remains 1, and scan for the ANn channels of group A selected in the ADANSB0 register, starts from the channel with the smallest number n. If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADDRy).

(6) After scan for group A is completed, a scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).

(7) The ADST bit is automatically cleared to 0 when scan for group A is completed, and the 12-bit A/D converter enters a wait state. Scan for groups C and B is not started until the next trigger corresponding to the group is input.

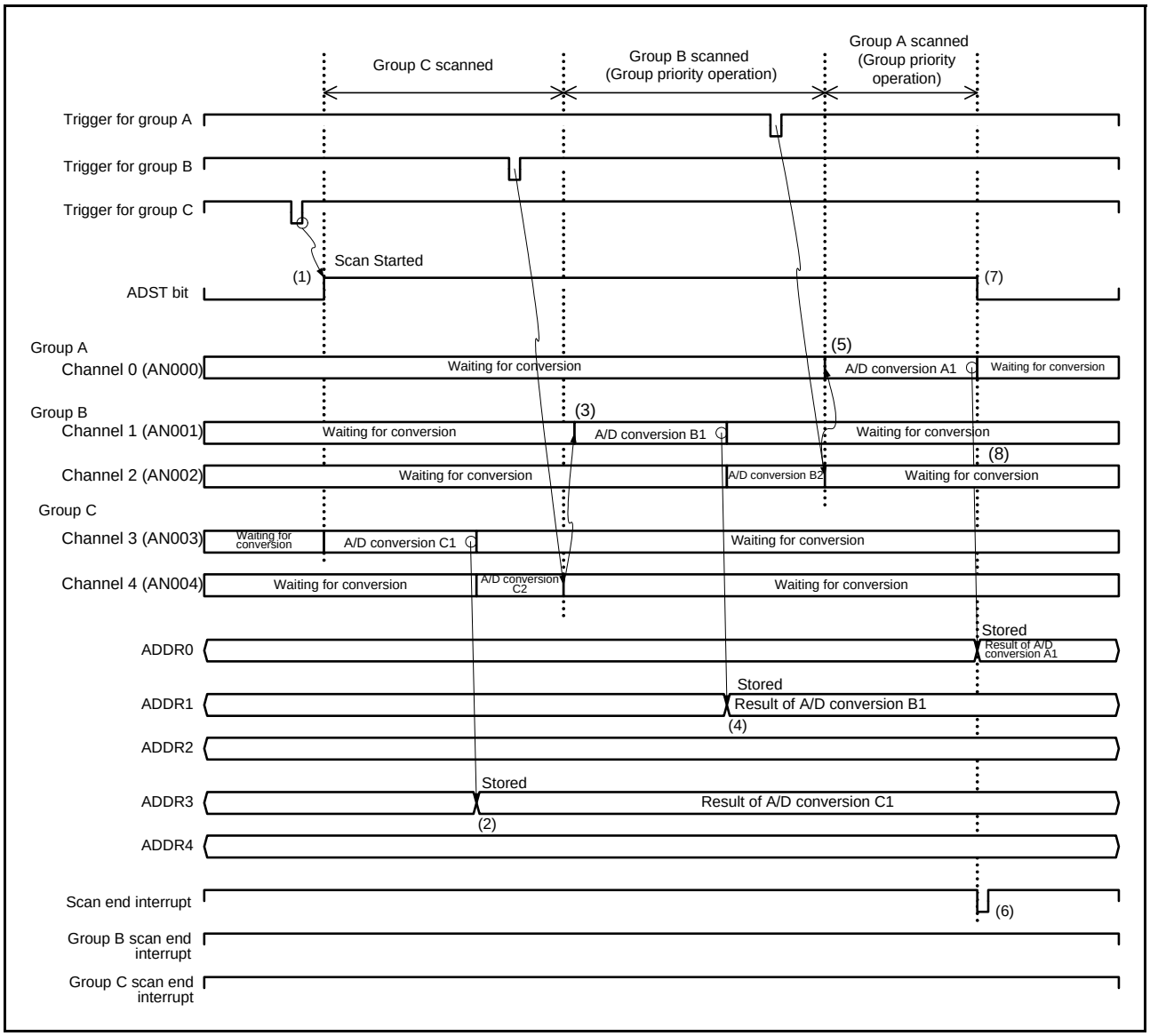


Figure 26.24 Example 4 of Group Priority Operation: Priority Group Trigger Input during Low-Priority Group Scan, Without Rescan Setting (ADGSPCR.GBRSCN = 0, ADGSPCR.GBRP = 0, ADGSPCR.LGRRS = 1)

Operation example 5 shows group priority operation in group scan mode (ADGSPCR.GBRSCN = 1, ADGSPCR.GBRP = 1, ADGSPCR.LGRRS = 1) when channel 0 is selected for group A, channel 1 is selected for group B, and channels 2 and 3 are selected for group C.
When the ADGCTRGR.GRCE bit is set to 1, single scan mode is continuously operated on group B and trigger input for group C is disabled.

Operation example 5: Continuous single scan operation on group C

(1) When setting ADGSPCR.GBRP to 1 sets the ADCSR.ADST bit to 1 (A/D conversion start), scan for the ANn channels selected in the ADANSC0 register, starts from the channel with the smallest number n.
(2) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(3) If a group B trigger is input during scan for group C, group C scan is discontinued while the ADCSR.ADST bit remains 1, and scan for the ANn channels of group B selected in the ADANSB0 register, starts from the channel with the smallest number n. If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADDRy).
(4) If a group A trigger is input during scan for group B, group B scan is discontinued while the ADCSR.ADST bit remains 1, and scan for the ANn channels of group A selected in the ADANSA0 register, starts from the channel with the smallest number n. If A/D conversion is not completed when scan was discontinued, the result is not stored in the corresponding A/D data register (ADDRy).
(5) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(6) A scan end interrupt request is generated if the ADCSR.ADIE bit is 1 (interrupt generation upon scanning completion enabled).
(7) If the ADGSPCR.GBRSCN bit is 1 (scan for the group is restarted after having been discontinued due to group priority operation), scan for the ANn channels of group B selected in the ADANSB0 register, restarts from the channel with the smallest number n while the ADCSR.ADST bit remains 1. If the ADGSPCR.LGRRS bit is set to 1 at this time, scan for group B starts from the channel on which A/D conversion was discontinued.
(8) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(9) A group B scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group B scan completion enabled).
(10) If the ADGSPCR.GBRSCN bit is 1 (scan for the group is restarted after having been discontinued due to group priority operation), scan for the ANn channels of group C selected in the ADANSC0 register, restarts from the channel with the smallest number n while the ADCSR.ADST bit remains 1. If the ADGSPCR.LGRRS bit is set to 1 at this time, scan for group C starts from the channel on which A/D conversion was discontinued.
(11) On completion of A/D conversion on a single channel, the result is stored in the corresponding A/D data register (ADDRy)
(12) A group C scan end interrupt request is generated if the ADCSR.GBADIE bit is 1 (interrupt generation upon group C scan completion enabled).
(13) If the ADGSPCR.GBRP bit is set to 1 (single scan is continuously operated), scan for the ANn channels of group C selected in the ADANSC0 register, restarts from the channel with the smallest number n while the ADCSR.ADST bit remains 1.

To continuously operate single scan for group C, disable trigger input for group B.
Steps 13, 11, 12, and then 13 are repeated as long as the ADGSPCR.GBRP bit remains 1.
Clearing of the ADCSR.ADST bit to 0 is prohibited while the ADGSPCR.GBRP bit is set to 1.
To forcibly stop scanning when ADGSPCR.GBRP = 1, follow the procedures for clear operation by software through the ADCSR.ADST bit shown in section 26.6.2, Notes on Stopping A/D Conversion.

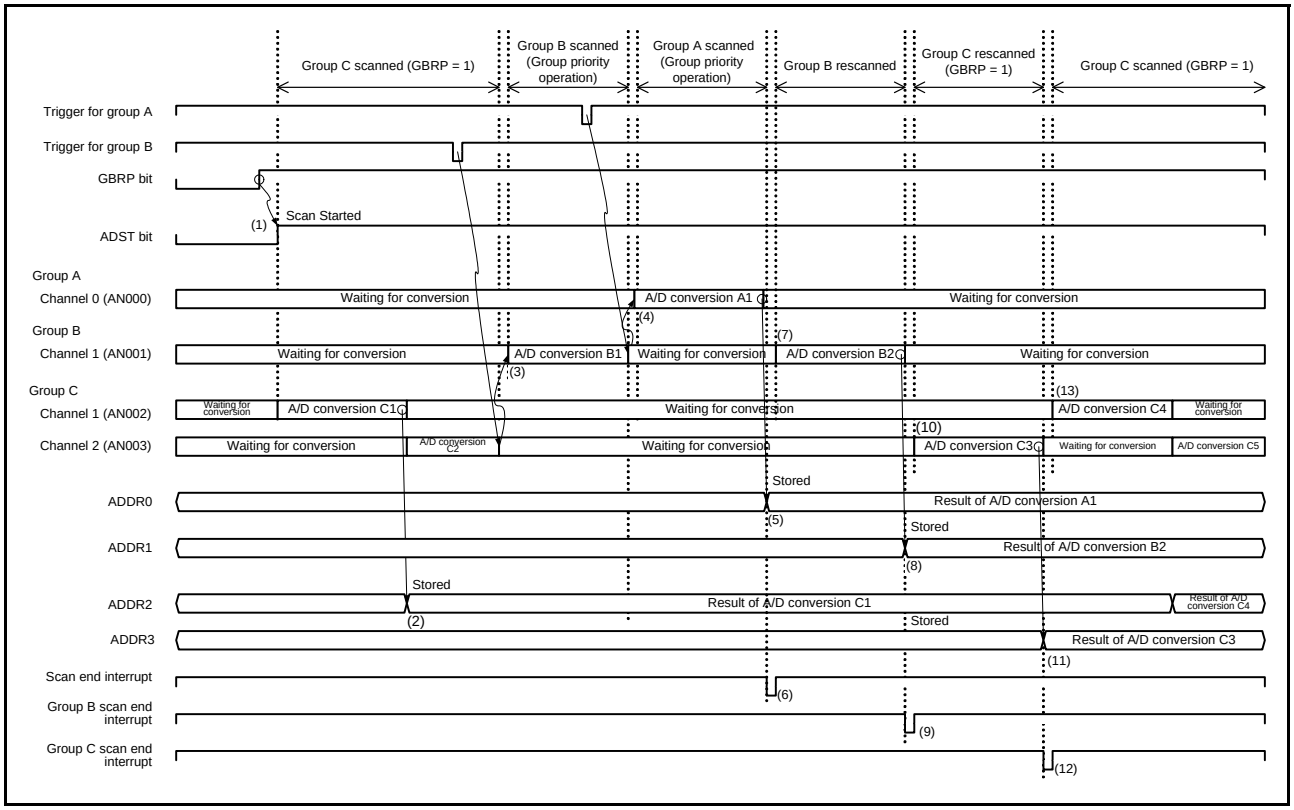


Figure 26.25 Example 5 for Group Priority Operation: Continuous Single Scan Operation on Group C (ADGSPCR.GBRP = 1, ADGSPCR.LGRRS = 1)

26.3.5 Analog Input Sampling Time and Scan Conversion Time

Figure 26.26 shows the scan conversion timing in single scan mode, in which scan conversion is activated by software or a synchronous trigger. Figure 26.27 shows the scan conversion timing in single scan mode, in which scan conversion is activated by an asynchronous trigger. The scan conversion time (t_{SCAN}) includes the start-of-scanning-delay time (t_D), channel-dedicated sample-and-hold circuit processing time (t_{SPLSH})*1, disconnection detection assistance processing time (t_{DIS})*2, self-diagnosis A/D conversion processing time (t_{DIAG})*3, A/D conversion processing time (t_{CONV}), channel-dedicated sample-and-hold circuit end time (t_{SHED})*4, and end-of-scanning-delay time (t_{ED}).
The A/D conversion processing time (t_{CONV}) consists of sampling time (t_{SPL}) and time for conversion by successive approximation (t_{SAM}). The sampling time (t_{SPL}) is used to charge sample-and-hold circuits in the A/D converter. If there is not sufficient sampling time due to the high impedance of an analog input signal source, or if the A/D conversion clock (ADCLK) is slow, sampling time can be adjusted using the ADSSTR register.
The time for conversion by successive approximation (t_{SAM}) is at 32 ADCLK states. Table 26.13 shows the scan conversion time.

The scan conversion time (t_{SCAN}) in single scan mode for which the number of selected channels is n can be determined as follows:

$$t_{SCAN} = t_D + t_{SPLSH} + (t_{DIS} \times n) + t_{DIAG} + (t_{CONV} \times n)^{*5} + t_{ED}$$

The scan conversion time for the first cycle in continuous scan mode is t_{SCAN} for single scan minus t_{ED} plus t_{SHED}.
The scan conversion time for the second and subsequent cycles in continuous scan mode is fixed to $t_{SPLSH} + (t_{DIS} \times n) + t_{DIAG} + t_{DSD} + (t_{CONV} \times n)^{*5} + t_{SHED}$.

Note 1. When no channel-dedicated sample-and-hold circuits are used, $t_{SPLSH} = 0$.
Note 2. When disconnection detection assistance is not selected, $t_{DIS} = 0$. The auto-discharge period of 15 ADCLK states is inserted only when internal reference voltage is A/D-converted.
Note 3. When the self-diagnosis function is not used, $t_{DIAG} = 0$, $t_{DSD} = 0$.
Note 4. When no channel-dedicated sample-and-hold circuits are used, $t_{SHED} = 0$. Here, continuous scan mode is assumed. In single scan mode and group scan mode, t_{SHED} is included in the end-of-scanning-delay time (t_{ED}).
Note 5. $t_{CONV} \times n$ when the sampling time (t_{SPL}) of selected channels is the same, but it is the total of the sampling time of each channel and time for conversion by successive approximation (t_{SAM}).

Table 26.13 Times for Conversion during Scanning (in Numbers of Cycles of ADCLK and PCLK)

Item			Symbol		Type/Conditions: Synchronous Trigger (MTU)	Type/Conditions: Asynchronous Trigger	Type/Conditions: Software Trigger	Unit
Scan start processing time *1, *2	A/D conversion on group under group priority control.	The low-priority group is to be stopped. (The priority group is activated after low-priority group B is stopped due to an A/D conversion source of the priority group.)	t_D		3 PCLKB + 6 ADCLK	—	—	Cycle
		The low-priority group is not to be stopped. (Activation by an A/D conversion source of the priority group.)			2 PCLKB + 4 ADCLK	—	—	
	A/D conversion when self-diagnosis is enabled	A/D conversion for self-diagnosis is to be started.			2 PCLKB + 6 ADCLK	4 PCLKB+ 6 ADCLK	6 ADCLK	
	Other than above				2 PCLKB + 4 ADCLK	2 PCLKB + 4 ADCLK	4 ADCLK	
Channel-dedicated sample-and-hold processing time*1	Sampling time		t_{SPLSH}	t_{SH}	The setting of ADSHCR.SSTSH[7:0] (initial value = 1Ah) × ADCLK			
	Wait time between sampling and A/D conversion			t_W	13 ADCLK			
Disconnection detection assistance processing time			t_{DIS}		The setting of ADNDIS[3:0] (initial value = 00h) × ADCLK*3			
Self-diagnosis conversion processing time*1	Sampling time		t_{DIAG}	t_{SPL}	The setting of ADSSTR0 (initial value = 0Dh) × ADCLK			
	Time for conversion by successive approximation			t_{SAM}	32 ADCLK			
	Normal A/D conversion is to be started after completion of self-diagnosis conversion.		t_{DED}		2 ADCLK			
	A/D conversion for self-diagnosis is to be started after completion of conversion for continuous scan on the last channel specified.		t_{DSD}		2 ADCLK			
A/D conversion processing time*1	Sampling time		t_{CONV}	t_{SPL}	The setting of ADSSTRn (n = 0 to 7, O) (initial value = 0Dh) × ADCLK			
	Time for conversion by successive approximation			t_{SAM}	32 ADCLK			
Channel-dedicated sample-and-hold end processing time			t_{SHED}		3 ADCLK			
Scan end processing time*1			t_{ED}		1 PCLKB + 3 ADCLK			

Note 1. For t_D, t_{SPLSH}, t_{DIAG}, t_{CONV}, and t_{ED}, see Figure 26.26 and Figure 26.27.
Note 2. This is the maximum time required from software writing or trigger input to A/D conversion start.
Note 3. The value is fixed to 0Fh (15 ADCLK) when the internal reference voltage is A/D-converted.

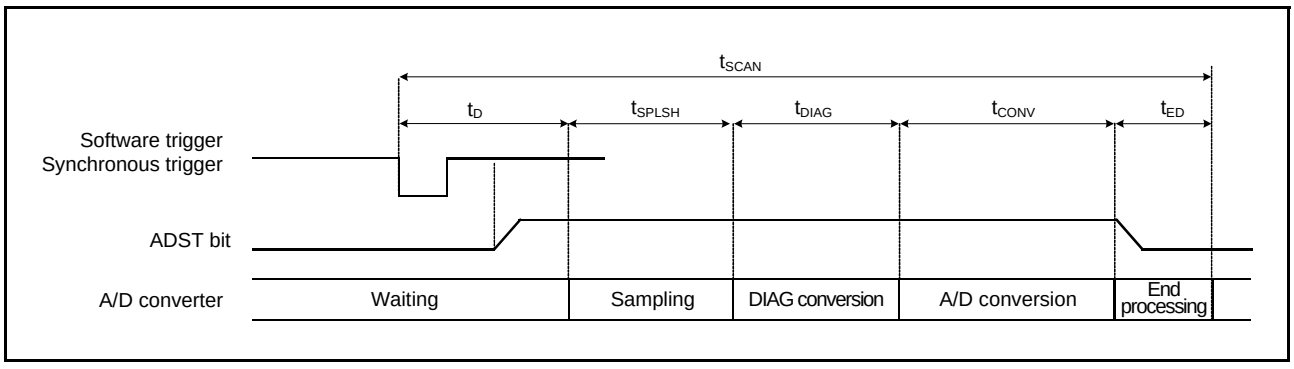


Figure 26.26 Scan Conversion Timing (Activated by Software or Synchronous Trigger)

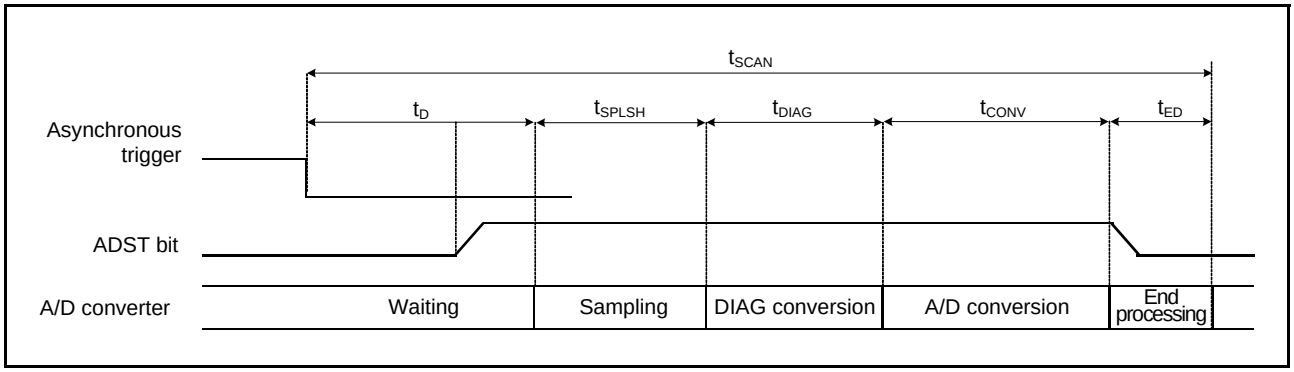


Figure 26.27 Scan Conversion Timing (Activated by Asynchronous Trigger)

26.3.6 Usage Example of A/D Data Register Automatic Clearing Function

Setting the ADCER.ACE bit to 1 automatically clears the A/D data registers (ADDRy, ADRD, ADOCDR, ADDBLDR, ADDBLDRA, ADDBLDRB) to 0000h when the A/D data registers (ADDRy, ADRD, ADOCDR, ADDBLDR, ADDBLDRA, ADDBLDRB) are read by the CPU or DTC.
This function enables detection of update failures of the A/D data registers (ADDRy, ADRD, ADOCDR, ADDBLDR, ADDBLDRA, ADDBLDRB). The following describes the examples in which the function to automatically clear the ADDRy register is enabled and disabled.
In a case where the ADCER.ACE bit is 0 (automatic clearing disabled), if the A/D conversion result (0222h) is not written to the ADDRy register for some reason, the old data (0111h) will be the ADDRy value. Furthermore, if this ADDRy value is read into a general register using an A/D scan end interrupt, the old data (0111h) can be saved in the general register. When checking whether there is an update failure, it is necessary to frequently save the old data in the RAM or a general register.
In a case where the ADCER.ACE bit is 1 (automatic clearing enabled), when ADDRy = 0111h is read by the CPU or DTC, ADDRy is automatically cleared to 0000h. After that, if the A/D conversion result 0222h cannot be transferred to ADDRy for some reason, the cleared data (0000h) remains as the ADDRy value. If this ADDRy value is read into a general register using an A/D scan end interrupt at this point, 0000h will be saved in the general register.
Occurrence of an ADDRy update failure can be determined by simply checking that the read data value is 0000h.

26.3.7 A/D-Converted Value Addition/Average Mode

In A/D-converted value addition mode, the same channel is A/D-converted 2, 3, 4, or 16 consecutive times and the sum of the converted values is stored in the data register. In A/D-converted value average mode, 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 are present. This function, however, cannot always guarantee an improvement in A/D conversion accuracy.
The A/D-converted value addition/average mode can be used when A/D conversion of the channel select analog input or internal reference voltage is selected.

26.3.8 Disconnection Detection Assist Function

This converter incorporates the function to fix the charge for sampling capacitance to the specified state before start of A/D conversion. This function enables disconnection detection in wiring of analog inputs. The disconnection detection assist function should be used while ADPGACR.PnENAMP = 0 (does not use the amplifier in the PGA) and ADSHCR.SHANS = 0 (bypass the sample-and-hold circuits).
Figure 26.28 illustrates the A/D conversion operation when the disconnection detection assist function is used. Figure 26.29 shows an example of disconnection detection when precharge is selected. Figure 26.30 shows an example of disconnection detection when discharge is selected.

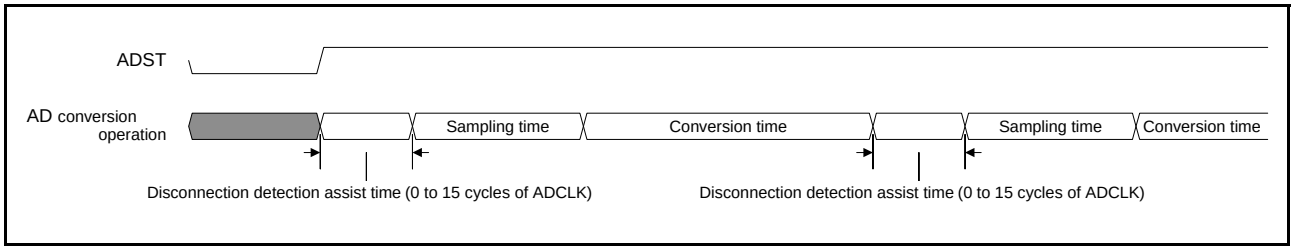
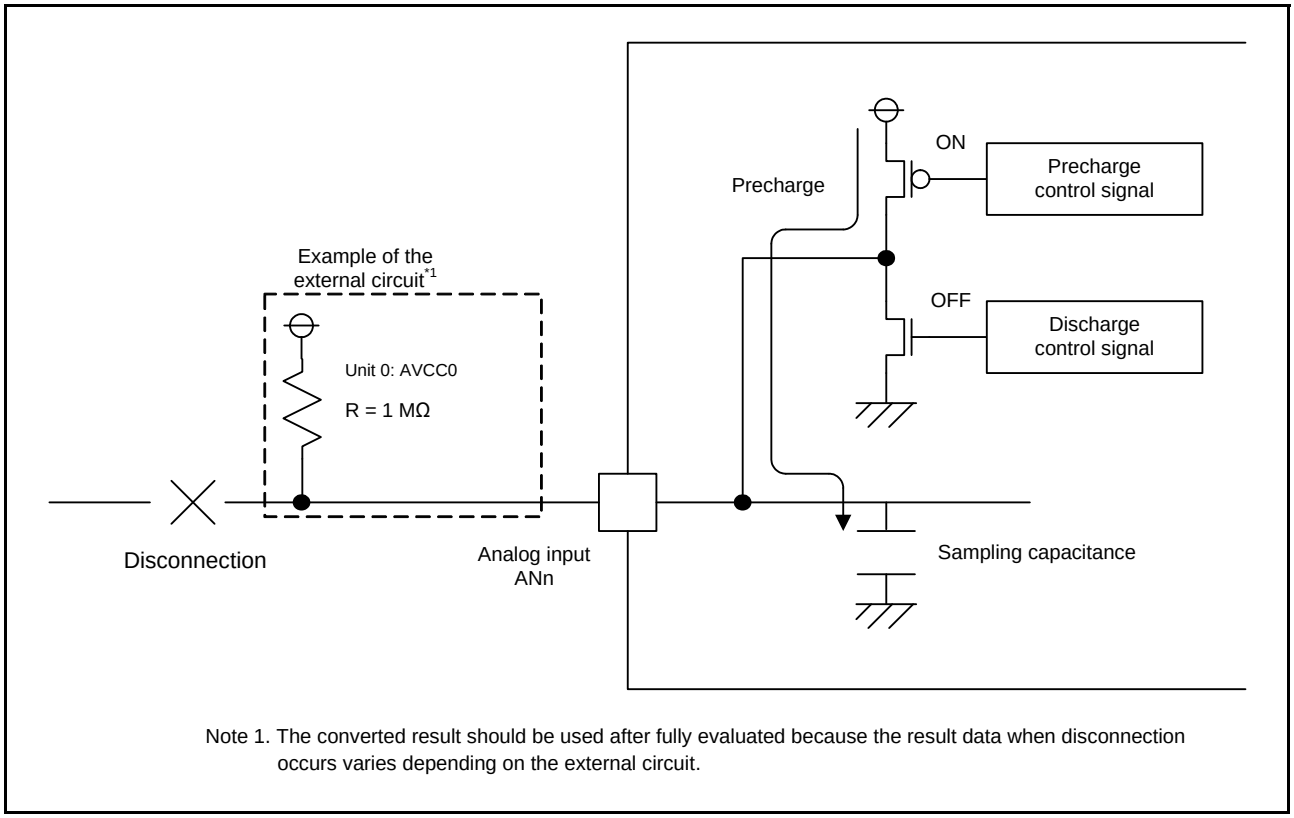
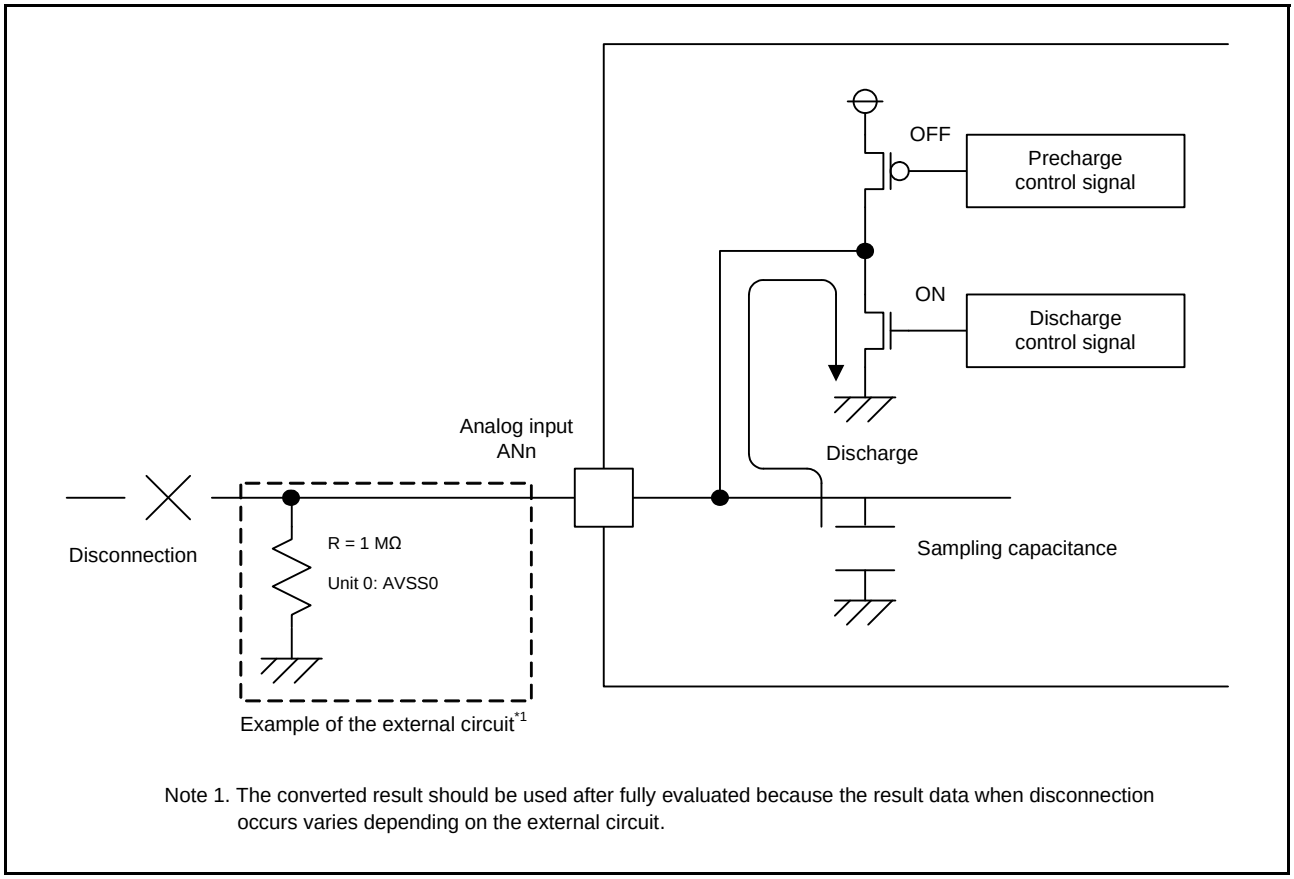


Figure 26.28 Operation of A/D Conversion When the Disconnection Detection Assist Function is Used



Note 1. The converted result should be used after fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

Figure 26.29 Example of Disconnection Detection When Precharge is Selected



Note 1. The converted result should be used after fully evaluated because the result data when disconnection occurs varies depending on the external circuit.

Figure 26.30 Example of Disconnection Detection When Discharge is Selected

26.3.9 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 high-level signal should be input to the asynchronous trigger (ADTRG0# pin). Then, the ADCSR.TRGE and ADCSR.EXTRG bits should be set to 1. Figure 26.31 shows a timing of the asynchronous trigger input.
For the time from when the ADST bit is set to 1 until conversion starts, refer to section 26.6.3, A/D Conversion Restarting Timing and Termination Timing.

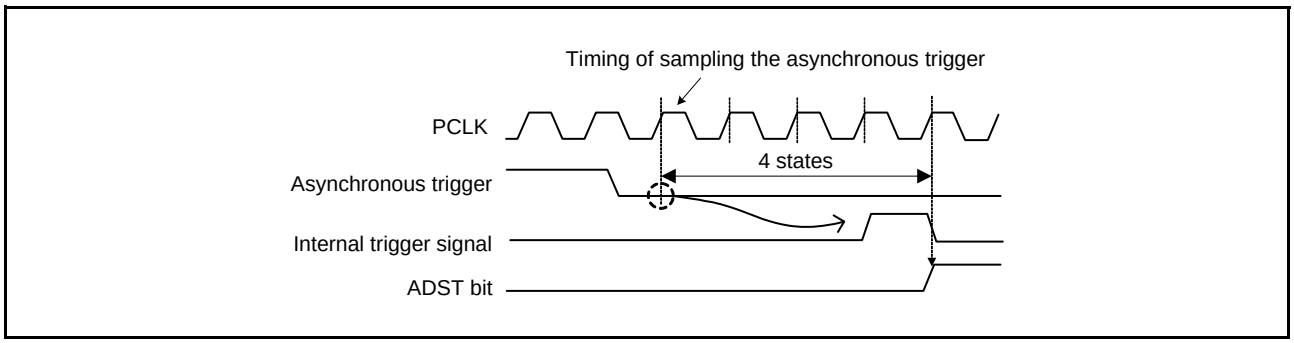


Figure 26.31 Timing of Sampling Asynchronous Trigger

26.3.10 Starting A/D Conversion with Synchronous Trigger from Peripheral Module

The A/D conversion can be started by a synchronous trigger. To start the A/D conversion by a synchronous trigger, the ADCSR.TRGE bit should be set to 1, the ADCSR.EXTRG bit should be cleared to 0, and the relevant sources should be selected by the ADSTRGR.TRSA[5:0] and ADSTRGR.TRSB[5:0] bits.

26.3.11 Programmable Gain Amplifier

A programmable gain amplifier is included in AN000 to AN002. Select the gain by setting the S12AD.ADPGAGS.PnGAIN[3:0] bits (n = 000 to 002) for the AN000 to AN002 pins. Select the operational amplifier to be used by setting the S12AD.ADPGACR.PnSEL1 bit (n = 000 to 002) for the AN000 to AN002 pins.
After setting the S12AD.ADPGACR.PnENAMP bit (n = 000 to 002) for the AN000 to AN002 pins to 1, wait for the period for stabilizing of the programmable gain amplifier operation and set the ADCSR.ADST bit of the unit to 1.

26.4 Interrupt Sources and DTC Transfer Requests

26.4.1 Interrupt Requests

The 12-bit A/D converter can send scan end interrupt requests S12ADI, GBADI, and GCADI to the CPU.
Setting the ADCSR.ADIE bit to 1 and 0 enables and disables an S12ADI interrupt, respectively; similarly, setting the ADCSR.GBADIE bit to 1 and 0 enables and disables a GBADI interrupt, respectively; similarly, setting the ADGCTRGR.GCADIE bit to 1 and 0 enables and disables a GCADI interrupt, respectively.
In addition, the DTC can be activated when an S12ADI, a GBADI, or GCADI interrupt is generated. Using an S12ADI, a GBADI, or GCADI interrupt to allow the DTC to read the converted data enables sequence conversion without burden on software.
For details on DTC settings, see section 16, Data Transfer Controller (DTCb).

26.5 Allowable Impedance of Signal Source

To achieve high-speed conversion of 1.4 µs, the analog input pins of this MCU are designed so that the conversion accuracy is guaranteed if the impedance of the input signal source is 1.0 kΩ or less. If an external capacitor of large capacitance is attached in the application in which only a single pin input is converted in single scan mode, the only load on input is virtually 2.5 kΩ of the internal input resistor; therefore, the impedance of the signal source can be ignored. Being a low-pass filter, however, an analog input circuit may not follow the analog signal with a large differential coefficient. When high-speed analog signals are to be converted or multiple pins are to be converted in scan mode, a low-impedance buffer should be used.
Figure 26.32 shows an equivalent circuit of an analog input pin and an external sensor.
To perform A/D conversion accurately, charging of the internal capacitor C shown in Figure 26.32 must be completed within the specified period of time. This specified period is referred to as sampling time.

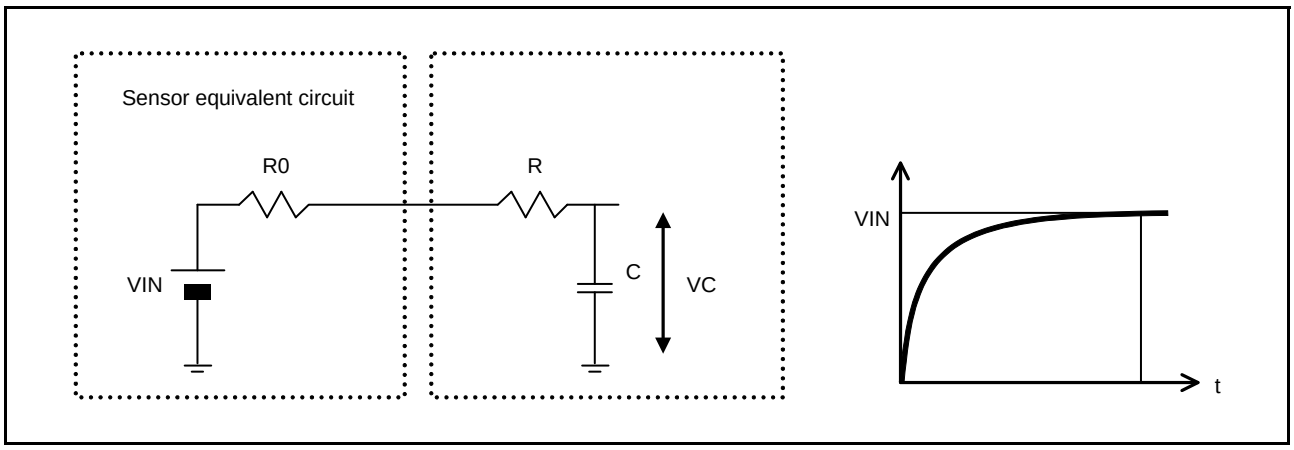


Figure 26.32 Equivalent Circuit of Analog Input Pin and External Sensor

26.6 Usage Notes

26.6.1 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 A/D internal reference voltage data register, and A/D self-diagnosis data register should be read in 16-bit units. If a register is read twice in 8-bit units, that is, the higher-order byte and lower-order byte are separately read, the A/D-converted value having been read first may disagree with the A/D-converted value having been read for the second time. To prevent this, the data registers should never be read in 8-bit units.

26.6.2 Notes on Stopping A/D Conversion

To stop A/D conversion when an asynchronous trigger or a synchronous trigger has been selected as the condition for starting A/D conversion, follow the procedure in Figure 26.33.

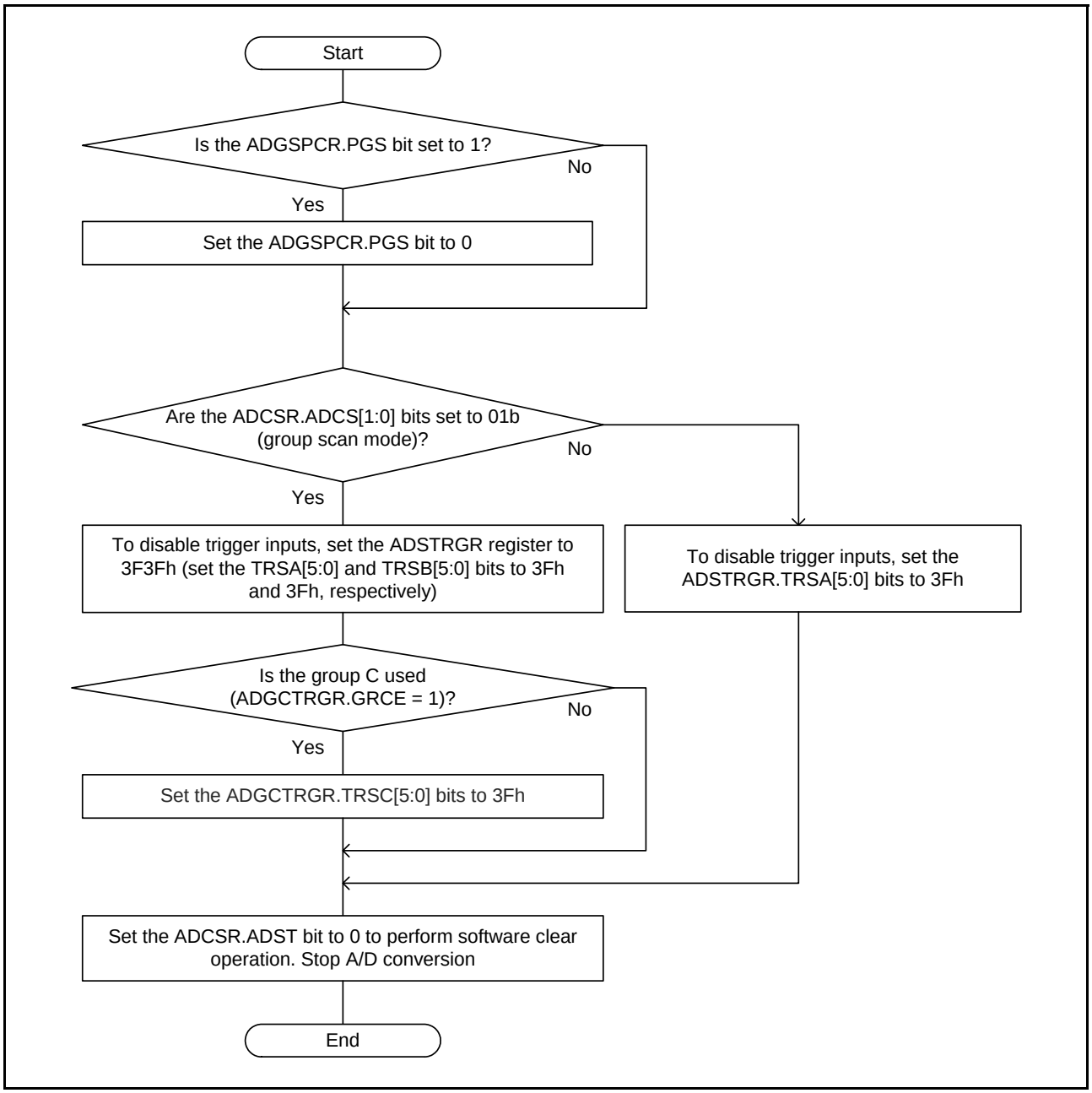


Figure 26.33 Procedure for Clear Operation by Software through the ADCSR.ADST Bit

26.6.3 A/D Conversion Restarting Timing and Termination Timing

It takes a maximum of six ADCLK cycles for the idle analog unit of the 12-bit A/D converter to be restarted by setting the ADCSR.ADST bit to 1. It takes a maximum of three ADCLK cycles for the operating analog unit of the 12-bit A/D converter to be terminated by setting the ADCSR.ADST bit to 0.

26.6.4 Notes on Scan End Interrupt Handling

When scanning the same analog input twice using any trigger, the first A/D-converted data is overwritten with the second A/D-converted data in the case that the CPU does not complete reading 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.

26.6.5 Module Stop Function Setting

Operation of the 12-bit A/D converter can be disabled or enabled by setting module stop control register. The initial setting is for operation of the 12-bit A/D converter to be halted. Register access is enabled by releasing the module stop state.
After the module stop state is released, wait for 1 µs to start A/D conversion. For details, refer to section 11, Low Power Consumption.

26.6.6 Notes on Entering Low Power Consumption States

Before entering the module stop state or software standby mode, make sure to stop A/D conversion. Here, set the ADCSR.ADST bit to 0, and secure certain period of time until the analog unit of the 12-bit A/D converter is stopped. Follow the procedure given below to secure this time.
Set the ADCSR.ADST bit to 0 by following the procedure in Figure 26.33 Procedure for Clear Operation by Software through the ADCSR.ADST Bit. Then wait for three clock cycles of ADCLK before entering the module stop mode or software standby mode.

26.6.7 Notes on Canceling Software Standby Mode

After software standby mode is canceled, wait until the crystal oscillation stabilization time or the PLL circuit stabilization time elapses, and then wait for 1 µs before starting A/D conversion. For details, refer to section 11, Low Power Consumption.

26.6.8 Error in Absolute Accuracy When Disconnection Detection Assistance is in Use

Using disconnection detection assistance leads to an error in absolute accuracy of the A/D converter. This is because an error voltage is input to the analog input pins due to the resistive voltage division between the pull-up or pull-down resistor (Rp) and the resistance of the signal source (Rs). The disconnection detection assist function should be used while ADPGACR.PnENAMP = 0 (does not use the amplifier in the PGA) and ADSHCR.SHANS = 0 (bypass the sample-and-hold circuits). This error in absolute accuracy is calculated from the following formula. Only use disconnection detection assistance after thorough evaluation.

Maximum error in absolute accuracy (LSB) = 4095 × Rs / Rp

26.6.9 Voltage Range of Analog Power Supply Pins

If this MCU is used with the voltages outside the following ranges, the reliability of the MCU may be affected.

- Analog input voltage range
  Voltage applied to analog input pins AN000 to AN007: AVSS0 ≤ VAN0 ≤ AVCC0
- Relationship between power supply pin pairs (AVCC0–AVSS0, VCC–VSS)
  The following condition should be satisfied: AVSS0 = VSS.
  A 0.1-µF capacitor should be connected between each pair of power supply pins to create a closed loop with the shortest route possible as shown in Figure 26.34, and connection should be made so that the following condition is satisfied at the supply side: AVSS0 = VSS
  When the 12-bit A/D converter is not used, the following conditions should be satisfied:
  AVCC0 = VCC and AVSS0 = VSS

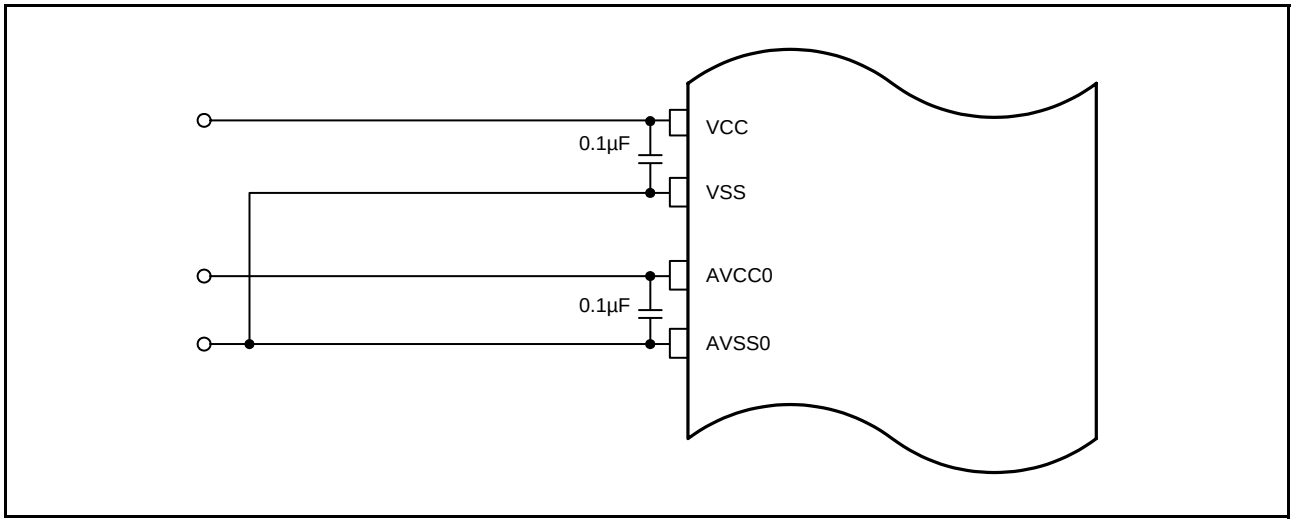


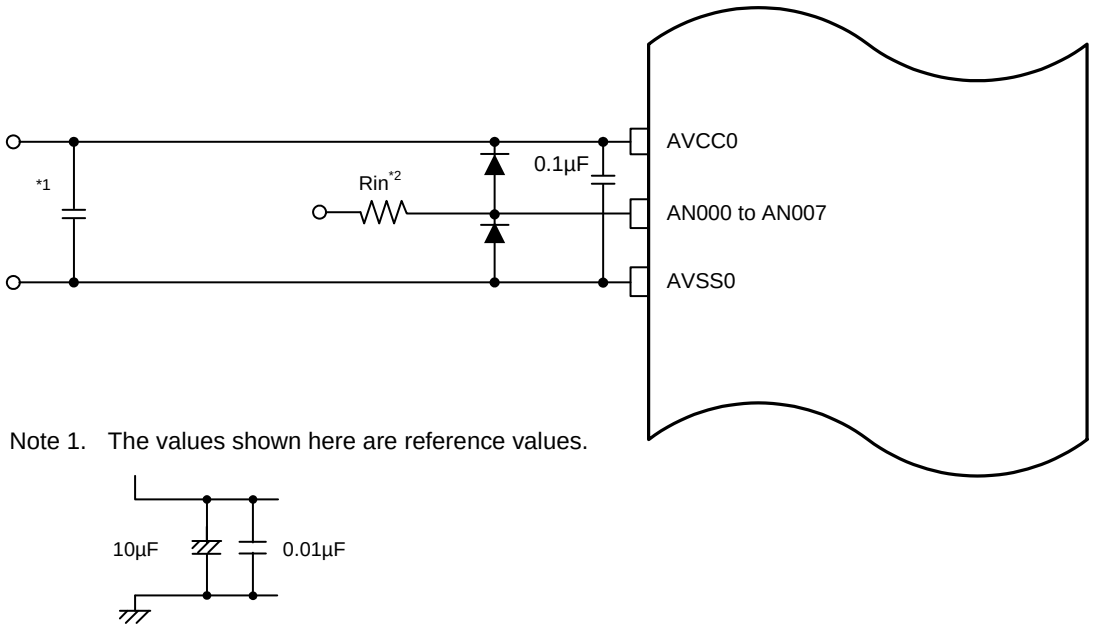
Figure 26.34 Power Supply Pin Connection Example

26.6.10 Notes on Board Design

The board should be designed so that digital circuits and analog circuits are separated from each other as far as possible. In addition, digital circuit signal lines and analog circuit signal lines should not intersect or placed near each other. If these rules are not followed, noise will be produced on analog signals and A/D conversion accuracy will be affected. The analog input pins (AN000 to AN007), and analog power supply (AVCC0) should be separated from digital circuits using the analog ground (AVSS0). The analog ground (AVSS0) should be connected to a stable digital ground (VSS) on the board (single-point ground plane connection).

26.6.11 Notes on Noise Prevention

To prevent the analog input pins (AN000 to AN007) from being destroyed by abnormal voltage such as excessive surge, a capacitor should be inserted between AVCC0 and AVSS0, and a protection circuit should be connected to protect the above analog input pins as shown Figure 26.35.



Note 1. The values shown here are reference values.

Note 2. Rin: Signal source impedance

Figure 26.35 Sample Protection Circuit for Analog Inputs

27. D/A Converter for Generating Comparator C Reference Voltage (DA)

27.1 Overview

This MCU includes one channel of 8-bit D/A converter. This D/A converter is used for generating comparator C reference voltage.

Table 27.1 lists the specifications of the 8-bit D/A converter and Figure 27.1 shows a block diagram of the 8-bit D/A converter.

Table 27.1 Specifications of 8-Bit D/A Converter

Item	Specifications
Resolution	8 bits
Output channels	One channel
Low power consumption function	Module stop state can be set.

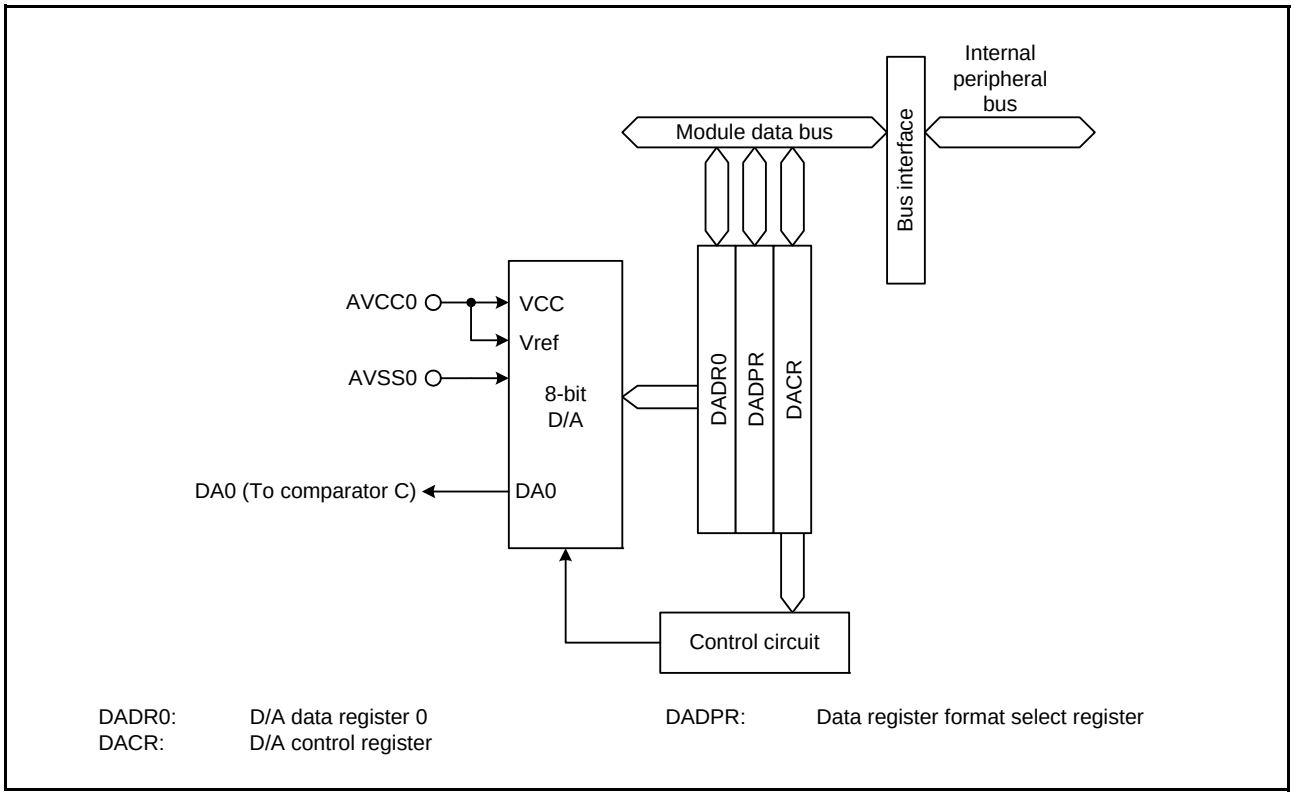


Figure 27.1 Block Diagram of 8-Bit D/A Converter

27.2 Register Descriptions

27.2.1 D/A Data Register 0 (DADR0)

Address(es): DA.DADR0 0008 80C0h

• DADPR.DPSEL bit = 0 (data is right-justified)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—								
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

• DADPR.DPSEL bit = 1 (data is left-justified)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
									—	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DADR0 register is a 16-bit readable/writable register, which stores data to which D/A conversion is to be performed. Whenever an analog output is enabled, the values in DADR0 are converted and the reference voltage for comparator C is supplied.

8-bit data can be relocated by setting the DADPR.DPSEL bit.

Bits “—” are read as 0. The write value should be 0.

27.2.2 D/A Control Register (DACR)

Address(es): DA.DACR 0008 80C4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	DAOE0	—	—	—	—	—	—
Value after reset:	0	0	0	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b4 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R
b6	DAOE0	D/A Output Enable 0	0: D/A conversion is disabled. 1: D/A conversion is enabled.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R

DAOE0 Bit (D/A Output Enable 0)

The DAOE0 bit controls the D/A conversion.

27.2.3 Data Register Format Select Register (DADPR)

Address(es): DA.DADPR 0008 80C5h

	b7	b6	b5	b4	b3	b2	b1	b0
	DPSEL	—	—	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DPSEL	Format Select	0: Data is right-justified. 1: Data is left-justified.	R/W

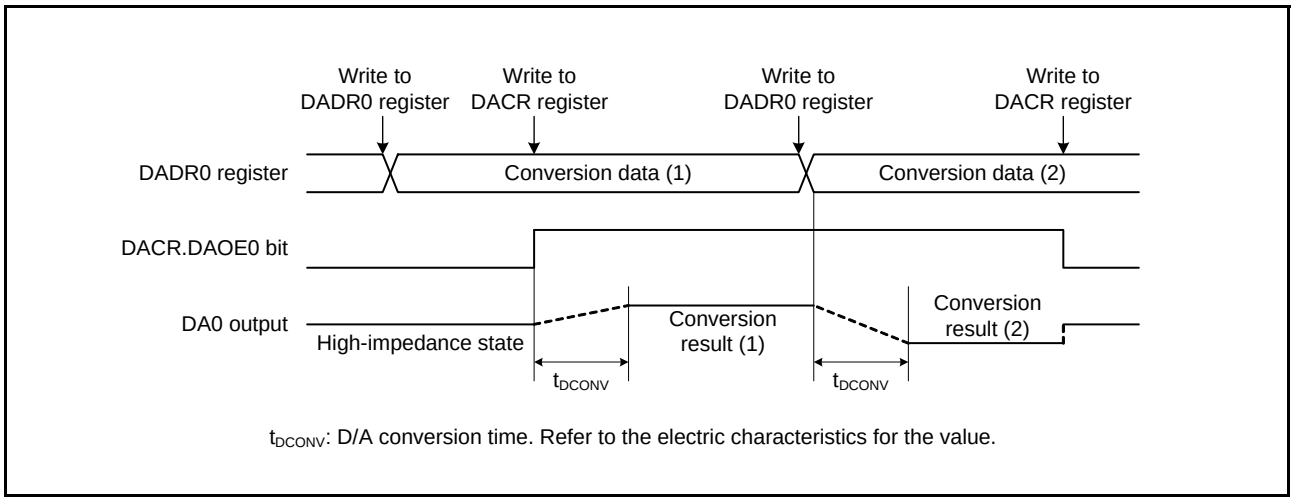
27.3 Operation

When the DACR.DAOE0 bit is set to 1, D/A converter is enabled and the conversion result is output.
An operation example of D/A conversion is shown below. Figure 27.2 shows the timing of this operation.

(1) Set the data for D/A conversion in the DADPR.DPSEL bit and the DADR0 register.
(2) Set the DACR.DAOE0 bit to 1 to start D/A conversion. The DA0 output settles to the voltage corresponding to the setting value after the conversion time tDCONV has elapsed. The DA0 output voltage is held at this level until the DADR0 register is updated or the DAOE0 bit is set to 0. The output voltage (reference) is expressed by the following formula:

$$\frac{\text{Value of DADR0 register}}{256} \times \text{AVCC0}$$

(3) When the DADR0 register is updated, the conversion starts. The DA0 output settles at the new output voltage after the conversion time tDCONV has elapsed.
(4) When the DAOE0 bit is set to 0, D/A conversion is disabled.



t_{DCONV}: D/A conversion time. Refer to the electric characteristics for the value.

Figure 27.2 Example of 8-Bit D/A Converter Operation

27.4 Usage Notes

27.4.1 Module Stop Function Setting

Operation of the 8-bit D/A converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the 8-bit D/A converter to be stopped. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

27.4.2 Operation of the D/A Converter in Module Stop State

When the MCU enters the module stop state with D/A conversion enabled, the D/A converter outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current has to be reduced in the module stop state, disable D/A conversion by setting the DACR. and DAOE0 bit to 0.

27.4.3 Operation of the D/A Converter in Software Standby Mode

When the MCU enters software standby mode with D/A conversion enabled, the D/A converter outputs are retained, and the analog power supply current is the same as during D/A conversion. If the analog power supply current has to be reduced in software standby mode, disable D/A conversion by setting the DACR. and DAOE0 bit to 0.

27.4.4 Setting the D/A Converter

When using the D/A converter output voltage as a reference input voltage of the comparator C, set the D/A converter and wait for the D/A converter output settling time (t_{DCONV}) before enabling the comparator. Similarly, before making any changes to the settings of the D/A converter stop the comparator temporarily, and after the changes are made, wait for the D/A converter output settling time before enabling the comparator.

28. Comparator C (CMPC)

28.1 Overview

Comparator C compares a reference input voltage to an analog input voltage.
The comparison result can be read by software and output externally, and an interrupt request can be generated upon any changes to the comparison result.
The reference input voltage of comparator C can be selected from the input from the CVREFC0 pin or the output from on-chip D/A converter 0.
There are four analog inputs, one of which is to be selected.
Table 28.1 lists the specification of comparator C, Figure 28.1 shows a block diagram of comparator C, and Table 28.2 shows a comparator C pin configuration.
Table 28.1 lists the specification of comparator C, Figure 28.1 shows a block diagram of comparator C, Table 28.2 shows a comparator C pin configuration, and Table 28.3 shows the analog input pin connections for comparator C.
In this section, “PCLK” is used to refer to PCLKB.

Table 28.1 Comparator C Specifications

Item	Specification
Number of channels	Three (comparator C0 to comparator C2)
Analog input voltages	Input voltage from the CMPCnm pin (n = channel number; m = 0 to 3)
Reference input voltage	Input voltage from the CVREFC0 pin or on-chip D/A converter 0 output voltage
Comparison result	The comparison result can be output externally.
Digital filter function	• One of three sampling periods can be selected. • The filter function can also be disabled. • A noise-filtered signal can be used to generate the interrupt request output, POE source output, and the signal can be used to read the comparison result via registers.
Interrupt request	• An interrupt request is generated upon detecting a valid edge of the comparison result. • A valid edge can be selected from a rising or a falling edge or both edges.
Low power consumption function	Module stop state can be set.

