

# RX630 Group

User's Manual: Hardware

RENESAS 32-Bit MCU  
RX Family / RX600 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>).

## 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 of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
2. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
3. Renesas Electronics does not assume any liability for infringement of 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. 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 should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from such alteration, modification, copy or otherwise misappropriation of Renesas Electronics product.
5. Renesas Electronics products are classified according to the following two quality grades: "Standard" and "High Quality". The recommended 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; and industrial robots etc.  
"High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; and safety equipment etc.  

Renesas Electronics products are neither intended nor 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 damages (nuclear reactor control systems, military equipment etc.). You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application for which it is not intended. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for which the product is not intended by Renesas Electronics.
6. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
7. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, 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, please evaluate the safety of the final products or systems manufactured by you.
8. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
9. Renesas Electronics products and technology may 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 should not use Renesas Electronics products or technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. When exporting the Renesas Electronics products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations.
10. It is the responsibility of the buyer or distributor of Renesas Electronics products, who distributes, disposes of, or otherwise places the product with a third party, to notify such third party in advance of the contents and conditions set forth in this document, Renesas Electronics assumes no responsibility for any losses incurred by you or third parties as a result of unauthorized use of Renesas Electronics products.
11. This document may not be reproduced or duplicated in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

## General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

### 1. Handling of Unused Pins

Handle unused pins in accord 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 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. Unused pins should be handled as described under Handling of Unused Pins in the manual.

### 2. Processing at Power-on

The state of the product is undefined at the moment 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 moment 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 moment 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 moment when power is supplied until the power reaches the level at which resetting has been specified.

### 3. Prohibition of Access to Reserved Addresses

Access to reserved addresses is prohibited.

- The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.

### 4. Clock Signals

After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has 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. Moreover, 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.

### 5. Differences between Products

Before changing from one product to another, i.e. to one with a different part number, confirm that the change will not lead to problems.

- The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

# Precautions when Porting Software among RX630 Group Products

When porting software from one RX630-group product to another, you need to pay attention to the following restrictions along with the differences in peripheral function and so on of the individual products.

## 1. Precautions on Low Power Consumption

For products with ROM capacities of 1.5 Mbytes or more or with 176 pins or more, precautions must be taken in transitions to the software standby mode as a low power consumption function. For details, see [section 11.7.8, Transition to Software Standby Mode for Products with 1.5-Mbyte ROM or More or Products with 176 Pins or More](#).

## 2. Precautions on D/A Converter (DAa)

The D/A converters (DAa) of products with ROM capacities of 1.5 Mbytes or more or with 176 pins or more do not include countermeasures against mutual interference between analog modules. For details, see [section 40.2.4, D/A A/D Synchronous Start Control Register \(DAADSCR\)](#).

## 3. Precautions on the Serial Communications Interface (SCIC)

The simple SPI mode of the SS input pin for the serial communications interface (SCIC) is not available in products with ROM capacities of 1.5 Mbytes or more or with 176 pins or more. For details, see [section 32, Serial Communications Interface \(SCIC, SCID\)](#).

## 4. Setting of Initial Value and Waiting Time for Sub-Clock Oscillator Wait Control Register (SOSCWTCR)

For products tagged with “\*2” in [Table 1.3, List of Products](#) in [section 1, Overview](#) of this manual, the initial value and waiting time after a reset by using the SOSCWTCR register are different from those for other products. For details, see [section 11.2.8, Sub-Clock Oscillator Wait Control Register \(SOSCWTCR\)](#).

## 5. Whether RTC Control Register 3 (RCR3) is Initialized by RTC Software Reset

For products tagged with “\*2” in [Table 1.3, List of Products](#), [List of Products](#) in [section 1, Overview](#) of this manual, whether the RCR3 register is or is not initialized by an RTC software reset differs from the case for other products. For details, see [section 28.2.19, RTC Control Register 3 \(RCR3\)](#).

# 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 RX630 Group. Before using any of the documents, please visit our web site to verify that you have the most up-to-date available version of the document.

Document Type	Contents	Document Title	Document No.
Data Sheet	Overview of hardware and electrical characteristics	RX630 Group Data sheet	—
User's manual: Hardware	Hardware specifications (pin assignments, memory maps, peripheral specifications, electrical characteristics, and timing charts) and descriptions of operation	RX630 Group User's manual: Hardware	This User's manual
User's manual: Software	Detailed descriptions of the CPU and instruction set	RX Family Series User's manual: Software	R01US0032EJ
Application Note	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 xxxxxh

b7	b6	b5	b4	b3	b2	b1	b0
—	...	...	...	—	—	—	...

Value after reset: x 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	...0	... Bit	0: ..... 1: Setting prohibited	R/W
b3 to b1	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b4	...4	... Bit	0: ..... 1: .....	R
b6, b5	...[1:0]	... Bit	0 0: ..... 0 1: ..... Settings other than above are prohibited.	R/(W)*
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. Make sure to 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 Communication 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

# Contents

1.	Overview .....	47
1.1	Outline of Specifications .....	47
1.2	List of Products .....	53
1.3	Block Diagram .....	56
1.4	Pin Functions .....	57
1.5	Pin Assignments .....	62
2.	CPU .....	97
2.1	Features.....	97
2.2	Register Set of the CPU .....	98
2.2.1	General-Purpose Registers (R0 to R15) .....	99
2.2.2	Control Registers .....	99
2.2.2.1	Interrupt Stack Pointer (ISP)/User Stack Pointer (USP) .....	99
2.2.2.2	Interrupt Table Register (INTB) .....	99
2.2.2.3	Program Counter (PC) .....	100
2.2.2.4	Processor Status Word (PSW) .....	100
2.2.2.5	Backup PC (BPC) .....	101
2.2.2.6	Backup PSW (BPSW) .....	102
2.2.2.7	Fast Interrupt Vector Register (FINTV) .....	102
2.2.2.8	Floating-Point Status Word (FPSW) .....	102
2.2.3	Register Associated with DSP Instructions .....	105
2.2.3.1	Accumulator (ACC) .....	105
2.3	Processor Mode.....	105
2.3.1	Supervisor Mode.....	105
2.3.2	User Mode .....	105
2.3.3	Privileged Instruction .....	105
2.3.4	Switching Between Processor Modes.....	106
2.4	Data Types .....	106
2.5	Endian.....	107
2.5.1	Switching the Endian.....	107
2.5.2	Access to I/O Registers.....	110
2.5.3	Notes on Access to I/O Registers .....	110
2.5.4	Data Arrangement.....	111
2.5.4.1	Data Arrangement in Registers .....	111
2.5.4.2	Data Arrangement in Memory .....	111
2.5.5	Notes on the Allocation of Instruction Codes .....	111
2.6	Vector Table .....	112
2.6.1	Fixed Vector Table .....	112
2.6.2	Relocatable Vector Table .....	112
2.7	Operation of Instructions.....	113
2.7.1	Data Prefetching by the RMPA Instruction and the String-Manipulation Instructions .....	113



2.8	Pipeline .....	113
2.8.1	Overview .....	113
2.8.2	Instructions and Pipeline Processing .....	115
2.8.2.1	Instructions Converted into Single Micro-Operation and Pipeline Processing .....	115
2.8.2.2	Instructions Converted into Multiple Micro-Operations and Pipeline Processing .....	117
2.8.2.3	Pipeline Basic Operation .....	120
2.8.3	Calculation of the Instruction Processing Time .....	122
2.8.4	Numbers of Cycles for Response to Interrupts.....	122
3.	Operating Modes .....	123
3.1	Operating Mode Types and Selection .....	123
3.2	Register Descriptions.....	124
3.2.1	Mode Monitor Register (MDMONR) .....	124
3.2.2	Mode Status Register (MDSR).....	124
3.2.3	System Control Register 0 (SYSCR0).....	125
3.2.4	System Control Register 1 (SYSCR1).....	126
3.3	Details of Operating Modes .....	127
3.3.1	Single-Chip Mode.....	127
3.3.2	On-Chip ROM Enabled Extended Mode.....	127
3.3.3	On-Chip ROM Disabled Extended Mode .....	127
3.3.4	Boot Mode .....	127
3.3.5	USB Boot Mode .....	127
3.3.6	User Boot Mode.....	127
3.4	Transitions of Operating Modes.....	129
3.4.1	Operating Mode Transitions Determined by the Mode-Setting Pins .....	129
3.4.2	Operating Mode Transitions According to Register Setting .....	130
4.	Address Space .....	131
4.1	Address Space .....	131
4.2	External Address Space .....	133
5.	I/O Registers .....	134
5.1	I/O Register Addresses (Address Order).....	136
6.	Resets .....	178
6.1	Overview.....	178
6.2	Register Descriptions.....	181
6.2.1	Reset Status Register 0 (RSTSR0) .....	181
6.2.2	Reset Status Register 1 (RSTSR1) .....	183
6.2.3	Reset Status Register 2 (RSTSR2) .....	183
6.2.4	Software Reset Register (SWRR).....	184
6.3	Operation .....	185
6.3.1	RES# pin Reset.....	185
6.3.2	Power-On Reset and Voltage Monitoring 0 Reset .....	185

6.3.3	Voltage Monitoring 1 Reset and Voltage Monitoring 2 Reset .....	186
6.3.4	Deep Software Standby Reset .....	188
6.3.5	Independent Watchdog Timer Reset .....	188
6.3.6	Watchdog Timer Reset .....	188
6.3.7	Software Reset .....	188
6.3.8	Determination of Cold/Warm Start .....	189
6.3.9	Determination of Reset Generation Source .....	190
7.	Option-Setting Memory .....	191
7.1	Overview .....	191
7.2	Register Descriptions .....	192
7.2.1	Option Function Select Register 0 (OFS0) .....	192
7.2.2	Option Function Select Register 1 (OFS1) .....	196
7.2.3	Endian Select Register B (MDEB), Endian Select Register S (MDES) .....	197
7.3	UB Codes .....	198
7.3.1	UB Code A .....	198
7.3.2	UB Code B .....	198
7.4	Usage Note .....	198
7.4.1	Setting Example of Option-Setting Memory .....	198
8.	Voltage Detection Circuit (LVDA) .....	199
8.1	Overview .....	199
8.2	Register Descriptions .....	202
8.2.1	Voltage Monitoring 1 Circuit Control Register 1 (LVD1CR1) .....	202
8.2.2	Voltage Monitoring 1 Circuit Status Register (LVD1SR) .....	202
8.2.3	Voltage Monitoring 2 Circuit Control Register 1 (LVD2CR1) .....	203
8.2.4	Voltage Monitoring 2 Circuit Status Register (LVD2SR) .....	203
8.2.5	Voltage Monitoring Circuit Control Register (LVCMPCR) .....	204
8.2.6	Voltage Detection Level Select Register (LVDLVLR) .....	204
8.2.7	Voltage Monitoring 1 Circuit Control Register 0 (LVD1CR0) .....	205
8.2.8	Voltage Monitoring 2 Circuit Control Register 0 (LVD2CR0) .....	206
8.3	VCC Input Voltage Monitor .....	208
8.3.1	Monitoring Vdet0 .....	208
8.3.2	Monitoring Vdet1 .....	208
8.3.3	Monitoring Vdet2 .....	208
8.4	Reset from Voltage Monitor 0 .....	209
8.5	Interrupt and Reset from Voltage Monitor 1 .....	210
8.6	Interrupt and Reset from Voltage Monitor 2 .....	212
9.	Clock Generation Circuit .....	214
9.1	Overview .....	214
9.2	Register Descriptions .....	216
9.2.1	System Clock Control Register (SCKCR) .....	216

9.2.2	System Clock Control Register 2 (SCKCR2).....	218
9.2.3	System Clock Control Register 3 (SCKCR3).....	219
9.2.4	PLL Control Register (PLLCR) .....	220
9.2.5	PLL Control Register 2 (PLLCR2) .....	221
9.2.6	External Bus Clock Control Register (BCKCR) .....	222
9.2.7	Main Clock Oscillator Control Register (MOSCCR).....	223
9.2.8	Sub-Clock Oscillator Control Register (SOSCCR).....	224
9.2.9	Low-Speed On-Chip Oscillator Control Register (LOCOCR).....	225
9.2.10	IWDT-Dedicated On-Chip Oscillator Control Register (ILOCOCR).....	226
9.2.11	High-Speed On-Chip Oscillator Control Register (HOCOGR) .....	227
9.2.12	Oscillation Stop Detection Control Register (OSTDCR).....	228
9.2.13	Oscillation Stop Detection Status Register (OSTDSR).....	229
9.2.14	Main Clock Oscillator Forced Oscillation Control Register (MOFCR).....	230
9.2.15	High-Speed On-Chip Oscillator Power Supply Control Register (HOCOPCR).....	231
<b>9.3</b>	<b>Main Clock Oscillator .....</b>	<b>231</b>
9.3.1	Connecting a Crystal Resonator .....	231
9.3.2	External Clock Input.....	233
9.3.3	Notes on the External Clock Input .....	233
<b>9.4</b>	<b>Sub-Clock Oscillator .....</b>	<b>233</b>
9.4.1	Connecting 32.768-kHz Crystal Resonator .....	233
9.4.2	Handling of Pins when Sub-Clock is Not Used .....	234
<b>9.5</b>	<b>Oscillation Stop Detection Function.....</b>	<b>235</b>
9.5.1	Oscillation Stop Detection and Operation after Detection .....	235
9.5.2	Oscillation Stop Detection Interrupts .....	236
<b>9.6</b>	<b>PLL Circuit .....</b>	<b>237</b>
<b>9.7</b>	<b>Internal Clock.....</b>	<b>237</b>
9.7.1	System Clock.....	237
9.7.2	Peripheral Module Clock.....	237
9.7.3	FlashIF Clock .....	237
9.7.4	External Bus Clock.....	237
9.7.5	USB Clock.....	238
9.7.6	CAN Clock .....	238
9.7.7	IEBUS Clock .....	238
9.7.8	RTC-Dedicated Clock .....	238
9.7.9	IWDT-Dedicated Clock.....	238
9.7.10	JTAG Clock.....	238
<b>9.8</b>	<b>Pin Settings When an Oscillator is Connected .....</b>	<b>239</b>
<b>9.9</b>	<b>Pin Settings When an External Clock is Connected .....</b>	<b>239</b>
<b>9.10</b>	<b>Usage Notes .....</b>	<b>239</b>
9.10.1	Notes on Clock Generation Circuit .....	239

9.10.2	Notes on Resonator.....	239
9.10.3	Notes on Board Design.....	239
9.10.4	Notes on Resonator Connect Pin.....	240
9.10.5	Notes on Sub-Clock Oscillator.....	240
10.	Frequency Measurement Circuit (MCK).....	242
10.1	Overview.....	242
10.2	Register Descriptions.....	245
10.2.1	Counter-Clock Extension Register n (SCKn) (n = 1, 2).....	245
10.3	Operation.....	246
10.4	Usage Notes.....	248
10.4.1	Setting the Module-Stop Control Registers.....	248
11.	Low Power Consumption.....	249
11.1	Overview.....	249
11.2	Register Descriptions.....	252
11.2.1	Standby Control Register (SBYCR).....	252
11.2.2	Module Stop Control Register A (MSTPCRA).....	253
11.2.3	Module Stop Control Register B (MSTPCRB).....	254
11.2.4	Module Stop Control Register C (MSTPCRC).....	256
11.2.5	Operating Power Control Register (OPCCR).....	257
11.2.6	Sleep Mode Return Clock Source Switching Register (RSTCKCR).....	259
11.2.7	Main Clock Oscillator Wait Control Register (MOSCWTCR).....	260
11.2.8	Sub-Clock Oscillator Wait Control Register (SOSCWTCR).....	261
11.2.9	PLL Wait Control Register (PLLWTCR).....	263
11.2.10	Deep Standby Control Register (DPSBYCR).....	265
11.2.11	Deep Standby Interrupt Enable Register 0 (DPSIER0).....	267
11.2.12	Deep Standby Interrupt Enable Register 1 (DPSIER1).....	268
11.2.13	Deep Standby Interrupt Enable Register 2 (DPSIER2).....	269
11.2.14	Deep Standby Interrupt Enable Register 3 (DPSIER3).....	270
11.2.15	Deep Standby Interrupt Flag Register 0 (DPSIFR0).....	271
11.2.16	Deep Standby Interrupt Flag Register 1 (DPSIFR1).....	272
11.2.17	Deep Standby Interrupt Flag Register 2 (DPSIFR2).....	273
11.2.18	Deep Standby Interrupt Flag Register 3 (DPSIFR3).....	275
11.2.19	Deep Standby Interrupt Edge Register 0 (DPSIEGR0).....	276
11.2.20	Deep Standby Interrupt Edge Register 1 (DPSIEGR1).....	277
11.2.21	Deep Standby Interrupt Edge Register 2 (DPSIEGR2).....	278
11.2.22	Deep Standby Interrupt Edge Register 3 (DPSIEGR3).....	278
11.2.23	Deep Standby Backup Register (DPSBKRY) (y = 0 to 31).....	279
11.3	Reducing Power Consumption by Switching Clock Signals.....	279
11.4	Module-Stop Function.....	279
11.5	Function for Lower Operating Power Consumption.....	280

11.5.1	Setting Operating Power Consumption Control Mode.....	280
11.6	Low Power Consumption Modes .....	281
11.6.1	Sleep Mode .....	281
11.6.1.1	Transition to Sleep Mode .....	281
11.6.1.2	Canceling Sleep Mode .....	281
11.6.1.3	Sleep Mode Return Clock Source Switching Function .....	282
11.6.2	All-Module Clock Stop Mode .....	282
11.6.2.1	Transition to All-Module Clock Stop Mode .....	282
11.6.2.2	Canceling All-Module Clock Stop Mode .....	283
11.6.3	Software Standby Mode .....	283
11.6.3.1	Transition to Software Standby Mode .....	283
11.6.3.2	Canceling Software Standby Mode .....	284
11.6.3.3	Example of Software Standby Mode Application .....	285
11.6.4	Deep Software Standby Mode.....	286
11.6.4.1	Transition to Deep Software Standby Mode .....	286
11.6.4.2	Canceling Deep Software Standby Mode .....	287
11.6.4.3	Pin States when Deep Software Standby Mode is Canceled .....	287
11.6.4.4	Example of Deep Software Standby Mode Application .....	288
11.6.4.5	Flowchart to Use Deep Software Standby Mode .....	289
11.7	Usage Notes.....	290
11.7.1	I/O Port States.....	290
11.7.2	Module-Stop State of DMAC and DTC .....	290
11.7.3	On-Chip Peripheral Module Interrupts.....	290
11.7.4	Write Access to MSTPCRA, MSTPCRB, and MSTPCRC .....	290
11.7.5	Input Buffer Control by DIRQnE Bit (n = 0 to 15).....	290
11.7.6	Timing of Wait Instructions .....	290
11.7.7	Rewrite the Register by DMAC and DTC in Sleep Mode .....	290
11.7.8	Transition to Software Standby Mode for Products with 1.5-Mbyte ROM or More or Products with 176 Pins or More .....	290
11.7.9	Canceling All-Module Clock Stop Mode.....	291
11.7.10	Point for Caution when Using the Sub-Clock as the Source of the System Clock .....	291
11.7.11	Points for Caution on Return from Software Standby.....	291
11.7.12	Point for Caution when Shifting from Low-Speed Operation Mode to Software Standby Mode .....	291
12.	Battery Backup Function.....	292
12.1	Overview.....	292
12.2	Operation.....	293
12.2.1	Battery Backup Function .....	293
12.3	Usage Notes .....	294
13.	Register Write Protection Function .....	295
13.1	Register Descriptions.....	296

13.1.1	Protect Register (PRCR).....	296
14.	Exception Handling .....	297
14.1	Exception Events.....	297
14.1.1	Undefined Instruction Exception.....	297
14.1.2	Privileged Instruction Exception .....	297
14.1.3	Access Exceptions .....	297
14.1.4	Floating-Point Exception.....	297
14.1.5	Reset .....	297
14.1.6	Non-Maskable Interrupt .....	298
14.1.7	Interrupt .....	298
14.1.8	Unconditional Trap.....	298
14.2	Exception Handling Procedure .....	299
14.3	Acceptance of Exception Events .....	300
14.3.1	Acceptance Timing and Saved PC Value.....	300
14.3.2	Vector and Site for Saving the Values in the PC and PSW.....	301
14.4	Hardware Processing for Accepting and Returning from Exceptions .....	301
14.5	Hardware Pre-Processing.....	302
14.5.1	Undefined Instruction Exception.....	302
14.5.2	Privileged Instruction Exception .....	302
14.5.3	Access Exceptions .....	302
14.5.4	Floating-Point Exception.....	302
14.5.5	Reset .....	303
14.5.6	Non-Maskable Interrupt .....	303
14.5.7	Interrupt .....	303
14.5.8	Unconditional Trap.....	303
14.6	Return from Exception Handling Routine .....	304
14.7	Priority of Exception Events.....	304
15.	Interrupt controller (ICUb) .....	305
15.1	Overview.....	305
15.2	Register Descriptions.....	307
15.2.1	Interrupt Request Register n (IRn) (n = interrupt vector number) .....	307
15.2.2	Interrupt Request Enable Register m (IERm) (m = 02h to 1Fh) .....	308
15.2.3	Interrupt Source Priority Register n (IPRn) (n = 000 to 253).....	309
15.2.4	Fast Interrupt Set Register (FIR) .....	310
15.2.5	Software Interrupt Activation Register (SWINTR).....	311
15.2.6	DTC Activation Enable Register n (DTCERn) (n = interrupt vector number) .....	311
15.2.7	DMAC Activation Request Select Register m (DMRSRm) (m = DMAC channel number)....	312
15.2.8	IRQ Control Register i (IRQCRi) (i = 0 to 15) .....	313
15.2.9	IRQ Pin Digital Filter Enable Register 0 (IRQFLTE0).....	314
15.2.10	IRQ Pin Digital Filter Enable Register 1 (IRQFLTE1).....	315

15.2.11	IRQ Pin Digital Filter Setting Register 0 (IRQFLTC0) .....	316
15.2.12	IRQ Pin Digital Filter Setting Register 1 (IRQFLTC1) .....	317
15.2.13	Non-Maskable Interrupt Status Register (NMISR) .....	318
15.2.14	Non-Maskable Interrupt Enable Register (NMIER) .....	320
15.2.15	Non-Maskable Interrupt Status Clear Register (NMICLR) .....	321
15.2.16	NMI Pin Interrupt Control Register (NMICR).....	322
15.2.17	NMI Pin Digital Filter Enable Register (NMIFLTE).....	322
15.2.18	NMI Pin Digital Filter Setting Register (NMIFLTC) .....	323
15.2.19	Group m Interrupt Source Register (GRPm) (m: group number) .....	324
15.2.20	Group m Interrupt Enable Register (GENm) (m = group number).....	327
15.2.21	Group m Interrupt Clear Register (GCRm) (m = group number) .....	329
15.2.22	Unit Selecting Register (SEL) .....	330
15.3	Vector Table .....	331
15.3.1	Interrupt Vector Table .....	331
15.3.2	Fast Interrupt Vector Table.....	338
15.3.3	Non-maskable Interrupt Vector Table.....	338
15.4	Peripheral Module Interrupt Request Groups and Unit Selection.....	338
15.4.1	Interrupt Request Groups.....	338
15.4.2	Unit Selection .....	340
15.5	Interrupt Operation.....	341
15.5.1	Detecting Interrupts .....	341
15.5.1.1	Operation of Status Flags for Edge-Detected Interrupts .....	342
15.5.1.2	Operation of Status Flags for Level-Detected Interrupts .....	344
15.5.1.3	Edge Detection Group Interrupts and Interrupt Status Flags .....	345
15.5.1.4	Level Detection Group Interrupts and Interrupt Status Flags .....	347
15.5.1.5	Unit Selection and Interrupt Status Flags .....	348
15.5.2	Enabling and Disabling Interrupt Sources.....	349
15.5.3	Selecting Interrupt Request Destinations .....	350
15.5.4	Determining Priority.....	351
15.5.5	Fast Interrupt.....	352
15.5.6	Digital Filter .....	352
15.5.7	External Pin Interrupts.....	353
15.6	Non-maskable Interrupt Operation .....	353
15.7	Return from Power-Down States .....	354
15.7.1	Return from Sleep Mode .....	354
15.7.2	Return from All-Module Clock Stop Mode .....	354
15.7.3	Return from Software Standby Mode.....	355
15.8	Usage Note.....	355
15.8.1	Note on WAIT Instruction Used with Non-Maskable Interrupt.....	355

16.	Buses.....	356
16.1	Overview.....	356
16.2	Description of Buses.....	358
16.2.1	CPU Buses.....	358
16.2.2	Memory Buses.....	358
16.2.3	Internal Main Buses.....	359
16.2.4	Internal Peripheral Buses.....	359
16.2.5	Write Buffer Function (Internal Peripheral Bus).....	360
16.2.6	External Bus .....	361
16.2.7	Parallel Operation.....	363
16.2.8	Bus Settings .....	363
16.2.9	Restrictions .....	363
16.3	Register Descriptions.....	364
16.3.1	CSn Control Register (CSnCR) (n = 0 to 7).....	364
16.3.2	CSn Recovery Cycle Register (CSnREC) (n = 0 to 7).....	365
16.3.3	CS Recovery Cycle Insertion Enable Register (CSRECEN) .....	367
16.3.4	CSn Mode Register (CSnMOD) (n = 0 to 7).....	369
16.3.5	CSn Wait Control Register 1 (CSnWCR1) (n = 0 to 7) .....	371
16.3.6	CSn Wait Control Register 2 (CSnWCR2) (n = 0 to 7) .....	374
16.3.7	Bus Error Status Clear Register (BERCLR).....	377
16.3.8	Bus Error Monitoring Enable Register (BEREN) .....	377
16.3.9	Bus Error Status Register 1 (BERSR1).....	378
16.3.10	Bus Error Status Register 2 (BERSR2) .....	378
16.3.11	Bus Priority Control Register (BUSPRI).....	379
16.4	Endian and Data Alignment.....	381
16.4.1	Data Alignment Control for CS Area.....	381
16.5	Operation of CS Area Controller.....	387
16.5.1	Separate Bus .....	387
16.5.2	Address/Data Multiplexed Bus.....	402
16.5.3	External Wait Function.....	405
16.5.4	Insertion of Recovery Cycles .....	407
16.5.5	No Access State .....	410
16.5.6	Write Buffer Function (External Bus).....	411
16.5.7	Limitations.....	411
16.6	Bus Error Monitoring Section.....	413
16.6.1	Types of Bus Error .....	413
16.6.1.1	Illegal Address Access .....	413
16.6.1.2	Timeout .....	413
16.6.2	Operations When a Bus Error Occurs .....	413
16.6.3	Conditions Leading to Bus Errors .....	414



17.	Memory-Protection Unit (MPU)	415
17.1	Overview	415
17.1.1	Types of Access Control	417
17.1.2	Regions for Access Control	417
17.1.3	Background Region	417
17.1.4	Overlap between Regions	417
17.1.5	Instructions and Data that Span Regions	417
17.2	Register Descriptions	418
17.2.1	Region-n Start Page Number Register (RSPAGEn) (n = 0 to 7)	418
17.2.2	Region-n End Page Number Register (REPAGEn) (n = 0 to 7)	419
17.2.3	Memory-Protection Enable Register (MPEN)	420
17.2.4	Background Access Control Register (MPBAC)	421
17.2.5	Memory-Protection Error Status-Clearing Register (MPECLR)	422
17.2.6	Memory-Protection Error Status Register (MPESTS)	423
17.2.7	Data Memory-Protection Error Address Register (MPDEA)	424
17.2.8	Region Search Address Register (MPSA)	424
17.2.9	Region Search Operation Register (MPOPS)	425
17.2.10	Region Invalidation Operation Register (MPOPI)	425
17.2.11	Instruction-Hit Region Register (MHITI)	426
17.2.12	Data-Hit Region Register (MHITD)	427
17.3	Functions	429
17.3.1	Memory Protection	429
17.3.2	Region Search	429
17.3.3	Protection of Registers Related to the Memory-Protection Unit	429
17.3.4	Flow for Determination of Access by the Memory-Protection Function	430
17.4	Procedures for Using Memory Protection	432
17.4.1	Setting Access-Control Information	432
17.4.2	Enabling Memory Protection	432
17.4.3	Transition to User Mode	432
17.4.4	Processing in Response to Memory-Protection Errors	432
18.	DMA Controller (DMACA)	434
18.1	Overview	434
18.2	Register Descriptions	436
18.2.1	DMA Source Address Register (DMSAR)	436
18.2.2	DMA Destination Address Register (DMDAR)	436
18.2.3	DMA Transfer Count Register (DMCRA)	437
18.2.4	DMA Block Transfer Count Register (DMCRB)	438
18.2.5	DMA Transfer Mode Register (DMTMD)	439
18.2.6	DMA Interrupt Setting Register (DMINT)	440
18.2.7	DMA Address Mode Register (DMAMD)	442

18.2.8	DMA Offset Register (DMOFR).....	445
18.2.9	DMA Transfer Enable Register (DMCNT).....	445
18.2.10	DMA Software Start Register (DMREQ) .....	446
18.2.11	DMA Status Register (DMSTS).....	447
18.2.12	DMA Activation Source Flag Control Register (DMCSL).....	448
18.2.13	DMACA Module Activation Register (DMAST) .....	449
18.3	Operation.....	450
18.3.1	Transfer Mode .....	450
18.3.2	Extended Repeat Area Function .....	454
18.3.3	Address Update Function using Offset.....	456
18.3.4	Activation Sources.....	460
18.3.5	Operation Timing .....	461
18.3.6	DMAC Execution Cycles .....	462
18.3.7	Activating the DMAC .....	463
18.3.8	Starting DMA Transfer.....	464
18.3.9	Registers during DMA Transfer.....	464
18.3.10	Channel Priority.....	465
18.4	Ending DMA Transfer .....	466
18.4.1	Transfer End by Completion of Specified Total Number of Transfer Operations.....	466
18.4.2	Transfer End by Repeat Size End Interrupt.....	466
18.4.3	Transfer End by Interrupt on Extended Repeat Area Overflow .....	466
18.5	Interrupts.....	467
18.6	Low-Power Consumption Function.....	469
18.7	Usage Notes .....	470
18.7.1	DMA Transfer to External Devices.....	470
18.7.2	DMA Transfer to Peripheral Modules.....	470
18.7.3	Access to the Registers during DMA Transfer.....	470
18.7.4	DMA Transfer to Reserved Areas .....	470
18.7.5	Interrupt Request by the DMA Activation Source Flag Control Register (DMCSL) at the End of Each Transfer .....	470
18.7.6	Setting of DMAC Activation Request Select Register of the Interrupt Controller (ICU.DMRSRm).....	470
18.7.7	Suspending or Restarting DMA Activation.....	470
19.	Data Transfer Controller (DTCa).....	471
19.1	Overview.....	471
19.2	Register Descriptions.....	473
19.2.1	DTC Mode Register A (MRA).....	473
19.2.2	DTC Mode Register B (MRB) .....	474
19.2.3	DTC Transfer Source Register (SAR).....	475
19.2.4	DTC Transfer Destination Register (DAR).....	475
19.2.5	DTC Transfer Count Register A (CRA).....	476

19.2.6	DTC Transfer Count Register B (CRB) .....	477
19.2.7	DTC Control Register (DTCCR).....	477
19.2.8	DTC Vector Base Register (DTCVBR) .....	478
19.2.9	DTC Address Mode Register (DTCADM0D).....	478
19.2.10	DTC Module Start Register (DTCST).....	479
19.2.11	DTC Status Register (DTCSTS).....	480
19.3	Sources of Activation .....	481
19.3.1	Allocating Transfer Data and DTC Vector Table .....	481
19.4	Operation.....	483
19.4.1	Transfer Data Read Skip Function .....	485
19.4.2	Transfer Data Write-Back Skip Function.....	486
19.4.3	Normal Transfer Mode .....	486
19.4.4	Repeat Transfer Mode .....	487
19.4.5	Block Transfer Mode.....	489
19.4.6	Chain Transfer .....	490
19.4.7	Operation Timing .....	491
19.4.8	Execution Cycles of the DTC.....	494
19.4.9	DTC Bus Mastership Release Timing .....	494
19.5	DTC Setting Procedure.....	495
19.6	Examples of DTC Usage .....	496
19.6.1	Normal Transfer .....	496
19.6.2	Chain Transfer .....	496
19.6.3	Chain Transfer when Counter = 0 .....	498
19.7	Interrupt Source .....	499
19.8	Low-Power Consumption Function.....	500
19.9	Usage Notes .....	500
19.9.1	Transfer Data Start Address/Source Address/Destination Address .....	500
19.9.2	Allocating Transfer Data .....	500
19.9.3	Setting the DTC Activation Enable Register (ICU.DTCERn) of the Interrupt controller .....	501
20.	I/O Ports.....	502
20.1	Overview.....	502
20.2	I/O Port Configuration .....	505
20.3	Register Descriptions.....	509
20.3.1	Port Direction Register (PDR).....	509
20.3.2	Port Output Data Register (PODR) .....	510
20.3.3	Port Input Data Register (PIDR) .....	511
20.3.4	Port Mode Register (PMR).....	512
20.3.5	Open Drain Control Register 0 (ODR0).....	513
20.3.6	Open Drain Control Register 1 (ODR1).....	514
20.3.7	Pull-Up Control Register (PCR).....	515

20.3.8	Driving Ability Control Register (DSCR).....	516
20.4	Handling of Unused Pins .....	517
20.5	Usage Notes .....	517
20.5.1	Products with Fewer than 176 Pins .....	517
21.	Multi-Function Pin Controller (MPC) .....	518
21.1	Overview.....	518
21.2	Register Descriptions.....	533
21.2.1	Write-Protect Register (PWPR).....	533
21.2.2	P0n Pin Function Control Register (P0nPFS) (n = 0 to 3, 5, 7) .....	534
21.2.3	P1n Pin Function Control Registers (P1nPFS) (n = 0 to 7).....	535
21.2.4	P2n Pin Function Control Registers (P2nPFS) (n = 0 to 7).....	537
21.2.5	P3n Pin Function Control Registers (P3nPFS) (n = 0 to 4).....	539
21.2.6	P4n Pin Function Control Registers (P4nPFS) (n = 0 to 7).....	540
21.2.7	P5n Pin Function Control Registers (P5nPFS) (n = 0 to 2, 4 to 6).....	541
21.2.8	P6n Pin Function Control Registers (P6nPFS) (n = 0, 1, 6 or 7) .....	542
21.2.9	P7n Pin Function Control Registers (P7nPFS) (n = 0, 3 to 7).....	543
21.2.10	P8n Pin Function Control Registers (P8nPFS) (n = 0 to 3, 6, 7).....	544
21.2.11	P9n Pin Function Control Registers (P9nPFS) (n = 0 to 3).....	545
21.2.12	PAn Pin Function Select Registers (PAnPFS) (n = 0 to 7) .....	546
21.2.13	PBn Pin Function Control Registers (PBnPFS) (n = 0 to 7) .....	548
21.2.14	PCn Pin Function Control Register (PCnPFS) (n = 0 to 7).....	550
21.2.15	PDn Pin Function Control Register (PDnPFS) (n = 0 to 7) .....	552
21.2.16	PEn Pin Function Control Register (PEnPFS) (n = 0 to 7) .....	553
21.2.17	PFn Pin Function Select Register (PFnPFS) (n = 0 to 2, 5) .....	555
21.2.18	PJ3 Pin Function Control Register (PJ3PFS) .....	556
21.2.19	PKn Pin Function Control Register (PKnPFS) (n = 2 to 5) .....	557
21.2.20	CS Output Enable Register (PFCSE) .....	557
21.2.21	CS Output Pin Select Register 0 (PFCSS0).....	558
21.2.22	CS Output Pin Select Register 1 (PFCSS1).....	559
21.2.23	Address Output Enable Register 0 (PFAOE0) .....	560
21.2.24	Address Output Enable Register 1 (PFAOE1) .....	560
21.2.25	External Bus Control Register 0 (PFBCR0).....	561
21.2.26	External Bus Control Register 1 (PFBCR1).....	561
21.2.27	USB0 Control Register (PFUSB0).....	562
21.3	How to Set the External Bus Interface.....	563
21.4	Usage Notes .....	566
21.4.1	Procedure for Specifying Input/Output Pin Function.....	566
21.4.2	Notes on MPC Register Setting.....	566
21.4.3	Notes on the Use of Analog Functions.....	567

22.	Multi-Function Timer Pulse Unit 2 (MTU2a).....	568
22.1	Overview.....	568
22.2	Register Descriptions.....	573
22.2.1	Timer Control Register (TCR) .....	573
22.2.2	Timer Mode Register (TMDR).....	576
22.2.3	Timer I/O Control Register (TIOR).....	578
22.2.4	Timer Compare Match Clear Register (TCNTCMPCLR) .....	589
22.2.5	Timer Interrupt Enable Register (TIER) .....	590
22.2.6	Timer Status Register (TSR) .....	592
22.2.7	Timer Buffer Operation Transfer Mode Register (TBTM).....	593
22.2.8	Timer Input Capture Control Register (TICCR) .....	594
22.2.9	Timer A/D Converter Start Request Control Register (TADCR) .....	595
22.2.10	Timer A/D Converter Start Request Cycle Set Registers A and B (TADCORA and TADCORB) .....	596
22.2.11	Timer A/D Converter Start Request Cycle Set Buffer Registers A and B (TADCOBRA and TADCOBRB).....	596
22.2.12	Timer Counter (TCNT) .....	597
22.2.13	Timer General Register (TGR).....	597
22.2.14	Timer Start Registers (TSTR).....	598
22.2.15	Timer Synchronous Registers (TSYR).....	599
22.2.16	Timer Read/Write Enable Registers (TRWER) .....	600
22.2.17	Timer Output Master Enable Registers (TOER) .....	601
22.2.18	Timer Output Control Registers 1 (TOCR1).....	602
22.2.19	Timer Output Control Registers 2 (TOCR2).....	604
22.2.20	Timer Output Level Buffer Registers (TOLBR) .....	606
22.2.21	Timer Gate Control Registers (TGCR) .....	607
22.2.22	Timer Subcounters (TCNTS) .....	608
22.2.23	Timer Dead Time Data Registers (TDDR).....	608
22.2.24	Timer Cycle Data Registers (TCDR) .....	609
22.2.25	Timer Cycle Buffer Registers (TCBR).....	609
22.2.26	Timer Interrupt Skipping Set Registers (TITCR).....	609
22.2.27	Timer Interrupt Skipping Counters (TITCNT).....	611
22.2.28	Timer Buffer Transfer Set Registers (TBTER) .....	612
22.2.29	Timer Dead Time Enable Registers (TDER) .....	613
22.2.30	Timer Waveform Control Registers (TWCR).....	614
22.2.31	Noise Filter Control Registers (NFCR).....	615
22.2.32	Bus Master Interface.....	617
22.3	Operation.....	617
22.3.1	Basic Functions.....	617
22.3.2	Synchronous Operation .....	622
22.3.3	Buffer Operation.....	624

22.3.4	Cascaded Operation.....	627
22.3.5	PWM Modes.....	632
22.3.6	Phase Counting Mode.....	636
22.3.7	Reset-Synchronized PWM Mode.....	642
22.3.8	Complementary PWM Mode.....	645
22.3.9	A/D Converter Start Request Delaying Function.....	673
22.3.10	External Pulse Width Measurement.....	676
22.3.11	Dead Time Compensation.....	677
22.3.12	Noise Filter.....	679
22.4	Interrupt Sources.....	680
22.4.1	Interrupt Sources and Priorities.....	680
22.4.2	DTC and DMAC Activation.....	681
22.4.3	A/D Converter Activation.....	681
22.5	Operation Timing.....	683
22.5.1	Input/Output Timing.....	683
22.5.2	Interrupt Signal Timing.....	688
22.6	Usage Notes.....	691
22.6.1	Module Clock Stop Mode Setting.....	691
22.6.2	Input Clock Restrictions.....	691
22.6.3	Note on Cycle Setting.....	691
22.6.4	Contention between TCNT Write and Clear Operations.....	692
22.6.5	Contention between TCNT Write and Increment Operations.....	692
22.6.6	Contention between TGR Write Operation and Compare Match.....	693
22.6.7	Contention between Buffer Register Write Operation and Compare Match.....	693
22.6.8	Contention between Buffer Register Write and TCNT Clear Operations.....	694
22.6.9	Contention between TGR Read Operation and Input Capture.....	694
22.6.10	Contention between TGR Write Operation and Input Capture.....	695
22.6.11	Contention between Buffer Register Write Operation and Input Capture.....	696
22.6.12	Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation.....	696
22.6.13	Counter Value when Stopped in Complementary PWM Mode.....	697
22.6.14	Buffer Operation Setting in Complementary PWM Mode.....	698
22.6.15	Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode.....	698
22.6.16	Overflow Flags in Reset-Synchronized PWM Mode.....	699
22.6.17	Contention between Overflow/Underflow and Counter Clearing.....	700
22.6.18	Contention between TCNT Write Operation and Overflow/Underflow.....	700
22.6.19	Note on Transition from Normal Mode or PWM Mode 1 to Reset-Synchronized PWM Mode.....	700
22.6.20	Output Level in Complementary PWM Mode or Reset-Synchronized PWM Mode.....	701
22.6.21	Interrupts During Periods in the Module-Stop State.....	701
22.6.22	Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection.....	701

22.6.23	Notes when Complementary PWM Mode Output Protection Functions are not Used .....	701
22.6.24	Point for Caution Regarding MTU5.TCNT and MTU5.TGR Registers .....	701
22.6.25	Points for Caution to Prevent Malfunctions in Synchronous Clearing for Complementary PWM Mode.....	702
22.6.26	Continuous Output of Interrupt Signal in Response to a Compare Match.....	704
22.7	MTU Output Pin Initialization .....	705
22.7.1	Operating Modes .....	705
22.7.2	Operation in Case of Re-Setting Due to Error during Operation .....	705
22.7.3	Overview of Pin Initialization Procedures and Mode Transitions in Case of Error during Operation .....	706
23.	Port Output Enable 2 (POE2a) .....	732
23.1	Overview.....	732
23.2	Register Descriptions.....	735
23.2.1	Input Level Control/Status Register 1 (ICSR1).....	735
23.2.2	Output Level Control/Status Register 1 (OCSR1) .....	737
23.2.3	Input Level Control/Status Register 2 (ICSR2).....	738
23.2.4	Software Port Output Enable Register (SPOER) .....	739
23.2.5	Port Output Enable Control Register 1 (POECR1) .....	740
23.2.6	Port Output Enable Control Register 2 (POECR2) .....	740
23.2.7	Input Level Control/Status Register 3 (ICSR3).....	741
23.3	Operation.....	742
23.3.1	Input Level Detection Operation .....	744
23.3.2	Output-Level Compare Operation .....	745
23.3.3	High-Impedance Control Using Registers.....	745
23.3.4	High-Impedance Control on Detection of Stopped Oscillation.....	745
23.3.5	Release from the High-Impedance .....	745
23.4	Interrupts.....	746
23.5	Usage Notes .....	746
23.5.1	Transitions to Software Standby Mode or Deep Standby Mode .....	746
23.5.2	When POE is not Used .....	746
23.5.3	Specifying Pins corresponding to the MTU .....	746
24.	16-Bit Timer Pulse Unit (TPUa) .....	747
24.1	Overview.....	747
24.2	Register Descriptions.....	753
24.2.1	Timer Control Register (TCR) .....	753
24.2.2	Timer Mode Register (TMDR).....	758
24.2.3	Timer I/O Control Register (TIORH, TIORL, TIOR).....	759
24.2.4	Timer Interrupt Enable Register (TIER) .....	769
24.2.5	Timer Status Register (TSR) .....	770
24.2.6	Timer Counter (TCNT) .....	772

24.2.7	Timer General Register A (TGRA) Timer General Register B (TGRB) Timer General Register C (TGRC) Timer General Register D (TGRD).....	773
24.2.8	Timer Start Register (TSTR).....	774
24.2.9	Timer Synchronous Register (TSYR).....	775
24.2.10	Noise Filter Control Register (NFCR).....	776
<b>24.3</b>	<b>Operation.....</b>	<b>778</b>
24.3.1	Basic Functions.....	778
24.3.2	Synchronous Operation.....	783
24.3.3	Buffer Operation.....	785
24.3.4	Cascaded Operation.....	788
24.3.5	PWM Modes.....	790
24.3.6	Phase Counting Mode.....	795
24.3.6.1	Phase Counting Mode Application Example.....	800
24.3.7	Noise Filters.....	801
24.4	Interrupt Sources.....	802
24.5	DTC Activation.....	804
24.6	DMAC Activation.....	804
24.7	A/D Converter Activation.....	804
24.8	PPG Trigger.....	804
24.9	Operation Timing.....	805
24.9.1	Input/Output Timing.....	805
24.9.2	Interrupt Signal Timing.....	809
24.10	Usage Notes.....	811
24.10.1	Module-Stop Function Setting.....	811
24.10.2	Input Clock Restrictions.....	811
24.10.3	Caution on Cycle Setting.....	812
24.10.4	Conflict between TPUM.TCNT Write and Clear Operations.....	812
24.10.5	Conflict between TPUM.TCNT Write and Increment Operations.....	812
24.10.6	Conflict between TPUM.TGRy Write and Compare Match.....	813
24.10.7	Conflict between Buffer Register Write and Compare Match.....	813
24.10.8	Conflict between TPUM.TGRy Read and Input Capture.....	814
24.10.9	Conflict between TPUM.TGRy Write and Input Capture.....	814
24.10.10	Conflict between Buffer Register Write and Input Capture.....	815
24.10.11	Conflict between Overflow/Underflow and Counter Clearing.....	815
24.10.12	Conflict between TPUM.TCNT Write and Overflow/Underflow.....	816
24.10.13	Multiplexing of I/O Pins.....	816
24.10.14	Continuous Output of Compare-Match Pulse Interrupt Signal.....	817
24.10.15	Continuous Output of Input-Capture Pulse Interrupt Signal.....	817
24.10.16	Continuous Output of Underflow Pulse Interrupt Signal.....	818



25.	Programmable Pulse Generator (PPG)	819
25.1	Overview	819
25.2	Register Descriptions	822
25.2.1	PPG Trigger Select Register (PTRSLR)	822
25.2.2	Next Data Enable Registers H (NDERH) Next Data Enable Registers L (NDERL)	823
25.2.3	Output Data Registers H (PODRH) Output Data Registers L (PODRL)	825
25.2.4	Next Data Registers H (NDRH) Next Data Registers L (NDRL)	828
25.2.5	PPG Output Control Register (PCR)	832
25.2.6	PPG Output Mode Register (PMR)	834
25.3	Operation	837
25.3.1	Output Timing	838
25.3.2	Sample Setup Procedure for Normal Pulse Output	839
25.3.3	Example of Normal Pulse Output (Example of Five-Phase Pulse Output)	841
25.3.4	Non-Overlapping Pulse Output	842
25.3.5	Sample Setup Procedure for Non-Overlapping Pulse Output	843
25.3.6	Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output)	845
25.3.7	Inverted Pulse Output	847
25.3.8	Pulse Output Triggered by Input Capture	848
25.4	Usage Note	848
25.4.1	Module-Stop Function Setting	848
26.	8-Bit Timer (TMR)	849
26.1	Overview	849
26.2	Register Descriptions	853
26.2.1	Timer Counter (TCNT)	854
26.2.2	Time Constant Register A (TCORA)	854
26.2.3	Time Constant Register B (TCORB)	855
26.2.4	Timer Control Register (TCR)	855
26.2.5	Timer Counter Control Register (TCCR)	856
26.2.6	Timer Control/Status Register (TCSR)	858
26.3	Operation	860
26.3.1	Pulse Output	860
26.3.2	Reset Input	860
26.4	Operation Timing	861
26.4.1	TCNT Count Timing	861
26.4.2	Timing of Interrupt Flag Signal Output on a Compare Match	862
26.4.3	Timing of Timer Output at Compare Match	863
26.4.4	Timing of Counter Clear by Compare Match	863

26.4.5	Timing of the External Reset for TCNT.....	864
26.4.6	Timing of Interrupt Signal Output on an Overflow.....	865
26.5	Operation with Cascaded Connection .....	865
26.5.1	16-Bit Count Mode.....	865
26.5.2	Compare Match Count Mode .....	865
26.6	Interrupt Sources .....	866
26.6.1	Interrupt Sources and DTC Activation.....	866
26.6.2	A/D Converter Activation .....	866
26.7	Usage Notes.....	867
26.7.1	Module-Stop State Setting.....	867
26.7.2	Notes on Setting Cycle .....	867
26.7.3	Conflict between TCNT Write and Counter Clear.....	867
26.7.4	Conflict between TCNT Write and Increment .....	868
26.7.5	Conflict between TCORA or TCORB Write and Compare Match.....	868
26.7.6	Conflict between Compare Matches A and B .....	869
26.7.7	Switching of Frequency Dividing Clocks and TCNT Operation .....	869
26.7.8	Clock Source Setting with Cascaded Connection .....	871
26.7.9	Continuous Output of Compare Match Interrupt Signal .....	871
27.	Compare Match Timer (CMT) .....	872
27.1	Overview.....	872
27.2	Register Descriptions.....	873
27.2.1	Compare Match Timer Start Register 0 (CMSTR0) .....	873
27.2.2	Compare Match Timer Start Register 1 (CMSTR1) .....	873
27.2.3	Compare Match Timer Control Register (CMCR).....	874
27.2.4	Compare Match Timer Counter (CMCNT).....	875
27.2.5	Compare Match Timer Constant Register (CMCOR).....	875
27.3	Operation.....	876
27.3.1	Periodic Count Operation .....	876
27.3.2	CMCNT Count Timing .....	876
27.4	Interrupts.....	877
27.4.1	Interrupt Sources.....	877
27.4.2	Timing of Compare Match Interrupt Generation .....	877
27.5	Usage Notes .....	878
27.5.1	Setting the Module-Stop Function.....	878
27.5.2	Conflict between Write and Compare-Match Processes of CMCNT.....	878
27.5.3	Conflict between Write and Count-Up Processes of CMCNT.....	878
28.	Realtime Clock (RTCa) .....	879
28.1	Overview.....	879
28.2	Register Descriptions.....	881
28.2.1	64-Hz Counter (R64CNT) .....	881

28.2.2	Second Counter (RSECCNT).....	882
28.2.3	Minute Counter (RMINCNT).....	882
28.2.4	Hour Counter (RHRCNT).....	883
28.2.5	Day-of-Week Counter (RWKCNT).....	884
28.2.6	Date Counter (RDAYCNT).....	885
28.2.7	Month Counter (RMONCNT).....	885
28.2.8	Year Counter (RYRCNT).....	886
28.2.9	Second Alarm Register (RSECAR).....	886
28.2.10	Minute Alarm Register (RMINAR).....	887
28.2.11	Hour Alarm Register (RHRAR).....	888
28.2.12	Day-of-Week Alarm Register (RWKAR).....	889
28.2.13	Date Alarm Register (RDAYAR).....	890
28.2.14	Month Alarm Register (RMONAR).....	891
28.2.15	Year Alarm Register (RYRAR).....	892
28.2.16	Year Alarm Enable Register (RYRAREN).....	892
28.2.17	RTC Control Register 1 (RCR1).....	893
28.2.18	RTC Control Register 2 (RCR2).....	894
28.2.19	RTC Control Register 3 (RCR3).....	896
28.2.20	RTC Control Register 4 (RCR4).....	897
28.2.21	Frequency Register H/L (RFRH/L).....	898
28.2.22	Time Error Adjustment Register (RADJ).....	899
28.2.23	Time Capture Control Register y (RTCCRy) (y = 0 to 2).....	900
28.2.24	Second Capture Register y (RSECCPy) (y = 0 to 2).....	901
28.2.25	Minute Capture Register y (RMINCPy) (y = 0 to 2).....	902
28.2.26	Hour Capture Register y (RHRCPy) (y = 0 to 2).....	902
28.2.27	Date Capture Register y (RDAYCPy) (y = 0 to 2).....	903
28.2.28	Month Capture Register y (RMONCPy) (y = 0 to 2).....	903
28.3	Operation.....	904
28.3.1	Outline of Initial Settings of Registers after Power-On.....	904
28.3.2	Clock Setting Procedure.....	905
28.3.3	Setting the Time.....	906
28.3.4	30-Second Adjustment.....	907
28.3.5	Reading 64-Hz Counter and Time.....	908
28.3.6	Alarm Function.....	909
28.3.7	Procedure for Disabling Alarm Interrupt.....	910
28.3.8	Time-Error Adjustment Function.....	910
28.3.8.1	Automatic Adjustment.....	910
28.3.8.2	Adjustment by Software.....	911
28.3.8.3	Procedure for Changing the Mode of Adjustment.....	911
28.3.8.4	Procedure for Stopping Adjustment.....	911

28.3.8.5	Capturing the Time .....	912
28.4	Interrupt Sources .....	913
28.5	Usage Notes .....	914
28.5.1	Register Writing during Counting .....	914
28.5.2	Use of Periodic Interrupts .....	914
28.5.3	RTCOUT (1-Hz) Output .....	915
28.5.4	Transitions to Low Power Consumption Modes after Setting Registers.....	915
28.5.5	Points for Caution When Writing to and Reading from Registers .....	915
28.5.6	Initialization Procedure when the Realtime Clock is not to be Used .....	916
29.	Watchdog Timer (WDTA) .....	917
29.1	Overview.....	917
29.2	Register Descriptions.....	919
29.2.1	WDT Refresh Register (WDTRR) .....	919
29.2.2	WDT Control Register (WDTCR) .....	920
29.2.3	WDT Status Register (WDTSR) .....	923
29.2.4	WDT Reset Control Register (WDTRCR).....	924
29.2.5	Option Function Select Register 0 (OFS0).....	924
29.3	Operation.....	925
29.3.1	Count Operation in Each Start Mode .....	925
29.3.1.1	Register Start Mode .....	925
29.3.1.2	Auto-Start Mode .....	926
29.3.2	Control over Writing to the WDTCR and WDTRCR Registers .....	927
29.3.3	Refresh Operation.....	928
29.3.4	Status Flags.....	929
29.3.5	Reset Output .....	930
29.3.6	Interrupt Source .....	930
29.3.7	Reading the Down-Counter Value .....	930
29.3.8	Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.....	931
30.	Independent Watchdog Timer (IWDTa).....	932
30.1	Overview.....	932
30.2	Register Descriptions.....	934
30.2.1	IWDT Refresh Register (IWDTRR).....	934
30.2.2	IWDT Control Register (IWDTCR).....	935
30.2.3	IWDT Status Register (IWDTSR).....	938
30.2.4	IWDT Reset Control Register (IWDTRCR) .....	939
30.2.5	IWDT Count Stop Control Register (IWDTCSTPR).....	939
30.2.6	Option Function Select Register 0 (OFS0).....	940
30.3	Operation.....	940
30.3.1	Count Operation in Each Start Mode .....	940
30.3.1.1	Register Start Mode .....	940

30.3.1.2	Auto-Start Mode .....	941
30.3.2	Control over Writing to the IWDTCR, IWDTRCR, and IWDTCS0 Registers .....	942
30.3.3	Refresh Operation.....	943
30.3.4	Status Flags.....	945
30.3.5	Reset Output .....	945
30.3.6	Interrupt Source .....	946
30.3.7	Reading the Down-Counter Value .....	946
30.3.8	Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers ....	947
30.4	Usage Notes.....	947
30.4.1	Refresh Operations .....	947
31.	USB 2.0 Function Module (USBa) .....	948
31.1	Overview.....	948
31.2	Register Descriptions.....	950
31.2.1	System Configuration Control Register (SYSCFG).....	950
31.2.2	System Configuration Status Register 0 (SYSSTS0).....	951
31.2.3	Device State Control Register 0 (DVSTCTR0) .....	952
31.2.4	CFIFO Port Register (CFIFO) D0FIFO Port Register (D0FIFO) D1FIFO Port Register (D1FIFO) .....	953
31.2.5	CFIFO Port Select Register (CFIFOSEL) D0FIFO Port Select Register (D0FIFOSEL) D1FIFO Port Select Register (D1FIFOSEL).....	955
31.2.6	CFIFO Port Control Register (CFIFOCTR) D0FIFO Port Control Register (D0FIFOCTR) D1FIFO Port Control Register (D1FIFOCTR) .....	959
31.2.7	Interrupt Enable Register 0 (INTENB0) .....	961
31.2.8	BRDY Interrupt Enable Register (BRDYENB).....	962
31.2.9	NRDY Interrupt Enable Register (NRDYENB) .....	963
31.2.10	BEMP Interrupt Enable Register (BEMPENB) .....	964
31.2.11	SOF Output Configuration Register (SOFCFG) .....	965
31.2.12	Interrupt Status Register 0 (INTSTS0).....	965
31.2.13	BRDY Interrupt Status Register (BRDYSTS) .....	968
31.2.14	NRDY Interrupt Status Register (NRDYSTS).....	969
31.2.15	BEMP Interrupt Status Register (BEMPSTS).....	970
31.2.16	Frame Number Register (FRMNUM) .....	971
31.2.17	Device State Changing Register (DVCHGR) .....	972
31.2.18	USB Address Register (USBADDR) .....	972
31.2.19	USB Request Type Register (USBREQ).....	973
31.2.20	USB Request Value Register (USBVAL).....	973
31.2.21	USB Request Index Register (USBINDX).....	974
31.2.22	USB Request Length Register (USBLENG).....	974
31.2.23	DCP Maximum Packet Size Register (DCPMAXP).....	975

31.2.24	DCP Control Register (DCPCTR).....	976
31.2.25	Pipe Window Select Register (PIPESEL) .....	978
31.2.26	Pipe Configuration Register (PIPECFG).....	979
31.2.27	Pipe Maximum Packet Size Register (PIPEMAXP) .....	981
31.2.28	Pipe Cycle Control Register (PIPEPERI).....	982
31.2.29	PIPEn Control Registers (PIPEnCTR) (n = 1 to 9) .....	983
31.2.30	PIPEn Transaction Counter Enable Registers (PIPEnTRE) (n = 1 to 5).....	989
31.2.31	PIPEn Transaction Counter Registers (PIPEnTRN) (n = 1 to 5) .....	990
31.2.32	Deep Standby USB Transceiver Control/Pin Monitor Register (DPUSR0R).....	991
31.2.33	Deep Standby USB Suspend/Resume Interrupt Register (DPUSR1R).....	992
<b>31.3</b>	<b>Operation</b> .....	<b>993</b>
31.3.1	System Control .....	993
31.3.1.1	Starting Operation .....	993
31.3.1.2	Canceling Deep Software Standby Mode by a USB Suspend/Resume Interrupt .....	995
31.3.2	Interrupt Sources.....	998
31.3.3	Interrupt Descriptions .....	1000
31.3.3.1	BRDY Interrupt .....	1000
31.3.3.2	NRDY Interrupt .....	1003
31.3.3.3	BEMP Interrupt .....	1005
31.3.3.4	Device State Transition Interrupt .....	1007
31.3.3.5	Control Transfer Stage Transition Interrupt .....	1008
31.3.3.6	Frame Update Interrupt .....	1009
31.3.3.7	VBUS Interrupt .....	1009
31.3.3.8	Resume Interrupt .....	1009
31.3.4	Pipe Control.....	1009
31.3.4.1	Pipe Control Register Switching Procedures .....	1010
31.3.4.2	Transfer Types .....	1010
31.3.4.3	Endpoint Number .....	1011
31.3.4.4	Maximum Packet Size Setting .....	1011
31.3.4.5	Transaction Counter (For PIPE1 to PIPE5 in Reading Direction) .....	1011
31.3.4.6	Response PID .....	1011
31.3.4.7	Data PID Sequence Bit .....	1012
31.3.4.8	Response PID = NAK Function .....	1012
31.3.4.9	Auto Response Mode .....	1012
31.3.4.10	OUT-NAK Mode .....	1012
31.3.4.11	Null Auto Response Mode .....	1013
31.3.5	FIFO Buffer Memory .....	1014
31.3.5.1	FIFO Buffer Memory .....	1014
31.3.5.2	FIFO Buffer Clearing .....	1014
31.3.5.3	FIFO Port Functions .....	1015

31.3.5.4	DMA Transfers (D0FIFO and D1FIFO Ports) .....	1016
31.3.6	Control Transfers (DCP) .....	1017
31.3.6.1	Control Transfers .....	1017
31.3.7	Bulk Transfers (PIPE1 to PIPE5) .....	1018
31.3.8	Interrupt Transfers (PIPE6 to PIPE9) .....	1018
31.3.9	Isochronous Transfers (PIPE1 and PIPE2) .....	1019
31.3.9.1	Error Detection in Isochronous Transfers .....	1019
31.3.9.2	DATA-PID .....	1020
31.3.9.3	Interval Counter .....	1020
31.3.10	SOF Interpolation Function .....	1025
31.4	Usage Notes .....	1025
31.4.1	Settings for the Module-Stop State .....	1025
32.	Serial Communications Interface (SC1c, SC1d) .....	1026
32.1	Overview .....	1026
32.2	Register Descriptions .....	1035
32.2.1	Receive Shift Register (RSR) .....	1035
32.2.2	Receive Data Register (RDR) .....	1035
32.2.3	Transmit Data Register (TDR) .....	1035
32.2.4	Transmit Shift Register (TSR) .....	1036
32.2.5	Serial Mode Register (SMR) .....	1036
32.2.6	Serial Control Register (SCR) .....	1039
32.2.7	Serial Status Register (SSR) .....	1043
32.2.8	Smart Card Mode Register (SCMR) .....	1047
32.2.9	Bit Rate Register (BRR) .....	1048
32.2.10	Serial Extended Mode Register (SEMR) .....	1056
32.2.11	Noise Filter Setting Register (SNFR) .....	1058
32.2.12	I <sup>2</sup> C Mode Register 1 (SIMR1) .....	1059
32.2.13	I <sup>2</sup> C Mode Register 2 (SIMR2) .....	1060
32.2.14	I <sup>2</sup> C Mode Register 3 (SIMR3) .....	1061
32.2.15	I <sup>2</sup> C Status Register (SISR) .....	1063
32.2.16	SPI Mode Register (SPMR) .....	1064
32.2.17	Extended Serial Module Enable Register (ESMER) .....	1065
32.2.18	Control Register 0 (CR0) .....	1066
32.2.19	Control Register 1 (CR1) .....	1067
32.2.20	Control Register 2 (CR2) .....	1068
32.2.21	Control Register 3 (CR3) .....	1069
32.2.22	Port Control Register (PCR) .....	1069
32.2.23	Interrupt Control Register (ICR) .....	1070
32.2.24	Status Register (STR) .....	1071
32.2.25	Status Clear Register (STCR) .....	1072

32.2.26	Control Field 0 Data Register (CF0DR).....	1072
32.2.27	Control Field 0 Compare Enable Register (CF0CR).....	1073
32.2.28	Control Field 0 Receive Data Register (CF0RR).....	1073
32.2.29	Primary Control Field 1 Data Register (PCF1DR).....	1073
32.2.30	Secondary Control Field 1 Data Register (SCF1DR).....	1074
32.2.31	Control Field 1 Compare Enable Register (CF1CR).....	1074
32.2.32	Control Field 1 Receive Data Register (CF1RR).....	1074
32.2.33	Timer Control Register (TCR) .....	1075
32.2.34	Timer Mode Register (TMR) .....	1075
32.2.35	Timer Prescaler Register (TPRE).....	1076
32.2.36	Timer Count Register (TCNT) .....	1076
<b>32.3</b>	<b>Operation in Asynchronous Mode .....</b>	<b>1077</b>
32.3.1	Serial Data Transfer Format .....	1077
32.3.2	Receive Data Sampling Timing and Reception Margin in Asynchronous Mode .....	1079
32.3.3	Clock.....	1080
32.3.4	CTS and RTS Functions .....	1080
32.3.5	SCI Initialization (Asynchronous Mode) .....	1081
32.3.6	Serial Data Transmission (Asynchronous Mode).....	1082
32.3.7	Serial Data Reception (Asynchronous Mode).....	1084
<b>32.4</b>	<b>Multi-Processor Communications Function .....</b>	<b>1088</b>
32.4.1	Multi-Processor Serial Data Transmission.....	1089
32.4.2	Multi-Processor Serial Data Reception .....	1090
<b>32.5</b>	<b>Operation in Clock Synchronous Mode .....</b>	<b>1093</b>
32.5.1	Clock.....	1093
32.5.2	CTS and RTS Functions.....	1093
32.5.3	SCI Initialization (Clock Synchronous Mode).....	1094
32.5.4	Serial Data Transmission (Clock Synchronous Mode) .....	1095
32.5.5	Serial Data Reception (Clock Synchronous Mode).....	1096
32.5.6	Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode).....	1099
<b>32.6</b>	<b>Operation in Smart Card Interface Mode.....</b>	<b>1100</b>
32.6.1	Sample Connection.....	1100
32.6.2	Data Format (Except in Block Transfer Mode).....	1100
32.6.3	Block Transfer Mode.....	1102
32.6.4	Receive Data Sampling Timing and Reception Margin.....	1102
32.6.5	Initialization of the SCI (Smart Card Interface Mode).....	1103
32.6.6	Serial Data Transmission (Except in Block Transfer Mode) .....	1104
32.6.7	Serial Data Reception (Except in Block Transfer Mode).....	1107
32.6.8	Clock Output Control .....	1108
<b>32.7</b>	<b>Operation in Simple I<sup>2</sup>C Mode .....</b>	<b>1110</b>
32.7.1	Generation of Start, Restart, and Stop Conditions.....	1111



32.7.2	Clock Synchronization .....	1112
32.7.3	SSDA Output Delay .....	1113
32.7.4	SCI Initialization (Simple I <sup>2</sup> C Mode).....	1114
32.7.5	Operation in Master Transmission (Simple I <sup>2</sup> C Mode) .....	1115
32.7.6	Master Reception (Simple I <sup>2</sup> C Mode).....	1117
32.8	Operation in Simple SPI Mode .....	1119
32.8.1	States of Pins in Master and Slave Modes.....	1120
32.8.2	SS Function in Master Mode .....	1120
32.8.3	SS Function in Slave Mode .....	1120
32.8.4	Relationship between Clock and Transmit/Receive Data .....	1121
32.8.5	SCI Initialization (Simple SPI Mode) .....	1121
32.8.6	Transmission and Reception of Serial Data (Simple SPI Mode) .....	1122
32.9	Extended Serial Mode Control Section: Description of Operation .....	1122
32.9.1	Serial Transfer Protocol.....	1122
32.9.2	Transmitting a Start Frame .....	1123
32.9.3	Receiving a Start Frame .....	1126
32.9.3.1	Priority Interrupt Bit .....	1131
32.9.4	Detection of Bus Collisions.....	1132
32.9.5	Digital Filter for Input on the RXDX Pin.....	1133
32.9.6	Bit-Rate Measurement.....	1134
32.9.7	Selectable Timing for Sampling Data Received through RXDX12.....	1135
32.9.8	Timer .....	1135
32.10	Noise Cancellation Function.....	1137
32.11	Interrupt Sources .....	1138
32.11.1	Buffer Operations for TXI and RXI Interrupts.....	1138
32.11.2	Interrupts in Serial Communications Interface and Simple SPI Mode .....	1138
32.11.3	Interrupts in Smart Card Interface Mode.....	1139
32.11.4	Interrupts in Simple I <sup>2</sup> C Mode .....	1140
32.11.5	Interrupts from the Extended Serial Mode Control Section.....	1141
32.12	Usage Notes.....	1141
32.12.1	Setting the Module-Stop Function.....	1141
32.12.2	Break Detection and Processing.....	1141
32.12.3	The Mark State and Production of Breaks.....	1141
32.12.4	Receive Error Flags and Transmit Operations (Clock Synchronous Mode Only).....	1142
32.12.5	Writing Data to TDR.....	1142
32.12.6	Restrictions on Clock Synchronous Transmission .....	1142
32.12.7	Restrictions on Using DTC or DMAC .....	1142
32.12.8	Points to Note on Starting Transfer .....	1142
32.12.9	SCI Operations during Low Power Consumption State.....	1142
32.12.10	External Clock Input in Clock Synchronous Mode.....	1145

32.12.11	Limitations on Simple SPI Mode .....	1145
32.12.12	Limitation 1 on Usage of the Extended Serial Mode Control Section .....	1145
32.12.13	Limitation 2 on Usage of the Extended Serial Mode Control Section .....	1146
33.	I <sup>2</sup> C Bus Interface (RIIC) .....	1147
33.1	Overview .....	1147
33.2	Register Descriptions .....	1150
33.2.1	I <sup>2</sup> C Bus Control Register 1 (ICCR1) .....	1150
33.2.2	I <sup>2</sup> C Bus Control Register 2 (ICCR2) .....	1152
33.2.3	I <sup>2</sup> C Bus Mode Register 1 (ICMR1) .....	1155
33.2.4	I <sup>2</sup> C Bus Mode Register 2 (ICMR2) .....	1156
33.2.5	I <sup>2</sup> C Bus Mode Register 3 (ICMR3) .....	1158
33.2.6	I <sup>2</sup> C Bus Function Enable Register (ICFER) .....	1160
33.2.7	I <sup>2</sup> C Bus Status Enable Register (ICSER) .....	1162
33.2.8	I <sup>2</sup> C Bus Interrupt Enable Register (ICIER) .....	1163
33.2.9	I <sup>2</sup> C Bus Status Register 1 (ICSR1) .....	1164
33.2.10	I <sup>2</sup> C Bus Status Register 2 (ICSR2) .....	1167
33.2.11	Slave Address Register Ly (SARLy) (y = 0 to 2) .....	1170
33.2.12	Slave Address Register Uy (SARUy) (y = 0 to 2) .....	1171
33.2.13	I <sup>2</sup> C Bus Bit Rate Low-Level Register (ICBRL) .....	1172
33.2.14	I <sup>2</sup> C Bus Bit Rate High-Level Register (ICBRH) .....	1173
33.2.15	I <sup>2</sup> C Bus Transmit Data Register (ICDRT) .....	1175
33.2.16	I <sup>2</sup> C Bus Receive Data Register (ICDRR) .....	1175
33.2.17	I <sup>2</sup> C Bus Shift Register (ICDRS) .....	1175
33.3	Operation .....	1176
33.3.1	Communication Data Format .....	1176
33.3.2	Initial Settings .....	1177
33.3.3	Master Transmit Operation .....	1178
33.3.4	Master Receive Operation .....	1181
33.3.5	Slave Transmit Operation .....	1185
33.3.6	Slave Receive Operation .....	1188
33.4	SCL Synchronization Circuit .....	1190
33.5	Facility for Delaying SDA Output .....	1191
33.6	Digital Noise-Filter Circuits .....	1192
33.7	Address Match Detection .....	1193
33.7.1	Slave-Address Match Detection .....	1193
33.7.2	Detection of the General Call Address .....	1195
33.7.3	Device-ID Address Detection .....	1195
33.7.4	Host Address Detection .....	1196
33.8	Automatically Low-Hold Function for SCL .....	1197
33.8.1	Function to Prevent Wrong Transmission of Transmit Data .....	1197

33.8.2	NACK Reception Transfer Suspension Function.....	1198
33.8.3	Function to Prevent Failure to Receive Data.....	1199
33.9	Arbitration-Lost Detection Functions.....	1201
33.9.1	Master Arbitration-Lost Detection (MALE Bit) .....	1201
33.9.2	Function to Detect Loss of Arbitration during NACK Transmission (NALE Bit) .....	1203
33.9.3	Slave Arbitration-Lost Detection (SALE Bit).....	1204
33.10	Start Condition/Restart Condition/Stop Condition Issuing Function .....	1205
33.10.1	Issuing a Start Condition .....	1205
33.10.2	Issuing a Restart Condition.....	1205
33.10.3	Issuing a Stop Condition.....	1206
33.11	Bus Hanging .....	1206
33.11.1	Timeout Function .....	1207
33.11.2	Extra SCL Clock Cycle Output Function.....	1208
33.11.3	RIIC Reset and Internal Reset .....	1209
33.12	SMBus Operation .....	1209
33.12.1	SMBus Timeout Measurement.....	1209
33.12.2	Packet Error Code (PEC).....	1211
33.12.3	SMBus Host Notification Protocol/Notify ARP Master .....	1211
33.13	Interrupt Request.....	1212
33.13.1	Buffer Operation for ICTXI and ICRXI Interrupts .....	1212
33.14	Reset States .....	1213
33.15	Usage Notes.....	1214
33.15.1	Setting Module-Stop Function.....	1214
33.15.2	Points to Note on Starting Transfer .....	1214
34.	CAN Module (CAN).....	1215
34.1	Overview.....	1215
34.2	Register Descriptions.....	1217
34.2.1	Control Register (CTRL).....	1217
34.2.2	Bit Configuration Register (BCR).....	1221
34.2.3	Mask Register k (MKRk) (k = 0 to 7) .....	1223
34.2.4	FIFO Received ID Compare Registers 0 and 1 (FIDCR0 and FIDCR1) .....	1224
34.2.5	Mask Invalid Register (MKIVLR) .....	1225
34.2.6	Mailbox Register j (MBj) (j = 0 to 31) .....	1226
34.2.7	Mailbox Interrupt Enable Register (MIER).....	1230
34.2.8	Message Control Register j (MCTLj) (j = 0 to 31) .....	1231
34.2.9	Receive FIFO Control Register (RFCR) .....	1234
34.2.10	Receive FIFO Pointer Control Register (RFPCR) .....	1237
34.2.11	Transmit FIFO Control Register (TFCR).....	1237
34.2.12	Transmit FIFO Pointer Control Register (TFPCR) .....	1239
34.2.13	Status Register (STR) .....	1240

34.2.14	Mailbox Search Mode Register (MSMR) .....	1242
34.2.15	Mailbox Search Status Register (MSSR) .....	1243
34.2.16	Channel Search Support Register (CSSR).....	1244
34.2.17	Acceptance Filter Support Register (AFSR) .....	1245
34.2.18	Error Interrupt Enable Register (EIER).....	1246
34.2.19	Error Interrupt Factor Judge Register (EIFR).....	1247
34.2.20	Receive Error Count Register (RECR).....	1249
34.2.21	Transmit Error Count Register (TECR) .....	1250
34.2.22	Error Code Store Register (ECSR).....	1250
34.2.23	Time Stamp Register (TSR) .....	1251
34.2.24	Test Control Register (TCR) .....	1252
34.3	Operating Mode .....	1254
34.3.1	CAN Reset Mode.....	1254
34.3.2	CAN Halt Mode.....	1255
34.3.3	CAN Sleep Mode.....	1256
34.3.4	CAN Operation Mode (Excluding Bus-Off State) .....	1257
34.3.5	CAN Operation Mode (Bus-Off State).....	1258
34.4	CAN Communication Speed Setting.....	1259
34.4.1	CAN Clock Setting.....	1259
34.4.2	Bit Timing Setting .....	1259
34.4.3	Bit Rate .....	1260
34.5	Mailbox and Mask Register Structure.....	1261
34.6	Acceptance Filtering and Masking Functions .....	1262
34.7	Reception and Transmission .....	1264
34.7.1	Reception .....	1265
34.7.2	Transmission.....	1267
34.8	CAN Interrupt.....	1268
34.9	Usage Notes .....	1268
34.9.1	Setting for the Module-Stop State .....	1268
35.	Serial Peripheral Interface (RSPI).....	1269
35.1	Overview.....	1269
35.2	Register Descriptions.....	1273
35.2.1	RSPI Control Register (SPCR).....	1273
35.2.2	RSPI Slave Select Polarity Register (SSLP) .....	1274
35.2.3	RSPI Pin Control Register (SPPCR) .....	1275
35.2.4	RSPI Status Register (SPSR).....	1276
35.2.5	RSPI Data Register (SPDR) .....	1278
35.2.6	RSPI Sequence Control Register (SPSCR) .....	1281
35.2.7	RSPI Sequence Status Register (SPSSR).....	1282
35.2.8	RSPI Bit Rate Register (SPBR).....	1283

35.2.9	RSPI Data Control Register (SPDCR) .....	1284
35.2.10	RSPI Clock Delay Register (SPCKD).....	1285
35.2.11	RSPI Slave Select Negation Delay Register (SSLND) .....	1286
35.2.12	RSPI Next-Access Delay Register (SPND).....	1286
35.2.13	RSPI Control Register 2 (SPCR2).....	1287
35.2.14	RSPI Command Registers 0 to 7 (SPCMD0 to SPCMD7) .....	1288
<b>35.3</b>	<b>Operation .....</b>	<b>1291</b>
35.3.1	Overview of RSPI Operations .....	1291
35.3.2	Controlling RSPI Pins .....	1292
35.3.3	RSPI System Configuration Examples.....	1293
35.3.3.1	Single Master/Single Slave (with This LSI Acting as Master) .....	1293
35.3.3.2	Single Master/Single Slave (with This LSI Acting as Slave) .....	1294
35.3.3.3	Single Master/Multi-Slave (with This LSI Acting as Master) .....	1295
35.3.3.4	Single Master/Multi-Slave (with This LSI Acting as Slave) .....	1296
35.3.3.5	Multi-Master/Multi-Slave (with This LSI Acting as Master) .....	1297
35.3.3.6	Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This LSI Acting as Master) .....	1298
35.3.3.7	Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This LSI Acting as Slave) .....	1298
35.3.4	Data Format .....	1299
35.3.4.1	When Parity is Disabled (SPCR2.SPPE = 0) .....	1300
35.3.4.2	When Parity is Enabled (SPCR2.SPPE = 1) .....	1304
35.3.5	Transfer Format .....	1308
35.3.5.1	CPHA = 0 .....	1308
35.3.5.2	CPHA = 1 .....	1309
35.3.6	Communications Operating Mode.....	1310
35.3.6.1	Full-Duplex Synchronous Serial Communications (SPCR.TXMD = 0) .....	1310
35.3.6.2	Transmit Operations Only (SPCR.TXMD = 1) .....	1311
35.3.7	Transmit Buffer Empty/Receive Buffer Full Interrupts .....	1312
35.3.8	Error Detection .....	1314
35.3.8.1	Overflow Error .....	1315
35.3.8.2	Parity Error .....	1316
35.3.8.3	Mode Fault Error .....	1317
35.3.9	Initializing RSPI .....	1318
35.3.9.1	Initialization by Clearing the SPE Bit .....	1318
35.3.9.2	System Reset .....	1318
35.3.10	SPI Operation .....	1319
35.3.10.1	Master Mode Operation .....	1319
35.3.10.2	Slave Mode Operation .....	1329
35.3.11	Clock Synchronous Operation.....	1333
35.3.12	Master Mode Operation.....	1333

35.3.13	Slave Mode Operation.....	1337
35.3.14	Loopback Mode.....	1339
35.3.15	Self-Diagnosis of Parity Bit Function .....	1340
35.3.16	Interrupt Sources.....	1341
35.4	Usage Note.....	1342
35.4.1	Setting Module-Stop Function.....	1342
35.4.2	Cautionary Note on the Low Power Consumption Functions.....	1342
35.4.3	Points to Note on Starting Transfer .....	1342
36.	IEBus Controller (IEB).....	1343
36.1	Overview.....	1343
36.1.1	Transfer Data (Data Field Contents) .....	1344
36.2	Register Descriptions.....	1349
36.2.1	IEBus Control Register (IECTR) .....	1349
36.2.2	IEBus Command Register (IECMR).....	1350
36.2.3	IEBus Master Control Register (IEMCR) .....	1351
36.2.4	IEBus Master Unit Address Register 1 (IEAR1).....	1352
36.2.5	IEBus Master Unit Address Register 2 (IEAR2).....	1352
36.2.6	IEBus Slave Address Setting Register 1 (IESA1).....	1353
36.2.7	IEBus Slave Address Setting Register 2 (IESA2).....	1353
36.2.8	IEBus Transmit Message Length Register (IETBFL).....	1354
36.2.9	IEBus Reception Master Address Register 1 (IEMA1).....	1354
36.2.10	IEBus Reception Master Address Register 2 (IEMA2).....	1355
36.2.11	IEBus Receive Control Field Register (IERCTL).....	1355
36.2.12	IEBus Receive Message Length Register (IERBFL) .....	1356
36.2.13	IEBus Lock Address Register 1 (IELA1).....	1356
36.2.14	IEBus Lock Address Register 2 (IELA2).....	1356
36.2.15	IEBus General Flag Register (IEFLG) .....	1357
36.2.16	IEBus Transmit Status Register (IETSR).....	1359
36.2.17	IEBus Transmit Interrupt Enable Register (IEIET).....	1361
36.2.18	IEBus Receive Status Register (IERSR) .....	1362
36.2.19	IEBus Receive Interrupt Enable Register (IEIER).....	1365
36.2.20	IEBus Clock Selection Register (IECKSR).....	1366
36.2.21	IEBus Transmit Data Buffer Registers 001 to 032 (IETB001 to IETB032) .....	1368
36.2.22	IEBus Receive Data Buffer Registers 001 to 032 (IERB001 to IERB032) .....	1368
36.3	Data Format.....	1369
36.3.1	Transmission Format .....	1369
36.3.2	Reception Format .....	1370
36.4	Control Flows.....	1371
36.4.1	Initial Setting .....	1371
36.4.2	Master Transmission.....	1372

36.4.3	Slave Reception .....	1373
36.4.4	Master Reception .....	1374
36.4.5	Slave Transmission.....	1375
36.5	Operation Timing .....	1376
36.5.1	Master Transmission.....	1376
36.5.2	Slave Reception .....	1377
36.5.3	Master Reception .....	1378
36.5.4	Slave Transmission.....	1379
36.6	Interrupt Sources .....	1380
36.7	Usage Notes .....	1381
36.7.1	Notes on Operation when Transfer Is Not Completed within Maximum Number of Transfer Bytes.....	1381
36.7.2	Notes on Operation when Message Length Is Greater than Maximum Number of Transfer Bytes.....	1383
36.7.3	Setting for the Module-Stop State .....	1384
37.	CRC Calculator (CRC).....	1385
37.1	Overview.....	1385
37.2	Register Descriptions.....	1386
37.2.1	CRC Control Register (CRCCR).....	1386
37.2.2	CRC Data Input Register (CRCDIR) .....	1386
37.2.3	CRC Data Output Register (CRCDOR) .....	1387
37.3	Operation.....	1388
37.4	Usage Notes .....	1391
37.4.1	Module-Stop Function Setting.....	1391
37.5	Note on Transmission.....	1391
38.	12-Bit A/D Converter (S12ADa) .....	1392
38.1	Overview.....	1392
38.2	Register Descriptions.....	1395
38.2.1	A/D Control Register (ADCSR).....	1395
38.2.2	A/D Channel Select Register 0 (ADANS0) .....	1397
38.2.3	A/D Channel Select Register 1 (ADANS1) .....	1397
38.2.4	A/D-Converted Value Addition Mode Select Register 0 (ADADS0).....	1398
38.2.5	A/D-Converted Value Addition Mode Select Register 1 (ADADS1).....	1398
38.2.6	A/D-Converted Value Addition Count Select Register (ADADC).....	1399
38.2.7	A/D Control Extended Register (ADCER) .....	1400
38.2.8	A/D Start Trigger Select Register (ADSTRGR) .....	1401
38.2.9	A/D-Converted Extended Input Control Register (ADEXICR).....	1402
38.2.10	A/D Temperature Sensor Data Register (ADTSDR) .....	1403
38.2.11	A/D Internal Reference Voltage Data Register (ADOCDR).....	1404
38.2.12	A/D Data Register y (ADDRy) (y = 0 to 20) .....	1405
38.2.13	A/D Sampling State Register 01 (ADSSSTR01).....	1406

38.2.14	A/D Sampling State Register 23 (ADSSSTR23).....	1407
<b>38.3</b>	<b>Operation.....</b>	<b>1408</b>
38.3.1	Scanning Operation.....	1408
38.3.2	Single Scan Mode.....	1408
38.3.2.1	Basic Operation.....	1408
38.3.2.2	A/D Conversion when selecting Temperature Sensor Output/Internal Reference Voltage.....	1409
38.3.3	Continuous Scan Mode.....	1410
38.3.3.1	Basic Operation.....	1410
38.3.4	Analog Input Scan Time.....	1411
38.3.5	Usage Example of ADDRy Register Automatic Clearing Function.....	1412
38.3.6	A/D-Converted Value Addition Function.....	1412
38.3.7	Starting A/D Conversion with Asynchronous Trigger.....	1412
38.3.8	Starting A/D Conversion with Synchronous Trigger from Peripheral Modules.....	1413
38.3.8.1	Starting A/D Conversion Using MTU TRG0AN_0 and TRG0BN_0.....	1414
38.3.8.2	Starting A/D Conversion Using MTU TRGAN_0 and TPU TRGAN_1.....	1414
38.3.8.3	Starting A/D Conversion Using MTU TRG0EN_0 and TRG0FN_0.....	1416
38.3.8.4	Starting A/D Conversion Using MTU TRG4ABN_0 and TPU TRG4ABN_1.....	1417
38.3.8.5	Starting A/D Conversion Using TMR TMTRG0AN_0 and TMTRG0AN_1.....	1418
<b>38.4</b>	<b>Interrupt Sources and DMA Transfer Requests.....</b>	<b>1418</b>
38.4.1	Interrupt Request on Completion of Each Scanning Conversion.....	1418
<b>38.5</b>	<b>Usage Notes.....</b>	<b>1419</b>
38.5.1	Notes on Reading Data Registers.....	1419
38.5.2	Notes on Forcibly Stopping A/D Conversion.....	1419
38.5.3	Notes on Restarting A/D Conversion after a Forcible Stop.....	1419
38.5.4	Module-Stop Function Setting.....	1419
38.5.5	Notes on Entering Low Power Consumption States.....	1419
38.5.6	Notes on Canceling Software Standby Mode.....	1419
38.5.7	Influences on Absolute Precision.....	1419
38.5.8	Analog Power Supply Pin Setting Range.....	1420
38.5.9	Notes on Board Design.....	1420
38.5.10	Notes on Noise Countermeasures.....	1421
38.5.11	Port Setting for Using 12-bit A/D Converter Inputs.....	1421
38.5.12	Notes on Using the External Bus.....	1421
<b>39.</b>	<b>10-Bit A/D Converter (ADb).....</b>	<b>1422</b>
<b>39.1</b>	<b>Overview.....</b>	<b>1422</b>
<b>39.2</b>	<b>Register Descriptions.....</b>	<b>1425</b>
39.2.1	A/D Data Register y (ADDRy) (y = A to H).....	1425
39.2.2	A/D Control/Status Register (ADCSR).....	1426
39.2.3	A/D Control Register (ADCR).....	1427



39.2.4	A/D Control Register 2 (ADCR2).....	1428
39.2.5	A/D Sampling State Register (ADSSTR).....	1428
39.2.6	A/D Self-Diagnostic Register (ADDIAGR).....	1429
<b>39.3</b>	<b>Operation.....</b>	<b>1430</b>
39.3.1	Single Channel Mode .....	1431
39.3.2	Scan Mode .....	1432
39.3.2.1	Continuous Scan Mode .....	1432
39.3.2.2	Single Scan Mode .....	1433
39.3.3	Extended Analog Input.....	1434
39.3.3.1	Usage of ANEX1 .....	1435
39.3.4	Input Sampling and A/D Conversion Time.....	1436
39.3.5	Starting 10-Bit A/D Conversion with Asynchronous Trigger.....	1438
39.3.6	Using Synchronous Triggers from Peripheral Modules to Start A/D Conversion.....	1439
39.3.6.1	Starting A/D Conversion Using MTU TRG0AN_0 .....	1439
39.3.6.2	Starting A/D Conversion Using MTU TRGAN_0 and TPU TRGAN_1 .....	1439
39.3.7	Starting A/D Conversion Using MTU TRG4ABN_0 and TPU TRG4ABN_1.....	1441
39.3.8	Starting A/D Conversion Using TMR TMTRG0AN_0 .....	1442
39.3.9	10-Bit A/D Converter Synchronous D/A Conversion Enable Signal.....	1442
<b>39.4</b>	<b>Interrupt Sources .....</b>	<b>1443</b>
<b>39.5</b>	<b>A/D Conversion Accuracy Definitions .....</b>	<b>1443</b>
<b>39.6</b>	<b>Usage Notes .....</b>	<b>1445</b>
39.6.1	Setting for the Module-Stop State .....	1445
39.6.2	Notes on Forcibly Stopping A/D Conversion.....	1445
39.6.3	A/D Conversion Restart Timing after Forced Stop.....	1445
39.6.4	Notes on Entering Power-Down States .....	1445
39.6.5	Notes on Performing 10-Bit A/D Converter Synchronous D/A Conversion .....	1445
39.6.6	Permissible Signal Source Impedance.....	1446
39.6.7	Influences on Absolute Precision .....	1446
39.6.8	Analog Power Supply Pin Setting Range.....	1447
39.6.9	Notes on Board Design.....	1447
39.6.10	Notes on Noise Countermeasures.....	1448
39.6.11	Notes on Using the External Bus.....	1448
39.6.12	Realizing High-Speed Conversion .....	1449
<b>40.</b>	<b>D/A Converter (DAa).....</b>	<b>1450</b>
40.1	Overview.....	1450
40.2	Register Descriptions.....	1451
40.2.1	D/A Data Register m (DADRm) (m = 0, 1) .....	1451
40.2.2	D/A Control Register (DACR).....	1452
40.2.3	DADRm Format Select Register (DADPR).....	1453
40.2.4	D/A A/D Synchronous Start Control Register (DAADSCR).....	1454

40.3	Operation.....	1455
40.3.1	Measure against Interference between D/A and A/D Conversion .....	1456
40.4	Usage Notes.....	1457
40.4.1	Module-Stop Function Setting.....	1457
40.4.2	Operation of the D/A Converter in Module-Stop State.....	1457
40.4.3	Operation of the D/A Converter in Software Standby Mode .....	1457
40.4.4	Note on Entering Deep Software Standby Mode .....	1457
40.4.5	Note on Usage when Measure against Interference between D/A and A/D Conversion is Enabled .....	1457
41.	Temperature Sensor .....	1458
41.1	Overview.....	1458
41.2	Register Descriptions.....	1459
41.2.1	Temperature Sensor Control Register (TSCR).....	1459
41.3	Using the Temperature Sensor.....	1460
41.3.1	Preparation for Using the Temperature Sensor .....	1460
41.3.2	Setting of 12-Bit A/D Converter .....	1461
41.3.3	Procedure for Using the Temperature Sensor.....	1462
41.3.4	Timing of A/D Conversion of Temperature Sensor Output.....	1463
41.4	Usage Note.....	1463
41.4.1	Module-Stop Function Setting.....	1463
42.	RAM.....	1464
42.1	Overview.....	1464
42.2	Operation.....	1464
42.2.1	Data Retention .....	1464
42.2.2	Low-Power Consumption Function.....	1464
43.	Flash Memory .....	1465
43.1	Overview.....	1465
43.1.1	Configuration of the ROM Area.....	1467
43.1.2	Block Configuration of the ROM.....	1468
43.1.3	Configuration of the E2 DataFlash Area .....	1470
43.1.4	Block Configuration of the E2 DataFlash .....	1470
43.2	Register Descriptions.....	1471
43.2.1	Flash Write Erase Protection Register (FWEPROR) .....	1471
43.2.2	Flash Mode Register (FMODR).....	1472
43.2.3	Flash Access Status Register (FASTAT) .....	1473
43.2.4	Flash Access Error Interrupt Enable Register (FAEINT) .....	1476
43.2.5	Flash Ready Interrupt Enable Register (FRDYIE).....	1477
43.2.6	E2 DataFlash Read Enable Register 0 (DFLRE0).....	1478
43.2.7	E2 DataFlash Read Enable Register 1 (DFLRE1).....	1479
43.2.8	E2 DataFlash P/E Enable Register 0 (DFLWE0) .....	1480

43.2.9	E2 DataFlash P/E Enable Register 1 (DFLWE1) .....	1481
43.2.10	FCU RAM Enable Register (FCURAME).....	1482
43.2.11	Flash Status Register 0 (FSTATR0).....	1483
43.2.12	Flash Status Register 1 (FSTATR1).....	1485
43.2.13	Flash P/E Mode Entry Register (FENTRYR) .....	1486
43.2.14	Flash Protection Register (FPROTR).....	1488
43.2.15	Flash Reset Register (FRESETR).....	1489
43.2.16	FCU Command Register (FCMDR).....	1490
43.2.17	FCU Processing Switching Register (FCPSR).....	1491
43.2.18	E2 DataFlash Blank Check Control Register (DFLBCCNT).....	1491
43.2.19	Flash P/E Status Register (FPESTAT).....	1492
43.2.20	E2 DataFlash Blank Check Status Register (DFLBCSTAT) .....	1492
43.2.21	Peripheral Clock Notification Register (PCKAR).....	1493
43.3	Operating Modes Associated with Flash Memory .....	1494
43.3.1	Erasure of Areas that are Subject to ID Code Protection .....	1494
43.4	FCU .....	1495
43.4.1	FCU Modes.....	1495
43.4.1.1	ROM Read Modes .....	1496
43.4.1.2	ROM/E <sup>2</sup> DataFlash Read Mode .....	1496
43.4.1.3	ROM P/E Modes .....	1496
43.4.1.4	E2 DataFlash P/E Modes .....	1497
43.4.2	FCU Commands .....	1498
43.4.3	Connections between FCU Modes and Commands .....	1500
43.4.4	FCU Command Usage.....	1501
43.4.4.1	Mode Transitions .....	1501
43.4.4.2	Programming and Erasure Procedures .....	1505
43.4.4.3	Suspension and Resumption .....	1513
43.4.4.4	Processing to Check for Errors and to Confirm the Value of the FRDY Bit .....	1516
43.5	Suspending Operation .....	1518
43.5.1	Suspension during Programming.....	1518
43.5.2	Suspension during Erasure (Suspension Priority Mode).....	1519
43.5.3	Suspension during Erasure (Erasure Priority Mode).....	1520
43.6	Protection.....	1520
43.6.1	Software Protection .....	1520
43.6.2	Command-Locked State .....	1521
43.7	User Boot Mode.....	1523
43.8	Boot Mode .....	1523
43.8.1	System Configuration .....	1523
43.8.2	State Transitions in Boot Mode.....	1524
43.8.3	Automatic Adjustment of the Bit Rate .....	1526

43.8.4	ID Code Protection (Boot Mode) .....	1527
43.8.5	UB Code A .....	1528
43.8.6	Configuration of Commands and Responses.....	1528
43.8.7	Inquiry/Selection Command Wait.....	1529
43.8.8	ID Code Wait State.....	1540
43.8.9	Programming/Erase Command Wait .....	1541
43.9	USB Boot Mode .....	1549
43.9.1	Features.....	1549
43.9.2	State Transitions .....	1550
43.9.3	Notes on Program Execution in USB Boot Mode.....	1551
43.10	ID Code Protection on Connection of the On-Chip Debugger .....	1551
43.11	ROM Code Protection.....	1552
43.12	Usage Notes (Common to the ROM/E2 DataFlash Memory).....	1553
43.13	Usage Notes (for E2 DataFlash).....	1554
44.	Boundary Scan .....	1555
44.1	Features.....	1555
44.2	Register Descriptions.....	1556
44.2.1	Instruction Register (JTIR).....	1557
44.2.2	ID Code Register (JTIDR).....	1557
44.2.3	Bypass Register (JTBPR) .....	1558
44.2.4	Boundary Scan Register (JTBSR) .....	1558
44.3	Operations .....	1568
44.3.1	TAP Controller .....	1568
44.3.2	List of Commands .....	1569
44.4	Usage Notes .....	1570
45.	Electrical Characteristics.....	1572
45.1	Absolute Maximum Ratings .....	1572
45.2	DC Characteristics.....	1573
45.3	AC Characteristics .....	1577
45.3.1	Reset Timing.....	1578
45.3.2	Clock Timing .....	1579
45.3.3	Timing of Recovery from Low Power Consumption Modes .....	1583
45.3.4	Control Signal Timing .....	1584
45.3.5	Bus Timing .....	1585
45.3.6	Timing of On-Chip Peripheral Modules.....	1591
45.4	USB Characteristics.....	1601
45.5	A/D Conversion Characteristics.....	1602
45.6	D/A Conversion Characteristics.....	1604
45.7	Temperature Sensor Characteristics .....	1604
45.8	Power-on Reset Circuit and Voltage Detection Circuit Characteristics .....	1605

45.9	Oscillation Stop Detection Timing.....	1608
45.10	Battery Backup Function Characteristics.....	1608
45.11	ROM (Flash Memory for Code Storage) Characteristics.....	1609
45.12	E2 Flash Characteristics.....	1610
Appendix 1.	Port States in Each Processing Mode .....	1612
Appendix 2.	Package Dimensions .....	1615
REVISION HISTORY	.....	1623

100-MHz 32-bit RX MCU, on-chip FPU, 165 DMIPS,  
up to 2-MB flash memory, USB 2.0 full-speed function interface,  
CAN, 10- & 12-bit A/D converter, RTC, up to 22 comms interfaces

## Features

### ■ 32-bit RX CPU core

- Max. operating frequency: 100 MHz  
Capable of 165 DMIPS in operation at 100 MHz
- Single precision 32-bit IEEE-754 floating point
- Two types of multiply-and-accumulation unit (between memories and between registers)
- 32-bit multiplier (fastest instruction execution takes one CPU clock cycle)
- Divider (fastest instruction execution takes two CPU clock cycles)
- Fast interrupt
- CISC Harvard architecture with 5-stage pipeline
- Variable-length instructions: Ultra-compact code
- Supports the memory protection unit (MPU)
- Two types of debugging interfaces: JTAG and FINE (two-line)

### ■ Low-power design and architecture

- Operation from a single 2.7- to 3.6-V supply
- Low power consumption: A product that supports all peripheral functions draws only 500  $\mu$ A/MHz.
- RTC is capable of operation from a dedicated power supply (min. operating voltage: 2.3 V).
- Four low-power modes

### ■ On-chip main flash memory, no wait states

- 100-MHz operation, 10-ns read cycle (no wait states)
- 384-Kbyte to 2-Mbyte capacities
- User code is programmable by on-board or off-board programming.

### ■ On-chip data flash memory

- Max. 32 Kbytes, reprogrammable up to 100,000 times
- Programming/erasing as background operations (BGOs)

### ■ On-chip SRAM, no wait states

- 32- to 128-Kbyte capacities
- For instructions and operands
- Can provide backup on deep software standby

### ■ DMA

- DMAC: Incorporates four channels
- DTC

### ■ Reset and supply management

- Power-on reset (POR)
- Low voltage detection (LVD) with voltage settings

### ■ Clock functions

- External crystal oscillator or internal PLL for operation at 4 to 16 MHz
- Internal 125-kHz LOCO and 50-MHz HOCO
- 125-kHz clock for the IWDT
- Frequency of the oscillator for sub-clock generation: 32 kHz

### ■ Real-time clock

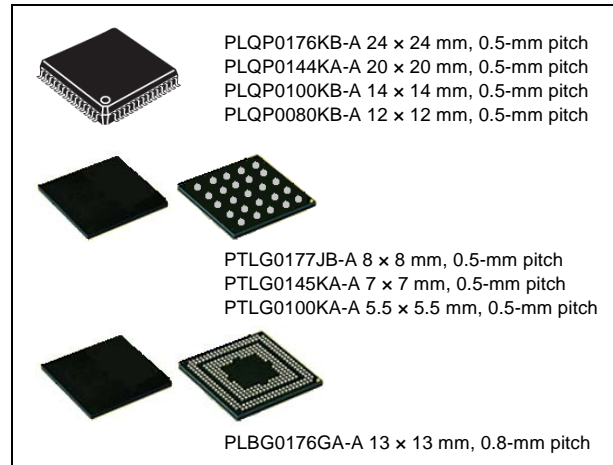
- Adjustment functions (30 seconds, leap year, and error)
- Time capture function  
(for capturing times in response to event-signal input on external pins)

### ■ Independent watchdog timer

- 125-kHz LOCO clock operation

### ■ Useful functions for IEC60730 compliance

- Oscillation-stop detection, frequency measurement, CRC, IWDT, self-diagnostic function for the A/D converter, etc.



### ■ Up to 22 communications interfaces

- USB 2.0 full-speed function interface (1 channel)
- CAN (compliant with ISO11898-1), incorporating 32 mailboxes (up to 3 channels)
- SCI with multiple functionalities (up to 13 channels)  
Choose from among asynchronous mode, clock-synchronous mode, smart-card interface mode, simple SPI, simple I<sup>2</sup>C, and extended serial mode.
- I<sup>2</sup>C bus interface for transfer at up to 1 Mbps (up to 4 channels)
- RSPI for high-speed transfer (up to 3 channels)

### ■ External address space

- 8 CS areas (8  $\times$  16 Mbytes)
- Multiplexed address data or separate address lines are selectable per area.
- 8-, 16-, or 32-bit bus space is selectable per area

### ■ Up to 20 extended-function timers

- 16-bit MTU2: input capture, output capture, complementary PWM output, phase-counting mode (6 channels)
- 16-bit TPU: input capture, output capture, phase-counting mode (12 channels)
- 8-bit TMR (4 channels)
- 16-bit compare-match timers (4 channels)

### ■ A/D converter for 1-MHz operation

- Up to 21 12-bit channels, and incorporating 1 sample-and-hold circuit
- Up to 8 10-bit channels, and incorporating 1 sample-and-hold circuit
- Addition of results of A/D conversion (in the 12-bit A/D converter)
- self-diagnosis (for the 10-bit A/D converter)

### ■ 10-bit D/A converter: 2 channels

### ■ Temperature sensor for measuring temperature within the chip

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

### ■ Up to 148 general I/O port pins for GPIO

- 5-V tolerance, open drain, input pull-up, switchable driving ability

### ■ Operating temp. range

- -40°C to +85°C

## 1. Overview

### 1.1 Outline of Specifications

Table 1.1 lists the specifications in outline, and Table 1.2 lists the functions of products.

Table 1.1 shows the outline of maximum specifications, and the number of peripheral module channels differs depending on the pin number on the package and the ROM capacity. For details, see Table 1.2, Comparison of Functions for Different Packages.

**Table 1.1 Outline of Specifications (1/5)**

Classification	Module/Function	Description
CPU	CPU	<ul style="list-style-type: none"> <li>Maximum operating frequency: 100 MHz</li> <li>32-bit RX CPU</li> <li>Minimum instruction execution time: One instruction per state (cycle of the system clock)</li> <li>Address space: 4-Gbyte linear</li> <li>Register set of the CPU General purpose: Sixteen 32-bit registers Control: Nine 32-bit registers Accumulator: One 64-bit register</li> <li>Basic instructions: 73</li> <li>Floating-point operation instructions: 8</li> <li>DSP instructions: 9</li> <li>Addressing modes: 10</li> <li>Data arrangement Instructions: Little endian Data: Selectable as little endian or big endian</li> <li>On-chip 32-bit multiplier: 32 × 32 → 64 bits</li> <li>On-chip divider: 32 / 32 → 32 bits</li> <li>Barrel shifter: 32 bits</li> <li>Memory protection unit (MPU)</li> </ul>
	FPU	<ul style="list-style-type: none"> <li>Single precision floating point (32 bits)</li> <li>Data types and floating-point exceptions in conformance with the IEEE754 standard</li> </ul>
Memory	ROM	<ul style="list-style-type: none"> <li>Capacity: 384 Kbytes, 512 Kbytes, 768 Kbytes, 1 Mbyte, 1.5 Mbytes, 2 Mbytes</li> <li>100 MHz, no-wait access</li> <li>On-board programming: Four types</li> <li>Off-board programming (parallel programmer mode)</li> </ul>
	RAM	<ul style="list-style-type: none"> <li>Capacity: 64 Kbytes, 96 Kbytes, 128 Kbytes</li> <li>100 MHz, no-wait access</li> </ul>
	E <sup>2</sup> data flash	<ul style="list-style-type: none"> <li>Capacity: 32 Kbytes</li> <li>Programming/erasing: 100,000 times</li> </ul>
MCU operating modes		Single-chip mode, on-chip ROM enabled extended mode, and on-chip ROM disabled extended mode (software switching)
Clock	Clock generation circuit	<ul style="list-style-type: none"> <li>Main clock oscillator, sub-clock oscillator, low-speed/high-speed on-chip oscillator, PLL frequency synthesizer, and dedicated on-chip oscillator for the IWDG</li> <li>Main-clock oscillation stop detection</li> <li>Separate frequency-division and multiplication settings for the system clock (ICLK), peripheral module clock (PCLK), FlashIF clock (FCLK) and external bus clock (BCLK) The CPU and other bus masters run in synchronization with the system clock (ICLK): Up to 100 MHz Peripheral modules run in synchronization with the peripheral module clock (PCLK): Up to 50 MHz Flash IF run in synchronization with the FlashIF clock (FCLK): Up to 50 MHz Devices connected to the external bus run in synchronization with the external bus clock (BCLK): Up to 50 MHz</li> </ul>
Reset		RES# pin reset, power-on reset, voltage-monitoring reset, independent watchdog timer reset, watchdog timer reset, deep software standby reset, and software reset
Voltage detection circuit		When the voltage on VCC passes the voltage detection level (Vdet), an internal reset or internal interrupt is generated.

**Table 1.1 Outline of Specifications (2/5)**

Classification	Module/Function	Description
Low power consumption	Low power consumption facilities	<ul style="list-style-type: none"> <li>Module stop function</li> <li>Four low power consumption modes Sleep mode, all-module clock stop mode, software standby mode, and deep software standby mode</li> <li>Battery backup function</li> </ul>
Interrupt	Interrupt controller (ICUb)	<ul style="list-style-type: none"> <li>Peripheral function interrupts: 180 sources</li> <li>External interrupts: 16 (pins IRQ0 to IRQ15)</li> <li>Software interrupts: One source</li> <li>Non-maskable interrupts: 6 sources</li> <li>Sixteen levels specifiable for the order of priority</li> </ul>
External bus extension		<ul style="list-style-type: none"> <li>The external address space can be divided into eight areas (CS0 to CS7), each with independent control of access settings. Capacity of each area: 16 Mbytes (CS0 to CS7) A chip-select signal (CS0# to CS7#) can be output for each area. Each area is specifiable as an 8-, 16- or 32-bit bus space The data arrangement in each area is selectable as little or big endian (only for data).</li> <li>Bus format: Separate bus, multiplex bus</li> <li>Wait control</li> <li>Write buffer facility</li> </ul>
DMA	DMA controller (DMACA)	<ul style="list-style-type: none"> <li>4 channels</li> <li>Three transfer modes: Normal transfer, repeat transfer, and block transfer</li> <li>Activation sources: Software trigger, external interrupts, and interrupt requests from peripheral functions</li> </ul>
	Data transfer controller (DTCa)	<ul style="list-style-type: none"> <li>Three transfer modes: Normal transfer, repeat transfer, and block transfer</li> <li>Activation sources: External interrupts and interrupt requests from peripheral functions</li> </ul>
I/O ports	General I/O port pins	<ul style="list-style-type: none"> <li>177-pin TFLGA (in planning), 176-pin LFBGA (in planning), 176-pin LQFP I/O pins: 148 Input pin: 1 Pull-up resistors: 148 Open-drain outputs: 148 5-V tolerance: 54</li> <li>145-pin TFLGA (in planning), 144-pin LQFP I/O pins: 117 Input pin: 1 Pull-up resistors: 117 Open-drain outputs: 117 5-V tolerance: 53</li> <li>100-pin TFLGA (in planning), 100-pin LQFP I/O pins: 78 Input pin: 1 Pull-up resistors: 78 Open-drain outputs: 78 5-V tolerance: 44</li> <li>80-pin LQFP (in planning) I/O pins: 58 Input pin: 1 Pull-up resistors: 58 Open-drain outputs: 58 5-V tolerance: 34</li> </ul>



**Table 1.1 Outline of Specifications (3/5)**

Classification	Module/Function	Description
Timers	16-bit timer pulse unit (TPUa)	<ul style="list-style-type: none"> <li>• (16 bits × 6 channels) × 2 units</li> <li>• Maximum of 16 pulse-input/output possible</li> <li>• Select from among seven or eight counter-input clock signals for each channel</li> <li>• Supports the input capture/output compare function</li> <li>• Output of PWM waveforms in up to 15 phases in PWM mode</li> <li>• Support for buffered operation, phase-counting mode (two phase encoder input) and cascade-connected operation (32 bits × 2 channels) depending on the channel.</li> <li>• PPG output trigger can be generated</li> <li>• Capable of generating conversion start triggers for the A/D converters</li> <li>• Signals from the input capture pins are input via a digital filter</li> <li>• Clock frequency measuring method</li> </ul>
	Multi-function timer pulse unit 2 (MTU2a)	<ul style="list-style-type: none"> <li>• (16 bits × 6 channels) × 1 unit</li> <li>• Time bases for the 6 16-bit timer channels can be provided via up to 16 pulse-input/output lines and three pulse-input lines</li> <li>• Select from among eight counter-input clock signals for each channel (PCLK/1, PCLK/4, PCLK/16, PCLK/64, MTCLKA, MTCLKB, MTCLKC, MTCLKD) other than channel 5, for which only four signals are available.</li> <li>• Input capture function</li> <li>• 21 output compare/input capture registers</li> <li>• Complementary PWM output mode</li> <li>• Reset synchronous PWM mode</li> <li>• Phase-counting mode</li> <li>• Generation of triggers for A/D converter conversion</li> <li>• Digital filter</li> <li>• Signals from the input capture pins are input via a digital filter</li> <li>• PPG output trigger can be generated</li> <li>• Clock frequency measuring function</li> </ul>
	Frequency measurement function (MCK)	The MTU or unit 0 TPU module can be used to monitor the main clock, sub-clock, HOCO clock, LOCO clock, and PLL clock for abnormal frequencies.
	Port output enable 2 (POE2a)	Controls the high-impedance state of the MTU's waveform output pins
	Programmable pulse generator (PPG)	<ul style="list-style-type: none"> <li>• (4 bits × 4 groups) × 2 units</li> <li>• Pulse output with the MTU or TPU output as a trigger</li> <li>• Maximum of 32 pulse-output possible</li> </ul>
	8-bit timers (TMR)	<ul style="list-style-type: none"> <li>• (8 bits × 2 channels) × 2 units</li> <li>• Select from among seven internal clock signals (PCLK/1, PCLK/2, PCLK/8, PCLK/32, PCLK/64, PCLK/1024, PCLK/8192) and one external clock signal</li> <li>• Capable of output of pulse trains with desired duty cycles or of PWM signals</li> <li>• The 2 channels of each unit can be cascaded to create a 16-bit timer</li> <li>• Generation of triggers for A/D converter conversion</li> <li>• Capable of generating baud-rate clocks for SCI5, SCI6, and SCI12</li> </ul>
	Compare match timer (CMT)	<ul style="list-style-type: none"> <li>• (16 bits × 2 channels) × 2 units</li> <li>• Select from among four internal clock signals (PCLK/8, PCLK/32, PCLK/128, PCLK/512)</li> </ul>
	Realtime clock (RTCa)	<ul style="list-style-type: none"> <li>• Clock sources: Main clock, sub-clock</li> <li>• Clock and calendar functions</li> <li>Interrupt sources: Alarm interrupt, periodic interrupt, and carry interrupt</li> <li>• Battery backup operation</li> <li>• Time-capture facility for three values</li> </ul>
	Watchdog timer (WDTA)	<ul style="list-style-type: none"> <li>• 14 bits × 1 channel</li> <li>• Select from among 6 counter-input clock signals (PCLK/4, PCLK/64, PCLK/128, PCLK/512, PCLK/2048, PCLK/8192)</li> </ul>
	Independent watchdog timer (IWDTA)	<ul style="list-style-type: none"> <li>• 14 bits × 1 channel</li> <li>• Counter-input clock: Dedicated on-chip oscillator for the IWDT</li> <li>• Dedicated clock/1, dedicated clock/16, dedicated clock/32, dedicated clock/64, dedicated clock/128, dedicated clock/256</li> </ul>

**Table 1.1 Outline of Specifications (4/5)**

Classification	Module/Function	Description
Communication function	USB 2.0 function module (USBa)	<ul style="list-style-type: none"> <li>Includes a UDC (USB Device Controller) and transceiver for USB 2.0</li> <li>Single port</li> <li>Compliance with the USB 2.0 specification</li> <li>Transfer rate: Full speed (12 Mbps)</li> <li>Self-power mode and bus power are selectable</li> <li>Incorporates 2 Kbytes of RAM as a transfer buffer</li> </ul>
	Serial communications interfaces (SC1c, SC1d)	<ul style="list-style-type: none"> <li>13 channels (SC1c: 12 channels + SC1d: 1 channel)</li> <li>SC1c <ul style="list-style-type: none"> <li>Serial communications modes: Asynchronous, clock synchronous, and smart-card interface</li> <li>Multi-processor function</li> <li>On-chip baud rate generator allows selection of the desired bit rate</li> <li>Choice of LSB-first or MSB-first transfer</li> <li>Average transfer rate clock can be input from TMR timers for SC15, SC16, and SC12</li> <li>Simple I<sup>2</sup>C</li> <li>Simple SPI</li> </ul> </li> <li>SC1d (The following functions are added to SC1c) <ul style="list-style-type: none"> <li>Supports the serial communications protocol, which contains the start frame and information frame</li> <li>Supports the LIN format</li> </ul> </li> </ul>
	I <sup>2</sup> C bus interfaces (RIIC)	<ul style="list-style-type: none"> <li>4 channels (one of them is FM+)</li> <li>Communication formats</li> <li>I<sup>2</sup>C bus format/SMBus format</li> <li>Supports the multi-master</li> <li>Max. transfer rate: 1 Mbps (channel 0)</li> </ul>
	IEBus (IEB)	<ul style="list-style-type: none"> <li>1 channel</li> <li>Supports protocol control for the IEBus</li> <li>Half-duplex asynchronous transfer</li> <li>Multi-master operation</li> <li>Broadcast communications function</li> <li>Two selectable modes, differentiated by transfer rate</li> </ul>
	CAN module (CAN)	<ul style="list-style-type: none"> <li>3 channels</li> <li>Compliance with the ISO11898-1 specification (standard frame and extended frame)</li> <li>32 mailboxes per channel</li> </ul>
	Serial peripheral interfaces (RSPI)	<ul style="list-style-type: none"> <li>3 channels</li> <li>RSPI transfer facility <ul style="list-style-type: none"> <li>Using the MOSI (master out, slave in), MISO (master in, slave out), SSL (slave select), and RSPCK (RSPI clock) signals enables serial transfer through SPI operation (four lines) or clock-synchronous operation (three lines)</li> <li>Capable of handling serial transfer as a master or slave</li> </ul> </li> <li>Data formats <ul style="list-style-type: none"> <li>Switching between MSB first and LSB first</li> <li>The number of bits in each transfer can be changed to any number of bits from 8 to 16, or to 20, 24, or 32 bits.</li> <li>128-bit buffers for transmission and reception</li> <li>Up to four frames can be transmitted or received in a single transfer operation (with each frame having up to 32 bits)</li> </ul> </li> <li>Buffered structure <ul style="list-style-type: none"> <li>Double buffers for both transmission and reception</li> </ul> </li> </ul>
	12-bit A/D converter (S12ADa)	<ul style="list-style-type: none"> <li>1 unit (1 unit × 21 channels)</li> <li>12-bit resolution</li> <li>Conversion time: 1.0 μs per channel (in operation with PCLK at 50 MHz)</li> <li>Operating mode <ul style="list-style-type: none"> <li>Scan mode (single scan mode or continuous scan mode)</li> </ul> </li> <li>Sample-and-hold function</li> <li>Reference voltage generation</li> <li>Three ways to start A/D conversion <ul style="list-style-type: none"> <li>Conversion can be started by a software trigger, a trigger from a timer (MTU, TPU, or TMR), or an external trigger signal</li> </ul> </li> <li>A/D conversion of the temperature sensor output</li> </ul>

**Table 1.1 Outline of Specifications (5/5)**

Classification	Module/Function	Description
10-bit A/D converter (ADb)		<ul style="list-style-type: none"> <li>• 1 unit (1 unit x 8 channels)</li> <li>• 10-bit resolution</li> <li>• Conversion time: 1.0 <math>\mu</math>s per channel (in operation with PCLK at 50 MHz)</li> <li>• Operating mode Scan mode (single scan mode or continuous scan mode) External amplifier connection mode</li> <li>• Sample-and-hold function</li> <li>• Three ways to start A/D conversion Conversion can be started by software, a conversion start trigger from a timer (MTU, TPU, or TMR), or an external trigger signal.</li> </ul>
D/A converter (DAa)		<ul style="list-style-type: none"> <li>• 2 channels</li> <li>• 10-bit resolution</li> <li>• Output voltage: 0 V to VREFH</li> </ul>
Temperature sensor		<ul style="list-style-type: none"> <li>• 1 channel</li> <li>• Precision: <math>\pm 1</math> <math>^{\circ}</math>C</li> <li>• The voltage of the temperature is converted into a digital value by the 12-bit A/D converter.</li> </ul>
CRC calculator (CRC)		<ul style="list-style-type: none"> <li>• CRC code generation for arbitrary amounts of data in 8-bit units</li> <li>• Select any of three generating polynomials: <math>X^8 + X^2 + X + 1</math>, <math>X^{16} + X^{15} + X^2 + 1</math>, or <math>X^{16} + X^{12} + X^5 + 1</math>.</li> <li>• Generation of CRC codes for use with LSB-first or MSB-first communications is selectable</li> </ul>
Operating frequency		Up to 100 MHz
Power supply voltage		VCC = AVCC0 = VREFH = VCC_USB = 2.7 to 3.6 V, VREFH0 = 2.7 V to AVCC0, Vbatt = 2.3 to 3.6 V
Operating temperature		-40 to +85 $^{\circ}$ C (products with wide-temperature-range spec.)
Package		177-pin TFLGA (PTLG0177KA-A) (in planning) 176-pin LFBGA (PLBG0176GA-A) (in planning) 176-pin LQFP (PLQP0176KB-A) 145-pin TFLGA (PTLG0145KA-A) (in planning) 144-pin LQFP (PLQP0144KA-A) 100-pin TFLGA (PTLG0100KA-A) (in planning) 100-pin LQFP (PLQP0100KB-A) 80-pin LQFP (PLQP0080KB-A) (in planning)
On-chip debugging system		<ul style="list-style-type: none"> <li>• E1 emulator (JTAG and FINE interfaces)</li> <li>• E20 emulator (JTAG interface)</li> </ul>

**Table 1.2 Comparison of Functions for Different Packages**

Functions		RX630 Group			
		177 Pins, 176 Pins	145 Pins, 144 Pins	100 Pins	80 Pins
External bus	External bus width	32 bits	16 bits		Not supported
DMA	DMA controller	Ch. 0 to 3			
	Data transfer controller	Supported			
Timers	16-bit timer pulse unit	Ch. 0 to 11		Ch. 0 to 5	
	Multi-function timer pulse unit 2	Ch. 0 to 5			
	Port output enable 2	Supported			
	Programmable pulse generator	Ch. 0 and 1			
	8-bit timers	Ch. 0 to 3			
	Compare match timer	Ch. 0 to 3			
	Realtime clock	Supported			
	Watchdog timer	Supported			
	Independent watchdog timer	Supported			
Communication function	USB 2.0 function module	Ch. 0			
	Serial communications interfaces (SC1c)	Ch. 0 to 11		Ch. 0 to 3, 5, 6, 8, 9	Ch. 1, 5, 6, 8, 9
	Serial communications interfaces (SC1d)	Ch. 12			
	I <sup>2</sup> C bus interfaces	Ch. 0 to 3		Ch. 0, 2	
	IEBus	Supported			
	Serial peripheral interfaces	Ch. 0 to 2		Ch. 0, 1	
	CAN module	For 1 M or less: Ch. 0, 1 For 1.5 M or more: Ch. 0 to 2		For 512 K or less: Ch. 1 For 768 K or more: Ch. 0, 1	Ch. 1
12-bit A/D converter		AN000 to 020		AN000 to 013	AN000 to 010
10-bit A/D converter		AN0 to 7			AN0 to 3
D/A converter		Ch. 0, 1		Ch. 1	
Temperature sensor		Supported			
CRC calculator		Supported			

## 1.2 List of Products

Table 1.3 is a list of products, and Figure 1.1 shows how to read the product part number.

**Table 1.3 List of Products (1/2)**

Group	Part No.	Package	ROM Capacity	RAM Capacity	E2 Data Flash	Operating Frequency (Max.)
RX630	R5F56307CDFN	PLQP0080KB-A*1	384 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F56307DDFN	PLQP0080KB-A*1	384 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F56307CDFP	PLQP0100KB-A	384 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F56307DDFP	PLQP0100KB-A	384 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F56307CDLA	PTLG0100KA-A*1	384 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F56307DDLA	PTLG0100KA-A*1	384 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F56308CDFN	PLQP0080KB-A*1	512 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F56308DDFN	PLQP0080KB-A*1	512 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F56308CDFP	PLQP0100KB-A	512 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F56308DDFP	PLQP0100KB-A	512 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F56308CDLA	PTLG0100KA-A*1	512 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F56308DDLA	PTLG0100KA-A*1	512 Kbytes	64 Kbytes	32 Kbytes	100 MHz
	R5F5630ACDFP*2	PLQP0100KB-A	768 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F5630ADDFP*2	PLQP0100KB-A	768 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F5630ACDFB*2	PLQP0144KA-A	768 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F5630ADDFB*2	PLQP0144KA-A	768 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F5630ACDLK*2	PTLG0145KA-A*1	768 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F5630ADDLK*2	PTLG0145KA-A*1	768 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F5630ACDFC	PLQP0176KB-A*1	768 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F5630ADDFC	PLQP0176KB-A*1	768 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F5630ACDBG	PLBG0176GA-A*1	768 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F5630ADDBG	PLBG0176GA-A*1	768 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F5630ACDLC	PTLG0177KA-A*1	768 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F5630ADDLC	PTLG0177KA-A*1	768 Kbytes	96 Kbytes	32 Kbytes	100 MHz
	R5F5630BCDFP*2	PLQP0100KB-A	1 Mbyte	96 Kbytes	32 Kbytes	100 MHz
	R5F5630BDDFP*2	PLQP0100KB-A	1 Mbyte	96 Kbytes	32 Kbytes	100 MHz
	R5F5630BCDFB*2	PLQP0144KA-A	1 Mbyte	96 Kbytes	32 Kbytes	100 MHz
	R5F5630BDDFB*2	PLQP0144KA-A	1 Mbyte	96 Kbytes	32 Kbytes	100 MHz
	R5F5630BCDLK*2	PTLG0145KA-A*1	1 Mbyte	96 Kbytes	32 Kbytes	100 MHz
	R5F5630BDDLK*2	PTLG0145KA-A*1	1 Mbyte	96 Kbytes	32 Kbytes	100 MHz
	R5F5630BCDFC	PLQP0176KB-A*1	1 Mbyte	96 Kbytes	32 Kbytes	100 MHz
	R5F5630BDDFC	PLQP0176KB-A*1	1 Mbyte	96 Kbytes	32 Kbytes	100 MHz
R5F5630BCDBG	PLBG0176GA-A*1	1 Mbyte	96 Kbytes	32 Kbytes	100 MHz	
R5F5630BDDBG	PLBG0176GA-A*1	1 Mbyte	96 Kbytes	32 Kbytes	100 MHz	
R5F5630BCDLC	PTLG0177KA-A*1	1 Mbyte	96 Kbytes	32 Kbytes	100 MHz	
R5F5630BDDLK	PTLG0177KA-A*1	1 Mbyte	96 Kbytes	32 Kbytes	100 MHz	
R5F5630DCDFP	PLQP0100KB-A	1.5 Mbytes	128 Kbytes	32 Kbytes	100 MHz	
R5F5630DDDFP	PLQP0100KB-A	1.5 Mbytes	128 Kbytes	32 Kbytes	100 MHz	
R5F5630DCDFB	PLQP0144KA-A	1.5 Mbytes	128 Kbytes	32 Kbytes	100 MHz	
R5F5630DDDFB	PLQP0144KA-A	1.5 Mbytes	128 Kbytes	32 Kbytes	100 MHz	
R5F5630DCDLK	PTLG0145KA-A*1	1.5 Mbytes	128 Kbytes	32 Kbytes	100 MHz	

**Table 1.3 List of Products (2/2)**

Group	Part No.	Package	ROM Capacity	RAM Capacity	E2 Data Flash	Operating Frequency (Max.)
RX630	R5F5630DDDLK	PTLG0145KA-A*1	1.5 Mbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5630DCDFC	PLQP0176KB-A	1.5 Mbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5630DDDFC	PLQP0176KB-A	1.5 Mbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5630DCDBG	PLBG0176GA-A*1	1.5 Mbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5630DDDBG	PLBG0176GA-A*1	1.5 Mbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5630DCDLC	PTLG0177KA-A*1	1.5 Mbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5630DDDLK	PTLG0177KA-A*1	1.5 Mbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5630ECDFP	PLQP0100KB-A	2 Mbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5630EDDFP	PLQP0100KB-A	2 Mbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5630ECDFB	PLQP0144KA-A	2 Mbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5630EDDFB	PLQP0144KA-A	2 Mbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5630ECDLK	PTLG0145KA-A*1	2 Mbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5630EDDLK	PTLG0145KA-A*1	2 Mbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5630ECDFC	PLQP0176KB-A	2 Mbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5630EDDFC	PLQP0176KB-A	2 Mbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5630ECDBG	PLBG0176GA-A*1	2 Mbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5630EDDBG	PLBG0176GA-A*1	2 Mbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5630ECDLC	PTLG0177KA-A*1	2 Mbytes	128 Kbytes	32 Kbytes	100 MHz
	R5F5630EDDLK	PTLG0177KA-A*1	2 Mbytes	128 Kbytes	32 Kbytes	100 MHz

Note 1. In planning

Note 2. The sub-clock oscillator and real-time clock have different specifications. For details, see section 11.2.8, Sub-Clock Oscillator Wait Control Register (SOSWTCR), and section 28.2.19, RTC Control Register 3 (RCR3).

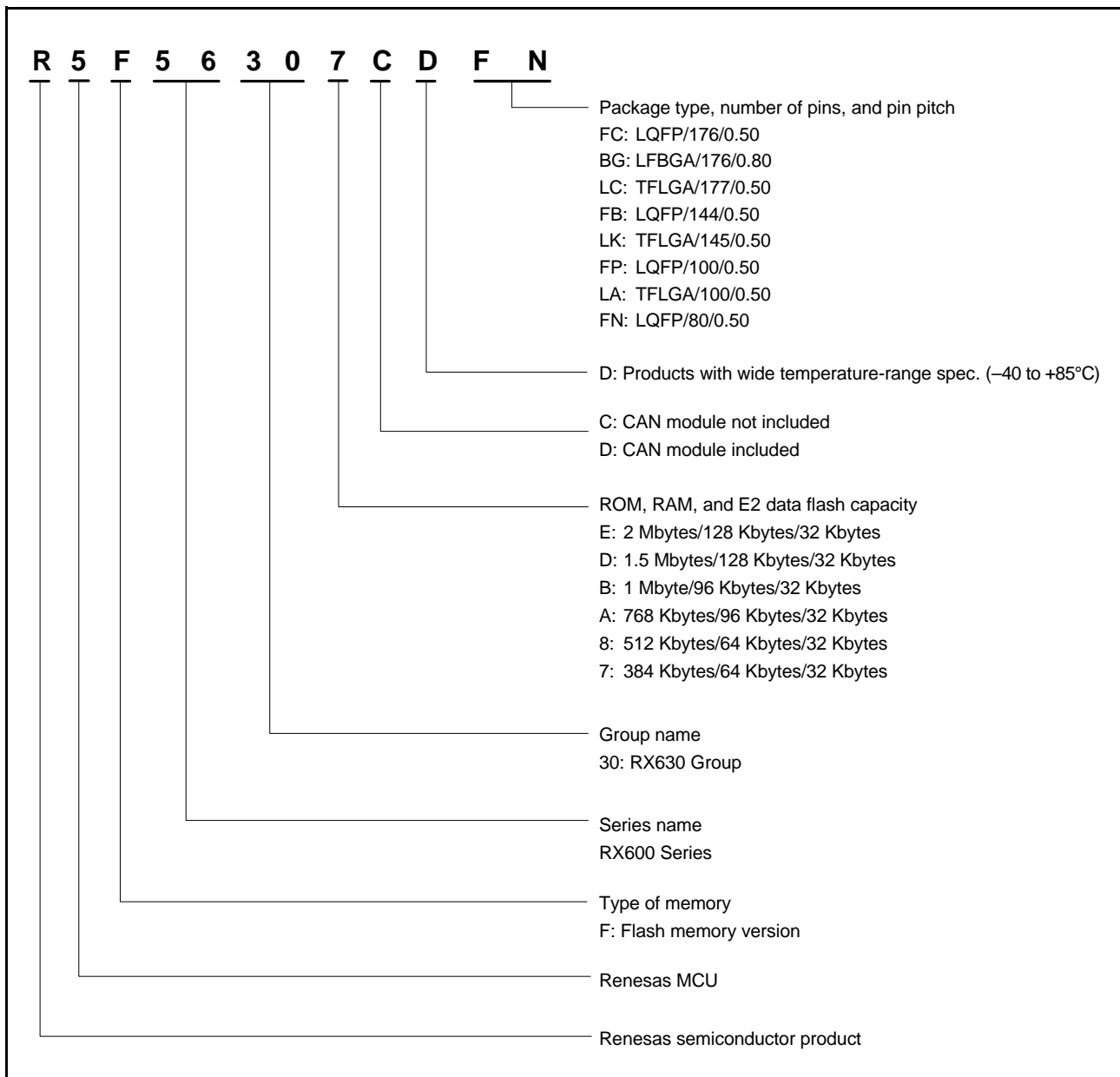


Figure 1.1 How to Read the Product Part Number

### 1.3 Block Diagram

Figure 1.2 shows a block diagram.

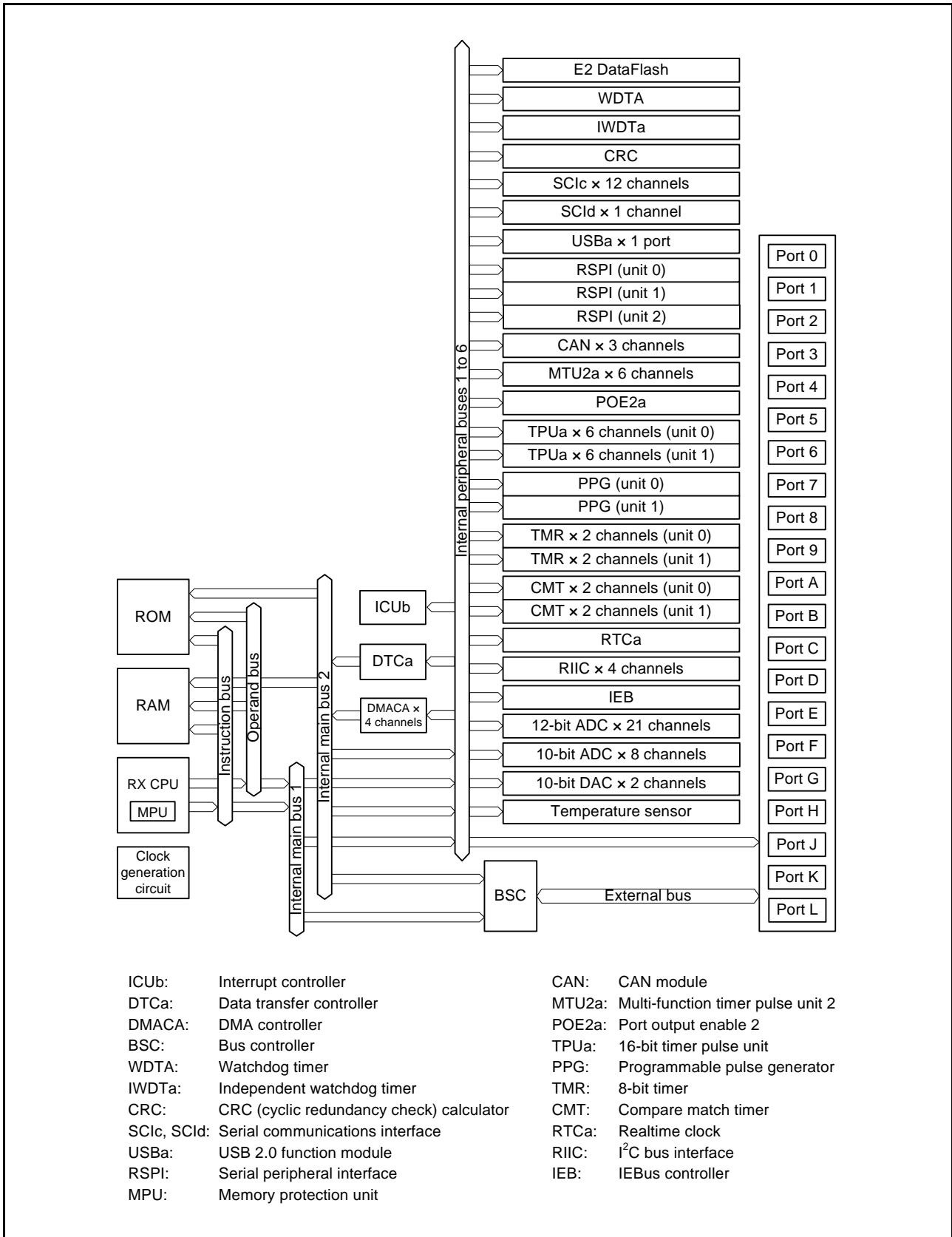


Figure 1.2 Block Diagram



## 1.4 Pin Functions

Table 1.4 lists the pin functions.

**Table 1.4 Pin Functions (1/5)**

Classifications	Pin Name	I/O	Description
Power supply	VCC	Input	Power supply pin. Connect it to the system power supply. Connect this pin to VSS via a 0.1- $\mu$ F capacitor. The capacitor should be placed close to the pin
	VCL	Input	Connect this pin to VSS via a 0.1- $\mu$ F capacitor. The capacitor should be placed close to the pin
	VSS	Input	Ground pin. Connect it to the system power supply (0 V)
	VBATT	Input	Backup power pin
Clock	XTAL	Output	Pins for a crystal resonator. An external clock signal can be input through the EXTAL pin
	EXTAL	Input	
	BCLK	Output	Outputs the external bus clock for external devices
	XCOUT	Output	Input/output pins for the sub-clock oscillator circuit. Connect a crystal resonator between XCOUT and XCIN
	XCIN	Input	
Operating mode control	MD	Input	Pin for setting the operating mode. The signal levels on these pins must not be changed during operation
System control	RES#	Input	Reset signal input pin. This LSI enters the reset state when this signal goes low
	EMLE	Input	Input pin for the on-chip emulator enable signal. When the on-chip emulator is used, this pin should be driven high. When not used, it should be driven low
	BSCANP	Input	Boundary scan enable pin. Boundary scan is enabled when this pin goes high. When not used, it should be driven low
On-chip emulator	FINEC	Input	Fine interface clock pin
	FINED	I/O	Fine interface pin
	TRST#	Input	On-chip emulator or boundary scan pins. When the EMLE pin is driven high, these pins are dedicated for the on-chip emulator
	TMS	Input	
	TDI	Input	
	TCK	Input	
	TDO	Output	
	TRCLK	Output	This pin outputs the clock for synchronization with the trace data
	TRSYNC	Output	This pin indicates that output from the TRDATA0 to TRDATA3 pins is valid
	TRDATA0 to TRDATA3	Output	These pins output the trace information
Address bus	A0 to A23	Output	Output pins for the address
Data bus	D0 to D31	I/O	Input and output pins for the bidirectional data bus
Multiplexed bus	A0/D0 to A15/D15	I/O	Address/data multiplexed bus

**Table 1.4 Pin Functions (2/5)**

Classifications	Pin Name	I/O	Description
Bus control	RD#	Output	Strobe signal which indicates that reading from the external bus interface space is in progress
	WR#	Output	Strobe signal which indicates that writing to the external bus interface space is in progress, in 1-write strobe mode
	WR0# to WR3#	Output	Strobe signals which indicate that either group of data bus pins (D7 to D0, D15 to D8, D23 to D16 and D31 to D24) is valid in writing to the external bus interface space, in byte strobe mode
	BC0# to BC3#	Output	Strobe signals which indicate that either group of data bus pins (D7 to D0, D15 to D8, D23 to D16 and D31 to D24) is valid in access to the external bus interface space, in 1-write strobe mode
	ALE	Output	Address latch signal when address/data multiplexed bus is selected
	WAIT#	Input	Input pin for wait request signals in access to the external space
	CS0# to CS7#	Output	Select signals for CS areas
Interrupt	NMI	Input	Non-maskable interrupt request pin
	IRQ0 to IRQ15	Input	Maskable interrupt request pin
Multi-function timer pulse unit 2	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/dead time compensation input pins
	MTCLKA, MTCLKB MTCLKC, MTCLKD	Input	Input pins for external clock
Port output enable 2	POE0# to POE3# POE8#	Input	Input pins for request signals to place the MTU large-current pins in the high impedance state

**Table 1.4 Pin Functions (3/5)**

Classifications	Pin Name	I/O	Description
16-bit timer pulse unit	TIOCA0, TIOCB0 TIOCC0, TIOCD0	I/O	The TGRA0 to TGRD0 input capture input/output compare output/ PWM output pins
	TIOCA1, TIOCB1	I/O	The TGRA1 and TGRB1 input capture input/output compare output/ PWM output pins
	TIOCA2, TIOCB2	I/O	The TGRA2 and TGRB2 input capture input/output compare output/ PWM output pins
	TIOCA3, TIOCB3 TIOCC3, TIOCD3	I/O	The TGRA3 to TGRD3 input capture input/output compare output/ PWM output pins
	TIOCA4, TIOCB4	I/O	The TGRA4 and TGRB4 input capture input/output compare output/ PWM output pins
	TIOCA5, TIOCB5	I/O	The TGRA5 and TGRB5 input capture input/output compare output/ PWM output pins
	TCLKA, TCLKB TCLKC, TCLKD	Input	Input pins for external clock signals
	TIOCA6, TIOCB6, TIOCC6, TIOCD6	I/O	The TGRA6 to TGRD6 input capture input/output compare output/ PWM output pins
	TIOCA7, TIOCB7	I/O	The TGRA7 and TGRB7 input capture input/output compare output/ PWM output pins
	TIOCA8, TIOCB8	I/O	The TGRA8 and TGRB8 input capture input/output compare output/ PWM output pins
	TIOCA9, TIOCB9, TIOCC9, TIOCD9	I/O	The TGRA9 to TGRD9 input capture input/output compare output/ PWM output pins
	TIOCA10, TIOCB10	I/O	The TGRA10 and TGRB10 input capture input/output compare output/PWM output pins
	TIOCA11, TIOCB11	I/O	The TGRA11 and TGRB11 input capture input/output compare output/PWM output pins
TCLKE, TCLKF, TCLKG, TCLKH	Input	Input pins for external clock signals	
Programmable pulse generator	PO0 to PO31	Output	Output pins for the pulse signals
8-bit timer	TMO0 to TMO3	Output	Compare match output pins
	TMCI0 to TMCI3	Input	Input pins for external clocks to be input to the counter
	TMRI0 to TMRI3	Input	Input pins for the counter reset
Serial communications interface (SClC)	• Asynchronous mode/clock synchronous mode		
	SCK0 to SCK11	I/O	Input/output pins for the clock
	RXD0 to RXD11	Input	Input pins for received data
	TXD0 to TXD11	Output	Output pins for transmitted data
	CTS0# to CTS11#	Input	Input pins for controlling the start of transmission and reception
	RTS0# to RTS11#	Output	Output pins for controlling the start of transmission and reception
	• Simple I <sup>2</sup> C mode		
	SSCL0 to SSCL11	I/O	Input/output pins for the I <sup>2</sup> C clock
	SSDA0 to SSDA11	I/O	Input/output pins for the I <sup>2</sup> C data
	• Simple SPI mode		
	SCK0 to SCK11	I/O	Input/output pins for the clock
	SMISO0 to SMISO11	I/O	Input/output pins for slave transmission of data
	SMOSI0 to SMOSI11	I/O	Input/output pins for master transmission of data
	SS0# to SS11#	Input	Chip-select input pins

**Table 1.4 Pin Functions (4/5)**

Classifications	Pin Name	I/O	Description	
Serial communications interface (SCId)	• 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 <sup>2</sup> C mode			
	SSCL12	I/O	Input/output pin for the I <sup>2</sup> C clock	
	SSDA12	I/O	Input/output pin for the I <sup>2</sup> C 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			
	RDX12	Input	Input pin for received data	
	TXDX12	Output	Output pin for transmitted data	
	SIOX12	I/O	Input/output pin for received or transmitted data	
	I <sup>2</sup> C bus interface	SCL0[FM+], SCL1 to SCL3	I/O	Input/output pins for clocks. Bus can be directly driven by the N-channel open drain
		SDA0[FM+], SDA1 to SDA3	I/O	Input/output pins for data. Bus can be directly driven by the N-channel open drain
USB 2.0 function module	VCC_USB	Input	Power supply pin	
	VSS_USB	Input	Ground pin	
	USB0_DP	I/O	Inputs or outputs D+ data for the USB bus	
	USB0_DM	I/O	Inputs or outputs D- data for the USB bus	
	USB0_DPUPE	Output	Pull-up pin	
	USB0_VBUS	Input	Input pin for detection of connection and disconnection of the USB cable	
CAN module	CRX0 to CRX2	Input	Input pins	
	CTX0 to CTX2	Output	Output pins	
Serial peripheral interface	RSPCKA, RSPCKB, RSPCKC	I/O	Clock input/output pins	
	MOSIA, MOSIB, MOSIC	I/O	Inputs or outputs data output from the master	
	MISOA, MISOB, MISOC	I/O	Inputs or outputs data output from the slave	
	SSLA0, SSLB0, SSLC0	I/O	Input or output pins for slave selection	
	SSLA1 to SSLA3, SSLB1 to SSLB3, SSLC1 to SSLC3	Output	Output pins for slave selection	
	IEBus controller	IERXD	Input	Input pin for data reception
IETXD		Output	Output pin for data transmission	
Realtime clock	RTCOUT	Output	Output pin for 1-Hz clock	
	RTCIC0 to RTCIC2	Input	Time capture event input pin	
12-bit A/D converter	AN000 to AN020	Input	Input pins for the analog signals to be processed by the A/D converter	
	ADTRG0#	Input	Input pins for the external trigger signals that start the A/D conversion	

**Table 1.4 Pin Functions (5/5)**

Classifications	Pin Name	I/O	Description
10-bit A/D converter	AN0 to AN7	Input	Input pins for the analog signals to be processed by the A/D converter
	ANEX0	Output	Extended analog output pin
	ANEX1	Input	Extended analog input pin
	ADTRG#	Input	Input pins for the external trigger signals that start the A/D conversion
D/A converter	DA0, DA1	Output	Output pins for the analog signals to be processed by the D/A converter
Analog power supply	AVCC0	Input	Analog voltage supply pin for the 12-bit A/D converter. Connect this pin to VCC if the 12-bit A/D converter is not to be used
	AVSS0	Input	Analog ground pin for the 12-bit A/D converter. Connect this pin to VSS if the 12-bit A/D converter is not to be used
	VREFH0	Input	Analog reference voltage supply pin for the 12-bit A/D converter. Connect this pin to VCC if the 12-bit A/D converter is not to be used
	VREFL0	Input	Analog reference ground pin for the 12-bit A/D converter. Connect this pin to VSS if the 12-bit A/D converter is not to be used
	VREFH	Input	Reference voltage input pin for the 10-bit A/D converter and D/A converter. This is used as the analog power supply for the respective modules. Connect this pin to VCC if neither the 10-bit A/D converter nor the D/A converter is in use
	VREFL	Input	Reference ground pin for the 10-bit A/D converter and D/A converter. This is used as the analog ground for the respective modules. Set this pin to the same potential as the VSS pin
I/O ports	P00 to P03, P05, P07	I/O	6-bit input/output pins
	P10 to P17	I/O	8-bit input/output pins
	P20 to P27	I/O	8-bit input/output pins
	P30 to P37	I/O	8-bit input/output pins (P35: input pin)
	P40 to P47	I/O	8-bit input/output pins
	P50 to P57	I/O	8-bit input/output pins
	P60 to P67	I/O	8-bit input/output pins
	P70 to P77	I/O	8-bit input/output pins
	P80 to P87	I/O	8-bit input/output pins
	P90 to P97	I/O	8-bit input/output pins
	PA0 to PA7	I/O	8-bit input/output pins
	PB0 to PB7	I/O	8-bit input/output pins
	PC0 to PC7	I/O	8-bit input/output pins
	PD0 to PD7	I/O	8-bit input/output pins
	PE0 to PE7	I/O	8-bit input/output pins
	PF0 to PF5	I/O	6-bit input/output pins
	PG0 to PG7	I/O	8-bit input/output pins
	PH4, PH5	I/O	2-bit input/output pins
	PJ3, PJ5	I/O	2-bit input/output pins
	PK0 to PK7	I/O	8-bit input/output pins
PL0 to PL4	I/O	5-bit input/output pins	

### 1.5 Pin Assignments

Figure 1.3 to Figure 1.10 show the pin assignments. Table 1.5 to Table 1.11 show the lists of pins and pin functions.

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R			
15	PE2	PE3	P70	P65	P67	VSS	VCC	PG7	PA6	PB0	P72	PB4	PL0	PL1	PC1	15		
14	PE1	PE0	PK4	PE7	PG3	PA0	PA1	PA2	PA7	PK7	PB1	PB5	P73	P75	P74	14		
13	P63	P64	PE4	PK5	PG2	PG4	PG6	PA3	PK6	P71	PB3	PB7	PC0	PC2	P76	13		
12	P60	PK3	P62	PE5	PE6	P66	PG5	PA4	PA5	PB2	PB6	P77	PC3	PC4	P80	12		
11	PD6	PG1	PK2	P61	RX630 Group PTLG0177KA-A (177-Pin TFLGA) (Upper perspective view)								P81	P82	PC6	VCC	11	
10	P97	PD4	PG0	PD7									PC5	PC7	P83	VSS	10	
9	PK0	P96	PD3	PD5									P50	P51	P52	P84	9	
8	P94	PD1	PD2	PK1									P53	PL2	PL3	PL4	8	
7	VSS	P92	PD0	P95									P54	P55	VSS_USB	USB0_DP	7	
6	VCC	P91	P90	P93									P56	P57	VCC_USB	USB0_DM	6	
5	P46	P47	P45	P44									NC	P13	P12	P10	P11	5
4	P42	P41	P43	P00									VSS	BSCANP	PF4	P35	PF3	PF1
3	VREFL0	P40	VREFH0	P03	PF5	PJ3	MD	RES#	P34	PF2	PF0	P24	P22	P87	P16	3		
2	AVCC0	P07	VREFH	P02	EMLE	VCL	XCOUT	VSS	VCC	P32	P30	P26	P23	P17	P20	2		
1	AVSS0	P05	VREFL	P01	PJ5	VBATT	XCIN	XTAL	EXTAL	P33	P31	P27	PH5	PH4	P21	1		
	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R			

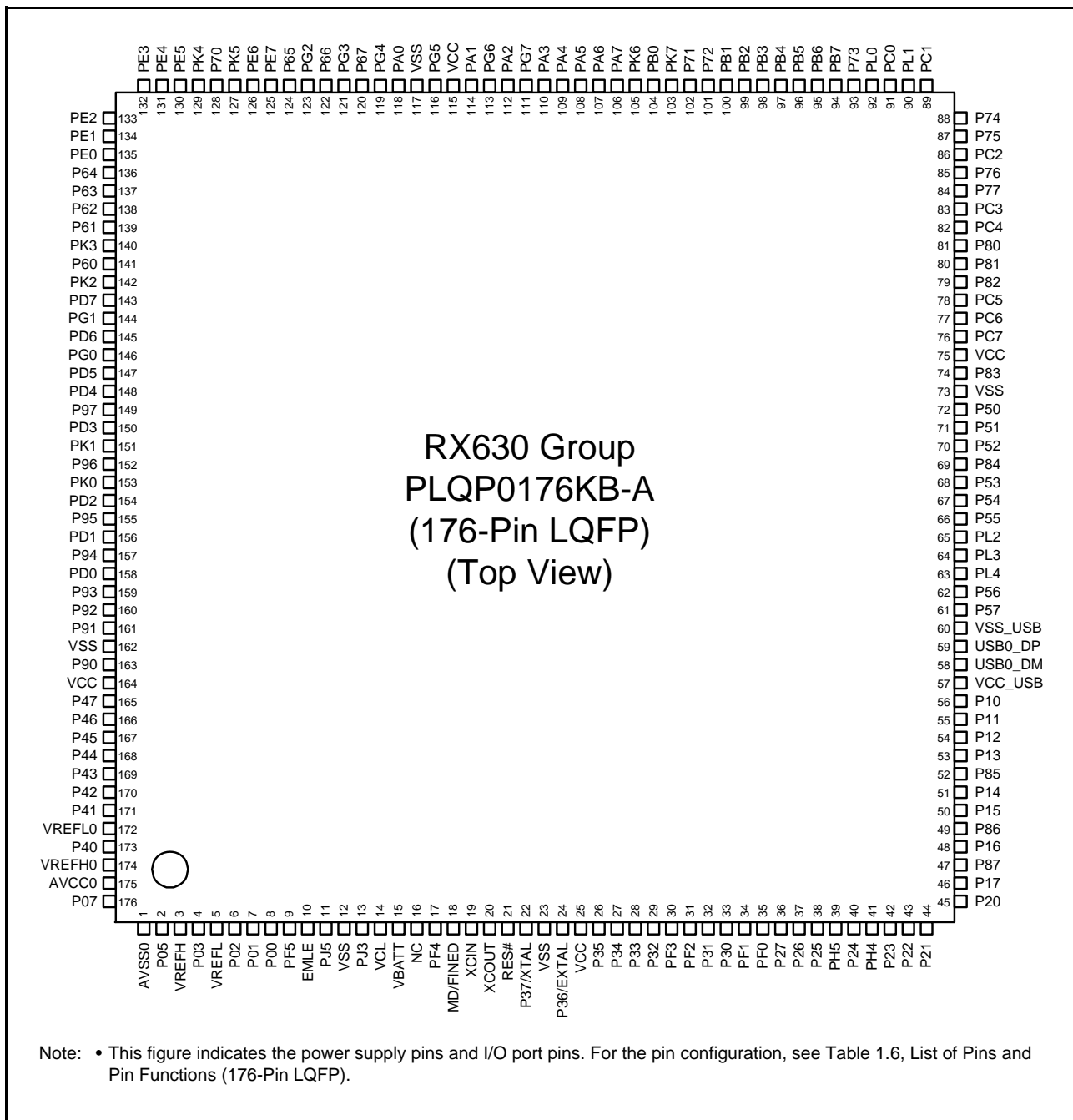
Note: • This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.5, List of Pins and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA).

**Figure 1.3 Pin Assignment (177-Pin TFLGA)**

	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R		
15	PE2	PE3	P70	P65	P67	VSS	VCC	PG7	PA6	PB0	P72	PB4	PL0	PL1	PC1	15	
14	PE1	PE0	PK4	PE7	PG3	PA0	PA1	PA2	PA7	PK7	PB1	PB5	P73	P75	P74	14	
13	P63	P64	PE4	PK5	PG2	PG4	PG6	PA3	PK6	P71	PB3	PB7	PC0	PC2	P76	13	
12	P60	PK3	P62	PE5	PE6	P66	PG5	PA4	PA5	PB2	PB6	P77	PC3	PC4	P80	12	
11	PD6	PG1	PK2	P61	<b>RX630 Group                      PLBG0176GA-A                      (176-Pin LFBGA)                      (Upper perspective view)</b>								P81	P82	PC6	VCC	11
10	P97	PD4	PG0	PD7									PC5	PC7	P83	VSS	10
9	PK0	P96	PD3	PD5									P50	P51	P52	P84	9
8	P94	PD1	PD2	PK1									P53	PL2	PL3	PL4	8
7	VSS	P92	PD0	P95									P54	P55	VSS_USB	USB0_DP	7
6	VCC	P91	P90	P93									P56	P57	VCC_USB	USB0_DM	6
5	P46	P47	P45	P44		P13	P12	P10	P11	5							
4	P42	P41	P43	P00	VSS	BSCANP	PF4	P35	PF3	PF1	P25	P86	P15	P14	P85	4	
3	VREFL0	P40	VREFH0	P03	PF5	PJ3	MD	RES#	P34	PF2	PF0	P24	P22	P87	P16	3	
2	AVCC0	P07	VREFH	P02	EMLE	VCL	XCOUT	VSS	VCC	P32	P30	P26	P23	P17	P20	2	
1	AVSS0	P05	VREFL	P01	PJ5	VBATT	XCIN	XTAL	EXTAL	P33	P31	P27	PH5	PH4	P21	1	
	A	B	C	D	E	F	G	H	J	K	L	M	N	P	R		

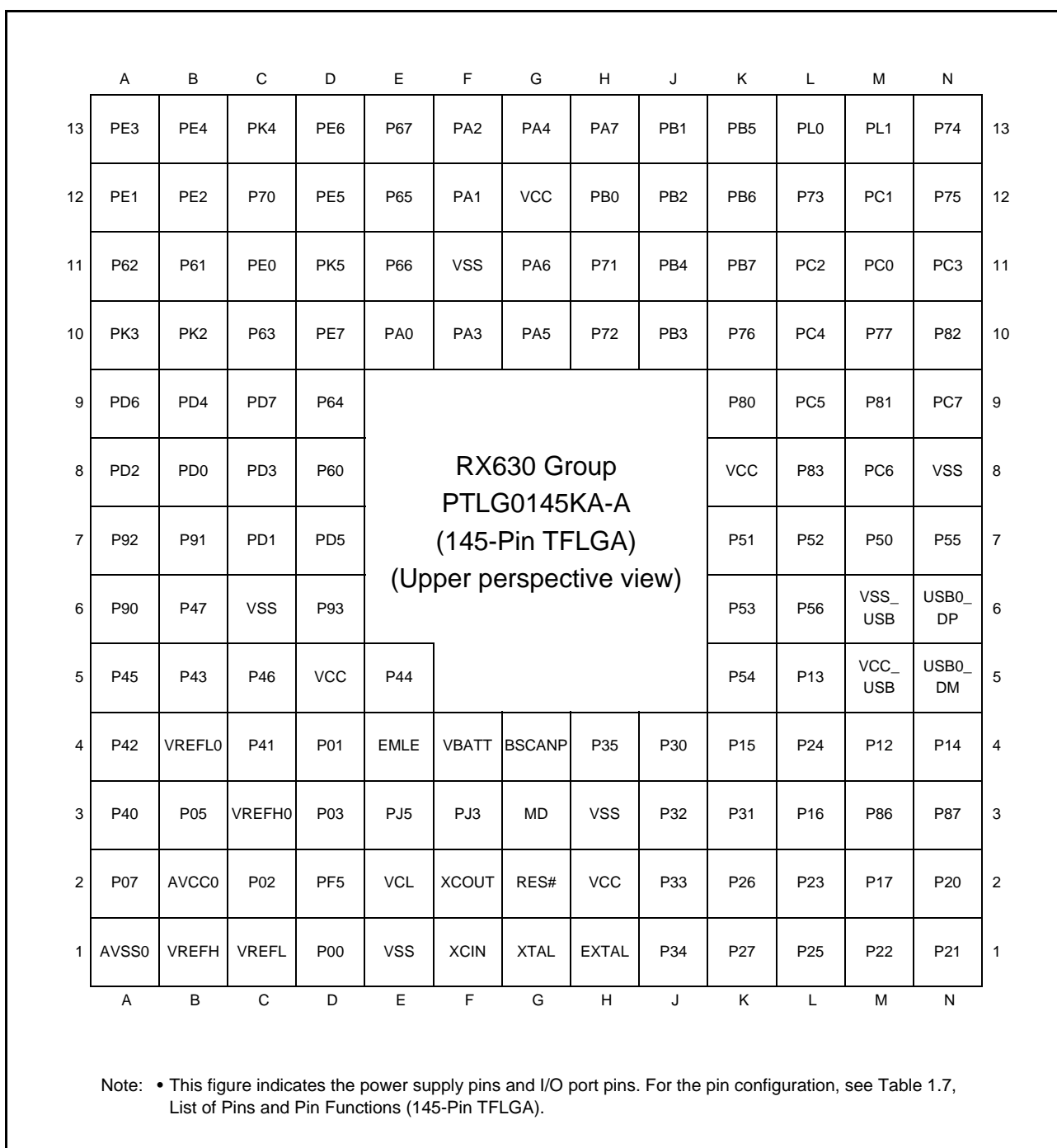
Note: • This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.5, List of Pins and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA).

**Figure 1.4 Pin Assignment (176-Pin LFBGA)**

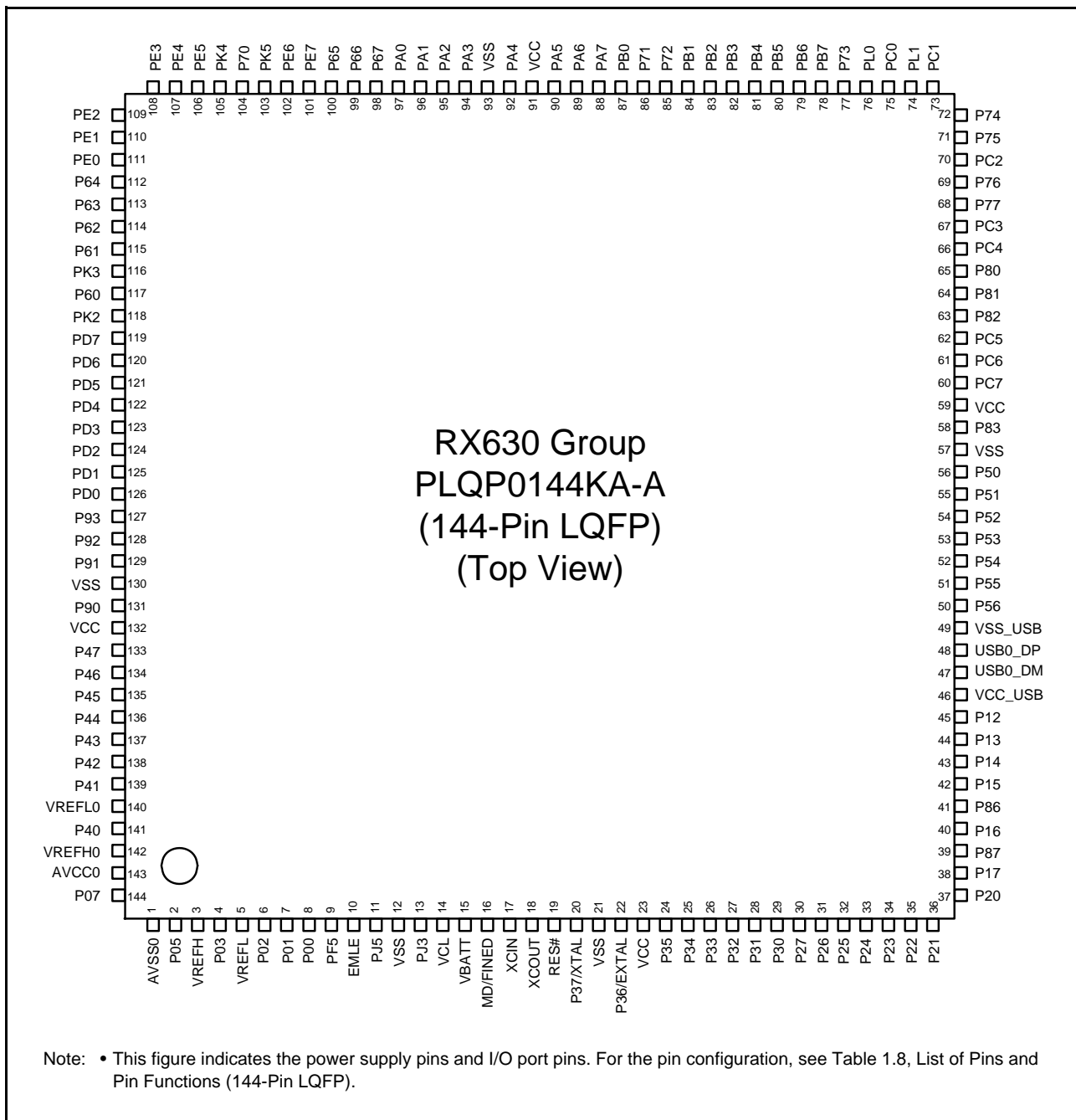


**Figure 1.5 Pin Assignment (176-Pin LQFP)**





**Figure 1.6 Pin Assignment (145-Pin TFLGA)**

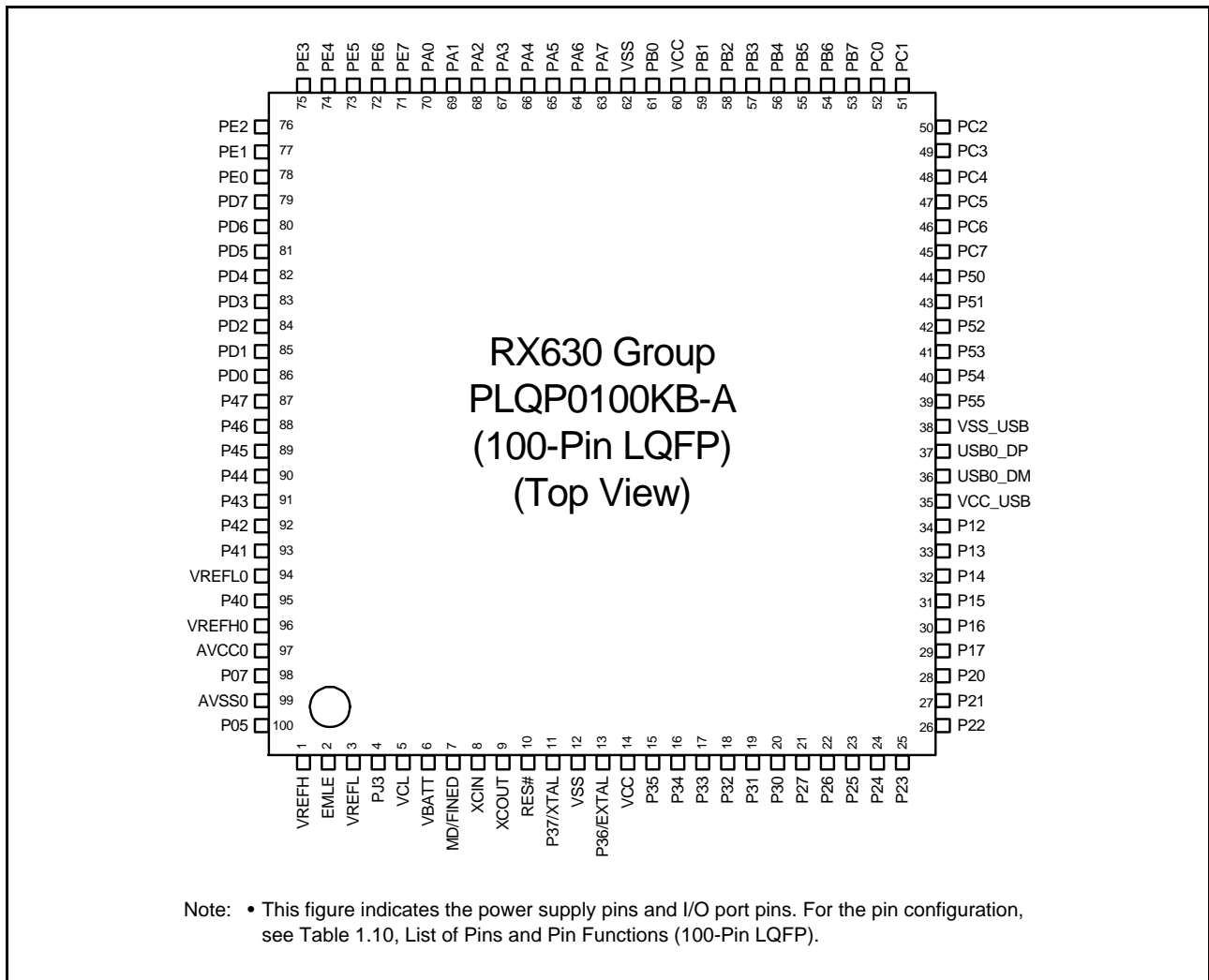


**RX630 Group  
PTLG0100KA-A (100-Pin TFLGA)  
(Top View)**

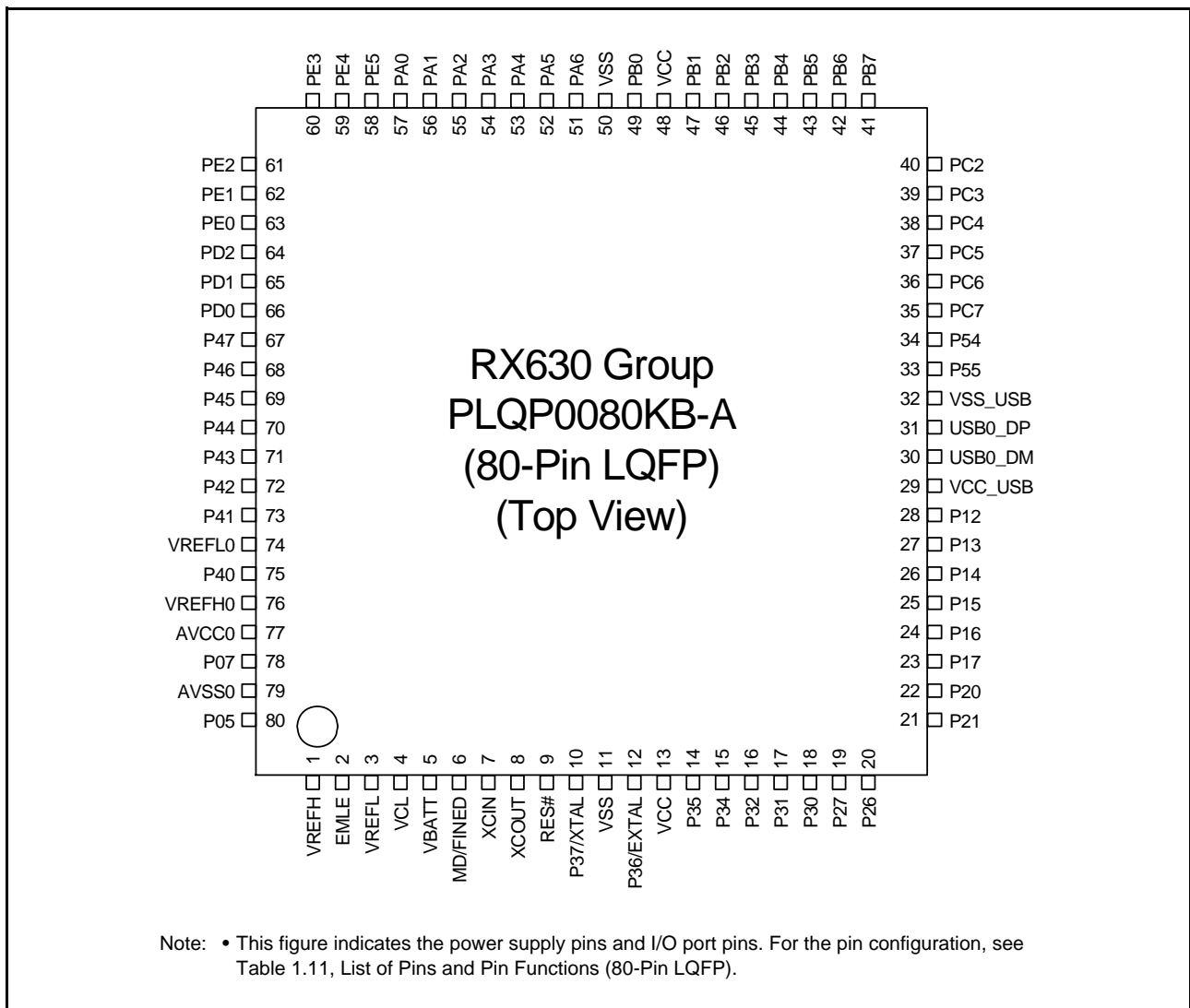
	A	B	C	D	E	F	G	H	J	K	
10	PE2	PE3	PE4	PA0	PA3	VSS	VCC	PB7	PC1	PC2	10
9	PE1	PD7	PE5	PA1	PA5	PA7	PB1	PB6	PC0	PC3	9
8	PE0	PD6	PD5	PE7	PA4	PB0	PB4	PC6	PC4	PC5	8
7	PD4	PD3	PD2	PE6	PA6	PB2	PB5	PC7	P50	P51	7
6	PD0	PD1	P47	P46	PA2	PB3	P52	P54	VCC_ USB	USB0_ DP	6
5	P43	P44	P42	P45	P41	P12	P53	P55	VSS_ USB	USB0_ DM	5
4	VREFL0	P40	VREFH0	VBATT	P34	P32	P27	P15	P13	P14	4
3	P07	AVCC0	PJ3	MD	RES#	P35	P30	P16	P17	P20	3
2	VREFH	AVSS0	VREFL	XCOUT	VSS	VCC	P31	P25	P21	P22	2
1	P05	EMLE	VCL	XCIN	XTAL	EXTAL	P33	P26	P24	P23	1
	A	B	C	D	E	F	G	H	J	K	

Note: • This figure indicates the power supply pins and I/O port pins. For the pin configuration, see Table 1.7, List of Pins and Pin Functions (145-Pin TFLGA).

**Figure 1.8 Pin Assignment (100-Pin TFLGA)**



**Figure 1.9 Pin Assignment (100-Pin LQFP)**



**Figure 1.10 Pin Assignment (80-Pin LQFP)**

**Table 1.5 List of Pins and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA) (1/5)**

Pin Number 177-Pin TFLGA 176-Pin LFBGA	Power Supply Clock System Control	I/O Port	Bus	Timer  (MTU, TPU, TMR, PPG, RTC, POE)	Communications  (SCiC, SCiD, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
A1	AVSS0						
A2	AVCC0						
A3	VREFL0						
A4		P42				IRQ10-DS	AN002
A5		P46				IRQ14-DS	AN006
A6	VCC						
A7	VSS						
A8		P94	A20/D20				
A9		PK0					
A10		P97	A23/D23				
A11		PD6	D6[A6/D6]	MTIC5V/POE1#	SSLC2	IRQ6	AN6
A12		P60	CS0#		SCK9		
A13		P63	CS3#				
A14		PE1	D9[A9/D9]	MTIOC4C/TIOCD9/PO18	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12/SSLB2/RSPCKB		ANEX1
A15		PE2	D10[A10/D10]	MTIOC4A/TIOCA9/PO23	RXD12/SMISO12/ SSCL12/RXDX12/SSLB3/ MOSIB	IRQ7-DS	AN0
B1		P05				IRQ13	DA1
B2		P07				IRQ15	ADTRG0#
B3		P40				IRQ8-DS	AN000
B4		P41				IRQ9-DS	AN001
B5		P47				IRQ15-DS	AN007
B6		P91	A17/D17		SCK7		AN015
B7		P92	A18/D18		RXD7/SMISO7/SSCL7		AN016
B8		PD1	D1[A1/D1]	MTIOC4B/TIOCB7/ TCLKG	MOSIC/CTX0	IRQ1	AN009
B9		P96	A22/D22				
B10		PD4	D4[A4/D4]	POE3#	SSLC0	IRQ4	AN012
B11		PG1	D25				
B12		PK3			RXD9/SMISO9/SSCL9		
B13		P64	CS4#				
B14		PE0	D8[A8/D8]	TIOCC9	SCK12/SSLB1		ANEX0
B15		PE3	D11[A11/D11]	MTIOC4B/TIOCB9/PO26/ POE8#	CTS12#/RTS12#/SS12#/ MISOB		AN1
C1	VREFL						
C2	VREFH						
C3	VREFH0						
C4		P43				IRQ11-DS	AN003
C5		P45				IRQ13-DS	AN005
C6		P90	A16/D16		TXD7/SMOSI7/SSDA7		AN014
C7		PD0	D0[A0/D0]	TIOCA7		IRQ0	AN008
C8		PD2	D2[A2/D2]	MTIOC4D/TIOCA8	MISOC/CRX0	IRQ2	AN010
C9		PD3	D3[A3/D3]	TIOCB8/TCLKH/POE8#	RSPCKC	IRQ3	AN011
C10		PG0	D24				
C11		PK2			TXD9/SMOSI9/SSDA9		
C12		P62	CS2#				

Table 1.5 List of Pins and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA) (2/5)

Pin Number	Power Supply Clock System Control	I/O Port	Bus	Timer  (MTU, TPU, TMR, PPG, RTC, POE)	Communications  (SCiC, SCiD, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
C13		PE4	D12[A12/D12]	MTIOC4D/MTIOC1A/ TIOCA10/PO28	SSLB0		AN2
C14		PK4			RXD4/SMISO4/SSCL4		
C15		P70			SCK4		
D1		P01		TMCI0	RXD6/SMISO6/SSCL6	IRQ9	AN019
D2		P02		TMCI1	SCK6	IRQ10	AN020
D3		P03				IRQ11	DA0
D4		P00		TMRI0	TXD6/SMOSI6/SSDA6	IRQ8	AN018
D5		P44				IRQ12-DS	AN004
D6		P93	A19/D19		CTS7#/RTS7#/SS7#		AN017
D7		P95	A21/D21				
D8		PK1					
D9		PD5	D5[A5/D5]	MTIC5W/POE2#	SSLC1	IRQ5	AN013
D10		PD7	D7[A7/D7]	MTIC5U/POE0#	SSLC3	IRQ7	AN7
D11		P61	CS1#		CTS9#/RTS9#/SS9#		
D12		PE5	D13[A13/D13]	MTIOC4C/MTIOC2B/ TIOCB10	RSPCKB	IRQ5	AN3
D13		PK5			TXD4/SMOSI4/SSDA4		
D14		PE7	D15[A15/D15]	TIOCB11	MISOB	IRQ7	AN5
D15		P65	CS5#				
E1		PJ5					
E2	EMLE						
E3		PF5				IRQ4	
E4	VSS						
E5*1	NC						
E12		PE6	D14[A14/D14]	TIOCA11	CTS4#/RTS4#/SS4#/ MOSIB	IRQ6	AN4
E13	TRDATA0	PG2	D26				
E14	TRDATA1	PG3	D27				
E15		P67	CS7#		CRX2*2	IRQ15	
F1	VBATT						
F2	VCL						
F3		PJ3		MTIOC3C	CTS6#/RTS6#/CTS0#/ RTS0#/SS6#/SS0#		
F4	BSCANP						
F12		P66	CS6#		CTX2*2		
F13	TRSYNC	PG4	D28				
F14		PA0	A0/BC0#	MTIOC4A/TIOCA0/PO16	SSLA1		
F15	VSS						
G1	XCIN						
G2	XCOUT						
G3	MD FINED						
G4	TRST#	PF4					
G12	TRCLK	PG5	D29				
G13	TRDATA2	PG6	D30				
G14		PA1	A1	MTIOC0B/MTCLKC/ TIOCB0/PO17	SCK5/SSLA2	IRQ11	
G15	VCC						

**Table 1.5 List of Pins and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA) (3/5)**

Pin Number 177-Pin TFLGA 176-Pin LFBGA	Power Supply Clock System Control	I/O Port	Bus	Timer  (MTU, TPU, TMR, PPG, RTC, POE)	Communications  (SCiC, SCiD, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
H1	XTAL	P37					
H2	VSS						
H3	RES#						
H4		P35				NMI	
H12		PA4	A4	MTIC5U/MTCLKA/ TIOCA1/TMRI0/PO20	TXD5/SMOSI5/SSDA5/ SSLA0	IRQ5-DS	
H13		PA3	A3	MTIOC0D/MTCLKD/ TIOCD0/TCLKB/PO19	RXD5/SMISO5/SSCL5	IRQ6-DS	
H14		PA2	A2	PO18	RXD5/SMISO5/SSCL5/ SSLA3		
H15	TRDATA3	PG7	D31				
J1	EXTAL	P36					
J2	VCC						
J3		P34		MTIOC0A/TMCI3/PO12/ POE2#	SCK6/SCK0	IRQ4	
J4	TMS	PF3					
J12		PA5	A5	TIOCB1/PO21	RSPCKA		
J13		PK6					
J14		PA7	A7	TIOCB2/PO23	MISOA		
J15		PA6	A6	MTIC5V/MTCLKB/ TIOCA2/TMCI3/PO22/ POE2#	CTS5#/RTS5#/SS5#/ MOSIA		
K1		P33		MTIOC0D/TIOCD0/ TMRI3/PO11/POE3#	RXD6/RXD0/SMISO6/ SMISO0/SSCL6/SSCL0/ CRX0	IRQ3-DS	
K2		P32		MTIOC0C/TIOCC0/ TMO3/PO10/RTCOUT/ RTCIC2	TXD6/TXD0/SMOSI6/ SMOSI0/SSDA6/SSDA0/ CTX0	IRQ2-DS	
K3	TDI	PF2			RXD1/SMISO1/SSCL1		
K4	TCK/FINEC	PF1			SCK1		
K12		PB2	A10	TIOCC3/TCLKC/PO26	CTS4#/RTS4#/CTS6#/ RTS6#/SS4#/SS6#		
K13		P71	CS1#				
K14		PK7					
K15		PB0	A8	MTIC5W/TIOCA3/PO24	RXD4/RXD6/SMISO4/ SMISO6/SSCL4/SSCL6/ RSPCKA	IRQ12	
L1		P31		MTIOC4D/TMCI2/PO9/ RTCIC1	CTS1#/RTS1#/SS1#/ SSLB0	IRQ1-DS	
L2		P30		MTIOC4B/TMRI3/PO8/ RTCIC0/POE8#	RXD1/SMISO1/SSCL1/ MISOB	IRQ0-DS	
L3	TDO	PF0			TXD1/SMOSI1/SSDA1		
L4		P25	CS5#	MTIOC4C/MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/SSCL3		ADTRG0#
L12		PB6	A14	MTIOC3D/TIOCA5/PO30	RXD9/SMISO9/SSCL9		
L13		PB3	A11	MTIOC0A/MTIOC4A/ TIOCD3/TCLKD/TMO0/ PO27/POE3#	SCK4/SCK6		
L14		PB1	A9	MTIOC0C/MTIOC4C/ TIOCB3/TMCI0/PO25	TXD4/TXD6/SMOSI4/ SMOSI6/SSDA4/SSDA6	IRQ4-DS	
L15		P72	CS2#				
M1		P27	CS7#	MTIOC2B/TMCI3/PO7	SCK1/RSPCKB		
M2		P26	CS6#	MTIOC2A/TMO1/PO6	TXD1/CTS3#/RTS3#/ SMOSI1/SS3#/SSDA1/ MOSIB		



**Table 1.5 List of Pins and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA) (4/5)**

Pin Number	Power Supply Clock System Control	I/O Port	Bus	Timer  (MTU, TPU, TMR, PPG, RTC, POE)	Communications  (SCiC, SCiD, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
M3		P24	CS4#	MTIOC4A/MTCLKA/ TIOCB4/TMRI1/PO4	SCK3		
M4		P86		TIOCA0			
M5		P13		MTIOC0B/TIOCA5/TMO3/ PO13	TXD2/SMOSI2/SSDA2/ SDA0[FM+]	IRQ3	ADTRG#
M6		P56	WR2#/BC2#	MTIOC3C/TIOCA1			
M7		P54	ALE	MTIOC4B/TMCI1	CTS2#/RTS2#/S2#/CTX1		
M8	BCLK	P53*3					
M9		P50	WR0#/WR#		TXD2/SMOSI2/SSDA2/ SSLB1		
M10		PC5	A21/CS2#/ WAIT#	MTIOC3B/MTCLKD/ TIOCD6/TCLKF/TMRI2/ PO29	SCK8/RSPCKA		
M11		P81		MTIOC3D/PO27	RXD10/SMISO10/SSCL10		
M12		P77	CS7#	PO23	TXD11/SMOSI11/SSDA11		
M13		PB7	A15	MTIOC3B/TIOCB5/PO31	TXD9/SMOSI9/SSDA9		
M14		PB5	A13	MTIOC2A/MTIOC1B/ TIOCB4/TMRI1/PO29/ POE1#	SCK9		
M15		PB4	A12	TIOCA4/PO28	CTS9#/RTS9#/SS9#		
N1		PH5					
N2		P23		MTIOC3D/MTCLKD/ TIOCD3/PO3	TXD3/CTS0#/RTS0#/ SMOSI3/SS0#/SSDA3		
N3		P22		MTIOC3B/MTCLKC/ TIOCC3/TMO0/PO2	SCK0		
N4		P15		MTIOC0B/MTCLKB/ TIOCB2/TCLKB/TMCI2/ PO13	RXD1/SCK3/SMISO1/ SSCL1/CRX1-DS	IRQ5	
N5		P12		MTIC5U/TMCI1	RXD2/SMISO2/SSCL2/ SCL0[FM+]	IRQ2	
N6		P57	WAIT#/WR3#/ BC3#				
N7		P55	WAIT#	MTIOC4D/TMO3	CRX1/	IRQ10	
N8		PL2					
N9		P51	WR1#/BC1#/ WAIT#		SCK2/SSLB2		
N10		PC7	A23/CS0#	MTIOC3A/MTCLKB/ TIOCB6/TMO2/PO31	TXD8/SMOSI8/SSDA8/ MISOA	IRQ14	
N11		P82		MTIOC4A/PO28	TXD10/SMOSI10/SSDA10		
N12		PC3	A19	MTIOC4D/TCLKB/PO24	TXD5/SMOSI5/SSDA5/ IETXD		
N13		PC0	A16	MTIOC3C/TCLKC/PO17	CTS5#/RTS5#/SS5#/ SSLA1/SCL3	IRQ14	
N14		P73	CS3#	PO16			
N15		PL0					
P1		PH4					
P2		P17		MTIOC3A/MTIOC3B/ TIOCB0/TCLKD/TMO1/ PO15/POE8#	SCK1/TXD3/SMOSI3/ SSDA3/MISOA/SDA2-DS/ IETXD	IRQ7	ADTRG#
P3		P87		TIOCA2			
P4		P14		MTIOC3A/MTCLKA/ TIOCB5/TCLKA/TMRI2/ PO15	CTS1#/RTS1#/SS1#/ CTX1/USB0_DPUPE	IRQ4	
P5		P10		MTIC5W/TMRI3		IRQ0	
P6	VCC_USB						

**Table 1.5 List of Pins and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA) (5/5)**

Pin Number	Power Supply Clock System Control	I/O Port	Bus	Timer  (MTU, TPU, TMR, PPG, RTC, POE)	Communications  (SC1c, SC1d, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
P7	VSS_USB						
P8		PL3					
P9		P52	RD#		RXD2/SMISO2/SSCL2/ SSLB3		
P10		P83		MTIOC4C	CTS10#/RTS10#/SS10#		
P11		PC6	A22/CS1#	MTIOC3C/MTCLKA/ TIOCA6/TMC12/PO30	RXD8/SMISO8/SSCL8/ MOSIA	IRQ13	
P12		PC4	A20/CS3#	MTIOC3D/MTCLKC/ TIOCC6/TCLKE/TMC11/ PO25/POE0#	SCK5/CTS8#/RTS8#/ SS8#/SSLA0		
P13		PC2	A18	MTIOC4B/TCLKA/PO21	RXD5/SMISO5/SSCL5/ SSLA3/IERXD		
P14		P75	CS5#	PO20	SCK11		
P15		PL1					
R1		P21		MTIOC1B/TIOCA3/ TMC10/PO1	RXD0/SMISO0/SSCL0/ SCL1	IRQ9	
R2		P20		MTIOC1A/TIOCB3/ TMR10/PO0	TXD0/SMOSI0/SSDA0/ SDA1	IRQ8	
R3		P16		MTIOC3C/MTIOC3D/ TIOCB1/TCLKC/TMO2/ PO14/RTCOU	TXD1/RXD3/SMOSI1/ SMISO3/SSDA1/SSCL3/ MOSIA/SCL2-DS/IERXD/ USB0_VBUS	IRQ6	ADTRG0#
R4		P85					
R5		P11		MTIC5V/TMC13	SCK2	IRQ1	
R6					USB0_DM		
R7					USB0_DP		
R8		PL4					
R9		P84					
R10	VSS						
R11	VCC						
R12		P80		MTIOC3B/PO26	SCK10		
R13		P76	CS6#	PO22	RXD11/SMISO11/SSCL11		
R14		P74	CS4#	PO19	CTS11#/RTS11#/SS11#		
R15		PC1	A17	MTIOC3A/TCLKD/PO18	SCK5/SSLA2/SDA3	IRQ12	

Note 1. The 176-pin LFBGA does not include the E5 pin.

Note 2. Enabled only for the ROM capacity: 2 MB/1.5 MB

Note 3. The BCLK function is multiplexed with the I/O port function for pin P53, so the port function is not available if the external bus is enabled.

Table 1.6 List of Pins and Pin Functions (176-Pin LQFP) (1/5)

Pin Number 176-Pin LQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SClC, SClD, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
1	AVSS0						
2		P05				IRQ13	DA1
3	VREFH						
4		P03				IRQ11	DA0
5	VREFL						
6		P02		TMC1	SCK6	IRQ10	AN020
7		P01		TMC10	RXD6/SMISO6/SSCL6	IRQ9	AN019
8		P00		TMR10	TXD6/SMOSI6/SSDA6	IRQ8	AN018
9		PF5				IRQ4	
10	EMLE						
11		PJ5					
12	VSS						
13		PJ3		MTIOC3C	CTS6#/RTS6#/CTS0#/ RTS0#/SS6#/SS0#		
14	VCL						
15	VBATT						
16	NC						
17	TRST#	PF4					
18	MD/FINED						
19	XCIN						
20	XCOUT						
21	RES#						
22	XTAL	P37					
23	VSS						
24	EXTAL	P36					
25	VCC						
26		P35				NMI	
27		P34		MTIOC0A/TMC13/PO12/ POE2#	SCK6/SCK0	IRQ4	
28		P33		MTIOC0D/TIOCD0/ TMR13/PO11/POE3#	RXD6/RXD0/SMISO6/ SMISO0/SSCL6/SSCL0/ CRX0	IRQ3-DS	
29		P32		MTIOC0C/TIOCC0/TMO3/ PO10/RTCOU7/RTCIC2	TXD6/TXD0/SMOSI6/ SMOSI0/SSDA6/SSDA0/ CTX0	IRQ2-DS	
30	TMS	PF3					
31	TDI	PF2			RXD1/SMISO1/SSCL1		
32		P31		MTIOC4D/TMC12/PO9/ RTCIC1	CTS1#/RTS1#/SS1#/ SSLB0	IRQ1-DS	
33		P30		MTIOC4B/TMR13/PO8/ RTCIC0/POE8#	RXD1/SMISO1/SSCL1/ MISOB	IRQ0-DS	
34	TCK/FINEC	PF1			SCK1		
35	TDO	PF0			TXD1/SMOSI1/SSDA1		
36		P27	CS7#	MTIOC2B/TMC13/PO7	SCK1/RSPCKB		
37		P26	CS6#	MTIOC2A/TMO1/PO6	TXD1/CTS3#/RTS3#/ SMOSI1/SS3#/SSDA1/ MOSIB		
38		P25	CS5#	MTIOC4C/MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/SSCL3		ADTRG0#
39		PH5					
40		P24	CS4#	MTIOC4A/MTCLKA/ TIOCB4/TMR11/PO4	SCK3		

Table 1.6 List of Pins and Pin Functions (176-Pin LQFP) (2/5)

Pin Number 176-Pin LQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SClC, SClD, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
41		PH4					
42		P23		MTIOC3D/MTCLKD/ TIOCD3/PO3	TXD3/CTS0#/RTS0#/ SMOSI3/SS0#/SSDA3		
43		P22		MTIOC3B/MTCLKC/ TIOCC3/TMO0/PO2	SCK0		
44		P21		MTIOC1B/TIOCA3/ TMCI0/PO1	RXD0/SMISO0/SSCL0/ SCL1	IRQ9	
45		P20		MTIOC1A/TIOCB3/ TMRI0/PO0	TXD0/SMOSI0/SSDA0/ SDA1	IRQ8	
46		P17		MTIOC3A/MTIOC3B/ TIOCB0/TCLKD/TMO1/ PO15/POE8#	SCK1/TXD3/SMOSI3/ SSDA3/MISOA/SDA2-DS/ IETXD	IRQ7	ADTRG#
47		P87		TIOCA2			
48		P16		MTIOC3C/MTIOC3D/ TIOCB1/TCLKC/TMO2/ PO14/RTCOUT	TXD1/RXD3/SMOSI1/ SMISO3/SSDA1/SSCL3/ MOSIA/SCL2-DS/IERXD/ USB0_VBUS	IRQ6	ADTRG0#
49		P86		TIOCA0			
50		P15		MTIOC0B/MTCLKB/ TIOCB2/TCLKB/TMCI2/ PO13	RXD1/SCK3/SMISO1/ SSCL1/CRX1-DS	IRQ5	
51		P14		MTIOC3A/MTCLKA/ TIOCB5/TCLKA/TMRI2/ PO15	CTS1#/RTS1#/SS1#/ CTX1/USB0_DPUPE	IRQ4	
52		P85					
53		P13		MTIOC0B/TIOCA5/ TMO3/PO13	TXD2/SMOSI2/SSDA2/ SDA0[FM+]	IRQ3	ADTRG#
54		P12		MTIC5U/TMCI1	RXD2/SMISO2/SSCL2/ SCL0[FM+]	IRQ2	
55		P11		MTIC5V/TMCI3	SCK2	IRQ1	
56		P10		MTIC5W/TMRI3		IRQ0	
57	VCC_USB						
58					USB0_DM		
59					USB0_DP		
60	VSS_USB						
61		P57	WAIT#/WR3#/ BC3#				
62		P56	WR2#/BC2#	MTIOC3C/TIOCA1			
63		PL4					
64		PL3					
65		PL2					
66		P55	WAIT#	MTIOC4D/TMO3	CRX1	IRQ10	
67		P54	ALE	MTIOC4B/TMCI1	CTS2#/RTS2#/SS2#/ CTX1		
68	BCLK	P53*1					
69		P84					
70		P52	RD#		RXD2/SMISO2/SSCL2/ SSLB3		
71		P51	WR1#/BC1#/ WAIT#		SCK2/SSLB2		
72		P50	WR0#/WR#		TXD2/SMOSI2/SSDA2/ SSLB1		
73	VSS						
74		P83		MTIOC4C	CTS10#/RTS10#/SS10#		
75	VCC						

Table 1.6 List of Pins and Pin Functions (176-Pin LQFP) (3/5)

Pin Number 176-Pin LQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SClC, SClD, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
76		PC7	A23/CS0#	MTIOC3A/MTCLKB/ TIOCB6/TMO2/PO31	TXD8/SMOSI8/SSDA8/ MISOA	IRQ14	
77		PC6	A22/CS1#	MTIOC3C/MTCLKA/ TIOCA6/TMCI2/PO30	RXD8/SMISO8/SSCL8/ MOSIA	IRQ13	
78		PC5	A21/CS2#/ WAIT#	MTIOC3B/MTCLKD/ TIOCD6/TCLKF/TMRI2/ PO29	SCK8/RSPCKA		
79		P82		MTIOC4A/PO28	TXD10/SMOSI10/SSDA10		
80		P81		MTIOC3D/PO27	RXD10/SMISO10/ SSCL10		
81		P80		MTIOC3B/PO26	SCK10		
82		PC4	A20/CS3#	MTIOC3D/MTCLKC/ TIOCC6/TCLKE/TMCI1/ PO25/POE0#	SCK5/CTS8#/RTS8#/ SS8#/SSLA0		
83		PC3	A19	MTIOC4D/TCLKB/PO24	TXD5/SMOSI5/SSDA5/ IETXD		
84		P77	CS7#	PO23	TXD11/SMOSI11/SSDA11		
85		P76	CS6#	PO22	RXD11/SMISO11/SSCL11		
86		PC2	A18	MTIOC4B/TCLKA/PO21	RXD5/SMISO5/SSCL5/ SSLA3/IERXD		
87		P75	CS5#	PO20	SCK11		
88		P74	CS4#	PO19	CTS11#/RTS11#/SS11#		
89		PC1	A17	MTIOC3A/TCLKD/PO18	SCK5/SSLA2/SDA3	IRQ12	
90		PL1					
91		PC0	A16	MTIOC3C/TCLKC/PO17	CTS5#/RTS5#/SS5#/ SSLA1/SCL3	IRQ14	
92		PL0					
93		P73	CS3#	PO16			
94		PB7	A15	MTIOC3B/TIOCB5/PO31	TXD9/SMOSI9/SSDA9		
95		PB6	A14	MTIOC3D/TIOCA5/PO30	RXD9/SMISO9/SSCL9		
96		PB5	A13	MTIOC2A/MTIOC1B/ TIOCB4/TMRI1/PO29/ POE1#	SCK9		
97		PB4	A12	TIOCA4/PO28	CTS9#/RTS9#/SS9#		
98		PB3	A11	MTIOC0A/MTIOC4A/ TIOCD3/TCLKD/TMO0/ PO27/POE3#	SCK4/SCK6		
99		PB2	A10	TIOCC3/TCLKC/PO26	CTS4#/RTS4#/CTS6#/ RTS6#/SS4#/SS6#		
100		PB1	A9	MTIOC0C/MTIOC4C/ TIOCB3/TMCI0/PO25	TXD4/TXD6/SMOSI4/ SMOSI6/SSDA4/SSDA6	IRQ4-DS	
101		P72	CS2#				
102		P71	CS1#				
103		PK7					
104		PB0	A8	MTIC5W/TIOCA3/PO24	RXD4/RXD6/SMISO4/ SMISO6/SSCL4/SSCL6/ RSPCKA	IRQ12	
105		PK6					
106		PA7	A7	TIOCB2/PO23	MISOA		
107		PA6	A6	MTIC5V/MTCLKB/ TIOCA2/TMCI3/PO22/ POE2#	CTS5#/RTS5#/SS5#/ MOSIA		
108		PA5	A5	TIOCB1/PO21	RSPCKA		
109		PA4	A4	MTIC5U/MTCLKA/ TIOCA1/TMRI0/PO20	TXD5/SMOSI5/SSDA5/ SSLA0	IRQ5-DS	

Table 1.6 List of Pins and Pin Functions (176-Pin LQFP) (4/5)

Pin Number 176-Pin LQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SC1c, SC1d, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
110		PA3	A3	MTIOC0D/MTCLKD/ TIOCD0/TCLKB/PO19	RXD5/SMISO5/SSCL5	IRQ6-DS	
111	TRDATA3	PG7	D31				
112		PA2	A2	PO18	RXD5/SMISO5/SSCL5/ SSLA3		
113	TRDATA2	PG6	D30				
114		PA1	A1	MTIOC0B/MTCLKC/ TIOCB0/PO17	SCK5/SSLA2	IRQ11	
115	VCC						
116	TRCLK	PG5	D29				
117	VSS						
118		PA0	A0/BC0#	MTIOC4A/TIOCA0/PO16	SSLA1		
119	TRSYNC	PG4	D28				
120		P67	CS7#		CRX2*2	IRQ15	
121	TRDATA1	PG3	D27				
122		P66	CS6#		CTX2*2		
123	TRDATA0	PG2	D26				
124		P65	CS5#				
125		PE7	D15[A15/D15]	TIOCB11	MISOB	IRQ7	AN5
126		PE6	D14[A14/D14]	TIOCA11	CTS4#/RTS4#/SS4#/ MOSIB	IRQ6	AN4
127		PK5			TXD4/SMOSI4/SSDA4		
128		P70			SCK4		
129		PK4			RXD4/SMISO4/SSCL4		
130		PE5	D13[A13/D13]	MTIOC4C/MTIOC2B/ TIOCB10	RSPCKB	IRQ5	AN3
131		PE4	D12[A12/D12]	MTIOC4D/MTIOC1A/ TIOCA10/PO28	SSLB0		AN2
132		PE3	D11[A11/D11]	MTIOC4B/TIOCB9/PO26/ POE8#	CTS12#/RTS12#/SS12#/ MISOB		AN1
133		PE2	D10[A10/D10]	MTIOC4A/TIOCA9/PO23	RXD12/SMISO12/ SSCL12/RDX12/SSLB3/ MOSIB	IRQ7-DS	AN0
134		PE1	D9[A9/D9]	MTIOC4C/TIOCD9/PO18	TXD12/SMOSI12/ SSDA12/TDX12/ SIOX12/SSLB2/RSPCKB		ANEX1
135		PE0	D8[A8/D8]	TIOCC9	SCK12/SSLB1		ANEX0
136		P64	CS4#				
137		P63	CS3#				
138		P62	CS2#				
139		P61	CS1#		CTS9#/RTS9#/SS9#		
140		PK3			RXD9/SMISO9/SSCL9		
141		P60	CS0#		SCK9		
142		PK2			TXD9/SMOSI9/SSDA9		
143		PD7	D7[A7/D7]	MTIC5U/POE0#	SSLC3	IRQ7	AN7
144		PG1	D25				
145		PD6	D6[A6/D6]	MTIC5V/POE1#	SSLC2	IRQ6	AN6
146		PG0	D24				
147		PD5	D5[A5/D5]	MTIC5W/POE2#	SSLC1	IRQ5	AN013
148		PD4	D4[A4/D4]	POE3#	SSLC0	IRQ4	AN012
149		P97	A23/D23				
150		PD3	D3[A3/D3]	TIOCB8/TCLKH/POE8#	RSPCKC	IRQ3	AN011

**Table 1.6 List of Pins and Pin Functions (176-Pin LQFP) (5/5)**

Pin Number 176-Pin LQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SClC, SClD, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
151		PK1					
152		P96	A22/D22				
153		PK0					
154		PD2	D2[A2/D2]	MTIOC4D/TIOCA8	MISOC/CRX0	IRQ2	AN010
155		P95	A21/D21				
156		PD1	D1[A1/D1]	MTIOC4B/TIOCB7/CLKG	MOSIC/CTX0	IRQ1	AN009
157		P94	A20/D20				
158		PD0	D0[A0/D0]	TIOCA7		IRQ0	AN008
159		P93	A19/D19		CTS7#/RTS7#/SS7#		AN017
160		P92	A18/D18		RXD7/SMISO7/SSCL7		AN016
161		P91	A17/D17		SCK7		AN015
162	VSS						
163		P90	A16/D16		TXD7/SMOSI7/SSDA7		AN014
164	VCC						
165		P47				IRQ15-DS	AN007
166		P46				IRQ14-DS	AN006
167		P45				IRQ13-DS	AN005
168		P44				IRQ12-DS	AN004
169		P43				IRQ11-DS	AN003
170		P42				IRQ10-DS	AN002
171		P41				IRQ9-DS	AN001
172	VREFLO						
173		P40				IRQ8-DS	AN000
174	VREFH0						
175	AVCC0						
176		P07				IRQ15	ADTRG0#

Note 1. The BCLK function is multiplexed with the I/O port function for pin P53, so the port function is not available if the external bus is enabled.

Note 2. Enabled only for the ROM capacity: 2 MB/1.5 MB

**Table 1.7 List of Pins and Pin Functions (145-Pin TFLGA) (1/4)**

Pin Number 145-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SCIC, SCID, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
A1	AVSS0						
A2		P07				IRQ15	ADTRG0#
A3		P40				IRQ8-DS	AN000
A4		P42				IRQ10-DS	AN002
A5		P45				IRQ13-DS	AN005
A6		P90	A16		TXD7/SMOSI7/SSDA7		AN014
A7		P92	A18		RXD7/SMISO7/SSCL7		AN016
A8		PD2	D2[A2/D2]	MTIOC4D/TIOCA8	MISOC/CRX0	IRQ2	AN010
A9		PD6	D6[A6/D6]	MTIC5V/POE1#	SSLC2	IRQ6	AN6
A10		PK3			RXD9/SMISO9/SSCL9		
A11		P62	CS2#				
A12		PE1	D9[A9/D9]	MTIOC4C/TIOCD9/PO18	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12/SSLB2/RSPCKB		ANEX1
A13		PE3	D11[A11/D11]	MTIOC4B/TIOCB9/PO26/ POE8#	CTS12#/RTS12#/SS12#/ MISOB		AN1
B1	VREFH						
B2	AVCC0						
B3		P05				IRQ13	DA1
B4	VREFLO						
B5		P43				IRQ11-DS	AN003
B6		P47				IRQ15-DS	AN007
B7		P91	A17		SCK7		AN015
B8		PD0	D0[A0/D0]	TIOCA7		IRQ0	AN008
B9		PD4	D4[A4/D4]	POE3#	SSLC0	IRQ4	AN012
B10		PK2			TXD9/SMOSI9/SSDA9		
B11		P61	CS1#		CTS9#/RTS9#/SS9#		
B12		PE2	D10[A10/D10]	MTIOC4A/TIOCA9/PO23	RXD12/SMISO12/ SSCL12/RXDX12/SSLB3/ MOSIB	IRQ7-DS	AN0
B13		PE4	D12[A12/D12]	MTIOC4D/MTIOC1A/ TIOCA10/PO28	SSLB0		AN2
C1	VREFL						
C2		P02		TMCI1	SCK6	IRQ10	AN020
C3	VREFH0						
C4		P41				IRQ9-DS	AN001
C5		P46				IRQ14-DS	AN006
C6	VSS						
C7		PD1	D1[A1/D1]	MTIOC4B/TIOCB7/ TCLKG	MOSIC/CTX0	IRQ1	AN009
C8		PD3	D3[A3/D3]	TIOCB8/TCLKH/POE8#	RSPCKC	IRQ3	AN011
C9		PD7	D7[A7/D7]	MTIC5U/POE0#	SSLC3	IRQ7	AN7
C10		P63	CS3#				
C11		PE0	D8[A8/D8]	TIOCC9	SCK12/SSLB1		ANEX0
C12		P70			SCK4		
C13		PK4			RXD4/SMISO4/SSCL4		
D1		P00		TMRI0	TXD6/SMOSI6/SSDA6	IRQ8	AN018
D2		PF5				IRQ4	
D3		P03				IRQ11	DA0
D4		P01		TMCI0	RXD6/SMISO6/SSCL6	IRQ9	AN019



Table 1.7 List of Pins and Pin Functions (145-Pin TFLGA) (2/4)

Pin Number 145-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SCIC, SCID, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
D5	VCC						
D6		P93	A19		CTS7#/RTS7#/SS7#		AN017
D7		PD5	D5[A5/D5]	MTIC5W/POE2#	SSLC1	IRQ5	AN013
D8		P60	CS0#		SCK9		
D9		P64	CS4#				
D10		PE7	D15[A15/D15]	TIOCB11	MISOB	IRQ7	AN5
D11		PK5			TXD4/SMOSI4/SSDA4		
D12		PE5	D13[A13/D13]	MTIOC4C/MTIOC2B/ TIOCB10	RSPCKB	IRQ5	AN3
D13		PE6	D14[A14/D14]	TIOCA11	CTS4#/RTS4#/SS4#/ MOSIB	IRQ6	AN4
E1	VSS						
E2	VCL						
E3		PJ5					
E4	EMLE						
E5		P44				IRQ12-DS	AN004
E10		PA0	A0/BC0#	MTIOC4A/TIOCA0/PO16	SSLA1		
E11		P66	CS6#		CTX2*1		
E12		P65	CS5#				
E13		P67	CS7#		CRX2*1	IRQ15	
F1	XCIN						
F2	XCOUT						
F3		PJ3		MTIOC3C	CTS6#/RTS6#/CTS0#/ RTS0#/SS6#/SS0#		
F4	VBATT						
F10		PA3	A3	MTIOC0D/MTCLKD/ TIOC0D/TCLKB/PO19	RXD5/SMISO5/SSCL5	IRQ6-DS	
F11	VSS						
F12		PA1	A1	MTIOC0B/MTCLKC/ TIOCB0/PO17	SCK5/SSLA2	IRQ11	
F13		PA2	A2	PO18	RXD5/SMISO5/SSCL5/ SSLA3		
G1	XTAL	P37					
G2	RES#						
G3	MD/FINED						
G4	BSCANP						
G10		PA5	A5	TIOCB1/PO21	RSPCKA		
G11		PA6	A6	MTIC5V/MTCLKB/ TIOCA2/TMC13/PO22/ POE2#	CTS5#/RTS5#/SS5#/ MOSIA		
G12	VCC						
G13		PA4	A4	MTIC5U/MTCLKA/ TIOCA1/TMRI0/PO20	TXD5/SMOSI5/SSDA5/ SSLA0	IRQ5-DS	
H1	EXTAL	P36					
H2	VCC						
H3	VSS						
H4		P35				NMI	
H10		P72	CS2#				
H11		P71	CS1#				
H12		PB0	A8	MTIC5W/TIOCA3/PO24	RXD4/RXD6/SMISO4/ SMISO6/SSCL4/SSCL6/ RSPCKA	IRQ12	

Table 1.7 List of Pins and Pin Functions (145-Pin TFLGA) (3/4)

Pin Number 145-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SCIC, SCID, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
H13		PA7	A7	TIOCB2/PO23	MISOA		
J1	TRST#	P34		MTIOC0A/TMCI3/PO12/ POE2#	SCK6/SCK0	IRQ4	
J2		P33		MTIOC0D/TIOCD0/ TMRI3/PO11/POE3#	RXD6/RXD0/SMISO6/ SMISO0/SSCL6/SSCL0/ CRX0	IRQ3-DS	
J3		P32		MTIOC0C/TIOCC0/TMO3/ PO10/RTCOU7/RTCIC2	TXD6/TXD0/SMOSI6/ SMOSI0/SSDA6/SSDA0/ CTX0	IRQ2-DS	
J4	TDI	P30		MTIOC4B/TMRI3/PO8/ RTCIC0/POE8#	RXD1/SMISO1/SSCL1/ MISOB	IRQ0-DS	
J10		PB3	A11	MTIOC0A/MTIOC4A/ TIOCD3/TCLKD/TMO0/ PO27/POE3#	SCK4/SCK6		
J11		PB4	A12	TIOCA4/PO28	CTS9#/RTS9#/SS9#		
J12		PB2	A10	TIOCC3/TCLKC/PO26	CTS4#/RTS4#/CTS6#/ RTS6#/SS4#/SS6#		
J13		PB1	A9	MTIOC0C/MTIOC4C/ TIOCB3/TMCI0/PO25	TXD4/TXD6/SMOSI4/ SMOSI6/SSDA4/SSDA6	IRQ4-DS	
K1	TCK/FINEC	P27	CS7#	MTIOC2B/TMCI3/PO7	SCK1/RSPCKB		
K2	TDO	P26	CS6#	MTIOC2A/TMO1/PO6	TXD1/CTS3#/RTS3#/ SMOSI1/SS3#/SSDA1/ MOSIB		
K3	TMS	P31		MTIOC4D/TMCI2/PO9/ RTCIC1	CTS1#/RTS1#/SS1#/ SSLB0	IRQ1-DS	
K4		P15		MTIOC0B/MTCLKB/ TIOCB2/TCLKB/TMCI2/ PO13	RXD1/SCK3/SMISO1/ SSCL1/CRX1-DS	IRQ5	
K5	TRDATA2	P54	ALE	MTIOC4B/TMCI1	CTS2#/RTS2#/SS2#/ CTX1		
K6		P53*2	BCLK				
K7		P51	WR1#/BC1#/ WAIT#		SCK2/SSLB2		
K8	VCC						
K9	TRDATA0	P80		MTIOC3B/PO26	SCK10		
K10		P76	CS6#	PO22	RXD11/SMISO11/SSCL11		
K11		PB7	A15	MTIOC3B/TIOCB5/PO31	TXD9/SMOSI9/SSDA9		
K12		PB6	A14	MTIOC3D/TIOCA5/PO30	RXD9/SMISO9/SSCL9		
K13		PB5	A13	MTIOC2A/MTIOC1B/ TIOCB4/TMRI1/PO29/ POE1#	SCK9		
L1		P25	CS5#	MTIOC4C/MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/SSCL3		ADTRG0#
L2		P23		MTIOC3D/MTCLKD/ TIOCD3/PO3	TXD3/CTS0#/RTS0#/ SMOSI3/SS0#/SSDA3		
L3		P16		MTIOC3C/MTIOC3D/ TIOCB1/TCLKC/TMO2/ PO14/RTCOU7	TXD1/RXD3/SMOSI1/ SMISO3/SSDA1/SSCL3/ MOSIA/SCL2-DS/IERXD/ USB0_VBUS	IRQ6	ADTRG0#
L4		P24	CS4#	MTIOC4A/MTCLKA/ TIOCB4/TMRI1/PO4	SCK3		
L5		P13		MTIOC0B/TIOCA5/TMO3/ PO13	TXD2/SMOSI2/SSDA2/ SDA0[FM+]	IRQ3	ADTRG#
L6		P56		MTIOC3C/TIOCA1			
L7		P52	RD#		RXD2/SMISO2/SSCL2/ SSLB3		
L8	TRCLK	P83		MTIOC4C	CTS10#/RTS10#/SS10#		

**Table 1.7 List of Pins and Pin Functions (145-Pin TFLGA) (4/4)**

Pin Number 145-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SClC, SClD, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
L9		PC5	A21/CS2#/ WAIT#	MTIOC3B/MTCLKD/ TIOCD6/TCLKF/TMRI2/ PO29	SCK8/RSPCKA		
L10		PC4	A20/CS3#	MTIOC3D/MTCLKC/ TIOCC6/TCLKE/TMCI1/ PO25/POE0#	SCK5/CTS8#/RTS8#/ SS8#/SSLA0		
L11		PC2	A18	MTIOC4B/TCLKA/PO21	RXD5/SMISO5/SSCL5/ SSLA3/IERXD		
L12		P73	CS3#	PO16			
L13		PL0					
M1		P22		MTIOC3B/MTCLKC/ TIOCC3/TMO0/PO2	SCK0		
M2		P17		MTIOC3A/MTIOC3B/ TIOCB0/TCLKD/TMO1/ PO15/POE8#	SCK1/TXD3/SMOSI3/ SSDA3/MISOA/SDA2-DS/ IETXD	IRQ7	ADTRG#
M3		P86		TIOCA0			
M4		P12		TMCI1	RXD2/SMISO2/SSCL2/ SCL0[FM+]	IRQ2	
M5	VCC_USB						
M6	VSS_USB						
M7		P50	WR0#/WR#		TXD2/SMOSI2/SSDA2/ SSLB1		
M8		PC6	A22/CS1#	MTIOC3C/MTCLKA/ TIOCA6/TMCI2/PO30	RXD8/SMISO8/SSCL8/ MOSIA	IRQ13	
M9	TRDATA1	P81		MTIOC3D/PO27	RXD10/SMISO10/SSCL10		
M10		P77	CS7#	PO23	TXD11/SMOSI11/SSDA11		
M11		PC0	A16	MTIOC3C/TCLKC/PO17	CTS5#/RTS5#/SS5#/ SSLA1/SCL3	IRQ14	
M12		PC1	A17	MTIOC3A/TCLKD/PO18	SCK5/SSLA2/SDA3	IRQ12	
M13		PL1					
N1		P21		MTIOC1B/TIOCA3/ TMCI0/PO1	RXD0/SMISO0/SSCL0/ SCL1	IRQ9	
N2		P20		MTIOC1A/TIOCB3/ TMRI0/PO0	TXD0/SMOSI0/SSDA0/ SDA1	IRQ8	
N3		P87		TIOCA2			
N4		P14		MTIOC3A/MTCLKA/ TIOCB5/TCLKA/TMRI2/ PO15	CTS1#/RTS1#/SS1#/ CTX1/USB0_DPUPE	IRQ4	
N5					USB0_DM		
N6					USB0_DP		
N7	TRDATA3	P55	WAIT#	MTIOC4D/TMO3	CRX1	IRQ10	
N8	VSS						
N9		PC7	A23/CS0#	MTIOC3A/MTCLKB/ TIOCB6/TMO2/PO31	TXD8/SMOSI8/SSDA8/ MISOA	IRQ14	
N10	TRSYNC	P82		MTIOC4A/PO28	TXD10/SMOSI10/SSDA10		
N11		PC3	A19	MTIOC4D/TCLKB/PO24	TXD5/SMOSI5/SSDA5/ IETXD		
N12		P75	CS5#	PO20	SCK11		
N13		P74	CS4#	PO19	CTS11#/RTS11#/SS11#		

Note 1. Enabled only for the ROM capacity: 2 MB/1.5 MB

Note 2. The BCLK function is multiplexed with the I/O port function for pin P53, so the port function is not available if the external bus is enabled.

**Table 1.8 List of Pins and Pin Functions (144-Pin LQFP) (1/4)**

Pin Number 144-Pin LQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SCIC, SCID, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
1	AVSS0						
2		P05				IRQ13	DA1
3	VREFH						
4		P03				IRQ11	DA0
5	VREFL						
6		P02		TMC1	SCK6	IRQ10	AN020
7		P01		TMC10	RXD6/SMISO6/SSCL6	IRQ9	AN019
8		P00		TMR10	TXD6/SMOSI6/SSDA6	IRQ8	AN018
9		PF5				IRQ4	
10	EMLE						
11		PJ5					
12	VSS						
13		PJ3		MTIOC3C	CTS6#/RTS6#/CTS0#/ RTS0#/SS6#/SS0#		
14	VCL						
15	VBATT						
16	MD/FINED						
17	XCIN						
18	XCOUT						
19	RES#						
20	XTAL	P37					
21	VSS						
22	EXTAL	P36					
23	VCC						
24		P35				NMI	
25	TRST#	P34		MTIOC0A/TMC13/PO12/ POE2#	SCK6/SCK0	IRQ4	
26		P33		MTIOC0D/TIOCD0/ TMR13/PO11/POE3#	RXD6/RXD0/SMISO6/ SMISO0/SSCL6/SSCL0/ CRX0	IRQ3-DS	
27		P32		MTIOC0C/TIOCC0/TMO3/ PO10/RTCOUT/RTCIC2	TXD6/TXD0/SMOSI6/ SMOSI0/SSDA6/SSDA0/ CTX0	IRQ2-DS	
28	TMS	P31		MTIOC4D/TMC12/PO9/ RTCIC1	CTS1#/RTS1#/SS1#/ SSLB0	IRQ1-DS	
29	TDI	P30		MTIOC4B/TMR13/PO8/ RTCIC0/POE8#	RXD1/SMISO1/SSCL1/ MISOB	IRQ0-DS	
30	TCK/FINEC	P27	CS7#	MTIOC2B/TMC13/PO7	SCK1/RSPCKB		
31	TDO	P26	CS6#	MTIOC2A/TMO1/PO6	TXD1/CTS3#/RTS3#/ SMOSI1/SS3#/SSDA1/ MOSIB		
32		P25	CS5#	MTIOC4C/MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/SSCL3		ADTRG0#
33		P24	CS4#	MTIOC4A/MTCLKA/ TIOCB4/TMR11/PO4	SCK3		
34		P23		MTIOC3D/MTCLKD/ TIOCD3/PO3	TXD3/CTS0#/RTS0#/ SMOSI3/SS0#/SSDA3		
35		P22		MTIOC3B/MTCLKC/ TIOCC3/TMO0/PO2	SCK0		
36		P21		MTIOC1B/TIOCA3/ TMC10/PO1	RXD0/SMISO0/SSCL0/ SCL1	IRQ9	
37		P20		MTIOC1A/TIOCB3/ TMR10/PO0	TXD0/SMOSI0/SSDA0/ SDA1	IRQ8	

Table 1.8 List of Pins and Pin Functions (144-Pin LQFP) (2/4)

Pin Number 144-Pin LQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SClC, SClD, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
38		P17		MTIOC3A/MTIOC3B/ TIOCB0/TCLKD/TMO1/ PO15/POE8#	SCK1/TXD3/SMOSI3/ SSDA3/MISOA/SDA2-DS/ IETXD	IRQ7	ADTRG#
39		P87		TIOCA2			
40		P16		MTIOC3C/MTIOC3D/ TIOCB1/TCLKC/TMO2/ PO14/RTCOU	TXD1/RXD3/SMOSI1/ SMISO3/SSDA1/SSCL3/ MOSIA/SCL2-DS/IERXD/ USB0_VBUS	IRQ6	ADTRG0#
41		P86		TIOCA0			
42		P15		MTIOC0B/MTCLKB/ TIOCB2/TCLKB/TMCI2/ PO13	RXD1/SCK3/SMOSI0/ SSCL1/CRX1-DS	IRQ5	
43		P14		MTIOC3A/MTCLKA/ TIOCB5/TCLKA/TMRI2/ PO15	CTS1#/RTS1#/SS1#/ CTX1/USB0_DPUPE	IRQ4	
44		P13		MTIOC0B/TIOCA5/TMO3/ PO13	TXD2/SMOSI2/SSDA2/ SDA0[FM+]	IRQ3	ADTRG#
45		P12		TMCI1	RXD2/SMISO2/SSCL2/ SCL0[FM+]	IRQ2	
46	VCC_USB						
47					USB0_DM		
48					USB0_DP		
49	VSS_USB						
50		P56		MTIOC3C/TIOCA1			
51	TRDATA3	P55	WAIT#	MTIOC4D/TMO3	CRX1	IRQ10	
52	TRDATA2	P54	ALE	MTIOC4B/TMCI1	CTS2#/RTS2#/SS2#/ CTX1		
53	BCLK	P53*1					
54		P52	RD#		RXD2/SMISO2/SSCL2/ SSLB3		
55		P51	WR1#/BC1#/ WAIT#		SCK2/SSLB2		
56		P50	WR0#/WR#		TXD2/SMOSI2/SSDA2/ SSLB1		
57	VSS						
58	TRCLK	P83		MTIOC4C	CTS10#/RTS10#/SS10#		
59	VCC						
60		PC7	A23/CS0#	MTIOC3A/MTCLKB/ TIOCB6/TMO2/PO31	TXD8/SMOSI8/SSDA8/ MISOA	IRQ14	
61		PC6	A22/CS1#	MTIOC3C/MTCLKA/ TIOCA6/TMCI2/PO30	RXD8/SMISO8/SSCL8/ MOSIA	IRQ13	
62		PC5	A21/CS2#/ WAIT#	MTIOC3B/MTCLKD/ TIOCD6/TCLKF/TMRI2/ PO29	SCK8/RSPCKA		
63	TRSYNC	P82		MTIOC4A/PO28	TXD10/SMOSI10/SSDA10		
64	TRDATA1	P81		MTIOC3D/PO27	RXD10/SMISO10/SSCL10		
65	TRDATA0	P80		MTIOC3B/PO26	SCK10		
66		PC4	A20/CS3#	MTIOC3D/MTCLKC/ TIOCC6/TCLKE/TMCI1/ PO25/POE0#	SCK5/CTS8#/RTS8#/ SS8#/SSLA0		
67		PC3	A19	MTIOC4D/TCLKB/PO24	TXD5/SMOSI5/SSDA5/ IETXD		
68		P77	CS7#	PO23	TXD11/SMOSI11/SSDA11		
69		P76	CS6#	PO22	RXD11/SMISO11/SSCL11		
70		PC2	A18	MTIOC4B/TCLKA/PO21	RXD5/SMISO5/SSCL5/ SSLA3/IERXD		
71		P75	CS5#	PO20	SCK11		

Table 1.8 List of Pins and Pin Functions (144-Pin LQFP) (3/4)

Pin Number 144-Pin LQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SCIC, SCID, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
72		P74	CS4#	PO19	CTS11#/RTS11#/SS11#		
73		PC1	A17	MTIOC3A/TCLKD/PO18	SCK5/SSLA2/SDA3	IRQ12	
74		PL1					
75		PC0	A16	MTIOC3C/TCLKC/PO17	CTS5#/RTS5#/SS5#/ SSLA1/SCL3	IRQ14	
76		PL0					
77		P73	CS3#	PO16			
78		PB7	A15	MTIOC3B/TIOCB5/PO31	TXD9/SMOSI9/SSDA9		
79		PB6	A14	MTIOC3D/TIOCA5/PO30	RXD9/SMISO9/SSCL9		
80		PB5	A13	MTIOC2A/MTIOC1B/ TIOCB4/TMRI1/PO29/ POE1#	SCK9		
81		PB4	A12	TIOCA4/PO28	CTS9#/RTS9#/SS9#		
82		PB3	A11	MTIOC0A/MTIOC4A/ TIOCD3/TCLKD/TMO0/ PO27/POE3#	SCK4/SCK6		
83		PB2	A10	TIOCC3/TCLKC/PO26	CTS4#/RTS4#/CTS6#/ RTS6#/SS4#/SS6#		
84		PB1	A9	MTIOC0C/MTIOC4C/ TIOCB3/TMCI0/PO25	TXD4/TXD6/SMOSI4/ SMOSI6/SSDA4/SSDA6	IRQ4-DS	
85		P72	CS2#				
86		P71	CS1#				
87		PB0	A8	MTIC5W/TIOCA3/PO24	RXD4/RXD6/SMISO4/ SMISO6/SSCL4/SSCL6/ RSPCKA	IRQ12	
88		PA7	A7	TIOCB2/PO23	MISOA		
89		PA6	A6	MTIC5V/MTCLKB/ TIOCA2/TMCI3/PO22/ POE2#	CTS5#/RTS5#/SS5#/ MOSIA		
90		PA5	A5	TIOCB1/PO21	RSPCKA		
91	VCC						
92		PA4	A4	MTIC5U/MTCLKA/ TIOCA1/TMRI0/PO20	TXD5/SMOSI5/SSDA5/ SSLA0	IRQ5-DS	
93	VSS						
94		PA3	A3	MTIOC0D/MTCLKD/ TIOCD0/TCLKB/PO19	RXD5/SMISO5/SSCL5	IRQ6-DS	
95		PA2	A2	PO18	RXD5/SMISO5/SSCL5/ SSLA3		
96		PA1	A1	MTIOC0B/MTCLKC/ TIOCB0/PO17	SCK5/SSLA2	IRQ11	
97		PA0	A0/BC0#	MTIOC4A/TIOCA0/PO16	SSLA1		
98		P67	CS7#		CRX2*2	IRQ15	
99		P66	CS6#		CTX2*2		
100		P65	CS5#				
101		PE7	D15[A15/D15]	TIOCB11	MISOB	IRQ7	AN5
102		PE6	D14[A14/D14]	TIOCA11	CTS4#/RTS4#/SS4#/ MOSIB	IRQ6	AN4
103		PK5			TXD4/SMOSI4/SSDA4		
104		P70			SCK4		
105		PK4			RXD4/SMISO4/SSCL4		
106		PE5	D13[A13/D13]	MTIOC4C/MTIOC2B/ TIOCB10	RSPCKB	IRQ5	AN3
107		PE4	D12[A12/D12]	MTIOC4D/MTIOC1A/ TIOCA10/PO28	SSLB0		AN2

**Table 1.8 List of Pins and Pin Functions (144-Pin LQFP) (4/4)**

Pin Number 144-Pin LQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SClC, SClD, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
108		PE3	D11[A11/D11]	MTIOC4B/TIOCB9/PO26/ POE8#	CTS12#/RTS12#/SS12#/ MISOB		AN1
109		PE2	D10[A10/D10]	MTIOC4A/TIOCA9/PO23	RXD12/SMISO12/ SSCL12/RXD12/SSLB3/ MOSIB	IRQ7-DS	AN0
110		PE1	D9[A9/D9]	MTIOC4C/TIOCD9/PO18	TXD12/SMOSI12/ SSDA12/TXD12/ SIOX12/SSLB2/RSPCKB		ANEX1
111		PE0	D8[A8/D8]	TIOCC9	SCK12/SSLB1		ANEX0
112		P64	CS4#				
113		P63	CS3#				
114		P62	CS2#				
115		P61	CS1#		CTS9#/RTS9#/SS9#		
116		PK3			RXD9/SMISO9/SSCL9		
117		P60	CS0#		SCK9		
118		PK2			TXD9/SMOSI9/SSDA9		
119		PD7	D7[A7/D7]	MTIC5U/POE0#	SSLC3	IRQ7	AN7
120		PD6	D6[A6/D6]	MTIC5V/POE1#	SSLC2	IRQ6	AN6
121		PD5	D5[A5/D5]	MTIC5W/POE2#	SSLC1	IRQ5	AN013
122		PD4	D4[A4/D4]	POE3#	SSLC0	IRQ4	AN012
123		PD3	D3[A3/D3]	TIOCB8/TCLKH/POE8#	RSPCKC	IRQ3	AN011
124		PD2	D2[A2/D2]	MTIOC4D/TIOCA8	MISOC/CRX0	IRQ2	AN010
125		PD1	D1[A1/D1]	MTIOC4B/TIOCB7/ TCLKG	MOSIC/CTX0	IRQ1	AN009
126		PD0	D0[A0/D0]	TIOCA7		IRQ0	AN008
127		P93	A19		CTS7#/RTS7#/SS7#		AN017
128		P92	A18		RXD7/SMISO7/SSCL7		AN016
129		P91	A17		SCK7		AN015
130	VSS						
131		P90	A16		TXD7/SMOSI7/SSDA7		AN014
132	VCC						
133		P47				IRQ15-DS	AN007
134		P46				IRQ14-DS	AN006
135		P45				IRQ13-DS	AN005
136		P44				IRQ12-DS	AN004
137		P43				IRQ11-DS	AN003
138		P42				IRQ10-DS	AN002
139		P41				IRQ9-DS	AN001
140	VREFLO						
141		P40				IRQ8-DS	AN000
142	VREFH0						
143	AVCC0						
144		P07				IRQ15	ADTRG0#

Note 1. The BCLK function is multiplexed with the I/O port function for pin P53, so the port function is not available if the external bus is enabled.

Note 2. Enabled only for the ROM capacity: 2 MB/1.5 MB

Table 1.9 List of Pins and Pin Functions (100-Pin TFLGA) (1/3)

Pin Number 100-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SClC, SClD, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
A1		P05				IRQ13	DA1
A2	VREFH						
A3		P07				IRQ15	ADTRG0#
A4	VREFLO						
A5		P43				IRQ11-DS	AN003
A6		PD0	D0[A0/D0]			IRQ0	AN008
A7		PD4	D4[A4/D4]	POE3#		IRQ4	AN012
A8		PE0	D8[A8/D8]		SCK12/SSLB1		ANEX0
A9		PE1	D9[A9/D9]	MTIOC4C/PO18	TXD12/SMOSI12/ SSDA12/TXDX12/ SIOX12/SSLB2/RSPCKB		ANEX1
A10		PE2	D10[A10/D10]	MTIOC4A/PO23	RXD12/SMISO12/ SSCL12/RXDX12/SSLB3/ MOSIB	IRQ7-DS	AN0
B1	EMLE						
B2	AVSS0						
B3	AVCC0						
B4		P40				IRQ8-DS	AN000
B5		P44				IRQ12-DS	AN004
B6		PD1	D1[A1/D1]	MTIOC4B	CTX0*1	IRQ1	AN009
B7		PD3	D3[A3/D3]	POE8#		IRQ3	AN011
B8		PD6	D6[A6/D6]	MTIC5V/POE1#		IRQ6	AN6
B9		PD7	D7[A7/D7]	MTIC5U/POE0#		IRQ7	AN7
B10		PE3	D11[A11/D11]	MTIOC4B/PO26/POE8#	CTS12#/RTS12#/SS12#/ MISOB		AN1
C1	VCL						
C2	VREFL						
C3		PJ3		MTIOC3C	CTS6#/RTS6#/CTS0#/ RTS0#/SS6#/SS0#		
C4	VREFH0						
C5		P42				IRQ10-DS	AN002
C6		P47				IRQ15-DS	AN007
C7		PD2	D2[A2/D2]	MTIOC4D	CRX0*1	IRQ2	AN010
C8		PD5	D5[A5/D5]	MTIC5W/POE2#		IRQ5	AN013
C9		PE5	D13[A13/D13]	MTIOC4C/MTIOC2B	RSPCKB	IRQ5	AN3
C10		PE4	D12[A12/D12]	MTIOC4D/MTIOC1A/ PO28	SSLB0		AN2
D1	XCIN						
D2	XCOUT						
D3	MD/FINED						
D4	VBATT						
D5		P45				IRQ13-DS	AN005
D6		P46				IRQ14-DS	AN006
D7		PE6	D14[A14/D14]		MOSIB	IRQ6	AN4
D8		PE7	D15[A15/D15]		MISOB	IRQ7	AN5
D9		PA1	A1	MTIOC0B/MTCLKC/ TIOC0B/PO17	SCK5/SSLA2	IRQ11	
D10		PA0	A0/BC0#	MTIOC4A/TIOCA0/PO16	SSLA1		
E1	XTAL	P37					
E2	VSS						
E3	RES#						



Table 1.9 List of Pins and Pin Functions (100-Pin TFLGA) (2/3)

Pin Number 100-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SCIC, SCID, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
E4	TRST#	P34		MTIOC0A/TMCI3/PO12/ POE2#	SCK6/SCK0	IRQ4	
E5		P41				IRQ9-DS	AN001
E6		PA2	A2	PO18	RXD5/SMISO5/SSCL5/ SSLA3		
E7		PA6	A6	MTIC5V/MTCLKB/ TIOCA2/TMCI3/PO22/ POE2#	CTS5#/RTS5#/SS5#/ MOSIA		
E8		PA4	A4	MTIC5U/MTCLKA/ TIOCA1/TMRI0/PO20	TXD5/SMISO5/SSDA5/ SSLA0	IRQ5-DS	
E9		PA5	A5	TIOCB1/PO21	RSPCKA		
E10		PA3	A3	MTIOC0D/MTCLKD/ TIOCDO/TCLKB/PO19	RXD5/SMISO5/SSCL5	IRQ6-DS	
F1	EXTAL	P36					
F2	VCC						
F3		P35				NMI	
F4		P32		MTIOC0C/TIOCC0/ TMO3/PO10/RTCOUT/ RTCIC2	TXD6/TXD0/SMISO6/ SMISO0/SSDA6/SSDA0/ CTX0*1	IRQ2-DS	
F5		P12		TMCI1	RXD2/SMISO2/SSCL2/ SCL0[FM+]	IRQ2	
F6		PB3	A11	MTIOC0A/MTIOC4A/ TIOC0D3/TCLKD/TMO0/ PO27/POE3#	SCK6		
F7		PB2	A10	TIOCC3/TCLKC/PO26	CTS6#/RTS6#/SS6#		
F8		PB0	A8	MTIC5W/TIOCA3/PO24	RXD6/SMISO6/SSCL6/ RSPCKA	IRQ12	
F9		PA7	A7	TIOCB2/PO23	MISOA		
F10	VSS						
G1		P33		MTIOC0D/TIOC0D/TMRI3 PO11/POE3#	RXD6/RXD0/SMISO6/ SMISO0/SSCL6/SSCL0/ CRX0*1	IRQ3-DS	
G2	TMS	P31		MTIOC4D/TMCI2/PO9/ RTCIC1	CTS1#/RTS1#/SS1#/ SSLB0	IRQ1-DS	
G3	TDI	P30		MTIOC4B/TMRI3/PO8/ RTCIC0/POE8#	RXD1/SMISO1/SSCL1/ MISOB	IRQ0-DS	
G4	TCK/FINEC	P27	CS7#	MTIOC2B/TMCI3/ PO7	SCK1/RSPCKB		
G5		P53*2	BCLK				
G6		P52	RD#		RXD2/SMISO2/SSCL2/ SSLB3		
G7		PB5	A13	MTIOC2A/MTIOC1B/ TIOCB4/TMRI1/PO29/ POE1#	SCK9		
G8		PB4	A12	TIOCA4/PO28	CTS9#/RTS9#/SS9#		
G9		PB1	A9	MTIOC0C/MTIOC4C/ TIOC0B3/TMCI0/PO25	TXD6/SMISO6/SSDA6	IRQ4-DS	
G10	VCC						
H1	TDO	P26	CS6#	MTIOC2A/TMO1/PO6	TXD1/CTS3#/RTS3#/ SMISO1/SS3#/SSDA1/ MOSIB		
H2		P25	CS5#	MTIOC4C/MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/SSCL3		ADTRG0#
H3		P16		MTIOC3C/MTIOC3D/ TIOCB1/TCLKC/TMO2/ PO14/RTCOUT	TXD1/RXD3/SMISO1/ SMISO3/SSDA1/SSCL3/ MOSIA/SCL2-DS/IERXD/ USB0_VBUS	IRQ6	ADTRG0#

**Table 1.9 List of Pins and Pin Functions (100-Pin TFLGA) (3/3)**

Pin Number 100-Pin TFLGA	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SClC, SClD, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
H4		P15		MTIOC0B/MTCLKB/ TIOCB2/TCLKB/TMCI2/ PO13	RXD1/SCK3/SMISO1/ SSCL1/CRX1-DS	IRQ5	
H5		P55	WAIT#	MTIOC4D/TMO3	CRX1	IRQ10	
H6		P54	ALE	MTIOC4B/TMCI1	CTS2#/RTS2#/SS2#/ CTX1		
H7		PC7	A23/CS0#	MTIOC3A/MTCLKB/ TMO2/PO31	TXD8/SMOSI8/SSDA8/ MISOA	IRQ14	
H8		PC6	A22/CS1#	MTIOC3C/MTCLKA/ TMCI2/PO30	RXD8/SMISO8/SSCL8/ MOSIA	IRQ13	
H9		PB6	A14	MTIOC3D/TIOCA5/PO30	RXD9/SMISO9/SSCL9		
H10		PB7	A15	MTIOC3B/TIOCB5/PO31	TXD9/SMOSI9/SSDA9		
J1		P24	CS4#	MTIOC4A/MTCLKA/ TIOCB4/TMRI1/PO4	SCK3		
J2		P21		MTIOC1B/TIOCA3/ TMCI0/PO1	RXD0/SMISO0/SSCL0	IRQ9	
J3		P17		MTIOC3A/MTIOC3B/ TIOCB0/TCLKD/TMO1/ PO15/POE8#	SCK1/TXD3/SMOSI3/ SSDA3/MISOA/SDA2-DS/ IETXD	IRQ7	ADTRG#
J4		P13		MTIOC0B/TIOCA5/TMO3/ PO13	TXD2/SMOSI2/SSDA2/ SDA0[FM+]	IRQ3	ADTRG#
J5	VSS_USB						
J6	VCC_USB						
J7		P50	WR0#/WR#		TXD2/SMOSI2/SSDA2/ SSLB1		
J8		PC4	A20/CS3#	MTIOC3D/MTCLKC/ TMCI1/PO25/POE0#	SCK5/CTS8#/RTS8#/ SS8#/SSLA0		
J9		PC0	A16	MTIOC3C/TCLKC/PO17	CTS5#/RTS5#/SS5#/ SSLA1	IRQ14	
J10		PC1	A17	MTIOC3A/TCLKD/PO18	SCK5/SSLA2	IRQ12	
K1		P23		MTIOC3D/MTCLKD/ TIOCD3/PO3	TXD3/CTS0#/RTS0#/ SMOSI3/SS0#/SSDA3		
K2		P22		MTIOC3B/MTCLKC/ TIOCC3/TMO0/PO2	SCK0		
K3		P20		MTIOC1A/TIOCB3/ TMRI0/PO0	TXD0/SMOSI0/SSDA0	IRQ8	
K4		P14		MTIOC3A/MTCLKA/ TIOCB5/TCLKA/TMRI2/ PO15	CTS1#/RTS1#/SS1#/ CTX1/USB0_DPUPE	IRQ4	
K5					USB0_DM		
K6					USB0_DP		
K7		P51	WR1#/BC1#/ WAIT#		SCK2/SSLB2		
K8		PC5	A21/CS2#/ WAIT#	MTIOC3B/MTCLKD/ TMRI2/PO29	SCK8/RSPCKA		
K9		PC3	A19	MTIOC4D/TCLKB/PO24	TXD5/SMOSI5/SSDA5/ IETXD		
K10		PC2	A18	MTIOC4B/TCLKA/PO21	RXD5/SMISO5/SSCL5/ SSLA3/IERXD		

Note 1. Enabled only for the ROM capacity of 768 Kbytes or more

Note 2. The BCLK function is multiplexed with the I/O port function for pin P53, so the port function is not available if the external bus is enabled.

Table 1.10 List of Pins and Pin Functions (100-Pin LQFP) (1/3)

Pin Number 100-Pin LQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
1	VREFH						
2	EMLE						
3	VREFL						
4		PJ3		MTIOC3C	CTS6#/RTS6#/CTS0#/ RTS0#/SS6#/SS0#		
5	VCL						
6	VBATT						
7	MD/FINED						
8	XCIN						
9	XCOUT						
10	RES#						
11	XTAL	P37					
12	VSS						
13	EXTAL	P36					
14	VCC						
15		P35				NMI	
16	TRST#	P34		MTIOC0A/TMCI3/PO12/ POE2#	SCK6/SCK0	IRQ4	
17		P33		MTIOC0D/TIOCD0/ TMRI3/PO11/POE3#	RXD6/RXD0/SMISO6/ SMISO0/SSCL6/SSCL0/ CRX0*1	IRQ3-DS	
18		P32		MTIOC0C/TIOCC0/TMO3/ PO10/RTCOU7/RTCIC2	TXD6/TXD0/SMOSI6/ SMOSI0/SSDA6/SSDA0/ CTX0*1	IRQ2-DS	
19	TMS	P31		MTIOC4D/TMCI2/PO9/ RTCIC1	CTS1#/RTS1#/SS1#/ SSLB0	IRQ1-DS	
20	TDI	P30		MTIOC4B/TMRI3/PO8/ RTCIC0/POE8#	RXD1/SMISO1/SSCL1/ MISOB	IRQ0-DS	
21	TCK/FINEC	P27	CS7#	MTIOC2B/TMCI3/PO7	SCK1/RSPCKB		
22	TDO	P26	CS6#	MTIOC2A/TMO1/PO6	TXD1/CTS3#/RTS3#/ SMOSI1/SS3#/SSDA1/ MOSIB		
23		P25	CS5#	MTIOC4C/MTCLKB/ TIOCA4/PO5	RXD3/SMISO3/SSCL3		ADTRG0#
24		P24	CS4#	MTIOC4A/MTCLKA/ TIOCB4/TMRI1/PO4	SCK3		
25		P23		MTIOC3D/MTCLKD/ TIOCD3/PO3	TXD3/CTS0#/RTS0#/ SMOSI3/SS0#/SSDA3		
26		P22		MTIOC3B/MTCLKC/ TIOCC3/TMO0/PO2	SCK0		
27		P21		MTIOC1B/TIOCA3/ TMCI0/PO1	RXD0/SMISO0/SSCL0	IRQ9	
28		P20		MTIOC1A/TIOCB3/ TMRI0/PO0	TXD0/SMOSI0/SSDA0	IRQ8	
29		P17		MTIOC3A/MTIOC3B/ TIOCB0/TCLKD/TMO1/ PO15/POE8#	SCK1/TXD3/SMOSI3/ SSDA3/MISOA/SDA2-DS/ IETXD	IRQ7	ADTRG#
30		P16		MTIOC3C/MTIOC3D/ TIOCB1/TCLKC/TMO2/ PO14/RTCOU7	TXD1/RXD3/SMOSI1/ SMISO3/SSDA1/SSCL3/ MOSIA/SCL2-DS/IERXD/ USB_VBUS	IRQ6	ADTRG0#
31		P15		MTIOC0B/MTCLKB/ TIOCB2/TCLKB/TMCI2/ PO13	RXD1/SCK3/SMISO1/ SSCL1/CRX1-DS	IRQ5	
32		P14		MTIOC3A/MTCLKA/ TIOCB5/TCLKA/TMRI2/ PO15	CTS1#/RTS1#/SS1#/ CTX1/USB0_DPUPE	IRQ4	

Table 1.10 List of Pins and Pin Functions (100-Pin LQFP) (2/3)

Pin Number 100-Pin LQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SClC, SClD, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
33		P13		MTIOC0B/TIOCA5/TMO3/ PO13	TXD2/SMOSI2/SSDA2/ SDA0[FM+]	IRQ3	ADTRG#
34		P12		TMC11	RXD2/SMISO2/SSCL2/ SCL0[FM+]	IRQ2	
35	VCC_USB						
36					USB0_DM		
37					USB0_DP		
38	VSS_USB						
39		P55	WAIT#	MTIOC4D/TMO3	CRX1	IRQ10	
40		P54	ALE	MTIOC4B/TMC11	CTS2#/RTS2#/SS2#/ CTX1		
41	BCLK	P53*2					
42		P52	RD#		RXD2/SMISO2/SSCL2/ SSLB3		
43		P51	WR1#/BC1#/ WAIT#		SCK2/SSLB2		
44		P50	WR0#/WR#		TXD2/SMOSI2/SSDA2/ SSLB1		
45		PC7	A23/CS0#	MTIOC3A/MTCLKB/ TMO2/PO31	TXD8/SMOSI8/SSDA8/ MISOA	IRQ14	
46		PC6	A22/CS1#	MTIOC3C/MTCLKA/ TMC12/PO30	RXD8/SMISO8/SSCL8/ MOSIA	IRQ13	
47		PC5	A21/CS2#/ WAIT#	MTIOC3B/MTCLKD/ TMR12/PO29	SCK8/RSPCKA		
48		PC4	A20/CS3#	MTIOC3D/MTCLKC/ TMC11/PO25/POE0#	SCK5/CTS8#/RTS8#/ SS8#/SSLA0		
49		PC3	A19	MTIOC4D/TCLKB/PO24	TXD5/SMOSI5/SSDA5/ IETXD		
50		PC2	A18	MTIOC4B/TCLKA/PO21	RXD5/SMISO5/SSCL5/ SSLA3/IERXD		
51		PC1	A17	MTIOC3A/TCLKD/PO18	SCK5/SSLA2	IRQ12	
52		PC0	A16	MTIOC3C/TCLKC/PO17	CTS5#/RTS5#/SS5#/ SSLA1	IRQ14	
53		PB7	A15	MTIOC3B/TIOCB5/PO31	TXD9/SMOSI9/SSDA9		
54		PB6	A14	MTIOC3D/TIOCA5/PO30	RXD9/SMISO9/SSCL9		
55		PB5	A13	MTIOC2A/MTIOC1B/ TIOCB4/TMR11/PO29/ POE1#	SCK9		
56		PB4	A12	TIOCA4/PO28	CTS9#/RTS9#/SS9#		
57		PB3	A11	MTIOC0A/MTIOC4A/ TIOC3/TCLKD/TMO0/ PO27/POE3#	SCK6		
58		PB2	A10	TIOCC3/TCLKC/PO26	CTS6#/RTS6#/SS6#		
59		PB1	A9	MTIOC0C/MTIOC4C/ TIOCB3/TMC10/PO25	TXD6/SMOSI6/SSDA6	IRQ4-DS	
60	VCC						
61		PB0	A8	MTIC5W/TIOCA3/PO24	RXD6/SMISO6/SSCL6/ RSPCKA	IRQ12	
62	VSS						
63		PA7	A7	TIOCB2/PO23	MISOA		
64		PA6	A6	MTIC5V/MTCLKB/ TIOCA2/TMC13/PO22/ POE2#	CTS5#/RTS5#/SS5#/ MOSIA		
65		PA5	A5	TIOCB1/PO21	RSPCKA		
66		PA4	A4	MTIC5U/MTCLKA/ TIOCA1/TMR10/PO20	TXD5/SMOSI5/SSDA5/ SSLA0	IRQ5-DS	

**Table 1.10 List of Pins and Pin Functions (100-Pin LQFP) (3/3)**

Pin Number 100-Pin LQFP	Power Supply Clock System Control	I/O Port	Bus	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SClC, SClD, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
67		PA3	A3	MTIOC0D/MTCLKD/ TIOC0D/TCLKB/PO19	RXD5/SMISO5/SSCL5	IRQ6-DS	
68		PA2	A2	PO18	RXD5/SMISO5/SSCL5/ SSLA3		
69		PA1	A1	MTIOC0B/MTCLKC/ TIOC0B/PO17	SCK5/SSLA2	IRQ11	
70		PA0	A0/BC0#	MTIOC4A/TIOCA0/PO16	SSLA1		
71		PE7	D15[A15/D15]		MISOB	IRQ7	AN5
72		PE6	D14[A14/D14]		MOSIB	IRQ6	AN4
73		PE5	D13[A13/D13]	MTIOC4C/MTIOC2B	RSPCKB	IRQ5	AN3
74		PE4	D12[A12/D12]	MTIOC4D/MTIOC1A/ PO28	SSLB0		AN2
75		PE3	D11[A11/D11]	MTIOC4B/PO26/POE8#	CTS12#/RTS12#/SS12#/ MISOB		AN1
76		PE2	D10[A10/D10]	MTIOC4A/PO23	RXD12/SMISO12/ SSCL12/RXDX12/SSLB3/ MOSIB	IRQ7-DS	AN0
77		PE1	D9[A9/D9]	MTIOC4C/PO18	TXD12/SMOS12/ SSDA12/TXDX12/ SIOX12/SSLB2/RSPCKB		ANEX1
78		PE0	D8[A8/D8]		SCK12/SSLB1		ANEX0
79		PD7	D7[A7/D7]	MTIC5U/POE0#		IRQ7	AN7
80		PD6	D6[A6/D6]	MTIC5V/POE1#		IRQ6	AN6
81		PD5	D5[A5/D5]	MTIC5W/POE2#		IRQ5	AN013
82		PD4	D4[A4/D4]	POE3#		IRQ4	AN012
83		PD3	D3[A3/D3]	POE8#		IRQ3	AN011
84		PD2	D2[A2/D2]	MTIOC4D	CRX0*1	IRQ2	AN010
85		PD1	D1[A1/D1]	MTIOC4B	CTX0*1	IRQ1	AN009
86		PD0	D0[A0/D0]			IRQ0	AN008
87		P47				IRQ15-DS	AN007
88		P46				IRQ14-DS	AN006
89		P45				IRQ13-DS	AN005
90		P44				IRQ12-DS	AN004
91		P43				IRQ11-DS	AN003
92		P42				IRQ10-DS	AN002
93		P41				IRQ9-DS	AN001
94	VREFL0						
95		P40				IRQ8-DS	AN000
96	VREFH0						
97	AVCC0						
98		P07				IRQ15	ADTRG0#
99	AVSS0						
100		P05				IRQ13	DA1

Note 1. Enabled only for the ROM capacity of 768 Kbytes or more

Note 2. The BCLK function is multiplexed with the I/O port function for pin P53, so the port function is not available if the external bus is enabled.

Table 1.11 List of Pins and Pin Functions (80-Pin LQFP) (1/3)

Pin Number 100-Pin LQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SC1c, SC1d, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
1	VREFH					
2	EMLE					
3	VREFL					
4	VCL					
5	VBATT					
6	MD/FINED					
7	XCIN					
8	XCOUT					
9	RES#					
10	XTAL	P37				
11	VSS					
12	EXTAL	P36				
13	VCC					
14		P35			NMI	
15	TRST#	P34	MTIOC0A/TMC13/PO12/ POE2#	SCK6	IRQ4	
16		P32	MTIOC0C/TIOCC0/TMO3/ PO10/RTCOU/RTCIC2	TXD6/SMOS16/SSDA6	IRQ2-DS	
17	TMS	P31	MTIOC4D/TMC12/PO9/ RTCIC1	CTS1#/RTS1#/SS1#/ SSLB0	IRQ1-DS	
18	TDI	P30	MTIOC4B/TMR13/PO8/ RTCIC0/POE8#	RXD1/SMISO1/SSCL1/ MISOB	IRQ0-DS	
19	TCK/FINEC	P27	MTIOC2B/TMC13/PO7	SCK1/RSPCKB		
20	TDO	P26	MTIOC2A/TMO1/PO6	TXD1/SMOS1/SSDA1/ MOSIB		
21		P21	MTIOC1B/TIOCA3/TMC10/ PO1		IRQ9	
22		P20	MTIOC1A/TIOCB3/TMR10/ PO0		IRQ8	
23		P17	MTIOC3A/MTIOC3B/ TIOCB0/TCLKD/TMO1/ PO15/POE8#	SCK1/MISOA/SDA2-DS/ IETXD	IRQ7	ADTRG#
24		P16	MTIOC3C/MTIOC3D/ TIOCB1/TCLKC/TMO2/ PO14/RTCOU	TXD1/SMOS1/SSDA1/ MOSIA/SCL2-DS/IERXD/ USB0_VBUS	IRQ6	ADTRG0#
25		P15	MTIOC0B/MTCLKB/ TIOCB2/TCLKB/TMC12/ PO13	RXD1/SMISO1/SSCL1/ CRX1-DS	IRQ5	
26		P14	MTIOC3A/MTCLKA/ TIOCB5/TCLKA/TMR12/ PO15	CTS1#/RTS1#/SS1#/ CTX1/USB0_DPUPE	IRQ4	
27		P13	MTIOC0B/TIOCA5/TMO3/ PO13	SDA0[FM+]	IRQ3	ADTRG#
28		P12	TMC11	SCL0[FM+]	IRQ2	
29	VCC_USB					
30				USB0_DM		
31				USB0_DP		
32	VSS_USB					
33		P55	MTIOC4D/TMO3	CRX1	IRQ10	
34		P54	MTIOC4B/TMC11	CTX1		
35		PC7	MTIOC3A/MTCLKB/TMO2/ PO31	TXD8/SMOS18/SSDA8/ MISOA	IRQ14	
36		PC6	MTIOC3C/MTCLKA/ TMC12/PO30	RXD8/SMISO8/SSCL8/ MOSIA	IRQ13	

Table 1.11 List of Pins and Pin Functions (80-Pin LQFP) (2/3)

Pin Number	Power Supply Clock System Control	I/O Port	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SC1c, SC1d, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
37		PC5	MTIOC3B/MTCLKD/ TMRI2/PO29	SCK8/RSPCKA		
38		PC4	MTIOC3D/MTCLKC/ TMC11/PO25/POE0#	SCK5/CTS8#/RTS8#/ SS8#/SSLA0		
39		PC3	MTIOC4D/TCLKB/PO24	TXD5/SMOSI5/SSDA5/ IETXD		
40		PC2	MTIOC4B/TCLKA/PO21	RXD5/SMISO5/SSCL5/ SSLA3/IERXD		
41		PB7	MTIOC3B/TIOC5B/PO31	TXD9/SMOSI9/SSDA9		
42		PB6	MTIOC3D/TIOCA5/PO30	RXD9/SMISO9/SSCL9		
43		PB5	MTIOC2A/MTIOC1B/ TIOCB4/TMRI1/PO29/ POE1#	SCK9		
44		PB4	TIOCA4/PO28	CTS9#/RTS9#/SS9#		
45		PB3	MTIOC0A/MTIOC4A/ TIOCD3/TCLKD/TMO0/ PO27/POE3#	SCK6		
46		PB2	TIOCC3/TCLKC/PO26	CTS6#/RTS6#/SS6#		
47		PB1	MTIOC0C/MTIOC4C/ TIOCB3/TMC10/PO25	TXD6/SMOSI6/SSDA6	IRQ4-DS	
48	VCC					
49		PB0	MTIC5W/TIOCA3/PO24	RXD6/SMISO6/SSCL6/ RSPCKA	IRQ12	
50	VSS					
51		PA6	MTIC5V/MTCLKB/TIOCA2/ TMC13/PO22/POE2#	CTS5#/RTS5#/SS5#/ MOSIA		
52		PA5	TIOCB1/PO21	RSPCKA		
53		PA4	MTIC5U/MTCLKA/TIOCA1/ TMRI0/PO20	TXD5/SMOSI5/SSDA5/ SSLA0	IRQ5-DS	
54		PA3	MTIOC0D/MTCLKD/ TIOCD0/TCLKB/PO19	RXD5/SMISO5/SSCL5	IRQ6-DS	
55		PA2	PO18	RXD5/SMISO5/SSCL5/ SSLA3		
56		PA1	MTIOC0B/MTCLKC/ TIOCB0/PO17	SCK5/SSLA2	IRQ11	
57		PA0	MTIOC4A/TIOCA0/PO16	SSLA1		
58		PE5	MTIOC4C/MTIOC2B	RSPCKB	IRQ5	AN3
59		PE4	MTIOC4D/MTIOC1A/PO28	SSLB0		AN2
60		PE3	MTIOC4B/PO26/POE8#	CTS12#/RTS12#/SS12#/ MISOB		AN1
61		PE2	MTIOC4A/PO23	RXD12/SMISO12/SSCL12/ RXDX12/SSLB3/MOSIB	IRQ7-DS	AN0
62		PE1	MTIOC4C/PO18	TXD12/SMOSI12/SSDA12/ TXDX12/SIOX12/SSLB2/ RSPCKB		ANEX1
63		PE0		SCK12/SSLB1		ANEX0
64		PD2	MTIOC4D		IRQ2	AN010
65		PD1	MTIOC4B		IRQ1	AN009
66		PD0			IRQ0	AN008
67		P47			IRQ15-DS	AN007
68		P46			IRQ14-DS	AN006
69		P45			IRQ13-DS	AN005
70		P44			IRQ12-DS	AN004
71		P43			IRQ11-DS	AN003
72		P42			IRQ10-DS	AN002

**Table 1.11 List of Pins and Pin Functions (80-Pin LQFP) (3/3)**

Pin Number 100-Pin LQFP	Power Supply Clock System Control	I/O Port	Timer (MTU, TPU, TMR, PPG, RTC, POE)	Communications (SCIc, SCId, RSPI, RIIC, CAN, IEB, USB)	Interrupt	S12AD, AD, DA
73		P41			IRQ9-DS	AN001
74	VREFL0					
75		P40			IRQ8-DS	AN000
76	VREFH0					
77	AVCC0					
78		P07			IRQ15	ADTRG0#
79	AVSS0					
80		P05			IRQ13	DA1



## 2. CPU

The RX630 Group is an MCU with the high-speed, high-performance 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 and 8 floating-point operation instructions, and nine DSP instructions, for a total of 90 instructions. It has 10 addressing modes and caters to register–register operations, register–memory operations, immediate–register operations, immediate–memory operations, memory–memory transfer, and bitwise operations. High-speed operation was realized by achieving execution in a single cycle not only for register–register operations, but also for other types of multiple 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, instruction 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 “out-of-order completion” of this kind, the execution of instructions is controlled to optimize numbers of clock cycles.

### 2.1 Features

- High instruction execution rate: One instruction in one 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
- 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 one to eight bytes)
  - Short formats for frequently used instructions
- Floating-point operation instructions: 8
- 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
- Memory protection unit
- Data arrangement
  - Selectable as little endian or big endian

## 2.2 Register Set of the CPU

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

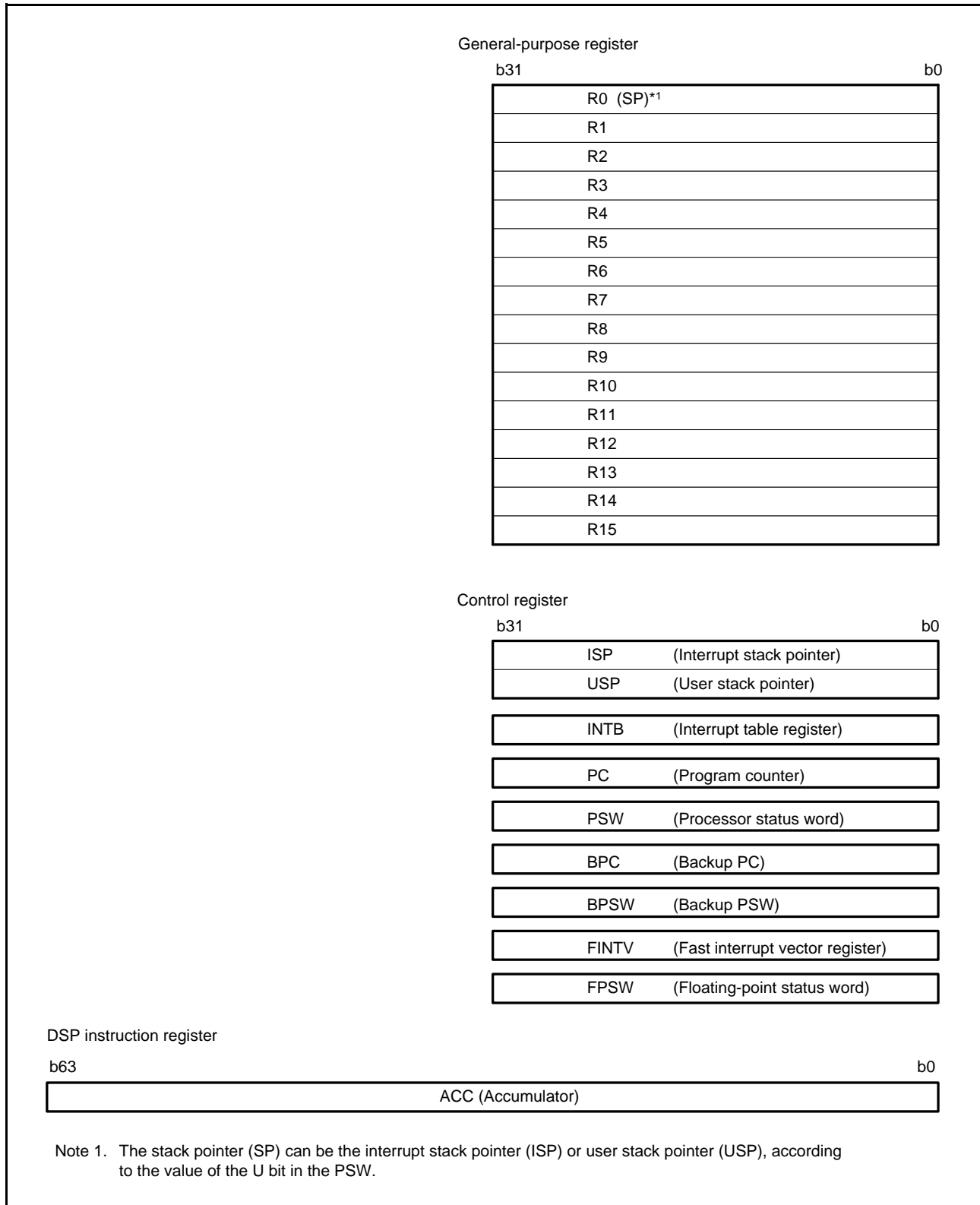


Figure 2.1 Register Set of the CPU

## 2.2.1 General-Purpose Registers (R0 to R15)

This CPU has sixteen general-purpose registers (R0 to R15). R1 to R15 can be used as data registers or address registers. R0, a general-purpose register, also functions as the stack pointer (SP).

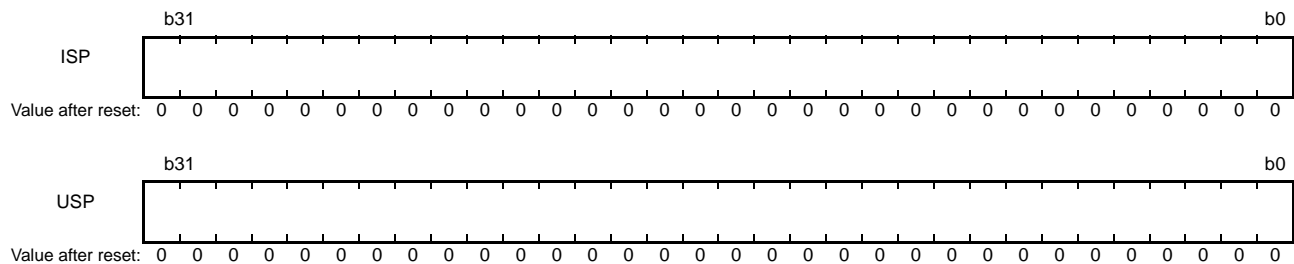
The stack pointer is switched to operate as the interrupt stack pointer (ISP) or user stack pointer (USP) by the value of the stack pointer select bit (U) in the processor status word (PSW).

## 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)

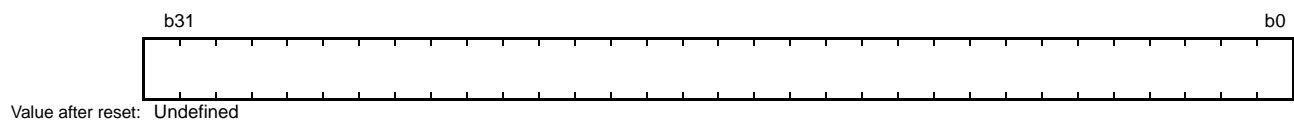
### 2.2.2.1 Interrupt Stack Pointer (ISP)/User Stack Pointer (USP)



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).

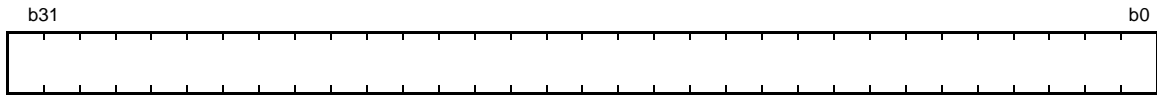
Set the ISP or USP to a multiple of four, as this reduces the numbers of cycles required to execute interrupt sequences and instructions entailing stack manipulation.

### 2.2.2.2 Interrupt Table Register (INTB)



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

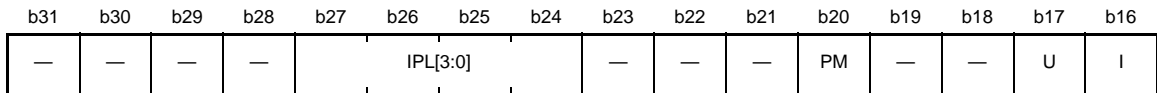
### 2.2.2.3 Program Counter (PC)



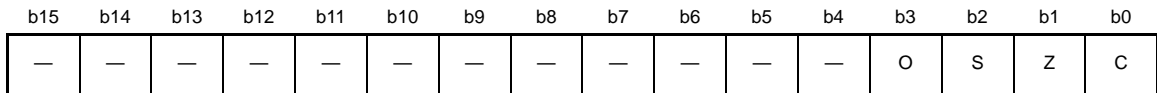
Value after reset: Contents of addresses FFFFFFFCh to FFFFFFFFh

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

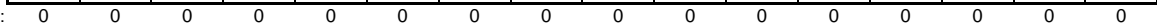
### 2.2.2.4 Processor Status Word (PSW)



Value after reset:



Value after reset:



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 non-zero. 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

Bit	Symbol	Bit Name	Description	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, the value of 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 is set to 0. When the processor mode is switched from supervisor mode to user mode, this bit is set to 1.

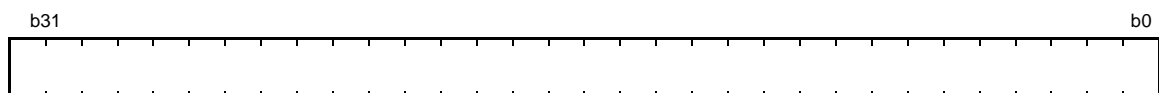
### PM Bit (Processor Mode Select)

This bit specifies the processor mode. When an exception is accepted, the value of 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 sixteen levels from zero to fifteen, wherein priority level zero is the lowest and priority level fifteen 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 fifteen (Fh) disables all interrupt requests. The IPL[3:0] bits are set to level fifteen (Fh) when a non-maskable interrupt is generated. When interrupts in general are generated, the bits are set to the priority levels of accepted interrupts.

#### 2.2.2.5 Backup PC (BPC)

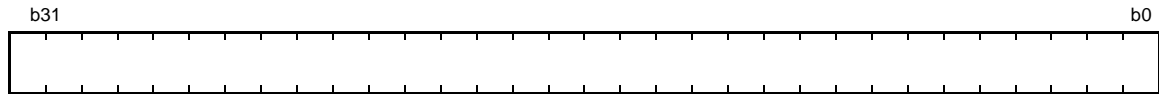


Value after reset: Undefined

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)

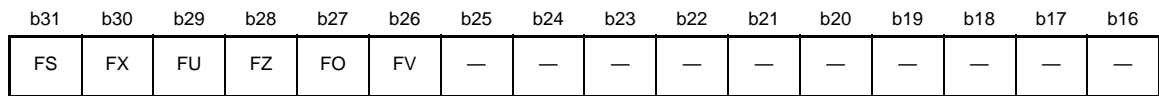


Value after reset: Undefined

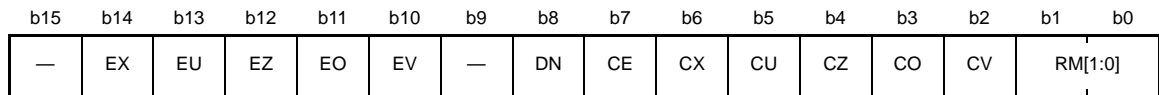
The fast interrupt vector register (FINTV) is provided to speed up response to interrupts.

The FINTV register specifies a branch destination address when a fast interrupt has been generated.

### 2.2.2.8 Floating-Point Status Word (FPSW)



Value after reset:



Value after reset:

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
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

Bit	Symbol	Bit Name	Description	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  $+\infty$ : An inexact result is rounded to the nearest available value in the direction of positive infinity.
- Rounding towards  $-\infty$ : 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  $+\infty$ , and rounding towards  $-\infty$  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 Flag (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)

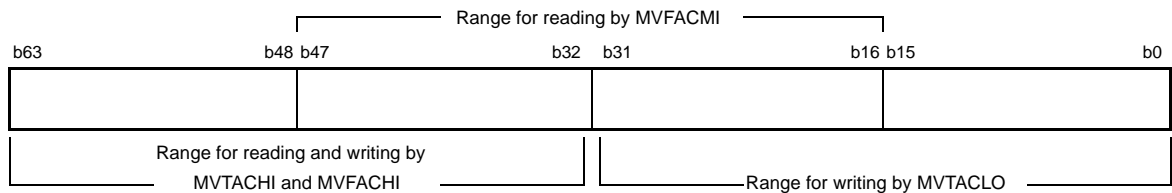
**FS Flag (Floating-Point Error Summary Flag)**

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 has been generated, 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 treatment of data is selectable as little or big endian.

### 2.5.1 Switching the Endian

As arrangements of bytes, the RX630 Group 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, see 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.

	D31 to D24	D23 to D16	D15 to D8	D7 to D0
General purpose register: Rm	HH	HL	LH	LL

**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 LH	—	—	—	—	—	—
Address 1	Transfer from LL	Transfer from LH	—	—	—	—	—
Address 2	—	Transfer from LL	Transfer from LH	—	—	—	—
Address 3	—	—	Transfer from LL	Transfer from LH	—	—	—
Address 4	—	—	—	Transfer from LL	Transfer from LH	—	—
Address 5	—	—	—	—	Transfer from LL	Transfer from LH	—
Address 6	—	—	—	—	—	Transfer from LL	Transfer from LH
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

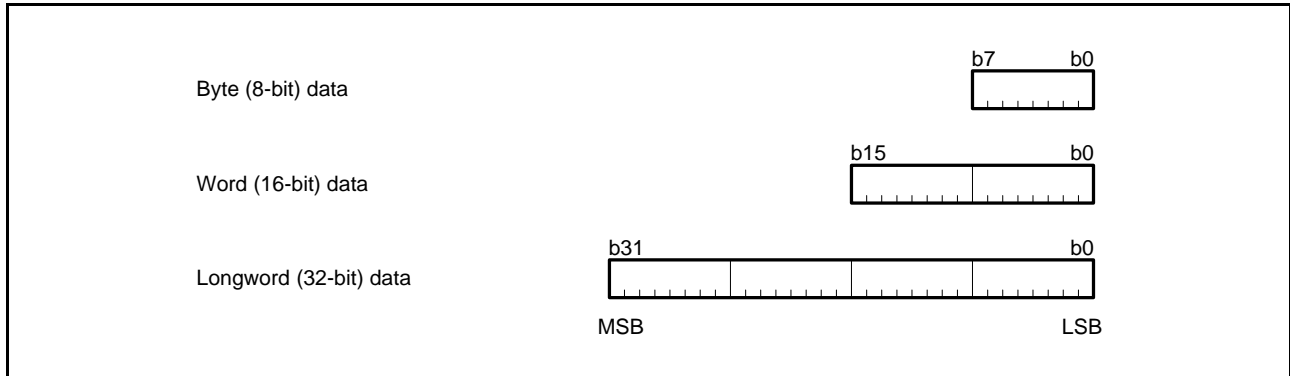
Ensure that access to I/O registers is in accord with the following rules.

- With I/O registers for which a bus width of eight bits is indicated, use instructions having operands of the same width (eight 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

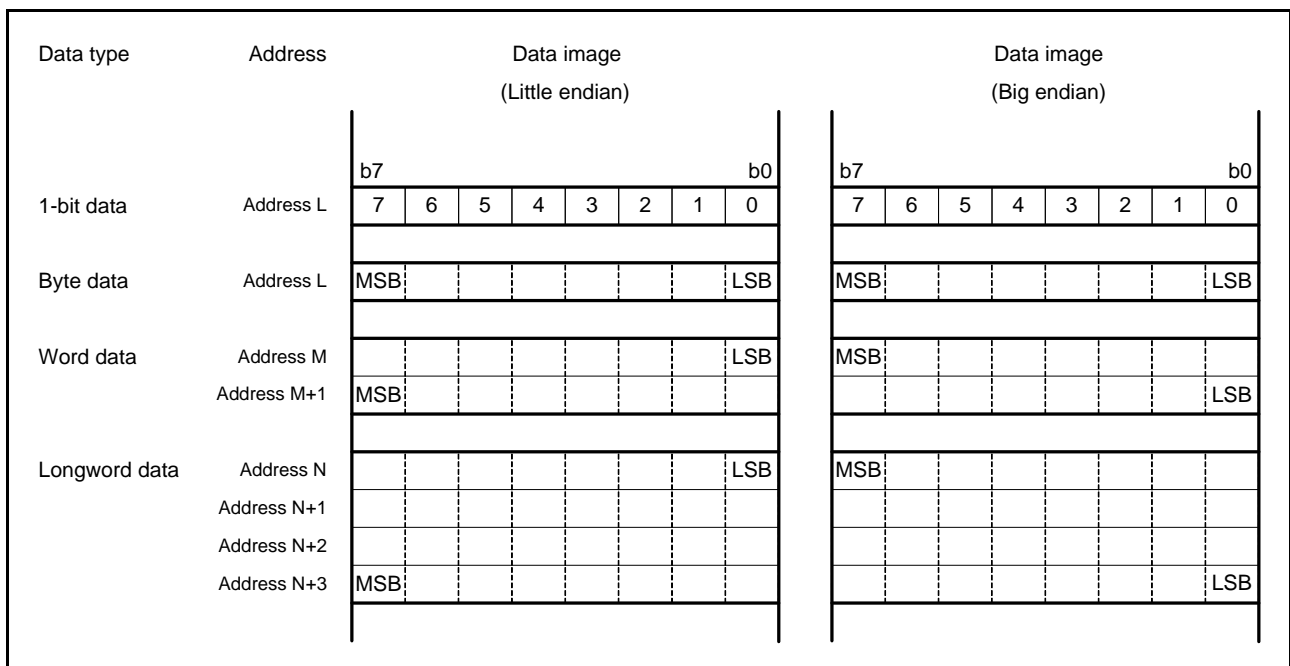
Figure 2.2 shows the relation between the sizes of registers and bit numbers.



**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 four 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, access exception, undefined instruction exception, floating-point exception, non-maskable interrupt, and reset are allocated to addresses in the range from FFFFFFFF80h to FFFFFFFFh. Figure 2.4 shows the fixed vector table.

	MSB	LSB
FFFFFFF80h	(Reserved)	
⋮	⋮	
FFFFFFFCCh	(Reserved)	
FFFFFFFD0h	Privileged instruction exception	
FFFFFFFD4h	Access exception	
FFFFFFFD8h	(Reserved)	
FFFFFFFDCh	Undefined instruction exception	
FFFFFFFE0h	(Reserved)	
FFFFFFFE4h	Floating-point exception	
FFFFFFFE8h	(Reserved)	
FFFFFFFECh	(Reserved)	
FFFFFFF0h	(Reserved)	
FFFFFFF4h	(Reserved)	
FFFFFFF8h	Non-maskable interrupt	
FFFFFFFCh	Reset	

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, see section 15.3.1, Interrupt Vector Table.



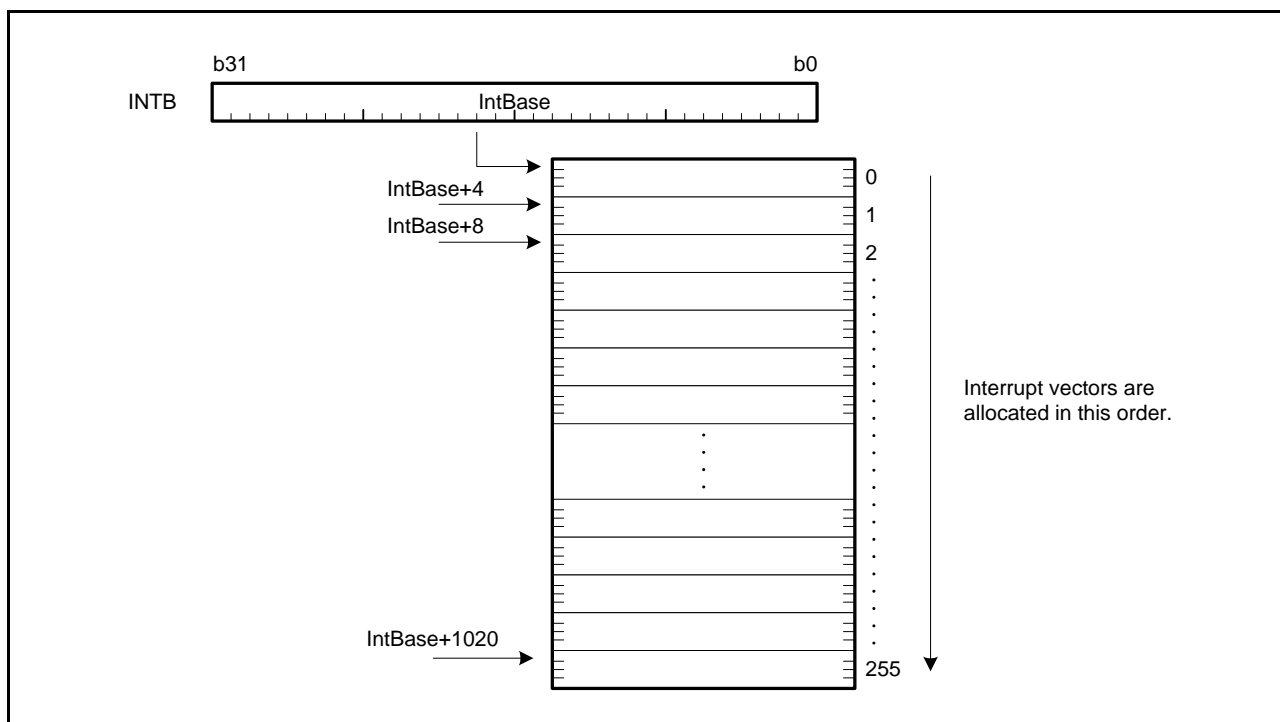


Figure 2.5 Relocatable Vector Table

## 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 three 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 5-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 8-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 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.

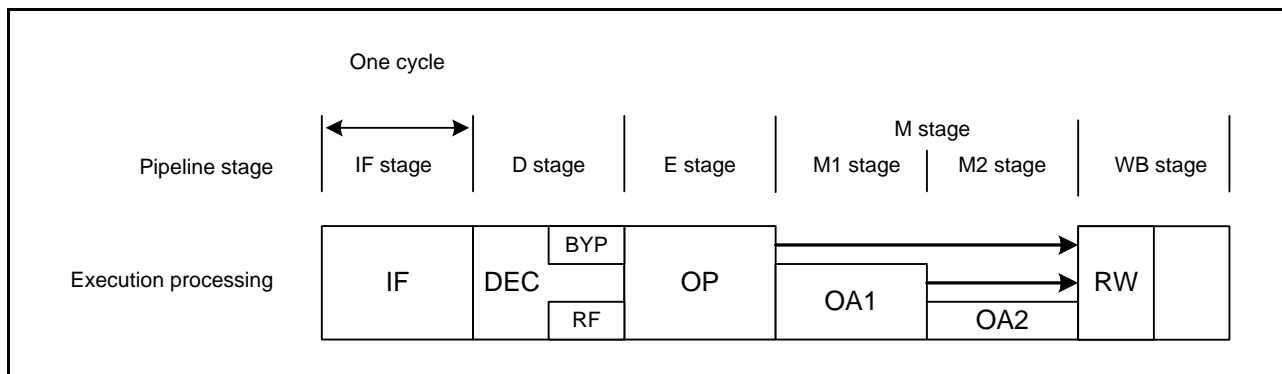


Figure 2.6 Pipeline Configuration and its Operation

## 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	<ul style="list-style-type: none"> <li>• {ABS, NEG, NOT} "Rd"/"Rs, Rd"</li> <li>• {ADC, MAX, MIN, ROTL, ROTR, XOR} "#IMM, Rd"/"Rs, Rd"</li> <li>• ADD "#IMM, Rd"/"Rs, Rd"/"#IMM, Rs, Rd"/"Rs, Rs2, Rd"</li> <li>• {AND, MUL, OR, SUB} "#IMM, Rd"/"Rs, Rd"/"Rs, Rs2, Rd"</li> <li>• {CMP, TST} "#IMM, Rs"/"Rs, Rs2"</li> <li>• NOP</li> <li>• {ROL, ROR, RORC, SAT} "Rd"</li> <li>• SBB "Rs, Rd"</li> <li>• {SHAR, SHLL, SHLR} "#IMM, Rd"/"Rs, Rd"/"#IMM, Rs, Rd"</li> </ul>	Figure 2.7	1
Arithmetic/logic instructions (division)	<ul style="list-style-type: none"> <li>• DIV "#IMM, Rd"/"Rs, Rd"</li> <li>• DIVU "#IMM, Rd"/"Rs, Rd"</li> </ul>	Figure 2.7	3 to 20*1
Data transfer instructions (register-register, immediate-register)	<ul style="list-style-type: none"> <li>• MOV "#IMM, Rd"/"Rs, Rd"</li> <li>• {MOVU, REVL, REVW} "Rs, Rd"</li> <li>• SCCnd "Rd"</li> <li>• {STNZ, STZ} "#IMM, Rd"</li> </ul>	Figure 2.7	1
Transfer instructions (load operation)	<ul style="list-style-type: none"> <li>• {MOV, MOVU} "[Rs], Rd"/"dsp[Rs], Rd"/"[Rs+], Rd"/"[-Rs], Rd"/"[Ri, Rb], Rd"</li> <li>• POP "Rd"</li> </ul>	Figure 2.8	Throughput: 1 Latency: 2*2
Transfer instructions (store operation)	<ul style="list-style-type: none"> <li>• MOV "Rs, [Rd]"/"Rs, dsp[Rd]"/"Rs, [Rd+]/"Rs, [-Rd]"/"Rs, [Ri, Rb]"/"#IMM, dsp[Rd]"/"#IMM, [Rd]"</li> <li>• PUSH "Rs"</li> <li>• PUSHC "CR"</li> <li>• SCCnd "[Rd]"/"dsp[Rd]"</li> </ul>	Figure 2.9	1
Bit manipulation instructions (register)	<ul style="list-style-type: none"> <li>• {BCLR, BNOT, BSET} "#IMM, Rd"/"Rs, Rd"</li> <li>• BMCnd "#IMM, Rd"</li> <li>• BTST "#IMM, Rs"/"Rs, Rs2"</li> </ul>	Figure 2.7	1
Branch instructions	<ul style="list-style-type: none"> <li>• BCnd "pcdsp"</li> <li>• {BRA, BSR} "pcdsp"/"Rs"</li> <li>• {JMP, JSR} "Rs"</li> </ul>	Figure 2.18	Branch taken: 3 Branch not taken: 1
Floating-point operation instructions (register-register, immediate-register)	<ul style="list-style-type: none"> <li>• FCMP "#IMM, Rd"/"Rs, Rs2"</li> </ul>	Figure 2.7	1
System manipulation instructions	<ul style="list-style-type: none"> <li>• {CLRPSW, SETPSW} "flag"</li> <li>• MVTC "#IMM, CR"/"Rs, CR"</li> <li>• MVFC "CR, Rd"</li> <li>• MVTIPL "#IMM"</li> </ul>	—	1
DSP instructions	<ul style="list-style-type: none"> <li>• {MACHI, MACLO, MULHI, MULLO} "Rs, Rs2"</li> <li>• {MVFACHI, MVFACMI} "Rd"</li> <li>• {MVTACHI, MVTACLO} "Rs"</li> <li>• RACW "#IMM"</li> </ul>	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, see 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.

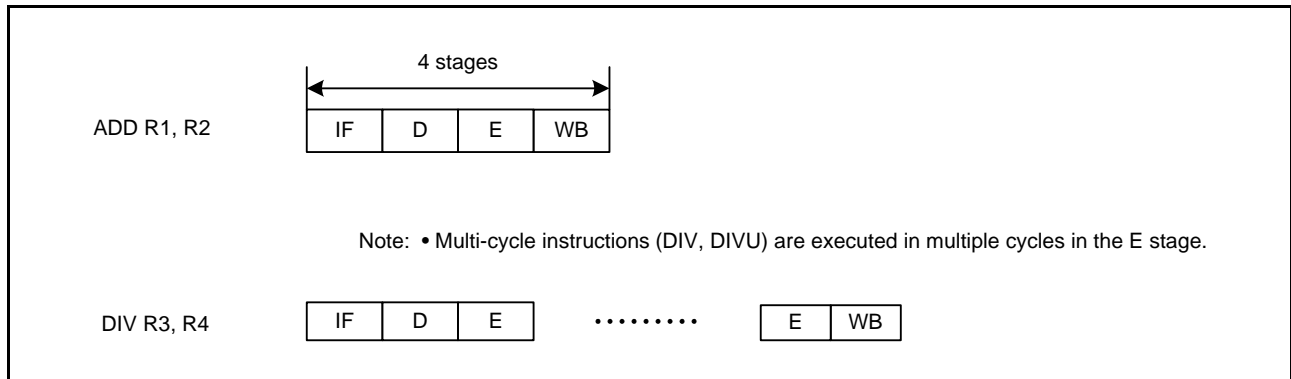


Figure 2.7 Operation for Register-Register, Immediate-Register

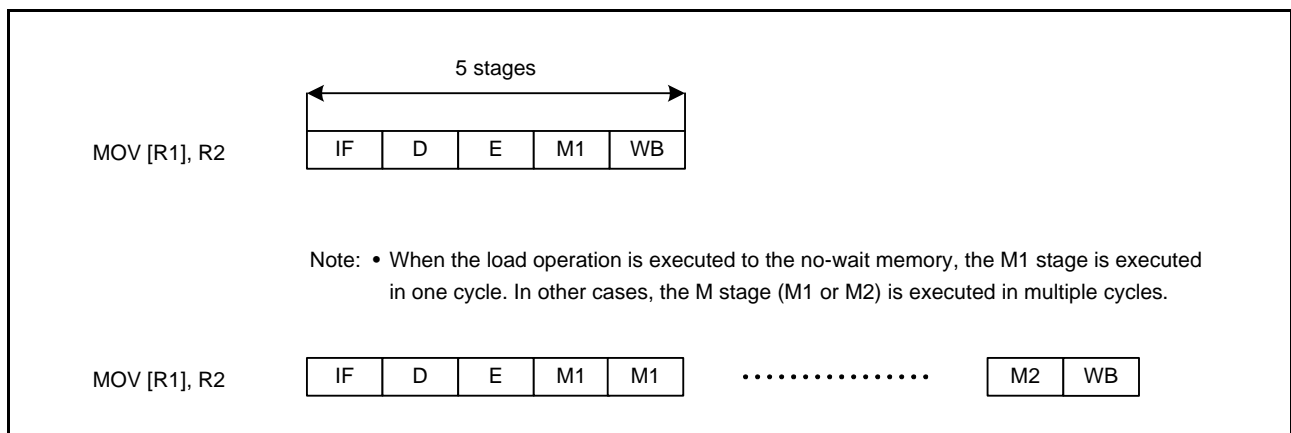


Figure 2.8 Load Operation

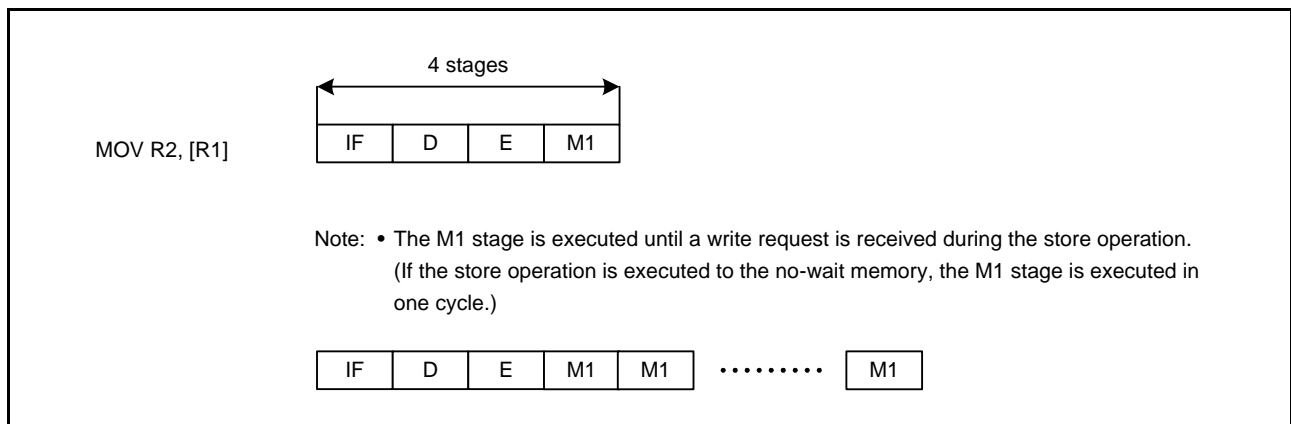


Figure 2.9 Store Operation

### 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)	<ul style="list-style-type: none"> <li>{ADC, ADD, AND, MAX, MIN, MUL, OR, SBB, SUB, XOR} "[Rs], Rd"/"dsp[Rs], Rd"</li> <li>{CMP, TST} "[Rs], Rs2"/"dsp[Rs], Rs2"</li> </ul>	Figure 2.10	3
Arithmetic/logic instructions (division)	<ul style="list-style-type: none"> <li>DIV "[Rs], Rd / dsp[Rs], Rd"</li> <li>DIVU "[Rs], Rd / dsp[Rs], Rd"</li> </ul>	—	5 to 22
Arithmetic/logic instructions (multiplier: 32 × 32 → 64 bits) (register-register, register-immediate)	<ul style="list-style-type: none"> <li>{EMUL, EMULU} "#IMM, Rd"/"Rs, Rd"</li> </ul>	Figure 2.12	2
Arithmetic/logic instructions (multiplier: 32 × 32 → 64 bits) (memory source operand)	<ul style="list-style-type: none"> <li>{EMUL, EMULU} "[Rs], Rd"/"dsp[Rs], Rd"</li> </ul>	—	4
Arithmetic/logic instructions (multiply-and-accumulate operation)	<ul style="list-style-type: none"> <li>RMPA.B</li> <li>RMPA.W</li> <li>RMPA.L</li> </ul>	—	6+7×floor(n/4)+4×(n%4) n: Number of processing bytes*1
Arithmetic/logic instruction (64-bit signed saturation processing for the RMPA instruction)	<ul style="list-style-type: none"> <li>SATR</li> </ul>	—	3
Data transfer instructions (memory-memory transfer)	<ul style="list-style-type: none"> <li>MOV "[Rs], [Rd]"/"dsp[Rs], [Rd]"/"Rs, dsp[Rd]"/"dsp[Rs], dsp[Rd]"</li> <li>PUSH "[Rs]"/"dsp[Rs]"</li> </ul>	Figure 2.11	3
Bit manipulation instructions (memory source operand)	<ul style="list-style-type: none"> <li>{BCLR, BNOT, BSET} "#IMM, [Rd]"/"#IMM, dsp[Rd]"/"Rs, [Rd]"/"Rs, dsp[Rd]"</li> <li>BMCnd "#IMM, [Rd]"/"#IMM, dsp[Rd]"</li> <li>BTST "#IMM, [Rs]"/"#IMM, dsp[Rs]"/"Rs, [Rs2]"/"Rs, dsp[Rs2]"</li> </ul>	Figure 2.11	3
Transfer instruction (load operation)	<ul style="list-style-type: none"> <li>POPC "CR"</li> </ul>	—	Throughput: 3 Latency: 4*2
Transfer instruction (save operation of multiple registers)	<ul style="list-style-type: none"> <li>PUSHM "Rs-Rs2"</li> </ul>	—	n n: Number of registers*3
Transfer instructions (restore operation of multiple registers)	<ul style="list-style-type: none"> <li>POPM "Rd-Rd2"</li> </ul>	—	Throughput: n Latency: n+1 n: Number of registers*2,*4
Transfer instruction (register-register)	<ul style="list-style-type: none"> <li>XCHG "Rs, Rd"</li> </ul>	Figure 2.13	2
Transfer instructions (memory-register)	<ul style="list-style-type: none"> <li>XCHG "[Rs], Rd"/"dsp[Rs], Rd"</li> </ul>	Figure 2.14	2
Branch instructions	<ul style="list-style-type: none"> <li>RTS</li> <li>RTSD "#IMM"</li> <li>RTSD "#IMM, Rd-Rd2"</li> </ul>	—	5 5 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* <sup>5</sup>	• SCMPU	—	$2+4 \times \text{floor}(n/4)+4 \times (n\%4)$ n: Number of comparison bytes* <sup>1</sup>
	• SMOVB	—	n>3? $6+3 \times \text{floor}(n/4)+3 \times (n\%4):$ 2+3n n: Number of transfer bytes* <sup>1</sup>
	• SMOVF, SMOVU	—	$2+3 \times \text{floor}(n/4)+3 \times (n\%4)$ n: Number of transfer bytes* <sup>1</sup>
	• SSTR.B	—	$2+\text{floor}(n/4)+n\%4$ n: Number of transfer bytes* <sup>1</sup>
	• SSTR.W	—	$2+\text{floor}(n/2)+n\%2$ n: Number of transfer words* <sup>1</sup>
	• SSTR.L	—	2+n n: Number of transfer longwords
	• SUNTIL.B, SWHILE.B	—	$3+3 \times \text{floor}(n/4)+3 \times (n\%4)$ n: Number of comparison bytes* <sup>1</sup>
	• SUNTIL.W, SWHILE.W	—	$3+3 \times \text{floor}(n/2)+3 \times (n\%2)$ n: Number of comparison words* <sup>1</sup>
	• 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, see 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 2.15 show the operation of instructions that are converted into basic multiple micro-operations.  
 Note: • mop: Micro-operation, stall: Pipeline stall

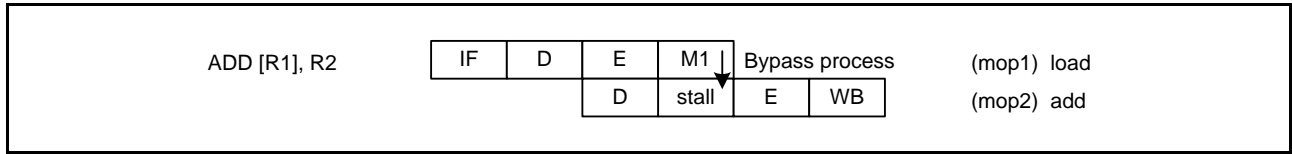


Figure 2.10 Arithmetic/Logic Instruction (Memory Source Operand)

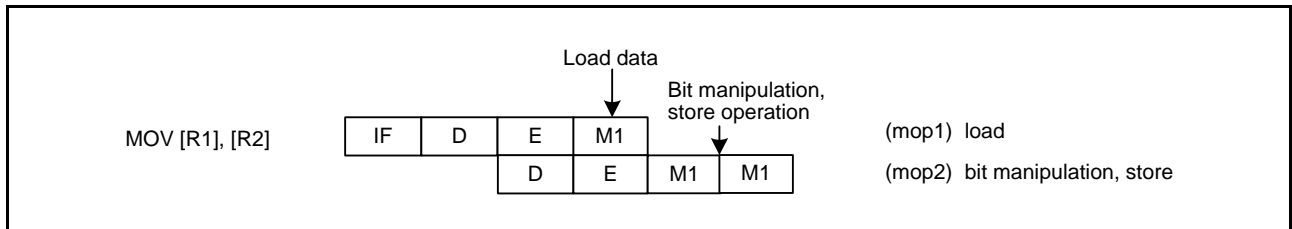


Figure 2.11 MOV Instruction (Memory-Memory), Bit Manipulation Instruction (Memory Source Operand)

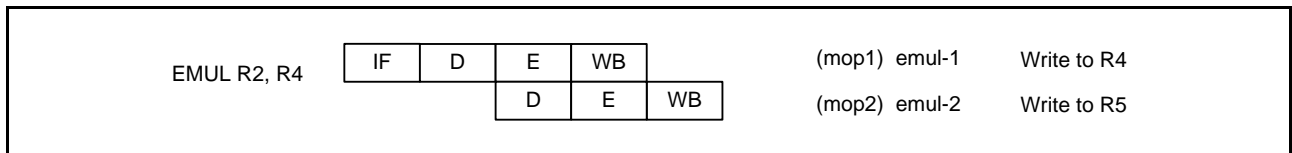


Figure 2.12 EMUL, EMULU Instructions (Register- Register, Register-Immediate)

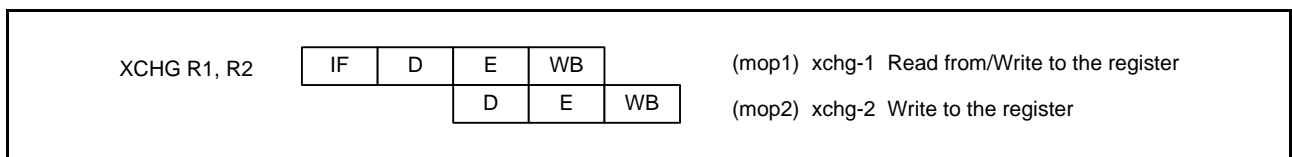


Figure 2.13 XCHG Instruction (Registers)

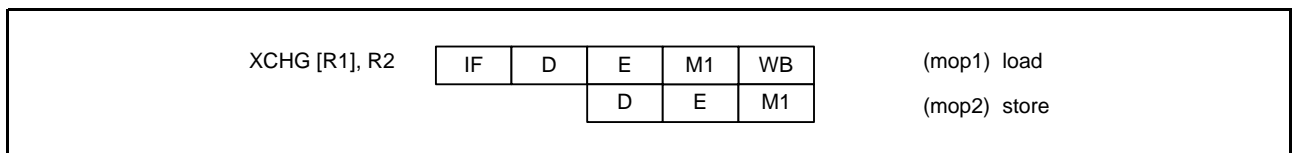


Figure 2.14 XCHG Instruction (Memory Source Operand)

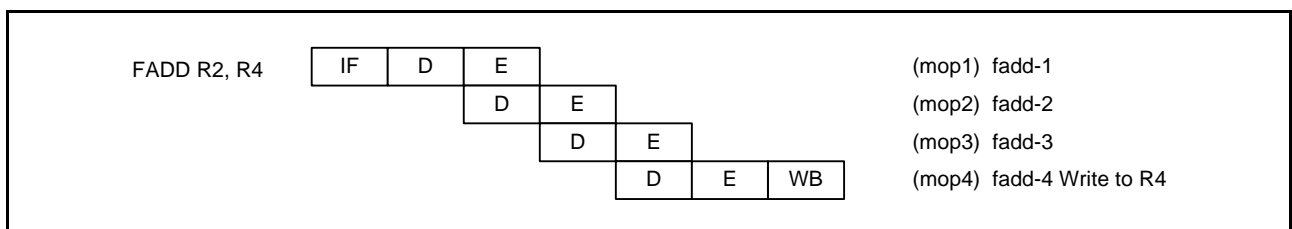


Figure 2.15 Floating-Point Operation Instruction (Register-Register, Immediate-Register)

### 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

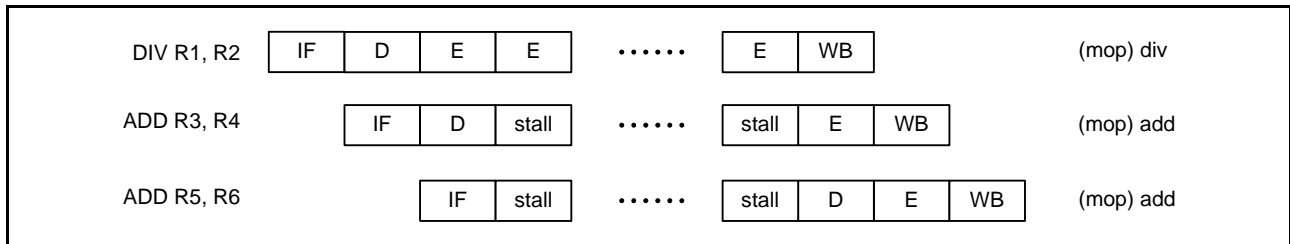


Figure 2.16 When an Instruction which Requires Multiple Cycles is Executed in the E Stage

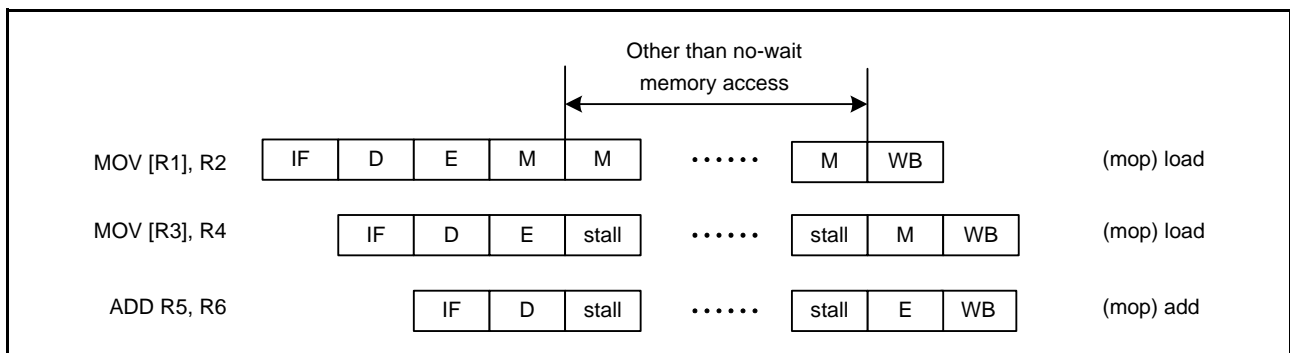


Figure 2.17 When an Instruction which Requires more than One Cycle for its Operand Access is Executed

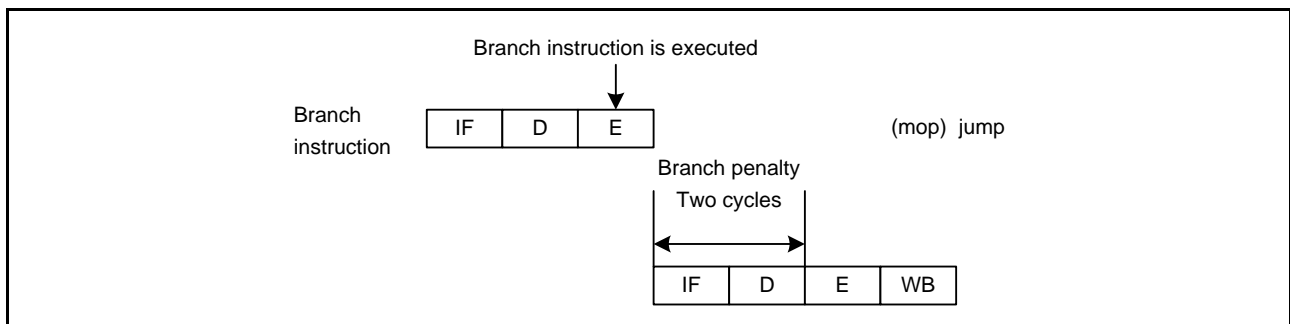


Figure 2.18 When a Branch Instruction is Executed (an Unconditional Branch Instruction is Executed or the Condition is Satisfied for a Conditional Branch Instruction)

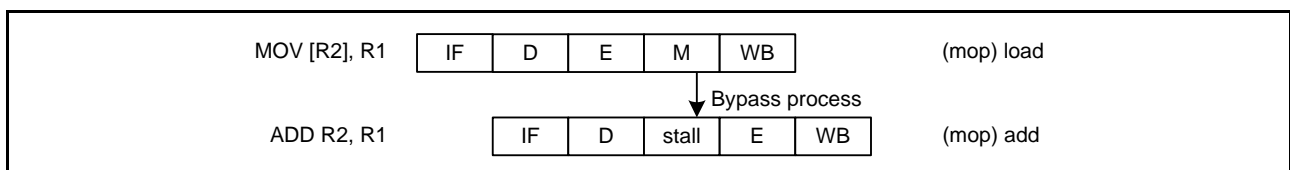


Figure 2.19 When the Subsequent Instruction Uses an Operand Read from the Memory



(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.

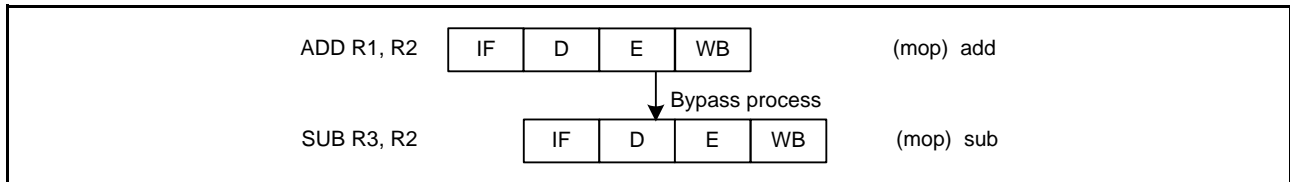


Figure 2.20 Bypass Process

(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.

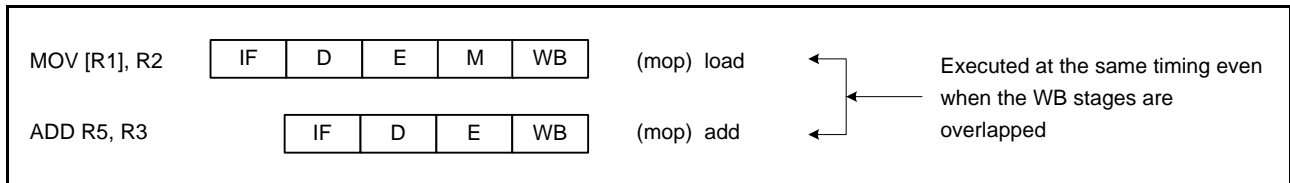


Figure 2.21 When WB Stages for the Memory Load and for the Operation are Overlapped

(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.

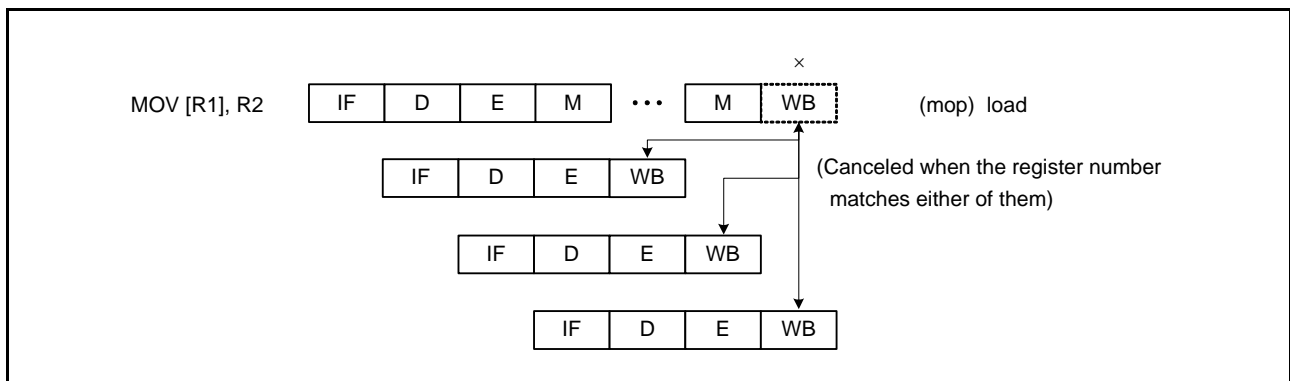


Figure 2.22 When Subsequent Instruction Writes to the Same Register before the End of Memory Load

(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).

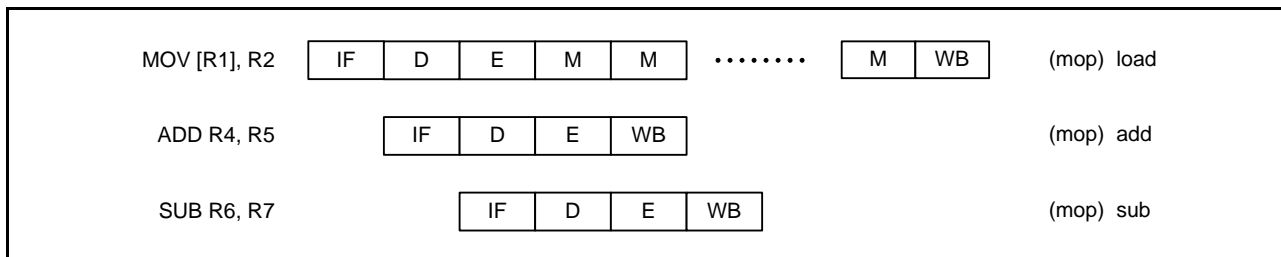


Figure 2.23 When Load Data is not Used by the Subsequent Instruction

### 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 RAM and ROM in products of the RX630 Groups 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, see section 14.3.1, 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 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, PC7) 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. Operation starts with the on-chip ROM (ROM, E2 DataFlash) enabled and the external bus disabled, regardless of the mode in which operation started. Set the SYSCR0.EXBE bit to 1 (external bus enabled) to enable the external bus.

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

Mode-Setting Pin		Operating Mode	SYSCR0 Initial State	
MD*1	PC7*2		ROME	EXBE
High	—	Single-chip mode	1 (On-chip ROM enabled)	0 (External bus disabled)
Low	Low	Boot mode		
	High	USB boot mode		
		User boot mode		

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

Note 2. The PC7 pin can also be used as a general port pin.

Table 3.2 gives a list of the operating mode settings that can be made with system control register 0 (SYSCR0). For details on each of the operating modes, see section 3.3, Details of Operating Modes.

**Table 3.2 Selection of Operating Modes by Register Setting**

SYSCR0		
ROME	EXBE	Operating Mode
0 (On-chip ROM disabled)*1	0 (external bus disabled)	Single-chip mode, user boot mode
1 (On-chip ROM enabled)	0 (external bus disabled)	
0 (On-chip ROM disabled)*1	1 (external bus enabled)	On-chip ROM disabled extended mode
1 (On-chip ROM enabled)	1 (external bus enabled)	On-chip ROM enabled extended mode

Note 1. Once the ROME bit is set to 0, it cannot be reverted to 1.

The endian is selectable in single-chip mode and user boot mode. Endian is set in the given operating mode by using the endian selection bits (MDE[2:0]) in the register indicated in Table 3.3. For the correspondence between the setting and endian, see Table 3.4.

**Table 3.3 Endian Setting Method**

Operating Mode	Endian Setting Method
Single-chip mode	Endian is set in the endian selection register (MDES) in the option-setting memory.
User boot mode/USB boot mode	Endian is set in the endian selection register (MDEB) in the option-setting memory.

**Table 3.4 Selection of Endian**

Setting of the MDE[2:0] Bits	Selected 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

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

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

### 3.2.2 Mode Status Register (MDSR)

Address(es): 0008 0002h

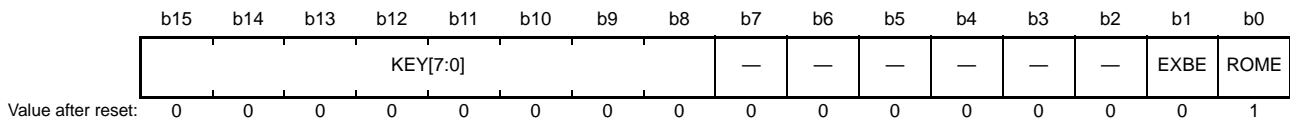
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	UBTS	—	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0	0	0	0/1*1	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	This bit is read as 1.	R
b4 to b1	—	Reserved	These bits are read as 0.	R
b5	UBTS	User Boot Mode Startup Flag	0: Started with single-chip mode. 1: Started with user boot mode.	R
b15 to b6	—	Reserved	These bits are read as 0.	R

Note 1. Depends on the operating mode at startup.

### 3.2.3 System Control Register 0 (SYSCR0)

Address(es): 0008 0006h



Bit	Symbol	Bit Name	Description	R/W
b0	ROME	On-Chip ROM Enable	0: The on-chip ROM is disabled. 1: The on-chip ROM is enabled.	R/W
b1	EXBE	External Bus Enable	0: The external bus is disabled. 1: The external bus is enabled.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	SYSCR0 Key Code	These bits control permission and prohibition of writing to the SYSCR0 register. To modify the SYSCR0 register, write 5Ah to the eight higher-order bits and the desired value to the eight lower-order bits as a 16-bit unit.	R/(W)*1

Note 1. Write data is not retained.

#### ROME Bit (On-Chip ROM Enable)

The ROME bit enables or disables the on-chip ROM (ROM, E2 DataFlash).

Once cleared to 0, it cannot be reverted to 1.

A 0 should not be written to this bit while a program is being executed from the on-chip ROM. After writing a 0 to this bit, be sure to disable the on-chip ROM by changing the ROME bit to 0 before proceeding with further processing.

#### EXBE Bit (External Bus Enable)

The EXBE bit enables or disables the external bus.

Do not write 0 to this bit while a program is running from an external address space. Write 0 to this bit after access to the external bus is completed. Furthermore, when an external address space is included in the range of transfer by the DMAC, prohibit DMA transfer before writing 0 to this bit.

After writing to the EXBE bit, confirm that its value has actually changed before proceeding with further processing.

When the EXBE bit is set to 1, the related I/O port settings must also be changed as required. For details, see section 21, Multi-Function Pin Controller (MPC).

### 3.2.4 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

#### RAME Bit (RAM Enable)

The RAME bit enables or disables the RAM.

The RAME bit is initialized to 1 after a reset is released.

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 (on-chip RAM disabled) to 1 (on-chip RAM enabled), make sure that the RAME bit is 1 before the access.

Even when the RAME bit is cleared to 0, the RAM retains its value. To retain the value in the RAM, keep the specified RAM standby voltage (VRAM). For details, see section 45, Electrical Characteristics.

### 3.3 Details of Operating Modes

#### 3.3.1 Single-Chip Mode

In single-chip mode, the external bus is disabled (SYSCR0.EXBE bit = 0) so that all I/O port pins are available for use as input or output port pins, inputs or outputs for peripheral functions, or as interrupt inputs.

If the high level is on the MD pin at the time of release from the reset state, the chip starts in single-chip mode. The on-chip ROM is enabled (SYSCR0.ROME bit = 1) at this time. The on-chip ROM can be disabled by software (by clearing the SYSCR0.ROME bit to 0), but it cannot be reenabled (by setting the SYSCR0.ROME bit to 1) once this is done.

Setting the SYSCR0.EXBE bit to 1 (enabling the external bus) causes a transition to on-chip ROM enabled extended mode or to on-chip ROM disabled extended mode, making the external bus available.

#### 3.3.2 On-Chip ROM Enabled Extended Mode

In this mode, the on-chip ROM is enabled (SYSCR0.ROME bit = 1) and the external bus extension is enabled (SYSCR0.EXBE bit = 1). This mode allows some I/O ports to be used as data bus input/output, address bus output, or bus control signal input/output. For details, see section 21, Multi-Function Pin Controller (MPC).

After the chip has started up in single-chip mode, setting the SYSCR0.EXBE bit to 1 (external bus enabled) causes it to make the transition to on-chip ROM enabled extended mode.

Writing 0 to the SYSCR0.EXBE bit (external bus disabled) causes a transition to single-chip mode (on-chip ROM enabled).

Writing 0 to the SYSCR0.ROME bit (on-chip ROM disabled) causes a transition to on-chip ROM disabled extended mode.

#### 3.3.3 On-Chip ROM Disabled Extended Mode

In this mode, the on-chip ROM is disabled (SYSCR0.ROME bit = 0) and the external bus extension is enabled (SYSCR0.EXBE bit = 1). This mode allows some I/O ports to be used as data bus input/output, address bus output, or bus control signal input/output. For details, see section 21, Multi-Function Pin Controller (MPC).

After the chip has started up in single-chip mode, setting the SYSCR0.EXBE bit to 1 (external bus enabled) and setting the SYSCR0.ROME bit to 0 (on-chip ROM disabled) causes it to make the transition to on-chip ROM disabled extended mode.

In this mode, the on-chip ROM cannot be enabled by setting the SYSCR0.ROME bit to 1.

Writing 0 to the SYSCR0.EXBE bit (external bus disabled) causes a transition to single-chip mode (on-chip ROM disabled).

#### 3.3.4 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, E2 DataFlash) can be modified from outside the MCU by using a universal asynchronous receiver/transmitter (SCI1). For details, see section 43, Flash Memory.

The chip starts up in boot mode if the low level is on both the MD and PC7 pins on release from the reset state.

#### 3.3.5 USB Boot Mode

In this mode, the on-chip flash memory modifying program (USB boot program) stored in the user boot area at the time of shipping operates. On-chip flash memory (ROM, E2 DataFlash) can be modified from outside the MCU by using the USB. For details, see section 43, Flash Memory.

The chip starts up in USB boot mode if the low level is on the MD pin, and the high level is on the PC7 pin on release from the reset state.

#### 3.3.6 User Boot Mode

In user boot mode, an on-chip flash memory modifying program (user boot program) created by the user operates. After release from the reset state, the chip starts up in a state equivalent to single-chip mode.

After programming the prescribed values for UB code A and the UB code B, the chip starts up in user boot mode if the low level is on the MD pin and the high level is on the PC7 pin on release from the reset state. For UB code A and UB code b, see section 7, Option-Setting Memory.

After the chip has started up in user boot mode, setting the SYSCR0.EXBE bit to 1 (external bus enabled) causes it to make the transition to on-chip ROM enabled extended mode.

At the time of shipping, USB boot program is stored in the user boot area. To store user boot program, erase the USB boot program. For details, see section 43, Flash Memory.

Note 1. In user mode, do not make a transition to software standby mode or deep software standby mode.

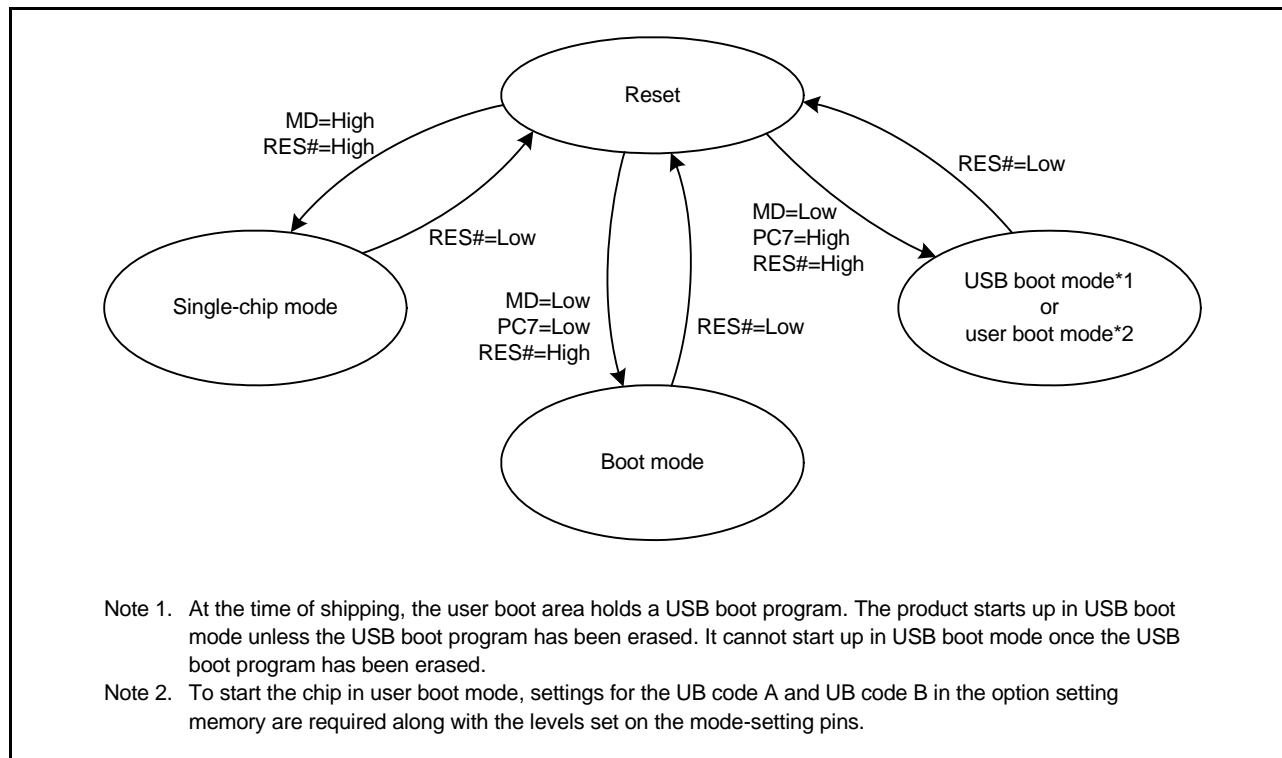
Note 2. The setting in the OFS0/OFS1 registers is ineffective in user boot mode, and the value becomes FFFF FFFFh.



### 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 and the PC7 pin.



**Figure 3.1 Mode-Setting Pin Level and Operating Mode**

### 3.4.2 Operating Mode Transitions According to Register Setting

Figure 3.2 shows operating mode transitions according to the setting of the ROME and EXBE bits in SYSCR0.

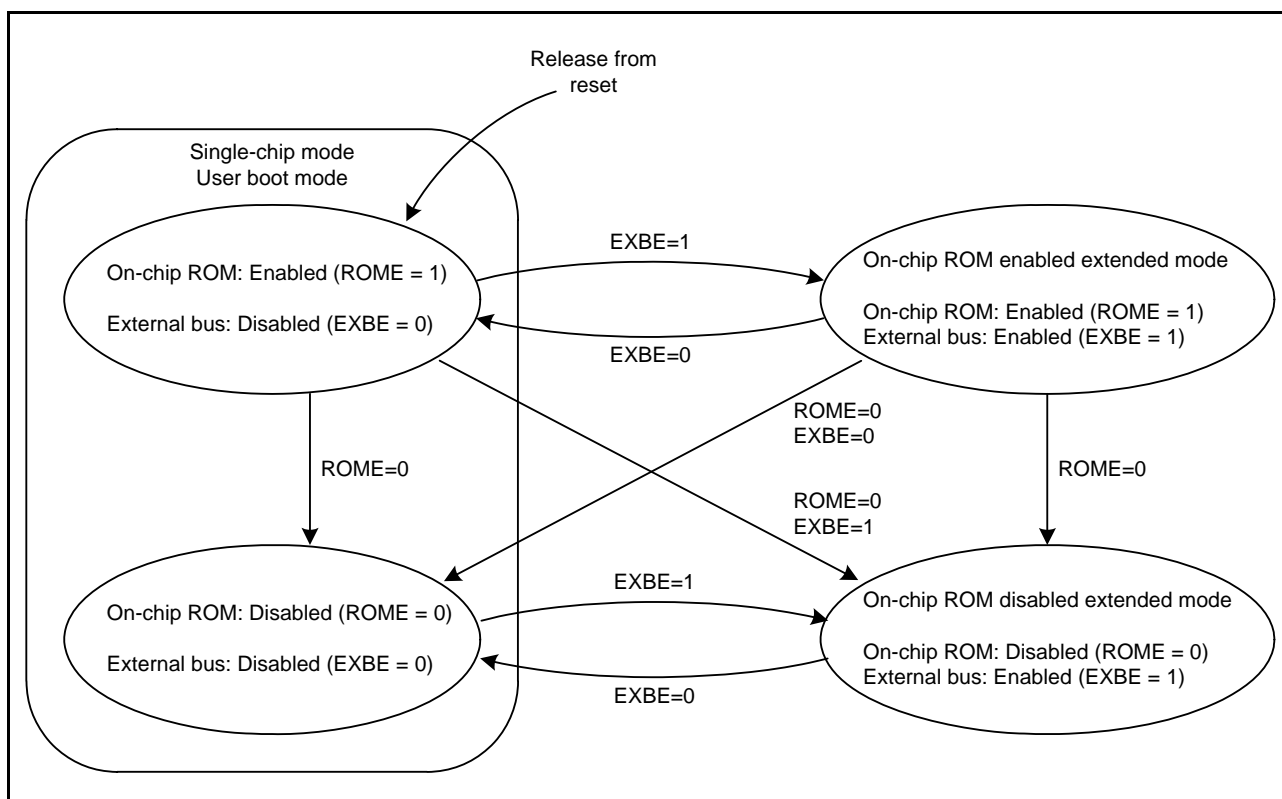


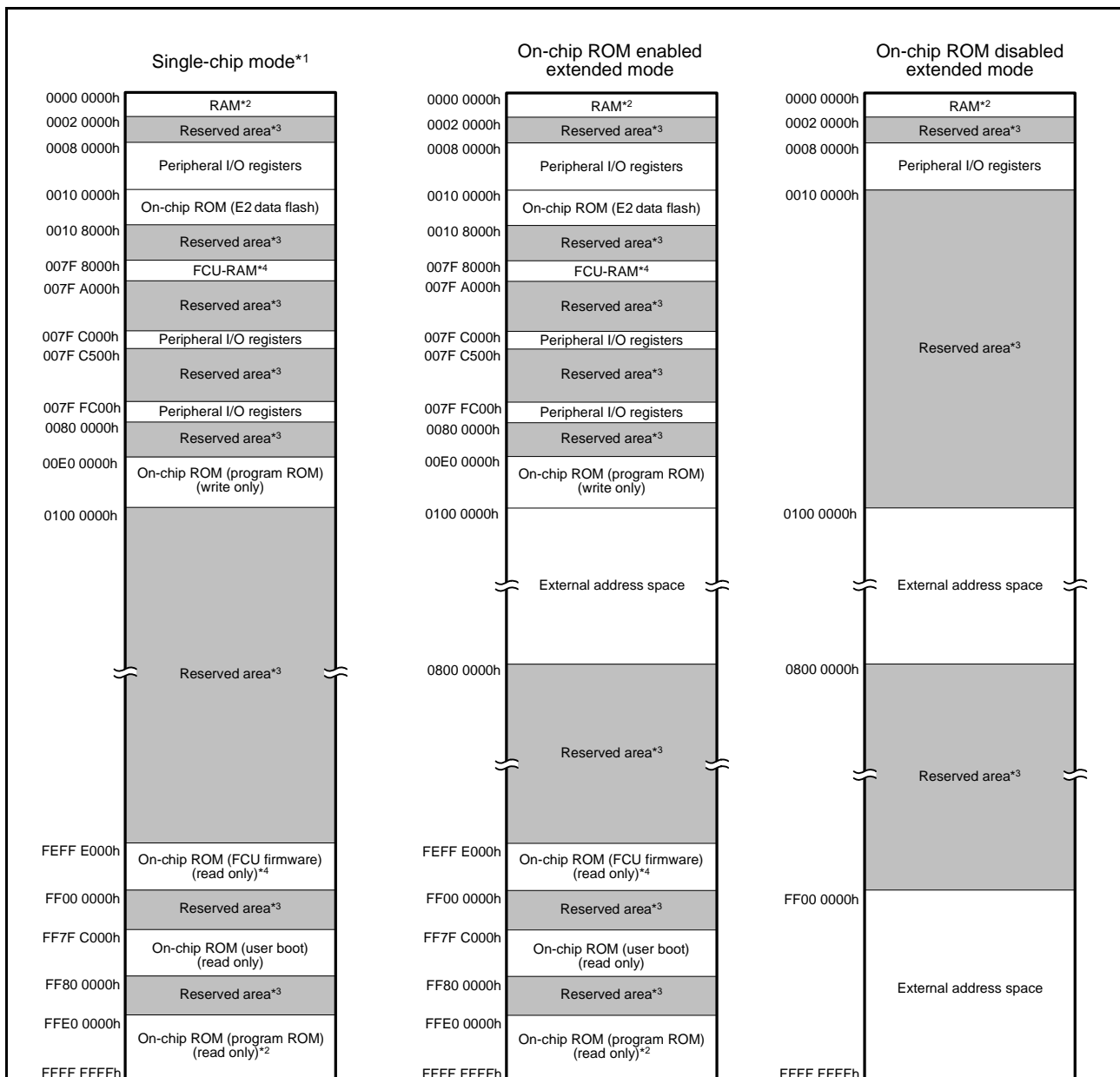
Figure 3.2 Setting of SYSCR0.ROME and EXBE Bits and Operating Modes

## 4. Address Space

### 4.1 Address Space

This LSI 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 in the respective operating modes. Accessible areas will differ according to the operating mode and states of control bits.



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

ROM (bytes)		RAM (bytes)	
Capacity	Address	Capacity	Address
2 M	FFE0 0000h to FFFF FFFFh	128 K	0000 0000h to 0001 FFFFh
1.5 M	FFE8 0000h to FFFF FFFFh		
1 M	FFF0 0000h to FFFF FFFFh	96 K	0000 0000h to 0001 7FFFh
768 K	FFF4 0000h to FFFF FFFFh		
512 K	FFF8 0000h to FFFF FFFFh	64 K	0000 0000h to 0000 FFFFh
384 K	FFFA 0000h to FFFF FFFFh		

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

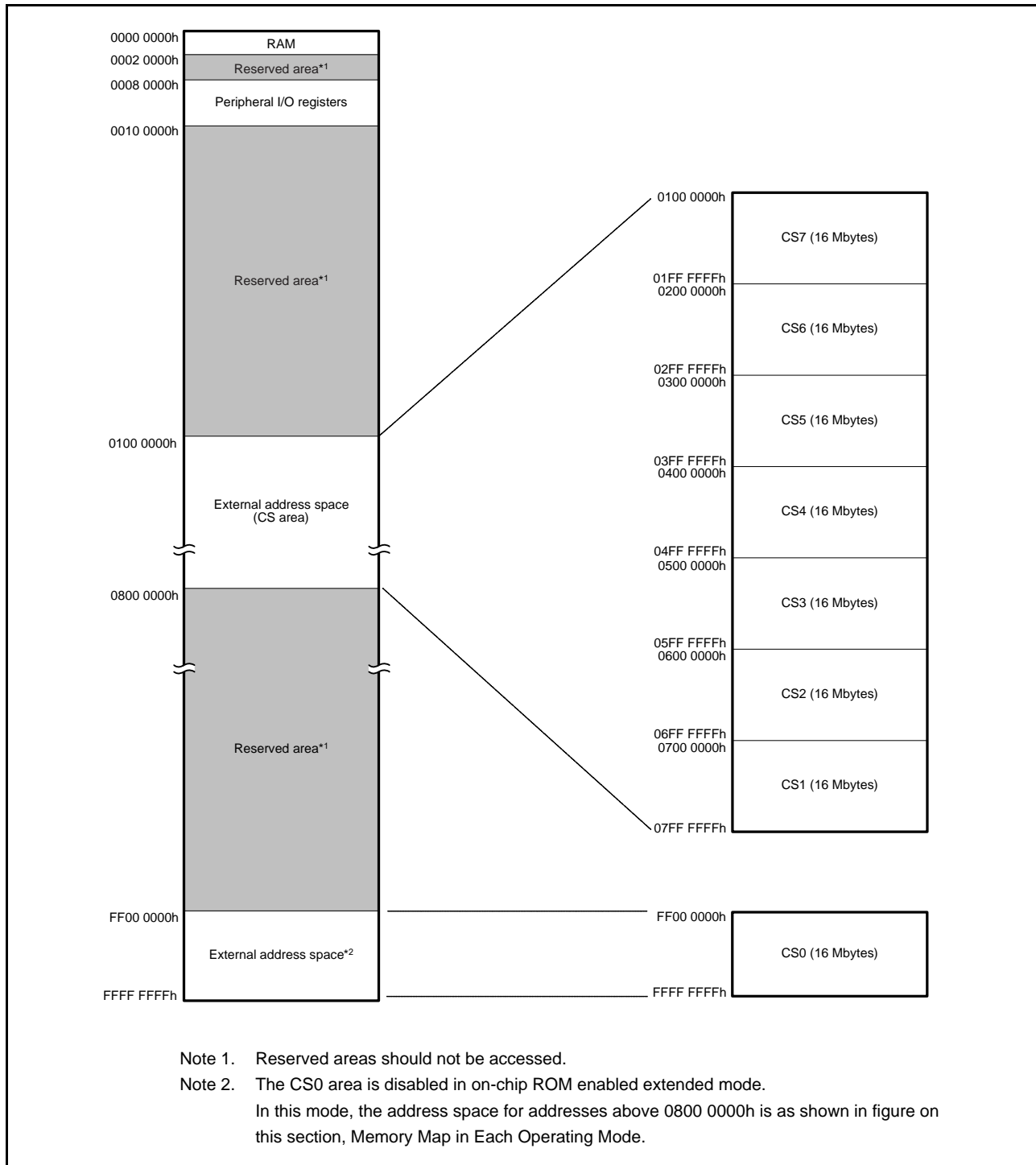
Note 3. Reserved areas should not be accessed.  
 Note 4. For details on the FCU, see section 43, Flash Memory.

Figure 4.1 Memory Map in Each Operating Mode

### 4.2 External Address Space

The external address space is divided into up to eight CS areas (CS0 to CS7), each corresponding to the CSn# signal output from a CSn# (n = 0 to 7) pin.

Figure 4.2 shows the address ranges corresponding to the individual CS areas (CS0 to CS7) in on-chip ROM disabled extended mode.



**Figure 4.2 Correspondence between External Address Spaces and CS Areas (In On-Chip ROM Disabled Extended Mode)**

## 5. I/O Registers

This section gives information on the on-chip I/O register addresses. The information is given as shown below. Notes on writing to registers are also given at the end.

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

- Registers are listed from the lower allocation addresses.
- Registers are classified according to module symbols.
- The number of access cycles indicates the number of cycles based on the specified reference 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.

- Write to an I/O register.
- Read the value from the I/O register to a general register.
- Execute the operation using the value read.
- 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 the number of I/O register access cycles, refer to 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

$$\begin{aligned} \text{Number of access cycles to I/O registers} = & \text{Number of bus cycles for internal main bus 1} + \\ & \text{Number of divided clock synchronization cycles} + \\ & \text{Number of bus cycles for internal peripheral busses 1 to 6} \end{aligned}$$

The number of bus cycles of internal peripheral bus 1 to 6 differs according to the register to be accessed. When peripheral functions connected to internal peripheral bus 2 to 6 or registers for the external bus control unit (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, BCLK) 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 states 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.

In the external bus control unit, the sum of the number of bus cycles for internal main bus 1 and the number of divided clock synchronization cycles will be one cycle of BCLK at a maximum. Therefore, one BCLK is added to the number of access cycles shown in Table 5.1.

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 (DMAC or DTC).