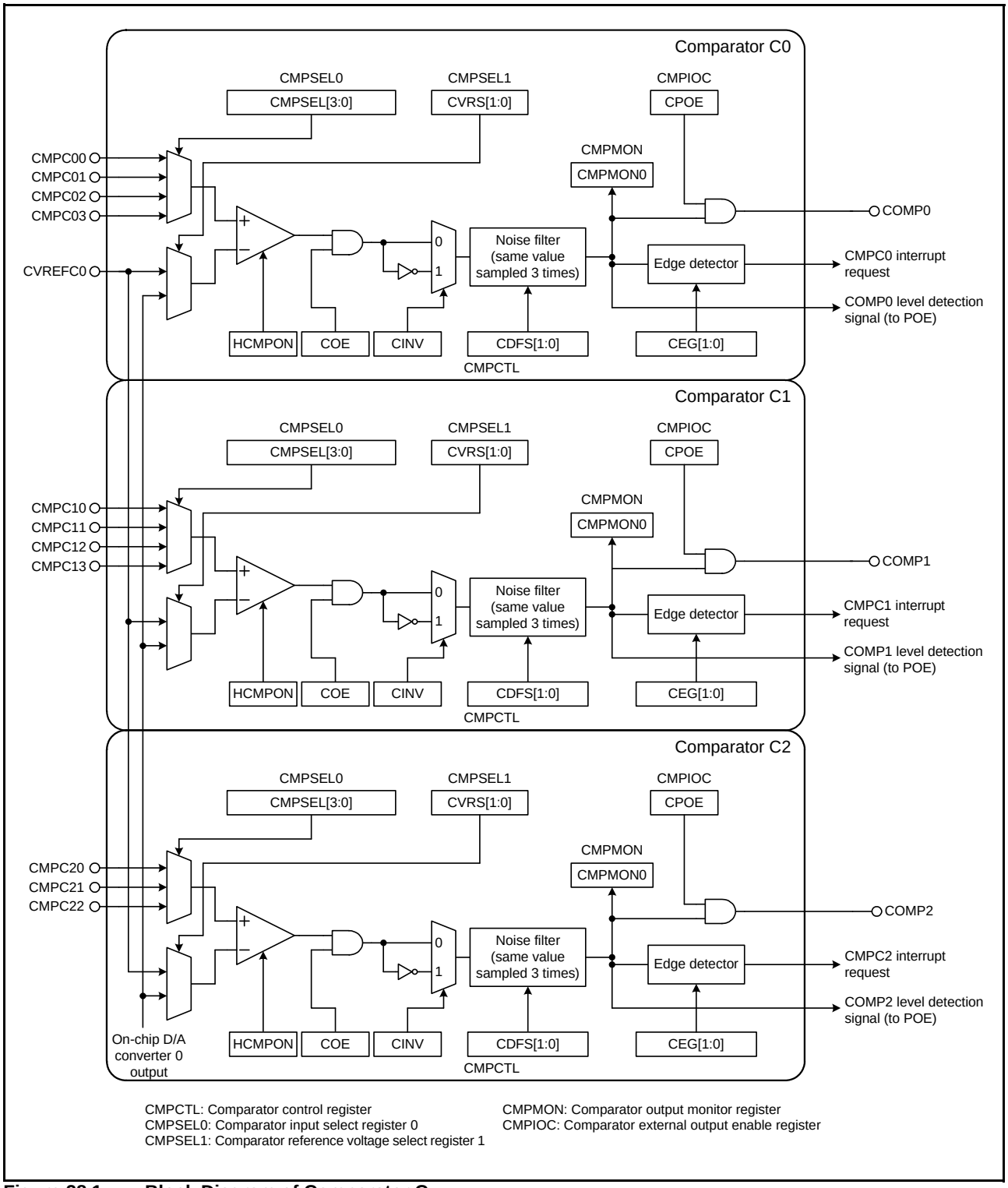


Figure 28.1 Block Diagram of Comparator C

Table 28.2 Comparator C Pin Configuration

Pin Name	I/O	Function
CMPC00, CMPC01, CMPC02, CMPC03	Input	Comparator C0 analog input pins
CMPC10, CMPC11, CMPC12, CMPC13	Input	Comparator C1 analog input pins
CMPC20, CMPC21, CMPC22	Input	Comparator C2 analog input pins
CVREFC0	Input	Reference input voltage pin 0
COMP0	Output	Comparator C0 comparison result output pin
COMP1	Output	Comparator C1 comparison result output pin
COMP2	Output	Comparator C2 comparison result output pin

Table 28.3 Analog Input Pin Connections for Comparator C

Analog Input Pin	Connection
CMPC00	AN000 pin
CMPC01	Programmable gain amplifier output for AN000 pin
CMPC02	AN003 pin
CMPC03	AN006 pin
CMPC10	AN001 pin
CMPC11	Programmable gain amplifier output for AN001 pin
CMPC12	AN004 pin
CMPC13	AN007 pin
CMPC20	AN002 pin
CMPC21	Programmable gain amplifier output for AN002 pin
CMPC22	AN005 pin

28.2 Register Descriptions

28.2.1 Comparator Control Register (CMPCTL)

Address(es): CMPC0.CMPCTL 000A 0C80h, CMPC1.CMPCTL 000A 0CA0h, CMPC2.CMPCTL 000A 0CC0h

	b7	b6	b5	b4	b3	b2	b1	b0
	HCMPON	CDFS[1:0]		CEG[1:0]		—	COE	CINV
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CINV	Comparator Output Polarity Select *1, *4	0: Comparator output not inverted 1: Comparator output inverted	R/W
b1	COE	Comparator Output Enable	0: Comparator output disabled (the output signal is fixed to 0) 1: Comparator output enabled	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4, b3	CEG[1:0]	Comparator Edge Interrupt Detection Select	b4 b3 0 0: Interrupt request is not generated. 0 1: Rising edge 1 0: Falling edge 1 1: Both edges	R/W
b6, b5	CDFS[1:0]	Noise Filter Sampling Select*1, *2, *4	b6 b5 0 0: Noise filter not used 0 1: Sampling frequency is PCLK/8. 1 0: Sampling frequency is PCLK/16. 1 1: Sampling frequency is PCLK/32.	R/W
b7	HCMPON	Comparator Operation Enable*3	0: Operation stopped (the output signal is fixed to 0) 1: Operation enabled (input to the comparator pins is enabled)	R/W

Note 1. Rewrite the CDFS[1:0] and CINV bits only after disabling the comparator output (COE bit = 0).
Note 2. If the CDFS[1:0] bits are changed from 00b (noise filter not used) to a value other than 00b (noise filter used), allow four sampling times to elapse until the filter output is updated, and then use the CMPCn interrupt request output.
Note 3. The operation stabilization wait time is required after enabling comparator operation (HCMPON bit = 1). As for the value, refer to section 32, Electrical Characteristics.
Note 4. Rewriting the CINV bit or CDFS[1:0] bits may generate a CMPCn interrupt request and POE source. Before changing these bits, set the registers in the POE so that comparator output is not used for output disabling control. After changing these bits, also set the corresponding interrupt status flag (IR) in the interrupt request register and the POE comparator channel n output detection flag (n = 0 to 2) to 0.

CEG[1:0] Bits (Comparator Edge Interrupt Detection Select)

These bits select which edge of comparator output signal is used to generate an interrupt request.

The valid edge is set for the signal after the comparator polarity is selected by the CINV bit and the filter is selected by CDFS[1:0] bits.

28.2.2 Comparator Input Select Register (CMPSEL0)

Address(es): CMPC0.CMPSEL0 000A 0C84h, CMPC1.CMPSEL0 000A 0CA4h, CMPC2.CMPSEL0 000A 0CC4h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	CMPSEL[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CMPSEL[3:0]	Comparator Input Select*1	• Comparator C0 b3 b0 0 0 0 0 : No input 0 0 0 1 : CMPC00 selected 0 0 1 0 : CMPC01 selected 0 1 0 0 : CMPC02 selected 1 0 0 0 : CMPC03 selected Settings other than above are prohibited. • Comparator C1 b3 b0 0 0 0 0 : No input 0 0 0 1 : CMPC10 selected 0 0 1 0 : CMPC11 selected 0 1 0 0 : CMPC12 selected 1 0 0 0 : CMPC13 selected Settings other than above are prohibited. • Comparator C2 b3 b0 0 0 0 0 : No input 0 0 0 1 : CMPC20 selected 0 0 1 0 : CMPC21 selected 0 1 0 0 : CMPC22 selected Settings other than above are prohibited.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Rewrite the CMPSEL[3:0] bits in the following procedure. Writing a value other than 0000b while the value of these bits is not 0000b is invalid. Writing 1 to two or more bits is also invalid. In both cases, the previous value is retained.
(1) Set the CMPCTL.COE bit to 0.
(2) Set the CMPSEL[3:0] bits to 0000b.
(3) Set a new value to the CMPSEL[3:0] bits (with 1 set in only one of the bits).
(4) Wait for the stabilization time for input selection. As for the value, refer to section 32, Electrical Characteristics.
(5) Set the CMPCTL.COE bit to 1.
(6) Set the corresponding interrupt status flag (IR) in the interrupt request register to 0.

28.2.3 Comparator Reference Voltage Select Register (CMPSEL1)

Address(es): CMPC0.CMPSEL1 000A 0C88h, CMPC1.CMPSEL1 000A 0CA8h, CMPC2.CMPSEL1 000A 0CC8h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CVRS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	CVRS[1:0]	Reference Input Voltage Select *1, *2	b1 b0 0 0: No input 0 1: Input voltage to the CVREFC0 pin selected as reference input voltage 1 0: On-chip D/A converter 0 output voltage selected as reference input voltage Settings other than above are prohibited.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the on-chip D/A converter output voltage is used, set the D/A converter for generating comparator C reference voltage before enabling comparator operation (CMPCTL.HCMPON bit = 1). For details on setting the D/A converter, refer to section 27, D/A Converter for Generating Comparator C Reference Voltage (DA).

Note 2. Rewrite the CVRS[1:0] bits in the following procedure. Be sure to set the CVRS[1:0] bits to 00b before changing the set value. Direct rewriting from 01b to 10b or 10b to 01b, for example, will be ignored even if it is tried.
(1) Set the CMPCTL.COE bit to 0.
(2) Set the CVRS[1:0] bits to 00b.
(3) Set a new value to the CVRS[1:0] bits (with 1 set in only one of the bits).
(4) Wait for the stabilization time for input selection. As for the value, refer to section 32, Electrical Characteristics.
(5) Set the CMPCTL.COE bit to 1.
(6) Set the corresponding interrupt status flag (IR) in the interrupt request register to 0.

28.2.4 Comparator Output Monitor Register (CMPMON)

Address(es): CMPC0.CMPMON 000A 0C8Ch, CMPC1.CMPMON 000A 0CACh, CMPC2.CMPMON 000A 0CCCh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CMPMON0
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CMPMON0	Comparator Output Monitor Flag *1	0: Comparator output is 0. 1: Comparator output is 1.	R
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When comparator operation is enabled (CMPCTL.HCMPON and COE bits are 1) while the noise filter is disabled (CMPCTL.CDFS[1:0] bits are 00b), read the CMPMON0 bit twice and use the value only when the results match.

28.2.5 Comparator External Output Enable Register (CMPIOC)

Address(es): CMPC0.CMPIOC 000A 0C90h, CMPC1.CMPIOC 000A 0CB0h, CMPC2.CMPIOC 000A 0CD0h

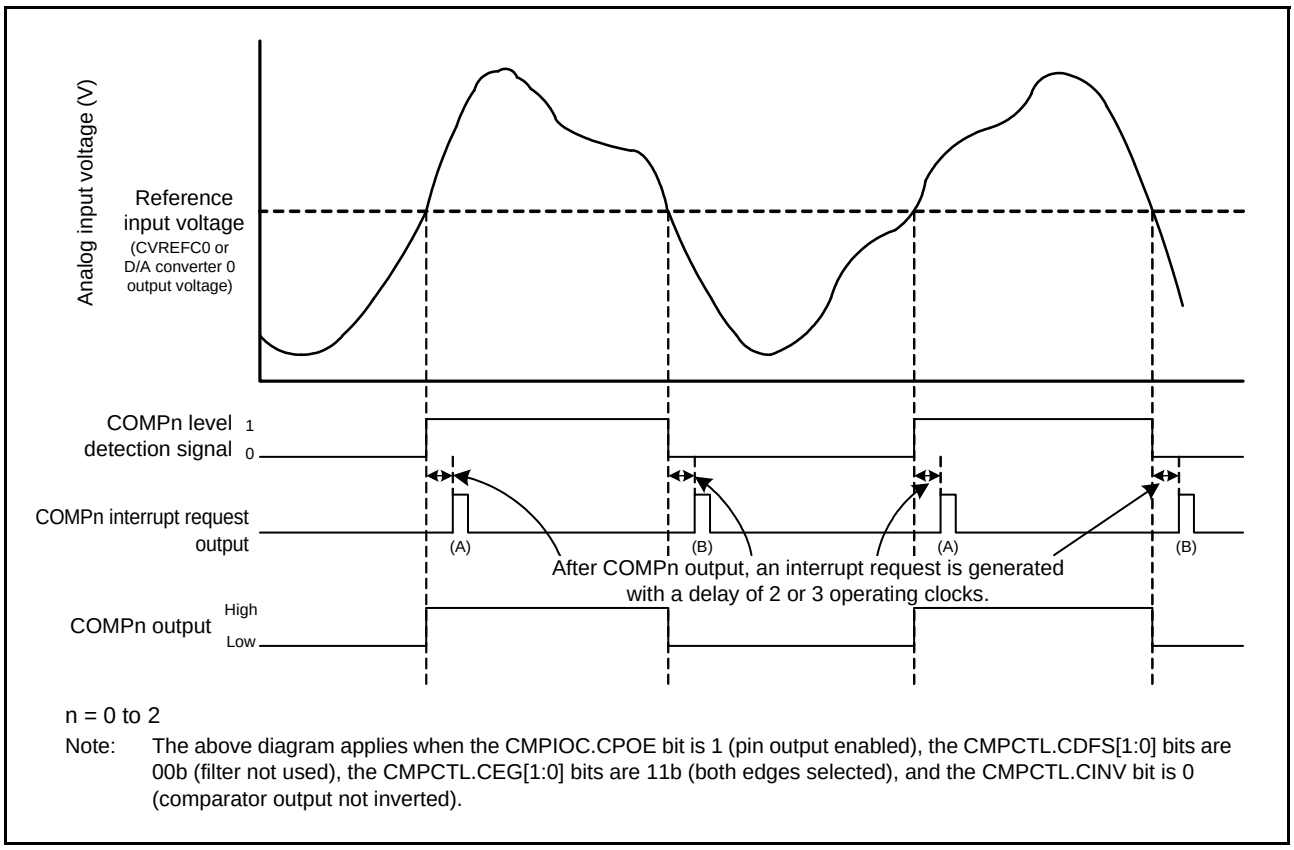
	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	CPOE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CPOE	External Pin Output Enable	Comparison result by the comparator is output to an external pin. 0: Output to the comparator external pin is disabled (the output signal is fixed to low) 1: Output to the comparator external pin is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

28.3 Operation

28.3.1 Comparator Operation Example

Figure 28.2 shows an operation example of the comparator. The COMPn level detection signal (n = 0 to 2) becomes high when the analog input voltage is higher than the reference input voltage, and the COMPn level detection signal becomes low when the analog input voltage is lower than the reference input voltage (when the CMPCTL.CINV bit is 0). When the CPOE bit in the corresponding CMPIOC register is 1, the COMPn level detection signal is output from the COMPn pin. Interrupt request is output in response to changes in the comparator output.



Note: The above diagram applies when the CMPIOC.CPOE bit is 1 (pin output enabled), the CMPCTL.CDFS[1:0] bits are 00b (filter not used), the CMPCTL.CEG[1:0] bits are 11b (both edges selected), and the CMPCTL.CINV bit is 0 (comparator output not inverted).

Figure 28.2 Comparator Operation Example

28.3.2 Noise Filter

Comparator C contains a noise filter. The sampling clock can be selected by the CMPCTL.CDFS[1:0] bits.
The comparator output signal is sampled every sampling clock, and if the same value is sampled three times, that value is determined as the noise filter output at the next sampling clock.
Figure 28.3 shows the configuration of the noise filter and edge detector and Figure 28.4 shows an example of the comparator noise filter and interrupt operation.

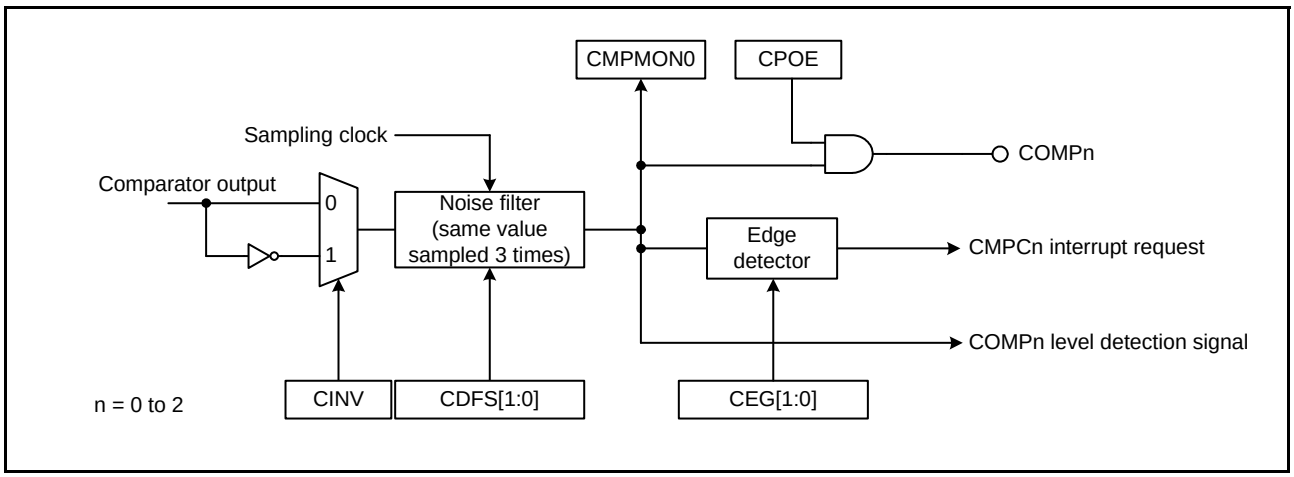
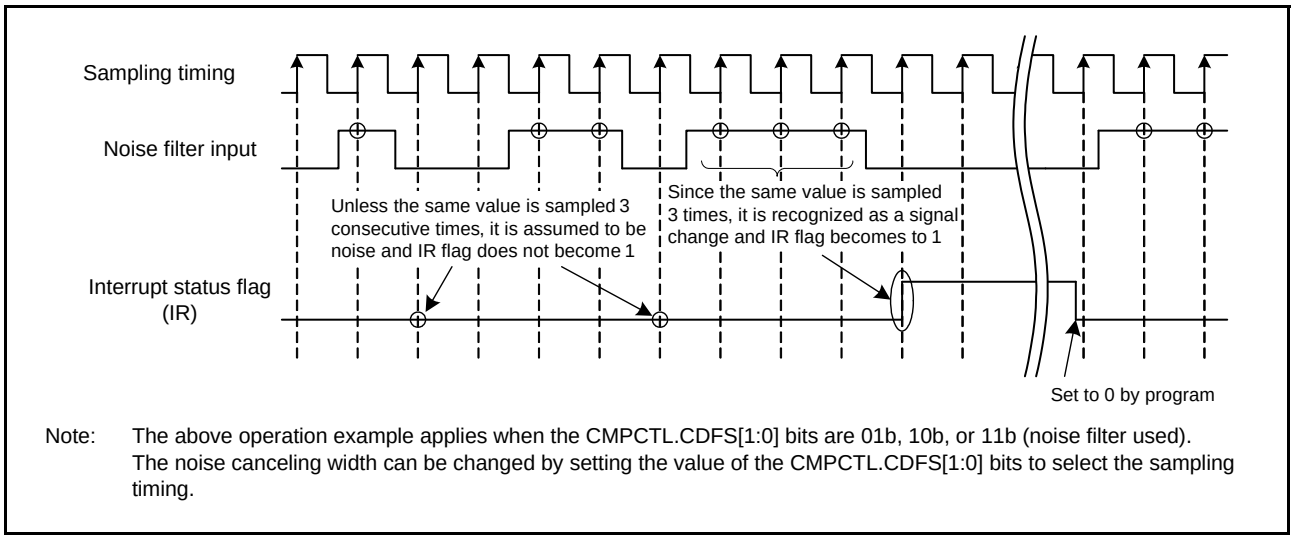


Figure 28.3 Noise Filter and Edge Detector Configuration



Note: The above operation example applies when the CMPCTL.CDFS[1:0] bits are 01b, 10b, or 11b (noise filter used).
The noise canceling width can be changed by setting the value of the CMPCTL.CDFS[1:0] bits to select the sampling timing.

Figure 28.4 Noise Filter and Interrupt Operation Example

28.3.3 Interrupts

Comparator C generates an interrupt request upon detecting any changes in the comparison result.
When using the CMPCn interrupt, set at least one of bits CMPCTL.CEG[1:0] to 1 (to a value other than 00b (interrupt request is not generated)).
To use the CMPCn interrupt, use the following setting procedure. Note that steps (1), (2), and (3) can be set in any order.

(1) When using the on-chip D/A converter output voltage as the reference input voltage, set the D/A converter for generating comparator C reference voltage and enable operation.
(2) Set the CMPSEL0 or CMPSEL1 register to set the input of the comparator.
(3) Set the CMPCTL.CINV and CDFS[1:0] bits to select inversion or non-inversion processing and the sampling timing of the noise filter.
(4) Enable the edge for interrupt detection (set the CMPCTL.CEG[1:0] bits to a value other than 00b).
(5) Enable input of the comparator (set the CMPCTL.HCMPON bit to 1) and wait for the time until the comparator operation is stabilized. As for the value, refer to section 32, Electrical Characteristics.
(6) Enable output of the comparator (set the CMPCTL.COE bit to 1).

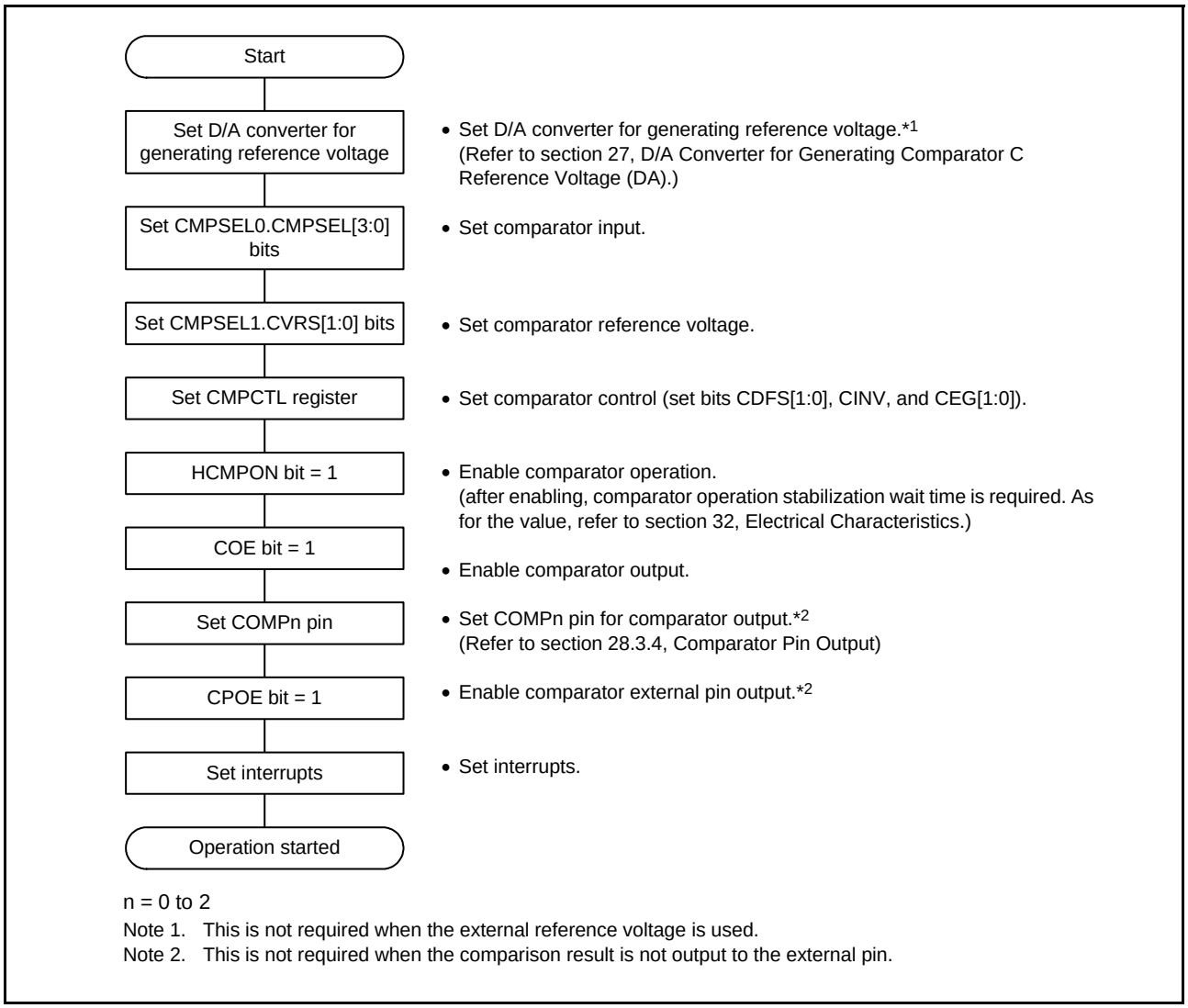
28.3.4 Comparator Pin Output

The comparison results can be output to the COMPn pins (n = 0 to 2). The CMPCTL.CINV bit can be used to set the output polarity (non-inverted output or inverted output), and the CMPIOC.CPOE bit can be used to enable or disable the output.
To output the comparison result to the external pin COMPn, use the following setting procedure. Note that the ports are set to input after reset.

(1) Execute steps (1) to (3) and steps (5) and (6) shown in section 28.3.3, Interrupts.
(2) Output the comparison result by the comparator to an external pin (set the CMPIOC.CPOE bit to 1).
(3) Set the port mode register and the pin function control register corresponding to each comparator output pin.

28.3.5 Comparator Setting Flowchart

Figure 28.5 shows the flowchart for setting the comparator-related registers.



n = 0 to 2

Note 1. This is not required when the external reference voltage is used.

Note 2. This is not required when the comparison result is not output to the external pin.

Figure 28.5 Comparator Operation Setting Flowchart

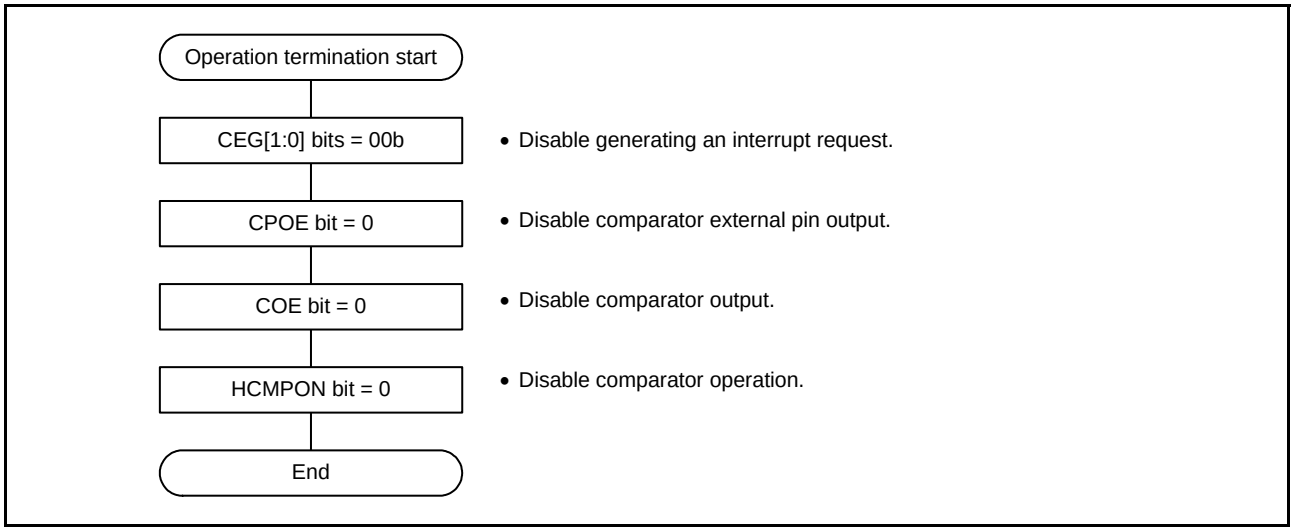


Figure 28.6 Comparator Operation Termination Flowchart

28.4 Usage Notes

28.4.1 Module Stop Function Setting

Operation of comparator C can be disabled or enabled using module stop control register B (MSTPCRB). After the reset, comparator C is halted. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

28.4.2 Comparator C Operation in Module Stop State

When the module stop state is entered while comparator C is operating, analog circuits in the comparator C is not stopped and the analog power supply current is the same as that when comparator C is being used. If the analog power supply current needs to be reduced in the module stop state, set the CMPCTL.HCMPON bit to 0 to stop comparator C.

28.4.3 Comparator C Operation in Software Standby Mode

When software standby mode is entered while comparator C is operating, analog circuits in the comparator C is not stopped and the analog power supply current is the same as that when comparator C is being used. If the analog power supply current needs to be reduced in software standby mode, set the CMPCTL.HCMPON bit to 0 to stop comparator C.

28.4.4 Comparator Operation while the 12-Bit A/D Convertor is in the Module-Stop State

The same module stop signal controls the programmable gain amplifiers (PGAs) and the 12-bit A/D convertor. The comparison of PGA output for the following pins is not possible while the 12-bit A/D convertor is in the module stop state.

- PGA output for AN000 pin
- PGA output for AN001 pin
- PGA output for AN002 pin

The comparison for the following analog input pins is possible since they are directly connected to the comparator, even if the 12-bit A/D convertor is in the module-stop state.

- AN000 pin
- AN001 pin
- AN002 pin
- AN003 pin
- AN004 pin
- AN005 pin
- AN006 pin
- AN007 pin

28.4.5 Setting the D/A Converter for Generating Reference Voltage

Set the D/A converter for generating reference voltage and wait for the D/A converter output settling time before enabling the comparator. Similarly, before making any changes to the settings of the D/A converter stop the comparator temporarily, and after the changes are made, wait for the D/A converter output settling time before enabling the comparator.

29. Data Operation Circuit (DOC)

29.1 Overview

The data operation circuit (DOC) is used to compare, add, or subtract 16-bit values.
Table 29.1 lists the specifications of the DOC and Figure 29.1 is a block diagram of the DOC.
An interrupt can be generated if the result of 16-bit comparison meets one of the set interrupt conditions.

Table 29.1 DOC Specifications

Item	Description
Data operation function	16-bit data comparison, addition, and subtraction
Lower power consumption function	The DOC can be placed in a module-stop state.
Interrupts	• The result of data comparison meets the detection condition. • The result of data addition is greater than FFFFh, which is an overflow. • The result of data subtraction is less than 0000h, which is an underflow.

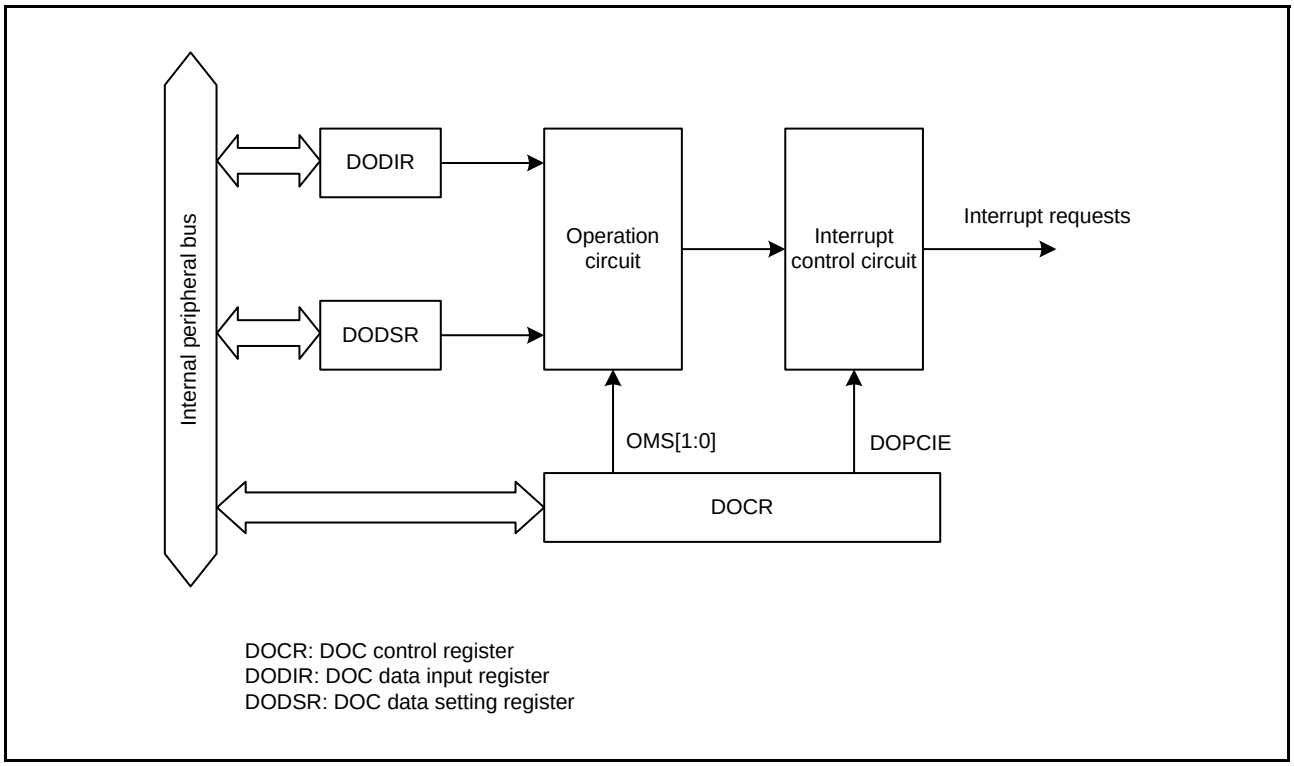


Figure 29.1 DOC Block Diagram

29.2 Register Descriptions

29.2.1 DOC Control Register (DOCR)

Address(es): DOC.DOCR 0008 B080h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	DOPCFCL	DOPCF	DOPCIE	—	DCSEL	OMS[1:0]	
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	OMS[1:0]	Operating Mode Select	b1 b0 0 0: Data comparison mode 0 1: Data addition mode 1 0: Data subtraction mode 1 1: Setting prohibited	R/W
b2	DCSEL	Detection Condition Select*1	0: ‘Not equal to’ is to be detected. 1: ‘Equal to’ is to be detected.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	DOPCIE	Data Operation Circuit Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/W
b5	DOPCF	Data Operation Result Flag	Indicates the result of an operation.	R
b6	DOPCFCL	Data Operation Result Clear	0: Retain the value of the DOPCF flag. 1: Clears the DOPCF flag.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Valid only when data comparison mode is selected.

The DOCR register specifies the operation of DOC, or enabling or disabling of the interrupt.

OMS[1:0] Bits (Operating Mode Select)

These bits select the operating mode of the DOC.

DCSEL Bit (Detection Condition Select)

This bit is valid only when data comparison mode is selected.

This bit selects the condition for detection in data comparison mode.

DOPCIE Bit (Data Operation Circuit Interrupt Enable)

Setting this bit to 1 enables interrupts from the DOC.

DOPCF Flag (Data Operation Result Flag)

[Setting conditions]

- The condition selected by the DCSEL bit is met
- A result of data addition is greater than FFFFh
- A result of data subtraction is less than 0000h

[Clearing condition]

- Writing 1 to the DOPCFCL bit

DOPCFCL Bit (Data Operation Result Clear)

Writing 1 to this bit clears the DOPCF flag.

This bit is read as 0.

29.2.2 DOC Data Input Register (DODIR)

Address(es): DOC.DODIR 0008 B082h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DODIR register is a readable and writable register that holds values for use in operations.

29.2.3 DOC Data Setting Register (DODSR)

Address(es): DOC.DODSR 0008 B084h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The DODSR register is a readable and writable register that holds values for use in comparison or the results of other operations.

In data comparison mode, store the standard value for use in comparison in this register.

In data addition or data subtraction mode, this register holds the results of operations.

29.3 Operation

29.3.1 Data Comparison Mode

Figure 29.2 shows an example of the steps involved in data comparison mode operation by the DOC.
An example of operation when DCSEL is set to 0 (‘not equal to’ is to be detected as the result of data comparison) is shown below.

(1) Writing 00b to the DOCR.OMS[1:0] bits places the DOC in the data comparison mode.
(2) Specify the standard value for comparison in the DODSR register.
(3) Write the value for comparison in the DODIR register.
(4) Write all values for use in comparison to the DODIR register.
(5) If the value written to the DODIR register is not equal to the value set in the DODSR register, the DOCR.DOPCF flag becomes 1. If the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also issued.

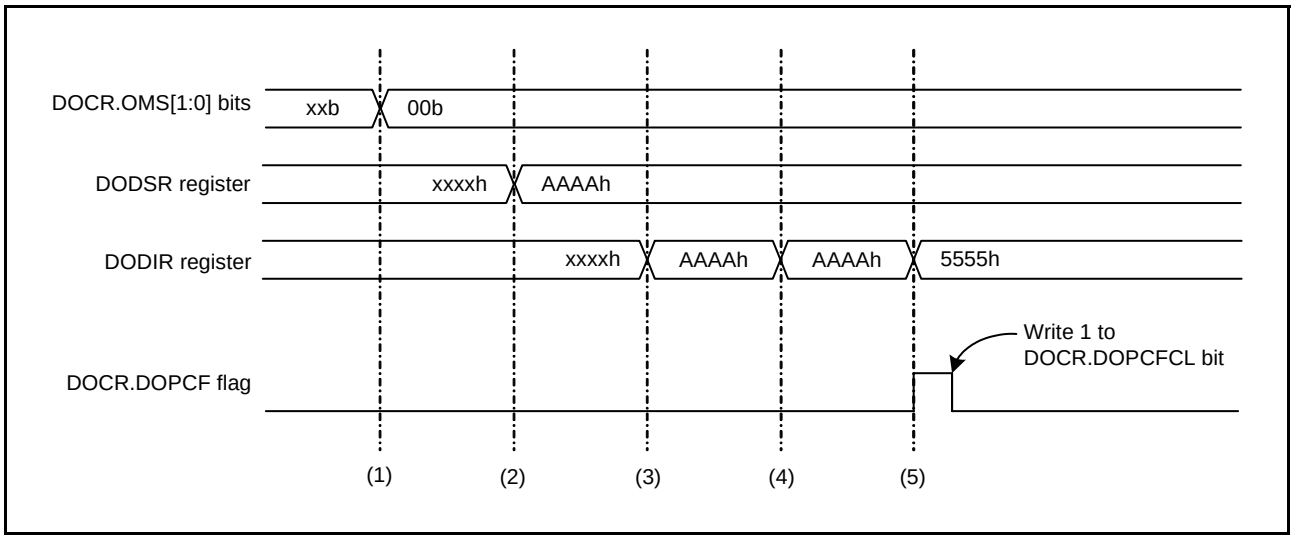


Figure 29.2 Example of Operation in Data Comparison Mode

29.3.2 Data Addition Mode

Figure 29.3 shows an example of the steps involved in data addition mode operation by the DOC.

(1) Writing 01b to the DOCR.OMS[1:0] bits selects data addition mode.
(2) Set the initial value in the DODSR register.
(3) Write the value for addition in the DODIR register. The result of the operation is stored in DODSR.
(4) Write all values for use in addition to the DODIR register.
(5) If the result of the operation is greater than FFFFh, the DOCR.DOPCF flag becomes 1. If the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also issued.

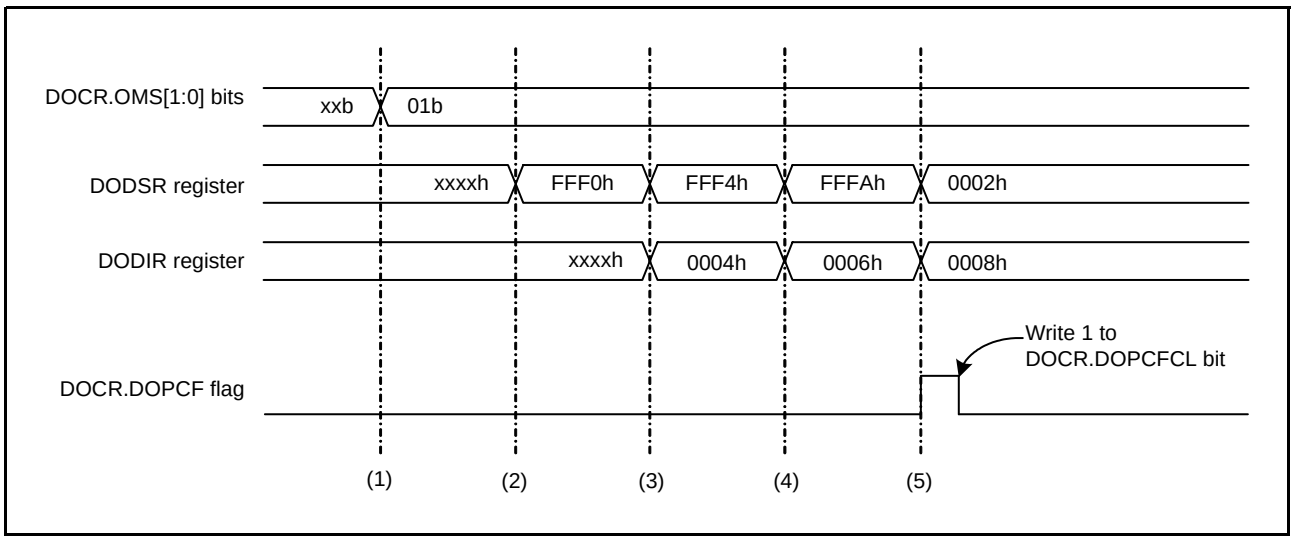


Figure 29.3 Example of Operation in Data Addition Mode

29.3.3 Data Subtraction Mode

Figure 29.4 shows an example of the steps involved in data subtraction mode operation by the DOC.

(1) Writing 10b to the DOCR.OMS[1:0] bits selects data subtraction mode.
(2) Set the initial value in the DODSR register.
(3) Write the value for subtraction in the DODIR register. The result of the operation is stored in DODSR.
(4) Write all values for use in subtraction to the DODIR register.
(5) If the result of the operation is less than 0000h, the DOCR.DOPCF flag becomes 1. If the DOCR.DOPCIE bit is 1, a data operation circuit interrupt is also issued.

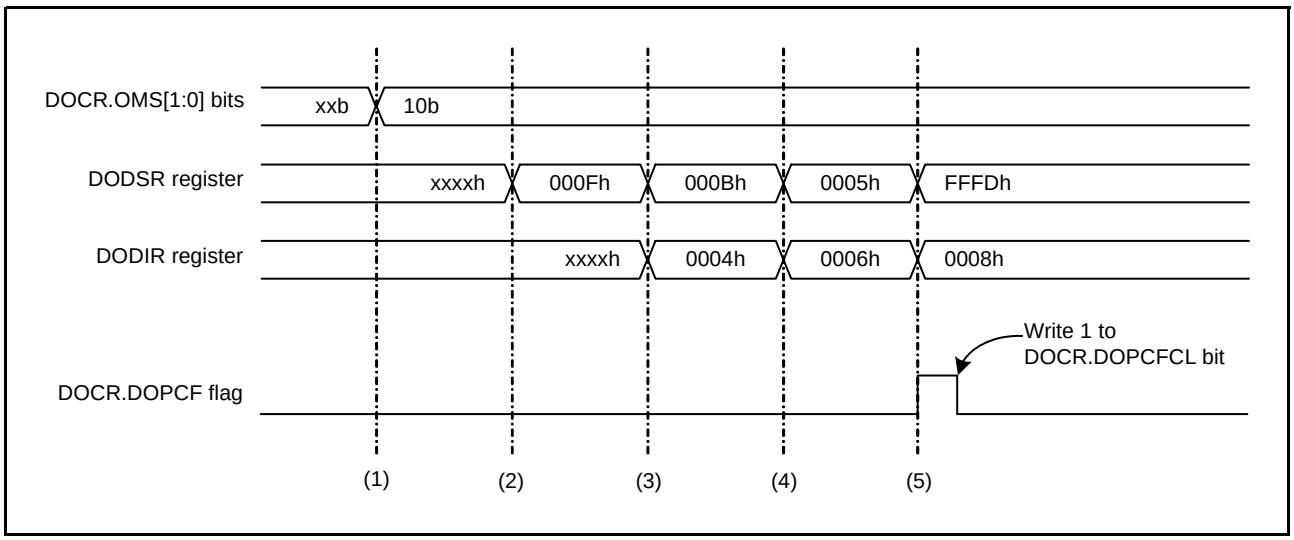


Figure 29.4 Example of Operation in Data Subtraction Mode

29.4 Interrupt Requests

The data operation circuit interrupt (DOPCI) is the interrupt request generated by the DOC. The DOCR.DOPCF flag becomes 1 when the interrupt source condition is satisfied.
Table 29.2 lists the details of the interrupt request.

Table 29.2 Interrupt Request from DOC

Interrupt Request	Data Operation Result Flag	Interrupt Generation Timing
Data operation circuit interrupt (DOPCI)	DOPCF	• The result of data comparison meets the detection condition. • The result of data addition is greater than FFFFh. • The result of data subtraction is less than 0000h.

29.5 Usage Note

29.5.1 Module Stop Function Setting

Operation of the DOC can be enabled or disabled by setting the MSTPB6 bit in module stop control register B (MSTPCRB). The DOC is initially disabled after a reset. Register access is enabled by releasing the module stop state. For details, refer to section 11, Low Power Consumption.

30. RAM

This MCU has an on-chip high-speed static RAM.

30.1 Overview

Table 30.1 lists the specifications of the RAM.

Table 30.1 Specifications of RAM

Item	Description
RAM capacity	12 Kbytes
RAM address	RAM0: 0000 0000h to 0000 2FFFh
Access	• Single-cycle access is possible for both reading and writing. • On-chip RAM can be enabled or disabled.*1
Low power consumption function	The module stop state is selectable for RAM0.

Note 1. Selectable by the RAME bit in SYSCR1. For details on SYSCR1, see section 3.2.2, System Control Register 1 (SYSCR1).

30.2 Operation

30.2.1 Low Power Consumption Function

Power consumption can be reduced by setting module stop control register C (MSTPCRC) to stop supply of the clock signal to the RAM.
Setting the MSTPCRC.MSTPC0 bit to 1 stops supply of the clock signal to RAM0.
Stopping supply of the clock signal places the RAM0 in the module stop state. The RAM operates after initialization by a reset.
The RAM is not accessible in the module stop state. Do not allow transitions to the module stop state while access to RAM is in progress.
For details on the MSTPCRC register, see section 11, Low Power Consumption.

31. Flash Memory (FLASH)

This MCU has packages with 64 and 128 Kbyte flash memory (ROM) for storing code and 4-Kbyte flash memory (E2 DataFlash) for storing data.
In this section, “PCLK” is used to refer to PCLKB.

31.1 Overview

Table 31.1 lists the flash memory specifications.
Table 31.6 lists the I/O pins used in boot mode.

Table 31.1 Flash Memory Specifications

Item	Description
Memory space	• User area: Up to 128 Kbytes • Data area: 4 Kbytes • Extra area: Stores the start-up area information, access window information, and unique ID
Software commands	• The following commands are implemented: Program, blank check, block erase, and unique ID read • The following commands are implemented for programming the extra area: Start-up area information program and access window information program
Value after erasure	• ROM: FFh • E2 DataFlash: FFh
Interrupt	An interrupt (FRDYI) is generated upon completion of software command processing or forced stop processing.
On-board programming	Boot mode (SCI Interface)*1 • Channel 1 of the serial communications interface (SCI1) is used for asynchronous serial communication. • The user area and data area are rewritable. Boot mode (FINE interface)*1 • The FINE is used. • The user area and data area are rewritable. Self-programming in single-chip mode • The user area and data area are rewritable using the flash rewrite routine in the user program.
Off-board programming	The user area and data area are rewritable using a flash programmer compatible with this MCU.
ID code protection	• Connection with the serial programmer can be enabled or disabled using ID codes in boot mode. • Connection with the on-chip debugging emulator can be enabled or disabled using ID codes.
Start-up program protection	This function is used to safely rewrite block 0 to block 15.
Area protection	This function enables rewriting only the selected blocks in the user area and disables the other blocks during self-programming.
Background Operation (BGO)	Programs on the ROM can be executed while rewriting the E2 DataFlash.

Note 1. Refer to the manual of each serial programmer and “Renesas Flash Programmer Flash memory programming software User’s Manual” for more details.

31.2 ROM Area and Block Configuration

The maximum ROM size of this MCU is 128 Kbytes. The ROM area is divided into blocks. A block is 1-Kbyte area. When executing the block erase command, the memory is erased by the block. Figure 31.1 shows the ROM area and block configuration.

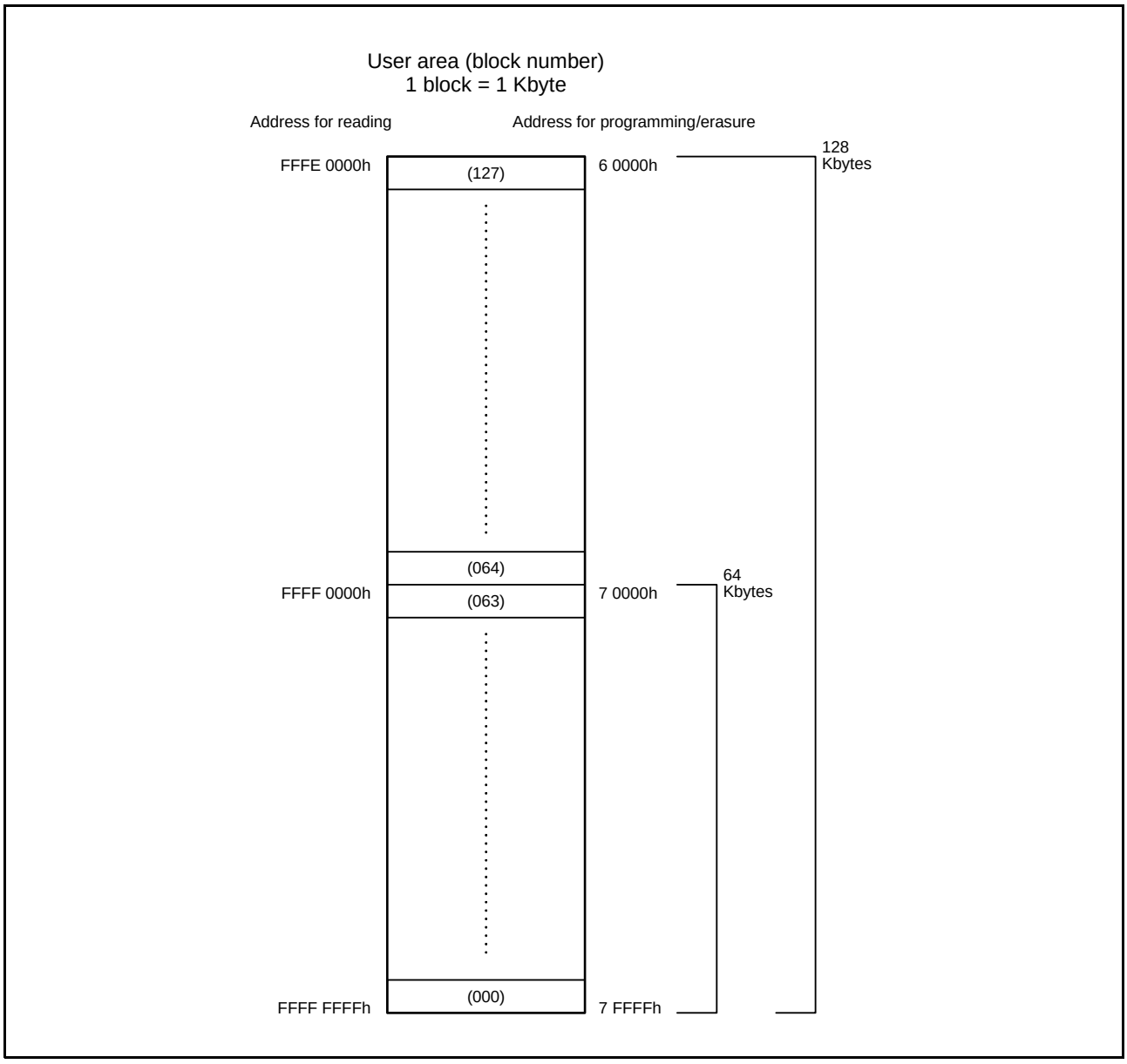


Figure 31.1 ROM Area and Block Configuration

Table 31.2 Correspondence Between ROM Capacity and Addresses for Reading

ROM Capacity	Addresses for Reading
128 Kbytes	FFFE 0000h to FFFF FFFFh
64 Kbytes	FFFF 0000h to FFFF FFFFh

31.3 E2 DataFlash Area and Block Configuration

The E2 DataFlash is 4 Kbytes in the MCU. The E2 DataFlash is divided into blocks and erased in block units. Figure 31.2 shows the E2 DataFlash area and the block configuration.

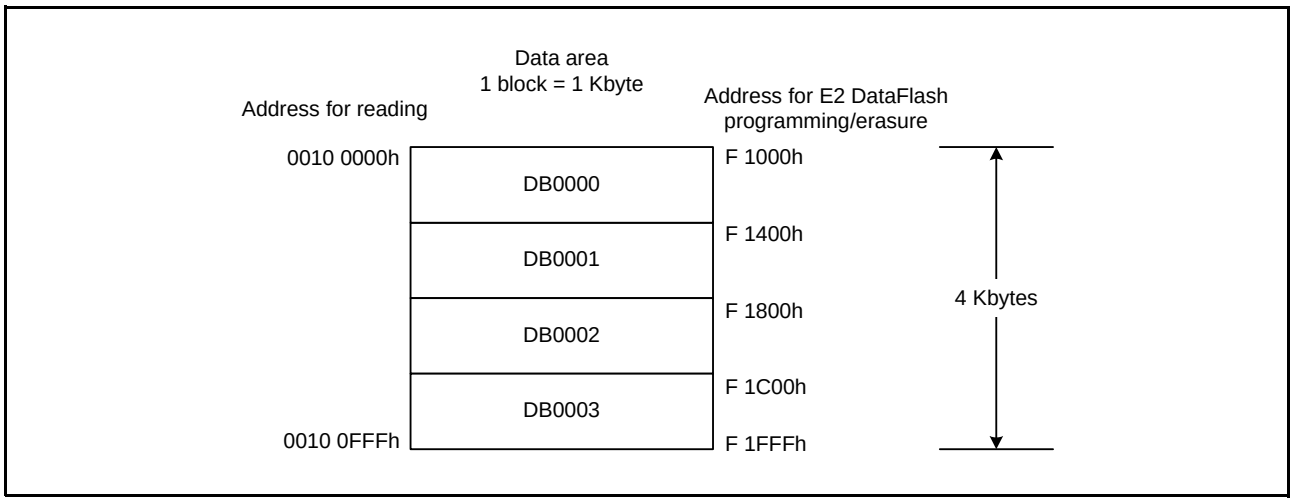


Figure 31.2 E2 DataFlash Area and Block Configuration

31.4 Register Descriptions

31.4.1 E2 DataFlash Control Register (DFLCTL)

Address(es): FLASH.DFLCTL 007F C090h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	DFLEN
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DFLEN	E2 DataFlash Access Enable	0: Access to E2 DataFlash and access to the extra area in P/E mode*1 disabled 1: Access to E2 DataFlash and access to the extra area in P/E mode*1 enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Unique ID read, start-up area information program, and access window information program

The DFLCTL register is used to enable or disable access (read, program, and erase) to the E2 DataFlash and access (unique ID read, start-up area information program, and access window information program) to the extra area in P/E mode.

When reading, programming, and erasing the E2 DataFlash, set the DFLCTL.DFLEN bit to 1 and wait for the E2 DataFlash STOP recovery time (tDSTOP) to elapse before reading the E2 DataFlash and entering E2 DataFlash P/E mode. Do not read the E2 DataFlash or enter E2 DataFlash P/E mode until tDSTOP has elapsed.

Refer to section 31.7.1, Sequencer Modes for details on E2 DataFlash P/E mode. Refer to section 32, Electrical Characteristics for E2 DataFlash STOP recovery time (tDSTOP).

31.4.2 Flash P/E Mode Entry Register (FENTRYR)

Address(es): FLASH.FENTRYR 007F FFB2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	FEKEY[7:0]								FENTRYD	—	—	—	—	—	—	FENTRY0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FENTRY0	ROM P/E Mode Entry 0	0: ROM is in read mode. 1: ROM can be placed in P/E mode.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	FENTRYD	E2 DataFlash P/E Mode Entry	0: E2 DataFlash is in read mode. 1: E2 DataFlash can be placed in P/E mode.	R/W
b15 to b8	FEKEY[7:0]	Key Code	The FEKEY[7:0] bits are used to control rewriting of the FENTRYR register. When rewriting the value of the lower 8 bits, set the FEKEY[7:0] bits to AAh at the same time (write this register in 16 bits). The FEKEY[7:0] bits are read as 00h.	R/W

To rewrite the ROM or E2 DataFlash, the FENTRYD or FENTRY0 bit must be set to 1 to place the ROM or E2 DataFlash in P/E mode.
When returning to read mode, set the FENTRYR register and confirm that its value has been rewritten before reading the ROM or E2 DataFlash.
Refer to section 31.7.1, Sequencer Modes for details on P/E mode and read mode.

FENTRY0 Bit (ROM P/E Mode Entry 0)

This bit is used to place the ROM in P/E mode.
[Setting condition]

- AA01h is written to the FENTRYR register when the FENTRYR register is 0000h.

Note: When entering ROM P/E mode, the instruction fetch address must be transferred to an area other than the ROM so that instruction fetching is not executed to the ROM. Copy necessary instruction code to the internal RAM and jump to the RAM. Note that E2 DataFlash can be rewritten by a program in the ROM.

[Clearing condition]

- AA00h is written to the FENTRYR register.

FENTRYD Bit (E2 DataFlash P/E Mode Entry)

This bit is used to place the E2 DataFlash in P/E mode.
[Setting condition]

- AA80h is written to the FENTRYR register when the FENTRYR register is 0000h.

[Clearing condition]

- AA00h is written to the FENTRYR register.

31.4.3 Protection Unlock Register (FPR)

Address(es): FLASH.FPR 007F C0C0h

	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	x	x	x	x	x	x	x	x

x: Undefined

This write-only register is used to protect the FPMCR register from being rewritten inadvertently when the CPU runs out of control. Writing to the FPMCR register is enabled only when the following procedure is used to access the register.

Procedure to unlock protection

(1) Write A5h to the FPR register.
(2) Write a set value to the FPMCR register.
(3) Write the inverted set value to the FPMCR register.
(4) Write a set value to the FPMCR register again.

When a procedure other than the above is used to write data, the FPSR.PERR flag is set to 1.

31.4.4 Protection Unlock Status Register (FPSR)

Address(es): FLASH.FPSR 007F C0C1h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	PERR
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PERR	Protect Error Flag	0: No error 1: An error occurs.	R
b7 to b1	—	Reserved	These bits are read as 0.	R

PERR Flag (Protect Error Flag)

When the FPMCR register is not accessed as described in the procedure to unlock protection, data is not written to the register and this flag is set to 1.

[Setting condition]

- The FPMCR register is not accessed as described in the procedure to unlock protection.

[Clearing condition]

- The FPMCR register is accessed according to the procedure to unlock protection described in section 31.4.3, Protection Unlock Register (FPR).

31.4.5 Flash P/E Mode Control Register (FPMCR)

Address(es): FLASH.FPMCR 007F FF80h

	b7	b6	b5	b4	b3	b2	b1	b0
	FMS2	LVPE	—	FMS1	RPDIS	—	FMS0	—
Value after reset:	0	0	0	0	1	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b1	FMS0	Flash Operating Mode Select 0	FMS2 FMS1 FMS0 0 0 0: ROM/E2 DataFlash read mode 0 1 0: E2 DataFlash P/E mode 0 1 1: Discharge mode 1 1 0 1: ROM P/E mode 1 1 1: Discharge mode 2 Settings other than above are prohibited.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	RPDIS	ROM P/E Disable	0: ROM programming/erasure enabled 1: ROM programming/erasure disabled	R/W
b4	FMS1	Flash Operating Mode Select 1	See the FMS0 bit.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	LVPE	Low-Voltage P/E Mode Enable	0: Low-voltage P/E mode disabled 1: Low-voltage P/E mode enabled	R/W
b7	FMS2	Flash Operating Mode Select 2	See the FMS0 bit.	R/W

The FPMCR register is used to set the operating mode of the flash memory.
This register is protected. Set its value using the procedure to unlock protection. For details, refer to section 31.4.3, Protection Unlock Register (FPR).
When entering discharge mode 2 or ROM P/E mode, or during either of these modes, an instruction must be executed on the RAM.

FMS0, FMS1, and FMS2 Bits (Flash Operating Mode Select 0 to Flash Operating Mode Select 2)

These bits are used to set the operating mode of the flash memory.

- Transition from read mode to ROM P/E mode
  Set the FMS2 bit = 0, the FMS1 bit = 1, the FMS0 bit = 1, and the RPDIS bit = 0.
  Wait for ROM mode transition wait time 1 (tDIS, refer to section 32, Electrical Characteristics).
  Set the FMS2 bit = 1, the FMS1 bit = 1, the FMS0 bit = 1, and the RPDIS bit = 0.
  Set the FMS2 bit = 1, the FMS1 bit = 0, the FMS0 bit = 1, and the RPDIS bit = 0.
  Wait for ROM mode transition wait time 2 (tMS, refer to section 32, Electrical Characteristics).
- Transition from ROM P/E mode to read mode
  Set the FMS2 bit = 1, the FMS1 bit = 1, the FMS0 bit = 1, and the RPDIS bit = 0.
  Wait for ROM mode transition wait time 1 (tDIS, refer to section 32, Electrical Characteristics).
  Set the FMS2 bit = 0, the FMS1 bit = 1, the FMS0 bit = 1, and the RPDIS bit = 0.
  Set the FMS2 bit = 0, the FMS1 bit = 0, the FMS0 bit = 0, and the RPDIS bit = 1.
  Wait for ROM mode transition wait time 2 (tMS, refer to section 32, Electrical Characteristics).
- Transition from read mode to E2 DataFlash P/E mode
  Set the FMS2 bit = 0, the FMS1 bit = 1, the FMS0 bit = 0, and the RPDIS bit = 0.
- Transition from E2 DataFlash P/E mode to read mode
  Set the FMS2 bit = 0, the FMS1 bit = 0, the FMS0 bit = 0, and the RPDIS bit = 1.

Wait for ROM mode transition wait time 2 (tMS, refer to section 32, Electrical Characteristics).

RPDIS Bit (ROM P/E Disable)

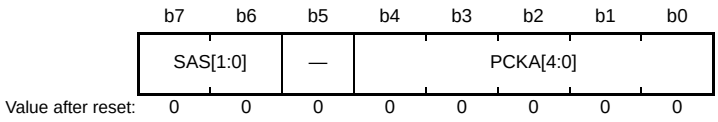
This bit is used to disable the execution of ROM programming/erasure with software.

LVPE Bit (Low-Voltage P/E Mode Enable)

Set this bit to 0 for programming/erasure in high-speed mode, and set this bit to 1 for programming/erasure in middle-speed mode.

31.4.6 Flash Initial Setting Register (FISR)

Address(es): FLASH.FISR 007F C0B6h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PCKA[4:0]	Peripheral Clock Notification	These bits are used to set the frequency of the FlashIF clock (FCLK).	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	SAS[1:0]	Start-Up Area Select	b7 b6 0 x: The start-up area is selected according to the start-up area settings of the extra area. 1 0: The start-up area is switched to the default area temporarily. 1 1: The start-up area is switched to the alternate area temporarily.	R/W

x: Don't care

Data can be written to the FISR register in ROM P/E mode or E2 DataFlash P/E mode.

PCKA[4:0] Bits (Peripheral Clock Notification)

These bits are used to set the frequency of the FlashIF clock (FCLK) when programming/erasing the ROM/E2 DataFlash.

Set the FCLK frequency in the PCKA[4:0] bits before programming/erasure. Do not change the frequency during programming/erasure of the ROM/E2 DataFlash.

- When FCLK is higher than 4 MHz
  Set a rounded-up value for a non-integer frequency.
  For example, set 32 MHz (PCKA[4:0] bits = 11111b) when the frequency is 31.5 MHz.
- When FCLK is 4 MHz or lower
  Do not use a non-integer frequency.
  Use the FCLK at a frequency of 1, 2, 3, or 4 MHz.

Note: When the PCKA[4:0] bits are set to a frequency different from the FCLK, the data in the ROM/E2 DataFlash may be damaged.

Table 31.3 Example of FlashIF Clock Frequency Settings

FlashIF Clock Frequency (MHz)	PCKA[4:0] Bit Setting	FlashIF Clock Frequency (MHz)	PCKA[4:0] Bit Setting	FlashIF Clock Frequency (MHz)	PCKA[4:0] Bit Setting
32	11111b	31	11110b	30	11101b
29	11100b	28	11011b	27	11010b
26	11001b	25	11000b	24	10111b
23	10110b	22	10101b	21	10100b
20	10011b	19	10010b	18	10001b
17	10000b	16	01111b	15	01110b
14	01101b	13	01100b	12	01011b
11	01010b	10	01001b	9	01000b
8	00111b	7	00110b	6	00101b
5	00100b	4	00011b	3	00010b
2	00001b	1	00000b	—	—

SAS[1:0] Bits (Start-Up Area Select)

These bits are used to select the start-up area. To change the start-up area, the following three methods can be used.

- When selecting the start-up area according to the start-up area settings of the extra area
  With the SAS[1:0] bits set to 00b or 01b, the start-up area is selected according to the start-up area settings of the extra area. The settings are enabled after a reset is released.
- When switching the start-up area to the default area temporarily
  When 10b is written to the SAS[1:0] bits, the start-up area is switched to the default area immediately after data is written to the register, regardless of the start-up area settings of the extra area.
  When a reset is generated after this, the area is selected according to the start-up area settings of the extra area.
- When switching the start-up area to the alternative area temporarily
  When 11b is written to the SAS[1:0] bits, the start-up area is switched to the alternative area, regardless of the start-up area settings of the extra area.
  When a reset is generated after this, the area is selected according to the start-up area settings of the extra area.

31.4.7 Flash Reset Register (FRESETR)

Address(es): FLASH.FRESETR 007F FF89h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	FRESET
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	FRESET	Flash Reset	0: Flash control circuit reset is released. 1: Flash control circuit is reset.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FRESET Bit (Flash Reset)

When this bit is set to 1, registers FASR, FSARH, FSARL, FEARH, FEARL, FWBH, FWBL, FCR, and FEXCR are reset. Also, the values of registers FEAMH and FEAML are undefined. Do not access these registers during a reset. To release the reset, set this bit to 0.

Do not write to this register while executing a software command or rewriting the extra area.

31.4.8 Flash Area Select Register (FASR)

Address(es): FLASH.FASR 007F FF81h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	EXS
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EXS	Extra Area Select	0: User area or data area 1: Extra area	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Data can be written to the FASR register in ROM P/E mode or E2 DataFlash P/E mode.

This register is initialized by a reset or setting the FRESETR.FRESET bit to 1.

Data cannot be written to this register while the FRESETR.FRESET bit is 1.

EXS Bit (Extra Area Select)

Set this bit to 1 before issuing a software command (unique ID read, start-up area information program, or access window information program) for the extra area. Set this bit to 0 before issuing a software command (program, blank check, or block erase) for the user area.

After issuing a software command, do not change the value until changing it for issuing the next software command.

31.4.9 Flash Control Register (FCR)

Address(es): FLASH.FCR 007F FF85h

	b7	b6	b5	b4	b3	b2	b1	b0
	OPST	STOP	—	DRC	CMD[3:0]			
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CMD[3:0]	Software Command Setting	b3 b0 0 0 0 1: Program 0 0 1 1: Blank check 0 1 0 0: Block erase 0 1 0 1: Unique ID read Settings other than above are prohibited.*1	R/W
b4	DRC	Data Read Completion	0: Start data read. 1: Complete data read.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	STOP	Forced Processing Stop	When this bit is set to 1, the processing being executed can be forcibly stopped.	R/W
b7	OPST	Processing Start	0: Processing stops. 1: Processing starts.	R/W

Note 1. This does not include set the FCR register to 00h when the FSTATR1.FRDY flag is 1.

Data can be written to the FCR register when in ROM P/E mode and the ROM can be programmed/erased or in E2 DataFlash P/E mode.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
Note that this register cannot be initialized by the FRESETR.FRESET bit while a software command is being executed.

CMD[3:0] Bits (Software Command Setting)

These bits are used to set a software command (program, blank check, block erase, or unique ID read).
The function of each command is described below.

- Program
  Write the value set in registers FWBH and FWBL to the address set in registers FSARH and FSARL.
- Blank check
  Check whether there is data in the area from the address set in registers FSARH and FSARL to the address set in registers FEARH and FEARL. Confirm that data is not programmed in the area. This command does not guarantee whether the area remains erased.
- Block erase
  Erase consecutive areas specified in the flash memory by the blocks. Set the beginning address of the block in registers FSARH and FSARL and the end address in registers FEARH and FEARL.
- Unique ID read
  When executing the unique ID read after setting registers FSARH, FSARL, FEARH, and FEARL to 00h, 0850h, 00h, and 086Fh, respectively, the unique ID is stored in registers FRBH and FRBL sequentially.

DRC Bit (Data Read Completion)

This bit is used with the unique ID read command to control the state of the sequencer.
When issuing the unique ID read command with this bit set to 0, data is read from the address set in registers FSARH and FSARL, and the data is stored in registers FRBH and FRBL.

When issuing the unique ID read command with this bit set to 1 after reading data from registers FRBH and FRBL, the sequencer ends the read cycle and enters the wait state.
When issuing the unique ID read command again with this bit set to 0, the internal address of the sequencer is incremented by 4, and the next data is read.

STOP Bit (Forced Processing Stop)

This bit is used to forcibly stop the processing (blank check or block erase) being executed.
After setting this bit to 1, wait until the FSTATR1.FRDY flag is 1 (processing completed) before setting the OPST bit to 0.

OPST Bit (Processing Start)

This bit is used to execute the command set in the CMD[2:0] bits.
This bit is not set to 0 again even when the processing is completed. Confirm that the FSTATR1.FRDY flag is 1 (processing completed) before setting the OPST bit to 0 again. After that, confirm that the FRDY flag is 0 before executing the next processing.

31.4.10 Flash Extra Area Control Register (FEXCR)

Address(es): FLASH.FEXCR 007F C0B7h

	b7	b6	b5	b4	b3	b2	b1	b0
	OPST	—	—	—	—	CMD[2:0]		
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CMD[2:0]	Software Command Setting	b2 b0 0 0 1: Start-up area information program 0 1 0: Access window information program Settings other than above are prohibited.*1	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	OPST	Processing Start	0: Processing stops. 1: Processing starts.	R/W

Note 1. This does not include set the FEXCR register to 00h when the FSTATR1.EXRDY flag is 1.

Data can be written to the FEXCR register when in ROM P/E mode and the ROM can be programmed/erased.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
Note that this register cannot be initialized by the FRESETR.FRESET bit while a software command is being executed.

CMD[2:0] Bits (Software Command Setting)

These bits are used to set a software command (start-up area information program or access window information program).
The details of each command are described below.

- Start-up area information program
  This command is used to switch the start-up area used for start-up program protection.
  When setting the start-up area to the default area, set the FWBH and FWBL registers to FFFFh, and execute this command.
  When setting the start-up area to the alternative area, set the FWBH register to FFFFh, set the FWBL register to FEFFh, and execute this command.
  When the FWBH and FWBL registers are set to values other than the above, do not execute the start-up area information program.
- Access window information program
  This command is used to set the access window used for area protection.
  Set the access window in block units.
  Specify the access window start address, which is the beginning address of the access window in the FWBL register, specify the access window end address, which is the next address of the last address of the access window in the FWBH register, and issue this command. Set bit 19 to bit 10 of the address for programming/erasure in each register.
  If the same value is set as the start address and end address, all areas can be accessed. Do not set the start address to a value larger than the value of the end address.

OPST Bit (Processing Start)

This bit is used to execute the command set in the CMD[2:0] bits.
This bit is not set to 0 again even when the processing is completed. Confirm that the FSTATR1.EXRDY flag is 1 (processing completed) before setting the OPST bit to 0 again. After that, confirm that the FSTATR1.EXRDY flag is 0

before executing the next processing.
Writing to the extra area is started by writing 1 to the OPST bit. Do not write to the CMD[2:0] bits while a software command is being executed.

31.4.11 Flash Processing Start Address Register H (FSARH)

Address(es): FLASH.FSARH 007F FF84h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—				
Value after reset:	0	0	0	0	0	0	0	0

The FSARH register is used to set the target processing address or the start address of the target processing range in the flash memory when a software command is executed.
Set bit 19 to bit 16 of the flash memory address for programming/erasure in this register.
Data can be written to this register in ROM P/E mode or E2 DataFlash P/E mode.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
If this register is read while executing a software command set by the FEXCR register, an undefined value is read.
Refer to Figure 31.1 and Figure 31.2 for details on the addresses of the flash memory.

31.4.12 Flash Processing Start Address Register L (FSARL)

Address(es): FLASH.FSARL 007F FF82h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FSARL register is used to set the target processing address or the start address of the target processing range in the flash memory when a software command is executed.
Set bit 15 to bit 0 of the flash memory address for programming/erasure in this register.
When the target is the ROM, set bit 1 and bit 0 to 00b.
Data can be written to this register in ROM P/E mode or E2 DataFlash P/E mode.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
If this register is read while executing a software command set by the FEXCR register, an undefined value is read.
Refer to Figure 31.1 and Figure 31.2 for details on the addresses of the flash memory.

31.4.13 Flash Processing End Address Register H (FEARH)

Address(es): FLASH.FEARH 007F FF88h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—				
Value after reset:	0	0	0	0	0	0	0	0

The FEARH register is used to set the end address of the target processing range in the flash memory when a software command is executed.
Set bit 19 to bit 16 of the flash memory address for programming/erasure in this register.
Data can be written to this register in ROM P/E mode or E2 DataFlash P/E mode.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
If this register is read while executing a software command set by the FEXCR register, an undefined value is read.
Refer to Figure 31.1 and Figure 31.2 for details on the addresses of the flash memory.

31.4.14 Flash Processing End Address Register L (FEARL)

Address(es): FLASH.FEARL 007F FF86h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

The FEARH register is used to set the end address of the target range for processing when a software command is executed.
Set bit 15 to bit 0 of the flash memory address for programming/erasure in this register.
When the target is the ROM, set bit 1 and bit 0 to 00b.
Data can be written to this register in ROM P/E mode or E2 DataFlash P/E mode.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
If this register is read while executing a software command set by the FEXCR register, an undefined value is read.
Refer to Figure 31.1 and Figure 31.2 for details on the addresses of the flash memory.

31.4.15 Flash Read Buffer Register H (FRBH)

Address(es): FLASH.FRBH 007F C0C4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to store the upper 2 bytes of the 4-byte data (part of the unique ID) that is read from the extra area when unique ID read is executed.

31.4.16 Flash Read Buffer Register L (FRBL)

Address(es): FLASH.FRBL 007F C0C2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to store the lower 2 bytes of the 4-byte data (part of the unique ID) that is read from the extra area when unique ID read is executed.

31.4.17 Flash Write Buffer Register H (FWBH)

Address(es): FLASH.FWBH 007F FF8Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to set the upper 16 bits of the data for programming the ROM. The data for programming the E2 DataFlash should be specified in the lower 8 bits in the FWBL register.
Data can be written to this register in ROM P/E mode or E2 DataFlash P/E mode.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
The value read from this register is undefined while a software command is being executed.
If this register is read while executing a software command set by the FEXCR register, an undefined value is read.

31.4.18 Flash Write Buffer Register L (FWBL)

Address(es): FLASH.FWBL 007F FF8Ch

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to set the lower 16 bits of the data for programming the ROM or the data for programming the E2 DataFlash. The data for programming the E2 DataFlash should be set in bit 7 to bit 0.
Data can be written to this register in ROM P/E mode or E2 DataFlash P/E mode.
This register is initialized by a reset or setting the FRESETR.FRESET bit to 1. Data cannot be written to this register while the FRESETR.FRESET bit is 1.
If this register is read while executing a software command set by the FEXCR register, an undefined value is read.

31.4.19 Flash Status Register 0 (FSTATR0)

Address(es): FLASH.FSTATR0 007F FF8Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	EILGLERR	ILGLERR	BCERR	—	PRGERR	ERERR
Value after reset:	x	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	ERERR	Erase Error Flag	0: Erasure terminates normally. 1: An error occurs during erasure.	R
b1	PRGERR	Program Error Flag	0: Programming terminates normally. 1: An error occurs during programming.	R
b2	—	Reserved	The read value is undefined.	R
b3	BCERR	Blank Check Error Flag	0: Blank checking terminates normally. 1: An error occurs during blank checking.	R
b4	ILGLERR	Illegal Command Error Flag	0: No illegal software command or illegal access is detected. 1: An illegal command or illegal access is detected.	R
b5	EILGLERR	Extra Area Illegal Command Error Flag	0: No illegal command or illegal access to the extra area is detected. 1: An illegal command or illegal access to the extra area is detected.	R
b7, b6	—	Reserved	The read value is undefined.	R

This register is a status register used to confirm the result of executing a software command. Each error flag is set to 0 when the next software command is executed.

ERERR Flag (Erase Error Flag)

This flag indicates the result of the erase processing for the ROM/E2 DataFlash.
[Setting condition]

- An error occurs during erasure.

[Clearing condition]

- The next software command is executed.
  The value read from this flag is undefined when the FCR.STOP bit is set to 1 (processing is forcibly stopped) during erasure.

PRGERR Flag (Program Error Flag)

This flag indicates the result of the program processing for the ROM/E2 DataFlash.
[Setting condition]

- An error occurs during programming.

[Clearing condition]

- The next software command is executed.

BCERR Flag (Blank Check Error Flag)

This flag indicates the result of the blank check processing for the ROM/E2 DataFlash.
[Setting condition]

- An error occurs during blank checking.

[Clearing condition]

- The next software command is executed.
  The value read from this flag is undefined when the FCR.STOP bit is set to 1 (processing is forcibly stopped) during blank checking.

ILGLERR Flag (Illegal Command Error Flag)

This flag indicates the result of executing a software command.
[Setting conditions]

- Programming/erasure is executed to an area other than the access window range.
- A blank check or block erase command is executed when the set value of registers FSARH and FSARL is larger than the set value of registers FEARH and FEARL.
- Program and block erase commands are executed when the FASR.EXS bit is 1.
- The E2 DataFlash address is set in registers FSARH and FSARL and a software command is executed when the ROM is in P/E mode.
- The ROM address is set in registers FSARH and FSARL and a software command is executed when the E2 DataFlash is in P/E mode.
- The ROM and E2 DataFlash are set to P/E mode and a software command is executed.

[Clearing condition]

- The next software command is executed.

EILGLERR Flag (Extra Area Illegal Command Error Flag)

This flag indicates the result of executing a software command for the extra area.
[Setting condition]

- A software command for the extra area is executed when the FASR.EXS bit is 0.

[Clearing condition]

- The next software command is executed.

31.4.20 Flash Status Register 1 (FSTATR1)

Address(es): FLASH.FSTATR1 007F FF8Bh

	b7	b6	b5	b4	b3	b2	b1	b0
	EXRDY	FRDY	—	—	—	—	DRRDY	—
Value after reset:	0	0	0	0	0	1	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 0.	R
b1	DRRDY	Data Read Ready Flag	0: No valid data in registers FRBH and FRBL 1: Valid data in registers FRBH and FRBL	R
b2	—	Reserved	This bit is read as 1.	R
b5 to b3	—	Reserved	These bits are read as 0.	R
b6	FRDY	Flash Ready Flag	0: Other than below 1: 00h can be written to the FCR register (processing to complete the software command).	R
b7	EXRDY	Extra Area Ready Flag	0: Other than below 1: 00h can be written to the FEXCR register (processing to complete the software command).	R

This register is a status register used to confirm the result of executing a software command. Each flag is set to 0 when the next software command is executed.

DRRDY Flag (Data Read Ready Flag)

This flag is used to check if the valid read data is stored in registers FRBH and FRBL.
When the sequencer stores data read from the flash memory to registers FRBH and FRBL, the DRRDY flag becomes 1.
When issuing the unique ID command with the FCR.DRC bit set to 1, the sequencer ends the read cycle, and the DRRDY flag becomes 0.
Note that, even if issuing the unique ID command with the FCR.DRC bit set to 0 after reading data from the address set in registers FEARH and FEARL, the DRRDY flag does not become 1, but the FRDY flag becomes 1.

FRDY Flag (Flash Ready Flag)

This flag is used to confirm whether a software command is executed.
This flag becomes 1 when processing of the executed software command or the forced stop processing is completed, and this flag becomes 0 when setting the FCR.OPST bit to 0.
Also, an interrupt (FRDYI) is generated when this flag becomes 1.

EXRDY Flag (Extra Area Ready Flag)

This flag is used to confirm whether a software command for the extra area is executed.
This flag is set to 1 when processing of the executed software command is completed, and 0 when the FEXCR.OPST bit is set to 0.
Also, an interrupt (FRDYI) is generated when this flag becomes 1.

31.4.21 Flash Error Address Monitor Register H (FEAMH)

Address(es): FLASH.FEAMH 007F C0BAh

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—				
Value after reset:	0	0	0	0	0	0	0	0

This register is used to check the address where the error has occurred if an error occurs during processing of a software command. This register stores bit 19 to bit 16 of the address where the error has occurred for the program command or blank check command, or it stores bit 19 to bit 16 of the beginning address of the area where the error has occurred for the block erase command.
Since this register value becomes undefined if setting the FRESETR.FRESET bit to 1, read the value before error processing.
If the software command terminates normally, this register stores bit 19 to bit 16 of the end address at execution of the command.
Refer to Figure 31.1 and Figure 31.2 for details on the addresses of the flash memory.

31.4.22 Flash Error Address Monitor Register L (FEAML)

Address(es): FLASH.FEAML 007F C0B8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to check the address where the error has occurred if an error occurs during processing of a software command. This register stores bit 15 to bit 0 of the address where the error has occurred for the program command or blank check command, or it stores bit 15 to bit 0 of the beginning address of the area where the error has occurred for the block erase command.
Since this register value becomes undefined if setting the FRESETR.FRESET bit to 1, read the value before error processing.
When the software command is normally completed, this register stores bit 15 to bit 0 of the last address at execution of the command.
When executing a software command for the ROM or the unique ID read command, lower 2 bits become 00b.
Refer to Figure 31.1 and Figure 31.2 for details on the addresses of the flash memory.

31.4.23 Flash Start-Up Setting Monitor Register (FSCMR)

Address(es): FLASH.FSCMR 007F C0B0h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	SASMF	—	—	—	—	—	—	—	—
Value after reset:	0	1	1	1	1	1	1	Value set by user*1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0.	R
b8	SASMF	Start-Up Area Setting Monitor Flag	0: Setting to start up using the alternative area 1: Setting to start up using the default area	R
b14 to b9	—	Reserved	These bits are read as 1. Writing to these bits has no effect.	R
b15	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

Note 1. The value of the blank product is 1. It is set to the same value set in bit 8 in the FWBL register after the start-up area information program command is executed.

SASMF Flag (Start-Up Area Setting Monitor Flag)

This flag is used to confirm the settings of the start-up area.

When this flag is 0, the user program is set to start up using the alternative area.

When this flag is 1, the user program is set to start up using the default area.

31.4.24 Flash Access Window Start Address Monitor Register (FAWSMR)

Address(es): FLASH.FAWSMR 007F C0B2h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	The value set by the user*1									
Value after reset:	0	0	0	0	0	0										

Note 1. The value of the blank product is 1. It is set to the same value set in bit 9 to bit 0 the FWBL register after the access window information program command is executed.

This register is used to confirm the set value of the access window start address used for area protection.

31.4.25 Flash Access Window End Address Monitor Register (FAWEMR)

Address(es): FLASH.FAWEMR 007F C0B4h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—										
Value after reset:	0	0	0	0	0	0	The value set by the user*1									

Note 1. The value of the blank product is 1. It is set to the same value set in bit 9 to bit 0 in the FWBH register after the access window information program command is executed.

This register is used to confirm the set value of the access window end address used for area protection.

31.4.26 Unique ID Register n (UIDRn) (n = 0 to 31)

Address(es): UIDR0 0850h to UIDR31 086Fh (extra area)

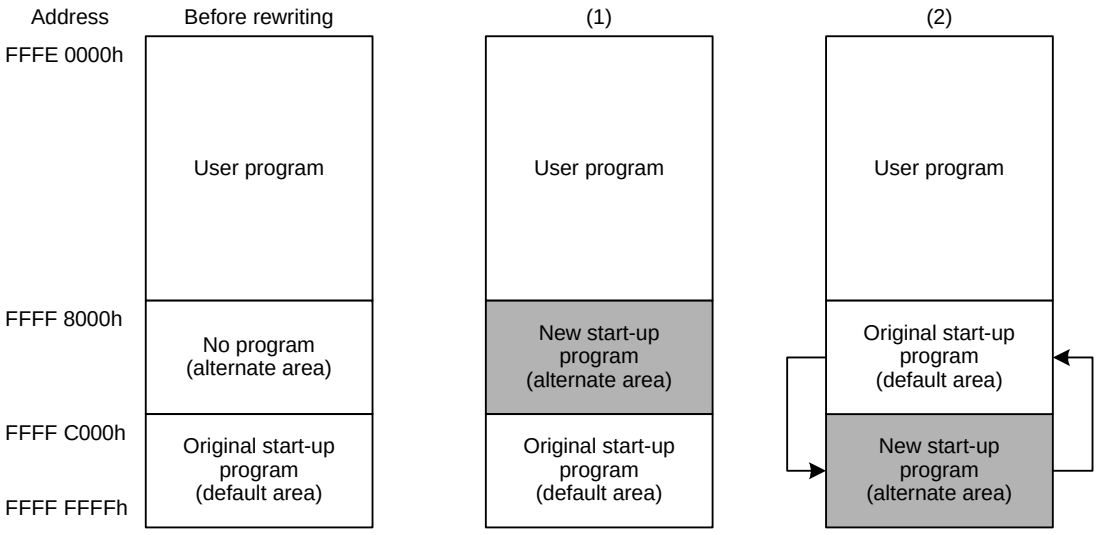
	b7	b6	b5	b4	b3	b2	b1	b0
Value after reset:	Unique value for each chip							

The UIDRn register stores a 32-byte ID code (unique ID) for identifying the individual MCU.
The unique ID is stored in the extra area of the flash memory and cannot be rewritten by the user. Use the unique ID read command to read the register value.

31.5 Start-Up Program Protection

When rewriting the start-up program*1 by self-programming, if the rewrite operation is interrupted due to temporary blackout, the start-up program may not be successfully programmed and the user program may not start properly. This problem can be avoided by rewriting the start-up program without erasing the existing start-up program using the start-up program protection. This function is available in products with a 32-Kbyte or larger ROM.
Figure 31.3 shows the overview of the start-up program protection. In this figure, the default area indicates block 0 to block 15, and the alternate area indicates block 16 to block 31.

Note 1. Program to perform operation to start the user program. It includes the fixed vector table.



(1) Program a new start-up program in the alternate area. If the alternate area fails to be rewritten, the new start-up program can be rewritten again after starting up using the default area, because the original start-up program is in the default area.

(2) After the alternate area is successfully rewritten, the default area and the alternate area are switched using the start-up area information program command. After that, the program in the alternate area starts after a reset.

Figure 31.3 Overview of the Start-Up Program Protection

31.6 Area Protection

Area protection enables rewriting only the selected blocks (access window) in the user area and disables rewriting the other blocks during self-programming. The access window cannot be set in the data area.
Specify the start address and end address to set the access window. While the access window can be set in boot mode or by self-programming, area protection is enabled only during self-programming in single-chip mode.
Figure 31.4 shows the overview of the area protection.

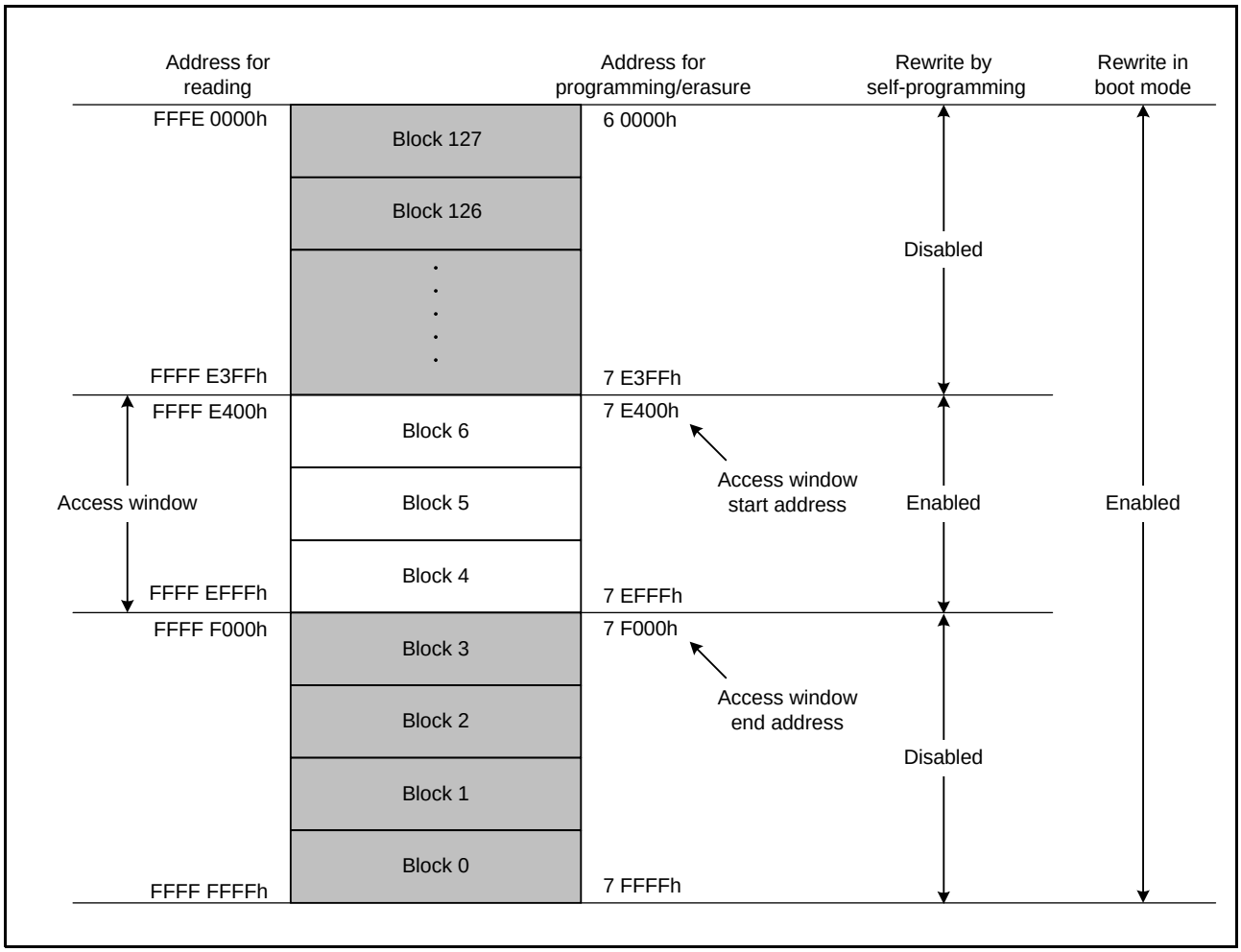


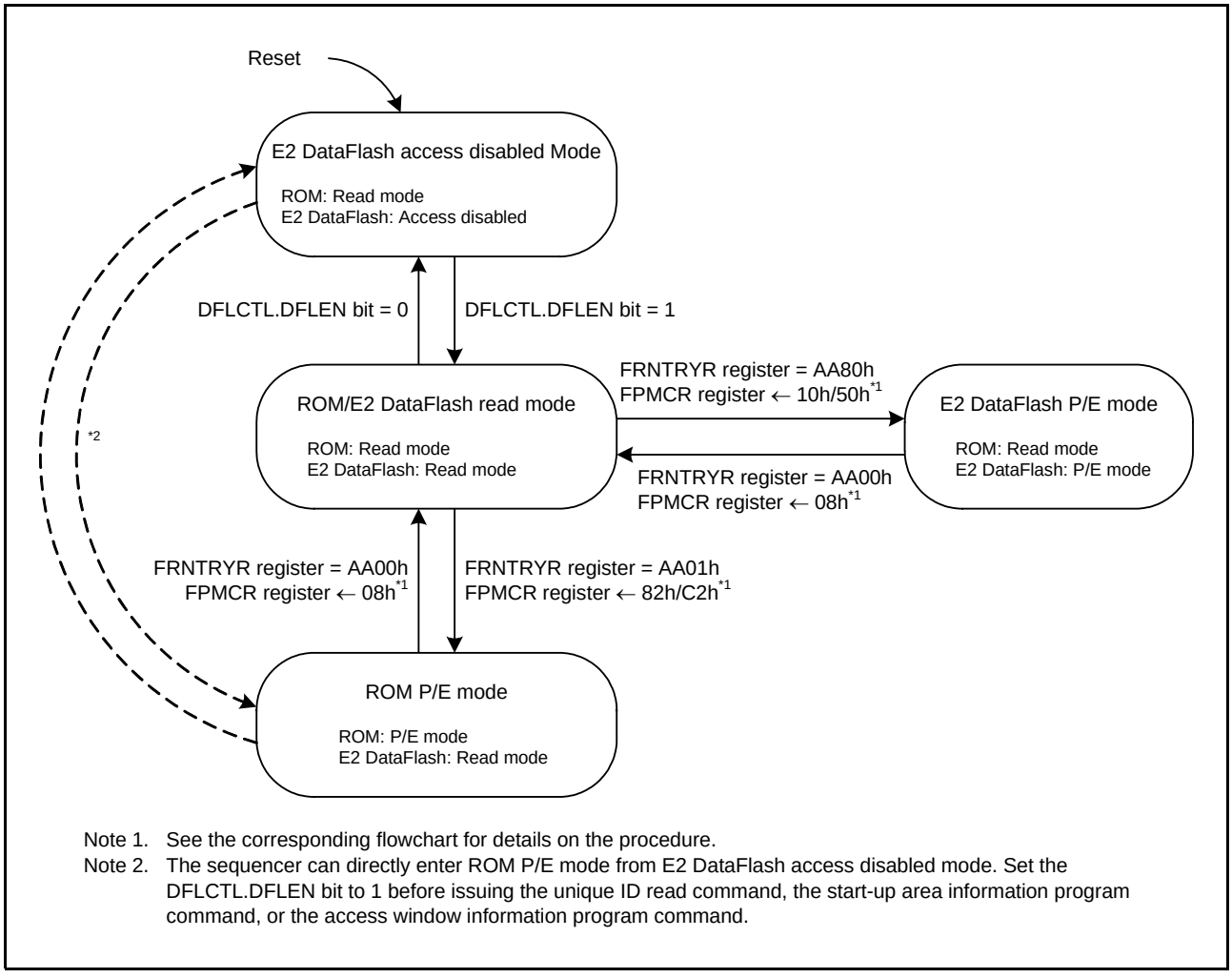
Figure 31.4 Area Protection Overview (When Blocks 4 to 6 are Set as the Access Window in Products with 128-Kbyte ROM)

31.7 Programming and Erasure

The ROM and E2 DataFlash can be programmed and erased by changing the mode of the dedicated sequencer for programming and erasure, and by issuing commands for programming and erasure.
The mode transitions and commands required to program or erase the ROM and E2 DataFlash are described below. The descriptions apply in common to boot mode and single-chip mode.

31.7.1 Sequencer Modes

The sequencer has four modes. Transitions between modes are caused by writing to the DFLCTL and FENTRYR registers and setting the FPMCR register. Figure 31.5 is a diagram of mode transitions of the flash memory.



Note 1. See the corresponding flowchart for details on the procedure.
Note 2. The sequencer can directly enter ROM P/E mode from E2 DataFlash access disabled mode. Set the DFLCTL.DFLEN bit to 1 before issuing the unique ID read command, the start-up area information program command, or the access window information program command.

Figure 31.5 Mode Transitions of the Flash Memory

31.7.1.1 E2 DataFlash Access Disabled Mode

In E2 DataFlash access disabled mode, access to the E2 DataFlash is disabled. After a reset, the sequencer enters this mode.
When setting the DFLCTL.DFLEN bit to 1, the E2 DataFlash is placed in read mode.

31.7.1.2 Read Mode

Read mode is for high-speed reading of the ROM/E2 DataFlash. Reading from a ROM address for reading can be accomplished in one ICLK clock.

(1) ROM/E2 DataFlash Read Mode

In this mode, both the ROM and E2 DataFlash are in read mode. The sequencer enters this mode from P/E mode when setting the FPMCR register to 08h, setting the FENTRYR.FENTRYD bit to 0, and setting the FENTRYR.FENTRY0 bit to 0.

31.7.1.3 P/E Modes

The P/E mode is for programming and erasure of the ROM/E2 DataFlash.

(1) ROM P/E Mode

In this mode, the ROM is in P/E mode, and the E2 DataFlash is in read mode. The sequencer enters this mode when setting the FENTRYR.FENTRYD to 0, setting the FENTRYR.FENTRY0 bit to 1, and setting the FPMCR register 82h or C2h.

(2) E2 DataFlash P/E Mode

In this mode, the ROM is in read mode, and the E2 DataFlash is in P/E mode. The sequencer enters this mode when the setting the FENTRYR.FENTRYD to 1, setting the FENTRYR.FENTRY0 bit to 0, and setting the FPMCR register 10h or 50h.

31.7.2 Mode Transitions

31.7.2.1 Transition from E2 DataFlash Access Disable Mode to Read Mode

Reading of the E2 DataFlash requires switching from E2 DataFlash access disabled mode to ROM/E2 DataFlash read mode.

Set the DFLCTL.DFLEN bit to 1 to switch to ROM/E2 DataFlash read mode.

Figure 31.6 shows the procedure for transition from E2 DataFlash access disabled mode to ROM/E2 DataFlash read mode.

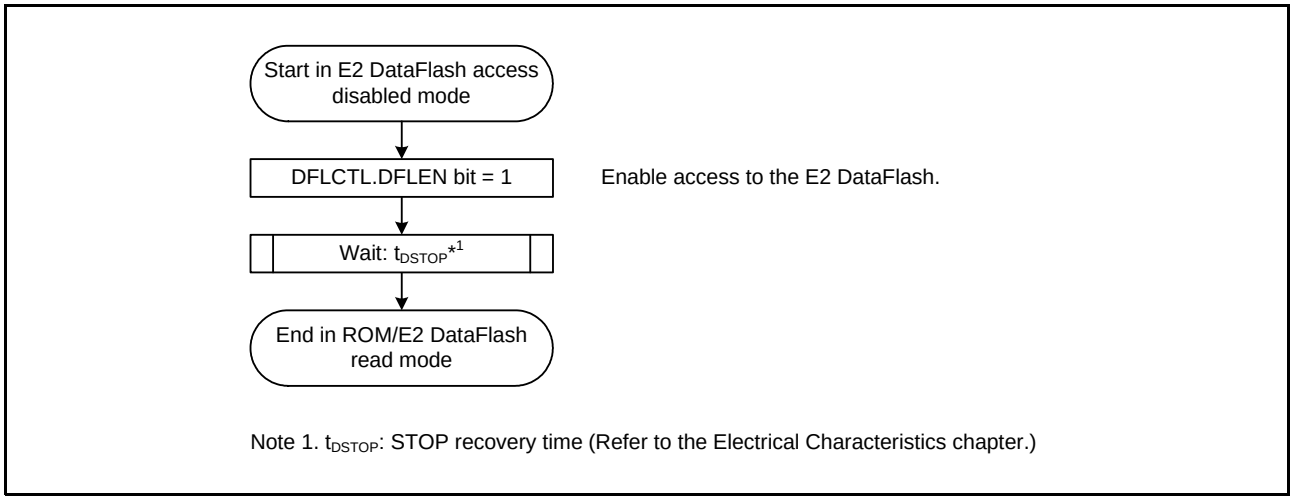


Figure 31.6 Procedure for Transition from E2 DataFlash Access Disabled Mode to ROM/E2 DataFlash Read Mode

31.7.2.2 Transition from Read Mode to P/E Mode

Switching to ROM P/E mode is required before executing a software command for the ROM.
Figure 31.7 shows the procedure for transition from ROM/E2 DataFlash read mode to ROM P/E mode. Figure 31.8 shows the procedure for transition from ROM/E2 DataFlash read mode to E2 DataFlash P/E mode.

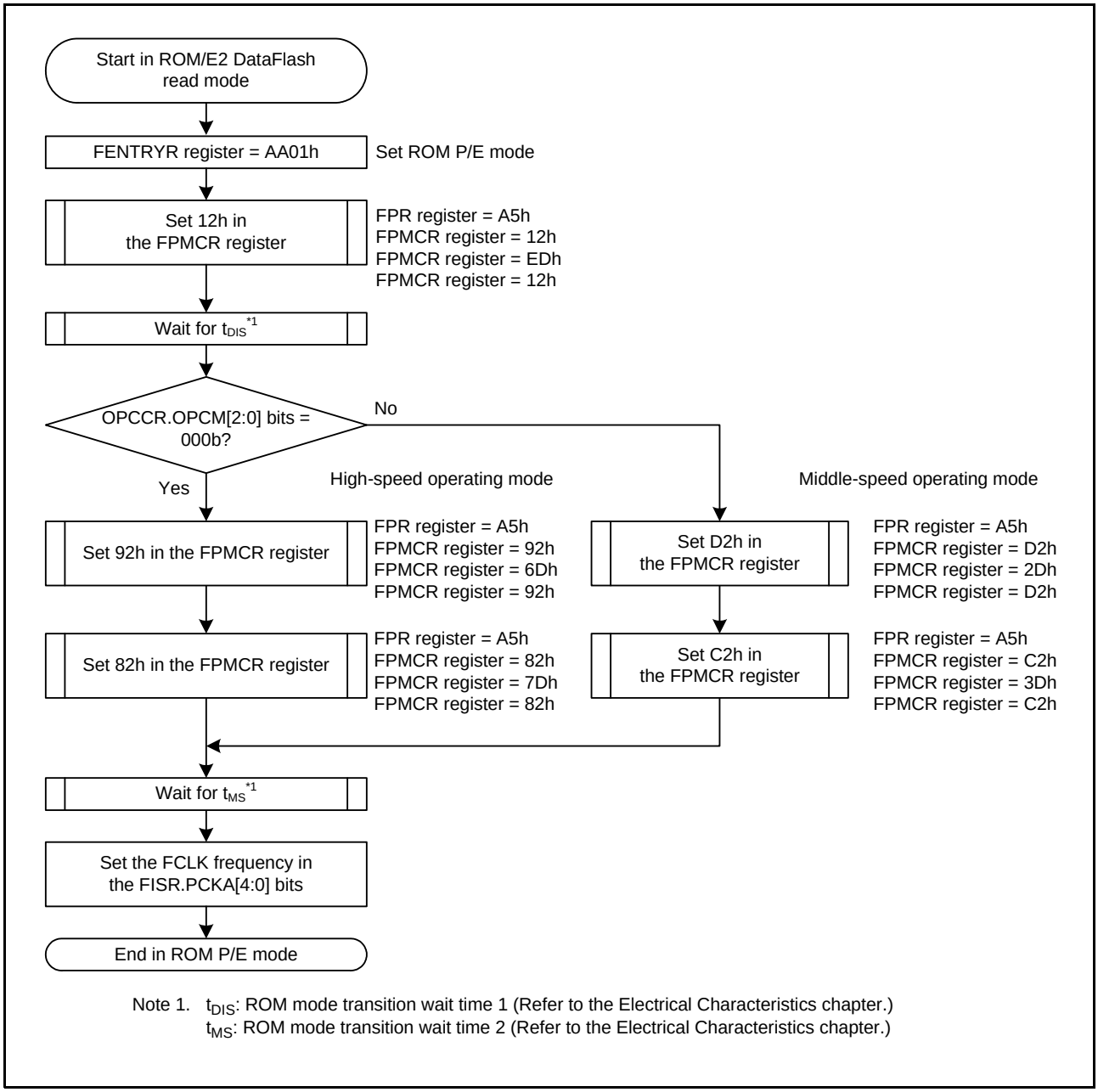


Figure 31.7 Procedure for Transition from ROM/E2 DataFlash Read Mode to ROM P/E Mode

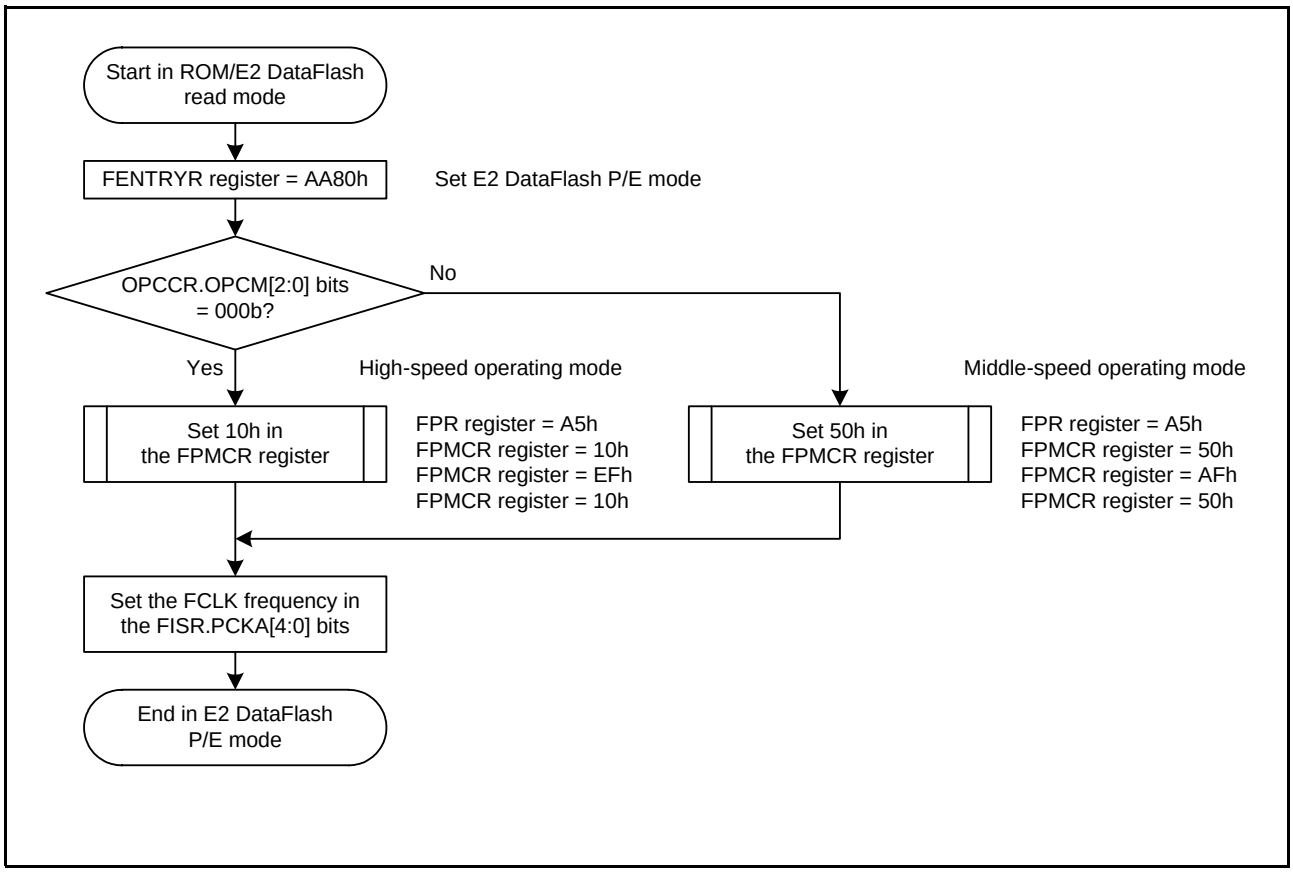
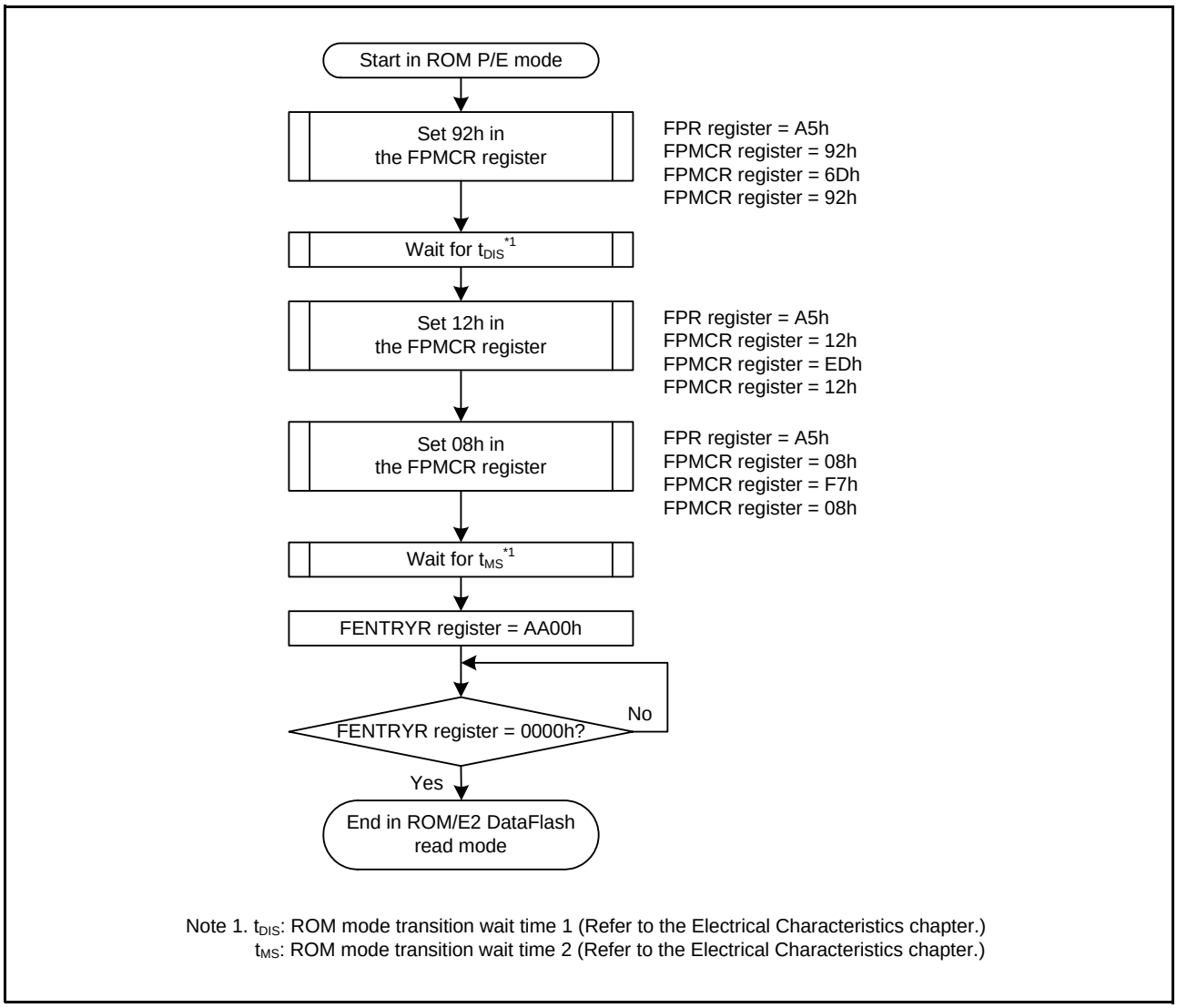


Figure 31.8 Procedure for Transition from ROM/E2 DataFlash Read Mode to E2 DataFlash P/E Mode

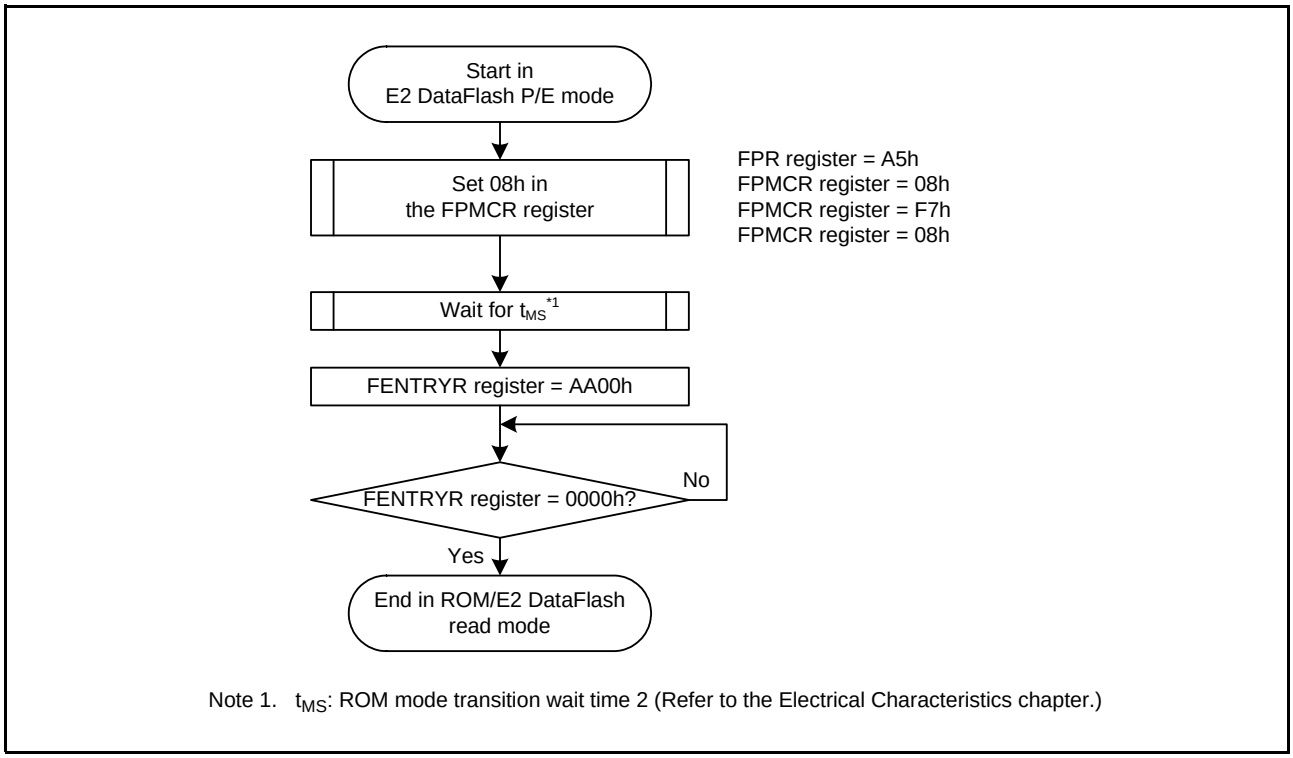
31.7.2.3 Transition from P/E Mode to Read Mode

High-speed reading of the ROM requires switching to ROM/E2 DataFlash read mode.
Figure 31.9 shows the procedure for transition from ROM P/E mode to ROM/E2 DataFlash read mode. Figure 31.10 shows the procedure for transition from E2 DataFlash P/E mode to ROM/E2 DataFlash read mode.



Note 1. t_{DIS}: ROM mode transition wait time 1 (Refer to the Electrical Characteristics chapter.)
t_{MS}: ROM mode transition wait time 2 (Refer to the Electrical Characteristics chapter.)

Figure 31.9 Procedure for Transition from ROM P/E Mode to ROM/E2 DataFlash Read Mode



Note 1. t_{MS}: ROM mode transition wait time 2 (Refer to the Electrical Characteristics chapter.)

Figure 31.10 Procedure for Transition from E2 DataFlash P/E Mode to ROM/E2 DataFlash Read Mode

31.7.3 Software Commands

Software commands consist of commands for programming and erasure and commands for programming start-up program area information and access window information. Table 31.4 lists the software commands for use with the flash memory.

Table 31.4 Software Commands

Command	Function
Program	• ROM programming (4 bytes) • E2 DataFlash programming (1 byte)
Block erase	ROM/E2 DataFlash erasure
Blank check	Check whether the specified area is blank. Confirm that data is not programmed in the area. This command does not guarantee whether the area remains erased.
Start-up area information program	Rewrite the start-up area switching information used for start-up program protection.
Access window information program	Set the access window used for area protection.
Unique ID read	Read the unique ID in the extra area.

31.7.4 Software Command Usage

This section describes how to use each software command, using flowcharts.

31.7.4.1 Program

Figure 31.11 and Figure 31.12 show the procedure to issue the program command.

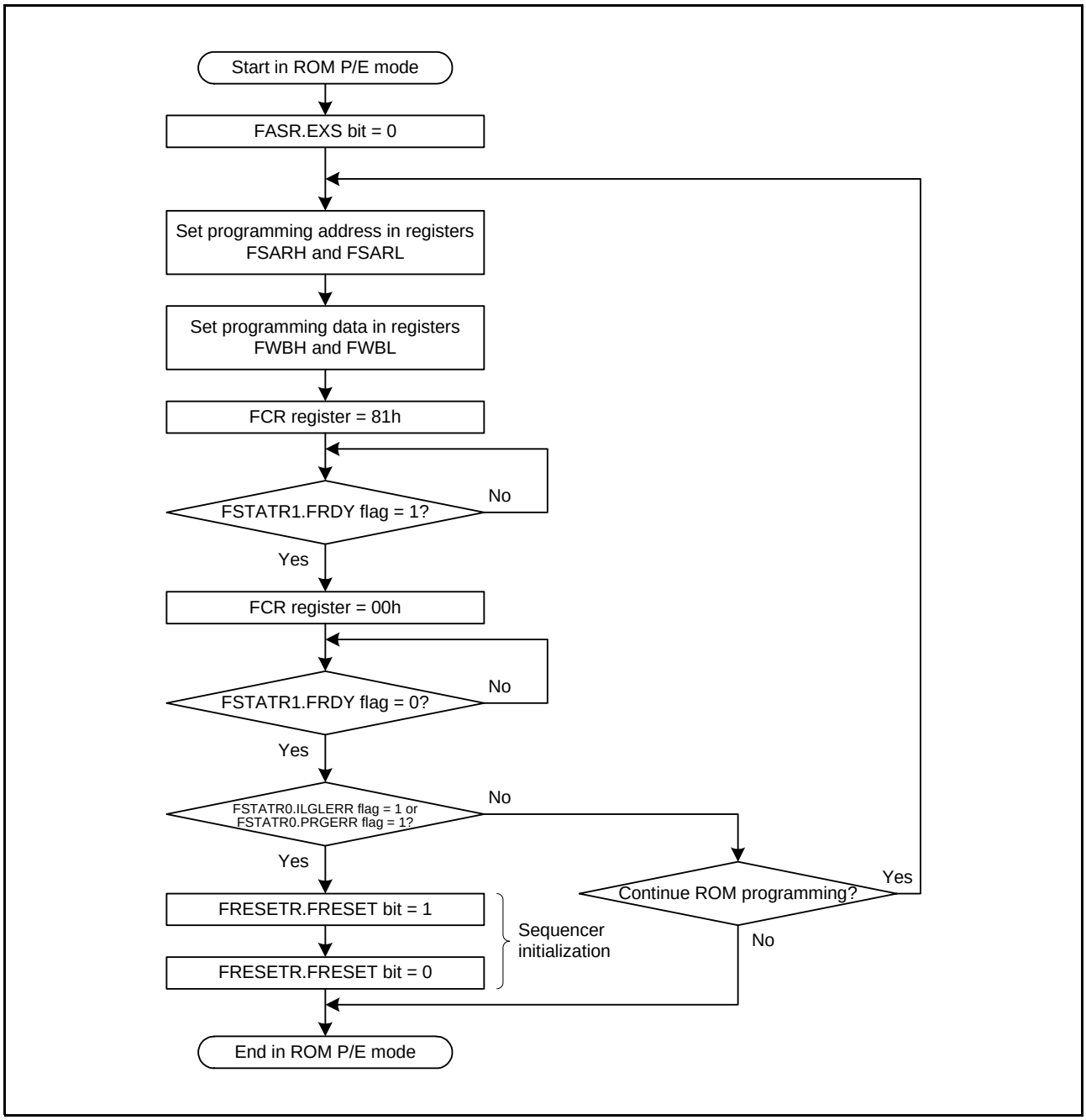


Figure 31.11 Procedure to Issue the Program Command for the ROM

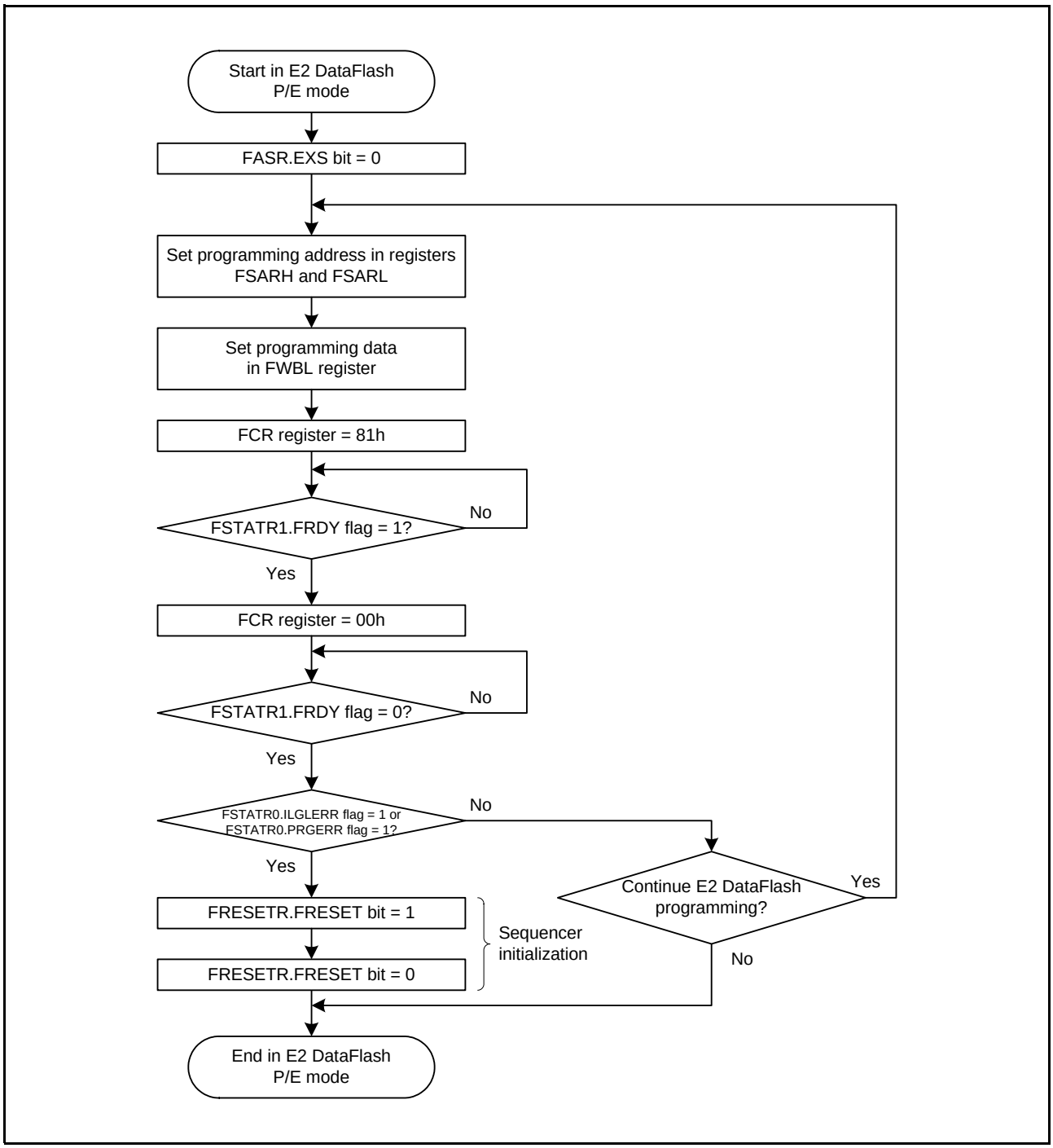


Figure 31.12 Procedure to Issue the Program Command for the E2 DataFlash

31.7.4.2 Block Erase

Figure 31.13 and Figure 31.14 show the procedure to issue the block erase command.