## 5.1 I/O Register Addresses (Address Order)

Table 5.1 List of I/O Registers (Address Order) (1/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 0000h	SYSTEM	Mode monitor register	MDMONR	16	16	3	ICLK	Operating Modes	124
0008 0002h	SYSTEM	Mode status register	MDSR	16	16	3	ICLK		124
0008 0006h	SYSTEM	System control register 0	SYSCR0	16	16	3	ICLK		125
0008 0008h	SYSTEM	System control register 1	SYSCR1	16	16	3	ICLK		126
0008 000Ch	SYSTEM	Standby control register	SBYCR	16	16	3	ICLK	Low Power Consumption	252
0008 0010h	SYSTEM	Module stop control register A	MSTPCRA	32	32	3	ICLK		253
0008 0014h	SYSTEM	Module stop control register B	MSTPCRB	32	32	3	ICLK		254
0008 0018h	SYSTEM	Module stop control register C	MSTPCRC	32	32	3	ICLK		256
0008 0020h	SYSTEM	System clock control register	SCKCR	32	32	3	ICLK	Clock Generation Circuit	216
0008 0024h	SYSTEM	System clock control register 2	SCKCR2	16	16	3	ICLK		218
0008 0026h	SYSTEM	System clock control register 3	SCKCR3	16	16	3	ICLK		219
0008 0028h	SYSTEM	PLL control register	PLLCR	16	16	3	ICLK		220
0008 002Ah	SYSTEM	PLL control register 2	PLLCR2	8	8	3	ICLK		221
0008 0030h	SYSTEM	External bus clock control register	BCKCR	8	8	3	ICLK		222
0008 0032h	SYSTEM	Main clock oscillator control register	MOSCCR	8	8	3	ICLK		223
0008 0033h	SYSTEM	Sub-clock oscillator control register	SOSCCR	8	8	3	ICLK		224
0008 0034h	SYSTEM	Low-speed on-chip oscillator control register	LOCOCR	8	8	3	ICLK		225
0008 0035h	SYSTEM	IWDT-dedicated on-chip oscillator control register	ILOCOCR	8	8	3	ICLK		226
0008 0036h	SYSTEM	High-speed on-chip oscillator control register	HOCOCR	8	8	3	ICLK		227
0008 0040h	SYSTEM	Oscillation stop detection control register	OSTDCR	8	8	3	ICLK		228
0008 0041h	SYSTEM	Oscillation stop detection status register	OSTDSR	8	8	3	ICLK		229
0008 00A0h	SYSTEM	Operating power control register	OPCCR	8	8	3	ICLK		Low Power Consumption
0008 00A1h	SYSTEM	Sleep mode return clock source switching register	RSTCKCR	8	8	3	ICLK	259	
0008 00A2h	SYSTEM	Main clock oscillator wait control register	MOSCWTCR	8	8	3	ICLK	260	
0008 00A3h	SYSTEM	Sub-clock oscillator wait control register	SOSCWTCR	8	8	3	ICLK	261	
0008 00A6h	SYSTEM	PLL wait control register	PLLWTCR	8	8	3	ICLK	263	
0008 00C0h	SYSTEM	Reset status register 2	RSTSR2	8	8	3	ICLK	Resets	183
0008 00C2h	SYSTEM	Software reset register	SWRR	16	16	3	ICLK		184
0008 00E0h	SYSTEM	Voltage monitoring 1 circuit control register 1	LVD1CR1	8	8	3	ICLK	LVDA	202
0008 00E1h	SYSTEM	Voltage monitoring 1 circuit status register	LVD1SR	8	8	3	ICLK		202
0008 00E2h	SYSTEM	Voltage monitoring 2 circuit control register 1	LVD2CR1	8	8	3	ICLK		203
0008 00E3h	SYSTEM	Voltage monitoring 2 circuit status register	LVD2SR	8	8	3	ICLK		203
0008 03FEh	SYSTEM	Protect register	PRCR	16	16	3	ICLK	Register Write Protection Function	296
0008 1300h	BSC	Bus error status clear register	BERCLR	8	8	2	ICLK	Buses	377
0008 1304h	BSC	Bus error monitoring enable register	BEREN	8	8	2	ICLK		377
0008 1308h	BSC	Bus error status register 1	BERSR1	8	8	2	ICLK		378
0008 130Ah	BSC	Bus error status register 2	BERSR2	16	16	2	ICLK		378
0008 1310h	BSC	Bus priority control register	BUSPRI	16	16	2	ICLK		379
0008 2000h	DMAC0	DMA source address register	DMSAR	32	32	2	ICLK	DMACA	436
0008 2004h	DMAC0	DMA destination address register	DMDAR	32	32	2	ICLK		436
0008 2008h	DMAC0	DMA transfer count register	DMCRA	32	32	2	ICLK		437
0008 200Ch	DMAC0	DMA block transfer count register	DMCRB	16	16	2	ICLK		438
0008 2010h	DMAC0	DMA transfer mode register	DMTMD	16	16	2	ICLK		439
0008 2013h	DMAC0	DMA interrupt setting register	DMINT	8	8	2	ICLK		440



Table 5.1 List of I/O Registers (Address Order) (2/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 2014h	DMAC0	DMA address mode register	DMAMD	16	16	2	ICLK	DMACA	442
0008 2018h	DMAC0	DMA offset register	DMOFR	32	32	2	ICLK		445
0008 201Ch	DMAC0	DMA transfer enable register	DMCNT	8	8	2	ICLK		445
0008 201Dh	DMAC0	DMA software start register	DMREQ	8	8	2	ICLK		446
0008 201Eh	DMAC0	DMA status register	DMSTS	8	8	2	ICLK		447
0008 201Fh	DMAC0	DMA activation source flag control register	DMCSL	8	8	2	ICLK		448
0008 2040h	DMAC1	DMA source address register	DMSAR	32	32	2	ICLK		436
0008 2044h	DMAC1	DMA destination address register	DMDAR	32	32	2	ICLK		436
0008 2048h	DMAC1	DMA transfer count register	DMCRA	32	32	2	ICLK		437
0008 204Ch	DMAC1	DMA block transfer count register	DMCRB	16	16	2	ICLK		438
0008 2050h	DMAC1	DMA transfer mode register	DMTMD	16	16	2	ICLK		439
0008 2053h	DMAC1	DMA interrupt setting register	DMINT	8	8	2	ICLK		440
0008 2054h	DMAC1	DMA address mode register	DMAMD	16	16	2	ICLK		442
0008 205Ch	DMAC1	DMA transfer enable register	DMCNT	8	8	2	ICLK		445
0008 205Dh	DMAC1	DMA software start register	DMREQ	8	8	2	ICLK		445
0008 205Eh	DMAC1	DMA status register	DMSTS	8	8	2	ICLK		447
0008 205Fh	DMAC1	DMA activation source flag control register	DMCSL	8	8	2	ICLK		448
0008 2080h	DMAC2	DMA source address register	DMSAR	32	32	2	ICLK		436
0008 2084h	DMAC2	DMA destination address register	DMDAR	32	32	2	ICLK		436
0008 2088h	DMAC2	DMA transfer count register	DMCRA	32	32	2	ICLK		437
0008 208Ch	DMAC2	DMA block transfer count register	DMCRB	16	16	2	ICLK		438
0008 2090h	DMAC2	DMA transfer mode register	DMTMD	16	16	2	ICLK		439
0008 2093h	DMAC2	DMA interrupt setting register	DMINT	8	8	2	ICLK		440
0008 2094h	DMAC2	DMA address mode register	DMAMD	16	16	2	ICLK	442	
0008 209Ch	DMAC2	DMA transfer enable register	DMCNT	8	8	2	ICLK	445	
0008 209Dh	DMAC2	DMA software start register	DMREQ	8	8	2	ICLK	446	
0008 209Eh	DMAC2	DMA status register	DMSTS	8	8	2	ICLK	447	
0008 209Fh	DMAC2	DMA activation source flag control register	DMCSL	8	8	2	ICLK	448	
0008 20C0h	DMAC3	DMA source address register	DMSAR	32	32	2	ICLK	436	
0008 20C4h	DMAC3	DMA destination address register	DMDAR	32	32	2	ICLK	436	
0008 20C8h	DMAC3	DMA transfer count register	DMCRA	32	32	2	ICLK	437	
0008 20CCh	DMAC3	DMA block transfer count register	DMCRB	16	16	2	ICLK	438	
0008 20D0h	DMAC3	DMA transfer mode register	DMTMD	16	16	2	ICLK	439	
0008 20D3h	DMAC3	DMA interrupt setting register	DMINT	8	8	2	ICLK	440	
0008 20D4h	DMAC3	DMA address mode register	DMAMD	16	16	2	ICLK	442	
0008 20DCh	DMAC3	DMA transfer enable register	DMCNT	8	8	2	ICLK	445	
0008 20DDh	DMAC3	DMA software start register	DMREQ	8	8	2	ICLK	446	
0008 20DEh	DMAC3	DMA status register	DMSTS	8	8	2	ICLK	447	
0008 20DFh	DMAC3	DMA activation source flag control register	DMCSL	8	8	2	ICLK	448	
0008 2200h	DMAC	DMA module activation register	DMAST	8	8	2	ICLK	449	
0008 2400h	DTC	DTC control register	DTCCR	8	8	2	ICLK	DTCa	477
0008 2404h	DTC	DTC vector base register	DTCVBR	32	32	2	ICLK		478
0008 2408h	DTC	DTC address mode register	DTCADMOD	8	8	2	ICLK		478
0008 240Ch	DTC	DTC module start register	DTCST	8	8	2	ICLK		479
0008 240Eh	DTC	DTC status register	DTCSTS	16	16	2	ICLK		480
0008 3002h	BSC	CS0 mode register	CS0MOD	16	16	1, 2	BCLK	Buses	369
0008 3004h	BSC	CS0 wait control register 1	CS0WCR1	32	32	1, 2	BCLK		371
0008 3008h	BSC	CS0 wait control register 2	CS0WCR2	32	32	1, 2	BCLK		374
0008 3012h	BSC	CS1 mode register	CS1MOD	16	16	1, 2	BCLK		369
0008 3014h	BSC	CS1 wait control register 1	CS1WCR1	32	32	1, 2	BCLK		371

Table 5.1 List of I/O Registers (Address Order) (3/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 3018h	BSC	CS1 wait control register 2	CS1WCR2	32	32	1, 2 BCLK		Buses	374
0008 3022h	BSC	CS2 mode register	CS2MOD	16	16	1, 2 BCLK			369
0008 3024h	BSC	CS2 wait control register 1	CS2WCR1	32	32	1, 2 BCLK			371
0008 3028h	BSC	CS2 wait control register 2	CS2WCR2	32	32	1, 2 BCLK			374
0008 3032h	BSC	CS3 mode register	CS3MOD	16	16	1, 2 BCLK			369
0008 3034h	BSC	CS3 wait control register 1	CS3WCR1	32	32	1, 2 BCLK			371
0008 3038h	BSC	CS3 wait control register 2	CS3WCR2	32	32	1, 2 BCLK			374
0008 3042h	BSC	CS4 mode register	CS4MOD	16	16	1, 2 BCLK			369
0008 3044h	BSC	CS4 wait control register 1	CS4WCR1	32	32	1, 2 BCLK			371
0008 3048h	BSC	CS4 wait control register 2	CS4WCR2	32	32	1, 2 BCLK			374
0008 3052h	BSC	CS5 mode register	CS5MOD	16	16	1, 2 BCLK			369
0008 3054h	BSC	CS5 wait control register 1	CS5WCR1	32	32	1, 2 BCLK			371
0008 3058h	BSC	CS5 wait control register 2	CS5WCR2	32	32	1, 2 BCLK			374
0008 3062h	BSC	CS6 mode register	CS6MOD	16	16	1, 2 BCLK			369
0008 3064h	BSC	CS6 wait control register 1	CS6WCR1	32	32	1, 2 BCLK			371
0008 3068h	BSC	CS6 wait control register 2	CS6WCR2	32	32	1, 2 BCLK			374
0008 3072h	BSC	CS7 mode register	CS7MOD	16	16	1, 2 BCLK			369
0008 3074h	BSC	CS7 wait control register 1	CS7WCR1	32	32	1, 2 BCLK			371
0008 3078h	BSC	CS7 wait control register 2	CS7WCR2	32	32	1, 2 BCLK			374
0008 3802h	BSC	CS0 control register	CS0CR	16	16	1, 2 BCLK			364
0008 380Ah	BSC	CS0 recovery cycle register	CS0REC	16	16	1, 2 BCLK			365
0008 3812h	BSC	CS1 control register	CS1CR	16	16	1, 2 BCLK			364
0008 381Ah	BSC	CS1 recovery cycle register	CS1REC	16	16	1, 2 BCLK			365
0008 3822h	BSC	CS2 control register	CS2CR	16	16	1, 2 BCLK			364
0008 382Ah	BSC	CS2 recovery cycle register	CS2REC	16	16	1, 2 BCLK			365
0008 3832h	BSC	CS3 control register	CS3CR	16	16	1, 2 BCLK			364
0008 383Ah	BSC	CS3 recovery cycle register	CS3REC	16	16	1, 2 BCLK			365
0008 3842h	BSC	CS4 control register	CS4CR	16	16	1, 2 BCLK			364
0008 384Ah	BSC	CS4 recovery cycle register	CS4REC	16	16	1, 2 BCLK			365
0008 3852h	BSC	CS5 control register	CS5CR	16	16	1, 2 BCLK			364
0008 385Ah	BSC	CS5 recovery cycle register	CS5REC	16	16	1, 2 BCLK			365
0008 3862h	BSC	CS6 control register	CS6CR	16	16	1, 2 BCLK			364
0008 386Ah	BSC	CS6 recovery cycle register	CS6REC	16	16	1, 2 BCLK			365
0008 3872h	BSC	CS7 control register	CS7CR	16	16	1, 2 BCLK		364	
0008 387Ah	BSC	CS7 recovery cycle register	CS7REC	16	16	1, 2 BCLK		365	
0008 3880h	BSC	CS recovery cycle insertion enable register	CSRECEN	16	16	1, 2 BCLK		367	
0008 6400h	MPU	Region-0 start page number register	RSPAGE0	32	32	1ICLK		MPU	418
0008 6404h	MPU	Region-0 end page number register	REPAGE0	32	32	1ICLK			419
0008 6408h	MPU	Region-1 start page number register	RSPAGE1	32	32	1ICLK			418
0008 640Ch	MPU	Region-1 end page number register	REPAGE1	32	32	1ICLK			419
0008 6410h	MPU	Region-2 start page number register	RSPAGE2	32	32	1ICLK			418
0008 6414h	MPU	Region-2 end page number register	REPAGE2	32	32	1ICLK			419
0008 6418h	MPU	Region-3 start page number register	RSPAGE3	32	32	1ICLK			418
0008 641Ch	MPU	Region-3 end page number register	REPAGE3	32	32	1ICLK			419
0008 6420h	MPU	Region-4 start page number register	RSPAGE4	32	32	1ICLK			418
0008 6424h	MPU	Region-4 end page number register	REPAGE4	32	32	1ICLK			419
0008 6428h	MPU	Region-5 start page number register	RSPAGE5	32	32	1ICLK			418
0008 642Ch	MPU	Region-5 end page number register	REPAGE5	32	32	1ICLK			419
0008 6430h	MPU	Region-6 start page number register	RSPAGE6	32	32	1ICLK			418
0008 6434h	MPU	Region-6 end page number register	REPAGE6	32	32	1ICLK			419

Table 5.1 List of I/O Registers (Address Order) (4/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page	
						ICLK ≥ PCLK	ICLK < PCLK			
0008 6438h	MPU	Region-7 start page number register	RSPAGE7	32	32	1ICLK		MPU	418	
0008 643Ch	MPU	Region-7 end page number register	REPAGE7	32	32	1ICLK			419	
0008 6500h	MPU	Memory-protection enable register	MPEN	32	32	1ICLK			420	
0008 6504h	MPU	Background access control register	MPBAC	32	32	1ICLK			421	
0008 6508h	MPU	Memory-protection error status-clearing register	MPECLR	32	32	1ICLK			422	
0008 650Ch	MPU	Memory-protection error status register	MPESTS	32	32	1ICLK			423	
0008 6514h	MPU	Data memory-protection error address register	MPDEA	32	32	1ICLK			424	
0008 6520h	MPU	Region search address register	MPSA	32	32	1ICLK			424	
0008 6524h	MPU	Region search operation register	MPOPS	16	16	1ICLK			425	
0008 6526h	MPU	Region invalidation operation register	MPOPI	16	16	1ICLK			425	
0008 6528h	MPU	Instruction-hit region register	MHITI	32	32	1ICLK			426	
0008 652Ch	MPU	Data-hit region register	MHITD	32	32	1ICLK			427	
0008 7010h	ICU	Interrupt request register 016	IR016	8	8	2 ICLK			ICUb	308
0008 7015h	ICU	Interrupt request register 021	IR021	8	8	2 ICLK				308
0008 7017h	ICU	Interrupt request register 023	IR023	8	8	2 ICLK		308		
0008 701Bh	ICU	Interrupt request register 027	IR027	8	8	2 ICLK		308		
0008 701Ch	ICU	Interrupt request register 028	IR028	8	8	2 ICLK		308		
0008 701Dh	ICU	Interrupt request register 029	IR029	8	8	2 ICLK		308		
0008 701Eh	ICU	Interrupt request register 030	IR030	8	8	2 ICLK		308		
0008 701Fh	ICU	Interrupt request register 031	IR031	8	8	2 ICLK		308		
0008 7021h	ICU	Interrupt request register 033	IR033	8	8	2 ICLK		308		
0008 7022h	ICU	Interrupt request register 034	IR034	8	8	2 ICLK		308		
0008 7023h	ICU	Interrupt request register 035	IR035	8	8	2 ICLK		308		
0008 7027h	ICU	Interrupt request register 039	IR039	8	8	2 ICLK		308		
0008 7028h	ICU	Interrupt request register 040	IR040	8	8	2 ICLK		308		
0008 7029h	ICU	Interrupt request register 041	IR041	8	8	2 ICLK		308		
0008 702Ah	ICU	Interrupt request register 042	IR042	8	8	2 ICLK		308		
0008 702Bh	ICU	Interrupt request register 043	IR043	8	8	2 ICLK		308		
0008 702Ch	ICU	Interrupt request register 044	IR044	8	8	2 ICLK		308		
0008 702Dh	ICU	Interrupt request register 045	IR045	8	8	2 ICLK		308		
0008 702Eh	ICU	Interrupt request register 046	IR046	8	8	2 ICLK		308		
0008 702Fh	ICU	Interrupt request register 047	IR047	8	8	2 ICLK		308		
0008 7030h	ICU	Interrupt request register 048	IR048	8	8	2 ICLK		308		
0008 7031h	ICU	Interrupt request register 049	IR049	8	8	2 ICLK		308		
0008 7032h	ICU	Interrupt request register 050	IR050	8	8	2 ICLK		308		
0008 7033h	ICU	Interrupt request register 051	IR051	8	8	2 ICLK		308		
0008 7034h	ICU	Interrupt request register 052	IR052	8	8	2 ICLK		308		
0008 7035h	ICU	Interrupt request register 053	IR053	8	8	2 ICLK		308		
0008 7036h	ICU	Interrupt request register 054	IR054	8	8	2 ICLK		308		
0008 7037h	ICU	Interrupt request register 055	IR055	8	8	2 ICLK		308		
0008 7038h	ICU	Interrupt request register 056	IR056	8	8	2 ICLK		308		
0008 7039h	ICU	Interrupt request register 057	IR057	8	8	2 ICLK		308		
0008 703Ah	ICU	Interrupt request register 058	IR058	8	8	2 ICLK		308		
0008 703Bh	ICU	Interrupt request register 059	IR059	8	8	2 ICLK		308		
0008 703Eh	ICU	Interrupt request register 062	IR062	8	8	2 ICLK		308		
0008 7040h	ICU	Interrupt request register 064	IR064	8	8	2 ICLK		308		
0008 7041h	ICU	Interrupt request register 065	IR065	8	8	2 ICLK		308		
0008 7042h	ICU	Interrupt request register 066	IR066	8	8	2 ICLK		308		
0008 7043h	ICU	Interrupt request register 067	IR067	8	8	2 ICLK		308		

Table 5.1 List of I/O Registers (Address Order) (5/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 7044h	ICU	Interrupt request register 068	IR068	8	8	2	ICLK	ICUb	308
0008 7045h	ICU	Interrupt request register 069	IR069	8	8	2	ICLK		308
0008 7046h	ICU	Interrupt request register 070	IR070	8	8	2	ICLK		308
0008 7047h	ICU	Interrupt request register 071	IR071	8	8	2	ICLK		308
0008 7048h	ICU	Interrupt request register 072	IR072	8	8	2	ICLK		308
0008 7049h	ICU	Interrupt request register 073	IR073	8	8	2	ICLK		308
0008 704Ah	ICU	Interrupt request register 074	IR074	8	8	2	ICLK		308
0008 704Bh	ICU	Interrupt request register 075	IR075	8	8	2	ICLK		308
0008 704Ch	ICU	Interrupt request register 076	IR076	8	8	2	ICLK		308
0008 704Dh	ICU	Interrupt request register 077	IR077	8	8	2	ICLK		308
0008 704Eh	ICU	Interrupt request register 078	IR078	8	8	2	ICLK		308
0008 704Fh	ICU	Interrupt request register 079	IR079	8	8	2	ICLK		308
0008 705Ah	ICU	Interrupt request register 090	IR090	8	8	2	ICLK		308
0008 705Ch	ICU	Interrupt request register 092	IR092	8	8	2	ICLK		308
0008 705Dh	ICU	Interrupt request register 093	IR093	8	8	2	ICLK		308
0008 7062h	ICU	Interrupt request register 098	IR098	8	8	2	ICLK		308
0008 7066h	ICU	Interrupt request register 102	IR102	8	8	2	ICLK		308
0008 706Ah	ICU	Interrupt request register 106	IR106	8	8	2	ICLK		308
0008 706Bh	ICU	Interrupt request register 107	IR107	8	8	2	ICLK		308
0008 706Ch	ICU	Interrupt request register 108	IR108	8	8	2	ICLK		308
0008 706Dh	ICU	Interrupt request register 109	IR109	8	8	2	ICLK		308
0008 706Eh	ICU	Interrupt request register 110	IR110	8	8	2	ICLK		308
0008 706Fh	ICU	Interrupt request register 111	IR111	8	8	2	ICLK		308
0008 7070h	ICU	Interrupt request register 112	IR112	8	8	2	ICLK		308
0008 7072h	ICU	Interrupt request register 114	IR114	8	8	2	ICLK		308
0008 707Ah	ICU	Interrupt request register 122	IR122	8	8	2	ICLK		308
0008 707Bh	ICU	Interrupt request register 123	IR123	8	8	2	ICLK		308
0008 707Ch	ICU	Interrupt request register 124	IR124	8	8	2	ICLK		308
0008 707Dh	ICU	Interrupt request register 125	IR125	8	8	2	ICLK		308
0008 707Eh	ICU	Interrupt request register 126	IR126	8	8	2	ICLK		308
0008 707Fh	ICU	Interrupt request register 127	IR127	8	8	2	ICLK		308
0008 7080h	ICU	Interrupt request register 128	IR128	8	8	2	ICLK		308
0008 7081h	ICU	Interrupt request register 129	IR129	8	8	2	ICLK		308
0008 7082h	ICU	Interrupt request register 130	IR130	8	8	2	ICLK		308
0008 7083h	ICU	Interrupt request register 131	IR131	8	8	2	ICLK		308
0008 7084h	ICU	Interrupt request register 132	IR132	8	8	2	ICLK		308
0008 7085h	ICU	Interrupt request register 133	IR133	8	8	2	ICLK		308
0008 7086h	ICU	Interrupt request register 134	IR134	8	8	2	ICLK		308
0008 7087h	ICU	Interrupt request register 135	IR135	8	8	2	ICLK		308
0008 7088h	ICU	Interrupt request register 136	IR136	8	8	2	ICLK		308
0008 7089h	ICU	Interrupt request register 137	IR137	8	8	2	ICLK		308
0008 708Ah	ICU	Interrupt request register 138	IR138	8	8	2	ICLK		308
0008 708Bh	ICU	Interrupt request register 139	IR139	8	8	2	ICLK		308
0008 708Ch	ICU	Interrupt request register 140	IR140	8	8	2	ICLK		308
0008 708Dh	ICU	Interrupt request register 141	IR141	8	8	2	ICLK		308
0008 708Eh	ICU	Interrupt request register 142	IR142	8	8	2	ICLK		308
0008 708Fh	ICU	Interrupt request register 143	IR143	8	8	2	ICLK	308	
0008 7090h	ICU	Interrupt request register 144	IR144	8	8	2	ICLK	308	
0008 7091h	ICU	Interrupt request register 145	IR145	8	8	2	ICLK	308	
0008 7092h	ICU	Interrupt request register 146	IR146	8	8	2	ICLK	308	

Table 5.1 List of I/O Registers (Address Order) (6/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 7093h	ICU	Interrupt request register 147	IR147	8	8	2	ICLK	ICUb	308
0008 7094h	ICU	Interrupt request register 148	IR148	8	8	2	ICLK		308
0008 7095h	ICU	Interrupt request register 149	IR149	8	8	2	ICLK		308
0008 7096h	ICU	Interrupt request register 150	IR150	8	8	2	ICLK		308
0008 7097h	ICU	Interrupt request register 151	IR151	8	8	2	ICLK		308
0008 7098h	ICU	Interrupt request register 152	IR152	8	8	2	ICLK		308
0008 7099h	ICU	Interrupt request register 153	IR153	8	8	2	ICLK		308
0008 709Ah	ICU	Interrupt request register 154	IR154	8	8	2	ICLK		308
0008 709Bh	ICU	Interrupt request register 155	IR155	8	8	2	ICLK		308
0008 709Ch	ICU	Interrupt request register 156	IR156	8	8	2	ICLK		308
0008 709Dh	ICU	Interrupt request register 157	IR157	8	8	2	ICLK		308
0008 709Eh	ICU	Interrupt request register 158	IR158	8	8	2	ICLK		308
0008 709Fh	ICU	Interrupt request register 159	IR159	8	8	2	ICLK		308
0008 70A0h	ICU	Interrupt request register 160	IR160	8	8	2	ICLK		308
0008 70A1h	ICU	Interrupt request register 161	IR161	8	8	2	ICLK		308
0008 70A2h	ICU	Interrupt request register 162	IR162	8	8	2	ICLK		308
0008 70A3h	ICU	Interrupt request register 163	IR163	8	8	2	ICLK		308
0008 70A4h	ICU	Interrupt request register 164	IR164	8	8	2	ICLK		308
0008 70A5h	ICU	Interrupt request register 165	IR165	8	8	2	ICLK		308
0008 70A6h	ICU	Interrupt request register 166	IR166	8	8	2	ICLK		308
0008 70A7h	ICU	Interrupt request register 167	IR167	8	8	2	ICLK		308
0008 70AAh	ICU	Interrupt request register 170	IR170	8	8	2	ICLK		308
0008 70ABh	ICU	Interrupt request register 171	IR171	8	8	2	ICLK		308
0008 70ACh	ICU	Interrupt request register 172	IR172	8	8	2	ICLK		308
0008 70ADh	ICU	Interrupt request register 173	IR173	8	8	2	ICLK		308
0008 70AEh	ICU	Interrupt request register 174	IR174	8	8	2	ICLK		308
0008 70AFh	ICU	Interrupt request register 175	IR175	8	8	2	ICLK		308
0008 70B0h	ICU	Interrupt request register 176	IR176	8	8	2	ICLK		308
0008 70B1h	ICU	Interrupt request register 177	IR177	8	8	2	ICLK		308
0008 70B2h	ICU	Interrupt request register 178	IR178	8	8	2	ICLK		308
0008 70B3h	ICU	Interrupt request register 179	IR179	8	8	2	ICLK		308
0008 70B4h	ICU	Interrupt request register 180	IR180	8	8	2	ICLK		308
0008 70B5h	ICU	Interrupt request register 181	IR181	8	8	2	ICLK		308
0008 70B6h	ICU	Interrupt request register 182	IR182	8	8	2	ICLK		308
0008 70B7h	ICU	Interrupt request register 183	IR183	8	8	2	ICLK		308
0008 70B8h	ICU	Interrupt request register 184	IR184	8	8	2	ICLK		308
0008 70B9h	ICU	Interrupt request register 185	IR185	8	8	2	ICLK		308
0008 70BAh	ICU	Interrupt request register 186	IR186	8	8	2	ICLK		308
0008 70BBh	ICU	Interrupt request register 187	IR187	8	8	2	ICLK	308	
0008 70BCh	ICU	Interrupt request register 188	IR188	8	8	2	ICLK	308	
0008 70BDh	ICU	Interrupt request register 189	IR189	8	8	2	ICLK	308	
0008 70BEh	ICU	Interrupt request register 190	IR190	8	8	2	ICLK	308	
0008 70BFh	ICU	Interrupt request register 191	IR191	8	8	2	ICLK	308	
0008 70C0h	ICU	Interrupt request register 192	IR192	8	8	2	ICLK	308	
0008 70C1h	ICU	Interrupt request register 193	IR193	8	8	2	ICLK	308	
0008 70C2h	ICU	Interrupt request register 194	IR194	8	8	2	ICLK	308	
0008 70C3h	ICU	Interrupt request register 195	IR195	8	8	2	ICLK	308	
0008 70C4h	ICU	Interrupt request register 196	IR196	8	8	2	ICLK	308	
0008 70C5h	ICU	Interrupt request register 197	IR197	8	8	2	ICLK	308	
0008 70C6h	ICU	Interrupt request register 198	IR198	8	8	2	ICLK	308	

Table 5.1 List of I/O Registers (Address Order) (7/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 70C7h	ICU	Interrupt request register 199	IR199	8	8	2	ICLK	ICUb	308
0008 70C8h	ICU	Interrupt request register 200	IR200	8	8	2	ICLK		308
0008 70C9h	ICU	Interrupt request register 201	IR201	8	8	2	ICLK		308
0008 70D6h	ICU	Interrupt request register 214	IR214	8	8	2	ICLK		308
0008 70D7h	ICU	Interrupt request register 215	IR215	8	8	2	ICLK		308
0008 70D8h	ICU	Interrupt request register 216	IR216	8	8	2	ICLK		308
0008 70D9h	ICU	Interrupt request register 217	IR217	8	8	2	ICLK		308
0008 70DAh	ICU	Interrupt request register 218	IR218	8	8	2	ICLK		308
0008 70DBh	ICU	Interrupt request register 219	IR219	8	8	2	ICLK		308
0008 70DCh	ICU	Interrupt request register 220	IR220	8	8	2	ICLK		308
0008 70DDh	ICU	Interrupt request register 221	IR221	8	8	2	ICLK		308
0008 70DEh	ICU	Interrupt request register 222	IR222	8	8	2	ICLK		308
0008 70DFh	ICU	Interrupt request register 223	IR223	8	8	2	ICLK		308
0008 70E0h	ICU	Interrupt request register 224	IR224	8	8	2	ICLK		308
0008 70E1h	ICU	Interrupt request register 225	IR225	8	8	2	ICLK		308
0008 70E2h	ICU	Interrupt request register 226	IR226	8	8	2	ICLK		308
0008 70E3h	ICU	Interrupt request register 227	IR227	8	8	2	ICLK		308
0008 70E4h	ICU	Interrupt request register 228	IR228	8	8	2	ICLK		308
0008 70E5h	ICU	Interrupt request register 229	IR229	8	8	2	ICLK		308
0008 70E6h	ICU	Interrupt request register 230	IR230	8	8	2	ICLK		308
0008 70E7h	ICU	Interrupt request register 231	IR231	8	8	2	ICLK		308
0008 70E8h	ICU	Interrupt request register 232	IR232	8	8	2	ICLK		308
0008 70E9h	ICU	Interrupt request register 233	IR233	8	8	2	ICLK		308
0008 70EAh	ICU	Interrupt request register 234	IR234	8	8	2	ICLK		308
0008 70EBh	ICU	Interrupt request register 235	IR235	8	8	2	ICLK		308
0008 70ECh	ICU	Interrupt request register 236	IR236	8	8	2	ICLK		308
0008 70EDh	ICU	Interrupt request register 237	IR237	8	8	2	ICLK		308
0008 70EEh	ICU	Interrupt request register 238	IR238	8	8	2	ICLK		308
0008 70EFh	ICU	Interrupt request register 239	IR239	8	8	2	ICLK		308
0008 70F0h	ICU	Interrupt request register 240	IR240	8	8	2	ICLK		308
0008 70F1h	ICU	Interrupt request register 241	IR241	8	8	2	ICLK		308
0008 70F2h	ICU	Interrupt request register 242	IR242	8	8	2	ICLK		308
0008 70F3h	ICU	Interrupt request register 243	IR243	8	8	2	ICLK		308
0008 70F4h	ICU	Interrupt request register 244	IR244	8	8	2	ICLK	308	
0008 70F5h	ICU	Interrupt request register 245	IR245	8	8	2	ICLK	308	
0008 70F6h	ICU	Interrupt request register 246	IR246	8	8	2	ICLK	308	
0008 70F7h	ICU	Interrupt request register 247	IR247	8	8	2	ICLK	308	
0008 70F8h	ICU	Interrupt request register 248	IR248	8	8	2	ICLK	308	
0008 70F9h	ICU	Interrupt request register 249	IR249	8	8	2	ICLK	308	
0008 70FAh	ICU	Interrupt request register 250	IR250	8	8	2	ICLK	308	
0008 70FBh	ICU	Interrupt request register 251	IR251	8	8	2	ICLK	308	
0008 70FCh	ICU	Interrupt request register 252	IR252	8	8	2	ICLK	308	
0008 70FDh	ICU	Interrupt request register 253	IR253	8	8	2	ICLK	308	
0008 711Bh	ICU	DTC activation enable register 027	DTCER027	8	8	2	ICLK	311	
0008 711Ch	ICU	DTC activation enable register 028	DTCER028	8	8	2	ICLK	311	
0008 711Dh	ICU	DTC activation enable register 029	DTCER029	8	8	2	ICLK	311	
0008 711Eh	ICU	DTC activation enable register 030	DTCER030	8	8	2	ICLK	311	
0008 711Fh	ICU	DTC activation enable register 031	DTCER031	8	8	2	ICLK	311	
0008 7121h	ICU	DTC activation enable register 033	DTCER033	8	8	2	ICLK	311	
0008 7122h	ICU	DTC activation enable register 034	DTCER034	8	8	2	ICLK	311	

Table 5.1 List of I/O Registers (Address Order) (8/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 7127h	ICU	DTC activation enable register 039	DTCER039	8	8	2	ICLK	ICUb	311
0008 7128h	ICU	DTC activation enable register 040	DTCER040	8	8	2	ICLK		311
0008 712Ah	ICU	DTC activation enable register 042	DTCER042	8	8	2	ICLK		311
0008 712Bh	ICU	DTC activation enable register 043	DTCER043	8	8	2	ICLK		311
0008 712Dh	ICU	DTC activation enable register 045	DTCER045	8	8	2	ICLK		311
0008 712Eh	ICU	DTC activation enable register 046	DTCER046	8	8	2	ICLK		311
0008 7140h	ICU	DTC activation enable register 064	DTCER064	8	8	2	ICLK		311
0008 7141h	ICU	DTC activation enable register 065	DTCER065	8	8	2	ICLK		311
0008 7142h	ICU	DTC activation enable register 066	DTCER066	8	8	2	ICLK		311
0008 7143h	ICU	DTC activation enable register 067	DTCER067	8	8	2	ICLK		311
0008 7144h	ICU	DTC activation enable register 068	DTCER068	8	8	2	ICLK		311
0008 7145h	ICU	DTC activation enable register 069	DTCER069	8	8	2	ICLK		311
0008 7146h	ICU	DTC activation enable register 070	DTCER070	8	8	2	ICLK		311
0008 7147h	ICU	DTC activation enable register 071	DTCER071	8	8	2	ICLK		311
0008 7148h	ICU	DTC activation enable register 072	DTCER072	8	8	2	ICLK		311
0008 7149h	ICU	DTC activation enable register 073	DTCER073	8	8	2	ICLK		311
0008 714Ah	ICU	DTC activation enable register 074	DTCER074	8	8	2	ICLK		311
0008 714Bh	ICU	DTC activation enable register 075	DTCER075	8	8	2	ICLK		311
0008 714Ch	ICU	DTC activation enable register 076	DTCER076	8	8	2	ICLK		311
0008 714Dh	ICU	DTC activation enable register 077	DTCER077	8	8	2	ICLK		311
0008 714Eh	ICU	DTC activation enable register 078	DTCER078	8	8	2	ICLK		311
0008 714Fh	ICU	DTC activation enable register 079	DTCER079	8	8	2	ICLK		311
0008 7162h	ICU	DTC activation enable register 098	DTCER098	8	8	2	ICLK		311
0008 7166h	ICU	DTC activation enable register 102	DTCER102	8	8	2	ICLK		311
0008 717Eh	ICU	DTC activation enable register 126	DTCER126	8	8	2	ICLK		311
0008 717Fh	ICU	DTC activation enable register 127	DTCER127	8	8	2	ICLK		311
0008 7180h	ICU	DTC activation enable register 128	DTCER128	8	8	2	ICLK		311
0008 7181h	ICU	DTC activation enable register 129	DTCER129	8	8	2	ICLK		311
0008 7182h	ICU	DTC activation enable register 130	DTCER130	8	8	2	ICLK		311
0008 7183h	ICU	DTC activation enable register 131	DTCER131	8	8	2	ICLK		311
0008 7184h	ICU	DTC activation enable register 132	DTCER132	8	8	2	ICLK		311
0008 7185h	ICU	DTC activation enable register 133	DTCER133	8	8	2	ICLK		311
0008 7186h	ICU	DTC activation enable register 134	DTCER134	8	8	2	ICLK		311
0008 7187h	ICU	DTC activation enable register 135	DTCER135	8	8	2	ICLK		311
0008 7188h	ICU	DTC activation enable register 136	DTCER136	8	8	2	ICLK		311
0008 7189h	ICU	DTC activation enable register 137	DTCER137	8	8	2	ICLK	311	
0008 718Ah	ICU	DTC activation enable register 138	DTCER138	8	8	2	ICLK	311	
0008 718Bh	ICU	DTC activation enable register 139	DTCER139	8	8	2	ICLK	311	
0008 718Ch	ICU	DTC activation enable register 140	DTCER140	8	8	2	ICLK	311	
0008 718Dh	ICU	DTC activation enable register 141	DTCER141	8	8	2	ICLK	311	
0008 718Eh	ICU	DTC activation enable register 142	DTCER142	8	8	2	ICLK	311	
0008 718Fh	ICU	DTC activation enable register 143	DTCER143	8	8	2	ICLK	311	
0008 7190h	ICU	DTC activation enable register 144	DTCER144	8	8	2	ICLK	311	
0008 7191h	ICU	DTC activation enable register 145	DTCER145	8	8	2	ICLK	311	
0008 7194h	ICU	DTC activation enable register 148	DTCER148	8	8	2	ICLK	311	
0008 7195h	ICU	DTC activation enable register 149	DTCER149	8	8	2	ICLK	311	
0008 7196h	ICU	DTC activation enable register 150	DTCER150	8	8	2	ICLK	311	
0008 7197h	ICU	DTC activation enable register 151	DTCER151	8	8	2	ICLK	311	
0008 7198h	ICU	DTC activation enable register 152	DTCER152	8	8	2	ICLK	311	
0008 7199h	ICU	DTC activation enable register 153	DTCER153	8	8	2	ICLK	311	

Table 5.1 List of I/O Registers (Address Order) (9/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 719Ah	ICU	DTC activation enable register 154	DTCER154	8	8	2	ICLK	ICUb	311
0008 719Bh	ICU	DTC activation enable register 155	DTCER155	8	8	2	ICLK		311
0008 719Ch	ICU	DTC activation enable register 156	DTCER156	8	8	2	ICLK		311
0008 719Dh	ICU	DTC activation enable register 157	DTCER157	8	8	2	ICLK		311
0008 719Eh	ICU	DTC activation enable register 158	DTCER158	8	8	2	ICLK		311
0008 719Fh	ICU	DTC activation enable register 159	DTCER159	8	8	2	ICLK		311
0008 71A0h	ICU	DTC activation enable register 160	DTCER160	8	8	2	ICLK		311
0008 71A1h	ICU	DTC activation enable register 161	DTCER161	8	8	2	ICLK		311
0008 71A2h	ICU	DTC activation enable register 162	DTCER162	8	8	2	ICLK		311
0008 71A3h	ICU	DTC activation enable register 163	DTCER163	8	8	2	ICLK		311
0008 71A4h	ICU	DTC activation enable register 164	DTCER164	8	8	2	ICLK		311
0008 71A5h	ICU	DTC activation enable register 165	DTCER165	8	8	2	ICLK		311
0008 71AAh	ICU	DTC activation enable register 170	DTCER170	8	8	2	ICLK		311
0008 71ABh	ICU	DTC activation enable register 171	DTCER171	8	8	2	ICLK		311
0008 71ADh	ICU	DTC activation enable register 173	DTCER173	8	8	2	ICLK		311
0008 71AEh	ICU	DTC activation enable register 174	DTCER174	8	8	2	ICLK		311
0008 71B0h	ICU	DTC activation enable register 176	DTCER176	8	8	2	ICLK		311
0008 71B1h	ICU	DTC activation enable register 177	DTCER177	8	8	2	ICLK		311
0008 71B3h	ICU	DTC activation enable register 179	DTCER179	8	8	2	ICLK		311
0008 71B4h	ICU	DTC activation enable register 180	DTCER180	8	8	2	ICLK		311
0008 71B7h	ICU	DTC activation enable register 183	DTCER183	8	8	2	ICLK		311
0008 71B8h	ICU	DTC activation enable register 184	DTCER184	8	8	2	ICLK		311
0008 71BBh	ICU	DTC activation enable register 187	DTCER187	8	8	2	ICLK		311
0008 71BCh	ICU	DTC activation enable register 188	DTCER188	8	8	2	ICLK		311
0008 71BFh	ICU	DTC activation enable register 191	DTCER191	8	8	2	ICLK		311
0008 71C0h	ICU	DTC activation enable register 192	DTCER192	8	8	2	ICLK		311
0008 71C3h	ICU	DTC activation enable register 195	DTCER195	8	8	2	ICLK		311
0008 71C4h	ICU	DTC activation enable register 196	DTCER196	8	8	2	ICLK		311
0008 71C6h	ICU	DTC activation enable register 198	DTCER198	8	8	2	ICLK		311
0008 71C7h	ICU	DTC activation enable register 199	DTCER199	8	8	2	ICLK		311
0008 71C8h	ICU	DTC activation enable register 200	DTCER200	8	8	2	ICLK		311
0008 71C9h	ICU	DTC activation enable register 201	DTCER201	8	8	2	ICLK		311
0008 71D6h	ICU	DTC activation enable register 214	DTCER214	8	8	2	ICLK		311
0008 71D7h	ICU	DTC activation enable register 215	DTCER215	8	8	2	ICLK		311
0008 71D9h	ICU	DTC activation enable register 217	DTCER217	8	8	2	ICLK		311
0008 71DAh	ICU	DTC activation enable register 218	DTCER218	8	8	2	ICLK		311
0008 71DCh	ICU	DTC activation enable register 220	DTCER220	8	8	2	ICLK		311
0008 71DDh	ICU	DTC activation enable register 221	DTCER221	8	8	2	ICLK		311
0008 71DFh	ICU	DTC activation enable register 223	DTCER223	8	8	2	ICLK		311
0008 71E0h	ICU	DTC activation enable register 224	DTCER224	8	8	2	ICLK		311
0008 71E2h	ICU	DTC activation enable register 226	DTCER226	8	8	2	ICLK		311
0008 71E3h	ICU	DTC activation enable register 227	DTCER227	8	8	2	ICLK		311
0008 71E5h	ICU	DTC activation enable register 229	DTCER229	8	8	2	ICLK	311	
0008 71E6h	ICU	DTC activation enable register 230	DTCER230	8	8	2	ICLK	311	
0008 71E8h	ICU	DTC activation enable register 232	DTCER232	8	8	2	ICLK	311	
0008 71E9h	ICU	DTC activation enable register 233	DTCER233	8	8	2	ICLK	311	
0008 71EBh	ICU	DTC activation enable register 235	DTCER235	8	8	2	ICLK	311	
0008 71ECh	ICU	DTC activation enable register 236	DTCER236	8	8	2	ICLK	311	
0008 71EEh	ICU	DTC activation enable register 238	DTCER238	8	8	2	ICLK	311	
0008 71EFh	ICU	DTC activation enable register 239	DTCER239	8	8	2	ICLK	311	



Table 5.1 List of I/O Registers (Address Order) (10/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 71F1h	ICU	DTC activation enable register 241	DTCER241	8	8	2	ICLK	ICUb	311
0008 71F2h	ICU	DTC activation enable register 242	DTCER242	8	8	2	ICLK		311
0008 71F4h	ICU	DTC activation enable register 244	DTCER244	8	8	2	ICLK		311
0008 71F5h	ICU	DTC activation enable register 245	DTCER245	8	8	2	ICLK		311
0008 71F7h	ICU	DTC activation enable register 247	DTCER247	8	8	2	ICLK		311
0008 71F8h	ICU	DTC activation enable register 248	DTCER248	8	8	2	ICLK		311
0008 71FAh	ICU	DTC activation enable register 250	DTCER250	8	8	2	ICLK		311
0008 71FBh	ICU	DTC activation enable register 251	DTCER251	8	8	2	ICLK		311
0008 7202h	ICU	Interrupt request enable register 02	IER02	8	8	2	ICLK		308
0008 7203h	ICU	Interrupt request enable register 03	IER03	8	8	2	ICLK		308
0008 7204h	ICU	Interrupt request enable register 04	IER04	8	8	2	ICLK	308	
0008 7205h	ICU	Interrupt request enable register 05	IER05	8	8	2	ICLK	308	
0008 7206h	ICU	Interrupt request enable register 06	IER06	8	8	2	ICLK	308	
0008 7207h	ICU	Interrupt request enable register 07	IER07	8	8	2	ICLK	308	
0008 7208h	ICU	Interrupt request enable register 08	IER08	8	8	2	ICLK	308	
0008 7209h	ICU	Interrupt request enable register 09	IER09	8	8	2	ICLK	308	
0008 720Bh	ICU	Interrupt request enable register 0B	IER0B	8	8	2	ICLK	308	
0008 720Ch	ICU	Interrupt request enable register 0C	IER0C	8	8	2	ICLK	308	
0008 720Dh	ICU	Interrupt request enable register 0D	IER0D	8	8	2	ICLK	308	
0008 720Eh	ICU	Interrupt request enable register 0E	IER0E	8	8	2	ICLK	308	
0008 720Fh	ICU	Interrupt request enable register 0F	IER0F	8	8	2	ICLK	308	
0008 7210h	ICU	Interrupt request enable register 10	IER10	8	8	2	ICLK	308	
0008 7211h	ICU	Interrupt request enable register 11	IER11	8	8	2	ICLK	308	
0008 7212h	ICU	Interrupt request enable register 12	IER12	8	8	2	ICLK	308	
0008 7213h	ICU	Interrupt request enable register 13	IER13	8	8	2	ICLK	308	
0008 7214h	ICU	Interrupt request enable register 14	IER14	8	8	2	ICLK	308	
0008 7215h	ICU	Interrupt request enable register 15	IER15	8	8	2	ICLK	308	
0008 7216h	ICU	Interrupt request enable register 16	IER16	8	8	2	ICLK	308	
0008 7217h	ICU	Interrupt request enable register 17	IER17	8	8	2	ICLK	308	
0008 7218h	ICU	Interrupt request enable register 18	IER18	8	8	2	ICLK	308	
0008 7219h	ICU	Interrupt request enable register 19	IER19	8	8	2	ICLK	308	
0008 721Ah	ICU	Interrupt request enable register 1A	IER1A	8	8	2	ICLK	308	
0008 721Bh	ICU	Interrupt request enable register 1B	IER1B	8	8	2	ICLK	308	
0008 721Ch	ICU	Interrupt request enable register 1C	IER1C	8	8	2	ICLK	308	
0008 721Dh	ICU	Interrupt request enable register 1D	IER1D	8	8	2	ICLK	308	
0008 721Eh	ICU	Interrupt request enable register 1E	IER1E	8	8	2	ICLK	308	
0008 721Fh	ICU	Interrupt request enable register 1F	IER1F	8	8	2	ICLK	308	
0008 72E0h	ICU	Software interrupt activation register	SWINTR	8	8	2	ICLK	311	
0008 72F0h	ICU	Fast interrupt set register	FIR	16	16	2	ICLK	310	
0008 7300h	ICU	Interrupt source priority register 000	IPR000	8	8	2	ICLK	309	
0008 7301h	ICU	Interrupt source priority register 001	IPR001	8	8	2	ICLK	309	
0008 7302h	ICU	Interrupt source priority register 002	IPR002	8	8	2	ICLK	309	
0008 7303h	ICU	Interrupt source priority register 003	IPR003	8	8	2	ICLK	309	
0008 7304h	ICU	Interrupt source priority register 004	IPR004	8	8	2	ICLK	309	
0008 7305h	ICU	Interrupt source priority register 005	IPR005	8	8	2	ICLK	309	
0008 7306h	ICU	Interrupt source priority register 006	IPR006	8	8	2	ICLK	309	
0008 7307h	ICU	Interrupt source priority register 007	IPR007	8	8	2	ICLK	309	
0008 7321h	ICU	Interrupt source priority register 033	IPR033	8	8	2	ICLK	309	
0008 7322h	ICU	Interrupt source priority register 034	IPR034	8	8	2	ICLK	309	
0008 7323h	ICU	Interrupt source priority register 035	IPR035	8	8	2	ICLK	309	

Table 5.1 List of I/O Registers (Address Order) (11/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 7327h	ICU	Interrupt source priority register 039	IPR039	8	8	2	ICLK	ICUb	309
0008 732Ah	ICU	Interrupt source priority register 042	IPR042	8	8	2	ICLK		309
0008 732Dh	ICU	Interrupt source priority register 045	IPR045	8	8	2	ICLK		309
0008 7330h	ICU	Interrupt source priority register 048	IPR048	8	8	2	ICLK		309
0008 7334h	ICU	Interrupt source priority register 052	IPR052	8	8	2	ICLK		309
0008 7338h	ICU	Interrupt source priority register 056	IPR056	8	8	2	ICLK		309
0008 733Eh	ICU	Interrupt source priority register 062	IPR062	8	8	2	ICLK		309
0008 7340h	ICU	Interrupt source priority register 064	IPR064	8	8	2	ICLK		309
0008 7341h	ICU	Interrupt source priority register 065	IPR065	8	8	2	ICLK		309
0008 7342h	ICU	Interrupt source priority register 066	IPR066	8	8	2	ICLK		309
0008 7343h	ICU	Interrupt source priority register 067	IPR067	8	8	2	ICLK		309
0008 7344h	ICU	Interrupt source priority register 068	IPR068	8	8	2	ICLK		309
0008 7345h	ICU	Interrupt source priority register 069	IPR069	8	8	2	ICLK		309
0008 7346h	ICU	Interrupt source priority register 070	IPR070	8	8	2	ICLK		309
0008 7347h	ICU	Interrupt source priority register 071	IPR071	8	8	2	ICLK		309
0008 7348h	ICU	Interrupt source priority register 072	IPR072	8	8	2	ICLK		309
0008 7349h	ICU	Interrupt source priority register 073	IPR073	8	8	2	ICLK		309
0008 734Ah	ICU	Interrupt source priority register 074	IPR074	8	8	2	ICLK		309
0008 734Bh	ICU	Interrupt source priority register 075	IPR075	8	8	2	ICLK		309
0008 734Ch	ICU	Interrupt source priority register 076	IPR076	8	8	2	ICLK		309
0008 734Dh	ICU	Interrupt source priority register 077	IPR077	8	8	2	ICLK		309
0008 734Eh	ICU	Interrupt source priority register 078	IPR078	8	8	2	ICLK		309
0008 734Fh	ICU	Interrupt source priority register 079	IPR079	8	8	2	ICLK		309
0008 735Ah	ICU	Interrupt source priority register 090	IPR090	8	8	2	ICLK		309
0008 735Ch	ICU	Interrupt source priority register 092	IPR092	8	8	2	ICLK		309
0008 735Dh	ICU	Interrupt source priority register 093	IPR093	8	8	2	ICLK		309
0008 7362h	ICU	Interrupt source priority register 098	IPR098	8	8	2	ICLK		309
0008 7366h	ICU	Interrupt source priority register 102	IPR102	8	8	2	ICLK		309
0008 736Ah	ICU	Interrupt source priority register 106	IPR106	8	8	2	ICLK		309
0008 736Bh	ICU	Interrupt source priority register 107	IPR107	8	8	2	ICLK		309
0008 736Ch	ICU	Interrupt source priority register 108	IPR108	8	8	2	ICLK		309
0008 736Dh	ICU	Interrupt source priority register 109	IPR109	8	8	2	ICLK		309
0008 736Eh	ICU	Interrupt source priority register 110	IPR110	8	8	2	ICLK		309
0008 736Fh	ICU	Interrupt source priority register 111	IPR111	8	8	2	ICLK		309
0008 7370h	ICU	Interrupt source priority register 112	IPR112	8	8	2	ICLK		309
0008 7372h	ICU	Interrupt source priority register 114	IPR114	8	8	2	ICLK		309
0008 737Ah	ICU	Interrupt source priority register 122	IPR122	8	8	2	ICLK		309
0008 737Eh	ICU	Interrupt source priority register 126	IPR126	8	8	2	ICLK		309
0008 7382h	ICU	Interrupt source priority register 130	IPR130	8	8	2	ICLK		309
0008 7384h	ICU	Interrupt source priority register 132	IPR132	8	8	2	ICLK		309
0008 7386h	ICU	Interrupt source priority register 134	IPR134	8	8	2	ICLK		309
0008 738Ah	ICU	Interrupt source priority register 138	IPR138	8	8	2	ICLK		309
0008 738Ch	ICU	Interrupt source priority register 140	IPR140	8	8	2	ICLK	309	
0008 738Eh	ICU	Interrupt source priority register 142	IPR142	8	8	2	ICLK	309	
0008 7392h	ICU	Interrupt source priority register 146	IPR146	8	8	2	ICLK	309	
0008 7394h	ICU	Interrupt source priority register 148	IPR148	8	8	2	ICLK	309	
0008 7396h	ICU	Interrupt source priority register 150	IPR150	8	8	2	ICLK	309	
0008 7398h	ICU	Interrupt source priority register 152	IPR152	8	8	2	ICLK	309	
0008 739Ch	ICU	Interrupt source priority register 156	IPR156	8	8	2	ICLK	309	
0008 73A0h	ICU	Interrupt source priority register 160	IPR160	8	8	2	ICLK	309	

Table 5.1 List of I/O Registers (Address Order) (12/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 73A1h	ICU	Interrupt source priority register 161	IPR161	8	8	2	ICLK	ICUb	309
0008 73A4h	ICU	Interrupt source priority register 164	IPR164	8	8	2	ICLK		309
0008 73A6h	ICU	Interrupt source priority register 166	IPR166	8	8	2	ICLK		309
0008 73AAh	ICU	Interrupt source priority register 170	IPR170	8	8	2	ICLK		309
0008 73ADh	ICU	Interrupt source priority register 173	IPR173	8	8	2	ICLK		309
0008 73B0h	ICU	Interrupt source priority register 176	IPR176	8	8	2	ICLK		309
0008 73B3h	ICU	Interrupt source priority register 179	IPR179	8	8	2	ICLK		309
0008 73B6h	ICU	Interrupt source priority register 182	IPR182	8	8	2	ICLK		309
0008 73B7h	ICU	Interrupt source priority register 183	IPR183	8	8	2	ICLK		309
0008 73B8h	ICU	Interrupt source priority register 184	IPR184	8	8	2	ICLK		309
0008 73B9h	ICU	Interrupt source priority register 185	IPR185	8	8	2	ICLK		309
0008 73BAh	ICU	Interrupt source priority register 186	IPR186	8	8	2	ICLK		309
0008 73BBh	ICU	Interrupt source priority register 187	IPR187	8	8	2	ICLK		309
0008 73BCh	ICU	Interrupt source priority register 188	IPR188	8	8	2	ICLK		309
0008 73BDh	ICU	Interrupt source priority register 189	IPR189	8	8	2	ICLK		309
0008 73BEh	ICU	Interrupt source priority register 190	IPR190	8	8	2	ICLK		309
0008 73BFh	ICU	Interrupt source priority register 191	IPR191	8	8	2	ICLK		309
0008 73C0h	ICU	Interrupt source priority register 192	IPR192	8	8	2	ICLK		309
0008 73C1h	ICU	Interrupt source priority register 193	IPR193	8	8	2	ICLK		309
0008 73C2h	ICU	Interrupt source priority register 194	IPR194	8	8	2	ICLK		309
0008 73C3h	ICU	Interrupt source priority register 195	IPR195	8	8	2	ICLK		309
0008 73C4h	ICU	Interrupt source priority register 196	IPR196	8	8	2	ICLK		309
0008 73C5h	ICU	Interrupt source priority register 197	IPR197	8	8	2	ICLK		309
0008 73C6h	ICU	Interrupt source priority register 198	IPR198	8	8	2	ICLK		309
0008 73C7h	ICU	Interrupt source priority register 199	IPR199	8	8	2	ICLK		309
0008 73C8h	ICU	Interrupt source priority register 200	IPR200	8	8	2	ICLK		309
0008 73C9h	ICU	Interrupt source priority register 201	IPR201	8	8	2	ICLK		309
0008 73D6h	ICU	Interrupt source priority register 214	IPR214	8	8	2	ICLK		309
0008 73D9h	ICU	Interrupt source priority register 217	IPR217	8	8	2	ICLK		309
0008 73DCh	ICU	Interrupt source priority register 220	IPR220	8	8	2	ICLK		309
0008 73DFh	ICU	Interrupt source priority register 223	IPR223	8	8	2	ICLK		309
0008 73E2h	ICU	Interrupt source priority register 226	IPR226	8	8	2	ICLK		309
0008 73E5h	ICU	Interrupt source priority register 229	IPR229	8	8	2	ICLK		309
0008 73E8h	ICU	Interrupt source priority register 232	IPR232	8	8	2	ICLK	309	
0008 73EBh	ICU	Interrupt source priority register 235	IPR235	8	8	2	ICLK	309	
0008 73EEh	ICU	Interrupt source priority register 238	IPR238	8	8	2	ICLK	309	
0008 73F1h	ICU	Interrupt source priority register 241	IPR241	8	8	2	ICLK	309	
0008 73F4h	ICU	Interrupt source priority register 244	IPR244	8	8	2	ICLK	309	
0008 73F7h	ICU	Interrupt source priority register 247	IPR247	8	8	2	ICLK	309	
0008 73FAh	ICU	Interrupt source priority register 250	IPR250	8	8	2	ICLK	309	
0008 73FDh	ICU	Interrupt source priority register 253	IPR253	8	8	2	ICLK	309	
0008 7400h	ICU	DMAC activation request select register 0	DMRSR0	8	8	2	ICLK	312	
0008 7404h	ICU	DMAC activation request select register 1	DMRSR1	8	8	2	ICLK	312	
0008 7408h	ICU	DMAC activation request select register 2	DMRSR2	8	8	2	ICLK	312	
0008 740Ch	ICU	DMAC activation request select register 3	DMRSR3	8	8	2	ICLK	312	
0008 7500h	ICU	IRQ control register 0	IRQCR0	8	8	2	ICLK	313	
0008 7501h	ICU	IRQ control register 1	IRQCR1	8	8	2	ICLK	313	
0008 7502h	ICU	IRQ control register 2	IRQCR2	8	8	2	ICLK	313	
0008 7503h	ICU	IRQ control register 3	IRQCR3	8	8	2	ICLK	313	
0008 7504h	ICU	IRQ control register 4	IRQCR4	8	8	2	ICLK	313	

Table 5.1 List of I/O Registers (Address Order) (13/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page	
						ICLK ≥ PCLK	ICLK < PCLK			
0008 7505h	ICU	IRQ control register 5	IRQCR5	8	8	2 ICLK		ICUb	313	
0008 7506h	ICU	IRQ control register 6	IRQCR6	8	8	2 ICLK			313	
0008 7507h	ICU	IRQ control register 7	IRQCR7	8	8	2 ICLK			313	
0008 7508h	ICU	IRQ control register 8	IRQCR8	8	8	2 ICLK			313	
0008 7509h	ICU	IRQ control register 9	IRQCR9	8	8	2 ICLK			313	
0008 750Ah	ICU	IRQ control register 10	IRQCR10	8	8	2 ICLK			313	
0008 750Bh	ICU	IRQ control register 11	IRQCR11	8	8	2 ICLK			313	
0008 750Ch	ICU	IRQ control register 12	IRQCR12	8	8	2 ICLK			313	
0008 750Dh	ICU	IRQ control register 13	IRQCR13	8	8	2 ICLK			313	
0008 750Eh	ICU	IRQ control register 14	IRQCR14	8	8	2 ICLK			313	
0008 750Fh	ICU	IRQ control register 15	IRQCR15	8	8	2 ICLK			313	
0008 7510h	ICU	IRQ pin digital filter enable register 0	IRQFLTE0	8	8	2 ICLK			314	
0008 7511h	ICU	IRQ pin digital filter enable register 1	IRQFLTE1	8	8	2 ICLK			315	
0008 7514h	ICU	IRQ pin digital filter setting register 0	IRQFLTC0	8	8	2 ICLK			316	
0008 7516h	ICU	IRQ pin digital filter setting register 1	IRQFLTC1	8	8	2 ICLK			317	
0008 7580h	ICU	Non-maskable interrupt status register	NMISR	8	8	2 ICLK		318		
0008 7581h	ICU	Non-maskable interrupt enable register	NMIER	8	8	2 ICLK		320		
0008 7582h	ICU	Non-maskable interrupt status clear register	NMICLR	8	8	2 ICLK		321		
0008 7583h	ICU	NMI pin interrupt control register	NMICR	8	8	2 ICLK		322		
0008 7590h	ICU	NMI pin digital filter enable register	NMIFLTE	8	8	2 ICLK		322		
0008 7594h	ICU	NMI pin digital filter setting register	NMIFLTC	8	8	2 ICLK		323		
0008 8000h	CMT	Compare match timer start register 0	CMSTR0	16	16	2, 3 PCLKB	2 ICLK	CMT	873	
0008 8002h	CMT0	Compare match timer control register	CMCR	16	16	2, 3 PCLKB	2 ICLK		874	
0008 8004h	CMT0	Compare match timer counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK		875	
0008 8006h	CMT0	Compare match timer constant register	CMCOR	16	16	2, 3 PCLKB	2 ICLK		875	
0008 8008h	CMT1	Compare match timer control register	CMCR	16	16	2, 3 PCLKB	2 ICLK		875	
0008 800Ah	CMT1	Compare match timer counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK		875	
0008 800Ch	CMT1	Compare match timer constant register	CMCOR	16	16	2, 3 PCLKB	2 ICLK		875	
0008 8010h	CMT	Compare match timer start register 1	CMSTR1	16	16	2, 3 PCLKB	2 ICLK		873	
0008 8012h	CMT2	Compare match timer control register	CMCR	16	16	2, 3 PCLKB	2 ICLK		874	
0008 8014h	CMT2	Compare match timer counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK		875	
0008 8016h	CMT2	Compare match timer constant register	CMCOR	16	16	2, 3 PCLKB	2 ICLK		875	
0008 8018h	CMT3	Compare match timer control register	CMCR	16	16	2, 3 PCLKB	2 ICLK		874	
0008 801Ah	CMT3	Compare match timer counter	CMCNT	16	16	2, 3 PCLKB	2 ICLK		875	
0008 801Ch	CMT3	Compare match timer constant register	CMCOR	16	16	2, 3 PCLKB	2 ICLK		875	
0008 8020h	WDT	WDT refresh register	WDTRR	8	8	2, 3 PCLKB	2 ICLK		WDTA	919
0008 8022h	WDT	WDT control register	WDTCR	16	16	2, 3 PCLKB	2 ICLK			920
0008 8024h	WDT	WDT status register	WDTSR	16	16	2, 3 PCLKB	2 ICLK			923
0008 8026h	WDT	WDT reset control register	WDTRCR	8	8	2, 3 PCLKB	2 ICLK			924
0008 8030h	IWDT	IWDT refresh register	IWDTRR	8	8	2, 3 PCLKB	2 ICLK	IWDTa	934	
0008 8032h	IWDT	IWDT control register	IWDTCR	16	16	2, 3 PCLKB	2 ICLK		935	
0008 8034h	IWDT	IWDT status register	IWDTSR	16	16	2, 3 PCLKB	2 ICLK		938	
0008 8036h	IWDT	IWDT reset control register	IWDTRCR	8	8	2, 3 PCLKB	2 ICLK		939	
0008 8038h	IWDT	IWDT count stop control register	IWDTCSTPR	8	8	2, 3 PCLKB	2 ICLK		939	
0008 80C0h	DA	D/A data register 0	DADR0	16	16	2, 3 PCLKB	2 ICLK	DAa	1451	
0008 80C2h	DA	D/A data register 1	DADR1	16	16	2, 3 PCLKB	2 ICLK		1451	
0008 80C4h	DA	D/A control register	DACR	8	8	2, 3 PCLKB	2 ICLK		1452	
0008 80C5h	DA	DADRm format select register	DADPR	8	8	2, 3 PCLKB	2 ICLK		1453	
0008 80C6h	DA	D/A A/D synchronous start control register	DAADSCR	8	8	2, 3 PCLKB	2 ICLK		1454	
0008 8100h	TPUA	Timer start register	TSTR	8	8	2, 3 PCLKB	2 ICLK	TPUa	774	

Table 5.1 List of I/O Registers (Address Order) (14/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 8101h	TPUA	Timer synchronous register	TSYR	8	8	2, 3 PCLKB	2 ICLK	TPUa	775
0008 8108h	TPU0	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK		776
0008 8109h	TPU1	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK		776
0008 810Ah	TPU2	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK		776
0008 810Bh	TPU3	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK		776
0008 810Ch	TPU4	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK		776
0008 810Dh	TPU5	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK		776
0008 8110h	TPU0	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK		753
0008 8111h	TPU0	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK		758
0008 8112h	TPU0	Timer I/O control register H	TIORH	8	8	2, 3 PCLKB	2 ICLK		759
0008 8113h	TPU0	Timer I/O control register L	TIORL	8	8	2, 3 PCLKB	2 ICLK	759	
0008 8114h	TPU0	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	769	
0008 8115h	TPU0	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	770	
0008 8116h	TPU0	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	772	
0008 8118h	TPU0	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	773	
0008 811Ah	TPU0	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	773	
0008 811Ch	TPU0	Timer general register C	TGRC	16	16	2, 3 PCLKB	2 ICLK	773	
0008 811Eh	TPU0	Timer general register D	TGRD	16	16	2, 3 PCLKB	2 ICLK	773	
0008 8120h	TPU1	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	753	
0008 8121h	TPU1	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	758	
0008 8122h	TPU1	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK	759	
0008 8124h	TPU1	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	769	
0008 8125h	TPU1	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	770	
0008 8126h	TPU1	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	772	
0008 8128h	TPU1	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	773	
0008 812Ah	TPU1	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	773	
0008 8130h	TPU2	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	753	
0008 8131h	TPU2	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	758	
0008 8132h	TPU2	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK	759	
0008 8134h	TPU2	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	769	
0008 8135h	TPU2	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	770	
0008 8136h	TPU2	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	772	
0008 8138h	TPU2	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	773	
0008 813Ah	TPU2	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	773	
0008 8140h	TPU3	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	753	
0008 8141h	TPU3	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	758	
0008 8142h	TPU3	Timer I/O control register H	TIORH	8	8	2, 3 PCLKB	2 ICLK	759	
0008 8143h	TPU3	Timer I/O control register L	TIORL	8	8	2, 3 PCLKB	2 ICLK	759	
0008 8144h	TPU3	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	769	
0008 8145h	TPU3	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	770	
0008 8146h	TPU3	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	772	
0008 8148h	TPU3	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	773	
0008 814Ah	TPU3	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	773	
0008 814Ch	TPU3	Timer general register C	TGRC	16	16	2, 3 PCLKB	2 ICLK	773	
0008 814Eh	TPU3	Timer general register D	TGRD	16	16	2, 3 PCLKB	2 ICLK	773	
0008 8150h	TPU4	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	753	
0008 8151h	TPU4	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	758	
0008 8152h	TPU4	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK	759	
0008 8154h	TPU4	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	769	
0008 8155h	TPU4	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	770	

Table 5.1 List of I/O Registers (Address Order) (15/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 8156h	TPU4	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	TPUa	772
0008 8158h	TPU4	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK		773
0008 815Ah	TPU4	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK		773
0008 8160h	TPU5	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK		753
0008 8161h	TPU5	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	758	
0008 8162h	TPU5	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK	759	
0008 8164h	TPU5	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	769	
0008 8165h	TPU5	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	770	
0008 8166h	TPU5	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	772	
0008 8168h	TPU5	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	773	
0008 816Ah	TPU5	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	773	
0008 8170h	TPUB	Timer start register	TSTR	8	8	2, 3 PCLKB	2 ICLK	774	
0008 8171h	TPUB	Timer synchronous register	TSYR	8	8	2, 3 PCLKB	2 ICLK	775	
0008 8178h	TPU6	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK	776	
0008 8179h	TPU7	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK	776	
0008 817Ah	TPU8	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK	776	
0008 817Bh	TPU9	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK	776	
0008 817Ch	TPU10	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK	776	
0008 817Dh	TPU11	Noise filter control register	NFCR	8	8	2, 3 PCLKB	2 ICLK	776	
0008 8180h	TPU6	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	753	
0008 8181h	TPU6	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	758	
0008 8182h	TPU6	Timer I/O control register H	TIORH	8	8	2, 3 PCLKB	2 ICLK	759	
0008 8183h	TPU6	Timer I/O control register L	TIORL	8	8	2, 3 PCLKB	2 ICLK	759	
0008 8184h	TPU6	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	769	
0008 8185h	TPU6	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	770	
0008 8186h	TPU6	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	772	
0008 8188h	TPU6	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	773	
0008 818Ah	TPU6	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	773	
0008 818Ch	TPU6	Timer general register C	TGRC	16	16	2, 3 PCLKB	2 ICLK	773	
0008 818Eh	TPU6	Timer general register D	TGRD	16	16	2, 3 PCLKB	2 ICLK	773	
0008 8190h	TPU7	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	753	
0008 8191h	TPU7	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	758	
0008 8192h	TPU7	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK	759	
0008 8194h	TPU7	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	769	
0008 8195h	TPU7	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	770	
0008 8196h	TPU7	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	772	
0008 8198h	TPU7	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	773	
0008 819Ah	TPU7	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	773	
0008 81A0h	TPU8	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	753	
0008 81A1h	TPU8	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	758	
0008 81A2h	TPU8	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK	759	
0008 81A4h	TPU8	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	769	
0008 81A5h	TPU8	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	770	
0008 81A6h	TPU8	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	772	
0008 81A8h	TPU8	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	773	
0008 81AAh	TPU8	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	773	
0008 81B0h	TPU9	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	753	
0008 81B1h	TPU9	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	758	
0008 81B2h	TPU9	Timer I/O control register H	TIORH	8	8	2, 3 PCLKB	2 ICLK	759	
0008 81B3h	TPU9	Timer I/O control register L	TIORL	8	8	2, 3 PCLKB	2 ICLK	759	

Table 5.1 List of I/O Registers (Address Order) (16/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 81B4h	TPU9	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	TPUa	769
0008 81B5h	TPU9	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK		770
0008 81B6h	TPU9	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK		772
0008 81B8h	TPU9	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK		773
0008 81BAh	TPU9	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK		773
0008 81BCh	TPU9	Timer general register C	TGRC	16	16	2, 3 PCLKB	2 ICLK		773
0008 81BEh	TPU9	Timer general register D	TGRD	16	16	2, 3 PCLKB	2 ICLK		773
0008 81C0h	TPU10	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK		753
0008 81C1h	TPU10	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK		758
0008 81C2h	TPU10	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK		759
0008 81C4h	TPU10	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK		769
0008 81C5h	TPU10	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK		770
0008 81C6h	TPU10	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK		772
0008 81C8h	TPU10	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK		773
0008 81CAh	TPU10	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	773	
0008 81D0h	TPU11	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	753	
0008 81D1h	TPU11	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK	758	
0008 81D2h	TPU11	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK	759	
0008 81D4h	TPU11	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK	769	
0008 81D5h	TPU11	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK	770	
0008 81D6h	TPU11	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK	772	
0008 81D8h	TPU11	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK	773	
0008 81DAh	TPU11	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	773	
0008 81E6h	PPG0	PPG output control register	PCR	8	8	2, 3 PCLKB	2 ICLK	PPG	832
0008 81E7h	PPG0	PPG output mode register	PMR	8	8	2, 3 PCLKB	2 ICLK		834
0008 81E8h	PPG0	Next data enable register H	NDERH	8	8	2, 3 PCLKB	2 ICLK		823
0008 81E9h	PPG0	Next data enable register L	NDERL	8	8	2, 3 PCLKB	2 ICLK		823
0008 81EAh	PPG0	Output data register H	PODRH	8	8	2, 3 PCLKB	2 ICLK		825
0008 81EBh	PPG0	Output data register L	PODRL	8	8	2, 3 PCLKB	2 ICLK		825
0008 81ECh*1	PPG0	Next data register H	NDRH	8	8	2, 3 PCLKB	2 ICLK		828
0008 81EDh*2	PPG0	Next data register L	NDRL	8	8	2, 3 PCLKB	2 ICLK		828
0008 81EEh*1	PPG0	Next data register H	NDRH2	8	8	2, 3 PCLKB	2 ICLK		828
0008 81EFh*2	PPG0	Next data register L	NDRL2	8	8	2, 3 PCLKB	2 ICLK		828
0008 81F0h	PPG1	PPG trigger select register	PTRSLR	8	8	2, 3 PCLKB	2 ICLK		822
0008 81F6h	PPG1	PPG output control register	PCR	8	8	2, 3 PCLKB	2 ICLK		832
0008 81F7h	PPG1	PPG output mode register	PMR	8	8	2, 3 PCLKB	2 ICLK		832
0008 81F8h	PPG1	Next data enable register H	NDERH	8	8	2, 3 PCLKB	2 ICLK		823
0008 81F9h	PPG1	Next data enable register L	NDERL	8	8	2, 3 PCLKB	2 ICLK	823	
0008 81FAh	PPG1	Output data register H	PODRH	8	8	2, 3 PCLKB	2 ICLK	825	
0008 81FBh	PPG1	Output data register L	PODRL	8	8	2, 3 PCLKB	2 ICLK	825	
0008 81FCh*3	PPG1	Next data register H	NDRH	8	8	2, 3 PCLKB	2 ICLK	828	
0008 81FDh*4	PPG1	Next data register L	NDRL	8	8	2, 3 PCLKB	2 ICLK	825	
0008 81FEh*3	PPG1	Next data register H	NDRH2	8	8	2, 3 PCLKB	2 ICLK	825	
0008 81FFh*4	PPG1	Next data register L	NDRL2	8	8	2, 3 PCLKB	2 ICLK	825	
0008 8200h	TMR0	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	TMR	855
0008 8201h	TMR1	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK		855
0008 8202h	TMR0	Timer control/status register	TCSR	8	8	2, 3 PCLKB	2 ICLK		858
0008 8203h	TMR1	Timer control/status register	TCSR	8	8	2, 3 PCLKB	2 ICLK		858
0008 8204h	TMR0	Time constant register A	TCORA	8	8	2, 3 PCLKB	2 ICLK		854
0008 8205h	TMR1	Time constant register A	TCORA	8	8*5	2, 3 PCLKB	2 ICLK		854

Table 5.1 List of I/O Registers (Address Order) (17/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page	
						ICLK ≥ PCLK	ICLK < PCLK			
0008 8206h	TMR0	Time constant register B	TCORB	8	8	2, 3 PCLKB	2 ICLK	TMR	855	
0008 8207h	TMR1	Time constant register B	TCORB	8	8*5	2, 3 PCLKB	2 ICLK		855	
0008 8208h	TMR0	Timer counter	TCNT	8	8	2, 3 PCLKB	2 ICLK		854	
0008 8209h	TMR1	Timer counter	TCNT	8	8*5	2, 3 PCLKB	2 ICLK		858	
0008 820Ah	TMR0	Timer counter control register	TCCR	8	8	2, 3 PCLKB	2 ICLK		856	
0008 820Bh	TMR1	Timer counter control register	TCCR	8	8*5	2, 3 PCLKB	2 ICLK		856	
0008 8210h	TMR2	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK		855	
0008 8211h	TMR3	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK		855	
0008 8212h	TMR2	Timer control/status register	TCSR	8	8	2, 3 PCLKB	2 ICLK		858	
0008 8213h	TMR3	Timer control/status register	TCSR	8	8	2, 3 PCLKB	2 ICLK		858	
0008 8214h	TMR2	Time constant register A	TCORA	8	8	2, 3 PCLKB	2 ICLK		854	
0008 8215h	TMR3	Time constant register A	TCORA	8	8*5	2, 3 PCLKB	2 ICLK		854	
0008 8216h	TMR2	Time constant register B	TCORB	8	8	2, 3 PCLKB	2 ICLK		855	
0008 8217h	TMR3	Time constant register B	TCORB	8	8*5	2, 3 PCLKB	2 ICLK		855	
0008 8218h	TMR2	Timer counter	TCNT	8	8	2, 3 PCLKB	2 ICLK		854	
0008 8219h	TMR3	Timer counter	TCNT	8	8*5	2, 3 PCLKB	2 ICLK		858	
0008 821Ah	TMR2	Timer counter control register	TCCR	8	8	2, 3 PCLKB	2 ICLK		856	
0008 821Bh	TMR3	Timer counter control register	TCCR	8	8*5	2, 3 PCLKB	2 ICLK		856	
0008 8280h	CRC	CRC control register	CRCCR	8	8	2, 3 PCLKB	2 ICLK		CRC	1386
0008 8281h	CRC	CRC data input register	CRCDIR	8	8	2, 3 PCLKB	2 ICLK			1386
0008 8282h	CRC	CRC data output register	CRCDOR	16	16	2, 3 PCLKB	2 ICLK			1387
0008 8300h	RIIC0	I <sup>2</sup> C bus control register 1	ICCR1	8	8	2, 3 PCLKB	2 ICLK		RIIC	1150
0008 8301h	RIIC0	I <sup>2</sup> C bus control register 2	ICCR2	8	8	2, 3 PCLKB	2 ICLK			1152
0008 8302h	RIIC0	I <sup>2</sup> C bus mode register 1	ICMR1	8	8	2, 3 PCLKB	2 ICLK			1155
0008 8303h	RIIC0	I <sup>2</sup> C bus mode register 2	ICMR2	8	8	2, 3 PCLKB	2 ICLK			1156
0008 8304h	RIIC0	I <sup>2</sup> C bus mode register 3	ICMR3	8	8	2, 3 PCLKB	2 ICLK			1158
0008 8305h	RIIC0	I <sup>2</sup> C bus function enable register	ICFER	8	8	2, 3 PCLKB	2 ICLK			1160
0008 8306h	RIIC0	I <sup>2</sup> C bus status enable register	ICSER	8	8	2, 3 PCLKB	2 ICLK			1162
0008 8307h	RIIC0	I <sup>2</sup> C bus interrupt enable register	ICIER	8	8	2, 3 PCLKB	2 ICLK			1163
0008 8308h	RIIC0	I <sup>2</sup> C bus status register 1	ICSR1	8	8	2, 3 PCLKB	2 ICLK			1164
0008 8309h	RIIC0	I <sup>2</sup> C bus status register 2	ICSR2	8	8	2, 3 PCLKB	2 ICLK			1167
0008 830Ah	RIIC0	Slave address register L0	SARL0	8	8	2, 3 PCLKB	2 ICLK			1170
0008 830Bh	RIIC0	Slave address register U0	SARU0	8	8	2, 3 PCLKB	2 ICLK	1171		
0008 830Ch	RIIC0	Slave address register L1	SARL1	8	8	2, 3 PCLKB	2 ICLK	1170		
0008 830Dh	RIIC0	Slave address register U1	SARU1	8	8	2, 3 PCLKB	2 ICLK	1171		
0008 830Eh	RIIC0	Slave address register L2	SARL2	8	8	2, 3 PCLKB	2 ICLK	1170		
0008 830Fh	RIIC0	Slave address register U2	SARU2	8	8	2, 3 PCLKB	2 ICLK	1171		
0008 8310h	RIIC0	I <sup>2</sup> C bus bit rate low-level register	ICBRL	8	8	2, 3 PCLKB	2 ICLK	1172		
0008 8311h	RIIC0	I <sup>2</sup> C bus bit rate high-level register	ICBRH	8	8	2, 3 PCLKB	2 ICLK	1173		
0008 8312h	RIIC0	I <sup>2</sup> C bus transmit data register	ICDRT	8	8	2, 3 PCLKB	2 ICLK	1175		
0008 8313h	RIIC0	I <sup>2</sup> C bus receive data register	ICDRR	8	8	2, 3 PCLKB	2 ICLK	1175		
0008 8320h	RIIC1	I <sup>2</sup> C bus control register 1	ICCR1	8	8	2, 3 PCLKB	2 ICLK	1150		
0008 8321h	RIIC1	I <sup>2</sup> C bus control register 2	ICCR2	8	8	2, 3 PCLKB	2 ICLK	1152		
0008 8322h	RIIC1	I <sup>2</sup> C bus mode register 1	ICMR1	8	8	2, 3 PCLKB	2 ICLK	1155		
0008 8323h	RIIC1	I <sup>2</sup> C bus mode register 2	ICMR2	8	8	2, 3 PCLKB	2 ICLK	1156		
0008 8324h	RIIC1	I <sup>2</sup> C bus mode register 3	ICMR3	8	8	2, 3 PCLKB	2 ICLK	1158		
0008 8325h	RIIC1	I <sup>2</sup> C bus function enable register	ICFER	8	8	2, 3 PCLKB	2 ICLK	1160		
0008 8326h	RIIC1	I <sup>2</sup> C bus status enable register	ICSER	8	8	2, 3 PCLKB	2 ICLK	1162		
0008 8327h	RIIC1	I <sup>2</sup> C bus interrupt enable register	ICIER	8	8	2, 3 PCLKB	2 ICLK	1163		
0008 8328h	RIIC1	I <sup>2</sup> C bus status register 1	ICSR1	8	8	2, 3 PCLKB	2 ICLK	1164		



Table 5.1 List of I/O Registers (Address Order) (18/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 8329h	RIIC1	I <sup>2</sup> C bus status register 2	ICSR2	8	8	2, 3 PCLKB	2 ICLK	RIIC	1167
0008 832Ah	RIIC1	Slave address register L0	SARL0	8	8	2, 3 PCLKB	2 ICLK		1170
0008 832Bh	RIIC1	Slave address register U0	SARU0	8	8	2, 3 PCLKB	2 ICLK		1171
0008 832Ch	RIIC1	Slave address register L1	SARL1	8	8	2, 3 PCLKB	2 ICLK		1170
0008 832Dh	RIIC1	Slave address register U1	SARU1	8	8	2, 3 PCLKB	2 ICLK		1171
0008 832Eh	RIIC1	Slave address register L2	SARL2	8	8	2, 3 PCLKB	2 ICLK		1170
0008 832Fh	RIIC1	Slave address register U2	SARU2	8	8	2, 3 PCLKB	2 ICLK		1171
0008 8330h	RIIC1	I <sup>2</sup> C bus bit rate low-level register	ICBRL	8	8	2, 3 PCLKB	2 ICLK		1172
0008 8331h	RIIC1	I <sup>2</sup> C bus bit rate high-level register	ICBRH	8	8	2, 3 PCLKB	2 ICLK		1173
0008 8332h	RIIC1	I <sup>2</sup> C bus transmit data register	ICDRT	8	8	2, 3 PCLKB	2 ICLK		1175
0008 8333h	RIIC1	I <sup>2</sup> C bus receive data register	ICDRR	8	8	2, 3 PCLKB	2 ICLK		1175
0008 8340h	RIIC2	I <sup>2</sup> C bus control register 1	ICCR1	8	8	2, 3 PCLKB	2 ICLK		1150
0008 8341h	RIIC2	I <sup>2</sup> C bus control register 2	ICCR2	8	8	2, 3 PCLKB	2 ICLK		1152
0008 8342h	RIIC2	I <sup>2</sup> C bus mode register 1	ICMR1	8	8	2, 3 PCLKB	2 ICLK		1155
0008 8343h	RIIC2	I <sup>2</sup> C bus mode register 2	ICMR2	8	8	2, 3 PCLKB	2 ICLK		1156
0008 8344h	RIIC2	I <sup>2</sup> C bus mode register 3	ICMR3	8	8	2, 3 PCLKB	2 ICLK		1158
0008 8345h	RIIC2	I <sup>2</sup> C bus function enable register	ICFER	8	8	2, 3 PCLKB	2 ICLK		1160
0008 8346h	RIIC2	I <sup>2</sup> C bus status enable register	ICSER	8	8	2, 3 PCLKB	2 ICLK	1162	
0008 8347h	RIIC2	I <sup>2</sup> C bus interrupt enable register	ICIER	8	8	2, 3 PCLKB	2 ICLK	1163	
0008 8348h	RIIC2	I <sup>2</sup> C bus status register 1	ICSR1	8	8	2, 3 PCLKB	2 ICLK	1164	
0008 8349h	RIIC2	I <sup>2</sup> C bus status register 2	ICSR2	8	8	2, 3 PCLKB	2 ICLK	1167	
0008 834Ah	RIIC2	Slave address register L0	SARL0	8	8	2, 3 PCLKB	2 ICLK	1170	
0008 834Bh	RIIC2	Slave address register U0	SARU0	8	8	2, 3 PCLKB	2 ICLK	1171	
0008 834Ch	RIIC2	Slave address register L1	SARL1	8	8	2, 3 PCLKB	2 ICLK	1170	
0008 834Dh	RIIC2	Slave address register U1	SARU1	8	8	2, 3 PCLKB	2 ICLK	1171	
0008 834Eh	RIIC2	Slave address register L2	SARL2	8	8	2, 3 PCLKB	2 ICLK	1170	
0008 834Fh	RIIC2	Slave address register U2	SARU2	8	8	2, 3 PCLKB	2 ICLK	1171	
0008 8350h	RIIC2	I <sup>2</sup> C bus bit rate low-level register	ICBRL	8	8	2, 3 PCLKB	2 ICLK	1172	
0008 8351h	RIIC2	I <sup>2</sup> C bus bit rate high-level register	ICBRH	8	8	2, 3 PCLKB	2 ICLK	1173	
0008 8352h	RIIC2	I <sup>2</sup> C bus transmit data register	ICDRT	8	8	2, 3 PCLKB	2 ICLK	1175	
0008 8353h	RIIC2	I <sup>2</sup> C bus receive data register	ICDRR	8	8	2, 3 PCLKB	2 ICLK	1175	
0008 8360h	RIIC3	I <sup>2</sup> C bus control register 1	ICCR1	8	8	2, 3 PCLKB	2 ICLK	1150	
0008 8361h	RIIC3	I <sup>2</sup> C bus control register 2	ICCR2	8	8	2, 3 PCLKB	2 ICLK	1152	
0008 8362h	RIIC3	I <sup>2</sup> C bus mode register 1	ICMR1	8	8	2, 3 PCLKB	2 ICLK	1155	
0008 8363h	RIIC3	I <sup>2</sup> C bus mode register 2	ICMR2	8	8	2, 3 PCLKB	2 ICLK	1156	
0008 8364h	RIIC3	I <sup>2</sup> C bus mode register 3	ICMR3	8	8	2, 3 PCLKB	2 ICLK	1158	
0008 8365h	RIIC3	I <sup>2</sup> C bus function enable register	ICFER	8	8	2, 3 PCLKB	2 ICLK	1160	
0008 8366h	RIIC3	I <sup>2</sup> C bus status enable register	ICSER	8	8	2, 3 PCLKB	2 ICLK	1162	
0008 8367h	RIIC3	I <sup>2</sup> C bus interrupt enable register	ICIER	8	8	2, 3 PCLKB	2 ICLK	1163	
0008 8368h	RIIC3	I <sup>2</sup> C bus status register 1	ICSR1	8	8	2, 3 PCLKB	2 ICLK	1164	
0008 8369h	RIIC3	I <sup>2</sup> C bus status register 2	ICSR2	8	8	2, 3 PCLKB	2 ICLK	1167	
0008 836Ah	RIIC3	Slave address register L0	SARL0	8	8	2, 3 PCLKB	2 ICLK	1170	
0008 836Bh	RIIC3	Slave address register U0	SARU0	8	8	2, 3 PCLKB	2 ICLK	1171	
0008 836Ch	RIIC3	Slave address register L1	SARL1	8	8	2, 3 PCLKB	2 ICLK	1170	
0008 836Dh	RIIC3	Slave address register U1	SARU1	8	8	2, 3 PCLKB	2 ICLK	1171	
0008 836Eh	RIIC3	Slave address register L2	SARL2	8	8	2, 3 PCLKB	2 ICLK	1170	
0008 836Fh	RIIC3	Slave address register U2	SARU2	8	8	2, 3 PCLKB	2 ICLK	1171	
0008 8370h	RIIC3	I <sup>2</sup> C bus bit rate low-level register	ICBRL	8	8	2, 3 PCLKB	2 ICLK	1172	
0008 8371h	RIIC3	I <sup>2</sup> C bus bit rate high-level register	ICBRH	8	8	2, 3 PCLKB	2 ICLK	1173	
0008 8372h	RIIC3	I <sup>2</sup> C bus transmit data register	ICDRT	8	8	2, 3 PCLKB	2 ICLK	1175	

Table 5.1 List of I/O Registers (Address Order) (19/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 8373h	RIIC3	I <sup>2</sup> C bus receive data register	ICDRR	8	8	2, 3 PCLKB	2 ICLK	RIIC	1175
0008 8380h	RSPI0	RSPI control register	SPCR	8	8	2, 3 PCLKB	2 ICLK	RSPI	1273
0008 8381h	RSPI0	RSPI slave select polarity register	SSLP	8	8	2, 3 PCLKB	2 ICLK		1274
0008 8382h	RSPI0	RSPI pin control register	SPPCR	8	8	2, 3 PCLKB	2 ICLK		1275
0008 8383h	RSPI0	RSPI status register	SPSR	8	8	2, 3 PCLKB	2 ICLK		1276
0008 8384h	RSPI0	RSPI data register	SPDR	32	16, 32	2, 3 PCLKB	2 ICLK		1278
0008 8388h	RSPI0	RSPI sequence control register	SPSCR	8	8	2, 3 PCLKB	2 ICLK		1281
0008 8389h	RSPI0	RSPI sequence status register	SPSSR	8	8	2, 3 PCLKB	2 ICLK		1282
0008 838Ah	RSPI0	RSPI bit rate register	SPBR	8	8	2, 3 PCLKB	2 ICLK		1283
0008 838Bh	RSPI0	RSPI data control register	SPDCR	8	8	2, 3 PCLKB	2 ICLK		1284
0008 838Ch	RSPI0	RSPI clock delay register	SPCKD	8	8	2, 3 PCLKB	2 ICLK		1285
0008 838Dh	RSPI0	RSPI slave select negation delay register	SSLND	8	8	2, 3 PCLKB	2 ICLK		1286
0008 838Eh	RSPI0	RSPI next-access delay register	SPND	8	8	2, 3 PCLKB	2 ICLK		1286
0008 838Fh	RSPI0	RSPI control register 2	SPCR2	8	8	2, 3 PCLKB	2 ICLK		1287
0008 8390h	RSPI0	RSPI command register 0	SPCMD0	16	16	2, 3 PCLKB	2 ICLK		1288
0008 8392h	RSPI0	RSPI command register 1	SPCMD1	16	16	2, 3 PCLKB	2 ICLK		1288
0008 8394h	RSPI0	RSPI command register 2	SPCMD2	16	16	2, 3 PCLKB	2 ICLK		1288
0008 8396h	RSPI0	RSPI command register 3	SPCMD3	16	16	2, 3 PCLKB	2 ICLK		1288
0008 8398h	RSPI0	RSPI command register 4	SPCMD4	16	16	2, 3 PCLKB	2 ICLK		1288
0008 839Ah	RSPI0	RSPI command register 5	SPCMD5	16	16	2, 3 PCLKB	2 ICLK		1288
0008 839Ch	RSPI0	RSPI command register 6	SPCMD6	16	16	2, 3 PCLKB	2 ICLK	1288	
0008 839Eh	RSPI0	RSPI command register 7	SPCMD7	16	16	2, 3 PCLKB	2 ICLK	1288	
0008 83A0h	RSPI1	RSPI control register	SPCR	8	8	2, 3 PCLKB	2 ICLK	1273	
0008 83A1h	RSPI1	RSPI slave select polarity register	SSLP	8	8	2, 3 PCLKB	2 ICLK	1274	
0008 83A2h	RSPI1	RSPI pin control register	SPPCR	8	8	2, 3 PCLKB	2 ICLK	1275	
0008 83A3h	RSPI1	RSPI status register	SPSR	8	8	2, 3 PCLKB	2 ICLK	1276	
0008 83A4h	RSPI1	RSPI data register	SPDR	32	16, 32	2, 3 PCLKB	2 ICLK	1278	
0008 83A8h	RSPI1	RSPI sequence control register	SPSCR	8	8	2, 3 PCLKB	2 ICLK	1281	
0008 83A9h	RSPI1	RSPI sequence status register	SPSSR	8	8	2, 3 PCLKB	2 ICLK	1282	
0008 83AAh	RSPI1	RSPI bit rate register	SPBR	8	8	2, 3 PCLKB	2 ICLK	1283	
0008 83ABh	RSPI1	RSPI data control register	SPDCR	8	8	2, 3 PCLKB	2 ICLK	1284	
0008 83ACh	RSPI1	RSPI clock delay register	SPCKD	8	8	2, 3 PCLKB	2 ICLK	1285	
0008 83ADh	RSPI1	RSPI slave select negation delay register	SSLND	8	8	2, 3 PCLKB	2 ICLK	1286	
0008 83AEh	RSPI1	RSPI next-access delay register	SPND	8	8	2, 3 PCLKB	2 ICLK	1286	
0008 83AFh	RSPI1	RSPI control register 2	SPCR2	8	8	2, 3 PCLKB	2 ICLK	1287	
0008 83B0h	RSPI1	RSPI command register 0	SPCMD0	16	16	2, 3 PCLKB	2 ICLK	1288	
0008 83B2h	RSPI1	RSPI command register 1	SPCMD1	16	16	2, 3 PCLKB	2 ICLK	1288	
0008 83B4h	RSPI1	RSPI command register 2	SPCMD2	16	16	2, 3 PCLKB	2 ICLK	1288	
0008 83B6h	RSPI1	RSPI command register 3	SPCMD3	16	16	2, 3 PCLKB	2 ICLK	1288	
0008 83B8h	RSPI1	RSPI command register 4	SPCMD4	16	16	2, 3 PCLKB	2 ICLK	1288	
0008 83BAh	RSPI1	RSPI command register 5	SPCMD5	16	16	2, 3 PCLKB	2 ICLK	1288	
0008 83BCh	RSPI1	RSPI command register 6	SPCMD6	16	16	2, 3 PCLKB	2 ICLK	1288	
0008 83BEh	RSPI1	RSPI command register 7	SPCMD7	16	16	2, 3 PCLKB	2 ICLK	1288	
0008 83C0h	RSPI2	RSPI control register	SPCR	8	8	2, 3 PCLKB	2 ICLK	1273	
0008 83C1h	RSPI2	RSPI slave select polarity register	SSLP	8	8	2, 3 PCLKB	2 ICLK	1274	
0008 83C2h	RSPI2	RSPI pin control register	SPPCR	8	8	2, 3 PCLKB	2 ICLK	1275	
0008 83C3h	RSPI2	RSPI status register	SPSR	8	8	2, 3 PCLKB	2 ICLK	1276	
0008 83C4h	RSPI2	RSPI data register	SPDR	32	16, 32	2, 3 PCLKB	2 ICLK	1278	
0008 83C8h	RSPI2	RSPI sequence control register	SPSCR	8	8	2, 3 PCLKB	2 ICLK	1281	
0008 83C9h	RSPI2	RSPI sequence status register	SPSSR	8	8	2, 3 PCLKB	2 ICLK	1282	

Table 5.1 List of I/O Registers (Address Order) (20/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 83CAh	RSPI2	RSPI bit rate register	SPBR	8	8	2, 3 PCLKB	2 ICLK	RSPI	1283
0008 83CBh	RSPI2	RSPI data control register	SPDCR	8	8	2, 3 PCLKB	2 ICLK		1284
0008 83CCh	RSPI2	RSPI clock delay register	SPCKD	8	8	2, 3 PCLKB	2 ICLK		1285
0008 83CDh	RSPI2	RSPI slave select negation delay register	SSLND	8	8	2, 3 PCLKB	2 ICLK		1286
0008 83CEh	RSPI2	RSPI next-access delay register	SPND	8	8	2, 3 PCLKB	2 ICLK		1286
0008 83CFh	RSPI2	RSPI control register 2	SPCR2	8	8	2, 3 PCLKB	2 ICLK		1287
0008 83D0h	RSPI2	RSPI command register 0	SPCMD0	16	16	2, 3 PCLKB	2 ICLK		1288
0008 83D2h	RSPI2	RSPI command register 1	SPCMD1	16	16	2, 3 PCLKB	2 ICLK		1288
0008 83D4h	RSPI2	RSPI command register 2	SPCMD2	16	16	2, 3 PCLKB	2 ICLK		1288
0008 83D6h	RSPI2	RSPI command register 3	SPCMD3	16	16	2, 3 PCLKB	2 ICLK		1288
0008 83D8h	RSPI2	RSPI command register 4	SPCMD4	16	16	2, 3 PCLKB	2 ICLK		1288
0008 83DAh	RSPI2	RSPI command register 5	SPCMD5	16	16	2, 3 PCLKB	2 ICLK		1288
0008 83DCh	RSPI2	RSPI command register 6	SPCMD6	16	16	2, 3 PCLKB	2 ICLK		1288
0008 83DEh	RSPI2	RSPI command register 7	SPCMD7	16	16	2, 3 PCLKB	2 ICLK		1288
0008 8600h	MTU3	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	MTU2a	573
0008 8601h	MTU4	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK		573
0008 8602h	MTU3	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK		576
0008 8603h	MTU4	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK		576
0008 8604h	MTU3	Timer I/O control register H	TIORH	8	8	2, 3 PCLKB	2 ICLK		578
0008 8605h	MTU3	Timer I/O control register L	TIORL	8	8	2, 3 PCLKB	2 ICLK		578
0008 8606h	MTU4	Timer I/O control register H	TIORH	8	8	2, 3 PCLKB	2 ICLK		578
0008 8607h	MTU4	Timer I/O control register L	TIORL	8	8	2, 3 PCLKB	2 ICLK		578
0008 8608h	MTU3	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK		590
0008 8609h	MTU4	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK		590
0008 860Ah	MTU	Timer output master enable register	TOER	8	8	2, 3 PCLKB	2 ICLK		601
0008 860Dh	MTU	Timer gate control register	TGCR	8	8	2, 3 PCLKB	2 ICLK		607
0008 860Eh	MTU	Timer output control register 1	TOCR1	8	8	2, 3 PCLKB	2 ICLK		602
0008 860Fh	MTU	Timer output control register 2	TOCR2	8	8	2, 3 PCLKB	2 ICLK		604
0008 8610h	MTU3	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK		597
0008 8612h	MTU4	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK		597
0008 8614h	MTU	Timer cycle data register	TCDR	16	16	2, 3 PCLKB	2 ICLK		609
0008 8616h	MTU	Timer dead time data register	TDDR	16	16	2, 3 PCLKB	2 ICLK		608
0008 8618h	MTU3	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK		597
0008 861Ah	MTU3	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK		597
0008 861Ch	MTU4	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK		597
0008 861Eh	MTU4	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK		597
0008 8620h	MTU	Timer subcounter	TCNTS	16	16	2, 3 PCLKB	2 ICLK		608
0008 8622h	MTU	Timer cycle buffer register	TCBR	16	16	2, 3 PCLKB	2 ICLK		609
0008 8624h	MTU3	Timer general register C	TGRC	16	16	2, 3 PCLKB	2 ICLK		597
0008 8626h	MTU3	Timer general register D	TGRD	16	16	2, 3 PCLKB	2 ICLK		597
0008 8628h	MTU4	Timer general register C	TGRC	16	16	2, 3 PCLKB	2 ICLK		597
0008 862Ah	MTU4	Timer general register D	TGRD	16	16	2, 3 PCLKB	2 ICLK		597
0008 862Ch	MTU3	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK		592
0008 862Dh	MTU4	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK		592
0008 8630h	MTU	Timer interrupt skipping set register	TITCR	8	8	2, 3 PCLKB	2 ICLK		609
0008 8631h	MTU	Timer interrupt skipping counter	TITCNT	8	8	2, 3 PCLKB	2 ICLK		611
0008 8632h	MTU	Timer buffer transfer set register	TBTER	8	8	2, 3 PCLKB	2 ICLK		612
0008 8634h	MTU	Timer dead time enable register	TDER	8	8	2, 3 PCLKB	2 ICLK		613
0008 8636h	MTU	Timer output level buffer register	TOLBR	8	8	2, 3 PCLKB	2 ICLK		606
0008 8638h	MTU3	Timer buffer operation transfer mode register	TBTM	8	8	2, 3 PCLKB	2 ICLK		593

Table 5.1 List of I/O Registers (Address Order) (21/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 8639h	MTU4	Timer buffer operation transfer mode register	TBTM	8	8	2, 3 PCLKB	2 ICLK	MTU2a	593
0008 8640h	MTU4	Timer A/D converter start request control register	TADCR	16	16	2, 3 PCLKB	2 ICLK		595
0008 8644h	MTU4	Timer A/D converter start request cycle set register A	TADCORA	16	16	2, 3 PCLKB	2 ICLK		596
0008 8646h	MTU4	Timer A/D converter start request cycle set register B	TADCORB	16	16	2, 3 PCLKB	2 ICLK		596
0008 8648h	MTU4	Timer A/D converter start request cycle set buffer register A	TADCOBRA	16	16	2, 3 PCLKB	2 ICLK		596
0008 864Ah	MTU4	Timer A/D converter start request cycle set buffer register B	TADCOBRB	16	16	2, 3 PCLKB	2 ICLK		596
0008 8660h	MTU	Timer waveform control register	TWCR	8	8, 16	2, 3 PCLKB	2 ICLK		614
0008 8680h	MTU	Timer start register	TSTR	8	8, 16	2, 3 PCLKB	2 ICLK		598
0008 8681h	MTU	Timer synchronous register	TSYR	8	8, 16	2, 3 PCLKB	2 ICLK		599
0008 8684h	MTU	Timer read/write enable register	TRWER	8	8, 16	2, 3 PCLKB	2 ICLK		600
0008 8690h	MTU0	Noise filter control register	NFCR	8	8, 16	2, 3 PCLKB	2 ICLK		615
0008 8691h	MTU1	Noise filter control register	NFCR	8	8, 16	2, 3 PCLKB	2 ICLK		615
0008 8692h	MTU2	Noise filter control register	NFCR	8	8, 16	2, 3 PCLKB	2 ICLK		615
0008 8693h	MTU3	Noise filter control register	NFCR	8	8, 16	2, 3 PCLKB	2 ICLK		615
0008 8694h	MTU4	Noise filter control register	NFCR	8	8, 16	2, 3 PCLKB	2 ICLK		615
0008 8695h	MTU5	Noise filter control register	NFCR	8	8, 16	2, 3 PCLKB	2 ICLK		615
0008 8700h	MTU0	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK		573
0008 8701h	MTU0	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK		576
0008 8702h	MTU0	Timer I/O control register H	TIORH	8	8	2, 3 PCLKB	2 ICLK		578
0008 8703h	MTU0	Timer I/O control register L	TIORL	8	8	2, 3 PCLKB	2 ICLK		578
0008 8704h	MTU0	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK		590
0008 8705h	MTU0	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK		592
0008 8706h	MTU0	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK		597
0008 8708h	MTU0	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK		597
0008 870Ah	MTU0	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK		597
0008 870Ch	MTU0	Timer general register C	TGRC	16	16	2, 3 PCLKB	2 ICLK		597
0008 870Eh	MTU0	Timer general register D	TGRD	16	16	2, 3 PCLKB	2 ICLK		597
0008 8720h	MTU0	Timer general register E	TGRE	16	16	2, 3 PCLKB	2 ICLK		597
0008 8722h	MTU0	Timer general register F	TGRF	16	16	2, 3 PCLKB	2 ICLK		597
0008 8724h	MTU0	Timer interrupt enable register2	TIER2	8	8	2, 3 PCLKB	2 ICLK		590
0008 8726h	MTU0	Timer buffer operation transfer mode register	TBTM	8	8	2, 3 PCLKB	2 ICLK		593
0008 8780h	MTU1	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK		573
0008 8781h	MTU1	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK		576
0008 8782h	MTU1	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK		578
0008 8784h	MTU1	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK		590
0008 8785h	MTU1	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK		592
0008 8786h	MTU1	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK		597
0008 8788h	MTU1	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK		597
0008 878Ah	MTU1	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK		597
0008 8790h	MTU1	Timer input capture control register	TICCR	8	8	2, 3 PCLKB	2 ICLK		594
0008 8800h	MTU2	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK		573
0008 8801h	MTU2	Timer mode register	TMDR	8	8	2, 3 PCLKB	2 ICLK		576
0008 8802h	MTU2	Timer I/O control register	TIOR	8	8	2, 3 PCLKB	2 ICLK		578
0008 8804h	MTU2	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK		590
0008 8805h	MTU2	Timer status register	TSR	8	8	2, 3 PCLKB	2 ICLK		592
0008 8806h	MTU2	Timer counter	TCNT	16	16	2, 3 PCLKB	2 ICLK		597
0008 8808h	MTU2	Timer general register A	TGRA	16	16	2, 3 PCLKB	2 ICLK		597

Table 5.1 List of I/O Registers (Address Order) (22/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 880Ah	MTU2	Timer general register B	TGRB	16	16	2, 3 PCLKB	2 ICLK	MTU2a	597
0008 8880h	MTU5	Timer counter U	TCNTU	16	16	2, 3 PCLKB	2 ICLK		597
0008 8882h	MTU5	Timer general register U	TGRU	16	16	2, 3 PCLKB	2 ICLK		597
0008 8884h	MTU5	Timer control register U	TCRU	8	8	2, 3 PCLKB	2 ICLK		573
0008 8886h	MTU5	Timer I/O control register U	TIORU	8	8	2, 3 PCLKB	2 ICLK		578
0008 8890h	MTU5	Timer counter V	TCNTV	16	16	2, 3 PCLKB	2 ICLK		597
0008 8892h	MTU5	Timer general register V	TGRV	16	16	2, 3 PCLKB	2 ICLK		597
0008 8894h	MTU5	Timer control register V	TCRV	8	8	2, 3 PCLKB	2 ICLK		573
0008 8896h	MTU5	Timer I/O control register V	TIORV	8	8	2, 3 PCLKB	2 ICLK		578
0008 88A0h	MTU5	Timer counter W	TCNTW	16	16	2, 3 PCLKB	2 ICLK		597
0008 88A2h	MTU5	Timer general register W	TGRW	16	16	2, 3 PCLKB	2 ICLK		597
0008 88A4h	MTU5	Timer control register W	TCRW	8	8	2, 3 PCLKB	2 ICLK		573
0008 88A6h	MTU5	Timer I/O control register W	TIORW	8	8	2, 3 PCLKB	2 ICLK		578
0008 88B2h	MTU5	Timer interrupt enable register	TIER	8	8	2, 3 PCLKB	2 ICLK		590
0008 88B4h	MTU5	Timer start register	TSTR	8	8	2, 3 PCLKB	2 ICLK		592
0008 88B6h	MTU5	Timer compare match clear register	TCNTCMPCLR	8	8	2, 3 PCLKB	2 ICLK		589
0008 8900h	POE	Input level control/status register 1	ICSR1	16	16	2, 3 PCLKB	2 ICLK	POE2a	735
0008 8902h	POE	Output level control/status register 1	OCSR1	16	16	2, 3 PCLKB	2 ICLK		737
0008 8908h	POE	Input level control/status register 2	ICSR2	16	16	2, 3 PCLKB	2 ICLK		738
0008 890Ah	POE	Software port output enable register	SPOER	8	8	2, 3 PCLKB	2 ICLK		739
0008 890Bh	POE	Port output enable control register 1	POECR1	8	8	2, 3 PCLKB	2 ICLK		740
0008 890Ch	POE	Port output enable control register 2	POECR2	8	8	2, 3 PCLKB	2 ICLK		740
0008 890Eh	POE	Input level control/status register 3	ICSR3	16	16	2, 3 PCLKB	2 ICLK	741	
0008 9000h	S12AD	A/D control register	ADCSR	8	8	2, 3 PCLKB	2 ICLK	S12ADa	1395
0008 9004h	S12AD	A/D channel select register 0	ADANS0	16	16	2, 3 PCLKB	2 ICLK		1397
0008 9006h	S12AD	A/D channel select register 1	ADANS1	16	16	2, 3 PCLKB	2 ICLK		1397
0008 9008h	S12AD	A/D-converted value addition mode select register 0	ADADS0	16	16	2, 3 PCLKB	2 ICLK		1398
0008 900Ah	S12AD	A/D-converted value addition mode select register 1	ADADS1	16	16	2, 3 PCLKB	2 ICLK		1398
0008 900Ch	S12AD	A/D-converted value addition count select register	ADADC	8	8	2, 3 PCLKB	2 ICLK		1399
0008 900Eh	S12AD	A/D control extended register	ADCER	16	16	2, 3 PCLKB	2 ICLK		1400
0008 9010h	S12AD	A/D start trigger select register	ADSTRGR	8	8	2, 3 PCLKB	2 ICLK		1401
0008 9012h	S12AD	A/D conversion extended input control register	ADEXICR	16	16	2, 3 PCLKB	2 ICLK		1402
0008 901Ah	S12AD	A/D temperature sensor data register	ADTSDR	16	16	2, 3 PCLKB	2 ICLK		1403
0008 901Ch	S12AD	A/D internal reference voltage data register	ADOCDR	16	16	2, 3 PCLKB	2 ICLK		1404
0008 9020h	S12AD	A/D data register 0	ADDR0	16	16	2, 3 PCLKB	2 ICLK		1405
0008 9022h	S12AD	A/D data register 1	ADDR1	16	16	2, 3 PCLKB	2 ICLK		1405
0008 9024h	S12AD	A/D data register 2	ADDR2	16	16	2, 3 PCLKB	2 ICLK		1405
0008 9026h	S12AD	A/D data register 3	ADDR3	16	16	2, 3 PCLKB	2 ICLK		1405
0008 9028h	S12AD	A/D data register 4	ADDR4	16	16	2, 3 PCLKB	2 ICLK		1405
0008 902Ah	S12AD	A/D data register 5	ADDR5	16	16	2, 3 PCLKB	2 ICLK	1405	
0008 902Ch	S12AD	A/D data register 6	ADDR6	16	16	2, 3 PCLKB	2 ICLK	1405	
0008 902Eh	S12AD	A/D data register 7	ADDR7	16	16	2, 3 PCLKB	2 ICLK	1405	
0008 9030h	S12AD	A/D data register 8	ADDR8	16	16	2, 3 PCLKB	2 ICLK	1405	
0008 9032h	S12AD	A/D data register 9	ADDR9	16	16	2, 3 PCLKB	2 ICLK	1405	
0008 9034h	S12AD	A/D data register 10	ADDR10	16	16	2, 3 PCLKB	2 ICLK	1405	
0008 9036h	S12AD	A/D data register 11	ADDR11	16	16	2, 3 PCLKB	2 ICLK	1405	
0008 9038h	S12AD	A/D data register 12	ADDR12	16	16	2, 3 PCLKB	2 ICLK	1405	

Table 5.1 List of I/O Registers (Address Order) (23/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 903Ah	S12AD	A/D data register 13	ADDR13	16	16	2, 3 PCLKB	2 ICLK	S12ADa	1405
0008 903Ch	S12AD	A/D data register 14	ADDR14	16	16	2, 3 PCLKB	2 ICLK		1405
0008 903Eh	S12AD	A/D data register 15	ADDR15	16	16	2, 3 PCLKB	2 ICLK		1405
0008 9040h	S12AD	A/D data register 16	ADDR16	16	16	2, 3 PCLKB	2 ICLK		1405
0008 9042h	S12AD	A/D data register 17	ADDR17	16	16	2, 3 PCLKB	2 ICLK		1405
0008 9044h	S12AD	A/D data register 18	ADDR18	16	16	2, 3 PCLKB	2 ICLK		1405
0008 9046h	S12AD	A/D data register 19	ADDR19	16	16	2, 3 PCLKB	2 ICLK		1405
0008 9048h	S12AD	A/D data register 20	ADDR20	16	16	2, 3 PCLKB	2 ICLK		1405
0008 9060h	S12AD	A/D sampling state register 01	ADSSTR01	16	16	2, 3 PCLKB	2 ICLK		1406
0008 9070h	S12AD	A/D sampling state register 23	ADSSTR23	16	16	2, 3 PCLKB	2 ICLK		1407
0008 9800h	AD	A/D data register A	ADDRA	16	16	2, 3 PCLKB	2 ICLK	ADb	1425
0008 9802h	AD	A/D data register B	ADDRB	16	16	2, 3 PCLKB	2 ICLK		1425
0008 9804h	AD	A/D data register C	ADDRC	16	16	2, 3 PCLKB	2 ICLK		1425
0008 9806h	AD	A/D data register D	ADDRD	16	16	2, 3 PCLKB	2 ICLK		1425
0008 9808h	AD	A/D data register E	ADDRE	16	16	2, 3 PCLKB	2 ICLK		1425
0008 980Ah	AD	A/D data register F	ADDRF	16	16	2, 3 PCLKB	2 ICLK		1425
0008 980Ch	AD	A/D data register G	ADDRG	16	16	2, 3 PCLKB	2 ICLK		1425
0008 980Eh	AD	A/D data register H	ADDRH	16	16	2, 3 PCLKB	2 ICLK		1425
0008 9810h	AD	A/D control/status register	ADCSR	8	8	2, 3 PCLKB	2 ICLK		1426
0008 9811h	AD	A/D control register	ADCR	8	8	2, 3 PCLKB	2 ICLK		1427
0008 9812h	AD	A/D control register 2	ADCR2	8	8	2, 3 PCLKB	2 ICLK		1428
0008 9813h	AD	A/D sampling state register	ADSSTR	8	8	2, 3 PCLKB	2 ICLK		1428
0008 981Fh	AD	A/D self-diagnostic register	ADDIAGR	8	8	2, 3 PCLKB	2 ICLK		1429
0008 A000h	SCI0	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK		SC1c, SC1d
0008 A001h	SCI0	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	1048	
0008 A002h	SCI0	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	1039	
0008 A003h	SCI0	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	1035	
0008 A004h	SCI0	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	1043	
0008 A005h	SCI0	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	1035	
0008 A006h	SCI0	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	1047	
0008 A007h	SCI0	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	1056	
0008 A008h	SCI0	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	1058	
0008 A009h	SCI0	I <sup>2</sup> C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	1059	
0008 A00Ah	SCI0	I <sup>2</sup> C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	1060	
0008 A00Bh	SCI0	I <sup>2</sup> C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	1061	
0008 A00Ch	SCI0	I <sup>2</sup> C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	1063	
0008 A00Dh	SCI0	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	1064	
0008 A020h	SCI1	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	1036	
0008 A021h	SCI1	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	1048	
0008 A022h	SCI1	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	1039	
0008 A023h	SCI1	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	1035	
0008 A024h	SCI1	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	1043	
0008 A025h	SCI1	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	1035	
0008 A026h	SCI1	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	1047	
0008 A027h	SCI1	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	1056	
0008 A028h	SCI1	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	1058	
0008 A029h	SCI1	I <sup>2</sup> C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	1059	
0008 A02Ah	SCI1	I <sup>2</sup> C bus mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	1060	
0008 A02Bh	SCI1	I <sup>2</sup> C bus mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	1061	
0008 A02Ch	SCI1	I <sup>2</sup> C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	1063	

Table 5.1 List of I/O Registers (Address Order) (24/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 A02Dh	SCI1	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	SCic, SCId	1064
0008 A040h	SCI2	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK		1036
0008 A041h	SCI2	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK		1048
0008 A042h	SCI2	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK		1039
0008 A043h	SCI2	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK		1035
0008 A044h	SCI2	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK		1043
0008 A045h	SCI2	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK		1035
0008 A046h	SCI2	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK		1047
0008 A047h	SCI2	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK		1056
0008 A048h	SCI2	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK		1058
0008 A049h	SCI2	I <sup>2</sup> C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK		1059
0008 A04Ah	SCI2	I <sup>2</sup> C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK		1060
0008 A04Bh	SCI2	I <sup>2</sup> C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK		1061
0008 A04Ch	SCI2	I <sup>2</sup> C status register	SISR	8	8	2, 3 PCLKB	2 ICLK		1063
0008 A04Dh	SCI2	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK		1064
0008 A060h	SCI3	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	1036	
0008 A061h	SCI3	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	1048	
0008 A062h	SCI3	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	1039	
0008 A063h	SCI3	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	1035	
0008 A064h	SCI3	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	1043	
0008 A065h	SCI3	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	1035	
0008 A066h	SCI3	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	1047	
0008 A067h	SCI3	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	1056	
0008 A068h	SCI3	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	1058	
0008 A069h	SCI3	I <sup>2</sup> C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	1059	
0008 A06Ah	SCI3	I <sup>2</sup> C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	1060	
0008 A06Bh	SCI3	I <sup>2</sup> C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	1061	
0008 A06Ch	SCI3	I <sup>2</sup> C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	1063	
0008 A06Dh	SCI3	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	1064	
0008 A080h	SCI4	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	1036	
0008 A081h	SCI4	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	1048	
0008 A082h	SCI4	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	1039	
0008 A083h	SCI4	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	1035	
0008 A084h	SCI4	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	1043	
0008 A085h	SCI4	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	1035	
0008 A086h	SCI4	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	1047	
0008 A087h	SCI4	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	1056	
0008 A088h	SCI4	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	1058	
0008 A089h	SCI4	I <sup>2</sup> C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	1059	
0008 A08Ah	SCI4	I <sup>2</sup> C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	1060	
0008 A08Bh	SCI4	I <sup>2</sup> C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	1061	
0008 A08Ch	SCI4	I <sup>2</sup> C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	1063	
0008 A08Dh	SCI4	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	1064	
0008 A0A0h	SCI5	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	1036	
0008 A0A1h	SCI5	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	1048	
0008 A0A2h	SCI5	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	1039	
0008 A0A3h	SCI5	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	1035	
0008 A0A4h	SCI5	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	1043	
0008 A0A5h	SCI5	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	1035	
0008 A0A6h	SCI5	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	1047	

Table 5.1 List of I/O Registers (Address Order) (25/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 A0A7h	SCI5	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	SCiC, SCiD	1056
0008 A0A8h	SCI5	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK		1058
0008 A0A9h	SCI5	I <sup>2</sup> C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK		1059
0008 A0AAh	SCI5	I <sup>2</sup> C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK		1060
0008 A0ABh	SCI5	I <sup>2</sup> C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK		1061
0008 A0ACh	SCI5	I <sup>2</sup> C status register	SISR	8	8	2, 3 PCLKB	2 ICLK		1063
0008 A0ADh	SCI5	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK		1064
0008 A0C0h	SCI6	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK		1036
0008 A0C1h	SCI6	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK		1048
0008 A0C2h	SCI6	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK		1039
0008 A0C3h	SCI6	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK		1035
0008 A0C4h	SCI6	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK		1043
0008 A0C5h	SCI6	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK		1035
0008 A0C6h	SCI6	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK		1047
0008 A0C7h	SCI6	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK		1056
0008 A0C8h	SCI6	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	1058	
0008 A0C9h	SCI6	I <sup>2</sup> C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	1059	
0008 A0CAh	SCI6	I <sup>2</sup> C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	1060	
0008 A0CBh	SCI6	I <sup>2</sup> C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	1061	
0008 A0CCh	SCI6	I <sup>2</sup> C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	1063	
0008 A0CDh	SCI6	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	1064	
0008 A0E0h	SCI7	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	1036	
0008 A0E1h	SCI7	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	1048	
0008 A0E2h	SCI7	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	1039	
0008 A0E3h	SCI7	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	1035	
0008 A0E4h	SCI7	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	1043	
0008 A0E5h	SCI7	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	1035	
0008 A0E6h	SCI7	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	1047	
0008 A0E7h	SCI7	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	1056	
0008 A0E8h	SCI7	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	1058	
0008 A0E9h	SCI7	I <sup>2</sup> C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	1059	
0008 A0EAh	SCI7	I <sup>2</sup> C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	1060	
0008 A0EBh	SCI7	I <sup>2</sup> C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	1061	
0008 A0ECh	SCI7	I <sup>2</sup> C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	1063	
0008 A0EDh	SCI7	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	1064	
0008 A100h	SCI8	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	1036	
0008 A101h	SCI8	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	1048	
0008 A102h	SCI8	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	1039	
0008 A103h	SCI8	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	1035	
0008 A104h	SCI8	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	1043	
0008 A105h	SCI8	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	1035	
0008 A106h	SCI8	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	1047	
0008 A107h	SCI8	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	1056	
0008 A108h	SCI8	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	1058	
0008 A109h	SCI8	I <sup>2</sup> C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	1059	
0008 A10Ah	SCI8	I <sup>2</sup> C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	1060	
0008 A10Bh	SCI8	I <sup>2</sup> C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	1061	
0008 A10Ch	SCI8	I <sup>2</sup> C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	1063	
0008 A10Dh	SCI8	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	1064	
0008 A120h	SCI9	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	1036	



Table 5.1 List of I/O Registers (Address Order) (26/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 A121h	SCI9	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	SCic, SCId	1048
0008 A122h	SCI9	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK		1039
0008 A123h	SCI9	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK		1035
0008 A124h	SCI9	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK		1043
0008 A125h	SCI9	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK		1035
0008 A126h	SCI9	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK		1047
0008 A127h	SCI9	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK		1056
0008 A128h	SCI9	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK		1058
0008 A129h	SCI9	I <sup>2</sup> C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK		1059
0008 A12Ah	SCI9	I <sup>2</sup> C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK		1060
0008 A12Bh	SCI9	I <sup>2</sup> C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK		1061
0008 A12Ch	SCI9	I <sup>2</sup> C status register	SISR	8	8	2, 3 PCLKB	2 ICLK		1063
0008 A12Dh	SCI9	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK		1064
0008 A140h	SCI10	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK		1036
0008 A141h	SCI10	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	1048	
0008 A142h	SCI10	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	1039	
0008 A143h	SCI10	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	1035	
0008 A144h	SCI10	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	1043	
0008 A145h	SCI10	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	1035	
0008 A146h	SCI10	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	1047	
0008 A147h	SCI10	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	1056	
0008 A148h	SCI10	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	1058	
0008 A149h	SCI10	I <sup>2</sup> C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	1059	
0008 A14Ah	SCI10	I <sup>2</sup> C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	1060	
0008 A14Bh	SCI10	I <sup>2</sup> C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	1061	
0008 A14Ch	SCI10	I <sup>2</sup> C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	1063	
0008 A14Dh	SCI10	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	1064	
0008 A160h	SCI11	Serial mode register	SMR	8	8	2, 3 PCLKB	2 ICLK	1036	
0008 A161h	SCI11	Bit rate register	BRR	8	8	2, 3 PCLKB	2 ICLK	1048	
0008 A162h	SCI11	Serial control register	SCR	8	8	2, 3 PCLKB	2 ICLK	1039	
0008 A163h	SCI11	Transmit data register	TDR	8	8	2, 3 PCLKB	2 ICLK	1035	
0008 A164h	SCI11	Serial status register	SSR	8	8	2, 3 PCLKB	2 ICLK	1043	
0008 A165h	SCI11	Receive data register	RDR	8	8	2, 3 PCLKB	2 ICLK	1035	
0008 A166h	SCI11	Smart card mode register	SCMR	8	8	2, 3 PCLKB	2 ICLK	1047	
0008 A167h	SCI11	Serial extended mode register	SEMR	8	8	2, 3 PCLKB	2 ICLK	1056	
0008 A168h	SCI11	Noise filter setting register	SNFR	8	8	2, 3 PCLKB	2 ICLK	1058	
0008 A169h	SCI11	I <sup>2</sup> C mode register 1	SIMR1	8	8	2, 3 PCLKB	2 ICLK	1059	
0008 A16Ah	SCI11	I <sup>2</sup> C mode register 2	SIMR2	8	8	2, 3 PCLKB	2 ICLK	1060	
0008 A16Bh	SCI11	I <sup>2</sup> C mode register 3	SIMR3	8	8	2, 3 PCLKB	2 ICLK	1061	
0008 A16Ch	SCI11	I <sup>2</sup> C status register	SISR	8	8	2, 3 PCLKB	2 ICLK	1063	
0008 A16Dh	SCI11	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	1064	
0008 A800h	IEB	IEBus control register	IECTR	8	8	3, 4 PCLKB	2, 3 ICLK	IEB	1349
0008 A801h	IEB	IEBus command register	IECMR	8	8	3, 4 PCLKB	2, 3 ICLK		1350
0008 A802h	IEB	IEBus master control register	IEMCR	8	8	3, 4 PCLKB	2, 3 ICLK		1351
0008 A803h	IEB	IEBus master unit address register 1	IEAR1	8	8	3, 4 PCLKB	2, 3 ICLK		1352
0008 A804h	IEB	IEBus master unit address register 2	IEAR2	8	8	3, 4 PCLKB	2, 3 ICLK		1352
0008 A805h	IEB	IEBus slave address setting register 1	IESA1	8	8	3, 4 PCLKB	2, 3 ICLK		1353
0008 A806h	IEB	IEBus slave address setting register 2	IESA2	8	8	3, 4 PCLKB	2, 3 ICLK		1353
0008 A807h	IEB	IEBus transmit message length register	IETBFL	8	8	3, 4 PCLKB	2, 3 ICLK		1354
0008 A809h	IEB	IEBus reception master address register 1	IEMA1	8	8	3, 4 PCLKB	2, 3 ICLK		1354

Table 5.1 List of I/O Registers (Address Order) (27/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 A80Ah	IEB	IEBus reception master address register 2	IEMA2	8	8	3, 4 PCLKB	2, 3 ICLK	IEB	1355
0008 A80Bh	IEB	IEBus receive control field register	IERCTL	8	8	3, 4 PCLKB	2, 3 ICLK		1355
0008 A80Ch	IEB	IEBus receive message length register	IERBFL	8	8	3, 4 PCLKB	2, 3 ICLK		1356
0008 A80Eh	IEB	IEBus lock address register 1	IELA1	8	8	3, 4 PCLKB	2, 3 ICLK		1356
0008 A80Fh	IEB	IEBus lock address register 2	IELA2	8	8	3, 4 PCLKB	2, 3 ICLK		1356
0008 A810h	IEB	IEBus general flag register	IEFLG	8	8	3, 4 PCLKB	2, 3 ICLK		1357
0008 A811h	IEB	IEBus transmit status register	IETSR	8	8	3, 4 PCLKB	2, 3 ICLK		1359
0008 A812h	IEB	IEBus transmit interrupt enable register	IEIET	8	8	3, 4 PCLKB	2, 3 ICLK		1361
0008 A814h	IEB	IEBus receive status register	IERSR	8	8	3, 4 PCLKB	2, 3 ICLK		1362
0008 A815h	IEB	IEBus receive interrupt enable register	IEIER	8	8	3, 4 PCLKB	2, 3 ICLK		1365
0008 A818h	IEB	IEBus clock select register	IECKSR	8	8	3, 4 PCLKB	2, 3 ICLK		1366
0008 A900h to 0008 A91Fh	IEB	IEBus transmit data buffer register 001 to 032	IETB001 to 032	8	8	3, 4 PCLKB	2, 3 ICLK		1368
0008 AA00h to 0008 AA1Fh	IEB	IEBus receive data buffer register 001 to 032	IERB001 to 032	8	8	3, 4 PCLKB	2, 3 ICLK		1368
0008 B300h	SCI12	Serial mode register	SMR12	8	8	3, 4 PCLKB	2, 3 ICLK		SClC, SCId
0008 B301h	SCI12	Bit rate register	BRR12	8	8	3, 4 PCLKB	2, 3 ICLK	1048	
0008 B302h	SCI12	Serial control register	SCR12	8	8	2, 3 PCLKB	2 ICLK	1039	
0008 B303h	SCI12	Transmit data register	TDR12	8	8	2, 3 PCLKB	2 ICLK	1036	
0008 B304h	SCI12	Serial status register	SSR12	8	8	2, 3 PCLKB	2 ICLK	1043	
0008 B305h	SCI12	Receive data register	RDR12	8	8	2, 3 PCLKB	2 ICLK	1035	
0008 B306h	SCI12	Smart card mode register	SCMR12	8	8	2, 3 PCLKB	2 ICLK	1047	
0008 B307h	SCI12	Serial extended mode register	SEMR12	8	8	2, 3 PCLKB	2 ICLK	1056	
0008 B308h	SCI12	Noise filter setting register	SNFR12	8	8	2, 3 PCLKB	2 ICLK	1058	
0008 B309h	SCI12	I <sup>2</sup> C mode register 1	SIMR112	8	8	2, 3 PCLKB	2 ICLK	1059	
0008 B30Ah	SCI12	I <sup>2</sup> C mode register 2	SIMR212	8	8	2, 3 PCLKB	2 ICLK	1060	
0008 B30Bh	SCI12	I <sup>2</sup> C mode register 3	SIMR312	8	8	2, 3 PCLKB	2 ICLK	1061	
0008 B30Ch	SCI12	I <sup>2</sup> C status register	SIS12	8	8	2, 3 PCLKB	2 ICLK	1063	
0008 B30Dh	SCI12	SPI mode register	SPMR	8	8	2, 3 PCLKB	2 ICLK	1064	
0008 B320h	SCI12	Extended serial module enable register	ESMER	8	8	2, 3 PCLKB	2 ICLK	1065	
0008 B321h	SCI12	Control register 0	CR0	8	8	2, 3 PCLKB	2 ICLK	1066	
0008 B322h	SCI12	Control register 1	CR1	8	8	2, 3 PCLKB	2 ICLK	1067	
0008 B323h	SCI12	Control register 2	CR2	8	8	2, 3 PCLKB	2 ICLK	1068	
0008 B324h	SCI12	Control register 3	CR3	8	8	2, 3 PCLKB	2 ICLK	1069	
0008 B325h	SCI12	Port control register	PCR	8	8	2, 3 PCLKB	2 ICLK	1069	
0008 B326h	SCI12	Interrupt control register	ICR	8	8	2, 3 PCLKB	2 ICLK	1070	
0008 B327h	SCI12	Status register	STR	8	8	2, 3 PCLKB	2 ICLK	1071	
0008 B328h	SCI12	Status clear register	STCR	8	8	2, 3 PCLKB	2 ICLK	1072	
0008 B329h	SCI12	Control field 0 data register	CF0DR	8	8	2, 3 PCLKB	2 ICLK	1072	
0008 B32Ah	SCI12	Control field 0 compare enable register	CF0CR	8	8	2, 3 PCLKB	2 ICLK	1073	
0008 B32Bh	SCI12	Control field 0 receive data register	CF0RR	8	8	2, 3 PCLKB	2 ICLK	1073	
0008 B32Ch	SCI12	Primary control field 1 data register	PCF1DR	8	8	2, 3 PCLKB	2 ICLK	1073	
0008 B32Dh	SCI12	Secondary control field 1 data register	SCF1DR	8	8	2, 3 PCLKB	2 ICLK	1074	
0008 B32Eh	SCI12	Control field 1 compare enable register	CF1CR	8	8	2, 3 PCLKB	2 ICLK	1074	
0008 B32Fh	SCI12	Control field 1 receive data register	CF1RR	8	8	2, 3 PCLKB	2 ICLK	1074	
0008 B330h	SCI12	Timer control register	TCR	8	8	2, 3 PCLKB	2 ICLK	1075	
0008 B331h	SCI12	Timer mode register	TMR	8	8	2, 3 PCLKB	2 ICLK	1075	
0008 B332h	SCI12	Timer prescaler register	TPRE	8	8	2, 3 PCLKB	2 ICLK	1076	
0008 B333h	SCI12	Timer count register	TCNT	8	8	2, 3 PCLKB	2 ICLK	1076	
0008 C000h	PORT0	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	509
0008 C001h	PORT1	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK		509

Table 5.1 List of I/O Registers (Address Order) (28/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 C002h	PORT2	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	509
0008 C003h	PORT3	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK		509
0008 C004h	PORT4	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK		509
0008 C005h	PORT5	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK		509
0008 C006h	PORT6	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK		509
0008 C007h	PORT7	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK		509
0008 C008h	PORT8	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK		509
0008 C009h	PORT9	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK		509
0008 C00Ah	PORTA	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK		509
0008 C00Bh	PORTB	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK		509
0008 C00Ch	PORTC	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK		509
0008 C00Dh	PORTD	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK		509
0008 C00Eh	PORTE	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK		509
0008 C00Fh	PORTF	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK		509
0008 C010h	PORTG	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK		509
0008 C011h	PORTH	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK		509
0008 C012h	PORTJ	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK		509
0008 C013h	PORTK	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK		509
0008 C014h	PORTL	Port direction register	PDR	8	8	2, 3 PCLKB	2 ICLK		509
0008 C020h	PORT0	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C021h	PORT1	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C022h	PORT2	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C023h	PORT3	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C024h	PORT4	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C025h	PORT5	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C026h	PORT6	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C027h	PORT7	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C028h	PORT8	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C029h	PORT9	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C02Ah	PORTA	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C02Bh	PORTB	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C02Ch	PORTC	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C02Dh	PORTD	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C02Eh	PORTE	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C02Fh	PORTF	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C030h	PORTG	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C031h	PORTH	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C032h	PORTJ	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C033h	PORTK	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C034h	PORTL	Port output data register	PODR	8	8	2, 3 PCLKB	2 ICLK	510	
0008 C040h	PORT0	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	511	
0008 C041h	PORT1	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	511	
0008 C042h	PORT2	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	511	
0008 C043h	PORT3	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	511	
0008 C044h	PORT4	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	511	
0008 C045h	PORT5	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	511	
0008 C046h	PORT6	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	511	
0008 C047h	PORT7	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	511	
0008 C048h	PORT8	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	511	
0008 C049h	PORT9	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	511	

Table 5.1 List of I/O Registers (Address Order) (29/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 C04Ah	PORTA	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	511
0008 C04Bh	PORTB	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK		511
0008 C04Ch	PORTC	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK		511
0008 C04Dh	PORTD	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK		511
0008 C04Eh	PORTE	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK		511
0008 C04Fh	PORTF	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK		511
0008 C050h	PORTG	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK		511
0008 C051h	PORTH	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK		511
0008 C052h	PORTJ	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK		511
0008 C053h	PORTK	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK		511
0008 C054h	PORTL	Port input data register	PIDR	8	8	2, 3 PCLKB	2 ICLK		511
0008 C060h	PORT0	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK		512
0008 C061h	PORT1	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK		512
0008 C062h	PORT2	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK		512
0008 C063h	PORT3	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK		512
0008 C064h	PORT4	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	512	
0008 C065h	PORT5	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	512	
0008 C066h	PORT6	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	512	
0008 C067h	PORT7	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	512	
0008 C068h	PORT8	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	512	
0008 C069h	PORT9	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	512	
0008 C06Ah	PORTA	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	512	
0008 C06Bh	PORTB	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	512	
0008 C06Ch	PORTC	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	512	
0008 C06Dh	PORTD	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	512	
0008 C06Eh	PORTE	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	512	
0008 C06Fh	PORTF	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	512	
0008 C070h	PORTG	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	512	
0008 C071h	PORTH	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	512	
0008 C072h	PORTJ	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	512	
0008 C073h	PORTK	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	512	
0008 C074h	PORTL	Port mode register	PMR	8	8	2, 3 PCLKB	2 ICLK	512	
0008 C080h	PORT0	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	513	
0008 C081h	PORT0	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	514	
0008 C082h	PORT1	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	513	
0008 C083h	PORT1	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	514	
0008 C084h	PORT2	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	513	
0008 C085h	PORT2	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	514	
0008 C086h	PORT3	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	513	
0008 C087h	PORT3	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	514	
0008 C088h	PORT4	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	513	
0008 C089h	PORT4	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	513	
0008 C08Ah	PORT5	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	514	
0008 C08Bh	PORT5	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	513	
0008 C08Ch	PORT6	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	514	
0008 C08Dh	PORT6	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	513	
0008 C08Eh	PORT7	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	514	
0008 C08Fh	PORT7	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	513	
0008 C090h	PORT8	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	514	
0008 C091h	PORT8	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK	513	

Table 5.1 List of I/O Registers (Address Order) (30/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 C092h	PORT9	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK	I/O Ports	514
0008 C093h	PORT9	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK		513
0008 C094h	PORTA	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK		514
0008 C095h	PORTA	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK		513
0008 C096h	PORTB	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK		514
0008 C097h	PORTB	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK		513
0008 C098h	PORTC	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK		514
0008 C099h	PORTC	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK		513
0008 C09Ah	PORTD	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK		514
0008 C09Bh	PORTD	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK		513
0008 C09Ch	PORTE	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK		514
0008 C09Dh	PORTE	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK		513
0008 C09Eh	PORTF	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK		514
0008 C09Fh	PORTF	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK		513
0008 C0A0h	PORTG	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK		514
0008 C0A1h	PORTG	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK		513
0008 C0A3h	PORTH	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK		514
0008 C0A4h	PORTJ	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK		513
0008 C0A5h	PORTJ	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK		514
0008 C0A6h	PORTK	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK		513
0008 C0A7h	PORTK	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK		514
0008 C0A8h	PORTL	Open drain control register 0	ODR0	8	8, 16	2, 3 PCLKB	2 ICLK		513
0008 C0A9h	PORTL	Open drain control register 1	ODR1	8	8, 16	2, 3 PCLKB	2 ICLK		514
0008 C0C0h	PORT0	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK		515
0008 C0C1h	PORT1	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK		515
0008 C0C2h	PORT2	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK		515
0008 C0C3h	PORT3	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK		515
0008 C0C4h	PORT4	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK		515
0008 C0C5h	PORT5	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK		515
0008 C0C6h	PORT6	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK		515
0008 C0C7h	PORT7	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	515	
0008 C0C8h	PORT8	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	515	
0008 C0C9h	PORT9	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	515	
0008 C0CAh	PORTA	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	515	
0008 C0CBh	PORTB	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	515	
0008 C0CCh	PORTC	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	515	
0008 C0CDh	PORTD	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	515	
0008 C0CEh	PORTE	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	515	
0008 C0CFh	PORTF	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	515	
0008 C0D0h	PORTG	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	515	
0008 C0D1h	PORTH	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	515	
0008 C0D2h	PORTJ	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	515	
0008 C0D3h	PORTK	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	515	
0008 C0D4h	PORTL	Pull-up control register	PCR	8	8	2, 3 PCLKB	2 ICLK	515	
0008 C0E0h	PORT0	Driving ability control register	DSCR	8	8	2, 3 PCLKB	2 ICLK	516	
0008 C0E2h	PORT2	Driving ability control register	DSCR	8	8	2, 3 PCLKB	2 ICLK	516	
0008 C0E5h	PORT5	Driving ability control register	DSCR	8	8	2, 3 PCLKB	2 ICLK	516	
0008 C0E6h	PORT6	Driving ability control register	DSCR	8	8	2, 3 PCLKB	2 ICLK	516	
0008 C0E7h	PORT7	Driving ability control register	DSCR	8	8	2, 3 PCLKB	2 ICLK	516	
0008 C0E9h	PORT9	Driving ability control register	DSCR	8	8	2, 3 PCLKB	2 ICLK	516	

Table 5.1 List of I/O Registers (Address Order) (31/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page	
						ICLK ≥ PCLK	ICLK < PCLK			
0008 C0EAh	PORTA	Driving ability control register	DSCR	8	8	2, 3 PCLKB	2 ICLK	I/O Ports	516	
0008 C0EBh	PORTB	Driving ability control register	DSCR	8	8	2, 3 PCLKB	2 ICLK		516	
0008 C0ECh	PORTC	Driving ability control register	DSCR	8	8	2, 3 PCLKB	2 ICLK		516	
0008 C0EDh	PORTD	Driving ability control register	DSCR	8	8	2, 3 PCLKB	2 ICLK		516	
0008 C0EEh	PORTE	Driving ability control register	DSCR	8	8	2, 3 PCLKB	2 ICLK		516	
0008 C0F0h	PORTG	Driving ability control register	DSCR	8	8	2, 3 PCLKB	2 ICLK		516	
0008 C100h	MPC	CS output enable register	PFCSE	8	8	2, 3 PCLKB	2 ICLK	MPC	557	
0008 C102h	MPC	CS output pin select register 0	PFCSS0	8	8	2, 3 PCLKB	2 ICLK		558	
0008 C103h	MPC	CS output pin select register 1	PFCSS1	8	8	2, 3 PCLKB	2 ICLK		559	
0008 C104h	MPC	Address output enable register 0	PFAOE0	8	8, 16	2, 3 PCLKB	2 ICLK		560	
0008 C105h	MPC	Address output enable register 1	PFAOE1	8	8, 16	2, 3 PCLKB	2 ICLK		560	
0008 C106h	MPC	External bus control register 0	PFBCR0	8	8, 16	2, 3 PCLKB	2 ICLK		561	
0008 C107h	MPC	External bus control register 1	PFBCR1	8	8, 16	2, 3 PCLKB	2 ICLK		561	
0008 C114h	MPC	USB0 control register	PFUSB0	8	8	2, 3 PCLKB	2 ICLK		562	
0008 C11Fh	MPC	Write-protect register	PWPR	8	8	2, 3 PCLKB	2 ICLK		533	
0008 C140h	MPC	P00 pin function control register	P00PFS	8	8	2, 3 PCLKB	2 ICLK		534	
0008 C141h	MPC	P01 pin function control register	P01PFS	8	8	2, 3 PCLKB	2 ICLK		534	
0008 C142h	MPC	P02 pin function control register	P02PFS	8	8	2, 3 PCLKB	2 ICLK		534	
0008 C143h	MPC	P03 pin function control register	P03PFS	8	8	2, 3 PCLKB	2 ICLK		534	
0008 C145h	MPC	P05 pin function control register	P05PFS	8	8	2, 3 PCLKB	2 ICLK		534	
0008 C147h	MPC	P07 pin function control register	P07PFS	8	8	2, 3 PCLKB	2 ICLK		534	
0008 C148h	MPC	P10 pin function control register	P10PFS	8	8	2, 3 PCLKB	2 ICLK		535	
0008 C149h	MPC	P11 pin function control register	P11PFS	8	8	2, 3 PCLKB	2 ICLK		535	
0008 C14Ah	MPC	P12 pin function control register	P12PFS	8	8	2, 3 PCLKB	2 ICLK		535	
0008 C14Bh	MPC	P13 pin function control register	P13PFS	8	8	2, 3 PCLKB	2 ICLK		535	
0008 C14Ch	MPC	P14 pin function control register	P14PFS	8	8	2, 3 PCLKB	2 ICLK		535	
0008 C14Dh	MPC	P15 pin function control register	P15PFS	8	8	2, 3 PCLKB	2 ICLK		535	
0008 C14Eh	MPC	P16 pin function control register	P16PFS	8	8	2, 3 PCLKB	2 ICLK		535	
0008 C14Fh	MPC	P17 pin function control register	P17PFS	8	8	2, 3 PCLKB	2 ICLK		535	
0008 C150h	MPC	P20 pin function control register	P20PFS	8	8	2, 3 PCLKB	2 ICLK		537	
0008 C151h	MPC	P21 pin function control register	P21PFS	8	8	2, 3 PCLKB	2 ICLK		537	
0008 C152h	MPC	P22 pin function control register	P22PFS	8	8	2, 3 PCLKB	2 ICLK		537	
0008 C153h	MPC	P23 pin function control register	P23PFS	8	8	2, 3 PCLKB	2 ICLK		537	
0008 C154h	MPC	P24 pin function control register	P24PFS	8	8	2, 3 PCLKB	2 ICLK		537	
0008 C155h	MPC	P25 pin function control register	P25PFS	8	8	2, 3 PCLKB	2 ICLK		537	
0008 C156h	MPC	P26 pin function control register	P26PFS	8	8	2, 3 PCLKB	2 ICLK		537	
0008 C157h	MPC	P27 pin function control register	P27PFS	8	8	2, 3 PCLKB	2 ICLK		537	
0008 C158h	MPC	P30 pin function control register	P30PFS	8	8	2, 3 PCLKB	2 ICLK		539	
0008 C159h	MPC	P31 pin function control register	P31PFS	8	8	2, 3 PCLKB	2 ICLK		539	
0008 C15Ah	MPC	P32 pin function control register	P32PFS	8	8	2, 3 PCLKB	2 ICLK		539	
0008 C15Bh	MPC	P33 pin function control register	P33PFS	8	8	2, 3 PCLKB	2 ICLK		539	
0008 C15Ch	MPC	P34 pin function control register	P34PFS	8	8	2, 3 PCLKB	2 ICLK		MPC	539
0008 C160h	MPC	P40 pin function control register	P40PFS	8	8	2, 3 PCLKB	2 ICLK			540
0008 C161h	MPC	P41 pin function control register	P41PFS	8	8	2, 3 PCLKB	2 ICLK	540		
0008 C162h	MPC	P42 pin function control register	P42PFS	8	8	2, 3 PCLKB	2 ICLK	540		
0008 C163h	MPC	P43 pin function control register	P43PFS	8	8	2, 3 PCLKB	2 ICLK	540		
0008 C164h	MPC	P44 pin function control register	P44PFS	8	8	2, 3 PCLKB	2 ICLK	540		
0008 C165h	MPC	P45 pin function control register	P45PFS	8	8	2, 3 PCLKB	2 ICLK	540		
0008 C166h	MPC	P46 pin function control register	P46PFS	8	8	2, 3 PCLKB	2 ICLK	540		
0008 C167h	MPC	P47 pin function control register	P47PFS	8	8	2, 3 PCLKB	2 ICLK	540		

Table 5.1 List of I/O Registers (Address Order) (32/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 C168h	MPC	P50 pin function control register	P50PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	541
0008 C169h	MPC	P51 pin function control register	P51PFS	8	8	2, 3 PCLKB	2 ICLK		541
0008 C16Ah	MPC	P52 pin function control register	P52PFS	8	8	2, 3 PCLKB	2 ICLK		541
0008 C16Ch	MPC	P54 pin function control register	P54PFS	8	8	2, 3 PCLKB	2 ICLK		541
0008 C16Dh	MPC	P55 pin function control register	P55PFS	8	8	2, 3 PCLKB	2 ICLK		541
0008 C16Eh	MPC	P56 pin function control register	P56PFS	8	8	2, 3 PCLKB	2 ICLK		541
0008 C170h	MPC	P60 pin function control register	P60PFS	8	8	2, 3 PCLKB	2 ICLK		542
0008 C171h	MPC	P61 pin function control register	P61PFS	8	8	2, 3 PCLKB	2 ICLK		542
0008 C176h	MPC	P66 pin function control register	P66PFS	8	8	2, 3 PCLKB	2 ICLK		542
0008 C177h	MPC	P67 pin function control register	P67PFS	8	8	2, 3 PCLKB	2 ICLK		542
0008 C178h	MPC	P70 pin function control register	P70PFS	8	8	2, 3 PCLKB	2 ICLK		543
0008 C17Bh	MPC	P73 pin function control register	P73PFS	8	8	2, 3 PCLKB	2 ICLK		543
0008 C17Ch	MPC	P74 pin function control register	P74PFS	8	8	2, 3 PCLKB	2 ICLK		543
0008 C17Dh	MPC	P75 pin function control register	P75PFS	8	8	2, 3 PCLKB	2 ICLK		543
0008 C17Eh	MPC	P76 pin function control register	P76PFS	8	8	2, 3 PCLKB	2 ICLK		543
0008 C17Fh	MPC	P77 pin function control register	P77PFS	8	8	2, 3 PCLKB	2 ICLK		543
0008 C180h	MPC	P80 pin function control register	P80PFS	8	8	2, 3 PCLKB	2 ICLK		544
0008 C181h	MPC	P81 pin function control register	P81PFS	8	8	2, 3 PCLKB	2 ICLK		544
0008 C182h	MPC	P82 pin function control register	P82PFS	8	8	2, 3 PCLKB	2 ICLK		544
0008 C183h	MPC	P83 pin function control register	P83PFS	8	8	2, 3 PCLKB	2 ICLK		544
0008 C186h	MPC	P86 pin function control register	P86PFS	8	8	2, 3 PCLKB	2 ICLK		544
0008 C187h	MPC	P87 pin function control register	P87PFS	8	8	2, 3 PCLKB	2 ICLK		544
0008 C188h	MPC	P90 pin function control register	P90PFS	8	8	2, 3 PCLKB	2 ICLK		545
0008 C189h	MPC	P91 pin function control register	P91PFS	8	8	2, 3 PCLKB	2 ICLK		545
0008 C18Ah	MPC	P92 pin function control register	P92PFS	8	8	2, 3 PCLKB	2 ICLK		545
0008 C18Bh	MPC	P93 pin function control register	P93PFS	8	8	2, 3 PCLKB	2 ICLK		545
0008 C190h	MPC	PA0 pin function control register	PA0PFS	8	8	2, 3 PCLKB	2 ICLK		546
0008 C191h	MPC	PA1 pin function control register	PA1PFS	8	8	2, 3 PCLKB	2 ICLK		546
0008 C192h	MPC	PA2 pin function control register	PA2PFS	8	8	2, 3 PCLKB	2 ICLK		546
0008 C193h	MPC	PA3 pin function control register	PA3PFS	8	8	2, 3 PCLKB	2 ICLK		546
0008 C194h	MPC	PA4 pin function control register	PA4PFS	8	8	2, 3 PCLKB	2 ICLK		546
0008 C195h	MPC	PA5 pin function control register	PA5PFS	8	8	2, 3 PCLKB	2 ICLK		546
0008 C196h	MPC	PA6 pin function control register	PA6PFS	8	8	2, 3 PCLKB	2 ICLK		546
0008 C197h	MPC	PA7 pin function control register	PA7PFS	8	8	2, 3 PCLKB	2 ICLK		546
0008 C198h	MPC	PB0 pin function control register	PB0PFS	8	8	2, 3 PCLKB	2 ICLK		548
0008 C199h	MPC	PB1 pin function control register	PB1PFS	8	8	2, 3 PCLKB	2 ICLK	548	
0008 C19Ah	MPC	PB2 pin function control register	PB2PFS	8	8	2, 3 PCLKB	2 ICLK	548	
0008 C19Bh	MPC	PB3 pin function control register	PB3PFS	8	8	2, 3 PCLKB	2 ICLK	548	
0008 C19Ch	MPC	PB4 pin function control register	PB4PFS	8	8	2, 3 PCLKB	2 ICLK	548	
0008 C19Dh	MPC	PB5 pin function control register	PB5PFS	8	8	2, 3 PCLKB	2 ICLK	548	
0008 C19Eh	MPC	PB6 pin function control register	PB6PFS	8	8	2, 3 PCLKB	2 ICLK	548	
0008 C19Fh	MPC	PB7 pin function control register	PB7PFS	8	8	2, 3 PCLKB	2 ICLK	548	
0008 C1A0h	MPC	PC0 pin function control register	PC0PFS	8	8	2, 3 PCLKB	2 ICLK	550	
0008 C1A1h	MPC	PC1 pin function control register	PC1PFS	8	8	2, 3 PCLKB	2 ICLK	550	
0008 C1A2h	MPC	PC2 pin function control register	PC2PFS	8	8	2, 3 PCLKB	2 ICLK	550	
0008 C1A3h	MPC	PC3 pin function control register	PC3PFS	8	8	2, 3 PCLKB	2 ICLK	550	
0008 C1A4h	MPC	PC4 pin function control register	PC4PFS	8	8	2, 3 PCLKB	2 ICLK	550	
0008 C1A5h	MPC	PC5 pin function control register	PC5PFS	8	8	2, 3 PCLKB	2 ICLK	550	
0008 C1A6h	MPC	PC6 pin function control register	PC6PFS	8	8	2, 3 PCLKB	2 ICLK	550	
0008 C1A7h	MPC	PC7 pin function control register	PC7PFS	8	8	2, 3 PCLKB	2 ICLK	550	

Table 5.1 List of I/O Registers (Address Order) (33/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0008 C1A8h	MPC	PD0 pin function control register	PD0PFS	8	8	2, 3 PCLKB	2 ICLK	MPC	552
0008 C1A9h	MPC	PD1 pin function control register	PD1PFS	8	8	2, 3 PCLKB	2 ICLK		552
0008 C1AAh	MPC	PD2 pin function control register	PD2PFS	8	8	2, 3 PCLKB	2 ICLK		552
0008 C1ABh	MPC	PD3 pin function control register	PD3PFS	8	8	2, 3 PCLKB	2 ICLK		552
0008 C1ACh	MPC	PD4 pin function control register	PD4PFS	8	8	2, 3 PCLKB	2 ICLK		552
0008 C1ADh	MPC	PD5 pin function control register	PD5PFS	8	8	2, 3 PCLKB	2 ICLK		552
0008 C1AEh	MPC	PD6 pin function control register	PD6PFS	8	8	2, 3 PCLKB	2 ICLK		552
0008 C1AFh	MPC	PD7 pin function control register	PD7PFS	8	8	2, 3 PCLKB	2 ICLK		552
0008 C1B0h	MPC	PE0 pin function control register	PE0PFS	8	8	2, 3 PCLKB	2 ICLK		553
0008 C1B1h	MPC	PE1 pin function control register	PE1PFS	8	8	2, 3 PCLKB	2 ICLK		553
0008 C1B2h	MPC	PE2 pin function control register	PE2PFS	8	8	2, 3 PCLKB	2 ICLK		553
0008 C1B3h	MPC	PE3 pin function control register	PE3PFS	8	8	2, 3 PCLKB	2 ICLK		553
0008 C1B4h	MPC	PE4 pin function control register	PE4PFS	8	8	2, 3 PCLKB	2 ICLK		553
0008 C1B5h	MPC	PE5 pin function control register	PE5PFS	8	8	2, 3 PCLKB	2 ICLK		553
0008 C1B6h	MPC	PE6 pin function control register	PE6PFS	8	8	2, 3 PCLKB	2 ICLK		553
0008 C1B7h	MPC	PE7 pin function control register	PE7PFS	8	8	2, 3 PCLKB	2 ICLK		553
0008 C1B8h	MPC	PF0 pin function control register	PF0PFS	8	8	2, 3 PCLKB	2 ICLK		555
0008 C1B9h	MPC	PF1 pin function control register	PF1PFS	8	8	2, 3 PCLKB	2 ICLK		555
0008 C1BAh	MPC	PF2 pin function control register	PF2PFS	8	8	2, 3 PCLKB	2 ICLK		555
0008 C1BDh	MPC	PF5 pin function control register	PF5PFS	8	8	2, 3 PCLKB	2 ICLK		555
0008 C1D3h	MPC	PJ3 pin function control register	PJ3PFS	8	8	2, 3 PCLKB	2 ICLK		556
0008 C1DAh	MPC	PK2 pin function control register	PK2PFS	8	8	2, 3 PCLKB	2 ICLK		557
0008 C1DBh	MPC	PK3 pin function control register	PK3PFS	8	8	2, 3 PCLKB	2 ICLK		557
0008 C1DCh	MPC	PK4 pin function control register	PK4PFS	8	8	2, 3 PCLKB	2 ICLK	557	
0008 C1DDh	MPC	PK5 pin function control register	PK5PFS	8	8	2, 3 PCLKB	2 ICLK	557	
0008 C280h	SYSTEM	Deep standby control register	DPSBYCR	8	8	4, 5 PCLKB	2, 3 ICLK	Low Power Consumption	265
0008 C282h	SYSTEM	Deep standby interrupt enable register 0	DPSIER0	8	8	4, 5 PCLKB	2, 3 ICLK		267
0008 C283h	SYSTEM	Deep standby interrupt enable register 1	DPSIER1	8	8	4, 5 PCLKB	2, 3 ICLK		268
0008 C284h	SYSTEM	Deep standby interrupt enable register 2	DPSIER2	8	8	4, 5 PCLKB	2, 3 ICLK		269
0008 C285h	SYSTEM	Deep standby interrupt enable register 3	DPSIER3	8	8	4, 5 PCLKB	2, 3 ICLK		270
0008 C286h	SYSTEM	Deep standby interrupt flag register 0	DPSIFR0	8	8	4, 5 PCLKB	2, 3 ICLK		271
0008 C287h	SYSTEM	Deep standby interrupt flag register 1	DPSIFR1	8	8	4, 5 PCLKB	2, 3 ICLK		272
0008 C288h	SYSTEM	Deep standby interrupt flag register 2	DPSIFR2	8	8	4, 5 PCLKB	2, 3 ICLK		273
0008 C289h	SYSTEM	Deep standby interrupt flag register 3	DPSIFR3	8	8	4, 5 PCLKB	2, 3 ICLK		275
0008 C28Ah	SYSTEM	Deep standby interrupt edge register 0	DPSIEGR0	8	8	4, 5 PCLKB	2, 3 ICLK		276
0008 C28Bh	SYSTEM	Deep standby interrupt edge register 1	DPSIEGR1	8	8	4, 5 PCLKB	2, 3 ICLK		277
0008 C28Ch	SYSTEM	Deep standby interrupt edge register 2	DPSIEGR2	8	8	4, 5 PCLKB	2, 3 ICLK		278
0008 C28Dh	SYSTEM	Deep standby interrupt edge register 3	DPSIEGR3	8	8	4, 5 PCLKB	2, 3 ICLK		278
0008 C290h	SYSTEM	Reset status register 0	RSTSR0	8	8	4, 5 PCLKB	2, 3 ICLK	Resets	181
0008 C291h	SYSTEM	Reset status register 1	RSTSR1	8	8	4, 5 PCLKB	2, 3 ICLK		183
0008 C293h	SYSTEM	Main clock oscillator forced oscillation control register	MOFCR	8	8	4, 5 PCLKB	2, 3 ICLK	Clock Generation Circuit	230
0008 C294h	SYSTEM	High-speed on-chip oscillator power supply control register	HOCOPCR	8	8	4, 5 PCLKB	2, 3 ICLK		231
0008 C296h	FLASH	Flash write erase protection register	FWEPROR	8	8	4, 5 PCLKB	2, 3 ICLK	ROM	1537
0008 C297h	SYSTEM	Voltage monitoring circuit control register	LVCMPCR	8	8	4, 5 PCLKB	2, 3 ICLK	LVDA	204
0008 C298h	SYSTEM	Voltage detection level select register	LVDLVL	8	8	4, 5 PCLKB	2, 3 ICLK		204
0008 C29Ah	SYSTEM	Voltage monitoring 1 circuit control register 0	LVD1CR0	8	8	4, 5 PCLKB	2, 3 ICLK		205
0008 C29Bh	SYSTEM	Voltage monitoring 2 circuit control register 0	LVD2CR0	8	8	4, 5 PCLKB	2, 3 ICLK		206



Table 5.1 List of I/O Registers (Address Order) (34/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page	
						ICLK ≥ PCLK	ICLK < PCLK			
0008 C2A0h to 0008 C2BFh	SYSTEM	Deep standby backup register 0 to 31	DPSBKR0 to 31	8	8	4, 5 PCLKB	2, 3 ICLK	Low Power Consumption	265	
0008 C300h	ICU	Group 0 interrupt source register	GRP00	32	32	1, 2 PCLKB	2 ICLK	ICUb	324	
0008 C304h	ICU	Group 1 interrupt source register	GRP01	32	32	1, 2 PCLKB	2 ICLK		324	
0008 C308h	ICU	Group 2 interrupt source register	GRP02	32	32	1, 2 PCLKB	2 ICLK		324	
0008 C30Ch	ICU	Group 3 interrupt source register	GRP03	32	32	1, 2 PCLKB	2 ICLK		324	
0008 C310h	ICU	Group 4 interrupt source register	GRP04	32	32	1, 2 PCLKB	2 ICLK		324	
0008 C314h	ICU	Group 5 interrupt source register	GRP05	32	32	1, 2 PCLKB	2 ICLK		324	
0008 C318h	ICU	Group 6 interrupt source register	GRP06	32	32	1, 2 PCLKB	2 ICLK		324	
0008 C330h	ICU	Group 12 interrupt source register	GRP12	32	32	1, 2 PCLKB	2 ICLK		324	
0008 C340h	ICU	Group 0 interrupt enable register	GEN00	32	32	1, 2 PCLKB	2 ICLK		327	
0008 C344h	ICU	Group 1 interrupt enable register	GEN01	32	32	1, 2 PCLKB	2 ICLK		327	
0008 C348h	ICU	Group 2 interrupt enable register	GEN02	32	32	1, 2 PCLKB	2 ICLK		327	
0008 C34Ch	ICU	Group 3 interrupt enable register	GEN03	32	32	1, 2 PCLKB	2 ICLK		327	
0008 C350h	ICU	Group 4 interrupt enable register	GEN04	32	32	1, 2 PCLKB	2 ICLK		327	
0008 C354h	ICU	Group 5 interrupt enable register	GEN05	32	32	1, 2 PCLKB	2 ICLK		327	
0008 C358h	ICU	Group 6 interrupt enable register	GEN06	32	32	1, 2 PCLKB	2 ICLK		327	
0008 C370h	ICU	Group 12 interrupt enable register	GEN12	32	32	1, 2 PCLKB	2 ICLK		327	
0008 C380h	ICU	Group 0 interrupt clear register	GCR00	32	32	1, 2 PCLKB	2 ICLK		329	
0008 C384h	ICU	Group 1 interrupt clear register	GCR01	32	32	1, 2 PCLKB	2 ICLK		329	
0008 C388h	ICU	Group 2 interrupt clear register	GCR02	32	32	1, 2 PCLKB	2 ICLK		329	
0008 C38Ch	ICU	Group 3 interrupt clear register	GCR03	32	32	1, 2 PCLKB	2 ICLK		329	
0008 C390h	ICU	Group 4 interrupt clear register	GCR04	32	32	1, 2 PCLKB	2 ICLK		329	
0008 C394h	ICU	Group 5 interrupt clear register	GCR05	32	32	1, 2 PCLKB	2 ICLK		329	
0008 C398h	ICU	Group 6 interrupt clear register	GCR06	32	32	1, 2 PCLKB	2 ICLK		329	
0008 C3C0h	ICU	Unit select register	SEL	32	32	1, 2 PCLKB	2 ICLK		330	
0008 C400h	RTC	64-Hz counter	R64CNT	8	8	2, 3 PCLKB	2 ICLK		RTCa	881
0008 C402h	RTC	Second counter	RSECCNT	8	8	2, 3 PCLKB	2 ICLK			882
0008 C404h	RTC	Minute counter	RMINCNT	8	8	2, 3 PCLKB	2 ICLK			882
0008 C406h	RTC	Hour counter	RHRCNT	8	8	2, 3 PCLKB	2 ICLK			883
0008 C408h	RTC	Day-of-week counter	RWKCNT	8	8	2, 3 PCLKB	2 ICLK			884
0008 C40Ah	RTC	Date counter	RDAYCNT	8	8	2, 3 PCLKB	2 ICLK			885
0008 C40Ch	RTC	Month counter	RMONCNT	8	8	2, 3 PCLKB	2 ICLK			885
0008 C40Eh	RTC	Year counter	RYRCNT	16	16	2, 3 PCLKB	2 ICLK			886
0008 C410h	RTC	Second alarm register	RSECAR	8	8	2, 3 PCLKB	2 ICLK	886		
0008 C412h	RTC	Minute alarm register	RMINAR	8	8	2, 3 PCLKB	2 ICLK	887		
0008 C414h	RTC	Hour alarm register	RHRAR	8	8	2, 3 PCLKB	2 ICLK	888		
0008 C416h	RTC	Day-of-week alarm register	RWKAR	8	8	2, 3 PCLKB	2 ICLK	889		
0008 C418h	RTC	Date alarm register	RDAYAR	8	8	2, 3 PCLKB	2 ICLK	890		
0008 C41Ah	RTC	Month alarm register	RMONAR	8	8	2, 3 PCLKB	2 ICLK	891		
0008 C41Ch	RTC	Year alarm register	RYRAR	16	16	2, 3 PCLKB	2 ICLK	892		
0008 C41Eh	RTC	Year alarm enable register	RYRAREN	8	8	2, 3 PCLKB	2 ICLK	892		
0008 C422h	RTC	RTC control register 1	RCR1	8	8	2, 3 PCLKB	2 ICLK	893		
0008 C424h	RTC	RTC control register 2	RCR2	8	8	2, 3 PCLKB	2 ICLK	894		
0008 C426h	RTC	RTC control register 3	RCR3	8	8	2, 3 PCLKB	2 ICLK	896		
0008 C428h	RTC	RTC control register 4	RCR4	8	8	2, 3 PCLKB	2 ICLK	897		
0008 C42Ah	RTC	Frequency register H	RFRH	16	16	2, 3 PCLKB	2 ICLK	898		
0008 C42Ch	RTC	Frequency register L	RFRL	16	16	2, 3 PCLKB	2 ICLK	898		
0008 C42Eh	RTC	Time error adjustment register	RADJ	8	8	2, 3 PCLKB	2 ICLK	899		

Table 5.1 List of I/O Registers (Address Order) (35/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page	
						ICLK ≥ PCLK	ICLK < PCLK			
0008 C440h	RTC	Time capture control register 0	RTCCR0	8	8	2, 3 PCLKB	2 ICLK	RTCa	900	
0008 C442h	RTC	Time capture control register 1	RTCCR1	8	8	2, 3 PCLKB	2 ICLK		900	
0008 C444h	RTC	Time capture control register 2	RTCCR2	8	8	2, 3 PCLKB	2 ICLK		900	
0008 C452h	RTC	Second capture register 0	RSECCP0	8	8	2, 3 PCLKB	2 ICLK		901	
0008 C454h	RTC	Minute capture register 0	RMINCP0	8	8	2, 3 PCLKB	2 ICLK		902	
0008 C456h	RTC	Hour capture register 0	RHRCP0	8	8	2, 3 PCLKB	2 ICLK		902	
0008 C45Ah	RTC	Date capture register 0	RDAYCP0	8	8	2, 3 PCLKB	2 ICLK		903	
0008 C45Ch	RTC	Month capture register 0	RMONCP0	8	8	2, 3 PCLKB	2 ICLK		903	
0008 C462h	RTC	Second capture register 1	RSECCP1	8	8	2, 3 PCLKB	2 ICLK		901	
0008 C464h	RTC	Minute capture register 1	RMINCP1	8	8	2, 3 PCLKB	2 ICLK		902	
0008 C466h	RTC	Hour capture register 1	RHRCP1	8	8	2, 3 PCLKB	2 ICLK		902	
0008 C46Ah	RTC	Date capture register 1	RDAYCP1	8	8	2, 3 PCLKB	2 ICLK		903	
0008 C46Ch	RTC	Month capture register 1	RMONCP1	8	8	2, 3 PCLKB	2 ICLK		903	
0008 C472h	RTC	Second capture register 2	RSECCP2	8	8	2, 3 PCLKB	2 ICLK		901	
0008 C474h	RTC	Minute capture register 2	RMINCP2	8	8	2, 3 PCLKB	2 ICLK		902	
0008 C476h	RTC	Hour capture register 2	RHRCP2	8	8	2, 3 PCLKB	2 ICLK		902	
0008 C47Ah	RTC	Date capture register 2	RDAYCP2	8	8	2, 3 PCLKB	2 ICLK		903	
0008 C47Ch	RTC	Month capture register 2	RMONCP2	8	8	2, 3 PCLKB	2 ICLK		903	
0008 C500h	TEMPS	Temperature sensor control register	TSCR	8	8	2, 3 PCLKB	2 ICLK		Temperature Sensor	1459
0008 C880h	SYSTEM	Counter-clock extension register 1	SCK1	8	8	2, 3 PCLKB	2 ICLK		MCK	245
0008 C890h	SYSTEM	Counter-clock extension register 2	SCK2	8	8	2, 3 PCLKB	2 ICLK			245
0009 0200h to 0009 03FFh	CAN0	Mailbox registers 0 to 31	MB0 to 31	128	8, 16, 32	2, 3 PCLKB	2 ICLK		CAN	1242
0009 0400h to 0009 041Fh	CAN0	Mask registers 0 to 7	MKR0 to 7	32	8, 16, 32	2, 3 PCLKB	2 ICLK			1242
0009 0420h	CAN0	FIFO received ID compare register 0	FIDCR0	32	8, 16, 32	2, 3 PCLKB	2 ICLK	1224		
0009 0424h	CAN0	FIFO received ID compare register 1	FIDCR1	32	8, 16, 32	2, 3 PCLKB	2 ICLK	1224		
0009 0428h	CAN0	Mask invalid register	MKIVLR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	1225		
0009 042Ch	CAN0	Mailbox interrupt enable register	MIER	32	8, 16, 32	2, 3 PCLKB	2 ICLK	1230		
0009 0820h to 0009 083Fh	CAN0	Message control registers 0 to 31	MCTL0 to 31	8	8	2, 3 PCLKB	2 ICLK	1231		
0009 0840h	CAN0	Control register	CTLR	16	8, 16	2, 3 PCLKB	2 ICLK	1217		
0009 0842h	CAN0	Status register	STR	16	8, 16	2, 3 PCLKB	2 ICLK	1240		
0009 0844h	CAN0	Bit configuration register	BCR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	1221		
0009 0848h	CAN0	Receive FIFO control register	RFCR	8	8	2, 3 PCLKB	2 ICLK	1234		
0009 0849h	CAN0	Receive FIFO pointer control register	RFPCR	8	8	2, 3 PCLKB	2 ICLK	1237		
0009 084Ah	CAN0	Transmit FIFO control register	TFCR	8	8	2, 3 PCLKB	2 ICLK	1237		
0009 084Bh	CAN0	Transmit FIFO pointer control register	TFPCR	8	8	2, 3 PCLKB	2 ICLK	1237		
0009 084Ch	CAN0	Error interrupt enable register	EIER	8	8	2, 3 PCLKB	2 ICLK	1246		
0009 084Dh	CAN0	Error interrupt factor judge register	EIFR	8	8	2, 3 PCLKB	2 ICLK	1247		
0009 084Eh	CAN0	Receive error count register	RECR	8	8	2, 3 PCLKB	2 ICLK	1249		
0009 084Fh	CAN0	Transmit error count register	TECR	8	8	2, 3 PCLKB	2 ICLK	1250		
0009 0850h	CAN0	Error code store register	ECSR	8	8	2, 3 PCLKB	2 ICLK	1250		
0009 0851h	CAN0	Channel search support register	CSSR	8	8	2, 3 PCLKB	2 ICLK	1244		
0009 0852h	CAN0	Mailbox search status register	MSSR	8	8	2, 3 PCLKB	2 ICLK	1244		
0009 0853h	CAN0	Mailbox search mode register	MSMR	8	8	2, 3 PCLKB	2 ICLK	1242		
0009 0854h	CAN0	Time stamp register	TSR	16	16	2, 3 PCLKB	2 ICLK	1242		
0009 0856h	CAN0	Acceptance filter support register	AFSR	16	16	2, 3 PCLKB	2 ICLK	1245		

Table 5.1 List of I/O Registers (Address Order) (36/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0009 0858h	CAN0	Test control register	TCR	8	8	2, 3 PCLKB	2 ICLK	CAN	1252
0009 1200h to 0009 13FFh	CAN1	Mailbox registers 0 to 31	MB0 to 31	128	8, 16, 32	2, 3 PCLKB	2 ICLK		1226
0009 1400h to 0009 141Fh	CAN1	Mask register 0 to 7	MKR0 to 7	32	8, 16, 32	2, 3 PCLKB	2 ICLK		1223
0009 1420h	CAN1	FIFO received ID compare register 0	FIDCR0	32	8, 16, 32	2, 3 PCLKB	2 ICLK		1224
0009 1424h	CAN1	FIFO received ID compare register 1	FIDCR1	32	8, 16, 32	2, 3 PCLKB	2 ICLK		1224
0009 1428h	CAN1	Mask invalid register	MKIVLR	32	8, 16, 32	2, 3 PCLKB	2 ICLK		1225
0009 142Ch	CAN1	Mailbox interrupt enable register	MIER	32	8, 16, 32	2, 3 PCLKB	2 ICLK		1230
0009 1820h to 0009 183Fh	CAN1	Message control registers 0 to 31	MCTL0 to 31	8	8	2, 3 PCLKB	2 ICLK		1231
0009 1840h	CAN1	Control register	CTLR	16	8, 16	2, 3 PCLKB	2 ICLK		1217
0009 1842h	CAN1	Status register	STR	16	8, 16	2, 3 PCLKB	2 ICLK		1240
0009 1844h	CAN1	Bit configuration register	BCR	32	8, 16, 32	2, 3 PCLKB	2 ICLK		1221
0009 1848h	CAN1	Receive FIFO control register	RFCR	8	8	2, 3 PCLKB	2 ICLK		1234
0009 1849h	CAN1	Receive FIFO pointer control register	RFPCR	8	8	2, 3 PCLKB	2 ICLK		1237
0009 184Ah	CAN1	Transmit FIFO control register	TFCR	8	8	2, 3 PCLKB	2 ICLK		1237
0009 184Bh	CAN1	Transmit FIFO pointer control register	TFPCR	8	8	2, 3 PCLKB	2 ICLK		1239
0009 184Ch	CAN1	Error interrupt enable register	EIER	8	8	2, 3 PCLKB	2 ICLK		1246
0009 184Dh	CAN1	Error interrupt factor judge register	EIFR	8	8	2, 3 PCLKB	2 ICLK		1247
0009 184Eh	CAN1	Receive error count register	RECR	8	8	2, 3 PCLKB	2 ICLK		1249
0009 184Fh	CAN1	Transmit error count register	TECR	8	8	2, 3 PCLKB	2 ICLK		1250
0009 1850h	CAN1	Error code store register	ECSR	8	8	2, 3 PCLKB	2 ICLK		1250
0009 1851h	CAN1	Channel search support register	CSSR	8	8	2, 3 PCLKB	2 ICLK		1244
0009 1852h	CAN1	Mailbox search status register	MSSR	8	8	2, 3 PCLKB	2 ICLK		1243
0009 1853h	CAN1	Mailbox search mode register	MSMR	8	8	2, 3 PCLKB	2 ICLK		1243
0009 1854h	CAN1	Time stamp register	TSR	16	8, 16	2, 3 PCLKB	2 ICLK		1251
0009 1856h	CAN1	Acceptance filter support register	AFSR	16	8, 16	2, 3 PCLKB	2 ICLK		1245
0009 1858h	CAN1	Test control register	TCR	8	8	2, 3 PCLKB	2 ICLK		1252
0009 2200h to 0009 23FFh	CAN2	Mailbox registers 0 to 31	MB0 to 31	128	8, 16, 32	2, 3 PCLKB	2 ICLK		1226
0009 2400h to 0009 241Fh	CAN2	Mask register 0 to 7	MKR0 to 7	32	8, 16, 32	2, 3 PCLKB	2 ICLK		1223
0009 2420h	CAN2	FIFO received ID compare register 0	FIDCR0	32	8, 16, 32	2, 3 PCLKB	2 ICLK		1224
0009 2424h	CAN2	FIFO received ID compare register 1	FIDCR1	32	8, 16, 32	2, 3 PCLKB	2 ICLK		1224
0009 2428h	CAN2	Mask invalid register	MKIVLR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	1225	
0009 242Ch	CAN2	Mailbox interrupt enable register	MIER	32	8, 16, 32	2, 3 PCLKB	2 ICLK	1230	
0009 2820h to 0009 283Fh	CAN2	Message control registers 0 to 31	MCTL0 to 31	8	8	2, 3 PCLKB	2 ICLK	1231	
0009 2820h to 0009 283Fh	CAN2	Message control registers 0 to 31	MCTL0 to 31	8	8	2, 3 PCLKB	2 ICLK	1231	
0009 2840h	CAN2	Control register	CTLR	16	8, 16	2, 3 PCLKB	2 ICLK	1217	
0009 2842h	CAN2	Status register	STR	16	8, 16	2, 3 PCLKB	2 ICLK	1240	
0009 2844h	CAN2	Bit configuration register	BCR	32	8, 16, 32	2, 3 PCLKB	2 ICLK	1221	
0009 2848h	CAN2	Receive FIFO control register	RFCR	8	8	2, 3 PCLKB	2 ICLK	1234	
0009 2849h	CAN2	Receive FIFO pointer control register	RFPCR	8	8	2, 3 PCLKB	2 ICLK	1237	
0009 284Ah	CAN2	Transmit FIFO control register	TFCR	8	8	2, 3 PCLKB	2 ICLK	1237	

Table 5.1 List of I/O Registers (Address Order) (37/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
0009 284Bh	CAN2	Transmit FIFO pointer control register	TFPCR	8	8	2, 3 PCLKB	2 ICLK	CAN	1239
0009 284Ch	CAN2	Error interrupt enable register	EIER	8	8	2, 3 PCLKB	2 ICLK		1246
0009 284Dh	CAN2	Error interrupt factor judge register	EIFR	8	8	2, 3 PCLKB	2 ICLK		1247
0009 284Eh	CAN2	Receive error count register	RECR	8	8	2, 3 PCLKB	2 ICLK		1249
0009 284Fh	CAN2	Transmit error count register	TECR	8	8	2, 3 PCLKB	2 ICLK		1250
0009 2850h	CAN2	Error code store register	ECSR	8	8	2, 3 PCLKB	2 ICLK		1250
0009 2851h	CAN2	Channel search support register	CSSR	8	8	2, 3 PCLKB	2 ICLK		1244
0009 2852h	CAN2	Mailbox search status register	MSSR	8	8	2, 3 PCLKB	2 ICLK		1243
0009 2853h	CAN2	Mailbox search mode register	MSMR	8	8	2, 3 PCLKB	2 ICLK		1242
0009 2854h	CAN2	Time stamp register	TSR	16	16	2, 3 PCLKB	2 ICLK		1251
0009 2856h	CAN2	Acceptance filter support register	AFSR	16	16	2, 3 PCLKB	2 ICLK		1245
0009 2858h	CAN2	Test control register	TCR	8	8	2, 3 PCLKB	2 ICLK		1252
000A 0000h	USB0	System configuration control register	SYSCFG	16	16	3, 4 PCLKB	2, 3 ICLK		USBa
000A 0004h	USB0	System configuration status register 0	SYSSTS0	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8	951	
000A 0008h	USB0	Device state control register 0	DVSTCTR0	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8	952	
000A 0014h	USB0	CFIFO port register	CFIFO	16	8, 16	3, 4 PCLKB	2, 3 ICLK	953	
000A 0018h	USB0	D0FIFO port register	D0FIFO	16	8, 16	3, 4 PCLKB	2, 3 ICLK	953	
000A 001Ch	USB0	D1FIFO port register	D1FIFO	16	8, 16	3, 4 PCLKB	2, 3 ICLK	953	
000A 0020h	USB0	CFIFO port select register	CFIFOSEL	16	16	3, 4 PCLKB	2, 3 ICLK	955	
000A 0022h	USB0	CFIFO port control register	CFIFOCTR	16	16	3, 4 PCLKB	2, 3 ICLK	959	
000A 0028h	USB0	D0FIFO port select register	D0FIFOSEL	16	16	3, 4 PCLKB	2, 3 ICLK	955	
000A 002Ah	USB0	D0FIFO port control register	D0FIFOCTR	16	16	3, 4 PCLKB	2, 3 ICLK	959	
000A 002Ch	USB0	D1FIFO port select register	D1FIFOSEL	16	16	3, 4 PCLKB	2, 3 ICLK	955	
000A 002Eh	USB0	D1FIFO port control register	D1FIFOCTR	16	16	3, 4 PCLKB	2, 3 ICLK	959	
000A 0030h	USB0	Interrupt enable register 0	INTENB0	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8	961	
000A 0036h	USB0	BRDY interrupt status register	BRDYENB	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8	968	
000A 0038h	USB0	NRDY interrupt status register	NRDYENB	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8	968	

Table 5.1 List of I/O Registers (Address Order) (38/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000A 003Ah	USB0	BEMP interrupt status register	BEMPENB	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8	USBa	970
000A 003Ch	USB0	SOF output configuration register	SOFCFG	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		965
000A 0040h	USB0	Interrupt status register 0	INTSTS0	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		965
000A 0046h	USB0	BRDY interrupt status register	BRDYSTS	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		968
000A 0048h	USB0	NRDY interrupt status register	NRDYSTS	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		969
000A 004Ah	USB0	BEMP interrupt status register	BEMPSTS	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		970
000A 004Ch	USB0	Frame number register	FRMNUM	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		971
000A 004Eh	USB0	Device state changing register	DVCHGR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		972
000A 0050h	USB0	USB address register	USBADDR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		972
000A 0054h	USB0	USB request type register	USBREQ	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		973

**Table 5.1 List of I/O Registers (Address Order) (39/42)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000A 0056h	USB0	USB request value register	USBVAL	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8	USBa	973
000A 0058h	USB0	USB request index register	USBINDX	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		974
000A 005Ah	USB0	USB request length register	USBLENG	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		974
000A 005Eh	USB0	DCP maximum packet size register	DCPMAXP	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		975
000A 0060h	USB0	DCP control register	DCPCTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		976
000A 0064h	USB0	Pipe window select register	PIPESEL	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		978
000A 0068h	USB0	Pipe configuration register	PIPECFG	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		979
000A 006Ch	USB0	Pipe maximum packet size register	PIPEMAXP	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		981
000A 006Eh	USB0	Pipe cycle control register	PIPEPERI	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		982
000A 0070h	USB0	Pipe 1 control register	PIPE1CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		983

Table 5.1 List of I/O Registers (Address Order) (40/42)

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000A 0072h	USB0	Pipe 2 control register	PIPE2CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8	USBa	983
000A 0074h	USB0	Pipe 3 control register	PIPE3CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		983
000A 0076h	USB0	Pipe 4 control register	PIPE4CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		983
000A 0078h	USB0	Pipe 5 control register	PIPE5CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		983
000A 007Ah	USB0	Pipe 6 control register	PIPE6CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		983
000A 007Ch	USB0	Pipe 7 control register	PIPE7CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		983
000A 007Eh	USB0	Pipe 8 control register	PIPE8CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		983
000A 0080h	USB0	Pipe 9 control register	PIPE9CTR	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		983
000A 0090h	USB0	Pipe 1 transaction counter enable register	PIPE1TRE	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		989
000A 0092h	USB0	Pipe 1 transaction counter register	PIPE1TRN	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		990

**Table 5.1 List of I/O Registers (Address Order) (41/42)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
000A 0094h	USB0	Pipe 2 transaction counter enable register	PIPE2TRE	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8	USBa	989
000A 0096h	USB0	Pipe 2 transaction counter register	PIPE2TRN	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		990
000A 0098h	USB0	Pipe 3 transaction counter enable register	PIPE3TRE	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8	USBa	989
000A 009Ah	USB0	Pipe 3 transaction counter register	PIPE3TRN	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		990
000A 009Ch	USB0	Pipe 4 transaction counter enable register	PIPE4TRE	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8	USBa	989
000A 009Eh	USB0	Pipe 4 transaction counter register	PIPE4TRN	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		990
000A 00A0h	USB0	Pipe 5 transaction counter enable register	PIPE5TRE	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8	USBa	989
000A 00A2h	USB0	Pipe 5 transaction counter register	PIPE5TRN	16	16	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		990
000A 0400h	USB0	Deep standby USB transceiver control/pin monitor register	DPUSR0R	32	32	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8	USBa	991
000A 0404h	USB0	Deep standby USB suspend/resume interrupt register	DPUSR1R	32	32	9 PCLKB or more	Rounded up to the nearest integer greater than 1 + 9/ (frequency ratio of ICLK/ PCLKB)*8		992



**Table 5.1 List of I/O Registers (Address Order) (42/42)**

Address	Module Symbol	Register Name	Register Symbol	Number of Bits	Access Size	Number of Access States		Related Function	Reference Page
						ICLK ≥ PCLK	ICLK < PCLK		
007F C402h	FLASH	Flash mode register	FMODR	8	8	2, 3 FCLK	2, 3 ICLK	Flash Memory	1472
007F C410h	FLASH	Flash access status register	FASTAT	8	8	2, 3 FCLK	2, 3 ICLK		1473
007F C411h	FLASH	Flash access error interrupt enable register	FAEINT	8	8	2, 3 FCLK	2, 3 ICLK		1476
007F C412h	FLASH	Flash ready interrupt enable register	FRDYIE	8	8	2, 3 FCLK	2, 3 ICLK		1477
007F C440h	FLASH	E2 DataFlash read enable register 0	DFLRE0	16	16	2, 3 FCLK	2, 3 ICLK		1478
007F C442h	FLASH	E2 DataFlash read enable register 1	DFLRE1	16	16	2, 3 FCLK	2, 3 ICLK		1479
007F C450h	FLASH	E2 DataFlash P/E enable register 0	DFLWE0	16	16	2, 3 FCLK	2, 3 ICLK		1480
007F C452h	FLASH	E2 DataFlash P/E enable register 1	DFLWE1	16	16	2, 3 FCLK	2, 3 ICLK		1481
007F C454h	FLASH	FCU RAM enable register	FCURAME	16	16	2, 3 FCLK	2, 3 ICLK		1482
007F FFB0h	FLASH	Flash status register 0	FSTATR0	8	8	2, 3 FCLK	2, 3 ICLK		1483
007F FFB1h	FLASH	Flash status register 1	FSTATR1	8	8	2, 3 FCLK	2, 3 ICLK		1485
007F FFB2h	FLASH	Flash P/E mode entry register	FENTRYR	16	16	2, 3 FCLK	2, 3 ICLK		1486
007F FFB4h	FLASH	Flash protection register	FPROTR	16	16	2, 3 FCLK	2, 3 ICLK		1488
007F FFB6h	FLASH	Flash reset register	FRESETR	16	16	2, 3 FCLK	2, 3 ICLK		1489
007F FFBAh	FLASH	FCU command register	FCMDR	16	16	2, 3 FCLK	2, 3 ICLK		1490
007F FFC8h	FLASH	FCU processing switching register	FCPSR	16	16	2, 3 FCLK	2, 3 ICLK		1491
007F FFCAh	FLASH	E2 data flash blank check control register	DFLBCCNT	16	16	2, 3 FCLK	2, 3 ICLK		1491
007F FFCCh	FLASH	Flash P/E status register	FPESTAT	16	16	2, 3 FCLK	2, 3 ICLK		1492
007F FFCEh	FLASH	E2 DataFlash blank check status register	DFLBCSTAT	16	16	2, 3 FCLK	2, 3 ICLK		1492
007F FFE8h	FLASH	Peripheral clock notification register	PCKAR	16	16	2, 3 FCLK	2, 3 ICLK		1493

o: Available, x: Not available

- Note 1. When the same output trigger is specified for pulse output groups 2 and 3 by the PPG0.PCR setting, the PPG0.NDRH address is 000881ECh. When different output triggers are specified, the PPG0.NDRH addresses for pulse output groups 2 and 3 are 000881EEh and 000881ECh, respectively.
- Note 2. When the same output trigger is specified for pulse output groups 0 and 1 by the PPG0.PCR setting, the PPG0.NDRL address is 000881EDh. When different output triggers are specified, the PPG0.NDRL addresses for pulse output groups 0 and 1 are 000881EFh and 000881EDh, respectively.
- Note 3. When the same output trigger is specified for pulse output groups 6 and 7 by the PPG1.PCR setting, the PPG1.NDRH address is 000881FCh. When different output triggers are specified, the PPG1.NDRH addresses for pulse output groups 6 and 7 are 000881FEh and 000881FCh, respectively.
- Note 4. When the same output trigger is specified for pulse output groups 4 and 5 by the PPG1.PCR setting, the PPG1.NDRL address is 000881FDh. When different output triggers are specified, the PPG1.NDRL addresses for pulse output groups 4 and 5 are 000881FFh and 000881FDh, respectively.
- Note 5. Odd addresses should not be accessed in 16-bit units. When accessing a register in 16-bit units, access the address of the TMR0 or TMR2 register. Table 26.4 lists register allocation for 16-bit access.
- Note 6. The CAN2 module is not provided in products less than 1 Mbyte of ROM.
- Note 7. The CAN0 module is not provided in products less than 512 Kbytes of ROM.
- Note 8. When the register is accessed while the USB is operating, a delay may be generated in accessing.

## 6. Resets

### 6.1 Overview

There are nine types of resets: RES# pin reset, power-on reset, voltage-monitoring 0 reset, voltage-monitoring 1 reset, voltage-monitoring 2 reset, deep software standby reset, independent watchdog timer reset, 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 detection: VPOR)*1
Voltage-monitoring 0 reset	VCC falls (voltage detection: Vdet0)*1
Voltage-monitoring 1 reset	VCC falls (voltage detection: Vdet1)*1
Voltage-monitoring 2 reset	VCC falls (voltage detection: Vdet2)*1
Deep software standby reset	Deep software standby mode is canceled by an interrupt.
Independent watchdog timer reset	The independent watchdog timer underflows, or a refresh error occurs.
Watchdog timer reset	The watchdog timer underflows, or a refresh error occurs.
Software reset	Register setting

Note 1. For details on details on the voltages to be monitored (VPOR, Vdet0, Vdet1, and Vdet2), see section 8, Voltage Detection Circuit (LVDA) and section 45, 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 to be Initialized by Each Reset Source (1/2)**

Targets to be Initialized	Reset Source								
	RES# pin Reset	Power-On Reset	Voltage- Monitoring 0 Reset	Independent Watchdog Timer Reset	Watchdog Timer Reset	Voltage- Monitoring 1 Reset	Voltage- Monitoring 2 Reset	Deep Software Standby Reset	Software Reset
Register related to the power-on reset detect flag (RSTSR0.PORF)	○	—	—	—	—	—	—	—	—
Register related to the cold start/warm start determination flag (RSTSR1.CWSF)	—	○	—	—	—	—	—	—	—
Register related to the voltage-monitoring 0 reset detect flag (RSTSR0.LVD0RF)	○	○	—	—	—	—	—	—	—
Independent watchdog timer reset detect flag (RSTSR2.IWDTRF)	○	○	○	—	—	—	—	○	—
Registers related to the independent watchdog timer (IWDTRR, IWDTCR, IWDTSR, IWDTRCR, IWDTCSTPR, ILOCCR)	○	○	○	—	—	—	—	○	—
Registers related to the watchdog timer reset detect flag (RSTSR2.WDTRF)	○	○	○	○	—	—	—	○	—
Registers related to the watchdog timer (WDTRR, WDTCR, WDTSR, WDTRCR)	○	○	○	○	—	—	—	○	—
Voltage-monitoring 1 reset detect flag (RSTSR0.LVD1RF)	○	○	○	○	○	—	—	—	—
Registers related to the voltage monitor function 1 (LVD1CR0, LVCMPCR.LVD1E, LVDLVLR.LVD1LVL)  (LVD1CR1, LVD1SR)	○ ○	○ ○	○ ○	○ ○	○ ○	— —	— —	— ○	— —
Voltage-monitoring 2 reset detect flag (RSTSR0.LVD2RF)	○	○	○	○	○	○	—	—	—
Registers related to the voltage monitor function 2 (RSTSR0.LVD2RF, LVD2CR0, LVCMPCR.LVD2E, LVDLVLR.LVD2LVL)  (LVD2CR1, LVD2SR)	○ ○	○ ○	○ ○	○ ○	○ ○	○ ○	— —	— ○	— —
Register related to the deep software standby reset detect flag (RSTSR0.DPSRSTF)	○	○	○	○	○	○	○	—	—
Register related to the software reset detect flag (RSTSR2.SWRF)	○	○	○	○	○	○	○	○	—
Register related to the realtime clock*1	—	—	—	—	—	—	—	—	—
Register related to high-speed on-chip oscillator (HOCOPCR.HOCOPCNT)	○	○	○	○	○	○	○	—	○
Register related to main clock oscillator (MOFCR)	○	○	○	○	○	○	○	—	○
Pin state	○	○	○	○	○	○	○	—	○

**Table 6.2 Targets to be Initialized by Each Reset Source (2/2)**

Targets to be Initialized	Reset Source								
	RES# pin Reset	Power-On Reset	Voltage- Monitoring 0 Reset	Independent Watchdog Timer Reset	Watchdog Timer Reset	Voltage- Monitoring 1 Reset	Voltage- Monitoring 2 Reset	Deep Software Standby Reset	Software Reset
Registers related to the low power-consumption function (DPSBYCR, DPSIER0 to DPSIER3, DPSIFR0 to DPSIFR3, DPSIEGR0 to DPSIEGR3, DPUSR0R, DPUSR1R)	○	○	○	○	○	○	○	—	○
Registers other than the above, CPU, and internal state	○	○	○	○	○	○	○	○	○

○: Targets to be initialized, —: No change occurs.

Note 1. Some control bits (RCR1.CIE, RCR2.RTCOE, ADJ30, and RESET) are initialized by all types of resets. For details on the target bits, refer to section 28, Realtime Clock (RTCa).

When a reset is canceled, the reset exception handling starts. For details on the reset exception handling, see section 14, 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
DPSRS TF	—	—	—	LVD2R F	LVD1R F	LVD0R F	PORF

Value after reset: 0\*1 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
b6 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DPSRSTF	Deep Software Standby Reset Flag	0: Deep software standby mode cancelation not requested by an interrupt. 1: Deep software standby mode cancelation requested by an interrupt.	R(W) *2

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 a reset listed in Table 6.2 occurs.
- 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 a reset listed in Table 6.2 occurs.
- 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.

**DPSRSTF Flag (Deep Software Standby Reset Flag)**

The DPSRSTF flag indicates that deep software standby mode has been canceled by an interrupt and that an internal reset (deep software standby reset) occurred.

[Setting condition]

- When deep software standby mode is cancelled by an internal interrupt.  
For details, see section 11, Low Power Consumption.

[Clearing conditions]

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

## 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 by a power-on reset. It is not initialized by a reset signal generated by the RES# pin reset.

[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	WDTR F	IWDTR F

Value after reset: 0 0 0 0 0 0\*1 0\*1 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	WDTRF	Watchdog Timer Reset Detect Flag	0: Watchdog timer reset not detected. 1: Watchdog timer reset detected.	R(W) *2
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.

**WDTRF Flag (Watchdog Timer Reset Detect Flag)**

The WDTRF flag indicates that a watchdog timer reset has occurred.

[Setting condition]

- When a watchdog timer reset occurs.

[Clearing conditions]

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

**SWRF Flag (Software Reset Detect Flag)**

The SWRF flag indicates that a software reset has occurred.

[Setting condition]

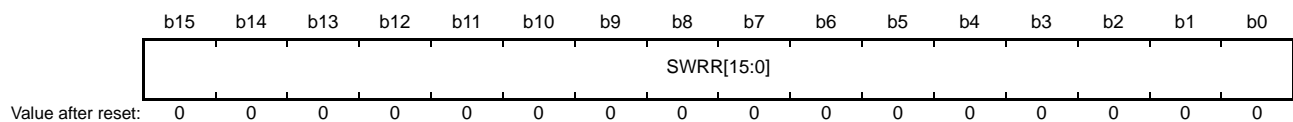
- When a software reset occurs.

[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



## 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# cancelation wait time (tRESWT) has elapsed, and then the CPU starts the reset exception handling.

For details, see section 45, 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. If the RES# pin is in a high level state when power is supplied, a power-on reset is generated. 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 of the external power supply and the LSI circuit. After a power-on reset has been generated, the PORF flag in RSTSR0 is set to 1. The PORF flag is initialized by the RES# pin reset.

The voltage monitoring 0 reset is an internal reset generated by the voltage monitoring circuit. If the voltage detection 0 level selection (LVDAS) bit in the option function select register 1 (OFS1) is 0 (voltage monitoring 0 reset is enabled after a reset) and VCC falls below Vdet0, the RSTSR0.LVDORF 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.

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 (LVDA).

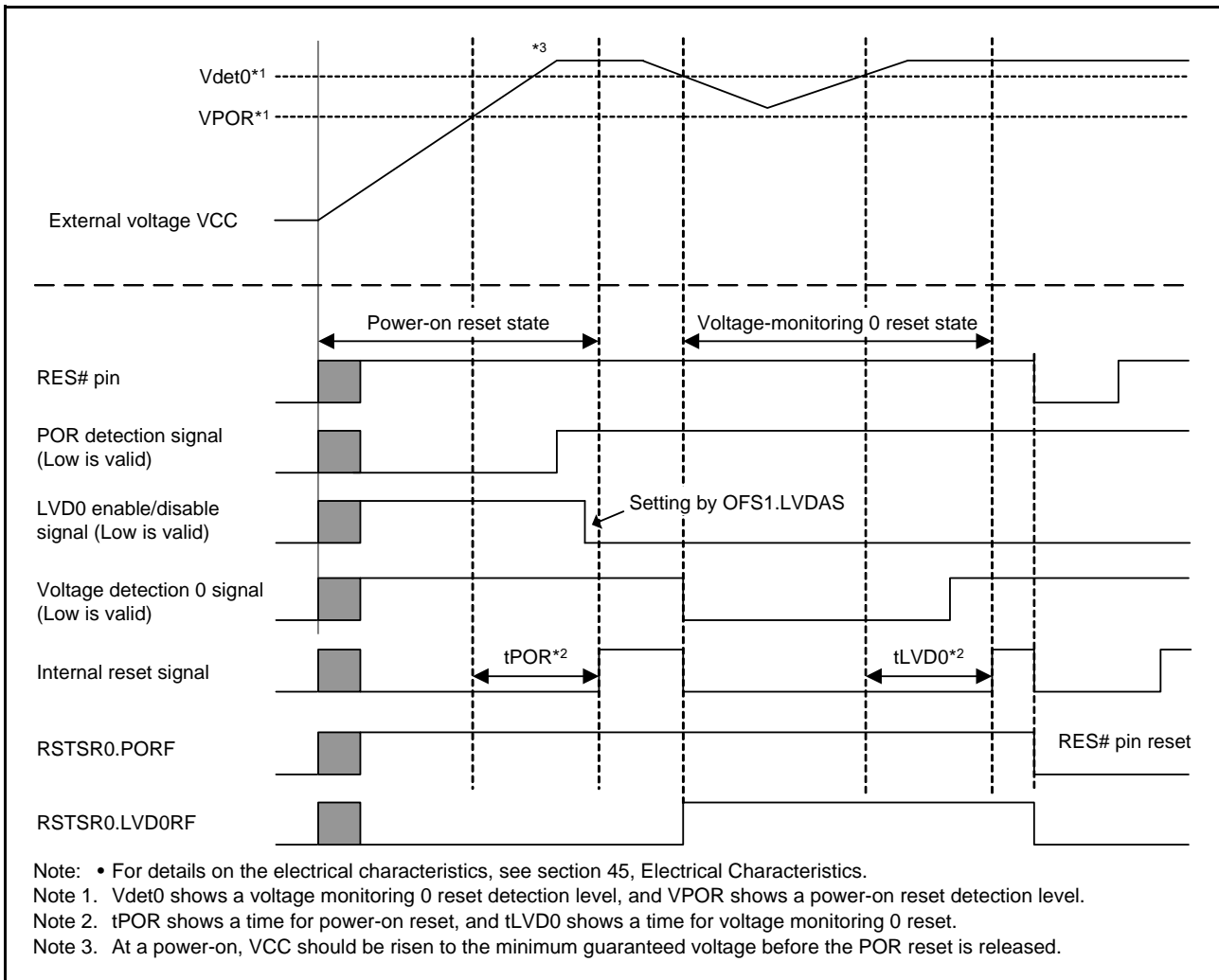


Figure 6.1 Operation Examples During a Power-On Reset and Voltage Monitoring 0 Reset

### 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 negate select bit (LVD1RN) in the LVD1CR0. 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 LVD1 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 LVD1 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 negate select bit (LVD2RN) in LDV2CR0 register. Detection levels Vdet1 and Vdet2 can be changed by settings in the voltage detection select register (LDVLVLR).

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 (LVDA).

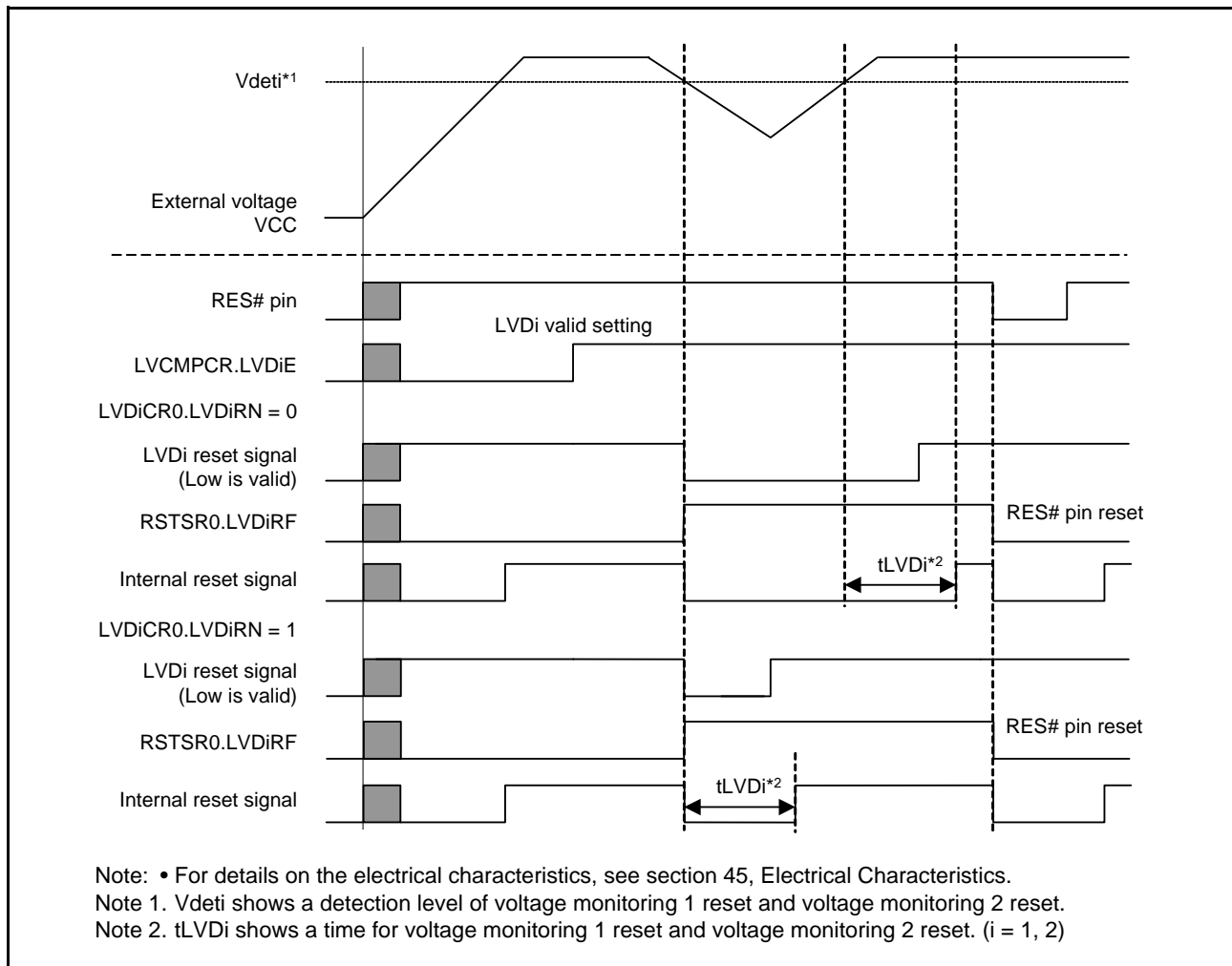


Figure 6.2 Operation Examples During Voltage Monitoring 1 Reset and Voltage Monitoring 2 Reset

### 6.3.4 Deep Software Standby Reset

This is an internal reset generated when deep software standby mode is canceled by an interrupt.

When a deep software standby mode cancelation source is generated, a deep software standby reset is generated. The deep software standby reset is canceled after tDSBY (return time after deep software standby mode cancelation) has elapsed. At the same time, deep software standby mode is also canceled.

When tDSBYWT (wait time after deep software standby mode cancelation) has elapsed after deep software standby mode has been canceled, the internal reset is canceled and the CPU starts the reset exception handling.

For details of the deep software standby reset, see section 11, Low Power Consumption.

### 6.3.5 Independent Watchdog Timer Reset

Independent watchdog timer reset is an internal reset generated by the independent watchdog timer.

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 when refresh operation is disabled. 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 30, Independent Watchdog Timer (IWDtA).

### 6.3.6 Watchdog Timer Reset

The watchdog-timer reset is an internal reset from the watchdog timer.

When output of the watchdog timer reset is selected, a watchdog timer reset is generated if the watchdog timer underflows, or if data is written when refresh operation is disabled. When the internal reset time (tRESW2) has elapsed after the watchdog timer reset is generated, the internal reset is canceled and the CPU starts the reset exception handling.

For details on the watchdog timer reset, see section 29, Watchdog Timer (WDTA).

### 6.3.7 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.

### 6.3.8 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.

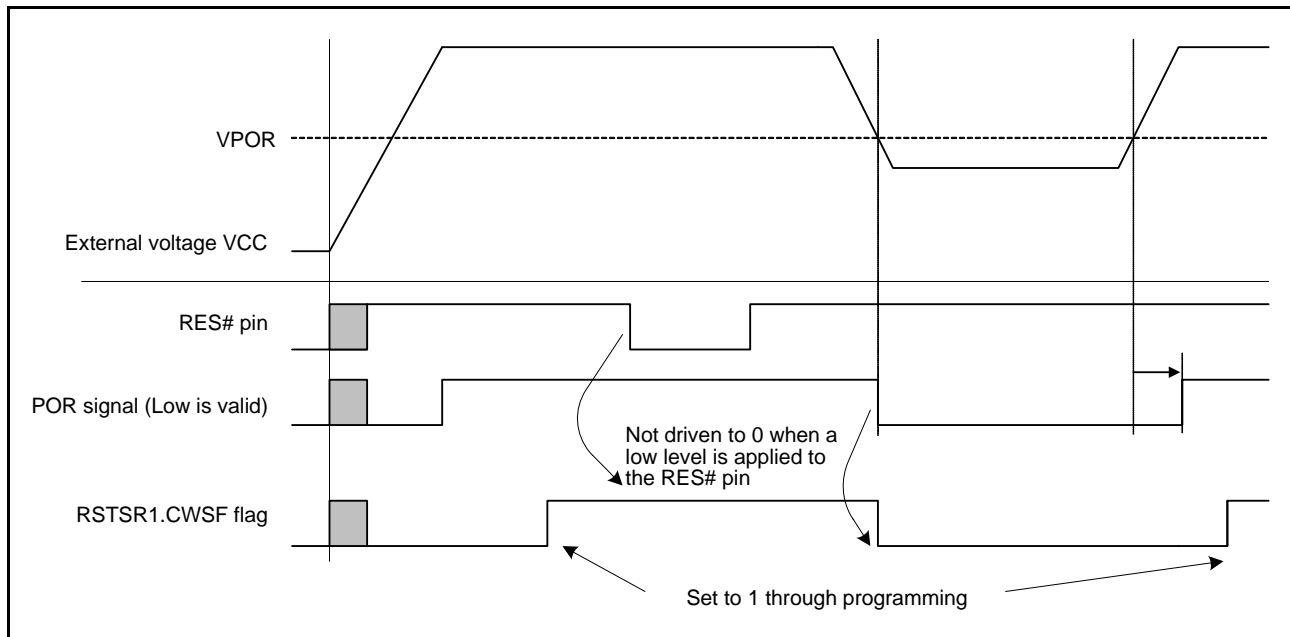


Figure 6.3 Example of Cold/Warm Start Determination Operation

### 6.3.9 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.

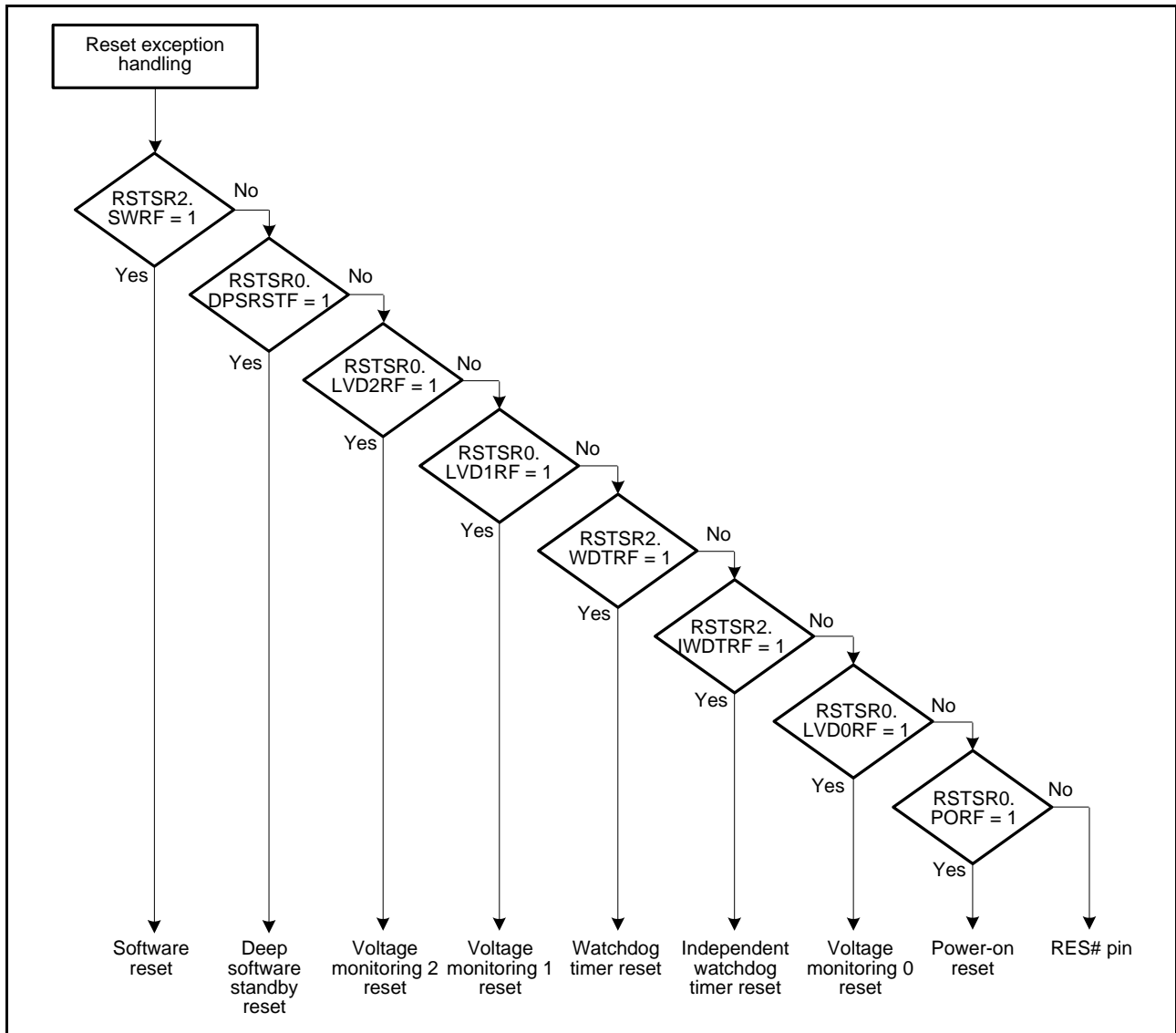


Figure 6.4 Example of Reset Generation Source Determination Flow

## 7. Option-Setting Memory

### 7.1 Overview

Option-setting memory 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.

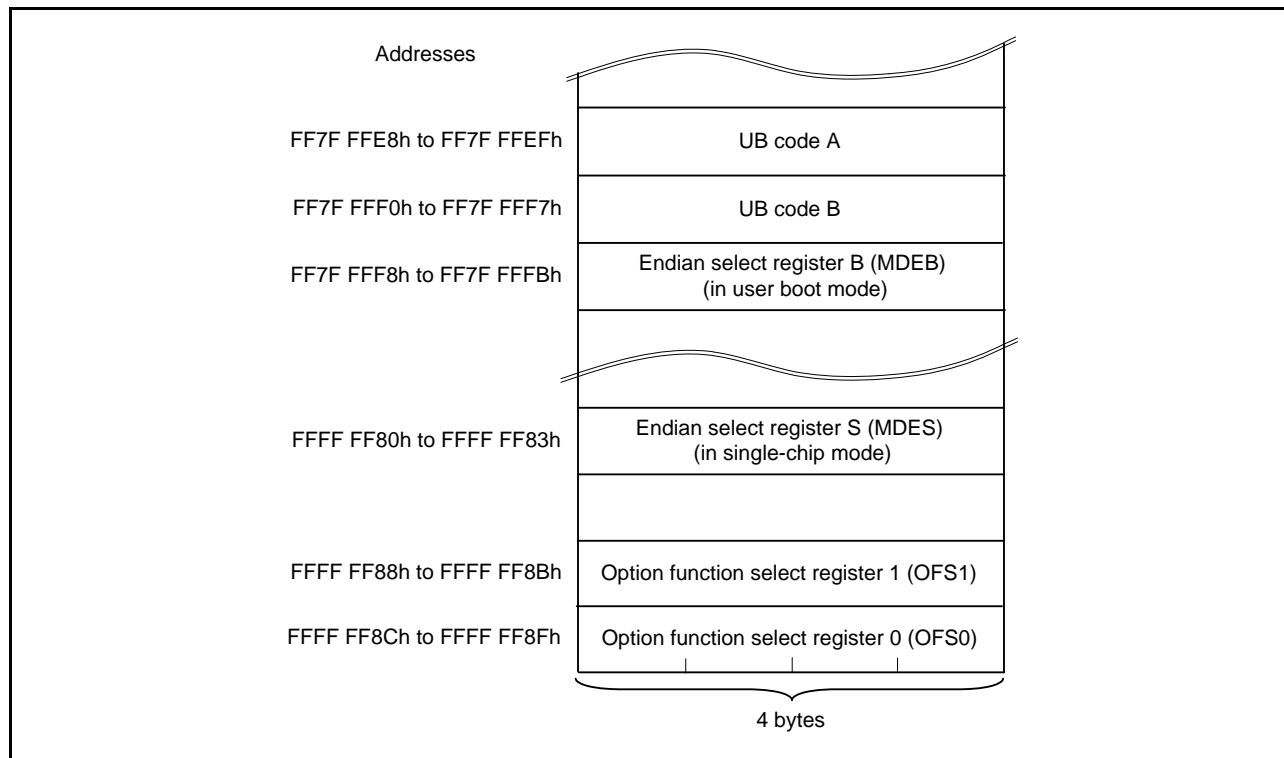


Figure 7.1 Option-Setting Memory Area

## 7.2 Register Descriptions

### 7.2.1 Option Function Select Register 0 (OFS0)

Address(es): FFFF FF8Ch

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	—	—	WDTRS TIRQS	WDTRPSS[1:0]	WDTRPES[1:0]	WDTCKS[3:0]			WDTTOPS[1:0]	WDTST RT	—				

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
—	IWDTS LCSTP	—	IWDR STIRQS	IWDRPSS[1:0]	IWDRPES[1:0]	IWDCKS[3:0]			IWDTTOPS[1:0]	IWDTST TRT	—				

Value after reset: The value set by the user\*1

Bit	Symbol	Bit Name	Description	R/W
b0	—	Reserved	When reading, this bit returns to 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: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh)	R
b7 to b4	IWDCKS[3:0]	IWDT Clock Frequency Division Ratio Select	b7 b4 0 0 0 0: x 1 (Cycle period: 131 ms) 0 0 1 0: x 1/16 (Cycle period: 2.10 s) 0 0 1 1: x 1/32 (Cycle period: 4.19 s) 0 1 0 0: x 1/64 (Cycle period: 8.39 s) 1 1 1 1: x 1/128 (Cycle period: 16.8 s) 0 1 0 1: x 1/256 (Cycle period: 33.6 s) Settings other than above are prohibited.	R
b9, b8	IWDRPES[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	IWDRPSS[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	IWDRSTIRQS	IWDT Reset Interrupt Request Select	0: Non-maskable interrupt request is enabled 1: Reset is enabled	R
b13	—	Reserved	When reading, this bit returns to 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, deep software standby, or all-module clock stop mode	R
b16, b15	—	Reserved	When reading, these bits return to the value written by the user. The write value should be 1.	R
b17	WDTSTRT	WDT Start Mode Select	0: WDT is automatically activated in auto-start mode after a reset 1: WDT is stopped after a reset	R



Bit	Symbol	Bit Name	Description	R/W
b19, b18	WDTTOPS[1:0]	WDT Timeout Period Select	b19 b18 0 0: 1024 cycles (03FFh) 0 1: 4096 cycles (0FFFh) 1 0: 8192 cycles (1FFFh) 1 1: 16384 cycles (3FFFh)	R
b23 to b20	WDTCKS[3:0]	WDT Clock Frequency Division Ratio Select	b23 b20 0 0 0 1: PCLK/4 0 1 0 0: PCLK/64 1 1 1 1: PCLK/128 0 1 1 0: PCLK/512 0 1 1 1: PCLK/2048 1 0 0 0: PCLK/8192 Settings other than above are prohibited.	R
b25, b24	WDRPES[1:0]	WDT Window End Position Select	b25 b24 0 0: 75% 0 1: 50% 1 0: 25% 1 1: 0% (No window end position setting)	R
b27, b26	WDRPSS[1:0]	WDT Window Start Position Select	b27 b26 0 0: 25% 0 1: 50% 1 0: 75% 1 1: 100% (No window start position setting)	R
b28	WDRSTIRQS	WDT Reset Interrupt Request Select	0: Non-maskable interrupt request is enabled 1: Reset is enabled	R
b31 to b29	—	Reserved	When reading, these bits return to the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. It is set to the written value after written by the user.

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 ineffective in user boot mode, and the value becomes 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 1024, 4096, 8192, or 16384 cycles of the frequency-divided clock set by the IWDTCKS[3:0] bits. The time (number of clock cycles for the IWDT) 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 30, 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 clock for the IWDT. Using the setting of these bits together with the IWDTTOPS[1:0] bit setting, the IWDT counting period can be set from 1024 to 4194304 clock cycles for the IWDT.

For details, see section 30, Independent Watchdog Timer (IWDTa).

### IWDRPES[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 30, 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 30, 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 30, Independent Watchdog Timer (IWDTa).

#### **IWDTSLCSTP Bit (IWDT Sleep Mode Count Stop Control)**

This bit selects to stop counting when entering sleep, software standby, deep software standby, or all-module clock stop mode.

For details, see section 30, Independent Watchdog Timer (IWDTa).

#### **WDTSTRT Bit (WDT Start Mode Select)**

This bit selects the mode in which the WDT 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 WDT is effective.

#### **WDTTOPS[1:0] Bits (WDT Timeout Period Select)**

These bits select the timeout period, i.e. the time it takes for the down-counter to underflow, as 1024, 4096, 8192, or 16384 cycles of the frequency-divided clock set by the WDTCKS[3:0] bits. The time (number of PCLK cycles) it takes to underflow after a refresh operation is determined by a combination of the WDTCKS[3:0] bits and WDTTOPS[1:0] bits.

For details, see section 29, Watchdog Timer (WDTA).

#### **WDTCKS[3:0] Bits (WDT Clock Frequency Division Ratio Select)**

These bits select, from 1/4, 1/64, 1/128, 1/512, 1/2048, and 1/8192, the division ratio of the prescaler to divide the frequency of PCLK. Using the setting of these bits together with the WDTTOPS[1:0] bit setting, the WDT counting period can be set from 4096 to 134217728 PCLK cycles.

For details, see section 29, Watchdog Timer (WDTA).

**WDTRPES[1:0] Bits (WDT Window End Position Select)**

These bits select the position of the end of the window on 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 WDTRPSS[1:0] and WDTRPES[1:0] bits vary with the setting of the WDTTOPS[1:0] bits.

For details, refer to section 29, Watchdog Timer (WDTA).

**WDTRPSS[1:0] Bits (WDT 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 29, Watchdog Timer (WDTA).

**WDTRSTIRQS Bit (WDT 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 a watchdog timer reset or a non-maskable interrupt request is selectable.

For details, refer to section 29, Watchdog Timer (WDTA).

## 7.2.2 Option Function Select Register 1 (OFS1)

Address(es): 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
—	—	—	—	—	—	—	HOCO EN	—	—	—	—	—	LVDAS	—	—

Value after reset: The value set by the user\*1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	When reading, these bits return to the value written by the user. The write value should be 1.	R
b2	LVDAS	Voltage Detection 0 Circuit Start	0: Voltage monitor 0 reset is enabled after a reset 1: Voltage monitor 0 reset is disabled after a reset	R
b7 to b3	—	Reserved	When reading, these bits return to 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 to the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. It is set to the written value after written by the user.

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 setting in the OFS1 register is ineffective, and the OFS1 register value becomes FFFF FFFFh.

The setting in the OFS1 register is ineffective in user boot mode, and the value becomes FFFF FFFFh.

### LVDAS Bit (Voltage Detection 0 Circuit Start)

This bit selects whether the voltage monitor 0 reset is enabled or disabled after a reset.

### HOCOEN Bit (HOCO Oscillation Enable)

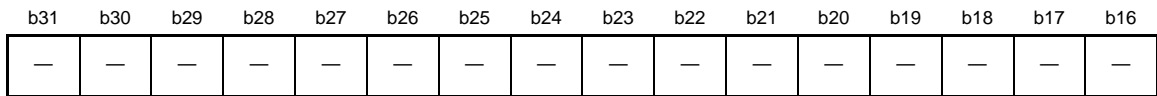
This bit selects whether the HOCO oscillation enable bit is effective or not after a reset.

Setting the HOCOEN bit to 0 allows the HOCO oscillation to be started before the CPU starts operation, and therefore reduces the waiting time for oscillation stabilization.

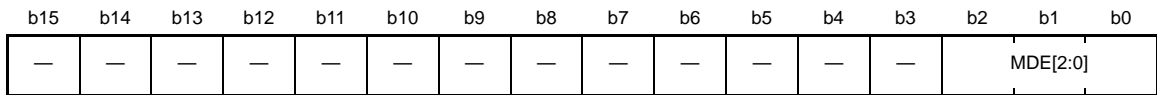
Note that even if the HOCOEN bit is set to 0, the system clock source is not switched to HOCO. The system clock source is switched to HOCO only by modifying the clock source select bits (SCKCR3.CKSEL[2:0]) from the CPU.

### 7.2.3 Endian Select Register B (MDEB), Endian Select Register S (MDES)

Address(es): FF7F FFF8h: MDEB (in user boot mode)  
 FFFF FF80h: MDES (in single-chip mode)



Value after reset: The value set by the user\*<sup>1</sup>



Value after reset: The value set by the user\*<sup>1</sup>

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, this bit returns to the value written by the user. The write value should be 1.	R

Note 1. The value of the blank product is FFFF FFFFh. It is set to the written value after written by the user.

The MDEn (n = B, S) register selects the endian for the CPU. In user boot mode, the endian select register B (MDEB) at address FF7F FFF8h is used to select the endian. In single-chip mode, the endian select register S (MDES) at address FFFF FF80h is used.

MDEn 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 MDEn register, the MDEn register value becomes FFFF FFFFh.

#### MDE[2:0] Bits (Endian Select)

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

The endian is determined by the value at address FF7F FFF8h in the user boot area when operating in user boot mode, and by the value at address FFFF FF80h in the user area when operating in single-chip mode.

### 7.3 UB Codes

UB codes A and B are required if user boot mode is to be employed. When the USB boot mode is used continuously, these codes should not be changed. The MCU will start up in user boot mode on release from the reset state if the four conditions below are satisfied.

- UB code A is 55736572h and 426F6F74h.
- UB code B is FFFF FF07h and 0008 C04Ch.
- The low level is being input on the MD pin.
- The high level is being input on the PC7 pin.

#### 7.3.1 UB Code A

UB code A consists of two 32-bit words. Set UB code A to 55736572h and 426F6F74h. Do not set other values for the code.

Figure 7.2 shows the structure of UB code A in memory. Set UB code A in 32-bit units.

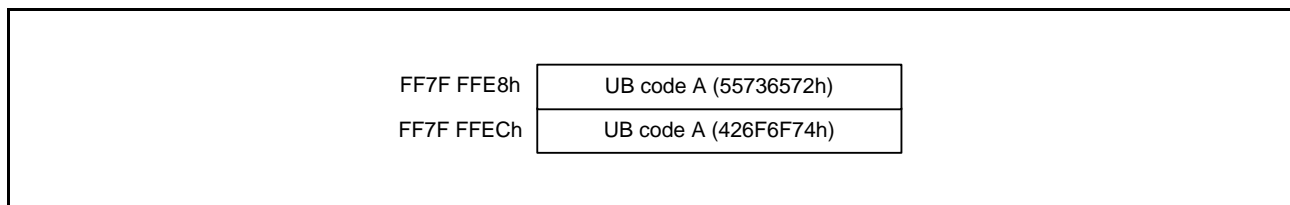


Figure 7.2 UB Code A Structure

#### 7.3.2 UB Code B

UB code B consists of two words, i.e. 32 bits. Set UB code B to FFFF FF07h and 0008 C04Ch. Do not set other values for the code.

Figure 7.3 shows the structure of UB code B in memory. Set UB code B in 32-bit units.