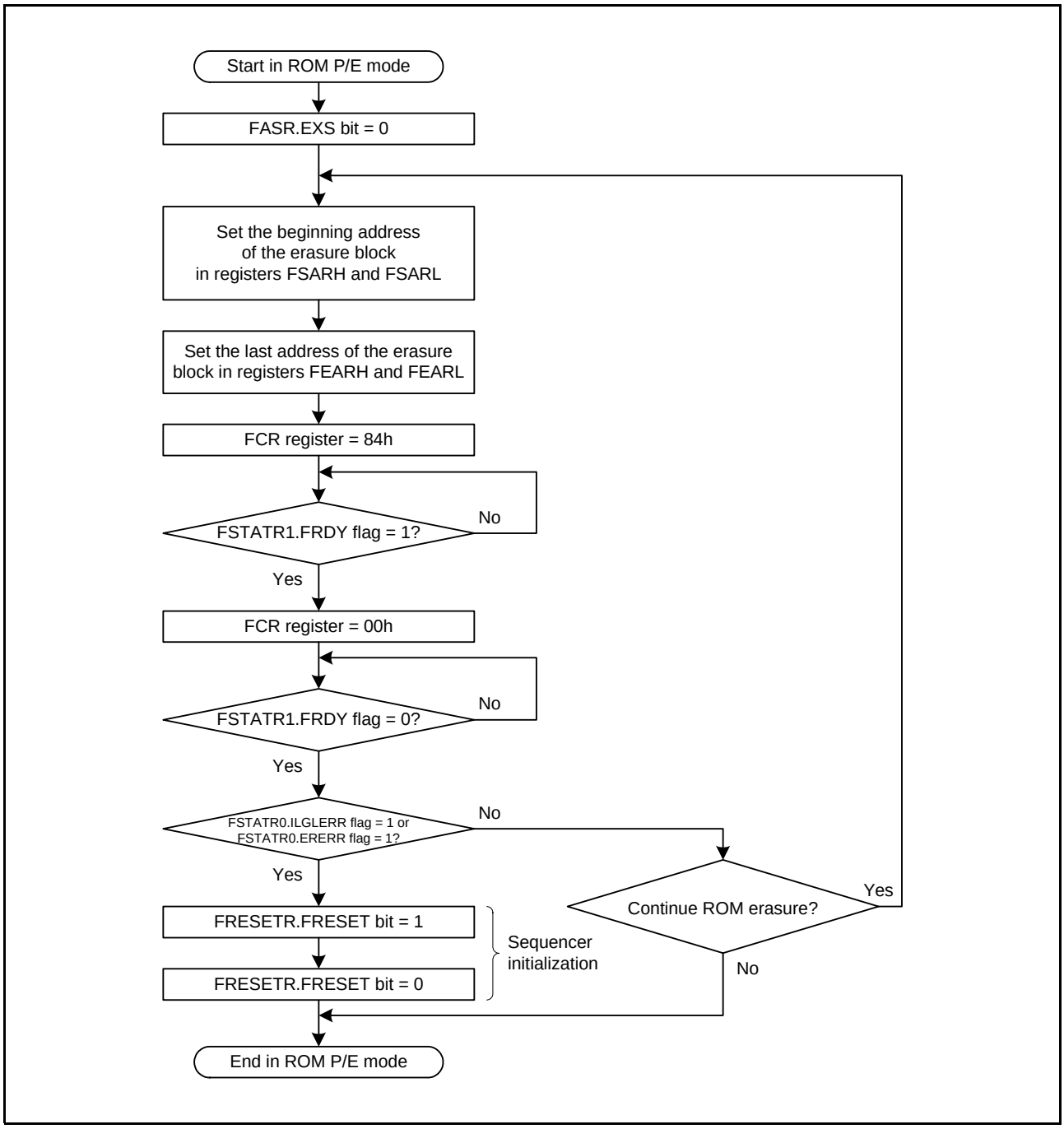


Figure 31.13 Procedure to Issue the Block Erase Command for the ROM

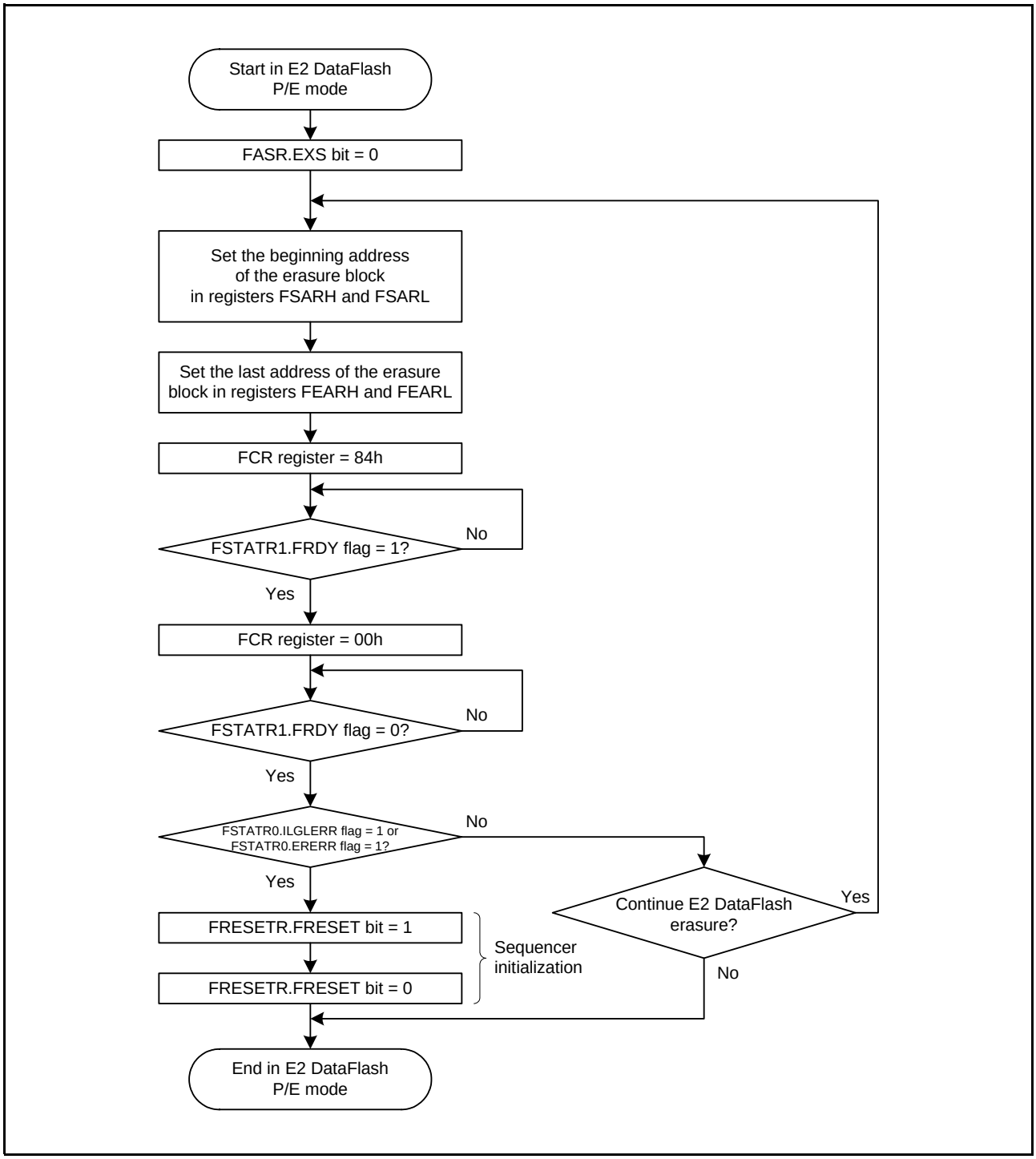


Figure 31.14 Procedure to Issue the Block Erase Command for the E2 DataFlash

31.7.4.3 Blank Check

Figure 31.15 and Figure 31.16 show the procedure to issue the blank check command.

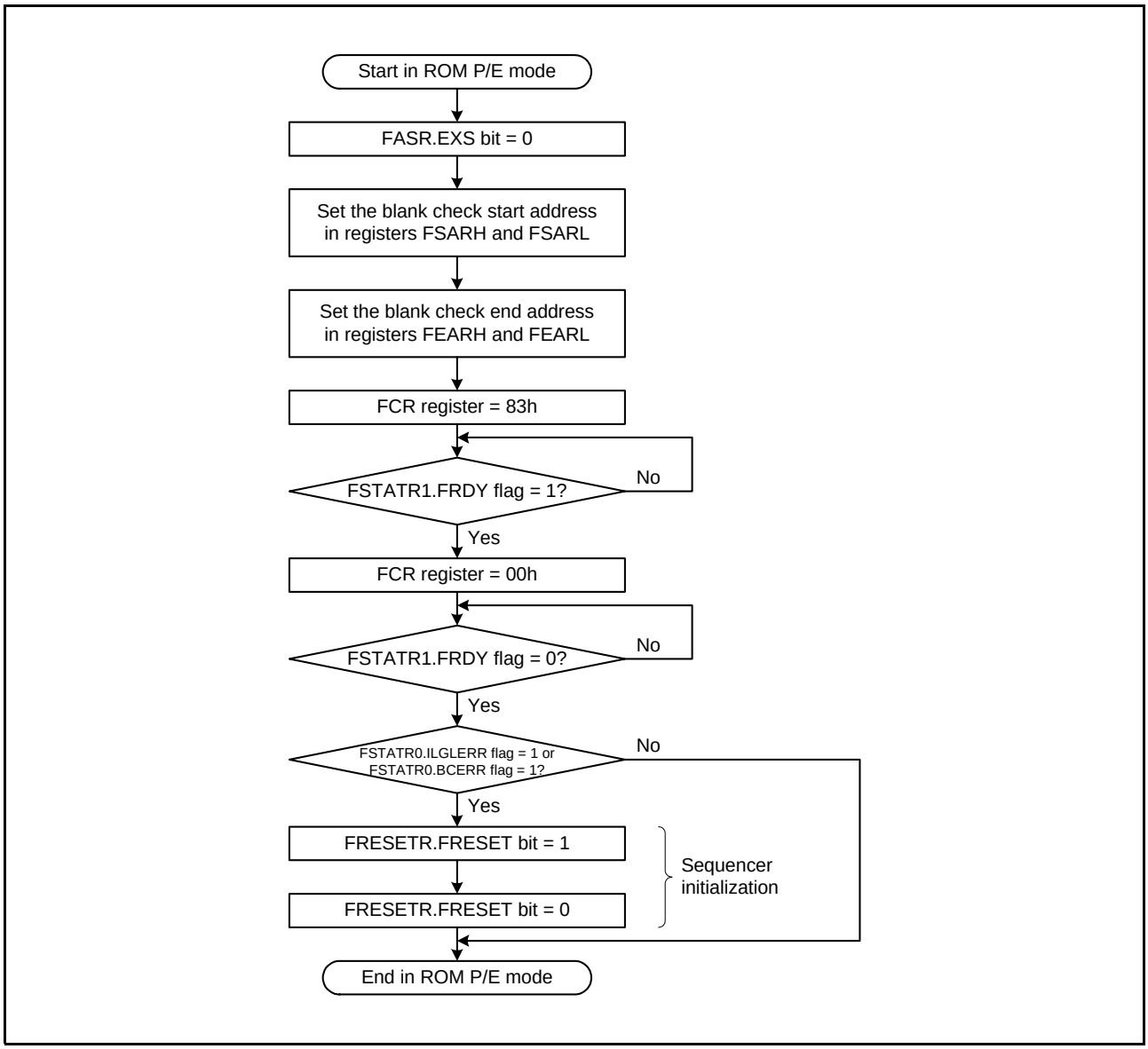


Figure 31.15 Procedure to Issue the Blank Check Command for the ROM

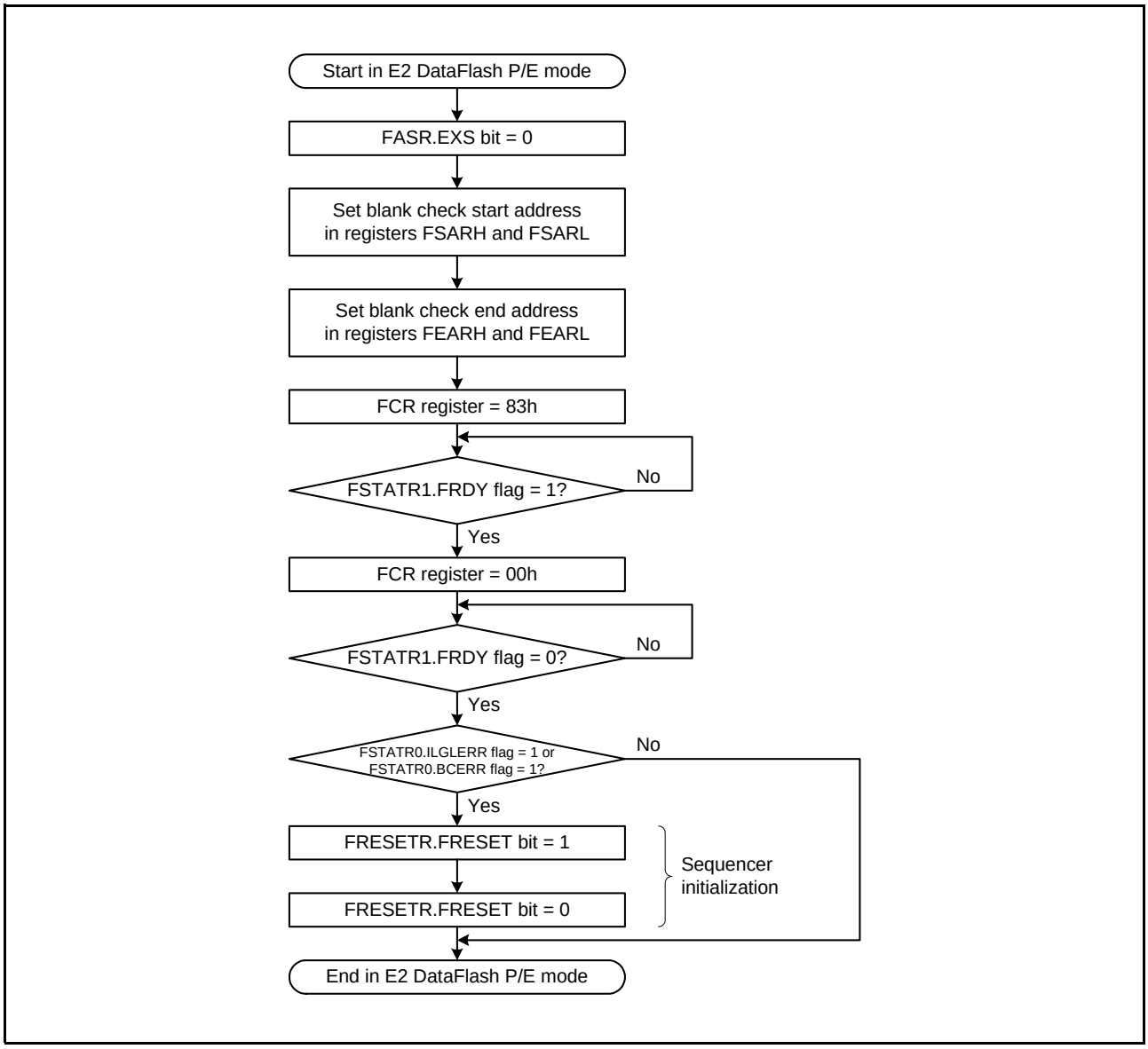


Figure 31.16 Procedure to Issue the Blank Check Command for the E2 DataFlash

31.7.4.4 Start-Up Area Information Program/Access Window Information Program

Figure 31.17 shows the procedure to issue the start-up area information program command and access window information program command.

When the sequencer has directly entered ROM/PE mode from E2 DataFlash access disabled mode, set the DFLCTL.DFLEN bit to 1 at the beginning of the procedure.

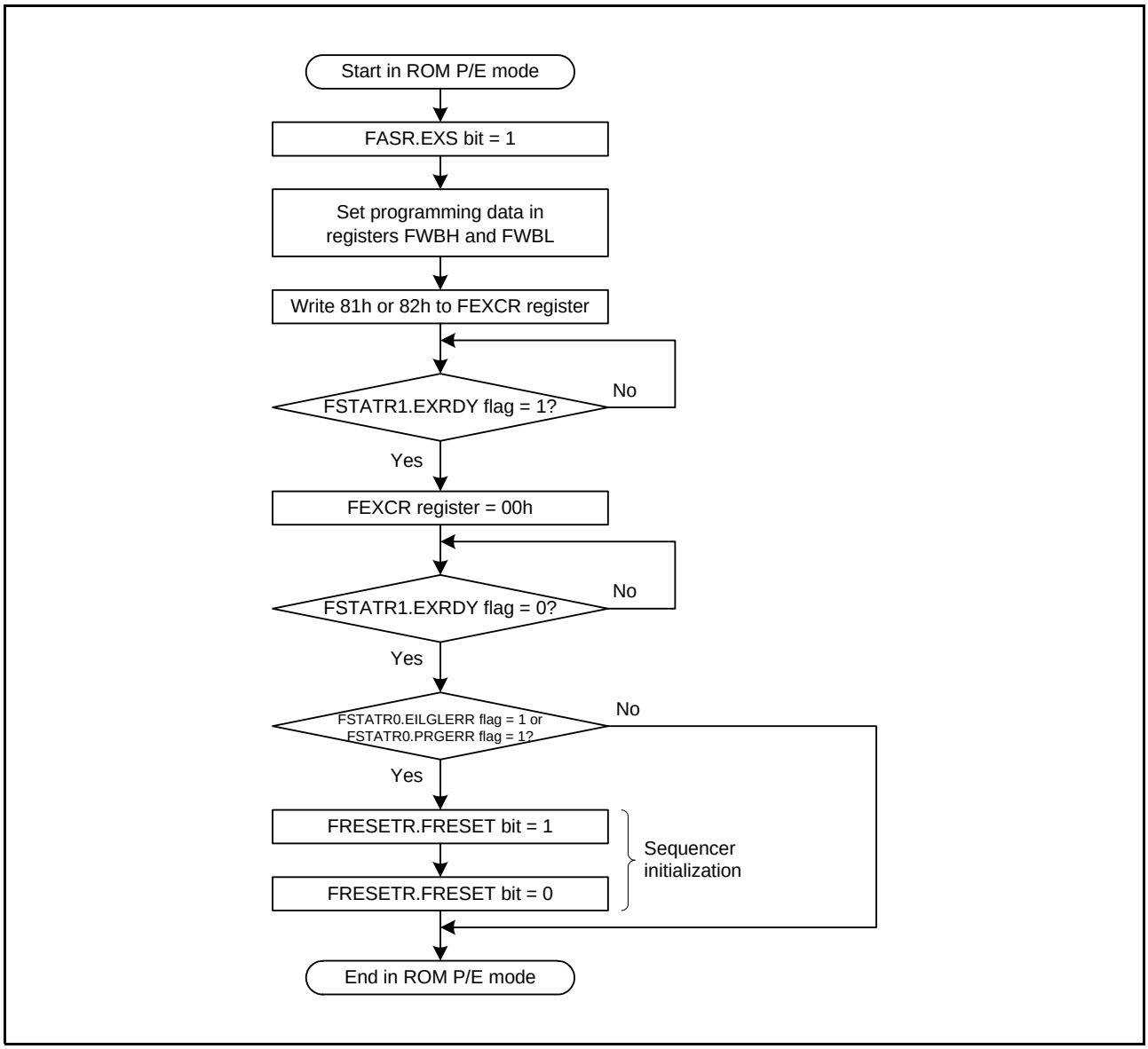


Figure 31.17 Procedure to Issue the Start-Up Area Information Program Command/Access Window Information Program Command

31.7.4.5 Unique ID Read

Figure 31.18 shows the procedure to issue the unique ID read command.

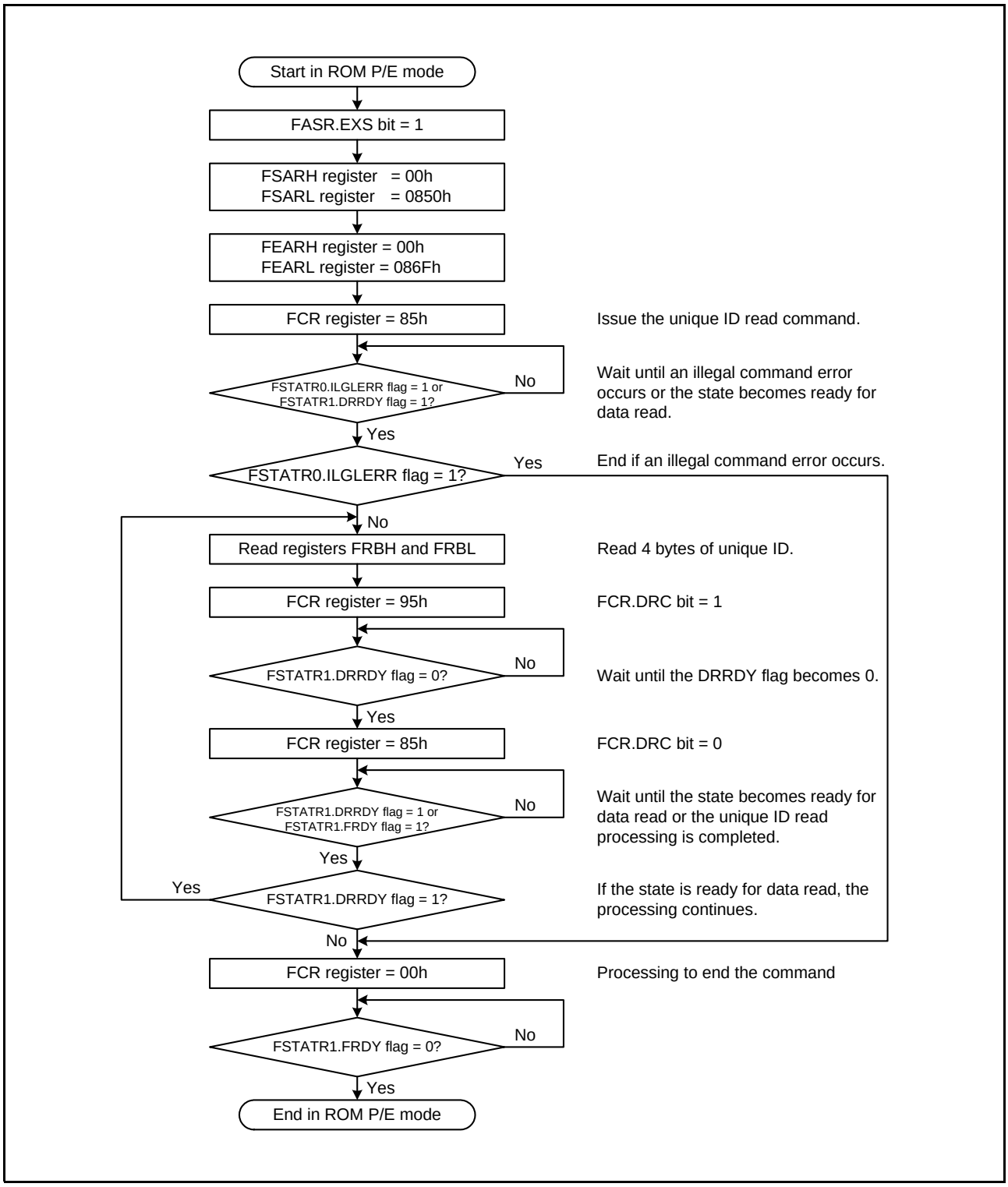


Figure 31.18 Procedure to Issue the Unique ID Read Command

31.7.4.6 Forced Stop of Software Commands

Perform the procedure shown in Figure 31.19 to forcibly stop the blank check command or block erase command. When the command processing is forcibly stopped, registers FEAMH and FEAML store the address at the time of the forced stop. For blank check, the stopped processing can be continued by copying the FEAMH and FEAML register values to registers FSARH and FSARL.

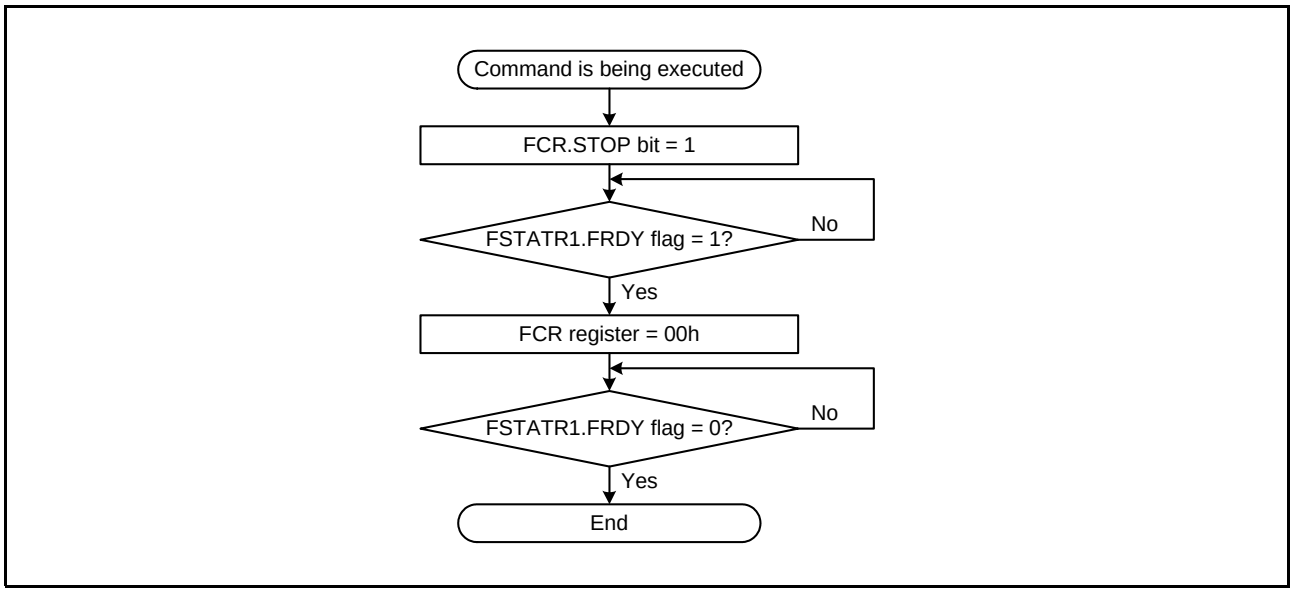


Figure 31.19 Procedure for Forced Stop of Software Commands

31.7.5 Interrupt

When software command processing or forced stop processing is completed, an interrupt (FRDYI) is generated.
When the FSTATR1.FRDY flag becomes 0 by setting the FCR.OPST bit to 0 and the FSTATR1.EXRDY flag becomes 0 by setting the FEXCR.OPST bit to 0, the next interrupt (FRDYI) can be accepted.
Clear the IRn.IR flag before setting the IERm.IEN bit of the ICU corresponding to this interrupt.

31.8 Boot Mode

The SCI or FINE interface is used in boot mode.
Table 31.5 lists the programmable and erasable areas and peripheral modules used in boot mode. Table 31.6 lists the I/O pins used in boot mode.

Table 31.5 Programmable and Erasable Areas and Peripheral Modules Used in Boot Mode

	Boot Mode	
Item	**SCI Interface**	**FINE Interface**
Programmable and erasable areas	User area Data area	User area Data area
Peripheral module	SCI1 (asynchronous serial communication)	FINE

Table 31.6 I/O Pins Used in Boot Mode

Pin Name	I/O	Mode	Description
MD	Input	Boot mode	Select operating mode (refer to section 3, Operating Modes).
MD/FINED	I/O	Boot mode (FINE interface)	Select operating mode/FINE data I/O
PB7/RXD1	Input	Boot mode (SCI interface)	Receive data*1
PB6/TXD1	Output		Transmit data*1

Note 1. When using the SCI, connect (pull up) this pin to VCC via a resistor.

31.8.1 Boot Mode (SCI Interface)

The flash memory can be programmed and erased using asynchronous serial communication in boot mode (SCI interface). The user area and data area can be rewritten.
When a reset is released while the MD pin is low, the MCU starts in boot mode (SCI interface).
Contact the manufacturer for details on the serial programmer.

31.8.1.1 Operating Conditions in Boot Mode (SCI Interface)

SCI1 is used to communicate with the serial programmer in boot mode (SCI interface).
Figure 31.20 shows an example of pin connections in boot mode (SCI interface). Table 31.7 lists pin handling in boot mode (SCI interface).
An example of pin connections shown in Figure 31.20 is a simplified circuit. Operations are not guaranteed in all systems.

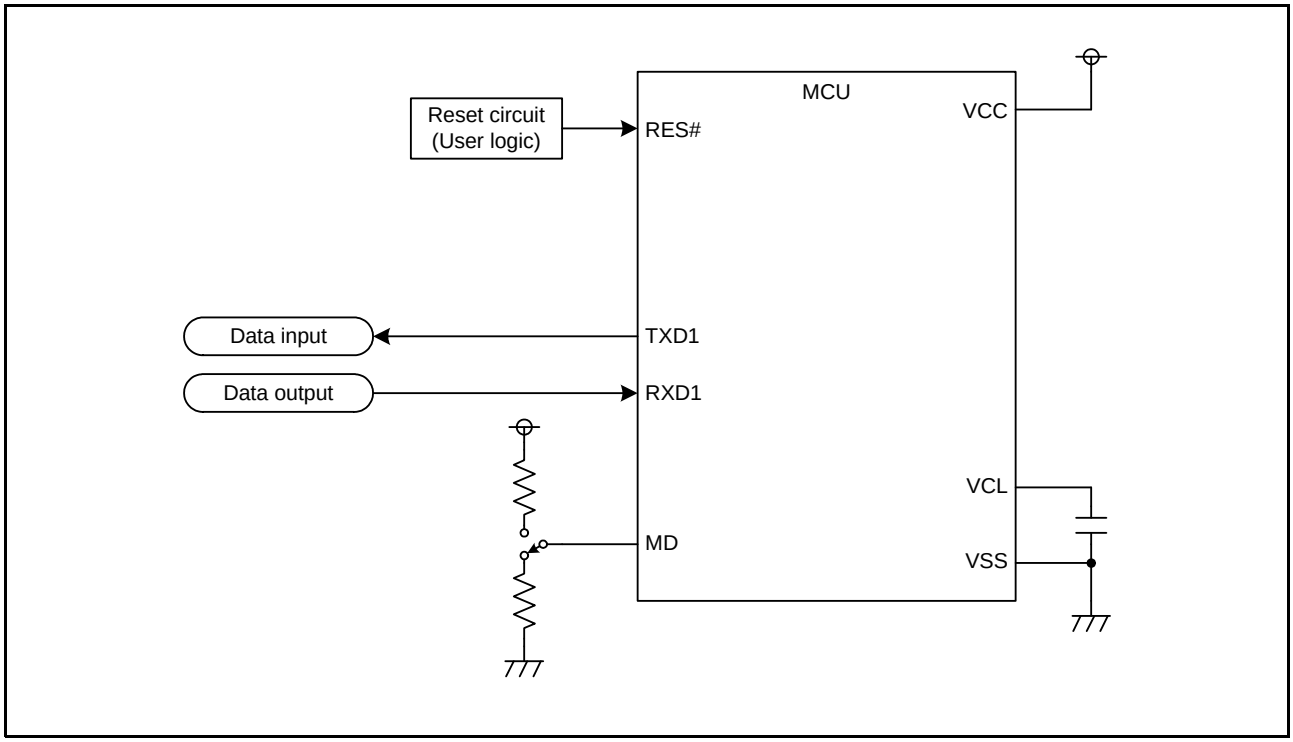


Figure 31.20 Example of Pin Connections in Boot Mode (SCI Interface)

Table 31.7 Pin Handling in Boot Mode (SCI Interface)

Pin Name	Name	I/O	Function
VCC, VSS	Power supply	—	Input 2.7 V or higher to the VCC pin. Input 0 V to the VSS pin.
VCL	Decoupling capacitor connect pin	—	Connect to the VSS pin via a decoupling capacitor for stabilizing the internal voltage.
MD	Operating mode control	Input	Input low.
RES#	Reset input	Input	Reset pin. Connect to the reset circuit.
PB7/RXD1	Data input RXD	Input	Input pin for serial data
PB6/TXD1	Data output TXD	Output	Output pin for serial data

As shown in Figure 31.21, set the format to 8-bit data, 1 stop bit, no parity, and LSB first to communicate with the serial programmer.

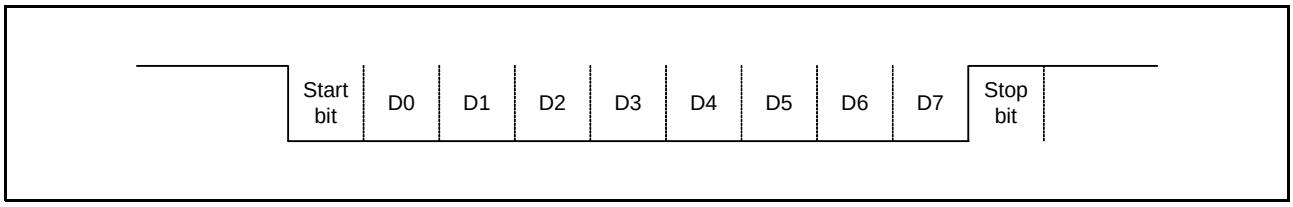


Figure 31.21 Communication Format

Initial communication with the programmer is performed at 9,600 or 19,200 bps. The communication bit rate can be changed after the MCU is connected with the programmer.
Table 31.8 lists the maximum communication bit rates for communication in boot mode (SCI interface).

Table 31.8 Conditions for Communication

Operating Voltage	Maximum Communication Bit Rate
2.7 V or higher, and lower than 3.0 V	500 kbps
3.0 V or higher, and 5.5 V or lower	2 Mbps

31.8.1.2 Starting Up in Boot Mode (SCI Interface)

To start the MCU in boot mode (SCI interface), a reset must be released by changing the RES# pin from low to high while the MD pin is low. After starting up in boot mode (SCI interface), wait at least 400 ms until communication with the MCU is enabled in boot mode (SCI interface).
As shown in Figure 31.22, keep the signal of each pin unchanged for 400 ms after the reset is released. Use resets according to the range described in section 32.4.2, Reset Timing.

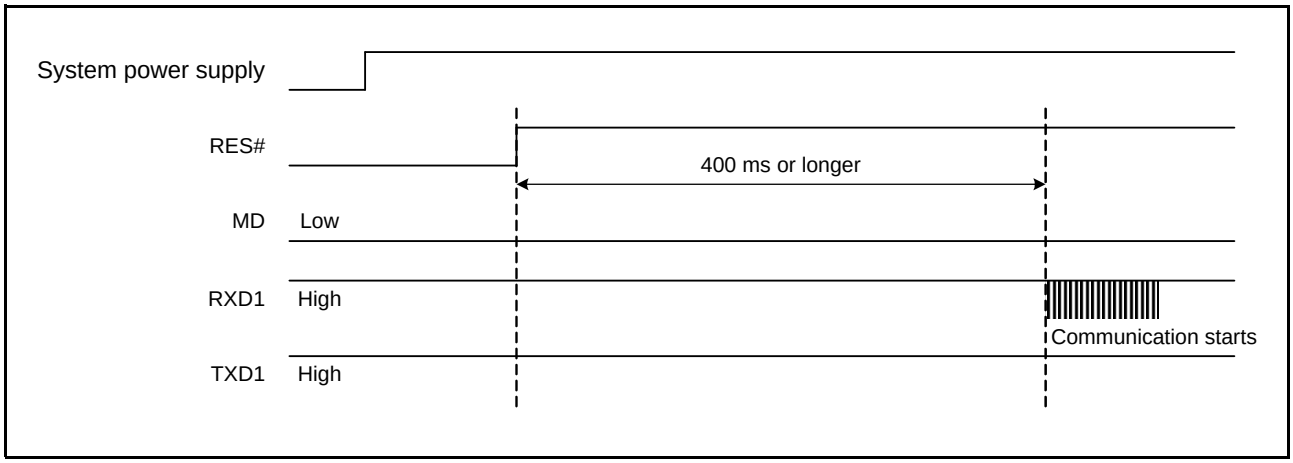


Figure 31.22 Wait Time until Communication Becomes Possible in Boot Mode (SCI Interface)

31.8.2 Boot Mode (FINE Interface)

The flash memory can be programmed and erased using the FINE in boot mode (FINE interface). The user area and data area can be rewritten.
Contact the manufacturer for details on the serial programmer.

31.8.2.1 Operating Conditions in Boot Mode (FINE Interface)

FINE is used to communicate with the serial programmer in boot mode (FINE interface).
Figure 31.23 shows an example of pin connections in boot mode (FINE interface). Table 31.9 lists pin handling in boot mode (FINE interface).
An example of pin connections shown in Figure 31.23 is a simplified circuit. Operations are not guaranteed in all systems.

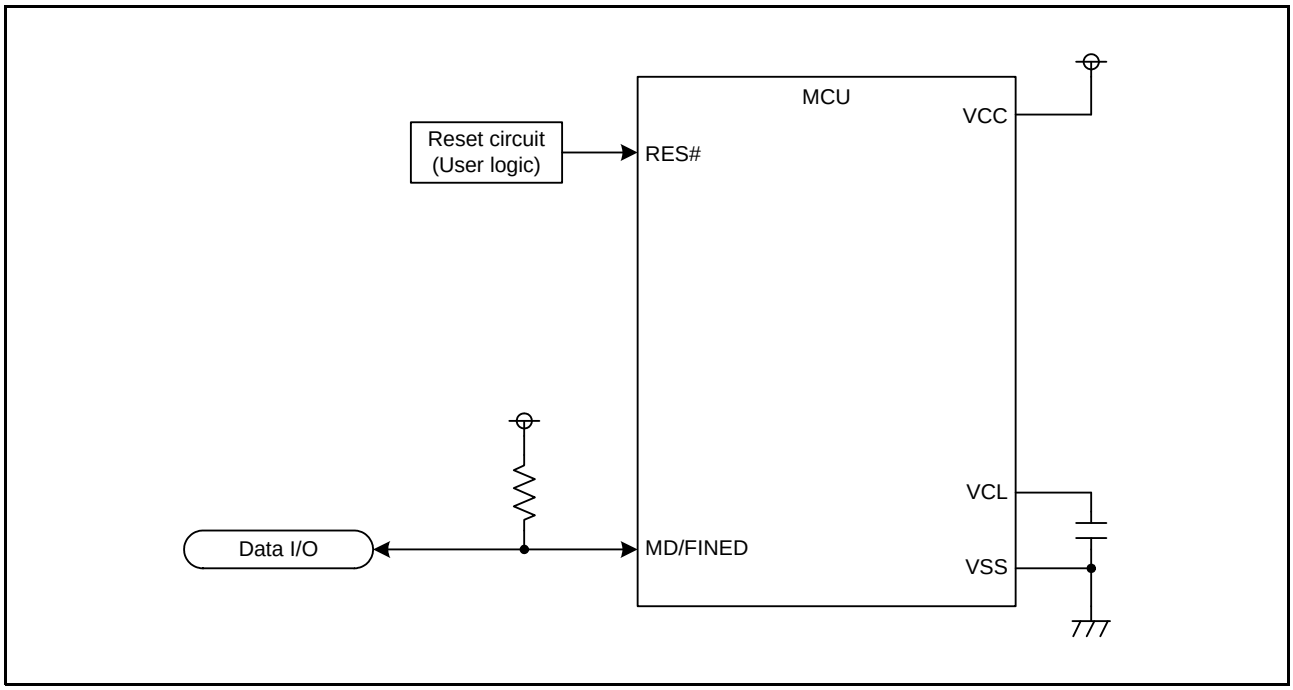


Figure 31.23 Example of Pin Connections in Boot Mode (FINE Interface)

Table 31.9 Pin Handling in Boot Mode (FINE Interface)

Pin Name	Name	I/O	Function
VCC, VSS	Power supply	—	Input 2.7 V or higher to the VCC pin. Input 0 V to the VSS pin.
VCL	Decoupling capacitor connect pin	—	Connect to the VSS pin via a decoupling capacitor for stabilizing the internal voltage.
MD/FINED	Operating mode control/data I/O	I/O	Connect to the VCC pin via a resistor (pull up).
RES#	Reset input	Input	Reset pin. Connect to the reset circuit.

31.9 Flash Memory Protection

Flash memory protection prevents the flash memory from being read or rewritten by the third party.
The boot mode ID code protection is for connecting the serial programmer, and the on-chip debugging emulator ID code protection is for connecting the on-chip debugging emulator.

31.9.1 ID Code Protection

There are two types of ID code protection: Boot mode ID code protection for connecting the serial programmer and on-chip debugging emulator ID code protection is for connecting the on-chip debugging emulator. The same ID codes are used for both functions, but operations differ.
ID codes consist of the control code and ID code 1 to ID code 15. Set ID codes to four 32-bit data in 32-bit units. Figure 31.24 shows the ID code configuration.

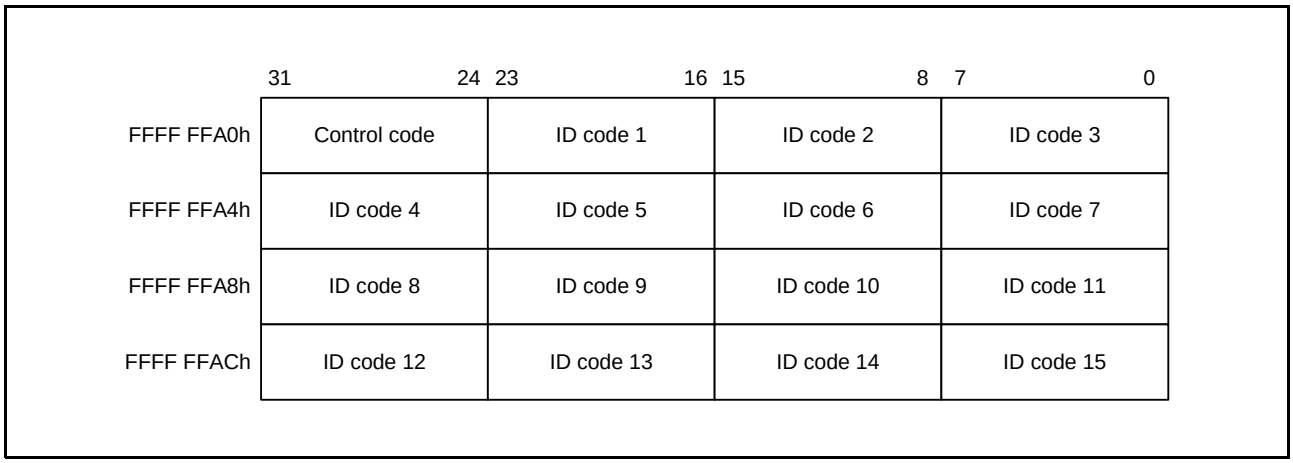


Figure 31.24 ID Code Configuration

The following shows a program example for setting ID codes.
This is an example when setting the control code to 45h and setting ID codes to 01h, 02h, 03h, 04h, 05h, 06h, 07h, 08h, 09h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, and 0Fh (from the ID code 1 field to the ID code 15 field).

C language:

```
#pragma address ID_CODE = 0xFFFFFFA0
const unsigned long ID_CODE [4] = {0x45010203, 0x04050607,0x08090A0B, 0x0C0D0E0F};
```

Assembly language:

```
.SECTION ID_CODE,CODE
.ORG 0FFFFFFA0h
.LWORD 45010203h
.LWORD 04050607h
.LWORD 08090A0Bh
.LWORD 0C0D0E0Fh
```

31.9.1.1 Boot Mode ID Code Protection

Boot mode ID code protection disables reading and programming of the user area and data area when the serial programmer is connected by the third party.
When the control code indicates 45h or 52h (boot mode ID code protection is enabled), the MCU compares 16-byte ID code sent from the serial programmer with the ID code in the user area. According to the comparison result, reading and programming the user area and data area are enabled.
When the control code indicates a value other than 45h and 52h (boot mode ID code protection is disabled), all blocks in the user area and data area are erased, and reading and programming the user area and data area are enabled.

The control code is used to enable or disable protection. Table 31.10 lists the specifications of boot mode ID code protection, and Figure 31.25 shows the authentication flow of boot mode ID code protection.
ID code 1 to ID code 15 can be set to any desired value.
However, only when disabling connection with the serial programmer, the ID codes must be set to 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, FFh, FFh, FFh, FFh, FFh, FFh, and FFh (from the ID code 1 field to the ID code 15 field).

Table 31.10 Boot Mode ID Code Protection Specifications

ID Code				
Control Code	**ID Code 1 to ID Code 15**	**Protection**	**ID Code Matching Result**	**Operation**
45h	Any desired value	Enabled	Matched	Exit the boot mode ID code authentication state and enter the program/erase host command wait state.
			Not matched	Continue the boot mode ID code authentication state.
			Not matched three times consecutively	Erase all blocks in the user area and data area, and continue boot mode ID code authentication state.
52h	50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., and FFh (8 bytes are all FFh)	Enabled	N/A	Disable reading or rewriting of the flash memory, regardless of the codes sent from the serial programmer.
	Other than above		Matched	Exit the boot mode ID code authentication state and enter the program/erase state.
			Not matched	Continue the boot mode ID code authentication state.
Other than above	Any desired value	Disabled	N/A	Erase all blocks in the user area and data area.

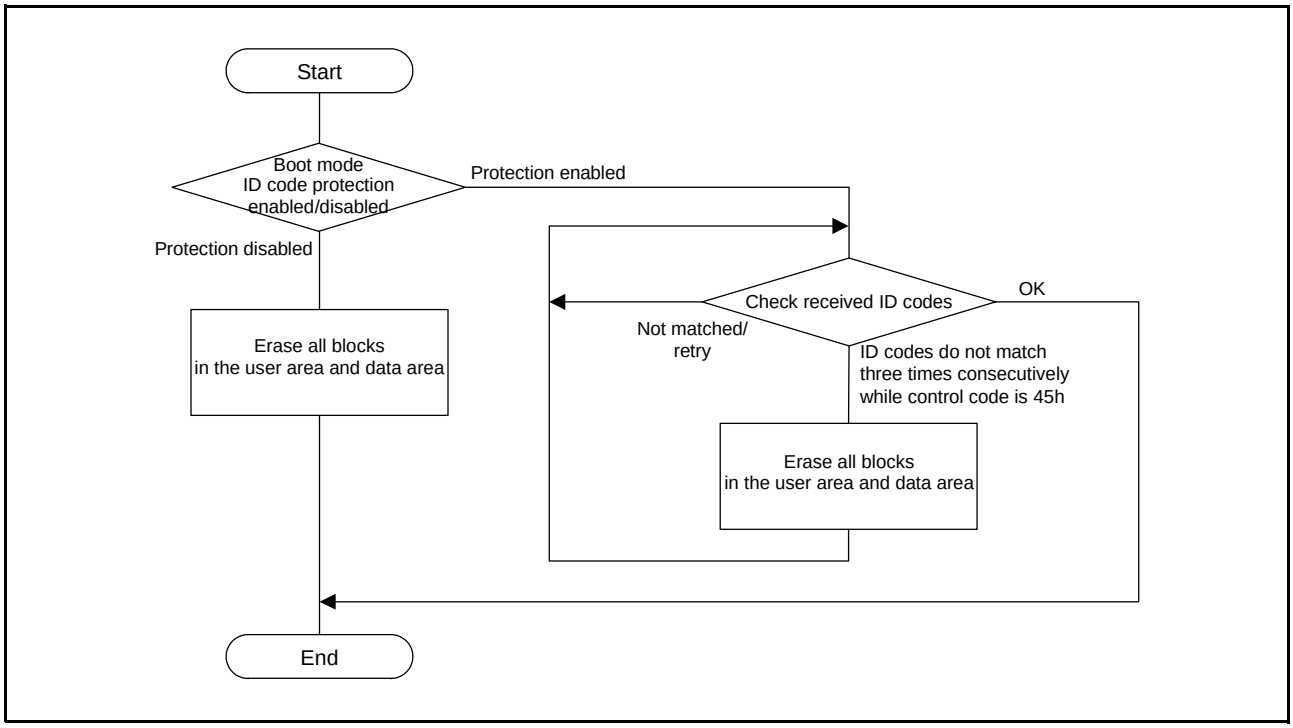


Figure 31.25 Authentication for Boot Mode ID Code Protection

31.9.1.2 On-Chip Debugging Emulator ID Code Protection

On-chip debugging emulator ID code protection enables or disables connection with the on-chip debugging emulator. When the on-chip debugging emulator ID code protection is disabled, connection with the on-chip debugging emulator is enabled. When 16-byte ID codes sent from the on-chip debugging emulator and ID codes in the user area match while on-chip debugging emulator ID code protection is enabled, connection with the on-chip debugging emulator is also enabled.

Table 31.11 lists the specifications of on-chip debugging emulator ID code protection.

Table 31.11 On-Chip Debugging Emulator ID Code Protection Specifications

ID Code				
Control Code	ID Code 1 to ID Code 15	Protection	ID Code Matching Result	Operation
FFh	FFh,…, and FFh (15 bytes are all FFh)	Disabled	N/A	Enable connection with the on-chip debugging emulator.
52h	50h, 72h, 6Fh, 74h, 65h, 63h, and 74h + any 8 bytes	Enabled	N/A	Disable connection with the on-chip debugging emulator, regardless of the codes sent from the on-chip debugging emulator.
Other than above	Other than above	Enabled	Matched	Enable connection with the on-chip debugging emulator.
			Not matched	Continue the ID code wait state.

31.10 Communication Protocol

This section describes the protocol used in boot mode. When developing a serial programmer, control with this communication protocol.

31.10.1 State Transition in Boot Mode (SCI Interface)

Figure 31.26 shows the state transition in boot mode (SCI interface).

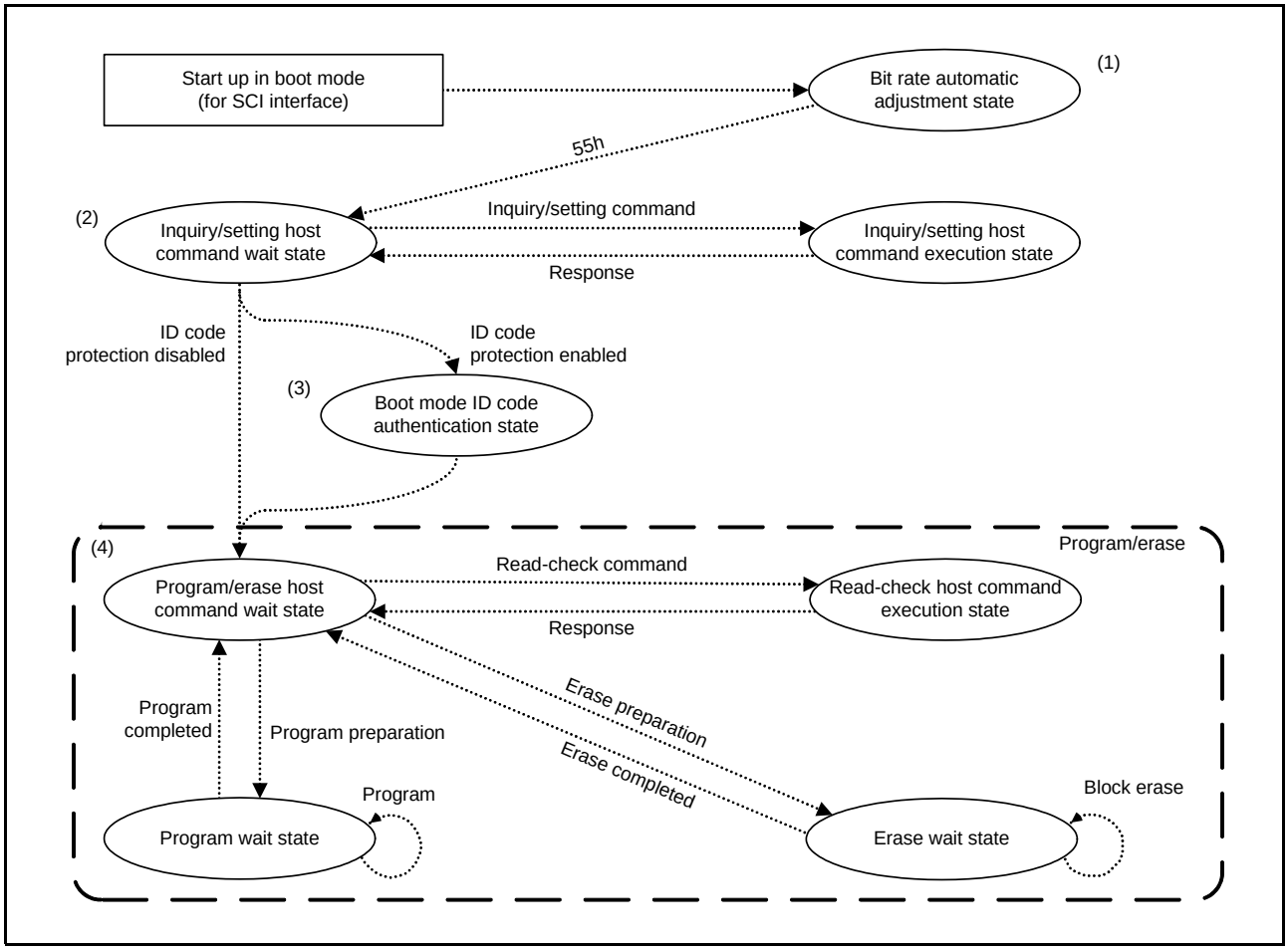


Figure 31.26 Boot Mode (SCI Interface) State Transition

(1) Bit rate automatic adjustment state

In this state, the bit rate is automatically adjusted to 9,600 or 19,200 bps for communication with the host.

When the bit rate adjustment is completed, the MCU sends 00h to the host. After that, when the MCU receives 55h sent from the host, the MCU sends E6h to the host, and enters the inquiry/setting host command wait state.

The host must not send data until 400 ms elapse after a reset of the MCU is released.

(2) Inquiry/setting host command wait state

In this state, the host can make inquiries for the MCU information including block configuration, size, and addresses where the user area and data area are allocated, and select the endian of data and a bit rate.

When the MCU receives the program/erase host state transition command from the host, it determines whether boot mode ID code protection is enabled or disabled. If boot mode ID code protection is disabled, the MCU enters the inquiry/setting host command wait state. If boot mode ID code protection is enabled, the MCU enters the boot mode ID code authentication state.

Refer to section 31.10.5, Inquiry Commands and section 31.10.6, Setting Commands for details on inquiry/setting commands.

(3) Boot mode ID code authentication state
In this state, the MCU accepts the ID code authentication command.
When boot mode ID codes do not match, the MCU remains in the boot mode ID code authentication state.
Refer to section 31.9.1.1, Boot Mode ID Code Protection for details on boot mode ID code protection. Refer to section 31.10.7, ID Code Authentication Command for details on the ID code authentication command.

(4) Program/erase state
In this state, the MCU executes program/erase or read-check commands according to commands sent from the host.
Refer to section 31.10.8, Program/Erase Commands for details on program/erase commands. Refer to section 31.10.9, Read-Check Commands for details on read-check commands.

31.10.2 Command and Response Configuration

The communication protocol is composed of a “Command” sent from the host to the MCU and a “Response” sent from the MCU to the host. Commands include 1-byte commands and multiple-byte commands. Responses include 1-byte responses, multiple-byte responses, and error responses.
A multiple-byte command and multiple-byte response have “Size” for informing the number of transmit/receive data bytes and “SUM” for detecting communication errors.
“Size” indicates the number of transmit/receive data bytes excluding Command code (the first byte), Size, and SUM.
“SUM” indicates byte data that is calculated so the total bytes of Command or Response becomes 00h.
The flash memory addresses for reading are used as the following addresses: the program address specified in the program command, the block start address specified in the block erase command, the AW start and end addresses specified in the access window information program command, and the AW start and end addresses received in the access window read command.

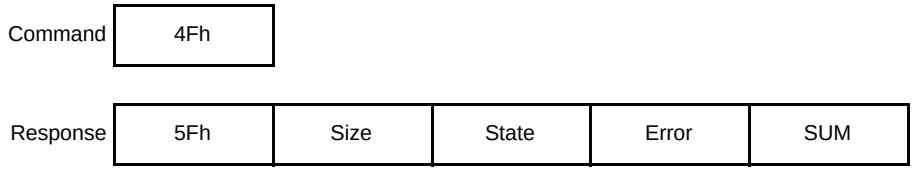
31.10.3 Response to Undefined Commands

When the MCU receives an undefined command, it sends a command error as a response. The contents of the response are shown below. “Command code” in the error response stores the first byte of the command sent from the MCU.

Error response	80h	Command code

31.10.4 Boot Mode Status Inquiry

This command is used to check the current state and which type of an error occurred immediately after a command issued in the boot program.
Table 31.12 and Table 31.13 list a state or error that the MCU responds to.
The boot mode status inquiry command can be used in the inquiry/setting host command wait state and program/erase host command wait state.



Size (1 byte): Total bytes of “State” and “Error” (the value is always 02h)
State (1 byte): MCU’s current state (see Table 31.12)
Error (1 byte): Information about the error occurred in response to a command issued immediately before (see Table 31.13)
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Table 31.12 Information Regarding the States

Code	State*1	Description
11h	Inquiry/setting host command wait state	Device selection wait state
12h/13h		Operating frequency selection wait state
1Fh		Program/erase host command wait state transition command wait state
31h	Boot mode ID code authentication state	The user area and data area are being erased
3Fh	Program/erase host command wait state	Program/erase command wait state
4Fh		Program data reception wait state
5Fh		Block erase specification wait state

Note 1. Refer to Figure 31.26 for details on the states.

Table 31.13 Error Information

Code	Description
00h	No error
11h	SUM error
21h	Device code error
24h	Bit rate selection error
29h	Block start address error
2Ah	Address error
2Bh	Data length error
51h	Erase error
52h	Not blank (blank check error)
53h	Program error
61h	ID code do not match
63h	ID code do not match and erase error
80h	Command error
FFh	Bit rate automatic adjustment error

31.10.5 Inquiry Commands

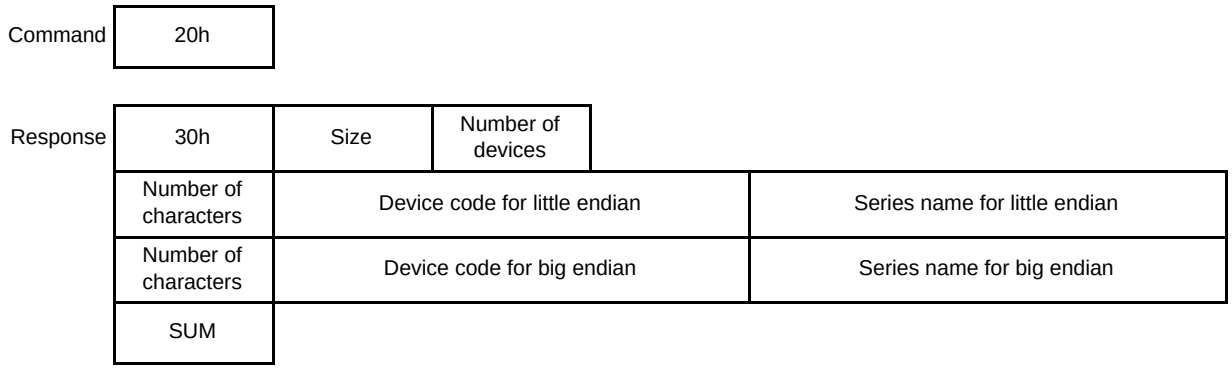
Inquiry commands are used to obtain necessary information for sending setting commands, program/erase commands, and read-check commands. Table 31.14 lists the inquiry commands. These commands can only be used in the inquiry/setting host command wait state.

Table 31.14 Inquiry Commands

Command	Description
Supported device inquiry	Inquiry for the device code and series name
Data area availability inquiry	Inquiry for the availability of the data area
User area information inquiry	Inquiry for the number of user areas, and the start and end addresses of the user area
Data area information inquiry	Inquiry for the number of data areas, and the start and end addresses of the data area
Block information inquiry	Inquiry for the start address, the block size, and the number of blocks of each of the user and data areas

31.10.5.1 Supported Device Inquiry

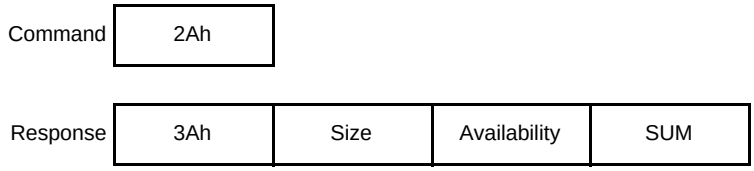
This command is used to obtain the device information for identifying the endian of developed software.
When the MCU receives this command, it sends the device information when developed software uses little endian data and the device information when developed software uses big endian data in this order.



Size (1 byte): Total bytes of Number of Devices, Characters, Device code, and Series name
Number of devices (1 byte): Number of endian types that the MCU supports (the value is always 02h)
Number of characters (1 byte): Number of characters for the device code and device name
Device code (4 bytes): Identification code indicating the endian of developed software
Series name (n bytes): The series name of the MCU (ASCII code) and the classification of little endian/big endian
SUM (1 byte): Value that is calculated so the sum of response data is 00h

31.10.5.2 Data Area Availability Inquiry

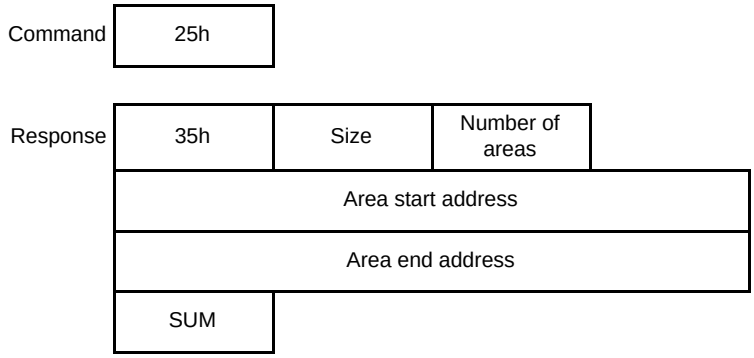
When the MCU receives this command, it sends the result indicating that the data area is available, area protection can be used, and the data area program command is available.



Size (1 byte): Number of characters of Availability (the value is always 01h)
Availability (1 byte): Availability of the data area (the value is always 1Dh)
1Dh represents the data area is available, area protection can be used, and data area program command is available.
SUM (1 byte): Value that is calculated so the sum of response data is 00h (the value is always A8h)

31.10.5.3 User Area Information Inquiry

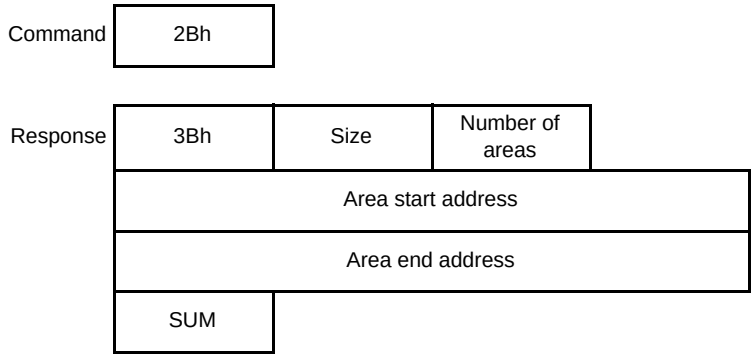
When the MCU receives this command, it sends the number of user areas and addresses.



Size (1 byte): Total bytes of Number of areas, Area start address, and Area end address (the value is always 09h)
Number of areas (1 byte): Number of user areas (the value is always 01h)
Area start address (4 bytes): Start address of the user area
Area end address (4 bytes): End address of the user area
SUM (1 byte): Value that is calculated so the sum of the response data is 00h

31.10.5.4 Data Area Information Inquiry

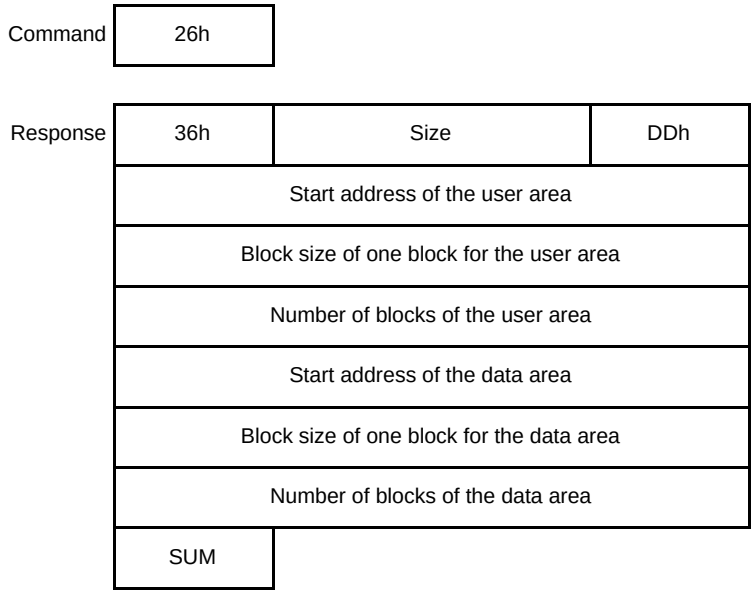
When the MCU receives this command, it sends the number of data areas and addresses.