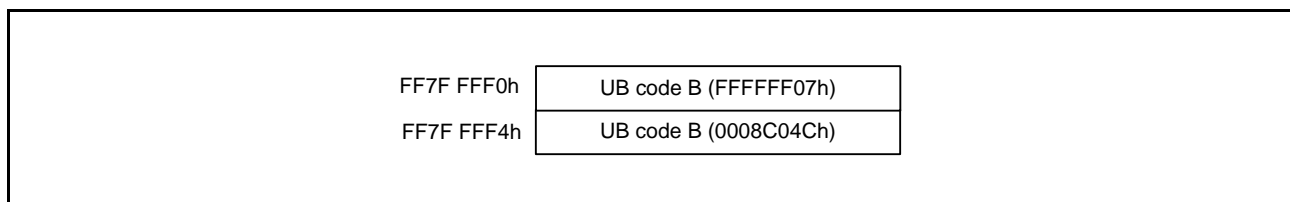


Figure 7.3 UB Code B Structure

### 7.4 Usage Note

#### 7.4.1 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 0fff ff8ch
.lword 0fffffff8h
```

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

## 8. Voltage Detection Circuit (LVDA)

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

### 8.1 Overview

In voltage detection 0, whether to enable or disable the reset of voltage monitoring 0 can be selected after the reset using the option function select register 1 (OFS1).

In voltage detection 1 and voltage detection 2, the detection voltage is set using the voltage detection level select register (LVDLVLR).

Reset of voltage monitoring 0, reset/interrupt of voltage monitoring 1, and reset/interrupt of voltage monitoring 2 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 Voltage Detection Circuit Specifications**

Item		Voltage Monitoring 0	Voltage Monitoring 1	Voltage Monitoring 2
VCC monitoring	Monitored voltage	Vdet0	Vdet1	Vdet2
	Detected event	Voltage drops past Vdet0	Voltage rises or drops past Vdet1	Voltage rises or drops past Vdet2
	Detection voltage	One level fixed	Specify voltage using LVDLVLR.LVD1LVLR[3:0] bits	Specify voltage using LVDLVLR.LVD2LVLR[3:0] bits
	Monitoring flag	None	LVD1SR.LVD1MON flag: Monitors whether voltage is higher or lower than Vdet1 LVD1SR.LVD1DET flag: Vdet1 passage detection	LVD2SR.LVD2MON flag: Monitors whether voltage is higher or lower than Vdet2 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 Vdet2 > VCC
	Interrupt	No interrupt	Voltage monitoring 1 interrupt Non-maskable interrupt Interrupt request issued when Vdet1 > VCC and VCC > Vdet1 or either	Voltage monitoring 2 interrupt Non-maskable interrupt Interrupt request issued when Vdet2 > VCC and VCC > Vdet2 or either
Digital filter	Enable/Disable switching	Digital filter function not available	Available	Available
	Sampling time	—	1/n LOCO frequency × 2 (n: 1, 2, 4, 8)	1/n LOCO frequency × 2 (n: 1, 2, 4, 8)

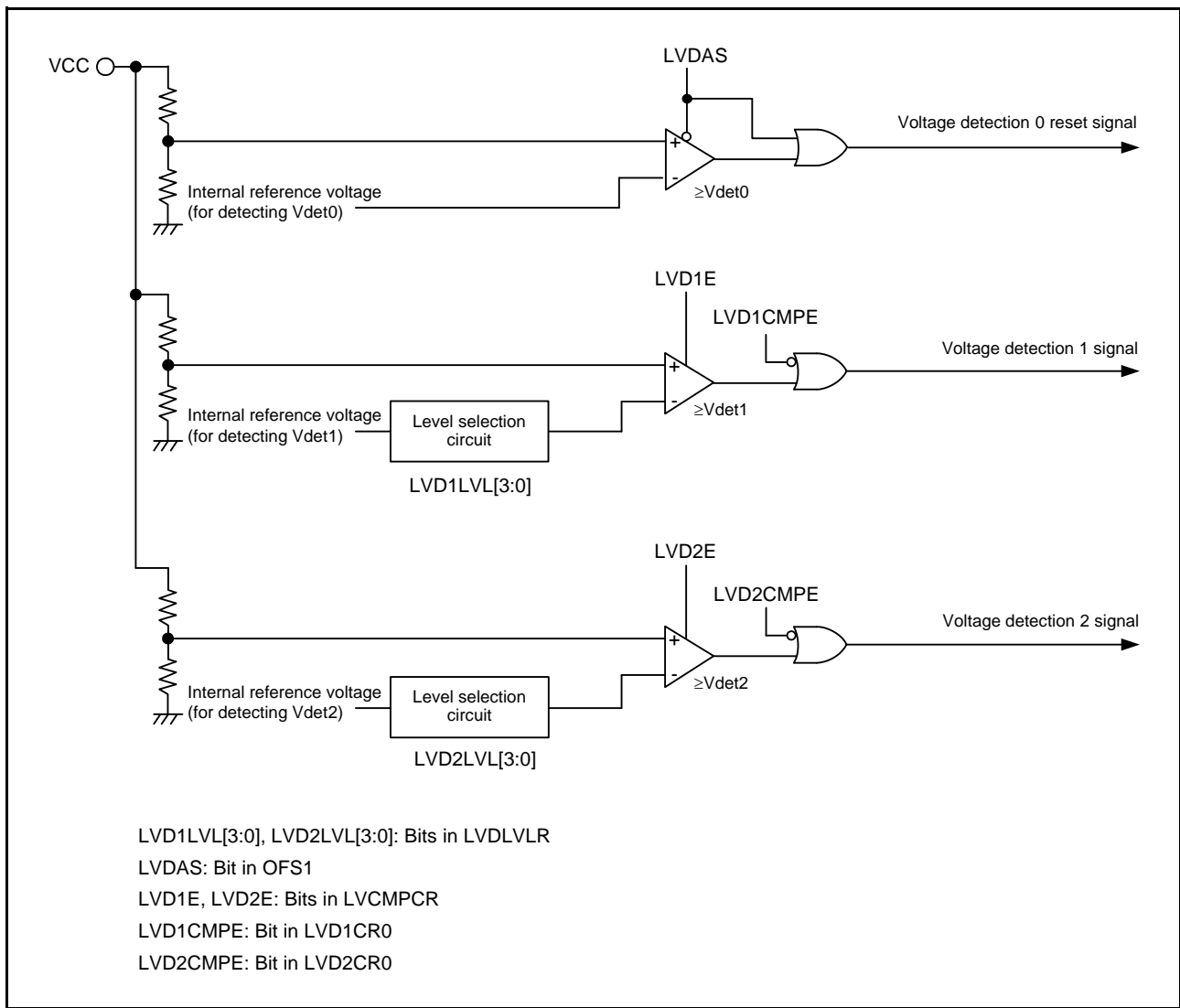


Figure 8.1 Block Diagram of Voltage Detection Circuit



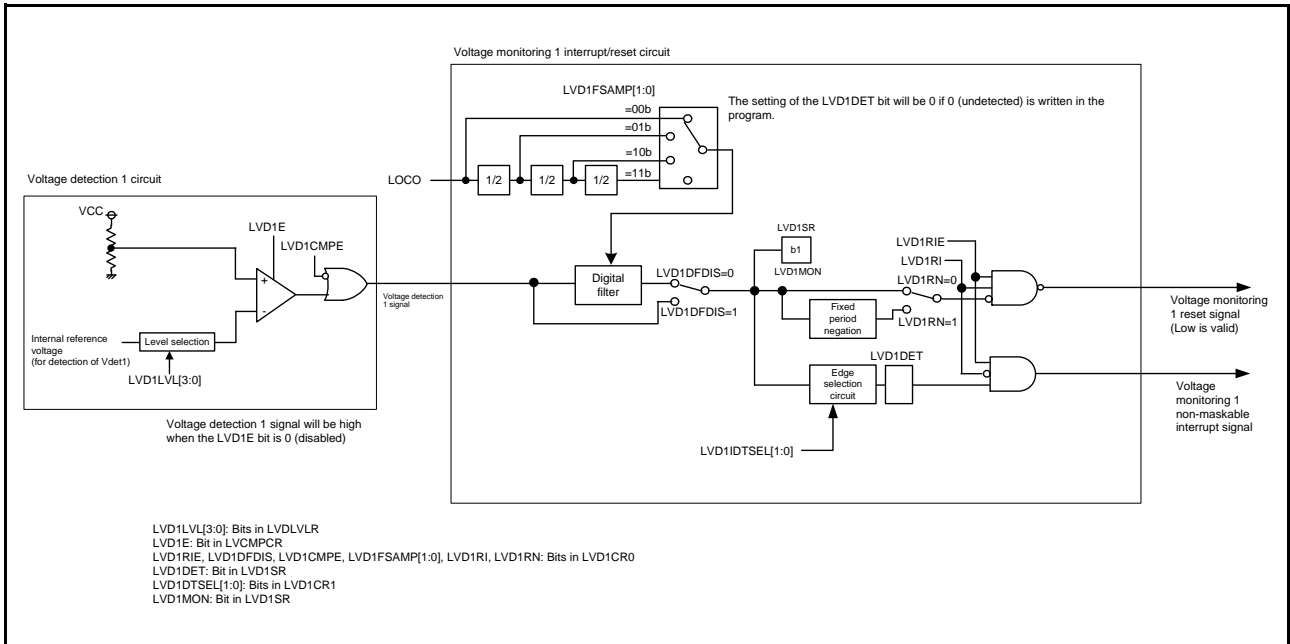


Figure 8.2 Block Diagram of Voltage Monitoring 1 Interrupt/Reset Circuit

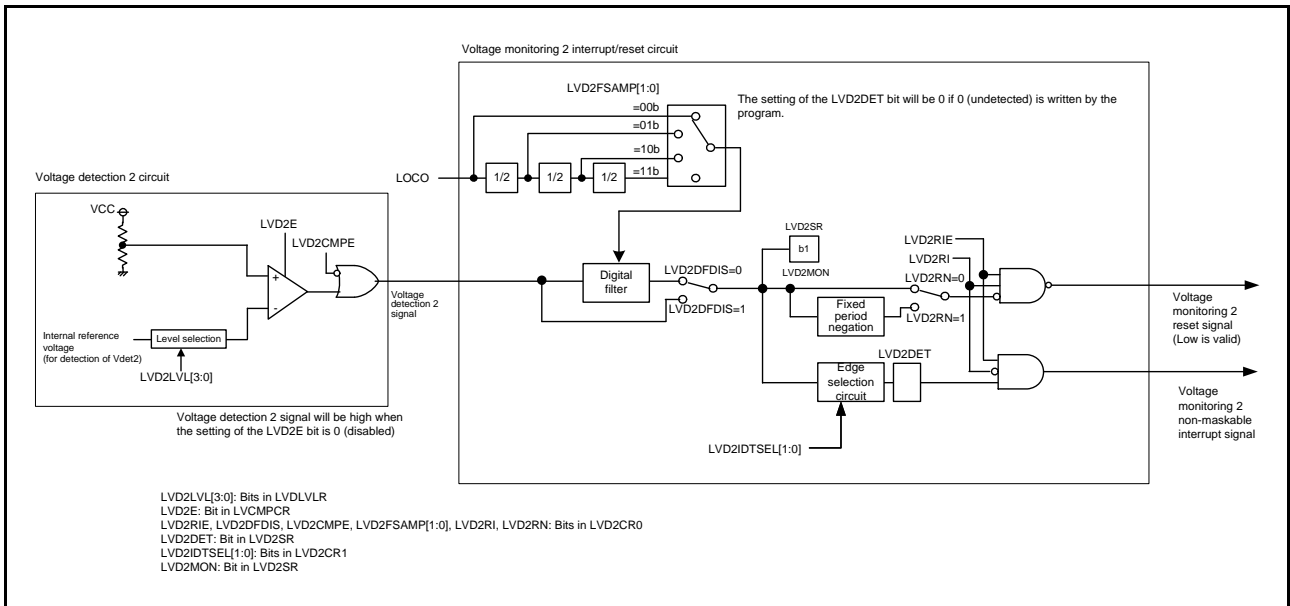
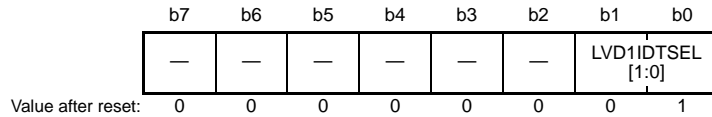


Figure 8.3 Block Diagram of Voltage Monitoring 2 Interrupt/Reset Circuit

## 8.2 Register Descriptions

### 8.2.1 Voltage Monitoring 1 Circuit Control Register 1 (LVD1CR1)

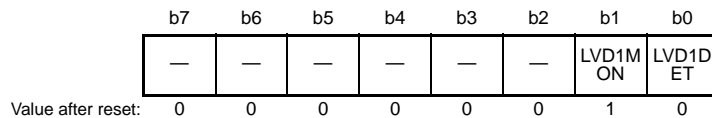
Address(es): 0008 00E0h



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 : Settings prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

### 8.2.2 Voltage Monitoring 1 Circuit Status Register (LVD1SR)

Address(es): 0008 00E1h



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 is disabled	R
b7 to b2	—	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. After writing 0 to this bit, it takes 2 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).

When LVD1CR0 and LVD1CR1 have been modified, the LVD1DET flag may become 1.

The LVD1DET flag should be set to 0 after LVD1CR0.LVD1RIE is set to 0 (disabled). LVD1CR0.LVD1RIE can be set to 1 (enabled) after a period of two or more cycles of PCLKB has elapsed.

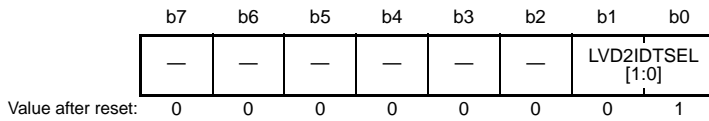
Depending on the number of cycles of PCLKB defined for access to read an I/O register, two or more cycles than PCLKB may have to be secured as waiting time.

#### 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)

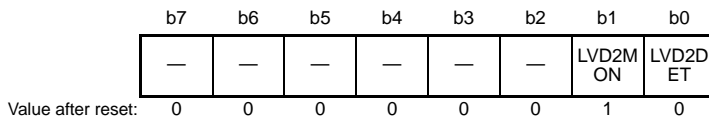
Address(es): 0008 00E2h



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: Settings prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

### 8.2.4 Voltage Monitoring 2 Circuit Status Register (LVD2SR)

Address(es): 0008 00E3h



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 1. Only 0 can be written to this bit. After writing 0 to this bit, it takes 2 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 LVCMPPCR.LVD2E bit is 1 (voltage detection 2 circuit enabled) and the LVD2CR0.LVD2CMPE bit is 1 (voltage monitoring 2 circuit comparison result output enabled).

When LVD2CR0 and LVD2CR1 have been modified, the LVD2DET flag may become 1.

The LVD2DET flag should be set to 0 after LVD2CR0.LVD2RIE is set to 0 (disabled). LVD2CR0.LVD2RIE can be set to 1 (enabled) after a period of two or more cycles of PCLKB has elapsed.

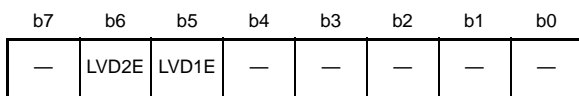
Depending on the number of cycles of PCLKB defined for access to read an I/O register, two or more cycles of PCLKB may have to be secured as waiting time.

#### LVD2MON Flag (Voltage Monitoring 2 Signal Monitor Flag)

The LVD2MON flag is enabled when the LVCMPPCR.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



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

#### LVD1E Bit (Voltage Detection 1 Enable)

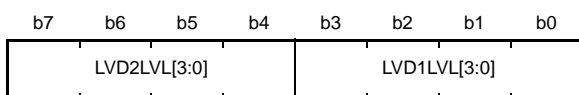
When using voltage detection 1 interrupt/reset or the LVD1SR.LVD1MON bit, set the LVD1E bit to 1. The voltage detection 1 circuit starts once td(E-A) passes after the LVD1E bit value is changed from 0 to 1. When using the voltage detection 1 circuit in deep software standby mode, do not set the DPSBYCR.DEEPCUT[1:0] bits to 11b.

#### LVD2E Bit (Voltage Detection 2 Enable)

When using voltage detection 2 interrupt/reset or the LVD2SR.LVD2MON bit, 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. When using the voltage detection 2 circuit in deep software standby mode, do not set the DPSBYCR.DEEPCUT[1:0] bits to 11b.

### 8.2.6 Voltage Detection Level Select Register (LVDLVLR)

Address(es): 0008 C298h



Value after reset: 1 x x x 1 x x x

x: Undefined

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 b2 b1 b0 1 0 1 0: 2.95 V Do not set otherwise.	R/W
b7 to b4	LVD2LVL[3:0]	Voltage Detection 2 Level Select (Standard voltage during drop in voltage)	b7 b6 b5 b4 1 0 1 0: 2.95 V Do not set otherwise.	R/W

The contents of the LVDLVLR register can only be changed if the LVCMPCR.LVD1E and LVCMPCR.LVD2E bits (voltage detection n circuit disable; n = 1, 2) are both 0.

## 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	LVD1FSAMP [1:0]	—	LVD1CMPE	LVD1DFDIS	LVD1RIE	
Value after reset:	1	0	0	0	x	0	1	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	LVD1DFDIS	Voltage Monitoring 1 Digital Filter Disable Mode Select	0: Digital filter enabled 1: Digital filter disabled	R/W
b2	LVD1CMPE	Voltage Monitoring 1 Circuit Comparison Result Output Enable	0: Voltage monitoring 1 circuit comparison result output disabled. 1: Voltage monitoring 1 circuit comparison result output enabled.	R/W
b3	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b5, b4	LVD1FSAMP [1:0]	Sampling Clock Select	b5 b4 0 0: 1/1 LOCO frequency 0 1: 1/2 LOCO frequency 1 0: 1/4 LOCO frequency 1 1: 1/8 LOCO frequency	R/W
b6	LVD1RI	Voltage Monitoring 1 Circuit Mode Select	0: Voltage monitoring 1 interrupt enabled when Vdet1 is crossed 1: Voltage monitoring 1 reset enabled when the voltage falls to and below Vdet1	R/W
b7	LVD1RN	Voltage Monitoring 1 Reset Negate Select	0: Negation follows a stabilization time (tLVD1) after VCC > Vdet1 is detected. 1: Negation follows a stabilization time (tLVD1) after assertion of the LVD1 reset.	R/W

### LVD1RIE Bit (Voltage Monitoring 1 Interrupt/Reset Enable)

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

### LVD1DFDIS Bit (Voltage Monitoring 1 Digital Filter Disable Mode Select)

Set the LOCOCR.LCSTP bit to 0 (the LOCO operates) if the LVD1DFDIS bit is 0 (enabling the digital filter circuit). Set the LVD1DFDIS bit to 1 (digital filter circuit disabled) when using voltage monitoring 1 circuit in software standby mode or deep software standby mode.

### LVD1FSAMP[1:0] Bits (Sampling Clock Select)

The LVD1FSAMP[1:0] bits can be modified only when the LVD1DFDIS bit is 1 (digital filter circuit disabled). The LVD1FSAMP[1:0] bits should not be modified when the LVD1DFDIS bit is 0 (digital filter circuit disabled).

### LVD1RI Bit (Voltage Monitoring 1 Circuit Mode Select)

When the LVD1RI bit is 1 (voltage monitoring 1 reset enabled) or when the LVD2CR0.LVD2RI bit is 1 (voltage monitoring 2 reset enabled), a transition to deep software standby mode cannot be made, instead a transition to software standby mode is made. To enter deep software standby mode, set the LVD1RI bit to 0 (voltage monitoring 1 interrupt enabled) and the LVD2CR0.LVD2RI bit to 0 (voltage monitoring 2 interrupt enabled).

**LVD1RN Bit (Voltage Monitoring 1 Reset Negate Select)**

If the LVD1RN bit is to be set to 1 (negation follows a stabilization time after assertion of the LVD1 reset signal), set the LOCOCR.LCSTP bit to 0 (the LOCO operates). Furthermore, if a transition to software standby or deep software standby is to be made, 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 LVD1 reset signal) when this is the case.

**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	LVD2FSAMP[1:0]	—	LVD2CMPE	LVD2DFDIS	LVD2RIE	
Value after reset:	1	0	0	0	x	0	1

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	LVD2RIE	Voltage Monitoring 2 Interrupt/Reset Enable	0: Disabled 1: Enabled	R/W
b1	LVD2DFDIS	Voltage Monitoring 2 Digital Filter Disable Mode Select	0: Digital filter enabled 1: Digital filter disabled	R/W
b2	LVD2CMPE	Voltage Monitoring 2 Circuit Comparison Result Output Enable	0: Voltage monitoring 2 circuit comparison result output disabled. 1: Voltage monitoring 2 circuit comparison result output enabled.	R/W
b3	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b5, b4	LVD2FSAMP[1:0]	Sampling Clock Select	b5 b4 0 0: 1/1 LOCO frequency 0 1: 1/2 LOCO frequency 1 0: 1/4 LOCO frequency 1 1: 1/8 LOCO frequency	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 Negate Select	0: Negation follows a stabilization time (tLVD2) after VCC > Vdet2 is detected. 1: Negation follows a stabilization time (tLVD2) after assertion of the LVD2 reset.	R/W

**LVD2RIE Bit (Voltage Monitoring 2 Interrupt/Reset Enable)**

Ensure that neither a voltage monitoring 2 reset nor a voltage monitoring 2 interrupt is generated during programming or erasure of the flash memory.

**LVD2DFDIS Bit (Voltage Monitoring 2 Digital Filter Disable Mode Select)**

Set the LOCOCR.LCSTP bit to 0 (the LOCO operates) if the LVD1DFDIS bit is 0 (enabling the digital filter circuit). Set the LVD2DFDIS bit to 1 (digital filter circuit disabled) when using voltage monitoring 2 circuit in software standby mode or deep software standby mode.

**LVD2FSAMP[1:0] Bits (Sampling Clock Select)**

The LVD2FSAMP[1:0] bits can be modified only when the LVD2DFDIS bit is 1 (digital filter circuit disabled). The LVD2FSAMP[1:0] bits should not be modified when the LVD2DFDIS bit is 0 (digital filter circuit enabled).

**LVD2RI Bit (Voltage Monitoring 2 Circuit Mode Select)**

When the LVD2RI bit is 1 (voltage monitoring 2 reset enabled) or when the LVD1CR0.LVD1RI bit is 1 (voltage monitoring 1 reset enabled), a transition to deep software standby mode cannot be made, instead a transition to software standby mode is made. To enter to deep software standby mode, set the LVD2RI bit to 0 (voltage monitoring 2 interrupt enabled) and the LVD1CR0.LVD1RI bit to 0 (voltage monitoring 1 interrupt enabled).

**LVD2RN Bit (Voltage Monitoring 2 Reset Negate Select)**

If the LVD2RN bit is to be set to 1 (negation follows a stabilization time after assertion of the LVD2 reset signal), set the LOCOCR.LCSTP bit to 0 (the LOCO operates). Furthermore, if a transition to software standby or deep software standby is to be made, the only possible value for the LVD2RN bit is 0 (negation follows a stabilization time after  $VCC > V_{det2}$  is detected). Do not set the LVD2RN bit to 1 (negation follows a stabilization time after assertion of the LVD2 reset signal) when this is the case.

### 8.3 VCC Input Voltage Monitor

#### 8.3.1 Monitoring Vdet0

Monitoring Vdet0 is not possible.

#### 8.3.2 Monitoring Vdet1

Table 8.2 lists the procedures for setting up monitoring against Vdet1. After the settings are completed, results of comparison by voltage monitoring 1 can be monitored by using the LVD1SR.LVD1MON flag.

**Table 8.2 Procedures for Setting up Monitoring against Vdet1**

Step	When the Digital Filter is in Use	When the Digital Filter is Not in Use
1	Specify the detection voltage by setting the LVDLVLR.LVD1LVL[3:0] bits (voltage detection 1 level select).	
2	Select the sampling clock for the digital filter by setting the LVD1CR0.LVD1FSAMP[1:0] bits.	Set the LVD1CR0.LVD1DFDIS bit to 1 (disabling the digital filter).
3	Set the LVD1CR0.LVD1CMPE bit to 1 (enabling output of the results of comparison by voltage monitor 1).	
4	Wait for at least one cycle of the LOCO.	—
5	Clear the LVD1CR0.LVD1DFDIS bit to 0 (enabling the digital filter).	—
6	Wait for at least $2n - 3$ cycles of the LOCO (where $n = 1, 2, 4, 8$ , and the sampling clock for the digital filter is the LOCO frequency-divided by $n$ ).	—
7	Set the LVCMPCR.LVD1E bit to 1 (enabling the voltage detection 1 circuit).	

#### 8.3.3 Monitoring Vdet2

Table 8.3 lists the procedures for setting up monitoring against Vdet2. After the settings are completed, results of comparison by voltage monitoring 2 can be monitored by using the LVD2SR.LVD2MON flag.

**Table 8.3 Procedures for Setting up Monitoring against Vdet2**

Step	When the Digital Filter is in Use	When the Digital Filter is Not in Use
1	Specify the detection voltage by setting the LVDLVLR.LVD2LVL[3:0] bits (voltage detection 2 level select).	
2	Select the sampling clock for the digital filter by setting the LVD2CR0.LVD2FSAMP[1:0] bits.	Set the LVD2CR0.LVD2DFDIS bit to 1 (disabling the digital filter).
3	Set the LVD2CR0.LVD2CMPE bit to 1 (enabling output of the results of comparison by voltage monitor 2).	
4	Wait for at least one cycle of the LOCO.	—
5	Clear the LVD2CR0.LVD2DFDIS bit to 0 (enabling the digital filter).	—
6	Wait for at least $2n - 3$ cycles of the LOCO (where $n = 1, 2, 4, 8$ , and the sampling clock for the digital filter is the LOCO frequency-divided by $n$ ).	—
7	Set the LVCMPCR.LVD2E bit to 1 (enabling the voltage detection 2 circuit).	



### 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.

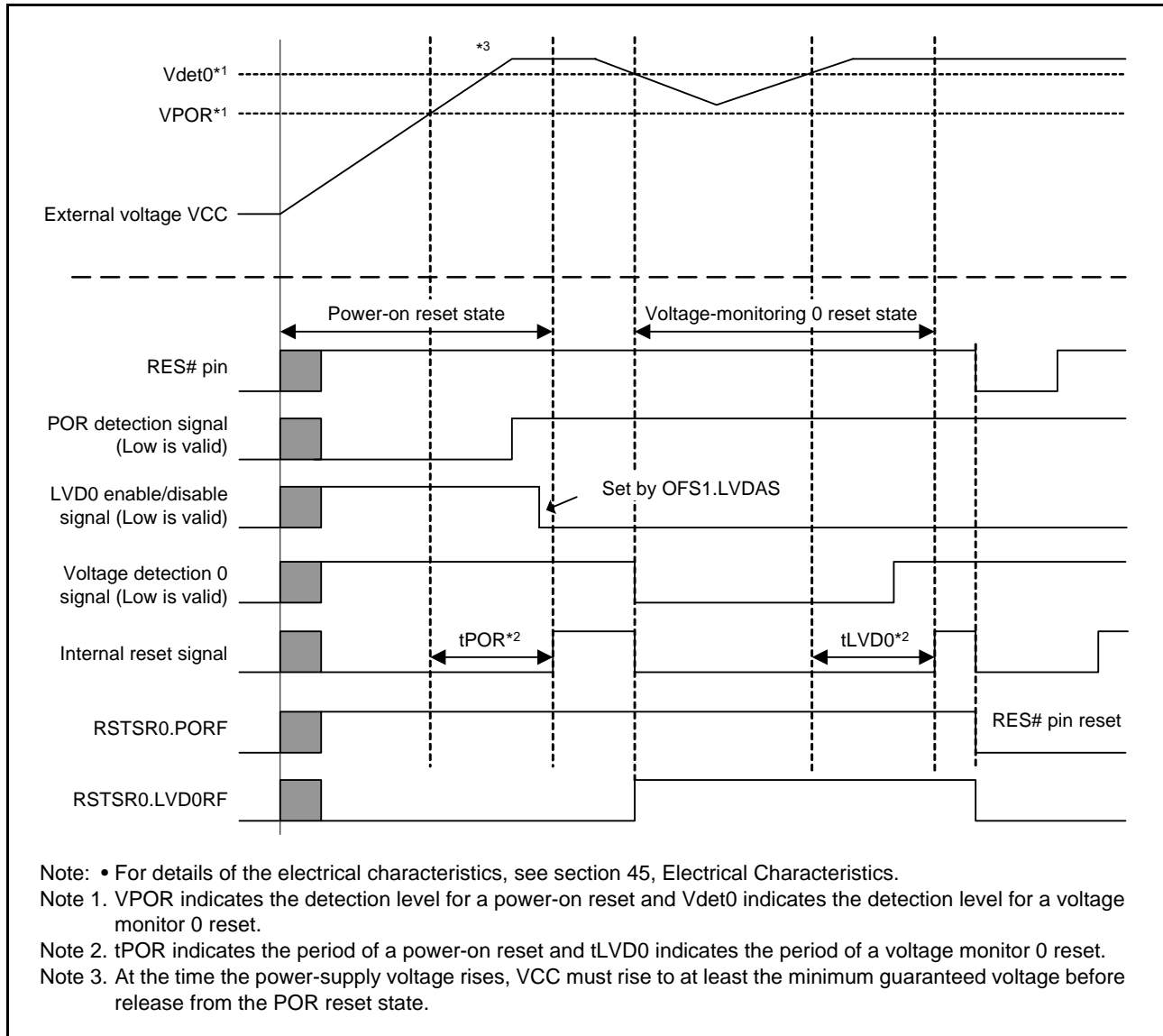


Figure 8.4 Example of Voltage Monitoring 0 Reset Operation

## 8.5 Interrupt and Reset from Voltage Monitor 1

Table 8.4 lists the procedures for setting bits related to the voltage monitor 1 interrupt and voltage monitor 1 reset so that voltage monitoring operates. Table 8.5 shows the procedure for setting bits related to the voltage monitor 1 interrupt and voltage monitor 1 reset so that voltage monitoring stops. Figure 8.5 shows an example of operations for a voltage monitor 1 interrupt. For the operation of the voltage monitor 1 reset, see Figure 6.2 in section 6, Resets.

Furthermore, set the LVD1CR0.LVD1DFDIS bit to 1 (disabling the digital filter) if you intend to use the voltage monitor 1 circuit in software standby or deep software standby mode.

**Table 8.4 Procedures for Setting Bits Related to the Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset so that Voltage Monitoring Operates**

Step	When the Digital Filter is in Use		When the Digital Filter is Not in Use	
	Voltage Monitor 1 Interrupt	Voltage Monitor 1 Reset	Voltage Monitor 1 Interrupt	Voltage Monitor 1 Reset
1*1	Specify the detection voltage by setting the LVDLVLR.LVD1LVL[3:0] bits.			
2*2	Select the sampling clock for the digital filter by setting the LVD1CR0.LVD1FSAMP[1:0] bits.		Set the LVD1CR0.LVD1DFDIS bit to 1 (disabling the digital filter).	
3 *1, *2	Clear the LVD1CR0.LVD1RI bit to 0 (selecting the voltage monitor 1 interrupt).	<ul style="list-style-type: none"> <li>Set the LVD1CR0.LVD1RI bit to 1 (selecting the voltage monitor 1 reset).</li> <li>Select the type of reset negation by setting the LVD1CR0.LVD1RN bit.</li> </ul>	Clear the LVD1CR0.LVD1RI bit to 0 (selecting the voltage monitor 1 interrupt).	<ul style="list-style-type: none"> <li>Set the LVD1CR0.LVD1RI bit to 1 (selecting the voltage monitor 1 reset).</li> <li>Select the type of the reset negation by setting the LVD1CR0.LVD1RN bit.</li> </ul>
4	Select the timing of interrupt requests by setting the LVD1CR1.LVD1IDTSEL[1:0] bits.	—	Select the timing of interrupt requests by setting the LVD1CR1.LVD1IDTSEL[1:0] bits.	—
5	Set the LVD1CR0.LVD1CMPE bit to 1 (enabling output of the results of comparison by voltage monitor 1).			
6	Wait for at least one cycle of the LOCO.		—	
7	Clear the LVD1CR0.LVD1DFDIS bit to 0 (enabling the digital filter).		—	
8	Wait for at least $2n + 3$ cycles of the LOCO (where $n = 1, 2, 4, 8$ , and the sampling clock for the digital filter is the LOCO frequency-divided by $n$ ).		— (waiting is not required)	
9	Clear the LVD1SR.LVD1DET flag to 0.	—	Clear the LVD1SR.LVD1DET flag to 0.	—
10	Set the LVD1CR0.LVD1RIE bit to 1 (enabling the voltage monitor 1 interrupt or reset).			
11*1	Set the LVCMPCR.LVD1E bit to 1 (enabling the voltage detection 1 circuit).			

Note 1. Steps 1, 3, and 11 are not required if operation is with the setting to select the voltage monitor 1 interrupt (LVD1CR0.LVD1RI = 0) and operation can be restarted by simply changing the settings of the LVD1CR0.LVD1FSAMP[1:0] and LVD1DFDIS bits or of the LVD1CR1.LVD1IDTSEL[1:0] bits, 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 monitor 1 reset (LVD1CR0.LVD1RI = 1), proceed through all steps from 1 to 11.

Note 2. Executing steps 2 and 3 at the same time (with a single instruction) creates no problems.

**Table 8.5 Procedures for Setting Bits Related to the Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset to Stop Voltage Monitoring**

Step	Setting Bits Related to the Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset to Stop Voltage Monitoring
1*1	Clear the LVCMPCR.LVD1E bit to 0 (disabling the voltage detection 1 circuit).
2*1	Wait for at least one cycle of the LOCO.
3	Clear the LVD1CR0.LVD1RIE bit to 0 (disabling the voltage monitor 1 interrupt or reset).
4	Clear the LVD1CR0.LVD1CMPE bit to 0 (disabling output of the results of comparison by voltage monitor 1).
5	Modify settings of bits related to the voltage detection circuit other than the LVCMPCR.LVD1E, LVD1CR0.LVD1CMPE, and LVD1RIE bits.

Note 1. Steps 1 and 2 are not required when operation is with the setting to select the voltage monitor 1 interrupt (LVD1CR0.LVD1RI = 0) and, after it is stopped, operation is to be restarted by simply changing the settings of the LVD1CR0.LVD1FSAMP[1:0] and LVD1DFDIS bits or the LVD1CR1.LVD1IDTSEL[1:0] bits, or when 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 monitor 1 reset (LVD1CR0.LVD1RI = 1), proceed through all steps from 1 to 5.

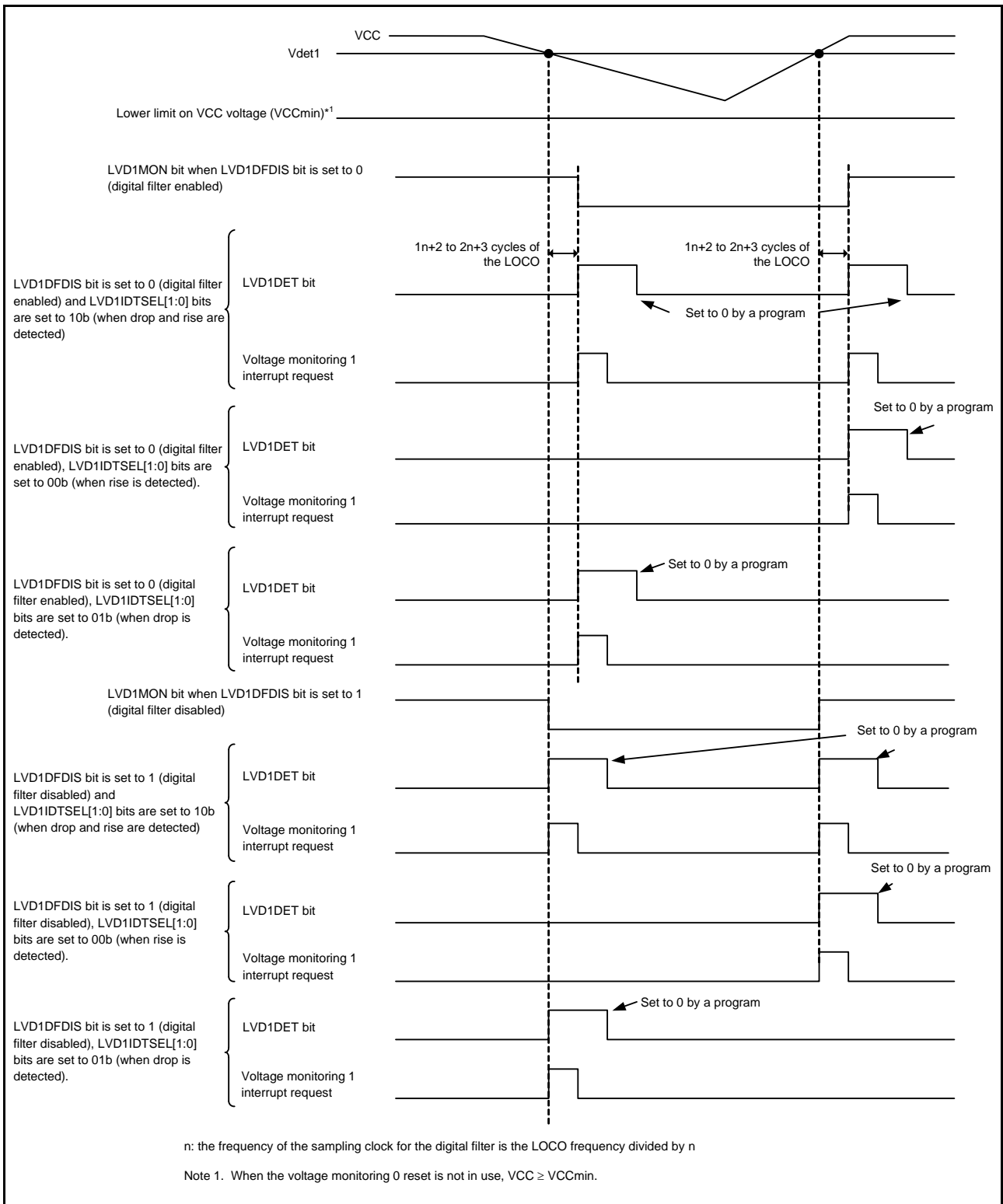


Figure 8.5 Example of Voltage Monitoring 1 Interrupt Operation

## 8.6 Interrupt and Reset from Voltage Monitor 2

Table 8.6 shows the procedures for setting bits related to the voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitoring operates. Table 8.7 shows the procedure for setting bits related to the voltage monitor 2 interrupt and voltage monitor 2 reset so that voltage monitoring stops. Figure 8.6 shows an example of operations for a voltage monitor 2 interrupt. For the operation of the voltage monitor 2 reset, see Figure 6.2 in section 6, Resets.

Furthermore, set the LVD2CR0.LVD2DFDIS bit to 1 (disabling the digital filter) if you intend to use the voltage monitor 2 circuit in software standby or deep software standby mode.

**Table 8.6 Procedures for Setting Bits Related to the Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset so that Voltage Monitoring Operates**

Step	When the Digital Filter is in Use		When the Digital Filter is Not in Use	
	Voltage Monitor 2 Interrupt	Voltage Monitor 2 Reset	Voltage Monitor 2 Interrupt	Voltage Monitor 2 Reset
1*1	Specify the detection voltage by setting the LVDLVLR.LVD2LVL[3:0] bits.			
2*2	Select the sampling clock for the digital filter by setting the LVD2CR0.LVD2FSAMP[1:0] bits.		Set the LVD2CR0.LVD2DFDIS bit to 1 (disabling the digital filter).	
3 *1,*2	Clear the LVD2CR0.LVD2RI bit to 0 (selecting the voltage monitor 2 interrupt).	<ul style="list-style-type: none"> <li>Set the LVD2CR0.LVD2RI bit to 1 (selecting the voltage monitor 2 reset).</li> <li>Select the type of reset negation by setting the LVD2CR0.LVD2RN bit.</li> </ul>	Clear the LVD2CR0.LVD2RI bit to 0 (selecting the voltage monitor 2 interrupt).	<ul style="list-style-type: none"> <li>Set the LVD2CR0.LVD2RI bit to 1 (selecting the voltage monitor 2 reset).</li> <li>Select the type of the reset negation by setting the LVD2CR0.LVD2RN bit.</li> </ul>
4	Select the timing of interrupt requests by setting the LVD2CR1.LVD2IDTSEL[1:0] bits.	—	Select the timing of interrupt requests by setting the LVD2CR1.LVD2IDTSEL[1:0] bits.	—
5	Set the LVD2CR0.LVD2CMPE bit to 1 (enabling output of the results of comparison by voltage monitor 2).			
6	Wait for at least one cycle of the LOCO.		—	
7	Clear the LVD2CR0.LVD2DFDIS bit to 0 (enabling the digital filter).		—	
8	Wait for at least $2n + 3$ cycles of the LOCO (where $n = 1, 2, 4, 8$ , and the sampling clock for the digital filter is the LOCO frequency-divided by $n$ ).		— (waiting is not required)	
9	Clear the LVD2SR.LVD2DET flag to 0.	—	Clear the LVD2SR.LVD2DET flag to 0.	—
10	Set the LVD2CR0.LVD2RIE bit to 1 (enabling the voltage monitor 2 interrupt or reset).			
11*1	Set the LVCMPCR.LVD2E bit to 1 (enabling the voltage detection 2 circuit).			

Note 1. Steps 1, 3, and 11 are not required if operation is with the setting to select the voltage monitor 2 interrupt (LVD2CR0.LVD2RI = 0) and operation can be restarted by simply changing the settings of the LVD2CR0.LVD2FSAMP[1:0] and LVD2DFDIS bits or of the LVD2CR1.LVD2IDTSEL[1:0] bits, 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 monitor 2 reset (LVD2CR0.LVD2RI = 1), proceed through all steps from 1 to 11.

Note 2. Executing steps 2 and 3 at the same time (with a single instruction) creates no problems.

**Table 8.7 Procedures for Setting Bits Related to the Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset to Stop Voltage Monitoring**

Step	Setting Bits Related to the Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset to Stop Voltage Monitoring
1*1	Clear the LVCMPCR.LVD2E bit to 0 (disabling the voltage detection 2 circuit).
2*1	Wait for at least one cycle of the LOCO.
3	Clear the LVD2CR0.LVD2RIE bit to 0 (disabling the voltage monitor 2 interrupt or reset).
4	Clear the LVD2CR0.LVD2CMPE bit to 0 (disabling output of the results of comparison by voltage monitor 2).
5	Modify settings of bits related to the voltage detection circuit other than the LVCMPCR.LVD2E, LVD2CR0.LVD2CMPE, and LVD2RIE bits.

Note 1. Steps 1 and 2 are not required when operation is with the setting to select the voltage monitor 2 interrupt (LVD2CR0.LVD2RI = 0) and, after it is stopped, operation is to be restarted by simply changing the settings of the LVD2CR0.LVD2FSAMP[1:0] and LVD2DFDIS bits or the LVD2CR1.LVD2IDTSEL[1:0] bits, or when 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 monitor 2 reset (LVD1CR0.LVD1RI = 1), proceed through all steps from 1 to 5.

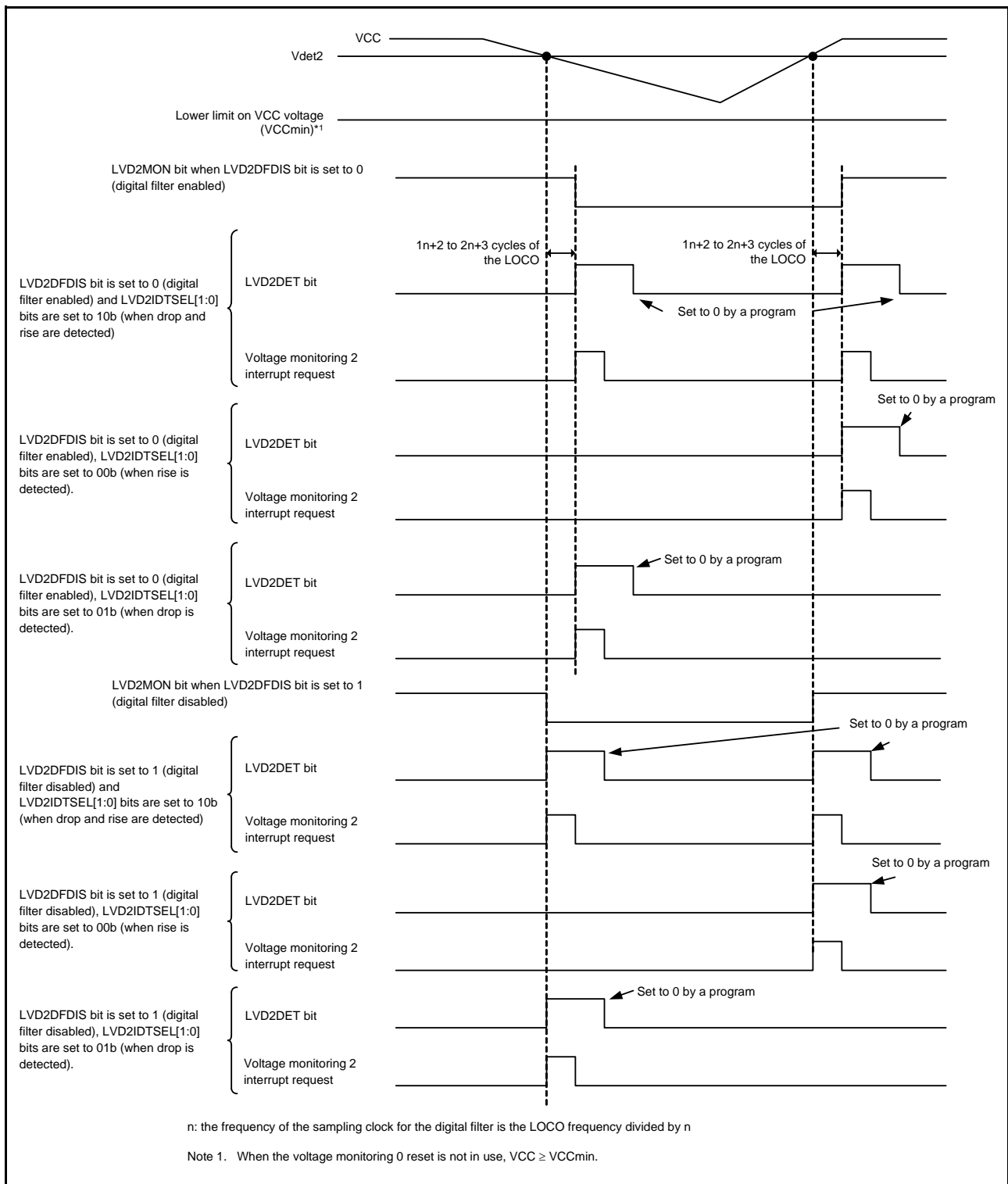


Figure 8.6 Example of Voltage Monitoring 2 Interrupt Operation

## 9. Clock Generation Circuit

### 9.1 Overview

The RX630 Group 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
Use	<ul style="list-style-type: none"> <li>Generates the system clock (ICLK) to be supplied to the CPU, DMAC, DTC, ROM, and RAM.</li> <li>Generates the peripheral module clock (PCLKB) to be supplied to peripheral modules.*1</li> <li>Generates the FlashIF clock (FCLK) to be supplied to the FlashIF.</li> <li>Generates the external bus clock (BCLK) to be supplied to the external bus.</li> <li>Generates the USB clock (UCLK) to be supplied to the USB.</li> <li>Generates the CAN clock (CANMCLK) to be supplied to the CAN.</li> <li>Generates the IEBUS clock (IECLK) to be supplied to the IEBUS.</li> <li>Generates the RTC-dedicated sub clock (RTCSCLK) to be supplied to the RTC.</li> <li>Generates the RTC-dedicated main clock (RTCMCLK) to be supplied to the RTC.</li> <li>Generates the IWDT-dedicated clock (IWDTCCLK) to be supplied to the IWDT.</li> <li>Generates the JTAG-dedicated clock (JTAGTCK) to be supplied to the JTAG.</li> </ul>
Operating frequency	<ul style="list-style-type: none"> <li>ICLK: 100 MHz (max)</li> <li>PCLKB: 50 MHz (max)</li> <li>FCLK: 4 MHz to 50 MHz (for programming and erasing the ROM and E2 DataFlash) 50 MHz (max) (for reading from the E2 DataFlash)</li> <li>BCLK: 50 MHz (max)</li> <li>BCLK pin output: 25 MHz (max)</li> <li>UCLK: 48 MHz (max)</li> <li>CANMCLK: 20 MHz (max)</li> <li>IECLK: 50 MHz (max)</li> <li>RTCSCLK: 32.768 kHz</li> <li>RTCMCLK: 4 MHz to 16 MHz</li> <li>IWDTCCLK: 125 kHz</li> <li>JTAGTCK: 10 MHz (max)</li> </ul>
Main clock oscillator	<ul style="list-style-type: none"> <li>Resonator frequency: 4 MHz to 16 MHz</li> <li>External clock input frequency: 20 MHz (max)</li> <li>Connectable resonator or additional circuit: ceramic resonator, crystal resonator</li> <li>Connection pin: EXTAL, XTAL</li> <li>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.</li> </ul>
Sub-clock oscillator	<ul style="list-style-type: none"> <li>Resonator frequency: 32.768 kHz</li> <li>Connectable resonator or additional circuit: crystal resonator</li> <li>Connection pin: XCIN, XCOUT</li> </ul>
PLL circuit	<ul style="list-style-type: none"> <li>Input clock source: Main clock</li> <li>Input pulse frequency division ratio: Selectable from 1, 2, and 4</li> <li>Input frequency: 4 MHz to 16 MHz</li> <li>Frequency multiplication ratio: Selectable from 8, 10, 12, 16, 20, 24, 25, and 50</li> <li>VCO oscillation frequency: 104 MHz to 200 MHz</li> </ul>
High-speed on-chip oscillator (HOCO)	<ul style="list-style-type: none"> <li>Oscillation frequency: 50 MHz</li> <li>HOCO power supply control</li> </ul>
Low-speed on-chip oscillator (LOCO)	Oscillation frequency: 125 kHz
IWDT-dedicated on-chip oscillator	Oscillation frequency: 125 kHz
External clock input (TCK) for JTAG	Input clock frequency: 10 MHz (max)
Control of output on the BCLK pin	<ul style="list-style-type: none"> <li>BCLK clock output or high output is selectable</li> <li>BCLK or BCLK/2 is selectable</li> </ul>

Note 1. In the RX630 Group, PCLKB = PCLK.

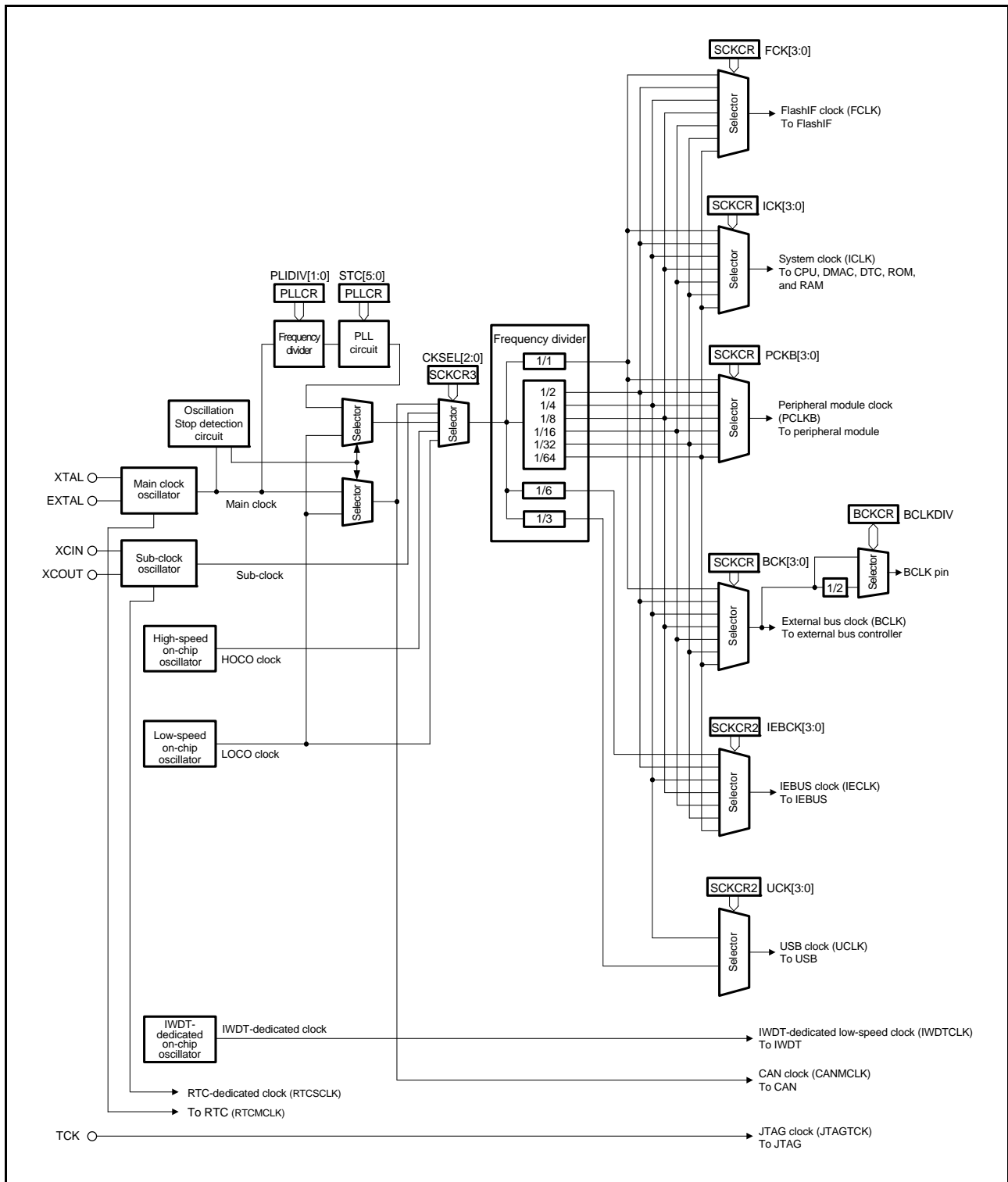


Figure 9.1 Block Diagram of Clock Generation Circuit

Table 9.2 lists the input/output pins of the clock generation circuit.

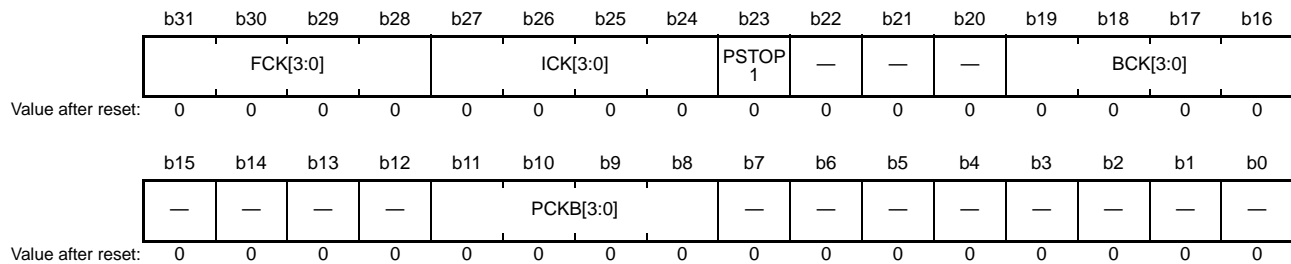
**Table 9.2 Input/Output Pins of Clock Generation Circuit**

Pin Name	I/O	Description
XTAL	Output	These pins are used to connect a crystal resonator. The EXTAL pin can also be used to input an external clock. For details, section 9.3.2, External Clock Input.
EXTAL	Input	
XCIN	Input	These pins are used to connect a 32.768-kHz crystal resonator.
XCOU	Output	
TCK	Input	This pin is used to input the clock for the JTAG.
BCLK	Output	This pin is used to supply external devices with the external bus clock (BCLK).

## 9.2 Register Descriptions

### 9.2.1 System Clock Control Register (SCKCR)

Address(es): 0008 0020h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits should be set to 0001b.	R/W
b7 to b4	—	Reserved	These bits should be set to 0001b.	R/W
b11 to b8	PCKB[3:0]	Peripheral Module Clock B (PCLKB) Select*1, *5	b11 b8 0 0 0 0: x1/1 0 0 0 1: x1/2 0 0 1 0: x1/4 0 0 1 1: x1/8 0 1 0 0: x1/16 0 1 0 1: x1/32 0 1 1 0: x1/64 Settings other than above are prohibited.	R/W
b15 to b12	—	Reserved	These bits should be set to 0001b.	R/W
b19 to b16	BCK[3:0]	External Bus Clock (BCLK) Select*1, *2, *5	b19 b16 0 0 0 0: x1/1 0 0 0 1: x1/2 0 0 1 0: x1/4 0 0 1 1: x1/8 0 1 0 0: x1/16 0 1 0 1: x1/32 0 1 1 0: x1/64 Settings other than above are prohibited.	R/W
b22 to b20	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23	PSTOP1	BCLK Pin Output Control*3	0: BCLK pin output is enabled. 1: BCLK pin output is disabled. (Fixed high)	R/W



Bit	Symbol	Bit Name	Description	R/W
b27 to b24	ICK[3:0]	System Clock (ICLK) Select*1, *2, *4, *5	b27 b24 0 0 0 0: x1/1 0 0 0 1: x1/2 0 0 1 0: x1/4 0 0 1 1: x1/8 0 1 0 0: x1/16 0 1 0 1: x1/32 0 1 1 0: x1/64 Settings other than above are prohibited.	R/W
b31 to b28	FCK[3:0]	FlashIF Clock (FCLK) Select*1, *4, *5	b31 b28 0 0 0 0: x1/1 0 0 0 1: x1/2 0 0 1 0: x1/4 0 0 1 1: x1/8 0 1 0 0: x1/16 0 1 0 1: x1/32 0 1 1 0: x1/64 Settings other than above are prohibited.	R/W

Note 1. The setting for division by one is prohibited if the PLL is selected.

Note 2. Do not make a setting such that the ICLK runs at a lower frequency than the external bus clock.

Note 3. When operation of the external bus clock is selected, the P53 I/O port pin function is not available because it is multiplexed on the same pin as the BCLK pin function.

Note 4. When the SCKCR3.CKSEL[2:0] bits are selecting the sub-clock oscillator in low-speed operating mode 2, division by one is the only frequency division setting allowed for the ICLK and FCLK.

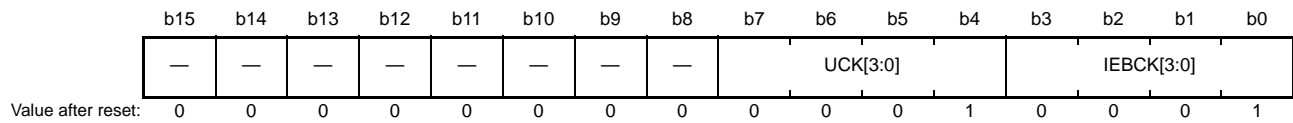
Note 5. The setting for division by one or two is prohibited if the SCKCR3.CKSEL[2:0] bits are set to 010b (the main clock oscillator is selected)

SCKCR should not be modified in the following cases:

- The operating power control mode transition status flag in the operating power control register (OPCCR.OPCMTSF) is 1 (a transition to operating power control mode in progress)
- The ROM P/E mode entry bit *i* in the flash P/E mode entry register (FENTRYR.FENTRY*i*) is 1 (ROM P/E mode, E2 DataFlash P/E mode) (*i* = 0 to 3, D)
- Time period from WAIT instruction execution for a transition to sleep mode, to return from sleep mode to normal operating mode

## 9.2.2 System Clock Control Register 2 (SCKCR2)

Address(es): 0008 0024h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IEBCK[3:0]	IEBUS Clock (IECLK) Select *1	b3 b0 0 0 0 1: $\times 1/2$ 0 0 1 0: $\times 1/4$ 0 0 1 1: $\times 1/8$ 0 1 0 0: $\times 1/16$ 0 1 0 1: $\times 1/32$ 0 1 1 0: $\times 1/64$ 1 1 0 0: $\times 1/6$ Settings other than above are prohibited.	R/W
b7 to b4	UCK[3:0]	USB Clock (UCLK) Select	b7 b4 0 0 1 0: $\times 1/3$ 0 0 1 1: $\times 1/4$ Settings other than above are prohibited when USB is in use. When USB is not in use, these bits are read as 0001b. The write value should be 0001b.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Do not select division by two when the SCKCR3.CKSEL[2:0] bits are 010b (selecting the main clock oscillator).

SCKCR2 should not be modified in the following cases:

- The operating power control mode transition status flag in the operating power control register (OPCCR.OPCMTSF) is 1 (a transition to operating power control mode in progress)
- The ROM P/E mode entry bit *i* in the flash P/E mode entry register (FENTRYR.FENTRY<sub>*i*</sub>) is 1 (ROM P/E mode, E2 DataFlash P/E mode) (*i* = 0 to 3, D)
- Time period from WAIT instruction execution for a transition to sleep mode, to return from sleep mode to normal operating mode

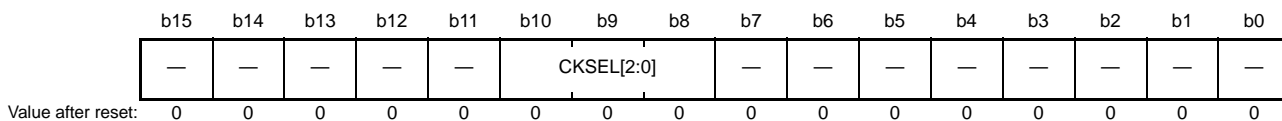
### UCK[3:0] Bits (USB Clock (UCLK) Select)

These bits select the frequency of the USB clock (UCLK).

The duty ratio is 2:1 when  $\times 1/3$  is selected.

### 9.2.3 System Clock Control Register 3 (SCKCR3)

Address(es): 0008 0026h



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 0 1 1: Sub-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

SCKCR3 should not be modified in the following cases:

- The operating power control mode transition status flag in the operating power control register (OPCCR.OPCMTSF) is 1 (a transition to operating power control mode in progress)
- The ROM P/E mode entry bit *i* in the flash P/E mode entry register (FENTRYR.FENTRY*i*) is 1 (ROM P/E mode, E2 DataFlash P/E mode) (*i* = 0 to 3, D)
- Time period from WAIT instruction execution for a transition to sleep mode, to return from sleep mode to normal operating mode

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

These bits select the source of the system clock (ICLK), peripheral module clock (PCLKB), FlashIF clock (FCLK), external bus clock (BCLK), IEBUS clock (IECLK), and USB clock (UCLK) from low-speed on-chip oscillator (LOCO), high-speed on-chip oscillator (HOCO), the main clock oscillator, the sub-clock oscillator, and the PLL circuit. Transitions to clock sources which are not in operation are prohibited.

### 9.2.4 PLL Control Register (PLLCR)

Address(es): 0008 0028h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	PLIDIV[1:0]	PLL Input Frequency Division Ratio Select	b1 b0 0 0: x1 0 1: x1/2 1 0: x1/4 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: x8 0 0 1 0 0 1: x10 0 0 1 0 1 1: x12 0 0 1 1 1 1: x16 0 1 0 0 1 1: x20 0 1 0 1 1 1: x24 0 1 1 0 0 0: x25 1 1 0 0 0 1: x50 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

Writing to the PLLCR is prohibited when the PLLCR2.PLEN bit is 0 (the PLL operates).

#### 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 16 MHz.

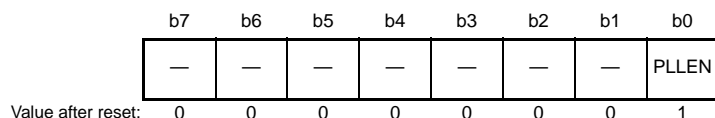
#### STC[5:0] Bits (Frequency Multiplication Factor Select)

These bits select the frequency multiplication factor of the PLL circuit.

Set these bits so that the output frequency is within the range of the VCO oscillation frequency for the PLL (104 MHz to 200 MHz).

### 9.2.5 PLL Control Register 2 (PLLCR2)

Address(es): 0008 002Ah



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

Only set this register after setting the PLL wait control register according to the procedure in section 11, Low Power Consumption.

#### PLLEN Bit (PLL Stop Control)

This bit runs or stops the PLL circuit.

After the setting of the PLLEN bit has been changed so that the PLL operates, only start using the PLL clock after the PLL clock oscillation stabilization waiting time ( $t_{PLLWT1}$  or  $t_{PLLWT2}$ ) has elapsed.

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.

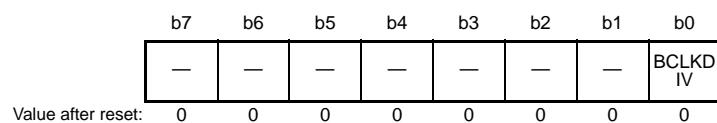
- When restarting the PLL after it has been stopped, allow at least five cycles of the PLL clock as an interval over which it is still stopped.
- Ensure that oscillation by the PLL is stable when making the setting to stop the PLL.
- Regardless of whether or not it is selected as the system clock, ensure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

Writing of 1 to the PLLEN bit (stopping the PLL) is prohibited while the PLL clock is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).

Writing of 0 to the PLLEN bit (making the PLL operate) is prohibited when the setting of the operating-power control mode selection bits in the operating-power control register (OPCCR.OPCM[2:0]) is for low-speed operating mode 1 or 2.

## 9.2.6 External Bus Clock Control Register (BCKCR)

Address(es): 0008 0030h



Bit	Symbol	Bit Name	Description	R/W
b0	BCLKDIV	BCLK Pin Output Select	0: BCLK 1: 1/2 BCLK	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

BCKCR should not be modified in the following cases:

- The operating power control mode transition status flag in the operating power control register (OPCCR.OPCMTSF) is 1 (a transition to operating power control mode in progress)
- The ROM P/E mode entry bit *i* in the flash P/E mode entry register (FENTRYR.FENTRY<sub>*i*</sub>) is 1 (ROM P/E mode, E2 DataFlash P/E mode) (*i* = 0 to 3, D)
- Time period from WAIT instruction execution for a transition to sleep mode, to return from sleep mode to normal operating mode

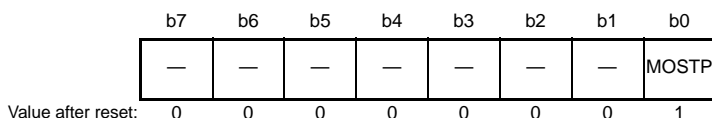
### BCLKDIV Bit (BCLK Pin Output Select)

This bit selects the clock signal for output from the BCLK pin.

Either the BCLK clock with the frequency selected by the BCK[3:0] bits in SCKCR or the BCLK clock divided by 2 can be selected.

### 9.2.7 Main Clock Oscillator Control Register (MOSCCR)

Address(es): 0008 0032h



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

Only set this register after setting the main clock oscillator wait control register according to the procedure in section 11, Low Power Consumption.

#### MOSTP Bit (Main Clock Oscillator Stop)

This bit runs or stops the main clock oscillator.

The main clock oscillator is operated or stopped by the MOSTP bit and main clock oscillator forced oscillation bit in the main clock oscillator forced oscillation control register (MOFCR.MOFXIN). The main clock oscillator can be started by setting the MOSTP bit to operating or by setting the MOFXIN bit to forced oscillation. When the MOFXIN bit is set to forced oscillation, the oscillator operates even in software standby mode and deep software standby mode.

When changing the value of the MOSTP bit or MOFCR.MOFXIN bit, execute subsequent instructions after reading the bit and checking that its value has actually been updated (refer to (2) Notes on writing to I/O registers, in section 5, I/O Registers).

When a crystal oscillator is connected to supply the main clock signal, after changing the MOSTP bit or MOFCR.MOFXIN bit so that the main clock oscillator operates, only use the main clock after the main clock oscillation stabilization waiting time (crystal; tMAINOSCWT) has elapsed.

When an external clock is connected to supply the main clock signal, after changing the MOSTP bit or MOFCR.MOFXIN bit so that the main clock oscillator operates, only use the main clock after the EXTAL external clock input waiting time (tEXWT) has elapsed.

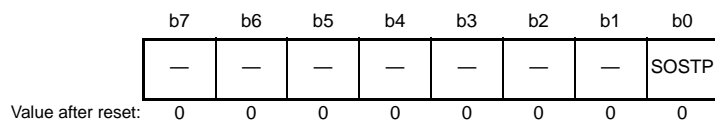
For the main clock oscillator, a fixed time is required for oscillation to become stable after the settings for operation have been made. Furthermore, a fixed time is required for oscillation to actually stop after the settings to stop oscillation have been made. Accordingly, take note of the following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

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

Writing of 1 to the MOSTP bit (stopping the main clock oscillator) is prohibited while the main clock oscillator or PLL is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).

### 9.2.8 Sub-Clock Oscillator Control Register (SOSCCR)

Address(es): 0008 0033h



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

Only set this register after setting the sub-clock oscillator wait control register according to the procedure in section 11, Low Power Consumption.

#### SOSTP Bit (Sub-Clock Oscillator Stop)

This bit runs or stops the sub-clock oscillator.

The SOSTP bit and the sub-clock control bit in the RTC control register 3 (RCR3.RTCEN) controls whether to operate or stop the sub-clock oscillator. If one of these bits is set so as to enable the operation, the sub-clock oscillator runs.

When changing the value of the SOSTP bit or RCR3.RTCEN bit, execute subsequent instructions after reading the bit and checking that its value has actually been updated (refer to (2) Notes on writing to I/O registers, in section 5, I/O Registers).

For the sub-clock to operate, the sub-clock oscillator wait control register (SOSCWTCR) must be set. For details on setting the SOSCWTCR register, see section 11, Low Power Consumption.

After the setting of the SOSTP bit or the RCR3.RTCEN bit has been changed so that the sub-clock oscillator operates, only start using the sub-clock after the sub-clock oscillation stabilization waiting time has elapsed.

That is, a fixed time for stabilization is required after the setting for sub-clock oscillator 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.

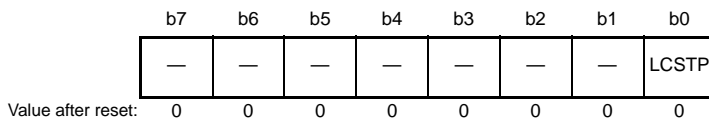
- When restarting the sub-clock oscillator after it has been stopped, allow at least five cycles of the sub-clock as an interval over which it is still stopped.
- Ensure that oscillation by the sub-clock oscillator is stable when making the setting to stop the sub-clock oscillator.
- Regardless of whether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

Writing of 1 to the SOSTP bit (stopping the sub-clock oscillator) is prohibited while the sub-clock oscillator is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).



### 9.2.9 Low-Speed On-Chip Oscillator Control Register (LOCOCR)

Address(es): 0008 0034h



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

#### 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 after the LOCO clock oscillation stabilization waiting time (tLOCOWT) 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 or deep software standby.
- When a transition to software standby or deep software standby 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.

Writing of 1 to the LCSTP bit (stopping the LOCO) is prohibited while the LOCO is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).

Writing of 1 to the LCSTP bit (stopping the LOCO) is prohibited if detection of oscillation stopping is enabled by the oscillation-stop detection-enable bit in the oscillation stop detection control register (OSTDCR.OSTDE).

### 9.2.10 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	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

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

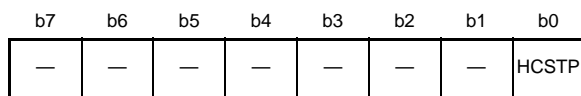
#### ILCSTP Bit (IWDT-Dedicated On-Chip Oscillator Stop)

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 within the LSI only starts after a time of waiting for stabilization of the LOCO (tLOCOWT) has elapsed. If the IWDT-dedicated clock is to be used, only start using the oscillator after this waiting time (tLOCOWT) 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 or deep software standby.

### 9.2.11 High-Speed On-Chip Oscillator Control Register (HOCOOCR)

Address(es): 0008 0036h



Value after reset: 0 0 0 0 0 0 0 0/1\*1

Note 1. The HCSTP bit value after a reset is 0 when the HOCO oscillation enable bit in the 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.

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

#### HCSTP Bit (HOCO Stop)

This bit runs or stops the HOCO.

After the setting of the HCSTP bit has been changed so that the HOCO operates, supply of the clock within the LSI only starts after time corresponding to the HOCO clock oscillation stabilization waiting time 2 (tHOCOWT2) has elapsed.

Only start using the oscillator after this waiting time (tHOCOWT2) has elapsed. 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.

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

Writing of 1 to the HCSTP bit (stopping the HOCO) is prohibited while the HOCO is selected by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).

Writing of 0 to the HCSTP bit (making the HOCO operate) is prohibited when the setting of the operating-power control mode selection bits in the operating-power control register (OPCCR.OPCM[2:0]) is for low-speed operating mode 2.

## 9.2.12 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

### 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 clearing 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 out a given I/O register, waiting 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.

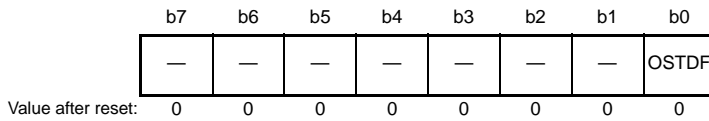
When the OSTDE bit is 1 (oscillation stop detection function enabled), the LOCO stop bit (LOCOCR.LCSTP) is cleared to 0 and the LOCO operation is started. The LOCO cannot be stopped while the oscillation stop detection function is enabled; writing 1 to the LOCOCR.LCSTP bit (LOCO stopped) is invalid.

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

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

### 9.2.13 Oscillation Stop Detection Status Register (OSTDSR)

Address(es): 0008 0041h



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 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 bit is not cleared to 0 even though the main clock oscillation is restarted. The OSTDF bit is cleared to 0 by reading 1 from the bit and then writing 0. At least 3 ICLK cycles of wait time is necessary between writing 0 to OSTDF and reading OSTDF as 0. If the OSTDF bit is cleared to 0 while the main clock oscillation is stopped, the OSTDF bit becomes 0 and then returns to 1.

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

#### [Setting condition]

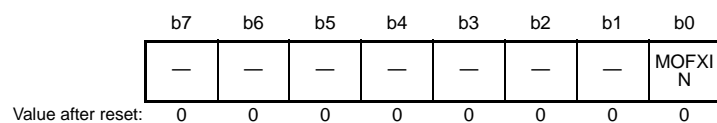
- The main clock oscillation is stopped with the OSTDCR.OSTDE being 1 (oscillation stop detection function enabled).

#### [Clearing condition]

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

### 9.2.14 Main Clock Oscillator Forced Oscillation Control Register (MOFCR)

Address(es): 0008 C293h



Bit	Symbol	Bit Name	Description	R/W
b0	MOFXIN	Main Clock Oscillator Forced Oscillation	0: Oscillator is not controlled by this bit. 1: The main clock oscillator is forcedly oscillated.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

The EXTAL/XTAL pin is also used as a port. In the initial setting state, the pin is set as a port. The EXTAL/XTAL signal can be used as the RTC clock source even in deep software standby mode.

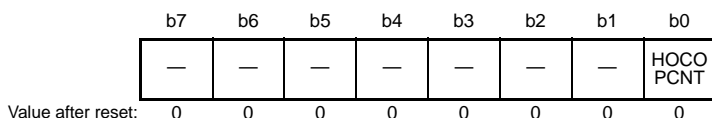
#### MOFXIN Bit (Main Clock Oscillator Forced Oscillation)

This bit controls forced oscillation of the main clock oscillator.

When changing the value of the MOSCCR.MOSTP bit or MOFXIN bit, execute subsequent instructions after reading the bit and checking that its value has actually been updated (refer to (2) Notes on writing to I/O registers, in section 5, I/O Registers).

### 9.2.15 High-Speed On-Chip Oscillator Power Supply Control Register (HOCOPCR)

Address(es): 0008 C294h



Bit	Symbol	Bit Name	Description	R/W
b0	HOCOPCNT	High-Speed On-Chip Oscillator Power Supply Control	0: Turns the power supply of the HOCO on. 1: Turns the power supply of the HOCO off.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

#### HOCOPCNT Bit (High-Speed On-Chip Oscillator Power Supply Control)

This bit controls the power supply for the HOCO.

When this bit is set to 0, the power supply of the HOCO is turned on, enabling oscillation.

When this bit is set to 1, the power supply of the HOCO is turned off, reducing power consumption.

When setting the HOCOPCNT bit to 1, set the HOCO stop bit in the high-speed on-chip oscillator control register (HOCOCCR.HCSTP) to 1 (HOCO stopped).

After the HOCOPCNT bit is changed from 1 to 0, oscillation settling time is required before the HOCOCCR.HCSTP bit is set to 0. For details, see section 45, Electrical Characteristics.

Do not change the value of the HOCOPCNT bit in the following cases:

- When the HOCO is selected as the clock source by the clock source select bits in system clock control register 3 (SCKCR3.CKSEL[2:0]).
- When the setting of the operating-power control mode selection bits in the operating-power control register (OPCCR.OPCM[2:0]) is for low-speed operating mode 1 or 2.

## 9.3 Main Clock Oscillator

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

### 9.3.1 Connecting a Crystal Resonator

Figure 9.2 shows an example of connecting a crystal resonator.

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.

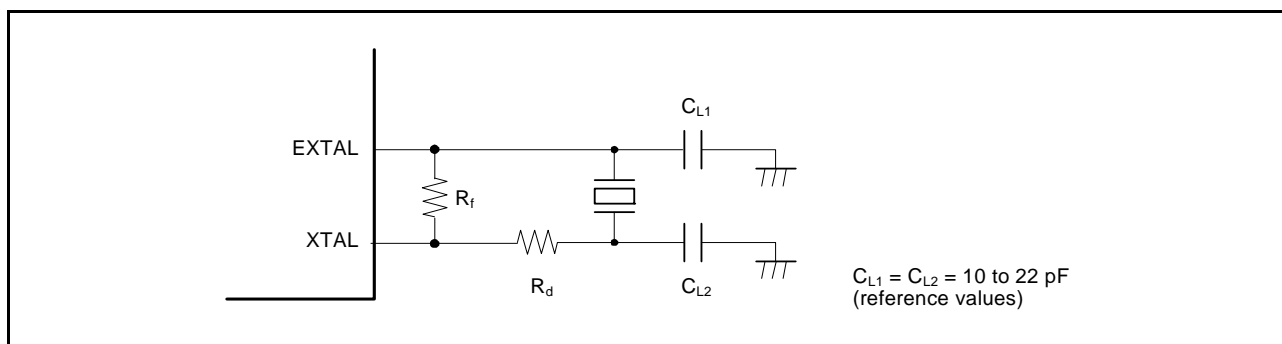


Figure 9.2 Example of Crystal Resonator Connection

Table 9.3 Damping Resistance (Reference Values)

Frequency (MHz)	4	8	12	16
Rd (Ω)	380	200	0	0

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

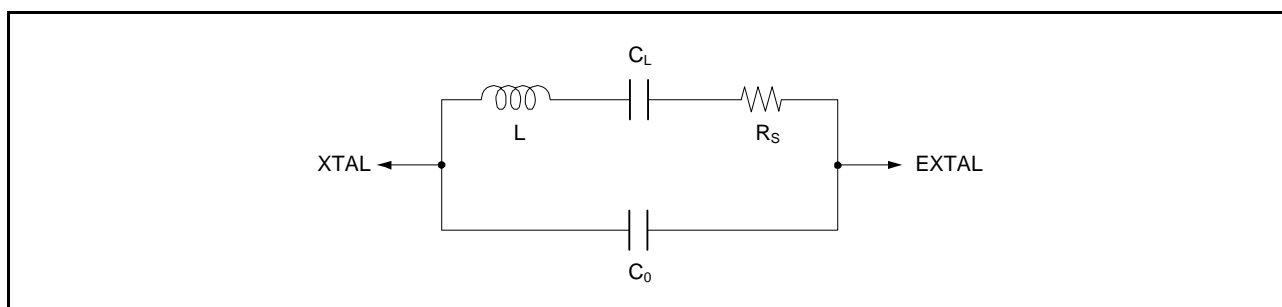


Figure 9.3 Equivalent Circuit of Crystal Resonator

Table 9.4 Crystal Resonator Characteristics (Reference Values)

Frequency (MHz)	4	8	12	16
RS max (Ω)	160	80	60	40
C0 max (pF)	7			



### 9.3.2 External Clock Input

Figure 9.4 shows examples of connection of external clock input. To leave the XTAL pin open, make the parasitic capacitance less than 5 pF.

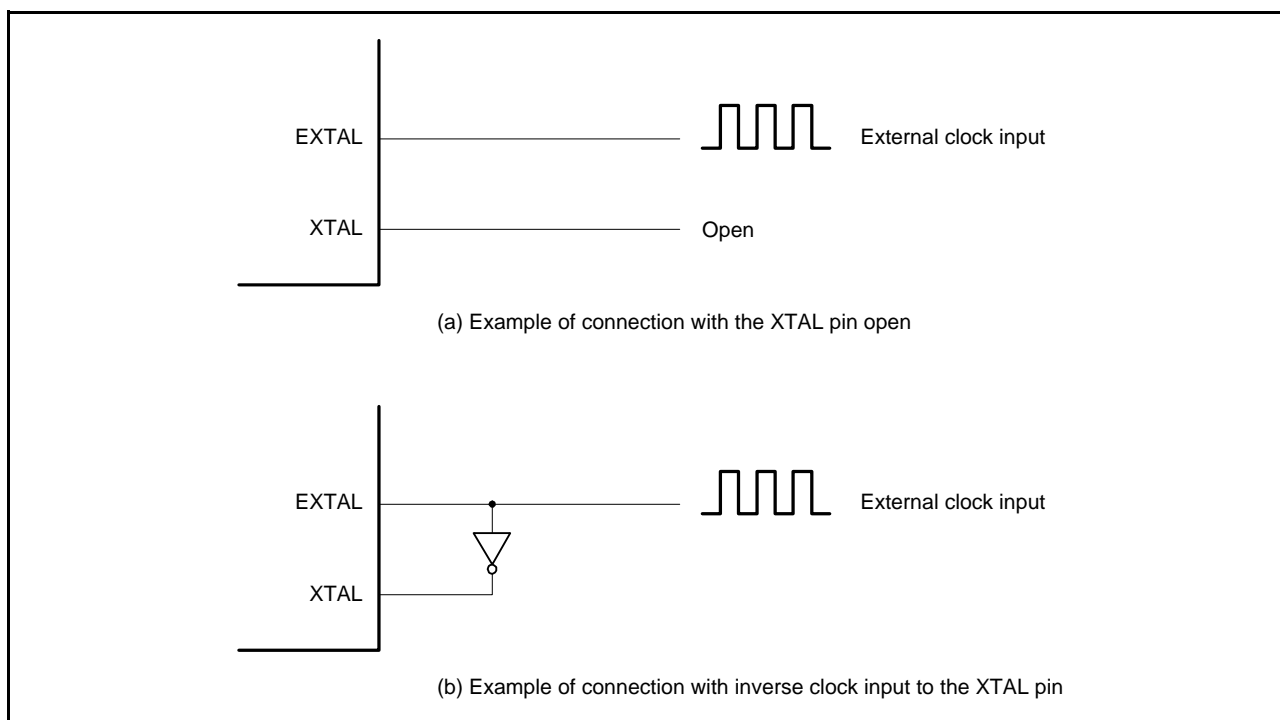


Figure 9.4 Equivalent Circuit for Crystal Resonator

### 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 (making the main clock oscillator run) or that of the main clock oscillator forced oscillation bit (MOFCR.MOFXIN) is 1 (forcing the main clock oscillator to run).

## 9.4 Sub-Clock Oscillator

The only way of supplying the clock signal to the sub-clock oscillator is connecting a crystal oscillator.

### 9.4.1 Connecting 32.768-kHz Crystal Resonator

To supply a clock to the sub-clock oscillator, connect a 32.768-kHz crystal resonator, as shown in Figure 9.5. 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 XCIN and XCOU 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 sub-clock oscillator described in Table 9.1.

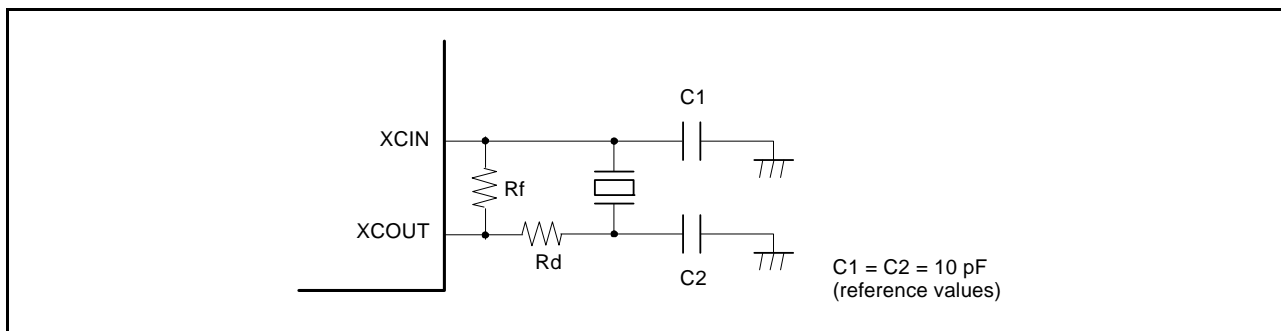


Figure 9.5 Connection Example of 32.768-kHz Crystal Resonator

Figure 9.6 shows an equivalent circuit for the 32.768-kHz crystal resonator. Use a crystal resonator that has the characteristics listed in Table 9.5.

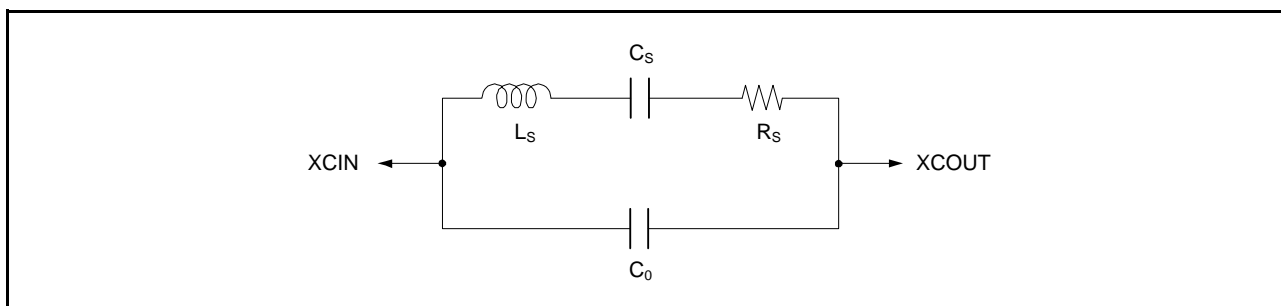


Figure 9.6 Equivalent Circuit for Crystal Resonator

Table 9.5 Crystal Resonator Characteristics (Reference Values)

Frequency (kHz)	32.768
$R_S$ max (k $\Omega$ )	35
$C_0$ max (pF)	2.0

### 9.4.2 Handling of Pins when Sub-Clock is Not Used

If the sub-clock is not in use, connect the XCIN pin to VSS via a resistor (to pull VSS down) and leave the XCOUT pin open-circuit as shown in Figure 9.7.

In addition, if an oscillator is not connected, set the sub-clock oscillator stop bit (SOSCCR.SOSTP) to 1 (stopping the oscillator) and the sub-clock oscillator control bit in RTC control register 3 (RCR3.RTCEN) to 0 (stopping the sub-clock oscillator). The state of the sub-clock control circuit will be undefined after a cold start. Accordingly, be sure to set these bits after a cold start.

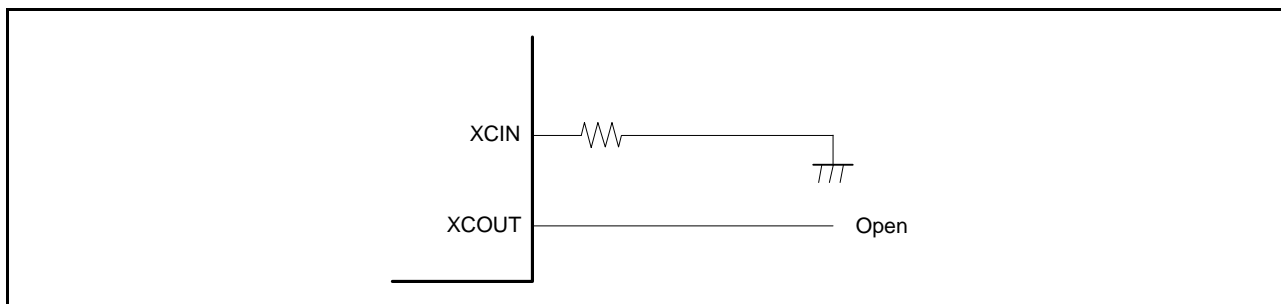


Figure 9.7 Pin Handling when Sub-Clock is not Used

## 9.5 Oscillation Stop Detection Function

### 9.5.1 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 or PLL 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, see section 22, Multi-Function Timer Pulse Unit 2 (MTU2a) and section 23, Port Output Enable 2 (POE2a).

In the RX630 Group, 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 (see Table 45.27, Oscillation Stop Detection Circuit Characteristics).

When an oscillation stop is detected, the main clock or PLL 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 or PLL clock selected as the system clock source, the system clock source is switched to the LOCO clock without a change of CKSEL[2:0].

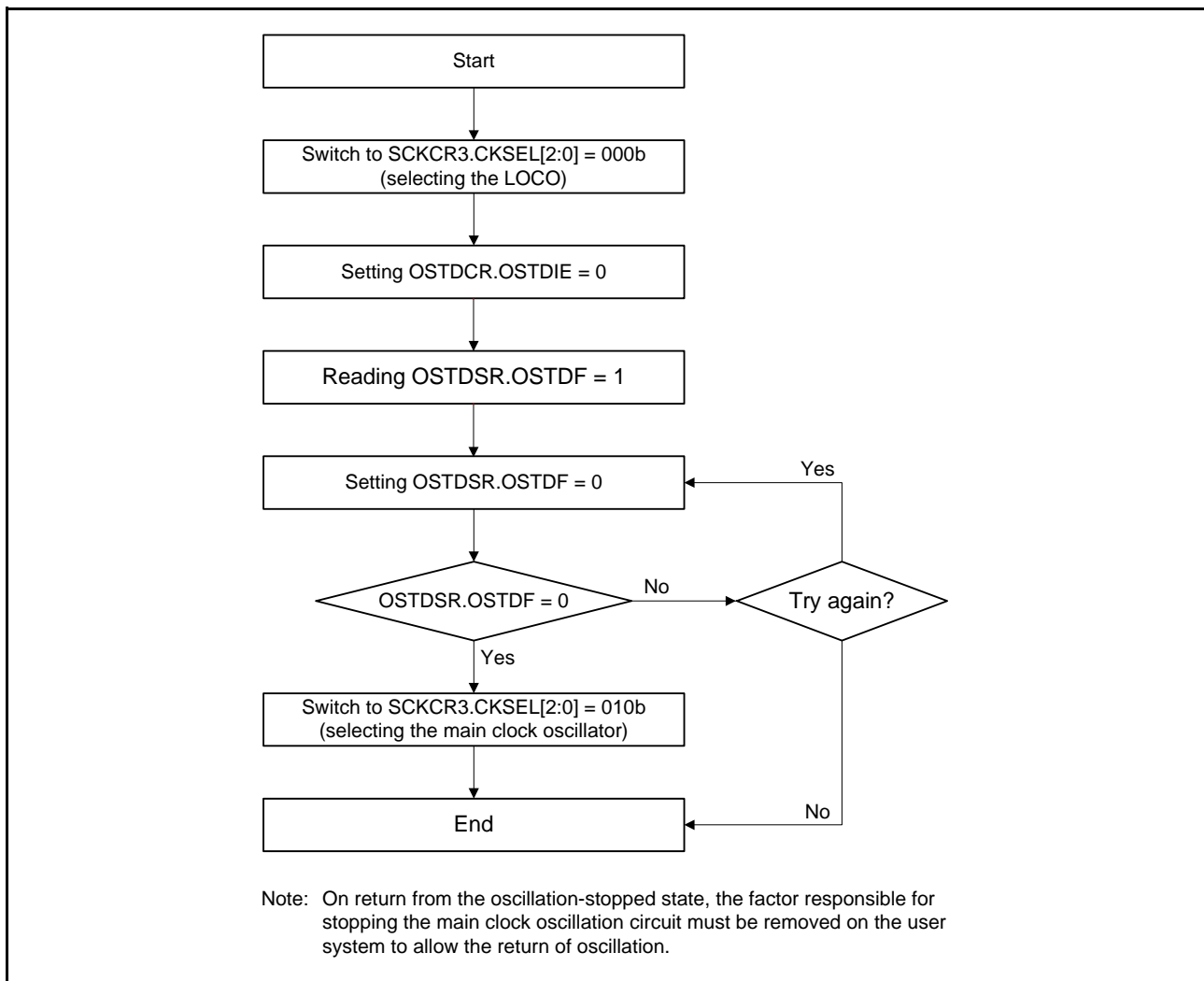
Switching between the main clock and LOCO clock or between the PLL clock and LOCO clock is controlled by the oscillation stop detection flag (OSTDSR.OSTDF). The clock source is switched to the LOCO clock when the OSTDF flag is 1, and is switched to the main clock or PLL clock again when the OSTDF flag is cleared 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 cleared 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 clear 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 settling 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 settling 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 or deep software standby mode.

The clocks that are switched to the LOCO clock by the oscillation stop detection are: the main clock, PLL clock, and CAN clock (CANMCLK), which are provided as the system clock sources. The main clock as the RTC-dedicated main clock source (RTCMCLK) is not switched to the low-speed clock.

The system clock (ICLK) frequency during the low-speed clock operation is specified by the LOCO oscillation frequency and the division ratio set by the system clock select bits (SCKCR.ICK[3:0])



**Figure 9.8** Flow of Recovery from Detection of Oscillator Stop

### 9.5.2 Oscillation Stop Detection Interrupts

An oscillation-stop detection interrupt (OSTDI) will be generated if the oscillation-stop detection flag (OSTDSR.OSTDF) becomes 1 while the oscillation-stop detection interrupt enable bit in the oscillation stop detection control register (OSTDCR.OSTDIE) is 1 (enabling interrupt generation on oscillation stop detection). At this time, the main clock oscillator stop is notified to the port output enable 2 (POE). On accepting the notification of the oscillation stop, the POE sets the OSTST high-impedance flag in input level control/status register 3 (ICSR3.OSTSTF) to 1. After the oscillation stop is detected, wait for at least 10 cycles of PCLK before writing to this ICSR3.OSTSTF flag. When the OSTDSR.OSTDF flag requires clearing, do so after clearing the oscillation-stop detection interrupt enable bit in the oscillation stop detection control register (OSTDCR.OSTDIE). 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 out a given I/O register, waiting 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, see section 15, Interrupt controller (ICUb).

## 9.6 PLL Circuit

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

## 9.7 Internal Clock

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

- (1) Operating clock of the CPU, DMAC, DTC, ROM, and RAM: System clock (ICLK)
- (2) Operating clock of peripheral modules: Peripheral module clock (PCLKB)
- (3) Operating clock of the FlashIF: FlashIF clock (FCLK)
- (4) Clock for the external bus controller and external pin output: External bus clock (BCLK)
- (5) Operating clock for the USB: USB clock (UCLK)
- (6) Operating clock for the CAN: CAN clock (CANMCLK)
- (7) Operating clock for the IEBUS: IEBUS clock (IECLK)
- (8) Operating clock for the RTC: RTC-dedicated main clock (RTCMCLK)
- (9) Operating clock for the RTC: RTC-dedicated sub-clock (RTCSCLK)
- (10) Operating clock for the IWDT: IWDT-dedicated clock (IWDTCLK)
- (11) Operating clock for the JTAG: JTAG clock (JTAGTCK)

Frequencies of the internal clocks are set by the combination of the divisors selected by the FCK[3:0], ICK[3:0], BCK[3:0], and PCKB[3:0] bits in SCKCR, the UCK[3:0] and IEBC[3:0] bits in SCKCR2, the clock source selected by the CKSEL[2:0] bits in SCKCR3, and the bits that select the frequency of the PLL circuit (STC[5:0] and PLIDIV[1:0] in PLLCR). If the value of any of these bits is changed, subsequent operation will be at the frequency determined by the new value.

### 9.7.1 System Clock

The system clock (ICLK) is used as the operating clock of the CPU, DMAC, DTC, ROM, and RAM.

The ICLK frequency is specified by the ICK[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, and the STC[5:0] and PLIDIV[1:0] bits in PLLCR.

### 9.7.2 Peripheral Module Clock

The peripheral module clock (PCLKB) is the operating clocks for use by peripheral modules.

The PCLKB frequencies are specified by the PCKB[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, and the STC[5:0] and PLIDIV[1:0] bits in PLLCR.

### 9.7.3 FlashIF Clock

The flash-interface clock (FCLK) is used as the operating clock for the flash-memory interfaces. That is, FCLK is used for programming and erasure of the ROM and E2 DataFlash, and reading from the E2 DataFlash.

The FCLK frequency is specified by the FCK[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, and the STC[5:0] and PLIDIV[1:0] bits in PLLCR.

### 9.7.4 External Bus Clock

The external bus clock (BCLK) is an operating clock for the external bus controller. It is also output externally from the BCLK pin for the external connection bus. When the external bus is enabled, P53 that is function-multiplexed with the BCLK pin cannot be used as an I/O port.

BCLK can be output from the BCLK pin by setting the SCKCR.PSTOP1 bit to 0 and setting the external bus enable bit in the system control register 0 (SYSCR0.EXBE) to 1. Make sure that modification of the SYSCR0.EXBE bit to 1 must

always be performed while the PSTOP1 bit in SCKCR is 1.

When the BCKCR.BCLKDIV bit is set to 1, the BCLK clock divided by 2 is output from the BCLK pin.

The BCLK frequency is specified by the BCK[3:0] bits in SCKCR, the CKSEL[2:0] bits in SCKCR3, and the STC[5:0] and PLIDIV[1:0] bits in PLLCR.

### 9.7.5 USB Clock

The USB clock (UCLK) is an operating clock for the USB.

The UCLK frequency is specified by the UCK[3:0] bits in SCKCR2, the CKSEL[2:0] bits in SCKCR3, and the STC[5:0] and PLIDIV[1:0] bits in PLLCR.

A 48-MHz clock must be supplied to the USB module. When the USB module is used, setting must be made so that UCLK is 48 MHz.

### 9.7.6 CAN Clock

The CAN clock (CANMCLK) is an operating clock for the CAN.

CANMCLK is generated by the main clock oscillator.

### 9.7.7 IEBUS Clock

The IEBUS clock (IECLK) is an operating clock for the IEBUS.

The IECLK frequency is specified by the IEBC[3:0] bits in SCKCR2, the CKSEL[2:0] bits in SCKCR3, and the STC[5:0] and PLIDIV[1:0] bits in PLLCR.

### 9.7.8 RTC-Dedicated Clock

The RTC-dedicated clock (RTCSCLK, RTCMCLK) is the operating clock for the RTC.

RTCSCLK is generated by the sub-clock oscillator, and RTCMCLK is generated by the main clock oscillator.

### 9.7.9 IWDT-Dedicated Clock

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

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

### 9.7.10 JTAG Clock

The JTAG-dedicated clock (JTAGTCK) is the operating clock for the JTAG.

JTAGTCK is generated by the external clock for JTAG (TCK).

## 9.8 Pin Settings When an Oscillator is Connected

### (1) Main clock

Set P36 and P37 as input pins and then clear the main clock oscillator stop bit (MOSTP in MOSCCR) to 0 so that the clock runs, or set the main clock oscillator forced oscillation bit (MOFXIN in MOFCR) to 1 so that it is forcibly made to oscillate.

### (2) Sub-clock

Clear the sub-clock oscillator stop bit (SOSTP in SOSCCR) to 0 (so that the clock runs), or set the sub-clock oscillator control bit (RCR3 in RTCEN) to 1 (so that the clock runs).

## 9.9 Pin Settings When an External Clock is Connected

### (1) Main clock

Set P36 and P37 as input pins and then clear the main clock oscillator stop bit (MOSTP in MOSCCR) to 0 (so that the clock runs), or set the main clock oscillator forced oscillation bit (MOFXIN in MOFCR) to 1 so that it is forcibly made to oscillate.

## 9.10 Usage Notes

### 9.10.1 Notes on Clock Generation Circuit

- (1) The frequencies of the system clock (ICLK), peripheral module clock (PCLKB), FlashIF clock (FCLK), and external bus clock (BCLK) supplied to each module change according to the settings of SCKCR. Each frequency should meet the following:

Select each frequency that is within the operation guaranteed range of clock cycle time (tcyc) 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. Note therefore that the operating speed of modules such as the timer and SCI varies before and after the frequency is changed.

- (2) The following relation is required between the frequencies of the system clock (ICLK) and external bus clock (BCLK).

$$\text{ICLK} \geq \text{BCLK}$$

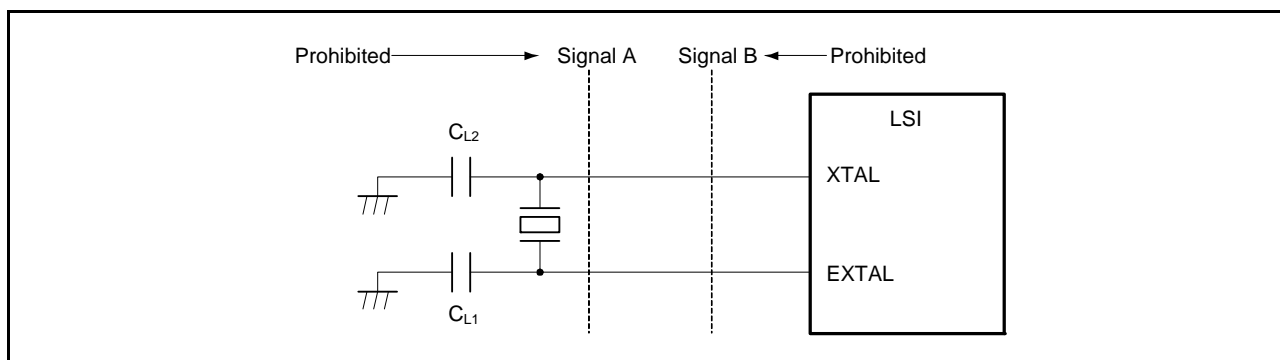
- (3) Do not change the clock frequency during external bus access. Furthermore, when access via the external bus is to start after a change to the clock frequency, only start access via the bus after confirming that the change to the frequencies has been completed.
- (4) To secure the processing after 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.

### 9.10.2 Notes on Resonator

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

### 9.10.3 Notes on Board Design

When using a crystal resonator, place the resonator and its load capacitors as close to the XTAL and EXTAL pins as possible. Other signal lines should be routed away from the oscillation circuit as shown in Figure 9.9 to prevent electromagnetic induction from interfering with correct oscillation.



**Figure 9.9 Notes on Board Design for Oscillation Circuit (Applies to the Sub-Clock Oscillator, in Case of the Main Clock Oscillator)**

#### 9.10.4 Notes on Resonator Connect Pin

When the main clock is not used, the EXTAL and XTAL pins can be used as general ports P36 and P37. When they are used as the general ports, the main clock should be stopped (MOSCCR.MOSTP should be set to 1 and MOFCR.MOFXIN should be set to 0). However, with the system using the main clock, the EXTAL (P36) and XTAL (P37) pins should not be used as output ports.

For the values of registers related to port settings, refer to Table 21.36, Register Settings.

Furthermore, since the main clock becomes essential if the function indicated below is in use, design the board so that both pins can be used for the main clock signal.

- Programming of flash memory in boot mode or USB boot mode\*1

Note 1. For the conditions on the oscillator in the various modes, see item 10 in section 43.12, Usage Notes (Common to the ROM/E2 DataFlash Memory).

#### 9.10.5 Notes on Sub-Clock Oscillator

The sub-clock can be used as the system clock, as the source to drive counting by the real-time clock, or as both.

Accordingly, take note of the following limitations and points for caution regarding the settings, including when the sub-clock is not in use.

- To select the sub-clock as the system clock, set the SOSCCR.SOSTP bit; to set the sub-clock as the source to drive counting by the realtime clock, use the SOSCCR.SOSTP and RCR3.RTCEN bits.

Furthermore, when the sub-clock is to be used as the clock source for counting by the realtime clock, make initial settings according to the procedure below and then follow the procedure for setting up the realtime clock described in section 28.3.2, Clock Setting Procedure

Procedure for initial settings:

1. Set the SOSCCR.SOSTP bit to 0.
2. Wait for the sub-clock oscillation stabilization waiting time\*1.
3. Confirm that the value of the SOSCCR.SOSTP bit has changed to 0, and then set the RCR3.RTCEN bit to 1.
4. Confirm that the value of the RCR3.RTCEN bit has changed to 1, and then set the SOSCCR.SOSTP bit to 1.
5. Set the SOSCWTCR.SSTS[4:0] bits to 00000b.
6. Wait for at least five cycles of the sub-clock.
7. Set the SOSCCR.SOSTP bit to 0.
8. Wait for at least two cycles of the sub-clock.
9. Confirm that the value of the SOSCCR.SOSTP bit has changed, and then set the RCR3.RTCEN bit to 0.

Note 1. For sub-clock oscillation stabilization waiting time, see 11.2.8 Sub-Clock Oscillator Wait Control Register (SOSCWTCR).



- When the sub-clock is the system clock, even if the RCR3.RTCEN bit is set to 1 and the sub-clock is already oscillating, wait for at least five cycles of the sub-clock after the SOSCCR.SOSTP bit changes from 1 (stopped) to 0 (operating) before starting to use the sub-clock as the system clock
- The state of the control circuit for the sub-clock oscillator is undefined after a cold start so the sub-clock must be initialized even if it is not going to be used. Setting both the SOSCCR.SOSTP bit and the RCR3.RTCEN bit to the “stopped” settings will initialize the control circuit. For RCR3.RTCEN bit, see section 28.2.19, RTC Control Register 3 (RCR3)
- For the sub-clock oscillator, the b3 to b1 bits in the RCR3 register must also be set. Be sure to set them to 110b. Moreover, this setting must be made while the sub-clock oscillator is stopped. Writing to these bits during operation is prohibited
- If the RCR3.RTCEN bit is modified after the SOSCCR.SOSTP bit is modified, or if the SOSCCR.SOSTP bit is modified after the RCR3.RTCEN bit is modified, make sure that first modified bit has been rewritten before the subsequent bit is modified.

## 10. Frequency Measurement Circuit (MCK)

The respective clock count extension circuits of the internal MTU (system 1) or unit 0 of TPU (system 2) can be used to monitor the main clock, sub-clock, LOCO (for system use), PLL, and HOCO for abnormal frequencies.

### 10.1 Overview

Table 10.1 lists the specifications of the frequency measurement circuit. Figure 10.1 and Figure 10.2 illustrate block diagrams (of system 1 and system 2), and Figure 10.3 illustrates the software configuration.

**Table 10.1 Specifications of Frequency Measurement Circuit**

Particulars	Description	
Clock to be monitored	Main clock, sub clock, LOCO (system), PLL, HOCO* <sup>1</sup>	
Circuit structure	<ul style="list-style-type: none"> <li>Channel 0 (MTU0) and channel 1 (MTU1) of MTU, counter-clock extension circuit 1 (system 1)</li> <li>Channel 0 (TPU0) and channel 1 (TPU1) of TPU unit 0, counter-clock extension circuit 2 (system 2)</li> </ul>	
Settings	Select clock to be monitored from the main clock, sub clock, and LOCO.* <sup>2</sup>	
Counter-clock extension register n (n = 1, 2)		
MTU0/ TPU0	Timer mode	Normal operation (compare match)
	Counter clock	MTCLKD/TCLKD external pin input (clock selected by counter-clock extension register n; n = 1, 2)
	Compare match	Compare match A, output disabled
	Counter clear	Cleared by compare match A
	Interrupt	Not used
MTU1/ TPU1	Timer mode	Normal operation (input capture)
	Counter clock	PCLK
	Input capture	Input capture A by compare match A of channel 0
	Counter clear	Cleared by input capture A
	Interrupt	Input capture A (TGI1A)
Minimum pulse width for input signals on the MTCLKD/TCLKD external pin (output signals for counter-clock extension circuit n; n = 1, 2)	<ul style="list-style-type: none"> <li>Single edge specification: <math>tTCKWH</math> minimum cycle = <math>1.5 \times (1/PCLK) tPcyc</math></li> <li>Double edge specification: <math>tTCKWL</math> minimum cycle = <math>2.5 \times (1/PCLK) tPcyc</math></li> </ul> ( $tTCKWH$ , $tTCKWL$ : Timer clock pulse width, $tPcyc$ : PCLK cycle)	
Power consumption reducing function	module-stop state can be set.	
Others	System 1 and system 2 are capable of operating independently.	

Note 1. PLL and HOCO can be selected as a clock to be monitored when they are system clocks (clock source of PCLK)

Note 2. When PLL is selected as a clock to be monitored, a clock other than the main clock should be selected by the counter-clock extension register n (n = 1, 2).

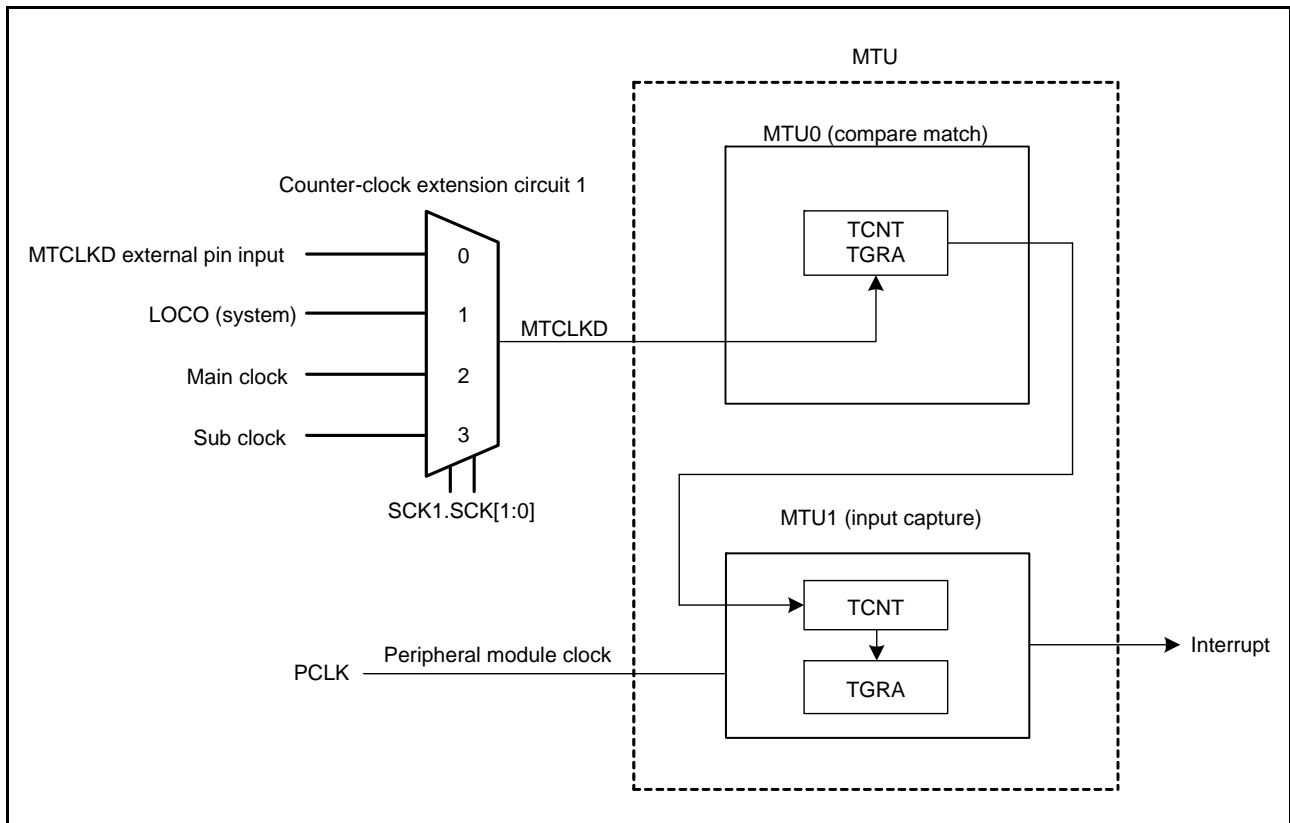


Figure 10.1 Block Diagram of Frequency Measurement Circuit with MTU (System 1)

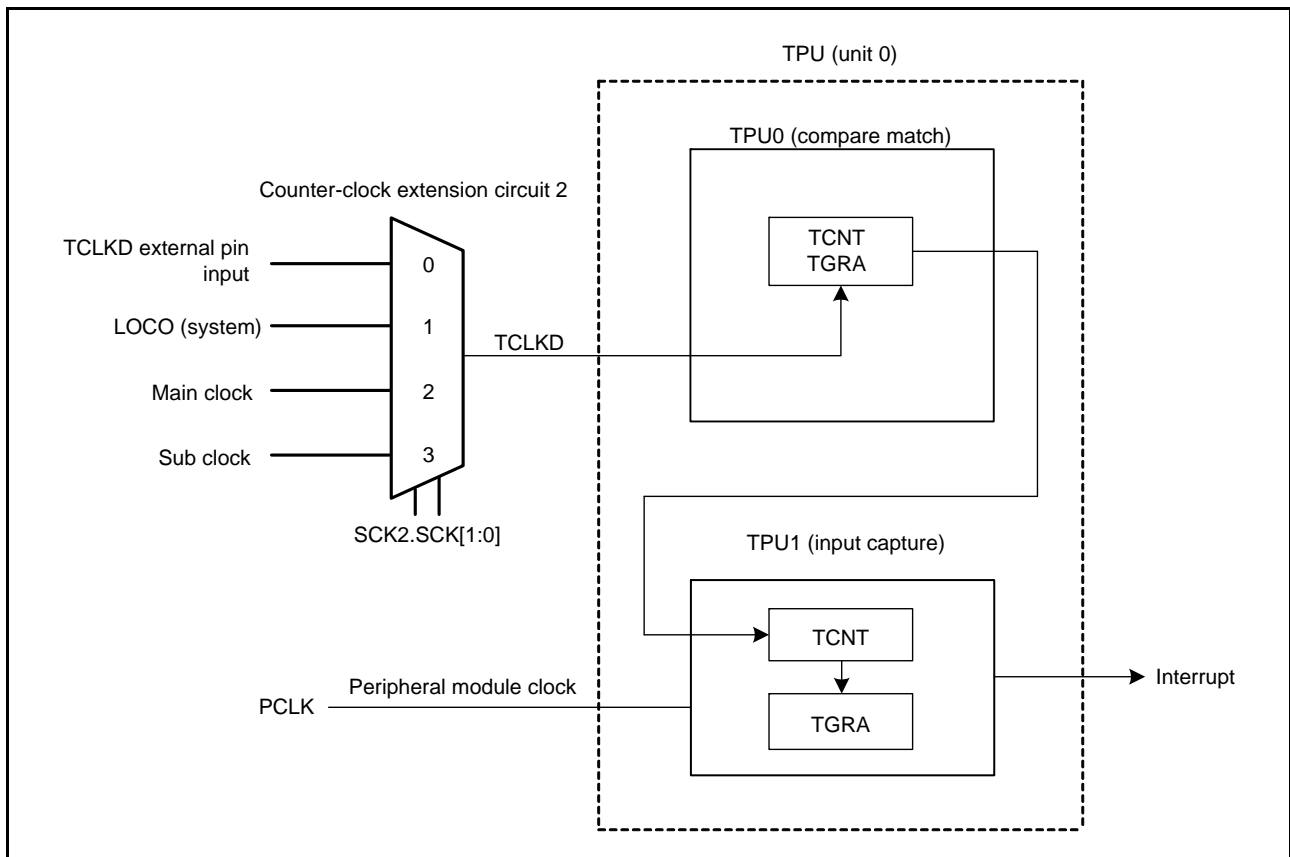
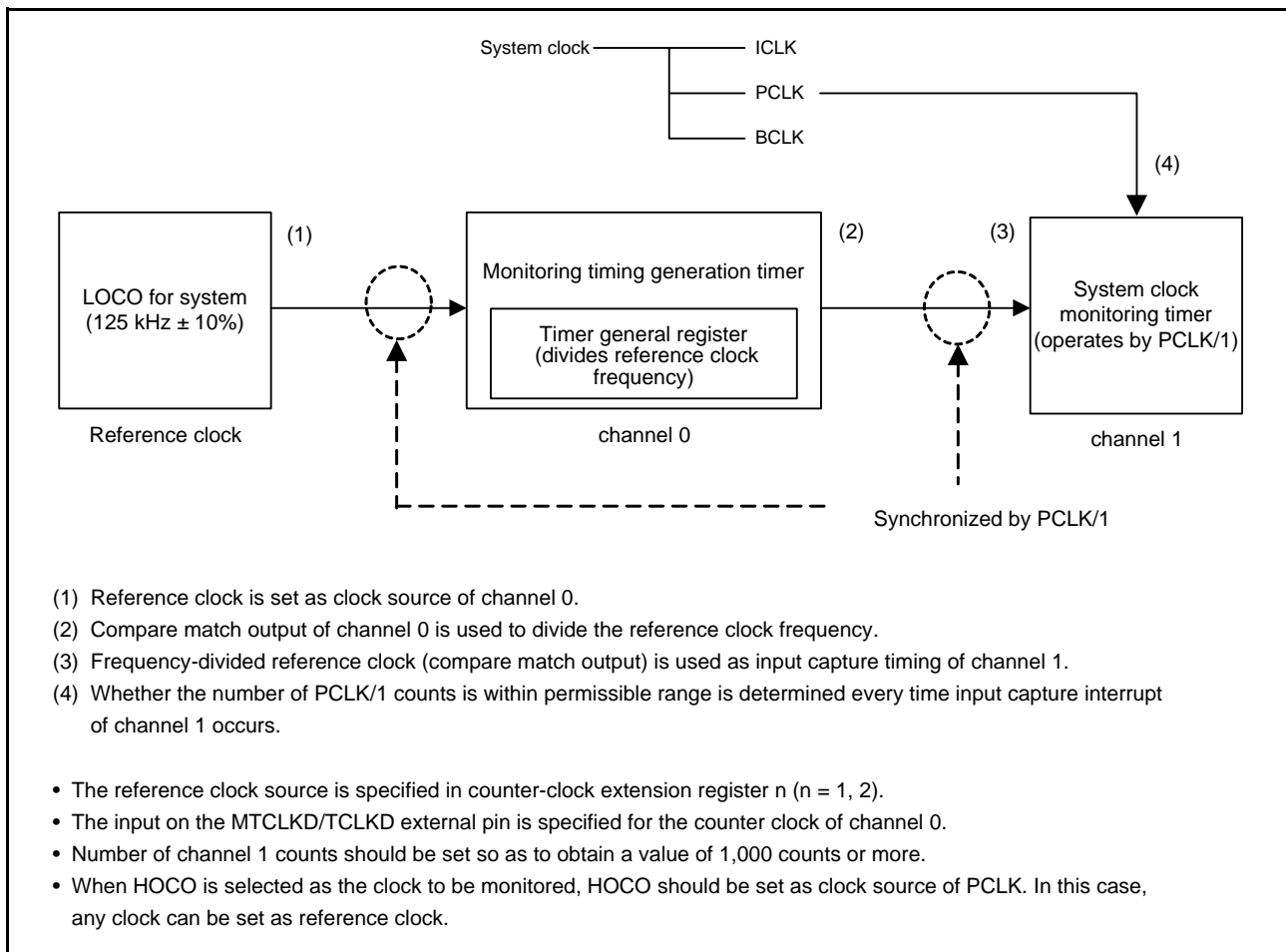


Figure 10.2 Block Diagram of Frequency Measurement Circuit with TPU Unit 0 (System 2)

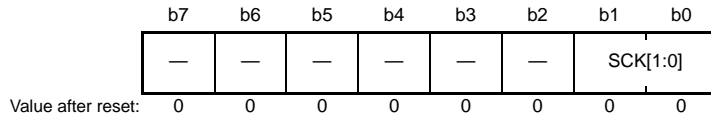


**Figure 10.3 Software Configuration**

## 10.2 Register Descriptions

### 10.2.1 Counter-Clock Extension Register n (SCKn) (n = 1, 2)

Address(es): 0008 C880h: SCK1, 0008 C890h: SCK2



Bit	Symbol	Bit Name	Description	R/W
b1, b0	SCK[1:0]	Reference Clock Select	b1 b2 0 0: MTCLKD/TCLKD external pin input*1 0 1: LOCO 1 0: Main clock 1 1: Sub clock	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Select MTCLKD external pin input for the SCK1 register and TCLKD external pin input for the SCK2 register.

SCK1 is used to select the counter clock of channel 0 of MTU (MTU0) from reference clocks.

SCK2 is used to select the counter clock of channel 0 of TPU unit 0 (TPU0).

In combination with SCKn register setting, select the input on the MTCLKD/TCLKD external pin at the counter clock for the counter of channel 0. Selecting the input on the MTCLKD/TCLKD external pin causes the clock signal selected in the SCKn register to become the counter clock for MTU0.

If the MTCLKD or TCLKD external pin is selected as the source of the clock for counting by the counter of a channel other than channel 0, the actual counter clock for the given channel becomes the signal selected in the SCKn register.

### 10.3 Operation

Figure 10.4 illustrates a flowchart of the frequency measurement circuit. Figure 10.5 illustrates an example of the MTU/TPU operation.

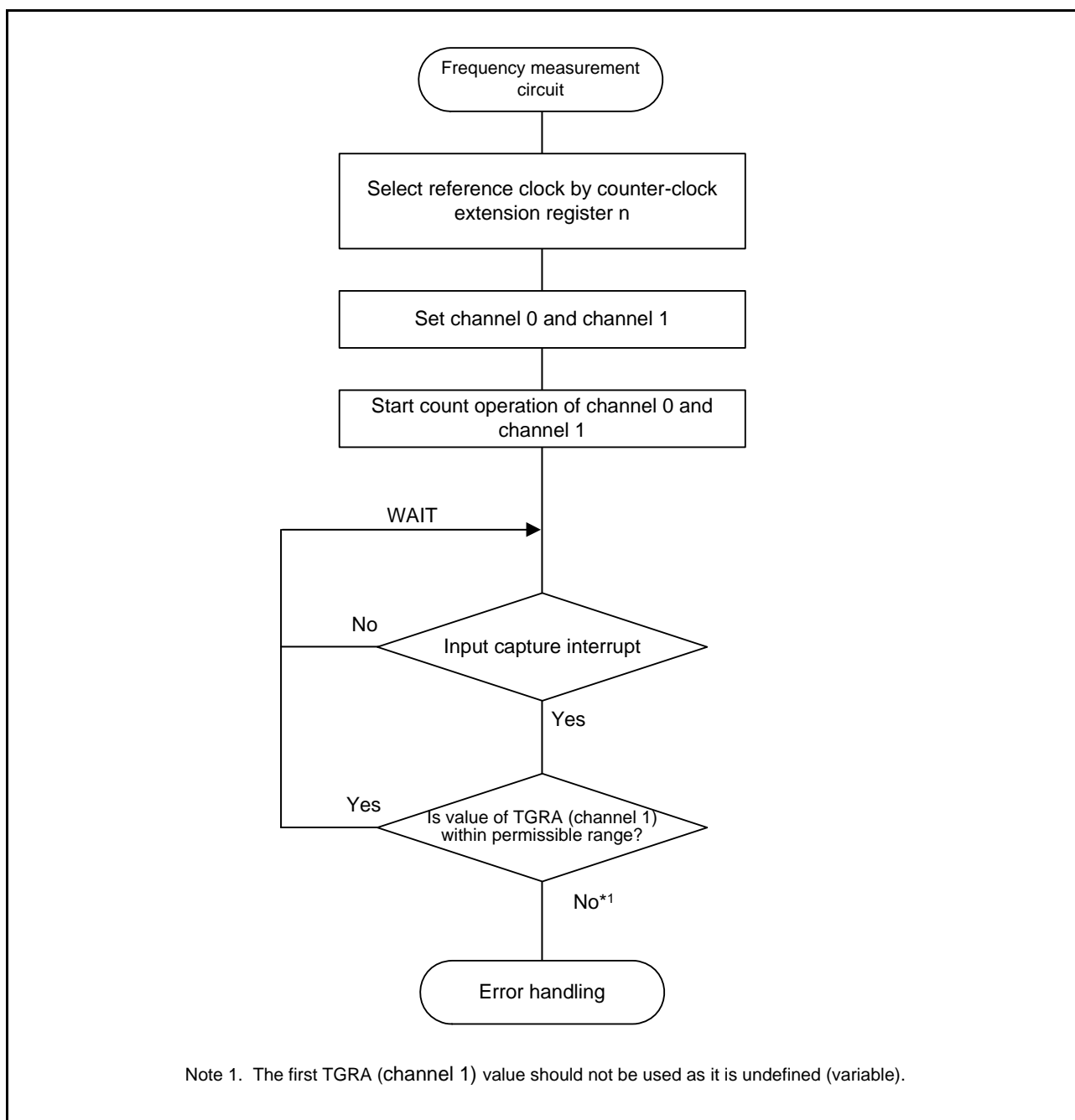


Figure 10.4 Operation Flow of Frequency Measurement Circuit

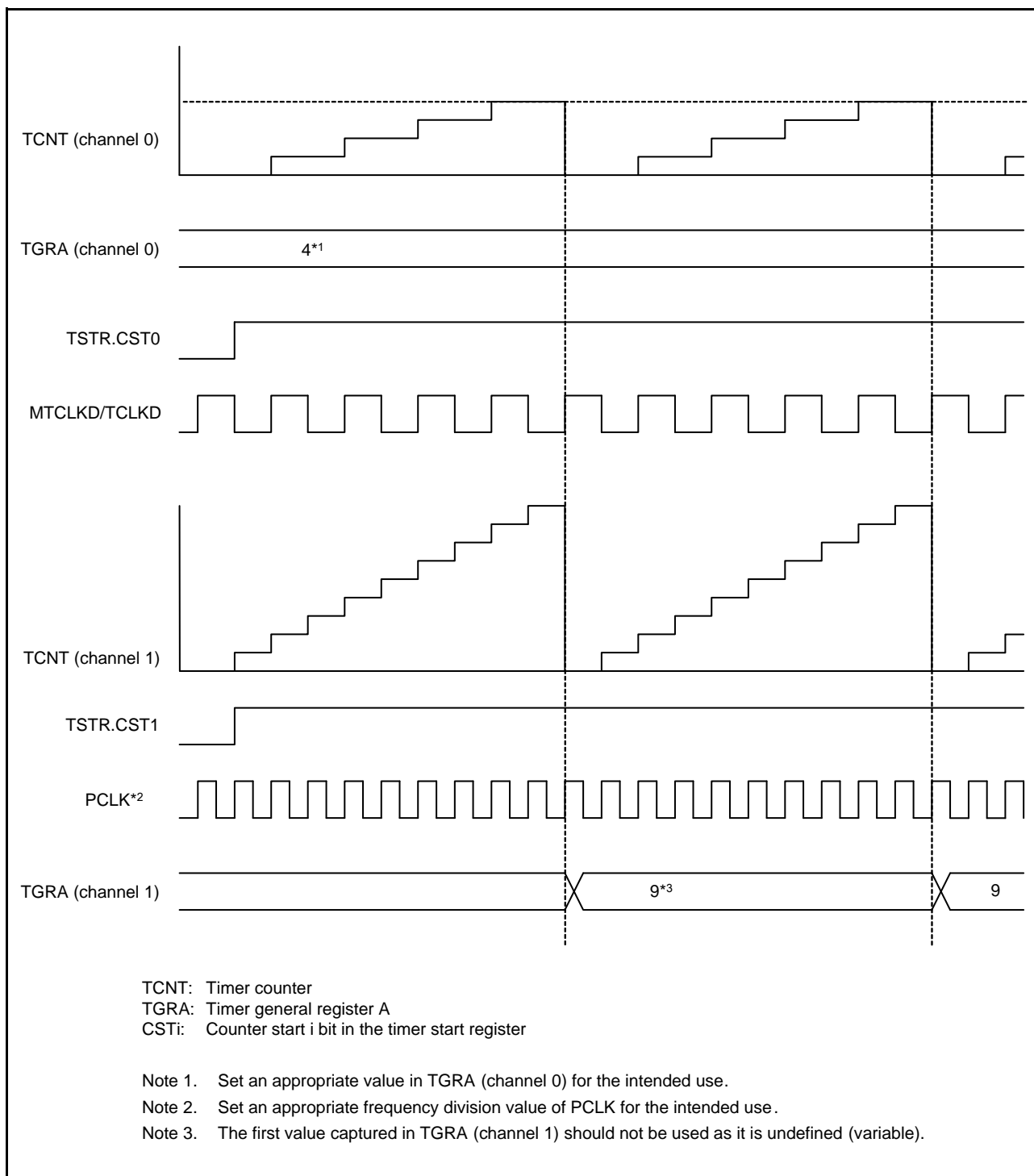


Figure 10.5 Example of MTU/TPU Operation

## 10.4 Usage Notes

### 10.4.1 Setting the Module-Stop Control Registers

The module-stop control registers can be used to place modules in and release modules from the module-stopped state. The several modules that realize frequency measurement are all stopped in their initial state. Releasing the modules from the stopped state makes operations for frequency measurement possible. The method of release is as described below.

Clearing the module-stop bit (MSTPA9) for multi-function timer pulse unit 2 in module stop control register A (MSTPCRA) makes the registers of MTU and of counter-clock extension circuit 1 accessible.

Clearing the module-stop bit (MSTPA13) for 16-bit timer pulse unit 0 (unit 0) in module stop control register A (MSTPCRA) makes the registers of TPU unit 0 and of counter-clock extension circuit 2 accessible.

Clearing the module-stop bit (MSTPC19) for the frequency measurement circuit in module stop control register C (MSTPCRC) enables input of the reference clock signals for counter-clock extension circuits 1 and 2.

For details on the module-stop control registers, refer to section 11, Low Power Consumption.



## 11. Low Power Consumption

### 11.1 Overview

The RX630 Group has several functions for reducing power consumption, including switching of clock signals to reduce power consumption, BCLK output control, stopping modules, functions for low power consumption in normal operation, and transitions to low power consumption states.

Table 11.1 lists the specifications of low power consumption functions, and Table 11.2 lists the conditions to shift to low power consumption modes, states of the CPU and peripheral modules, and the method for canceling each mode. After a reset, this LSI enters the normal program execution state, but modules except for the DMACA, DTC, and RAM do not operate.

**Table 11.1 Specifications of Low Power Consumption Functions**

Item	Specification
Reducing power consumption by switching clock signals	The frequency division ratio is settable independently for the system clock (ICLK), peripheral module clock (PCLKB), external bus clock (BCLK), and flash interface clock (FCLK).*1
BCLK output control function	BCLK output or high-level output can be selected.*1
Module-stop function	Functions can be stopped independently for each peripheral module.
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	<ul style="list-style-type: none"> <li>• Sleep mode</li> <li>• All-module clock stop mode</li> <li>• Software standby mode</li> <li>• Deep software standby mode</li> </ul>
Function for lower operating power consumption	<ul style="list-style-type: none"> <li>• Power consumption can be reduced in normal operation, sleep mode, and all-module clock stop mode by selecting an appropriate operating power consumption control mode according to the operating frequency and operating voltage.</li> <li>• Three operating power control modes               <ul style="list-style-type: none"> <li>High-speed operating mode</li> <li>Low-speed operating mode 1</li> <li>Low-speed operating mode 2</li> </ul> </li> </ul>

Note 1. For details, see section 9, Clock Generation Circuit.

**Table 11.2 Entering and Exiting Low Power Consumption Modes and Operating States in Each Mode (1/2)**

Entering and Exiting Low Power Consumption Modes and Operating States	Sleep Mode	All-Module Clock Stop Mode	Software Standby Mode	Deep Software Standby Mode
Transition condition	Control register + instruction	Control register + instruction	Control register + instruction	Control register + instruction
Canceling method other than reset	Interrupt	Interrupt*1	Interrupt*2	Interrupt*3
State after cancellation*4	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (interrupt processing)	Program execution state (reset processing)
Main clock oscillator	Operating possible	Operating possible	Operating possible*5	Operating possible*5
Sub-clock oscillator	Operating possible	Operating possible	Operating possible*6	Operating possible*6
High-speed on-chip oscillator	Operating possible	Operating possible	Stopped	Stopped
Low-speed on-chip oscillator	Operating possible	Operating possible	Stopped	Stopped
Dedicated on-chip oscillator for the IWDG	Operating possible*7	Operating possible*7	Operating possible*7	Stopped (Undefined)*7
PLL	Operating possible	Operating possible	Stopped	Stopped
CPU	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Undefined)
RAM1 (0001 0000h to 0001 FFFFh)	Operating possible (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Undefined)
RAM0 (0000 0000h to 0000 FFFFh)	Operating possible (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Retained/Undefined)*8
Flash memory	Operating	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)
USB 2.0 function module (USB)	Operating possible	Stopped*9	Stopped*9	Stopped (Retained/Undefined)*10

**Table 11.2 Entering and Exiting Low Power Consumption Modes and Operating States in Each Mode (2/2)**

Entering and Exiting Low Power Consumption Modes and Operating States	Sleep Mode	All-Module Clock Stop Mode	Software Standby Mode	Deep Software Standby Mode
Watchdog timer (WDT)	Stopped (Retained)	Stopped (Retained)	Stopped (Retained)	Stopped (Undefined)
Independent watchdog timer (IWDT)	Operating possible*7	Operating possible*7	Operating possible*7	Stopped (Undefined)*7
Realtime clock (RTC)	Operating possible	Operating possible	Operating possible	Operating possible
8-bit timer (unit 0, unit 1) (TMR)	Operating possible	Operating possible*11	Stopped (Retained)	Stopped (Undefined)
Voltage detection circuit (LVD)	Operating possible	Operating possible	Operating possible	Operating possible*12, *13
Power-on reset circuit	Operating	Operating	Operating	Operating*13
Peripheral modules	Operating possible	Stopped (Retained)	Stopped (Retained)	Stopped (Undefined)
I/O ports	Operating	Retained*14	Retained*15	Retained*15

"Operating possible" means that operating or stopped can be controlled by the control register setting.

"Stopped (Retained)" means that internal register values are retained and internal operations are suspended.

"Stopped (Undefined)" means that internal register values are undefined and power is not supplied to the internal circuit.

- Note 1. "Interrupts" here indicates an external pin interrupt (the NMI or IRQ0 to IRQ15) or any of peripheral interrupts (the 8-bit timer, RTC alarm, RTC interval, IWDT, USB suspend/resume, voltage monitoring 1, voltage monitoring 2, and oscillator-stopped detection interrupts).
- Note 2. "Interrupts" here indicates an external pin interrupt (the NMI or IRQ0 to IRQ15) or any of peripheral interrupts (the RTC alarm, RTC interval, IWDT, USB suspend/resume, voltage monitoring 1, and voltage monitoring 2 interrupts).
- Note 3. "Interrupts" here indicates a certain external pin interrupt source pin (the NMI, IRQ0-DS to IRQ15-DS, SCL2-DS, SDA2-DS, or CRX1-DS) or any of peripheral interrupts (the RTC alarm, RTC interval, USB suspend/resume, voltage monitoring 1, and voltage monitoring 2 interrupts). However, these interrupts are enabled only when the corresponding bit in the deep standby interrupt enable registers i (DPSIERi) (i = 0 to 3) is set to 1. Pins that have "-DS" appended to their names can be used as triggers for release from deep software standby.
- Note 4. This does not include release initiated by the RES# pin reset, power-on reset, voltage monitoring reset, or independent watchdog-timer reset. The transition is to the reset state when release is initiated by one of these reset sources.
- Note 5. Operation or stopping can be selected by the main clock oscillator forced oscillation bit (MOFXIN) in the main clock oscillator forced oscillation control register (MOFCR).
- Note 6. Operation or stopping is selected by the sub-clock control bit (RTCEN) in the RTC control register 3 (RCR3).
- Note 7. Operation or stopping is selected by the setting of the IWDT sleep mode count stop control bit (IWDTSLCSTP) in the option function select register 0 (OFS0) in IWDT auto start mode. If the OFS0.IWDTSLCSTP bit is 0 (disabling stopping of the counter when a transition to low power consumption mode is made), the transition is to software standby mode rather than deep software standby mode. In any mode other than IWDT auto start mode, operation or stopping is selected by the setting of the sleep mode counter stop control bit (SLCSTP) in the IWDT counter stop control register (IWDTCSSTPR). If the IWDTCSSTPR.SLCSTP bit is 0 (disabling stopping of the counter when a transition to low power consumption mode is made), the transition is to software standby mode rather than deep software standby mode.
- Note 8. Retention or undefined is selectable by the setting of the deep cut bits (DEEPCUT[1:0]) in the deep standby control register (DPSBYCR).
- Note 9. Detection of USB resumption is possible.
- Note 10. Disabling or enabling of detection of USB resumption is controllable by the deep cut bits (DEEPCUT[1:0]) in the deep standby control register (DPSBYCR). When detection of USB resumption is enabled, the values of the registers in the USB resume detecting unit are only held even in deep software standby mode.
- Note 11. Stopping or operation is controlled by the module-stop setting bits (MSTPA4 and MSTPA5, respectively) in module-stop control register A (MSTPCRA) for 8-bit timers 0 and 1 (unit 0) and 2 and 3 (unit 1).
- Note 12. If the voltage monitoring 1 circuit mode selection bit in the voltage monitoring 1 circuit control register 0 (LVD1CR0.LVD1RI) or the voltage monitoring 2 circuit mode selection bit in the voltage monitoring 2 circuit control register 0 (LVD2CR0.LVD2RI) is 1, the transition is to software standby mode rather than deep software standby mode.
- Note 13. When the deep cut bits in the deep standby control register (DPSBYCR.DEEPCUT[1:0]) are set to 11b and the LSI enters deep software standby mode, the voltage detection circuit stops and the low power consumption function is enabled.
- Note 14. If pin P53 is being used for the BCLK signal, operation continues with as-is output of BCLK. While the 8-bit timer and RTC are operated, the related pins continue operation.
- Note 15. Retention of levels or placement in the high-impedance state is selectable for the address bus and bus control signals (CS0# to CS7#, RD#, WR0# to WR3#, WR#, BC0# to BC3#, and ALE) by the output port enable bit (OPE) in the standby control register (SBYCR).

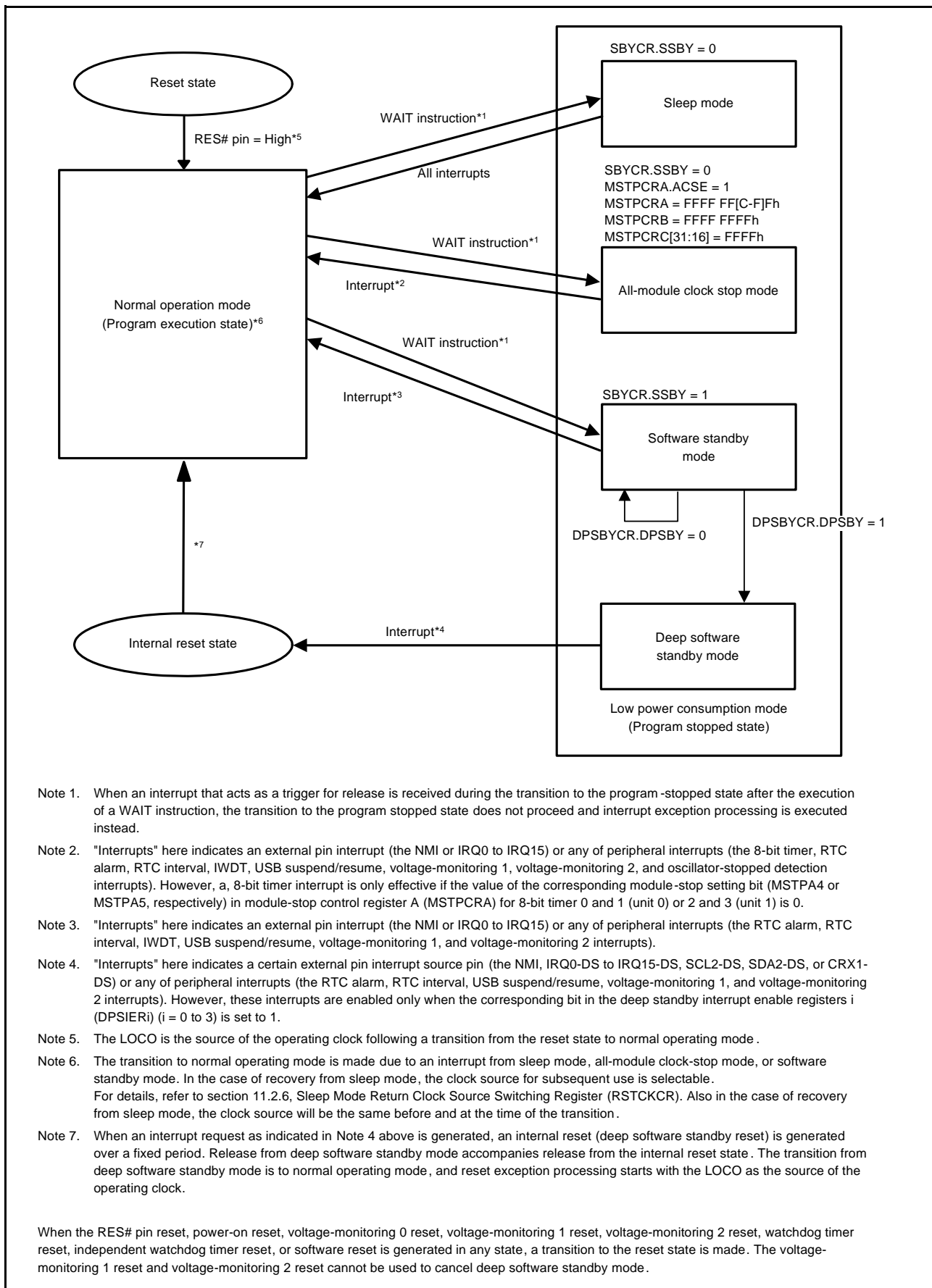


Figure 11.1 Mode Transitions

## 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	OPE	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Value after reset:	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b13 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	OPE	Output Port Enable	0: In software standby mode or deep software standby mode, the address bus and bus control signals are set to the high-impedance state. 1: In software standby mode or deep software standby mode, the address bus and bus control signals retain the output state.	R/W
b15	SSBY	Software Standby	0: Shifts to sleep mode or all-module clock stop mode after the WAIT instruction is executed 1: Shifts to software standby mode after the WAIT instruction is executed	R/W

#### OPE Bit (Output Port Enable)

The OPE bit specifies whether to retain the output of the address bus and bus control signals (CS0# to CS7#, RD#, WR0# to WR3#, WR#, BC0# to BC3#, and ALE) in software standby mode or deep software standby mode, or to set the output to the high-impedance state.

#### 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 LSI enters software standby mode after execution of the WAIT instruction. When the LSI returns to normal mode after an interrupt has initiated release from software standby mode, the SSBY bit remains 1. Write 0 to this bit to clear it.

When the oscillation stop detection function enable bit (OSTDCR.OSTDE) is 1, setting of the SSBY bit is invalid. Even if the SSBY bit is 1, the LSI will enter sleep mode or all module clock stop mode on 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
	ACSE	—	MSTPA 29	MSTPA 28	MSTPA 27	—	—	MSTPA 24	MSTPA 23	—	—	—	MSTPA 19	—	MSTPA 17	—
Value after reset:	0	1	0	0	0	1	1	0	1	1	1	1	1	1	1	1
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	MSTPA 15	MSTPA 14	MSTPA 13	MSTPA 12	MSTPA 11	MSTPA 10	MSTPA 9	—	—	—	MSTPA 5	MSTPA 4	—	—	—	—
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	MSTPA4	8-Bit Timer 3/2 (Unit 1) Module Stop	Target module: TMR3/TMR2 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b5	MSTPA5	8-Bit Timer 1/0 (Unit 0) Module Stop	Target module: TMR1/TMR0 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b8 to b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b9	MSTPA9	Multifunction Timer Pulse Unit 2 Module Stop	Target module: MTU (MTU0 to MTU5) 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b10	MSTPA10	Programmable Pulse Generator (Unit 1) Module Stop	Target module: PPG1 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b11	MSTPA11	Programmable Pulse Generator (Unit 0) Module Stop	Target module: PPG0 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b12	MSTPA12	16-Bit Timer Pulse Unit 1 (Unit 1) Module Stop	Target module: TPU unit 1 (TPU6 to TPU11) 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b13	MSTPA13	16-Bit Timer Pulse Unit 0 (Unit 0) Module Stop	Target module: TPU unit 0 (TPU0 to TPU5) 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b14	MSTPA14	Compare Match Timer (Unit 1) Module Stop	Target module: CMT unit 1 (CMT2, CMT3) 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b15	MSTPA15	Compare Match Timer (Unit 0) Module Stop	Target module: CMT unit 0 (CMT0, CMT1) 0: The module-stop state is canceled 1: Transition to the module-stop state is made	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 (Unit 1) Module Stop	Target module: S12AD1 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b18	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b19	MSTPA19	D/A Converter Module Stop	Target module: DA 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b22 to b20	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b23	MSTPA23	10-bit A/D Converter Module Stop	Target module: AD 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W

Bit	Symbol	Bit Name	Description	R/W
b24	MSTPA24	Module Stop A24	Writing to and reading from this bit is enabled. When a transition to all-module clock stop mode is made, be sure that 1 has been written to this bit.	R/W
b26, b25	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b27	MSTPA27	Module Stop A27	Writing to and reading from this bit is enabled. When a transition to all-module clock stop mode is made, be sure that 1 has been written to this bit.	R/W
b28	MSTPA28	DMA Controller/Data Transfer Controller Module Stop	Target module: DMAC/DTC 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b29	MSTPA29	Module Stop A29	Writing to and reading from this bit is enabled. When a transition to all-module clock stop mode is made, be sure that 1 has been written to this bit.	R/W
b30	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b31	ACSE	All-Module Clock Stop Mode Enable	0: All-module clock stop mode is disabled 1: All-module clock stop mode is enabled	R/W

**ACSE Bit (All-Module Clock Stop Mode Enable)**

The ACSE bit enables or disables a transition to all-module clock stop mode.

With the ACSE bit set to 1, when the CPU executes the WAIT instruction with the SBYCR.SSBY bit, MSTPCRA, MSTPCRB, and MSTPCRC satisfying specified conditions, the LSI enters all-module clock stop mode. For details, see section 11.6.2, All-Module Clock Stop Mode.

Whether to stop the 8-bit timers or not can be selected by the MSTPA5 and MSTPA4 bits.

When the MSTPCRA.ACSE bit = 0 while the SBYCR.SSBY = 0, a transition to sleep mode is made after the WAIT instruction is executed.

**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
MSTPB 31	MSTPB 30	MSTPB 29	MSTPB 28	MSTPB 27	MSTPB 26	MSTPB 25	MSTPB 24	MSTPB 23	—	MSTPB 21	MSTPB 20	MSTPB 19	—	MSTPB 17	MSTPB 16
Value after reset:	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
—	—	—	—	—	—	—	MSTPB 8	—	—	—	MSTPB 4	—	MSTPB 2	MSTPB 1	MSTPB 0
Value after reset:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	MSTPB0	CAN Module 0 Module Stop*1	Target module: CAN0 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b1	MSTPB1	CAN Module 1 Module Stop*1	Target module: CAN1 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b2	MSTPB2	CAN Module 2 Module Stop*1	Target module: CAN2 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W

Bit	Symbol	Bit Name	Description	R/W
b4	MSTPB4	Serial Communication Interface SCId Module Stop	Target module: SCId (SCI12) 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b7 to b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b8	MSTPB8	Temperature Sensor Module Stop	Target module: Temperature sensor 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b15 to b9	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b16	MSTPB16	Serial Peripheral Interface 1 Module Stop	Target module: RSP11 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b17	MSTPB17	Serial Peripheral Interface 0 Module Stop	Target module: RSP10 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b18	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b19	MSTPB19	Universal Serial Bus Interface (Port 0) Module Stop*2	Target module: USB0 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b20	MSTPB20	I <sup>2</sup> C Bus Interface 1 Module Stop	Target module: RIIC1 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b21	MSTPB21	I <sup>2</sup> C Bus Interface 0 Module Stop	Target module: RIIC0 0: The module-stop state is canceled 1: Transition to the module-stop state is made	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: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b24	MSTPB24	Serial Communication Interface 7 Module Stop	Target module: SCI7 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b25	MSTPB25	Serial Communication Interface 6 Module Stop	Target module: SCI6 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b26	MSTPB26	Serial Communication Interface 5 Module Stop	Target module: SCI5 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b27	MSTPB27	Serial Communication Interface 4 Module Stop	Target module: SCI4 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b28	MSTPB28	Serial Communication Interface 3 Module Stop	Target module: SCI3 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b29	MSTPB29	Serial Communication Interface 2 Module Stop	Target module: SCI2 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b30	MSTPB30	Serial Communication Interface 1 Module Stop	Target module: SCI1 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b31	MSTPB31	Serial Communication Interface 0 Module Stop	Target module: SC10 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W

Note 1. The MSTPBi bit should be rewritten while the oscillation of the clock controlled by MSTPBi is stabilized. For entering software standby mode after rewriting the MSTPBi bit, wait for two CAN clock (CANCLK) cycles after rewriting, and execute the WAIT instruction (i = 0 to 2).

Note 2. For entering software standby mode after rewriting the MSTPB19 bit, wait for two USB clock (UCLK) cycles after rewriting, and execute the WAIT instruction.

## 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
	—	—	—	—	MSTPC 27	MSTPC 26	MSTPC 25	MSTPC 24	—	MSTPC 22	—	—	MSTPC 19	MSTPC 18	MSTPC 17	MSTPC 16
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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	MSTPC 1	MSTPC 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
b0	MSTPC0	RAM0 Module Stop*1	Target module: RAM0 (0000 0000h to 0000 FFFFh) 0: RAM0 operating 1: RAM0 stopped	R/W
b1	MSTPC1	RAM1 Module Stop*1	Target module: RAM1 (0001 0000h to 0001 FFFFh) 0: RAM1 operating 1: RAM1 stopped	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	MSTPC16	I <sup>2</sup> C Bus Interface 3 Module Stop	Target module: RIIC3 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b17	MSTPC17	I <sup>2</sup> C Bus Interface 2 Module Stop	Target module: RIIC2 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b18	MSTPC18	IEBUS Module Stop*2	Target module: IEBUS 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b19	MSTPC19	Frequency Measurement Circuit Module Stop	Target module: MCK 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b21, b20	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b22	MSTPC22	Serial Peripheral Interface 2 Module Stop	Target module: RSPI2 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b23	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b24	MSTPC24	Serial Communication Interface 11 Module Stop	Target module: SCI11 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b25	MSTPC25	Serial Communication Interface 10 Module Stop	Target module: SCI10 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b26	MSTPC26	Serial Communication Interface 9 Module Stop	Target module: SCI9 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b27	MSTPC27	Serial Communication Interface 8 Module Stop	Target module: SCI8 0: The module-stop state is canceled 1: Transition to the module-stop state is made	R/W
b31 to b28	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

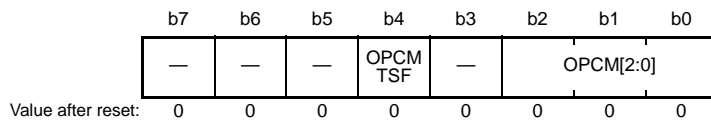
Note 1. The MSTPC1 or MSTPC0 bit should not be set to 1 during access to the corresponding on-chip RAM. The corresponding RAM should not be accessed while the MSTPC1 or MSTPC0 bit is set to 1.

Note 2. For entering software standby mode after rewriting the MSTPC18 bit, wait for two IEBUS clock (IECLK) cycles after rewriting, and execute the WAIT instruction.



### 11.2.5 Operating Power Control Register (OPCCR)

Address(es): 0008 00A0h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	OPCM[2:0]	Operating Power Control Mode Select	b2 b1 b0 0 0 0: High-speed operating mode 1 1 0: Low-speed operating mode 1 1 1 1: Low-speed operating mode 2 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	<ul style="list-style-type: none"> <li>Read</li> <li>0: Transition completed</li> <li>1: During transition</li> <li>Write</li> </ul> The write value should be 0.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

OPCCR is used to reduce power consumption in normal operating mode, sleep mode, and all-module clock stop mode. Power consumption can be reduced according to the operating frequency and operating voltage to be used by the OPCCR setting.

OPCCR should not be modified in the following cases:

- When the operating power control mode transition status flag (OPCMTSF) is 1 (operating power control mode switching is in progress)
- When the ROM P/E mode entry bit *i* in the flash P/E mode entry register (FENTRYR.FENTRY<sub>*i*</sub>) is 1 (ROM P/E mode, E2 DataFlash P/E mode) (*i* = 0 to 3, D)
- Period from the time of WAIT instruction execution for a sleep mode transition, to return from sleep mode to normal operation

Writing to the flash memory while it is being programmed or erased is impossible because write access to the OPCCR register is not allowed.

For the procedure to use in shifting to operating power control mode, refer to [section 11.5, Function for Lower Operating Power Consumption](#).

On return from software standby, the chip enters the high-speed operating mode. Even if a WAIT instruction is executed, if release from software standby precedes completion of the transition, the mode remains the same as before execution of the WAIT instruction. If this creates a problem, set the OPCCR.OPCM[2:0] bits to 000b during processing of the return interrupt.

#### 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 all-module clock stop mode.

Table 11.3 shows the operating power control modes along with the operating frequency ranges, operating voltage ranges, and power consumption.

**Table 11.3 Relationship between Operating Power Control Mode, Operating Range, and Power Consumption**

Operating Power Control Mode	OPCM[2:0] Bits	Operating Frequency Range					Operating Voltage Range		Power Consumption
		Flash memory Read				Flash memory P/E	Flash memory Read	Flash memory P/E	
		ICLK	FCLK	PCLKB	BCLK	FCLK			
High-speed operating mode	000b	100 MHz max	50 MHz max	50 MHz max	50 MHz max	4 MHz to 50 MHz	2.7 V to 3.6 V	2.7 V to 3.6 V	High ↓ Low
Low-speed operating mode 1	110b	1 MHz max	1 MHz max	1 MHz max	1 MHz max	P/E disabled	2.7 V to 3.6 V	P/E disabled	
Low-speed operating mode 2	111b	32 kHz to 125 kHz	32 kHz to 125 kHz	125 kHz max	125 kHz max	P/E disabled	2.7 V to 3.6 V	P/E disabled	

Each operating power control mode is described below.

- High-speed operating mode

This mode allows high-speed operation.

During reading the flash memory (FLASH), the maximum operating frequency of ICLK is 100 MHz, and that of FCLK, PCLKB, and BCLK is 50 MHz. During Flash memory programming/erasure (P/E), the FCLK can be operated in the operating frequency from 4 MHz to 50 MHz. The operating voltage is in the range of 2.7 to 3.6 V both for Flash memory read and P/E. After a reset cancellation, this LSI is activated in this mode.

- Low-speed operating mode 1

This mode reduces power consumption for low-speed operation.

During reading the flash memory (FLASH), the maximum operating frequency of ICLK, FCLK, PCLKB and BCLK is 1 MHz. The operating voltage is in the range of 2.7 to 3.6 V.

In low-speed operating mode 1, P/E operation of Flash memory is disabled, and writing to set the PLLCR2.PLEN bit to 0 (PLL operation) is prohibited.

In this mode, lower power consumption is possible than in high-speed operating mode when the same operation is performed under the same conditions (operating frequency, operating voltage).

- Low-speed operating mode 2

As compare to low-speed operating mode 1, this mode reduces power consumption for low-speed operation.

During reading the flash memory (FLASH), the maximum operating frequency of ICLK, FCLK, PCLKB and BCLK is 125 kHz, and the minimum operating frequency of ICLK and FCLK is 32 kHz. The operating voltage is in the range of 2.7 to 3.6 V.

The following restrictions apply when low-speed operating mode 2 is selected:

- P/E operations for flash memory are prohibited.
- Reading of data flash is prohibited.
- Using the PLL or HOCO is prohibited.
- Using the oscillation stop detection function of the main clock oscillator is prohibited.

In this mode, lower power consumption is possible than in low-speed operating mode 1 when the same operation is performed under the same conditions (operating frequency, operating voltage).

When the clock source select bits in the system clock control register 3 (SCKCR3.CKSEL[2:0]) are 011b (sub-clock oscillator selected) and the system clock select bits (ICK[3:0]) or FlashIF clock select bits (FCK[3:0]) in the system clock control register (SCKCR) are not 0000b (no frequency dividing), OPCM[2:0] bits cannot be set to 111b.

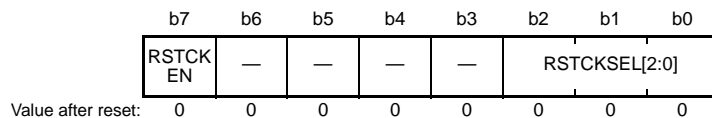
When the PLL stop control bit (PLLCR2.PLEN) in PLL control register 2 is 0 (PLL operation), writing 110b (low-speed operating mode 1) and 111b (low-speed operating mode 2) to the OPCM[2:0] bits is not possible.

### OPCMTSF Flag (Operating Power Control Mode Transition Status Flag)

The OPCMTSF flag indicates the switching control state when the operating power control mode is switched. When a write access is attempted to change the operating power control mode, the OPCMTSF flag is set to 1. The flag becomes 0 after a transition to the changed control mode is completed. Make sure that the OPCMTSF flag is 0 (completed operating power control mode transition) before the next processing.

### 11.2.6 Sleep Mode Return Clock Source Switching Register (RSTCKCR)

Address(es): 0008 00A1h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RSTCKSEL [2:0]	Sleep Mode Return Clock Source Select	b2 b0 0 0 1: HOCO is selected 0 1 0: Main clock oscillator is selected Settings other than above are prohibited while the RSTCKEN bit is 1.	R/W
b6 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	RSTCKEN	Sleep Mode Return Clock Source Switching Enable	0: Clock source switching at sleep mode cancellation is disabled 1: Clock source switching at sleep mode cancellation is enabled	R/W

RSTCKCR is used to control clock source switching at cancellation of sleep mode.

When operation is restored from sleep mode by setting RSTCKCR, the main clock oscillator stop bit in the main clock oscillator control register (MOSCCR.MOSTP) and HOCO stop bit in the high-speed on-chip oscillator control register (HOCOCCR.HCSTP) corresponding to the clock source to be used on restoration are automatically modified to the operating state. The value of RSTCKSEL[2:0] bits is automatically reloaded to the clock source select bits in the system clock control register 3 (SCKCR3.CKSEL[2:0]).

When the setting of register RSTCKCR is for the HOCO to be used in recovery from sleep mode, the power supply for the HOCO is not automatically switched on. If the HOCO to be used in recovery from sleep mode, the power supply for the HOCO must be on when the transition to sleep mode takes place.

When return from sleep mode is made while clock source switching at sleep mode cancellation is enabled (RSTCKCR.RSTCKEN is 1), and operating power control mode select bits (OPCCR.OPCM[2:0]) are set so as to select low-speed operating mode 1 (110b) or low-speed operating mode 2 (111b), the OPCCR.OPCM[2:0] bits are automatically switched to high-speed mode (000b).

#### RSTCKSEL[2:0] Bits (Sleep Mode Return Clock Source Select)

The RSTCKSEL[2:0] bits select the clock source to be used when sleep mode is canceled. The clock source selected by the RSTCKSEL[2:0] bits is enabled only when the RSTCKEN bit is 1.

#### RSTCKEN Bit (Sleep Mode Return Clock Source Switching Enable)

The RSTCKEN bit enables or disables clock source switching when sleep mode is canceled.

When sleep mode is canceled, the clock source should be switched only when LOCO or sub clock is selected as a clock for a transition to sleep mode. To make a transition to sleep mode with HOCO, main clock, or PLL selected as the clock source, the RSTCKEN bit should not be set to 1.

## 11.2.7 Main Clock Oscillator Wait Control Register (MOSCWTCR)

Address(es): 0008 00A2h



Bit	Symbol	Bit Name	Description	R/W																																																			
b4 to b0	MSTS[4:0]	Main Clock Oscillator Wait Time Select	<table style="width: 100%; border: none;"> <tr> <td style="width: 10%;">b4</td> <td style="width: 10%;">b0</td> <td></td> </tr> <tr> <td>0 0 0 0</td> <td>0</td> <td>Waiting time = 2 cycles</td> </tr> <tr> <td>0 0 0 0</td> <td>1</td> <td>Waiting time = 4 cycles</td> </tr> <tr> <td>0 0 0 1</td> <td>0</td> <td>Waiting time = 8 cycles</td> </tr> <tr> <td>0 0 0 1</td> <td>1</td> <td>Waiting time = 16 cycles</td> </tr> <tr> <td>0 0 1 0</td> <td>0</td> <td>Waiting time = 32 cycles</td> </tr> <tr> <td>0 0 1 0</td> <td>1</td> <td>Waiting time = 64 cycles</td> </tr> <tr> <td>0 0 1 1</td> <td>0</td> <td>Waiting time = 512 cycles</td> </tr> <tr> <td>0 0 1 1</td> <td>1</td> <td>Waiting time = 1024 cycles</td> </tr> <tr> <td>0 1 0 0</td> <td>0</td> <td>Waiting time = 2048 cycles</td> </tr> <tr> <td>0 1 0 0</td> <td>1</td> <td>Waiting time = 4096 cycles</td> </tr> <tr> <td>0 1 0 1</td> <td>0</td> <td>Waiting time = 16384 cycles</td> </tr> <tr> <td>0 1 0 1</td> <td>1</td> <td>Waiting time = 32768 cycles</td> </tr> <tr> <td>0 1 1 0</td> <td>0</td> <td>Waiting time = 65536 cycles</td> </tr> <tr> <td>0 1 1 0</td> <td>1</td> <td>Waiting time = 131072 cycles</td> </tr> <tr> <td>0 1 1 1</td> <td>0</td> <td>Waiting time = 262144 cycles</td> </tr> <tr> <td>0 1 1 1</td> <td>1</td> <td>Waiting time = 524288 cycles</td> </tr> </table> Settings other than above are prohibited.	b4	b0		0 0 0 0	0	Waiting time = 2 cycles	0 0 0 0	1	Waiting time = 4 cycles	0 0 0 1	0	Waiting time = 8 cycles	0 0 0 1	1	Waiting time = 16 cycles	0 0 1 0	0	Waiting time = 32 cycles	0 0 1 0	1	Waiting time = 64 cycles	0 0 1 1	0	Waiting time = 512 cycles	0 0 1 1	1	Waiting time = 1024 cycles	0 1 0 0	0	Waiting time = 2048 cycles	0 1 0 0	1	Waiting time = 4096 cycles	0 1 0 1	0	Waiting time = 16384 cycles	0 1 0 1	1	Waiting time = 32768 cycles	0 1 1 0	0	Waiting time = 65536 cycles	0 1 1 0	1	Waiting time = 131072 cycles	0 1 1 1	0	Waiting time = 262144 cycles	0 1 1 1	1	Waiting time = 524288 cycles	R/W
b4	b0																																																						
0 0 0 0	0	Waiting time = 2 cycles																																																					
0 0 0 0	1	Waiting time = 4 cycles																																																					
0 0 0 1	0	Waiting time = 8 cycles																																																					
0 0 0 1	1	Waiting time = 16 cycles																																																					
0 0 1 0	0	Waiting time = 32 cycles																																																					
0 0 1 0	1	Waiting time = 64 cycles																																																					
0 0 1 1	0	Waiting time = 512 cycles																																																					
0 0 1 1	1	Waiting time = 1024 cycles																																																					
0 1 0 0	0	Waiting time = 2048 cycles																																																					
0 1 0 0	1	Waiting time = 4096 cycles																																																					
0 1 0 1	0	Waiting time = 16384 cycles																																																					
0 1 0 1	1	Waiting time = 32768 cycles																																																					
0 1 1 0	0	Waiting time = 65536 cycles																																																					
0 1 1 0	1	Waiting time = 131072 cycles																																																					
0 1 1 1	0	Waiting time = 262144 cycles																																																					
0 1 1 1	1	Waiting time = 524288 cycles																																																					
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W																																																			

MOSCWTCR is used to control the time to wait until supplying the output of the main clock oscillator to the internal circuits. Supply of the main clock signal within the MCU starts after counting of the number of cycles of the main clock specified in the MOSCWTCR register plus 16384.

Set the MSTS[4:0] bits so that the waiting time is at least as long as the main clock oscillator stabilization time ( $t_{\text{MAINOSC}}$ ). Only use the main clock after the main clock oscillation settling time ( $t_{\text{MAINOSCWT}}$ ) has elapsed. Regarding the main clock oscillation settling time, refer to Table 45.11.

The waiting time is not required when the main clock is being externally input to the main clock oscillator. Only write new values to MOSCWTCR while the MOSCCR.MOSTP bit is 1; do not write to MOSCWTCR if this is not the case.

**Example:** When oscillation is at 12 MHz and the crystal resonator is to be used after 10 ms (= 10000 $\mu$ s) of stabilization time

To satisfy the relation  $\text{waiting time} \geq t_{\text{MAINOSC}} \times f_{\text{MAIN}} = 10000 [\mu\text{s}] \times 12 [\text{MHz}] = 120000 [\text{cycles}]$ , set the MSTS[4:0] bits to 01101b (131072 cycles).

At this time, when the main clock stabilization time is calculated from the formula in Table 45.11,

$$\begin{aligned}
 t_{\text{MAINOSCWT}} &= t_{\text{MAINOSC}} + \frac{n + 16384}{f_{\text{MAIN}}} \\
 &= 10000[\mu\text{s}] + \frac{131072[\text{cycles}] + 16384}{12[\text{MHz}]} \\
 &= 22288[\mu\text{s}]
 \end{aligned}$$

22288  $\mu$ s are required from the time the main clock starts oscillating until it is usable.

### 11.2.8 Sub-Clock Oscillator Wait Control Register (SOSCWTCR)

Address(es): 0008 00A3h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	SSTS[4:0]	Sub-Clock Oscillator Wait Time Select	b4    b0 0 0 0 0 0: Waiting time = 2 cycles 0 0 0 0 1: Waiting time = 4 cycles 0 0 0 1 0: Waiting time = 8 cycles 0 0 0 1 1: Waiting time = 16 cycles 0 0 1 0 0: Waiting time = 32 cycles 0 0 1 0 1: Waiting time = 64 cycles 0 0 1 1 0: Waiting time = 512 cycles 0 0 1 1 1: Waiting time = 1024 cycles 0 1 0 0 0: Waiting time = 2048 cycles 0 1 0 0 1: Waiting time = 4096 cycles 0 1 0 1 0: Waiting time = 16384 cycles 0 1 0 1 1: Waiting time = 32768 cycles 0 1 1 0 0: Waiting time = 65536 cycles 0 1 1 0 1: Waiting time = 131072 cycles 0 1 1 1 0: Waiting time = 262144 cycles 0 1 1 1 1: Waiting time = 524288 cycles Settings other than above are prohibited.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Products tagged with “\*2” in Table 1.3 List of Products, have different initial values.

SOSCWTCR is used to control the time to wait until the output of the sub-clock clock oscillator is supplied to the internal circuits. Supply of the sub-clock clock signal to the MCU internally starts after the number of cycles of the sub-clock specified in the SOSCWTCR register have been counted and the sub-clock clock oscillation settling time (tSUBOSCWT0) has also elapsed.

Set the SSTS[4:0] bits so that the waiting time is at least as long as the sub-clock oscillator stabilization time (tSUBOSC). However, for the products tagged with “\*2” in Table 1.3, List of Products, the waiting time consists of the number of cycles set in this register plus a specified offset time period (the sub-clock oscillation stabilization wait offset time).

Once the sub-clock has started oscillating, only use it after the oscillation stabilization waiting time (tSUBOSCWT) has elapsed. For the sub-clock oscillation stabilization waiting time, refer to Table 45.10.

New values can only be written to the SOSCWTCR bit while the SOSCCR.SOSTP bit is 1; do not write to SOSCWTCR if this is not the case.

Example: When oscillation is at 32.768 kHz and the crystal resonator is to be used after 2 s (= 2000 ms) of stabilization time

- For products not tagged with “\*2” in Table 1.3

To satisfy the relation  $\text{waiting time} \geq t_{\text{SUBOSC}} \times f_{\text{SUB}} = 2000 \text{ [ms]} \times 32.768 \text{ [kHz]} = 65536 \text{ [cycles]}$ , set the SSTS[4:0] bits to 01100b (65536 cycles).

At this time, when the sub-clock stabilization time is calculated from the formula in Table 45.11,

$$\begin{aligned} t_{\text{SUBOSCWT}} &= t_{\text{SUBOSC}} + \frac{n}{f_{\text{SUB}}} \\ &= 2000[\text{ms}] + \frac{65536[\text{cycles}]}{32.768[\text{kHz}]} \\ &= 4000[\text{ms}] \end{aligned}$$

4 s are required from the time the sub-clock starts oscillating until it is usable.

- For products tagged with "\*2" in Table 1.3

To satisfy the relation waiting time  $\geq (t_{\text{SUBOSC}} - f_{\text{SUBOSCWTO}}) \times f_{\text{SUB}} = (2000 - 1800) [\text{ms}] \times 32.768 [\text{kHz}] = 6553.6 [\text{cycles}]$ , set the SSTS[4:0] bits to 01010b (16384 cycles).

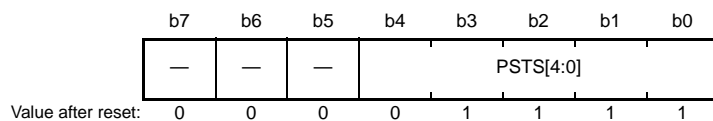
At this time, when the sub-clock stabilization time is calculated from the formula in Table 45.11,

$$\begin{aligned} t_{\text{SUBOSCWT}} &= \max(t_{\text{SUBOSC}}, t_{\text{SUBOSCWTO}}) + \frac{n}{f_{\text{SUB}}} \\ &= \max(2000[\text{ms}], 2600[\text{ms}]) + \frac{16384[\text{cycles}]}{32.768[\text{kHz}]} \\ &= 3100[\text{ms}] \end{aligned}$$

3.1 s are required from the time the sub-clock starts oscillating until it is usable.

### 11.2.9 PLL Wait Control Register (PLLWTCR)

Address(es): 0008 00A6h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	PSTS[4:0]	PLL Wait Time Select	b4    b0 0 0 0 0: Waiting time = 16 cycles 0 0 0 1: Waiting time = 32 cycles 0 0 0 1 0: Waiting time = 64 cycles 0 0 0 1 1: Waiting time = 512 cycles 0 0 1 0 0: Waiting time = 1024 cycles 0 0 1 0 1: Waiting time = 2048 cycles 0 0 1 1 0: Waiting time = 4096 cycles 0 0 1 1 1: Waiting time = 16384 cycles 0 1 0 0 0: Waiting time = 32768 cycles 0 1 0 0 1: Waiting time = 65536 cycles 0 1 0 1 0: Waiting time = 131072 cycles 0 1 0 1 1: Waiting time = 262144 cycles 0 1 1 0 0: Waiting time = 524288 cycles 0 1 1 0 1: Waiting time = 1048576 cycles 0 1 1 1 0: Waiting time = 2097152 cycles 0 1 1 1 1: Waiting time = 4194304 cycles Settings other than above are prohibited.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PLLWTCR is used to control the time to wait until the output of the PLL is supplied to the internal circuits.

Supply of the PLL clock signal within the MCU starts after counting of the number of cycles of the PLL clock specified in the PLLWTCR register + 131072.

Set the PSTS[4:0] bits so that the waiting time is greater than the PLL clock stabilization time ( $t_{PLL1}$  or  $t_{PLL2}$ ). Only use the PLL clock after waiting for the PLL clock oscillation stabilization waiting time ( $t_{PLLWT1}$  or  $t_{PLLWT2}$ ) to elapse after oscillation of the PLL has started. See Table 46.9 for the PLL clock oscillation stabilization waiting time. New values can only be written to the PLLWTCR register while the PLLCR2.PLEN bit is 1 (stopping the PLL). Do not write to the register if this is not the case.

**Example:** When oscillation is at 12 MHz, the crystal resonator is to be used after 10 ms (= 10000 $\mu$ s) of stabilization time, and the PLL is to oscillate at 200 MHz

- When the PLL starts operating after oscillation of the main clock has become stable

To satisfy the relation waiting time  $\geq t_{PLL} \times f_{PLL} = 500 [\mu\text{s}] \times 200 [\text{MHz}] = 100000 [\text{cycles}]$ , set the PSTS[4:0] bits to 01010b (131072 cycles).

At this time, when the PLL clock stabilization time is calculated from the formula in Table 45.11,

$$\begin{aligned}
 t_{PLLWT1} &= t_{PLL1} + \frac{n + 131072}{f_{PLL}} \\
 &= 500[\mu\text{s}] + \frac{131072[\text{cycles}] + 131072}{200[\text{MHz}]} \\
 &= 1810.72[\mu\text{s}]
 \end{aligned}$$

1811  $\mu$ s of waiting are required from the time the PLL clock starts oscillating until it is usable.

- When the PLL starts operating before oscillation of the main clock has become stable

To satisfy the relation waiting time  $\geq (t_{\text{MAINOSC}} + t_{\text{PLL1}}) \times f_{\text{PLL}} = (10000 [\mu\text{s}] + 500 [\mu\text{s}]) \times 200 [\text{MHz}] = 2100000$  [cycles], set the PSTS[4:0] bits to 01111b (4194304 cycles).

At this time, when the PLL clock stabilization time is calculated from the formula in Table 45.11,

$$\begin{aligned}
 t_{\text{PLLWT2}} &= t_{\text{PLL2}} + \frac{n + 131072}{f_{\text{PLL}}} \\
 &= t_{\text{MAINOSC}} + t_{\text{PLL1}} + \frac{n + 131072}{f_{\text{PLL}}} \\
 &= 10000[\mu\text{s}] + 500[\mu\text{s}] + \frac{4194304[\text{cycles}] + 131072}{200[\text{MHz}]} \\
 &= 32126.88[\mu\text{s}]
 \end{aligned}$$

about 32.13 ms of waiting are required from the time the PLL clock starts oscillating until it is usable.



## 11.2.10 Deep Standby Control Register (DPSBYCR)

Address(es): 0008 C280h

b7	b6	b5	b4	b3	b2	b1	b0
DPSBY	IOKEEP P	—	—	—	—	DEEPCUT [1:0]	
Value after reset:	0	0	0	0	0	0	1

Bit	Symbol	Bit Name	Description	R/W
b1, b0	DEEPCUT [1:0]	Deep Cut	b1 b0 0 0: Power is supplied to the RAM (RAM0*1) and USB resume detecting unit in deep software standby mode 0 1: Power is not supplied to the RAM (RAM0*1) and USB resume detecting unit in deep software standby mode 1 0: Setting prohibited 1 1: Power is not supplied to the RAM (RAM0*1) and USB resume detecting unit in deep software standby mode. In addition, LVD is stopped and the low power consumption function in a power-on reset circuit is enabled.	R/W
b5 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	IOKEEP	I/O Port Retention	0: Deep software standby mode and I/O port retention are canceled simultaneously. 1: The I/O port state is retained even after deep software standby mode is canceled. Then, writing 0 to the IOKEEP bit cancels the I/O port retention.	R/W
b7	DPSBY	Deep Software Standby	SSBY b7 0 0: Transition to sleep mode or all-module clock stop mode is made after the WAIT instruction is executed 0 1: Transition to sleep mode or all-module clock stop mode is made after the WAIT instruction is executed 1 0: Transition to software standby mode is made after the WAIT instruction is executed 1 1: Transition to deep software standby mode is made after the WAIT instruction is executed	R/W

Note 1. For the RAM address space, see Table 11.2.

DPSBYCR is not initialized by the internal reset signal that is the source to cancel the deep software standby mode. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

### DEEPCUT[1:0] Bits (Deep Cut)

The DEEPCUT[1:0] bits control the internal power supply to the RAM and USB resume detecting unit in deep software standby mode. In addition, these bits control the state of LVD and power-on reset circuit in deep software standby mode. The internal power supply of RAM0 and USB resume detecting unit can be controlled by the setting of the DEEPCUT[1:0] bits.

When a USB suspend/resume interrupt is used as a deep software standby mode canceling source, the DEEPCUT[1:0] bits must be set to 00b.

When an LVD interrupt is used in deep software standby mode, the DEEPCUT[1:0] bits must be set to 00b or 01b.

For lower power consumption, set the DEEPCUT[1:0] bits to 11b so that the LVD is stopped and the low power consumption function of the power-on reset circuit is enabled.

The internal power supply of RAM1 is stopped in deep software standby mode regardless of the setting of the DEEPCUT[1:0] bits.

### IOKEEP Bit (I/O Port Retention)

In deep software standby mode, I/O ports keep retaining the same states from software standby mode. The IOKEEP bit specifies whether to keep retaining the I/O port states from deep software standby mode even after deep software standby mode is canceled, or to cancel retaining the I/O port states.

**DPSBY Bit (Deep Software Standby)**

The DPSBY bit controls transitions to deep software standby mode.

When the WAIT instruction is executed while the SBYCR.SSBY and DPSBY bits are both 1, the LSI enters deep software standby mode through software standby mode.

The DPSBY bit remains 1 when deep software standby mode is canceled by certain pins which are sources of external pin interrupts (NM1, IRQ0-DS to IRQ15-DS, SCL2-DS, SDA2-DS, and CRXI-DS) or a peripheral interrupt (RTC alarm, RTC interval, USB suspend/resume, voltage monitoring 1, or voltage monitoring 2). Write 0 to this bit to clear it. The setting of the DPSBY bit becomes invalid when the IWDT is in auto-start mode and the OFS0.IWDTSLCSTP is 0 (counting continues) or the IWDT is in register start mode and the SLCSTP bit in IWDTDCSTPR is 0.

Instead, even when the SBYCR.SSBY bit is 1 and the DPSBY bit 1, the transition after the execution of a WAIT instruction is to software standby mode.

The setting of the DPSBY bit becomes invalid when voltage monitoring 1 reset is enabled by the voltage monitoring 1 circuit mode select bit (LVD1CR0.LVD1RI = 1) or when a voltage monitoring 2 reset is selected by the voltage monitoring 2 circuit mode bit (LVD2CR0.LVD2RI = 1). In this case, even when the SBYCR.SSBY bit is 1 and the DPSBY bit is 1, the transition after the execution of a WAIT instruction is to software standby mode.

### 11.2.11 Deep Standby Interrupt Enable Register 0 (DPSIER0)

Address(es): 0008 C282h

b7	b6	b5	b4	b3	b2	b1	b0
DIRQ7 E	DIRQ6 E	DIRQ5 E	DIRQ4 E	DIRQ3 E	DIRQ2 E	DIRQ1 E	DIRQ0 E

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0E	IRQ0-DS Pin Enable	0: Canceling deep software standby mode by the IRQ0-DS pin is disabled 1: Canceling deep software standby mode by the IRQ0-DS pin is enabled	R/W
b1	DIRQ1E	IRQ1-DS Pin Enable	0: Canceling deep software standby mode by the IRQ1-DS pin is disabled 1: Canceling deep software standby mode by the IRQ1-DS pin is enabled	R/W
b2	DIRQ2E	IRQ2-DS Pin Enable	0: Canceling deep software standby mode by the IRQ2-DS pin is disabled 1: Canceling deep software standby mode by the IRQ2-DS pin is enabled	R/W
b3	DIRQ3E	IRQ3-DS Pin Enable	0: Canceling deep software standby mode by the IRQ3-DS pin is disabled 1: Canceling deep software standby mode by the IRQ3-DS pin is enabled	R/W
b4	DIRQ4E	IRQ4-DS Pin Enable	0: Canceling deep software standby mode by the IRQ4-DS pin is disabled 1: Canceling deep software standby mode by the IRQ4-DS pin is enabled	R/W
b5	DIRQ5E	IRQ5-DS Pin Enable	0: Canceling deep software standby mode by the IRQ5-DS pin is disabled 1: Canceling deep software standby mode by the IRQ5-DS pin is enabled	R/W
b6	DIRQ6E	IRQ6-DS Pin Enable	0: Canceling deep software standby mode by the IRQ6-DS pin is disabled 1: Canceling deep software standby mode by the IRQ6-DS pin is enabled	R/W
b7	DIRQ7E	IRQ7-DS Pin Enable	0: Canceling deep software standby mode by the IRQ7-DS pin is disabled 1: Canceling deep software standby mode by the IRQ7-DS pin is enabled	R/W

DPSIER0 is not initialized by the internal reset signal used as deep software standby mode canceling source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

After the setting of DPSIER0 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR0 being set to 1. Therefore, DPSIFR0 should be cleared to 0 before a transition to deep software standby mode.

Even when DPSIER0 is 0, a rising edge may be internally generated at a transition to deep software standby mode with a specific pin state, resulting in DPSIFR0 being set to 1. However, when DPSIEGR0 is 0, the rising edge is not detected, resulting in DPSIFR0 not being set to 1.

## 11.2.12 Deep Standby Interrupt Enable Register 1 (DPSIER1)

Address(es): 0008 C283h

b7	b6	b5	b4	b3	b2	b1	b0
DIRQ15E	DIRQ14E	DIRQ13E	DIRQ12E	DIRQ11E	DIRQ10E	DIRQ9E	DIRQ8E

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ8E	IRQ8-DS Pin Enable	0: Canceling deep software standby mode by the IRQ8-DS pin is disabled 1: Canceling deep software standby mode by the IRQ8-DS pin is enabled	R/W
b1	DIRQ9E	IRQ9-DS Pin Enable	0: Canceling deep software standby mode by the IRQ9-DS pin is disabled 1: Canceling deep software standby mode by the IRQ9-DS pin is enabled	R/W
b2	DIRQ10E	IRQ10-DS Pin Enable	0: Canceling deep software standby mode by the IRQ10-DS pin is disabled 1: Canceling deep software standby mode by the IRQ10-DS pin is enabled	R/W
b3	DIRQ11E	IRQ11-DS Pin Enable	0: Canceling deep software standby mode by the IRQ11-DS pin is disabled 1: Canceling deep software standby mode by the IRQ11-DS pin is enabled	R/W
b4	DIRQ12E	IRQ12-DS Pin Enable	0: Canceling deep software standby mode by the IRQ12-DS pin is disabled 1: Canceling deep software standby mode by the IRQ12-DS pin is enabled	R/W
b5	DIRQ13E	IRQ13-DS Pin Enable	0: Canceling deep software standby mode by the IRQ13-DS pin is disabled 1: Canceling deep software standby mode by the IRQ13-DS pin is enabled	R/W
b6	DIRQ14E	IRQ14-DS Pin Enable	0: Canceling deep software standby mode by the IRQ14-DS pin is disabled 1: Canceling deep software standby mode by the IRQ14-DS pin is enabled	R/W
b7	DIRQ15E	IRQ15-DS Pin Enable	0: Canceling deep software standby mode by the IRQ15-DS pin is disabled 1: Canceling deep software standby mode by the IRQ15-DS pin is enabled	R/W

DPSIER1 is not initialized by the internal reset signal used as deep software standby mode canceling source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

After the setting of DPSIER1 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR1 being set to 1. Therefore, DPSIFR1 should be cleared to 0 before a transition to deep software standby mode.

Even when DPSIER1 is 0, a rising edge may be internally generated at a transition to deep software standby mode with a specific pin state, resulting in DPSIFR1 being set to 1. However, when DPSIEGR1 is 0, the rising edge is not detected, resulting in DPSIFR1 not being set to 1.

### 11.2.13 Deep Standby Interrupt Enable Register 2 (DPSIER2)

Address(es): 0008 C284h

b7	b6	b5	b4	b3	b2	b1	b0
DUSBI E	DRIICC IE	DRIICD IE	DNMIE	DRTCA IE	DRTCII E	DLVD2I E	DLVD1I E
Value after reset: 0 0 0 0 0 0 0 0							

Bit	Symbol	Bit Name	Description	R/W
b0	DLVD1IE	LVD1 Deep Standby Cancel Signal Enable	0: Disable canceling deep software standby mode by the voltage monitoring 1 signal 1: Enable canceling deep software standby mode by the voltage monitoring 1 signal	R/W
b1	DLVD2IE	LVD2 Deep Standby Cancel Signal Enable	0: Disable canceling deep software standby mode by the voltage monitoring 2 signal 1: Enable canceling deep software standby mode by the voltage monitoring 2 signal	R/W
b2	DRTCIE	RTC Interval Interrupt Deep Standby Cancel Signal Enable	0: Canceling deep software standby mode by the RTC interval interrupt signal is disabled 1: Canceling deep software standby mode by the RTC interval interrupt signal is enabled	R/W
b3	DRTCAIE	RTC Alarm Interrupt Deep Standby Cancel Signal Enable	0: Canceling deep software standby mode by the RTC alarm interrupt signal is disabled 1: Canceling deep software standby mode by the RTC alarm interrupt signal is enabled	R/W
b4	DNMIE	NMI Pin Enable	0: Canceling deep software standby mode by the NMI pin is disabled 1: Canceling deep software standby mode by the NMI pin is enabled	R/W*1
b5	DRIICDIE	SDA2-DS Deep Standby Cancel Signal Enable	0: Canceling deep software standby mode by the SDA2-DS signal is disabled 1: Canceling deep software standby mode by the SDA2-DS signal is enabled	R/W
b6	DRIICDIE	SCL2-DS Deep Standby Cancel Signal Enable	0: Canceling deep software standby mode by the SCL2-DS signal is disabled 1: Canceling deep software standby mode by the SCL2-DS signal is enabled	R/W
b7	DUSBIE	USB Suspend/Resume Deep Standby Cancel Signal Enable	0: Canceling deep software standby mode by the USB suspend/resume is disabled 1: Canceling deep software standby mode by the USB suspend/resume is enabled	R/W

Note 1. 1 can be written only once. Once 1 is written to this bit, subsequent write accesses are disabled.

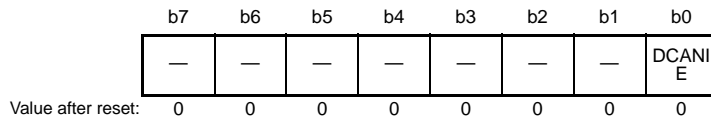
DPSIER2 is not initialized by the internal reset signal used as deep software standby mode canceling source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

After the setting of DPSIER2 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR2 being set to 1. Therefore, DPSIFR2 should be cleared to 0 before a transition to deep software standby mode.

Even when DPSIER2 is 0, a rising edge may be internally generated at a transition to deep software standby mode with a specific pin state, resulting in DPSIFR2 being set to 1. However, when DPSIEGR2 is 0, the rising edge is not detected, resulting in DPSIFR2 not being set to 1.

### 11.2.14 Deep Standby Interrupt Enable Register 3 (DPSIER3)

Address(es): 0008 C285h



Bit	Symbol	Bit Name	Description	R/W
b0	DCANIE	CRX1-DS Deep Standby Cancel Signal Enable	0: Canceling deep software standby mode by the CRX1-DS pin is disabled 1: Canceling deep software standby mode by the CRX1-DS pin is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DPSIER3 is not initialized by the internal reset signal used as deep software standby mode canceling source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

After the setting of DPSIER3 is modified, an edge may be internally generated depending on the state of the pin, resulting in DPSIFR3 being set to 1. Therefore, DPSIFR3 should be cleared to 0 before a transition to deep software standby mode.

Even when DPSIER3 is 0, a rising edge may be internally generated at a transition to deep software standby mode with a specific pin state, resulting in DPSIFR3 being set to 1. However, when DPSIEGR3 is 0, the rising edge is not detected, resulting in DPSIFR3 not being set to 1 because the rising edge is not detected.

## 11.2.15 Deep Standby Interrupt Flag Register 0 (DPSIFR0)

Address(es): 0008 C286h

b7	b6	b5	b4	b3	b2	b1	b0
DIRQ7 F	DIRQ6 F	DIRQ5 F	DIRQ4 F	DIRQ3 F	DIRQ2 F	DIRQ1 F	DIRQ0 F

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0F	IRQ0-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ0-DS pin is generated 1: A cancel request by the IRQ0-DS pin is generated	R(W) *1
b1	DIRQ1F	IRQ1-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ1-DS pin is generated 1: A cancel request by the IRQ1-DS pin is generated	R(W) *1
b2	DIRQ2F	IRQ2-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ2-DS pin is generated 1: A cancel request by the IRQ2-DS pin is generated	R(W) *1
b3	DIRQ3F	IRQ3-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ3-DS pin is generated 1: A cancel request by the IRQ3-DS pin is generated	R(W) *1
b4	DIRQ4F	IRQ4-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ4-DS pin is generated 1: A cancel request by the IRQ4-DS pin is generated	R(W) *1
b5	DIRQ5F	IRQ5-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ5-DS pin is generated 1: A cancel request by the IRQ5-DS pin is generated	R(W) *1
b6	DIRQ6F	IRQ6-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ6-DS pin is generated 1: A cancel request by the IRQ6-DS pin is generated	R(W) *1
b7	DIRQ7F	IRQ7-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ7-DS pin is generated 1: A cancel request by the IRQ7-DS pin is generated	R(W) *1

Note 1. Only 0 can be written to clear the flag.

Each flag is set to 1 when a cancel request specified by DPSIEGR0 is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not even deep software standby mode) or when the setting of DPSIER0 is modified. Therefore, a transition to deep software standby mode should be made after DPSIFR0 is cleared to 00h.

To clear DPSIFR0 to 00h after modifying DPSIER0, wait for at least six PCLKB cycles, read DPSIFR0, and then write 0 to DPSIFR0. Six or more PCLKB cycles can be secured, for example, by reading DPSIER0.

DPSIFR0 is not initialized by the internal reset signal used as deep software standby mode canceling source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

### DIRQnF Flags (IRQn Deep Standby Cancel Flag) (n = 0 to 7)

These flags indicate that a cancel request by the IRQn-DS pin has been generated.

[Setting condition]

- A cancel request by the IRQn-DS pin specified by DPSIEGR0 is generated

[Clearing condition]

- Each bit is read as 1 and then written by 0

## 11.2.16 Deep Standby Interrupt Flag Register 1 (DPSIFR1)

Address(es): 0008 C287h

	b7	b6	b5	b4	b3	b2	b1	b0
	DIRQ1 5F	DIRQ1 4F	DIRQ1 3F	DIRQ1 2F	DIRQ11 F	DIRQ1 0F	DIRQ9 F	DIRQ8 F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ8F	IRQ8-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ8-DS pin is generated 1: A cancel request by the IRQ8-DS pin is generated	R(W) *1
b1	DIRQ9F	IRQ9-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ9-DS pin is generated 1: A cancel request by the IRQ9-DS pin is generated	R(W) *1
b2	DIRQ10F	IRQ10-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ10-DS pin is generated 1: A cancel request by the IRQ10-DS pin is generated	R(W) *1
b3	DIRQ11F	IRQ11-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ11-DS pin is generated 1: A cancel request by the IRQ11-DS pin is generated	R(W) *1
b4	DIRQ12F	IRQ12-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ12-DS pin is generated 1: A cancel request by the IRQ12-DS pin is generated	R(W) *1
b5	DIRQ13F	IRQ13-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ13-DS pin is generated 1: A cancel request by the IRQ13-DS pin is generated	R(W) *1
b6	DIRQ14F	IRQ14-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ14-DS pin is generated 1: A cancel request by the IRQ14-DS pin is generated	R(W) *1
b7	DIRQ15F	IRQ15-DS Deep Standby Cancel Flag	0: No cancel request by the IRQ15-DS pin is generated 1: A cancel request by the IRQ15-DS pin is generated	R(W) *1

Note 1. Only 0 can be written to clear the flag.

Each flag is set to 1 when a cancel request specified by DPSIEGR1 is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not even deep software standby mode) or when the setting of DPSIER1 is modified. Therefore, a transition to deep software standby mode should be made after DPSIFR1 is cleared to 00h.

To clear DPSIFR1 to 00h after modifying DPSIER1, wait for at least six PCLKB cycles, read DPSIFR1, and then write 0 to DPSIFR1. Six or more PCLKB cycles can be secured, for example, by reading DPSIER1.

DPSIFR1 is not initialized by the internal reset signal used as deep software standby mode canceling source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

### DIRQnF Flags (IRQn Deep Standby Cancel Flag) (n = 8 to 15)

These flags indicate that a cancel request by the IRQn-DS pin has been generated.

[Setting condition]

- A cancel request by the IRQn-DS pin specified by DPSIEGR1 is generated

[Clearing condition]

- Each bit is read as 1 and then written by 0



## 11.2.17 Deep Standby Interrupt Flag Register 2 (DPSIFR2)

Address(es): 0008 C288h

	b7	b6	b5	b4	b3	b2	b1	b0
	DUSBI F	DRIICC IF	DRIICD IF	DNMIF	DRTCA IF	DRTCII F	DLVD2I F	DLVD1I F
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DLVD1IF	LVD1 Deep Standby Cancel Flag	0: No cancel request by the voltage monitor 1 signal is generated 1: A cancel request by the voltage monitor 1 signal is generated	R/(W) *1
b1	DLVD2IF	LVD2 Deep Standby Cancel Flag	0: No cancel request by the voltage monitor 2 signal is generated 1: A cancel request by the voltage monitor 2 signal is generated	R/(W) *1
b2	DRTCIIIF	RTC Interval Interrupt Deep Standby Cancel Flag	0: No cancel request by the RTC interval interrupt signal is generated 1: A cancel request by the RTC interval interrupt signal is generated	R/(W) *1
b3	DRTCAIF	RTC Alarm Interrupt Deep Standby Cancel Flag	0: No cancel request by the RTC alarm interrupt signal is generated 1: A cancel request by the RTC alarm interrupt signal is generated	R/(W) *1
b4	DNMIF	NMI Deep Standby Cancel Flag	0: No cancel request by the NMI pin is generated 1: A cancel request by the NMI pin is generated	R/(W) *1
b5	DRIICDIF	SDA2-DS Deep Standby Cancel Flag	0: No cancel request by the SDA2-DS signal is generated 1: A cancel request by the SDA2-DS signal is generated	R/(W) *1
b6	DRIICCIF	SCL2-DS Deep Standby Cancel Flag	0: No cancel request by the SCL2-DS signal is generated 1: A cancel request by the SCL2-DS signal is generated	R/(W) *1
b7	DUSBIF	USB Suspend/Resume Deep Standby Cancel Flag	0: No cancel request by the USB suspend/resume is generated 1: A cancel request by the USB suspend/resume is generated	R/(W) *1

Note 1. Only 0 can be written to clear the flag.

Each flag is set to 1 when a cancel request specified by DPSIEGR2 is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not even deep software standby mode) or when the setting of DPSIER2 is modified. Therefore, a transition to deep software standby mode should be made after DPSIFR2 is cleared to 00h.

To clear DPSIFR2 to 00h after modifying DPSIER2, wait for at least six PCLKB cycles, read DPSIFR2, and then write 0 to DPSIFR2. Six or more PCLKB cycles can be secured, for example, by reading DPSIER2.

DPSIFR2 is not initialized by the internal reset signal used as deep software standby mode canceling source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

### DLVDmIF Flag (LVDm Deep Standby Cancel Flag) (m = 1 or 2)

This flag indicates that a cancel request by the voltage monitor m signal has been generated.

[Setting condition]

- A cancel request is generated by the voltage monitoring m signal that is selected in DPSIEGR2

[Clearing condition]

- This bit is read as 1 and then written by 0

### DRTCIIIF Flag (RTC Interval Interrupt Deep Standby Cancel Flag)

This flag indicates that a cancel request by the RTC interval interrupt signal has been generated.

[Setting condition]

- A cancel request by the RTC interval interrupt signal is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

**DRTCAIF Flag (RTC Alarm Interrupt Deep Standby Cancel Flag)**

This flag indicates that a cancel request by the RTC alarm interrupt signal has been generated.

[Setting condition]

- A cancel request by the RTC alarm interrupt signal is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

**DNMIF Flag (NMI Deep Standby Cancel Flag)**

This flag indicates that a cancel request by the NMI pin has been generated.

[Setting condition]

- A cancel request by the NMI pin specified by DPSIEGR2 is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

**DRIICDIF Flag (SDA2-DS Deep Standby Cancel Flag)**

This flag indicates that a cancel request by the SDA2-DS signal has been generated.

[Setting condition]

- A cancel request by the SDA2-DS pin specified by DPSIEGR2 is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

**DRIICCIF Flag (SCL2-DS Deep Standby Cancel Flag)**

This flag indicates that a cancel request by the SCL2-DS signal has been generated.

[Setting condition]

- A cancel request by the SCL2-DS pin specified by DPSIEGR2 is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

**DUSBIF Flag (USB Suspend/Resume Deep Standby Cancel Flag)**

This flag indicates that a cancel request by the USB suspend/resume has been generated.

[Setting condition]

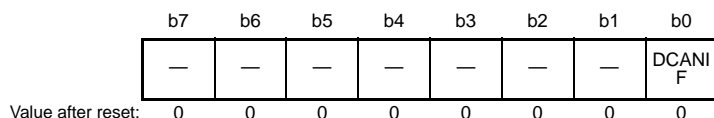
- A cancel request by the USB suspend/resume is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

### 11.2.18 Deep Standby Interrupt Flag Register 3 (DPSIFR3)

Address(es): 0008 C289h



Bit	Symbol	Bit Name	Description	R/W
b0	DCANIF	CRX1-DS Deep Standby Cancel Flag	0: No cancel request by the CRX1-DS pin is generated 1: A cancel request by the CRX1-DS pin 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. Only 0 can be written to clear the flag.

Each flag is set to 1 when a cancel request specified by DPSIEGR3 is generated.

Each flag may be set to 1 when a cancel request is generated in any mode (not even deep software standby mode) or when the setting of DPSIER3 is modified. Therefore, a transition to deep software standby mode should be made after DPSIFR3 is cleared to 00h.

To clear DPSIFR3 to 00h after modifying DPSIER3, wait for at least six PCLKB cycles, read DPSIFR3, and then write 0 to DPSIFR3. Six or more PCLKB cycles can be secured, for example, by reading DPSIER3.

DPSIFR3 is not initialized by the internal reset signal used as deep software standby mode canceling source. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

#### DCANIF Flag (CAN Deep Standby Cancel Flag)

This flag indicates that a cancel request by the CRXD-DS pin has been generated.

[Setting condition]

- A cancel request by the CRX1-DS pin specified by DPSIEGR3 is generated

[Clearing condition]

- This bit is read as 1 and then written by 0

### 11.2.19 Deep Standby Interrupt Edge Register 0 (DPSIEGR0)

Address(es): 0008 C28Ah

b7	b6	b5	b4	b3	b2	b1	b0
DIRQ7 EG	DIRQ6 EG	DIRQ5 EG	DIRQ4 EG	DIRQ3 EG	DIRQ2 EG	DIRQ1 EG	DIRQ0 EG

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ0EG	IRQ0-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b1	DIRQ1EG	IRQ1-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b2	DIRQ2EG	IRQ2-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b3	DIRQ3EG	IRQ3-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b4	DIRQ4EG	IRQ4-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b5	DIRQ5EG	IRQ5-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b6	DIRQ6EG	IRQ6-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b7	DIRQ7EG	IRQ7-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W

DPSIEGR0 is not initialized by the internal reset signal that is the source to cancel the deep software standby mode. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

## 11.2.20 Deep Standby Interrupt Edge Register 1 (DPSIEGR1)

Address(es): 0008 C28Bh

b7	b6	b5	b4	b3	b2	b1	b0
DIRQ15EG	DIRQ14EG	DIRQ13EG	DIRQ12EG	DIRQ11EG	DIRQ10EG	DIRQ9EG	DIRQ8EG

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	DIRQ8EG	IRQ8-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b1	DIRQ9EG	IRQ9-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b2	DIRQ10EG	IRQ10-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b3	DIRQ11EG	IRQ11-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b4	DIRQ12EG	IRQ12-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b5	DIRQ13EG	IRQ13-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b6	DIRQ14EG	IRQ14-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b7	DIRQ15EG	IRQ15-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W

DPSIEGR1 is not initialized by the internal reset signal that is the source to cancel the deep software standby mode. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

### 11.2.21 Deep Standby Interrupt Edge Register 2 (DPSIEGR2)

Address(es): 0008 C28Ch

b7	b6	b5	b4	b3	b2	b1	b0
—	DRIICC EG	DRIICD EG	DNMIE G	—	—	DLVD2 EG	DLVD1 EG

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	DLVD1EG	LVD1 Edge Select	0: A cancel request is generated when VCC < Vdet1 (fall) is detected 1: A cancel request is generated when VCC ≥ Vdet1 (rise) is detected	R/W
b1	DLVD2EG	LVD2 Edge Select	0: A cancel request is generated when VCC < Vdet2 (fall) is detected 1: A cancel request is generated when VCC ≥ Vdet2 (rise) is detected	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DNMIEG	NMI Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b5	DRIICDEG	SDA2-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b6	DRIICCEG	SCL2-DS Edge Select	0: A cancel request is generated at a falling edge 1: A cancel request is generated at a rising edge	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

DPSIEGR2 is not initialized by the internal reset signal that is the source to cancel the deep software standby mode. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

### 11.2.22 Deep Standby Interrupt Edge Register 3 (DPSIEGR3)

Address(es): 0008 C28Dh

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	DCANI EG

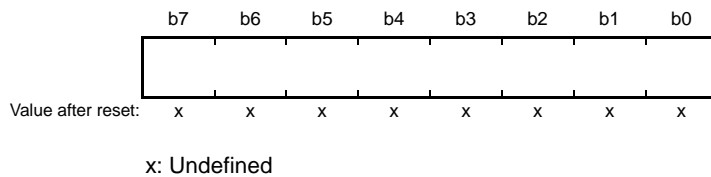
Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	DCANIEG	CRX1-DS Edge Select	0: A cancellation request is generated at a falling edge. 1: A cancellation request is generated at a rising edge.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DPSIEGR3 is not initialized by the internal reset signal that is the source to cancel the deep software standby mode. For details, see Table 6.2, Targets to be Initialized by Each Reset Source.

### 11.2.23 Deep Standby Backup Register (DPSBKRY) (y = 0 to 31)

Address(es): 0008 C2A0h to 0008 C2BFh



DPSBKRY is a 32-byte readable/writable register to store data during deep software standby mode. The value of this register is retained even in deep software standby mode where RAM data is not retained. DPSBKRY is not initialized, and the register value is undefined immediately after power-on.

## 11.3 Reducing Power Consumption by Switching Clock Signals

When the SCKCR.FCK[3:0], ICK[3:0], BCK[3:0], and PCKB[3:0] bits are set, the clock frequency changes. The CPU, DMAC, DTC, ROM, and RAM operate on the operating clock specified by the ICK[3:0] bits.

Peripheral modules operate on the operating clock specified by the PCKB[3:0] bit.

The flash memory interface operates on the operating clock specified by the FCK[3:0] bits.

The external bus operates on the operating clock specified by the BCK[3:0] bits. For details, see 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 = 31 to 0) 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. Clearing the MSTPmi bit to 0 cancels the module-stop state, allowing the module to restart 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 DMAC, DTC, and RAM are placed in the module-stop state. Though read/write access cannot be made to the registers of the module that are in the module-stop state, some registers may be written to directly after the setting to the module-stop state. Therefore, care should be paid.

## 11.5 Function for Lower Operating Power Consumption

By selecting an appropriate operating power consumption control mode according to the operating frequency and operating voltage, power consumption can be reduced in normal operation, sleep mode and all-module clock stop mode.

### 11.5.1 Setting Operating Power Consumption Control Mode

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

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

Example: From high-speed operation mode to low-speed operation mode 1

(High-speed operation in the operating power consumption control mode used before mode-switching)

↓

Set to switch from the HOCO clock to the LOCO clock (clock source and frequency division ratio)

↓

Write to OPCCR

↓

Confirm that the OPCCR.OPCMTSF flag is 0

↓

(Low-speed operation in the switched operating power consumption control mode)

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

Example: From low-speed operation mode 2 to high-speed operation mode

(Low-speed operation in the operating power consumption control mode used before mode-switching)

↓

Write to OPCCR

↓

Confirm that the OPCCR.OPCMTSF flag is 0

↓

Set to switch from the LOCO clock to the HOCO clock (clock source and frequency division ratio)

↓

(High-speed operation in the switched operating power consumption control mode)



## 11.6 Low Power Consumption Modes

### 11.6.1 Sleep Mode

#### 11.6.1.1 Transition 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.

When the WDT is used, the WDT stops counting when sleep mode is entered.

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 SLCSTP bit in IWDTCSSTPR 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 SLCSTP bit in IWDTCSSTPR is 0.

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

- (1) Clear the PSW.I bit\*<sup>1</sup> of the CPU to 0.
- (2) Set the interrupt destination to be used for recovery from sleep mode to the CPU.
- (3) Set the priority\*<sup>2</sup> of the interrupt to be used for recovery from sleep mode to a level higher than the setting of the PSW.IPL[3:0] bits\*<sup>1</sup> of the CPU.
- (4) Set the IERm.IENj bit\*<sup>2</sup> for the interrupt to 1.
- (5) For the last I/O register to which writing proceeded, read the register to confirm that the value written has been reflected.
- (6) Execute the WAIT instruction (this automatically sets the I bit\*<sup>1</sup> in the PSW of the CPU to 1).

Note 1. For details, see section 2, CPU.

Note 2. For details, see section 15, Interrupt controller (ICUb).

#### 11.6.1.2 Canceling Sleep Mode

Sleep mode is canceled by any interrupt, the reset signal from the RES# pin reset, a power-on reset, a voltage monitoring reset, or a reset caused by an IWDT underflow.

- Canceling by an interrupt  
When an interrupt occurs, sleep mode is canceled and the interrupt exception handling starts. If a maskable interrupt has been masked by the CPU (the priority level\*<sup>1</sup> of the interrupt has been set to a value lower than that of the PSW.IPL[3:0] bits\*<sup>2</sup> of the CPU), sleep mode is not canceled.
- Canceling by the RES# pin reset  
When the RES# pin is driven low, the LSI 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.
- Canceling by a power-on reset  
Sleep mode is canceled by a power-on reset.
- Canceling by a voltage monitoring reset  
Sleep mode is canceled by a voltage monitoring reset from the voltage detection circuit.
- Canceling by the independent watchdog timer reset  
Sleep mode is canceled by an internal reset generated by an IWDT underflow. However, when such conditions are set that stop IWDT counting in sleep mode (OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSSTPR.SLCSTP = 1), the IWDT is stopped and sleep mode cannot be canceled by the independent watchdog timer reset.

Note 1. For details, see section 15, Interrupt controller (ICUb).

Note 2. For details, see section 2, CPU.

### 11.6.1.3 Sleep Mode Return Clock Source Switching Function

To switch the clock source used on return from sleep mode, the clock used after return needs to be set by the sleep mode return clock source switching register (RSTCKCR) and the wait control register needs to be set for each clock source. When the return interrupt is generated, after oscillation settling of the oscillator specified as the return clock, the clock source is automatically switched, and then operation returns from sleep mode. At this time, the registers related to clock source switching are automatically rewritten.

For details, see section 11.2.6, Sleep Mode Return Clock Source Switching Register (RSTCKCR). In addition, for details on the oscillation settling time, see section 11.2.7, Main Clock Oscillator Wait Control Register (MOSCWTCR).

## 11.6.2 All-Module Clock Stop Mode

### 11.6.2.1 Transition to All-Module Clock Stop Mode

After setting the MSTPCRA.ACSE bit to 1 and placing modules controlled by MSTPCRA, MSTPCRB, and MSTPCRC registers in the module-stop state (MSTPCRA = FFFF FF[C-F]Fh, MSTPCRB = FFFF FFFFh, MSTPCRC[31:16] = FFFFh), executing a WAIT instruction while the SBYCR.SSBY bit is 0 stops the bus controller, I/O ports, and all modules except for the 8-bit timers\*1, POE\*5, IWDT, RTC, power-on reset circuit, voltage detection circuit at the end of the current bus cycle, and the chip enters all-module clock stop mode\*2.

When the WDT is used, the WDT stops counting when all-module clock stop mode is entered.

Counting by the IWDT stops if a transition to all-module clock-stop 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 all-module clock-stop mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 1.

Furthermore, counting by the IWDT continues if a transition to all-module clock-stop 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 all-module clock-stop mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 0.

To use all-module clock-stop mode, make the following settings and then execute a WAIT instruction.

- (1) Clear the PSW.I bit\*3 of the CPU to 0.
- (2) Set the interrupt destination to be used for recovery from all-module clock stop mode to the CPU.
- (3) Set the priority\*4 of the interrupt to be used for recovery from all-module clock stop mode to a level higher than the setting of the PSW.IPL[3:0] bits\*3 of the CPU.
- (4) Set the IERm.IENj bit\*4 for the interrupt to be used for recovery from all-module clock stop mode to 1.
- (5) Read the last I/O register to have been written and confirm that its value reflects the value written.
- (6) Execute a WAIT instruction (executing a WAIT instruction causes automatic setting of the PSW.I bit\*3 of the CPU to 1).

Note 1. The MSTPCRA.MSTPA4 and MSTPA5 bits select operation or stop of these modules.

Note 2. Transitions to all-module clock stop mode are not to be made in some states of DTC or DMAC operations. Before setting the MSTPCRA.MSTPA28 bit to 1, clear the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC so that the DTC and DMACA are not activated.

Note 3. For details, see section 2, CPU.

Note 4. For details, see section 15, Interrupt controller (ICUb).

Note 5. When a POE interrupt source condition is satisfied while the setting to enable POE interrupts is in place, recovery from all-module clock stop mode does not proceed but the flag to indicate satisfaction of the source condition is retained. If a different source leads to recovery from all-module clock stop mode in this situation, a POE interrupt is generated after recovery.

### 11.6.2.2 Canceling All-Module Clock Stop Mode

Release from all-module clock-stop mode is initiated by an external pin interrupt (the NMI or IRQ0 to IRQ15), a peripheral interrupt (8-bit timer\*1, RTC alarm, RTC interval, IWDT\*2, USB suspend/resume, voltage monitoring 1, voltage monitoring 2, or oscillator-stopped detection interrupt), a RES# pin reset, a power-on reset, a voltage monitoring reset, or an independent watchdog timer reset, and the transition to the normal program execution state proceeds via exception processing for the given interrupt or reset.

However, note that in cases where a maskable interrupt has been masked by the CPU (priority level\*3 of the interrupt has been set to a value lower than that of the PSW.IPL[3:0] bits\*4 of the CPU) or a maskable interrupt has been set up as a trigger to activate the DTC or DMAC, the interrupt will not cancel all-module clock stop mode.

Note 1. The MSTPA4 and MSTPA5 bits of MSTPCRA select operation or stopping of these modules.

Note 2. If a condition for the independent watchdog timer to stop counting applied (OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1) at the time of a transition to all-module clock-stop mode, using a reset from the independent watchdog timer to release the chip from all-module clock-stop mode is impossible because the independent watchdog timer is stopped.

Note 3. For details, see section 15, Interrupt controller (ICUb).

Note 4. For details, see section 2, CPU.

### 11.6.3 Software Standby Mode

#### 11.6.3.1 Transition to Software Standby Mode

When a WAIT instruction is executed with the SBYCR.SSBY bit set to 1 and the DPSBYCR.DPSBY bit cleared to 0, a transition to software standby mode is made. In this mode, the CPU, on-chip peripheral functions, and all the oscillator functions stop. However, the contents of the CPU internal registers, RAM data, on-chip peripheral functions, and the states of the I/O ports are retained. Whether the address bus and bus control signals are placed in the high-impedance or the output state is retained can be specified by the SBYCR.OPE bit. Software standby mode allows significant reduction in power consumption because the oscillator stops in this mode.

Clear the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC to 0 before executing the WAIT instruction.

When the WDT is used, the WDT stops counting when software standby mode is entered because the oscillator stops. 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 SLCSTP bit in IWDTCSTPR 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 SLCSTP bit in IWDTCSTPR is 0.

When the oscillation stop detection function is enabled (OSTDCR.OSTDE = 1), software standby mode cannot be entered. To make a transition to software standby mode, execute a WAIT instruction after disabling the oscillation stop detection function (OSTDCR.OSTDE = 0).

To use software standby mode, make the following settings and then execute a WAIT instruction.

- (1) Clear the PSW.I bit\*1 of the CPU to 0.
- (2) Set the interrupt destination to be used for recovery from software standby mode to the CPU.
- (3) Set the priority\*2 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\*2 for the interrupt to be used for recovery from software standby mode to 1.
- (5) For the last I/O register to which writing proceeded, read the register to confirm that the value written 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, see section 2, CPU.

Note 2. For details, see section 15, Interrupt controller (ICUb).

### 11.6.3.2 Canceling Software Standby Mode

Release from software standby mode is initiated by an external pin interrupt (the NMI or IRQ0 to IRQ15), peripheral interrupts (the RTC alarm, RTC interval, IWDT, USB suspend/resume, voltage monitoring 1, and voltage monitoring 2 interrupts), an RES# pin reset, a power-on reset, a voltage monitoring reset, or an independent watchdog timer reset. When an interrupt initiates release from software standby, the oscillators which were operating before the transition to software standby are restarted. After the oscillation of all these oscillators has been stabilized, operation returns from software standby mode.

- Release due to an interrupt

When an interrupt request from the NMI, IRQ0 to IRQ15, RTC alarm, RTC cycle, IWDT, USB suspend/resume, voltage monitoring 1, or voltage monitoring 2 interrupt is generated, each oscillator which was operating before a transition to software standby mode resumes oscillation. After the time set by the MOSCWTCR.MSTS[4:0], SOSCWTCR.SSTS[4:0], or PLLWTCR.PSTS[4:0] bits has elapsed, the chip is released from software standby and starts interrupt exception processing.

- Release due to a reset on the RES# pin reset

Clock oscillation starts when the low level is applied to the RES# pin. Clock supply for the LSI 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.

- Release due to a power-on reset

Release from software standby mode proceeds after a fall in the power-supply voltage leads to the generation of a power-on reset.

- Release due to a voltage monitoring reset

Release from software standby mode proceeds after a fall in the power-supply voltage leads to the generation of a voltage monitoring reset.

- Release due to an independent watchdog timer reset

An internal reset due to an underflow of the IWDT leads to release from software standby mode.

However, if a condition for the independent watchdog timer to stop counting applied (OFS0.IWDTSTRT = 0 and OFS0.IWDTSLCSTP = 1, or OFS0.IWDTSTRT = 1 and IWDTCSTPR.SLCSTP = 1) at the time of a transition to software standby, using a reset from the independent watchdog timer to release the chip from software standby is impossible because the independent watchdog timer is stopped.

### 11.6.3.3 Example of Software Standby Mode Application

Figure 11.2 shows an example where a transition to software standby mode is made at the falling edge of the IRQn pin, and software standby mode is canceled at 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 a transition to software standby mode is made. After that, software standby mode is canceled at the rising edge of the IRQn pin.

To return from software standby mode, settings of the interrupt controller (ICU) are also necessary. For details, see section 15, Interrupt controller (ICUb).

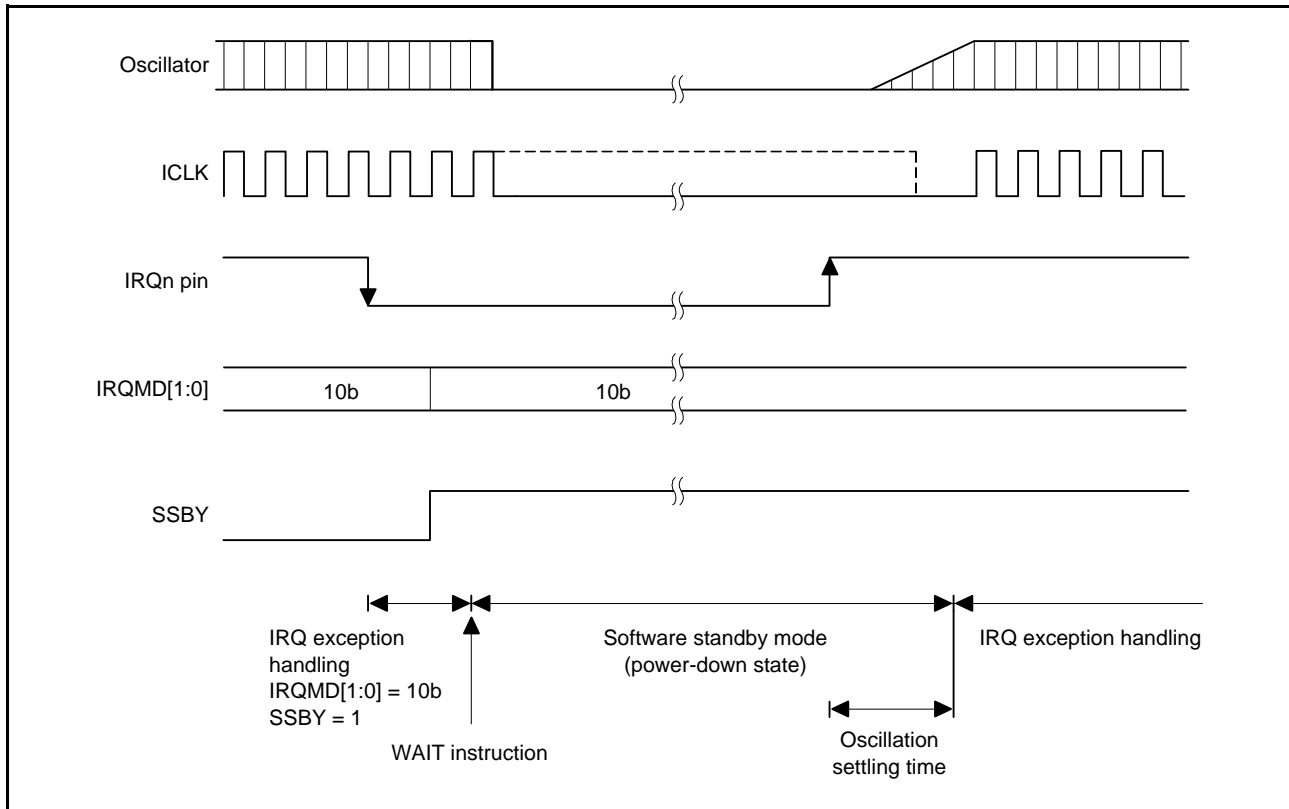


Figure 11.2 Example of Software Standby Mode Application

## 11.6.4 Deep Software Standby Mode

### 11.6.4.1 Transition to Deep Software Standby Mode

When the WAIT instruction is executed with the SBYCR.SSBY bit set to 1, a transition to software standby mode\*1 is made. At this time, when the DPSBYCR.DPSBY bit is set to 1, a transition to deep software standby mode is made. On deep software standby mode, the CPU, internal peripheral modules (except for parts of the RTC alarm, RTC interval, SCL2-DS, SDA2-DS, CRX1-DS, and USB suspend/resume detecting unit), RAM1\*2, and all functions of the oscillators are stopped; furthermore, since the internal supply of power for these modules is stopped, power consumption is markedly reduced. At this time, the contents of all the registers of the CPU and internal peripheral modules (except for parts of the RTC alarm, RTC interval, SCL2-DS, SDA2-DS, CRX1-DS, and USB suspend/resume detecting unit) become undefined. All data in the RAM1\*2 become undefined, regardless of the setting of the DPSBYCR.DEEPCUT[1:0] bits.

Data in the RAM0\*2 are preserved if the setting of the DEEPCUT[1:0] bits is 00b. If the setting of the DEEPCUT[1:0] bits is 01b, the internal supply of power to the RAM0\*2 and the section for detecting USB resumption is cut off, reducing power consumption. Data in the RAM0\*2 become undefined at this time. If the setting of the DEEPCUT[1:0] bits is 11b, the internal supply of power to the RAM0\*2, and the USB resume detecting unit is cut off, the LVD is stopped, and the low-power-consumption function of the power-on reset circuit is enabled, so power consumption is further reduced. For details, see section 45, Electrical Characteristics.

When the WDT is in use, since the oscillators and power supply to the WDT are stopped by the transition to deep software standby mode, counting also stops.

Power supply to the IWDT-dedicated clock and the IWDT is stopped and counting by the IWDT stops if a transition to deep 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, power supply to the IWDT-dedicated clock and the IWDT is stopped and counting by the IWDT stops if a transition to deep software standby mode is made while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 1.

Furthermore, counting by the IWDT continues if a transition to software standby mode is made but not to deep software standby mode 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 but not to deep software standby mode while the IWDT is being used in register start mode and the SLCSTP bit in IWDTCSTPR is 0.

When the voltage monitoring 1 reset function (LVD1CR0.LVD1RI = 1) or voltage monitoring 2 reset function (LVD2CR0.LVD2RI = 1) is selected for the voltage detection circuit, a transition to deep software standby mode cannot be made, but to software standby.

The I/O port states remain unchanged from software standby mode.

Note 1. Conditions on the DTC, DMAC, and IWDT for transition to software standby mode should be met before the WAIT instruction is executed. For details, see section 11.6.3, Software Standby Mode.

Note 2. The RAM address space is divided into the RAM0 area and RAM1 area. For the RAM address space, see Table 11.2.

### 11.6.4.2 Canceling Deep Software Standby Mode

Release from deep software standby mode is initiated by any of the external pin interrupt source pins (the NMI, IRQ0-DS to IRQ15-DS, SCL2-DS, SDA2-DS, or CRX1-DS), peripheral interrupts (the RTC alarm, RTC interval, USB suspend/resume, voltage monitoring 1, and voltage monitoring 2 interrupts), an RES# pin reset, a power-on reset, or a voltage monitoring 0 reset.

(1) Canceling by an external interrupt pin or internal interrupt signal

Cancellation of deep software standby mode is controlled by DPSIER<sub>n</sub> (n = 0 to 3) and DPSIFR<sub>n</sub> (n = 0 to 3). When a deep software standby canceling interrupt is generated, the corresponding flag in DPSIFR<sub>n</sub> is set to 1. At this time, if the canceling source is enabled in DPSIER<sub>n</sub>, deep software standby mode is canceled. Rising edge or falling edge can be selected by DPSIEGR<sub>n</sub> (n = 0 to 3). The interrupts for which an edge can be selected are the NMI, IRQ0-DS to IRQ15-DS, SCL2-DS, SDA2-DS, CRX1-DS, voltage monitoring 1, and voltage monitoring 2 interrupts.

When a deep software standby mode canceling source is generated, the internal power supply and LOCO clock oscillation begin, and then the internal reset (deep software standby reset) is generated for the entire LSI.

A stable LOCO clock is then supplied to the entire LSI and deep software standby reset is canceled. At the same time, deep software standby mode is canceled and the reset exception handling starts.

When deep software standby mode is canceled by an external interrupt pin or internal interrupt signal, the RSTSR0.DPSRSTF flag is set to 1.

(2) Release due to a reset on the RES# pin

Deep software standby mode is canceled when the low level is applied to the RES# pin.

At this time, the RES# pin should be held low according to the specifications described in section 45, Electrical Characteristics. Reset exception processing starts when the high level is applied to the RES# pin.

(3) Release due to a power-on reset

Release from deep software standby mode proceeds after a fall in the power-supply voltage leads to the generation of a power-on reset.

(4) Release due to a voltage monitoring 0 reset

Release from deep software standby mode proceeds after a fall in the power-supply voltage leads to the generation of a voltage monitoring 0 reset.

### 11.6.4.3 Pin States when Deep Software Standby Mode is Canceled

In deep software standby mode, the I/O ports retain the same states from software standby mode. The inside of the LSI is initialized by an internal reset generated when deep software standby mode is canceled. Upon cancellation of deep software standby mode, the reset exception handling starts immediately. The following shows the states of I/O ports at this time.

Whether to initialize the I/O ports or to retain the I/O port states at the time of software standby mode can be selected by the DPSBYCR.IOKEEP bit.

- When the DPSBYCR.IOKEEP bit = 0

I/O ports are initialized by an internal reset generated when deep software standby mode is canceled.

- When the DPSBYCR.IOKEEP bit = 1

Although the inside of the LSI is initialized by an internal reset generated when deep software standby mode is canceled, I/O ports retain their states from software standby mode regardless of the LSI internal state. At this time, the I/O port states remain unchanged from software standby mode even if settings of I/O ports or peripheral modules are made. Then, the retained I/O port states are released by clearing the DPSBYCR.IOKEEP bit to 0, and the LSI operates according to the internal state.

The DPSBYCR.IOKEEP bit is not initialized by an internal reset generated when deep software standby mode is canceled.

### 11.6.4.4 Example of Deep Software Standby Mode Application

Figure 11.3 shows an example where a transition to deep software standby mode is made at the falling edge of the IRQn-DS pin, and deep software standby mode is canceled at the rising edge of the IRQn-DS pin.

In this example, an IRQn interrupt is accepted with the IRQCRi.IRQMD[1:0] bits of the ICU set to 01b (falling edge). Then, after the DPSIEGRy.DIRQnEG (y = 0 or 1, n = 0 to 15) bit is set to 1 (rising edge) and the SBYCR.SSBY bit and DPSBYCR.DPSBY bit are both set to 1, the WAIT instruction is executed. Thus a transition to deep software standby mode is made.

After that, deep software standby mode is canceled at the rising edge of the IRQ-DS pin.

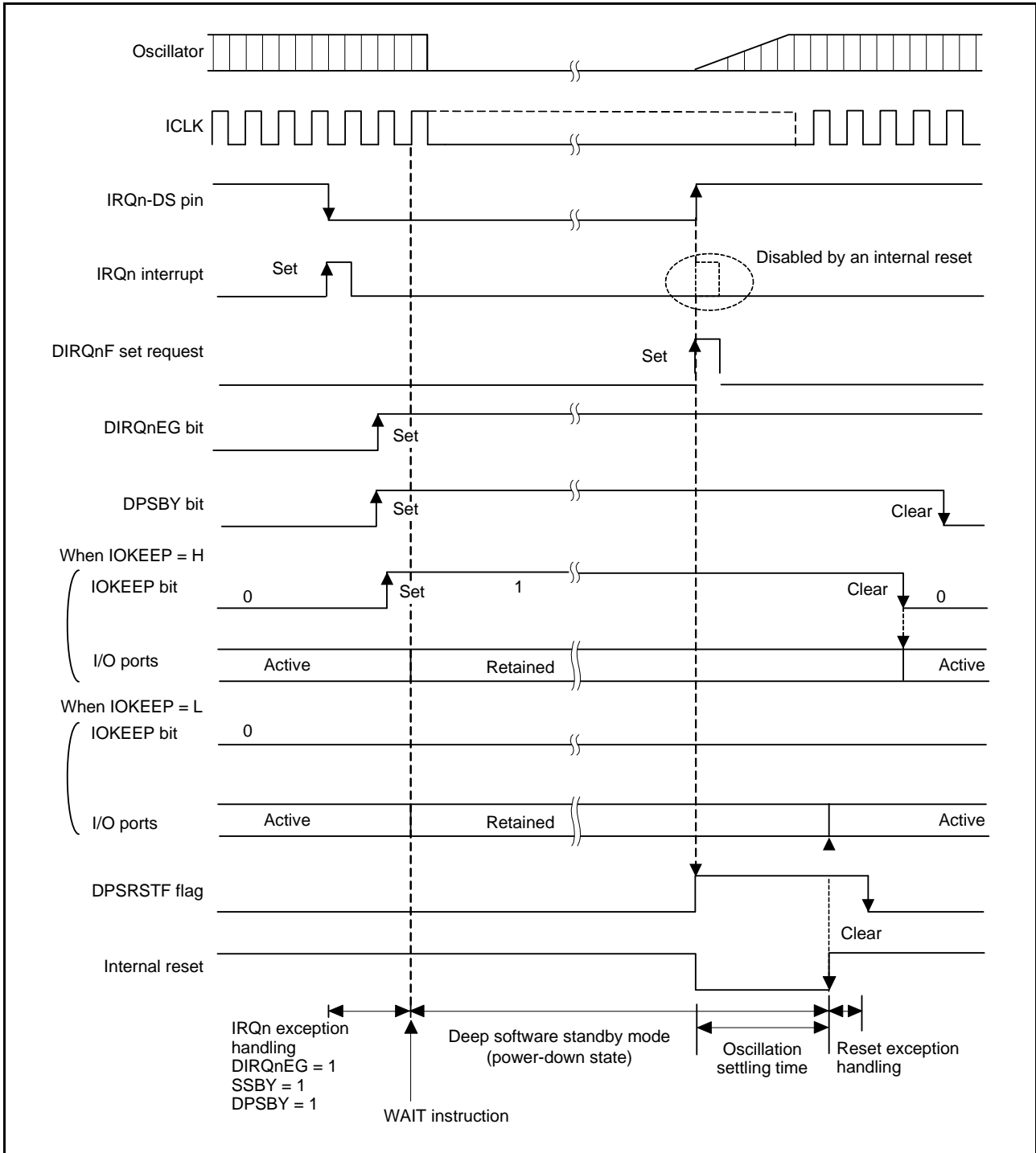


Figure 11.3 Example of Deep Software Standby Mode Application



### 11.6.4.5 Flowchart to Use Deep Software Standby Mode

Figure 11.4 shows an example of a flowchart to use deep software standby mode.

In this example, the RSTSR0.DPSRSTF flag of the reset function is read after the reset exception handling to determine whether a reset was generated by the RES# pin or by the cancellation of deep software standby mode.

In the case of a reset by the RES# pin, a transition to deep software standby mode is made after the required register settings have been made.

In the case of a reset by the cancellation of deep software standby mode, the DPSBYCR.IOKEEP bit is cleared to 0 after the I/O port settings have been made.

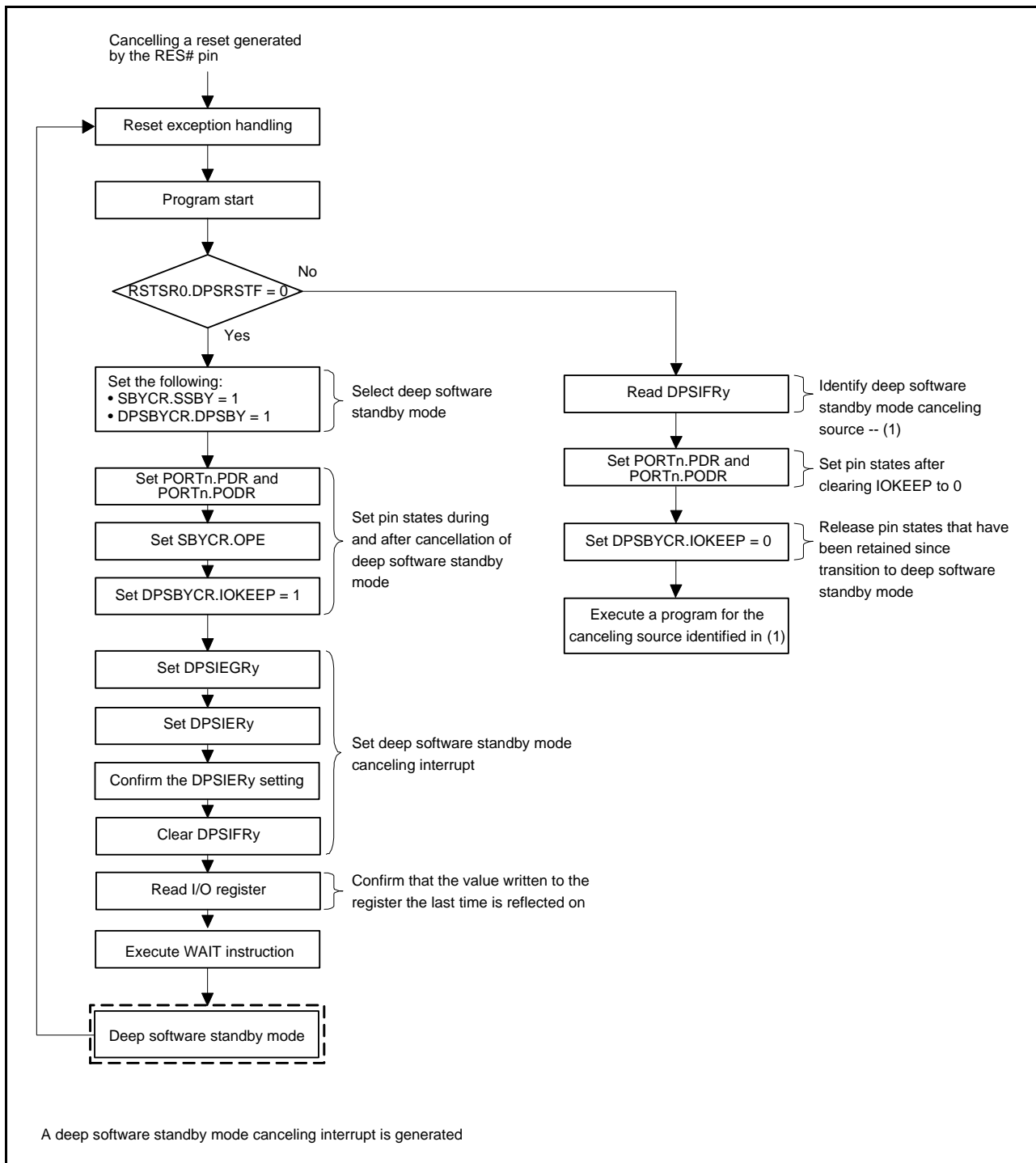


Figure 11.4 Example of Flowchart to Use Deep Software Standby Mode

## 11.7 Usage Notes

### 11.7.1 I/O Port States

I/O port states are retained in software standby mode and deep software standby mode. Therefore, the supply current is not reduced while output signals are held high.

### 11.7.2 Module-Stop State of DMAC and DTC

Before setting the MSTPCRA.MSTPA28 bit to 1, clear the DMAST.DMST bit of the DMAC and the DTCST.DTCST bit of the DTC to 0 so that the DMAC and DTC are not activated.

For details, see section 18, DMA Controller (DMACA) and section 19, Data Transfer Controller (DTCa).

### 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 DMAC or 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 Input Buffer Control by DIRQnE Bit (n = 0 to 15)

Setting the DPSIERy.DIRQnE (y = 0 or 1, n = 0 to 15) bit to 1 enables the input buffer of the IRQ0-DS to IRQ15-DS pins. Therefore, note that, although inputs to these pins are sent to the DPSIFRy.DIRQnF (y = 0 or 1, n = 0 to 15) bits, they are not sent to the interrupt controller, peripheral modules, and I/O ports.

### 11.7.6 Timing of Wait Instructions

The WAIT instruction is executed before completion of the preceding register write. The WAIT instruction may be executed before the change to the setting of an I/O register is reflected, in which case operation may not be as intended. To avoid this, always execute the WAIT instruction after confirming that the last write to the register has completed.

### 11.7.7 Rewrite the Register by DMAC and DTC in Sleep Mode

The WDT stops in sleep mode. Do not set up the DMACA and DTC to rewrite any registers related to the WDT while the chip is in sleep mode.

According to the settings of the OFS0.IWDTSLCSTP bit and IWDTCSSTPR.SLCSTP bit, the IWDT may also stop in sleep mode. If that is the case, do not set up the DMAC and DTC to rewrite any registers related to the IWDT in sleep mode.

The RSTCKCR register can be set so that the clock source is switched on recovery from sleep mode. For this reason, rewriting the register while the chip is in sleep mode may lead to unintended operation, so do not allow rewriting of the RSTCKCR register in sleep mode.

### 11.7.8 Transition to Software Standby Mode for Products with 1.5-Mbyte ROM or More or Products with 176 Pins or More

With the product with 1.5-Mbyte ROM or more or the product with the 176 pins or more, software standby mode is not entered even when a WAIT instruction is executed while the main clock oscillator or PLL circuit is selected as the clock source for the system clock. To enter software standby mode, select the low-speed on-chip oscillator (LOCO), high-speed on-chip oscillator (HOCO), or sub-clock oscillator as the clock source before executing the WAIT instruction.

### 11.7.9 Canceling All-Module Clock Stop Mode

If the ICLK is set so as to be slower than the PCLKB, a TMR interrupt cannot be used to cancel all-module clock stop mode. To use the TMR interrupt as the all-module clock stop mode cancelling source, change the ICLK so as to be faster than the PCLKB before all-module clock stop mode is entered.

### 11.7.10 Point for Caution when Using the Sub-Clock as the Source of the System Clock

If the sub-clock is in use as the source of the system clock, make sure that the RTC or the low-speed on-chip oscillator is operating (by setting the RCR3.RTCEN = 1 or the LOCOCR.LCSTP = 0, respectively) for a transition to software standby mode.

### 11.7.11 Points for Caution on Return from Software Standby

When interrupts IRQ0 to IRQ15 are not set as triggers for release from software standby, the corresponding input buffers become invalid on software standby, so the input signal within the LSI is fixed to the high level. Accordingly, depending on the state of the pin, a transition to software standby may set the interrupt status flag (ICU.IRi.IR) to 1.

When placing the chip on software standby, execute the WAIT instruction after setting the IERi.IENj bits to prohibit whichever of interrupts IRQ0 to IRQ15 are not to be used as triggers for release from standby. Moreover, after return from software standby, clear the interrupt status flags.

### 11.7.12 Point for Caution when Shifting from Low-Speed Operation Mode to Software Standby Mode

On return from software standby, the chip enters the high-speed operating mode. Even if a WAIT instruction is executed, if release from software standby precedes completion of the transition, the mode remains the same as before execution of the WAIT instruction. If this creates a problem, set the OPCCR.OPCM[2:0] bits to 000b during processing of the return interrupt.

## 12. Battery Backup Function

### 12.1 Overview

When the voltage at the VCC pin is dropped, power can be supplied to the realtime clock (RTC) and sub-clock oscillator from the dedicated battery backup power pin (VBATT pin). When the voltage drop at the VCC pin is detected, connection to power is switched to the VBATT pin.

Figure 12.1 shows the configuration of the battery backup function.

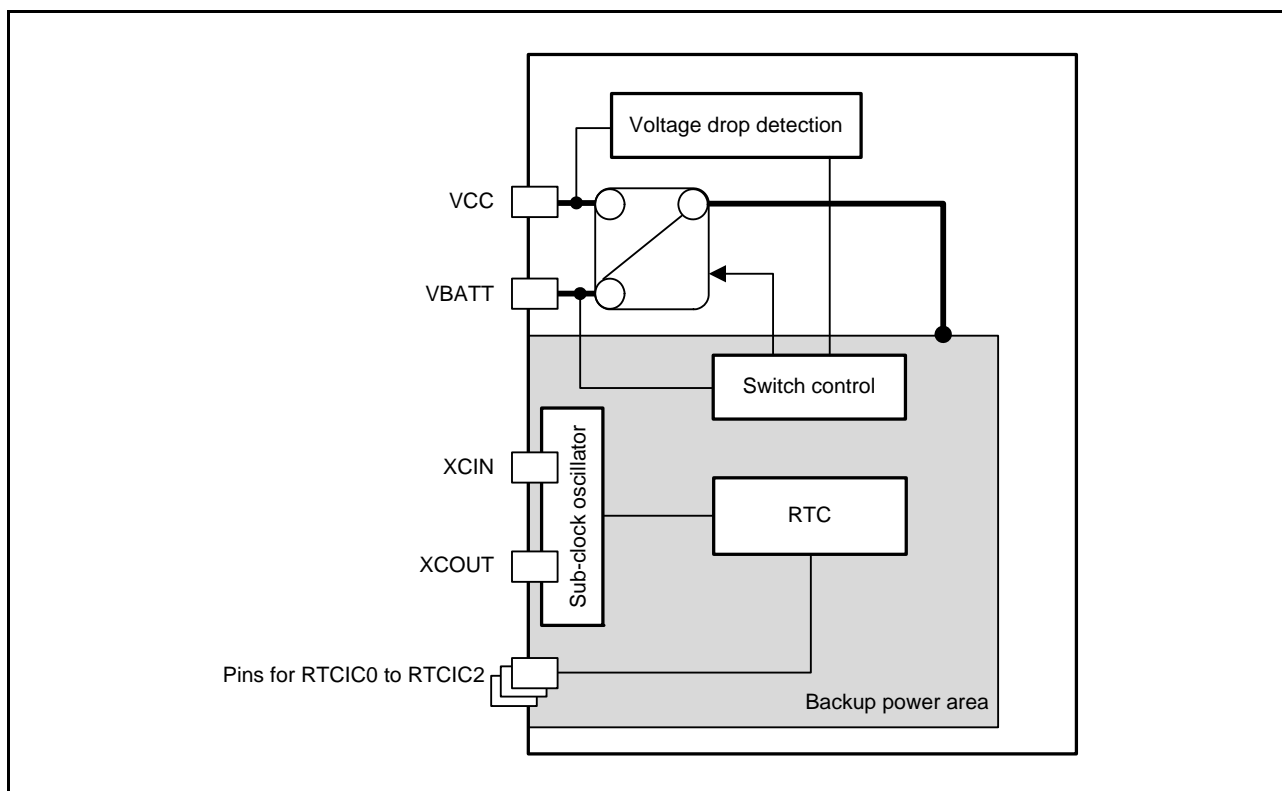


Figure 12.1 Configuration of Battery Backup Function

## 12.2 Operation

### 12.2.1 Battery Backup Function

When the voltage at the VCC pin is dropped, power can be supplied to the RTC and sub-clock oscillator from the VBATT pin. When the power supply reduction from the VCC pin is detected, connection to power is switched to the power supply from the VBATT pin. The power supply from the VCC pin is resumed when the voltage at the VCC pin exceeds VDET<sub>BATT</sub> while the RTC is operating on the power supply from the VBATT pin. This power supply change does not affect the RTC operation. When the voltage level at the VBATT pin voltage falls below the operation guaranteed voltage, operation of the RTC cannot be guaranteed.

In addition, the battery backup function should be used after the voltage monitoring 0 reset is enabled.

The power is supplied to the following modules from the VBATT pin.

- RTC
- Sub-clock oscillator (including XCIN and XCOU pins)

Figure 12.2 shows the operation for switching to the battery backup function.

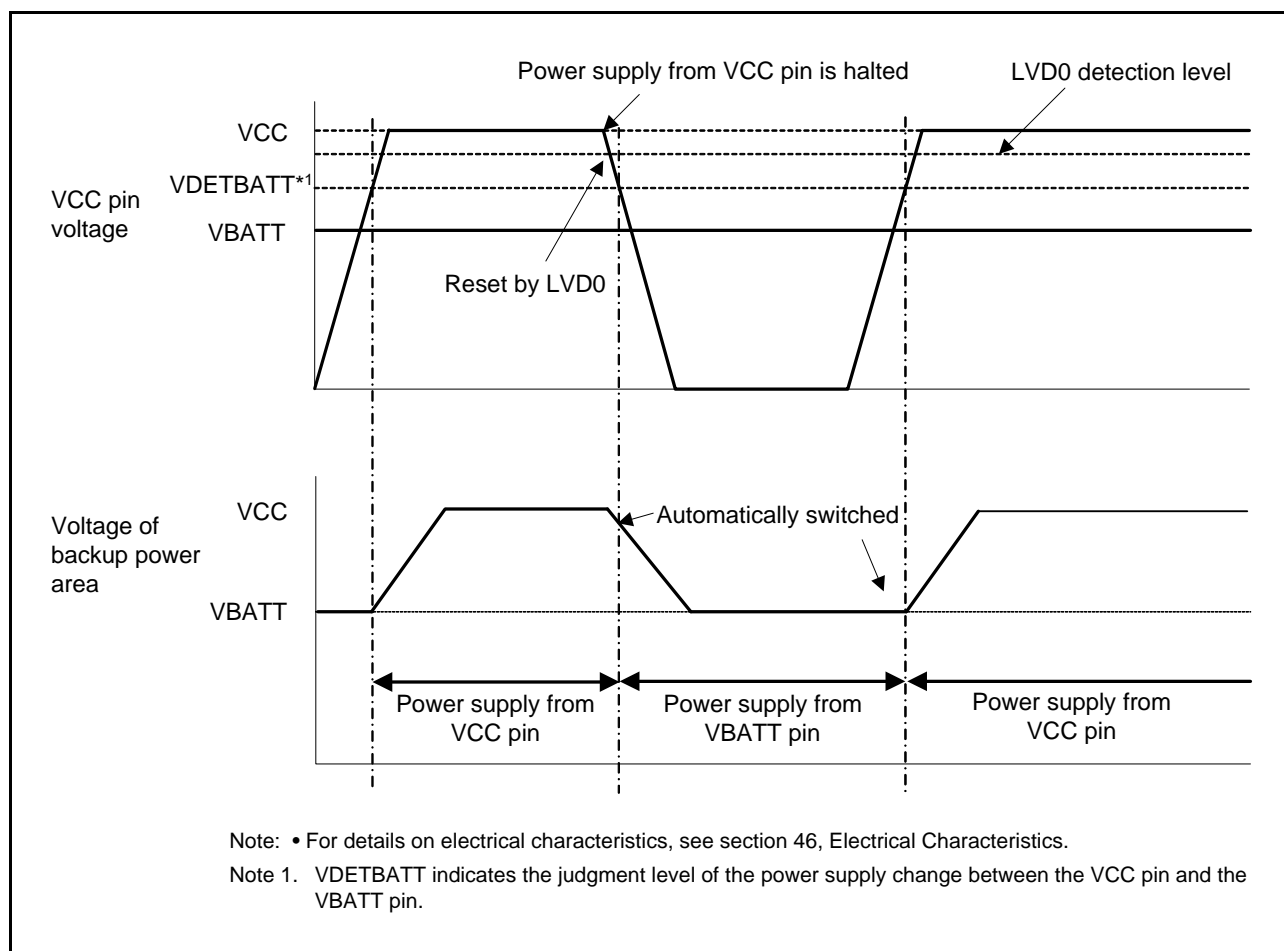


Figure 12.2 Operation for Switching to Battery Backup Function

### 12.3 Usage Notes

1. When the VBATT pin is not in use, connect the VBATT pin to the VCC pin.
2. When the voltage level at the VBATT is lower than the guaranteed operation range, operation of the sub-clock and RTC cannot be guaranteed. The RTC must be initialized to restart power supply after the VBATT pin falls below the operation guaranteed voltage.
3. Writing to the RTC registers should be performed while power is being supplied from the VCC pin.
4. When VCC is higher than VDET<sub>BATT</sub>, the VCC pin and VBATT pin are not connected by means of circuitry. When VCC is lower than VDET<sub>BATT</sub> and the switch is connected to the VBATT pin, if the voltage at the VBATT pin becomes lower than (the range from VCC to 0.6 V), current may flow into the VBATT pin through the parasitic diode between the VCC and VBATT pins.
5. During RTC operation using the voltage from the VBATT pin, I/O ports (P30, P31, and P32) within the backup power supply area can only be used as time capture event input pins for the RTC.

## 13. 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 13.1 lists the association between the PRCR bits and the registers to be protected.

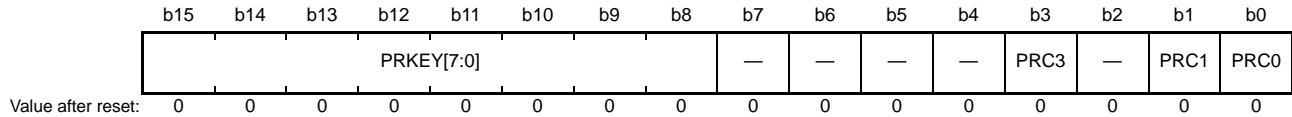
**Table 13.1 Association between PRCR Bits and Registers to be Protected**

PRCR Bit	Register to be Protected
PRC0	<ul style="list-style-type: none"> <li>Registers related to the clock generation circuit: SCKCR, SCKCR2, SCKCR3, PLLCR, PLLCR2, BCKCR, MOSCCR, SOSCCR, LOCOCR, ILOCOCR, HOCOGR, OSTDCR, OSTDSR</li> </ul>
PRC1	<ul style="list-style-type: none"> <li>Registers related to the operating modes: SYSCR0, SYSCR1</li> <li>Registers related to the low power consumption functions: SBYCR, MSTPCRA, MSTPCRB, MSTPCRC, OPCCR, RSTCKCR, MOSCWTCR, SOSCWTCR, PLLWTCR, DPSBYCR, DPSIER0 to DPSIER3, DPSIFR0 to DPSIFR3, DPSIEGR0 to DPSIEGR3</li> <li>Registers related to clock generation circuit: MOFCR, HOCOPCR</li> <li>Software reset register: SWRR</li> </ul>
PRC3	<ul style="list-style-type: none"> <li>Registers related to the LVD: LVCMPCCR, LVDLVLRL, LVD1CR0, LVD1CR1, LVD1SR, LVD2CR0, LVD2CR1, LVD2SR</li> </ul>

## 13.1 Register Descriptions

### 13.1.1 Protect Register (PRCR)

Address(es): 0008 03FEh



Bit	Symbol	Bit Name	Function	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, 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 eight higher-order bits and the desired value to the eight 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 enables and disables writing to the corresponding registers to be protected, respectively.



## 14. Exception Handling

### 14.1 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 14.1.

The occurrence of an exception causes the processor mode to shift to supervisor mode.

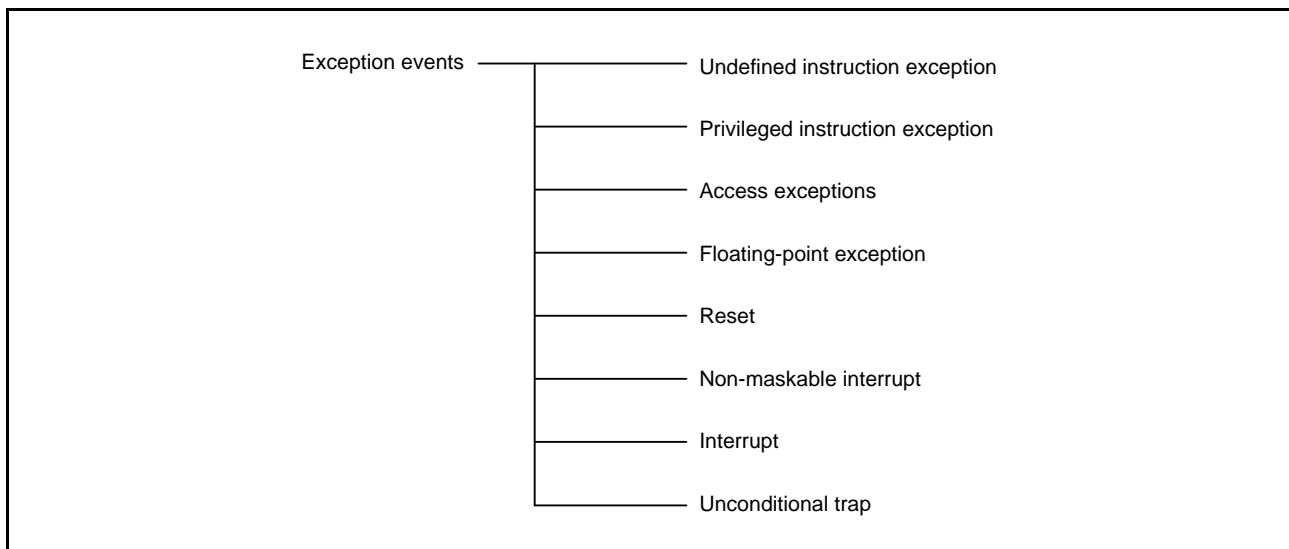


Figure 14.1 Types of Exception Events

#### 14.1.1 Undefined Instruction Exception

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

#### 14.1.2 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.

#### 14.1.3 Access Exceptions

An access exception occurs when an error is detected in access to memory by the CPU. If the memory-protection unit detects an instruction memory-protection error, an instruction-access exception occurs, and if the unit detects a data memory protection error, an operand-access exception occurs.

#### 14.1.4 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.

#### 14.1.5 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.

### 14.1.6 Non-Maskable Interrupt

The 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.

### 14.1.7 Interrupt

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.

### 14.1.8 Unconditional Trap

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

### 14.2 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 14.2 shows the processing procedure when an exception other than a reset is accepted.

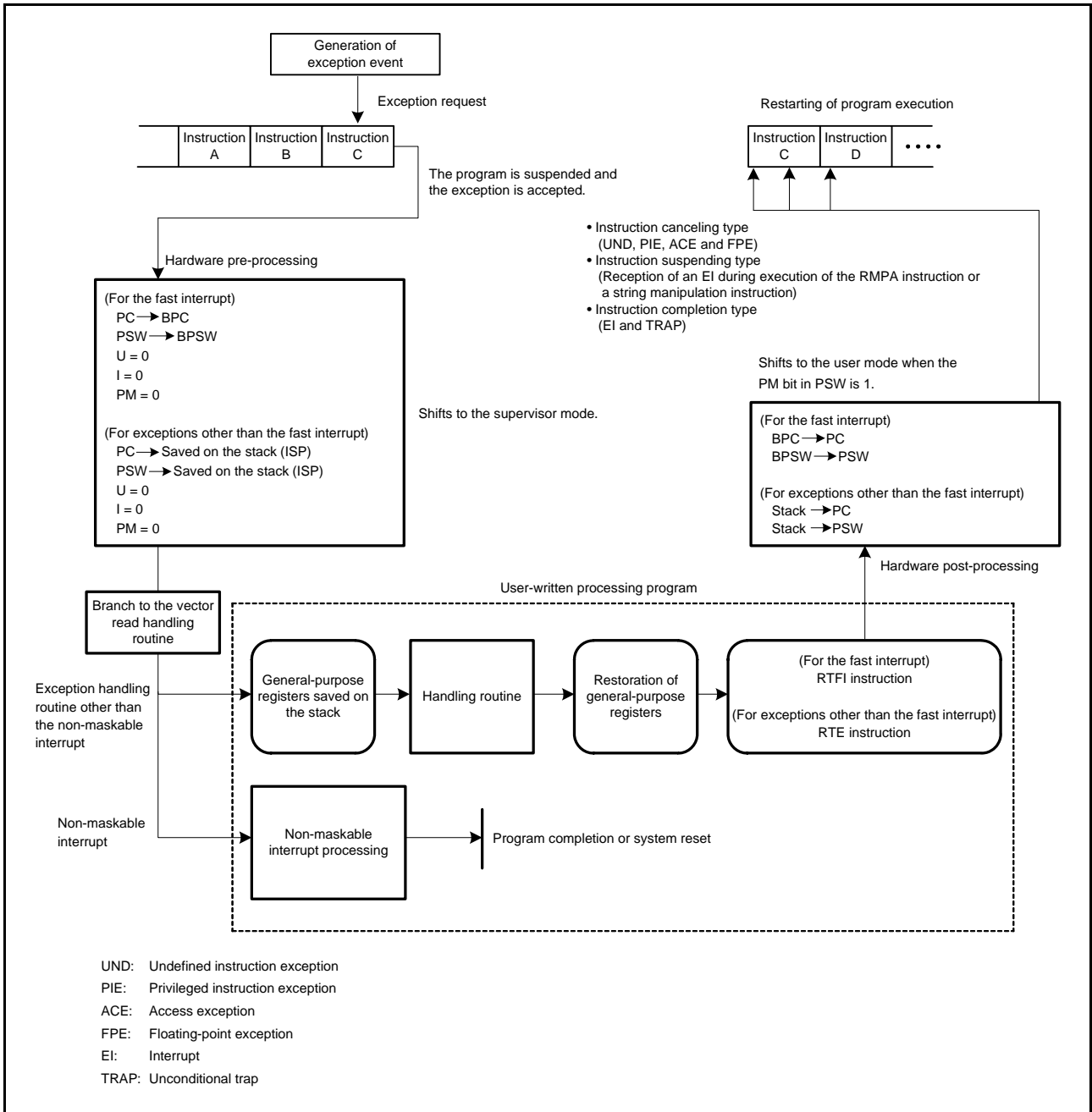


Figure 14.2 Outline of Exception Handling Procedure

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, registers saved on the stack are restored and the RTE instruction is executed to restore execution from the exception handling routine to the original program. 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.

### 14.3 Acceptance of Exception Events

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

#### 14.3.1 Acceptance Timing and Saved PC Value

Table 14.1 lists the timing of acceptance and the program counter (PC) value to be saved for each exception event.

**Table 14.1 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	
Access 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	

### 14.3.2 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 14.2.

**Table 14.2 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
Access 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
	Other than above	Relocatable vector table (INTB)
Unconditional trap	Relocatable vector table (INTB)	Stack

## 14.4 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

## 14.5 Hardware Pre-Processing

The hardware pre-processing from reception of each exception request to execution of the associated exception handling routine are explained below.

### 14.5.1 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 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 FFDCh.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

### 14.5.2 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 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 FFD0h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

### 14.5.3 Access Exceptions

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 FFFFFFFD4h.
5. The fetched vector is set to the PC and processing branches to the exception handling routine.

### 14.5.4 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.

#### 14.5.5 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.

#### 14.5.6 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 cleared 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.

#### 14.5.7 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 cleared 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.

#### 14.5.8 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 cleared 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.

## 14.6 Return from Exception Handling Routine

Executing the instruction listed in Table 14.3 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 14.3 Return from Exception Handling Routine**

Exception	Instruction for Return	
Undefined instruction exception	RTE	
Privileged instruction exception	RTE	
Access 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	

## 14.7 Priority of Exception Events

The priority of exception events is listed in Table 14.4. When multiple exceptions are generated at the same time, the exception with the highest priority is accepted first.

**Table 14.4 Priority of Exception Events**

Priority	Exception Event
High  Low	1 Reset
	2 Non-maskable interrupt
	3 Interrupt
	4 Instruction access exception
	5 Undefined instruction exception Privileged instruction exception
	6 Unconditional trap
	7 Operand access exception
	8 Floating-point exception



## 15. Interrupt controller (ICUb)

### 15.1 Overview

The interrupt controller receives interrupt signals from the peripheral modules and external pins, sends interrupts to the CPU, and activates the DTC and DMAC.

Table 15.1 lists the specifications of the interrupt controller, and Figure 15.1 shows a block diagram of the interrupt controller.

**Table 15.1 Specifications of Interrupt Controller**

Item	Description
Interrupts	Peripheral function interrupts <ul style="list-style-type: none"> <li>• Interrupts from peripheral modules</li> <li>• Interrupt detection: Edge detection/level detection Edge detection or level detection is fixed for each source of connected peripheral modules.</li> <li>• Interrupt grouping: Multiple interrupt requests can be allocated to a single interrupt vector. Number of groups for edge detection interrupts: 7 (groups 0 to 6) Number of groups for level detection interrupts: 1 (group 12)</li> <li>• Interrupt unit selection: One of the two interrupt request units can be selected as an interrupt request source. Number of interrupts per unit: 6</li> </ul>
External pin interrupts	<ul style="list-style-type: none"> <li>• Interrupts from pins IRQ0 to IRQ15</li> <li>• Number of sources: 16</li> <li>• Interrupt detection: Low level/falling edge/rising edge/rising and falling edges One of these detection methods can be set for each source.</li> <li>• Digital filter function: supported</li> </ul>
Software interrupt	<ul style="list-style-type: none"> <li>• Interrupt generated by writing to a register</li> <li>• One interrupt source</li> </ul>
Interrupt priority	Specified by registers.
Fast interrupt function	Faster interrupt processing of the CPU can be set only for a single interrupt source.
DTC/DMAC control	The DTC and DMAC can be activated by interrupt sources.*1
Non-maskable interrupts	NMI pin interrupt <ul style="list-style-type: none"> <li>• Interrupt from the NMI pin</li> <li>• Interrupt detection: Falling edge/rising edge</li> <li>• Digital filter function: supported</li> </ul>
Oscillation stop detection interrupt	Interrupt on detection of oscillation having stopped
WDT underflow/refresh error	Interrupt on an underflow of the down-counter or occurrence of a refresh error
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 detection circuit 1 (LVD1)
Voltage-monitoring 2 interrupt	Voltage monitoring interrupt of voltage detection circuit 2 (LVD2)
Return from power-down modes	<ul style="list-style-type: none"> <li>• Sleep mode: Return is initiated by non-maskable interrupts or any other interrupt source.</li> <li>• All-module clock stop mode: Return is initiated by non-maskable interrupts, IRQ0 to IRQ15 interrupts, TMR interrupts, USB resume interrupts, or RTC alarm/periodic interrupts.</li> <li>• Software standby mode: Return is initiated by non-maskable interrupts, IRQ0 to IRQ15 interrupts, USB resume interrupts, or RTC alarm/periodic interrupts.</li> </ul>

Note 1. For the DTC and DMAC activation sources, refer to Table 15.3, Interrupt Vector Table.

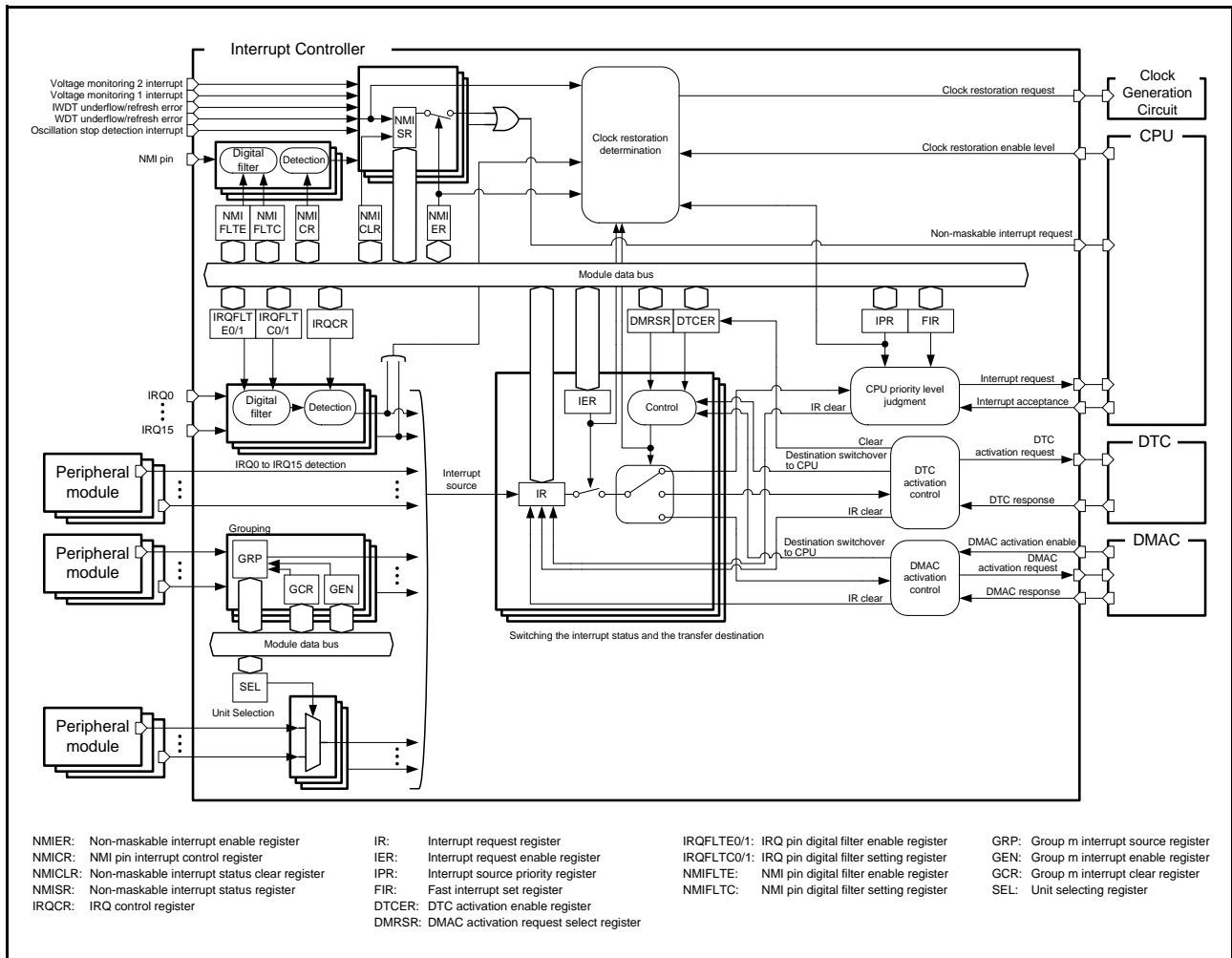


Figure 15.1 Block Diagram of Interrupt Controller

Table 15.2 lists the input/output pins of the interrupt controller.

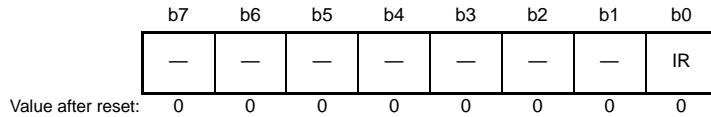
Table 15.2 Pin Configuration of Interrupt Controller

Pin Name	I/O	Description
NMI	Input	Non-maskable interrupt request pin
IRQ0 to IRQ15	Input	External interrupt request pins

## 15.2 Register Descriptions

### 15.2.1 Interrupt Request Register n (IRn) (n = interrupt vector number)

Address(es): 0008 7010h to 0008 70FDh



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 15.3, Interrupt Vector Table.

#### IR Flag (Interrupt Status Flag)

This bit 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 pins, edge detection or level detection is selected by the setting of the IRQMD[1:0] bits in the corresponding IRQCRi (i = 0 to 15). For detection of the various interrupt sources, see Table 15.3, Interrupt Vector Table.

Grouped interrupt requests are detected using group m interrupt source register (GRPm; m = group number). The interrupt request detected by GRPm is further detected as a level detection interrupt request by the IR flag corresponding to the group. For details of the interrupt grouping function, see section 15.4, Peripheral Module Interrupt Request Groups and Unit Selection.

For grouped interrupt requests, see Table 15.4, Group m Interrupt Requests (1/2).

#### (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 or DMAC.

## (2) Level detection

[Setting conditions]

- The flag remains set to 1 while an interrupt request is being sent from the corresponding peripheral module or IRQ<sub>i</sub> pin. For interrupt generation by the various peripheral modules, refer to the sections describing the modules.
- For grouped interrupt requests, the flag is set to 1 when both the interrupt request enable bit *j* in group *m* interrupt enable register (GEN<sub>m</sub>.EN<sub>j</sub>; *m* = group number and *j* = bit number) and interrupt status flag in group *m* interrupt source register (GRP<sub>m</sub>.IS<sub>j</sub>) are 1.

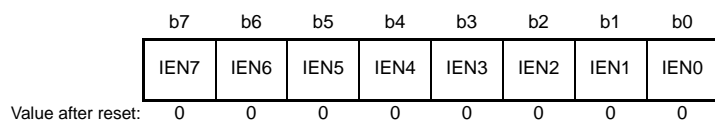
[Clearing conditions]

- 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.
- For grouped interrupt requests, the flag is cleared to 0 when either the GEN<sub>m</sub>.EN<sub>j</sub> bit or GRP<sub>m</sub>.IS<sub>j</sub> flag is 0.

When level detection has been selected for an IRQ<sub>i</sub> pin, the interrupt request is withdrawn by driving the IRQ<sub>i</sub> pin high. Do not write 0 or 1 to the IR flag while level detection is selected.

### 15.2.2 Interrupt Request Enable Register *m* (IER<sub>m</sub>) (*m* = 02h to 1Fh)

Address(es): 0008 7202h to 0008 721Fh



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.

#### IEN<sub>j</sub> Bits (Interrupt Request Enable *j*) (*j* = 7 to 0)

When an IEN<sub>j</sub> bit is 1, the corresponding interrupt request will be output to the destination selected for the request. When an IEN<sub>j</sub> bit is 0, the corresponding interrupt request will not be output to the destination selected for the request. The setting of an IEN<sub>j</sub> bit does not affect the IR<sub>n</sub>.IR flag. Even if the corresponding IEN<sub>j</sub> bit is 0, the IR flag value changes according to the descriptions in section 15.2.1, Interrupt Request Register *n* (IR<sub>n</sub>) (*n* = interrupt vector number).

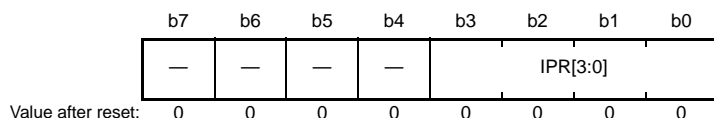
The IER<sub>m</sub>.IEN<sub>j</sub> bit is set for each request source (vector number).

For the correspondence between interrupt sources and IER<sub>m</sub>.IEN<sub>j</sub> bits, see Table 15.3, Interrupt Vector Table.

For the procedure for setting IER<sub>m</sub>.IEN<sub>j</sub> bits during the selection of destinations for interrupt requests, refer to section 15.5.3, Selecting Interrupt Request Destinations.

### 15.2.3 Interrupt Source Priority Register n (IPRn) (n = 000 to 253)

Address(es): 0008 7300h to 0008 73FDh



Bit	Symbol	Bit Name	Description	R/W																																																			
b3 to b0	IPR[3:0]	Interrupt Priority Level Select	<table border="0"> <tr> <td>b3</td> <td>b0</td> <td></td> </tr> <tr> <td>0 0 0 0</td> <td>0</td> <td>: Level 0 (interrupt disabled)*1</td> </tr> <tr> <td>0 0 0 1</td> <td>1</td> <td>: Level 1</td> </tr> <tr> <td>0 0 1 0</td> <td>0</td> <td>: Level 2</td> </tr> <tr> <td>0 0 1 1</td> <td>1</td> <td>: Level 3</td> </tr> <tr> <td>0 1 0 0</td> <td>0</td> <td>: Level 4</td> </tr> <tr> <td>0 1 0 1</td> <td>1</td> <td>: Level 5</td> </tr> <tr> <td>0 1 1 0</td> <td>0</td> <td>: Level 6</td> </tr> <tr> <td>0 1 1 1</td> <td>1</td> <td>: Level 7</td> </tr> <tr> <td>1 0 0 0</td> <td>0</td> <td>: Level 8</td> </tr> <tr> <td>1 0 0 1</td> <td>1</td> <td>: Level 9</td> </tr> <tr> <td>1 0 1 0</td> <td>0</td> <td>: Level 10</td> </tr> <tr> <td>1 0 1 1</td> <td>1</td> <td>: Level 11</td> </tr> <tr> <td>1 1 0 0</td> <td>0</td> <td>: Level 12</td> </tr> <tr> <td>1 1 0 1</td> <td>1</td> <td>: Level 13</td> </tr> <tr> <td>1 1 1 0</td> <td>0</td> <td>: Level 14</td> </tr> <tr> <td>1 1 1 1</td> <td>1</td> <td>: Level 15 (highest)</td> </tr> </table>	b3	b0		0 0 0 0	0	: Level 0 (interrupt disabled)*1	0 0 0 1	1	: Level 1	0 0 1 0	0	: Level 2	0 0 1 1	1	: Level 3	0 1 0 0	0	: Level 4	0 1 0 1	1	: Level 5	0 1 1 0	0	: Level 6	0 1 1 1	1	: Level 7	1 0 0 0	0	: Level 8	1 0 0 1	1	: Level 9	1 0 1 0	0	: Level 10	1 0 1 1	1	: Level 11	1 1 0 0	0	: Level 12	1 1 0 1	1	: Level 13	1 1 1 0	0	: Level 14	1 1 1 1	1	: Level 15 (highest)	R/W
b3	b0																																																						
0 0 0 0	0	: Level 0 (interrupt disabled)*1																																																					
0 0 0 1	1	: Level 1																																																					
0 0 1 0	0	: Level 2																																																					
0 0 1 1	1	: Level 3																																																					
0 1 0 0	0	: Level 4																																																					
0 1 0 1	1	: Level 5																																																					
0 1 1 0	0	: Level 6																																																					
0 1 1 1	1	: Level 7																																																					
1 0 0 0	0	: Level 8																																																					
1 0 0 1	1	: Level 9																																																					
1 0 1 0	0	: Level 10																																																					
1 0 1 1	1	: Level 11																																																					
1 1 0 0	0	: Level 12																																																					
1 1 0 1	1	: Level 13																																																					
1 1 1 0	0	: Level 14																																																					
1 1 1 1	1	: Level 15 (highest)																																																					
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 15.3, 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 activation requests to the DTC and DMAC.

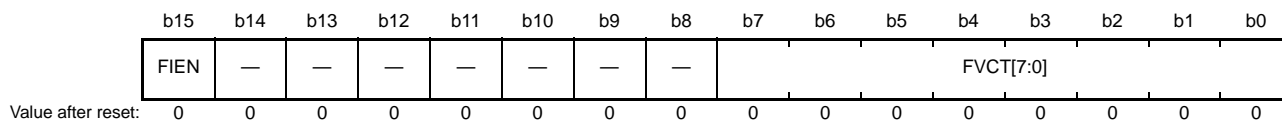
The CPU accepts only interrupt requests higher than the priority level specified by the IPL[3:0] bits in PSW, and handles accepted interrupts.

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).

## 15.2.4 Fast Interrupt Set Register (FIR)

Address(es): 0008 72F0h



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 or DMAC.

Before writing to this register, be sure to disable interrupt requests (IERm.IENj bit = 0).

### 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. When using the fast interrupt for returning from the software standby mode, see [section 15.7.3, Return from Software Standby Mode](#).

If the setting of the IERm.IENj (m = 02h to 1Fh, j = 7 to 0) 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.

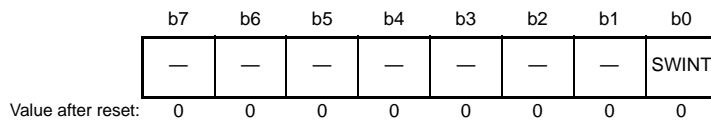
For settable vector numbers, see [Table 15.3, Interrupt Vector Table](#).

Do not write any reserved vector numbers to the FVCT[7:0] bits.

For details on the fast interrupt, see [section 14, Exception Handling](#), and [section 15.5.5, Fast Interrupt](#).

### 15.2.5 Software Interrupt Activation Register (SWINTR)

Address(es): 0008 72E0h



Bit	Symbol	Bit Name	Description	R/W
b0	SWINT	Software Interrupt Activation	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 Activation)

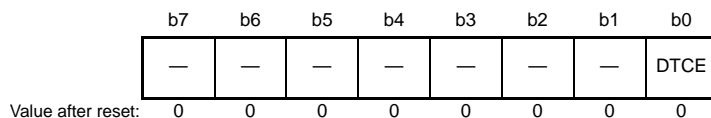
When 1 is written to the SWINT bit, the interrupt request register 027 (IR027) is set to 1.

If 1 is written to the SWINT bit when the DTC activation 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 activation enable register 027 (DTCER027) is set to 1, a DTC activation request is issued.

### 15.2.6 DTC Activation Enable Register n (DTCERn) (n = interrupt vector number)

Address(es): 0008 711Bh to 0008 71FBh



Bit	Symbol	Bit Name	Description	R/W
b0	DTCE	DTC Activation Enable	0: DTC activation is disabled 1: DTC activation is enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

An interrupt source that has been selected as a source for DMAC activation should not be specified as a source for DTC activation. See Table 15.3, Interrupt Vector Table, for the interrupt sources that are selectable as sources for DTC activation.

#### DTCE Bit (DTC Activation Enable)

When the DTCE bit is set to 1, the corresponding interrupt source is selected as the source for the DTC activation.

[Setting condition]

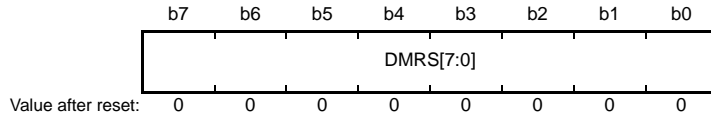
- When 1 is written to the DTCE bit

[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

### 15.2.7 DMAC Activation Request Select Register m (DMRSRm) (m = DMAC channel number)

Address(es): DMRSR0 0008 7400h, DMRSR1 0008 7404h  
DMRSR2 0008 7408h, DMRSR3 0008 740Ch



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DMRS[7:0]	DMAC Activation Source Select	These bits specify the vector number for the DMAC activation request.	R/W

To specify the same interrupt source for multiple DMRSRm registers is disabled. The interrupt source that has been selected for the DMRSRm activation should not be specified as the source for the DTC activation. Otherwise, the correct operation is not guaranteed.

#### DMRS[7:0] Bits (DMAC Activation Source Select)

The vector number of the interrupt source for DMAC activation is specified in 8 bits. Do not set the vector numbers that are not assigned for the DMAC activation.

For the correspondence between interrupt sources and interrupt vector numbers, see Table 15.3, Interrupt Vector Table.

Write to the DMRSRm register while the DMA transfer enable bit of the DMA transfer enable register (DMACm.DMCNT.DTE) is cleared to 0.



### 15.2.8 IRQ Control Register i (IRQCRi) (i = 0 to 15)

Address(es): 0008 7500h to 0008 750Fh



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 (IEN<sub>j</sub> bit in IER<sub>m</sub> is 0). After changing the setting, clear the IR flag in IR<sub>n</sub> 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 detection sensing method of external pin interrupt sources IRQ0 to IRQ15. For the external pin interrupt detection setting, see section 15.5.7, External Pin Interrupts.

### 15.2.9 IRQ Pin Digital Filter Enable Register 0 (IRQFLTE0)

Address(es): 0008 7510h

b7	b6	b5	b4	b3	b2	b1	b0
FLTEN 7	FLTEN 6	FLTEN 5	FLTEN 4	FLTEN 3	FLTEN 2	FLTEN 1	FLTEN 0

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
b6	FLTEN6	IRQ6 Digital Filter Enable		R/W
b7	FLTEN7	IRQ7 Digital Filter Enable		R/W

#### FLTEN<sub>i</sub> Bits (IRQ<sub>i</sub> Digital Filter Enable) (i = 0 to 7)

These bits enable the digital filter used for the external pin interrupt sources IRQ0 to IRQ7.

The digital filter is enabled when the FLTEN<sub>i</sub> bit is 1, and disabled when the FLTEN<sub>i</sub> bit is 0.

The IRQ<sub>i</sub> 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 15.5.6, Digital Filter.

### 15.2.10 IRQ Pin Digital Filter Enable Register 1 (IRQFLTE1)

Address(es): 0008 7511h

b7	b6	b5	b4	b3	b2	b1	b0
FLTEN 15	FLTEN 14	FLTEN 13	FLTEN 12	FLTEN 11	FLTEN 10	FLTEN 9	FLTEN 8

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	FLTEN8	IRQ8 Digital Filter Enable	0: Digital filter is disabled. 1: Digital filter is enabled.	R/W
b1	FLTEN9	IRQ9 Digital Filter Enable		R/W
b2	FLTEN10	IRQ10 Digital Filter Enable		R/W
b3	FLTEN11	IRQ11 Digital Filter Enable		R/W
b4	FLTEN12	IRQ12 Digital Filter Enable		R/W
b5	FLTEN13	IRQ13 Digital Filter Enable		R/W
b6	FLTEN14	IRQ14 Digital Filter Enable		R/W
b7	FLTEN15	IRQ15 Digital Filter Enable		R/W

#### FLTEN<sub>i</sub> Bits (IRQ<sub>i</sub> Digital Filter Enable) (i = 8 to 15)

These bits enable the digital filter used for the external pin interrupt sources IRQ8 to IRQ15.

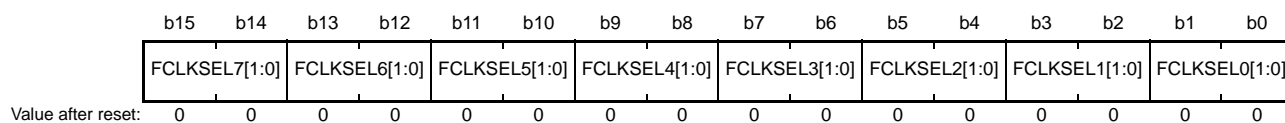
The digital filter is enabled when the FLTEN<sub>i</sub> bit is 1, and disabled when the FLTEN<sub>i</sub> bit is 0.

The IRQ<sub>i</sub> pin level is sampled at the sampling clock cycle specified with the IRQFLTC1.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 15.5.6, Digital Filter.

### 15.2.11 IRQ Pin Digital Filter Setting Register 0 (IRQFLTC0)

Address(es): 0008 7514h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	FCLKSEL0[1:0]	IRQ0 Digital Filter Sampling Clock	0 0: PCLK 0 1: PCLK/8	R/W
b3, b2	FCLKSEL1[1:0]	IRQ1 Digital Filter Sampling Clock	1 0: PCLK/32 1 1: PCLK/64	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
b13, b12	FCLKSEL6[1:0]	IRQ6 Digital Filter Sampling Clock		R/W
b15, b14	FCLKSEL7[1:0]	IRQ7 Digital Filter Sampling Clock		R/W

#### FCLKSELi[1:0] Bits (IRQi Digital Filter Sampling Clock) (i = 0 to 7)

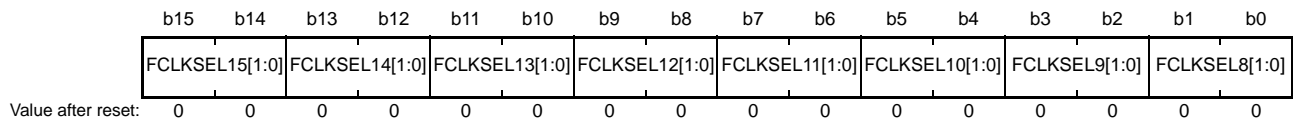
These bits select the cycle of the digital filter sampling clock for the external pin interrupt request pins IRQ0 to IRQ7.

The sampling clock cycle can be selected from among the PCLK (every cycle), PCKL/8 (once every eight cycles), PCKL/32 (once every 32 cycles), and PCKL/64 (once every 64 cycles).

For details of the digital filter, see section 15.5.6, Digital Filter.

### 15.2.12 IRQ Pin Digital Filter Setting Register 1 (IRQFLTC1)

Address(es): 0008 7516h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	FCLKSEL8[1:0]	IRQ8 Digital Filter Sampling Clock	0 0: PCLK 0 1: PCLK/8	R/W
b3, b2	FCLKSEL9[1:0]	IRQ9 Digital Filter Sampling Clock	1 0: PCLK/32	R/W
b5, b4	FCLKSEL10[1:0]	IRQ10 Digital Filter Sampling Clock	1 1: PCLK/64	R/W
b7, b6	FCLKSEL11[1:0]	IRQ11 Digital Filter Sampling Clock		R/W
b9, b8	FCLKSEL12[1:0]	IRQ12 Digital Filter Sampling Clock		R/W
b11, b10	FCLKSEL13[1:0]	IRQ13 Digital Filter Sampling Clock		R/W
b13, b12	FCLKSEL14[1:0]	IRQ14 Digital Filter Sampling Clock		R/W
b15, b14	FCLKSEL15[1:0]	IRQ15 Digital Filter Sampling Clock		R/W

#### FCLKSELi[1:0] Bits (IRQi Digital Filter Sampling Clock) (i = 8 to 15)

These bits select the cycle of the digital filter sampling clock for the external pin interrupt request pins IRQ8 to IRQ15.

The sampling clock cycle can be selected from among the PCLK (every cycle), PCKL/8 (once every eight cycles), PCKL/32 (once every 32 cycles), and PCKL/64 (once every 64 cycles).

For details of the digital filter, see section 15.5.6, Digital Filter.

### 15.2.13 Non-Maskable Interrupt Status Register (NMISR)

Address(es): 0008 7580h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	LVD2S T	LVD1S T	IWDTS T	WDTST	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	WDTST	WDT Underflow/Refresh Error Status Flag	0: WDT underflow/refresh error interrupt is not requested. 1: WDT underflow/refresh error interrupt is requested.	R
b3	IWDTS	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 and cannot be modified.	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

**WDTST Flag (WDT Underflow/Refresh Error Status Flag)**

This flag indicates the WDT underflow/refresh error interrupt request.  
The WDTST flag is read-only, and cleared by the NMICLR.WDTCLR bit.

[Setting condition]

- When the WDT underflow/refresh error interrupt is generated

[Clearing condition]

- When 1 is written to the NMICLR.WDTCLR 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

## 15.2.14 Non-Maskable Interrupt Enable Register (NMIER)

Address(es): 0008 7581h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	LVD2E N	LVD1E N	IWDTE N	WDTE N	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	WDTEN	WDT Underflow/Refresh Error Enable	0: WDT underflow/refresh error interrupt is disabled 1: WDT underflow/refresh error interrupt is enabled	R/(W) *1
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.

### WDTEN Bit (WDT Underflow/Refresh Error Enable)

This bit enables the WDT 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.

### 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.

**15.2.15 Non-Maskable Interrupt Status Clear Register (NMICLR)**

Address(es): 0008 7582h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	LVD2C LR	LVD1C LR	IWDTCL LR	WDTCL R	OSTCL R	NMICL R
0	0	0	0	0	0	0	0

Value after reset:

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	WDTCLR	WDT Clear	This bit is read as 0. Writing 1 to this bit clears the NMISR.WDTST flag. Writing 0 to this bit has no effect.	R/(W) *1
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.

**WDTCLR Bit (WDT Clear)**

Writing 1 to the WDTCLR bit clears the NMISR.WDTST 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.

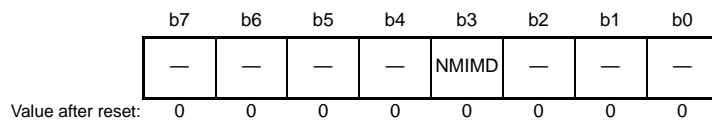
**LVD2CLR Bit (LVD2 Clear)**

Writing 1 to the LVD2CLR bit clears the NMISR.LVD2ST flag.

This bit is read as 0.

**15.2.16 NMI Pin Interrupt Control Register (NMICR)**

Address(es): 0008 7583h

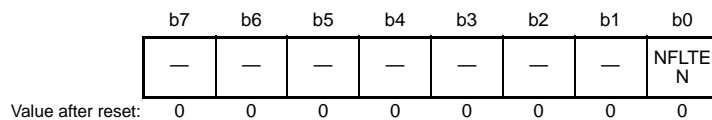


Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	NMIMD	NMI Detection Set	0: Falling edge 1: Rising edge	R/W
b7 to b4	—	Reserved	These bits are read as 0 and cannot be modified.	R/W

Change the setting of the NMICR register before the NMI pin interrupt is enabled (before setting the NMIER.NMIEN bit to 1).

**15.2.17 NMI Pin Digital Filter Enable Register (NMIFLTE)**

Address(es): 0008 7590h



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.

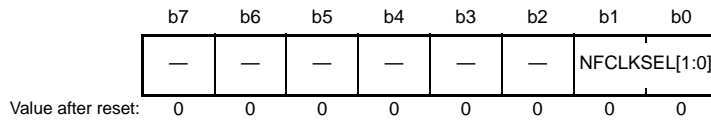
The digital filter is enabled when the NFLTEN bit is 1, and disabled when the NFLTEN bit is 0.

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 15.5.6, Digital Filter.

### 15.2.18 NMI Pin Digital Filter Setting Register (NMIFLTC)

Address(es): 0008 7594h



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), PCKL/8 (once every eight cycles), PCKL/32 (once every 32 cycles), and PCKL/64 (once every 64 cycles).

For details of the digital filter, see section 15.5.6, Digital Filter.

### 15.2.19 Group m Interrupt Source Register (GRPm) (m: group number)

- GRP00, GRP01, GRP02

Address(es): GRP00 0008 C300h, GRP01 0008 C304h, GRP02 0008 C308h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	IS2	IS1	IS0
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	IS0	Interrupt Status Flag 0	0: Interrupt is not requested. 1: Interrupt is requested.	R
b1	IS1	Interrupt Status Flag 1		R
b2	IS2	Interrupt Status Flag 2		R
b31 to b3	—	Reserved	These bits are read as 0 and cannot be modified.	R

- GRP03, GRP04, GRP05, GRP06

Address(es): GRP03 0008 C30Ch, GRP04 0008 C310h, GRP05 0008 C314h, GRP06 0008 C318h

	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
	—	—	—	—	—	—	—	—	—	—	—	IS4	IS3	IS2	IS1	IS0
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	IS0	Interrupt Status Flag 0	0: Interrupt is not requested. 1: Interrupt is requested.	R
b1	IS1	Interrupt Status Flag 1		R
b2	IS2	Interrupt Status Flag 2		R
b3	IS3	Interrupt Status Flag 3		R
b4	IS4	Interrupt Status Flag 4		R
b31 to b5	—	Reserved	These bits are read as 0 and cannot be modified.	R

- GRP12

Address(es): GRP12 0008 C330h

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
IS15	IS14	IS13	IS12	IS11	IS10	IS9	IS8	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
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	IS0	Interrupt Status Flag 0	0: Interrupt is not requested. 1: Interrupt is requested.	R
b1	IS1	Interrupt Status Flag 1		R
b2	IS2	Interrupt Status Flag 2		R
b3	IS3	Interrupt Status Flag 3		R
b4	IS4	Interrupt Status Flag 4		R
b5	IS5	Interrupt Status Flag 5		R
b6	IS6	Interrupt Status Flag 6		R
b7	IS7	Interrupt Status Flag 7		R
b8	IS8	Interrupt Status Flag 8		R
b9	IS9	Interrupt Status Flag 9		R
b10	IS10	Interrupt Status Flag 10		R
b11	IS11	Interrupt Status Flag 11		R
b12	IS12	Interrupt Status Flag 12		R
b13	IS13	Interrupt Status Flag 13		R
b14	IS14	Interrupt Status Flag 14		R
b15	IS15	Interrupt Status Flag 15		R
b31 to b16	—	Reserved	These bits are read as 0 and cannot be modified.	R

GRP<sub>m</sub> is provided for each group, where m indicates the group number (m: 00 to 06 or 12).

For details of the grouping function, see section 15.4.1, **Interrupt Request Groups**. For correspondence between the interrupt request allocated to each group and the bit number in the GRP<sub>m</sub> register, see Table 15.4, **Group m Interrupt Requests**. For correspondence between grouped interrupt sources and interrupt vector numbers, see Table 15.3, **Interrupt Vector Table**.

### IS<sub>j</sub> Flag (Interrupt Status Flag j) (j = bit number)

This bit is the status flag of the peripheral module interrupt request allocated to the j<sup>th</sup> bit in group m. This flag is set to 1 when the interrupt request is generated and is enabled by the interrupt request enable bit j in group m interrupt enable register (GEN<sub>m</sub>.EN<sub>j</sub>; m = group number and j = bit number).

When both the GEN<sub>m</sub>.EN<sub>j</sub> bit and IS<sub>j</sub> flag bit for any of the interrupt sources in a group are 1, the interrupt status flag in interrupt request register n (IR<sub>n</sub>.IR; n = interrupt vector number) corresponding to the group is set to 1.

Groups 0 to 6 comprise edge detection interrupt requests and group 12 comprises level detection interrupt requests.

(1) Groups 0 to 6

[Setting condition]

- The IS<sub>j</sub> flag is set to 1 when the corresponding peripheral module interrupt request is generated and GEN<sub>m</sub>.EN<sub>j</sub> is 1.

[Clearing condition]

- The ISj flag is cleared to 0 by writing 1 to the interrupt source clear bit in group m interrupt clear register (GCRm.CLRj).

(2) Group 12

[Setting condition]

- The ISj flag remains 1 while the corresponding peripheral module interrupt request is being sent and GENm.ENj is 1.

[Clearing conditions]

- The ISj flag is cleared to 0 when the source of the interrupt request is cleared.
- The ISj flag is cleared to 0 when GENm.ENj is 0.

Writing to the ISj flag is disabled.

### 15.2.20 Group m Interrupt Enable Register (GENm) (m = group number)

- GEN00, GEN01, GEN02

Address(es): GEN00 0008 C340h, GEN01 0008 C344h, GEN02 0008 C348h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	EN2	EN1	EN0
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	EN0	Interrupt request enable 0	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b1	EN1	Interrupt request enable 1		R/W
b2	EN2	Interrupt request enable 2		R/W
b31 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- GEN03, GEN04, GEN05, GEN06

Address(es): GEN03 0008 C34Ch, GEN04 0008 C350h, GEN05 0008 C354h, GEN06 0008 C358h

	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
	—	—	—	—	—	—	—	—	—	—	—	EN4	EN3	EN2	EN1	EN0
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	EN0	Interrupt request enable 0	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b1	EN1	Interrupt request enable 1		R/W
b2	EN2	Interrupt request enable 2		R/W
b3	EN3	Interrupt request enable 3		R/W
b4	EN4	Interrupt request enable 4		R/W
b31 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- GEN12

Address(es): GEN12 0008 C370h

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
EN15	EN14	EN13	EN12	EN11	EN10	EN9	EN8	EN7	EN6	EN5	EN4	EN3	EN2	EN1	EN0
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	EN0	Interrupt request enable 0	0: Interrupt request is disabled. 1: Interrupt request is enabled.	R/W
b1	EN1	Interrupt request enable 1		R/W
b2	EN2	Interrupt request enable 2		R/W
b3	EN3	Interrupt request enable 3		R/W
b4	EN4	Interrupt request enable 4		R/W
b5	EN5	Interrupt request enable 5		R/W
b6	EN6	Interrupt request enable 6		R/W
b7	EN7	Interrupt request enable 7		R/W
b8	EN8	Interrupt request enable 8		R/W
b9	EN9	Interrupt request enable 9		R/W
b10	EN10	Interrupt request enable 10		R/W
b11	EN11	Interrupt request enable 11		R/W
b12	EN12	Interrupt request enable 12		R/W
b13	EN13	Interrupt request enable 13		R/W
b14	EN14	Interrupt request enable 14		R/W
b15	EN15	Interrupt request enable 15		R/W
b31 to b16	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

For details of the grouping function, see section 15.4.1, Interrupt Request Groups.

GEN $m$  is provided for each group, where  $m$  indicates the group number ( $m$ : 00 to 06 or 12).

### EN $j$ Bits (Interrupt Request Enable $j$ ) ( $j$ = bit number)

- Groups 00 to 06

When an EN $j$  bit is 1, interrupt request detection by the corresponding GRP $m$ .IS $j$  flag ( $m$  = group number and  $j$  = bit number) is enabled. When an interrupt is detected, the GRP $m$ .IS $j$  flag is set to 1.

When an EN $j$  bit is 0, interrupt request detection is disabled.

Writing 0 to the EN $j$  bit does not affect the corresponding GRP $m$ .IS $j$  flag.

- Group 12

When an EN $j$  bit is 1, interrupt request detection by the corresponding GRP $m$ .IS $j$  flag ( $m$  = group number and  $j$  = bit number) is enabled. When an interrupt is detected, the GRP $m$ .IS $j$  flag is set to 1.

When an EN $j$  bit is 0, interrupt request detection is disabled.

Writing 0 to the EN $j$  bit does not affect the corresponding GRP $m$ .IS $j$  flag.



### 15.2.21 Group m Interrupt Clear Register (GCRm) (m = group number)

- GCR00, GCR01, GCR02

Address(es): GCR00 0008 C380h, GCR01 0008 C384h, GCR02 0008 C388h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	CLR2	CLR1	CLR0
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	CLR0	Interrupt source clear 0	These bits are read as 0. Writing 1 to this bit clears the interrupt status flag bit (GRPm.ISj) of the same group and bit. Writing 0 is invalid.	R(W) *1
b1	CLR1	Interrupt source clear 1		R(W) *1
b2	CLR2	Interrupt source clear 2		R(W) *1
b31 to b3	—	Reserved	These bits are read as 0 and cannot be modified.	R

Note 1. Only 1 can be written.

- GCR03, GCR04, GCR05, GCR06

Address(es): GCR03 0008 C38Ch, GCR04 0008 C390h, GCR05 0008 C394h, GCR06 0008 C398h

	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
	—	—	—	—	—	—	—	—	—	—	—	CLR4	CLR3	CLR2	CLR1	CLR0
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	CLR0	Interrupt source clear 0	Writing 1 to this bit clears the interrupt status flag bit (GRPm.ISj) of the same group and bit. Writing 0 is invalid.	R(W) *1
b1	CLR1	Interrupt source clear 1		R(W) *1
b2	CLR2	Interrupt source clear 2		R(W) *1
b3	CLR3	Interrupt source clear 3		R(W) *1
b4	CLR4	Interrupt source clear 4		R(W) *1
b31 to b5	—	Reserved	These bits are read as 0 and cannot be modified.	R/W

Note 1. Only 1 can be written.

For details of the grouping function, see section 15.4.1, Interrupt Request Groups. GCRm is provided for each group, where m indicates the group number (m: 00 to 06).

#### CLRj Bits (Interrupt Source Clear j) (j = bit number)

Writing 1 to this bit clears the corresponding GRPm.ISj flag (m = group number and j = bit number) to 0.

## 15.2.22 Unit Selecting Register (SEL)

Address(es): 0008 C3C0h

	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
	—	—	—	—	—	—	—	—	—	—	CN5	CN4	CN3	CN2	CN1	CN0
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	CN0	Unit Select 0	0: MTU0 interrupt requests are selected. 1: TPU6 interrupt requests are selected.	R/W
b1	CN1	Unit Select 1	0: MTU1 interrupt requests are selected. 1: TPU7 interrupt requests are selected.	R/W
b2	CN2	Unit Select 2	0: MTU2 interrupt requests are selected. 1: TPU8 interrupt requests are selected.	R/W
b3	CN3	Unit Select 3	0: MTU3 interrupt requests are selected. 1: TPU9 interrupt requests are selected.	R/W
b4	CN4	Unit Select 4	0: MTU4 interrupt requests are selected. 1: TPU10 interrupt requests are selected.	R/W
b5	CN5	Unit Select 5	0: MTU5 interrupt requests are selected. 1: TPU11 interrupt requests are selected.	R/W
b6	—	Reserved	This bit is read as 0 and cannot be modified.	R

For details of the unit selection function and MTUn (n = 0 to 5) and TPU<sub>n</sub> (n = 6 to 11) interrupt sources, see section 15.4.2, Unit Selection.

### CN<sub>j</sub> Bits (Unit Select j) (j = 0 to 5)

When a CN<sub>j</sub> bit is 0, the MTUn (n = 0 to 5) is selected. When a CN<sub>j</sub> bit is 1, the TPU<sub>n</sub> (n = 6 to 11) is selected. Set the CN<sub>j</sub> bits before generating an interrupt request from the peripheral module of the selected unit; otherwise, operation cannot be guaranteed.

## 15.3 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 four-byte vector address from the vector table.

### 15.3.1 Interrupt Vector Table

The interrupt vector table is placed in the 1024-byte range (4 bytes x 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 15.3, 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 15.3 lists details of the interrupt vectors. Details of the headings in Table 15.3 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.
Form of interrupt detection	"Edge" or "level" as the method for detection of the interrupt
CPU interrupt	"o" in this column indicates usability as a CPU interrupt.
DTC activation	"o" in this column indicates usability as a request for DTC activation.
DMAC activation	"o" in this column indicates usability as a request for DMAC activation.
sstb return	"o" in this column indicates usability as a request for return from software-standby mode.
sacs return	"o" in this column indicates usability as a request for return from all-module clock-stop mode.
IER	Name of the interrupt request enable register (IER) and bit corresponding to the vector number
IPR	Name of the interrupt source priority register (IPR) corresponding to the interrupt source
DTCER	Name of the DTC activation enable register (DTCER) corresponding to the DTC activation source

**Table 15.3 Interrupt Vector Table (1/6)**

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU Interrupt	DTC Activation	DMAC Activation	ssth Return	sacs Return	IER	IPR	DTCER
—	For an unconditional trap	0	0000h	—	x	x	x	x	x	—	—	—
—	For an unconditional trap	1	0004h	—	x	x	x	x	x	—	—	—
—	For an unconditional trap	2	0008h	—	x	x	x	x	x	—	—	—
—	For an unconditional trap	3	000Ch	—	x	x	x	x	x	—	—	—
—	For an unconditional trap	4	0010h	—	x	x	x	x	x	—	—	—
—	For an unconditional trap	5	0014h	—	x	x	x	x	x	—	—	—
—	For an unconditional trap	6	0018h	—	x	x	x	x	x	—	—	—
—	For an unconditional trap	7	001Ch	—	x	x	x	x	x	—	—	—
—	For an unconditional trap	8	0020h	—	x	x	x	x	x	—	—	—
—	For an unconditional trap	9	0024h	—	x	x	x	x	x	—	—	—
—	For an unconditional trap	10	0028h	—	x	x	x	x	x	—	—	—
—	For an unconditional trap	11	002Ch	—	x	x	x	x	x	—	—	—
—	For an unconditional trap	12	0030h	—	x	x	x	x	x	—	—	—
—	For an unconditional trap	13	0034h	—	x	x	x	x	x	—	—	—
—	For an unconditional trap	14	0038h	—	x	x	x	x	x	—	—	—
—	For an unconditional trap	15	003Ch	—	x	x	x	x	x	—	—	—
BSC	BUSERR	16	0040h	Level	o	x	x	x	x	IER02.IEN0	IPR000	—
—	Reserved	17	0044h	—	x	x	x	x	x	—	—	—
—	Reserved	18	0048h	—	x	x	x	x	x	—	—	—
—	Reserved	19	004Ch	—	x	x	x	x	x	—	—	—
—	Reserved	20	0050h	—	x	x	x	x	x	—	—	—
FCU	FIFERR	21	0054h	Level	o	x	x	x	x	IER02.IEN5	IPR001	—
—	Reserved	22	0058h	—	x	x	x	x	x	—	—	—
FCU	FRDYI	23	005Ch	Edge	o	x	x	x	x	IER02.IEN7	IPR002	—
—	Reserved	24	0060h	—	x	x	x	x	x	—	—	—
—	Reserved	25	0064h	—	x	x	x	x	x	—	—	—
—	Reserved	26	0068h	—	x	x	x	x	x	—	—	—
ICU	SWINT	27	006Ch	Edge	o	o	x	x	x	IER03.IEN3	IPR003	DTCER027
CMT0	CMI0	28	0070h	Edge	o	o	o	x	x	IER03.IEN4	IPR004	DTCER028
CMT1	CMI1	29	0074h	Edge	o	o	o	x	x	IER03.IEN5	IPR005	DTCER029
CMT2	CMI2	30	0078h	Edge	o	o	o	x	x	IER03.IEN6	IPR006	DTCER030
CMT3	CMI3	31	007Ch	Edge	o	o	o	x	x	IER03.IEN7	IPR007	DTCER031
—	Reserved	32	0080h	—	x	x	x	x	x	—	—	—
USB0	D0FIFO0	33	0084h	Edge	o	o	o	x	x	IER04.IEN1	IPR033	DTCER033
	D1FIFO0	34	0088h	Edge	o	o	o	x	x	IER04.IEN2	IPR034	DTCER034
	USBIO	35	008Ch	Edge	o	x	x	x	x	IER04.IEN3	IPR035	—
—	Reserved	36	0090h	—	x	x	x	x	x	—	—	—
—	Reserved	37	0094h	—	x	x	x	x	x	—	—	—
—	Reserved	38	0098h	—	x	x	x	x	x	—	—	—
RSPI0	SPRI0	39	009Ch	Edge	o	o	o	x	x	IER04.IEN7	IPR039	DTCER039
	SPTI0	40	00A0h	Edge	o	o	o	x	x	IER05.IEN0		DTCER040
	SPII0	41	00A4h	Level	o	x	x	x	x	IER05.IEN1		—
RSPI1	SPRI1	42	00A8h	Edge	o	o	o	x	x	IER05.IEN2	IPR042	DTCER042
	SPTI1	43	00ACh	Edge	o	o	o	x	x	IER05.IEN3		DTCER043
	SPII1	44	00B0h	Level	o	x	x	x	x	IER05.IEN4		—

Table 15.3 Interrupt Vector Table (2/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU Interrupt	DTC Activation	DMAC Activation	sstb Return	sacs Return	IER	IPR	DTCER
RSPI2	SPRI2	45	00B4h	Edge	○	○	○	×	×	IER05.IEN5	IPR045	DTCER045
	SPTI2	46	00B8h	Edge	○	○	○	×	×	IER05.IEN6		DTCER046
	SPII2	47	00BCh	Level	○	×	×	×	×	IER05.IEN7		—
CAN0	RXF0	48	00C0h	Edge	○	×	×	×	×	IER06.IEN0	IPR048	—
	TXF0	49	00C4h	Edge	○	×	×	×	×	IER06.IEN1		—
	RXM0	50	00C8h	Edge	○	×	×	×	×	IER06.IEN2		—
	TXM0	51	00CCh	Edge	○	×	×	×	×	IER06.IEN3		—
CAN1	RXF1	52	00D0h	Edge	○	×	×	×	×	IER06.IEN4	IPR052	—
	TXF1	53	00D4h	Edge	○	×	×	×	×	IER06.IEN5		—
	RXM1	54	00D8h	Edge	○	×	×	×	×	IER06.IEN6		—
	TXM1	55	00DCh	Edge	○	×	×	×	×	IER06.IEN7		—
CAN2	RXF2	56	00E0h	Edge	○	×	×	×	×	IER07.IEN0	IPR056	—
	TXF2	57	00E4h	Edge	○	×	×	×	×	IER07.IEN1		—
	RXM2	58	00E8h	Edge	○	×	×	×	×	IER07.IEN2		—
	TXM2	59	00ECh	Edge	○	×	×	×	×	IER07.IEN3		—
—	Reserved	60	00F0h	—	×	×	×	×	×	—	—	—
—	Reserved	61	00F4h	—	×	×	×	×	×	—	—	—
RTC	CUP	62	00F8h	Edge	○	×	×	×	×	IER07.IEN6	IPR062	—
—	Reserved	63	00FCh	—	×	×	×	×	×	—	—	—
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
	IRQ6	70	0118h	Edge/Level	○	○	×	○	○	IER08.IEN6	IPR070	DTCER070
	IRQ7	71	011Ch	Edge/Level	○	○	×	○	○	IER08.IEN7	IPR071	DTCER071
	IRQ8	72	0120h	Edge/Level	○	○	×	○	○	IER09.IEN0	IPR072	DTCER072
	IRQ9	73	0124h	Edge/Level	○	○	×	○	○	IER09.IEN1	IPR073	DTCER073
	IRQ10	74	0128h	Edge/Level	○	○	×	○	○	IER09.IEN2	IPR074	DTCER074
	IRQ11	75	012Ch	Edge/Level	○	○	×	○	○	IER09.IEN3	IPR075	DTCER075
	IRQ12	76	0130h	Edge/Level	○	○	×	○	○	IER09.IEN4	IPR076	DTCER076
	IRQ13	77	0134h	Edge/Level	○	○	×	○	○	IER09.IEN5	IPR077	DTCER077
	IRQ14	78	0138h	Edge/Level	○	○	×	○	○	IER09.IEN6	IPR078	DTCER078
	IRQ15	79	013Ch	Edge/Level	○	○	×	○	○	IER09.IEN7	IPR079	DTCER079
—	Reserved	80	0140h	—	×	×	×	×	×	—	—	—
—	Reserved	81	0144h	—	×	×	×	×	×	—	—	—
—	Reserved	82	0148h	—	×	×	×	×	×	—	—	—
—	Reserved	83	014Ch	—	×	×	×	×	×	—	—	—
—	Reserved	84	0150h	—	×	×	×	×	×	—	—	—
—	Reserved	85	0154h	—	×	×	×	×	×	—	—	—
—	Reserved	86	0158h	—	×	×	×	×	×	—	—	—
—	Reserved	87	015Ch	—	×	×	×	×	×	—	—	—
—	Reserved	88	0160h	—	×	×	×	×	×	—	—	—
—	Reserved	89	0164h	—	×	×	×	×	×	—	—	—
USB	USB0	90	0168h	Level	○	×	×	○	○	IER0B.IEN2	IPR090	—

Table 15.3 Interrupt Vector Table (3/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU Interrupt	DTC Activation	DMAC Activation	sstb Return	sacs Return	IER	IPR	DTCER
—	Reserved	91	016Ch	—	x	x	x	x	x	—	—	—
RTC	ALM	92	0170h	Edge	o	x	x	o	o	IER0B.IEN4	IPR092	—
	PRD	93	0174h	Edge	o	x	x	o	o	IER0B.IEN5	IPR093	—
—	Reserved	94	0178h	—	x	x	x	x	x	—	—	—
—	Reserved	95	017Ch	—	x	x	x	x	x	—	—	—
—	Reserved	96	0180h	—	x	x	x	x	x	—	—	—
—	Reserved	97	0184h	—	x	x	x	x	x	—	—	—
AD	ADIO	98	0188h	Edge	o	o	o	x	x	IER0C.IEN2	IPR098	DTCER098
—	Reserved	99	018Ch	—	x	x	x	x	x	—	—	—
—	Reserved	100	0190h	—	x	x	x	x	x	—	—	—
—	Reserved	101	0194h	—	x	x	x	x	x	—	—	—
S12AD0	S12ADI	102	0198h	Edge	o	o	o	x	x	IER0C.IEN6	IPR102	DTCER102
—	Reserved	103	019Ch	—	x	x	x	x	x	—	—	—
—	Reserved	104	01A0h	—	x	x	x	x	x	—	—	—
—	Reserved	105	01A4h	—	x	x	x	x	x	—	—	—
ICU*2	GROUP0	106	01A8h	Level	o	x	x	x	x	IER0D.IEN2	IPR106	—
	GROUP1	107	01ACh	Level	o	x	x	x	x	IER0D.IEN3	IPR107	—
	GROUP2	108	01B0h	Level	o	x	x	x	x	IER0D.IEN4	IPR108	—
	GROUP3	109	01B4h	Level	o	x	x	x	x	IER0D.IEN5	IPR109	—
	GROUP4	110	01B8h	Level	o	x	x	x	x	IER0D.IEN6	IPR110	—
	GROUP5	111	01BCh	Level	o	x	x	x	x	IER0D.IEN7	IPR111	—
	GROUP6	112	01C0h	Level	o	x	x	x	x	IER0E.IEN0	IPR112	—
—	Reserved	113	01C4h	—	x	x	x	x	x	—	—	—
ICU	GROUP12	114	01C8h	Level	o	x	x	x	x	IER0E.IEN2	IPR114	—
—	Reserved	115	01CCh	—	x	x	x	x	x	—	—	—
—	Reserved	116	01D0h	—	x	x	x	x	x	—	—	—
—	Reserved	117	01D4h	—	x	x	x	x	x	—	—	—
—	Reserved	118	01D8h	—	x	x	x	x	x	—	—	—
—	Reserved	119	01DCh	—	x	x	x	x	x	—	—	—
—	Reserved	120	01E0h	—	x	x	x	x	x	—	—	—
—	Reserved	121	01E4h	—	x	x	x	x	x	—	—	—
SCI12	SCIX0	122	01E8h	Level	o	x	x	x	x	IER0F.IEN2	IPR122	—
	SCIX1	123	01ECh	Level	o	x	x	x	x	IER0F.IEN3		—
	SCIX2	124	01F0h	Level	o	x	x	x	x	IER0F.IEN4		—
	SCIX3	125	01F4h	Level	o	x	x	x	x	IER0F.IEN5		—
TPU0	TGI0A	126	01F8h	Edge	o	o	o	x	x	IER0F.IEN6	IPR126	DTCER126
	TGI0B	127	01FCh	Edge	o	o	x	x	x	IER0F.IEN7		DTCER127
	TGI0C	128	0200h	Edge	o	o	x	x	x	IER10.IEN0		DTCER128
	TGI0D	129	0204h	Edge	o	o	x	x	x	IER10.IEN1		DTCER129
TPU1	TGI1A	130	0208h	Edge	o	o	o	x	x	IER10.IEN2	IPR130	DTCER130
	TGI1B	131	020Ch	Edge	o	o	x	x	x	IER10.IEN3		DTCER131
TPU2	TGI2A	132	0210h	Edge	o	o	o	x	x	IER10.IEN4	IPR132	DTCER132
	TGI2B	133	0214h	Edge	o	o	x	x	x	IER10.IEN5		DTCER133

Table 15.3 Interrupt Vector Table (4/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU Interrupt	DTC Activation	DMAC Activation	sstb Return	sacs Return	IER	IPR	DTCER
TPU3	TGI3A	134	0218h	Edge	○	○	○	×	×	IER10.IEN6	IPR134	DTCER134
	TGI3B	135	021Ch	Edge	○	○	×	×	×	IER10.IEN7		DTCER135
	TGI3C	136	0220h	Edge	○	○	×	×	×	IER11.IEN0		DTCER136
	TGI3D	137	0224h	Edge	○	○	×	×	×	IER11.IEN1		DTCER137
TPU4	TGI4A	138	0228h	Edge	○	○	○	×	×	IER11.IEN2	IPR138	DTCER138
	TGI4B	139	022Ch	Edge	○	○	×	×	×	IER11.IEN3		DTCER139
TPU5	TGI5A	140	0230h	Edge	○	○	○	×	×	IER11.IEN4	IPR140	DTCER140
	TGI5B	141	0234h	Edge	○	○	×	×	×	IER11.IEN5		DTCER141
TPU6/MTU0	TGI6A(TPU6)/ TGIA0(MTU0)	142	0238h	Edge	○	○	○	×	×	IER11.IEN6	IPR142	DTCER142
	TGI6B(TPU6)/ TGIB0(MTU0)	143	023Ch	Edge	○	○	×	×	×	IER11.IEN7		DTCER143
	TGI6C(TPU6)/ TGIC0(MTU0)	144	0240h	Edge	○	○	×	×	×	IER12.IEN0		DTCER144
	TGI6D(TPU6)/ TGID0(MTU0)	145	0244h	Edge	○	○	×	×	×	IER12.IEN1		DTCER145
	TGIE0(MTU0)	146	0248h	Edge	○	×	×	×	×	IER12.IEN2	IPR146	—
	TGIF0(MTU0)	147	024Ch	Edge	○	×	×	×	×	IER12.IEN3		—
TPU7/MTU1	TGI7A(TPU7)/ TGIA1(MTU1)	148	0250h	Edge	○	○	○	×	×	IER12.IEN4	IPR148	DTCER148
	TGI7B(TPU7)/ TGIB1(MTU1)	149	0254h	Edge	○	○	×	×	×	IER12.IEN5		DTCER149
TPU8/MTU2	TGI8A(TPU8)/ TGIA2(MTU2)	150	0258h	Edge	○	○	○	×	×	IER12.IEN6	IPR150	DTCER150
	TGI8B(TPU8)/ TGIB2(MTU2)	151	025Ch	Edge	○	○	×	×	×	IER12.IEN7		DTCER151
TPU9/MTU3	TGI9A(TPU9)/ TGIA3(MTU3)	152	0260h	Edge	○	○	○	×	×	IER13.IEN0	IPR152	DTCER152
	TGI9B(TPU9)/ TGIB3(MTU3)	153	0264h	Edge	○	○	×	×	×	IER13.IEN1		DTCER153
	TGI9C(TPU9)/ TGIC3(MTU3)	154	0268h	Edge	○	○	×	×	×	IER13.IEN2		DTCER154
	TGI9D(TPU9)/ TGID3(MTU3)	155	026Ch	Edge	○	○	×	×	×	IER13.IEN3		DTCER155
TPU10/MTU4	TGI10A(TPU10)/ TGIA4(MTU4)	156	0270h	Edge	○	○	○	×	×	IER13.IEN4	IPR156	DTCER156
	TGI10B(TPU10)/ TGIB4(MTU4)	157	0274h	Edge	○	○	×	×	×	IER13.IEN5		DTCER157
	TGIC4(MTU4)	158	0278h	Edge	○	○	×	×	×	IER13.IEN6		DTCER158
	TGID4(MTU4)	159	027Ch	Edge	○	○	×	×	×	IER13.IEN7		DTCER159
	TCIV4(MTU4)	160	0280h	Edge	○	○	×	×	×	IER14.IEN0	IPR160	DTCER160
TPU11/MTU5	TGIU5(MTU5)	161	0284h	Edge	○	○	×	×	×	IER14.IEN1	IPR161	DTCER161
	TGIV5(MTU5)	162	0288h	Edge	○	○	×	×	×	IER14.IEN2		DTCER162
	TGIW5(MTU5)	163	028Ch	Edge	○	○	×	×	×	IER14.IEN3		DTCER163
	TGI11A(TPU11)	164	0290h	Edge	○	○	○	×	×	IER14.IEN4	IPR164	DTCER164
	TGI11B(TPU11)	165	0294h	Edge	○	○	×	×	×	IER14.IEN5		DTCER165
POE	OEI1	166	0298h	Level	○	×	×	×	×	IER14.IEN6	IPR166	—
	OEI2	167	029Ch	Level	○	×	×	×	×	IER14.IEN7		—
—	Reserved	168	02A0h	—	×	×	×	×	×	—	—	—
—	Reserved	169	02A4h	—	×	×	×	×	×	—	—	—
TMR0	CMIA0	170	02A8h	Edge	○	○	×	×	○*3	IER15.IEN2	IPR170	DTCER170
	CMIB0	171	02ACh	Edge	○	○	×	×	○*3	IER15.IEN3		DTCER171
	OVI0	172	02B0h	Edge	○	×	×	×	○*3	IER15.IEN4		—

Table 15.3 Interrupt Vector Table (5/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU Interrupt	DTC Activation	DMAC Activation	sstb Return	sacs Return	IER	IPR	DTCER
TMR1	CMIA1	173	02B4h	Edge	o	o	x	x	o*3	IER15.IEN5	IPR173	DTCER173
	CMIB1	174	02B8h	Edge	o	o	x	x	o*3	IER15.IEN6		DTCER174
	OVI1	175	02BCh	Edge	o	x	x	x	o*3	IER15.IEN7		—
TMR2	CMIA2	176	02C0h	Edge	o	o	x	x	o*3	IER16.IEN0	IPR176	DTCER176
	CMIB2	177	02C4h	Edge	o	o	x	x	o*3	IER16.IEN1		DTCER177
	OVI2	178	02C8h	Edge	o	x	x	x	o*3	IER16.IEN2		—
TMR3	CMIA3	179	02CCh	Edge	o	o	x	x	o*3	IER16.IEN3	IPR179	DTCER179
	CMIB3	180	02D0h	Edge	o	o	x	x	o*3	IER16.IEN4		DTCER180
	OVI3	181	02D4h	Edge	o	x	x	x	o*3	IER16.IEN5		—
RIIC0	EEI0	182	02D8h	Level	o	x	x	x	x	IER16.IEN6	IPR182	—
	RXI0	183	02DCh	Edge	o	o	o	x	x	IER16.IEN7	IPR183	DTCER183
	TXI0	184	02E0h	Edge	o	o	o	x	x	IER17.IEN0	IPR184	DTCER184
	TEI0	185	02E4h	Level	o	x	x	x	x	IER17.IEN1	IPR185	—
RIIC1	EEI1	186	02E8h	Level	o	x	x	x	x	IER17.IEN2	IPR186	—
	RXI1	187	02ECh	Edge	o	o	o	x	x	IER17.IEN3	IPR187	DTCER187
	TXI1	188	02F0h	Edge	o	o	o	x	x	IER17.IEN4	IPR188	DTCER188
	TEI1	189	02F4h	Level	o	x	x	x	x	IER17.IEN5	IPR189	—
RIIC2	EEI2	190	02F8h	Level	o	x	x	x	x	IER17.IEN6	IPR190	—
	RXI2	191	02FCh	Edge	o	o	o	x	x	IER17.IEN7	IPR191	DTCER191
	TXI2	192	0300h	Edge	o	o	o	x	x	IER18.IEN0	IPR192	DTCER192
	TEI2	193	0304h	Level	o	x	x	x	x	IER18.IEN1	IPR193	—
RIIC3	EEI3	194	0308h	Level	o	x	x	x	x	IER18.IEN2	IPR194	—
	RXI3	195	030Ch	Edge	o	o	o	x	x	IER18.IEN3	IPR195	DTCER195
	TXI3	196	0310h	Edge	o	o	o	x	x	IER18.IEN4	IPR196	DTCER196
	TEI3	197	0314h	Level	o	x	x	x	x	IER18.IEN5	IPR197	—
DMAC	DMAC0I	198	0318h	Edge	o	o	x	x	x	IER18.IEN6	IPR198	DTCER198
	DMAC1I	199	031Ch	Edge	o	o	x	x	x	IER18.IEN7	IPR199	DTCER199
	DMAC2I	200	0320h	Edge	o	o	x	x	x	IER19.IEN0	IPR200	DTCER200
	DMAC3I	201	0324h	Edge	o	o	x	x	x	IER19.IEN1	IPR201	DTCER201
—	Reserved	202	0328h	—	x	x	x	x	x	—	—	—
—	Reserved	203	032Ch	—	x	x	x	x	x	—	—	—
—	Reserved	204	0330h	—	x	x	x	x	x	—	—	—
—	Reserved	205	0334h	—	x	x	x	x	x	—	—	—
—	Reserved	206	0338h	—	x	x	x	x	x	—	—	—
—	Reserved	207	033Ch	—	x	x	x	x	x	—	—	—
—	Reserved	208	0340h	—	x	x	x	x	x	—	—	—
—	Reserved	209	0344h	—	x	x	x	x	x	—	—	—
—	Reserved	210	0348h	—	x	x	x	x	x	—	—	—
—	Reserved	211	034Ch	—	x	x	x	x	x	—	—	—
—	Reserved	212	0350h	—	x	x	x	x	x	—	—	—
—	Reserved	213	0354h	—	x	x	x	x	x	—	—	—
SCI0	RXI0	214	0358h	Edge	o	o	o	x	x	IER1A.IEN6	IPR214	DTCER214
	TXI0	215	035Ch	Edge	o	o	o	x	x	IER1A.IEN7		DTCER215
	TEI0	216	0360h	Level	o	x	x	x	x	IER1B.IEN0		—



Table 15.3 Interrupt Vector Table (6/6)

Source of Interrupt Request Generation	Name	Vector No.*1	Vector Address Offset	Form of Interrupt Detection	CPU Interrupt	DTC Activation	DMAC Activation	sstb Return	sacs Return	IER	IPR	DTCER
SCI1	RXI1	217	0364h	Edge	○	○	○	×	×	IER1B.IEN1	IPR217	DTCER217
	TXI1	218	0368h	Edge	○	○	○	×	×	IER1B.IEN2		DTCER218
	TEI1	219	036Ch	Level	○	×	×	×	×	IER1B.IEN3		—
SCI2	RXI2	220	0370h	Edge	○	○	○	×	×	IER1B.IEN4	IPR220	DTCER220
	TXI2	221	0374h	Edge	○	○	○	×	×	IER1B.IEN5		DTCER221
	TEI2	222	0378h	Level	○	×	×	×	×	IER1B.IEN6		—
SCI3	RXI3	223	037Ch	Edge	○	○	○	×	×	IER1B.IEN7	IPR223	DTCER223
	TXI3	224	0380h	Edge	○	○	○	×	×	IER1C.IEN0		DTCER224
	TEI3	225	0384h	Level	○	×	×	×	×	IER1C.IEN1		—
SCI4	RXI4	226	0388h	Edge	○	○	○	×	×	IER1C.IEN2	IPR226	DTCER226
	TXI4	227	038Ch	Edge	○	○	○	×	×	IER1C.IEN3		DTCER227
	TEI4	228	0390h	Level	○	×	×	×	×	IER1C.IEN4		—
SCI5	RXI5	229	0394h	Edge	○	○	○	×	×	IER1C.IEN5	IPR229	DTCER229
	TXI5	230	0398h	Edge	○	○	○	×	×	IER1C.IEN6		DTCER230
	TEI5	231	039Ch	Level	○	×	×	×	×	IER1C.IEN7		—
SCI6	RXI6	232	03A0h	Edge	○	○	○	×	×	IER1D.IEN0	IPR232	DTCER232
	TXI6	233	03A4h	Edge	○	○	○	×	×	IER1D.IEN1		DTCER233
	TEI6	234	03A8h	Level	○	×	×	×	×	IER1D.IEN2		—
SCI7	RXI7	235	03ACh	Edge	○	○	○	×	×	IER1D.IEN3	IPR235	DTCER235
	TXI7	236	03B0h	Edge	○	○	○	×	×	IER1D.IEN4		DTCER236
	TEI7	237	03B4h	Level	○	×	×	×	×	IER1D.IEN5		—
SCI8	RXI8	238	03B8h	Edge	○	○	○	×	×	IER1D.IEN6	IPR238	DTCER238
	TXI8	239	03BCh	Edge	○	○	○	×	×	IER1D.IEN7		DTCER239
	TEI8	240	03C0h	Level	○	×	×	×	×	IER1E.IEN0		—
SCI9	RXI9	241	03C4h	Edge	○	○	○	×	×	IER1E.IEN1	IPR241	DTCER241
	TXI9	242	03C8h	Edge	○	○	○	×	×	IER1E.IEN2		DTCER242
	TEI9	243	03CCh	Level	○	×	×	×	×	IER1E.IEN3		—
SCI10	RXI10	244	03D0h	Edge	○	○	○	×	×	IER1E.IEN4	IPR244	DTCER244
	TXI10	245	03D4h	Edge	○	○	○	×	×	IER1E.IEN5		DTCER245
	TEI10	246	03D8h	Level	○	×	×	×	×	IER1E.IEN6		—
SCI11	RXI11	247	03DCh	Edge	○	○	○	×	×	IER1E.IEN7	IPR247	DTCER247
	TXI11	248	03E0h	Edge	○	○	○	×	×	IER1F.IEN0		DTCER248
	TEI11	249	03E4h	Level	○	×	×	×	×	IER1F.IEN1		—
SCI12	RXI12	250	03E8h	Edge	○	○	○	×	×	IER1F.IEN2	IPR250	DTCER250
	TXI12	251	03ECh	Edge	○	○	○	×	×	IER1F.IEN3		DTCER251
	TEI12	252	03F0h	Level	○	×	×	×	×	IER1F.IEN4		—
IEB	IEBINT	253	03F4h	Level	○	×	×	×	×	IER1F.IEN5	IPR253	—
—	Reserved	254	03F8h	—	×	×	×	×	×	—	—	—
—	Reserved	255	03FCh	—	×	×	×	×	×	—	—	—

Note: • This table lists the interrupt vectors for the maximum specification. The interrupt vectors for individual products correspond to the functions listed in Table 1.2. For details, see Table 1.2, Comparison of Functions for Different Packages.

Note 1. An interrupt source with a smaller vector number takes precedence.

Note 2. For the allocation of interrupt requests to the various interrupt groups, see Table 15.4, Group m Interrupt Requests (1/2).

Note 3. When the setting is set so that the ICLK is slower than the PCLKB, the TMR interrupt cannot be used for canceling all-module clock stop mode.

### 15.3.2 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.

### 15.3.3 Non-maskable Interrupt Vector Table

The non-maskable interrupt vector table is at FFFF FFF8h.

## 15.4 Peripheral Module Interrupt Request Groups and Unit Selection

### 15.4.1 Interrupt Request Groups

A maximum of 32 interrupt signals from peripheral modules are divided into groups and the interrupt signals in each group can be handled collectively as a single interrupt request.

Groups 0 to 6 comprise edge detection interrupt requests and group 12 comprises level detection interrupt requests.

When an interrupt request is detected and GENm.ENj bit ( $m = \text{group number}$  and  $j = \text{bit number}$ ) is 1, the GRPm.ISj flag is set to 1.

When a GENm.ENj flag in group  $m$  is 1, the IRn.IR flag ( $n = \text{interrupt vector number}$ ) corresponding to the group is set to 1.

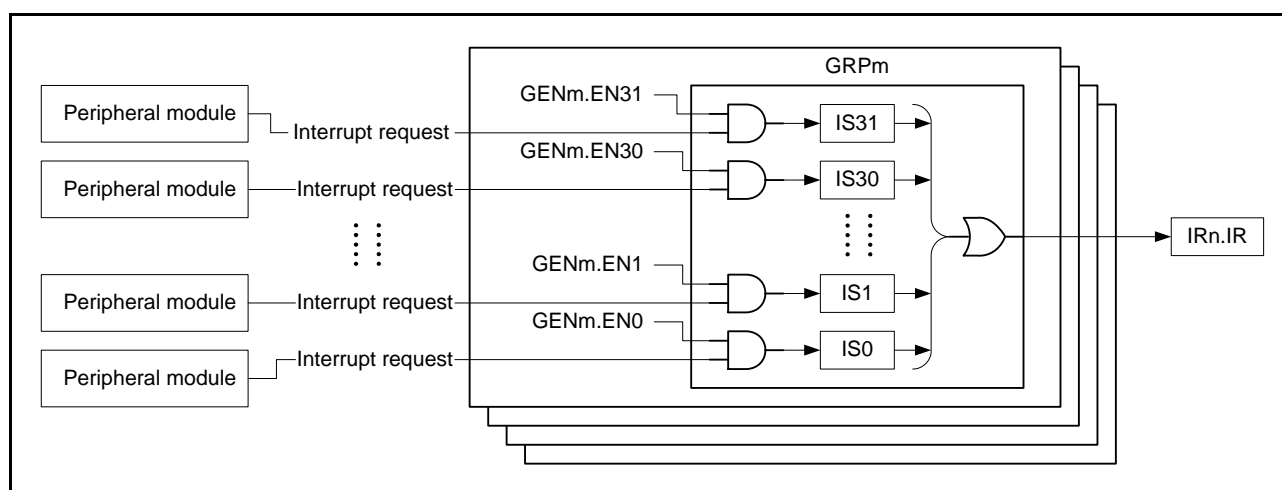


Figure 15.2 Interrupt Request Grouping

#### (1) Edge Detection Interrupt Requests

Groups 0 to 6 comprise edge detection interrupt requests.

When an interrupt request is detected and GENm.ENj bit ( $m = \text{group number}$  and  $j = \text{bit number}$ ) is 1, the GRPm.ISj flag bit is set to 1. The interrupt request is not detected when GENm.ENj bit is 0.

The GRPm.ISj flag is cleared to 0 by writing 1 to the GCRm.CLRj bit.

The GRPm.ISj flag remains 1 if the GENm.ENj bit is cleared to 0 when the GRPm.ISj flag is 1.

For operation examples, see section 15.5.1.3, Edge Detection Group Interrupts and Interrupt Status Flags.

#### (2) Level Detection Interrupt Requests

Group 12 comprises level detection interrupt requests.

When an interrupt request is detected and GENm.ENj bit ( $m = \text{group number}$  and  $j = \text{bit number}$ ) is 1, the GRPm.ISj flag bit is set to 1. When the interrupt request is detected as 0, the GRPm.ISj flag is also cleared to 0.

When the GENm.ENj bit is 0, the interrupt request is not detected and the GRPm.ISj flag is cleared to 0.

For operation examples, see section 15.5.1.4, Level Detection Group Interrupts and Interrupt Status Flags.

## (3) List of Interrupt Requests for Each Group

Table 15.4 lists the interrupt requests by group.

**Table 15.4 Group m Interrupt Requests (1/2)**

Group	Interrupt Request Source Interrupt Request	Name	GENm.ENj Bit	GRPm.ISj Flag	GCRm.CLRj Flag	Vector No. (IRn.IR)
Group 0	CAN0	ERS0 (error interrupt)	GEN00.EN0	GRP00.IS0	GCR00.CLR0	106
	CAN1	ERS1 (error interrupt)	GEN00.EN1	GRP00.IS1	GCR00.CLR1	
	CAN2	ERS2 (error interrupt)	GEN00.EN2	GRP00.IS2	GCR00.CLR2	
Group 1	MTU0	TCIV0 (overflow)	GEN01.EN0	GRP01.IS0	GCR01.CLR0	107
	MTU1	TCIV1 (overflow)	GEN01.EN1	GRP01.IS1	GCR01.CLR1	
		TCIU1 (underflow)	GEN01.EN2	GRP01.IS2	GCR01.CLR2	
Group 2	MTU2	TCIV2 (overflow)	GEN02.EN0	GRP02.IS0	GCR02.CLR0	108
		TCIU2 (underflow)	GEN02.EN1	GRP02.IS1	GCR02.CLR1	
	MTU3	TCIV3 (overflow)	GEN02.EN2	GRP02.IS2	GCR02.CLR2	
Group 3	TPU0	TCI0V (overflow)	GEN03.EN0	GRP03.IS0	GCR03.CLR0	109
	TPU1	TCI1V (overflow)	GEN03.EN1	GRP03.IS1	GCR03.CLR1	
		TCI1U (underflow)	GEN03.EN2	GRP03.IS2	GCR03.CLR2	
	TPU5	TCI5V (overflow)	GEN03.EN3	IGRP03.S3	GCR03.CLR3	
		TCI5U (underflow)	GEN03.EN4	IGRP03.S4	GCR03.CLR4	
Group 4	TPU2	TCI2V (overflow)	GEN04.EN0	GRP04.IS0	GCR04.CLR0	110
		TCI2U (underflow)	GEN04.EN1	GRP04.IS1	GCR04.CLR1	
	TPU3	TCI3V (overflow)	GEN04.EN2	GRP04.IS2	GCR04.CLR2	
	TPU4	TCI4V (overflow)	GEN04.EN3	GRP04.IS3	GCR04.CLR3	
		TCI4U (underflow)	GEN04.EN4	GRP04.IS4	GCR04.CLR4	
Group 5	TPU6	TCI6V (overflow)	GEN05.EN0	GRP05.IS0	GCR05.CLR0	111
	TPU7	TCI7V (overflow)	GEN05.EN1	GRP05.IS1	GCR05.CLR1	
		TCI7U (underflow)	GEN05.EN2	GRP05.IS2	GCR05.CLR2	
	TPU11	TCI11V (overflow)	GEN05.EN3	GRP05.IS3	GCR05.CLR3	
		TCI11U (underflow)	GEN05.EN4	GRP05.IS4	GCR05.CLR4	
Group 6	TPU8	TCI8V (overflow)	GEN06.EN0	GRP06.IS0	GCR06.CLR0	112
		TCI8U (underflow)	GEN06.EN1	GRP06.IS1	GCR06.CLR1	
	TPU9	TCI9V (overflow)	GEN06.EN2	GRP06.IS2	GCR06.CLR2	
	TPU10	TCI10V (overflow)	GEN06.EN3	GRP06.IS3	GCR06.CLR3	
		TCI10U (underflow)	GEN06.EN4	GRP06.IS4	GCR06.CLR4	

**Table 15.4 Group m Interrupt Requests (2/2)**

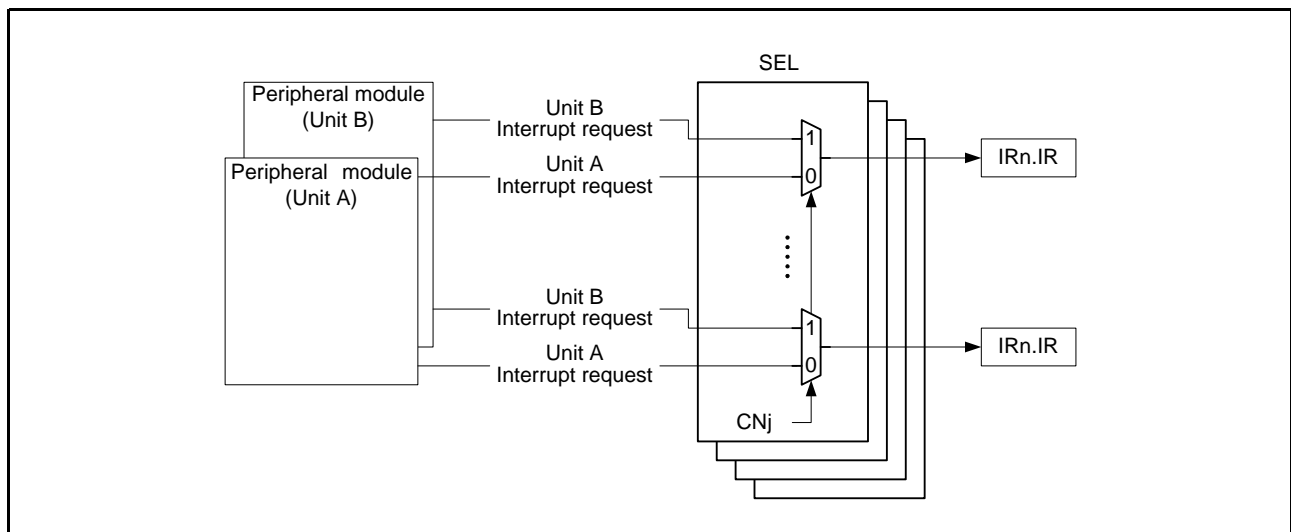
Group	Interrupt Request Source Interrupt Request	Name	GENm.ENj Bit	GRPm.ISj Flag	Vector No. (IRn.IR)
Group 12	SCI0	ERI0 (SCI0 reception error)	GEN12.EN0	GRP12.IS0	114
	SCI1	ERI1 (SCI1 reception error)	GEN12.EN1	GRP12.IS1	
	SCI2	ERI2 (SCI2 reception error)	GEN12.EN2	GRP12.IS2	
	SCI3	ERI3 (SCI3 reception error)	GEN12.EN3	GRP12.IS3	
	SCI4	ERI4 (SCI4 reception error)	GEN12.EN4	GRP12.IS4	
	SCI5	ERI5 (SCI5 reception error)	GEN12.EN5	GRP12.IS5	
	SCI6	ERI6 (SCI6 reception error)	GEN12.EN6	GRP12.IS6	
	SCI7	ERI7 (SCI7 reception error)	GEN12.EN7	GRP12.IS7	
	SCI8	ERI8 (SCI8 reception error)	GEN12.EN8	GRP12.IS8	
	SCI9	ERI9 (SCI9 reception error)	GEN12.EN9	GRP12.IS9	
	SCI10	ERI10 (SCI10 reception error)	GEN12.EN10	GRP12.IS10	
	SCI11	ERI11 (SCI11 reception error)	GEN12.EN11	GRP12.IS11	
	SCI12	ERI12 (SCI12 reception error)	GEN12.EN12	GRP12.IS12	
	RSPi0	SPEi0 (error interrupt)	GEN12.EN13	GRP12.IS13	
	RSPi1	SPEi1 (error interrupt)	GEN12.EN14	GRP12.IS14	
RSPi2	SPEi2 (error interrupt)	GEN12.EN15	GRP12.IS15		

### 15.4.2 Unit Selection

The interrupts from two peripheral modules are handled as one interrupt request. One of each unit pair (MTUn and TPUn) is selected using the SEL.CNj bit (j = 0 to 5) and the interrupt signal from the selected unit is subjected to detection.

For operation examples, see section 15.5.1.5, Unit Selection and Interrupt Status Flags.

Table 15.5 lists the correspondence between the unit pairs and specific interrupt requests.



**Figure 15.3 Unit Selection**

**Table 15.5 Unit n Interrupt Sources (n = 0 to 5)**

Unit Pair No. (selected by SEL.CNj bit)	TPU		MTU2		Interrupt Type	Vector No. (IRn.IR)
	Interrupt Request Source Interrupt Request	Name	Interrupt Request Source Interrupt Request	Name		
Unit pair 0 (selected by SEL.CN0)	TPU6	TGI6A	MTU0	TGIA0	Input capture/compare match	142
	TPU6	TGI6B	MTU0	TGIB0	Input capture/compare match	143
	TPU6	TGI6C	MTU0	TGIC0	Input capture/compare match	144
	TPU6	TGI6D	MTU0	TGID0	Input capture/compare match	145
	—	—	MTU0	TGIE0	Compare match	146
	—	—	MTU0	TGIF0	Compare match	147
Unit pair 1 (selected by SEL.CN1)	TPU7	TGI7A	MTU1	TGIA1	Input capture/compare match	148
	TPU7	TGI7B	MTU1	TGIB1	Input capture/compare match	149
Unit pair 2 (selected by SEL.CN2)	TPU8	TGI8A	MTU2	TGIA2	Input capture/compare match	150
	TPU8	TGI8B	MTU2	TGIA2	Input capture/compare match	151
Unit pair 3 (selected by SEL.CN3)	TPU9	TGI9A	MTU3	TGIA3	Input capture/compare match	152
	TPU9	TGI9B	MTU3	TGIB3	Input capture/compare match	153
	TPU9	TGI9C	MTU3	TGIC3	Input capture/compare match	154
	TPU9	TGI9D	MTU3	TGID3	Input capture/compare match	155
Unit pair 4 (selected by SEL.CN4)	TPU10	TGI10A	MTU4	TGIA4	Input capture/compare match	156
	TPU10	TGI10B	MTU4	TGIB4	Input capture/compare match	157
	—	—	MTU4	TGIC4	Input capture/compare match	158
	—	—	MTU4	TGID4	Input capture/compare match	159
	—	—	MTU4	TCIV4	Overflow/underflow	160
Unit pair 5 (selected by SEL.CN5)	—	—	MTU5	TGIU5	Input capture/compare match	161
	—	—	MTU5	TGIV5	Input capture/compare match	162
	—	—	MTU5	TGIW5	Input capture/compare match	163
	TPU11	TGI11A	—	—	Input capture/compare match	164
	TPU11	TGI11B	—	—	Input capture/compare match	165

Note: • Interrupts to which only the TPU<sub>n</sub> or MTU<sub>n</sub> is allocated are valid when selected and masked when not selected.

## 15.5 Interrupt Operation

The interrupt controller performs the following processing.

- Detecting interrupts
- Enabling and disabling interrupts
- Selecting interrupt request destinations (CPU interrupt, DTC activation, or DMAC activation)
- Determining priority

### 15.5.1 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 IRQ<sub>i</sub> pins (i = 0 to 15) as external interrupt requests by the setting of the IRQMD[1:0] bits in IRQCR<sub>i</sub>.

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 15.3, Interrupt Vector Table.

Grouped interrupt requests are classified into two groups; edge detection interrupt request group and level detection interrupt request group, and retained in the GRPm register (m = group number, 0 to 6 or 12). In both groups, the IRn.IR flag operates as the level interrupt. For details of the interrupt grouping function, see section 15.4, Peripheral Module Interrupt Request Groups and Unit Selection. For grouped interrupt requests, see Table 15.4, Group m Interrupt Requests.

### 15.5.1.1 Operation of Status Flags for Edge-Detected Interrupts

Figure 15.4 shows the operation of the IR flag in IRn 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 DMAC or DTC is the request destination for the interrupt, the IRn.IR flag operation differs according to the DMAC/DTC transfer settings and transfer count. For details, see Table 15.6, Operation at DMAC/DTC Activation.

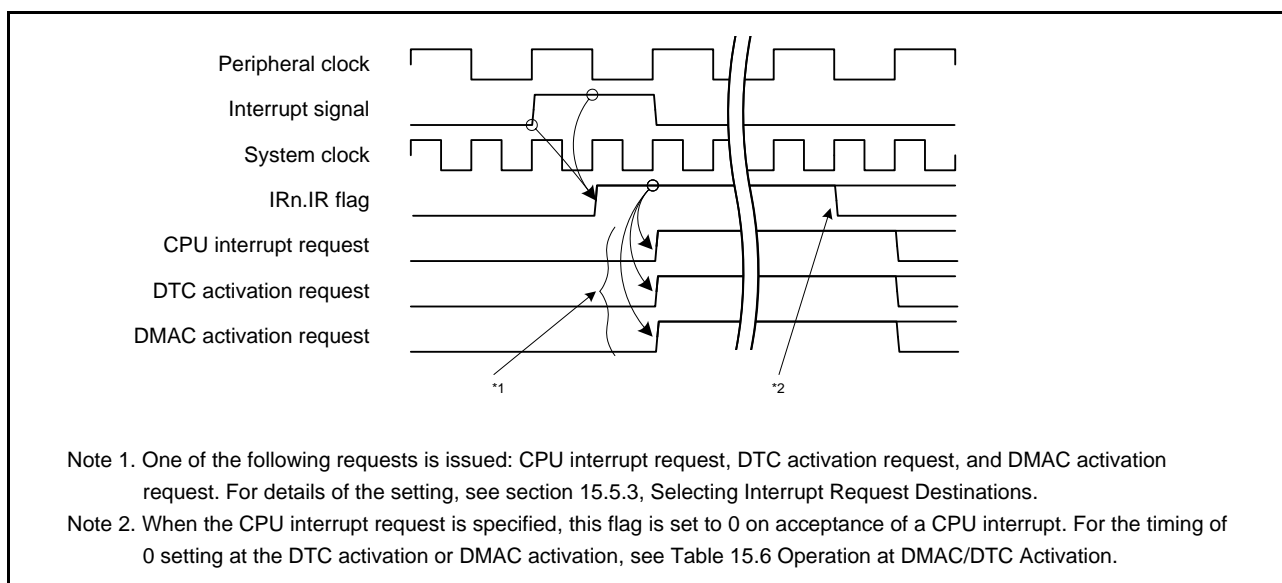


Figure 15.4 IRn.IR Flag Operation for Edge Detection Interrupts

Figure 15.5 to Figure 15.8 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 or peripheral clock, whichever is slower, between issuance of continuous interrupt requests.

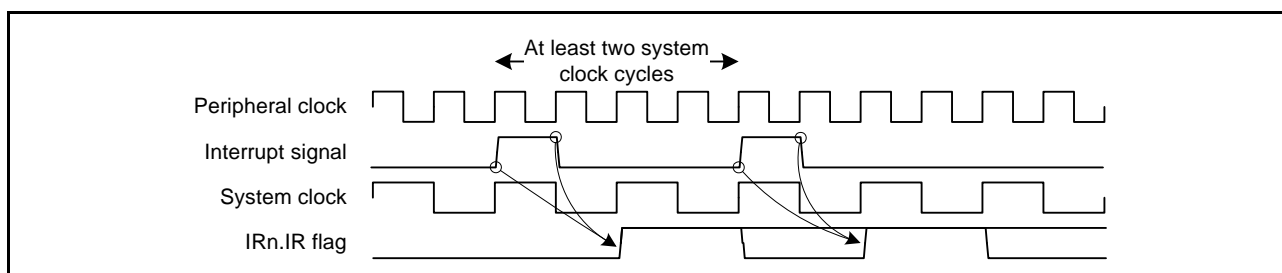


Figure 15.5 Interval Required between Issuance of Continuous Interrupt Requests (when the Frequency of System Clock is Slower than that of the Peripheral Clock)

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 15.6 shows the timing for IRn.IR flag re-setting.

Note 1. When the transmission or reception interrupt of the SCI, RSPI, 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 32, Serial Communications Interface (SC1c, SC1d), section 33, I<sup>2</sup>C Bus Interface (RIIC), and section 35, Serial Peripheral Interface (RSPI).

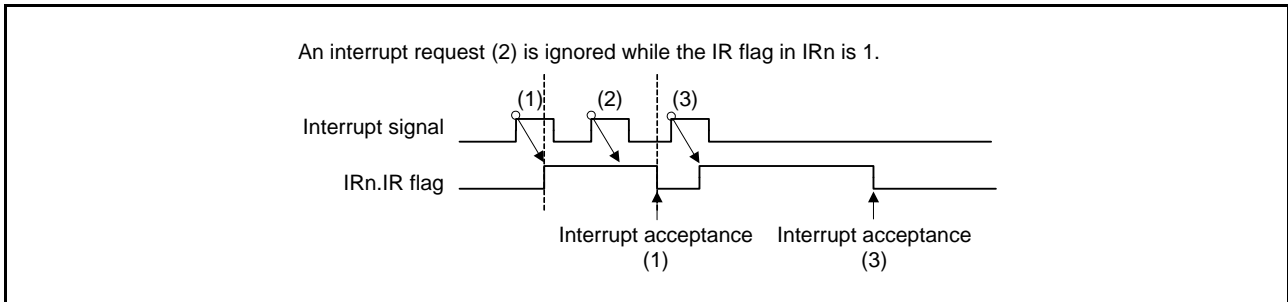


Figure 15.6 Timing for IRn.IR Flag Re-Setting

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 15.7 shows operation when the interrupt is disabled.

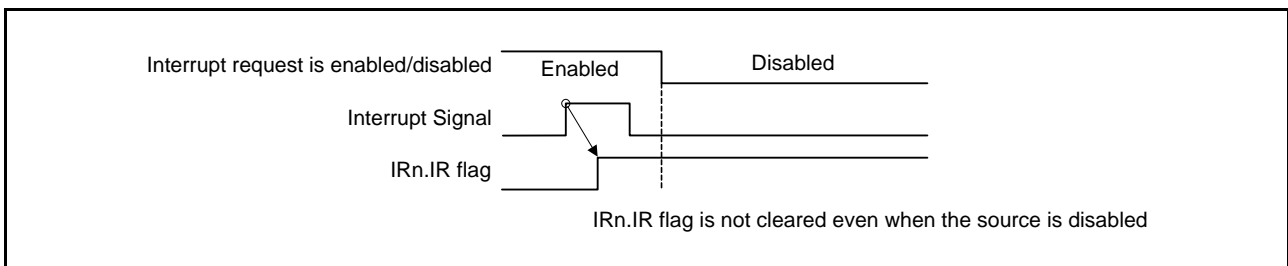


Figure 15.7 Relationship between IRn.IR Flag Operation and Disabling of Interrupt Request

### 15.5.1.2 Operation of Status Flags for Level-Detected Interrupts

Figure 15.8 shows the operation of the interrupt status flag (IR flag) in IRn 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.

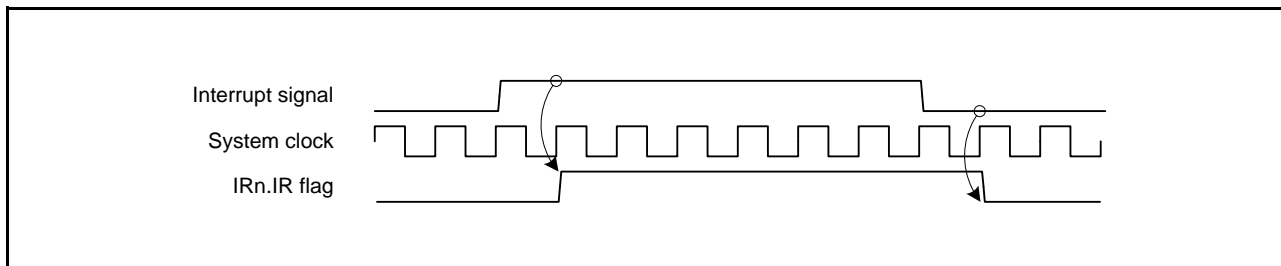


Figure 15.8 IRn.IR Flag Operation for Level Detection Interrupts

Figure 15.9 shows the procedure for handling level detection interrupts.

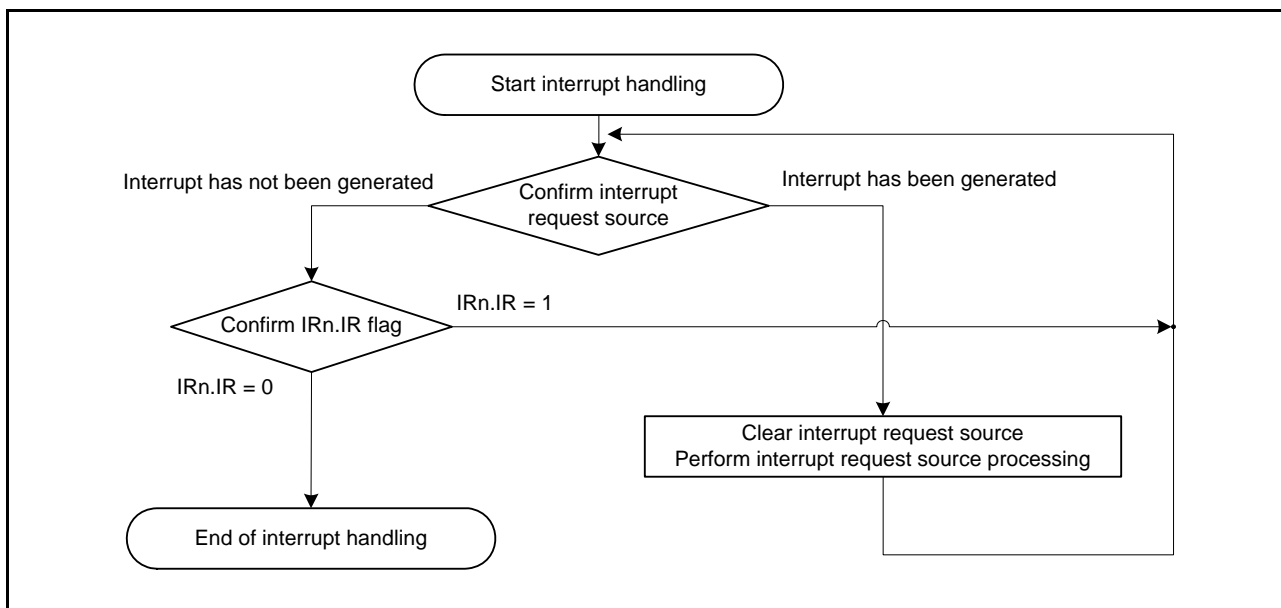


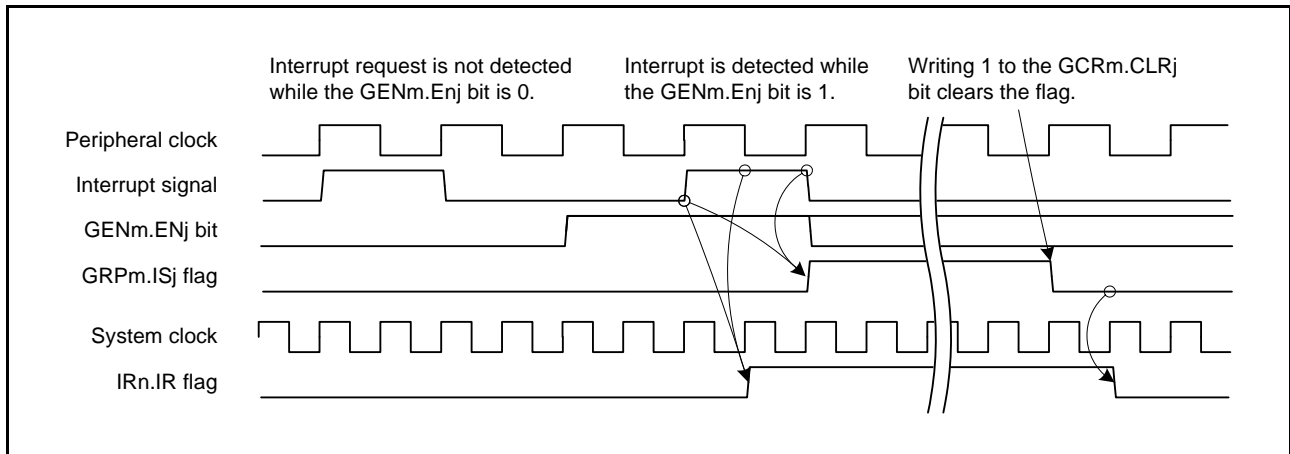
Figure 15.9 Procedure for Handling Level Detection Interrupts



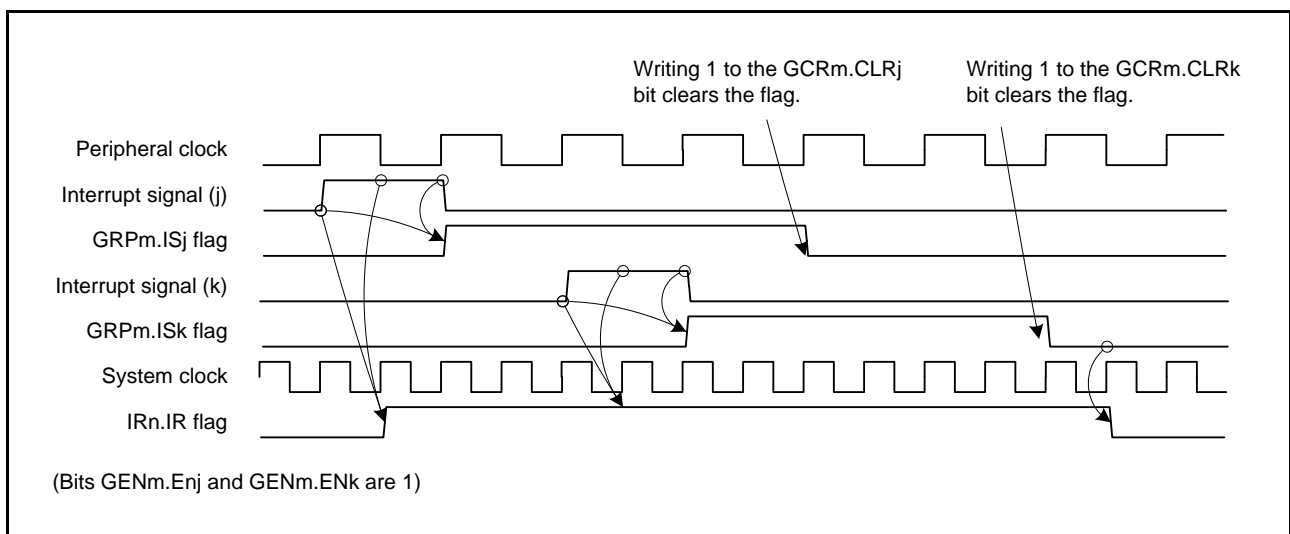
### 15.5.1.3 Edge Detection Group Interrupts and Interrupt Status Flags

Edge detection interrupt requests are grouped into groups 0 to 6. The IRn.IR flag corresponding to the groups operates assuming the interrupt requests as level detection interrupts.

Figure 15.10 shows an operation example in which an edge detection interrupt request is generated and Figure 15.11 shows an operation example in which multiple edge detection interrupt requests allocated to a group are generated.



**Figure 15.10 Operation Example in Which an Edge Detection Group Interrupt Request is Generated**



**Figure 15.11 Operation Example in Which Multiple Edge Detection Interrupt Requests allocated to a Group are Generated**

As for the interrupt handling of edge detection group 0, follow the procedure shown in Figure 15.12.

As for the interrupt handling of edge detection groups 1 to 6, follow the procedure shown in Figure 15.13.

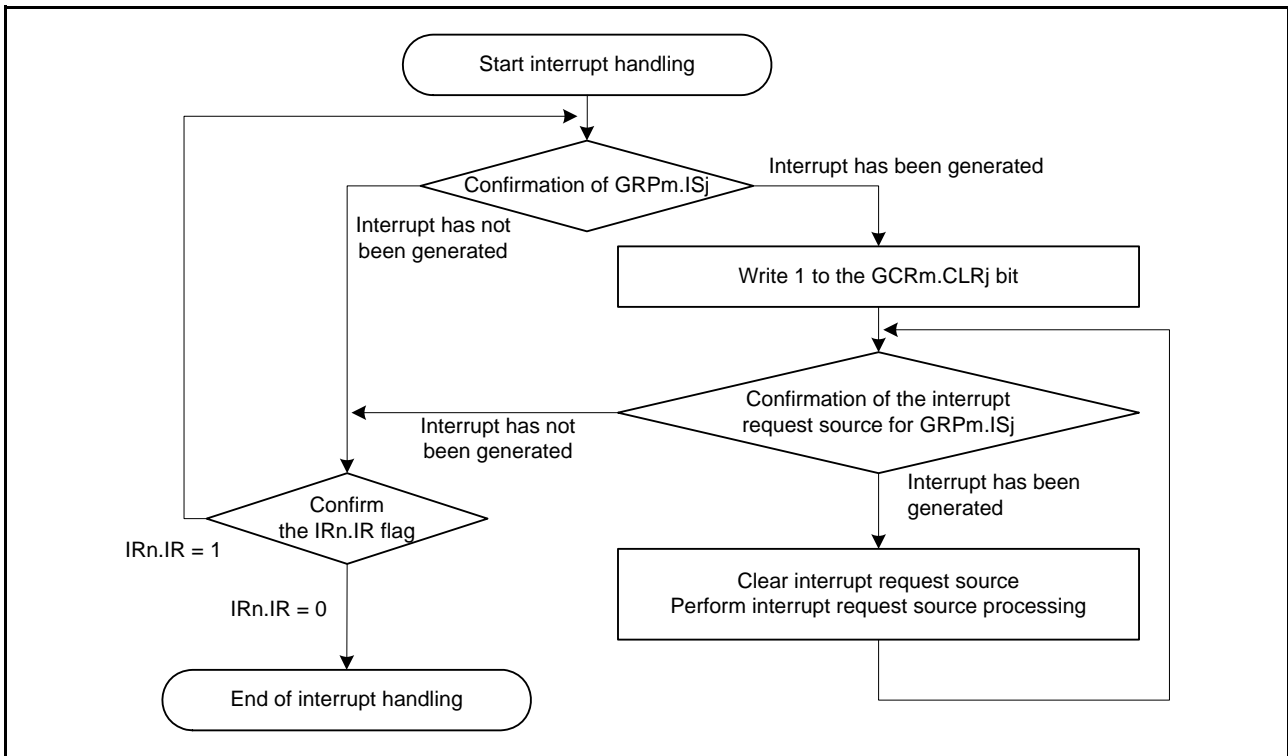


Figure 15.12 Procedure for Interrupt Handling of Edge Detection Group 0

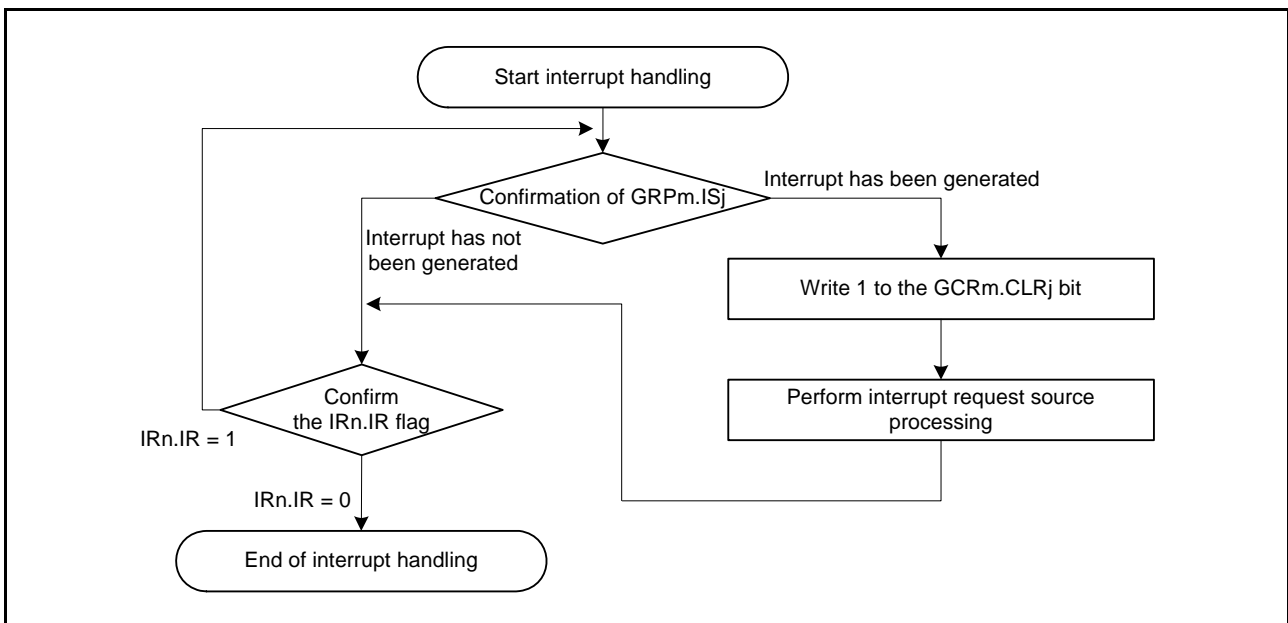
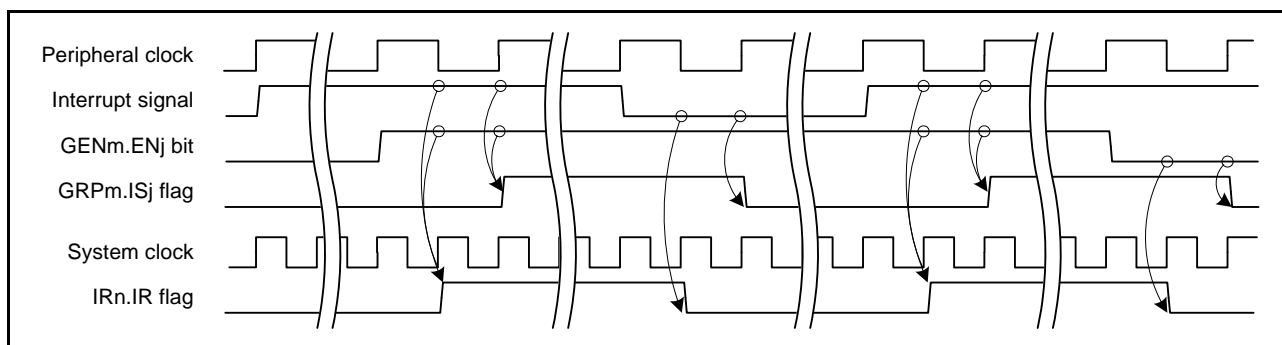


Figure 15.13 Procedure for Interrupt Handling of Edge Detection Groups 1 to 6

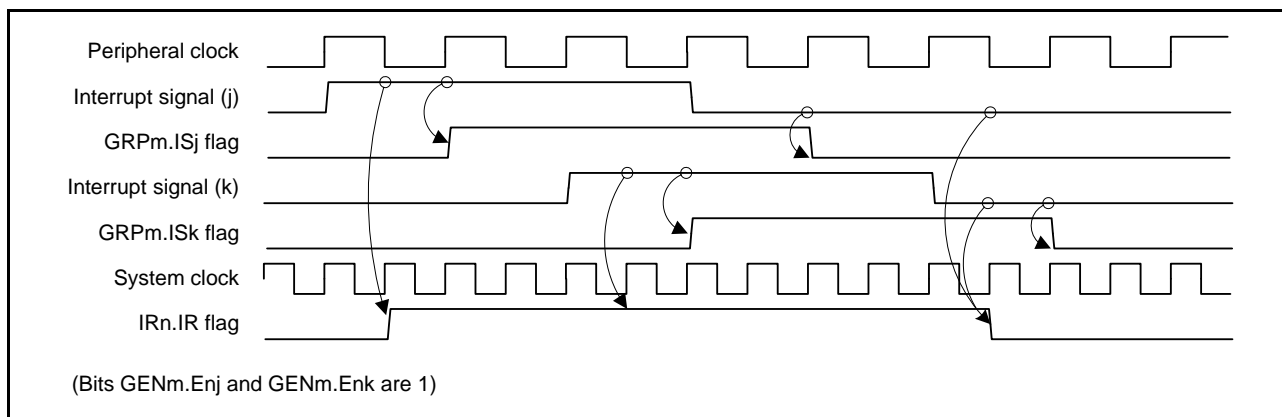
### 15.5.1.4 Level Detection Group Interrupts and Interrupt Status Flags

Level detection interrupt requests are grouped into group 12. The IRn.IR flag corresponding to the group operates assuming the interrupt requests as level detection interrupts.

Figure 15.14 shows an operation example in which a level detection interrupt request is generated and Figure 15.15 shows an operation example in which multiple level detection interrupt requests of a group are generated.



**Figure 15.14 Operation Example in Which a Level Detection Group Interrupt Request is Generated**



**Figure 15.15 Operation Example in Which Multiple Level Detection Interrupt Requests of a Group are Generated**

As for the interrupt handling of a level detection group, follow the procedure shown in Figure 15.16.

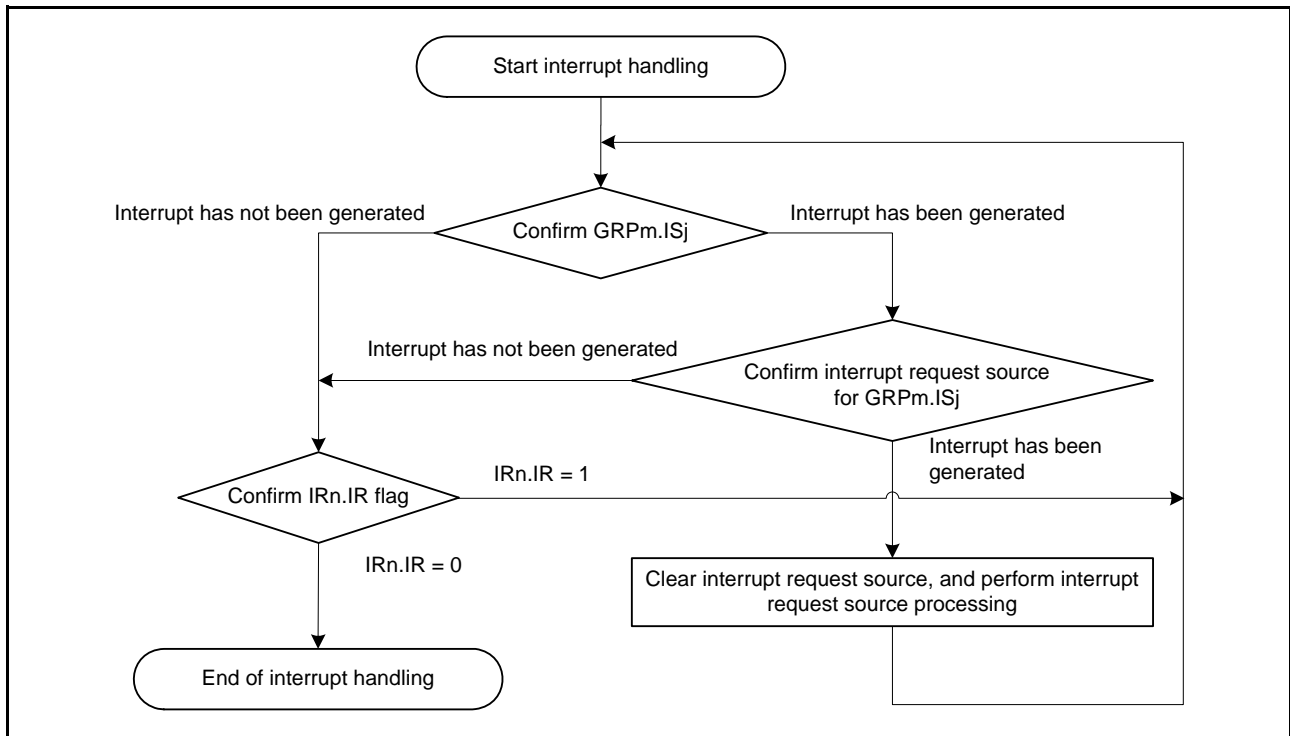


Figure 15.16 Procedure for Interrupt Handling of a Level Detection Group

### 15.5.1.5 Unit Selection and Interrupt Status Flags

Use the following procedure for switching units.

- Disable output of both MTUn (n = 0 to 5) and TPU<sub>n</sub> (n = 6 to 11) interrupts at the corresponding output sources of the interrupt requests, read the last written register, and confirm that writing is completed.
- Confirm that the IR<sub>n</sub>.IR flag is 0.
- Modify the SEL.CN<sub>j</sub> bit.
- Enable output of both MTUn (n = 0 to 5) and TPU<sub>n</sub> (n = 6 to 11) interrupts at the corresponding output sources of the interrupt requests.

Figure 15.17 shows a unit selection example.

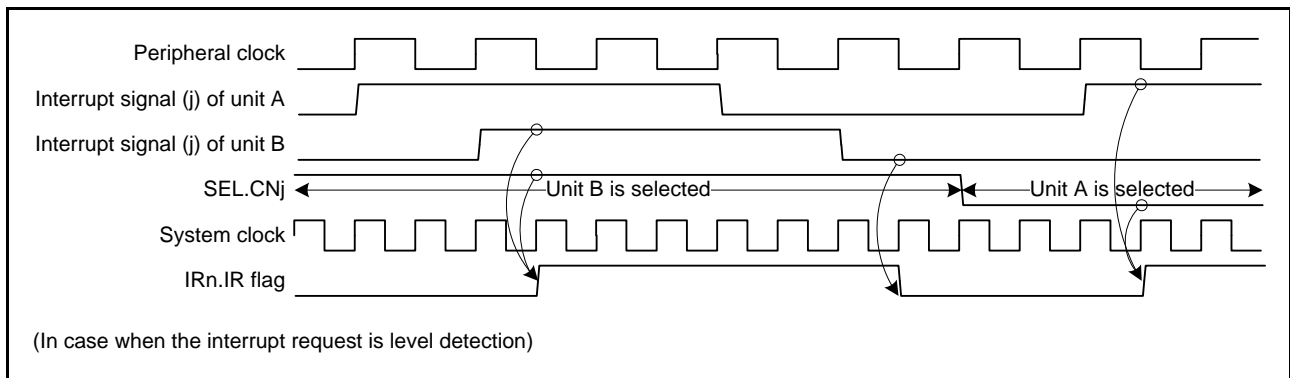


Figure 15.17 Unit Selection Example

### 15.5.2 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. For grouped interrupt requests, enabling the interrupts by the GENm.ENj bits
3. Enabling of the interrupt by the IERm.IENj bit

When an interrupt request that is enabled at the corresponding source is generated, the corresponding IRn.IR flag is set to 1. When a grouped interrupt request is generated, the corresponding GRPm.ISj flag is set to 1 and the IRn.IR flag corresponding to the group 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. For grouped interrupt requests, set the GENm.ENj bit to disable.
3. Set the peripheral module interrupt output enable bit to disable the output. Read the last written register and confirm that writing is completed.
4. Check the IRn.IR flag, and clear the IRn.IR flag if necessary.\*1  
For grouped interrupt requests, check the GRPm.ISj flag or clear the GRPm.ISj flag to 0.

Note 1. To disable the transmission or reception interrupt of the SCI, RSPI, 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 32, Serial Communications Interface (SC1c, SC1d), section 33, I<sup>2</sup>C Bus Interface (RIIC), and section 35, Serial Peripheral Interface (RSPI).

### 15.5.3 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 15.3, Interrupt Vector Table, are not possible. Do not make an interrupt request destination setting that is not indicated by a O in Table 15.3.

If the DMAC or DTC is selected as the destination for requests from an IRQ pin, 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) DMAC Activation

Make the following settings for each source while the IERm.IENj bit is 0.

1. Specify the vector number of the desired interrupt in the DMAC activation request select register (DMRSRm) for the required channel of the DMAC.\*1
2. Set the activation source for the target DMAC channel (DMACm.DMTMD.DCTG[1:0]) to 01b (interrupt module detection).
3. Set the DMAC transfer enable bit for the target DMAC channel (DMACm.DMCNT.DTE) to 1.

After making the above settings, set the IERm.IENj bit to 1.

In addition, set the DMAC operation enable bit (DMAST.DMST) to 1. The order of making settings for each interrupt and enabling the DMAC operation enable bit does not matter.

For the DMAC setting procedure, refer to section 18.3.7, Activating the DMAC in section 18, DMA Controller (DMACA).

#### (2) DTC Activation

Make the following settings for each source while the IERm.IENj bit is 0.

1. Set the DTC activation enable bit in the DTC activation enable register (DTCERn.DTCE) for the pertinent source to 1.\*1

After making the above settings, set the IERm.IENj bit to 1.

In addition, set the DTC module activation bit (DTCST.DTCST) to 1. The order of making settings for each interrupt and enabling the DTC module activation bit does not matter.

For the DTC setting procedure, refer to section 19.5, DTC Setting Procedure, in section 19, Data Transfer Controller (DTCa).

Note 1. Do not set a DTC activation enable bit (DTCERn.DTCE) and a DMAC activation request select register (DMRSRm) to select the same source. Do not select the same source in more than one DMRSRm register.

#### (3) CPU Interrupt Request

If the interrupt request destination is neither the DMAC nor the DTC, the interrupt request is sent to the CPU. Set the IERm.IENj bit to 1 while neither the DMAC activation settings nor the DTC activation settings described above are in place.

Table 15.6 shows operation when the DTC or the DMAC is the request destination.

**Table 15.6 Operation at DMAC/DTC Activation**

Interrupt Request Destination	DISEL	Remaining Number of Transfer Operations	Operation per Request	IR*1	Interrupt Request Destination after Transfer
DMAC	1	≠ 0	DMA transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	DMAC
		= 0	DMA transfer → CPU interrupt	Cleared on interrupt acceptance by the CPU	The DMACm.DMCNT.DTE bit is cleared and the CPU becomes the destination.
	0	≠ 0	DMA transfer	Cleared at the start of DMAC transfer	DMAC
		= 0	DMA transfer*2	Cleared at the start of DMAC transfer*2	The DMACm.DMCNT.DTE bit is cleared and the CPU becomes the destination.
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 DTCER.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 *2	Cleared on interrupt acceptance by the CPU*2	The DTCER.DTCE bit is cleared and the CPU becomes the destination.

DISEL for the DMAC is set by the DMACm.DMCSL.DISEL bit; DISEL for the DTC is set by the DTC.MRB.DISEL bit.

Note 1. When the IRn.IR flag is 1, an interrupt request (DTC or DMAC activation request) that is generated again will be ignored.

Note 2. When the DISEL bit is 0, operation with the remaining number of transfer operations being 0 differs according to whether the source is for DTC or DMAC.

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 19.3, Chain Transfer Conditions in section 19, Data Transfer Controller (DTCa).

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 DMA activation source is to be changed while a transfer is not complete (i.e. while the DMACm.DMCNT.DTE bit has not been cleared) after the settings described under (1) DMAC Activation have been made, follow the procedure below.

1. For both the source to be withdrawn and the source that will have a new target for activation, clear the IENj bits in IERm to 0.
2. Check the state of transfer by the DMAC. If transfer is in progress, wait for its completion.
3. Make the settings described under (1) DMAC Activation.

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 has not been cleared) after the settings described under (2) DTC Activation have been made, follow the procedure below.

1. For both the source to be withdrawn and the source that will have a new target for activation, 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 (2) DTC Activation.

### 15.5.4 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 Signal

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 Signal

The IPR[3:0] bits in IPR<sub>n</sub> have no effect. An interrupt source with a smaller vector number takes precedence.

## (3) Determining Priority when the DMAC is the Request Destination of the Interrupt Signal

The IPR[3:0] bits in IPR<sub>n</sub> have no effect. Regarding the order of priority of DMAC channels, see section 18, DMA Controller (DMACA).

### 15.5.5 Fast Interrupt

The fast interrupt is a facility for faster interrupt processing by the CPU, so is only effective for an interrupt request being conveyed to the CPU. That is, the fast interrupt setting has no effect on interrupt requests for the DTC and DMAC.

The fast interrupt is set up by specifying the vector number of an interrupt source in the FVCT[7:0] bits in FIR and enabling the fast interrupt by setting the FIEN bit in FIR to 1. When the given source generates an interrupt, the interrupt is output to the CPU for handling as the fast interrupt.

The interrupt source selected for the fast interrupt has the highest priority regardless of the setting of the IPR[3:0] bits in IPR<sub>n</sub>.

For details on the fast interrupt, see section 14, Exception Handling.

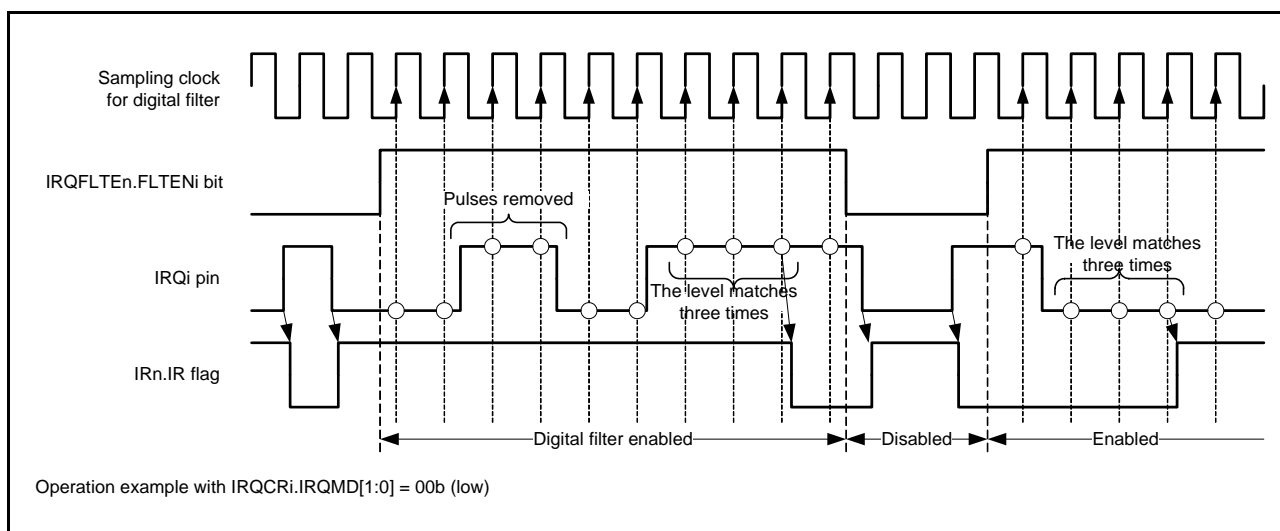
### 15.5.6 Digital Filter

The digital filter function is provided for the external interrupt request IRQ<sub>i</sub> pins ( $i = 0$  to 15) 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 IRQ<sub>i</sub> pin, set the sampling clock cycle (PCLK, PCLK/8, PCLK/32, or PCLK/64) with the IRQFLTCn.FCLKSEL<sub>i</sub>[1:0] bits ( $n = 0$  or 1,  $i = 0$  to 15) and set the IRQFLTE<sub>n</sub>.FLTEN<sub>i</sub> 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 15.18 shows an example of digital filter operation.



**Figure 15.18 Digital Filter Operation Example**

Before software standby mode is entered, set the IRQFLTE<sub>n</sub>.FLTEN<sub>i</sub> and NMIFLTE.NFLTEN bits to 0 (digital filter disabled). To use the digital filter again after return from software standby mode, set the IRQFLTE<sub>n</sub>.FLTEN<sub>i</sub> or NMIFLTE.NFLTEN bit to 1 (digital filter enabled).



### 15.5.7 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 to 0 (interrupt request disabled).
2. Clear the IRQFLTCn.FCLKSELi[1:0] bits (n = 0 or 1, i = 0 to 15) to 0 (digital filter disabled).
3. Set the digital filter sampling clock with the IRQFLTCn.FCLKSELi[1:0] bits.
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 to 0 (if edge detection is in use).
7. Set the IRQFLTEn.FLTENi bit to 1 (digital filter enabled).
8. If the interrupt is to be used for DMAC activation, set the DMRSRm.DMRS[7:0] bits. If the interrupt is to be used for DTC activation, set the DTCERn.DTCE bit. The interrupt will be a CPU interrupt if neither of these settings is made.
9. Set the IERm.IENj bit to 1 (interrupt request enabled).

### 15.6 Non-maskable Interrupt Operation

There are six types of non-maskable interrupt: the NMI pin interrupt, oscillation stop detection interrupt, WDT underflow/refresh error, 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 or DMAC activation. 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.

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).
3. To use the NMI pin, set the digital filter sampling clock with the NMIFLTC.NFCLKSEL[1:0] bits.
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).
7. Enable the non-maskable interrupt by writing 1 to the corresponding bit in the non-maskable interrupt enable register (NMIER).

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 14, 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.WDTCLR bit clears the WDT underflow/refresh error status flag (NMISR.WDTST) 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.

## 15.7 Return from Power-Down States

The interrupt sources that can be used to return operation from sleep mode, all-module clock stop mode, or software standby mode are listed in Table 15.3, 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 power-down mode.

### 15.7.1 Return from 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 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.
  4. Use the ENj bit in GENm to enable the given grouped interrupt requests.
- Non-maskable interrupts

Use the NMIER register to enable the given interrupt request.

### 15.7.2 Return from All-Module Clock Stop Mode

If the interrupt controller is to return operation from all-module clock stop mode in response to an interrupt or non-maskable interrupt, make the following settings for the interrupt.

- Interrupts
  1. Select the interrupt source that enables the return from the all-module clock stop mode.
  2. Select the CPU as the interrupt request destination.
  3. Use the IENj bit in IERm 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.
- Non-maskable interrupts

Use the NMIER register to enable the given interrupt request.

### 15.7.3 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 IEN<sub>j</sub> bit in IER<sub>m</sub> 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 (IPR<sub>n</sub>) 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.

- 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 (IRQFLTE<sub>n</sub>.FLTEN<sub>i</sub> = 0, NMIFLTE.NFLTEN = 0).
  2. To use the digital filter again after return from software standby mode, enable the digital filter (IRQFLTE<sub>n</sub>.FLTEN<sub>i</sub> = 1, NMIFLTE.NFLTEN = 1).

## 15.8 Usage Note

### 15.8.1 Note on WAIT Instruction Used with Non-Maskable Interrupt

Confirm that all status flags in register NMISR are 0 whenever a WAIT instruction is to be executed.

## 16. Buses

### 16.1 Overview

Table 16.1 lists the bus specifications, Figure 16.1 shows the bus configuration, and Table 16.2 lists the addresses assigned for each bus.

**Table 16.1 Bus Specifications**

Bus Type		Description
CPU bus	Instruction bus	<ul style="list-style-type: none"> <li>Connected to the CPU (for instructions)</li> <li>Connected to on-chip memory (RAM, ROM)</li> <li>Operates in synchronization with the system clock (ICLK)</li> </ul>
	Operand bus	<ul style="list-style-type: none"> <li>Connected to the CPU (for operands)</li> <li>Connected to on-chip memory (RAM, ROM)</li> <li>Operates in synchronization with the system clock (ICLK)</li> </ul>
Memory bus	Memory bus 1	<ul style="list-style-type: none"> <li>Connected to RAM</li> </ul>
	Memory bus 2	<ul style="list-style-type: none"> <li>Connected to ROM</li> </ul>
Internal main bus	Internal main bus 1	<ul style="list-style-type: none"> <li>Connected to the CPU</li> <li>Operates in synchronization with the system clock (ICLK)</li> </ul>
	Internal main bus 2	<ul style="list-style-type: none"> <li>Connected to the DMAC, DTC</li> <li>Connected to on-chip memory (RAM, ROM)</li> <li>Operates in synchronization with the system clock (ICLK)</li> </ul>
Internal peripheral bus	Internal peripheral bus 1	<ul style="list-style-type: none"> <li>Connected to peripheral modules (DTC, DMAC, interrupt controller, and bus error monitoring section)</li> <li>Operates in synchronization with the system clock (ICLK)</li> </ul>
	Internal peripheral bus 2	<ul style="list-style-type: none"> <li>Connected to peripheral modules (modules other than those connected to internal peripheral buses 1, 3, 4, and 5)</li> <li>Operates in synchronization with the peripheral-module clock (PCLKB)</li> </ul>
	Internal peripheral bus 3	<ul style="list-style-type: none"> <li>Connected to peripheral modules (USB)</li> <li>Operates in synchronization with the peripheral-module clock (PCLKB)</li> </ul>
	Internal peripheral bus 4	Reserved area
	Internal peripheral bus 5	Reserved area
	Internal peripheral bus 6	<ul style="list-style-type: none"> <li>Connected to ROM (P/E) and E2 DataFlash memory</li> <li>Operates in synchronization with the FlashIF clock (FCLK)</li> </ul>
External bus	CS area	<ul style="list-style-type: none"> <li>Connected to the external devices</li> <li>Operates in synchronization with the external-bus clock (BCLK)</li> </ul>

P/E: Programming/Erase

BCLK (external-bus clock): 50 MHz (max.) The CSC (CS area controller) operate in synchronization with the BCLK.

BCLK pin output: The frequency is the same as the BCLK as default. 1/2 BCLK can be supplied by setting the BCLK pin output select bit (BCKCR.BCLKDIV) in the external bus clock control register. For details, see section 9, Clock Generation Circuit.

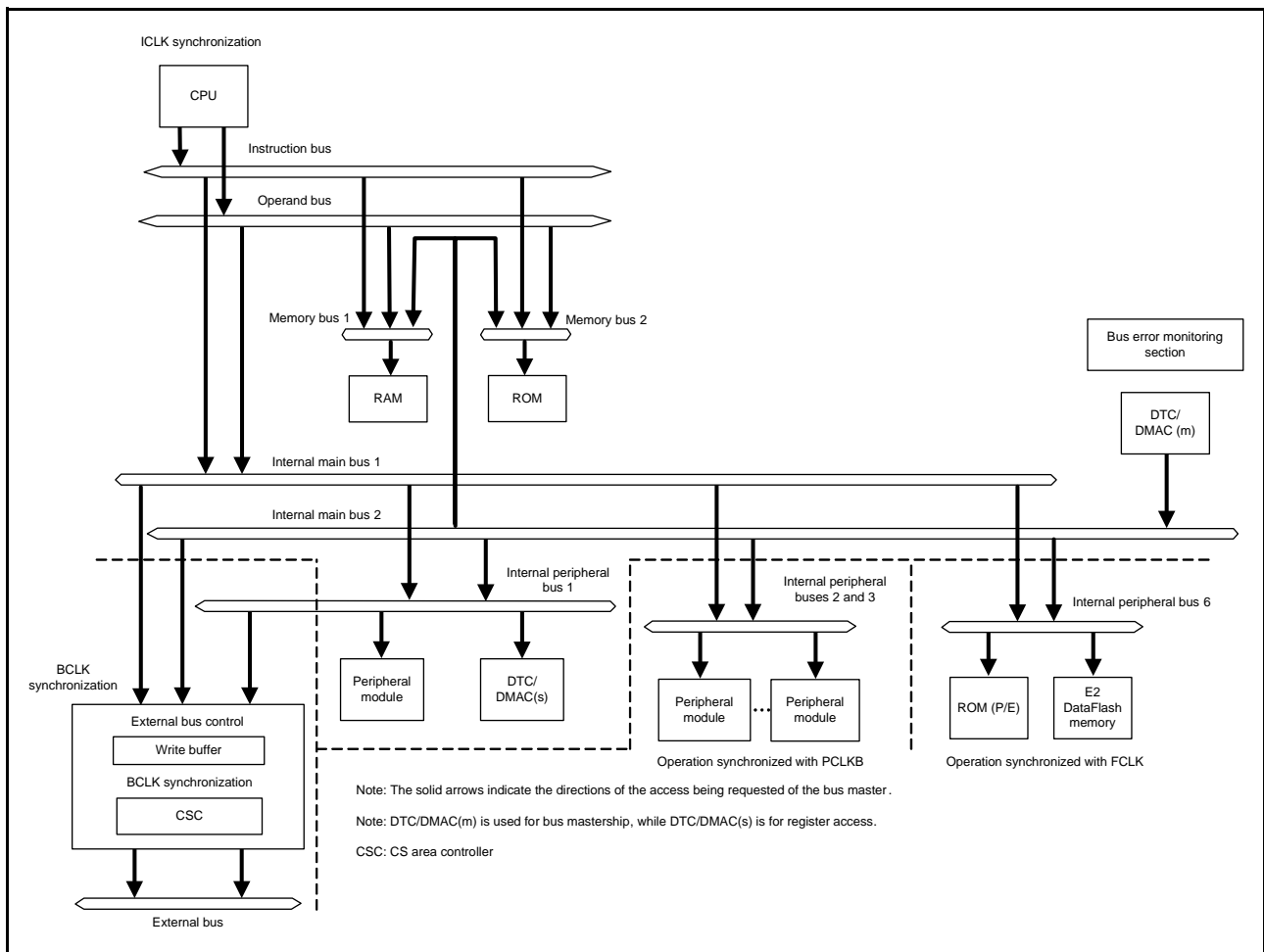


Figure 16.1 Bus Configuration

**Table 16.2** Addresses Assigned for Each Bus

Address	Bus		Area	
	On-Chip ROM Mode		On-Chip ROM Mode	
	Enabled	Disabled	Enabled	Disabled
0000 0000h to 0001 FFFFh	Memory bus 1		RAM	
0002 0000h to 0007 FFFFh			Reserved area	
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			
000C 0000h to 000D FFFFh	Reserved area			
000E 0000h to 000F FFFFh	Reserved area			
0010 0000h to 00FF FFFFh	Internal peripheral bus 6	Reserved area	E2 DataFlash memory, FCU RAM, ROM (P/E) (for programming only)	Reserved area
0100 0000h to 07FF FFFFh	External bus		External address space (CS1 to CS7)	
0800 0000h to 0FFF FFFFh			Reserved area	
1000 0000h to 7FFF FFFFh	Reserved area		Reserved area	
8000 0000h to FEFF FFFFh	Memory bus 2	Reserved area	ROM (for reading only)	Reserved area
FF00 0000h to FFFF FFFFh		External bus		External address space (CS0)

## 16.2 Description of Buses

### 16.2.1 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 or to ROM and external space is possible.

### 16.2.2 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, a bus has a lower priority when the request of that bus is accepted.

### 16.2.3 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, DMAC (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, DMAC are arbitrated by internal main bus 2. The order of priority is DMAC, and then DTC as listed in Table 16.3.

Between the DTC and DMAC, only the one that accepted the activation request issues the bus mastership request. The priority order of activation requests between the DTC and DMAC is DMAC0, DMAC1, DMAC2, DMAC3, and then DTC, regardless of the BUSPRI setting.

If the CPU and another bus master are requesting access to different buses (on-chip memory, internal peripheral buses 1 to 6, and external bus), 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 16.3 Order of Priority for Bus Masters**

Priority	Bus Master
High	DMAC
↑	DTC
Low	CPU

### 16.2.4 Internal Peripheral Buses

Connection of peripheral modules to the internal peripheral buses is as described in Table 16.4.

**Table 16.4 Connection of Peripheral Modules to the Internal Peripheral Buses**

Type of Bus	Peripheral Modules
Internal peripheral bus 1	DTC, DMAC, interrupt controller, and bus error monitoring section
Internal peripheral bus 2	Peripheral modules other than those connected to internal peripheral buses 1, 3, 4, and 5
Internal peripheral bus 3	USB
Internal peripheral bus 4	Reserved area
Internal peripheral bus 5	Reserved area
Internal peripheral bus 6	ROM (P/E)/E2 DataFlash memory, and FCU RAM

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 6.

The priority order of 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 bus 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 (Refer to Figure 16.2).

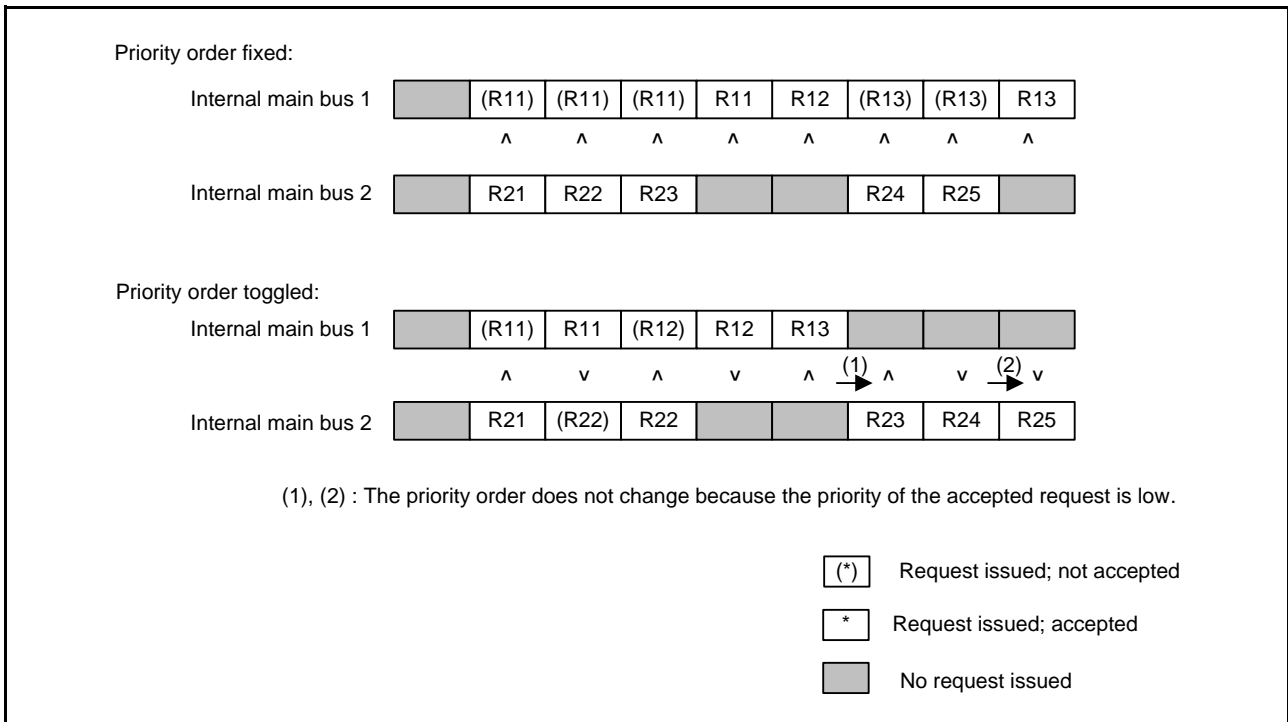


Figure 16.2 Priority Order between Internal Peripheral Bus Accesses

### 16.2.5 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 (Refer to Figure 16.3).

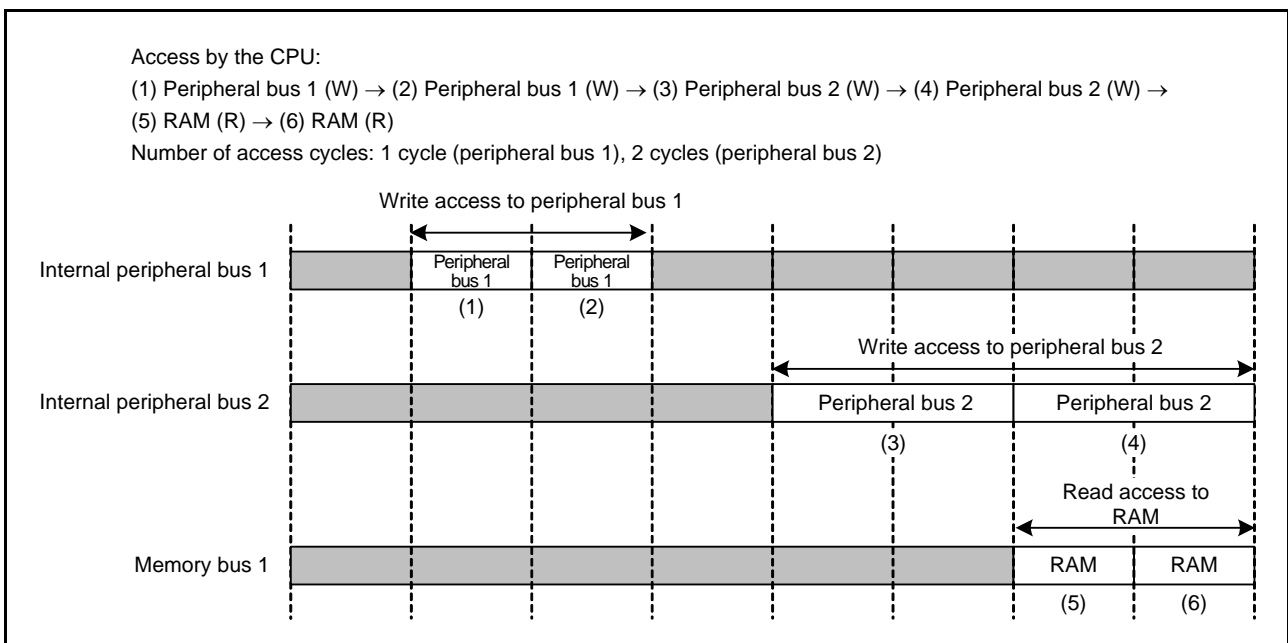


Figure 16.3 Write Buffer Function



### 16.2.6 External Bus

Table 16.5 lists the specifications of the external bus.

The external bus controller arbitrates requests for bus mastership from internal main bus 1, and internal main bus 2..

The priority order of these two buses can be set using the external bus priority control bits (BPEB[1:0]) in the bus priority control register (BUSPRI). When the priority order is fixed, the order is internal main bus 2, and then internal main bus 1. When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted, between internal main bus 1 and internal main bus 2.

The order of accepting requests may change depending on the BUSPRI setting (Refer to Figure 16.4).

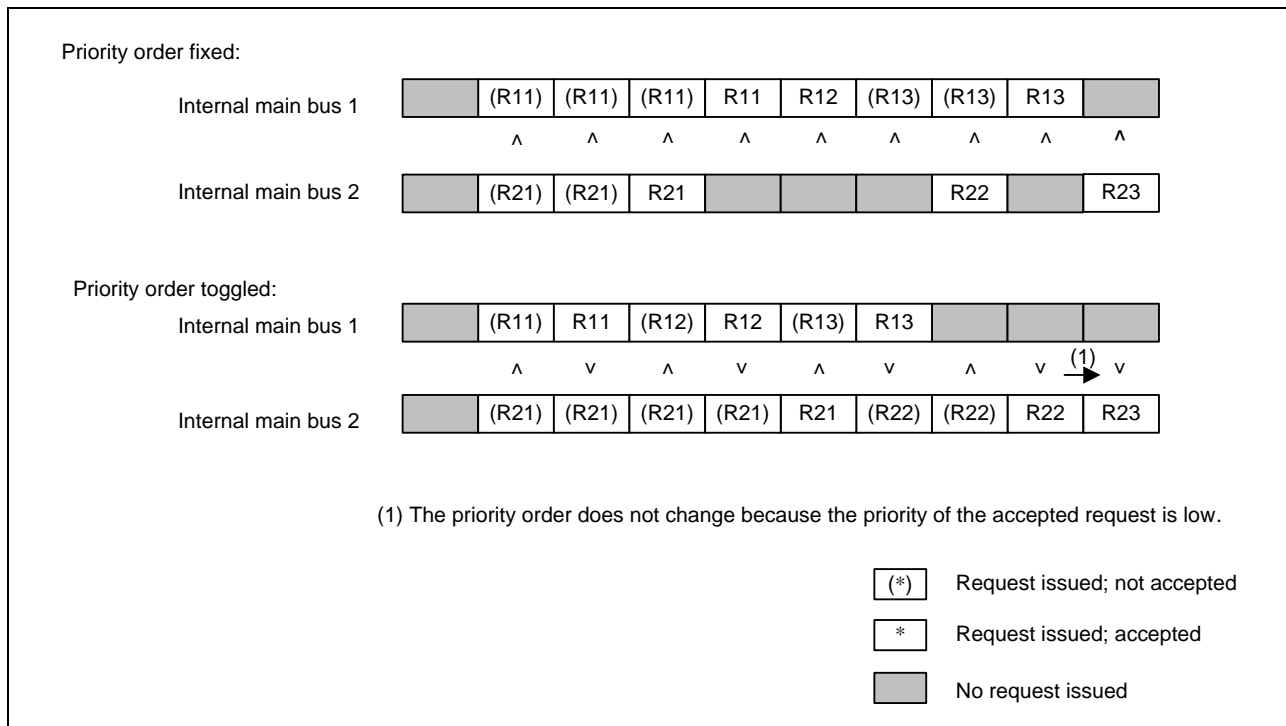


Figure 16.4 Priority Order of Internal Peripheral Bus Accesses

Table 16.5 Specifications of the External Bus

Item	Description
External address space	<ul style="list-style-type: none"> <li>An external address space is divided into eight CS areas (CS0 to CS7) for management.</li> <li>Chip select signals can be output for each area.</li> <li>Bus width can be set for each area.                      Separate bus: An 8, 16, or 32-bit bus space is selectable.                      Address/data multiplexed bus: An 8 or 16-bit bus space is selectable.</li> <li>An endian mode can be specified for each area.</li> </ul>
CS area controller	<ul style="list-style-type: none"> <li>Recovery cycles can be inserted.                      Read recovery: Up to 15 cycles                      Write recovery: Up to 15 cycles</li> <li>Cycle wait function: Wait for up to 31 cycles (page access: up to 7 cycles)</li> <li>Wait control can be used to set up the following.                      Timing of assertion and negation for chip-select signals (CS0# to CS7#)                      The timing of assertion of the read signal (RD#) and write signals (WR#, WR0# to WR3#)                      The timing with which data output starts and ends</li> <li>Write access mode: Single write strobe mode/byte strobe mode</li> <li>Separate bus or address/data multiplexed bus can be set for each area.</li> </ul>
Write buffer function	When write data from the bus master has been written to the write buffer, write access by the bus master is completed.
Frequency	<ul style="list-style-type: none"> <li>The CS area controller (CSC) operates in synchronization with the external-bus clock (BCLK).</li> </ul>

Table 16.6 lists the input/output pins of the external bus.

**Table 16.6 Pin Configuration of the External Bus**

Pin Name	I/O	Description
A23 to A0*1	Output	Address output pins
D31 to D0	I/O	Data input/output pins D31 to D0 pins are enabled when the 32-bit bus space is specified. D15 to D0 pins are enabled when the 16-bit bus space is specified. D7 to D0 pins are enabled when the 8-bit bus space is specified.
BC0#*1	Output	A strobe signal; (the BC0# signal being at the low level) during access to an external address space in single write strobe mode indicates that D7 to D0 are valid. When an 8-bit bus space is specified, this output pin is always held low regardless of write access mode.
BC1#	Output	A strobe signal; (the BC1# signal being at the low level) during access to an external address space in single write strobe mode indicates that D15 to D8 are valid. This pin is not used when the 8-bit bus space is specified.
BC2#	Output	A strobe signal; (the BC2# signal being at the low level) during access to an external address space in single write strobe mode indicates that D23 to D16 are valid. This pin is not used when the 8- or 16-bit bus space is specified.
BC3#	Output	A strobe signal; (the BC3# signal being at the low level) during access to an external address space in single write strobe mode indicates that D31 to D24 are valid. This pin is not used when the 8- or 16-bit bus space is specified.
CS0#	Output	A chip select signal for area 0 (CS0)
CS1#	Output	A chip select signal for area 1 (CS1)
CS2#	Output	A chip select signal for area 2 (CS2)
CS3#	Output	A chip select signal for area 3 (CS3)
CS4#	Output	A chip select signal for area 4 (CS4)
CS5#	Output	A chip select signal for area 5 (CS5)
CS6#	Output	A chip select signal for area 6 (CS6)
CS7#	Output	A chip select signal for area 7 (CS7)
RD#	Output	A strobe signal indicating that reading from an external address space (CS0 to CS7) is in progress
WR0#/WR#*2	Output	WR0# signal is a strobe signal indicates that (the WR0# signal being at the low level) writing to an external address space is in progress in byte strobe mode, and D7 to D0 are valid. WR# signal is a strobe signal that indicates writing to an external address space is in progress in single write strobe mode. When an 8-bit bus space is specified, this output pin is held low during a write access regardless of write access mode.
WR1#	Output	A strobe signal; (the WR1# signal being at the low level) during writing to an external address space in byte strobe mode indicates that D15 to D8 are valid. This signal is invalid in single write strobe mode. This pin is not used when the 8-bit bus space is specified.
WR2#	Output	A strobe signal; (the WR2# signal being at the low level) during writing to an external address space in byte strobe mode indicates that D23 to D16 are valid. This signal is invalid in single write strobe mode. This pin is not used when the 8- or 16-bit bus space is specified.
WR3#	Output	A strobe signal; (the WR3# signal being at the low level) during writing to an external address space in byte strobe mode indicates that D31 to D24 are valid. This signal is invalid in single write strobe mode. This pin is not used when the 8- or 16-bit bus space is specified.
ALE	Output	Address latch signal when address/data multiplexed bus is selected.
WAIT#	Input	A wait request signal when accessing the external address space (CS0 to CS7) (Low: Wait request)

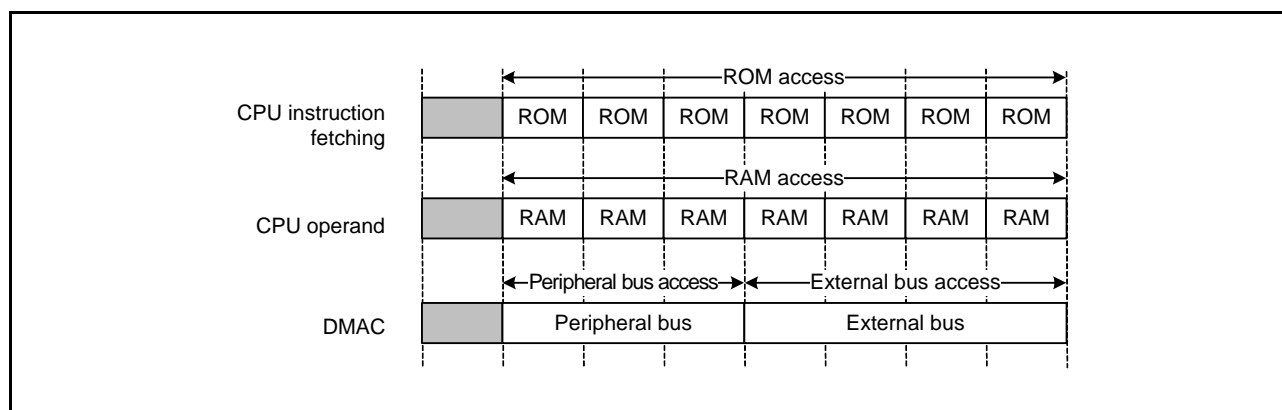
Note 1. The A0 and BC0# pin functions share the same pin, and either becomes effective according to the area, with the function being A0 in byte strobe mode and BC0# in single write strobe mode. Note that setting the 8-bit external bus width is prohibited in single write strobe mode. For information on other multiplexed pin functions, see section 20, I/O Ports.

Note 2. The WR0# signal and WR# signal are identical. The WR0# signal is particularly referred to as WR# in single write strobe mode.

### 16.2.7 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 DMAC is able to handle transfer between a peripheral bus and the external bus at the same time.

An example of parallel operations is shown in Figure 16.5. In this example, the CPU is able to employ the instruction and operand buses for simultaneous access to ROM and RAM, respectively. Furthermore, the DMACA simultaneously employs internal main bus 2 for access to a peripheral bus or the external bus during access to RAM and ROM by the CPU.



**Figure 16.5 Example of Parallel Operations**

### 16.2.8 Bus Settings

- (1) Set the mode of the external bus in the CSn mode register (CSnMOD), CSn wait-control register 1 (CSnWCR1), CSn wait-control register 2 (CSnWCR2), CSn control register (CSnCR), CSn recovery-cycle setting register (CSnREC), CS recovery cycle insertion enable register (CSRECEEN), bus error monitoring enable register (BEREN), and bus priority control register (BUSPRI).
- (2) Make settings for pins in the CS output enable register (PFCSE), CS output pin selection register 0 (PFCSS0), CS output pin selection register 1 (PFCSS1), address output enable register 0 (PFAOE0), address output enable register 1 (PFAOE1), external-bus control register 0 (PFBCR0), and external-bus control register 1 (PFBCR1).
- (3) Set up pins to be used as input port pins.
- (4) Set the external-bus enable bit (EXBE) in the system control register 0 (SYSCR0) to 1 (enabling the external bus).

### 16.2.9 Restrictions

#### (1) Prohibition of Access that Spans 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. Setting must be made so that two areas are not accessed at the same time by a single word or longword access.

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

- (a) Although the external space has a per-area ending-switching facility (only for data), the allocation of data to be handled by RMPA or string-manipulation instructions to an area where the endian differs from that of the chip is prohibited, and operation is not guaranteed if this restriction is not observed. If data to be handled by RMPA or string-manipulation instructions are allocated to the external space, they must be allocated to areas where the endian setting is the same as that for the chip.
- (b) 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.

#### (3) Restriction on Endian

When the endian setting for each area is different from that for the chip, no instruction code can be allocated in the area. The instruction code should be allocated to the external space whose endian setting is the same as that for the chip.

## 16.3 Register Descriptions

### 16.3.1 CSn Control Register (CSnCR) (n = 0 to 7)

Address(es): CS0CR 0008 3802h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	MPXEN	—	—	—	EMOD E	—	—	BSIZE[1:0]		—	—	—	EXENB
Value after reset:															
0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1

Address(es): CS1CR 0008 3812h, CS2CR 0008 3822h, CS3CR 0008 3832h,  
CS4CR 0008 3842h, CS5CR 0008 3852h, CS6CR 0008 3862h, CS7CR 0008 3872h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	MPXEN	—	—	—	EMOD E	—	—	BSIZE[1:0]		—	—	—	EXENB
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	EXENB	Operation Enable	0: Operation is disabled 1: Operation is enabled	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	BSIZE[1:0]	External Bus Width Select	b5 b4 0 0: A 16-bit bus space is selected 0 1: A 32-bit bus space is selected 1 0: An 8-bit bus space is selected 1 1: Setting prohibited	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	EMODE	Endian Mode	0: Endian of area n is the same as the endian of operating mode. 1: Endian of area n is not the endian of operating mode. (n = 0 to 7)	R/W
b11 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	MPXEN	Address/Data Multiplexed I/O Interface Select	0: Separate bus interface is selected for area n. 1: Address/data multiplexed I/O interface is selected for area n. (n = 0 to 7)	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Do not write to the CSnCR register while access to the CSn area is in progress.

#### EXENB Bit (Operation Enable)

This bit enables or disables operation of the respective CS areas.

After this LSI is reset, operation is enabled (EXENB = 1) only for area 0; operation in other areas is disabled (EXENB = 0).

An attempt at access to an area for which operation has been disabled does not lead to access via the external bus.

However, if the illegal address access detection enable bit in the bus error monitoring enable register has been set to enable detection (BEREN.IGAEN = 1), such an attempt will lead to an illegal-access error.

**BSIZE[1:0] Bits (External Bus Width Select)**

These bits specify the data bus width of each area.

The data bus width of area 0 (CS0) after a reset depends on the setting of the bus width in operating mode.

When the address/data multiplexed I/O interface is selected with the MPXEN bit, the BSIZE[1:0] bits should not be set to the 32-bit bus space. If set, the operation cannot be guaranteed.

**EMODE Bit (Endian Mode)**

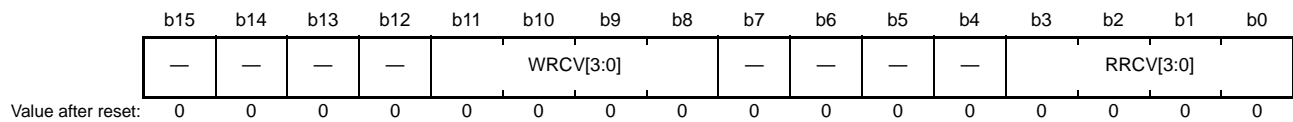
This bit specifies the endian of each area.

When the endian setting for each area is different from that for the chip, no instruction code can be allocated in the area.

The instruction code should be allocated to the external space whose endian setting is the same as that for the chip.

**16.3.2 CSn Recovery Cycle Register (CSnREC) (n = 0 to 7)**

Address(es): CS0REC 0008 380Ah, CS1REC 0008 381Ah, CS2REC 0008 382Ah, CS3REC 0008 383Ah, CS4REC 0008 384Ah, CS5REC 0008 385Ah, CS6REC 0008 386Ah, CS7REC 0008 387Ah



Bit	Symbol	Bit Name	Description	R/W																																																			
b3 to b0	RRCV[3:0]	Read Recovery	<table border="0"> <tr> <td>b3</td> <td>b0</td> <td></td> </tr> <tr> <td>0 0 0 0</td> <td>0</td> <td>No recovery cycle is inserted.</td> </tr> <tr> <td>0 0 0 1</td> <td>1</td> <td>1 recovery cycle is inserted.</td> </tr> <tr> <td>0 0 1 0</td> <td>2</td> <td>2 recovery cycles are inserted.</td> </tr> <tr> <td>0 0 1 1</td> <td>3</td> <td>3 recovery cycles are inserted.</td> </tr> <tr> <td>0 1 0 0</td> <td>4</td> <td>4 recovery cycles are inserted.</td> </tr> <tr> <td>0 1 0 1</td> <td>5</td> <td>5 recovery cycles are inserted.</td> </tr> <tr> <td>0 1 1 0</td> <td>6</td> <td>6 recovery cycles are inserted.</td> </tr> <tr> <td>0 1 1 1</td> <td>7</td> <td>7 recovery cycles are inserted.</td> </tr> <tr> <td>1 0 0 0</td> <td>8</td> <td>8 recovery cycles are inserted.</td> </tr> <tr> <td>1 0 0 1</td> <td>9</td> <td>9 recovery cycles are inserted.</td> </tr> <tr> <td>1 0 1 0</td> <td>10</td> <td>10 recovery cycles are inserted.</td> </tr> <tr> <td>1 0 1 1</td> <td>11</td> <td>11 recovery cycles are inserted.</td> </tr> <tr> <td>1 1 0 0</td> <td>12</td> <td>12 recovery cycles are inserted.</td> </tr> <tr> <td>1 1 0 1</td> <td>13</td> <td>13 recovery cycles are inserted.</td> </tr> <tr> <td>1 1 1 0</td> <td>14</td> <td>14 recovery cycles are inserted.</td> </tr> <tr> <td>1 1 1 1</td> <td>15</td> <td>15 recovery cycles are inserted.</td> </tr> </table>	b3	b0		0 0 0 0	0	No recovery cycle is inserted.	0 0 0 1	1	1 recovery cycle is inserted.	0 0 1 0	2	2 recovery cycles are inserted.	0 0 1 1	3	3 recovery cycles are inserted.	0 1 0 0	4	4 recovery cycles are inserted.	0 1 0 1	5	5 recovery cycles are inserted.	0 1 1 0	6	6 recovery cycles are inserted.	0 1 1 1	7	7 recovery cycles are inserted.	1 0 0 0	8	8 recovery cycles are inserted.	1 0 0 1	9	9 recovery cycles are inserted.	1 0 1 0	10	10 recovery cycles are inserted.	1 0 1 1	11	11 recovery cycles are inserted.	1 1 0 0	12	12 recovery cycles are inserted.	1 1 0 1	13	13 recovery cycles are inserted.	1 1 1 0	14	14 recovery cycles are inserted.	1 1 1 1	15	15 recovery cycles are inserted.	R/W
b3	b0																																																						
0 0 0 0	0	No recovery cycle is inserted.																																																					
0 0 0 1	1	1 recovery cycle is inserted.																																																					
0 0 1 0	2	2 recovery cycles are inserted.																																																					
0 0 1 1	3	3 recovery cycles are inserted.																																																					
0 1 0 0	4	4 recovery cycles are inserted.																																																					
0 1 0 1	5	5 recovery cycles are inserted.																																																					
0 1 1 0	6	6 recovery cycles are inserted.																																																					
0 1 1 1	7	7 recovery cycles are inserted.																																																					
1 0 0 0	8	8 recovery cycles are inserted.																																																					
1 0 0 1	9	9 recovery cycles are inserted.																																																					
1 0 1 0	10	10 recovery cycles are inserted.																																																					
1 0 1 1	11	11 recovery cycles are inserted.																																																					
1 1 0 0	12	12 recovery cycles are inserted.																																																					
1 1 0 1	13	13 recovery cycles are inserted.																																																					
1 1 1 0	14	14 recovery cycles are inserted.																																																					
1 1 1 1	15	15 recovery cycles are inserted.																																																					
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W																																																			
b11 to b8	WRCV[3:0]	Write Recovery	<table border="0"> <tr> <td>b11</td> <td>b8</td> <td></td> </tr> <tr> <td>0 0 0 0</td> <td>0</td> <td>No recovery cycle is inserted.</td> </tr> <tr> <td>0 0 0 1</td> <td>1</td> <td>1 recovery cycle is inserted.</td> </tr> <tr> <td>0 0 1 0</td> <td>2</td> <td>2 recovery cycles are inserted.</td> </tr> <tr> <td>0 0 1 1</td> <td>3</td> <td>3 recovery cycles are inserted.</td> </tr> <tr> <td>0 1 0 0</td> <td>4</td> <td>4 recovery cycles are inserted.</td> </tr> <tr> <td>0 1 0 1</td> <td>5</td> <td>5 recovery cycles are inserted.</td> </tr> <tr> <td>0 1 1 0</td> <td>6</td> <td>6 recovery cycles are inserted.</td> </tr> <tr> <td>0 1 1 1</td> <td>7</td> <td>7 recovery cycles are inserted.</td> </tr> <tr> <td>1 0 0 0</td> <td>8</td> <td>8 recovery cycles are inserted.</td> </tr> <tr> <td>1 0 0 1</td> <td>9</td> <td>9 recovery cycles are inserted.</td> </tr> <tr> <td>1 0 1 0</td> <td>10</td> <td>10 recovery cycles are inserted.</td> </tr> <tr> <td>1 0 1 1</td> <td>11</td> <td>11 recovery cycles are inserted.</td> </tr> <tr> <td>1 1 0 0</td> <td>12</td> <td>12 recovery cycles are inserted.</td> </tr> <tr> <td>1 1 0 1</td> <td>13</td> <td>13 recovery cycles are inserted.</td> </tr> <tr> <td>1 1 1 0</td> <td>14</td> <td>14 recovery cycles are inserted.</td> </tr> <tr> <td>1 1 1 1</td> <td>15</td> <td>15 recovery cycles are inserted.</td> </tr> </table>	b11	b8		0 0 0 0	0	No recovery cycle is inserted.	0 0 0 1	1	1 recovery cycle is inserted.	0 0 1 0	2	2 recovery cycles are inserted.	0 0 1 1	3	3 recovery cycles are inserted.	0 1 0 0	4	4 recovery cycles are inserted.	0 1 0 1	5	5 recovery cycles are inserted.	0 1 1 0	6	6 recovery cycles are inserted.	0 1 1 1	7	7 recovery cycles are inserted.	1 0 0 0	8	8 recovery cycles are inserted.	1 0 0 1	9	9 recovery cycles are inserted.	1 0 1 0	10	10 recovery cycles are inserted.	1 0 1 1	11	11 recovery cycles are inserted.	1 1 0 0	12	12 recovery cycles are inserted.	1 1 0 1	13	13 recovery cycles are inserted.	1 1 1 0	14	14 recovery cycles are inserted.	1 1 1 1	15	15 recovery cycles are inserted.	R/W
b11	b8																																																						
0 0 0 0	0	No recovery cycle is inserted.																																																					
0 0 0 1	1	1 recovery cycle is inserted.																																																					
0 0 1 0	2	2 recovery cycles are inserted.																																																					
0 0 1 1	3	3 recovery cycles are inserted.																																																					
0 1 0 0	4	4 recovery cycles are inserted.																																																					
0 1 0 1	5	5 recovery cycles are inserted.																																																					
0 1 1 0	6	6 recovery cycles are inserted.																																																					
0 1 1 1	7	7 recovery cycles are inserted.																																																					
1 0 0 0	8	8 recovery cycles are inserted.																																																					
1 0 0 1	9	9 recovery cycles are inserted.																																																					
1 0 1 0	10	10 recovery cycles are inserted.																																																					
1 0 1 1	11	11 recovery cycles are inserted.																																																					
1 1 0 0	12	12 recovery cycles are inserted.																																																					
1 1 0 1	13	13 recovery cycles are inserted.																																																					
1 1 1 0	14	14 recovery cycles are inserted.																																																					
1 1 1 1	15	15 recovery cycles are inserted.																																																					
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W																																																			

Do not write to the CSnREC register while access to the CSn area is in progress.

When the preceding bus access is a separate bus access, CSnREC is valid when the recovery cycle insertion is enabled with the separate bus recovery cycle insertion enable bit (RCVENj (j = 0 to 7)) in CSRECEN. When the preceding bus access is an address/data multiplexed bus access, CSnREC is valid when the recovery cycle insertion is enabled with the multiplexed bus recovery cycle insertion enable bit (RCVENMj) in CSRECEN.

#### **RRCV[3:0] Bits (Read Recovery)**

These bits specify the number of recovery cycles to be inserted after a read access to the external bus.

When the recovery cycle insertion is enabled and a value except 0000b is written to these bits, one to 15 recovery cycles are inserted in the following cases.

- After a read access to the external bus, a read access is made to the external bus in the same area.
- After a read access to the external bus, a read access is made to the external bus in a different area.
- After a read access to the external bus, a write access is made to the external bus in the same area.
- After a read access to the external bus, a write access is made to the external bus in a different area.

#### **WRCV[3:0] Bits (Write Recovery)**

These bits specify the number of recovery cycles to be inserted after a write access to the external bus.

When the recovery cycle insertion is enabled and a value except 0000b is written to these bits, one to 15 recovery cycles are inserted in the following cases.

- After a write access to the external bus, a read access is made to the external bus in the same area.
- After a write access to the external bus, a read access is made to the external bus in a different area.
- After a write access to the external bus, a write access is made to the external bus in the same area.
- After a write access to the external bus, a write access is made to the external bus in a different area.

### 16.3.3 CS Recovery Cycle Insertion Enable Register (CSRECEN)

Address(es): 0008 3880h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	RCVEN M7	RCVEN M6	RCVEN M5	RCVEN M4	RCVEN M3	RCVEN M2	RCVEN M1	RCVEN M0	RCVEN 7	RCVEN 6	RCVEN 5	RCVEN 4	RCVEN 3	RCVEN 2	RCVEN 1	RCVEN 0
Value after reset:	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1	0

Bit	Symbol	Bit Name	Description	R/W
b0	RCVEN0	Separate Bus Recovery Cycle Insertion Enable	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b1	RCVEN1	Separate Bus Recovery Cycle Insertion Enable	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b2	RCVEN2	Separate Bus Recovery Cycle Insertion Enable 2	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b3	RCVEN3	Separate Bus Recovery Cycle Insertion Enable 3	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b4	RCVEN4	Separate Bus Recovery Cycle Insertion Enable 4	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b5	RCVEN5	Separate Bus Recovery Cycle Insertion Enable 5	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b6	RCVEN6	Separate Bus Recovery Cycle Insertion Enable 6	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b7	RCVEN7	Separate Bus Recovery Cycle Insertion Enable 7	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b8	RCVENM0	Multiplexed Bus Recovery Cycle Insertion Enable 0	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b9	RCVENM1	Multiplexed Bus Recovery Cycle Insertion Enable 1	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b10	RCVENM2	Multiplexed Bus Recovery Cycle Insertion Enable 2	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b11	RCVENM3	Multiplexed Bus Recovery Cycle Insertion Enable 3	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b12	RCVENM4	Multiplexed Bus Recovery Cycle Insertion Enable 4	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b13	RCVENM5	Multiplexed Bus Recovery Cycle Insertion Enable 5	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b14	RCVENM6	Multiplexed Bus Recovery Cycle Insertion Enable 6	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W
b15	RCVENM7	Multiplexed Bus Recovery Cycle Insertion Enable 7	0: Recovery cycle insertion is disabled. 1: Recovery cycle insertion is enabled.	R/W

Do not write to the CSRECEN register while access to the CSn area is in progress.

#### RCVENn Bit (Separate Bus Recovery Cycle Insertion Enable 0) (n = 0 to 7)

This bit enables or disables the insertion of read recovery cycles when, after a read access to the external bus, a read access is made to the external bus in the same area.

#### RCVENMn Bit (Multiplexed Bus Recovery Cycle Insertion Enable 0) (n = 0 to 7)

This bit enables or disables the insertion of read recovery cycles when, after a read access to the external bus, a read access is made to the external bus in the same area.

**Table 16.7 Insertion of Recovery Cycles**

<b>Access Type</b>	<b>External Address Space</b>	<b>Insertion of Recovery Cycles</b>	<b>Corresponding Bits (Separate/Multiplexed)</b>
Read access after read access	Same area	Recovery cycles specified by the RRCV[3:0] bits are inserted.	RCVEN0/RCVENM0
	Different area	Recovery cycles specified by the RRCV[3:0] bits are inserted.	RCVEN1/RCVENM1
Read access after write access	Same area	Recovery cycles specified by the RRCV[3:0] bits are inserted.	RCVEN2/RCVENM2
	Different area	Recovery cycles specified by the RRCV[3:0] bits are inserted.	RCVEN3/RCVENM3
Write access after read access	Same area	Recovery cycles specified by the WRCV[3:0] bits are inserted.	RCVEN4/RCVENM4
	Different area	Recovery cycles specified by the WRCV[3:0] bits are inserted.	RCVEN5/RCVENM5
Write access after write access	Same area	Recovery cycles specified by the WRCV[3:0] bits are inserted.	RCVEN6/RCVENM6
	Different area	Recovery cycles specified by the WRCV[3:0] bits are inserted.	RCVEN7/RCVENM7



### 16.3.4 CSn Mode Register (CSnMOD) (n = 0 to 7)

Address(es): CS0MOD 0008 3002h, CS1MOD 0008 3012h, CS2MOD 0008 3022h, CS3MOD 0008 3032h,  
CS4MOD 0008 3042h, CS5MOD 0008 3052h, CS6MOD 0008 3062h, CS7MOD 0008 3072h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
PRMOD	—	—	—	—	—	PWENB	PRENB	—	—	—	—	EWENB	—	—	WRMOD
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	WRMOD	Write Access Mode Select	0: Byte strobe mode 1: Single write strobe mode	R/W
b2, b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	EWENB	External Wait Enable	0: External wait is disabled 1: External wait is enabled	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	PRENB	Page Read Access Enable	0: Page read access is disabled 1: Page read access is enabled	R/W
b9	PWENB	Page Write Access Enable	0: Page write access is disabled 1: Page write access is enabled	R/W
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	PRMOD	Page Read Access Mode Select	0: Normal access compatible mode 1: External data read continuous assertion mode	R/W

Do not write to the CSnMOD register while access to the CSn area is in progress.

#### WRMOD Bit (Write Access Mode Select)

This bit selects a write access operating mode.

Writing 0 to this bit selects byte strobe mode where data write operation is controlled by the WRn# (n = 0 to 3) signal corresponding to the respective byte positions.

Writing 1 to this bit selects single write strobe mode where data write operation is controlled by the BCn# (n = 0 to 3) signal and the WR# signal corresponding to respective byte positions. Note that setting the external bus width of 8 bits is prohibited in single write strobe mode.

**Table 16.8 Control Signals for Write Access Mode**

Mode	Pin Name							
	WR3#	WR2#	WR1#	WR0#/WR#	BC3#	BC2#	BC1#	BC0#
Byte strobe mode	○	○	○	○ (WR0#)	×	×	×	×
Single write strobe mode	×	×	×	○ (WR#)	○	○	○	○

○: Enabled, ×: Disabled

#### EWENB Bit (External Wait Enable)

This bit enables or disables external wait.

Writing 1 to this bit selects external wait and allows control of the number of waits in each cycle with the WAIT# signal. In this state, wait cycles are inserted while the WAIT# signal is at the low level.

Writing 0 to this bit disables the WAIT# signal.

**PRENB Bit (Page Read Access Enable)**

This bit enables or disables page read accesses.

Note: • When the address/data multiplexed I/O interface is selected with the MPXEN bit in CSnCR, this bit should not be set to enable page read accesses. Page read accesses are not supported in the address/data multiplexed I/O interface.

**PWENB Bit (Page Write Access Enable)**

This bit enables or disables page write accesses.

Note: • When the address/data multiplexed I/O interface is selected with the MPXEN bit in CSnCR, this bit should not be set to enable page read accesses. Page read accesses are not supported in the address/data multiplexed I/O interface.

**PRMOD Bit (Page Read Access Mode Select)**

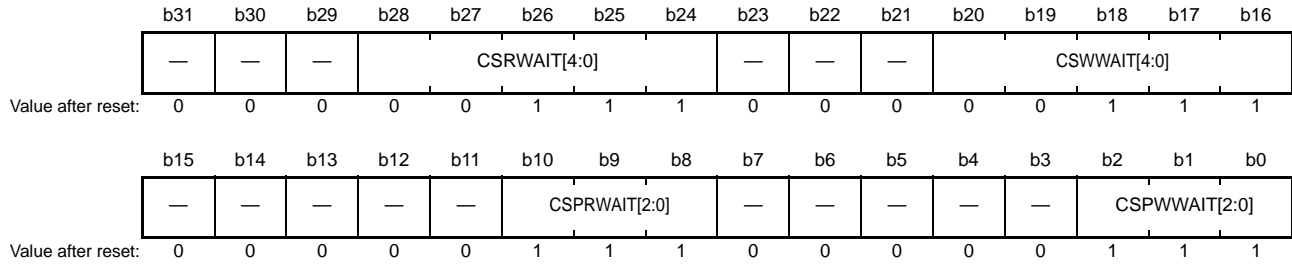
This bit selects a page read access operating mode.

Writing 0 to this bit selects normal access compatible mode where the RD# signal is negated and RD assert wait is inserted each time a piece of data is read. However, when there is no RD assert wait, the RD# signal is negated only in the final transfer of the external bus access.

Writing 1 to this bit selects external data read continuous assertion mode where RD assert wait is inserted and the RD# signal is continuously asserted during this time period.

### 16.3.5 CSn Wait Control Register 1 (CSnWCR1) (n = 0 to 7)

Address(es): CS0WCR1 0008 3004h, CS1WCR1 0008 3014h, CS2WCR1 0008 3024h, CS3WCR1 0008 3034h,  
 CS3WCR1 0008 3044h, CS5WCR1 0008 3054h, CS6WCR1 0008 3064h, CS7WCR1 0008 3074h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CSPWWAIT[2:0]	Page Write Cycle Wait Select*1	b2 b0 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10 to b8	CSPRWAIT[2:0]	Page Read Cycle Wait Select*2	b10 b8 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b20 to b16	CSWWAIT[4:0]	Normal Write Cycle Wait Select	b20      b16 0 0 0 0 0: No wait is inserted. 0 0 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 0 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 0 0 1 1: Wait with a length of 3 clock cycles are inserted. 0 0 1 0 0: Wait with a length of 4 clock cycles are inserted. 0 0 1 0 1: Wait with a length of 5 clock cycles are inserted. 0 0 1 1 0: Wait with a length of 6 clock cycles are inserted. 0 0 1 1 1: Wait with a length of 7 clock cycles are inserted. 0 1 0 0 0: Wait with a length of 8 clock cycles are inserted. 0 1 0 0 1: Wait with a length of 9 clock cycles are inserted. 0 1 0 1 0: Wait with a length of 10 clock cycles are inserted. 0 1 0 1 1: Wait with a length of 11 clock cycles are inserted. 0 1 1 0 0: Wait with a length of 12 clock cycles are inserted. 0 1 1 0 1: Wait with a length of 13 clock cycles are inserted. 0 1 1 1 0: Wait with a length of 14 clock cycles are inserted. 0 1 1 1 1: Wait with a length of 15 clock cycles are inserted. 1 0 0 0 0: Wait with a length of 16 clock cycles are inserted. 1 0 0 0 1: Wait with a length of 17 clock cycles are inserted. 1 0 0 1 0: Wait with a length of 18 clock cycles are inserted. 1 0 0 1 1: Wait with a length of 19 clock cycles are inserted. 1 0 1 0 0: Wait with a length of 20 clock cycles are inserted. 1 0 1 0 1: Wait with a length of 21 clock cycles are inserted. 1 0 1 1 0: Wait with a length of 22 clock cycles are inserted. 1 0 1 1 1: Wait with a length of 23 clock cycles are inserted. 1 1 0 0 0: Wait with a length of 24 clock cycles are inserted. 1 1 0 0 1: Wait with a length of 25 clock cycles are inserted. 1 1 0 1 0: Wait with a length of 26 clock cycles are inserted. 1 1 0 1 1: Wait with a length of 27 clock cycles are inserted. 1 1 1 0 0: Wait with a length of 28 clock cycles are inserted. 1 1 1 0 1: Wait with a length of 29 clock cycles are inserted. 1 1 1 1 0: Wait with a length of 30 clock cycles are inserted. 1 1 1 1 1: Wait with a length of 31 clock cycles are inserted.	R/W
b23 to b21	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b28 to b24	CSRWAIT[4:0]	Normal Read Cycle Wait Select	b28      b24 0 0 0 0 0: No wait is inserted. 0 0 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 0 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 0 0 1 1: Wait with a length of 3 clock cycles are inserted. 0 0 1 0 0: Wait with a length of 4 clock cycles are inserted. 0 0 1 0 1: Wait with a length of 5 clock cycles are inserted. 0 0 1 1 0: Wait with a length of 6 clock cycles are inserted. 0 0 1 1 1: Wait with a length of 7 clock cycles are inserted. 0 1 0 0 0: Wait with a length of 8 clock cycles are inserted. 0 1 0 0 1: Wait with a length of 9 clock cycles are inserted. 0 1 0 1 0: Wait with a length of 10 clock cycles are inserted. 0 1 0 1 1: Wait with a length of 11 clock cycles are inserted. 0 1 1 0 0: Wait with a length of 12 clock cycles are inserted. 0 1 1 0 1: Wait with a length of 13 clock cycles are inserted. 0 1 1 1 0: Wait with a length of 14 clock cycles are inserted. 0 1 1 1 1: Wait with a length of 15 clock cycles are inserted. 1 0 0 0 0: Wait with a length of 16 clock cycles are inserted. 1 0 0 0 1: Wait with a length of 17 clock cycles are inserted. 1 0 0 1 0: Wait with a length of 18 clock cycles are inserted. 1 0 0 1 1: Wait with a length of 19 clock cycles are inserted. 1 0 1 0 0: Wait with a length of 20 clock cycles are inserted. 1 0 1 0 1: Wait with a length of 21 clock cycles are inserted. 1 0 1 1 0: Wait with a length of 22 clock cycles are inserted. 1 0 1 1 1: Wait with a length of 23 clock cycles are inserted. 1 1 0 0 0: Wait with a length of 24 clock cycles are inserted. 1 1 0 0 1: Wait with a length of 25 clock cycles are inserted. 1 1 0 1 0: Wait with a length of 26 clock cycles are inserted. 1 1 0 1 1: Wait with a length of 27 clock cycles are inserted. 1 1 1 0 0: Wait with a length of 28 clock cycles are inserted. 1 1 1 0 1: Wait with a length of 29 clock cycles are inserted. 1 1 1 1 0: Wait with a length of 30 clock cycles are inserted. 1 1 1 1 1: Wait with a length of 31 clock cycles are inserted.	R/W
b31 to b29	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. The CSPWWAIT[2:0] value is valid only when the PWENB bit in CSnMOD is set to 1.

Note 2. The CSPRWAIT[2:0] value is valid only when the PRENB bit in CSnMOD is set to 1.

Do not write to the CSnWCR1 register while access to the CSn area is in progress.

Set each of these bits within a range of the restrictions described in section 16.5.7 (1) Limitations on Using Separate Bus Interface or section 16.5.7 (2) Limitations on Using Address/Data Multiplexed Bus Interface, according to the bus interface used.

#### **CSPWAIT[2:0] Bits (Page Write Cycle Wait Select)**

These bits specify the number of wait cycles to be inserted into the second and subsequent accesses during a page write cycle.

This setting is enabled when the PWENB bit in CSnMOD is set to 1.

Note: • Be sure to satisfy  $1 \leq \text{CSnWCR2.WDON}[2:0] \text{ value} \leq \text{CSnWCR2.WRON}[2:0] \text{ value} \leq \text{CSnWCR1.CSPWAIT}[2:0] \text{ value}$  and  $\text{CSnWCR2.CSON}[2:0] \text{ value} \leq \text{CSnWCR2.WRON}[2:0] \text{ value} \leq \text{CSnWCR1.CSPWAIT}[2:0] \text{ value}$ .

#### **CSPRWAIT[2:0] Bits (Page Read Cycle Wait Select)**

These bits specify the number of wait cycles to be inserted into the second and subsequent accesses during a page read cycle.

This setting is enabled when the PRENB bit in CSnMOD is set to 1.

Note: • Be sure to satisfy  $\text{CSnWCR2.CSON}[2:0] \text{ value} \leq \text{CSnWCR2.RDON}[2:0] \text{ value} \leq \text{CSnWCR1.CSPRWAIT}[2:0] \text{ value}$ .

#### **CSWAIT[4:0] Bits (Normal Write Cycle Wait Select)**

These bits specify the number of wait cycles to be inserted into the first access during a normal write cycle or page write cycle.

Note: • Be sure to satisfy  $1 \leq \text{CSnWCR2.WDON}[2:0] \text{ value} \leq \text{CSnWCR2.WRON}[2:0] \text{ value} \leq \text{CSnWCR1.CSWAIT}[4:0] \text{ value}$  and  $\text{CSnWCR2.CSON}[2:0] \text{ value} \leq \text{CSnWCR2.WRON}[2:0] \text{ value} \leq \text{CSnWCR1.CSWAIT}[4:0] \text{ value}$ .

#### **CSRWAIT[4:0] Bits (Normal Read Cycle Wait Select)**

These bits specify the number of wait cycles to be inserted into the first access during a normal read cycle or page read cycle.

Note: • Be sure to satisfy  $\text{CSnWCR2.CSON}[2:0] \text{ value} \leq \text{CSnWCR2.RDON}[2:0] \text{ value} \leq \text{CSnWCR1.CSRWAIT}[4:0] \text{ value}$ .

### 16.3.6 CSn Wait Control Register 2 (CSnWCR2) (n = 0 to 7)

Address(es): CS0WCR2 0008 3008h, CS1WCR2 0008 3018h, CS2WCR2 0008 3028h, CS3WCR2 0008 3038h,  
CS4WCR2 0008 3048h, CS5WCR2 0008 3058h, CS6WCR2 0008 3068h, CS7WCR2 0008 3078h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
—	CSO[N[2:0]]			—	WDO[N[2:0]]			—	WRO[N[2:0]]			—	RDO[N[2:0]]		
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
—	—	AWAIT[1:0]		—	WDOFF[2:0]			—	CSWOFF[2:0]			—	CSROFF[2:0]		
Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1															

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CSROFF[2:0]	Read-Access CS Extension Cycle Select	b2 b0 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6 to b4	CSWOFF[2:0]	Write-Access CS Extension Cycle Select	b6 b4 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	WDOFF[2:0]	Write Data Output Extension Cycle Select	b10 b8 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13, b12	AWAIT[1:0]	Address Cycle Wait Select	b13 b12 0 0: No wait is inserted. 0 1: Wait with a length of 1 clock cycle is inserted. 1 0: Wait with a length of 2 clock cycles are inserted. 1 1: Wait with a length of 3 clock cycles are inserted.	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b18 to b16	RDO[N[2:0]]	RD Assert Wait Select	b18 b16 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b19	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Bit	Symbol	Bit Name	Description	R/W
b22 to b20	WRON[2:0]	WR Assert Wait Select	b22 b20 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b23	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b26 to b24	WDON[2:0]	Write Data Output Wait Select	b26 b24 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b30 to b28	CSON[2:0]	CS Assert Wait Select	b30 b28 0 0 0: No wait is inserted. 0 0 1: Wait with a length of 1 clock cycle is inserted. 0 1 0: Wait with a length of 2 clock cycles are inserted. 0 1 1: Wait with a length of 3 clock cycles are inserted. 1 0 0: Wait with a length of 4 clock cycles are inserted. 1 0 1: Wait with a length of 5 clock cycles are inserted. 1 1 0: Wait with a length of 6 clock cycles are inserted. 1 1 1: Wait with a length of 7 clock cycles are inserted.	R/W
b31	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Do not write to the CSnWCR2 register while access to the CSn area is in progress.

Set each of these bits within a range of the restrictions described in section 16.5.7 (1) Limitations on Using Separate Bus Interface or section 16.5.7 (2) Limitations on Using Address/Data Multiplexed Bus Interface, according to the bus interface used.

#### CSROFF[2:0] Bits (Read-Access CS Extension Cycle Select)

These bits specify the number of wait cycles to be inserted in a time period from the end of a wait cycle (RD# signal negated) until the CSn# signal (n = 0 to 7) is negated in read access mode.

#### CSWOFF[2:0] Bits (Write-Access CS Extension Cycle Select)

These bits specify the number of wait cycles to be inserted in a time period from the end of a wait cycle (WRn# signal (n = 0 to 3) negated) until the CSn# signal (n = 0 to 3) is negated in write access mode.

Note: • Be sure to satisfy WDOFF[2:0] value  $\leq$  CSWOFF[2:0] value.

#### WDOFF[2:0] Bits (Write Data Output Extension Cycle Select)

These bits specify the number of wait cycles to be inserted in a time period from the end of a wait cycle (WRn# signal (n = 0 to 3) negated) until the write data output is completed in write access mode.

Note: • Be sure to satisfy WDOFF[2:0] value  $\leq$  CSWOFF[2:0] value.

#### AWAIT[1:0] Bits (Address Cycle Wait Select)

These bits specify the number of wait cycles to be inserted into an address output cycle with the address/data multiplexed I/O interface.

Note: • CSnWCR2.CSON[2:0] value  $\leq$  CSnWCR2.AWAIT[1:0] value

For read access, satisfy CSnWCR2.AWAIT[1:0] value + 2  $\leq$  CSnWCR2.RDON[2:0] value  $\leq$  CSnWCR1.CSRWAIT[4:0] value.

For write access, satisfy CSnWCR2.AWAIT[1:0] value + 2  $\leq$  CSnWCR2.WRON[2:0] value  $\leq$

$CSnWCR1.CSWWAIT[4:0]$  value and  $CSnWCR2.AWAIT[1:0]$  value + 2  $\leq$   $CSnWCR2.WDON[2:0]$  value  $\leq$   $CSnWCR1.CSWWAIT[4:0]$  value.

#### **RDON[2:0] Bits (RD Assert Wait Select)**

These bits specify the number of wait cycles to be inserted before the RD# signal is asserted.

Note: • For normal read access, satisfy  $CSnWCR2.CSON[2:0]$  value  $\leq$   $CSnWCR2.RDON[2:0]$  value  $\leq$   $CSnWCR1.CSRWAIT[4:0]$  value.

For page read access, satisfy  $CSnWCR2.CSON[2:0]$  value  $\leq$   $CSnWCR2.RDON[2:0]$  value  $\leq$   $CSnWCR1.CSPRWAIT[2:0]$  value.

Note: • When the address/data multiplexed I/O interface is selected, satisfy  $CSnWCR2.AWAIT[1:0]$  value + 2  $\leq$   $CSnWCR2.RDON[2:0]$  value  $\leq$   $CSnWCR1.CSRWAIT[4:0]$  value.

#### **WRON[2:0] Bits (WR Assert Wait Select)**

These bits specify the number of wait cycles to be inserted before the WRn# signal (n = 0 to 3) is asserted.

Note: • For normal write access, satisfy  $1 \leq CSnWCR2.WDON[2:0]$  value  $\leq$   $CSnWCR2.WRON[2:0]$  value  $\leq$   $CSnWCR1.CSWWAIT[4:0]$  value and  $CSnWCR2.CSON[2:0]$  value  $\leq$   $CSnWCR2.WRON[2:0]$  value  $\leq$   $CSnWCR1.CSWWAIT[4:0]$  value.

For page write access, satisfy  $1 \leq CSnWCR2.WDON[2:0]$  value  $\leq$   $CSnWCR2.WRON[2:0]$  value  $\leq$   $CSnWCR1.CSPWAIT[2:0]$  value and  $CSnWCR2.CSON[2:0]$  value  $\leq$   $CSnWCR2.WRON[2:0]$  value  $\leq$   $CSnWCR1.CSPWAIT[2:0]$  value.

Note: • When the address/data multiplexed I/O interface is selected, satisfy  $CSnWCR2.AWAIT[1:0]$  value + 2  $\leq$   $CSnWCR2.WRON[2:0]$  value  $\leq$   $CSnWCR1.CSWWAIT[4:0]$  value.

#### **WDON[2:0] Bits (Write Data Output Wait Select)**

These bits specify the number of wait cycles to be inserted before the write data is output.

Note: • For normal write access, satisfy  $1 \leq CSnWCR2.WDON[2:0]$  value  $\leq$   $CSnWCR2.WRON[2:0]$  value  $\leq$   $CSnWCR1.CSWWAIT[4:0]$  value.

For page write access, satisfy  $1 \leq CSnWCR2.WDON[2:0]$  value  $\leq$   $CSnWCR2.WRON[2:0]$  value  $\leq$   $CSnWCR1.CSPWAIT[2:0]$  value.

Note: • When the address/data multiplexed I/O interface is selected, satisfy  $CSnWCR2.AWAIT[1:0]$  value + 2  $\leq$   $CSnWCR2.WDON[2:0]$  value  $\leq$   $CSnWCR1.CSWWAIT[4:0]$  value.

#### **CSON[2:0] Bits (CS Assert Wait Select)**

These bits specify the number of wait cycles to be inserted before the CSn# signal (n = 0 to 7) is asserted.

Note: • For normal read access, satisfy  $CSnWCR2.CSON[2:0]$  value  $\leq$   $CSnWCR2.RDON[2:0]$  value  $\leq$   $CSnWCR1.CSRWAIT[4:0]$  value.

For page read access, satisfy  $CSnWCR2.CSON[2:0]$  value  $\leq$   $CSnWCR2.RDON[2:0]$  value  $\leq$   $CSnWCR1.CSPRWAIT[2:0]$  value.

For normal write access, satisfy  $CSnWCR2.CSON[2:0]$  value  $\leq$   $CSnWCR2.WRON[2:0]$  value  $\leq$   $CSnWCR1.CSWWAIT[4:0]$  value.

For page write access, satisfy  $CSnWCR2.CSON[2:0]$  value  $\leq$   $CSnWCR2.WRON[2:0]$  value  $\leq$   $CSnWCR1.CSPWAIT[2:0]$  value.

Note: • When the address/data multiplexed I/O interface is selected, satisfy  $CSnWCR2.CSON[2:0]$  value  $\leq$   $CSnWCR2.AWAIT[1:0]$  value.



### 16.3.7 Bus Error Status Clear Register (BERCLR)

Address(es): 0008 1300h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	STSCLR
0	0	0	0	0	0	0	0

Value after reset:

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.

### 16.3.8 Bus Error Monitoring Enable Register (BEREN)

Address(es): 0008 1304h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	TOEN	IGAEN
0	0	0	0	0	0	0	0

Value after reset:

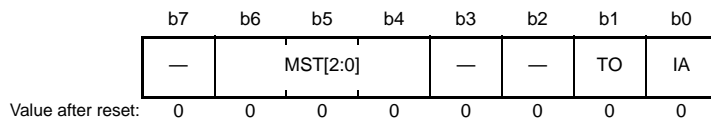
Bit	Symbol	Bit Name	Description	R/W
b0	IGAEN	Illegal Address Access Detection Enable	0: Illegal address access detection is disabled. 1: Illegal address access detection is enabled.	R/W
b1	TOEN	Timeout Detection Enable*1,*2	0: Bus timeout detection is disabled. 1: Bus timeout detection is enabled.	R/W
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 clear the TOEN bit to 0 (bus timeout detection disabled) while timeout errors are being detected.

### 16.3.9 Bus Error Status Register 1 (BERSR1)

Address(es): 0008 1308h



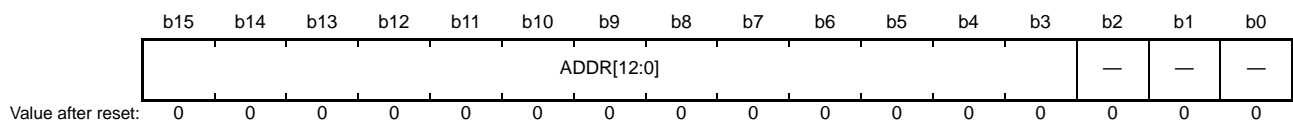
Bit	Symbol	Bit Name	Description	R/W																											
b0	IA	Illegal Address Access	0: Illegal address access not made 1: Illegal address access made	R																											
b1	TO	Timeout	0: Timeout not generated 1: Timeout generated	R																											
b3, b2	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R																											
b6 to b4	MST[2:0]	Bus Master Code	<table style="font-size: small; border: none;"> <tr> <td style="padding-right: 10px;">b6</td> <td style="padding-right: 10px;">b4</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0: CPU</td> </tr> <tr> <td>0</td> <td>0</td> <td>1: Reserved</td> </tr> <tr> <td>0</td> <td>1</td> <td>0: Reserved</td> </tr> <tr> <td>0</td> <td>1</td> <td>1: DTC/DMAC</td> </tr> <tr> <td>1</td> <td>0</td> <td>0: Reserved</td> </tr> <tr> <td>1</td> <td>0</td> <td>1: Reserved</td> </tr> <tr> <td>1</td> <td>1</td> <td>0: Reserved</td> </tr> <tr> <td>1</td> <td>1</td> <td>1: Reserved</td> </tr> </table>	b6	b4		0	0	0: CPU	0	0	1: Reserved	0	1	0: Reserved	0	1	1: DTC/DMAC	1	0	0: Reserved	1	0	1: Reserved	1	1	0: Reserved	1	1	1: Reserved	R
b6	b4																														
0	0	0: CPU																													
0	0	1: Reserved																													
0	1	0: Reserved																													
0	1	1: DTC/DMAC																													
1	0	0: Reserved																													
1	0	1: Reserved																													
1	1	0: Reserved																													
1	1	1: Reserved																													
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.

### 16.3.10 Bus Error Status Register 2 (BERSR2)

Address(es): 0008 130Ah



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

### 16.3.11 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
—	—	BPEB[1:0]	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 Bus 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
b13, b12	BPEB[1:0]	External Bus Priority Control	b13 b12 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, b14	—	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 and DMAC 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 Bus 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.

**BPEB[1:0] Bits (External Bus Priority Control)**

These bits specify the priority order for the external bus.  
When the priority order is fixed, the order is internal main bus 2, and then internal main bus 1.  
When the priority order is toggled, a bus has a lower priority when the request of that bus is accepted, between internal main bus 1 and internal main bus 2.

## 16.4 Endian and Data Alignment

The external bus has a data-alignment function to control which byte of the data bus (D31 to D24, D23 to D16, D15 to D8, or D7 to D0) is used according to the bus specifications of the area to be accessed (8-bit, 16-bit, or 32-bit bus space), data size, and endian format when accessing the external address space (the CS).

### 16.4.1 Data Alignment Control for CS Area

#### (1) 32-Bit Bus Space

When a 32-bit width is selected for a bus space by the BSIZE[1:0] bits in CSnCR, the address buses A23 to A2 are enabled to output address signals in units of 32 bits, and the address buses A1 and A0 are disabled (always output the low level).

When byte strobe mode is selected (the WRMOD bit = 0 in CSnMOD), the WR0# to WR3# pins are enabled. The BC0# to BC3# pins are not used.

When single write strobe mode is selected (the WRMOD bit = 1 in CSnMOD), only the WR0# pin is enabled and always outputs the low level during write access, regardless of the data size. Here, the WR1# to WR3# pins are invalid (always output the high level). The valid byte position is indicated by the BC0# to BC3# pins.

In 32-bit bus space, the valid positions of data external to the chip and of control signals differ according to whether the endian is big or little.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	Control Signals													
						WR3#/BC3#	WR2#/BC2#	WR1#/BC1#	WR0#/BC0#										
						RD#													
						Data Bus													
						D31	D24	D23	D16	D15	D8	D7	D0						
8 bits	4n	One	First	8 bits	4n							7	0						
	4n+1	One	First	8 bits	4n							7	0						
	4n+2	One	First	8 bits	4n							7	0						
	4n+3	One	First	8 bits	4n							7	0						
16 bits	4n	One	First	16 bits	4n							15	8	7	0				
	4n+1	Two	First	8 bits	4n								7	0					
			Second	8 bits	4n								15	8					
	4n+2	One	First	16 bits	4n							15	8	7	0				
	4n+3	Two	First	8 bits	4n								7	0					
			Second	8 bits	4n+4									15	8				
32 bits	4n	One	First	32 bits	4n							31	24	23	16	15	8	7	0
	4n+1	Three	First	8 bits	4n								7	0					
			Second	16 bits	4n									23	16	15	8		
			Third	8 bits	4n+4									31	24				
	4n+2	Two	First	16 bits	4n								15	8	7	0			
			Second	16 bits	4n+4									31	24	23	16		
	4n+3	Three	First	8 bits	4n								7	0					
			Second	16 bits	4n+4									23	16	15	8		
			Third	8 bits	4n+4									31	24				

Figure 16.6 Data Alignment (Little Endian) in 32-Bit Bus Space

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	WR3#/BC3#	WR2#/BC2#	WR1#/BC1#	WR0#/BC0#				
						RD#							
Data Bus						D31	D24	D23	D16	D15	D8	D7	D0
8 bits	4n	One	First	8 bits	4n	[7 0]							
	4n+1	One	First	8 bits	4n	[7 0]							
	4n+2	One	First	8 bits	4n	[7 0]							
	4n+3	One	First	8 bits	4n	[7 0]							
16 bits	4n	One	First	16 bits	4n	[15 8   7 0]							
	4n+1	Two	First	8 bits	4n	[15 8]							
			Second	8 bits	4n	[7 0]							
	4n+2	One	First	16 bits	4n	[15 8   7 0]							
	4n+3	Two	First	8 bits	4n	[15 8]							
			Second	8 bits	4n+4	[7 0]							
32 bits	4n	One	First	32 bits	4n	[31 24   23 16   15 8   7 0]							
	4n+1	Three	First	8 bits	4n	[31 24]							
			Second	16 bits	4n	[23 16   15 8]							
			Third	8 bits	4n+4	[7 0]							
	4n+2	Two	First	16 bits	4n	[31 24   23 16]							
			Second	16 bits	4n+4	[15 8   7 0]							
	4n+3	Three	First	8 bits	4n	[31 24]							
			Second	16 bits	4n+4	[23 16   15 8]							
			Third	8 bits	4n+4	[7 0]							

Figure 16.7 Data Alignment (Big Endian) in 32-Bit Bus Space

(2) 16-Bit Bus Space

When a 16-bit width is selected for a bus space by the BSIZE[1:0] bits in CSnCR, the address buses A23 to A1 are enabled to output address signals in units of 16 bits, and the address bus A0 is disabled (always output the low level).

When byte strobe mode is selected (the WRMOD bit = 0 in CSnMOD), the WR0# and WR1# pins are enabled, and the WR2# and WR3# pins are disabled (fixed high). The BC0#, and BC1# to BC3# pins are not used.

When single write strobe mode is selected (the WRMOD bit = 1 in CSnMOD), only the WR0# pin is enabled and always outputs the low level during write access, regardless of the data size. Here, the WR1# to WR3# pins are invalid (always output the high level). The valid byte position is indicated by the BC0# and BC1# pins. The WR2# and WR3# pins are not used.

In 16-bit bus space, page access can occur in access to data in 32-bit units. Specifically, page access can occur when an access does not spread over a 32-bit boundary and causes no change in BC0# and BC1# signals. The situations in which page access occurs are indicated by the letter (p) in Figure 16.8 and Figure 16.9.

In 16-bit bus space, the valid positions of data external to the chip and of control signals differ according to whether the endian is big or little.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	Data Bus			
						D15	D8	D7	D0
8 bits	4n	One	First	8 bits	4n	[ 7   0 ]			
	4n+1	One	First	8 bits	4n	[ 7   0 ]			
	4n+2	One	First	8 bits	4n+2	[ 7   0 ]			
	4n+3	One	First	8 bits	4n+2	[ 7   0 ]			
16 bits	4n	One	First	16 bits	4n	[ 15   8   7   0 ]			
	4n+1	Two	First	8 bits	4n	[ 7   0 ]			
			Second	8 bits	4n+2	[ 15   8 ]			
	4n+2	One	First	16 bits	4n+2	[ 15   8   7   0 ]			
4n+3	Two	First	8 bits	4n+2	[ 7   0 ]				
		Second	8 bits	4n+4	[ 15   8 ]				
32 bits	4n	Two	First	16 bits	4n	[ 15   8   7   0 ]			
			Second	16 bits	4n+2 (p)	[ 31   24   23   16 ]			
	4n+1	Three	First	8 bits	4n	[ 7   0 ]			
			Second	16 bits	4n+2	[ 23   16   15   8 ]			
			Third	8 bits	4n+4	[ 31   24 ]			
	4n+2	Two	First	16 bits	4n+2	[ 15   8   7   0 ]			
			Second	16 bits	4n+4	[ 31   24   23   16 ]			
	4n+3	Three	First	8 bits	4n+2	[ 7   0 ]			
Second			16 bits	4n+4	[ 23   16   15   8 ]				
Third			8 bits	4n+6	[ 31   24 ]				

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 16.8 Data Alignment (Little Endian) in 16-Bit Bus Space

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	WR1#/BC1#	WR0#/BC0#	
						RD#		
						Data Bus		
						D15	D8   D7	D0
8 bits	4n	One	First	8 bits	4n	7	0	
	4n+1	One	First	8 bits	4n		7   0	
	4n+2	One	First	8 bits	4n+2	7	0	
	4n+3	One	First	8 bits	4n+2		7   0	
16 bits	4n	One	First	16 bits	4n	15	8   7   0	
	4n+1	Two	First	8 bits	4n		15   8	
			Second	8 bits	4n+2	7	0	
	4n+2	One	First	16 bits	4n+2	15	8   7   0	
4n+3	Two	First	8 bits	4n+2		15   8		
		Second	8 bits	4n+4	7	0		
32 bits	4n	Two	First	16 bits	4n	31	24   23   16	
			Second	16 bits	4n+2 (p)	15	8   7   0	
	4n+1	Three	First	8 bits	4n		31   24	
			Second	16 bits	4n+2	23	16   15   8	
			Third	8 bits	4n+4	7	0	
	4n+2	Two	First	16 bits	4n+2	31	24   23   16	
			Second	16 bits	4n+4	15	8   7   0	
	4n+3	Three	First	8 bits	4n+2		31   24	
Second			16 bits	4n+4	23	16   15   8		
Third			8 bits	4n+6	7	0		

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 16.9 Data Alignment (Big Endian) in 16-Bit Bus Space



### (3) 8-Bit Bus Space

When an 8-bit bus space is selected by the BSIZE[1:0] bits in CSnCR, the address buses A23 to A0 are enabled to output address signals in byte units.

In 8-bit bus space, only the WR0# pin is valid regardless of write access mode, and always outputs the low level during write access. The WR1# to WR3# pins and the BC0# to BC3# pins are not used.

Page access can occur in access to data in 16- or 32-bit units. Specifically, page access can occur when an access does not spread over a 32-bit boundary. The situations in which page access occurs are indicated by the letter (p) in Figure 16.10 and Figure 16.11.

In 8-bit bus space, the valid positions of data external to the chip are D7 to D0 and WR0# is used as the control signal, regardless of the endian mode.

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	Data Bus			
						D15	D8	D7	D0
						WR1#/BC1#		WR0#/BC0#	
						RD#			
						Data Bus			
						D15	D8	D7	D0
8 bits	4n	One	First	8 bits	4n	7		0	
	4n+1	One	First	8 bits	4n+1	7		0	
	4n+2	One	First	8 bits	4n+2	7		0	
	4n+3	One	First	8 bits	4n+3	7		0	
16 bits	4n	Two	First	8 bits	4n	7		0	
			Second	8 bits	4n+1 (p)	15		8	
	4n+1	Two	First	8 bits	4n+1	7		0	
			Second	8 bits	4n+2 (p)	15		8	
	4n+2	Two	First	8 bits	4n+2	7		0	
			Second	8 bits	4n+3 (p)	15		8	
	4n+3	Two	First	8 bits	4n+3	7		0	
			Second	8 bits	4n+4	15		8	
32 bits	4n	Four	First	8 bits	4n	7		0	
			Second	8 bits	4n+1 (p)	15		8	
			Third	8 bits	4n+2 (p)	23		16	
			Fourth	8 bits	4n+3 (p)	31		24	
	4n+1	Four	First	8 bits	4n+1	7		0	
			Second	8 bits	4n+2 (p)	15		8	
			Third	8 bits	4n+3 (p)	23		16	
			Fourth	8 bits	4n+4	31		24	
	4n+2	Four	First	8 bits	4n+2	7		0	
			Second	8 bits	4n+3 (p)	15		8	
			Third	8 bits	4n+4	23		16	
			Fourth	8 bits	4n+5 (p)	31		24	
	4n+3	Four	First	8 bits	4n+3	7		0	
			Second	8 bits	4n+4	15		8	
			Third	8 bits	4n+5 (p)	23		16	
			Fourth	8 bits	4n+6 (p)	31		24	

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 16.10 Data Alignment (Little Endian) in 8-Bit Bus Space

Data Size	Access Address	Number of Access	Bus Cycle	Unit of Data	Address	WR1#/BC1#   WR0#/BC0#		RD#		Data Bus				
						D15	D8	D7	D0					
8 bits	4n	One	First	8 bits	4n					7			0	
	4n+1	One	First	8 bits	4n+1					7			0	
	4n+2	One	First	8 bits	4n+2					7			0	
	4n+3	One	First	8 bits	4n+3					7			0	
16 bits	4n	Two	First	8 bits	4n					15			8	
			Second	8 bits	4n+1 (p)						7			0
	4n+1	Two	First	8 bits	4n+1					15			8	
			Second	8 bits	4n+2 (p)						7			0
	4n+2	Two	First	8 bits	4n+2					15			8	
			Second	8 bits	4n+3 (p)						7			0
	4n+3	Two	First	8 bits	4n+3					15			8	
			Second	8 bits	4n+4						7			0
32 bits	4n	Four	First	8 bits	4n					31			24	
			Second	8 bits	4n+1 (p)						23			16
			Third	8 bits	4n+2 (p)						15			8
			Fourth	8 bits	4n+3 (p)						7			0
	4n+1	Four	First	8 bits	4n+1					31			24	
			Second	8 bits	4n+2 (p)						23			16
			Third	8 bits	4n+3 (p)						15			8
			Fourth	8 bits	4n+4						7			0
	4n+2	Four	First	8 bits	4n+2					31			24	
			Second	8 bits	4n+3 (p)						23			16
			Third	8 bits	4n+4						15			8
			Fourth	8 bits	4n+5 (p)						7			0
	4n+3	Four	First	8 bits	4n+3					31			24	
			Second	8 bits	4n+4						23			16
			Third	8 bits	4n+5 (p)						15			8
			Fourth	8 bits	4n+6 (p)						7			0

(p): Page access (only when page access is enabled with the PRENB and PWENB bits in CSnMOD)

Figure 16.11 Data Alignment (Big Endian) in 8-Bit Bus Space

## 16.5 Operation of CS Area Controller

### 16.5.1 Separate Bus

The various periods in the timing charts are described below.

The CS area controller (CSC) operates in synchronization with the external bus clock (BCLK). The operation cycles, such as wait cycles specified with the CSC register, are counted on BCLK. In the following description, frequencies of BCLK and BCLK pin output are the same, unless otherwise noted.

Access via the external bus starts at the same point as the output of a rising edge on the BCLK pin. However, if the external bus clock (BCLK) and the output on the BCLK pin are at different frequencies so that a single request from a bus master for transfer leads to two or more rounds of access via the external bus, the wait settings may cause the start of access for the second and subsequent rounds to coincide with the falling edge of the output on the BCLK pin (see Figure 16.19 to Figure 16.23). If recovery cycles are inserted for bus access, the setting for the number of recovery cycles may also cause the start of access for the second and subsequent rounds to coincide with the falling edge of the output on the BCLK pin (see Figure 16.39).

#### (a) Tw1 to Twn (Clock Cycles of Waiting for a Normal Read Cycle or Normal Write Cycle)

The period Tw1 to Twn is made up of the number of clock cycles between the start of access via the external bus clock and one cycle before the strobe signal is valid. The number of cycles are selectable within the range from zero to 31.

Within this period, the timing of CSn#, RD#, and WRn# assertion (placing the signals at the low level) is determined by the respective wait settings. Specifically, the periods of waiting are controlled by the CS assert wait select bits (CSON), the RD assert wait select bits (RDON), the WR assert wait select bits (WRON), and the write data output wait select bits (WDON) in CSn wait control register 2 (CSnWCR2). The number of clock cycles for each of these periods of waiting is selectable as a value from zero to seven counted from the start of external bus access. Selectable numbers of cycles are also within the overall number of clock cycles of waiting for reading or writing.

#### (b) Tend (Clock Cycle where the Strobe Signal is Valid)

Tend is the next clock cycle after completion of the period of waiting for a normal cycle of reading or writing or for a cycle of page reading or page writing. If each wait select bit for a normal cycle of reading or writing or for a cycle of page reading or page writing is zero, the clock cycle where bus access starts is the clock cycle where the strobe signal is valid. The RD# and WRn# signals are negated in the next clock cycle after the cycle where the strobe signal is valid. In the case of read access, the clock cycle where the strobe signal is valid becomes the clock cycle where the data to be read are sampled.

If an external wait is enabled, the wait signal is sampled at the time of the cycle where the strobe signal is valid. The bus cycle is extended if the wait signal is at the low level. The bus cycle is completed in the next clock cycle if the wait signal is at the high level. Tend indicates the cycle where sampling of the wait signal starts.

After the first cycle where the strobe signal is valid during page access, second and subsequent page access operations (point (e) below) start in the next cycle except in cases of write access where a setting (other than zero) for write-data output extension clock cycles (point (d) below) has been made. If the setting for the RD or WR assertion wait is a value other than zero, the RD# and WRn# signals are negated in the next clock cycle. If the setting is zero, assertion continues. Furthermore, the CSn# signal continues to be asserted rather than being negated.

#### (c) Tn1 to Tnm (Clock Cycles of CS Extension)

In the case of normal access, Tn1 to Tnm represent the clock cycles of the period following the cycle where the strobe signal is valid (Tend) up to negation of the CSn# signal. For read or write access, the timing of negation can be controlled by the read-access CS extension cycle select bits (CSROFF) and the write-access CS extension cycle select bits (CSWOFF) in the CSn wait control register 2 (CSnWCR2), respectively.

The number of cycles are counted from the cycle following the cycle where the strobe signal is valid.

In the case of page access, Tn1 to Tnm represent the clock cycles of the period for the cycle following the last cycle where the strobe signal is valid up to negation of the CSn# signal.

For write access, setting the write data output extension cycle select bits (WDOFF) controls extension of the period where the address and output data are valid.

#### (d) Tdw1 to Tdwn (Write-Data Output Extension Clock Cycles)

For write access, if the setting for write-data output extension wait is a value other than zero, clock cycles of write-data output extension are inserted from the cycle that follows the cycle where the strobe signal is valid (Tend).

In the case of normal access, this is inserted within the period of clock cycles of CS extension (point (c) above).

In the case of page access, this is inserted within the period of the cycle where the strobe signal is valid and subsequent page access or within the period of clock cycles of CS extension (point (c) above). Valid address and data output are extended over this period, and the WRn# signal is negated.

#### (e) Tpw1 to TpwN (Page-Read Cycle Wait or Page-Write Cycle Wait)

For the second and subsequent bus cycles during page access, the values for a page-read cycle wait or page-write cycle wait are used instead of the settings for a normal read or write cycle wait. Setting the WR assert wait select bits becomes enabled in the same way as for the first round of access. How the setting for RD assertion controls operation depends on the setting for page-read access mode (the PRMOD bit in CSnMOD) as described below.

CSnMOD.PRMOD = 0: A wait until RD assertion is inserted in the same way as for the first round of access, and the RD# signal is negated.

CSnMOD.PRMOD = 1: Although a wait until RD assertion is inserted in the same way as for normal-access compatibility mode, the RD# signal continues to be asserted over this period.

#### (f) Tr1 to Trn (Recovery Cycles)

Recovery cycles can be inserted from the point where a bus cycle is completed (CSn# signal negation). The number of recovery cycles can be controlled by the setting of the read recovery (RRCV) or write recovery (WRCV) bits in the CSn recovery cycle register (CSnREC). Both numbers of recovery cycles are counted from the end of a bus cycle (CSn# negation) and can be selected from 0 to 15 cycles. For details on recovery cycles, see section 16.5.4, Insertion of Recovery Cycles.

#### (1) Normal Access

When the PRENB and PWENB bits in CSnMOD are set to 0 to disable page-read and page-write access, respectively, all bus accesses will take the form of normal read and write operations.

Even when the PRENB and PWENB bits in CSnMOD are set to 1 to enable page-read and page-write access, respectively, bus access other than page access will take the form of normal read and write operations.

Figure 16.12 to Figure 16.14 show the normal access operations.

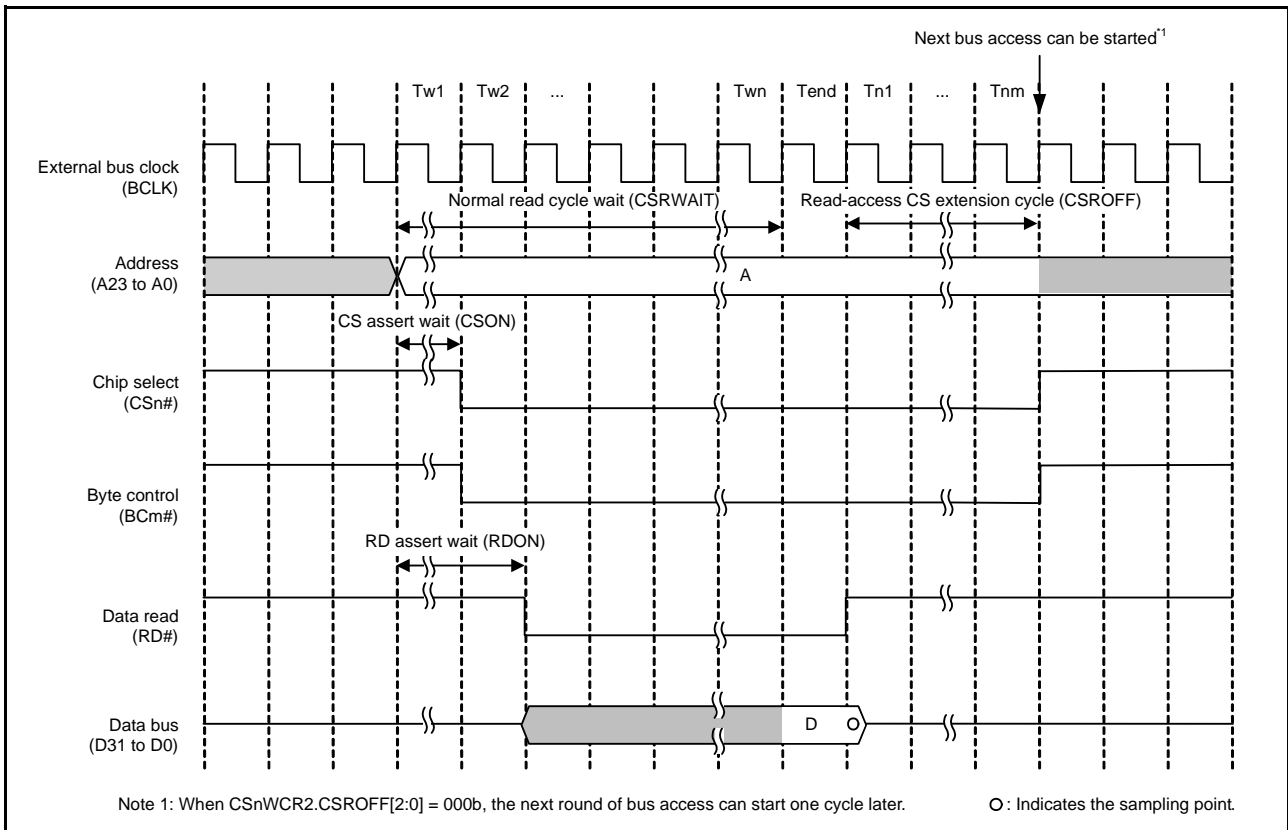


Figure 16.12 Bus Timing (Normal-Read Operation) (n = 0 to 7, m = 0 to 3)

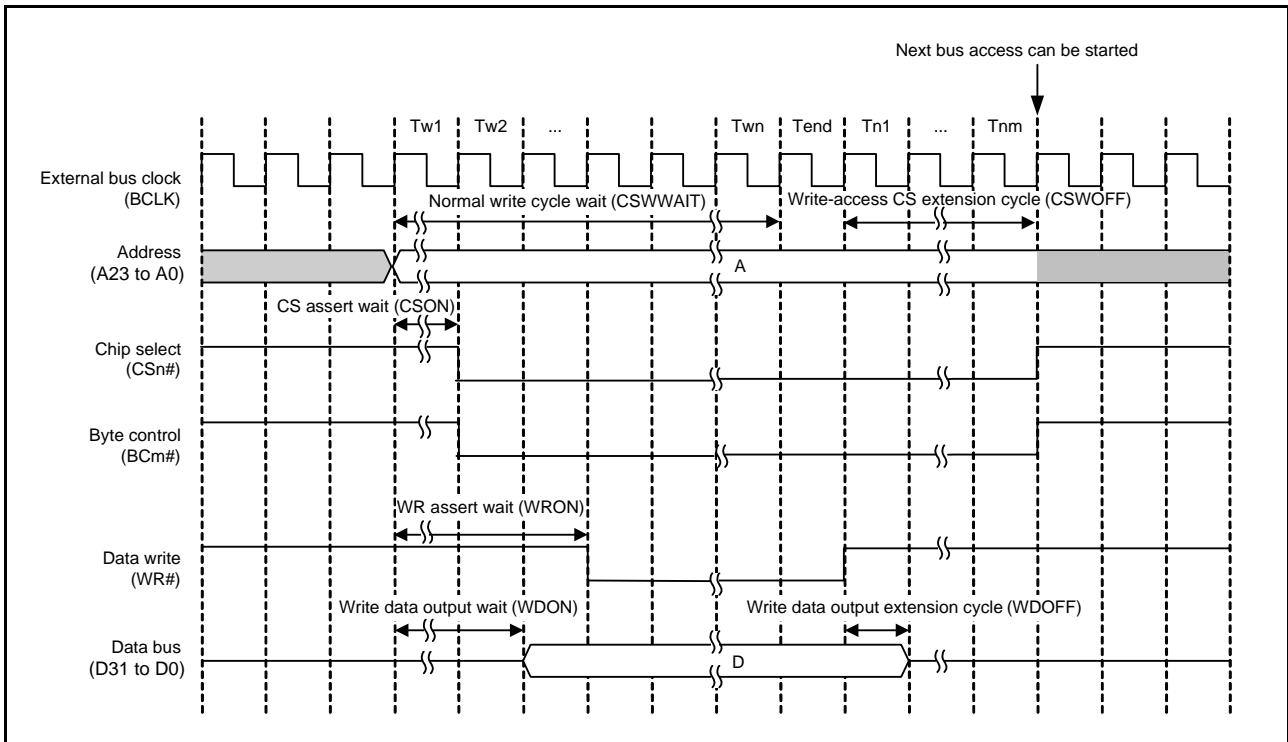
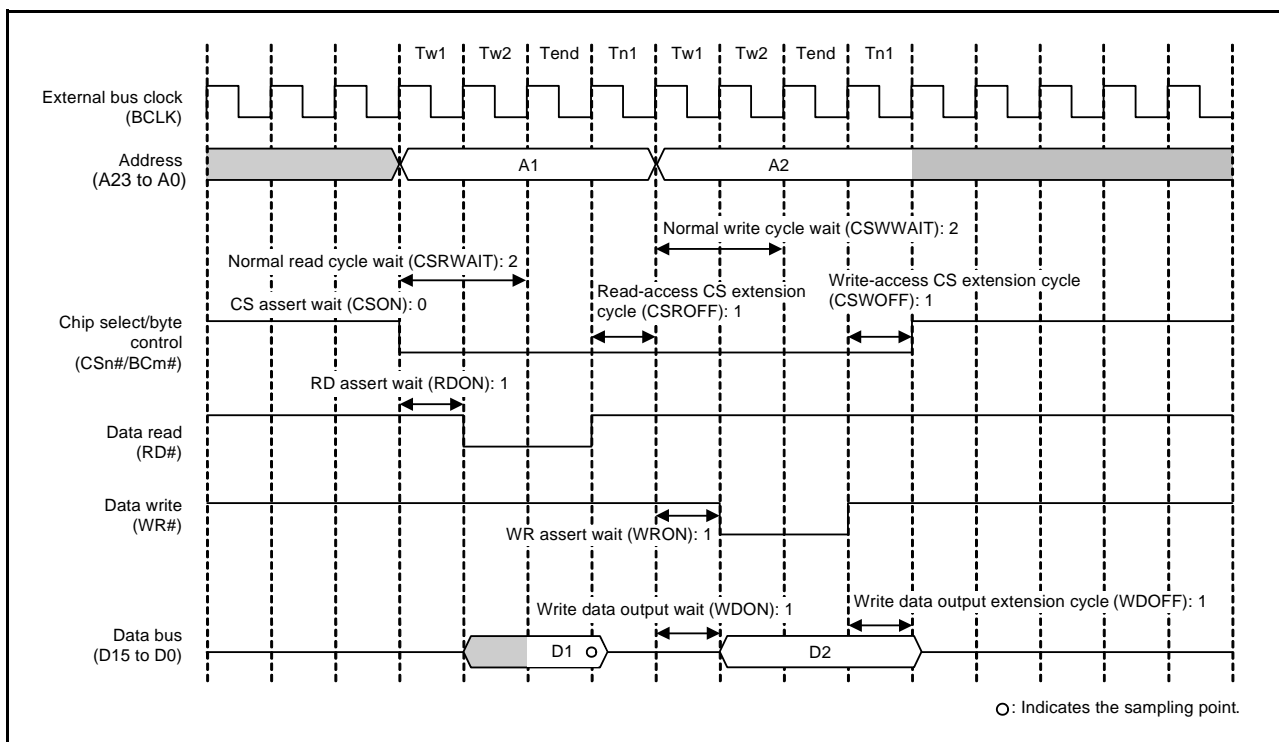


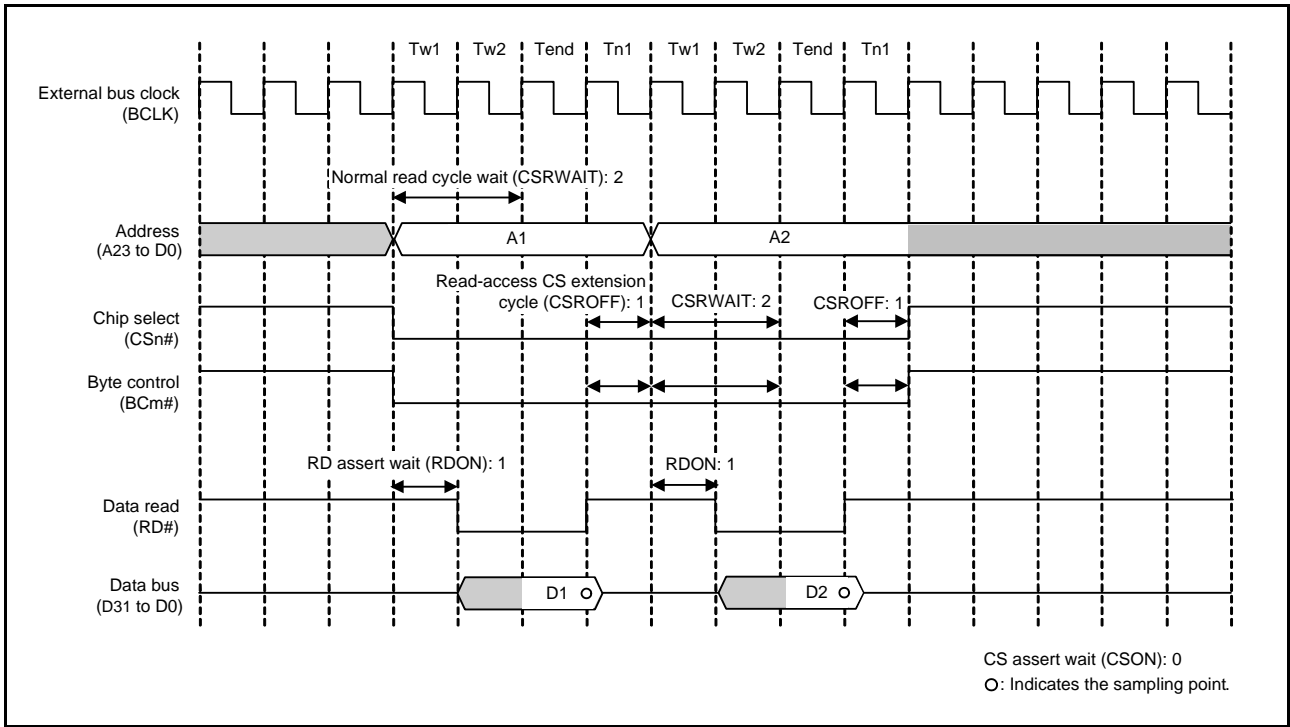
Figure 16.13 Bus Timing (Normal-Write Operation, Single Write Strobe Mode) (n = 0 to 7, m = 0 to 3)



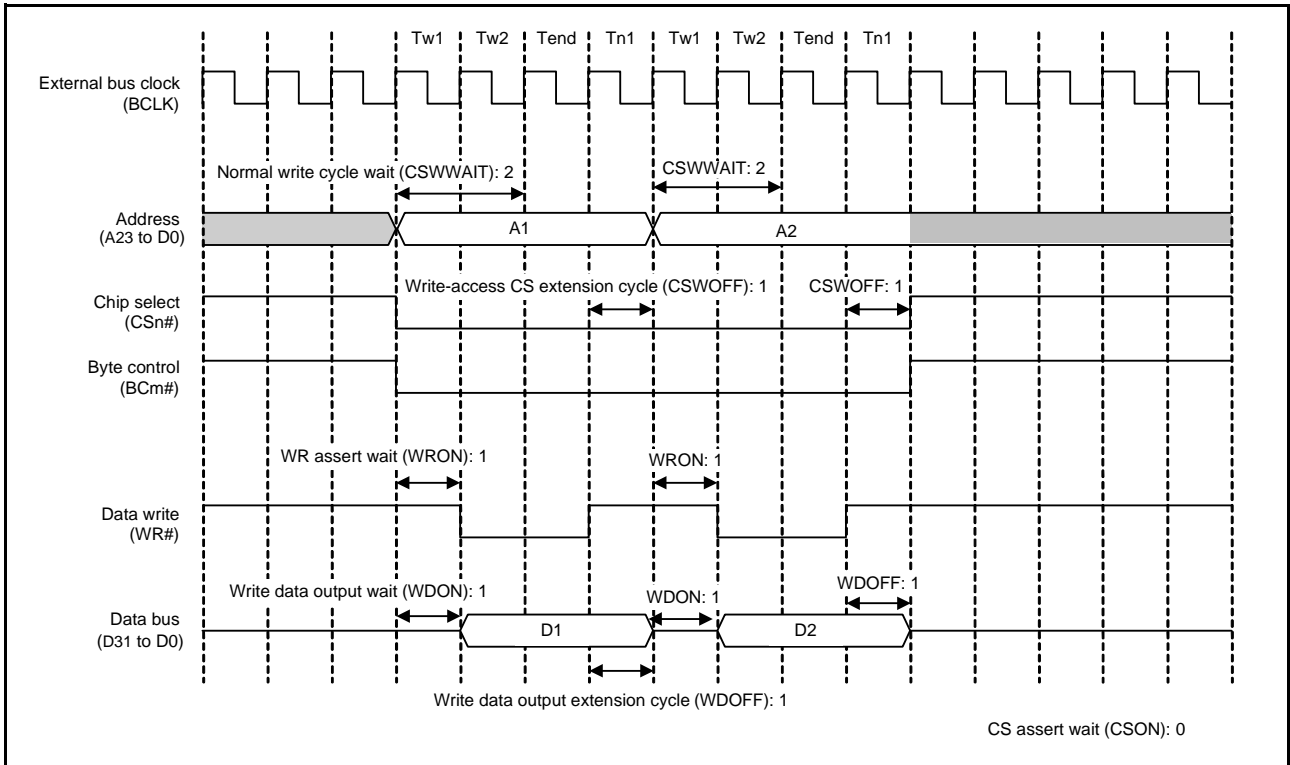
**Figure 16.14 Example of Normal Access Operation (Read/Write) (n = 0 to 7, m = 0 to 3)**

When two or more rounds of external bus access are required in response to a single request for transfer from a bus master, normal access operations (steps (a) to (d) above) are repeated. Figure 16.15 and Figure 16.16 show examples of operations when two rounds of bus access are generated in response to a single request for transfer. If the recovery cycle insertion condition is satisfied, recovery cycles (step (f) above) are also inserted in the second and subsequent external bus accesses (see Figure 16.37).

The values of the wait control registers are example settings. In practice, set the register bits according to the specifications of connected devices.



**Figure 16.15 Example of Normal-Read Operation**  
 (when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer)  
 (n = 0 to 7, m = 0 to 3)



**Figure 16.16 Example of Normal-Write Operation**  
 (when Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer, in  
 Single Write Strobe Mode) (n = 0 to 7, m = 0 to 3)

Figure 16.17 and Figure 16.18 show examples of normal read and write accesses to a 32-bit bus space in 16 bits.

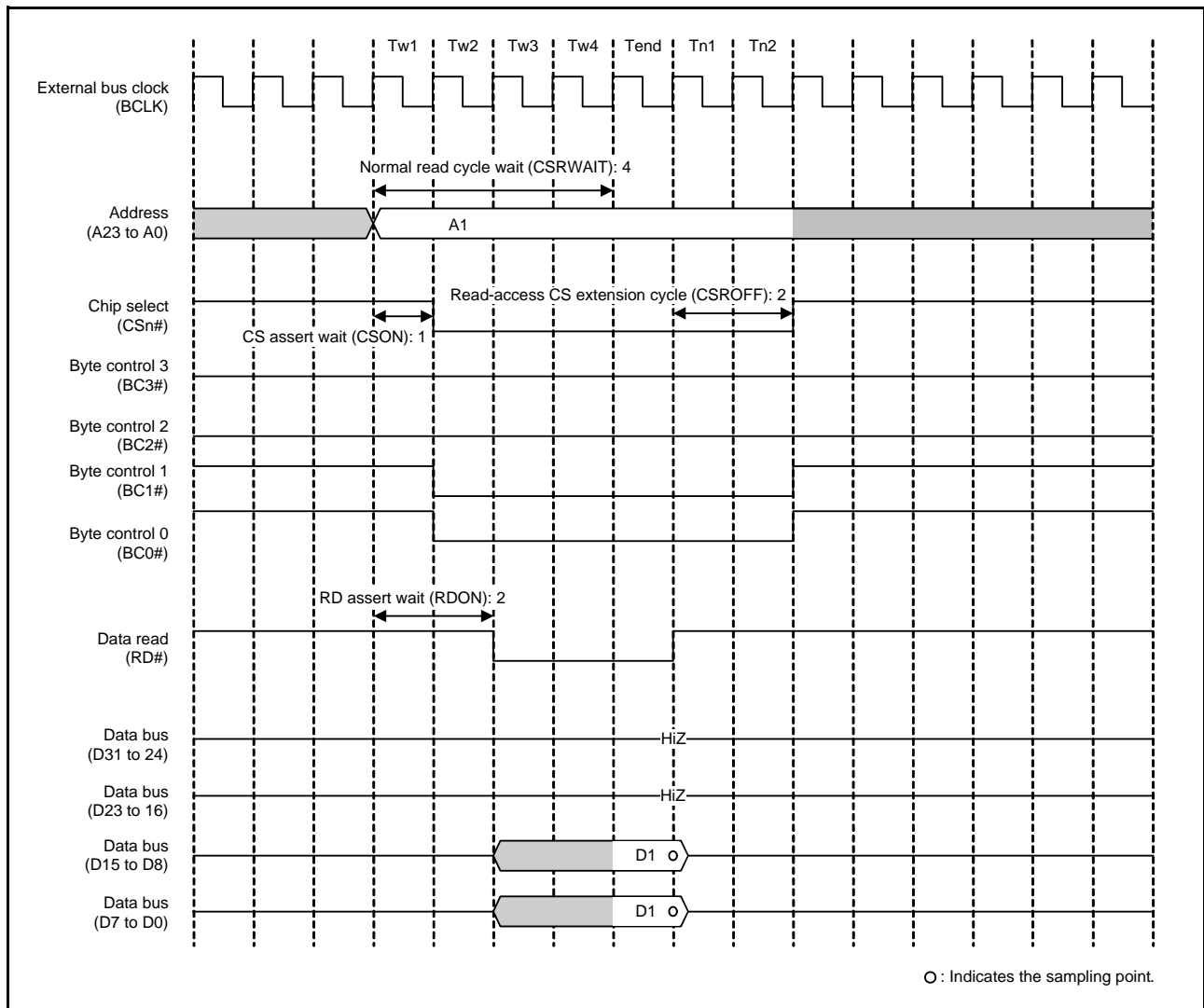
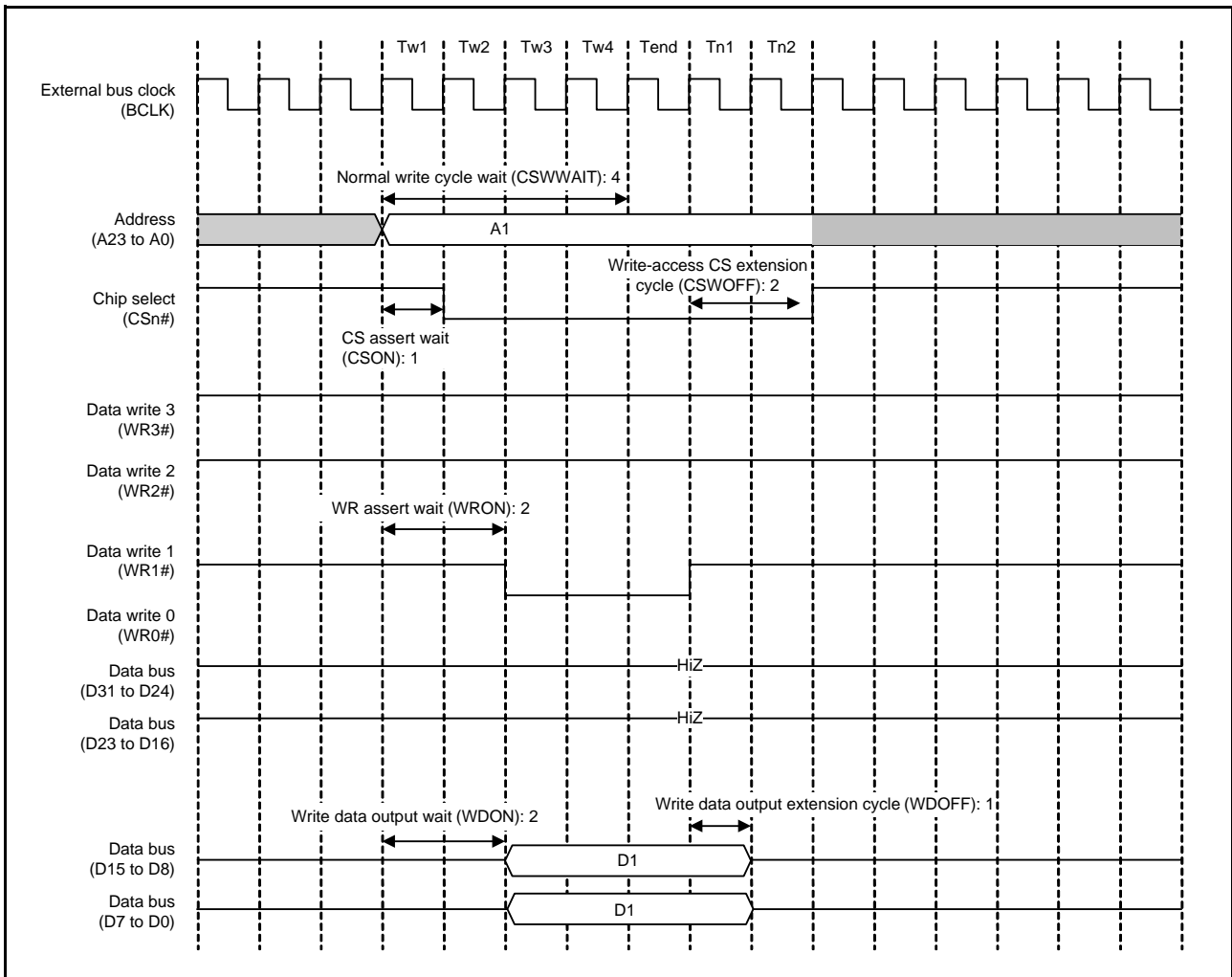


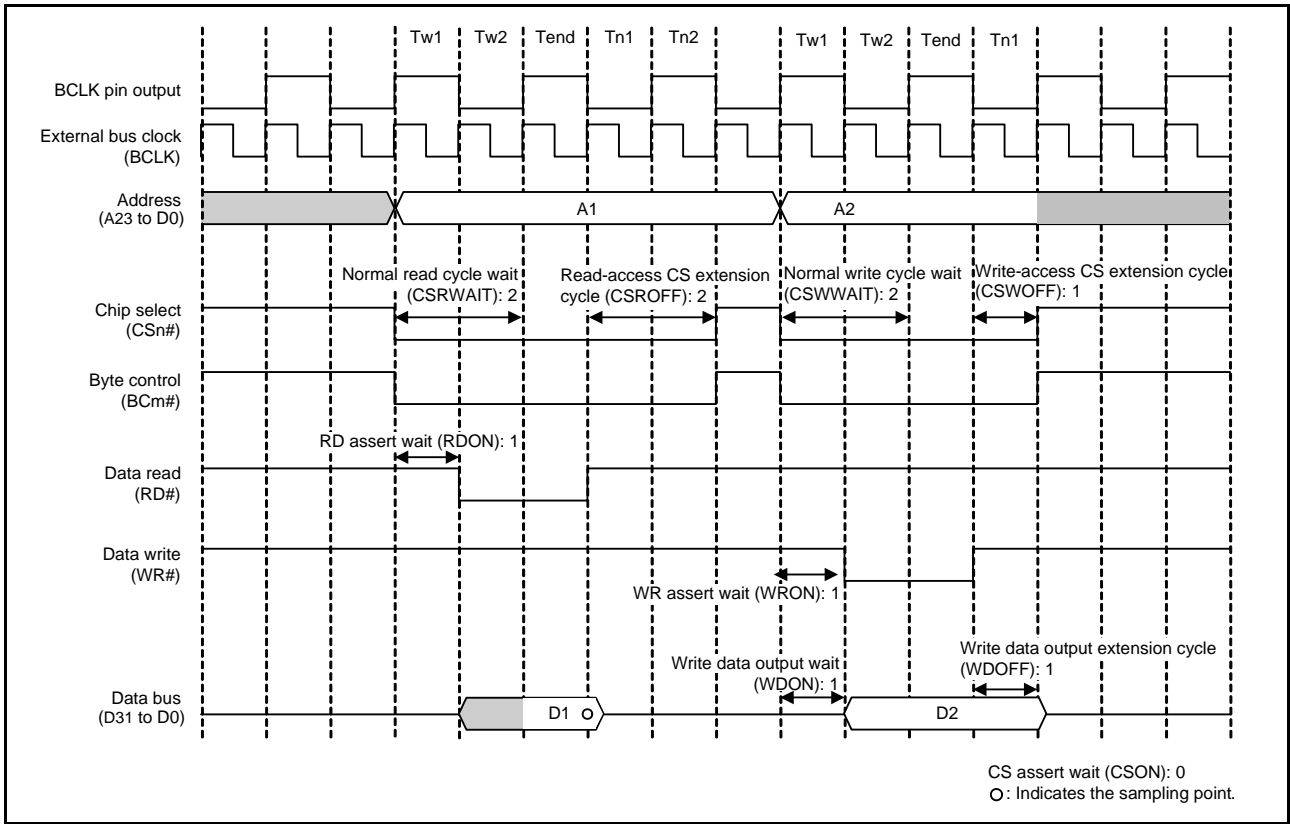
Figure 16.17 Example of Normal-Read Operation (when 32-Bit Bus Space is Accessed in 16 Bits) (n = 0 to 7)



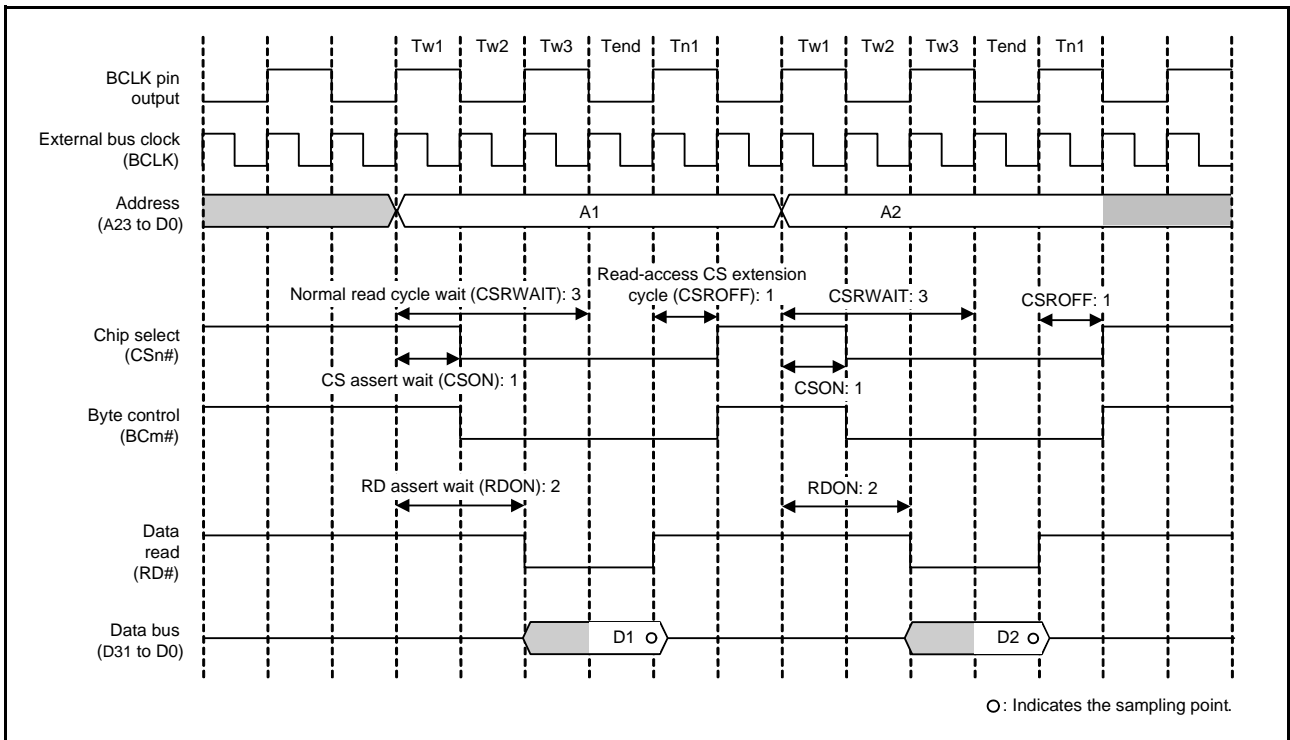


**Figure 16.18 Example of Normal-Write Operation**  
**(when 32-Bit Bus Space is Accessed in 16 Bits, in Byte Strobe Mode) (n = 0 to 7)**

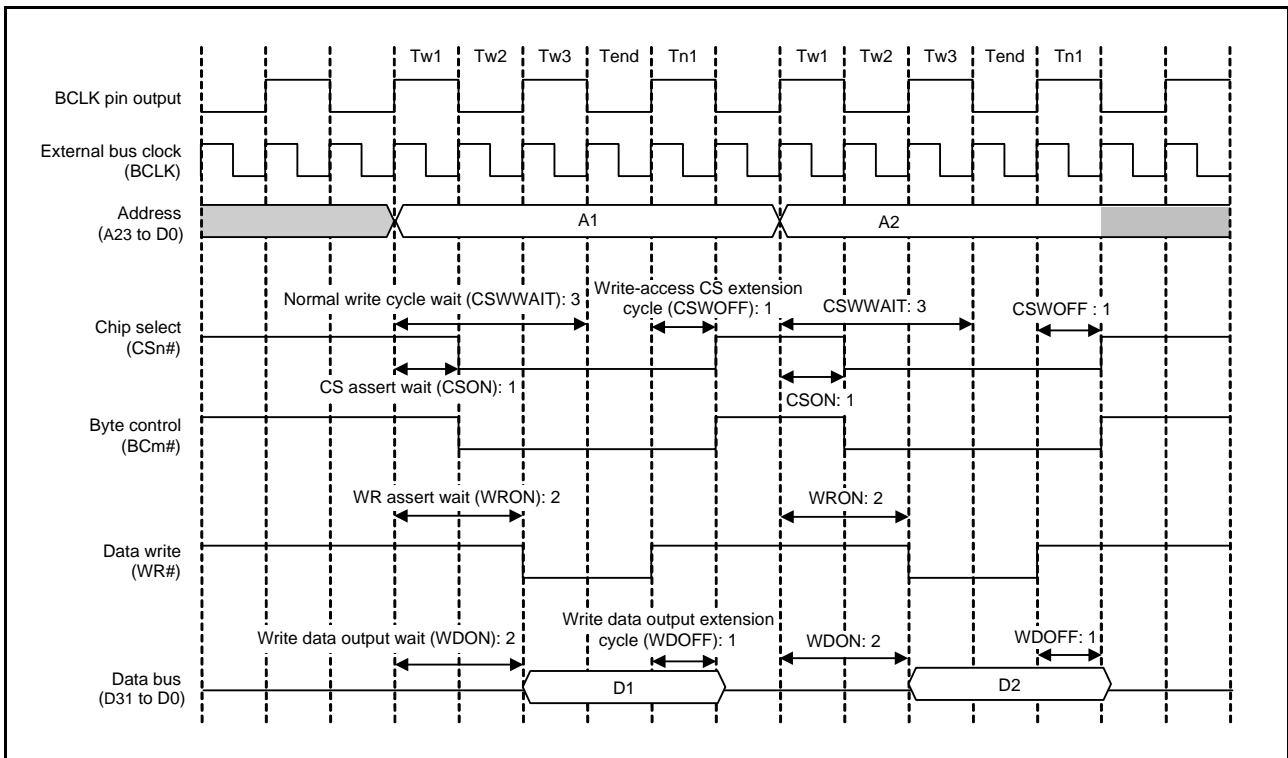
Figure 16.19 to Figure 16.23 show examples of normal accesses made with the 1/2 BCLK selected with the BCLK pin output select bit.



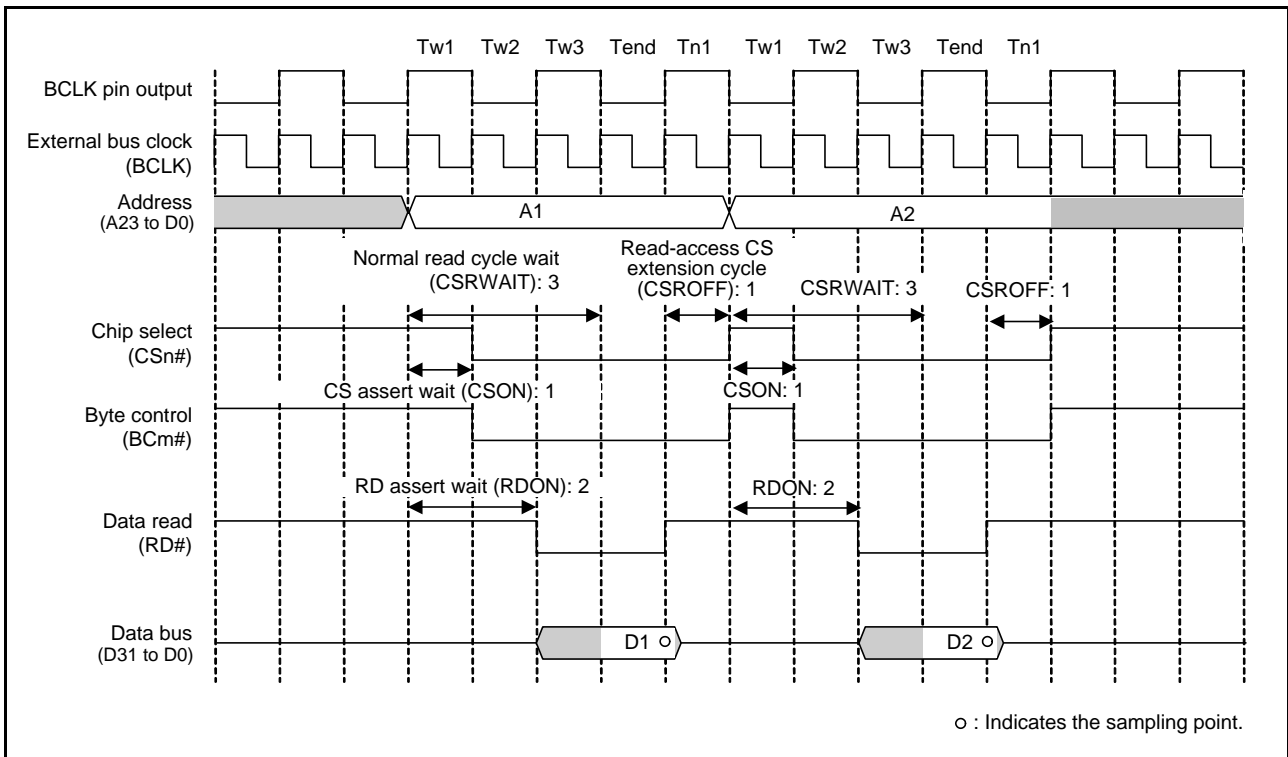
**Figure 16.19** Example of Normal Access (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit) (n = 0 to 7, m = 0 to 3)



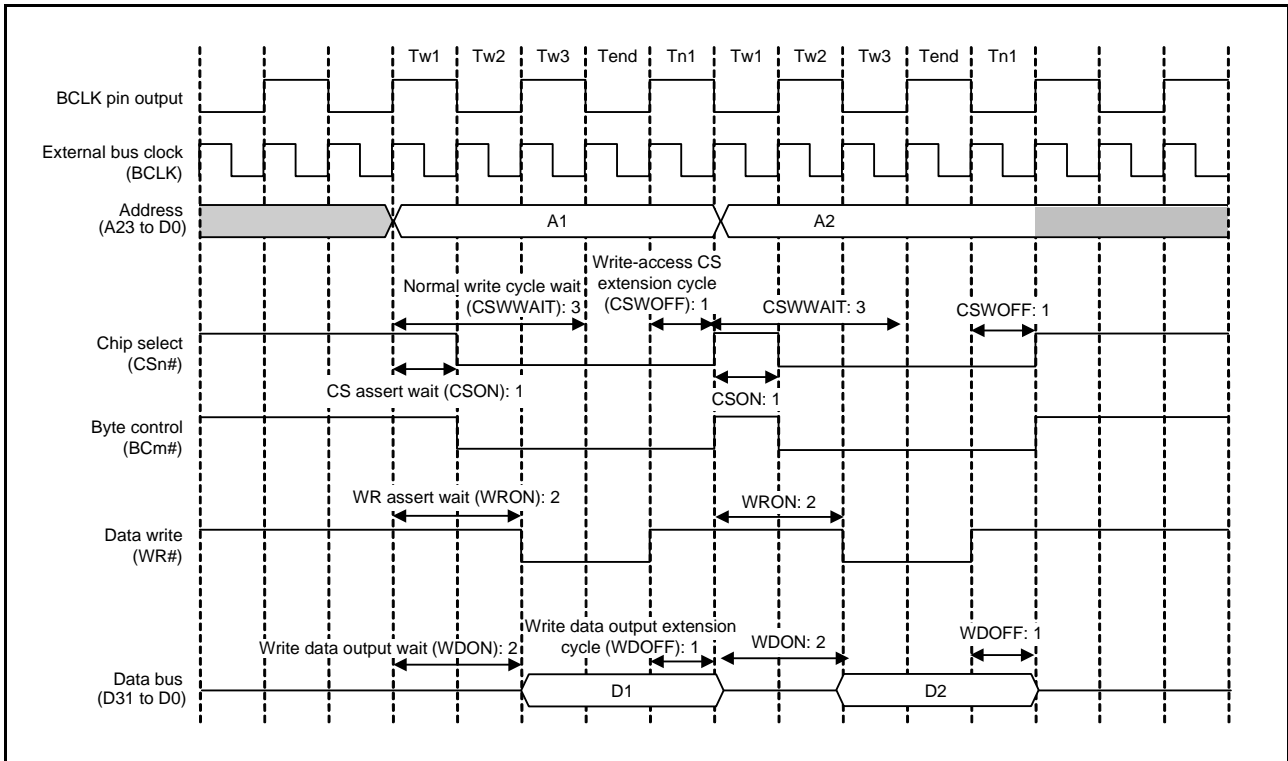
**Figure 16.20** Example of Normal-Read Operation (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit) (n = 0 to 7, m = 0 to 3)



**Figure 16.21 Example of Normal-Write Operation**  
 (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit) (n = 0 to 7, m = 0 to 3)



**Figure 16.22 Example of Normal-Read Operation**  
 (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 7, m = 0 to 3)



**Figure 16.23 Example of Normal-Write Operation**  
**(when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 7, m = 0 to 3)**

(2) Page Access

When the PRENB and PWENB bits in CSnMOD are set to 1 to enable page-read and page-write access, respectively, the bus access for page access operations becomes page reading and writing. Specifically, page access can occur when two or more rounds of external bus access are required for a single transfer request from the bus master. However, normal access is made when the split accesses are not aligned or the access spreads over the 32-bit boundary. See Figure 16.8 to Figure 16.11 for the conditions under which page access occurs.

Figure 16.24 and Figure 16.25 show examples of page access operations.

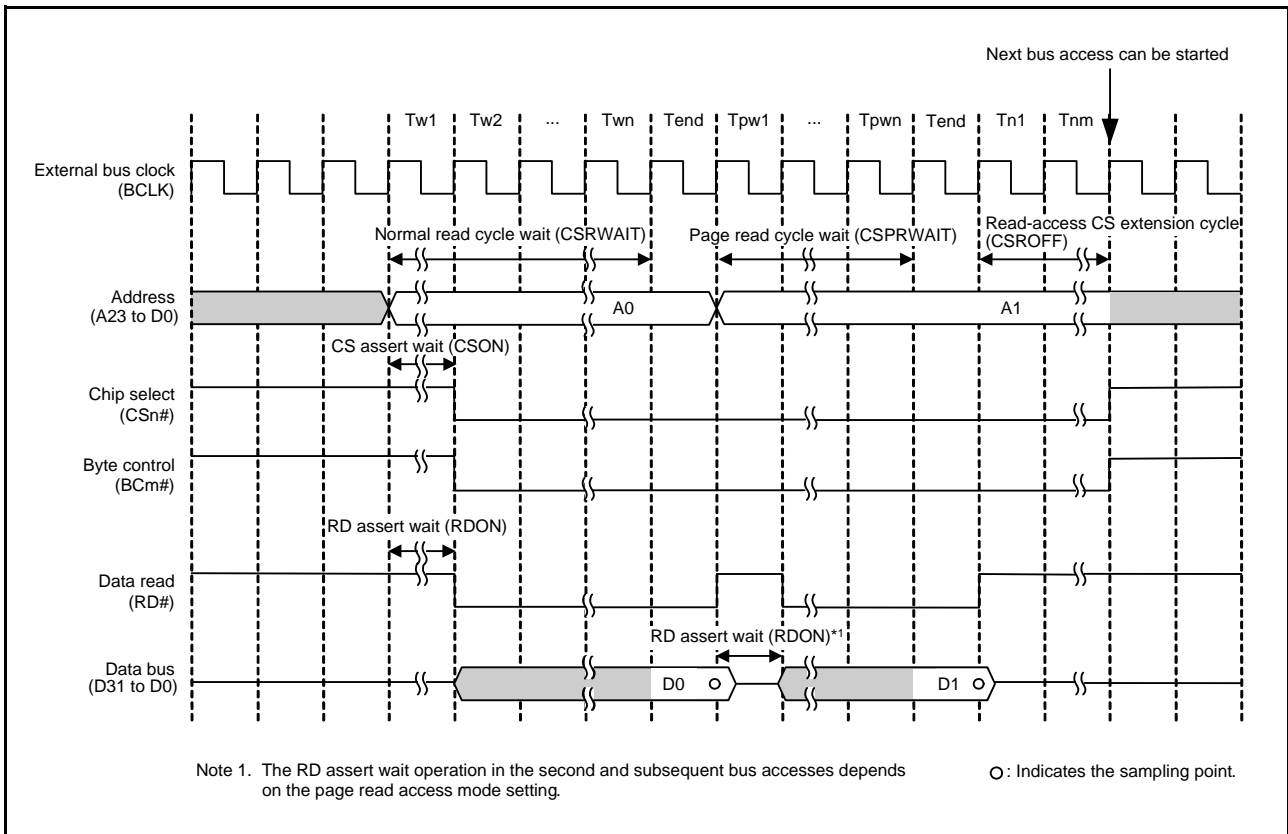


Figure 16.24 Page-Read Access Timing (n = 0 to 7, m = 0 to 3)

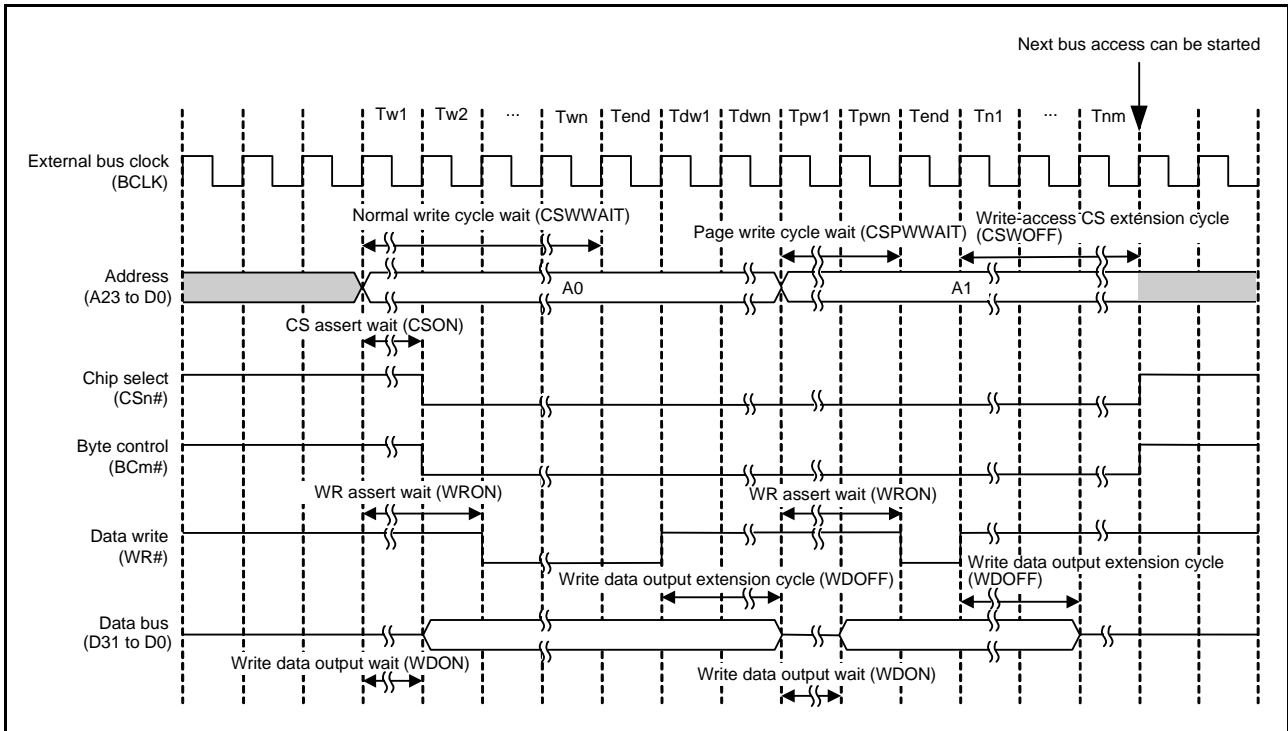


Figure 16.25 Page-Write Access Timing (n = 0 to 7, m = 0 to 3)

Figure 16.26 and Figure 16.27 show examples of operations for access to a 16-bit bus space in 32 bits. The values of the wait control registers are example settings. In practice, the register settings will correspond to the specifications of connected devices.

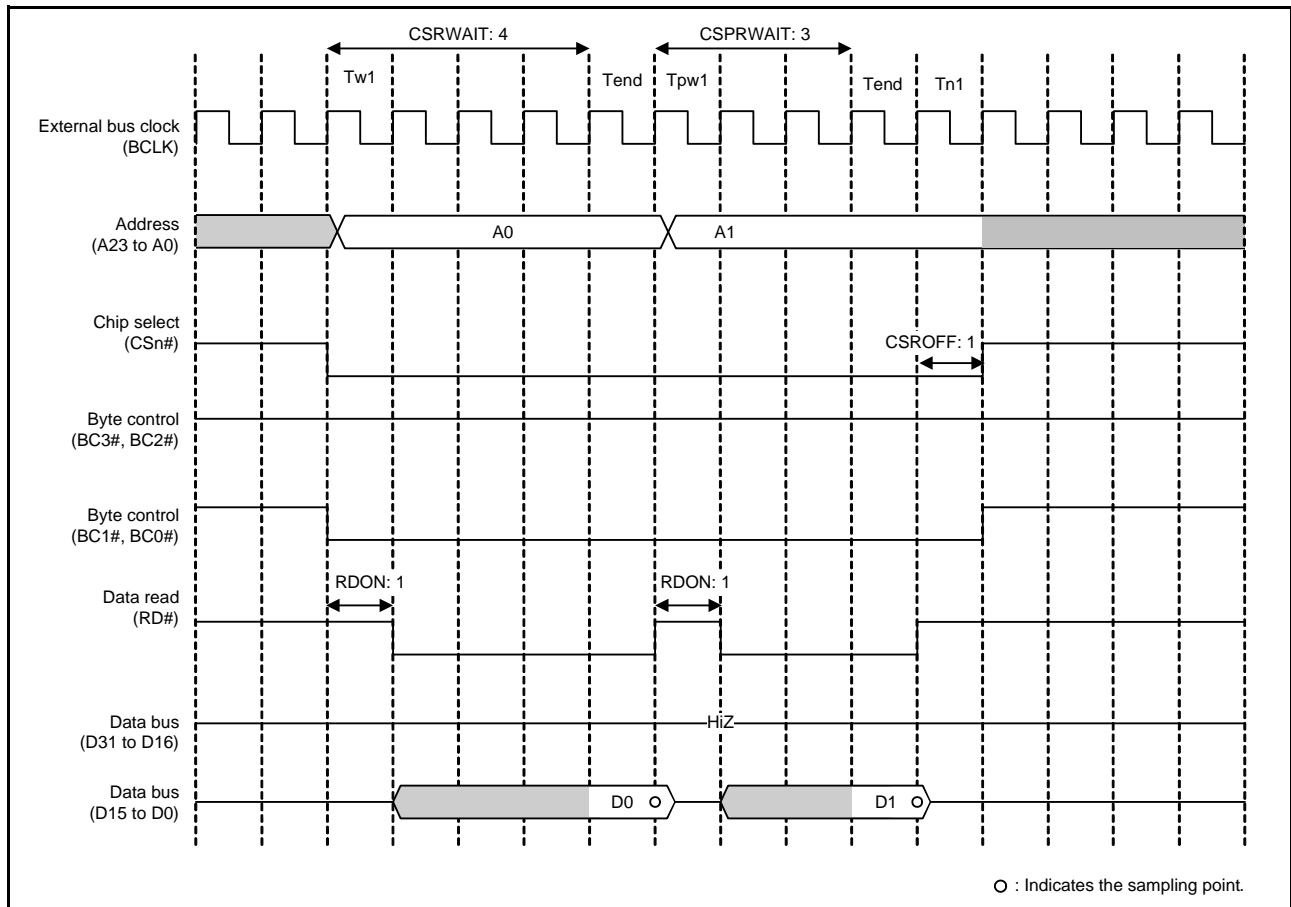
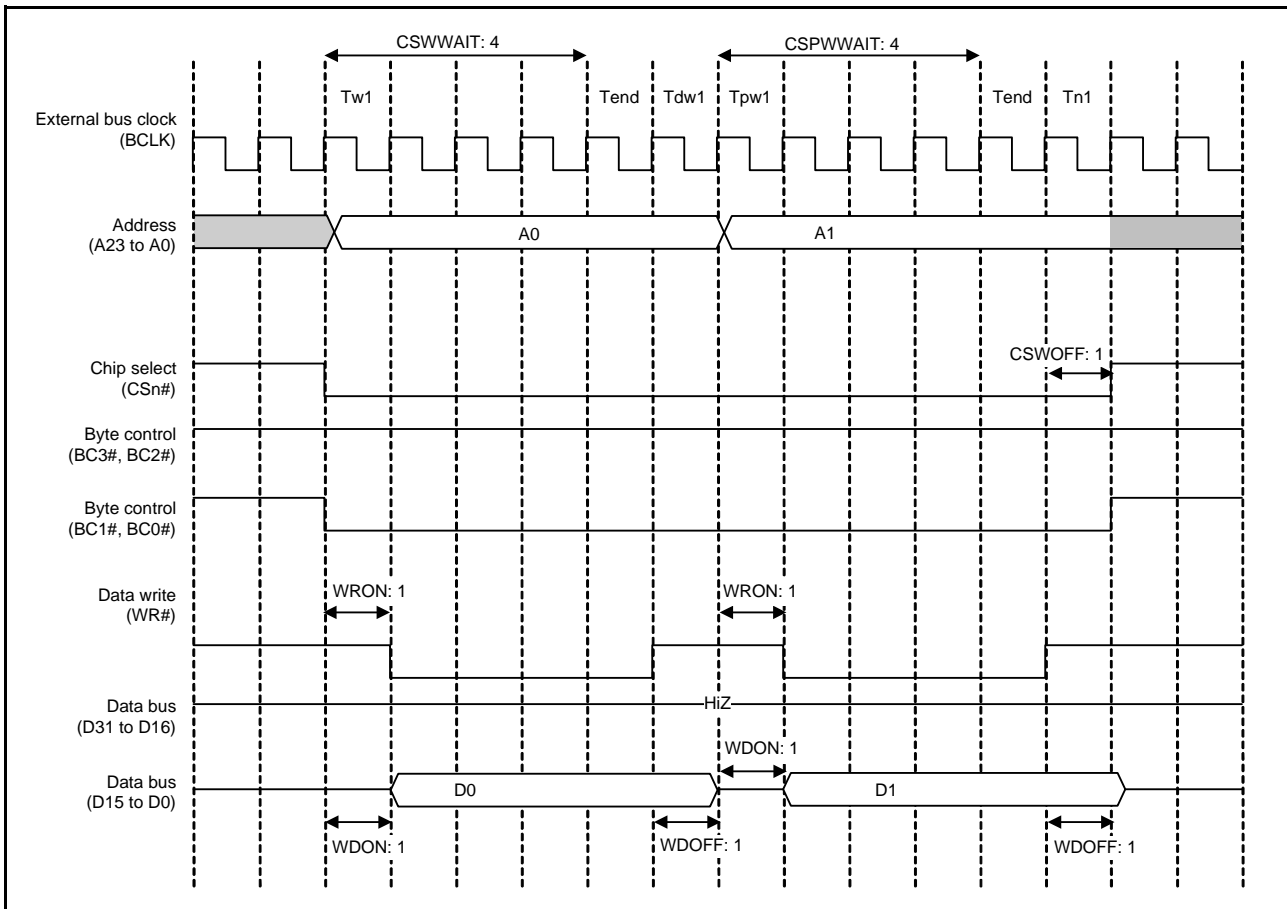
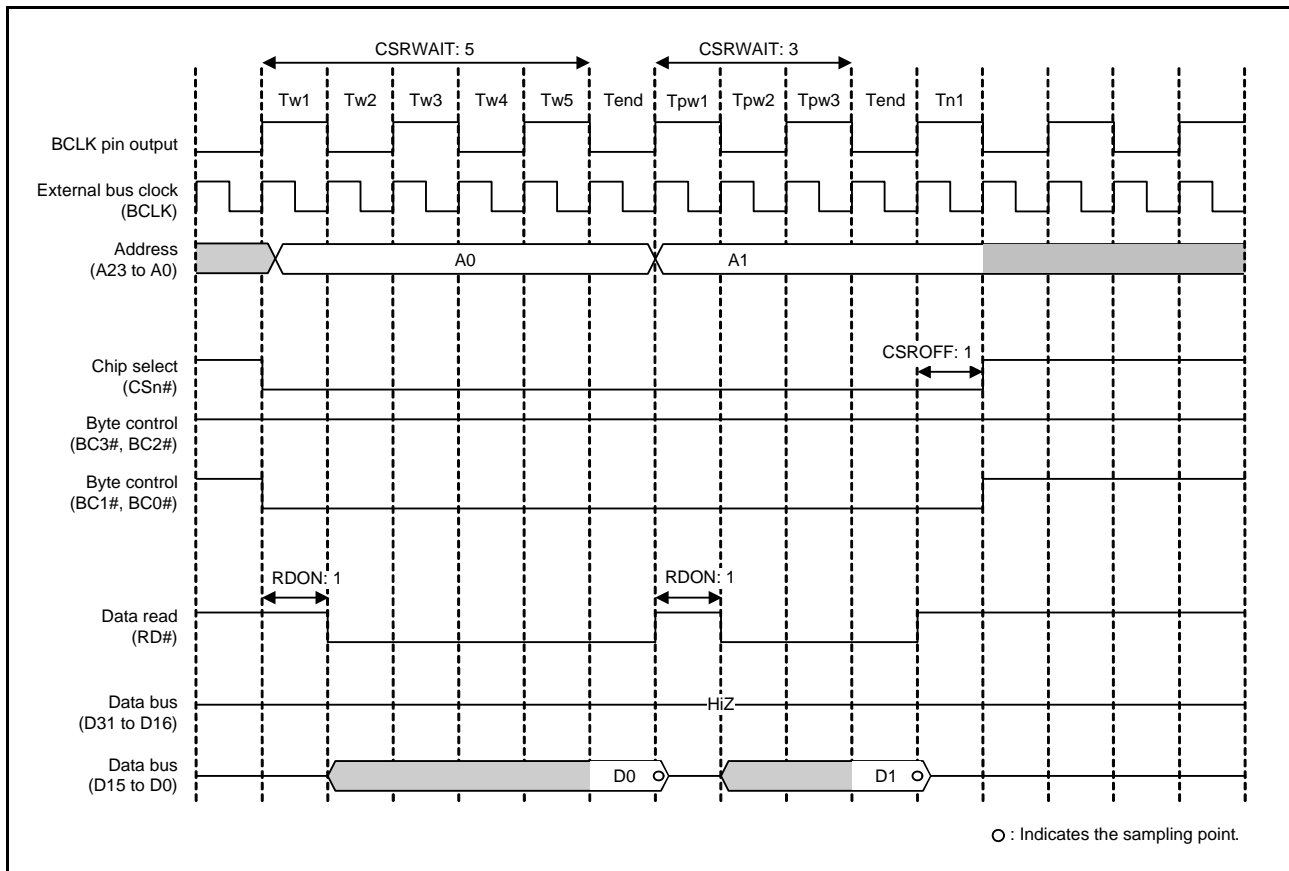


Figure 16.26 Example of Page-Read Access Operation (when 16-Bit Bus Space is Accessed in 32 Bits) (n = 0 to 7)



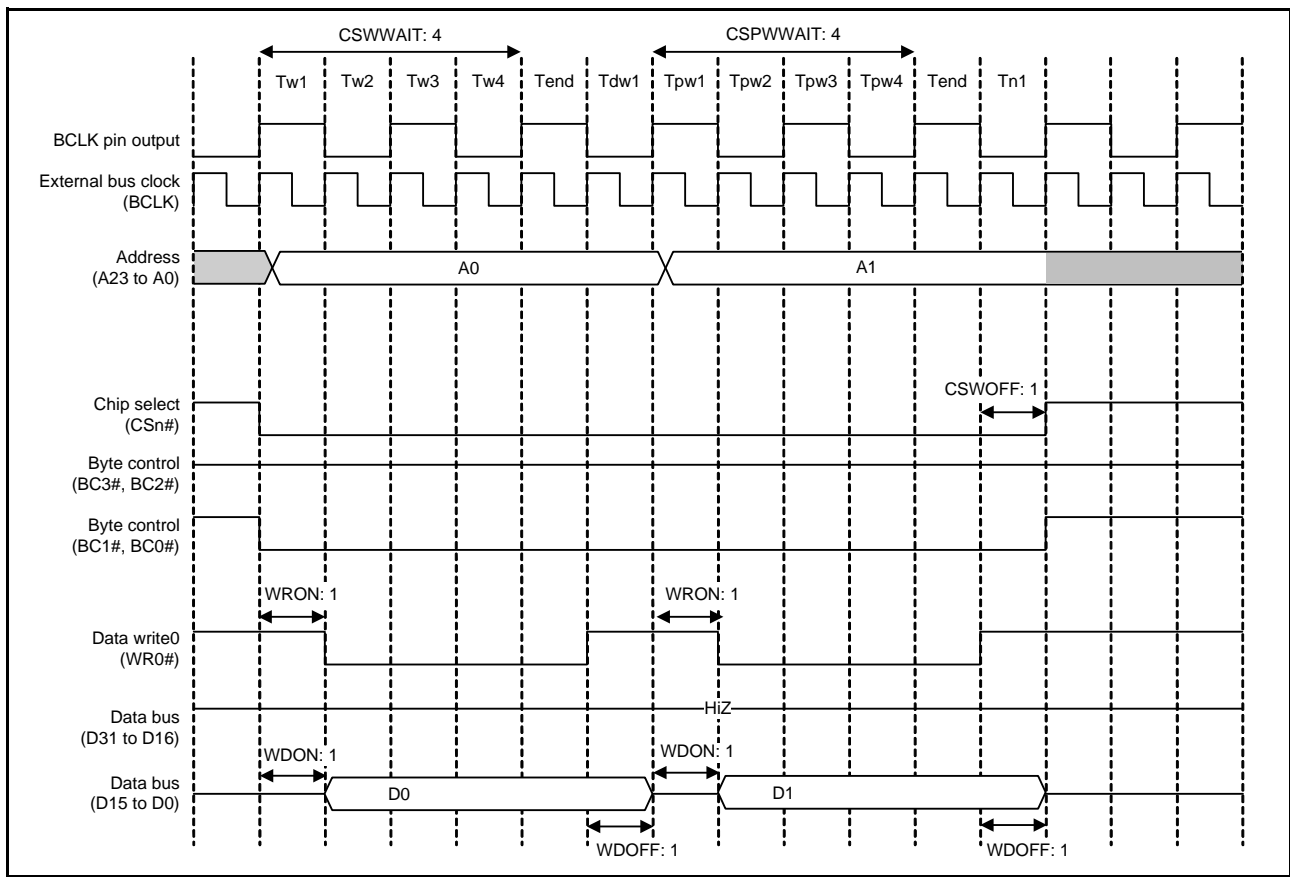
**Figure 16.27 Example of Page-Write Access Operation**  
**(when 16-Bit Bus Space is Accessed in 32 Bits, in Single Write Strobe Mode) (n = 0 to 7)**

Figure 16.28 and Figure 16.29 show examples of page access operations performed with the 1/2 BCLK selected with the BCLK pin output select bit.



**Figure 16.28 Example of Page Read Access Operation**  
**(when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer) (n = 0 to 7)**





**Figure 16.29 Example of Page Write Access Operation**  
 (when 1/2 BCLK is Selected with the BCLK Pin Output Select Bit and Two Rounds of Bus Access are Generated in Response to a Single Request for Transfer, in Single Write Strobe Mode) (n = 0 to 7)

## 16.5.2 Address/Data Multiplexed Bus

When the address/data multiplexed I/O interface select bit (MPXEN) in CSnCR is set to 1, addresses and data can be multiplexing input/output to/from the D15 to D0 pins in the corresponding area. Using this function enables direct connection of this LSI to peripheral LSIs requiring address/data multiplexing. When 8-bit width is selected with the BSIZE[1:0] bits in CSnCR, D7 to D0 are multiplexed with A7 to A0. When 16-bit width is selected, D15 to D0 are multiplexed with A15 to A0. In the address/data multiplexed I/O space, accesses are controlled with the ALE, RD#, WRn#, and BCn# signals.

Byte strobe mode or single-write strobe mode is selectable in the same way as for a separate bus. However, with regard to the BCn# signals within the address cycle, the byte-control signal is output for the data being read or written.

During the address/data multiplexed I/O space access, after the number of wait cycles specified by the address cycle wait select bits (AWAIT[1:0]) in CSnWCR2 is inserted in the address output cycle, data access is performed.

- Ta1 to Tan (Address Cycle Wait)

The period Ta1 to Tan is valid only when the address/data multiplexed I/O space is specified. This period is made up of the number of clock cycles between the start of external bus access and one cycle before the address latch (ALE) signal is negated. The number of cycles are selectable within the range from zero to three. Addresses are output until the next cycle of ALE signal negation (address cycle). The timing of ALE signal is the same as that of CS# assertion. After the address cycle, a data cycle is started. CSnWCR1 and CSnWCR2 should be set so that an address cycle and a data cycle do not overlap.

Page access to the address/data multiplexed I/O space is invalid. When the PRENB or PWENB bit in CSnMOD is set to 1 to enable page-read or page-write access, these settings are ignored and normal read or write operation is performed.

Figure 16.30 to Figure 16.32 show examples of operations with the address/data multiplexed I/O interface.

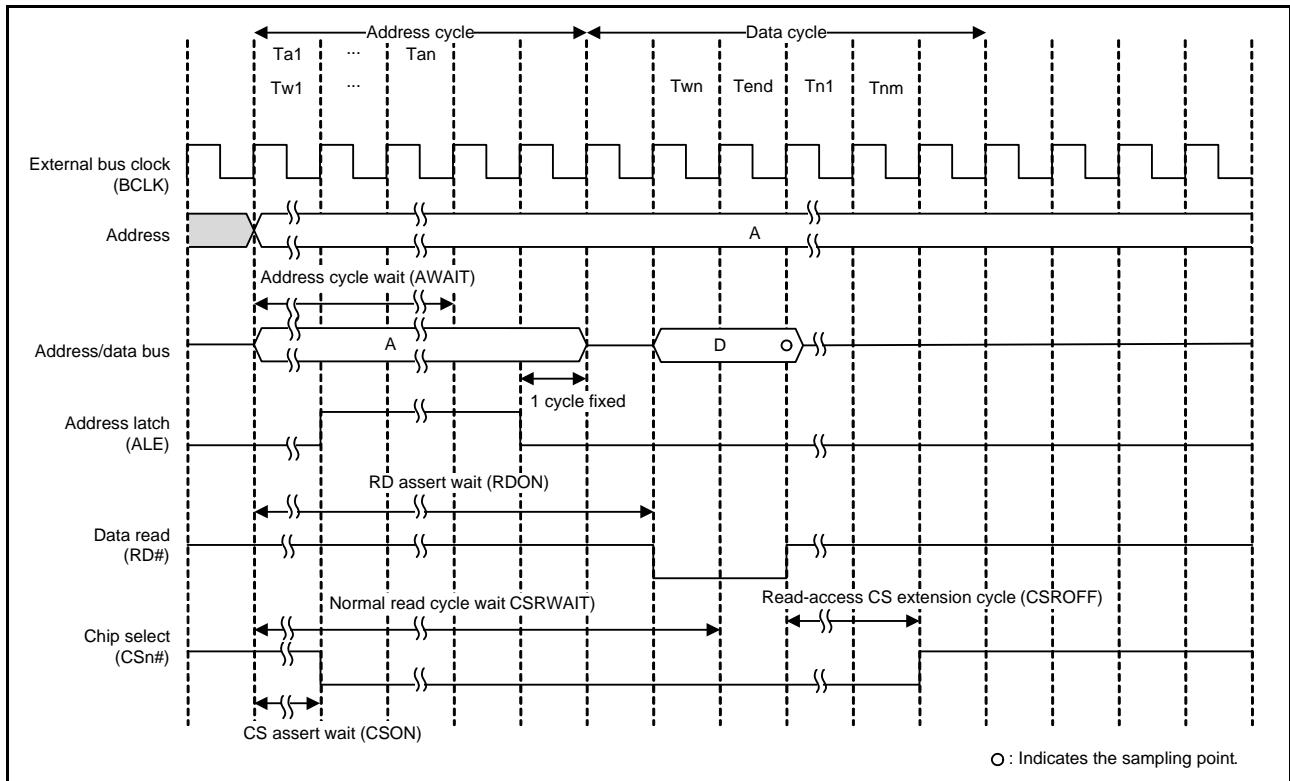


Figure 16.30 Example of Read Access Operation (with Address/Data Multiplexed I/O Interface)

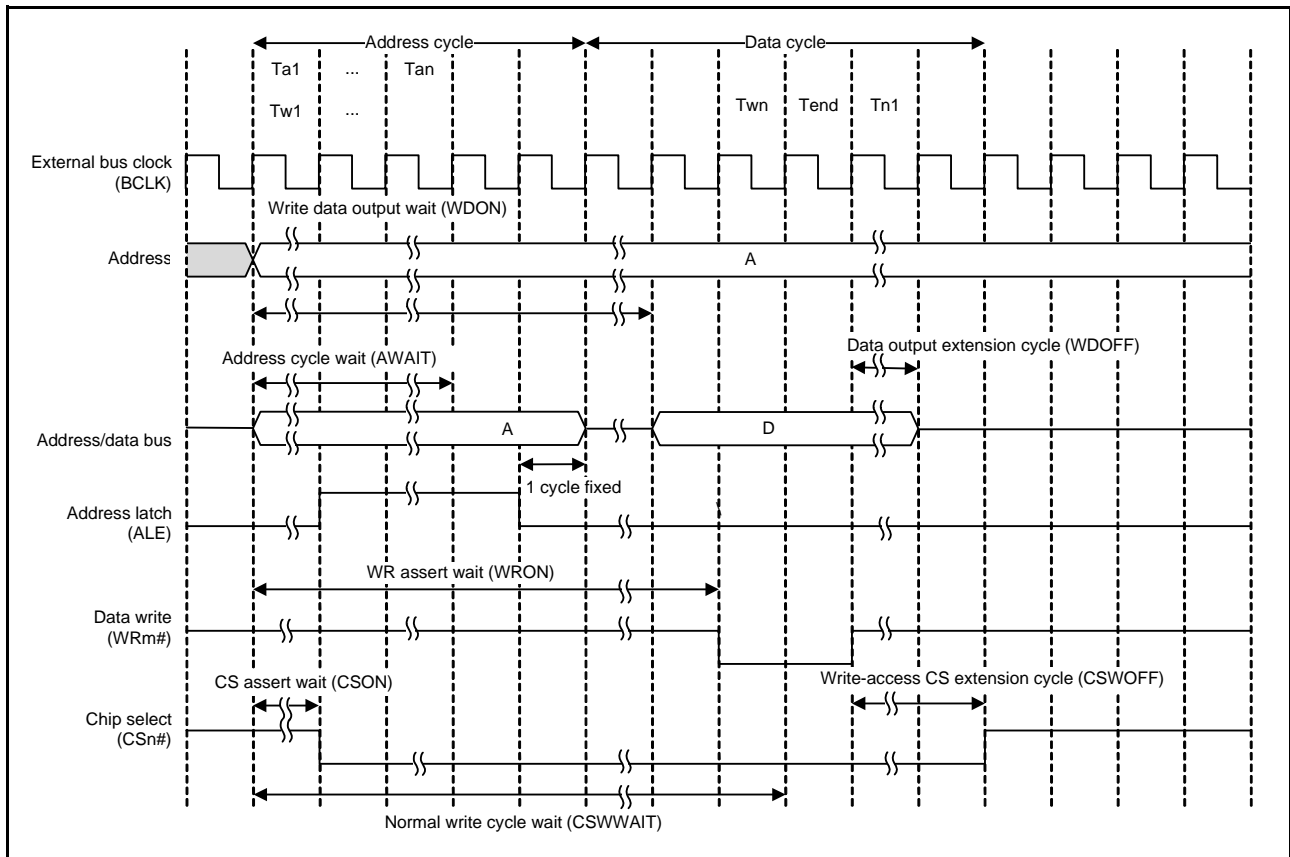


Figure 16.31 Example of Write Access Operation (with Address/Data Multiplexed I/O Interface; m = 0, 1)

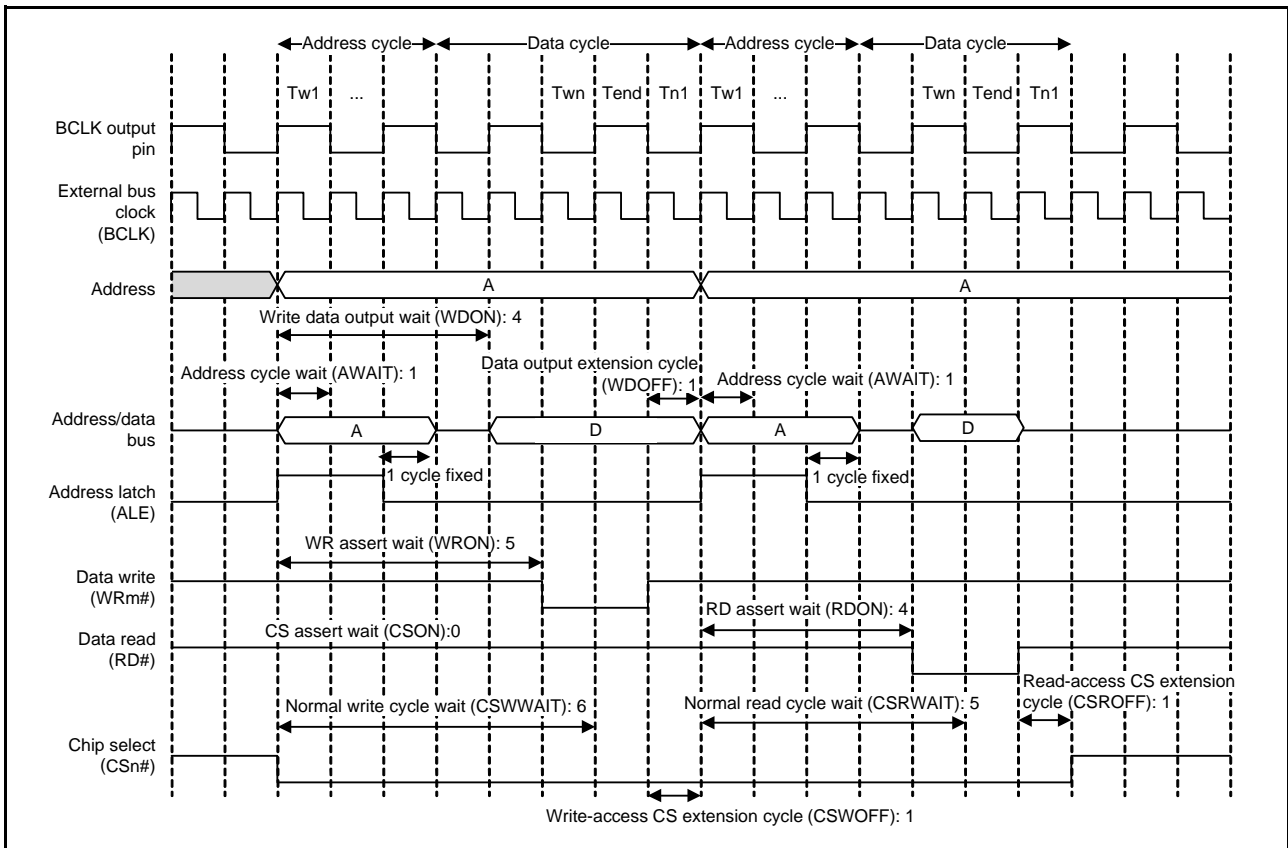


Figure 16.32 Example of Bus Timing (with Address/Data Multiplexed I/O Interface; m = 0, 1)

### 16.5.3 External Wait Function

Wait cycles can be extended by the WAIT# signal over the length of normal access cycle wait (specified by the CSRWAIT[4:0] and CSWAIT[4:0] bits in CSnWCR1) and page access cycle wait (specified by the CSPRWAIT[2:0] and CSPWAIT[2:0] bits in CSnWCR1).

When external wait is enabled (the EWENB bit = 1 in CSnMOD), wait cycles are inserted while the WAIT# signal is held low. When external wait is disabled (the EWENB bit = 0 in CSnMOD), the WAIT# signal has no effect.

All wait cycles specified in CSnWCR1 are inserted independently of the WAIT# signal.

#### (1) Normal Access

Sampling of the WAIT# signal begins upon completion of the wait cycle (Tend) specified in CSnWCR1. The bus cycle is extended while the WAIT# signal is held low. The wait cycle ends (Tend) at the next cycle after the WAIT# signal becomes high.

#### (2) Page Access

The first access operation is the same as the normal access operation. Sampling of the WAIT# signal begins upon completion of the wait cycle (Tend) specified in the CSnWCR1 register. The bus cycle is extended while the WAIT# signal is held low. The wait cycle ends (Tend) at the next cycle after the WAIT# signal becomes high.

With respect to the second and subsequent accesses, sampling of the WAIT# signal begins upon completion of the wait cycle of the page access (Tend). The wait cycle of the page access is extended while the WAIT# signal is held low, and ends (Tend) at the next cycle after the WAIT# signal becomes high.

Figure 16.33 and Figure 16.34 show examples of external wait insertion timing with the separate bus interface.

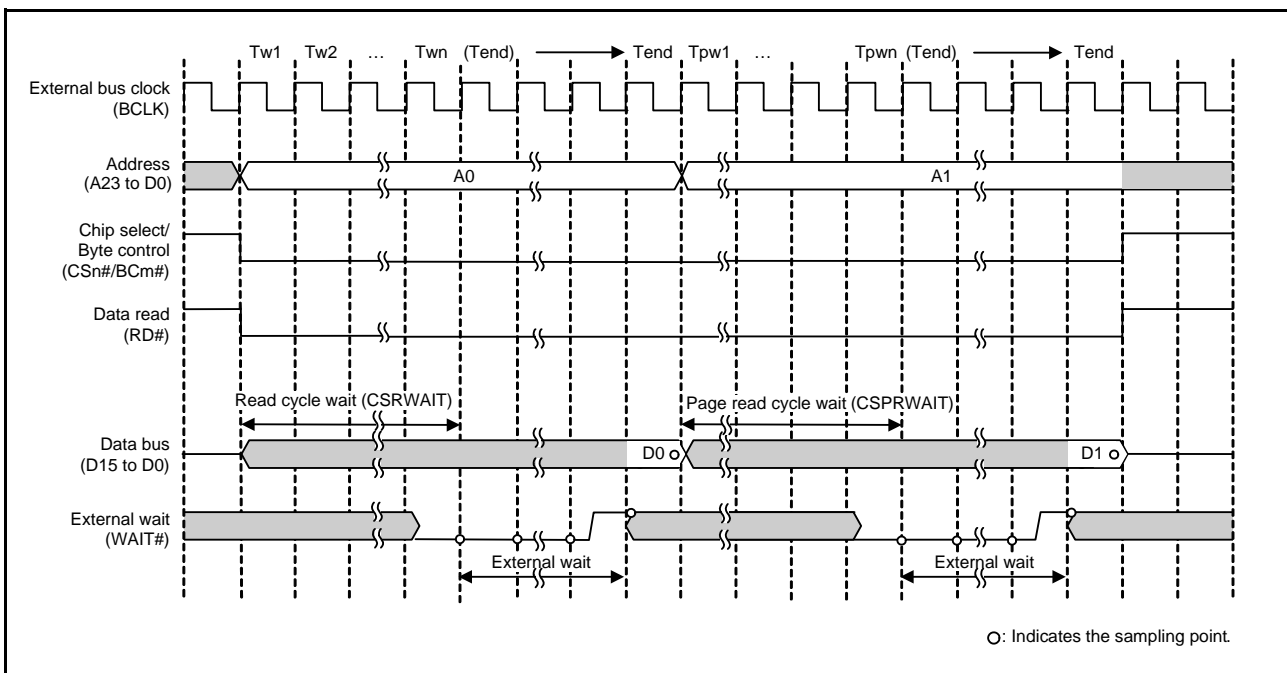
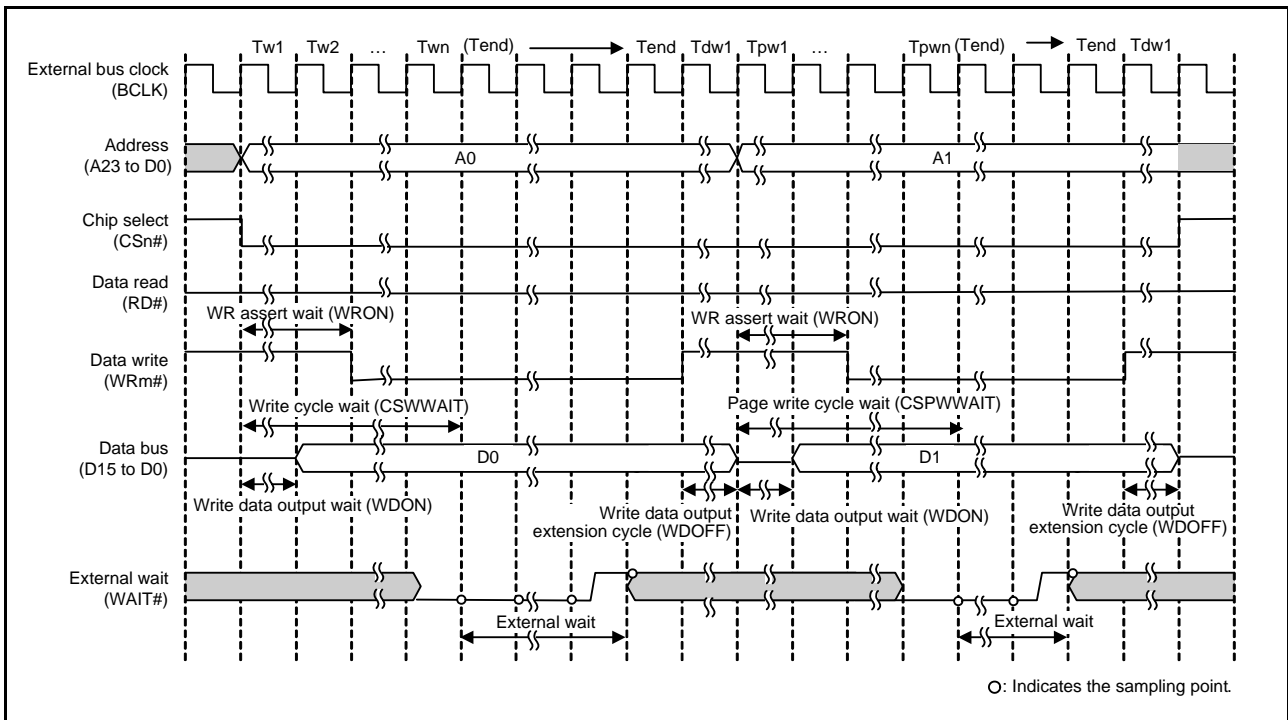


Figure 16.33 Example of External Wait Timing (Page-Read Access to 16-Bit Bus Space) (n = 0 to 7, m = 0, 1)

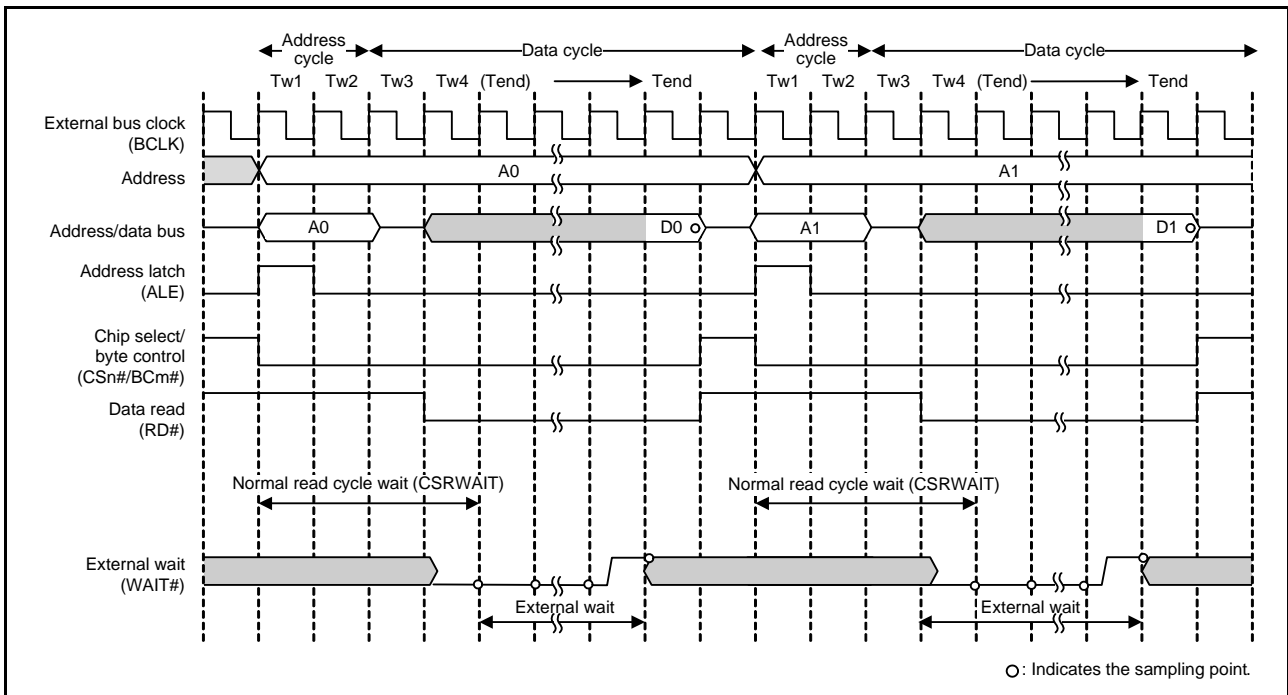


**Figure 16.34 Example of External Wait Timing (Page-Write Access to 16-Bit Bus Space, in Byte Strobe Mode) (n = 0 to 7, m = 0, 1)**

**(3) Address/Data Multiplexed I/O Interface**

In a data cycle with the address/data multiplexed I/O interface, programmed waits and pin waits using the WAIT pin can be inserted in the same way as that with the separate bus interface.

Address cycles are not affected by the wait control settings. Figure 16.35 shows an example of external wait insertion timing with the address/data multiplexed I/O interface.



**Figure 16.35 Example of External Wait Insertion Timing (with Address/Data Multiplexed I/O Interface; m = 0, 1)**

### 16.5.4 Insertion of Recovery Cycles

Recovery cycles can be inserted between consecutive rounds of external bus access by setting the recovery cycle insertion enable bit in CSRECEN to 1.

The number of recovery cycles to be inserted after read cycles and write cycles can be separately set for each area using CSnREC. When the preceding bus cycle is a write access, the number of write recovery cycles should be set with the WRCV[3:0] bits for the area. When the preceding bus cycle is a read access, the number of read recovery cycles should be set with the RRCV[3:0] bits for the area. For example, when CS1 read access occurs after CS0 read access, the number of recovery cycles to be inserted between them is set by the RRCV[3:0] bits in CS0REC.

Recovery cycles can be inserted on any of the following eight conditions. The recovery cycle insertion can be enabled or disabled with the RCVENj (j = 0 to 7) in CSRECEN when the preceding bus access is a separate bus access, and with RCVENMj (j = 0 to 7) when the preceding bus access is an address/data multiplexed bus access.

- After a read access to the external bus, a read access is made to the external bus in the same area.
- After a read access to the external bus, a read access is made to the external bus in a different area.
- After a read access to the external bus, a write access is made to the external bus in the same area.
- After a read access to the external bus, a write access is made to the external bus in a different area.
- After a write access to the external bus, a read access is made to the external bus in the same area.
- After a write access to the external bus, a read access is made to the external bus in a different area.
- After a write access to the external bus, a write access is made to the external bus in the same area.
- After a write access to the external bus, a write access is made to the external bus in a different area.

The recovery cycle starts at the end of the preceding bus cycle, i.e. when the CSn# signal (n = 0 to 7) is negated. A high-level period of the CSn# signal is inserted for the specified recovery cycle period starting from this point.

The CSn# signal for the next round of bus access is asserted immediately after the end of recovery cycles in the fastest case. Even if the next request for access to an external address space is generated during the recovery period, the next round of access over the external bus will start immediately after the end of recovery cycles.

When two or more external bus access cycles are required for a single transfer request from a bus master and the recovery cycle insertion condition is satisfied, recovery cycles are also inserted between these bus access cycles. However, when page read access is enabled (CSnMOD.PRENB = 1) or page write access is enabled (CSnMOD.PWENB = 1), recovery cycles are not inserted except after the last bus access cycle of the transfer even if the recovery cycle insertion condition is satisfied (Figure 16.38). Similarly, during normal accesses with page access enabled, recovery cycles are not inserted between bus access cycles but inserted only after the last bus access cycle of the transfer. Similarly, during normal accesses with page access enabled, with the separate bus interface, recovery cycles are not inserted between bus access cycles but inserted only after the last bus access cycle of the transfer. With the address/data multiplexed I/O interface, when the recovery cycle insertion condition is satisfied, recovery cycles are inserted between bus access cycles regardless of the page access enable setting.

Figure 16.36 to Figure 16.38 show examples of recovery cycle insertion with the separate bus interface.

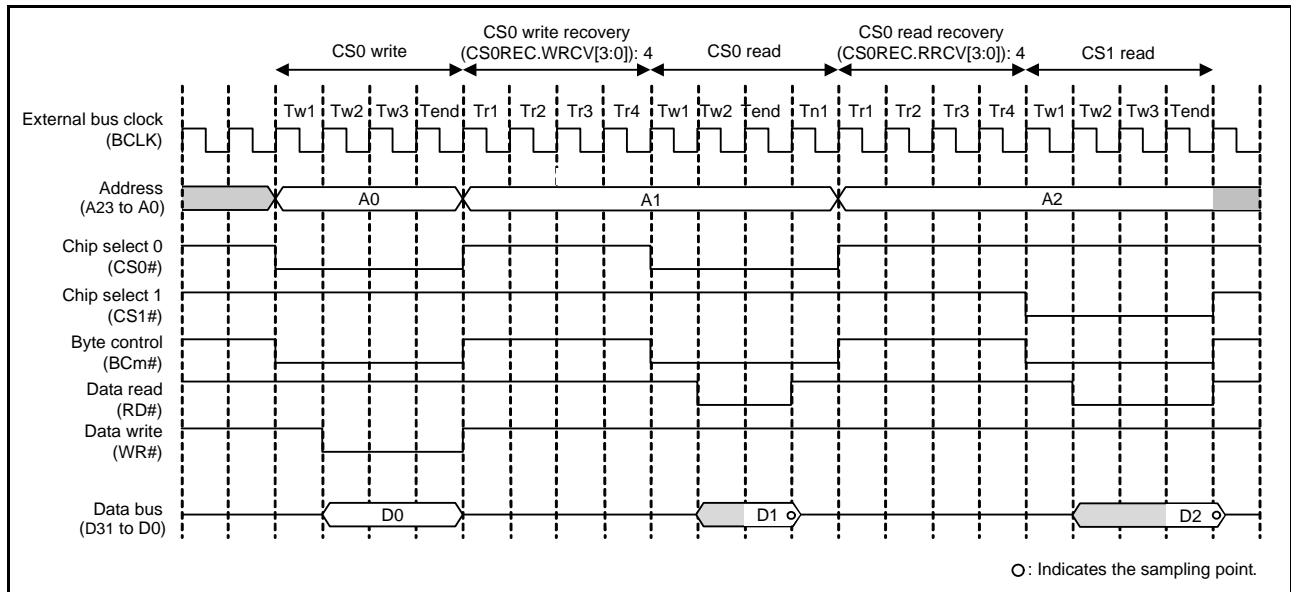


Figure 16.36 Example of Recovery Cycle Insertion (with Separate Bus Interface) (m = 0 to 3)

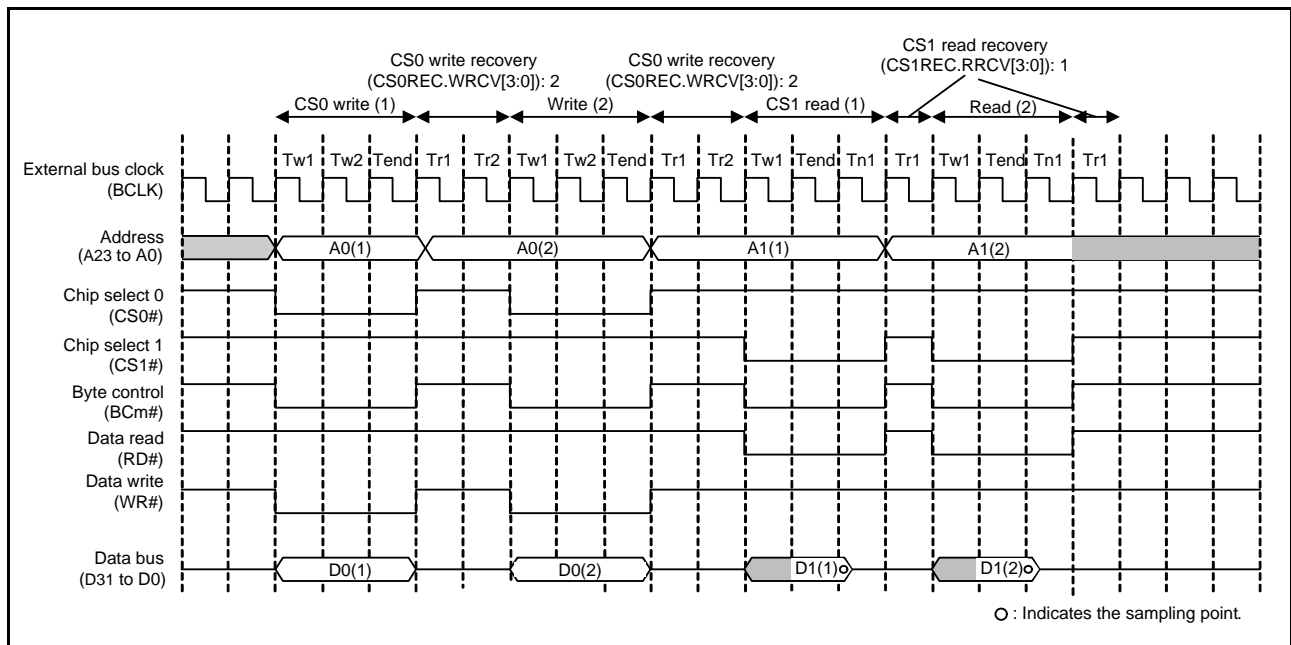
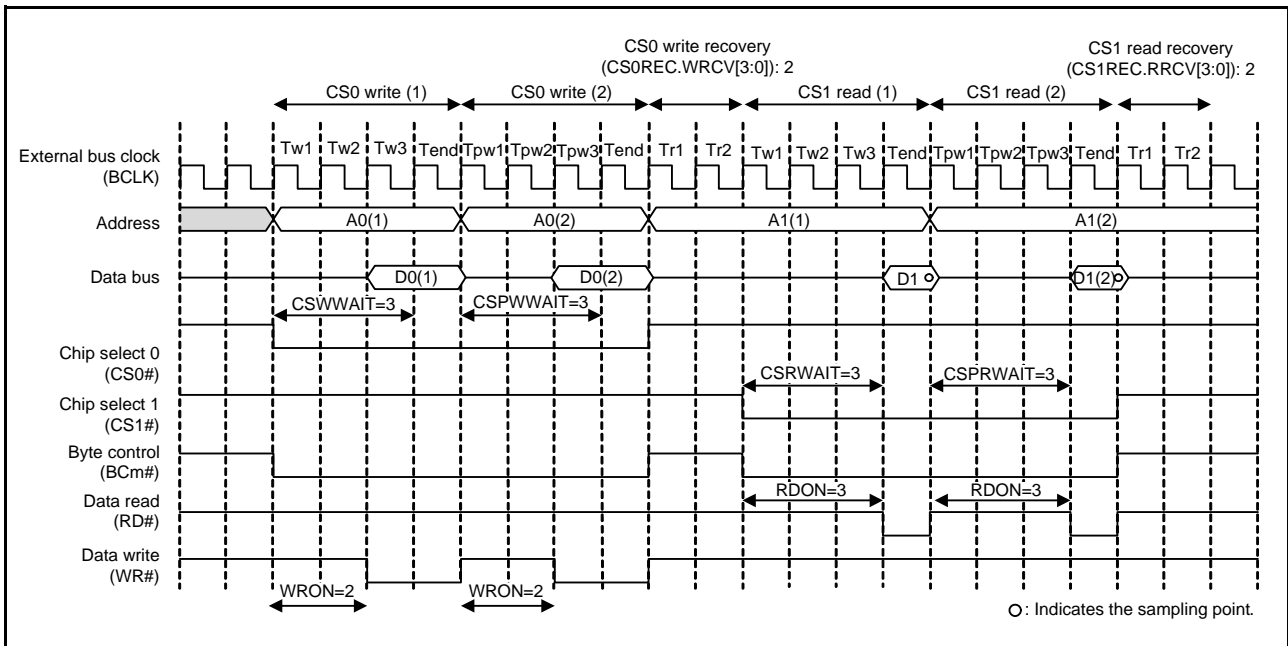


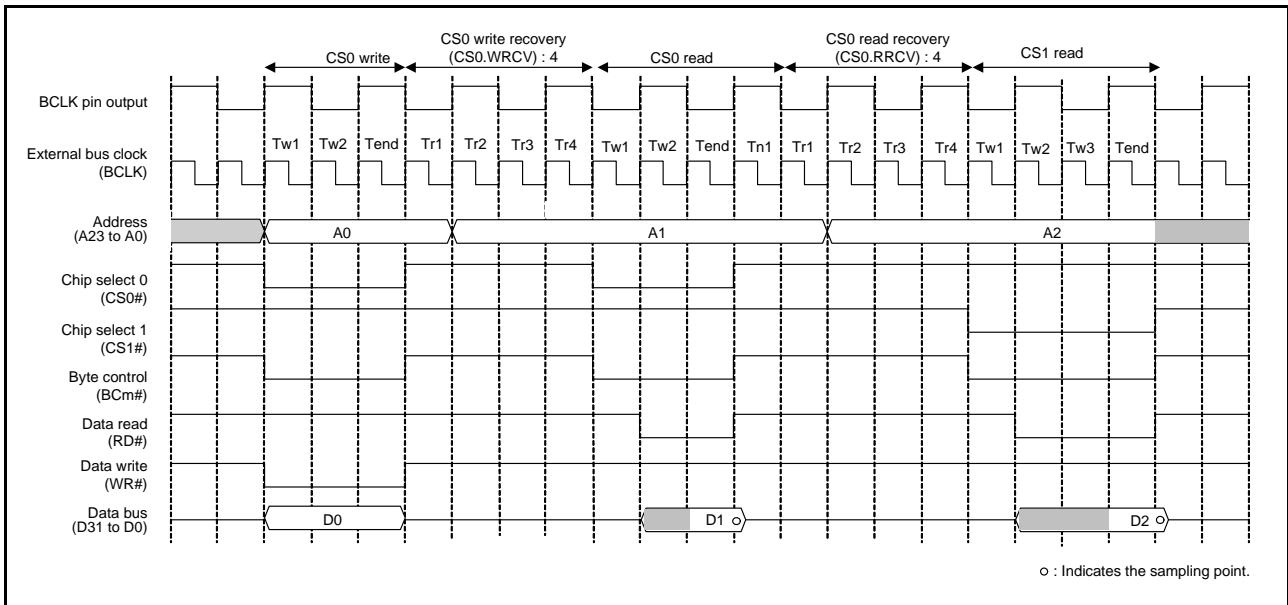
Figure 16.37 Example of Recovery Cycle Insertion When a Bus Access is Split (with Separate Bus Interface, Normal Access) (m = 0 to 3)





**Figure 16.38** Example of Recovery Cycle Insertion When a Bus Access is Split (with Separate Bus Interface, Page Access) (m = 0 to 3)

Figure 16.39 shows examples of operations when the BCLK pin output selection bits are set for frequency-division of BCLK by 2.



**Figure 16.39** Example of Operation for Recovery Cycles when the BCLK Pin Output Selection Bits Are Set for Frequency-Division of BCLK by 2 (For the Case of Normal Access Through a Separate Bus Interface; m = 0 to 3)

With the address/data multiplexed I/O interface, recovery cycles are inserted in the same way as that with the separate bus interface. Figure 16.40 and Figure 16.41 show examples of recovery cycle insertion with the address/data multiplexed I/O interface.

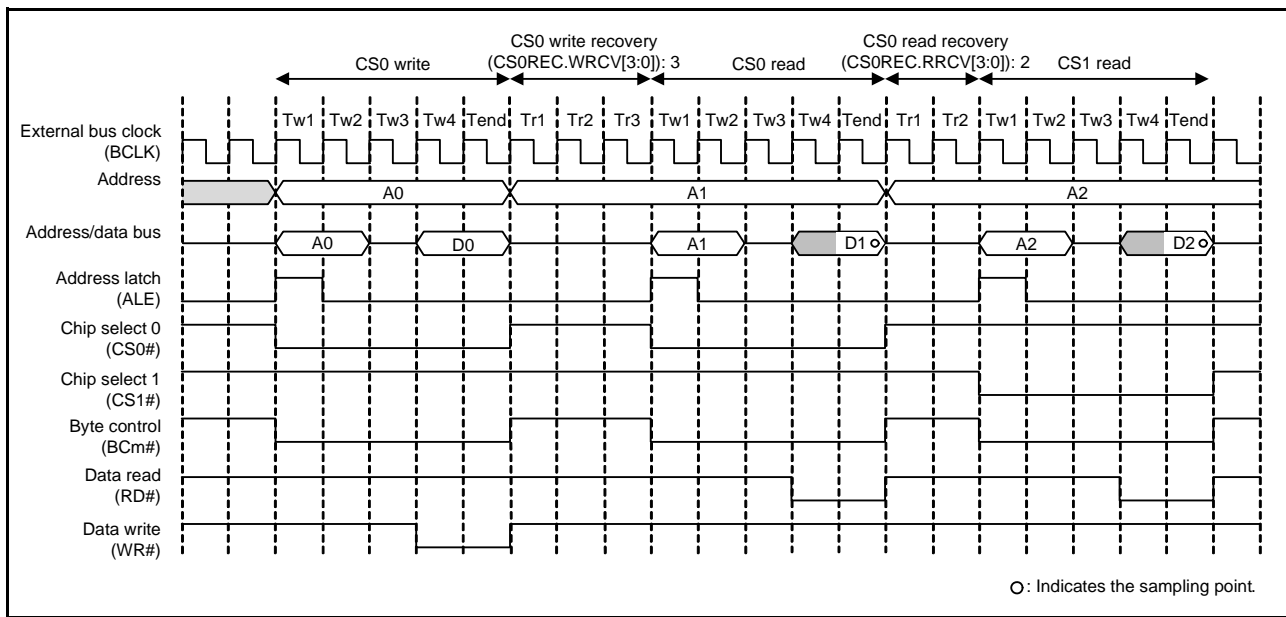


Figure 16.40 Example of Recovery Cycle Insertion (with Address/Data Multiplexed I/O Interface; m = 0, 1)

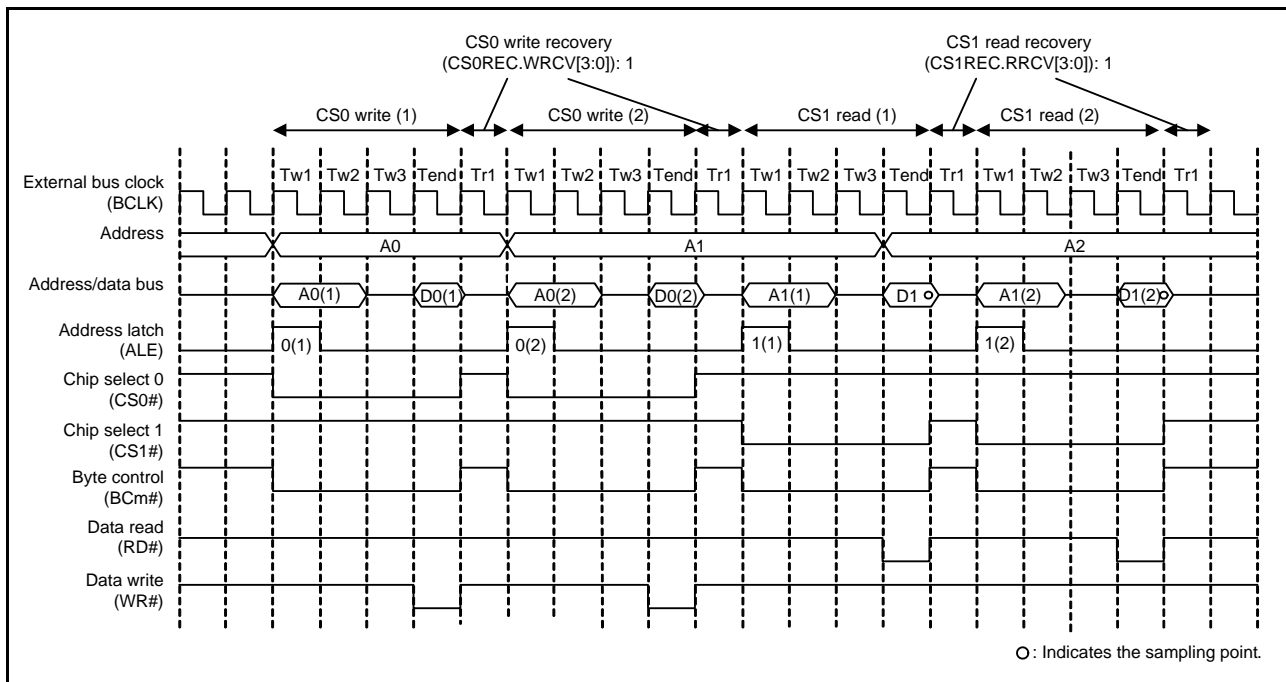


Figure 16.41 Example of Recovery Cycle Insertion When a Bus Access is Split (with Address/Data Multiplexed I/O Interface; m = 0, 1)

### 16.5.5 No Access State

When no external address space is accessed, CSn#, BCn#, WRn#, and RDn# signals are high, ALE signal is low, and D31 to D0 are in the high-impedance state.

### 16.5.6 Write Buffer Function (External Bus)

The internal main bus is released by writing data to the write buffer before the write access is completed, which allows the next round of bus access to start. However, if the following round of bus access is to an external address space or to a register of the external bus controller, it is suspended until the external bus operations already in progress are completed. Figure 16.42 shows an example of operation when the write-buffer function is in use. When this function is in use, if the next operation after an external write is internal access, the internal access (access to on-chip memory or a peripheral module) is executed in parallel with the external write, i.e. without waiting for completion of the latter operation.

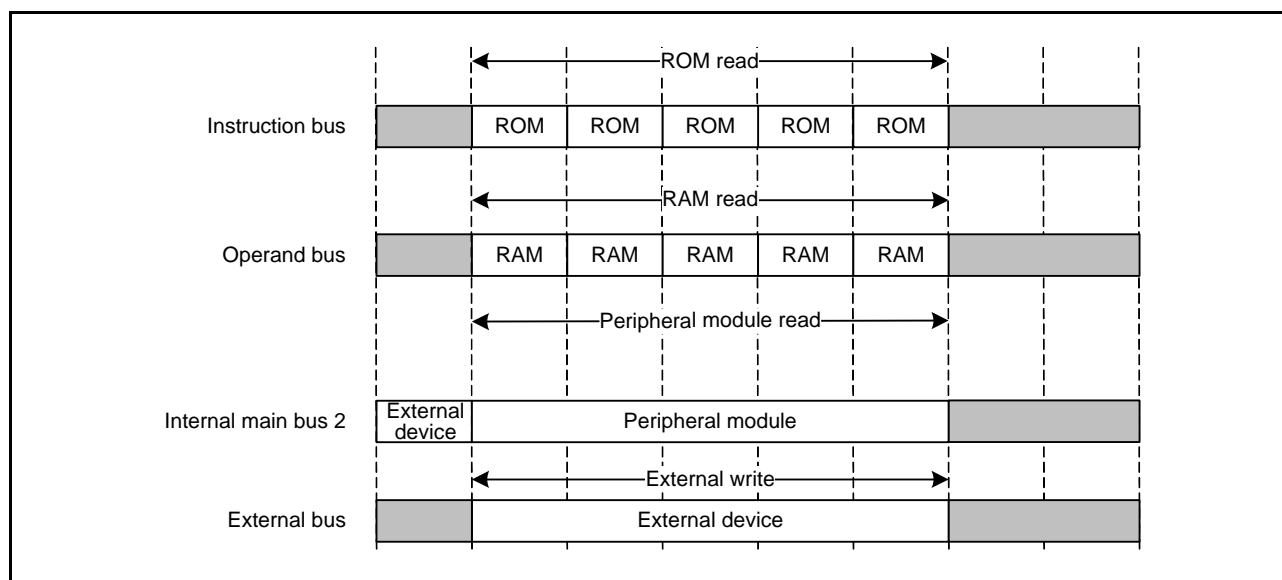


Figure 16.42 Example of Operation when the Write-Buffer Function is in Use

### 16.5.7 Limitations

#### (1) Limitations on Using Separate Bus Interface

- Limitations that apply to various bits of CSn wait control register 1 (CSnWCR1) and CSn wait control register 2 (CSnWCR2) at the times of normal and page accesses are listed in Table 16.9.

Even if the setting of the page-read access enable bit in the CSn mode register or the page-write access enable bit in the CSn mode register selects permission (CSnMOD.PRENB = 1 or CSnMOD.PWENB = 1), the first round of access for page access and the access that does not fall within the scope of page access are normal access operation, and thus limitations on normal access must be satisfied.

Table 16.9 Limitations at the Time of Normal and Page Access

Limitations at the Time of Normal Access		Limitations at the Time of Page Access	
Reading	Writing	Reading	Writing
CSn[2:0] ≤ CSRWAIT	1 ≤ WDon[2:0]	CSn[2:0] ≤ CSPRWAIT	1 ≤ WDon[2:0]
RDon[2:0] ≤ CSRWAIT	CSn[2:0] ≤ CSWWAIT	RDon[2:0] ≤ CSPRWAIT	CSn[2:0] ≤ CSPWWAIT
CSn[2:0] ≤ RDon	WRon[2:0] ≤ CSWWAIT	CSn[2:0] ≤ RDon	WRon[2:0] ≤ CSPWWAIT
	WDon[2:0] ≤ CSWWAIT		WDon[2:0] ≤ CSPWWAIT
	WDOFF[2:0] ≤ CSWOFF		WDOFF[2:0] ≤ CSWOFF
	WDon[2:0] ≤ WRon		WDon[2:0] ≤ WRon
	CSn[2:0] ≤ WRon		CSn[2:0] ≤ WRon

- When two or more external bus access cycles are required for a single transfer request from a bus master and the recovery cycle insertion condition is satisfied with page read access enabled (CSnMOD.PRENB = 1) or page write access enabled (CSnMOD.PWENB = 1), recovery cycles are not inserted between bus access cycles but inserted only after the last bus access cycle of the transfer.

## (2) Limitations on Using Address/Data Multiplexed Bus Interface

- In the address/data multiplexed I/O space, page accesses are invalid. If a page access setting is specified, the setting is ignored and the normal read or write operation is performed.
- When the address/data multiplexed I/O interface is set, the BSIZE[1:0] bits in CSnCR should not be set to the 32-bit bus space. If set, the operation cannot be guaranteed.

**Table 16.10 Limitations at the Time of Normal and Page Access**

Limitations at the Time of Normal Access	
Reading	Writing
$CSON[2:0] \leq CSRWAIT$	$CSON[2:0] \leq CSWWAIT$
$RDON[2:0] \leq CSRWAIT$	$WRON[2:0] \leq CSWWAIT$
$CSON[2:0] \leq RDON$	$WDON[2:0] \leq CSWWAIT$
$AWAIT[1:0]+2 \leq RDON$	$WDOFF[2:0] \leq CSWOFF$
$CSON[2:0] \leq AWAIT$	$WDON[2:0] \leq WRON$
	$CSON[2:0] \leq WRON$
	$AWAIT[1:0]+2 \leq WRON$
	$AWAIT[1:0]+2 \leq WDON$
	$CSON[2:0] \leq AWAIT$

## (3) Limitation when a Pin is Multiplexed between A0 and BC0# Functions

When the A0 and BC0# pin functions share the same pin, setting the single write strobe mode is prohibited in the 8-bit bus space; otherwise the operation is not guaranteed.

## (4) Limitations when 1/2 BCLK is Selected with BCLK Pin Output Select Bit

When 1/2 BCLK is selected through the BCLK pin output select bit, the external bus access cycle starts at the rising edge of the BCLK pin output. However, when two or more external bus access cycles are generated for a single transfer request from a bus master, the second or subsequent external bus access cycle may start at the falling edge of the BCLK pin output depending on the wait cycle settings. Make appropriate register settings according to the specifications of the device to be connected.

## (5) Prohibition of Access that Spans Areas of Address Space

Single access that spans several areas of the address space is prohibited, and operation of such an access is not guaranteed. Setting must be made so that two areas are not accessed at the same time by a single word or longword access.

## (6) Restrictions on RMPA and String-Manipulation Instructions

- Although the external space has a per-area ending-switching facility (only for data), the allocation of data to be handled by RMPA or string-manipulation instructions to an area where the endian differs from that of the chip is prohibited, and operation is not guaranteed if this restriction is not observed. If data to be handled by RMPA or string-manipulation instructions are allocated to the external space, they must be allocated to areas where the endian setting is the same as that for the chip.
- 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.

## (7) Restriction on Instruction Code

When the endian setting for each area is different from that for the chip, no instruction code can be allocated in the area. The instruction code should be allocated to the external space whose endian setting is the same as that for the chip.

## 16.6 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.

### 16.6.1 Types 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.

#### 16.6.1.1 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 of the following types leads to illegal address access errors.

- Access to areas of external space for which operation has been disabled (CSnCR.EXENB = 0)
- With respect to areas other than those described above, access to illegal address ranges  
The address ranges where access will lead to illegal address access errors are indicated in Table 16.11.

#### 16.6.1.2 Timeout

When the timeout detection enable bit (TOEN) in the bus error monitoring enable register (BEREN) is set to 1, bus access that is not completed within 768 cycles leads to a timeout error.

- CS areas (CS0 to CS7): Bus access is not completed (the WAIT# signal is not negated) within 768 external bus clock (BCLK) cycles from the start of the access.  
Once a timeout error occurs, accesses from the bus master are rejected for 256 BCLK cycles. If multiple external bus accesses are generated with a single request from the bus master during the transfer, the bus accesses cannot be stopped by a timeout. At this time, timeout errors may occur repeatedly.
- Internal peripheral buses (2 and 3): Bus access is not completed within 768 peripheral module clock (PCLKB) cycles from the start of the access.  
Once a timeout error occurs, accesses from the bus master are rejected 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.  
Once a timeout error occurs, accesses from the bus master are rejected for 256 FCLK cycles.

### 16.6.2 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 IERn register in the ICU can specify whether to generate an interrupt in the case of a bus error.

### 16.6.3 Conditions Leading to Bus Errors

Table 16.11 lists the types 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 on 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 BERSRn is cleared.

**Table 16.11 Types of Bus Errors**

Address	Type of Area		Type of Error			
			Illegal Address Access		Timeout	
	On-chip ROM Enabled	On-chip ROM Disabled	On-chip ROM Enabled	On-chip ROM Disabled	On-chip ROM Enabled	On-chip ROM Disabled
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		—	
000A 0000h to 000B FFFFh	Internal peripheral bus 3		Δ		—	
000C 0000h to 000D FFFFh	Reserved area		Δ		—	
000E 0000h to 000F FFFFh	Reserved area		—		—	
0010 0000h to 00FF FFFFh	Internal peripheral bus 6	Reserved area	Δ	○	—	—
0100 0000h to 07FF FFFFh	External bus (CS1 to CS7)		[IA]		[TO]	
0800 0000h to 0FFF FFFFh	Reserved area		○		—	—
1000 0000h to 7FFF FFFFh	Reserved area		○		—	—
8000 0000h to FEFF FFFFh	Memory bus 2	Reserved area	—	○	—	—
FF00 0000h to FF7F FFFFh		External bus (CS0)	—	[IA]	—	[TO]
FF80 0000h to FFFF FFFFh			—		—	

—: A bus error does not result.

Δ: A bus error may or may not result.

○: A bus error results.

[IA]: Access to this area leads to detection of a bus error if operation for this area is disabled (CSnCR.EXENB = 0; n = 0 to 7).

[TO]: Bus access not being completed within 768 cycles leads to detection of a bus error.

Note: • The capacity of the RAM, E2 DataFlash, and ROM differs depending on the product. For details, see section 42, RAM, section 43, Flash Memory.

## 17. Memory-Protection Unit (MPU)

### 17.1 Overview

The RX CPU incorporates a memory-protection unit that checks the addresses of CPU access to the overall address space (0000 0000h to FFFF FFFFh).

Access-control information can be set for up to eight regions, and permission for access to each region is in accord with this information. The default response to the detection of access to a region where permission has not been set is the generation of a memory-protection error.

The supported access-control information for the individual regions consists of permission to read, permission to write, and permission to execute. This access-control information is effective when the processor mode of the CPU is user mode. Memory protection is not applied when the CPU is in supervisor mode.

Table 17.1 lists the specifications of the memory-protection unit, and Figure 17.1 shows a block diagram of the memory-protection unit.

**Table 17.1 Specifications of Memory Protection**

Specifications	Description
Region to be covered by memory protection and processor mode	0000 0000h to FFFF FFFFh (in user mode) No memory protection in supervisor mode
Number of regions	8
Page size (smallest unit of protection)	16 bytes
Specifying addresses of individual regions	Setting the page numbers where regions start and end
Setting to make memory protection effective or ineffective in individual regions	A V (valid) bit in each region-n end page number register (REPAGEn) makes the settings effective or ineffective for the corresponding region (n = 0 to 7).
Access-control information settings for individual regions	Instruction execution: Permission to execute Operand access: Permission to read, permission to write
Start of memory-protection operations	After the memory-protection unit has been enabled, access monitoring start-ing up with the transition to user mode.
Memory-protection error processing	Generation of access exceptions
Addresses where memory-protection errors are generated	Address in instruction execution: The PC value is preserved on the stack. Address in operand access: The address is stored in the data memory-protection error address register (MPDEA).
Determining the reasons for memory-protection errors	The memory-protection error status register (MPESTS) holds indicators of the reason.
Background region setting	Access-control information can be set for the background region (the whole address space).
Processing where regions overlap	The access-control information for access to an overlap between regions is the logical OR of the attributes for the given regions, and permission is given priority.

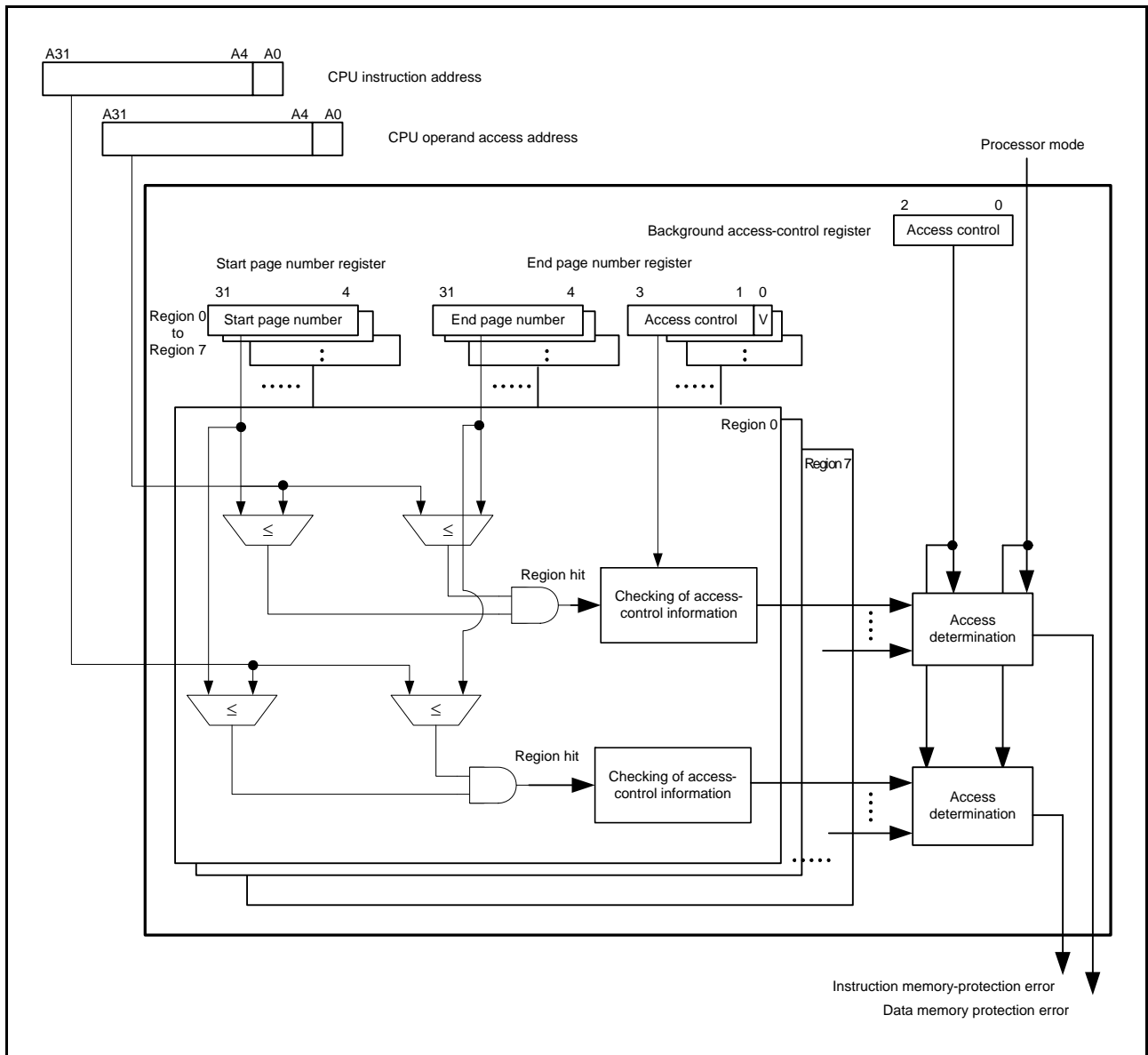


Figure 17.1 Block Diagram of the Memory-Protection Unit



### 17.1.1 Types of Access Control

There are three types of access control information: permission for instruction execution, permission to read operands, and permission to write operands. Violations of these types of access control are only detected when programs are running in user mode. Violations are not detected when programs are running in supervisor mode.

### 17.1.2 Regions for Access Control

Up to eight regions for access control are definable. Settings of the range of memory for each access-control region are made in the corresponding region-n start page number register (RSPAGEn) and region-n end page number register (REPAGEn), where n = 0 to 7.

The minimum unit for control of access is the “page”, by which the address space is divided into 16-byte units. The 28 higher-order bits ([31:4]) of the address [31:0] bits correspond to the page number.

The REPAGEn register specifies the access-control information for each area and whether the area is enabled or not.

### 17.1.3 Background Region

“Background region” refers to the whole address space (0000 0000h to FFFF FFFFh). Access-control information for the background region is set in the background-region access-control register (MPBAC). In contrast to the access-control information for the eight individual regions, protection information for the background region is effective as long as memory protection is enabled (the MPEN bit in the MPEN register is 1).

### 17.1.4 Overlap between Regions

In cases of overlap between multiple regions, the access-control information becomes the logical OR of the access-control bits for the overlapping regions (including the background region), with permission given priority.

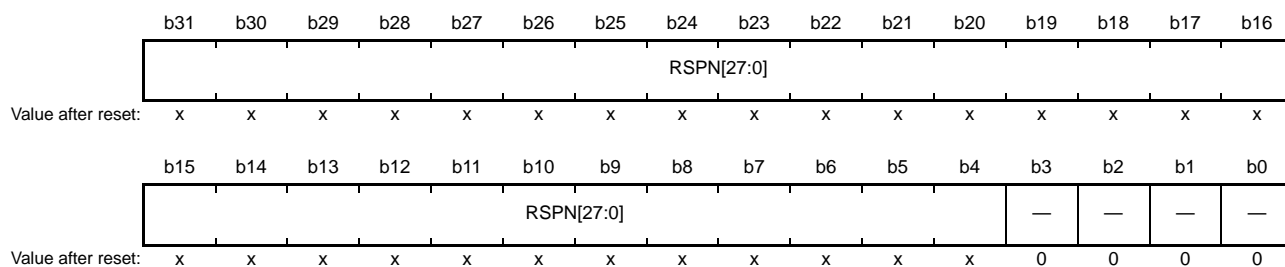
### 17.1.5 Instructions and Data that Span Regions

Operations in response to the detection of memory-protection errors when instructions or data span regions for which different access-control settings have been made are undefined. Ensure that instructions and data do not span regions for which different access-control settings have been made.

## 17.2 Register Descriptions

### 17.2.1 Region-n Start Page Number Register (RSPAGEn) (n = 0 to 7)

Addresses: RSPAGE0 0008 6400h, RSPAGE1 0008 6408h, RSPAGE2 0008 6410h, RSPAGE3 0008 6418h  
 RSPAGE4 0008 6420h, RSPAGE5 0008 6428h, RSPAGE6 0008 6430h, RSPAGE7 0008 6438h



x: Undefined

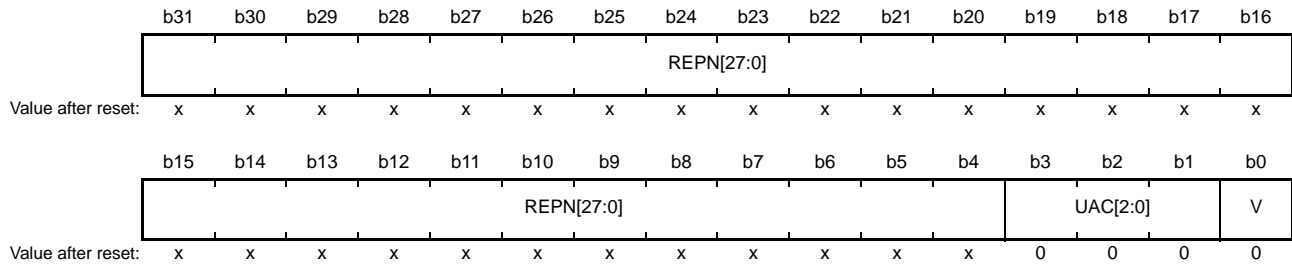
Bit	Symbol	Bit Name	Function	R/W
b3 to b0	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b31 to b4	RSPN[27:0]	Region-Start Page Number	Page number where the region starts, for use in region determination	R/W

#### RSPN[27:0] Bits (Region-Start Page Number)

These bits specify the page number where the region starts.

### 17.2.2 Region-n End Page Number Register (REPAGEn) (n = 0 to 7)

Addresses: REPAGE0 0008 6404h, REPAGE1 0008 640Ch, REPAGE2 0008 6414h, REPAGE3 0008 641Ch  
 REPAGE4 0008 6424h, REPAGE5 0008 642Ch, REPAGE6 0008 6434h, REPAGE7 0008 643Ch



x: Undefined

Bit	Symbol	Bit Name	Function	R/W
b0	V	Valid Bit	0: Region setting invalid 1: Region setting valid	R/W
b3 to b1	UAC[2:0]	Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R/W
b31 to b4	REPN[27:0]	Region-End Page Number	Page number where the region ends, for use in region determination	R/W

#### V Bit (Valid Bit)

This bit enables or disables the settings for the corresponding region.

This bit is cleared to 0 when the region invalidation operation register (MPOPI) invalidates all access-controlled areas.

#### UAC[2:0] Bits (Access Control Bits in User Mode)

These bits specify the access control in user mode.

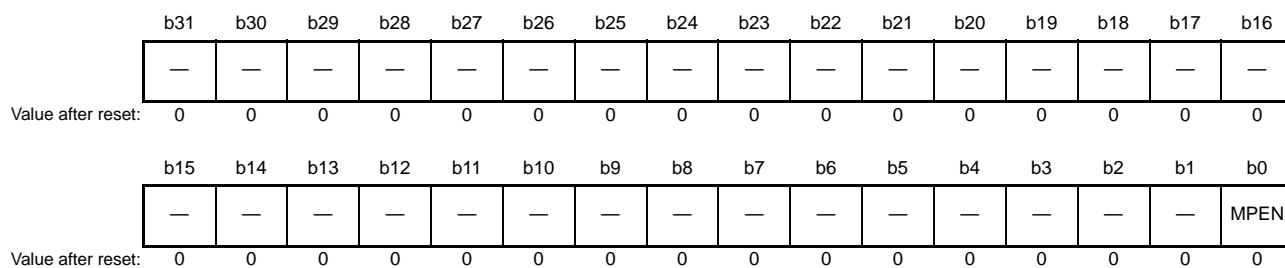
#### REPN[27:0] Bits (Region-End Page Number)

These bits specify the page number where the region ends.

Specify a value that is greater than or equal to the page number where the corresponding region starts. The page specified by the region-end page number is part of the target region for memory protection.

### 17.2.3 Memory-Protection Enable Register (MPEN)

Address: 0008 6500h



Bit	Symbol	Bit Name	Function	R/W
b0	MPEN	Memory-Protection Enable	1: The memory protection is enabled. 0: The memory protection is disabled.	R/W
b31 to b1	—	Reserved	The read value is 0. The write value should always be 0	R/W

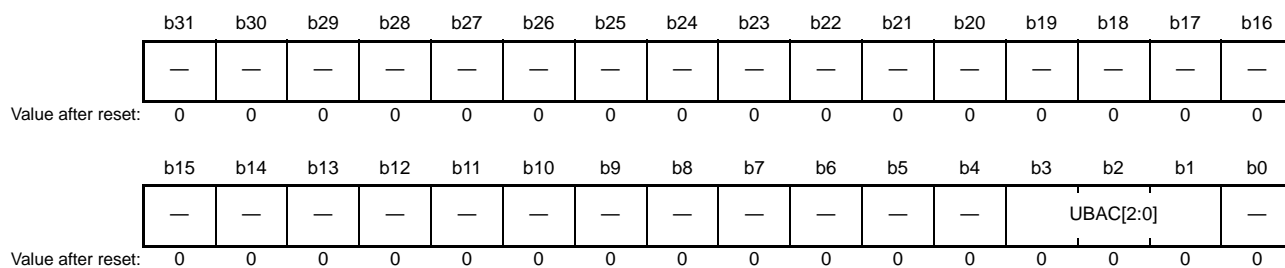
#### MPEN Bit (Memory-Protection Enable)

This bit enables or disables the memory protection.

After 1 has been written to this bit, address checking for memory protection by the CPU starts on the execution of a branch instruction (RTE and RTFI) that shifts operation to the user mode.

### 17.2.4 Background Access Control Register (MPBAC)

Address: 0008 6504h



Bit	Symbol	Bit Name	Function	R/W
b0	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b3 to b1	UBAC[2:0]	Background Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R/W
b31 to b4	—	Reserved	The read value is 0. The write value should always be 0.	R/W

#### UBAC[2:0] Bits (Background Access Control Bits in User Mode)

These bits specify the background access control in user mode.

### 17.2.5 Memory-Protection Error Status-Clearing Register (MPECLR)

Address: 0008 6508h

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	CLR
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
b0	CLR	Error Status-Clearing	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done. 1: The DRW, DA, and IA bits in the MPESTS are cleared to 0.	R/W
b31 to b1	—	Reserved	The read value is 0. The write value should always be 0.	R/W

#### CLR Bit (Error Status-Clearing)

This bit clears the data read/write bit (DRW), the data memory-protection error generated bit (DA), and the instruction memory-protection error generated bit (IA) in the memory-protection error status register (MPESTS) to 0.

## 17.2.6 Memory-Protection Error Status Register (MPESTS)

Address: 0008 650Ch

	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
	—	—	—	—	—	—	—	—	—	—	—	—	—	DRW	DA	IA
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Function	R/W
b0	IA	Instruction Memory-Protection Error Generated Bit	0: No instruction memory-protection error was generated. 1: Instruction memory-protection error was generated.	R
b1	DA	Data Memory-Protection Error Generated Bit	0: No data memory-protection error was generated. 1: Data memory-protection error was generated.	R
b2	DRW	Data Read/Write Bit	0: Data were read. 1: Data were written.	R
b31 to b3	—	Reserved	The read value is 0. The write value should always be 0.	R/W

### IA Bit (Instruction Memory-Protection Error Generated Bit)

This bit indicates the state of memory-protection error generation by instruction execution.

Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

### DA Bit (Data Memory-Protection Error Generated Bit)

This bit indicates the state of memory-protection error generation by operand access.

Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

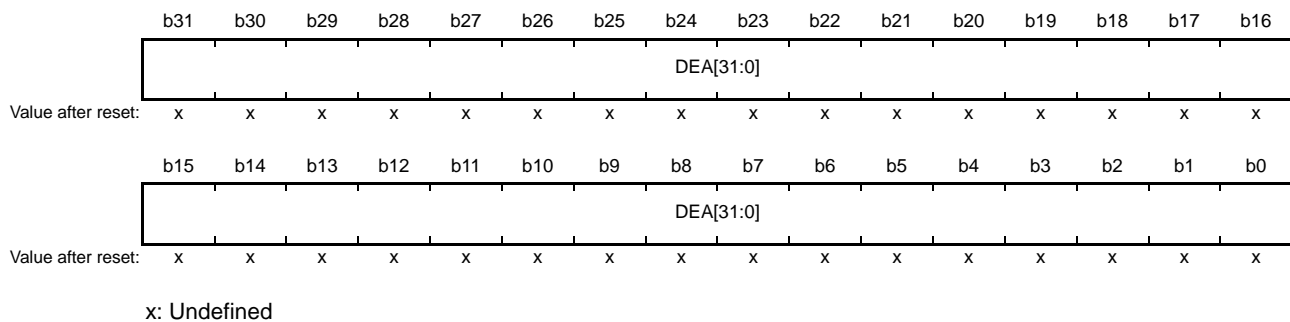
### DRW Bit (Data Read/Write Bit)

For a memory-protection error produced by operand access, this bit indicates the read/write attribute of the access operation. This bit is only valid when the data memory-protection error generated bit (DA) is 1.

Setting the error status-clearing bit (CLR) in the memory-protection error status-clearing register (MPECLR) to 1 clears this bit to 0.

### 17.2.7 Data Memory-Protection Error Address Register (MPDEA)

Address: 0008 6514h



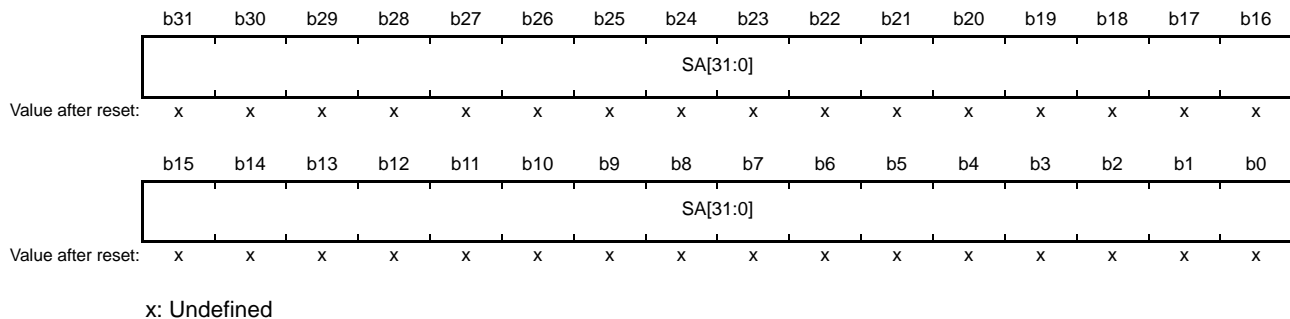
Bit	Symbol	Bit Name	Function	R/W
b31 to b0	DEA[31:0]	Data Memory-Protection Error Address	Data memory-protection error address	R

#### DEA[31:0] Bits (Data Memory-Protection Error Address)

These bits retain the address for which operand access generated a memory-protection error.

### 17.2.8 Region Search Address Register (MPSA)

Address: 0008 6520h



Bit	Symbol	Bit Name	Function	R/W
b31 to b0	SA[31:0]	Region Search Address	Address for region searching	R/W

#### SA[31:0] Bits (Region Search Address)

These bits specify the address for use in comparison with region-start addresses in the region-n start page number registers (RSPAGEn) and region-end addresses in the region-n end page number registers (REPAGEn).



### 17.2.9 Region Search Operation Register (MPOPS)

Address: 0008 6524h

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

Bit	Symbol	Bit Name	Function	R/W
b0	S	Region Search Operation	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done. 1: A region-search operation proceeds.	R/W
b15 to b1	—	Reserved	The read value is 0. The write value should always be 0.	R/W

#### S Bit (Region Search Operation)

Setting this bit to 1 makes the memory-protection unit perform a region-search operation. The address specified in the region search address register (MPSA) is compared with the address information for individual regions to search for a hitting region.

The result of searching is stored in the data-hit region bits (HITD[7:0]) of the data-hit region register (MHITD).

Moreover, the logical OR of the respective access control bits for hitting regions is stored in the data-hit region access control bits (UHACD[2:0]) in user mode.

### 17.2.10 Region Invalidation Operation Register (MPOPI)

Address: 0008 6526h

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

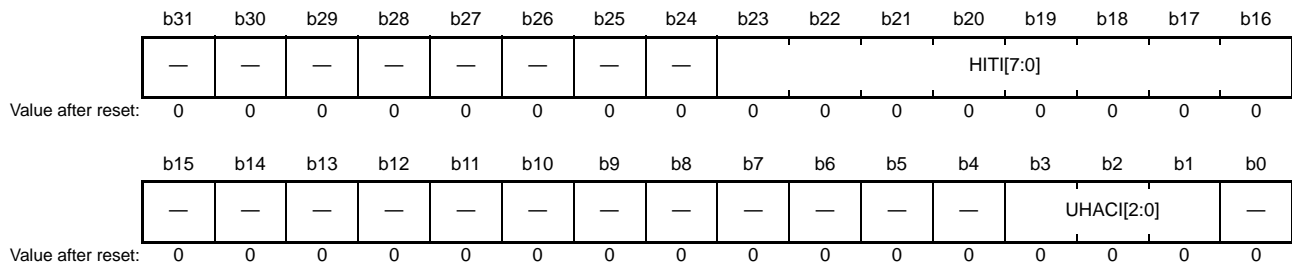
Bit	Symbol	Bit Name	Function	R/W
b0	INV	Region Invalidate Start	[Reading] 0: Fixed value for reading [Writing] 0: Nothing is done. 1: All access-controlled areas are invalidated.	R/W
b15 to b1	—	Reserved	The read value is 0. The write value should always be 0.	R/W

#### INV Bit (Region Invalidate Start)

Setting this bit to 1 clears the valid (V) bits in all of the region-n end page number registers (REPAGEn) to 0. After a V bit is cleared to 0, all settings other than background access-control settings are invalid.

### 17.2.11 Instruction-Hit Region Register (MHITI)

Address: 0008 6528h



Bit	Symbol	Bit Name	Function	R/W
b0	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b3 to b1	UHACI[2:0]	Instruction-Hit Region Access Control Bits in User Mode	b3 0: Reading prohibited 1: Read permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution is permitted.	R
b15 to b4	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b23 to b16	HITI[7:0]	Instruction-Hit Region	When the instruction memory-protection error generated (IA) bit in the MPESTS = 1, [b23:b16] = 0000 0000b indicates that attempted access to the background region led to an instruction memory-protection error.  Other than above b23 0: Instruction memory-protection error was not generated in region 7. 1: Instruction memory-protection error was generated in region 7. b22 0: Instruction memory-protection error was not generated in region 6. 1: Instruction memory-protection error was generated in region 6. b21 0: Instruction memory-protection error was not generated in region 5. 1: Instruction memory-protection error was generated in region 5. b20 0: Instruction memory-protection error was not generated in region 4. 1: Instruction memory-protection error was generated in region 4. b19 0: Instruction memory-protection error was not generated in region 3. 1: Instruction memory-protection error was generated in region 3. b18 0: Instruction memory-protection error was not generated in region 2. 1: Instruction memory-protection error was generated in region 2. b17 0: Instruction memory-protection error was not generated in region 1. 1: Instruction memory-protection error was generated in region 1. b16 0: Instruction memory-protection error was not generated in region 0. 1: Instruction memory-protection error was generated in region 0.	R
b31 to b24	—	Reserved	The read value is 0. The write value should always be 0.	R/W

**UHACI[2:0] Bits (Instruction-Hit Region Access Control Bits in User Mode)**

These bits hold the user-mode access control bits (REPAGEn.UAC[2:0]) for the region where the instruction memory-protection error was generated.

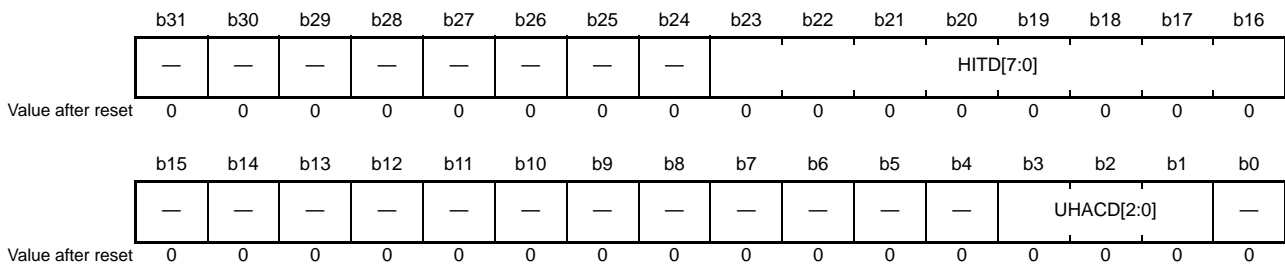
If the error was generated in an overlap between regions, the value stored here is the logical OR of the user-mode access control bits for the corresponding regions (including the background region).

**HITI[7:0] Bits (Instruction-Hit Region)**

These bits indicate the region where an instruction memory-protection error was generated. These bits are set to 0000 0000b in response to the generation of an instruction memory-protection error in the background region.

**17.2.12 Data-Hit Region Register (MHITD)**

Address: 0008 652Ch



Bit	Symbol	Bit Name	Function	R/W
b0	—	Reserved	The read value is 0. The write value should always be 0.	R/W
b3 to b1	UHACD[2:0]	Data-Hit Region Access Control Bits in User Mode	b3 0: Reading prohibited 1: Reading permitted b2 0: Writing prohibited 1: Writing permitted b1 0: Execution prohibited 1: Execution permitted	R
b15 to b4	—	Reserved	The read value is 0. The write value should always be 0.	R/W

Bit	Symbol	Bit Name	Function	R/W
b23 to b16	HITD[7:0]	Data-Hit Region	<p>When the data memory-protection error generated (DA) bit = 1, [b23:b16] = 0000 0000b indicates that attempted access to the background region led to a data memory-protection error.</p> <p>Other than above</p> <p>b23 0: Neither a data memory-protection error nor a search hit was generated in region 7. 1: A data memory-protection error or search hit was generated in region 7.</p> <p>b22 0: Neither a data memory-protection error nor a search hit was generated in region 6. 1: A data memory-protection error or search hit was generated in region 6.</p> <p>b21 0: Neither a data memory-protection error nor a search hit was generated in region 5. 1: A data memory-protection error or search hit was generated in region 5.</p> <p>b20 0: Neither a data memory-protection error nor a search hit was generated in region 4. 1: A data memory-protection error or search hit was generated in region 4.</p> <p>b19 0: Neither a data memory-protection error nor a search hit was generated in region 3. 1: A data memory-protection error or search hit was generated in region 3.</p> <p>b18 0: Neither a data memory-protection error nor a search hit was generated in region 2. 1: A data memory-protection error or search hit was generated in region 2.</p> <p>b17 0: Neither a data memory-protection error nor a search hit was generated in region 1. 1: A data memory-protection error or search hit was generated in region 1.</p> <p>b16 0: Neither a data memory-protection error nor a search hit was generated in region 0. 1: A data memory-protection error or search hit was generated in region 0.</p>	R
b31 to b24	—	Reserved	The read value is 0. The write value should always be 0.	R/W

### UHACD[2:0] Bits (Data-Hit Region Access Control Bits in User Mode)

These bits hold the user-mode access control bits (REPAGEn.UAC[2:0]) that have been set for the region where a data memory-protection error was generated or the region that produced a hit in region searching.

When an error is generated in an overlap between regions or a hit was generated in region searching, the value stored here is the logical OR of the user-mode access control bits for the corresponding regions (including the background region).

### HITD[7:0] Bits (Data-Hit Region)

These bits indicate the region where a data memory-protection error was generated or the region that produced a hit in a region search. These bits are set to 0000 0000b for a data memory-protection error generated in the background region.

Note: When access to a register of memory protection unit in user mode generates a data memory-protection error, the value in this register is cleared to 0000 0000h.

## 17.3 Functions

### 17.3.1 Memory Protection

Memory protection means monitoring, in accord with the access-control information that has been set for the individual access-control regions and the background region, whether or not access by programs running in user mode violates the access-control settings. The memory-protection unit notifies the CPU of access-control violations (or memory-protection errors) when they are detected, causing the CPU to start access-exception processing.

Memory protection is enabled by setting the memory-protection enable (MPEN) bit in the memory-protection enable (MPEN) register to 1.

An instruction memory-protection error is generated on detection of an instruction-execution violation and a data memory-protection error is generated on detection of an operand-access reading or writing violation. Operand access that leads to a data memory-protection error is not actually executed.

### 17.3.2 Region Search

Region search means enquiry as to which of the eight specified access regions was “hit” and how the access-control information (permission to execute, to read, and to write) is set.

When the region search operation (S) bit in the region-search operation (MPOP) register is set to 1, the address specified in the region search address (MPSA) register is compared with the addresses for the individual regions. After a region search is executed, the data-hit region register (MHITD) indicates the logical OR of the access-control information for the region which was “hit” and for the other regions.

### 17.3.3 Protection of Registers Related to the Memory-Protection Unit

Registers related to the memory-protection unit are not accessible through means of access other than operand access by the CPU. Attempted access to registers related to the memory-protection unit in user mode through operand access by the CPU leads to a data memory-protection error regardless of whether or not memory protection is in effect at the given location.

### 17.3.4 Flow for Determination of Access by the Memory-Protection Function

Figure 17.2 shows the flow of determination in the case of data access and Figure 17.3 shows the flow of determination in the case of instruction access.

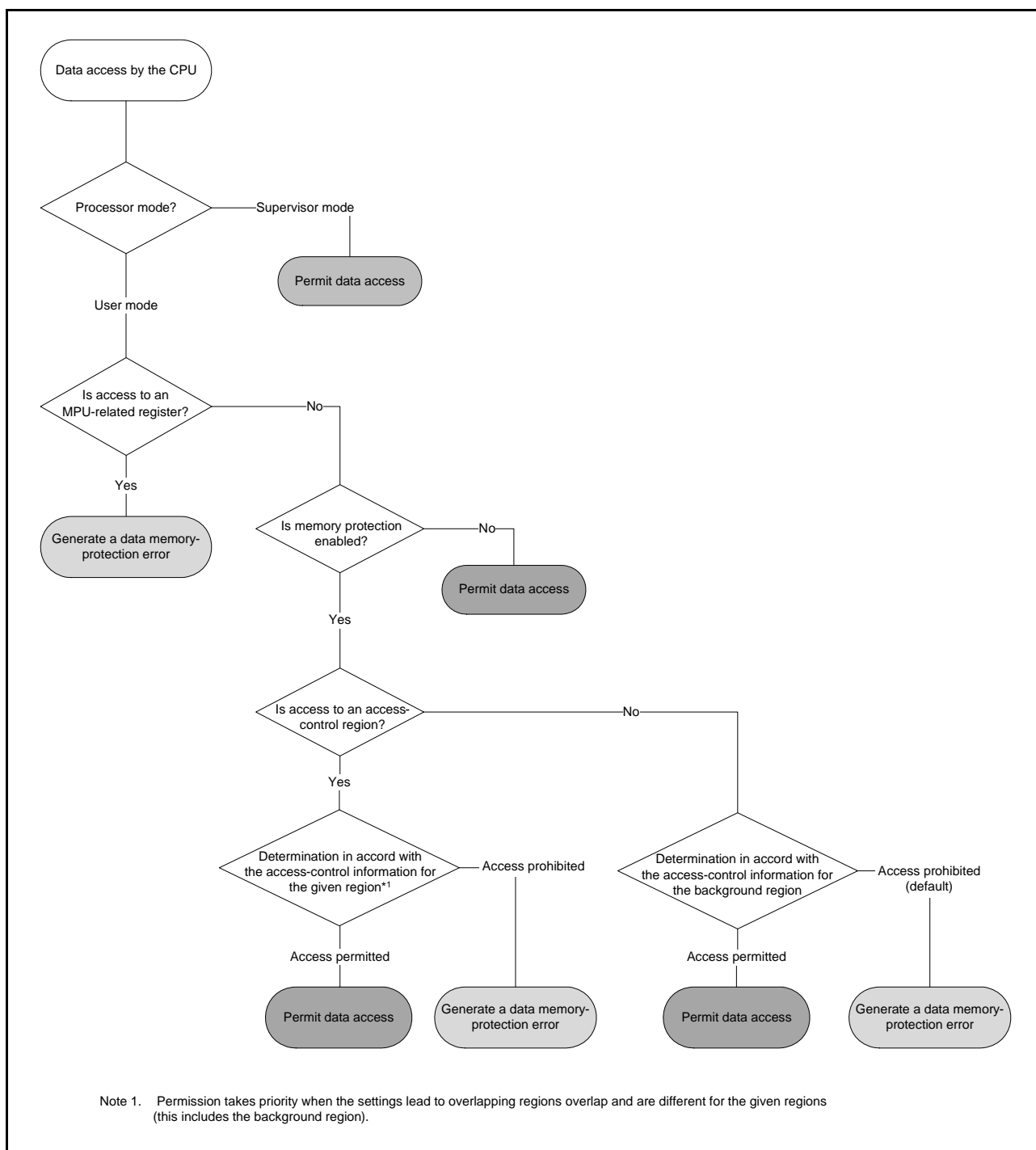


Figure 17.2 Flow of Determination for Data Access

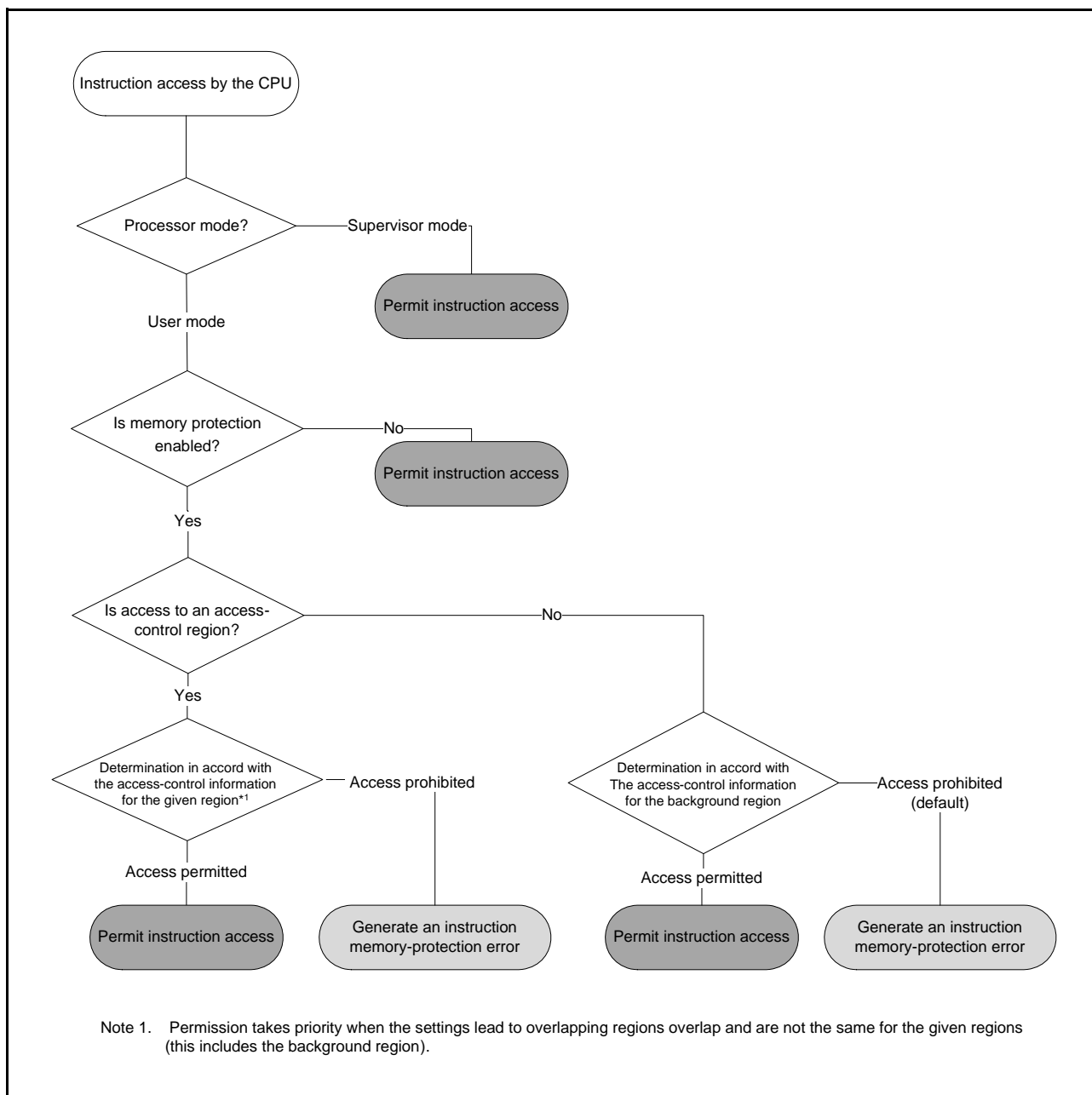


Figure 17.3 Flow of Determination for Instruction Access

## 17.4 Procedures for Using Memory Protection

### 17.4.1 Setting Access-Control Information

Access-control information for the various regions is set in supervisor mode.

Settings for up to eight access-control regions are made in the region-n start page number registers (RSPAGEn) and region-n end page number registers (REPAGEn), where n = 0 to 7.

Settings for the background access-control region are made in the background access-control register (MPBAC).

### 17.4.2 Enabling Memory Protection

Setting the memory-protection enable (MPEN) bit in the memory-protection enable (MPEN) register to 1 while operation is in supervisor mode enables memory protection.

### 17.4.3 Transition to User Mode

After updating the registers related to the memory-protection unit, be sure to read the registers for which writing was performed and check that the settings have been made as the final step before the transition to user mode.

Either of the methods below can be used for the transition from supervisor mode to user mode.

- Set the processor mode setting (PM) bit in the copy of the processor status word (PSW) saved in the stack area to 1 (the setting for user mode) and then execute an RTE instruction.
- Set the PM bit in the backup processor status word (BPSW) to 1 and then execute an RTFI instruction.

Note: Using an MVTC or POPC instruction to write to the PSW.PM bit is invalid. Use an RTE or RTFI instruction to update the value of the PSW.PM bit.

The memory-protection unit starts checking instruction-execution access and operand access by the CPU on the transition to user mode.

### 17.4.4 Processing in Response to Memory-Protection Errors

The CPU starts access-exception processing on detection of a violation of protection set up by the access-control information (i.e. a memory-protection error). For details on CPU operations in access-exception processing, refer to section 14., Exception Handling.

To determine whether an instruction memory-protection error or data memory-protection error has been generated, check the values of the instruction memory-protection error generated (IA) and data memory-protection error generated (DA) bits in the memory-protection error status (MPESTS) register from within the exception-processing routine.

After confirming the type of error, clear the memory-protection error status (MPESTS) register by writing 1 to the status clearing (MPE) bit in the memory-protection error status clearing (MPECLR) register.



### (1) When a data memory-protection error is generated

Access-exception processing by the CPU saves the address of the instruction that led to the memory-protection error on the stack. Furthermore, the address of the operand for which access led to a memory-protection error is stored in the data memory-protection error address register (MPDEA) and the region information for the region where the memory-protection error was generated is stored in the data-hit region register (MHITD).

- Violations of access control in access to valid regions 0 to 7

The data-hit region bits (MHITD.HITD[7:0]) with the same region number as the region where the error occurred is set to 1. In user mode, the logical “or” of the region access-control information for the location where the error occurred is set in the data-hit region access-control bits (MHITD.UHACD[2:0]).