Size (1 byte): Total bytes of data of Number of areas, Area start address, and Area end address (the value is always 09h)
Number of areas (1 byte): Number of areas in the data area (the value is always 01h)
Area start address (4 bytes): Start address of the data area (the value is always 0010 0000h)
Area end address (4 bytes): End address of the data area (the value is always 0010 0FFFh)
SUM (1 byte): Value that is calculated so the sum of the response data is 00h (the value is always 8Dh)

31.10.5.5 Block Information Inquiry

When the MCU receives this command, it sends the start address, the size of one block, and the number of blocks in the user area and data area.



Size (2 bytes): Total bytes of data from DDh to Number of blocks of the data area (the value is always 00 19h)
Start address of the user area (4 bytes): Start address of the user area
Block size of one block for the user area (4 bytes): Memory size of one block (the value is always 00 00 04 00h)
Number of blocks of the user area (4 bytes): Number of blocks in the user area
Start address of the data area (4 bytes): Start address of the data area (the value is always 00 10 00 00h)
Block size of one block for the data area (4 bytes): Memory size of one block (the value is always 00 00 04 00h)
Number of blocks of the data area (4 bytes): Number of blocks in the data area (the value is always 00 00 00 04h)
SUM (1 byte): Value that is calculated so the sum of response data is 00h

31.10.6 Setting Commands

Setting commands are used to configure the settings necessary to execute program/erase commands in the MCU. Table 31.15 lists setting commands. These commands can be used only in the inquiry/setting host command wait state.

Table 31.15 Setting Commands

Command	Function
Device select	Select a device code.
Operating frequency select	Change the bit rate for communication.
Program/erase host command wait state transition	Enter the program/erase host command wait state or boot mode ID code authentication state.

31.10.6.1 Device Select

This command is used to specify the endian of developed software. Select a device code from among the device codes obtained in the response to the support device inquiry command.
If the received device code matches the supported device, the MCU sends a response (46h).
If the device is not supported or the SUM of the received command does not match, the MCU sends an error response.

Command	10h	Size	Device code	SUM

Size (1 byte): Number of characters of the device code (the value is always 04h)
Device code (4 bytes): Identification code to identify an endian of the developed software
(code in the response to the support device inquiry command)
SUM (1 byte): Value that is calculated so the sum of command data is 00h

Response	46h	

Error response	90h	Error

Error (1 byte): Error code
11h: SUM error
21h: Device code error

31.10.6.2 Operating Frequency Select

This command is used to specify the operating frequency of the MCU and a bit rate for communication with the flash memory programmer. The bit rate selected in this command should be set to a value with error of less than 4% compared to the bit rate obtained by dividing 32 or 8 MHz that corresponds to the operating voltage.
If the specified settings can be supported, the MCU sends a response (06h). If the bit rate error is 4% or more or the SUM of the received command does not match, the MCU sends an error response.

After the host receives a response, wait for at least a 1-bit period at the old bit rate, and send communication confirmation data at the new bit rate.
If the MCU successfully receives communication confirmation data, the MCU sends a response (06h). If the MCU fails to receive the communication confirmation data, the MCU sends an error response.

Command	3Fh	Size	Bit rate	Dummy data
	Number of clocks	Multiplier 1	Multiplier 2	
	SUM			

Size (1 byte): Total bytes of data of Bit rate, Dummy data, Number of clocks, and Multiplier (the value is always 07h)
Bit rate (2 bytes): New bit rate
The value is calculated by dividing the bit rate by 100 (Example: Set 00C0h for 19200 bps)
Dummy data (2 bytes): The value should always be set to 0000h
Number of clocks (1 byte): Types of clocks for multiplier setting (the value is always 02h)
Multiplier 1 (1 byte): Multiplier of the system clock (ICLK) (the value is always 01h)
Multiplier 2 (1 byte): Multiplier of the peripheral module clock (PCLK) (the value is always 01h)
SUM (1 byte): Value that is calculated so the sum of command data including dummy data is 00h

Response	06h	

Error response	BFh	Error

Error (1 byte): Error code
11h: SUM error
24h: Bit rate selection error

Communication confirmation	06h

Response	06h

Error response	FFh

- Bit rate selection error

A bit rate selection error occurs when the bit rate specified with the operating frequency select command cannot be set to a value with error of less than 4%. When the new bit rate specified with the operating frequency select command is B, and 32 (MHz) or 8 (MHz) corresponding to the operating voltage is Pφ, the bit rate error is calculated by the following formula:

$$\text{Error}(\%) = \left(\frac{P\varphi \times 10^6}{B \times 32 \times N} - 1\right) \times 100$$

$$N = \text{INT}\left(\frac{P\varphi \times 10^6}{B \times 32}\right)$$

Pφ: 32 (MHz) when the operating voltage is 3.0 V or above
8 (MHz) when the operating voltage is below 3.0 V
B: New bit rate (bps)
N: Ratio between Pφ and the new bit rate multiplied by 32 (however, 1 ≤ N ≤ 256)

31.10.6.3 Program/Erase Host Command Wait State Transition

This command is used for the transition from the inquiry/setting host command wait state to the program/erase host command wait state.
When the MCU receives this command, it determines whether boot mode ID code protection is enabled or disabled.

When boot mode ID code protection is disabled, all blocks in the user area and data area are erased.
When all blocks are successfully erased, the MCU sends a response (06h) and enters the program/erase host command wait state. If not all blocks are successfully erased, the MCU sends an error response.
When boot mode ID code protection is enabled, the MCU sends a response (16h) and enters boot mode ID code authentication state.

Command	40h

Response	ACK

ACK (1 byte): ACK code
06h: ID code protection is disabled.
16h: ID code protection is enabled.

Error response	C0h	Error

Error (1 byte): Error code
51h: Erase error

31.10.7 ID Code Authentication Command

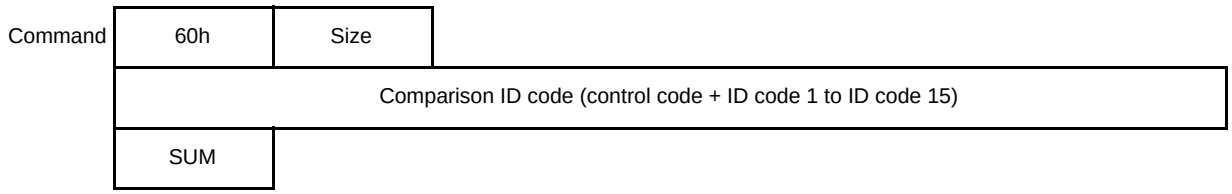
This command is used for ID code authentication when boot mode ID code protection is enabled.
Table 31.16 lists ID code authentication command. This command can be used only in the boot mode ID code authentication state.

Table 31.16 ID Code Authentication Command

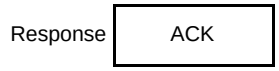
Command	Function
ID code check	Compare the 16-byte code sent from the host and ID code.

31.10.7.1 ID Code Check

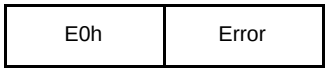
This command is used to unlock boot mode ID code protection.
The comparison ID code specified with the command should be set to the same value as the control code and ID code 1 to ID code 15.
If the comparison ID code sent from the host matches the ID code programmed in the user area, the MCU sends a response (06h) and enters program/erase host command wait state.
If the codes do not match or the SUM of the received command does not match, the MCU sends an error response.
When the ID codes do not match three times consecutively while the control code is 45h, all blocks in the user area and data area are erased. If an error occurs during erasure, the MCU sends an error response.
Also, even if all blocks are successfully erased, the MCU sends an error response and continues the boot mode ID code state. Reset the MCU to enter the program/erase host command wait state.



Size (1 byte): Number of bytes of ID codes (the value is always 10h)
ID code (16 bytes): Control code (1 byte) + ID code 1 to ID code 15 (15 bytes)
SUM (1 byte): Value that is calculated so the sum of the command data is 00h



ACK (1 byte): ACK code
06h: The MCU enters the program/erase host command wait state.



Error (1 byte): Error code
11h: SUM error
61h: ID codes do not match
63h: ID codes do not match and erase error

31.10.8 Program/Erase Commands

Program/erase commands are used to program or erase the user area or data area based on the response to inquiry commands. Table 31.17 lists commands used in each of the program/erase host command wait state, program wait state, and erase wait state. Table 31.18 lists commands that can be accepted in each state.
When a command that is not listed in Table 31.18 is received in each state, the MCU sends a command error response.

Table 31.17 Program/Erase Commands

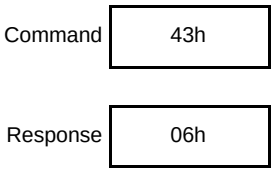
Command	Function
User/data area program preparation	Select the user area or data area to program, and enter the program wait state.
Program	Program the specified data to the selected area in the user area or data area. Or enter the program/erase host command wait state (end of program).
Data area program	Program the specified-size data to the selected area in the data area. Or enter the program/erase host command wait state (end of program).
Erase preparation	Enter the erase wait state.
Block erase	Erase the selected block, or enter the program/erase host command wait state (end of erase).

Table 31.18 Acceptable Commands for Each State

State	Acceptable Command
Program/erase host command wait state	User/data area program preparation command, and erase preparation command
Program wait state	Program command and data area program command
Erase wait state	Block erase command

31.10.8.1 User/Data Area Program Preparation

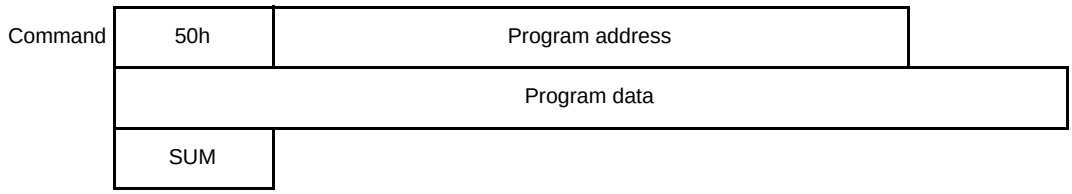
This command is used to prepare for accepting the program command and the data area program command.
When the MCU receives this command, it recognizes that an instruction to prepare for the program command is issued from the host. Then, the MCU enters the program wait state, where only the program command and the data area program command can be accepted, and sends a response (06h).



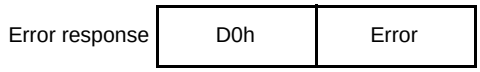
31.10.8.2 Program

This command is used to program the specified data to the user area or data area. Set the lower 8 bits to 0 for the program address selected in this command. When the data length is shorter than 256 bytes, the data cannot be programmed. Fill the gaps with FFh.
When the program from the selected address is successfully completed, the MCU sends a response (06h). If the SUM of the received command does not match or an error occurs during a program operation, the MCU sends an error response.

To enter the program/erase host command wait state after the program operation ends, send 50h FFh FFh FFh FFh B4h from the host. The MCU sends a response (06h), and enters the program/erase host command wait state.



Program address (4 bytes): Address for program destination
Set the lower 8 bits to 0
Set FFFF FFFFh for end of program
Program data (n bytes): Program data (n = 256, 0 for end of program)
When the program data is less than n bytes, set FFh for the missing data.
No program data for the end of program
SUM (1 byte): Value that is calculated so the sum of command data is 00h

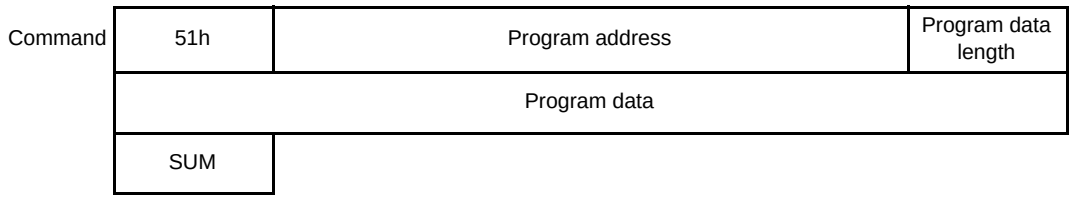


Error (1 byte): Error code
11h: SUM error
2Ah: Address error (the address is not in the selected area.)
53h: Program error (the data cannot be programmed.)

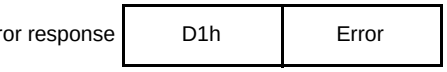
31.10.8.3 Data Area Program

This command is used to program the specified data to the data area. Set the lower 2 bits to 0 for the program address selected in this command. When the data length is shorter than 4 bytes, the data cannot be programmed. Fill the gaps with FFh.
When the program from the selected address is successfully completed, the MCU sends a response (06h). If the SUM of the received command does not match or an error occurs during a program operation, the MCU sends an error response.

To enter the program/erase host command wait state after the program operation ends, send 51h FFh FFh FFh FFh 00h B3h from the host. The MCU sends a response (06h), and enters the program/erase host command wait state.



Program address (4 bytes): Address for program destination
Set the lower 2 bits of the selected address to 0
Set FFFF FFFFh for end of data area program
Program data length (1 byte): Size of program data
Set 4-byte data
Set 00h for end of data area program
Program data (n bytes): Program data for the data area (n = program data length, 0 for end of program)
Set data of the program data length
When the program is less than n bytes, set FFh for the missing data.
No program data for the end of data area program
SUM (1 byte): Value that is calculated so the sum of command data is 00h



Error (1 byte): Error code
11h: SUM error
2Ah: Address error (the address is not in the selected area.)
2Bh: Program data length error
53h: Program error (the data or program data cannot be programmed.)

31.10.8.4 Erase Preparation

This command is used to prepare for accepting the block erase command.
When the MCU receives this command, it recognizes that an instruction to prepare for the erase command is issued from the host. Then, the MCU enters the erase wait state, where only the block erase command can be accepted, and sends a response (06h).

Command	48h
Response	06h

31.10.8.5 Block Erase

This command is used to erase the selected block in the user area or data area.
Specify the block start address selected in the command by calculating the address based on the response to the block information inquiry command.
When the block selected in the block start address is successfully erased, the MCU sends a response (06h). If the SUM of the received command does not match or an error occurs during an erase operation, the MCU sends an error response.

To enter the program/erase host command wait state after the erase operation ends, send 59h 04h FFh FFh FFh FFh A7h from the host. The MCU enters the program/erase host command wait state and sends a response (06h).

Command	59h	Size
	Block start address	
Response	SUM	

Size (1 byte): Total bytes of Block start address (the value is always 04h)
Block start address (4 bytes): Start address of the block that is erased
Set FFFF FFFFh for end of erase
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Response	06h	
Error response	D9h	Error

Error (1 byte): Error code
11h: SUM error
29h: Block start address error
51h: Erase error (the selected block cannot be erased)

31.10.9 Read-Check Commands

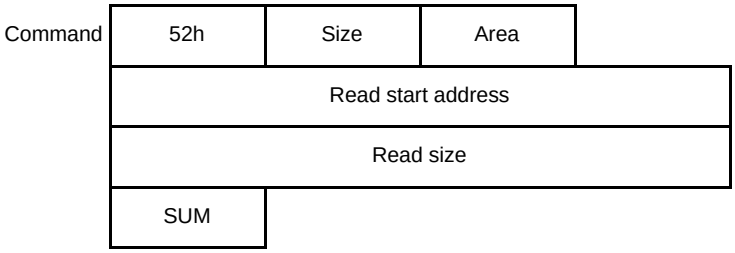
Read-check commands are used to read data or check whether data is programmed in the user area or data area in the MCU based on the response to inquiry commands.
Table 31.19 lists read-check commands used in the program/erase host command wait state.

Table 31.19 Read-Check Commands

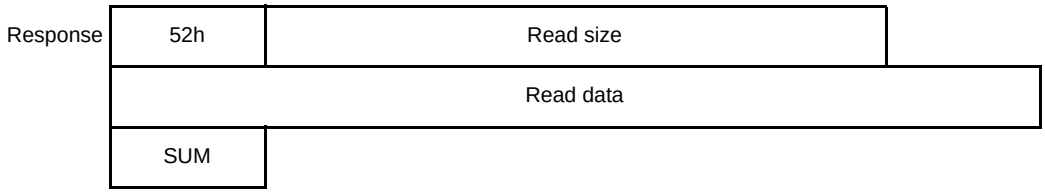
Command	Function
Memory read	Read data from the user area or data area.
User area checksum	Obtain the checksum of the entire user area.
Data area checksum	Obtain the checksum of the entire data area.
User area blank check	Check whether data is programmed in the user area.
Data area blank check	Check whether data is programmed in the data area.
Access window information program	Set the access window.
Access window read	Read the settings of the access window.

31.10.9.1 Memory Read

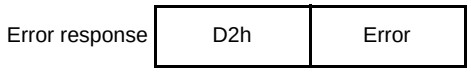
This command is used to read data programmed in the user area or data area.
For a read start address selected in the command, set a value within the range from the area start address to the area end address received in the response to the user area information inquiry command or the data area information inquiry command.
For a read size selected in the command, set a value so the sum of the read start address and the read size is within the range from the area start address to the area end address received in the response to the user area information inquiry command or the data area information inquiry command.
When the MCU performs a read successfully, it sends data of the specified range. If the SUM of the received command does not match or the MCU fails to perform a read successfully, it sends an error response.



Size (1 byte): Total bytes for Read start address and Read size
Area (1 byte): Area that is read
01h: User area or data area
Read start address (4 bytes): Start address of the area that is read
Read size (4 bytes): Size of data that is read (in bytes)
SUM (1 byte): Value that is calculated so the sum of response data is 00h



Read size (4 bytes): Size of Data that is read (in bytes)
Read data (n bytes): Data read from the specified range (n = read size)
SUM (1 byte): Value that is calculated so the sum of response data is 00h

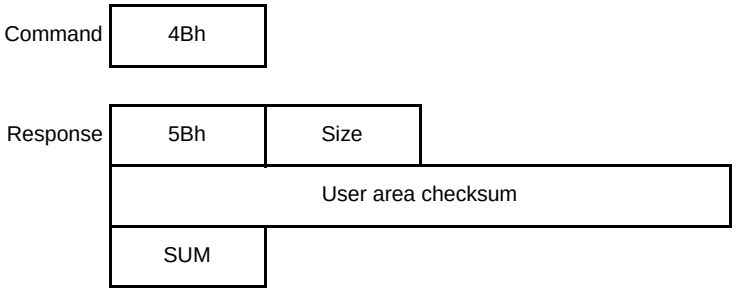


Error (1 byte): Error code
11h: SUM error
2Ah: Address error
- A value other than 01h is set for the “Area” field.
- The read start address is not in the selected area.

2Bh: Size error
- The read size is set to 0000 0000h.
- The read size exceeds the area size.
- The address calculated from the read start address and read size is not in the selected area.

31.10.9.2 User Area Checksum

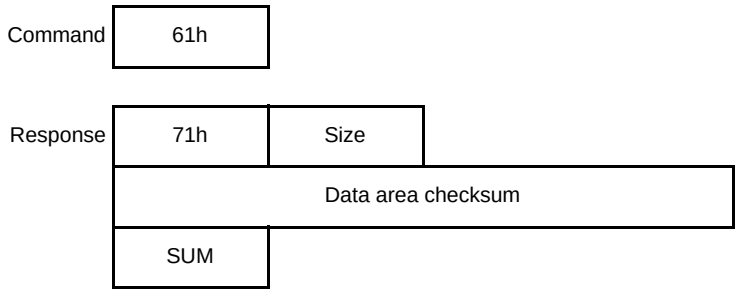
This command used to obtain the checksum of the entire user area.
When the MCU receives this command, it adds data from the start address to the end address in bytes in the user area, and sends the calculated result (checksum) as a response.



Size (1 byte): Number of bytes for checksum of the user area (the value is always 04h)
User area checksum (4 bytes): Calculated result of the data in the user area in bytes
SUM (1 byte): Value that is calculated so the sum of response data is 00h

31.10.9.3 Data Area Checksum

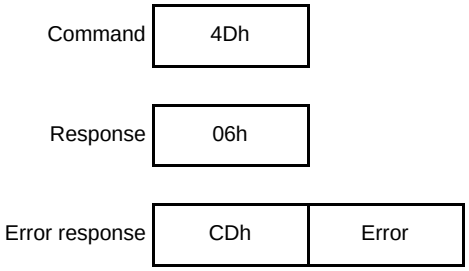
This command used to obtain the checksum of the entire data area.
When the MCU receives this command, it adds data from the start address to the end address in bytes in the data area, and sends the calculated result (checksum) as a response.



Size (1 byte): Number of bytes for checksum of the data area (the value is always 04h)
Data area checksum (4 bytes): Calculated result of the data in the data area in bytes
SUM (1 byte): Value that is calculated so the sum of response data is 00h

31.10.9.4 User Area Blank Check

This command is used to check whether data is programmed in the user area.
When the MCU receives this command, it checks whether there is data in the entire user area. If there is no programmed data, the MCU sends a response (06h). If there is at least 1 byte of programmed data, the MCU sends an error response.



Error (1 byte): Error code
52h: Not blank

31.10.9.5 Data Area Blank Check

This command is used to check whether data is programmed in the user area.
When the MCU receives this command, it checks whether there is programmed data in the entire user area. If there is no programmed data, the MCU sends a response (06h). If there is at least 1 byte of programmed data, the MCU sends an error response.

Command	62h	
Response	06h	
Error response	E2h	Error

Error (1 byte): Error code
52h: Not blank

31.10.9.6 Access Window Information Program

This command is used to set the access window used for area protection.
For the access window start address selected in the command, set the start address of the start block. For the access window end address, set the end address of the end block.
When the specified access window settings are successfully completed, the MCU sends a response (06h). If the SUM of the received command does not match or an error occurs during the access window settings, the MCU sends an error response.
For details on the access window, see section 31.6, Area Protection.

Command	74h	05h	Access window	
	Access window start address LH	Access window start address HL	Access window end address LH	Access window end address HL
	SUM			

Access window (1 byte): Set the access window or clear the access window settings
Set 00h to set the access window
Set FFh to clear the access window settings
Access window start address LH (1 byte): Start address of the access window (A15 to A8)
Set A15 to A8 of the start address of the start block.
Set FFh to clear the access window settings
Access window start address HL (1 byte): Start address of the access window (A23 to A16)
Set A23 to A16 of the start address of the start block.
Set FFh to clear the access window settings
Access window end address LH (1 byte): End address of the access window (A15 to A8)
Set A15 to A8 of the end address of the end block.
Set FFh to clear the access window settings
Access window end address HL (1 byte): End address of the access window (A23 to A16)
Set A23 to A16 of the end address of the end block.
Set FFh to clear the access window settings
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Response	06h	
Error response	F4h	Error

Error (1 byte): Error code
11h: SUM error
2Ah: Address error (specified address is not in the area)
53h: Program error (access window cannot be set)

31.10.9.7 Access Window Read

This command is used to check the set range of the access window.
When the MCU successfully obtains the access window range, the MCU sends the access window start address and end address that it read. If the SUM of the received command does not match, the MCU sends an error response.

Command	73h	01h	FFh	8Dh

Response	73h	05h		
	Access window start address LH	Access window start address HL	Access window end address LH	Access window end address HL
	FFh			
	SUM			

Access window start address LH (1 byte): Start address of the access window range (A15 to A8)
Access window start address HL (1 byte): Start address of the access window range (A23 to A16)
Access window end address LH (1 byte): End address of the access window range (A15 to A8)
Access window end address HL (1 byte): End address of the access window range (A23 to A16)
SUM (1 byte): Value that is calculated so the sum of response data is 00h

Error response	F3h	Error

Error (1 byte): Error code
11h: SUM error

31.11 Serial Programmer Operation in Boot Mode (SCI Interface)

The following describes the procedure for the serial programmer to program/erase the user area and data area in boot mode (SCI Interface).

1. Automatically adjust the bit rate
2. Receive the MCU information*1
3. Select the device and change the bit rate
4. Enter the program/erase host command wait state
5. Unlock boot mode ID code protection
6. Erase the user area and data area*2, *3
7. Program the user area and data area*2, *3
8. Check data in the user area*2
9. Check data in the data area*2
10. Set the access window in the user area*2
11. Reset the MCU

Note 1. If the necessary information has been already received, step 2 can be skipped.
Note 2. Processing steps from 6 to 10 can be proceeded as necessary, and their order can be changed.
Note 3. When a timeout occurs or invalid response data is received, stop the operation and perform step 11 (reset the MCU).

Refer to section 31.10.5, Inquiry Commands, section 31.10.6, Setting Commands, section 31.10.7, ID Code Authentication Command, section 31.10.8, Program/Erase Commands, and section 31.10.9, Read-Check Commands for details on the commands used in the above steps 2 to 10.

31.11.1 Bit Rate Automatic Adjustment

The MCU measures the low width of data 00h that is sent from the serial programmer at 9,600 or 19,200 bps to automatically adjust the bit rate.

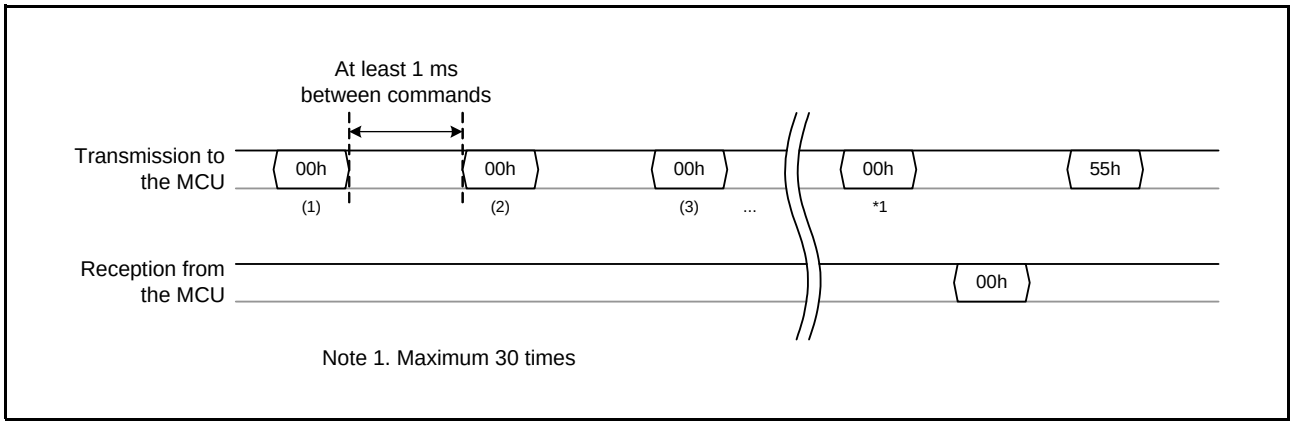


Figure 31.27 Transmit/Receive Data for Bit Rate Automatic Adjustment

After starting up in boot mode, wait for at least 400 ms and then send 00h to the MCU from the serial programmer. When the bit rate adjustment is completed, the MCU sends 00h to the programmer. When the programmer receives 00h, send 55h to the MCU from the programmer. When the programmer can not receive 00h, wait for at least 1 ms and send 00h to the MCU again. When the programmer fails to receive 00h even if it send 00h 30 times, restart the MCU in boot mode and perform the automatic adjustment for the bit rate again.
When the MCU receives 55h, the MCU sends E6h and enters the inquiry/setting command wait state. If the MCU fails to receive 55h, the MCU sends FFh. When the programmer receives FFh, restart the MCU in boot mode, and perform the automatic adjustment for the bit rate again.

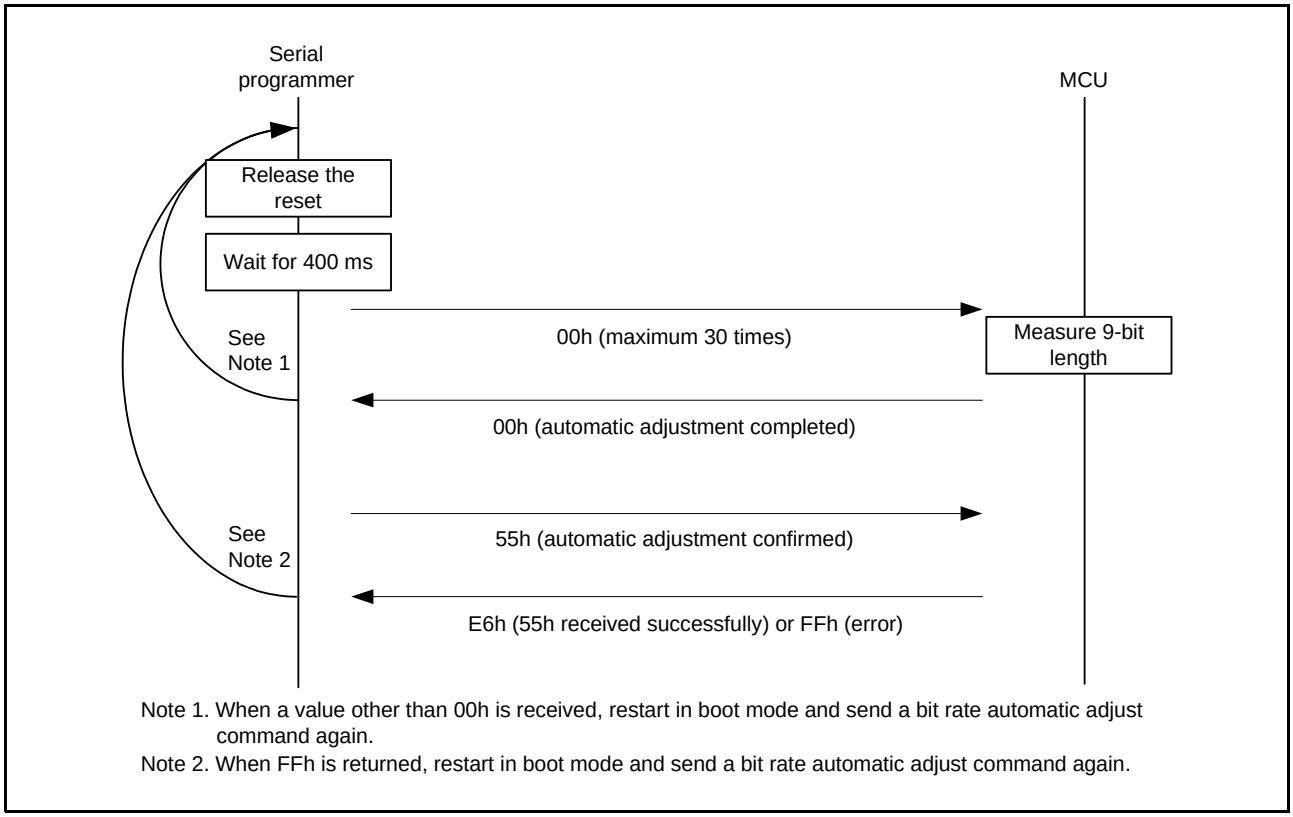


Figure 31.28 Bit Rate Automatic Adjustment Procedure

31.11.2 Receive the MCU Information

Procedure to send inquiry commands, and receive the information necessary to send setting commands, program/erase commands, and read-check commands is as follows.

(1) Send a support device inquiry command (20h) to check what type of endianness the MCU supports. The MCU returns all device codes and series names that it supports.
(2) Send a user area information inquiry command (25h) to check the start and end addresses of the user area. The MCU returns the start and end addresses of the user area.
(3) Send a block information inquiry command (26h) to check the block configuration. The MCU returns the start address, the size of one block, and the number of blocks for the user area and data area.
(4) Send a data area information inquiry command (2Bh) to check the start and end addresses of the data area. The MCU returns the start and end addresses of the data area.

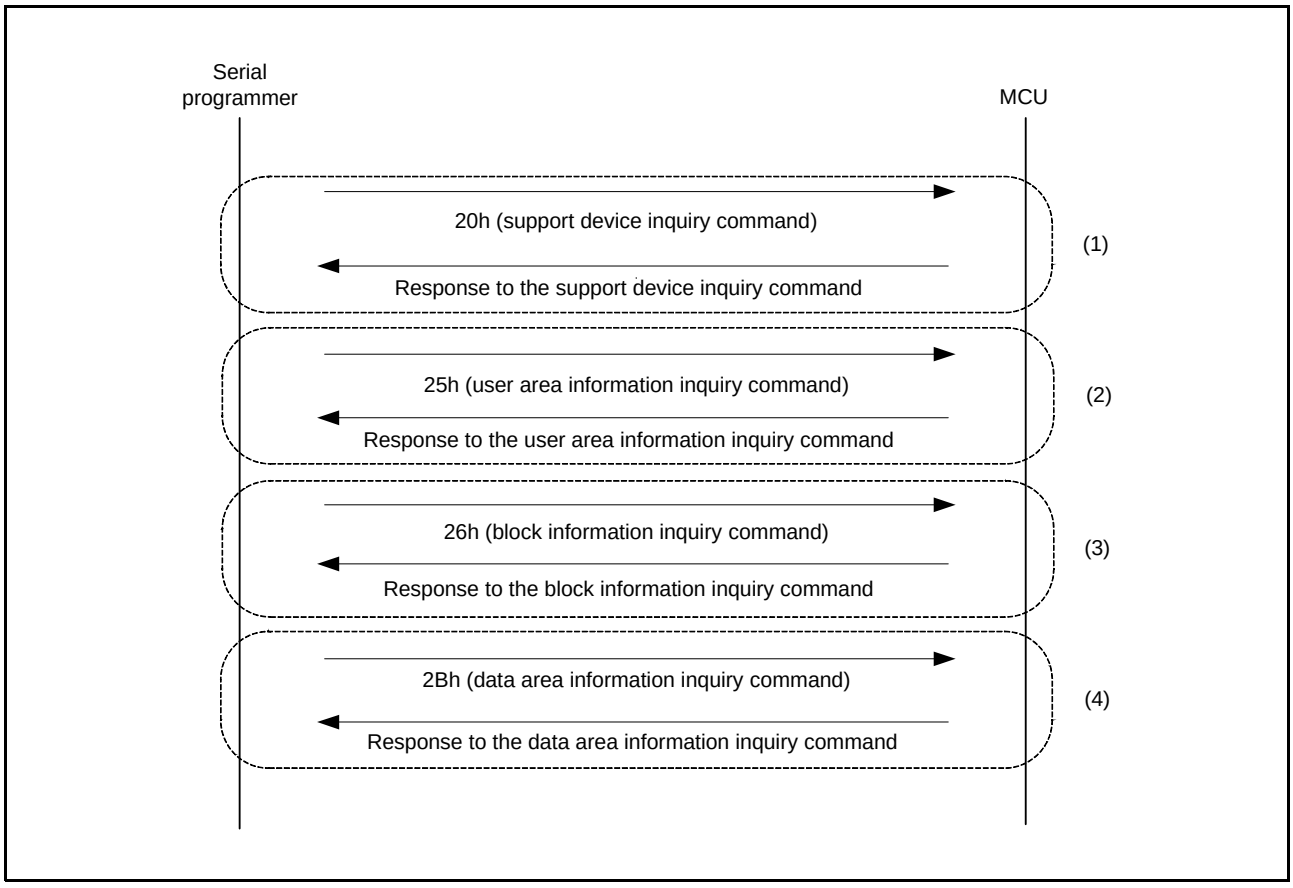


Figure 31.29 Procedure to Receive the MCU Information

31.11.3 Select the Device and Change the Bit Rate

Procedure to select the device to connect with the serial programmer and to change the bit rate for communication is as follows.

(1) Send the device select command (10h). Select the device code according to the endian of developed software.
(2) Send the operating frequency select command (3Fh) to change the communication bit rate from 9,600 or 19,200 bps.

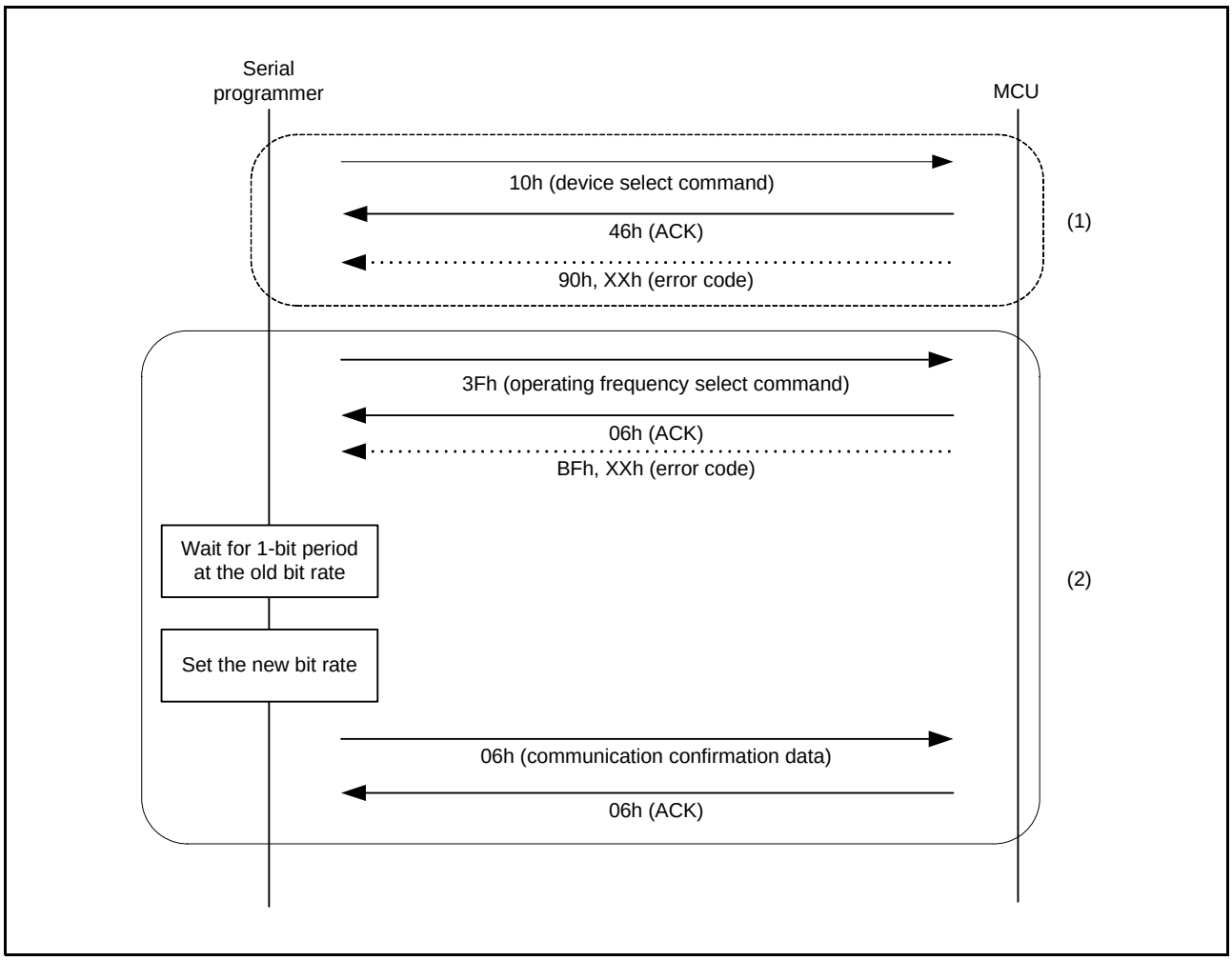


Figure 31.30 Procedure to Select the Device and Change the Bit Rate

31.11.4 Enter the Program/Erase Host Command Wait State

Send the program/erase host command wait state transition command to perform program/erase operations. The MCU sends a response according to whether boot mode ID code protection is enabled or disabled.

(1) When boot mode ID code protection is disabled, the MCU sends a response (06h), and enters the program/erase host command wait state. Use the serial programmer to start from the operation described in section 31.11.6, Erase the User Area and Data Area.

(2) When the boot mode ID code protection is enabled, the MCU sends a response (16h), and enters the ID code authentication wait state. Use the serial programmer to start from the operation described in section 31.11.5, Unlock Boot Mode ID Code Protection.

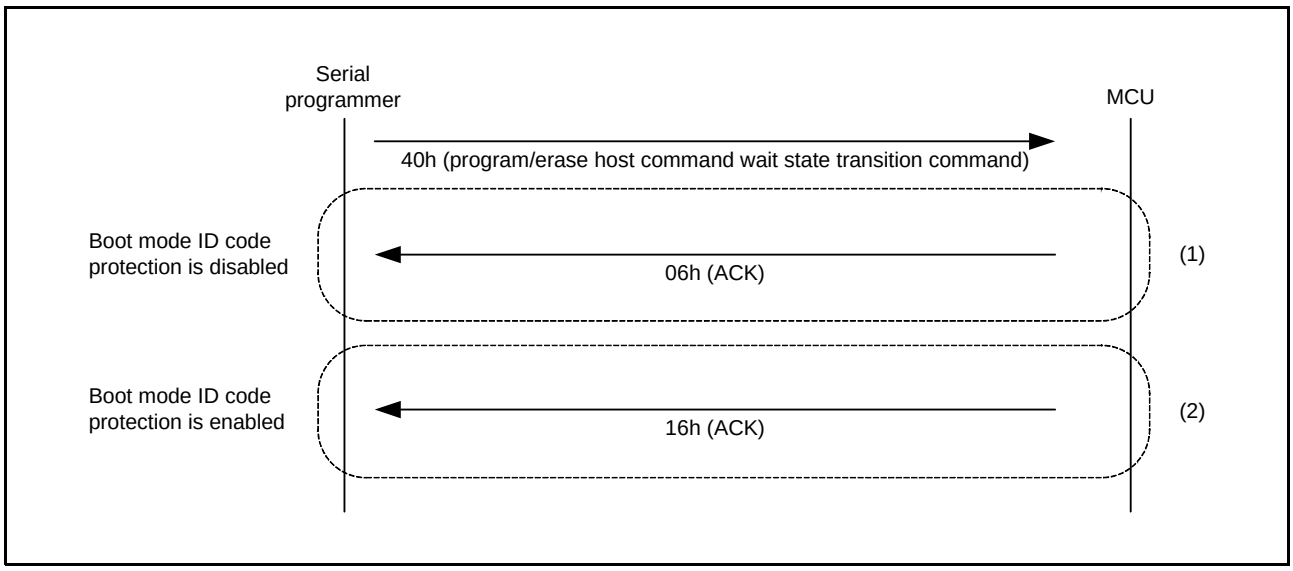


Figure 31.31 Procedure to Transition to the Program/Erase Host Command Wait State

31.11.5 Unlock Boot Mode ID Code Protection

Send the ID code check command to unlock boot mode ID code protection.

(1) When ID codes match, the MCU enters the program/erase host command wait state. Data in the user area and data area are not erased. Use the serial programmer to start from the operation described in section 31.11.6, Erase the User Area and Data Area.

(2) If ID codes do not match consecutively, the MCU remains in the boot mode ID code authentication state. Reset the MCU, and then use the serial programmer to start again from section 31.11.1, Bit Rate Automatic Adjustment.

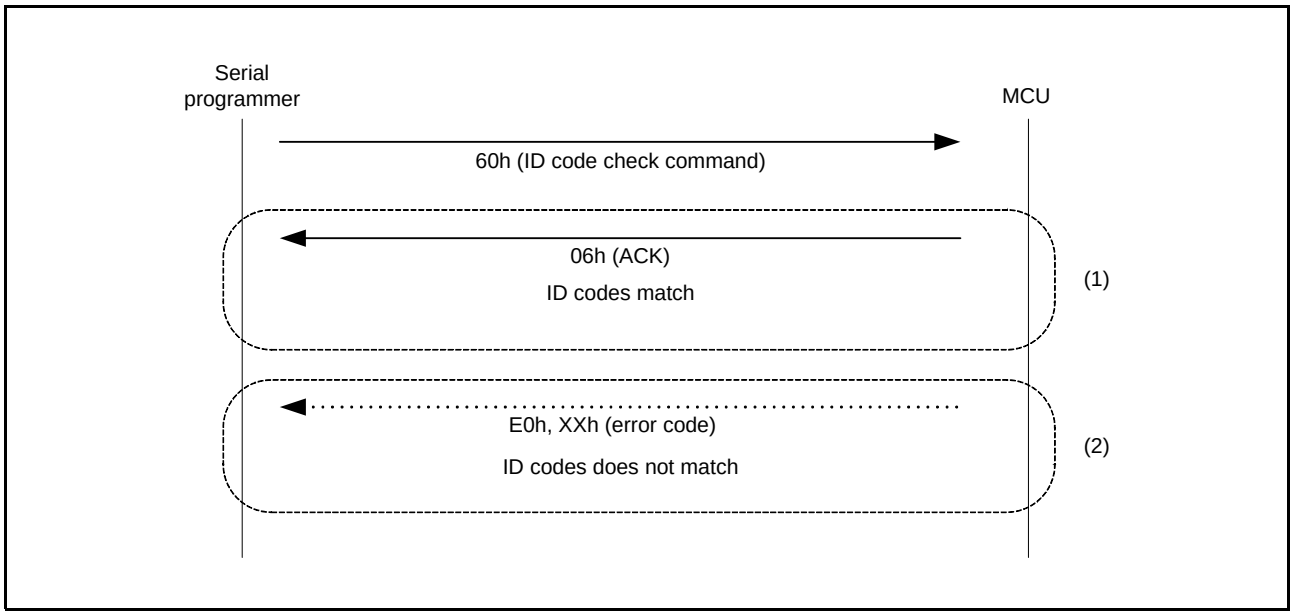


Figure 31.32 Procedure to Unlock ID Code Protection

31.11.6 Erase the User Area and Data Area

Procedure to erase blocks that are programmed in the user area and data area to program a user program and data is as follows.

(1) Send an erase preparation command (48h).
(2) Send a block erase command (59h).
(3) To place the MCU in the program/erase host command wait state, send a block erase command for ending the erasure (59h 04h FFh FFh FFh FFh A7h).

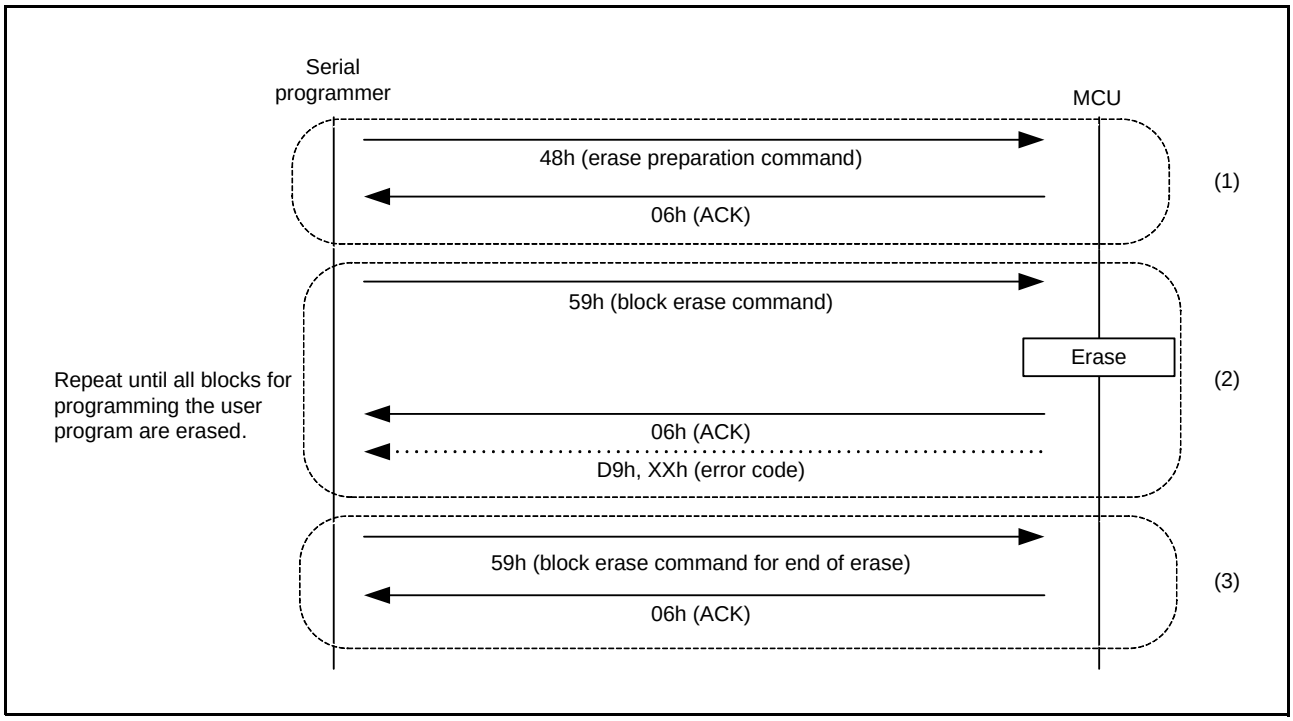


Figure 31.33 Procedure to Erase the User Area and Data Area

31.11.7 Program the User Area and Data Area

Procedure to program a user program and data in the user area and data area is as follows.

(1) Send the user/data area program preparation command (43h).
(2) Send the program command (50h) or the data area program command (51h).
(3) To place the MCU in the program/erase host command wait state, send the program command (50h FFh FFh FFh FFh B4h) or the data area program command (51h FFh FFh FFh FFh 00h B3h) for ending the programming.

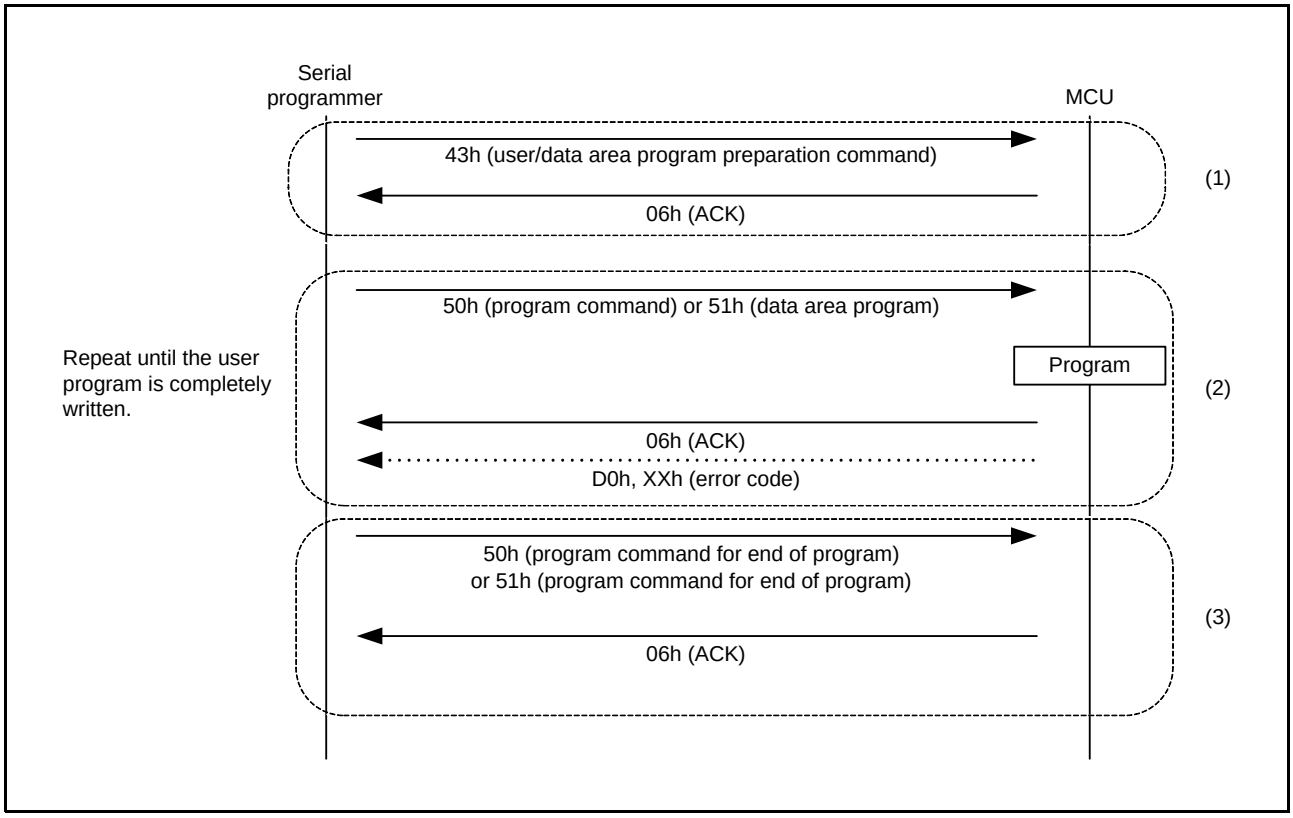


Figure 31.34 Procedure to Program the User Area and Data Area

31.11.8 Check Data in the User Area

Procedure to read and check, checksum, and blank check the user area to check the programmed data in the user area is as follows.

(1) The read and check operation is used to read data in the user area and compare the read data with the programmed data to check if the program operation is performed successfully. Send a memory read command (52h) to read data in the user area.
(2) Send the user area checksum command (4Bh) to check program data using the checksum of user area.
(3) Send a user area blank check command (4Dh) to check if the user area has data.

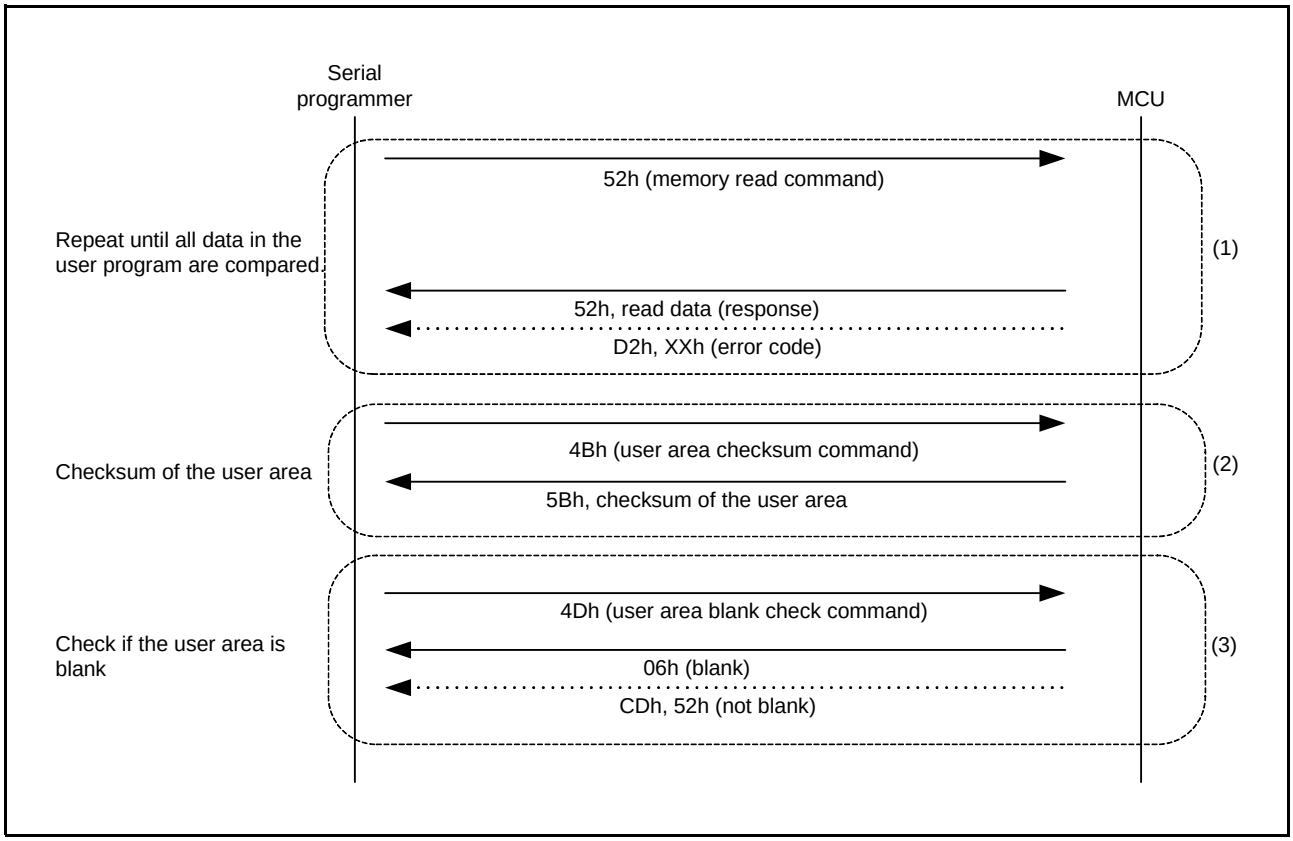


Figure 31.35 Procedure to Check Data in the User Area

31.11.9 Check Data in the Data Area

Procedure to read and check, checksum, and blank check the data area to check the programmed data in the data area is as follows.

(1) The read and check operation is used to read data in the data area and compare the read data with the programmed data to check if the program operation is performed successfully. Send a memory read command (52h) to read data in the data area.
(2) Send the data area checksum command (61h) to check program data using the checksum of data area.
(3) Send the data area blank check command (62h) to check if the data area has data.

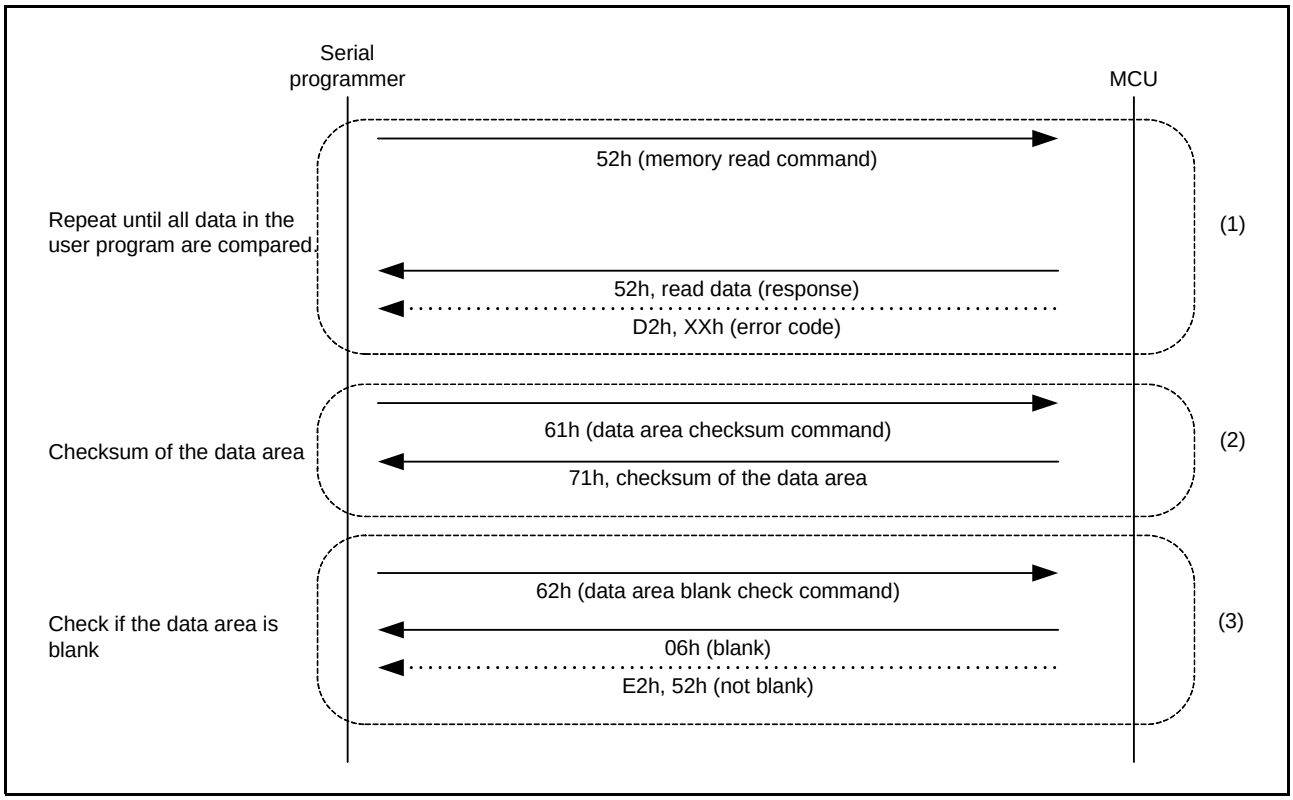


Figure 31.36 Procedure to Check Data in the Data Area

31.11.10 Set the Access Window in the User Area

Procedure to set the access window to avoid unintentionally rewriting the user area during the self-programming is as follows.

(1) Send the access window program command (74h) to set the access window settings.
(2) Send the access window read command (73h) to confirm the access window settings.