- Violations of access control for the background region, besides access to valid regions 0 to 7

The data-hit region bits (MHITD.HITD[7:0]) are set to 0000 0000b. In user mode, the access-control information for the background region is set in the data-hit region access-control bits (MHITD.UHACD[2:0]).

Referring to this information can pinpoint the sources of errors.

### (2) When an instruction memory-protection error is generated

Access-exception processing by the CPU saves the address of the instruction that led to the memory-protection error on the stack. Furthermore, the region information for the region where the memory-protection error was generated is stored in the instruction-hit region register (MHITI).

- Violations of access control in access to valid regions 0 to 7

The instruction-hit region bit (MHITI.HITI[7:0]) with the same region number as the region where the error occurred is set to 1. In user mode, the logical “or” of the region access-control information for the location where the error occurred is set in the instruction-hit region access-control bits (MHITI.UHACI[2:0]).

- Violations of access control for the background region, besides access to valid regions 0 to 7

The instruction-hit region bits (MHITI.HITI[7:0]) are set to 0000 0000b. In user mode, the access-control information for the background region is set in the instruction-hit region access-control bits (MHITI.UHACI[2:0]).

Referring to this information can pinpoint the sources of errors.

## 18. DMA Controller (DMACA)

The RX630 Group incorporates a 4-channel direct memory access controller (DMAC).

The DMAC is a module to transfer data without the CPU. When a DMA transfer request is generated, the DMAC transfers data stored at the transfer source address to the transfer destination address.

### 18.1 Overview

Table 18.1 lists the specifications of the DMAC, and Figure 18.1 shows a block diagram of the DMAC.

**Table 18.1 Specifications of DMAC**

Item		Description			
Number of channels		4 (DMAC <sub>m</sub> (m = 0 to 3))			
Transfer space		512 Mbytes (0000 0000h to 0FFF FFFFh and F000 0000h to FFFF FFFFh excluding reserved areas)			
Maximum transfer volume		1 Mbyte (Maximum number of transfers in block transfer mode: 1,024 data × 1,024 blocks)			
DMA request source		<ul style="list-style-type: none"> <li>Activation source selectable for each channel</li> <li>Software trigger</li> <li>Interrupt requests from peripheral modules or trigger input to external interrupt input pins*1</li> </ul>			
Channel priority		Channel 0 > Channel 1 > Channel 2 > Channel 3 (Channel 0: Highest)			
Transfer data	Single data	Bit length: 8, 16, 32 bits			
	Block size	Number of data: 1 to 1,024			
Transfer mode	Normal transfer mode	<ul style="list-style-type: none"> <li>One data transfer by one DMA transfer request</li> <li>Free running mode (setting in which total number of data transfers is not specified) settable</li> </ul>			
	Repeat transfer mode	<ul style="list-style-type: none"> <li>One data transfer by one DMA transfer request</li> <li>Program returns to the transfer start address on completion of the repeat size of data transfer specified for the transfer source or destination.</li> <li>Maximum settable repeat size: 1,024</li> </ul>			
	Block transfer mode	<ul style="list-style-type: none"> <li>One block data transfer by one DMA transfer request</li> <li>Maximum settable block size: 1,024 data</li> </ul>			
Selective functions	Extended repeat area function	<ul style="list-style-type: none"> <li>Function in which data can be transferred by repeating the address values in the specified range with the upper bit values in the transfer address register fixed</li> <li>Area of 2 bytes to 128 Mbytes separately settable as extended repeat area for transfer source and destination</li> </ul>			
	Interrupt request	<table border="0"> <tr> <td>Transfer end interrupt</td> <td>Generated on completion of transferring data volume specified by the transfer counter.</td> </tr> <tr> <td>Transfer escape end interrupt</td> <td>Generated when the repeat size of data transfer is completed or the extended repeat area overflows.</td> </tr> </table>	Transfer end interrupt	Generated on completion of transferring data volume specified by the transfer counter.	Transfer escape end interrupt
Transfer end interrupt	Generated on completion of transferring data volume specified by the transfer counter.				
Transfer escape end interrupt	Generated when the repeat size of data transfer is completed or the extended repeat area overflows.				
Power consumption reduction function		Module-stop state can be set.			

Note 1. For details on DMAC activation sources, see Table 15.3, Interrupt Vector Table in section 15, Interrupt controller (ICUb).

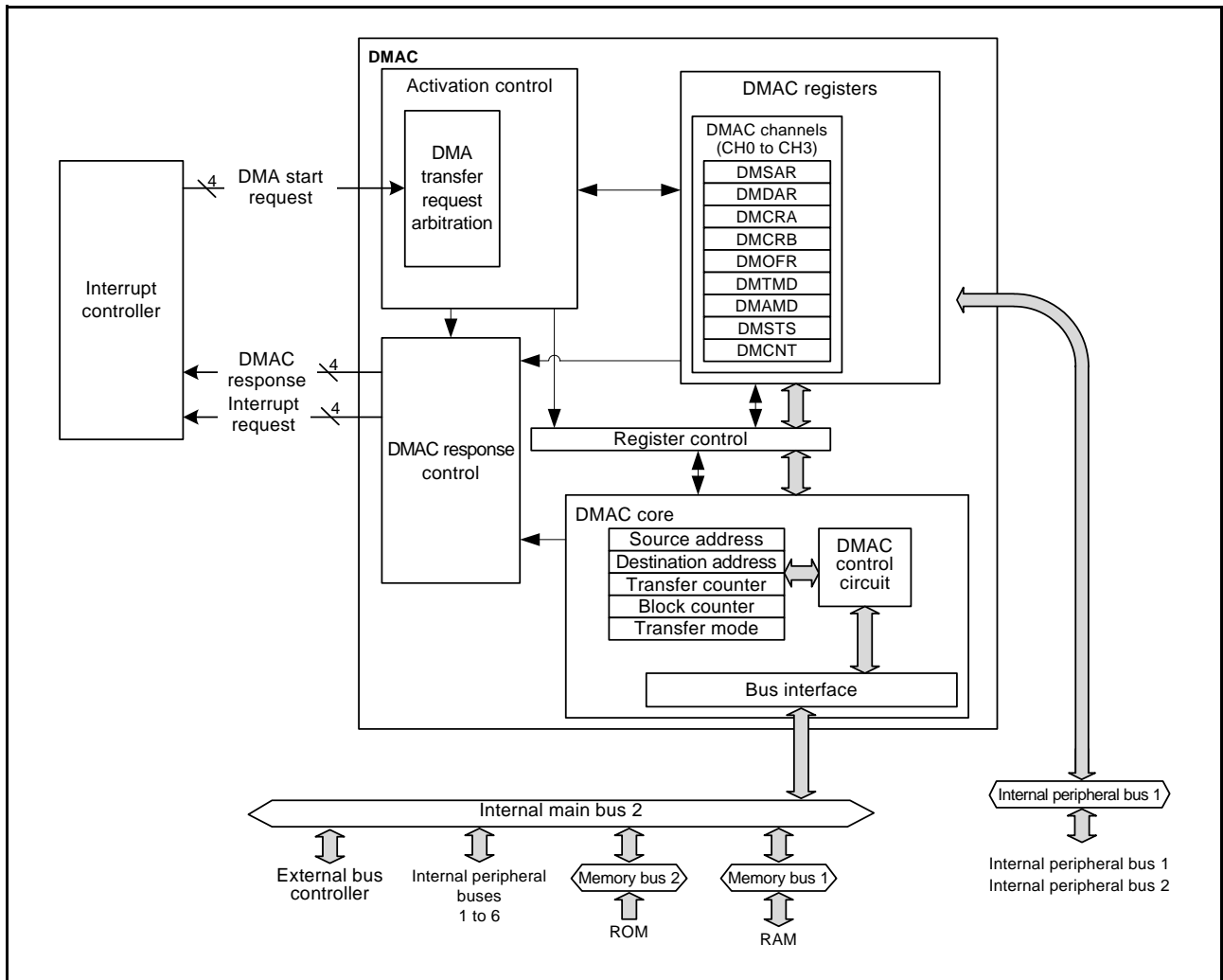
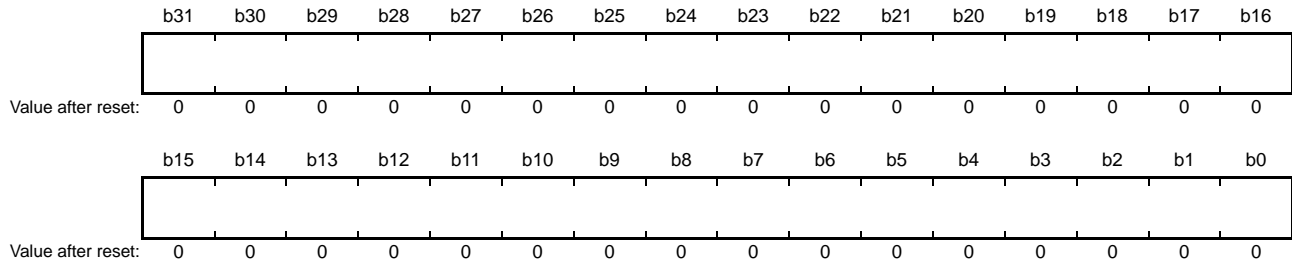


Figure 18.1 Block Diagram of DMAC

## 18.2 Register Descriptions

### 18.2.1 DMA Source Address Register (DMSAR)

Address(es): DMAC0.DMSAR 0008 2000h, DMAC1.DMSAR 0008 2040h  
 DMAC2.DMSAR 0008 2080h, DMAC3.DMSAR 0008 20C0h



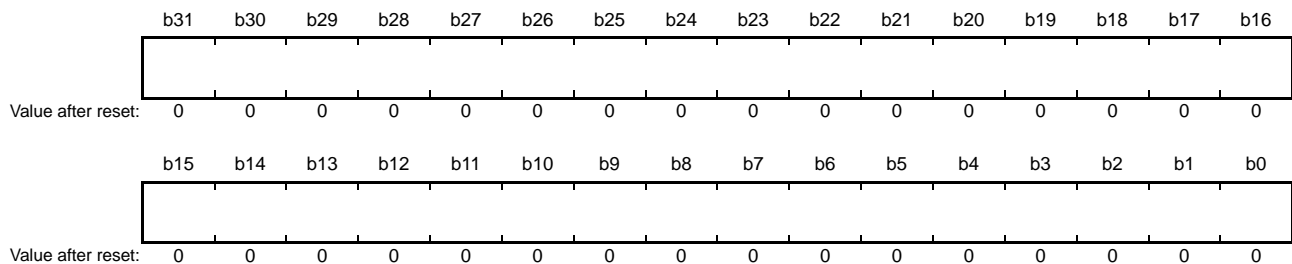
Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer source start address.	0000 0000h to 0FFF FFFFh (256 Mbytes) F000 0000h to FFFF FFFFh (256 Mbytes)	R/W

Set DMSAR while DMAC activation is disabled (the DMST bit in DMAST = 0) or DMA transfer is disabled (the DTE bit in DMCNT = 0).

Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading DMSAR returns the extended value.

### 18.2.2 DMA Destination Address Register (DMDAR)

Address(es): DMAC0.DMDAR 0008 2004h, DMAC1.DMDAR 0008 2044h  
 DMAC2.DMDAR 0008 2084h, DMAC3.DMDAR 0008 20C4h



Bit	Description	Setting Range	R/W
b31 to b0	Specifies the transfer destination start address.	0000 0000h to 0FFF FFFFh (256 Mbytes) F000 0000h to FFFF FFFFh (256 Mbytes)	R/W

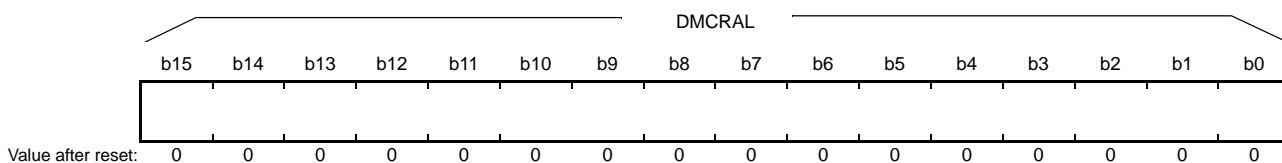
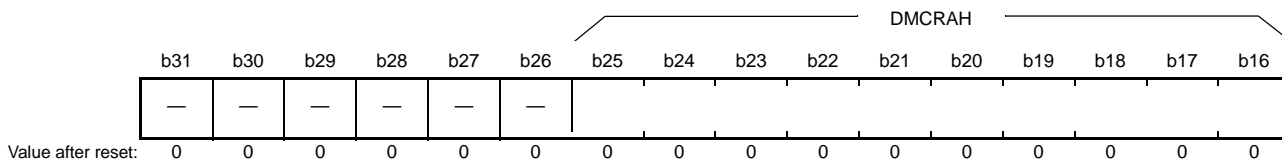
Set DMDAR while DMAC activation is disabled (the DMST bit in DMAST = 0) or DMA transfer is disabled (the DTE bit in DMCNT = 0).

Setting bits 31 to 29 is invalid; a value of bit 28 is extended to bits 31 to 29. Reading DMDAR returns the extended value.

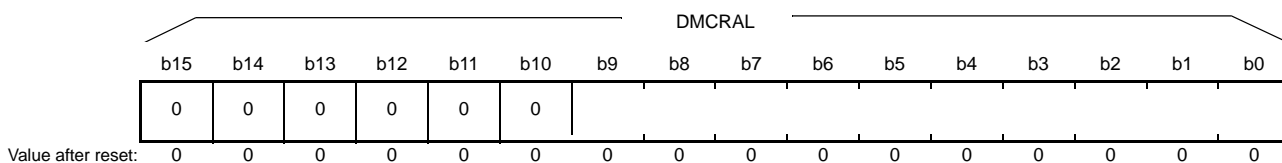
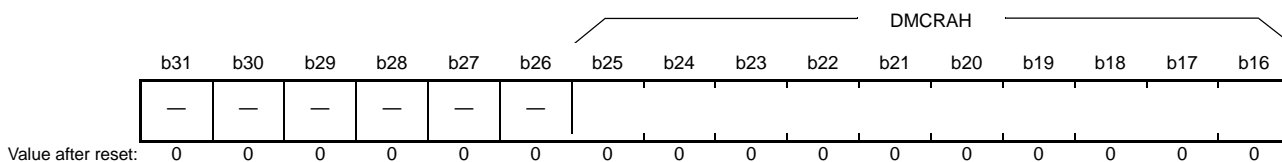
### 18.2.3 DMA Transfer Count Register (DMCRA)

Address(es): DMAC0.DMCRA 0008 2008h, DMAC1.DMCRA 0008 2048h  
 DMAC2.DMCRA 0008 2088h, DMAC3.DMCRA 0008 20C8h

- Normal transfer mode



- Repeat transfer mode, block transfer mode



Symbol	Bit Name	Description	R/W
DMCRAL	Lower bits of transfer count	Specifies the number of transfer operations	R/W
DMCRAH	Upper bits of transfer count		R/W

Note: • Set the same value for DMCRAH and DMCRAL in repeat transfer mode and block transfer mode.

#### (1) Normal Transfer Mode (MD[1:0] Bits in DMACm.DMTMD = 00b)

DMCRAL functions as a 16-bit transfer counter. The number of transfer operations is one when the setting is 0001h, and 65535 when it is FFFFh.

The value is decremented by one each time data is transferred.

When the setting is 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running mode).

DMCRAH is not used in normal transfer mode. Write 0000h to DMCRAH.

**(2) Repeat Transfer Mode (MD[1:0] Bits in DMACm.DMTMD = 01b)**

DMCRAH specifies the repeat size and DMCRAL functions as a 10-bit transfer counter.

The number of transfer operations is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In repeat transfer mode, a value in the range of 000h to 3FFh (1 to 1024) can be set for DMCRAH and DMCRAL.

Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.

The value in DMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in DMCRAH is loaded into DMCRAL.

**(3) Block Transfer Mode (MD[1:0] Bits in DMACm.DMTMD = 10b)**

DMCRAH specifies the block size and DMCRAL functions as a 10-bit block size counter.

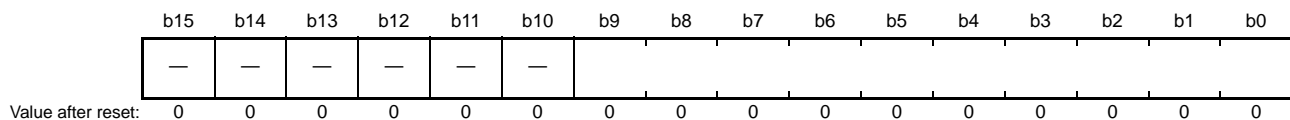
The block size is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h. In block transfer mode, a value in the range of 000h to 3FFh can be set for DMCRAH and DMCRAL.

Setting bits 15 to 10 in DMCRAL is invalid. Write 0 to these bits.

The value in DMCRAL is decremented by one each time data is transferred until it reaches 000h, at which the value in DMCRAH is loaded into DMCRAL.

**18.2.4 DMA Block Transfer Count Register (DMCRB)**

Address(es): DMAC0.DMCRB 0008 200Ch, DMAC1.DMCRB 0008 204Ch  
DMAC2.DMCRB 0008 208Ch, DMAC3.DMCRB 0008 20CCh



Bit	Description	Setting Range	R/W
b9 to b0	Specifies the number of block transfer operations or repeat transfer operations.	001h to 3FFh (1 to 1023) 000h (1024)	R/W
b15 to b10	Reserved	These bits are read as 0. The write value should be 0.	R/W

DMCRB specifies the number of block transfer operations and repeat transfer operations in block and repeat transfer mode, respectively.

The number of transfer operations is one when the setting is 001h, 1023 when it is 3FFh, and 1024 when it is 000h.

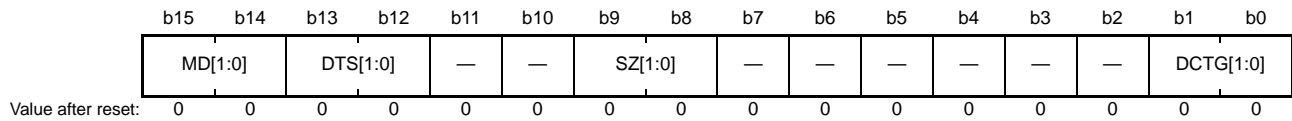
In repeat transfer mode, the value is decremented by one when the final data of one repeat size is transferred.

In block transfer mode, the value is decremented by one when the final data of one block size is transferred.

In normal transfer mode, DMCRB is not used. The setting is invalid.

## 18.2.5 DMA Transfer Mode Register (DMTMD)

Address(es): DMAC0.DMTMD 0008 2010h, DMAC1.DMTMD 0008 2050h  
DMAC2.DMTMD 0008 2090h, DMAC3.DMTMD 0008 20D0h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	DCTG[1:0]	DMA Request Source Select	b1 b0 0 0: Software 0 1: Interrupts*1 from peripheral modules or external interrupt input pins 1 0: Setting prohibited 1 1: Setting prohibited	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b9, b8	SZ[1:0]	Transfer Data Size Select	b9 b8 0 0: 8 bits 0 1: 16 bits 1 0: 32 bits 1 1: Setting prohibited	R/W
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13, b12	DTS[1:0]	Repeat Area Select	b13 b12 0 0: The destination is specified as the repeat area or block area. 0 1: The source is specified as the repeat area or block area. 1 0: The repeat area or block area is not specified. 1 1: Setting prohibited	R/W
b15, b14	MD[1:0]	Transfer Mode Select	b15 b14 0 0: Normal transfer 0 1: Repeat transfer 1 0: Block transfer 1 1: Setting prohibited	R/W

Note 1. DMAC activation source is selected using the DMRSRm registers of the ICU. For details on DMAC activation sources, see Table 15.3, Interrupt Vector Table in section 15, Interrupt controller (ICUb).

### DTS[1:0] Bits (Repeat Area Select)

DTS[1:0] select either the source or destination as the repeat area in repeat or block transfer mode. In normal transfer mode, setting these bits is invalid.

## 18.2.6 DMA Interrupt Setting Register (DMINT)

Address(es): DMAC0.DMINT 0008 2013h, DMAC1.DMINT 0008 2053h  
DMAC2.DMINT 0008 2093h, DMAC3.DMINT 0008 20D3h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	DTIE	ESIE	RPTIE	SARIE	DARIE

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	DARIE	Destination Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the destination address 1: Enables an interrupt request for an extended repeat area overflow on the destination address	R/W
b1	SARIE	Source Address Extended Repeat Area Overflow Interrupt Enable	0: Disables an interrupt request for an extended repeat area overflow on the source address 1: Enables an interrupt request for an extended repeat area overflow on the source address	R/W
b2	RPTIE	Repeat Size End Interrupt Enable	0: Disables the repeat size end interrupt request. 1: Enables the repeat size end interrupt request.	R/W
b3	ESIE	Transfer Escape End Interrupt Enable	0: Disables the transfer escape end interrupt request. 1: Enables the transfer escape end interrupt request.	R/W
b4	DTIE	Transfer End Interrupt Enable	0: Disables the transfer end interrupt request. 1: Enables the transfer end interrupt request.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

### DARIE Bit (Destination Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the destination address occurs while this bit is set to 1, the DTE bit in DMCNT is cleared to 0. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that an interrupt by an extended repeat area overflow on the destination address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DTE bit in DMACm.DMCNT of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the destination address, this bit is ignored.

### SARIE Bit (Source Address Extended Repeat Area Overflow Interrupt Enable)

When an extended repeat area overflow on the source address occurs while this bit is set to 1, the DTE bit in DMCNT is cleared to 0. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that an interrupt by an extended repeat area overflow on the source address is requested.

When block transfer mode is used with the extended repeat area function, an interrupt is requested after completion of a 1-block size transfer. When setting 1 in the DTE bit in DMACm.DMCNT of the channel for which a transfer has been stopped, the transfer is resumed from the state when the transfer is stopped.

When the extended repeat area is not specified for the source address, this bit is ignored.



**RPTIE Bit (Repeat Size End Interrupt Enable)**

When this bit is set to 1 in repeat transfer mode, the DTE bit in DMCNT is cleared to 0 after completion of a 1-repeat size data transfer. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DTS[1:0] bits in DMTMD are 10b (= repeat area or block area is not specified).

When this bit is set to 1 in block transfer mode, the DTE bit in DMCNT is cleared to 0 after completion of a 1-block data transfer in the same way as repeat transfer mode. At the same time, the ESIF flag in DMSTS is set to 1 to indicate that the repeat size end interrupt request has been generated. The repeat size end interrupt request can be generated even when the DTS[1:0] bits in DMTMD are 10b (= repeat area or block area is not specified).

**ESIE Bit (Transfer Escape End Interrupt Enable)**

This bit enables or disables the transfer escape end interrupt requests (repeat size end interrupt request and extended repeat area overflow interrupt request) that are generated during DMA transfer.

The transfer escape end interrupt is generated when the ESIF flag in DMSTS is set to 1 with this bit set to 1. The transfer escape end interrupt is cleared by clearing this bit or the ESIF flag in DMSTS to 0.

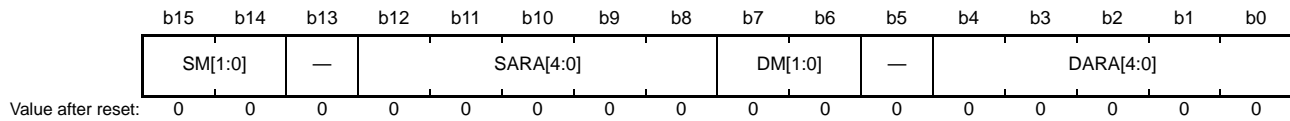
**DTIE Bit (Transfer End Interrupt Enable)**

This bit enables or disables the transfer end interrupt request to be generated on completion of a specified number of data transfers.

The transfer end interrupt is generated when the DTIF bit in DMSTS is set to 1 with this bit set to 1. The transfer end interrupt is cleared by clearing this bit or the DTIF bit in DMSTS to 0.

## 18.2.7 DMA Address Mode Register (DMAMD)

Address(es): DMAC0.DMAMD 0008 2014h, DMAC1.DMAMD 0008 2054h  
DMAC2.DMAMD 0008 2094h, DMAC3.DMAMD 0008 20D4h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	DARA[4:0]	Destination Address Extended Repeat Area	Specifies the extended repeat area on the destination address. For details on the settings, see Table 18.2.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7, b6	DM[1:0]	Destination Address Update Mode	b7 b6 0 0: Destination address is fixed. 0 1: Offset addition*1 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W
b12 to b8	SARA[4:0]	Source Address Extended Repeat Area	Specifies the extended repeat area on the source address. For details on the settings, see Table 18.2.	R/W
b13	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b15, b14	SM[1:0]	Source Address Update Mode	b15 b14 0 0: Destination address is fixed. 0 1: Offset addition*1 1 0: Destination address is incremented. 1 1: Destination address is decremented.	R/W

Note 1. Offset addition can be specified only for DMAC0.

### DARA[4:0] Bits (Destination Address Extended Repeat Area)

These bits specify the extended repeat area on the destination address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes.

When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer destination, do not specify the extended repeat area on the destination address. When repeat transfer or block transfer is selected, or when DMACm.DMTMD.DTS[1:0] = 00b (the transfer destination is specified as the repeat area or block area), write 00000b in the DARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the DARIE bit in DMINT set to 1. Table 18.2 lists the settings and the corresponding extended repeat areas.

### DM[1:0] Bits (Destination Address Update Mode)

These bits select the mode of updating the destination address.

When increment is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the destination address is incremented by 1, 2, and 4, respectively.

When decrement is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the destination address is decremented by 1, 2, and 4, respectively.

When offset addition is selected, the offset specified by the DMAC0.DMOFR register is added to the address.

Offset addition can be specified only for DMAC0.

**SARA[4:0] Bits (Source Address Extended Repeat Area)**

These bits specify the extended repeat area on the source address. The extended repeat area function is realized by updating the specified lower address bits with the remaining upper address bits fixed. The size of the extended repeat area can be any power of two between 2 bytes and 128 Mbytes.

When the lower address overflows the extended repeat area by address increment, the start address of the extended repeat area is set. Similarly, when the lower address underflows the extended repeat area by address decrement, the end address of the extended repeat area is set.

When the repeat area or block area is specified as a transfer source, do not specify the extended repeat area on the source address. When repeat transfer or block transfer is selected, or when DMACm.DMTMD.DTS[1:0] = 01b (the transfer source is specified as the repeat area or block area), write 00000b in the SARA[4:0] bits.

An interrupt can be requested when an overflow or underflow occurs in the extended repeat area with the SARIE bit in DMINT set to 1. Table 18.2 lists the settings and the corresponding extended repeat areas.

**SM Bit (Source Address Update Mode)**

These bits select the mode of updating the source address.

When increment is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the source address is incremented by 1, 2, and 4, respectively.

When decrement is selected and the SZ[1:0] bits in DMTMD are set to 00b, 01b, and 10b, the source address is decremented by 1, 2, and 4, respectively.

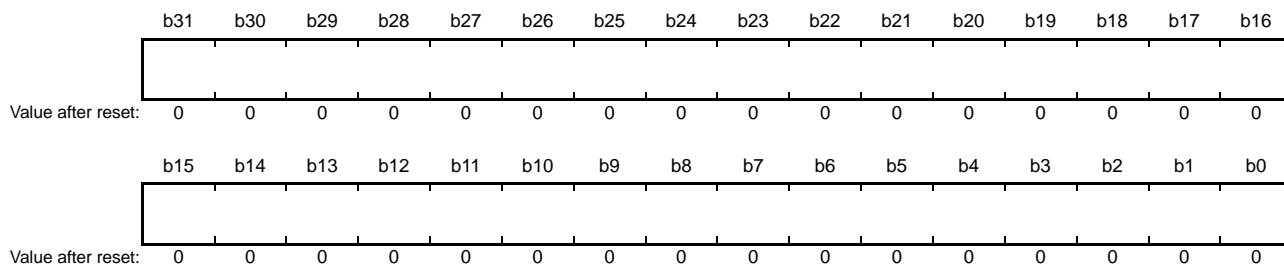
When offset addition is selected, the offset specified by the DMAC0.DMOFR register is added to the address. Offset addition can be specified only for DMAC0.

**Table 18.2 SARA[4:0] or DARA[4:0] Settings and Corresponding Repeat Areas**

SARA4 to SARA0 or DARA4 to DARA0	Extended Repeat Area
00000b	Not specified
00001b	2 bytes specified as extended repeat area by the lower 1 bit of the address
00010b	4 bytes specified as extended repeat area by the lower 2 bits of the address
00011b	8 bytes specified as extended repeat area by the lower 3 bits of the address
00100b	16 bytes specified as extended repeat area by the lower 4 bits of the address
00101b	32 bytes specified as extended repeat area by the lower 5 bits of the address
00110b	64 bytes specified as extended repeat area by the lower 6 bits of the address
00111b	128 bytes specified as extended repeat area by the lower 7 bits of the address
01000b	256 bytes specified as extended repeat area by the lower 8 bits of the address
01001b	512 bytes specified as extended repeat area by the lower 9 bits of the address
01010b	1 Kbyte specified as extended repeat area by the lower 10 bits of the address
01011b	2 Kbytes specified as extended repeat area by the lower 11 bits of the address
01100b	4 Kbytes specified as extended repeat area by the lower 12 bits of the address
01101b	8 Kbytes specified as extended repeat area by the lower 13 bits of the address
01110b	16 Kbytes specified as extended repeat area by the lower 14 bits of the address
01111b	32 Kbytes specified as extended repeat area by the lower 15 bits of the address
10000b	64 Kbytes specified as extended repeat area by the lower 16 bits of the address
10001b	128 Kbytes specified as extended repeat area by the lower 17 bits of the address
10010b	256 Kbytes specified as extended repeat area by the lower 18 bits of the address
10011b	512 Kbytes specified as extended repeat area by the lower 19 bits of the address
10100b	1 Mbyte specified as extended repeat area by the lower 20 bits of the address
10101b	2 Mbytes specified as extended repeat area by the lower 21 bits of the address
10110b	4 Mbytes specified as extended repeat area by the lower 22 bits of the address
10111b	8 Mbytes specified as extended repeat area by the lower 23 bits of the address
11000b	16 Mbytes specified as extended repeat area by the lower 24 bits of the address
11001b	32 Mbytes specified as extended repeat area by the lower 25 bits of the address
11010b	64 Mbytes specified as extended repeat area by the lower 26 bits of the address
11011b	128 Mbytes specified as extended repeat area by the lower 27 bits of the address
11100b to 11111b	Setting prohibited.

## 18.2.8 DMA Offset Register (DMOFR)

Address(es): DMAC0.DMOFR 0008 2018h

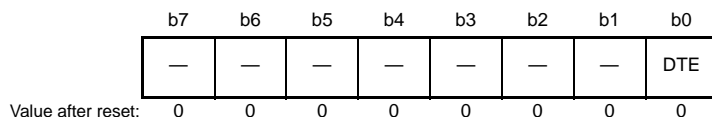


Bit	Description	Setting Range	R/W
b31 to b0	Specifies the offset when offset addition is selected as the address update mode for transfer source or destination.	0000 0000h to 00FF FFFFh (0 bytes to (16 M – 1) bytes) FF00 0000h to FFFF FFFFh (–16 Mbytes to –1 byte)	R/W

Write to this register while the DMAC operation is stopped or DMA transfer is disabled (not during data transfer). Setting bits 31 to 25 is invalid; a value of bit 24 is extended to bits 31 to 25. Reading DMOFR returns the extended value.

## 18.2.9 DMA Transfer Enable Register (DMCNT)

Address(es): DMAC0.DMCNT 0008 201Ch, DMAC1.DMCNT 0008 205Ch  
DMAC2.DMCNT 0008 209Ch, DMAC3.DMCNT 0008 20DCh



Bit	Symbol	Bit Name	Description	R/W
b0	DTE	DMA Transfer Enable	0: Disables DMA transfer. 1: Enables DMA transfer.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

### DTE Bit (DMA Transfer Enable)

When the DMST bit in DMAST is set to 1 (DMAC activation is enabled) and this bit is set to 1 (DMA transfer is enabled), DMA transfer can be started for the corresponding channel.

[Setting condition]

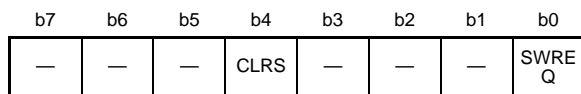
- When 1 is written to this bit.

[Clearing conditions]

- When 0 is written to this bit.
- When the specified total volume of data transfer is completed.
- When DMA transfer is stopped by the repeat size end interrupt.
- When DMA transfer is stopped by the extended repeat area overflow interrupt.

### 18.2.10 DMA Software Start Register (DMREQ)

Address(es): DMAC0.DMREQ 0008 201Dh, DMAC1.DMREQ 0008 205Dh  
DMAC2.DMREQ 0008 209Dh, DMAC3.DMREQ 0008 20DDh



Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	SWREQ	DMA Software Start	0: DMA transfer is not requested. 1: DMA transfer is requested.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	CLRS	DMA Software Start Bit Auto Clear Select	0: SWREQ bit is cleared after DMA transfer is started by software. 1: SWREQ bit is not cleared after DMA transfer is started by software.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

#### SWREQ Bit (DMA Software Start)

When 1 is written to this bit, a DMA transfer request is generated. After DMA transfer is started in response to the request, this bit is cleared to 0 if the CLRS bit is set to 0. This bit is not cleared to 0 while the CLRS bit is set to 1. In this case, a DMA transfer request can be issued again after completion of a transfer.

Note that, however, setting this bit is valid and DMA transfer by software is enabled only when the DCTG[1:0] bits in DMTMD are set to 00b (DMA activation source is software).

Setting this bit is invalid when the DCTG[1:0] bits in DMTMD are set to a value other than 00b.

To start DMA transfer by software with the CLRS bit being 0, ensure that the SWREQ bit is 0, and then write 1 to the SWREQ bit.

[Setting condition]

- When 1 is written to this bit.

[Clearing conditions]

- When a DMA transfer request by software is accepted and DMA transfer is started while the CLRS bit is set to 0 (the SWREQ bit is cleared after DMA transfer is started by software).
- When 0 is written to this bit.

#### CLRS Bit (DMA Software Start Bit Auto Clear Select)

This bit specifies whether to clear the SWREQ bit to 0 after DMA transfer is started in response to the DMA transfer request generated by setting the SWREQ bit to 1. With this bit set to 0, the SWREQ bit is cleared to 0 after DMA transfer is started. With this bit set to 1, the SWREQ bit is not cleared to 0. In this case, a DMA transfer request can be issued again after completion of a transfer.

### 18.2.11 DMA Status Register (DMSTS)

Address(es): DMAC0.DMSTS 0008 201Eh, DMAC1.DMSTS 0008 205Eh  
DMAC2.DMSTS 0008 209Eh, DMAC3.DMSTS 0008 20DEh

b7	b6	b5	b4	b3	b2	b1	b0
ACT	—	—	DTIF	—	—	—	ESIF

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	ESIF	Transfer Escape End Interrupt Flag	0: A transfer escape end interrupt has not been generated. 1: A transfer escape end interrupt has been generated.	R/W*1
b3 to b1	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b4	DTIF	Transfer End Interrupt Flag	0: A transfer end interrupt has not been generated. 1: A transfer end interrupt has been generated.	R/W*1
b6, b5	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R
b7	ACT	DMA Active Flag	0: DMAC operation is suspended. 1: DMAC is operating.	R

Note 1. Only 0 can be written to clear the flag.

#### ESIF Flag (Transfer Escape End Interrupt Flag)

This flag indicates that the transfer escape end interrupt has been generated.

[Setting conditions]

- When 1-repeat size data transfer is completed in repeat transfer mode with the RPTIE bit in DMINT set to 1.
- When 1-block data transfer is completed in block transfer mode with the RPTIE bit in DMINT set to 1.
- When an extended repeat area overflow on the source address occurs while the SARIE bit in DMINT is set to 1 and the SARA[4:0] bits in DMAMD are set to a value other than 00000b (extended repeat area is specified on the transfer source address)
- When an extended repeat area overflow on the destination address occurs while the DARIE bit in DMINT is set to 1 and the DARA[4:0] bits in DMAMD are set to a value other than 00000b (extended repeat area is specified on the transfer destination address)

[Clearing conditions]

- When 0 is written to this bit.
- When 1 is written to the DTE bit in DMCNT.

**DTIF Flag (Transfer End Interrupt Flag)**

This flag indicates that the transfer end interrupt has been generated.

[Setting conditions]

- When the specified number of unit-transfers are completed in normal transfer mode (the value of DMCRAL becoming 0 on completion of transfer)
- When the specified number of repeat transfer operations are completed in repeat transfer mode (the value of DMCRB becoming 0 on completion of transfer))
- When the specified number of blocks have been transferred in block transfer mode (the value of DMCRB becoming 0 on completion of transfer)

[Clearing conditions]

- When 0 is written to this bit
- When 1 is written to the DTE bit in DMCNT

**ACT Flag (DMA Active Flag)**

This flag indicates whether the DMAC is in the idle or active state.

[Setting condition]

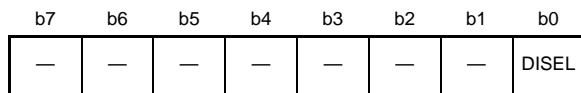
- When the DMAC starts data transfer operation

[Clearing condition]

- When data transfer in response to one transfer request is completed

**18.2.12 DMA Activation Source Flag Control Register (DMCSL)**

Address(es): DMAC0.DMCSL 0008 201Fh, DMAC1.DMCSL 0008 205Fh  
 DMAC2.DMCSL 0008 209Fh, DMAC3.DMCSL 0008 20DFh



Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	DISEL	Interrupt Select	0: At the beginning of transfer, clear the interrupt flag of the activation source to 0. 1: At the end of transfer, the interrupt flag of the activation source issues an interrupt to the CPU.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

**DISEL Bit (Interrupt Select)**

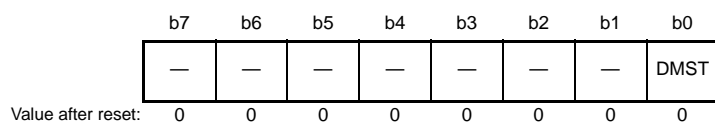
This bit selects whether the interrupt flag of the activation source of the DMAC is cleared to 0 or issues an interrupt to the CPU, at the beginning of transfer.

When DMTMD.DCTG[1:0] = 00b (activation by software), the setting of the DISEL bit does not affect the operation.



### 18.2.13 DMACA Module Activation Register (DMAST)

Address(es): 0008 2200h



Bit	Symbol	Bit Name	Description	R/W
b0	DMST	DMAC Operation Enable	0: DMAC activation is disabled. 1: DMAC activation is enabled.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

#### DMST Bit (DMAC Operation Enable)

When this bit is set to 1, DMAC activation is enabled for all channels.

When 1 is written to the DMACm.DMCNT.DTE bit (DMA transfer is enabled) of multiple channels and then this bit is set to 1 (DMAC activation is enabled), the corresponding multiple channels can be placed in the transfer request acceptable state at the same time.

When the DMST bit is cleared to 0 during DMA transfer, DMA transfer is suspended after completion of the current data transfer corresponding to a single transfer request. DMA transfer is resumed by setting the DMST bit to 1 again.

[Setting condition]

- When 1 is written to this bit

[Clearing condition]

- When 0 is written to this bit

## 18.3 Operation

### 18.3.1 Transfer Mode

#### (1) Normal Transfer Mode

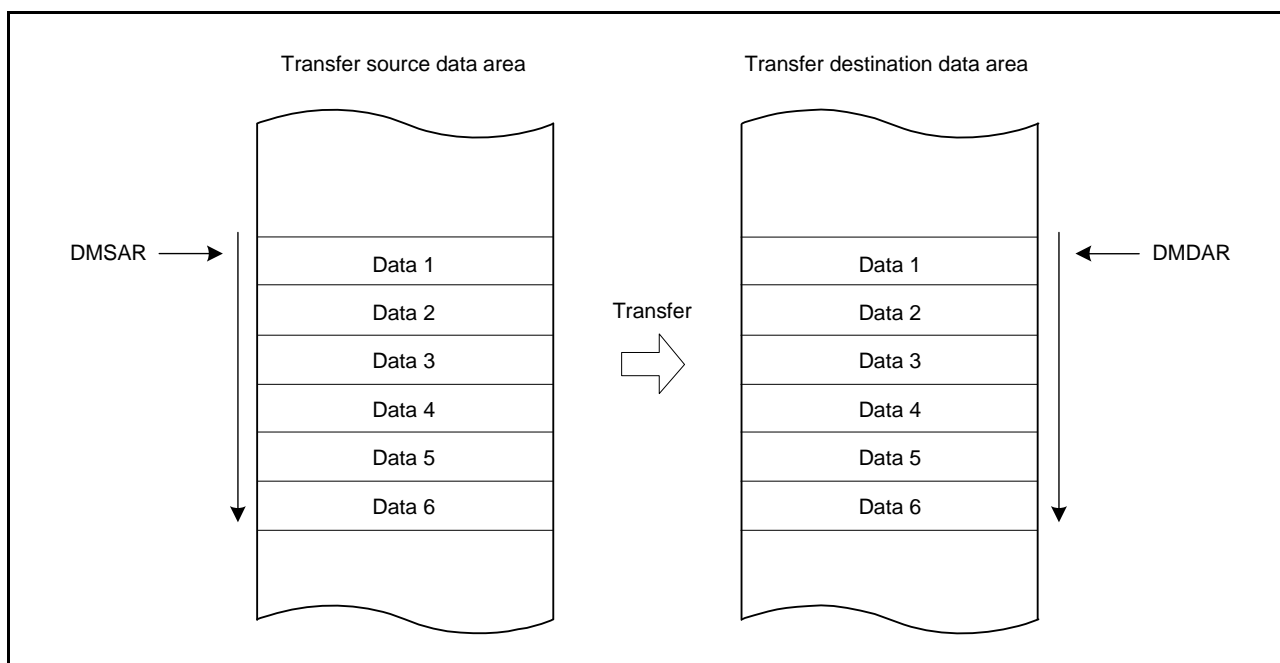
In normal transfer mode, one data is transferred by one transfer request. A maximum of 65535 can be set as the number of transfer operations using the DMCRAL of DMACm. When these bits are set to 0000h, no specific number of transfer operations is set; data transfer is performed with the transfer counter stopped (free running mode). Setting DMCRB of DMACm is invalid in normal transfer mode. Except in free running mode, a transfer end interrupt request can be generated after completion of the specified number of transfer operations.

Table 18.3 summarizes the register update operation in normal transfer mode, and Figure 18.2 shows the operation in normal transfer mode.

**Table 18.3 Register Update Operation in Normal Transfer Mode**

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request
DMACm.DMSAR	Transfer source address	Increment/decrement/fixd/offset addition*1
DMACm.DMDAR	Transfer destination address	Increment/decrement/fixd/offset addition*1
DMACm.DMCRAL	Transfer count	Decrementd by one/not updated (in free running mode)
DMACm.DMCRAH	—	Not updated (Not used in normal transfer mode)
DMACm.DMCRB	—	Not updated (Not used in normal transfer mode)

Note 1. Offset addition can be specified only for DMAC0.



**Figure 18.2 Operation in Normal Transfer Mode**

## (2) Repeat Transfer Mode

In repeat transfer mode, one data is transferred by one transfer request.

A maximum of 1K data can be set as a total repeat transfer size using DMCRA of the DMACm.

A maximum of 1K can be set as the number of repeat transfer operations using DMCRB of the DMACm; therefore, a maximum of 1M data (1K data × 1K count of repeat transfer operations) can be set as a total data transfer size.

Either the transfer source or transfer destination can be specified as a repeat area. When transfer of the repeat size data is completed, the address of the specified repeat area (DMSAR or DMDAR of the DMACm) returns to the transfer start address. When data of the specified repeat size has all been transferred in repeat transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DTE bit in DMCNT of DMACm in the repeat size end interrupt handling.

A transfer end interrupt request can be generated after completion of the specified number of repeat transfer operations. Table 18.4 summarizes the register update operation in repeat transfer mode, and Figure 18.3 shows the operation in repeat transfer mode.

**Table 18.4 Register Update Operation in Repeat Transfer Mode**

Register	Function	Update Operation after Completion of a Transfer by One Transfer Request	
		When DMACm.DMCRAL is not 1	When DMACm.DMCRAL is 1 (Transfer of the Last Data in Repeat Size)
DMACm.DMSAR	Transfer source address	Increment/decrement/fixe d/offset addition*1	<ul style="list-style-type: none"> <li>• DMACm.DMTMD.DTS[1:0] = 00b Increment/decrement/fixe d/offset addition*1</li> <li>• DMACm.DMTMD.DTS[1:0] = 01b Initial value of DMACm.DMSAR</li> <li>• DMACm.DMTMD.DTS[1:0] = 10b Increment/decrement/fixe d/offset addition*1</li> </ul>
DMACm.DMDAR	Transfer destination address	Increment/decrement/fixe d/offset addition*1	<ul style="list-style-type: none"> <li>• DMACm.DMTMD.DTS[1:0] = 00b Initial value of DMACm.DMDAR</li> <li>• DMACm.DMTMD.DTS[1:0] = 01b Increment/decrement/fixe d/offset addition*1</li> <li>• DMACm.DMTMD.DTS[1:0] = 10b Increment/decrement/fixe d/offset addition*1</li> </ul>
DMACm.DMCRAH	Repeat size	Not updated	Not updated
DMACm.DMCRAL	Transfer count	Decremente d by one	DMACm.DMCRAH
DMACm.DMCRB	Count of repeat transfer operations	Not updated	Decremente d by one

Note 1. Offset addition can be specified only for DMAC0.

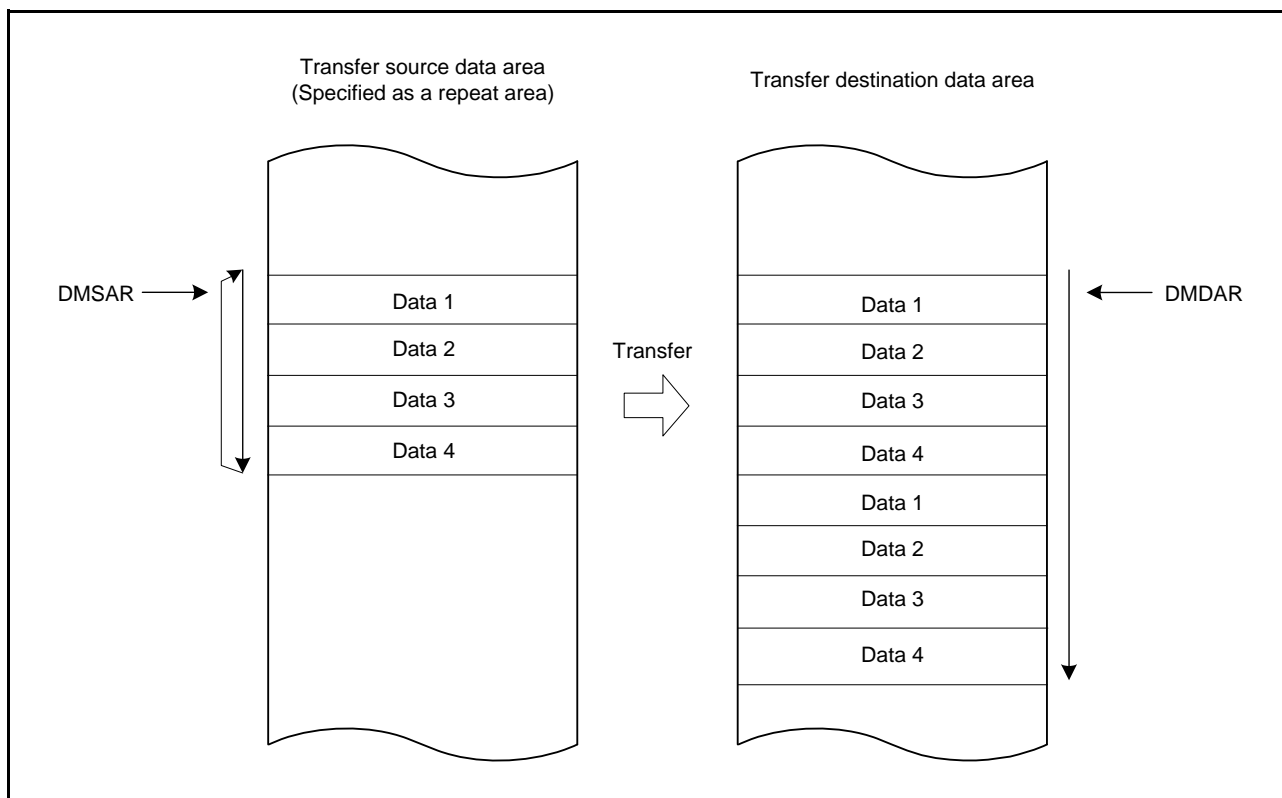


Figure 18.3 Operation in Repeat Transfer Mode

### (3) Block Transfer Mode

In block transfer mode, a single block data is transferred by one transfer request.

A maximum of 1K data can be set as a total block transfer size using DMCRA of the DMACm.

A maximum of 1M can be set as the number of block transfer operations using DMCRB of the DMACm; therefore, a maximum of 1M data (1K data × 1K count of block transfer operations) can be set as a total data transfer size.

Either the transfer source or transfer destination can be specified as a block area. When transfer of a single block data is completed, the address of the specified block area (DMSAR or DMDAR of the DMACm) returns to the transfer start address. When a single block data has all been transferred in block transfer mode, DMA transfer can be stopped and the repeat size end interrupt can be requested. DMA transfer can be resumed by writing 1 to the DTE bit in DMCNT of DMACm in the repeat size end interrupt handling.

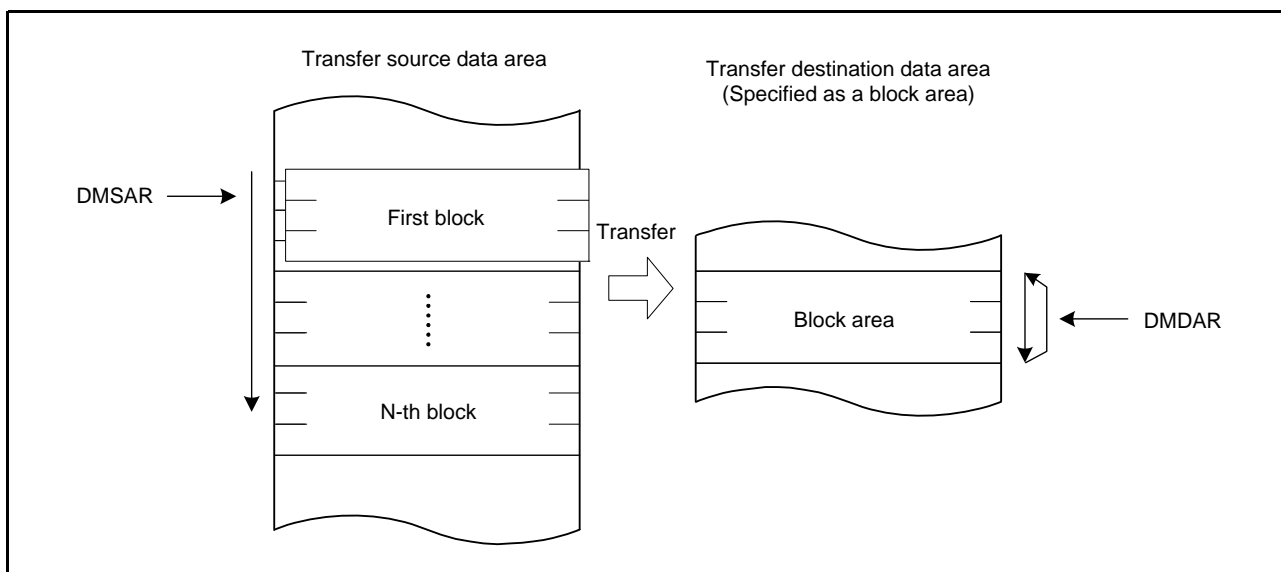
Transfer end interrupt request can be generated after completion of the specified number of block transfer operations.

Table 18.5 summarizes the register update operation in block transfer mode, and Figure 18.4 shows the operation in block transfer mode.

**Table 18.5 Register Update Operation in Block Transfer Mode**

Register	Function	Update Operation after Completion of Single-Block Transfer by One Transfer Request
DMACm.DMSAR	Transfer source address	<ul style="list-style-type: none"> <li>DMACm.DMTMD.DTS[1:0] = 00b Increment/decrement/fixed/offset addition*1</li> <li>DMACm.DMTMD.DTS[1:0] = 01b Initial value of DMACm.DMSAR</li> <li>DMACm.DMTMD.DTS[1:0] = 10b Increment/decrement/fixed/offset addition*1</li> </ul>
DMACm.DMDAR	Transfer destination address	<ul style="list-style-type: none"> <li>DMACm.DMTMD.DTS[1:0] = 00 b Initial value of DMACm.DMDAR</li> <li>DMACm.DMTMD.DTS[1:0] = 01 b Increment/decrement/fixed/offset addition*1</li> <li>DMACm.DMTMD.DTS[1:0] = 10 b Increment/decrement/fixed/offset addition*1</li> </ul>
DMACm.DMCRAH	Block size	Not updated
DMACm.DMCRAL	Transfer count	DMACm.DMCRAH
DMACm.DMCRB	Count of block transfer operations	Decrement by one

Note 1. Offset addition can be specified only for DMAC0.



**Figure 18.4 Operation in Block Transfer Mode**

### 18.3.2 Extended Repeat Area Function

The DMAC supports a function to specify the extended repeat areas on the transfer source and destination addresses. With the extended repeat areas set, the address registers repeatedly indicate the addresses of the specified extended repeat areas.

The extended repeat areas can be specified separately to the transfer source address register (DMSAR) and transfer destination address register (DMDAR) of DMACm.

The extended repeat area on the source address is specified by the SARA[4:0] bits in DMAMD of DMACm. The extended repeat area on the destination address is specified by the DARA[4:0] bits in DMAMD of DMACm. The size can be specified separately for the source and destination sides.

However, the area (of transfer source or transfer destination) which is specified as the repeat area or block area should not be specified as the extended repeat area.

When the address register value reaches the end address of the extended repeat area and the extended repeat area overflows, DMA transfer is stopped and an interrupt by an extended repeat area overflow can be requested. When an overflow occurs in the extended repeat area on the transfer source while the SARIE bit in DMINT of DMACm is set to 1, the ESIF flag in DMSTS of DMACm is set to 1 and the DTE bit in DMCNT of DMACm is cleared to 0 to stop DMA transfer. At this time, if the ESIE bit in DMINT of DMACm is set to 1, an interrupt by an extended repeat area overflow is requested. When the DARIE bit in DMINT of DMACm is set to 1, the destination address register becomes a target to apply the function. DMA transfer can be resumed by writing 1 to the DTE bit in DMCNT of DMACm in the interrupt handling.

Figure 18.5 shows an example of the extended repeat area operation.

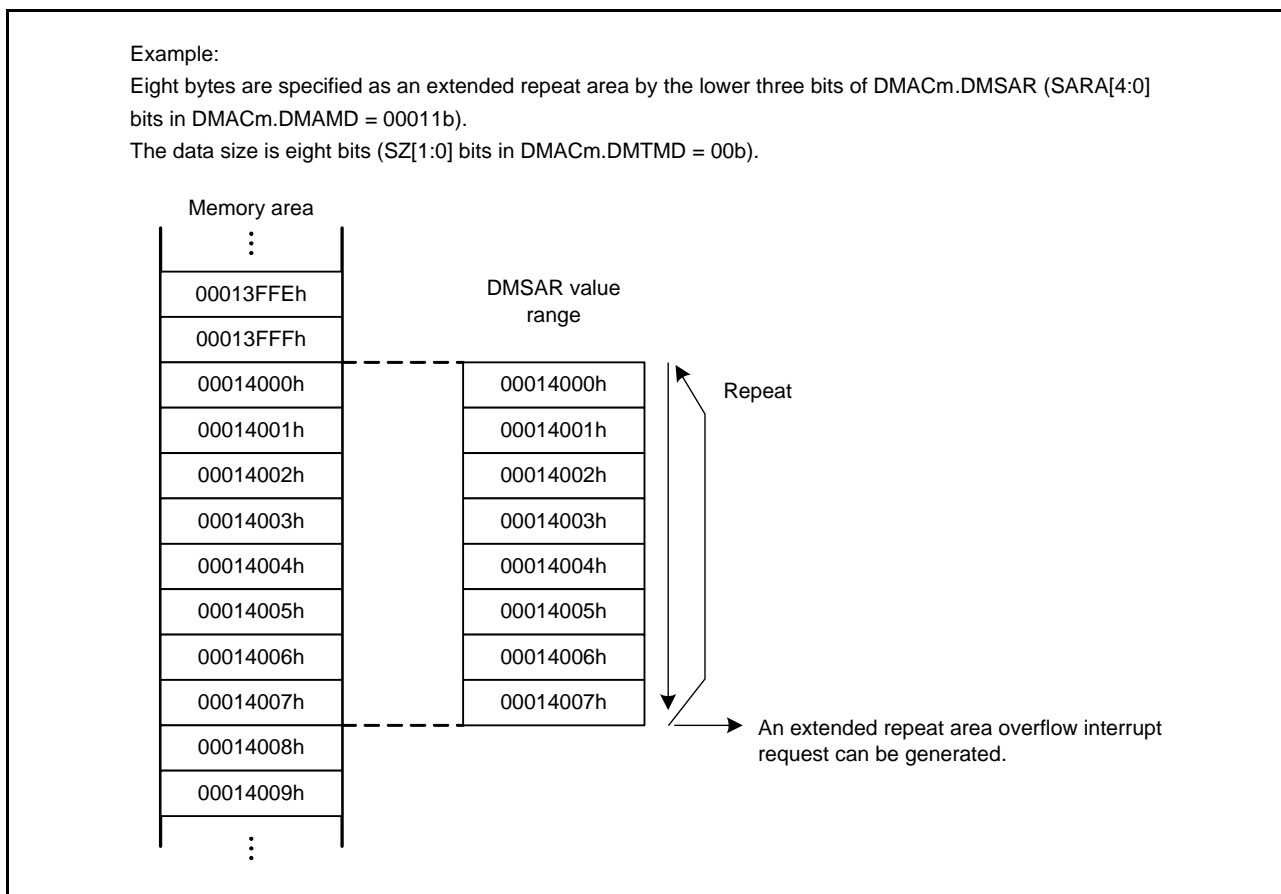


Figure 18.5 Example of Extended Repeat Area Operation

When an interrupt by an extended repeat area overflow is used in block transfer mode, the following should be taken into consideration.

When a transfer is stopped by an interrupt by an extended repeat area overflow, the address register must be set so that the block size is a power of 2 or the block size boundary is aligned with the extended repeat area boundary. When an overflow on the extended repeat area occurs during a transfer of one block, the interrupt by the overflow is suspended until transfer of the block is completed, and the transfer overruns.

Figure 18.6 shows an example when the extended repeat area function is used in block transfer mode.

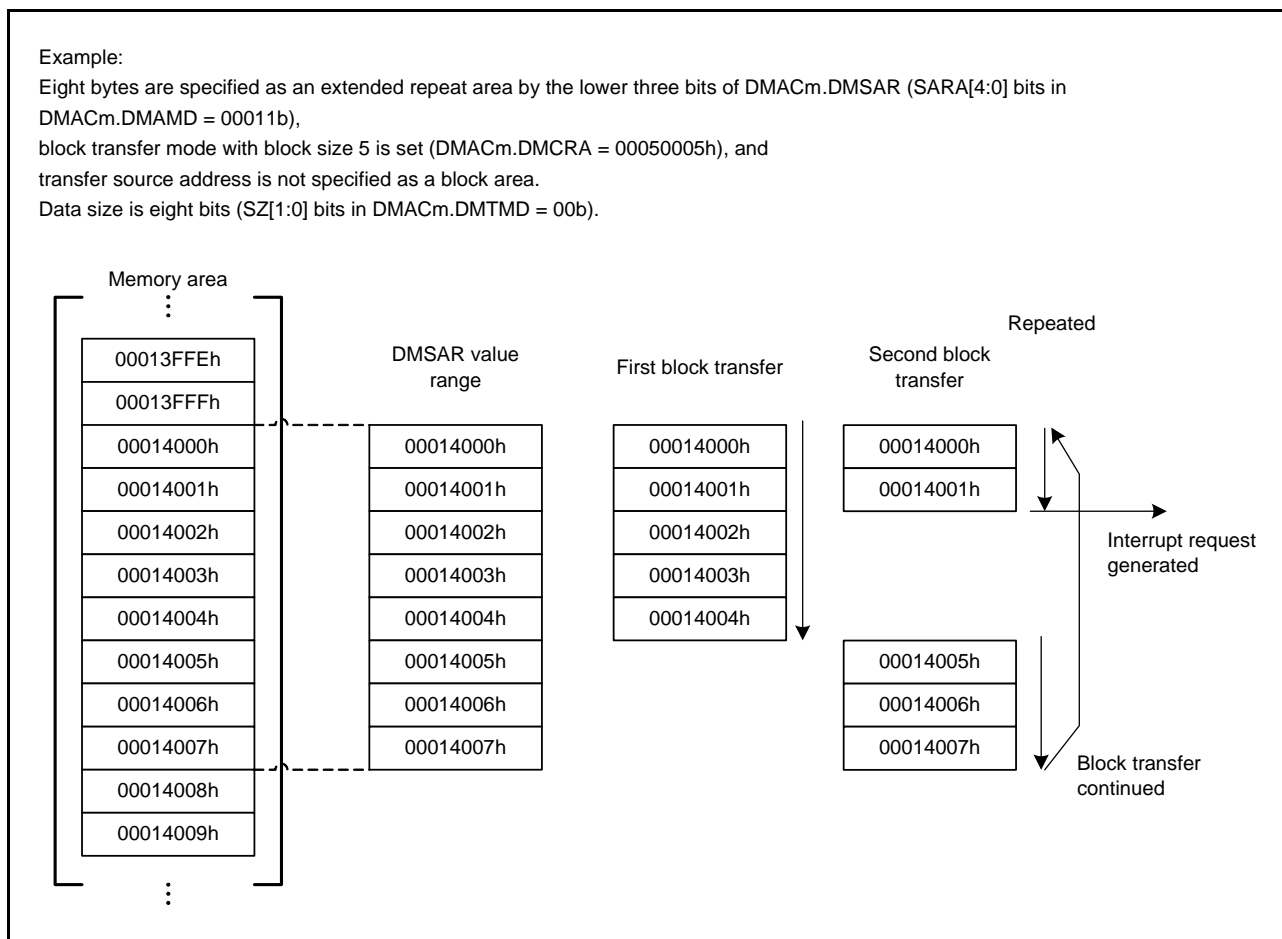


Figure 18.6 Example of Extended Repeat Area Function in Block Transfer Mode

### 18.3.3 Address Update Function using Offset

The source and destination addresses can be updated by fixing, increment, decrement, or offset addition. When the offset addition is selected, the offset specified by the DMA offset register (DMOFR of DMAC0) is added to the address every time the DMAC performs one data transfer. This function realizes a data transfer where addresses are allocated to separated areas.

Offset subtraction can also be realized by setting a negative value in DMOFR of DMAC0. In this case, the negative value must be 2's complement.

Address update function using offset can be specified only for the DMAC0 channel.

Table 18.6 shows the address update method in each address update mode.

**Table 18.6 Address Update Method in Each Address Update Mode**

Address Update Mode	Settings of DMACm.DMAMD.SM[1:0] and DMACm.DMAMD.DM[1:0] for Address Update Modes	Address Update Method (for Different SZ[1:0] Settings in DMTMD of DMACm)		
		SZ[1:0] = 00b	SZ[1:0] = 01b	SZ[1:0] = 10b
Address fixed	00b	Fixed		
Offset addition	01b	+DMACm.DMOFR*1		
Increment	10b	+1	+2	+4
Decrement	11b	-1	-2	-4

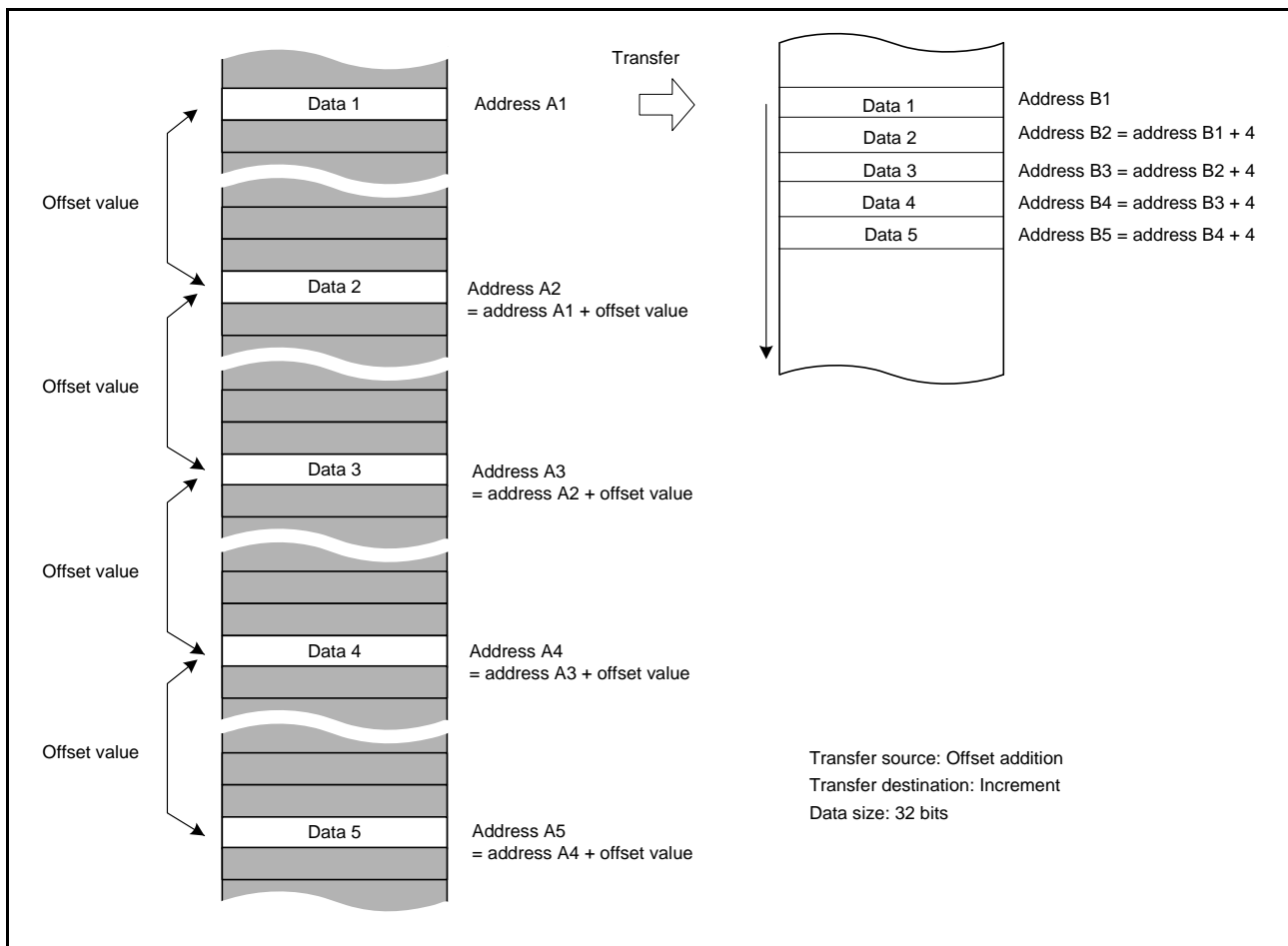
Note 1. When setting a negative value in the DMA offset register, the value must be 2's complement. The 2's complement is obtained by the following formula.

2's complement of a negative offset value =  $\sim(\text{offset}) + 1$  ( $\sim$ : bit inversion)



### (1) Basic Transfer Using Offset Addition

Figure 18.7 shows an example of address updating using offset addition.



**Figure 18.7 Example of Address Updating by Offset Addition**

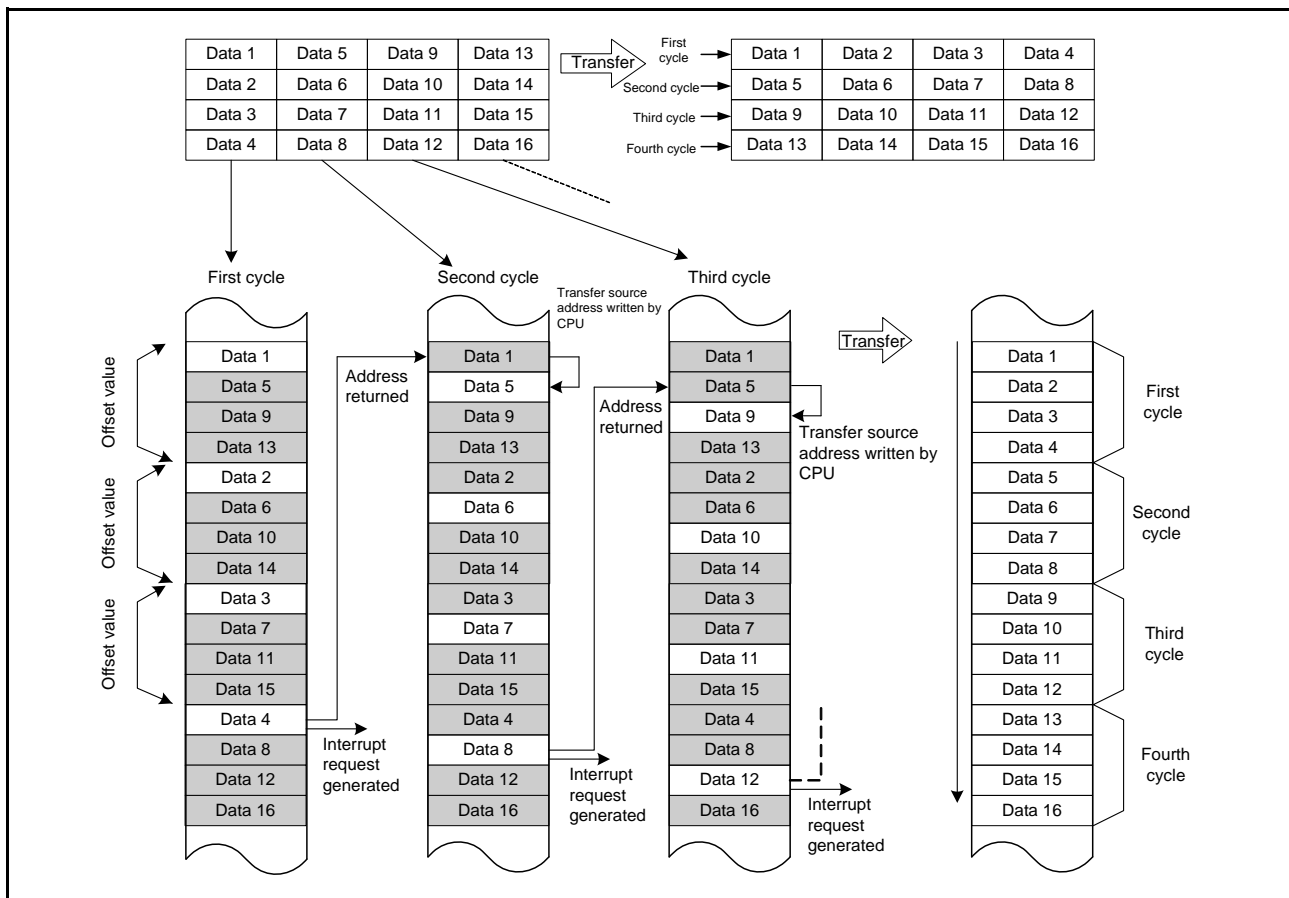
In Figure 18.7, the transfer data is 32 bits long, and offset addition and increment are set as the transfer source address update mode and transfer destination address update mode, respectively. The second and subsequent data is each read from the transfer source address obtained by adding the offset value to the previous address. The data read from the addresses at the specified intervals is written to the continuous locations on the destination.

### (2) Example of XY Conversion Using Offset Addition

Figure 18.8 shows the XY conversion using offset addition in repeat transfer mode.

Settings are as follows:

- DMAC0.DMAMD: Transfer source address update mode: Offset addition
- DMAC0.DMAMD: Transfer destination address update mode: Destination address is incremented.
- DMAC0.DMTMD: Transfer data size select: 32 bits
- DMAC0.DMTMD: Transfer mode select: Repeat transfer
- DMAC0.DMTMD: Repeat area select: The source is specified as the repeat area.
- DMAC0.DMOFR: Offset address: 10h
- DMAC0.DMCRA: Repeat size: 4h
- DMAC0.DMINT: The repeat size end interrupt is enabled.



**Figure 18.8 XY Conversion Operation Using Offset Addition in Repeat Transfer Mode**

When a transfer starts, the offset value is added to the transfer source address every time data is transferred. The transfer data is written to the destination continuous addresses. When data 4 is transferred, which means that the repeat size of transfers is completed, the transfer source address returns to the transfer start address (address of data 1 on the transfer source) and a repeat size end interrupt is requested. While this interrupt stops the transfer temporarily, perform the following.

- DMAC0.DMSAR: Rewrite the DMA transfer source address to the address of data 5 (with the above example, the data 1 address + 4).
- DMAC0.DMCNT: Set the DTE bit to 1.

The DMA transfer is resumed from the state when the DMA transfer is stopped. After that, the operations described above are repeated until the transfer source data is transposed to the destination area (XY conversion).

Figure 18.9 shows a flowchart of the XY conversion.

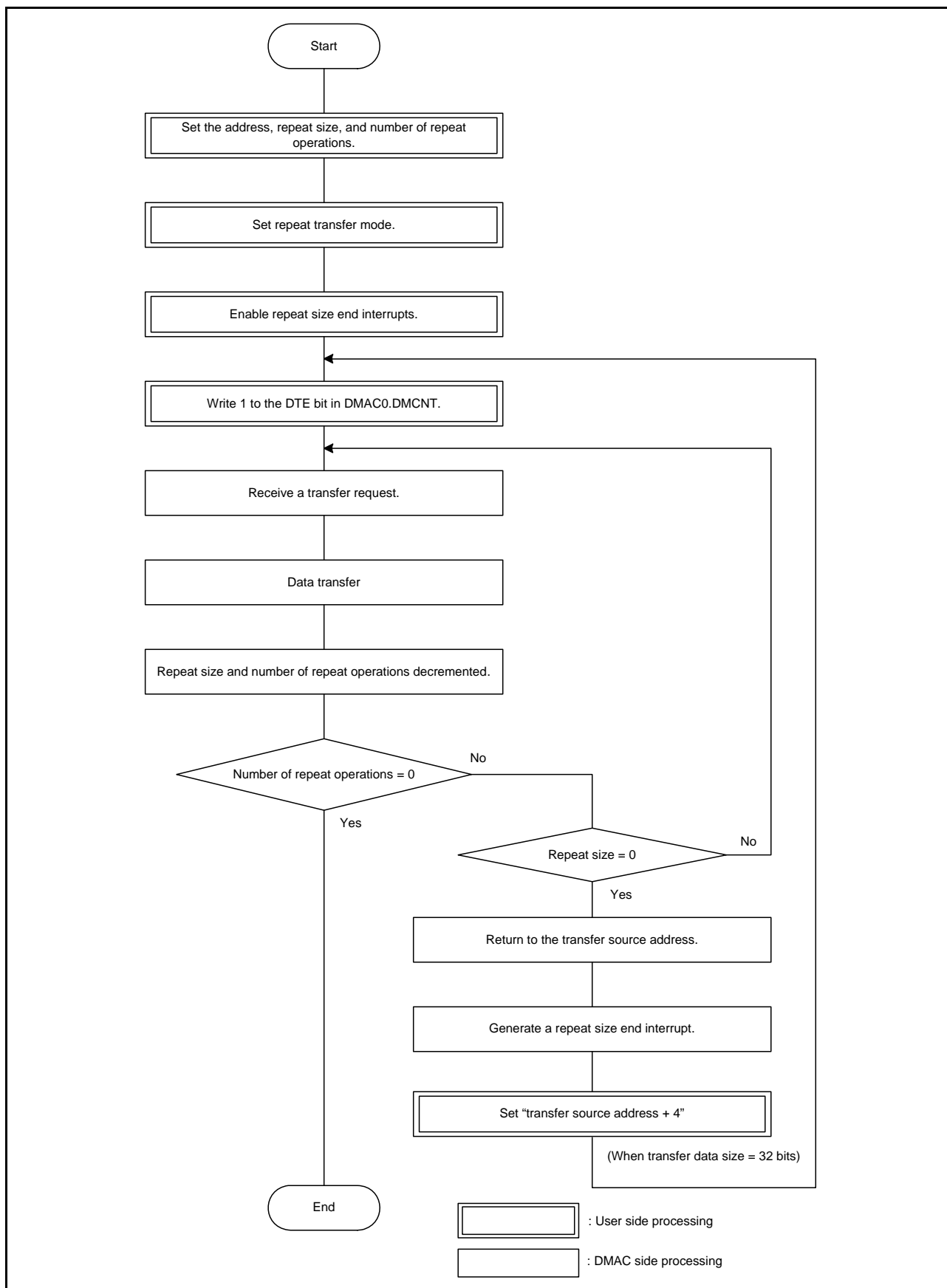


Figure 18.9 XY Conversion Flowchart Using Offset Addition in Repeat Transfer Mode

### 18.3.4 Activation Sources

Software, the interrupt requests from the peripheral modules, and the external interrupt requests can be specified as the DMAC activation sources. Setting the DCTG[1:0] bits in DMTMD of DMACm selects the activation source.

#### (1) DMAC Activation by Software

Setting the DCTG[1:0] bits in DMTMD of DMACm to 00b enables the DMAC activation by software.

To start DMA transfer by software, set the DCTG[1:0] bits in DMTMD of DMACm to 00b, and then set the DTE bit in DMCNT of DMACm to 1 (DMA transfer is enabled) and the SWREQ bit in DMREQ of DMACm to 1 (DMA transfer is requested) with the DMST bit in DMAST set to 1 (DMAC activation enabled).

When the DMAC is activated by software while the CLRS bit in DMREQ of DMACm is 0, the SWREQ bit in DMREQ of DMACm is cleared to 0 after data transfer is started in response to a DMA transfer request.

When the DMAC is activated by software while the CLRS bit is 1, the SWREQ bit is not cleared to 0 after data transfer is started. In this case, a DMA transfer request is issued again after completion of a transfer.

#### (2) DMAC Activation by Interrupt Requests from On-Chip Peripheral Modules or External Interrupt Requests

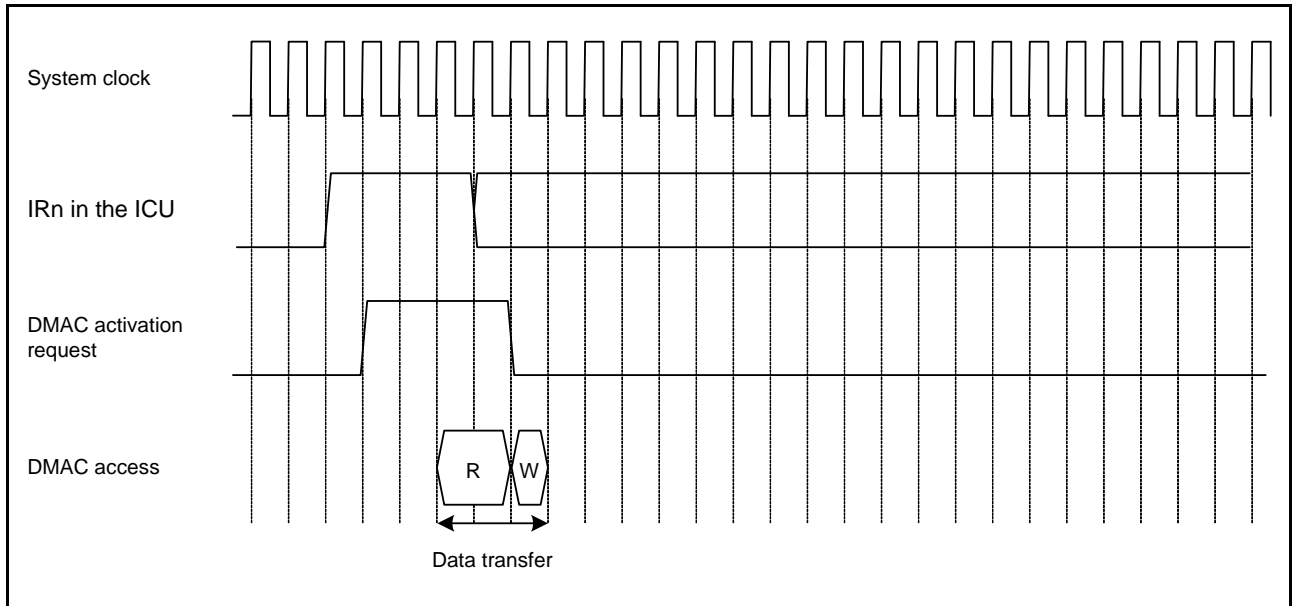
Interrupt requests from the on-chip peripheral modules and external interrupt requests can be specified as the DMAC activation sources. The activation source can be selected separately for each channel using the DMRSRm registers (m = 0 to 3) of the ICU.

The DMAC is activated when an interrupt request from the on-chip peripheral module or an external interrupt request is generated while the DCTG[1:0] bits in DMTMD of DMACm are set to 01b (interrupts from the peripheral modules and the external interrupt pins are selected), the DTE bit in DMCNT of DMACm is set to 1 (DMA transfer is enabled), and the DMST bit in DMAST is set to 1 (DMAC activation is enabled).

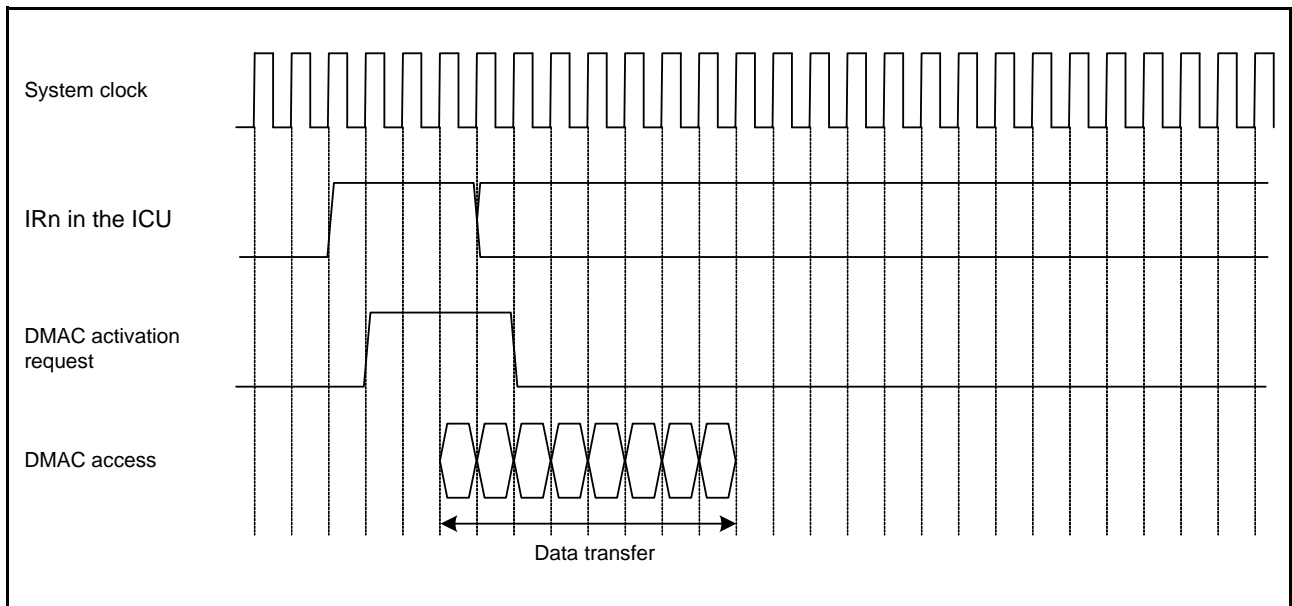
For interrupt requests specified as DMAC activation sources, see Table 15.3, Interrupt Vector Table, in section 15, Interrupt controller (ICUb).

### 18.3.5 Operation Timing

Figure 18.10 and Figure 18.11 show DMAC operation timing examples.



**Figure 18.10 DMAC Operation Timing Example (1) (DMA Activation by Interrupt from Peripheral Module/ External Interrupt Input Pin, Normal Transfer Mode, Repeat Transfer Mode)**



**Figure 18.11 DMAC Operation Timing Example (2) (DMA Activation by Interrupt from Peripheral Module/ External Interrupt Input Pin, Block Transfer Mode, Block Size = 4)**

### 18.3.6 DMAC Execution Cycles

Table 18.7 lists execution cycles in one DMAC data transfer operation.

**Table 18.7 DMAC Execution Cycles**

Transfer Mode	Data Transfer (Read)	Data Transfer (Write)
Normal	Cr+1	Cw
Repeat	Cr+1	Cw
Block*1	P × Cr	P × Cw

Note 1. This is the case when the block size is 2 or more. When the block size is 1, normal transfer cycle is applied.

P: Block size (DMCRAH register setting)

Cr: Data read destination access cycle

Cw: Data write destination access cycle

Cr and Cw depend on the access destination. For the number of cycles for each access destination, see section 42, RAM, section 43, Flash Memory, section 5, I/O Registers, and section 16.2.6, External Bus.

The unit for +1 in “Data Transfer (Read)” column is one system clock cycle (ICLK).

For the operation example, see section 18.3.5, Operation Timing.

### 18.3.7 Activating the DMAC

Figure 18.12 shows the register setting procedure.

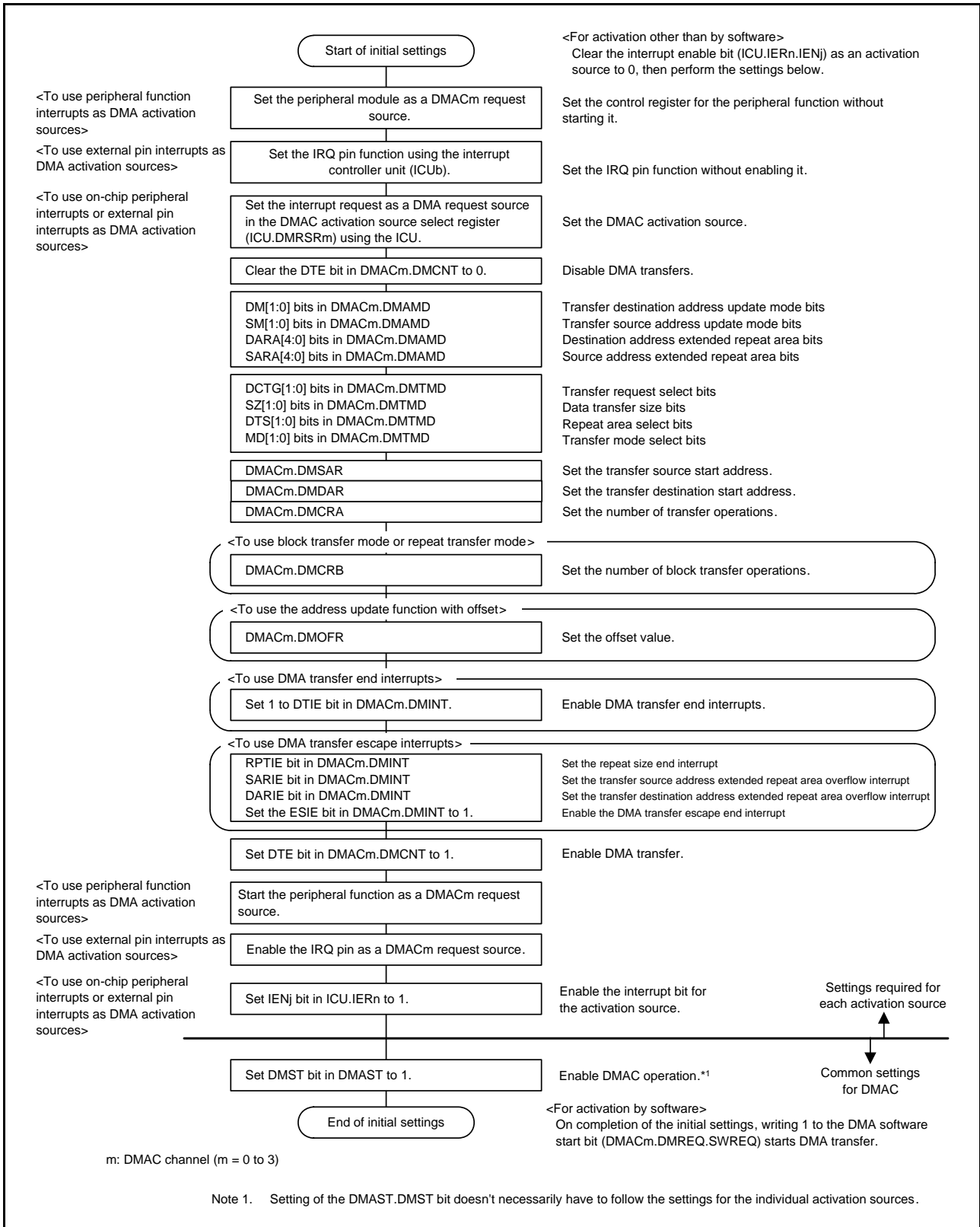


Figure 18.12 Register Setting Procedure

### 18.3.8 Starting DMA Transfer

Setting the DTE bit in DMCNT of DMACm to 1 (DMA transfer enabled) and setting the DMST bit in DMAST to 1 (DMAC start enabled) enable DMA transfer of channel m (m = 0 to 3).

Another activation request cannot be accepted during the transfer of other DMAC channel or DTC. When the proceeding transfer is completed, channel arbitration is performed where a DMA transfer request of the highest priority channel is accepted and DMA transfer of the channel starts. When DMA transfer starts, the ACT bit in DMSTS of DMACm is set to 1 (the DMAC is in the active state).

### 18.3.9 Registers during DMA Transfer

The DMAC registers are updated by a DMA transfer. The value to be updated differs according to the other settings and the transfer state. The registers to be updated are DMSAR, DMDAR, DMCRA, DMCRB, DMCNT, and DMSTS of DMACm.

#### (1) DMA Source Address Register (DMACm.DMSAR)

When data has been transferred in response to one transfer request, the contents of DMSAR are updated to the address to be accessed by the next transfer request.

For details on register update operation in each transfer mode, refer to Table 18.3 to Table 18.5.

#### (2) DMA Destination Address Register (DMACm.DMDAR)

When data has been transferred in response to one transfer request, the contents of DMDAR are updated to the address to be accessed by the next transfer request.

For details on register update operation in each transfer mode, refer to Table 18.3 to Table 18.5.

#### (3) DMA Transfer Count Register (DMACm.DMCRA)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, refer to Table 18.3 to Table 18.5.

#### (4) DMA Block Transfer Count Register (DMACm.DMCRB)

When data has been transferred in response to one transfer request, the count value is updated. The update operation depends on the transfer mode selected.

For details on register update operation in each transfer mode, refer to Table 18.3 to Table 18.5.

#### (5) DMA Transfer Enable Bit (DMACm.DMCNT.DTE)

Although the DMACm.DMCNT.DTE bit enables or disables data transfer by the register write access, it is automatically cleared to 0 by the DMAC according to the DMA transfer state.

The conditions for clearing this bit by the DMAC are as follows:

- When the specified total volume of data transfer is completed
- When DMA transfer is stopped by the repeat size end interrupt
- When DMA transfer is stopped by the extended repeat area overflow interrupt

Writing to the registers for the channels when the corresponding DMACm.DMCNT.DTE bit is set to 1 is prohibited (except for DMACm.DMCNT). In this case, writing must be performed after the bit is cleared to 0.



#### (6) DMA Active Flag (DMACm.DMSTS.ACT)

The ACT bit in DMSTS of DMACm indicates whether the DMACm is in the idle or active state.

This flag is set to 1 when the DMAC starts data transfer, and is cleared to 0 when data transfer in response to one transfer request is completed.

Even when DMA transfer is stopped by writing 0 to the DTE bit in DMCNT of DMACm during DMA transfer, this flag remains 1 until DMA transfer is completed.

#### (7) Transfer End Interrupt Flag (DMACm.DMSTS.DTIF)

The DTIF flag in DMSTS of DMACm is set to 1 after DMA transfer of the total transfer size of data is completed.

When both this flag and the DTIE bit in DMINT of DMACm are set to 1, a transfer end interrupt is requested.

This flag is set to 1 when the DMA transfer bus cycle is completed and the ACT flag in DMSTS of DMACm is cleared to 0 indicating the DMA transfer end.

This flag is automatically cleared to 0 when the DTE bit in DMCNT of DMACm is set to 1 during the interrupt handling.

#### (8) Transfer Escape End Interrupt Flag (DMACm.DMSTS.ESIF)

The ESIF flag in DMSTS of DMACm is set to 1 when a repeat size end interrupt or extended repeat area overflow interrupt is requested. When this bit and the ESIE bit in DMINT of DMACm are set to 1, a transfer escape end interrupt is requested.

This flag is set to 1 when the bus cycle of the DMA transfer having caused the interrupt request is completed and the ACT flag in DMSTS of DMACm is cleared to 0 indicating the DMA transfer end.

This flag is automatically cleared to 0 when the DTE bit in DMCNT of DMACm is set to 1 during an interrupt handling.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set.

For details, see section 15, Interrupt controller (ICUb).

### 18.3.10 Channel Priority

When multiple DMA transfer requests are present, the DMAC determines the priority of channels that have DMA transfer requests.

The channel priority is fixed as channel 0 > channel 1 > channel 2 > channel 3 (channel 0: highest).

When a DMA transfer request is generated during data transfer, channel arbitration is started after the final data has been transferred, and DMA transfer of the higher-priority channel starts.

## 18.4 Ending DMA Transfer

The operation for ending DMA transfer depends on the transfer end conditions. When DMA transfer ends, the DTE bit in DMCNT and the ACT flag in DMSTS of DMACm are changed from 1 to 0, indicating that DMA transfer has ended.

### 18.4.1 Transfer End by Completion of Specified Total Number of Transfer Operations

#### (1) In Normal Transfer Mode (DMACm.DMTMD.MD[1:0] = 00b)

When the value of DMCRAL of DMACm changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in DMCNT of DMACm is cleared to 0 and the DTIF bit in DMSTS of DMACm is set to 1 at the same time. If the DTIE bit in DMINT of DMACm is 1 at this time, a transfer end interrupt request is issued to the CPU or the DTC.

#### (2) In Repeat Transfer Mode (DMACm.DMTMD.MD[1:0] = 01b)

When the value of DMCRB of DMACm changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in DMCNT of DMACm is cleared to 0 and the DTIF bit in DMSTS of DMACm is set to 1 at the same time. If the DTIE bit in DMINT of DMACm is 1 at this time, an interrupt request is issued to the CPU or the DTC.

#### (3) In Block Transfer Mode (DMACm.DMTMD.MD[1:0] = 10b)

When the value of DMCRB of DMACm changes from 1 to 0, DMA transfer ends on the corresponding channel, and the DTE bit in DMCNT of DMACm is cleared to 0 and the DTIF bit in DMSTS of DMACm is set to 1 at the same time. If the DTIE bit in DMINT of DMACm is 1 at this time, an interrupt request is issued to the CPU or the DTC.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt controller (ICUb).

### 18.4.2 Transfer End by Repeat Size End Interrupt

In repeat transfer mode, a repeat size end interrupt is requested when transfer of a 1-repeat size of data is completed while the RPTIE bit in DMINT of DMACm is set to 1. When the interrupt is requested to complete DMA transfer, the DTE bit in DMCNT of DMACm is cleared to 0 and the ESIF flag in DMSTS of DMACm is set to 1. If the ESIE bit in DMINT of DMACm is 1 at this time, an interrupt request is issued to the CPU or the DTC. Here, the transfer can be resumed by writing 1 to the DTE bit in DMCNT of DMACm.

A repeat size end interrupt can be requested also in block transfer mode. In block transfer mode, the interrupt is requested in the same way as in repeat transfer mode when transfer of a 1-block size data is completed.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt controller (ICUb).

### 18.4.3 Transfer End by Interrupt on Extended Repeat Area Overflow

When an overflow on the extended repeat area occurs while the extended repeat area is specified and the SARIE or DARIE bit in DMINT of DMACm is set to 1, an interrupt by an extended repeat area overflow is requested. When the interrupt is requested, the DMA transfer is terminated, the DTE bit in DMCNT of DMACm is cleared to 0, and the ESIF flag in DMSTS of DMACm is set to 1. If the ESIE bit in DMINT of DMACm is 1 at this time, an interrupt request is issued to the CPU or the DTC.

Even if an interrupt by an extended repeat area overflow is requested during a read cycle, the following write cycle is performed.

In block transfer mode, even if an interrupt by an extended repeat area overflow is requested during a 1-block transfer, the remaining data in the block is transferred; transfer is terminated after a block transfer.

Before sending an interrupt request from the DMAC to the CPU or the DTC, the interrupt control register must be set. For details, see section 15, Interrupt controller (ICUb).

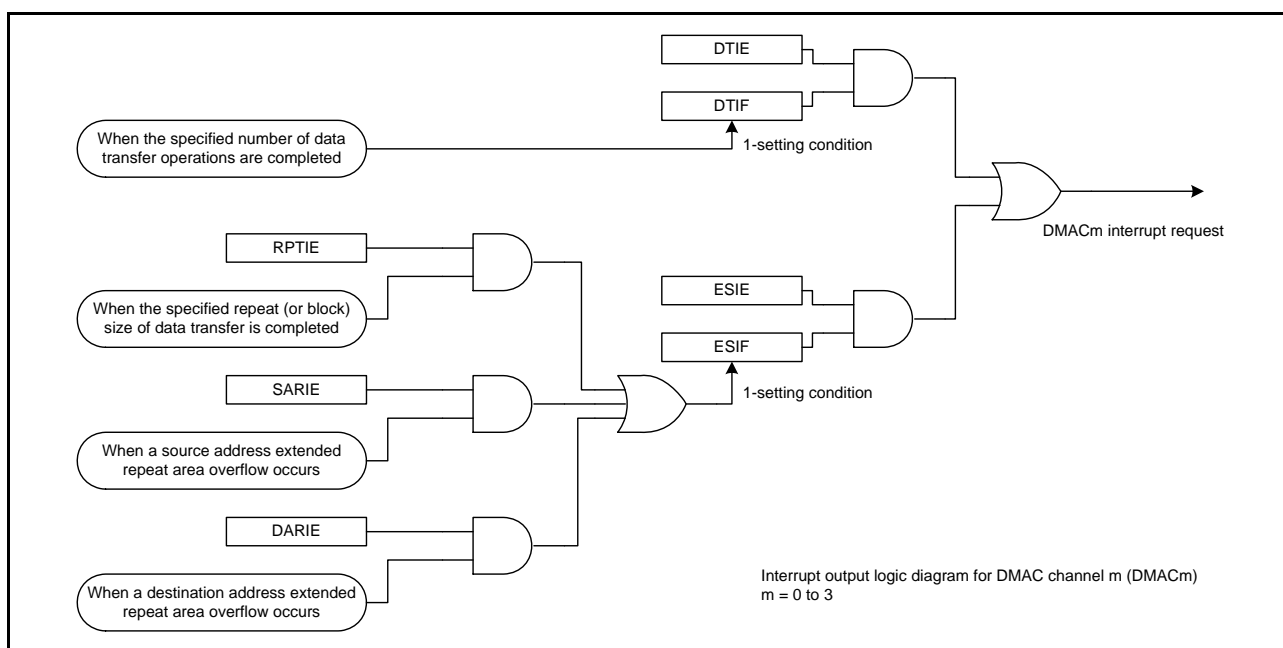
### 18.5 Interrupts

Each DMAC channel can output an interrupt request to the CPU or the DTC after transfer in response to one request is completed. When the transfer destination is the external bus or the on-chip peripheral bus, an interrupt request is generated upon completion of data write to the write buffer not to the actual transfer destination.

Table 18.8 lists the relation among the interrupt sources, the interrupt status flags, and the interrupt enable bits. Figure 18.13 shows the schematic logic diagram of interrupt outputs. Figure 18.14 shows the DMAC interrupt handling routine to resume or terminate DMA transfer.

**Table 18.8 Relation among Interrupt Sources, Interrupt Status Flags, and Interrupt Enable Bits**

Interrupt Sources		Interrupt Enable Bits	Interrupt Status Flags	Request Output Enable Bits
Transfer end		—	DMACm.DMSTS.DTIF	DMACm.DMINT.DTIE
Escape transfer end	Repeat size end	DMACm.DMINT.RPTIE	DMACm.DMSTS.ESIF	DMACm.DMINT.ESIE
	Source address extended repeat area overflow	DMACm.DMINT.SARIE		
	Destination address extended repeat area overflow	DMACm.DMINT.DARIE		



**Figure 18.13 Schematic Logic Diagram of Interrupt Outputs**

Specifically, the different procedures are used for canceling an interrupt to restart DMA transfer in the following two cases: (1) discontinuing or terminating DMA transfer and (2) continuing DMA transfer.

(1) When Discontinuing or Terminating DMA Transfer

Write 0 to the DTIF bit in DMSTS of DMACm to clear a transfer end interrupt, and to the ESIF bit in DMSTS of DMACm to clear a repeat size interrupt and an extended repeat area overflow interrupt. The DMACm remains in the stop state. When starting another DMA transfer after that, set the appropriate registers, and set the DTE bit in DMCNT of DMACm to 1 (DMA transfer enabled).

(2) When Continuing DMA Transfer

Write 1 to the DTE bit in DMCNT of DMACm. The ESIF bit in DMSTS of DMACm is automatically cleared to 0 (interrupt source cleared), and DMA transfer is resumed.

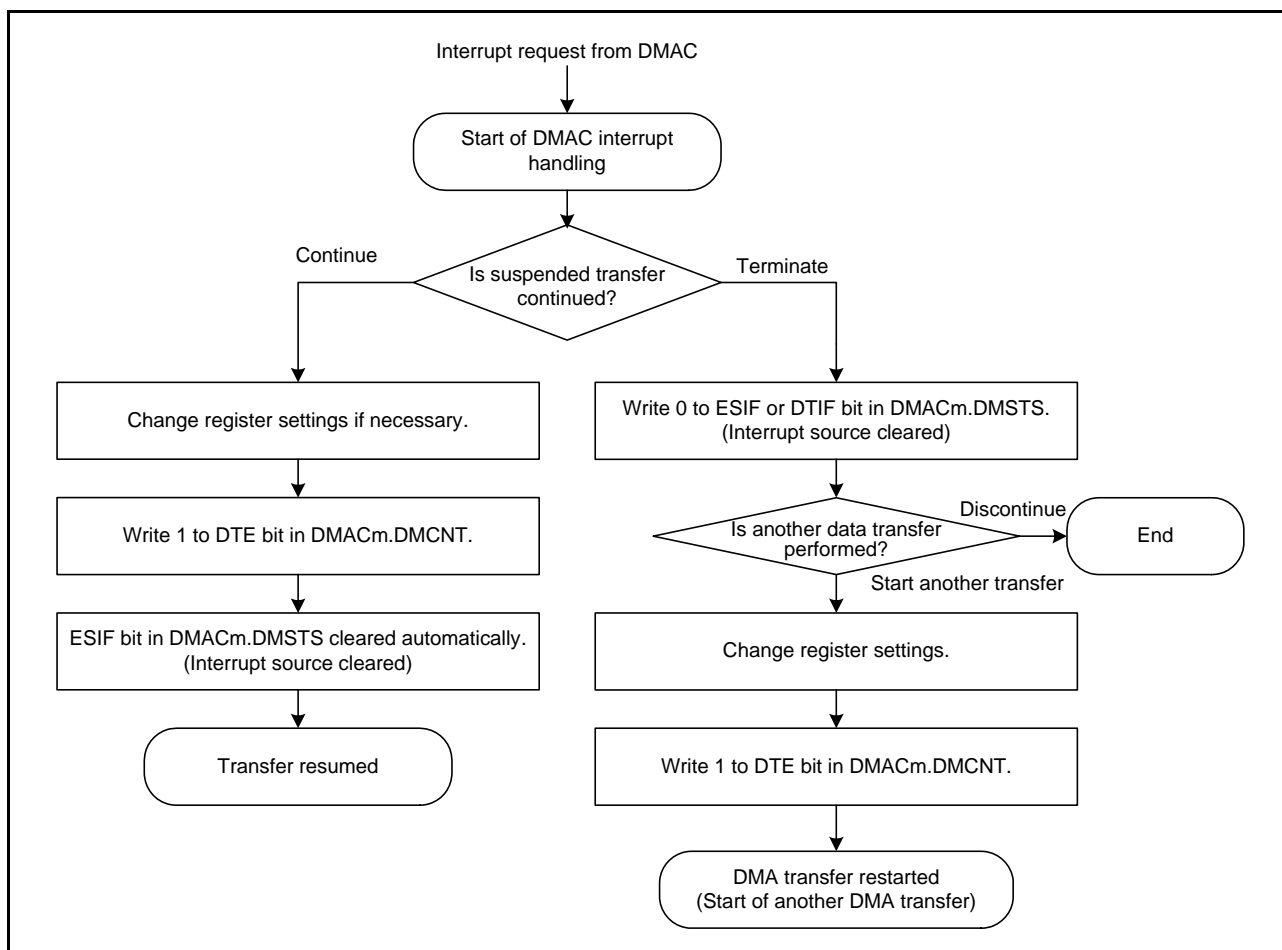


Figure 18.14 DMAC Interrupt Handling Routine to Resume/Terminate DMA Transfer

## 18.6 Low-Power Consumption Function

Before transition to the module-stop state, all-module clock stop mode, software standby mode, or deep software standby mode, clear the DMST bit in DMAST to 0 (the DMAC suspended), and then perform the following.

### (1) Module-Stop Function

Writing 1 to the MSTPA28 bit (transition to the module-stop state) in MSTPCRA enables the module-stop function of the DMAC. If DMA transfer is in progress at the time a 1 is written to the MSTPA28 bit, the transition to the module-stop state proceeds after DMA transfer has ended. While the MSTPA28 bit is 1, accessing the DMAC registers are prohibited.

Writing 0 to the MSTPA28 bit releases the DMAC from the module-stop state.

### (2) All-Module Clock Stop Mode

Make settings in accord with the procedure under section 11.6.2.1, Transition to All-Module Clock Stop Mode, in section 11, Low Power Consumption.

If DMA transfer operations are in progress at the time the WAIT instruction is executed, the transition to all-module clock stop mode follows the completion of DMA transfer.

The DMAC is released from the module-stop state by writing 0 to the MSTPCRA.MSTPA28 bit following recovery from all-module clock stop mode.

### (3) Software Standby and Deep Software Standby Modes

Make settings in accord with the procedure under section 11.6.3.1, Transition to Software Standby Mode or section 11.6.4.1, Transition to Deep Software Standby Mode, in section 11, Low Power Consumption.

If DMA transfer operations are in progress at the time the WAIT instruction is executed, the transition to software standby or deep software standby follows the completion of DMA transfer.

### (4) Note on Low-Power Consumption Function

For the WAIT instruction and the register setting procedure, see section 11.7.6, Timing of Wait Instructions in section 11, Low Power Consumption.

To perform DMA transfer after returning from low-power consumption mode, set the DMST bit in DMAST to 1 again. To use a request that is generated in all-module clock-stop mode and software standby mode as an interrupt request to the CPU but not as a DMAC startup request, specify the CPU as the interrupt request destination in accordance with the description in section 15.5.3, Selecting Interrupt Request Destinations in section 15, Interrupt controller (ICUb), and then execute the WAIT instruction.

## 18.7 Usage Notes

### 18.7.1 DMA Transfer to External Devices

In DMA transfer to an external device, the ACT bit in DMSTS of DMACm may be cleared to 0 (DMAC transfer suspended) during the period from the beginning of the final data write to the end of the external bus access.

### 18.7.2 DMA Transfer to Peripheral Modules

In DMA transfer to a peripheral module, the ACT bit in DMSTS of DMACm may be cleared to 0 (DMAC transfer suspended) during the period from the beginning of the final data write to the end of the peripheral bus access.

### 18.7.3 Access to the Registers during DMA Transfer

The DMSAR, DMDAR, DMCRA, DMCRB, DMTMD, DMINT, DMAMD, DMOFR, and DMCSL registers of DMACm must not be accessed while the ACT bit in DMSTS of the same channel is set to 1 (DMAC active state) or the DTE bit in DMCNT of the same channel is set to 1 (DMA transfer enabled).

### 18.7.4 DMA Transfer to Reserved Areas

DMA transfer to the reserved areas is prohibited. If such an access is made, transfer results are not guaranteed. For details on the reserved areas, see section 4, Address Space.

### 18.7.5 Interrupt Request by the DMA Activation Source Flag Control Register (DMCSL) at the End of Each Transfer

While the DMACm.DMCSL.DISEL bit is 1, an interrupt is issued to the CPU at the end of each transfer that has been activated by one DMA request. Unlike the transfer end interrupt that the DMAC outputs or the escape end interrupt, the interrupt of this type is issued to the CPU at the end of DMA transfer without clearing the interrupt flag of the DMAC activation source to 0 by changing the interrupt request destination to the CPU. In this case, since the interrupt flag is not cleared to 0 at the end of DMAC transfer, it should be cleared to 0 by the CPU interrupt routine.

The interrupt flag is cleared when the CPU interrupt is accepted.

For the change of the settings on the interrupt flag or the interrupt request destination, see section 15, Interrupt controller (ICUb). For the DMACm.DMCSL.DISEL bit setting, see section 18.2.12, DMA Activation Source Flag Control Register (DMCSL).

### 18.7.6 Setting of DMAC Activation Request Select Register of the Interrupt Controller (ICU.DMRSRm)

The DMAC activation request select register (ICU.DMRSRm) should be set while the DMA transfer enable bit (DMACm.DMCNT.DTE) is cleared to 0 (DMA transfer is disabled). Moreover, the DTC activation enable register (ICU.DTCERm) that corresponds to the same vector number that has been set by the ICU.DMRSRm register should not be set to 1. For details on the ICU.DTCERn and ICU.DMRSRm, see section 15, Interrupt controller (ICUb).

### 18.7.7 Suspending or Restarting DMA Activation

To suspend a DMA activation request, write 0 to the interrupt enable bit for the activation source (ICU.IERn.IENj bit).

To restart the DMA transfer, write 1 to the ICU.IERn.IENj bit with the setting shown in section 18.3.7, Activating the DMAC.

## 19. Data Transfer Controller (DTCa)

The RX630 Group incorporates a data transfer controller (DTC).

The DTC is activated by an interrupt request to control data transfer.

### 19.1 Overview

Table 19.1 lists the specifications of the DTC, and Figure 19.1 shows a block diagram of the DTC.

**Table 19.1 DTC Specifications**

Item	Description
Transfer mode	<ul style="list-style-type: none"> <li>• Normal transfer mode A single activation leads to a single data transfer.</li> <li>• Repeat transfer mode A single activation 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 repeat size is 256.</li> <li>• Block transfer mode A single activation leads to the transfer of a single block. The maximum block size is 256 data.</li> </ul>
Transfer channel	<ul style="list-style-type: none"> <li>• Channel transfer corresponding to the interrupt source is possible (transferred by DTC activation request from the ICU).</li> <li>• Data of multiple channels can be transferred on a single activation source (chain transfer).</li> <li>• Either "executed when the counter is 0" or "always executed" can be selected for chain transfer.</li> </ul>
Transfer space	<ul style="list-style-type: none"> <li>• In short-address mode: 16 Mbytes (Areas from 0000 0000h to 007F FFFFh and FF80 0000h to FFFF FFFFh excepting reserved areas)</li> <li>• In full-address mode: 4 Gbytes (Area from 0000 0000h to FFFF FFFFh excepting reserved areas)</li> </ul>
Data transfer units	<ul style="list-style-type: none"> <li>• Length of a single data: 8, 16, or 32 bits</li> <li>• Number of data for a single block: 1 to 256 data</li> </ul>
CPU interrupt source	<ul style="list-style-type: none"> <li>• An interrupt request can be generated to the CPU on a DTC activation interrupt.</li> <li>• An interrupt request can be generated to the CPU after a single data transfer.</li> <li>• An interrupt request can be generated to the CPU after data transfer of specified volume.</li> </ul>
Read skip	Transfer data read skip can be specified.
Write-back skip	When "fixed" is selected for transfer source address or transfer destination address, write-back skip execution is provided.
Lower power consumption function	Module-stop state can be specified.

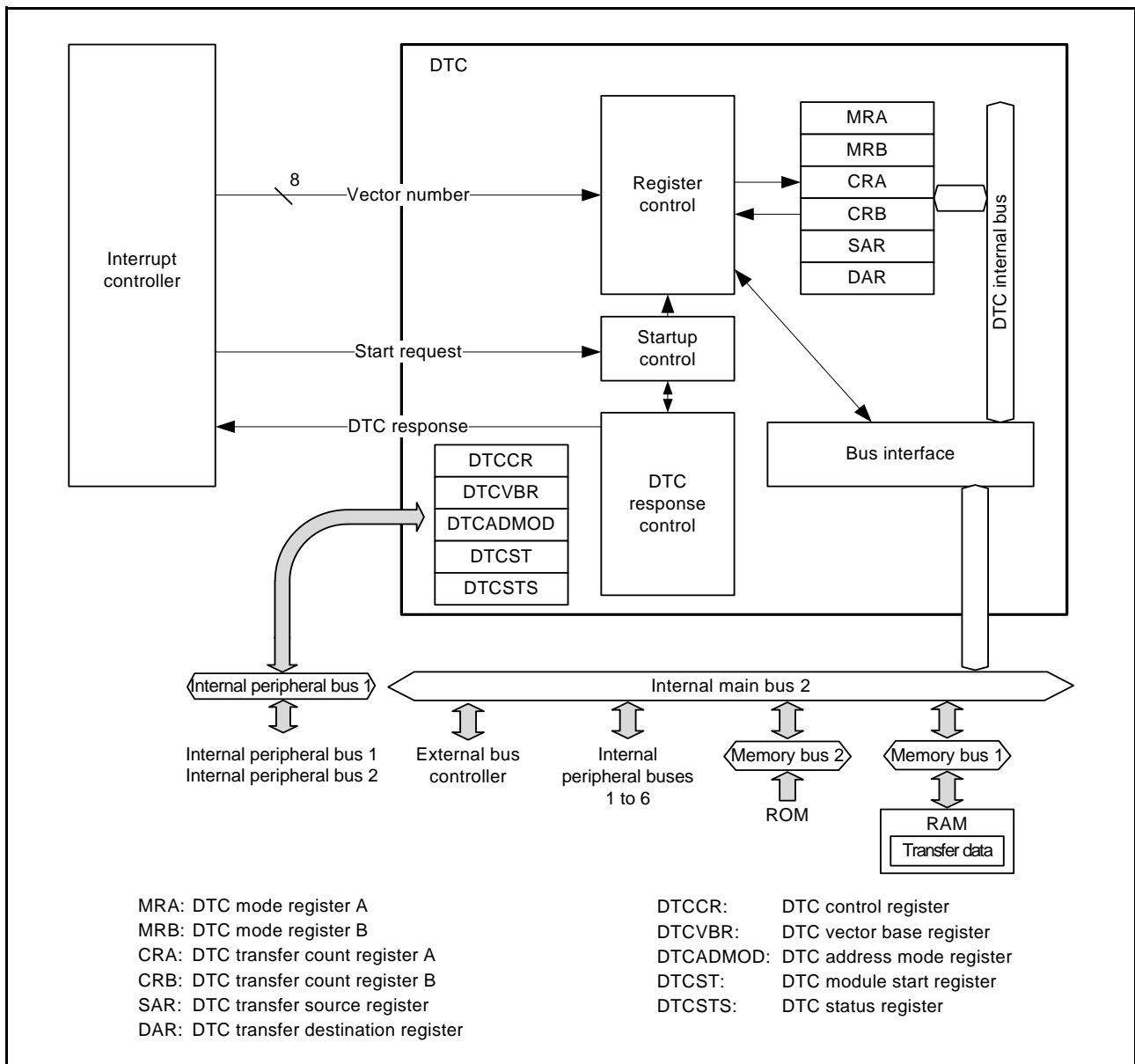


Figure 19.1 Block Diagram of DTC

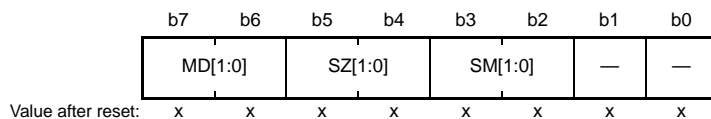


## 19.2 Register Descriptions

Registers MRA, MRB, SAR, DAR, CRA, and CRB are DTC internal registers, which cannot be directly accessed from the CPU. Values to be set in the DTC internal registers are placed in the RAM area as transfer information data. When an activation request is generated, the DTC reads the transfer information data from the RAM area and set them in the internal registers. After the data transfer ends, the internal register contents are written back to the RAM area as transfer information data.

### 19.2.1 DTC Mode Register A (MRA)

Address(es): (inaccessible directly from the CPU)



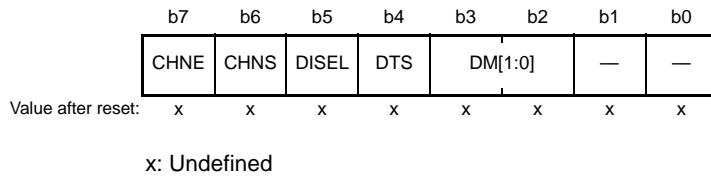
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as undefined. The write value should be 0.	—
b3, b2	SM[1:0]	Transfer Source Address Addressing Mode	b3 b2 0 0: Address in the SAR register is fixed (Write-back to SAR is skipped) 0 1: Address in the SAR register is fixed (Write-back to SAR is skipped) 1 0: SAR value is incremented after data transfer (+1 when SZ[1:0] bits = 00b, +2 when SZ[1:0] bits = 01b, +4 when SZ[1:0] bits = 10b) 1 1: SAR value is decremented after data transfer (−1 when SZ[1:0] bits = 00b, −2 when SZ[1:0] bits = 01b, −4 when SZ[1:0] bits = 10b)	—
b5, b4	SZ[1:0]	DTC Data Transfer Size	b5 b4 0 0: 8-bit (byte) transfer 0 1: 16-bit (word) transfer 1 0: 32-bit (longword) 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 cannot be accessed directly from the CPU.

## 19.2.2 DTC Mode Register B (MRB)

Address(es): (inaccessible directly from the CPU)



Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as undefined. The write value should be 0.	—
b3, b2	DM[1:0]	Transfer Destination Address Addressing Mode	b3 b2 0 0: Address in the DAR register is fixed (Write-back to DAR is skipped) 0 1: Address in the DAR register is fixed (Write-back to DAR is skipped) 1 0: DAR value is incremented after data transfer (+1 when SZ[1:0] bits in MRA = 00b, +2 when SZ[1:0] bits = 01b, +4 when SZ[1:0] bits = 10b) 1 1: DAR value is decremented after data transfer (−1 when SZ[1:0] bits in MRA = 00b, −2 when SZ[1:0] bits = 01b, −4 when SZ[1:0] bits = 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 when specified data transfer is completed 1: An interrupt request to the CPU is generated each time DTC data transfer is performed	—
b6	CHNS	DTC Chain Transfer Select	0: Chain transfer is performed continuously 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	—

MRB cannot be accessed directly from the CPU.

### 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, see Table 19.3, Chain Transfer Conditions.

When the next transfer is chain transfer, completion of the specified number of transfers is not determined, the startup source flag is not cleared, and an interrupt request to the CPU is not generated.

### CHNE Bit (DTC Chain Transfer Enable)

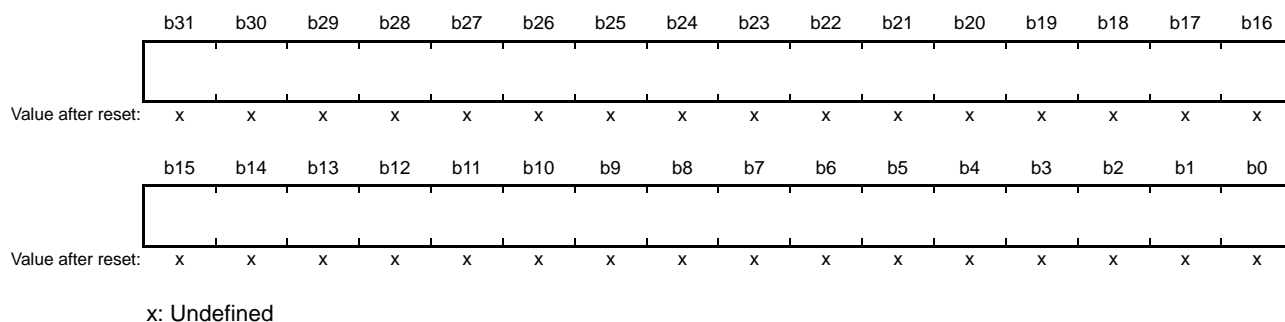
The CHNE bit enables or disables chain transfer.

The chain transfer condition is selected by the CHNS bit.

For details of chain transfer, see section 19.4.6, Chain Transfer.

### 19.2.3 DTC Transfer Source Register (SAR)

Address(es): (inaccessible directly from the CPU)



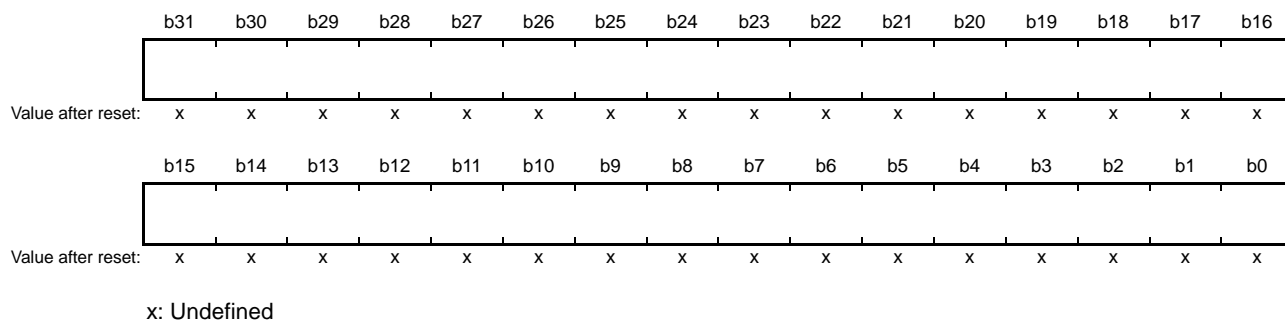
In full-address mode, 32 bits are valid.

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 cannot be accessed directly from the CPU.

### 19.2.4 DTC Transfer Destination Register (DAR)

Address(es): (inaccessible directly from the CPU)



In full-address mode, 32 bits are valid.

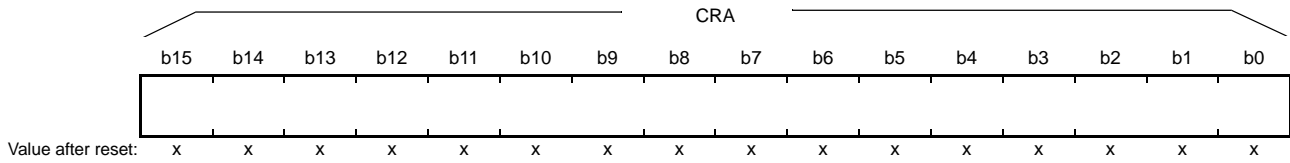
In short-address mode, lower 24 bits are valid and upper 8 bits (b31 to b24) are ignored. The address of this register is extended by the value specified by b23.

DAR cannot be accessed directly from the CPU.

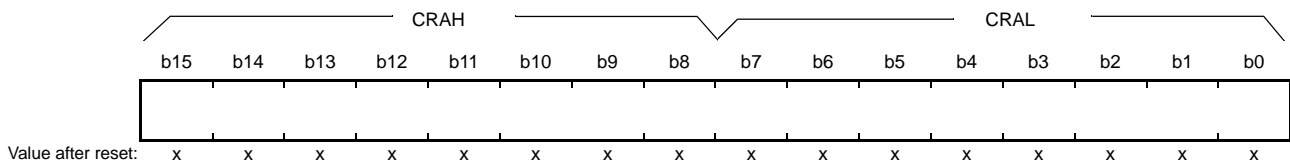
### 19.2.5 DTC Transfer Count Register A (CRA)

Address(es): (inaccessible directly from the CPU)

- Normal transfer mode



- Repeat transfer mode/block transfer mode



x: Undefined

Note: • The function depends on transfer mode.

Symbol	Register Name	Description	R/W
CRAL	Transfer Counter A Lower Register	Set transfer count.	—
CRAH	Transfer Counter A Upper Register		—

Note: • Set CRAH and CRAL to the same value in repeat transfer mode and block transfer mode.

CRA cannot be accessed directly from the CPU.

#### (1) Normal transfer mode (MD[1:0] bits in MRA = 00b)

CRA 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 (MD[1:0] bits in MRA = 01b)

The CRAH register retains 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 transferred to CRAL.

#### (3) Block transfer mode (MD[1:0] bits in MRA = 10b)

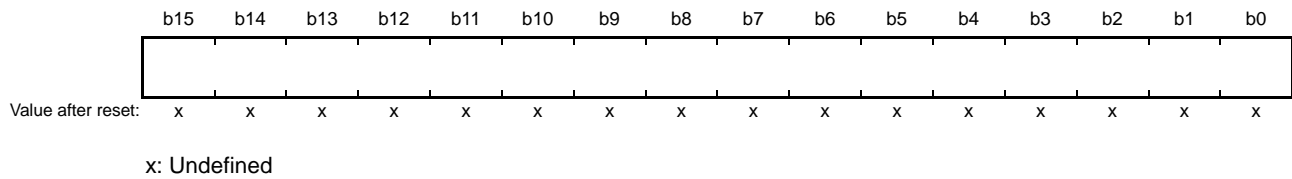
The CRAH register retains block size and the CRAL register functions as an 8-bit block size counter.

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 transferred to CRAL.

### 19.2.6 DTC Transfer Count Register B (CRB)

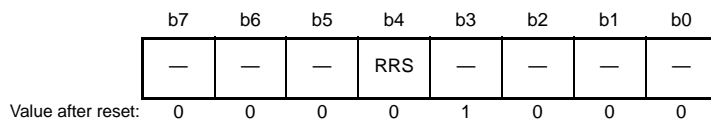
Address(es): (inaccessible directly from the CPU)



CRB is used to set the block transfer count for block transfer mode. 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. CRB cannot be accessed directly from the CPU.

### 19.2.7 DTC Control Register (DTCCR)

Address(es): 0008 2400h



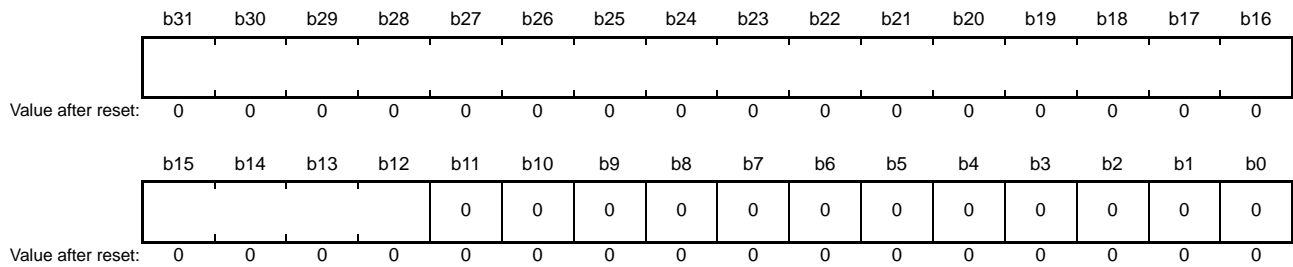
Bit	Symbol	Bit Name	Description	R/W
b2 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b3	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b4	RRS	DTC Transfer Data Read Skip Enable	0: Transfer data read is not skipped 1: Transfer data read is skipped when vector numbers match.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

#### RRS Bit (DTC Transfer Data Read Skip Enable)

The DTC vector number is always compared with the vector number in the previous startup process. When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the transferred data. However, when the previous transfer was chain transfer, the transferred data is always read regardless of the value of 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 data is always read regardless of the value of RRS bit.

### 19.2.8 DTC Vector Base Register (DTCVBR)

Address(es): 0008 2404h



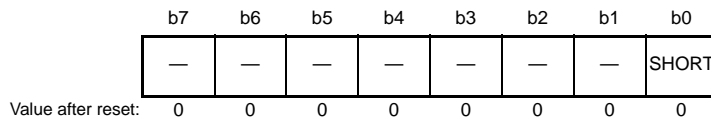
Bit	Bit Name	Description	R/W
b11 to b0	DTC vector address (the lower 12 bits)	These bits are read as 0. The write value should be 0.	R
b31 to b12	DTC vector base address (the upper 20 bits)	The upper 4 bits (b31 to b28) are ignored, and the address of this register is extended by the value specified by b27.	R/W

DTCVBR is used to set the base address for calculating the DTC vector table address.

It can be set in the range of 0000 0000h to 07FF F000h and F800 0000h to FFFF F000h in 4K-byte units.

### 19.2.9 DTC Address Mode Register (DTCADM0D)

Address(es): 0008 2408h



Bit	Symbol	Bit Name	Description	R/W
b0	SHORT	Short-Address Mode	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

DTCADM0D is used to specify the area accessible by the DTC.

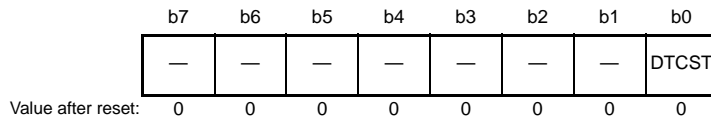
#### SHORT Bit (Short-Address Mode)

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).

### 19.2.10 DTC Module Start Register (DTCST)

Address(es): 0008 240Ch



Bit	Symbol	Bit Name	Description	R/W
b0	DTCST	DTC Module Start	0: DTC module-stop 1: DTC module-stop	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)

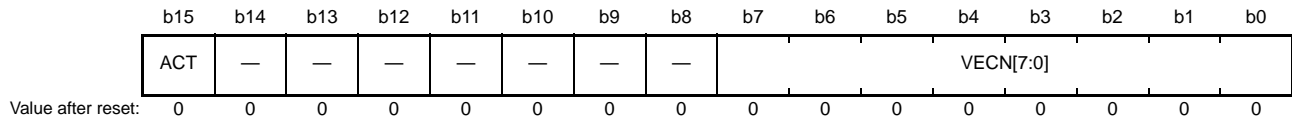
Set the DTCST bit to 1 to enable the DTC to accept transfer requests. When this bit is cleared to 0, transfer requests are no longer accepted.

If this bit is cleared to 0 during data transfer, the accepted transfer request is active until the processing is completed. Before making transition to the module-stop state, all-module clock-stop mode, software standby mode, or deep software standby mode, the DTCST bit must be set to 0.

For details on transitions to the module-stop state, all-module clock-stop mode, software standby mode, and deep software standby mode, refer to section 19.8, Low-Power Consumption Function, and section 11, Low Power Consumption.

### 19.2.11 DTC Status Register (DTCSTS)

Address(es): 0008 240Eh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	VECN[7:0]	DTC-Activating Vector Number Monitoring	These bits indicate the vector number for the activating source when DTC transfer is in progress. The value is only valid if DTC 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 this bit has no effect.	R
b15	ACT	DTC Active Flag	0: DTC transfer operation is not in progress. 1: DTC transfer operation is in progress.	R

#### VECN[7:0] Bits (DTC-Activating Vector Number Monitoring)

While transfer by the DTC is in progress, these bits indicate the vector number corresponding to the activating source for the transfer.

When the DTCSTS register is read, the value read from the VECN[7:0] bits is valid if the value of the ACT flag was 1 (indicating DTC transfer in progress) and invalid if the value of the ACT flag was 0 (indicating no current DTC transfer). For the correspondence between the DTC startup sources and the vector addresses, see Table 15.3, Interrupt Vector Table in section 15, Interrupt controller (ICUb).

#### ACT Bit (DTC Active Flag)

This bit indicates the state of DTC transfer operation.

[Setting condition]

- When the DTC is activated by a transfer request

[Clearing condition]

- When transfer by the DTC is completed in response to a transfer request.



### 19.3 Sources of Activation

The DTC is activated by an interrupt request. Setting the DTCERn.DTCE bit (where n is the interrupt vector number of the given interrupt) of the ICU to 1 selects the corresponding interrupt as an activation source for the DTC.

For the correspondence between the DTC startup sources and the vector addresses, see Table 15.3, Interrupt Vector Table in section 15, Interrupt controller (ICUb). For startup by software, see section 15.2.5, Software Interrupt Activation Register (SWINTR).

Once the DTC has accepted a startup request, it does not accept another startup request until transfer for that single request is completed, regardless of the priority of the requests. When multiple startup requests are generated during DMAC/DTC transfer, a request with the highest priority on completion of the transfer is accepted. When multiple startup requests are generated while the DTC module start (DTCST) bit in DTCST is 0, a request with the highest priority at the moment when the bit is subsequently set to 1 is accepted.

The following operations are performed for each single round of data transfer (or the last of the consecutive transfers in the case of a chained transfer).

- On completion of a specified round of data transfer, the DTCERn.DTCE bit is cleared 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 startup source is cleared to 0 at the start of data transfer.

#### 19.3.1 Allocating Transfer Data and DTC Vector Table

The DTC reads the start address of the transfer data corresponding to each startup source from the vector table and reads the transfer data starting at that address.

The vector table should be located so that the lower 12 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 data is allocated in the RAM area. In the RAM area, the start address of the transfer data (n) with vector number n should be 4n added to the base address in the vector table.

Transfer data can be allocated in short-address mode (3 longwords) or full-address mode (4 longwords). Use the SHORT bit in DTCADMOD to select short-address mode (SHORT bit = 1) or full-address mode (SHORT bit = 0).

Figure 19.2 shows the relationship between the DTC vector table and transfer data.

Figure 19.3 shows the allocation of transfer data in the RAM area. The lower addresses vary according to the endian of the corresponding allocation area. For details, see section 19.9.2, Allocating Transfer Data.

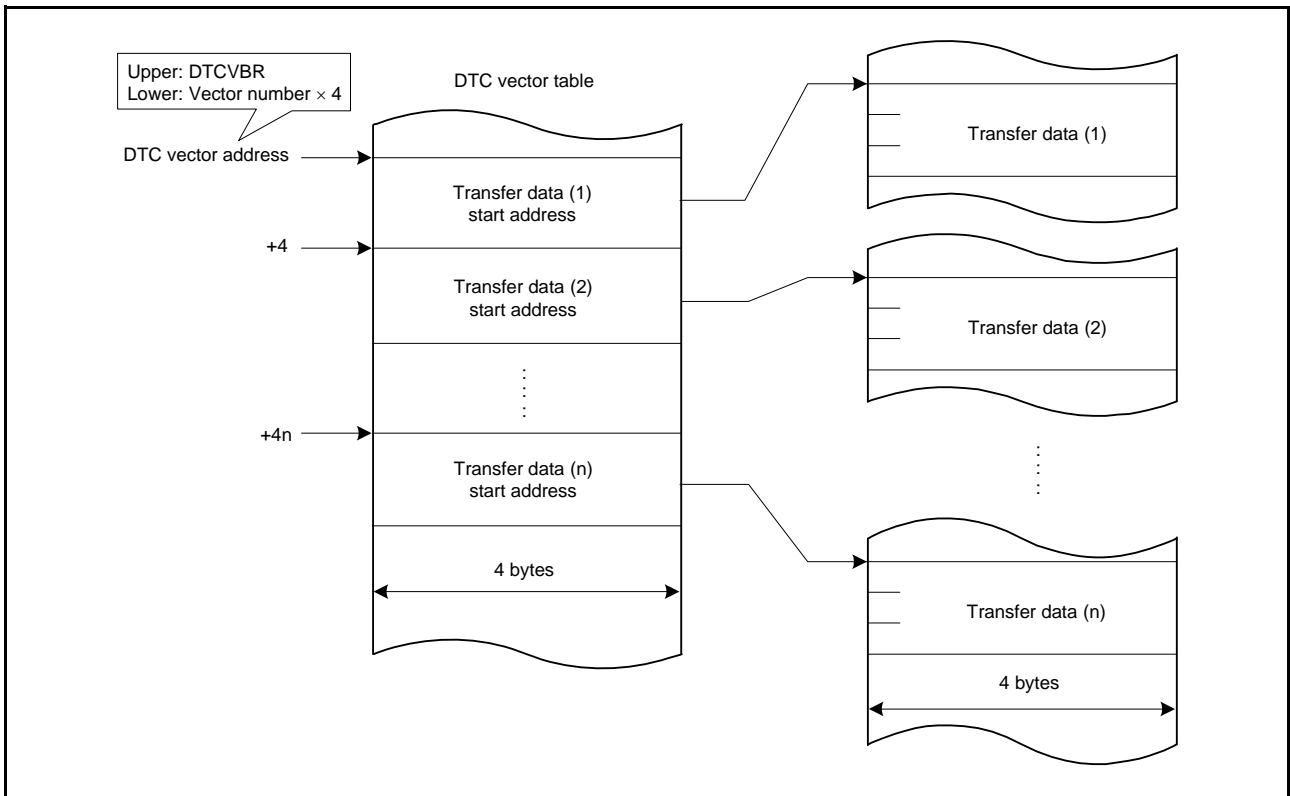


Figure 19.2 DTC Vector Table and Transfer Data

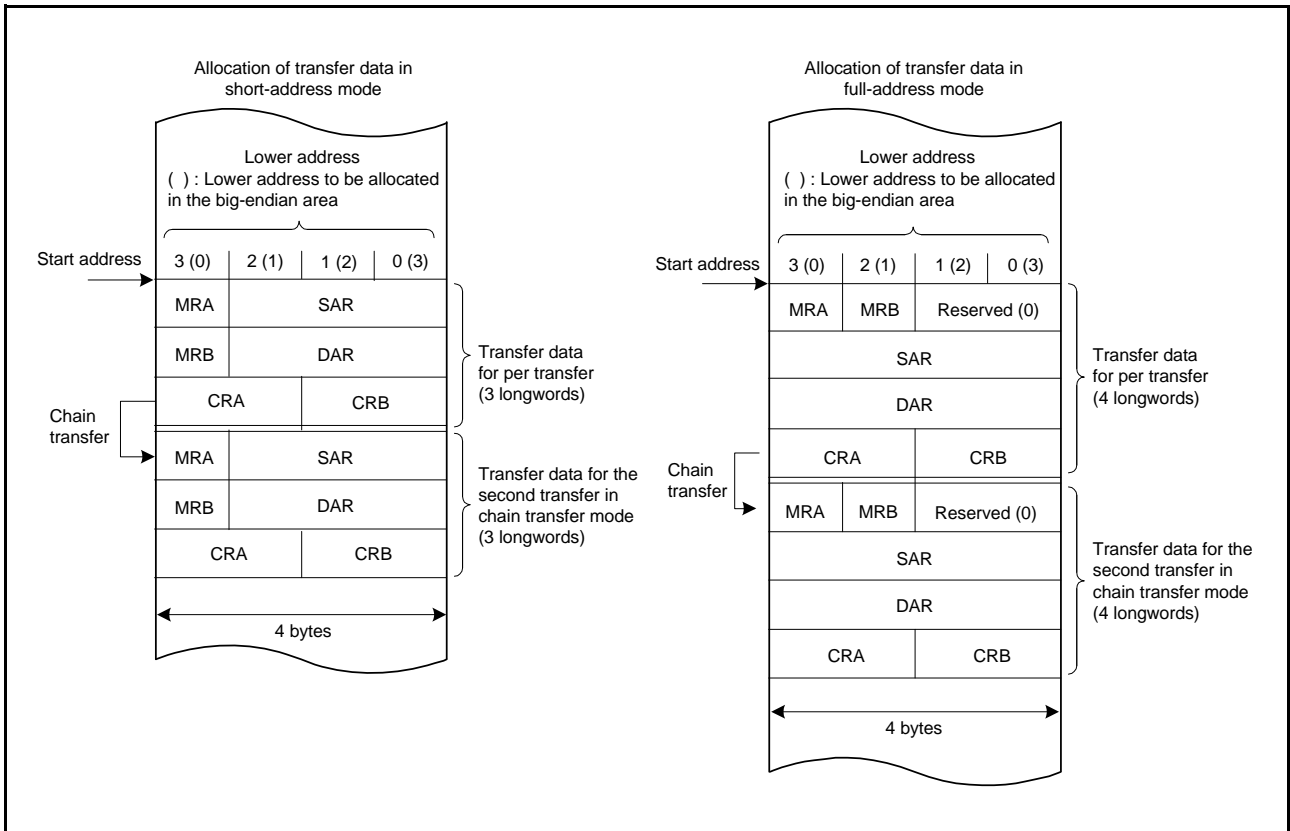


Figure 19.3 Allocation of Transfer Data in the RAM Area

## 19.4 Operation

The DTC transfers data in accordance with the transfer data. Storage of the transfer data in the RAM area is required before DTC operation.

When the DTC is activated, it reads the DTC vector corresponding to the vector number. Then the DTC reads transfer data from the transfer data store address pointed by the DTC vector, transfers data, and then writes back the transfer data after the data transfer. Storing transfer data 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.

The DTC specifies a transfer source address in SAR and a transfer destination address in DAR. The values of these registers are incremented, decremented, or address-fixed independently after data transfer.

Table 19.2 lists transfer modes of the DTC.

**Table 19.2 Transfer Modes of the DTC**

Transfer Mode	Data Size Transferred on a Single Transfer Request	Increment/Decrement of Memory Address	Settable Transfer Count
Normal transfer mode	1 byte/word/longword	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 65536
Repeat transfer mode* <sup>1</sup>	1 byte/word/longword	Incremented/decremented by 1, 2, or 4 or address fixed	1 to 256* <sup>3</sup>
Block transfer mode* <sup>2</sup>	Block size specified in CRAH (1 to 256 bytes/words/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 CHNE bit in MRB to 1 allows multiple transfers (chain transfer) on a single startup source. Setting the CHNS bit in MRB also enables chain transfer when specified data transfer is completed.

Figure 19.4 shows the operation flowchart of the DTC. Table 19.3 lists chain transfer conditions.

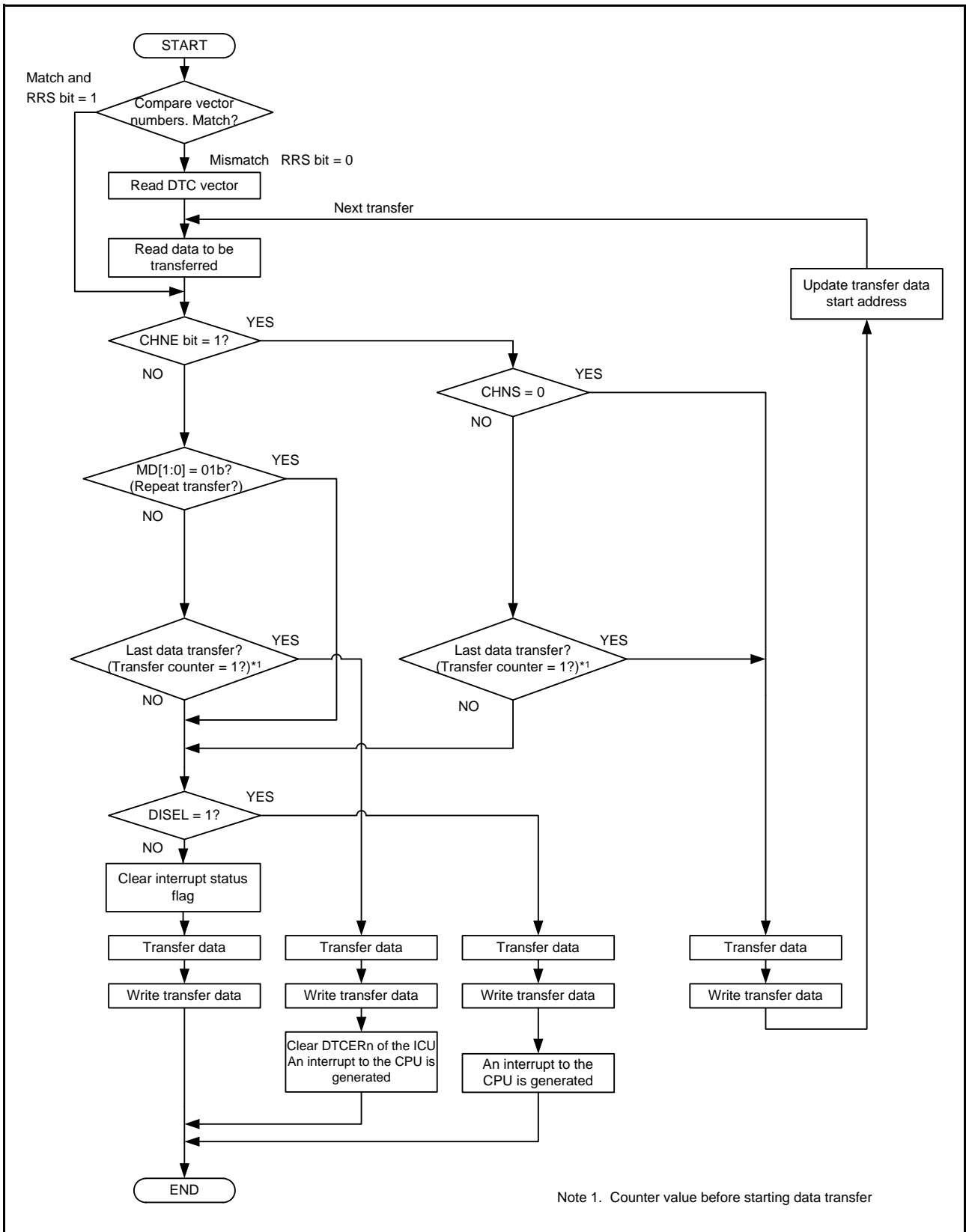


Figure 19.4 Operation Flowchart of the DTC

**Table 19.3 Chain Transfer Conditions**

First Transfer				Second Transfer <sup>*3</sup>				DTC Transfer
CHNE Bit	CHNS Bit	DISEL Bit	Transfer Counter <sup>*1,*2</sup>	CHNE Bit	CHNS Bit	DISEL Bit	Transfer Counter <sup>*1,*2</sup>	
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 CHNE bit = 1” is omitted.

### 19.4.1 Transfer Data Read Skip Function

Vector address read and transfer data read can be skipped by the setting of the RRS bit in DTCCR.

When a DTC startup request is generated, the current DTC vector number is always compared with the DTC vector number in the previous startup process. When these vector numbers match and the RRS bit is set to 1, DTC data transfer is performed without reading the vector address and transfer data. However, when the previous transfer was chain transfer, the vector address and transfer data are always 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 data is always read regardless of the value of RRS bit. Figure 19.13 shows an example of transfer data read skip.

To update the vector table and transfer data, set the RRS bit to 0, update the vector table and transfer data, and then set the RRS bit to 1. When the RRS bit is set to 0, the retained vector number is discarded and the vector table and transfer data that are updated in the following startup process are read.

### 19.4.2 Transfer Data Write-Back Skip Function

When the SM[1:0] bits in MRA or the DM[1:0] bits in MRB are set to “address fixed”, a part of transfer data is not written back. This function is performed independently of the setting of short-address mode or full-address mode. Table 19.4 lists transfer data write-back skip conditions and applicable registers.

The CRA and CRB registers are always written back independently of the setting of short-address mode or full-address mode. Furthermore, in full-address mode, write-back of the MRA and MRB registers are always skipped.

**Table 19.4 Transfer Data Write-Back Skip Conditions and Applicable Registers**

SM[1:0] Bits in MRA		DM[1:0] Bits in MRB		SAR Register	DAR Register
b3	b2	b3	b2		
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		

### 19.4.3 Normal Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single startup source. 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 19.5 lists register functions in normal transfer mode, and Figure 19.5 shows the memory map of normal transfer mode.

**Table 19.5 Register Functions in Normal Transfer Mode**

Register	Description	Value Written Back by Writing Transfer Data
SAR	Transfer source address	Increment/decrement/fix <sup>*1</sup>
DAR	Transfer destination address	Increment/decrement/fix <sup>*1</sup>
CRA	Transfer counter A	CRA - 1
CRB	Transfer counter B	Not updated

Note 1. Write-back operation is skipped in address-fixed mode.

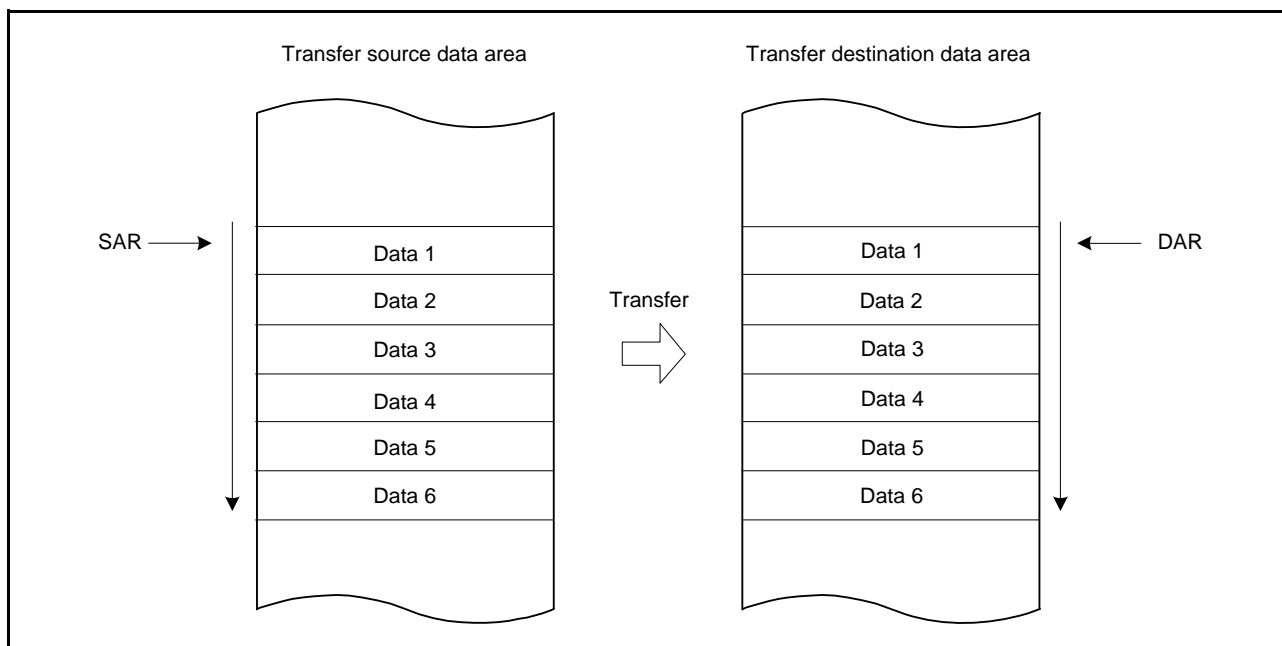


Figure 19.5 Memory Map of Normal Transfer Mode

### 19.4.4 Repeat Transfer Mode

This mode allows 1-byte, 1-word, or 1-longword data transfer on a single startup source.

Specify either transfer source or transfer destination for the repeat area by the DTS bit in MRB. 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 CRAH. Thus the transfer counter does not become 00h, which inhibits generation of interrupt request to the CPU when the DISEL bit in MRB is set to 0 (an interrupt request to the CPU is generated when specified data transfer is completed).

Table 19.6 lists the register functions in repeat transfer mode, and Figure 19.6 shows the memory map of repeat transfer mode.

Table 19.6 Register Functions in Repeat Transfer Mode

Register	Description	Value Written Back by Writing Transfer Data	
		When CRAL is not 1	When CRAL is 1
SAR	Transfer source address	Increment/decrement/fixe*1	(When the DTS bit in MRB is 0) Increment/decrement/fixe*1 (When the DTS bit in MRB is 1) SAR register initial value
DAR	Transfer destination address	Increment/decrement/fixe*1	(When the DTS bit in MRB is 0) DAR register initial value (When the DTS bit in MRB is 1) Increment/decrement/fixe*1
CRAH	Retains transfer counter	CRAH	CRAH
CRAL	Transfer counter A	CRAL - 1	CRAH
CRB	Transfer counter B	Not updated	Not updated

Note 1. Write-back is skipped in address-fixed mode.

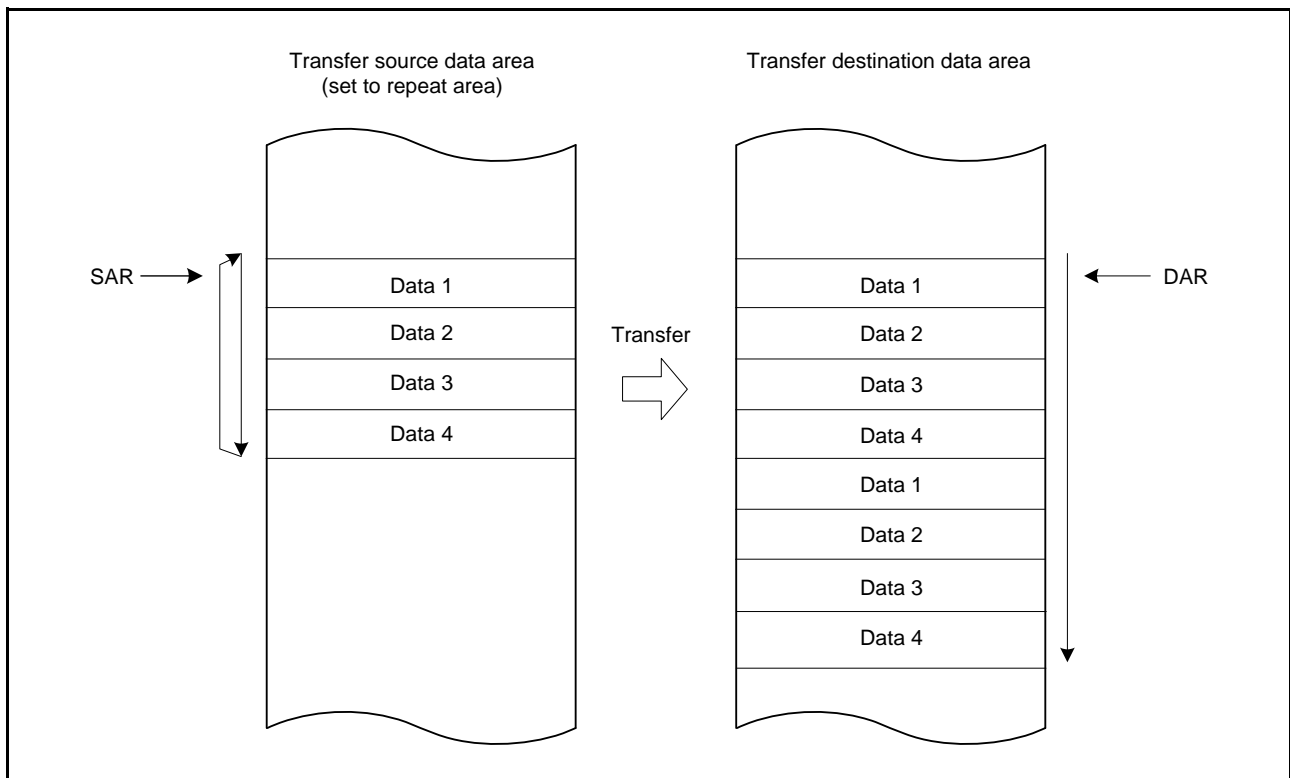


Figure 19.6 Memory Map of Repeat Transfer Mode (Transfer Source: Repeat Area)



### 19.4.5 Block Transfer Mode

This mode allows single-block data transfer on a single startup source.

Specify either transfer source or transfer destination for the block area by the DTS bit in MRB. The block size can be set to 1 to 256 bytes (or 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 (SAR when the DTS bit = 1 or DAR when the DTS bit = 0) specified in the block area are restored. The other address register is incremented or decremented continuously or remains unchanged.

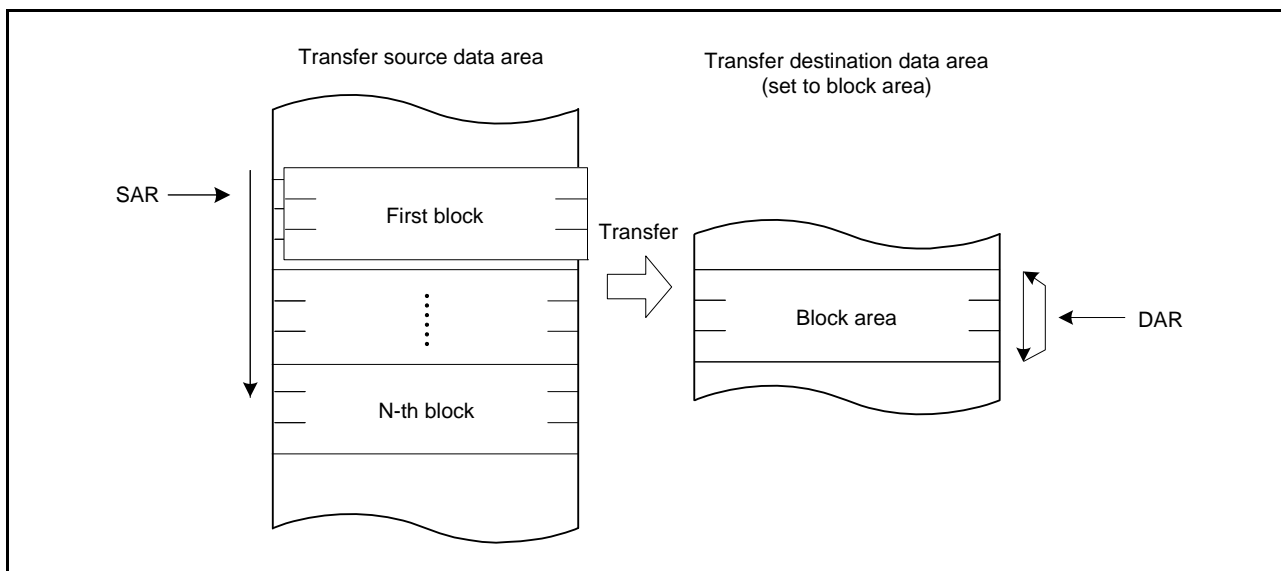
The transfer count (block count) can be set to 1 to 65536. This mode enables an interrupt request to the CPU to be generated at the end of specified-count block transfer.

Table 19.7 lists register functions in block transfer mode, and Figure 19.7 shows the memory map of block transfer mode.

**Table 19.7 Register Functions in Block Transfer Mode**

Register	Description	Value Written Back by Writing Transfer Data
SAR	Transfer source address	(When DTS bit in MRB is 0) Increment/decrement/fix*1 (When DTS bit in MRB is 1) SAR register initial value
DAR	Transfer destination address	(When DTS bit in MRB is 0) DAR register initial value (When DTS bit in MRB is 1) Increment/decrement/fix*1
CRAH	Retains block size	CRAH
CRAL	Block size counter	CRAH
CRB	Block transfer counter	CRB - 1

Note 1. Write-back is skipped in address-fixed mode.



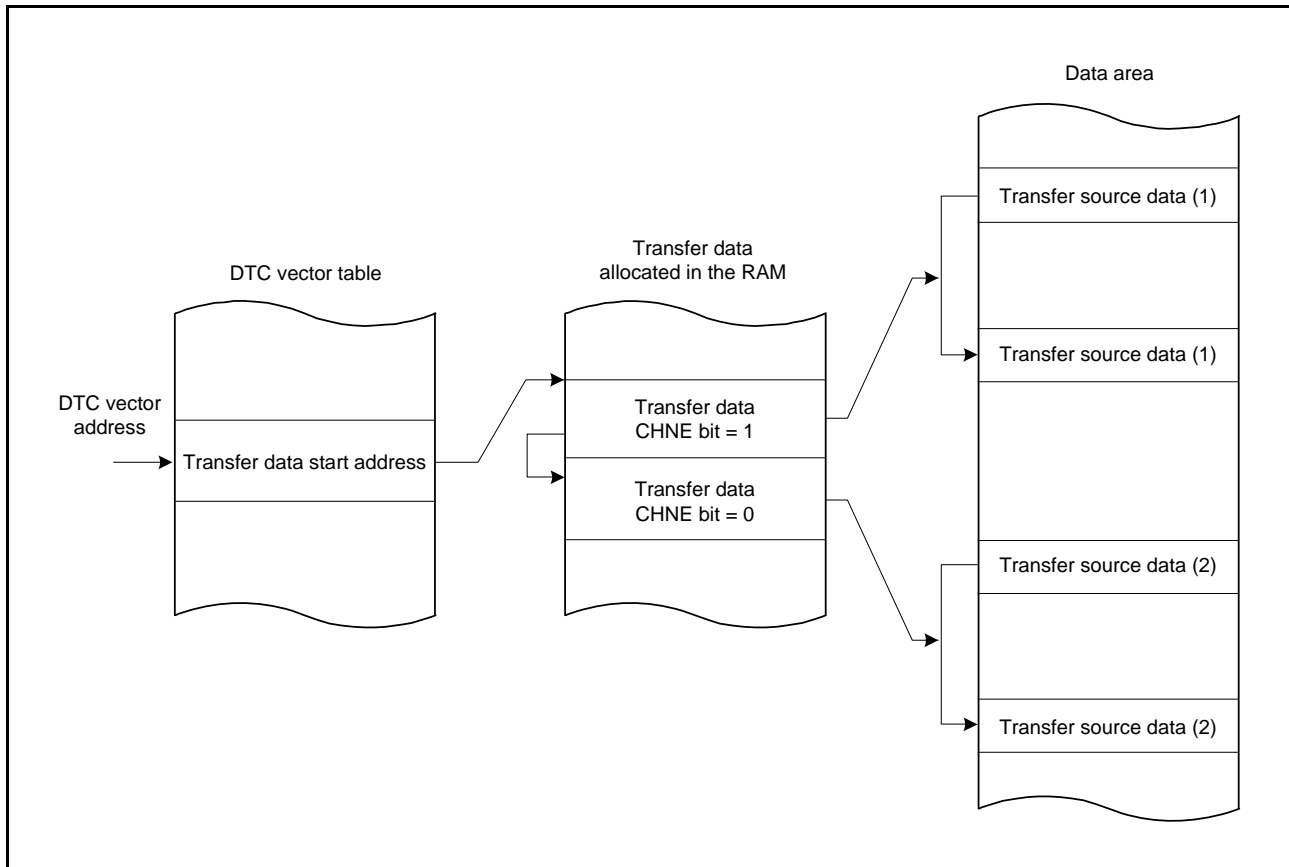
**Figure 19.7 Memory Map of Block Transfer Mode (Transfer Destination: Block Area)**

### 19.4.6 Chain Transfer

Setting the CHNE bit in MRB to 1 allows chain transfer to be performed continuously on a single startup source.

If the CHNE bit in MRB is set to 1, and the CHNS bit in MRB is set to 0, an interrupt request to the CPU is not generated by completion of specified number of rounds of transfer or by setting the DISEL bit in MRB to 1 (an interrupt request to the CPU is generated each time DTC data transfer is performed), and data transfer has no effect on the interrupt status flag that has started up the transfer.

The SAR, DAR, CRA, CRB, MRA, and MRB registers can be set independently of each other to define data transfer. Figure 19.8 shows chain transfer operation.



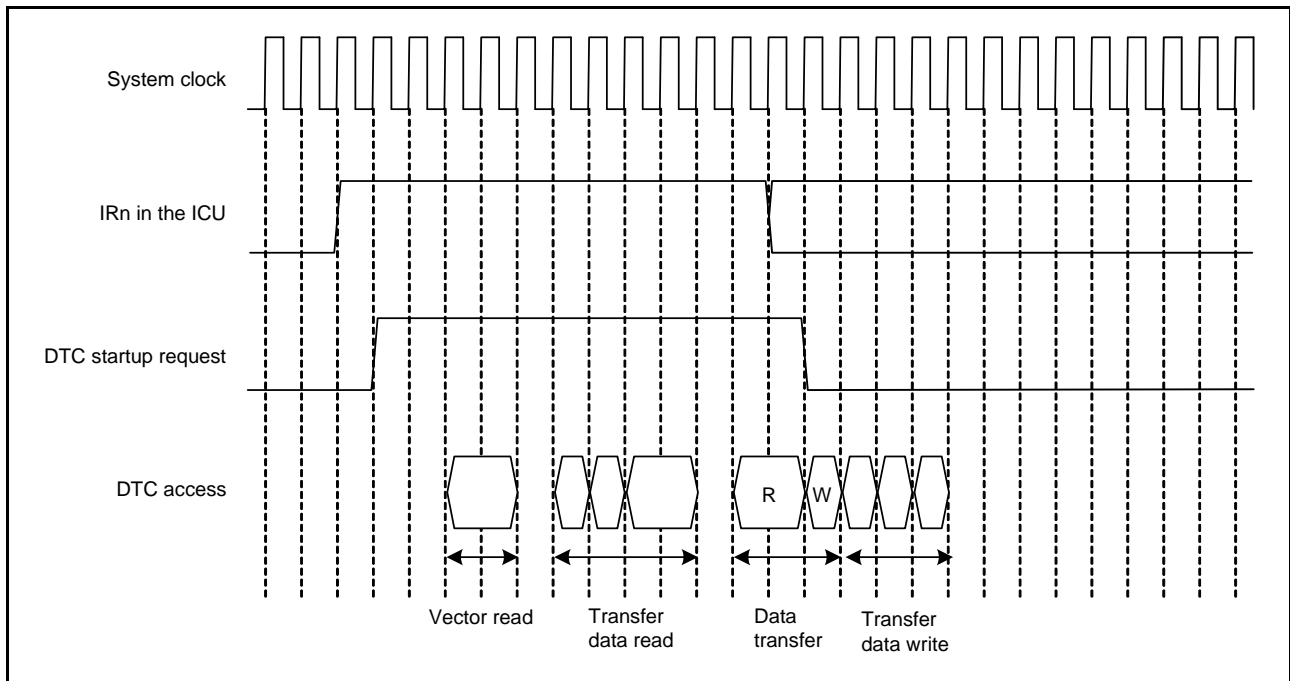
**Figure 19.8 Chain Transfer Operation**

Writing 1 to the CHNE and CHNS bits in MRB enables chain transfer to be performed only after completion of specified data transfer. In repeat transfer mode, chain transfer is performed after completion of specified data transfer.

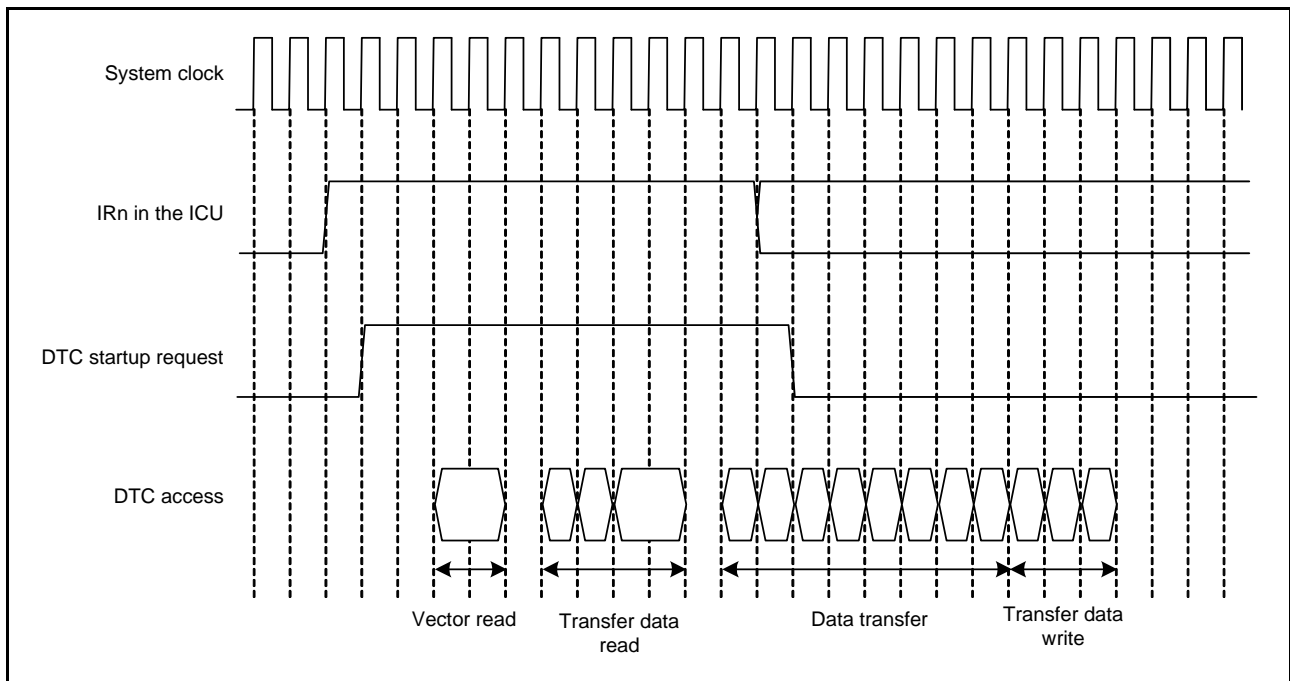
For details on chain transfer conditions, see Table 19.3, Chain Transfer Conditions.

### 19.4.7 Operation Timing

Figure 19.9 to Figure 19.13 show examples of DTC operation timing.



**Figure 19.9 Example (1) of DTC Operation Timing (Short-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)**



**Figure 19.10 Example (2) of DTC Operation Timing (Short-Address Mode, Block Transfer Mode, Block Size = 4)**

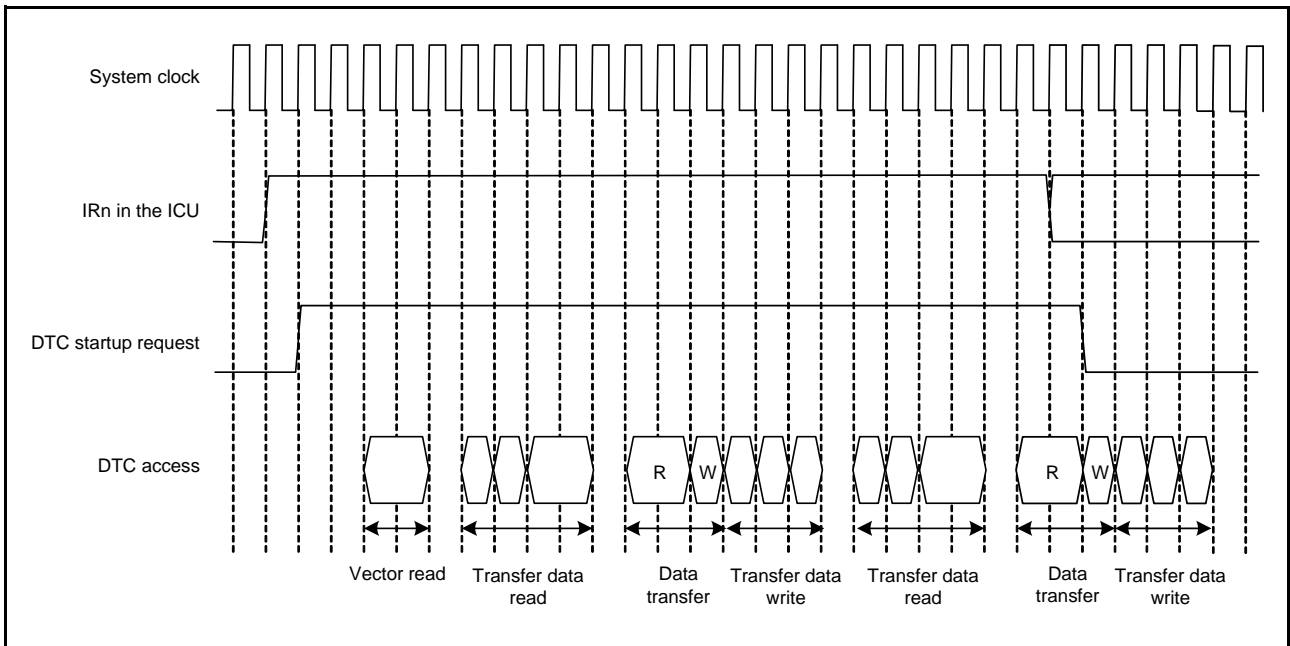


Figure 19.11 Example (3) of DTC Operation Timing (Short-Address Mode, Chain Transfer)

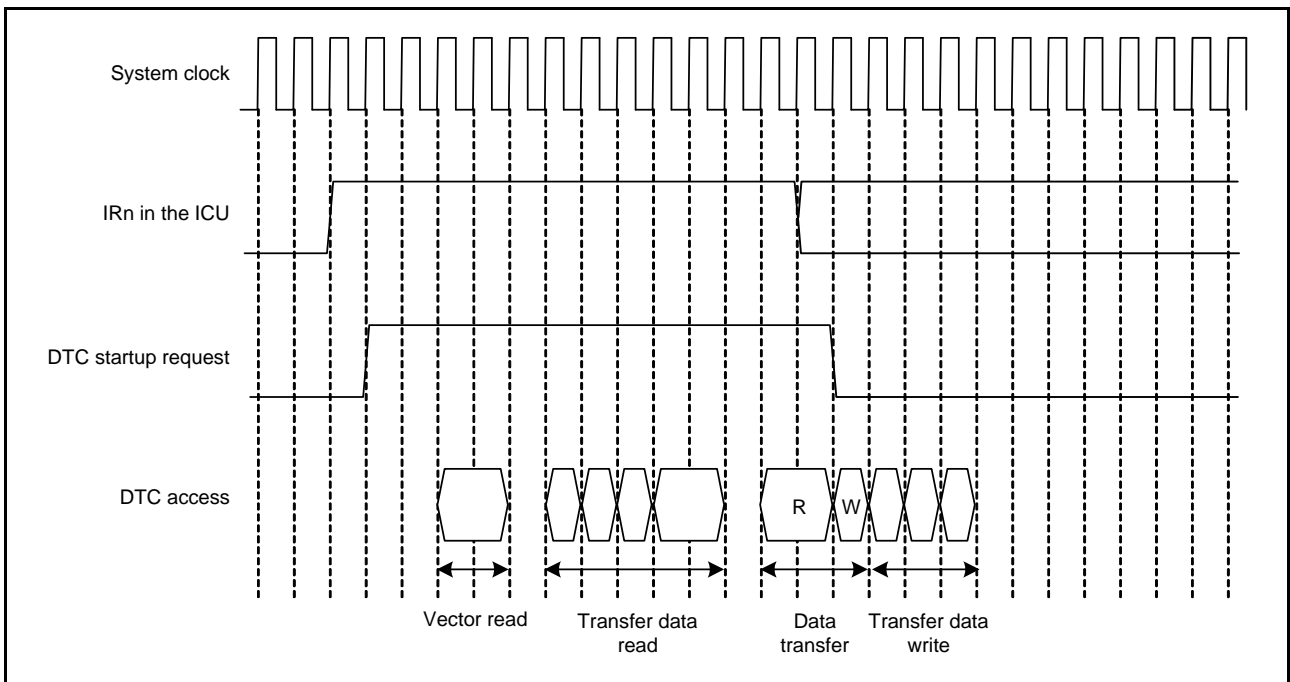
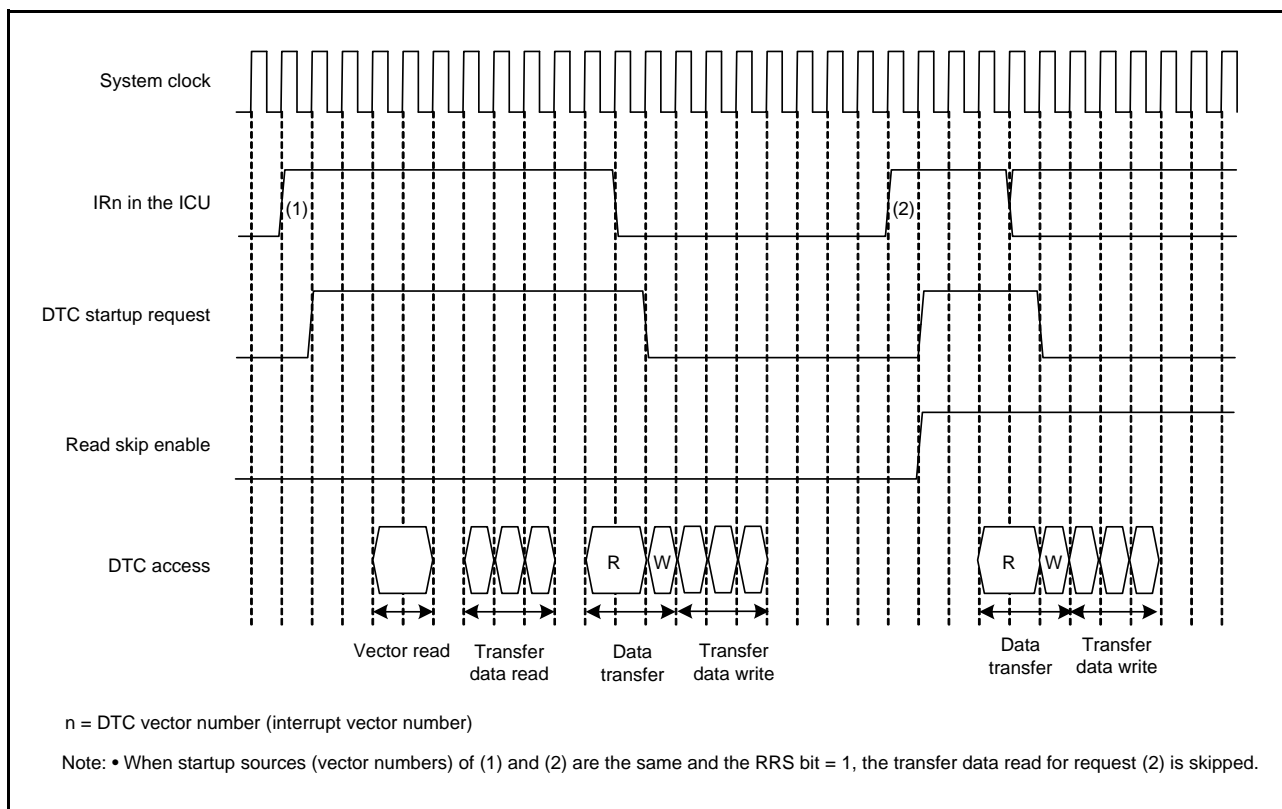


Figure 19.12 Example (4) of DTC Operation Timing (Full-Address Mode, Normal Transfer Mode, Repeat Transfer Mode)



**Figure 19.13 Example of Operation when Transfer Information Skip is Executed (Vector, Transfer Information, and Transfer Destination Data on the RAM, and Transfer Source Data on the Peripheral Module)**

### 19.4.8 Execution Cycles of the DTC

Table 19.8 lists the execution cycles of single data transfer of the DTC.

For the order of the execution states, refer to section 19.4.7, Operation Timing.

**Table 19.8 Execution Cycles of the DTC**

Transfer Mode	Vector Read		Transfer Data Read			Transfer Data Write			Data Transfer		Internal Operation	
									Read	Write		
Normal	$Cv+1$	$0*1$	$4 \times Ci+1*2$	$3 \times Ci+1*3$	$0*1$	$3 \times Ci*4$	$2 \times Ci*5$	$Ci*6$	$Cr+1$	$Cw$	2	$0*1$
Repeat									$Cr+1$	$Cw$		
Block*7									$P \times Cr$	$P \times Cw$		

Note 1. When transfer data 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 mode

Note 5. When SAR or DAR is set to address-fixed mode

Note 6. When SAR and DAR are set to address-fixed mode

Note 7. When the block size is 2 or larger. 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 data storage destination

Ci: Cycles for access to transfer data 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 Data 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, see section 42, RAM, section 43, Flash Memory, section 5, I/O Registers, and section 16.2.6, External Bus.)

### 19.4.9 DTC Bus Mastership Release Timing

The DTC does not release the bus mastership during transfer data read and transfer data write. While transfer data is not read or written, bus arbitration is made according to the priority determined by the bus master arbitrator.

For bus arbitration, see section 16, Buses.

### 19.5 DTC Setting Procedure

Before using the DTC, set the DTC vector base register (DTCVBR).

Figure 19.14 shows the procedure to set the DTC.

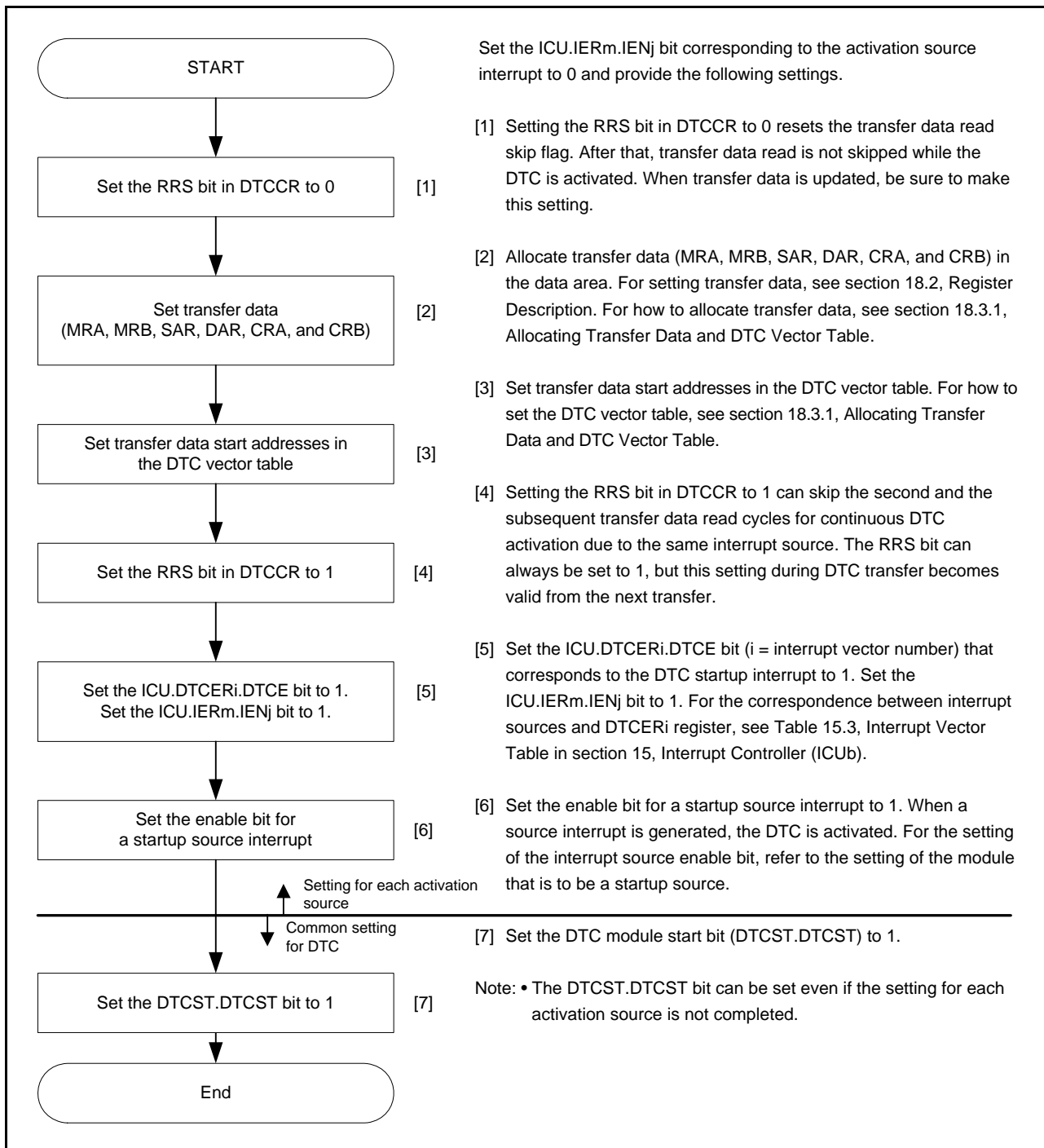


Figure 19.14 Procedure to Set the DTC

## 19.6 Examples of DTC Usage

### 19.6.1 Normal Transfer

As an example of DTC usage, its employment in the transfer of 128 bytes of data by an SCI is described below.

#### (1) Transfer Data Set

In the MRA register, select a fixed source address (MRA.SM[1:0] = 00b), normal transfer mode (MRA.MD[1:0] = 00b), and byte-sized transfer (MRA.SZ[1:0] = 00b). In the MRB register, specify incrementation of the destination address (MRB.DM[1:0] = 10b) and single data transfer by a single interrupt (MRB.CHNE bit = 0 and MRB.DISEL bit = 0). The MRB.DTS bit can be set to any value. Set the address of the RDR register in the SCIm (m = 0 to 12) 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

The address where the transfer-control information for use with the RXI starts is set in the vector table for the DTC.

#### (3) ICU Set and DTC Module Activation

Set the corresponding ICU.DTCERi.DTCE bit to 1 and the ICU.IERi.IENj bit to 1.  
Set the DTCST.DTCST bit to 1.

#### (4) SCI Set

Enable the receive end interrupt (RXI) by setting the SCR.RIE bit in the SCIm 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 activate the DTC. The DTC transfers the received byte from the RDR of the SCIm to RAM, after which the DAR register is incremented and the CRA register is decremented.

#### (6) Interrupt Handling

After 128 rounds of data transfer have been completed and the value in the CRA register becomes 0, an RXI interrupt request is generated for the CPU. Processing for completion is performed in the processing routine for this interrupt.

### 19.6.2 Chain Transfer

As an example of chained transfer by the DTC, its employment in the output of pulses by a PPG is described below.

Chained transfer is used to transfer pulse output data and change the period of the output trigger for the PPG. For the first half of the chained transfer, repeat transfer mode for transfer to the PPGm.NDRH and PPGm.NDRL registers (m = 0, 1) is specified. For the second half, normal transfer mode for transfer to the MTUm.TGR registers (m = 0 to 5) is specified. This is because clearing of the activation source and generation of an interrupt on completion of the specified number of rounds of transfer are restricted to the second half of the chained transfer (transfer while MRB.CHNE = 0).

An example of how to use the compare-match interrupt for an MTUm.TGRA register (for m = 0 to 4) as an activating source for the DTC is provided below.



### (1) First Transfer Data Set

Settings are made for transfer to the PPGm.NDRH and PPGm.NDRL registers. In the MRA register, make the settings to select incrementation of the source address (MRA.SM[1:0] = 10b), transfer in repeated-transfer mode (MRA.MD[1:0] = 01b), and word-sized transfer (MRA.SZ[1:0] = 01b). In the MRB register, make the settings for the destination address fixed (MRB.DM[1:0] = 00b) and for chained transfer (MRB.CHNE = 1 and MRB.CHNS = 0). Set the source side on the repeat area (MRB.DTS = 1). Set the SAR to the first address of the data table, the DAR register to the address of the PPGm.NDRH register, and the CRAH and CRAL registers to the size of the data table. The CRB register can be set to any value.

### (2) Second Transfer Data Set

Settings are made for transfer to the MTUm.TGRA register. In the MRA register, make the settings to select incrementation of the source address (MRA.SM[1:0] = 10b), transfer in normal mode (MRA.MD[1:0] = 00b), and word-sized transfer (MRA.SZ[1:0] = 01b). In the MRB register, make the settings for the destination address fixed (MRB.DM[1:0] = 00b) and for the single data transfer per interrupt (MRB.CHNE = 0, MRB.DISEL = 0). The MRB.DTS bit can be set to any value. Set the SAR register to the first address of the data table, the DAR register to the address of the MTUm.TGRA register, and the CRA register to the size of the data table. The CRB register can be set to any value.

### (3) Transfer Data Assignment

Place the transfer-control information for use in transfer to the MTU immediately after the transfer-control information for use in transfer to the PPGm.NDRH and PPGm.NDRL registers.

### (4) DTC Vector Table

In the DTC vector table, set the address where the transfer-control information for use in transfer to the PPGm.NDRH and PPGm.NDRL registers starts.

### (5) ICU Set and DTC Module Activation

Set the ICU.DTCERi.DTCE bit corresponding to the TGIA interrupt and the ICU.IERi.IENj bit to 1.  
Set the DTCST.DTCST bit to 1.

### (6) MTU Set

In the given MTUm, set the TIOR register so that the TGRA register operates as an output-compare register (with output disabled) and make the TIER setting to enable TGImA interrupt requests.

### (7) PPG Set

Set the default output values in the PPGm.PODRH and PPGm.PODRL registers and the next output values in the PPGm.NDRH and PPGm.NDRL registers. Also, select a compare-match signal of the MTU as the output trigger in the PPGm.PCR (m = 0, 1).

### (8) MTU Activation

Set the MTU.TSTR.CSTj (j = 0 to 4) bits to 1 to start counting operation of the MTUm.TCNT counter.

### (9) DTC Transfer

Every time a compare-match with the MTUm.TGRA register is generated, next output values are transferred to the PPGm.NDRH and PPGm.NDRL registers and the setting for the next output-trigger period is transferred to the MTUm.TGRA register.

### (10) Interrupt Handling

After the specified number of rounds of data transfer has been completed (i.e. when the value in the CRA register of the MTU has become 0), a TGIAM interrupt request is issued for the CPU. Processing for completion is performed in the processing routine for this interrupt.

### 19.6.3 Chain Transfer when Counter = 0

The second data transfer is performed only when the counter = 0. Repeat transfer of a transfer count of 256 or more is enabled by the re-setting for the first data transfer.

The following shows an example of configuring a 128-Kbyte input buffer, where the input buffer is set so that its lower address starts with 0000h. Figure 19.15 shows a chain transfer when the counter = 0.

1. Set normal transfer mode for input data for the first data transfer. Set the following:  
Transfer source address: Fixed, CRA = 0000h (65,536 times), CHNE bit = 1 (chain transfer enabled) in MRB, CHNS bit = 1 (chain transfer is performed only when the transfer counter is 0) in MRB, and DISEL bit = 0 (an interrupt request to the CPU is generated when specified data transfer is completed) in MRB.
2. Prepare the upper 8-bit address 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). For example, when setting the input buffer to 200000h to 21FFFFh, prepare 21h and 20h.
3. For the second data transfer, set repeat transfer mode (source side: repeat area) for re-setting the transfer destination address of the first data transfer. Specify the upper 8 bits of DAR in the first transfer data area for the transfer destination. At this time, set CHNE bit = 0 (chain transfer disabled) in MRB and DISEL bit = 0 (an interrupt request to the CPU is generated when specified data transfer is completed) in MRB. When setting the input buffer mentioned above to 200000h to 21FFFFh, set the transfer counter to 2.
4. The first data transfer is performed by an interrupt 65,536 times. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the upper 8 bits of the transfer source address of the first data transfer to 21h. The transfer counter (lower 16 bits) of the transfer destination address of the first data transfer is 0000h.
5. In succession, the first data transfer is performed by an interrupt 65,536 times specified for the first data transfer. When the transfer counter of the first data transfer becomes 0, the second data transfer starts. Set the upper 8 bits of the transfer source address of the first data transfer to 20h. The transfer counter (lower 16 bits) of the transfer destination address of the first data transfer is 0000h.
6. Steps 4 and 5 above are repeated infinitely. Since the second data transfer is in repeat transfer mode, no interrupt request to the CPU is generated.

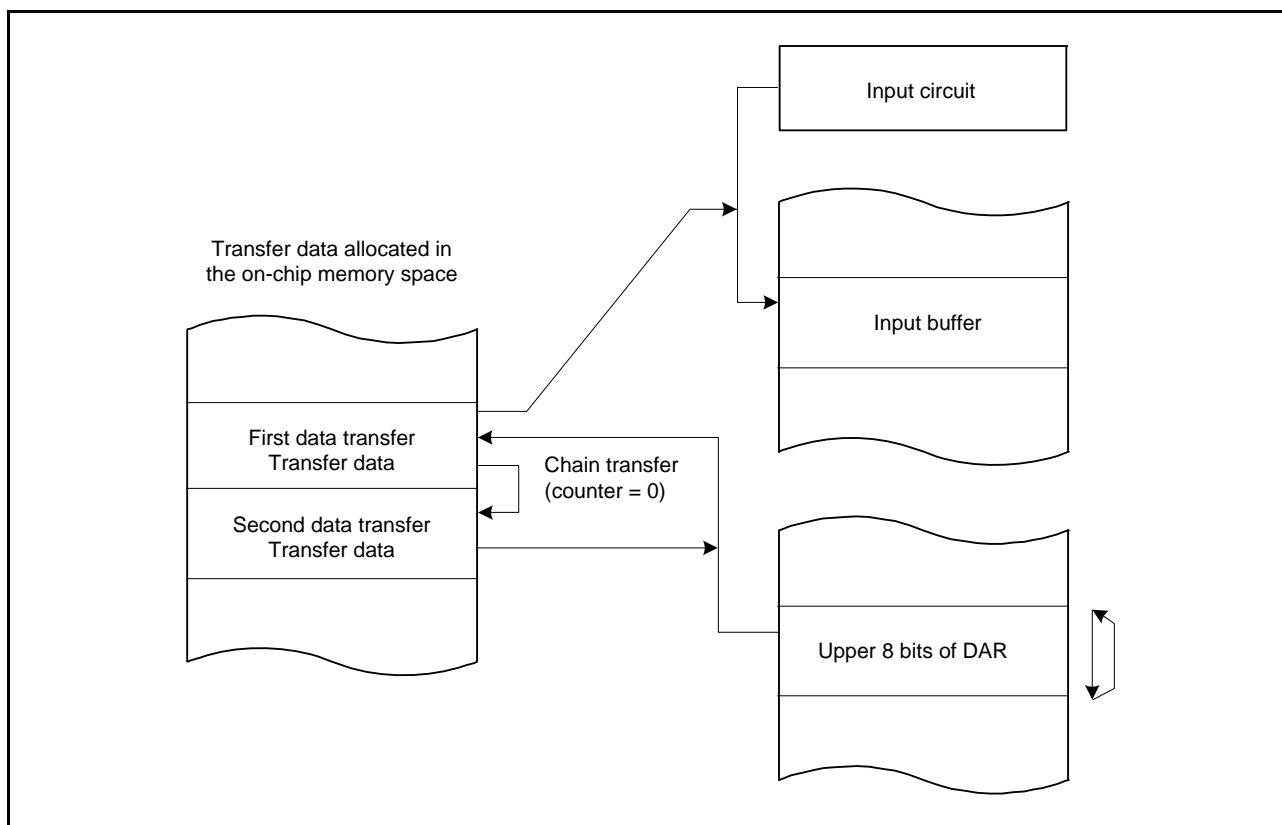


Figure 19.15 Chain Transfer when Counter = 0

### 19.7 Interrupt Source

When the DTC has finished data transfer of specified count or when data transfer with the DISEL bit in MRB set to 1 (an interrupt request to the CPU is generated each time DTC data transfer is performed) has been completed, an interrupt to the CPU is generated by the DTC startup 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.

## 19.8 Low-Power Consumption Function

Before transition to the module-stop state, all-module clock stop mode, software standby mode, or deep software standby mode, clear the DTCST bit in DTCST to 0 (the DTC suspended), and then perform the following.

### (1) Module-Stop Function

Writing 1 to the MSTPA28 bit (transition to the module-stop state) in MSTPCRA enables the module-stop function of the DTC. If DTC transfer is in progress at the time a 1 is written to the MSTPA28 bit, the transition to the module-stop state proceeds after DTC transfer has ended. While the MSTPA28 bit is 1, accessing the DTC registers are prohibited. Writing 0 to the MSTPA28 bit releases the DTC from the module-stop state.

### (2) All-Module Clock-Stop Mode

Make settings in accord with the procedure under section 11.6.2.1, Transition to All-Module Clock Stop Mode, in section 11, Low Power Consumption.

If DTC transfer operations are in progress at the time the WAIT instruction is executed, the transition to all-module clock stop mode follows the completion of DTC transfer.

The DTC is released from the module-stop state by writing 0 to the MSTPCRA.MSTPA28 bit following recovery from all-module clock stop mode.

### (3) Software Standby and Deep Software Standby Modes

Make settings in accord with the procedure under section 11.6.3.1, Transition to Software Standby Mode, or section 11.6.4.1, Transition to Deep Software Standby Mode, in section 11, Low Power Consumption.

If DTC transfer operations are in progress at the time the WAIT instruction is executed, the transition to software standby or deep software standby follows the completion of DTC transfer.

### (4) Notes on Low-Power Consumption Function

For the WAIT instruction and the register setting procedure, section 11.7.6, Timing of Wait Instructions in section 11, Low Power Consumption.

To perform DTC transfer after returning from low-power consumption mode, set the DTCST bit in DTCST to 1 again. To use a request that is generated in all-module clock stop mode and software standby mode as an interrupt request to the CPU but not as a DTC startup request, specify the CPU as the interrupt request destination in accordance with the description in section 15.5.3, Selecting Interrupt Request Destinations in section 15, Interrupt controller (ICUb), and then execute the WAIT instruction.

## 19.9 Usage Notes

### 19.9.1 Transfer Data Start Address/Source Address/Destination Address

Be sure to set multiples of 4 for the transfer data start addresses in the vector table. Otherwise, such addresses are accessed with their lowest 2 bits regarded as 00b.

### 19.9.2 Allocating Transfer Data

Allocate transfer data in the memory area according to the endian of the area as shown in Figure 19.16.

For example, when writing CRA and CRB setting data with 16 bits in big endian, write the CRA setting data to lower address 0 and the CRB setting data to lower address 2. In little endian, write the CRB setting data to lower address 0 and the CRA setting data to lower address 2. When writing CRA and CRB setting data with 32 bits, place the CRA setting data at the MSB side and the CRB setting data at the LSB side regardless of endian, and then write the data to lower address 0.

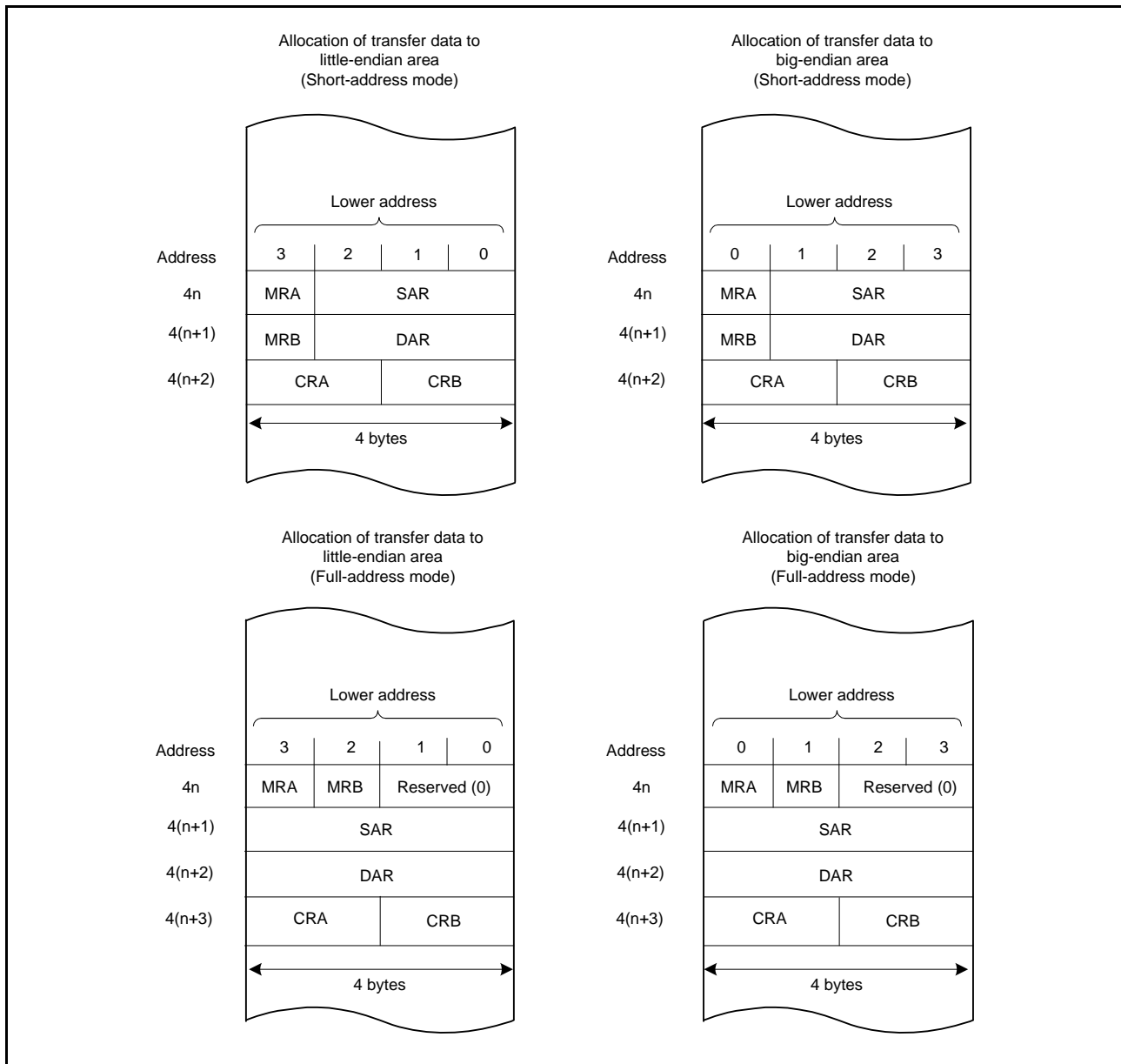


Figure 19.16 Allocation of Transfer Data

### 19.9.3 Setting the DTC Activation Enable Register (ICU.DTCERn) of the Interrupt controller

The DMAC should not be activated by setting the DMAC activation request select register (ICU.DMRSRn (n = number of DMAC channel)) to the same vector number that has been specified by setting the ICU.DTCERn register to 1 (DTC transfer enable). For details on the ICU.DTCERn and ICU.DMRSRn registers (n = number of DMACA channel), refer to section 15, Interrupt controller (ICUb).

## 20. I/O Ports

### 20.1 Overview

The I/O ports of the RX630 Group function as a programmable I/O port, an I/O pin of a peripheral module, an input pin for an interrupt, or a bus control pin.

Each pin is 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 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 driving ability control register (DSCR) that selects the driving ability, and the port mode register (PMR) that specifies the pin function of each port.

For details on the PMR register, see section 21, Multi-Function Pin Controller (MPC).

The configuration of the I/O ports differs depending on the package. Table 20.1 lists the specifications of I/O ports, and Table 20.2 lists the port functions.

**Table 20.1 Specifications of I/O Ports**

Port	Package		Package		Package		Package				
	177 or 176 Pins	Number of Pin	145 or 144 Pins	Number of Pin	100 Pins	Number of Pin	80 Pins	Number of Pin			
PORT0	P00 to P03, P05, P07	6	P00 to P03, P05, P07	6	P05, P07	2	P05, P07	2			
PORT1	P10 to P17	8	P12 to P17	6	P12 to P17	6	P12 to P17	6			
PORT2	P20 to P27	8	P20 to P27	8	P20 to P27	8	P20, P21, P26, P27	4			
PORT3	P30 to P37	8	P30 to P37	8	P30 to P37	8	P30 to P32, P34 to P37	7			
PORT4	P40 to P47	8	P40 to P47	8	P40 to P47	8	P40 to P47	8			
PORT5	P50 to P57	8	P50 to P56	7	P50 to P55	6	P54, P55	2			
PORT6	P60 to P67	8	P60 to P67	8	Not provided	0	Not provided	0			
PORT7	P70 to P77	8	P70 to P77	8	Not provided	0	Not provided	0			
PORT8	P80 to P87	8	P80 to P83, P86, P87	6	Not provided	0	Not provided	0			
PORT9	P90 to P97	8	P90 to P93	4	Not provided	0	Not provided	0			
PORTA	PA0 to PA7	8	PA0 to PA7	8	PA0 to PA7	8	PA0 to PA6	7			
PORTB	PB0 to PB7	8	PB0 to PB7	8	PB0 to PB7	8	PB0 to PB7	8			
PORTC	PC0 to PC7	8	PC0 to PC7	8	PC0 to PC7	8	PC2 to PC7	6			
PORTD	PD0 to PD7	8	PD0 to PD7	8	PD0 to PD7	8	PD0 to PD2	3			
PORTE	PE0 to PE7	8	PE0 to PE7	8	PE0 to PE7	8	PE0 to PE5	6			
PORTF	PF0 to PF5	6	PF5	1	Not provided	0	Not provided	0			
PORTG	PG0 to PG7	8	Not provided	0	Not provided	0	Not provided	0			
PORTH	PH4, PH5	2	Not provided	0	Not provided	0	Not provided	0			
PORTJ	PJ3, PJ5	2	PJ3, PJ5	2	PJ3	1	Not provided	0			
PORTK	PK0 to PK7	8	PK2 to PK5	4	Not provided	0	Not provided	0			
PORTL	PL0 to PL4	5	PL0, PL1	2	Not provided	0	Not provided	0			
Total of Pins		149	Total of Pins		118	Total of Pins		79	Total of Pins		59

**Table 20.2 Port Functions**

Port	Pin	Input Pull-up	Open Drain Output	Driving Ability Switching	5-V Tolerant
PORT0	P00 to P02	○	○	○	—
	P03, P05	○	○	Fixed to high driving ability output	—
	P07	○	○	Fixed to high driving ability output	○
PORT1	P10, P11	○	○	Fixed to high driving ability output	—
	P12 to P17	○	○	Fixed to high driving ability output	○
PORT2	P20 to P25	○	○	Fixed to high driving ability output	○
	P26	○	○	Fixed to high driving ability output	—
	P27	○	○	○	—
PORT3	P30 to P34	○	○	Fixed to high driving ability output	○
	P35	—	—	—	—
	P36	○	○	Fixed to normal output	—
	P37	○	○	Fixed to high driving ability output	—
PORT4	P40 to P47	○	○	Fixed to normal output	—
PORT5	P50 to P52, P56, P57	○	○	○	○
	P53	○	○	Fixed to high driving ability output	—
	P54, P55	○	○	Fixed to high driving ability output	○
PORT6	P60 to P66	○	○	○	—
	P67	○	○	○	○
PORT7	P70 to P73	○	○	○	—
	P74 to P77	○	○	○	○
PORT8	P80 to P82	○	○	Fixed to high driving ability output	○
	P83 to P87	○	○	Fixed to high driving ability output	—
PORT9	P90 to P97	○	○	○	—
PORTA	PA0, PA5, PA7	○	○	○	—
	PA1 to PA4, PA6	○	○	○	○
PORTB	PB0 to PB7	○	○	○	○
PORTC	PC0 to PC7	○	○	○	○
PORTD	PD0 to PD7	○	○	○	—
PORTE	PE0 to PE7	○	○	○	—
PORTF	PF0 to PF5	○	○	Fixed to high driving ability output	—
PORTG	PG0, PG1	○	○	○	—
	PG2 to PG7	○	○	Fixed to high driving ability output	—
PORTH	PH4, PH5	○	○	Fixed to high driving ability output	—

**Table 20.2 Port Functions**

Port	Pin	Input Pull-up	Open Drain Output	Driving Ability Switching	5-V Tolerant
PORTJ	PJ3, PJ5	○	○	Fixed to high driving ability output	—
PORTK	PK0 to PK7	○	○	Fixed to high driving ability output	—
PORTL	PL0 to PK4	○	○	Fixed to high driving ability output	—

Specifying input pull-up, open-drain output, switching of driving ability, or 5-V tolerance is available for other signals on pins that also function as general I/O pins.



20.2 I/O Port Configuration

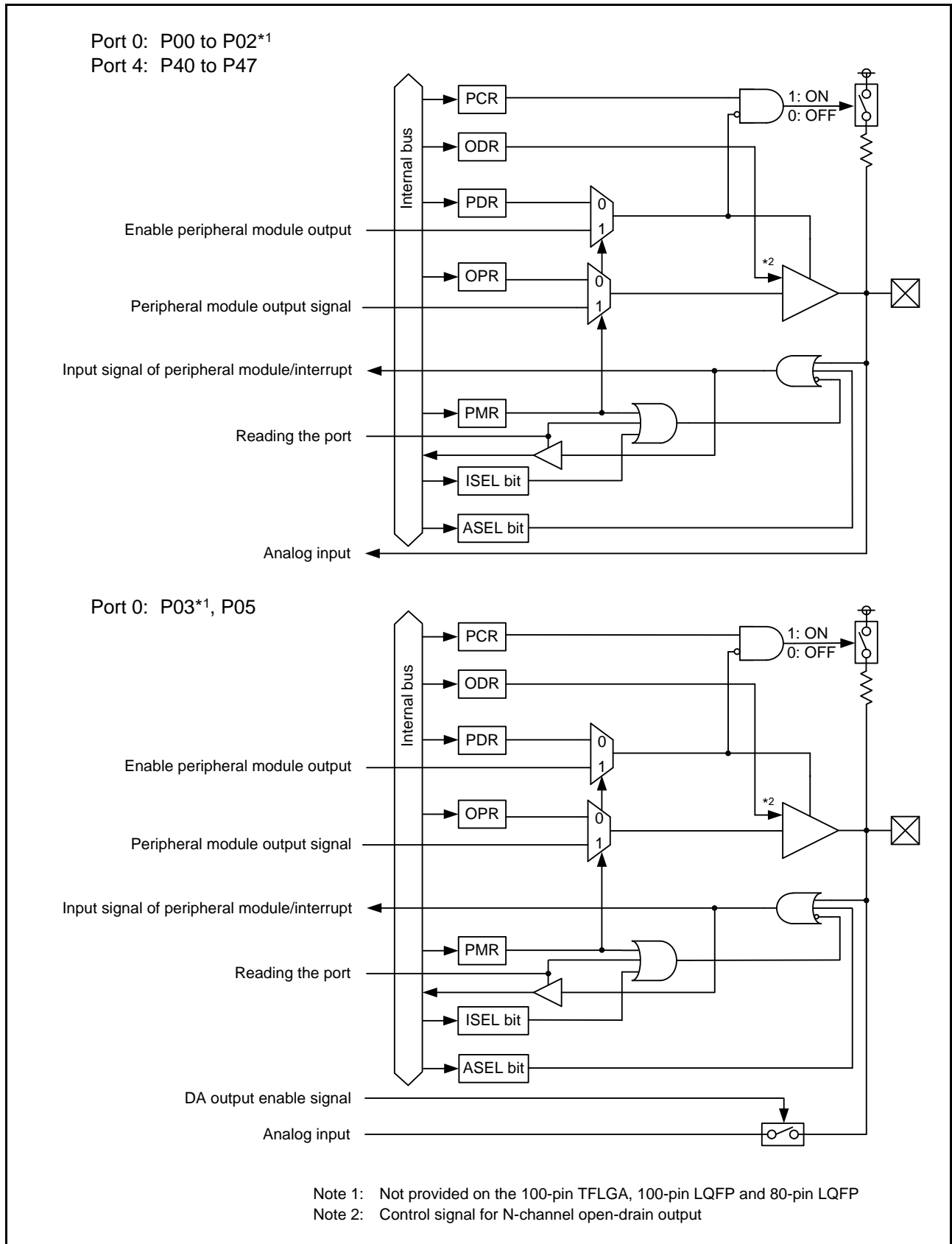
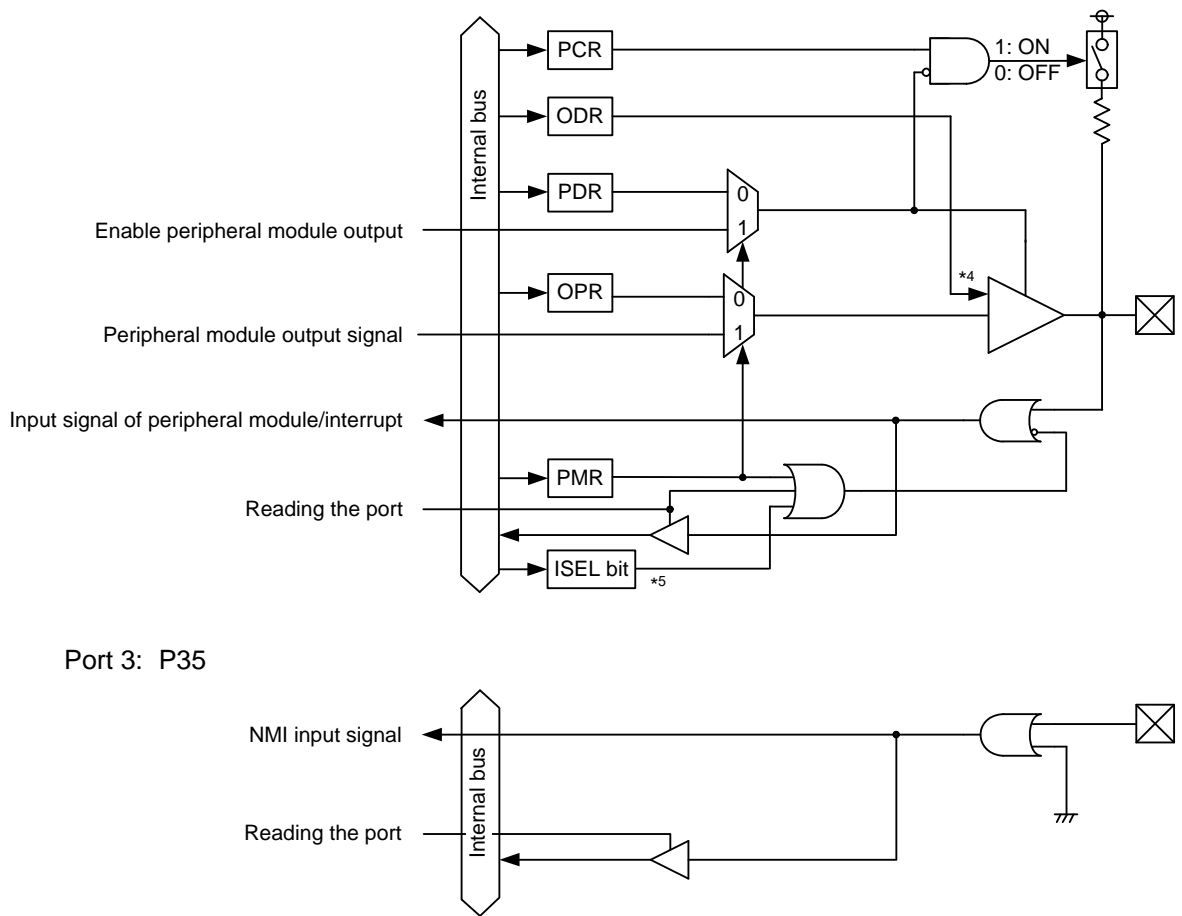


Figure 20.1 I/O Port Configuration (1)

- Port 0: P07
- Port 1: P10\*1 to \*3, P11\*1 to \*3, P12 to P17
- Port 2: P20, P21, P22\*3, P23\*3
- Port 3: P30 to P32, P33\*3, P34, P36, P37
- Port 7: P70\*2, \*3
- Port 8: P80 to P83\*2, \*3, P84\*1 to \*3, P85\*1 to \*3, P86\*2, \*3, P87\*2, \*3
- Port A: PA0 to PA6, PA7\*3
- Port F: PF0 to PF4\*1 to \*3, PF5\*2, \*3
- Port H: PH4\*1 to \*3, PH5\*1 to \*3
- Port J: PJ3\*3, PJ5\*2, \*3
- Port K: PK0\*1 to \*3, PK1\*1 to \*3, PK2 to PK5\*2, \*3, PK6\*1 to \*3, PK7\*1 to \*3
- Port L: PL0\*2, 3, PL1\*2, \*3, PL2 to PL4\*1 to \*3



- Note 1. Not provided on the 145-pin TFLGA and 144-pin LQFP
- Note 2. Not provided on the 100-pin TFLGA and 100-pin LQFP
- Note 3. Not provided on the 80-pin LQFP
- Note 4. Control signal for N-channel open-drain output
- Note 5. An external interrupt function is multiplexed on this pin

Figure 20.2 I/O Port Configuration (2)

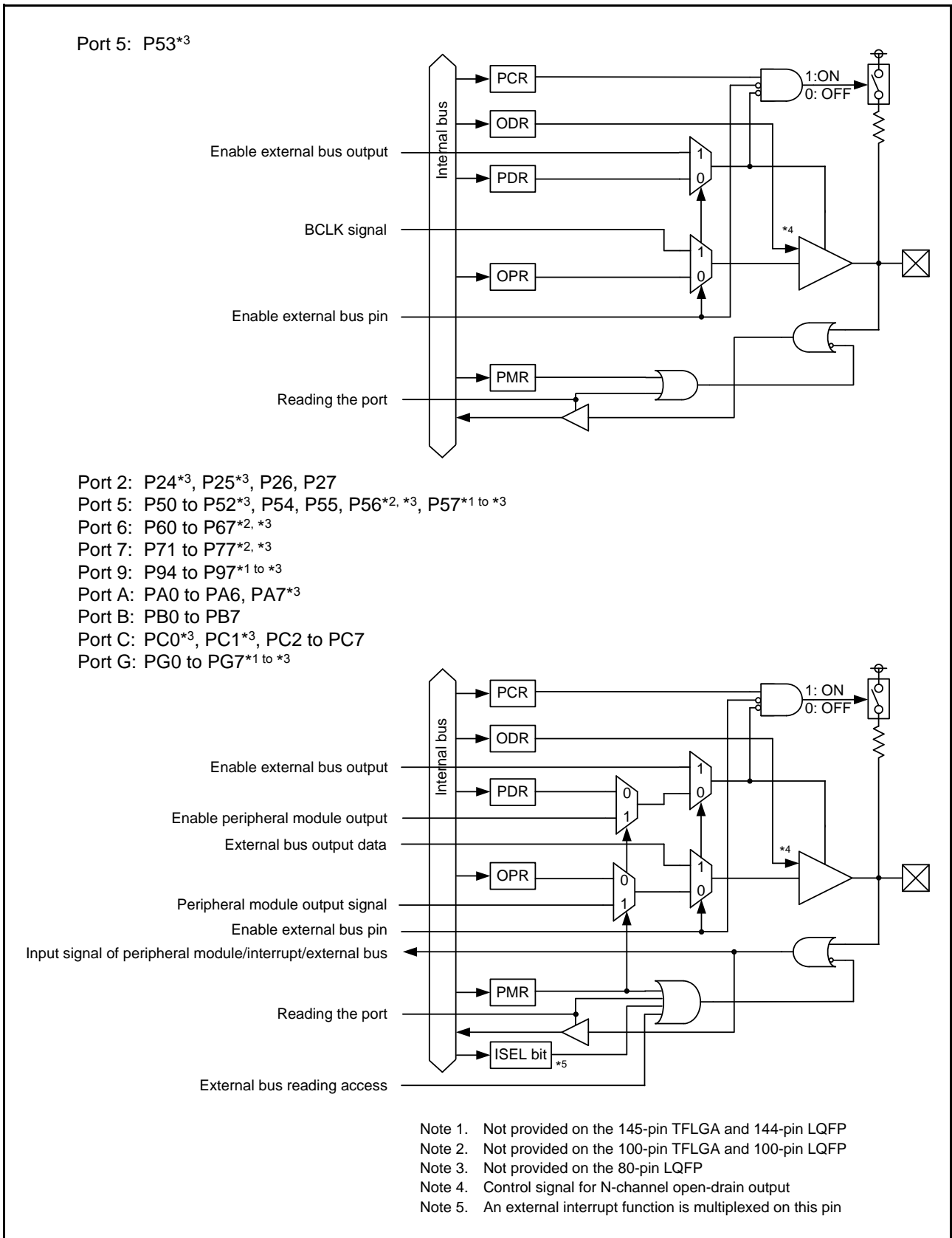


Figure 20.3 I/O Port Configuration (3)

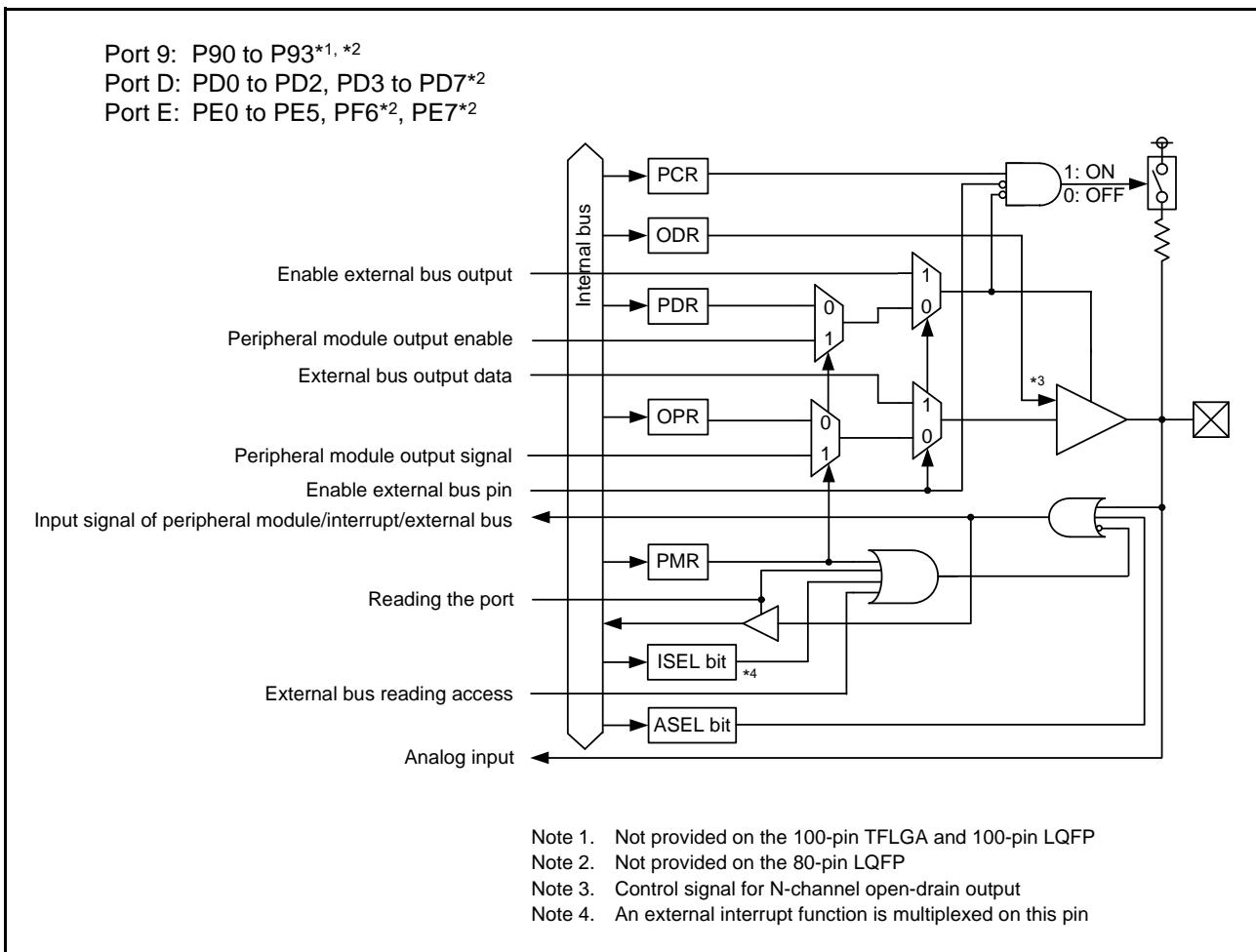


Figure 20.4 I/O Port Configuration (4)

## 20.3 Register Descriptions

### 20.3.1 Port Direction Register (PDR)

Address(es): PORT0.PDR 0008 C000h, PORT1.PDR 0008 C001h, PORT2.PDR 0008 C002h, PORT3.PDR 0008 C003h, PORT4.PDR 0008 C004h, PORT5.PDR 0008 C005h, PORT6.PDR 0008 C006h, PORT7.PDR 0008 C007h, PORT8.PDR 0008 C008h, PORT9.PDR 0008 C009h, PORTA.PDR 0008 C00Ah, PORTB.PDR 0008 C00Bh, PORTC.PDR 0008 C00Ch, PORTD.PDR 0008 C00Dh, PORTE.PDR 0008 C00Eh, PORTF.PDR 0008 C00Fh, PORTG.PDR 0008 C010h, PORTH.PDR 0008 C011h, PORTJ.PDR 0008 C012h, PORTK.PDR 0008 C013h, PORTL.PDR 0008 C014h

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 = 0 to 9, A to H, J to L

PDR is a register which is used to select the input or output direction for individual pins of the corresponding port 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. However, bits that correspond to port m on the 176-pin product but do not exist on a product with fewer pins are reserved. When writing, write 1 (output) to these bits.

The B5 bit in PORT3.PDR is reserved, because the P35 pin is input only. 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.

### 20.3.2 Port Output Data Register (PODR)

Address(es): PORT0.PODR 0008 C020h, PORT1.PODR 0008 C021h, PORT2.PODR 0008 C022h, PORT3.PODR 0008 C023h, PORT4.PODR 0008 C024h, PORT5.PODR 0008 C025h, PORT6.PODR 0008 C026h, PORT7.PODR 0008 C027h, PORT8.PODR 0008 C028h, PORT9.PODR 0008 C029h, PORTA.PODR 0008 C02Ah, PORTB.PODR 0008 C02Bh, PORTC.PODR 0008 C02Ch, PORTD.PODR 0008 C02Dh, PORTE.PODR 0008 C02Eh, PORTF.PODR 0008 C02Fh, PORTG.PODR 0008 C030h, PORTH.PODR 0008 C031h, PORTJ.PODR 0008 C032h, PORTK.PODR 0008 C033h, PORTL.PODR 0008 C034h

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	0: Low output	R/W
b1	B1	Pm1 Output Data Store	1: High output	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 = 0 to 9, A to H, J to L

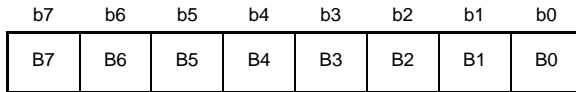
PODR is a register which holds the data to be output from the pins used for general I/O.

Bits that correspond to port m on the 176-pin product but do not exist on a product with fewer pins are reserved. When writing, write 0 (low output) to these bits.

The B5 bit in PORT3.PODR is reserved, because the P35 pin is input only. Data is not output from the corresponding pins even if these bits are set. 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.

### 20.3.3 Port Input Data Register (PIDR)

Address(es): PORT0.PIDR 0008 C040h, PORT1.PIDR 0008 C041h, PORT2.PIDR 0008 C042h, PORT3.PIDR 0008 C043h, PORT4.PIDR 0008 C044h, PORT5.PIDR 0008 C045h, PORT6.PIDR 0008 C046h, PORT7.PIDR 0008 C047h, PORT8.PIDR 0008 C048h, PORT9.PIDR 0008 C049h, PORTA.PIDR 0008 C04Ah, PORTB.PIDR 0008 C04Bh, PORTC.PIDR 0008 C04Ch, PORTD.PIDR 0008 C04Dh, PORTE.PIDR 0008 C04Eh, PORTF.PIDR 0008 C04Fh, PORTG.PIDR 0008 C050h, PORTH.PIDR 0008 C051h, PORTJ.PIDR 0008 C052h, PORTK.PIDR 0008 C053h, PORTL.PIDR 0008 C054h



Value after reset: x x x x x x x x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	B0	Pm0	0: Low input 1: High input	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 = 0 to 9, A to H, J to L

PIDR is a register which reflects individual pin states of the port.

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 P35 bit.

The bit corresponding to a pin that does not exist is reserved. A reserved bit is read as undefined, and cannot be modified.

### 20.3.4 Port Mode Register (PMR)

Address(es): PORT0.PMR 0008 C060h, PORT1.PMR 0008 C061h, PORT2.PMR 0008 C062h, PORT3.PMR 0008 C063h, PORT4.PMR 0008 C064h, PORT5.PMR 0008 C065h, PORT6.PMR 0008 C066h, PORT7.PMR 0008 C067h, PORT8.PMR 0008 C068h, PORT9.PMR 0008 C069h, PORTA.PMR 0008 C06Ah, PORTB.PMR 0008 C06Bh, PORTC.PMR 0008 C06Ch, PORTD.PMR 0008 C06Dh, PORTE.PMR 0008 C06Eh, PORTF.PMR 0008 C06Fh, PORTG.PMR 0008 C070h, PORTH.PMR 0008 C071h, PORTJ.PMR 0008 C072h, PORTK.PMR 0008 C073h, PORTL.PMR 0008 C074h

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: Uses the pin as a general I/O pin. 1: Uses the pin as an 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 = 0 to 9, A to H, J to L

Each bit of PORTm.PMR corresponds to each pin of port m; pin function can be specified in 1-bit units. However, bits that correspond to port m on the 176-pin product but do not exist on a product with fewer pins are reserved. When writing, write 0 (general I/O port) 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.



### 20.3.5 Open Drain Control Register 0 (ODR0)

Address(es): PORT0.ODR0 0008 C080h, PORT1.ODR0 0008 C082h, PORT2.ODR0 0008 C084h, PORT3.ODR0 0008 C086h, PORT4.ODR0 0008 C088h, PORT5.ODR0 0008 C08Ah, PORT6.ODR0 0008 C08Ch, PORT7.ODR0 0008 C08Eh, PORT8.ODR0 0008 C090h, PORT9.ODR0 0008 C092h, PORTA.ODR0 0008 C094h, PORTB.ODR0 0008 C096h, PORTC.ODR0 0008 C098h, PORTD.ODR0 0008 C09Ah, PORTE.ODR0 0008 C09Ch, PORTF.ODR0 0008 C09Eh, PORTG.ODR0 0008 C0A0h, PORTJ.ODR0 0008 C0A4h, PORTK.ODR0 0008 C0A6h, PORTL.ODR0 0008 C0A8h

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	Pm0 Output Type Select	For pins other than the port PE1 pins	R/W
b1	—	Reserved	Odd bit Even bit	R/W
b2	B2	Pm1 Output Type Select	x 0: CMOS output	R/W
b3	—	Reserved	x 1: N-channel open-drain output (b1, b3, b5, b7: Reserved)	R/W
b4	B4	Pm2 Output Type Select	For port PE1 pins	R/W
b5	—	Reserved	b3 b2	R/W
b6	B6	Pm3 Output Type Select	0 0: CMOS output	R/W
b7	—	Reserved	0 1: N-channel open-drain output 1 0: P-channel open-drain output 1 1: Setting prohibited	R/W

m = 0 to 9, A to G, J to L

In the registers other than PORTE.ODR0, the odd bits (b1, b3, b5, and b7) are reserved. Bits that correspond to port m on the 176-pin product but do not exist on a product with fewer pins are reserved. When writing, write 0 (CMOS output) to these bits.

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.

### 20.3.6 Open Drain Control Register 1 (ODR1)

Address(es): PORT0.ODR1 0008 C081h, PORT1.ODR1 0008 C083h, PORT2.ODR1 0008 C085h, PORT3.ODR1 0008 C087h, PORT4.ODR1 0008 C089h, PORT5.ODR1 0008 C08Bh, PORT6.ODR1 0008 C08Dh, PORT7.ODR1 0008 C08Fh, PORT8.ODR1 0008 C091h, PORT9.ODR1 0008 C093h, PORTA.ODR1 0008 C095h, PORTB.ODR1 0008 C097h, PORTC.ODR1 0008 C099h, PORTD.ODR1 0008 C09Bh, PORTE.ODR1 0008 C09Dh, PORTF.ODR1 0008 C09Fh, PORTG.ODR1 0008 C0A1h, PORTH.ODR1 0008 C0A3h, PORTJ.ODR1 0008 C0A5h, PORTK.ODR1 0008 C0A7h, PORTL.ODR1 0008 C0A9h

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 output	R/W
b1	—	Reserved		R/W
b2	B2	Pm5 Output Type Select		R/W
b3	—	Reserved		R/W
b4	B4	Pm6 Output Type Select		R/W
b5	—	Reserved		R/W
b6	B6	Pm7 Output Type Select		R/W
b7	—	Reserved		R/W

m = 0 to 9, A to H, J to L

The odd bits (B1, B3, B5, and B7) of ODR1 are reserved. Bits that correspond to port m on the 176-pin product but do not exist on a product with fewer pins are reserved. When writing, write 0 (CMOS output) to these bits.

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.

### 20.3.7 Pull-Up Control Register (PCR)

Address(es): PORT0.PCR 0008 C0C0h, PORT1.PCR 0008 C0C1h, PORT2.PCR 0008 C0C2h, PORT3.PCR 0008 C0C3h, PORT4.PCR 0008 C0C4h, PORT5.PCR 0008 C0C5h, PORT6.PCR 0008 C0C6h, PORT7.PCR 0008 C0C7h, PORT8.PCR 0008 C0C8h, PORT9.PCR 0008 C0C9h, PORTA.PCR 0008 C0CAh, PORTB.PCR 0008 C0CBh, PORTC.PCR 0008 C0CCh, PORTD.PCR 0008 C0CDh, PORTE.PCR 0008 C0CEh, PORTF.PCR 0008 C0CFh, PORTG.PCR 0008 C0D0h, PORTH.PCR 0008 C0D1h, PORTJ.PCR 0008 C0D2h, PORTK.PCR 0008 C0D3h, PORTL.PCR 0008 C0D4h

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 = 0 to 9, A to H, J to L

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 set as an external bus pin, 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 PCR.

The pull-up resistor is also disabled in the reset state.

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.

### 20.3.8 Driving Ability Control Register (DSCR)

Address(es): PORT0.DSCR 0008 C0E0h, PORT2.DSCR 0008 C0E2h, PORT5.DSCR 0008 C0E5h, PORT6.DSCR 0008 C0E6h, PORT7.DSCR 0008 C0E7h, PORT9.DSCR 0008 C0E9h, PORTA.DSCR 0008 C0EAh, PORTB.DSCR 0008 C0EBh, PORTC.DSCR 0008 C0ECh, PORTD.DSCR 0008 C0EDh, PORTE.DSCR 0008 C0EEh, PORTG.DSCR 0008 C0F0h

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 Driving Ability Control	0: Normal drive output	R/W
b1	B1	Pm1 Driving Ability Control	1: High-drive output	R/W
b2	B2	Pm2 Driving Ability Control		R/W
b3	B3	Pm3 Driving Ability Control		R/W
b4	B4	Pm4 Driving Ability Control		R/W
b5	B5	Pm5 Driving Ability Control		R/W
b6	B6	Pm6 Driving Ability Control		R/W
b7	B7	Pm7 Driving Ability Control		R/W

m = 0, 2, 5 to 7, 9, A to E, G

The bits corresponding to pins for which the driving ability is fixed are readable and writable but the driving ability cannot be switched.

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.

## 20.4 Handling of Unused Pins

Table 20.3 lists handling of unused pins.

**Table 20.3 Handling of Unused Pins**

Pin	Handling Description
EMLE	Connect to VSS through resistor (pull-down)
MD	(Used as a mode pin)
RES#	Connect to VCC through resistor (pull-up)
USB0_DP	Keep the pin open
USB0_DM	
P35/NMI	Connect to VCC through resistor (pull-up)
EXTAL	Connect to VSS through resistor (pull-down)
XTAL	Keep the pin open
XCIN	Connect to VSS through resistor (pull-down)
XCOU	Keep the pin open
Ports 0 to 9, A to H, J to L	<ul style="list-style-type: none"> <li>• 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</li> <li>• If the direction setting is for output (PORTn.PDR = 1), the pin is released.*1, *2</li> </ul>
VREFH0	Connect to AVCC0
VREFL0	Connect to AVSS0
VREFH	Connect to VCC
VREFL	Connect to VSS

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.

## 20.5 Usage Notes

### 20.5.1 Products with Fewer than 176 Pins

For port m pins that are not present because a product has fewer than 176 pins, the bits of the port direction register (PDR) become reserved bits. When writing to these bits, write 1 (output).

## 21. Multi-Function Pin Controller (MPC)

### 21.1 Overview

The multi-function pin controller (MPC) selects and assigns input/output of peripheral functions and interrupt input signals from multiple ports. The MPC also assigns the port of external bus related signals.

Table 21.1 lists the functions assigned to each multiplexed pin. The symbols ○ and × in the table indicate whether the pin is available or unavailable for the package. Selecting a single function for multiple pins is prohibited.

Table 21.1 Functions Assigned to Each Multiplexed Pin (1/14)

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				177-pin, 176-pin	145-pin, 144-pin	100-Pin	80-Pin
Interrupt		NMI (input)	P35	○	○	○	○
Interrupt	IRQ0	IRQ0-DS (input)	P30	○	○	○	○
		IRQ0 (input)	P10	○	×	×	×
			PD0	○	○	○	○
	IRQ1	IRQ1-DS (input)	P31	○	○	○	○
		IRQ1 (input)	P11	○	×	×	×
			PD1	○	○	○	○
	IRQ2	IRQ2-DS (input)	P32	○	○	○	○
		IRQ2 (input)	P12	○	○	○	○
			PD2	○	○	○	○
	IRQ3	IRQ3-DS (input)	P33	○	○	○	×
		IRQ3 (input)	P13	○	○	○	○
			PD3	○	○	○	×
	IRQ4	IRQ4-DS (input)	PB1	○	○	○	○
		IRQ4 (input)	P14	○	○	○	○
			P34	○	○	○	○
			PD4	○	○	○	×
			PF5	○	○	×	×
	IRQ5	IRQ5-DS (input)	PA4	○	○	○	○
		IRQ5 (input)	P15	○	○	○	○
			PD5	○	○	○	×
			PE5	○	○	○	○
	IRQ6	IRQ6-DS (input)	PA3	○	○	○	○
		IRQ6 (input)	P16	○	○	○	○
			PD6	○	○	○	×
			PE6	○	○	○	×
	IRQ7	IRQ7-DS (input)	PE2	○	○	○	○
		IRQ7 (input)	P17	○	○	○	○
			PD7	○	○	○	×
			PE7	○	○	○	×
	IRQ8	IRQ8-DS (input)	P40	○	○	○	○
		IRQ8 (input)	P00	○	○	×	×
			P20	○	○	○	○
	IRQ9	IRQ9-DS (input)	P41	○	○	○	○
		IRQ9 (input)	P01	○	○	×	×
			P21	○	○	○	○
	IRQ10	IRQ10-DS (input)	P42	○	○	○	○
		IRQ10 (input)	P02	○	○	×	×
			P55	○	○	○	○
	IRQ11	IRQ11-DS (input)	P43	○	○	○	○
		IRQ11 (input)	P03	○	○	×	×
			PA1	○	○	○	○

**Table 21.1 Functions Assigned to Each Multiplexed Pin (2/14)**

Module/Function	Channel	Pin Functions	Allocation Port	Package					
				177-pin, 176-pin	145-pin, 144-pin	100-Pin	80-Pin		
Interrupt	IRQ12	IRQ12-DS (input)	P44	○	○	○	○		
		IRQ12 (input)	PB0	○	○	○	○		
			PC1	○	○	○	×		
	IRQ13	IRQ13-DS (input)	P45	○	○	○	○		
		IRQ13 (input)	P05	○	○	○	○		
			PC6	○	○	○	○		
	IRQ14	IRQ14-DS (input)	P46	○	○	○	○		
		IRQ14 (input)	PC0	○	○	○	×		
			PC7	○	○	○	○		
	IRQ15	IRQ15-DS (input)	P47	○	○	○	○		
		IRQ15 (input)	P07	○	○	○	○		
			P67	○	○	×	×		
	Multi-function timer unit 2	MTU0	MTIOC0A (input/output)	P34	○	○	○	○	
				PB3	P13	○	○	○	○
					P15	○	○	○	○
MTIOC0C (input/output)			PA1	○	○	○	○		
			P32	○	○	○	○		
			PB1	P33	○	○	○	○	
PA3				○	○	○	○		
MTU1			MTIOC1A (input/output)	P20	○	○	○	○	
				PE4	P21	○	○	○	○
		PB5			○	○	○	○	
		MTIOC1B (input/output)	P26	○	○	○	○		
			PB5	P27	○	○	○	○	
				PE5	○	○	○	○	



Table 21.1 Functions Assigned to Each Multiplexed Pin (3/14)

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				177-pin, 176-pin	145-pin, 144-pin	100-Pin	80-Pin
Multi-function timer unit 2	MTU3	MTIOC3A (input/output)	P14	○	○	○	○
			P17	○	○	○	○
			PC1	○	○	○	×
			PC7	○	○	○	○
		MTIOC3B (input/output)	P17	○	○	○	○
			P22	○	○	○	×
			P80	○	○	×	×
			PB7	○	○	○	○
			PC5	○	○	○	○
		MTIOC3C (input/output)	P16	○	○	○	○
			P56	○	○	×	×
			PC0	○	○	○	×
	PC6		○	○	○	○	
	PJ3		○	○	○	×	
	MTIOC3D (input/output)	P16	○	○	○	○	
		P23	○	○	○	×	
		P81	○	○	×	×	
		PB6	○	○	○	○	
		PC4	○	○	○	○	
	MTU4	MTIOC4A (input/output)	P24	○	○	○	×
			P82	○	○	×	×
			PA0	○	○	○	○
			PB3	○	○	○	○
			PE2	○	○	○	○
		MTIOC4B (input/output)	P30	○	○	○	○
			P54	○	○	○	○
			PC2	○	○	○	○
			PD1	○	○	○	○
			PE3	○	○	○	○
		MTIOC4C (input/output)	P25	○	○	○	×
			P83	○	○	×	×
			PB1	○	○	○	○
PE1			○	○	○	○	
PE5			○	○	○	○	
MTIOC4D (input/output)		P31	○	○	○	○	
		P55	○	○	○	○	
		PC3	○	○	○	○	
		PD2	○	○	○	○	
		PE4	○	○	○	○	

Table 21.1 Functions Assigned to Each Multiplexed Pin (4/14)

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				177-pin, 176-pin	145-pin, 144-pin	100-Pin	80-Pin
Multi-function timer unit 2	MTU5	MTIC5U (input)	P12	○	×	×	×
			PA4	○	○	○	○
			PD7	○	○	○	×
		MTIC5V (input)	P11	○	×	×	×
			PA6	○	○	○	○
			PD6	○	○	○	×
		MTIC5W (input)	P10	○	×	×	×
			PB0	○	○	○	○
			PD5	○	○	○	×
	MTU	MTCLKA (input)	P14	○	○	○	○
			P24	○	○	○	×
			PA4	○	○	○	○
			PC6	○	○	○	○
		MTCLKB (input)	P15	○	○	○	○
			P25	○	○	○	×
			PA6	○	○	○	○
			PC7	○	○	○	○
		MTCLKC (input)	P22	○	○	○	×
			PA1	○	○	○	○
			PC4	○	○	○	○
		MTCLKD (input)	P23	○	○	○	×
PA3	○		○	○	○		
PC5	○		○	○	○		
Port output enable 2	POE0	POE0# (input)	PC4	○	○	○	○
			PD7	○	○	○	×
	POE1	POE1# (input)	PB5	○	○	○	○
			PD6	○	○	○	×
	POE2	POE2# (input)	P34	○	○	○	○
			PA6	○	○	○	○
	POE3	POE3# (input)	PD5	○	○	○	×
			P33	○	○	○	×
			PB3	○	○	○	○
	POE8	POE8# (input)	PD4	○	○	○	×
			P17	○	○	○	○
			P30	○	○	○	○
	PE3	PE3	PD3	○	○	○	×
			PE3	○	○	○	○

Table 21.1 Functions Assigned to Each Multiplexed Pin (5/14)

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				177-pin, 176-pin	145-pin, 144-pin	100-Pin	80-Pin
16-bit timer pulse unit	TPU0	TIOCA0 (input/output)	P86	○	○	×	×
			PA0	○	○	○	○
		TIOCB0 (input/output)	P17	○	○	○	○
			PA1	○	○	○	○
		TIOCC0 (input/output)	P32	○	○	○	○
		TIOCD0 (input/output)	P33	○	○	○	×
	PA3		○	○	○	○	
	TPU1	TIOCA1 (input/output)	P56	○	○	×	×
			PA4	○	○	○	○
		TIOCB1 (input/output)	P16	○	○	○	○
	PA5		○	○	○	○	
	TPU2	TIOCA2 (input/output)	P87	○	○	×	×
			PA6	○	○	○	○
		TIOCB2 (input/output)	P15	○	○	○	○
	PA7		○	○	○	×	
	TPU3	TIOCA3 (input/output)	P21	○	○	○	○
			PB0	○	○	○	○
		TIOCB3 (input/output)	P20	○	○	○	○
			PB1	○	○	○	○
		TIOCC3 (input/output)	P22	○	○	○	×
PB2			○	○	○	○	
TPU4	TIOCA4 (input/output)	P23	○	○	○	×	
		PB3	○	○	○	○	
	TIOCB4 (input/output)	P25	○	○	○	×	
PB4		○	○	○	○		
P24		○	○	○	×		
TPU5	TIOCA5 (input/output)	P13	○	○	○	○	
		PB6	○	○	○	○	
	TIOCB5 (input/output)	P14	○	○	○	○	
		PB7	○	○	○	○	

Table 21.1 Functions Assigned to Each Multiplexed Pin (6/14)

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				177-pin, 176-pin	145-pin, 144-pin	100-Pin	80-Pin
16-bit timer pulse unit	TPU (unit 0)	TCLKA (input)	P14	○	○	○	○
			PC2	○	○	○	○
		TCLKB (input)	P15	○	○	○	○
			PA3	○	○	○	○
			PC3	○	○	○	○
			PC0	○	○	○	×
		TCLKC (input)	P16	○	○	○	○
			PB2	○	○	○	○
			PC1	○	○	○	×
		TCLKD (input)	P17	○	○	○	○
	PB3		○	○	○	○	
	PC1		○	○	○	×	
	PC1		○	○	○	×	
	TPU6	TIOCA6 (input/output)	PC6	○	○	×	×
			PC7	○	○	×	×
			PC4	○	○	×	×
			PC5	○	○	×	×
	TPU7	TIOCA7 (input/output)	PD0	○	○	×	×
			PD1	○	○	×	×
	TPU8	TIOCA8 (input/output)	PD2	○	○	×	×
PD3			○	○	×	×	
TPU9	TIOCA9 (input/output)	PE2	○	○	×	×	
		PE3	○	○	×	×	
		PE0	○	○	×	×	
		PE1	○	○	×	×	
TPU10	TIOCA10 (input/output)	PE4	○	○	×	×	
		PE5	○	○	×	×	
TPU11	TIOCA11 (input/output)	PE6	○	○	×	×	
		PE7	○	○	×	×	
TPU (unit 1)	TCLKE (input)	PC4	○	○	×	×	
		PC5	○	○	×	×	
		PD1	○	○	×	×	
		PD3	○	○	×	×	
Programmable pulse generator	PPG0	PO0 (output)	P20	○	○	○	○
		PO1 (output)	P21	○	○	○	○
		PO2 (output)	P22	○	○	○	×
		PO3 (output)	P23	○	○	○	×
		PO4 (output)	P24	○	○	○	×
		PO5 (output)	P25	○	○	○	×
		PO6 (output)	P26	○	○	○	○
		PO7 (output)	P27	○	○	○	○
		PO8 (output)	P30	○	○	○	○
		PO9 (output)	P31	○	○	○	○
		PO10 (output)	P32	○	○	○	○
		PO11 (output)	P33	○	○	○	×
		PO12 (output)	P34	○	○	○	○

Table 21.1 Functions Assigned to Each Multiplexed Pin (7/14)

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				177-pin, 176-pin	145-pin, 144-pin	100-Pin	80-Pin
Programmable pulse generator	PPG0	PO13 (output)	P13	○	○	○	○
			P15	○	○	○	○
		PO14 (output)	P16	○	○	○	○
		PO15 (output)	P14	○	○	○	○
	P17		○	○	○	○	
	PPG1	PO16 (output)	P73	○	○	×	×
			PA0	○	○	○	○
		PO17 (output)	PA1	○	○	○	○
			PC0	○	○	○	×
		PO18 (output)	PA2	○	○	○	○
			PC1	○	○	○	×
			PE1	○	○	○	○
		PO19 (output)	P74	○	○	×	×
			PA3	○	○	○	○
		PO20 (output)	P75	○	○	×	×
			PA4	○	○	○	○
		PO21 (output)	PA5	○	○	○	○
			PC2	○	○	○	○
		PO22 (output)	P76	○	○	×	×
			PA6	○	○	○	○
		PO23 (output)	P77	○	○	×	×
			PA7	○	○	○	×
			PE2	○	○	○	○
		PO24 (output)	PB0	○	○	○	○
			PC3	○	○	○	○
		PO25 (output)	PB1	○	○	○	○
			PC4	○	○	○	○
		PO26 (output)	P80	○	○	×	×
			PB2	○	○	○	○
			PE3	○	○	○	○
		PO27 (output)	P81	○	○	×	×
			PB3	○	○	○	○
PO28 (output)		P82	○	○	×	×	
		PB4	○	○	○	○	
		PE4	○	○	○	○	
PO29 (output)		PB5	○	○	○	○	
	PC5	○	○	○	○		
PO30 (output)	PB6	○	○	○	○		
	PC6	○	○	○	○		
PO31 (output)	PB7	○	○	○	○		
	PC7	○	○	○	○		

Table 21.1 Functions Assigned to Each Multiplexed Pin (8/14)

Module/Function	Channel	Pin Functions	Allocation Port	Package						
				177-pin, 176-pin	145-pin, 144-pin	100-Pin	80-Pin			
8-bit timer	TMR0	TMO0 (output)	P22	○	○	○	×			
			PB3	○	○	○	○			
		TMCi0 (input)	P01	○	○	×	×			
			P21	○	○	○	○			
			PB1	○	○	○	○			
		TMRi0 (input)	P00	○	○	×	×			
	P20		○	○	○	○				
	PA4		○	○	○	○				
	TMR1	TMO1 (output)	P17	○	○	○	○			
			P26	○	○	○	○			
		TMCi1 (input)	P02	○	○	×	×			
			P12	○	○	○	○			
			P54	○	○	○	○			
		TMRi1 (input)	P24	○	○	○	×			
	PB5		○	○	○	○				
	TMR2	TMO2 (output)	P16	○	○	○	○			
			PC7	○	○	○	○			
		TMCi2 (input)	P15	○	○	○	○			
			P31	○	○	○	○			
			PC6	○	○	○	○			
		TMRi2 (input)	P14	○	○	○	○			
	PC5		○	○	○	○				
	TMR3	TMO3 (output)	P13	○	○	○	○			
			P32	○	○	○	○			
P55			○	○	○	○				
TMCi3 (input)		P11	○	×	×	×				
		P27	○	○	○	○				
		P34	○	○	○	○				
TMRi3 (input)		PA6	○	○	○	○				
		P10	○	×	×	×				
		P30	○	○	○	○				
			P33	○	○	○	×			
			Serial communications interface	SCI0	RXD0 (input)/ SMISO0 (input/output)/ SSCL0 (input/output)	P21	○	○	○	×
						P33	○	○	○	×
TXD0 (output)/ SMOSI0 (input/output)/ SSDA0 (input/output)	P20	○			○	○	×			
	P32	○	○	○	×					
SCK0 (input/output)			P22	○	○	○	×			
			P34	○	○	○	×			
CTS0# (input)/ RTS0# (output)/ SS0# (input)			P23	○	○	○	×			
			PJ3	○	○	○	×			

Table 21.1 Functions Assigned to Each Multiplexed Pin (9/14)

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				177-pin, 176-pin	145-pin, 144-pin	100-Pin	80-Pin
Serial communications interface	SCI1	RXD1 (input)/ SMISO1 (input/output)/ SSCL1 (input/output)	P15	○	○	○	○
			P30	○	○	○	○
			PF2	○	×	×	×
		TXD1 (output)/ SMOSI1 (input/output)/ SSDA1 (input/output)	P16	○	○	○	○
			P26	○	○	○	○
			PF0	○	×	×	×
		SCK1 (input/output)	P17	○	○	○	○
			P27	○	○	○	○
			PF1	○	×	×	×
		CTS1# (input)/ RTS1# (output)/ SS1# (input)	P14	○	○	○	○
			P31	○	○	○	○
	SCI2	RXD2 (input)/ SMISO2 (input/output)/ SSCL2 (input/output)	P12	○	○	○	×
			P52	○	○	○	×
		TXD2 (output)/ SMOSI2 (input/output)/ SSDA2 (input/output)	P13	○	○	○	×
			P50	○	○	○	×
		SCK2 (input/output)	P11	○	×	×	×
			P51	○	○	○	×
	CTS2# (input)/ RTS2# (output)/ SS2# (input)	P54	○	○	○	×	
	SCI3	RXD3 (input)/ SMISO3 (input/output)/ SSCL3 (input/output)	P16	○	○	○	×
			P25	○	○	○	×
		TXD3 (output)/ SMOSI3 (input/output)/ SSDA3 (input/output)	P17	○	○	○	×
			P23	○	○	○	×
SCK3 (input/output)		P15	○	○	○	×	
		P24	○	○	○	×	
CTS3# (input)/ RTS3# (output)/ SS3# (input)		P26	○	○	○	×	
SCI4	RXD4 (input)/ SMISO4 (input/output)/ SSCL4 (input/output)	PB0	○	○	×	×	
		PK4	○	○	×	×	
	TXD4 (output)/ SMOSI4 (input/output)/ SSDA4 (input/output)	PB1	○	○	×	×	
		PK5	○	○	×	×	
	SCK4 (input/output)	P70	○	○	×	×	
		PB3	○	○	×	×	
	CTS4# (input)/ RTS4# (output)/ SS4# (input)	PB2	○	○	×	×	
		PE6	○	○	×	×	

**Table 21.1 Functions Assigned to Each Multiplexed Pin (10/14)**

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				177-pin, 176-pin	145-pin, 144-pin	100-Pin	80-Pin
Serial communications interface	SCI5	RXD5 (input)/ SMISO5 (input/output)/ SSCL5 (input/output)	PA2	○	○	○	○
			PA3	○	○	○	○
			PC2	○	○	○	○
		TXD5 (output)/ SMOSI5 (input/output)/ SSDA5 (input/output)	PA4	○	○	○	○
			PC3	○	○	○	○
			PC1	○	○	○	×
		SCK5 (input/output)	PA1	○	○	○	○
			PC1	○	○	○	×
			PC4	○	○	○	○
		CTS5# (input)/ RTS5# (output)/ SS5# (input)	PA6	○	○	○	○
			PC0	○	○	○	×
	SCI6	RXD6 (input)/ SMISO6 (input/output)/ SSCL6 (input/output)	P01	○	○	×	×
			P33	○	○	○	×
			PB0	○	○	○	○
		TXD6 (output)/ SMOSI6 (input/output)/ SSDA6 (input/output)	P00	○	○	×	×
			P32	○	○	○	○
			PB1	○	○	○	○
		SCK6 (input/output)	P02	○	○	×	×
			P34	○	○	○	○
			PB3	○	○	○	○
		CTS6# (input)/ RTS6# (output)/ SS6# (input)	PB2	○	○	○	○
			PJ3	○	○	○	×
SCI7	RXD7 (input)/ SMISO7 (input/output)/ SSCL7 (input/output)	P92	○	○	×	×	
		P90	○	○	×	×	
SCI8	RXD8 (input)/ SMISO8 (input/output)/ SSCL8 (input/output)	P91	○	○	×	×	
		P93	○	○	×	×	
SCI8	TXD8 (output)/ SMOSI8 (input/output)/ SSDA8 (input/output)	PC6	○	○	○	○	
		PC7	○	○	○	○	
		PC5	○	○	○	○	
		PC4	○	○	○	○	
SCI8	SCK8 (input/output)	PC6	○	○	○	○	
		PC7	○	○	○	○	
		PC5	○	○	○	○	
		PC4	○	○	○	○	
SCI8	CTS8# (input)/ RTS8# (output)/ SS8# (input)	PC6	○	○	○	○	
		PC7	○	○	○	○	
		PC5	○	○	○	○	
		PC4	○	○	○	○	



Table 21.1 Functions Assigned to Each Multiplexed Pin (11/14)

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				177-pin, 176-pin	145-pin, 144-pin	100-Pin	80-Pin
Serial communications interface	SCI9	RXD9 (input)/ SMISO9 (input/output)/ SSCL9 (input/output)	PB6	○	○	○	○
			PK3	○	○	×	×
		TXD9 (output)/ SMOSI9 (input/output)/ SSDA9 (input/output)	PB7	○	○	○	○
			PK2	○	○	×	×
		SCK9 (input/output)	P60	○	○	×	×
			PB5	○	○	○	○
		CTS9# (input)/ RTS9# (output)/ SS9# (input)	P61	○	○	×	×
			PB4	○	○	○	○
	SCI10	RXD10 (input)/ SMISO10 (input/output)/ SSCL10 (input/output)	P81	○	○	×	×
		TXD10 (output)/ SMOSI10 (input/output)/ SSDA10 (input/output)	P82	○	○	×	×
		SCK10 (input/output)	P80	○	○	×	×
		CTS10# (input)/ RTS10# (output)/ SS10# (input)	P83	○	○	×	×
	SCI11	RXD11 (input)/ SMISO11 (input/output)/ SSCL11 (input/output)	P76	○	○	×	×
		TXD11 (output)/ SMOSI11 (input/output)/ SSDA11 (input/output)	P77	○	○	×	×
		SCK11 (input/output)	P75	○	○	×	×
		CTS11# (input)/ RTS11# (output)/ SS11# (input)	P74	○	○	×	×
	SCI12	RXD12 (input)/ SMISO12 (input/output)/ SSCL12 (input/output)/ RXDX12 (input)	PE2	○	○	○	○
		TXD12 (output)/ SMOSI12 (input/output)/ SSDA12 (input/output)/ TXDX12 (output)/ SIOX12 (input/output)	PE1	○	○	○	○
SCK12 (input/output)		PE0	○	○	○	○	
CTS12# (input)/ RTS12# (output)/ SS12# (input)		PE3	○	○	○	○	
I <sup>2</sup> C bus interface	RIIC0	SCL0[FM+] (input/output)	P12	○	○	○	○
		SDA0[FM+] (input/output)	P13	○	○	○	○
	RIIC1	SCL1 (input/output)	P21	○	○	×	×
		SDA1 (input/output)	P20	○	○	×	×
	RIIC2	SCL2-DS (input/output)	P16	○	○	○	○
		SDA2-DS (input/output)	P17	○	○	○	○
	RIIC3	SCL3 (input/output)	PC0	○	○	×	×
SDA3 (input/output)		PC1	○	○	×	×	

Table 21.1 Functions Assigned to Each Multiplexed Pin (12/14)

Module/Function	Channel	Pin Functions	Allocation Port	Package				
				177-pin, 176-pin	145-pin, 144-pin	100-Pin	80-Pin	
USB 2.0 Function Module		USB0_DPUPE (output)	P14	○	○	○	○	
		USB0_VBUS (input)	P16	○	○	○	○	
CAN module	CAN0	CRX0 (input)*1	P33	○	○	○	×	
			PD2	○	○	○	×	
		CTX0 (output)*1	P32	○	○	○	×	
			PD1	○	○	○	×	
	CAN1	CRX1-DS (input)	P15	○	○	○	○	
		CRX1 (input)	P55	○	○	○	○	
		CTX1 (output)	P14	○	○	○	○	
			P54	○	○	○	○	
	CAN2	CRX2 (input)*2	P67	○	○	×	×	
		CTX2 (output)*2	P66	○	○	×	×	
	Serial peripheral interface	RSPIO	RSPCKA (input/output)	PA5	○	○	○	○
				PB0	○	○	○	○
PC5				○	○	○	○	
MOSIA (input/output)			P16	○	○	○	○	
			PA6	○	○	○	○	
			PC6	○	○	○	○	
MISOA (input/output)			P17	○	○	○	○	
			PA7	○	○	○	×	
			PC7	○	○	○	○	
SSLA0 (input/output)			PA4	○	○	○	○	
			PC4	○	○	○	○	
SSLA1 (output)			PA0	○	○	○	○	
			PC0	○	○	○	×	
SSLA2 (output)			PA1	○	○	○	○	
			PC1	○	○	○	×	
SSLA3 (output)			PA2	○	○	○	○	
			PC2	○	○	○	○	

Table 21.1 Functions Assigned to Each Multiplexed Pin (13/14)

Module/Function	Channel	Pin Functions	Allocation Port	Package				
				177-pin, 176-pin	145-pin, 144-pin	100-Pin	80-Pin	
Serial peripheral interface	RSP11	RSPCKB (input/output)	P27	○	○	○	○	
			PE1	○	○	○	○	
			PE5	○	○	○	○	
		MOSIB (input/output)	P26	○	○	○	○	
			PE2	○	○	○	○	
			PE6	○	○	○	×	
		MISOB (input/output)	P30	○	○	○	○	
			PE3	○	○	○	○	
			PE7	○	○	○	×	
		SSLB0 (input/output)	P31	○	○	○	○	
			PE4	○	○	○	○	
		SSLB1 (output)	P50	○	○	○	×	
			PE0	○	○	○	○	
		SSLB2 (output)	P51	○	○	○	×	
	PE1		○	○	○	○		
	SSLB3 (output)	P52	○	○	○	×		
		PE2	○	○	○	○		
	RSP12	RSP12	RSPCKC (input/output)	PD3	○	○	×	×
			MOSIC (input/output)	PD1	○	○	×	×
			MISOC (input/output)	PD2	○	○	×	×
			SSLC0 (input/output)	PD4	○	○	×	×
SSLC1 (output)			PD5	○	○	×	×	
SSLC2 (output)			PD6	○	○	×	×	
SSLC3 (output)			PD7	○	○	×	×	
IEBus controller	IERXD (input)	P16	○	○	○	○		
		PC2	○	○	○	○		
	IETXD (output)	P17	○	○	○	○		
		PC3	○	○	○	○		
Realtime clock	RTCOUT (output)	P16	○	○	○	○		
		P32	○	○	○	○		
	RTCIC0 (input)*3	P30	○	○	○	○		
	RTCIC1 (input)*3	P31	○	○	○	○		
	RTCIC2 (input)*3	P32	○	○	○	○		
12-bit A/D converter	12-bit A/D converter	AN000 (input)*3	P40	○	○	○	○	
		AN001 (input)*3	P41	○	○	○	○	
		AN002 (input)*3	P42	○	○	○	○	
		AN003 (input)*3	P43	○	○	○	○	
		AN004 (input)*3	P44	○	○	○	○	
		AN005 (input)*3	P45	○	○	○	○	
		AN006 (input)*3	P46	○	○	○	○	
		AN007 (input)*3	P47	○	○	○	○	
		AN008 (input)*3	PD0	○	○	○	○	
		AN009 (input)*3	PD1	○	○	○	○	
		AN010 (input)*3	PD2	○	○	○	○	

**Table 21.1 Functions Assigned to Each Multiplexed Pin (14/14)**

Module/Function	Channel	Pin Functions	Allocation Port	Package			
				177-pin, 176-pin	145-pin, 144-pin	100-Pin	80-Pin
12-bit A/D converter		AN011 (input)*3	PD3	○	○	○	×
		AN012 (input)*3	PD4	○	○	○	×
		AN013 (input)*3	PD5	○	○	○	×
		AN014 (input)*3	P90	○	○	×	×
		AN015 (input)*3	P91	○	○	×	×
		AN016 (input)*3	P92	○	○	×	×
		AN017 (input)*3	P93	○	○	×	×
		AN018 (input)*3	P00	○	○	×	×
		AN019 (input)*3	P01	○	○	×	×
		AN020 (input)*3	P02	○	○	×	×
		ADTRG0# (input)	P07	○	○	○	○
		P16	○	○	○	○	
		P25	○	○	○	×	
10-bit A/D converter		AN0 (input)*3	PE2	○	○	○	○
		AN1 (input)*3	PE3	○	○	○	○
		AN2 (input)*3	PE4	○	○	○	○
		AN3 (input)*3	PE5	○	○	○	○
		AN4 (input)*3	PE6	○	○	○	×
		AN5 (input)*3	PE7	○	○	○	×
		AN6 (input)*3	PD6	○	○	○	×
		AN7 (input)*3	PD7	○	○	○	×
		ANEX0 (output)*3	PE0	○	○	○	○
		ANEX1 (input)*3	PE1	○	○	○	○
		ADTRG# (input)	P13	○	○	○	○
		P17	○	○	○	○	
D/A converter		DA0 (output)*3	P03	○	○	×	×
		DA1 (output)*3	P05	○	○	○	○

Note 1. This pin is not provided in products less than 512 Kbytes of ROM.

Note 2. This pin is not provided in products with less than 1 Mbyte of ROM.

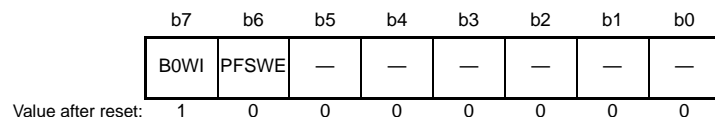
Note 3. To use this pin function, set the corresponding pin as general input (set the PORTm.PDR.Bn and PMR.Bn bits to 0).

## 21.2 Register Descriptions

The registers and bits of unsupported pins, depending on the package, are reserved. The write value to the reserved bits is the value after a reset.

### 21.2.1 Write-Protect Register (PWPR)

Address(es): 0008 C11Fh



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 (PFS Register Write Enable)

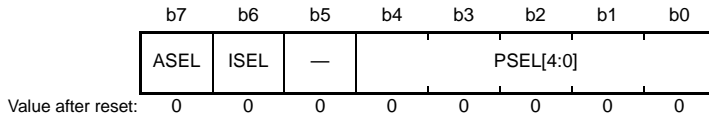
Only when 1 is written to the PFSWE bit, writing to the PmnPFS register is allowed.  
To set the PFSWE bit to 1, write 0 to the B0WI bit before setting the PFSWE bit to 1.

#### B0WI Bit (PFSWE Bit Write Disable)

Only when 0 is written to the B0WI bit, writing to the PFSWE bit is allowed.

21.2.2 P0n Pin Function Control Register (P0nPFS) (n = 0 to 3, 5, 7)

Address(es): P00PFS 0008 C140h, P01PFS 0008 C141h, P02PFS 0008 C142h, P03PFS 0008 C143h, P05PFS 0008 C145h, P07PFS 0008 C147h



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 21.2.	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	R/W
b7	ASEL	Analog Input Function Select	0: Used other than as analog pin. 1: Used as analog pin.	R/W

The port mn pin function select register (PmnPFS) selects the pin function. Bits PSEL[4:0] select the peripheral function which is assigned to bits.

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 ASEL bit is set when a pin is used as an analog pin. When the pin is set as an analog pin by the ASEL bit, select the general I/O port by the port mode register (PORTm.PMR) and specify input by the port direction register (PORTm.PDR). The pin state cannot be read at this point, since the PmnPFS register is protected by the write-protect register (PWPR). Modify the register after releasing the protection.

The ISEL bit to which IRQn is not specified is reserved. The ASEL bit to which analog input/output is not specified is reserved.

**Table 21.2 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LQFP**

PSEL[4:0] Settings	Pin			
	P00	P01	P02	P07
00000b (initial value)	Hi-Z			
00101b	TMR10	TMC10	TMC11	—
01001b	—	—	—	ADTRG0#
01010b	TXD6 SMOSI6 SSDA6	RXD6 SMISO6 SSCL6	SCK6	—

—: Do not specify this value.

**Table 21.3 Register Settings for Input/Output Pin Function in 100-Pin TFLGA, 100-/80-Pin LQFP**

PSEL[4:0] Settings	Pin
	P07
00000b (initial value)	Hi-Z
00101b	—
01001b	ADTRG0#
01010b	—

—: Do not specify this value.

### 21.2.3 P1n Pin Function Control Registers (P1nPFS) (n = 0 to 7)

Address(es): P10PFS 0008 C148h, P11PFS 0008 C149h, P12PFS 0008 C14Ah, P13PFS 0008 C14Bh  
 P14PFS 0008 C14Ch, P15PFS 0008 C14Dh, P16PFS 0008 C14Eh, P17PFS 0008 C14Fh



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 21.4.	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	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

**Table 21.4 Register Settings for Input/Output Pin Function in 177-Pin TFLGA, 176-Pin LFGA, 176-Pin LQFP**

PSEL[4:0] Settings	Pin							
	P10	P11	P12	P13	P14	P15	P16	P17
00000b (initial value)	Hi-Z							
00001b	MTIC5W	MTIC5V	MTIC5U	MTIOC0B	MTIOC3A	MTIOC0B	MTIOC3C	MTIOC3A
00010b	—	—	—	—	MTCLKA	MTCLKB	MTIOC3D	MTIOC3B
00011b	—	—	—	TIOCA5	TIOCB5	TIOCB2	TIOCB1	TIOCB0
00100b	—	—	—	—	TCLKA	TCLKB	TCLKC	TCLKD
00101b	TMRI3	TMCI3	TMCI1	TMO3	TMRI2	TMCI2	TMO2	TMO1
00110b	—	—	—	PO13	PO15	PO13	PO14	PO15
00111b	—	—	—	—	—	—	RTCOUT	POE8#
01001b	—	—	—	ADTRG#	—	—	ADTRG0#	ADTRG#
01010b	—	SCK2	RXD2 SMISO2 SSCL2	TXD2 SMOSI2 SSDA2	—	RXD1 SMISO1 SSCL1	TXD1 SMOSI1 SSDA1	SCK1
01011b	—	—	—	—	CTS1# RTS1# SS1#	SCK3	RXD3 SMISO3 SSCL3	TXD3 SMOSI3 SSDA3
01101b	—	—	—	—	—	—	MOSIA	MISOA
01111b	—	—	SCL0[FM+]	SDA0[FM+]	—	—	SCL2-DS	SDA2-DS
10000b	—	—	—	—	CTX1	CRX1-DS	IERXD	IETXD
10001b	—	—	—	—	USB0_DPUPE	—	USB0_VBUS	—

—: Do not specify this value.

**Table 21.5 Register Settings for Input/Output Pin Function in 145-/100-Pin TFLGA, 144-/100-Pin LQFP**

PSEL[4:0] Settings	Pin					
	P12	P13	P14	P15	P16	P17
00000b (initial value)	Hi-Z					
00001b	—	MTIOC0B	MTIOC3A	MTIOC0B	MTIOC3C	MTIOC3A
00010b	—	—	MTCLKA	MTCLKB	MTIOC3D	MTIOC3B
00011b	—	TIOCA5	TIOCB5	TIOCB2	TIOCB1	TIOCB0
00100b	—	—	TCLKA	TCLKB	TCLKC	TCLKD
00101b	TMC1	TMO3	TMRI2	TMCi2	TMO2	TMO1
00110b	—	PO13	PO15	PO13	PO14	PO15
00111b	—	—	—	—	RTCOUT	POE8#
01001b	—	ADTRG#	—	—	ADTRG0#	ADTRG#
01010b	RXD2 SMISO2 SSCL2	TXD2 SMOSI2 SSDA2	—	RXD1 SMISO1 SSCL1	TXD1 SMOSI1 SSDA1	SCK1
01011b	—	—	CTS1# RTS1# SS1#	SCK3	RXD3 SMISO3 SSCL3	TXD3 SMOSI3 SSDA3
01101b	—	—	—	—	MOSIA	MISOA
01111b	SCL0[FM+]	SDA0[FM+]	—	—	SCL2-DS	SDA2-DS
10000b	—	—	CTX1	CRX1-DS	IERXD	IETXD
10001b	—	—	USB0_DPUPE	—	USB0_VBUS	—

—: Do not specify this value.

**Table 21.6 Register Settings for Input/Output Pin Function in 80-Pin LQFP**

PSEL[4:0] Settings	Pin					
	P12	P13	P14	P15	P16	P17
00000b (initial value)	Hi-Z					
00001b	—	MTIOC0B	MTIOC3A	MTIOC0B	MTIOC3C	MTIOC3A
00010b	—	—	MTCLKA	MTCLKB	MTIOC3D	MTIOC3B
00011b	—	TIOCA5	TIOCB5	TIOCB2	TIOCB1	TIOCB0
00100b	—	—	TCLKA	TCLKB	TCLKC	TCLKD
00101b	TMC1	TMO3	TMRI2	TMCi2	TMO2	TMO1
00110b	—	PO13	PO15	PO13	PO14	PO15
00111b	—	—	—	—	RTCOUT	POE8#
01001b	—	ADTRG#	—	—	ADTRG0#	ADTRG#
01010b	—	—	—	RXD1 SMISO1 SSCL1	TXD1 SMOSI1 SSDA1	SCK1
01011b	—	—	CTS1# RTS1# SS1#	—	—	—
01101b	—	—	—	—	MOSIA	MISOA
01111b	SCL0[FM+]	SDA0[FM+]	—	—	SCL2-DS	SDA2-DS
10000b	—	—	CTX1	CRX1-DS	IERXD	IETXD
10001b	—	—	USB0_DPUPE	—	USB0_VBUS	—

—: Do not specify this value.



21.2.4 P2n Pin Function Control Registers (P2nPFS) (n = 0 to 7)

Address(es): P20PFS 0008 C150h, P21PFS 0008 C151h, P22PFS 0008 C152h, P23PFS 0008 C153h  
 P24PFS 0008 C154h, P25PFS 0008 C155h, P26PFS 0008 C156h, P27PFS 0008 C157h



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 21.7.	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	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Table 21.7 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin CFBGA, 176-/144-Pin LQFP

PSEL[4:0] Settings	Pin							
	P20	P21	P22	P23	P24	P25	P26	P27
00000b (initial value)	Hi-Z							
00001b	MTIOC1A	MTIOC1B	MTIOC3B	MTIOC3D	MTIOC4A	MTIOC4C	MTIOC2A	MTIOC2B
00010b	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB	—	—
00011b	TIOCB3	TIOCA3	TIOCC3	TIOCD3	TIOCB4	TIOCA4	—	—
00101b	TMRI0	TMCIO	TMO0	—	TMRI1	—	TMO1	TMCIO3
00110b	PO0	PO1	PO2	PO3	PO4	PO5	PO6	PO7
01001b	—	—	—	—	—	ADTRG0#	—	—
01010b	TXD0 SMOSI0 SSDA0	RXD0 SMISO0 SSCL0	SCK0	TXD3 SMOSI3 SSDA3	SCK3	RXD3 SMISO3 SSCL3	TXD1 SMOSI1 SSDA1	SCK1
01011b	—	—	—	CTS0# RTS0# SS0#	—	—	CTS3# RTS3# SS3#	—
01101b	—	—	—	—	—	—	MOSIB	RSPCKB
01111b	SDA1	SCL1	—	—	—	—	—	—

—: Do not specify this value.

**Table 21.8 Register Settings for Input/Output Pin Function in 100-Pin TFLGA, 100-Pin LQFP**

PSEL[4:0] Settings	Pin							
	P20	P21	P22	P23	P24	P25	P26	P27
00000b (initial value)	Hi-Z							
00001b	MTIOC1A	MTIOC1B	MTIOC3B	MTIOC3D	MTIOC4A	MTIOC4C	MTIOC2A	MTIOC2B
00010b	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB	—	—
00011b	TIOCB3	TIOCA3	TIOCC3	TIOCD3	TIOCB4	TIOCA4	—	—
00101b	TMRI0	TMCIO	TMO0	—	TMRI1	—	TMO1	TMCIO3
00110b	PO0	PO1	PO2	PO3	PO4	PO5	PO6	PO7
01001b	—	—	—	—	—	ADTRG0#	—	—
01010b	TXD0 SMOSI0 SSDA0	RXD0 SMISO0 SSCL0	SCK0	TXD3 SMOSI3 SSDA3	SCK3	RXD3 SMISO3 SSCL3	TXD1 SMOSI1 SSDA1	SCK1
01011b	—	—	—	CTS0# RTS0# SS0#	—	—	CTS3# RTS3# SS3#	—
01101b	—	—	—	—	—	—	MOSIB	RSPCKB
01111b	—	—	—	—	—	—	—	—

—: Do not specify this value.

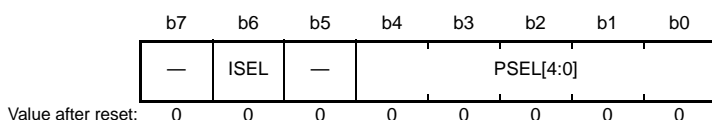
**Table 21.9 Register Settings for Input/Output Pin Function in 80-Pin LQFP**

PSEL[4:0] Settings	Pin			
	P20	P21	P26	P27
00000b (initial value)	Hi-Z			
00001b	MTIOC1A	MTIOC1B	MTIOC2A	MTIOC2B
00010b	—	—	—	—
00011b	TIOCB3	TIOCA3	—	—
00101b	TMRI0	TMCIO	TMO1	TMCIO3
00110b	PO0	PO1	PO6	PO7
01001b	—	—	—	—
01010b	—	—	TXD1 SMOSI1 SSDA1	SCK1
01101b	—	—	MOSIB	RSPCKB
01111b	—	—	—	—

—: Do not specify this value.

### 21.2.5 P3n Pin Function Control Registers (P3nPFS) (n = 0 to 4)

Address(es): P30PFS 0008 C158h, P31PFS 0008 C159h, P32PFS 0008 C15Ah, P33PFS 0008 C15Bh, P34PFS 0008 C15Ch



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 21.10.	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	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

**Table 21.10 Register Settings for Input/Output Pin Function in 177-/145-/100-Pin TFLGA, 176-Pin LFBGA, 176-/144-/100-Pin LQFP**

PSEL[4:0] Settings	Pin				
	P30	P31	P32	P33	P34
00000b (initial value)	Hi-Z				
00001b	MTIOC4B	MTIOC4D	MTIOC0C	MTIOC0D	MTIOC0A
00011b	—	—	TIOCC0	TIOCD0	—
00101b	TMRI3	TMCI2	TMO3	TMRI3	TMCI3
00110b	PO8	PO9	PO10	PO11	PO12
00111b	POE8#	—	RTCOUT	POE3#	POE2#
01010b	RXD1 SMISO1 SSCL1	—	TXD6 SMOSI6 SSDA6	RXD6 SMISO6 SSCL6	SCK6
01011b	—	CTS1# RTS1# SS1#	TXD0 SMOSI0 SSDA0	RXD0 SMISO0 SSCL0	SCK0
01101b	MISOB	SSLB0	—	—	—
10000b	—	—	CTX0*1	CRX0*1	—

—: Do not specify this value.

Note 1. Not supported in 100-pin products with ROM capacities of 512 Kbytes or less.

**Table 21.11 Register Settings for Input/Output Pin Function in 80-Pin LQFP**

PSEL[4:0] Settings	Pin			
	P30	P31	P32	P34
00000b (initial value)	Hi-Z			
00001b	MTIOC4B	MTIOC4D	MTIOC0C	MTIOC0A
00011b	—	—	TIOCC0	—
00101b	TMRI3	TMCI2	TMO3	TMCI3
00110b	PO8	PO9	PO10	PO12
00111b	POE8#	—	RTCOUT	POE2#
01010b	RXD1 SMISO1 SSCL1	—	TXD6 SMOSI6 SSDA6	SCK6
01011b	—	CTS1# RTS1# SS1#	—	—
01101b	MISOB	SSLB0	—	—
10000b	—	—	—	—

—: Do not specify this value.

### 21.2.6 P4n Pin Function Control Registers (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

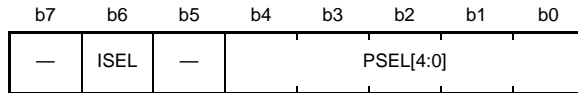
b7	b6	b5	b4	b3	b2	b1	b0
ASEL	ISEL	—	—	—	—	—	—

Value after reset: 0 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	ISEL	Interrupt Input Function Select	0: Not used as IRQn input pin 1: Used as IRQn input pin	R/W
b7	ASEL	Analog Input Function Select	0: Used other than as analog pin. 1: Used as analog pin.	R/W

### 21.2.7 P5n Pin Function Control Registers (P5nPFS) (n = 0 to 2, 4 to 6)

Address(es): P50PFS 0008 C168h, P51PFS 0008 C169h, P52PFS 0008 C16Ah  
 P54PFS 0008 C16Ch, P55PFS 0008 C16Dh, P56PFS 0008 C16Eh



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 21.12.	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	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

**Table 21.12 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LQFP**

PSEL[4:0] Settings	Pin					
	P50	P51	P52	P54	P55	P56
00000b (initial value)	Hi-Z					
00001b	—	—	—	MTIOC4B	MTIOC4D	MTIOC3C
00011b	—	—	—	—	—	TIOCA1
00101b	—	—	—	TMC11	TMO3	—
01010b	TXD2 SMOSI2 SSDA2	SCK2	RXD2 SMISO2 SSCL2	—	—	—
01011b	—	—	—	CTS2# RTS2# SS2#	—	—
01101b	SSLB1	SSLB2	SSLB3	—	—	—
10000b	—	—	—	CTX1	CRX1	—

—: Do not specify this value.

**Table 21.13 Register Settings for Input/Output Pin Function in 100-Pin TFLGA, 100-Pin LQFP**

PSEL[4:0] Settings	Pin				
	P50	P51	P52	P54	P55
00000b (initial value)	Hi-Z				
00001b	—	—	—	MTIOC4B	MTIOC4D
00011b	—	—	—	—	—
00101b	—	—	—	TMC11	TMO3
01010b	TXD2 SMOSI2 SSDA2	SCK2	RXD2 SMISO2 SSCL2	—	—
01011b	—	—	—	CTS2# RTS2# SS2#	—
01101b	SSLB1	SSLB2	SSLB3	—	—
10000b	—	—	—	CTX1	CRX1

—: Do not specify this value.

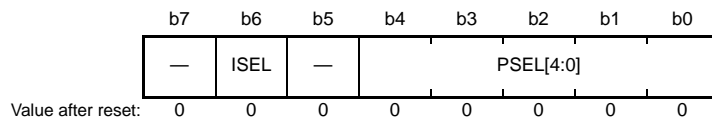
**Table 21.14 Register Settings for Input/Output Pin Function in 80-Pin LQFP**

PSEL[4:0] Settings	Pin	
	P54	P55
00000b (initial value)	Hi-Z	
00001b	MTIOC4B	MTIOC4D
00011b	—	—
00101b	TMCI1	TMO3
01010b	—	—
01011b	CTS2# RTS2# SS2#	—
01101b	—	—
10000b	CTX1	CRX1

—: Do not specify this value.

## 21.2.8 P6n Pin Function Control Registers (P6nPFS) (n = 0, 1, 6 or 7)

Address(es): P60PFS 0008 C170h, P61PFS 0008 C171h, P66PFS 0008 C176h, P67PFS 0008 C177h



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 21.15.	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	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

**Table 21.15 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LQFP**

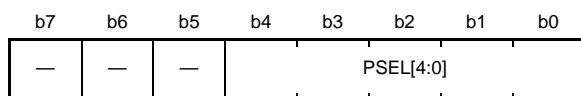
PSEL[4:0] Settings	Pin			
	P60	P61	P66	P67
00000b (initial value)	Hi-Z			
01010b	SCK9	—	—	—
01011b	—	CTS9# RTS9# SS9#	—	—
10000b	—	—	CTX2*1	CRX2*1

—: Do not specify this value.

Note 1. Not supported in products with ROM capacities of 1 Mbyte or less.

### 21.2.9 P7n Pin Function Control Registers (P7nPFS) (n = 0, 3 to 7)

Address(es): P70PFS 0008 C178h, P73PFS 0008 C17Bh, P74PFS 0008 C17Ch, P75PFS 0008 C17Dh, P76PFS 0008 C17Eh, P77PFS 0008 C17Fh



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 21.16.	R/W
b7 to b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

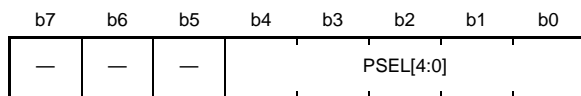
**Table 21.16 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LQFP**

PSEL[4:0] Settings	Pin					
	P70	P73	P74	P75	P76	P77
00000b (initial value)	Hi-Z					
00110b	—	PO16	PO19	PO20	PO22	PO23
01010b	SCK4	—	—	SCK11	RXD11 SMISO11 SSCL11	TXD11 SMOSI11 SSDA11
01011b	—	—	CTS11# RTS11 SS11#	—	—	—

—: Do not specify this value.

21.2.10 P8n Pin Function Control Registers (P8nPFS) (n = 0 to 3, 6, 7)

Address(es): P80PFS 0008 C180h, P81PFS 0008 C181h, P82PFS 0008 C182h, P83PFS 0008 C183h  
 P86PFS 0008 C186h, P87PFS 0008 C187h



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 21.17.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Table 21.17 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LQFP

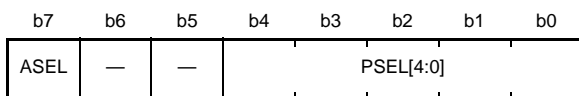
PSEL[4:0] Settings	Pin					
	P80	P81	P82	P83	P86	P87
00000b (initial value)	Hi-Z					
00001b	MTIOC3B	MTIOC3D	MTIOC4A	MTIOC4C	—	—
00011b	—	—	—	—	TIOCA0	TIOCA2
00110b	PO26	PO27	PO28	—	—	—
01010b	SCK10	RXD10 SMISO10 SSCL10	TXD10 SMOSI10 SSDA10	—	—	—
01011b	—	—	—	CTS10# RTS10# SS10#	—	—

—: Do not specify this value.



### 21.2.11 P9n Pin Function Control Registers (P9nPFS) (n = 0 to 3)

Address(es): P90PFS 0008 C188h, P91PFS 0008 C189h, P92PFS 0008 C18Ah, P93PFS 0008 C18Bh



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 21.18.	R/W
b6 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ASEL	Analog Input Function Select	0: Used other than as analog pin. 1: Used as analog pin.	R/W

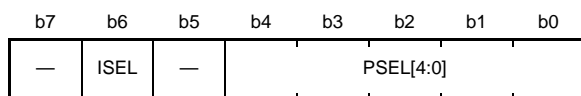
**Table 21.18 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LQFP**

PSEL[4:0] Settings	Pin			
	P90	P91	P92	P93
00000b (initial value)	Hi-Z			
01010b	TXD7 SMOSI7 SSDA7	SCK7	RXD7 SMISO7 SSCL7	—
01011b	—	—	—	CTS7# RTS7# SS7#

—: Do not specify this value.

### 21.2.12 PAn Pin Function Select Registers (PAnPFS) (n = 0 to 7)

Address(es): PA0PFS 0008 C190h, PA1PFS 0008 C191h, PA2PFS 0008 C192h, PA3PFS 0008 C193h  
 PA4PFS 0008 C194h, PA5PFS 0008 C195h, PA6PFS 0008 C196h, PA7PFS 0008 C197h



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 21.19.	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	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

**Table 21.19 Register Settings for Input/Output Pin Function in 177-/145-/100-Pin TFLGA, 176-Pin LFBGA, 176-/144-/100-Pin LQFP**

PSEL[4:0] Settings	Pin							
	PA0	PA1	PA2	PA3	PA4	PA5	PA6	PA7
00000b (initial value)	Hi-Z							
00001b	MTIOC4A	MTIOC0B	—	MTIOC0D	MTIC5U	—	MTIC5V	—
00010b	—	MTCLKC	—	MTCLKD	MTCLKA	—	MTCLKB	—
00011b	TIOCA0	TIOCB0	—	TIOC0D	TIOCA1	TIOCB1	TIOCA2	TIOCB2
00100b	—	—	—	TCLKB	—	—	—	—
00101b	—	—	—	—	TMRI0	—	TMCI3	—
00110b	PO16	PO17	PO18	PO19	PO20	PO21	PO22	PO23
00111b	—	—	—	—	—	—	POE2#	—
01010b	—	SCK5	RXD5 SMISO5 SSCL5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	—	—	—
01011b	—	—	—	—	—	—	CTS5# RTS5# SS5#	—
01101b	SSLA1	SSLA2	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA

—: Do not specify this value.

**Table 21.20 Register Settings for Input/Output Pin Function in 80-Pin LQFP**

PSEL[4:0] Settings	Pin						
	PA0	PA1	PA2	PA3	PA4	PA5	PA6
00000b (initial value)	Hi-Z						
00001b	MTIOC4A	MTIOC0B	—	MTIOC0D	MTIC5U	—	MTIC5V
00010b	—	MTCLKC	—	MTCLKD	MTCLKA	—	MTCLKB
00011b	TIOCA0	TIOCB0	—	TIOCD0	TIOCA1	TIOCB1	TIOCA2
00100b	—	—	—	TCLKB	—	—	—
00101b	—	—	—	—	TMRI0	—	TMCI3
00110b	PO16	PO17	PO18	PO19	PO20	PO21	PO22
00111b	—	—	—	—	—	—	POE2#
01010b	—	SCK5	RXD5 SMISO5 SSCL5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	—	—
01011b	—	—	—	—	—	—	CTS5# RTS5# SS5#
01101b	SSLA1	SSLA2	SSLA3	—	SSLA0	RSPCKA	MOSIA

—: Do not specify this value.

21.2.13 P<sub>B</sub><sub>n</sub> Pin Function Control Registers (P<sub>B</sub><sub>n</sub>PFS) (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



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 21.21.	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 IRQ <sub>n</sub> input pin 1: Used as IRQ <sub>n</sub> input pin	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

**Table 21.21 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LQFP**

PSEL[4:0] Settings	Pin							
	PB0	PB1	PB2	PB3	PB4	PB5	PB6	PB7
00000b (initial value)	Hi-Z							
00001b	MTIC5W	MTIOC0C	—	MTIOC0A	—	MTIOC2A	MTIOC3D	MTIOC3B
00010b	—	MTIOC4C	—	MTIOC4A	—	MTIOC1B	—	—
00011b	TIOCA3	TIOCB3	TIOCC3	TIOCD3	TIOCA4	TIOCB4	TIOCA5	TIOCB5
00100b	—	—	TCLKC	TCLKD	—	—	—	—
00101b	—	TMCIO	—	TMO0	—	TMR1I	—	—
00110b	PO24	PO25	PO26	PO27	PO28	PO29	PO30	PO31
00111b	—	—	—	POE3#	—	POE1#	—	—
01010b	RXD4 SMISO4 SSCL4	TXD4 SMOS14 SSDA4	CTS4# RTS4# SS4#	SCK4	—	SCK9	RXD9 SMISO9 SSCL9	TXD9 SMOS19 SSDA9
01011b	RXD6 SMISO6 SSCL6	TXD6 SMOS16 SSDA6	CTS6# RTS6# SS6#	SCK6	CTS9# RTS9# SS9#	—	—	—
01101b	RSPCKA	—	—	—	—	—	—	—

—: Do not specify this value.

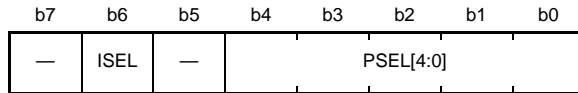
**Table 21.22 Register Settings for Input/Output Pin Function in 100-Pin TFLGA, 100-/80-Pin LQFP**

PSEL[4:0] Settings	Pin							
	PB0	PB1	PB2	PB3	PB4	PB5	PB6	PB7
00000b (initial value)	Hi-Z							
00001b	MTIC5W	MTIOC0C	—	MTIOC0A	—	MTIOC2A	MTIOC3D	MTIOC3B
00010b	—	MTIOC4C	—	MTIOC4A	—	MTIOC1B	—	—
00011b	TIOCA3	TIOCB3	TIOCC3	TIOCD3	TIOCA4	TIOCB4	TIOCA5	TIOCB5
00100b	—	—	TCLKC	TCLKD	—	—	—	—
00101b	—	TMCIO	—	TMO0	—	TMRI1	—	—
00110b	PO24	PO25	PO26	PO27	PO28	PO29	PO30	PO31
00111b	—	—	—	POE3#	—	POE1#	—	—
01010b	—	—	—	—	—	SCK9	RXD9 SMISO9 SSCL9	TXD9 SMOSI9 SSDA9
01011b	RXD6 SMISO6 SSCL6	TXD6 SMOSI6 SSDA6	CTS6# RTS6# SS6#	SCK6	CTS9# RTS9# SS9#	—	—	—
01101b	RSPCKA	—	—	—	—	—	—	—

—: Do not specify this value.

## 21.2.14 PCn Pin Function Control Register (PCnPFS) (n = 0 to 7)

Address(es): PC0PFS 0008 C1A0h, PC1PFS 0008 C1A1h, PC2PFS 0008 C1A2h, PC3PFS 0008 C1A3h  
PC4PFS 0008 C1A4h, PC5PFS 0008 C1A5h, PC6PFS 0008 C1A6h, PC7PFS 0008 C1A7h



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 21.23.	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	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

**Table 21.23 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LQFP**

PSEL[4:0] Settings	Pin							
	PC0	PC1	PC2	PC3	PC4	PC5	PC6	PC7
00000b (initial value)	Hi-Z							
00001b	MTIOC3C	MTIOC3A	MTIOC4B	MTIOC4D	MTIOC3D	MTIOC3B	MTIOC3C	MTIOC3A
00010b	—	—	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB
00011b	TCLKC	TCLKD	TCLKA	TCLKB	TIOCC6	TIOCD6	TIOCA6	TIOCB6
00100b	—	—	—	—	TCLKE	TCLKF	—	—
00101b	—	—	—	—	TMCI1	TMRI2	TMCI2	TMO2
00110b	PO17	PO18	PO21	PO24	PO25	PO29	PO30	PO31
00111b	—	—	—	—	POE0#	—	—	—
01010b	—	SCK5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	SCK5	SCK8	RXD8 SMISO8 SSCL8	TXD8 SMOSI8 SSDA8
01011b	CTS5# RTS5# SS5#	—	—	—	CTS8# RTS8# SS8#	—	—	—
01101b	SSLA1	SSLA2	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA
01111b	SCL3	SDA3	—	—	—	—	—	—
10000b	—	—	IERXD	IETXD	—	—	—	—

—: Do not specify this value.

**Table 21.24 Register Settings for Input/Output Pin Function in 100-Pin TFLGA, 100-Pin LQFP**

PSEL[4:0] Settings	Pin							
	PC0	PC1	PC2	PC3	PC4	PC5	PC6	PC7
00000b (initial value)	Hi-Z							
00001b	MTIOC3C	MTIOC3A	MTIOC4B	MTIOC4D	MTIOC3D	MTIOC3B	MTIOC3C	MTIOC3A
00010b	—	—	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB
00011b	TCLKC	TCLKD	TCLKA	TCLKB	—	—	—	—
00100b	—	—	—	—	—	—	—	—
00101b	—	—	—	—	TMC11	TMRI2	TMC12	TMO2
00110b	PO17	PO18	PO21	PO24	PO25	PO29	PO30	PO31
00111b	—	—	—	—	POE0#	—	—	—
01010b	—	SCK5	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	SCK5	SCK8	RXD8 SMISO8 SSCL8	TXD8 SMOSI8 SSDA8
01011b	CTS5# RTS5# SS5#	—	—	—	CTS8# RTS8# SS8#	—	—	—
01101b	SSLA1	SSLA2	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA
01111b	—	—	—	—	—	—	—	—
10000b	—	—	IERXD	IETXD	—	—	—	—

—: Do not specify this value.

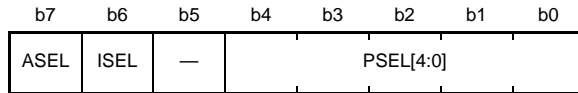
**Table 21.25 Register Settings for Input/Output Pin Function in 80-Pin LQFP**

PSEL[4:0] Settings	Pin					
	PC2	PC3	PC4	PC5	PC6	PC7
00000b (initial value)	Hi-Z					
00001b	MTIOC4B	MTIOC4D	MTIOC3D	MTIOC3B	MTIOC3C	MTIOC3A
00010b	—	—	MTCLKC	MTCLKD	MTCLKA	MTCLKB
00011b	TCLKA	TCLKB	—	—	—	—
00100b	—	—	—	—	—	—
00101b	—	—	TMC11	TMRI2	TMC12	TMO2
00110b	PO21	PO24	PO25	PO29	PO30	PO31
00111b	—	—	POE0#	—	—	—
01010b	RXD5 SMISO5 SSCL5	TXD5 SMOSI5 SSDA5	SCK5	SCK8	RXD8 SMISO8 SSCL8	TXD8 SMOSI8 SSDA8
01011b	—	—	CTS8# RTS8# SS8#	—	—	—
01101b	SSLA3	—	SSLA0	RSPCKA	MOSIA	MISOA
10000b	IERXD	IETXD	—	—	—	—

—: Do not specify this value.

### 21.2.15 PDn Pin Function Control Register (PDnPFS) (n = 0 to 7)

Address(es): PD0PFS 0008 C1A8h, PD1PFS 0008 C1A9h, PD2PFS 0008 C1AAh, PD3PFS 0008 C1ABh  
 PD4PFS 0008 C1ACh, PD5PFS 0008 C1ADh, PD6PFS 0008 C1AEh, PD7PFS 0008 C1AFh



Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b4-b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.26.	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	R/W
b7	ASEL	Analog Input Function Select	0: Used other than as analog pin. 1: Used as analog pin.	R/W

**Table 21.26 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-LQFP**

PSEL[4:0] Settings	Pin							
	PD0	PD1	PD2	PD3	PD4	PD5	PD6	PD7
00000b (initial value)	Hi-Z							
00001b	—	MTIOC4B	MTIOC4D	—	—	MTIC5W	MTIC5V	MTIC5U
00011b	TIOCA7	TIOCB7	TIOCA8	TIOCB8	—	—	—	—
00100b	—	TCLKG	—	TCLKH	—	—	—	—
00111b	—	—	—	POE8#	POE3#	POE2#	POE1#	POE0#
01101b	—	MOSIC	MISOC	RSPCKC	SSLC0	SSLC1	SSLC2	SSLC3
10000b	—	CTX0	CRX0	—	—	—	—	—

—: Do not specify this value.

**Table 21.27 Register Settings for Input/Output Pin Function in 100-Pin TFLGA, 100-Pin LQFP**

PSEL[4:0] Settings	Pin						
	PD1	PD2	PD3	PD4	PD5	PD6	PD7
00000b (initial value)	Hi-Z						
00001b	MTIOC4B	MTIOC4D	—	—	MTIC5W	MTIC5V	MTIC5U
00111b	—	—	POE8#	POE3#	POE2#	POE1#	POE0#
10000b	CTX0*1	CRX0*1	—	—	—	—	—

—: Do not specify this value.

Note 1. Not supported in products with ROM capacities of 512 Kbytes or less.

**Table 21.28 Register Settings for Input/Output Pin Function in 80-Pin LQFP**

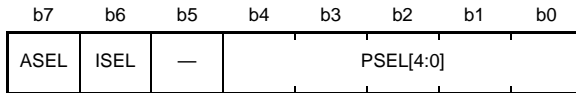
PSEL[4:0] Settings	Pin	
	PD1	PD2
00000b (initial value)	Hi-Z	
00001b	MTIOC4B	MTIOC4D

—: Do not specify this value.



21.2.16 PEn Pin Function Control Register (PEnPFS) (n = 0 to 7)

Address(es): PE0PFS 0008 C1B0h, PE1PFS 0008 C1B1h, PE2PFS 0008 C1B2h, PE3PFS 0008 C1B3h  
 PE4PFS 0008 C1B4h, PE5PFS 0008 C1B5h, PE6PFS 0008 C1B6h, PE7PFS 0008 C1B7h



Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b4-b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.29.	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	R/W
b7	ASEL	Analog Input Function Select	0: Used other than as analog pin. 1: Used as analog pin.	R/W

**Table 21.29 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-LQFP**

PSEL[4:0] Settings	Pin							
	PE0	PE1	PE2	PE3	PE4	PE5	PE6	PE7
00000b (initial value)	Hi-Z							
00001b	—	MTIOC4C	MTIOC4A	MTIOC4B	MTIOC4D	MTIOC4C	—	—
00010b	—	—	—	—	MTIOC1A	MTIOC2B	—	—
00011b	TIOCC9	TIOCD9	TIOCA9	TIOCB9	TIOCA10	TIOCB10	TIOCA11	TIOCB11
00110b	—	PO18	PO23	PO26	PO28	—	—	—
00111b	—	—	—	POE8#	—	—	—	—
01011b	—	—	—	—	—	—	CTS4# RTS4# SS4#	—
01100b	SCK12	TXD12 SMOS12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSCL12 RXDX12	CTS12# RTS12# SS12#	—	—	—	—
01101b	SSLB1	SSLB2	SSLB3	MISOB	SSLB0	RSPCKB	MOSIB	MISOB
01110b	—	RSPCKB	MOSIB	—	—	—	—	—

—: Do not specify this value.

**Table 21.30 Register Settings for Input/Output Pin Function in 100-Pin TFLGA, 100-Pin LQFP**

PSEL[4:0] Settings	Pin							
	PE0	PE1	PE2	PE3	PE4	PE5	PE6	PE7
00000b (initial value)	Hi-Z							
00001b	—	MTIOC4C	MTIOC4A	MTIOC4B	MTIOC4D	MTIOC4C	—	—
00010b	—	—	—	—	MTIOC1A	MTIOC2B	—	—
00110b	—	PO18	PO23	PO26	PO28	—	—	—
00111b	—	—	—	POE8#	—	—	—	—
01100b	SCK12	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSCL12 RXDX12	CTS12# RTS12# SS12#	—	—	—	—
01101b	SSLB1	SSLB2	SSLB3	MISOB	SSLB0	RSPCKB	MOSIB	MISOB
01110b	—	RSPCKB	MOSIB	—	—	—	—	—

—: Do not specify this value.

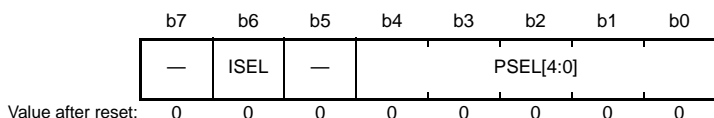
**Table 21.31 Register Settings for Input/Output Pin Function in 80-Pin LQFP**

PSEL[4:0] Settings	Pin					
	PE0	PE1	PE2	PE3	PE4	PE5
00000b (initial value)	Hi-Z					
00001b	—	MTIOC4C	MTIOC4A	MTIOC4B	MTIOC4D	MTIOC4C
00010b	—	—	—	—	MTIOC1A	MTIOC2B
00110b	—	PO18	PO23	PO26	PO28	—
00111b	—	—	—	POE8#	—	—
01100b	SCK12	TXD12 SMOSI12 SSDA12 TXDX12 SIOX12	RXD12 SMISO12 SSCL12 RXDX12	CTS12# RTS12# SS12#	—	—
01101b	SSLB1	SSLB2	SSLB3	MISOB	SSLB0	RSPCKB
01110b	—	RSPCKB	MOSIB	—	—	—

—: Do not specify this value.

21.2.17 PF<sub>n</sub> Pin Function Select Register (PF<sub>n</sub>PFS) (n = 0 to 2, 5)

Address(es): PF0PFS 0008 C1B8h, PF1PFS 0008 C1B9h, PF2PFS 0008 C1BAh, PF5PFS 0008 C1BDh



Bit	Symbol	Bit Name	Description	R/W
b4-b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.32.	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 IRQ <sub>n</sub> input pin 1: Used as IRQ <sub>n</sub> input pin	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

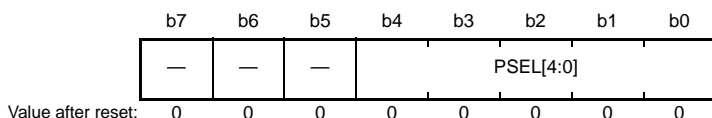
**Table 21.32 Register Settings for Input/Output Pin Function in 177-Pin TFLGA, 176-Pin LFBGA, 176-Pin LQFP**

PSEL[4:0] Settings	Pin		
	PF0	PF1	PF2
00000b (initial value)	Hi-Z		
01010b	TXD1 SMOS1 SSDA1	SCK1	RXD1 SMISO1 SSCL1

—: Do not specify this value.

### 21.2.18 PJ3 Pin Function Control Register (PJ3PFS)

Address(es): PJ3PFS 0008 C1D3h



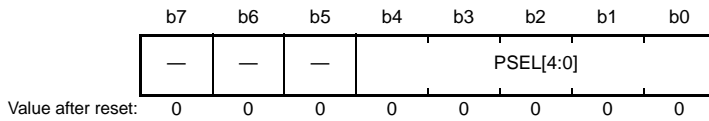
Bit	Symbol	Bit Name	Description	R/W
b4-b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.33.	R/W
b7-b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

**Table 21.33 Register Settings for Input/Output Pin Function in 177-/145-/100-Pin TFLGA, 176-Pin LFBGA, 176-/144-/100-Pin LQFP**

PSEL[4:0] Settings	Pin
	PJ3
00000b (initial value)	Hi-Z
00001b	MTIOC3C
01010b	CTS6# RTS6# SS6#
01011b	CTS0# RTS0# SS0#

### 21.2.19 P<sub>K</sub>n Pin Function Control Register (P<sub>K</sub>nPFS) (n = 2 to 5)

Address(es): PK2PFS 0008 C1DAh, PK3PFS 0008 C1DBh, PK4PFS 0008 C1DCh, PK5PFS 0008 C1DDh



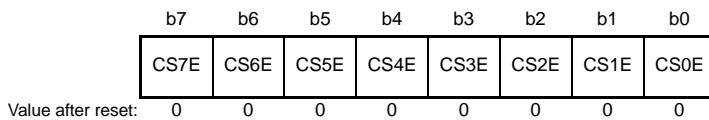
Bit	Symbol	Bit Name	Description	R/W
b4-b0	PSEL[4:0]	Pin Function Select	These bits select the peripheral function. For individual pin functions, see Table 21.34.	R/W
b7-b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

**Table 21.34 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin LQFP**

PSEL[4:0] Settings	Pin			
	PK2	PK3	PK4	PK5
00000b (initial value)	Hi-Z			
01010b	TXD9 SMOSI9 SSDA9	RXD9 SMISO9 SSCL9	RXD4 SMISO4 SSCL4	TXD4 SMOSI4 SSDA4

### 21.2.20 CS Output Enable Register (PFCSE)

Address(es): 0008 C100h



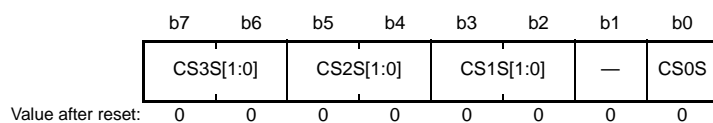
Bit	Symbol	Bit Name	Description	R/W
b0	CS0E	CS0 Enable	0: CSn# output disabled	R/W
b1	CS1E	CS1 Enable	1: CSn# output enabled	R/W
b2	CS2E	CS2 Enable	(n = 0 to 7)	R/W
b3	CS3E	CS3 Enable		R/W
b4	CS4E	CS4 Enable		R/W
b5	CS5E	CS5 Enable		R/W
b6	CS6E	CS6 Enable		R/W
b7	CS7E	CS7 Enable		R/W

#### CSnE Bits (CSn Enable) (n = 0 to 7)

These bits enable or disable the corresponding pins to output the CSn# signal. To enable output of the CSn# signal, set the corresponding CSnE bit in PFCSE to 1.

## 21.2.21 CS Output Pin Select Register 0 (PFCSS0)

Address(es): 0008 C102h



Bit	Symbol	Bit Name	Description	R/W
b0	CS0S	CS0# Output Pin Select	0: Set P60 as CS0# output pin 1: Set PC7 as CS0# output pin	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3, b2	CS1S[1:0]	CS1# Output Pin Select	b3 b2 0 0: Set P61 as CS1# output pin 0 1: Set P71 as CS1# output pin 1 x: Set PC6 as CS1# output pin	R/W
b5, b4	CS2S[1:0]	CS2# Output Pin Select	b5 b4 0 0: Set P62 as CS2# output pin 0 1: Set P72 as CS2# output pin 1 x: Set PC5 as CS2# output pin	R/W
b7, b6	CS3S[1:0]	CS3# Output Pin Select	b7 b6 0 0: Set P63 as CS3# output pin 0 1: Set P73 as CS3# output pin 1 x: Set PC4 as CS3# output pin	R/W

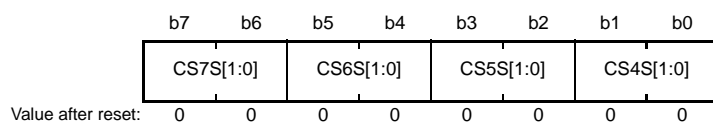
x: Don't care

**CS0S Bit (CS0# Output Pin Select)****CSnS[1:0] Bits (CSn# Output Pin Select) (n = 1 to 3)**

When CSn# output is enabled (the PFCSE.CSnE bit = 1), the CSn# output pin is selected.

## 21.2.22 CS Output Pin Select Register 1 (PFCSS1)

Address(es): 0008 C103h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	CS4S[1:0]	CS4# Output Pin Select	b1 b0 0 0: Set P64 as CS4# output pin 0 1: Set P74 as CS4# output pin 1 x: Set P24 as CS4# output pin	R/W
b3, b2	CS5S[1:0]	CS5# Output Pin Select	b3 b2 0 0: Set P65 as CS5# output pin 0 1: Set P75 as CS5# output pin 1 x: Set P25 as CS5# output pin	R/W
b5, b4	CS6S[1:0]	CS6# Output Pin Select	b5 b4 0 0: Set P66 as CS6# output pin 0 1: Set P76 as CS6# output pin 1 x: Set P26 as CS6# output pin	R/W
b7, b6	CS7S[1:0]	CS7# Output Pin Select	b7 b6 0 0: Set P67 as CS7# output pin 0 1: Set P77 as CS7# output pin 1 x: Set P27 as CS7# output pin	R/W

x: Don't care

**CSnS[1:0] Bits (CSn# Output Pin Select) (n = 4 to 7)**

When CSn# output is enabled (the PFCSE.CSnE bit = 1), the CSn# output pin is selected.

### 21.2.23 Address Output Enable Register 0 (PFAOE0)

Address(es): 0008 C104h

b7	b6	b5	b4	b3	b2	b1	b0
A15E	A14E	A13E	A12E	A11E	A10E	A9E	A8E

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	A8E	Address A8 Output Enable	0: Disables A8 output. 1: Enables A8 output.	R/W
b1	A9E	Address A9 Output Enable	0: Disables A9 output. 1: Enables A9 output.	R/W
b2	A10E	Address A10 Output Enable	0: Disables A10 output. 1: Enables A10 output.	R/W
b3	A11E	Address A11 Output Enable	0: Disables A11 output. 1: Enables A11 output.	R/W
b4	A12E	Address A12 Output Enable	0: Disables A12 output. 1: Enables A12 output.	R/W
b5	A13E	Address A13 Output Enable	0: Disables A13 output. 1: Enables A13 output.	R/W
b6	A14E	Address A14 Output Enable	0: Disables A14 output. 1: Enables A14 output.	R/W
b7	A15E	Address A15 Output Enable	0: Disables A15 output. 1: Enables A15 output.	R/W

### 21.2.24 Address Output Enable Register 1 (PFAOE1)

Address(es): 0008 C105h

b7	b6	b5	b4	b3	b2	b1	b0
A23E	A22E	A21E	A20E	A19E	A18E	A17E	A16E

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	A16E	AddressA16 Output Enable	0: Disables A16 output. 1: Enables A16 output.	R/W
b1	A17E	AddressA17 Output Enable	0: Disables A17 output. 1: Enables A17 output.	R/W
b2	A18E	AddressA18 Output Enable	0: Disables A18 output. 1: Enables A18 output.	R/W
b3	A19E	AddressA19 Output Enable	0: Disables A19 output. 1: Enables A19 output.	R/W
b4	A20E	AddressA20 Output Enable	0: Disables A20 output. 1: Enables A20 output.	R/W
b5	A21E	AddressA21 Output Enable	0: Disables A21 output. 1: Enables A21 output.	R/W
b6	A22E	AddressA22 Output Enable	0: Disables A22 output. 1: Enables A22 output.	R/W
b7	A23E	AddressA23 Output Enable	0: Disables A23 output. 1: Enables A23 output.	R/W



### 21.2.25 External Bus Control Register 0 (PFBCR0)

Address(es): 0008 C106h

b7	b6	b5	b4	b3	b2	b1	b0
WR32B C32E	WR1B C1E	DH32E	DHE	—	—	ADRH MS	ADRLE

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	ADRLE	A0 to A7 Output Enable	0: Configures PA0 to PA7 as the I/O port pins. 1: Configures PA0 to PA7 as the external address bus A0 to A7.	R/W
b1	ADRHMS	A16 to A23 Output Enable	0: Configures PC0 to PC7 as the external address bus A16 to A23. 1: Configures P90 to P97 as the external address bus A16 to A23.	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DHE	D8 to D15 Output Enable	0: Configures PE0 to PE7 as the I/O port pins. 1: Configures PE0 to PE7 as the external data bus D8 to D15.	R/W
b5	DH32E	D16 to D31 Output Enable	0: Set PG0 to PG7, and P90 to P97 as I/O port 1: Set PG0 to PG7, and P90 to P97 as external data bus D16 to D31	R/W
b6	WR1BC1E	WR1#/BC1# Output Enable	0: Configures P51 as the I/O port pin. 1: Configures P51 as the WR1# or BC1# pin.	R/W
b7	WR32BC32E	WR3#/BC3# Output Enable WR2#/BC2# Output Enable	0: Set P56 and P57 as I/O port 1: Set P56 as WR#2 or BC2#, and set P57 as WR#3 or BC3#	R/W

### 21.2.26 External Bus Control Register 1 (PFBCR1)

Address(es): 0008 C107h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	ALEOE	WAITS[1:0]	—

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	WAITS[1:0]	WAIT Select	b1 b0 0 0: Configures P57 as the WAIT# input pin. 0 1: Configures P55 as the WAIT# input pin. 1 0: Configures PC5 as the WAIT# input pin. 1 1: Configures P51 as the WAIT# input pin.	R/W
b2	ALEOE	ALE Output Enable	0: Configures P54 as an I/O port pin. 1: Configures P54 as the ALE output pin.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

#### WAITS[1:0] Bits (WAIT Select Bits)

The port pin specified by the WAIT[1:0] bits becomes the WAIT# pin when the external bus is enabled. However, if the specified port pin is not to be used as the WAIT# pin, the external wait enable bit (EWENB) in the CSn mode register (CSnMOD) can be cleared (disabling external wait) to make the pin available for use as a general input port pin.

If the specified WAIT# pin is not to be used as the WAIT input or as a general input port pin, pull the level on the WAIT# pin up or down.

### 21.2.27 USB0 Control Register (PFUSB0)

Address(es): 0008 C114h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	PUPHZ S	—	—
Value after reset:	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
b2	PUPHZS	PUPHZ Select	0: USB0_DPUPE pin: high-level output or low-level output (external pull-up control signal) 1: USB0_DPUPE pin: high-level output or Hi-Z state (USB0_DP pin pull-up output)	R/W
b7-b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

When the DPUPE pin function has been assigned to a pin by the settings of the port mode register (PMR) and Pmn pin function control register (PmnPFS), the function of the PUPHZS bit becomes effective.

#### PUPHZS Bit (PUPHZ Select)

This bit selects the output mode (external pull-up control/pin pull-up output) for the DPUPE pin of the USB0.

When the PUPHZS bit is set to 0, the control signal output mode for the external pull-up IC is selected and the high-active control signal is output from the DPUPE pin. When the DP pin of the USB0 is pulled up, the DPUPE pin is set to the high-level output state. When the pulled-up state of the DP pin is canceled, the DPUPE pin is set to the low-level output state.

When the PUPHZS bit is set to 1, the output mode where the DP pin is directly pulled-up is selected. When the DP pin is pulled up, the DPUPE pin is set to the high-level output state. When the pulled-up state of the DP pin is canceled, the DPUPE pin is set to the Hi-Z state.

### 21.3 How to Set the External Bus Interface

If the external bus interface is to be used, set the MPC registers according to Table 21.35 and then set the external bus enable bit (EXBE) in system control register 0 (SYSCR0) to 1.

Table 21.35 list how to set up port pins to act as the external bus interface. For details on the relevant registers of the MPC, refer to 21.2, Register Descriptions.

**Table 21.35 How to Set the External Bus Interface (1/3)**

Port	Output Signal	Settings of MPC Registers		
		177-Pin, 176-Pin	145-Pin, 144-Pin	100-Pin
P24	CS4#	PFCSE.CS4E=1, PFCSS1.CS4S[1:0] = 10/11		
P25	CS5#	PFCSE.CS5E=1, PFCSS1.CS5S[1:0] = 10/11		
P26	CS6#	PFCSE.CS6E=1, PFCSS1.CS6S[1:0] = 10/11		
P27	CS7#	PFCSE.CS7E=1, PFCSS1.CS7S[1:0] = 10/11		
P50	WR0# / WR#	—		
P51	WR1# / BC1#	PFBCR0.WR1BC1E = 1		
	WAIT#	PFBCR1.WAITS[1:0] = 11		
P52	RD#	—		
P53	BCLK	—		
P54	ALE	PFBCR1.ALEOE = 1		
P55	WAIT#	PFBCR1.WAITS[1:0] = 01		
P56	WR2# / BC2#	PFBCR0.WR32BC32E = 1	(not provided)	(not provided)
P57	WR3# / BC3#	PFBCR0.WR32BC32E = 1	(not provided)	(not provided)
	WAIT#	PFBCR1.WAITS[1:0] = 00	(not provided)	(not provided)
P60	CS0#	PFCSE.CS0E = 1, PFCSS0.CS0S=0		(not provided)
P61	CS1#	PFCSE.CS1E = 1, PFCSS0.CS1S[1:0] = 00		(not provided)
P62	CS2#	PFCSE.CS2E = 1, PFCSS0.CS2S[1:0] = 00		(not provided)
P63	CS3#	PFCSE.CS3E = 1, PFCSS0.CS3S[1:0] = 00		(not provided)
P64	CS4#	PFCSE.CS4E = 1, PFCSS1.CS4S[1:0] = 00		(not provided)
P65	CS5#	PFCSE.CS5E = 1, PFCSS1.CS5S[1:0] = 00		(not provided)
P66	CS6#	PFCSE.CS6E = 1, PFCSS1.CS6S[1:0] = 00		(not provided)
P67	CS7#	PFCSE.CS7E = 1, PFCSS1.CS7S[1:0] = 00		(not provided)
P71	CS1#	PFCSE.CS1E = 1, PFCSS0.CS1S[1:0] = 01		(not provided)
P72	CS2#	PFCSE.CS2E = 1, PFCSS0.CS2S[1:0] = 01		(not provided)
P73	CS3#	PFCSE.CS3E = 1, PFCSS0.CS3S[1:0] = 01		(not provided)
P74	CS4#	PFCSE.CS4E = 1, PFCSS1.CS4S[1:0] = 01		(not provided)
P75	CS5#	PFCSE.CS5E = 1, PFCSS1.CS5S[1:0] = 01		(not provided)
P76	CS6#	PFCSE.CS6E = 1, PFCSS1.CS6S[1:0] = 01		(not provided)
P77	CS7#	PFCSE.CS7E = 1, PFCSS1.CS7S[1:0] = 01		(not provided)
P90	D16	PFBCR0.DH32E = 1	(not provided)	(not provided)
	A16	PFAOE1.A16E = 1, PFBCR0.ADRHMS = 1		(not provided)
P91	D17	PFBCR0.DH32E = 1	(not provided)	(not provided)
	A17	PFAOE1.A17E = 1, PFBCR0.ADRHMS = 1		(not provided)
P92	D18	PFBCR0.DH32E = 1	(not provided)	(not provided)
	A18	PFAOE1.A18E = 1, PFBCR0.ADRHMS=1		(not provided)
P93	D19	PFBCR0.DH32E=1	(not provided)	(not provided)
	A19	PFAOE1.A19E = 1, PFBCR0.ADRHMS = 1		(not provided)

Table 21.35 How to Set the External Bus Interface (2/3)

Port	Output Signal	Settings of MPC Registers		
		177-Pin, 176-Pin	145-Pin, 144-Pin	100-Pin
P94	D20	PFBCR0.DH32E = 1	(not provided)	(not provided)
	A20	PFAOE1.A20E = 1, PFBCR0.ADRHMS = 1	(not provided)	(not provided)
P95	D21	PFBCR0.DH32E = 1	(not provided)	(not provided)
	A21	PFAOE1.A21E = 1, PFBCR0.ADRHMS = 1	(not provided)	(not provided)
P96	D22	PFBCR0.DH32E=1	(not provided)	(not provided)
	A22	PFAOE1.A22E = 1, PFBCR0.ADRHMS = 1	(not provided)	(not provided)
P97	D23	PFBCR0.DH32E = 1	(not provided)	(not provided)
	A23	PFAOE1.A23E = 1, PFBCR0.ADRHMS = 1	(not provided)	(not provided)
PA0	A0 / BC0#	PFBCR0.ADRLE = 1		
PA1	A1	PFBCR0.ADRLE = 1		
PA2	A2	PFBCR0.ADRLE = 1		
PA3	A3	PFBCR0.ADRLE = 1		
PA4	A4	PFBCR0.ADRLE = 1		
PA5	A5	PFBCR0.ADRLE = 1		
PA6	A6	PFBCR0.ADRLE = 1		
PA7	A7	PFBCR0.ADRLE = 1		
PB0	A8	PFAOE0.A8E = 1		
PB1	A9	PFAOE0.A9E = 1		
PB2	A10	PFAOE0.A10E = 1		
PB3	A11	PFAOE0.A11E = 1		
PB4	A12	PFAOE0.A12E = 1		
PB5	A13	PFAOE0.A13E = 1		
PB6	A14	PFAOE0.A14E = 1		
PB7	A15	PFAOE0.A15E = 1		
PC0	A16	PFAOE1.A16E = 1, PFBCR0.ADRHMS = 0		
PC1	A17	PFAOE1.A17E = 1, PFBCR0.ADRHMS = 0		
PC2	A18	PFAOE1.A18E = 1, PFBCR0.ADRHMS = 0		
PC3	A19	PFAOE1.A19E = 1, PFBCR0.ADRHMS = 0		
PC4	A20	PFAOE1.A20E = 1, PFBCR0.ADRHMS = 0		
	CS3#	PFCSE.CS3E = 1, PFCSS0.CS3S[1:0] = 10/11		
PC5	A21	PFAOE1.A21E = 1, PFBCR0.ADRHMS = 0		
	CS2#	PFCSE.CS2E = 1, PFCSS0.CS2S[1:0] = 10/11		
	WAIT#	PFBCR1.WAITS[1:0] = 10		
PC6	A22	PFAOE1.A22E = 1, PFBCR0.ADRHMS = 0		
	CS1#	PFCSE.CS1E = 1, PFCSS0.CS1S[1:0] = 10/11		
PC7	A23	PFAOE1.A23E = 1, PFBCR0.ADRHMS = 0		
	CS0#	PFCSE.CS0E = 1, PFCSS0.CS0S = 1		
PD0	D0[A0/D0]	—		
PD1	D1[A1/D1]	—		
PD2	D2[A2/D2]	—		
PD3	D3[A3/D3]	—		
PD4	D4[A4/D4]	—		

**Table 21.35 How to Set the External Bus Interface (3/3)**

Port	Output Signal	Settings of MPC Registers		
		177-Pin, 176-Pin	145-Pin, 144-Pin	100-Pin
PD5	D5[A5/D5]	—		
PD6	D6[A6/D6]	—		
PD7	D7[A7/D7]	—		
PE0	D8[A8/D8]	PFBCR0.DHE = 1		
PE1	D9[A9/D9]	PFBCR0.DHE = 1		
PE2	D10[A10/D10]	PFBCR0.DHE = 1		
PE3	D11[A11/D11]	PFBCR0.DHE = 1		
PE4	D12[A12/D12]	PFBCR0.DHE = 1		
PE5	D13[A13/D13]	PFBCR0.DHE = 1		
PE6	D14[A14/D14]	PFBCR0.DHE = 1		
PE7	D15[A15/D15]	PFBCR0.DHE = 1		
PG0	D24	PFBCR0.DH32E = 1	(not provided)	(not provided)
PG1	D25	PFBCR0.DH32E = 1	(not provided)	(not provided)
PG2	D26	PFBCR0.DH32E = 1	(not provided)	(not provided)
PG3	D27	PFBCR0.DH32E = 1	(not provided)	(not provided)
PG4	D28	PFBCR0.DH32E = 1	(not provided)	(not provided)
PG5	D29	PFBCR0.DH32E = 1	(not provided)	(not provided)
PG6	D30	PFBCR0.DH32E = 1	(not provided)	(not provided)
PG7	D31	PFBCR0.DH32E = 1	(not provided)	(not provided)

## 21.4 Usage Notes

### 21.4.1 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 = 0 to 9, A to F, J, and K, n = 0 to 7)
- (4) Specify the input/output function for the pin through the PSEL[4:0] bit settings in the PFS 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.

### 21.4.2 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 port mn pin function select register is set while the PMR register 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 port mn pin function select 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) Ports 0, 4, 9, D, and E also function as analog Input/output pins for the A/D converter and D/A converter. When using these ports as analog Input/output pins, set them to general input pins by clearing the corresponding bits in the port mode register (PMR) and port direction register (PDR) to 0 and set the PmnPFS.ASEL bit to 1, to avoid degradation of accuracy.
- (5) The initial value of the time capture event input pin enable bit (TCEN) in the time capture control register y (RTCCRY, y = 0 to 2) is undefined after a reset. Therefore, set this bit to 0 to avoid unnecessary input.
- (6) 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 21.36. Pin states are readable while the value of the ASEL bit is 0. Only change the value of the PSEL[4:0] bits while the PMR.Bj bit is 0.

**Table 21.36 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 PmnPFS.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 21.2 to Table 21.34)	Set the PmnPFS.ISEL bit to 1 if these are multiplexed with interrupt inputs.
Interrupt inputs	0	0	0	1	x	
NMI	x	x	x	x*1	x	Register settings are not required.
Analog inputs and outputs	0	0	1	x*1	x	Set these as general input port pins so that the output buffers are turned off.
Time-capture event-input pins	0	0	x	0/1	x	Set these as general input port pins so that the output buffers are turned off.
External bus	0	x	0	0	x	Set the PMR.Bn bit to 0 and do not select the peripheral function.
JTAG interface	0	x	x	0	x	Set the PMR.Bn bit and the PmnPFS.ISEL bit to 0 and switch the input buffers off.
FINE interface	0	x	x	0	x	Set the PMR.Bn bit and the PmnPFS.ISEL bit to 0 and switch the input buffers off.
EXTAL/XTAL	0	0	x	x*1	x	Set these as general input port pins so that the output buffers are turned off.
XCIN/XCOUT	0	0	x	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 1. Even if the PmnPFS.ISEL bit is set to 1, the pin will not function as an IRQn input pin.

- 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 PMR.Bn (to 0); pulling up is automatically turned off for outputs from peripheral modules other than the RIIC.
  - If an input pin for time-capture events is not in use, clear the time capture event input pin enable bit (TCEN) in time capture control register y (RTCCRy) to 0 (disabled). The value of each TCEN bit in a time capture control register y (RTCCRy; y = 0 to 2) is undefined after a reset.
  - Do not make settings to assign multiple external bus signals to a single pin.

### 21.4.3 Notes on the Use of Analog Functions

To use an analog function, set the corresponding bits in both the port mode register (PMR) and port direction register (PDR) to 0 so that the pin acts as a general input port. After that, set the pin function select bit in the Pmn pin function control register (PmnPFS.ASEL[1:0]) to 1.

## 22. Multi-Function Timer Pulse Unit 2 (MTU2a)

### 22.1 Overview

The RX630 Group has an on-chip multi-function timer pulse unit 2 (MTU). Each unit comprises a 16-bit timer with six channels (MTU0 to MTU5).

Table 22.1 lists the specifications of the MTU, and Table 22.2 lists the function list. Figure 22.1 shows a block diagram of the MTU.

**Table 22.1 Specifications of MTU**

Item	Description
Pulse input/output	16 lines max.
Pulse input	3 lines
Count clock	Eight clocks or seven clocks for each channel (four clocks for MTU5)
Available operations	<p>[MTU0 to MTU4]</p> <ul style="list-style-type: none"> <li>Waveform output at compare match</li> <li>Input capture function (noise filter set function)</li> <li>Counter clear operation</li> <li>Simultaneous writing to multiple timer counters (TCNT)</li> <li>Simultaneous clearing by compare match or input capture</li> <li>Simultaneous register input/output by synchronous counter operation</li> <li>A maximum of 12-phase PWM output is available in combination with synchronous operation</li> </ul> <hr/> <p>[MTU0, MTU3, MTU4]</p> <ul style="list-style-type: none"> <li>Buffer operation specifiable</li> <li>AC synchronous motor (brushless DC motor) drive mode using complementary PWM output and reset-synchronized PWM output is settable and the selection of two types of waveform outputs (chopping and level) is possible.</li> </ul> <hr/> <p>[MTU1, MTU2]</p> <ul style="list-style-type: none"> <li>Phase counting mode specifiable independently</li> <li>Cascade connection operation</li> </ul> <hr/> <p>[MTU3, MTU4]</p> <ul style="list-style-type: none"> <li>A total of six-phase waveform output, which includes three phases each for positive and negative complementary PWM or reset PWM output, by interlocking operation</li> </ul> <hr/> <p>[MTU5]</p> <ul style="list-style-type: none"> <li>Dead time compensation counter</li> </ul>
Complementary PWM mode	<ul style="list-style-type: none"> <li>Interrupts at the crest and trough of the counter value</li> <li>A/D converter start triggers can be skipped</li> </ul>
Interrupt sources	28 sources
Buffer operation	Automatic transfer of register data
Trigger generation	<p>Programmable pulse generator (PPG) output trigger can be generated</p> <hr/> <p>A/D converter start trigger can be generated</p>
Power reduction	Each unit can be independently set to module-stop stat



**Table 22.2 MTU Functions (1/2)**

Item	MTU0	MTU1	MTU2	MTU3	MTU4	MTU5
Count clock	PCLK/1	PCLK/1	PCLK/1	PCLK/1	PCLK/1	PCLK/1
	PCLK/4	PCLK/4	PCLK/4	PCLK/4	PCLK/4	PCLK/4
	PCLK/16	PCLK/16	PCLK/16	PCLK/16	PCLK/16	PCLK/16
	PCLK/64	PCLK/64	PCLK/64	PCLK/64	PCLK/64	PCLK/64
	MTCLKA	PCLK/256	PCLK/1024	PCLK/256	PCLK/256	PCLK/256
	MTCLKB	MTCLKA	MTCLKA	PCLK/1024	PCLK/1024	PCLK/1024
	MTCLKC	MTCLKB	MTCLKB	MTCLKA	MTCLKA	MTCLKA
	MTCLKD	MTCLKC	MTCLKB	MTCLKB	MTCLKB	MTCLKB
General registers (TGR)	TGRA	TGRA	TGRA	TGRA	TGRA	TGRU
	TGRB	TGRB	TGRB	TGRB	TGRB	TGRV
	TGRE					TGRW
General registers/ buffer registers	TGRC	—	—	TGRC	TGRC	—
	TGRD			TGRD	TGRD	
	TGRF					
I/O pins	MTIOC0A	MTIOC1A	MTIOC2A	MTIOC3A	MTIOC4A	Input pins MTIC5U MTIC5V MTIC5W
	MTIOC0B	MTIOC1B	MTIOC2B	MTIOC3B	MTIOC4B	
	MTIOC0C			MTIOC3C	MTIOC4C	
	MTIOC0D			MTIOC3D	MTIOC4D	
Counter clear function	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	○	○	○	○	—
	1 output	○	○	○	○	—
	Toggle output	○	○	○	○	—
Input capture function	○	○	○	○	○	○
Synchronous operation	○	○	○	○	○	—
PWM mode 1	○	○	○	○	○	—
PWM mode 2	○	○	○	—	—	—
Complementary PWM mode	—	—	—	○	○	—
Reset-synchronized PWM	—	—	—	○	○	—
AC synchronous motor drive mode	○	—	—	○	○	—
Phase counting mode	—	○	○	—	—	—
Buffer operation	○	—	—	○	○	—
Dead time compensation counter function	—	—	—	—	—	○
DMAC activation	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	—
DTC activation	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture and TCNT overflow or underflow	TGR compare match or input capture

**Table 22.2 MTU Functions (2/2)**

Item	MTU0	MTU1	MTU2	MTU3	MTU4	MTU5
A/D converter start trigger	TGRA compare match or input capture TGRB compare match or input capture TGRE compare match TGRF compare match	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	— TCNT underflow (trough) in complementary PWM mode
PPG trigger	TGRA and TGRB compare match or input capture	TGRA and TGRB compare match or input capture	TGRA and TGRB compare match or input capture	TGRA and TGRB compare match or input capture	—	—
Interrupt sources	7 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 0A</li> <li>• Compare match or input capture 0B</li> <li>• Compare match or input capture 0C</li> <li>• Compare match or input capture 0D</li> <li>• Compare match 0E</li> <li>• Compare match 0F</li> <li>• Overflow</li> </ul>	4 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 1A</li> <li>• Compare match or input capture 1B</li> <li>• Overflow</li> <li>• Underflow</li> </ul>	4 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 2A</li> <li>• Compare match or input capture 2B</li> <li>• Overflow</li> <li>• Underflow</li> </ul>	5 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 3A</li> <li>• Compare match or input capture 3B</li> <li>• Compare match or input capture 3C</li> <li>• Compare match or input capture 3D</li> <li>• Overflow</li> </ul>	5 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 4A</li> <li>• Compare match or input capture 4B</li> <li>• Compare match or input capture 4C</li> <li>• Compare match or input capture 4D</li> <li>• Overflow or underflow</li> </ul>	3 sources <ul style="list-style-type: none"> <li>• Compare match or input capture 5U</li> <li>• Compare match or input capture 5V</li> <li>• Compare match or input capture 5W</li> </ul>
A/D converter start request delaying function	—	—	—	—	• A/D converter start request at a match between TADCORA and TCNT or A/D converter start request at a match between TADCORB and TCNT	—
Interrupt skipping function	—	—	—	• Skips TGRA compare match interrupts	• Skips TCIV interrupts	—
Module-stop function	MSTPCRA.MSTPA9*1					

○: Possible

—: Not possible

Note 1. For details on the module-stop function, see section 11, Low Power Consumption.

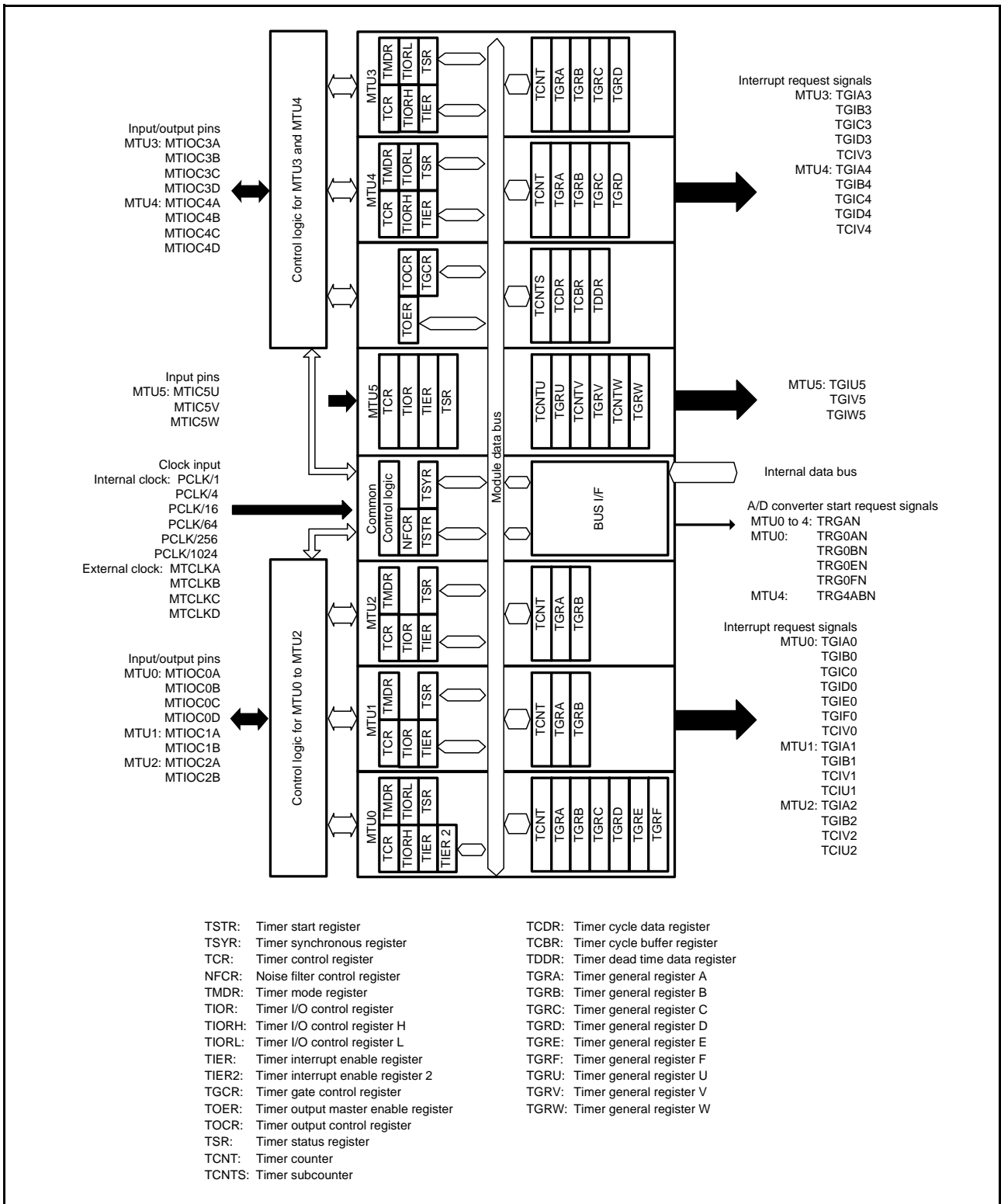


Figure 22.1 Block Diagram of MTU

Table 22.3 lists the pin configuration of the MTU.

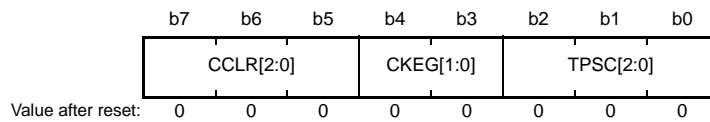
**Table 22.3 Pin Configuration of MTU**

Module Symbol	Pin Name	I/O	Function
MTU	MTCLKA	Input	External clock A input pin (MTU1 phase counting mode A phase input)
	MTCLKB	Input	External clock B input pin (MTU1 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)
MTU0	MTIOC0A	I/O	TGRA0 input capture input/output compare output/PWM output pin
	MTIOC0B	I/O	TGRB0 input capture input/output compare output/PWM output pin
	MTIOC0C	I/O	TGRC0 input capture input/output compare output/PWM output pin
	MTIOC0D	I/O	TGRD0 input capture input/output compare output/PWM output pin
MTU1	MTIOC1A	I/O	TGRA1 input capture input/output compare output/PWM output pin
	MTIOC1B	I/O	TGRB1 input capture input/output compare output/PWM output pin
MTU2	MTIOC2A	I/O	TGRA2 input capture input/output compare output/PWM output pin
	MTIOC2B	I/O	TGRB2 input capture input/output compare output/PWM output pin
MTU3	MTIOC3A	I/O	TGRA3 input capture input/output compare output/PWM output pin
	MTIOC3B	I/O	TGRB3 input capture input/output compare output/PWM output pin
	MTIOC3C	I/O	TGRC3 input capture input/output compare output/PWM output pin
	MTIOC3D	I/O	TGRD3 input capture input/output compare output/PWM output pin
MTU4	MTIOC4A	I/O	TGRA4 input capture input/output compare output/PWM output pin
	MTIOC4B	I/O	TGRB4 input capture input/output compare output/PWM output pin
	MTIOC4C	I/O	TGRC4 input capture input/output compare output/PWM output pin
	MTIOC4D	I/O	TGRD4 input capture input/output compare output/PWM output pin
MTU5	MTIC5U	Input	TGRU5 input capture input/external pulse input pin
	MTIC5V	Input	TGRV5 input capture input/external pulse input pin
	MTIC5W	Input	TGRW5 input capture input/external pulse input pin