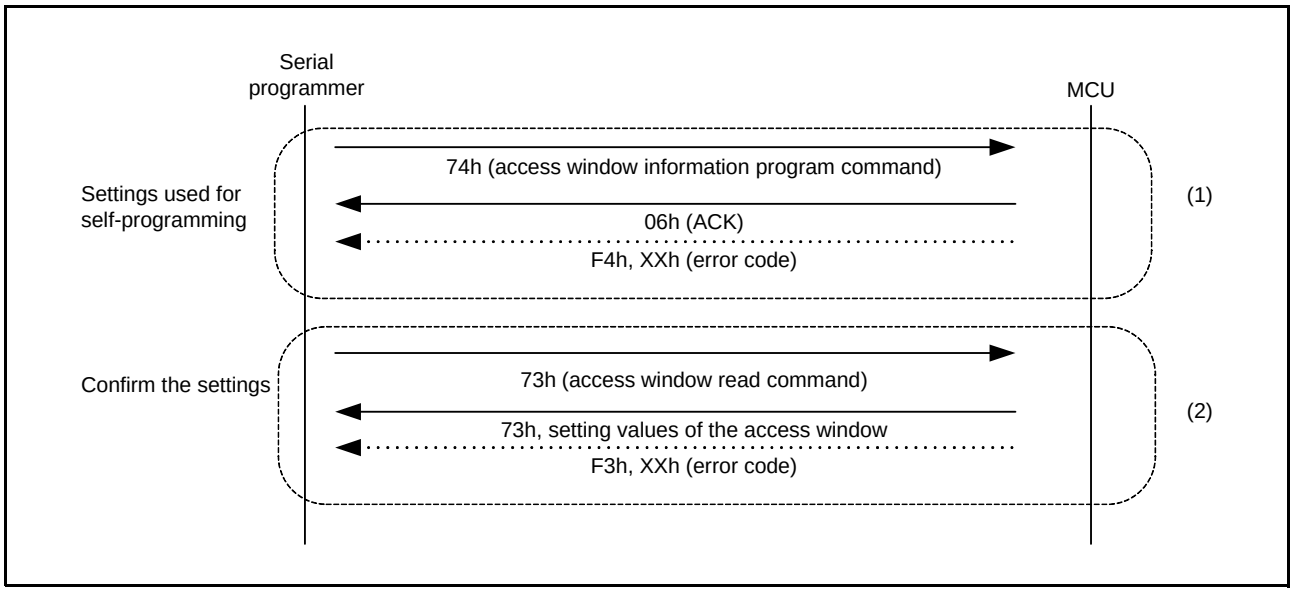
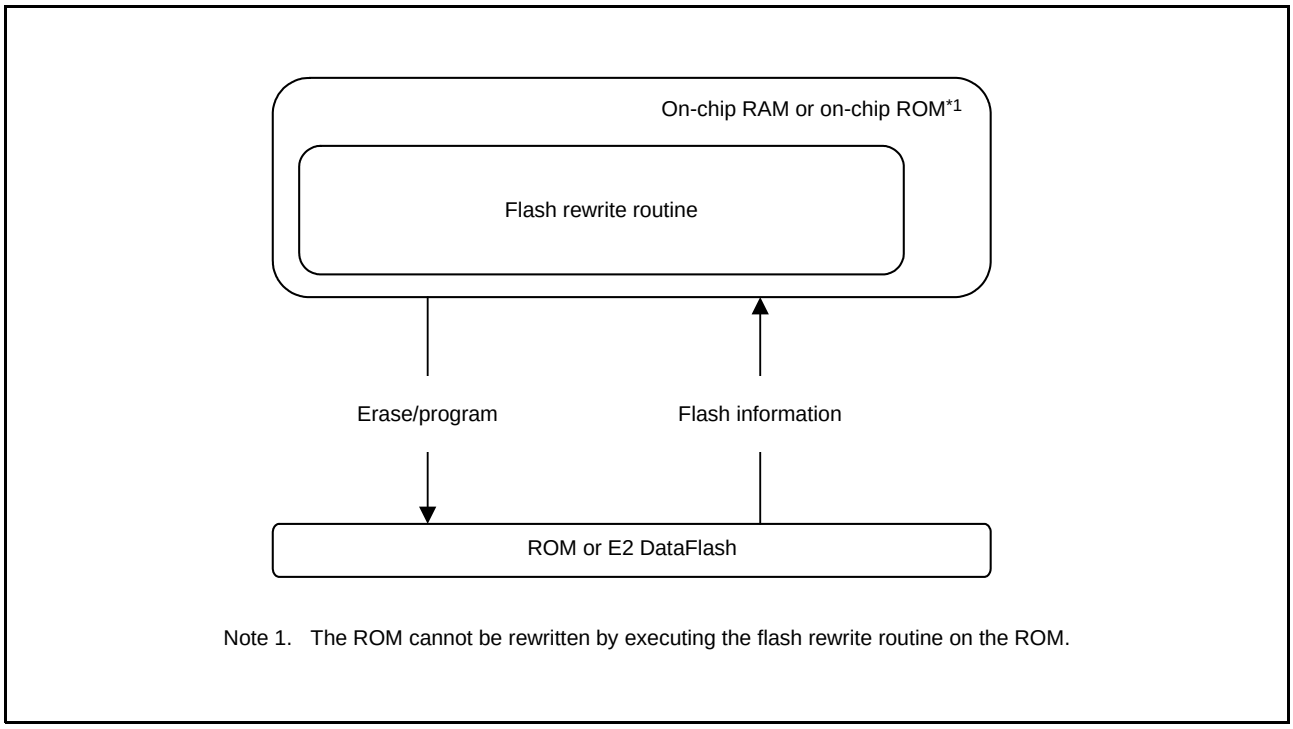


Figure 31.37 Procedure to Set the Access Window in the User Area

31.12 Rewriting by Self-Programming

31.12.1 Overview

The MCU supports rewriting of the flash memory by the user program. The ROM and E2 DataFlash can be rewritten by preparing a routine to rewrite the flash memory (flash rewrite routine) in the user program.
When rewriting the E2 DataFlash, the BGO can be used to execute the flash rewrite routine on the ROM. The E2 DataFlash can also be rewritten by executing the flash rewrite routine that is transferred on the RAM in advance.



Note 1. The ROM cannot be rewritten by executing the flash rewrite routine on the ROM.

Figure 31.38 Self-Programming Overview

31.13 Usage Notes

(1) Access the Block Where Erase Operation is Forcibly Stopped
When forcibly stopping an erase operation, data in the block where the erase operation is aborted is undefined. To avoid malfunctions caused by reading undefined data, do not execute instructions or read data in the block where an erase operation is forcibly stopped.

(2) Processing After Forced Stop of Erase Operation
When an erase operation is forcibly stopped, issue a block erase command again to the same block.

(3) Additional Programming Disabled
The same address cannot be programmed more than once. When programming an area that has been already programmed, erase the area first.

(4) Reset during Program/Erase
If inputting a reset from the RES# pin, release the reset after reset input time of at least tRESW (refer to section 32, Electrical Characteristics) within the range of the operating voltage defined in the electrical characteristics.
The IWDT reset and software reset can be used regardless of tRESW.

(5) Non-maskable Interrupt Disabled during Program/Erase
When a non-maskable interrupt (NMI pin interrupt, oscillation stop detection interrupt, IWDT underflow/refresh error, voltage monitoring 1 interrupt, or voltage monitoring 2 interrupt) occurs during a program/erase operation, the vectors are fetched from the ROM, and undefined data is read. Therefore, do not generate a non-maskable interrupt during a program/erase operation on the ROM.
(The description in (5) applies only to the ROM.)

(6) Location of Interrupt Vectors during a Program/Erase Operation
When an interrupt occurs during a program/erase operation, the vector may be fetched from the ROM. To avoid fetching the vector from the ROM, set the destination for fetching interrupt vectors to an area other than the ROM with the CPU interrupt table register (INTB).

(7) Abnormal Termination during Program/Erase
When the voltage exceeds the range of the operating voltage during a program/erase operation or when a program/erase operation is not completed successfully due to a reset or prohibited actions described in (8), erase the area again.

(8) Actions Prohibited during Program/Erase
To prevent the damage to the flash memory, comply with the following instructions.

- Do not use the MCU power supply that is outside the operating voltage range.
- Do not update the value of the OPCCR.OPCM[2:0] bits.
- Do not change the clock source select bit in the SCKCR3 register.
- Do not change the division ratio of the flash interface clock (FCLK).
- Do not place the MCU in deep sleep mode or software standby mode.
- Do not access the E2 DataFlash during a program/erase operation to the ROM.
- Do not change the DFLCTL.DFLEN bit value during a program/erase operation to the E2 DataFlash.

(9) FCLK during Program/Erase
For programming/erasure by self-programming, set the frequency of the FlashIF clock (FCLK), and specify an integer FCLK frequency (MHz) in FISR.PCKA[4:0] bits. Note that when the FCLK is 4 to 32 MHz, a rounded-up value should be set for a non-integer frequency such as 12.5 MHz (i.e. 12.5 MHz should be set rounded up to 13 MHz). If the FCLK is equal to or less than 4 MHz, only 1, 2, 3, or 4 MHz can be used.

31.14 Usage Notes in Boot Mode

(1) Notes on Communication Errors in Boot Mode
When communication with the MCU cannot be performed properly, reset and start up in boot mode again.

(2) Notes on Power Supply Voltage in Boot Mode (SCI Interface)
When the bit rate exceeds 500 kbps in boot mode (SCI Interface), use a voltage that is 3.0 V or higher.

(3) Notes on Option-Setting Memory in Boot Mode
The settings of option function select register 0 (OFS0), option function select register 1 (OFS1), and endian select register (MDE) are disabled in boot mode.

(4) Notes on Switching the Start-Up Area
Switch the start-up area by self-programming.

32. Electrical Characteristics

32.1 Absolute Maximum Ratings

Table 32.1 Absolute Maximum Ratings

Conditions: VSS = AVSS0 = 0 V

Item		Symbol	Value	Unit
Power supply voltage		VCC	–0.3 to +6.5	V
Input voltage	P40 to P47	V_{in}	–0.3 to AVCC0+0.3	V
	PB1, PB2 (5-V tolerant)		–0.3 to +6.5	
	Other than above		–0.3 to VCC+0.3	
Analog power supply voltage		AVCC0	–0.3 to +6.5	V
Analog input voltage	When AN000 to AN007 used	V_{AN}	–0.3 to AVCC0+0.3	V
Storage temperature		T_{stg}	–55 to +125	°C

Caution: Permanent damage to the MCU may result if absolute maximum ratings are exceeded.
To preclude malfunctions due to noise interference, insert capacitors with high frequency characteristics between the VCC and VSS pins, and between the AVCC0 and AVSS0 pins. Place capacitors with values of about 0.1 µF as close as possible to every power supply pin and use the shortest and widest possible traces for the wiring.
Connect the VCL pin to a VSS pin via a 4.7-µF capacitor. The capacitor must be placed close to the pin.
Do not input signals to ports other than 5-V tolerant ports while the device is not powered.
The current injection that results from input of such a signal may cause malfunction and the abnormal current that passes in the device at this time may cause degradation of internal elements.
Even if –0.3 to +6.5 V is input to 5-V tolerant ports, it will not cause problems such as damage to the MCU.

32.2 Recommended operating conditions

Table 32.2 Recommended Operating Conditions (1)

Item		Symbol	Min.	Typ.	Max.	Unit
Power supply voltages		VCC*1, *2	2.7	—	5.5	V
		VSS	—	0	—	
Analog power supply voltages		AVCC0*1, *2	VCC	—	5.5	V
		AVSS0	—	0	—	
Operating temperature	D version	T_{opr}	–40	—	85	°C
	G version		–40	—	105	

Note 1. AVCC0 and VCC can be set individually within the operating range.
Note 2. When powering on the VCC and AVCC0 pins, power them on at the same time or the VCC pin first and then the AVCC0 pin.

Table 32.3 Recommended Operating Conditions (2)

Item	Symbol	Value
Decoupling capacitance to stabilize the internal voltage	C_{VCL}	4.7 µF ±3.0%*1

Note 1. Use a multilayer ceramic capacitor whose nominal capacitance is 4.7 µF and a capacitance tolerance is ±30% or better.

32.3 DC Characteristics

Table 32.4 DC Characteristics (1)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	RIIC input pin (except for SMBus, 5-V tolerant)	V_{IH}	0.7 × VCC	—	5.8	V	
	PB1, PB2 (5-V tolerant)		0.8 × VCC	—	5.8		
	P10, P11, P22 to P24, P36, P37, P70 to P76, P93, P94, PA2, PA3, PB0, PB3 to PB7, PD3 to PD6, PE2, RES#		0.8 × VCC	—	VCC + 0.3		
	P40 to P47		0.8 × AVCC0	—	AVCC0 + 0.3		
	RIIC input pin (except for SMBus)	V_{IL}	–0.3	—	0.3 × VCC		
	P40 to P47		–0.3	—	0.2 × AVCC0		
	Other than RIIC input pin or P40 to P47		–0.3	—	0.2 × VCC		
Schmitt trigger input hysteresis	RIIC input pin (except for SMBus)	ΔV_T	0.05 × VCC	—	—		
	P40 to P47		0.1 × AVCC0	—	—		
	Other than RIIC input pin or P40 to P47		0.1 × VCC	—	—		
Input level voltage (except for Schmitt trigger input pins)	MD	V_{IH}	0.9 × VCC	—	VCC + 0.3	V	
	EXTAL (external clock input)		0.8 × VCC	—	VCC + 0.3		
	RIIC input pin (SMBus)		2.1	—	VCC + 0.3		
	MD	V_{IL}	–0.3	—	0.1 × VCC		
	EXTAL (external clock input)		–0.3	—	0.2 × VCC		
	RIIC input pin (SMBus)		–0.3	—	0.8		

Table 32.5 DC Characteristics (2)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input leakage current	RES#, MD, PE2	$\lvert I_{in} \rvert$	—	—	1.0	μA	V_{in} = 0 V, VCC
Three-state leakage current (off-state)	P40 to P47	$\lvert I_{TSI} \rvert$	—	—	1.0	μA	V_{in} = 0 V, AVCC0
	PB1, PB2 (5-V tolerant)		—	—	1.0		V_{in} = 0 V, 5.8 V
	Other than above		—	—	0.2		V_{in} = 0 V, VCC
Input capacitance	All input pins	C_{in}	—	4	15	pF	V_{in} = 0 mV, f = 1 MHz, T_a = 25°C
Input pull-up resistor	All ports (except for PE2)	R_U	10	20	50	kΩ	V_{in} = 0 V

Table 32.6 DC Characteristics (3)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item					Symbol	Typ.*4	Max.	Unit	Test Conditions
Supply current *1	High-speed operating mode	Normal operating mode	No peripheral operation*2	ICLK = 32 MHz	I_{CC}	3.1	—	mA	
				ICLK = 16 MHz		2.1	—		
				ICLK = 8 MHz		1.6	—		
			All peripheral operation: Normal *3	ICLK = 32 MHz		10.6	—		
				ICLK = 16 MHz		6.0	—		
				ICLK = 8 MHz		3.6	—		
			All peripheral operation: Max.*3	ICLK = 32 MHz		—	18.1		
		Sleep mode	No peripheral operation*2	ICLK = 32 MHz		1.5	—		
				ICLK = 16 MHz		1.2	—		
				ICLK = 8 MHz		1.0	—		
			All peripheral operation: Normal *3	ICLK = 32 MHz		5.6	—		
				ICLK = 16 MHz		3.3	—		
				ICLK = 8 MHz		2.1	—		
		Deep sleep mode	No peripheral operation*2	ICLK = 32 MHz		1.0	—		
				ICLK = 16 MHz		0.9	—		
				ICLK = 8 MHz		0.8	—		
			All peripheral operation: Normal *3	ICLK = 32 MHz		3.8	—		
				ICLK = 16 MHz		2.3	—		
				ICLK = 8 MHz		1.6	—		
		Increase during BGO operation*5				2.5	—		
	Middle-speed operating modes	Normal operating mode	No peripheral operation*6	ICLK = 12 MHz		1.9	—		
				ICLK = 8 MHz		1.3	—		
				ICLK = 1 MHz		0.3	—		
			All peripheral operation: Normal*7	ICLK = 12 MHz		4.8	—		
				ICLK = 8 MHz		3.3	—		
				ICLK = 1 MHz		0.9	—		
			All peripheral operation: Max.*7	ICLK = 12 MHz		—	8.2		
		Sleep mode	No peripheral operation*6	ICLK = 12 MHz		1.2	—		
				ICLK = 8 MHz		0.7	—		
				ICLK = 1 MHz		0.2	—		
			All peripheral operation: Normal*7	ICLK = 12 MHz		2.8	—		
				ICLK = 8 MHz		1.9	—		
				ICLK = 1 MHz		0.7	—		
		Deep sleep mode	No peripheral operation*6	ICLK = 12 MHz		1.0	—		
				ICLK = 8 MHz		0.6	—		
				ICLK = 1 MHz		0.1	—		
			All peripheral operation: Normal*7	ICLK = 12 MHz		2.1	—		
				ICLK = 8 MHz		1.5	—		
				ICLK = 1 MHz		0.6	—		
		Increase during BGO operation*5				2.5	—		

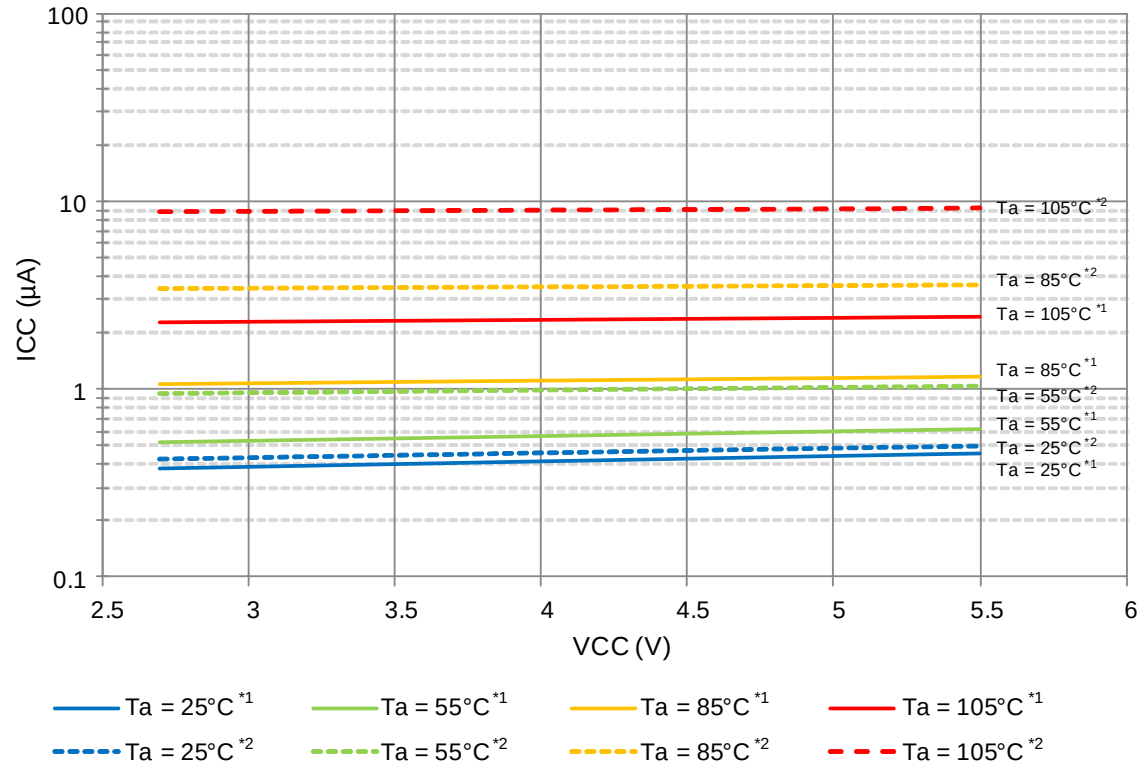
Note 1. Supply current values do not include output charge/discharge current from all pins. The values apply when internal pull-up resistors are disabled.
Note 2. Peripheral module clocks are stopped. This does not include BGO operation. The clock source is the PLL. FCLK and PCLK are set for division by 64.
Note 3. Peripheral module clocks are supplied. This does not include BGO operation. The clock source is the PLL. The FCLK and PCLK operating clocks run at the same frequency as ICLK.
Note 4. Values when VCC = 5 V.
Note 5. This is an increase caused by program/erase operation to the ROM or E2 DataFlash during executing the user program.
Note 6. Peripheral module clocks are stopped. The clock source is the PLL when ICLK is 12 MHz, is the HOCO when the ICLK is at 8 MHz, or is the LOCO when the ICLK is at another frequency. FCLK and PCLK are set for division by 64.
Note 7. Peripheral module clocks are supplied. The clock source is the PLL when ICLK is 12 MHz, is the HOCO when the ICLK is at 8 MHz, or is the LOCO when the ICLK is at another frequency. The FCLK and PCLK operating clocks run at the same frequency as ICLK.

Table 32.7 DC Characteristics (4)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

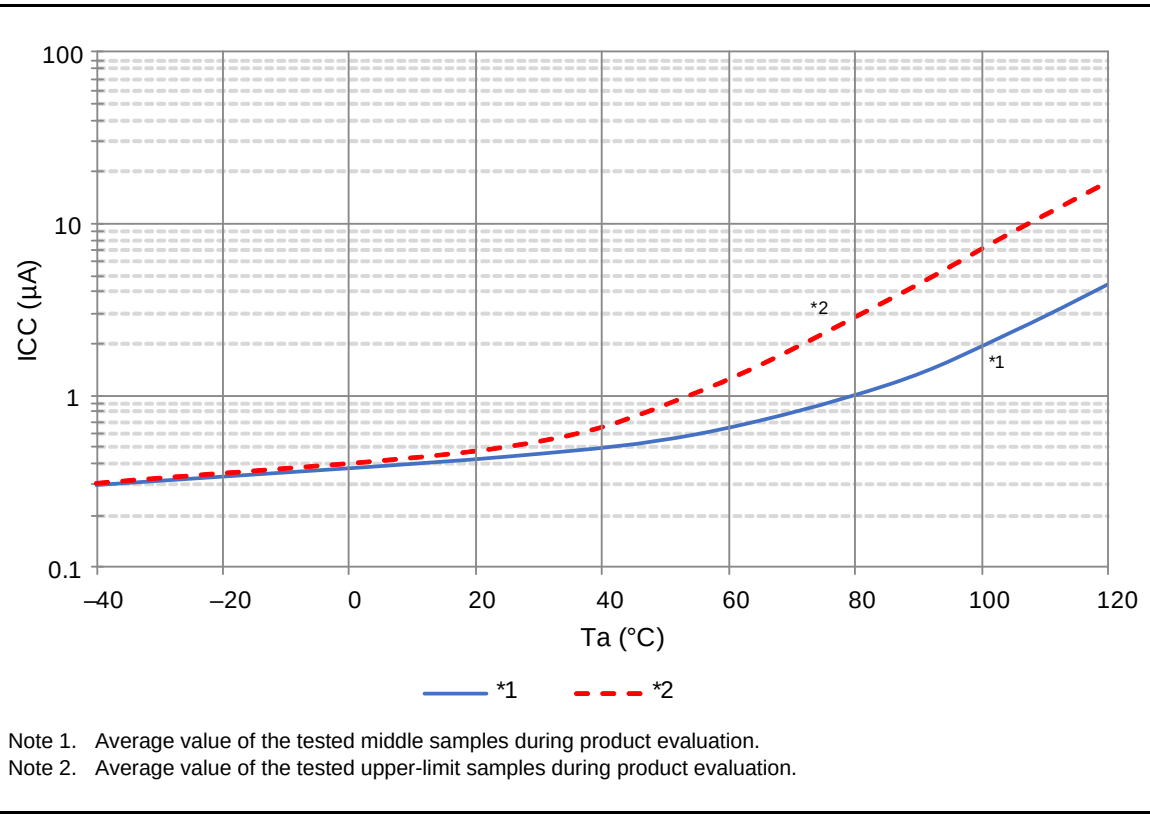
Item			Symbol	Typ.*3	Max.	Unit	Test Conditions
Supply current*1	Software standby mode*2	T_a = 25°C	I_{CC}	0.44	0.74	µA	
		T_a = 55°C		0.60	1.78		
		T_a = 85°C		1.16	8.36		
		T_a = 105°C		2.38	20.49		

Note 1. Supply current values are with all output pins unloaded and all input pull-up resistors are disabled.
Note 2. The IWDT and LVD are stopped.
Note 3. VCC = 5 V.



Note 1. Indicates the average of the typical samples through actual measurement during product evaluation.
Note 2. Indicates the average of the upper-limit samples through actual measurement during product evaluation.

Figure 32.1 Voltage Dependency in Software Standby Mode (Reference Data)



Note 1. Average value of the tested middle samples during product evaluation.
Note 2. Average value of the tested upper-limit samples during product evaluation.

Figure 32.2 Temperature Dependency in Software Standby Mode (Reference Data)

Table 32.8 DC Characteristics (5)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Typ.	Max.	Unit	Test Conditions
Permissible total consumption power*1	Pd	—	300	mW	D-version product
Permissible total consumption power*1	Pd	—	105	mW	G-version product

Note: Please contact a Renesas Electronics sales office for information on the derating of the G-version product. Derating is the systematic reduction of load for the sake of improved reliability.

Note 1. Total power dissipated by the entire chip (including output currents)

Table 32.9 DC Characteristics (6)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.*2	Max.	Unit	Test Conditions
Analog power supply current	During A/D conversion (when the sample-and-hold circuit and programmable gain amplifier are in use)	I_{AVCC}	—	4.6	6.9	mA	
	During A/D conversion (when the sample-and-hold circuit is in use but the programmable gain amplifier is not)		—	3.1	4.8		
	During A/D conversion (when the sample-and-hold circuit is not in use but the programmable gain amplifier is)		—	2.5	3.9		
	During A/D conversion (when neither the sample-and-hold circuit nor the programmable gain amplifier is in use)		—	1.0	1.8		
	During D/A conversion*1		—	0.7	1.0		
	Waiting for A/D and D/A conversion (all units)		—	—	1.4	µA	
Comparator C operating current*3	Comparator enabled (per channel)	I_{CMP}	—	40	60	µA	

Note 1. The value of the D/A converter is the value of the power supply current including the reference current.

Note 2. When VCC = AVCC0 = 5 V.

Note 3. Current consumed only by the comparator C module.

Table 32.10 DC Characteristics (7)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Power-on VCC rising gradient	At normal startup	SrVCC	0.02	—	20	ms/V	
	Voltage monitoring 0 reset enabled at startup*1, *2		0.02	—	—		

Note 1. When OFS1.LVDAS = 0.

Note 2. Turn on the power supply voltage according to the normal startup rising gradient because the register settings set by OFS1 are not read in boot mode.

Table 32.11 DC Characteristics (8)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

The ripple voltage must meet the allowable ripple frequency $f_{r\,(VCC)}$ within the range between the VCC upper limit (5.5 V) and lower limit (2.7 V). When VCC change exceeds VCC ±10%, the allowable voltage change rising/falling gradient dt/dVCC must be met.

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Allowable ripple frequency	$f_{r\,(VCC)}$	—	—	10	kHz	Figure 32.3 $V_{r\,(VCC)} \leq 0.2 \times VCC$
		—	—	1	MHz	Figure 32.3 $V_{r\,(VCC)} \leq 0.08 \times VCC$
		—	—	10		Figure 32.3 $V_{r\,(VCC)} \leq 0.06 \times VCC$
Allowable voltage change rising/falling gradient	dt/dVCC	1.0	—	—	ms/V	When VCC change exceeds VCC ±10%

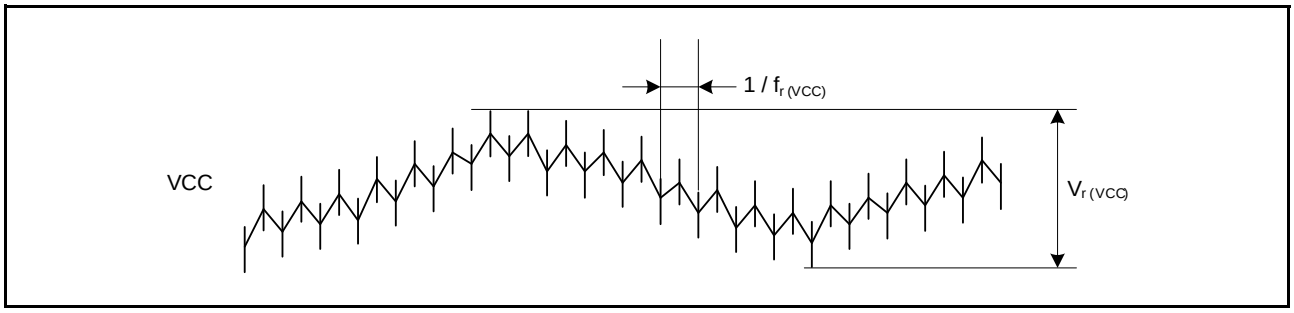


Figure 32.3 Ripple Waveform

Table 32.12 Permissible Output Currents

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Max.	Unit
Permissible low-level output current	Large current ports (P71 to P76, PB6)		I_{OL}	10.0	mA
	RIIC pins			6.0	
	Ports other than above	Normal output mode		4.0	
		High-drive output mode		8.0	
Permissible low-level output current	Total of large current ports		ΣI_{OL}	50	mA
	Total of all output pins			110	
Permissible high-level output current	Large current ports (P71 to P76, PB6)		I_{OH}	–5.0	mA
	Ports other than above	Normal output mode		–4.0	
		High-drive output mode		–8.0	
Permissible high-level output current	Total of large current ports		ΣI_{OH}	–25	mA
	Total of all output pins			–35	

Note: Do not exceed the permissible total supply current.

Table 32.13 Output Values of Voltage (1)

Conditions: VCC = 2.7 V to 4.0 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Min.	Max.	Unit	Test Conditions
Low-level output voltage	Large current ports (P71 to P76, PB6)		V_{OL}	—	0.5	V	I_{OL} = 10.0 mA
	RIIC pins	Standard mode		—	0.4		I_{OL} = 3.0 mA
		Fast mode		—	0.6		I_{OL} = 6.0 mA
	Ports other than above	Normal output mode		—	0.5		I_{OL} = 1.0 mA
		High-drive output mode		—	0.5		I_{OL} = 2.0 mA
High-level output voltage	Large current ports (P71 to P76, PB6)		V_{OH}	VCC – 0.5	—	V	I_{OH} = –5.0 mA
	P40 to P47			AVCC0 – 0.5	—		I_{OH} = –1.0 mA
	Ports other than above	Normal output mode		VCC – 0.5	—		I_{OH} = –1.0 mA
		High-drive output mode		VCC – 0.5	—		I_{OH} = –2.0 mA

Table 32.14 Output Values of Voltage (2)

Conditions: VCC = 4.0 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Min.	Max.	Unit	Test Conditions
Low-level output voltage	Large current ports (P71 to P76, PB6)		V_{OL}	—	0.8	V	I_{OL} = 10.0 mA
	RIIC pins	Standard mode		—	0.4		I_{OL} = 3.0 mA
		Fast mode		—	0.6		I_{OL} = 6.0 mA
	Ports other than above	Normal output mode		—	0.8		I_{OL} = 2.0 mA
		High-drive output mode		—	0.8		I_{OL} = 4.0 mA
High-level output voltage	Large current ports (P71 to P76, PB6)		V_{OH}	VCC – 0.8	—	V	I_{OH} = –5.0 mA
	P40 to P47			AVCC0 – 0.8	—		I_{OH} = –2.0 mA
	Ports other than above	Normal output mode		VCC – 0.8	—		I_{OH} = –2.0 mA
		High-drive output mode		VCC – 0.8	—		I_{OH} = –4.0 mA

32.3.1 Normal I/O Pin Output Voltage Characteristics

Table 32.15 Normal I/O Pin V_{OH} Voltage Characteristics (Reference Data)

Conditions: VCC = AVCC0 = 3.3 V, VSS = AVSS0 = 0 V, Ta = 25°C

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
High-level output voltage	All output pins (except for P71 to P76, PB6)	Normal output mode	V_{OH}	—	3.27	—	V	I_{OH} = –0.5 mA
				—	3.24	—		I_{OH} = –1.0 mA
				—	3.19	—		I_{OH} = –2.0 mA
				—	3.06	—		I_{OH} = –4.0 mA
				—	2.79	—		I_{OH} = –8.0 mA
		High-drive output mode	V_{OH}	—	3.29	—	V	I_{OH} = –0.5 mA
				—	3.28	—		I_{OH} = –1.0 mA
				—	3.25	—		I_{OH} = –2.0 mA
				—	3.21	—		I_{OH} = –4.0 mA
				—	3.11	—		I_{OH} = –8.0 mA
	P71 to P76, PB6	Large current Ports	V_{OH}	—	3.29	—	V	I_{OH} = –1.0 mA
				—	3.27	—		I_{OH} = –2.0 mA
				—	3.24	—		I_{OH} = –4.0 mA
				—	3.23	—		I_{OH} = –5.0 mA
				—	3.15	—		I_{OH} = –10.0 mA

Table 32.16 Normal I/O Pin V_{OH} Voltage Characteristics (Reference Data)

Conditions: VCC = AVCC0 = 5.0 V, VSS = AVSS0 = 0 V, Ta = 25°C

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
High-level output voltage	All output pins (except for P71 to P76, PB6)	Normal output mode	V_{OH}	—	4.98	—	V	I_{OH} = –0.5 mA
				—	4.96	—		I_{OH} = –1.0 mA
				—	4.92	—		I_{OH} = –2.0 mA
				—	4.83	—		I_{OH} = –4.0 mA
				—	4.65	—		I_{OH} = –8.0 mA
		High-drive output mode	V_{OH}	—	4.99	—	V	I_{OH} = –0.5 mA
				—	4.98	—		I_{OH} = –1.0 mA
				—	4.97	—		I_{OH} = –2.0 mA
				—	4.93	—		I_{OH} = –4.0 mA
				—	4.86	—		I_{OH} = –8.0 mA
	P71 to P76, PB6	Large current Ports	V_{OH}	—	4.99	—	V	I_{OH} = –1.0 mA
				—	4.98	—		I_{OH} = –2.0 mA
				—	4.96	—		I_{OH} = –4.0 mA
				—	4.95	—		I_{OH} = –5.0 mA
				—	4.89	—		I_{OH} = –10.0 mA

Table 32.17 Normal I/O Pin V_{OL} Voltage Characteristics (Reference Data)

Conditions: VCC = AVCC0 = 3.3 V, VSS = AVSS0 = 0 V, Ta = 25°C

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Low-level output voltage	All output pins (except for P71 to P76, PB6)	Normal output mode	V_{OL}	—	0.02	—	V	I_{OL} = 0.5 mA
				—	0.05	—		I_{OL} = 1.0 mA
				—	0.09	—		I_{OL} = 2.0 mA
				—	0.20	—		I_{OL} = 4.0 mA
				—	0.43	—		I_{OL} = 8.0 mA
		High-drive output mode	V_{OL}	—	0.01	—	V	I_{OL} = 0.5 mA
				—	0.02	—		I_{OL} = 1.0 mA
				—	0.04	—		I_{OL} = 2.0 mA
				—	0.08	—		I_{OL} = 4.0 mA
				—	0.16	—		I_{OL} = 8.0 mA
	P71 to P76, PB6	Large current Ports	V_{OL}	—	0.01	—	V	I_{OL} = 1.0 mA
				—	0.02	—		I_{OL} = 2.0 mA
				—	0.05	—		I_{OL} = 4.0 mA
				—	0.06	—		I_{OL} = 5.0 mA
				—	0.12	—		I_{OL} = 10.0 mA

Table 32.18 Normal I/O Pin V_{OL} Voltage Characteristics (Reference Data)

Conditions: VCC = AVCC0 = 5.0 V, VSS = AVSS0 = 0 V, Ta = 25°C

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Low-level output voltage	All output pins (except for P71 to P76, PB6)	Normal output mode	V_{OL}	—	0.02	—	V	I_{OL} = 0.5 mA
				—	0.03	—		I_{OL} = 1.0 mA
				—	0.07	—		I_{OL} = 2.0 mA
				—	0.14	—		I_{OL} = 4.0 mA
				—	0.29	—		I_{OL} = 8.0 mA
		High-drive output mode	V_{OL}	—	0.01	—	V	I_{OL} = 0.5 mA
				—	0.01	—		I_{OL} = 1.0 mA
				—	0.03	—		I_{OL} = 2.0 mA
				—	0.05	—		I_{OL} = 4.0 mA
				—	0.11	—		I_{OL} = 8.0 mA
				—	0.23	—		I_{OL} = 16.0 mA
	P71 to P76, PB6	Large current Ports	V_{OL}	—	0.01	—	V	I_{OL} = 1.0 mA
				—	0.02	—		I_{OL} = 2.0 mA
				—	0.03	—		I_{OL} = 4.0 mA
				—	0.04	—		I_{OL} = 5.0 mA
				—	0.09	—		I_{OL} = 10.0 mA

32.4 AC Characteristics

32.4.1 Clock Timing

Table 32.19 Operating Frequency Value (High-Speed Operating Mode)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit
Maximum operating frequency*4	System clock (ICLK)	f_{max}	—	—	32	MHz
	FlashIF clock (FCLK)*1, *2		—	—	32	
	Peripheral module clock (PCLKB)		—	—	32	
	Peripheral module clock (PCLKD)*3		—	—	32	

Note 1. The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note 2. The frequency accuracy of FCLK should be ±3.5%.
Note 3. The minimum frequency of PCLKD is 1 MHz when the A/D converter is to be used.
Note 4. The maximum operating frequencies do not take errors in the HOCO frequency and jitters in the PLL signal. Refer to Table 32.21, Clock Timing.

Table 32.20 Operating Frequency Value (Middle-Speed Operating Mode)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit
Maximum operating frequency*4	System clock (ICLK)	f_{max}	—	—	12	MHz
	FlashIF clock (FCLK)*1, *2		—	—	12	
	Peripheral module clock (PCLKB)		—	—	12	
	Peripheral module clock (PCLKD)*3		—	—	12	

Note 1. The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note 2. The frequency accuracy of FCLK should be ±3.5%.
Note 3. The minimum frequency of PCLKD is 1 MHz when the A/D converter is to be used.
Note 4. The maximum operating frequencies do not take errors in the HOCO frequency and jitters in the PLL signal. Refer to Table 32.21, Clock Timing.

Table 32.21 Clock Timing

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
EXTAL external clock input cycle time	t_{Xcyc}	50	—	—	ns	Figure 32.4
EXTAL external clock input high pulse width	t_{XH}	20	—	—	ns	
EXTAL external clock input low pulse width	t_{XL}	20	—	—	ns	
EXTAL external clock rise time	t_{Xr}	—	—	5	ns	
EXTAL external clock fall time	t_{Xf}	—	—	5	ns	
EXTAL external clock input wait time*1	t_{EXWT}	0.5	—	—	μs	
Main clock oscillator oscillation frequency*2	f_{MAIN}	1	—	20	MHz	
Main clock oscillation stabilization time (crystal)*2	$t_{MAINOSC}$	—	3	—	ms	Figure 32.5
Main clock oscillation stabilization time (ceramic resonator)*2	$t_{MAINOSC}$	—	50	—	μs	
LOCO clock oscillation frequency	f_{LOCO}	3.44	4.0	4.56	MHz	
LOCO clock oscillation stabilization time	t_{LOCO}	—	—	0.5	μs	Figure 32.6
IWDT-dedicated clock oscillation frequency	f_{ILOCO}	12.75	15	17.25	kHz	
IWDT-dedicated clock oscillation stabilization time	t_{ILOCO}	—	—	50	μs	Figure 32.7
HOCO clock oscillation frequency	f_{HOCO}	31.52	32	32.48	MHz	T_a = –40 to +85°C
		31.68	32	32.32		T_a = –20 to +85°C
		31.36	32	32.64		T_a = –40 to +105°C
HOCO clock oscillation stabilization time	t_{HOCO}	—	—	41.3	μs	Figure 32.9
PLL circuit oscillation frequency	f_{PLL}	24	—	32	MHz	
PLL clock oscillation stabilization time	t_{PLL}	—	—	74.4	μs	Figure 32.10
PLL free-running oscillation frequency	f_{PLLFR}	—	8	—	MHz	

Note 1. Time until the clock can be used after the main clock oscillator stop bit (MOSCCR.MOSTP) is set to 0 (operating) when the external clock is stable.

Note 2. Reference values when an 8-MHz resonator is used.
When specifying the main clock oscillator stabilization time, set the MOSCWTCR register with a stabilization time value that is equal to or greater than the resonator-manufacturer-recommended value.
After changing the setting of the MOSCCR.MOSTP bit so that the main clock oscillator operates, read the OSCOVFSR.MOOVF flag to confirm that is has become 1, and then start using the main clock.

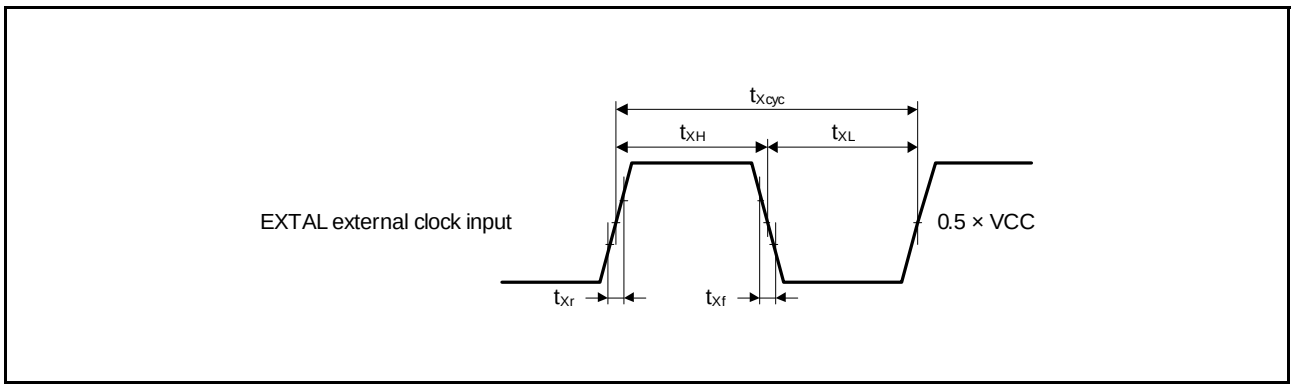


Figure 32.4 EXTAL External Clock Input Timing

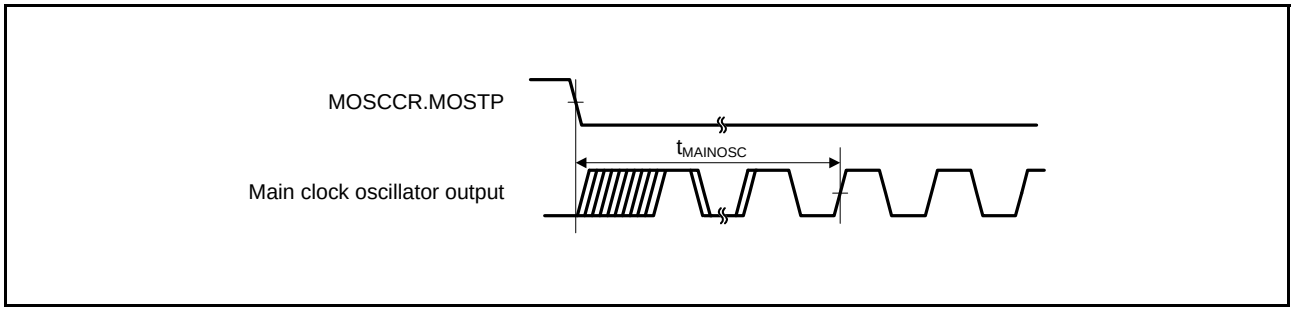


Figure 32.5 Main Clock Oscillation Start Timing

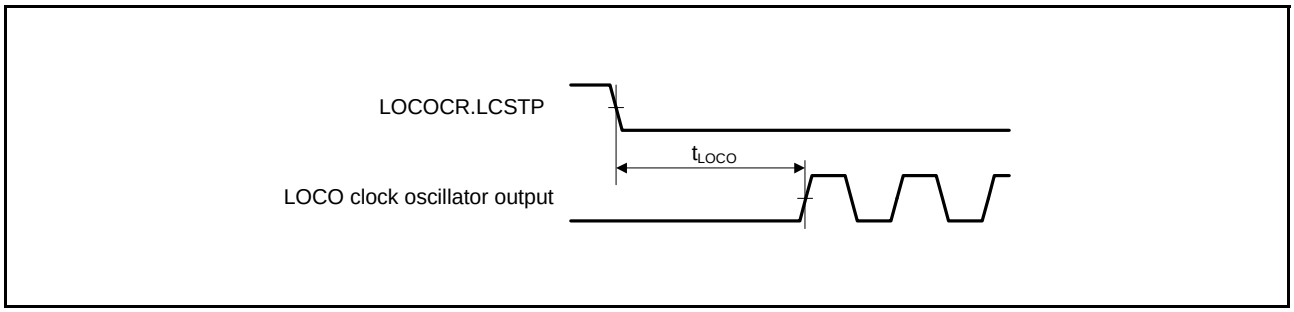


Figure 32.6 LOCO Clock Oscillation Start Timing

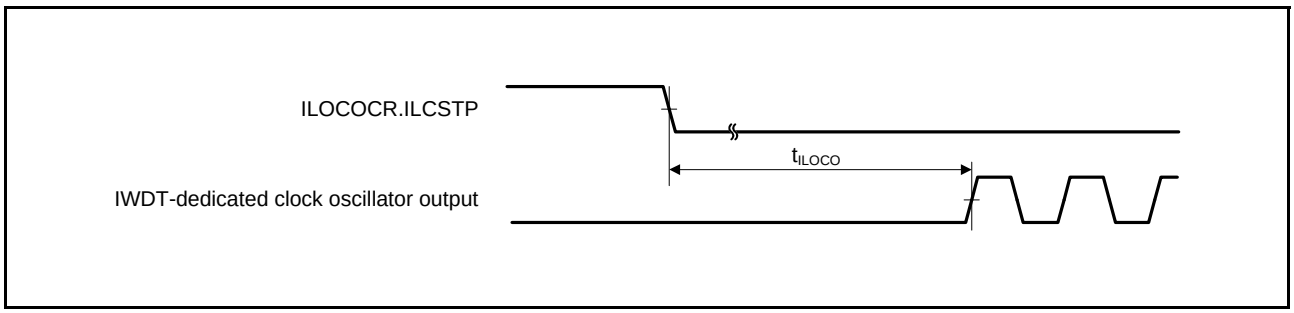


Figure 32.7 IWDT-Dedicated Clock Oscillation Start Timing

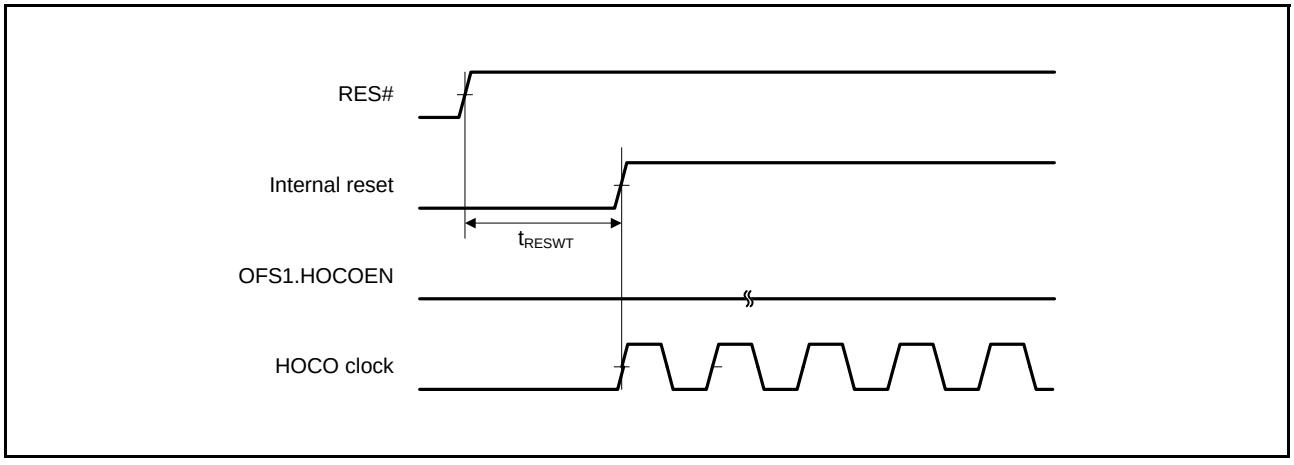


Figure 32.8 HOCO Clock Oscillation Start Timing (After Reset is Canceled by Setting OFS1.HOCOEN Bit to 0)

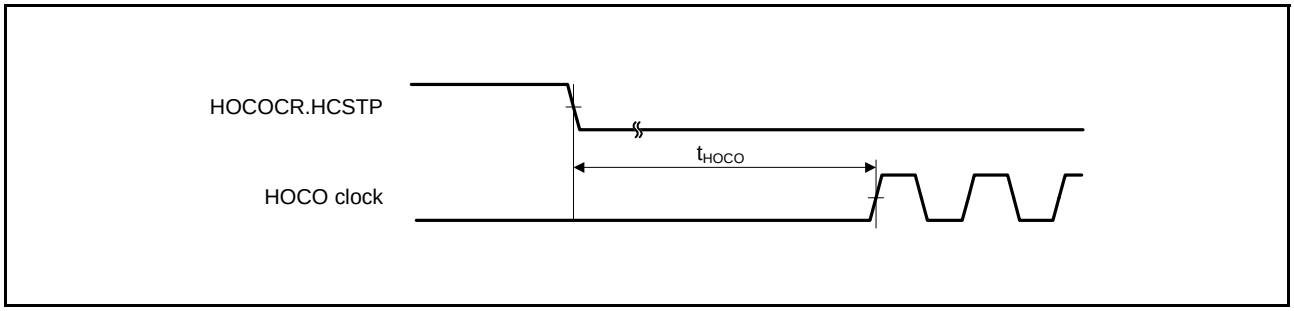


Figure 32.9 HOCO Clock Oscillation Start Timing (Oscillation is Started by Setting HOCOCR.HCSTP Bit)

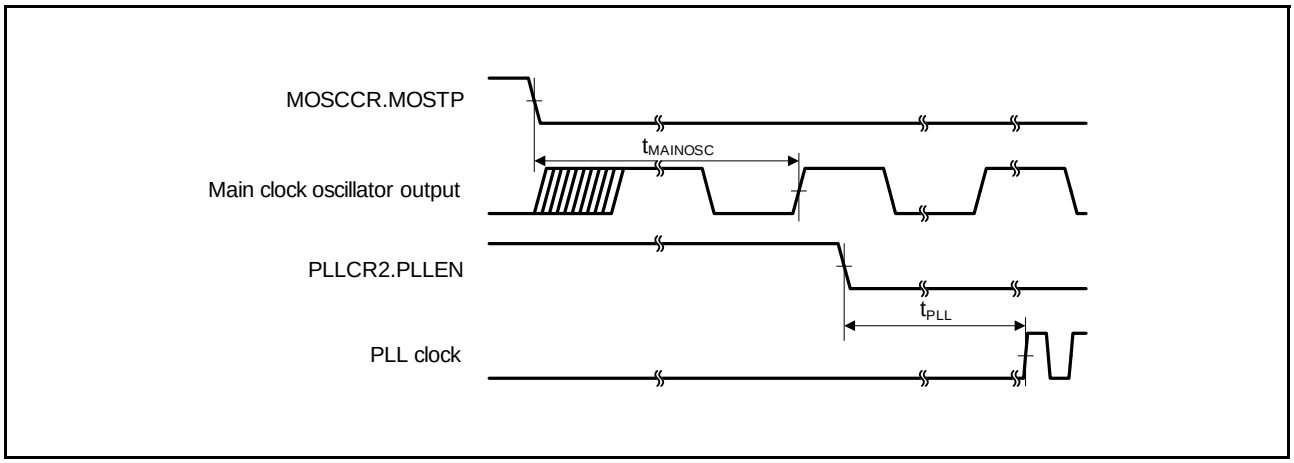


Figure 32.10 PLL Clock Oscillation Start Timing (PLL is Operated after Main Clock Oscillation Has Settled)

32.4.2 Reset Timing

Table 32.22 Reset Timing

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
RES# pulse width	At power-on	t_{RESWP}	3	—	—	ms	Figure 32.11
	Other than above	t_{RESW}	30	—	—	µs	Figure 32.12
Wait time after RES# cancellation (at power-on)		t_{RESWT}	—	27.5	—	ms	Figure 32.11
Wait time after RES# cancellation (during powered-on state)		t_{RESWT}	—	120	—	µs	Figure 32.12
Independent watchdog timer reset period		t_{RESWIW}	—	1	—	IWDT clock cycle	Figure 32.13
Software reset period		t_{RESWSW}	—	1	—	ICLK cycle	
Wait time after independent watchdog timer reset cancellation*1		t_{RESW2}	—	300	—	µs	
Wait time after software reset cancellation		t_{RESW2}	—	170	—	µs	

Note 1. When IWDTCR.CKS[3:0] = 0000b.

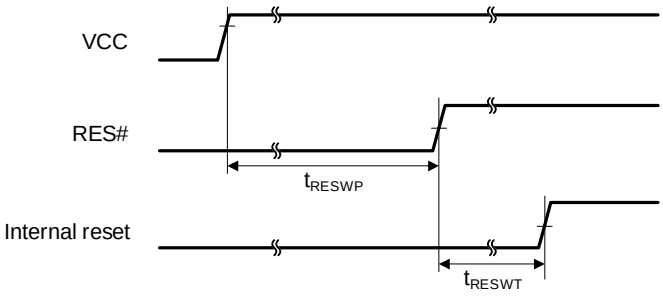


Figure 32.11 Reset Input Timing at Power-On

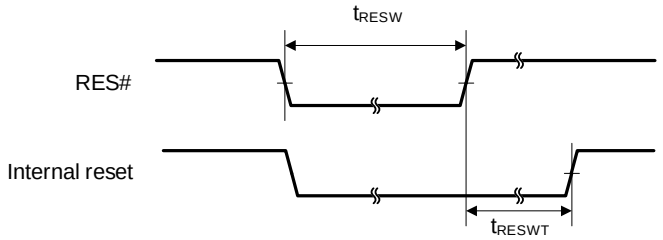


Figure 32.12 Reset Input Timing (1)

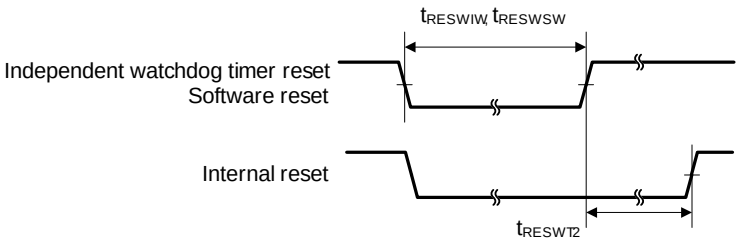


Figure 32.13 Reset Input Timing (2)

32.4.3 Timing of Recovery from Low Power Consumption Modes

Table 32.23 Timing of Recovery from Low Power Consumption Modes (1)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item				Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Recovery time from software standby mode*1	High-speed mode	Crystal connected to main clock oscillator	Main clock oscillator operating*2	t_{SBYMC}	—	2	3	ms	Figure 32.14
			Main clock oscillator and PLL circuit operating*3	t_{SBYPC}	—	2	3		
		External clock input to main clock oscillator	Main clock oscillator operating*4	t_{SBYEX}	—	35	50	µs	
			Main clock oscillator and PLL circuit operating*5	t_{SBYPE}	—	70	95		
		HOCO clock oscillator operating		t_{SBYHO}	—	40	55		
		LOCO clock oscillator operating		t_{SBYLO}	—	40	55		

Note: Values when the frequencies of PCLKB, PCLKD, and FCLK are not divided.
Note 1. The recovery time varies depending on the state of each oscillator when the WAIT instruction is executed. The recovery time when multiple oscillators are operating varies depending on the operating state of the oscillators that are not selected as the system clock source. The above table applies when only the corresponding clock is operating.
Note 2. When the frequency of crystal is 20 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 04h.
Note 3. When the frequency of PLL is 32 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 04h.
Note 4. When the frequency of the external clock is 20 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 00h.
Note 5. When the frequency of PLL is 32 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 00h.

Table 32.24 Timing of Recovery from Low Power Consumption Modes (2)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item				Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Recovery time from software standby mode*1	Middle-speed mode	Crystal connected to main clock oscillator	Main clock oscillator operating*2	t_{SBYMC}	—	2	3	ms	Figure 32.14
			Main clock oscillator and PLL circuit operating*3	t_{SBYPC}	—	2	3		
		External clock input to main clock oscillator	Main clock oscillator operating*4	t_{SBYEX}	—	3	4	µs	
			Main clock oscillator and PLL circuit operating*5	t_{SBYPE}	—	65	85		
		HOCO clock oscillator operating		t_{SBYHO}	—	40	50		
		LOCO clock oscillator operating		t_{SBYLO}	—	5	7		

Note: Values when the frequencies of PCLKB, PCLKD, and FCLK are not divided.
Note 1. The recovery time varies depending on the state of each oscillator when the WAIT instruction is executed. The recovery time when multiple oscillators are operating varies depending on the operating state of the oscillators that are not selected as the system clock source. The above table applies when only the corresponding clock is operating.
Note 2. When the frequency of the crystal is 12 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 04h.
Note 3. When the frequency of the PLL is 24 MHz and that of the ILCK is 12 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 04h.
Note 4. When the frequency of the external clock is 12 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 00h.
Note 5. When the frequency of the PLL is 24 MHz and that of the ILCK is 12 MHz.
When the main clock oscillator wait control register (MOSCWTCR) is set to 00h.

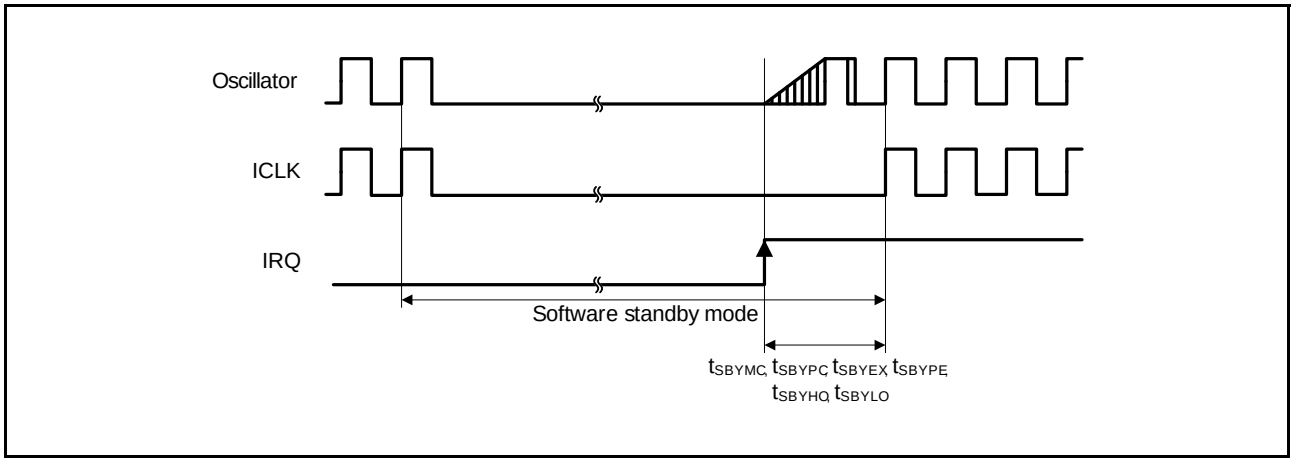


Figure 32.14 Software Standby Mode Recovery Timing

Table 32.25 Timing of Recovery from Low Power Consumption Modes (3)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Recovery time from deep sleep mode*1	High-speed mode*2	t_{DSLP}	—	2	3.5	µs	Figure 32.15
	Middle-speed mode*3	t_{DSLP}	—	3	4		

Note: Values when the frequencies of PCLKB, PCLKD, and FCLK are not divided.
Note 1. Oscillators continue oscillating in deep sleep mode.
Note 2. When the frequency of the system clock is 32 MHz.
Note 3. When the frequency of the system clock is 12 MHz.

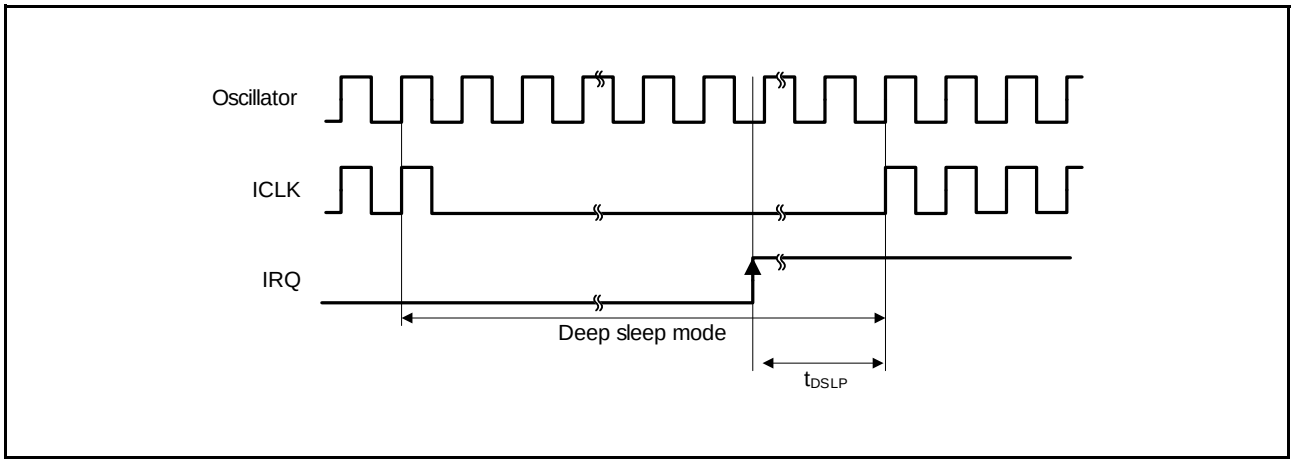


Figure 32.15 Deep Sleep Mode Recovery Timing

Table 32.26 Operating Mode Transition Time

Conditions: VCC = 2.7 V to AVCC0, AVCC0 = 2.7 V to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Mode before Transition	Mode after Transition	ICLK Frequency	Transition Time			Unit
			Min.	Typ.	Max.	
High-speed operating mode	Middle-speed operating modes	8 MHz	—	10	—	µs
Middle-speed operating modes	High-speed operating mode	8 MHz	—	37.5	—	µs

Note: Values when the frequencies of PCLKB, PCLKD, and FCLK are not divided.

32.4.4 Control Signal Timing

Table 32.27 Control Signal Timing

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
NMI pulse width	t_{NMIW}	200	—	—	ns	NMI digital filter disabled (NMIFLTE.NFLTEN = 0)	t_{Pcyc} × 2 ≤ 200 ns
		t_{Pcyc} × 2*1	—	—			t_{Pcyc} × 2 > 200 ns
		200	—	—		NMI digital filter enabled (NMIFLTE.NFLTEN = 1)	t_{NMICK} × 3 ≤ 200 ns
		t_{NMICK} × 3.5*2	—	—			t_{NMICK} ×3 > 200 ns
IRQ pulse width	t_{IRQW}	200	—	—	ns	IRQ digital filter disabled (IRQFLTE0.FLTENi = 0)	t_{Pcyc} × 2 ≤ 200 ns
		t_{Pcyc} × 2*1	—	—			t_{Pcyc} × 2 > 200 ns
		200	—	—		IRQ digital filter enabled (IRQFLTE0.FLTENi = 1)	t_{IRQCK} × 3 ≤ 200 ns
		t_{IRQCK} ×3.5 *3	—	—			t_{IRQCK} × 3 > 200 ns

Note: 200 ns minimum in software standby mode.
Note 1. t_{Pcyc} indicates the cycle of PCLKB.
Note 2. t_{NMICK} indicates the cycle of the NMI digital filter sampling clock.
Note 3. t_{IRQCK} indicates the cycle of the IRQi digital filter sampling clock (i = 0 to 5).