## 22.2 Register Descriptions

### 22.2.1 Timer Control Register (TCR)

Address(es): MTU0.TCR 0008 8700h, MTU1.TCR 0008 8780h, MTU2.TCR 0008 8800h,  
MTU3.TCR 0008 8600h, MTU4.TCR 0008 8601h, MTU5.TCRU 0008 8884h,  
MTU5.TCRV 0008 8894h, MTU5.TCRW 0008 88A4h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC[2:0]	Time Prescaler Select	See Table 22.6 to Table 22.10.	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	See Table 22.4 and Table 22.5.	R/W

x: Don't care

The MTU has a total of eight TCR registers, one each for MTU0 to MTU4 and three (TCRU, TCRV, and TCRW) for MTU5.

TCR is an 8-bit readable/writable register that controls the TCNT operation for each channel. TCR values should be specified only while TCNT operation is stopped.

#### TPSC[2:0] Bits (Time Prescaler Select)

These bits select the TCNT counter clock. The clock source can be selected independently for each channel. See Table 22.6 to Table 22.10 for details.

#### CKEG[1:0] Bits (Clock Edge Select)

These bits select the input clock edge. When the input clock is counted at both edges, the input clock period is halved (e.g. PCLK/4 at both edges = PCLK/2 at rising edge). If phase counting mode is used on MTU1 and MTU2, the setting of these bits is ignored and the phase counting mode setting has priority. Internal clock edge selection is valid when the input clock is PCLK/4 or slower. When PCLK/1 or the overflow/underflow in another channel is selected for the input clock, a value can be written to these bits but counter operation compiles with the initial value.

#### CCLR[2:0] Bits (Counter Clear)

These bits select the TCNT counter clearing source. See Table 22.4 and Table 22.5 for details.

**Table 22.4 CCLR[2:0] (MTU0, MTU3, and MTU4)**

Channel	Bit 7	Bit 6	Bit 5	Description
	CCLR2	CCLR1	CCLR0	
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 TSYR.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 22.5 CCLR[2:0] (MTU1 and MTU2)**

Channel	Bit 7	Bit 6	Bit 5	Description
	Reserved*2	CCLR1	CCLR0	
MTU1, MTU2	0	0	0	TCNT clearing disabled
	0	0	1	TCNT cleared by TGRA compare match/input capture
	0	1	0	TCNT cleared by TGRB compare match/input capture
	0	1	1	TCNT cleared by counter clearing in another channel performing synchronous clearing/synchronous operation*1

Note 1. Synchronous operation is selected by setting the TSYR.SYNC bit to 1.

Note 2. Bit 7 is reserved in MTU1 and MTU2. It is always read as 0. The write value should be 0.

**Table 22.6 TTPSC[2:0] (MTU0)**

Channel	Bit 2	Bit 1	Bit 0	Description
	TPSC2	TPSC1	TPSC0	
MTU0	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on MTCLKA pin input
	1	0	1	External clock: counts on MTCLKB pin input
	1	1	0	External clock: counts on MTCLKC pin input
	1	1	1	External clock: counts on MTCLKD pin input

**Table 22.7 TPSC[2:0] (MTU1)**

Channel	Bit 2	Bit 1	Bit 0	Description
	TPSC2	TPSC1	TPSC0	
MTU1	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on MTCLKA pin input
	1	0	1	External clock: counts on MTCLKB pin input
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	Counts on MTU2.TCNT overflow/underflow

Note: • This setting is ignored when MTU1 is in phase counting mode.

**Table 22.8 TPSC[2:0] (MTU2)**

Channel	Bit 2	Bit 1	Bit 0	Description
	TPSC2	TPSC1	TPSC0	
MTU2	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock: counts on MTCLKA pin input
	1	0	1	External clock: counts on MTCLKB pin input
	1	1	0	External clock: counts on MTCLKC pin input
	1	1	1	Internal clock: counts on PCLK/1024

Note: • This setting is ignored when MTU2 is in phase counting mode.

**Table 22.9 TPSC[2:0] (MTU3 and MTU4)**

Channel	Bit 2	Bit 1	Bit 0	Description
	TPSC2	TPSC1	TPSC0	
MTU3, MTU4	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	Internal clock: counts on PCLK/256
	1	0	1	Internal clock: counts on PCLK/1024
	1	1	0	External clock: counts on MTCLKA pin input
	1	1	1	External clock: counts on MTCLKB pin input

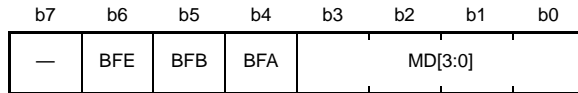
**Table 22.10 TPSC[1:0] (MTU5)**

Channel	Bit 1	Bit 0	Description
	TPSC1	TPSC0	
MTU5	0	0	Internal clock: counts on PCLK/1
	0	1	Internal clock: counts on PCLK/4
	1	0	Internal clock: counts on PCLK/16
	1	1	Internal clock: counts on PCLK/64

Note: • Bits 7 to 2 are reserved in MTU5. These bits are always read as 0. The write value should be 0.

## 22.2.2 Timer Mode Register (TMDR)

Address(es): MTU0.TMDR 0008 8701h, MTU1.TMDR 0008 8781h, MTU2.TMDR 0008 8801h,  
MTU3.TMDR 0008 8602h, MTU4.TMDR 0008 8603h



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. See Table 22.11 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: TGRA and TGRC 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

TMDR is an 8-bit readable/writable register that specifies the operating mode of each channel. TMDR values should be specified only while TCNT operation is stopped.

**Table 22.11 Operating Mode Setting by MD[3:0] Bits**

Bit 3	Bit 2	Bit 1	Bit 0	Description
MD3	MD2	MD1	MD0	
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*1
0	1	0	0	Phase counting mode 1*2
0	1	0	1	Phase counting mode 2*2
0	1	1	0	Phase counting mode 3*2
0	1	1	1	Phase counting mode 4*2
1	0	0	0	Reset-synchronized PWM mode*3
1	0	0	1	Setting prohibited
1	0	1	x	Setting prohibited
1	1	0	0	Setting prohibited
1	1	0	1	Complementary PWM mode 1 (transfer at crest)*3
1	1	1	0	Complementary PWM mode 2 (transfer at trough)*3
1	1	1	1	Complementary PWM mode 3 (transfer at crest and trough)*3

x: Don't care

Note 1. PWM mode 2 cannot be set for MTU3 and MTU4.

Note 2. Phase counting mode cannot be set for MTU0, MTU3, and MTU4.

Note 3. 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 conform to the MTU3 setting, respectively. The initial values should be set for MTU4.

Reset-synchronized PWM mode and complementary PWM mode cannot be set for MTU0, MTU1 and MTU2.



**BFA Bit (Buffer Operation A)**

This bit specifies normal operation for TGRA or buffered operation of the combination of TGRA and TGRC. 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 cleared to 0.

When MTU3 or MTU4 is set to reset-synchronized PWM mode or complementary PWM mode, the buffer operation conforms to the MTU3 setting. Set the BFA bit in MTU4.TMDR 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 22.40 for an illustration of the Tb interval in complementary PWM mode.

**BFB Bit (Buffer Operation B)**

This bit specifies normal operation for TGRB or buffered operation of the combination of TGRB and TGRD. 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 in the Tb interval in complementary PWM mode, the TGIED bit in timer interrupt enable register 3 or 4 (MTU3.TIER or MTU4.TIER) should be cleared to 0.

When MTU3 or MTU4 is set to reset-synchronized PWM mode or complementary PWM mode, the buffer operation conforms to the MTU3 setting. Set the TMDR.BFB bit in MTU4 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 22.40 for an illustration of the Tb interval in complementary PWM mode.

**BFE Bit (Buffer Operation E)**

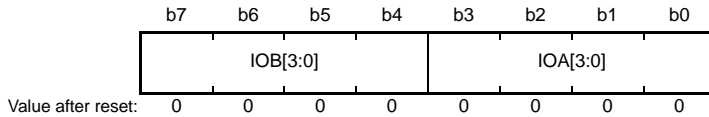
This bit specifies normal operation or buffered operation for MTU0.TGRE and MTU0.TGRF. Compare match with TGRF occurs even when TGRF is used as a buffer register.

In MTU1 to MTU4, this bit is reserved. It is read as 0. The write value should be 0.

### 22.2.3 Timer I/O Control Register (TIOR)

- MTU0.TIORH, MTU1.TIOR, MTU2.TIOR, MTU3.TIORH, MTU4.TIORH

Address(es): MTU0.TIORH 0008 8702h, MTU1.TIOR 0008 8782h, MTU2.TIOR 0008 8802h, MTU3.TIORH 0008 8604h, MTU4.TIORH 0008 8606h

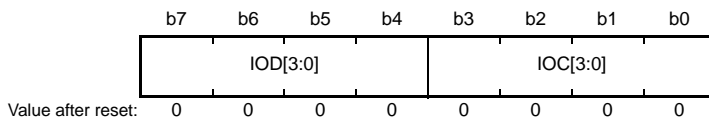


Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOA[3:0]	I/O Control A	See the following tables.*1 MTU0.TIORH: Table 22.20 MTU1.TIOR: Table 22.22 MTU2.TIOR: Table 22.23 MTU3.TIORH: Table 22.24 MTU4.TIORH: Table 22.26	R/W
b7 to b4	IOB[3:0]	I/O Control B	See the following tables.*1 MTU0.TIORH: Table 22.12 MTU1.TIOR: Table 22.14 MTU2.TIOR: Table 22.15 MTU3.TIORH: Table 22.16 MTU4.TIORH: Table 22.18	R/W

Note 1. If the IOn[3:0] (n = A, B) bits are changed to an “output prohibited” setting (0000b or 0100b) while the output of 0, output of 1, or toggling of the output in response to compare matches is in progress, the output becomes high impedance.

- MTU0.TIORL, MTU3.TIORL, MTU4.TIORL

Address(es): MTU0.TIORL 0008 8703h, MTU3.TIORL 0008 8605h, MTU4.TIORL 0008 8607h

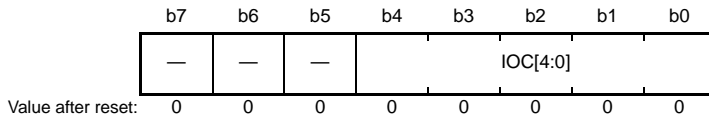


Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOC[3:0]	I/O Control C	See the following tables.*1 MTU0.TIORL: Table 22.21 MTU3.TIORL: Table 22.25 MTU4.TIORL: Table 22.27	R/W
b7 to b4	IOD[3:0]	I/O Control D	See the following tables.*1 MTU0.TIORL: Table 22.13 MTU3.TIORL: Table 22.17 MTU4.TIORL: Table 22.19	R/W

Note 1. If the IOn[3:0] (n = C, D) bits are changed to an “output prohibited” setting (0000b or 0100b) while the output of 0, output of 1, or toggling of the output in response to compare matches is in progress, the output becomes high impedance.

- MTU5.TIORU, MTU5.TIORV, MTU5.TIORW

Address(es): MTU5.TIORU 0008 8886h, MTU5.TIORV 0008 8896h, MTU5.TIORW 0008 88A6h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	IOC[4:0]	I/O Control C	See the following table. MTU5.TIORU, MTU5.TIORV, MTU5.TIORW: Table 22.28	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

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/V/W) each for MTU5.

TIOR should be set when TMDR is set to select normal mode, PWM mode, or phase counting mode.

The initial output specified by TIOR is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.

When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

**Table 22.12 TIORH (MTU0)**

Bit 7	Bit 6	Bit 5	Bit 4	Description
IOB3	IOB2	IOB1	IOB0	MTU0.TGRB Function      MTIOC0B Pin Function
0	0	0	0	Output compare register Output prohibited
0	0	0	1	Initial output is 0. 0 output at compare match.
0	0	1	0	Initial output is 0. 1 output at compare match.
0	0	1	1	Initial output is 0. Toggle output at compare match.
0	1	0	0	Output prohibited
0	1	0	1	Initial output is 1. 0 output at compare match.
0	1	1	0	Initial output is 1. 1 output at compare match.
0	1	1	1	Initial output is 1. 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 count clock in MTU1. Input capture at MTU1.TCNT up-count/down-count.

x: Don't care

Table 22.13 TIORL (MTU0)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD3	IOD2	IOD1	IOD0	MTU0.TGRD Function	MTIOC0D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 count clock in MTU1. Input capture at MTU1.TCNT up-count/down-count.

x: Don't care

Note 1. When the MTU0.TMDR.BFB is set to 1 and MTU0.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.14 TIOR (MTU1)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB3	IOB2	IOB1	IOB0	MTU1.TGRB Function	MTIOC1B Pin Function
0	0	0	0	MTU1.TGRB works as an output compare register	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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.TGRC compare match/input capture.

x: Don't care

Table 22.15 TIOR (MTU2)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB3	IOB2	IOB1	IOB0	MTU2.TGRB Function	MTIOC2B Pin Function
0	0	0	0	MTU2.TGRB works as an output compare register	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 22.16 TIORH (MTU3)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB3	IOB2	IOB1	IOB0	MTU3.TGRB Function	MTIOC3B Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 22.17 TIORL (MTU3)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD3	IOD2	IOD1	IOD0	MTU3.TGRD Function	MTIOC3D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 BFB bit in MTU3.TMDR is set to 1 and MTU3.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.18 TIORH (MTU4)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOB3	IOB2	IOB1	IOB0	MTU4.TGRB Function	MTIOC4B Pin Function
0	0	0	0	MTU4.TGRB works as an output compare register	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 22.19 TIORL (MTU4)

Bit 7	Bit 6	Bit 5	Bit 4	Description	
IOD3	IOD2	IOD1	IOD0	MTU4.TGRD Function	MTIOC4D Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 BFB bit in MTU4.TMDR is set to 1 and MTU4.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.20 TIORH (MTU0)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA3	IOA2	IOA1	IOA0	MTU0.TGRA Function	MTIOC0A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 count clock in MTU1. Input capture at MTU1.TCNT up-count/down-count.

x: Don't care

Table 22.21 TIORL (MTU0)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC3	IOC2	IOC1	IOC0	MTU0.TGRC Function	MTIOC0C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 count clock in MTU1. Input capture at MTU1.TCNT up-count/down-count.

x: Don't care

Note 1. When the BFA bit in MTU0.TMDR is set to 1 and MTU0.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.22 TIOR (MTU1)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA3	IOA2	IOA1	IOA0	MTU1.TGRA Function	MTIOC1A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 22.23 TIOR (MTU2)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA3	IOA2	IOA1	IOA0	MTU2.TGRA Function	MTIOC2A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 22.24 TIORH (MTU3)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA3	IOA2	IOA1	IOA0	MTU3.TGRA Function	MTIOC3A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 22.25 TIORL (MTU3)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC3	IOC2	IOC1	IOC0	MTU3.TGRC Function	MTIOC3C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 BFA bit in MTU3.TMDR is set to 1 and MTU3.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.26 TIORH (MTU4)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOA3	IOA2	IOA1	IOA0	MTU4.TGRA Function	MTIOC4A Pin Function
0	0	0	0	Output compare register	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 22.27 TIORL (MTU4)

Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC3	IOC2	IOC1	IOC0	MTU4.TGRC Function	MTIOC4C Pin Function
0	0	0	0	Output compare register*1	Output prohibited
0	0	0	1		Initial output is 0. 0 output at compare match.
0	0	1	0		Initial output is 0. 1 output at compare match.
0	0	1	1		Initial output is 0. Toggle output at compare match.
0	1	0	0		Output prohibited
0	1	0	1		Initial output is 1. 0 output at compare match.
0	1	1	0		Initial output is 1. 1 output at compare match.
0	1	1	1		Initial output is 1. 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 BFA bit in MTU4.TMDR is set to 1 and MTU4.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Table 22.28 TIORU, TIORV, and TIORW (MTU5)

Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Description	
IOC4	IOC3	IOC2	IOC1	IOC0	MTU5.TGRU, MTU5.TGRV, MTU5.TGRW Function	MTIC5U, MTIC5V, MTIC5W Pin Function
0	0	0	0	0	Compare match register	Compare match
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	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

### 22.2.4 Timer Compare Match Clear Register (TCNTCMPCLR)

Address(es): MTU5.TCNTCMPCLR 0008 88B6h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	CMPCLR5U	CMPCLR5V	CMPCLR5W
Value after reset:	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 is an 8-bit readable/writable register that specifies requests to clear MTU5.TCNTU, MTU5.TCNTV, and MTU5.TCNTW.

## 22.2.5 Timer Interrupt Enable Register (TIER)

- TIER (MTU0 to MTU4)

Address(es): MTU0.TIER 0008 8704h, MTU1.TIER 0008 8784h, MTU2.TIER 0008 8804h,  
MTU3.TIER 0008 8608h, MTU4.TIER 0008 8609h

b7	b6	b5	b4	b3	b2	b1	b0
TTGE	TTGE2	TCIEU	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 Converter Start Request Enable 2	0: A/D converter start request generation by MTU4.TCNT underflow (trough) disabled 1: A/D converter start request generation by MTU4.TCNT underflow (trough) enabled	R/W
b7	TTGE	A/D Converter Start Request Enable	0: A/D converter start request generation disabled 1: A/D converter start request generation enabled	R/W

The MTU has a total of seven TIER registers, two each for MTU0 and one each for MTU1 to MTU5. TIER is an 8-bit readable/writable register that enables or disables interrupt requests in each channel.

### TGIEA and TGIEB Bits (TGR Interrupt Enable A and B)

Each bit enables or disables interrupt requests (TGIn) (n = A or B).

### TGIEC and TGIED Bits (TGR Interrupt Enable C and D)

Each bit enables or disables interrupt requests (TGIn) in MTU0, MTU3 and MTU4 (n = C or 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 MTU1 and MTU2. In MTU0, MTU3, and MTU4, this bit is reserved. It is read as 0. The write value should be 0.

**TTGE2 Bit (A/D Converter Start Request Enable 2)**

This bit enables or disables generation of A/D converter start requests by MTU4.TCNT underflow (trough) in complementary PWM mode.

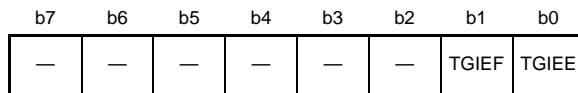
In MTU0 to MTU3, this bit is reserved. It is read as 0. The write value should be 0.

**TTGE Bit (A/D Converter Start Request Enable)**

This bit enables or disables generation of A/D converter start requests by TGRA input capture/compare match.

- TIER2 (MTU0)

Address(es): MTU0.TEIR2 0008 8724h



Value after reset: 0 0 0 0 0 0 0 0

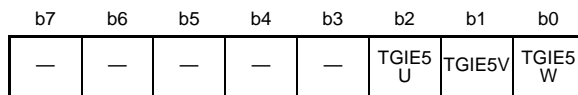
Bit	Symbol	Bit Name	Description	R/W
b0	TGIEE	TGR Interrupt Enable E	0: Interrupt requests (TGIE) disabled 1: Interrupt requests (TGIE) enabled	R/W
b1	TGIEF	TGR Interrupt Enable F	0: Interrupt requests (TGIF) disabled 1: Interrupt requests (TGIF) enabled	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	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.TGRm (m = E or F).

- TIER (MTU5)

Address(es): MTU5.TIER 0008 88B2h



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 TGI5W disabled 1: Interrupt requests TGI5W enabled	R/W
b1	TGIE5V	TGR Interrupt Enable 5V	0: Interrupt requests TGI5V disabled 1: Interrupt requests TGI5V enabled	R/W
b2	TGIE5U	TGR Interrupt Enable 5U	0: Interrupt requests TGI5U disabled 1: Interrupt requests TGI5U enabled	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

**TGIE5W, TGIE5V, and TGIE5U Bits (TGR Interrupt Enable 5m)**

Each bit enables or disables interrupt requests (TGI5m) (m = W, V, or U).

## 22.2.6 Timer Status Register (TSR)

- TSR (MTU0 to MTU4)

Address(es): MTU0.TSR 0008 8705h, MTU1.TSR 0008 8785h, MTU2.TSR 0008 8805h,  
MTU3.TSR 0008 862Ch, MTU4.TSR 0008 862Dh

	b7	b6	b5	b4	b3	b2	b1	b0
	TCFD	—	—	—	—	—	—	—

Value after reset: 1 1 x x x x x x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	—	Reserved	These bits are read as 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

The MTU has a total of five TSR registers, one each for MTU0 to MTU4.  
TSR is an 8-bit readable/writable register that indicates the status of each channel.

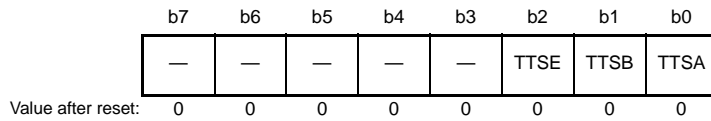
### TCFD Flag (Count Direction Flag)

Status flag that shows the direction in which TCNT counts in MTU1 to MTU4.  
In MTU0, this bit is reserved. It is read as 1. The write value should be 1.



### 22.2.7 Timer Buffer Operation Transfer Mode Register (TBTM)

Address(es): MTU0.TBTM 0008 8726h, MTU3.TBTM 0008 8638h, MTU4.TBTM 0008 8639h



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 in MTU0, 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

The MTU has a total of three TBTM registers, one each for MTU0, MTU3 and MTU4.

TBTM is an 8-bit readable/writable register that specifies the timing for transferring data from the buffer register to the timer general register in PWM mode.

#### 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 MTU0 is not set to PWM mode, do not set the TTSE bit to 1.

## 22.2.8 Timer Input Capture Control Register (TICCR)

Address(es): MTU1.TICCR 0008 8790h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	I2BE	I2AE	I1BE	I1AE

Value after reset: 0 0 0 0 0 0 0 0

x: Undefined

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 TMTIOC1B pin in the MTU2.TGRB input capture conditions 1: Includes the TMTIOC1B 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

The MTU has one TICCR for MTU1.

TICCR specifies input capture conditions when MTU1.TCNT and MTU2.TCNT are cascaded.

## 22.2.9 Timer A/D Converter Start Request Control Register (TADCR)

Address(es): MTU4.TADCR 0008 8640h

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	0: TCIV4 interrupt skipping is not linked 1: TCIV4 interrupt skipping is linked	R/W*1
b1	ITB3AE	TGIA3 Interrupt Skipping Link Enable	0: TGI3A interrupt skipping is not linked 1: TGI3A interrupt skipping is linked	R/W*1
b2	ITA4VE	TCIV4 Interrupt Skipping Link Enable	0: TCIV4 interrupt skipping is not linked 1: TCIV4 interrupt skipping is linked	R/W*1
b3	ITA3AE	TGIA3 Interrupt Skipping Link Enable	0: TGI3A interrupt skipping is not linked 1: TGI3A interrupt skipping is linked	R/W*1
b4	DT4BE	Down-Count TRG4BN Enable	0: A/D converter start requests (TRG4BN) disabled during MTU4.TCNT down-count operation 1: A/D converter start requests (TRG4BN) enabled during MTU4.TCNT down-count operation	R/W*1
b5	UT4BE	Up-Count TRG4BN Enable	0: A/D converter start requests (TRG4BN) disabled during MTU4.TCNT up-count operation 1: A/D converter start requests (TRG4BN) enabled during MTU4.TCNT up-count operation	R/W
b6	DT4AE	Down-Count TRG4AN Enable	0: A/D converter start requests (TRG4AN) disabled during MTU4.TCNT down-count operation 1: A/D converter start requests (TRG4AN) enabled during MTU4.TCNT down-count operation	R/W*1
b7	UT4AE	Up-Count TRG4AN Enable	0: A/D converter start requests (TRG4AN) disabled during MTU4.TCNT up-count operation 1: A/D converter start requests (TRG4AN) enabled during MTU4.TCNT up-count operation	R/W
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/B Transfer Timing Select	See Table 22.29 for details.	R/W

Note: • TADCR must not be accessed in 8-bit units; it should be accessed in 16-bit units.

Note: • When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), do not link A/D converter start requests with interrupt skipping operation (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR) to 0).

Note: • If link with interrupt skipping is enabled while interrupt skipping is disabled, A/D converter start requests will not be issued.

Note 1. Do not set any bit from among b6 and b4 to b0 to 1 unless complementary PWM mode is not selected.

TADCR is a 16-bit readable/writable register that enables or disables A/D converter start requests and specifies whether to link A/D converter start requests with interrupt skipping operation.

**Table 22.29 Setting of Transfer Timing by BF0[1:0] Bits**

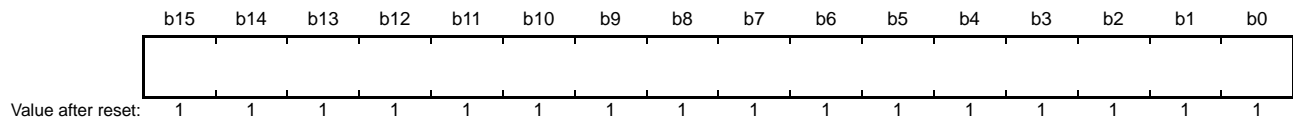
Bit 15	Bit 14	Description
BF1	BF0	
0	0	Does not transfer data from the cycle set buffer register to the cycle set register.
0	1	Transfers data from the cycle set buffer register to the cycle set register at the crest of the MTU4.TCNT count.*1
1	0	Transfers data from the cycle set buffer register to the cycle set register at the trough of the MTU4.TCNT count.*2
1	1	Transfers data from the cycle set buffer register to the cycle set register at the crest and trough of the MTU4.TCNT count.*2

Note 1. Data is transferred from the cycle set buffer register to the cycle set register when the crest of the MTU4.TCNT count is reached in complementary PWM mode, when a compare match occurs between MTU3.TCNT and MTU3.TGRA in reset-synchronized PWM mode, or when a compare match occurs between MTU4.TCNT and MTU4.TGRA in PWM mode 1 or normal mode.

Note 2. These settings are prohibited when complementary PWM mode is not selected.

### 22.2.10 Timer A/D Converter Start Request Cycle Set Registers A and B (TADCORA and TADCORB)

Address(es): MTU4.TADCORA 0008 8644h, MTU4.TADCORB 0008 8646h



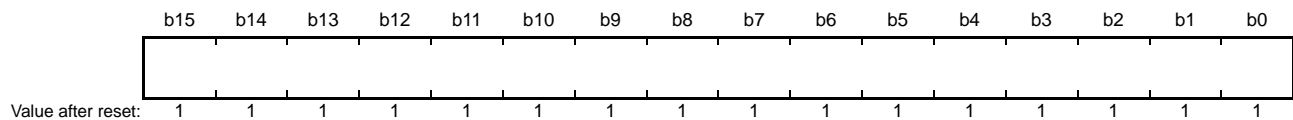
Note 1. TADCORA\_4 and TADCORB\_4 must not be accessed in 8-bit units; they should always be accessed in 16-bit units.

TADCORA and TADCORB are 16-bit readable/writable registers. When the MTU4.TCNT count reaches the value in TADCORA or TADCORB, a corresponding A/D converter start request will be issued.

The TADCORA and TADCORB values after reset are FFFFh.

### 22.2.11 Timer A/D Converter Start Request Cycle Set Buffer Registers A and B (TADCOBRA and TADCOBRB)

Address(es): MTU4.TADCOBRA 0008 8648h, MTU4.TADCOBRB 0008 864Ah



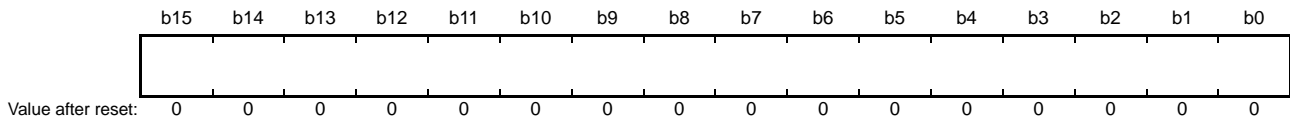
Note 1. TADCORA\_4 and TADCORB\_4 must not be accessed in 8-bit units; they should always be accessed in 16-bit units.

TADCOBRA and TADCOBRB are 16-bit readable/writable registers. When the crest or trough of the MTU4.TCNT count is reached, these register values are transferred to TADCORA and TADCORB, respectively.

The TADCOBRA and TADCOBRB values after reset are FFFFh.

### 22.2.12 Timer Counter (TCNT)

Address(es): MTU0.TCNT 0008 8706h, MTU1.TCNT 0008 8786h, MTU2.TCNT 0008 8806h, MTU3.TCNT 0008 8610h,  
MTU4.TCNT 0008 8612h, MTU5.TCNTU 0008 8880h, MTU5.TCNTV 0008 8890h, MTU5.TCNTW 0008 88A0h



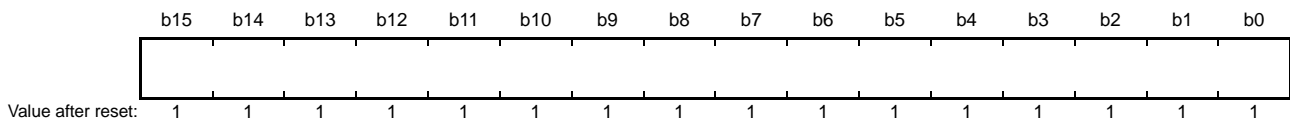
Note 1. The TCNT counters must not be accessed in 8-bit units; they should always be accessed in 16-bit units.

The MTU has a total of eight TCNT counters, one each for MTU0 to MTU4 and three (MTU5.TCNTU, TCNTV, and TCNTW) for MTU5. TCNT is a 16-bit readable/writable counter.

TCNT is initialized to 0000h by a reset.

### 22.2.13 Timer General Register (TGR)

Address(es): MTU0.TGRA 0008 8708h, MTU0.TGRB 0008 870Ah, MTU0.TGRC 0008 870Ch, MTU0.TGRD 0008 870Eh,  
MTU0.TGRE 0008 8720h, MTU0.TGRF 0008 8722h, MTU1.TGRA 0008 8788h, MTU1.TGRB 0008 878Ah,  
MTU2.TGRA 0008 8808h, MTU2.TGRB 0008 880Ah, MTU3.TGRA 0008 8618h, MTU3.TGRB 0008 861Ah,  
MTU3.TGRC 0008 8624h, MTU3.TGRD 0008 8626h, MTU4.TGRA 0008 861Ch, MTU4.TGRB 0008 861Eh,  
MTU4.TGRC 0008 8628h, MTU4.TGRD 0008 862Ah, MTU5.TGRU 0008 8882h, MTU5.TGRV 0008 8892h,  
MTU5.TGRW 0008 88A2h



Note 1. The TGR registers must not be accessed in 8-bit units; they should always be accessed in 16-bit units. TGR registers are initialized to FFFFh.

The MTU has a total of 21 TGR registers, six for MTU0, two each for MTU1 and MTU2, four each for MTU3 and MTU4, and three for MTU5. TGR is a 16-bit readable/writable register.

TGRA, TGRB, TGRC, and TGRD function as either output compare or input capture registers. TGRC and TGRD 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 converter start request can be issued. TGRF 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.

### 22.2.14 Timer Start Registers (TSTR)

- TSTR (MTU0 to MTU4)

Address(es): MTU.TSTR 0008 8680h

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 performs count stop 1: MTU0.TCNT performs count operation	R/W
b1	CST1	Counter Start 1	0: MTU1.TCNT performs count stop 1: MTU1.TCNT performs count operation	R/W
b2	CST2	Counter Start 2	0: MTU2.TCNT performs count stop 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 performs count stop 1: MTU3.TCNT performs count operation	R/W
b7	CST4	Counter Start 4	0: MTU4.TCNT performs count stop 1: MTU4.TCNT performs count operation	R/W

TSTR starts or stops TCNT operation in MTU0 to MTU4.

Before setting the operating mode in TMDR or setting the TCNT count clock in TCR, be sure to stop the TCNT counter.

#### CSTn Bits (Counter Start n) (n = 0 to 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 but 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.

- TSTR (MTU5)

Address(es): MTU5.TSTR 0008 88B4h

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 count operation is stopped 1: MTU5.TCNTW performs count operation	R/W
b1	CSTV5	Counter Start V5	0: MTU5.TCNTV count operation is stopped 1: MTU5.TCNTV performs count operation	R/W
b2	CSTU5	Counter Start U5	0: MTU5.TCNTU count operation 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

## 22.2.15 Timer Synchronous Registers (TSYR)

Address(es): MTU.TSYR 0008 8681h

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 presetting/clearing is not related to other channels). 1: MTU0.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled.	R/W
b1	SYNC1	Timer Synchronous Operation 1	0: MTU1.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU1.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled.	R/W
b2	SYNC2	Timer Synchronous Operation 2	0: MTU2.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU2.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled.	R/W
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 presetting/clearing is not related to other channels). 1: MTU3.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled.	R/W
b7	SYNC4	Timer Synchronous Operation 4	0: MTU4.TCNT operates independently (TCNT presetting/clearing is not related to other channels). 1: MTU4.TCNT performs synchronous operation. TCNT synchronous presetting/synchronous clearing is enabled.	R/W

TSYR 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 n Operation) (n = 0 to 4)

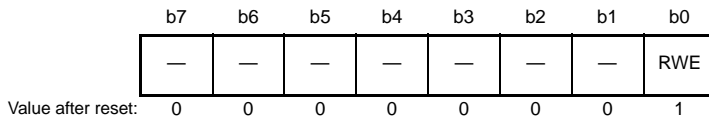
Each bit selects whether operation is independent of or synchronized with other channels.

When synchronous operation is selected, the TCNT synchronous presetting of multiple channels and synchronous clearing by counter clearing on another channel are possible.

To set synchronous operation, the SYNCn bits for at least two channels must be set to 1. To set synchronous clearing, in addition to the SYNCn bit, the TCNT clearing source must also be set by means of TCR.CCLR[2:0] bits.

### 22.2.16 Timer Read/Write Enable Registers (TRWER)

Address(es): MTU.TRWER 0008 8684h



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

TRWER 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 conditions]

- When 0 is written to the RWE bit after reading RWE bit = 1
- Registers and Counters having Write-Protection Capability against Accidental Modification  
22 registers: MTUm.TCR, MTUm.TMDR, MTUm.TIORH, MTUm.TIORL, MTUm.TIER, MTUm.TGRA, MTUm.TGRB, TOER, TOCR1, TOCR2, TGCR, TCDR, TDDR, and MTUm.TCNT (m = 3, 4)



## 22.2.17 Timer Output Master Enable Registers (TOER)

Address(es): MTU.TOER 0008 860Ah

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 (inactive level)*1 1: MTU output is enabled	R/W
b1	OE4A	Master Enable MTIOC4A	0: MTU output is disabled (inactive level)*1 1: MTU output is enabled	R/W
b2	OE4B	Master Enable MTIOC4B	0: MTU output is disabled (inactive level)*1 1: MTU output is enabled	R/W
b3	OE3D	Master Enable MTIOC3D	0: MTU output is disabled (inactive level)*1 1: MTU output is enabled	R/W
b4	OE4C	Master Enable MTIOC4C	0: MTU output is disabled (inactive level)*1 1: MTU output is enabled	R/W
b5	OE4D	Master Enable MTIOC4D	0: MTU output is disabled (inactive level)*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. The inactive level is determined by the settings in timer output control registers 1 and 2 (TOCR1 and TOCR2). For details, refer to section 22.2.18, Timer Output Control Registers 1 (TOCR1), and section 22.2.19, Timer Output Control Registers 2 (TOCR2). Set these bits to 1 to enable MTU output when complementary PWM or reset-synchronized PWM mode is not selected. When these bits are set to 0, the inactive level is output by setting the timer output control registers 1 and 2 (TOCR1 and TOCR2).

TOER enables or disables output settings for output pins MTIOC4D, MTIOC4C, MTIOC3D, MTIOC4B, MTIOC4A, and MTIOC3B.

These pins do not output correctly if the TOER bits have not been set. In MTU3 and MTU4, set TOER prior to setting TIOR.

Set TOER after clearing the CST3 and CST4 bits in TSTR to 0 (see Figure 22.35).

## 22.2.18 Timer Output Control Registers 1 (TOCR1)

Address(es): MTU.TOCR 0008 860Eh

	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*2, *3	See Table 22.30.	R/W
b1	OLSN	Output Level Select N*2, *3	See Table 22.31.	R/W
b2	TOCS	TOC Select	0: TOCR1 setting is selected 1: TOCR2 setting is selected	R/W
b3	TOCL	TOC Register Write Protection*1	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*4
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 TOCR1.TOCL bit to 1 prevents accidental modification when the CPU goes out of control.

Note 2. Clearing the TOCR1.TOCS bit to 0 makes this bit setting valid.

Note 3. If dead-time is not generated, the negative-phase output is always 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 power on reset. After 1 is written, 0 cannot be written to the bit.

TOCR1 is 8-bit readable/writable registers that enable or disable PWM-synchronized toggle output in complementary PWM mode and reset-synchronized PWM mode, and control inversion of PWM output level.

### OLSP Bit (Output Level Select P)

This bit selects the positive-phase output level in reset-synchronized PWM mode and complementary PWM mode.

### OLSN Bit (Output Level Select N)

This bit selects the negative-phase output level in reset-synchronized PWM mode and complementary PWM mode.

### TOCS Bit (TOC Select)

This bit selects either the TOCR1 or TOCR2 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 TOCR1.

### PSYE Bit (PWM Synchronous Output Enable)

This bit enables or disables toggle output synchronized with the PWM cycle.

**Table 22.30 Output Level Select Function**

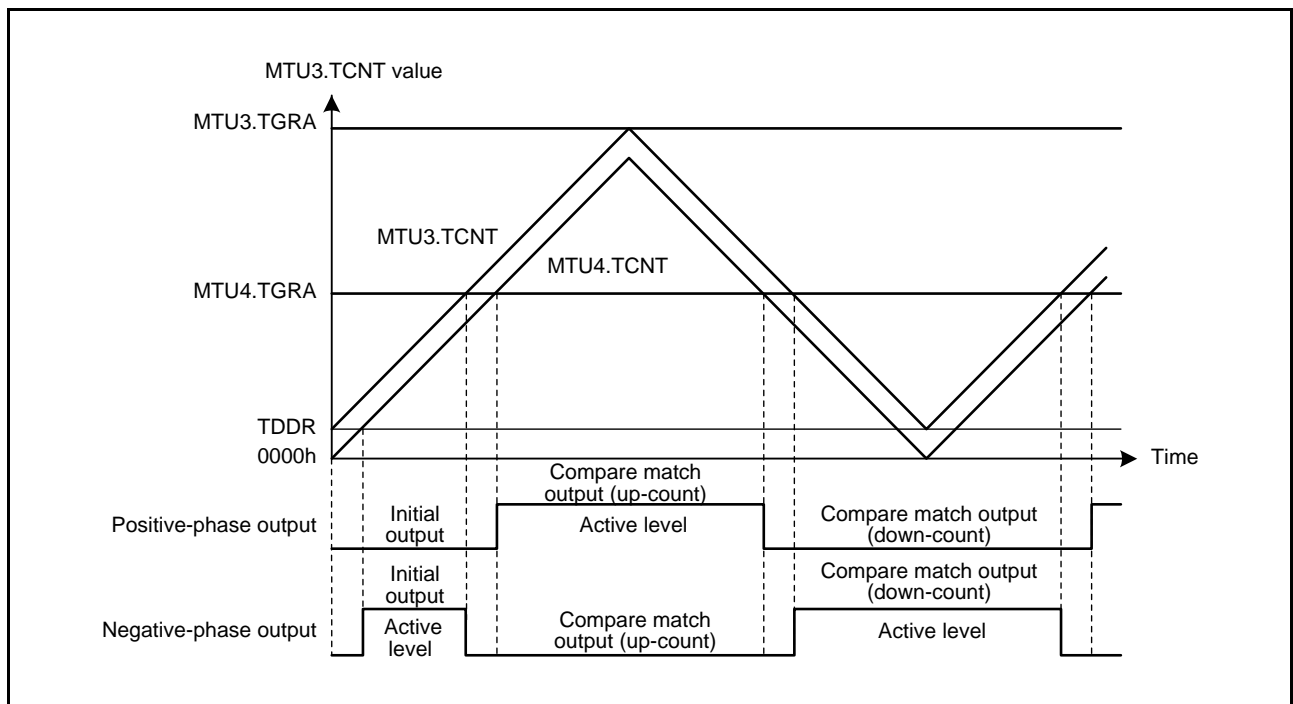
Bit 0	Function			
OLSP	Initial Output	Active Level	Compare Match Output	
			Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

**Table 22.31 Output Level Select Function**

Bit 1	Function			
OLSN	Initial Output	Active Level	Compare Match Output	
			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.

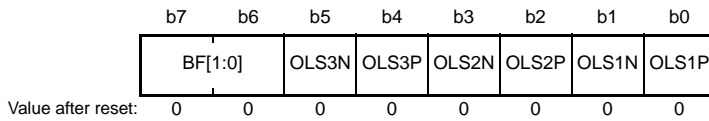
Figure 22.2 shows an example of output in complementary PWM mode (one phase) when OLSN = 1 and OLSP = 1.



**Figure 22.2 Example of Output in Complementary PWM Mode**

### 22.2.19 Timer Output Control Registers 2 (TOCR2)

Address(es): MTU.TOCR2 0008 860Fh



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. See Table 22.32.	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. See Table 22.33.	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. See Table 22.34.	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. See Table 22.35.	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. See Table 22.36.	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. See Table 22.37.	R/W
b7, b6	BF[1:0]	TOLBR Buffer Transfer Timing Select	These bits select the timing for transferring data from TOLBR to TOCR2. See Table 22.38 for details.	R/W

Note 1. Setting the TOCR1.TOCS bit to 1 makes this bit setting valid.

Note 2. If dead-time is not generated, the negative-phase output is always the exact inverse of the positive-phase output. In these cases, only the OLSiP bits are valid (i = 1 to 3).

TOCR2 control inversion of PWM output level in complementary PWM mode and reset-synchronized PWM mode.

**Table 22.32 MTIOC3B Output Level Select Function**

Bit 0	Function			
OLS1P	Initial Output	Active Level	Compare Match Output	
			Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

**Table 22.33 MTIOC3D Output Level Select Function**

Bit 1	Function			
OLS1N	Initial Output	Active Level	Compare Match Output	
			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.

**Table 22.34 MTIOC4A Output Level Select Function**

Bit 2	Function			
OLS2P	Initial Output	Active Level	Compare Match Output	
			Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

**Table 22.35 MTIOC4C Output Level Select Function**

Bit 3	Function			
OLS2N	Initial Output	Active Level	Compare Match Output	
			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.

**Table 22.36 MTIOC4B Output Level Select Function**

Bit 4	Function			
OLS3P	Initial Output	Active Level	Compare Match Output	
			Up-Counting	Down-Counting
0	High level	Low level	Low level	High level
1	Low level	High level	High level	Low level

**Table 22.37 MTIOC4D Output Level Select Function**

Bit 5	Function			
OLS3N	Initial Output	Active Level	Compare Match Output	
			Up-Counting	Down-Counting
0	High level	Low level	High level	Low level
1	Low level	High level	Low level	High level

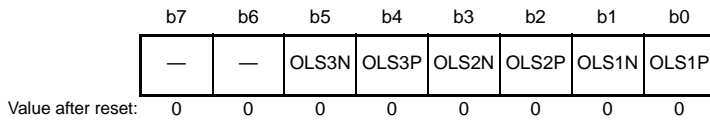
Note: • The initial output value of negative-phase waveform changes to an active level after the dead time has passed since counting starts.

**Table 22.38 Setting of TOCR2.BF[1:0] Bits**

Bit 7	Bit 6	Description	
BF1	BF0	Complementary PWM Mode	Reset-Synchronized PWM Mode
0	0	Does not transfer data from the buffer register (TOLBR) to TOCR2.	Does not transfer data from the buffer register (TOLBR) to TOCR2.
0	1	Transfers data from the buffer register (TOLBR) to TOCR2 at the crest of the MTU4.TCNT count.	Transfers data from the buffer register (TOLBR) to TOCR2 when MTU4.TCNT or MTU3.TCNT is cleared.
1	0	Transfers data from the buffer register (TOLBR) to TOCR2 at the trough of the MTU4.TCNT count.	Setting prohibited
1	1	Transfers data from the buffer register (TOLBR) to TOCR2 at the crest and trough of the MTU4.TCNT count.	Setting prohibited

### 22.2.20 Timer Output Level Buffer Registers (TOLBR)

Address(es): MTU.TOLBR 0008 8636h



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 TOCR2.	R/W
b1	OLS1N	Output Level Select 1N	Specify the buffer value to be transferred to the OLS1N bit in TOCR2.	R/W
b2	OLS2P	Output Level Select 2P	Specify the buffer value to be transferred to the OLS2P bit in TOCR2.	R/W
b3	OLS2N	Output Level Select 2N	Specify the buffer value to be transferred to the OLS2N bit in TOCR2.	R/W
b4	OLS3P	Output Level Select 3P	Specify the buffer value to be transferred to the OLS3P bit in TOCR2.	R/W
b5	OLS3N	Output Level Select 3N	Specify the buffer value to be transferred to the OLS3N bit in TOCR2.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TOLBR is 8-bit readable/writable registers that function as buffer registers for TOCR2 and specify the PWM output level in complementary PWM mode and reset-synchronized PWM mode.

Figure 22.3 shows an example of the PWM output level setting procedure in buffer operation.

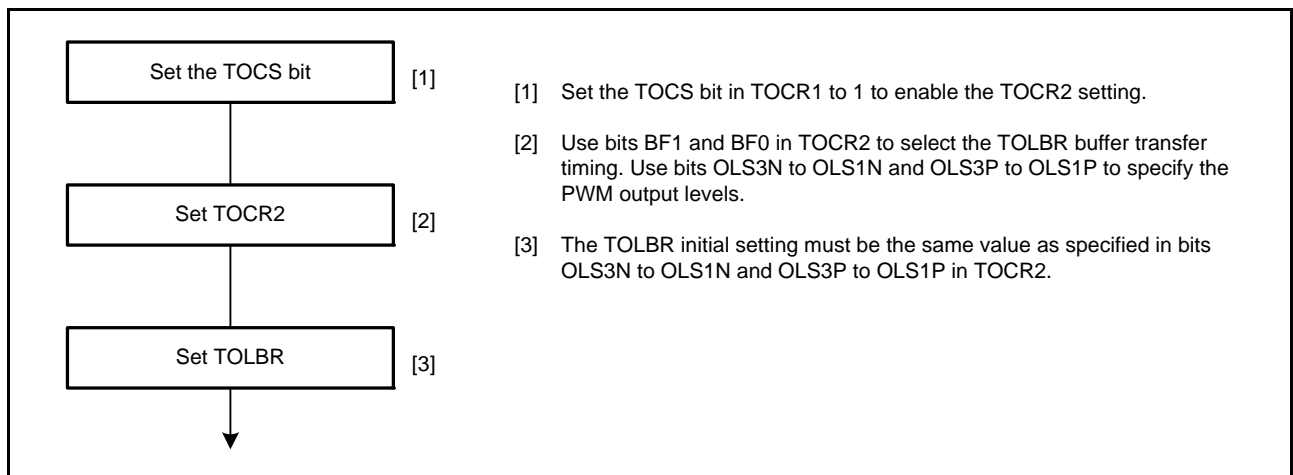


Figure 22.3 Example of PWM Output Level Setting Procedure in Buffer Operation

### 22.2.21 Timer Gate Control Registers (TGCR)

Address(es): MTU.TGCR 0008 860Dh

	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 TGCR.FB bit s set to 1.	R/W
b1	VF		In this case, the setting of b0 to b2 is used instead of the external input. See Table 22.39.	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 (TGCR'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

TGCR control the output waveform necessary for brushless DC motor control in reset-synchronized PWM mode and complementary PWM mode. These 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 TGCR.FB bit is set to 1. In this case, the setting of b0 to b2 is used instead of the external input. See Table 22.39.

#### 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 TGCR.

#### 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)

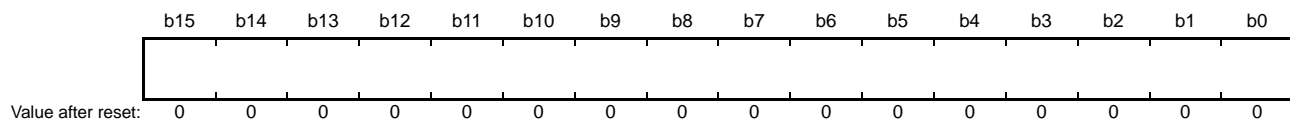
This bit selects whether to make the functions of TGCR effective or ineffective.

**Table 22.39 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

### 22.2.22 Timer Subcounters (TCNTS)

Address(es): MTU.TCNTS 0008 8620h

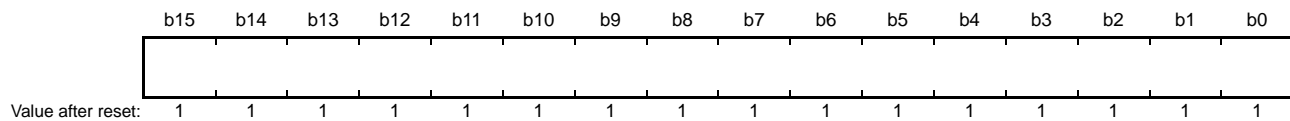


Note: • Accessing the TCNTS in 8-bit units is prohibited. Always access in 16-bit units.

TCNTS is 16-bit read-only counters that are used only in complementary PWM mode. The TCNTS value after reset is 0000h.

### 22.2.23 Timer Dead Time Data Registers (TDDR)

Address(es): MTU.TDDR 0008 8616h



Note: • Accessing the TDDR in 8-bit units is prohibited. Always access in 16-bit units.

TDDR is 16-bit registers, 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 TDDR value is loaded into the MTU3.TCNT counter and the count operation starts. The TDDR value after reset is FFFFh.



### 22.2.24 Timer Cycle Data Registers (TCDR)

Address(es): MTU.TCDR 0008 8614h

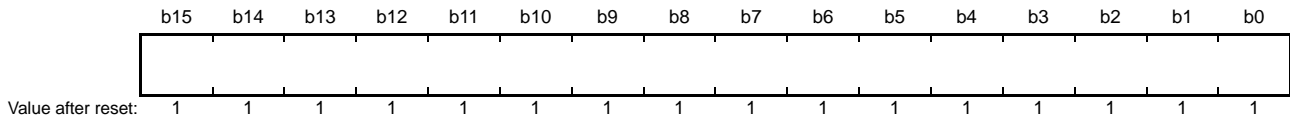


Note: • Accessing the TCDR in 8-bit units is prohibited. Always access in 16-bit units.

TCDR is 16-bit registers used only in complementary PWM mode. Set half the PWM carrier cycle as the TCDR value. TCDR is constantly compared with the TCNTS counter in complementary PWM mode, and when a match occurs, the TCNTS counter switches direction (down-count to up-count). The TCDR value after reset is FFFFh.

### 22.2.25 Timer Cycle Buffer Registers (TCBR)

Address(es): MTU.TCBR 0008 8622h

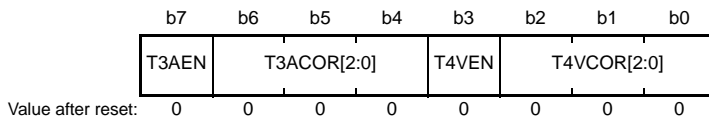


Note: • Accessing the TCBR in 8-bit units is prohibited. Always access in 16-bit units.

TCBR is 16-bit registers, used only in complementary PWM mode, that function as buffer registers for TCDR. The TCBR value is transferred to TCDR with the transfer timing set in TMDR. The TCBR value after reset is FFFFh.

### 22.2.26 Timer Interrupt Skipping Set Registers (TITCR)

Address(es): MTU.TITCR 0008 8630h



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, see Table 22.40.	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, see Table 22.41.	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 clear the TITCR.T3AEN and TITCR.T4VEN bits to 0 to clear the skipping counter (TITCNT).

**T4VCOR[2:0] Bits (TCIV4 Interrupt Skipping Count Setting)****T3ACOR[2:0] Bits (TGIA3 Interrupt Skipping Count Setting)**

These bits specify the TCIV3 and TGIA4 interrupt skipping count within the range from 0 to 7. For details, see Table 22.40 and Table 22.41.

**Table 22.40 Setting of Interrupt Skipping Count by T4VCOR[2:0] Bits**

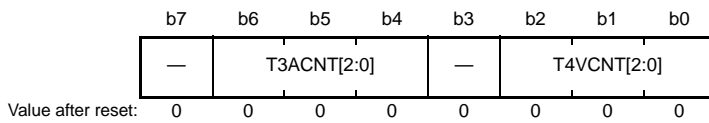
Bit 2	Bit 1	Bit 0	
T4VCOR2	T4VCOR1	T4VCOR0	Description
0	0	0	Does not perform TCIV4 interrupt skipping.
0	0	1	Sets the TCIV4 interrupt skipping count to 1.
0	1	0	Sets the TCIV4 interrupt skipping count to 2.
0	1	1	Sets the TCIV4 interrupt skipping count to 3.
1	0	0	Sets the TCIV4 interrupt skipping count 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 22.41 Setting of Interrupt Skipping Count by T3ACOR[2:0] Bits**

Bit 6	Bit 5	Bit 4	
T3ACOR2	T3ACOR1	T3ACOR0	Description
0	0	0	Does not perform TGIA3 interrupt skipping.
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.

### 22.2.27 Timer Interrupt Skipping Counters (TITCNT)

Address(es): MTU.TITCNT 0008 8631h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	T4VCNT[2:0]	TCIV4 Interrupt Counter	While the T4VEN bit in TITCR 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. Writing to this bit has no effect.	R
b6 to b4	T3ACNT[2:0]	TGIA3 Interrupt Counter	While the T3AEN bit in TITCR 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. Writing to this bit has no effect.	R

Note: • To clear the TITCNT, clear the T3AEN and T4VEN bits in TITCR to 0.

TITCNT is 8-bit readable/writable counters. The MTU2 has two TITCNT counter. TITCNT retain their values even after stopping the count operation of MTU4.TCNT and MTU3.TCNT.

#### T4VCNT[2:0] Bits (TCIV4 Interrupt Counter)

[Clearing conditions]

- When the T4VCNT[2:0] bits in TITCNT match the T4VCR[2:0] bits in TITCR
- When the T4VEN bit in TITCR is cleared to 0
- When the T4VCOR[2:0] bits in TITCR are cleared to 000b

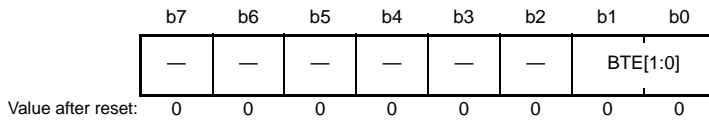
#### T3ACNT[2:0] Bits (TGIA3 Interrupt Counter)

[Clearing conditions]

- When the T3ACNT[2:0] bits in TITCNT match the T3ACOR[2:0] bits in TITCR
- When the T3AEN bit in TITCR is cleared to 0
- When the TITCR.T3ACOR[2:0] bits are cleared to 000b

## 22.2.28 Timer Buffer Transfer Set Registers (TBTER)

Address(es): MTU.TBTER 0008 8632h



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 used in complementary PWM mode to the temporary registers and specify whether to link the transfer with interrupt skipping operation. See Table 22.42 for details.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TBTER is 8-bit readable/writable registers that enable or disable transfer from the buffer registers used in complementary PWM mode to the temporary registers and specify whether to link the transfer with interrupt skipping operation.

**Table 22.42 Setting of TBTER.BTE[1:0] Bits**

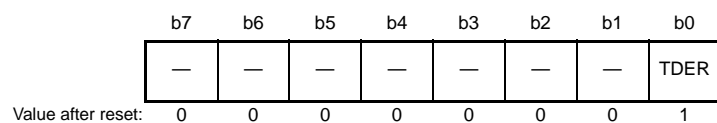
Bit 1	Bit 0	Description
BTE1	BTE0	Description
0	0	Enables transfer from the buffer registers to the temporary registers* <sup>1</sup> and does not link the transfer with interrupt skipping operation.
0	1	Disables transfer from the buffer registers to the temporary registers.
1	0	Links transfer from the buffer registers to the temporary registers with interrupt skipping operation.* <sup>2</sup>
1	1	Setting prohibited

Note 1. Data is transferred according to the MD3 to MD0 bit setting in TMDR. For details, refer to section 22.3.8, Complementary PWM Mode.

Note 2. When interrupt skipping is disabled (the T3AEN and T4VEN bits are cleared to 0 in the timer interrupt skipping set register (TITCR) or the skipping count set bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), be sure to disable link of buffer transfer with interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TBTER) to 0). If link with interrupt skipping is enabled while interrupt skipping is disabled, buffer transfer will not be performed

### 22.2.29 Timer Dead Time Enable Registers (TDER)

Address(es): MTU.TDER 0008 8634h



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. TDDR must be set to 1 or a larger value.

TDER is 8-bit readable/writable registers that control dead time generation in complementary PWM mode. The MTU3 has one TDER register. TDER 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 bit after reading TDER bit = 1

### 22.2.30 Timer Waveform Control Registers (TWCR)

Address(es): MTU.TWCR 0008 8660h

	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	Initial Output Inhibition Enable	0: Initial value specified in TOCR is output 1: Initial output is inhibited	R/(W) *1
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	CCE	Compare Match Clear Enable	0: Counters are not cleared at MTU3.TGRA compare match 1: Counters are cleared at MTU3.TGRA compare match	R/(W) *2

Note 1. Do not set this bit to 1 unless complementary PWM mode is selected.

Note 2. Do not set this bit to 1 unless complementary PWM mode 1 is selected.

TWCR is 8-bit readable/writable registers. TWCR controls the output waveform when synchronous counter clearing occurs in MTU3.TNCT and MTU4.TNCT in complementary PWM mode and specifies whether to clear the counters at MTU3.TGRA compare match.

The CCE bit and WRE bit in TWCR should be modified only while TCNT stops.

#### WRE Bit (Initial Output Inhibition Enable)

This bit selects the waveform output when synchronous counter clearing occurs in complementary PWM mode.

The initial output is prohibited only when synchronous clearing occurs within the  $T_b$  interval at the trough in complementary PWM mode. When synchronous clearing occurs outside this interval, the initial value specified in TOCR is output regardless of the WRE bit setting. The initial value specified in TOCR is also output when synchronous clearing occurs in the  $T_b$  interval at the trough immediately after MTU3.TCNT and MTU4.TCNT start operation.

For the  $T_b$  interval at the trough in complementary PWM mode, see Figure 22.40.

[Setting condition]

- When 1 is written to WRE bit after reading WRE bit = 0

#### CCE Bit (Compare Match Clear Enable)

This bit specifies whether to clear counters at TGRA3 compare match in complementary PWM mode.

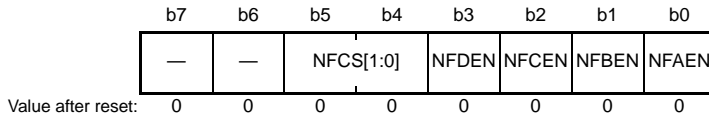
[Setting condition]

- When 1 is written to CCE bit after reading CCE bit = 0

### 22.2.31 Noise Filter Control Registers (NFCR)

- NFCR (MTU0 to MTU4)

Address(es): MTU0.NFCR 0008 8690h, MTU1.NFCR 0008 8691h, MTU2.NFCR 0008 8692h,  
MTU3.NFCR 0008 8693h, MTU4.NFCR 0008 8694h



Bit	Symbol	Bit Name	Description	R/W
b0	NFAEN	Noise Filter A Enable Bit	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 Bit	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 Bit	0: The noise filter for the MTIOCnC pin is disabled. 1: The noise filter for the MTIOCnC pin is enabled.	R/W *1
b3	NFDEN	Noise Filter D Enable Bit	0: The noise filter for the MTIOCnD pin is disabled. 1: The noise filter for the MTIOCnD pin is enabled.	R/W *1
b5, b4	NFCS[1:0]	Noise Filter Clock Select	00: PCLK/1 01: PCLK/8 10: PCLK/32 11: The clock source for counting is the external clock.	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 the NFCRs for MTU1 and MTU2. These bits are read as 0, and writing to them is not possible.

MTUn.NFCR is 8-bit readable and writable register (n = 0 to 4). These registers control enabling and disabling of the noise filters for the MTIOCnm (n = 0 to 4; m = A to D) pins and sets the sampling clocks for the noise filters.

#### NFAEN Bit (Noise Filter A Enable)

This bit disables or enables the noise filter for input from the MTIOCnA pin. Since unexpected edges may be internally generated when the value of NFAEN is changed, select the output compare function in the timer I/O control register and set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before changing the value.

#### NFBEN Bit (Noise Filter B Enable)

This bit disables or enables the noise filter for input from the MTIOCnB pin. Since unexpected edges may be internally generated when the value of NFBEN is changed, select the output compare function in the timer I/O control register and set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before changing the value.

#### NFCEN Bit (Noise Filter C Enable)

This bit disables or enables the noise filter for input from the MTIOCnC pin. Since unexpected edges may be internally generated when the value of NFCEN is changed, select the output compare function in the timer I/O control register and set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before changing the value.

#### NFDEN Bit (Noise Filter D Enable)

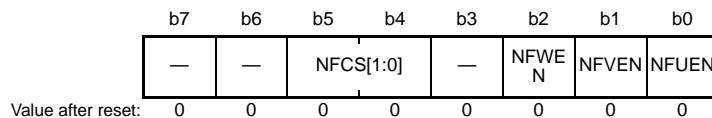
This bit disables or enables the noise filter for input from the MTIOCnD pin. Since unexpected edges may be internally generated when the value of NFDEN is changed, select the output compare function in the timer I/O control register and set the TMDR.MD[3:0] bits to a value other than that for normal mode (0000b) before changing the value.

**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, selecting the external clock as the source to drive counting, wait for two cycles of the external clock before setting the input-capture function.

- NFCR (MTU5)

Address(es): MTU5.NFCR 0008 8695h



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	00: PCLK/1 01: PCLK/8 10: PCLK/32 11: The clock source for counting is the external clock.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

MTU5.NFCR is 8-bit readable and writable register. This register controls enabling and disabling of the noise filters for the MTIC5m (m = U, V, W) pins and sets the sampling clock for the noise filters.

**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 NFUEN is changed, select the compare-match function in the timer I/O control register before changing the value.

**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 NFVEN is changed, select the compare-match function in the timer I/O control register before changing the value.

**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 NFWEN is changed, select the compare-match function in the timer I/O control register before changing the value.

**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.



## 22.2.32 Bus Master Interface

The timer counters (TCNT), general registers (TGR), timer subcounter (TCNTS), timer cycle buffer register (TCBR), timer dead time data register (TDDR), timer cycle data register (TCDR), timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (TADCORA/B), and timer A/D converter start request cycle set buffer registers (TADCOBRA/B) are 16-bit registers. A 16-bit data bus to the bus master enables 16-bit read/write access. 8-bit read/write is not allowed. Always access the registers in 16-bit units.

All registers other than the above registers are 8-bit registers, so read/write access should be performed in 8-bit units.

## 22.3 Operation

### 22.3.1 Basic Functions

Each channel has TCNT and TGR. TCNT performs up-counting, and is also capable of free-running operation, periodic counting, and external event counting.

Each TGR can be used as an input capture register or an output compare register.

#### (1) Counter Operation

When one of bits CST0 to CST4 in TSTR or bits CSTU5, CSTV5, and CSTW5 in MTU5.TSTR 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 22.4 shows an example of the count operation setting procedure.

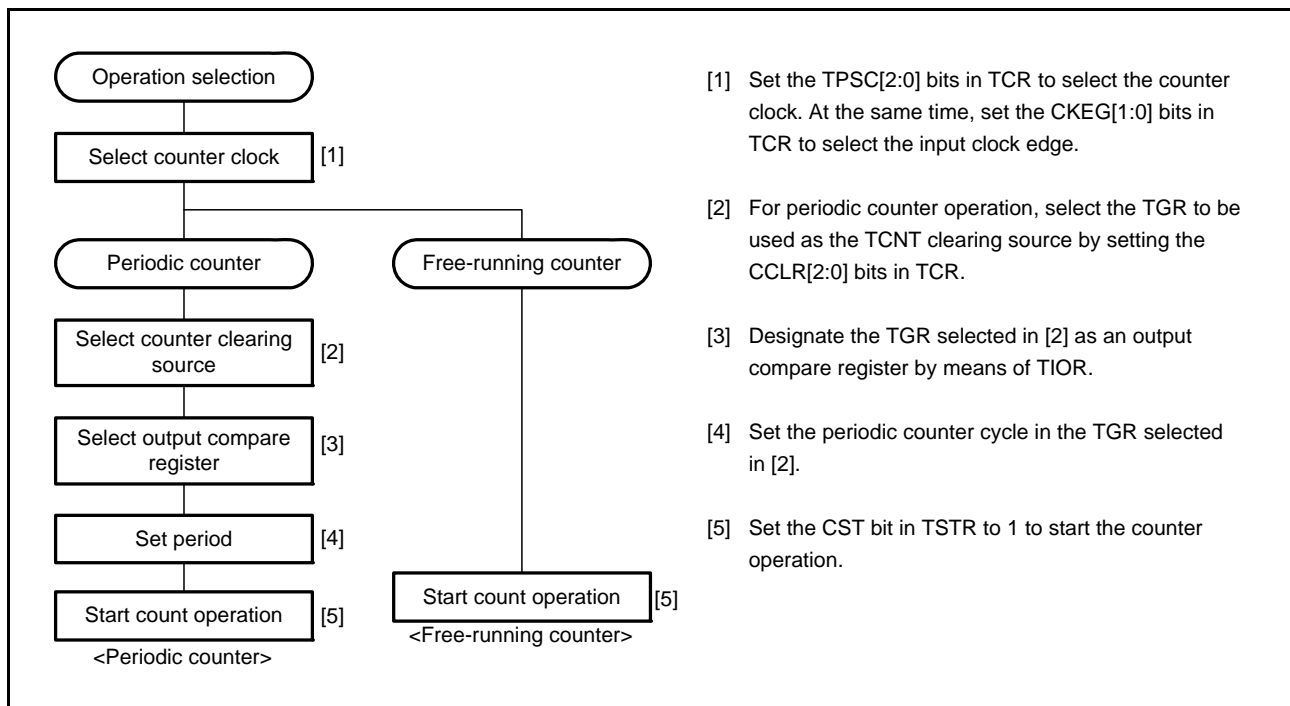


Figure 22.4 Example of Counter Operation Setting Procedure

### (b) Free-Running Count Operation and Periodic Count Operation

Immediately after a reset, the MTU's TCNT counters are all designated as free-running counters. When the relevant bit in 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), the MTU requests an interrupt if the corresponding TCIEV bit in TIER is 1. After an overflow, TCNT starts counting up again from 0000h.

Figure 22.5 illustrates free-running counter operation.

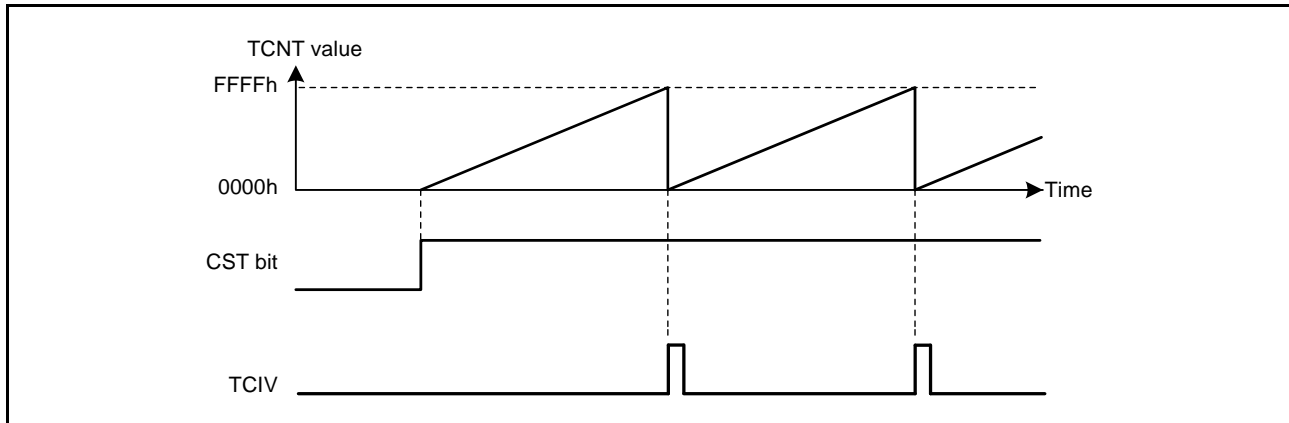


Figure 22.5 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, TCNT for the relevant channel performs periodic count operation. TGR for setting the cycle is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR[2:0] in TCR. After the settings have been made, TCNT starts up-count operation as a periodic counter when the corresponding bit in TSTR is set to 1. When the count matches the value in TGR, TCNT is cleared to 0000h.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the MTU requests an interrupt. After a compare match, TCNT starts counting up again from 0000h.

Figure 22.6 illustrates periodic counter operation.

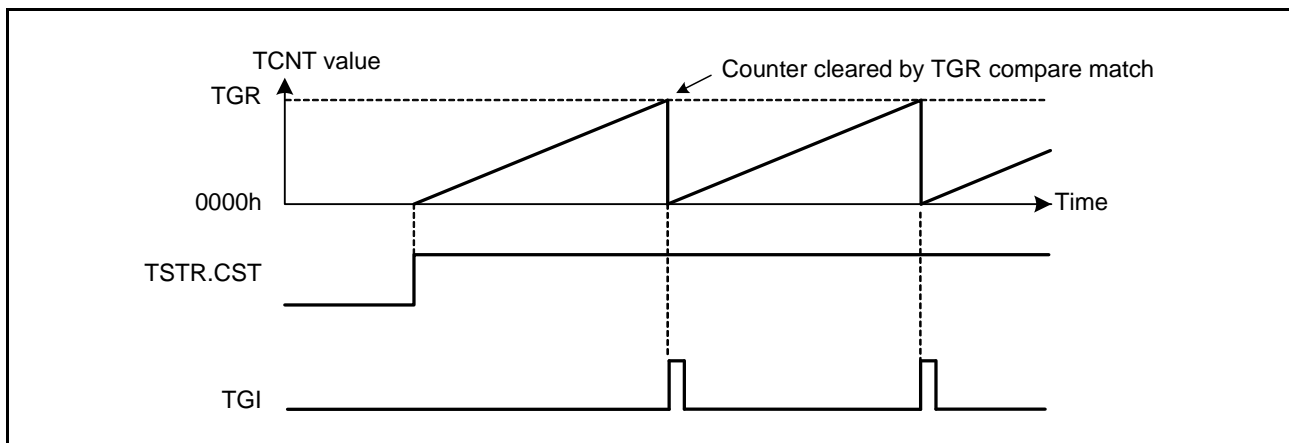


Figure 22.6 Periodic Counter Operation

(2) Waveform Output by Compare Match

The MTU can be set to output 0, output 1, or toggle the level on the corresponding output pin in response to compare matches.

(a) Example of Procedure for Setting Waveform Output by Compare Match

Figure 22.7 shows an example of the procedure for setting waveform output by compare match

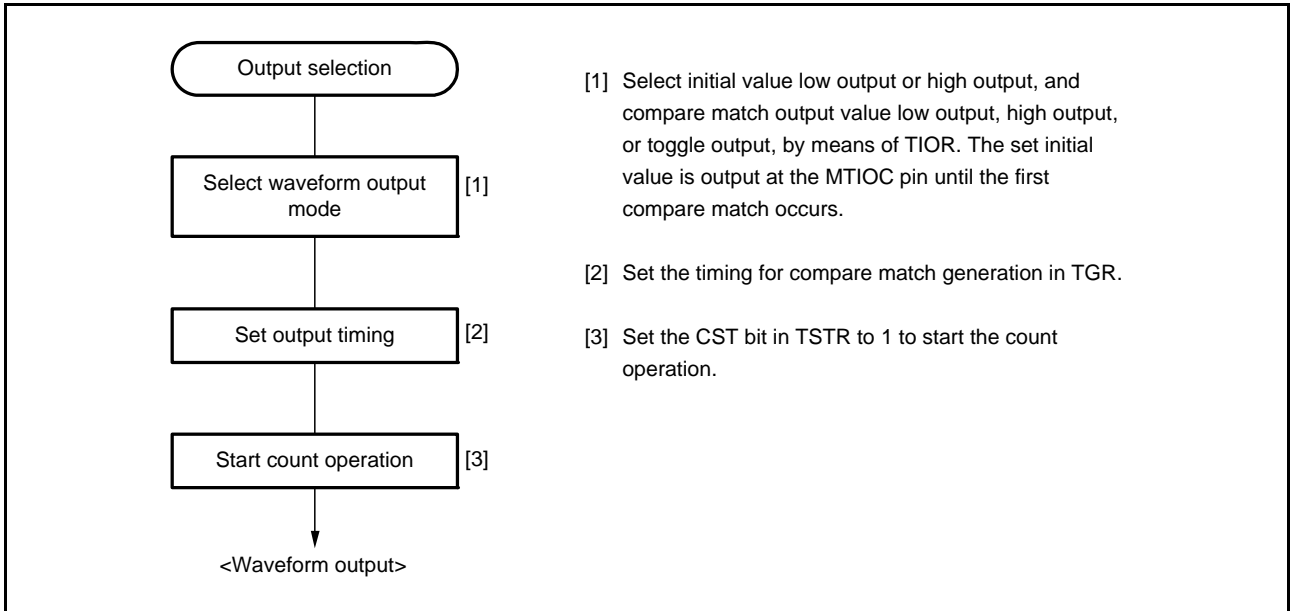


Figure 22.7 Example of Procedure for Setting Waveform Output by Compare Match

(b) Examples of Waveform Output Operation

Figure 22.8 shows an example of the output of 0 and of 1 in response to compare matches.

In this example, TCNT has been designated as a free-running counter, and settings have been made so that 1 is output in response to compare match A and 0 is output in response to compare match B. When the pin level is the same as the specified level, the pin level does not change.

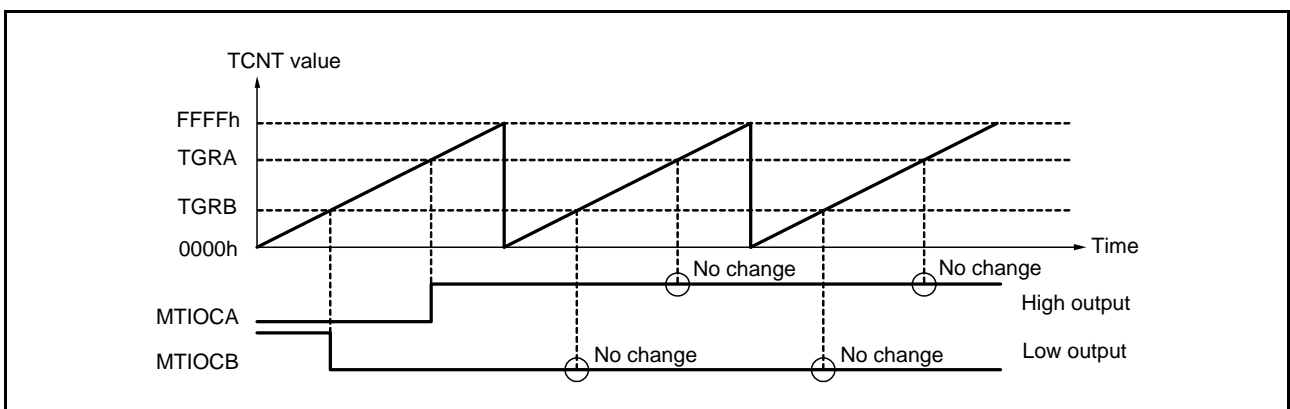


Figure 22.8 Example of Operation in the Output of 0 and of 1

Figure 22.9 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.

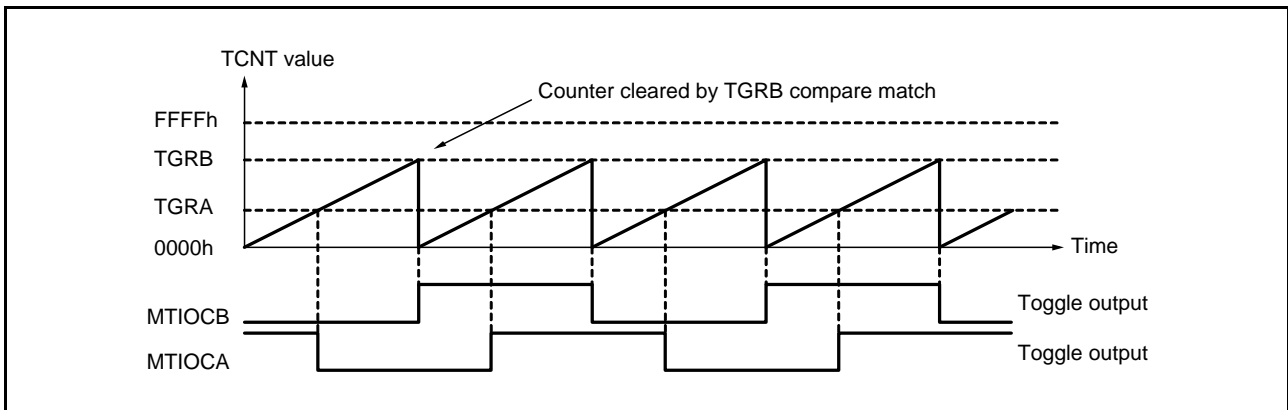


Figure 22.9 Example of Toggle Output Operation

### (3) Input Capture Function

The TCNT value can be transferred to TGR on detection of the MTIOC pin input edge.

The rising edge, falling edge, or both edges can be selected as the detection edge. For MTU0 and MTU1, another channel's counter input clock or compare match signal can also be specified as the input capture source.

Note: • When another channel's counter input clock is used as the input capture input for MTU0 and MTU1, PCLK/1 should not be selected as the counter input clock used for input capture input. Input capture will not be generated if PCLK/1 is selected.

#### (a) Example of Input Capture Operation Setting Procedure

Figure 22.10 shows an example of the input capture operation setting procedure.

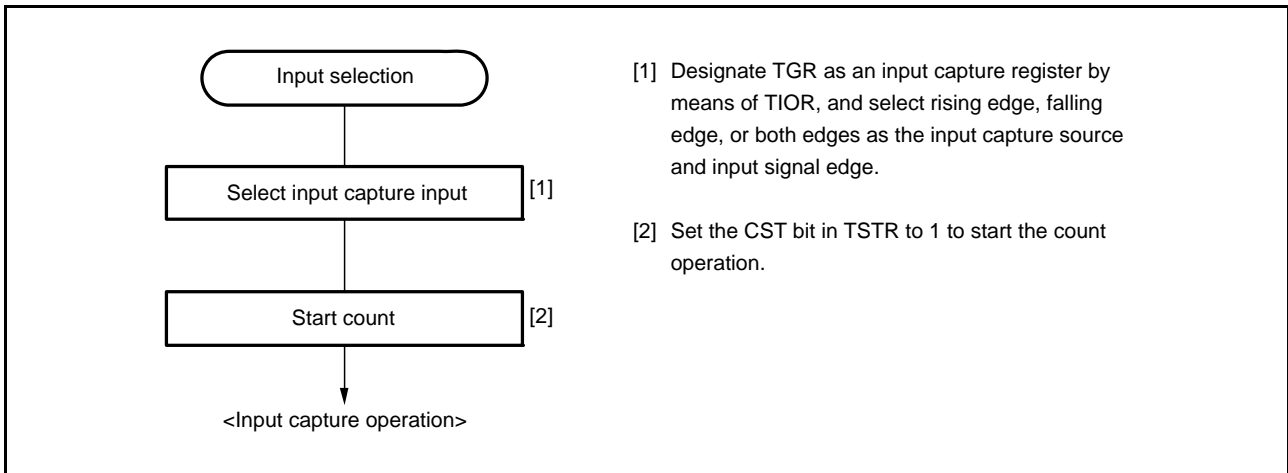


Figure 22.10 Example of Input Capture Operation Setting Procedure

(b) Example of Input Capture Operation

Figure 22.11 shows an example of input capture operation.

In this example, both rising and falling edges have been selected as the MTIOCA pin input capture input edge, the falling edge has been selected as the MTIOCB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

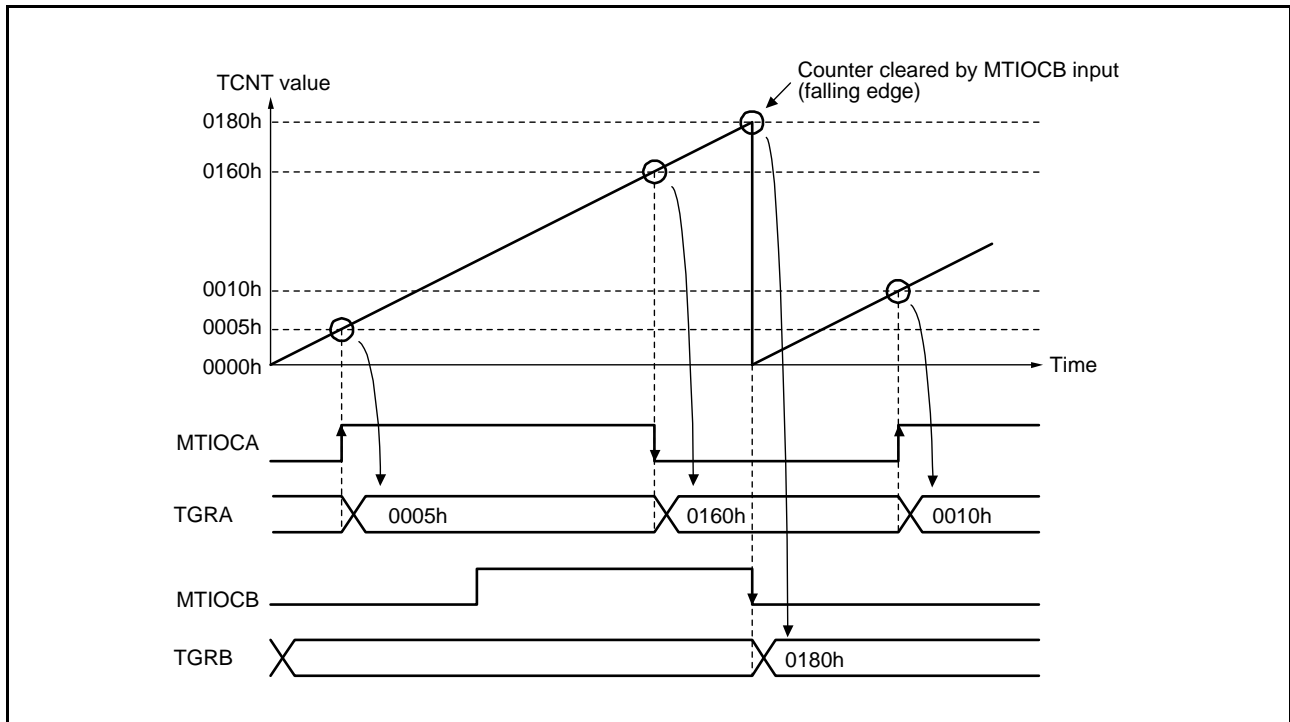


Figure 22.11 Example of Input Capture Operation

### 22.3.2 Synchronous Operation

In synchronous operation, the values in multiple TCNT counters can be modified simultaneously (synchronous presetting). In addition, multiple TCNT counters can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TCR.

Synchronous operation increases the number of TGR registers assigned to a single time base.

MTU0 to MTU4 can all be designated for synchronous operation.

MTU5 cannot be used for synchronous operation.

#### (1) Example of Synchronous Operation Setting Procedure

Figure 22.12 shows an example of the synchronous operation setting procedure.

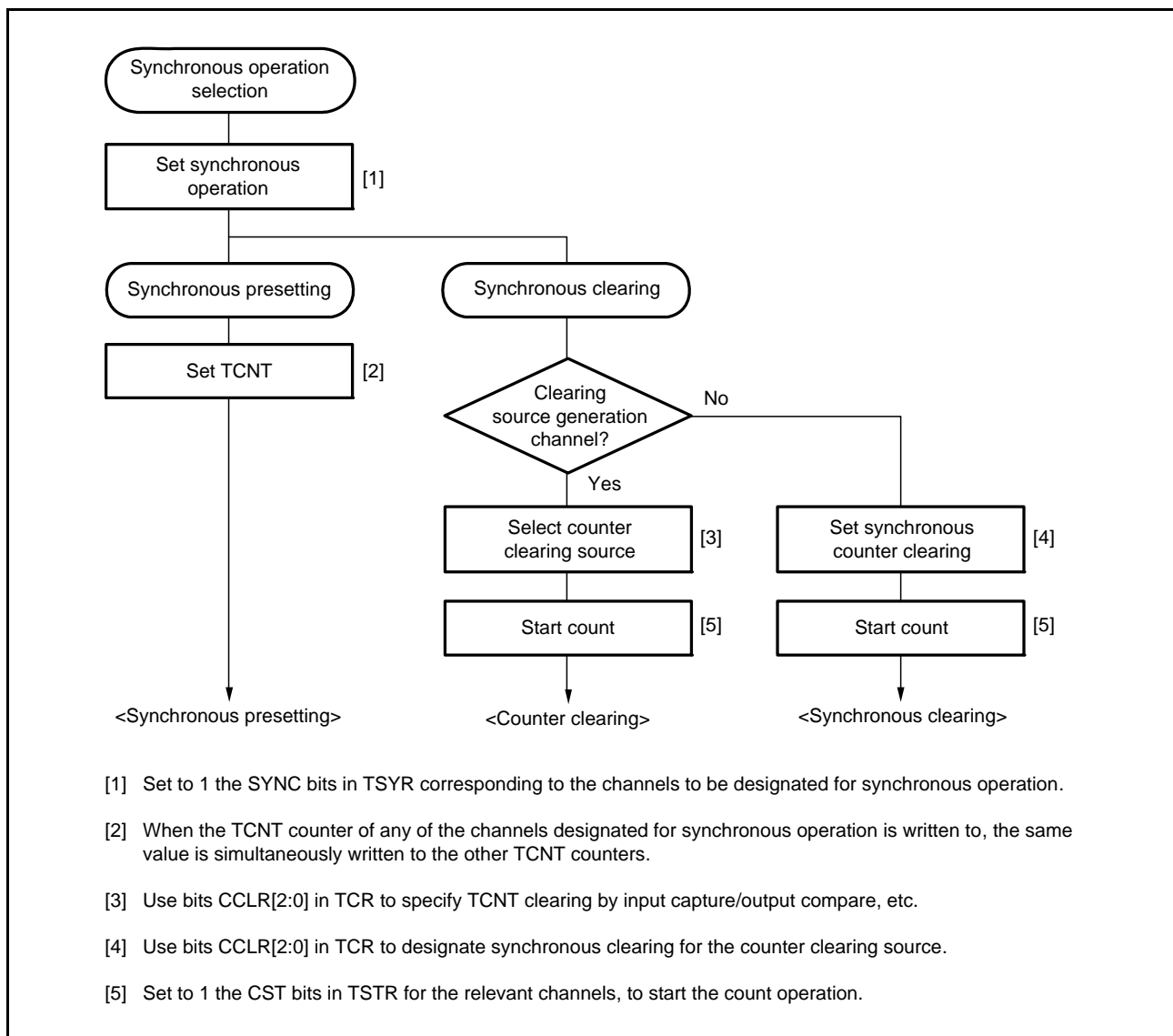


Figure 22.12 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 22.13 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for MTU0 to MTU2, MTU0.TGRB compare match has been set as the counter clearing source in MTU0, and synchronous clearing has been set for the counter clearing source in MTU1 and MTU2.

Three-phase PWM waveforms are output from pins MTIOC0A, MTIOC1A, and MTIOC2A. At this time, synchronous presetting and synchronous clearing by MTU0.TGRB compare match are performed for the TCNT counters in MTU0 to MTU2, and the data set in MTU0.TGRB is used as the PWM cycle.

For details of PWM modes, see section 22.3.5, PWM Modes.

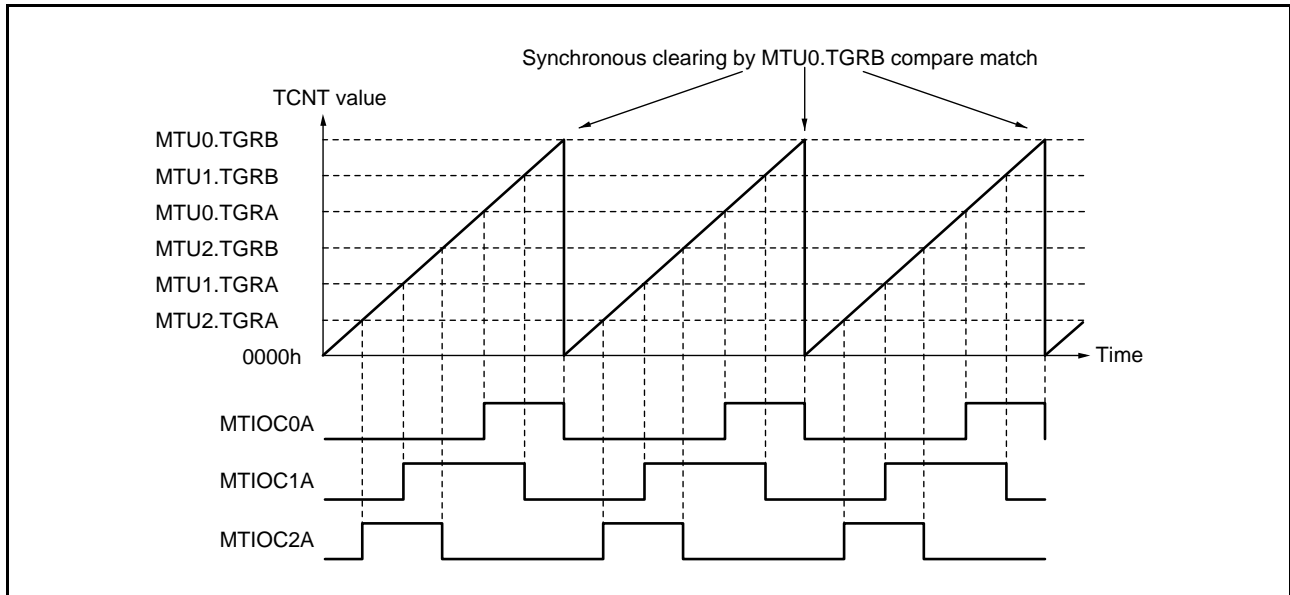


Figure 22.13 Example of Synchronous Operation

### 22.3.3 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 22.43 lists the register combinations used in buffer operation.

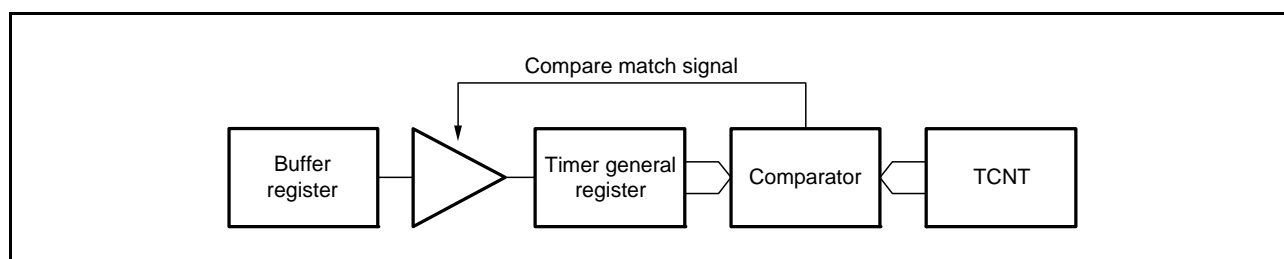
**Table 22.43 Register Combinations in Buffer Operation**

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 22.14.

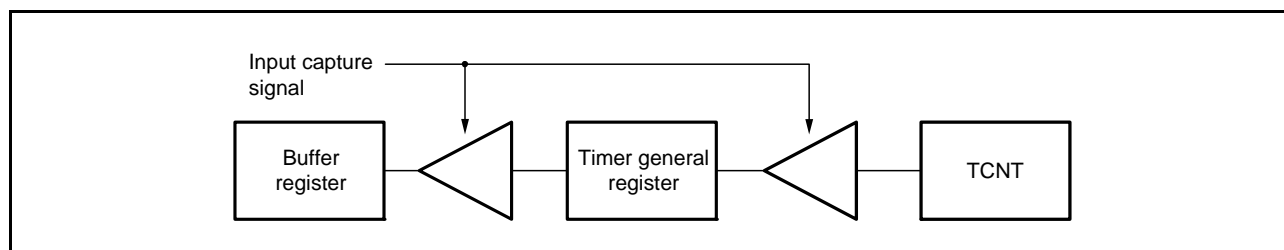


**Figure 22.14 Compare Match Buffer Operation**

- 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 22.15.



**Figure 22.15 Input Capture Buffer Operation**



(1) Example of Buffer Operation Setting Procedure

Figure 22.16 shows an example of the buffer operation setting procedure.

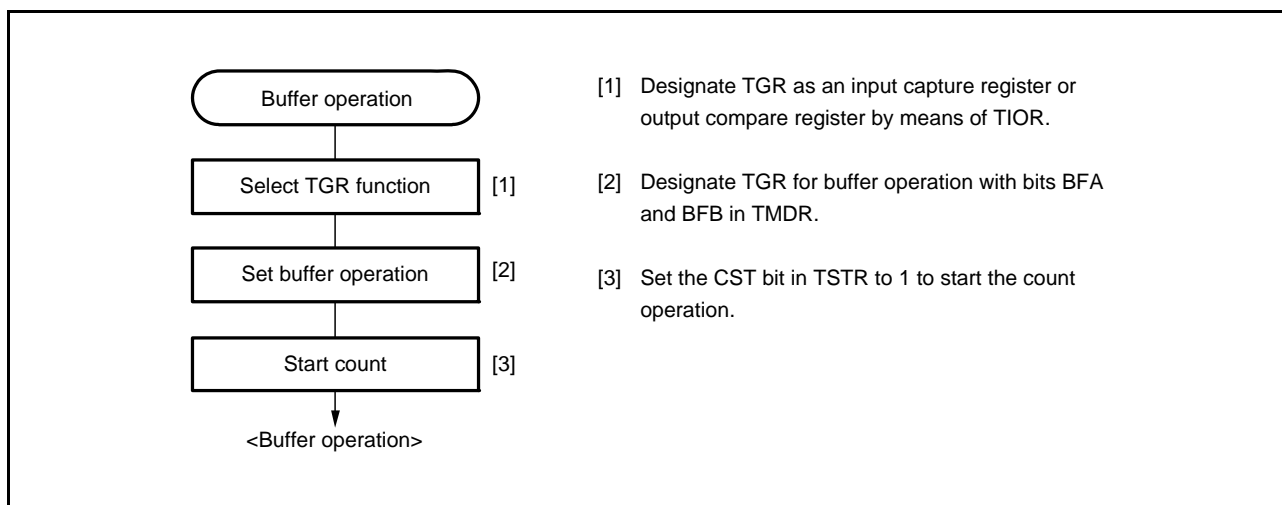


Figure 22.16 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TGR is an Output Compare Register

Figure 22.17 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 for TCNT clearing in response to compare match B, output of 1 in response to compare match A, and then output of 0 in response to compare match B. In this example, the TTSA bit in TBTM is cleared to 0.

As buffer operation has been set, when compare match A occurs, the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs.

For details of PWM modes, see section 22.3.5, PWM Modes.

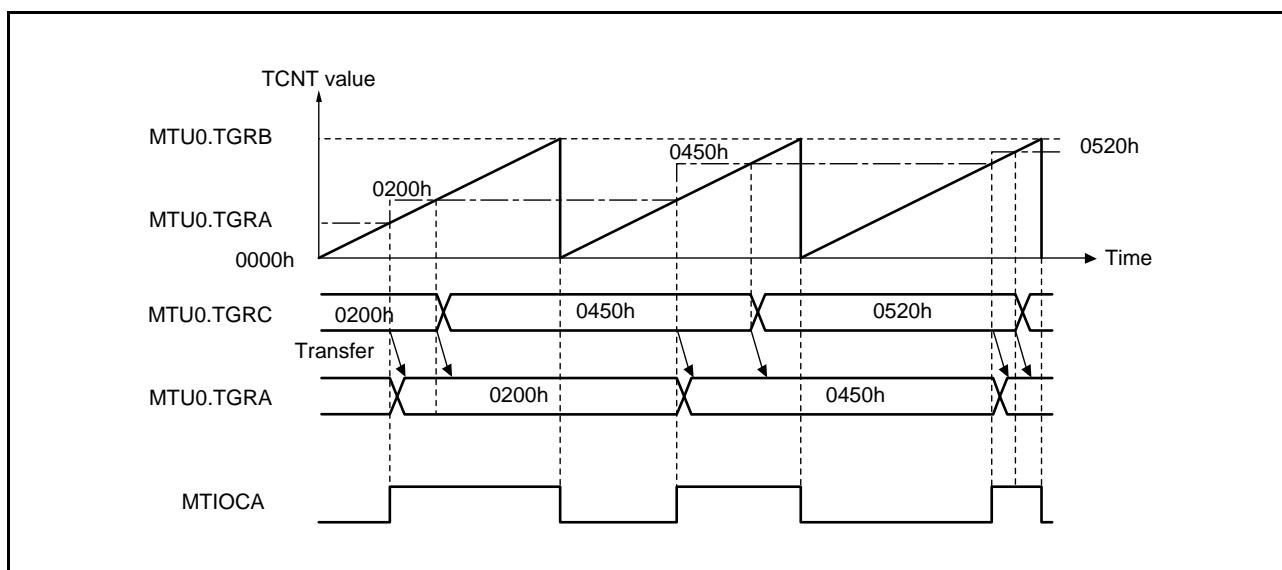


Figure 22.17 Example of Buffer Operation (1)

(b) When TGR is an Input Capture Register

Figure 22.18 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.

As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

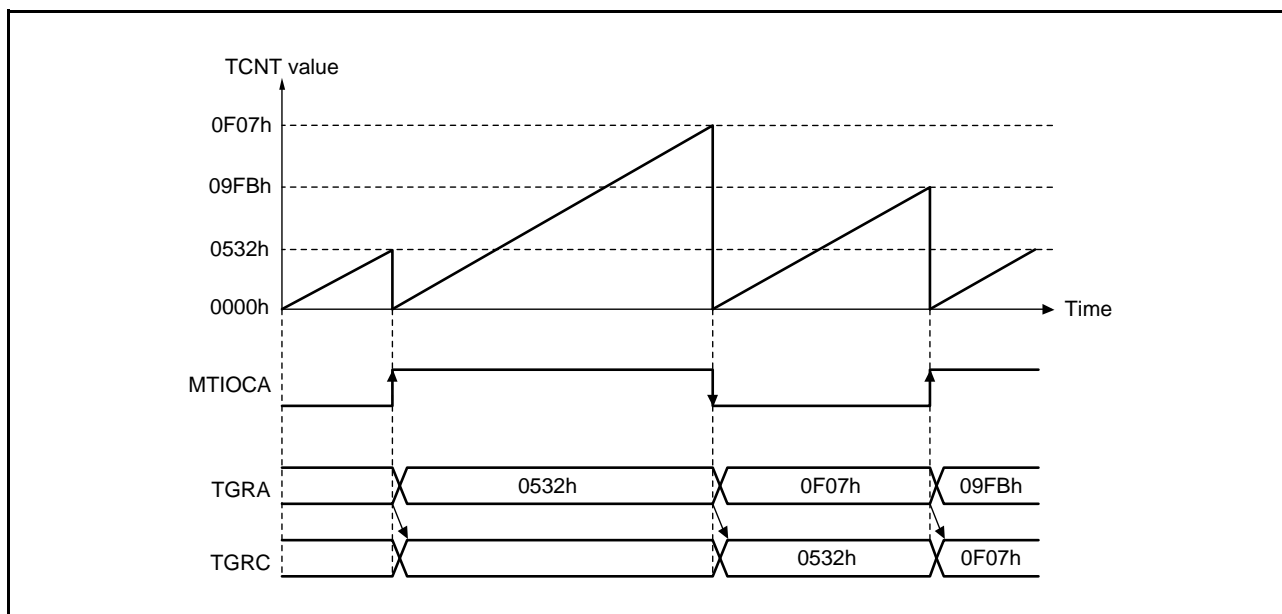


Figure 22.18 Example of Buffer Operation (2)

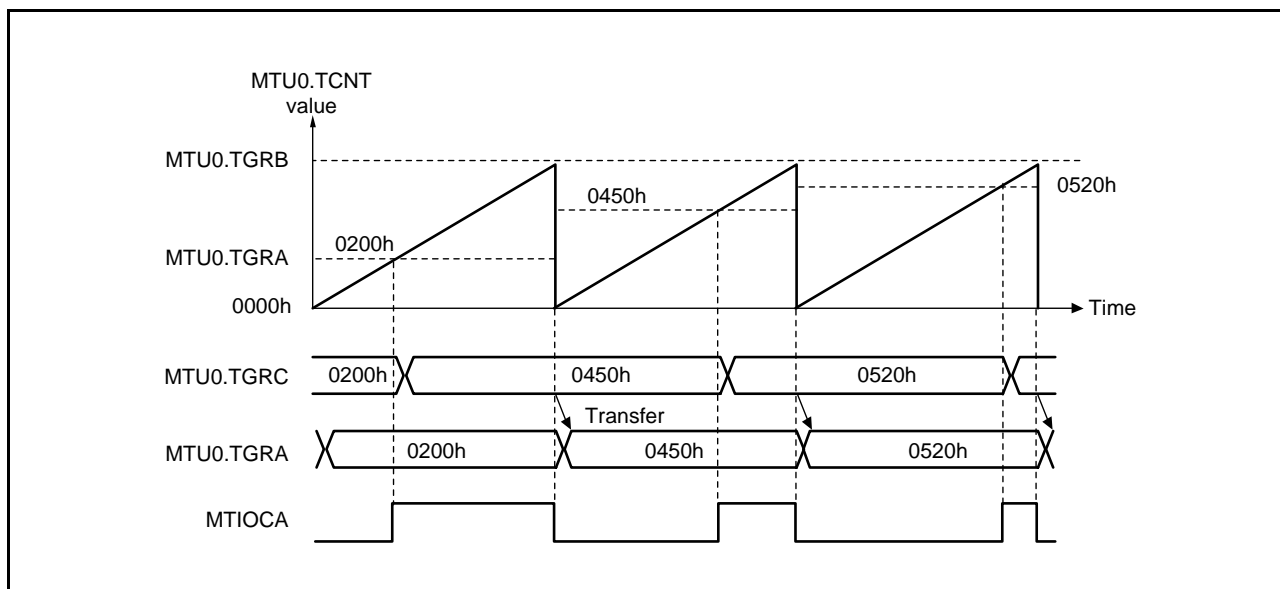
### (3) Selecting Timing for Transfer from Buffer Registers to Timer General Registers in Buffer Operation

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 (MTU0.TBTM, MTU3.TBTM, and MTU4.TBTM). Either compare match (initial setting) 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 → 0000h)
- When 0000h is written to TCNT during counting
- When TCNT is cleared to 0000h under the condition specified in the CCLR[2:0] bits in TCR

Note: • TBTM must be modified only while TCNT stops.

Figure 22.19 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 for MTU0.TCNT clearing in response to compare match B, output of 1 in response to compare match A, and then output of 0 in response to compare match B. The TTSA bit in MTU0.TBTM is set to 1.



**Figure 22.19 Example of Buffer Operation When MTU0.TCNT Clearing is Selected for MTU0.TGRC-to-MTU0.TGRA Transfer Timing**

#### 22.3.4 Cascaded Operation

In cascaded operation, 16-bit counters in different two channels are used together as a 32-bit counter. This function works when overflow/underflow of MTU2.TCNT is selected as the counter clock for MTU1 through the TPSC[2:0] bits in TCR.

Underflow occurs only when the lower 16 bits of TCNT is in phase counting mode.

Table 22.44 lists the register combinations used in cascaded operation.

Note: • When phase counting mode is set for MTU1 or MTU2, the counter clock setting is invalid and the counters operate independently in phase counting mode.

**Table 22.44 Cascaded Combinations**

Combination	Upper 16 Bits	Lower 16 Bits
MTU1 and MTU2	MTU1.TCNT	MTU2.TCNT

For simultaneous input capture of MTU1.TCNT and MTU2.TCNT during cascaded operation, additional input capture input pins can be specified by the input capture control register (TICCR). The input-capture condition is of edges in the signal produced by taking the logical OR of the input level on the main input pin and the input level on the added input pin. Accordingly, if either is at the high level, a change in the level of the other will not produce an edge for detection. For details, see (4) Cascaded Operation Example (c). For input capture in cascade connection, refer to section 22.6.22, Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection.

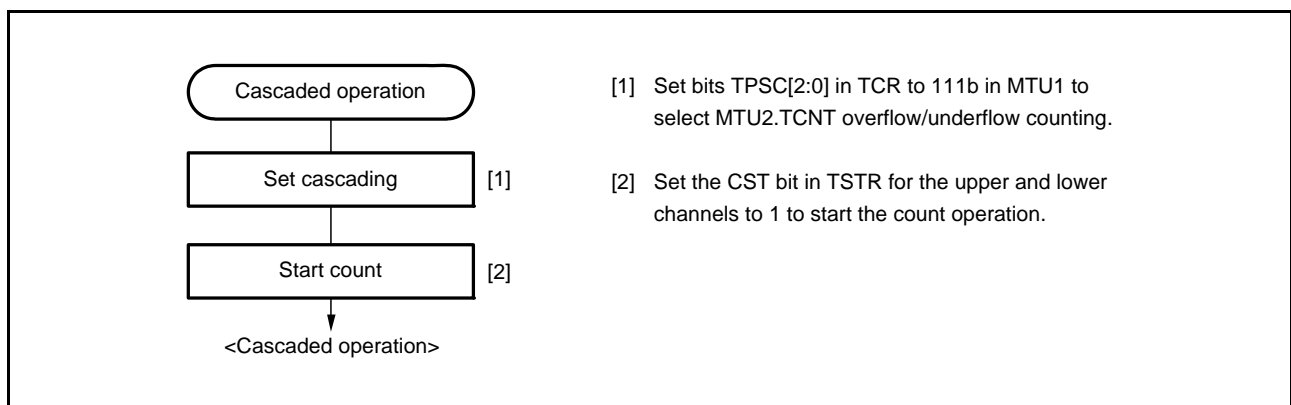
Table 22.45 lists the TICCR setting and input capture input pins.

**Table 22.45 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 22.20 shows an example of the cascaded operation setting procedure.



**Figure 22.20 Cascaded Operation Setting Procedure**

(2) Cascaded Operation Example (a)

Figure 22.21 illustrates the operation when MTU2.TCNT overflow/underflow counting has been set for MTU1.TCNT and phase counting mode has been designated for MTU2.

MTU1.TCNT is incremented by MTU2.TCNT overflow and decremented by MTU2.TCNT underflow.

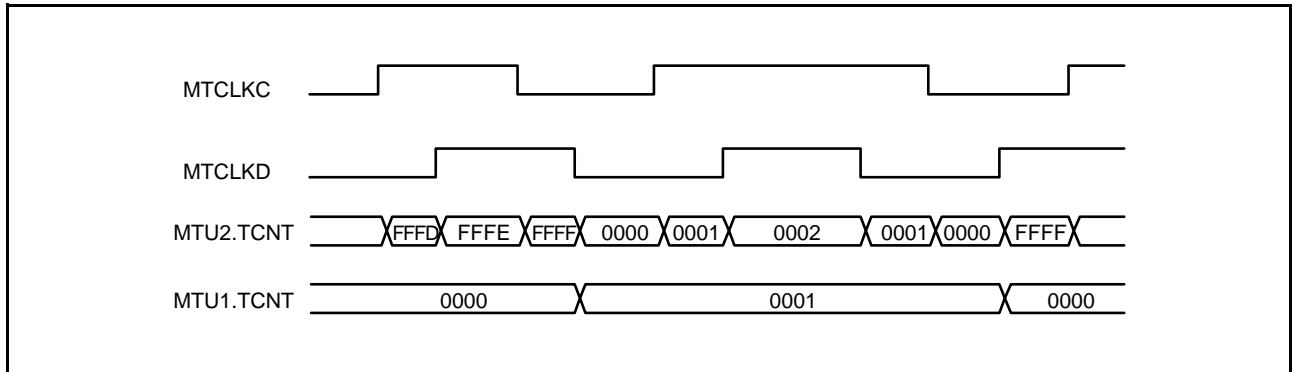


Figure 22.21 Cascaded Operation Example (a)

(3) Cascaded Operation Example (b)

Figure 22.22 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the I2AE bit in TICCRR has been set to 1 to include the MTIOC2A pin in the MTU1.TGRA input capture conditions. In this example, the IOA3 to IOA0 bits in MTU1.TIOR have selected the MTIOC1A rising edge for the input capture timing while the IOA3 to IOA0 bits in MTU2.TIOR 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.

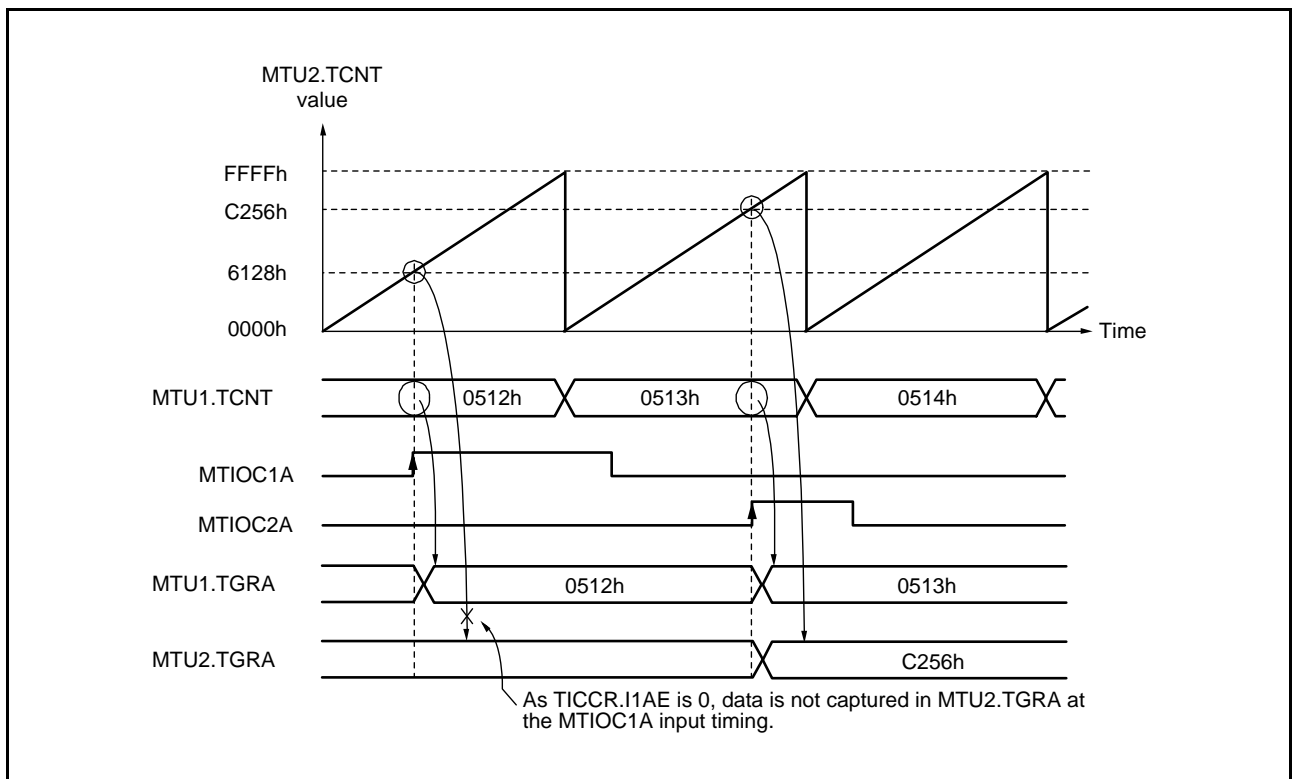


Figure 22.22 Cascaded Operation Example (b)

(4) Cascaded Operation Example (c)

Figure 22.23 illustrates the operation when MTU1.TCNT and MTU2.TCNT have been cascaded and the I2AE and I1AE bits in TICCRA 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 IOA3 to IOA0 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 OR result of MTIOC1A and MTIOC2A input is used for the MTU1.TGRA and MTU2.TGRA input capture conditions.

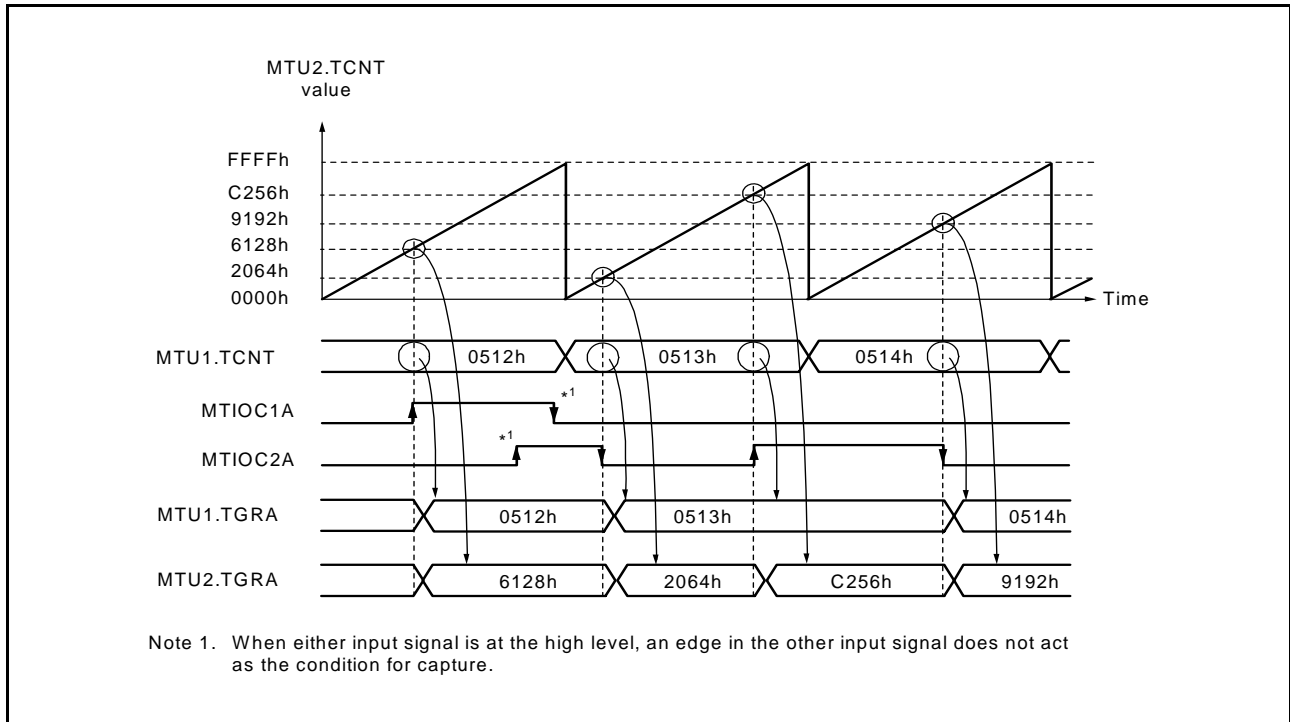


Figure 22.23 Cascaded Operation Example (c)

(5) Cascaded Operation Example (d)

Figure 22.24 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 IOA3 to IOA0 bits in MTU1.TIOR have selected occurrence of MTU0.TGRA compare match or input capture for the input capture timing while the IOA3 to IOA0 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.

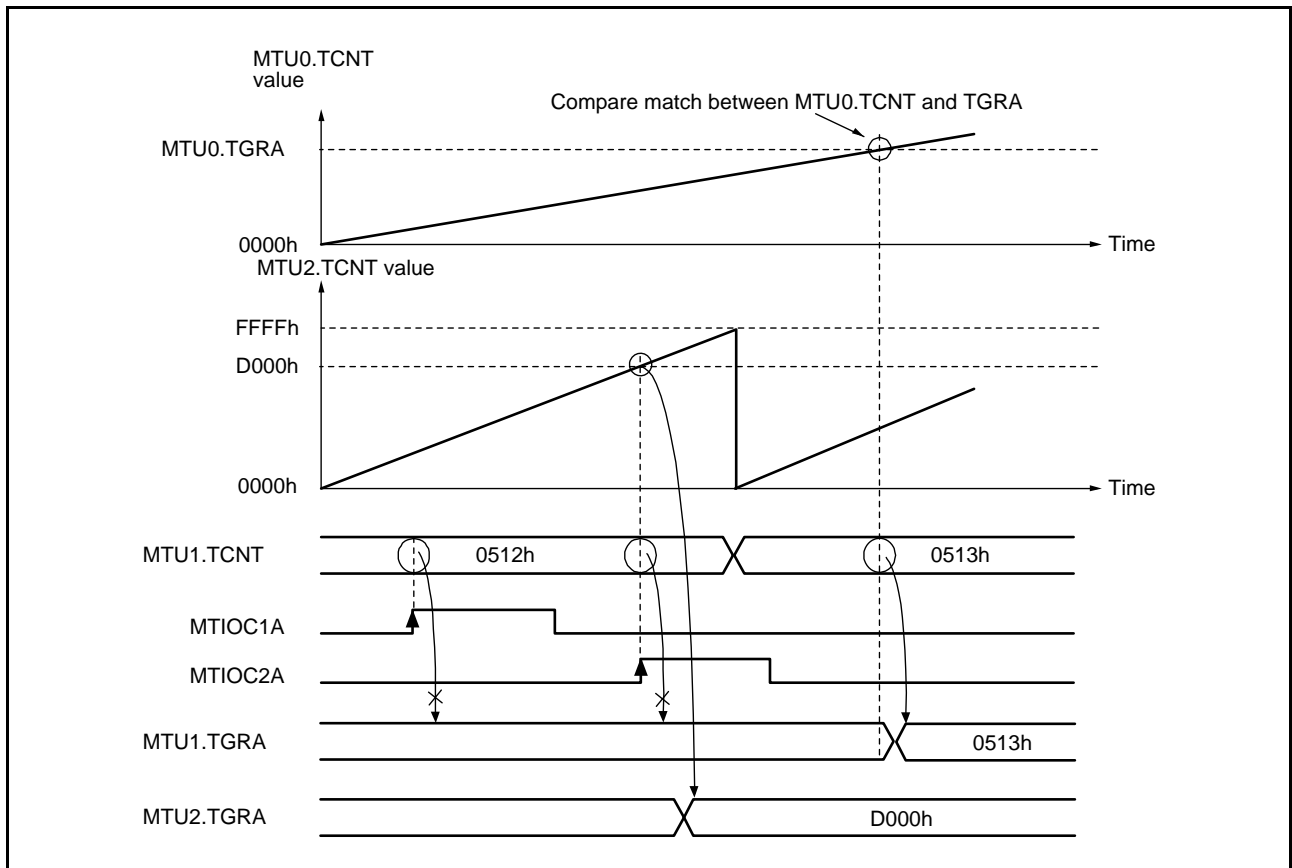


Figure 22.24 Cascaded Operation Example (d)

### 22.3.5 PWM Modes

PWM modes are provided to output PWM waveforms from the external pins. The output of a 0, the output of a 1, or toggling of the output in response to a compare match with each TGR is selectable.

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 cycle can be specified in that register.

Every channel can be set to PWM mode independently. Synchronous operation is also possible.

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 bits IOA3 to IOA0 and IOC3 to IOC0 in TIOR are output from the MTIOCnA and MTIOCnC pins at compare matches A and C, and the level specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR are output at compare matches B and D. 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, up to eight phases of PWM waveforms can be output.

#### (b) PWM Mode 2

PWM output is generated using one TGR as the cycle register and the others as duty registers. The level specified in TIOR is output at compare matches. Upon counter clearing by a synchronized register compare match, the initial value set in TIOR is output from each pin. If the values set in the cycle and duty registers are identical, the output value does not change even when a compare match occurs.

In PWM mode 2, up to eight phases of PWM waveforms can be output when using synchronous operation in combination.

The correspondence between PWM output pins and registers is listed in Table 22.46.

**Table 22.46 PWM Output Registers and Output Pins**

Channel	Register	Output Pins	
		PWM Mode 1	PWM Mode 2
MTU0	MTU0.TGRA	MTIOC0A	MTIOC0A
	MTU0.TGRB		MTIOC0B
	MTU0.TGRC	MTIOC0C	MTIOC0C
	MTU0.TGRD		MTIOC0D
MTU1	MTU1.TGRA	MTIOC1A	MTIOC1A
	MTU1.TGRB		MTIOC1B
MTU2	MTU2.TGRA	MTIOC2A	MTIOC2A
	MTU2.TGRB		MTIOC2B
MTU3	MTU3.TGRA	MTIOC3A	Setting prohibited
	MTU3.TGRB		
	MTU3.TGRC	MTIOC3C	
	MTU3.TGRD		
MTU4	MTU4.TGRA	MTIOC4A	
	MTU4.TGRB		
	MTU4.TGRC	MTIOC4C	
	MTU4.TGRD		

Note: • In PWM mode 2, PWM output is not possible for the TGR register in which the PWM cycle is set.



(1) Example of PWM Mode Setting Procedure

Figure 22.25 shows an example of the PWM mode setting procedure.

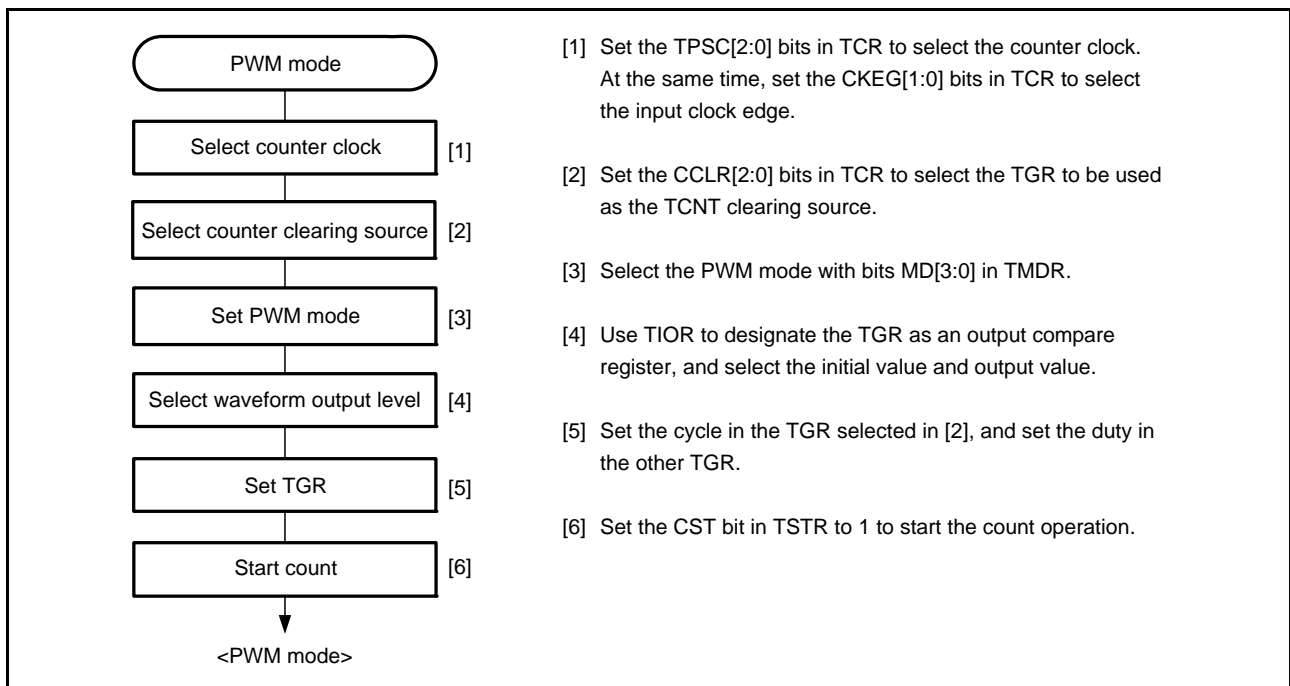


Figure 22.25 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 22.26 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 cycle, and the value set in TGRB is used as the duty.

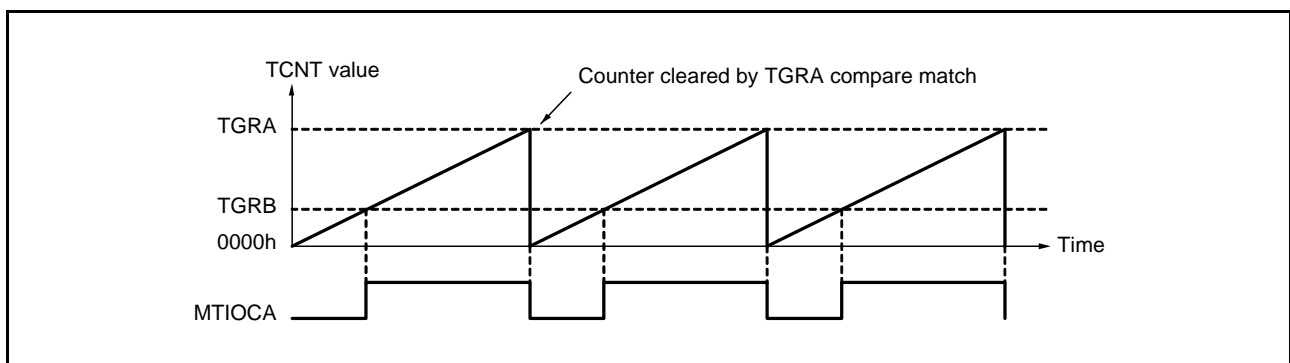


Figure 22.26 Example of PWM Mode Operation

Figure 22.27 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 0 is set as the initial output value and 1 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 cycle, and the values set in the other TGRs are used as the duty.

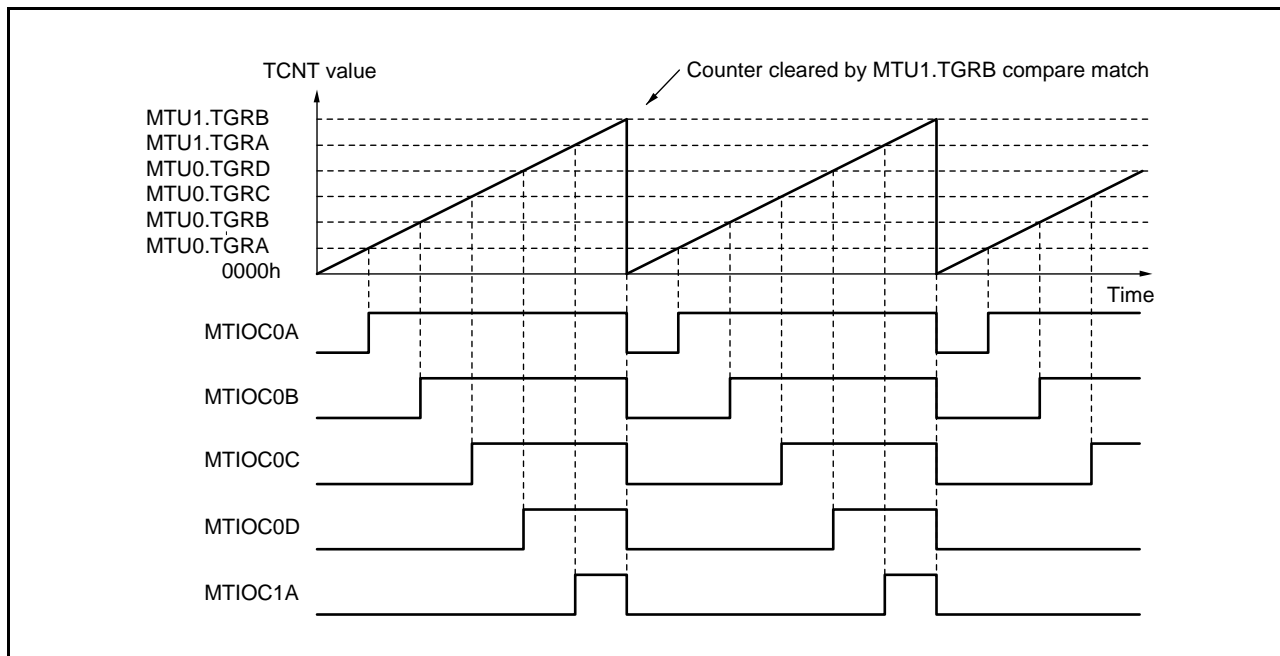


Figure 22.27 Example of PWM Mode Operation

Figure 22.28 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

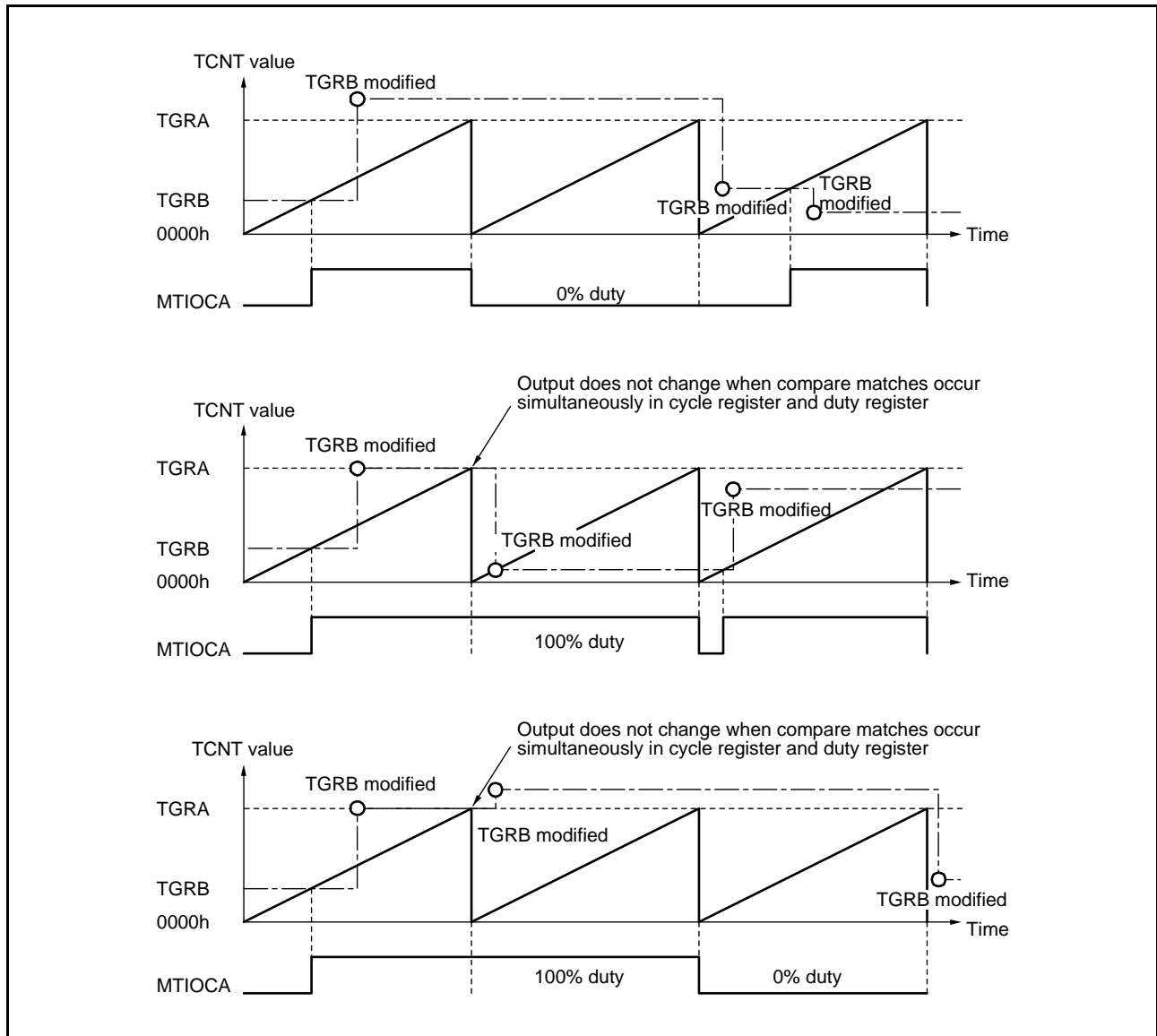


Figure 22.28 Examples of PWM Mode Operation

### 22.3.6 Phase Counting Mode

In phase counting mode, the phase difference between two external input clocks is detected and TCNT is incremented or decremented accordingly. This mode can be set for MTU1 and MTU2.

When phase counting mode is specified, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC[2:0] and bits CKEG[1:0] in TCR. However, the functions of bits CCLR[1:0] in TCR and of TIOR, TIER, and TGR are valid, and input capture/compare match and interrupt functions can be used.

This can be used for two-phase encoder pulse input.

If an overflow occurs while TCNT is counting up, a TCIV interrupt is generated while the TCIEV bit in the corresponding TIER is 1. If an underflow occurs while TCNT is counting down, a TCIU interrupt is generated while the TCIEU bit in the corresponding TIER is 1.

The TCFD bit in TSR is the count direction flag. Read the TCFD flag to check whether TCNT is counting up or down.

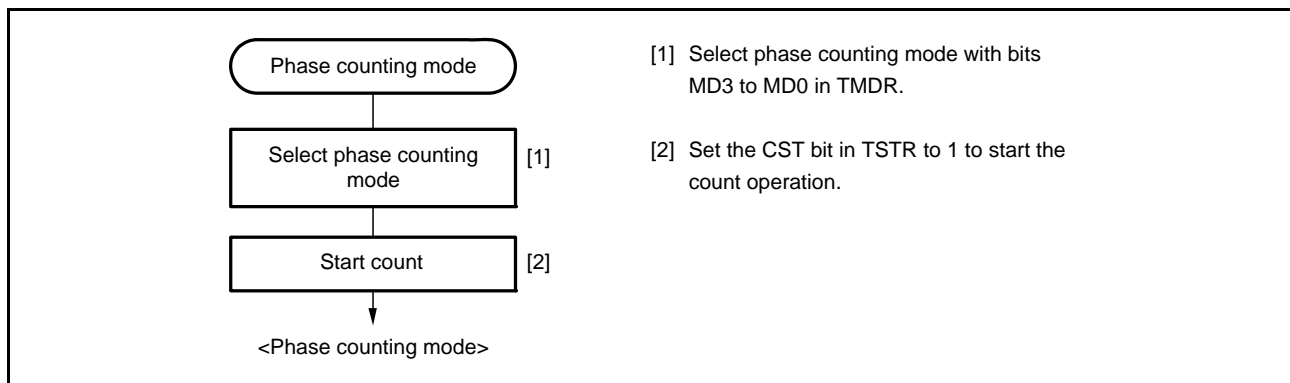
Table 22.47 lists the correspondence between external clock pins and channels.

**Table 22.47 Clock Input Pins in Phase Counting Mode**

Channel	External Clock Input Pins	
	A-Phase	B-Phase
MTU1	MTCLKA	MTCLKB
MTU2	MTCLKC	MTCLKD

#### (1) Example of Phase Counting Mode Setting Procedure

Figure 22.29 shows an example of the phase counting mode setting procedure.



**Figure 22.29 Example of Phase Counting Mode Setting Procedure**

(2) Examples of Phase Counting Mode Operation

In phase counting mode, TCNT is incremented or decremented according to the phase difference between two external clocks. There are four modes according to the count conditions.

(a) Phase Counting Mode 1

Figure 22.30 shows an example of operation in phase counting mode 1, and Table 22.48 summarizes the TCNT up/down-count conditions.

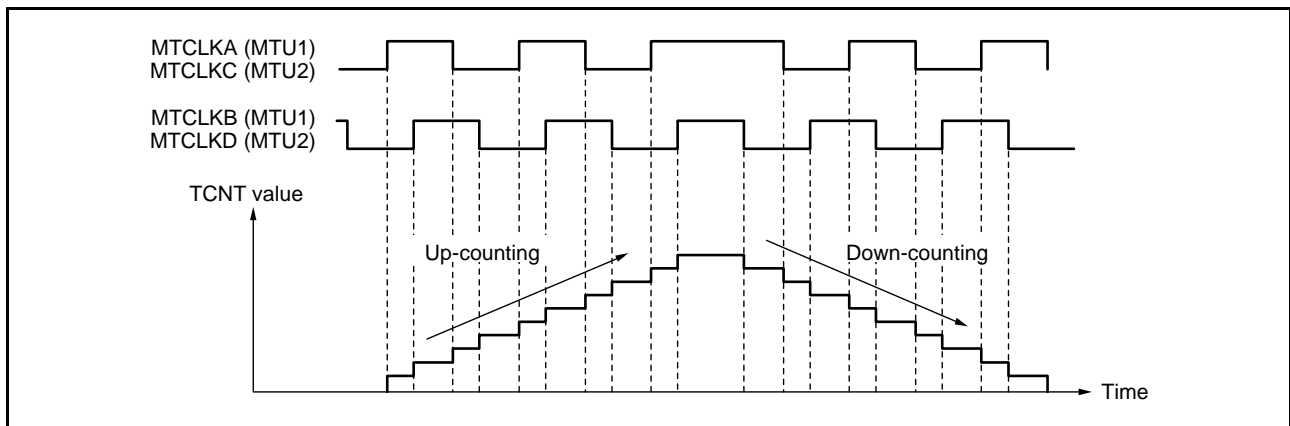


Figure 22.30 Example of Operation in Phase Counting Mode

Table 22.48 Up/Down-Count 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 22.31 shows an example of operation in phase counting mode 2, and Table 22.49 summarizes the TCNT up/down-count conditions.

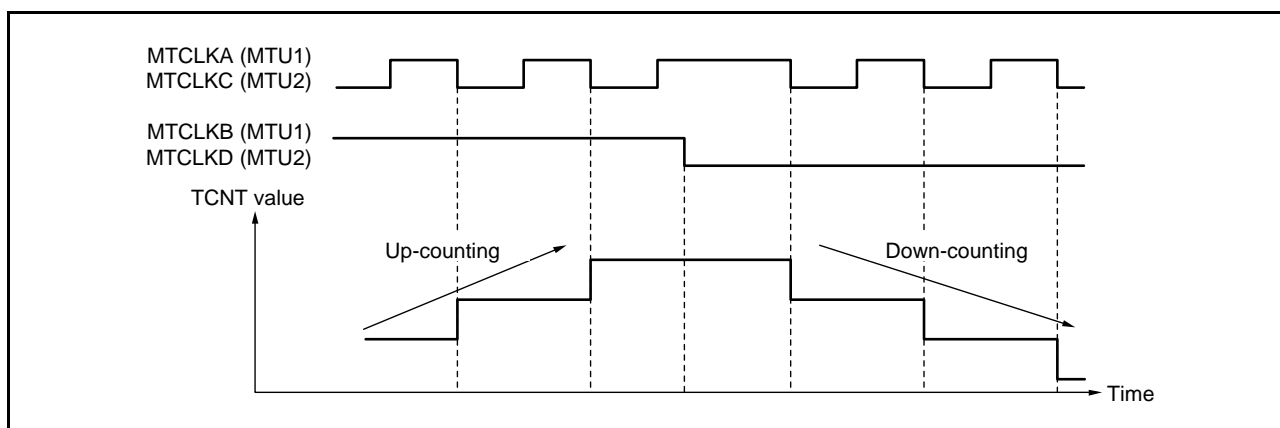


Figure 22.31 Example of Operation in Phase Counting Mode 2

Table 22.49 Up/Down-Count Conditions in Phase Counting Mode 2

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High		None (Don't care)
Low		None (Don't care)
	Low	None (Don't care)
	High	Up-counting
High		None (Don't care)
Low		None (Don't care)
	High	None (Don't care)
	Low	Down-counting

: Rising edge  
 : Falling edge

(c) Phase Counting Mode 3

Figure 22.32 shows an example of operation in phase counting mode 3, and Table 22.50 summarizes the TCNT up/down-count conditions.

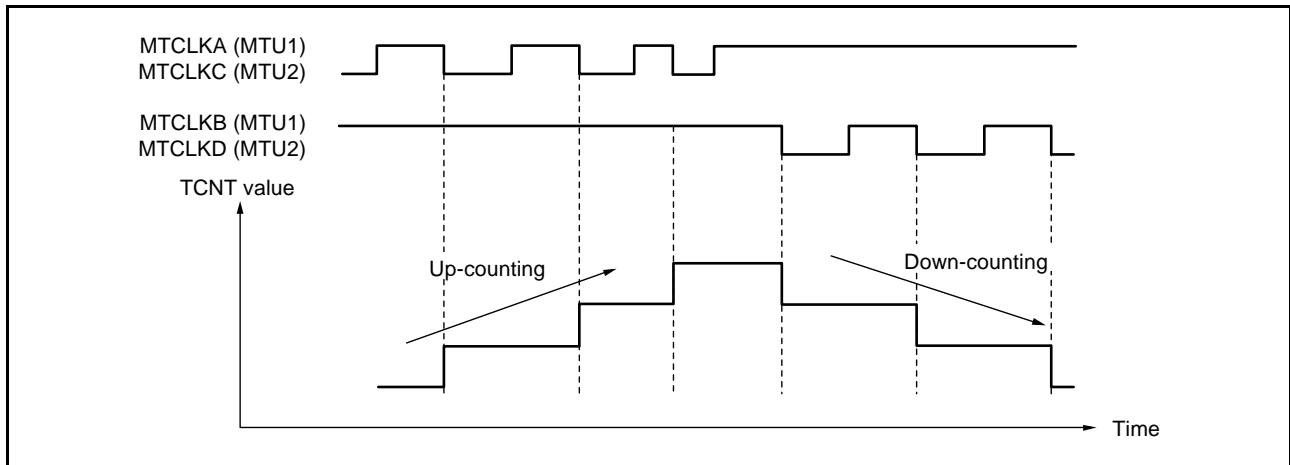


Figure 22.32 Example of Operation in Phase Counting Mode 3

Table 22.50 Up/Down-Count Conditions in Phase Counting Mode 3

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High		None (Don't care)
Low		None (Don't care)
	Low	None (Don't care)
	High	Up-counting
High		Down-counting
Low		None (Don't care)
	High	None (Don't care)
	Low	None (Don't care)

: Rising edge  
 : Falling edge

(d) Phase Counting Mode 4

Figure 22.33 shows an example of operation in phase counting mode 4, and Table 22.51 summarizes the TCNT up/down-count conditions.

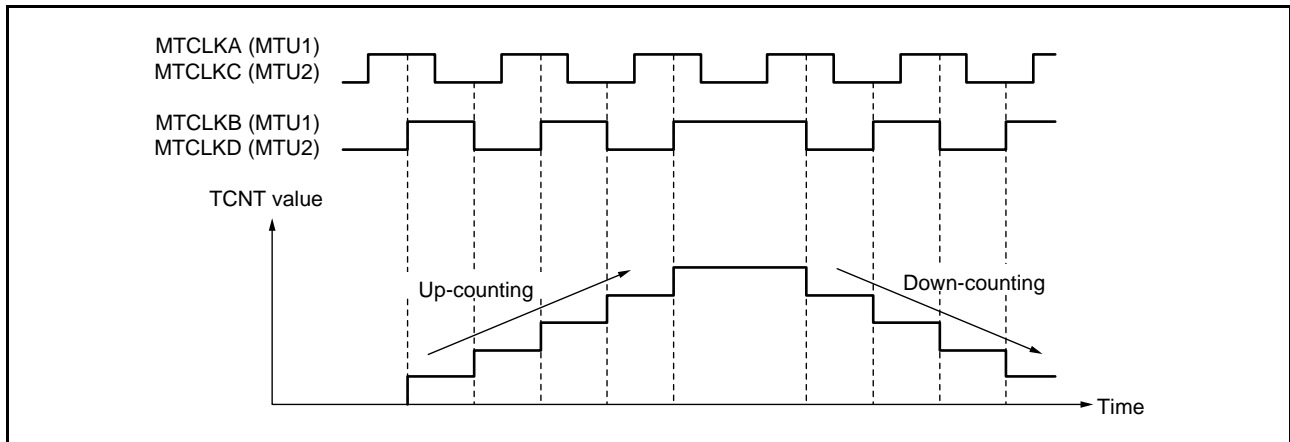


Figure 22.33 Example of Operation in Phase Counting Mode

Table 22.51 Up/Down-Count Conditions in Phase Counting Mode 4

MTCLKA (MTU1) MTCLKC (MTU2)	MTCLKB (MTU1) MTCLKD (MTU2)	Operation
High		Up-counting
Low		
	Low	None (Don't care)
	High	
High		Down-counting
Low		
	High	None (Don't care)
	Low level	

: Rising edge

: Falling edge



### (3) Phase Counting Mode Application Example

Figure 22.34 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 TGRC are used for the compare match function and are set with the speed control cycle and position control cycle. MTU0.TGRB is used for input capture, with MTU0.TGRB and TGRD operating in buffer mode. The MTU1 counter input 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 TGRB for MTU1 are designated for the input capture function and MTU0.TGRA and TGRC compare matches in MTU0 are selected as the input capture sources to store the up/down-counter values for the control cycles. This procedure enables the accurate detection of position and speed.

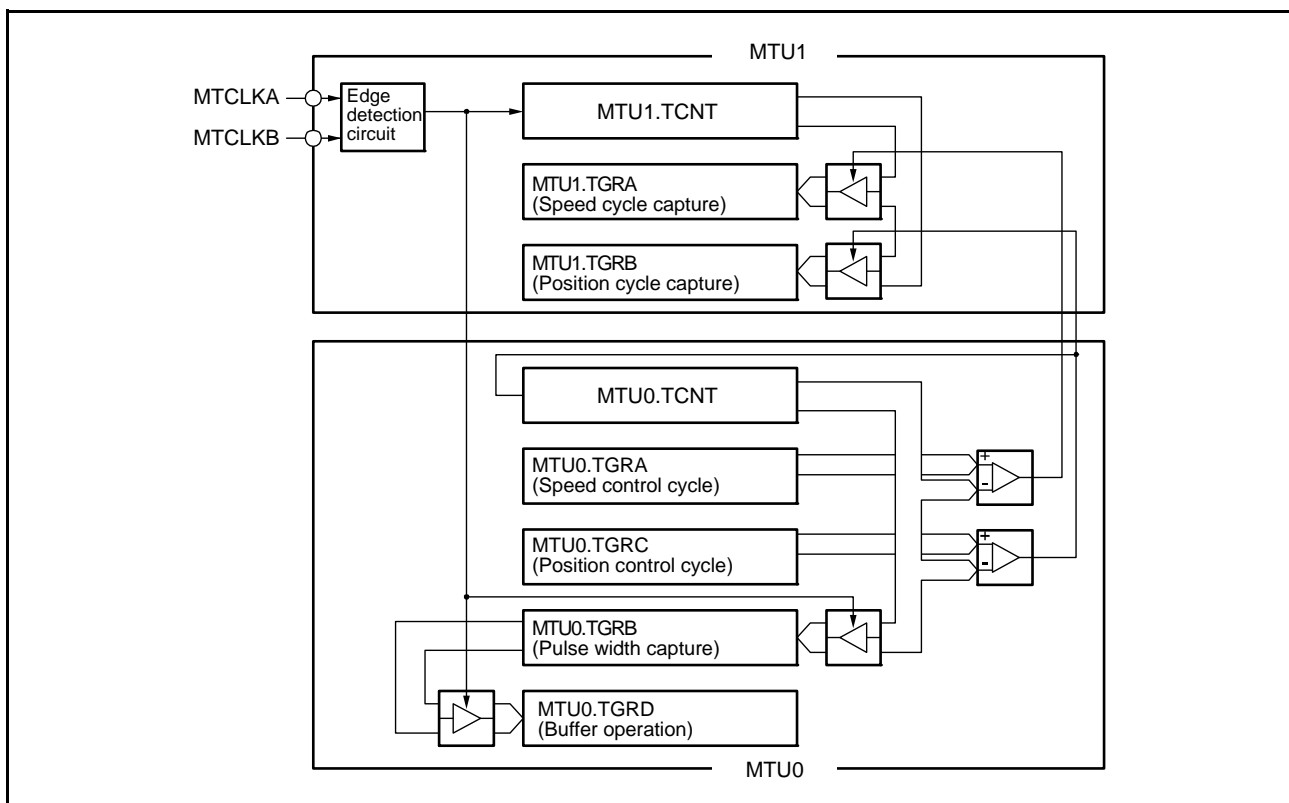


Figure 22.34 Phase Counting Mode Application Example

### 22.3.7 Reset-Synchronized PWM Mode

In the reset-synchronized PWM mode, three phases of positive and negative PWM waveforms 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 22.52 lists the PWM output pins. Table 22.53 lists the settings of the registers.

**Table 22.52 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 22.53 Register Settings for Reset-Synchronized PWM Mode**

Register	Setting
MTU3.TCNT	Initial setting (0000h)
MTU4.TCNT	Initial setting (0000h)
MTU3.TGRA	Set the count cycle 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 22.35 shows an example of procedure for setting the reset-synchronized PWM mode.

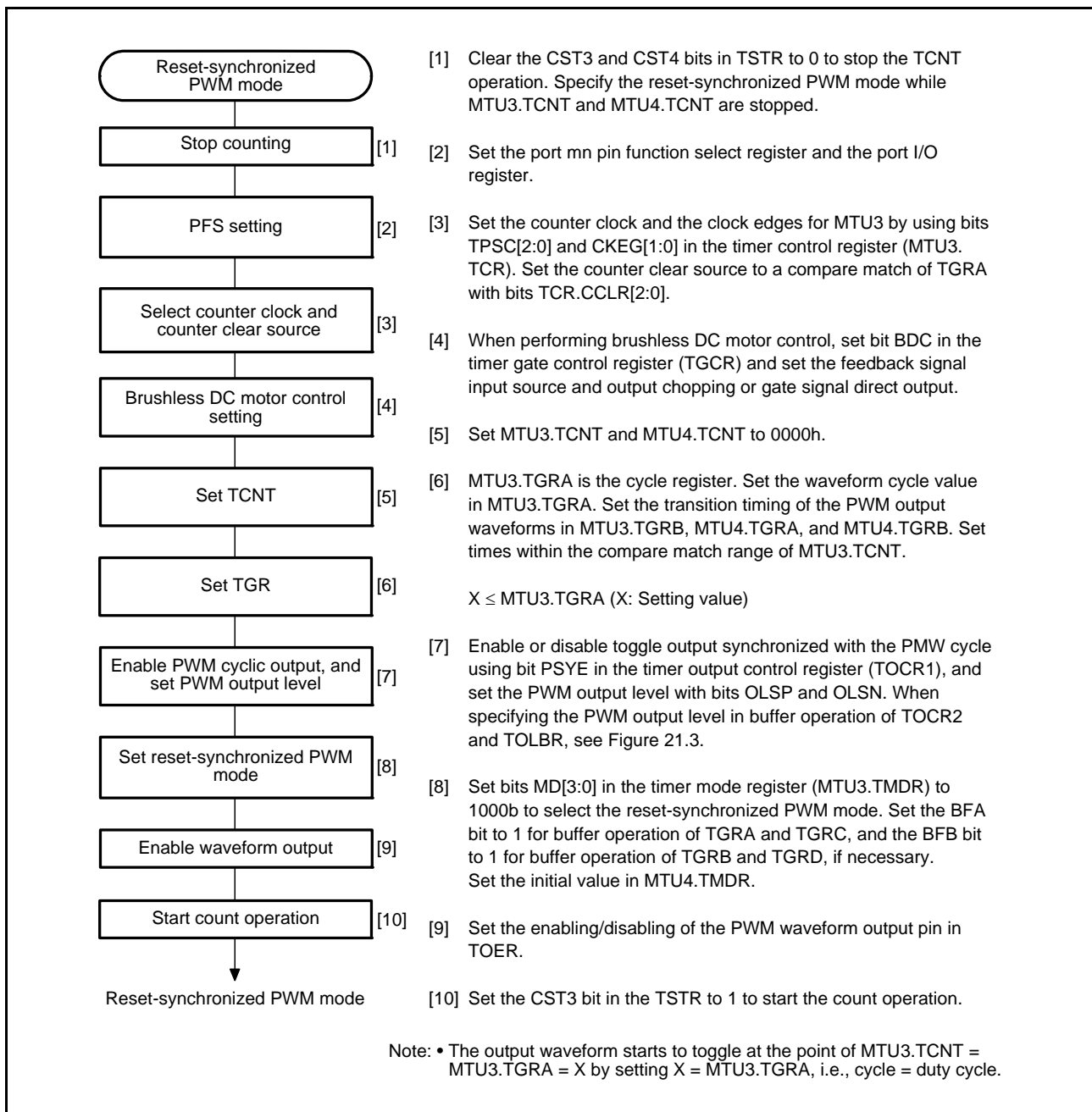


Figure 22.35 Procedure for Selecting Reset-Synchronized PWM Mode

(2) Example of Reset-Synchronized PWM Mode Operation

Figure 22.36 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.

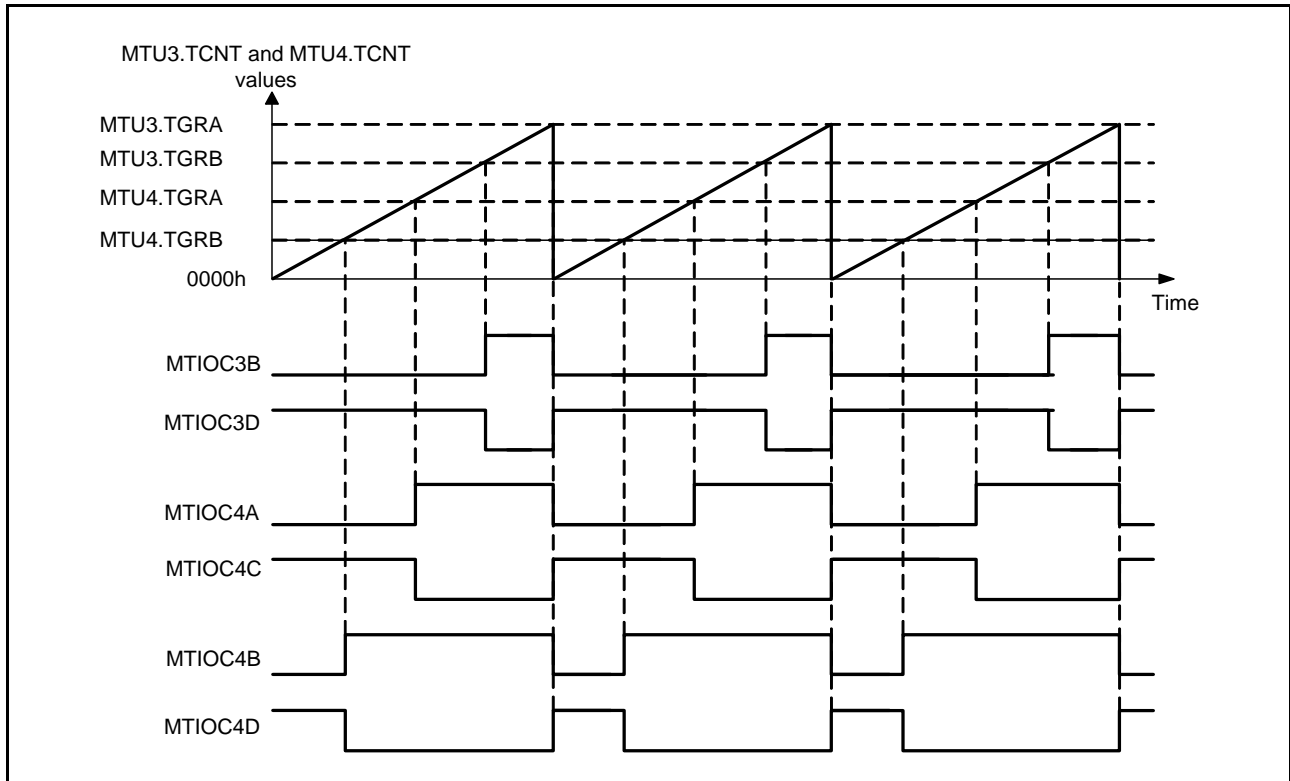


Figure 22.36 Example of Reset-Synchronized PWM Mode Operation (When TOCR's OLSN = 1 and OLSP = 1)

### 22.3.8 Complementary PWM Mode

In complementary PWM mode, three phases of non-overlapping positive and negative PWM waveforms can be output by combining MTU3 and MTU4. PWM waveforms without non-overlapping interval are also available.

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 cycle. MTU3.TCNT and MTU4.TCNT function as up/down-counters.

Table 22.54 lists the PWM output pins used. Table 22.55 lists 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 22.54 Output Pins for Complementary PWM Mode**

Channel	Output Pin	Description
MTU3	MTIOC3A	Toggle output synchronized with PWM cycle (or I/O port)
	MTIOC3B	PWM output pin 1
	MTIOC3C	I/O port*1
	MTIOC3D	PWM output pin 1' (non-overlapping negative-phase waveform of PWM output 1; PWM output without non-overlapping interval is also available)
MTU4	MTIOC4A	PWM output pin 2
	MTIOC4C	PWM output pin 2' (non-overlapping negative-phase waveform of PWM output 2; PWM output without non-overlapping interval is also available)
	MTIOC4B	PWM output pin 3
	MTIOC4D	PWM output pin 3' (non-overlapping negative-phase waveform of PWM output 3; PWM output without non-overlapping interval is also available)

Note 1. Avoid setting the MTIOC3C pin as a timer I/O pin in complementary PWM mode.

**Table 22.55 Register Settings for Complementary PWM Mode**

Channel	Counter/ Register	Description	Read/Write from CPU
MTU3	MTU3.TCNT	Starts up-counting from the value set in the dead time register	Maskable by TRWER setting*1
	MTU3.TGRA	Set MTU3.TCNT upper limit value (1/2 carrier cycle + dead time)	Maskable by TRWER setting*1
	MTU3.TGRB	PWM output 1 compare register	Maskable by TRWER setting*1
	MTU3.TGRC	MTU3.TGRA buffer register	Always readable/writable
	MTU3.TGRD	PWM output 1/MTU3.TGRB buffer register	Always readable/writable
MTU4	MTU4.TCNT	Starts up-counting after being initialized to 0000h	Maskable by TRWER setting*1
	MTU4.TGRA	PWM output 2 compare register	Maskable by TRWER setting*1
	MTU4.TGRB	PWM output 3 compare register	Maskable by TRWER setting*1
	MTU4.TGRC	PWM output 2/MTU4.TGRA buffer register	Always readable/writable
	MTU4.TGRD	PWM output 3/MTU4.TGRB buffer register	Always readable/writable
Timer dead time data register (TDDR)		Set MTU4.TCNT and MTU3.TCNT offset value (dead time value)	Maskable by TRWER setting*1
Timer cycle data register (TCDR)		Set MTU4.TCNT upper limit value (1/2 carrier cycle)	Maskable by TRWER setting*1
Timer cycle buffer register (TCBR)		TCDR buffer register	Always readable/writable
Subcounter (TCNTS)		Subcounter for dead time generation	Read-only
Temporary register 1 (TEMP1)		PWM output 1/MTU3.TGRB temporary register	Not readable/writable
Temporary register 2 (TEMP2)		PWM output 2/MTU4.TGRA temporary register	Not readable/writable
Temporary register 3 (TEMP3)		PWM output 3/MTU4.TGRB temporary register	Not readable/writable

Note 1. Access can be enabled or disabled according to the setting in TRWER (timer read/write enable register).

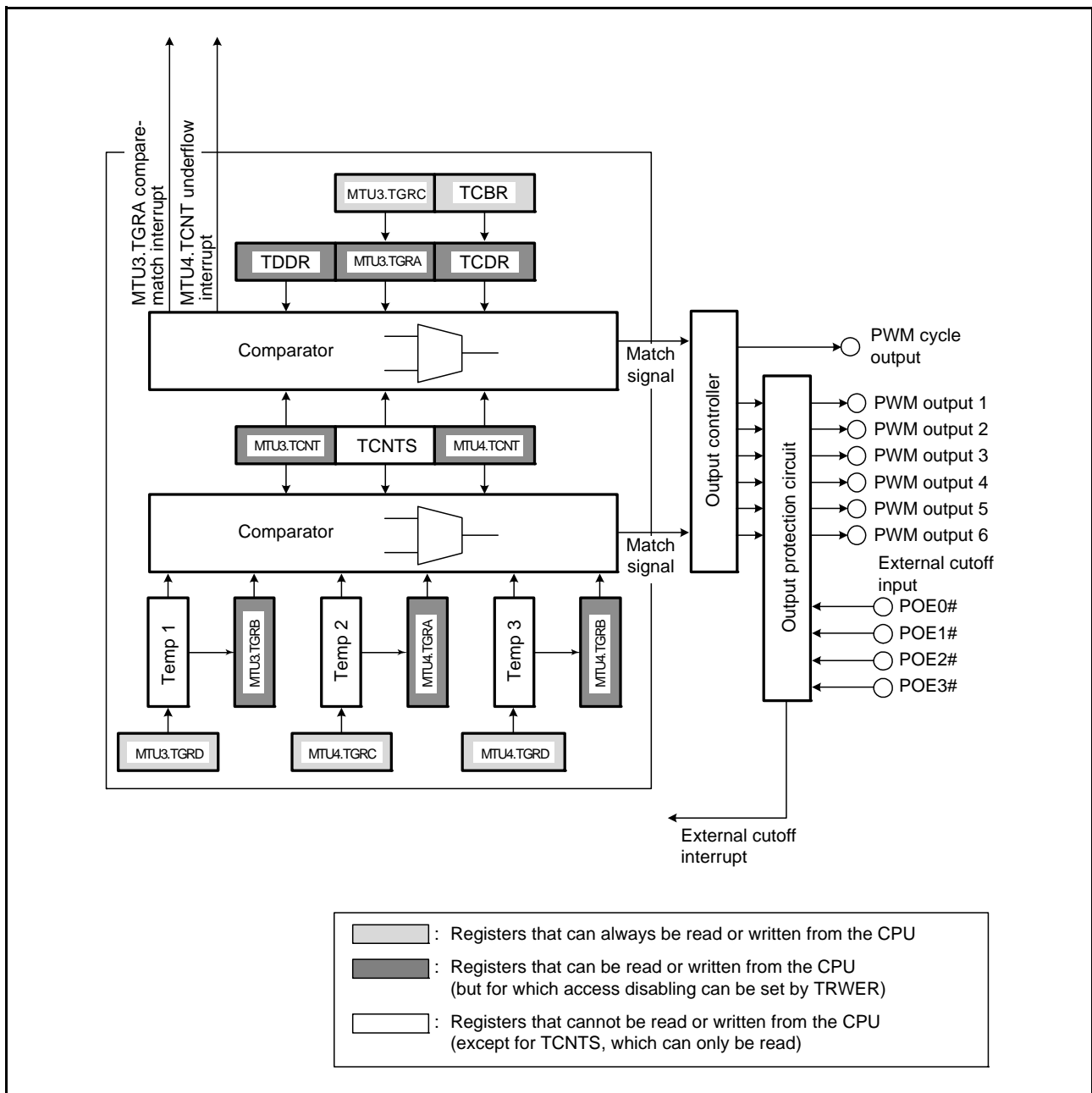


Figure 22.37 Block Diagram of MTU3 and MTU4 in Complementary PWM Mode

(1) Example of Complementary PWM Mode Setting Procedure

Figure 22.38 shows an example of the complementary PWM mode setting procedure.

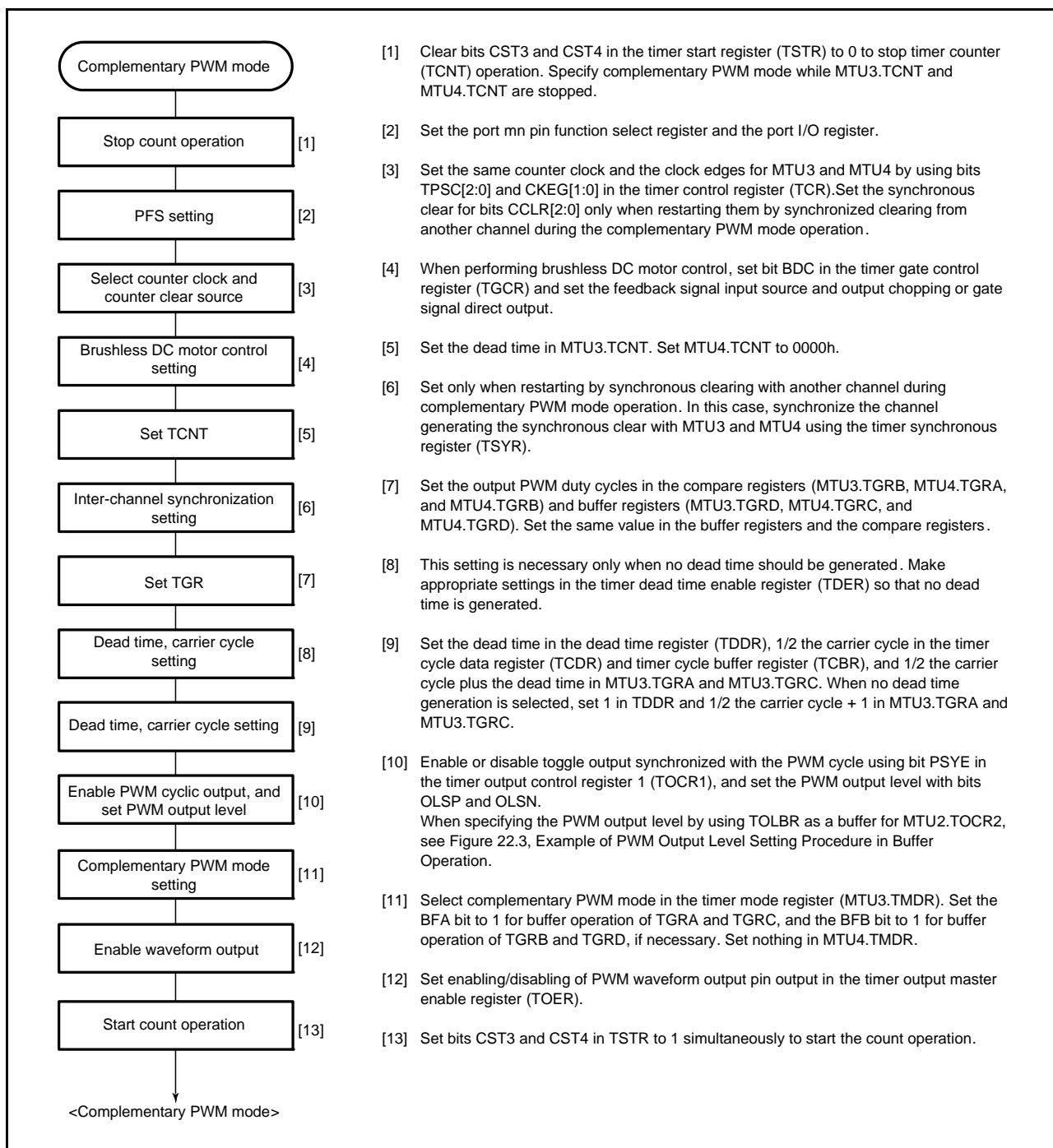


Figure 22.38 Example of Complementary PWM Mode Setting Procedure

## (2) Outline of Complementary PWM Mode Operation

In complementary PWM mode, six phases of PWM waveforms can be output. Figure 22.39 illustrates counter operation in complementary PWM mode, and Figure 22.40 shows an example of operation in complementary PWM mode.

### (a) Counter Operation

In complementary PWM mode, three counters—MTU3.TCNT, MTU4.TCNT, and TCNTS—perform up-/down-count operations.

MTU3.TCNT is automatically initialized to the value set in TDDR when complementary PWM mode is selected and the CST bit in TSTR is 0.

When the CST 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 MTU3.TCNT value matches TDDR, the counter switches to up-counting, and the operation is repeated in this way.

MTU4.TCNT should be initialized to 0000h.

When the CST bit is set to 1, MTU4.TCNT counts up in synchronization with MTU3.TCNT, and switches to down-counting when it matches TCDR. On reaching 0000h, MTU4.TCNT switches to up-counting, and the operation is repeated in this way.

TCNTS is a read-only counter. It does not need to be initialized.

When MTU3.TCNT matches TCDR during up-/down-counting of TCNT in MTU3 and MTU4, TCNTS starts down-counting, and when TCNTS matches TCDR, the operation switches to up-counting. When TCNTS matches MTU3.TGRA, it is cleared to 0000h.

When MTU4.TCNT matches TDDR during down-counting of MTU3.TCNT and MTU4.TCNT, TCNTS starts up-counting, and when TCNTS matches TDDR, the operation switches to down-counting. When TCNTS reaches 0000h, it is set with the value in MTU3.TGRA.

TCNTS is compared with the compare register and temporary register, in which the PWM duty is specified, only during the count operation.

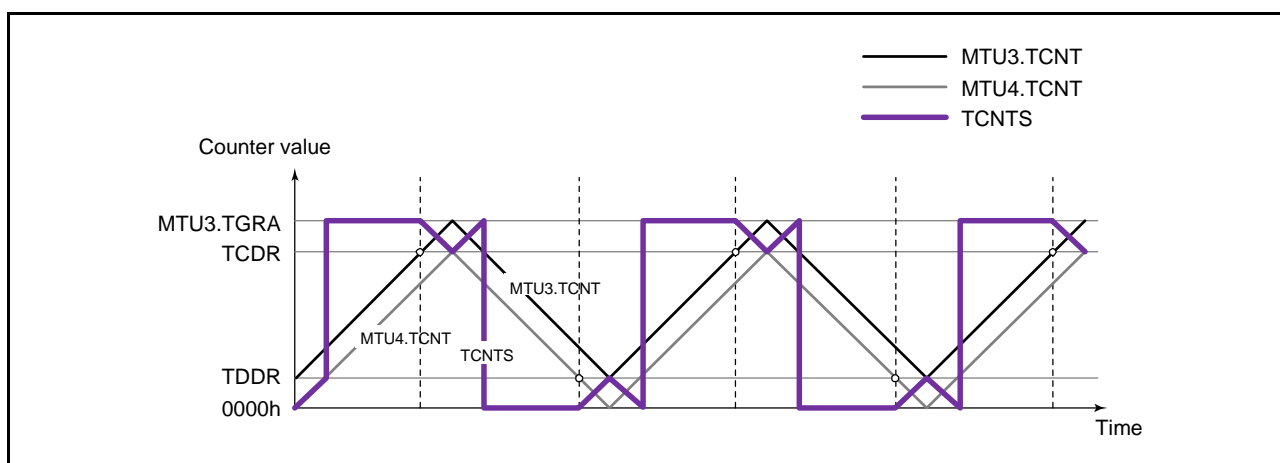


Figure 22.39 Counter Operation in Complementary PWM Mode



### (b) Register Operation

In complementary PWM mode, nine registers (compare registers, buffer registers, and temporary registers) are used. Figure 22.40 shows an example of operation in complementary PWM mode.

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 bits OLSN and OLSP in the timer output control register 1 (TOCR1) is output.

MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD are buffer registers for these compare registers. Between a buffer register and a compare register, there is a temporary register. The temporary registers cannot be accessed by the CPU. 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.

The data written to a buffer register is constantly transferred to the temporary register in the Ta interval. Data is not transferred to the temporary register in the Tb interval. Data written to a buffer register in this interval is transferred to the temporary register at the end of the Tb interval.

The value transferred to a temporary register is transferred to the compare register when TCNTS for which the Tb interval ends matches MTU3.TGRA while TCNTS is counting up, or 0000h while counting down. The timing for transfer from the temporary register to the compare register can be selected with bits MD[3:0] in the timer mode register (TMDR). Figure 22.40 shows an example in which the trough is selected for the transfer timing.

In the Tb (Tb2 in Figure 22.40) interval in which data is not transferred to the temporary register, 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 TCNTS) and two registers (compare register and temporary register) are compared, and PWM output is controlled accordingly.

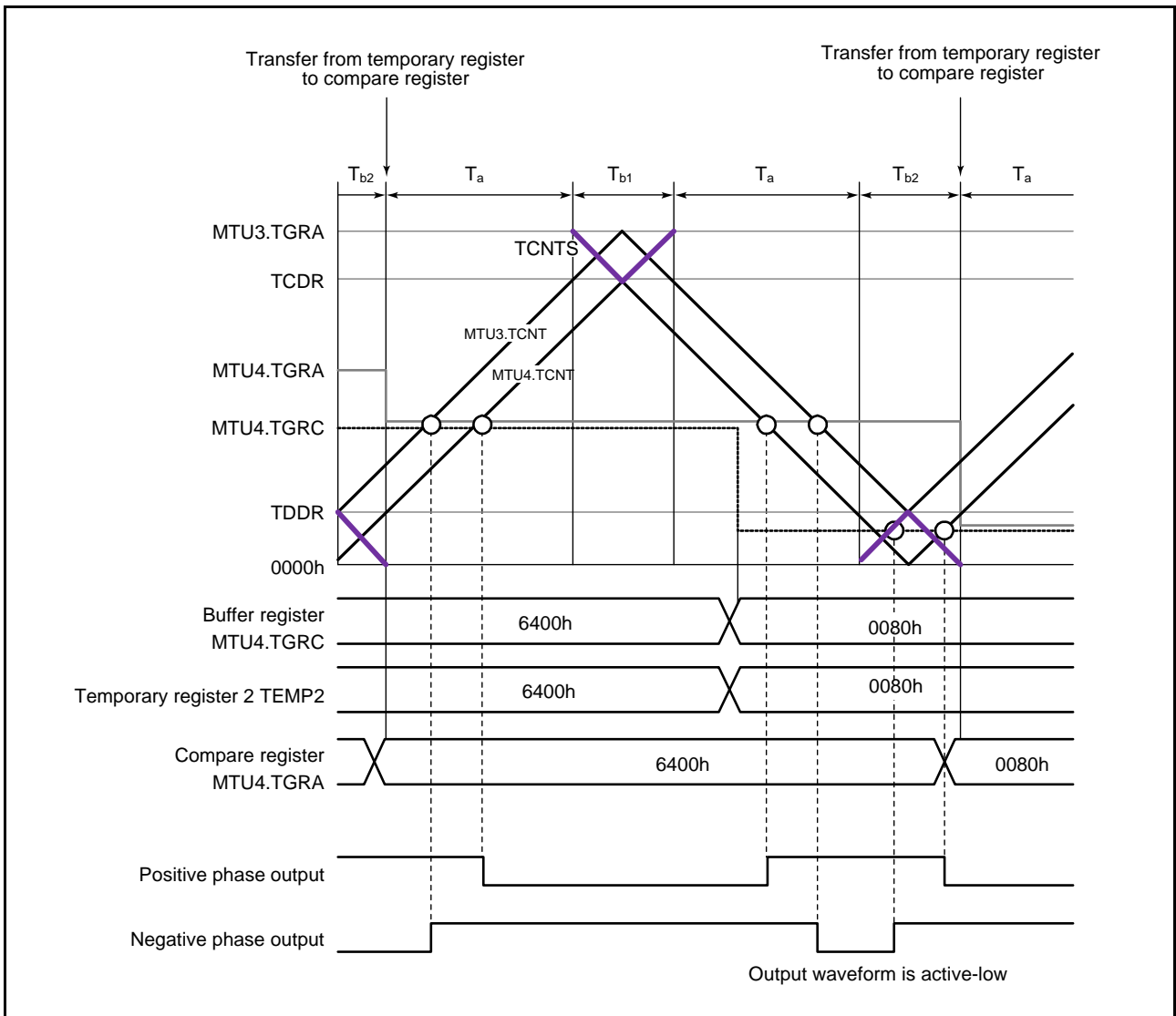


Figure 22.40 Example of Operation in Complementary PWM Mode

(c) Initial Setting

In complementary PWM mode, there are six 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 bits MD[3:0] in the timer mode register (TMDR), initial values should be set in the following registers.

MTU3.TGRC operates as the buffer register for MTU3.TGRA, and should be set with 1/2 the PWM carrier cycle + dead time  $T_d$ . The timer cycle buffer register (TCBR) operates as the buffer register for the timer cycle data register (TCDR), and should be set with 1/2 the PWM carrier cycle. Set dead time  $T_d$  in the timer dead time data register (TDDR).

When dead time is not needed, the TDER bit in the timer dead time enable register (TDER) should be cleared to 0, MTU3.TGRC and MTU3.TGRA should be set to 1/2 the PWM carrier cycle + 1, and TDDR should be set to 1. Set the respective initial PWM duty values in three buffer registers MTU3.TGRD, MTU4.TGRC, and MTU4.TGRD. The values set in the five buffer registers excluding TDDR 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 22.56 Registers and Counters Requiring Initial Setting**

Register and Counter	Setting
MTU3.TGRC	1/2 PWM carrier cycle + dead time Td (1/2 PWM carrier cycle + 1 when dead time generation is disabled by TDER)
TDDR	Dead time Td (1 when dead time generation is disabled by TDER)
TCBR	1/2 PWM carrier cycle
MTU3.TGRD, MTU4.TGRC, MTU4.TGRD	Initial PWM duty value for each phase
MTU4.TCNT	0000h

Note: • The value set in MTU3.TGRC should be the sum of 1/2 the PWM carrier cycle set in TCBR and dead time Td set in TDDR. When dead time generation is disabled by TDER, TGRC should be set to 1/2 the PWM carrier cycle + 1.

#### (d) PWM Output Level Setting

In complementary PWM mode, the PWM pulse output level is set with bits OLSN and OLSP in timer output control register 1 (TOCR1) or bits OLS1P to OLS3P and OLS1N to OLS3N in timer output control register 2 (TOCR2).

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, PWM pulses are output with a non-overlapping relationship between the positive and negative phases. This non-overlap time is called the dead time.

The non-overlap time is set in the timer dead time data register (TDDR). The value set in TDDR 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 TDDR.

#### (f) Dead Time Suppressing

Dead time generation is suppressed by clearing the TDER bit in the timer dead time enable register (TDER) to 0. TDER bit can be cleared 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 carrier cycle + 1 and the timer dead time data register (TDDR) should be set to 1.

By the above settings, PWM waveforms without dead time can be obtained. Figure 22.41 shows an example of operation without dead time.

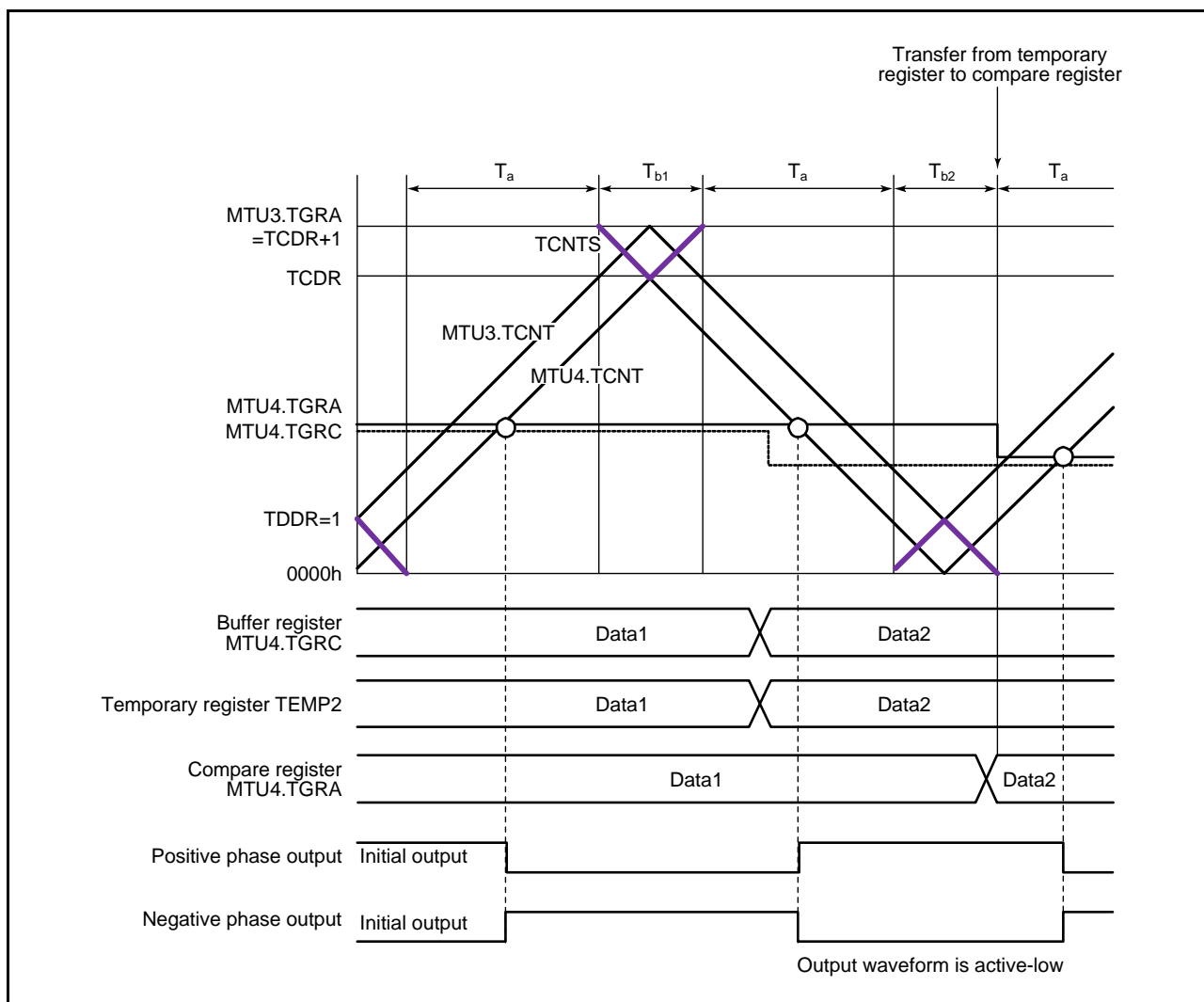


Figure 22.41 Example of Operation without Dead Time

(g) PWM Cycle Setting

In complementary PWM mode, the PWM pulse cycle is set in two registers—MTU3.TGRA, in which the MTU3.TCNT upper limit value is set, and TCDR, 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:

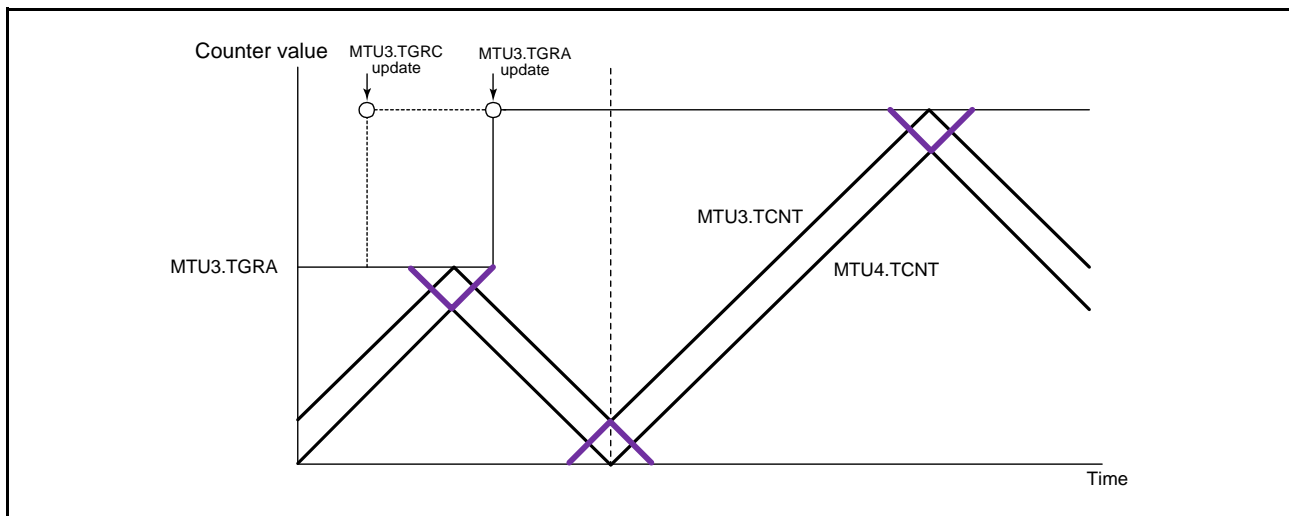
With dead time:  $MTU3.TGRA \text{ setting} = TCDR \text{ setting} + TDDR \text{ setting}$

Without dead time:  $MTU3.TGRA \text{ setting} = TCDR \text{ setting} + 1$

The MTU3.TGRA and TCDR settings are made by setting values in buffer registers MTU3.TGRC and TCBR. The values set in MTU3.TGRC and TCBR are transferred simultaneously to MTU3.TGRA and TCDR with the transfer timing selected with bits MD[3:0] in the timer mode register (TMDR).

The new PWM cycle is reflected from the next cycle when data is updated at the crest, or from the current cycle when updated in the trough. Figure 22.42 illustrates the operation when the PWM cycle is updated at the crest.

See the following section (h), Register Data Updating, for the method of updating the data in each buffer register.



**Figure 22.42** Example of PWM Cycle Updating

#### (h) Register Data Updating

In complementary PWM mode, the buffer register is used to update the data in a compare register. The update data can be written to the buffer register at any time. There are five registers (PWM duty and carrier cycle registers) that have buffer registers and can be updated during operation.

There is a temporary register between each of these registers and its buffer register. While subcounter TCNTS 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 TCNTS is counting; in this case, the value written to a buffer register is transferred after TCNTS halts.

The temporary register value is transferred to the compare register at the data update timing set with bits MD[3:0] in the timer mode register (TMDR). Figure 22.43 shows an example of data updating in complementary PWM mode. 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.

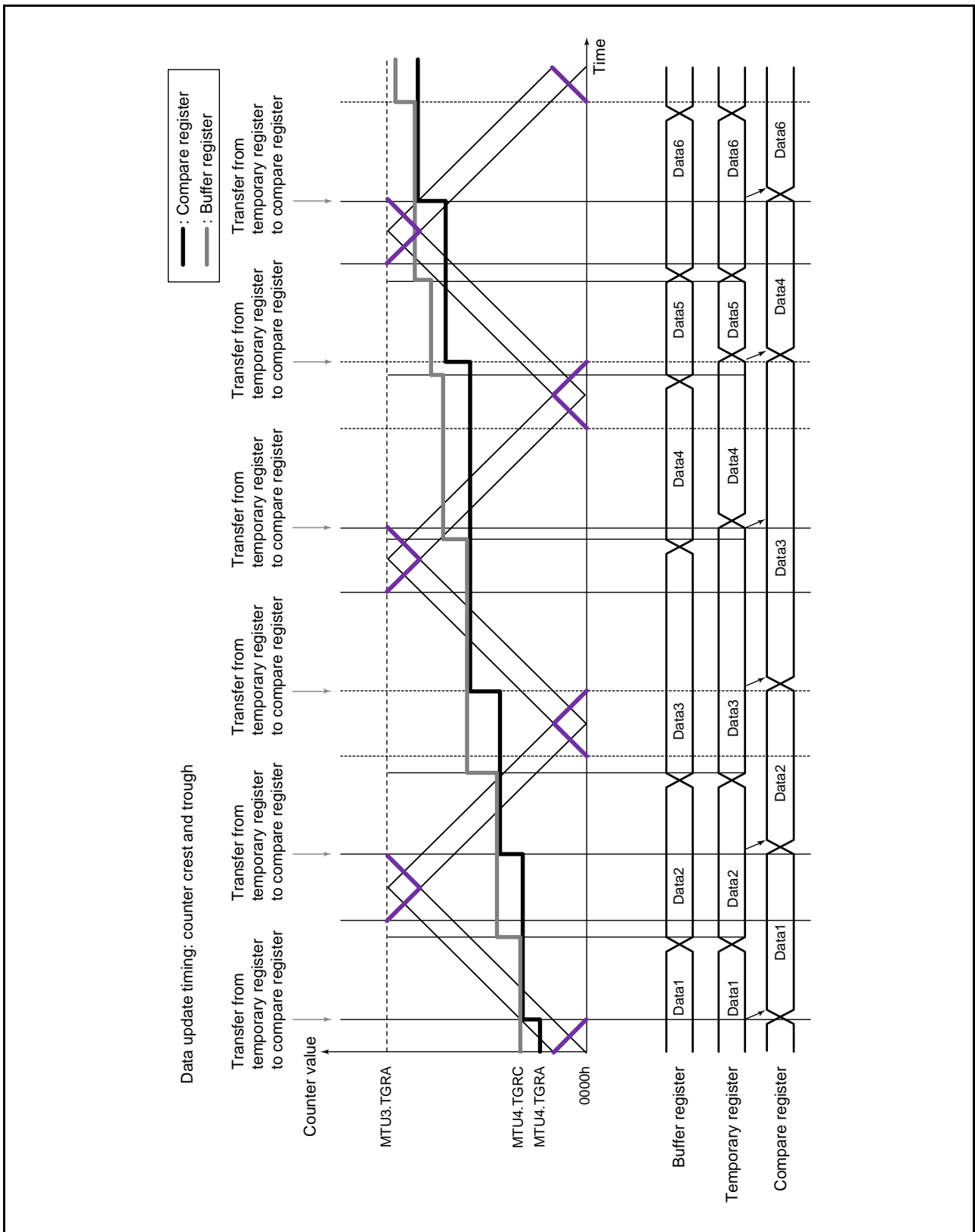


Figure 22.43 Example of Data Updating in Complementary PWM Mode

(i) Initial Output in Complementary PWM Mode

In complementary PWM mode, the initial output is determined by the setting of bits OLSN and OLSP in timer output control register 1 (TOCR1) or bits OLS1N to OLS3N and OLS1P to OLS3P in timer output control register 2 (TOCR2). This initial output is the non-active level of the PWM pulse and continues from when complementary PWM mode is set with the timer mode register (TMDR) until MTU4.TCNT exceeds the value set in the dead time register (TDDR). Figure 22.44 shows an example of the initial output in complementary PWM mode.

An example of the waveform when the initial PWM duty value is smaller than the TDDR value is shown in Figure 22.45.

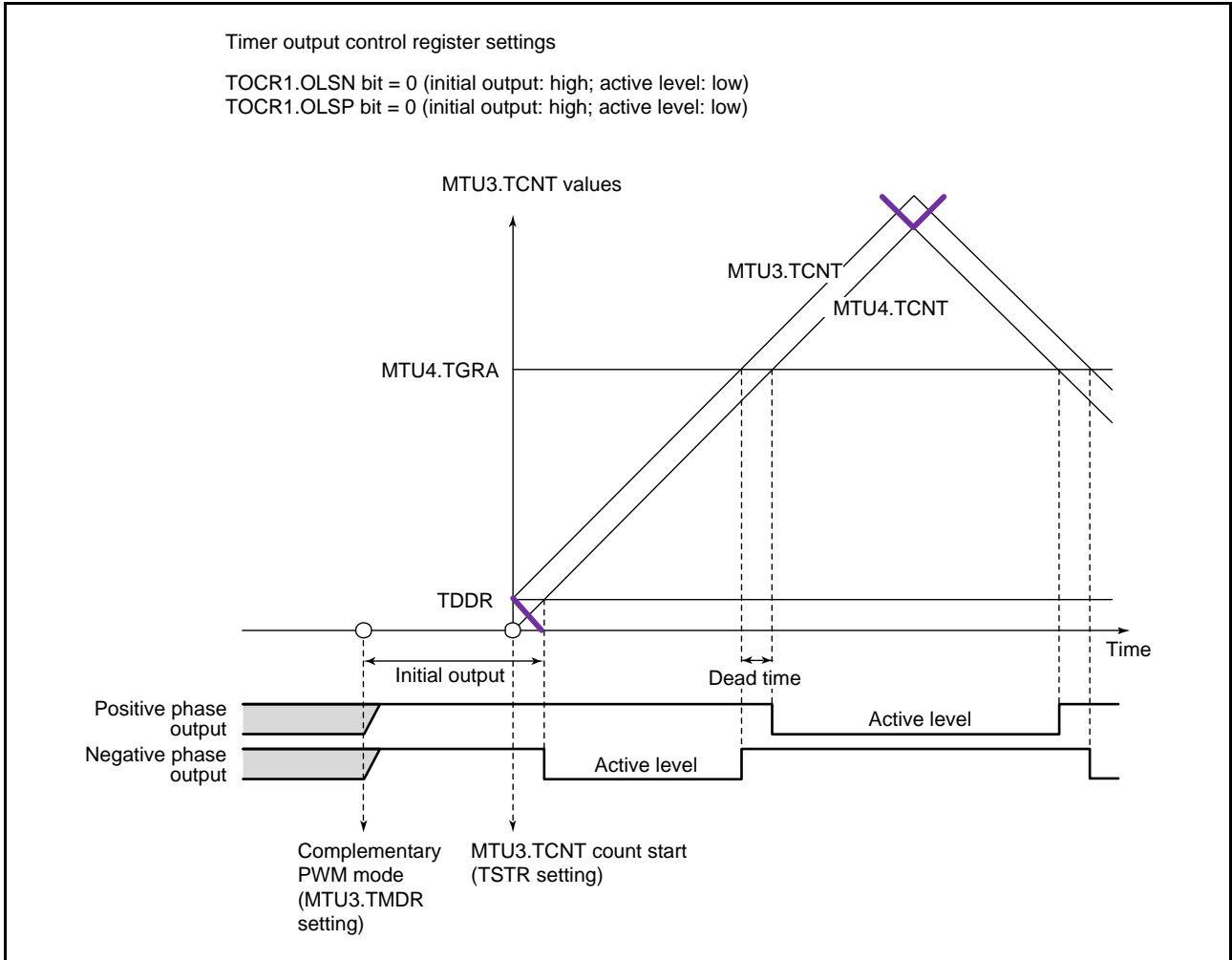


Figure 22.44 Example of Initial Output in Complementary PWM Mode (1)

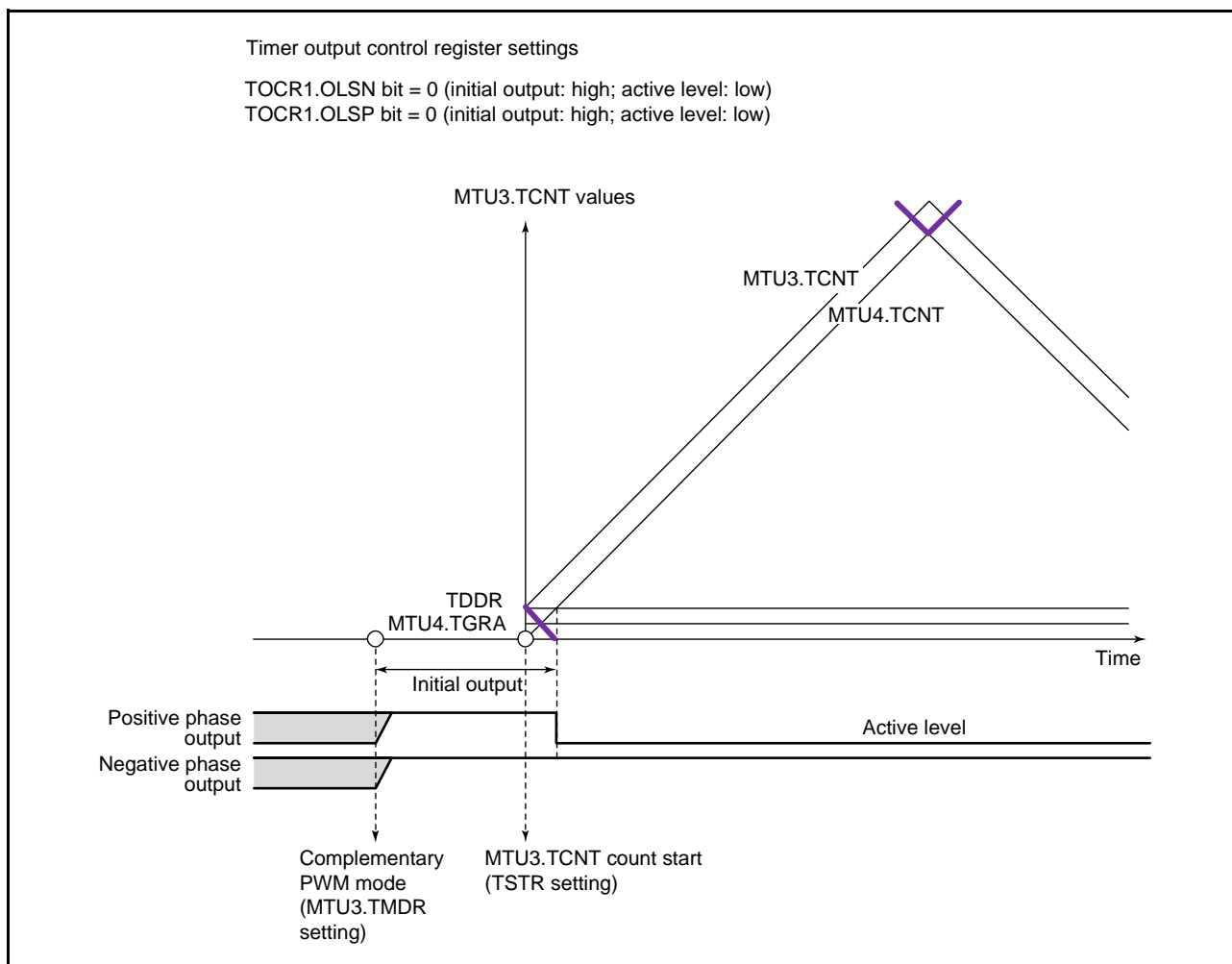


Figure 22.45 Example of Initial Output in Complementary PWM Mode (2)

(j) Method for Generating PWM Output in Complementary PWM Mode

In complementary PWM mode, three phases of PWM waveforms are output with a non-overlap time between the positive and negative phases. This non-overlap time is called the dead time.

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 TCNTS is counting, the compare register and temporary register values are simultaneously compared to create consecutive PWM pulses from 0 to 100%. 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 22.46 to Figure 22.48 show examples of waveform generation in complementary PWM mode.

The positive-phase and negative-phase signals are turned off on compare matches with the value of the counter (MTU3.TCNT), and are turned on compare matches with the MTU4.TCNT counter plus a delay of the dead time. 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 22.46. 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 22.47, 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 higher 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 22.48, compare match a' with new data in the temporary register occurs before compare match c, but until compare match c, which turns off the positive phase, other compare matches are ignored. As a result, the negative phase is not turned on.

Thus, in complementary PWM mode, compare matches at turn-off timings take precedence, and turn-on timing compare matches that occur before a turn-off timing compare match are ignored.

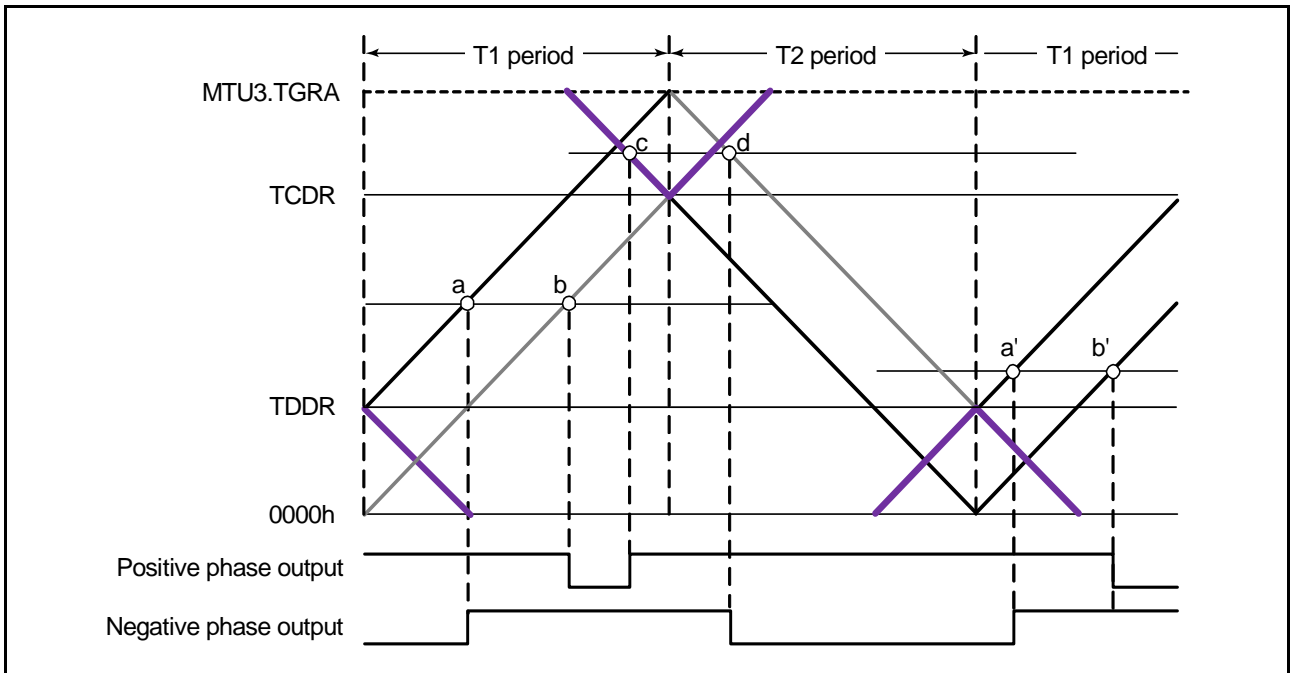


Figure 22.46 Example of Waveform Output in Complementary PWM Mode (1)

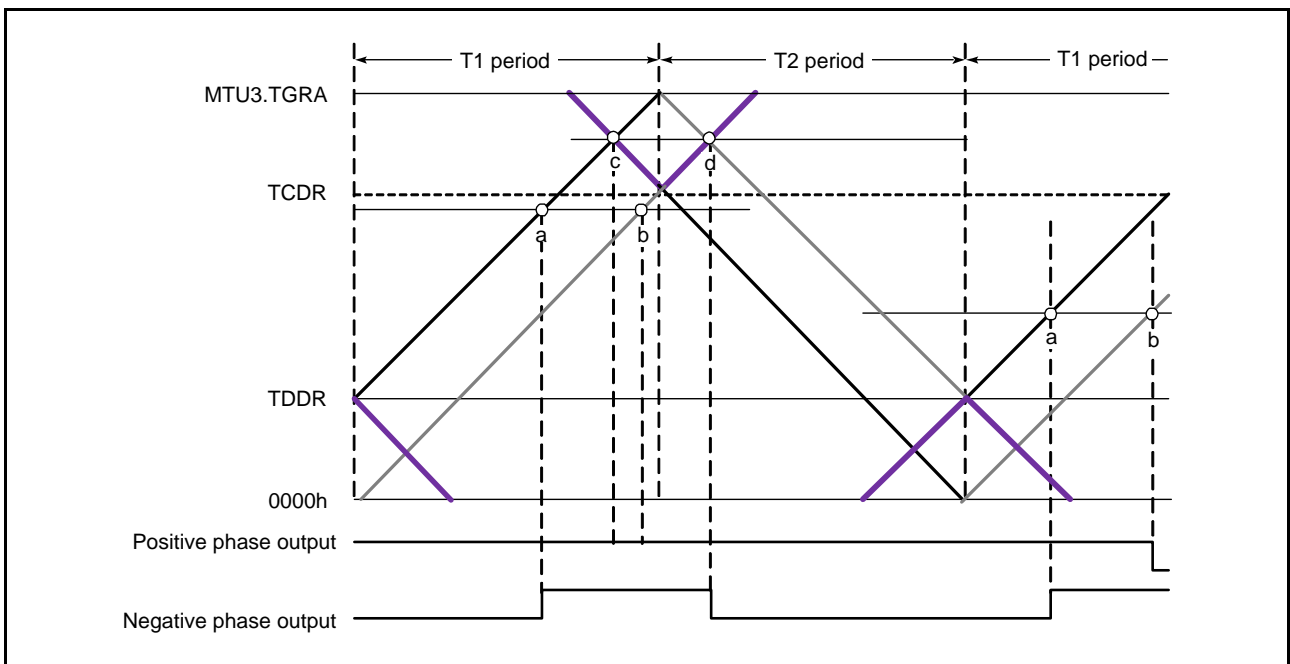


Figure 22.47 Example of Waveform Output in Complementary PWM Mode (2)

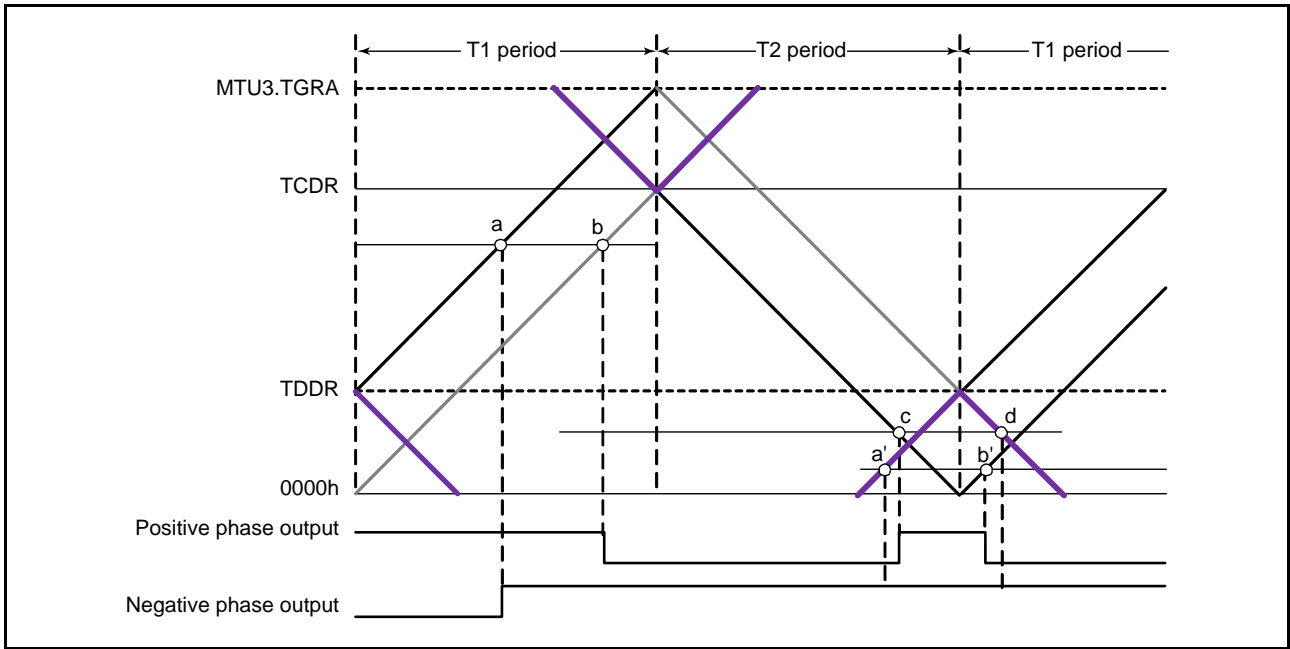


Figure 22.48 Example of Waveform Output in Complementary PWM Mode (3)

(k) 0% and 100% Duty Cycle Output in Complementary PWM Mode

In complementary PWM mode, 0% and 100% duty cycle PWM waveforms can be output as required. Figure 22.49 to Figure 22.53 show output examples.

A 100% duty cycle 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 cycle 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.

On and 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.

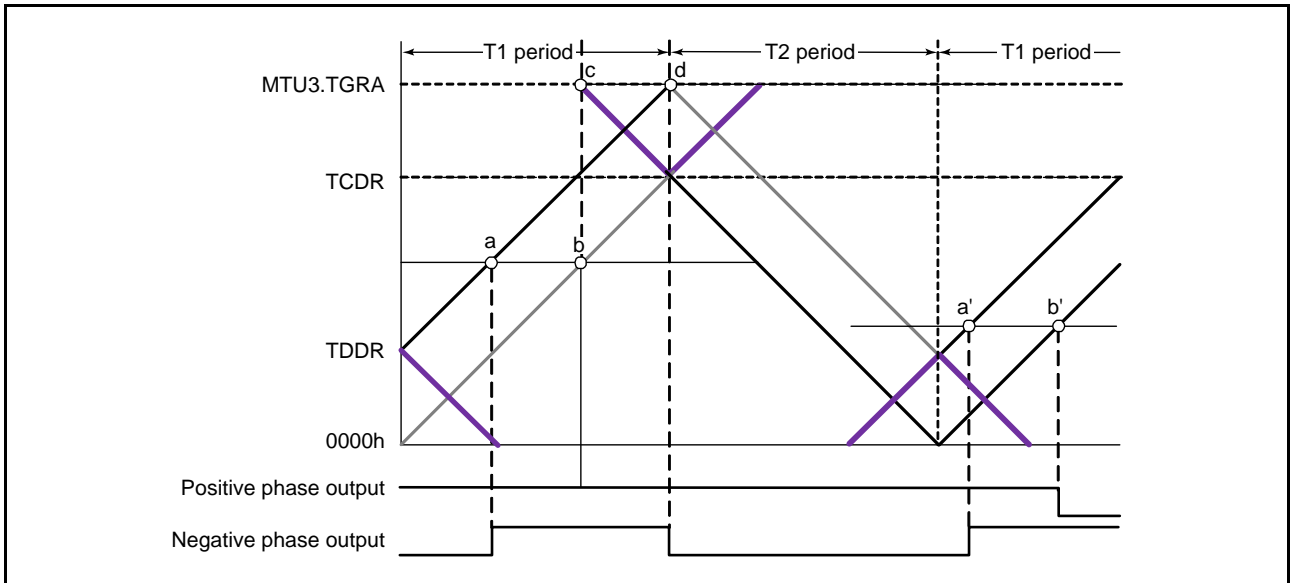


Figure 22.49 Example of 0% and 100% Waveform Output in Complementary PWM Mode (1)

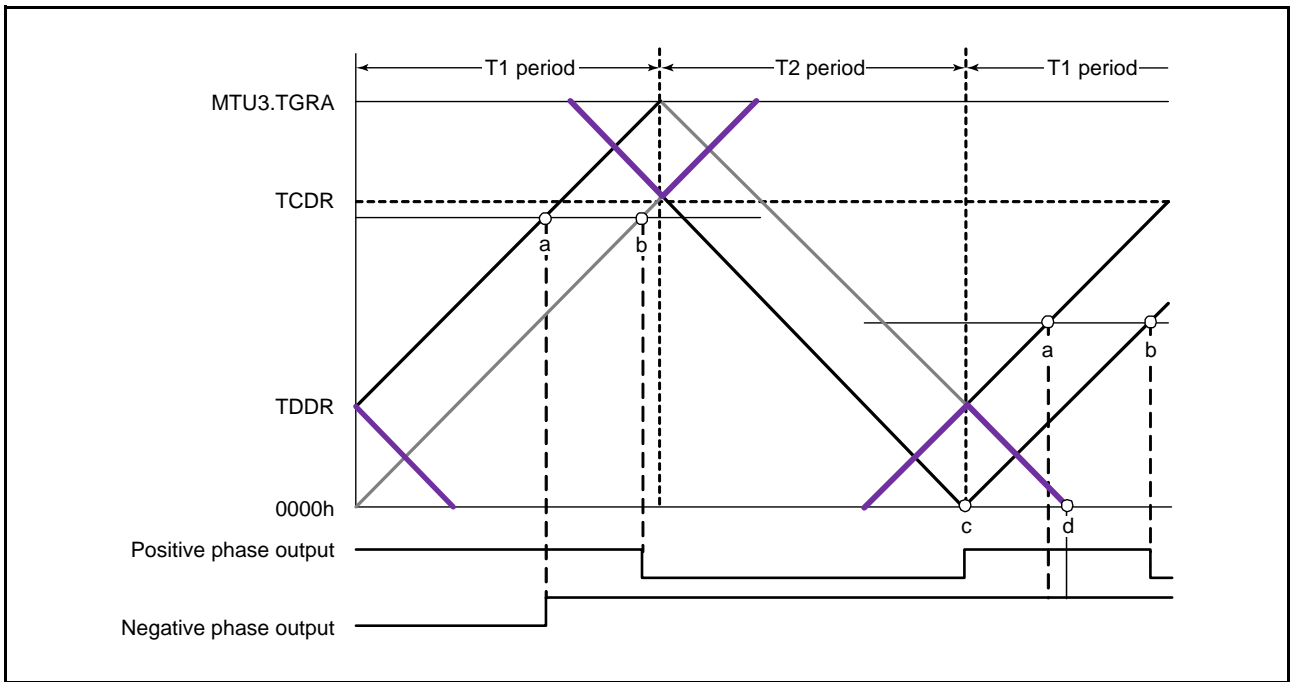


Figure 22.50 Example of 0% and 100% Waveform Output in Complementary PWM Mode (2)

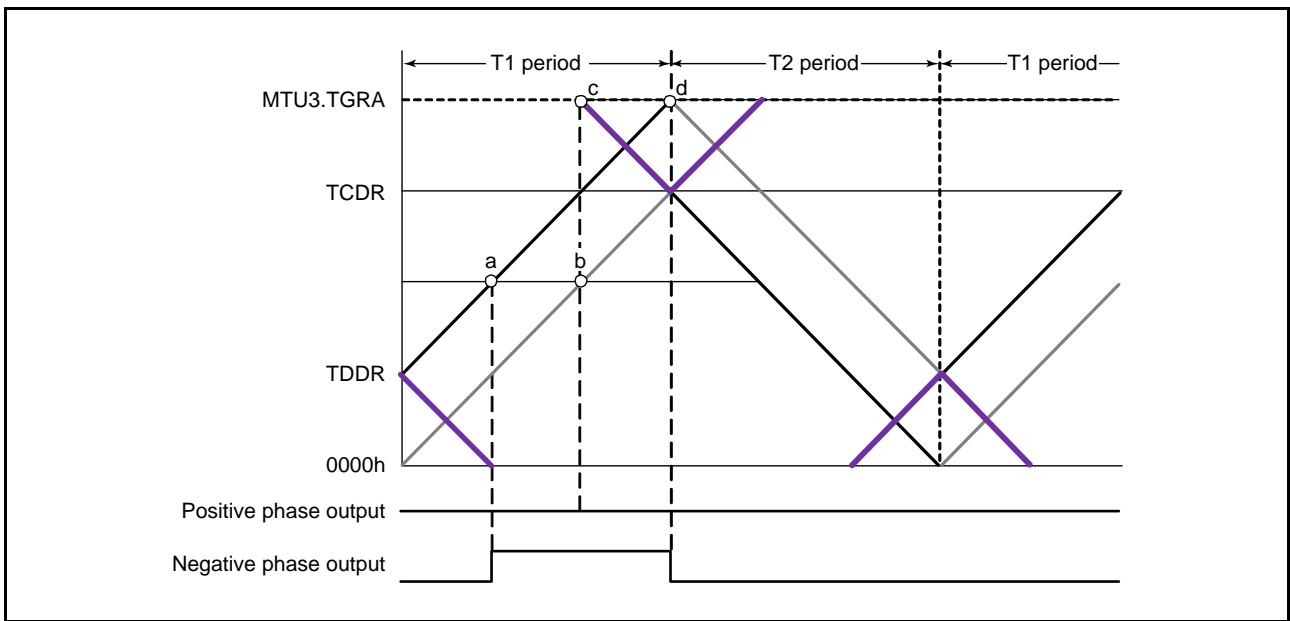


Figure 22.51 Example of 0% and 100% Waveform Output in Complementary PWM Mode (3)

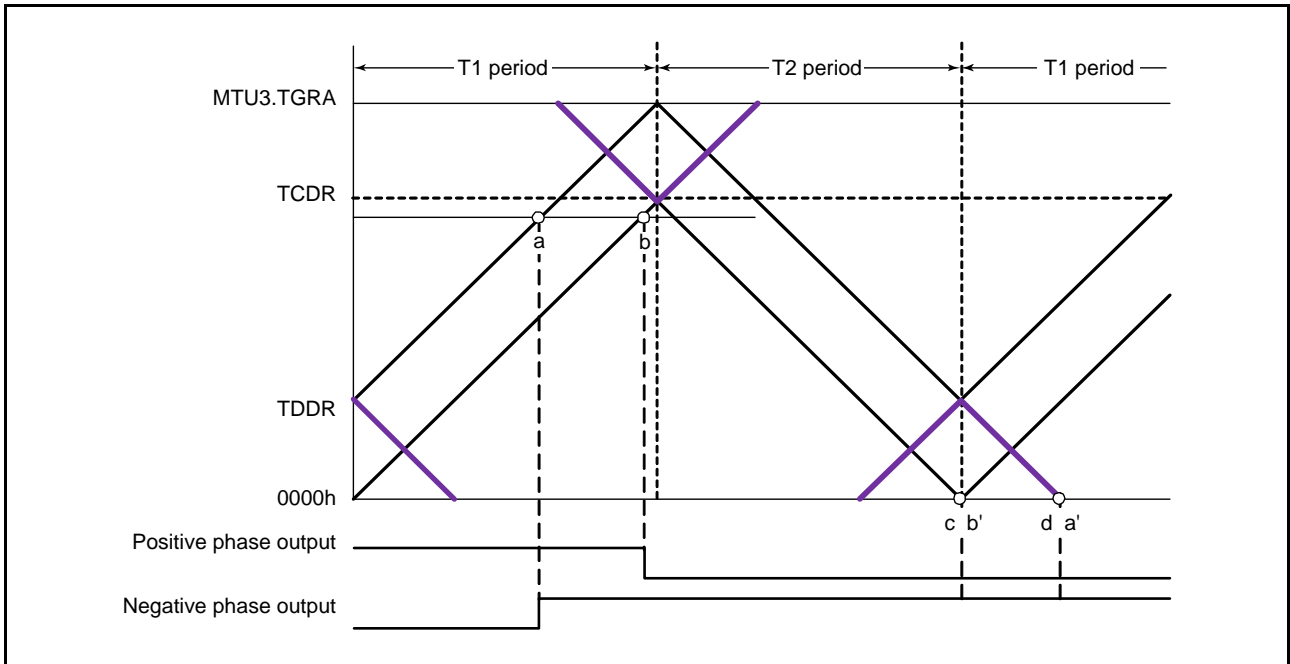


Figure 22.52 Example of 0% and 100% Waveform Output in Complementary PWM Mode (4)

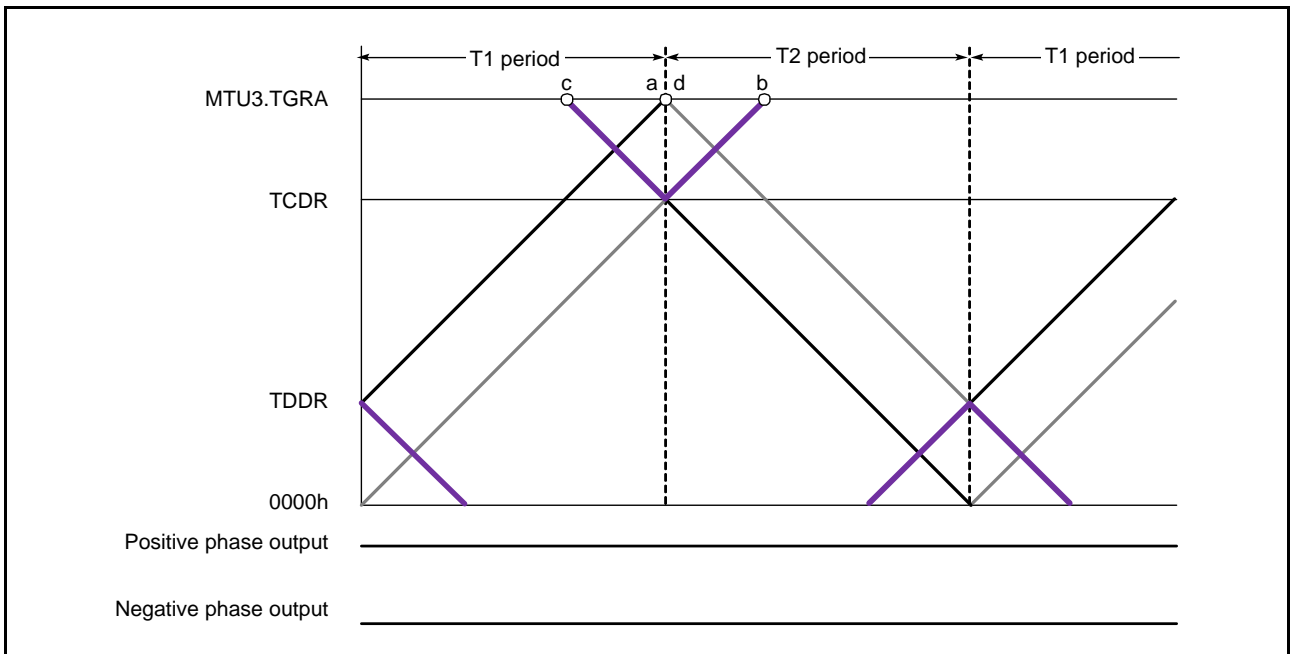


Figure 22.53 Example of 0% and 100% Waveform Output in Complementary PWM Mode (5)

(I) Toggle Output Synchronized with PWM Cycle

In complementary PWM mode, toggle output in synchronization with the PWM carrier cycle can be generated by setting the PSYE bit to 1 in the timer output control register (TOCR). An example of a toggle output waveform is shown in Figure 22.54.

This output is toggled by a compare match between MTU3.TCNT and MTU3.TGRA and a compare match between MTU4.TCNT and 0000h.

The MTIOC3A pin is assigned for this toggle output. The initial output is 1.

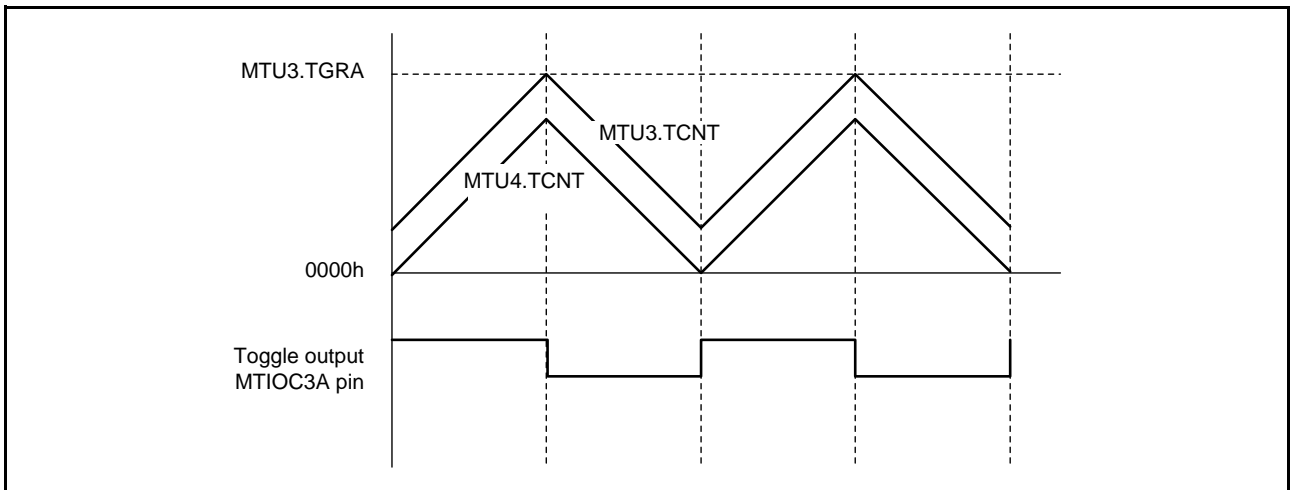


Figure 22.54 Example of Toggle Output Waveform Synchronized with PWM Output

(m) Counter Clearing by Another Channel

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTS can be cleared by another channel when a mode for synchronization with another channel is specified through the timer synchronous register (TSYR) and synchronous clearing is selected with bits CCLR[2:0] in the timer control register (TCR).

Figure 22.55 illustrates an example of this operation.

Use of this function enables a counter to be cleared and restarted through an external signal.

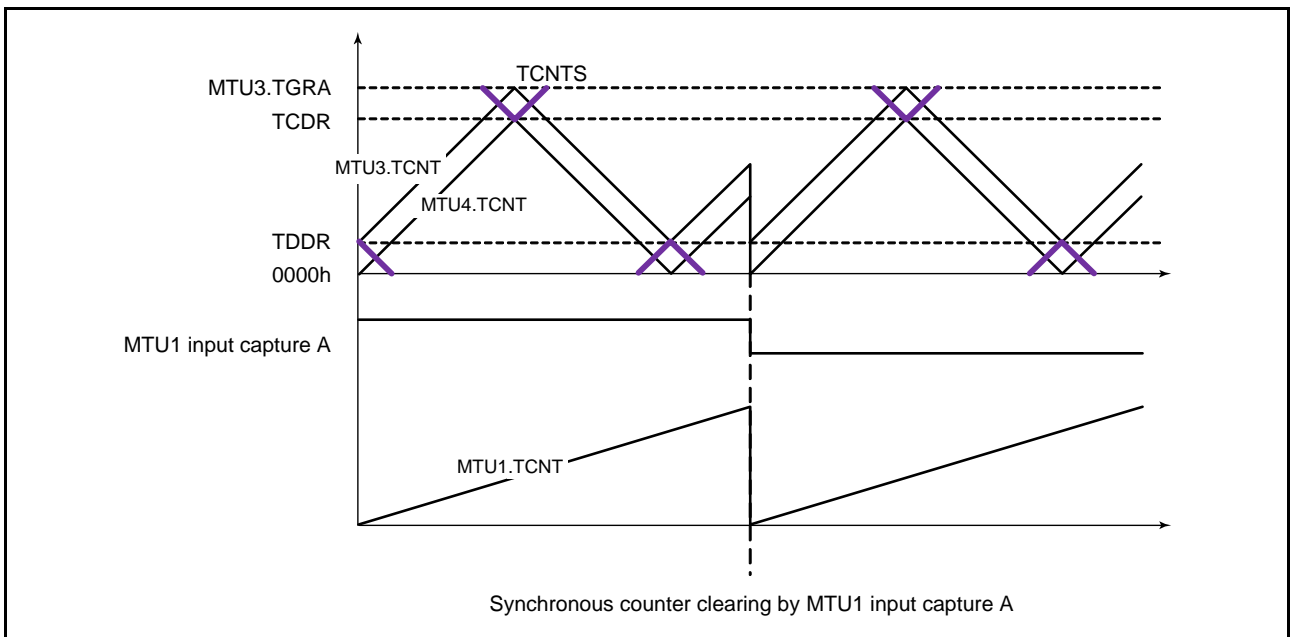


Figure 22.55 Counter Clearing Synchronized with Another Channel

(n) Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

Setting the WRE bit in TWCR to 1 suppresses initial output when synchronous counter clearing occurs in the Tb interval at the trough in complementary PWM mode and controls abrupt change in duty cycle at synchronous counter clearing. Initial output suppression through setting WRE bit to 1 is applicable only when synchronous clearing occurs in the Tb interval at the trough as indicated by (10) or (11) in Figure 22.56. When synchronous clearing occurs outside that interval, the initial value specified by the OLSN and OLSP bits in TOCR1 is output. Even in the Tb interval at the trough, if synchronous clearing occurs in the initial output period (indicated by (1) in Figure 22.56) immediately after the counters start operation, initial value output is not suppressed.

Synchronous clearing generated in MTU0 to MTU2 can cause counter clearing in the MTU.

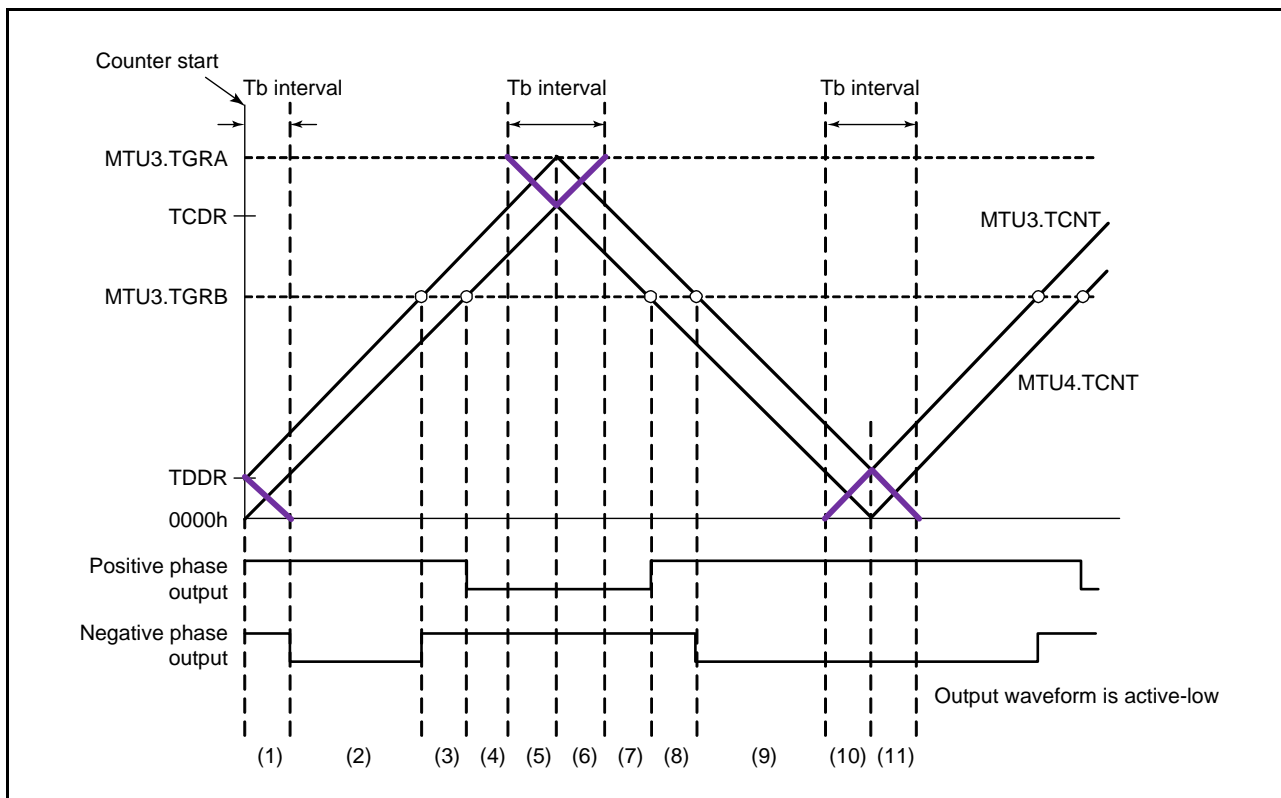
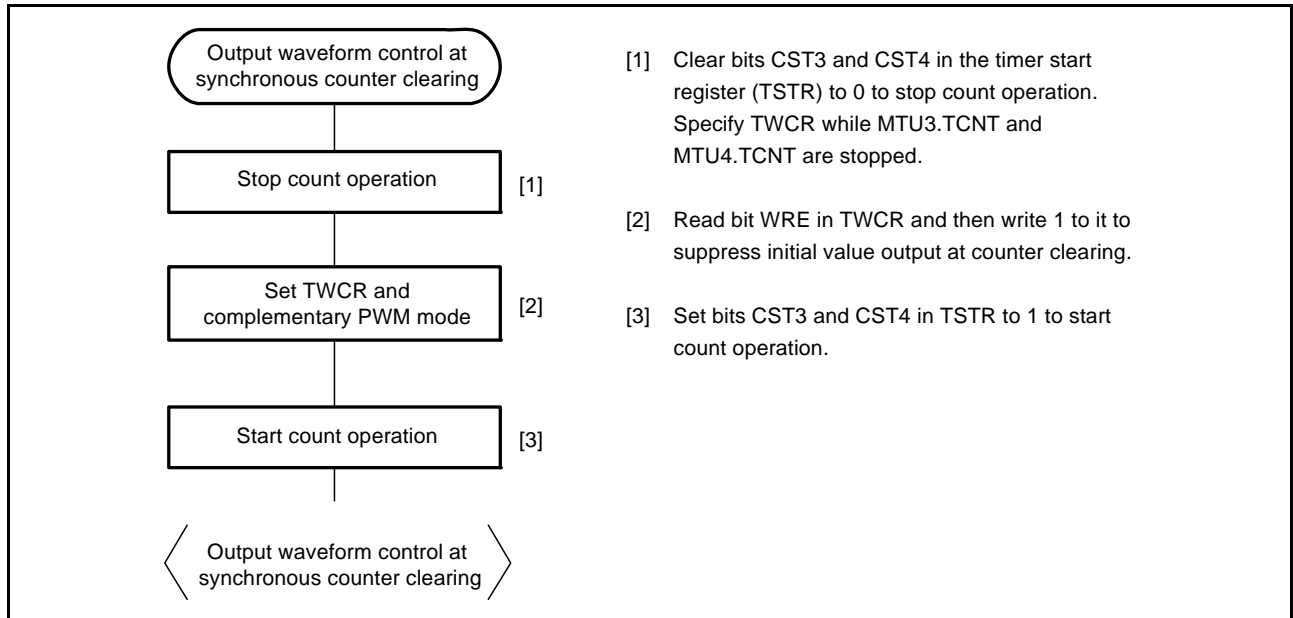


Figure 22.56 Timing for Synchronous Counter Clearing

- Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode.

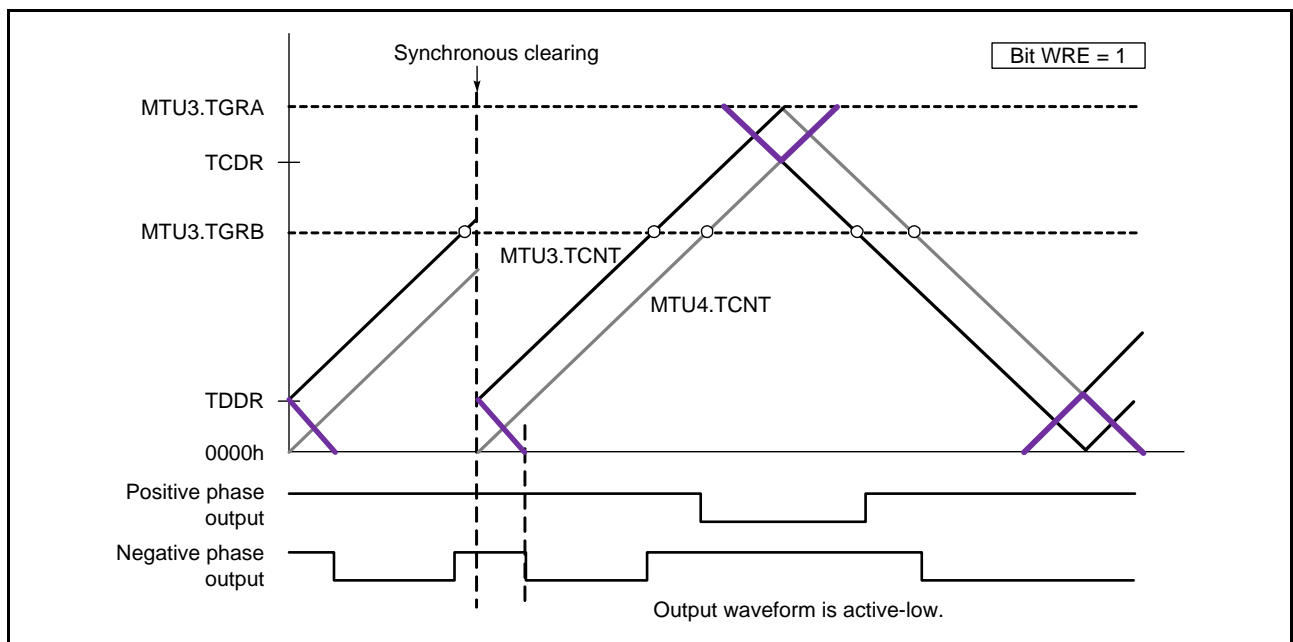
An example of the procedure for setting output waveform control at synchronous counter clearing in complementary PWM mode is shown in Figure 22.57.



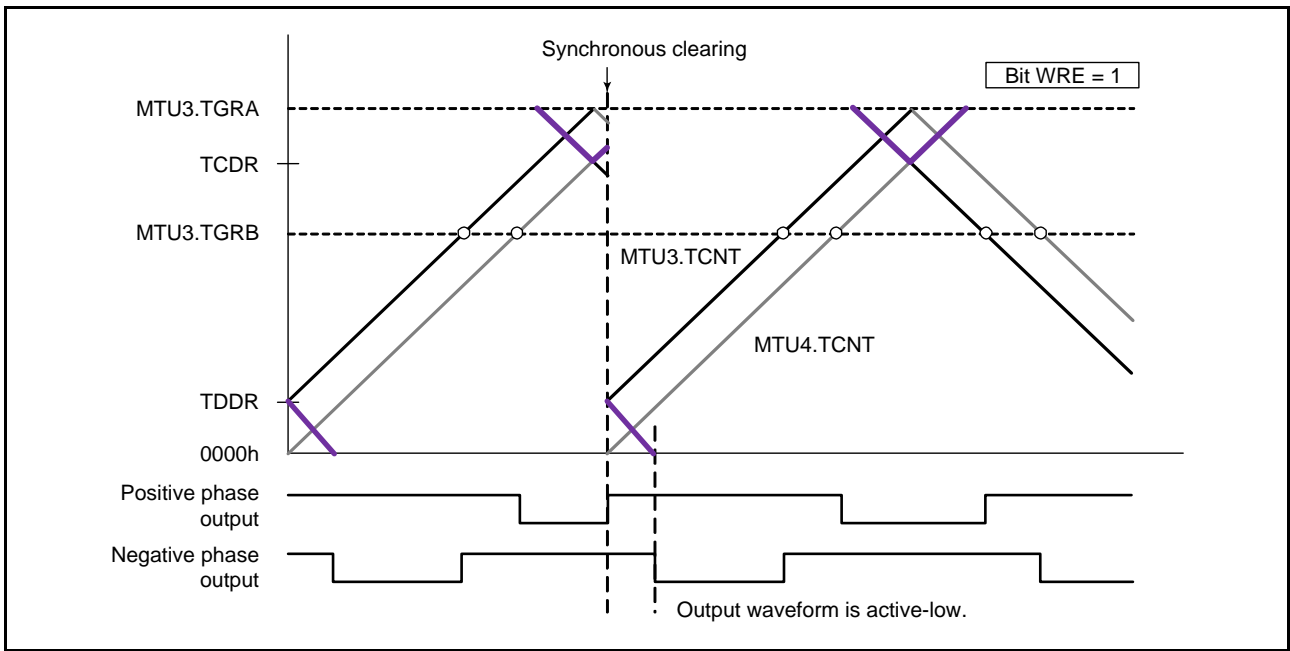
**Figure 22.57** Example of Procedure for Setting Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

- Examples of Output Waveform Control at Synchronous Counter Clearing in Complementary PWM Mode

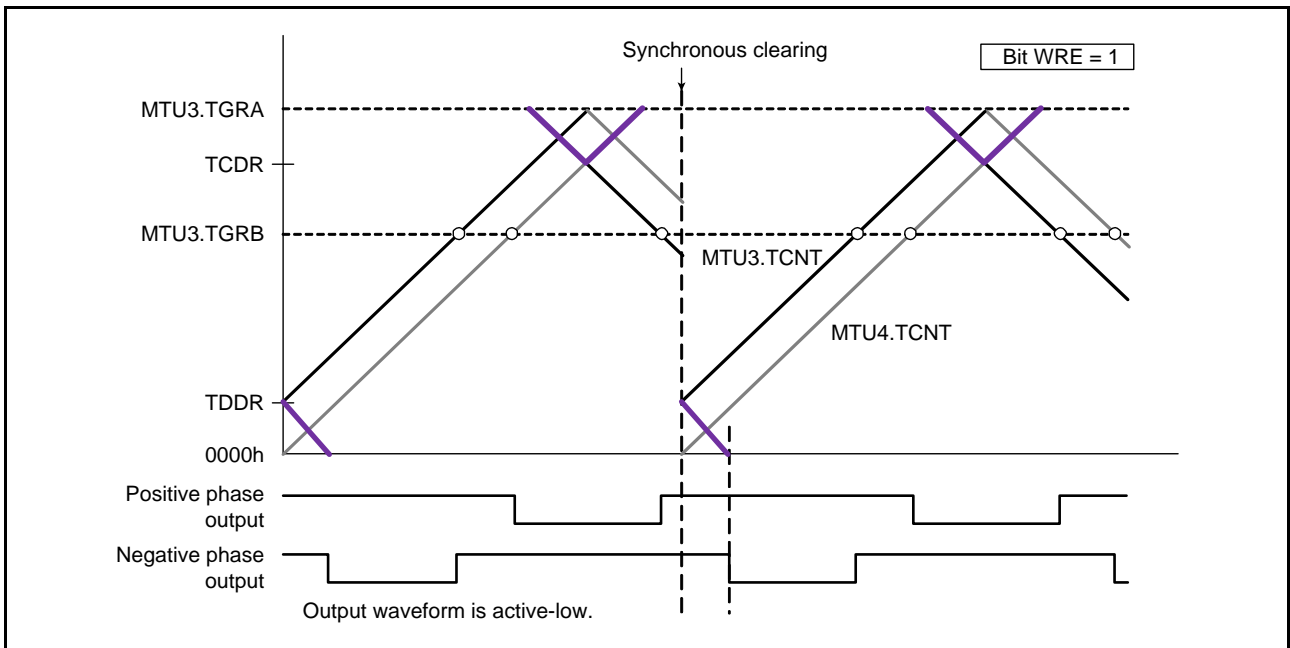
Figure 22.58 to Figure 22.61 show examples of output waveform control in which the MTU operates in complementary PWM mode and synchronous counter clearing is generated while the WRE bit in TWCR is set to 1. In the examples shown in Figure 22.58 to Figure 22.61, synchronous counter clearing occurs at timing (3), (6), (8), and (11) shown in Figure 22.56, respectively.



**Figure 22.58** Example of Synchronous Clearing in Dead Time during Up-Counting (Timing (3) in Figure 22.56; Bit WRE of TWCR is 1 in the MTU)

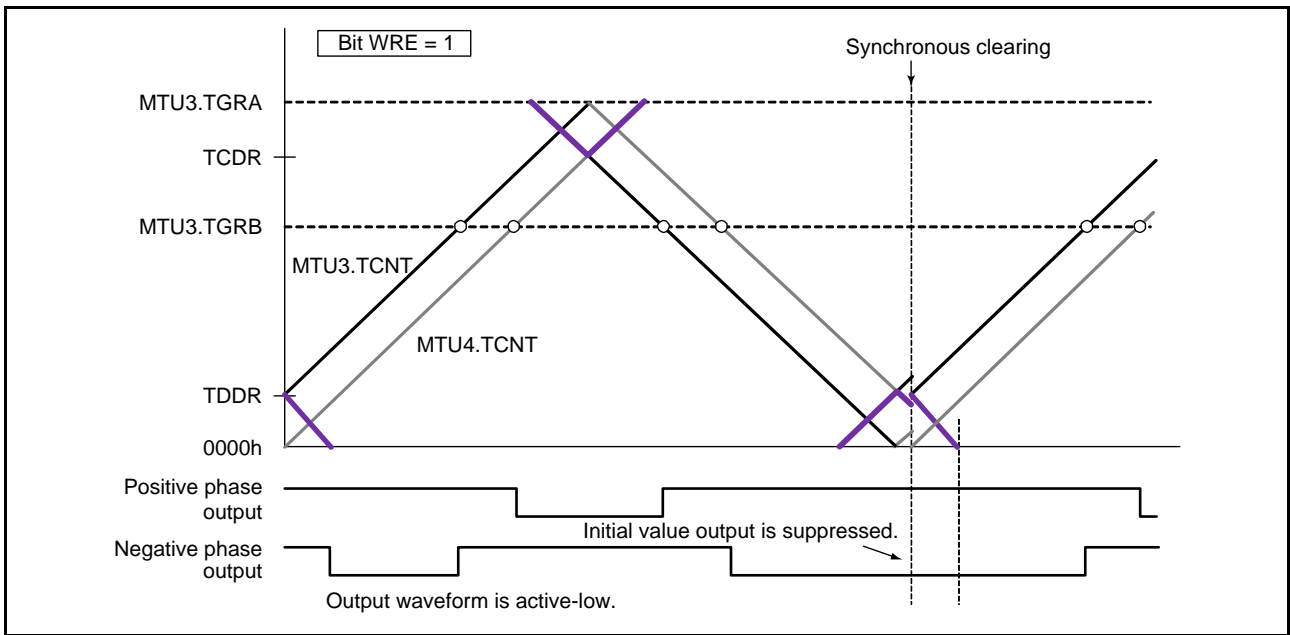


**Figure 22.59** Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 22.56; Bit WRE of TWCR is 1 in the MTU)



**Figure 22.60** Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 22.56; Bit WRE of TWCR is 1)





**Figure 22.61 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 22.56; Bit WRE of TWCR is 1)**

**(o) Counter Clearing by MTU3.TGRA Compare Match**

In complementary PWM mode, MTU3.TCNT, MTU4.TCNT, and TCNTS can be cleared by MTU3.TGRA compare match when the CCE bit is set in the timer waveform control register (TWCR).

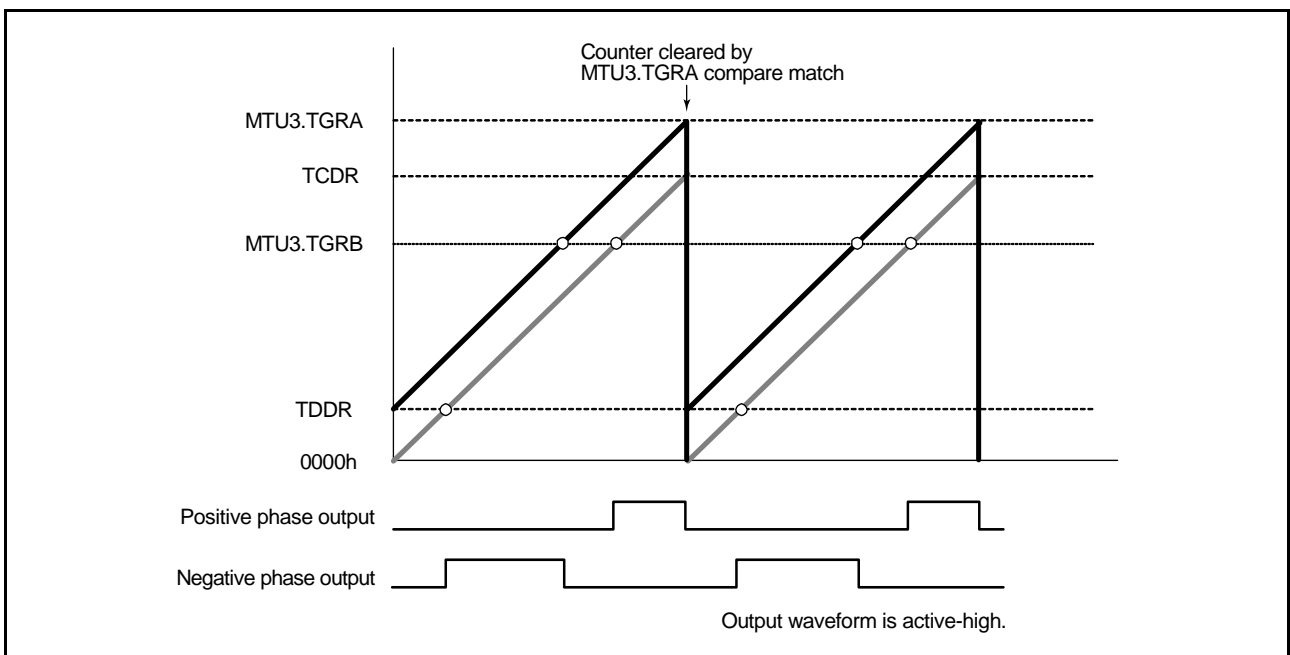
Figure 22.62 illustrates an operation example.

Note: • Use this function only in complementary PWM mode 1 (transfer at crest).

Note: • Do not specify synchronous clearing by another channel (do not set the SYNC0 to SYNC4 bits in the timer synchronous register (TSYR) to 1).

Note: • Do not set the PWM duty cycle value to 0000h.

Note: • Do not set the PSYE bit in timer output control register 1 (TOCR1) to 1.



**Figure 22.62 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, a brushless DC motor can easily be controlled using the timer gate control register (TGCR). Figure 22.63 to Figure 22.66 show examples of brushless DC motor driving waveforms created using TGCR. To switch the output phases for a 3-phase brushless DC motor by means of external signals detected with a Hall element, etc., clear the FB bit in TGCR 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 (set with PFS). When an edge is detected at pin MTIOC0A, MTIOC0B, or MTIOC0C, the output on/off state is switched automatically.

When the FB bit is 1, the output on/off state is switched when the UF, VF, or WF bit in TGCR is cleared to 0 or set to 1. The driving waveforms are output from the 6-phase output pins for complementary PWM mode.

With this 6-phase output, while the output is turned on, chopping output is available through complementary PWM mode output function by setting the N bit or P bit in TGCR 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 OLSN and OLSP bits in the timer output control register 1 (TOCR1) regardless of the setting of the N and P bits.

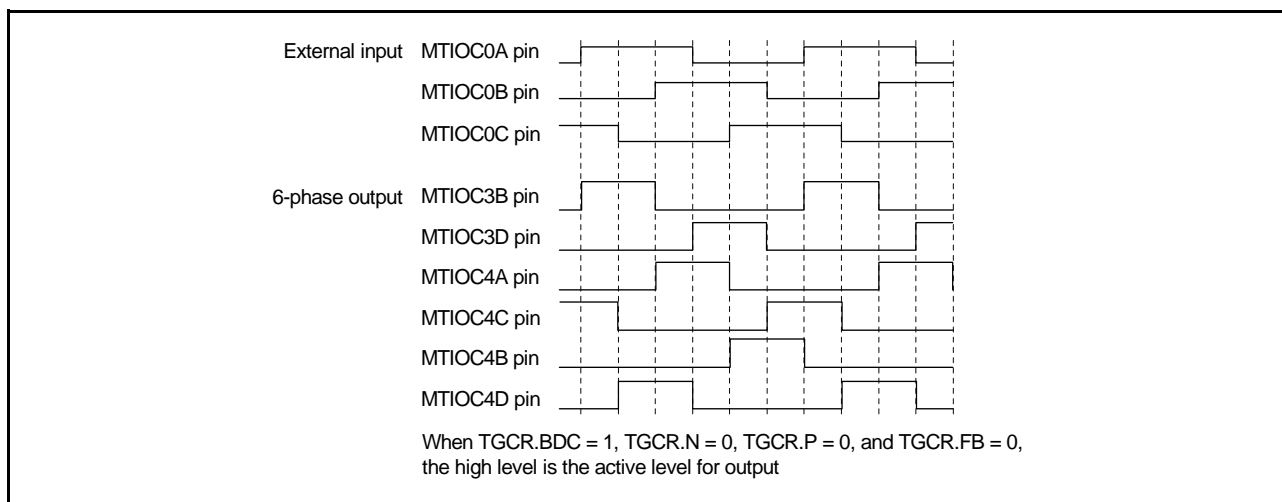


Figure 22.63 Example of Output Phase Switching by External Input (1)

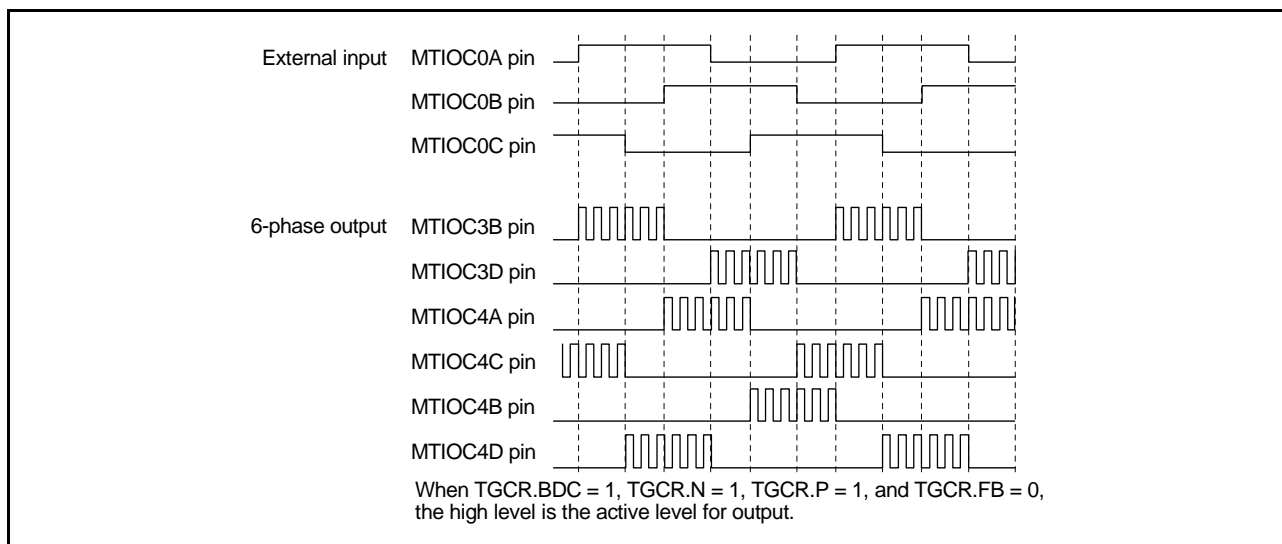


Figure 22.64 Example of Output Phase Switching by External Input (2)

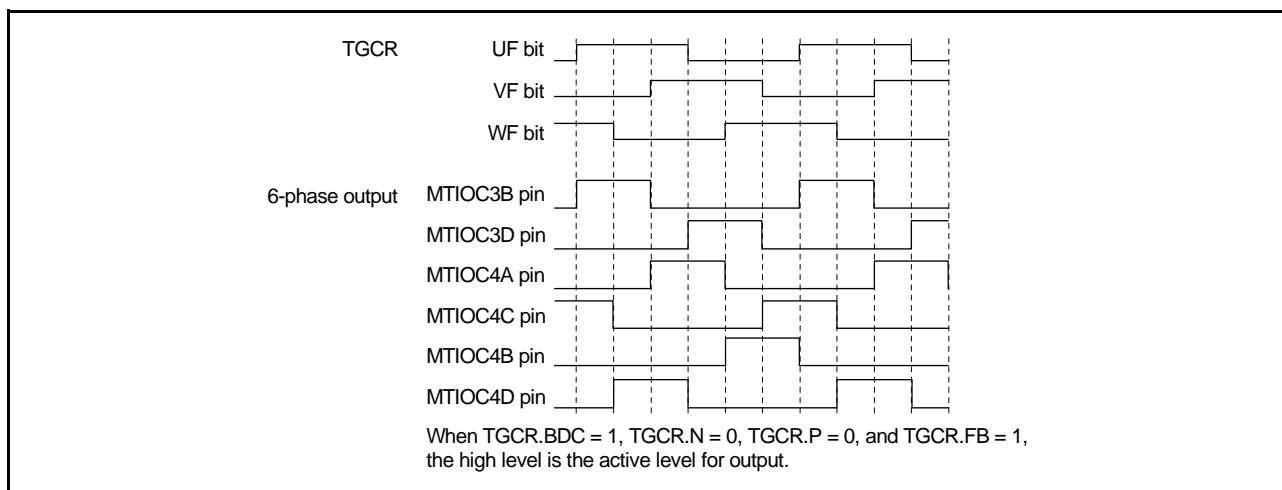


Figure 22.65 Example of Output Phase Switching through UF, VF, and WF Bit Settings (1)

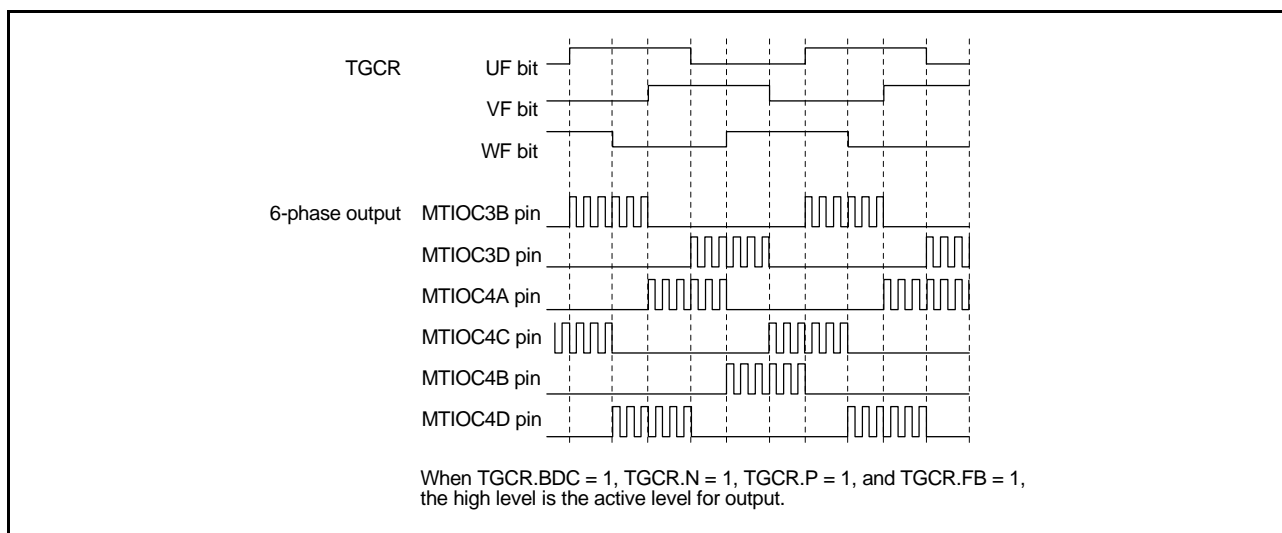


Figure 22.66 Example of Output Phase Switching through UF, VF, and WF Bit Settings (2)

(q) A/D Converter Start Request Setting

In complementary PWM mode, an A/D converter 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 converter start requests can be specified by setting the TTGE bit to 1 in the timer interrupt enable register (TIER). To issue an A/D converter start request at an MTU4.TCNT underflow (trough), set the TTGE2 bit in MTU4.TIER to 1.

(3) Interrupt Skipping 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 timer interrupt skipping set register (TITCR).

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 timer buffer transfer register (TBTER). For the linkage with buffer registers, refer to description (c), Buffer Transfer Control Linked with Interrupt Skipping, below.

A/D converter start requests generated by the A/D converter start request delaying function can also be skipped in coordination with interrupt skipping by making settings in the timer A/D converter request control register (TADCR).

For the linkage with the A/D converter start request delaying function, refer to section 22.3.9, A/D Converter Start Request Delaying Function.

The timer interrupt skipping setting register (TITCR) should be set while the TGIA3 and TCIV4 interrupt requests are disabled by the settings of MTU3.TIER and MTU4.TIER under the conditions in which compare match never occur and TGIA3 and TGIA4 interrupt requests by compare match are never generated. Before changing the skipping count, be sure to clear the T3AEN and T4VEN bits in TITCR 0 to clear the skipping counter.

(a) Example of Interrupt Skipping Operation Setting Procedure

Figure 22.67 shows an example of the interrupt skipping operation setting procedure. Figure 22.68 shows the periods during which interrupt skipping count can be changed.

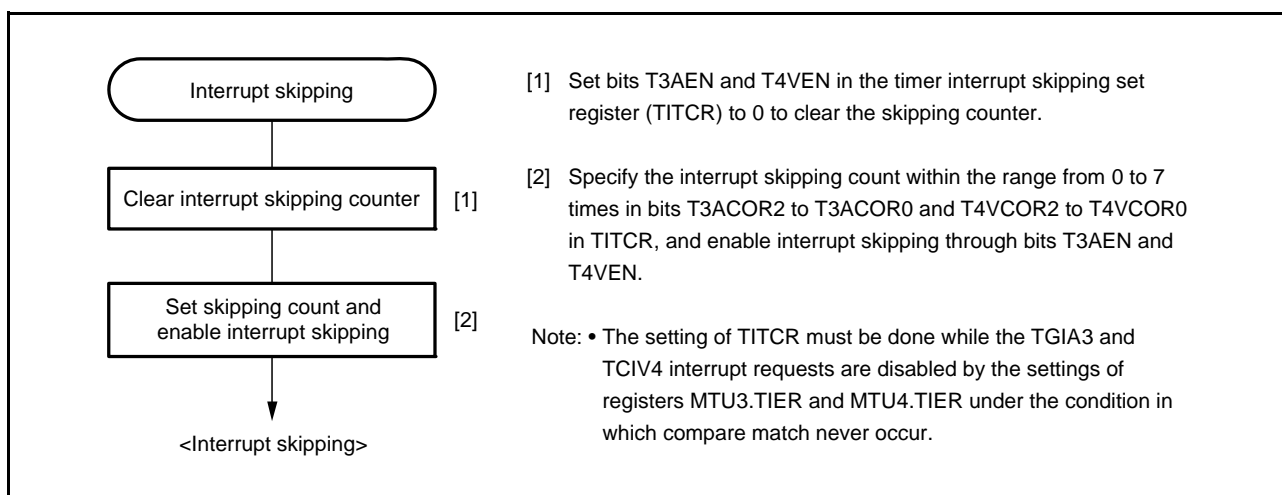


Figure 22.67 Example of Interrupt Skipping Operation Setting Procedure

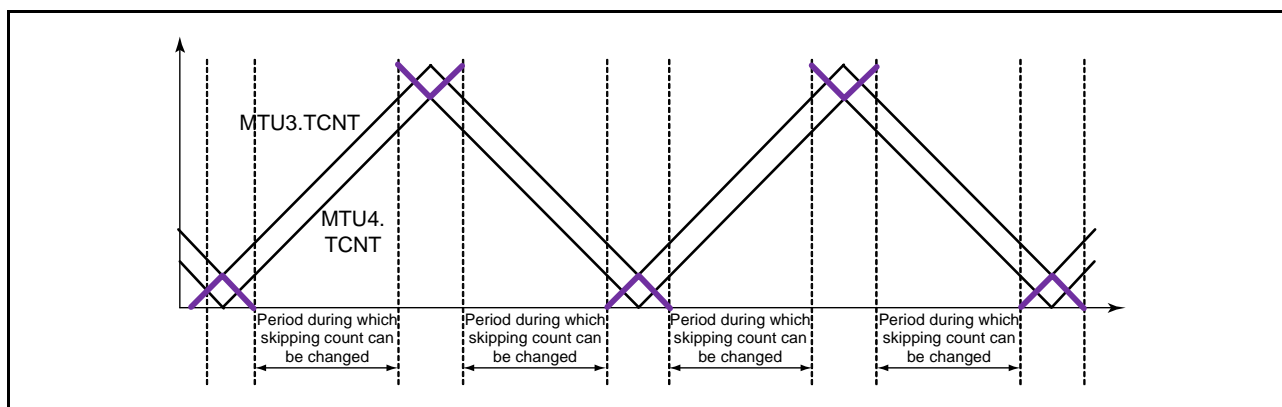
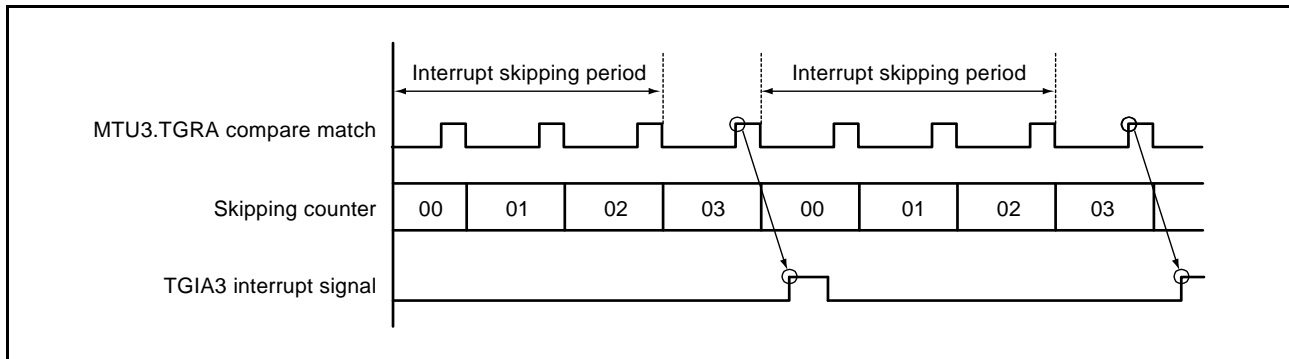


Figure 22.68 Periods during which Interrupt Skipping Count can be Changed

### (b) Example of Interrupt Skipping Operation

Figure 22.69 shows an example of MTU3.TGIA 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 timer interrupt skipping set register (TITCR).



**Figure 22.69 Example of Interrupt Skipping Operation**

### (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 timer buffer transfer set register (TBTER).

Figure 22.70 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 22.71 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 to the temporary register outside the buffer transfer-enabled period.

Note that the buffer transfer-enabled period depends on the T3AEN and T4VEN bit settings in the timer interrupt skipping set register (TITCR). Figure 22.72 shows the relationship between the T3AEN and T4VEN bit settings in TITCR and buffer transfer-enabled period.

Note: • This function must always be used in combination with interrupt skipping.

When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), make sure that buffer transfer is not linked with interrupt skipping (clear the BTE1 bit in the timer buffer transfer set register (TBTER) to 0).

If buffer transfer is linked with interrupt skipping while interrupt skipping is disabled, buffer transfer is never performed.

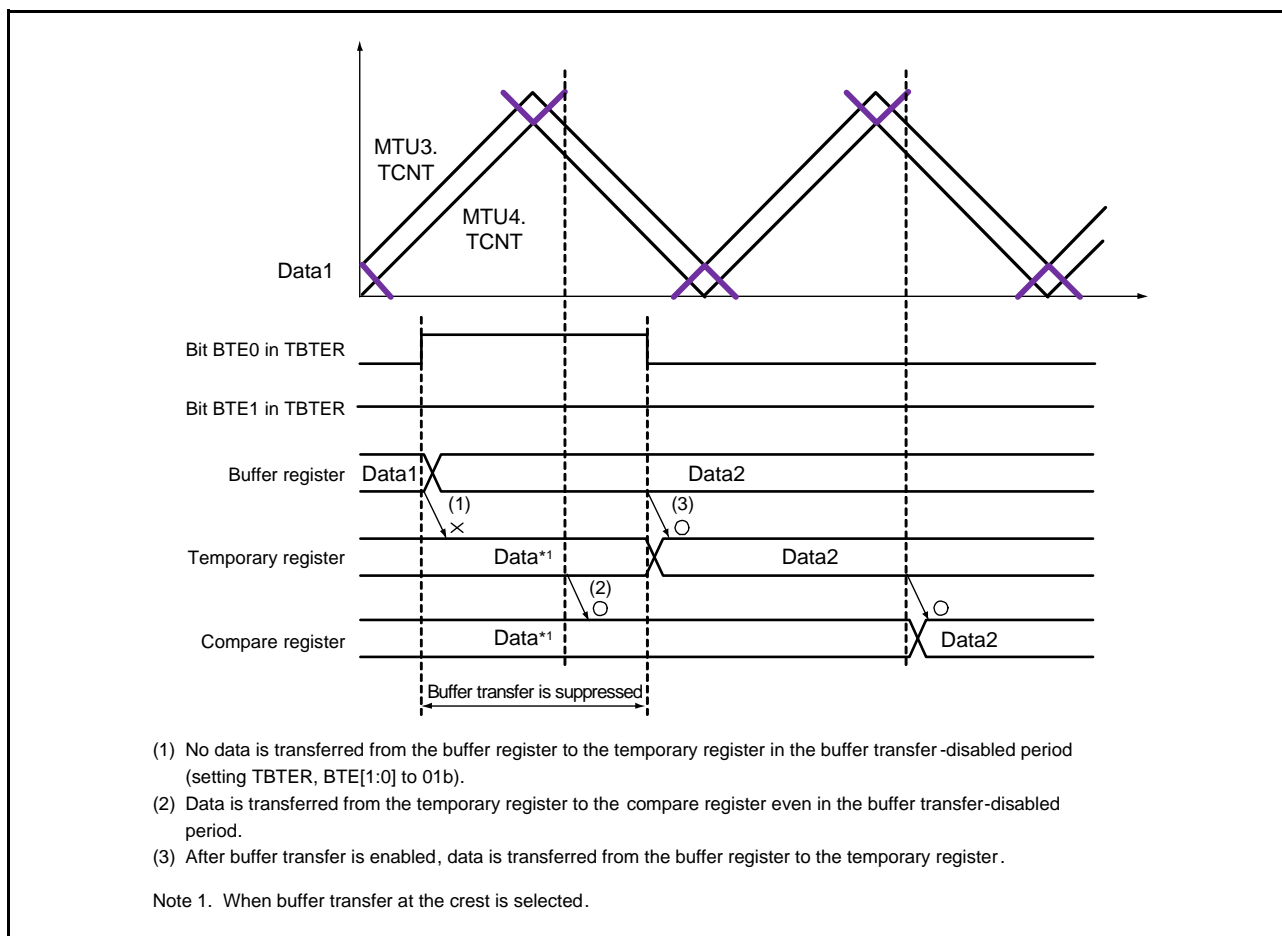
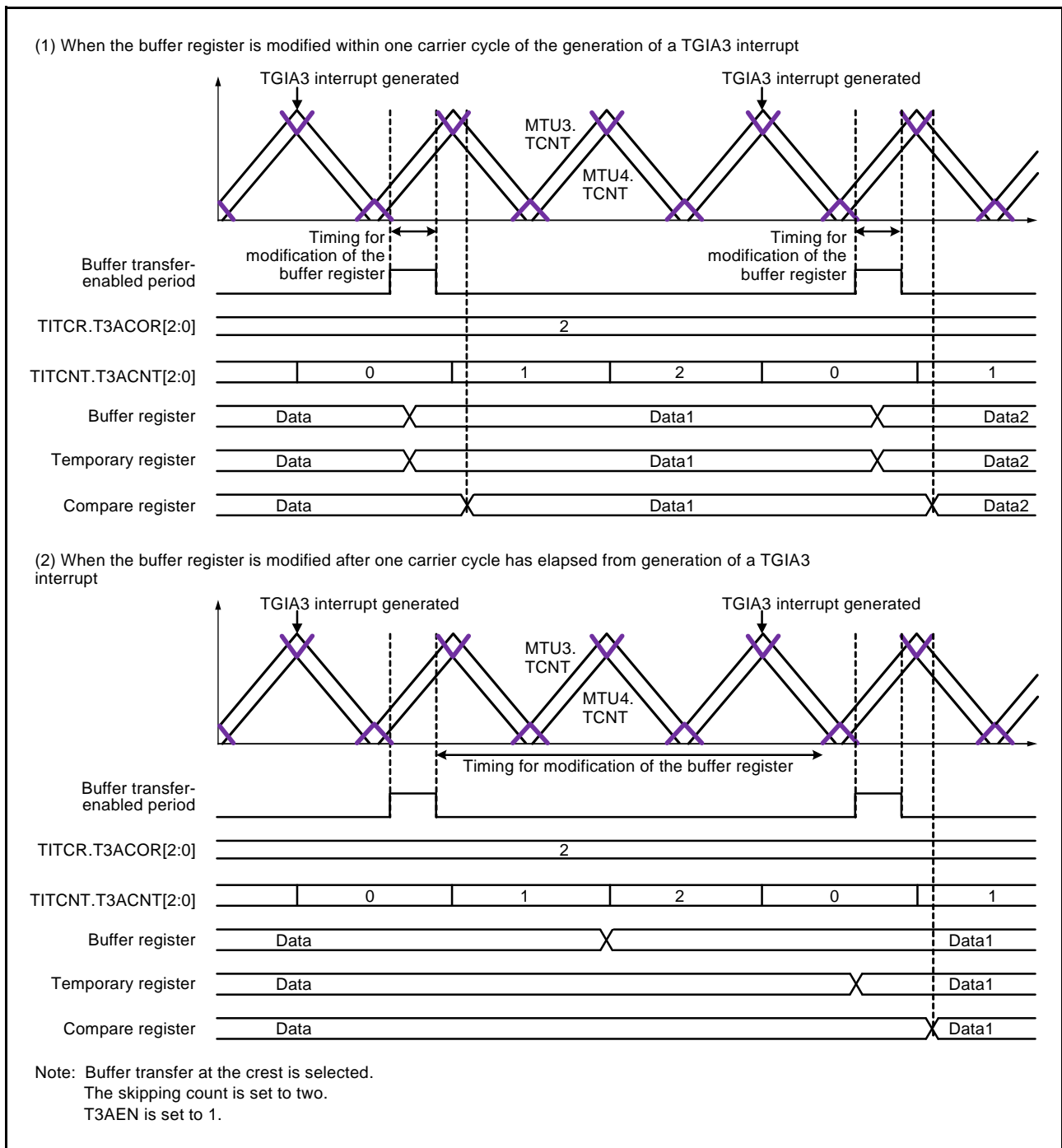
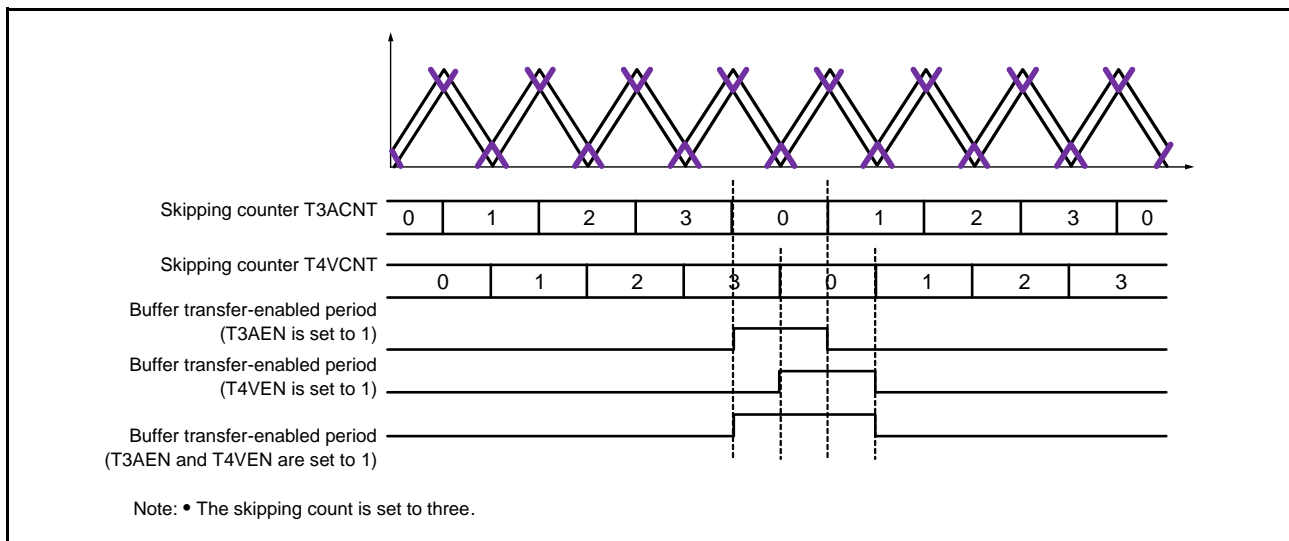


Figure 22.70 Example of Operation when Buffer Transfer is Disabled (BTE[1:0] = 01b)



**Figure 22.71 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE[1:0] = 10b)**



**Figure 22.72 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period**

#### (4) Complementary PWM Mode Output Protection Functions

The MTU2A provides the following protection functions for complementary PWM mode output.

##### (a) Register and Counter Miswrite Prevention Function

With the exception of the buffer registers, which can be modified at any time, access by the CPU can be enabled or disabled for the mode registers, control registers, compare registers, and counters used in complementary PWM mode by means of the RWE bit in the timer read/write enable register (TRWER). The applicable registers are some of the registers in MTU3 and MTU4 shown below:

22 registers in total

MTU3.TCR and MTU4.TCR, MTU3.TMDR and MTU4.TMDR, MTU3.TIORH and MTU4.TIORH, MTU3.TIORL and MTU4.TIORL, MTU3.TIER and MTU4.TIER, MTU3.TCNT and MTU4.TCNT, MTU3.TGRA and MTU4.TGRA, MTU3.TGRB and MTU4.TGRB, TOER, TOCR1, TOCR2, TGCR, TCDR, and TDDR

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 6-phase PWM output pins can be set to the high-impedance state automatically by inputting specified external signals.

See section 23, Port Output Enable 2 (POE2a), for details.

##### (c) Halting of PWM Output when Oscillator is Stopped

Upon detecting that the clock input to RX630 has stopped, the 6-phase PWM output pins are automatically set to the high-impedance state. Note that the pin states are not guaranteed when the clock is restarted.

See section 9.5, Oscillation Stop Detection Function.



### 22.3.9 A/D Converter Start Request Delaying Function

A/D converter start requests can be issued in MTU4 by making settings in the timer A/D converter start request control register (TADCR), timer A/D converter start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB), and timer A/D converter start request cycle set buffer registers (MTU4.TADCOBRA and MTU4.TADCOBRB).

The A/D converter 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 converter start request (TRG4AN or TRG4BN).

A/D converter 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 TADCR.

#### (1) Example of Procedure for Specifying A/D Converter Start Request Delaying Function

Figure 22.73 shows an example of procedure for specifying the A/D converter start request delaying function.

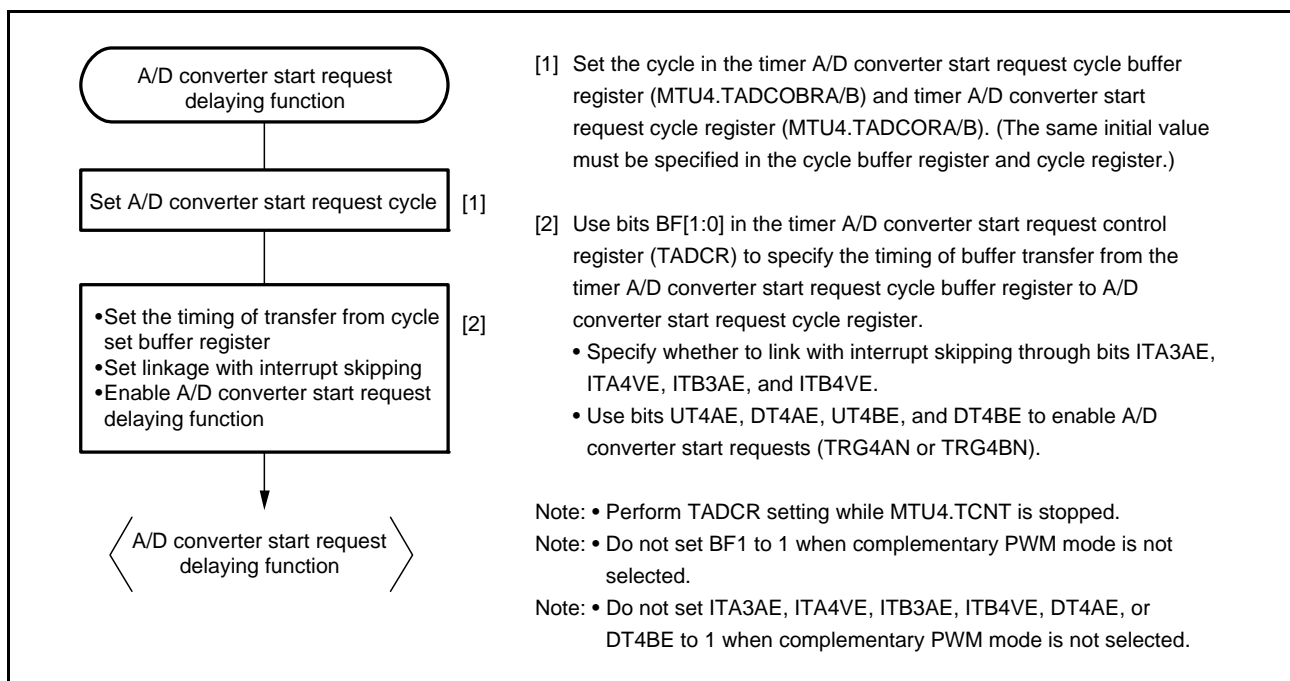
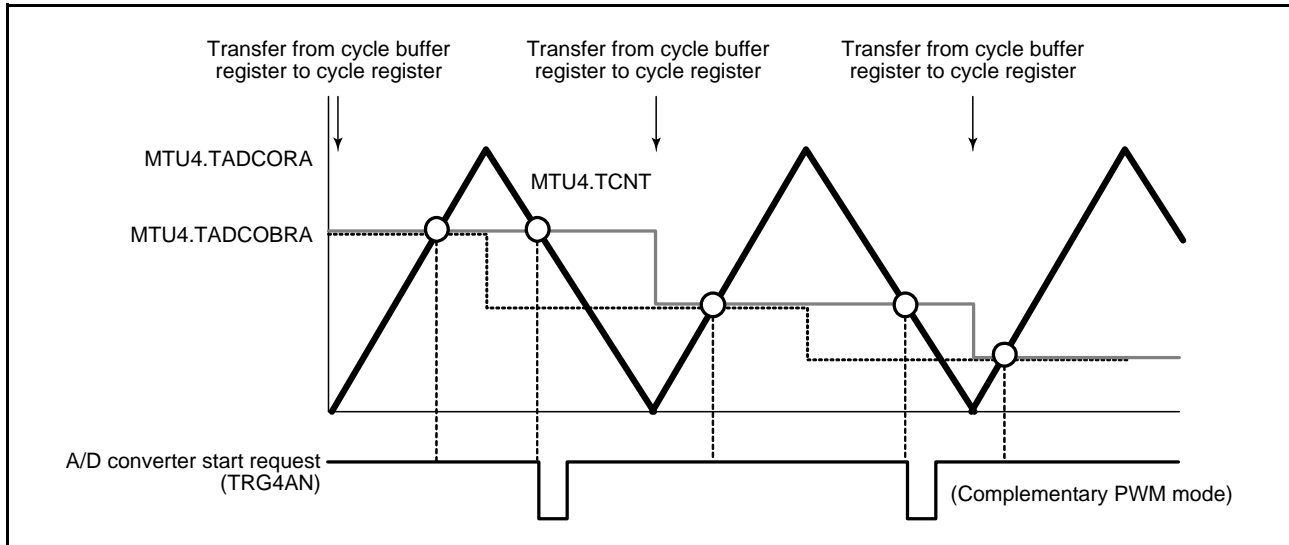


Figure 22.73 Example of Procedure for Specifying A/D Converter Start Request Delaying Function

### (2) Basic Example of A/D Converter Start Request Delaying Function Operation

Figure 22.74 shows a basic example of A/D converter start request signal (TRG4AN) operation when the trough of MTU4.TCNT is specified for the buffer transfer timing and an A/D converter start request signal is output during MTU4.TCNT down-counting.



**Figure 22.74 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation**

### (3) Buffer Transfer

The data in the timer A/D converter start request cycle set registers (MTU4.TADCORA and MTU4.TADCORB) is updated by writing data to the timer A/D converter 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 timer A/D converter start request control register (MTU4.TADCR).

### (4) A/D Converter Start Request Delaying Function Linked with Interrupt Skipping

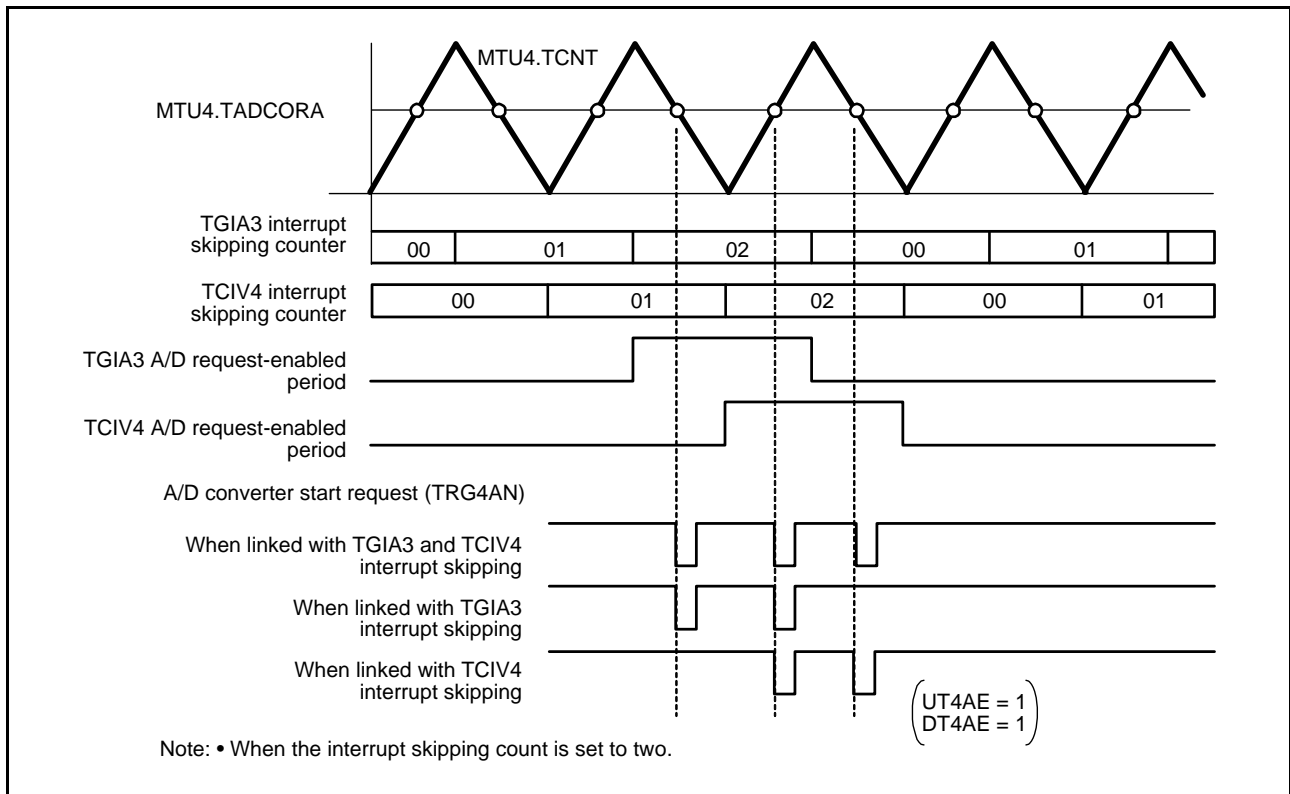
A/D converter start requests (TRG4AN and TRG4BN) can be issued in coordination with interrupt skipping by making settings in the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR). Figure 22.75 shows an example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during MTU4.TCNT up-counting and down-counting and A/D converter start requests are linked with interrupt skipping.

Figure 22.76 shows another example of A/D converter start request signal (TRG4AN) operation when TRG4AN output is enabled during MTU4.TCNT up-counting and A/D converter start requests are linked with interrupt skipping.

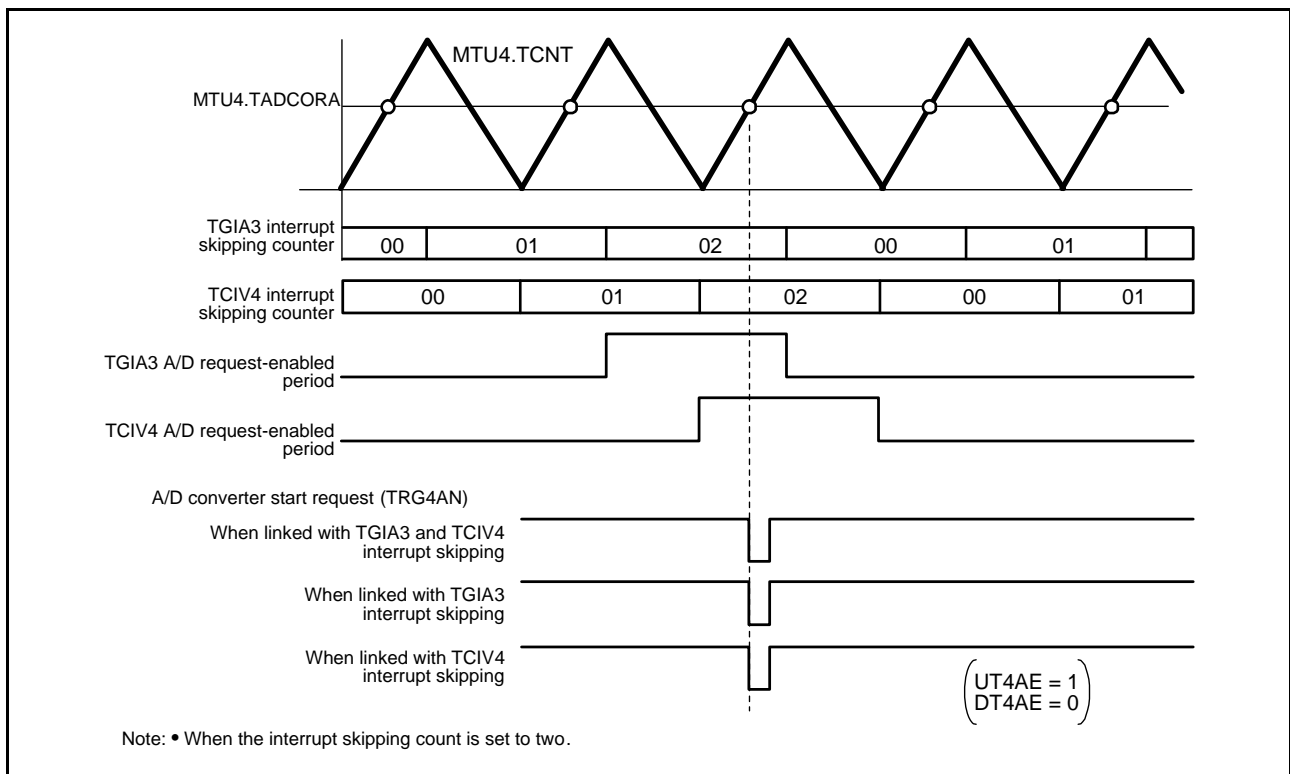
Note: • This function should be used in combination with interrupt skipping.

When interrupt skipping is disabled (the T3AEN and T4VEN bits in the timer interrupt skipping set register (TITCR) are cleared to 0 or the skipping count set bits (T3ACOR and T4VCOR) in TITCR are cleared to 0), make sure that A/D converter start requests are not linked with interrupt skipping (clear the ITA3AE, ITA4VE, ITB3AE, and ITB4VE bits in the timer A/D converter start request control register (TADCR) to 0).

Note that TRG4ABN (TRG4AN or TRG4BN) is output as the A/D converter start request signal in this case.



**Figure 22.75** Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping (when the output of TRG4AN in counting up and down by TCNT is enabled)



**Figure 22.76** Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping (when the output of TRG4AN in counting up by TCNT is enabled)

### 22.3.10 External Pulse Width Measurement

Up to three external pulse widths can be measured in MTU5.

#### (1) Example of External Pulse Width Measurement Setting Procedure

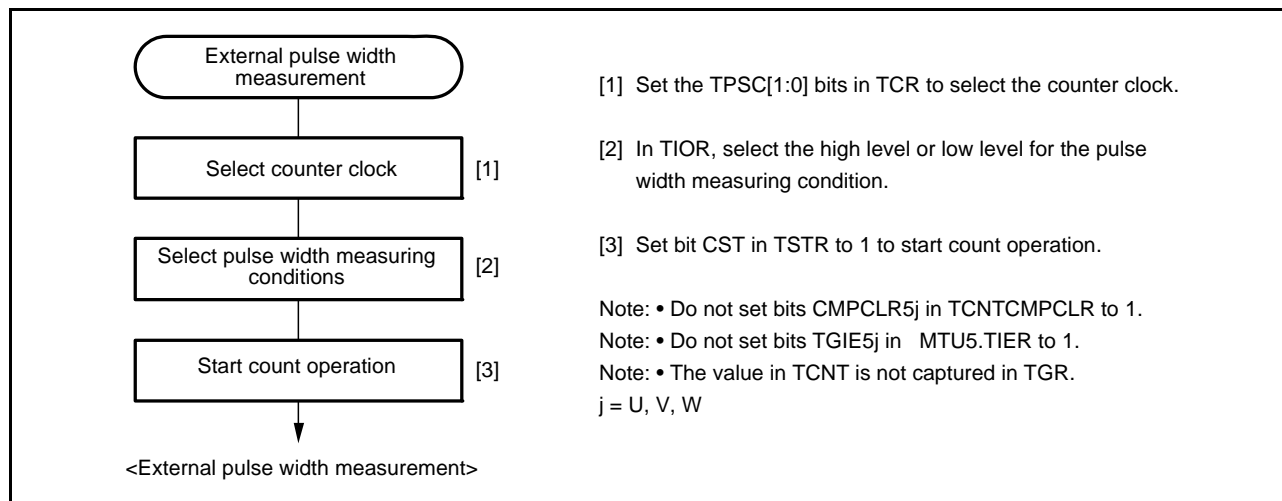


Figure 22.77 Example of External Pulse Width Measurement Setting Procedure

#### (2) Example of External Pulse Width Measurement

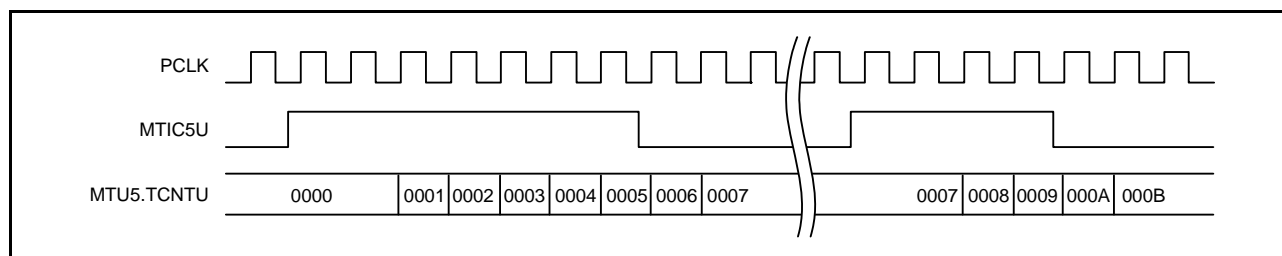


Figure 22.78 Example of External Pulse Width Measurement (Measuring High Pulse Width)

### 22.3.11 Dead Time Compensation

By measuring the delay of the output waveform and reflecting it to duty, the external pulse width measurement function can be used as the dead time compensation function to PWM output waveform while the complementary PWM mode is in operation.

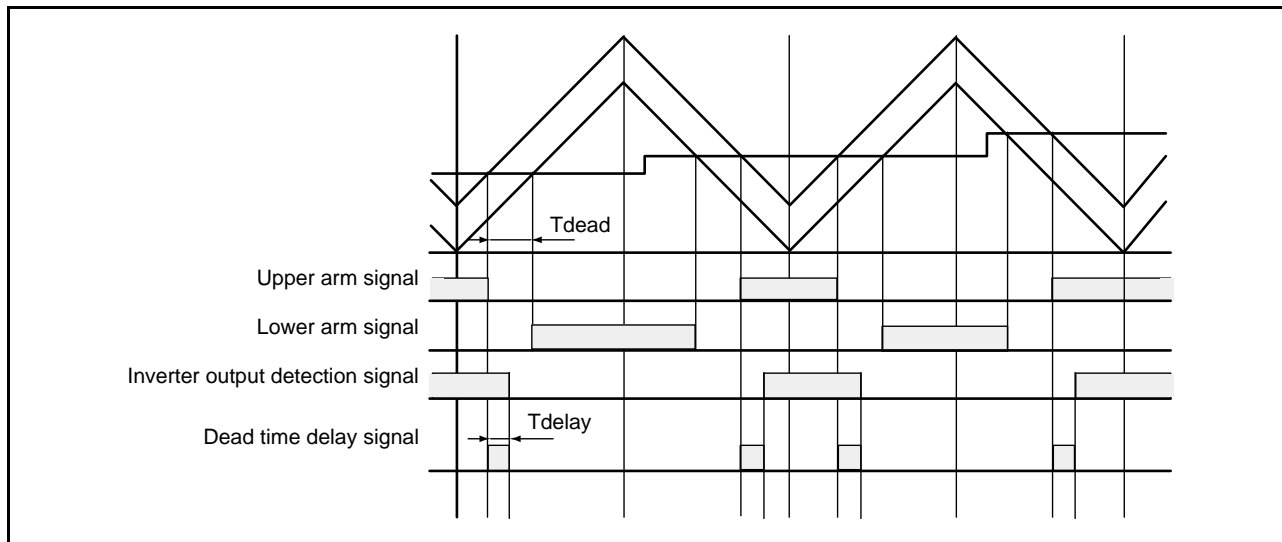


Figure 22.79 Delay in Dead Time in Complementary PWM Mode Operation

(1) Example of Dead Time Compensation Setting Procedure

Figure 22.80 shows an example of dead time compensation setting procedure by using three counters in MTU5.

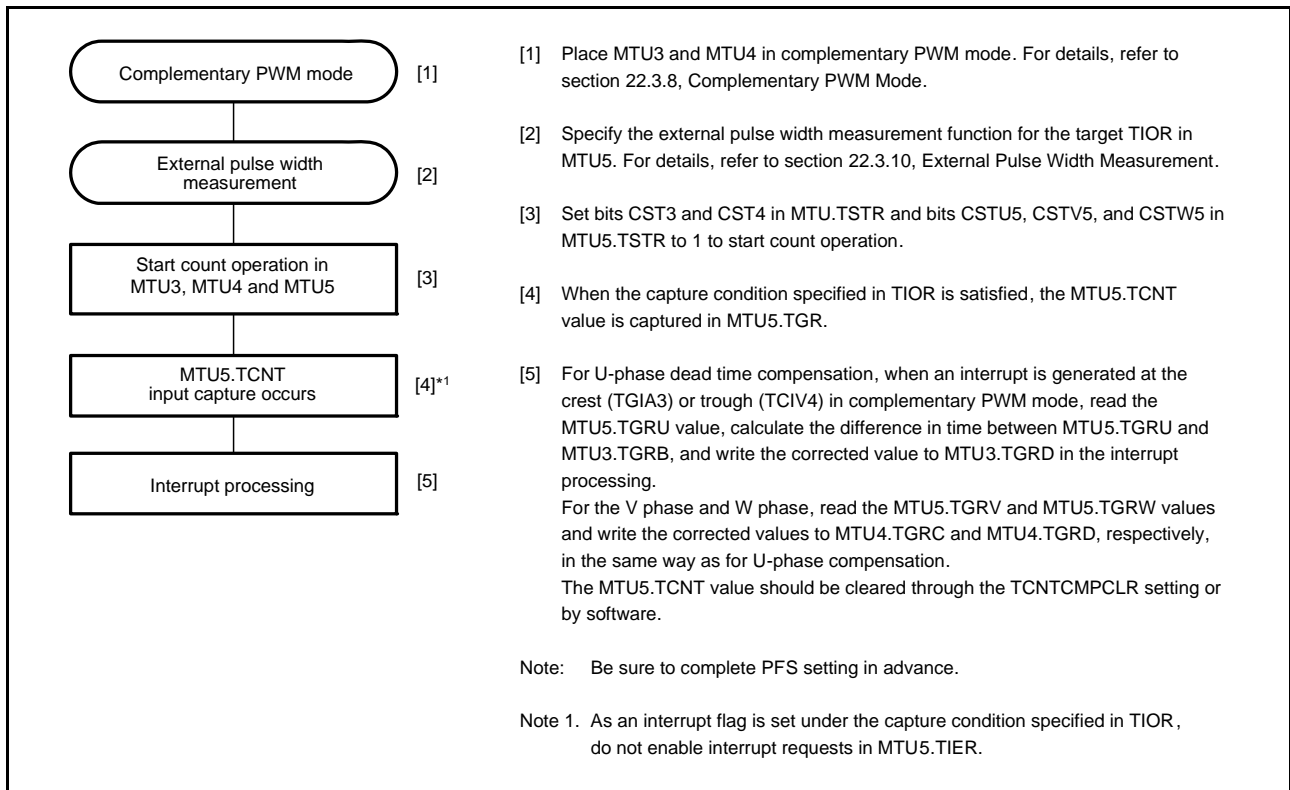


Figure 22.80 Example of Dead Time Compensation Setting Procedure

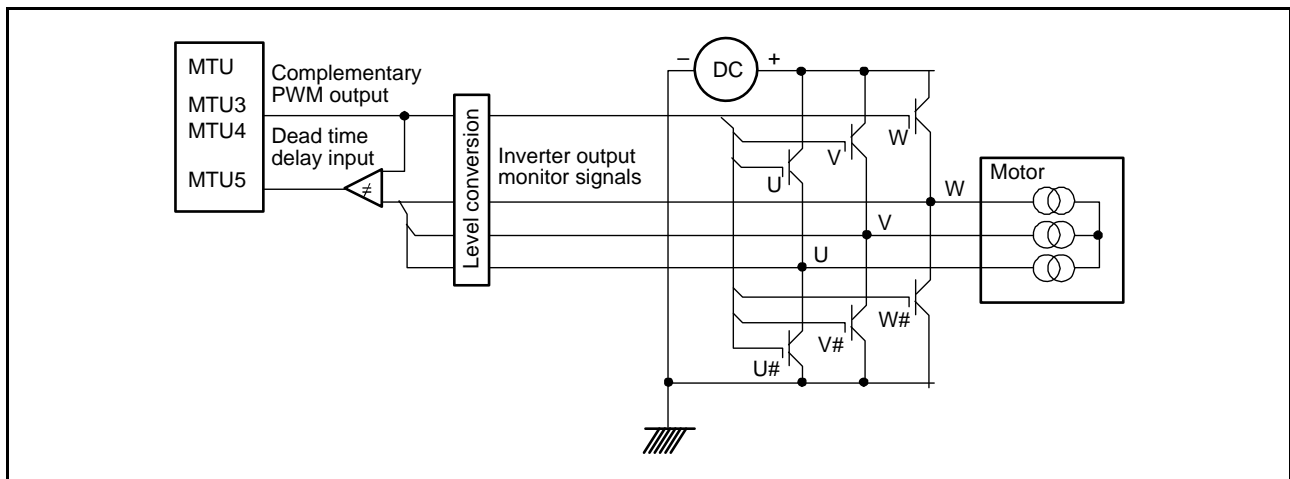


Figure 22.81 Example of Motor Control Circuit Configuration

(2) TCNT Capture at Crest and/or Trough in Complementary PWM Mode Operation

The MTU5.TCNT value is captured in MTU5.TGR at either the crest or trough or at both the crest and trough during complementary PWM mode operation. The timing for capturing in MTU5.TGR can be selected by TIOR.

Figure 22.82 shows operation for the capture of MTU5.TCNT on crests and in troughs in complementary PWM mode.

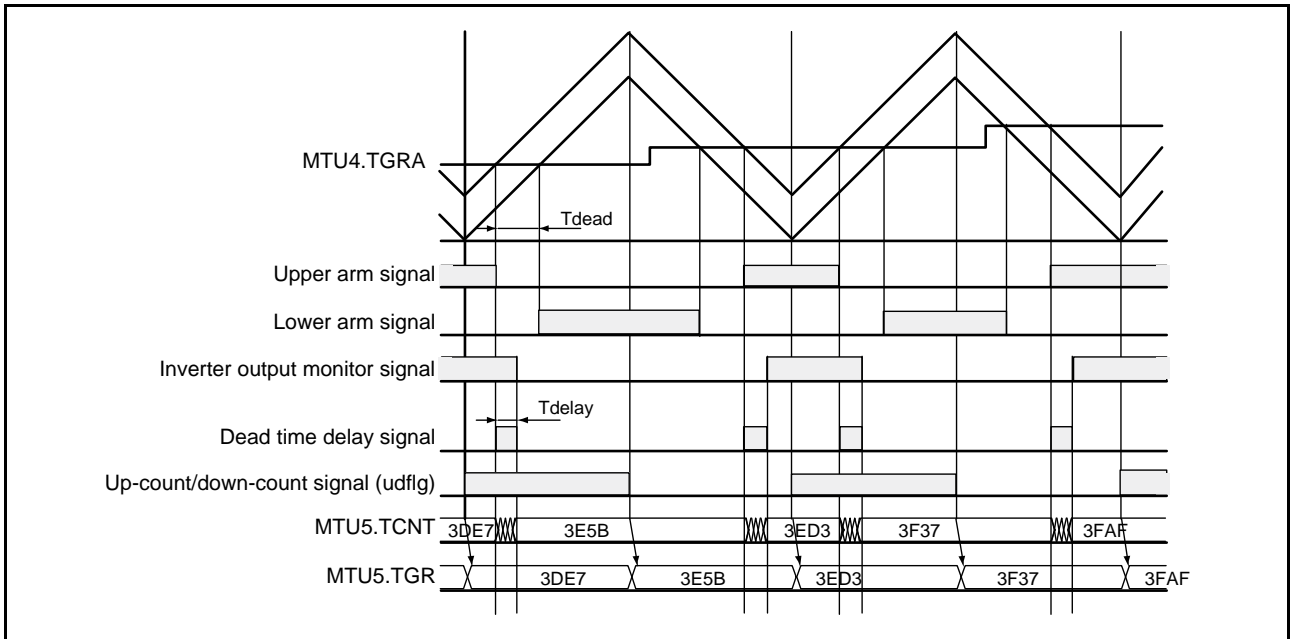


Figure 22.82 MTU5. TCNT Capture at Crest and/or Trough in Complementary PWM Mode Operation

22.3.12 Noise Filter

Each pin for use in input capture and external pulse input to the MTU is equipped with a noise filter. The noise filter samples input signals at the sampling clock and removes the pulses of which length is less than three sampling cycles. The noise filter functionality includes enabling and disabling of the noise filter for each pin and setting of the sampling clock for each channel. Figure 22.83 shows the timing of noise filtering.

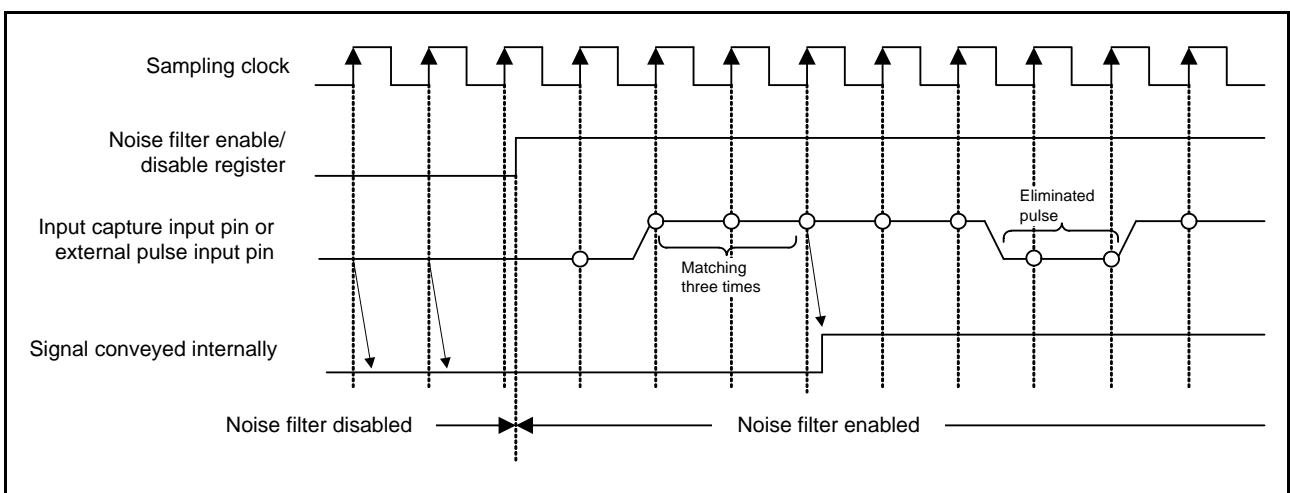


Figure 22.83 Timing of Noise Filtering





### (1) Input Capture/Compare Match Interrupt

An interrupt is requested if the TGIE bit in TIER is set to 1 when a TGR input capture/compare match occurs on a channel. 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

An interrupt is requested if the TCIEV bit in TIER is set to 1 when a TCNT overflow occurs on a channel. The MTU has five overflow interrupts (one for each channel).

### (3) Underflow Interrupt

An interrupt is requested if the TCIEU bit in TIER is set to 1 when a TCNT underflow occurs on a channel. The MTU has two underflow interrupts (one each for MTU1 and MTU2).

## 22.4.2 DTC and DMAC Activation

### (1) DTC Activation

The DTC can be activated by the TGR input capture/compare match interrupt in each channel or the overflow interrupt in MTU4. For details, see section 19, Data Transfer Controller (DTCa).

The MTU provides a total of 20 input capture/compare match interrupts and overflow interrupts that can be used as DTC activation sources: four each for MTU0 and MTU3, two each for MTU1 and MTU2, five for MTU4, and three for MTU5.

### (2) DMAC Activation

The DMAC can be activated by the TGRA input capture/compare match interrupt in each channel. For details, see section 18, DMA Controller (DMACA).

The MTU provides a total of five TGRA input capture/compare match interrupts that can be used as DMAC activation sources: one each for MTU0 to MTU4.

When the DMAC is activated by the MTU, the activation source is cleared when the DMAC requests the internal bus mastership. Therefore, the request for DMAC transfer may be kept pending for a certain period even after the activation source is cleared depending on the internal bus state.

## 22.4.3 A/D Converter Activation

The A/D converter can be activated by one of the following five methods in the MTU. Table 22.58 lists the relationship between interrupt sources and A/D converter start request signals.

### (1) A/D Converter Activation by TGRA Input Capture/Compare Match or at MTU4.TCNT Trough in Complementary PWM Mode

The A/D converter can be activated by the occurrence of a TGRA input capture/compare match in each channel. In addition, if complementary PWM mode operation is performed while the TTGE2 bit in MTU4.TIER is set to 1, the A/D converter can be activated at the trough of MTU4.TCNT count (MTU4.TCNT = 0000h).

A/D converter start request signal TRGAN is issued to the A/D converter under either of the following conditions.

- When a TGRA input capture/compare match occurs on a channel while the TTGE bit in TIER is set to 1
- When the MTU4.TCNT count reaches the trough (MTU4.TCNT = 0000h) during complementary PWM mode operation while the TTGE2 bit in MTU4.TIER is set to 1

When either condition is satisfied, if A/D converter start signal TRGAN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

**(2) A/D Converter Activation by Compare Match between MTU0.TCNT and MTU0.TGRE**

A compare match between MTU0.TCNT and MTU0.TGRE activates the A/D converter.

A/D converter start request signal TRG0EN is issued when a compare match occurs between MTU0.TCNT and MTU0.TGRE. If A/D converter start signal TRG0EN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

**(3) A/D Converter Activation by Compare Match between MTU0.TCNT and MTU0.TGRF**

A compare match between MTU0.TCNT and MTU0.TGRF activates the A/D converter.

A/D converter start request signal TRG0FN is issued when a compare match occurs between MTU0.TCNT and MTU0.TGRF. If A/D converter start signal TRG0FN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

**(4) A/D Converter Activation by Input Capture or Compare Match with MTU0.TGRA or MTU0.TGRB**

The A/D converter can be activated when an input capture or compare match occurs between MTU0.TCNT and MTU0.TGRA or MTU0.TGRB.

When an input capture or compare match occurs between MTU0.TCNT and MTU0.TGRA or MTU0.TGRB. A/D converter start request signal TRG0AN or TRG0BN is issued. If A/D converter start signal TRG0AN or TRG0BN from the MTU is selected as the trigger in the A/D converter, A/D conversion will start.

**(5) A/D Converter Activation by A/D Converter Start Request Delaying Function**

The A/D converter can be activated by generating A/D converter start request signal TRG4AN or TRG4BN when the MTU4.TCNT count matches the TADCORA or TADCORB value if the UT4AE, DT4AE, UT4BE, or DT4BE bit in the A/D converter start request control register (TADCR) is set to 1. For details, refer to section 22.3.9, A/D Converter Start Request Delaying Function.

A/D conversion will start if A/D converter start signal TRG4ABN from the MTU is selected as the trigger in the A/D converter when TRG4AN or TRG4BN is generated.

**Table 22.58 Interrupt Sources and A/D Converter Start Request Signals**

Target Registers	A/D Start Request Source	A/D Converter Start Request Signal
MTU0.TGRA and MTU0.TCNT	Input capture/compare match	TRGAN
MTU1.TGRA and MTU1.TCNT		
MTU2.TGRA and MTU2.TCNT		
MTU3.TGRA and MTU3.TCNT		
MTU4.TGRA and MTU4.TCNT		
MTU4.TCNT	MTU4.TCNT trough in complementary PWM mode	
MTU0.TGRA and MTU0.TCNT	Input capture/compare match	TRG0AN
MTU0.TGRB and MTU0.TCNT		TRG0BN
MTU0.TGRE and MTU0.TCNT	Compare match	TRG0EN
MTU0.TGRF and MTU0.TCNT		TRG0FN
TADCORA and MTU4.TCNT or TADCORB and MTU4.TCNT		TRG4ABN

## 22.5 Operation Timing

### 22.5.1 Input/Output Timing

#### (1) TCNT Count Timing

Figure 22.84 and Figure 22.85 show the TCNT count timing for TGI interrupt in internal clock operation, Figure 22.86 shows the TCNT count timing in external clock operation (normal mode), and Figure 22.87 shows the TCNT count timing in external clock operation (phase counting mode).

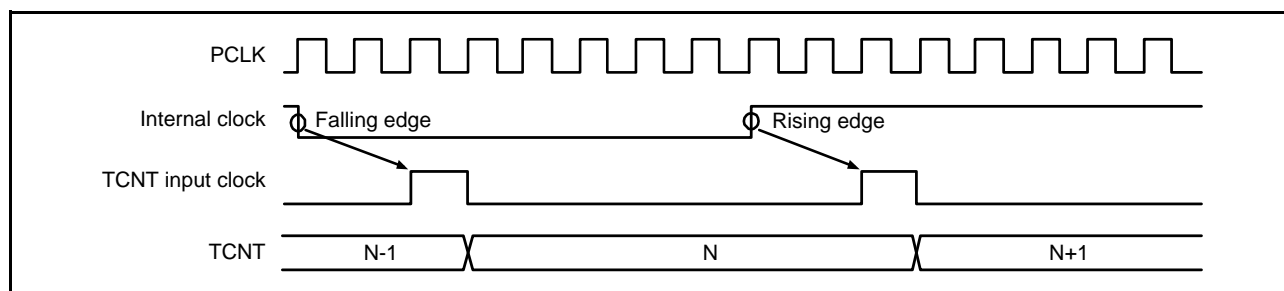


Figure 22.84 Count Timing in Internal Clock Operation (MTU0 to MTU4)

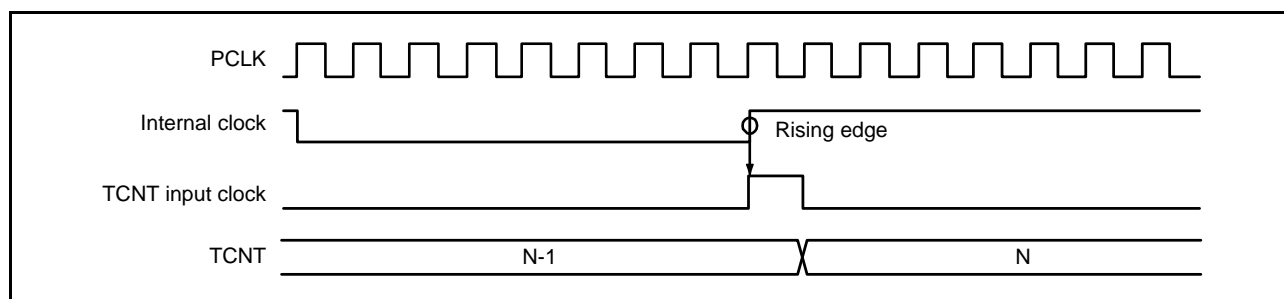


Figure 22.85 Count Timing in Internal Clock Operation (MTU5)

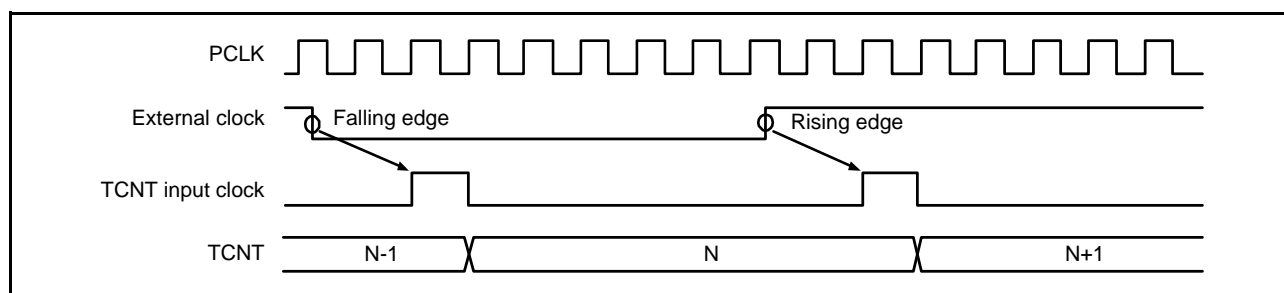


Figure 22.86 Count Timing in External Clock Operation (MTU0 to MTU 4)

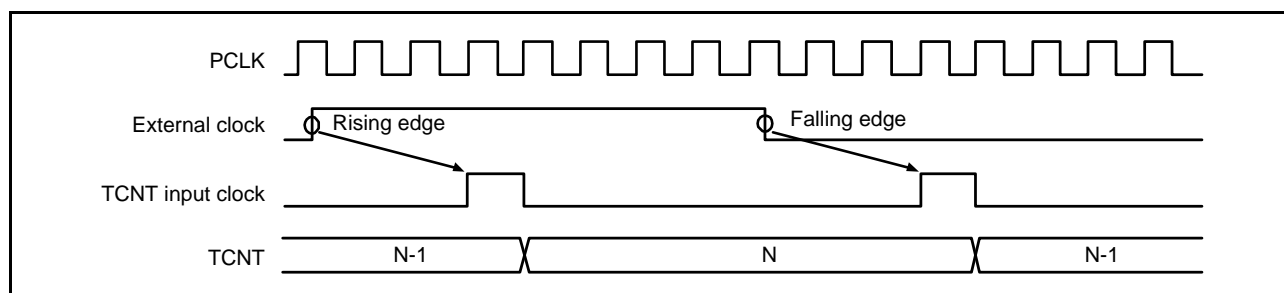


Figure 22.87 Count Timing in External Clock Operation (Phase Counting Mode)

(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 is updated by TCNT). When a compare match signal is generated, the value set in TIOR is output to the output compare output pin (MTIOC pin). After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 22.88 shows the output compare output timing (normal mode or PWM mode) and Figure 22.89 shows the output compare output timing (complementary PWM mode or reset-synchronized PWM mode).

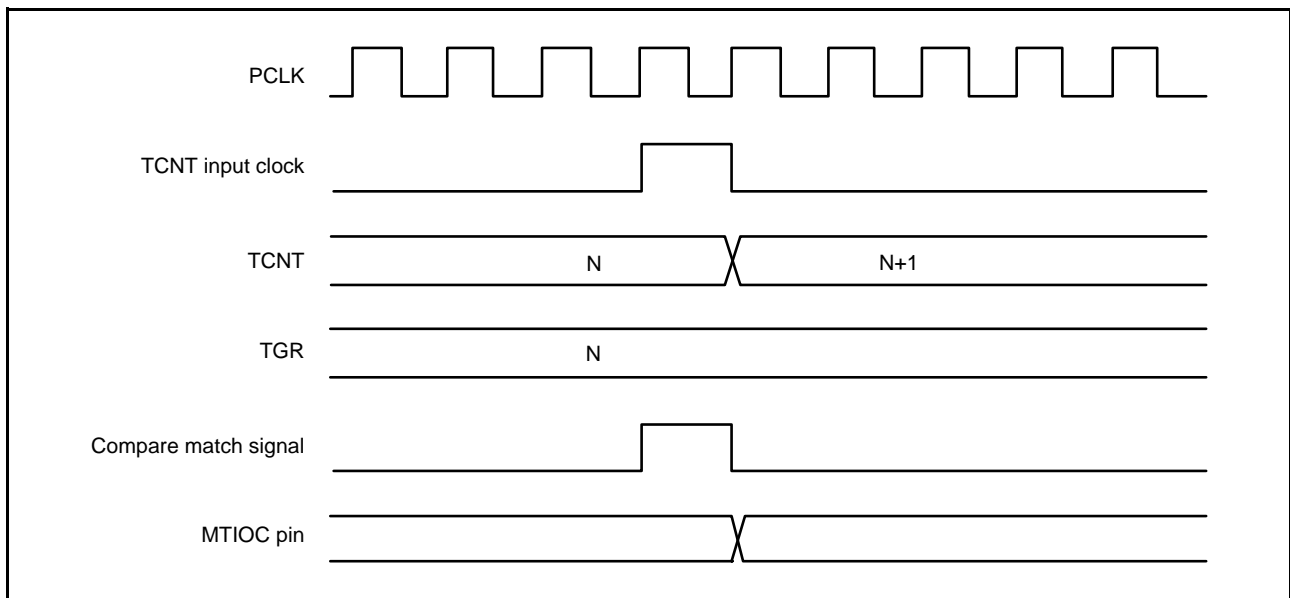


Figure 22.88 Output Compare Output Timing (Normal Mode or PWM Mode)

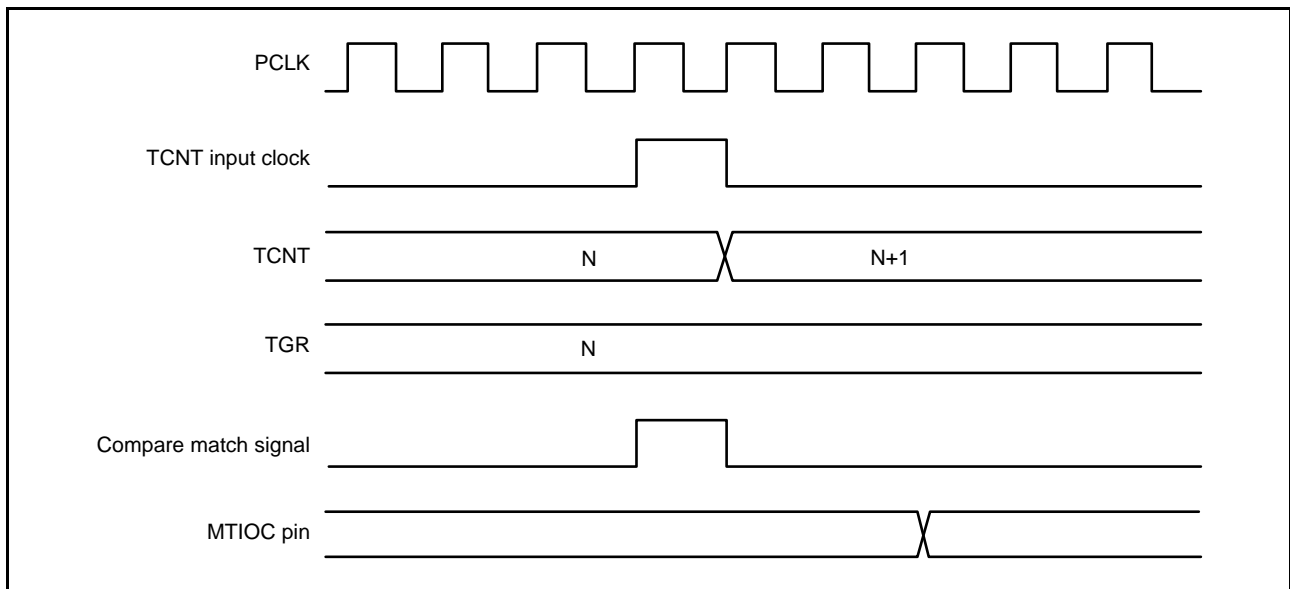


Figure 22.89 Output Compare Output Timing (Complementary PWM Mode or Reset-Synchronized PWM Mode)

(3) Input Capture Signal Timing

Figure 22.90 shows the input capture signal timing.

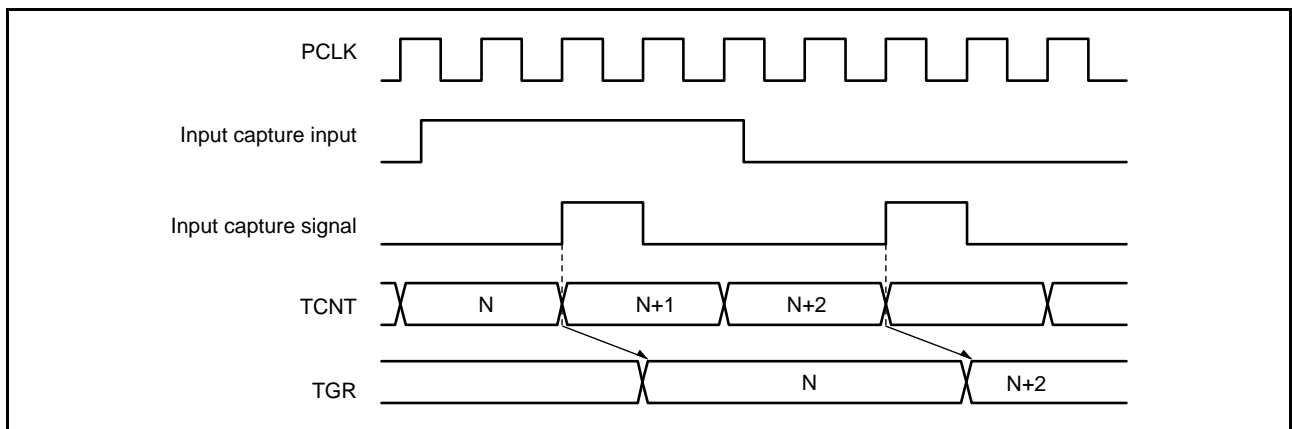


Figure 22.90 Input Capture Input Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 22.91 and Figure 22.92 show the timing when counter clearing on compare match is specified, and Figure 22.93 shows the timing when counter clearing on input capture is specified.

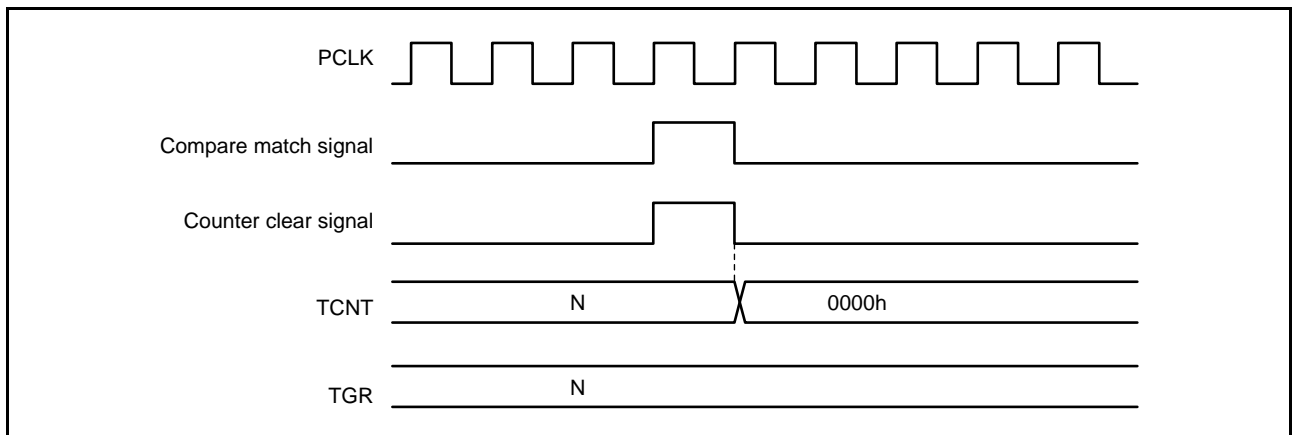


Figure 22.91 Counter Clear Timing (Compare Match) (MTU0 to MTU4)

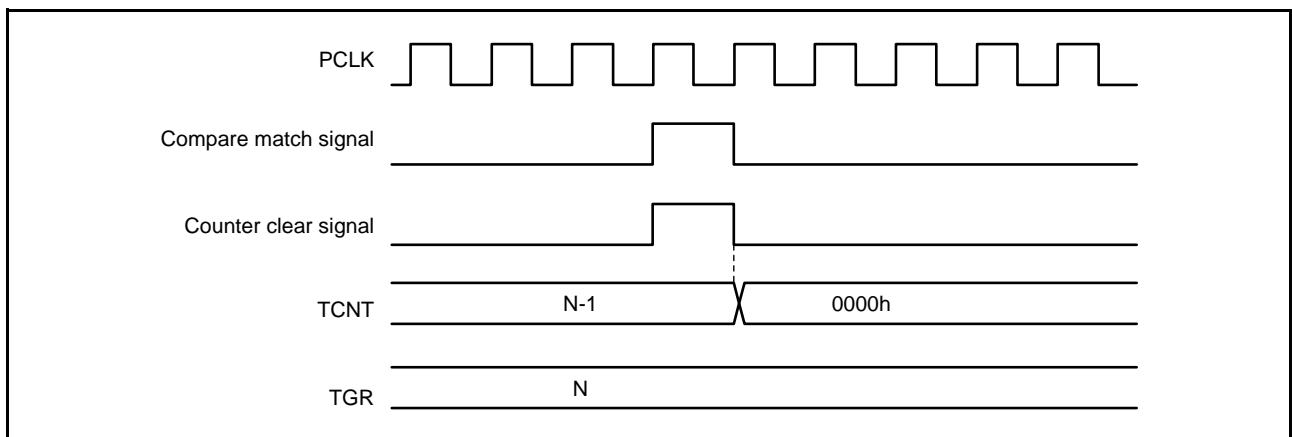


Figure 22.92 Counter Clear Timing (Compare Match) (MTU5)

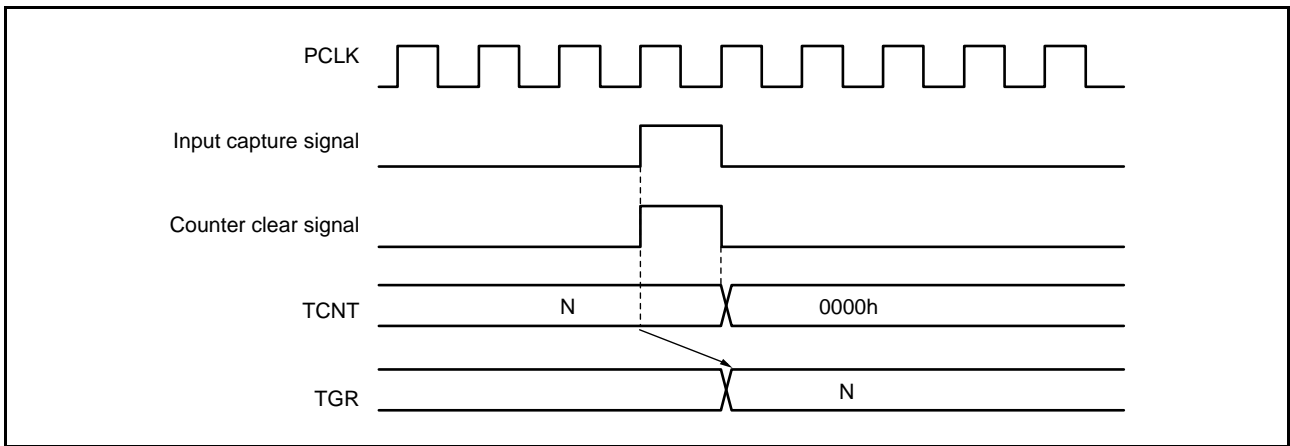


Figure 22.93 Counter Clear Timing (Input Capture) (MTU0 to MTU5)

(5) Buffer Operation Timing

Figure 22.94 to Figure 22.96 show the timing in buffer operation.

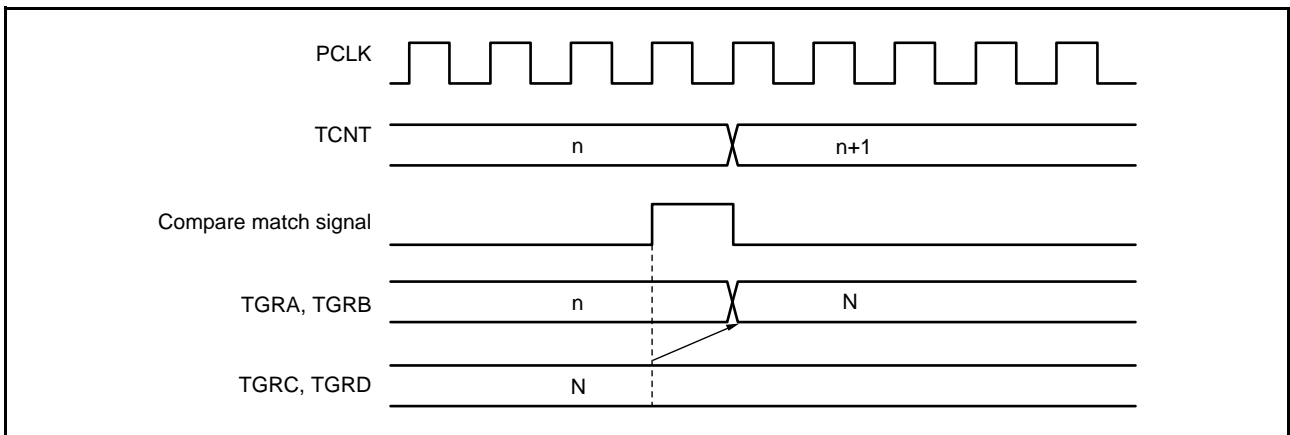


Figure 22.94 Buffer Operation Timing (Compare Match)

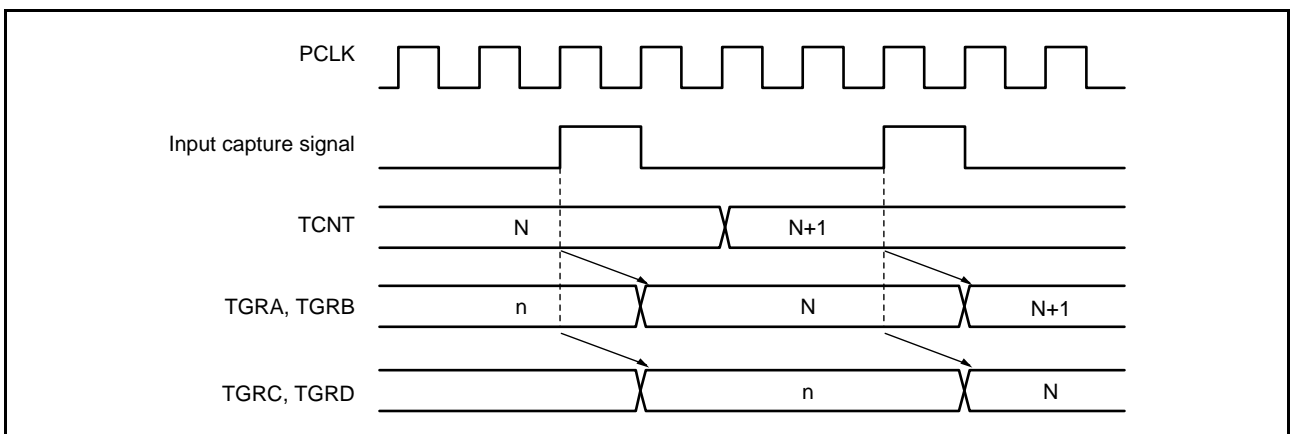


Figure 22.95 Buffer Operation Timing (Input Capture)

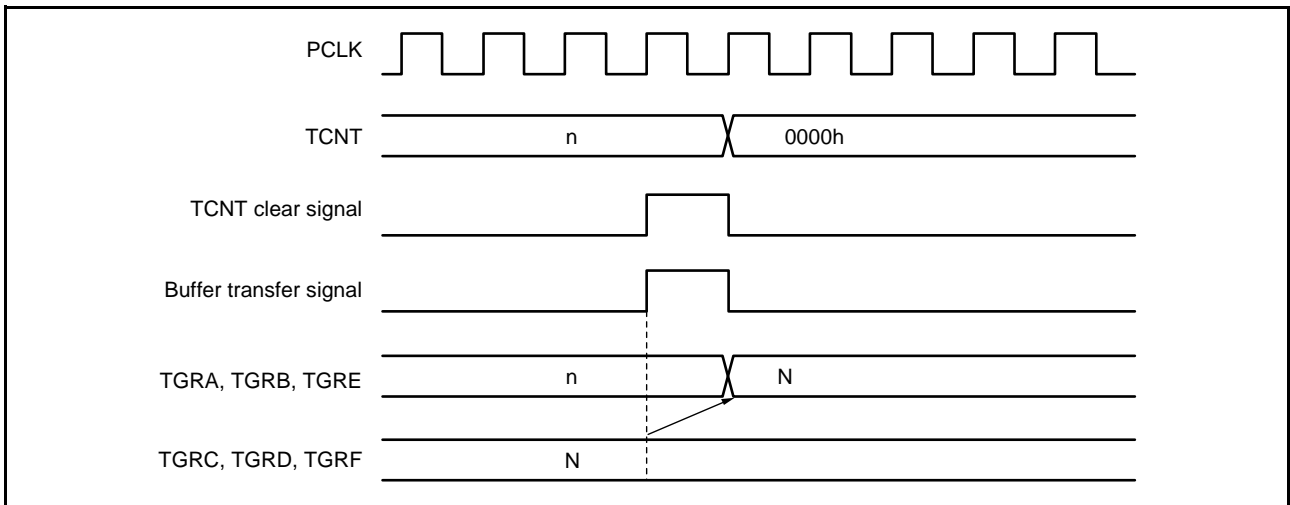


Figure 22.96 Buffer Operation Timing (when TCNT Cleared)

(6) Buffer Transfer Timing (Complementary PWM Mode)

Figure 22.97 to Figure 22.99 show the buffer transfer timing in complementary PWM mode.

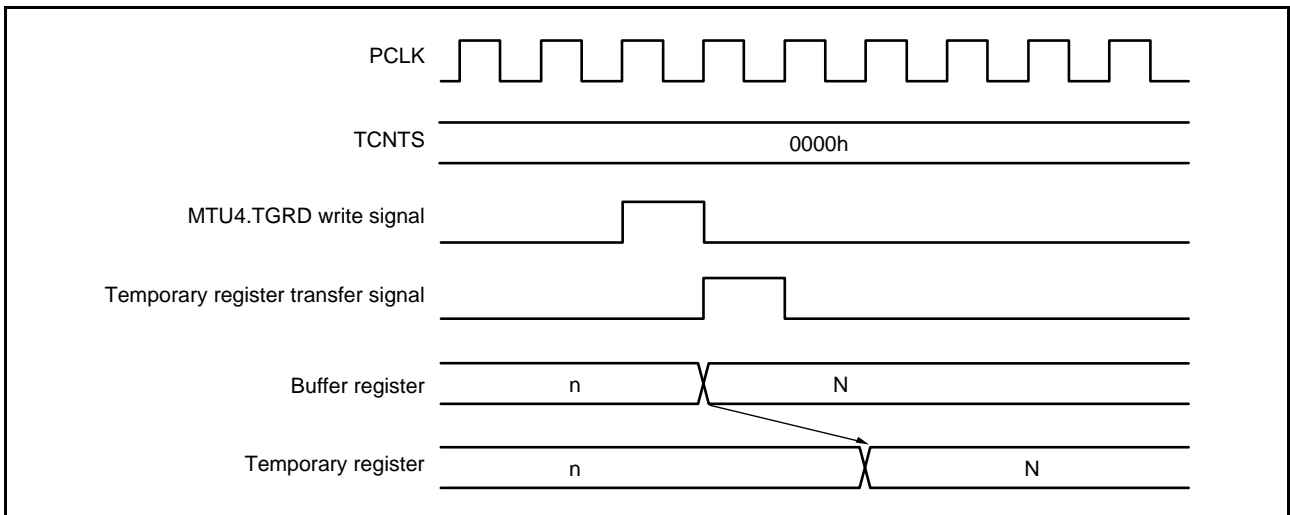


Figure 22.97 Transfer Timing from Buffer Register to Temporary Register (TCNTS Stop)

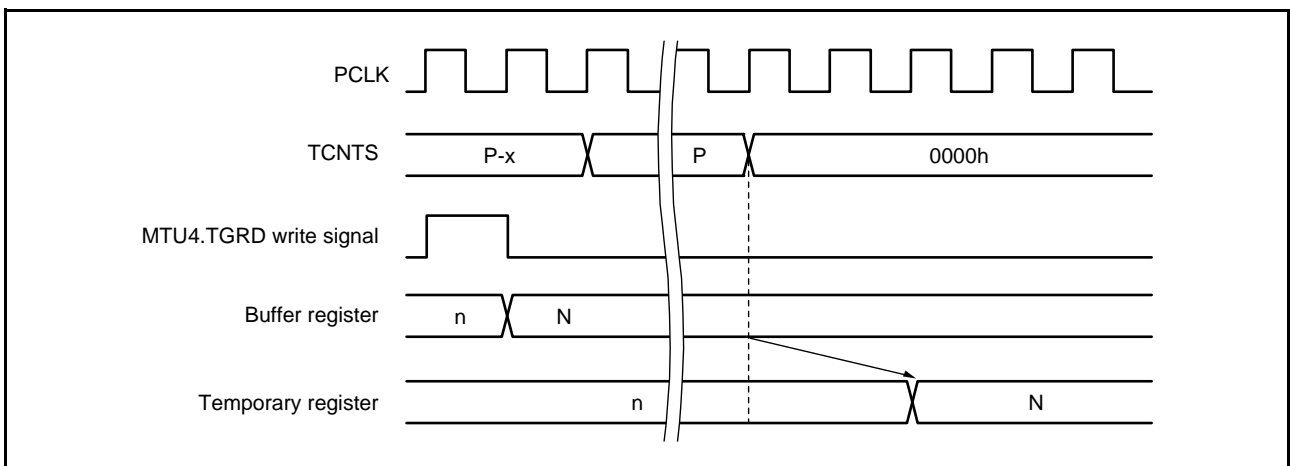


Figure 22.98 Transfer Timing from Buffer Register to Temporary Register (TCNTS Operating)

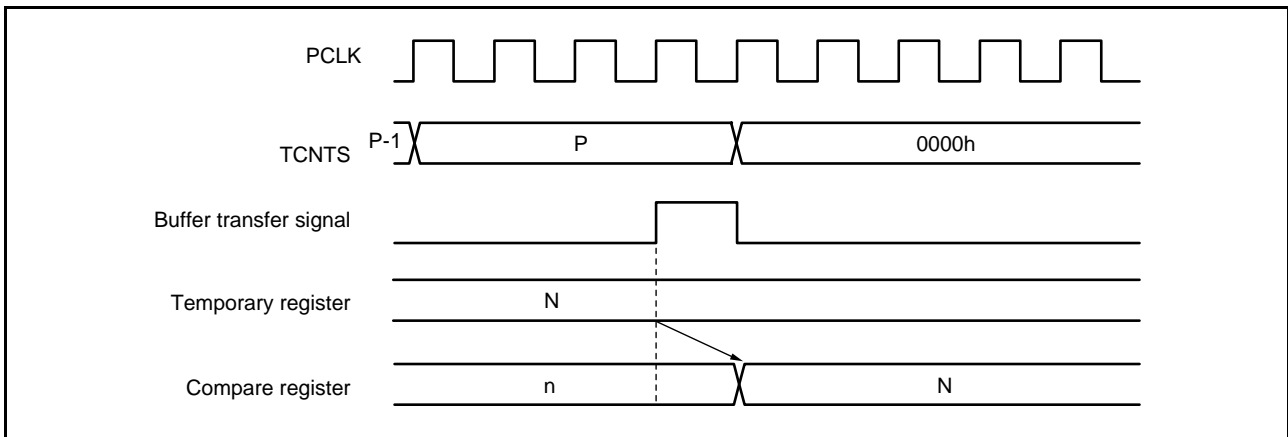


Figure 22.99 Transfer Timing from Temporary Register to Compare Register

### 22.5.2 Interrupt Signal Timing

#### (1) Timing for TGI Interrupt by Compare Match

Figure 22.100 and Figure 22.101 show the TGI interrupt request signal timing on compare match.

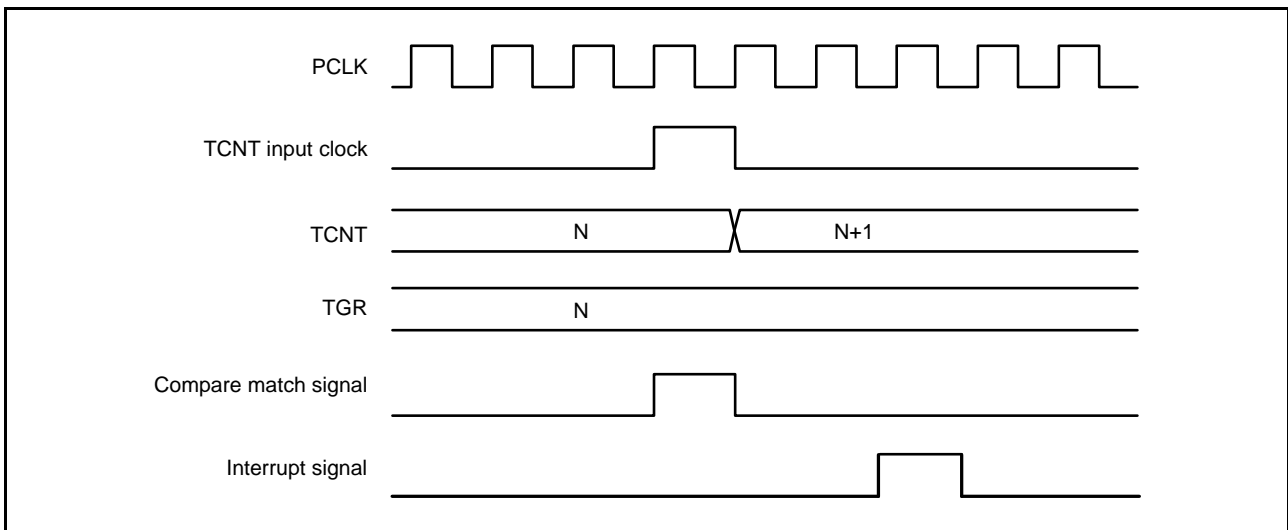


Figure 22.100 TGI Interrupt Timing (Compare Match) (MTU0 to MTU4)



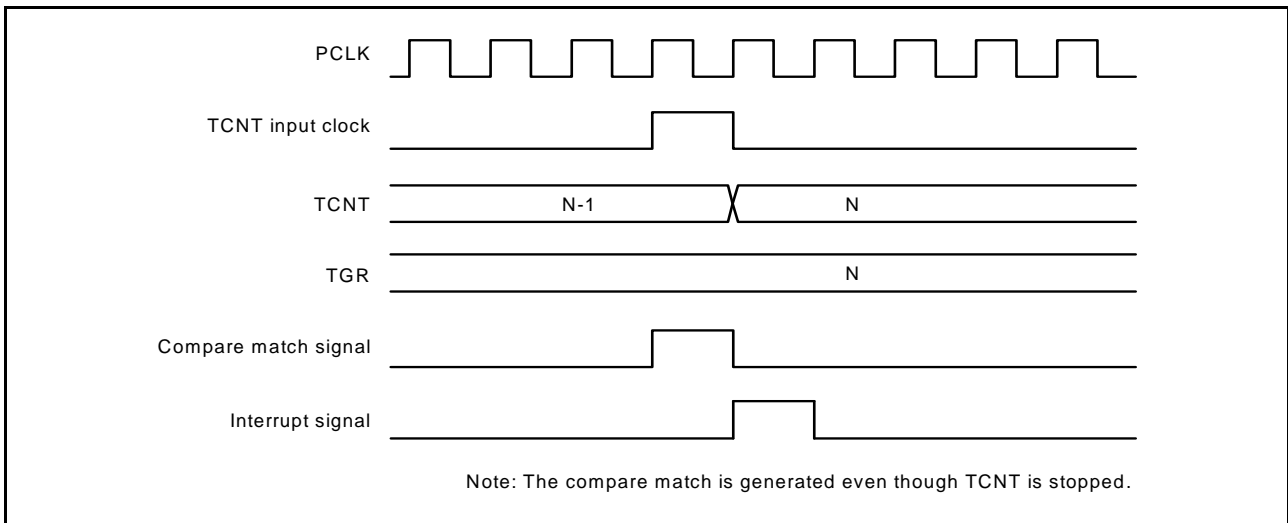


Figure 22.101 TGI Interrupt Timing (Compare Match) (MTU5)

(2) Timing for TGI Interrupt by Input Capture

Figure 22.102 and Figure 22.103 show TGI interrupt request signal timing on input capture.

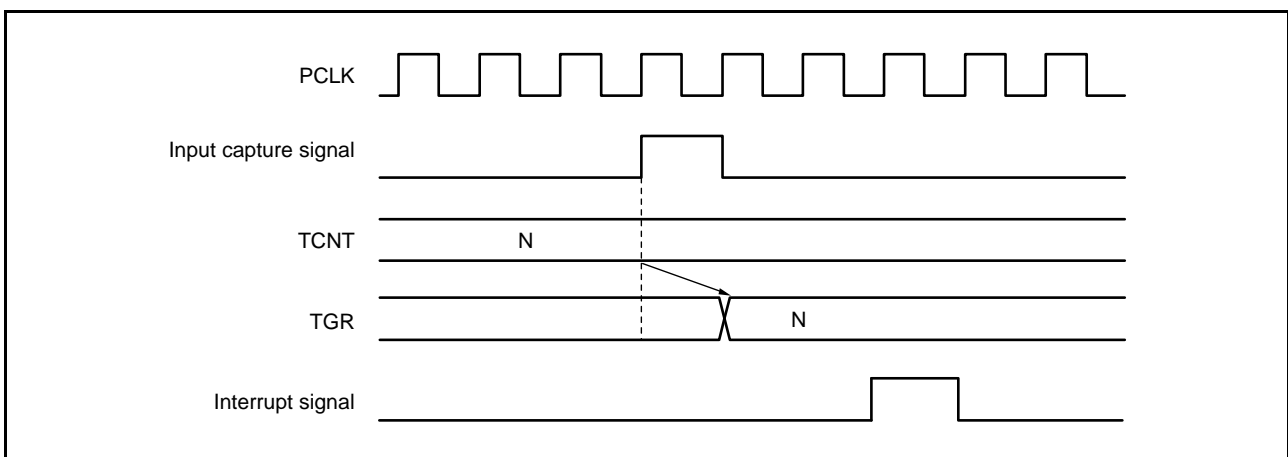


Figure 22.102 TGI Interrupt Timing (Input Capture) (MTU0 to MTU4)

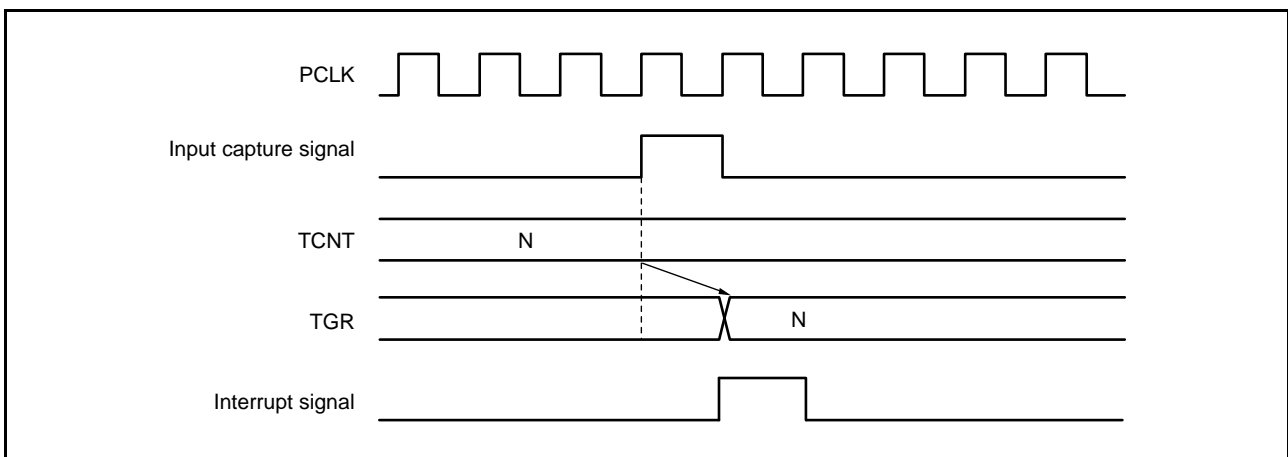


Figure 22.103 TGI Interrupt Timing (Input Capture) (MTU5)

(3) TCIV and TCIU Interrupt Timing

Figure 22.104 shows the TCIV interrupt request signal timing on overflow.

Figure 22.105 shows the TCIU interrupt request signal timing on underflow.

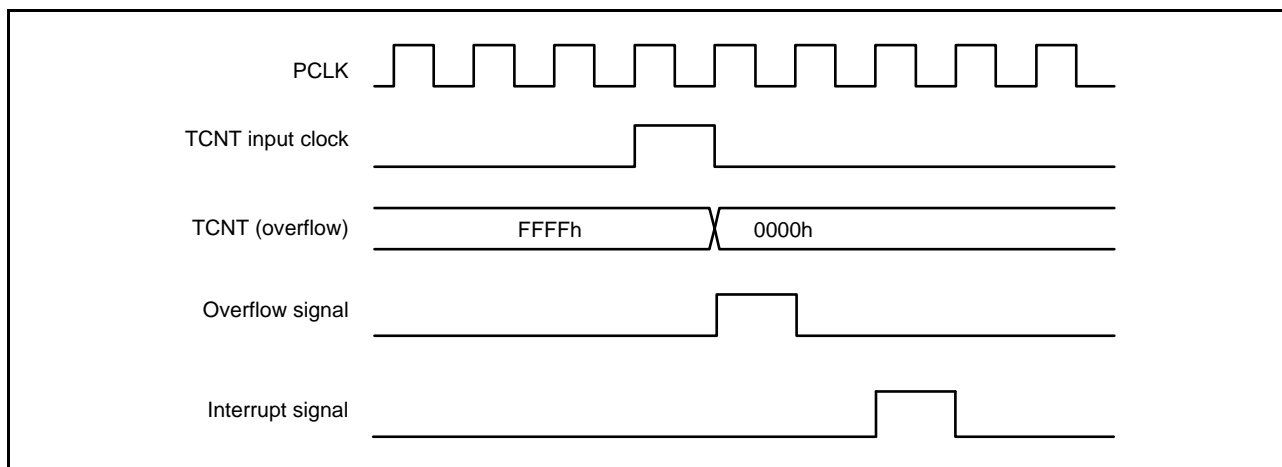


Figure 22.104 TCIV Interrupt Timing

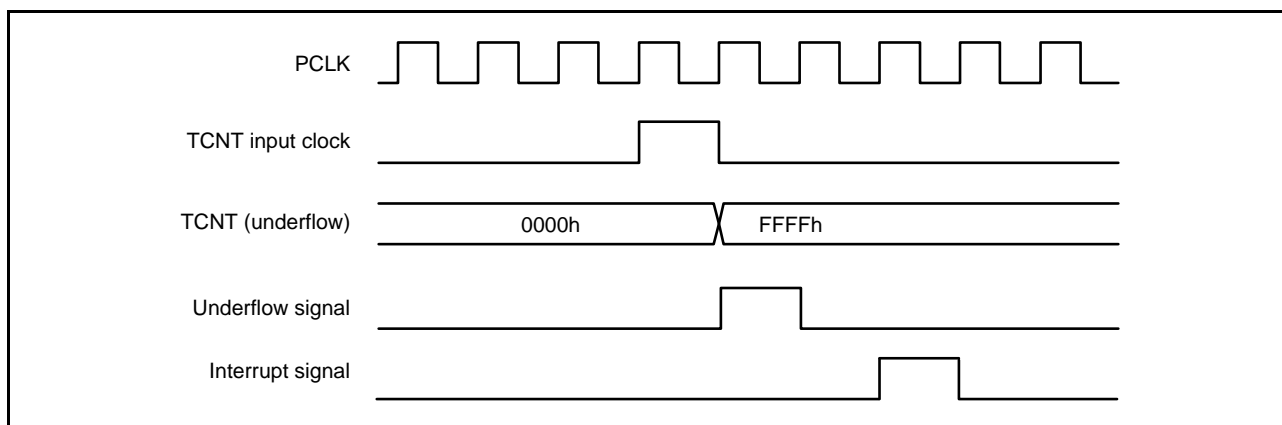


Figure 22.105 TCIU Interrupt Timing

## 22.6 Usage Notes

### 22.6.1 Module Clock Stop Mode 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 clearing the module stop state. For details, refer to section 11, Low Power Consumption.

### 22.6.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 PCLK for single-edge detection, and at least 2.5 PCLK 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 1.5 PCLK, and the pulse width must be at least 2.5 PCLK. Figure 22.106 shows the input clock conditions in phase counting mode.

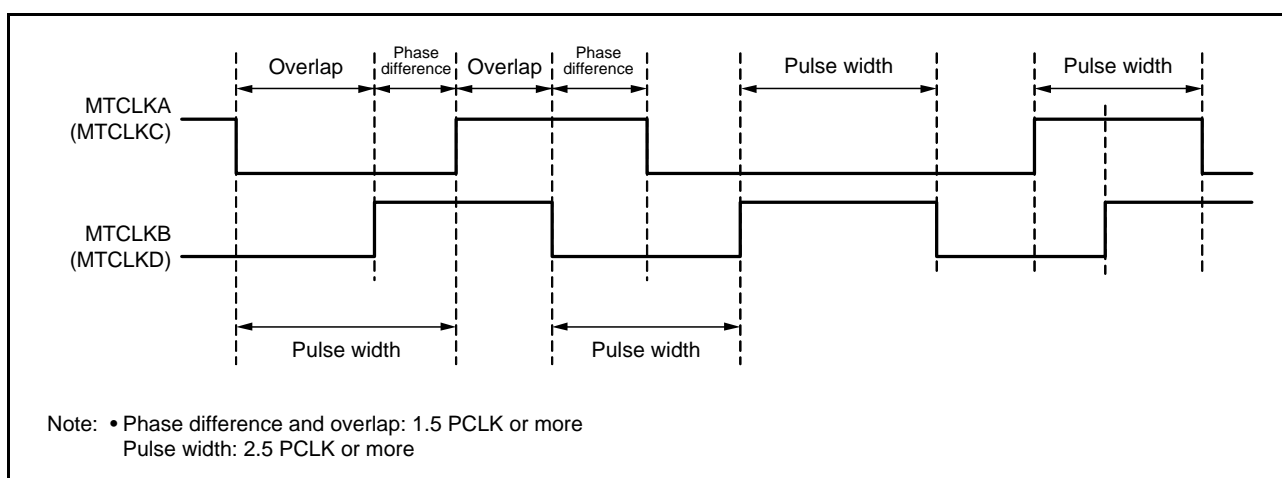


Figure 22.106 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

### 22.6.3 Note on Cycle 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

$$f = \frac{\text{CNTCLK}}{(N+1)}$$

- MTU5

$$f = \frac{\text{CNTCLK}}{N}$$

f : Counter frequency  
CNTCLK : The counter-clock frequency set by TCR.TPSC[2:0]  
N : TGR setting

### 22.6.4 Contention between TCNT Write and Clear Operations

If the counter clear signal is generated in a TCNT write cycle, TCNT clearing takes precedence and TCNT write operation is not performed.

Figure 22.107 shows the timing in this case.

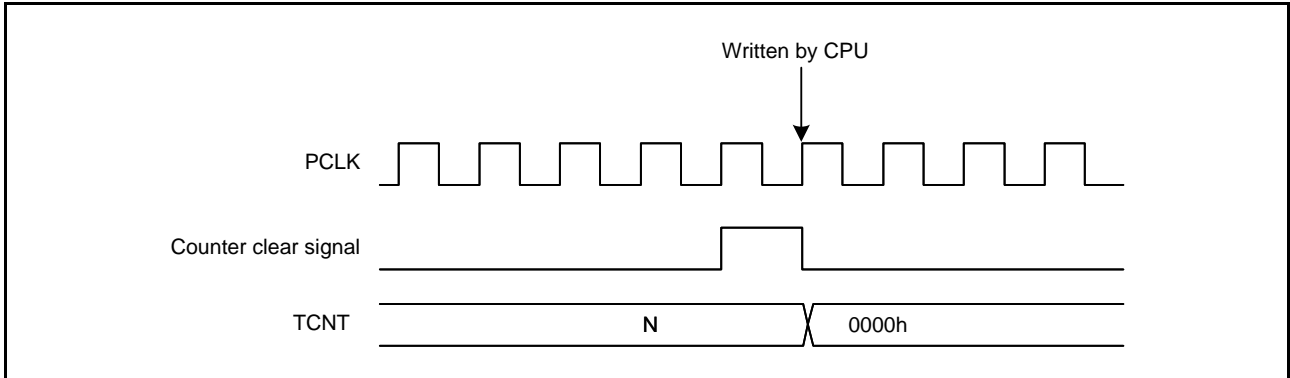


Figure 22.107 Contention between TCNT Write and Counter Clear Operations

### 22.6.5 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 22.108 shows the timing in this case.

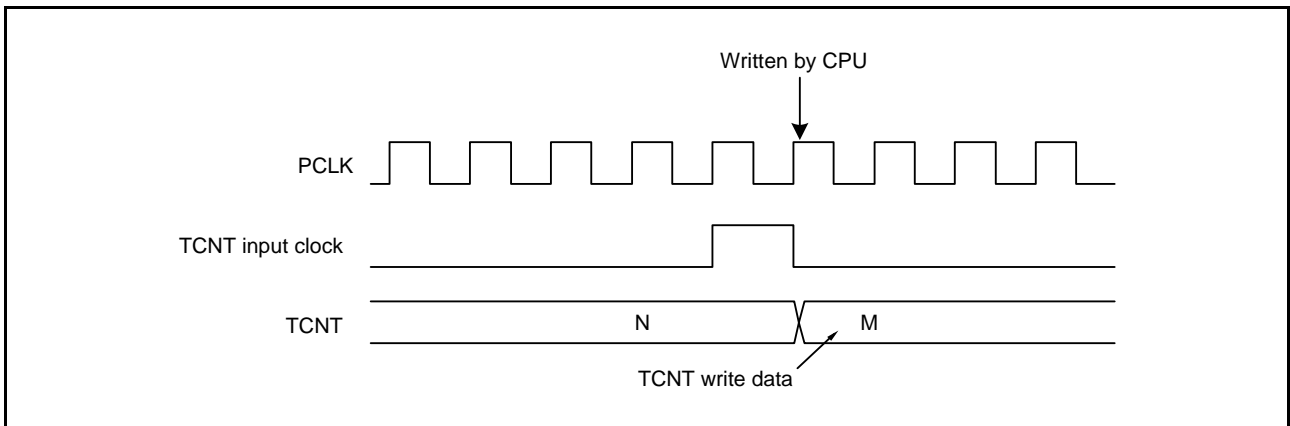


Figure 22.108 Contention between TCNT Write and Increment Operations

### 22.6.6 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 22.109 shows the timing in this case.

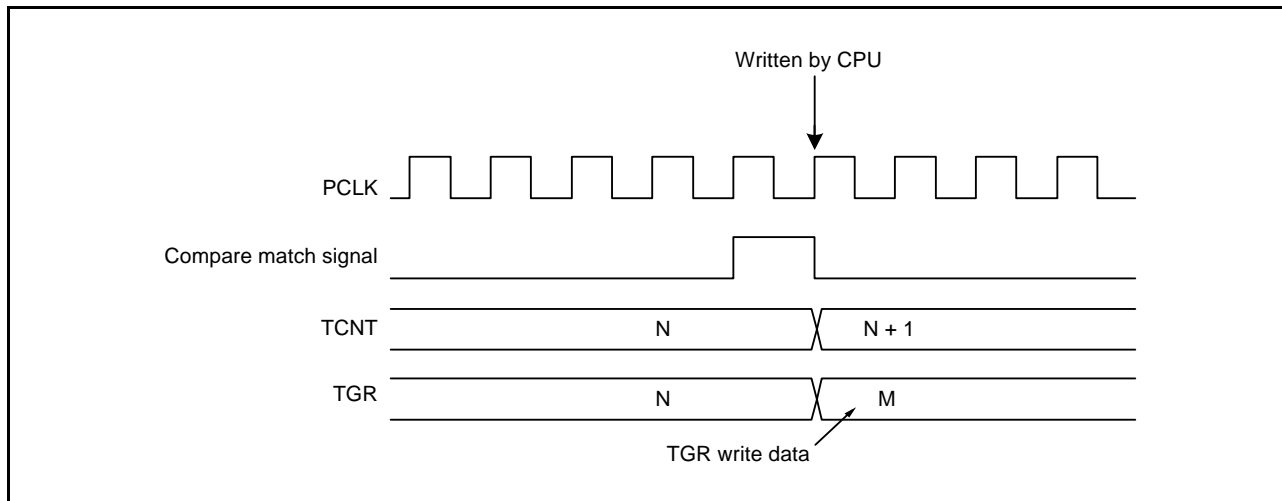


Figure 22.109 Contention between TGR Write Operation and Compare Match

### 22.6.7 Contention between Buffer Register Write Operation and Compare Match

If a compare match occurs in a TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.

Figure 22.110 shows the timing in this case.

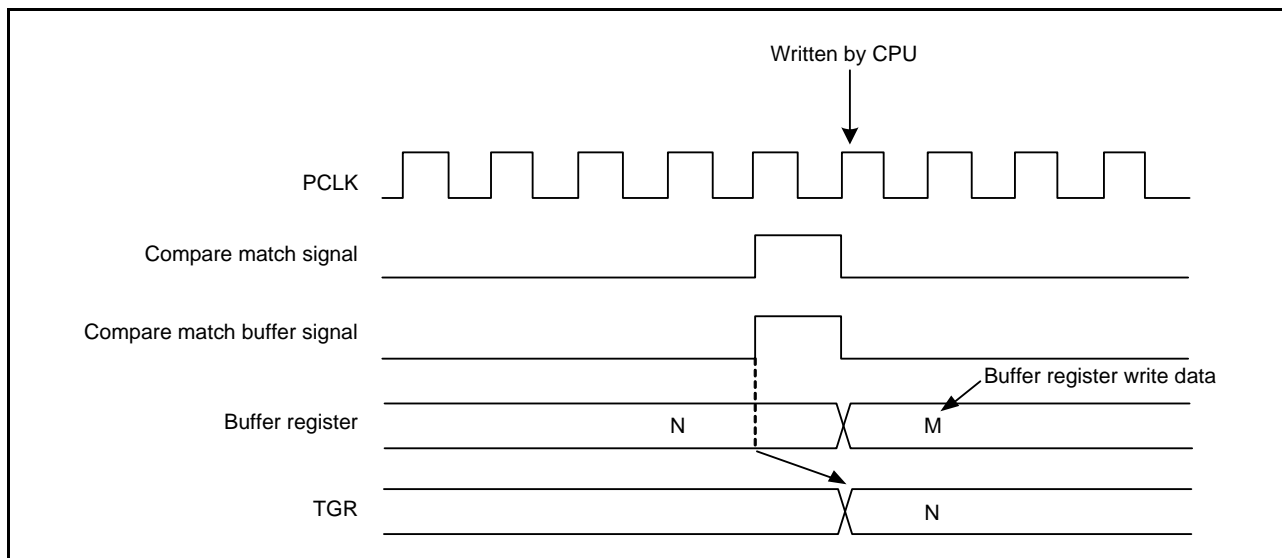


Figure 22.110 Contention between Buffer Register Write Operation and Compare Match

### 22.6.8 Contention between Buffer Register Write and TCNT Clear Operations

When the buffer transfer timing is set at the TCNT clear timing by the buffer transfer mode register (TBTM), if TCNT clearing occurs in a TGR write cycle, the data before write operation is transferred to TGR by the buffer operation.

Figure 22.111 shows the timing in this case.

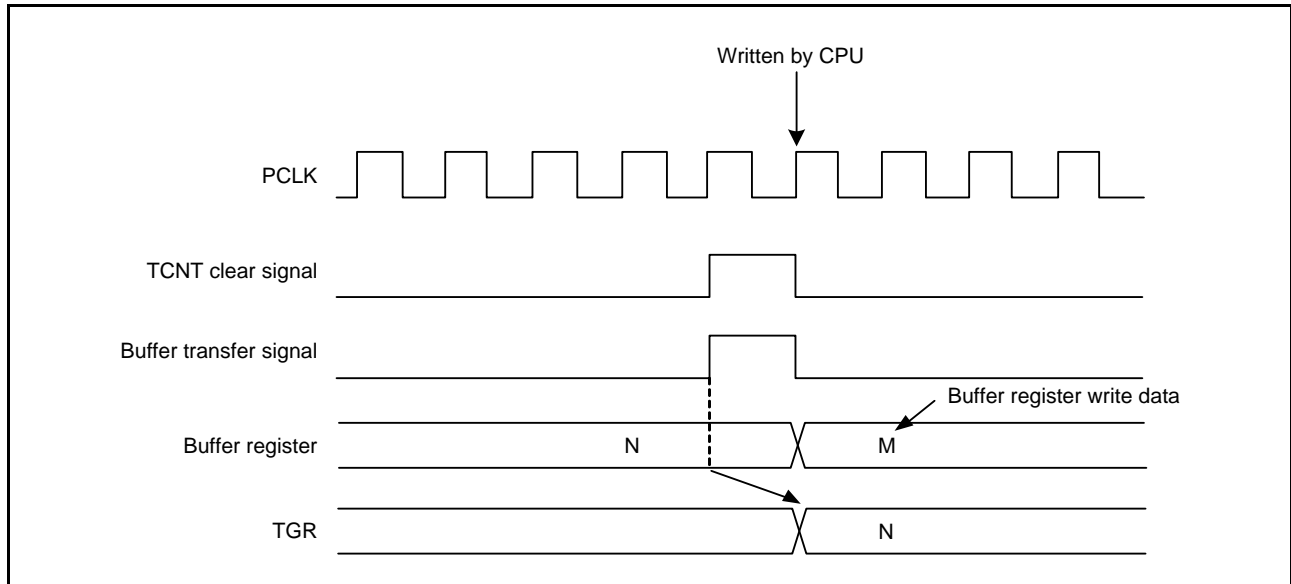


Figure 22.111 Contention between Buffer Register Write and TCNT Clear Operations

### 22.6.9 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 22.112 show the timing in this case.

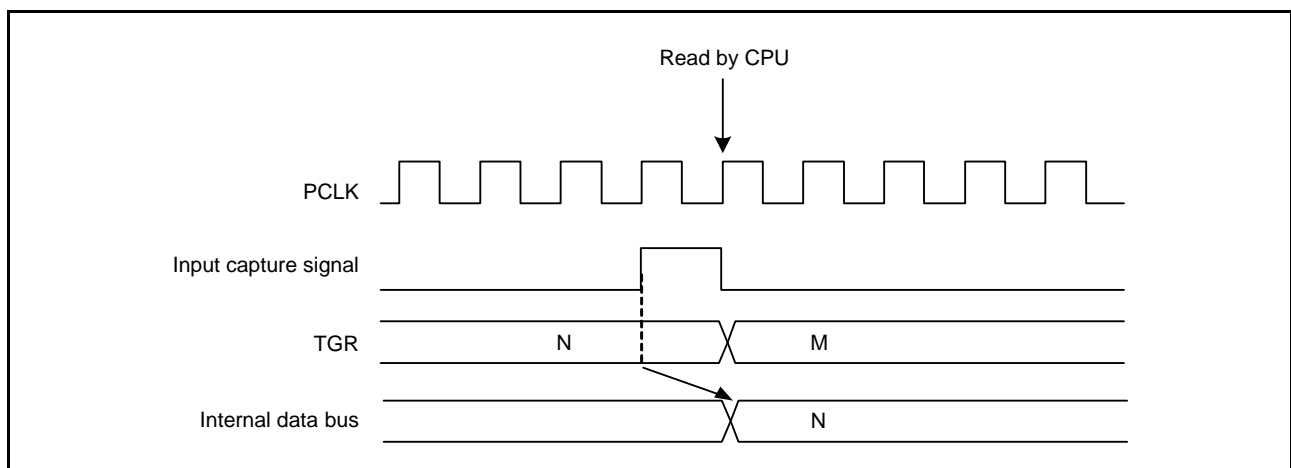


Figure 22.112 Contention between TGR Read Operation and Input Capture (MTU0 to MTU5)

### 22.6.10 Contention between TGR Write Operation and Input Capture

If an input capture signal is generated in a TGR write cycle, the input capture operation takes precedence and the TGR write operation is not performed in MTU0 to MTU4. In MTU5, the TGR write operation is performed and the input capture signal is generated.

Figure 22.113 and Figure 22.114 show the timing in this case.

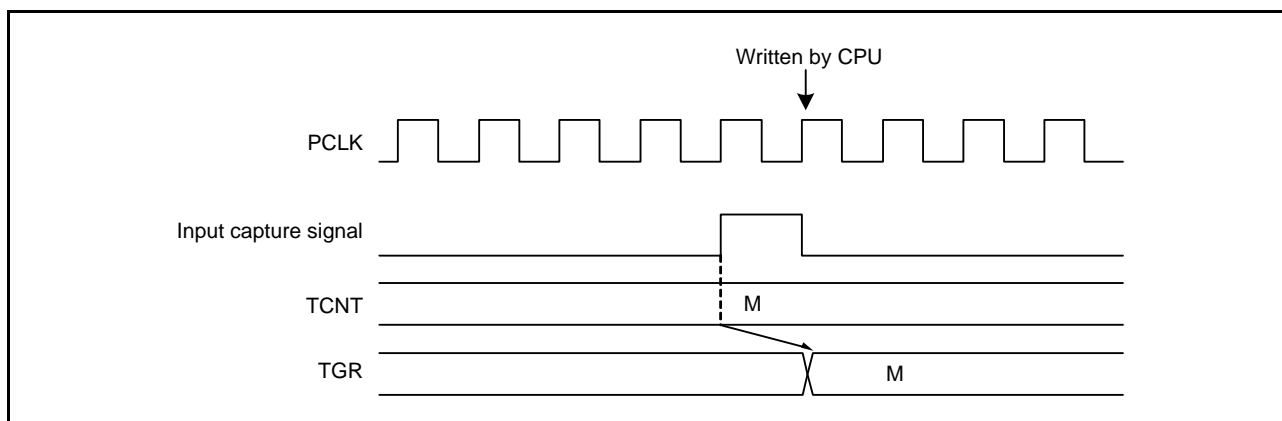


Figure 22.113 Contention between TGR Write Operation and Input Capture (MTU0 to MTU4)

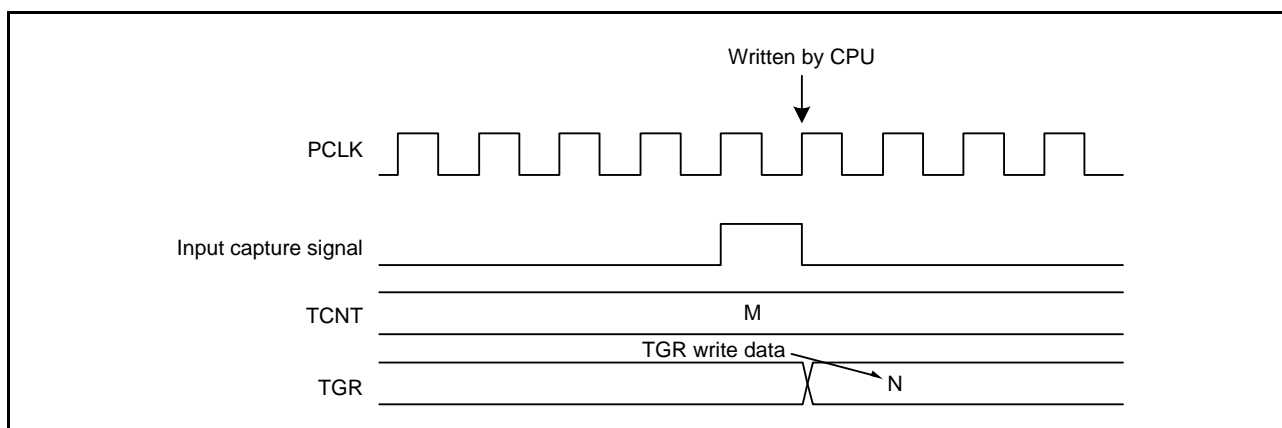


Figure 22.114 Contention between TGR Write Operation and Input Capture (MTU5)

### 22.6.11 Contention between Buffer Register Write Operation and Input Capture

If an input capture signal is generated in a buffer register write cycle, the buffer operation takes precedence and the buffer register write operation is not performed.

Figure 22.115 shows the timing in this case.

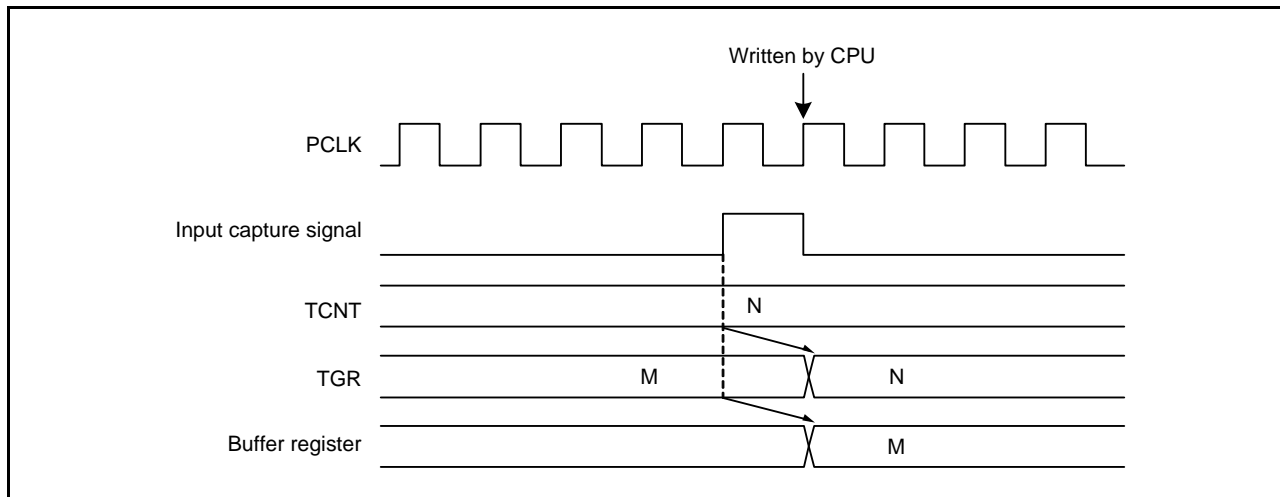


Figure 22.115 Contention between Buffer Register Write Operation and Input Capture

### 22.6.12 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, 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 22.116 shows the timing in this case.

When setting the TCNT clearing function in cascaded operation, be sure to synchronize MTU1 and MTU2.



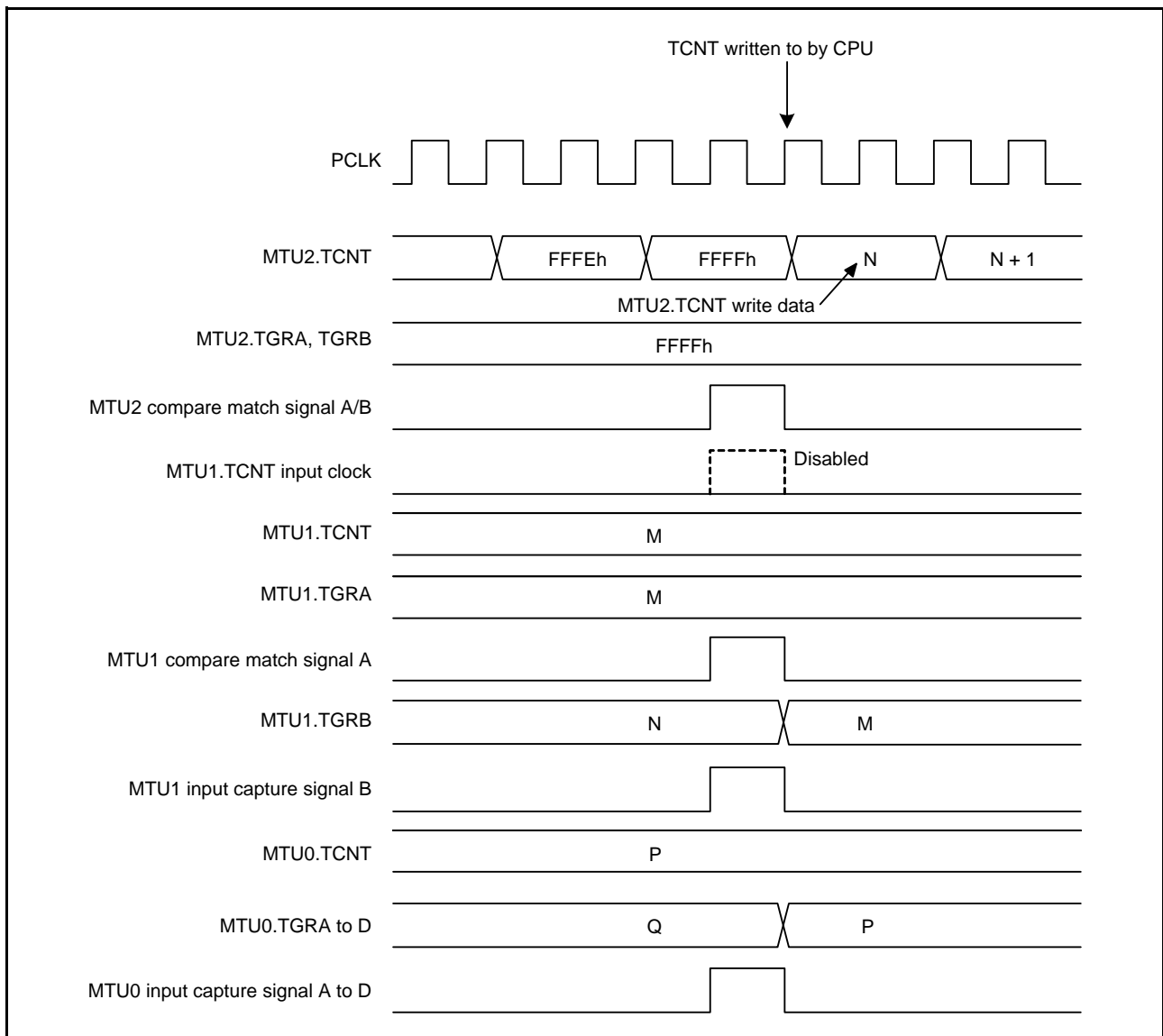


Figure 22.116 Contention between MTU2.TCNT Write Operation and Overflow/Underflow in Cascaded Operation

### 22.6.13 Counter Value when Stopped in Complementary PWM Mode

When counting operation in MTU3.TCNT and MTU4.TCNT is stopped in complementary PWM mode, MTU3.TCNT is set to the timer dead time register (TDDR) 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 22.117 shows this operation.

When counting begins in another operating mode, be sure to make initial settings in MTU3.TCNT and MTU4.TCNT.

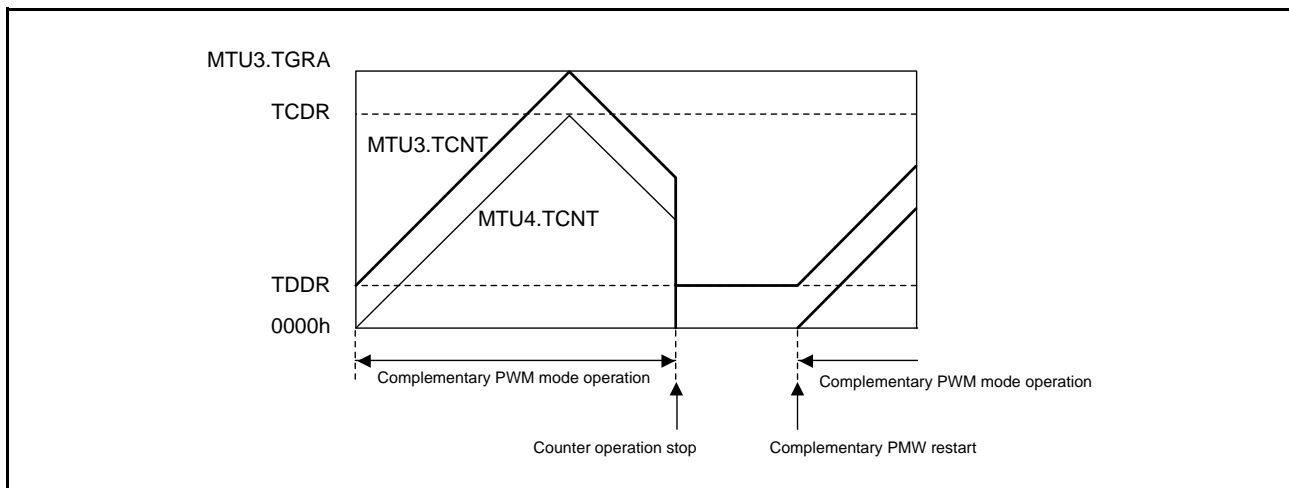


Figure 22.117 Counter Value when Stopped in Complementary PWM Mode (MTU3 and MTU4 Operation)

#### 22.6.14 Buffer Operation Setting in Complementary PWM Mode

When modifying the PWM cycle set register (MTU3.TGRA), timer cycle data register (TCDR), and compare registers (MTU3.TGRB, MTU4.TGRA, and MTU4.TGRB) in complementary PWM mode, be sure to use buffer operation. Also the BFA and BFB bits in MTU4.TMDR should be set to 0. Setting the BFA bit in MTU4.TMDR to 1 disables MTIOC4C pin waveform output. Setting the BFB bit in MTU4.TMDR to 1 also disables MTIOC4D pin waveform output. In complementary PWM mode, buffer operation in MTU3 and MTU4 depends on the settings in bits BFA and BFB of MTU3.TMDR. When the BFA bit in MTU3.TMDR 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 TCBR functions as a buffer register for TCDR.

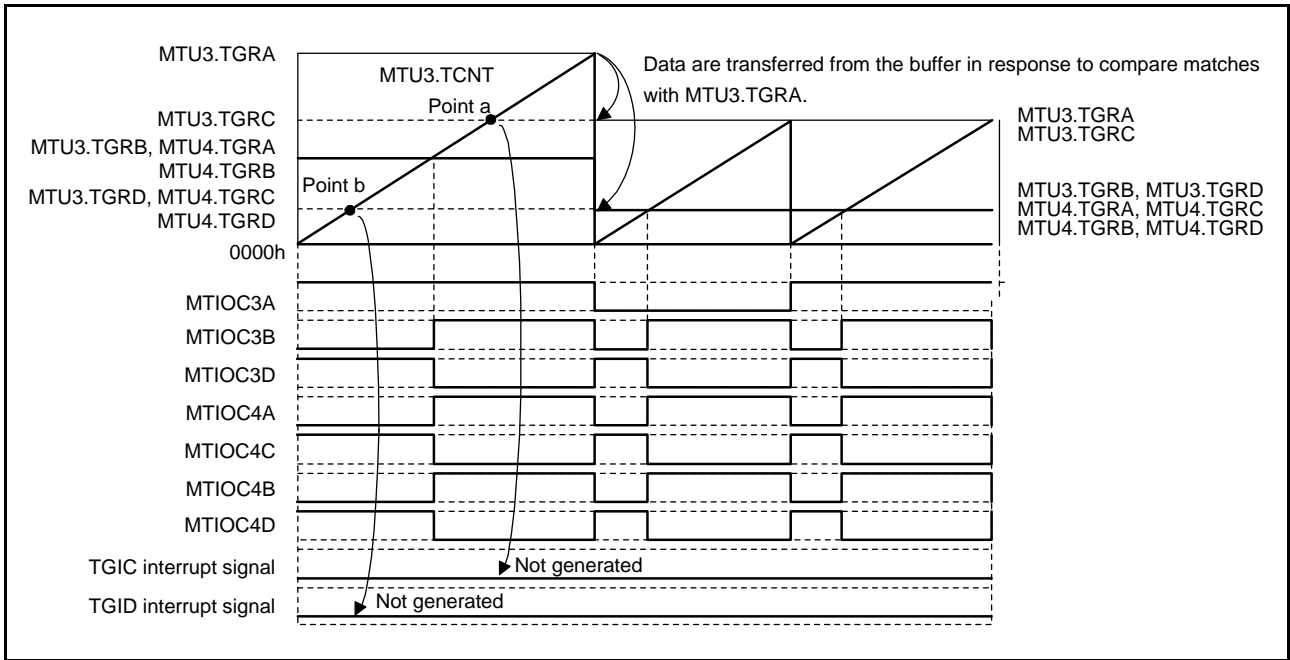
#### 22.6.15 Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode

When setting buffer operation in reset-synchronized PWM mode, set the BFA and BFB bits in MTU4.TMDR to 0. Setting the BFA bit in MTU4.TMDR to 1 disables MTIOC4C pin waveform output. Setting the BFB bit in MTU4.TMDR to 1 also disables MTIOC4D pin waveform output.

In reset-synchronized PWM mode, buffer operation in MTU3 and MTU4 depends on the settings in the BFA and BFB bits of MTU3.TMDR. For example, if the BFA bit in MTU3.TMDR 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, the corresponding TGIC and TGID interrupt requests are never generated.

Figure 22.118 shows an example of MTU3.TGR, MTU4.TGR, MTIOC3m, and MTIOC4m operation with the BFA and BFB bits in MTU3.TMDR set to 1 and the BFA and BFB bits in MTU4.TMDR set to 0. (m = A to D)



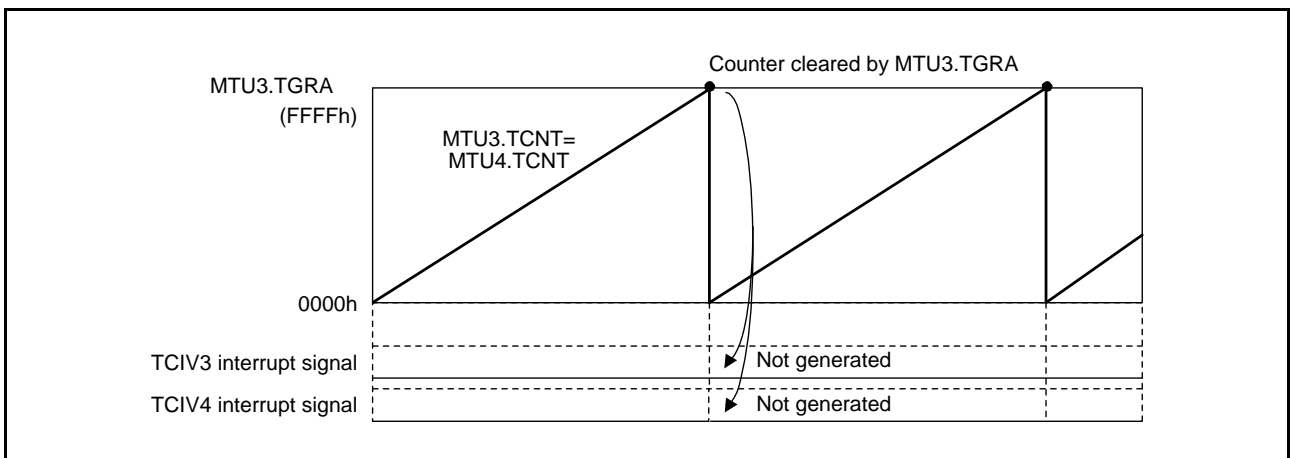
**Figure 22.118 Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode**

### 22.6.16 Overflow Flags in Reset-Synchronized PWM Mode

After reset-synchronized PWM mode is selected, MTU3.TCNT and MTU4.TCNT start counting when the CST3 bit of TSTR 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 cycle 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, the corresponding TCIV interrupt request is not generated.

Figure 22.119 shows an operation example in reset-synchronized PWM mode with cycle register MTU3.TGRA set to FFFFh and the MTU3.TGRA compare match specified for the counter clearing source without synchronous operation setting.



**Figure 22.119 Overflow Flags in Reset-Synchronized PWM Mode**

### 22.6.17 Contention between Overflow/Underflow and Counter Clearing

If an overflow/underflow and counter clearing occur simultaneously, TCNT clearing takes precedence and the corresponding TCIV interrupt is not generated.

Figure 22.120 shows the operation timing when a TGR compare match is specified as the clearing source and TGR is set to FFFFh.

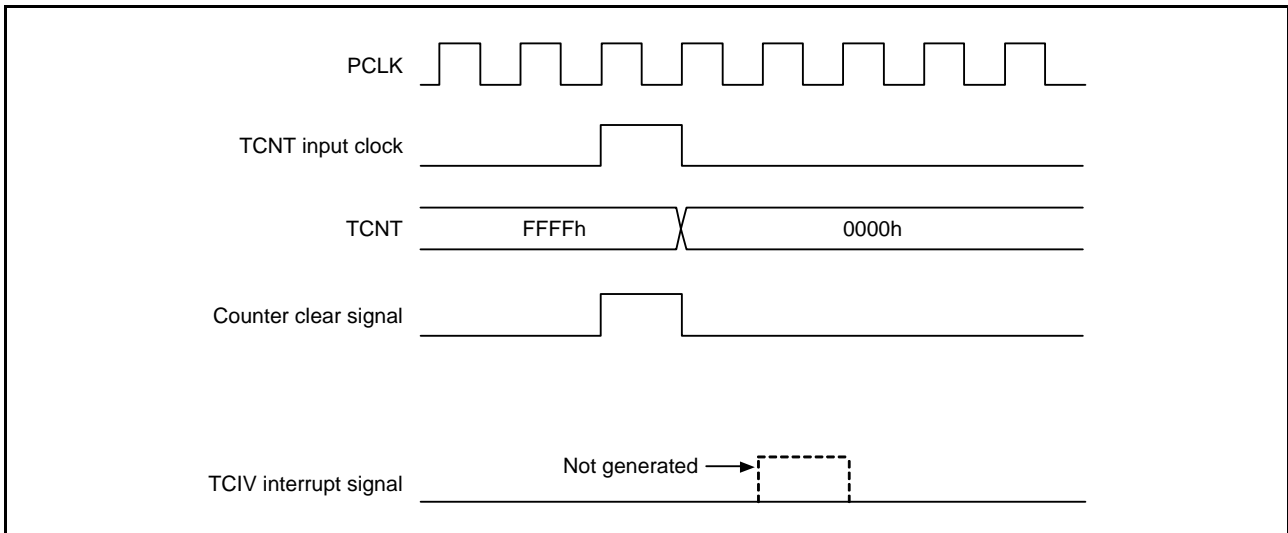


Figure 22.120 Contention between Overflow and Counter Clearing

### 22.6.18 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. The corresponding interrupt is not generated.

Figure 22.121 shows the operation timing when there is contention between TCNT write operation and overflow.

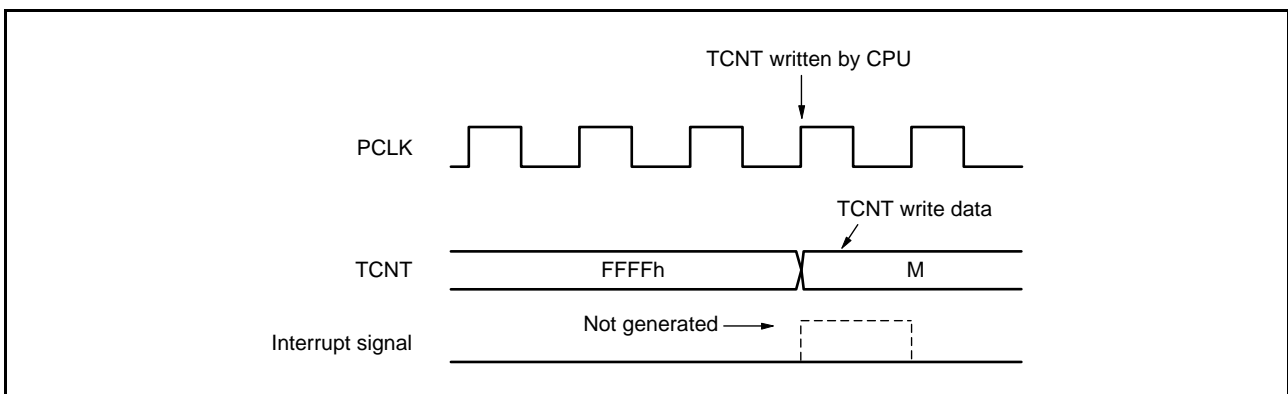


Figure 22.121 Contention between TCNT Write Operation and Overflow

### 22.6.19 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.

### 22.6.20 Output Level in Complementary PWM Mode or Reset-Synchronized PWM Mode

When complementary PWM mode or reset-synchronized PWM mode is selected for MTU3 or MTU4, use the OLSP and OLSN bits in the timer output control register 1 (TOCR1) to set the levels for PWM waveform output. Also, when either of these modes is in use, set TIOR to 00h.

### 22.6.21 Interrupts During Periods in the Module-Stop State

When an module that has issued an interrupt request enters the module-stop state, clearing the source of the interrupt for the CPU or activation signal for the DMACA or DTC is not possible.

Accordingly, disable interrupts, etc. before making the settings for the module-stop state.

### 22.6.22 Simultaneous Input Capture in MTU1.TCNT and MTU2.TCNT in Cascade Connection

When timer counters (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 of 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 new function that allows simultaneous capture of MTU1.TCNT and MTU2.TCNT with a single input-capture input as the trigger. This function allows reading of the 32-bit counter such that MTU1.TCNT and MTU2.TCNT are captured at the same time. For details, see section 22.2.8, Timer Input Capture Control Register (TICCR).

### 22.6.23 Notes when Complementary PWM Mode Output Protection Functions are not Used

The complementary PWM mode output protection functions are initially enabled. If the functions are not used, the POE.POECR2 register should be set to 00h.

### 22.6.24 Point for Caution Regarding MTU5.TCNT and MTU5.TGR Registers

Do not set an MTU5.TGR<sub>m</sub> (m = U, V, W) bit to the value of the corresponding MTU5.TCNT<sub>m</sub> (m = U, V, W) register plus one while counting by the MTU5.TCNT<sub>m</sub> (m = U, V, W) register is stopped. If an MTU5.TGR<sub>m</sub> (m = U, V, W) bit is set to the value of the corresponding MTU5.TCNT<sub>m</sub> (m = U, V, W) register plus one while counting by the MTU5.TCNT<sub>m</sub> (m = U, V, W) register is stopped, a compare-match will be generated even though counting is stopped. In this case, if the corresponding MTU5.TIER.TGIE5<sub>m</sub> (m = U, V, W) bit is also set to 1 (enabling interrupts), a compare-match interrupt will also be generated. If the value of the timer compare match clear register is also 1 (enabled), the timer is automatically cleared to 0000h when the compare-match is generated, regardless of whether interrupts from the MTU5.TCNT<sub>m</sub> (m = U, V, W) are enabled or disabled.

### 22.6.25 Points for Caution to Prevent Malfunctions in Synchronous Clearing for Complementary PWM Mode

If control of the output waveform is enabled (TWCR.WRE = 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 PWM negative phase output pins beyond the period for active-level output.

Condition 1: In portion (10) of the initial output inhibition period in Figure 22.122, 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 22.123, synchronous clearing occurs when any condition from among  $MTU3.TGRB \leq TDDR$ ,  $MTU4.TGRA \leq TDDR$ , or  $MTU4.TGRB \leq TDDR$  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 dead time data register (TDDR).

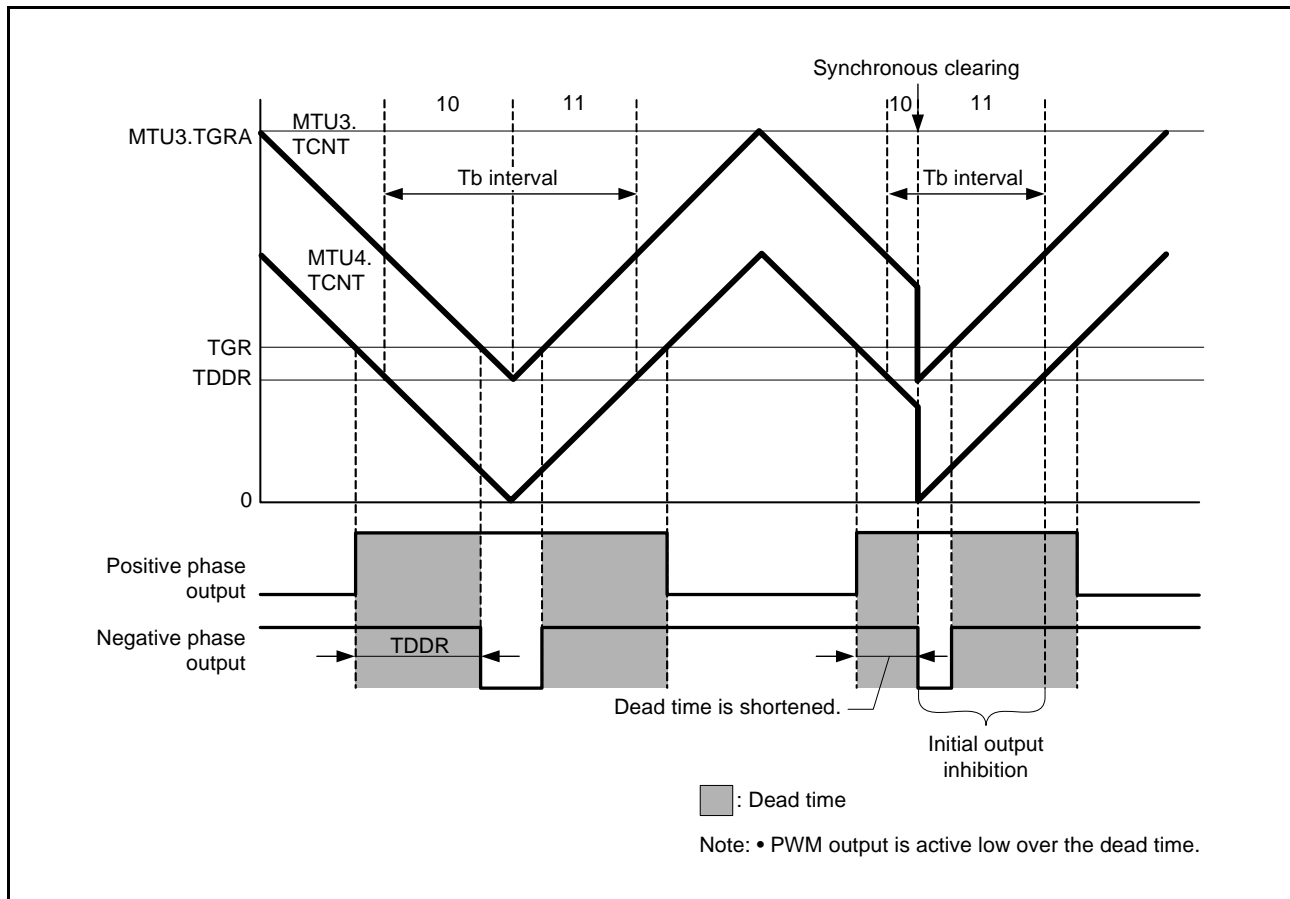


Figure 22.122 Example of Synchronous Clearing (when Condition 1 Applies)

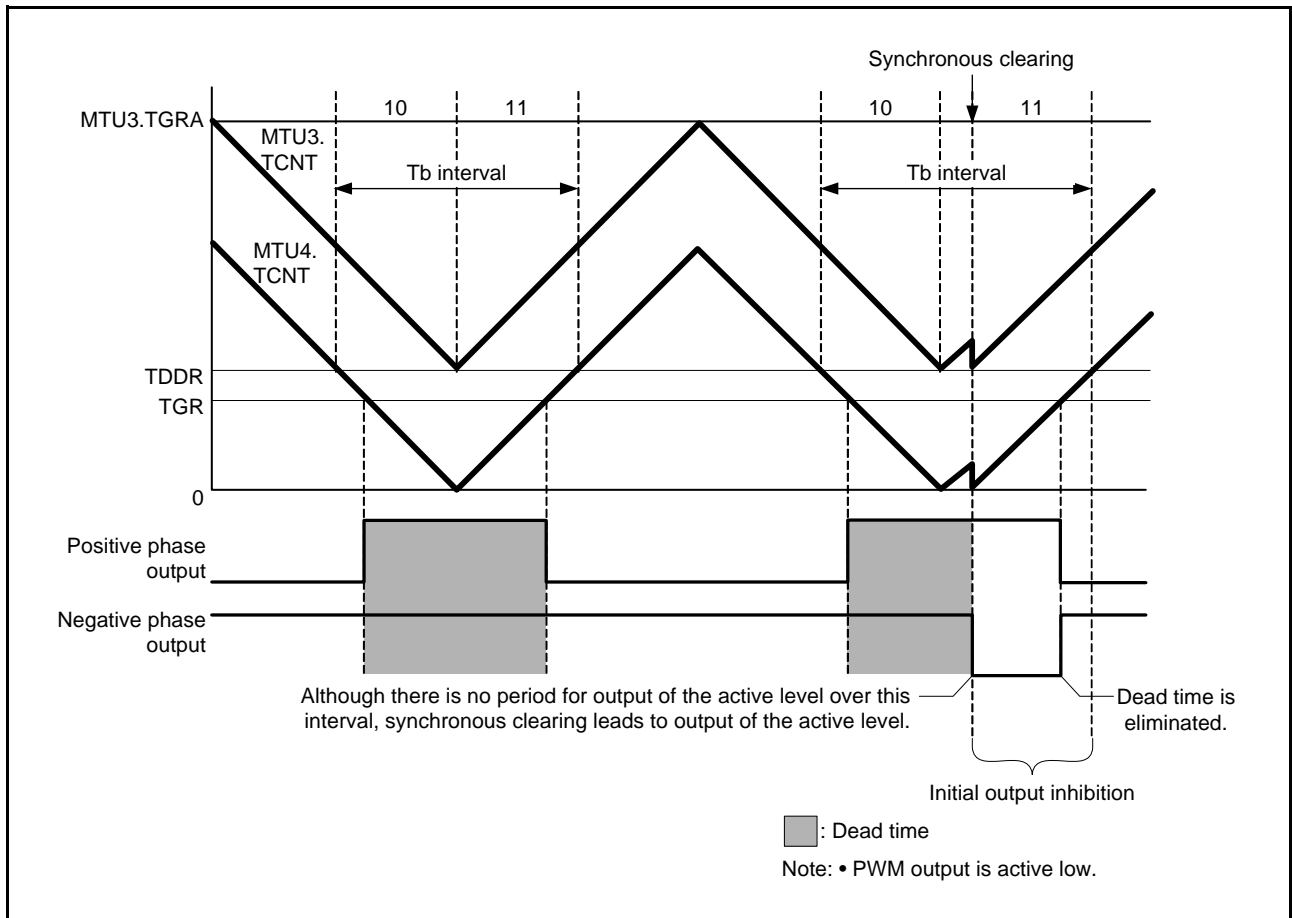


Figure 22.123 Example of Synchronous Clearing (when Condition 2 Applies)

### 22.6.26 Continuous Output of Interrupt Signal in Response to a Compare Match

When TGR is set to 0000h, PCLK/1 is set as the counter clock, and compare match is set as the trigger for clearing of the counter clock, the value of the counter (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 22.124 shows the timing for continuous output of the interrupt signal in response to a compare match.

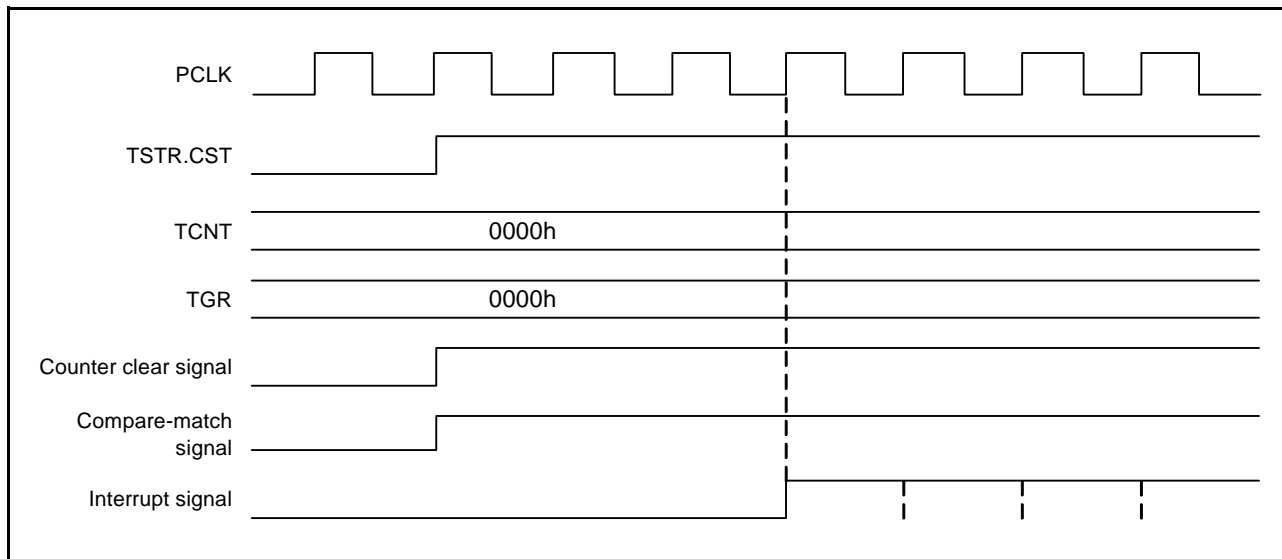


Figure 22.124 Continuous Output of Interrupt Signal in Response to a Compare Match



## 22.7 MTU Output Pin Initialization

### 22.7.1 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 4 (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.

### 22.7.2 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. For an I/O port that is shut down, set the port-direction registers (PDR), the port output data register (PODR), and the port mode register (PMR) to switch the port pins to be general output pins and for output of the non-active level. Set the TIOR for the MTU pins to disable output. Set the TOER for the complementary PWM output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D). For PWM output pins, output can also be cut by hardware, using port output enable 2 (POE). 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 listed in Table 22.59.

**Table 22.59 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 4

CPWM: Complementary PWM mode

RPWM: Reset-synchronized PWM mode

### 22.7.3 Overview of Pin Initialization Procedures and Mode Transitions in Case of Error during Operation

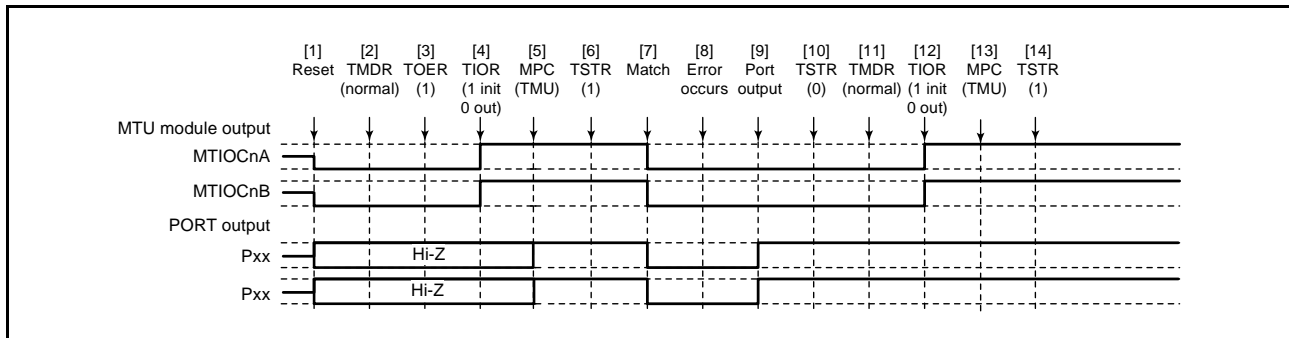
- When making a transition to a mode (Normal, PWM1, PWM2, or PCM) in which the pin output level is selected by the timer I/O control register (TIOR) setting, initialize the pins by means of TIOR setting.
- In PWM mode 1, since a waveform is not output to the MTIOCnB (MTIOCnD) pins, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 1.
- In PWM mode 2, since a waveform is not output to the cycle register pins, setting TIOR will not initialize the pins. If initialization is required, carry it out in normal mode, then switch to PWM mode 2.
- In normal mode or PWM mode 2, if TGRC and TGRD operate as buffer registers, setting TIOR will not initialize the buffer register pins. If initialization is required, clear buffer mode, carry out initialization, then set buffer mode again.
- In PWM mode 1, if either TGRC or TGRD operates as a buffer register, setting TIOR will not initialize the TGRC pin. To initialize the TGRC pin, clear buffer mode, carry out initialization, then set buffer mode again.
- When making a transition to a mode (CPWM or RPWM) in which the pin output level is selected by the timer output control register (TOCR) setting, switch to normal mode, perform initialization with TIOR, restore TIOR to its initial value, and temporarily disable output in MTU3 and MTU4 with the timer output master enable register (TOER). After that, operate the MTU in accordance with the mode setting procedure (TOCR setting, TMDR setting, and TOER setting).

Note: • Channel number is substituted for “n” indicated in this section.

Pin initialization procedures are described below for the numbered combinations in Table 22.59. The active level is assumed to be low.

## (1) Operation when Error Occurs in Normal Mode and Operation is Restarted in Normal Mode

Figure 22.125 shows a case in which an error occurs in normal mode and operation is restarted in normal mode after re-setting.



**Figure 22.125 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 TMDR setting is for normal mode.
- [3] For MTU3 and MTU4, enable output with TOER before initializing the pins with TIOR.
- [4] Initialize the pins with TIOR. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)
- [5] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
- [6] Start count operation by setting TSTR.
- [7] Output goes low on compare match occurrence.
- [8] An error occurs.
- [9] Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
- [10] Stop count operation by setting TSTR.
- [11] This step is not necessary when restarting in normal mode.
- [12] Initialize the pins with TIOR.
- [13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
- [14] Restart operation by setting TSTR.

(2) Operation when Error Occurs in Normal Mode and Operation is Restarted in PWM Mode 1

Figure 22.126 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 1 after re-setting.

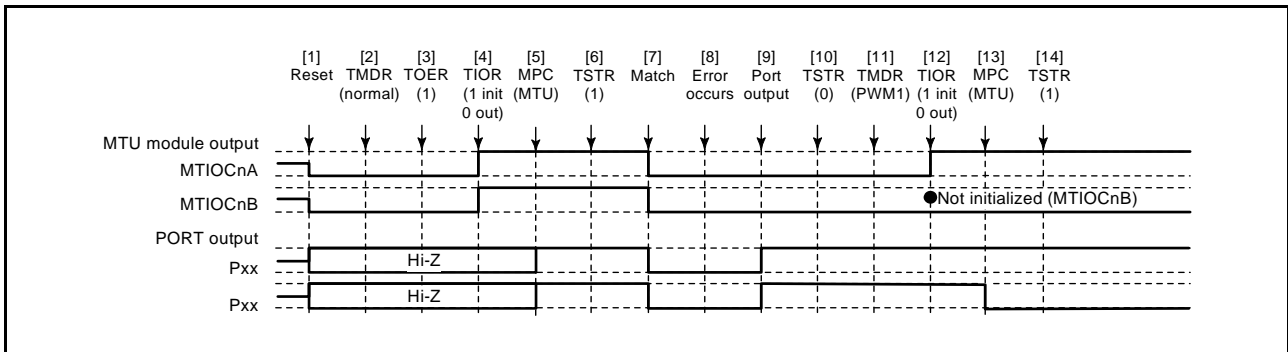


Figure 22.126 Error Occurrence in Normal Mode, Recovery in PWM Mode 1

[1] to [10] are the same as in Figure 22.125.

[11] Set PWM mode 1.

[12] Set the TIOR register to initialize pins, i.e. so that the MTIOCnB (or MTIOCnD) does not produce a waveform in PWM mode 1. If a particular level should be output, set the port direction register (PDR) and the port output data register (PODR) so that the pins of the I/O port operate as general outputs.

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

(3) Operation when Error Occurs in Normal Mode and Operation is Restarted in PWM Mode 2

Figure 22.127 shows a case in which an error occurs in normal mode and operation is restarted in PWM mode 2 after re-setting.

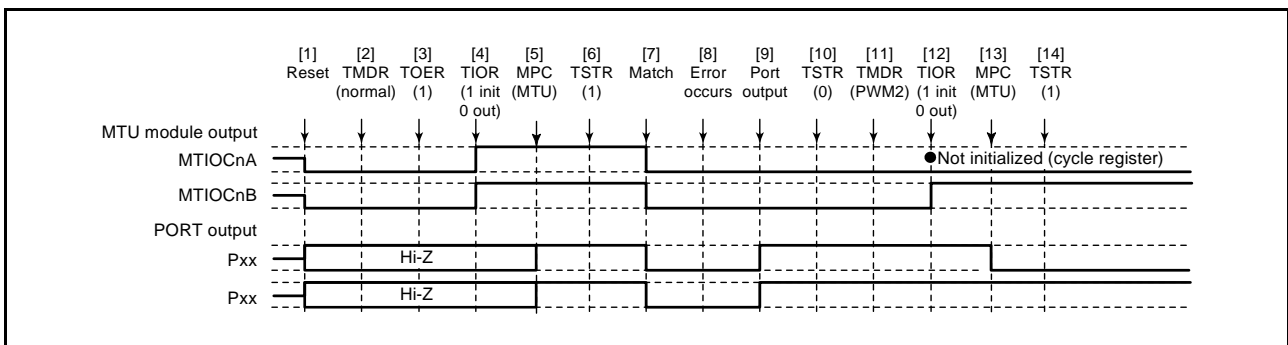


Figure 22.127 Error Occurrence in Normal Mode, Recovery in PWM Mode 2

[1] to [10] are the same as in Figure 22.125.

[11] Set PWM mode 2.

[12] Initialize the pins with TIOR. (In PWM mode 2, the cycle register pins are not initialized. If initialization is required, initialize in normal mode, then switch to PWM mode 2.)

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

Note: • PWM mode 2 can only be selected for MTU0 to MTU2, and therefore TOER setting is not necessary.

(4) Operation when Error Occurs in Normal Mode and Operation is Restarted in Phase Counting Mode

Figure 22.128 shows a case in which an error occurs in normal mode and operation is restarted in phase counting mode after re-setting.

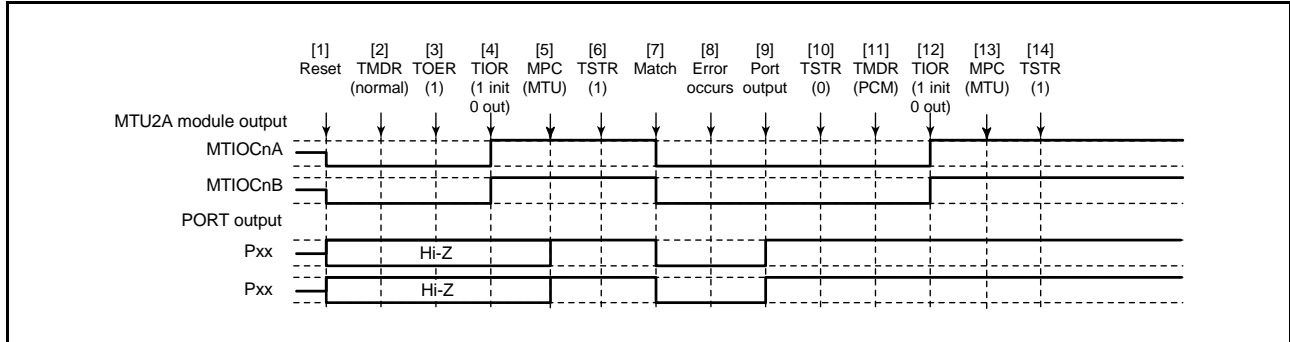


Figure 22.128 Error Occurrence in Normal Mode, Recovery in Phase Counting Mode

[1] to [10] are the same as in Figure 22.125.

[11] Set the phase counting mode.

[12] Initialize the pins with TIOR.

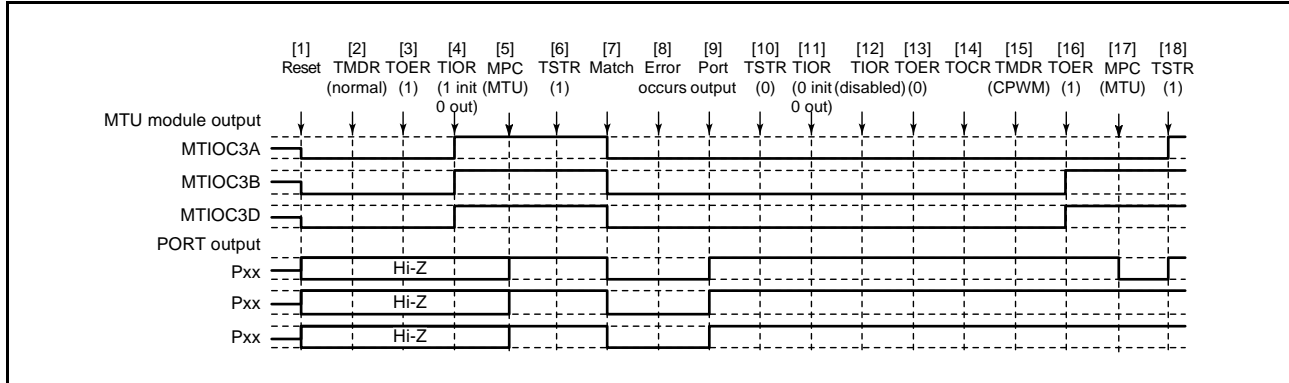
[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

Note: • The phase counting mode can only be selected for MTU1 and MTU2, and therefore TOER setting is not necessary.

### (5) Operation when Error Occurs in Normal Mode and Operation is Restarted in Complementary PWM Mode

Figure 22.129 shows a case in which an error occurs in normal mode and operation is restarted in complementary PWM mode after re-setting.



**Figure 22.129 Error Occurrence in Normal Mode, Recovery in Complementary PWM Mode**

[1] to [10] are the same as in Figure 22.125.

[11] Initialize the normal mode waveform generation section with TIOR.

[12] Disable operation of the normal mode waveform generation section with TIOR.

[13] Disable output in MTU3 and MTU4 with TOER.

[14] Select the complementary PWM output level and enable or disable cyclic output with TOCR.

[15] Set complementary PWM mode.

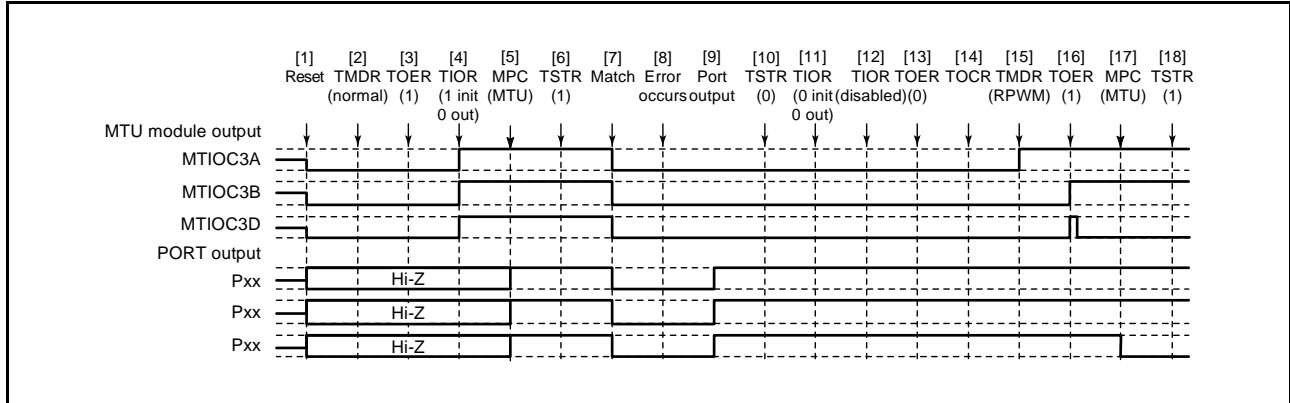
[16] Enable output in MTU3 and MTU4 with TOER.

[17] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[18] Restart operation by setting TSTR.

(6) Operation when Error Occurs in Normal Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 22.130 shows a case in which an error occurs in normal mode and operation is restarted in reset-synchronized PWM mode after re-setting.



**Figure 22.130 Error Occurrence in Normal Mode, Recovery in Reset-Synchronized PWM Mode**

[1] to [13] are the same as in Figure 22.125.

[14] Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR.

[15] Set reset-synchronized PWM mode.

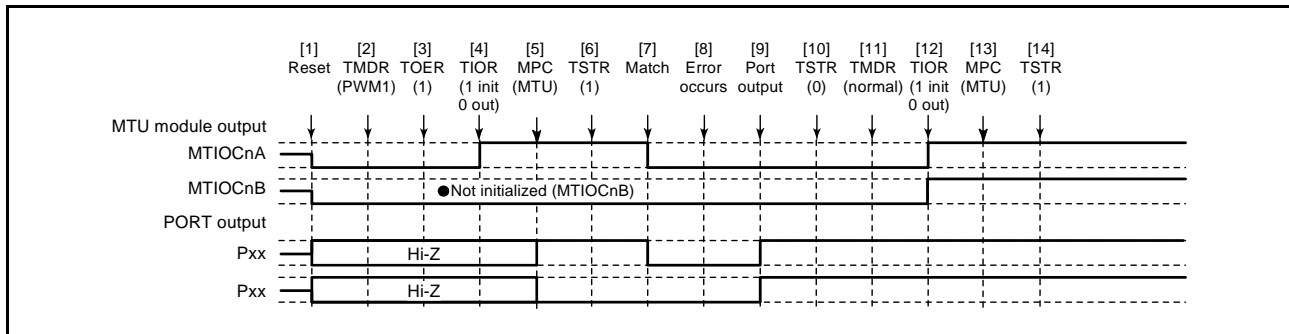
[16] Enable output in MTU3 and MTU4 with TOER.

[17] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[18] Restart operation by setting TSTR.

## (7) Operation when Error Occurs in PWM Mode 1 and Operation is Restarted in Normal Mode

Figure 22.131 shows a case in which an error occurs in PWM mode 1 and operation is restarted in normal mode after re-setting.



**Figure 22.131 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 TOER before initializing the pins with TIOR.
- [4] Initialize the pins with TIOR. (In the example, the initial output is a high level, and a low level is output on compare match occurrence. In PWM mode 1, the MTIOCnB side is not initialized.)
- [5] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
- [6] Start count operation by setting TSTR.
- [7] Output goes low on compare match occurrence.
- [8] An error occurs.
- [9] Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
- [10] Stop count operation by setting TSTR.
- [11] Set normal mode.
- [12] Initialize the pins with TIOR.
- [13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
- [14] Restart operation by setting TSTR.



(8) Operation when Error Occurs in PWM Mode 1 and Operation is Restarted in PWM Mode 1

Figure 22.132 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 1 after re-setting.

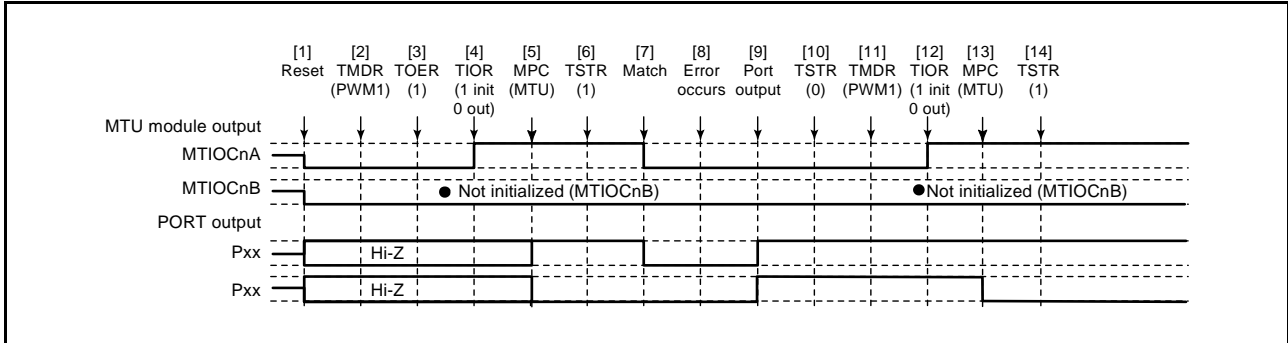


Figure 22.132 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 1

[1] to [10] are the same as in Figure 22.131.

[11] This step is not necessary when restarting in PWM mode 1.

[12] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR)).

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

(9) Operation when Error Occurs in PWM Mode 1 and Operation is Restarted in PWM Mode 2

Figure 22.133 shows a case in which an error occurs in PWM mode 1 and operation is restarted in PWM mode 2 after re-setting.

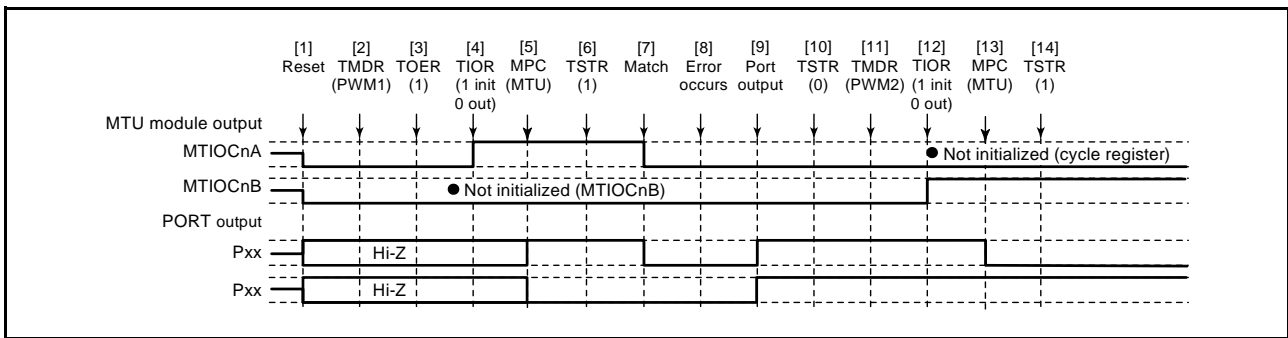


Figure 22.133 Error Occurrence in PWM Mode 1, Recovery in PWM Mode 2

[1] to [10] are the same as in Figure 22.131.

[11] Set PWM mode 2.

[12] Initialize the pins with the TIOR register. (In PWM mode 2, a waveform is not output on the cycle register pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR)).

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

Note: • PWM mode 2 can only be selected for MTU0 to MTU2, and therefore, TOER setting is not necessary.

(10) Operation when Error Occurs in PWM Mode 1 and Operation is Restarted in Phase Counting Mode

Figure 22.134 shows a case in which an error occurs in PWM mode 1 and operation is restarted in phase counting mode after re-setting.

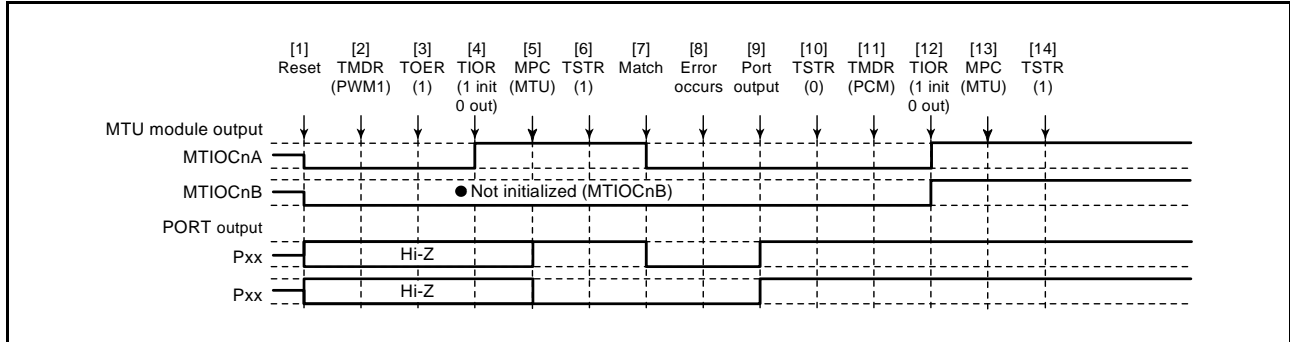


Figure 22.134 Error Occurrence in PWM Mode 1, Recovery in Phase Counting Mode

[1] to [10] are the same as in Figure 22.131.

[11] Set the phase counting mode.

[12] Initialize the pins with TIOR.

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

Note: • The phase counting mode can only be selected for MTU1 and MTU2, and therefore TOER setting is not necessary.

(11) Operation when Error Occurs in PWM Mode 1 and Operation is Restarted in Complementary PWM Mode

Figure 22.135 shows a case in which an error occurs in PWM mode 1 and operation is restarted in complementary PWM mode after re-setting.

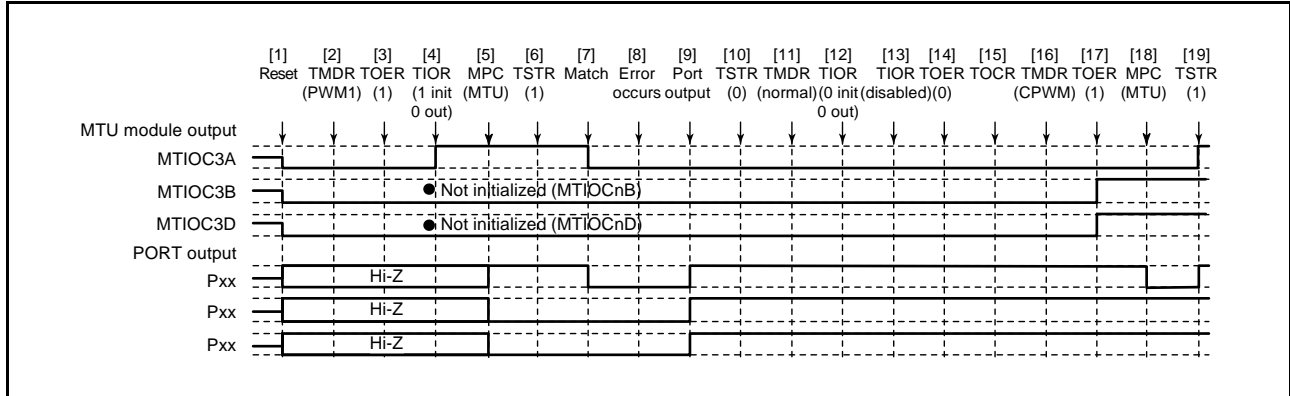


Figure 22.135 Error Occurrence in PWM Mode 1, Recovery in Complementary PWM Mode

[1] to [10] are the same as in Figure 22.131.

[11] Set normal mode to initialize the normal mode waveform generation section.

[12] Initialize the PWM mode 1 waveform generation section with TIOR.

[13] Disable operation of the PWM mode 1 waveform generation section with TIOR

[14] Disable output in MTU3 and MTU4 with TOER.

[15] Select the complementary PWM output level and enable or disable cyclic output with TOCR.

[16] Set complementary PWM mode.

[17] Enable output in MTU3 and MTU4 with TOER.

[18] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[19] Restart operation by setting TSTR.

(12) Operation when Error Occurs in PWM Mode 1 and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 22.136 shows a case in which an error occurs in PWM mode 1 and operation is restarted in reset-synchronized PWM mode after re-setting.

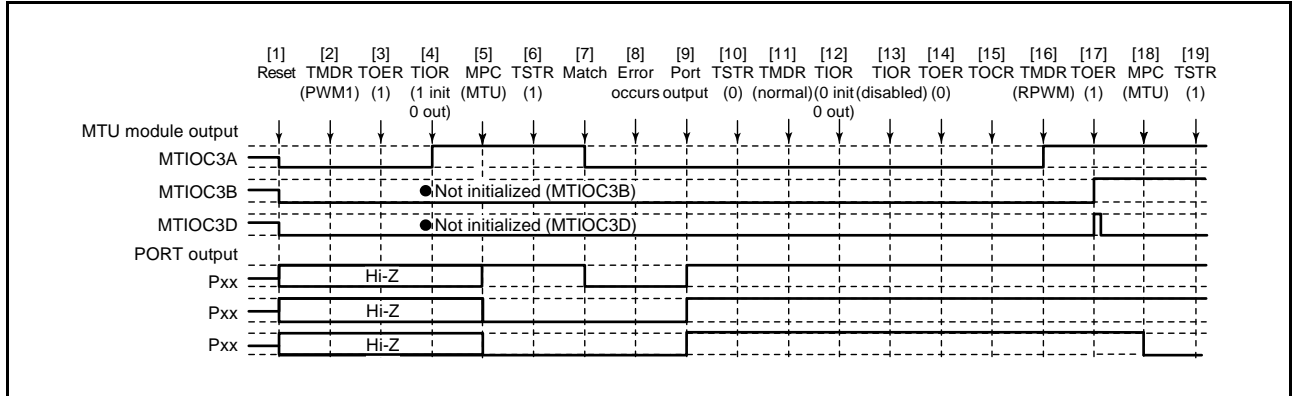


Figure 22.136 Error Occurrence in PWM Mode 1, Recovery in Reset-Synchronized PWM Mode

[1] to [14] are the same as in Figure 22.135.

[15] Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR.

[16] Set reset-synchronized PWM mode.

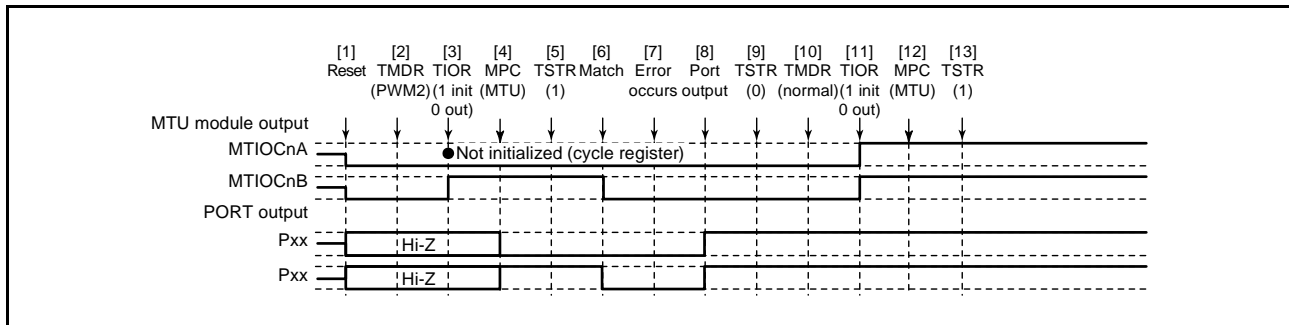
[17] Enable output in MTU3 and MTU4 with TOER.

[18] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[19] Restart operation by setting TSTR.

## (13) Operation when Error Occurs in PWM Mode 2 and Operation is Restarted in Normal Mode

Figure 22.137 shows a case in which an error occurs in PWM mode 2 and operation is restarted in normal mode after re-setting.



**Figure 22.137 Error Occurrence in PWM Mode 2, Recovery in Normal Mode**

- [1] After a reset, the MTU output goes low and the ports enter high-impedance state.
- [2] Set PWM mode 2.
- [3] Initialize the pins with TIOR. (In the example, the initial output is a high level, and a low level is output on compare match occurrence. In PWM mode 2, the cycle register pins are not initialized. In the example, MTIOCnA is the cycle register.)
- [4] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
- [5] Start count operation by setting TSTR.
- [6] Output goes low on compare match occurrence.
- [7] An error occurs.
- [8] Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
- [9] Stop count operation by setting TSTR.
- [10] Set normal mode.
- [11] Initialize the pins with TIOR.
- [12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
- [13] Restart operation by setting TSTR.

(14) Operation when Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 1

Figure 22.138 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 1 after re-setting.

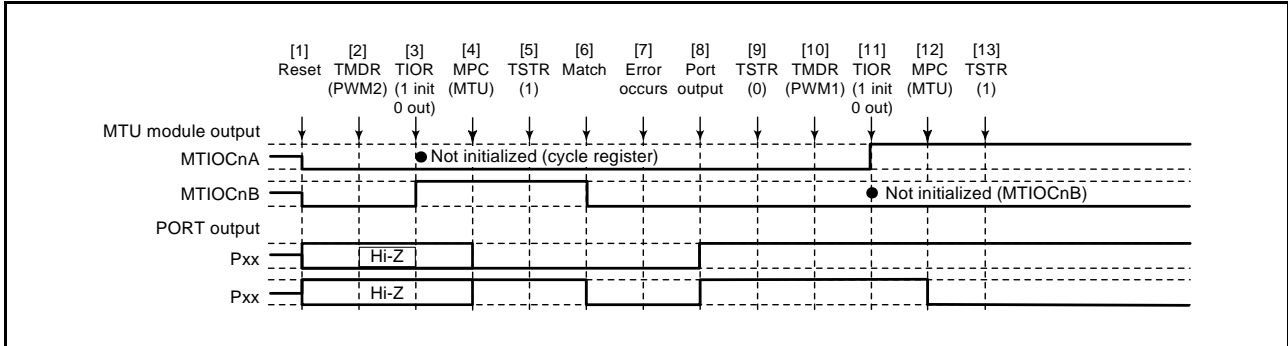


Figure 22.138 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 1

[1] to [9] are the same as in Figure 22.137.

[10] Set PWM mode 1.

[11] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR)).

[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[13] Restart operation by setting TSTR.

(15) Operation when Error Occurs in PWM Mode 2 and Operation is Restarted in PWM Mode 2

Figure 22.139 shows a case in which an error occurs in PWM mode 2 and operation is restarted in PWM mode 2 after re-setting.

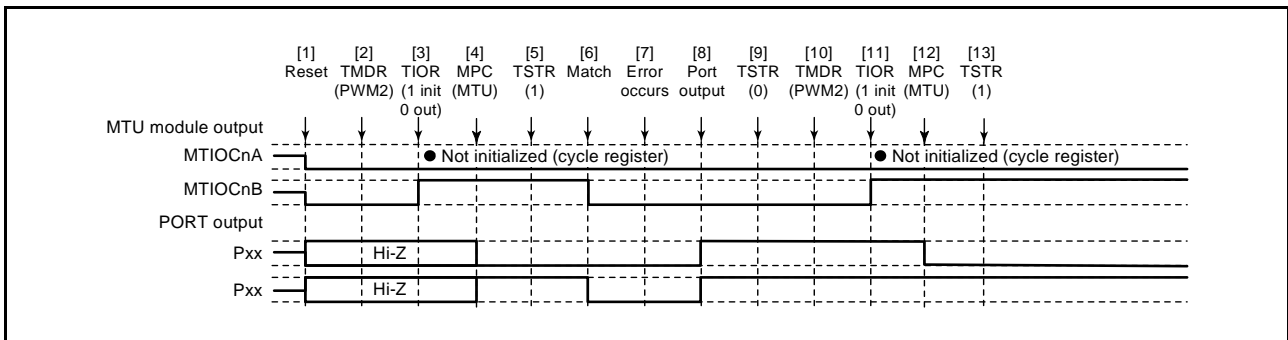


Figure 22.139 Error Occurrence in PWM Mode 2, Recovery in PWM Mode 2

[1] to [9] are the same as in Figure 22.137.

[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 on the cycle register pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR)).

[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[13] Restart operation by setting TSTR.

(16) Operation when Error Occurs in PWM Mode 2 and Operation is Restarted in Phase Counting Mode

Figure 22.140 shows a case in which an error occurs in PWM mode 2 and operation is restarted in phase counting mode after re-setting.

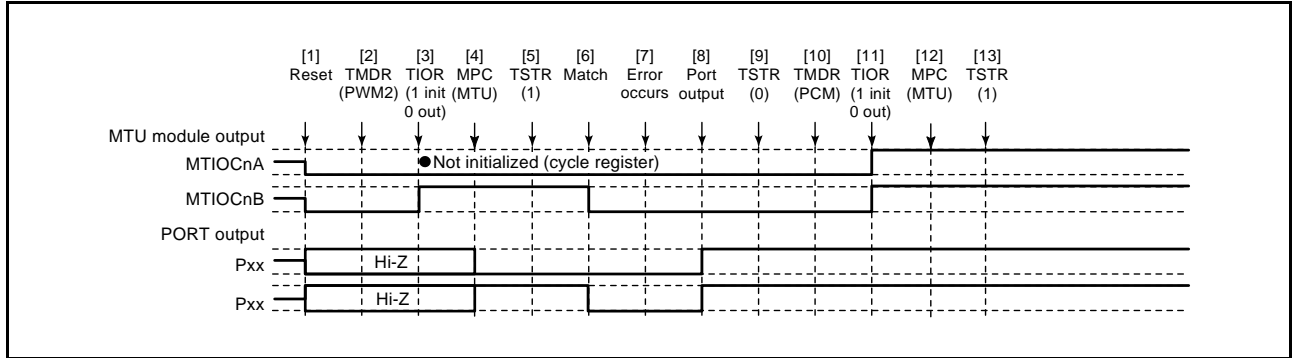


Figure 22.140 Error Occurrence in PWM Mode 2, Recovery in Phase Counting Mode

[1] to [9] are the same as in Figure 22.137.

[10] Set the phase counting mode.

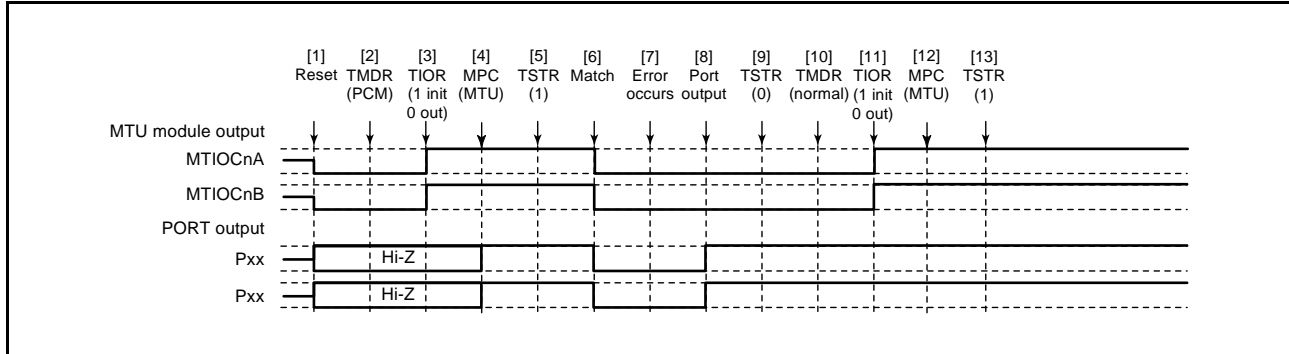
[11] Initialize the pins with TIOR.

[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[13] Restart operation by setting TSTR.

### (17) Operation when Error Occurs in Phase Counting Mode and Operation is Restarted in Normal Mode

Figure 22.141 shows a case in which an error occurs in phase counting mode and operation is restarted in normal mode after re-setting.



**Figure 22.141 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 TIOR. (In the example, the initial output is a high level, and a low level is output on compare match occurrence.)
- [4] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
- [5] Start count operation by setting TSTR.
- [6] Output goes low on compare match occurrence.
- [7] An error occurs.
- [8] Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
- [9] Stop count operation by setting TSTR.
- [10] Set normal mode.
- [11] Initialize the pins with TIOR.
- [12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
- [13] Restart operation by setting TSTR.



(18) Operation when Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 1

Figure 22.142 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 1 after re-setting.

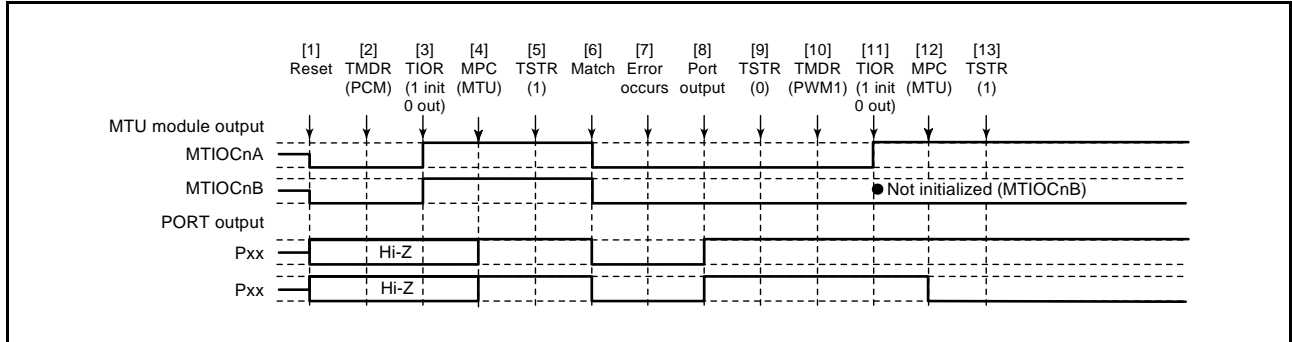


Figure 22.142 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 1

[1] to [9] are the same as in Figure 22.141.

[10] Set PWM mode 1.

[11] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOcNB (MTIOcND) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR)).

[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[13] Restart operation by setting TSTR.

(19) Operation when Error Occurs in Phase Counting Mode and Operation is Restarted in PWM Mode 2

Figure 22.143 shows a case in which an error occurs in phase counting mode and operation is restarted in PWM mode 2 after re-setting.

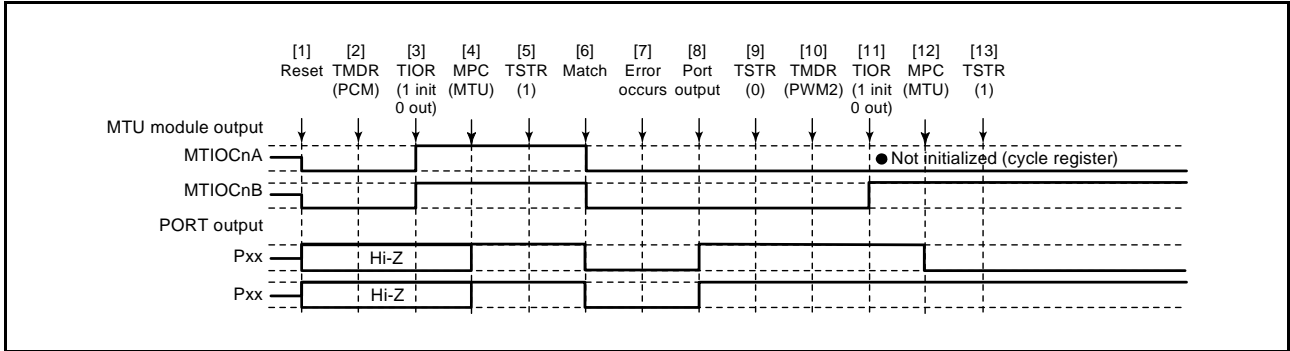


Figure 22.143 Error Occurrence in Phase Counting Mode, Recovery in PWM Mode 2

[1] to [9] are the same as in Figure 22.141.

[10] Set PWM mode 2.

[11] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR)).

[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[13] Restart operation by setting TSTR.

(20) Operation when Error Occurs in Phase Counting Mode and Operation is Restarted in Phase Counting Mode

Figure 22.144 shows a case in which an error occurs in phase counting mode and operation is restarted in phase counting mode after re-setting.

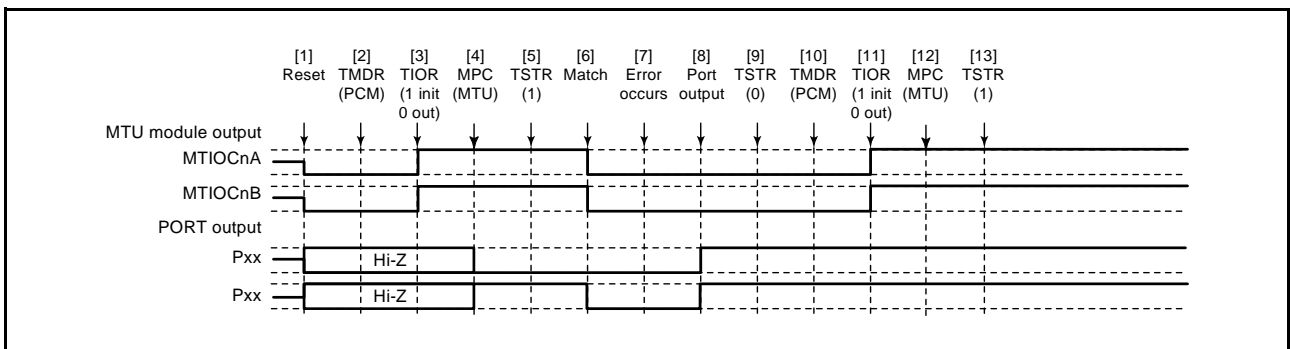


Figure 22.144 Error Occurrence in Phase Counting Mode, Recovery in Phase Counting Mode

[1] to [9] are the same as in Figure 22.141.

[10] This step is not necessary when restarting in phase counting mode.

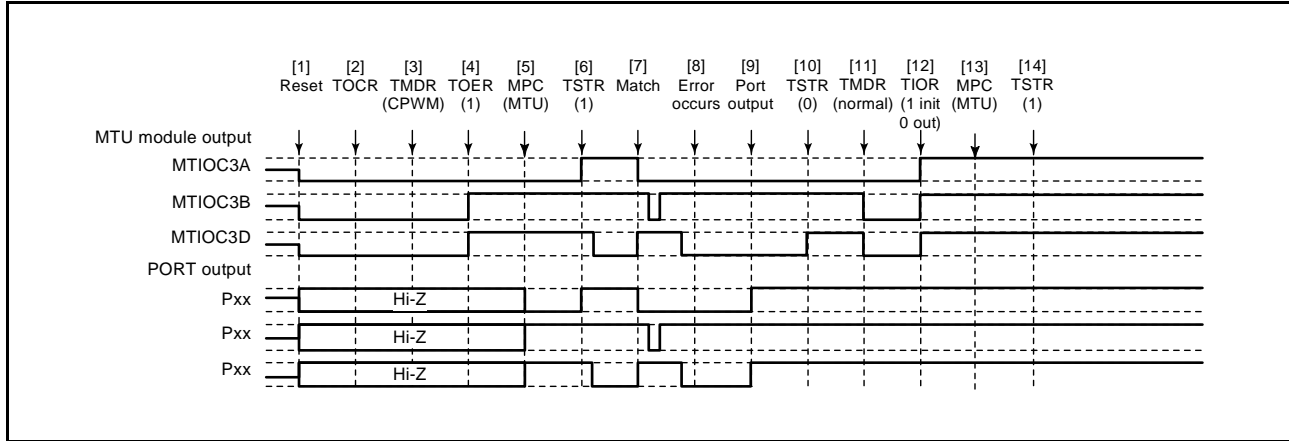
[11] Initialize the pins with TIOR.

[12] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[13] Restart operation by setting TSTR.

### (21) Operation when Error Occurs in Complementary PWM Mode and Operation is Restarted in Normal Mode

Figure 22.145 shows a case in which an error occurs in complementary PWM mode and operation is restarted in normal mode after re-setting.



**Figure 22.145 Error Occurrence in Complementary PWM Mode, Recovery in Normal Mode**

- [1] After a reset, the MTU output goes low and the ports enter high-impedance state.
- [2] Select the complementary PWM output level and enable or disable cyclic output with TOCR.
- [3] Set complementary PWM mode.
- [4] Enable output in MTU3 and MTU4 with TOER.
- [5] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
- [6] Start count operation by setting TSTR.
- [7] The complementary PWM waveform is output on compare match occurrence.
- [8] An error occurs.
- [9] Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
- [10] Stop count operation by setting TSTR. (MTU output becomes the initial complementary PWM output value).
- [11] Set normal mode (MTU output goes low).
- [12] Initialize the pins with TIOR.
- [13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
- [14] Restart operation by setting TSTR.

(22) Operation when Error Occurs in Complementary PWM Mode and Operation is Restarted in PWM Mode 1

Figure 22.146 shows a case in which an error occurs in complementary PWM mode and operation is restarted in PWM mode 1 after re-setting.

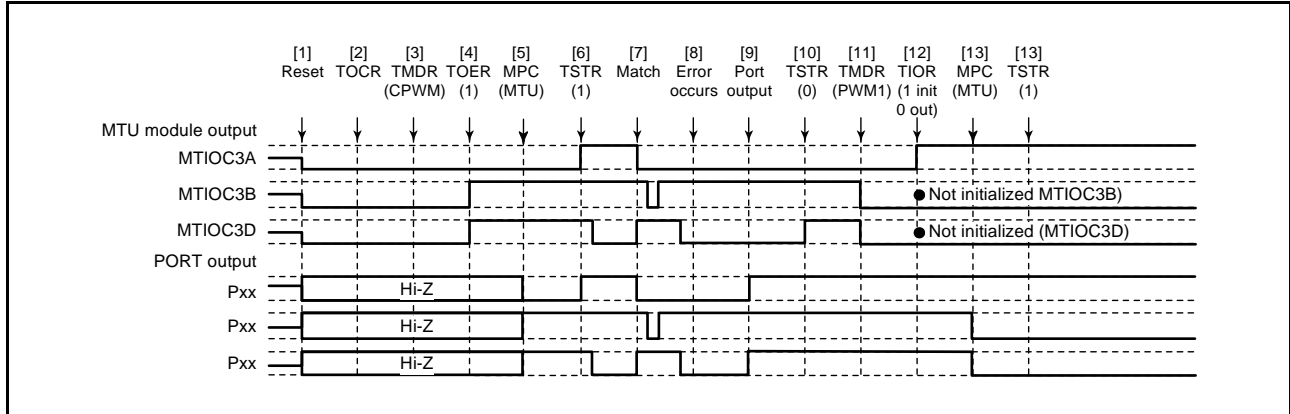


Figure 22.146 Error Occurrence in Complementary PWM Mode, Recovery in PWM Mode 1

[1] to [10] are the same as in Figure 22.145.

[11] Set PWM mode 1 (MTU output goes low).

[12] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR)).

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

(23) Operation when Error Occurs in Complementary PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 22.147 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 cycle and duty settings at the time of stopping the counter).

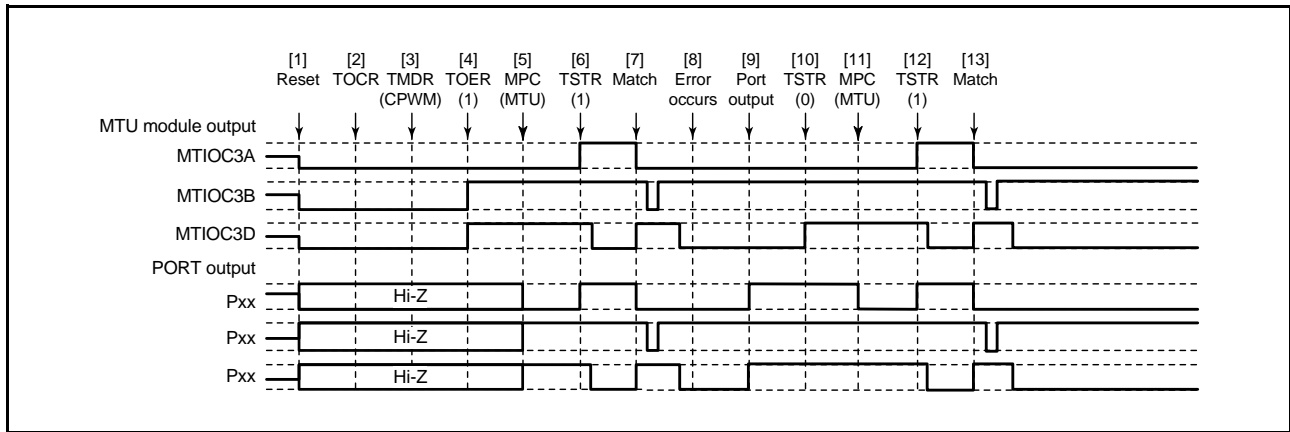


Figure 22.147 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode

[1] to [10] are the same as in Figure 22.145.

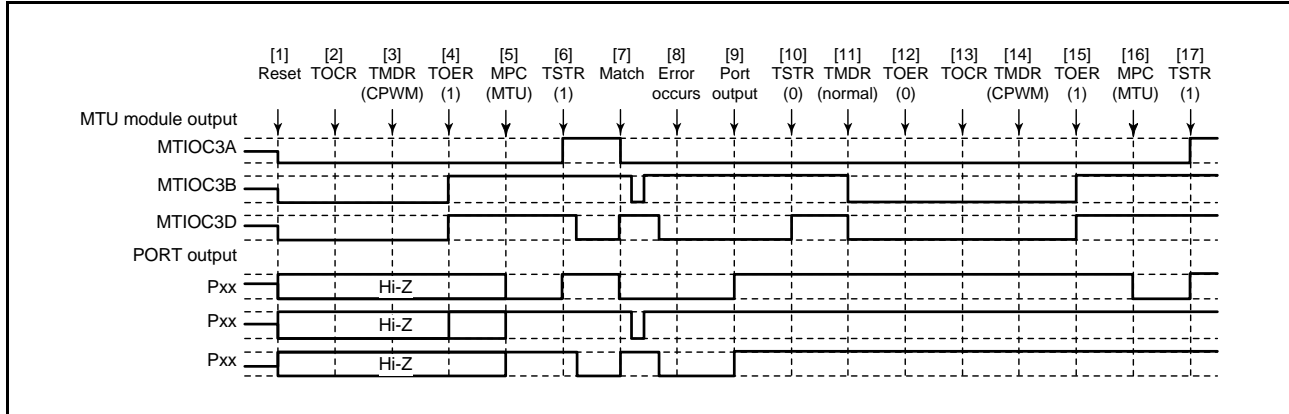
[11] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[12] Restart operation by setting TSTR.

[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 22.148 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 cycle and duty settings).



**Figure 22.148 Error Occurrence in Complementary PWM Mode, Recovery in Complementary PWM Mode**

[1] to [10] are the same as in Figure 22.145.

[11] Set normal mode and make new settings (MTU output goes low).

[12] Disable output in MTU3 and MTU4 with TOER.

[13] Select the complementary PWM output level and enable or disable cyclic output with TOCR.

[14] Set complementary PWM mode.

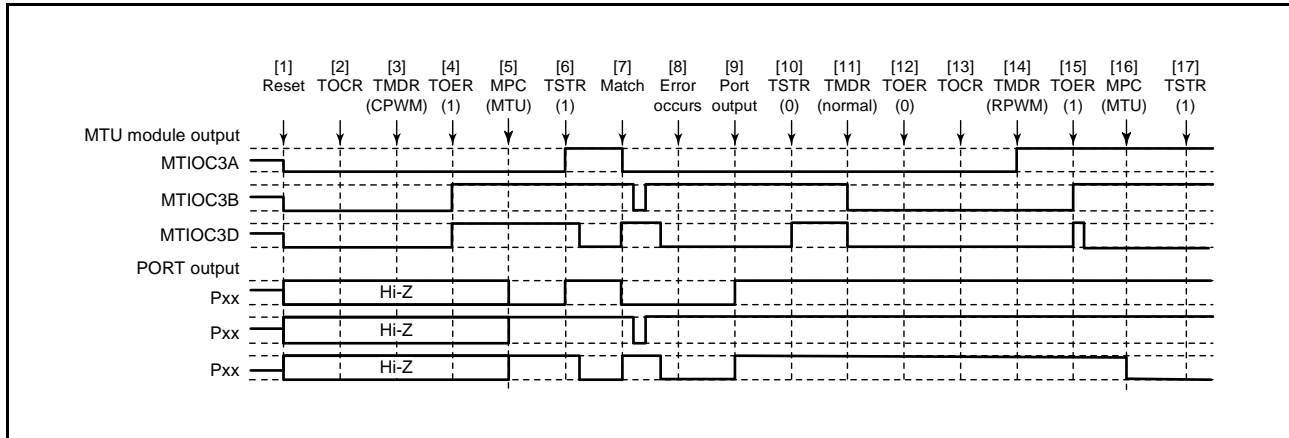
[15] Enable output in MTU3 and MTU4 with TOER.

[16] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[17] Restart operation by setting TSTR.

### (25) Operation when Error Occurs in Complementary PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 22.149 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 22.149 Error Occurrence in Complementary PWM Mode, Recovery in Reset-Synchronized PWM Mode**

[1] to [10] are the same as in Figure 22.145.

[11] Set normal mode (MTU output goes low).

[12] Disable output in MTU3 and MTU4 with TOER.

[13] Select the reset-synchronized PWM output level and enable or disable cyclic output with TOCR.

[14] Set reset-synchronized PWM mode.

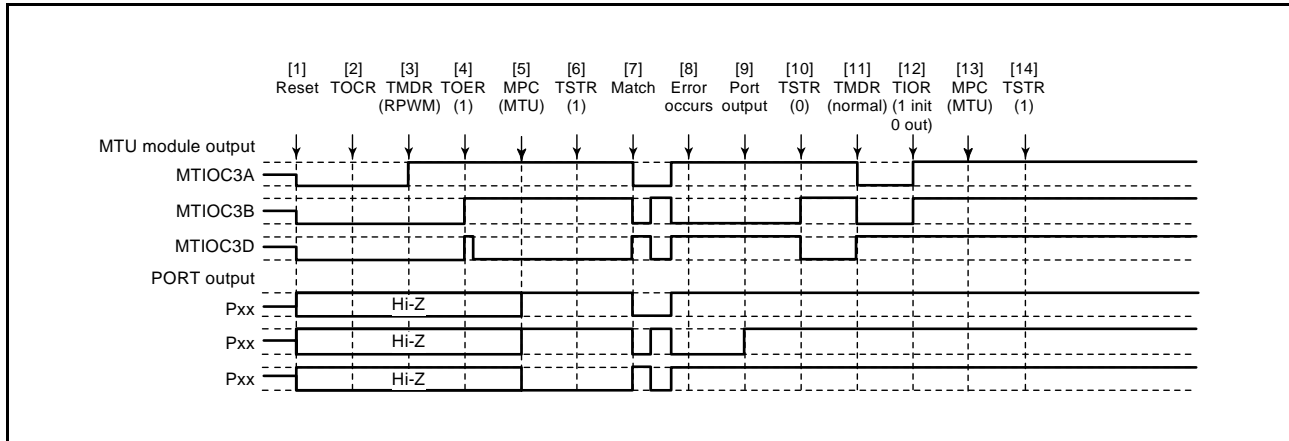
[15] Enable output in MTU3 and MTU4 with TOER.

[16] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[17] Restart operation by setting TSTR.

## (26) Operation when Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Normal Mode

Figure 22.150 shows a case in which an error occurs in reset-synchronized PWM mode and operation is restarted in normal mode after re-setting.



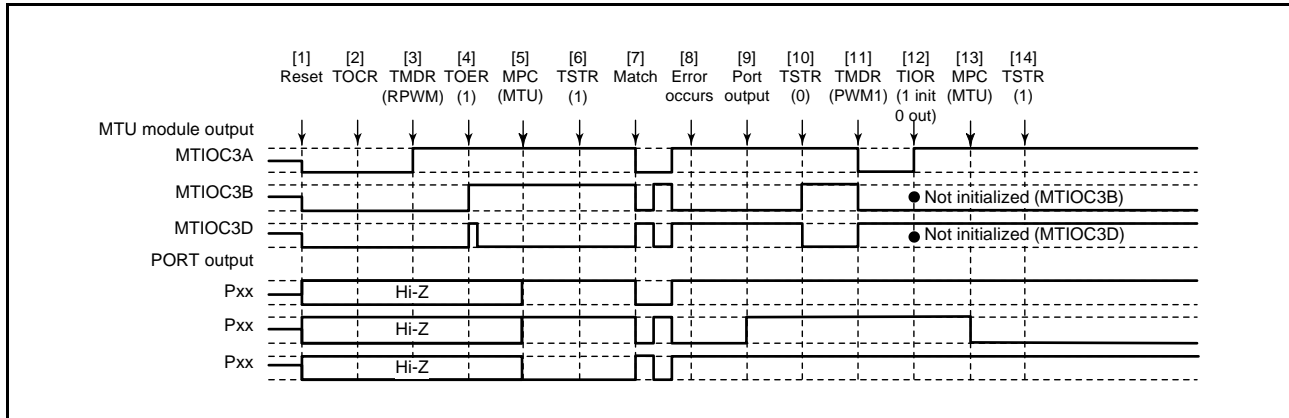
**Figure 22.150 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Normal Mode**

- [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 TOCR.
- [3] Set reset-synchronized PWM mode.
- [4] Enable output in MTU3 and MTU4 with TOER.
- [5] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
- [6] Start count operation by setting TSTR.
- [7] The reset-synchronized PWM waveform is output on compare match occurrence.
- [8] An error occurs.
- [9] Use the port direction register (PDR) and port mode register (PMR) for the input port pin to switch it to operate as a general output port pin, and the port output data register (PODR) to select output of the non-active level.
- [10] Stop count operation by setting TSTR. (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 TIOR.
- [13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.
- [14] Restart operation by setting TSTR.



### (27) Operation when Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in PWM Mode 1

Figure 22.151 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 22.151 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in PWM Mode 1**

[1] to [10] are the same as in Figure 22.150.

[11] Set PWM mode 1 (positive-phase MTU output goes low, and negative-phase output goes high).

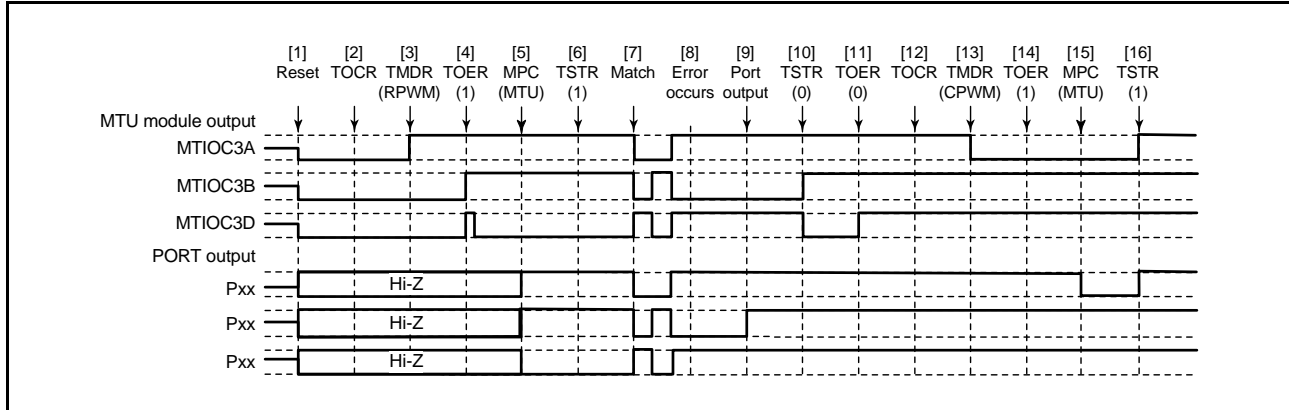
[12] Initialize the pins with the TIOR register. (In PWM mode 1, a waveform is not output on the MTIOCnB (MTIOCnD) pins. If a level is to be output, make the required general output port settings in the I/O port's port direction register (PDR) and port output data register (PODR)).

[13] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[14] Restart operation by setting TSTR.

### (28) Operation when Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Complementary PWM Mode

Figure 22.152 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 22.152 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Complementary PWM Mode**

[1] to [10] are the same as in Figure 22.150.

[11] Disable output in MTU3 and MTU4 with TOER.

[12] Select the complementary PWM output level and enable or disable cyclic output with TOCR.

[13] Set complementary PWM mode (MTU cyclic output pin goes low).

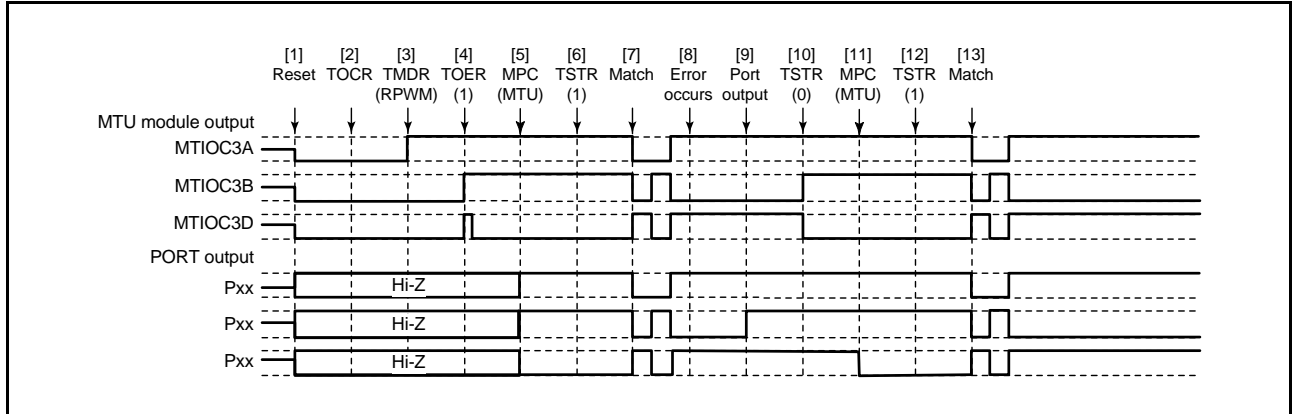
[14] Enable output in MTU3 and MTU4 with TOER.

[15] Use the MPC and the port mode register (PMR) for the I/O port to set up MTU output.

[16] Restart operation by setting TSTR.

(29) Operation when Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode

Figure 22.153 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 22.153 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode**

[1] to [10] are the same as in Figure 22.150.

[11] Make MPC settings and port mode register (PMR) settings for the I/O port pins to operate as MTU outputs.

[12] Use the TSTR for a restart.

[13] The reset-synchronized PWM waveform is output on compare match occurrence.

## 23. Port Output Enable 2 (POE2a)

The port output enable 2 (POE) module can be used to place the states of the pins for complementary PWM output by the MTU (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D), and the states of pins for MTU0 (MTIOC0A, MTIOC0B, MTIOC0C, and MTIOC0D) in the high-impedance in response to changes in the input levels on the POE0# to POE3# and POE8# pins, in the output levels on pins for complementary PWM output by the MTU, oscillation stop detection by the clock generation circuit, and changes to register settings (SPOER).

It can also generate simultaneous interrupt requests.

### 23.1 Overview

Table 23.1 lists the specifications of the POE, and Figure 23.1 shows a block diagram of the POE.

**Table 23.1 POE Specifications**

Item	Description
High-impedance is controlled by the input level detection	<ul style="list-style-type: none"> <li>Falling-edge detection or sampling of the low level 16 times at PCLK/8, PCLK/16, or PCLK/128 can be set for each of the POE0# to POE3# and POE8# input pins.</li> <li>Pins for complementary PWM output from the MTU can be placed in the high-impedance on detection of falling edges or sampling of the low level on the POE0# to POE3# pins.</li> <li>Pins for output from MTU0 can be placed in the high-impedance on detection of falling edges or sampling of the low level on the POE8# pin.</li> </ul>
High-impedance is controlled by the output level comparison	<ul style="list-style-type: none"> <li>Levels output on pins for complementary PWM output from the MTU are compared, and when simultaneous output of the active level continues for one or more cycles, the pins can be placed in the high-impedance.</li> </ul>
High-impedance is controlled by the oscillation stop detection	<ul style="list-style-type: none"> <li>Pins for complementary PWM output from the MTU and output pins for MTU0 can be placed in the high-impedance when oscillation by the clock generation circuit stops.</li> </ul>
High-impedance is controlled by software (registers)	<ul style="list-style-type: none"> <li>Pins for complementary PWM output from the MTU and output pins for MTU0 can be placed in the high-impedance by modifying settings of POE registers.</li> </ul>
Interrupts	<ul style="list-style-type: none"> <li>Interrupts are generated in response to the results of POE0# to POE3# and POE8# input-level detection and MTU complementary PWM output-level comparison.</li> </ul>

The POE has input-level detection circuits, output-level comparison circuits, an input for the oscillation-stopped detection signal from the clock generation circuit, and a high-impedance request/interrupt request generating circuit as shown in Figure 23.1.

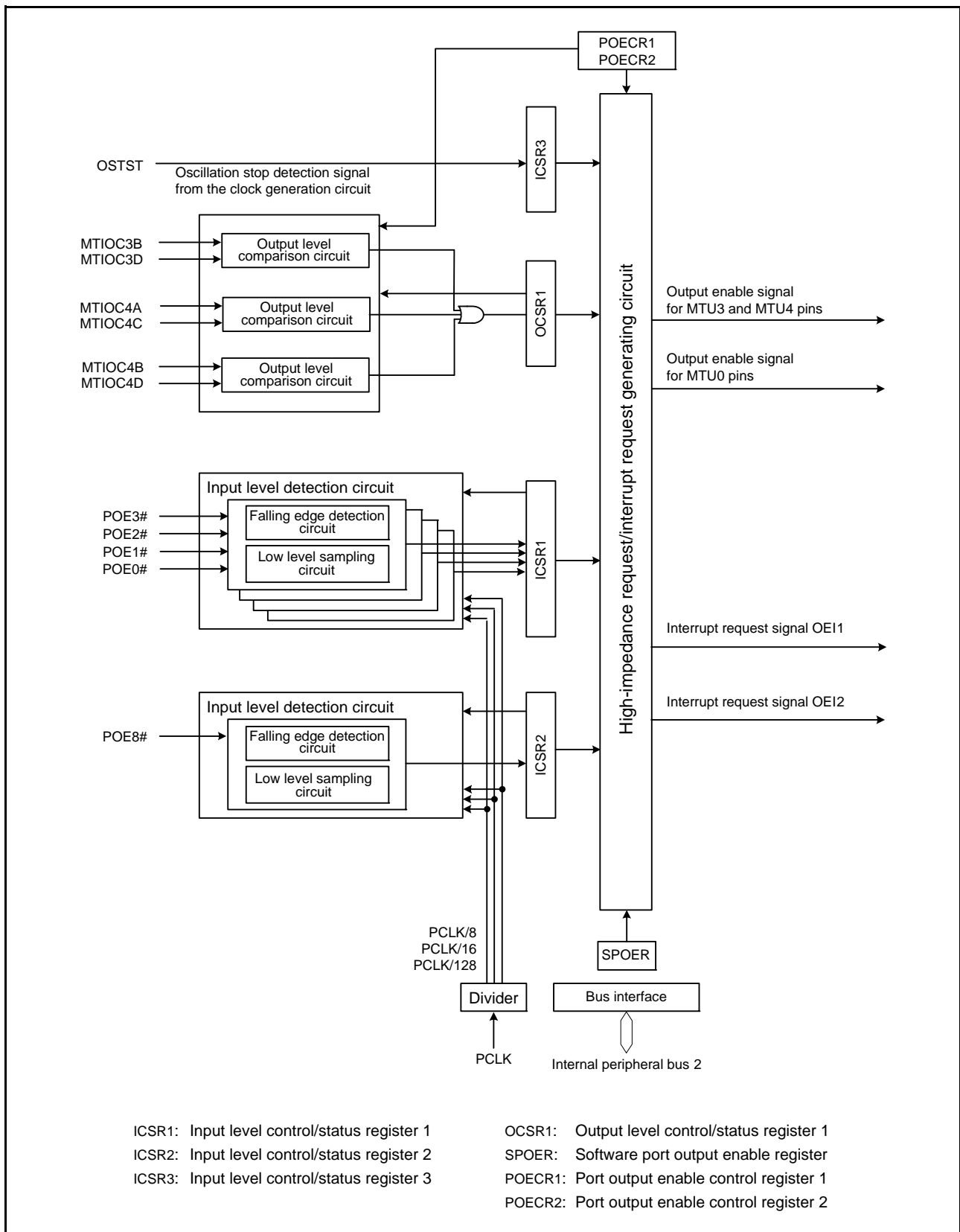


Figure 23.1 POE Block Diagram

Table 23.2 lists input/output pins to be used by the POE.

**Table 23.2 LSI Input/Output Pins to be Used by POE**

Pin Name	I/O	Description
POE0# to POE3#	Input	Request signals to place the pins for MTU complementary PWM output in high-impedance.
POE8#	Input	Request signals to place the MTU0 pins in high-impedance.
MTIOC3B	Output	MTU3 complementary PWM output pin
MTIOC3D	Output	MTU3 complementary PWM output pin
MTIOC4A	Output	MTU4 complementary PWM output pin
MTIOC4B	Output	MTU4 complementary PWM output pin
MTIOC4C	Output	MTU4 complementary PWM output pin
MTIOC4D	Output	MTU4 complementary PWM output pin
MTIOC0A	Output	MTU0 output pin
MTIOC0B	Output	MTU0 output pin
MTIOC0C	Output	MTU0 output pin
MTIOC0D	Output	MTU0 output pin

Table 23.3 lists output-level comparisons with pin combinations.

**Table 23.3 Pin Combinations**

Pin Combination	I/O	Description
MTIOC3B and MTIOC3D	Output	Pin combinations for output-level comparison and high-impedance control can be selected by POE registers.
MTIOC4A and MTIOC4C	Output	The pins for MTU complementary PWM output are placed in high-impedance when the pins simultaneously output an active level for one or more cycles of PCLK. (When the MTU. TOCR1.TOCS bit = 0: Low level if the MTU.TOCR1.OLSP and OSLN bits are 0, and high level if the MTU.TOCR1.OLSP and OSLN bits are 1. When the MTU. TOCR1.TOCS bit = 1: Low level if the MTU.TOCR2.OLS3N, OLS3P, OLS2N, OLS2P, OLS1N and OLS1P bits are 0, and high level if the MTU.TOCR2.OLS3N, OLS3P, OLS2N, OLS2P, OLS1N and OLS1P bits are 1.)
MTIOC4B and MTIOC4D	Output	

## 23.2 Register Descriptions

### 23.2.1 Input Level Control/Status Register 1 (ICSR1)

Address(es): 0008 8900h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
POE3F	POE2F	POE1F	POE0F	—	—	—	PIE1	POE3M[1:0]	POE2M[1:0]	POE1M[1:0]	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# input. 0 1: Accepts a request when POE0# input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE0# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE0# input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b3, b2	POE1M[1:0]	POE1 Mode Select	b3 b2 0 0: Accepts a request on the falling edge of POE1# input. 0 1: Accepts a request when POE1# input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE1# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE1# input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b5, b4	POE2M[1:0]	POE2 Mode Select	b5 b4 0 0: Accepts a request on the falling edge of POE2# input. 0 1: Accepts a request when POE2# input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE2# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE2# input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b7, b6	POE3M[1:0]	POE3 Mode Select	b7 b6 0 0: Accepts a request on the falling edge of POE3# input. 0 1: Accepts a request when POE3# input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE3# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE3# input has been sampled 16 times at PCLK/128 clock pulses and all are low level.	R/W*1
b8	PIE1	Port Interrupt Enable 1	0: OEI1 interrupt requests by the input level detection disabled 1: OEI1 interrupt requests by the input level detection 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
b13	POE1F	POE1 Flag	0: Indicates that a high impedance request has not been input to the POE1# pin. 1: Indicates that a high impedance request has been input to the POE1# pin.	R/(W) *2
b14	POE2F	POE2 Flag	0: Indicates that a high impedance request has not been input to the POE2# pin. 1: Indicates that a high impedance request has been input to the POE2# pin.	R/(W) *2
b15	POE3F	POE3 Flag	0: Indicates that a high impedance request has not been input to the POE3# pin. 1: Indicates that a high impedance request has been input to the POE3# pin.	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.

When low-level sampling has been set by the POE0M[1:0] to POE3M[1:0] bits, writing 0 to the POE0F to POE3F flags requires high level input on the POE0# to POE3# pins.

For details, see section 23.3.5, Release from the High-Impedance.

### **PIE1 Bit (Port Interrupt Enable 1)**

This bit enables or disables OEI1 interrupt requests when any one of the POE0F to POE3F flags is set to 1.

### **POE0F Flag (POE0 Flag)**

This flag indicates that a high impedance request has been input to the POE0# pin.

[Clearing condition]

- By writing 0 to POE0F after reading POE0F = 1

[Setting condition]

- When the input set by POE0M[1:0] occurs at the POE0# pin

### **POE1F Flag (POE1 Flag)**

This flag indicates that a high impedance request has been input to the POE1# pin.

[Clearing condition]

- By writing 0 to POE1F after reading POE1F = 1

[Setting condition]

- When the input set by POE1M[1:0] occurs at the POE1# pin

### **POE2F Flag (POE2 Flag)**

This flag indicates that a high impedance request has been input to the POE2# pin.

[Clearing condition]

- By writing 0 to POE2F after reading POE2F = 1

[Setting condition]

- When the input set by POE2M[1:0] occurs at the POE2# pin

### **POE3F Flag (POE3 Flag)**

This flag indicates that a high impedance request has been input to the POE3# pin.

[Clearing condition]

- By writing 0 to POE3F after reading POE3F = 1

[Setting condition]

- When the input set by POE3M[1:0] occurs at the POE3# pin



### 23.2.2 Output Level Control/Status Register 1 (OCSR1)

Address(es): 0008 8902h

	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	Output Short Interrupt Enable 1	0: OEI1 interrupt requests by the output level comparison disabled 1: OEI1 interrupt requests by the output level comparison enabled	R/W
b9	OCE1	Output Short High-Impedance Enable 1	0: Does not place the pins in high-impedance. 1: Places the pins in high-impedance.	R/W*1
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	OSF1	Output Short 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.

#### OIE1 Bit (Output Short Interrupt Enable 1)

This bit enables or disables OEI1 interrupt requests when the OSF1 flag is set to 1.

#### OCE1 Bit (Output Short High-Impedance Enable 1)

This bit specifies whether to place the MTU complementary PWM output pins in high-impedance when the OSF1 flag is set to 1.

#### OSF1 Flag (Output Short Flag 1)

This flag indicates that any one of the three pairs of two-phase outputs for MTU complementary PWM output to be compared in Table 23.3 has simultaneously become an active level. If the PnCZEA (n = 1, 2, 3) bits in POE2CR2 are 0 or the output comparison function of the MTU is not enabled, the OSF1 flag will not be set to 1 even if both pins in the corresponding complementary output pair of the MTU are simultaneously active.

[Clearing condition]

- By writing 0 to OSF1 after reading OSF1 = 1

The complementary output pins for the MTU must be at the inactive level when 0 is written to the flag. For details, see section 23.3.5, Release from the High-Impedance.

[Setting condition]

- When any one of the three pairs of two-phase outputs has simultaneously become an active level

### 23.2.3 Input Level Control/Status Register 2 (ICSR2)

Address(es): 0008 8908h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	POE8F	—	—	POE8E	PIE2	—	—	—	—	—	—	POE8M[1:0]	—
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Value after reset:

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# input 0 1: Accepts a request when POE8# input has been sampled 16 times at PCLK/8 clock pulses and all are low level. 1 0: Accepts a request when POE8# input has been sampled 16 times at PCLK/16 clock pulses and all are low level. 1 1: Accepts a request when POE8# 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	PIE2	Port Interrupt Enable 2	0: OEI2 interrupt requests disabled 1: OEI2 interrupt requests enabled	R/W
b9	POE8E	POE8 High-Impedance Enable	0: Does not place the MTIOC0A, MTIOC0B, MTIOC0C, and MTIOC0D pins in high-impedance. 1: Places the MTIOC0A, MTIOC0B, MTIOC0C, and MTIOC0D pins in high-impedance.	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.

#### PIE2 Bit (Port Interrupt Enable 2)

This bit enables or disables OEI2 interrupt requests when the POE8F flag is set to 1.

#### POE8E Bit (POE8 High-Impedance Enable)

This bit specifies whether to place the MTU0 pins in high-impedance 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.

[Clearing condition]

Writing 0 to POE8F after reading POE8F = 1

When writing 0 to the flag while low-level sampling is selected for the POE8M[1:0] bits, the POE8# pin input must be at the high level.

For details, see section 23.3.5, Release from the High-Impedance.

[Setting condition]

- When the input set by POE8M[1:0] occurs at the POE8# pin

### 23.2.4 Software Port Output Enable Register (SPOER)

Address(es): 0008 890Ah

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	CH0HI Z	CH34HI Z
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	CH34HIZ	MTU3 and MTU4 Output High-Impedance Enable	0: Does not place the pins in high-impedance. 1: Places the pins in high-impedance.	R/W
b1	CH0HIZ	MTU0 Output High-Impedance Enable	0: Does not place the pins in high-impedance. 1: Places the pins in high-impedance.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

#### CH34HIZ Bit (MTU3 and MTU4 Output High-Impedance Enable)

This bit specifies whether to place the MTU complementary PWM output pins (MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, MTIOC4D) in high-impedance.

[Clearing condition]

- By writing 0 to CH34HIZ after reading CH34HIZ = 1

[Setting condition]

- By writing 1 to CH34HIZ

#### CH0HIZ Bit (MTU0 Output High-Impedance Enable)

This bit specifies whether to place the MTU0 pins (MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D) in high-impedance.

[Clearing condition]

- By writing 0 to CH0HIZ after reading CH0HIZ = 1

[Setting condition]

- By writing 1 to CH0HIZ

### 23.2.5 Port Output Enable Control Register 1 (POECR1)

Address(es): 0008 890Bh

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	PE3ZE	PE2ZE	PE1ZE	PE0ZE

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	PE0ZE	MTIOC0A High-Impedance Enable	0: Does not place the pin in high-impedance. 1: Places the pin in high-impedance.	R/W*1
b1	PE1ZE	MTIOC0B High-Impedance Enable	0: Does not place the pin in high-impedance. 1: Places the pin in high-impedance.	R/W*1
b2	PE2ZE	MTIOC0C High-Impedance Enable	0: Does not place the pin in high-impedance. 1: Places the pin in high-impedance.	R/W*1
b3	PE3ZE	MTIOC0D High-Impedance Enable	0: Does not place the pin in high-impedance. 1: Places the pin in high-impedance.	R/W*1
b7 to b4	—	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.

### 23.2.6 Port Output Enable Control Register 2 (POECR2)

Address(es): 0008 890Ch

b7	b6	b5	b4	b3	b2	b1	b0
—	P1CZEA	P2CZEA	P3CZEA	—	—	—	—

Value after reset: 0 1 1 1 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	P3CZEA	MTU Port 3 High-Impedance Enable	0: Comparison of output levels does not proceed and the pins are not placed in the high-impedance. 1: The pins are placed in the high-impedance.	R/W*1
b5	P2CZEA	MTU Port 2 High-Impedance Enable	0: Comparison of output levels does not proceed and the pins are not placed in the high-impedance. 1: The pins are placed in the high-impedance.	R/W*1
b6	P1CZEA	MTU Port 1 High-Impedance Enable	0: Comparison of output levels does not proceed and the pins are not placed in the high-impedance. 1: The pins are placed in the high-impedance.	R/W*1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Can be modified only once after a reset.

When this function is not used, write 00h to this register.

#### P3CZEA Bit (MTU Port 3 High-Impedance Enable)

This bit gives permission regarding whether or not the MTIOC4B and MTIOC4D pins for complementary PWM output from the MTU are placed in the high-impedance. It also gives permission regarding whether or not the levels on the MTIOC4B and MTIOC4D pins are compared.

**P2CZEA Bit (MTU Port 2 High-Impedance Enable)**

This bit gives permission regarding whether or not the MTIOC4A and MTIOC4C pins for complementary PWM output from the MTU are placed in the high-impedance. It also gives permission regarding whether or not the levels on the MTIOC4A and MTIOC4C pins are compared.

**P1CZEA Bit (MTU Port 1 High-Impedance Enable)**

This bit gives permission regarding whether or not the MTIOC3B and MTIOC3D pins for complementary PWM output from the MTU are placed in the high-impedance. It also gives permission regarding whether or not the levels on the MTIOC3B and MTIOC3D pins are compared.

**23.2.7 Input Level Control/Status Register 3 (ICSR3)**

Address(es): 0008 890Eh

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	OSTST F	—	—	OSTST E	—	—	—	—	—	—	—	—	—
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Value after reset:

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	OSTST High-Impedance Enable	0: Does not place the MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D pins in high-impedance. 1: Places the MTIOC0A, MTIOC0B, MTIOC0C, MTIOC0D, MTIOC3B, MTIOC3D, MTIOC4A, MTIOC4B, MTIOC4C, and MTIOC4D pins in high-impedance.	R/W*1
b11, b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	OSTSTF	OSTST High-Impedance Flag	0: Oscillation stop is not producing a request to place pins in the high-impedance. 1: Oscillation stop is producing a request to place pins in the high-impedance.	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.

**OSTSTE Bit (OSTST High-Impedance Enable)**

This bit permits or prohibits placement of pins for complementary PWM output from MTU and output pins for MTU0 in the high-impedance on detection that oscillation has stopped.

**OSTSTF Flag (OSTST High-Impedance Flag)**

The OSTSTF flag is a status flag that indicates the state of requests to place pins in the high-impedance due to oscillation having stopped. The value of the flag becomes 1 when oscillation stops. Ensure that the oscillation-stopped detection signal is negated when clearing the flag by writing 0 to it. Writing 0 to the OSTSTF flag while the oscillation-stopped detection signal is being asserted will not clear the flag. The assertion of the oscillation-stopped detection signal takes 10 cycles of PCLK after detecting the oscillation stop.

[Clearing condition]

- Writing 0 to the bit after having read its value as 1.

[Setting condition]

- Detection of the oscillation-stopped state

### 23.3 Operation

The target pins for high-impedance control and conditions to place the pins in high-impedance are described below.

#### (1) MTU0 pin (MTIOC0A)

When any of the following conditions is satisfied, the pin is placed to the high-impedance state.

- POE8# input level detection  
When ICSR2.POE8F flag is set to 1 with POECR1.PE0ZE and ICSR2.POE8E set to 1.
- SPOER setting  
When SPOER.CH0HIZ bit is set to 1 with POECR1.PE0ZE set to 1.
- Detection of stopped oscillation  
When OSTSTF flag is set to 1 with POECR1.PE0ZE and ICSR3.OSTSTE set to 1.

#### (2) MTU0 pin (MTIOC0B)

When any of the following conditions is satisfied, the pin is placed to the high-impedance state.

- POE8# input level detection  
When ICSR2.POE8F flag is set to 1 with POECR1.PE1ZE and ICSR2.POE8E set to 1.
- SPOER setting  
When SPOER.CH0HIZ bit is set to 1 with POECR1.PE1ZE set to 1.
- Detection of stopped oscillation  
When OSTSTF flag is set to 1 with POECR1.PE1ZE and ICSR3.OSTSTE set to 1.

#### (3) MTU0 pin (MTIOC0C)

When any of the following conditions is satisfied, the pin is placed to the high-impedance state.

- POE8# input level detection  
When ICSR2.POE8F flag is set to 1 with POECR1.PE2ZE and ICSR2.POE8E set to 1.
- SPOER setting  
When SPOER.CH0HIZ bit is set to 1 with POECR1.PE2ZE set to 1.
- Detection of stopped oscillation  
When OSTSTF flag is set to 1 with POECR1.PE2ZE and ICSR3.OSTSTE set to 1.

#### (4) MTU0 pin (MTIOC0D)

When any of the following conditions is satisfied, the pin is placed to the high-impedance state.

- POE8# input level detection  
When ICSR2.POE8F flag is set to 1 with POECR1.PE3ZE and ICSR2.POE8E set to 1.
- SPOER setting  
When SPOER.CH0HIZ bit is set to 1 with POECR1.PE3ZE set to 1.
- Detection of stopped oscillation  
When OSTSTF flag is set to 1 with POECR1.PE3ZE and ICSR3.OSTSTE set to 1.

#### (5) MTU3 pins (MTIOC3B and MTIOC3D)

When any of the following conditions is satisfied, the pins are placed to the high-impedance state.

- POE0# to POE3# input level detection  
When ICSR1.POE3F, POE2F, POE1F, or POE0F flag is set to 1 with POECR2.P1CZEA set to 1.
- MTIOC3B and MTIOC3D output level comparison  
When OCSR1.OSF1 flag is set to 1 with POECR2.P1CZEA and OCSR1.OCE1 set to 1.
- SPOER setting  
When SPOER.CH34HIZ bit is set to 1 with POECR2.P1CZEA set to 1.

- Detection of stopped oscillation  
When ICSR3.OSTSTF flag is set to 1 with POE2R.P1CZEA and ICSR3.OSTSTE set to 1.

(6) MTU4 pins (MTIOC4A and MTIOC4C)

When any of the following conditions is satisfied, the pins are placed to the high-impedance state.

- POE0# to POE3# input level detection  
When ICSR1.POE3F, POE2F, POE1F, or POE0F flag is set to 1 with POE2R.P2CZEA set to 1.
- MTIOC4A and MTIOC4C output level comparison  
When OCSR1.OSF1 flag is set to 1 with POE2R.P2CZEA and OCSR1.OCE1 set to 1.
- SPOER setting  
When SPOER.CH34HIZ bit is set to 1 with POE2R.P2CZEA set to 1.
- Detection of stopped oscillation  
When ICSR3.OSTSTF flag is set to 1 with POE2R.P2CZEA and ICSR3.OSTSTE set to 1.

(7) MTU4 pins (MTIOC4B and MTIOC4D)

When any of the following conditions is satisfied, the pins are placed to the high-impedance state.

- POE0# to POE3# input level detection  
When ICSR1.POE3F, POE2F, POE1F, or POE0F flag is set to 1 with POE2R.P3CZEA set to 1.
- MTIOC4B and MTIOC4D output level comparison  
When OCSR1.OSF1 flag is set to 1 with POE2R.P3CZEA and OCSR1.OCE1 set to 1.
- SPOER setting  
When SPOER.CH34HIZ bit is set to 1 with POE2R.P3CZEA set to 1.
- Detection of stopped oscillation  
When ICSR3.OSTSTF flag is set to 1 with POE2R.P3CZEA and ICSR3.OSTSTE set to 1.

### 23.3.1 Input Level Detection Operation

If the input conditions set by ICSR1 and ICSR2 occur on the POE0# to POE3# and POE8# pins, the pins for the MTU complementary PWM output and MTU0 are placed in high-impedance.

#### (1) Falling Edge Detection

When a change from a high to low level is input to the POE0# to POE3# and POE8# pins, the pins for the MTU complementary PWM output and MTU0 are placed in high-impedance.

A falling edge is detected after PCLK causes sampling to proceed. If the low level is input to the POE0# to POE3# or POE8# pin over less than one full cycle of PCLK, whether the falling edge will or will not be detected cannot be guaranteed.

Figure 23.2 shows the timing of sampling after the level changes in input to the POE0# to POE3# and POE8# pins until the respective pins enter high-impedance.

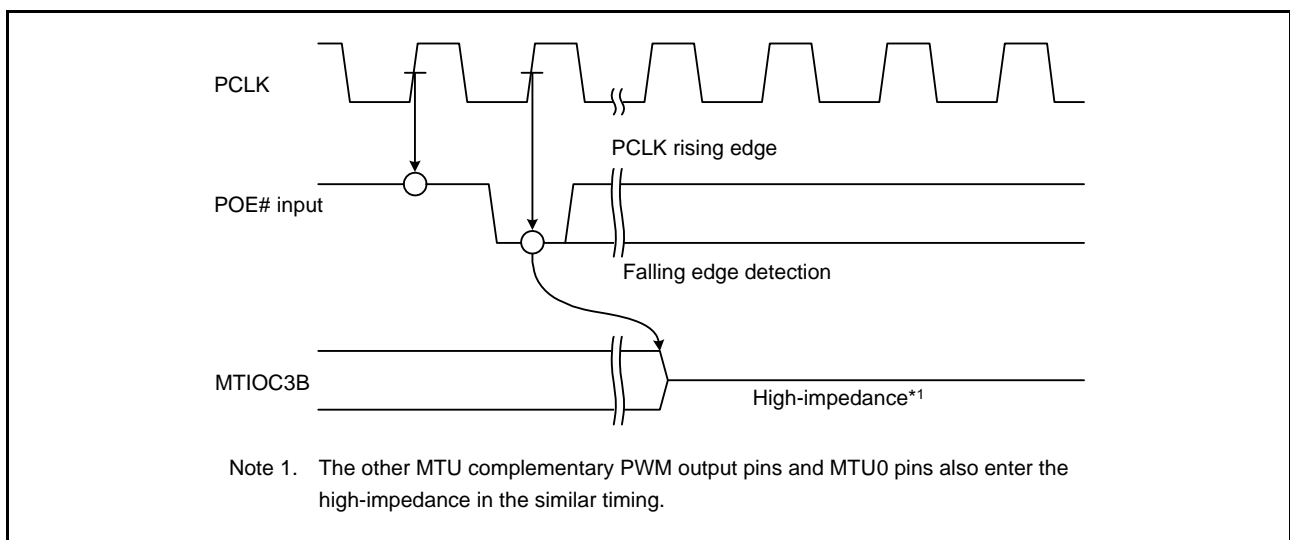


Figure 23.2 Falling Edge Detection

#### (2) Low-Level Detection

Figure 23.3 shows the low-level detection operation. Sixteen continuous low levels are sampled with the sampling clock selected by ICSR1 and ICSR2. If even one high level is detected during this interval, the low level is not accepted.

Furthermore, in an interval over which the sampling clock is not being output, changes to the levels on the POE0# to POE3# and POE8# pins are ignored.

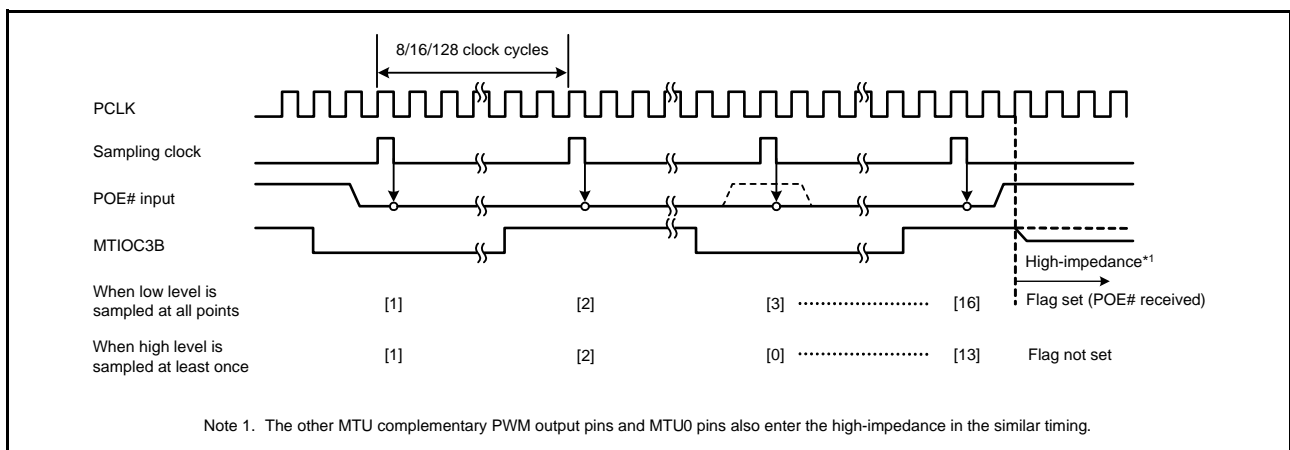
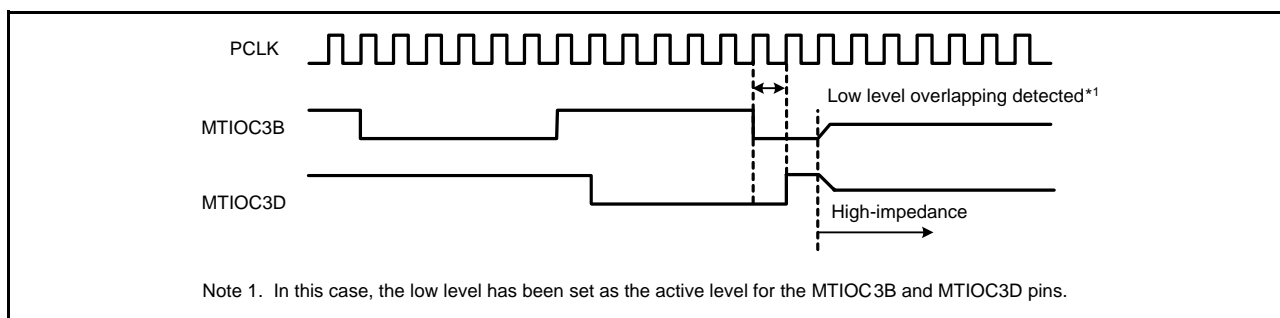


Figure 23.3 Low-Level Detection Operation



### 23.3.2 Output-Level Compare Operation

Figure 23.4 shows an example of the output-level compare operation for the combination of MTIOC3B and MTIOC3D (MTU complementary PWM output pins). The operation is the same for the other pin combinations.



**Figure 23.4** Output-Level Compare Operation

### 23.3.3 High-Impedance Control Using Registers

The high-impedance of the MTU complementary PWM output and MTU0 pins can be directly controlled using the software port output enable register (SPOER).

Setting the CH34HIZ bit in SPOER to 1 places the MTU complementary PWM output pins (MTU3 and MTU4) specified by the port output enable control register 2 (POE2) in the high-impedance.

The high-impedance of other pins can also be controlled by setting the appropriate bits in SPOER.

### 23.3.4 High-Impedance Control on Detection of Stopped Oscillation

When the oscillation-stop detection circuit in the clock pulse generator detects stopped oscillation, pins for complementary PWM output from MTU and pins for MTU0 as specified by the input level control and status register 3 (ICSR3) can be placed in the high-impedance.

### 23.3.5 Release from the High-Impedance

Pins for complementary PWM output from MTU and pins for MTU0 which have been placed in the high-impedance due to input-level detection can be released from that state by either returning them to their initial state with a reset or clearing all of the POE3F to POE0F flags in ICSR1 and the POE8F flag in ICSR2. Note, however, that when low-level sampling is selected by the POE3M[1:0] to POE0M[1:0] bits in ICSR1 and the POE8M[1:0] bits in ICSR2, if a high level is being input to the corresponding pin from among POE0# to POE3# and POE#8 but has not yet been sampled, writing 0 to the flag is ignored (the flag is not cleared).

MTU complementary PWM output pins which have been placed in the high-impedance due to output-level comparison can be released from that state by either returning them to their initial state with a reset or clearing the OSF1 flag in OCSR1. Note, however, that if the inactive level is not yet being output from the MTU complementary PWM output pins, writing 0 to the flag is ignored (the flag is not cleared). Inactive-level outputs can be obtained by setting the MTU registers.

For MTU complementary PWM output pins and pins for MTU0 that have been placed in the high-impedance because oscillation by the clock generation circuit has stopped, clearing the OSTSTF or OSTSTE bit in ICSR3 releases the pins from the high-impedance.

For MTU complementary PWM output pins and pins for MTU0 that have been placed in the high-impedance by the SPOER.CH34HIZ or SPOER.CH0HIZ bit, clearing the corresponding bits (SPOER.CH34HIZ and SPOER.CH0HIZ) releases the pins from the high-impedance.

## 23.4 Interrupts

The POE issues a request to generate an interrupt when the corresponding condition below is matched during input-level detection, output-level comparison, or oscillation stop by the clock generation circuit. Table 23.4 shows the interrupt sources and their request conditions.

On acceptance of an OEI1 or OEI2 interrupt, the first line of the exception handling routine for the given interrupt should confirm that the flag for the given flag has been set to 1.

**Table 23.4 Interrupt Sources and Conditions**

Name	Interrupt Source	Interrupt Flag	Condition
OEI1	Output enable interrupt 1	POE0F, POE1F, POE2F, POE3F, OSF1	When ICSR1.POE0F, POE1F, POE2F, or POE3F flag is set to 1 with ICSR1.PIE1 set to 1, or when OCSR1.OSF1 flag is set to 1 with OCSR1.OIE1 set to 1.
OEI2	Output enable interrupt 2	POE8F	When ICSR2.POE8F flag is set to 1 with ICSR2.PIE2 set to 1.

## 23.5 Usage Notes

### 23.5.1 Transitions to Software Standby Mode or Deep Standby Mode

When the POE is used, do not make a transition to software standby mode or deep software standby mode. In these modes, the POE stops and thus the high-impedance of pins cannot be controlled.

### 23.5.2 When POE is not Used

When the POE is not used, write 00h to the port output enable control registers 1 and 2 (POECR1 and POECR2), respectively.

### 23.5.3 Specifying Pins corresponding to the MTU

The POE controls high-impedance outputs only when a pin has been specified so that the pin corresponds to the MTU by setting the port mode register (PMR). When the pin has been specified as a general I/O pin, the POE does not control high-impedance outputs.

## 24. 16-Bit Timer Pulse Unit (TPUa)

The RX630 Group has two on-chip 16-bit timer pulse units (TPU), unit 0 and unit 1, each comprising six channels. Therefore, this LSI includes twelve channels (TPU0 to TPU11).

### 24.1 Overview

Specifications of the TPU are listed in Table 24.1. Functions of TPU (unit 0) and TPU (unit 1) are listed in Table 24.2 and Table 24.3, respectively.

Block diagrams of TPU (unit 0) and TPU (unit 1) are shown in Figure 24.1 and Figure 24.2, respectively.

**Table 24.1 Specifications of TPU**

Item	Description
Pulse input/output	Maximum 32 (for unit 0: 16, for unit 1: 16)
Count clock	Seven or eight types are provided for each channel.
Settable operations	<ul style="list-style-type: none"> <li>• Waveform output at compare match</li> <li>• Input capture function (noise filters can be set)</li> <li>• Counter clear operation</li> <li>• Simultaneous writing to multiple timer counters (TCNT)</li> <li>• Simultaneous clearing by compare match and input capture</li> <li>• Synchronous input/output for registers by counter synchronous operation</li> <li>• Maximum of 15-phase PWM output by combination with synchronous operation</li> <li>• Cascaded operation</li> </ul>
Channels 0, 3, 6, and 9	Buffer operation can be set.
Channels 1, 2, 4, 5, 7, 8, 10, and 11	Phase counting mode can be set
Interrupt source	52 sources (for unit 0: 26, for unit 1: 26)
Buffer operation	Automatic transfer of register data
Generation of trigger	Programmable pulse generator (PPG) output trigger can be generated. Conversion start trigger for the A/D converter can be generated.
Low-power consumption function	Module-stop state can be set for each unit.

**Table 24.2 TPU (Unit 0) Functions**

Item	TPU0	TPU1	TPU2	TPU3	TPU4	TPU5
Count clock	PCLK/1 PCLK/4 PCLK/16 PCLK/64 TCLKA TCLKB TCLKC TCLKD	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 TCLKA TCLKB	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 TCLKA TCLKB TCLKC	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 PCLK/1024 PCLK/4096 TCLKA	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 TCLKA TCLKC	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 TCLKA TCLKC TCLKD
Timer general registers	TGRA TGRB TGRC*1 TGRD*1	TGRA TGRB	TGRA TGRB	TGRA TGRB TGRC*1 TGRD*1	TGRA TGRB	TGRA TGRB
I/O pins	TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2	TIOCA3 TIOCB3 TIOCC3 TIOCD3	TIOCA4 TIOCB4	TIOCA5 TIOCB5
Counter clear function	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture
Compare match output	Low output	Possible	Possible	Possible	Possible	Possible
	High output	Possible	Possible	Possible	Possible	Possible
	Toggle output	Possible	Possible	Possible	Possible	Possible
Input capture function	Possible	Possible	Possible	Possible	Possible	Possible
Synchronous operation	Possible	Possible	Possible	Possible	Possible	Possible
PWM mode	Possible	Possible	Possible	Possible	Possible	Possible
Phase counting mode	Not possible	Possible	Possible	—	Possible	Possible
Buffer operation	Possible	Not possible	Not possible	Possible	Not possible	Not possible
DTC activation	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture
DMAC activation	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture
A/D conversion start trigger	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	Not possible
PPG trigger	TGRA/TGRB compare match or input capture	TGRA/TGRB compare match or input capture	TGRA/TGRB compare match or input capture	TGRA/TGRB compare match or input capture	Not possible	Not possible
Interrupt sources	5 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 • Overflow	4 sources • Compare match or input capture 1A • Compare match or input capture 1B  • Overflow • Underflow	4 sources • Compare match or input capture 2A • Compare match or input capture 2B  • Overflow • Underflow	5 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	4 sources • Compare match or input capture 4A • Compare match or input capture 4B  • Overflow • Underflow	4 sources • Compare match or input capture 5A • Compare match or input capture 5B  • Overflow • Underflow
Module-stop setting*2	MSTPA13 bit in MSTPCRA					

Note 1. TGRC and TGRD can be set as a buffer register.

Note 2. For details, see section 11, Low Power Consumption.

**Table 24.3 TPU (Unit 1) Functions**

Item	TPU6	TPU7	TPU8	TPU9	TPU10	TPU11
Count clock	PCLK/1 PCLK/4 PCLK/16 PCLK/64 TCLKE TCLKF TCLKG TCLKH	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 TCLKE TCLKF	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 TCLKE TCLKF TCLKG	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 PCLK/1024 PCLK/4096 TCLKE	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/1024 TCLKE TCLKG	PCLK/1 PCLK/4 PCLK/16 PCLK/64 PCLK/256 TCLKE TCLKG TCLKH
Timer general registers	TGRA TGRB TGRC*1 TGRD*1	TGRA TGRB	TGRA TGRB	TGRA TGRB TGRC*1 TGRD*1	TGRA TGRB	TGRA TGRB
I/O pins	TIOCA6 TIOCB6 TIOCC6 TIOCD6	TIOCA7 TIOCB7	TIOCA8 TIOCB8	TIOCA9 TIOCB9 TIOCC9 TIOCD9	TIOCA10 TIOCB10	TIOCA11 TIOCB11
Counter clear function	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture
Compare match output	Low output	Possible	Possible	Possible	Possible	Possible
	High output	Possible	Possible	Possible	Possible	Possible
	Toggle output	Possible	Possible	Possible	Possible	Possible
Input capture function	Possible	Possible	Possible	Possible	Possible	Possible
Synchronous operation	Possible	Possible	Possible	Possible	Possible	Possible
PWM mode	Possible	Possible	Possible	Possible	Possible	Possible
Phase counting mode	Not possible	Possible	Possible	Not possible	Possible	Possible
Buffer operation	Possible	Not possible	Not possible	Possible	Not possible	Not possible
DTC activation	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture	TGRy compare match or input capture
DMAC activation	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture	TGRA compare match or input capture
PPG trigger	TGRA/TGRB compare match or input capture	TGRA/TGRB compare match or input capture	TGRA/TGRB compare match or input capture	TGRA/TGRB compare match or input capture	Not possible	Not possible
Interrupt sources	5 sources • Compare match or input capture 6A • Compare match or input capture 6B • Compare match or input capture 6C • Compare match or input capture 6D • Overflow	4 sources • Compare match or input capture 7A • Compare match or input capture 7B  • Overflow • Underflow	4 sources • Compare match or input capture 8A • Compare match or input capture 8B  • Overflow • Underflow	5 sources • Compare match or input capture 9A • Compare match or input capture 9B • Compare match or input capture 9C • Compare match or input capture 9D • Overflow	4 sources • Compare match or input capture 10A • Compare match or input capture 10B  • Overflow • Underflow	4 sources • Compare match or input capture 11A • Compare match or input capture 11B  • Overflow • Underflow
Module-stop setting*2	MSTPA12 bit in MSTPCRA					

Note 1. TGRC and TGRD can be set as a buffer register.

Note 2. For details, see section 11, Low Power Consumption.

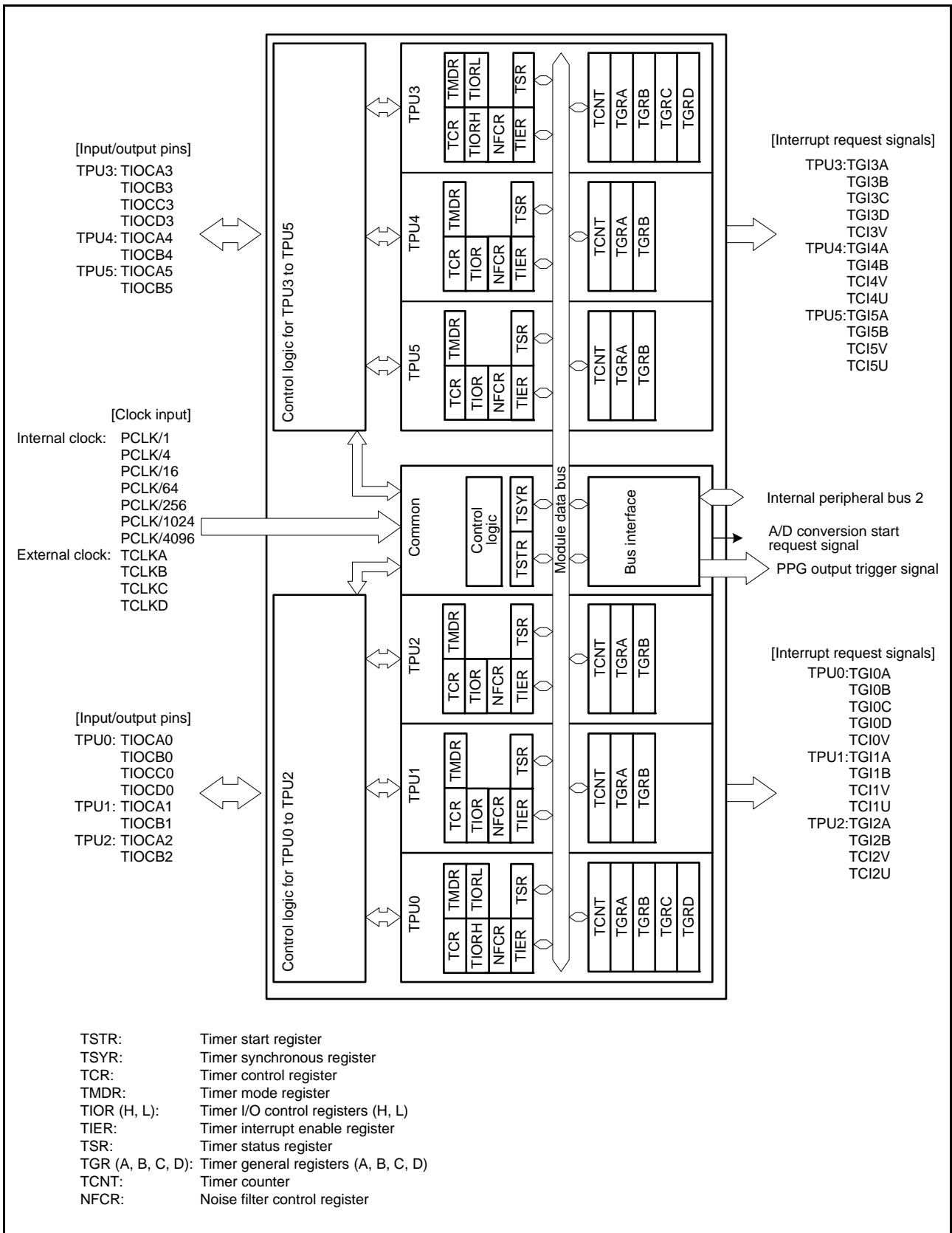


Figure 24.1 Block Diagram of TPU (Unit 0)

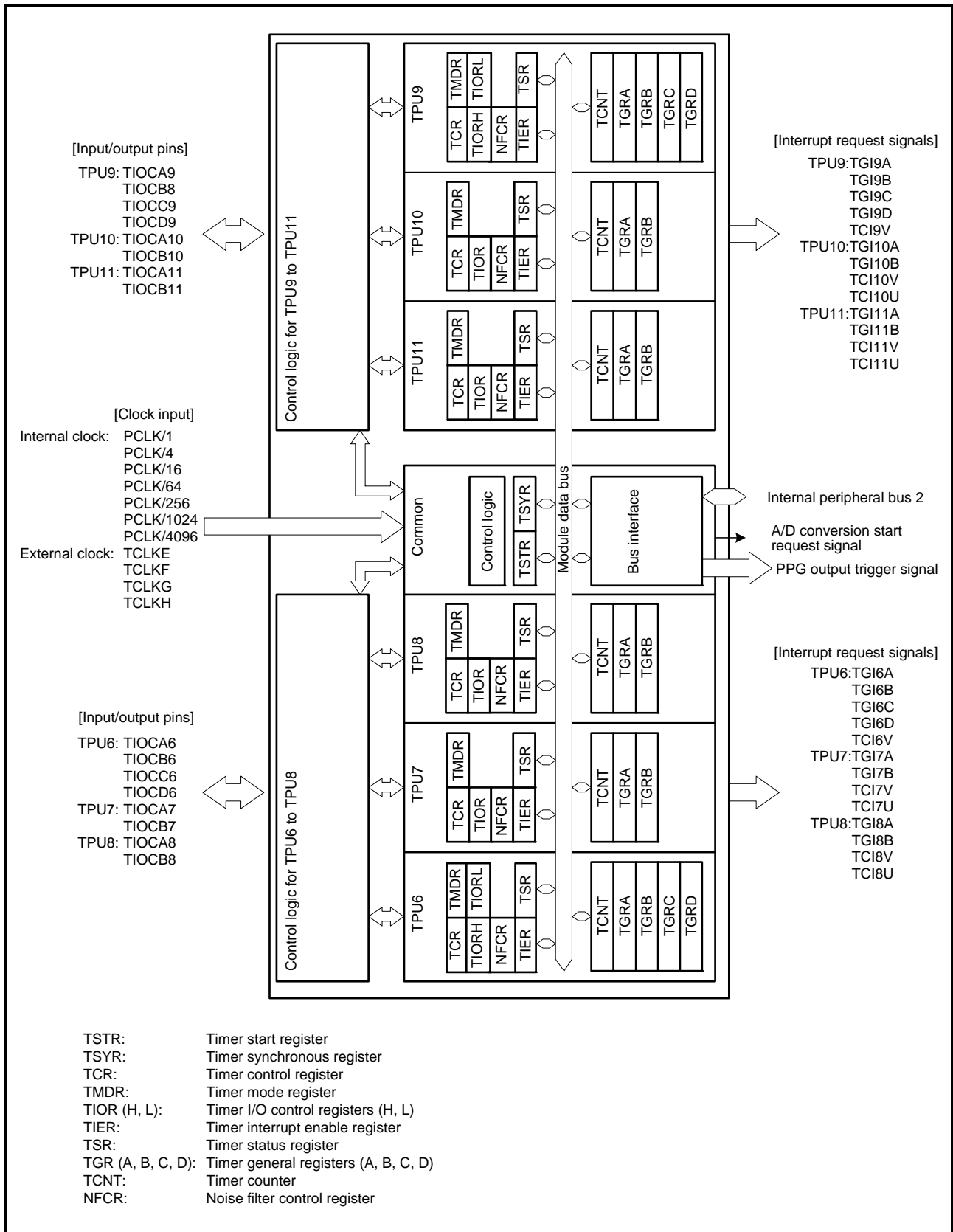


Figure 24.2 Block Diagram of TPU (Unit 1)

Table 24.4 lists the input/output pins of the TPU.

**Table 24.4 Pin Configuration of TPU**

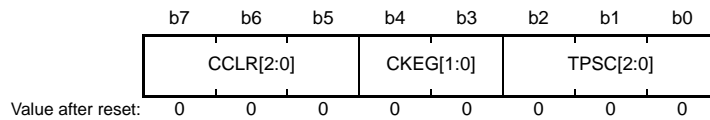
Unit	Channel	Pin Name	I/O	Description
Unit 0	Common to unit 0	TCLKA	Input	External clock A input pin (TPU1 and TPU5 phase counting mode A phase input)
		TCLKB	Input	External clock B input pin (TPU1 and TPU5 phase counting mode B phase input)
		TCLKC	Input	External clock C input pin (TPU2 and TPU4 phase counting mode A phase input)
		TCLKD	Input	External clock D input pin (TPU2 and TPU4 phase counting mode B phase input)
	TPU0	TIOCA0	I/O	TPU0.TGRA input capture input/output compare output/PWM output pin
		TIOCB0	I/O	TPU0.TGRB input capture input/output compare output/PWM output pin
		TIOCC0	I/O	TPU0.TGRC input capture input/output compare output/PWM output pin
		TIOCD0	I/O	TPU0.TGRD input capture input/output compare output/PWM output pin
	TPU1	TIOCA1	I/O	TPU1.TGRA input capture input/output compare output/PWM output pin
		TIOCB1	I/O	TPU1.TGRB input capture input/output compare output/PWM output pin
	TPU2	TIOCA2	I/O	TPU2.TGRA input capture input/output compare output/PWM output pin
		TIOCB2	I/O	TPU2.TGRB input capture input/output compare output/PWM output pin
	TPU3	TIOCA3	I/O	TPU3.TGRA input capture input/output compare output/PWM output pin
		TIOCB3	I/O	TPU3.TGRB input capture input/output compare output/PWM output pin
		TIOCC3	I/O	TPU3.TGRC input capture input/output compare output/PWM output pin
		TIOCD3	I/O	TPU3.TGRD input capture input/output compare output/PWM output pin
	TPU4	TIOCA4	I/O	TPU4.TGRA input capture input/output compare output/PWM output pin
		TIOCB4	I/O	TPU4.TGRB input capture input/output compare output/PWM output pin
	TPU5	TIOCA5	I/O	TPU5.TGRA input capture input/output compare output/PWM output pin
		TIOCB5	I/O	TPU5.TGRB input capture input/output compare output/PWM output pin
Unit 1	Common to unit 1	TCLKE	Input	External clock E input pin (TPU7 and TPU11 phase counting mode A phase input)
		TCLKF	Input	External clock F input pin (TPU7 and TPU11 phase counting mode B phase input)
		TCLKG	Input	External clock G input pin (TPU8 and TPU10 phase counting mode A phase input)
		TCLKH	Input	External clock H input pin (TPU8 and TPU10 phase counting mode B phase input)
	TPU6	TIOCA6	I/O	TPU6.TGRA input capture input/output compare output/PWM output pin
		TIOCB6	I/O	TPU6.TGRB input capture input/output compare output/PWM output pin
		TIOCC6	I/O	TPU6.TGRC input capture input/output compare output/PWM output pin
		TIOCD6	I/O	TPU6.TGRD input capture input/output compare output/PWM output pin
	TPU7	TIOCA7	I/O	TPU7.TGRA input capture input/output compare output/PWM output pin
		TIOCB7	I/O	TPU7.TGRB input capture input/output compare output/PWM output pin
	TPU8	TIOCA8	I/O	TPU8.TGRA input capture input/output compare output/PWM output pin
		TIOCB8	I/O	TPU8.TGRB input capture input/output compare output/PWM output pin
	TPU9	TIOCA9	I/O	TPU9.TGRA input capture input/output compare output/PWM output pin
		TIOCB9	I/O	TPU9.TGRB input capture input/output compare output/PWM output pin
		TIOCC9	I/O	TPU9.TGRC input capture input/output compare output/PWM output pin
		TIOCD9	I/O	TPU9.TGRD input capture input/output compare output/PWM output pin
	TPU10	TIOCA10	I/O	TPU10.TGRA input capture input/output compare output/PWM output pin
		TIOCB10	I/O	TPU10.TGRB input capture input/output compare output/PWM output pin
	TPU11	TIOCA11	I/O	TPU11.TGRA input capture input/output compare output/PWM output pin
		TIOCB11	I/O	TPU11.TGRB input capture input/output compare output/PWM output pin



## 24.2 Register Descriptions

### 24.2.1 Timer Control Register (TCR)

Address(es): TPU0.TCR 0008 8110h, TPU1.TCR 0008 8120h, TPU2.TCR 0008 8130h,  
 TPU3.TCR 0008 8140h, TPU4.TCR 0008 8150h, TPU5.TCR 0008 8160h,  
 TPU6.TCR 0008 8180h, TPU7.TCR 0008 8190h, TPU8.TCR 0008 81A0h,  
 TPU9.TCR 0008 81B0h, TPU10.TCR 0008 81C0h, TPU11.TCR 0008 81D0h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	TPSC[2:0]	Timer Prescaler Select	See Table 24.5 to Table 24.10.	R/W
b4, b3	CKEG[1:0]	Input Clock Edge Select	See Table 24.11.	R/W
b7 to b5	CCLR[2:0] *1	Counter Clear Source Select	See Table 24.12 and Table 24.13.	R/W

Note 1. Bit 7 in TCR of TPU1, TPU2, TPU4, and TPU5 of unit 0 and bit 7 in TCR of TPU7, TPU8, TPU10, and TPU11 of unit 1 are reserved. These bits are read as 0. The write value should be 0.

TPUm.TCR settings should be made while TPUm.TCNT counter operation is stopped.

#### TPSC[2:0] Bits (Timer Prescaler Select)

These bits select the TCNT counter clock. The clock source can be selected independently for each channel.

To select the external clock as the clock source, set the bit in the port direction register (PDR) for the corresponding pin to 0 (input port), and set the bit in the port mode register (PMR) to 1 (uses the pin as an I/O port for peripheral functions). For details, see section 20, I/O Ports.

#### CKEG[1:0] Bits (Input Clock Edge Select)

These bits select the input clock edge.

When the internal clock is counted using both edges, the input clock period is halved (e.g. Both edges of PCLK/4 = PCLK/2 rising edge).

Internal clock edge selection is valid when the input clock is PCLK/4 or slower. This setting is ignored if the input clock is PCLK/1, or when overflow/underflow of another channel is selected.

**Table 24.5 Bits TPSC[2:0] (TPU0, TPU6)**

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU0 (unit 0) TPU6 (unit 1)	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock <ul style="list-style-type: none"> <li>• TPU0 (unit 0): counts on TCLKA pin input</li> <li>• TPU6 (unit 1): counts on TCLKE pin input</li> </ul>
	1	0	1	External clock <ul style="list-style-type: none"> <li>• TPU0 (unit 0): counts on TCLKB pin input</li> <li>• TPU6 (unit 1): counts on TCLKF pin input</li> </ul>
	1	1	0	External clock <ul style="list-style-type: none"> <li>• TPU0 (unit 0): counts on TCLKC pin input</li> <li>• TPU6 (unit 1): counts on TCLKG pin input</li> </ul>
	1	1	1	External clock <ul style="list-style-type: none"> <li>• TPU0 (unit 0): counts on TCLKD pin input</li> <li>• TPU6 (unit 1): counts on TCLKH pin input</li> </ul>

**Table 24.6 Bits TPSC[2:0] (TPU1, TPU7)**

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU1 (unit 0) TPU7 (unit 1)	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock <ul style="list-style-type: none"> <li>• TPU1 (unit 0): counts on TCLKA pin input</li> <li>• TPU7 (unit 1): counts on TCLKE pin input</li> </ul>
	1	0	1	External clock <ul style="list-style-type: none"> <li>• TPU1 (unit 0): counts on TCLKB pin input</li> <li>• TPU7 (unit 1): counts on TCLKF pin input</li> </ul>
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	<ul style="list-style-type: none"> <li>• TPU1 (unit 0) Counts on TPU2.TCNT counter overflow/underflow</li> <li>• TPU7 (unit 1) Counts on TPU8.TCNT counter overflow/underflow</li> </ul>

Note 1. This setting is invalid when TPU1 or TPU7 is in phase counting mode.

**Table 24.7 Bits TPSC[2:0] (TPU2, TPU8)**

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU2 (unit 0)	0	0	0	Internal clock: counts on PCLK/1
TPU8 (unit 1)	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock <ul style="list-style-type: none"> <li>• TPU2 (unit 0): counts on TCLKA pin input</li> <li>• TPU8 (unit 1): counts on TCLKE pin input</li> </ul>
	1	0	1	External clock <ul style="list-style-type: none"> <li>• TPU2 (unit 0): counts on TCLKB pin input</li> <li>• TPU8 (unit 1): counts on TCLKF pin input</li> </ul>
	1	1	0	External clock <ul style="list-style-type: none"> <li>• TPU2 (unit 0): counts on TCLKC pin input</li> <li>• TPU8 (unit 1): counts on TCLKG pin input</li> </ul>
	1	1	1	Internal clock: counts on PCLK/1024

Note 1. This setting is invalid when TPU2 or TPU8 is in phase counting mode.

**Table 24.8 Bits TPSC[2:0] (TPU3, TPU9)**

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU3 (unit 0)	0	0	0	Internal clock: counts on PCLK/1
TPU9 (unit 1)	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock <ul style="list-style-type: none"> <li>• TPU3 (unit 0): counts on TCLKA pin input</li> <li>• TPU9 (unit 1): counts on TCLKE pin input</li> </ul>
	1	0	1	Internal clock: counts on PCLK/1024
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	Internal clock: counts on PCLK/4096

**Table 24.9 Bits TPSC[2:0] (TPU4, TPU10)**

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU4 (unit 0) TPU10 (unit 1)	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock <ul style="list-style-type: none"> <li>• TPU4 (unit 0): counts on TCLKA pin input</li> <li>• TPU10 (unit 1): counts on TCLKE pin input</li> </ul>
	1	0	1	External clock <ul style="list-style-type: none"> <li>• TPU4 (unit 0): counts on TCLKC pin input</li> <li>• TPU10 (unit 1): counts on TCLKG pin input</li> </ul>
	1	1	0	Internal clock: counts on PCLK/1024
	1	1	1	<ul style="list-style-type: none"> <li>• TPU4 (unit 0) Counts on TPU5.TCNT counter overflow/underflow</li> <li>• TPU10 (unit 1) Counts on TPU11.TCNT counter overflow/underflow</li> </ul>

Note 1. This setting is invalid when TPU4 or TPU10 is in phase counting mode.

**Table 24.10 Bits TPSC[2:0] (TPU5, TPU11)**

Channel	Bits TPSC[2:0]			Description
	b2	b1	b0	
TPU5 (unit 0) TPU11 (unit 1)	0	0	0	Internal clock: counts on PCLK/1
	0	0	1	Internal clock: counts on PCLK/4
	0	1	0	Internal clock: counts on PCLK/16
	0	1	1	Internal clock: counts on PCLK/64
	1	0	0	External clock <ul style="list-style-type: none"> <li>• TPU5 (unit 0): counts on TCLKA pin input</li> <li>• TPU11 (unit 1): counts on TCLKE pin input</li> </ul>
	1	0	1	External clock <ul style="list-style-type: none"> <li>• TPU5 (unit 0): counts on TCLKC pin input</li> <li>• TPU11 (unit 1): counts on TCLKG pin input</li> </ul>
	1	1	0	Internal clock: counts on PCLK/256
	1	1	1	External clock <ul style="list-style-type: none"> <li>• TPU5 (unit 0): counts on TCLKD pin input</li> <li>• TPU11 (unit 1): counts on TCLKH pin input</li> </ul>

Note 1. This setting is invalid when TPU5 or TPU11 is in phase counting mode.

**Table 24.11 Bits CKEG[1:0]**

Bits CKEG[1:0]		Input Clock	
b4	b3	Internal Clock	External clock
0	0	Counted at falling edge	Counted at rising edge
0	1	Counted at rising edge	Counted at falling edge
1	0	Counted at both edges	Counted at both edges
1	1	Counted at both edges	Counted at both edges

**Table 24.12 Bits CCLR[2:0] (TPU0, TPU3, TPU6, TPU9)**

Channel	Bits CCLR[2:0]			Description
	b7	b6	b5	
(Unit 0) TPU0, TPU3	0	0	0	TCNT counter clearing disabled
	0	0	1	TCNT counter cleared by TGRA compare match/input capture
(Unit 1) TPU6, TPU9	0	1	0	TCNT counter cleared by TGRB compare match/input capture
	0	1	1	TCNT counter cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*2
	1	0	0	TCNT counter clearing disabled
	1	0	1	TCNT counter cleared by TGRC compare match/input capture*1
	1	1	0	TCNT counter cleared by TGRD compare match/input capture*1
	1	1	1	TCNT counter cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*2

Note 1. When TGRC or TGRD is used as a buffer register, TCNT counter is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.

Note 2. Synchronous operation is selected by setting the TPUA.TSYR.SYNCj bit (j = 0, 3) and the TPUB.TSYR.SYNCj bit (j = 6, 9) to 1.

**Table 24.13 Bits CCLR[2:0] (TPU1, TPU2, TPU4, TPU5, TPU7, TPU8, TPU10, TPU11)**

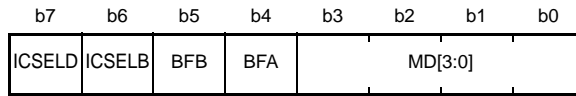
Channel	Bits CCLR[2:0]*1			Description
	b7	b6	b5	
(Unit 0) TPU1, TPU2, TPU4, TPU5	0	0	0	TCNT counter clearing disabled
	0	0	1	TCNT counter cleared by TGRA compare match/input capture
(Unit 1) TPU7, TPU8, TPU10, TPU11	0	1	0	TCNT counter cleared by TGRB compare match/input capture
	0	1	1	TCNT counter cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation*2
	1	0	0	Setting prohibited
	1	0	1	Setting prohibited
	1	1	0	Setting prohibited
	1	1	1	Setting prohibited

Note 1. Bit 7 in TCR of TPU1, TPU2, TPU4, and TPU5 of unit 0 and bit 7 in TCR of TPU7, TPU8, TPU10, and TPU11 of unit 1 are reserved. These bits are read as 0. The write value should be 0.

Note 2. Synchronous operation is selected by setting the TPUA.TSYR.SYNCj bit (j = 1, 2, 4, 5) and the TPUB.TSYR.SYNCj bit (j = 7, 8, 10, 11) to 1.

## 24.2.2 Timer Mode Register (TMDR)

Address(es): TPU0.TMDR 0008 8111h, TPU1.TMDR 0008 8121h, TPU2.TMDR 0008 8131h, TPU3.TMDR 0008 8141h, TPU4.TMDR 0008 8151h, TPU5.TMDR 0008 8161h, TPU6.TMDR 0008 8181h, TPU7.TMDR 0008 8191h, TPU8.TMDR 0008 81A1h, TPU9.TMDR 0008 81B1h, TPU10.TMDR 0008 81C1h, TPU11.TMDR 0008 81D1h



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	b3 <sup>1</sup> b0 0 0 0 0: Normal operation 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 <sup>2</sup> 0 1 0 1: Phase counting mode 2 <sup>2</sup> 0 1 1 0: Phase counting mode 3 <sup>2</sup> 0 1 1 1: Phase counting mode 4 <sup>2</sup> Settings other than above are prohibited.	R/W
b4	BFA <sup>*3</sup>	Buffer Operation A	0: TPUm.TGRA operates normally 1: TPUm.TGRA and TPUm.TGRC used together for buffer operation (m = 0, 3, 6, 9)	R/W
b5	BFB <sup>*4</sup>	Buffer Operation B	0: TPUm.TGRB operates normally 1: TPUm.TGRB and TPUm.TGRD used together for buffer operation (m = 0, 3, 6, 9)	R/W
b6	ICSELB	TGRB Input Capture Input Select	0: Input capture input source is TIOCBn pin 1: Input capture input source is TIOCA <sub>n</sub> pin (n = 0 to 11)	R/W
b7	ICSELD <sup>*4</sup>	GRD Input Capture Input Select	0: Input capture input source is TIOCDn pin 1: Input capture input source is TIOCC <sub>n</sub> pin (n = 0, 3, 6, 9)	R/W

Note 1. Bit 3 is reserved. This bit is read as 0. The write value should be 0.

Note 2. Phase counting mode cannot be set for TPU0, TPU3 (unit 0), TPU6, and TPU9 (unit 1). A 0 should be written to bit 2 for them.

Note 3. Bit 4 of TPU1, TPU2, TPU4, TPU5 (unit 0), TPU7, TPU8, TPU10, and TPU11 (unit 1) that do not have TGRC is reserved. This bit is read as 0. The write value should be 0.

Note 4. Bits 5 and 7 of TPU1, TPU2, TPU4, TPU5 (unit 0), TPU7, TPU8, TPU10, and TPU11 (unit 1) that do not have TGRD are reserved. These bits are read as 0. The write value should be 0.

TPUm.TMDR settings should be made while TPUm.TCNT counter operation is stopped.

### BFA Bit (Buffer Operation A)

Specifies whether TPUm.TGRA (m = 0, 3, 6, 9) is to normally operate, or TPUm.TGRA and TPUm.TGRC (m = 0, 3, 6, 9) are to be used together for buffer operation.

When TGRC is used as a buffer register, TGRC input capture/output compare is not generated.

### BFB Bit (Buffer Operation B)

Specifies whether TPUm.TGRB (m = 0, 3, 6, 9) is to normally operate, or TPUm.TGRB and TPUm.TGRD (m = 0, 3, 6, 9) are to be used together for buffer operation.

When TGRD is used as a buffer register, TGRD input capture/output compare is not generated.

### ICSELB Bit (TGRB Input Capture Input Select)

Selects the input capture input for TPUm.TGRB (m = 0 to 11).

This function allows measurement of high-level width and period of the input pulse on a TIOCA<sub>n</sub> input pin.

**ICSELD Bit (TGRD Input Capture Input Select)**

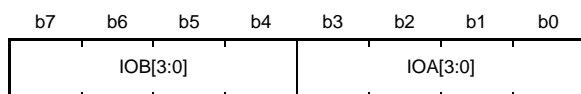
Selects the input capture input for TPU<sub>m</sub>.TGRD (m = 0, 3, 6, 9).

This function allows measurement of high-level width and period of the input pulse on a TIOCC<sub>n</sub> input pin.

**24.2.3 Timer I/O Control Register (TIORH, TIORL, TIOR)**

- Unit 0 (TPU0.TIORH, TPU1.TIOR, TPU2.TIOR, TPU3.TIORH, TPU4.TIOR, TPU5.TIOR)  
Unit 1 (TPU6.TIORH, TPU7.TIOR, TPU8.TIOR, TPU9.TIORH, TPU10.TIOR, TPU11.TIOR)

Address(es): TPU0.TIORH 0008 8112h, TPU1.TIOR 0008 8122h, TPU2.TIOR 0008 8132h,  
TPU3.TIORH 0008 8142h, TPU4.TIOR 0008 8152h, TPU5.TIOR 0008 8162h,  
TPU6.TIORH 0008 8182h, TPU7.TIOR 0008 8192h, TPU8.TIOR 0008 81A2h,  
TPU9.TIORH 0008 81B2h, TPU10.TIOR 0008 81C2h, TPU11.TIOR 0008 81D2h



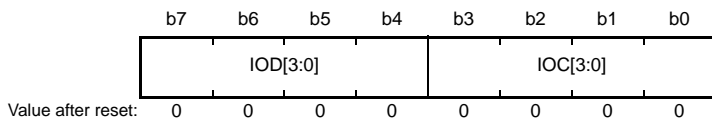
Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOA[3:0]	TGRA Control	See Table 24.14 to Table 24.19.*1	R/W
b7 to b4	IOB[3:0]	TGRB Control	See Table 24.14 to Table 24.19.*1	R/W

Note 1. If the IO<sub>n</sub>[3:0] bit (n = A or B) values are changed to output disabled (0000b or 0100b) during low/high/toggle output on compare match, the TIOCA<sub>n</sub>/TIOCB<sub>n</sub> pin (n = 0 to 11) is placed in high impedance state.

- Unit 0 (TPU0.TIORL, TPU3.TIORL)  
Unit 1 (TPU6.TIORL, TPU9.TIORL)

Address(es): TPU0.TIORL 0008 8113h, TPU3.TIORL 0008 8143h,  
TPU6.TIORL 0008 8183h, TPU9.TIORL 0008 81B3h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	IOC[3:0]	TGRC Control	See Table 24.20 and Table 24.21.*1	R/W
b7 to b4	IOD[3:0]	TGRD Control	See Table 24.20 and Table 24.21.*1	R/W

Note 1. If the IOn[3:0] bit (n = C or D) values are changed to output disabled (0000b or 0100b) during low/high/toggle output on compare match, the TIOCCn/TIOCDn pin (n = 0, 3, 6, or 9) is placed in high impedance state.

TPU has four TIORH registers, one for TPU0, TPU3, TPU6, and TPU9, and four TIORL registers, one for TPU0, TPU3, TPU6, and TPU9, and also has eight TIOR registers, one for TPU1, TPU2, TPU4, TPU5, TPU7, TPU8, TPU10, and TPU11. Thus the TPU has sixteen timer I/O control registers in total.

TIORH, TIORL, and TIOR control registers TGRA, TGRB, TGRC, and TGRD.

Note that TIORH, TIORL, and TIOR are affected by the TMDR setting. For details, see Table 24.14 to Table 24.21.

The initial output specified by TIORH, TIORL, and TIOR is valid when the counter is stopped (the TPUA.TSTR.CSTj bit (j = 0 to 5) and the TPUB.TSTR.CSTj bit (j = 6 to 11) are cleared to 0). In PWM mode 2, the output at the time when the TCNT counter is cleared to 0 is specified as the initial output.

When buffer operation has been selected for register TGRC or TGRD, the settings of the corresponding set of IOC[3:0] or IOD[3:0] bits becomes ineffective, and the TGRC or TGRD register simply operates as a buffer.

To specify the input capture pin in TIORH, TIORL, or TIOR, set the bit in the port direction register (PDR) for the corresponding pin to 0 (input port), and set the bit in the port mode register (PMR) to 1 (uses the pin as an I/O port for peripheral functions). For details, see section 20, I/O Ports.

### IOA[3:0] Bits (TGRA Control)

Select the function of TPUm.TGRA (m = 0 to 11).

### IOB[3:0] Bits (TGRB Control)

Select the function of TPUm.TGRB (m = 0 to 11).

### IOC[3:0] Bits (TGRC Control)

Select the function of TPUm.TGRC (m = 0, 3, 6, 9).

### IOD[3:0] Bits (TGRD Control)

Select the function of TPUm.TGRD (m = 0, 3, 6, 9).



**Table 24.14 TPU0.TIORH, TPU6.TIORH**

Bits IOA[3:0]				Description	
b3	b2	b1	b0	TPUm.TGRA (m = 0, 6) Function	TIOCA <sub>n</sub> Pin (n = 0, 6) Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0		Input capture register
1	0	0	1	Capture input source is TIOCA <sub>n</sub> pin; input capture at falling edge	
1	0	1	x	Capture input source is TIOCA <sub>n</sub> pin; input capture at both edges	
1	1	x	x	<ul style="list-style-type: none"> <li>• TPU0 Capture input source is TPU1 count clock Input capture at TPU1.TCNT count-up/count-down*1</li> <li>• TPU6 Capture input source is TPU7 count clock Input capture at TPU7.TCNT count-up/count-down*1</li> </ul>	

Bits IOB[3:0]				Description	
b7	b6	b5	b4	TPUm.TGRB (m = 0, 6) Function	TIOCB <sub>n</sub> Pin (n = 0, 6) Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0		Input capture register
1	0	0	1	Capture input source is TIOCB <sub>n</sub> or TIOCA <sub>n</sub> pin*2; input capture at falling edge	
1	0	1	x	Capture input source is TIOCB <sub>n</sub> or TIOCA <sub>n</sub> pin*2; input capture at both edges	
1	1	x	x	<ul style="list-style-type: none"> <li>• TPU0 Capture input source is TPU1 count clock Input capture at TPU1.TCNT count-up/count-down*1</li> <li>• TPU6 Capture input source is TPU7 count clock Input capture at TPU7.TCNT count-up/count-down*1</li> </ul>	

x: Don't care

Note 1. When the TPSC[2:0] bits in TPUm.TCR are set to 000b and PCLK/1 is used as the TPUm.TCNT count clock, this setting is invalid and input capture is not generated (m = 1, 7).

Note 2. Selected by the ICSELB bit in TPUm.TMDR (m = 0, 6).

**Table 24.15 TPU1.TIOR, TPU7.TIOR**

Bits IOA[3:0]				Description	
b3	b2	b1	b0	TPUm.TGRA (m = 1, 7) Function	TIOCA <sub>n</sub> Pin (n = 1, 7) Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0		Input capture register
1	0	0	1	Capture input source is TIOCA <sub>n</sub> pin; input capture at falling edge	
1	0	1	x	Capture input source is TIOCA <sub>n</sub> pin; input capture at both edges	
1	1	x	x	<ul style="list-style-type: none"> <li>• TPU1 Capture input source is TPU0.TGRA compare match/input capture Input capture at generation of TPU0.TGRA compare match/input capture</li> <li>• TPU7 Capture input source is TPU6.TGRA compare match/input capture Input capture at generation of TPU6.TGRA compare match/input capture</li> </ul>	

Bits IOB[3:0]				Description	
b7	b6	b5	b4	TPUm.TGRB (m = 1, 7) Function	TIOCB <sub>n</sub> Pin (n = 1, 7) Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0		Input capture register
1	0	0	1	Capture input source is TIOCB <sub>n</sub> or TIOCA <sub>n</sub> pin*1; input capture at falling edge	
1	0	1	x	Capture input source is TIOCB <sub>n</sub> or TIOCA <sub>n</sub> pin*1; input capture at both edges	
1	1	x	x	<ul style="list-style-type: none"> <li>• TPU1 Capture input source is TPU0.TGRC compare match/input capture Input capture at generation of TPU0.TGRC compare match/input capture</li> <li>• TPU7 Capture input source is TPU6.TGRC compare match/input capture Input capture at generation of TPU6.TGRC compare match/input capture</li> </ul>	

x: Don't care

Note 1. Selected by the ICSELB bit in TPUm.TMDR (m = 1, 7).

**Table 24.16 TPU2.TIOR, TPU8.TIOR**

Bits IOA[3:0]				Description	
b3	b2	b1	b0	TPUm.TGRA (m = 2, 8) Function	TIOCA <sub>n</sub> Pin (n = 2, 8) Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	x	0	0		Input capture register
1	x	0	1	Capture input source is TIOCA <sub>n</sub> pin; input capture at falling edge	
1	x	1	x	Capture input source is TIOCA <sub>n</sub> pin; input capture at both edges	

Bits IOB[3:0]				Description	
b7	b6	b5	b4	TPUm.TGRB (m = 2, 8) Function	TIOCB <sub>n</sub> Pin (n = 2, 8) Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	x	0	0		Input capture register
1	x	0	1	Capture input source is TIOCB <sub>n</sub> or TIOCA <sub>n</sub> pin* <sup>1</sup> ; input capture at falling edge	
1	x	1	x	Capture input source is TIOCB <sub>n</sub> or TIOCA <sub>n</sub> pin* <sup>1</sup> ; input capture at both edges	

x: Don't care

Note 1. Selected by the ICSELB bit in TPUm.TMDR (m = 2, 8).

Table 24.17 TPU3.TIORH, TPU9.TIORH

Bits IOA[3:0]				Description	
b3	b2	b1	b0	TPUm.TGRA (m = 3, 9) Function	TIOCA <sub>n</sub> Pin (n = 3, 9) Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA <sub>n</sub> pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCA <sub>n</sub> pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCA <sub>n</sub> pin; input capture at both edges
1	1	x	x		<ul style="list-style-type: none"> <li>• TPU3 Capture input source is TPU4 count clock Input capture at TPU4.TCNT count-up/count-down*1</li> <li>• TPU9 Capture input source is TPU10 count clock Input capture at TPU10.TCNT count-up/count-down*1</li> </ul>

Bits IOB[3:0]				Description	
b7	b6	b5	b4	TPUm.TGRB (m = 3, 9) Function	TIOCB <sub>n</sub> Pin (n = 3, 9) Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB <sub>n</sub> or TIOCA <sub>n</sub> pin*2; input capture at rising edge
1	0	0	1		Capture input source is TIOCB <sub>n</sub> or TIOCA <sub>n</sub> pin*2; input capture at falling edge
1	0	1	x		Capture input source is TIOCB <sub>n</sub> or TIOCA <sub>n</sub> pin*2; input capture at both edges
1	1	x	x		<ul style="list-style-type: none"> <li>• TPU3 Capture input source is TPU4 count clock Input capture at TPU4.TCNT count-up/count-down*1</li> <li>• TPU9 Capture input source is TPU10 count clock Input capture at TPU10.TCNT count-up/count-down*1</li> </ul>

x: Don't care

Note 1. When the TPSC[2:0] bits in TPUm.TCR are set to 000b and PCLK/1 is used as the TPUm.TCNT count clock, this setting is invalid and input capture is not generated (m = 4, 10).

Note 2. Selected by the ICSELB bit in TPUm.TMDR (m = 3, 9).

**Table 24.18 TPU4.TIOR, TPU10.TIOR**

Bits IOA[3:0]				Description	
b3	b2	b1	b0	TPUm.TGRA (m = 4, 10) Function	TIOCA <sub>n</sub> Pin (n = 4, 10) Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCA <sub>n</sub> pin; input capture at rising edge
1	0	0	1		Capture input source is TIOCA <sub>n</sub> pin; input capture at falling edge
1	0	1	x		Capture input source is TIOCA <sub>n</sub> pin; input capture at both edges
1	1	x	x		<ul style="list-style-type: none"> <li>• TPU4 Capture input source is TPU3.TGRA compare match/input capture Input capture at generation of TPU3.TGRA compare match/input capture</li> <li>• TPU10 Capture input source is TPU9.TGRA compare match/input capture Input capture at generation of TPU9.TGRA compare match/input capture</li> </ul>

Bits IOB[3:0]				Description	
b7	b6	b5	b4	TPUm.TGRB (m = 4, 10) Function	TIOCB <sub>n</sub> Pin (n = 4, 10) Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0	Input capture register	Capture input source is TIOCB <sub>n</sub> or TIOCA <sub>n</sub> pin*1; input capture at rising edge
1	0	0	1		Capture input source is TIOCB <sub>n</sub> or TIOCA <sub>n</sub> pin*1; input capture at falling edge
1	0	1	x		Capture input source is TIOCB <sub>n</sub> or TIOCA <sub>n</sub> pin*1; input capture at both edges
1	1	x	x		<ul style="list-style-type: none"> <li>• TPU4 Capture input source is TPU3.TGRC compare match/input capture Input capture at generation of TPU3.TGRC compare match/input capture</li> <li>• TPU10 Capture input source is TPU9.TGRC compare match/input capture Input capture at generation of TPU9.TGRC compare match/input capture</li> </ul>

x: Don't care

Note 1. Selected by the ICSELB bit in TPUm.TMDR (m = 4, 10).

**Table 24.19 TPU5.TIOR, TPU11.TIOR**

Bits IOA[3:0]				Description	
b3	b2	b1	b0	TPUm.TGRA (m = 5, 11) Function	TIOCA <sub>n</sub> Pin (n = 5, 11) Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	x	0	0		Input capture register
1	x	0	1	Capture input source is TIOCA <sub>n</sub> pin; input capture at falling edge	
1	x	1	x	Capture input source is TIOCA <sub>n</sub> pin; input capture at both edges	

Bits IOB[3:0]				Description	
b7	b6	b5	b4	TPUm.TGRB (m = 5, 11) Function	TIOCB <sub>n</sub> Pin (n = 5, 11) Function and Related Issue
0	0	0	0	Output compare register	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	x	0	0		Input capture register
1	x	0	1	Capture input source is TIOCB <sub>n</sub> or TIOCA <sub>n</sub> pin*1; input capture at falling edge	
1	x	1	x	Capture input source is TIOCB <sub>n</sub> or TIOCA <sub>n</sub> pin*1; input capture at both edges	

x: Don't care

Note 1. Selected by the ICSELB bit in TPUm.TMDR (m = 5, 11).

**Table 24.20 TPU0.TIORL, TPU6.TIORL**

Bits IOC[3:0]				Description	
b3	b2	b1	b0	TPUm.TGRC (m = 0, 6) Function	TIOCCn Pin (n = 0, 6) Function and Related Issue
0	0	0	0	Output compare register*1	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0		Input capture register*1
1	0	0	1	Capture input source is TIOCCn pin; input capture at falling edge	
1	0	1	x	Capture input source is TIOCCn pin; input capture at both edges	
1	1	x	x	<ul style="list-style-type: none"> <li>TPU0 Capture input source is TPU1 count clock Input capture at TPU1.TCNT count-up/count-down*3</li> <li>TPU6 Capture input source is TPU7 count clock Input capture at TPU7.TCNT count-up/count-down*3</li> </ul>	

Bits IOD[3:0]				Description	
b7	b6	b5	b4	TPUm.TGRD (m = 0, 6) Function	TIOCDn Pin (n = 0, 6) Function and Related Issue
0	0	0	0	Output compare register*2	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0		Input capture register*2
1	0	0	1	Capture input source is TIOCDn or TIOCCn pin*4; input capture at falling edge	
1	0	1	x	Capture input source is TIOCDn or TIOCCn pin*4; input capture at both edges	
1	1	x	x	<ul style="list-style-type: none"> <li>TPU0 Capture input source is TPU1 count clock Input capture at TPU1.TCNT count-up/count-down*3</li> <li>TPU6 Capture input source is TPU7 count clock Input capture at TPU7.TCNT count-up/count-down*3</li> </ul>	

x: Don't care

Note 1. When the BFA bit in TPUm.TMDR is set to 1 (TPUm.TGRA and TPUm.TGRC are used for buffer operation) and TPUm.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated (m = 0, 6).

Note 2. When the BFB bit in TPUm.TMDR is set to 1 (TPUm.TGRB and TPUm.TGRD are used for buffer operation) and TPUm.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated (m = 0, 6).

Note 3. When the TPSC[2:0] bits in TPUm.TCR are set to 000b and PCLK/1 is used as the TPUm.TCNT count clock, this setting is invalid and input capture is not generated (m = 1, 7).

Note 4. Selected by the ICSELD bit in TPUm.TMDR (m = 0, 6).

**Table 24.21 TPU3.TIORL, TPU9.TIORL**

Bits IOC[3:0]				Description	
b3	b2	b1	b0	TPUm.TGRC (m = 3, 9) Function	TIOCCn Pin (m = 3, 9) Function and Related Issue
0	0	0	0	Output compare register*1	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0		Input capture register*1
1	0	0	1	Capture input source is TIOCCn pin; input capture at falling edge	
1	0	1	x	Capture input source is TIOCCn pin; input capture at both edges	
1	1	x	x	<ul style="list-style-type: none"> <li>• TPU3 Capture input source is TPU4 count clock Input capture at TPU4.TCNT count-up/count-down*3</li> <li>• TPU9 Capture input source is TPU10 count clock Input capture at TPU10.TCNT count-up/count-down*3</li> </ul>	

Bits IOD[3:0]				Description	
b7	b6	b5	b4	TPUm.TGRD (m = 3, 9) Function	TIOCDn Pin (n = 3, 9) Function and Related Issue
0	0	0	0	Output compare register*2	Output disabled
0	0	0	1		Initial output is low output; low output at compare match
0	0	1	0		Initial output is low output; high output at compare match
0	0	1	1		Initial output is low output; toggle output at compare match
0	1	0	0		Output disabled
0	1	0	1		Initial output is high output; low output at compare match
0	1	1	0		Initial output is high output; high output at compare match
0	1	1	1		Initial output is high output; toggle output at compare match
1	0	0	0		Input capture register*2
1	0	0	1	Capture input source is TIOCDn or TIOCCn pin*4; input capture at falling edge	
1	0	1	x	Capture input source is TIOCDn or TIOCCn pin*4; input capture at both edges	
1	1	X	x	<ul style="list-style-type: none"> <li>• TPU3 Capture input source is TPU4 count clock Input capture at TPU4.TCNT count-up/count-down*3</li> <li>• TPU9 Capture input source is TPU10 count clock Input capture at TPU10.TCNT count-up/count-down*3</li> </ul>	

x: Don't care

Note 1. When the BFA bit in TPUm.TMDR is set to 1 (TPUm.TGRA and TPUm.TGRC are used for buffer operation) and TPUm.TGRC is used as a buffer register, this setting is invalid and input capture/output compare is not generated (m = 3, 9).

Note 2. When the BFB bit in TPUm.TMDR is set to 1 (TPUm.TGRB and TPUm.TGRD are used for buffer operation) and TPUm.TGRD is used as a buffer register, this setting is invalid and input capture/output compare is not generated (m = 3, 9).

Note 3. When the TPSC[2:0] bits in TPUm.TCR are set to 000b and PCLK/1 is used as the TPUm.TCNT count clock, this setting is invalid and input capture is not generated (m = 4, 10).

Note 4. Selected by the ICSELD bit in TPUm.TMDR (m = 3, 9).



## 24.2.4 Timer Interrupt Enable Register (TIER)

Address(es): TPU0.TIER 0008 8114h, TPU1.TIER 0008 8124h, TPU2.TIER 0008 8134h,  
 TPU3.TIER 0008 8144h, TPU4.TIER 0008 8154h, TPU5.TIER 0008 8164h,  
 TPU6.TIER 0008 8184h, TPU7.TIER 0008 8194h, TPU8.TIER 0008 81A4h,  
 TPU9.TIER 0008 81B4h, TPU10.TIER 0008 81C4h, TPU11.TIER 0008 81D4h

b7	b6	b5	b4	b3	b2	b1	b0
TTGE	—	TCIEU	TCIEV	TGIED	TGIEC	TGIEB	TGIEA

Value after reset: 0 1 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	TGIEA	TGRA Interrupt Enable	0: Interrupt requests (TGImA) disabled 1: Interrupt requests (TGImA) enabled (m = 0 to 11)	R/W
b1	TGIEB	TGRB Interrupt Enable	0: Interrupt requests (TGImB) disabled 1: Interrupt requests (TGImB) enabled (m = 0 to 11)	R/W
b2	TGIEC*1	TGRC Interrupt Enable	0: Interrupt requests (TGImC) disabled 1: Interrupt requests (TGImC) enabled (m = 0, 3, 6, 9)	R/W
b3	TGIED*1	TGRD Interrupt Enable	0: Interrupt requests (TGImD) disabled 1: Interrupt requests (TGImD) enabled (m = 0, 3, 6, 9)	R/W
b4	TCIEV	Overflow Interrupt Enable	0: Interrupt requests (TCImV) disabled 1: Interrupt requests (TCImV) enabled (m = 0 to 11)	R/W
b5	TCIEU*2	Underflow Interrupt Enable	0: Interrupt requests (TCImU) disabled 1: Interrupt requests (TCImU) enabled (m = 1, 2, 4, 5, 7, 8, 10, 11)	R/W
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7	TTGE*3	A/D Conversion Start Request Enable	0: A/D conversion start request generation disabled 1: A/D conversion start request generation enabled	R/W

Note 1. Bits 3 and 2 in TIER of TPU1, TPU2, TPU4, TPU5 (unit 0), TPU7, TPU8, TPU10, and TPU11 (unit 1) are reserved. These bits are read as 0. The write value should be 0.

Note 2. Bit 5 in TIER of TPU0, TPU3 (unit 0), TPU6, and TPU9 (unit 1) is reserved. This bit is read as 0. The write value should be 0.

Note 3. Bit 7 in TIER of TPU5 (unit 0), and TPU6 to TPU11 (unit 1) is reserved. This bit is read as 0. The write value should be 0.

### TTGE Bit (A/D Conversion Start Request Enable)

Enables/disables generation of A/D conversion start requests by TPU<sub>m</sub>.TGRA (m = 0 to 11) input capture/compare match.

## 24.2.5 Timer Status Register (TSR)

Address(es): TPU0.TSR 0008 8115h, TPU1.TSR 0008 8125h, TPU2.TSR 0008 8135h,  
 TPU3.TSR 0008 8145h, TPU4.TSR 0008 8155h, TPU5.TSR 0008 8165h,  
 TPU6.TSR 0008 8185h, TPU7.TSR 0008 8195h, TPU8.TSR 0008 81A5h,  
 TPU9.TSR 0008 81B5h, TPU10.TSR 0008 81C5h, TPU11.TSR 0008 81D5h

b7	b6	b5	b4	b3	b2	b1	b0
TCFD	—	TCFU	TCFV	TGFD	TGFC	TGFB	TGFA

Value after reset: 1 1 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	TGFA	Input Capture/Output Compare Flag A	0: Input capture to TPUM.TGRA or compare match with TPUM.TGRA has not occurred. 1: Input capture to TPUM.TGRA or compare match with TPUM.TGRA has occurred. (m = 0 to 11)	R/W*2
b1	TGFB	Input Capture/Output Compare Flag B	0: Input capture to TPUM.TGRB or compare match with TPUM.TGRB has not occurred. 1: Input capture to TPUM.TGRB or compare match with TPUM.TGRB has occurred. (m = 0 to 11)	R/W*2
b2	TGFC*4	Input Capture/Output Compare Flag C	0: Input capture to TPUM.TGRC or compare match with TPUM.TGRC has not occurred. 1: Input capture to TPUM.TGRC or compare match with TPUM.TGRC has occurred. (m = 0, 3, 6, 9)	R/W*2
b3	TGFD*4	Input Capture/Output Compare Flag D	0: Input capture to TPUM.TGRD or compare match with TPUM.TGRD has not occurred. 1: Input capture to TPUM.TGRD or compare match with TPUM.TGRD has occurred. (m = 0, 3, 6, 9)	R/W*2
b4	TCFV	Overflow Flag	0: TPUM.TCNT has not overflowed. 1: TPUM.TCNT has overflowed. (m = 0 to 11)	R/W*2
b5	TCFU*3	Underflow Flag	0: TPUM.TCNT has not underflowed. 1: TPUM.TCNT has underflowed. (m = 1, 2, 4, 5, 7, 8, 10, 11)	R/W*2
b6	—	Reserved	This bit is read as 1. The write value should be 1.	R
b7	TCFD*1	Counting Direction Flag	0: Counter (TPUM.TCNT) counts down. 1: Counter (TPUM.TCNT) counts up. (m = 1, 2, 4, 5, 7, 8, 10, 11)	R

Note 1. Bit 7 of registers TPU0.TSR, TPU3.TSR, TPU6.TSR, and TPU9.TSR is reserved. The bit is read as 1. When writing, write 1 to this bit.

Note 2. Only writing 0 to this bit is possible; this clears the flag.

Note 3. Bit 5 of registers TPU0.TSR, TPU3.TSR, TPU6.TSR, and TPU9.TSR is reserved. The bit is read as 0. When writing, write 0 to this bit.

Note 4. Bits 2 and 3 of registers TPU1.TSR, TPU2.TSR, TPU4.TSR, TPU5.TSR, TPU7.TSR, TPU8.TSR, TPU10.TSR, and TPU11.TSR are reserved. The bits are read as 0. When writing, write 0 to these bits.

### TGFA Flag (Input Capture/Output Compare Flag A)

This status flag indicates that input capture to TPUM.TGRA or compare match with TPUM.TGRA (m = 0 to 11) has occurred.

[Setting conditions]

- When TPUM.TGRA holds the value for comparison in output-compare operations, TPUM.TCNT matches TPUM.TGRA.

- When TPUm.TGRA is serving as an input-capture register, the input-capture signal has caused transfer of the value in TPUm.TCNT to TPUm.TGRA.

[Clearing conditions]

- Activation of the DTC by the TGImA interrupt and clearing of the DTC.MRB.DISEL bit.
- Writing 0 to TGFA after reading its value as 1.

#### **TGFB Flag (Input Capture/Output Compare Flag B)**

This status flag indicates that input capture to TPUm.TGRB or compare match with TPUm.TGRB (m = 0 to 11) has occurred.

[Setting conditions]

- When TPUm.TGRB holds the value for comparison in output-compare operations, TPUm.TCNT matches TPUm.TGRB.
- When TPUm.TGRB is serving as an input-capture register, the input-capture signal has caused transfer of the value in TPUm.TCNT to TPUm.TGRB.

[Clearing conditions]

- Activation of the DTC by the TGImB interrupt and clearing of the DTC.MRB.DISEL bit.
- Writing 0 to TGFB after reading its value as 1.

#### **TGFC Flag (Input Capture/Output Compare Flag C)**

This status flag indicates that input capture to TPUm.TGRC or compare match with TPUm.TGRC (m = 0, 3, 6, 9) has occurred.

[Setting conditions]

- When TPUm.TGRC holds the value for comparison in output-compare operations, TPUm.TCNT matches TPUm.TGRC.
- When TPUm.TGRC is serving as an input-capture register, the input-capture signal has caused transfer of the value in TPUm.TCNT to TPUm.TGRC.

[Clearing conditions]

- Activation of the DTC by the TGImC interrupt and clearing of the DTC.MRB.DISEL bit.
- Writing 0 to TGFC after reading its value as 1.

#### **TGFD Flag (Input Capture/Output Compare Flag D)**

This status flag indicates that input capture to TPUm.TGRD or compare match with TPUm.TGRD (m = 0, 3, 6, 9) has occurred.

[Setting conditions]

- When TPUm.TGRD holds the value for comparison in output-compare operations, TPUm.TCNT matches TPUm.TGRD.
- When TPUm.TGRD is serving as an input-capture register, the input-capture signal has caused transfer of the value in TPUm.TCNT to TPUm.TGRD.

[Clearing conditions]

- Activation of the DTC by the TGImD interrupt and clearing of the DTC.MRB.DISEL bit.
- Writing 0 to TGFD after reading its value as 1.

**TCFV Flag (Overflow Flag)**

This status flag indicates an overflow of TPU<sub>m</sub>.TCNT (m = 0 to 11).

[Setting condition]

- Overflow of the value in TPU<sub>m</sub>.TCNT (TCNT counted from FFFFh to 0000h).

[Clearing condition]

- Writing 0 to TCFV after reading its value as 1.

**TCFU Flag (Underflow Flag)**

This status flag indicates an underflow of TPU<sub>m</sub>.TCNT (m = 1, 2, 4, 5, 7, 8, 10, 11).

[Setting condition]

- Underflow of the value in TPU<sub>m</sub>.TCNT (TCNT counted from 0000h to FFFFh).

[Clearing condition]

- Writing 0 to TCFU after reading its value as 1.

**24.2.6 Timer Counter (TCNT)**

Address(es): TPU0.TCNT 0008 8116h, TPU1.TCNT 0008 8126h, TPU2.TCNT 0008 8136h,  
TPU3.TCNT 0008 8146h, TPU4.TCNT 0008 8156h, TPU5.TCNT 0008 8166h,  
TPU6.TCNT 0008 8186h, TPU7.TCNT 0008 8196h, TPU8.TCNT 0008 81A6h,  
TPU9.TCNT 0008 81B6h, TPU10.TCNT 0008 81C6h, TPU11.TCNT 0008 81D6h



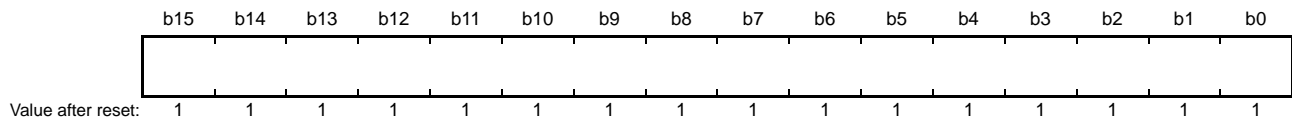
TPU<sub>m</sub>.TCNT is a 16-bit counter that counts the internal clock or external events.

This counter can be read/written in 16-bit units.

This counter is initialized to 0000h by a reset.

### 24.2.7 Timer General Register A (TGRA) Timer General Register B (TGRB) Timer General Register C (TGRC) Timer General Register D (TGRD)

Address(es): TPU0.TGRA 0008 8118h, TPU0.TGRB 0008 811Ah, TPU0.TGRC 0008 811Ch, TPU0.TGRD 0008 811Eh,  
 TPU1.TGRA 0008 8128h, TPU1.TGRB 0008 812Ah,  
 TPU2.TGRA 0008 8138h, TPU2.TGRB 0008 813Ah,  
 TPU3.TGRA 0008 8148h, TPU3.TGRB 0008 814Ah, TPU3.TGRC 0008 814Ch, TPU3.TGRD 0008 814Eh,  
 TPU4.TGRA 0008 8158h, TPU4.TGRB 0008 815Ah,  
 TPU5.TGRA 0008 8168h, TPU5.TGRB 0008 816Ah,  
 TPU6.TGRA 0008 8188h, TPU6.TGRB 0008 818Ah, TPU6.TGRC 0008 818Ch, TPU6.TGRD 0008 818Eh,  
 TPU7.TGRA 0008 8198h, TPU7.TGRB 0008 819Ah,  
 TPU8.TGRA 0008 81A8h, TPU8.TGRB 0008 81AAh,  
 TPU9.TGRA 0008 81B8h, TPU9.TGRB 0008 81BAh, TPU9.TGRC 0008 81BCh, TPU9.TGRD 0008 81BEh,  
 TPU10.TGRA 0008 81C8h, TPU10.TGRB 0008 81CAh,  
 TPU11.TGRA 0008 81D8h, TPU11.TGRB 0008 81DAh



TPU has 32 TGR registers in total, four each for TPU0, TPU3, TPU6, and TPU9 and two each for TPU1, TPU2, TPU4, TPU5, TPU7, TPU8, TPU10, and TPU11.

TPUm.TGRA (m = 0 to 11), TPUm.TGRB (m = 0 to 11), TPUm.TGRC (m = 0, 3, 6, 9), and TPUm.TGRD (m = 0, 3, 6, 9) are 16-bit registers with a dual function as output compare and input capture registers.

These registers can be read/written in 16-bit units.

TPUm.TGRC and TPUm.TGRD can also be specified for operation as buffer registers. Register combinations during buffer operations are TPUm.TGRA—TPUm.TGRC and TPUm.TGRB—TPUm.TGRD.

## 24.2.8 Timer Start Register (TSTR)

- Unit 0 (TPUA.TSTR)

Address(es): TPUA.TSTR 0008 8100h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	CST5	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: TCNT count operation is stopped 1: TCNT performs count operation	R/W
b1	CST1	Counter Start 1		R/W
b2	CST2	Counter Start 2		R/W
b3	CST3	Counter Start 3		R/W
b4	CST4	Counter Start 4		R/W
b5	CST5	Counter Start 5		R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- Unit 1 (TPUB.TSTR)

Address(es): TPUB.TSTR 0008 8170h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	CST1 1	CST1 0	CST9	CST8	CST7	CST6

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	CST6	Counter Start 6	0: TCNT count operation is stopped 1: TCNT performs count operation	R/W
b1	CST7	Counter Start 7		R/W
b2	CST8	Counter Start 8		R/W
b3	CST9	Counter Start 9		R/W
b4	CST10	Counter Start 10		R/W
b5	CST11	Counter Start 11		R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TSTRA starts or stops TCNT count operation for TPU0 to TPU5.

TSTRB starts or stops TCNT count operation for TPU6 to TPU11.

Before setting the operating mode in TPUm.TMDR or setting the TPUm.TCNT count clock in TPUm.TCR, stop the TPUm.TCNT counter.

### CSTn Bits (Counter Start) (n = 0 to 11)

These bits start or stop the TCNT counter.

When the CSTn bit is cleared to 0 with CSTn = 1 and the corresponding TIOCyn pin (y = A to D, n = 0 to 11) specified for output, the counter stops but the output compare output level of the corresponding TIOCyn pin is retained.

If TIORH, TIORL, or TIOR is written to when the CSTn bit is 0, the pin output level will be changed to the set initial output value.

## 24.2.9 Timer Synchronous Register (TSYR)

- Unit 0 (TPUA.TSYR)

Address(es): TPUA.TSYR 0008 8101h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	SYNC5	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 Synchronization 0	0: TCNT operates independently (TCNT presetting/clearing is unrelated to other channels)	R/W
b1	SYNC1	Timer Synchronization 1	1: TCNT performs synchronous operation*1 (TCNT synchronous presetting/synchronous clearing is possible)	R/W
b2	SYNC2	Timer Synchronization 2		R/W
b3	SYNC3	Timer Synchronization 3		R/W
b4	SYNC4	Timer Synchronization 4		R/W
b5	SYNC5	Timer Synchronization 5		R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To set synchronous operation, the SYNCn bit (n = 0 to 5) for at least two channels must be set to 1. To set synchronous clearing, the TCNT clearing source must also be set by the CCLR[2:0] bits in TCR in addition to the SYNCn bit.

- Unit 1 (TPUB.TSYR)

Address(es): TPUB.TSYR 0008 8171h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	SYNC11	SYNC10	SYNC9	SYNC8	SYNC7	SYNC6
Value after reset: 0 0 0 0 0 0 0 0							

Bit	Symbol	Bit Name	Description	R/W
b0	SYNC6	Timer Synchronization 6	0: TCNT operates independently (TCNT presetting/clearing is unrelated to other channels)	R/W
b1	SYNC7	Timer Synchronization 7	1: TCNT performs synchronous operation*1 (TCNT synchronous presetting/synchronous clearing is possible)	R/W
b2	SYNC8	Timer Synchronization 8		R/W
b3	SYNC9	Timer Synchronization 9		R/W
b4	SYNC10	Timer Synchronization 10		R/W
b5	SYNC11	Timer Synchronization 11		R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To set synchronous operation, the SYNCn bit (n = 6 to 11) for at least two channels must be set to 1. To set synchronous clearing, the TCNT clearing source must also be set by the CCLR[2:0] bits in TCR in addition to the SYNCn bit.

TPUA.TSYR selects independent operation or synchronous operation for the TCNT counters of TPU0 to TPU5.  
TPUB.TSYR selects independent operation or synchronous operation for the TCNT counters of TPU6 to TPU11.

### SYNCn Bits (Timer Synchronization) (n = 0 to 11)

These bits select whether the TCNT operation is independent of or synchronized with TCNT of other channels.

When synchronous operation is selected, synchronous presetting of multiple TCNT counters and synchronous clearing through counter clearing on another channel are possible.

### 24.2.10 Noise Filter Control Register (NFCR)

Address(es): TPU0.NFCR 0008 8108h, TPU1.NFCR 0008 8109h, TPU2.NFCR 0008 810Ah, TPU3.NFCR 0008 810Bh, TPU4.NFCR 0008 810Ch, TPU5.NFCR 0008 810Dh, TPU6.NFCR 0008 8178h, TPU7.NFCR 0008 8179h, TPU8.NFCR 0008 817Ah, TPU9.NFCR 0008 817Bh, TPU10.NFCR 0008 817Ch, TPU11.NFCR 0008 817Dh

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 Enable A	0: The noise filter for TIOCAm is disabled. 1: The noise filter for TIOCAm is enabled. (m = 0 to 11)	R/W
b1	NFBEN	Noise Filter Enable B	0: The noise filter for TIOCBm is disabled. 1: The noise filter for TIOCBm is enabled. (m = 0 to 11)	R/W
b2	NFCEN*1	Noise Filter Enable C	0: The noise filter for TIOCCm is disabled. 1: The noise filter for TIOCCm is enabled. (m = 0, 3, 6, 9)	R/W
b3	NFDEN*1	Noise Filter Enable D	0: The noise filter for TIOCDm is disabled. 1: The noise filter for TIOCDm is enabled. (m = 0, 3, 6, 9)	R/W
b5, b4	NFCS[1:0]	Noise Filter Clock Select	00: PCLK/1 01: PCLK/8 10: PCLK/32 11: Clock source that drives counting	R/W
b7, b6	—	Reserved	These bits are read as 0. Writing to these bits is not possible.	R

Note 1. Bits 2 and 3 of TPU1.NFCR, TPU2.NFCR, TPU4.NFCR, TPU5.NFCR, TPU7.NFCR, TPU8.NFCR, TPU10.NFCR, and TPU11.NFCR are reserved. The bits are read as 0. Writing to these bits is not possible.

Only set the TPUm.NFCR registers while the TPUm.TCNT counter is stopped.

#### NFAEN Bit (Noise Filter Enable A)

This bit disables or enables the noise filter for the TIOCAm pin (m = 0 to 11).

Since unexpected edges may be internally generated when the value of NFAEN is changed, select the output compare function in the timer I/O control register before changing the NFAEN value.

#### NFBEN Bit (Noise Filter Enable B)

This bit disables or enables the noise filter for the TIOCBm pin (m = 0 to 11).

Since unexpected edges may be internally generated when the value of NFBEN is changed, select the output compare function in the timer I/O control register before changing the NFBEN value.

#### NFCEN Bit (Noise Filter Enable C)

This bit disables or enables the noise filter for the TIOCCm pin (m = 0, 3, 6, 9).

Since unexpected edges may be internally generated when the value of NFCEN is changed, select the output compare function in the timer I/O control register before changing the NFCEN value.



**NFDEN Bit (Noise Filter Enable D)**

This bit disables or enables the noise filter for the TIOCD<sub>m</sub> pin ( $m = 0, 3, 6, 9$ ).

Since unexpected edges may be internally generated when the value of NFDEN is changed, select the output compare function in the timer I/O control register before changing the NFDEN value.

**NFCS[1:0] Bits (Noise Filter Clock Select)**

These bits select the sampling clock for the noise filter.

When the count source is selected with NFCS[1:0] bits set to 11b, the clock that can be used as sampling clock are the internal clocks other than PCLK/1 specified with the TPSC[2:0] bits and the external clock. To select the PCLK/1 as both the counter clock and the sampling clock, set the NFCS[1:0] bits to 00b.

The input-capture signal is sampled on rising edges of the selected clock signal. If the sampled levels match three times in a row, the given level is passed through as the input-capture signal. If the levels do not match, the existing value is retained.

After setting the NFCS[1:0] bits, wait for two selected sampling periods before setting the input capture function.

## 24.3 Operation

### 24.3.1 Basic Functions

Each channel has a TPUM.TCNT counter and a TPUM.TGRy register (y = A to D).

TCNT is a 16-bit up-counter, which can function as a free-running counter, periodic counter, or event counter.

TGRy can be used as an input capture register or output compare register.

#### (1) Counter Operation

When the CSTj bit (j = 0 to 5) in TPUA.TSTR or the CSTj bit (j = 6 to 11) in TPUB.TSTR is set to 1, the TCNT counter for the corresponding channel starts counting.

#### (a) Example of count operation setting procedure

Figure 24.3 shows an example of the count operation setting procedure.

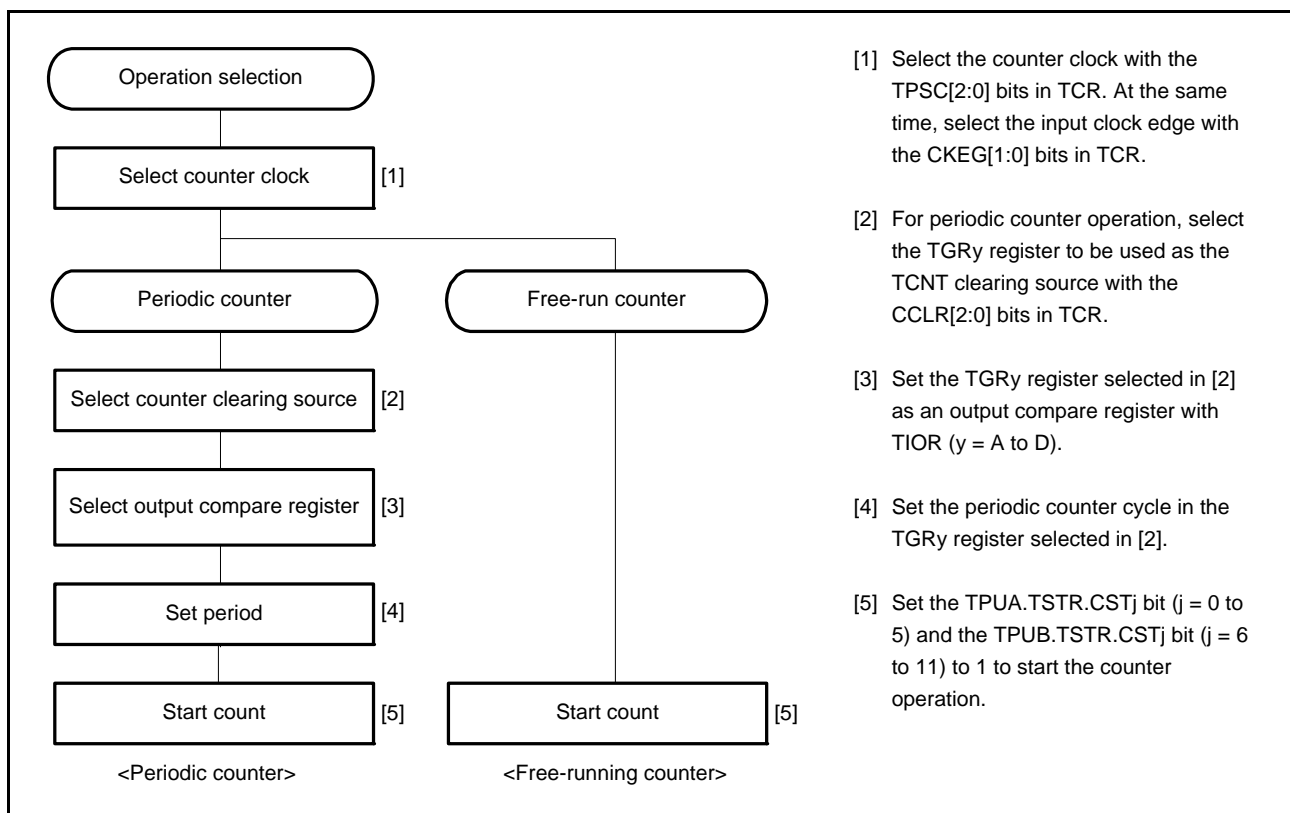


Figure 24.3 Example of Counter Operation Setting Procedure

(b) Free-running count operation and periodic count operation

Immediately after a reset, the TPUm.TCNT counters are all set as free-running counters. When the relevant bit in TPUA.TSTR or TPUB.TSTR is set to 1, the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (changes from FFFFh to 0000h), the TPU requests an interrupt. After an overflow, TCNT restarts counting up from 0000h.

Figure 24.4 shows free-running counter operation.

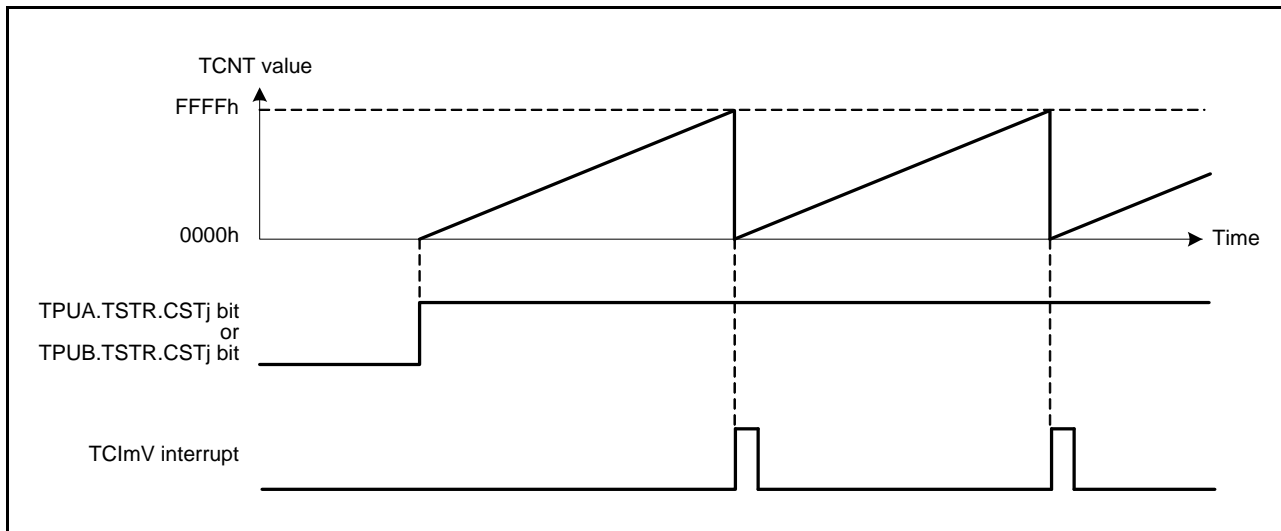


Figure 24.4 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The TPUm.TGRy for setting the period is set as an output compare register, and counter clearing by compare match is selected by the CCLR[2:0] bits in TPUm.TCR. After the settings have been made, TCNT starts count-up operation as a periodic counter when the corresponding bit in TPUA.TSTR or TPUB.TSTR is set to 1. When the count value matches the TGRy value, TCNT is cleared to 0000h.

At this time, the TPU requests an interrupt. After a compare match, TCNT restarts counting up from 0000h.

Figure 24.5 shows periodic counter operation.

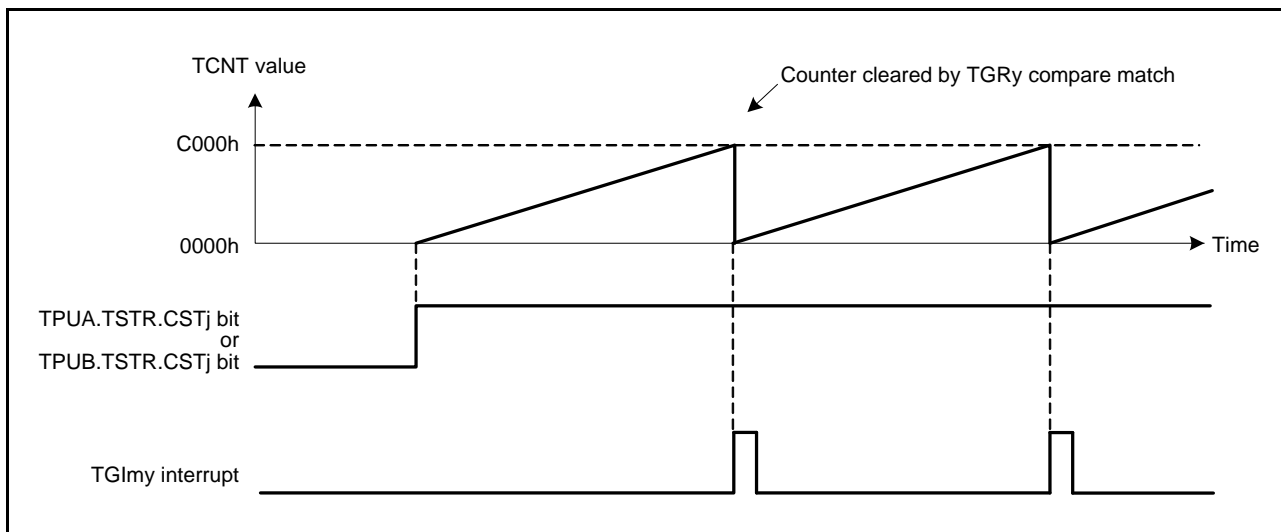


Figure 24.5 Periodic Counter Operation

(2) Waveform Output by Compare Match

The TPU can perform low, high, or toggle output from the corresponding output pin using a compare match.

(a) Example of setting procedure for waveform output by compare match

Figure 24.6 shows an example of the setting procedure for waveform output by a compare match.

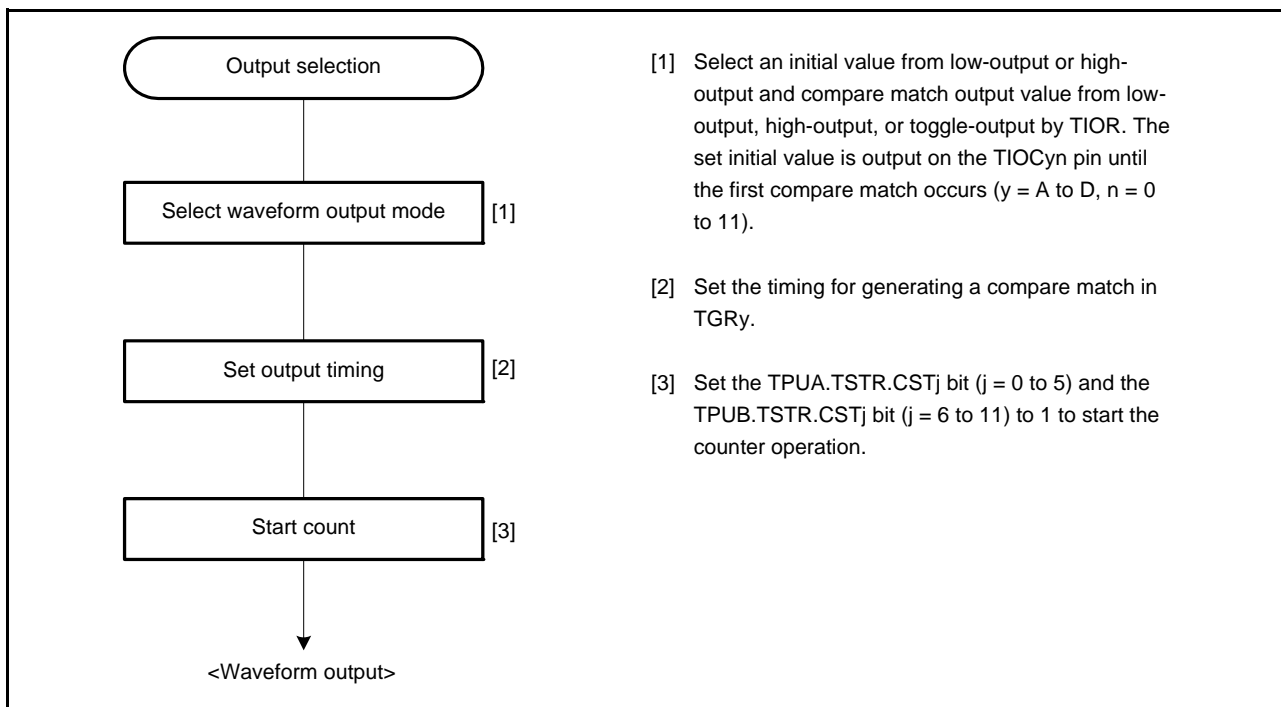


Figure 24.6 Example of Setting Procedure for Waveform Output by Compare Match

(b) Examples of waveform output operation

Figure 24.7 shows an example of low output/high output.

In this example, TPUm.TCNT has been set 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 set level and the pin level match, the pin level does not change.

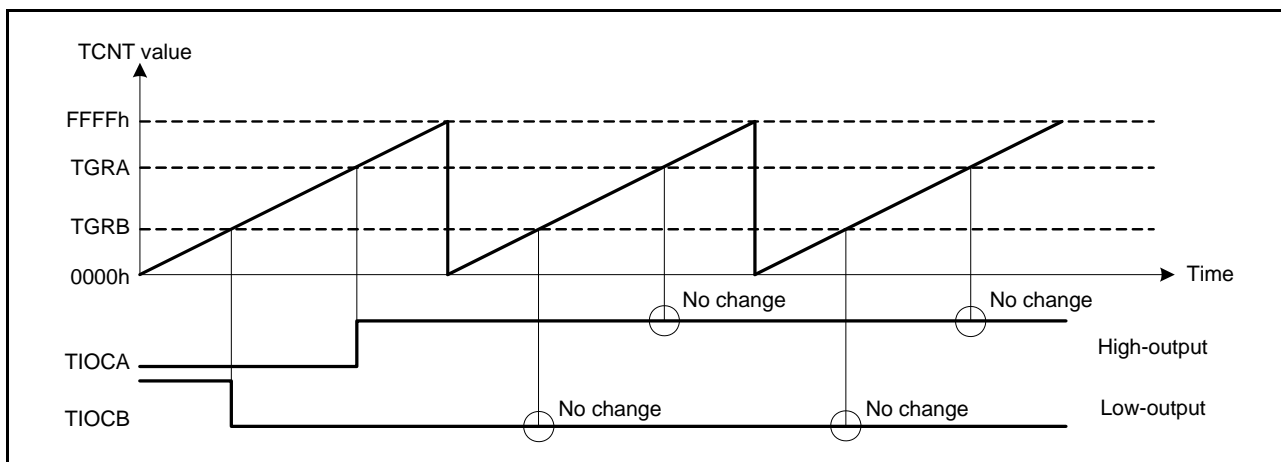


Figure 24.7 Example of Low-Output/High-Output Operation

Figure 24.8 shows an example of toggle output.

In this example, TPUm.TCNT has been set as a periodic counter (with counter clearing performed by compare match B), and settings have been made so that output is toggled by both compare match A and compare match B.

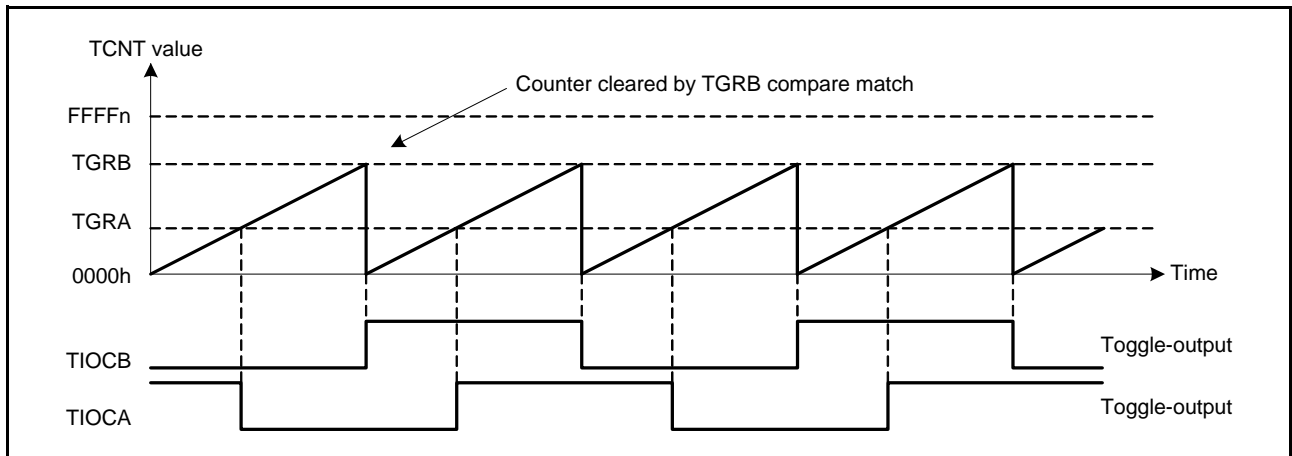


Figure 24.8 Example of Toggle Output Operation

### (3) Input Capture Function

The TPUm.TCNT value can be transferred to TPUm.TGRy on detection of the TIOCyn pin (y = A to D, n = 0 to 11) input edge.

The rising edge, the falling edge, or both edges can be selected as the detection edge. It is also possible to specify the counter input clock or compare match signal of TPU0, TPU1, TPU3, and TPU4 (TPU6, TPU7, TPU9, and TPU10) as the input capture source. Noise filtering can be applied to the input capture input.

- Note 1.
- Even if the counter is halted, an input capture is generated, and flag and interrupt signals are generated.
  - When another channel's counter input clock is used as the input capture input for TPU0 and TPU3 (TPU6 and TPU9), PCLK/1 should not be selected as the counter input clock used for input capture input. Input capture will not be generated if PCLK/1 is selected.

#### (a) Example of setting procedure for input capture operation

Figure 24.9 shows an example of the setting procedure for input capture operation.

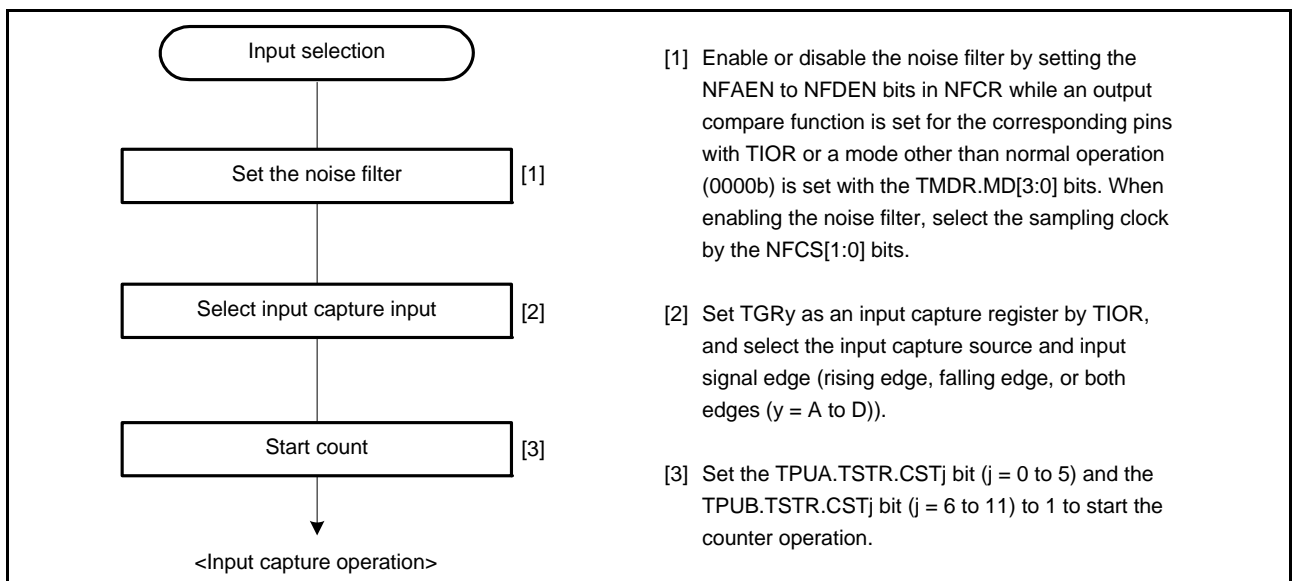


Figure 24.9 Example of Setting Procedure for Input Capture Operation

(b) Example of input capture operation

Figure 24.10 shows an example of input capture operation when the noise filter is stopped.

In this example, both rising and falling edges have been selected as the TIOCA<sub>n</sub> pin input capture input edge, the falling edge has been selected as the TIOCB<sub>n</sub> pin input capture input edge, and counter clearing by TPUm.TGRB input capture has been set for TPUm.TCNT.

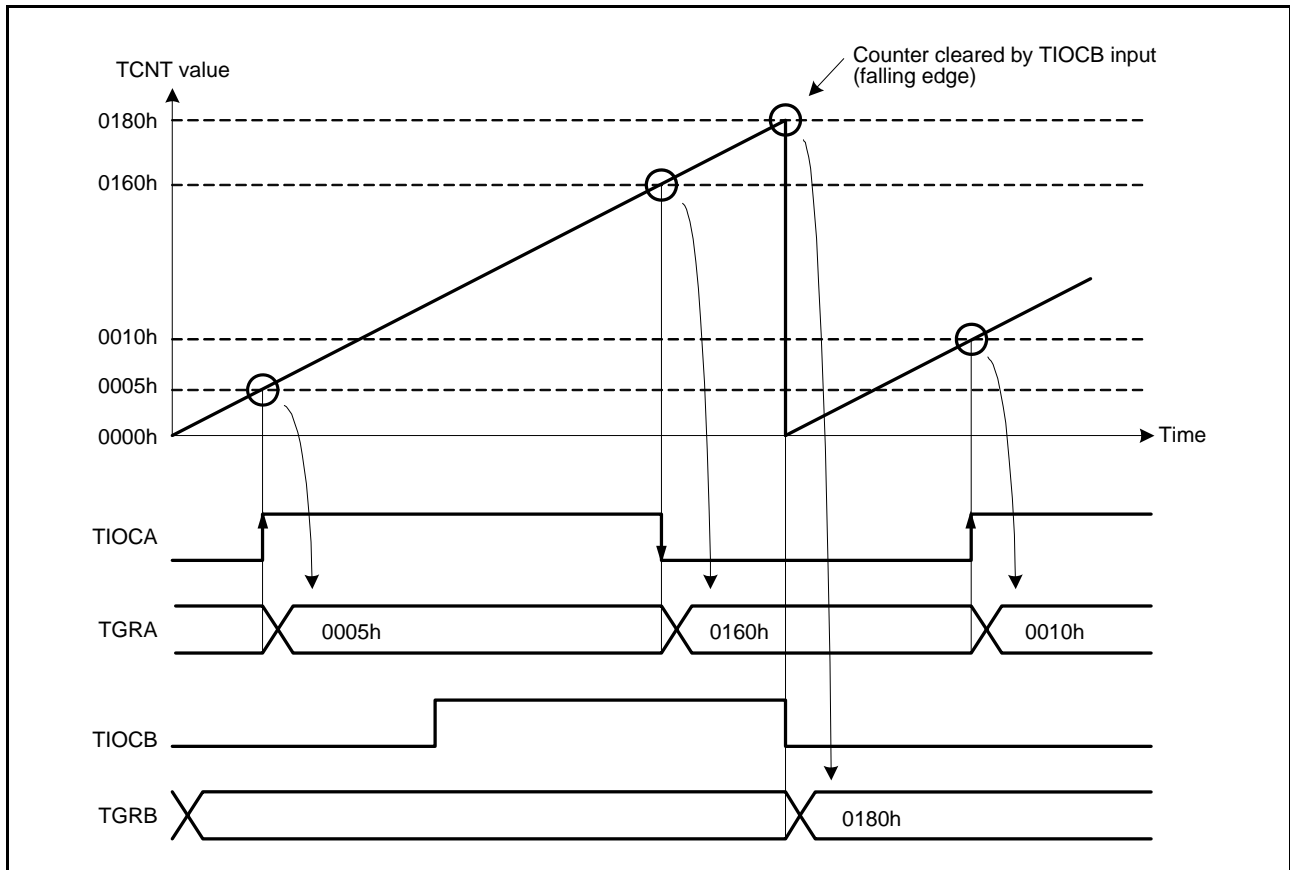


Figure 24.10 Example of Input Capture Operation (with Noise Filter Stopped)

If noise filtering is enabled, input capture operation is performed on the edges of noise-filtered signal after a delay of (minimum sampling interval × 2 + PCLK) due to noise filtering for the input capture input.

### 24.3.2 Synchronous Operation

In synchronous operation, the values in multiple TPUm.TCNT counters can be rewritten simultaneously (synchronous presetting). Also, multiple TCNT counters can be cleared simultaneously (synchronous clearing) by making the appropriate setting in TPUm.TCR.

Synchronous operation enables TPUm.TGRy to be incremented with respect to a single time base.

TPU0 to TPU5 (TPU6 to TPU11) can all be set for synchronous operation.

#### (1) Example of Synchronous Operation Setting Procedure

Figure 24.11 shows an example of the synchronous operation setting procedure.

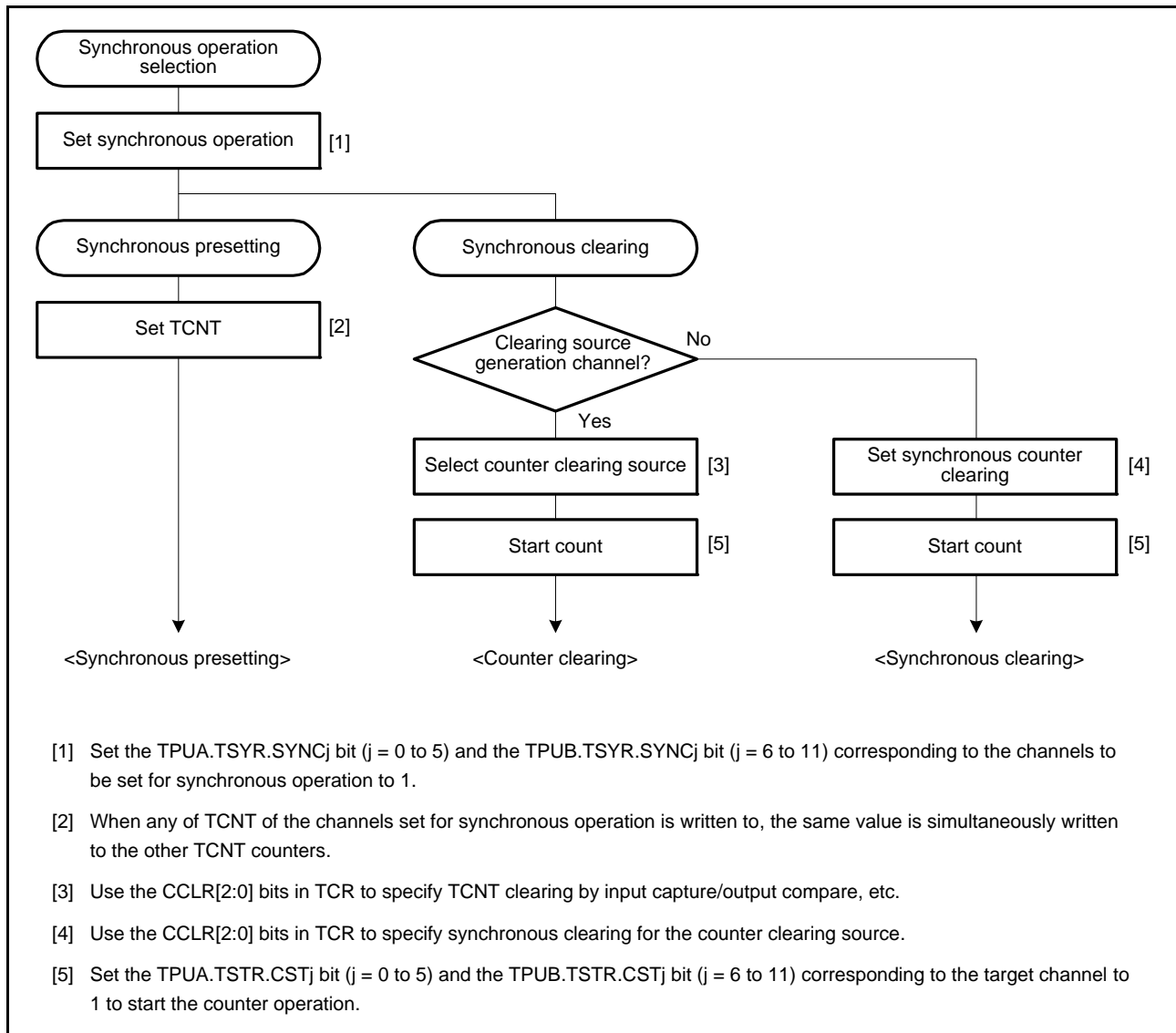


Figure 24.11 Example of Synchronous Operation Setting Procedure

(2) Example of Synchronous Operation

Figure 24.12 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been set for TPU0 to TPU2, TPU0.TGRB compare match has been set as the TPU0 counter clearing source, and synchronous clearing has been set for the TPU1 and TPU2 counter clearing source.

Three-phase PWM waveforms are output from pins TIOCA0, TIOCA1, and TIOCA2. At this time, synchronous presetting and synchronous clearing by TPU0.TGRB compare match are performed for TPUm.TCNT of TPU0 to TPU2, and the data set in TPU0.TGRB is used as the PWM cycle.

For details on PWM modes, see section 24.3.5, PWM Modes.

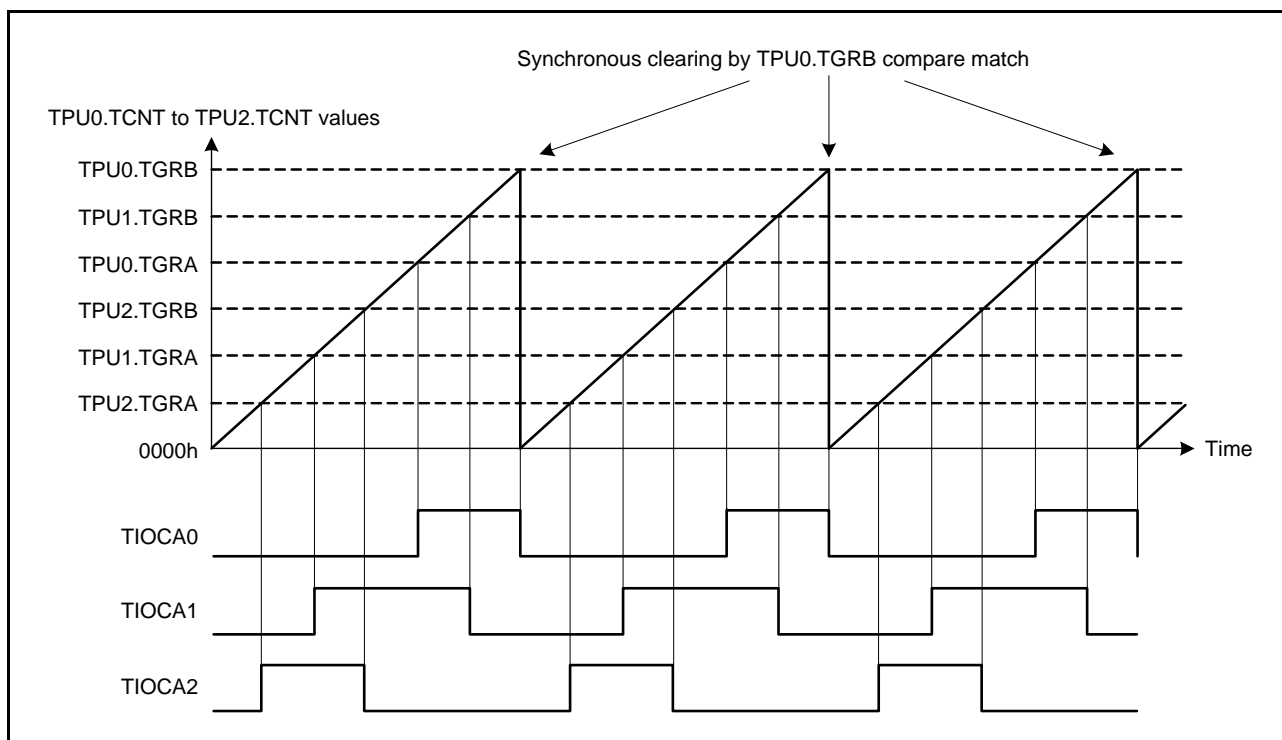


Figure 24.12 Example of Synchronous Operation



### 24.3.3 Buffer Operation

Buffer operation, provided for TPU0 and TPU3 (TPU6 and TPU9), enables TPUm.TGRC and TPUm.TGRD to be used as buffer registers.

Buffer operation differs depending on whether TPUm.TGRy has been set as an input capture register or a compare match register.

Table 24.22 lists the register combinations used in buffer operation.

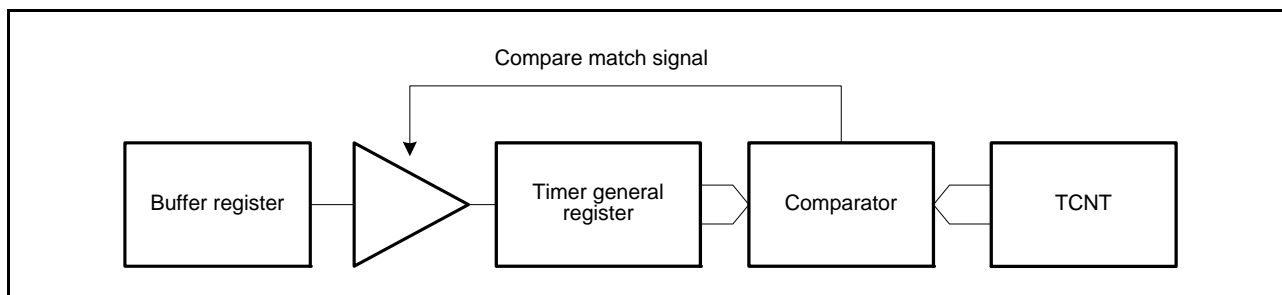
**Table 24.22 Register Combinations in Buffer Operation**

Unit	Channel	Timer General Register	Buffer Register
0	TPU0	TPU0.TGRA	TPU0.TGRC
		TPU0.TGRB	TPU0.TGRD
	TPU3	TPU3.TGRA	TPU3.TGRC
		TPU3.TGRB	TPU3.TGRD
1	TPU6	TPU6.TGRA	TPU6.TGRC
		TPU6.TGRB	TPU6.TGRD
	TPU9	TPU9.TGRA	TPU9.TGRC
		TPU9.TGRB	TPU9.TGRD

- When TPUm.TGRy 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 shown in Figure 24.13.

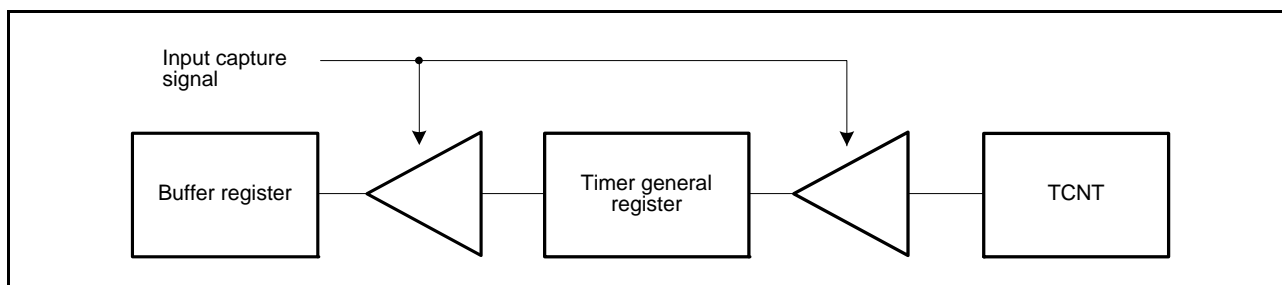


**Figure 24.13 Compare Match Buffer Operation**

- When TPUm.TGRy is an input capture register

When input capture occurs, the value in TPUm.TCNT is transferred to TGRy and the value previously held in TGRy is simultaneously transferred to the buffer register.

This operation is shown in Figure 24.14.



**Figure 24.14 Input Capture Buffer Operation**

(1) Example of Buffer Operation Setting Procedure

Figure 24.15 shows an example of the buffer operation setting procedure.

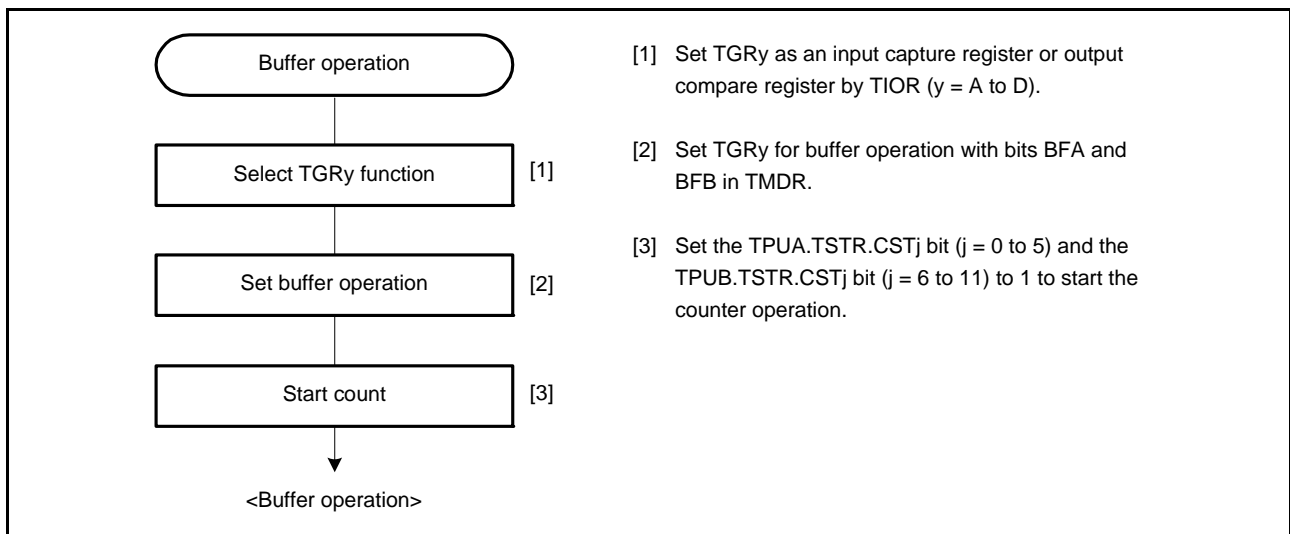


Figure 24.15 Example of Buffer Operation Setting Procedure

(2) Examples of Buffer Operation

(a) When TPUm.TGRy is an output compare register

Figure 24.16 shows an operation example in which PWM mode 1 has been set for TPU0, and buffer operation has been set for TPU0.TGRA and TPU0.TGRC. The settings used in this example are TPU0.TCNT clearing by compare match B, high output at compare match A, and low output at compare match B.

As buffer operation has been set, when compare match A occurs, the output changes and the TPU0.TGRC value is simultaneously transferred to TPU0.TGRA. This operation is repeated each time compare match A occurs.

For details on PWM modes, see section 24.3.5, PWM Modes.

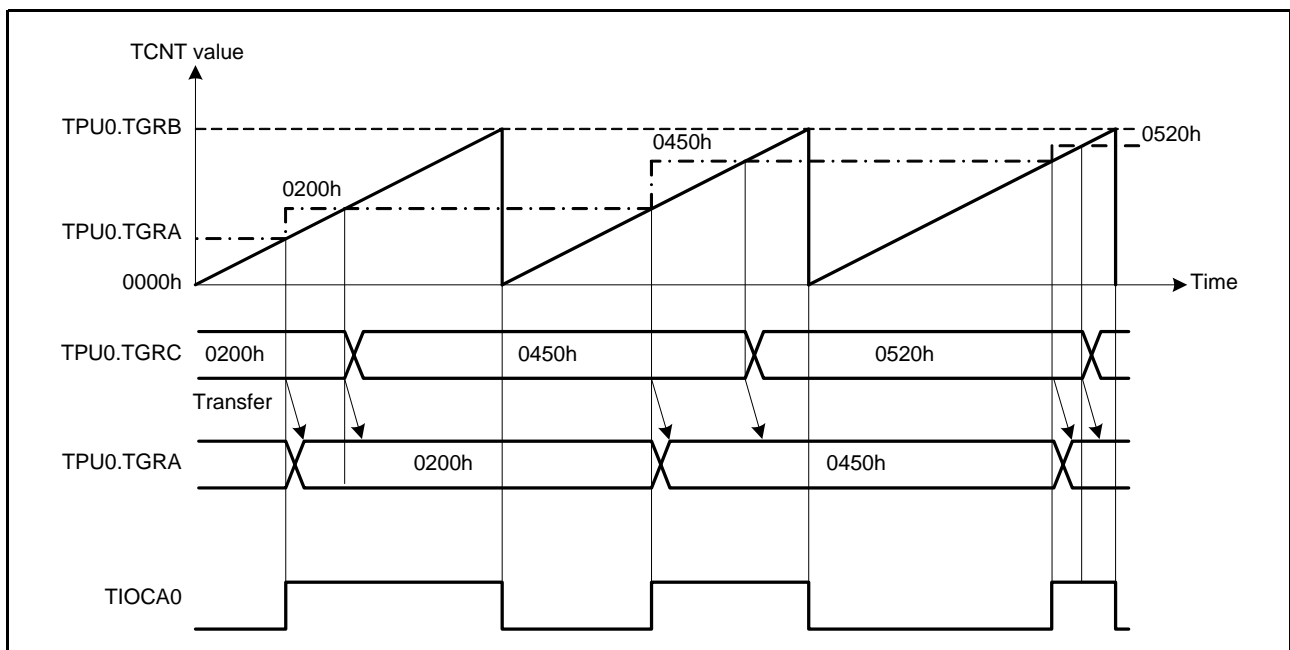


Figure 24.16 Example of Buffer Operation (1)

(b) When TPUM.TGRy is an input capture register

Figure 24.17 shows an operation example in which TPUM.TGRA has been set as an input capture register, and buffer operation has been set for the TGRA register and TPUM.TGRC.

Counter clearing by TGRA input capture has been set for TPUM.TCNT, and both rising and falling edges have been selected as the TIOCA pin input capture input edge.

As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

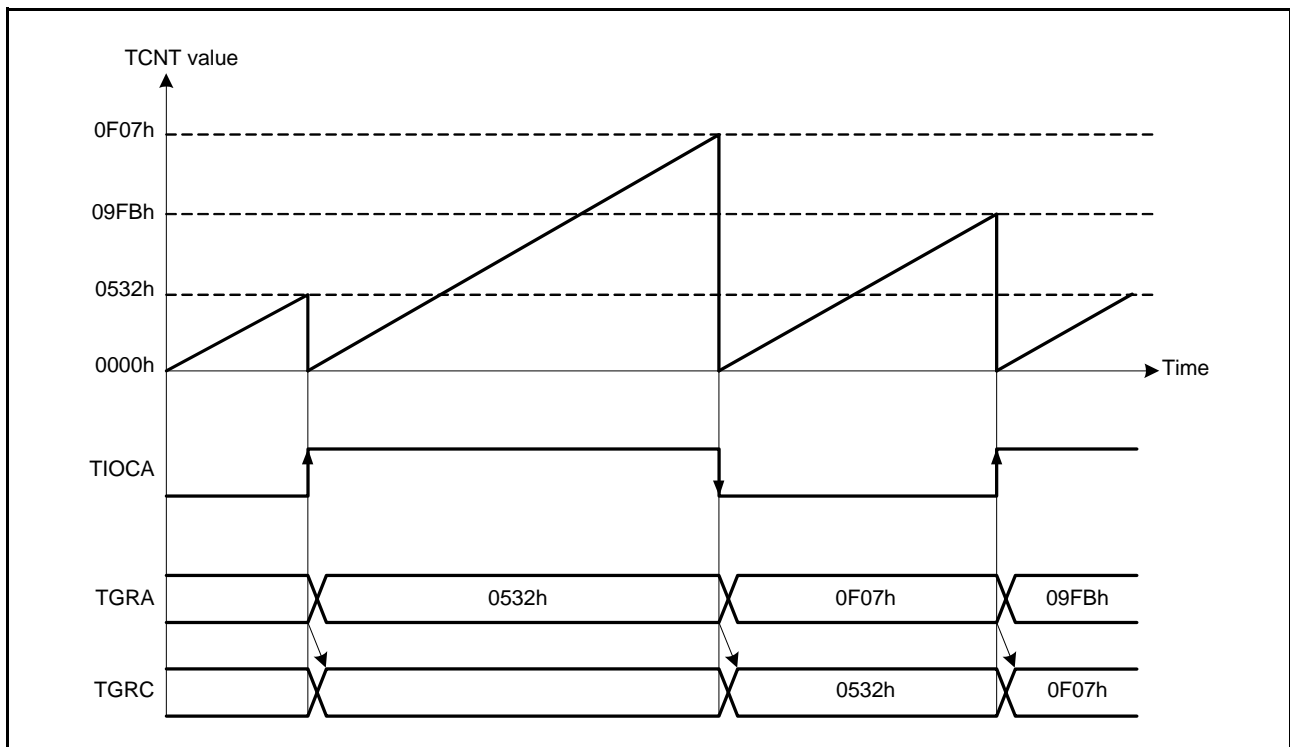


Figure 24.17 Example of Buffer Operation (2)

### 24.3.4 Cascaded Operation

In cascaded operation, two 16-bit counters for different channels are used together as a 32-bit counter.

In the case of unit 0, this function works by counting the TPU1 (TPU4) counter clock at overflow/underflow of TPU2.TCNT (TPU5.TCNT) as set by the TPSC[2:0] bits in TPU1.TCR (TPSC[2:0] bits in TPU4.TCR).

In the case of unit 1, this function works by counting the TPU7 (TPU10) counter clock at overflow/underflow of TPU8.TCNT (TPU11.TCNT) as set by the TPSC[2:0] bits in TPU7.TCR (TPSC[2:0] bits in TPU10.TCR).

Underflow occurs only when the lower 16-bit TPU<sub>m</sub>.TCNT is in phase counting mode.

Table 24.23 lists the register combinations used in cascaded operation.

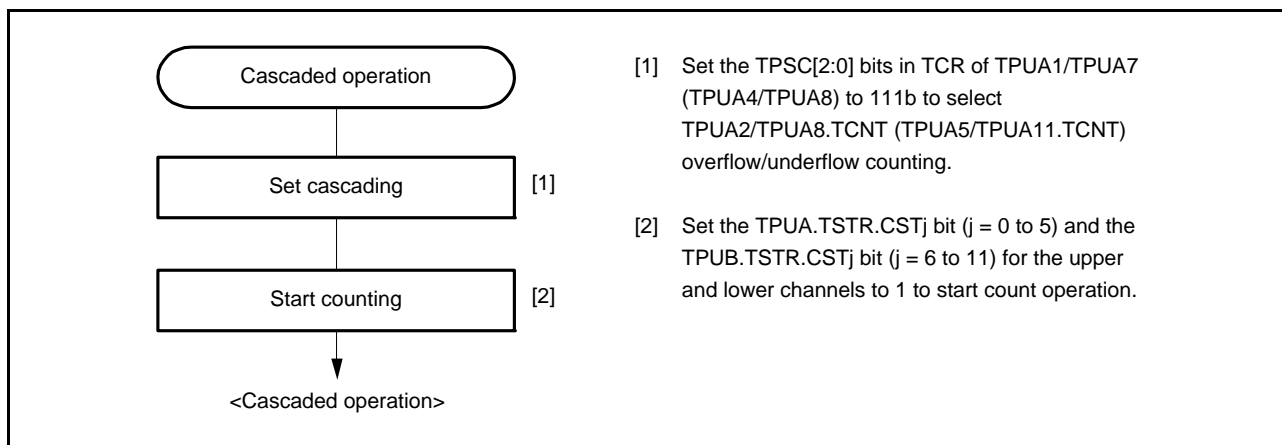
Note 1. When phase counting mode is set for TPU1 or TPU4 (TPU7 or TPU10), the counter clock setting is invalid and the counter operates independently in phase counting mode.

**Table 24.23 Cascaded Combinations**

Unit	Combination	Upper 16 Bits	Lower 16 Bits
0	TPU1 and TPU2	TPU1.TCNT	TPU2.TCNT
	TPU4 and TPU5	TPU4.TCNT	TPU5.TCNT
1	TPU7 and TPU8	TPU7.TCNT	TPU8.TCNT
	TPU10 and TPU11	TPU10.TCNT	TPU11.TCNT

#### (1) Example of Cascaded Operation Setting Procedure

Figure 24.18 shows an example of the setting procedure for cascaded operation.



**Figure 24.18 Cascaded Operation Setting Procedure**

(2) Examples of Cascaded Operation

Figure 24.19 shows the operation when counting upon TPU2.TCNT overflow/underflow has been set for TPU1.TCNT, TPU1.TGRA and TPU2.TGRA have been set as input capture registers, and the rising edge of the TIOCA1 and TIOCA2 pins has been selected.

When a rising edge is input to the TIOCA1 and TIOCA2 pins simultaneously, the upper 16 bits of the 32-bit data are transferred to TPU1.TGRA, and the lower 16 bits to TPU2.TGRA.

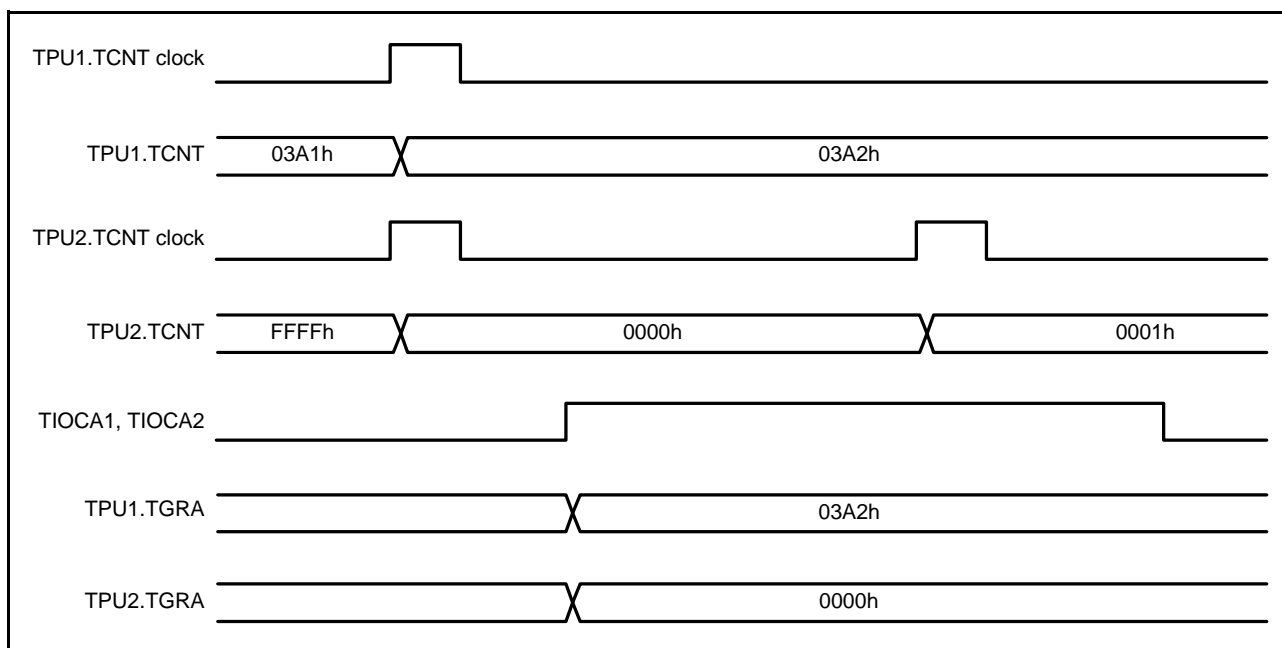


Figure 24.19 Example of Cascaded Operation (1)

Figure 24.20 shows the operation when counting upon TPU2.TCNT overflow/underflow has been set for TPU1.TCNT, and phase counting mode has been specified for TPU2.

TPU1.TCNT is incremented by TPU2.TCNT overflow and decremented by TPU2.TCNT underflow.

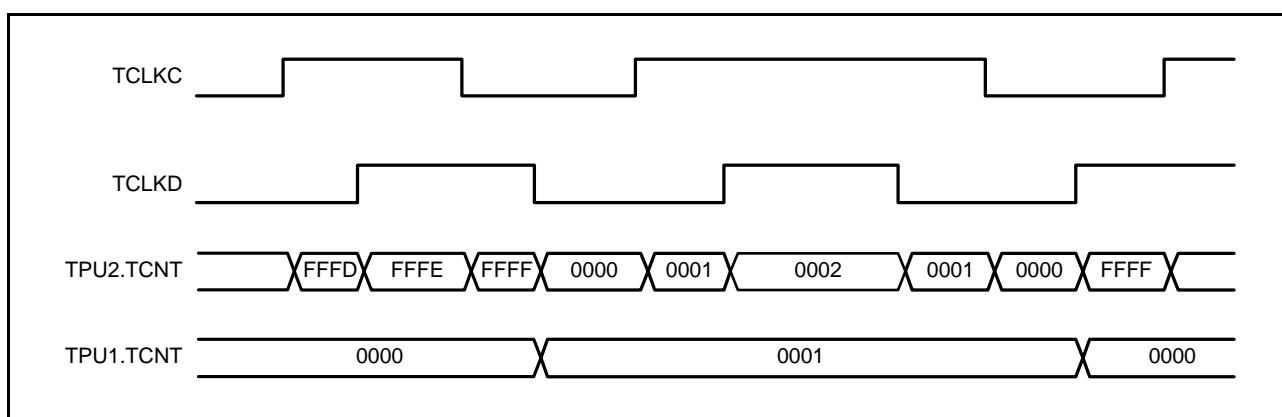


Figure 24.20 Example of Cascaded Operation (2)

### 24.3.5 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. low-, high-, or toggle-output can be selected as the output level in response to compare match of each TPUm.TGRy.

Settings of TGRy registers can output a PWM waveform in the range of 0% to 100% duty cycle.

Specifying TGRy compare match as the counter clearing source enables the cycle to be set in that register. All channels can be set for PWM mode independently. Synchronous operation is also possible.

There are two PWM modes, as described below.

#### 1. PWM mode 1

PWM waveform is generated from the TIOCA<sub>n</sub> and TIOCC<sub>n</sub> pins by pairing TPUm.TGRA with TPUm.TGRB and TPUm.TGRC with TPUm.TGRD. The outputs specified by the IOA[3:0] bits in TPUm.TIOR(H) and IOC[3:0] bits in TPUm.TIORL are output from the TIOCA<sub>n</sub> and TIOCC<sub>n</sub> pins at compare matches A and C, respectively. The outputs specified by the IOB[3:0] bits in TPUm.TIOR(H) and IOD[3:0] bits in TPUm.TIORL are output from the TIOCA<sub>n</sub> and TIOCC<sub>n</sub> pins at compare matches B and D, respectively. The initial output value is the value set in TGRA or TGRC. If the set values of paired TGRy registers are identical, the output value does not change even when a compare match occurs.

In PWM mode 1, a maximum 8-phase PWM output is possible.

#### 2. PWM mode 2

PWM waveform is generated by using one TPUm.TGRy as the cycle register and the others as duty cycle registers. The output specified in TPUm.TIORH, TPUm.TIORL, or TPUm.TIOR is performed by compare matches. Upon counter clearing by a synchronous register compare match, the output value of each pin is the initial value set in TIORH, TIORL, or TIOR. If the set values of the cycle register and duty cycle register are identical, the output value does not change even when a compare match occurs.

In PWM mode 2, a maximum 15-phase PWM waveform is possible by combined use with synchronous operation.

The correspondence between PWM output pins and registers is listed in Table 24.24.

**Table 24.24 PWM Output Registers and Output Pins**

Unit	Channel	Register	Output Pin	
			PWM Mode 1	PWM Mode 2
0	TPU0	TPU0.TGRA	TIOCA0	TIOCA0
		TPU0.TGRB		TIOCB0
		TPU0.TGRC	TIOCC0	TIOCC0
		TPU0.TGRD		TIOCD0
	TPU1	TPU1.TGRA	TIOCA1	TIOCA1
		TPU1.TGRB		TIOCB1
	TPU2	TPU2.TGRA	TIOCA2	TIOCA2
		TPU2.TGRB		TIOCB2
	TPU3	TPU3.TGRA	TIOCA3	TIOCA3
		TPU3.TGRB		TIOCB3
		TPU3.TGRC	TIOCC3	TIOCC3
		TPU3.TGRD		TIOCD3
	TPU4	TPU4.TGRA	TIOCA4	TIOCA4
		TPU4.TGRB		TIOCB4
	TPU5	TPU5.TGRA	TIOCA5	TIOCA5
		TPU5.TGRB		TIOCB5
1	TPU6	TPU6.TGRA	TIOCA6	TIOCA6
		TPU6.TGRB		TIOCB6
		TPU6.TGRC	TIOCC6	TIOCC6
		TPU6.TGRD		TIOCD6
	TPU7	TPU7.TGRA	TIOCA7	TIOCA7
		TPU7.TGRB		TIOCB7
	TPU8	TPU8.TGRA	TIOCA8	TIOCA8
		TPU8.TGRB		TIOCB8
	TPU9	TPU9.TGRA	TIOCA9	TIOCA9
		TPU9.TGRB		TIOCB9
		TPU9.TGRC	TIOCC9	TIOCC9
		TPU9.TGRD		TIOCD9
	TPU10	TPU10.TGRA	TIOCA10	TIOCA10
		TPU10.TGRB		TIOCB10
	TPU11	TPU11.TGRA	TIOCA11	TIOCA11
		TPU11.TGRB		TIOCB11

Note: • In PWM mode 2, PWM waveform output is not possible for the TPU<sub>m</sub>.TGR<sub>y</sub> register in which the cycle is set.

(1) Example of PWM Mode Setting Procedure

Figure 24.21 shows an example of the PWM mode setting procedure.

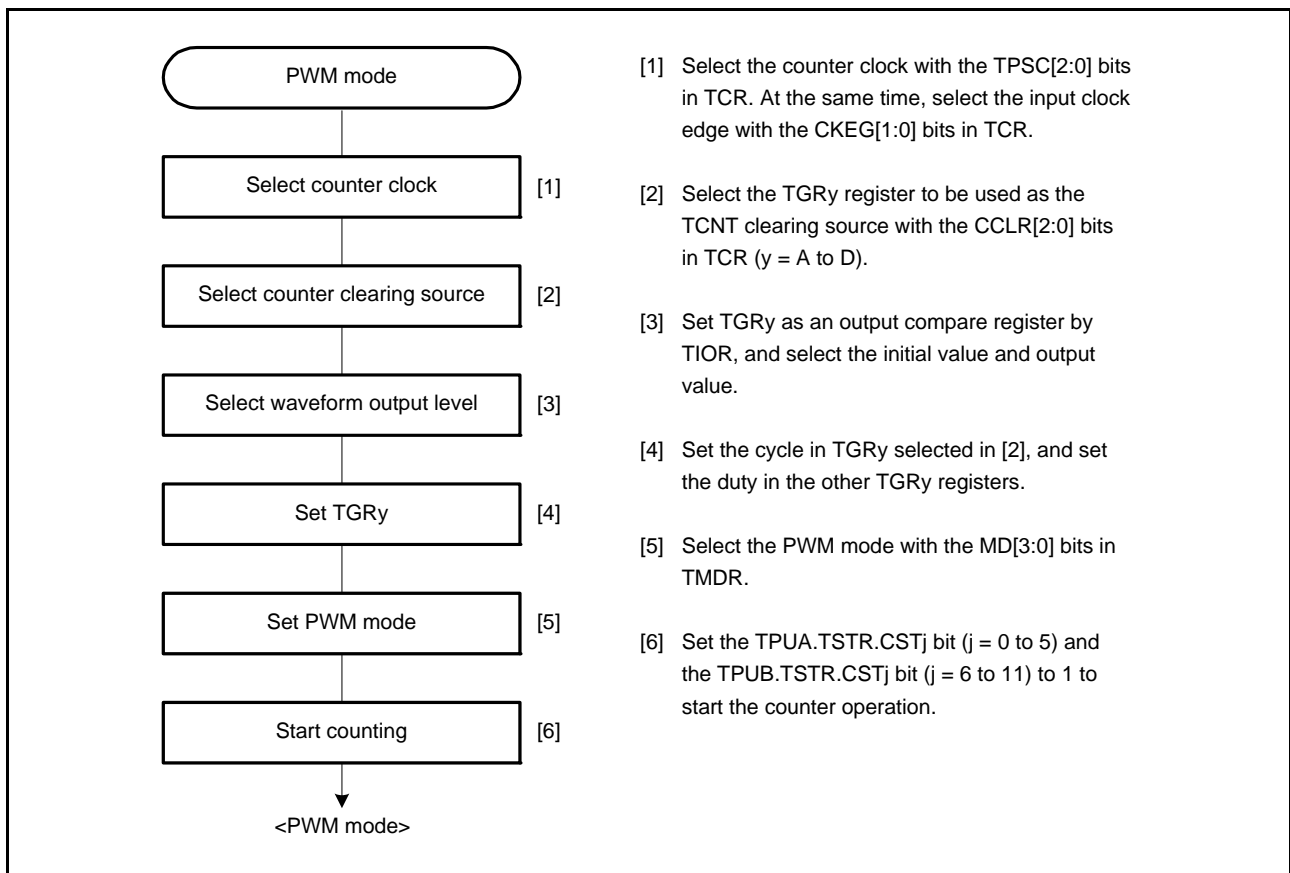


Figure 24.21 Example of PWM Mode Setting Procedure

(2) Examples of PWM Mode Operation

Figure 24.22 shows an example of PWM mode 1 operation.

In this example, TPUm.TGRA compare match is set as the TPUm.TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TPUm.TGRB output value.

In this case, the value set in TGRA is used as the cycle, and the value set in TGRB is used as the duty cycle.

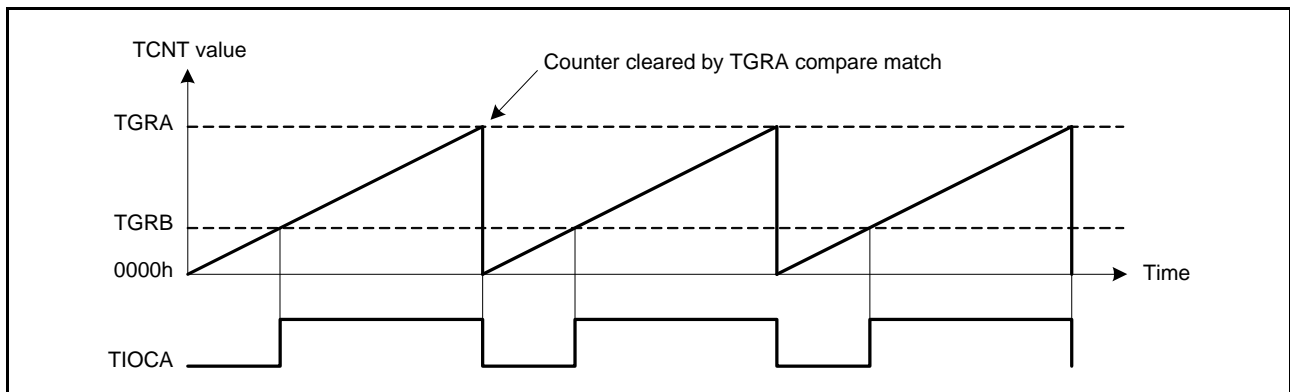


Figure 24.22 Example of PWM Mode Operation (1)



Figure 24.23 shows an example of PWM mode 2 operation.

In this example, synchronous operation is specified for TPU0 and TPU1, TPU1.TGRB compare match is set as the TPUm.TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TPUm.TGRy registers (TPU0.TGRA to TPU0.TGRD and TPU1.TGRA), to output a 5-phase PWM waveform.

In this case, the value set in TPU1.TGRB is used as the cycle, and the values set in the other TGRy registers are used as the duty cycle.

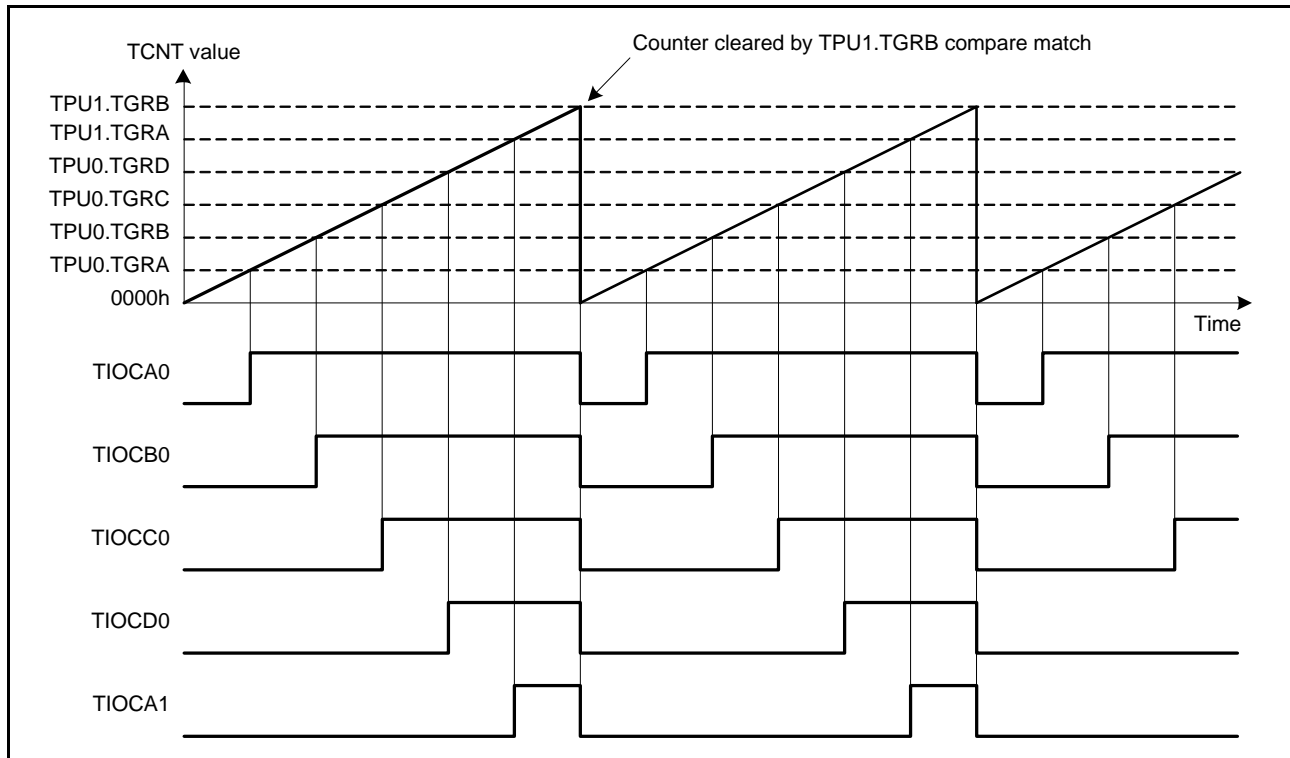


Figure 24.23 Example of PWM Mode Operation (2)

Figure 24.24 shows examples of PWM waveform output with 0% duty cycle and 100% duty cycle in PWM mode.

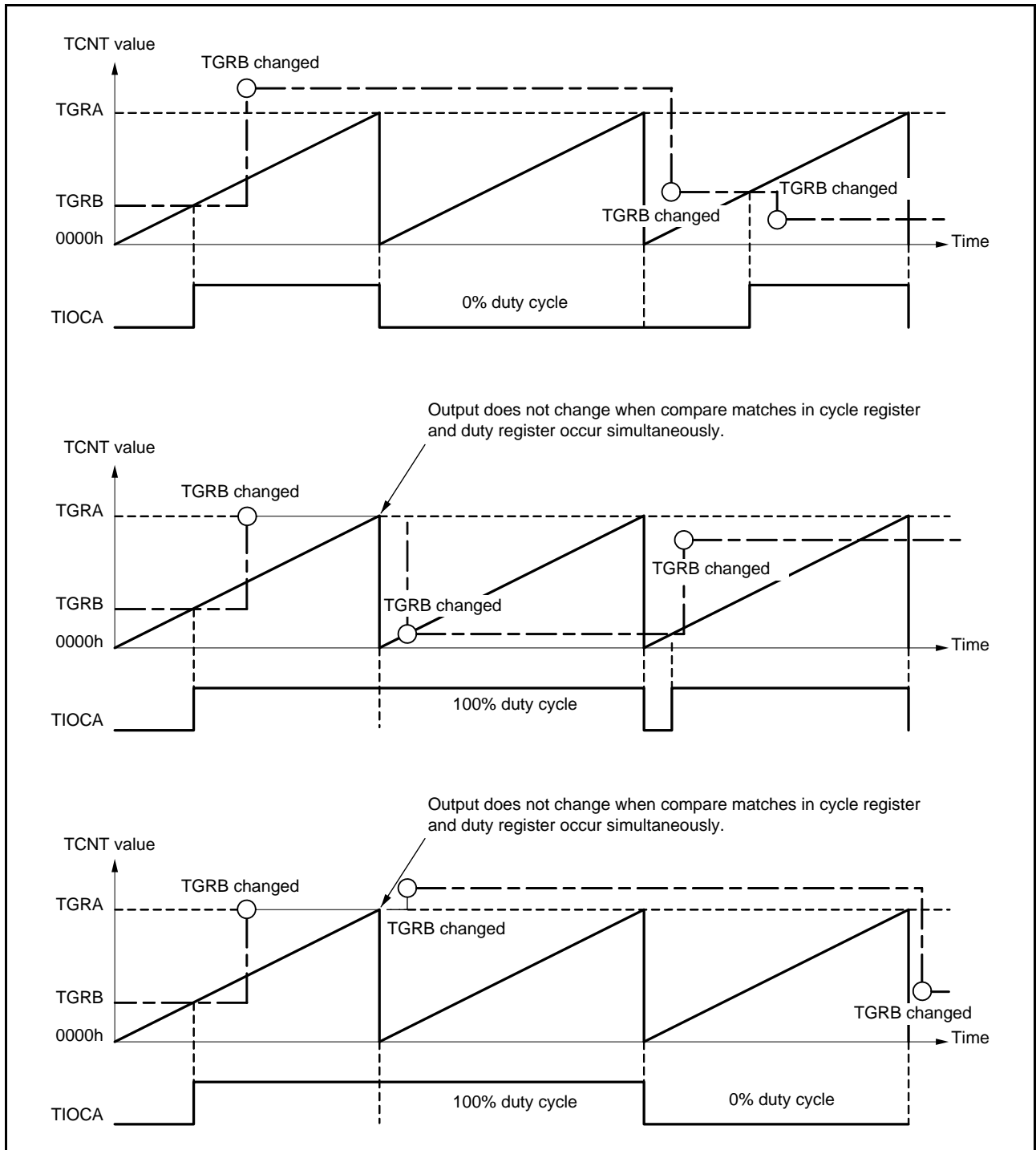


Figure 24.24 Example of PWM Mode Operation (3)

### 24.3.6 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected by the settings for channels 1, 2, 4, and 5 (unit 0) and channels 7, 8, 10, and 11 (unit 1), and TPUm.TCNT is incremented/decremented accordingly. When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up-/down-counter regardless of the setting of the TPSC[2:0] bits and CKEG[1:0] bits in TPUm.TCR. However, the lower 2 bits of the CCLR[2:0] bits in TPUm.TCR and the functions of TPUm.TIORH, TPUm.TIORL, TPUm.TIOR, TPUm.TIER, and TPUm.TGRy are valid, and therefore input capture/compare match and interrupt functions are available.

This can be used for two-phase encoder pulse input.

When an overflow occurs while TCNT is counting up, a TCIV interrupt request is generated; when an underflow occurs while TCNT is counting down, a TCIU interrupt request is generated. The TCFD bit in TPUm.TSR is the count direction flag. Reading the TCFD flag provides an indication of whether TCNT is counting up or down.

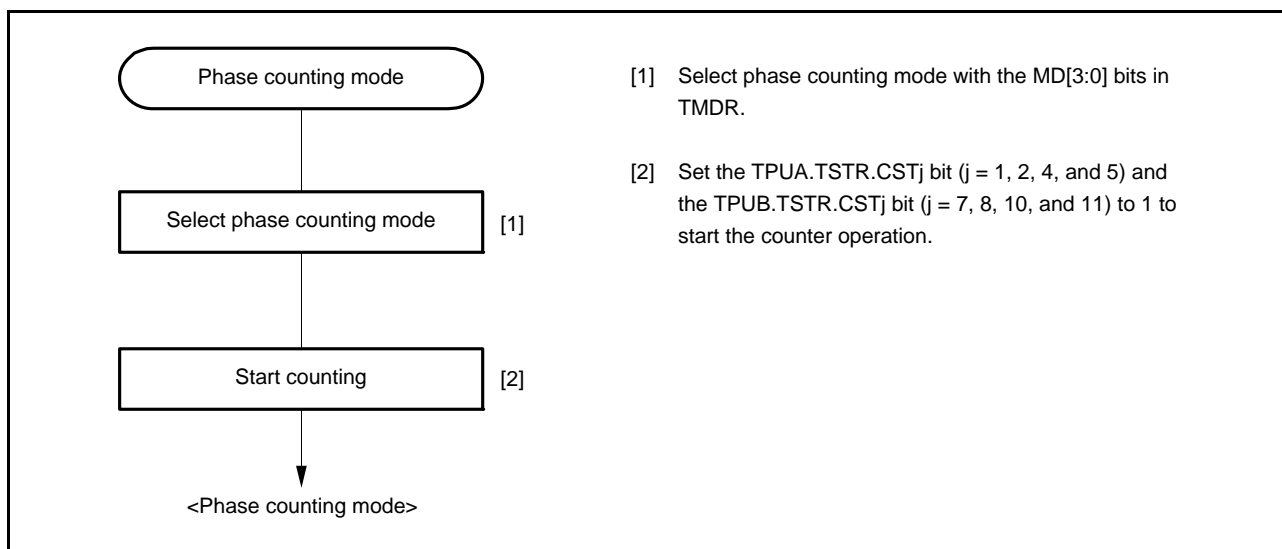
Table 24.25 lists the correspondence between external clock pins and channels.

**Table 24.25 Clock Input Pins in Phase Counting Mode**

Unit	Channel	External Clock Pins	
		A-Phase	B-Phase
0	When TPU1 or TPU5 is set to phase counting mode	TCLKA	TCLKB
	When TPU2 or TPU4 is set to phase counting mode	TCLKC	TCLKD
1	When TPU7 or TPU11 is set to phase counting mode	TCLKE	TCLKF
	When TPU8 or TPU10 is set to phase counting mode	TCLKG	TCLKH

#### (1) Example of Phase Counting Mode Setting Procedure

Figure 24.25 shows an example of the phase counting mode setting procedure.



**Figure 24.25 Example of Phase Counting Mode Setting Procedure**

(2) Examples of Phase Counting Mode Operation

In phase counting mode, TPUm.TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

(a) Phase counting mode 1

Figure 24.26 shows an example of phase counting mode 1 operation, and Table 24.26 lists the TPUm.TCNT up/down-count conditions.

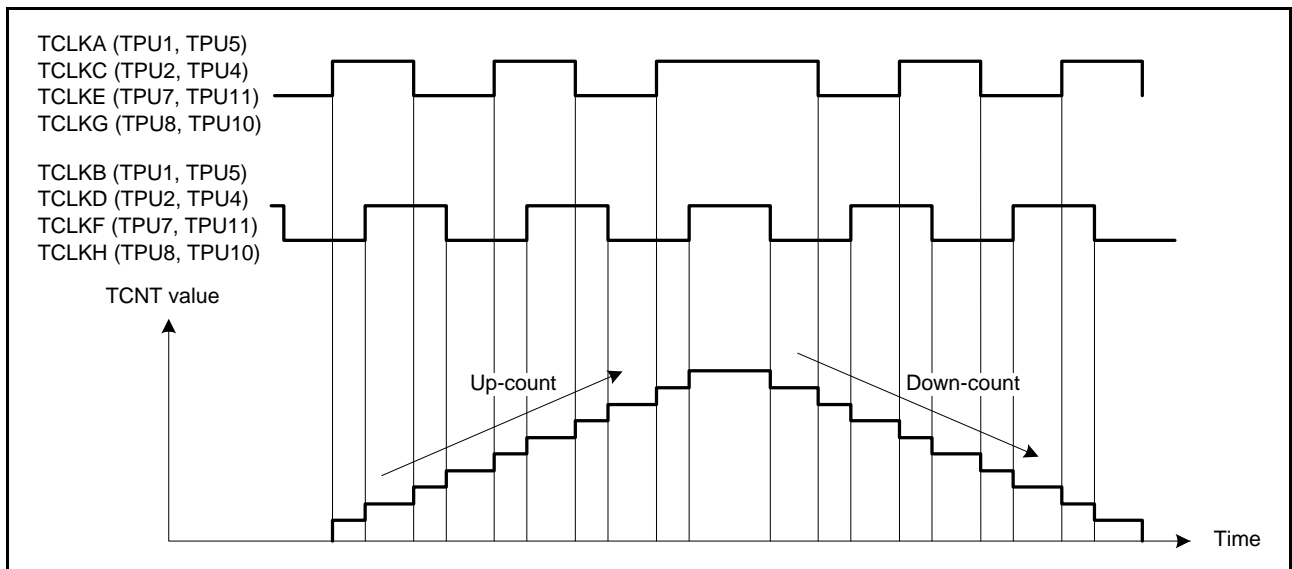


Figure 24.26 Example of Phase Counting Mode 1 Operation

Table 24.26 Up-/Down-Count Conditions in Phase Counting Mode 1

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4) TCLKE (TPU7, TPU11) TCLKG (TPU8, TPU10)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4) TCLKF (TPU7, TPU11) TCLKH (TPU8, TPU10)	Operation
High		Up-count
Low		
	Low	
	High	
High		Down-count
Low		
	High	
	Low	

: Rising edge  
 : Falling edge

(b) Phase counting mode 2

Figure 24.27 shows an example of phase counting mode 2 operation, and Table 24.27 lists the TPUm.TCNT up-/down-count conditions.

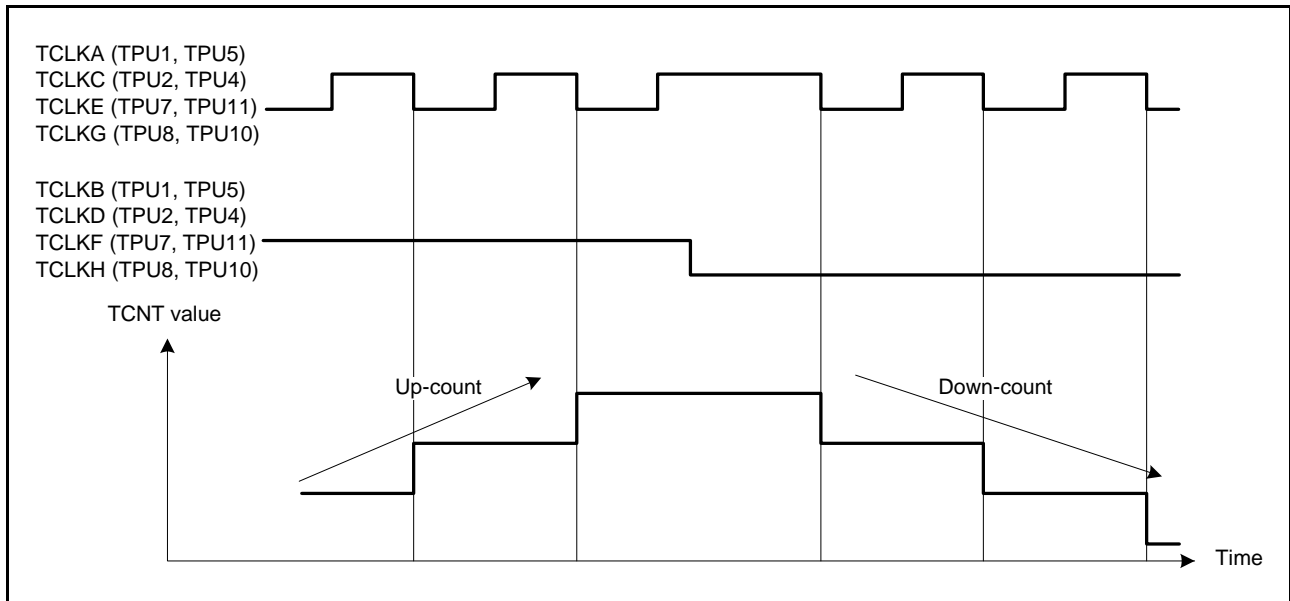


Figure 24.27 Example of Phase Counting Mode 2 Operation

Table 24.27 Up-/Down-Count Conditions in Phase Counting Mode 2

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4) TCLKE (TPU7, TPU11) TCLKG (TPU8, TPU10)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4) TCLKF (TPU7, TPU11) TCLKH (TPU8, TPU10)	Operation
High	↑	Don't care
Low	↓	Don't care
↑	Low	Don't care
↓	High	Up-count
High	↓	Don't care
Low	↑	Don't care
↑	High	Don't care
↓	Low	Down-count

↑ : Rising edge  
↓ : Falling edge

(c) Phase counting mode 3

Figure 24.28 shows an example of phase counting mode 3 operation, and Table 24.28 lists the TPUm.TCNT up-/down-count conditions.

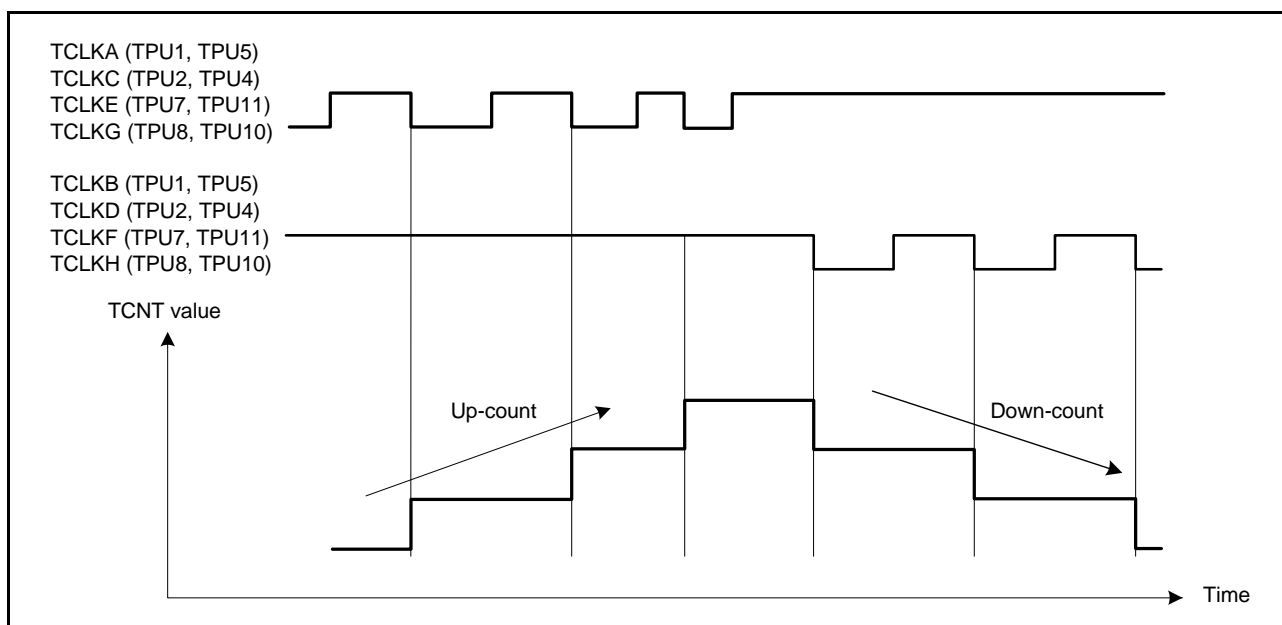


Figure 24.28 Example of Phase Counting Mode 3 Operation

Table 24.28 Up-/Down-Count Conditions in Phase Counting Mode 3

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4) TCLKE (TPU7, TPU11) TCLKG (TPU8, TPU10)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4) TCLKF (TPU7, TPU11) TCLKH (TPU8, TPU10)	Operation
High		Don't care
Low		Don't care
	Low	Don't care
	High	Up-count
High		Down-count
Low		Don't care
	High	Don't care
	Low	Don't care

: Rising edge  
 : Falling edge

(d) Phase counting mode 4

Figure 24.29 shows an example of phase counting mode 4 operation, and Table 24.29 lists the TPUm.TCNT up-/down-count conditions.

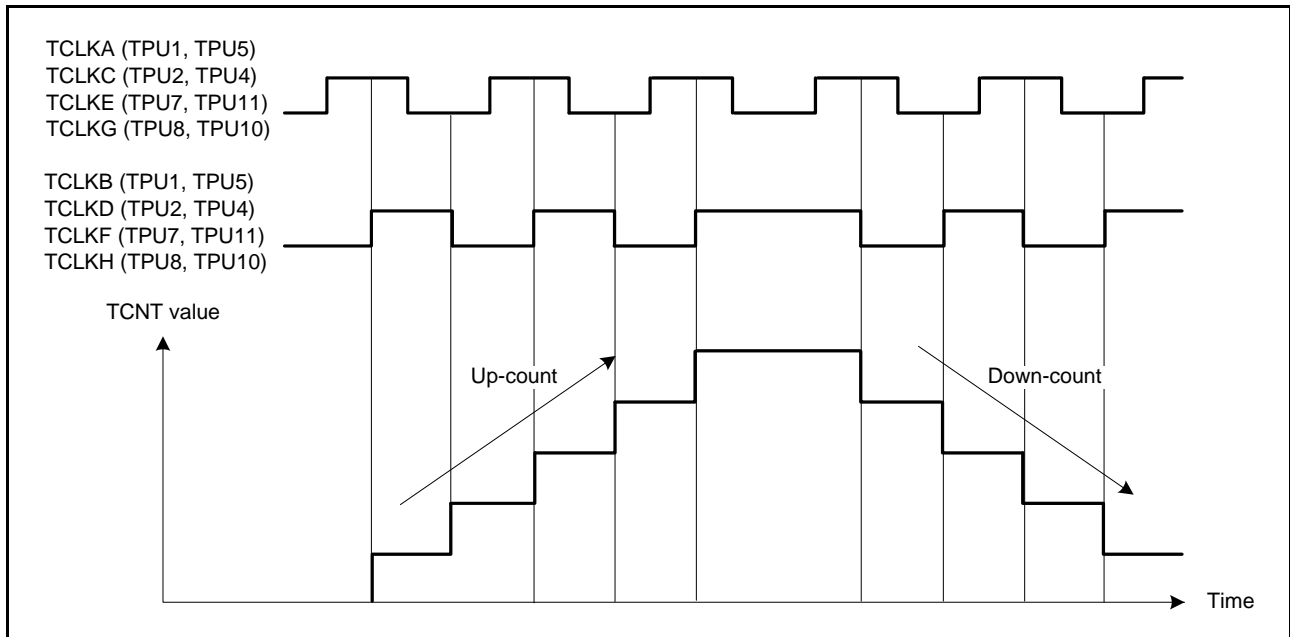


Figure 24.29 Example of Phase Counting Mode 4 Operation

Table 24.29 Up-/Down-Count Conditions in Phase Counting Mode 4

TCLKA (TPU1, TPU5) TCLKC (TPU2, TPU4) TCLKE (TPU7, TPU11) TCLKG (TPU8, TPU10)	TCLKB (TPU1, TPU5) TCLKD (TPU2, TPU4) TCLKF (TPU7, TPU11) TCLKH (TPU8, TPU10)	Operation
High		Up-count
Low		Up-count
	Low	Don't care
	High	Don't care
High		Down-count
Low		Down-count
	High	Don't care
	Low	Don't care

: Rising edge  
 : Falling edge

### 24.3.6.1 Phase Counting Mode Application Example

Figure 24.30 shows an example in which phase counting mode is set for TPU1, and TPU1 is coupled with TPU0 to input servo motor 2-phase encoder pulses in order to detect the position or speed.

TPU1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to the TCLKA and TCLKB pins.

TPU0 operates with TPU0.TCNT counter clearing by TPU0.TGRC compare match; TPU0.TGRA and TPU0.TGRC are used for the compare match function and are set with the speed control cycle and position control cycle. TPU0.TGRB is used for input capture, with TPU0.TGRB and TPU0.TGRD operating in buffer mode. The TPU1 counter input clock is specified as the TPU0.TGRB input capture source, and the pulse width of 2-phase encoder 4-multiplication pulses is detected.

TPU1.TGRA and TPU1.TGRB for TPU1 are specified for input capture, TPU0.TGRA and TPU0.TGRC compare matches are selected as the input capture source, and the up-/down-counter values for the control cycles are stored. This procedure enables accurate position/speed detection to be achieved.

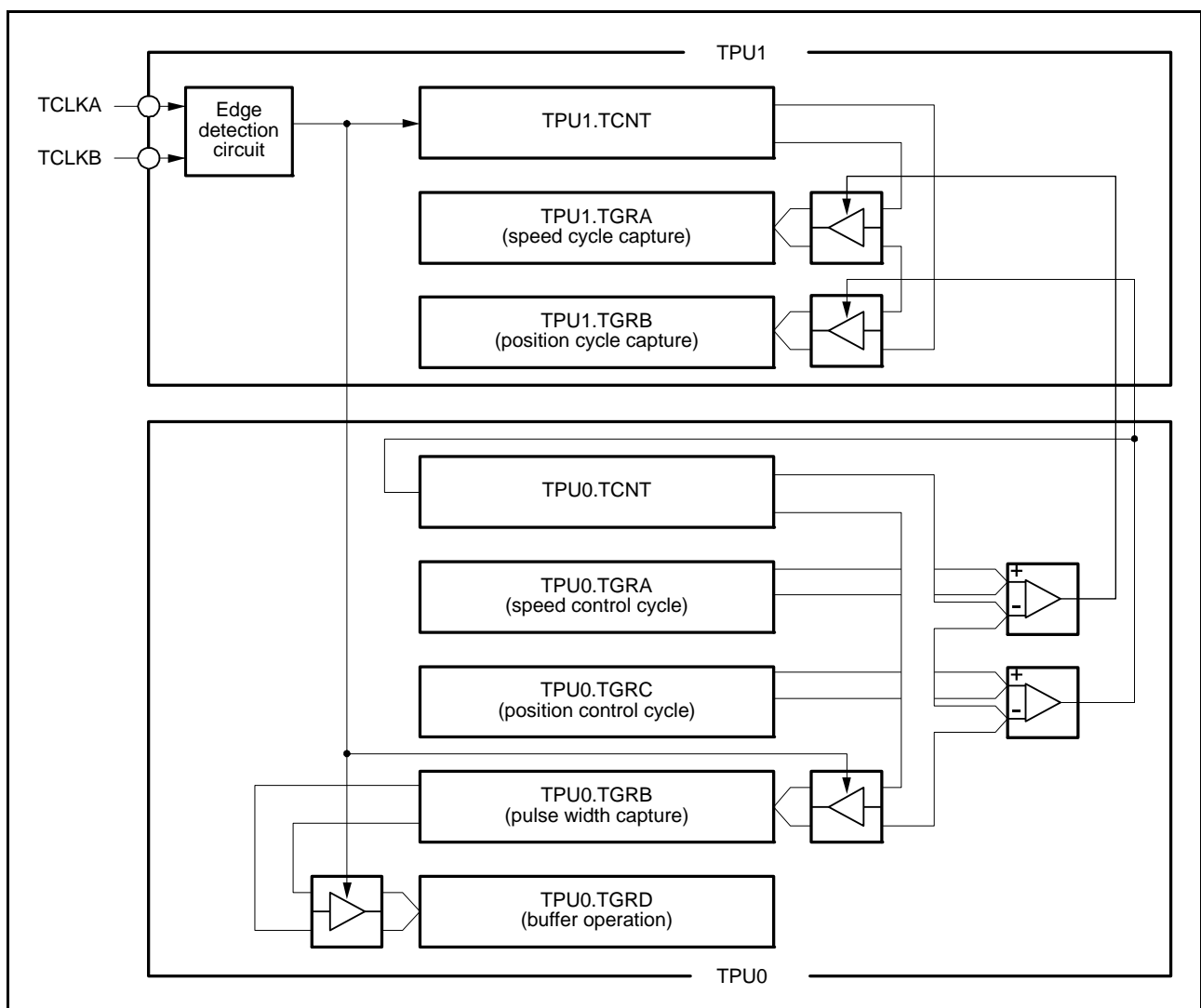


Figure 24.30 Phase Counting Mode Application Example



### 24.3.7 Noise Filters

Each pin for use in input capture by TPU is equipped with a noise filter. The noise filter samples the level on the pin three times at the selected sampling interval, conveys the level to the internal circuits if the samples match, and continues to convey that level until the other level is sampled from the pins three times in a row. The noise filter function can be enabled or disabled for each pin. Furthermore, sampling clock settings can be made for each channel.

Figure 24.31 is a timing chart for the noise filter.

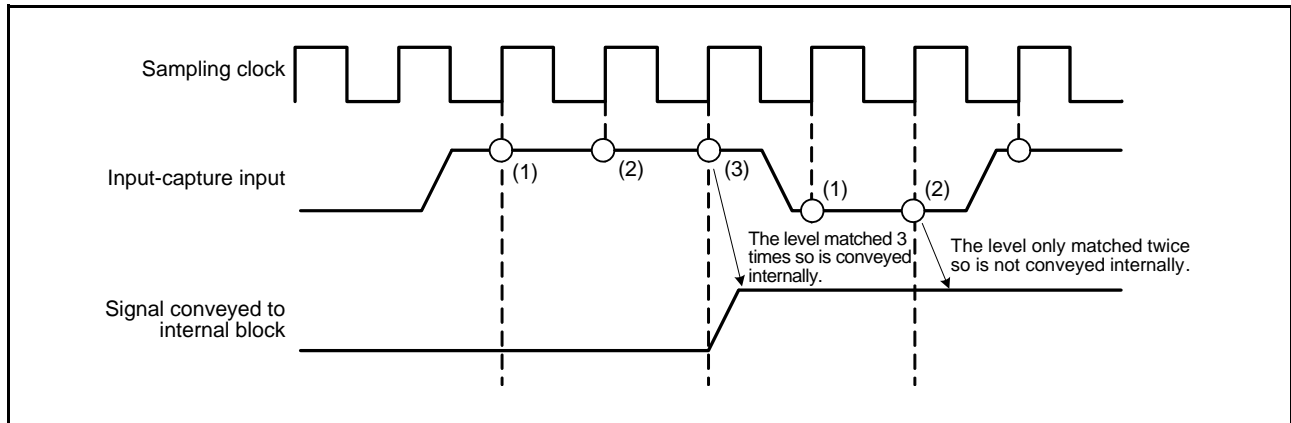


Figure 24.31 Timing Chart for the Noise Filter

## 24.4 Interrupt Sources

There are three kinds of TPU interrupt sources: TPU<sub>m</sub>.TGR<sub>y</sub> input capture/compare match, TPU<sub>m</sub>.TCNT overflow, and TPU<sub>m</sub>.TCNT underflow.

Relative channel priority levels can be changed by the interrupt controller, but the priority within a channel is fixed. For details, see section 15, Interrupt controller (ICUb).

Table 24.30 lists the TPU interrupt sources.

**Table 24.30 TPU Interrupt Sources (1/2)**

Unit	Channel	Name	Interrupt Source	DTC Activation	DMAC Activation
0	TPU0	TGI0A	TPU0.TGRA input capture/compare match	Possible	Possible
		TGI0B	TPU0.TGRB input capture/compare match	Possible	Not possible
		TGI0C	TPU0.TGRC input capture/compare match	Possible	Not possible
		TGI0D	TPU0.TGRD input capture/compare match	Possible	Not possible
		TCI0V	TPU0.TCNT overflow	Not possible	Not possible
TPU1	TPU1	TGI1A	TPU1.TGRA input capture/compare match	Possible	Possible
		TGI1B	TPU1.TGRB input capture/compare match	Possible	Not possible
		TCI1V	TPU1.TCNT overflow	Not possible	Not possible
		TCI1U	TPU1.TCNT underflow	Not possible	Not possible
TPU2	TPU2	TGI2A	TPU2.TGRA input capture/compare match	Possible	Possible
		TGI2B	TPU2.TGRB input capture/compare match	Possible	Not possible
		TCI2V	TPU2.TCNT overflow	Not possible	Not possible
		TCI2U	TPU2.TCNT underflow	Not possible	Not possible
TPU3	TPU3	TGI3A	TPU3.TGRA input capture/compare match	Possible	Possible
		TGI3B	TPU3.TGRB input capture/compare match	Possible	Not possible
		TGI3C	TPU3.TGRC input capture/compare match	Possible	Not possible
		TGI3D	TPU3.TGRD input capture/compare match	Possible	Not possible
		TCI3V	TPU3.TCNT overflow	Not possible	Not possible
TPU4	TPU4	TGI4A	TPU4.TGRA input capture/compare match	Possible	Possible
		TGI4B	TPU4.TGRB input capture/compare match	Possible	Not possible
		TCI4V	TPU4.TCNT overflow	Not possible	Not possible
		TCI4U	TPU4.TCNT underflow	Not possible	Not possible
TPU5	TPU5	TGI5A	TPU5.TGRA input capture/compare match	Possible	Possible
		TGI5B	TPU5.TGRB input capture/compare match	Possible	Not possible
		TCI5V	TPU5.TCNT overflow	Not possible	Not possible
		TCI5U	TPU5.TCNT underflow	Not possible	Not possible

**Table 24.30 TPU Interrupt Sources (2/2)**

Unit	Channel	Name	Interrupt Source	DTC Activation	DMAC Activation
1	TPU6	TGI6A	TPU6.TGRA input capture/compare match	Possible	Possible
		TGI6B	TPU6.TGRB input capture/compare match	Possible	Not possible
		TGI6C	TPU6.TGRC input capture/compare match	Possible	Not possible
		TGI6D	TPU6.TGRD input capture/compare match	Possible	Not possible
		TCI6V	TPU6.TCNT overflow	Not possible	Not possible
	TPU7	TGI7A	TPU7.TGRA input capture/compare match	Possible	Possible
		TGI7B	TPU7.TGRB input capture/compare match	Possible	Not possible
		TCI7V	TPU7.TCNT overflow	Not possible	Not possible
		TCI7U	TPU7.TCNT underflow	Not possible	Not possible
	TPU8	TGI8A	TPU8.TGRA input capture/compare match	Possible	Possible
		TGI8B	TPU8.TGRB input capture/compare match	Possible	Not possible
		TCI8V	TPU8.TCNT overflow	Not possible	Not possible
		TCI8U	TPU8.TCNT underflow	Not possible	Not possible
	TPU9	TGI9A	TPU9.TGRA input capture/compare match	Possible	Possible
		TGI9B	TPU9.TGRB input capture/compare match	Possible	Not possible
		TGI9C	TPU9.TGRC input capture/compare match	Possible	Not possible
		TGI9D	TPU9.TGRD input capture/compare match	Possible	Not possible
		TCI9V	TPU9.TCNT overflow	Not possible	Not possible
	TPU10	TGI10A	TPU10.TGRA input capture/compare match	Possible	Possible
		TGI10B	TPU10.TGRB input capture/compare match	Possible	Not possible
TCI10V		TPU10.TCNT overflow	Not possible	Not possible	
TCI10U		TPU10.TCNT underflow	Not possible	Not possible	
TPU11	TGI11A	TPU11.TGRA input capture/compare match	Possible	Possible	
	TGI11B	TPU11.TGRB input capture/compare match	Possible	Not possible	
	TCI11V	TPU11.TCNT overflow	Not possible	Not possible	
	TCI11U	TPU11.TCNT underflow	Not possible	Not possible	

Note: • This table lists the initial state immediately after a reset. The relative channel priority levels can be changed by the interrupt controller.

### (1) Input Capture/Compare Match Interrupt

An interrupt is requested when the TGIE<sub>y</sub> bit ( $y = A, B, C, D$ ) in TPUM.TIER is set to 1 by the occurrence of a TPUM.TGR<sub>y</sub> input capture/compare match on a channel. The TPU has 32 input capture/compare match interrupts, four each for TPU0, TPU3, TPU6 and TPU9, and two each for TPU1, TPU2, TPU4, TPU5, TPU7, TPU8, TPU10, and TPU11.

### (2) Overflow Interrupt

An interrupt is requested when the TCIEV bit in TPUM.TIER is set to 1 by the occurrence of a TPUM.TCNT overflow on a channel. The TPU has twelve overflow interrupts, one for each channel.

### (3) Underflow Interrupt

An interrupt is requested when the TCIEU bit in TPUM.TIER is set to 1 by the occurrence of a TPUM.TCNT underflow on a channel. The TPU has eight underflow interrupts, one each for TPU1, TPU2, TPU4, TPU5, TPU7, TPU8, TPU10, and TPU11.

## 24.5 DTC Activation

The DTC can be activated by the TPUM.TGR<sub>y</sub> input capture/compare match interrupt of each channel. For details, see section 19, Data Transfer Controller (DTCa).

A total of 32 input capture/compare match interrupts can be used as DTC activation sources, four each for TPU0, TPU3, TPU6, and TPU9, and two each for TPU1, TPU2, TPU4, TPU5, TPU7, TPU8, TPU10, and TPU11.

## 24.6 DMAC Activation

The DMAC can be activated by the TPUM.TGRA input capture/compare match interrupt of each channel. For details, see section 18, DMA Controller (DMACA).

A total of twelve TPUM.TGRA input capture/compare match interrupts can be used as DMAC activation sources, one for each channel.

## 24.7 A/D Converter Activation

The TPU can activate the A/D converter by the TPUM.TGRA input capture/compare match for each channel. When the TTGE bit in TPUM.TIER is set to 1, the TPU requests the A/D converter to start A/D conversion by the occurrence of a TPUM.TGRA input capture/compare match on a particular channel.

For the corresponding unit of A/D converter, see section 39, 10-Bit A/D Converter (ADb). ( $m = 0$  to 4)

## 24.8 PPG Trigger

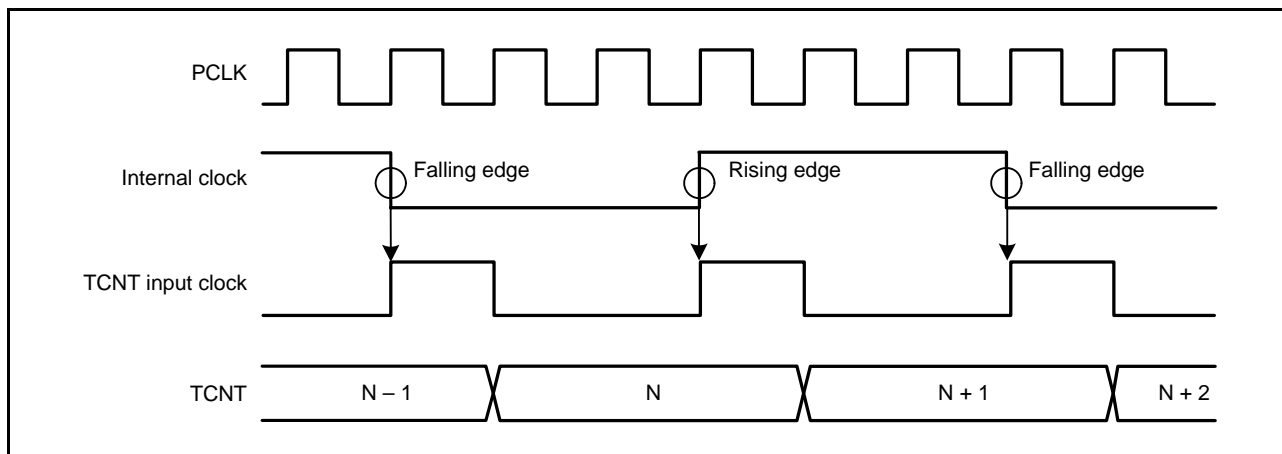
Input capture to or compare match with TGRA and TGRB in TPU0 to TPU3 (or TPU6 to TPU8) can be made to act as a PPG1 waveform trigger.

## 24.9 Operation Timing

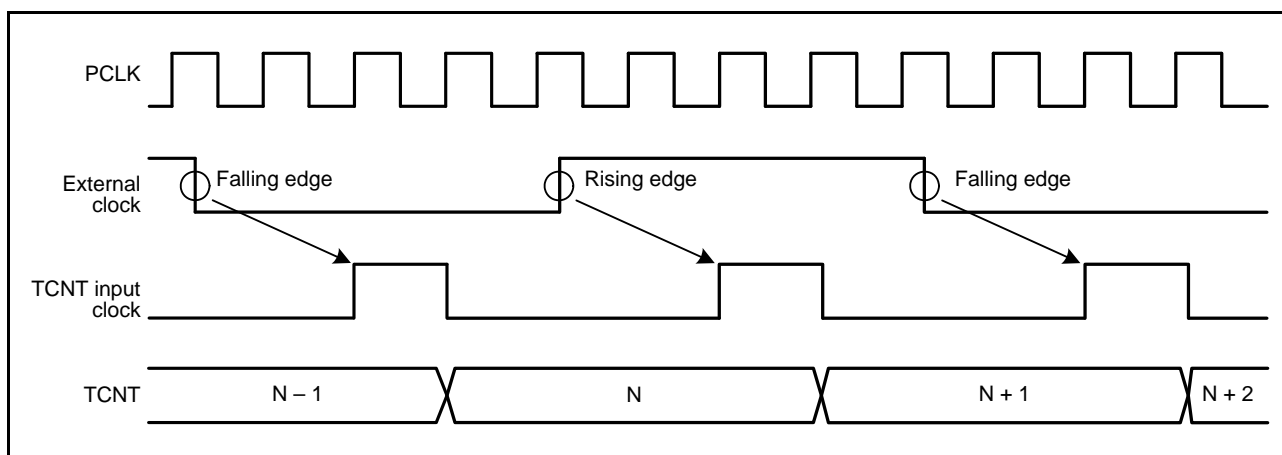
### 24.9.1 Input/Output Timing

#### (1) TPUm.TCNT Count Timing

Figure 24.32 shows TPUm.TCNT count timing in internal clock operation, and Figure 24.33 shows TCNT count timing in external clock operation.



**Figure 24.32** Count Timing in Internal Clock Operation



**Figure 24.33** Count Timing in External Clock Operation

(2) Output Compare Output Timing

A compare match signal is generated in the final state in which TPUm.TCNT and TPUm.TGRy match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TPUm.TIORH, TPUm.TIORL, or TPUm.TIOR is output to the output compare output pin TIOCyn (y = A to D, n = 0 to 11). After a match between TCNT and TGRy, the compare match signal is not generated until the TCNT input clock is generated.

Figure 24.34 shows output compare output timing.

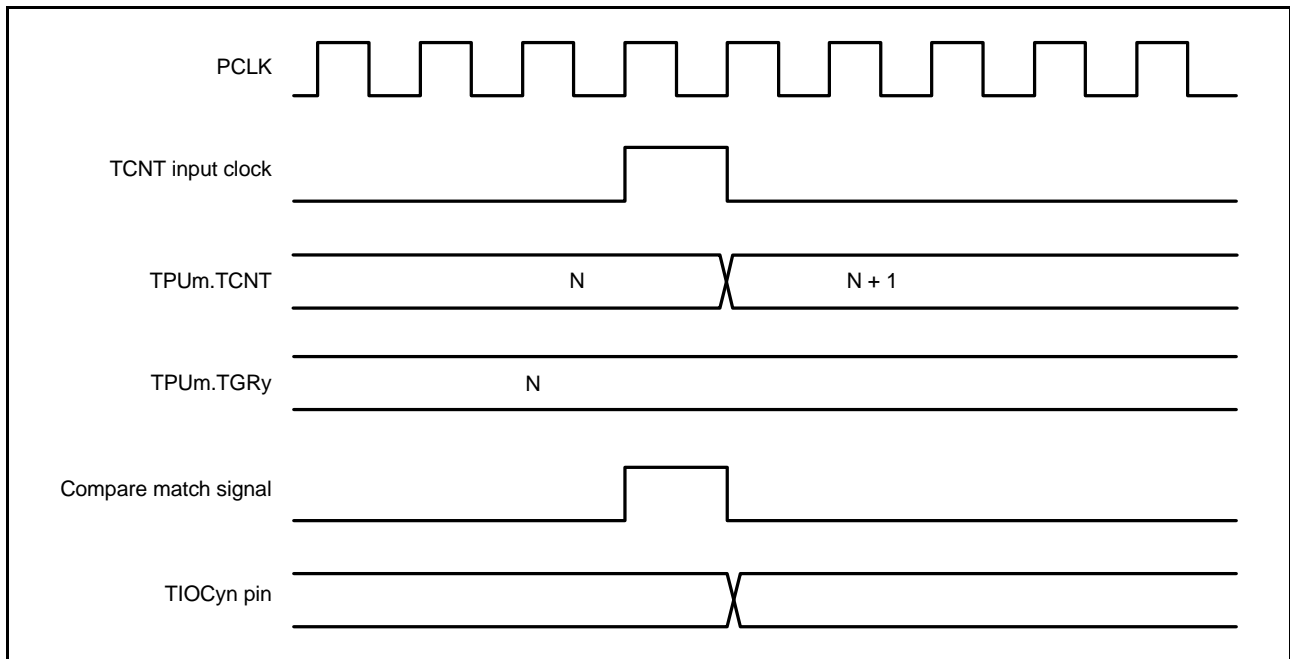


Figure 24.34 Output Compare Output Timing

(3) Input Capture Signal Timing

Figure 24.35 shows input capture signal timing.

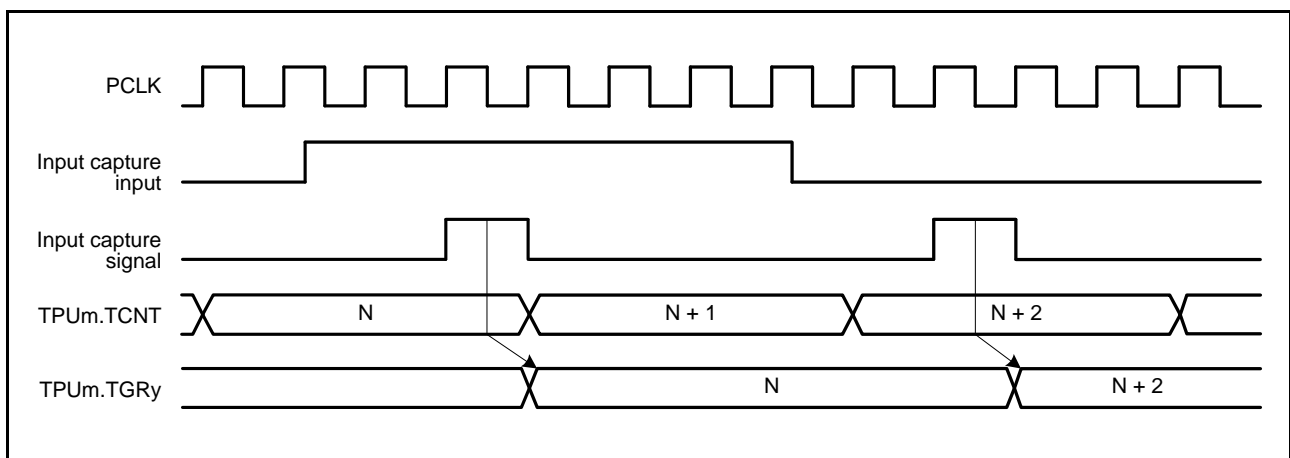


Figure 24.35 Input Capture Signal Timing

(4) Timing for Counter Clearing by Compare Match/Input Capture

Figure 24.36 shows the timing when counter clearing by compare match occurrence is specified, and Figure 24.37 shows the timing when counter clearing by input capture occurrence is specified.

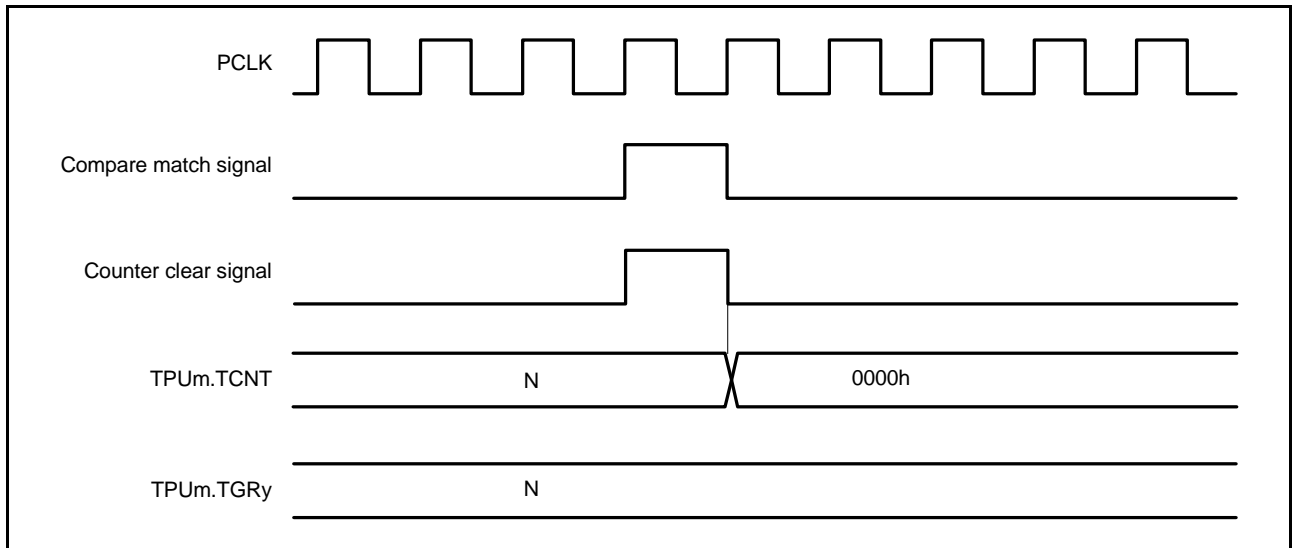


Figure 24.36 Counter Clear Timing (Compare Match)

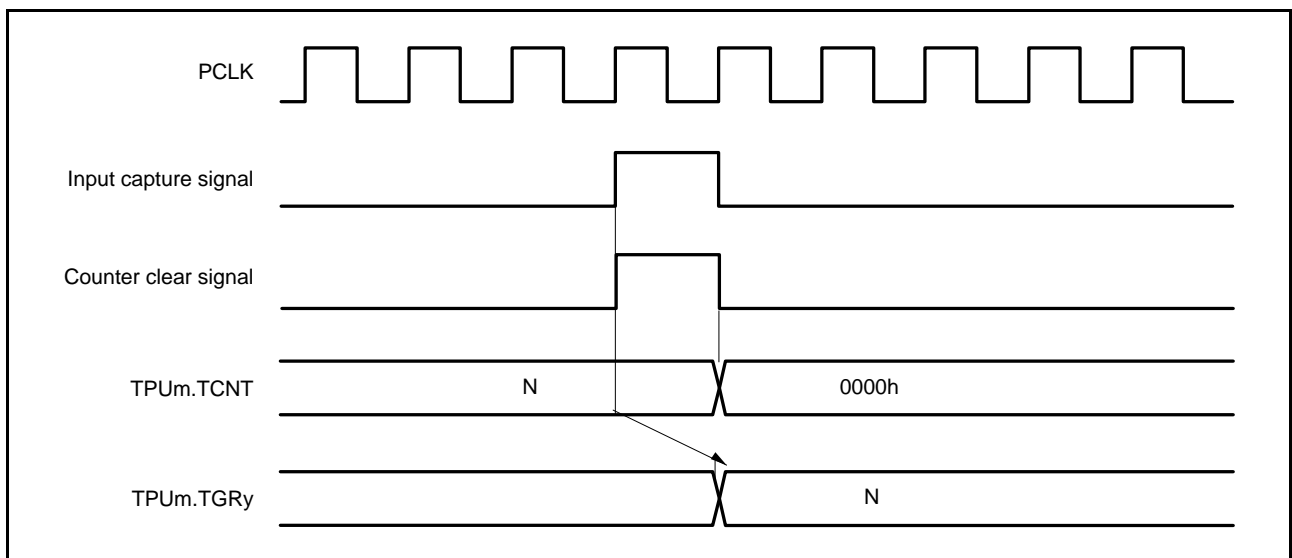


Figure 24.37 Counter Clear Timing (Input Capture)

(5) Buffer Operation Timing

Figure 24.38 and Figure 24.39 show the timings in buffer operation.

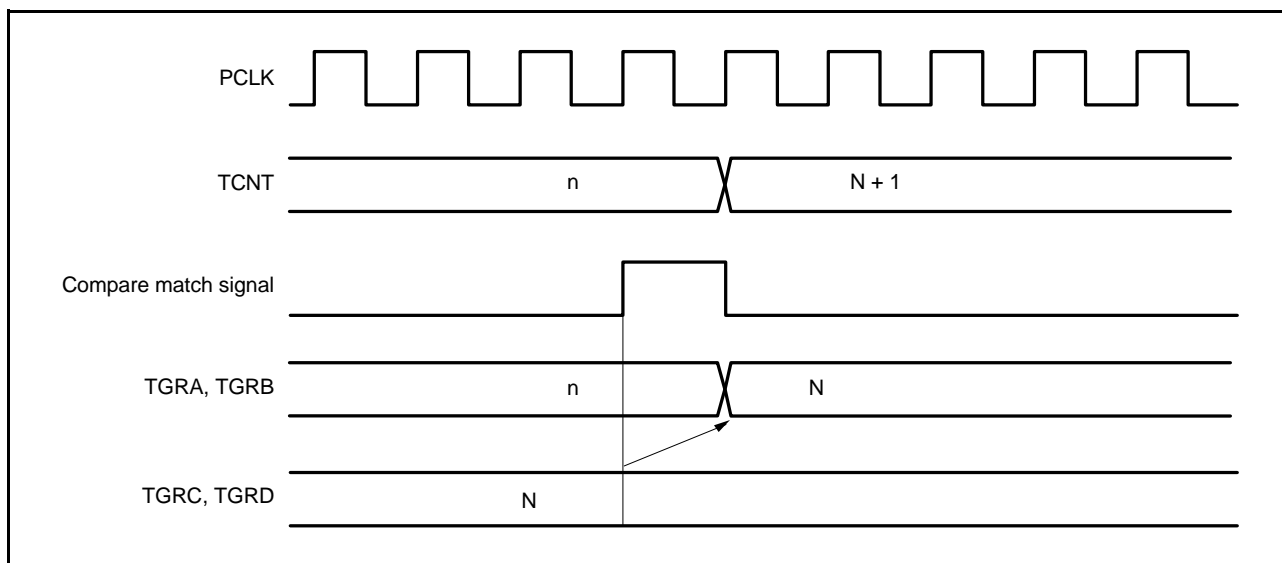


Figure 24.38 Buffer Operation Timing (Compare Match)

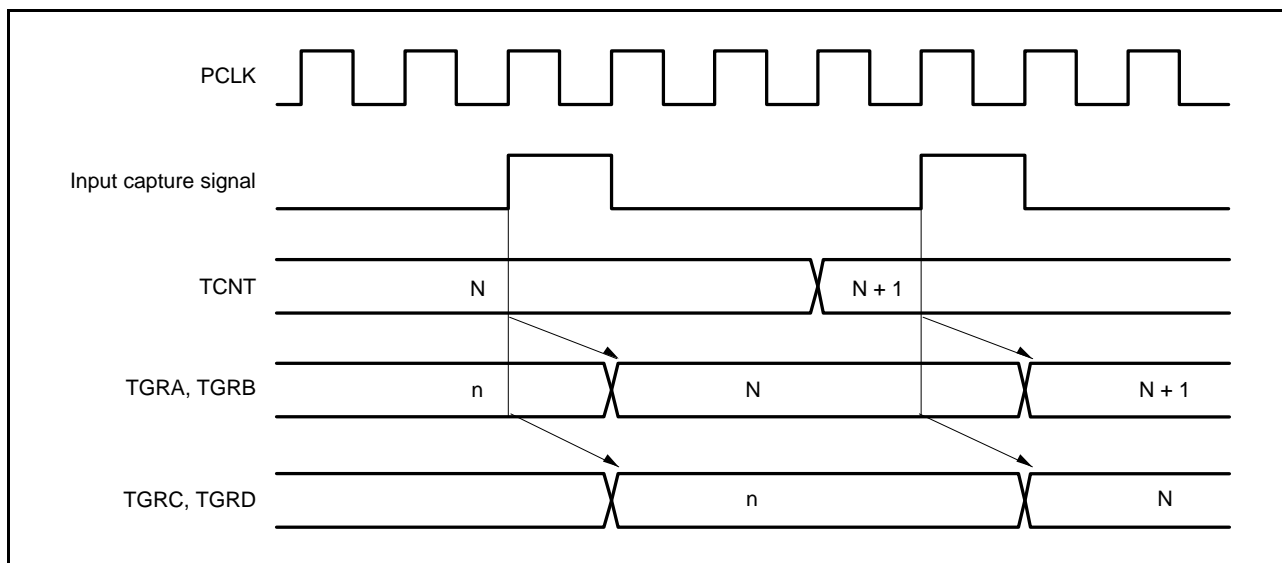


Figure 24.39 Buffer Operation Timing (Input Capture)



### 24.9.2 Interrupt Signal Timing

#### (1) Timing of Interrupt Signal Setting on Compare Match

Figure 24.40 shows the timing for setting the interrupt signal by compare match occurrence.

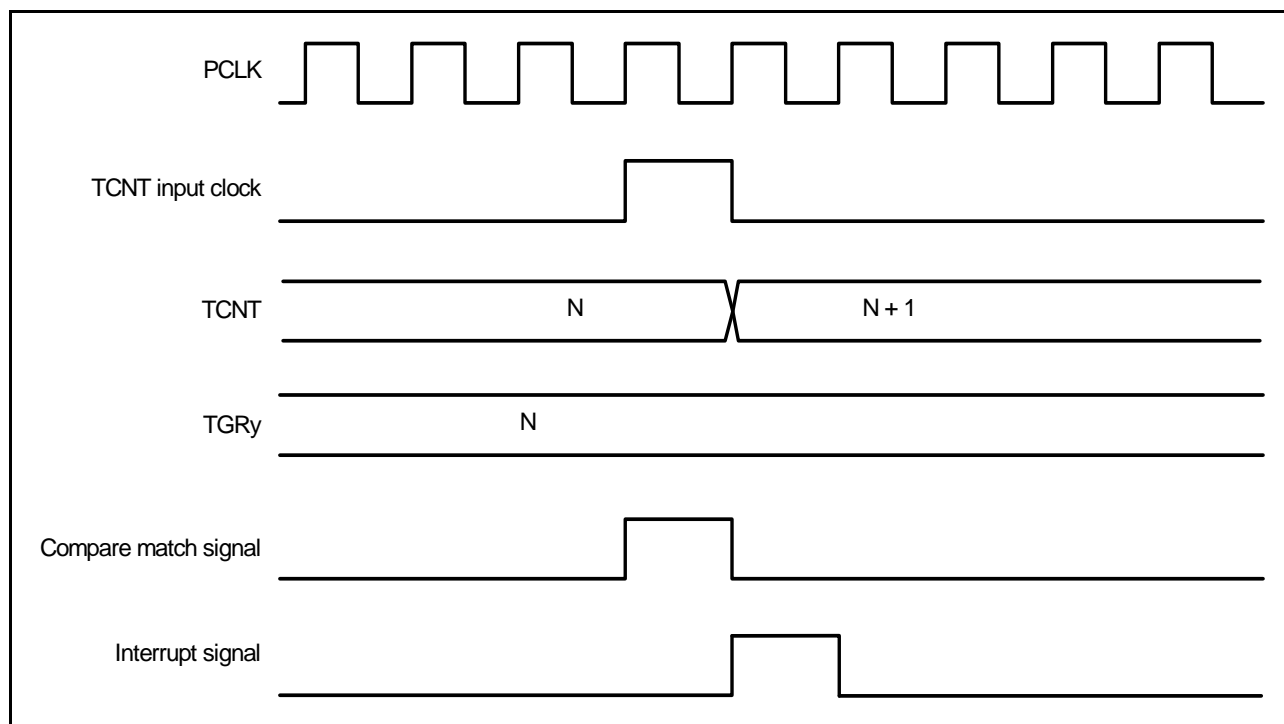


Figure 24.40 TGI $y$  Interrupt Timing (Compare Match)

#### (2) Timing of Interrupt Signal Setting on Input Capture

Figure 24.41 shows the timing for setting the interrupt signal by input capture occurrence.

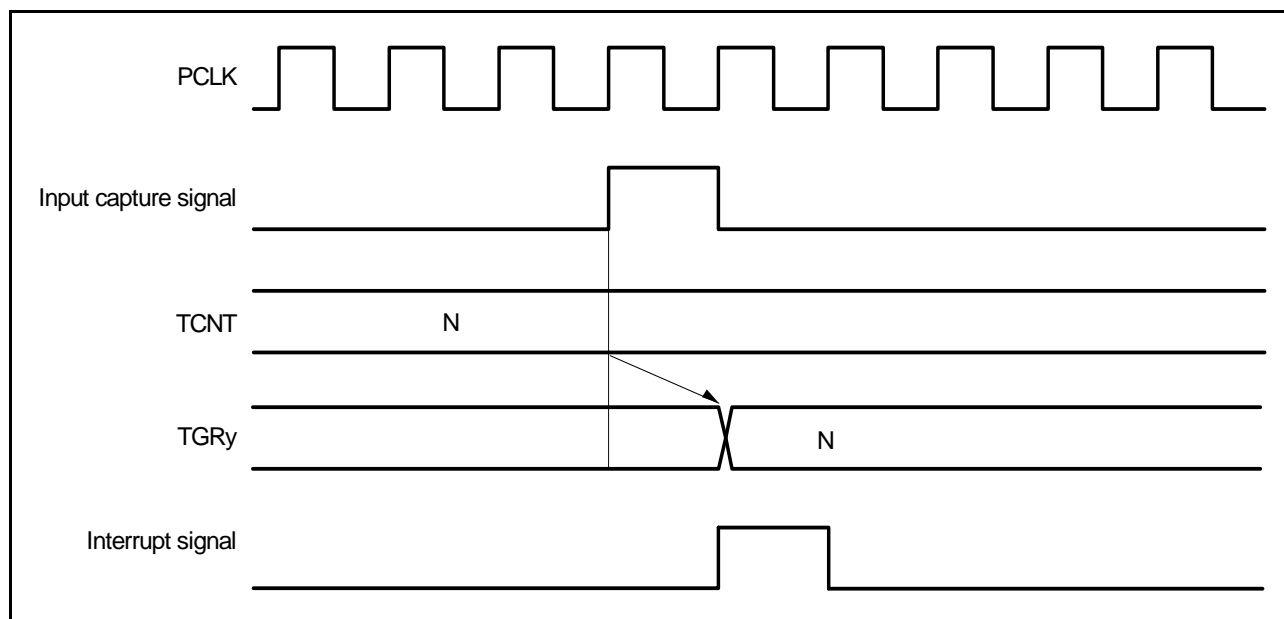


Figure 24.41 TGI $y$  Interrupt Timing (Input Capture)

(3) Timing of TCImV/TCImU Interrupt Signal Setting

Figure 24.42 shows the timing for generating the TCImV interrupt signal by overflow occurrence.

Figure 24.43 shows the timing for generating the TCImU interrupt signal by underflow occurrence.

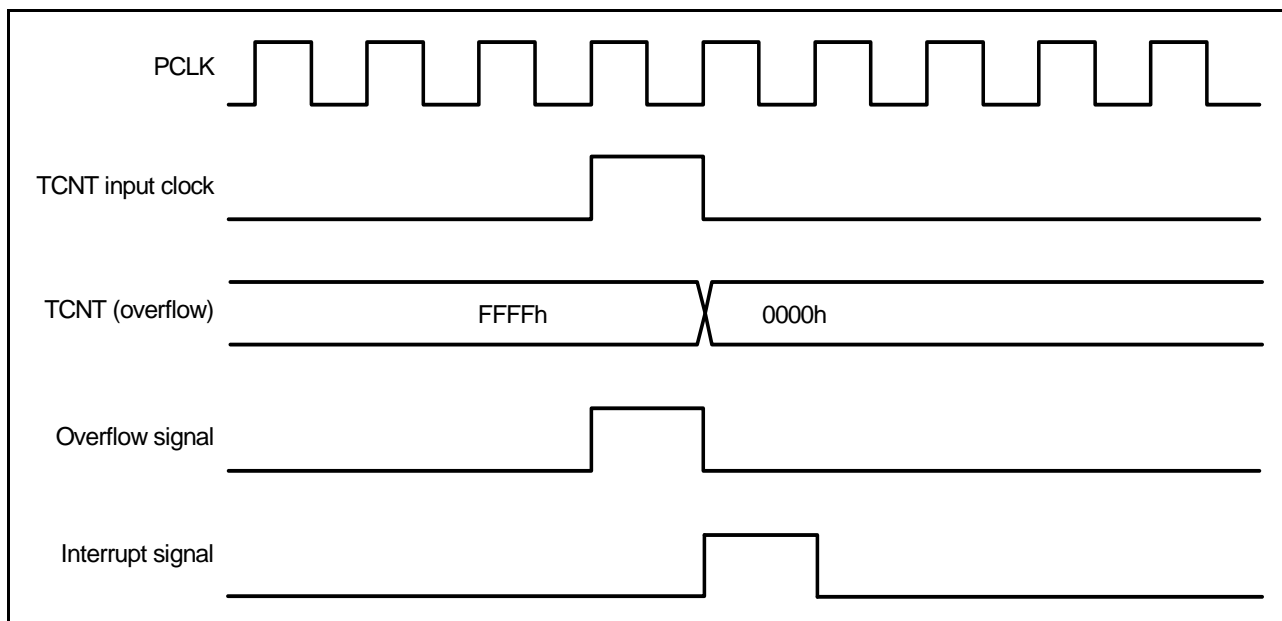


Figure 24.42 TCImV Interrupt Setting Timing

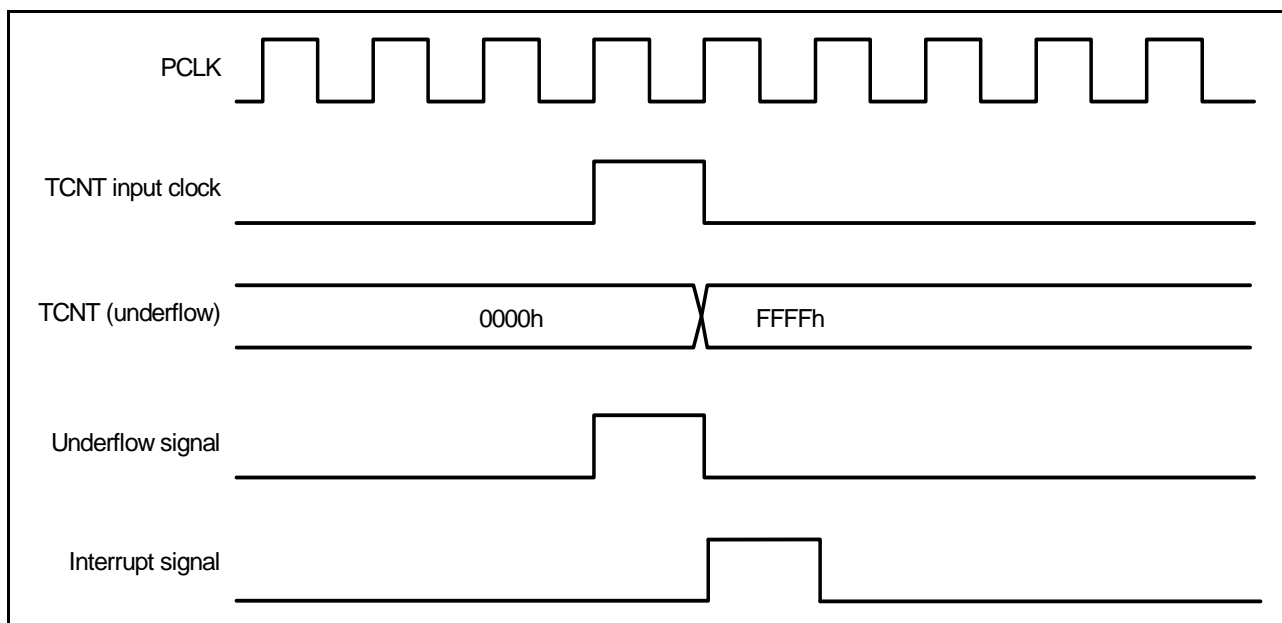


Figure 24.43 TCImU Interrupt Setting Timing

## 24.10 Usage Notes

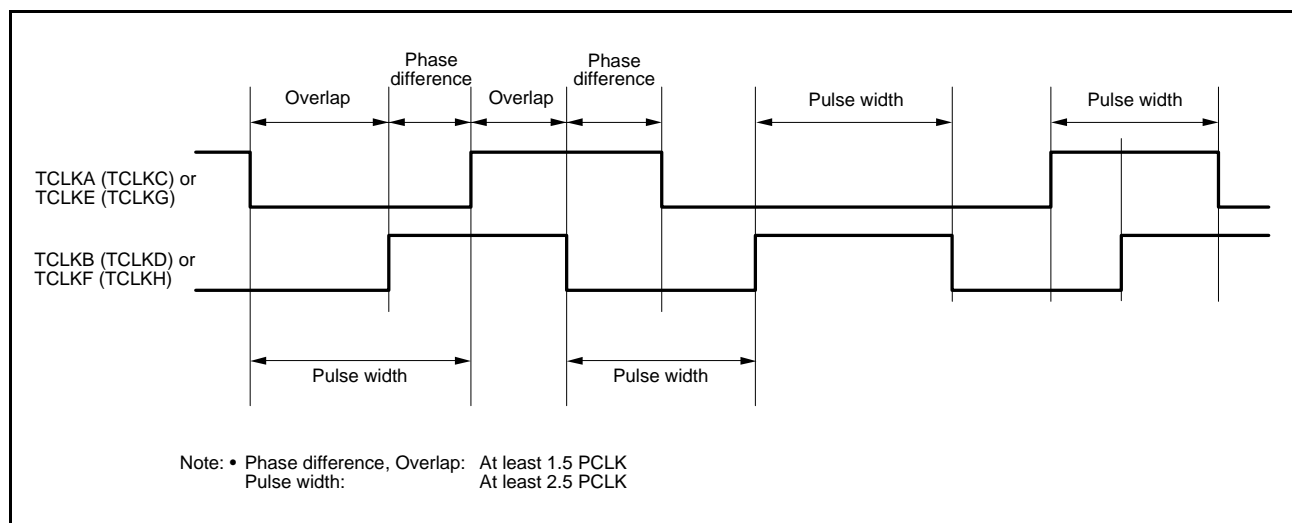
### 24.10.1 Module-Stop Function Setting

Operation of the TPU can be disabled or enabled using the module-stop control register. The TPU does not operate with the initial setting. Register access is enabled by clearing module-stop state. For details, see section 11, Low Power Consumption.

### 24.10.2 Input Clock Restrictions

The input clock pulse width must be at least 1.5 PCLK in the case of single-edge detection, and at least 2.5 PCLK in the case of both-edge detection. The TPU will not operate properly with a narrower pulse width.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 PCLK, and the pulse width must be at least 2.5 PCLK. Figure 24.44 shows the input clock conditions in phase counting mode.



**Figure 24.44 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode**

### 24.10.3 Caution on Cycle Setting

When counter clearing by compare match is set, TPUm.TCNT is cleared in the final state in which it matches the TPUm.TGRy value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

$$f = \frac{f_{TCNT\_CLK}}{(N+1)}$$

- f: Counter frequency
- f<sub>TCNT\_CLK</sub>: Counter clock frequency
- N: TGRy set value

### 24.10.4 Conflict between TPUm.TCNT Write and Clear Operations

If the counter clearing signal is generated in a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed. Figure 24.45 shows the timing in this case.

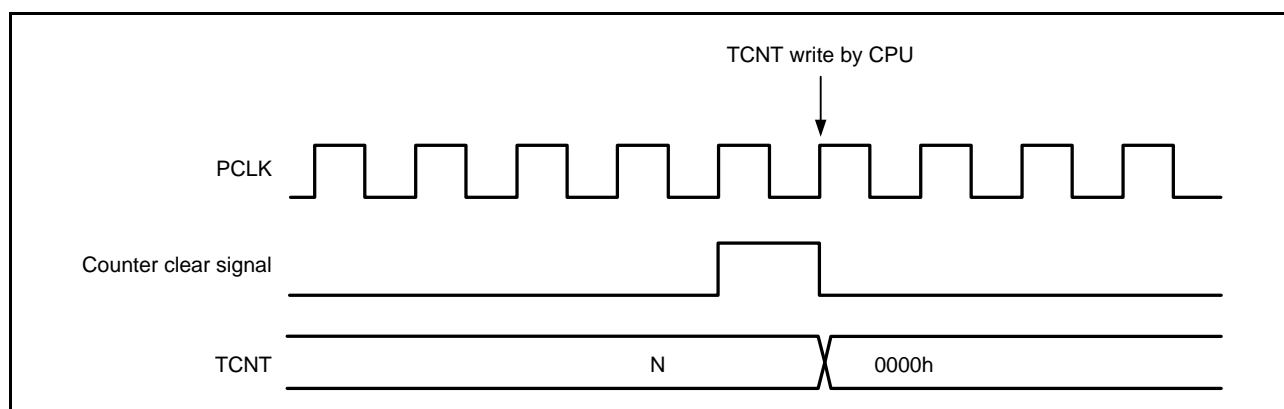


Figure 24.45 Conflict between TPUm.TCNT Write and Clear Operations

### 24.10.5 Conflict between TPUm.TCNT Write and Increment Operations

If incrementing occurs in a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented. Figure 24.46 shows the timing in this case.

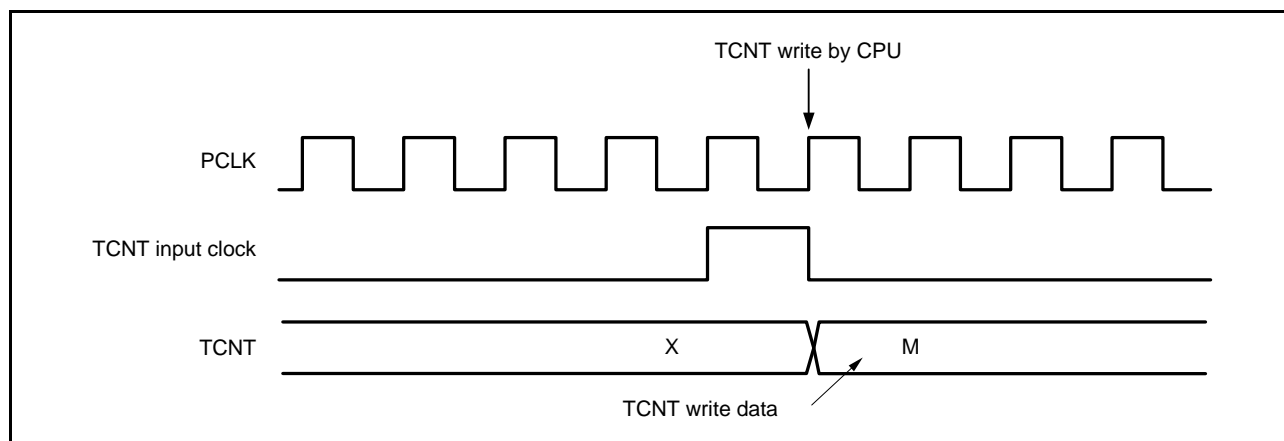


Figure 24.46 Conflict between TPUm.TCNT Write and Increment Operations

### 24.10.6 Conflict between TPUM.TGRy Write and Compare Match

If a compare match occurs in a TGRy write cycle, the TGRy write takes precedence and the compare match signal is disabled. A compare match also does not occur when the same value as before is written.

Figure 24.47 shows the timing in this case.

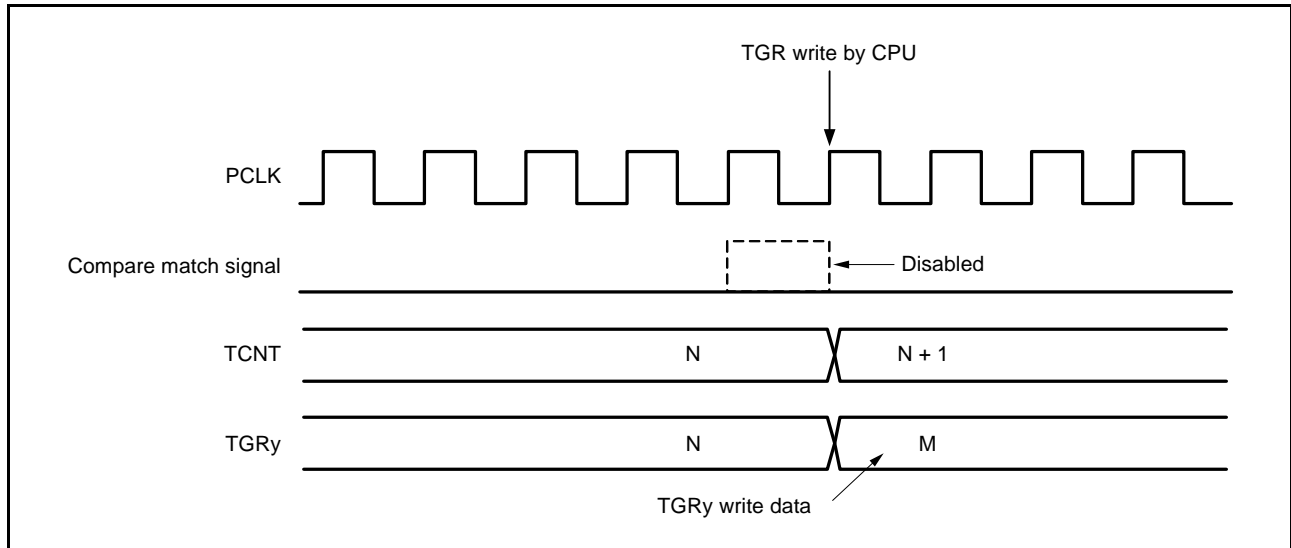


Figure 24.47 Conflict between TPUM.TGRy Write and Compare Match

### 24.10.7 Conflict between Buffer Register Write and Compare Match

If a compare match occurs in a TPUM.TGRy write cycle, the data transferred to TGRy by the buffer operation will be the data before writing.

Figure 24.48 shows the timing in this case.

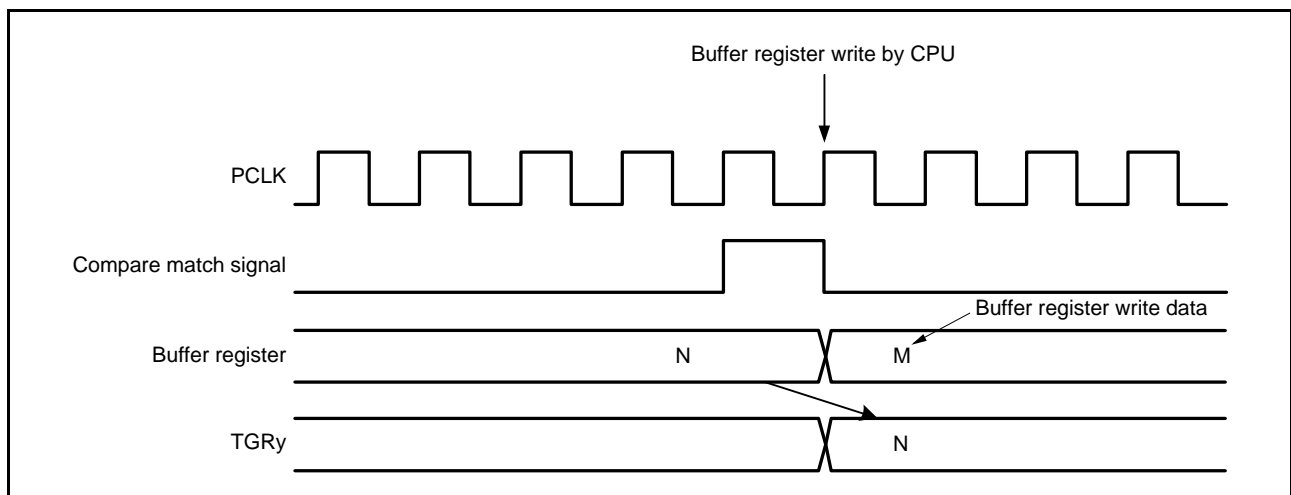


Figure 24.48 Conflict between Buffer Register Write and Compare Match

### 24.10.8 Conflict between TPUM.TGRy Read and Input Capture

If the input capture signal is generated in a TGRy read cycle, the data that is read will be the data before input capture transfer.

Figure 24.49 shows the timing in this case.

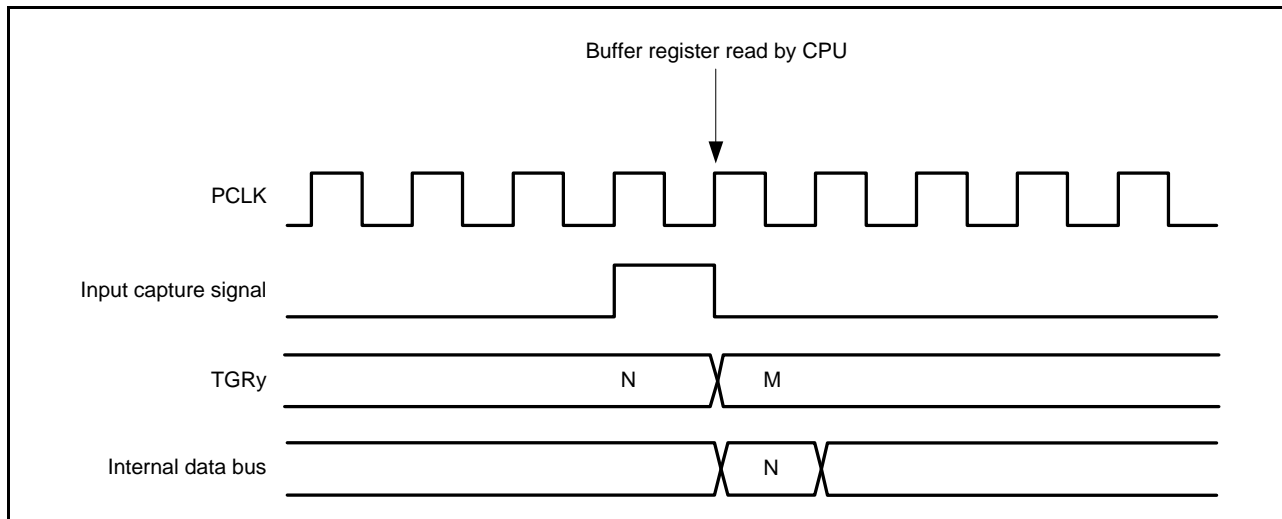


Figure 24.49 Conflict between TPUM.TGRy Read and Input Capture

### 24.10.9 Conflict between TPUM.TGRy Write and Input Capture

If the input capture signal is generated in a TGRy write cycle, the input capture operation takes precedence and the write to TGRy is not performed. Figure 24.50 shows the timing in this case.

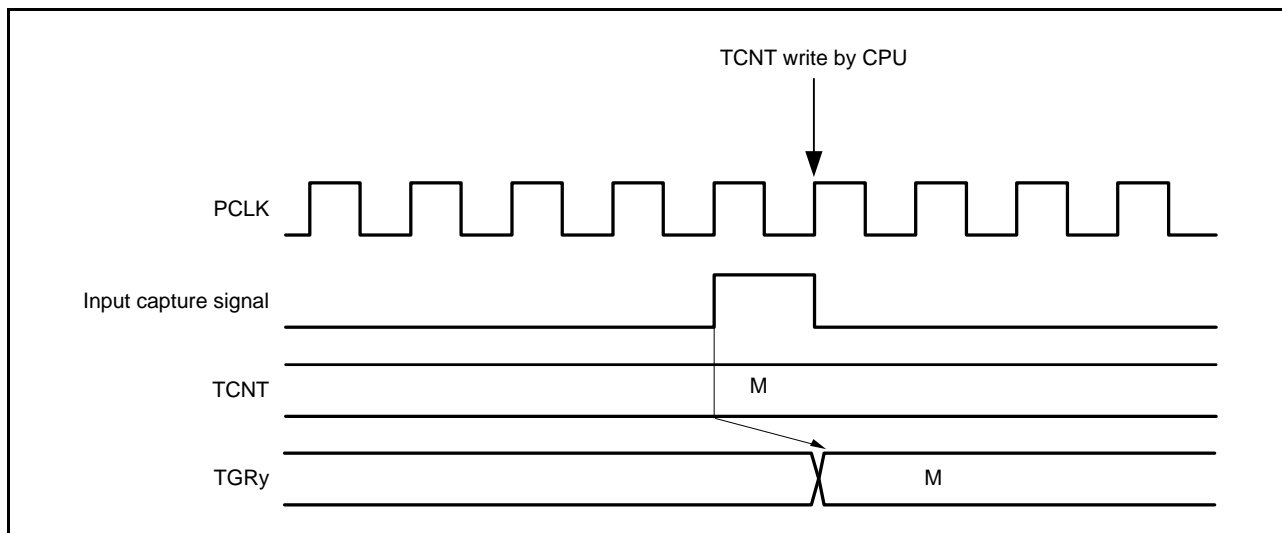
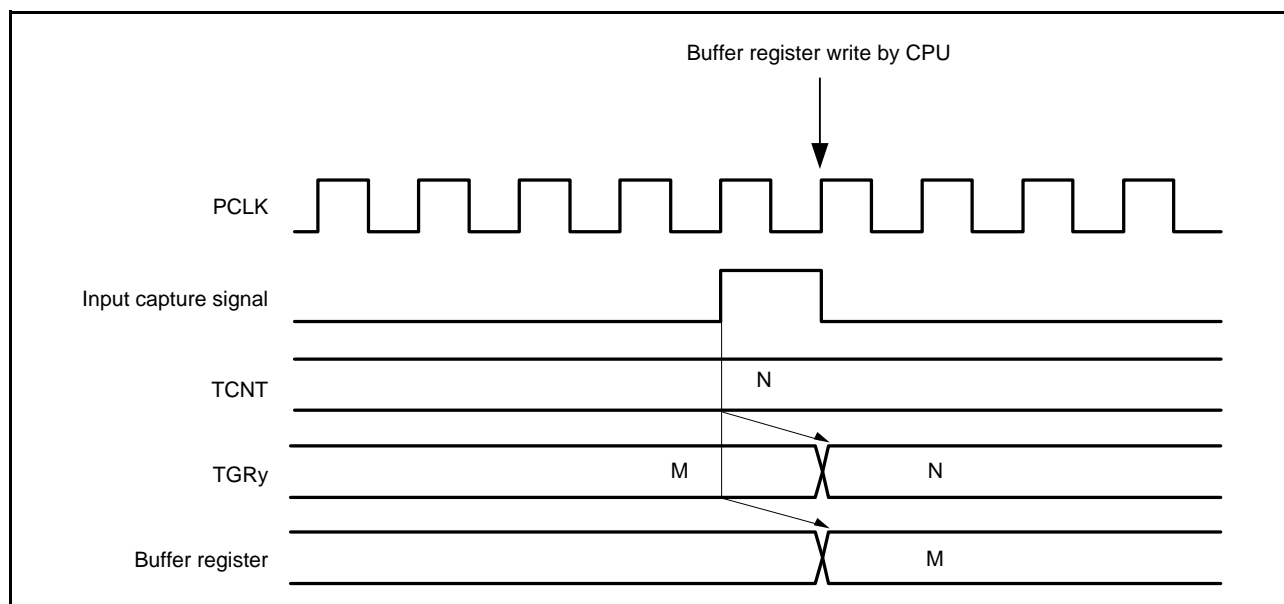


Figure 24.50 Conflict between TPUM.TGRy Write and Input Capture

### 24.10.10 Conflict between Buffer Register Write and Input Capture

If the input capture signal is generated in a buffer register write cycle, the buffer operation takes precedence and the write to the buffer register is not performed. Figure 24.51 shows the timing in this case.

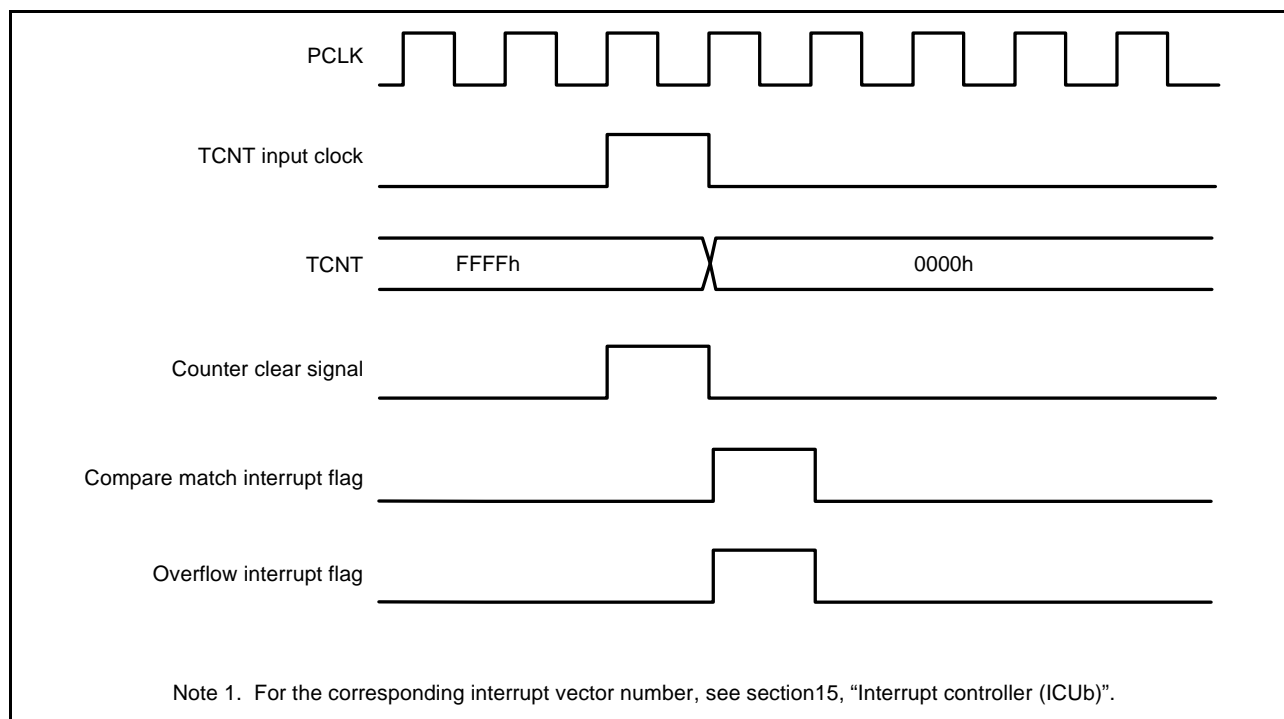


**Figure 24.51 Conflict between Buffer Register Write and Input Capture**

### 24.10.11 Conflict between Overflow/Underflow and Counter Clearing

If overflow/underflow and counter clearing occur simultaneously, TPUm.TCNT counter is cleared with the generation of the compare match interrupt and an overflow interrupt is generated.

Figure 24.52 shows the operation timing when a TPUm.TGRy compare match is specified as the clearing source and FFFFh is set in TGRy.

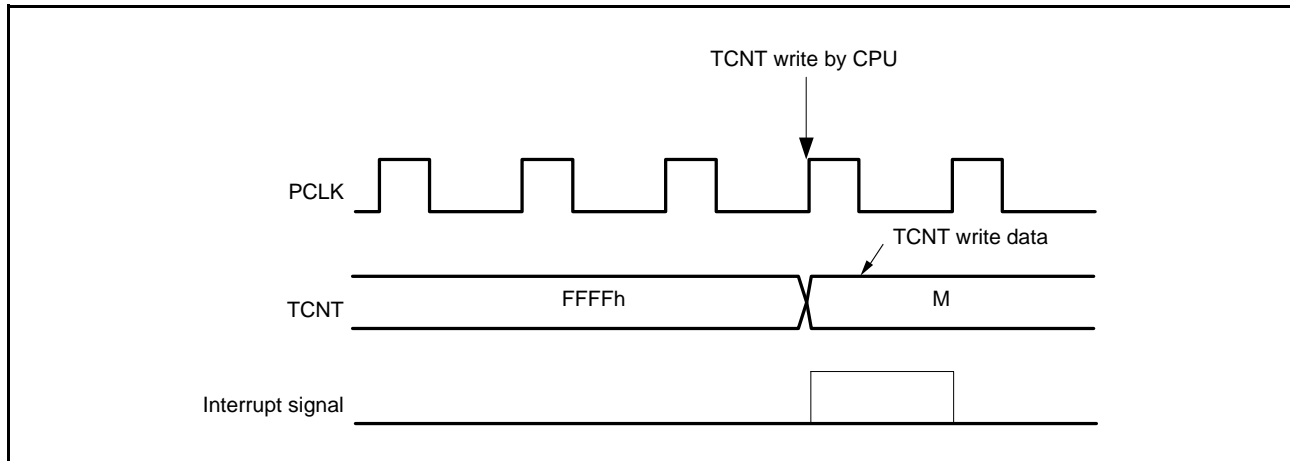


Note 1. For the corresponding interrupt vector number, see section 15, "Interrupt controller (ICUb)".

**Figure 24.52 Conflict between Overflow and Counter Clearing**

### 24.10.12 Conflict between TPUm.TCNT Write and Overflow/Underflow

If an overflow/underflow occurs due to increment/decrement in a TCNT write cycle, the TCNT write takes precedence. Figure 24.53 shows the operation timing when there is conflict between TCNT write and overflow.



**Figure 24.53** Conflict between TPUm.TCNT Write and Overflow

### 24.10.13 Multiplexing of I/O Pins

In the RX630 Group, the TCLKA input pin is multiplexed with the TIOCB5 I/O pin, the TCLKB input pin with the TIOCB2 I/O pin, the TCLKC input pin with the TIOCB1 I/O pin, the TCLKD input pin with the TIOCB0 I/O pin, the TCLKG input pin with the TIOCB7 I/O pin, the TCLKH input pin with the TIOCB8 I/O pin, the TCLKB input pin with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCC3 I/O pin, the TCLKD input pin with the TIOCD3 I/O pin, the TCLKE input pin with the TIOCC6 I/O pin, and the TCLKF input pin with the TIOCD6 I/O pin. When an external clock is input, compare match output should not be performed from a multiplexed pin.



### 24.10.14 Continuous Output of Compare-Match Pulse Interrupt Signal

When TGR is set to 0000h, PCLK/1 is set as the counter clock, and compare match is set as the counter clear source, the TCNT counter remains 0000h and is not updated, and a compare-match pulse interrupt signal is output continuously to form a flat signal level.

When a pulse interrupt signal is used, the interrupt controller cannot detect the second and subsequent interrupts. Figure 24.54 shows an operation timing when the compare-match pulse interrupt signal is continuously output.

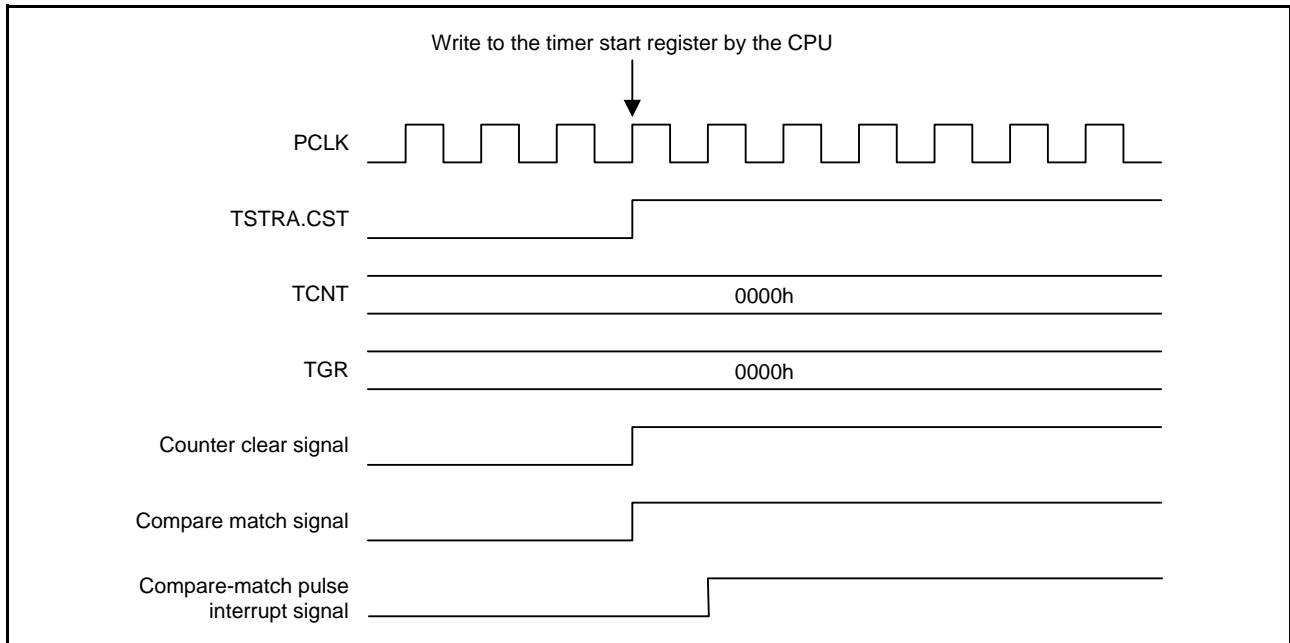


Figure 24.54 Continuous Output of Compare-Match Pulse Interrupt Signal

### 24.10.15 Continuous Output of Input-Capture Pulse Interrupt Signal

When input-capture signal is set on both edges and when the pulse width of the input-capture input equals to one PCLK cycle detected by internal sampling, input capture is generated continuously on the rising and falling edges. Therefore, an input-capture pulse interrupt signal is output continuously to form a flat signal level.

When a pulse interrupt signal is used, the interrupt controller cannot detect the second and subsequent interrupts. Figure 24.55 shows an operation timing when the input-capture pulse interrupt signal is output continuously.

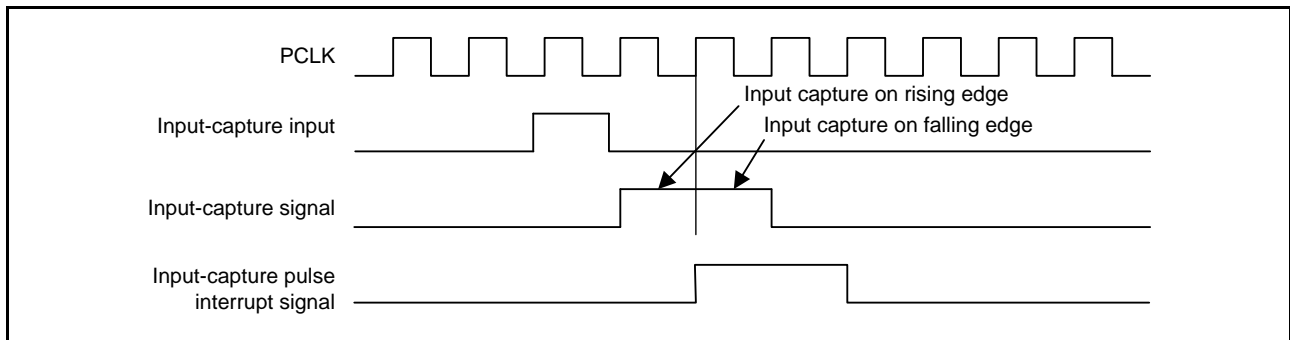


Figure 24.55 Continuous Output of Input-Capture Pulse Interrupt Signal

### 24.10.16 Continuous Output of Underflow Pulse Interrupt Signal

If two external clock signals' same direction edges to be phase counted are generated within two PCLK cycles in phase counting mode 1, with TGR being 0000h, and compare match set as the counter clear source, the TCNT counter remains 0000h and is not updated, and a compare-match pulse interrupt signal and an underflow interrupt signal are output continuously to form a flat signal level.

When a pulse interrupt signal is used, the interrupt controller cannot detect the second and subsequent interrupts.

Figure 24.56 shows an operation timing when the underflow pulse interrupt signal is output continuously.

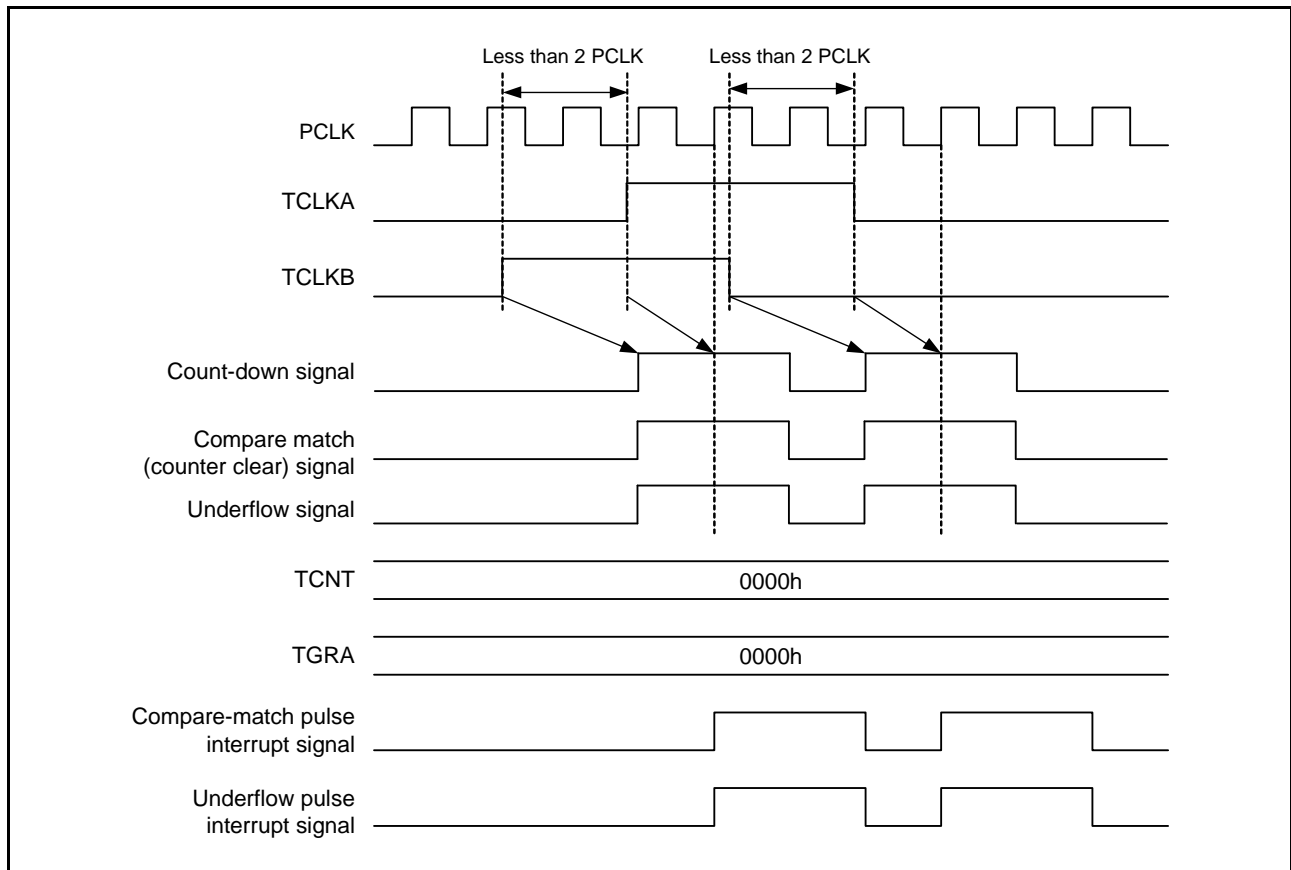


Figure 24.56 Continuous Output of Underflow Pulse Interrupt Signal

## 25. Programmable Pulse Generator (PPG)

The programmable pulse generator (PPG) generates pulse outputs by using the 16-bit timer pulse unit (TPU) and the multi-function timer pulse unit 2 (MTU) as a time base.

The RX630 Group has two PPG units, each of which controls up to 16 pulse output pins. The pulse outputs from the PPGs are divided into 4-bit groups that can operate all simultaneously and independently.

### 25.1 Overview

Table 25.1 lists the specifications of the PPG and Table 25.2 lists PPG functions.

Figure 25.1 and Figure 25.2 show block diagrams of the PPGs.

**Table 25.1 Specifications of PPG**

Item	Specifications
Number of output bits	Up to 32 bits
Pulse output	<ul style="list-style-type: none"> <li>• Two units, each capable of output through four pin groups</li> <li>• Output trigger signals are selectable.</li> <li>• Non-overlapping operation is possible.</li> <li>• Inverted output is selectable.</li> </ul>
Output data transfer	Can operate together with the DTC and DMAC (when TPU and MTU interrupts are in use)
Power consumption reducing function	Module-stop state can be set for each unit.

**Table 25.2 List of PPG Functions**

Item			PPG0	PPG1
PPG output trigger	MTU channels 0 to 3 (MTU0 to MTU3)	Compare match	○	○
		Input capture	○	○
	TPU (unit 0) channels 0 to 3 (TPU0 to TPU3)	Compare match	—	○
		Input capture	—	○
Non-overlapping operation			○	○
Output data transfer	DTC		○	○
	DMAC		○	○
Selecting inverted output			○	○
Setting the module-stop state*1			The MSTPA11 bit in MSTPCRA	The MSTPA10 bit in MSTPCRA

○: Possible

—: Not possible

Note 1. For details, see section 11, Low Power Consumption.

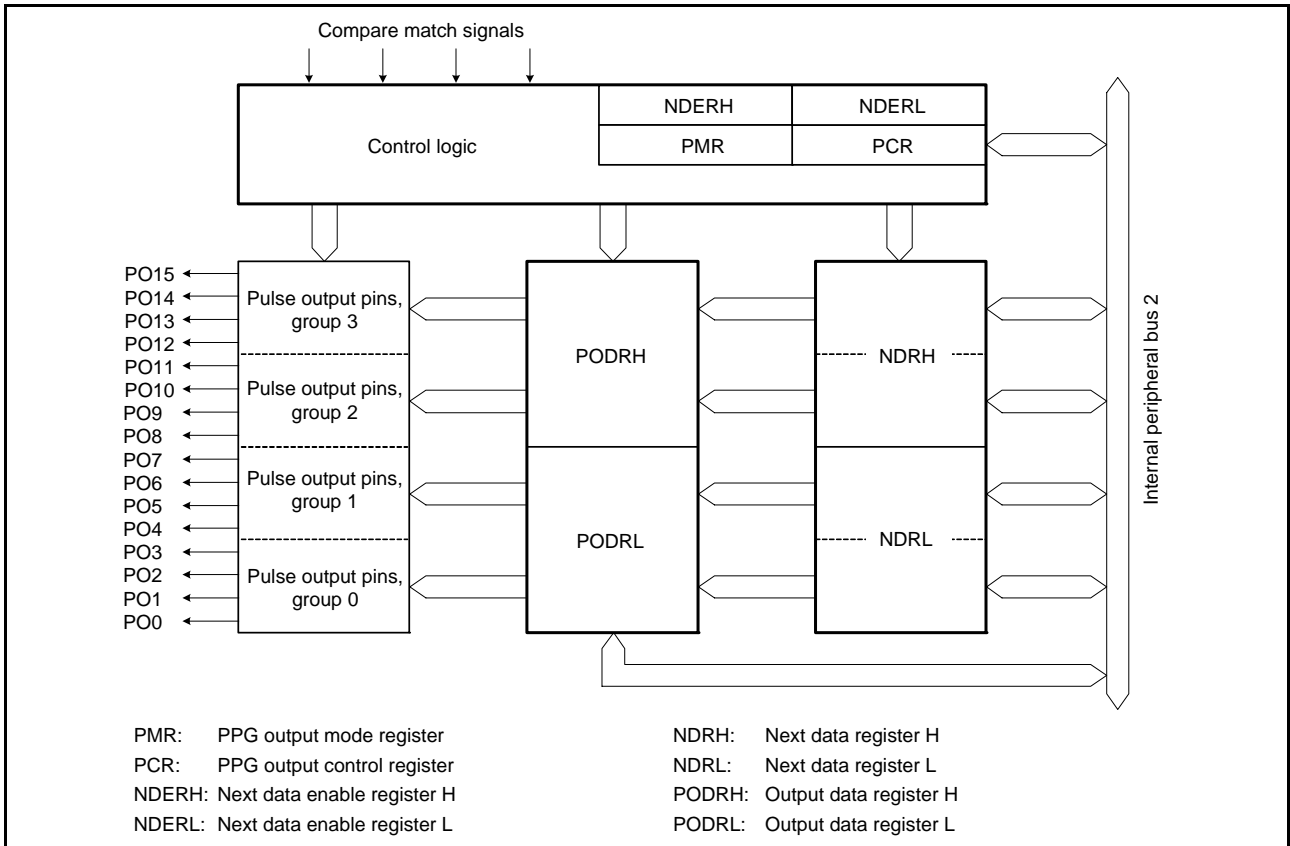


Figure 25.1 Block Diagram of PPG0

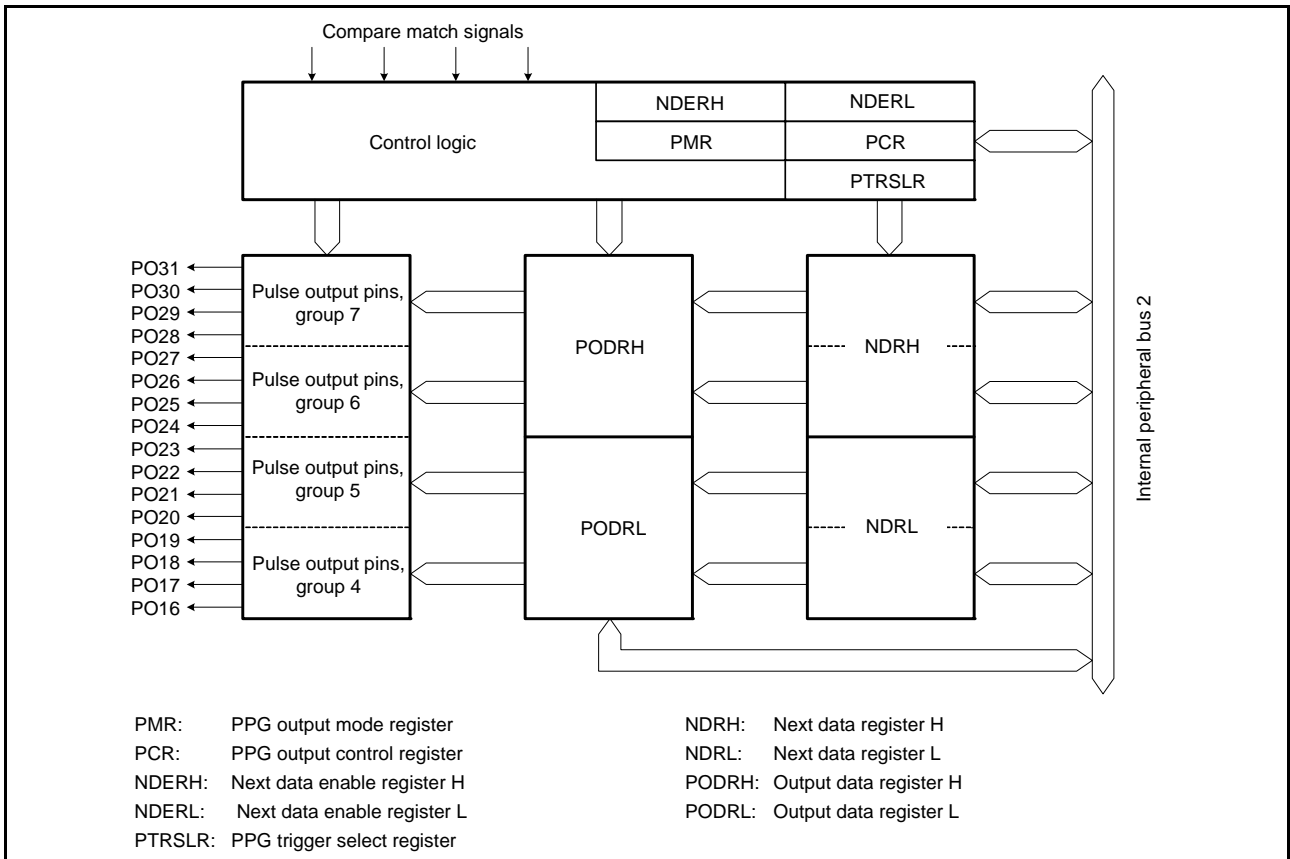


Figure 25.2 Block Diagram of PPG1

Table 25.3 lists the pin configuration of the PPG.

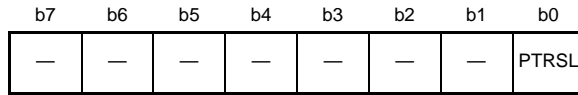
**Table 25.3 Pin Configuration of PPG**

Unit	Pin Name	I/O	Function	
PPG0	PO0	Output	Group 0 pulse output	
	PO1	Output		
	PO2	Output		
	PO3	Output		
	PO4	Output	Group 1 pulse output	
	PO5	Output		
	PO6	Output		
	PO7	Output		
	PO8	Output	Group 2 pulse output	
	PO9	Output		
	PO10	Output		
	PO11	Output		
	PPG1	PO12	Output	Group 3 pulse output
		PO13	Output	
		PO14	Output	
PO15		Output		
PO16		Output	Group 4 pulse output	
PO17		Output		
PO18		Output		
PO19		Output		
PO20		Output	Group 5 pulse output	
PO21		Output		
PO22	Output			
PO23	Output			
PO24	Output	Group 6 pulse output		
PO25	Output			
PO26	Output			
PO27	Output			
PO28	Output	Group 7 pulse output		
PO29	Output			
PO30	Output			
PO31	Output			

## 25.2 Register Descriptions

### 25.2.1 PPG Trigger Select Register (PTRSLR)

Address(es): 0008 81F0h



Value after reset: 0 0 0 0 0 0 0 1

- PPG1.PTRSLR

Bit	Symbol	Bit Name	Description	R/W
b0	PTRSL	PPG Trigger Select	0: Selects the set of MTU0 to MTU3 as the trigger channels for PPG1. 1: Selects the set of TPU0 to TPU3 as the trigger channels for PPG1.	R/W
b1 to 7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

#### PTRSL Bit (PPG Trigger Select)

This bit selects either MTU0 to MTU3 or TPU0 to TPU3 as a set of trigger channels for PPG1.

When this bit is set to 0, MTU0 to MTU3 are selected as a set of trigger channels for PPG1. When it is set to 1, TPU0 to TPU3 are selected as a set of trigger channels for PPG1.

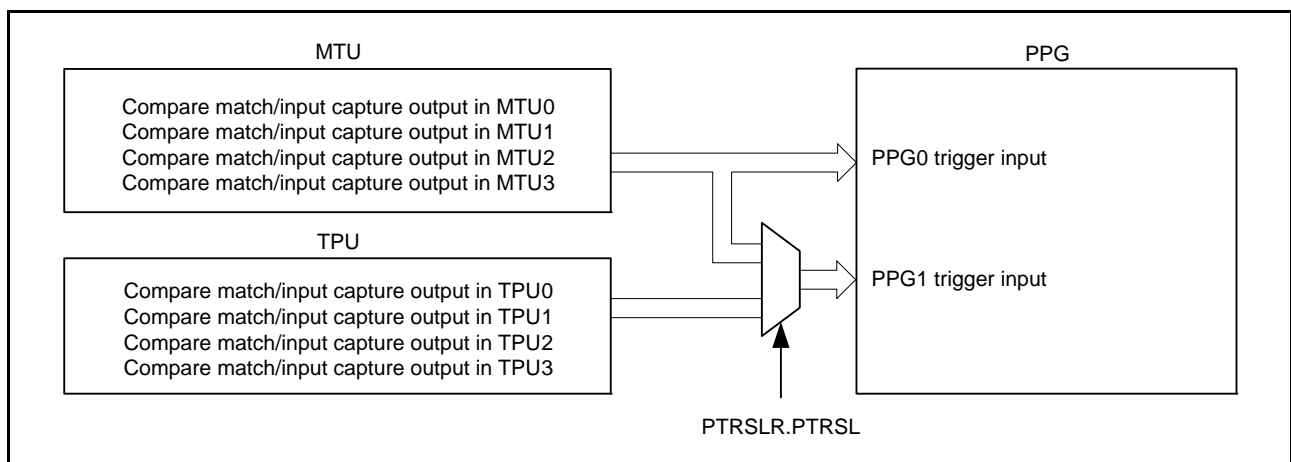


Figure 25.3 Block Diagram of PPG Trigger Selection

## 25.2.2 Next Data Enable Registers H (NDERH) Next Data Enable Registers L (NDERL)

Address(es): 0008 81E8h

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.NDERH	NDER 15	NDER 14	NDER 13	NDER 12	NDER 11	NDER 10	NDER9	NDER8
Value after reset:	0	0	0	0	0	0	0	0

Address(es): 0008 81E9h

	b7	b6	b5	b4	b3	b2	b1	b0
PPG0.NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
Value after reset:	0	0	0	0	0	0	0	0

- PPG0.NDERH

Bit	Symbol	Bit Name	Description	R/W
b0	NDER8	Next Data Transfer Enable	0: Data transfer is disabled.	R/W
b1	NDER9	Next Data Transfer Enable	1: Data transfer is enabled.	R/W
b2	NDER 10	Next Data Transfer Enable		R/W
b3	NDER 11	Next Data Transfer Enable		R/W
b4	NDER 12	Next Data Transfer Enable		R/W
b5	NDER 13	Next Data Transfer Enable		R/W
b6	NDER 14	Next Data Transfer Enable		R/W
b7	NDER 15	Next Data Transfer Enable		R/W

PPG0.NDERH selects the pins (PO15 to PO8) for outputs of pulse from the PPG on a bit-by-bit basis.

### NDER<sub>i</sub> Bits (Next Data Transfer Enable) (i = 15 to 8)

When these bits are set to 1, the PPG0.PCR specified trigger transfers data from the corresponding bit in PPG0.NDRH to the bit in PPG0.PODRH.

- PPG0.NDERL

Bit	Symbol	Bit Name	Description	R/W
b0	NDER0	Next Data Transfer Enable	0: Data transfer is disabled.	R/W
b1	NDER1	Next Data Transfer Enable	1: Data transfer is enabled.	R/W
b2	NDER2	Next Data Transfer Enable		R/W
b3	NDER3	Next Data Transfer Enable		R/W
b4	NDER4	Next Data Transfer Enable		R/W
b5	NDER5	Next Data Transfer Enable		R/W
b6	NDER6	Next Data Transfer Enable		R/W
b7	NDER7	Next Data Transfer Enable		R/W

PPG0.NDERL selects the pins (PO7 to PO0) for outputs of pulse from the PPG on a bit-by-bit basis.

### NDER<sub>i</sub> Bits (Next Data Transfer Enable) (i = 7 to 0)

When these bits are set to 1, the PPG0.PCR specified trigger transfers data from the corresponding bit in PPG0.NDRL to the bit in PPG0.PODRL.

Address(es): 0008 81F8h

	b7	b6	b5	b4	b3	b2	b1	b0
PPG1.NDERH	NDER 31	NDER 30	NDER 29	NDER 28	NDER 27	NDER 26	NDER 25	NDER 24
Value after reset:	0	0	0	0	0	0	0	0

Address(es): 0008 81F9h

	b7	b6	b5	b4	b3	b2	b1	b0
PPG1.NDERL	NDER 23	NDER 22	NDER 21	NDER 20	NDER 19	NDER 18	NDER 17	NDER 16
Value after reset:	0	0	0	0	0	0	0	0

- PPG1.NDERH

Bit	Symbol	Bit Name	Description	R/W
b0	NDER 24	Next Data Transfer Enable	0: Data transfer is disabled.	R/W
b1	NDER 25	Next Data Transfer Enable	1: Data transfer is enabled.	R/W
b2	NDER 26	Next Data Transfer Enable		R/W
b3	NDER 27	Next Data Transfer Enable		R/W
b4	NDER 28	Next Data Transfer Enable		R/W
b5	NDER 29	Next Data Transfer Enable		R/W
b6	NDER 30	Next Data Transfer Enable		R/W
b7	NDER 31	Next Data Transfer Enable		R/W

PPG1.NDERH selects the pins (PO31 to PO24) for outputs of pulse from the PPG on a bit-by-bit basis.

#### NDER<sub>i</sub> Bits (Next Data Transfer Enable) (i = 31 to 24)

When these bits are set to 1, the PPG1.PCR specified trigger transfers data from the corresponding bit in PPG1.NDRH to the bit in PPG1.PODRH.

- PPG1.NDERL

Bit	Symbol	Bit Name	Description	R/W
b0	NDER 16	Next Data Transfer Enable	0: Data transfer is disabled.	R/W
b1	NDER 17	Next Data Transfer Enable	1: Data transfer is enabled.	R/W
b2	NDER 18	Next Data Transfer Enable		R/W
b3	NDER 19	Next Data Transfer Enable		R/W
b4	NDER 20	Next Data Transfer Enable		R/W
b5	NDER 21	Next Data Transfer Enable		R/W
b6	NDER 22	Next Data Transfer Enable		R/W
b7	NDER 23	Next Data Transfer Enable		R/W

PPG1.NDERL selects the pins (PO23 to PO16) for outputs of pulse from the PPG on a bit-by-bit basis.

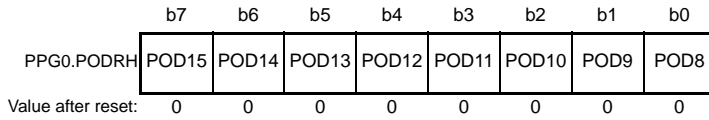
#### NDER<sub>i</sub> Bits (Next Data Transfer Enable) (i = 23 to 16)

When these bits are set to 1, the PPG1.PCR specified trigger transfers data from the corresponding bit in PPG1.NDRL to the bit in PPG1.PODRL.

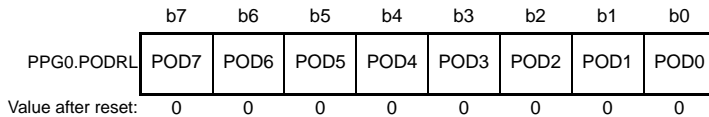


### 25.2.3 Output Data Registers H (PODRH) Output Data Registers L (PODRL)

Address(es): 0008 81EAh



Address(es): 0008 81EBh



- PPG0.PODRH

Bit	Symbol	Bit Name	Description	R/W
b0	POD8	Output Data Register	0: The low level is output on the POi pin.	R/W
b1	POD9	Output Data Register	1: The high level is output on the POi pin. (i = 15 to 8)	R/W
b2	POD10	Output Data Register		R/W
b3	POD11	Output Data Register		R/W
b4	POD12	Output Data Register		R/W
b5	POD13	Output Data Register		R/W
b6	POD14	Output Data Register		R/W
b7	POD15	Output Data Register		R/W

PPG0.PODRH stores pulse output values. For bits corresponding to pins that have been set for pulse output by PPG0.NDERH, the output trigger transfers the values in PPG0.NDRH to this register.

#### PODi Bit (Output Data Register) (i = 15 to 8)

When an output trigger is generated during PPG operation, the values of bits for which data transfer is enabled in the PPG0.NDERH register are transferred from the PPG0.NDRH register to this register. Writing from the CPU is impossible while any of the NDERi (i = 15 to 8) bits in PPG0.NDERH is 1. The initial output level for the pulse signal can be set when the value in the PPG0.NDERH register is 00h.

- PPG0.PODRL

Bit	Symbol	Bit Name	Description	R/W
b0	POD0	Output Data Register	0: The low level is output on the POi pin.	R/W
b1	POD1	Output Data Register	1: The high level is output on the POi pin. (i = 7 to 0)	R/W
b2	POD2	Output Data Register		R/W
b3	POD3	Output Data Register		R/W
b4	POD4	Output Data Register		R/W
b5	POD5	Output Data Register		R/W
b6	POD6	Output Data Register		R/W
b7	POD7	Output Data Register		R/W

PPG0.PODRL stores pulse output values. For bits corresponding to pins that have been set for pulse output by PPG0.NDERL, the output trigger transfers the values in PPG0.NDRL to this register.

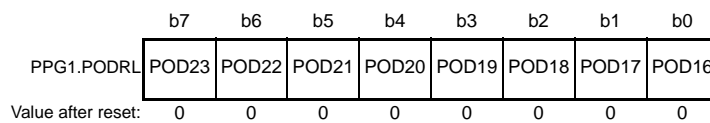
**PODi Bit (Output Data Register) (i = 7 to 0)**

When an output trigger is generated during PPG operation, the values of bits for which data transfer is enabled in the PPG0.NDERL register are transferred from the PPG0.NDRL register to this register. Writing from the CPU is impossible while any of the NDERi (i = 7 to 0) bits in PPG0.NDERL is 1. The initial output level for the pulse signal can be set when the value in the PPG0.NDERL register is 00h.

Address(es): 0008 81FAh



Address(es): 0008 81FBh



- PPG1.PODRH

Bit	Symbol	Bit Name	Description	R/W
b0	POD24	Output Data Register	0: The low level is output on the POi pin.	R/W
b1	POD25	Output Data Register	1: The high level is output on the POi pin.	R/W
b2	POD26	Output Data Register	(i = 31 to 24)	R/W
b3	POD27	Output Data Register		R/W
b4	POD28	Output Data Register		R/W
b5	POD29	Output Data Register		R/W
b6	POD30	Output Data Register		R/W
b7	POD31	Output Data Register		R/W

PPG1.PODRH stores pulse output values. For bits corresponding to pins that have been set for pulse output by PPG1.NDERH, the output trigger transfers the values in PPG1.NDRH to this register.

**PODi Bit (Output Data Register) (i = 31 to 24)**

When an output trigger is generated during PPG operation, the values of bits for which data transfer is enabled in the PPG1.NDERH register are transferred from the PPG1.NDRH register to this register. Writing from the CPU is impossible while any of the NDERi (i = 31 to 24) bits in PPG1.NDERH is 1. The initial output level for the pulse signal can be set when the value in the PPG1.NDERH register is 00h.

- PPG1.PODRL

Bit	Symbol	Bit Name	Description	R/W
b0	POD16	Output Data Register	0: The low level is output on the POi pin.	R/W
b1	POD17	Output Data Register	1: The high level is output on the POi pin.	R/W
b2	POD18	Output Data Register	(i = 23 to 16)	R/W
b3	POD19	Output Data Register		R/W
b4	POD20	Output Data Register		R/W
b5	POD21	Output Data Register		R/W
b6	POD22	Output Data Register		R/W
b7	POD23	Output Data Register		R/W

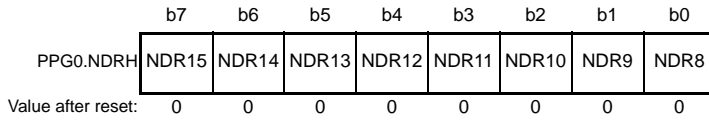
PPG1.PODRL stores pulse output values. For bits corresponding to pins that have been set for pulse output by PPG1.NDERL, the output trigger transfers the values in PPG1.NDRL to this register.

#### PODi Bit (Output Data Register) (i = 23 to 16)

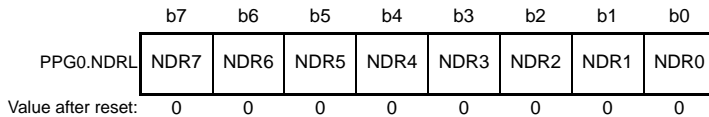
When an output trigger is generated during PPG operation, the values of bits for which data transfer is enabled in the PPG1.NDERL register are transferred from the PPG1.NDRL register to this register. Writing from the CPU is impossible while any of the NDERi (i = 23 to 16) bits in PPG1.NDERL is 1. The initial output level for the pulse signal can be set when the value in the PPG1.NDERL register is 00h.

## 25.2.4 Next Data Registers H (NDRH) Next Data Registers L (NDRL)

Address(es): 0008 81ECh, 0008 81EEh



Address(es): 0008 81EDh, 0008 81EFh



- PPG0.NDRH

PPG0.NDRH stores the next data for pulse output. The PPG0.NDRH address differs depending on whether pulse output groups have the same output trigger or different output triggers.

(1) When pulse output groups 2 and 3 have the same output trigger

If pulse output groups 2 and 3 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time.

Pulse output groups 2 and 3: 0008 81ECh

Bit	Symbol	Bit Name	Description	R/W
b0	NDR8	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRH.	R/W
b1	NDR9	Next Data Register		R/W
b2	NDR10	Next Data Register		R/W
b3	NDR11	Next Data Register		R/W
b4	NDR12	Next Data Register		R/W
b5	NDR13	Next Data Register		R/W
b6	NDR14	Next Data Register		R/W
b7	NDR15	Next Data Register		R/W

Note: • The address (0008 81EEh) to which PPG0.NDRH address has not been assigned is read as FFh, and cannot be modified.

(2) When pulse output groups 2 and 3 have different output triggers

If pulse output groups 2 and 3 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses.

Pulse output group 3: 0008 81ECh

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	NDR12	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRH.	R/W
b5	NDR13	Next Data Register		R/W
b6	NDR14	Next Data Register		R/W
b7	NDR15	Next Data Register		R/W

## Pulse output group 2: 0008 81EEh

Bit	Symbol	Bit Name	Description	R/W
b0	NDR8	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRH.	R/W
b1	NDR9	Next Data Register		R/W
b2	NDR10	Next Data Register		R/W
b3	NDR11	Next Data Register		R/W
b7 to b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

- PPG0.NDRL

PPG0.NDRL stores the next data for pulse output. The PPG0.NDRL address differs depending on whether pulse output groups have the same output trigger or different output triggers.

## (1) When pulse output groups 0 and 1 have the same output trigger

If pulse output groups 0 and 1 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time.

## Pulse output groups 0 and 1: 0008 81EDh

Bit	Symbol	Bit Name	Description	R/W
b0	NDR0	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRL.	R/W
b1	NDR1	Next Data Register		R/W
b2	NDR2	Next Data Register		R/W
b3	NDR3	Next Data Register		R/W
b4	NDR4	Next Data Register		R/W
b5	NDR5	Next Data Register		R/W
b6	NDR6	Next Data Register		R/W
b7	NDR7	Next Data Register		R/W

Note: • The address (0008 81EFh) to which PPG0.NDRL address has not been assigned is read as FFh, and cannot be modified.

## (2) When pulse output groups 0 and 1 have different output triggers

If pulse output groups 0 and 1 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses.

## Pulse output group 1: 0008 81EDh

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	NDR4	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRL.	R/W
b5	NDR5	Next Data Register		R/W
b6	NDR6	Next Data Register		R/W
b7	NDR7	Next Data Register		R/W

## Pulse output group 0: 0008 81EFh

Bit	Symbol	Bit Name	Description	R/W
b0	NDR0	Next Data Register	The output trigger specified by PPG0.PCR transfers the values in this register to the corresponding bits in PPG0.PODRL.	R/W
b1	NDR1	Next Data Register		R/W
b2	NDR2	Next Data Register		R/W
b3	NDR3	Next Data Register		R/W
b7 to b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

Address(es): 0008 81FCh, 0008 81FEh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG1.NDRH	NDR31	NDR30	NDR29	NDR28	NDR27	NDR26	NDR25	NDR24
Value after reset:	0	0	0	0	0	0	0	0

Address(es): 0008 81FDh, 0008 81FFh

	b7	b6	b5	b4	b3	b2	b1	b0
PPG1.NDRL	NDR23	NDR22	NDR21	NDR20	NDR19	NDR18	NDR17	NDR16
Value after reset:	0	0	0	0	0	0	0	0

- PPG1.NDRH

PPG1.NDRH stores the next data for pulse output. The PPG1.NDRH address differs depending on whether pulse output groups have the same output trigger or different output triggers.

(1) When pulse output groups 6 and 7 have the same output trigger

If pulse output groups 6 and 7 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time.

Pulse output groups 6 and 7: 0008 81FCh

Bit	Symbol	Bit Name	Description	R/W
b0	NDR24	Next Data Register	The output trigger specified by PPG1.PCR transfers the values in this register to the corresponding bits in PPG1.PODRH.	R/W
b1	NDR25	Next Data Register		R/W
b2	NDR26	Next Data Register		R/W
b3	NDR27	Next Data Register		R/W
b4	NDR28	Next Data Register		R/W
b5	NDR29	Next Data Register		R/W
b6	NDR30	Next Data Register		R/W
b7	NDR31	Next Data Register		R/W

Note: • The address (0008 81FEh) to which PPG1.NDRH address has not been assigned is read as FFh, and cannot be modified.

(2) When pulse output groups 6 and 7 have different output triggers

If pulse output groups 6 and 7 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses.

Pulse output group 7: 0008 81FCh

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	NDR28	Next Data Register	The output trigger specified by PPG1.PCR transfers the values in this register to the corresponding bits in PPG1.PODRH.	R/W
b5	NDR29	Next Data Register		R/W
b6	NDR30	Next Data Register		R/W
b7	NDR31	Next Data Register		R/W

## Pulse output group 6: 0008 81FEh

Bit	Symbol	Bit Name	Description	R/W
b0	NDR24	Next Data Register	The output trigger specified by PPG1.PCR transfers the values in this register to the corresponding bits in PPG1.PODRH.	R/W
b1	NDR25	Next Data Register		R/W
b2	NDR26	Next Data Register		R/W
b3	NDR27	Next Data Register		R/W
b7 to b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

- PPG1.NDRL

PPG1.NDRL stores the next data for pulse output. The PPG1.NDRL address differs depending on whether pulse output groups have the same output trigger or different output triggers.

## (1) When pulse output groups 4 and 5 have the same output trigger

If pulse output groups 4 and 5 have the same output trigger, all eight bits are mapped to the same address and can be accessed at one time.

## Pulse output groups 4 and 5: 0008 81FDh

Bit	Symbol	Bit Name	Description	R/W
b0	NDR16	Next Data Register	The output trigger specified by PPG1.PCR transfers the values in this register to the corresponding bits in PPG1.PODRL.	R/W
b1	NDR17	Next Data Register		R/W
b2	NDR18	Next Data Register		R/W
b3	NDR19	Next Data Register		R/W
b4	NDR20	Next Data Register		R/W
b5	NDR21	Next Data Register		R/W
b6	NDR22	Next Data Register		R/W
b7	NDR23	Next Data Register		R/W

Note: • The address (0008 81FFh) to which PPG1.NDRL address has not been assigned is read as FFh, and cannot be modified.

## (2) When pulse output groups 4 and 5 have different output triggers

If pulse output groups 4 and 5 have different output triggers, upper 4 bits and lower 4 bits are mapped to the different addresses.

## Pulse output group 5: 0008 81FDh

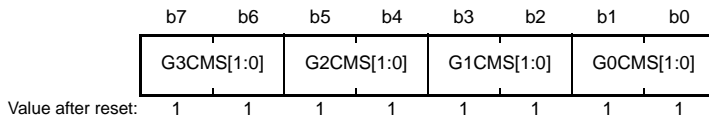
Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 1. The write value should be 1.	R/W
b4	NDR20	Next Data Register	The output trigger specified by PPG1.PCR transfers the values in this register to the corresponding bits in PPG1.PODRL.	R/W
b5	NDR21	Next Data Register		R/W
b6	NDR22	Next Data Register		R/W
b7	NDR23	Next Data Register		R/W

## Pulse output group 4: 0008 81FFh

Bit	Symbol	Bit Name	Description	R/W
b0	NDR16	Next Data Register	The output trigger specified by PPG1.PCR transfers the values in this register to the corresponding bits in PPG1.PODRL.	R/W
b1	NDR17	Next Data Register		R/W
b2	NDR18	Next Data Register		R/W
b3	NDR19	Next Data Register		R/W
b7 to b4	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

## 25.2.5 PPG Output Control Register (PCR)

Address(es): PPG0.PCR 0008 81E6h, PPG1.PCR 0008 81F6h



### • PPG0.PCR

Bit	Symbol	Bit Name	Description	R/W
b1, b0	G0CMS[1:0]	Group 0 Compare Match Select	b1 b0 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3	R/W
b3, b2	G1CMS[1:0]	Group 1 Compare Match Select	b3 b2 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3	R/W
b5, b4	G2CMS[1:0]	Group 2 Compare Match Select	b5 b4 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3	R/W
b7, b6	G3CMS[1:0]	Group 3 Compare Match Select	b7 b6 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3	R/W

### • PPG1.PCR

Bit	Symbol	Bit Name	Description	R/W
b1, b0	G0CMS[1:0]	Group 4 Compare Match Select	<ul style="list-style-type: none"> <li>• When the PTRSL bit in PPG1.PTRSLR is set to 0.</li> </ul> b1 b0 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3 <ul style="list-style-type: none"> <li>• When the PTRSL bit in PPG1.PTRSLR is set to 1.</li> </ul> b1 b0 0 0: Compare match in TPU0 0 1: Compare match in TPU1 1 0: Compare match in TPU2 1 1: Compare match in TPU3	R/W
b3, b2	G1CMS[1:0]	Group 5 Compare Match Select	<ul style="list-style-type: none"> <li>• When the PTRSL bit in PPG1.PTRSLR is set to 0.</li> </ul> b3 b2 0 0: Compare match in MTU0 0 1: Compare match in MTU1 1 0: Compare match in MTU2 1 1: Compare match in MTU3 <ul style="list-style-type: none"> <li>• When the PTRSL bit in PPG1.PTRSLR is set to 1.</li> </ul> b3 b2 0 0: Compare match in TPU0 0 1: Compare match in TPU1 1 0: Compare match in TPU2 1 1: Compare match in TPU3	R/W



Bit	Symbol	Bit Name	Description	R/W
b5, b4	G2CMS[1:0]	Group 6 Compare Match Select	<ul style="list-style-type: none"> <li>When the PTRSL bit in PPG1.PTRSLR is set to 0.               <ul style="list-style-type: none"> <li>b5 b4</li> <li>0 0: Compare match in MTU0</li> <li>0 1: Compare match in MTU1</li> <li>1 0: Compare match in MTU2</li> <li>1 1: Compare match in MTU3</li> </ul> </li> <li>When the PTRSL bit in PPG1.PTRSLR is set to 1.               <ul style="list-style-type: none"> <li>b5 b4</li> <li>0 0: Compare match in TPU0</li> <li>0 1: Compare match in TPU1</li> <li>1 0: Compare match in TPU2</li> <li>1 1: Compare match in TPU3</li> </ul> </li> </ul>	R/W
b7, b6	G3CMS[1:0]	Group 7 Compare Match Select	<ul style="list-style-type: none"> <li>When the PTRSL bit in PPG1.PTRSLR is set to 0.               <ul style="list-style-type: none"> <li>b7 b6</li> <li>0 0: Compare match in MTU0</li> <li>0 1: Compare match in MTU1</li> <li>1 0: Compare match in MTU2</li> <li>1 1: Compare match in MTU3</li> </ul> </li> <li>When the PTRSL bit in PPG1.PTRSLR is set to 1.               <ul style="list-style-type: none"> <li>b7 b6</li> <li>0 0: Compare match in TPU0</li> <li>0 1: Compare match in TPU1</li> <li>1 0: Compare match in TPU2</li> <li>1 1: Compare match in TPU3</li> </ul> </li> </ul>	R/W

PPGn.PCR (n = 0, 1) selects pulse output trigger signals on a group-by-group basis. For details on output trigger selection, see section 25.2.6, PPG Output Mode Register (PMR).

## 25.2.6 PPG Output Mode Register (PMR)

Address(es): PPG0.PMR 0008 81E7h, PPG1.PMR 0008 81F7h

b7	b6	b5	b4	b3	b2	b1	b0
G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV

Value after reset: 1 1 1 1 0 0 0 0

- PPG0.PMR

Bit	Symbol	Bit Name	Description	R/W
b0	G0NOV	Group 0 Non-Overlap	0: Normal operation (Output values updated on compare match A in the selected MTUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3)	R/W
b1	G1NOV	Group 1 Non-Overlap	0: Normal operation (Output values updated on compare match A in the selected MTUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3)	R/W
b2	G2NOV	Group 2 Non-Overlap	0: Normal operation (Output values updated on compare match A in the selected MTUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3)	R/W
b3	G3NOV	Group 3 Non-Overlap	0: Normal operation (Output values updated on compare match A in the selected MTUn) 1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3)	R/W
b4	G0INV	Group 0 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b5	G1INV	Group 1 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b6	G2INV	Group 2 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b7	G3INV	Group 3 Output Polarity Change	0: Inverted output 1: Direct output	R/W

- PPG1.PMR

Bit	Symbol	Bit Name	Description	R/W
b0	G0NOV	Group 4 Non-Overlap	<ul style="list-style-type: none"> <li>• When the PPG1.PTRSLR.PTRSL bit is 0</li> <li>0: Normal operation (Output values updated on compare match A in the selected MTUn)</li> <li>1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3)</li> <li>• When the PPG1.PTRSLR.PTRSL bit is 1</li> <li>0: Normal operation (Output values updated on compare match A in the selected TPU<sub>n</sub>)</li> <li>1: Non-overlapping operation (Output values updated on compare match A or B in the selected TPU<sub>n</sub>) (n = 0 to 3)</li> </ul>	R/W
b1	G1NOV	Group 5 Non-Overlap	<ul style="list-style-type: none"> <li>• When the PPG1.PTRSLR.PTRSL bit is 0</li> <li>0: Normal operation (Output values updated on compare match A in the selected MTUn)</li> <li>1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3)</li> <li>• When the PPG1.PTRSLR.PTRSL bit is 1</li> <li>0: Normal operation (Output values updated on compare match A in the selected TPU<sub>n</sub>)</li> <li>1: Non-overlapping operation (Output values updated on compare match A or B in the selected TPU<sub>n</sub>) (n = 0 to 3)</li> </ul>	R/W
b2	G2NOV	Group 6 Non-Overlap	<ul style="list-style-type: none"> <li>• When the PPG1.PTRSLR.PTRSL bit is 0</li> <li>0: Normal operation (Output values updated on compare match A in the selected MTUn)</li> <li>1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3)</li> <li>• When the PPG1.PTRSLR.PTRSL bit is 1</li> <li>0: Normal operation (Output values updated on compare match A in the selected TPU<sub>n</sub>)</li> <li>1: Non-overlapping operation (Output values updated on compare match A or B in the selected TPU<sub>n</sub>) (n = 0 to 3)</li> </ul>	R/W
b3	G3NOV	Group 7 Non-Overlap	<ul style="list-style-type: none"> <li>• When the PPG1.PTRSLR.PTRSL bit is 0</li> <li>0: Normal operation (Output values updated on compare match A in the selected MTUn)</li> <li>1: Non-overlapping operation (Output values updated on compare match A or B in the selected MTUn) (n = 0 to 3)</li> <li>• When the PPG1.PTRSLR.PTRSL bit is 1</li> <li>0: Normal operation (Output values updated on compare match A in the selected TPU<sub>n</sub>)</li> <li>1: Non-overlapping operation (Output values updated on compare match A or B in the selected TPU<sub>n</sub>) (n = 0 to 3)</li> </ul>	R/W

Bit	Symbol	Bit Name	Description	R/W
b4	G0INV	Group 4 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b5	G1INV	Group 5 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b6	G2INV	Group 6 Output Polarity Change	0: Inverted output 1: Direct output	R/W
b7	G3INV	Group 7 Output Polarity Change	0: Inverted output 1: Direct output	R/W

PPGn.PMR (n = 0, 1) selects the pulse output mode of the PPG on a group-by-group basis.

While inverted output is selected, a low-level pulse is output when the values in PPGn.PODRH and PPGn.PODRL are 1, and a high-level pulse is output when the values in PPGn.PODRH and PPGn.PODRL are 0.

In addition, when non-overlapping operation is selected, the PPG updates its output values on compare match A or B in an MTU or TPU channel that functions as an output trigger.

For details, see section 25.3.4, Non-Overlapping Pulse Output.

### 25.3 Operation

Figure 25.4 shows a schematic diagram of the PPG.

PPG pulse output is enabled when the corresponding bits in PPGn.NDERH and PPGn.NDERL ( $n = 0, 1$ ) are set to 1 (data transfer is enabled).

An initial output value is determined by the initial settings in the corresponding PPGn.PODRH and PPGn.PODRL.

When the compare match event selected in PPGn.PCR occurs, the output values are updated by transfer of the values in the corresponding PPGn.NDRH and PPGn.NDRL to PPGn.PODRH and PPGn.PODRL, respectively.

Consecutive output of up to 16 bits of data is possible by writing new output data to PPGn.NDRH and PPGn.NDRL before the next compare match.

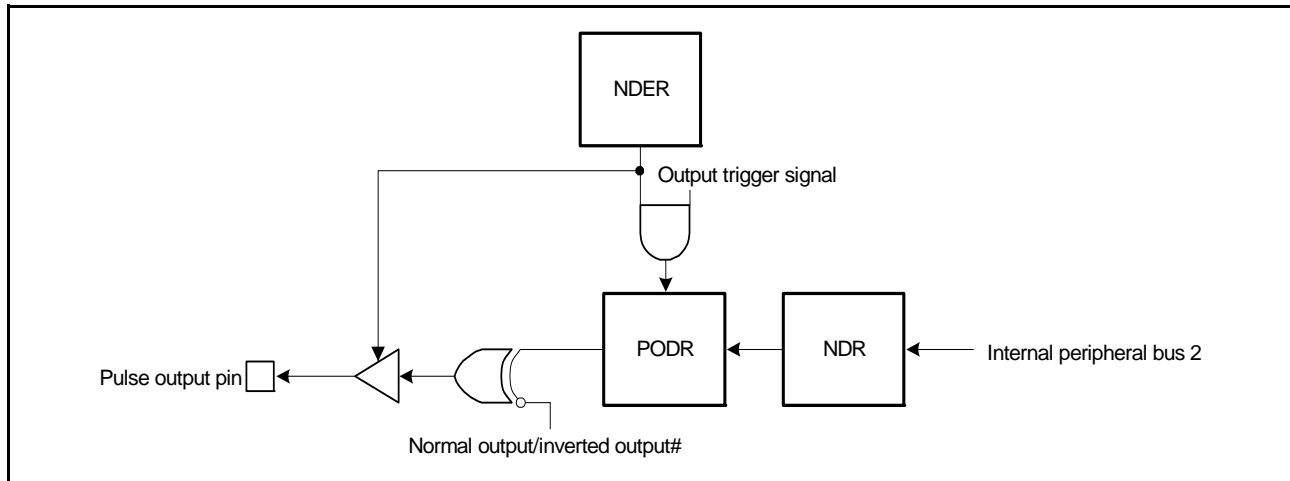


Figure 25.4 Schematic Diagram of PPG

### 25.3.1 Output Timing

When the selected compare match event occurs while pulse output is enabled, the values in PPGn.NDRH and PPGn.NDRL ( $n = 0, 1$ ) are transferred to PPGn.PODRH and PPGn.PODRL, respectively, and then output on the corresponding pins.

Figure 25.5 shows the timing of the above operation. In this case, the timing when compare match A triggers normal output from groups 2 and 3 is shown.

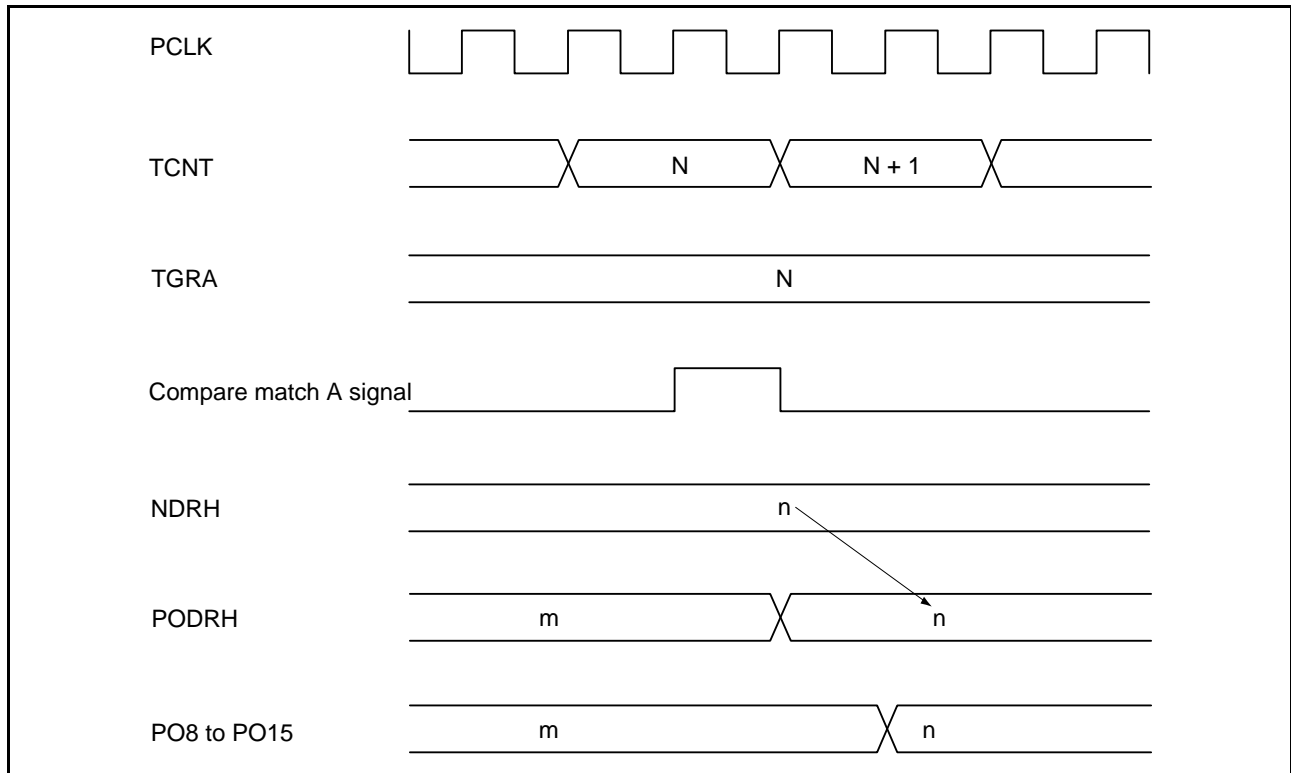


Figure 25.5 Timing of Transfer and Output of the Values in NDR (Example)

### 25.3.2 Sample Setup Procedure for Normal Pulse Output

Figure 25.6 and Figure 25.7 show sample procedures for setting normal pulse output.

#### (1) PPG0 Setting

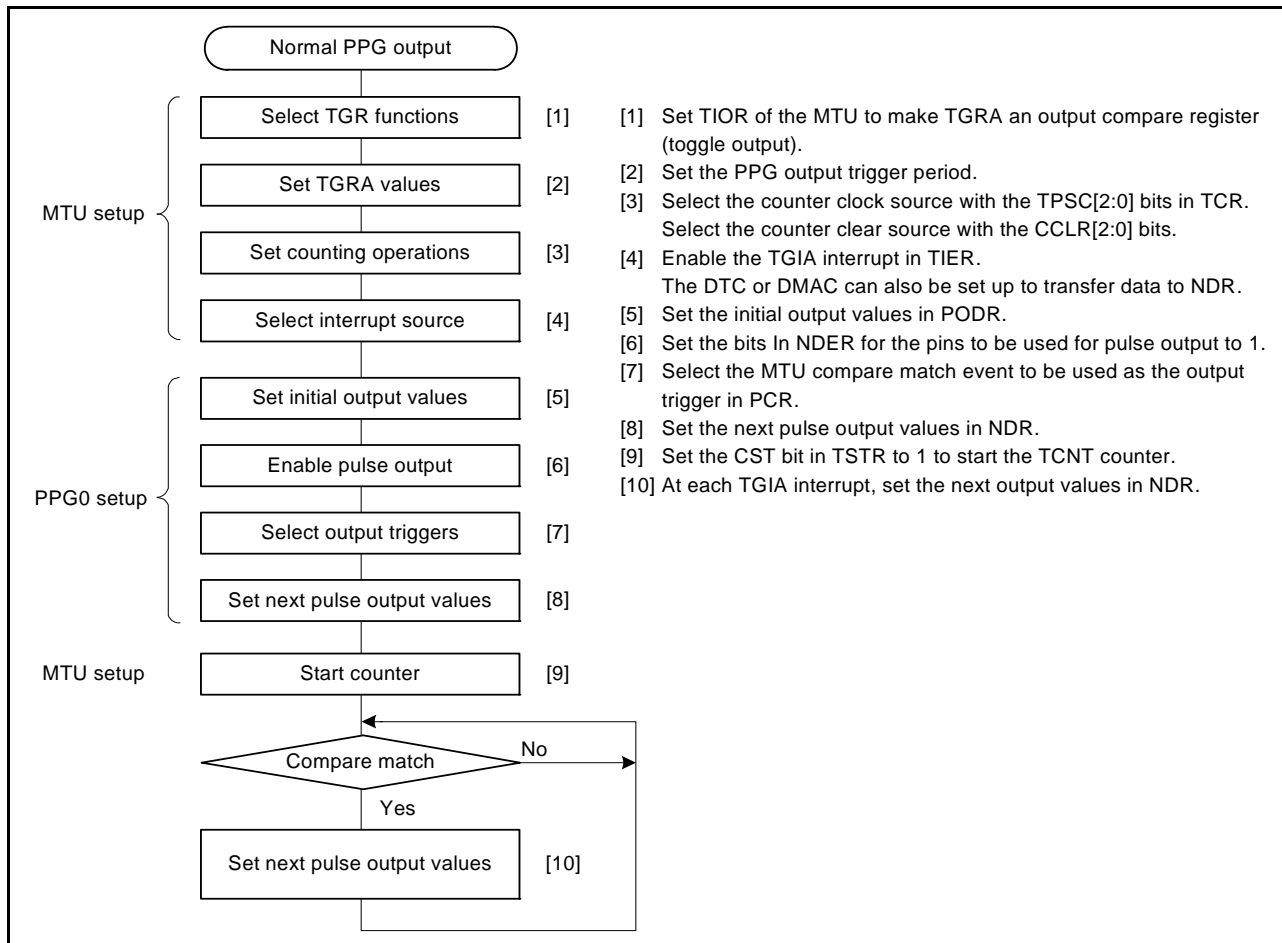


Figure 25.6 Sample Setup Procedure for Normal Pulse Output (PPG0 Setting)

(2) PPG1 Setting

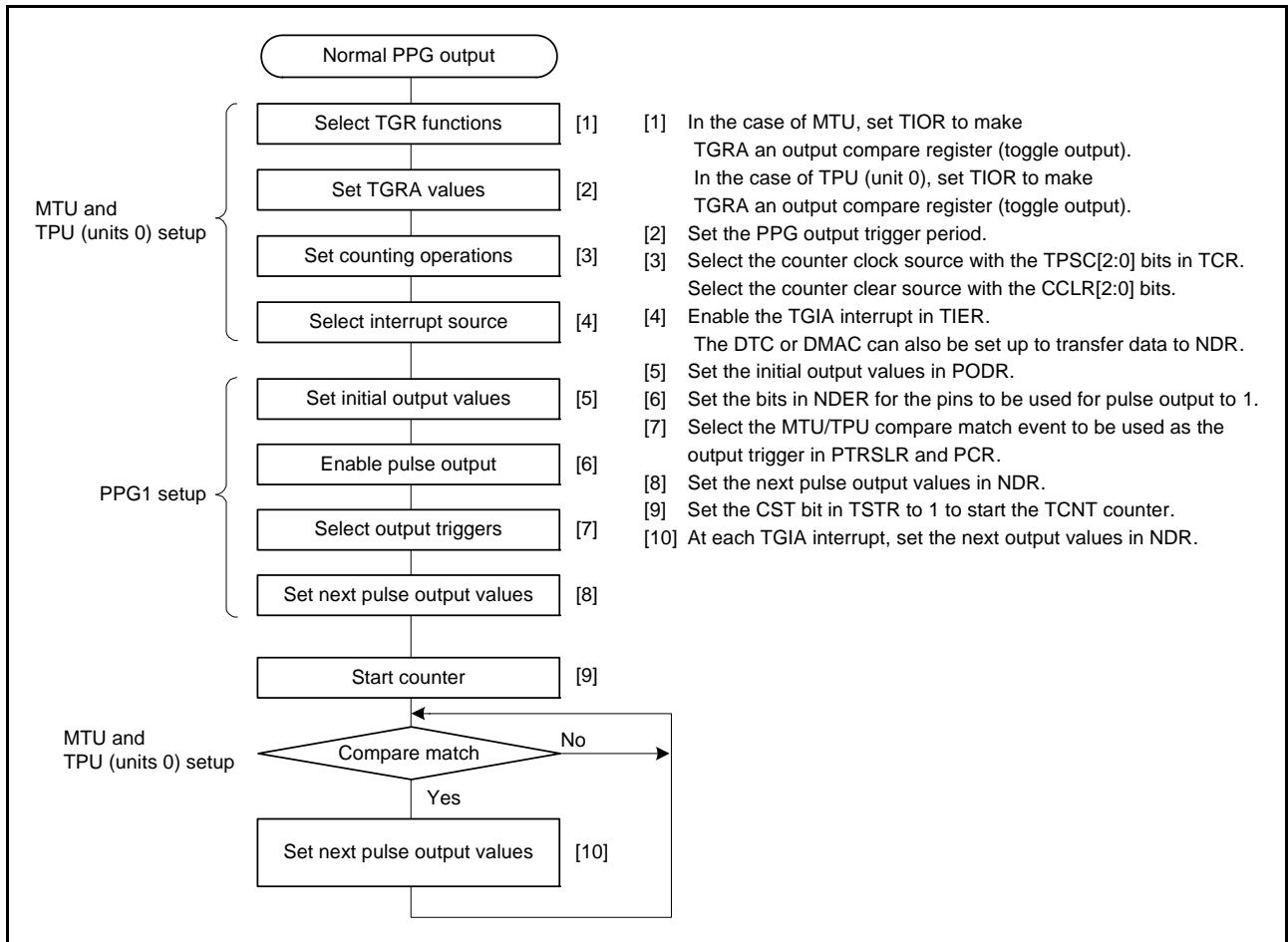
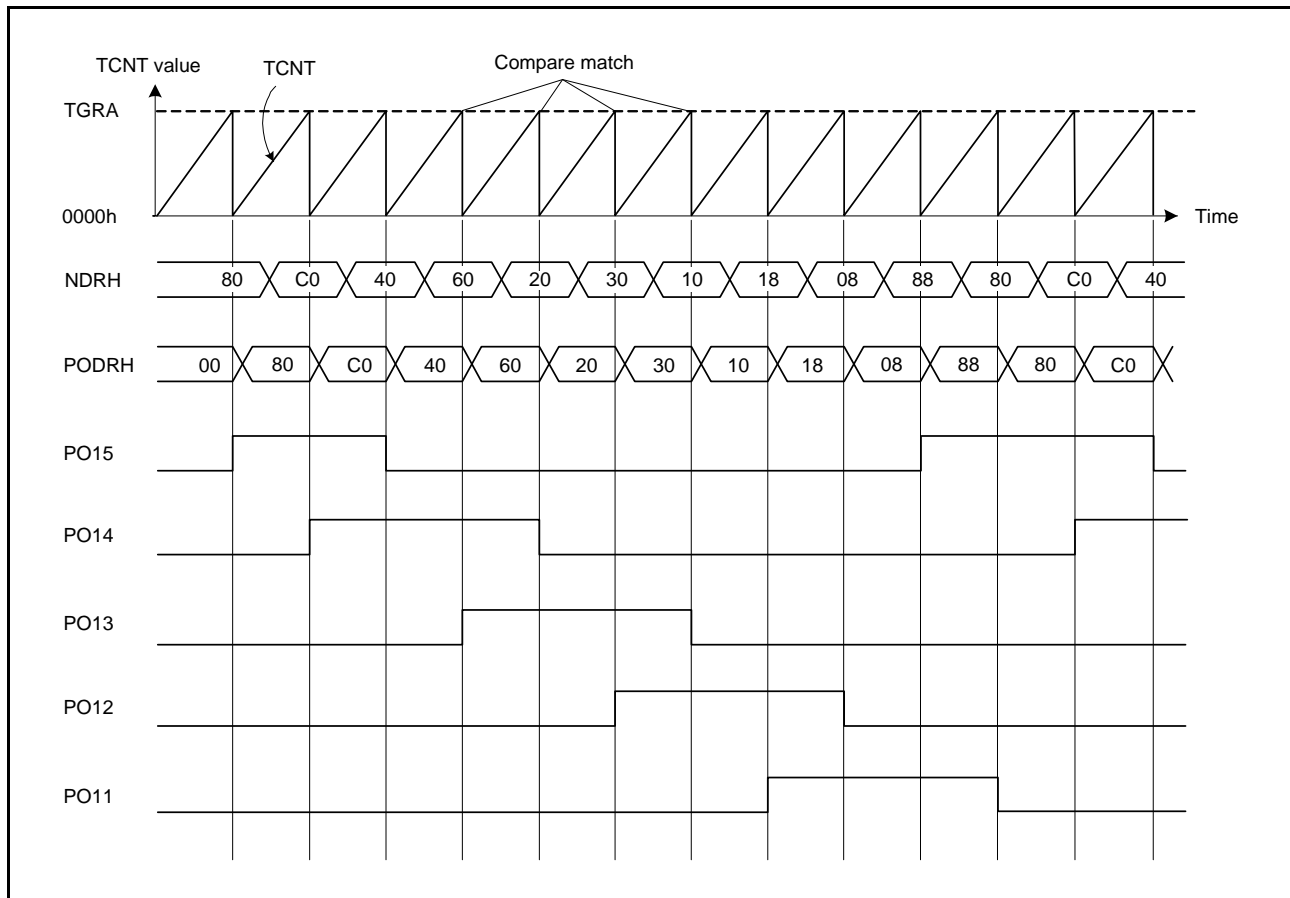


Figure 25.7 Sample Setup Procedure for Normal Pulse Output (PPG1 Setting)



### 25.3.3 Example of Normal Pulse Output (Example of Five-Phase Pulse Output)

Figure 25.8 shows an example in which pulse output from the PPG0 is used for cyclic five-phase pulse output.



**Figure 25.8 Example of Normal Pulse Output (Example of Five-Phase Pulse Output)**

1. Set an output compare register of the MTUn.TGRA ( $n = 0$  to 3) of MTU so that the corresponding compare match signal is the output trigger. Set a cycle in TGRA so that the counter will be cleared by compare match A. Set the MTUn.TIER.TGIEA bit to 1 to enable the compare match/input capture A (TGIA) interrupt.
2. Write F8h to PPG0.NDRH, and set the G3CMS[1:0] and G2CMS[1:0] bits in PPG0.PCR to select the respective compare matches in the MTUn selected in the previous step to be the output triggers. Write output data 80h to PPG0.NDRH.
3. The timer counter in the MTU starts. When compare match A occurs, the values in PPG0.NDRH are transferred to PPG0.PODRH and output. The TGIA interrupt handling routine writes the next output data C0h to PPG0.NDRH.
4. Five-phase pulse output (one or two phases active at a time) can be obtained subsequently by writing 40h, 60h, 20h, 30h, 10h, 18h, 08h, 88h... at successive TGIA interrupts.

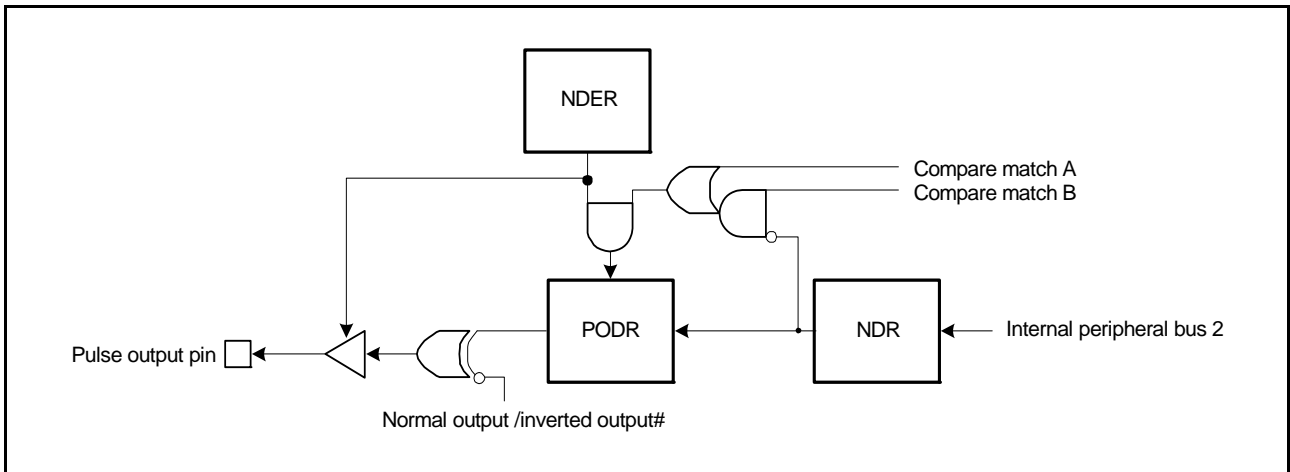
If the DTC or DMAC is set for activation by the TGIA interrupt, pulse output can be obtained without imposing a load on the CPU.

### 25.3.4 Non-Overlapping Pulse Output

During non-overlapping operation, data transfer from PPGn.NDRH and PPGn.NDRL (n = 0, 1) to PPGn.PODRH and PPGn. PODRL is performed as follows.

- On compare match A, the values in PPGn.NDRH and PPGn.NDRL are always transferred to PPGn.PODRH and PPGn. PODRL.
- On compare match B, data transfer proceeds for bits in PPGn.NDRH and PPGn.NDRL that have the value 0. It does not proceed for bits having the value 1.

Figure 25.9 shows the non-overlapping pulse output operation.

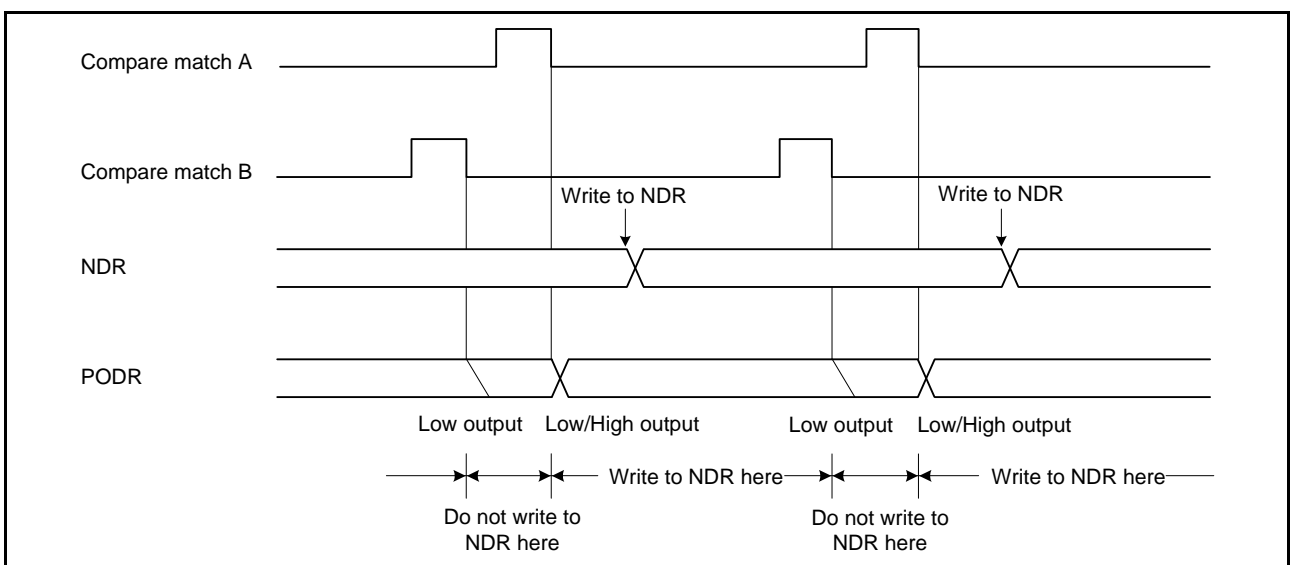


**Figure 25.9 Non-Overlapping Pulse Output**

Therefore, compare match B before compare match A allows 0-valued data to be transferred in advance of 1-valued data. Do not change the values in PPGn.NDRH and PPGn.NDRL during the interval from compare match B to compare match A (the non-overlap margin).

To transfer 0-valued data in advance of 1-valued data, write the next data to PPGn.NDRH and PPGn.NDRL from within the TGIA interrupt handling routine or by using a TGIA interrupt to activate transfer by the DTC or DMAC. In any case, the next data must be written before the next compare match B occurs.

Figure 25.10 shows the timing of the above procedure.



**Figure 25.10 Non-Overlapping Operation and Write Timing to PPGn.NDRH and PPGn.NDRL**

### 25.3.5 Sample Setup Procedure for Non-Overlapping Pulse Output

Figure 25.11 and Figure 25.12 show sample procedures for setting up non-overlapping pulse outputs.

(1) PPG0 Setting

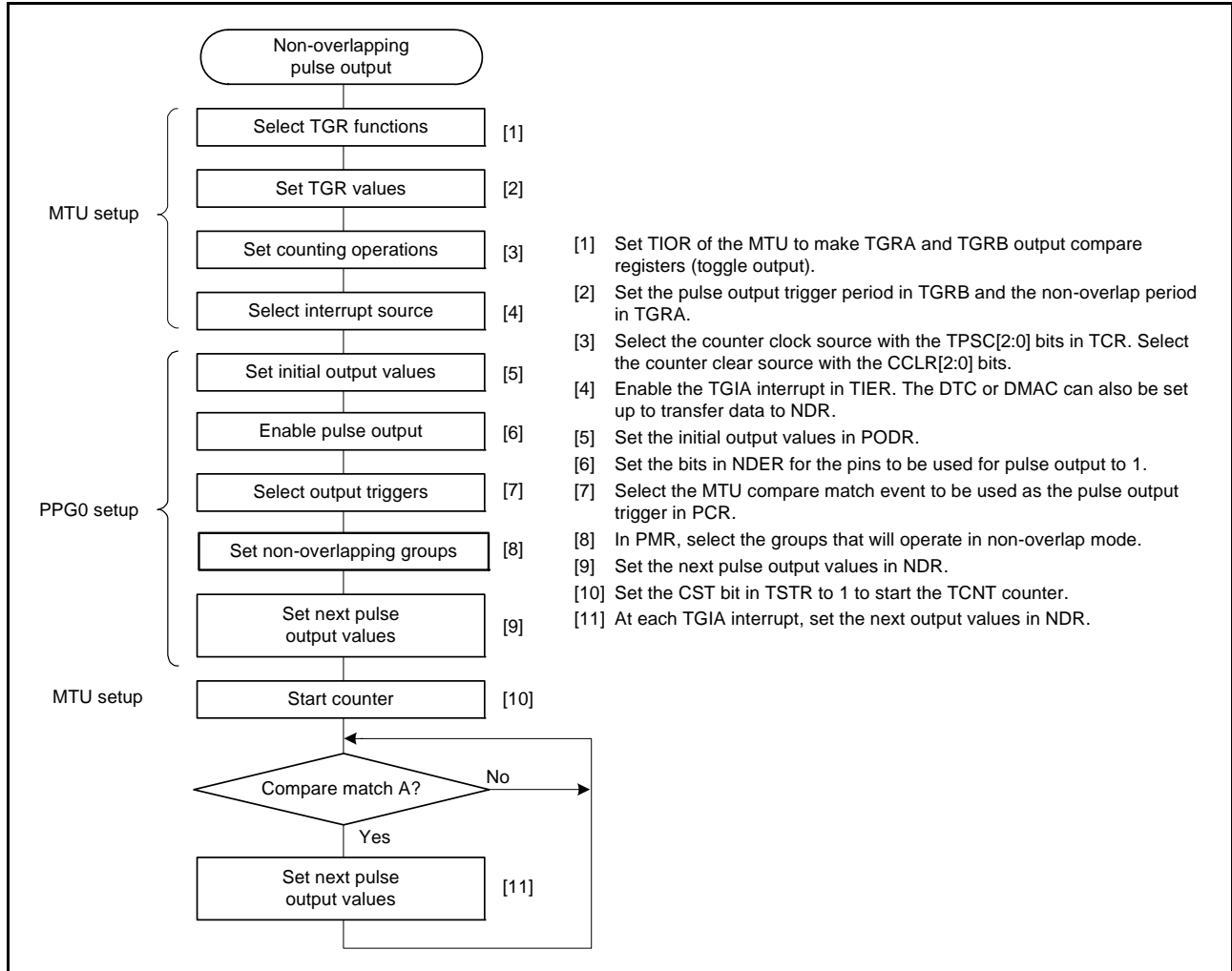


Figure 25.11 Sample Setup Procedure for Non-Overlapping Pulse Output (PPG0 Setting)

(2) PPG1 Setting

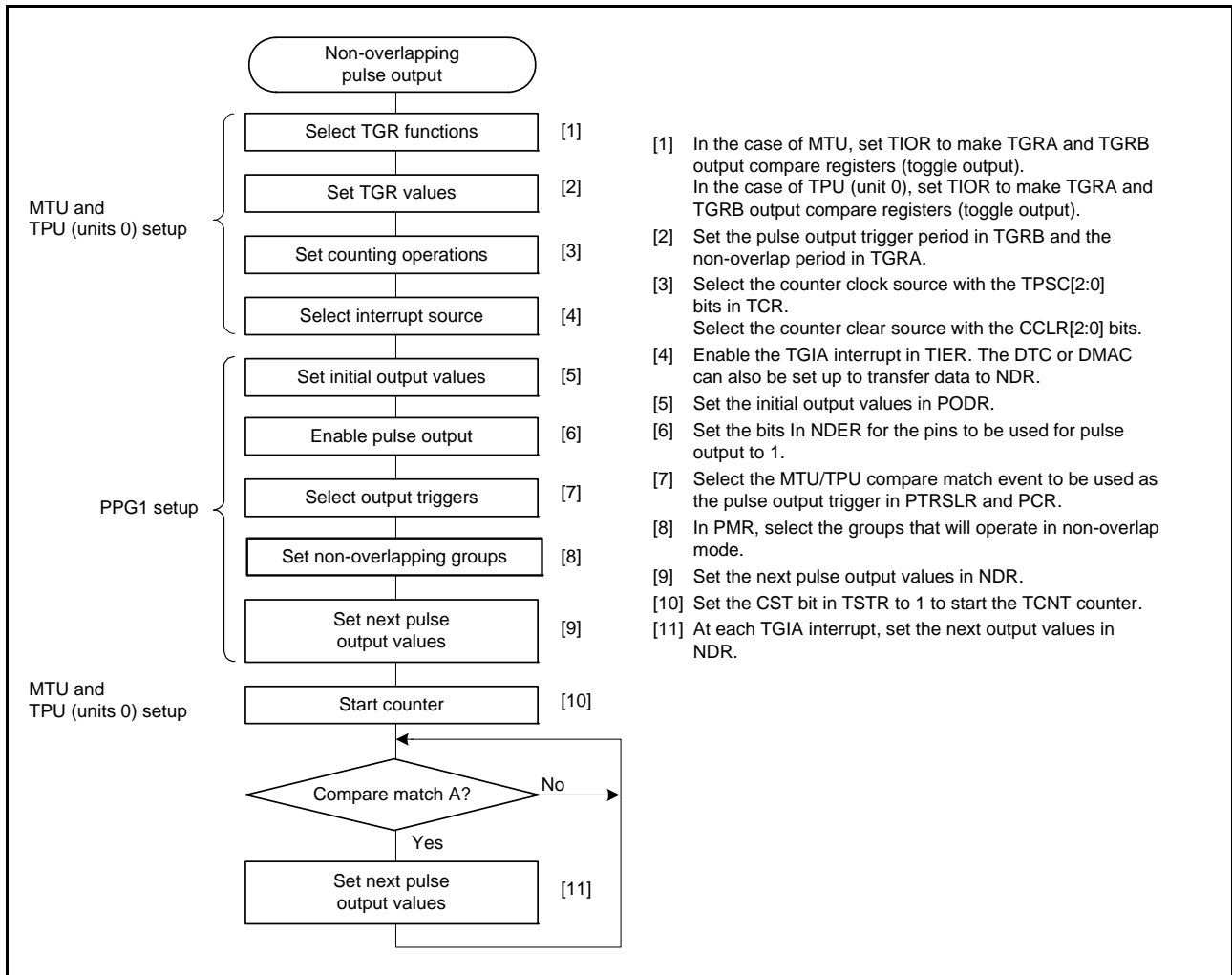
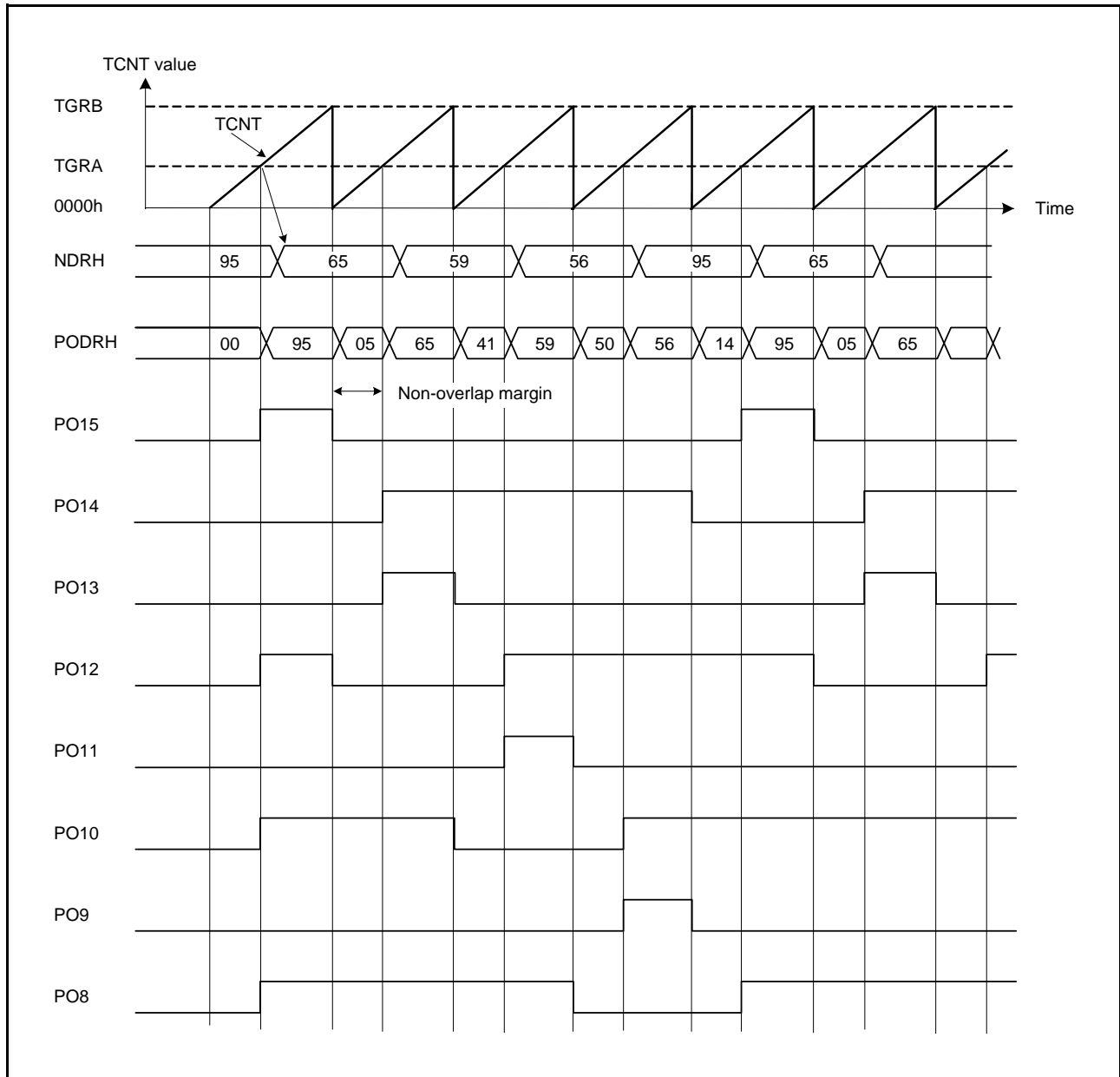


Figure 25.12 Sample Setup Procedure for Non-Overlapping Pulse Output (PPG1 Setting)

### 25.3.6 Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output)

Figure 25.13 shows an example in which pulse output from the PPG0 is used for four-phase complementary non-overlapping pulse output.



**Figure 25.13 Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output)**

1. Set output compare registers of the MTUn.TGRA and MTUn.TGRB (n = 0 to 3) of MTU so that the corresponding compare match signals are the output triggers. Set the trigger period in TGRB and the non-overlap margin in TGRA, and set the counter to be cleared by compare match B. Set the MTUn.TIER.TGIEA bit to 1 to enable the compare match/input capture A (TGIA) interrupt.
2. Write FFh in PPG0.NDERH, and set the G3CMS[1:0] and G2CMS[1:0] bits in PPG0.PCR to select the respective compare matches in the MTUn selected in the previous step to be the output triggers. Set the G3NOV and G2NOV bits in PPG0.PMR to 1 to select non-overlapping outputs. Write output data 95h in PPG0.NDRH.
3. The timer counter in the MTU starts. When a compare match with TGRB occurs, outputs change from high to low. When a compare match with TGRA occurs, outputs change from low to high (the change from low to high is delayed by the value set in TGRA).  
The TGIA interrupt handling routine writes the next output data 65h in PPG0.NDRH.
4. Four-phase complementary non-overlapping pulse output can be obtained subsequently by writing 59h, 56h, 95h... at successive TGIA interrupts.  
If the DTC or DMAC is set for activation by the TGIA interrupt, pulse output can be obtained without imposing a load on the CPU.

### 25.3.7 Inverted Pulse Output

When the G3INV, G2INV, G1INV, and G0INV bits in PPG0.PMR are cleared to 0, the values that are the inverse of the respective values in PPG0.PODRH and PPG0.PODRL can be output.

Figure 25.14 shows the outputs when the G3INV and G2INV bits are cleared to 0 in addition to the settings in Figure 25.13.

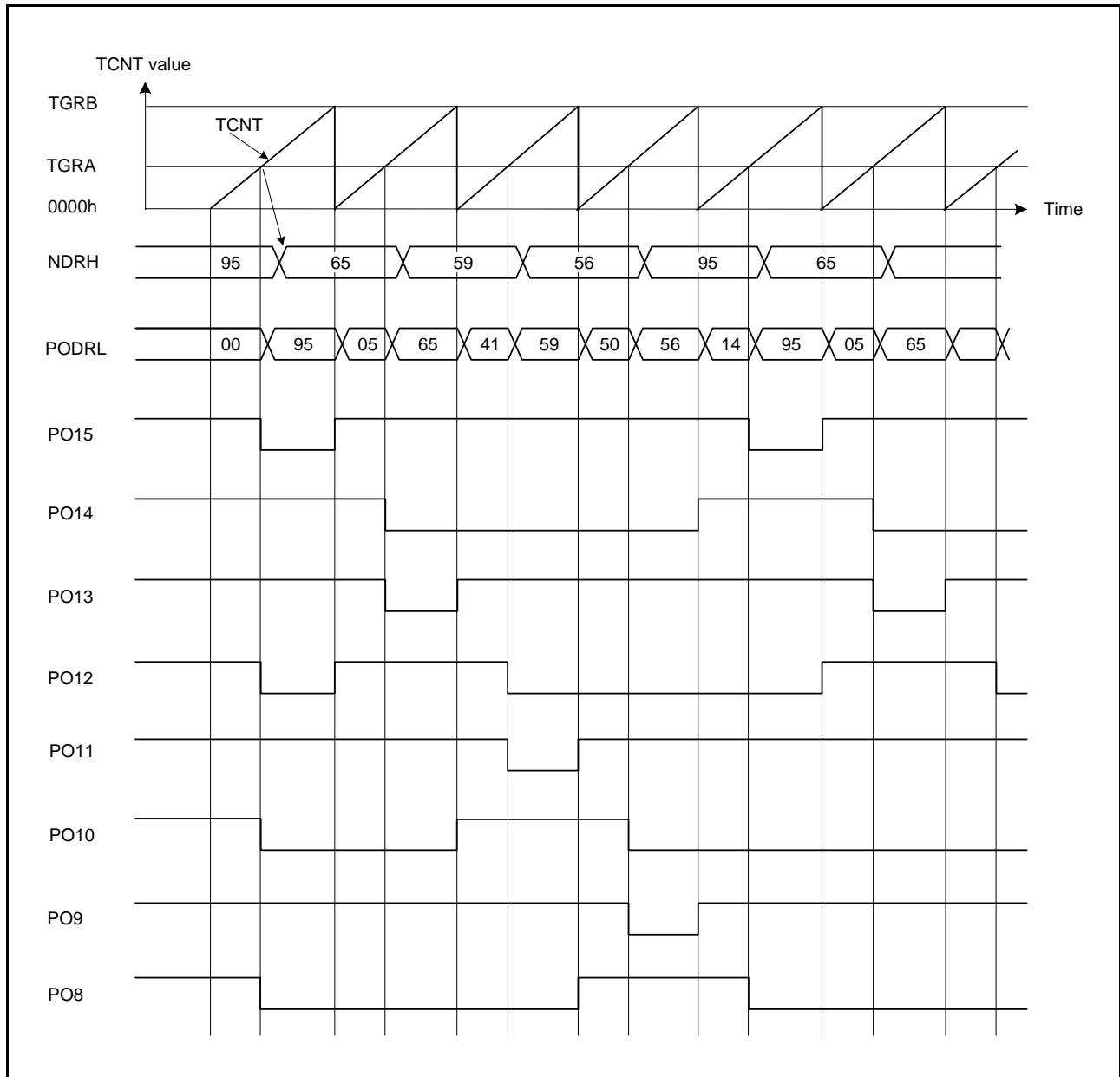


Figure 25.14 Inverted Pulse Output (Example)

### 25.3.8 Pulse Output Triggered by Input Capture

Pulse output from the PPG0 can be triggered by the MTU input capture as well as by compare match. When MTUn.TGRA (n = 0 to 3) functions as an input capture register in the MTU channel selected by PPG0.PCR, pulse output is triggered by the input capture signal.

Figure 25.15 shows the timing of pulse output triggered by input capture.

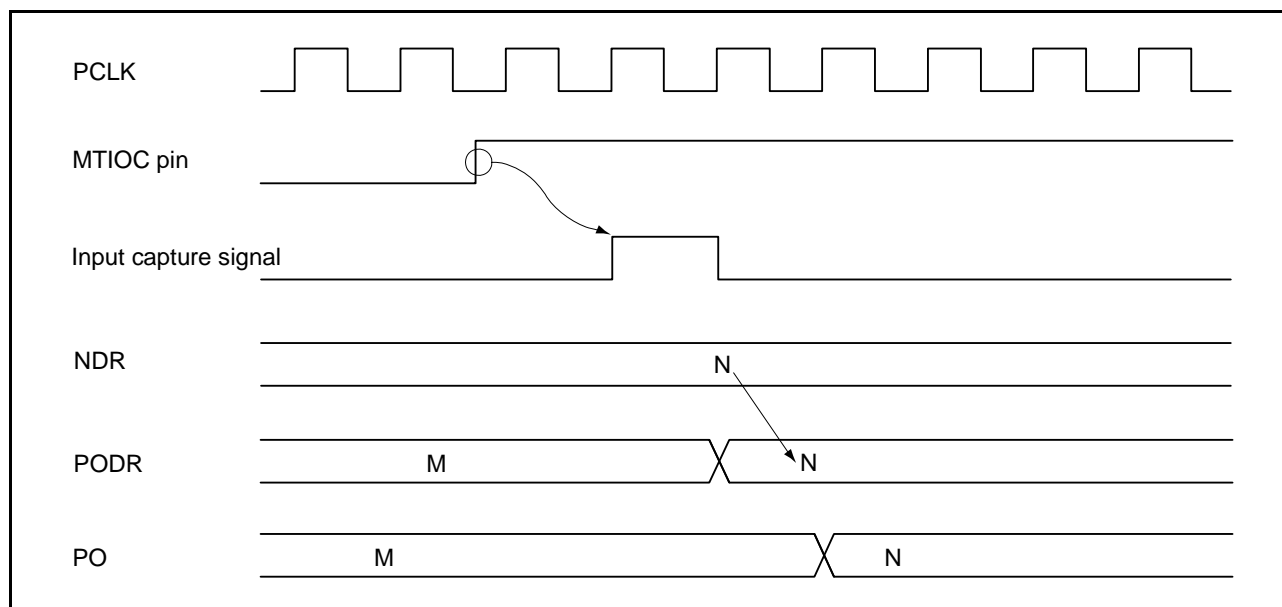


Figure 25.15 Timing of Pulse Output Triggered by Input Capture (Example)

## 25.4 Usage Note

### 25.4.1 Module-Stop Function Setting

Operation of the PPG can be disabled or enabled by the module stop control register. The initial setting is for operation of the PPG to be stopped. Register access is enabled by clearing module-stop state. For details, see section 11, Low Power Consumption.



## 26. 8-Bit Timer (TMR)

The RX630 Group has two units (unit 0, unit 1) of an on-chip 8-bit timer (TMR) module that comprise two 8-bit counter channels, totaling four channels. The 8-bit timer module can be used to count external events and also be used as a multifunction timer in a variety of applications, such as generation of counter reset, interrupt requests, and pulse output with a desired duty cycle using a compare-match signal with two registers.

Unit 0 and unit 1 can generate a baud rate clock signal for the SCI and have the same functions.

### 26.1 Overview

Table 26.1 lists the specifications of the TMR.

Figure 26.1 shows a block diagram of the 8-bit timer module (unit 0), and Figure 26.2 shows that of the 8-bit timer module (unit 1).

**Table 26.1 Specifications of TMR**

Item	Description
Count clock	<ul style="list-style-type: none"> <li>Frequency dividing clock: PCLK/1, PCLK/2, PCLK/8, PCLK/32, PCLK/64, PCLK/1024, PCLK/8192</li> <li>External clock</li> </ul>
Number of channels	(8 bits × 2 channels) × 2 units
Compare match	<ul style="list-style-type: none"> <li>8-bit mode (compare match A, compare match B)</li> <li>16-bit mode (compare match A, compare match B)</li> </ul>
Counter clear	Selected by compare match A or B, or an external reset signal.
Timer output	Output pulses with a desired duty cycle or PWM output
Cascading of two channels	<ul style="list-style-type: none"> <li>16-bit count mode 16-bit timer using TMR0 for the upper 8 bits and TMR1 for the lower 8 bits (TMR2 for the upper 8 bits and TMR3 for the lower 8 bits)</li> <li>Compare match count mode TMR1 can be used to count TMR0 compare matches (TMR3 can be used to count TMR2 compare matches).</li> </ul>
Interrupt sources	Compare match A, compare match B, and overflow
DTC activation	DTC can be activated by compare match A interrupts or compare match B interrupts.
Generation of trigger to start A/D converter conversion	Compare match A of TMR0 and TMR2*1
Capable of generating baud rate clock for SCI	Generates baud rate clock for SCI.*2
Low power consumption facilities	Each unit can be placed in a module-stop state.

Note 1. For details, see section 38, 12-Bit A/D Converter (S12ADa), and section 39, 10-Bit A/D Converter (ADb).

Note 2. For details, see section 32, Serial Communications Interface (SCId, SCId).

Table 26.2 Pin Configuration of TMR

Item		Unit 0			Unit 1		
		8 Bits		16 Bits	8 Bits		16 Bits
Counter mode							
Channel		TMR0	TMR1	TMR0 + TMR1	TMR2	TMR3	TMR2 + TMR3
Count clock		PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMC10	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMC11	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMC11	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMC12	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMC13	PCLK/1 PCLK/2 PCLK/8 PCLK/32 PCLK/64 PCLK/1024 PCLK/8192 TMC13
Counter clear		TMR0.TCORA TMR0.TCORB TMR10	TMR1.TCORA TMR1.TCORB TMR11	TMR0.TCORA + TMR1.TCORA  TMR0.TCORB+ TMR1.TCORB  TMR10	TMR2.TCORA TMR2.TCORB TMR12	TMR3.TCORA TMR3.TCORB TMR13	TMR2.TCORA + TMR3.TCORA  TMR2.TCORB+ TMR3.TCORB  TMR12
Compare match	Compare match A	○	○	○	○	○	○
	Compare match B	○	○	○	○	○	○
Timer output	Low output	○	○	○	○	○	○
	High output	○	○	○	○	○	○
	Toggle output	○	○	○	○	○	○
DTC activation	Compare match A	○	○	○	○	○	○
	Compare match B	○	○	○	○	○	○
	TCNT overflow	—	—	—	—	—	—
Interrupt	Compare match A	CMIA0	CMIA1	CMIA0	CMIA2	CMIA3	CMIA2
	Compare match B	CMIB0	CMIB1	CMIB0	CMIB2	CMIB3	CMIB2
	TCNT overflow	OVI0	OVI1	OVI0	OVI2	OVI3	OVI2
Cascaded connection		TMR1 overflow	TMR0 compare match A	—	TMR3 overflow	TMR2 compare match A	—
A/D converter conversion start trigger*1		○	—	○	○	—	○
SCI baud rate clock generation*2		○		—	○		—
Module-stop setting*3		MSTPCRA.MSTPA5 bit (unit 0), MSTPCRA.MSTPA4 bit (unit 1)					

○: Possible

—: Impossible

Note 1. For details, see section 38, 12-Bit A/D Converter (S12ADa), and section 39, 10-Bit A/D Converter (ADb).

Note 2. For details, see section 32, Serial Communications Interface (SC1c, SC1d).

Note 3. For details, see section 11, Low Power Consumption.

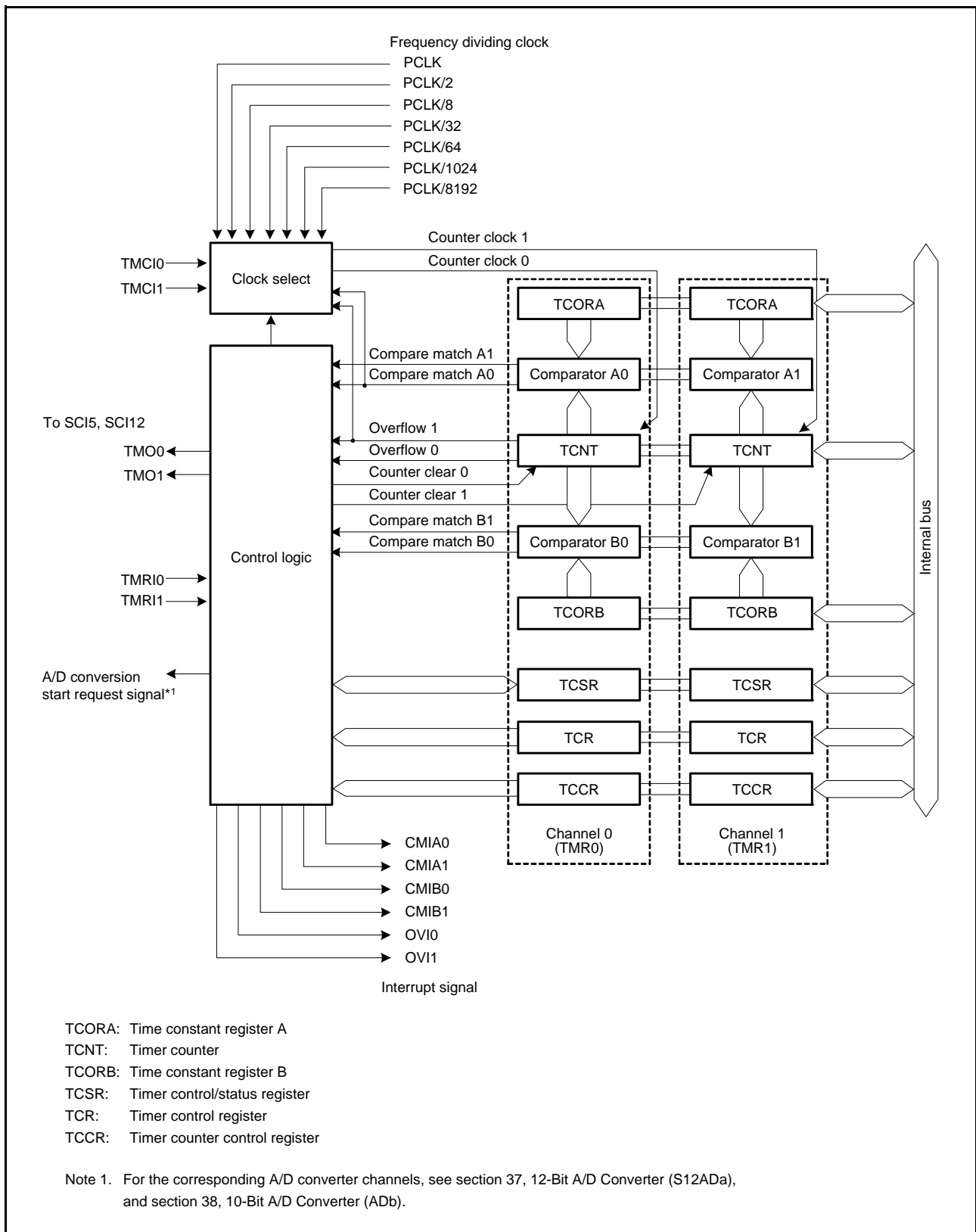


Figure 26.1 Block Diagram of TMR (Unit 0)

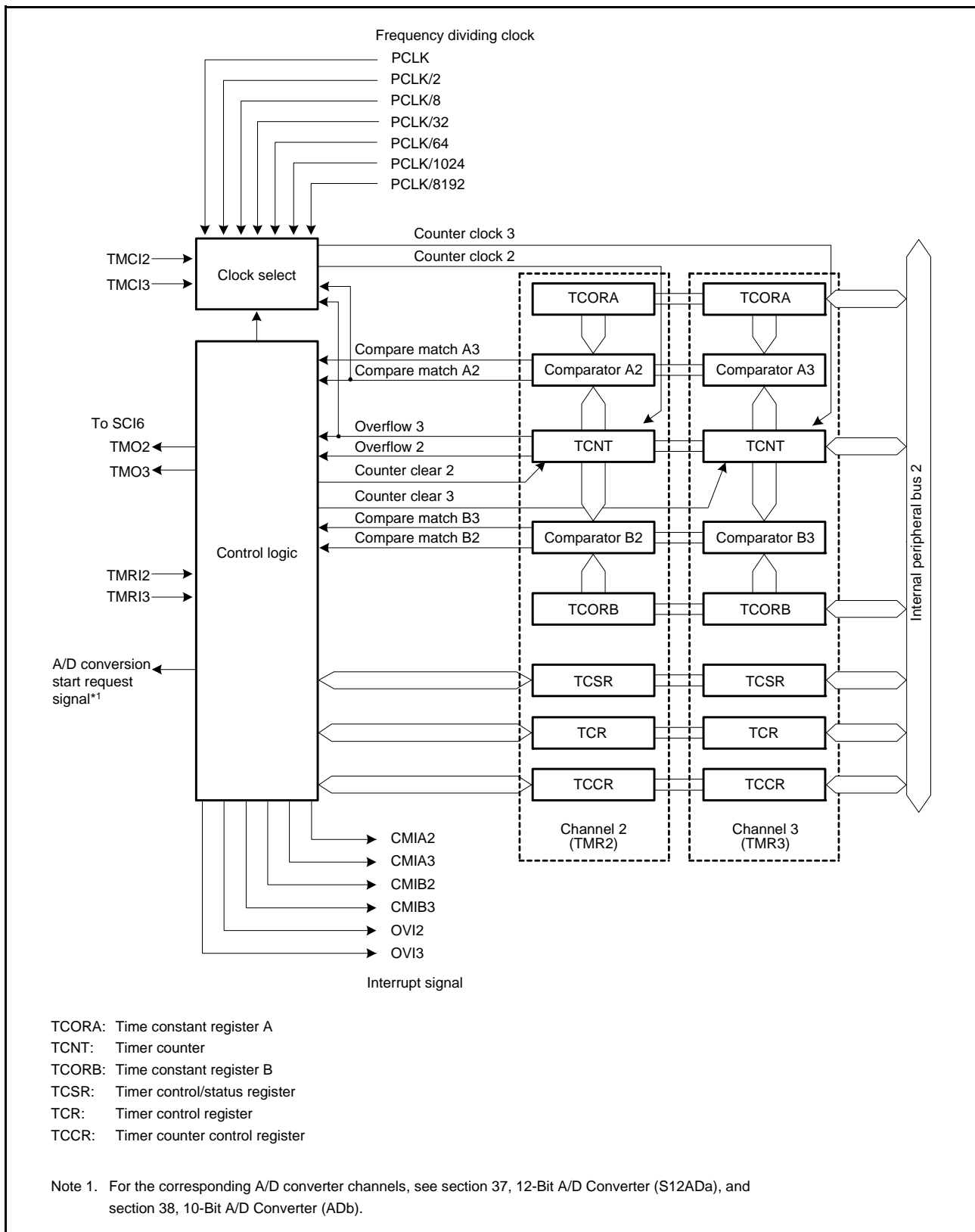


Figure 26.2 Block Diagram of TMR (Unit 1)

Table 26.3 lists the input/output pins of the TMR.

**Table 26.3 Pin Configuration of TMR**

Unit	Channel	Pin Name	I/O	Description
0	TMR0	TMO0	Output	Outputs compare match
		TMC10	Input	Inputs external clock for counter
		TMR10	Input	Inputs external reset to counter
	TMR1	TMO1	Output	Outputs compare match
		TMC11	Input	Inputs external clock for counter
		TMR11	Input	Inputs external reset to counter
1	TMR2	TMO2	Output	Outputs compare match
		TMC12	Input	Inputs external clock for counter
		TMR12	Input	Inputs external reset to counter
	TMR3	TMO3	Output	Outputs compare match
		TMC13	Input	Inputs external clock for counter
		TMR13	Input	Inputs external reset to counter

## 26.2 Register Descriptions

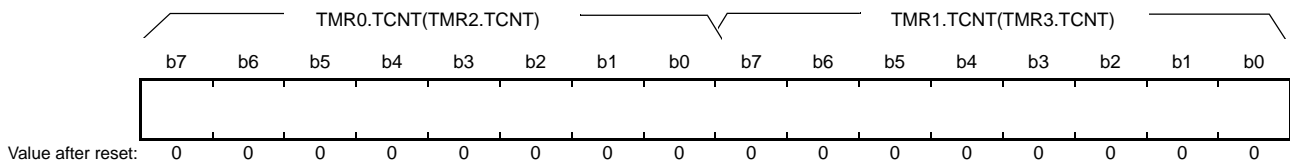
Table 26.4 lists register allocation for 16-bit access.

**Table 26.4 Register Allocation for 16-Bit Access**

Address	Upper 8 Bits	Lower 8 Bits
0008 8208h	TMR0.TCNT	TMR1.TCNT
0008 8204h	TMR0.TCORA	TMR1.TCORA
0008 8206h	TMR0.TCORB	TMR1.TCORB
0008 820Ah	TMR0.TCCR	TMR1.TCCR
0008 8218h	TMR2.TCNT	TMR3.TCNT
0008 8214h	TMR2.TCORA	TMR3.TCORA
0008 8216h	TMR2.TCORB	TMR3.TCORB
0008 821Ah	TMR2.TCCR	TMR3.TCCR

### 26.2.1 Timer Counter (TCNT)

Address(es): TMR0.TCNT 0008 8208h, TMR1.TCNT 0008 8209h  
TMR2.TCNT 0008 8218h, TMR3.TCNT 0008 8219h



TCNT is an 8-bit readable/writable up-counter.

TMR0.TCNT and TMR1.TCNT (TMR2.TCNT and TMR3.TCNT) comprise a single 16-bit counter so they can be accessed together by a word transfer instruction.

The TCCR.CSS[1:0] and CKS[2:0] bits are used to select a counter clock.

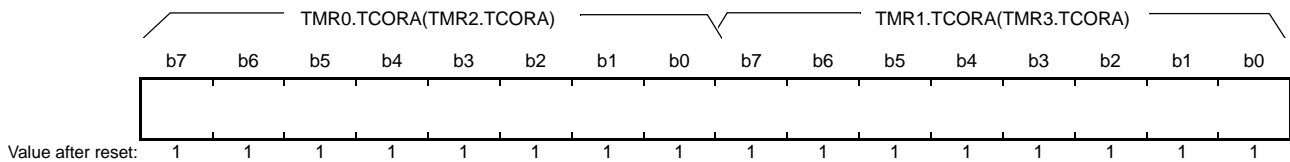
TCNT can be cleared by an external reset input signal, compare match A, or compare match B. Which compare match to be used for clearing is selected by the TCR.CCLR[1:0] bits.

When TCNT overflows (its value changes from FFh to 00h), an overflow interrupt (low-level pulse) is output provided the interrupt request is enabled by the TCR.OVIE bit.

For details on the corresponding interrupt vector number, see section 15, Interrupt controller (ICUb), and Table 26.6, TMR Interrupt Sources.

### 26.2.2 Time Constant Register A (TCORA)

Address(es): TMR0.TCORA 0008 8204h, TMR1.TCORA 0008 8205h  
TMR2.TCORA 0008 8214h, TMR3.TCORA 0008 8215h



TCORA is an 8-bit readable/writable register.

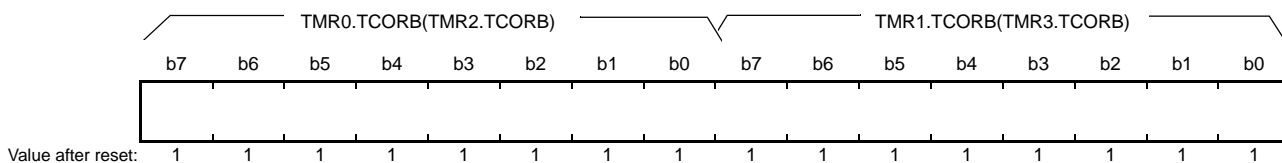
TMR0.TCORA and TMR1.TCORA (TMR2.TCORA and TMR3.TCORA) comprise a single 16-bit register so they can be accessed together by a word transfer instruction.

The value in TCORA is continually compared with the value in TCNT. When a match is detected, the corresponding compare match A is generated, and a compare match A interrupt (low-level pulse) is output provided the interrupt request is enabled by the TCR.CMIEA bit.

However, comparison is not performed during writing to TCORA. The timer output from the TMR pin can be freely controlled by this compare match A and the settings of the TCSR.OSA[1:0] bits.

### 26.2.3 Time Constant Register B (TCORB)

Address(es): TMR0.TCORB 0008 8206h, TMR1.TCORB 0008 8207h  
 TMR2.TCORB 0008 8216h, TMR3.TCORB 0008 8217h



TCORB is an 8-bit readable/writable register.

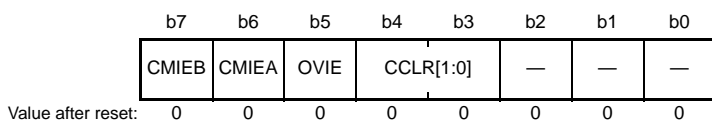
TMR0.TCORB and TMR1.TCORB (TMR2.TCORB and TMR3.TCORB) comprise a single 16-bit register so they can be accessed together by a word transfer instruction.

The value in TCORB is continually compared with the value in TCNT. When a match is detected, the corresponding compare match B is generated, and a compare match B interrupt (low-level pulse) is output provided the interrupt request is enabled by the TCR.CMIEB bit.

However, comparison is not performed during writing to TCORBn. The timer output from the TMO<sub>n</sub> pin can be freely controlled by this compare match B and the settings of the TCSRn.OSB[1:0] bits.

### 26.2.4 Timer Control Register (TCR)

Address(es): TMR0.TCR 0008 8200h, TMR1.TCR 0008 8201h  
 TMR2.TCR 0008 8210h, TMR3.TCR 0008 8211h



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
b4, b3	CCLR[1:0]	Counter Clear*1	b4 b3 0 0: Clearing is disabled 0 1: Cleared by compare match A 1 0: Cleared by compare match B 1 1: Cleared by the external reset input (Select edge or level by the TMRIS bit in TCCR.)	R/W
b5	OVIE	Timer Overflow Interrupt Enable	0: Overflow interrupt requests (OVIn) are disabled 1: Overflow interrupt requests (OVIn) are enabled	R/W
b6	CMIEA	Compare Match Interrupt Enable A	0: Compare match A interrupt requests (CMIA <sub>n</sub> ) are disabled 1: Compare match A interrupt requests (CMIA <sub>n</sub> ) are enabled	R/W
b7	CMIEB	Compare Match Interrupt Enable B	0: Compare match B interrupt requests (CMIB <sub>n</sub> ) are disabled 1: Compare match B interrupt requests (CMIB <sub>n</sub> ) are enabled	R/W

Note 1. To use an external reset, set the PORTn.PDR.Bn bit for the corresponding pin to 0 and the PORTn.PMR.Bn bit to 1. For details, see section 20, I/O Ports.

#### CMIEA Bit (Compare Match Interrupt Enable A)

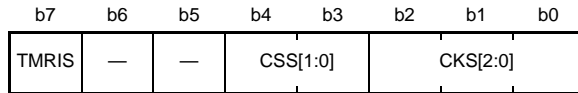
Selects whether compare match A interrupt requests (CMIA<sub>n</sub>) that are issued when the value of TCORA corresponds to that of TCNT are enabled or disabled.

#### CMIEB Bit (Compare Match Interrupt Enable B)

Selects whether compare match B interrupt requests (CMIB<sub>n</sub>) that are issued when the value of TCORB corresponds to that of TCNT are enabled or disabled.

## 26.2.5 Timer Counter Control Register (TCCR)

Address(es): TMR0.TCCR 0008 820Ah, TMR1.TCCR 0008 820Bh  
TMR2.TCCR 0008 821Ah, TMR3.TCCR 0008 821Bh



Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CKS[2:0]	Clock Select* <sup>1</sup>	See Table 26.5.	R/W
b4, b3	CSS[1:0]	Clock Source Select	See Table 26.5.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TMRIS	Timer Reset Detection Condition Select	0: Cleared at rising edge of the external reset 1: Cleared when the external reset is high	R/W

Note 1. To use an external reset, set the PORTn.PDR.Bn bit for the corresponding pin to 0 and the PORTn.PMR.Bn bit to 1. For details, see section 20, I/O Ports.

### CKS[2:0] Bits (Clock Select)

### CSS[1:0] Bits (Clock Source Select)

The CKS[2:0] and CSS[1:0] bits select a clock. For details, see Table 26.5.

### TMRIS Bit (Timer Reset Detection Condition Select)

This bit is enabled when the TCR.CCLR[1:0] bits are 11b (cleared by external reset input) and selects the condition for detecting external reset (level or edge).



Table 26.5 Clock Input to TCNT and Count Condition

Channel	TCCR Register					Description	
	CSS[1:0]		CKS[2:0]				
	b4	b3	b2	b1	b0		
TMR0 (TMR2)	0	0	—	0	0	Clock input prohibited	
					1	Uses external clock. Counts at rising edge*1.	
					1	Uses external clock. Counts at falling edge*1.	
					1	Uses external clock. Counts at both rising and falling edges*1.	
	0	1	0	0	0	Uses frequency dividing clock. Counts at PCLK.	
					1	Uses frequency dividing clock. Counts at PCLK/2.	
					1	Uses frequency dividing clock. Counts at PCLK/8.	
					1	Uses frequency dividing clock. Counts at PCLK/32.	
				1	0	0	Uses frequency dividing clock. Counts at PCLK/64.
						1	Uses frequency dividing clock. Counts at PCLK/1024.
						1	Uses frequency dividing clock. Counts at PCLK/8192.
						1	Clock input prohibited
	1	0	—	—	—	Setting prohibited	
	1	1	—	—	—	Counts at TMR1.TCNT (TMR3.TCNT) overflow signal*2.	
TMR1 (TMR3)	0	0	—	0	0	Clock input prohibited	
					1	Uses external clock. Counts at rising edge*1.	
					1	Uses external clock. Counts at falling edge*1.	
					1	Uses external clock. Counts at both rising and falling edges*1.	
	0	1	0	0	0	Uses frequency dividing clock. Counts at PCLK.	
					1	Uses frequency dividing clock. Counts at PCLK/2.	
					1	Uses frequency dividing clock. Counts at PCLK/8.	
					1	Uses frequency dividing clock. Counts at PCLK/32.	
				1	0	0	Uses frequency dividing clock. Counts at PCLK/64.
						1	Uses frequency dividing clock. Counts at PCLK/1024.
						1	Uses frequency dividing clock. Counts at PCLK/8192.
						1	Clock input prohibited
	1	0	—	—	—	Setting prohibited	
	1	1	—	—	—	Counts at TMR0.TCNT (TMR2.TCNT) compare match A*2.	

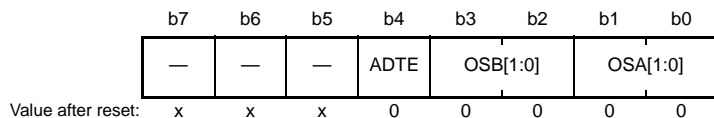
Note 1. To use an external reset, set the PORTn.PDR.Bn bit for the corresponding pin to 0 and the PORTn.PMR.Bn bit to 1. For details, see section 20, I/O Ports.

Note 2. If the clock input of TMR0 (TMR2) is the overflow signal of the TMR1.TCNT (TMR3.TCNT) counter and that of TMR1 (TMR3) is the compare match signal of the TMR0.TCNT (TMR2.TCNT) counter, no incrementing clock is generated. Do not use this setting.

## 26.2.6 Timer Control/Status Register (TCSR)

- TMR0.TCSR, TMR2.TCSR

Address(es): TMR0.TCSR 0008 8202h, TMR2.TCSR 0008 8212h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	OSA[1:0]	Output Select A*1	b1 b0 0 0: No change when compare match A occurs 0 1: Low is output when compare match A occurs 1 0: High is output when compare match A occurs 1 1: Output is inverted when compare match A occurs (toggle output)	R/W
b3, b2	OSB[1:0]	Output Select B*1	b3 b2 0 0: No change when compare match B occurs 0 1: Low is output when compare match B occurs 1 0: High is output when compare match B occurs 1 1: Output is inverted when compare match B occurs (toggle output)	R/W
b4	ADTE	A/D Trigger Enable*2	0: A/D converter start requests by compare match A are disabled 1: A/D converter start requests by compare match A are enabled	R/W
b7 to b5	—	Reserved	These bits are read as an undefined value. The write value should be 1.	R/W

Note 1. If the output enable signal for the corresponding TMO pin is negated while all of the OSA[1:0] and OSB[1:0] bits are 0, a request for high-impedance output is issued for the given I/O port pin. If any of the OSA[1:0] or OSB[1:0] bits is 1, the timer output remains at the low level until the first compare match after a reset.

Note 2. For the corresponding A/D converter channels, see section 38, 12-Bit A/D Converter (S12ADa), and section 39, 10-Bit A/D Converter (ADb).

### OSA[1:0] Bits (Output Select A)

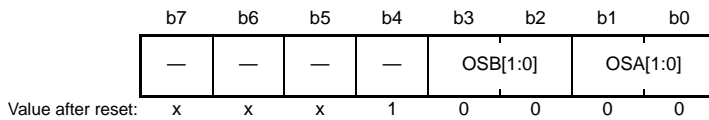
These bits select a method of TMO pin output when compare match A of TCORA and TCNT occurs.

### OSB[1:0] Bits (Output Select B)

These bits select a method of TMO pin output when compare match B of TCORB and TCNT occurs.

- TMR1.TCSR, TMR3.TCSR

Address(es): TMR1.TCSR 0008 8203h, TMR3.TCSR 0008 8213h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	OSA[1:0]	Output Select A* <sup>1</sup>	b1 b0 0 0: No change when compare match A occurs 0 1: Low is output when compare match A occurs 1 0: High is output when compare match A occurs 1 1: Output is inverted when compare match A occurs (toggle output)	R/W
b3, b2	OSB[1:0]	Output Select B* <sup>1</sup>	b3 b2 0 0: No change when compare match B occurs 0 1: Low is output when compare match B occurs 1 0: High is output when compare match B occurs 1 1: Output is inverted when compare match B occurs (toggle output)	R/W
b4	—	Reserved	This bit is read as 1. The write value should be 1.	R/W
b7 to b5	—	Reserved	These bits are read as an undefined value. The write value should be 1.	R/W

Note 1. If the output enable signal for the corresponding TMO pin is negated while all of the OSA[1:0] and OSB[1:0] bits are 0, a request for high-impedance output is issued for the given I/O port pin. If any of the OSA[1:0] or OSB[1:0] bits is 1, the timer output remains at the low level until the first compare match after a reset.

#### OSA[1:0] Bits (Output Select A)

These bits select a method of TMO pin output when compare match A of TCORA and TCNT occurs.

#### OSB[1:0] Bits (Output Select B)

These bits select a method of TMO pin output when compare match B of TCORB and TCNT occurs.

## 26.3 Operation

### 26.3.1 Pulse Output

Figure 26.3 shows an example of the 8-bit timer being used to generate a pulse output with a desired duty cycle.

1. Set the TCR.CCLR[1:0] bits to 01b (cleared by compare match A) so that TCNT is cleared at a compare match of TCORA.
2. Set the TCSR.OSA[1:0] bits to 10b (high output) and TCSR.OSB[1:0] bits to 01b (low output), causing the output to change to high at a compare match of TCORA and to low at a compare match of TCORB.

With these settings, the 8-bit timer provides pulses output at a cycle determined by TCORA with a pulse width determined by TCORB. No software intervention is required.

The timer output is low after the TCSR.OSA[1:0] or TCSR.OSB[1:0] bits are set until the first compare match occurs after a reset.

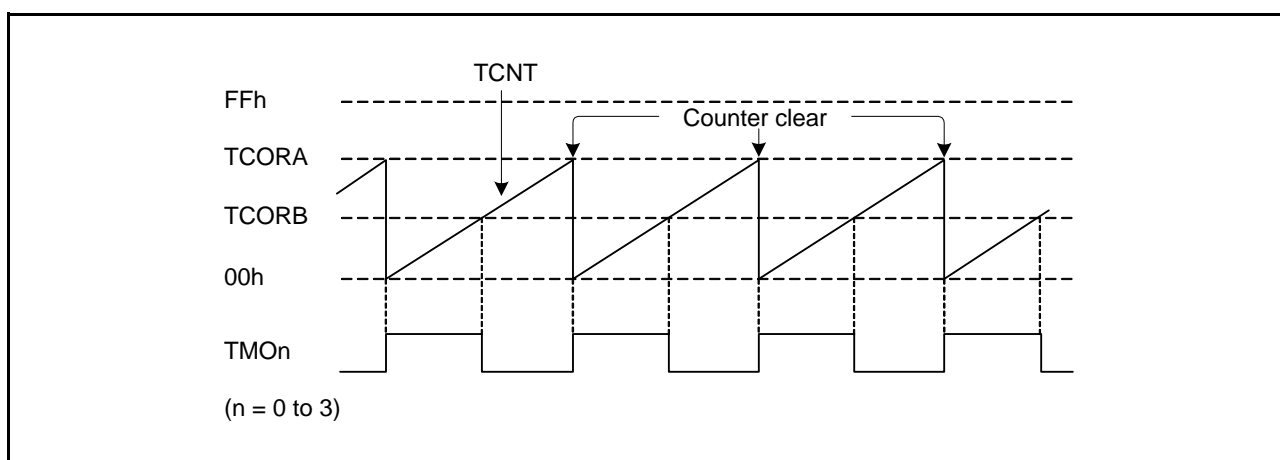


Figure 26.3 Example of Pulse Output

### 26.3.2 Reset Input

Figure 26.4 shows an example of the 8-bit timer being used to generate a pulse which is output after a desired delay time from a TMRIn input.

1. Set the TCR.CCLR[1:0] bits to 11b (cleared by external reset input) and set the TMRIS bit in TCCR to 1 (cleared when the external reset is high) so that TCNT is cleared at the high level input of the TMRIn signal.
2. Set the TCSR.OSA[1:0] bits to 10b (high output) and the TCSR.OSB[1:0] bits to 01b (low output), causing the output to change to high at a compare match of TCORA and to low at a compare match of TCORB.

With these settings, the 8-bit timer provides pulses output at a desired delay time from a TMRIn input determined by TCORA and with a pulse width determined by TCORB and TCORA.

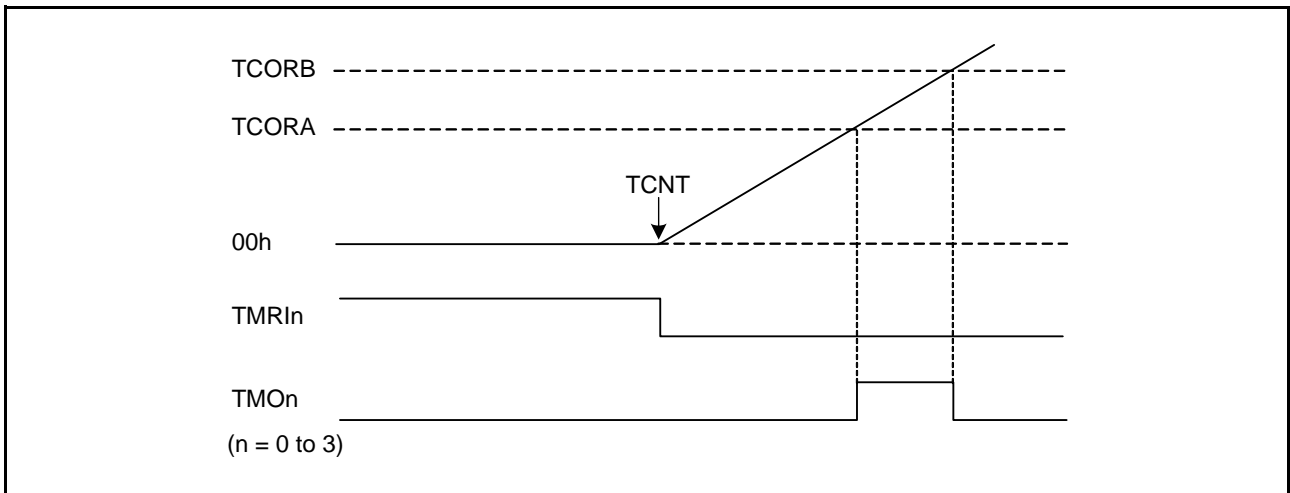


Figure 26.4 Example of Reset Input

## 26.4 Operation Timing

### 26.4.1 TCNT Count Timing

Figure 26.5 shows the count timing of TCNT for frequency dividing clock input. Figure 26.6 shows the count timing of TCNT for external clock input.

Note that the external clock pulse width must be at least 1.5 or more PCLK cycles for increment at a single edge, and at least 2.5 or more PCLK cycles for increment at both edges. The counter will not increment correctly if the pulse width is less than these values.

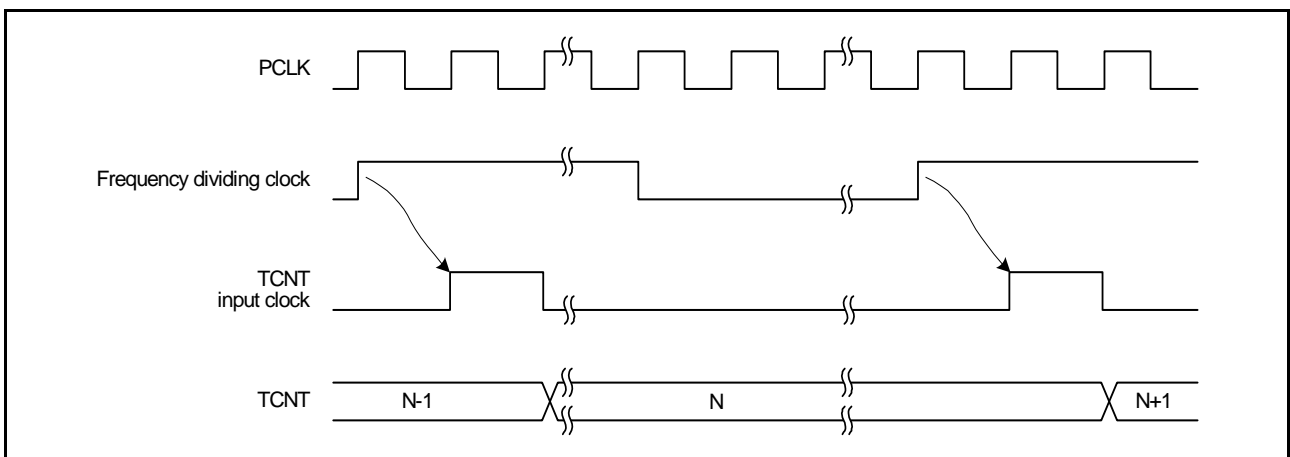


Figure 26.5 Count Timing for Frequency Dividing Clock Input

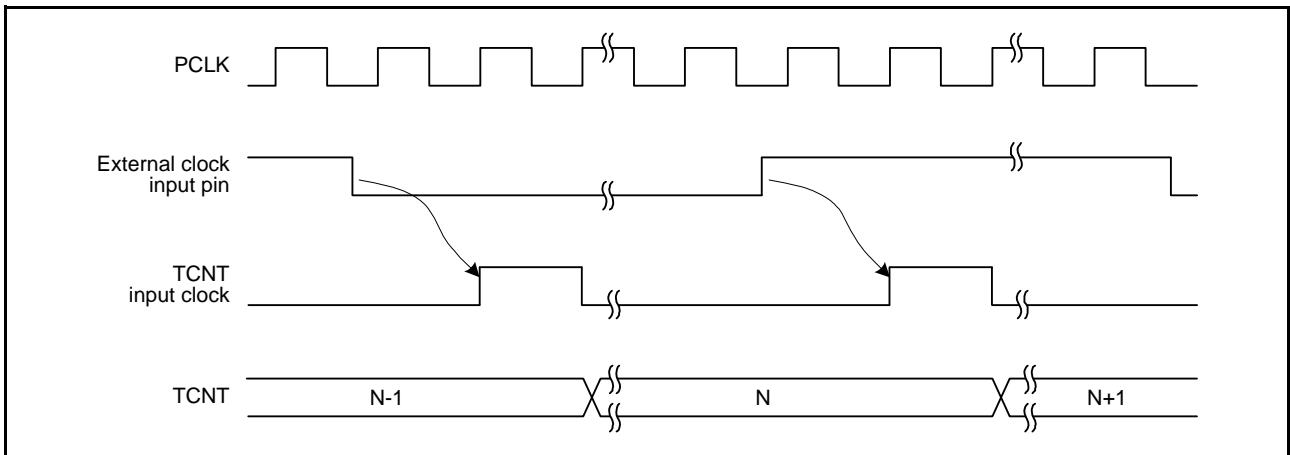


Figure 26.6 Count Timing for External Clock Input (at Both Edges)

### 26.4.2 Timing of Interrupt Flag Signal Output on a Compare Match

A compare match refers to a match between the value of the TCORA or TCORB register and the TCNT, and a compare match interrupt signal is output at this time if the interrupt request is enabled. The compare match is generated in the last cycle in which the values match (at the time at which the value counted by TCNT to produce the match is updated). Accordingly, after a match between TCNT and the TCORA or TCORB register is detected, the compare match is not actually generated until the next cycle of the input clock for the TCNT counter. Figure 26.7 shows the timing of output of the interrupt signal.

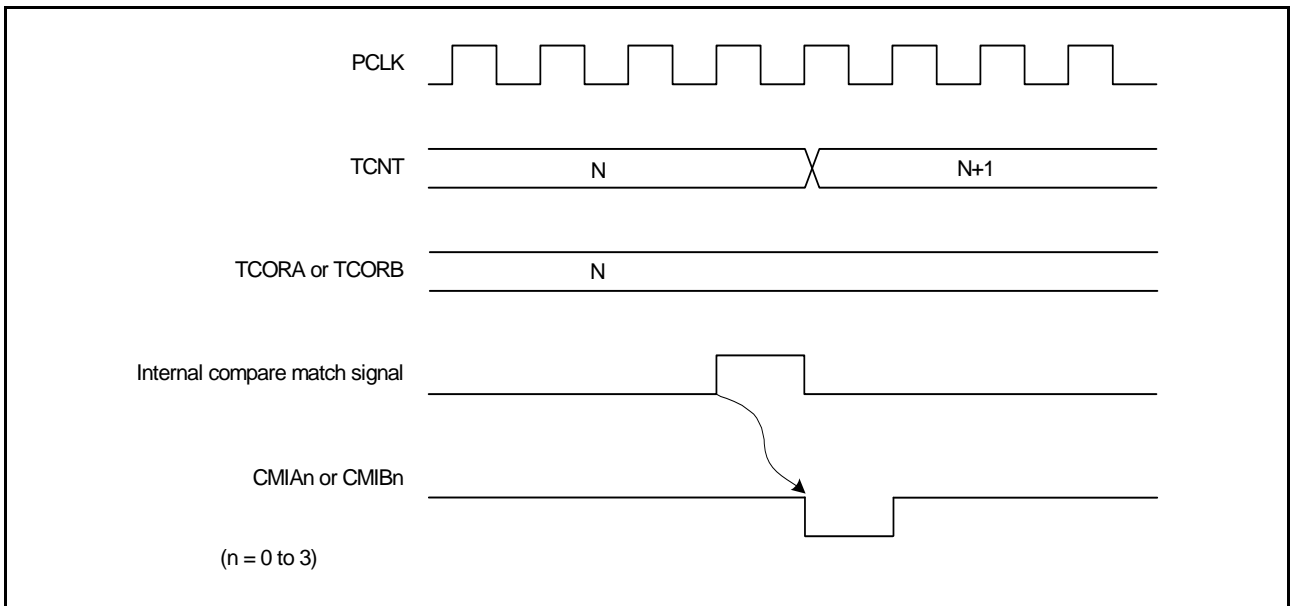


Figure 26.7 Timing of Interrupt Flag Signal Output on a Compare Match

### 26.4.3 Timing of Timer Output at Compare Match

When a compare match signal is generated, the output value specified by the TCSR.OSA[1:0] and OSB[1:0] bits is output on the timer output pin (TMO<sub>n</sub>).

Figure 26.8 shows the timing when the timer output is toggled by the compare match A signal.

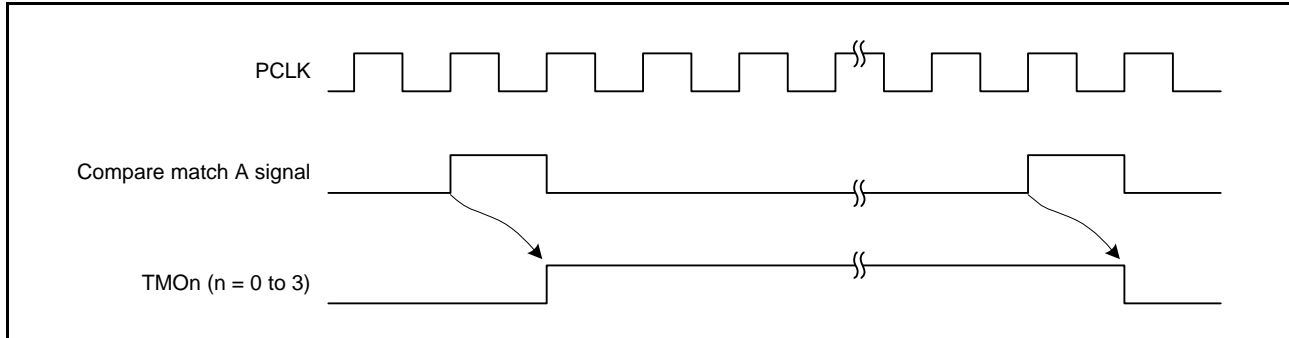


Figure 26.8 Timing of Timer Output at Compare Match A Signal

### 26.4.4 Timing of Counter Clear by Compare Match

TCNT is cleared when compare match A or B occurs, depending on the settings of the TCR.CCLR[1:0] bits.

Figure 26.9 shows the timing of this operation.

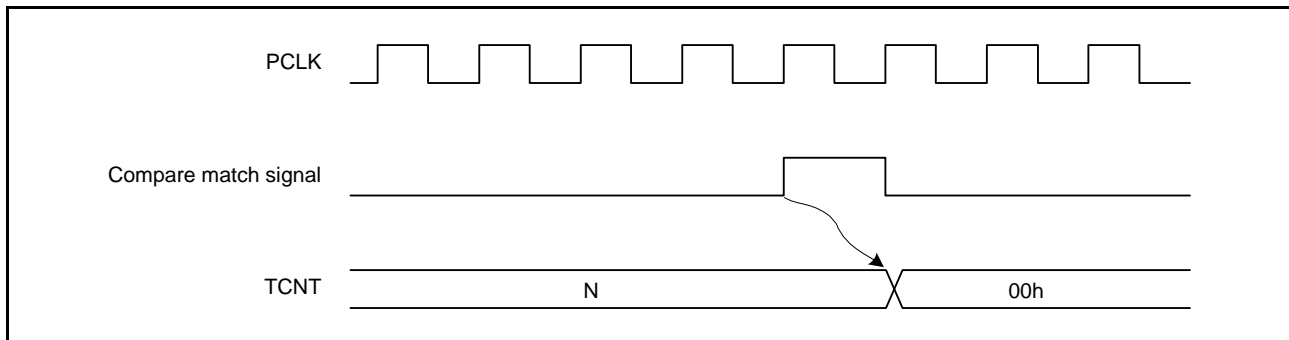


Figure 26.9 Timing of Counter Clear by Compare Match

### 26.4.5 Timing of the External Reset for TCNT

TCNT is cleared at the rising edge or high level of an external reset input, depending on the settings of the TCR.CCLR[1:0] bits. At least two or more PCLK cycles are required from an external reset input to clearing of TCNT. Figure 26.10 and Figure 26.11 show the timing of this operation.

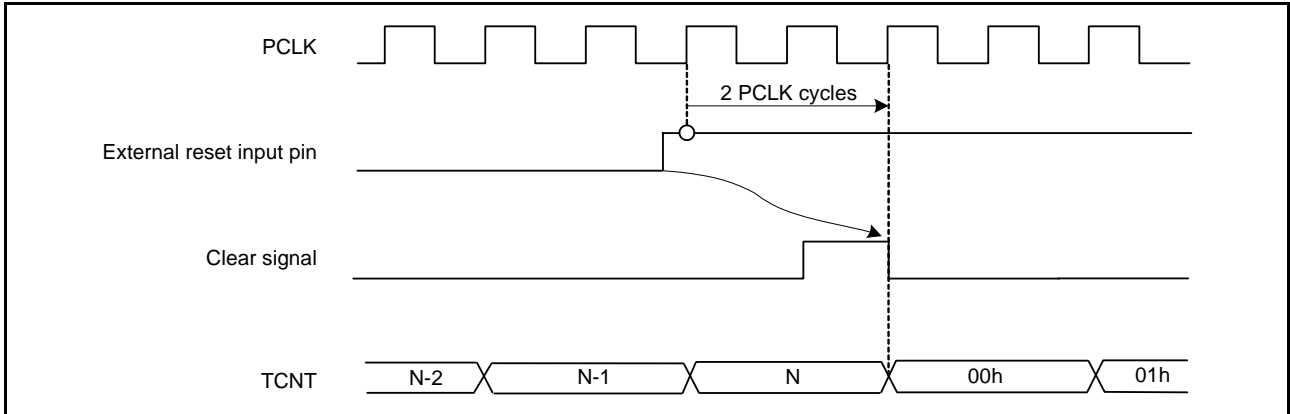


Figure 26.10 Timing of Clearance by External Reset (Rising Edge)

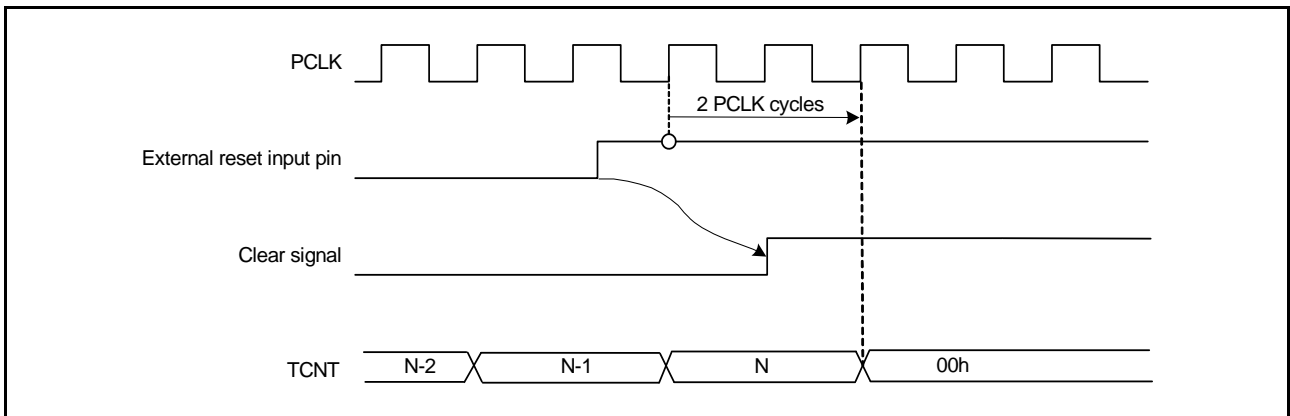


Figure 26.11 Timing of Clearance by External Reset (High Level)



### 26.4.6 Timing of Interrupt Signal Output on an Overflow

When TCNT overflows (changes from FFh to 00h), an overflow interrupt signal is output if this interrupt request is enabled.

Figure 26.12 shows the timing of output of the interrupt signal.

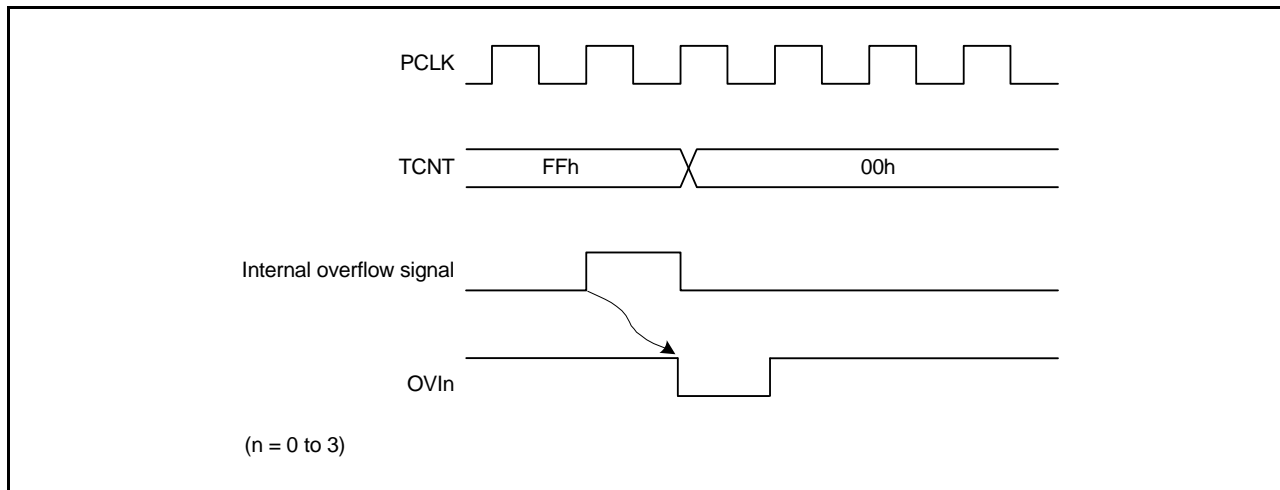


Figure 26.12 Timing of Interrupt Signal Output on an Overflow

## 26.5 Operation with Cascaded Connection

If the CSS[1:0] bits in either TMR0.TCCR or TMR1.TCCR are set to 11b, the TMR of the two channels are cascaded. With this configuration, a single 16-bit timer could be used (16-bit counter mode) or compare matches of TMR0 could be counted by TMR1 (compare match count mode).

Supplementary information: This section describes unit 0. The operation of unit 1 with cascaded connection is the same as unit 0.

### 26.5.1 16-Bit Count Mode

When the TMR0.TCCR.CSS[1:0] bits are set to 11b, the timer functions as a single 16-bit timer with TMR0 occupying the upper 8 bits and TMR1 occupying the lower 8 bits.

#### (1) Counter Clear Specification

- The settings of the TMR0.TCR.CCLR[1:0] bits become effective for the 16-bit counter. If the TMR0.TCR.CCLR[1:0] bits have been set for counter clear at compare match, the 16-bit counter (TMR0.TCNT and TMR1.TCNT together) is cleared when a 16-bit compare match event occurs. The 16-bit counter (TMR0.TCNT and TMR1.TCNT together) is cleared even if counter clear by the TMRIO pin has been set.
- The settings of the TMR1.TCR.CCLR[1:0] bits are ignored.

#### (2) Pin Output

- Control of output from the TMO0 pin by the TMR0.TCSR.OSA[1:0] and OSB[1:0] bits is in accordance with the 16-bit compare match conditions.
- Control of output from the TMO1 pin by the TMR1.TCSR.OSA[1:0] and OSB[1:0] bits is in accordance with the lower 8-bit compare match conditions.

### 26.5.2 Compare Match Count Mode

When the TMR1.TCCR.CSS[1:0] bits are set to 11b, TMR1.TCNT counts the number of occurrences of compare match A for TMR0. TMR0 and TMR1 are controlled independently. Conditions such as generation of interrupts, output from the TMO<sub>n</sub> (n = 0, 1) pin, and counter clear are in accordance with the settings for each channel.

## 26.6 Interrupt Sources

### 26.6.1 Interrupt Sources and DTC Activation

There are three interrupt sources for TMRn: CMIA<sub>n</sub>, CMIB<sub>n</sub>, and OVI<sub>n</sub>. Their interrupt sources and priorities are listed in Table 26.6.

It is also possible to activate the DTC by means of CMIA<sub>n</sub> and CMIB<sub>n</sub> interrupts. The DMAC cannot be activated by the interrupt sources for TMRn.

**Table 26.6 TMR Interrupt Sources**

Name	Interrupt Sources	DTC Activation	Priority
CMIA0	TMR0.TCORA compare match	Possible	High
CMIB0	TMR0.TCORB compare match	Possible	↑ Low
OVI0	TMR0.TCNT overflow	Not possible	
CMIA1	TMR1.TCORA compare match	Possible	
CMIB1	TMR1.TCORB compare match	Possible	
OVI1	TMR1.TCNT overflow	Not possible	
CMIA2	TMR2.TCORA compare match	Possible	
CMIB2	TMR2.TCORB compare match	Possible	
OVI2	TMR2.TCNT overflow	Not possible	
CMIA3	TMR3.TCORA compare match	Possible	
CMIB3	TMR3.TCORB compare match	Possible	
OVI3	TMR3.TCNT overflow	Not possible	

### 26.6.2 A/D Converter Activation

The A/D converter\*1 can be activated by a compare match A for TMR0 or TMR2.

If the TMRn.TCSR.ADTE bit is set to 1 (A/D converter start requests by compare match A are enabled), a request to start A/D conversion is sent to the A/D converter by the occurrence of a compare match A. If the 8-bit timer conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started.

Note 1. For the corresponding A/D converter units, see section 38, 12-Bit A/D Converter (S12ADa), and section 39, 10-Bit A/D Converter (ADb).

**Table 26.7 A/D Converter Activation**

Module Symbol	Unit	Target	The Source of A/D Activation	A/D Conversion Start Request
S12AD	0	TMR0.TCORA, TMR0.TCNT	Compare match	TMTRG0AN_0
	1	TMR2.TCORA, TMR2.TCNT		TMTRG0AN_1
AD	0	TMR0.TCORA, TMR0.TCNT		TMTRG0AN_0

## 26.7 Usage Notes

### 26.7.1 Module-Stop State Setting

Operation of the TMR can be disabled or enabled by using the module-stop control registers. The initial setting is for halting of TMR operation. Register access becomes possible after release from the module-stop state. For details, see section 11, Low Power Consumption.

### 26.7.2 Notes on Setting Cycle

If the compare match is selected for counter clear, TCNT is cleared at the last PCLK in the cycle in which the value of TCNT matches with that of TCORA or TCORB. TCNT updates the counter value at this last state. Therefore, the counter frequency is obtained by the following formula (f: Counter frequency, PCLK: Operating frequency, N: TCORA and TCORB register setting value).

$$f = \text{PCLK} / (N + 1)$$

### 26.7.3 Conflict between TCNT Write and Counter Clear

If a counter clear signal is generated concurrently with CPU write to TCNT, the clear takes priority and the write is not performed as shown in Figure 26.13.

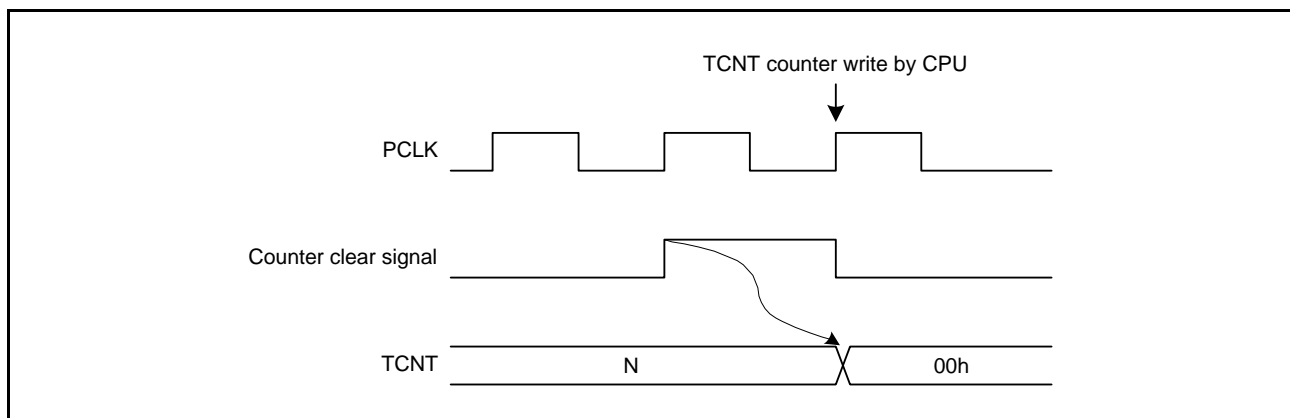


Figure 26.13 Conflict between TCNT Write and Counter Clear

### 26.7.4 Conflict between TCNT Write and Increment

Even if a counting-up signal is generated concurrently with CPU write to TCNT, the counting-up is not performed and the write takes priority as shown in Figure 26.14.

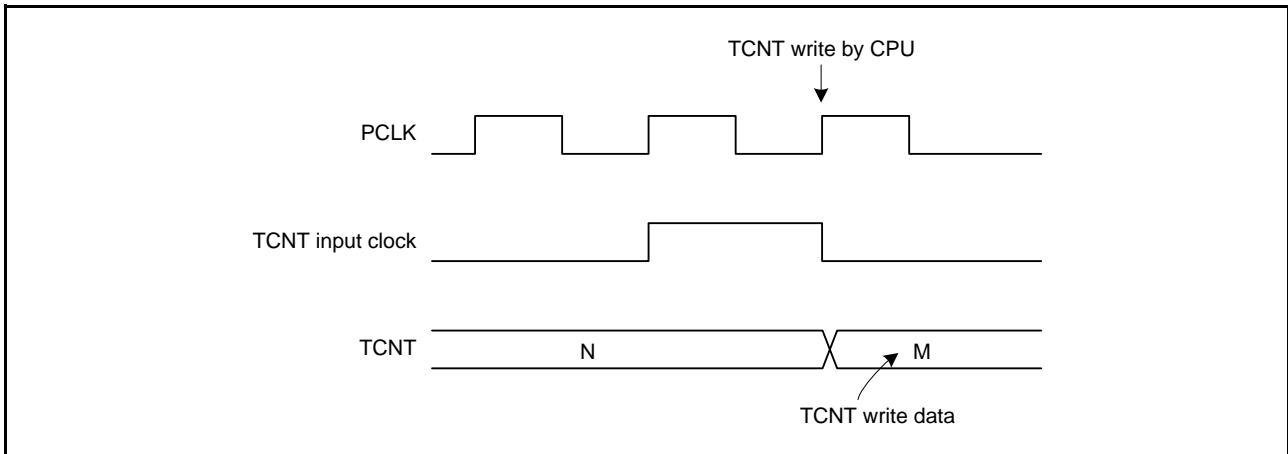


Figure 26.14 Conflict between TCNT Write and Increment

### 26.7.5 Conflict between TCORA or TCORB Write and Compare Match

Even if a compare match signal is generated simultaneously with CPU write to TCORA or TCORB, the write takes priority and the compare match signal is not generated.

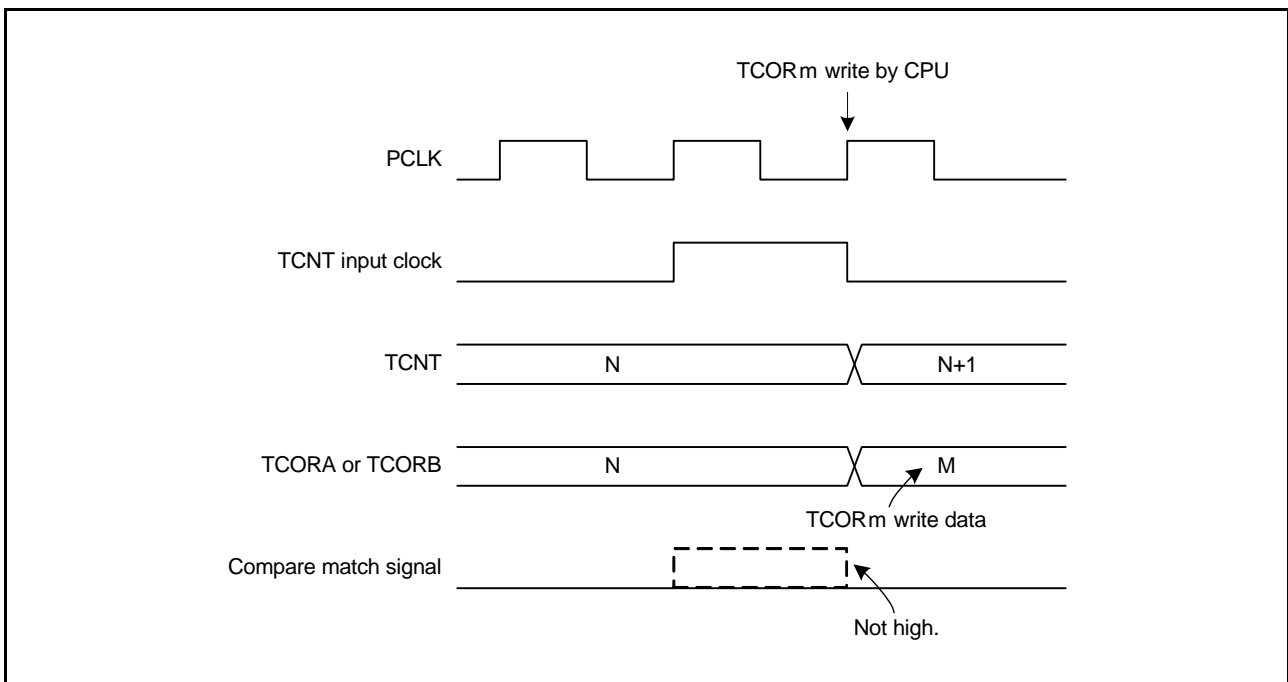


Figure 26.15 Conflict between TCORA or TCORB Write and Compare Match

### 26.7.6 Conflict between Compare Matches A and B

If compare match events A and B occur at the same time, the 8-bit timer operates in accordance with the priorities for the output statuses high for compare match A and compare match B, as listed in Table 26.8.

**Table 26.8 Timer Output Priorities**

Output Setting	Priority
Toggle output	High
High output	↑
Low output	
No change	Low

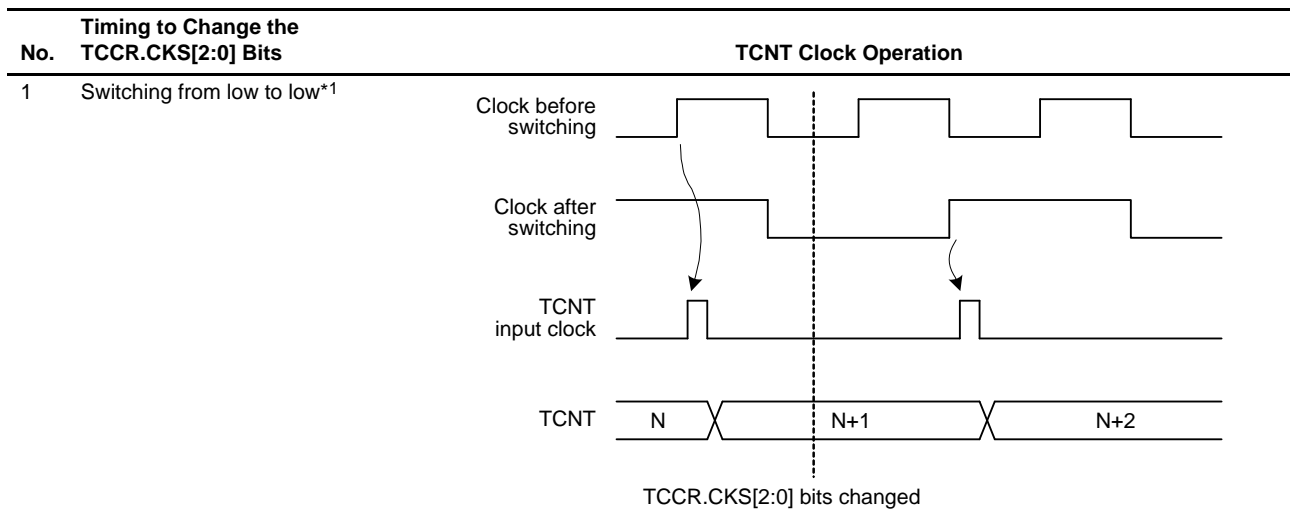
### 26.7.7 Switching of Frequency Dividing Clocks and TCNT Operation

TCNT may be incremented erroneously depending on when the frequency dividing clock is switched. Table 26.9 lists the relationship between the timing at which the frequency dividing clock is switched (by writing to the TCCR.CKS[2:0] bits) and the operation of TCNT.

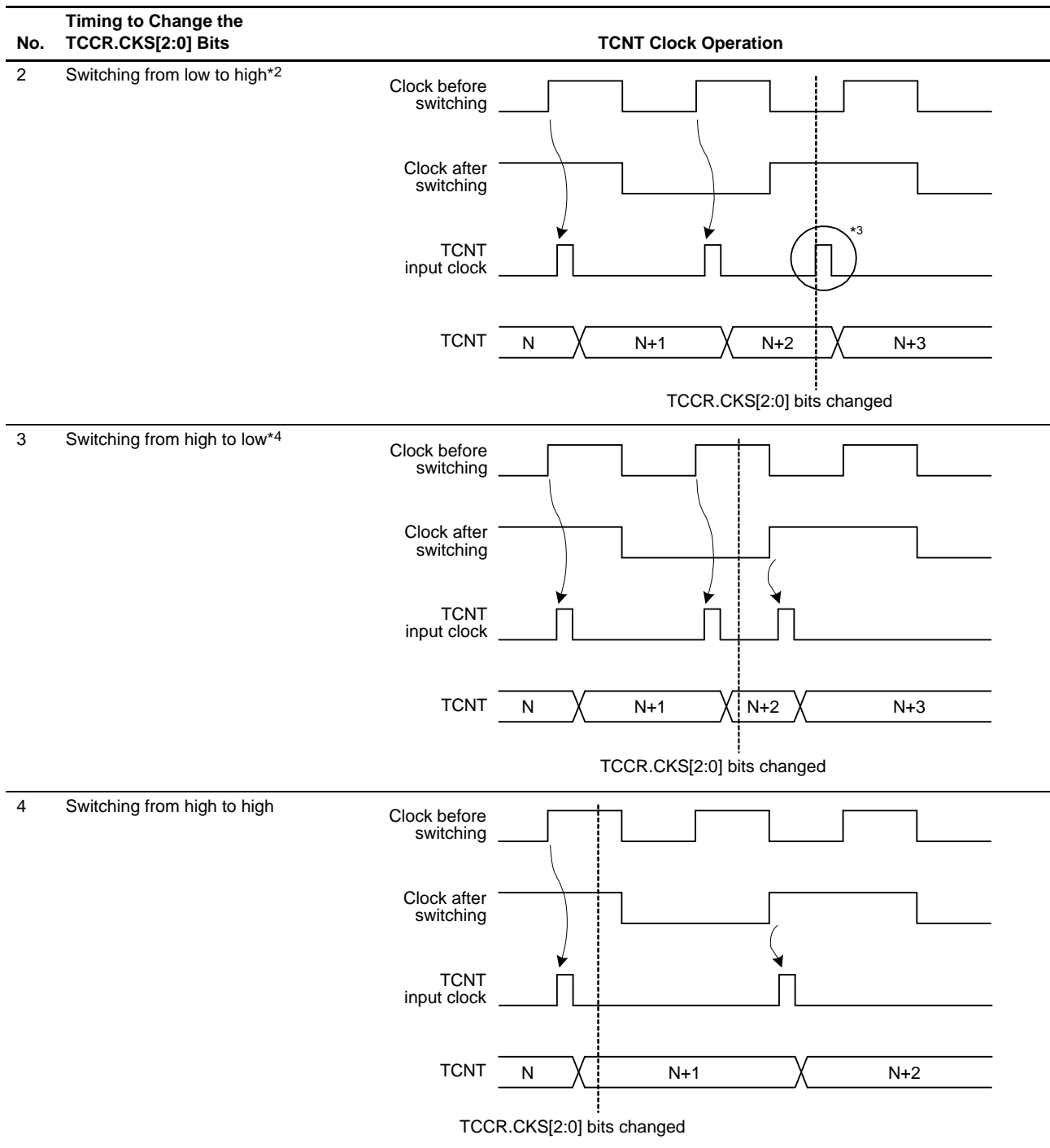
When TCNT clock is generated from an frequency dividing clock, the rising edge of the frequency dividing clock pulse are always monitored. If the signal levels of the clocks before and after switching change from low to high as shown in No. 2 in Table 26.9, the change is considered as an edge. Therefore, a TCNT clock pulse is generated and TCNT is incremented.

The erroneous increment of TCNT can also happen when switching between external and frequency dividing clocks.

**Table 26.9 Switching of Frequency Dividing Clocks and TCNT Operation (1/2)**



**Table 26.9 Switching of Frequency Dividing Clocks and TCNT Operation (2/2)**



Note 1. Includes switching from low to stop, and from stop to low.

Note 2. Includes switching from stop to high.

Note 3. Generated because the change of the signal levels is considered as an edge; TCNT is incremented.

Note 4. Includes switching from high to stop.

### 26.7.8 Clock Source Setting with Cascaded Connection

If 16-bit counter mode and compare match count mode are specified at the same time, input clocks for TMR0.TCNT and TMR1.TCNT (TMR2.TCNT and TMR3.TCNT) are not generated, and the counter stops. Do not specify 16-bit counter mode and compare match count mode simultaneously.

### 26.7.9 Continuous Output of Compare Match Interrupt Signal

When TCORA or TCORB is set to 00h, PCLK/1 is set as the frequency dividing clock, and compare match is set as the counter clear source, the TCNT counter remains 00h and is not updated, and a compare match interrupt signal is output continuously to form a flat signal level.

At this time, the interrupt controller cannot detect the second and subsequent interrupts.

Figure 26.16 shows operation timing when the compare match interrupt signal is continuously output.

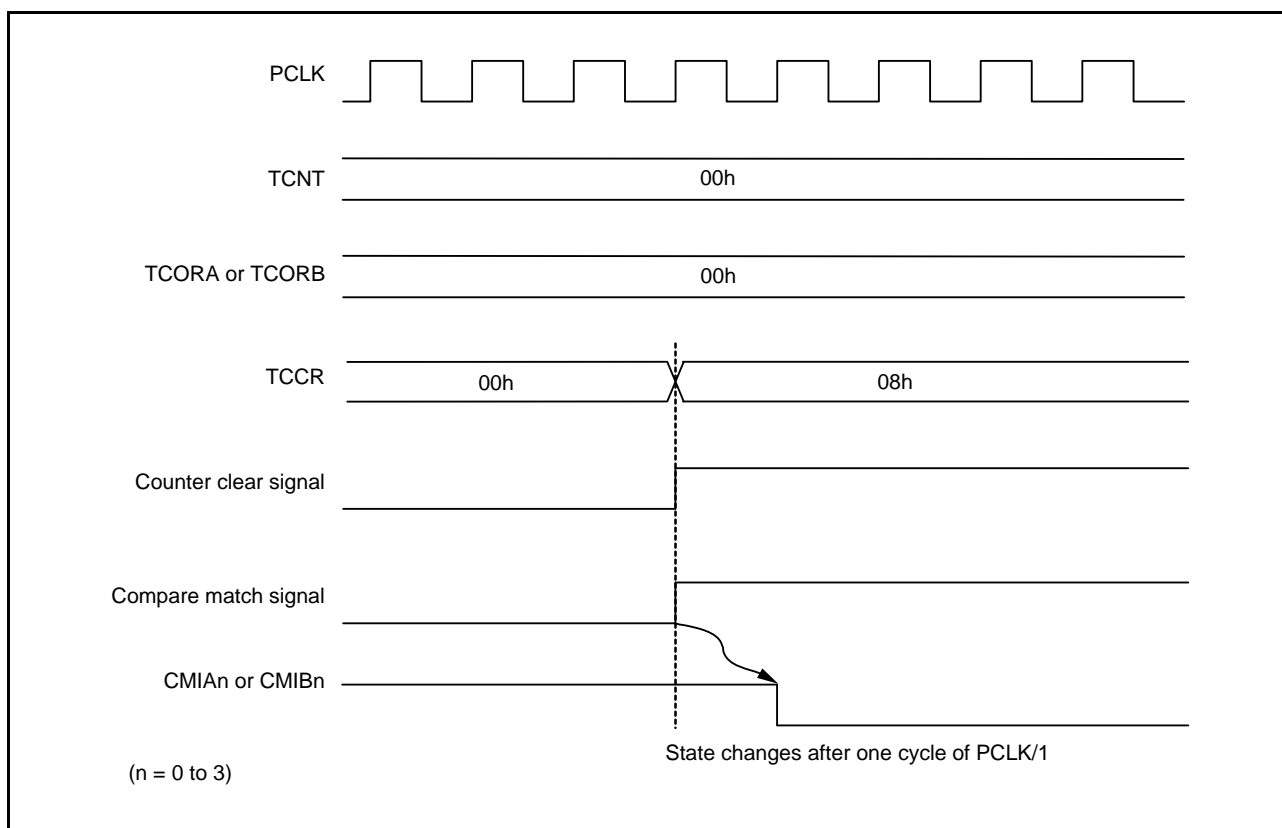


Figure 26.16 Continuous Output of Compare Match Interrupt Signal

## 27. Compare Match Timer (CMT)

The RX630 Group has two on-chip compare match timer (CMT) units (unit 0 and unit 1) each consisting of a two-channel 16-bit timer (i.e., a total of four channels). The CMT has a 16-bit counter, and can generate interrupts at set intervals.

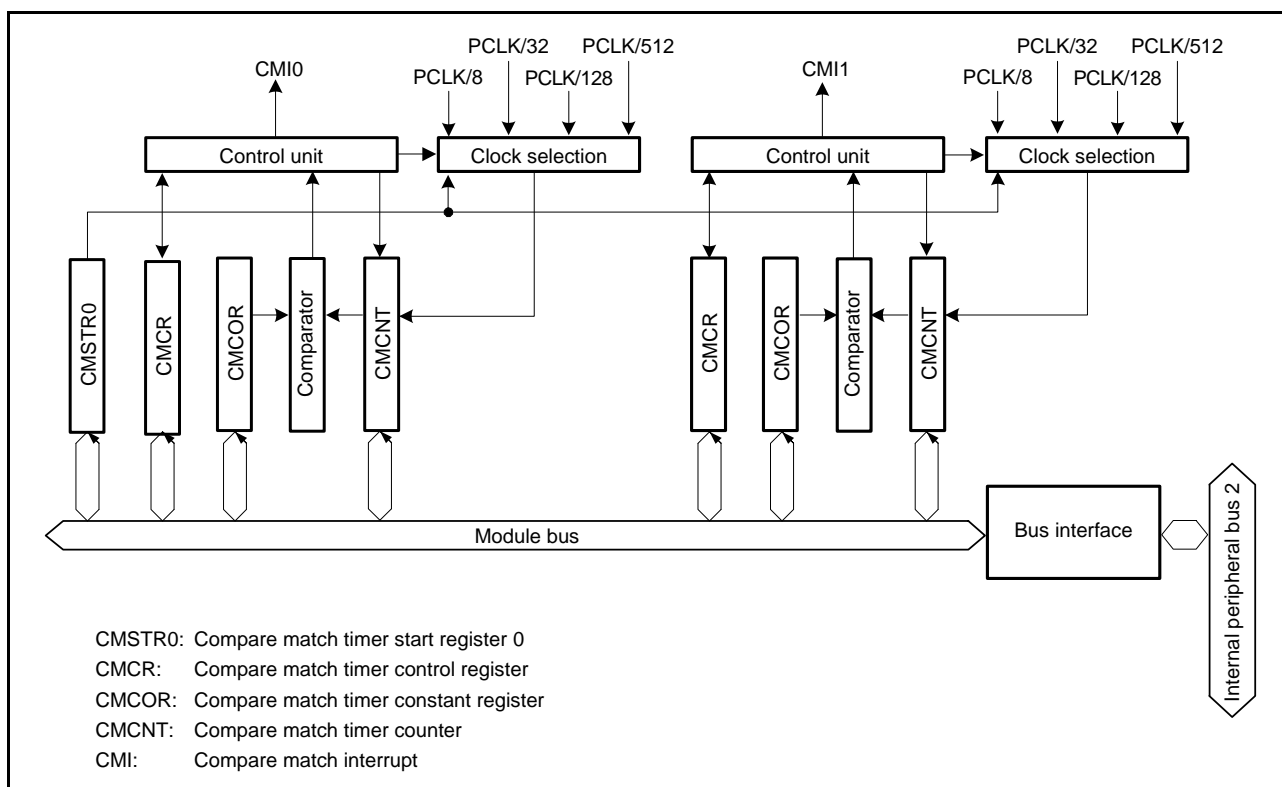
### 27.1 Overview

Table 27.1 lists the specifications for the CMT.

Figure 27.1 shows a block diagram of the CMT (unit 0). A two-channel CMT constitutes a unit. Unit 0 and unit 1 are the same in terms of specifications.

**Table 27.1 Specifications of CMT**

Item	Description
Count clock	<ul style="list-style-type: none"> <li>Four frequency dividing clocks</li> <li>One clock from PCLK/8, PCLK/32, PCLK/128, and PCLK/512 can be selected individually for each channel.</li> </ul>
Interrupt	A compare match interrupt can be requested individually for each channel.
Low power consumption facilities	Each unit can be placed in a module-stop state.



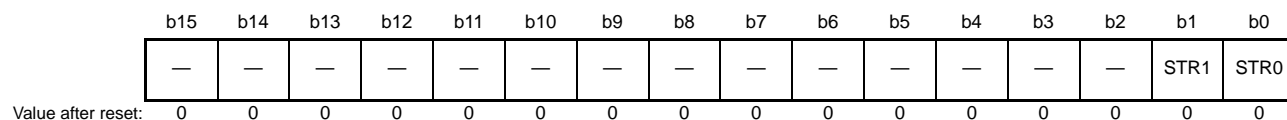
**Figure 27.1 Block Diagram of CMT (Unit 0)**



## 27.2 Register Descriptions

### 27.2.1 Compare Match Timer Start Register 0 (CMSTR0)

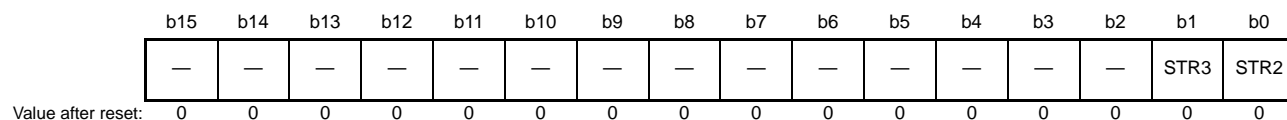
Address(es): 0008 8000h



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

### 27.2.2 Compare Match Timer Start Register 1 (CMSTR1)

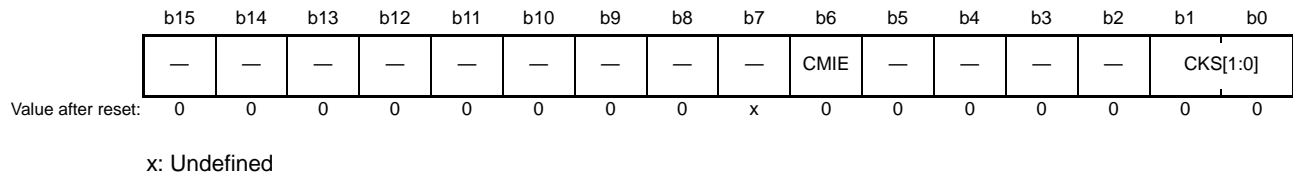
Address(es): 0008 8010h



Bit	Symbol	Bit Name	Description	R/W
b0	STR2	Count Start 2	0: CMT2.CMCNT count is stopped 1: CMT2.CMCNT count is started	R/W
b1	STR3	Count Start 3	0: CMT3.CMCNT count is stopped 1: CMT3.CMCNT count is started	R/W
b15 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

### 27.2.3 Compare Match Timer Control Register (CMCR)

Address(es): CMT0.CMCR 0008 8002h, CMT1.CMCR 0008 8008h,  
CMT2.CMCR 0008 8012h, CMT3.CMCR 0008 8018h



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 clock to be input to CMCNT from four frequency dividing clocks obtained by dividing the peripheral clock (PCLK).

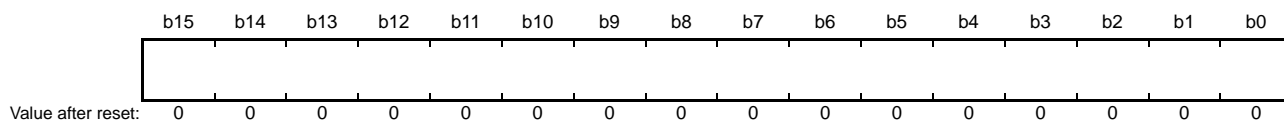
When the STR<sub>n</sub> (n = 0 to 3) bit in CMSTR<sub>m</sub> (m = 0 or 1) is set to 1, CMCNT starts counting up on the clock selected with bits CKS[1:0].

#### CMIE Bit (Compare Match Interrupt Enable)

The CMIE bit enables or disables compare match interrupt (CMIn) (n = 0 to 3) generation when CMCNT and CMCOR values match.

### 27.2.4 Compare Match Timer Counter (CMCNT)

Address(es): CMT0.CMCNT 0008 8004h, CMT1.CMCNT 0008 800Ah,  
CMT2.CMCNT 0008 8014h, CMT3.CMCNT 0008 801Ah



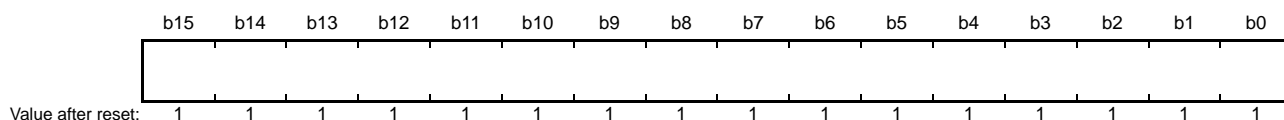
CMCNT is a readable/writable up-counter to generate interrupt requests.

When an frequency dividing clock is selected by bits CKS[1:0] in CMCR and the STRn (n = 0 to 3) bit in CMSTRm (m = 0 or 1) is set to 1, CMCNT starts counting up using the selected clock.

When the value in CMCNT and the value in CMCOR match, CMCNT is cleared to 0000h. At the same time, a compare match interrupt (CMIn) (n = 0 to 3) is generated.

### 27.2.5 Compare Match Timer Constant Register (CMCOR)

Address(es): CMT0.CMCOR 0008 8006h, CMT1.CMCOR 0008 800Ch,  
CMT2.CMCOR 0008 8016h, CMT3.CMCOR 0008 801Ch



CMCOR is a readable/writable register to set a compare match cycle with CMCNT.

### 27.3 Operation

#### 27.3.1 Periodic Count Operation

When an frequency dividing clock is selected by bits CKS[1:0] in CMCR and the STR<sub>n</sub> (n = 0 to 3) bit in CMSTR<sub>m</sub> (m = 0 or 1) is set to 1, CMCNT starts counting up using the selected clock.

When the value in CMCNT and the value in CMCOR match, CMCNT is cleared to 0000h. At the same time, a compare match interrupt (CMI<sub>n</sub>) (n = 0 to 3) is generated. CMCNT then starts counting up again from 0000h. Figure 27.2 shows the operation of the CMCNT counter.

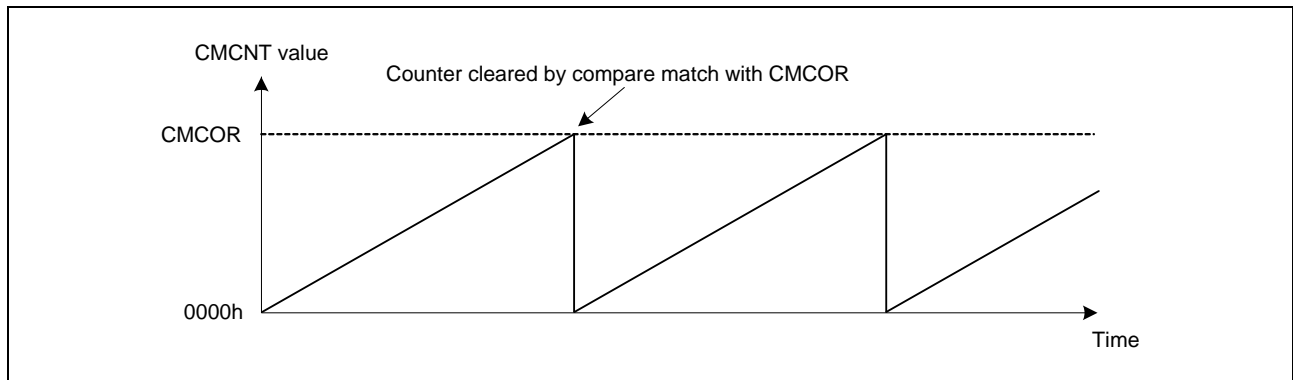


Figure 27.2 CMCNT Counter Operation

#### 27.3.2 CMCNT Count Timing

As the count clock to be input to CMCNT, one of four frequency dividing clocks (PCLK/8, PCLK/32, PCLK/128, and PCLK/512) obtained by dividing the peripheral clock (PCLK) can be selected with the CKS[1:0] bits in CMCR. Figure 27.3 shows the timing of CMCNT.

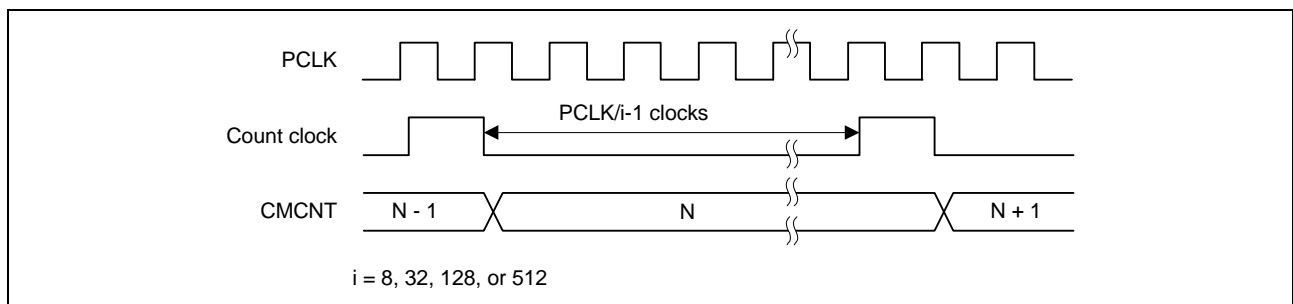


Figure 27.3 CMCNT Count Timing

## 27.4 Interrupts

### 27.4.1 Interrupt Sources

The CMT has channels and each of them to which a different vector address is allocated has a compare match interrupt (CMI<sub>n</sub>) (n = 0 to 3). When a compare match interrupt occurs, the corresponding interrupt request is output.

When the interrupt is used to activate a CPU interrupt, the priority of channels can be changed by the interrupt controller settings. For details, see section 15, Interrupt controller (ICUb).

**Table 27.2 CMT Interrupt Sources**

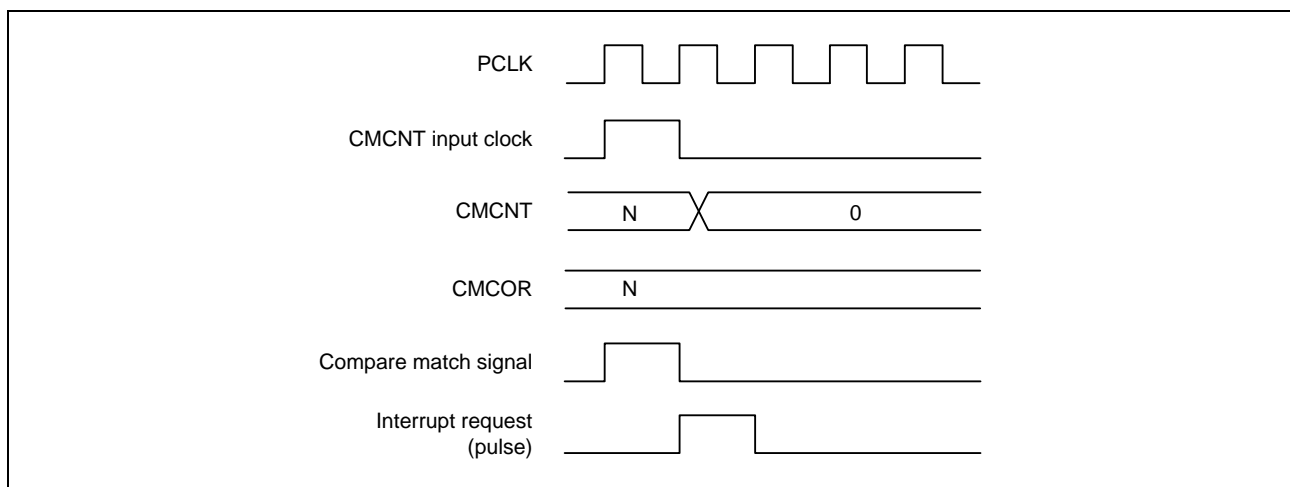
Name	Interrupt Sources	DTC Activation	DMAC Activation
CMI0	Compare match between CMT0.CMCNT and CMT0.CMCOR	Possible	Possible
CMI1	Compare match between CMT1.CMCNT and CMT1.CMCOR	Possible	Possible
CMI2	Compare match between CMT2.CMCNT and CMT2.CMCOR	Possible	Possible
CMI3	Compare match between CMT3.CMCNT and CMT3.CMCOR	Possible	Possible

### 27.4.2 Timing of Compare Match Interrupt Generation

When CMCNT and CMCOR match, a compare match interrupt (CMI<sub>n</sub>) (n = 0 to 3) 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 CMCOR and CMCNT, the compare match signal is not generated until the next CMCNT input clock.

Figure 27.4 shows the timing for the setting of a compare match interrupt.



**Figure 27.4 Timing for the Setting of a Compare Match Interrupt**

## 27.5 Usage Notes

### 27.5.1 Setting the Module-Stop Function

The CMT can be enabled or disabled using the module-stop control register. The CMT is disabled by default. The registers can be accessed by canceling the module-stop state. For details, see section 11, Low Power Consumption.

### 27.5.2 Conflict between Write and Compare-Match Processes of CMCNT

When the compare match signal is generated while writing to CMCNT, clearing CMCNT has priority over writing to it. In this case, CMCNT is not written to. Figure 27.5 shows the timing to clear the CMCNT counter.

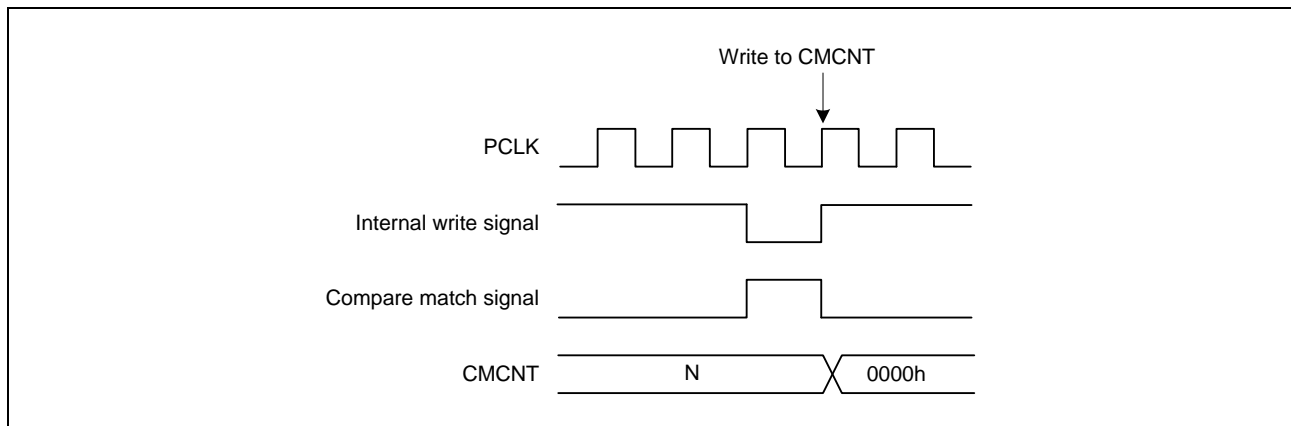


Figure 27.5 Conflict between Write and Compare Match Processes of CMCNT

### 27.5.3 Conflict between Write and Count-Up Processes of CMCNT

Even when the count-up occurs while writing to CMCNT, the writing has priority over the count-up. In this case, the count-up is not performed. Figure 27.6 shows the timing to write the CMCNT counter.

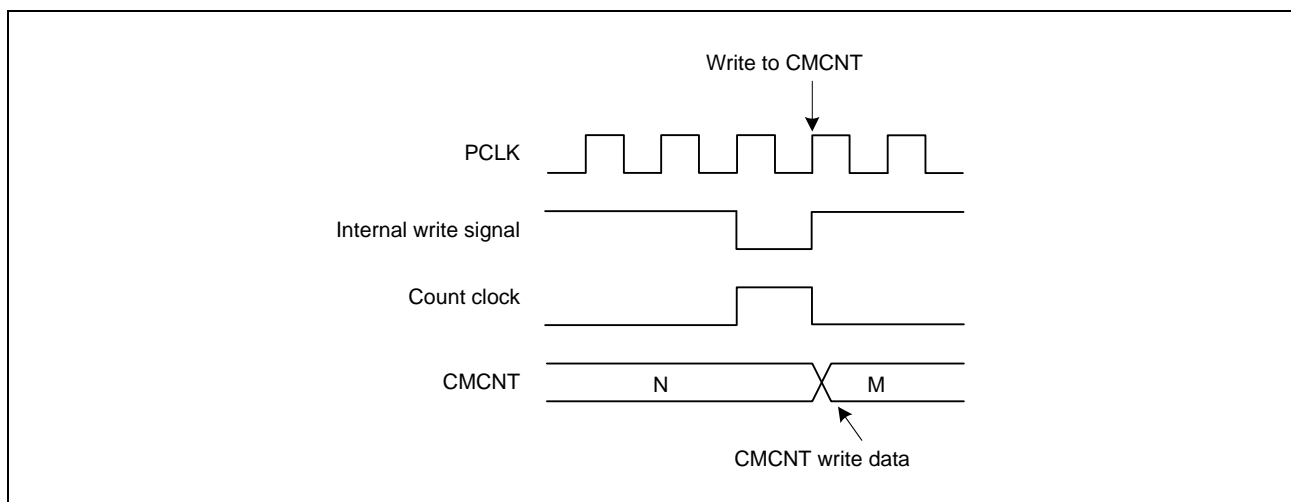


Figure 27.6 Conflict between Write and Count-Up Processes of CMCNT

## 28. Realtime Clock (RTCa)

### 28.1 Overview

The RTC is capable of counting 100 years from year 00 to year 99. If we place 20 in the hundreds and thousands positions, counting is automatically adjusted for leap years.

The source to drive counting of the time counters is selectable as the sub-clock or a main clock.

With the source for counting pre-scaled to produce a 128-Hz clock signal as the base clock, cycles of this signal are counted in year, month, date, day of the week, a.m. or p.m. (in 12-hour mode), hour, minute, second, and 1/128 second units.

Table 28.1 lists the specifications of the RTC, Figure 28.1 shows a block diagram of the RTC, and Table 28.2 shows the pin configuration of the RTC.

**Table 28.1 Specifications of RTC**

Item	Description
Count source*1	Sub-clock (XCIN) or main clock (EXTAL)
Clock and calendar functions	<ul style="list-style-type: none"> <li>Year, month, date, day of the week, hours, minutes, and seconds are counted and represented in BCD</li> <li>Indicates the state of 1, 2, 4, 8, 16, 32, or 64 Hz in binary</li> <li>Selection of 12- or 24-hour mode</li> <li>Start/stop function</li> <li>30-second adjustment (30 seconds or less are rounded down to 00 second, and 30 seconds or more are rounded up to one minute)</li> <li>Automatic leap year adjustment</li> <li>Output a 1-Hz clock</li> <li>Time error adjustment function</li> </ul>
Interrupts	<ul style="list-style-type: none"> <li>Alarm interrupt (ALM) Year, month, date, day of the week, hours, minutes, and seconds can be selected as conditions for the alarm interrupt</li> <li>Periodic interrupt (PRD) 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, or 1/256 second can be selected as an interrupt period.</li> <li>Carry interrupt (CUP) Indicates occurrence of a carry to the seconds counter or a carry to the 64-Hz counter during reading of the 64-Hz counter</li> <li>Recovery from software standby mode or deep software standby mode can be performed by an alarm interrupt or periodic interrupt</li> </ul>
Time-capture function	<ul style="list-style-type: none"> <li>Times when any of three event signals are input can be captured The month, date, hour, minute, and second are captured for each event</li> </ul>

Note 1. Set the count source so that the peripheral module clock frequency is greater than the count source clock frequency.

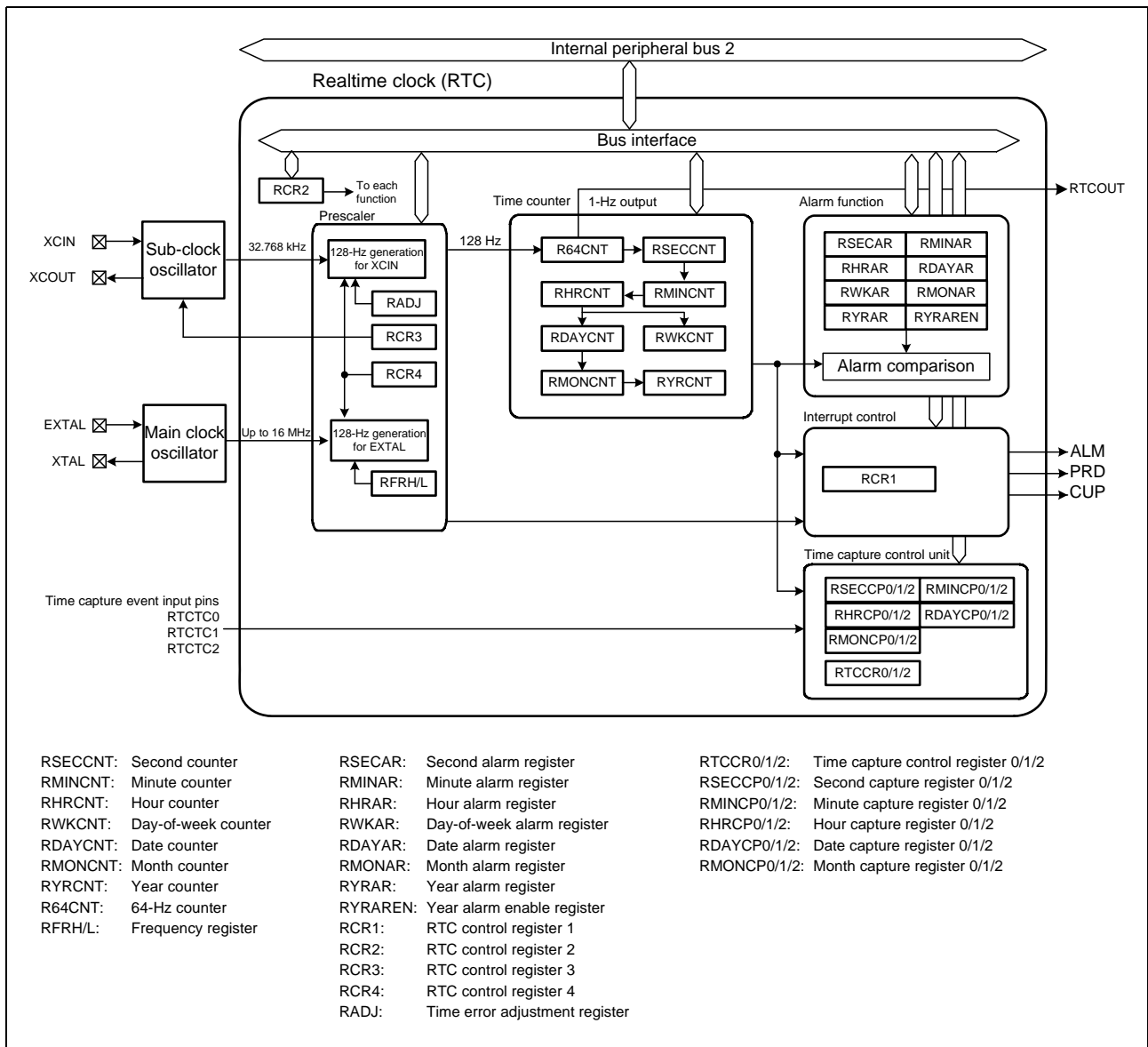


Figure 28.1 Block Diagram of RTC

Table 28.2 Pin Configuration of RTC

Pin Name	I/O	Function
XCIN	Input	Connect a 32.768-kHz crystal resonator for the RTC to these pins. For details, see section 9, Clock Generation Circuit.
XCOU	Output	
EXTAL	Input	Connect a crystal resonator for the main clock. For details, see section 9, Clock Generation Circuit.
XTAL	Output	
RTCOUT	Output	Output a 1-Hz waveform, but this clock is not output in deep software standby mode.
RTIC0	Input	Pins for time capture event input
RTIC1	Input	
RTIC2	Input	



## 28.2 Register Descriptions

When writing to or reading from RTC registers, do so in accord with section 28.5.5, Points for Caution When Writing to and Reading from Registers.

If the value in an RTC register after a reset is given as x (undefined bits) in the list, it is not initialized by a reset or in deep standby mode. When RTC enters the reset state or a low power consumption state during counting operations (i.e. while the RCR2.START bit is 1), the year, month, date, day of the week, hours, minutes, seconds, and 64-Hz counters continue to operate. However, note that a reset generated during writing to or updating of a register might destroy the register value. In addition, do not allow the chip to enter software standby mode or deep software standby mode just after setting any of these registers. For details, refer to section, section 28.5.4, Transitions to Low Power Consumption Modes after Setting Registers.

### 28.2.1 64-Hz Counter (R64CNT)

Address(es): 0008 C400h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	F1HZ	F2HZ	F4HZ	F8HZ	F16HZ	F32HZ	F64HZ
Value after reset:	0	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	F64HZ	64Hz	Indicate the state between 1-Hz and 64-Hz.	R
b1	F32HZ	32Hz		R
b2	F16HZ	16Hz		R
b3	F8HZ	8Hz		R
b4	F4HZ	4Hz		R
b5	F2HZ	2Hz		R
b6	F1HZ	1Hz		R
b7	—	Reserved	This bit is read as 0. Writing to this bit has no effect.	R

The 64-Hz counter (R64CNT) generates the period for a second by counting up periods of the 128-Hz clock.

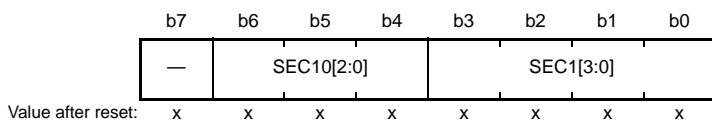
The state in the sub-seconds range can be confirmed by reading this counter.

This counter is cleared to 00h by a RTC software reset or executing 30-second adjustment.

To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

## 28.2.2 Second Counter (RSECCNT)

Address(es): 0008 C402h



Value after reset: x x x x x x x x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	Ones Place of Seconds	Counts from 0 to 9 once per second. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	SEC10[2:0]	Tens Place of Seconds	Counts from 0 to 5 for 60-second counting.	R/W
b7	—	Reserved	Set this bit to 0. It is read as 0.	R/W

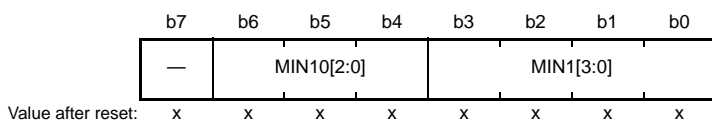
RSECCNT is used for setting and counting the BCD-coded second value. It counts carries generated once per second in the 64-Hz counter.

The setting range is decimal 00 to 59. The RTC will not operate normally if any other value is set. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

## 28.2.3 Minute Counter (RMINCNT)

Address(es): 0008 C404h



Value after reset: x x x x x x x x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	Ones Place of Minutes	Counts from 0 to 9 once per minute. When a carry is generated, 1 is added to the tens place.	R/W
b6 to b4	MIN10[2:0]	Tens Place of Minutes	Counts from 0 to 5 for 60-minute counting.	R/W
b7	—	Reserved	Set this bit to 0. It is read as 0.	R/W

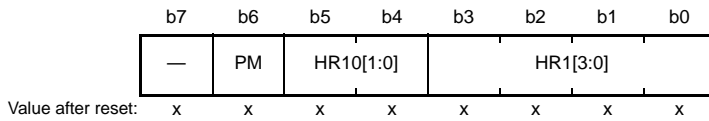
RMINCNT is used for setting and counting the BCD-coded minute value. It counts carries generated once per minute in the second counter.

A value from 00 through 59 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

## 28.2.4 Hour Counter (RHRCNT)

Address(es): 0008 C406h



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	Ones Place of Hours	Counts from 0 to 9 once per hour. When a carry is generated, 1 is added to the tens place.	R/W
b5, b4	HR10[1:0]	Tens Place of Hours	Counts from 0 to 2 once per carry from the ones place.	R/W
b6	PM	PM	Time Counter Setting for a.m./p.m. 0: a.m. 1: p.m.	R/W
b7	—	Reserved	Set this bit to 0. It is read as 0.	R/W

RHRCNT is used for setting and counting the BCD-coded hour value. It counts carries generated once per hour in the minute counter.

The specifiable time differs according to the setting in the time mode bit (RCR2.HR24).

When the RCR2.HR24 bit is 0: From 00 to 11 (practically in BCD)

When the RCR2.HR24 bit is 1: From 00 to 23 (practically in BCD)

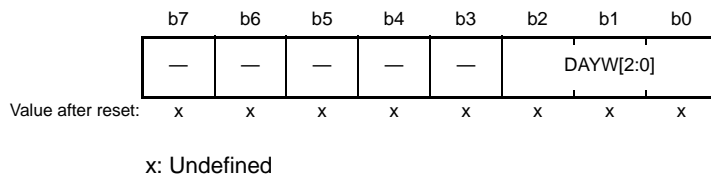
If a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

The PM bit is only enabled when the RCR2.HR24 bit is 0. Otherwise, the setting in the PM bit has no effect.

To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

## 28.2.5 Day-of-Week Counter (RWKCNT)

Address(es): 0008 C408h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DAYW[2:0]	Day-of-Week Counting	b2 b0 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting Prohibited	R/W
b7 to b3	—	Reserved	Set these bits to 0. They are read as 0.	R/W

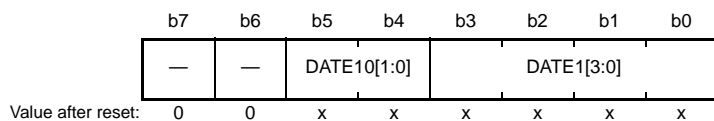
RWKCNT is used for setting and counting in the BCD-coded day-of-week value. It counts carries generated once per day in the hour counter.

A value from 0 through 6 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

## 28.2.6 Date Counter (RDAYCNT)

Address(es): 0008 C40Ah



Value after reset:

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DATE1[3:0]	Ones Place of Days	Counts from 0 to 9 once per day. When a carry is generated, 1 is added to the tens place.	R/W
b5, b4	DATE10[1:0]	Tens Place of Days	Counts from 0 to 3 once per carry from the ones place.	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

RDAYCNT is used for setting and counting the BCD-coded date value. It counts carries generated once per day in the hour counter. The count operation depends on the month and whether the year is a leap year.

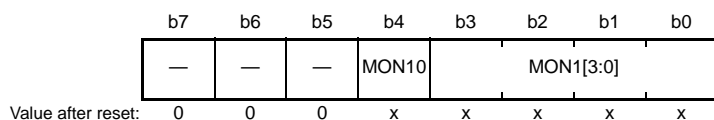
Leap years are determined according to whether the year counter (RYRCNT) value is divisible by 400, 100, and 4.

A value from 01 through 31 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. (When specifying a value, note that the range of specifiable days depends on the month and whether the year is a leap year.) Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

## 28.2.7 Month Counter (RMONCNT)

Address(es): 0008 C40Ch



Value after reset:

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MON1[3:0]	Ones Place of Months	Counts from 0 to 9 once per month. When a carry is generated, 1 is added to the tens place.	R/W
b4	MON10	Tens Place of Months	Counts from 0 to 1 once per carry from the ones place.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

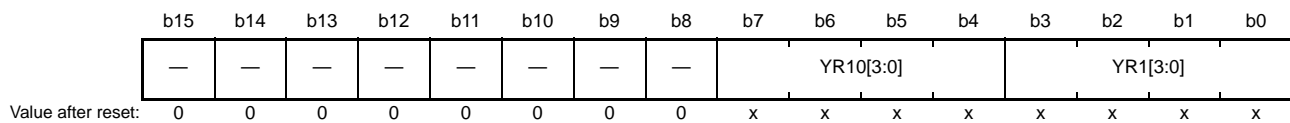
RMONCNT is used for setting and counting the BCD-coded month value. It counts carries generated once per month in the date counter.

A value from 01 through 12 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

## 28.2.8 Year Counter (RYRCNT)

Address(es): 0008 C40Eh



Value after reset:

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	YR1[3:0]	Ones Place of Years	Counts from 0 to 9 once per year. When a carry is generated, 1 is added to the tens place.	R/W
b7 to b4	YR10[3:0]	Tens Place of Years	Counts from 0 to 9 once per carry from ones place. When a carry is generated in the tens place, 1 is added to the hundreds place.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

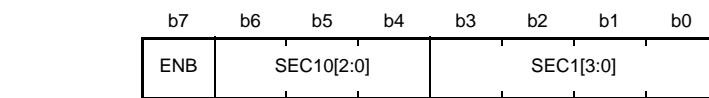
RYRCNT is used for setting and counting the BCD-coded year value. It counts carries generated once per year in the month counter.

A value from 00 through 99 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

To read this counter, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.

## 28.2.9 Second Alarm Register (RSECAR)

Address(es): 0008 C410h



Value after reset:

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	1 Second	Value for the ones place of seconds	R/W
b6 to b4	SEC10[2:0]	10 Seconds	Value for the tens place of seconds	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RSECCNT value.	R/W

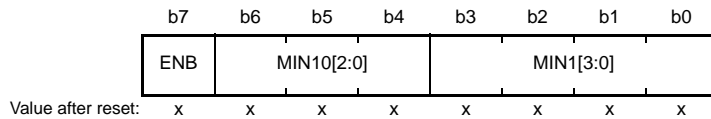
RSECAR is an alarm register corresponding to the BCD-coded second counter RSECCNT. When the ENB bit is set to 1, the RSECAR value is compared with the RSECCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.

RSECAR values from 00 through 59 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

RSECAR is cleared to 00h by a software reset of the RTC.

### 28.2.10 Minute Alarm Register (RMINAR)

Address(es): 0008 C412h



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	1 Minute	Value for the ones place of minutes	R/W
b6 to b4	MIN10[2:0]	10 Minutes	Value for the tens place of minutes	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RMINCNT value.	R/W

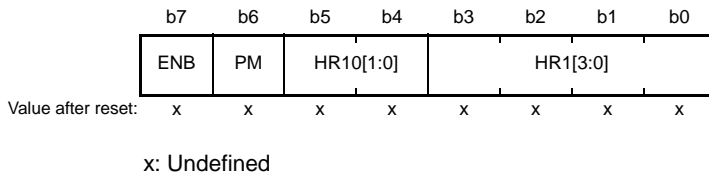
RMINAR is an alarm register corresponding to the BCD-coded minute counter RMINCNT. When the ENB bit is set to 1, the RMINAR value is compared with the RMINCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.

RMINAR values from 00 through 59 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

RMINAR is cleared to 00h by a software reset of the RTC.

### 28.2.11 Hour Alarm Register (RHRAR)

Address(es): 0008 C414h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	1 Hour	Value for the ones place of hours	R/W
b5, b4	HR10[1:0]	10 Hours	Value for the tens place of hours	R/W
b6	PM	PM	Time Alarm Setting for a.m./p.m. 0: a.m. 1: p.m.	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RHCNT value.	R/W

RHRAR is an alarm register corresponding to the BCD-coded hour counter RHCNT. When the ENB bit is set to 1, the RHRAR value is compared with the RHCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.

The specifiable time differs according to the setting in the hours mode bit (RCR2.HR24).

When the RCR2.HR24 bit is 0: From 00 to 11 (practically in BCD)

When the RCR2.HR24 bit is 1: From 00 to 23 (practically in BCD)

If a value outside of this range is specified, the RTC does not operate correctly.

When the RCR2.HR24 bit is 0, be sure to set the PM bit.

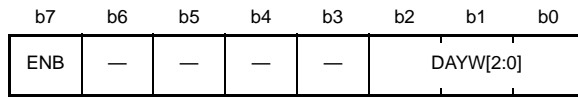
When the RCR2.HR24 bit is 1, the setting in the PM bit has no effect.

RHRAR is cleared to 00h by a software reset of the RTC.



## 28.2.12 Day-of-Week Alarm Register (RWKAR)

Address(es): 0008 C416h



Value after reset: x x x x x x x x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	DAYW[2:0]	Day-of-Week Setting	b2 b0 0 0 0: Sunday 0 0 1: Monday 0 1 0: Tuesday 0 1 1: Wednesday 1 0 0: Thursday 1 0 1: Friday 1 1 0: Saturday 1 1 1: Setting Prohibited	R/W
b6 to b3	—	Reserved	Set these bits to 0. They are read as 0.	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RWKCNT value.	R/W

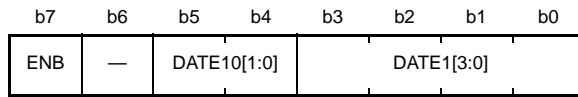
RWKAR is an alarm register corresponding to the BCD-coded day-of-week counter RWKCNT. When the ENB bit is set to 1, the RWKAR value is compared with the RWKCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.

RWKAR values from 0 through 6 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

RWKAR is cleared to 00h by a software reset of the RTC.

### 28.2.13 Date Alarm Register (RDAYAR)

Address(es): 0008 C418h



Value after reset: x x x x x x x x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DATE1[3:0]	1 Day	Value for the ones place of days	R/W
b5, b4	DATE10[1:0]	10 Days	Value for the tens place of days	R/W
b6	—	Reserved	Set this bit to 0. It is read as 0.	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RDAYCNT value.	R/W

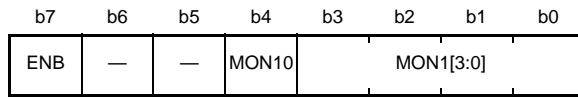
RDAYAR is an alarm register corresponding to the BCD-coded date counter RDAYCNT. When the ENB bit is set to 1, the RDAYAR value is compared with the RDAYCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.

RDAYAR values from 01 through 31 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

RDAYAR is cleared to 00h by a software reset of the RTC.

### 28.2.14 Month Alarm Register (RMONAR)

Address(es): 0008 C41Ah



Value after reset: x x x x x x x x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MON1[3:0]	1 Month	Value for the ones place of months	R/W
b4	MON10	10 Months	Value for the tens place of months	R/W
b6, b5	—	Reserved	Set these bits to 0. They are read as 0.	R/W
b7	ENB	ENB	When this bit is set to 1, this register value is compared with the RMONCNT value.	R/W

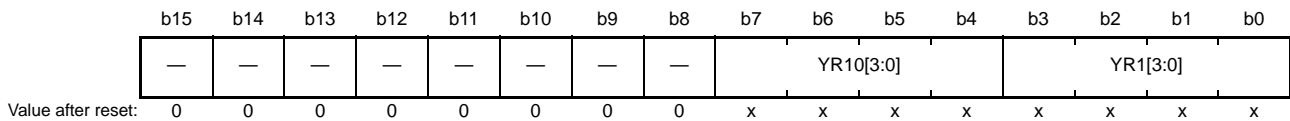
RMONAR is an alarm register corresponding to the BCD-coded month counter RMONCNT. When the ENB bit is set to 1, the RMONAR value is compared with the RMONCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.

RMONAR values from 01 through 12 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

RMONAR is cleared to 00h by a software reset of the RTC.

## 28.2.15 Year Alarm Register (RYRAR)

Address(es): 0008 C41Ch



Value after reset:

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	YR1[3:0]	1 Year	Value for the ones place of years	R/W
b7 to b4	YR10[3:0]	10 Years	Value for the tens place of years	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

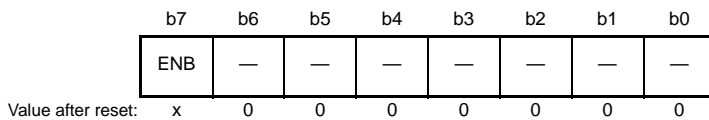
RYRAR is an alarm register corresponding to the BCD-coded year counter RYRCNT.

RYRAR values from 00 through 99 (practically in BCD) can be specified; if a value outside of this range is specified, the RTC does not operate correctly.

RYRAR is cleared to 0000h by a software reset of the RTC.

## 28.2.16 Year Alarm Enable Register (RYRAREN)

Address(es): 0008 C41Eh



Value after reset:

x: Undefined

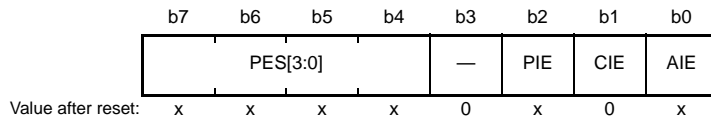
Bit	Symbol	Bit Name	Description	R/W
b6 to b0	—	Reserved	Set these bits to 0. They are read as 0.	R/W
b7	ENB	ENB	When this bit is set to 1, the RYRAR value is compared with the RYRCNT value.	R/W

When the ENB bit in RYRAREN is set to 1, the RYRAR value is compared with the RYRCNT value. From among the alarm registers (RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, and RYRAREN), only those selected with the ENB bits set to 1 are compared with the corresponding counters. When the respective values all match, the corresponding IR92.IR flag of the ICU is set to 1.

RYRAREN is cleared to 00h by a software reset of the RTC.

## 28.2.17 RTC Control Register 1 (RCR1)

Address(es): 0008 C422h



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	AIE	Alarm Interrupt Enable	0: An alarm interrupt request is disabled 1: An alarm interrupt request is enabled	R/W
b1	CIE	Carry Interrupt Enable	0: A carry interrupt request is disabled. 1: A carry interrupt request is enabled.	R/W
b2	PIE	Periodic Interrupt Enable	0: A periodic interrupt request is disabled. 1: A periodic interrupt request is enabled.	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7 to b4	PES[3:0]	Periodic Interrupt Select	<div style="display: flex; align-items: flex-start;"> <div style="margin-right: 10px;"> <b>b7</b> 0 1 1 0 0 1 1 1 1 0 0 0 1 0 0 1 1 0 1 0 1 0 1 1 1 1 0 0 1 1 0 1 1 1 1 0 1 1 1 1 </div> <div>           0: A periodic interrupt is generated every 1/256 second.            (However, when the main clock is selected (RCR4.RCKSEL = 1) while PES[3:0] = 0110b, a periodic interrupt is generated every 1/128 second.)            1: A periodic interrupt is generated every 1/128 second.            0 1 0 0: A periodic interrupt is generated every 1/64 second.            1 0 0 1: A periodic interrupt is generated every 1/32 second.            1 0 1 0: A periodic interrupt is generated every 1/16 second.            1 0 1 1: A periodic interrupt is generated every 1/8 second.            1 1 0 0: A periodic interrupt is generated every 1/4 second.            1 1 0 1: A periodic interrupt is generated every 1/2 second.            1 1 1 0: A periodic interrupt is generated every 1 second.            1 1 1 1: A periodic interrupt is generated every 2 seconds.            Other than above, no periodic interrupts are generated.         </div> </div>	R/W

Bits AIE, PIE, and PES[3:0] are updated in synchronization with the count source. When RCR1 is modified, check that all the bits have been updated without fail before continuing with further processing.

### AIE Bit (Alarm Interrupt Enable)

This bit enables or disables alarm interrupt requests.

If the times indicated by the counters and alarm settings match in deep software standby mode, the mode is canceled regardless of the value of the AIE bit.

### CIE Bit (Carry Interrupt Enable)

This bit enables or disables interrupt requests when a carry to the second counter (RSECCNT) occurs or a carry to the 64-Hz counter (R64CNT) occurs during read access to the 64-Hz counter.

### PIE Bit (Periodic Interrupt Enable)

This bit enables or disables a periodic interrupt.

If the times indicated by the counters and PES[3:0] settings match in deep software standby mode, the mode is canceled regardless of the value of the PIE bit.

### PES[3:0] Bits (Periodic Interrupt Select)

These bits specify the period for the periodic interrupt. A periodic interrupt (PRD) is generated with the period specified by these bits.

## 28.2.18 RTC Control Register 2 (RCR2)

Address(es): 0008 C424h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	HR24	AADJP	AADJE	RTCOE	ADJ30	RESET	START
Value after reset:	x	x	x	x	0	0	0	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	START	Start	0: Year, month, day-of-week, date, hour, minute, second, and 64-Hz counters, and prescaler are stopped. 1: Year, month, day-of-week, date, hour, minute, second, and 64-Hz counters, and prescaler operate normally.	R/W
b1	RESET	RTC Software Reset	<ul style="list-style-type: none"> <li>In writing               <ul style="list-style-type: none"> <li>0: Writing is invalid.</li> <li>1: The prescaler and target registers (R64CNT, RSECAR, RMINAR, RHRAR, RWKAR, RDAYAR, RMONAR, RYRAR, RYRAREN, RADJ, RTCCRY, RSECCPY, RMINCPY, RHRCPY, RDAYCPY, RMONCPY, RCR2.ADJ30, RCR2.AADJE, and RCR2.RTCOE*2) are reset by RTC software reset.</li> </ul> </li> <li>In reading               <ul style="list-style-type: none"> <li>0: In normal clock operation, or RTC software reset has completed.</li> <li>1: During a RTC software reset</li> </ul> </li> </ul>	R/W
b2	ADJ30	30-Second Adjustment	<ul style="list-style-type: none"> <li>In writing               <ul style="list-style-type: none"> <li>0: Writing is invalid.</li> <li>1: 30-second adjustment is executed.</li> </ul> </li> <li>In reading               <ul style="list-style-type: none"> <li>0: In normal clock operation, or 30-second adjustment has completed.</li> <li>1: During 30-second adjustment</li> </ul> </li> </ul>	R/W
b3	RTCOE	RTCOUT Output Enable	0: RTCOUT output disabled. 1: RTCOUT output enabled.	R/W
b4	AADJE	Automatic Adjustment Enable*1	0: Automatic adjustment is disabled. 1: Automatic adjustment is enabled.	R/W
b5	AADJP	Automatic Adjustment Period Select*1	0: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every one minute. 1: The RADJ.ADJ[5:0] setting value is adjusted from the count value of the prescaler every ten seconds.	R/W
b6	HR24	Hours Mode	0: The RTC operates in 12-hour mode. 1: The RTC operates in 24-hour mode.	R/W
b7	—	Reserved	Set this bit to 0. It is read as 0.	R/W

Note 1. When the main clock is selected, the AADJE and AADJP settings are invalid.

Note 2. In products tagged with “\*2” in Figure 1.3, List of Products, RCR3 is not initialized by software resetting the RTC. For other products, an RTC software reset clears RCR3 to 00h.

### START Bit (Start)

This bit stops or restarts the prescaler or counter (clock) operation.

The START bit is updated in synchronization with the count source. When the START bit is modified, check that the bit is updated without fail, and then make next settings.

### RESET Bit (RTC Software Reset)

This bit initializes the prescaler and registers to be reset by RTC software reset.

When 1 is written to the RESET bit, the initialization starts in synchronization with the count source. When the

initialization is completed, the RESET bit is automatically cleared to 0.

When 1 is written to the RESET bit, check that the bit is cleared to 0, and then make next settings.

#### **ADJ30 Bit (30-Second Adjustment)**

This bit is for 30-second adjustment.

When 1 is written to the ADJ30 bit, the RSECCNT value of 30 seconds or less is rounded down to 00 second and the value of 30 seconds or more is rounded up to one minute.

The 30-second adjustment is performed in synchronization with the count source. When 1 is written to this bit, the ADJ30 bit is automatically cleared to 0 after the 30-second adjustment is completed. In case when 1 is written to the ADJ30 bit, check that the bit is cleared to 0, and then make next settings.

When the 30-second adjustment is performed, the prescaler and R64CNT are also reset.

The ADJ30 bit is cleared to 0 by RTC software reset.

#### **RTCOE Bit (RTCOU Output Enable)**

This bit enables output of a 1-Hz clock signal from the RTCOUT pin.

Use the START bit to stop counting by the counters before changing the value of the RTCOE bit. Do not stop counting (write 0 to the START bit) and change the value of the RTCOE bit at the same time.

When RTCOUT is to be output from an external pin, enable the RTCOE bit and set up the port control for the pin.

#### **AADJE Bit (Automatic Adjustment Enable)**

This bit controls (enables or disables) automatic adjustment.

Set the plus-minus bits (PMADJ[1:0] in RADJ) to 00b (no adjustment) before changing the value of the AADJE bit.

The AADJE bit is cleared to 0 by RTC software reset.

#### **AADJP Bit (Automatic Adjustment Period Select)**

This bit selects the automatic-adjustment period.

Set the plus-minus bits (PMADJ[1:0] in RADJ) to 00b (no adjustment) before changing the value of the AADJP bit.

The AADJP bit is cleared to 0 by RTC software reset.

#### **HR24 Bit (Hours Mode)**

This bit specifies whether the RTC will operate in 12- or 24-hour mode.

Use the START bit to stop counting by the counters before changing the value of the HR24 bit. Do not stop counting (write 0 to the START bit) and change the value of the HR24 bit at the same time.

### 28.2.19 RTC Control Register 3 (RCR3)

Address(es): 0008 C426h



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	RTCEN	Sub-clock Control	0: Sub-clock oscillator is stopped. 1: Sub-clock oscillator is running.	R/W
b3 to b1	—	Reserved	Set these bits to 110b.*1 They are read as 110b.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Writing to bits b3 to b1 during operation of the sub-clock oscillator is prohibited. Only write to the b3 to b1 bits while the sub-clock oscillator stop bit (SOSCCR.SOSTP) in the sub-clock oscillator control register is set to 1 and the sub-clock oscillator control bit (RCR3.RTCEN) in RTC control register 3 is set to 0.

RCR3 controls the sub-clock oscillator in the clock generation circuit.

When writing a new value to the RCR3 register, check that the value written is reflected in all of the bits before proceeding with further processing.