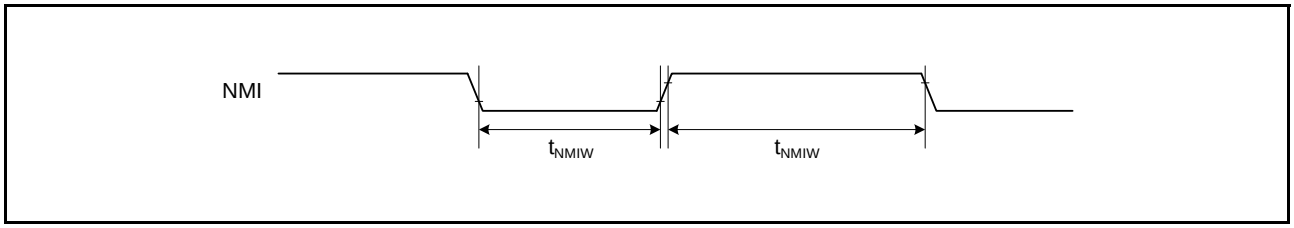


Figure 32.16 NMI Interrupt Input Timing

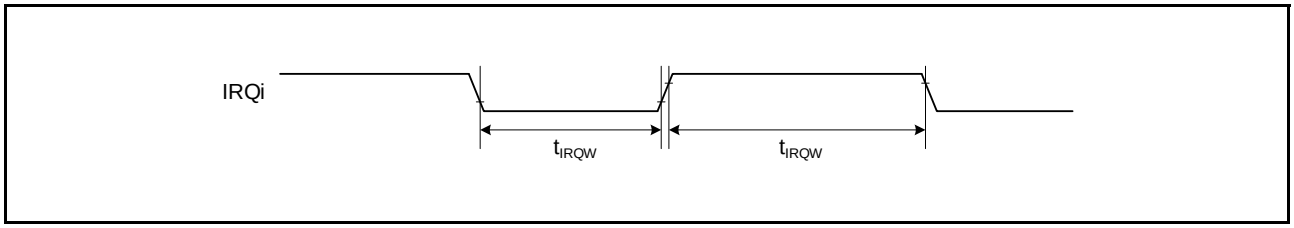


Figure 32.17 IRQ Interrupt Input Timing

32.4.5 Timing of On-Chip Peripheral Modules

32.4.5.1 I/O ports

Table 32.28 Timing of I/O ports

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Max.	Unit*1	Test Conditions
I/O ports	Input data pulse width	t_{PRW}	1.5	—	t_{Pcyc}	Figure 32.18

Note 1. t_{Pcyc}: PCLK cycle

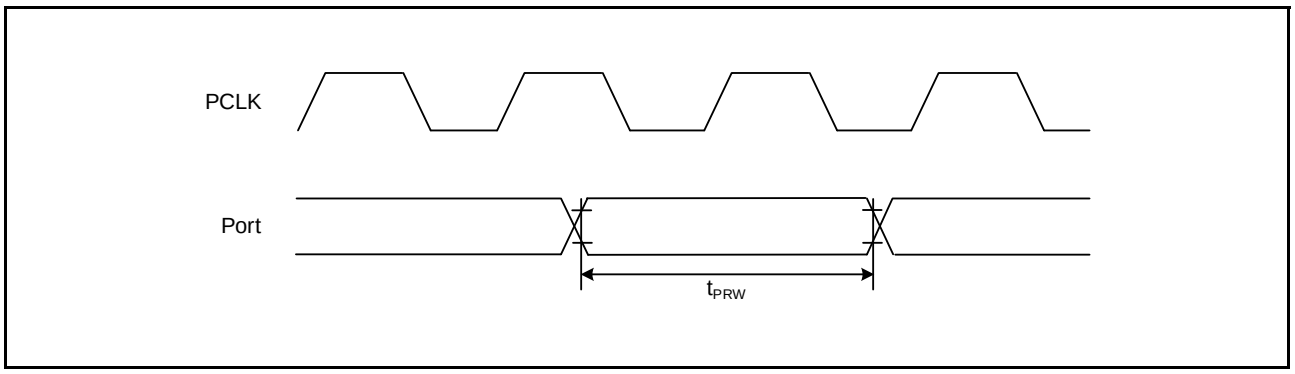


Figure 32.18 I/O Port Input Timing

32.4.5.2 MTU

Table 32.29 Timing of MTU

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
MTU	Input capture input pulse width	Single-edge setting	t_{TICW}	1.5	—	t_{Pcyc}	Figure 32.19
		Both-edge setting		2.5	—		
	Timer clock pulse width	Single-edge setting	t_{TCKWH}, t_{TCKWL}	1.5	—		Figure 32.20
		Both-edge setting		2.5	—		
		Phase counting mode		2.5	—		

Note 1. t_{Pcyc}: PCLK cycle

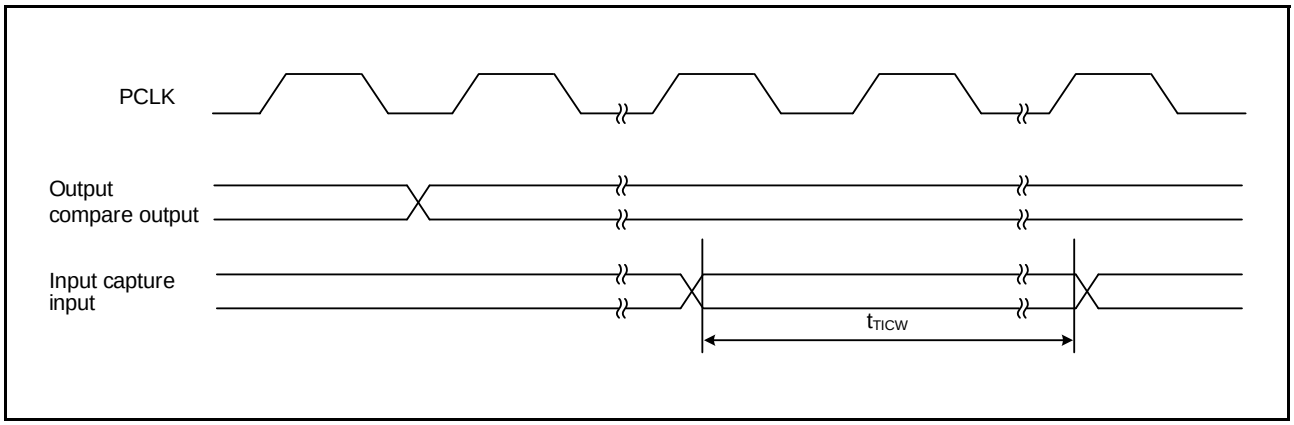


Figure 32.19 MTU Input/Output Timing

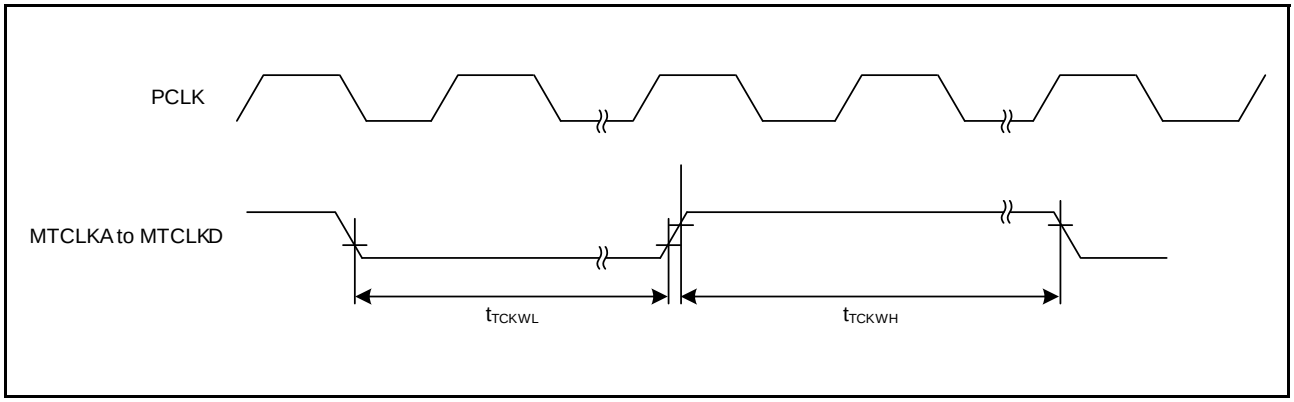


Figure 32.20 MTU Clock Input Timing

32.4.5.3 POE

Table 32.30 Timing of POE

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
POE	POE# input pulse width		t_{POEW}	1.5	—	t_{Pcyc}	Figure 32.21
	Output disable time	Transition of the POE# signal level	t_{POEDI}	—	5 t_{Pcyc} + 0.24	μs	Figure 32.22 In the case of falling-edge detection (ICSRm.POEnM[3:0] = 0000 (m =1, 3, 4; n = 0, 8,10))
		Simultaneous conduction of output pins	t_{POEDO}	—	3 t_{Pcyc} + 0.2		Figure 32.23
		Detection of comparator outputs	t_{POEDC}	—	5 t_{Pcyc} + 0.2		Figure 32.24 When the noise filter for a comparator C is not in use (CMPCTL.CDFS[1:0] = 00), and the values exclude the time until the level of the detection signal changes after a comparator C detects the required change in voltage.
		Register setting	t_{POEDS}	—	1 t_{Pcyc} + 0.2		Figure 32.25 Time for access to the register is not included.
		Oscillation stop detection	t_{POEDOS}	—	21		Figure 32.26

Note 1. t_{Pcyc}: PCLK cycle

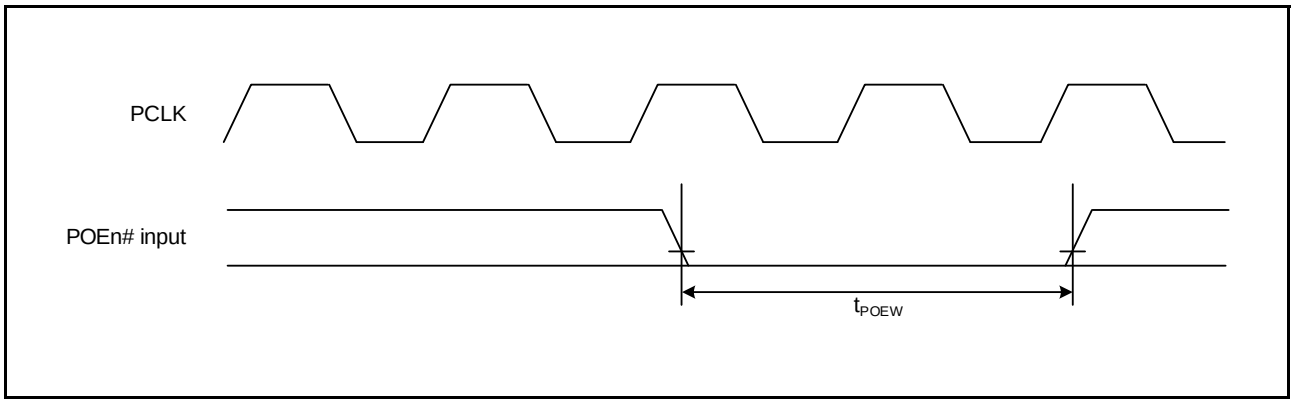


Figure 32.21 POE# Input Timing (n = 0, 8, 10)

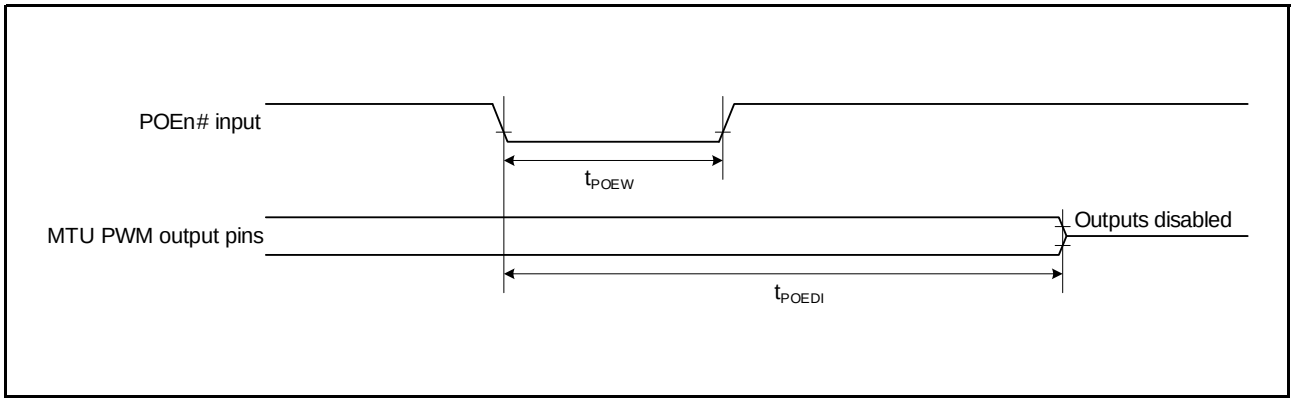
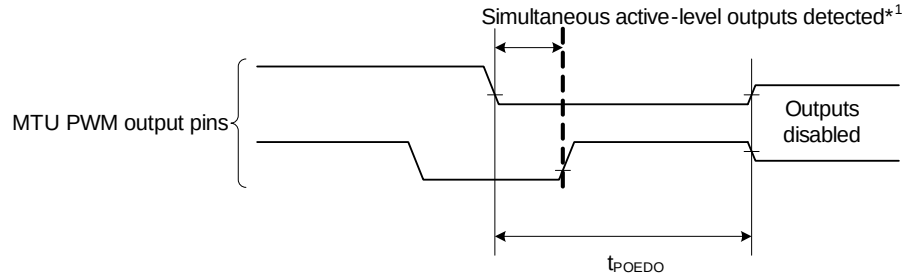


Figure 32.22 Output Disable Time for POE in Response to Transition of the POEn# Signal Level (n = 0, 8, 10)



Note 1. When the active level is set to low .

Figure 32.23 Output Disable Time for POE in Response to the Simultaneous Conduction of Output Pins

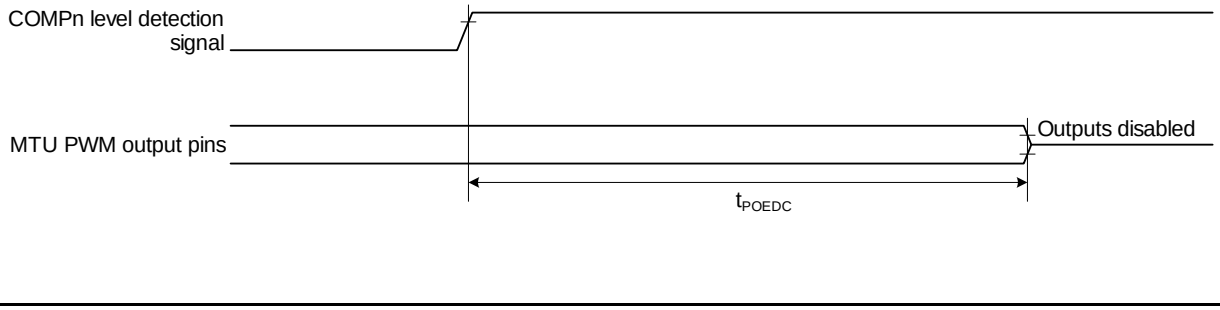


Figure 32.24 Output Disable Time for POE in Response to Detection of the Comparator Outputs (n = 0 to 2)

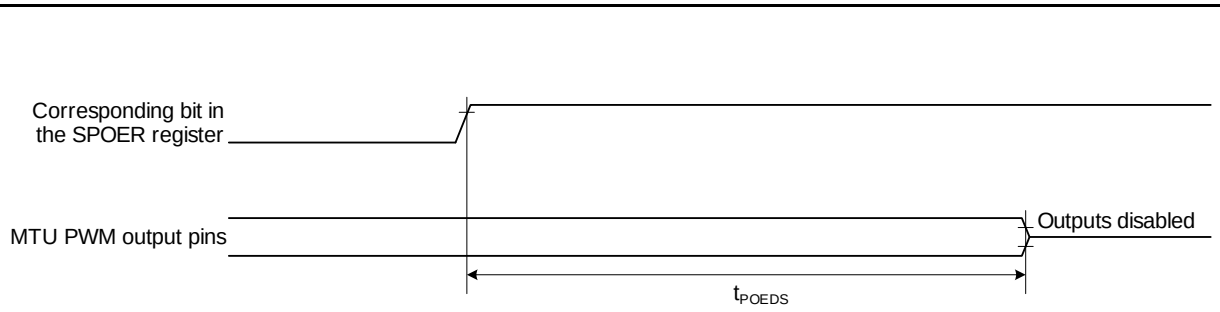


Figure 32.25 Output Disable Time for POE in Response to the Register Setting

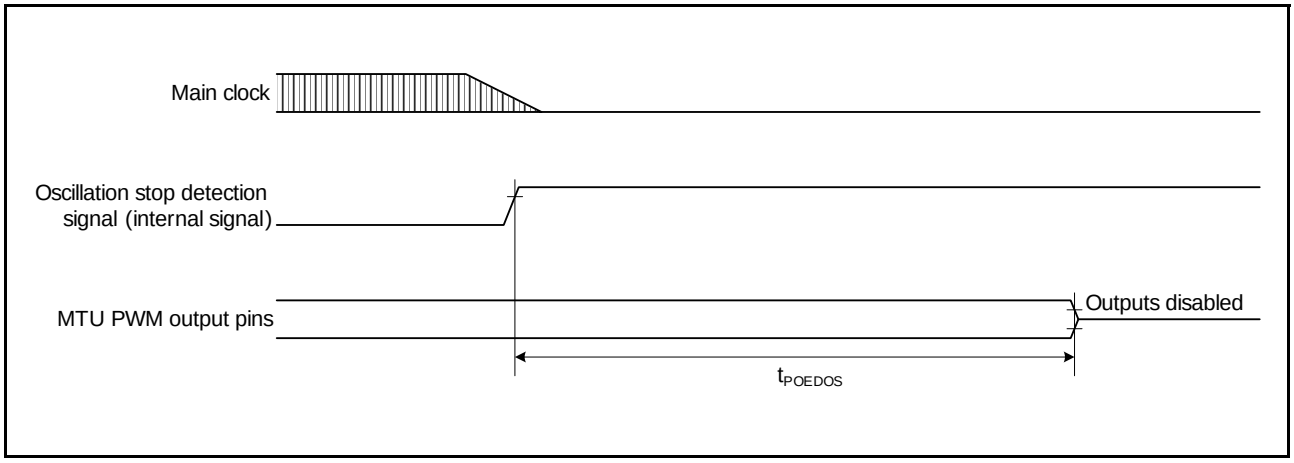


Figure 32.26 Output Disable Time for POE in Response to the Oscillation Stop Detection

32.4.5.4 SCI

Table 32.31 Timing of SCI

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item				Symbol	Min.	Max.	Unit*1	Test Conditions
SCI	Input clock cycle		Asynchronous	t_{Scyc}	4	—	t_{Pcyc}	Figure 32.27
			Clock synchronous		6	—		
	Input clock pulse width			t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time			t_{SCKr}	—	20	ns	
	Input clock fall time			t_{SCKf}	—	20		
	Output clock cycle		Asynchronous	t_{Scyc}	16	—	t_{Pcyc}	Figure 32.28 C = 30 pF
			Clock synchronous		4	—		
	Output clock pulse width			t_{SCKW}	0.4	0.6	t_{Scyc}	
	Output clock rise time			t_{SCKr}	—	20	ns	
	Output clock fall time			t_{SCKf}	—	20		
	Transmit data delay time (master)	Clock synchronous		t_{TXD}	—	40		
	Transmit data delay time (slave)	Clock synchronous	VCC = 4.0 V or above		—	40		
			VCC = 2.7 V or above		—	65		
	Receive data setup time (master)	Clock synchronous	VCC = 4.0 V or above	t_{RXS}	40	—		
			VCC = 2.7 V or above		65	—		
	Receive data setup time (slave)	Clock synchronous			40	—		
	Receive data hold time	Clock synchronous		t_{RXH}	40	—		

Note 1. t_{Pcyc}: PCLK cycle

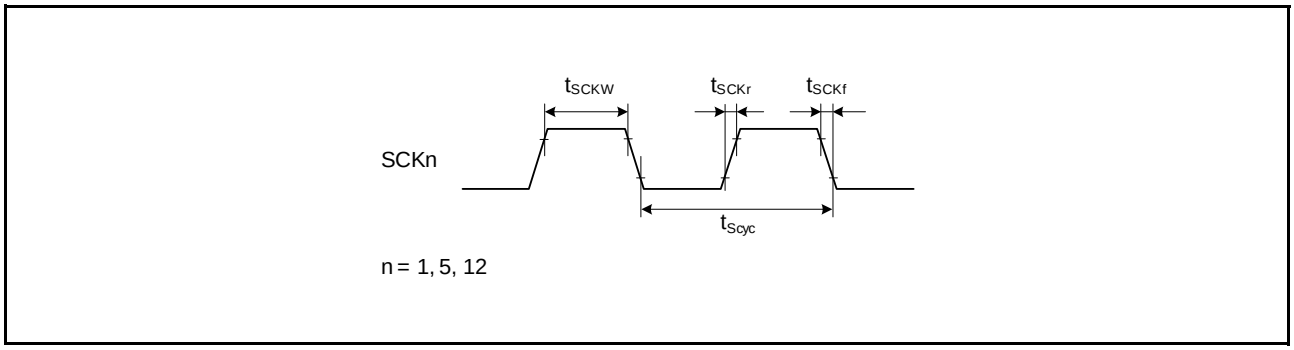


Figure 32.27 SCK Clock Input Timing

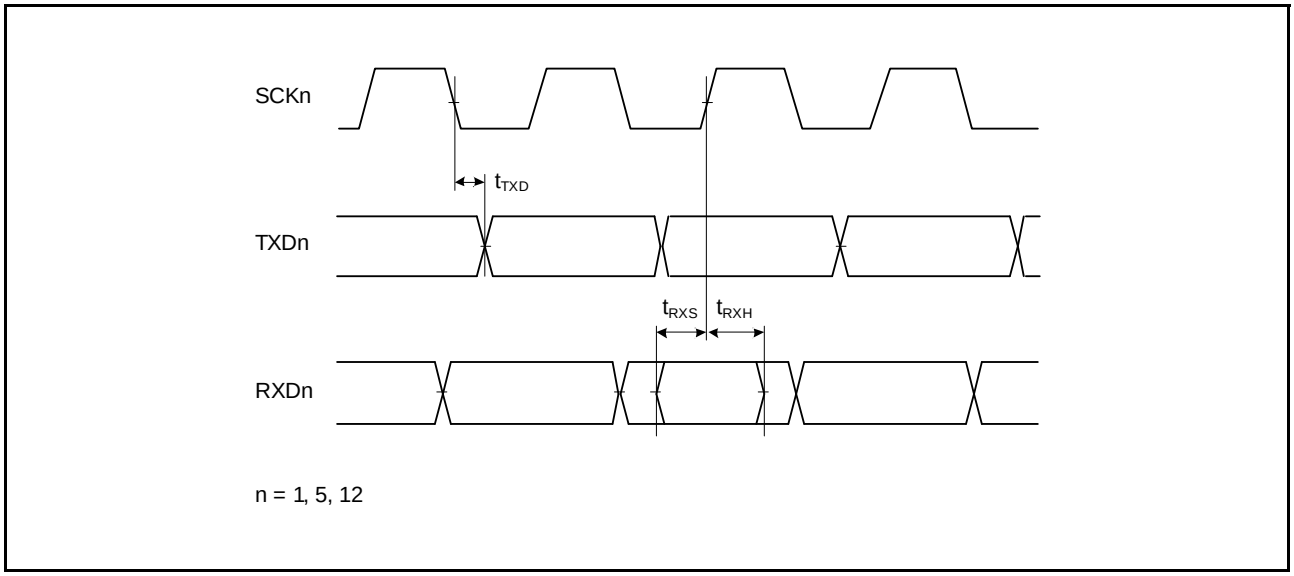


Figure 32.28 SCI Input/Output Timing: Clock Synchronous Mode

Table 32.32 Timing of Simple I²C

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Max.	Unit	Test Conditions
Simple I²C (Standard mode)	SSCL, SSDA input rise time	t_{Sr}	—	1000	ns	Figure 32.29
	SSCL, SSDA input fall time	t_{Sf}	—	300		
	SSCL, SSDA input spike pulse removal time	t_{SP}	0	4 × t_{Pcyc}*1		
	Data input setup time	t_{SDAS}	250	—		
	Data input hold time	t_{SDAH}	0	—		
	SSCL, SSDA capacitive load	C_b*2	—	400	pF	
Simple I²C (Fast mode)	SSCL, SSDA input rise time	t_{Sr}	—	300	ns	Figure 32.29
	SSCL, SSDA input fall time	t_{Sf}	—	300		
	SSCL, SSDA input spike pulse removal time	t_{SP}	0	4 × t_{Pcyc}*1		
	Data input setup time	t_{SDAS}	100	—		
	Data input hold time	t_{SDAH}	0	—		
	SSCL, SSDA capacitive load	C_b*2	—	400	pF	

Note 1. t_{Pcyc}: PCLK cycle
Note 2. C_b is the total capacitance of the bus lines

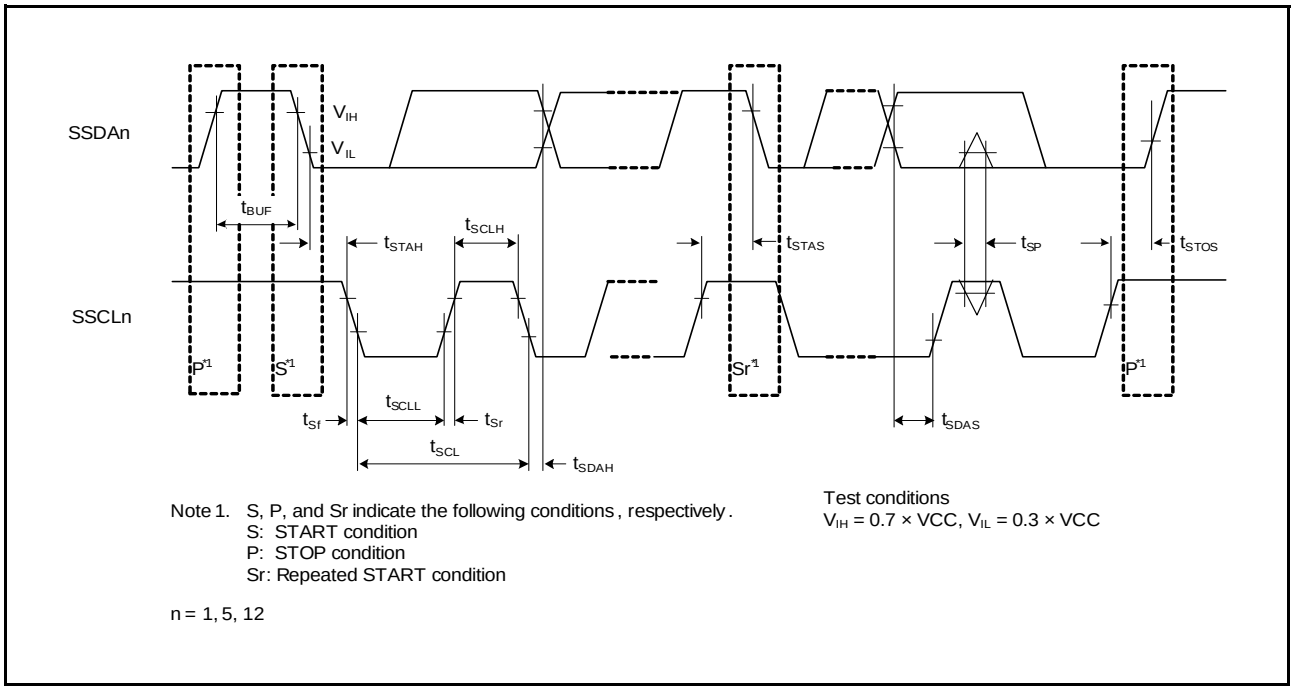


Figure 32.29 Simple I²C Bus Interface Input/Output Timing

Table 32.33 Timing of Simple SPI

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C, C = 30 pF

Item			Symbol	Min.	Max.	Unit*1	Test Conditions
Simple SPI	SCK clock cycle output (master)		t_{SPcyc}	4	65536	t_{Pcyc}	Figure 32.30
	SCK clock cycle input (slave)			6	—		
	SCK clock high pulse width		t_{SPCKWH}	0.4	0.6	t_{SPcyc}	
	SCK clock low pulse width		t_{SPCKWL}	0.4	0.6		
	SCK clock rise/fall time		t_{SPCKr}, t_{SPCKf}	—	20	ns	
	Data input setup time (master)	VCC = 4.0 V or above	t_{SU}	40	—	ns	Figure 32.31, Figure 32.32
		VCC = 2.7 V or above		65	—		
	Data input setup time (slave)			40	—		
	Data input hold time		t_H	40	—		
	SS input setup time		t_{LEAD}	3	—	t_{SPcyc}	
	SS input hold time		t_{LAG}	3	—		
	Data output delay time (master)		t_{OD}	—	40	ns	
	Data output delay time (slave)	VCC = 4.0 V or above		—	40		
		VCC = 2.7 V or above		—	65		
	Data output hold time	Master	t_{OH}	–10	—		
		Slave		–10	—		
	Data rise/fall time		t_{Dr}, t_{Df}	—	20		
	SS input rise/fall time		t_{SSLr}, t_{SSLf}	—	20		
	Slave access time		t_{SA}	—	6	t_{Pcyc}	Figure 32.33, Figure 32.34
	Slave output release time		t_{REL}	—	6		

Note 1. t_{Pcyc}: PCLK cycle

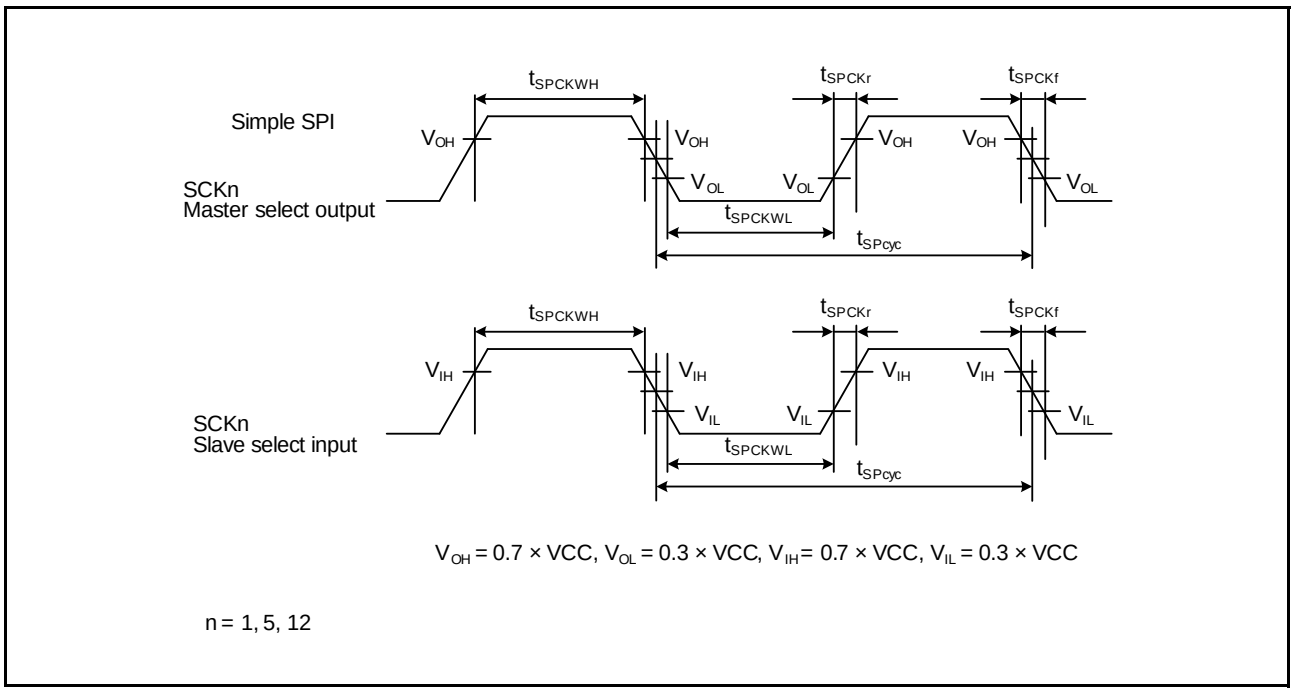


Figure 32.30 Simple SPI Clock Timing

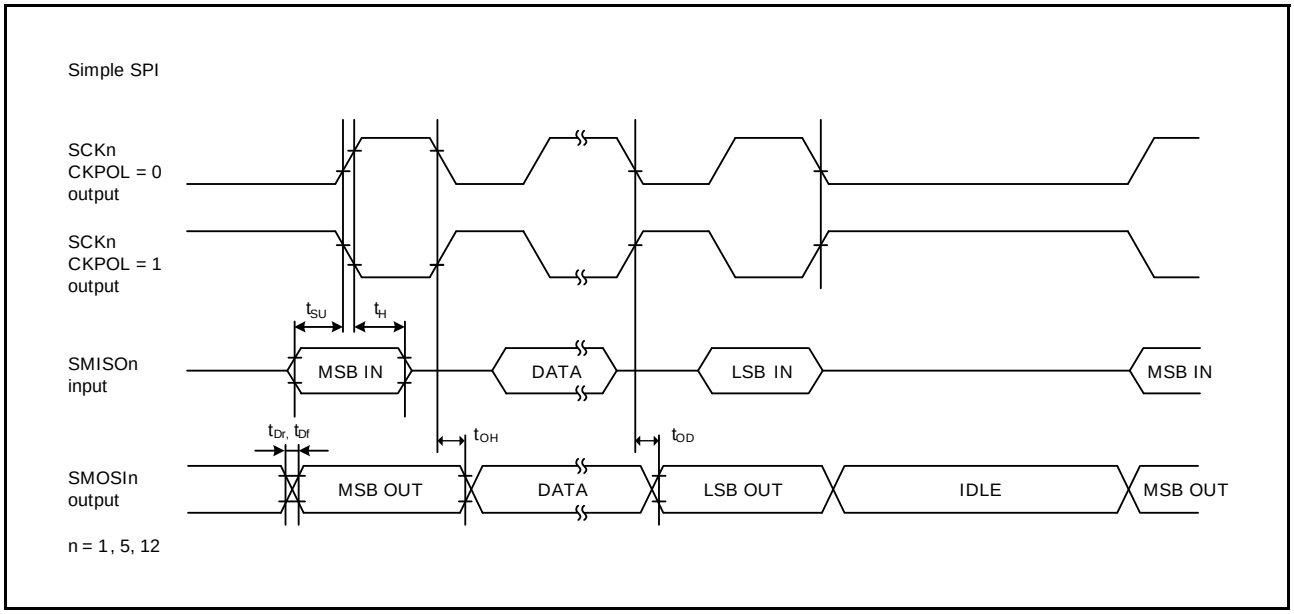


Figure 32.31 Simple SPI Clock Timing (Master, CKPH = 1)

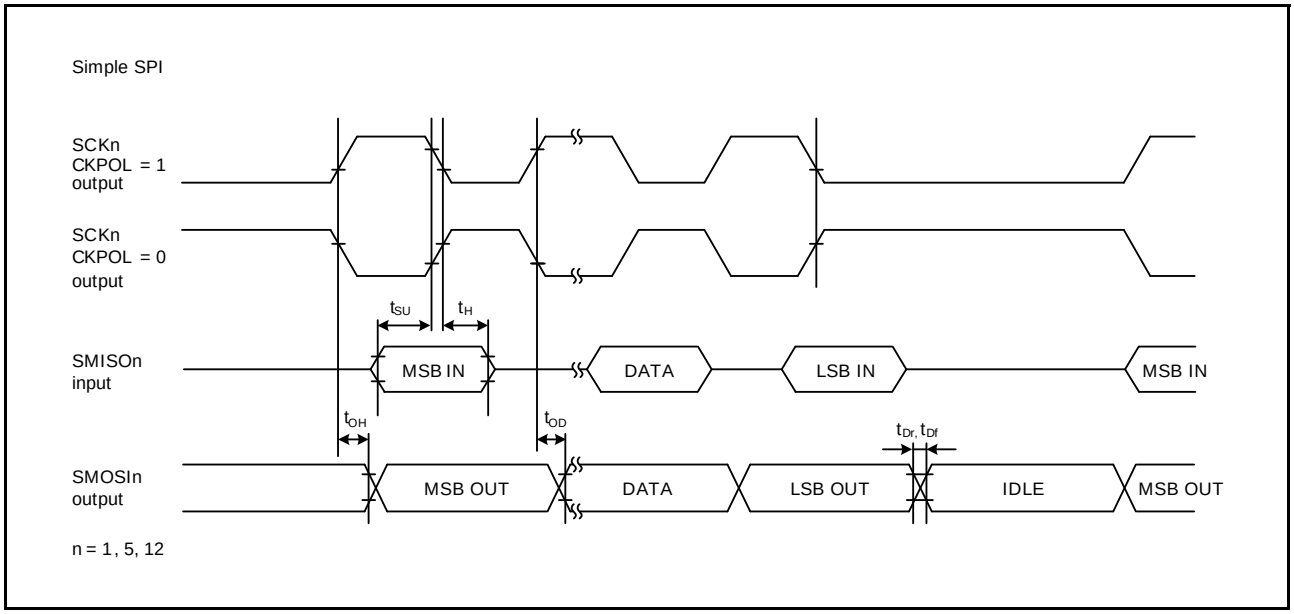


Figure 32.32 Simple SPI Clock Timing (Master, CKPH = 0)

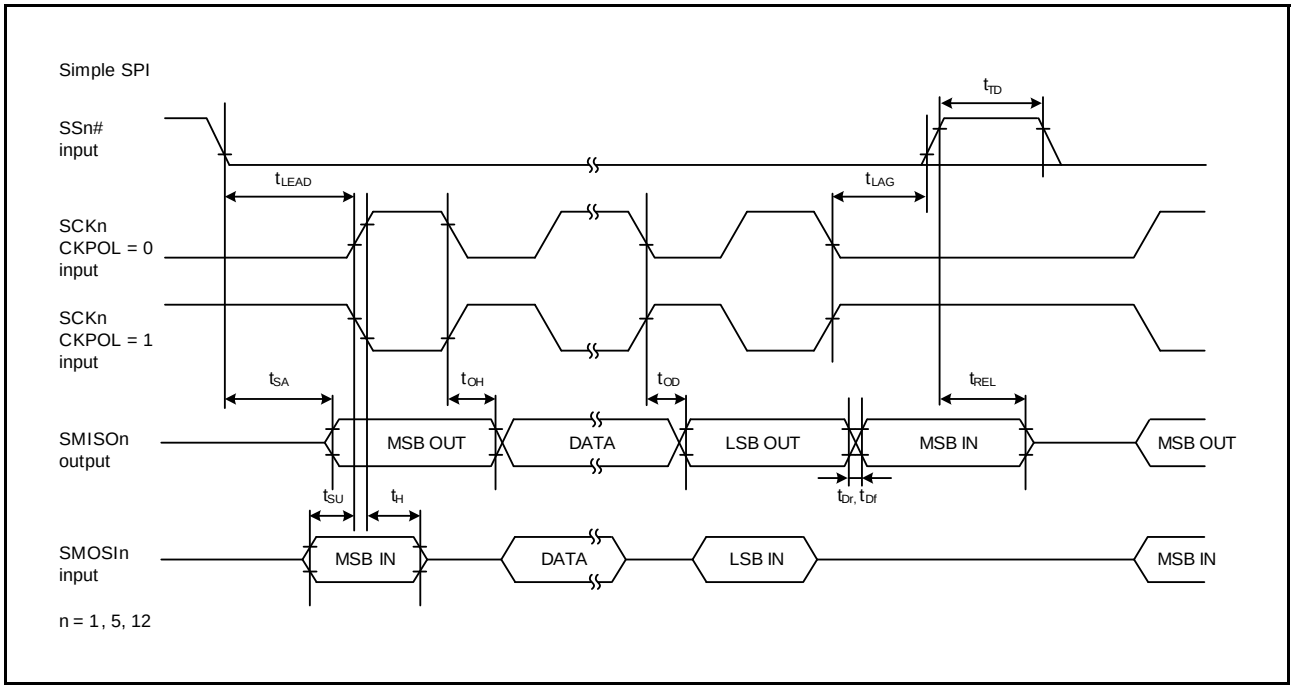


Figure 32.33 Simple SPI Clock Timing (Slave, CKPH = 1)

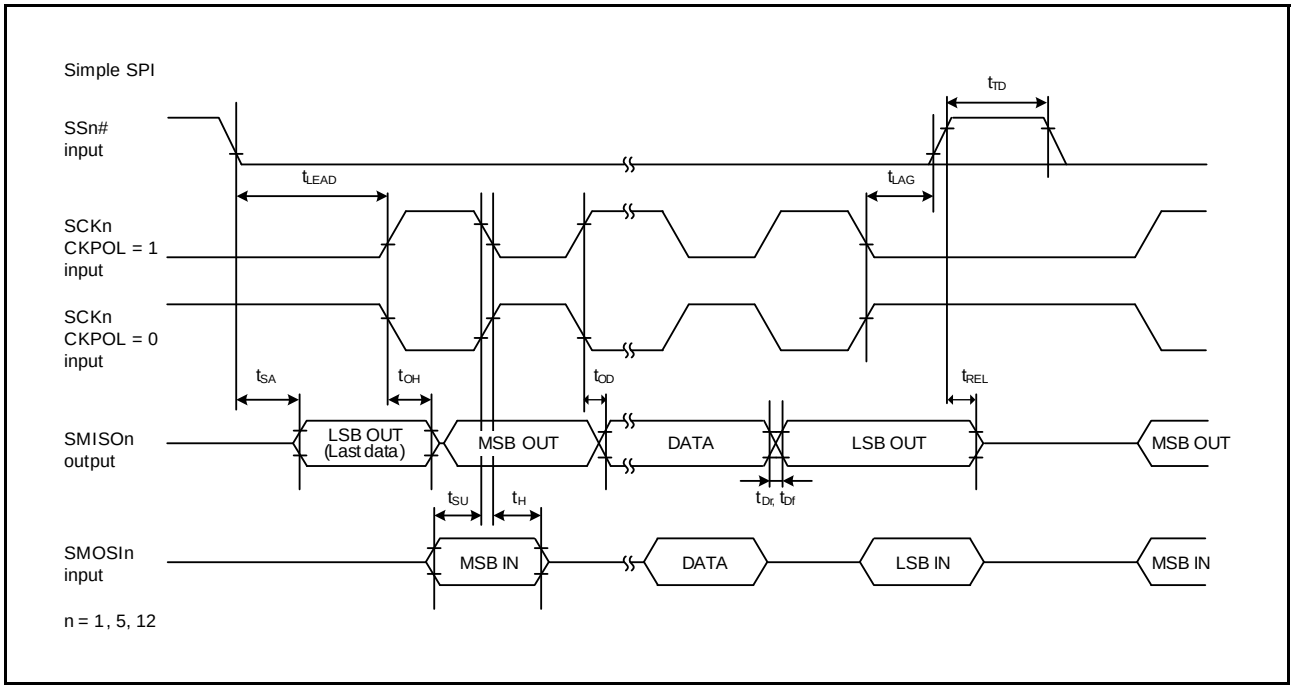


Figure 32.34 Simple SPI Clock Timing (Slave, CKPH = 0)

32.4.5.5 A/D converter

Table 32.34 Timing of A/D converter

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Max.	Unit*1	Test Conditions
A/D converter	Trigger input pulse width	t_{TRGW}	1.5	—	t_{Pcyc}	Figure 32.35

Note 1. t_{Pcyc}: PCLK cycle

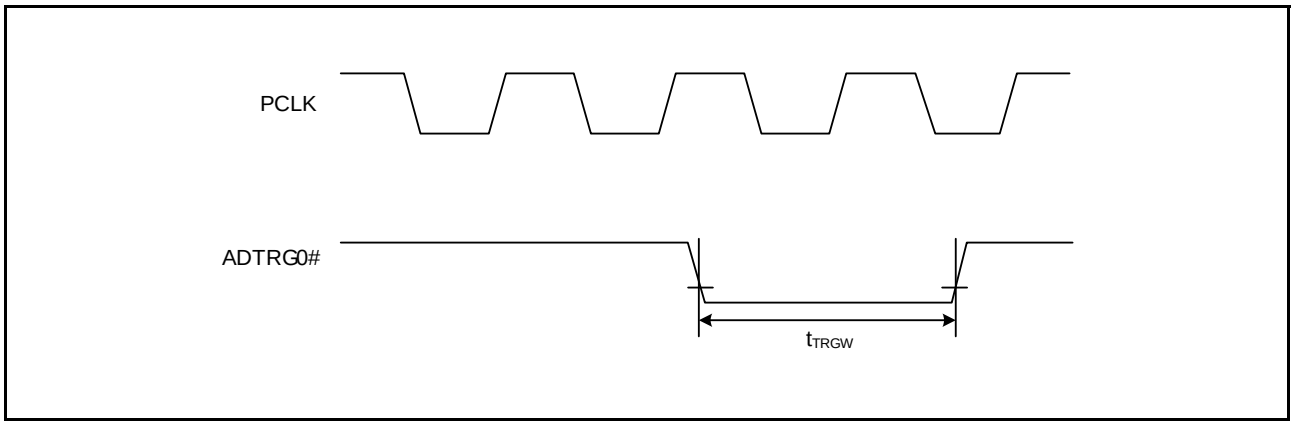


Figure 32.35 A/D Converter External Trigger Input Timing

32.4.5.6 CAC

Table 32.35 Timing of CAC

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item			Symbol	Min.	Max.	Unit	Test Conditions
CAC	CACREF input pulse width	$t_{Pcyc} \le t_{cac}$	t_{CACREF}	$4.5\ t_{cac} + 3\ t_{Pcyc}$	—	ns	
		$t_{Pcyc} > t_{cac}$		$5\ t_{cac} + 6.5\ t_{Pcyc}$			

Note: t_{Pcyc}: PCLK cycle
Note: C_b is the total capacitance of the bus lines.

32.4.5.7 RIIC

Table 32.36 Timing of RIIC

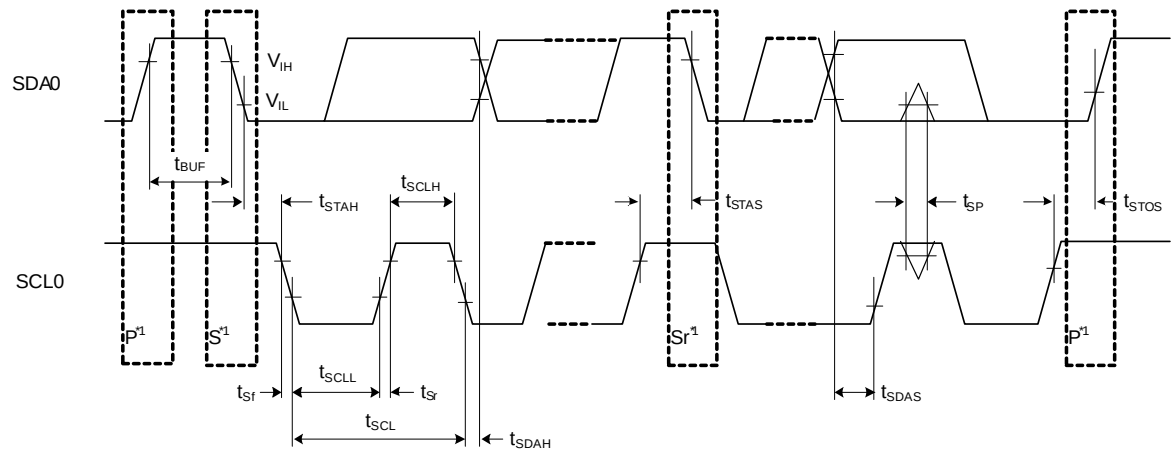
Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.*1, *2	Max.	Unit	Test Conditions
RIIC (Standard mode, SMBus)	SCL cycle time	t_{SCL}	6 (12) × t_{IICcyc} + 1300	—	ns	Figure 32.36
	SCL high pulse width	t_{SCLH}	3 (6) × t_{IICcyc} + 300	—		
	SCL low pulse width	t_{SCLL}	3 (6) × t_{IICcyc} + 300	—		
	SCL, SDA rise time	t_{Sr}	—	1000		
	SCL, SDA fall time	t_{Sf}	—	300		
	SCL, SDA spike pulse removal time	t_{SP}	0	1 (4) × t_{IICcyc}		
	SDA bus free time	t_{BUF}	3 (6) × t_{IICcyc} + 300	—		
	START condition hold time	t_{STAH}	t_{IICcyc} + 300	—		
	Repeated START condition setup time	t_{STAS}	1000	—		
	STOP condition setup time	t_{STOS}	1000	—		
	Data setup time	t_{SDAS}	t_{IICcyc} + 50	—		
	Data hold time	t_{SDAH}	0	—		
	SCL, SDA capacitive load	C_b*3	—	400	pF	
RIIC (Fast mode)	SCL cycle time	t_{SCL}	6 (12) × t_{IICcyc} + 600	—	ns	Figure 32.36
	SCL high pulse width	t_{SCLH}	3 (6) × t_{IICcyc} + 300	—		
	SCL low pulse width	t_{SCLL}	3 (6) × t_{IICcyc} + 300	—		
	SCL, SDA rise time	t_{Sr}	—	300		
	SCL, SDA fall time	t_{Sf}	—	300		
	SCL, SDA spike pulse removal time	t_{SP}	0	1 (4) × t_{IICcyc}		
	SDA bus free time	t_{BUF}	3 (6) × t_{IICcyc} + 300	—		
	START condition hold time	t_{STAH}	t_{IICcyc} + 300	—		
	Repeated START condition setup time	t_{STAS}	300	—		
	STOP condition setup time	t_{STOS}	300	—		
	Data setup time	t_{SDAS}	t_{IICcyc} + 50	—		
	Data hold time	t_{SDAH}	0	—		
	SCL, SDA capacitive load	C_b*3	—	400	pF	

Note 1. t_{IICcyc}: RIIC internal reference count clock (IICφ) cycle

Note 2. The value in parentheses is used when the ICMR3.NF[1:0] bits are set to 11b while a digital filter is enabled with the ICFER.NFE bit = 1.

Note 3. C_b is the total capacitance of the bus lines.



Note 1. S, P, and Sr indicate the following conditions, respectively.
S: START condition
P: STOP condition
Sr: Repeated START condition

Test conditions
V_{IH} = 0.7 × VCC, V_{IL} = 0.3 × VCC

Figure 32.36 RIIC Bus Interface Input/Output Timing

32.5 A/D Conversion Characteristics

Table 32.37 A/D Conversion Characteristics (1)

Conditions: VCC = 4.5 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C, Source impedance = 1.0 kΩ

Item		Min.	Typ.	Max.	Unit	Test Conditions
Frequency		1	—	32	MHz	
Resolution		—	—	12	Bit	
Conversion time*1 (Operation at PCLKD = 32 MHz)	Sample-and-hold circuit not in use	1.41	—	—	μs	High-precision channel ADSSTRn.SST[7:0] bits = 0Dh
	Sample-and-hold circuit in use	2.16	—	—		High-precision channel ADSSTRn.SST[7:0] bits = 0Dh ADSHCR.SSTSH[7:0] bits = 0Bh AN000 to 002 = 0.25 V to AVCC0 – 0.25 V
Analog input capacitance		—	—	12	pF	
Offset error	Sample-and-hold circuit not in use	—	±0.5	±4.5	LSB	
	Sample-and-hold circuit in use	—	±1.5	±6.5		
Full-scale error	Sample-and-hold circuit not in use	—	±0.75	±4.5	LSB	
	Sample-and-hold circuit in use	—	±1.5	±6.5		
Quantization error		—	±0.5	—	LSB	
Absolute accuracy	Sample-and-hold circuit not in use	—	±1.25	±5.0	LSB	
	Sample-and-hold circuit in use	—	±3.0	±8.0		AN000 to 002 = 0.25V to AVCC0 – 0.25
DNL differential nonlinearity error		—	±0.5	±1.5	LSB	
INL integral nonlinearity error		—	±1.5	±4.0	LSB	

Note: The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Table 32.38 A/D Conversion Characteristics (2)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C, Source impedance = 1.0 kΩ

Item		Min.	Typ.	Max.	Unit	Test Conditions
Frequency		1	—	32	MHz	
Resolution		—	—	12	Bit	
Conversion time*1 (Operation at PCLKD = 32 MHz)	Sample-and-hold circuit not in use	1.41	—	—	μs	High-precision channel ADSSTRn.SST[7:0] bits = 0Dh
	Sample-and-hold circuit in use	2.25	—	—		High-precision channel ADSSTRn.SST[7:0] bits = 0Dh ADSHCR.SSTSH[7:0] bits = 0Eh AN000 to 002 = 0.25 V to AVCC0 – 0.25 V
Analog input capacitance		—	—	12	pF	
Offset error	Sample-and-hold circuit not in use	—	±0.5	±4.5	LSB	
	Sample-and-hold circuit in use	—	±1.5	±6.5		
Full-scale error	Sample-and-hold circuit not in use	—	±0.75	±4.5	LSB	
	Sample-and-hold circuit in use	—	±1.5	±6.5		
Quantization error		—	±0.5	—	LSB	
Absolute accuracy	Sample-and-hold circuit not in use	—	±1.25	±5.0	LSB	
	Sample-and-hold circuit in use	—	±3.0	±8.0		AN000 to 002 = 0.25V to AVCC0 – 0.25
DNL differential nonlinearity error		—	±0.5	±1.5	LSB	
INL integral nonlinearity error		—	±1.5	±4.0	LSB	

Note: The characteristics apply when no pin functions other than A/D converter input are used. Absolute accuracy includes quantization errors. Offset error, full-scale error, DNL differential nonlinearity error, and INL integral nonlinearity error do not include quantization errors.

Note 1. The conversion time is the sum of the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Table 32.39 A/D Converter Channel Classification

Classification	Channel	Conditions	Remarks
High-precision channel	AN000 to AN007	AVCC0 = 2.7 to 5.5V	Pins AN000 to AN007 cannot be used as digital outputs when the A/D converter is in use.
Internal reference voltage input channel	Internal reference voltage	AVCC0 = 2.7 to 5.5V	

Table 32.40 A/D Internal Reference Voltage Characteristics

Conditions: VCC = 2.7 V to 5.5V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item	Min.	Typ.	Max.	Unit	Test Conditions
Internal reference voltage input channel*1	1.36	1.43	1.50	V	

Note 1. The A/D internal reference voltage indicates the voltage when the internal reference voltage is input to the A/D converter.

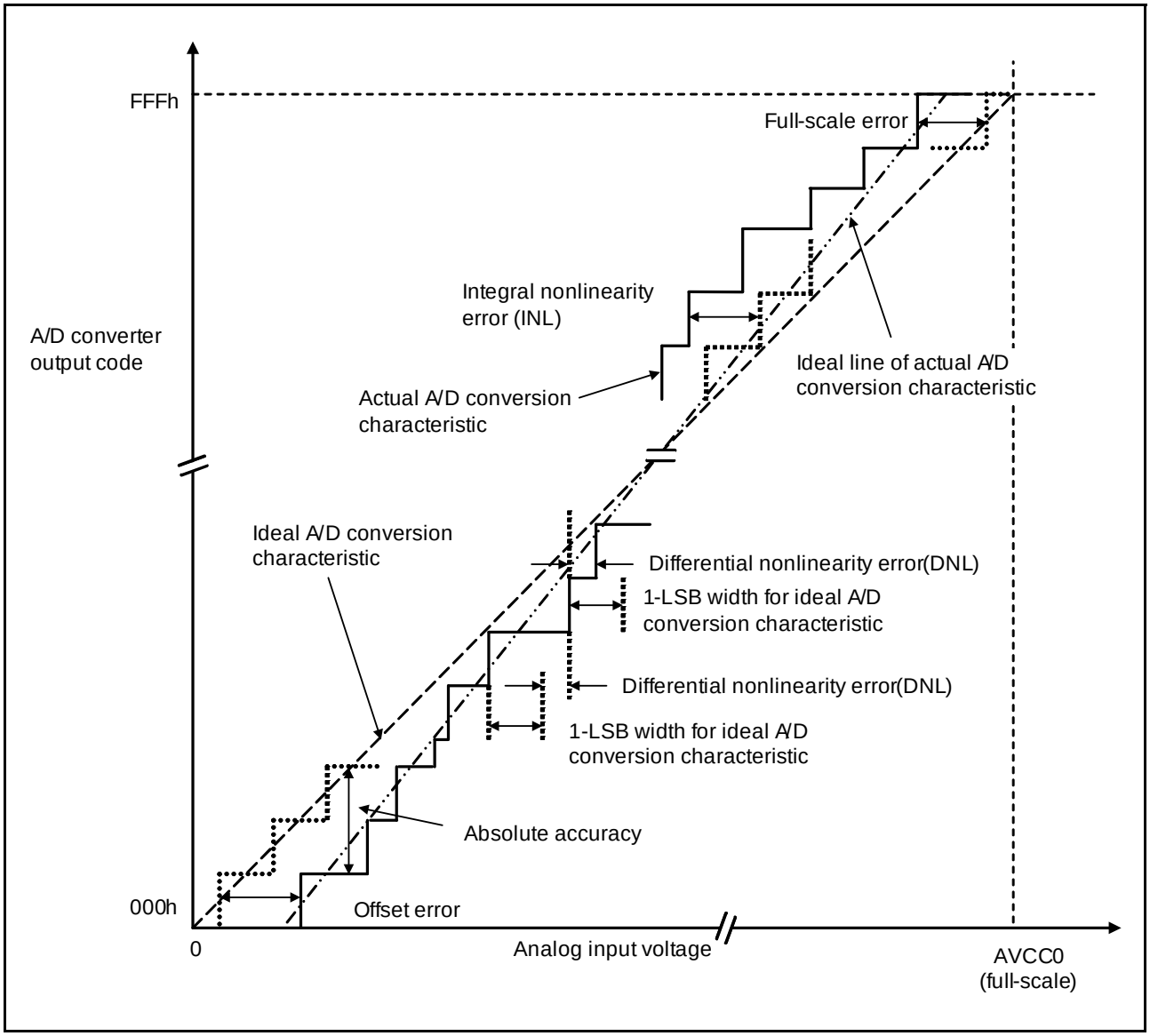


Figure 32.37 Illustration of A/D Converter Characteristic Terms

Absolute accuracy

Absolute accuracy is the difference between output code based on the theoretical A/D conversion characteristics, and the actual A/D conversion result. When measuring absolute accuracy, the voltage at the midpoint of the width of analog input voltage (1-LSB width), that can meet the expectation of outputting an equal code based on the theoretical A/D conversion characteristics, is used as an analog input voltage. For example, if 12-bit resolution is used and if reference voltage (AVCC0) is 3.072 V, then 1-LSB width becomes 0.75 mV, and 0 mV, 0.75 mV, 1.5 mV, ... are used as analog input voltages.

If analog input voltage is 6 mV, absolute accuracy = ±5 LSB means that the actual A/D conversion result is in the range of 003h to 00Dh though an output code, 008h, can be expected from the theoretical A/D conversion characteristics.

Integral nonlinearity error (INL)

Integral nonlinearity error is the maximum deviation between the ideal line when the measured offset and full-scale errors are zeroed, and the actual output code.

Differential nonlinearity error (DNL)

Differential nonlinearity error is the difference between 1-LSB width based on the ideal A/D conversion characteristics and the width of the actual output code.

Offset error

Offset error is the difference between a transition point of the ideal first output code and the actual first output code.

Full-scale error

Full-scale error is the difference between a transition point of the ideal last output code and the actual last output code.

32.6 Programmable Gain Amplifier Characteristics

Table 32.41 Programmable Gain Amplifier Characteristics

Conditions: VCC = 2.7 V to AVCC0, AVCC0 = 4.5 V to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Input offset voltage		V_{poff}	—	—	8	mV	
Input voltage range		V_{pin}	V_{pout} (min)/G	—	V_{pout} (max)/G	V	
Output voltage range	G = 2.000, 2.500, 3.077	V_{pout}	0.1 × AVCC0	—	0.9 × AVCC0	V	
	G = 5.000, 8.000, 10.000		0.15 × AVCC0	—	0.85 × AVCC0		
Gain		G	2.000	—	10.000		
Gain error	G = 2.000, 2.500, 3.077	G_{err}	—	±1.0	±1.5	%	
	G = 5.000, 8.000, 10.000		—	±1.5	±2.5		
Slew rate		SR	10	—	—	V/µs	
Operation stabilization wait time		t_{start}	—	—	5.0	µs	

32.7 Comparator Characteristics

Table 32.42 Comparator Characteristics

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Offset voltage	V_{cioff}	—	—	20	mV	
Reference input voltage range	V_{cref}	0	—	AVCC0	V	
Response time	t_{cr}	—	—	200	ns	VOD = 100 mV CMPCTL.CDFS = 0
	t_{cf}	—	—	200		
Stabilization wait time for input selection	t_{cwait}	300	—	—	ns	
Operation stabilization wait time	t_{cmp}	—	—	1	µs	

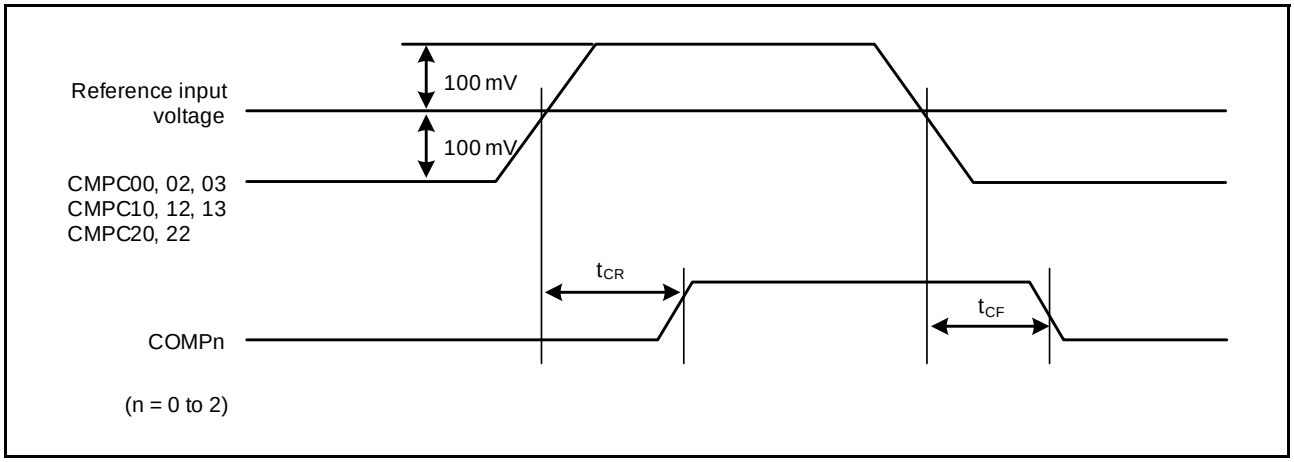


Figure 32.38 Comparator Response Time

32.8 D/A Conversion Characteristics

Table 32.43 D/A Conversion Characteristics

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	—	—	—	8	Bit	
Conversion time	t_{DCONV}	—	—	3.0	µs	
Absolute accuracy	—	—	±1.0	±3.0	LSB	

32.9 Power-On Reset Circuit and Voltage Detection Circuit Characteristics

Table 32.44 Power-On Reset Circuit and Voltage Detection Circuit Characteristics (1)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, T_a = –40 to +105°C

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Voltage detection level	Power-on reset (POR)	V_{POR}	1.35	1.50	1.65	V	Figure 32.39, Figure 32.40
	Voltage detection circuit (LVD0)*1	V_{det0_0}	3.67	3.84	3.97		Figure 32.41 At falling edge VCC
		V_{det0_1}	2.70	2.82	3.00		
		V_{det0_2}	2.37	2.51	2.67		
	Voltage detection circuit (LVD1)*2	V_{det1_0}	4.12	4.29	4.42		Figure 32.42 At falling edge VCC
		V_{det1_1}	3.98	4.14	4.28		
		V_{det1_2}	3.86	4.02	4.16		
		V_{det1_3}	3.68	3.84	3.98		
		V_{det1_4}	2.99	3.10	3.29		
		V_{det1_5}	2.89	3.00	3.19		
		V_{det1_6}	2.79	2.90	3.09		
		V_{det1_7}	2.68	2.79	2.98		
		V_{det1_8}	2.57	2.68	2.87		
	Voltage detection circuit (LVD2)*3	V_{det2_0}	4.08	4.29	4.48		Figure 32.43 At falling edge VCC
		V_{det2_1}	3.95	4.14	4.35		
		V_{det2_2}	3.82	4.02	4.22		
		V_{det2_3}	3.62	3.84	4.02		

Note: These characteristics apply when noise is not superimposed on the power supply. When a setting is made so that the voltage detection level overlaps with that of the voltage detection circuit (LVD2), it cannot be specified which of LVD1 and LVD2 is used for voltage detection.

Note 1. n in the symbol Vdet0_n denotes the value of the OFS1.VDSEL[1:0] bits.

Note 2. n in the symbol Vdet1_n denotes the value of the LVDLVLR.LVD1LVL[3:0] bits.

Note 3. n in the symbol Vdet2_n denotes the value of the LVDLVLR.LVD2LVL[1:0] bits.

Table 32.45 Power-On Reset Circuit and Voltage Detection Circuit Characteristics (2)

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, Ta = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Wait time after power-on reset cancellation	t_{POR}	—	28.4	—	ms	Figure 32.40
Wait time after voltage monitoring 0 reset cancellation	t_{LVD0}	—	568	—	µs	Figure 32.41
Wait time after voltage monitoring 1 reset cancellation	t_{LVD1}	—	100	—	µs	Figure 32.42
Wait time after voltage monitoring 2 reset cancellation	t_{LVD2}	—	100	—	µs	Figure 32.43
Response delay time	t_{det}	—	—	350	µs	Figure 32.39
Minimum VCC down time*1	t_{VOFF}	350	—	—	µs	Figure 32.39, VCC = 1.0 V or above
Power-on reset enable time	$t_{W(POR)}$	1	—	—	ms	Figure 32.40, VCC = below 1.0 V
LVD operation stabilization time (after LVD is enabled)	$Td_{(E-A)}$	—	—	300	µs	Figure 32.42, Figure 32.43
Hysteresis width (power-on reset (POR))	V_{PORH}	—	110	—	mV	
Hysteresis width (LVD0, LVD1 and LVD2)	V_{LVH}	—	70	—	mV	Vdet0_0 to 2 selected Vdet1_0 to 4 selected
		—	60	—		Vdet1_5 to 8, LVD2 selected

Note: These characteristics apply when noise is not superimposed on the power supply. When a setting is made so that the voltage detection level overlaps with that of the voltage detection circuit (LVD1), it cannot be specified which of LVD1 and LVD2 is used for voltage detection.

Note 1. The minimum VCC down time indicates the time when VCC is below the minimum value of voltage detection levels V_{POR}, V_{det0}, V_{det1}, and V_{det2} for the POR/LVD.

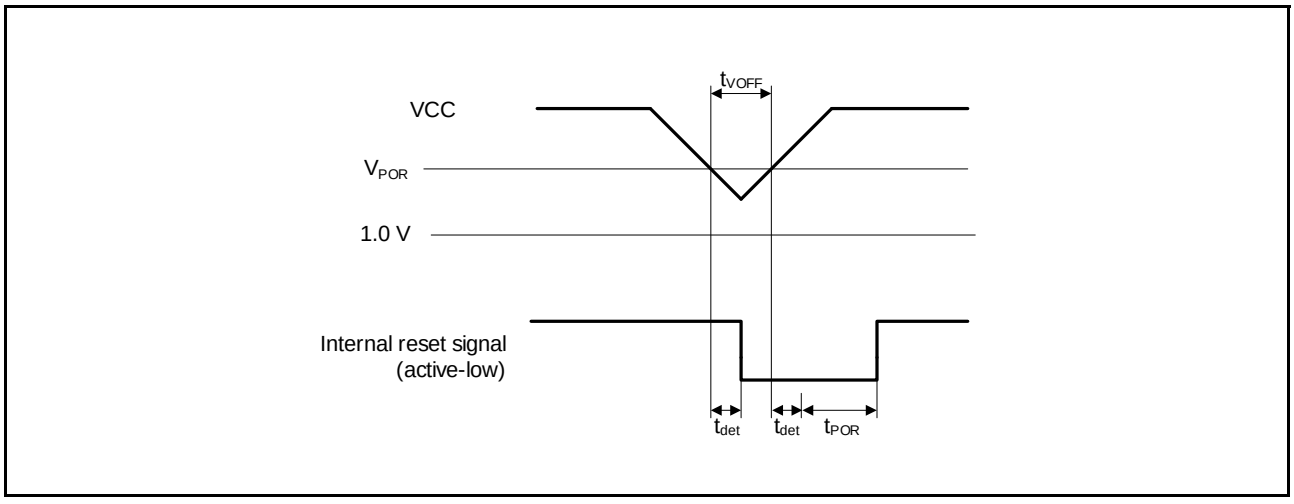
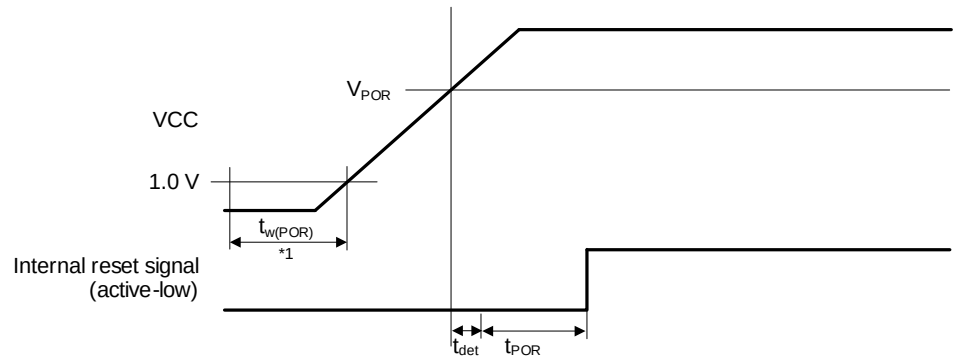


Figure 32.39 Voltage Detection Reset Timing



Note 1. $t_{w(POR)}$ is the time required for a power-on reset to be enabled while the external power VCC is being held below the valid voltage (1.0 V).
When VCC turns on, maintain $t_{w(POR)}$ for 1.0 ms or more.

Figure 32.40 Power-On Reset Timing

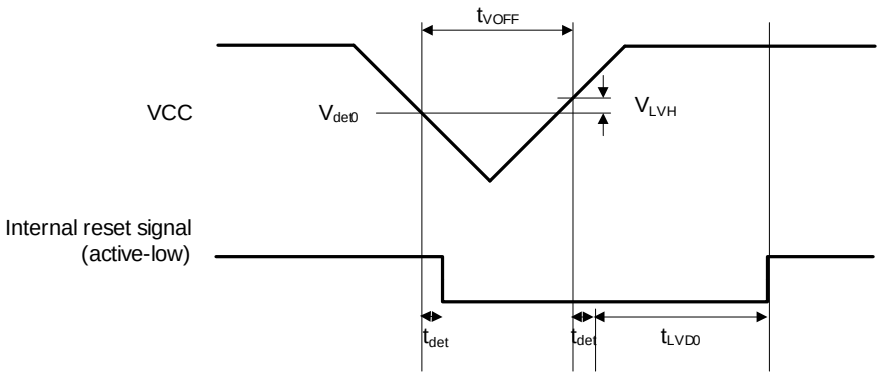


Figure 32.41 Voltage Detection Circuit Timing (V_{det0})

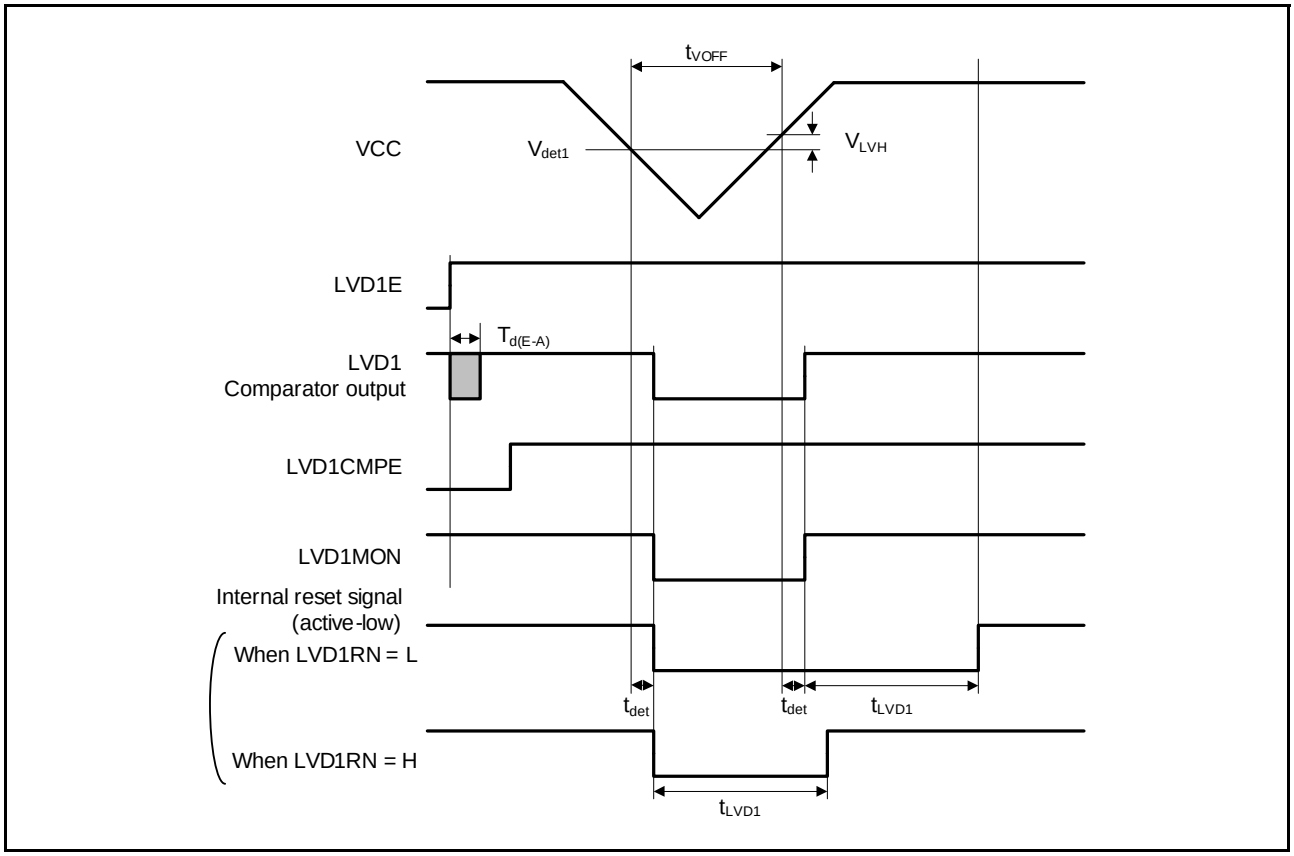


Figure 32.42 Voltage Detection Circuit Timing (V_{det1})

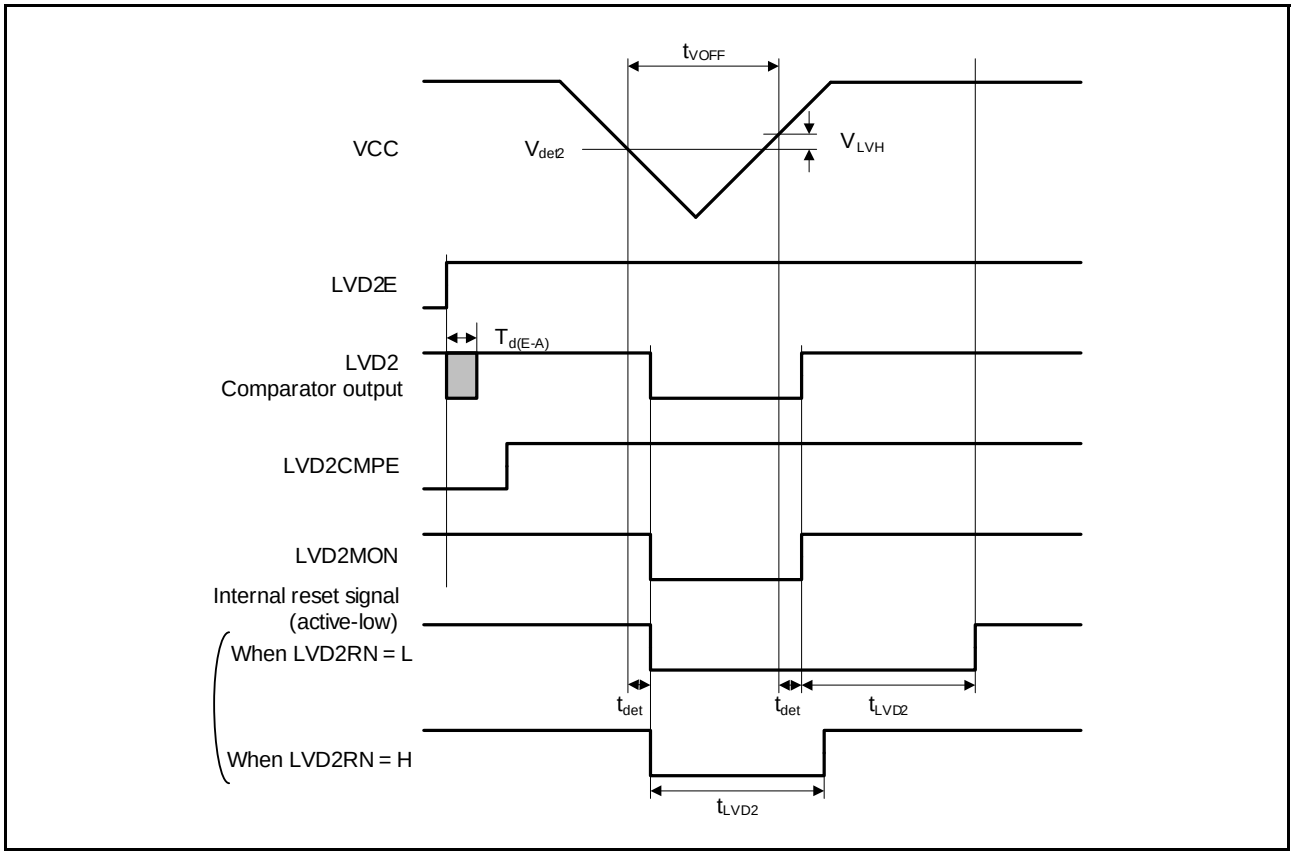


Figure 32.43 Voltage Detection Circuit Timing (V_{det2})

32.10 Oscillation Stop Detection Timing

Table 32.46 Oscillation Stop Detection Timing

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V, Ta = –40 to +105°C

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Detection time	t_{dr}	—	—	1	ms	Figure 32.44

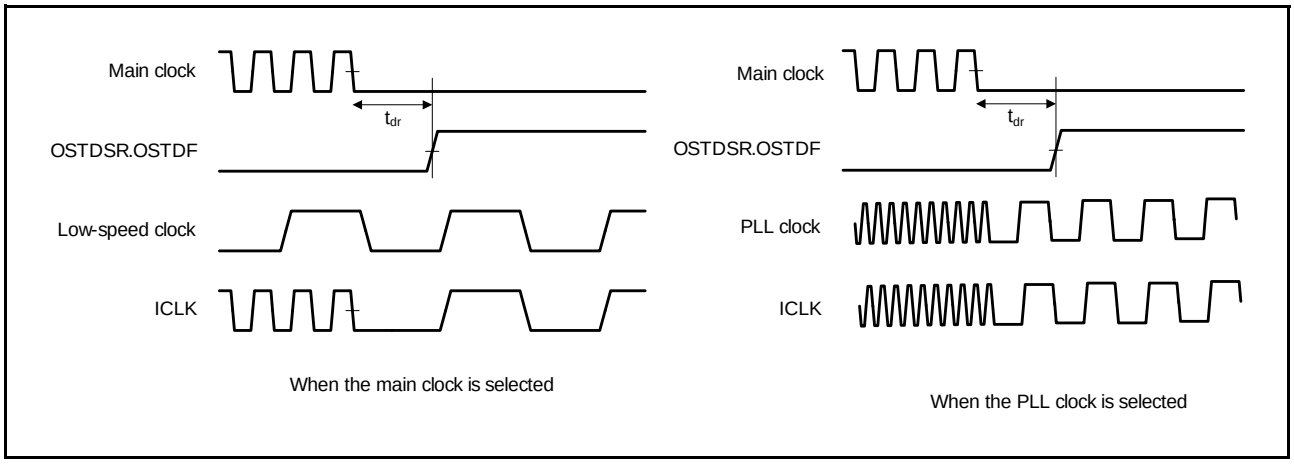


Figure 32.44 Oscillation Stop Detection Timing

32.11 ROM (Code Flash Memory) Characteristics

Table 32.47 ROM (Code Flash Memory) Characteristics (1)

Item		Symbol	Min.	Typ.	Max.	Unit	Conditions
Program/erase cycle*1		N_{PEC}	1000	—	—	Times	
Data retention	After 1000 times of erase	t_{DRP}	20*2, *3	—	—	Year	T_a = +85°C

Note 1. Definition of program/erase cycle: The program/erase cycle is the number of erasing for each block. When the number of program/erase cycles is n, each block can be erased n times. For instance, when 4-byte program is performed 256 times for different addresses in a 1-Kbyte block and then the block is erased, the program/erase cycle is counted as one. However, the same address cannot be programmed more than once before the next erase cycle (overwriting is prohibited).
Note 2. Characteristic when using the flash programmer and the self-programming library provided from Renesas Electronics.
Note 3. This result is obtained from reliability testing.

Table 32.48 ROM (Code Flash Memory) Characteristics (2) High-Speed Operating Mode

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V

Temperature range for program/erase: T_a = –40 to +105°C

Item		Symbol	FCLK = 1 MHz			FCLK = 32 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Program time	4-byte	t_{P4}	—	103	931	—	52	489	µs
Erase time	1-Kbyte	t_{E1K}	—	8.23	267	—	5.48	214	ms
	128-Kbyte	t_{E128K}	—	203	463	—	20	228	
Blank check time	4-byte	t_{BC4}	—	—	48	—	—	15.9	µs
	1-Kbyte	t_{BC1K}	—	—	1.58	—	—	0.127	ms
Erase operation forcible stop time		t_{SED}	—	—	21.6	—	—	12.8	µs
Start-up area switching setting time		t_{SAS}	—	12.6	543	—	6.16	432	ms
Access window setting time		t_{AWS}	—	12.6	543	—	6.16	432	ms
ROM mode transition wait time 1		t_{DIS}	2	—	—	2	—	—	µs
ROM mode transition wait time 2		t_{MS}	5	—	—	5	—	—	µs

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: The frequency accuracy of FCLK should be ±3.5%. Check the accuracy of the frequency from the clock source.

Table 32.49 ROM (Code Flash Memory) Characteristics (3) Middle-Speed Operating Mode

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V

Temperature range for program/erase: T_a = –40 to +85°C

Item		Symbol	FCLK = 1 MHz			FCLK = 8 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Program time	4-byte	t_{P4}	—	143	1330	—	96.8	932	µs
Erase time	1-Kbyte	t_{E1K}	—	8.3	269	—	5.85	219	ms
	128-Kbyte	t_{E128K}	—	203	464	—	46	260	
Blank check time	4-byte	t_{BC4}	—	—	78	—	—	50	µs
	1-Kbyte	t_{BC1K}	—	—	1.61	—	—	0.369	ms
Erase operation forcible stop time		t_{SED}	—	—	33.6	—	—	25.6	µs
Start-up area switching setting time		t_{SAS}	—	13.2	549	—	7.6	445	ms
Access window setting time		t_{AWS}	—	13.2	549	—	7.6	445	ms
ROM mode transition wait time 1		t_{DIS}	2	—	—	2	—	—	µs
ROM mode transition wait time 2		t_{MS}	3	—	—	3	—	—	µs

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.

Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.

Note: The frequency accuracy of FCLK should be ±3.5%. Check the accuracy of the frequency from the clock source.

32.12 E2 DataFlash (Data Flash Memory) Characteristics

Table 32.50 E2 DataFlash Characteristics (1)

Item		Symbol	Min.	Typ.	Max.	Unit	Conditions
Program/erase cycle*1		N_{DPEC}	100000	1000000	—	Times	
Data retention	After 10000 times of erase	t_{DDRP}	20*2, *3	—	—	Year	T_a = +85°C
	After 100000 times of erase		5*2, *3	—	—		
	After 1000000 times of erase		—	1*2, *3	—		T_a = +25°C

Note 1. Definition of program/erase cycle: The program/erase cycle is the number of erasing for each block. When the number of program/erase cycles is n, each block can be erased n times. For instance, when 1-byte program is performed 1000 times for different addresses in a 1-Kbyte block and then the block is erased, the program/erase cycle is counted as one. However, the same address cannot be programmed more than once before the next erase cycle (overwriting is prohibited).
Note 2. Characteristic when using the flash programmer and the self-programming library provided from Renesas Electronics.
Note 3. These results are obtained from reliability testing.

Table 32.51 E2 DataFlash Characteristics (2): high-speed operating mode

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V

Temperature range for program/erase: T_a = –40 to +105°C

Item		Symbol	FCLK = 1 MHz			FCLK = 32 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Program time	1-byte	t_{DP1}	—	86	761	—	40.5	374	µs
Erase time	1-Kbyte	t_{DE1K}	—	17.4	456	—	6.15	228	ms
	4-Kbyte	t_{DE4K}	—	35.8	474	—	7.5	229	
Blank check time	1-byte	t_{DBC1}	—	—	48	—	—	15.9	µs
	1-Kbyte	t_{DBC1K}	—	—	1.58	—	—	0.127	ms
Erase operation forcible stop time		t_{DSED}	—	—	21.5	—	—	12.8	µs
DataFlash STOP recovery time		t_{DSTOP}	5.0	—	—	5	—	—	µs

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: The frequency accuracy of FCLK should be ±3.5%.

Table 32.52 E2 DataFlash Characteristics (3): middle-speed operating mode

Conditions: VCC = 2.7 V to 5.5 V, AVCC0 = VCC to 5.5 V, VSS = AVSS0 = 0 V

Temperature range for program/erase: T_a = –40 to +85°C

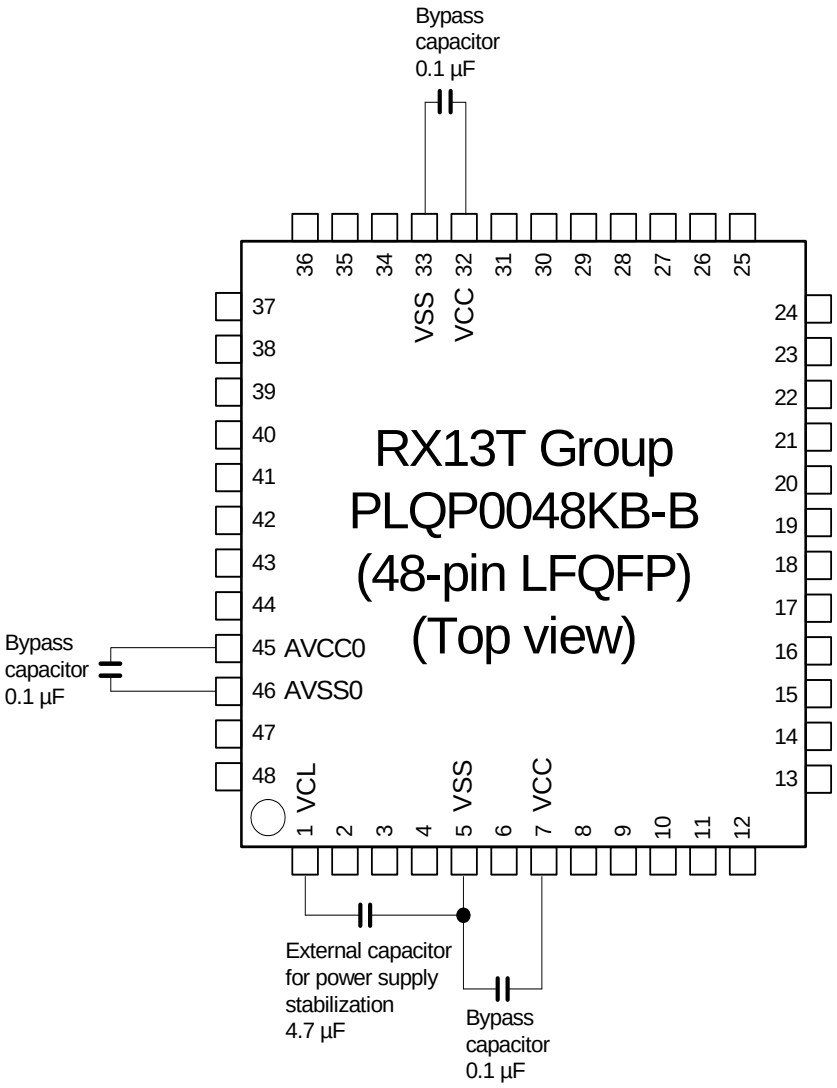
Item		Symbol	FCLK = 1 MHz			FCLK = 8 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Program time	1-byte	t_{DP1}	—	126	1160	—	85.4	818	µs
Erase time	1-Kbyte	t_{DE1K}	—	17.5	457	—	7.76	259	ms
	4-Kbyte	t_{DE4K}	—	35.9	476	—	9.0	260	
Blank check time	1-byte	t_{DBC1}	—	—	78	—	—	50	µs
	1-Kbyte	t_{DBC1K}	—	—	1.61	—	—	0.369	ms
Erase operation forcible stop time		t_{DSED}	—	—	33.5	—	—	25.5	µs
DataFlash STOP recovery time		t_{DSTOP}	720	—	—	720	—	—	ns

Note: Does not include the time until each operation of the flash memory is started after instructions are executed by software.
Note: The lower-limit frequency of FCLK is 1 MHz during programming or erasing of the flash memory. When using FCLK at below 4 MHz, the frequency can be set to 1 MHz, 2 MHz, or 3 MHz. A non-integer frequency such as 1.5 MHz cannot be set.
Note: The frequency accuracy of FCLK should be ±3.5%.

32.13 Usage Notes

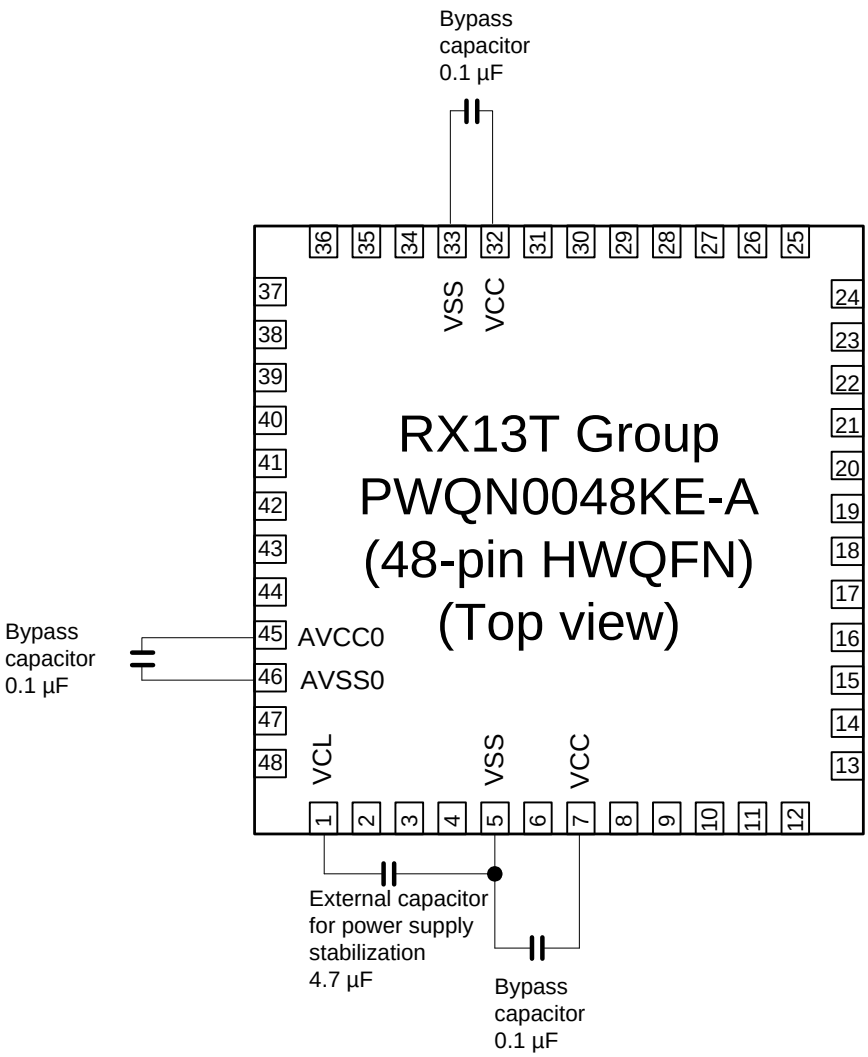
32.13.1 Connecting VCL Capacitor and Bypass Capacitors

This MCU integrates an internal voltage-down circuit, which is used for lowering the power supply voltage in the internal MCU to adjust automatically to the optimum level. A 4.7-µF capacitor needs to be connected between this internal voltage-down power supply (VCL pin) and VSS pin. Figure 32.45 to Figure 32.48 shows how to connect external capacitors. Place an external capacitor close to the pins. Do not apply the power supply voltage to the VCL pin. Insert a multilayer ceramic capacitor as a bypass capacitor between each pair of the power supply pins. Implement a bypass capacitor to the MCU power supply pins as close as possible. Use a recommended value of 0.1 µF as the capacitance of the capacitors. For the capacitors related to crystal oscillation, see section 9, Clock Generation Circuit. For the capacitors related to analog modules, also see section 26, 12-Bit A/D Converter (S12ADF).
For notes on designing the printed circuit board, see the descriptions of the application note “Hardware Design Guide” (R01AN1411EJ). The latest version can be downloaded from Renesas Electronics Website.



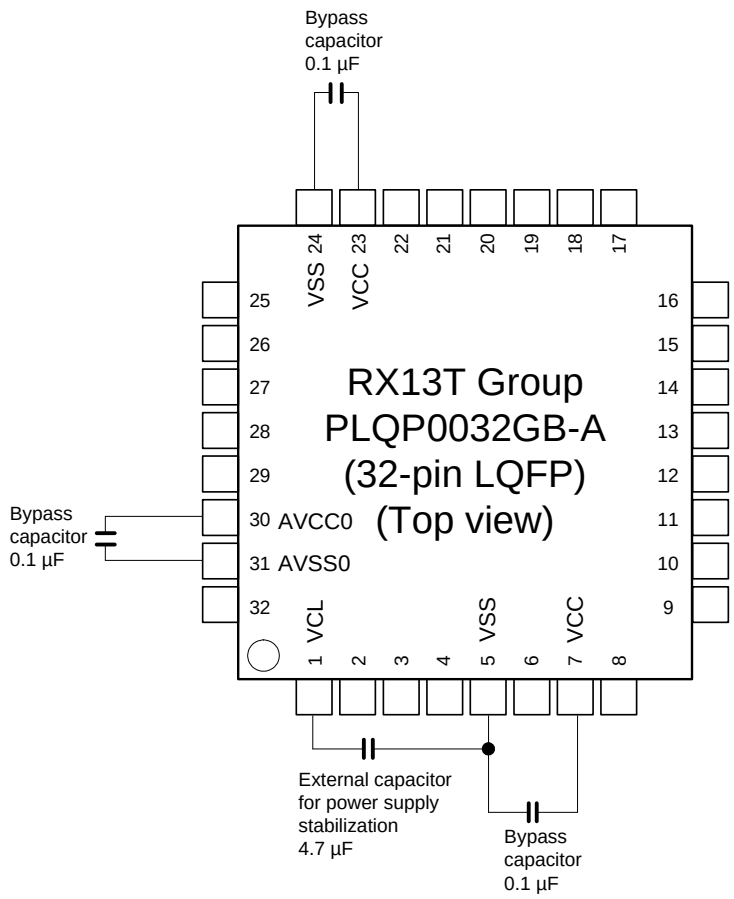
Note. Do not apply the power supply voltage to the VCL pin.
Use a 4.7-µF multilayer ceramic for the VCL pin and place it close to the pin.
A recommended value is shown for the capacitance of the bypass capacitors.

Figure 32.45 Connecting Capacitors (48-Pin LFQFP)



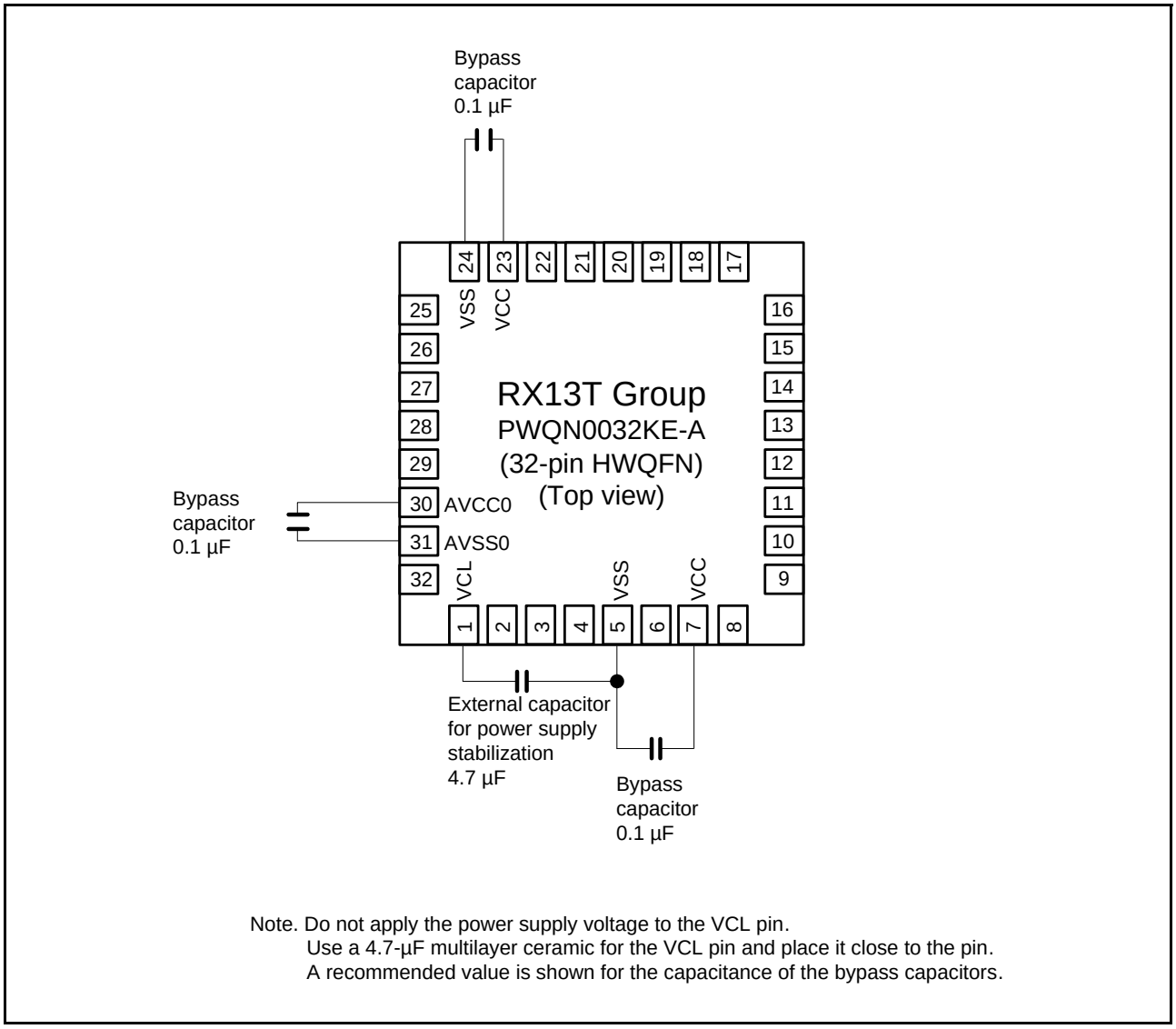
Note. Do not apply the power supply voltage to the VCL pin.
Use a 4.7-µF multilayer ceramic for the VCL pin and place it close to the pin.
A recommended value is shown for the capacitance of the bypass capacitors.

Figure 32.46 Connecting Capacitors (48-Pin HWQFN)



Note. Do not apply the power supply voltage to the VCL pin.
Use a 4.7-µF multilayer ceramic for the VCL pin and place it close to the pin.
A recommended value is shown for the capacitance of the bypass capacitors.

Figure 32.47 Connecting Capacitors (32-Pin LQFP)



Note. Do not apply the power supply voltage to the VCL pin.
Use a 4.7-µF multilayer ceramic for the VCL pin and place it close to the pin.
A recommended value is shown for the capacitance of the bypass capacitors.

Figure 32.48 Connecting Capacitors (32-Pin HWQFN)

Appendix 1. Port States in Each Processing Mode

Table 1.1 Port States in Each Processing Mode

Port Name (Pin Name)	Reset	Software Standby Mode
P10, P11 (IRQ0, IRQ1)	Hi-Z	Keep-O*1
P22, P23, P24 (IRQ2, IRQ4, IRQ3)	Hi-Z	Keep-O*1
P36, P37	Hi-Z	Keep-O
P40 to P47	Hi-Z	Keep-O
P70 (IRQ5)	Hi-Z	Keep-O*1
P71 to P76	Hi-Z	Keep-O
P93, P94 (IRQ0, IRQ1)	Hi-Z	Keep-O*1
PA2 (IRQ4)	Hi-Z	Keep-O*1
PA3	Hi-Z	Keep-O
PB0, PB2, PB3, PB5, PB6	Hi-Z	Keep-O
PB1, PB4, PB7 (IRQ2, IRQ3, IRQ5)	Hi-Z	Keep-O*1
PD3	Hi-Z	Keep-O
PD4, PD5, PD6 (IRQ2, IRQ3, IRQ5)	Hi-Z	Keep-O*1
PE2 (NMI/IRQ0)	Hi-Z	Keep-O*1

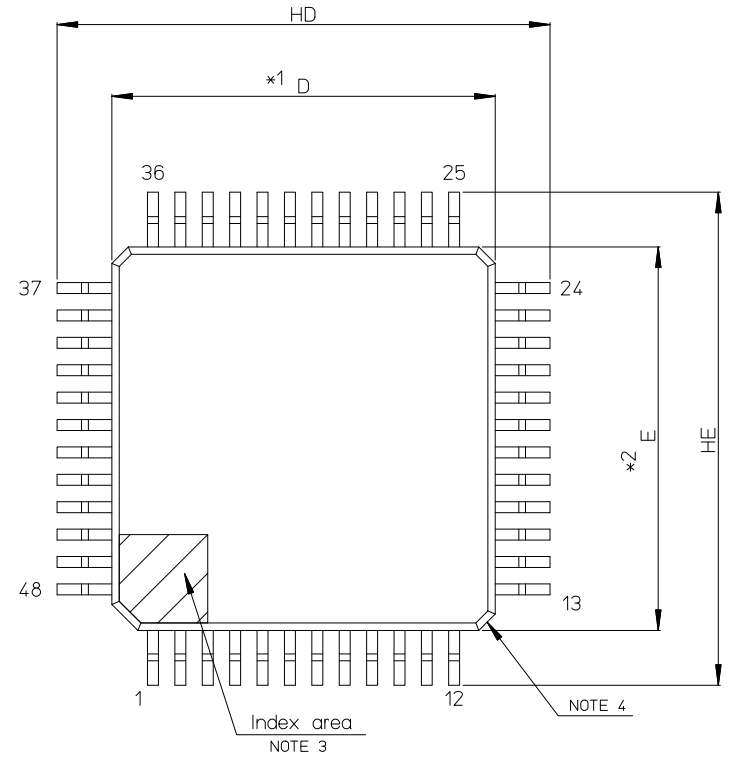
Keep-O: Output pins retain their previous values, and input pins become high-impedance.
Hi-Z: High-impedance

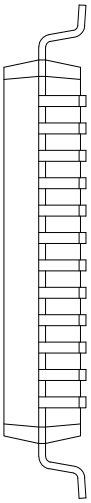
Note 1. Input is enabled if the pin is specified as the software standby mode canceling source while it is used as an external interrupt pin.

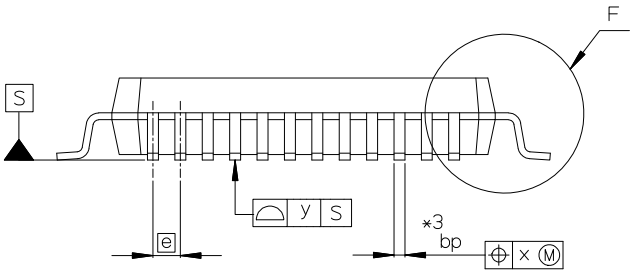
Appendix 2. Package Dimensions

Information on the latest version of the package dimensions or mountings has been displayed in “Packages” on Renesas Electronics Corporation website.

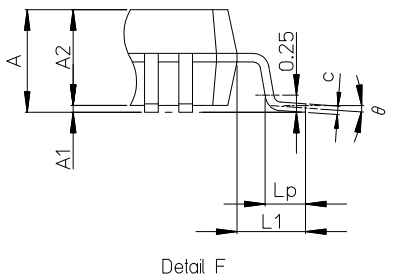
JEITA Package Code	RENESAS Code	Previous Code	MASS[Typ.]
P-LFQFP48-7x7-0.50	PLQP0048KB-B	——	0.2g







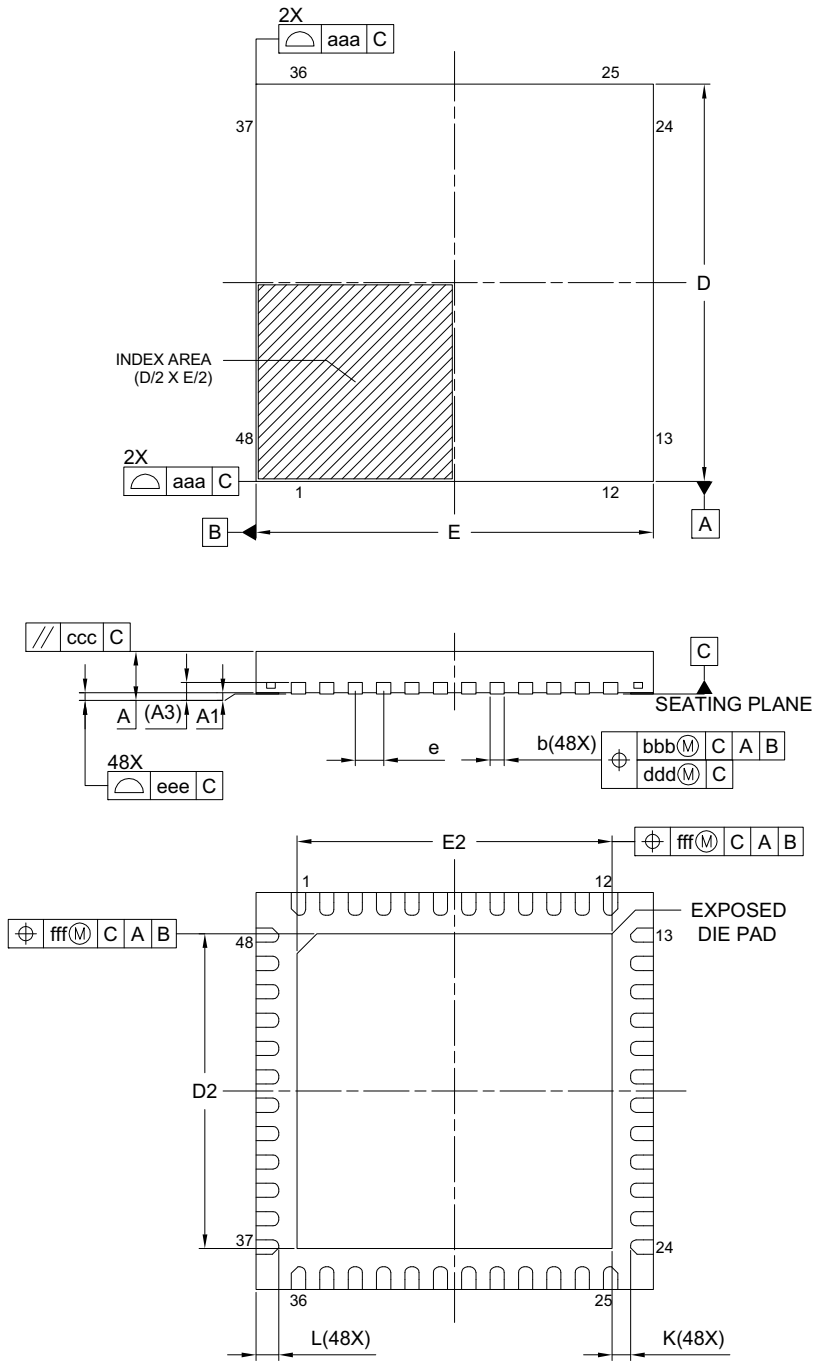
NOTE)
1. DIMENSIONS "*1" AND "*2" DO NOT INCLUDE MOLD FLASH.
2. DIMENSION "*3" DOES NOT INCLUDE TRIM OFFSET.
3. PIN 1 VISUAL INDEX FEATURE MAY VARY, BUT MUST BE LOCATED WITHIN THE HATCHED AREA.
4. CHAMFERS AT CORNERS ARE OPTIONAL, SIZE MAY VARY.



Reference Symbol	Dimension in Millimeters		
	Min	Nom	Max
D	6.9	7.0	7.1
E	6.9	7.0	7.1
A2	——	1.4	——
HD	8.8	9.0	9.2
HE	8.8	9.0	9.2
A	——	——	1.7
A1	0.05	——	0.15
bp	0.17	0.20	0.27
c	0.09	——	0.20
θ	0°	3.5°	8°
e	——	0.5	——
x	——	——	0.08
y	——	——	0.08
Lp	0.45	0.6	0.75
L1	——	1.0	——

Figure A 48-Pin LFQFP (PLQP0048KB-B)

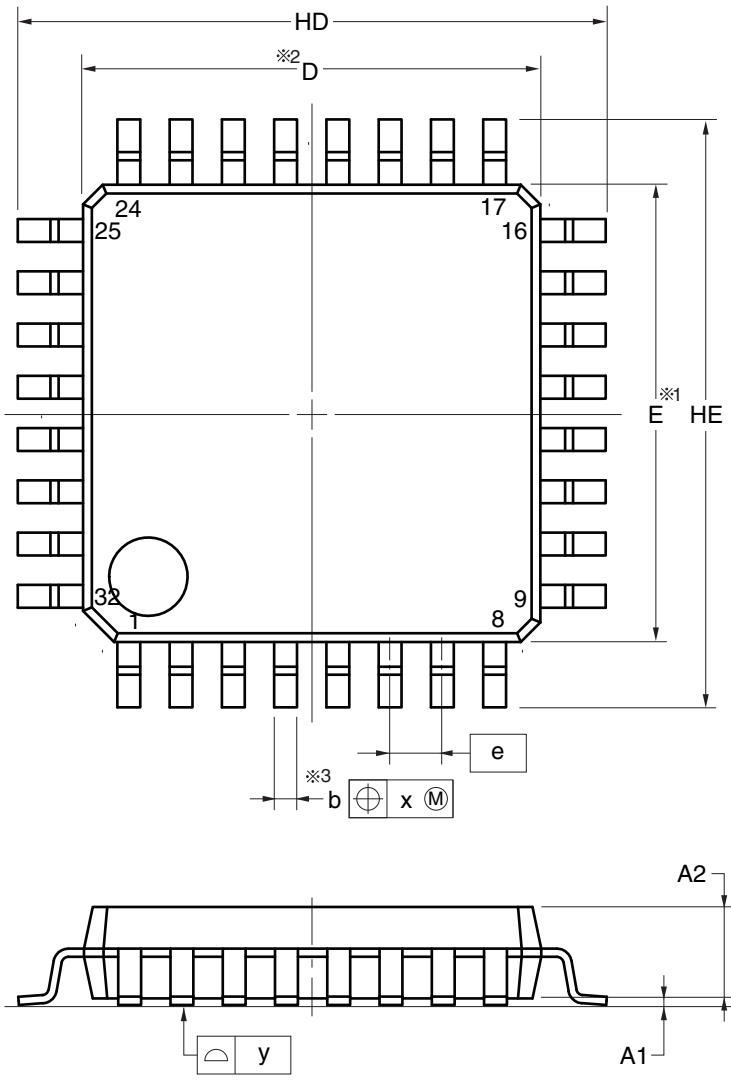
JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN048-7x7-0.50	PWQN0048KE-A	0.13

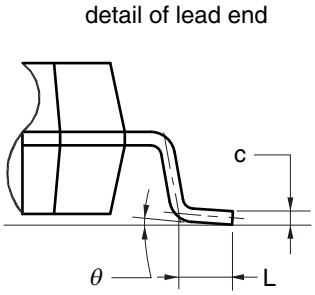


Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.20	0.25	0.30
D		7.00 BSC	
E		7.00 BSC	
e		0.50 BSC	
L	0.30	0.40	0.50
K	0.20	—	—
D_2	5.50	5.55	5.60
E_2	5.50	5.55	5.60
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

Figure B 48-Pin HWQFN (PWQN0048KE-A)

JEITA Package Code	RENESAS Code	Previous Code	MASS (TYP.) [g]
P-LQFP32-7x7-0.80	PLQP0032GB-A	P32GA-80-GBT-1	0.2





(UNIT:mm)

ITEM	DIMENSIONS
D	7.00±0.10
E	7.00±0.10
HD	9.00±0.20
HE	9.00±0.20
A	1.70 MAX.
A1	0.10±0.10
A2	1.40
b	0.37±0.05
c	0.145±0.055
L	0.50±0.20
θ	0° to 8°
e	0.80
x	0.20
y	0.10

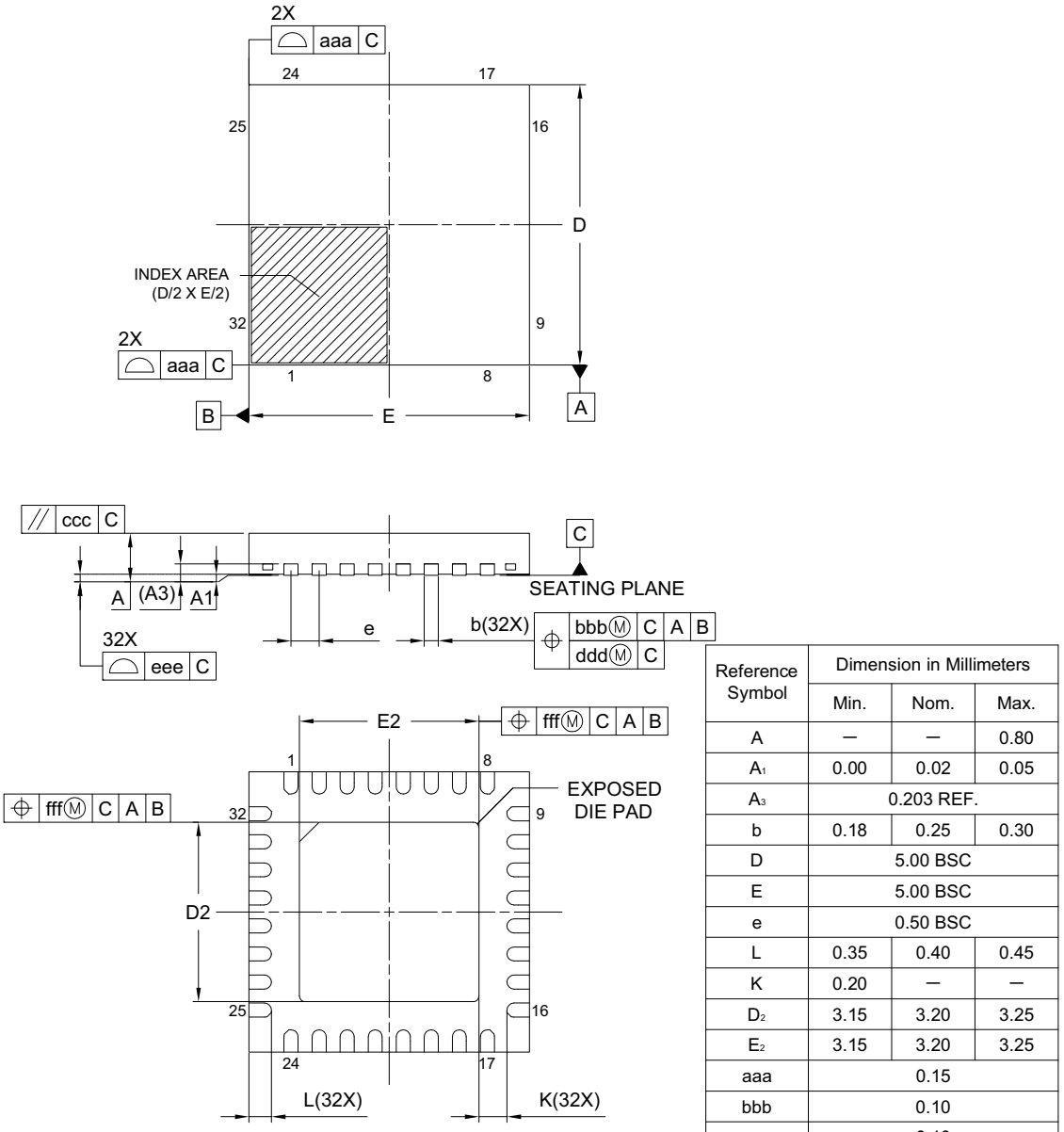
NOTE

1.Dimensions "※1" and "※2" do not include mold flash.

2.Dimension "※3" does not include trim offset.

Figure C 32-Pin LQFP (PLQP0032GB-A)

JEITA Package code	RENESAS code	MASS(TYP.)[g]
P-HWQFN032-5x5-0.50	PWQN0032KE-A	0.06



Reference Symbol	Dimension in Millimeters		
	Min.	Nom.	Max.
A	—	—	0.80
A_1	0.00	0.02	0.05
A_3		0.203 REF.	
b	0.18	0.25	0.30
D		5.00 BSC	
E		5.00 BSC	
e		0.50 BSC	
L	0.35	0.40	0.45
K	0.20	—	—
D_2	3.15	3.20	3.25
E_2	3.15	3.20	3.25
aaa		0.15	
bbb		0.10	
ccc		0.10	
ddd		0.05	
eee		0.08	
fff		0.10	

Figure D 32-Pin HWQFN (PWQN0032KE-A)

REVISION HISTORY	RX13T Group User's Manual: Hardware

Classifications

- Items with Technical Update document number: Changes according to the corresponding issued Technical Update
- Items without Technical Update document number: Minor changes that do not require Technical Update to be issued

Rev.	Date	Description		Classification
		Page	Summary	
1.00	Jul 31, 2019	—	First edition, issued	
1.10	Mar 16, 2021	Features		
		32	48-pin HWQFN and 32-pin HWQFN Package Images, added	
		1. Overview		
		All	48-pin HWQFN and 32-pin HWQFN specifications, added	
		9. Clock Generation Circuit		
		159	9.7.2 Note on Rewriting the SCKCR3 Register, added	TN-RX*-A0224B/E
		10. Clock Frequency Accuracy Measurement Circuit (CAC)		
		170, 171	10.3.1 Measuring Clock Frequency, changed	
		11. Low Power Consumption		
		181, 182	11.2.5 Operating Power Control Register (OPCCR)	
		14. Interrupt Controller (ICUb)		
		221	14.3.1 Interrupt Vector Table, changed	
		222 to 227	Table 14.3 Interrupt Vector Table, changed	
		16. Data Transfer Controller (DTCb)		
		261	16.2.9 DTC Vector Base Register (DTCVBR), changed	
		264	16.2.13 DTC Index Table Base Register (DTCIBR), changed	
		19. Multi-Function Timer Pulse Unit 3 (MTU3c)		
		329, 330	Table 19.2 MTU Functions, changed	
		334	Table 19.5 CCLR[2:0] (MTU1 and MTU2), changed	
		380	19.2.31 Timer Waveform Control Register (TWCRA), changed	
		23. Serial Communications Interface (SCIg, SCIh)		
		602 to 605	23.2.8 Serial Control Register (SCR), changed	
		611, 612	23.2.10 Smart Card Mode Register (SCMR), changed	
		648	23.3.6 SCI Initialization (Asynchronous Mode), changed	
		648	Figure 23.7 Sample SCI Initialization Flowchart (Asynchronous Mode), changed	
		664	23.5.2 CTS and RTS Functions, changed	
		665	32.5.3 SCI Initialization (Clock Synchronous Mode), changed	
		665	Figure 23.23 Example of SCI Initialization Flowchart (Clock Synchronous Mode), changed	
		678	Figure 23.37 Example of SCI Initialization Flowchart (Smart Card Interface Mode), changed	
		691	Figure 23.50 Example of the Flowchart of SCI Initialization (for Simple I^2C Mode), changed	
		693	Figure 23.53 Example of the Procedure for Master Transmission Operations in Simple I^2C Mode (with Transmission Interrupts and Reception Interrupts in Use), changed	
		700	23.8.5 SCI Initialization (Simple SPI Mode), changed	
		718	23.12.2 Interrupts in Asynchronous Mode, Clock Synchronous Mode, and Simple SPI Mode, changed	
		24. I^2C-bus Interface (RIICa)		
		731, 732	Table 24.1 RIIC Specifications, changed	
		732	Figure 24.1 RIIC Block Diagram, changed	
		733	Figure 24.2 I/O Pin Connection to the External Circuit (I^2C-bus Configuration Example), changed	
		734, 735	24.2.1 I^2C-bus Control Register 1 (ICCR1), changed	
		736 to 739	24.2.2 I^2C-bus Control Register 2 (ICCR2), changed	
		740	24.2.3 I^2C-bus Mode Register 1 (ICMR1), changed	
		741, 742	24.2.4 I^2C-bus Mode Register 2 (ICMR2), changed	
		743, 744	24.2.5 I^2C-bus Mode Register 3 (ICMR3), changed	

Rev.	Date	Description		Classification
		Page	Summary	
1.10	Mar 16, 2021	745, 746	24.2.6 I^2C-bus Function Enable Register (ICFER), changed	TN-RX*-A0232A/E
		751 to 753	24.2.9 I^2C-bus Status Register 1 (ICSR1), changed	
		754 to 756	24.2.10 I^2C-bus Status Register 2 (ICSR2), changed	TN-RX*-A0232A/E
		755	Table 24.4 Relationship between Arbitration-Lost Generation Sources and Arbitration-Lost Enable Functions, changed	
		759	24.2.13 I^2C-bus Bit Rate Low-Level Register (ICBRL), changed	
		762	24.2.16 I^2C-bus Receive Data Register (ICDRR), changed	
		765	24.3.3 Master Transmit Operation, changed	
		768, 769	24.3.4 Master Receive Operation, changed	
		774	24.3.5 Slave Transmit Operation, changed	
		780	24.5 SDA Output Delay Function, changed	
		780	Figure 24.22 SDA Output Delay Function, changed	
		782	24.7.1 Slave-Address Match Detection, changed	
		785	24.7.3 Device-ID Address Detection, changed	TN-RX*-A0232A/E
		786	Figure 24.28 Set/Clear Timing of the AASy and DID Flags during Reception of Device-ID Address, changed	TN-RX*-A0232A/E
		789	24.8.2 NACK Reception Transfer Suspension Function, changed	TN-RX*-A0232A/E
		789	Figure 24.31 Suspension of Data Transmission When NACK is Received (NACKE = 1), changed	TN-RX*-A0232A/E
		800	24.11.2 Additional SCL Output Function, changed	TN-RX*-A0232A/E
		800	Figure 24.40 Additional SCL Output Function (CLO Bit), changed	TN-RX*-A0232A/E
		805	Table 24.7 Reset States of Registers and Functions When a Reset is Applied or a Condition is Detected	TN-RX*-A0232A/E
		29. Data Operation Circuit (DOC)		
		All	Term unified Notation unified	
		31. Flash Memory (FLASH)		
		929	31.4.17 Flash Write Buffer Register H (FWBH), changed	
		936	31.4.26 Unique ID Register n (UIDRn) (n = 0 to 31), changed	
		938	31.6 Area Protection, changed	
		940	Figure 31.6 Procedure for Transition from E2 DataFlash Access Disabled Mode to ROM/E2 DataFlash Read Mode, changed	
		962, 963	31.10.1 State Transition in Boot Mode (SCI Interface), changed	
		964	Table 31.12 Information Regarding the States Note 1, changed	
		965	31.10.5.1 Supported Device Inquiry, changed	
		972	Table 31.18 Acceptable Commands for Each State, changed	
		974	31.10.8.3 Data Area Program, changed	
		976, 977	31.10.9.1 Memory Read, changed	
		981	31.11 Serial Programmer Operation in Boot Mode (SCI Interface), changed	
		982	31.11.1 Bit Rate Automatic Adjustment, changed	
		983	31.11.2 Receive the MCU Information, changed	
		984	31.11.3 Select the Device and Change the Bit Rate, changed	
		985	31.11.4 Enter the Program/Erase Host Command Wait State, changed	
		986	31.11.5 Unlock Boot Mode ID Code Protection, changed	
		987	31.11.6 Erase the User Area and Data Area, changed	
		988	31.11.7 Program the User Area and Data Area, changed	
		989	31.11.8 Check Data in the User Area, changed	
		990	31.11.9 Check Data in the Data Area, changed	
		991	31.11.10 Set the Access Window in the User Area, changed	
		32. Electrical Characteristics		
		1013 to 1026	32.4.5 Timing of On-Chip Peripheral Modules, order of tables changed	

RX13T Group User’s Manual: Hardware

Publication Date: Rev.1.00 Jul 31, 2019
Rev.1.10 Mar 16, 2021

Published by: Renesas Electronics Corporation

RX13T Group

Renesas Electronics Corporation

R01UH0822EJ0110