For control on sub-clock oscillator, see section 9, Clock Generation Circuit.

In products tagged with “\*2” in Table 1.3, List of Products, RCR3 is not initialized by software resetting the RTC. For other products, an RTC software reset clears RCR3 to 00h.

#### RTCEN Bit (Sub-Clock Oscillator Control)

This bit makes the sub-clock oscillator run or stop.

Running and stopping of the sub-clock oscillator is controlled by the RTCEN bit and a clock generation circuit register. To use the sub-clock as the clock source for counting by the RTC, set the RTCEN bit to cause the sub-clock oscillator to operate.

When the extended clock is selected (RCR4.RCKSEL bit = 1), making the sub-clock oscillator run or stop is controlled only by the clock generation circuit register regardless of the value set in RTCEN bit.

#### RTCDV[2:0] Bits (Sub-clock Oscillator Driving Ability Control)

The driving ability of the sub-clock oscillator is controllable. The setting 110b is for crystal oscillators to drive a standard clock load and the setting 001b is for crystal oscillators to drive a low clock load.

Set the RTCDV[2:0] bits while the value of the sub-clock oscillator stop bit (SOSTP) in the sub-clock oscillator control register (SOSCCR) is 1 and that of the sub-clock control bit (RTCEN) in RTC control register 3 (RCR3) is 0.



### 28.2.20 RTC Control Register 4 (RCR4)

Address(es): 0008 C428h

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

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	RCKSEL	Count Source Select	0: The sub-clock is selected. 1: The main clock is selected.	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

When the RCKSEL bit is set to 0, the time is counted with the sub-clock. When the bit is set to 1, the time is counted with the main clock.

#### RCKSEL Bit (Count Source Select)

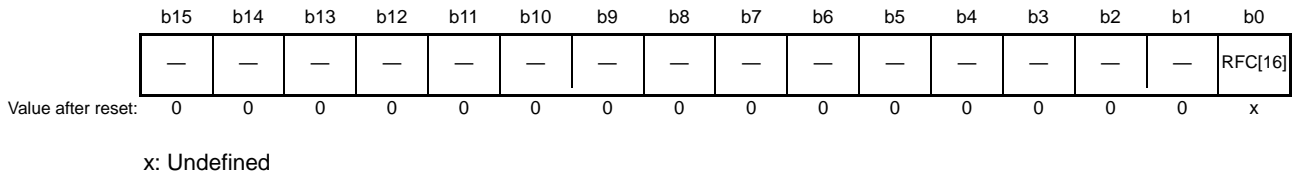
This bit selects the count source from the sub-clock and the main clock.

The count source should be selected only once before the initial settings of the RTC registers at power on.

## 28.2.21 Frequency Register H/L (RFRH/L)

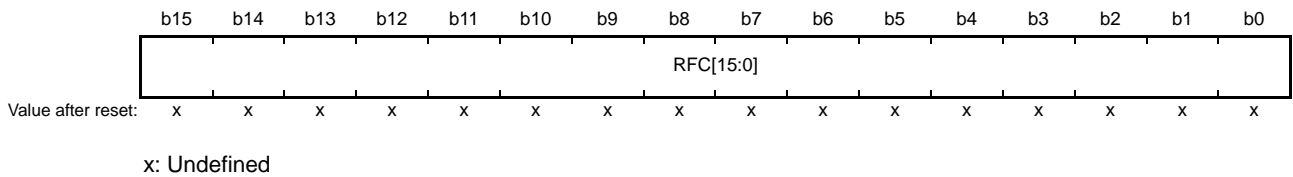
- RFRH register

Address(es): 0008 C42Ah



- RFRL register

Address(es): 0008 C42Ch



### RFRH register

Bit	Symbol	Bit Name	Description	R/W
b0	RFC[16]	Frequency Comparison Value	To produce the operation clock from the main clock, this bit sets the comparison value of the 128-Hz clock cycle.	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

### RFRL register

Bit	Symbol	Bit Name	Description	R/W
b15 to b0	RFC[15:0]	Frequency Comparison Value	To prepare the operation clock from the main clock, these bits set the comparison value of the 128-Hz clock cycle.	R/W

RFRH/L is a register to control the prescaler when the main clock is selected.

The RTC time counter operates on a 128-Hz clock signal as the base clock. Therefore, when the main clock is selected, the main clock is divided by the prescaler to produce a 128-Hz clock signal. Set the frequency comparison value in bits RFC[16:0] to produce a 128-Hz clock from the main clock frequency. The frequency comparison value should be changed depending on the main clock frequency. As for the calculation method, refer to the formula below.

A value from 0000 0007h through 0001 FFFFh can be specified as the frequency comparison value; if a value outside of this range is specified, the RTC does not operate correctly. Before writing to this register, be sure to stop the count operation through the setting of the START bit in RCR2.

In addition, when the main clock is 16.778 MHz or more, the frequency comparison value cannot be set.

The operating frequency of the peripheral module clock and the main clock should be in the relationship that peripheral module clock  $\geq$  main clock.

- Calculation method of frequency comparison value

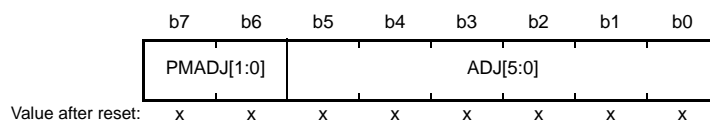
$$\text{RFC}[16:0] = (\text{Main clock frequency})/128 - 1$$

**Table 28.3 RFRH/L Register Settings by the Main Clock Frequency**

Main Clock Frequency	RFRH/L Register Settings
4 MHz	0000 7A11h
8 MHz	0000 F423h
10 MHz	0001 312Ch
12 MHz	0001 6E35h
16 MHz	0001 E847h

### 28.2.22 Time Error Adjustment Register (RADJ)

Address(es): 0008 C42Eh



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b5 to b0	ADJ[5:0]	Adjustment Value	These bits specify the adjustment value from the prescaler.	R/W
b7, b6	PMADJ[1:0]	Plus–Minus	b7 b6 0 0: Adjustment is not performed. 0 1: Adjustment is performed by the addition to the prescaler. 1 0: Adjustment is performed by the subtraction from the prescaler. 1 1: Setting prohibited	R/W

RADJ is a register to adjust the time error.

Adjustment is performed by the addition to or subtraction from the prescaler.

When the sub-clock is selected, settings in RADJ are valid.

When the main clock is selected, adjustment is not performed regardless of settings in RADJ.

In case when the automatic adjustment enable (RCR2.AADJE) bit is 0, adjustment is performed when writing to the RADJ.

In case when the RCR2.AADJE bit is 1, adjustment is performed in the interval specified by the automatic adjustment period select (RCR2.AADJP) bit.

The current adjustment by software may be invalid if the following adjustment value is specified within 320 cycles of the count source after the register setting. To perform adjustment consecutively, wait for 320 cycles or more of the count source after the register setting and then specify the next adjustment value.

RADJ is updated in synchronization with the count source. When RADJ is modified, check that all the bits have been updated without fail before continuing with further processing.

This register is cleared to 00h by RTC software reset.

#### ADJ[5:0] Bits (Adjustment Value)

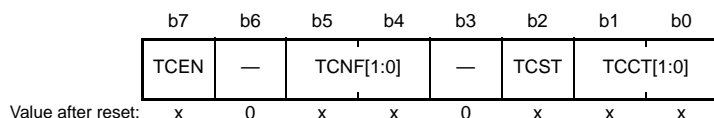
These bits specify the adjustment value (clock cycle number of sub-clock) from the prescaler.

#### PMADJ[1:0] Bits (Plus–Minus)

These bits select whether the clock is set ahead or back depending on the error-adjustment value set in the ADJ[5:0] bits.

### 28.2.23 Time Capture Control Register y (RTCCRy) (y = 0 to 2)

Address(es): RTCCR0 0008 C440h, RTCCR1 0008 C442h, RTCCR2 0008 C444h



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TCCT[1:0]	Time Capture Control	b1 b0 0 0: No event is detected. 0 1: Rising edge is detected. 1 0: Falling edge is detected. 1 1: Both edges are detected.	R/W
b2	TCST	Time Capture Status	Indicates that an event has been detected. Writing 0 clears this bit. 0: No event is detected. 1: An event is detected.*1	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5, b4	TCNF[1:0]	Time Capture Noise Filter Control	b5 b4 0 0: The noise filter is off. 0 1: Setting prohibited 1 0: The noise filter is on (count source). 1 1: The noise filter is on (count source of 32-division).	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	TCEN	Time Capture Event Input Pin Enable	0: The RTCICn pin is invalid as the time capture event input. 1: The RTCICn pin is valid as the time capture event input. (n = 0 to 2)	R/W

Note 1. Writing 1 to this bit has no effect.

RTCCR0, RTCCR1, and RTCCR2 control the RTCIC0, RTCIC1, and RTCIC2 pins, respectively.

RTCCRy is updated in synchronization with the count source. When RTCCRy is modified, check that all the bits except for the TCST bit have been updated without fail before continuing with further processing.

This register is cleared to 00h by RTC software reset.

#### TCCT[1:0] Bits (Time Capture Control)

These pins control the edge detection of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2). The detection edge is selectable. The TCCT[1:0] bits should be set while the TCEN bit is 1.

#### TCST Bit (Time Capture Status)

This pin indicates that an event of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2) has been detected. When the TCST bit is 0, no event is detected.

When the TCST bit is 1, this bit indicates that an event of the corresponding pin has been detected and the capture register is valid. When multiple events have been detected, the capture time for the first event is retained.

If an event is detected during the count operation is stopped (the RCR2.START bit is 0), the captured value is not guaranteed. In this case, write 0 to clear the TCST bit for deleting the captured value.

Writing 0 clears the TCST bit. In addition, writing any other value except 0 has no effect.

Clear the TCST bit while the TCCT[1:0] bits are 00b (no event is detected).

The TCST bit is cleared in synchronization with the count source. When the TCST bit is cleared, check that the bit has been updated without fail before continuing with further processing.

**TCNF[1:0] Bits (Time Capture Noise Filter Control)**

These bits control the noise filter of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2).

When the noise filter is on, the count source of 1- and 32-division is selectable. In this case, when the input level on the time capture event input pin matches three times at the set sampling period, the input level is determined.

Set the TCNF[1:0] bits while the TCCT[1:0] bits are 00b (no event is detected). When the noise filter is used, set the TCNF[1:0] bits, wait for three cycles of the specified sampling period, and then set the TCCT[1:0] bits. Set the TCNF[1:0] bits when the TCEN bit is 1.

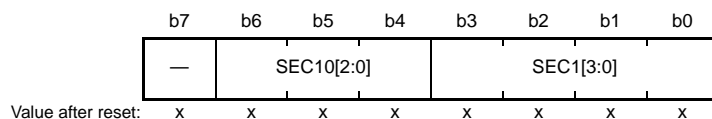
**TCEN Bit (Time Capture Event Input Pin Enable)**

This bit enables or disables the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2).

When the functions of the time capture event input pins (RTCIC0, RTCIC1, and RTCIC2) are multiplexed, set the port control and enable this bit. In this case, the port control should be set first. If the TCEN bit is set to 0, set also the TCCT[1:0] bits to 00b.

**28.2.24 Second Capture Register y (RSECCPy) (y = 0 to 2)**

Address(es): RSECCP0 0008 C452h, RSECCP1 0008 C462h, RSECCP2 0008 C472h



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	SEC1[3:0]	Ones Place of Seconds Captured	Capture value for the ones place of seconds	R
b6 to b4	SEC10[2:0]	Tens Place of Seconds Captured	Capture value for the tens place of seconds	R
b7	—	Reserved	Set this bit to 0. It is read as 0.	R/W

RSECCPy is a readable register that captures the RSECCNT value when a time capture event is detected.

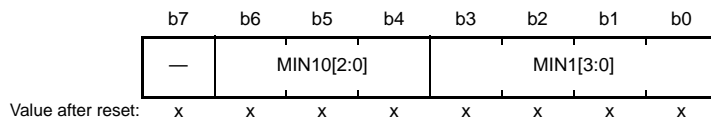
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RSECCP0, RSECCP1, and RSECCP2 registers, respectively.

This register is cleared to 00h by RTC software reset.

Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

### 28.2.25 Minute Capture Register y (RMINCPy) (y = 0 to 2)

Address(es): RMINCP0 0008 C454h, RMINCP1 0008 C464h, RMINCP2 0008 C474h



Value after reset: x x x x x x x x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MIN1[3:0]	Ones Place of Minutes Captured	Capture value for the ones place of minutes	R
b6 to b4	MIN10[2:0]	Tens Place of Minutes Captured	Capture value for the tens place of minutes	R
b7	—	Reserved	Set this bit to 0. It is read as 0.	R/W

RMINCPy is a readable register that captures the RMINCNT value when a time capture event is detected.

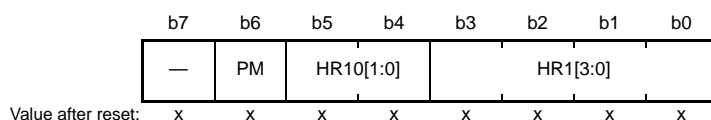
The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMINCP0, RMINCP1, and RMINCP2 registers, respectively.

This register is cleared to 00h by RTC software reset.

Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

### 28.2.26 Hour Capture Register y (RHRCPy) (y = 0 to 2)

Address(es): RHRCP0 0008 C456h, RHRCP1 0008 C466h, RHRCP2 0008 C476h



Value after reset: x x x x x x x x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	HR1[3:0]	Ones Place of Hours Captured	Capture value for the ones place of hours	R
b5, b4	HR10[1:0]	Tens Place of Hours Captured	Capture value for the tens place of hours	R
b6	PM	PM	0: a.m. 1: p.m.	R
b7	—	Reserved	Set this bit to 0. It is read as 0.	R/W

RHRCPy is a readable register that captures the RHRCNT value when a time capture event is detected.

The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RHRCP0, RHRCP1, and RHRCP2 registers, respectively.

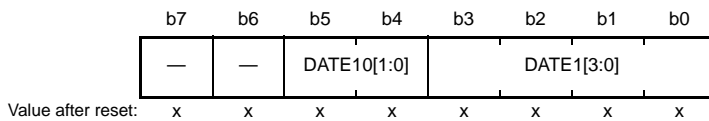
The PM bit is only enabled when the RCR2.HR24 bit is 0 (in 12-hour mode).

This register is cleared to 00h by RTC software reset.

Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

### 28.2.27 Date Capture Register y (RDAYCPy) (y = 0 to 2)

Address(es): RDAYCP0 0008 C45Ah, RDAYCP1 0008 C46Ah, RDAYCP2 0008 C47Ah



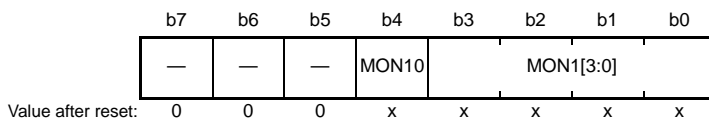
x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DATE1[3:0]	Ones Place of Days Captured	Capture value for the ones place of days	R
b5, b4	DATE10[1:0]	Tens Place of Days Captured	Capture value for the tens place of days	R
b7, b6	—	Reserved	Set these bits to 0. They are read as 0.	R/W

RDAYCPy is a readable register that captures the RDAYCNT value when a time capture event is detected. The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RDAYCP0, RDAYCP1, and RDAYCP2 registers, respectively. This register is cleared to 00h by RTC software reset. Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

### 28.2.28 Month Capture Register y (RMONCPy) (y = 0 to 2)

Address(es): RMONCP0 0008 C45Ch, RMONCP1 0008 C46Ch, RMONCP2 0008 C47Ch



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	MON1[3:0]	Ones Place of Months Captured	Capture value for the ones place of months	R
b4	MON10	Tens Place of Months Captured	Capture value for the tens place of months	R
b7 to b5	—	Reserved	These bits are read as 0. Writing to these bits has no effect.	R

RMONCPy is a readable register that captures the RMONCNT value when a time capture event is detected. The event detection times detected by the RTCIC0, RTCIC1, and RTCIC2 pins are stored in the RMONCP0, RMONCP1, and RMONCP2 registers, respectively. This register is cleared to 00h by RTC software reset. Before reading from this register, be sure to stop the time capture event detection through the setting of the RTCCRy.TCCT[1:0] bits.

## 28.3 Operation

### 28.3.1 Outline of Initial Settings of Registers after Power-On

After the power is turned on, the initial settings for the clock setting, time-error adjustment, time setting, alarm, interrupt, and time capture control register should be performed.

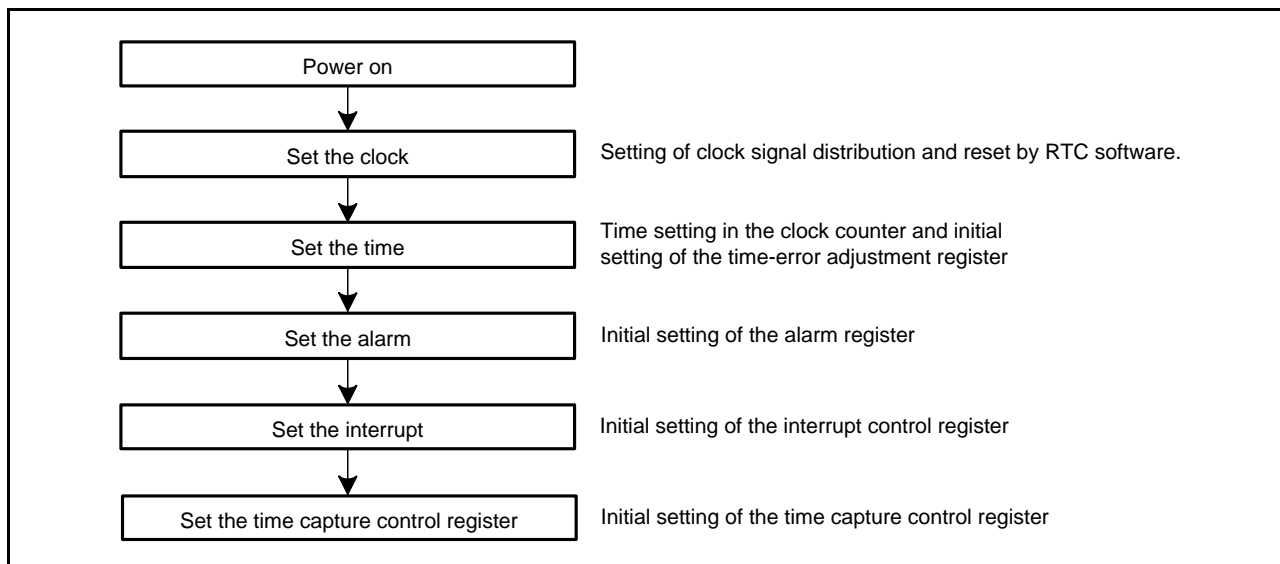


Figure 28.2 Outline of Initial Settings after Power-On



### 28.3.2 Clock Setting Procedure

Figure 28.3 shows how to set the clock.

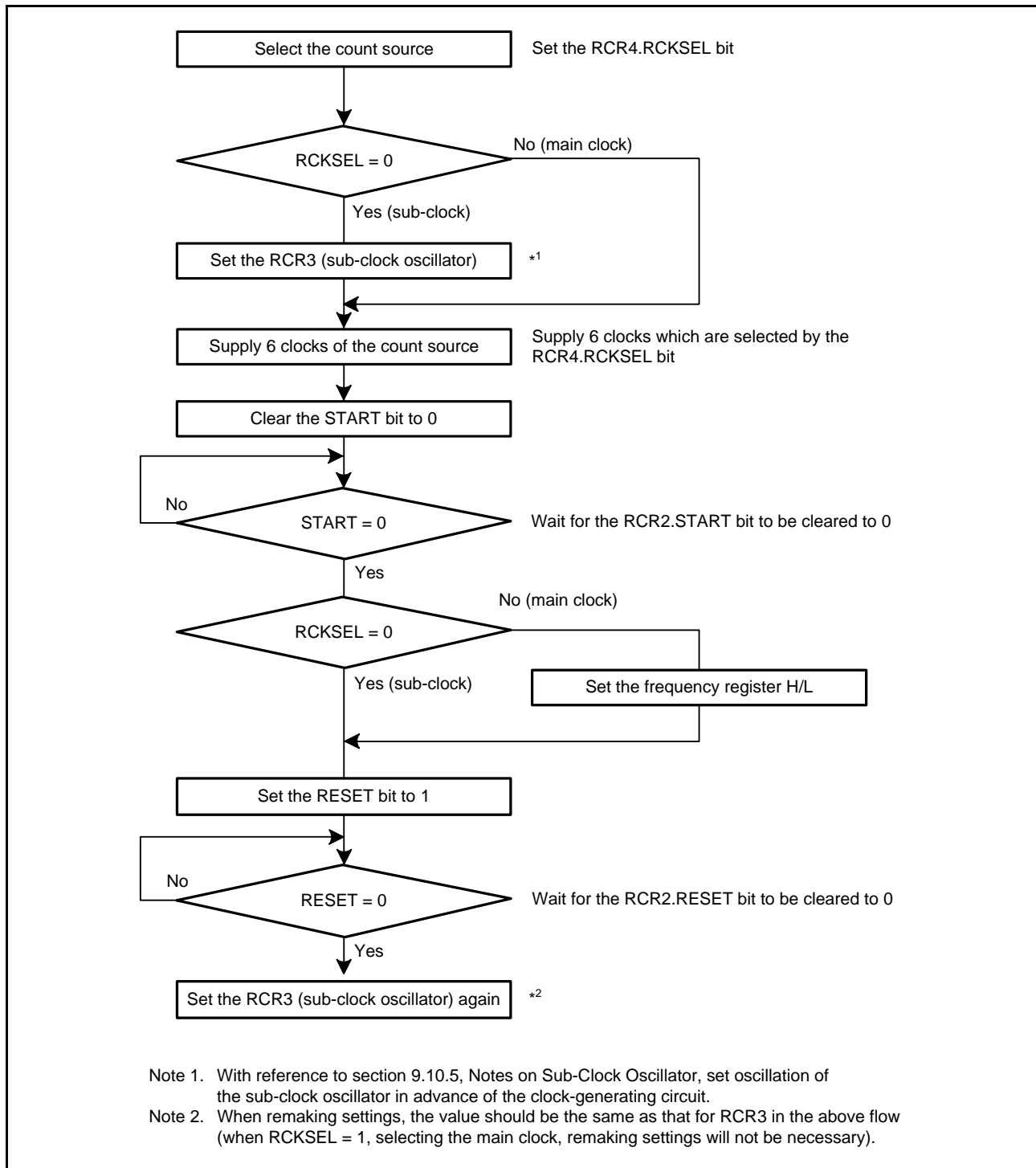


Figure 28.3 Clock Setting Procedure

### 28.3.3 Setting the Time

Figure 28.4 shows how to set the time.

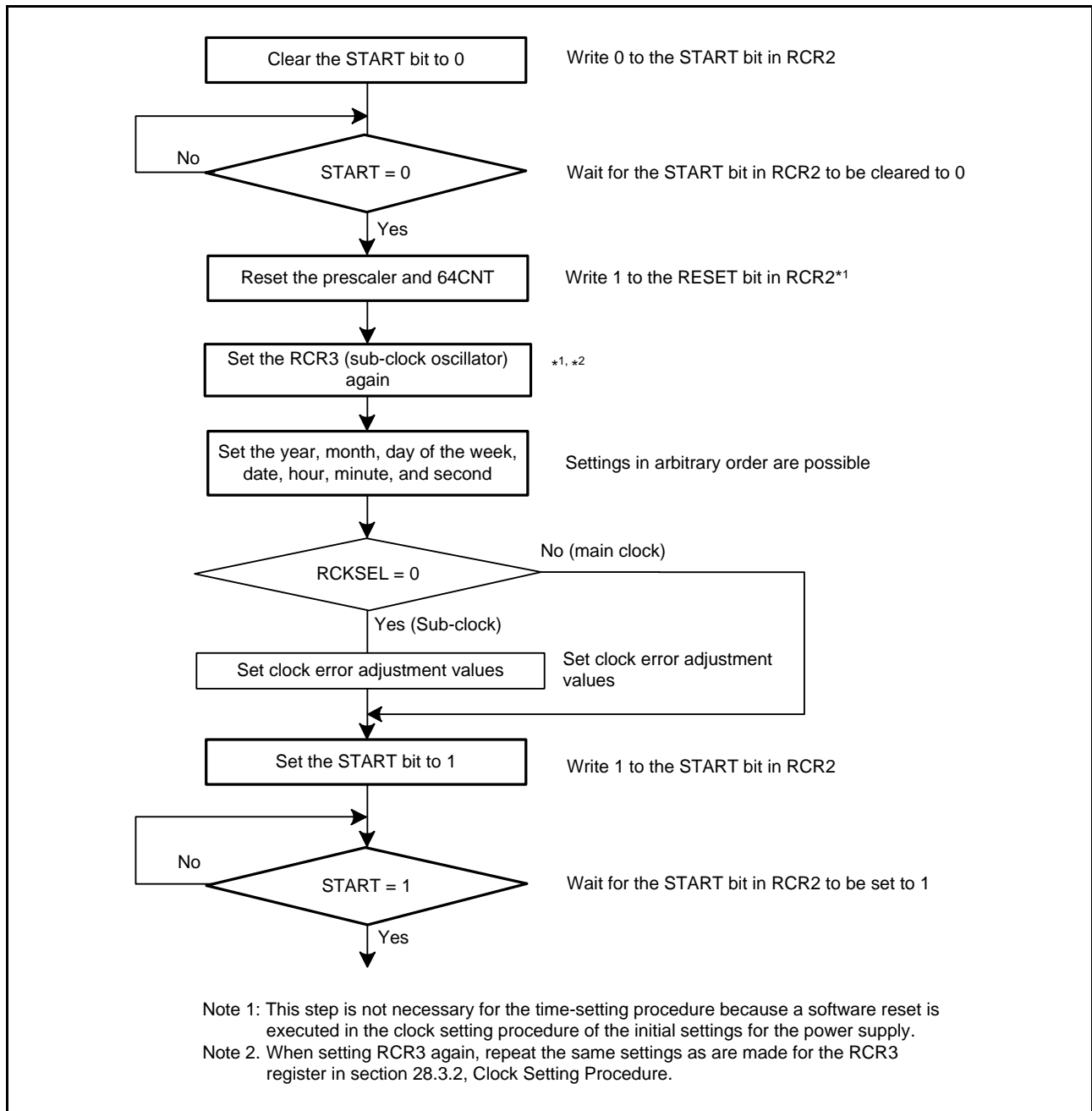
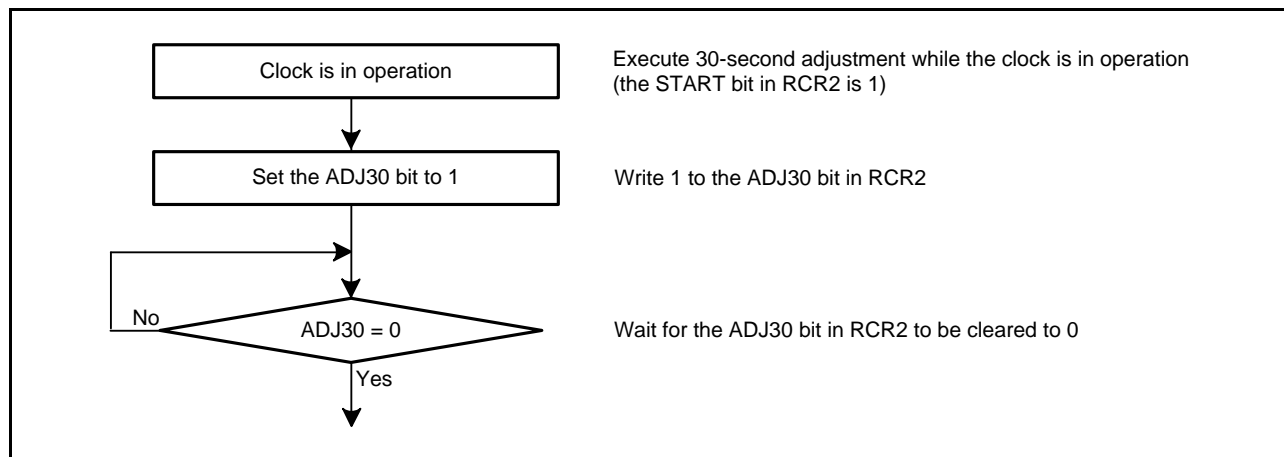


Figure 28.4 Setting the Time

### 28.3.4 30-Second Adjustment

Figure 28.5 shows how to execute 30-second adjustment.



**Figure 28.5 30-Second Adjustment**

### 28.3.5 Reading 64-Hz Counter and Time

Figure 28.6 shows how to read the 64-Hz counter and time.

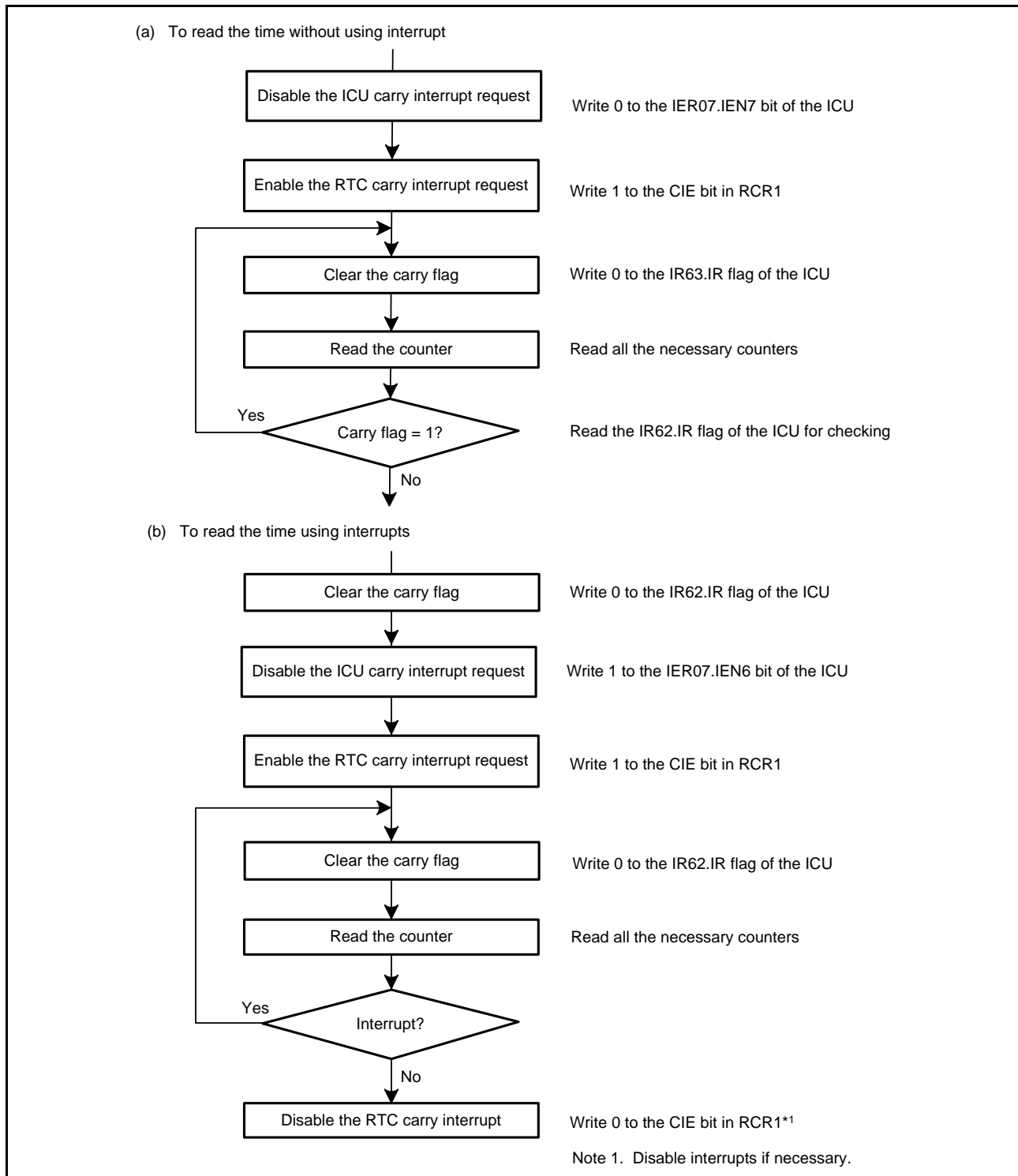
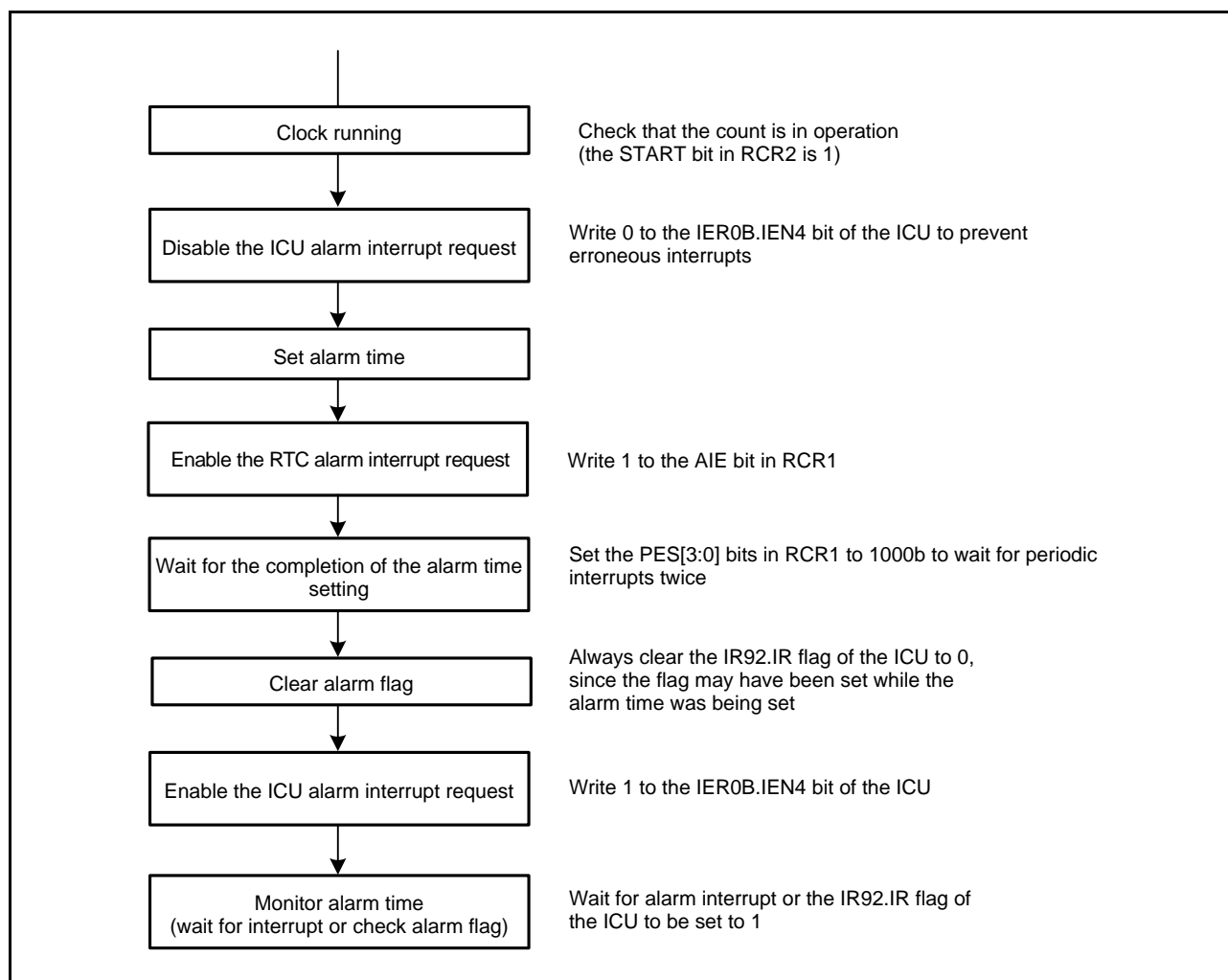


Figure 28.6 Reading Time

If a carry occurs while the 64-Hz counter and time are being read, the correct time will not be obtained, so they must be read again. The procedure for reading the time without using interrupts is shown in (a), and the procedure using carry interrupts in (b) in Figure 28.6. To keep the program simple, method (a) should be used in most cases.

### 28.3.6 Alarm Function

Figure 28.7 shows how to use the alarm function.



**Figure 28.7 Using Alarm Function**

An alarm can be generated by the year, month, date, day-of-week, hour, minute, or second value, or a combination of these. Write 1 to the ENB bit in the alarm registers involved in the alarm setting, and set the alarm time in the lower bits. Write 0 to the ENB bit in registers not involved in the alarm setting.

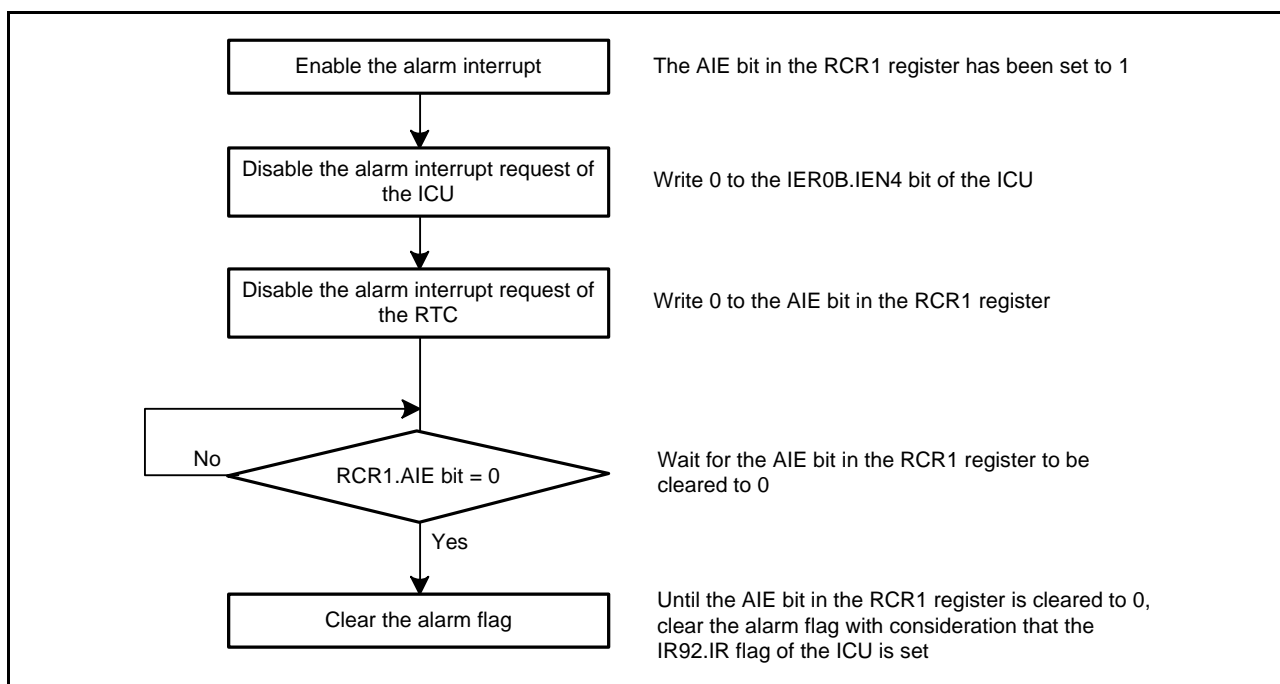
When the counter and the alarm time match, the IR92.IR flag of the ICU is set to 1. Alarm detection can be confirmed by reading this bit, but an interrupt should be used in most cases. If 1 has been set in the IER0B.IEN4 bit of the ICU, an alarm interrupt is generated in the event of alarm, enabling the alarm to be detected.

Writing 0 clears the IR92.IR flag of the ICU.

When the counter and the alarm time match in a low power consumption state, this LSI returns from the low power consumption state.

### 28.3.7 Procedure for Disabling Alarm Interrupt

Figure 28.8 shows the procedure for disabling the enabled alarm interrupt request.



**Figure 28.8 Procedure for Disabling Alarm Interrupt Request**

### 28.3.8 Time-Error Adjustment Function

The time-error adjustment function is used to correct errors (running fast or slow) in the time due to the precision of oscillation by the sub-clock. Since 32,768 cycles of the sub-clock constitute one second of operation, the clock runs fast if the sub-clock frequency is high and slow if the sub-clock frequency is low. This function can be used to correct errors due to the clock running fast or slow.

Two types of time-error adjustment functions are provided: automatic adjustment and adjustment by software.

Use the RCR2.AADJE bit to select automatic adjustment or adjustment by software.

#### 28.3.8.1 Automatic Adjustment

Enable automatic adjustment by setting the AADJE bit in RCR2 to 1.

Automatic adjustment is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register every time the adjustment period selected by the AADJP bit in RCR2 elapses.

Examples are shown below.

[Example 1] Sub-clock running at 32.769 kHz

Adjustment procedure:

At 32.769 kHz, 1 second elapses on counting up by 1 from 32,768. Since adding 60 is suitable if we have 1 minute, the time can be adjusted by adding 60 to the value counted by the prescaler once per minute.

Register settings:

- RCR2.AADJP = 0 (adjustment every minute)
- RADJ.PMADJ[1:0] = 10b (subtraction)
- RADJ.ADJ[5:0] = 60 (3Ch)

[Example 2] Sub-clock running at 32.766 kHz

Adjustment procedure:

At 32.766 kHz, 1 second elapses on counting up by 2 from 32,768. Since adding 20 is suitable if we have 10 seconds, the time can be adjusted by adding 20 to the value counted by the prescaler once every 10 seconds.

Register settings:

- RCR2.AADJP = 1 (adjustment every 10 seconds)
- RADJ.PMADJ[1:0] = 01b (addition)
- RADJ.ADJ[5:0] = 20 (14h)

### 28.3.8.2 Adjustment by Software

Enable adjustment by software by setting the AADJE bit in RCR2 to 0.

Adjustment by software is the addition or subtraction of the value counted by the prescaler to or from the value in the RADJ register at the time of execution of an instruction for writing to the RADJ register.

An example is shown below.

[Example 1] Sub-clock running at 32.769 kHz

Adjustment procedure:

At 32.769 kHz, 1 second elapses on counting up by 1 from 32,768. The time can be adjusted by adding 1 to the value counted by the prescaler once per second.

Register settings:

- RADJ.PMADJ[1:0] = 10b (subtraction)
- RADJ.ADJ[5:0] = 1 (01h)

This is written to the RADJ register once per 1-second interrupt.

### 28.3.8.3 Procedure for Changing the Mode of Adjustment

When changing the mode of adjustment, change the value of the AADJE bit in RCR2 after clearing the PMADJ[1:0] bits in RADJ to 00b (no adjustment).

Changing from adjustment by software to automatic adjustment:

- (1) Clear the PMADJ[1:0] bits in RADJ to 00b (no adjustment).
- (2) Set the AADJE bit in RCR2 to 1 (enabling automatic adjustment).
- (3) Use the AADJP bit in RCR2 to select the period of adjustment.
- (4) In RADJ, set the PMADJ[1:0] bits for addition or subtraction and the ADJ[5:0] bits to the value for use in time-error adjustment.

Changing from adjustment by automatic to software adjustment:

- (1) Clear the PMADJ[1:0] bits in RADJ to 00b (no adjustment).
- (2) Set the AADJE bit in RCR2 to 0 (enabling adjustment by software).
- (3) Proceed with adjustment by setting the RADJ.PMADJ[1:0] bits for addition or subtraction and the RADJ.ADJ[5:0] bits to the value for use in time-error adjustment at the desired time. After that, the time is adjusted every time a value is written to RADJ.

### 28.3.8.4 Procedure for Stopping Adjustment

Stop adjustment by clearing the PMADJ[1:0] bits in RADJ to 00b (no adjustment).

### 28.3.8.5 Capturing the Time

The RTC is capable of storing the month, date, hour, minute and second by detecting an edge of a signal on a time-capture event-input pin.

A noise filter can also be used on a time-capture event-input pin. When the level on the pin matches three times at the set sampling period, the noise filter conveys the level to the RTC's internal circuits, and this level is retained within the RTC until the level on the target pin for sampling has again matched three times.

The noise filter can be switched on or off for each of the time-capture event-input pins.

Operation when the noise filter is off is shown in Figure 28.9 and operation when the noise filter is on is shown in Figure 28.10.

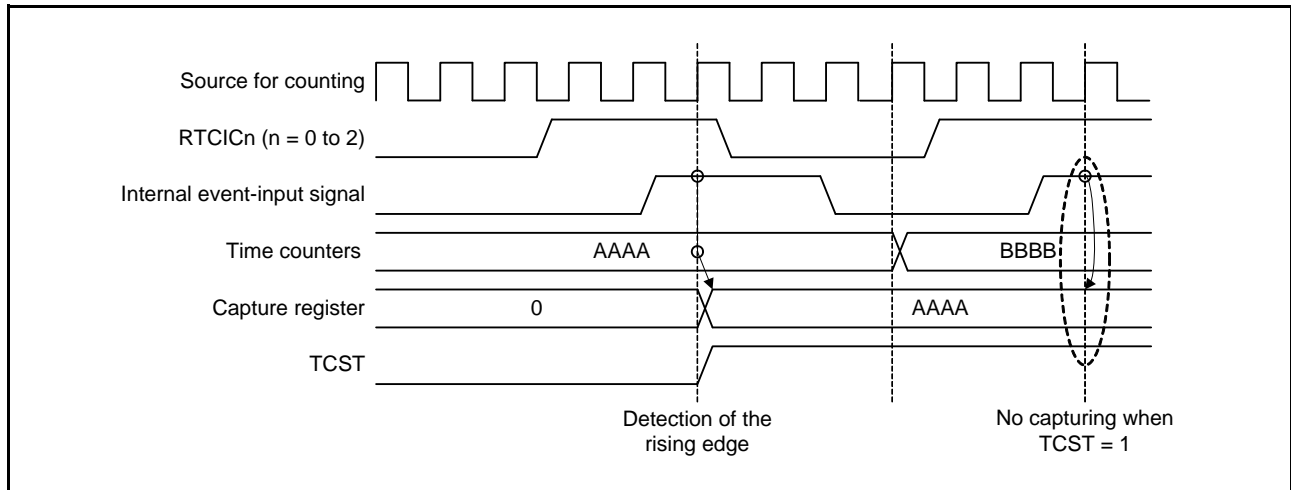


Figure 28.9 Timing of a Time-Capture Operation (with the Filter Off)

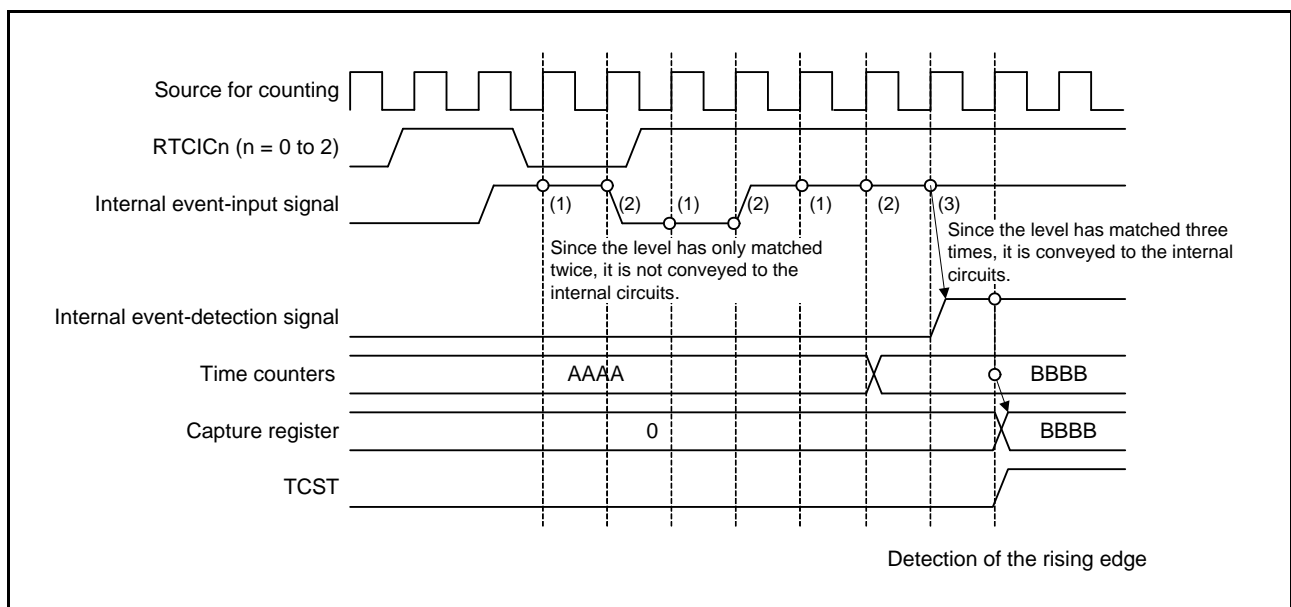


Figure 28.10 Timing of a Time-Capture Operation (with the Filter On)



## 28.4 Interrupt Sources

There are three interrupt sources in the realtime clock. Table 28.4 lists interrupt sources for the RTC.

**Table 28.4 RTC Interrupt Sources**

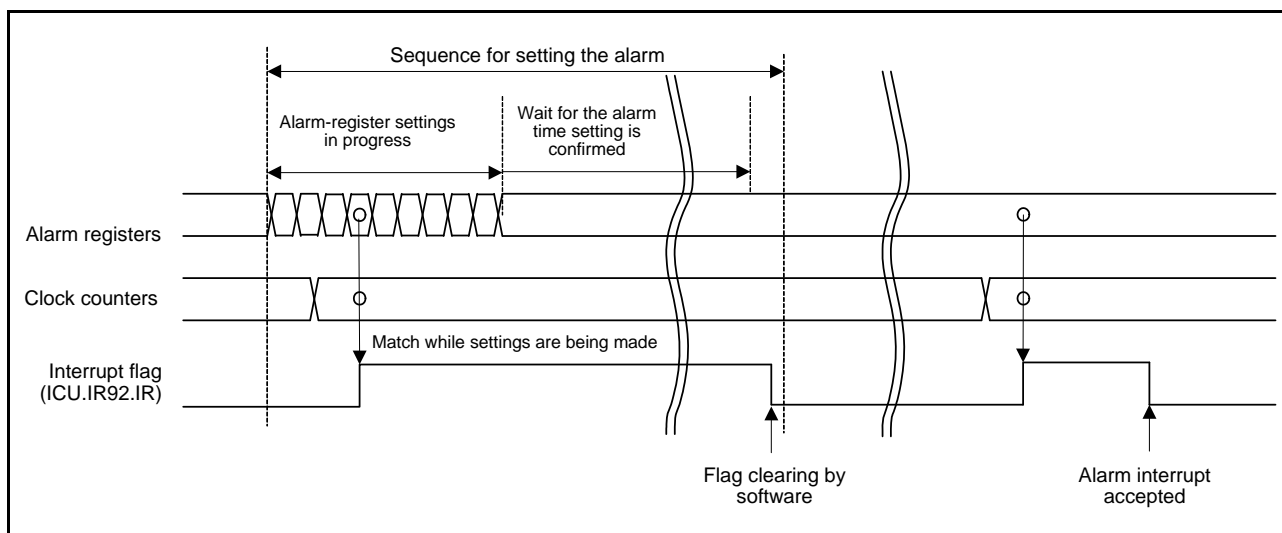
Name	Interrupt Sources	Interrupt Status Flag*1
ALM	Alarm interrupt	IPR092.IR
PRD	Periodic interrupt	IPR093.IR
CUP	Carry interrupt	IPR062.IR

Note 1. For details, see section 15, Interrupt controller (ICUb). Refer to the flag on the low power consumption side to determine the source after a return from deep software standby mode. For details, See section 11, Low Power Consumption.

### (1) Alarm interrupt (ALM)

This interrupt is generated according to the result of comparison between the alarm registers and realtime clock counters (for details, refer to the description of each alarm register).

Since there is a possibility of the interrupt flag being set when the settings of the alarm registers match the clock counters, clear the IR flag in IR92 for the interrupt after modifying values of the alarm registers. Once the interrupt flag for the alarm interrupt has been cleared and the state has returned to non-matching of the alarm registers and clock counters, the flag will not be set again until there is a further match or the values of the alarm registers are modified again.



**Figure 28.11 Timing Chart for the Alarm Interrupt (ALM)**

### (2) Periodic interrupt (PRD)

This interrupt is generated at intervals of 2 seconds, 1 second, 1/2 second, 1/4 second, 1/8 second, 1/16 second, 1/32 second, 1/64 second, 1/128 second, or 1/256 second. The interrupt interval can be selected through RCR1.PES[3:0] bits.

### (3) Carry interrupt (CUP)

This interrupt is generated when a carry to the second counter occurs or a carry to the R64CNT counter occurs during read access to the 64-Hz counter.

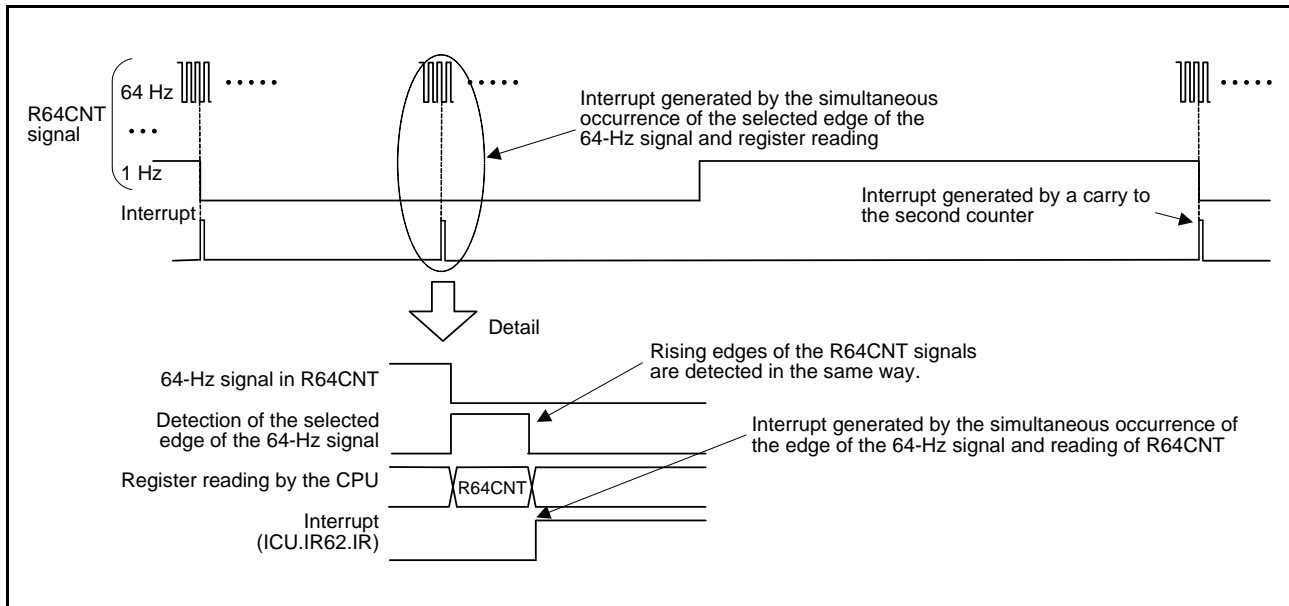


Figure 28.12 Carry Interrupt (CUP) Timing Chart

## 28.5 Usage Notes

### 28.5.1 Register Writing during Counting

The following registers should not be written to during counting (while the RCR2.START bit = 1).

RSECCNT, RMINCNT, RHRCNT, RDAYCNT, RWKCNT, RMONCNT, RYRCNT,  
RCR2.RTCOE, RCR2.HR24, RFRH, RFRL

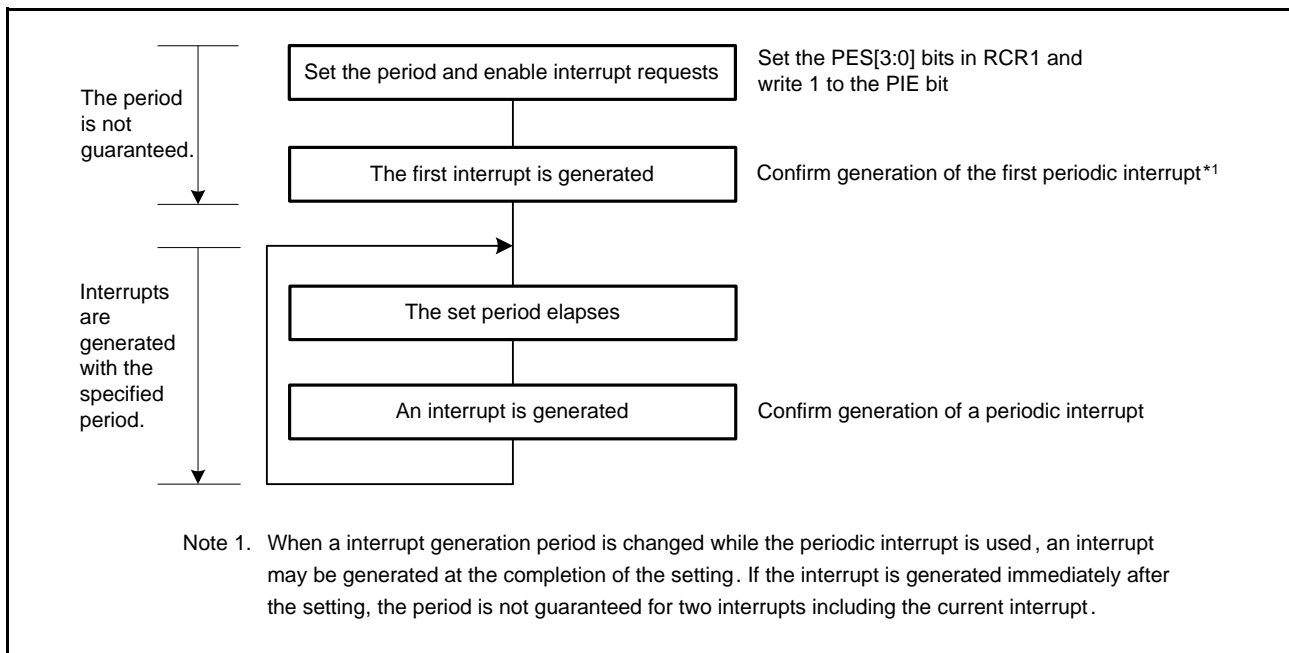
The counter must be stopped before writing to any of the above registers.

### 28.5.2 Use of Periodic Interrupts

The procedure for using periodic interrupts is shown in Figure 28.13.

The generation and period of the periodic interrupt can be changed by the setting of the PES[3:0] bits in RCR1. However, since the prescaler, R64CNT, and RSECCNT are used to generate interrupts, the interrupt period is not guaranteed immediately after setting of the PES[3:0] bits in RCR1.

Furthermore, stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the interrupt period. When the time-error adjustment function is used, the interrupt generation period after adjustment is added or subtracted according to the adjustment value.



**Figure 28.13 Using Periodic Interrupt Function**

### 28.5.3 RTCOUT (1-Hz) Output

Stopping/restarting or resetting counter operation, reset by RTC software, and the 30-second adjustment by changing the RCR2 value affects the period of 1-Hz RTCOUT output. When the time-error adjustment function is used, the period of 1-Hz RTCOUT output after the adjustment is added or subtracted according to the error adjustment value.

### 28.5.4 Transitions to Low Power Consumption Modes after Setting Registers

A transition to a low power consumption state (software standby mode, deep software standby mode, or the battery backup state) during writing to or updating of an RTC register might destroy the register's value. After setting a register, confirm that the setting is in place before initiating a transition to a low power consumption state.

### 28.5.5 Points for Caution When Writing to and Reading from Registers

- When reading a counter register such as the second counter after having written to the counter register, follow the procedure in section 28.3.5, Reading 64-Hz Counter and Time.
- When reading the count registers, alarm registers, year alarm enable register, bits RCR2.AADJE, AADJP, and HR24, registers RCR3 and RCR4, or frequency register after having written to the registers, perform three dummy read operations to ensure that the written value has been reflected in the register.
- When reading the RCR1.CIE and RCR2.RTCOE bits after having written to the register, the written value can be read.
- To read the value from the timer counter after return from a reset, or period in software standby mode, deep software standby mode, or battery backup function, wait for 1/128 second while the clock is operating (the RCR2.START bit = 1).

### 28.5.6 Initialization Procedure when the Realtime Clock is not to be Used

Registers in the RTC are not initialized by a reset. Accordingly, depending on the initial state, the generation of an unintentional interrupt or operation of the counter may lead to increased power consumption.

For products that do not require a realtime clock, initialize the registers by following the initialization procedure shown in Figure 28.14.

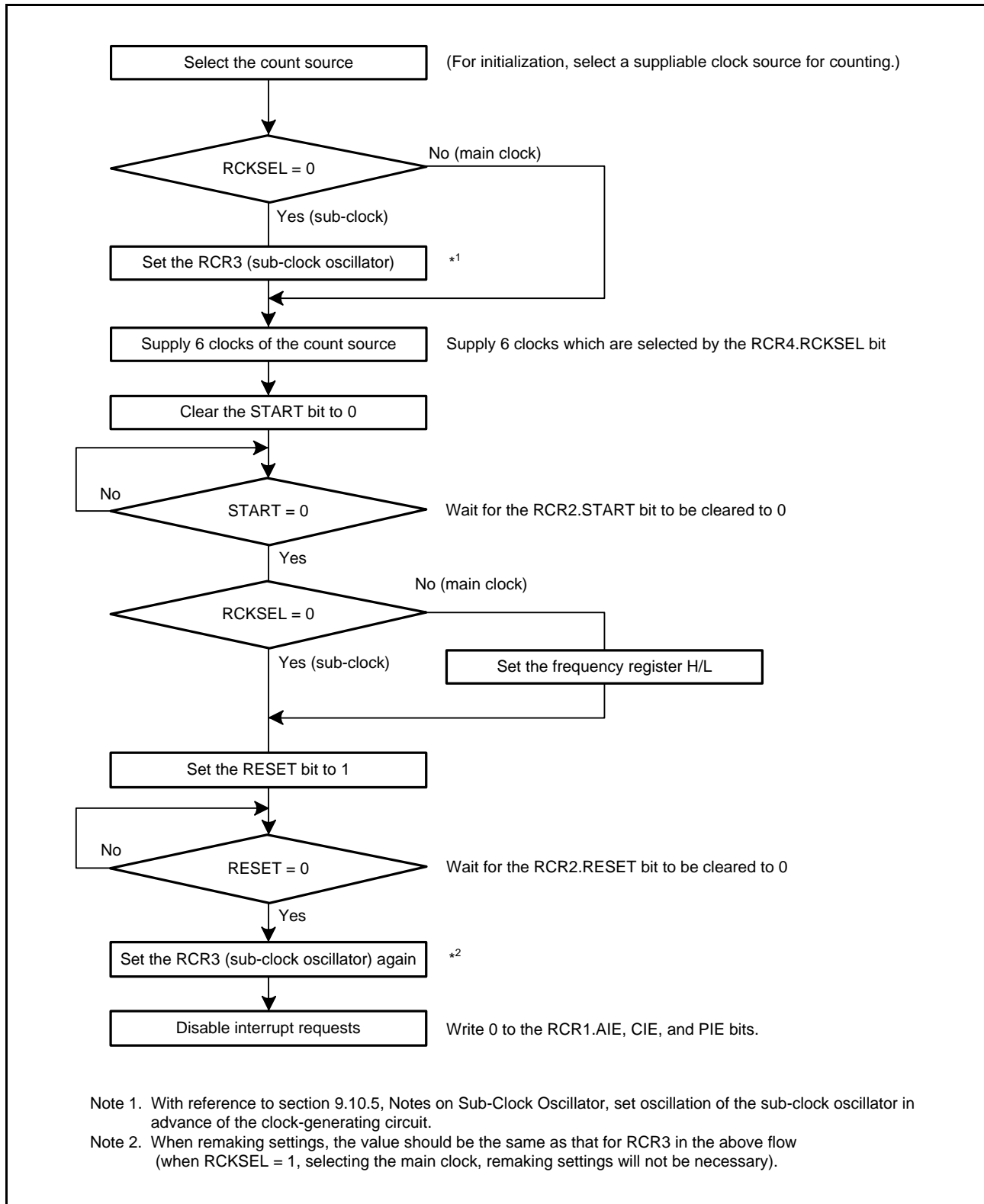


Figure 28.14 Initialization Procedure

## 29. Watchdog Timer (WDTA)

The WDT has a 14-bit down-counter, and can be set up so that the chip is reset by a reset output when counting down from the initial value causes an underflow of the counter. Alternatively, generation of an interrupt request is selectable when the counter underflows. The initial value for counting can be restored to the down-counter by refreshing its value. The interval over which refreshing is possible can also be selected. Refreshing the counter during this interval will restore its initial value for counting, while attempting to refresh the counter beyond this interval leads to the output of a reset or interrupt request. The refresh interval can be adjusted and used to detect the program entering runaway conditions. The WDT stops counting after an underflow or an attempt at refreshing the counter beyond the allowed interval. Counting is restarted by refreshing the counter when the WDT is in register start mode. When the WDT is in auto-start mode, counting is restarted automatically after output of the reset or interrupt request.

### 29.1 Overview

The WDT has two start modes: auto-start mode, in which counting automatically starts after release from the reset state, and register start mode, in which counting is started by refreshing the WDT (writing to the register).

In auto-start mode, necessary settings (clock division ratio, refresh window start and end positions, time-out period, and reset or non-maskable interrupt request output at an underflow) should be made in the option function select register 0 (OFS0) before release from the reset state. In register start mode, necessary settings (clock division ratio, refresh window start and end positions, time-out period, and reset or non-maskable interrupt request output at an underflow) should be made in the respective registers before the counter is started by refreshing after release from the reset state.

Set the WDT start mode select bit (OFS0.WDTSTRT bit) to select auto-start mode or register start mode.

When auto-start mode is selected (OFS0.WDTSTRT = 0), the WDT control register (WDTCR), and WDT reset control register (WDTRCR) settings are disabled and the settings of the OFS0 register are enabled.

When register start mode is selected (OFS0.WDTSTRT = 1), the settings of the OFS0 register are disabled and the WDTCR and WDTRCR settings are enabled.

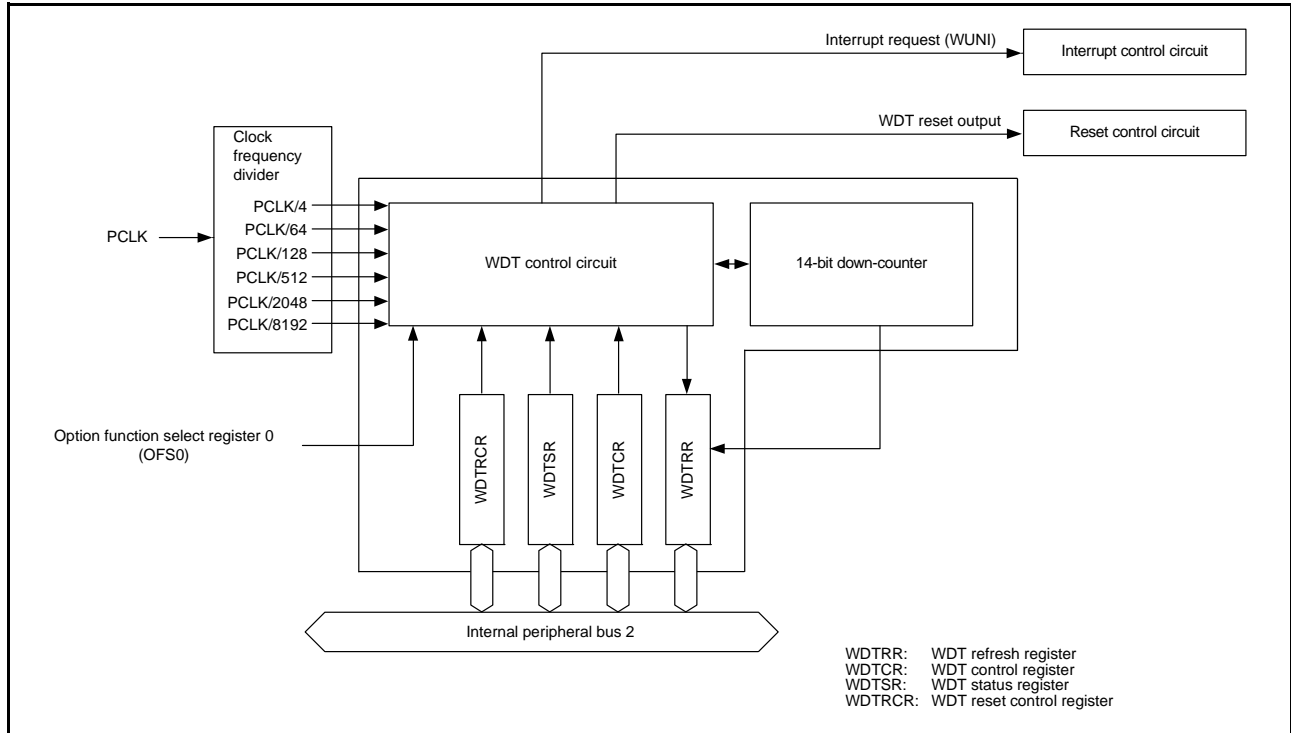
Specifications of the WDT are listed in Table 29.1. Figure 29.1 is a block diagram of the WDT.

**Table 29.1 Specifications of WDT (1/2)**

Item	Specifications
Count source	Peripheral clock (PCLK)
Clock division ratio	Divide by 4, 64, 128, 512, 2,048, or 8,192
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	<ul style="list-style-type: none"> <li>Counting automatically starts after a reset (auto-start mode)</li> <li>Counting is started by refreshing the WDTRR register (writing 00h and then FFh) (register start mode)</li> </ul>
Conditions for stopping the counter	<ul style="list-style-type: none"> <li>Pin reset (the down-counter and registers return to their initial values)</li> <li>A counter underflows or a refresh error is generated Count restarts automatically in auto-start mode, or by refreshing the counter in register start mode.</li> </ul>
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Reset-output sources	<ul style="list-style-type: none"> <li>Down-counter underflows</li> <li>Refreshing outside the refresh-permitted period (refresh error)</li> </ul>
Interrupt request output sources	<ul style="list-style-type: none"> <li>A non-maskable interrupt (WUNI) is generated by an underflow of the down-counter</li> <li>Refreshing outside the refresh-permitted period (refresh error)</li> </ul>
Reading the counter value	The down-counter value can be read by the WDTSR register.
Output signal (internal signal)	<ul style="list-style-type: none"> <li>Reset output</li> <li>Interrupt request output</li> </ul>
Auto-start mode (controlled by the option function select register 0 (OFS0))	<ul style="list-style-type: none"> <li>Selecting the clock frequency division ratio after a reset (OFS0.WDTCKS[3:0] bits)</li> <li>Selecting the time-out period of the watchdog timer (OFS0.WDTPOPS[1:0] bits)</li> <li>Selecting the window start position in the watchdog timer (OFS0.WDTRPSS[1:0] bits)</li> <li>Selecting the window end position in the watchdog timer (OFS0.WDTRPES[1:0] bits)</li> <li>Selecting the reset output or interrupt request output (OFS0.WDTRSTIRQS bit)</li> </ul>

**Table 29.1 Specifications of WDT (2/2)**

Item	Specifications
Register start mode (controlled by the WDT registers)	<ul style="list-style-type: none"> <li>• Selecting the clock frequency division ratio after refreshing (WDTCR.CKS[3:0] bits)</li> <li>• Selecting the time-out period of the watchdog timer (WDTCR.TOPS[1:0] bits)</li> <li>• Selecting the window start position in the watchdog timer (WDTCR.RPSS[1:0] bits)</li> <li>• Selecting the window end position in the watchdog timer (WDTCR.RPES[1:0] bits)</li> <li>• Selecting the reset output or interrupt request output (WDTRCR.RSTIRQS bit)</li> </ul>

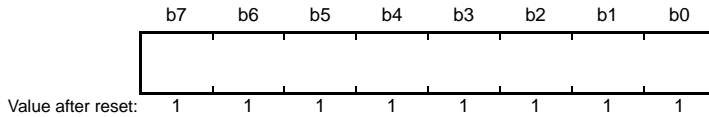


**Figure 29.1 Block Diagram of WDT**

## 29.2 Register Descriptions

### 29.2.1 WDT Refresh Register (WDTRR)

Address(es): 0008 8020h



Bit	Description	R/W
b7 to b0	The down-counter is refreshed by writing 00h and then writing FFh to this register	R/W

WDTRR refreshes the down-counter of the WDT.

The down-counter of the WDT is refreshed by writing 00h and then writing FFh to WDTRR (refresh operation) within the refresh-permitted period.

After the down-counter has been refreshed, it starts counting down from the value selected by setting the WDT time-out period selection (OFS0.WDTPS[1:0]) bits in option function select register 0 in auto-start mode. In register start mode, counting down starts from the value selected by setting the time-out period selection (WDTCR.TOPS[1:0]) bits in the WDT control register by the first refresh operation after release from the reset state.

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 29.3.3, Refresh Operation.

## 29.2.2 WDT Control Register (WDTCR)

Address(es): 0008 8022h

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]				
0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1

Value after reset:

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOPS[1:0]	Time-Out Period Selection	b1 b0 0 0: 1,024 cycles (03FFh) 0 1: 4,096 cycles (0FFFh) 1 0: 8,192 cycles (1FFFh) 1 1: 16,384 cycles (3FFFh)	R/W
b3, b2	—	Reserved	These bits are read as 0 and cannot be modified.	R
b7 to b4	CKS[3:0]	Clock Division Ratio Selection	b7 b4 0 0 0 1: PCLK/4 0 1 0 0: PCLK/64 1 1 1 1: PCLK/128 0 1 1 0: PCLK/512 0 1 1 1: PCLK/2048 1 0 0 0: PCLK/8192 Other settings are prohibited.	R/W
b9, b8	RPES[1:0]	Window End Position Selection	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 and cannot be modified.	R
b13, b12	RPSS[1:0]	Window Start Position Selection	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 and cannot be modified.	R

There are some restrictions on writing to the WDTCR register. For details, refer to section 29.3.2, Control over Writing to the WDTCR and WDTRCR Registers.

In auto-start mode, the settings in the WDTCR register are disabled, and the settings in the option function select register 0 (OFS0) are enabled. The bit setting made to the WDTCR register can also be made in the OFS0 register. For details, refer to section 29.3.8, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

### TOPS[1:0] Bits (Time-Out Period Selection)

The TOPS[1:0] bits select the time-out period (period until the down-counter underflows) from among 1,024, 4,096, 8,192, and 16,384 cycles, taking the divided clock specified by the CKS[3:0] bits as one cycle.

After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of PCLK cycles) until the counter underflows.

Relations between the CKS[3:0] and TOPS[1:0] bit settings, the time-out period, and the number of PCLK cycles are listed in Table 29.2.



**Table 29.2 Time-Out Period Settings**

CKS[3:0] Bits				TOPS[1:0] Bits		Clock Division Ratio	Time-Out Period (Number of Cycles)	Cycles of PCLK Clock
b7	b6	b5	b4	b1	b0			
0	0	0	1	0	0	PCLK/4	1024	4096
				0	1		4096	16384
				1	0		8192	32768
				1	1		16384	65536
0	1	0	0	0	0	PCLK/64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
1	1	1	1	0	0	PCLK/128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
0	1	1	0	0	0	PCLK/512	1024	524288
				0	1		4096	2097152
				1	0		8192	4194304
				1	1		16384	8388608
0	1	1	1	0	0	PCLK/2048	1024	2097152
				0	1		4096	8388608
				1	0		8192	16777216
				1	1		16384	33554432
1	0	0	0	0	0	PCLK/8192	1024	8388608
				0	1		4096	33554432
				1	0		8192	67108864
				1	1		16384	134217728

**CKS[3:0] Bits (Clock Division Ratio Selection)**

These bits select the peripheral clock (PCLK) division ratio from among division by 4, 64, 128, 512, 2,048, and 8,192. Combined with the TOPS[1:0] bit setting, a count period between 4,096 and 134, 217, 728 cycles of the PCLK clock can be selected for the WDT.

**RPES[1:0] Bits (Window End Position Selection)**

These bits select 75%, 50%, 25%, or 0% of the count period for the window end position of the down-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 29.3 lists the counter values for the window start and end positions corresponding to the TOPS[1:0] bit values.

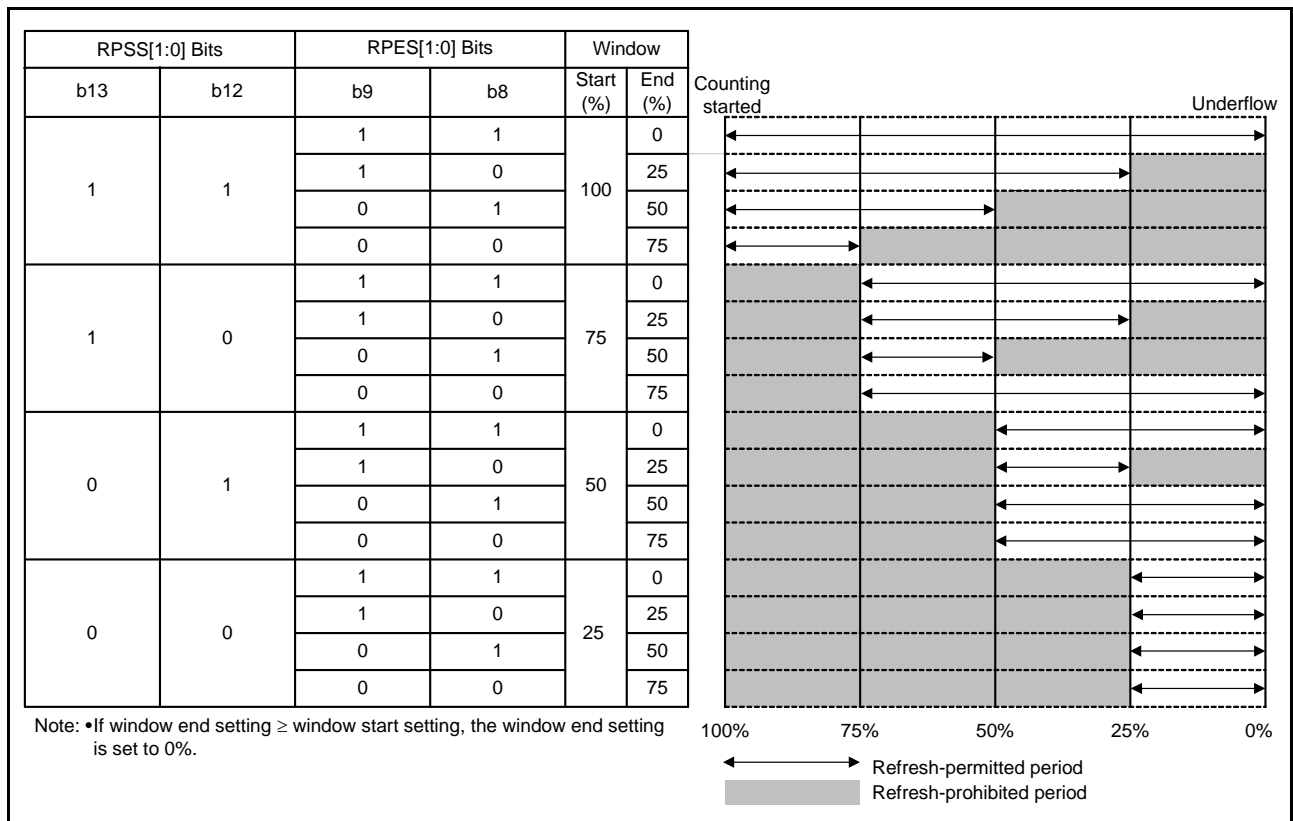
**Table 29.3 Correspondence between Time-Out Period and Window Start and End Counter Values**

TOPS[1:0] Bits		Time-Out Period		Window Start and End Counter Value			
		Cycles	Counter Value	100%	75%	50%	25%
0	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
0	1	4096	0FFFh	0FFFh	0BFFh	07FFh	03FFh
1	0	8192	1FFFh	1FFFh	17FFh	0FFFh	07FFh
1	1	16384	3FFFh	3FFFh	2FFFh	1FFFh	0FFFh

**RPSS[1:0] Bits (Window Start Position Selection)**

These bits select 100%, 75%, 50%, or 25% of the count period for the window start position of the down-counter (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.

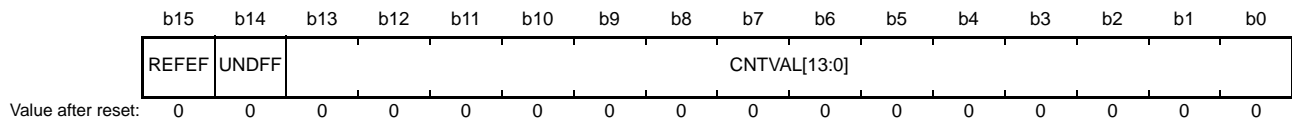
Figure 29.2 shows the relationship between the RPSS[1:0] and RPES[1:0] bit settings and the refresh-permitted and refresh-prohibited periods.



**Figure 29.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period**

### 29.2.3 WDT Status Register (WDTSR)

Address(es): 0008 8024h



Bit	Symbol	Bit Name	Description	R/W
b13 to b0	CNTVAL[13:0]	Down-Counter Value	Value counted by the down-counter	R
b14	UNDFE	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.

WDTSR is initialized by the reset source of the WDT. WDTSR is not initialized by other reset sources.

#### CNTVAL[13:0] Bits (Down-Counter Value)

Read these bits to confirm the value of the down-counter, but note that the read value may differ from the actual count by a value of one count.

#### UNDFE Flag (Underflow Flag)

Read this bit to confirm whether or not an underflow has occurred in the down-counter.

The value 1 indicates that the down-counter has underflowed. The value 0 indicates that the down-counter has not underflowed.

Write 0 to the UNDFE flag to set the value to 0. Writing 1 has no effect.

#### REFEF Flag (Refresh Error Flag)

Read this bit to confirm whether or not a refresh error (performing a refresh operation during a refresh-prohibited period) has occurred.

The value 1 indicates that a refresh error has occurred. The value 0 indicates that no refresh error has occurred.

Clear the REFEF flag by writing 0 to it. Writing 1 has no effect.

### 29.2.4 WDT Reset Control Register (WDTRCR)

Address(es): 0008 8026h

b7	b6	b5	b4	b3	b2	b1	b0
RSTIR QS	—	—	—	—	—	—	—

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 and cannot be modified.	R
b7	RSTIRQS	Reset Interrupt Request Selection	0: Non-maskable interrupt request output is enabled 1: Reset output is enabled	R/W

There are some restrictions on writing to the WDTRCR register. For details, refer to section 29.3.2, Control over Writing to the WDTCR and WDTRCR Registers.

In auto-start mode, the WDTRCR register settings are disabled, and the settings in the option function select register 0 (OFS0) enabled. The bit setting made to the WDTCR register can also be made in the OFS0 register. For details, refer to section 29.3.8, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

### 29.2.5 Option Function Select Register 0 (OFS0)

For details on the OFS0 register, refer to section 29.3.8, Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers.

## 29.3 Operation

### 29.3.1 Count Operation in Each Start Mode

Select the WDT start mode by setting the WDT start mode selection bit (OFS0.WDTSTRT) in the option function select register 0.

When the OFS0.WDTSTRT bit is 1 (register start mode), the WDT control register (WDTCR) and WDT reset control register (WDTRCR) are enabled, and counting is started by refreshing (writing) the WDT refresh register (WDTRR). When the OFS0.WDTSTRT bit is 0 (auto-start mode), the OFS0 register is enabled, and counting automatically starts after reset.

#### 29.3.1.1 Register Start Mode

When the WDT start mode selection (OFS0.WDTSTRT) bit in the option function select register 0 is 1, register start mode is selected, and the WDT control register (WDTCR) and WDT reset control register (WDTRCR) are enabled. After cancelling from the reset, set the clock division ratio, window start and end positions, and time-out period in the WDTCR register, and the reset output or interrupt request output in the WDTRCR register. Then, refresh the down-counter to start counting down from the value selected by setting the time-out period selection (WDTCR.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 WDT does not output the reset signal as long as this continues. However, if the down-counter underflows because the down-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 WDT outputs a reset signal or a non-maskable interrupt request (WUNI). Select reset output or interrupt request output by setting the reset interrupt request selection (WDTRCR.RSTIRQS) bit.

Figure 29.3 shows an example of operation under the following conditions.

- The WDT start mode selection (OFS0.WDTSTRT) bit is 1 (register start mode)
- The reset interrupt request selection (WDTRCR.RSTIRQS) bit is 1 (reset output is enabled)
- The window start position selection (WDTCR.RPSS[1:0]) bits are 10b (75%)
- The window end position selection (WDTCR.RPES[1:0]) bits are 10b (25%)

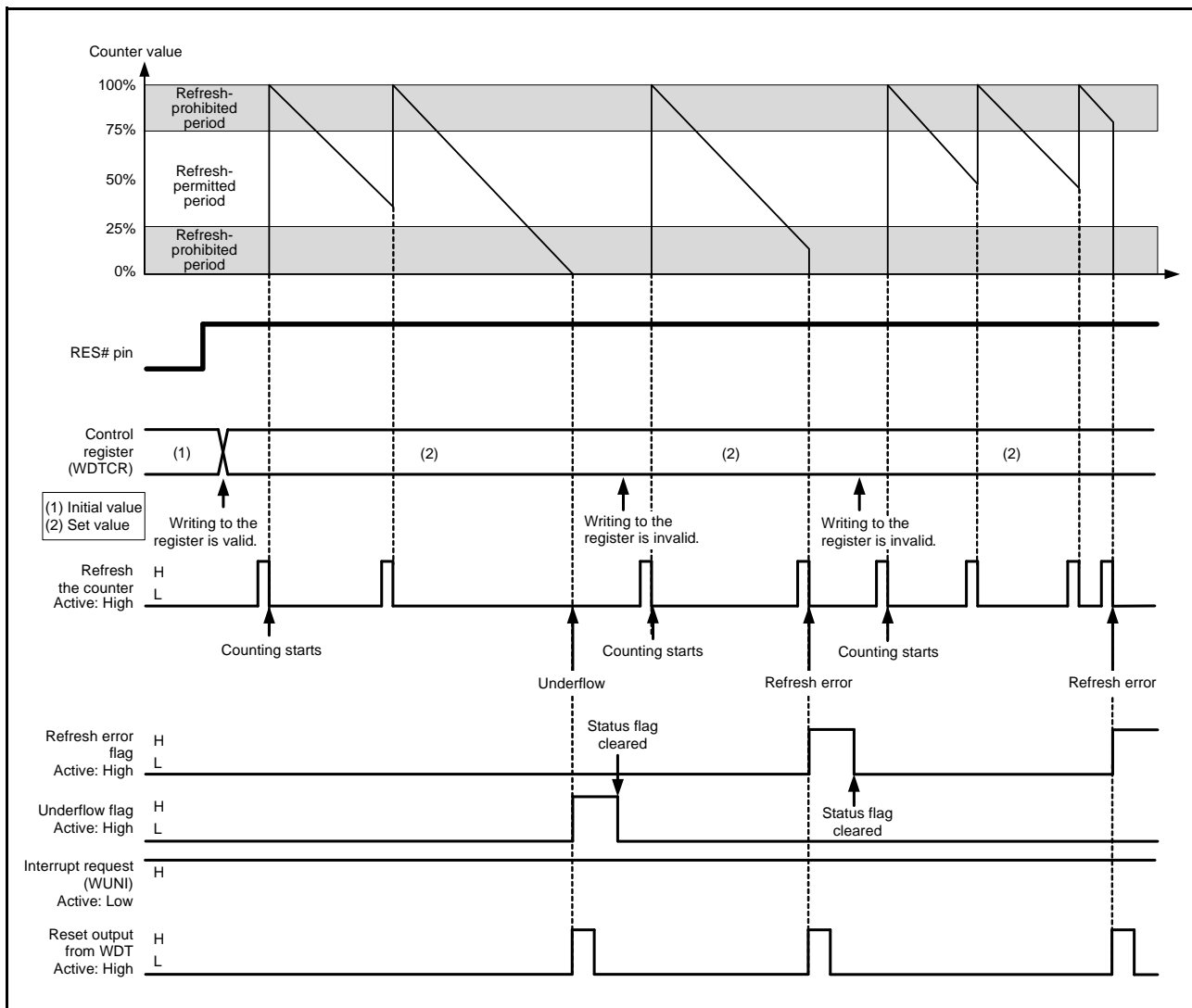


Figure 29.3 Operation Example in Register Start Mode

### 29.3.1.2 Auto-Start Mode

When the WDT start mode selection (OFS0.WDTSTRT) bit in the option function select register 0 is 0, auto-start mode is selected, and the WDT control register (WDTCR) and WDT reset control register (WDTRCR) are disabled.

Within the reset state, the clock division ratio, window start and end positions, time-out period, and reset output or interrupt request output are set by the option function select register (OFS0). When the reset state is canceled, the down-counter automatically starts counting down from the value selected by the WDT time-out period selection (OFS0.WDTPRS[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 when the counter is refreshed and counting down continues. The WDT does not output the reset signal as long as this continues. However, if the down-counter underflows because refreshing of the down-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 WDT outputs the reset signal or non-maskable interrupt request (WUNI). After output of the reset or non-maskable interrupt request for one cycle of counting, the down-counter is reloaded and counting is restarted when the time-out interval elapses. The reset output or interrupt request output can be selected through the WDT reset interrupt request selection (OFS0.WDTRSTIRQS) bit.

Figure 29.4 shows an example of operation under the following conditions.

- The WDT start mode select (OFS0.WDTSTRT) bit is 0 (auto-start mode)
- The reset interrupt request select (OFS0.WDTRSTIRQS) bit is 0 (non-maskable interrupt request output is enabled)
- The window start position select (OFS0.WDTRPSS[1:0]) bits are 10b (75%)
- The window end position select (OFS0.WDTRPES[1:0]) bits are 10b (25%)

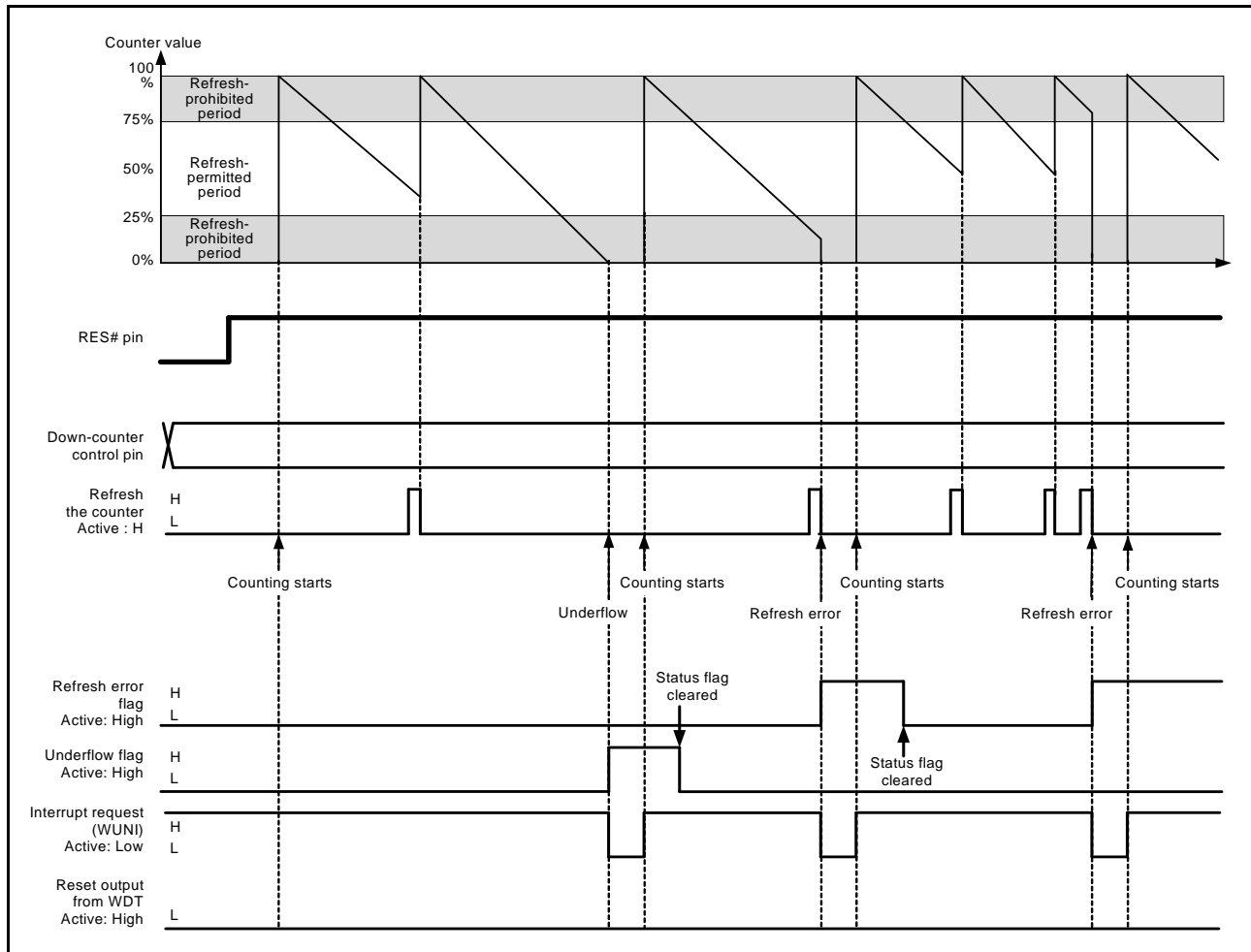


Figure 29.4 Operation Example in Auto-Start Mode

### 29.3.2 Control over Writing to the WDTCR and WDTRCR Registers

Writing to the WDT control register (WDTCR) or WDT reset control register (WDTRCR) is only possible once between the release from the reset state and the first refresh operation.

After a refresh operation (counting starts) or by writing to WDTCR or WDTRCR, the protection signal in the WDT becomes 1 to protect WDTCR and WDTRCR against subsequent attempts at writing.

This protection is released by the reset source of the WDT. With other reset sources, the protection is not released.

Figure 29.5 shows control waveforms produced in response to writing to the WDTCR.

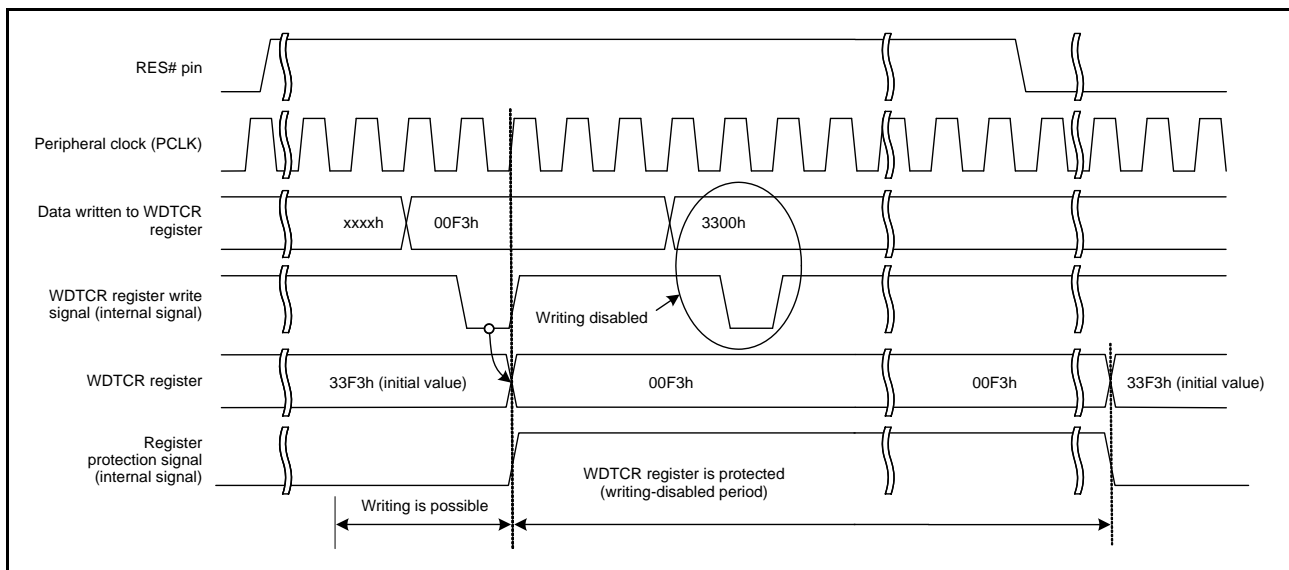


Figure 29.5 Control Waveforms Produced in Response to Writing to the WDTCR

### 29.3.3 Refresh Operation

The down-counter is refreshed and starts operation (counting is started by refreshing) by writing the values 00h and then FFh to the WDT refresh register (WDTRR). If a value other than FFh is written after 00h, the down-counter is not refreshed. After such invalid writing, correct refreshing is performed by again writing 00h and then FFh to the WDT refresh register (WDTRR).

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 (n-th 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 WDTRR is accessed or WDTRR is read between writing 00h and writing FFh to WDTRR, correct refreshing will be done.

[Sample sequences of writing that are valid for refreshing the counter]

- 00h→FFh
- 00h (n-1-th time)→00h (n-th time)→FFh
- 00h→access to another register or read from WDTRR→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 WDTRR outside the refresh-permitted period, if FFh is written to WDTRR in the refresh-permitted period, the writing sequence is valid and refreshing will be done.

After FFh is written to the WDTRR register, refreshing the down-counter requires up to four cycles of the signal for counting (the clock division ratio selection (WDTCR.CKS[3:0]) bits determine how many cycles of the peripheral clock (PCLK) make up one cycle for counting). Therefore, writing FFh to the WDTRR should be completed four-count cycles before the end position of the refresh-permitted period or a counter underflow. The value of the down-counter can be checked by the down-counter value bits (WDTSR.CNTVAL[13:0]).



[Sample refreshing timings]

- When the window start position is set to 1FFFh, even if 00h is written to WDTRR before 1FFFh is reached (2002h, for example), refreshing is done if FFh is written to WDTRR after the value of the WDTSR.CNTVAL[13:0] bits has reached 1FFFh.
- When the window end position is set to 1FFFh, refreshing is done if 2003h (four-count cycles before 1FFFh) or a greater value is read from the WDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to WDTRR.
- 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 WDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to WDTRR, no underflow occurs and refreshing is done.

Figure 29.6 shows the WDT refresh-operation waveforms when the clock division ratio = PCLK/64.

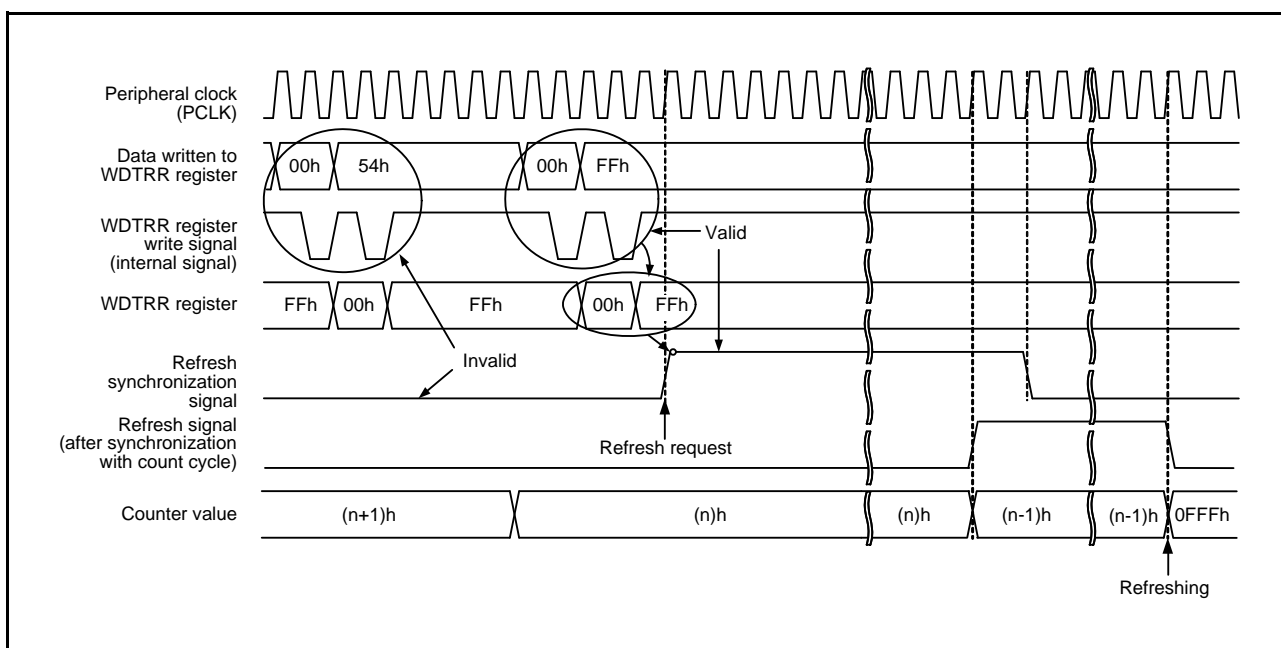


Figure 29.6 WDT Refresh Operation Waveforms (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)

### 29.3.4 Status Flags

The refresh error (WDTSR.REFEF) and underflow (WDTSR.UNDF) flags retain the source of the reset signal output from the WDT or the source of the interrupt request from the WDT.

Thus, after release from the reset state or interrupt request generation, read the WDTSR.REFEF and WDTSR.UNDF 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 WDT, the earlier reset or interrupt source is cleared and the new reset or interrupt source is written.

In addition, several (at least five) cycles of the PCLK, are required to read the reflected value after clearing the flag by writing 0 to the bit.

### 29.3.5 Reset Output

When the reset interrupt selection (WDTRCR.RSTIRQS) bit is set to 1 in register start mode or when the WDT reset interrupt request selection (OFS0.WDTRSTIRQS) bit in the option function select register 0 is set to 1 in auto-start mode, a reset signal is output for one-count cycle when an underflow in the down-counter or a refresh error occurs. In register start mode, the down-counter is initialized (all bits cleared to 0) and kept in that state after assertion of the reset signal. After the reset is cancelled 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.

### 29.3.6 Interrupt Source

When the reset interrupt selection (WDTRCR.RSTIRQS) bit is set to 0 in register start mode or when the WDT reset interrupt request selection (OFS0.WDTRSTIRQS) bit in the option function select register 0 is set to 0 in auto-start mode, a non-maskable interrupt (WUNI) signal is output when an underflow in the down-counter or a refresh error occurs.

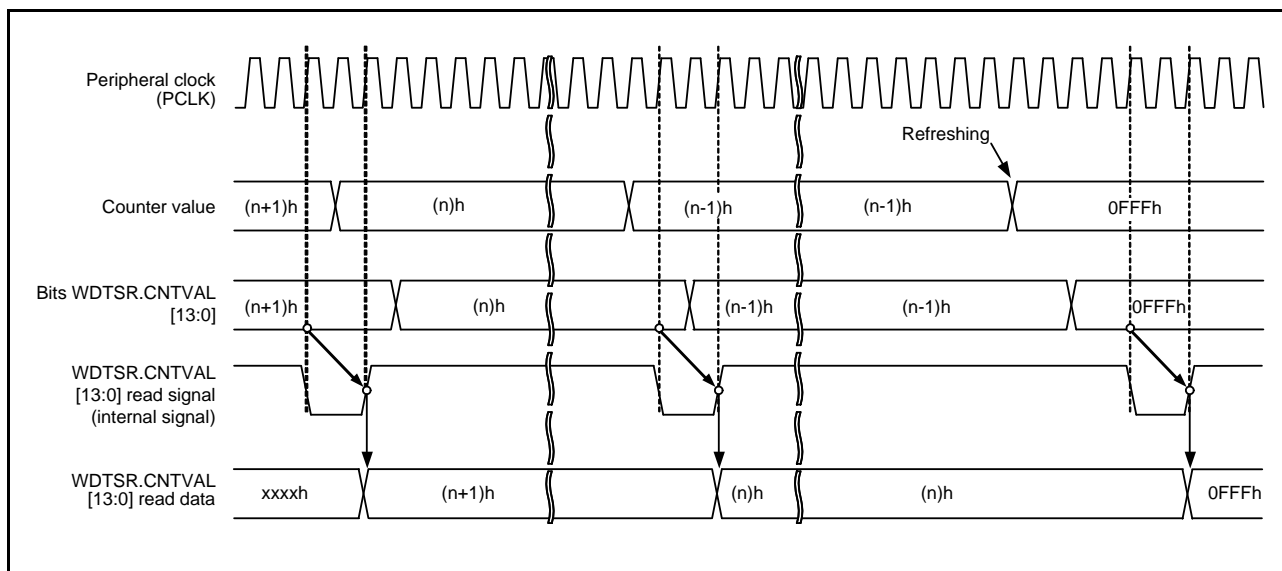
**Table 29.4 WDT Interrupt Source**

Name	Interrupt Source	DTC Activation	DMAC Activation
WUNI	Down-counter underflow Refresh error	Not possible	Not possible

### 29.3.7 Reading the Down-Counter Value

The WDT stores the counter value in the down-counter value (WDTSR.CNTVAL[13:0]) bits of the WDT status register. Thus, the counter value can be checked through the WDTSR.CNTVAL[13:0] bits.

Figure 29.7 shows the processing for reading the WDT down-counter value when the clock division ratio = PCLK/64.



**Figure 29.7 Processing for Reading WDT Down-Counter Value (WDTCR.CKS[3:0] = 0100b, WDTCR.TOPS[1:0] = 01b)**

### 29.3.8 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers

Table 29.5 lists the correspondence between the option function select register 0 (OFS0) and the WDT registers (WDT control register (WDTCR) and WDT reset control register (WDTRCR)) regarding control of the down-counter and reset or interrupt request output. Control can be switched between the OFS0 register and the WDT registers (WDTCR and WDTRCR) through the setting of the WDT start mode selection (OFS0.WDTSTRT) bit.

Note that the OFS0 register setting should be kept unchanged during WDT operation.

For details on the OFS0 register, see section 7.2.1, Option Function Select Register 0 (OFS0).

**Table 29.5 Correspondence between Option Function Select Register 0 (OFS0) and WDT Registers**

Target of Control	Function	OFS0 Register (Effective in Auto-Start Mode) OFS0.WDTSTRT = 0	WDT Registers (Effective in Register Start Mode) OFS0.WDTSTRT = 1
Down-counter	Time-out period selection	OFS0.WDTPS[1:0]	WDTCR.TOPS[1:0]
	Clock division ratio selection	OFS0.WDTCKS[3:0]	WDTCR.CKS[3:0]
	Window start position selection	OFS0.WDTRPSS[1:0]	WDTCR.RPSS[1:0]
	Window end position selection	OFS0.WDTRPES[1:0]	WDTCR.RPES[1:0]
Reset output or interrupt request output	Reset output or interrupt request output selection	OFS0.WDTRSTIRQS	WDTRCR.RSTIRQS

## 30. Independent Watchdog Timer (IWDTa)

The independent watchdog timer (IWDT) is used independently of the conventional watchdog timer to detect programs entering runaway conditions. The IWDT has a 14-bit down-counter, and can be set up so that the chip is reset by a reset output when counting down from the initial value causes an underflow of the counter. Alternatively, generation of an interrupt request is selectable when the counter underflows. The initial value for counting can be restored to the down-counter by refreshing its value. The interval over which refreshing is possible can also be selected. Refreshing the counter during this interval will restore its initial value for counting, while attempting to refresh the counter beyond this interval leads to the output of a reset or interrupt request. The refresh interval can be adjusted and used to detect the program entering runaway conditions. The IWDT stops counting after an underflow or an attempt at refreshing the counter beyond the allowed interval. Counting is restarted by refreshing the counter when the IWDT is in register start mode. When the IWDT is in auto-start mode, counting is restarted automatically after output of the reset or interrupt request.

### 30.1 Overview

The IWDT has two start modes: auto-start mode, in which counting automatically starts after release from the reset state, and register start mode, in which counting is started by refreshing the IWDT (writing to the register).

In auto-start mode, necessary settings (clock division ratio, refresh window start and end positions, time-out period, reset or non-maskable interrupt request output at an underflow, and count stop control in sleep mode) should be made in the option function select register 0 (OFS0) before release from the reset state. In register start mode, necessary settings (clock division ratio, refresh window start and end positions, time-out period, reset or non-maskable interrupt request output at an underflow, and count stop control in sleep mode) should be made in the respective registers before the counter is started by refreshing after release from the reset state.

In register start mode, necessary settings (clock division ratio, refresh window start and end positions, time-out period, reset or non-maskable interrupt request output at an underflow, and count stop control in sleep mode) should be made in the respective registers before the counter is started by refreshing after release from the reset state.

Set the IWDT start mode select bit (OFS0.IWDTSTRT) to select auto-start mode or register start mode.

When auto-start mode is selected (OFS0.IWDTSTRT = 0), the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR) settings are disabled and the OFS0 register settings are enabled.

When register start mode is selected (OFS0.IWDTSTRT = 1), the OFS0 register settings are ignored and the IWDTCR, IWDTRCR, and IWDTCSTPR settings take effect.

Specifications of the IWDT are listed in Table 30.1.

**Table 30.1 Specifications of IWDT (1/2)**

Item	Specifications
Count source*1	IWDT-dedicated clock (IWDTCLK)
Clock division ratio	Division by 1, 16, 32, 64, 128, or 256
Counter operation	Counting down using a 14-bit down-counter
Conditions for starting the counter	<ul style="list-style-type: none"> <li>Counting automatically starts after a reset (auto-start mode)</li> <li>Counting is started by refreshing the IWDTRR register (writing 00h and then FFh) (register start mode)</li> </ul>
Conditions for stopping the counter	<ul style="list-style-type: none"> <li>Pin reset (the down-counter and other registers return to their initial values)</li> <li>A counter underflows or a refresh error is generated Count restarts automatically in auto-start mode, or by refreshing the counter in register start mode</li> </ul>
Window function	Window start and end positions can be specified (refresh-permitted and refresh-prohibited periods)
Reset-output sources	<ul style="list-style-type: none"> <li>Down-counter underflows</li> <li>Refreshing outside the refresh-permitted period (refresh error)</li> </ul>
Interrupt request output sources	<ul style="list-style-type: none"> <li>A non-maskable interrupt (WUNI) is generated by an underflow of the down-counter</li> <li>When refreshing is done outside the refresh-permitted period (refresh error)</li> </ul>

Note 1. Set the count source so that the peripheral module clock frequency  $\geq 4 \times$  (the count source clock divided frequency).

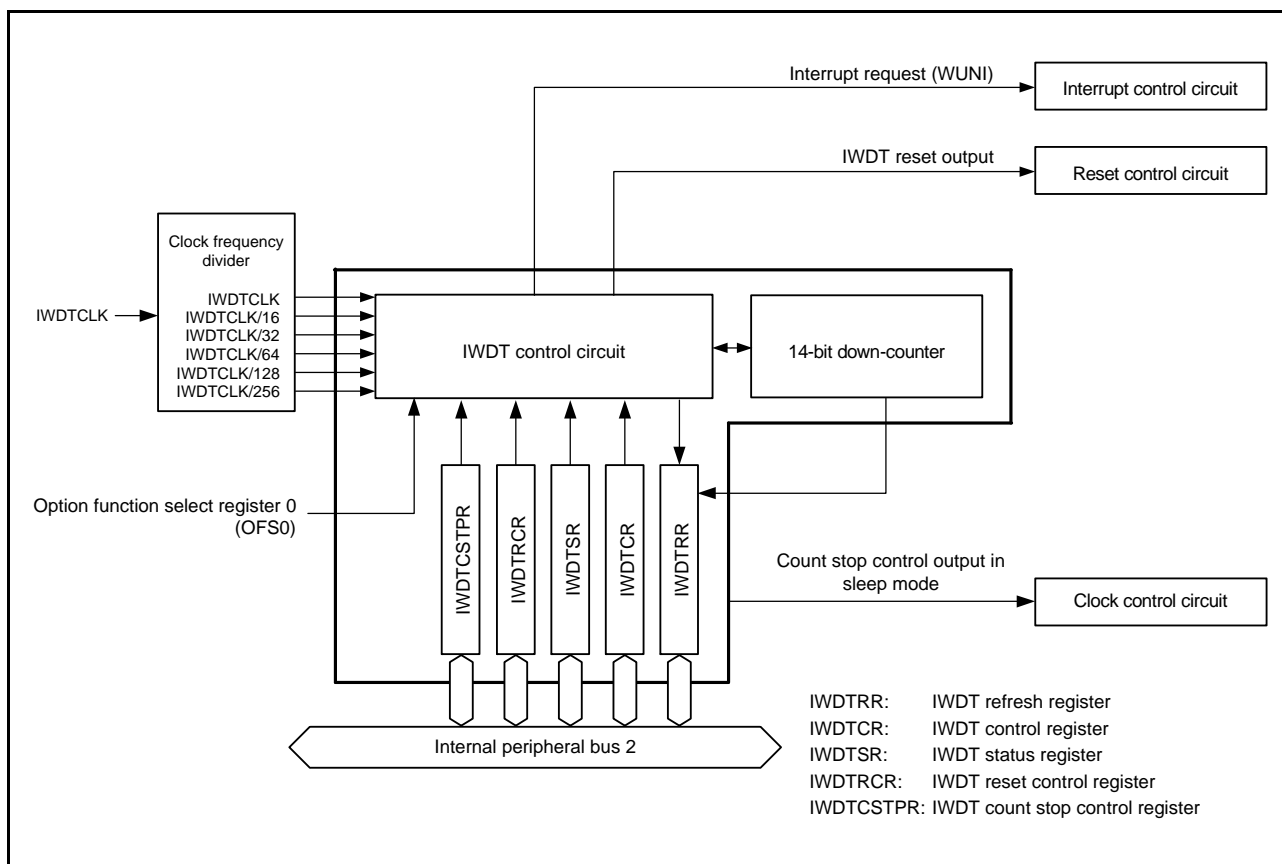
**Table 30.1 Specifications of IWDT (2/2)**

Item	Specifications
Reading the counter value	The down-counter value can be read by the IWDTSR register.
Output signal (internal signal)	<ul style="list-style-type: none"> <li>Reset output</li> <li>Interrupt request output</li> <li>Sleep-mode count stop control output</li> </ul>
Auto-start mode (controlled by the option function select register 0 (OFS0))	<ul style="list-style-type: none"> <li>Selecting the clock frequency division ratio after a reset (OFS0.IWDTCKS[3:0] bits)</li> <li>Selecting the time-out period of the watchdog timer (OFS0.IWDTTOPS[1:0] bits)</li> <li>Selecting the window start position in the watchdog timer (OFS0.IWDRPSS[1:0] bits)</li> <li>Selecting the window end position in the watchdog timer (OFS0.IWDRPES[1:0] bits)</li> <li>Selecting the reset output or interrupt request output (OFS0.IWDRSTIRQS bit)</li> <li>Selecting the down-count stop function at transition to sleep mode, software standby mode, deep software standby mode, or all-module clock stop mode (OFS0.IWDTSLCSTP bit)</li> </ul>
Register start mode (controlled by the IWDT registers)	<ul style="list-style-type: none"> <li>Selecting the clock frequency division ratio after refreshing (IWDTCR.CKS[3:0] bits)</li> <li>Selecting the time-out period of the watchdog timer (IWDTCR.TOPS[1:0] bits)</li> <li>Selecting the window start position in the watchdog timer (IWDTCR.RPSS[1:0] bits)</li> <li>Selecting the window end position in the watchdog timer (IWDTCR.RPES[1:0] bits)</li> <li>Selecting the reset output or interrupt request output (IWDTCCR.RSTIRQS bit)</li> <li>Selecting the down-count stop function at transition to sleep mode, software standby mode, deep software standby mode, or all-module clock stop mode (IWDTCSTPR.SLCSTP bit)</li> </ul>

To use the IWDT, two clocks (peripheral clock (PCLK) and IWDT-dedicated clock (IWDTCLK)) should be supplied so that the IWDT works while the peripheral clock (PCLK) stops. The bus interface and registers operate with PCLK, and the 14-bit down-counter and control circuits operate with IWDTCLK.

Signal lines between the blocks operating with the peripheral clock and IWDT-dedicated are connected through synchronization circuits.

Figure 30.1 is a block diagram of the IWDT

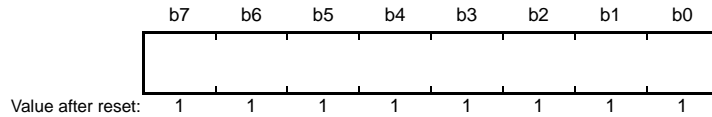


**Figure 30.1 Block Diagram of IWDT**

## 30.2 Register Descriptions

### 30.2.1 IWDt Refresh Register (IWDTRR)

Address(es): 0008 8030h



Bit	Description	R/W
b7 to b0	The down-counter is refreshed by writing 00h and then writing FFh to this register	R/W

IWDTRR refreshes the down-counter of the IWDt.

The down-counter of the IWDt is refreshed by writing 00h and then writing FFh to IWDTRR (refresh operation) within the refresh-permitted period.

After the down-counter has been refreshed, it starts counting down from the value selected by the IWDt time-out period selection (OFS0.IWDTTOPS[1:0]) bits in option function select register 0 in auto-start mode. In register start mode, counting down starts from the value selected by setting the time-out period selection (TOPS[1:0]) bits in the IWDt control register (IWDTCR) in the first refresh operation after release from the reset state.

When 00h is written, the read value is 00h. When a value other than 00h is written, the read value FFh.

For details of the refresh operation, refer to section 30.3.3, Refresh Operation.

### 30.2.2 IWDT Control Register (IWDTCR)

Address(es): 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]				
0	0	1	1	0	0	1	1	1	1	1	1	0	0	1	1

Value after reset:

Bit	Symbol	Bit Name	Description	R/W
b1, b0	TOPS[1:0]	Time-Out Period Selection	b1 b0 0 0: 1,024 cycles (03FFh) 0 1: 4,096 cycles (0FFFh) 1 0: 8,192 cycles (1FFFh) 1 1: 16,384 cycles (3FFFh)	R/W
b3, b2	—	Reserved	These bits are read as 0 and cannot be modified.	R
b7 to b4	CKS[3:0]	Clock Division Ratio Selection	b7 b4 0 0 0 0: IWDTCLK 0 0 1 0: IWDTCLK/16 0 0 1 1: IWDTCLK/32 0 1 0 0: IWDTCLK/64 1 1 1 1: IWDTCLK/128 0 1 0 1: IWDTCLK/256 Other settings are prohibited.	R/W
b9, b8	RPES[1:0]	Window End Position Selection	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 and cannot be modified.	R
b13, b12	RPSS[1:0]	Window Start Position Selection	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 and cannot be modified.	R

There are some restrictions on writing to the IWDTCR register. For details, refer to section 30.3.2, Control over Writing to the IWDTCR, IWDTSCR, and IWDTCSR Registers.

In auto-start mode, the settings in the IWDTCR register are disabled, and the settings in the 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 30.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

#### TOPS[1:0] Bits (Time-Out Period Selection)

The TOPS[1:0] bits select the time-out period (period until the down-counter underflows) from among 1,024, 4,096, 8,192, or 16,384 cycles, taking the divided clock specified by the CKS[3:0] bits as one cycle.

After the down-counter is refreshed, the combination of the CKS[3:0] and TOPS[1:0] bits determines the time (number of IWDTCLK cycles) until the counter underflows.

Relations between the CKS[3:0] and TOPS[1:0] bit settings, the time-out period, and the number of IWDTCLK cycles are listed in Table 30.2.

**Table 30.2 Settings and Time-Out Periods**

CKS[3:0] Bits				TOPS[1:0] Bits		Clock Division Ratio	Time-Out Period (Number of Cycles)	Cycles of IWDTCLK
b7	b6	b5	b4	b1	b0			
0	0	0	0	0	0	IWDTCLK	1024	1024
				0	1		4096	4096
				1	0		8192	8192
				1	1		16384	16384
0	0	1	0	0	0	IWDTCLK/16	1024	16384
				0	1		4096	65536
				1	0		8192	131072
				1	1		16384	262144
0	0	1	1	0	0	IWDTCLK/32	1024	32768
				0	1		4096	131072
				1	0		8192	262144
				1	1		16384	524288
0	1	0	0	0	0	IWDTCLK/64	1024	65536
				0	1		4096	262144
				1	0		8192	524288
				1	1		16384	1048576
1	1	1	1	0	0	IWDTCLK/128	1024	131072
				0	1		4096	524288
				1	0		8192	1048576
				1	1		16384	2097152
0	1	0	1	0	0	IWDTCLK/256	1024	262144
				0	1		4096	1048576
				1	0		8192	2097152
				1	1		16384	4194304

**CKS[3:0] Bits (Clock Division Ratio Selection)**

These bits select the IWDTCLK clock division ratio from among division by 1, 16, 32, 64, 128, and 256. Combined with the TOPS[1:0] bit setting, a count period between 1,024 and 4,194, 304 cycles of the IWDTCLK clock can be selected for the IWDT.



**RPES[1:0] Bits (Window End Position Selection)**

These bits select 75%, 50%, 25% or 0% of the count period for the window end position of the down-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 30.3 lists the counter values for the window start and end positions corresponding to TOPS[1:0] bit values.

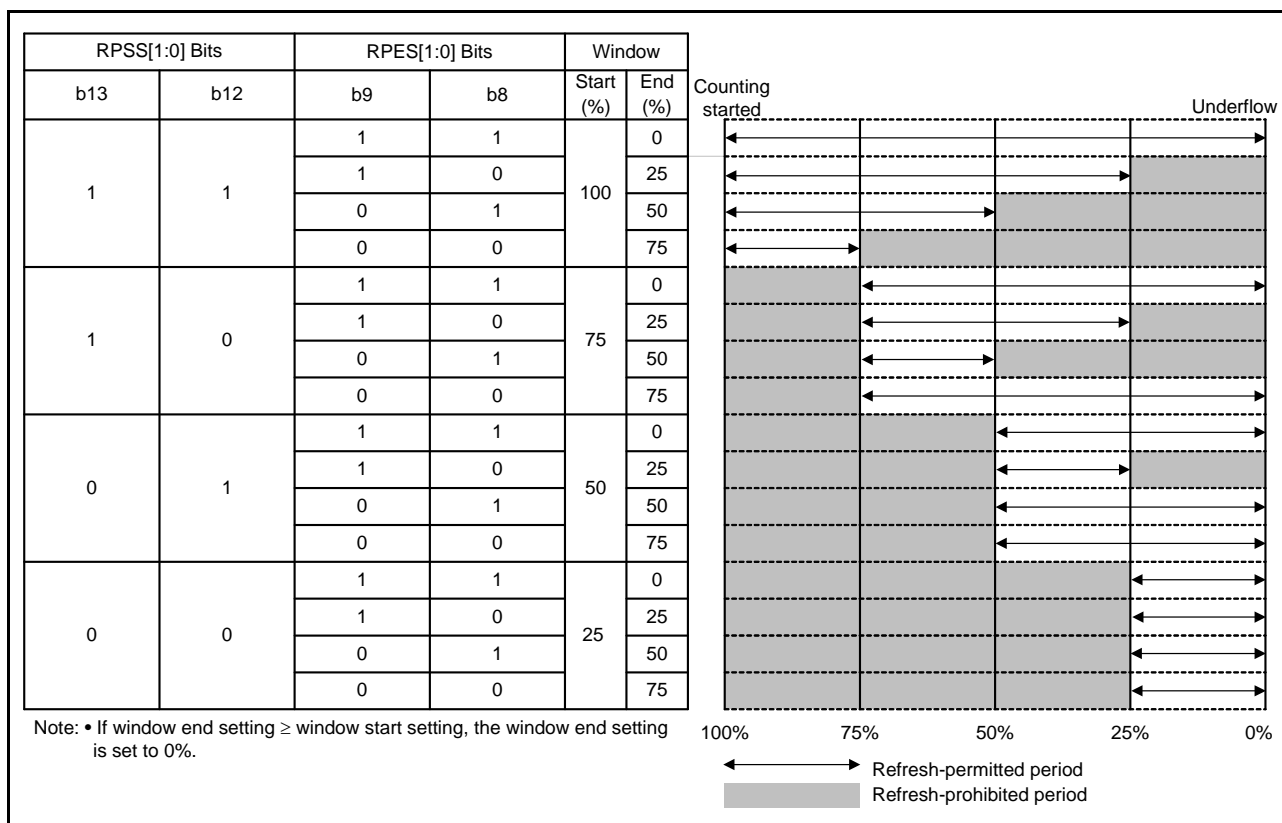
**Table 30.3 Relationship between Time-Out Period and Window Start and End Counter Values**

TOPS[1:0] Bits		Time-Out Period		Window Start and End Counter Value			
b1	b0	Cycles	Counter Value	100%	75%	50%	25%
0	0	1024	03FFh	03FFh	02FFh	01FFh	00FFh
0	1	4096	0FFFh	0FFFh	0BFFh	07FFh	03FFh
1	0	8192	1FFFh	1FFFh	17FFh	0FFFh	07FFh
1	1	16384	3FFFh	3FFFh	2FFFh	1FFFh	0FFFh

**RPSS[1:0] Bits (Window Start Position Selection)**

These bits select 100%, 75%, 50%, or 25% of the count period for the window start position for the down-counter (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.

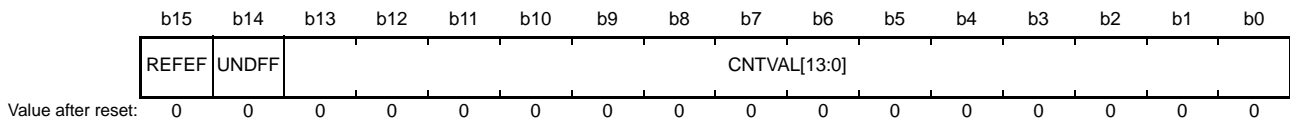
Figure 30.2 shows the relationship between the RPSS[1:0] and RPES[1:0] bit setting and the refresh-permitted and refresh-prohibited periods.



**Figure 30.2 RPSS[1:0] and RPES[1:0] Bit Settings and the Refresh-Permitted Period**

### 30.2.3 IWDT Status Register (IWDTSR)

Address(es): 0008 8034h



Bit	Symbol	Bit Name	Description	R/W
b13 to b0	CNTVAL[13:0]	Down-Counter Value	Value counted by the down-counter	R
b14	UNDFE	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.

IWDTSR is initialized by the reset source of the IWDT. IWDTSR is not initialized by other reset sources.

#### CNTVAL[13:0] Bits (Down-Counter Value)

Read these bits to confirm the value of the down-counter, but note that the read value may differ from the actual count by a value of one count.

#### UNDFE Flag (Underflow Flag)

Read this bit to confirm whether or not an underflow has occurred in the down-counter.

The value 1 indicates that the down-counter has underflowed. The value 0 indicates that the down-counter has not underflowed.

Write 0 to the UNDFE flag to set the value to 0. Writing 1 has no effect.

#### REFEF Flag (Refresh Error Flag)

Read this bit 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.

### 30.2.4 IWDTRCR Reset Control Register (IWDTRCR)

Address(es): 0008 8036h

b7	b6	b5	b4	b3	b2	b1	b0
RSTIR QS	—	—	—	—	—	—	—

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 and cannot be modified.	R
b7	RSTIRQS	Reset Interrupt Request Selection	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 30.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the IWDTRCR register settings are disabled, and the settings in the option function select register 0 (OFS0) enabled. The bit setting mode to the IWDTRCR register can also be made in the OFS0 register. For details, refer to section 30.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDTR Registers.

### 30.2.5 IWDTCSTPR IWDTCSTPR Register (IWDTCSTPR)

Address(es): 0008 8038h

b7	b6	b5	b4	b3	b2	b1	b0
SLCST P	—	—	—	—	—	—	—

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 and cannot be modified.	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, deep software standby mode, or all-module clock stop mode	R/W

There are some restrictions on writing to the IWDTCSTPR register. For details, refer to section 30.3.2, Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSTPR Registers.

In auto-start mode, the settings in the IWDTCSTPR register are ignored, and the settings in the option function select register 0 (OFS0) take effect. The bit setting mode to the IWDTCSTPR register can also be made in the OFS0 register. For details, refer to section 30.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDTR Registers.

#### SLCSTP Bit (Sleep-Mode Count Stop Control)

This bit selects whether to stop counting at a transition to sleep mode, software standby mode, deep software standby mode, or all-module clock stop mode.

### 30.2.6 Option Function Select Register 0 (OFS0)

For the OFS0 register, refer to section 30.3.8, Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers.

## 30.3 Operation

### 30.3.1 Count Operation in Each Start Mode

Select the IWDT start mode by setting the IWDT start mode selection bit (OFS0.IWDTSTRT) in the option function select register 0.

When the OFS0.IWDTSTRT bit is 1 (register start mode), the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR) are enabled, and counting is started by refreshing (writing) the IWDT refresh register (IWDTRR). When the OFS0.IWDTSTRT bit is 0 (auto-start mode), the setting of the OFS0 register is enabled, and counting automatically starts after reset.

#### 30.3.1.1 Register Start Mode

When the IWDT start mode selection (OFS0.IWDTSTRT) bit in the option function select register 0 is 1, register start mode is selected, and the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDTCSTPR) are enabled.

After cancelling from the reset, set the clock division ratio, window start and end positions, and time-out period in the IWDTCR register, the reset output or interrupt request output in the IWDTRCR register, and the down-counter stop control at transitions to low-power-consumption modes in the IWDTCSTPR register. Then, refresh the down-counter to start counting down from the value selected by setting the time-out period selection (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 down-counter underflows because the down-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). Select reset output or interrupt request output by setting the reset interrupt request selection (IWDTRCR.RSTIRQS) bit.

Figure 30.3 shows an example of operation under the following conditions.

- The IWDT start mode selection (OFS0.IWDTSTRT) bit is 1 (register start mode)
- The reset interrupt request selection (IWDTRCR.RSTIRQS) bit is 1 (reset output is enabled)
- The window start position selection (IWDTCR.RPSS[1:0]) bits are 10b (75%)
- The window end position selection (IWDTCR.RPES[1:0]) bits are 10b (25%)

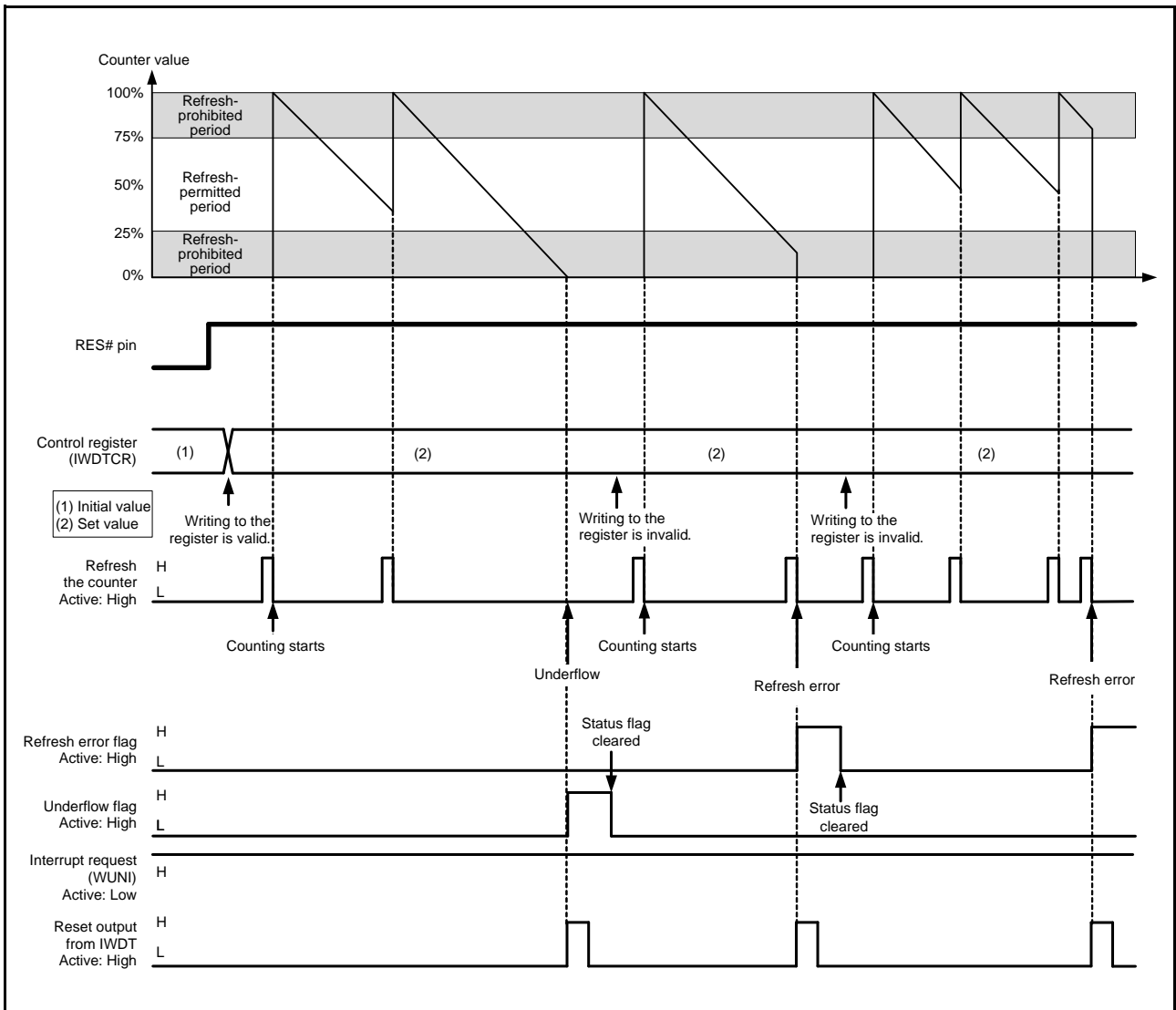


Figure 30.3 Operation Example in Register Start Mode

### 30.3.1.2 Auto-Start Mode

When the IWDT start mode selection (OFS0.IWDTSTRT) bit in the option function select register 0 is 0, auto-start mode is selected, and the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCCR), and IWDT count stop control register (IWDTCTPR) are disabled.

Within the reset state, the clock division ratio, window start and end positions, time-out period, reset output or interrupt request output, and down-counter stop control at transitions to low-power-consumption modes should be specified in the OFS0 register. When the reset state is canceled, the down-counter automatically starts counting down from the value selected by the IWDT time-out period selection (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 when the counter is refreshed and counting down continues. The IWDT does not output the reset signal as long as this continues. However, if the down-counter underflows because refreshing of the down-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 output of the reset or non-maskable interrupt request for one cycle of counting, the down-counter is reloaded and counting is restarted when the time-out interval elapses. The reset output or interrupt request output can be selected through the IWDT reset interrupt request selection (OFS0.IWDRSTIRQS) bit.

Figure 30.4 shows an example of operation under the following conditions.

- The IWDT start mode selection (OFS0.IWDTSTRT) bit is 0 (auto-start mode)
- The reset interrupt request selection (OFS0.IWDRSTIRQS) bit is 0 (non-maskable interrupt request output is enabled)
- The window start position selection (OFS0.IWDTRPSS[1:0]) bits are 10b (75%)
- The window end position selection (OFS0.IWDTRPES[1:0]) bits are 10b (25%)

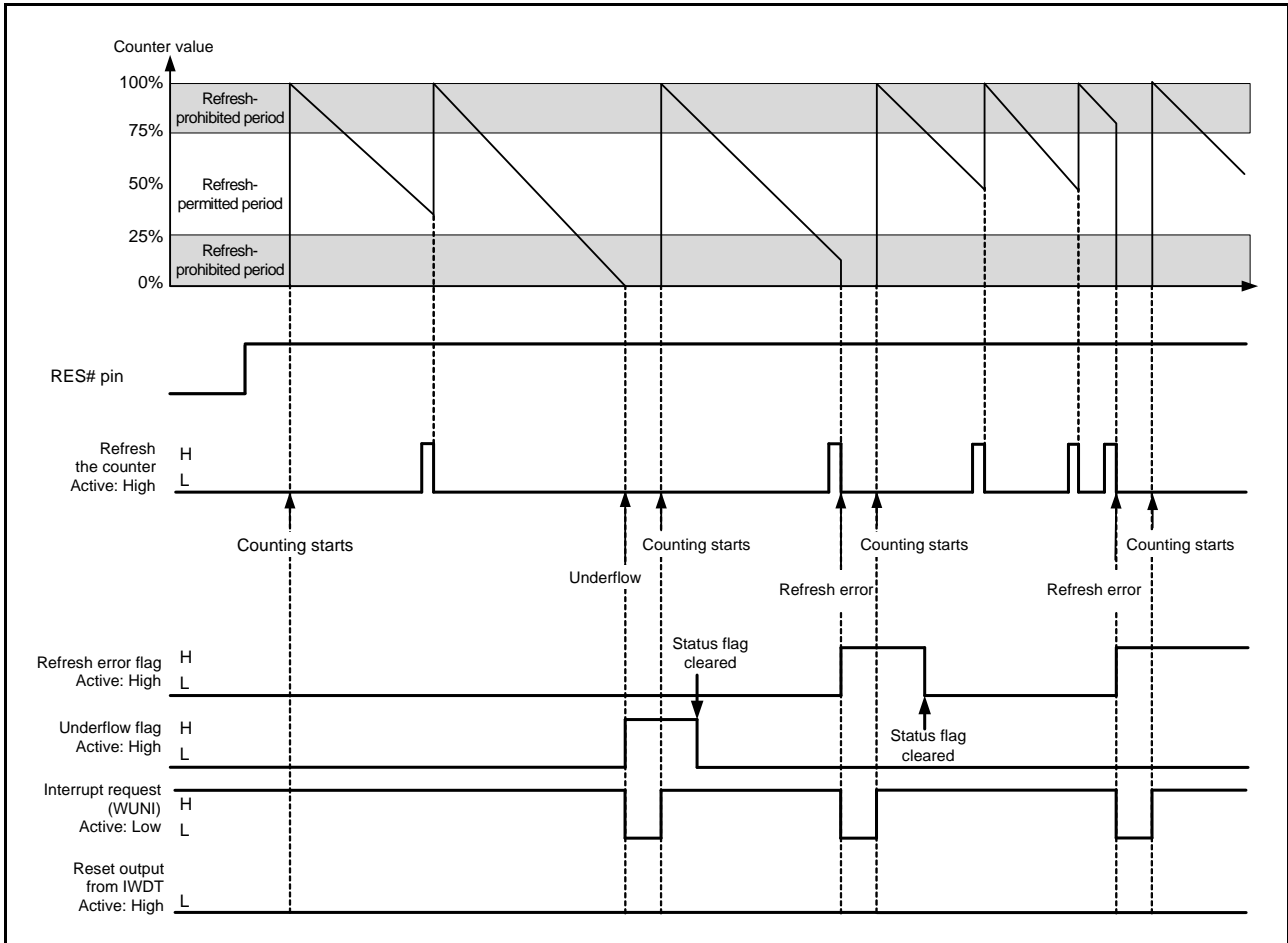


Figure 30.4 Operation Example in Auto-Start Mode

### 30.3.2 Control over Writing to the IWDTCR, IWDTRCR, and IWDTCSSTPR Registers

Writing to the IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), or IWDT count stop control register (IWDTCSSTPR) is only possible once between the release from the reset state and the first refresh operation. After a refresh operation (counting starts) or IWDTCR, IWDTRCR, or IWDTCSSTPR is written to, the protection signal in the IWDT becomes 1 to protect IWDTCR, IWDTRCR, and IWDTCSSTPR 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 30.5 shows control waveforms produced in response to writing to the IWDTCR.

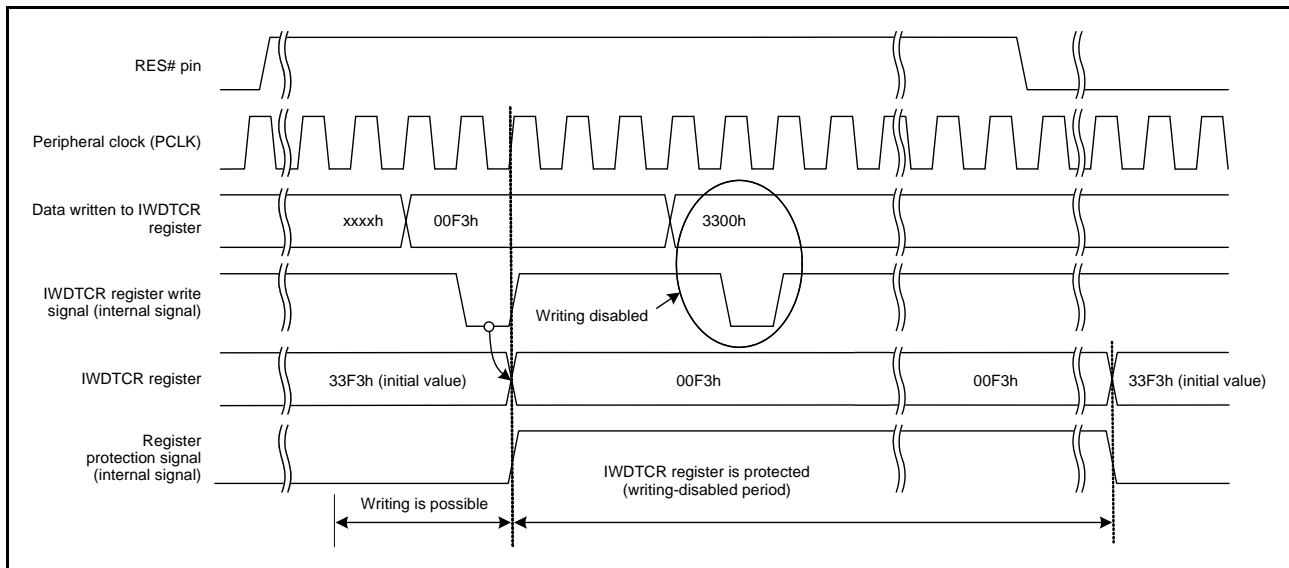


Figure 30.5 Control Waveforms Produced in Response to Writing to the IWDTCR

### 30.3.3 Refresh Operation

The down-counter is refreshed and starts operation (counting is started by refreshing) by writing the values 00h and then FFh to the IWDT refresh register (IWDTRR). If a value other than FFh is written after 00h, the down-counter is not refreshed. After such invalid writing, correct refreshing is performed by again writing 00h and then FFh to the IWDT refresh register (IWDTRR).

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 (n-th 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 IWDTRR is accessed or IWDTRR is read between writing 00h and writing FFh to IWDTRR, correct refreshing will be done.

[Sample sequences of writing that are valid for refreshing the counter]

- 00h→FFh
- 00h (n-1-th time) →00h (n-th time) →FFh
- 00h→access to another register or read from IWDTRR→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 IWDTRR outside the refresh-permitted period, if FFh is written to IWDTRR 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 down-counter requires up to four cycles of the signal for counting (the clock division ratio selection (the clock division ratio selection (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 should be completed four-count cycles before the end position of the refresh-permitted period or a counter underflow. The value of the down-counter can be checked by the counter bits (IWDTSR.CNTVAL[13:0]).

[Sample refreshing timings]

- When the window start position is set to 1FFFh, even if 00h is written to IWDTRR before 1FFFh is reached (2002h, for example), refreshing is done if FFh is written to IWDTRR after the value of the IWDTSR.CNTVAL[13:0] bits has reached 1FFFh.
- When the window end position is set to 1FFFh, refreshing is done if 2003h (four-count cycles before 1FFFh) or a greater value is read from the IWDTSR.CNTVAL[13:0] bits immediately after writing 00h → FFh to IWDTRR.
- 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 IWDTRR, no underflow occurs and refreshing is done.

Figure 30.6 shows the IWDT refresh-operation waveforms when  $PCLK > IWDTCCLK$  and clock division ratio =  $IWDTCCLK$ , and Figure 30.7 shows those when  $PCLK < IWDTCCLK$  and clock division ratio =  $IWDTCCLK/16$ .

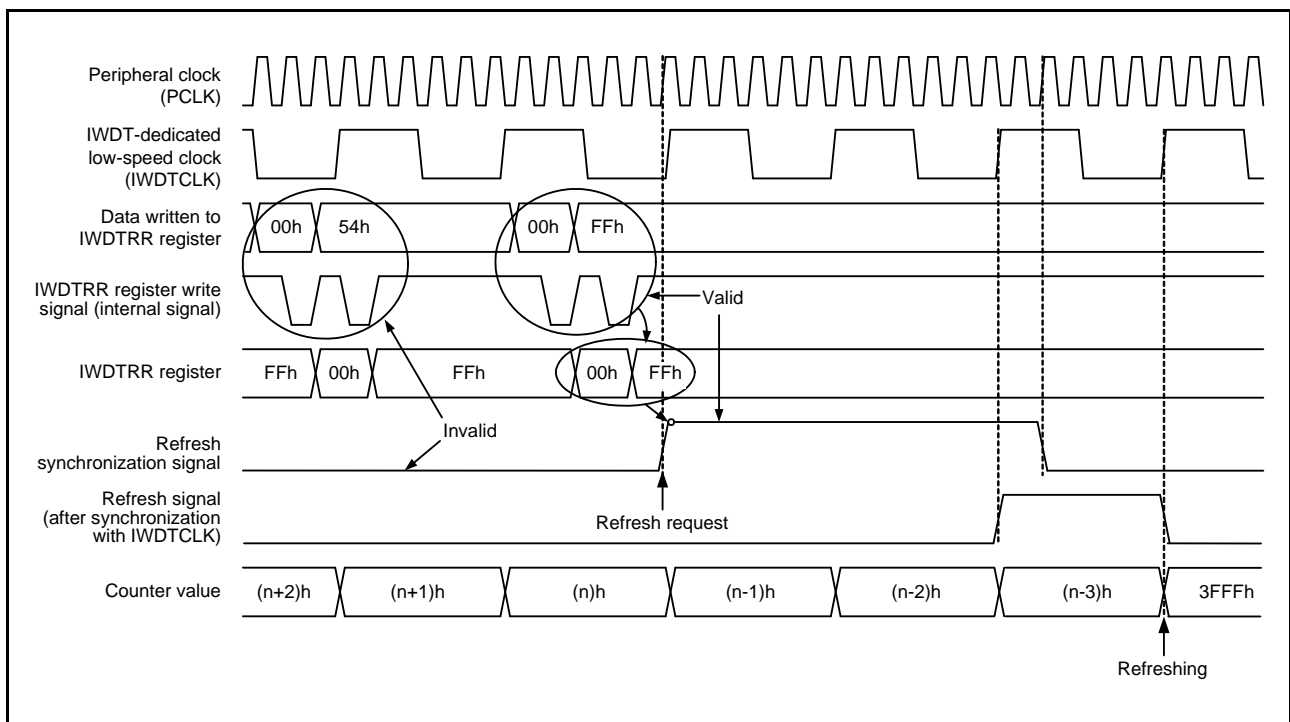


Figure 30.6 IWDT Refresh Operation Waveforms (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)



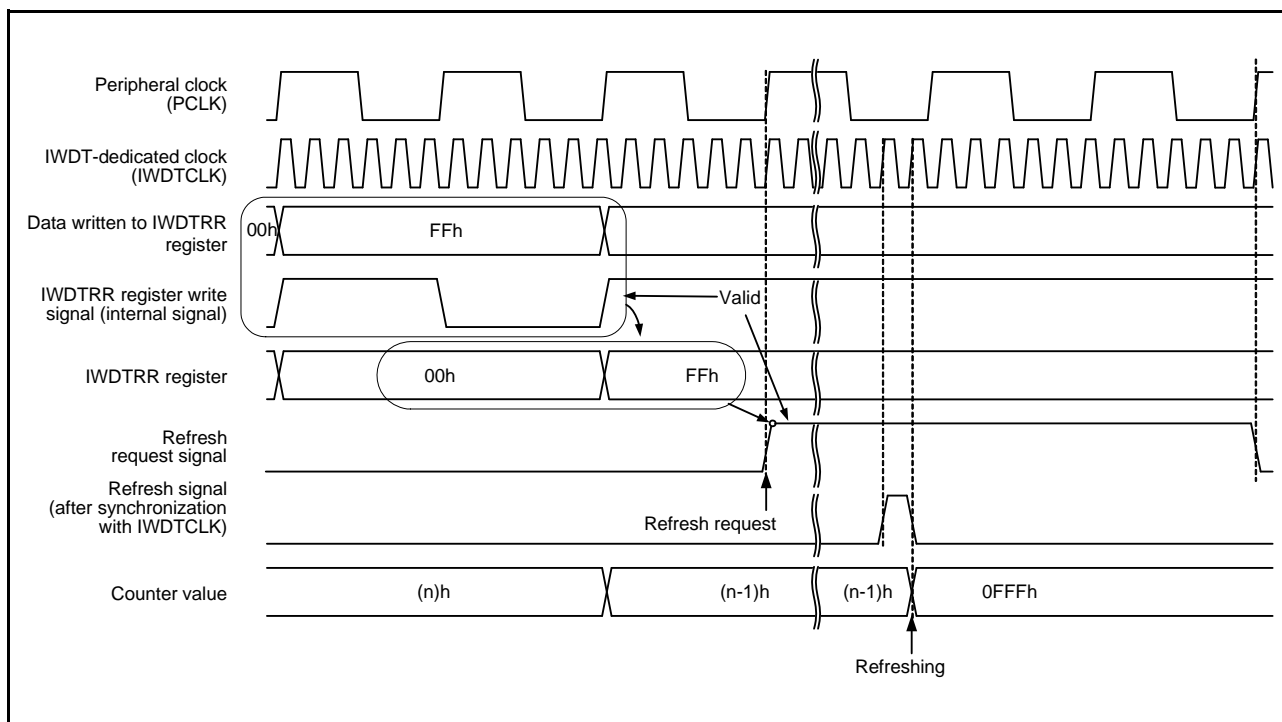


Figure 30.7 IWDT Refresh Operation Waveforms (IWDTCR.CKS[3:0] = 0010b, IWDTCR.TOPS[1:0] = 01b)

### 30.3.4 Status Flags

The refresh error (IWDTSR.REFEF) and underflow (IWDTSR.UNDF) 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.UNDF 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.

In addition, several cycles of the IWDTCLK (at least three) and of the PCLK (at least two) are required to read the reflected value after clearing the flag by writing 0 to the bit.

### 30.3.5 Reset Output

When the reset interrupt selection (IWDTSCR.RSTIRQS) bit is set to 1 in register start mode or when the IWDT reset interrupt request selection (OFS0.IWDRSTIRQS) bit in the option function select register 0 is set to 1 in auto-start mode, a reset signal is output for one-count cycle when an underflow in the down-counter or a refresh error occurs.

In register start mode, the down-counter is initialized (all bits cleared to 0) and kept in that state after assertion of the reset signal. After the reset is cancelled 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.

### 30.3.6 Interrupt Source

When the reset interrupt selection (IWDTRCR.RSTIRQS) bit is set to 0 in register start mode or when the IWDT reset interrupt request selection (OFS0.IWDTRSTIRQS) bit in the option function select register 0 is set to 0 in auto-start mode, a non-maskable interrupt (WUNI) signal is output when an underflow in the down-counter or a refresh error occurs.

**Table 30.4 IWDT Interrupt Source**

Name	Interrupt Source	DTC Activation	DMAC Activation
WUNI	Down-counter underflow Refresh error	Not possible	Not possible

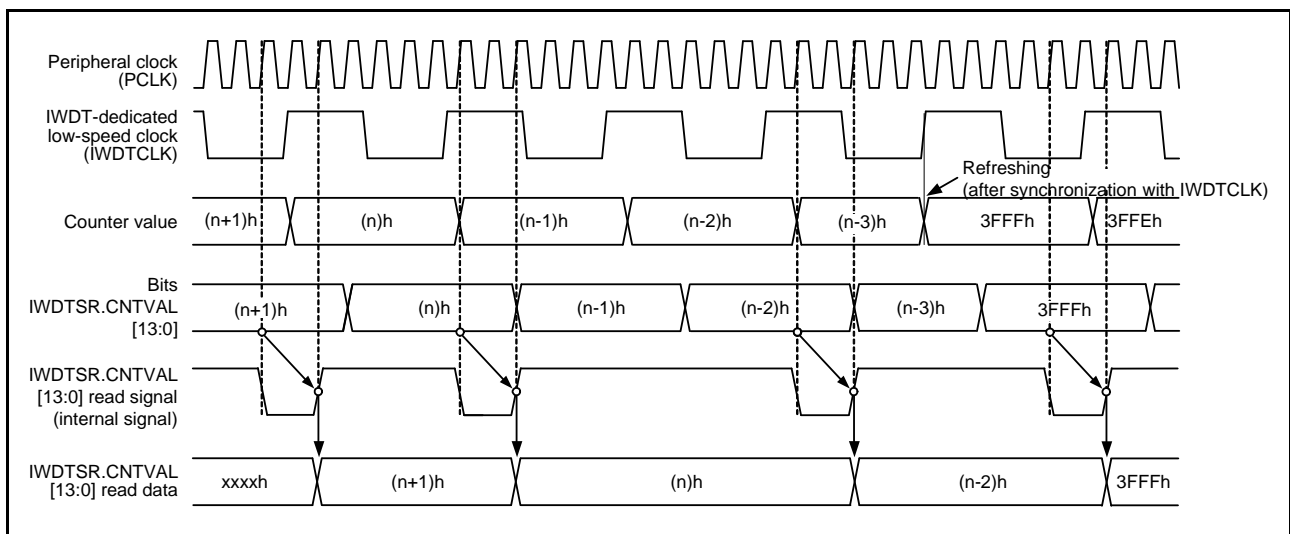
### 30.3.7 Reading the Down-Counter Value

As the down-counter in IWDT-dedicated clock (IWDTCLK), the counter value cannot be read directly. The IWDT synchronizes the counter value with the peripheral clock (PCLK) and stores it in the down-counter value (IWDTSR.CNTVAL[13:0]) bits of the IWDT status register.

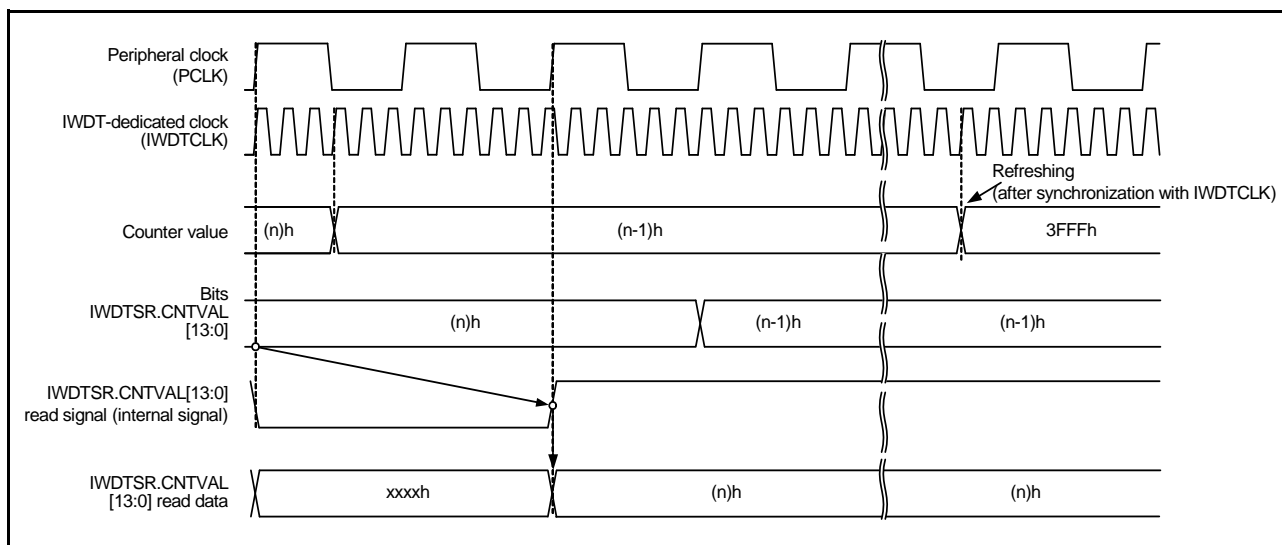
Thus, the counter value can be checked indirectly through the IWDTSR.CNTVAL[13:0] bits.

Reading the down-counter value requires multiple PCLK clock cycles (up to four clock cycles), and the read counter value may differ from the actual down-counter value by a value of one count.

Figure 30.8 shows the processing for reading the IWDT down-counter value when  $PCLK > IWDTCLK$  and clock division ratio =  $IWDTCLK$ , and Figure 30.9 shows the processing when  $PCLK < IWDTCLK$  and clock division ratio =  $IWDTCLK/16$ .



**Figure 30.8 Processing for Reading IWDT Down-Counter Value (IWDTCR.CKS[3:0] = 0000b, IWDTCR.TOPS[1:0] = 11b)**



**Figure 30.9 Processing for Reading IWDT Down-Counter Value (IWDTCR.CKS[3:0] = 0010b, IWDTCR.TOPS[1:0] = 11b)**

### 30.3.8 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers

Table 30.5 lists the correspondence between the option function select register 0 (OFS0) and the IWDT registers (IWDT control register (IWDTCR), IWDT reset control register (IWDTRCR), and IWDT count stop control register (IWDCSTPR)) regarding control of the down-counter, reset or interrupt request output, and count stop function. Control can be switched between the option function select register 0 (OFS0) and the IWDT registers (IWDTCR, IWDTRCR, and IWDCSTPR) through the setting of the IWDT start mode selection (OFS0.IWDTSTRT) bit in the option function select register 0 (OFS0).

Note that the option function select register 0 (OFS0) setting should be kept unchanged during IWDT operation. For details on the option function select register 0 (OFS0), see section 7.2.1, Option Function Select Register 0 (OFS0).

**Table 30.5 Correspondence between Option Function Select Register 0 (OFS0) and IWDT Registers**

Target of Control	Function	OFS0 Register (Effective in Auto-Start Mode) OFS0.IWDTSTRT = 0	IWDT Registers (Effective in Register Start Mode) OFS0.IWDTSTRT = 1
Down-counter	Time-out period selection	OFS0.IWDTTOPS[1:0]	IWDTCR.TOPS[1:0]
	Clock division ratio selection	OFS0.IWDTCKS[3:0]	IWDTCR.CKS[3:0]
	Window start position selection	OFS0.IWDRPSS[1:0]	IWDTCR.RPSS[1:0]
	Window end position selection	OFS0.IWDRPES[1:0]	IWDTCR.RPES[1:0]
Reset output or interrupt request output	Reset output or interrupt request output selection	OFS0.IWDRSTIRQS	IWDTRCR.RSTIRQS
Count stop	Sleep-mode count stop selection	OFS0.IWDTSLCSTP	IWDCSTPR.SLCSTP

## 30.4 Usage Notes

### 30.4.1 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.

## 31. USB 2.0 Function Module (USBa)

### 31.1 Overview

The RX630 Group provides one port of USB 2.0 function module (USB).

The USB is a USB controller which provides capabilities as a USB function controller. The USB supports full-speed transfer when used as the function controller. The USB has a USB transceiver and supports all of the transfer types defined by the USB Specifications.

The USB has buffer memory for data transfer, providing a maximum of ten pipes. Any endpoint numbers can be assigned to PIPE1 to PIPE9, based on the peripheral devices or user system for communication.

Table 31.1 lists the specifications of the USB.

**Table 31.1 Specifications of USB**

Item	Specifications
Features	<ul style="list-style-type: none"> <li>• USB Device Controller (UDC) and transceiver for USB 2.0 are incorporated.</li> <li>• One port is provided.</li> <li>• Self-power mode or bus-power mode can be selected.</li> <li>• Programmable intervals for isochronous and interrupt transfers</li> <li>• Full-speed transfer (12 Mbps) is supported</li> <li>• Control transfer stage control function</li> <li>• Device state control function</li> <li>• Auto response function for SET_ADDRESS request</li> <li>• SOF interpolation function</li> </ul>
Communication data transfer type	<ul style="list-style-type: none"> <li>• Control transfer</li> <li>• Bulk transfer</li> <li>• Interrupt transfer</li> <li>• Isochronous transfer</li> </ul>
Internal bus interface	<ul style="list-style-type: none"> <li>• Connected to internal peripheral bus 2</li> </ul>
Pipe configuration	<ul style="list-style-type: none"> <li>• Buffer memory for USB communications is provided.</li> <li>• Up to ten pipes can be selected (including the default control pipe).</li> <li>• Endpoint numbers can be assigned flexibly to PIPE1 to PIPE9.</li> <li>• Transfer conditions that can be set for each pipe:</li> <li>• PIPE0: Control transfer only (default control pipe: DCP) Buffer size: 8, 16, 32, or 64 bytes (single buffer)</li> <li>• PIPE1 and PIPE2: Bulk transfer or isochronous transfer Buffer size: 8, 16, 32, or 64 bytes for bulk transfer or 1 to 256 bytes for isochronous transfer (double buffer can be specified)</li> <li>• PIPE3 to PIPE5: Bulk transfer only Buffer size: 8, 16, 32, or 64 bytes (double buffer can be specified)</li> <li>• PIPE6 to PIPE9: Interrupt transfer only Buffer size: 1 to 64 bytes (single buffer)</li> </ul>
Others	<ul style="list-style-type: none"> <li>• Reception ending function using transaction count</li> <li>• Function that changes the BRDY interrupt event notification timing (BFRE)</li> <li>• Function that automatically clears the buffer memory after the data for the pipe specified at the DnFIFO (n = 0 or 1) port has been read (DCLRM)</li> <li>• NAK setting function for response PID generated by end of transfer (SHTNAK)</li> </ul>
Power consumption reducing function	<ul style="list-style-type: none"> <li>• Module-stop state can be set.</li> </ul>

Figure 31.1 shows a block diagram of the USB module.

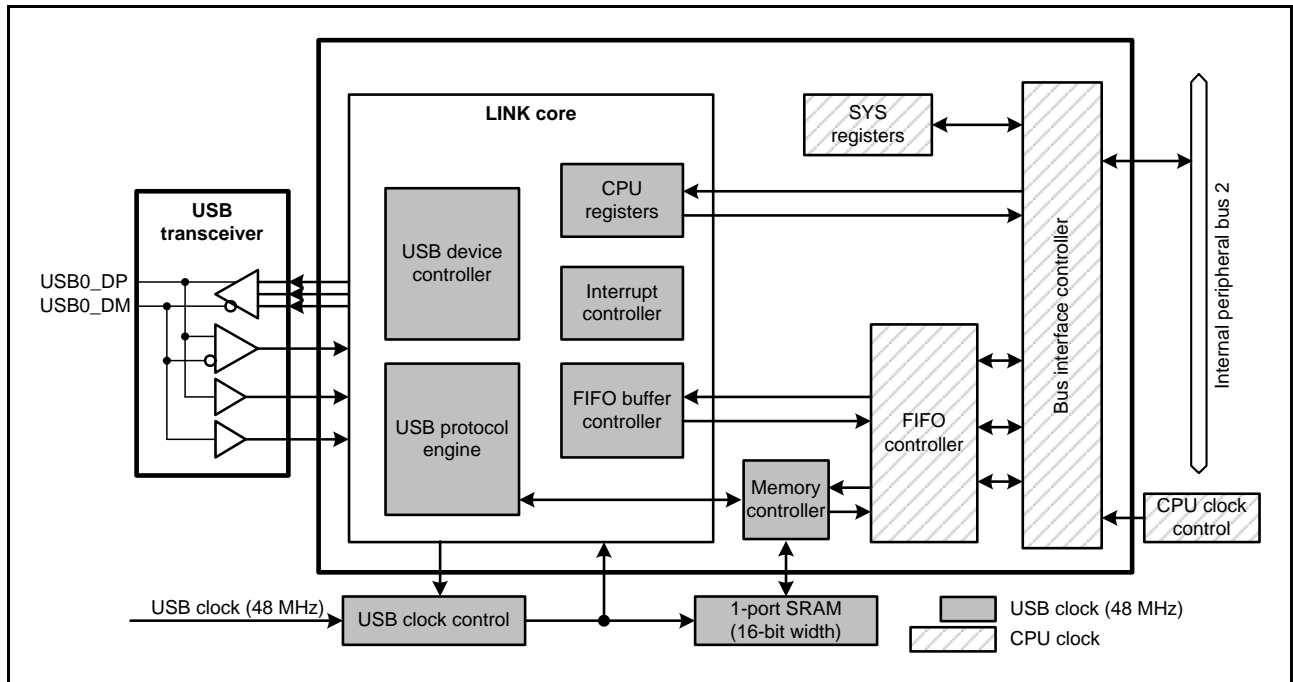


Figure 31.1 Block Diagram of USB

Table 31.2 lists the input/output pins of the USB.

Table 31.2 USB Pin Configuration

Pin Name	I/O	Function
USB0_DP	I/O	D+ I/O pin of the USB on-chip transceiver This pin should be connected to the D+ pin of the USB bus.
USB0_DM	I/O	D- I/O pin of the USB on-chip transceiver This pin should be connected to the D- pin of the USB bus.
USB0_VBUS	Input	USB cable connection monitor pin This pin should be connected to VBUS of the USB bus. Whether VBUS is connected or disconnected can be detected.
USB0_DPUPE	Output	1.5-kΩ pull-up resistor control signal for USB D+ signal
VCC_USB	Input	USB power supply pin
VSS_USB	Input	USB ground pin

## 31.2 Register Descriptions

### 31.2.1 System Configuration Control Register (SYSCFG)

Address(es): USB0.SYSCFG 000A 0000h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	SCKE	—	—	—	—	—	DPRPU	—	—	—	USBE
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	USBE	USB Module Operation Enable	0: USB module operation is disabled. 1: USB module operation is enabled.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	DPRPU	D+ Line Resistor Control	0: Pulling up the line is disabled. 1: Pulling up the line is enabled.	R/W
b9 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b10	SCKE	USB Module Clock Enable	0: Stops supplying the clock signal to the USB module. 1: Enables supplying the clock signal to the USB module.	R/W
b15 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

#### USBE Bit (USB Module Operation Enable)

The USBE bit enables or disables operation of the USB module.

Modifying the USBE bit from 1 to 0 initializes some register bits as listed in Table 31.3.

This bit should be modified while the SCKE bit is 1.

#### DPRPU Bit (D+ Line Resistor Control)

The DPRPU bit enables or disables pulling up the D+ line.

When the DPRPU bit is set to 1, the USB module asserts the USB0\_DPUPE pin to notify the USB host of connection.

Modifying the DPRPU bit from 1 to 0 allows the USB module to negate the USB0\_DPUPE pin, thus notifying the USB host of disconnection.

#### SCKE Bit (USB Module Clock Enable)

The SCKE bit stops or enables supplying 48-MHz clock signals to the USB module.

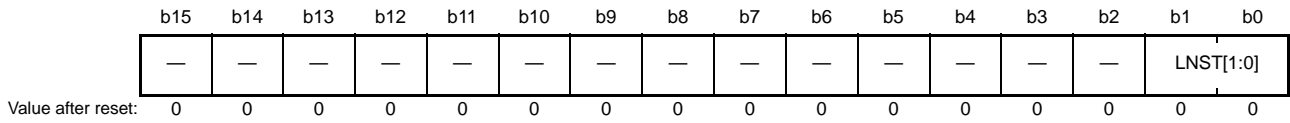
When this bit is 0, only SYSCFG, DMA0PCFG, and DMA1PCFG can be read from and written to; the other registers in the USB module cannot be read from or written to.

**Table 31.3 Registers Initialized by Writing SYSCFG.USBE = 0**

Register	Bits
SYSSTS0	LNST[1:0]
DVSTCTR0	RHST[2:0]
INTSTS0	DVSQ[2:0]
USBADDR	USBADDR[6:0]
USBREQ	BREQUEST[7:0], BMREQUESTTYPE[7:0]
USBVAL	WVALUE[15:0]
USBINDX	WINDEX[15:0]
USBLENG	WLENGTH[15:0]

### 31.2.2 System Configuration Status Register 0 (SYSSTS0)

Address(es): USB0.SYSSTS0 000A 0004h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	LNST[1:0]	USB Data Line Status Monitor	b1 b0 0 0: SE0 0 1: J-state 1 0: K-state 1 1: SE1	R
b15 to b2	—	Reserved	These bits are read as 0 and cannot be modified.	R

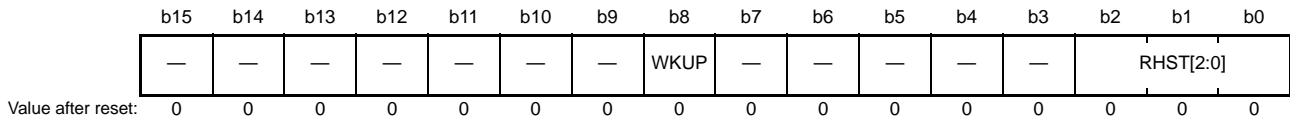
#### LNST[1:0] Bits (USB Data Line Status Monitor)

The LNST[1:0] bits indicate the status of the USB data bus lines (D+ and D-).

The LNST[1:0] bits should be read after the connection processing (SYSCFG.DPRPU = 1 is set).

### 31.2.3 Device State Control Register 0 (DVSTCTR0)

Address(es): USB0.DVSTCTR0 000A 0008h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	RHST[2:0]	USB Bus Reset Status	b2 b1 b0 0 0 0: Communication speed not determined 1 0 0: USB bus reset in progress 0 1 0: Full-speed connection	R
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	WKUP	Wakeup Output	0: Remote wakeup signal is not output. 1: Remote wakeup signal is output.	R/W*1
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Only 1 can be written.

#### RHST[2:0] Bits (USB Bus Reset Status)

The RHST[2:0] bits indicate the status of the USB bus reset.

The USB module fixes the value of the RHST[2:0] bits when software writes 0 to the USBRST bit and the USB module completes SE0 driving.

A DVST interrupt is generated as soon as the USB module detects the USB bus reset and then the RHST[2:0] bits are fixed to 010b.

#### WKUP Bit (Wakeup Output)

The WKUP bit enables or disables outputting the remote wakeup signal (resume signal) to the USB bus.

The USB module controls the output time of a remote wakeup signal. When this bit is set to 1, the USB module clears this bit to 0 after outputting the 10-ms K-state.

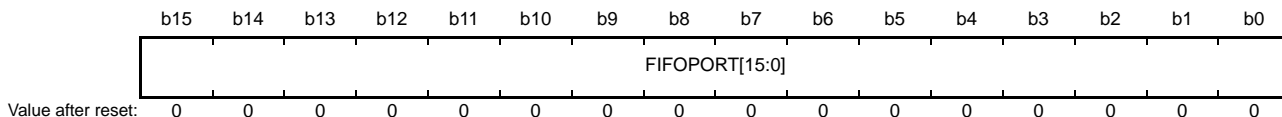
According to the USB Specifications, the USB bus idle state must be kept for 5 ms or longer before a remote wakeup signal is sent. If the USB module writes 1 to this bit right after detection of the suspended state, the K-state will be output after 2 ms.

Do not write 1 to this bit, unless the device state is in the suspended state (the INTSTS0.DVSQ[2:0] bits = 1xx) and the USB host enables the remote wakeup signal. When this bit is set to 1, the internal clock must not be stopped even in the suspended state (write 1 to this bit while the SYSCFG.SCKE = 1).



### 31.2.4 CFIFO Port Register (CFIFO) D0FIFO Port Register (D0FIFO) D1FIFO Port Register (D1FIFO)

Address(es): USB0.CFIFO 000A 0014h, USB0.D0FIFO 000A 0018h, USB0.D1FIFO 000A 001Ch



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	FIFOPORT[15:0]	FIFO Port	The valid bits in a FIFO port register depend on the settings of the corresponding MBW and BIGEND bits as listed in Table 31.4 and Table 31.5.	R/W

CFIFO, D0FIFO, and D1FIFO are port registers that are used to read data from the FIFO buffer memory and write data to the FIFO buffer memory.

There are three FIFO ports: CFIFO, D0FIFO, and D1FIFO ports. Each FIFO port is configured of a port register (CFIFO, D0FIFO, or D1FIFO) that handles reading of data from the FIFO buffer memory and writing of data to the FIFO buffer memory, a port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL) that is used to select the pipe assigned to the FIFO port, and a port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR).

Each FIFO port has the following features.

- The FIFO buffer for DCP (control transfer) should be accessed through the CFIFO port.
- Accessing the FIFO buffer using DMA transfer should be performed through the D0FIFO or D1FIFO port.
- The D1FIFO and D0FIFO ports can be accessed also by the CPU.
- When using functions specific to the FIFO port, the pipe number (selected pipe) specified by the CURPIPE[3:0] bits in the port select register cannot be changed (when the DMA transfer function is used, etc.).
- Registers configuring a FIFO port do not affect other FIFO ports.
- The same pipe should not be assigned to two or more FIFO ports.
- There are two FIFO buffer states: the access right is on the CPU side and it is on the SIE side. When the FIFO buffer access right is on the SIE side, the FIFO buffer cannot be accessed from the CPU.

#### FIFOPORT[15:0] Bits (FIFO Port)

Accessing the FIFOPORT[15:0] bits allows reading the received data from the FIFO buffer or writing the transmit data to the FIFO buffer.

Each FIFO port register can be accessed only while the FRDY bit in each port control register (CFIFOCTR, D0FIFOCTR, or D1FIFOCTR) is 1.

The valid bits in a FIFO port register depend on the settings of the corresponding MBW and BIGEND bits of port select register (CFIFOSEL, D0FIFOSEL, or D1FIFOSEL), as listed in Table 31.4 and Table 31.5.

**Table 31.4 Endian Operation in 16-Bit Access**

CFIFOSEL.BIGEND Bit D0FIFOSEL.BIGEND Bit D1FIFOSEL.BIGEND Bit	Bits 15 to 8	Bits 7 to 0
0	N + 1 data	N + 0 data
1	N + 0 data	N + 1 data

**Table 31.5 Endian Operation in 8-Bit Access**

CFIFOSEL.BIGEND Bit D0FIFOSEL.BIGEND Bit D1FIFOSEL.BIGEND Bit	Bits 15 to 8	Bits 7 to 0
0	Access prohibited*1	N + 0 data
1	Access prohibited*1	N + 0 data

Note 1. Accessing an access-prohibited area is not allowed.

### 31.2.5 CFIFO Port Select Register (CFIFOSEL) D0FIFO Port Select Register (D0FIFOSEL) D1FIFO Port Select Register (D1FIFOSEL)

- CFIFOSEL

Address(es): USB0.CFIFOSEL 000A 0020h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
RCNT	REW	—	—	—	MBW	—	BIGEND	—	—	ISEL	—	CURPIPE[3:0]			
Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CURPIPE[3:0]	CFIFO Port Access Pipe Specification	b3 b2 b1 b0 0 0 0 0: DCP (Default control pipe) 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Other than above: Setting prohibited	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	ISEL	CFIFO Port Access Direction When DCP is Selected	0: Reading from the buffer memory is selected 1: Writing to the buffer memory is selected	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BIGEND	CFIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	MBW	CFIFO Port Access Bit Width	0: 8-bit width 1: 16-bit width	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	REW	Buffer Pointer Rewind	0: The buffer pointer is not rewind. 1: The buffer pointer is rewind.	R/W*1
b15	RCNT	Read Count Mode	0: The DTLN[8:0] bits are cleared when all of the receive data has been read from the CFIFO. (In double buffer mode, the DTLN bit value is cleared when all the data has been read from only a single plane.) 1: The DTLN[8:0] bits are decremented each time the receive data is read from the CFIFO.	R/W

Note 1. This bit is read as 0.

The same pipe should not be specified by the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL. When the CURPIPE[3:0] bits in D0FIFOSEL and D1FIFOSEL are cleared to 0000b, no pipe is selected.

The pipe number should not be changed while the DMA transfer is enabled.

#### CURPIPE[3:0] Bits (CFIFO Port Access Pipe Specification)

The CURPIPE[3:0] bits specify the pipe number using which data is read or written through the CFIFO port.

After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process.

Do not set the same pipe number to the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL. Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.

#### **ISEL Bit (CFIFO Port Access Direction When DCP is Selected)**

After writing to the ISEL bit with the DCP being a selected pipe, read this bit to check that the written value agrees with the read value before proceeding to the next process.

Set this bit and the CURPIPE[3:0] bits simultaneously.

#### **MBW Bit (CFIFO Port Access Bit Width)**

The MBW bit specifies the bit width for accessing the CFIFO port.

When the selected pipe is in the receiving direction, once reading data is started after setting this bit, this bit should not be modified until all the data has been read.

When the selected pipe is in the receiving direction, set the CURPIPE[3:0] and MBW bits simultaneously.

When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory.

An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

#### **REW Bit (Buffer Pointer Rewind)**

The REW bit specifies whether or not to rewind the buffer pointer.

When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed).

Do not set the REW bit to 1 simultaneously with modifying the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY bit is 1.

To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

• D0FIFOSEL, D1FIFOSEL

Address(es): USB0.D0FIFOSEL 000A 0028h, USB0.D1FIFOSEL 000A 002Ch



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CURPIPE[3:0]	FIFO Port Access Pipe Specification	b3 b2 b1 b0 0 0 0 0: DCP (Default control pipe) 0 0 0 1: Pipe 1 0 0 1 0: Pipe 2 0 0 1 1: Pipe 3 0 1 0 0: Pipe 4 0 1 0 1: Pipe 5 0 1 1 0: Pipe 6 0 1 1 1: Pipe 7 1 0 0 0: Pipe 8 1 0 0 1: Pipe 9 Other than above: Setting prohibited	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BIGEND	FIFO Port Endian Control	0: Little endian 1: Big endian	R/W
b9	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10	MBW	FIFO Port Access Bit Width	0: 8-bit width 1: 16-bit width	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b12	DREQE	DMA Transfer Request Enable	0: DMA transfer request is disabled. 1: DMA transfer request is enabled.	R/W
b13	DCLRM	Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read	0: Auto buffer clear mode is disabled. 1: Auto buffer clear mode is enabled.	R/W
b14	REW	Buffer Pointer Rewind	0: The buffer pointer is not rewind. 1: The buffer pointer is rewind.	R/W*1
b15	RCNT	Read Count Mode	0: The DTLN bits are cleared when all of the receive data has been read from the DnFIFO. (In double buffer mode, the DTLN bit value is cleared when all the data has been read from only a single plane.) 1: The DTLN[8:0] bits are decremented each time the receive data is read from the DnFIFO. (n = 0, 1)	R/W

Note 1. This bit is read as 0.

The same pipe should not be specified by the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL. When the CURPIPE[3:0] bits in D0FIFOSEL and D1FIFOSEL are cleared to 0000b, no pipe is selected.

The pipe number should not be changed while the DMA transfer is enabled.

**CURPIPE[3:0] Bits (FIFO Port Access Pipe Specification)**

The CURPIPE[3:0] bits specify the pipe number using which data is read or written through the D0FIFO port or D1FIFO port.

After writing to these bits, read these bits to check that the written value agrees with the read value before proceeding to the next process.

Do not set the same pipe number to the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL. Even if an attempt is made to modify the setting of these bits during access to the FIFO buffer, the current access setting is retained until the access is completed. Then, the modification becomes effective, thus enabling continuous access.

#### **MBW Bit (FIFO Port Access Bit Width)**

The MBW bit specifies the bit width for accessing the D0FIFO port or D1FIFO port.

When the selected pipe is in the receiving direction, once reading data is started after setting this bit, this bit should not be modified until all the data has been read.

When the selected pipe is in the receiving direction, set the CURPIPE[3:0] bits and MBW bits simultaneously.

When the selected pipe is in the transmitting direction, the bit width cannot be changed from 8-bit width to 16-bit width while data is being written to the buffer memory.

An odd number of bytes can also be written through byte-access control even when 16-bit width is selected.

#### **DREQE Bit (DMA Transfer Request Enable)**

The DREQE bit enables or disables the DMA transfer request to be issued.

Before setting the DREQE bit to 1 to enable the DMA transfer request to be issued, set the CURPIPE[3:0] bits.

When modifying the setting of the CURPIPE[3:0] bits, set this bit to 0 first.

#### **DCLRM Bit (Auto Buffer Memory Clear Mode Accessed after Specified Pipe Data is Read)**

The DCLRM bit enables or disables the buffer memory to be cleared automatically after data has been read out using the selected pipe.

With this bit set to 1, the USB module sets the BCLR bit to 1 for the FIFO buffer of the selected pipe on receiving a zero-length packet while the FIFO buffer assigned to the selected pipe is empty, or on receiving a short packet and reading the data while the PIPECFG.BFRE bit is 1.

When using the USB module with the SOFCFG.BRDYM bit set to 1, set this bit to 0.

#### **REW Bit (Buffer Pointer Rewind)**

The REW bit specifies whether or not to rewind the buffer pointer.

When the selected pipe is in the receiving direction, setting the REW bit to 1 while the FIFO buffer is being read allows re-reading the FIFO buffer from the first data (in double buffer mode, re-reading the currently-read FIFO buffer plane from the first data is allowed).

Do not set the REW bit to 1 simultaneously with modifying the CURPIPE[3:0] bits. Before setting the REW bit to 1, be sure to check that the FRDY bit is 1.

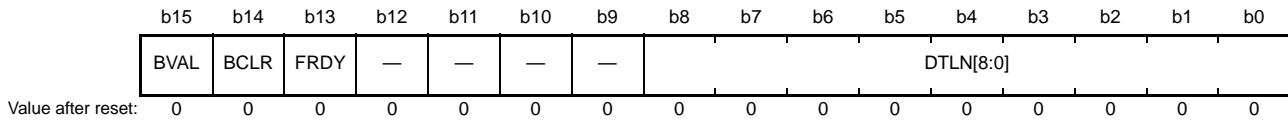
To re-write to the FIFO buffer again from the first data for the pipe in the transmitting direction, use the BCLR bit.

#### **RCNT Bit (Read Count Mode)**

When accessing DnFIFO with the PIPECFG.BFRE bit set to 1, set the RCNT bit to 0.

### 31.2.6 CFIFO Port Control Register (CFIFOCTR) D0FIFO Port Control Register (D0FIFOCTR) D1FIFO Port Control Register (D1FIFOCTR)

Address(es): USB0.CFIFOCTR 000A 0022h, USB0.D0FIFOCTR 000A 002Ah, USB0.D1FIFOCTR 000A 002Eh



Bit	Symbol	Bit Name	Description	R/W
b8 to b0	DTLN[8:0]	Receive Data Length	Indicates the length of the receive data. These bits indicate different values depending on the setting of the RCNT bit in the port select register. For details, refer to the description on the DTLN[8:0] bits shown below.	R
b12 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b13	FRDY	FIFO Port Ready	0: FIFO port access is disabled. 1: FIFO port access is enabled.	R
b14	BCLR	CPU Buffer Clear	0: Invalid 1: Clears the buffer memory on the CPU side.	R/W*1
b15	BVAL	Buffer Memory Valid Flag	0: Invalid 1: Writing ended	R/W*2

Note 1. Only 0 can be read and 1 can be written.

Note 2. Only 1 can be written.

The FIFO port control registers are used for the corresponding FIFO ports.

#### DTLN[8:0] Bits (Receive Data Length)

The DTLN[8:0] bits indicate the length of the receive data.

While the FIFO buffer is being read, the DTLN[8:0] bits indicate different values depending on the DnFIFOSEL.RCNT (n = 0, 1) bit value as described below.

- RCNT = 0

The USB module sets the DTLN[8:0] bits to indicate the length of the receive data until the CPU (DMAC) has read all the received data from a single FIFO buffer plane.

While PIPECFG.BFRE = 1, these bits retain the length of the receive data until the BCLR bit is set to 1 even after all the data has been read.

- RCNT = 1

The USB module decrements the value indicated by these bits each time data is read from the FIFO buffer. (The value is decremented by one when the MBW bit is 0, and by two when the MBW bit is 1.)

The USB module sets DTLN[8:0] bits to 0 when all the data has been read from one FIFO buffer plane. However, in double buffer mode, if all the data has been received in the other FIFO buffer plane before all the data has been read from one plane, the USB module sets DTLN[8:0] bits to indicate the length of the receive data in the other plane when all the data has been read from the one plane.

#### FRDY Bit (FIFO Port Ready)

The FRDY bit indicates whether the FIFO port can be accessed by the CPU (DMAC).

In the following cases, the USB module sets the FRDY bit to 1 but data cannot be read via the FIFO port because there is no data to be read. In these cases, set the BCLR bit to 1 to clear the FIFO buffer, and enable transmission and reception of the next data.

- A zero-length packet is received when the FIFO buffer assigned to the selected pipe is empty.

- A short packet is received and the data is completely read while PIPECFG.BFRE = 1.

### **BCLR Bit (CPU Buffer Clear)**

The BCLR bit should be set to 1 to clear the FIFO buffer on the CPU side for the selected pipe.

When double buffer mode is set for the FIFO buffer assigned to the selected pipe, the USB module clears only one plane of the FIFO buffer even when both planes are read-enabled.

When the selected pipe is the DCP, setting the BCLR bit to 1 allows the USB module to clear the FIFO buffer regardless of whether the FIFO buffer is on the CPU side or SIE side. To clear the buffer on the SIE side, set the DCPCTR.PID[1:0] bits for the DCP to NAK before setting the BCLR bit to 1.

When the selected pipe is in the transmitting direction, if 1 is written to the BVAL and BCLR bits simultaneously, the USB module clears the data that has been written before it, enabling transmission of a zero-length packet.

When the selected pipe is not the DCP, writing 1 to the BCLR bit should be done while the FRDY bit in the FIFO port control register is 1 (set by the USB module).

### **BVAL Bit (Buffer Memory Valid Flag)**

The BVAL bit should be set to 1 when data has been completely written to the FIFO buffer on the CPU side for the pipe selected using the CURPIPE[3:0] bits (selected pipe).

When the selected pipe is in the transmitting direction, set the BVAL bit to 1 in the following cases. Then, the USB module switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.

- To transmit a short packet, set the BVAL bit to 1 after data has been written.
- To transmit a zero-length packet, set the BVAL bit to 1 before data is written to the FIFO buffer.

When data of the maximum packet size has been written for the pipe in continuous transfer mode, the USB module sets the BVAL bit to 1 and switches the FIFO buffer from the CPU side to the SIE side, thus enabling transmission.

Writing 1 to the BVAL bit should be done while the FRDY bit is 1 (set by the USB module).

When the selected pipe is in the receiving direction, do not set the BVAL bit to 1.



### 31.2.7 Interrupt Enable Register 0 (INTENB0)

Address(es): USB0.INTENB0 000A 0030h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
VBSE	RSME	SOFE	DVSE	CTRE	BEMPE	NRDYE	BRDYE	—	—	—	—	—	—	—	—
Value after reset:															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	BRDYE	Buffer Ready Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	NRDYE	Buffer Not Ready Response Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b10	BEMPE	Buffer Empty Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b11	CTRE	Control Transfer Stage Transition Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b12	DVSE	Device State Transition Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b13	SOFE	Frame Number Update Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b14	RSME	Resume Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15	VBSE	VBUS Interrupt Enable	0: Interrupt output disabled 1: Interrupt output enabled	R/W

On detecting the interrupt corresponding to the bit in INTENB0 to which software has set 1, the USB module generates the USB interrupt.

The USB module sets 1 to each status bit in INTSTS0 when a detection condition of the corresponding interrupt source has been satisfied regardless of the setting in INTENB0 (regardless of whether the interrupt output is enabled or disabled).

While the status bit in INTSTS0 corresponding to the interrupt source indicates 1, the USB module generates the USB interrupt when software modifies the corresponding interrupt enable bit in INTENB0 from 0 to 1.

### 31.2.8 BRDY Interrupt Enable Register (BRDYENB)

Address(es): USB0.BRDYENB 000A 0036h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9 BRDYE	PIPE8 BRDYE	PIPE7 BRDYE	PIPE6 BRDYE	PIPE5 BRDYE	PIPE4 BRDYE	PIPE3 BRDYE	PIPE2 BRDYE	PIPE1 BRDYE	PIPE0 BRDYE
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	PIPE0 BRDYE	BRDY Interrupt Enable for PIPE0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1 BRDYE	BRDY Interrupt Enable for PIPE1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2 BRDYE	BRDY Interrupt Enable for PIPE2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3 BRDYE	BRDY Interrupt Enable for PIPE3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4 BRDYE	BRDY Interrupt Enable for PIPE4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5 BRDYE	BRDY Interrupt Enable for PIPE5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6 BRDYE	BRDY Interrupt Enable for PIPE6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7 BRDYE	BRDY Interrupt Enable for PIPE7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8 BRDYE	BRDY Interrupt Enable for PIPE8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9 BRDYE	BRDY Interrupt Enable for PIPE9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

On detecting a BRDY interrupt for a pipe corresponding to a bit in BRDYENB that has been set to 1 by software, the USB module sets the corresponding BRDYSTS.PIPE $n$ BRDY bit ( $n = 0$  to 9) and INTSTS0.BRDY bit to 1, and generates the BRDY interrupt.

As long as at least one of the BRDYSTS.PIPE $n$ BRDY bits has the value 1, the USB module generates the BRDY interrupt when software modifies the corresponding interrupt enable bit in BRDYENB from 0 to 1.

### 31.2.9 NRDY Interrupt Enable Register (NRDYENB)

Address(es): USB0.NRDYENB 000A 0038h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9 NRDYE	PIPE8 NRDYE	PIPE7 NRDYE	PIPE6 NRDYE	PIPE5 NRDYE	PIPE4 NRDYE	PIPE3 NRDYE	PIPE2 NRDYE	PIPE1 NRDYE	PIPE0 NRDYE
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	PIPE0 NRDYE	NRDY Interrupt Enable for PIPE0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1 NRDYE	NRDY Interrupt Enable for PIPE1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2 NRDYE	NRDY Interrupt Enable for PIPE2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3 NRDYE	NRDY Interrupt Enable for PIPE3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4 NRDYE	NRDY Interrupt Enable for PIPE4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5 NRDYE	NRDY Interrupt Enable for PIPE5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6 NRDYE	NRDY Interrupt Enable for PIPE6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7 NRDYE	NRDY Interrupt Enable for PIPE7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8 NRDYE	NRDY Interrupt Enable for PIPE8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9 NRDYE	NRDY Interrupt Enable for PIPE9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

On detecting a NRDY interrupt for a pipe corresponding to a bit in NRDYENB that has been set to 1 by software, the USB module sets the corresponding NRDYSTS.PIPEnNRDY bit (n = 0 to 9) and INTSTS0.NRDY bit to 1, and generates the NRDY interrupt.

As long as at least one of the NRDYSTS.PIPEnNRDY bits has the value 1, the USB module generates the NRDY interrupt when software modifies the corresponding interrupt enable bit in NRDYENB from 0 to 1.

### 31.2.10 BEMP Interrupt Enable Register (BEMPENB)

Address(es): USB0.BEMPENB 000A 003Ah

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9 BEMPE	PIPE8 BEMPE	PIPE7 BEMPE	PIPE6 BEMPE	PIPE5 BEMPE	PIPE4 BEMPE	PIPE3 BEMPE	PIPE2 BEMPE	PIPE1 BEMPE	PIPE0 BEMPE
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

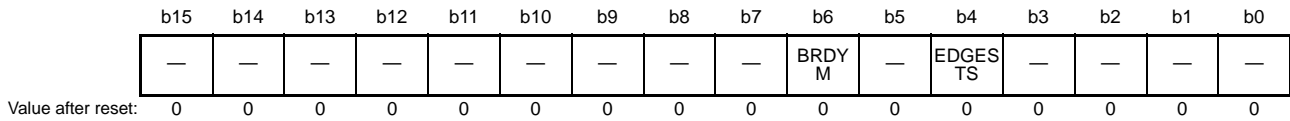
Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0 BEMPE	BEMP Interrupt Enable for PIPE0	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b1	PIPE1 BEMPE	BEMP Interrupt Enable for PIPE1	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b2	PIPE2 BEMPE	BEMP Interrupt Enable for PIPE2	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b3	PIPE3 BEMPE	BEMP Interrupt Enable for PIPE3	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b4	PIPE4 BEMPE	BEMP Interrupt Enable for PIPE4	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b5	PIPE5 BEMPE	BEMP Interrupt Enable for PIPE5	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b6	PIPE6 BEMPE	BEMP Interrupt Enable for PIPE6	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b7	PIPE7 BEMPE	BEMP Interrupt Enable for PIPE7	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b8	PIPE8 BEMPE	BEMP Interrupt Enable for PIPE8	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b9	PIPE9 BEMPE	BEMP Interrupt Enable for PIPE9	0: Interrupt output disabled 1: Interrupt output enabled	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

On detecting a BEMP interrupt for a pipe corresponding to a bit in BEMPENB that has been set to 1 by software, the USB module sets the corresponding BEMPSTS.PIPEnBEMP bit (n = 0 to 9) and INTSTS0.BEMP bit to 1, and generates the BEMP interrupt.

As long as at least one of the BEMPSTS.PIPEnBEMP bits has the value 1, the USB module generates the BEMP interrupt when software modifies the corresponding interrupt enable bit in BEMPENB from 0 to 1.

### 31.2.11 SOF Output Configuration Register (SOFCFG)

Address(es): USB0.SOFCFG 000A 003Ch



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	EDGESTS*1	Edge Interrupt Output Status Monitor	Indicates 1 when the edge interrupt output signal is in the middle of the edge processing.	R
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	BRDYM	BRDY Interrupt Status Clear Timing	0: Software clears the status. 1: The USB module clears the status when data has been read from the FIFO buffer or data has been written to the FIFO buffer.	R/W
b15 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. This bit is unnecessary in a system which does not use the edge interrupt output signal.

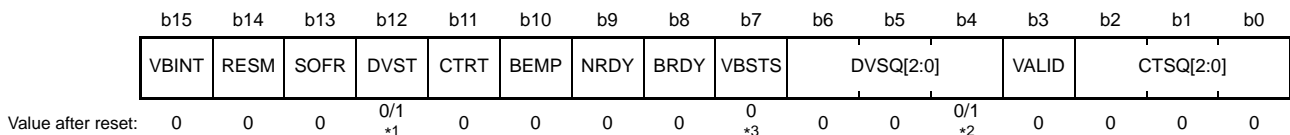
#### EDGESTS Bit (Edge Interrupt Output Status Monitor)

The EDGESTS bit indicates 1 when the edge interrupt output signal is in the middle of the edge processing. Make sure the EDGESTS bit is 0 when stopping the clock supply to the USB module.

Note: Make sure the EDGESTS bit is 0 when stopping the clock supply to the USB module.

### 31.2.12 Interrupt Status Register 0 (INTSTS0)

Address(es): USB0.INTSTS0 000A 0040h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CTSQ[2:0]	Control Transfer Stage	b2 b1 b0 0 0 0: Idle or setup stage 0 0 1: Control read data stage 0 1 0: Control read status stage 0 1 1: Control write data stage 1 0 0: Control write status stage 1 0 1: Control write (no data) status stage 1 1 0: Control transfer sequence error 1 1 1: Setting prohibited	R
b3	VALID	USB Request Reception	0: Not detected 1: Setup packet reception	R/W*4

Bit	Symbol	Bit Name	Description	R/W
b6 to b4	DVSQ[2:0]	Device State	b6 b5 b4 0 0 0: Powered state 0 0 1: Default state 0 1 0: Address state 0 1 1: Configured state 1 x x: Suspended state x: Don't care	R
b7	VBSTS	VBUS Input Status	0: USB0_VBUS pin is low. 1: USB0_VBUS pin is high.	R
b8	BRDY	Buffer Ready Interrupt Status	0: BRDY interrupts are not generated. 1: BRDY interrupts are generated.	R
b9	NRDY	Buffer Not Ready Interrupt Status	0: NRDY interrupts are not generated. 1: NRDY interrupts are generated.	R
b10	BEMP	Buffer Empty Interrupt Status	0: BEMP interrupts are not generated. 1: BEMP interrupts are generated.	R
b11	CTRT	Control Transfer Stage Transition Interrupt Status*6	0: Control transfer stage transition interrupts are not generated. 1: Control transfer stage transition interrupts are generated.	R/W*4
b12	DVST	Device State Transition Interrupt Status*6	0: Device state transition interrupts are not generated. 1: Device state transition interrupts are generated.	R/W*4
b13	SOFR	Frame Number Refresh Interrupt Status	0: SOF interrupts are not generated. 1: SOF interrupts are generated. The USB module sets the SOFR bit to 1 on updating the frame number. (A SOFR interrupt is detected every 1 ms.) The USB module can detect an SOFR interrupt through the internal interpolation function even when a damaged SOF packet is received from the USB host.	R/W*4
b14	RESM	Resume Interrupt Status*5,*6	0: Resume interrupts are not generated. 1: Resume interrupts are generated.	R/W*4
b15	VBINT	VBUS Interrupt Status*5	0: VBUS interrupts are not generated. 1: VBUS interrupts are generated.	R/W*4

Note 1. This bit is initialized to 0b by a power-on reset and 1b by a USB bus reset.

Note 2. These bits are initialized to 000b by a power-on reset and 001b by a USB bus reset.

Note 3. This bit is 1 when the USB0\_VBUS pin input is high and 0 when the input is low.

Note 4. To clear the VBINT, RESM, SOFR, DVST, CTRT, or VALID bit, write 0 only to the bits to be cleared; write 1 to the other bits. Do not write 0 to the status bits indicating 0.

Note 5. A change in the status indicated by the VBINT and RESM bits can be detected even while the clock supply is stopped (SCKE = 0), and the interrupts are output when the corresponding interrupt enable bits are enabled. Clearing the status through software should be done after enabling the clock supply.

Note 6. Disable the corresponding interrupt enable bits (set to 0).

The DVSQ[2:0] bits are initialized by a USB bus reset.

### BRDY Bit (Buffer Ready Interrupt Status)

Indicates the BRDY interrupt status.

When the BRDYENB.PIPEnBRDYE bit (n = 0 to 9) for a pipe has been set to 1 and the corresponding BRDYSTS.PIPEnBRDY bit is set to 1, (i.e. when the USB module detects the BRDY interrupt status in at least one of the pipes for which software has enabled output of the BRDY interrupt), the USB module sets the BRDY bit to 1.

For the conditions for PIPEnBRDY status assertion, refer to section 31.3.3.1, BRDY Interrupt.

The USB module clears the BRDY bit to 0 when software writes 0 to all the PIPEnBRDY bits corresponding to the PIPEnBRDYE bits to which 1 has been set.

The BRDY bit cannot be cleared to 0 even if software writes 0 to this bit.

### NRDY Bit (Buffer Not Ready Interrupt Status)

When the NRDYENB.PIPEnNRDYE bit (n = 0 to 9) for a pipe has been set to 1 and the corresponding NRDYSTS.PIPEnNRDY bit is set to 1, (i.e. when the USB module detects the NRDY interrupt status in at least one of the pipes for which software has enabled output of the NRDY interrupt), the USB module sets the NRDY bit to 1.

For the conditions for PIPEnNRDY status assertion, refer to section 31.3.3.2, NRDY Interrupt.

The USB module clears the NRDY bit to 0 when software writes 0 to all the PIPEnNRDY bits corresponding to the

PIPE<sub>n</sub>NRDYE bits to which 1 has been set.

The NRDY bit cannot be cleared to 0 even if software writes 0 to this bit.

#### **BEMP Bit (Buffer Empty Interrupt Status)**

When the BEMPENB.PIPE<sub>n</sub>BEMPE bit (n = 0 to 9) for a pipe has been set to 1 and the corresponding BEMPSTS.PIPE<sub>n</sub>BEMP bit is set to 1, (i.e. when the USB module detects the BEMP interrupt status in at least one of the pipes for which software has enabled output of the BEMP interrupt), the USB module sets the BEMP bit to 1.

For the conditions for PIPE<sub>n</sub>BEMP status assertion, refer to section 31.3.3.3, BEMP Interrupt.

The USB module clears the BEMP bit to 0 when software writes 0 to all the PIPE<sub>n</sub>BEMP bits corresponding to the PIPE<sub>n</sub>BEMPE bits to which 1 has been set.

The BEMP bit cannot be cleared to 0 even if software writes 0 to this bit.

#### **CTRTR Bit (Control Transfer Stage Transition Interrupt Status)**

The USB module updates the value of the CTSQ[2:0] bits and sets the CTRTR bit to 1 on detecting a change in the control transfer stage.

When a control transfer stage transition interrupt is generated, clear the status before the USB module detects the next control transfer stage transition.

#### **DVST Bit (Device State Transition Interrupt Status)**

The USB module updates the value of the DVSQ[2:0] bits and sets the DVST bit to 1 on detecting a change in the device state.

When a device state transition interrupt is generated, clear the status before the USB module detects the next device state transition.

#### **SOFR Bit (Frame Number Refresh Interrupt Status)**

The USB module sets the SOFR bit to 1 on updating the frame number. (A frame number refresh interrupt is detected every 1 ms.)

The USB module can detect an SOFR interrupt through the internal interpolation function even when a damaged SOF packet is received from the USB host.

#### **RESM Bit (Resume Interrupt Status)**

The USB module sets the RESM bit to 1 on detecting the falling edge of the signal on the USB0\_DP pin in the suspended state (DVSQ[2:0] = 1xx).

#### **VBINT Bit (VBUS Interrupt Status)**

The USB module sets the VBINT bit to 1 on detecting a level change (high to low or low to high) in the USB0\_VBUS pin input value. The USB module sets the VBSTS bit to indicate the USB0\_VBUS pin input value. When the VBUS interrupt is generated, use software to repeat reading the VBSTS bit until the same value is read three or more times, and eliminate chattering.

### 31.2.13 BRDY Interrupt Status Register (BRDYSTS)

Address(es): USB0.BRDYSTS 000A 0046h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9BRDY	PIPE8BRDY	PIPE7BRDY	PIPE6BRDY	PIPE5BRDY	PIPE4BRDY	PIPE3BRDY	PIPE2BRDY	PIPE1BRDY	PIPE0BRDY
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	PIPE0BRDY	BRDY Interrupt Status for PIPE0*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b1	PIPE1BRDY	BRDY Interrupt Status for PIPE1*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b2	PIPE2BRDY	BRDY Interrupt Status for PIPE2*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b3	PIPE3BRDY	BRDY Interrupt Status for PIPE3*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b4	PIPE4BRDY	BRDY Interrupt Status for PIPE4*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b5	PIPE5BRDY	BRDY Interrupt Status for PIPE5*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b6	PIPE6BRDY	BRDY Interrupt Status for PIPE6*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b7	PIPE7BRDY	BRDY Interrupt Status for PIPE7*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b8	PIPE8BRDY	BRDY Interrupt Status for PIPE8*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b9	PIPE9BRDY	BRDY Interrupt Status for PIPE9*2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. When the SOFCFG.BRDYM bit is set to 0, to clear the status indicated by the bits in BRDYSTS, write 0 only to the bits to be cleared; write 1 to the other bits.

Note 2. When the SOFCFG.BRDYM bit is set to 0, clearing BRDY interrupts should be done before accessing the FIFO.



## 31.2.14 NRDY Interrupt Status Register (NRDYSTS)

Address(es): USB0.NRDYSTS 000A 0048h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9NRDY	PIPE8NRDY	PIPE7NRDY	PIPE6NRDY	PIPE5NRDY	PIPE4NRDY	PIPE3NRDY	PIPE2NRDY	PIPE1NRDY	PIPE0NRDY
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	PIPE0NRDY	NRDY Interrupt Status for PIPE0	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b1	PIPE1NRDY	NRDY Interrupt Status for PIPE1	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b2	PIPE2NRDY	NRDY Interrupt Status for PIPE2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b3	PIPE3NRDY	NRDY Interrupt Status for PIPE3	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b4	PIPE4NRDY	NRDY Interrupt Status for PIPE4	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b5	PIPE5NRDY	NRDY Interrupt Status for PIPE5	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b6	PIPE6NRDY	NRDY Interrupt Status for PIPE6	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b7	PIPE7NRDY	NRDY Interrupt Status for PIPE7	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b8	PIPE8NRDY	NRDY Interrupt Status for PIPE8	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b9	PIPE9NRDY	NRDY Interrupt Status for PIPE9	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To clear the status indicated by the bits in NRDYSTS, write 0 only to the bits to be cleared; write 1 to the other bits.

### 31.2.15 BEMP Interrupt Status Register (BEMPSTS)

Address(es): USB0.BEMPSTS 000A 004Ah

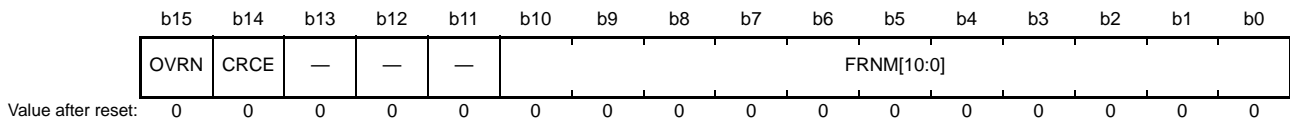
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	PIPE9B EMP	PIPE8B EMP	PIPE7B EMP	PIPE6B EMP	PIPE5B EMP	PIPE4B EMP	PIPE3B EMP	PIPE2B EMP	PIPE1B EMP	PIPE0B EMP
Value after reset:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PIPE0BEMP	BEMP Interrupt Status for PIPE0	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b1	PIPE1BEMP	BEMP Interrupt Status for PIPE1	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b2	PIPE2BEMP	BEMP Interrupt Status for PIPE2	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b3	PIPE3BEMP	BEMP Interrupt Status for PIPE3	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b4	PIPE4BEMP	BEMP Interrupt Status for PIPE4	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b5	PIPE5BEMP	BEMP Interrupt Status for PIPE5	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b6	PIPE6BEMP	BEMP Interrupt Status for PIPE6	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b7	PIPE7BEMP	BEMP Interrupt Status for PIPE7	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b8	PIPE8BEMP	BEMP Interrupt Status for PIPE8	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b9	PIPE9BEMP	BEMP Interrupt Status for PIPE9	0: Interrupts are not generated. 1: Interrupts are generated.	R/W*1
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. To clear the status indicated by the bits in BEMPSTS, write 0 only to the bits to be cleared; write 1 to the other bits.

### 31.2.16 Frame Number Register (FRMNUM)

Address(es): USB0.FRNUM 000A 004Ch



Bit	Symbol	Bit Name	Description	R/W
b10 to b0	FRNM[10:0]	Frame Number	These bits indicate a frame number.	R
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	CRCE	Receive Data Error	0: No error 1: An error occurred	R/W*1
b15	OVRN	Overrun/Underrun Detection Status	0: No error 1: An error occurred	R/W*1

Note 1. Only 0 can be written.

#### FRNM[10:0] Bits (Frame Number)

The USB module sets the FRNM[10:0] bits to indicate the latest frame number, which is updated every time an SOF packet is issued or received (every 1 ms).

Repeat reading the FRNM[10:0] bits until the same value is read twice.

#### CRCE Bit (Receive Data Error)

Indicates whether a CRC error or bit stuffing error has been detected in the pipe during isochronous transfer.

Software can clear the CRCE bit to 0 by writing 0 to the CRCE bit. Here, 1 should be written to the other bits in FRMNUM.

On detecting a CRC error, the USB module does not generate the internal NRDY interrupt request.

#### OVRN Bit (Overrun/Underrun Detection Status)

Indicates whether an overrun/underrun error has been detected in the pipe during isochronous transfer.

Software can clear the OVRN bit to 0 by writing 0 to the OVRN bit. Here, 1 should be written to the other bits in FRMNUM.

The USB module sets the OVRN bit to 1 on any of the following conditions.

- For the isochronous transfer pipe in the transmitting direction, the IN token is received before all the transmit data has been written to the FIFO buffer.
- For the isochronous transfer pipe in the receiving direction, the OUT token is received when no FIFO buffer planes are empty.

### 31.2.17 Device State Changing Register (DVCHGR)

Address(es): USB0.DVCHGR 000A 004Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	DVCHG	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
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	DVCHG	Device State Changing	0: Writing to the USBADDR.STSRECOV[3:0] and USBADDR.USBADDR[6:0] bits is disabled. 1: Writing to the USBADDR.STSRECOV[3:0] and USBADDR.USBADDR[6:0] bits is enabled.	R/W

For details, refer to section 31.3.1.2, Canceling Deep Software Standby Mode by a USB Suspend/Resume Interrupt.

### 31.2.18 USB Address Register (USBADDR)

Address(es): USB0.USBADDR 000A 0050h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	STSRECOV[3:0]			—	USBADDR[6:0]							
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															
b6 to b0	USBADDR[6:0]	USB Address	These bits indicate the USB address assigned by the host when the SET_ADDRESS request is successfully processed.	R/W															
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W															
b11 to b8	STSRECOV[3:0]	Status Recovery	<table border="0"> <tr> <td>b11</td><td>b8</td><td></td></tr> <tr> <td>1 0 0 1:</td><td>Recovery is to the full-speed state (DVSTCTR0.RHST[2:0] = 010), and INSTS0.DVSQ[2:0] = 001 (default state).</td><td></td></tr> <tr> <td>1 0 1 0:</td><td>Recovery is to the full-speed state (DVSTCTR0.RHST[2:0] = 010), and INSTS0.DVSQ[2:0] = 010 (address state).</td><td></td></tr> <tr> <td>1 0 1 1:</td><td>Recovery is to the full-speed state (DVSTCTR0.RHST[2:0] = 010), and INSTS0.DVSQ[2:0] = 011 (configured state).</td><td></td></tr> <tr> <td colspan="3">Other than above: Setting prohibited</td></tr> </table>	b11	b8		1 0 0 1:	Recovery is to the full-speed state (DVSTCTR0.RHST[2:0] = 010), and INSTS0.DVSQ[2:0] = 001 (default state).		1 0 1 0:	Recovery is to the full-speed state (DVSTCTR0.RHST[2:0] = 010), and INSTS0.DVSQ[2:0] = 010 (address state).		1 0 1 1:	Recovery is to the full-speed state (DVSTCTR0.RHST[2:0] = 010), and INSTS0.DVSQ[2:0] = 011 (configured state).		Other than above: Setting prohibited			R/W
b11	b8																		
1 0 0 1:	Recovery is to the full-speed state (DVSTCTR0.RHST[2:0] = 010), and INSTS0.DVSQ[2:0] = 001 (default state).																		
1 0 1 0:	Recovery is to the full-speed state (DVSTCTR0.RHST[2:0] = 010), and INSTS0.DVSQ[2:0] = 010 (address state).																		
1 0 1 1:	Recovery is to the full-speed state (DVSTCTR0.RHST[2:0] = 010), and INSTS0.DVSQ[2:0] = 011 (configured state).																		
Other than above: Setting prohibited																			
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W															

#### USBADDR[6:0] Bits (USB Address)

On detecting the USB bus reset, the USB module sets the USBADDR[6:0] bits to 00h.

The writing to the USBADDR[6:0] bits is enabled while the DVCHGR.DVCHG bits are set to 1. On returning from the USB power shut-off, the operation can resume to the USB address before the shut-off by the software.

The USBADDR[6:0] bits are initialized by a USB bus reset detection.

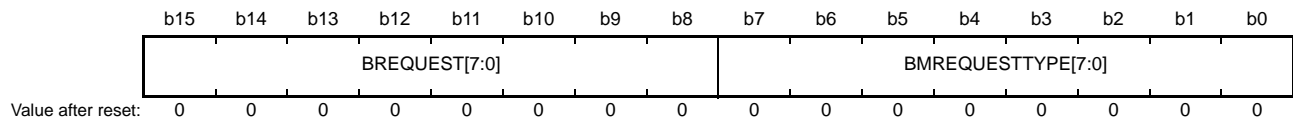
**STSRECOV[3:0] Bits (Status Recovery)**

These bits are used in recovery from interruptions to the USB power supply when recovery is to the pre-interruption state of the USB module's internal sequencer. For details, see section 31.3.1.2, Canceling Deep Software Standby Mode by a USB Suspend/Resume Interrupt.

Writing to the STSRECOV[3:0] bits is only possible when the DVCHG bit in the DVCHGR is 1.

**31.2.19 USB Request Type Register (USBREQ)**

Address(es): USB0.USBREQ 000A 0054h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	BMREQUESTTYPE[7:0]	Request Type	The USB request bmRequestType value	R
b15 to b8	BREQUEST[7:0]	Request	The USB request bRequest value	R

USBREQ stores setup requests for control transfers.

USBREQ is initialized by a USB bus reset.

**BMREQUESTTYPE[7:0] Bits (Request Type)**

The BMREQUESTTYPE[7:0] bits store the USB request bmRequestType value.

These bits indicate the USB request data value received during the setup transaction. Writing to these bits is invalid.

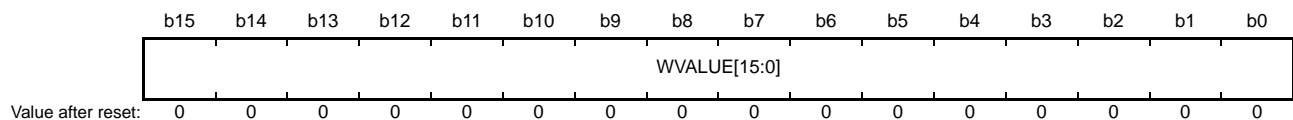
**BREQUEST[7:0] Bits (Request)**

The BREQUEST[7:0] bits store the USB request bRequest value.

These bits indicate the USB request data value received during the setup transaction. Writing to these bits is invalid.

**31.2.20 USB Request Value Register (USBVAL)**

Address(es): USB0.USBVAL 000A 0056h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	WVALUE[15:0]	Value	The USB request wValue value	R

USBVAL is initialized by a USB bus reset.

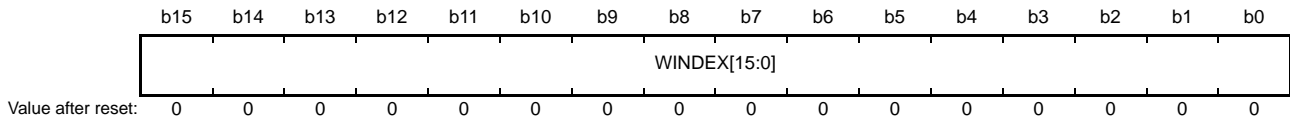
**WVALUE[15:0] Bits (Value)**

The WVALUE[15:0] bits store the USB request wValue value.

These bits indicate the USB request wValue value received during the setup transaction. Writing to these bits is invalid.

### 31.2.21 USB Request Index Register (USBINDX)

Address(es): USB0.USBINDX 000A 0058h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	WINDEX[15:0]	Index	The USB request wIndex value	R

USBINDX stores setup requests for control transfers.

USBINDX is initialized by a USB bus reset.

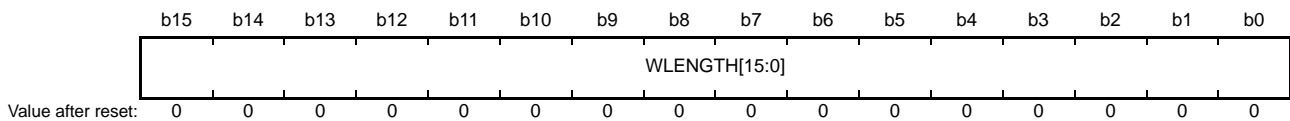
#### WINDEX[15:0] Bits (Index)

The WINDEX[15:0] bits store the USB request wIndex value.

These bits indicate the USB request wIndex value received during the setup transaction. Writing to these bits is invalid.

### 31.2.22 USB Request Length Register (USBLENG)

Address(es): USB0.USBLENG 000A 005Ah



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	WLENGTH[15:0]	Length	The USB request wLength value	R

USBLENG stores setup requests for control transfers.

USBLENG is initialized by a USB bus reset.

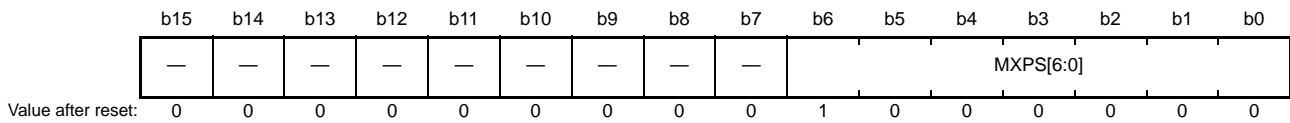
#### WLENGTH[15:0] Bits (Length)

The WLENGTH[15:0] bits store the USB request wLength value.

These bits indicate the USB request wLength value received during the setup transaction. Writing to these bits is invalid.

### 31.2.23 DCP Maximum Packet Size Register (DCPMAXP)

Address(es): USB0.DCPMAXP 000A 005Eh



Bit	Symbol	Bit Name	Description	R/W
b6 to b0	MXPS[6:0]	Maximum Packet Size	These bits specify the maximum data payload (maximum packet size) for the DCP.	R/W
b15 to b7	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

#### MXPS[6:0] Bits (Maximum Packet Size)

Set the maximum data payload (maximum packet size) for the DCP to the MXPS[6:0] bits.

The value after reset is 40h (64 bytes).

These bits should be set to the value based on the USB Specifications.

While MXPS[6:0] = 0, do not write to the FIFO buffer or do not set PID to BUF.

Modify the MXPS[6:0] bits while PID is NAK. Before modifying these bits after modifying the PID[1:0] bits for the DCP from BUF to NAK, check that the DCPCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB module, checking the DCPCTR.PBUSY bit through software is not necessary. After modifying the MXPS[6:0] bits and the DCP has been set to the C/DnFIFOSEL.CURPIPE[3:0] bits, clear the buffer by setting the C/DnFIFOCTR.BCLR bit to 1.

### 31.2.24 DCP Control Register (DCPCTR)

Address(es): USB0.DCPCTR 000A 0060h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BSTS	—	—	—	—	—	—	SQCLR	SQSET	SQMON	PBUSY	—	—	CCPL	PID[1:0]	
Value after reset:															
0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
b2	CCPL	Control Transfer End Enable	0: Invalid 1: Completion of control transfer is enabled.	R/W
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy	0: DCP is not used for the transaction. 1: DCP is used for the transaction.	R
b6	SQMON	Sequence Toggle Bit Monitor	0: DATA0 1: DATA1	R
b7	SQSET	Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W*1
b8	SQCLR	Toggle Bit Clear*2	0: Invalid 1: Specifies DATA0.	R/W*1
b14 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	BSTS	Buffer Status	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. This bit is read as 0. Only 1 can be written.

Note 2. Write 1 to the SQSET and SQCLR bits while PID is NAK. Before modifying these bits after modifying the PID[1:0] bits for the DCP from BUF to NAK, check that the DCPCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB module, checking the DCPCTR.PBUSY bit through software is not necessary.

The CCPL and PID[1:0] bits in DCPCTR are initialized by a USB bus reset.

#### PID[1:0] Bits (Response PID)

The PID[1:0] bits control the response type of the USB module during control transfer.

The USB module modifies the setting of the PID[1:0] bits as follows.

- The USB module modifies the PID[1:0] bits to NAK on receiving the setup packet. Here, the USB module sets VALID to 1. Software cannot modify the setting of the PID[1:0] bits until software sets VALID to 0.
- The USB module sets PID to STALL (11b) on receiving the data of a size exceeding the maximum packet size when software has set the PID[1:0] bits to BUF.
- The USB module sets PID to STALL (1xb) on detecting the control transfer sequence error.
- The USB module sets PID to NAK on detecting the USB bus reset.

The USB module does not reference to the setting of the PID[1:0] bits while the SET\_ADDRESS request is processed (auto processing).

#### CCPL Bit (Control Transfer End Enable)

Setting the CCPL bit to 1 enables the status stage of the control transfer to be completed.

When software sets the CCPL bit to 1 while the corresponding PID[1:0] bits are set to BUF, the USB module completes the control transfer stage.



Specifically, during control read transfer, the USB module transmits the ACK handshake in response to the OUT transaction from the USB host, and transmits the zero-length packet in response to the IN transaction from the USB host during control write or no-data control transfer. However, on detecting the SET\_ADDRESS request, the USB module operates in auto response mode from the setup stage up to the status stage completion irrespective of the setting of the CCPL bit.

The USB module modifies the CCPL bit from 1 to 0 on receiving a new setup packet.  
Software cannot write 1 to the CCPL bit while the VALID bit is 1.

### **PBUSY Bit (Pipe Busy)**

The PBUSY bit indicates whether DCP is used or not for the transaction when DCP changes the PID[1:0] bits from BUF to NAK.

The USB module modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction.

Reading the PBUSY bit after software has set PID to NAK allows checking whether modification of the pipe settings is possible.

For details, refer to section 31.3.4.1, Pipe Control Register Switching Procedures.

### **SQMON Bit (Sequence Toggle Bit Monitor)**

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction during the DCP transfer. The USB module allows the SQMON bit to toggle upon normal completion of the transaction. However, the SQMON bit is not allowed to toggle when a DATA-PID disagreement occurs during the transfer in the receiving direction.

The USB module sets the SQMON bit to 1 (specifies DATA1 as the expected value) upon successful reception of the setup packet.

The USB module does not reference the SQMON bit during the IN/OUT transaction of the status stage, and does not allow the SQMON bit to toggle upon normal completion.

### **SQSET Bit (Toggle Bit Set)**

The SQSET bit specifies DATA1 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer.

Do not set the SQCLR and SQSET bits to 1 simultaneously.

### **SQCLR Bit (Toggle Bit Clear)**

The SQCLR bit specifies DATA0 as the expected value of the sequence toggle bit for the next transaction during the DCP transfer. The SQCLR bit always indicates 0

Do not set the SQCLR and SQSET bits to 1 simultaneously.

### **BSTS Bit (Buffer Status)**

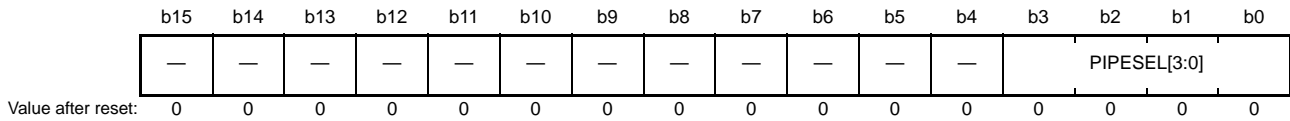
Indicates whether DCP FIFO buffer access is enabled or disabled.

The meaning of the BSTS bit differs according to the setting for the ISEL bit in the port select register as follows.

- When ISEL = 0, the BSTS bit indicates whether the received data can be read from the buffer.
- When ISEL = 1, the BSTS bit indicates whether the data to be transmitted can be written to the buffer.

### 31.2.25 Pipe Window Select Register (PIPESEL)

Address(es): USB0.PIPESEL 000A 0064h



Bit	Symbol	Bit Name	Description	R/W																																																																																								
b3 to b0	PIPESEL[3:0]	Pipe Window Select	<table style="width:100%; border:none;"> <tr> <td style="text-align:right;">b3</td><td style="text-align:right;">b2</td><td style="text-align:right;">b1</td><td style="text-align:right;">b0</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>0</td> </tr> <tr> <td colspan="4">0: No pipe selected</td> </tr> <tr> <td>0</td><td>0</td><td>0</td><td>1</td> </tr> <tr> <td colspan="4">1: PIPE1</td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>0</td> </tr> <tr> <td colspan="4">0: PIPE2</td> </tr> <tr> <td>0</td><td>0</td><td>1</td><td>1</td> </tr> <tr> <td colspan="4">1: PIPE3</td> </tr> <tr> <td>0</td><td>1</td><td>0</td><td>0</td> </tr> <tr> <td colspan="4">0: PIPE4</td> </tr> <tr> <td>0</td><td>1</td><td>0</td><td>1</td> </tr> <tr> <td colspan="4">1: PIPE5</td> </tr> <tr> <td>0</td><td>1</td><td>1</td><td>0</td> </tr> <tr> <td colspan="4">0: PIPE6</td> </tr> <tr> <td>0</td><td>1</td><td>1</td><td>1</td> </tr> <tr> <td colspan="4">1: PIPE7</td> </tr> <tr> <td>1</td><td>0</td><td>0</td><td>0</td> </tr> <tr> <td colspan="4">0: PIPE8</td> </tr> <tr> <td>1</td><td>0</td><td>0</td><td>1</td> </tr> <tr> <td colspan="4">1: PIPE9</td> </tr> <tr> <td colspan="4">Other than above: Setting prohibited</td> </tr> </table>	b3	b2	b1	b0	0	0	0	0	0: No pipe selected				0	0	0	1	1: PIPE1				0	0	1	0	0: PIPE2				0	0	1	1	1: PIPE3				0	1	0	0	0: PIPE4				0	1	0	1	1: PIPE5				0	1	1	0	0: PIPE6				0	1	1	1	1: PIPE7				1	0	0	0	0: PIPE8				1	0	0	1	1: PIPE9				Other than above: Setting prohibited				R/W
b3	b2	b1	b0																																																																																									
0	0	0	0																																																																																									
0: No pipe selected																																																																																												
0	0	0	1																																																																																									
1: PIPE1																																																																																												
0	0	1	0																																																																																									
0: PIPE2																																																																																												
0	0	1	1																																																																																									
1: PIPE3																																																																																												
0	1	0	0																																																																																									
0: PIPE4																																																																																												
0	1	0	1																																																																																									
1: PIPE5																																																																																												
0	1	1	0																																																																																									
0: PIPE6																																																																																												
0	1	1	1																																																																																									
1: PIPE7																																																																																												
1	0	0	0																																																																																									
0: PIPE8																																																																																												
1	0	0	1																																																																																									
1: PIPE9																																																																																												
Other than above: Setting prohibited																																																																																												
b15 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W																																																																																								

PIPE1 to PIPE 9 should be set using PIPESEL, PIPECFG, PIPEMAXP, PIPEPERI, PIPEnCTR, PIPEnTRE, and PIPEnTRN.

After selecting the pipe using PIPESEL, functions of the pipe should be set using PIPECFG, PIPEMAXP, and PIPEPERI. PIPEnCTR, PIPEnTRE, and PIPEnTRN can be set regardless of the pipe selection in PIPESEL.

#### PIPESEL[3:0] Bits (Pipe Window Select)

The PIPESEL[3:0] bits select the pipe number corresponding to PIPECFG, PIPEMAXP, and PIPEPERI which data are written to or read from.

Selecting a pipe number through the PIPESEL[3:0] bits allows writing to and reading from PIPECFG, PIPEMAXP, and PIPEPERI which correspond to the selected pipe number.

When PIPESEL[3:0] = 0000b, 0 is read from all of the bits in PIPECFG, PIPEMAXP, PIPEPERI, and PIPEnCTR.

Writing to these bits is invalid.

## 31.2.26 Pipe Configuration Register (PIPECFG)

Address(es): USB0.PIPECFG 000A 0068h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
TYPE[1:0]		—	—	—	BFRE	DBLB	—	SHTNAK	—	—	DIR	EPNUM[3:0]			
Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	EPNUM[3:0]	Endpoint Number*1	These bits specify the endpoint number for the selected pipe. Setting 0000 means unused pipe.	R/W
b4	DIR	Transfer Direction*2,*3	0: Receiving direction 1: Transmitting direction	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SHTNAK	Pipe Disabled at End of Transfer*1	0: Pipe continued at the end of transfer 1: Pipe disabled at the end of transfer	R/W
b8	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b9	DBLB	Double Buffer Mode*2,*3	0: Single buffer 1: Double buffer	R/W
b10	BFRE	BRDY Interrupt Operation Specification *2,*3	0: BRDY interrupt upon transmitting or receiving data 1: BRDY interrupt upon completion of reading data	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15, b14	TYPE[1:0]	Transfer Type*1	<ul style="list-style-type: none"> <li>• PIPE1 and PIPE2 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Isochronous transfer</li> <li>• PIPE3 to PIPE5 b15 b14 0 0: Pipe not used 0 1: Bulk transfer 1 0: Setting prohibited 1 1: Setting prohibited</li> <li>• PIPE6 to PIPE9 b15 b14 0 0: Pipe not used 0 1: Setting prohibited 1 0: Interrupt transfer 1 1: Setting prohibited</li> </ul>	R/W

Note 1. Modify the TYPE[1:0], SHTNAK, and EPNUM[3:0] bits while PID is NAK. Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB module, checking the DCPCTR.PBUSY bit through software is not necessary.

Note 2. Modify the BFRE, DBLB, and DIR bits while PID is NAK and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying these bits after modifying the DCPCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the DCPCTR.PBUSY bit is 0. However, if the DCPCTR.PID[1:0] bits have been modified to NAK by the USB module, checking the DCPCTR.PBUSY bit through software is not necessary.

Note 3. To modify the BFRE, DBLB, and DIR bits after completing USB communication using the selected pipe, write 1 and then 0 to the PIPEnCTR.ACLRM bit continuously through software to clear the FIFO buffer assigned to the selected pipe while the PID and CURPIPE[3:0] bits are in the state described in the above note 2.

**EPNUM[3:0] Bits (Endpoint Number)**

The EPNUM[3:0] bits specify the endpoint number for the selected pipe.

Setting 0000b means unused pipe.

Do not make the settings such that the combination of the settings of the DIR and EPNUM bits should be the same for two or more pipes (EPNUM[3:0] = 0000b can be set for all of the pipes).

**DIR Bit (Transfer Direction)**

The DIR bit specifies the transfer direction for the selected pipe.

When software has set the DIR bit to 0, the USB module uses the selected pipe in the receiving direction, and when software has set the DIR bit to 1, the USB module uses the selected pipe in the transmitting direction.

**SHTNAK Bit (Pipe Disabled at End of Transfer)**

The SHTNAK bit specifies whether to modify PID to NAK upon the end of transfer when the selected pipe is in the receiving direction.

The SHTNAK bit is valid when the selected pipe is PIPE1 to PIPE5 in the receiving direction.

When software has set the SHTNAK bit to 1 for the selected pipe in the receiving direction, the USB module modifies the DCPCTR.PID[1:0] bits corresponding to the selected pipe to NAK on determining the end of the transfer. The USB module determines that the transfer has ended on any of the following conditions.

- A short packet (including a zero-length packet) is successfully received.
- The transaction counter is used and the number of packets specified by the counter are successfully received.

**DBLB Bit (Double Buffer Mode)**

The DBLB bit selects either single or double buffer mode for the FIFO buffer used by the selected pipe.

The DBLB bit is valid when PIPE1 to PIPE5 are selected.

**BFRE Bit (BRDY Interrupt Operation Specification)**

Specifies the BRDY interrupt generation timing from the USB module to the CPU with respect to the selected pipe.

When software has set the BFRE bit to 1 and the selected pipe is in the receiving direction, the USB module detects the transfer completion and generates the BRDY interrupt on having read the relevant packet.

When the BRDY interrupt is generated with the above conditions, software needs to write 1 to the BCLR bit in the port control register. The FIFO buffer assigned to the selected pipe is not enabled for reception until 1 is written to the BCLR bit.

When software has set the BFRE bit to 1 and the selected pipe is in the transmitting direction, the USB module does not generate the BRDY interrupt.

For details, refer to section 31.3.3.1, BRDY Interrupt.

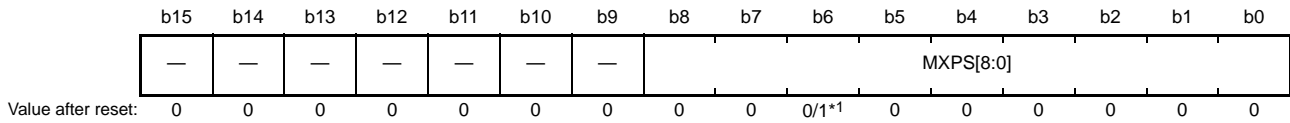
**TYPE[1:0] Bits (Transfer Type)**

The TYPE[1:0] bits select the transfer type for the pipe selected by the PIPESEL.PIPESEL[3:0] bits (selected pipe).

Before setting PID to BUF for the selected pipe (before starting USB communication using the selected pipe), be sure to set the TYPE[1:0] bits to a value other than 00b.

### 31.2.27 Pipe Maximum Packet Size Register (PIPEMAXP)

Address(es): USB0.PIPEMAXP 000A 006Ch



Note 1. The initial value of the MXPS[8:0] bits is 0000h when no pipe is selected with the PIPESEL bits in PIPESEL and 0040h when a pipe is selected.

Bit	Symbol	Bit Name	Description	R/W
b8 to b0	MXPS[8:0]	Maximum Packet Size*1	<ul style="list-style-type: none"> <li>PIPE1 and PIPE2 1 byte (001h) to 256 bytes (100h)</li> <li>PIPE3 to PIPE5 8 bytes (008h), 16 bytes (010h), 32 bytes (020h), 64 bytes (040h) (Bits [8:7] and [2:0] are not provided.)</li> <li>PIPE6 to PIPE9 1 byte (001h) to 64 bytes (040h) (Bits [8:7] are not provided.)</li> </ul>	R/W
b15 to b9	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Modify the MXPS[8:0] bits while PID is NAK, and no pipe is selected with the DnFIFOSEL.CURPIPE[3:0] bits. Before modifying these bits after modifying the PID[1:0] bits for the selected pipe from BUF to NAK, check that the DCPCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB module, checking the DCPCTR.PBUSY bit through software is not necessary.

#### MXPS[8:0] Bits (Maximum Packet Size)

Specifies the maximum data payload (maximum packet size) for the selected pipe.

These bits should be set to the appropriate value for each transfer type based on the USB Specifications. While MXPS[8:0] = 0, do not write to the FIFO buffer or set PID to BUF.

### 31.2.28 Pipe Cycle Control Register (PIPEPERI)

Address(es): USB0.PIPEPERI 000A 006Eh



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	IITV[2:0]	Interval Error Detection Interval	Specifies the interval error detection timing for the selected pipe in terms of frames, which is expressed as n-th power of 2.	R/W
b11 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b12	IFIS	Isochronous IN Buffer Flush	0: The buffer is not flushed. 1: The buffer is flushed.	R/W
b15 to b13	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

#### IITV[2:0] Bits (Interval Error Detection Interval)

Before modifying the IITV[2:0] bits after USB communication has been completed with the IITV[2:0] bits set to a certain value, set PID to NAK and then set the PIPEnCTR.ACLRM bit to 1 to initialize the interval timer. The IITV[2:0] bits are invalid for PIPE3 to PIPE5; set the IITV[2:0] bits to 000b for PIPE3 to PIPE5. Modify the IITV[2:0] bits while PID is NAK. Before modifying these bits after modifying the PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB module, checking the PIPEnCTR.PBUSY bit through software is not necessary.

#### IFIS Bit (Isochronous IN Buffer Flush)

This bit specifies whether buffer flushing is or is not applied during isochronous IN transfer through the pipe specified by the PIPESEL.PIPESEL[3:0] bits (selected pipe).  
 When the transfer type for the selected pipe is isochronous IN, the USB module automatically clears the FIFO buffer if the module fails to receive an IN token within the frame from the USB host each time the interval set by the IITV[2:0] bits elapses.  
 When double buffering is set (PIPECFG.DBLB = 1), the USB module only clears data from the older plane. The FIFO buffer is cleared at the time of SOF packet reception after the frame that was expected to include an IN token. Even if the SOF packet is corrupted, internal interpolation allows clearing of the FIFO buffer at the time the SOF packet should have been received.  
 If the selected type for transfer is other than isochronous, set the IITV[2:0] bits to 000b.

### 31.2.29 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9)

- PIPEnCTR (n = 1 to 5)

Address(es): USB0.PIPE1CTR 000A 0070h, USB0.PIPE2CTR 000A 0072h, USB0.PIPE3CTR 000A 0074h, USB0.PIPE4CTR 000A 0076h, USB0.PIPE5CTR 000A 0078h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
BSTS	INBUFM	—	—	—	ATREPM	ACLARM	SQCLR	SQSET	SQMON	PBUSY	—	—	—	PID[1:0]	
Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															

Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy	0: The relevant pipe is not used for the transaction. 1: The relevant pipe is used for the transaction.	R
b6	SQMON	Toggle Bit Confirmation	0: DATA0 1: DATA1	R
b7	SQSET	Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W*1
b8	SQCLR	Toggle Bit Clear*2	0: Invalid 1: Specifies DATA0.	R/W*1
b9	ACLARM	Auto Buffer Clear Mode*3	0: Disabled 1: Enabled (all buffers are initialized)	R/W
b10	ATREPM	Auto Response Mode*2	0: Auto response is disabled. 1: Auto response is enabled.	R/W
b13 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b14	INBUFM	Transmit Buffer Monitor	0: There is no data to be transmitted in the buffer memory. 1: There is data to be transmitted in the buffer memory.	R
b15	BSTS	Buffer Status	0: Buffer access by the CPU is disabled. 1: Buffer access by the CPU is enabled.	R

Note 1. Only 0 can be read and 1 can be written.

Note 2. Modify the ATREPM bit or write 1 to the SQCLR or SQSET bit while PID is NAK. Before modifying these bits after modifying the PID[1:0] bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB module, checking the PBUSY bit in DCPCTR through software is not necessary.

Note 3. Modify the ACLARM bit while PID is NAK and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying this bit after modifying the PID[1:0] bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

PIPEnCTR can be set regardless of the pipe selection in PIPESEL.

#### PID[1:0] Bits (Response PID)

The PID[1:0] bits specify the response type for the next transaction of the relevant pipe.

The default setting of the PID[1:0] bits is NAK. Modify the setting of the PID[1:0] bits to BUF to use the relevant pipe for USB transfer. Table 31.6 lists the basic operation (operation when there are no errors in the transmitted and received packets) of the USB module depending on the PID[1:0] bit setting.

After modifying the setting of the PID[1:0] bits through software from BUF to NAK during USB communication using the relevant pipe, check that the PBUSY bit is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state.

The USB module modifies the setting of the PID[1:0] bits in the following cases.

- The USB module sets PID to NAK on recognizing the completion of the transfer when the relevant pipe is in the receiving direction and software has set the PIPECFG.SHTNAK bit for the selected pipe to 1.
- The USB module sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB module sets PID to NAK on detecting a USB bus reset.

To specify each response type, set the PID[1:0] bits as follows.

- To make a transition from NAK (00b) to STALL, set 10b.
- To make a transition from BUF (01b) to STALL, set 11b.
- To make a transition from STALL (11b) to NAK, set 10b and then 00b.

To make a transition from STALL to BUF, set 00b (NAK) and then 01b (BUF).

### **PBUSY Bit (Pipe Busy)**

The PBUSY bit indicates whether the relevant pipe is being currently used or not for the transaction.

The USB module modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction.

Reading the PBUSY bit after software has set PID to NAK allows checking whether modification of the pipe settings is possible.

For details, refer to section 31.3.4.1, Pipe Control Register Switching Procedures.

### **SQMON Bit (Toggle Bit Confirmation)**

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe.

When the relevant pipe is not for the isochronous transfer, the USB module allows the SQMON bit to toggle upon normal completion of the transaction. However, the SQMON bit is not allowed to toggle when a DATA-PID disagreement occurs during the transfer in the receiving direction.

### **SQSET Bit (Toggle Bit Set)**

The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe.

Setting the SQSET bit to 1 through software allows the USB module to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB module sets the SQSET bit to 0.

### **SQCLR Bit (Toggle Bit Clear)**

The SQCLR bit should be set to 1 to clear the value of the sequence toggle bit for the next transaction of the relevant pipe to DATA0.

Setting the SQCLR bit to 1 through software allows the USB module to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB module sets the SQCLR bit to 0.

### **ACLRM Bit (Auto Buffer Clear Mode)**

Enables or disables auto buffer clear mode for the relevant pipe.

To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously.

Table 31.7 lists the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.

### **ATREPM Bit (Auto Response Mode)**

Enables or disables auto response mode for the relevant pipe.

When the relevant pipe is for bulk transfer, the ATREPM bit can be set to 1.



When the ATREPM bit is set to 1, the USB module responds to the token from the USB host as described below.

(1) When the relevant pipe is for bulk IN transfer (PIPECFG.TYPE[1:0] = 01b and PIPECFG.DIR = 1)

When ATREPM = 1 and PID = BUF, the USB module transmits a zero-length packet in response to the IN token.

The USB module updates (allows toggling of) the sequence toggle bit (DATA-PID) each time the USB module receives ACK from the USB host (in a single transaction, IN token is received, zero-length packet is transmitted, and then ACK is received.).

In this case, the USB module does not generate the BRDY or BEMP interrupt.

(2) When the relevant pipe is for bulk OUT transfer (PIPECFG.TYPE[1:0] = 01b and PIPECFG.DIR = 0)

When ATREPM = 1 and PID = BUF, the USB module returns NAK in response to the OUT token and generates the NRDY interrupt.

For USB communication in auto response mode, set the ATREPM bit to 1 while the FIFO buffer is empty. Do not write to the FIFO buffer during USB communication in auto response mode.

When the relevant pipe is for isochronous transfer, be sure to set the ATREPM bit to 0.

### INBUFM Bit (Transmit Buffer Monitor)

Indicates the relevant FIFO buffer status when the relevant pipe is in the transmitting direction.

When the relevant pipe is in the transmitting direction (PIPECFG.DIR = 1), the USB module sets the INBUFM bit to 1 when software (or DMAC) completes writing data to at least one FIFO buffer plane.

The USB module sets the INBUFM bit to 0 when the USB module completes transmitting the data from the FIFO buffer plane to which all the data has been written. In double buffer mode (DBLB = 1), the USB module sets the INBUFM bit to 0 when the USB module completes transmitting the data from the two FIFO buffer planes before software (or DMAC) completes writing data to one FIFO buffer plane.

The INBUFM bit indicates the same value as the BSTS bit when the relevant pipe is in the receiving direction (PIPECFG.DIR = 0).

### BSTS Bit (Buffer Status)

Indicates the FIFO buffer status for the relevant pipe.

The meaning of the BSTS bit depends on the settings of the PIPECFG.DIR bit, PIPECFG.BFRE bit, and DnFIFOSEL.DCLRM bit as listed in Table 31.8.

**Table 31.6 USB Module Operation as Determined by the Setting of the PID[1:0] Bits (1/2)**

Bits PID[1:0]	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB Module
00b (NAK)	Bulk or interrupt	Operation does not depend on the setting.	Returns NAK in response to the token from the USB host. For the operation when the ATREPM bit is 1, refer to the description of the ATREPM bit.
	Isochronous	Operation does not depend on the setting.	Returns nothing in response to the token from the USB host.
01b (BUF)	Bulk	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception. Returns NAK if not ready.
	Interrupt	Receiving direction (DIR = 0)	Receives data and returns ACK in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception. Returns NAK if not ready.
	Bulk or interrupt	Transmitting direction (DIR = 1)	Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Returns NAK if not ready.
	Isochronous	Receiving direction (DIR = 0)	Receives data in response to the OUT token from the USB host if the FIFO buffer corresponding to the relevant pipe is ready for reception. Discards data if not ready.
Transmitting direction (DIR = 1)		Transmits data in response to the token from the USB host if the corresponding FIFO buffer is ready for transmission. Transmits the zero-length packet if not ready.	

**Table 31.6 USB Module Operation as Determined by the Setting of the PID[1:0] Bits (2/2)**

Bits PID[1:0]	Transfer Type	Transfer Direction (DIR Bit)	Operation of USB Module
10b (STALL) or 11b (STALL)	Bulk or interrupt	Operation does not depend on the setting.	Returns STALL in response to the token from the USB host.
	Isochronous	Operation does not depend on the setting.	Returns nothing in response to the token from the USB host.

**Table 31.7 Information Cleared by USB Module by Setting ACLRM = 1**

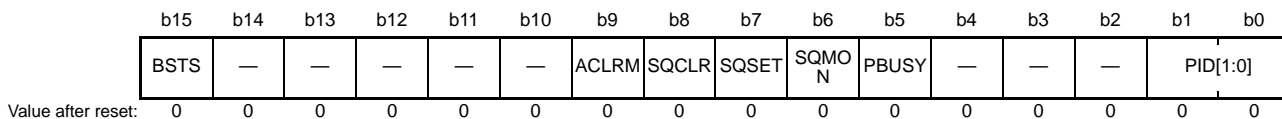
No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the relevant pipe (both FIFO buffer planes are cleared when double buffer mode is selected)	When the pipe is to be initialized
2	The interval count value when the relevant pipe is for isochronous transfer	When the interval count value is to be reset
3	Internal flags concerning the PIPECFG.BFRE bit	When the PIPECFG.BFRE setting is modified
4	FIFO buffer toggle control	When the PIPECFG.DBLB setting is modified
5	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

**Table 31.8 Operation of BSTS Bit**

DIR Bit	BFRE Bit	DCLRM Bit	BSTS Bit Function
0	0	0	1: The received data can be read from the FIFO buffer. 0: The received data has been completely read from the FIFO buffer.
		1	Setting prohibited
	1	0	1: The received data can be read from the FIFO buffer. 0: Software has set the BCLR bit in the port control register to 1 after the received data has been completely read from the FIFO buffer.
		1	1: The received data can be read from the FIFO buffer. 0: The received data has been completely read from the FIFO buffer.
1	0	0	1: The transmit data can be written to the FIFO buffer. 0: The transmit data has been completely written to the FIFO buffer.
		1	Setting prohibited
	1	0	Setting prohibited
		1	Setting prohibited

• PIPEnCTR (n = 6 to 9)

Address(es): USB0.PIPE6CTR 000A 007Ah, USB0.PIPE7CTR 000A 007Ch, USB0.PIPE8CTR 000A 007Eh, USB0.PIPE9CTR 000A 0080h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	PID[1:0]	Response PID	b1 b0 0 0: NAK response 0 1: BUF response (depending on the buffer state) 1 0: STALL response 1 1: STALL response	R/W
b4 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5	PBUSY	Pipe Busy	0: The relevant pipe is not used at the USB bus. 1: The relevant pipe is used at the USB bus.	R
b6	SQMON	Toggle Bit Confirmation	0: DATA0 1: DATA1	R
b7	SQSET	Toggle Bit Set*2	0: Invalid 1: Specifies DATA1.	R/W*1
b8	SQCLR	Toggle Bit Clear*2	0: Invalid 1: Specifies DATA0.	R/W*1
b9	ACLRM	Auto Buffer Clear Mode*2,*3	0: Auto buffer clear mode is disabled. 1: Auto buffer clear mode is enabled (all buffers are initialized)	R/W
b14 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	BSTS	Buffer Status	0: Buffer access is disabled. 1: Buffer access is enabled.	R

Note 1. Only 0 can be read and 1 can be written.

Note 2. Write 1 to the SQCLR or SQSET bit while PID is NAK. Before modifying these bits after modifying the PID[1:0] bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

Note 3. Modify the ACLRM bit while PID is NAK and before the pipe is selected by the CURPIPE[3:0] bits in the port select register. Before modifying this bit after modifying the PID[1:0] bits for the selected pipe from BUF to NAK, check that the PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

**PID[1:0] Bits (Response PID)**

The PID[1:0] bits specify the response type for the next transaction of the relevant pipe.

The default setting of the PID[1:0] bits is NAK. Modify the setting of the PID[1:0] bits to BUF to use the relevant pipe for USB transfer. Table 31.6 lists the basic operation (operation when there are no errors in the transmitted and received packets) of the USB module depending on the PID[1:0] bit setting.

After modifying the setting of the PID[1:0] bits through software from BUF to NAK during USB communication using the relevant pipe, check that the PBUSY bit is 1 to see if USB transfer using the relevant pipe has actually entered the NAK state.

The USB module modifies the setting of the PID[1:0] bits in the following cases.

- The USB module sets PID to STALL (11b) on receiving a data packet with a payload exceeding the maximum packet size of the relevant pipe.
- The USB module sets PID to NAK on detecting a USB bus reset.

To specify each response type, set the PID[1:0] bits as follows.

- To make a transition from NAK (00b) to STALL, set 10.
- To make a transition from BUF (01b) to STALL, set 11.
- To make a transition from STALL (11b) to NAK, set 10 and then 00.
- To make a transition from STALL to BUF, set 00b (NAK) and then 01b (BUF).

### PBUSY Bit (Pipe Busy)

The PBUSY bit indicates whether the relevant pipe is being currently used or not for the USB bus.

The USB module modifies the PBUSY bit from 0 to 1 upon start of the USB transaction for the relevant pipe, and modifies the PBUSY bit from 1 to 0 upon completion of one transaction.

Reading the PBUSY bit after software has set PID to NAK allows checking whether modification of the pipe settings is possible.

### SQMON Bit (Toggle Bit Confirmation)

The SQMON bit indicates the expected value of the sequence toggle bit for the next transaction of the relevant pipe.

The USB module allows the SQMON bit to toggle upon normal completion of the transaction. However, the SQMON bit is not allowed to toggle when a DATA-PID disagreement occurs during the transfer in the receiving direction.

### SQSET Bit (Toggle Bit Set)

The SQSET bit should be set to 1 to set DATA1 as the expected value of the sequence toggle bit for the next transaction of the relevant pipe.

Setting the SQSET bit to 1 through software allows the USB module to set DATA1 as the expected value of the sequence toggle bit of the relevant pipe. The USB module sets the SQSET bit to 0.

### SQCLR Bit (Toggle Bit Clear)

The SQCLR bit should be set to 1 to clear the expected value of the sequence toggle bit for the next transaction of the relevant pipe to DATA0.

Setting the SQCLR bit to 1 through software allows the USB module to set DATA0 as the expected value of the sequence toggle bit of the relevant pipe. The USB module sets the SQCLR bit to 0.

### ACLRM Bit (Auto Buffer Clear Mode)

Enables or disables auto buffer clear mode for the relevant pipe.

To delete the information in the FIFO buffer assigned to the relevant pipe completely, write 1 and then 0 to the ACLRM bit continuously.

Table 31.9 lists the information cleared by writing 1 and 0 to the ACLRM bit continuously and the cases in which clearing the information is necessary.

### BSTS Bit (Buffer Status)

Indicates the FIFO buffer status for the relevant pipe.

The meaning of the BSTS bit depends on the settings of the PIPRCFG.DIR, PIPRCFG.BFRE, and DnFIFOSEL.DCLRM bits as listed in Table 31.8.

**Table 31.9 Information Cleared by USB Module by Setting ACLRM = 1**

No.	Information Cleared by ACLRM Bit Manipulation	Cases in which Clearing Information is Necessary
1	All the information in the FIFO buffer assigned to the selected pipe	When the pipe is to be initialized
2	Internal flags concerning the PIPECFG.BFRE bit	When the PIPECFG.BFRE setting is modified
3	Internal flags concerning the transaction count	When the transaction count function is forcibly terminated

### 31.2.30 PIPEn Transaction Counter Enable Registers (PIPEnTRE) (n = 1 to 5)

Address(es): USB0.PIPE1TRE 000A 0090h, USB0.PIPE2TRE 000A 0094h, USB0.PIPE3TRE 000A 0098h,  
USB0.PIPE4TRE 000A 009Ch, USB0.PIPE5TRE 000A 00A0h

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	TRENB	TRCLR	—	—	—	—	—	—	—	—
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	TRCLR	Transaction Counter Clear	0: Invalid 1: The current counter value is cleared.	R/W
b9	TRENB	Transaction Counter Enable	0: Transaction counter is disabled. 1: Transaction counter is enabled.	R/W
b15 to b10	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note: • Modify each bit in PIPEnTRE while PID is NAK. Before modifying these bits after modifying the PIPEnCTR.PID[1:0] bits for the selected pipe from BUF to NAK, check that the PIPEnCTR.PBUSY bit is 0. However, if the PID[1:0] bits have been modified to NAK by the USB module, checking the PBUSY bit through software is not necessary.

#### TRCLR Bit (Transaction Counter Clear)

Clears the current value of the transaction counter corresponding to the relevant pipe and then sets the TRCLR bit to 0.

#### TRENB Bit (Transaction Counter Enable)

Enables or disables the transaction counter.

For the pipe in the receiving direction, setting the TRENB bit to 1 after setting the total number of the packets to be received in the PIPEnTRN.TRNCNT[15:0] bits through software allows the USB module to control hardware as described below on having received the number of packets equal to the setting of the PIPEnTRN.TRNCNT[15:0] bits.

- While the PIPECFG.SHTNAK bit is 1, the USB module modifies the PID[1:0] bits to NAK for the corresponding pipe on having received the number of packets equal to the setting of the PIPEnTRN.TRNCNT[15:0] bits.
- While the PIPECFG.BFRE bit is 1, the USB module asserts the BRDY interrupt on having received the number of packets equal to the setting of the PIPEnTRN.TRNCNT[15:0] bits and then reading out the last received data.

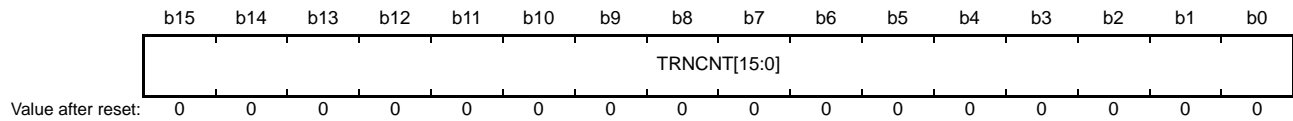
For the pipe in the transmitting direction, set the TRENB bit to 0.

When the transaction counter is not used, set the TRENB bit to 0.

When the transaction counter is used, set the TRNCNT[15:0] bits before setting the TRENB bit to 1. Set the TRENB bit to 1 before receiving the first packet to be counted by the transaction counter.

### 31.2.31 PIPE<sub>n</sub> Transaction Counter Registers (PIPE<sub>n</sub>TRN) (n = 1 to 5)

Address(es): USB0.PIPE1TRN 000A 0092h, USB0.PIPE2TRN 000A 0096h, USB0.PIPE3TRN 000A 009Ah,  
USB0.PIPE4TRN 000A 009Eh, USB0.PIPE5TRN 000A 00A2h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	TRNCNT[15:0]	Transaction Counter	<ul style="list-style-type: none"> <li>When written to Specifies the number of transactions to be transferred through DMA.</li> <li>When read from Indicates the specified number of transactions if the PIPE<sub>n</sub>TRE.TRENB bit is 0. Indicates the number of currently counted transactions if the PIPE<sub>n</sub>TRE.TRENB bit is 1.</li> </ul>	R/W

PIPE<sub>n</sub>TRN retains the setting by a USB bus reset.

#### TRNCNT[15:0] Bits (Transaction Counter)

The USB module increments the value of the TRNCNT[15:0] bits by one when all of the following conditions are satisfied on receiving the packet.

- PIPE<sub>n</sub>TRE.TRENB = 1
- (TRNCNT[15:0] set value ≠ current counter value + 1) on receiving the packet.
- The payload of the received packet agrees with the setting of the PIPEMAXP.MXPS[8:0] bits.

The USB module clears the value of the TRNCNT[15:0] bits to 0 when any of the following conditions are satisfied.

(1) All of the following conditions are satisfied.

- PIPE<sub>n</sub>TRE.TRENB bit = 1
- (TRNCNT[15:0] set value = current counter value + 1) on receiving the packet.
- The payload of the received packet agrees with the setting of the PIPEMAXP.MXPS[8:0] bits.

(2) All of the following conditions are satisfied.

- PIPE<sub>n</sub>TRE.TRENB bit = 1
- The USB module has received a short packet.

(3) All of the following conditions are satisfied.

- PIPE<sub>n</sub>TRE.TRENB bit = 1
- Software has set the PIPE<sub>n</sub>TRE.TRCLR bit to 1.

For the pipe in the transmitting direction, set the TRNCNT[15:0] bits to 0.

When the transaction counter is not used, set the TRNCNT[15:0] bits to 0.

Modify the TRNCNT[15:0] bits while the PIPE<sub>n</sub>TRE.TRENB bit is set to 0.

To modify the value of the TRNCNT[15:0] bits, set the PIPE<sub>n</sub>TRE.TRCLR bit to 1 before setting the PIPE<sub>n</sub>TRE.TRENB bit to 1.

### 31.2.32 Deep Standby USB Transceiver Control/Pin Monitor Register (DPUSR0R)

Address(es): 000A 0400h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DVBST S0	—	—	—	—	—	DM0	DP0
Value after reset:	x	0	x	x	0	0	x	x	x	0	x	x	0	0	x	x
	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	FIXPH Y0	—	—	—	SRPC0
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	SRPC0	USB0 Single End Receiver Control	0: DP and DM inputs are disabled. 1: DP and DM inputs are enabled.	R/W
b3 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	FIXPHY0	USB0 Transceiver Output Fix	0: The outputs are fixed during normal mode or on return from deep software standby mode. 1: The outputs are fixed on making a transition to deep software standby mode.	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	DP0	USB0 DP Input	DP input signal on the USB port 0 side is indicated.	R
b17	DM0	USB0 DM Input	DM input signal on the USB port 0 side is indicated.	R
b19, b18	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b21, b20	—	Reserved	These bits are read as undefined. Writing to these bits has no effect.	R
b22	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b23	DVBSTS0	USB0 VBUS Input	VBUS input signal on the USB port 0 side is indicated.	R
b25, b24	—	Reserved	These bits are read as undefined. Writing to these bits has no effect.	R
b27, b26	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b29, b28	—	Reserved	These bits are read as undefined. Writing to these bits has no effect.	R
b30	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b31	—	Reserved	This bit is read as undefined. Writing to this bit has no effect.	R

#### SRPC0 Bit (USB0 Single End Receiver Control)

The SRPC0 bit controls the DP and DM inputs of the transceiver on the USB port 0 side.

This bit is valid only when the FIXPHY0 bit is 1.

#### FIXPHY0 Bit (USB0 Transceiver Output Fix)

The FIXPHY0 bit holds the outputs to the transceiver on the USB port 0 side disabled.

### 31.2.33 Deep Standby USB Suspend/Resume Interrupt Register (DPUSR1R)

Address(es): 000A 0404h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	—	—	—	—	—	—	—	—	DVBINT0	—	—	—	—	—	—	DPINT0
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
	—	—	—	—	—	—	—	—	DVBSE0	—	—	—	—	—	—	DPINTE0
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	DPINTE0	USB0 DP Interrupt Enable/Clear	0: Recovery from deep software standby mode is disabled. 1: Recovery from deep software standby mode is enabled.	R/W
b6 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	DVBSE0	USB0 VBUS Interrupt Enable/Clear	0: Recovery from deep software standby mode is disabled. 1: Recovery from deep software standby mode is enabled.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b16	DPINT0	USB0 DP Interrupt Source Recovery	0: The system has not returned from deep software standby mode. 1: The system has returned from deep software standby mode.	R
b22 to b17	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b23	DVBINT0	USB0 VBUS Interrupt Source Recovery	0: The system has not returned from deep software standby mode. 1: The system has returned from deep software standby mode.	R
b31 to b24	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

#### DPINTE0 Bit (USB0 DP Interrupt Enable/Clear)

The DPINTE0 bit enables or disables a recovery from deep software standby mode by the DP input in the USB port 0 side. When 0 is written to this bit with the DPINT0 bit set to 1, DPINT0 is cleared to 0.

#### DVBSE0 Bit (USB0 VBUS Interrupt Enable/Clear)

The DVBSE0 bit enables or disables a recovery from deep software standby mode by the VBUS input in the USB port 0 side. When 0 is written to this bit with the DVBINT0 bit set to 1, DVBINT0 bit is cleared to 0.

#### DPINT0 Bit (USB0 DP Interrupt Source Recovery)

The DPINT0 bit indicates that the system has returned from deep software standby mode by the DP input in the USB port 0 side. A recovery from deep software standby mode by USB0 DP input is enabled only when DPINTE0 bit is 1. This bit is cleared to 0 when 0 is written to DPINTE0 bit with this bit set to 1.

#### DVBINT0 Bit (USB0 VBUS Interrupt Source Recovery)

The DVBINT0 bit indicates that the system has returned from deep software standby mode by the VBUS input in the USB port 0 side. A recovery from deep software standby mode by USB0 VBUS input is enabled only when DVBSE0 bit is 1.

This bit is cleared to 0 when 0 is written to DVBSE0 bit with this bit set to 1.



### 31.3 Operation

#### 31.3.1 System Control

This section describes the register settings that are necessary for initialization of this module and power consumption control.

##### 31.3.1.1 Starting Operation

Setting the SYSCFG.USB\_E bit to 1 after starting the clock supply to the USB module (SYSCFG.SCKE = 1) enables and starts USB module operation.

Figure 31.2 shows an example of peripheral connection of the USB connector (USB0) in the self-powered state.

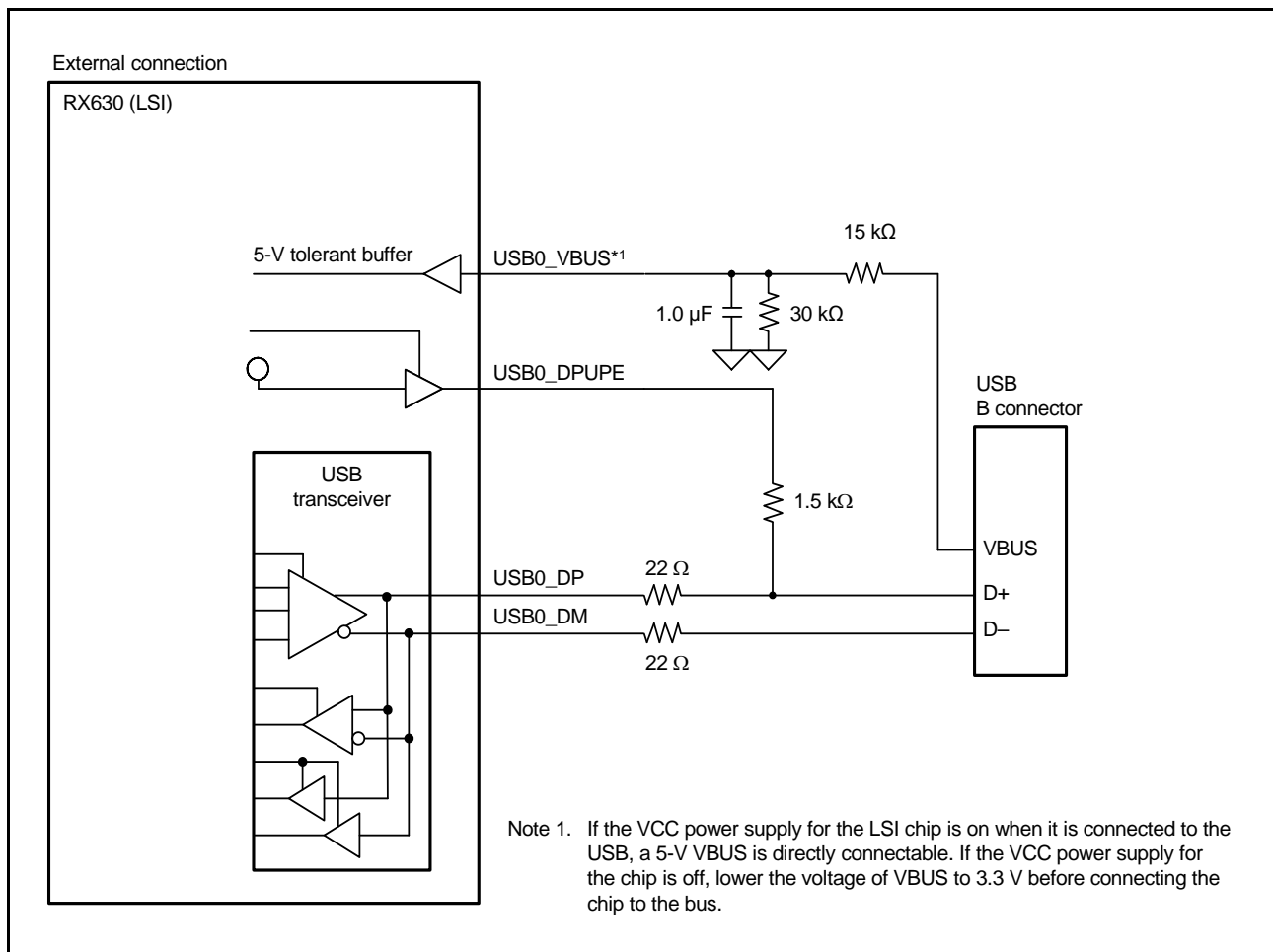


Figure 31.2 Sample Peripheral Connection of USB Connector (USB0) in Self-Powered State

Figure 31.3 shows an example of peripheral connection of the USB connector (USB0) in the bus-powered state.

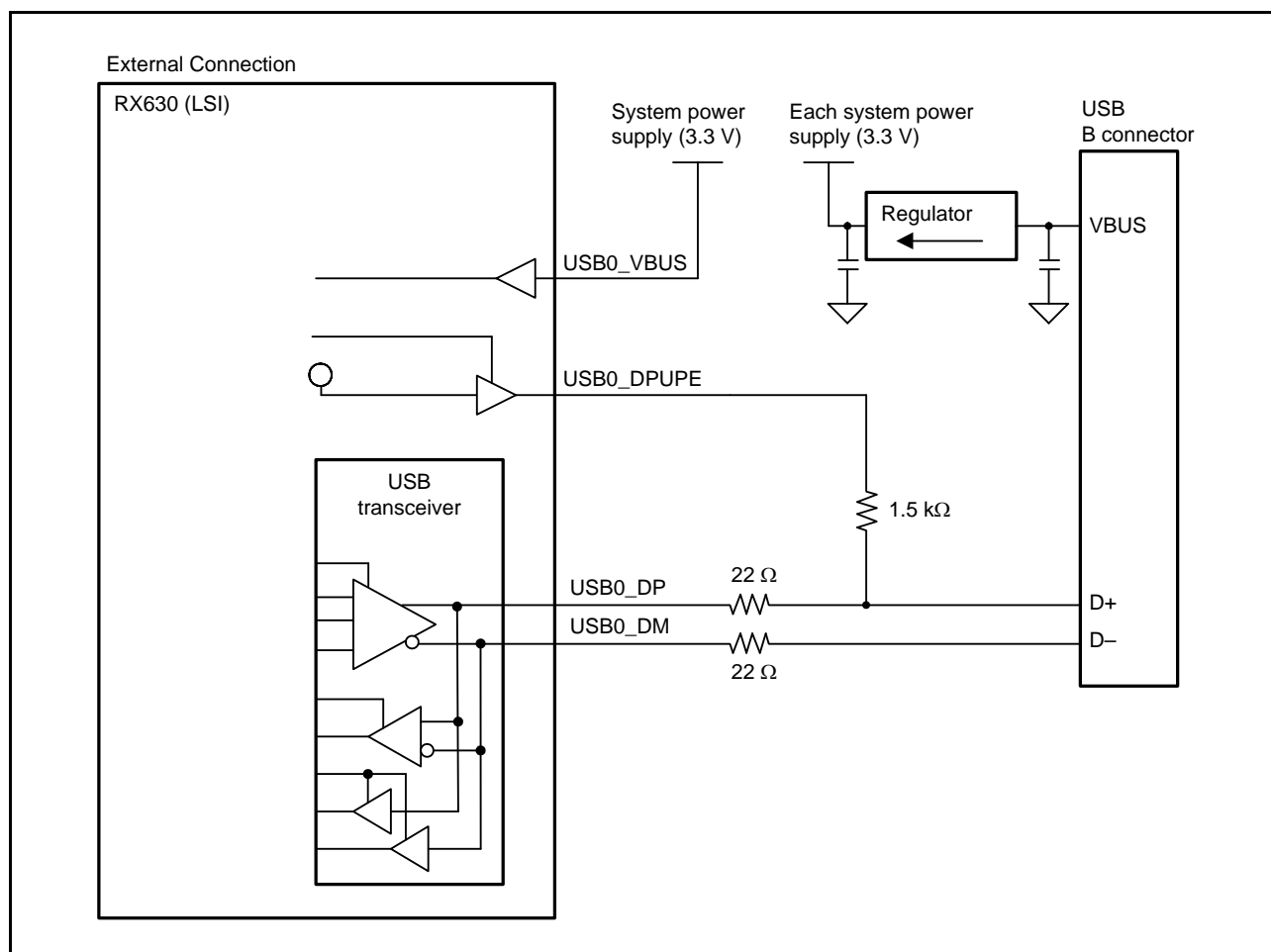


Figure 31.3 Sample Peripheral Connection of USB Connector (USB0) in Bus-Powered State

The examples of external circuits described in this section are only simplified circuits, and their operations are not guaranteed under any system.

### 31.3.1.2 Canceling Deep Software Standby Mode by a USB Suspend/Resume Interrupt

Deep software standby mode can be canceled by a USB suspend/resume interrupt. A USB suspend/resume interrupt is detected at the USB resume detecting unit. The USB resume detecting unit controls and monitors the I/O pins for USB0 to detect USB suspend/resume interrupts.

Figure 31.4 shows a schematic diagram of connection between the USB resume detecting unit and the I/O pins for USB0.

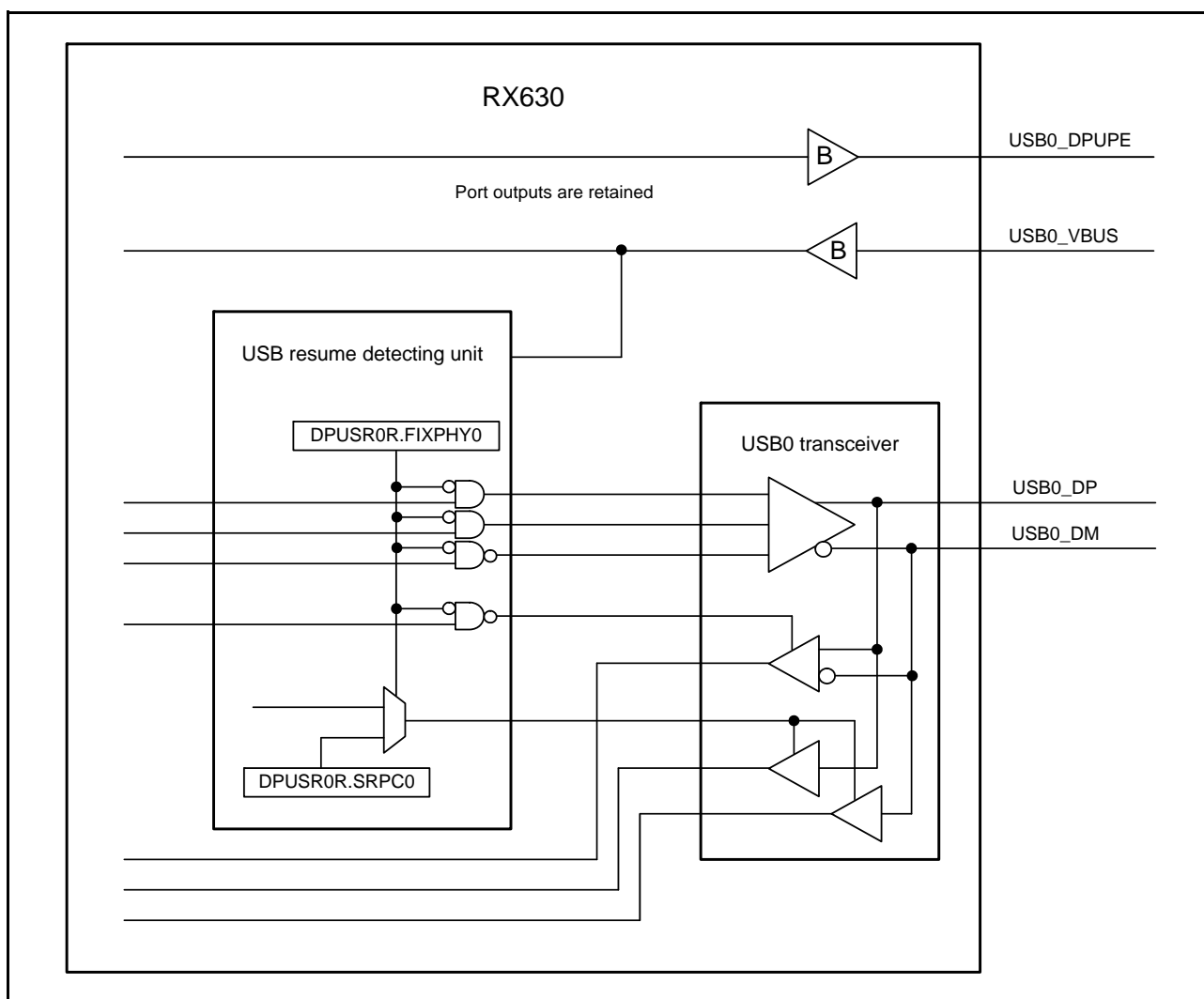


Figure 31.4 Overview of Connection between USB Resume Detecting Unit and USB0 I/O Pins

Table 31.10 shows the USB suspend/resume interrupt sources and their corresponding I/O pins.

Table 31.10 USB Suspend/Resume Interrupt Sources and Corresponding I/O Pins

Source	Pin Name
Resume	USB0_DP
Connection/Disconnection	USB0_VBUS

When canceling deep software standby mode using a USB suspend/resume interrupt, set the DPSBYCR.IOKEEP bit to have the outputs of the I/O ports retained.

Figure 31.5 shows a flowchart for setting the USB when entering deep software standby mode. Figure 31.6 shows a flowchart for setting the USB when canceling deep software standby mode.

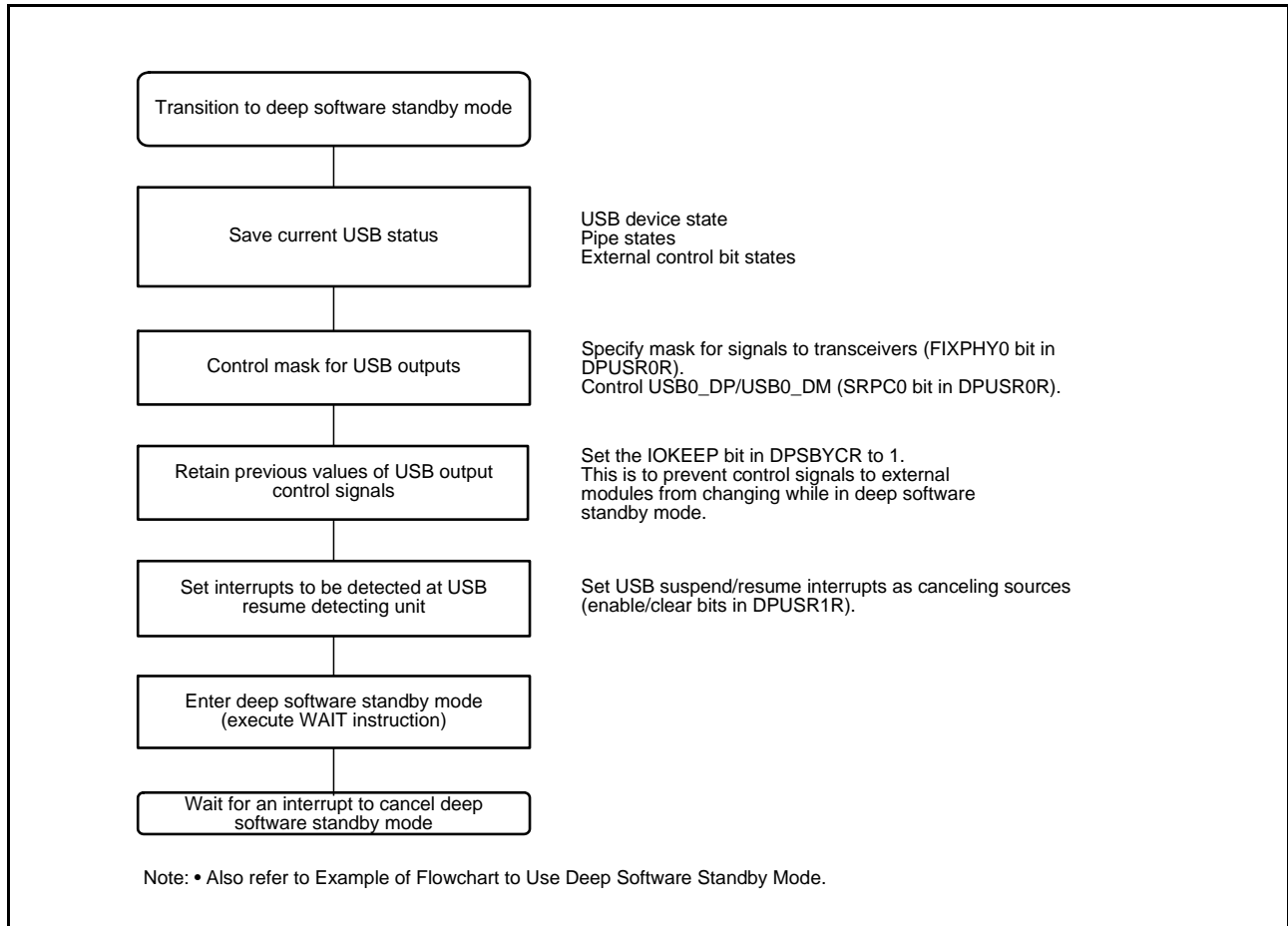


Figure 31.5 USB Setting Flowchart for Transition to Deep Software Standby Mode

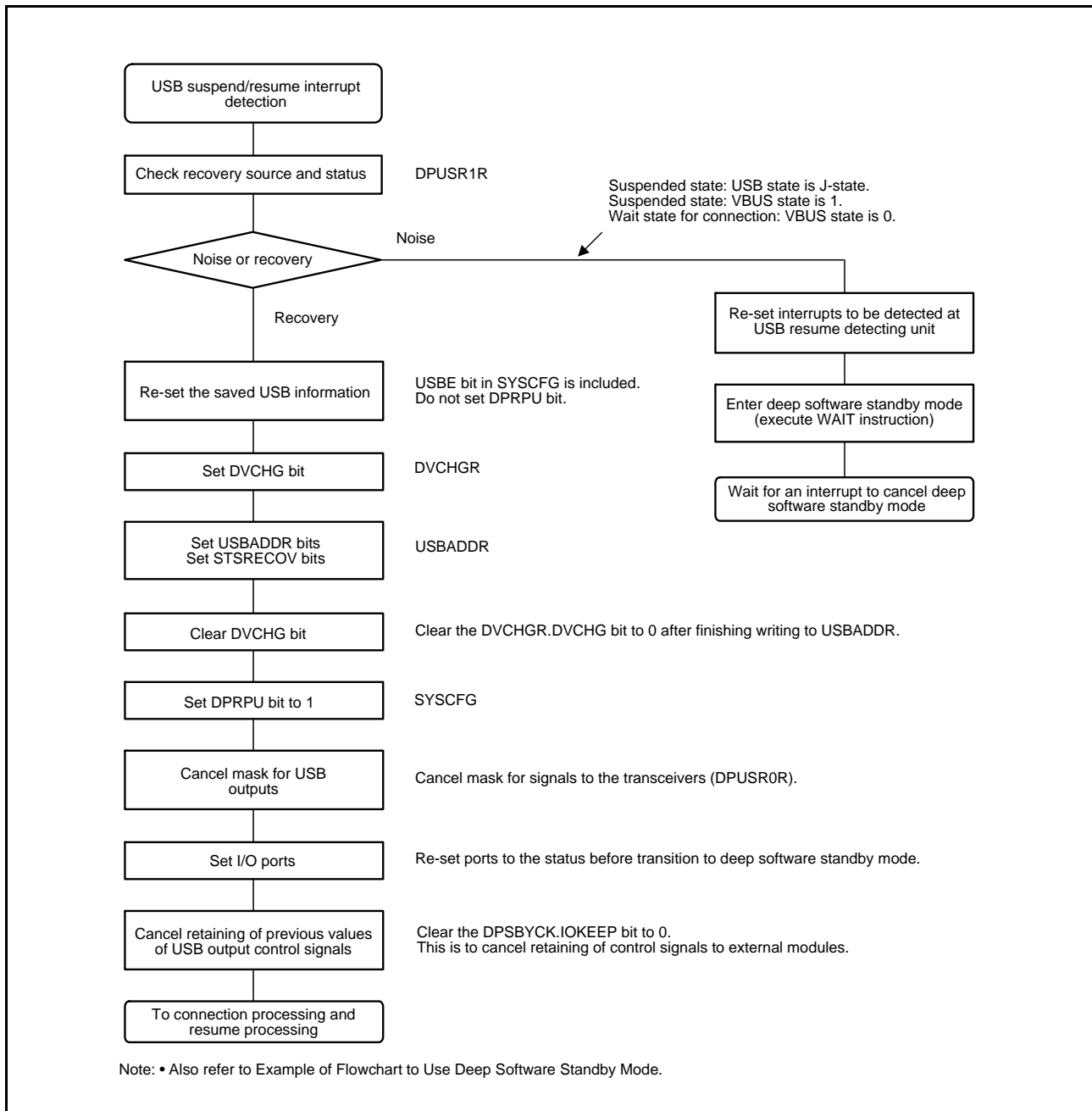


Figure 31.6 USB Setting Flowchart for Canceling Deep Software Standby Mode

### 31.3.2 Interrupt Sources

Table 31.11 lists the interrupt sources in the USB module.

When an interrupt generation condition is satisfied and the interrupt output is enabled using the corresponding interrupt enable register, the USB issues a USB interrupt request to the interrupt controller (ICU) and an USB interrupt will be generated.

**Table 31.11 Interrupt Sources**

Bit to be Set	Name	Interrupt Source	Status Flag
VBINT	VBUS interrupt	<ul style="list-style-type: none"> <li>When a change in the state of the USB0_VBUS input pin has been detected (low to high or high to low)</li> </ul>	INTSTS0.VB STS
RESM	Resume interrupt	<ul style="list-style-type: none"> <li>When a change in the state of the USB bus has been detected in the suspended state (J-state to K-state or J-state to SE0)</li> </ul>	—
SOFR	Frame number update interrupt	<ul style="list-style-type: none"> <li>When an SOF packet with a different frame number has been received</li> </ul>	—
DVST	Device state transition interrupt	<ul style="list-style-type: none"> <li>When a device state transition has been detected (any of the following conditions)               <ul style="list-style-type: none"> <li>A USB bus reset detected</li> <li>Suspend state detected</li> <li>SET_ADDRESS request received</li> <li>SET_CONFIGURATION request received</li> </ul> </li> </ul>	INTSTS0.DV SQ[2:0]
CTRT	Control transfer stage transition interrupt	<ul style="list-style-type: none"> <li>When a stage transition has been detected in control transfer (any of the following conditions)               <ul style="list-style-type: none"> <li>Setup stage completed</li> <li>Control write transfer status stage transition</li> <li>Control read transfer status stage transition</li> <li>Control transfer completed</li> <li>A control transfer sequence error occurred</li> </ul> </li> </ul>	INTSTS0.CT SQ[2:0]
BEMP	Buffer empty interrupt	<ul style="list-style-type: none"> <li>When transmission of all data in the buffer memory has been completed and the buffer has become empty</li> <li>When a packet larger than the maximum packet size has been received</li> </ul>	BEMPSTS. PIPEnBEMP
NRDY	Buffer not ready interrupt	<ul style="list-style-type: none"> <li>When NAK has been returned for an IN or OUT token with the DCPCTR.PID[1:0] bits = BUF.</li> <li>When a CRC error or a bit stuffing error occurred during data reception in isochronous transfer</li> <li>When an overrun/underrun occurred during data reception in isochronous transfer</li> </ul>	NRDYSTS. PIPEnNRDY
BRDY	Buffer ready interrupt	<ul style="list-style-type: none"> <li>When the buffer has become ready (reading or writing is enabled)</li> </ul>	BRDYSTS. PIPEnBRDY

Note: • All bits without register name are in INTSTS0.

Figure 31.7 shows the circuits related to the interrupts in the USB0.

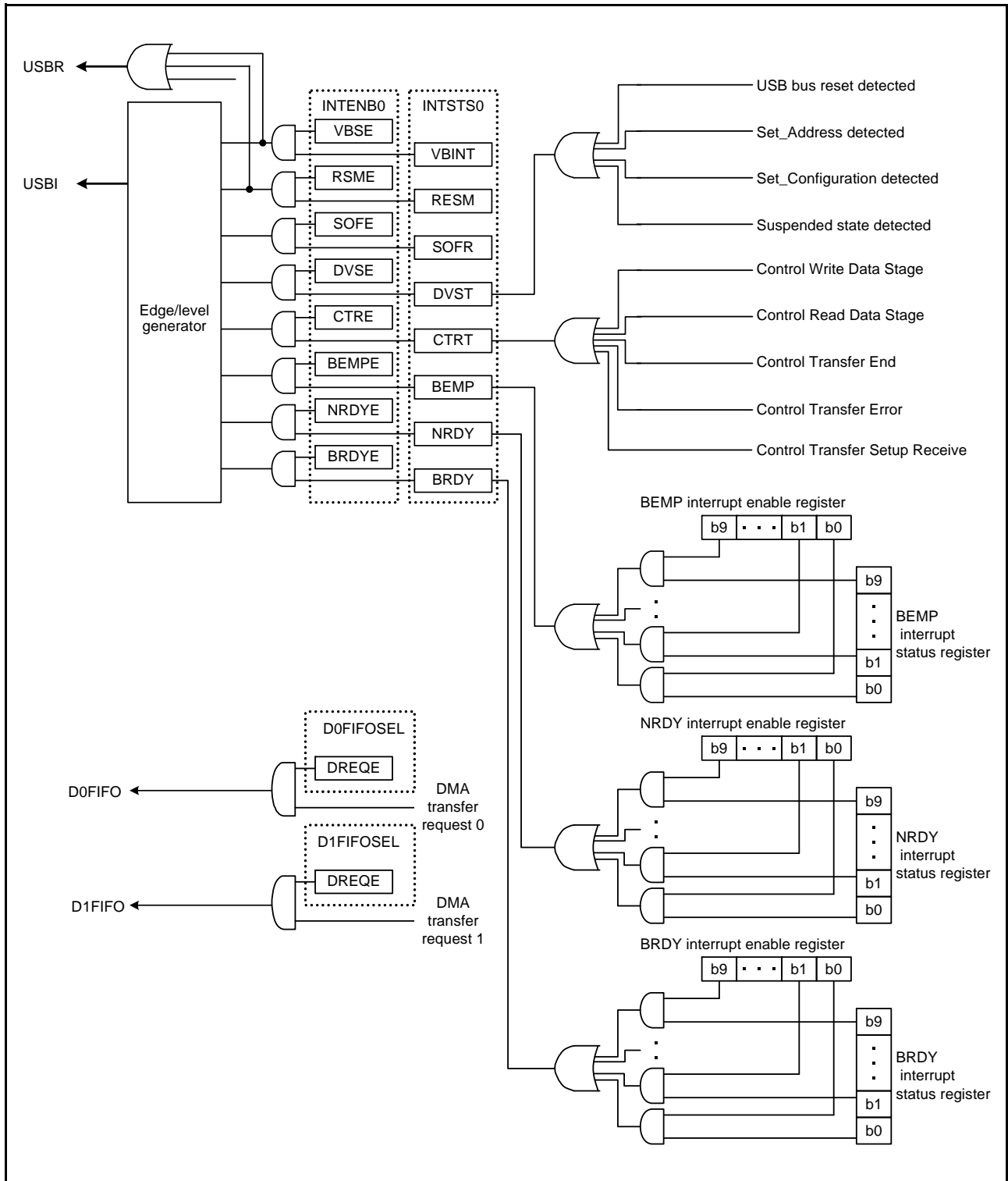


Figure 31.7 Circuits Related to Interrupts

Table 31.12 lists the interrupts generated in the USB0.

**Table 31.12 USB0 Interrupts**

Interrupt Name	Interrupt Flag	DTC Activation	DMAC Activation	Priority
D0FIFO	DMA transfer request 0	Possible	Possible	High
D1FIFO	DMA transfer request 1	Possible	Possible	↑ Low
USBI	VBUS interrupt, resume interrupt, frame number update interrupt, device state transition interrupt, control transfer stage transition interrupt, buffer empty interrupt, buffer not ready interrupt, buffer ready interrupt, bus change interrupt, and disconnection detection during full-speed operation	Not possible	Not possible	
USBR	VBUS interrupt, resume interrupt, and bus change interrupt	Not possible	Not possible	—

### 31.3.3 Interrupt Descriptions

#### 31.3.3.1 BRDY Interrupt

The following shows the conditions under which the USB module sets 1 to a corresponding bit in BRDYSTS. Under this condition, the USB module generates a BRDY interrupt if software has set 1 to the BRDYENB.PIPEnBRDYE bit that corresponds to the pipe and 1 to the INTENB0.BRDYE bit.

The conditions for generating and clearing the BRDY interrupt depend on the settings of the SOFCFG.BRDYM bit and PIPECFG.BFRE bit for each pipe as described below.

##### (1) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 0

With these settings, the BRDY interrupt indicates that the FIFO port is accessible.

On any of the following conditions, the USB module generates an internal BRDY interrupt request trigger and sets 1 to the BRDYSTS.PIPEBRDY bit corresponding to the pertinent pipe.

For the pipe in the transmitting direction:

- When software changes the DIR bit from 0 to 1.
- When packet transmission is completed using the pertinent pipe while write-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the PIPEnCTR.BSTS bit is read as 0).
- When one FIFO buffer is empty on completion of writing data to the other FIFO buffer in double buffer mode.
- No request trigger is generated until completion of writing data to the currently-written FIFO buffer even if transmission to the other FIFO buffer is completed.
- When the hardware flushes the buffer of the pipe for isochronous transfers.
- When 1 is written to the PIPEnCTR.ACLRM bit, which causes the FIFO buffer to make transition from the write-disabled to write-enabled state.

No request trigger is generated for the DCP (that is, during data transmission for control transfers).



For the pipe in the receiving direction:

- When packet reception is completed successfully thus enabling the FIFO buffer to be read while read-access from the CPU to the FIFO buffer for the pertinent pipe is disabled (when the PIPEnCTR.BSTS bit is read as 0). No request trigger is generated for the transaction in which DATA-PID disagreement has occurred.
- When one FIFO buffer is read-enabled on completion of reading data from the other FIFO buffer in double buffer mode. No request trigger is generated until completion of reading data from the currently-read FIFO buffer even if reception by the other FIFO buffer is completed.

The BRDY interrupt is not generated in the status stage of control transfers.

The PIPEBRDY interrupt status of the pertinent pipe can be cleared to 0 by writing 0 to the corresponding BRDYSTS.PIPEnBRDY bit through software. In this case, 1 should be written to the PIPEnBRDY bits for the other pipes.

Be sure to clear the BRDY status before accessing the FIFO buffer.

### (2) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 1

With these settings, the USB module generates a BRDY interrupt on completion of reading all data for a single transfer using the pipe in the receiving direction, and sets 1 to the bit in BRDYSTS corresponding to the pertinent pipe. On any of the following conditions, the USB module determines that the last data for a single transfer has been received.

- When a short packet including a zero-length packet is received.
- When the transaction counter register (PIPEnTRN.TRNCNT[15:0] bits) is used and the number of packets specified by the TRNCNT[51:0] bits are completely received.

When the pertinent data is completely read after any of the above conditions has been satisfied, the USB module determines that all data for a single transfer has been completely read.

When a zero-length packet is received while the FIFO buffer is empty, the USB module determines that all data for a single transfer has been completely read upon passing the zero-length packet data to the CPU. In this case, to start the next transfer, write 1 to the BCLR bit in the corresponding port control register through software.

With these settings, the USB module does not detect a BRDY interrupt for the pipe in the transmitting direction.

The PIPEBRDY interrupt status of the pertinent pipe can be cleared to 0 by writing 0 to the corresponding BRDYSTS.PIPEBRDY bit through software. In this case, 1 should be written to the PIPEBRDY bits for the other pipes. In this mode, the PIPECFG.BFRE bit setting should not be modified until all data for a single transfer has been processed.

When it is necessary to modify the PIPECFG.BFRE bit before completion of processing, all FIFO buffers for the pertinent pipe should be cleared using the PIPEnCTR.ACLRM bit.

### (3) When SOFCFG.BRDYM = 1 and PIPECFG.BFRE = 0

With these settings, the values of the BRDYSTS.PIPEBRDY are linked to the PIPEnCTR.BSTS bit setting for each pipe. In other words, the BRDY interrupt status bits are set to 1 or 0 by the USB module depending on the FIFO buffer status.

- For the pipe in the transmitting direction:  
The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for write access, and are set to 0 when it is not ready.  
However, the BRDY interrupt is not generated even if the DCP in the transmitting direction is ready for write access.

- For the pipe in the receiving direction:  
 The BRDY interrupt status bits are set to 1 when the FIFO buffer is ready for read access, and are set to 0 when all data have been read (not ready for read access).  
 When a zero-length packet is received while the FIFO buffer is empty, the pertinent bit is set to 1 and the BRDY interrupt is continuously generated until the CFIFOCTR, D0FIFOCTR or D1FIFOCTR.BCLR = 1 is written through software.  
 With this setting, the BRDYSTS.PIPEBRDY bit cannot be cleared to 0 through software.  
 When the SOFCFG.BRDYM is set to 1, all the PIPECFG.BFRE bits (for all pipes) should be cleared to 0.

Figure 31.8 shows the timing of BRDY interrupt generation.

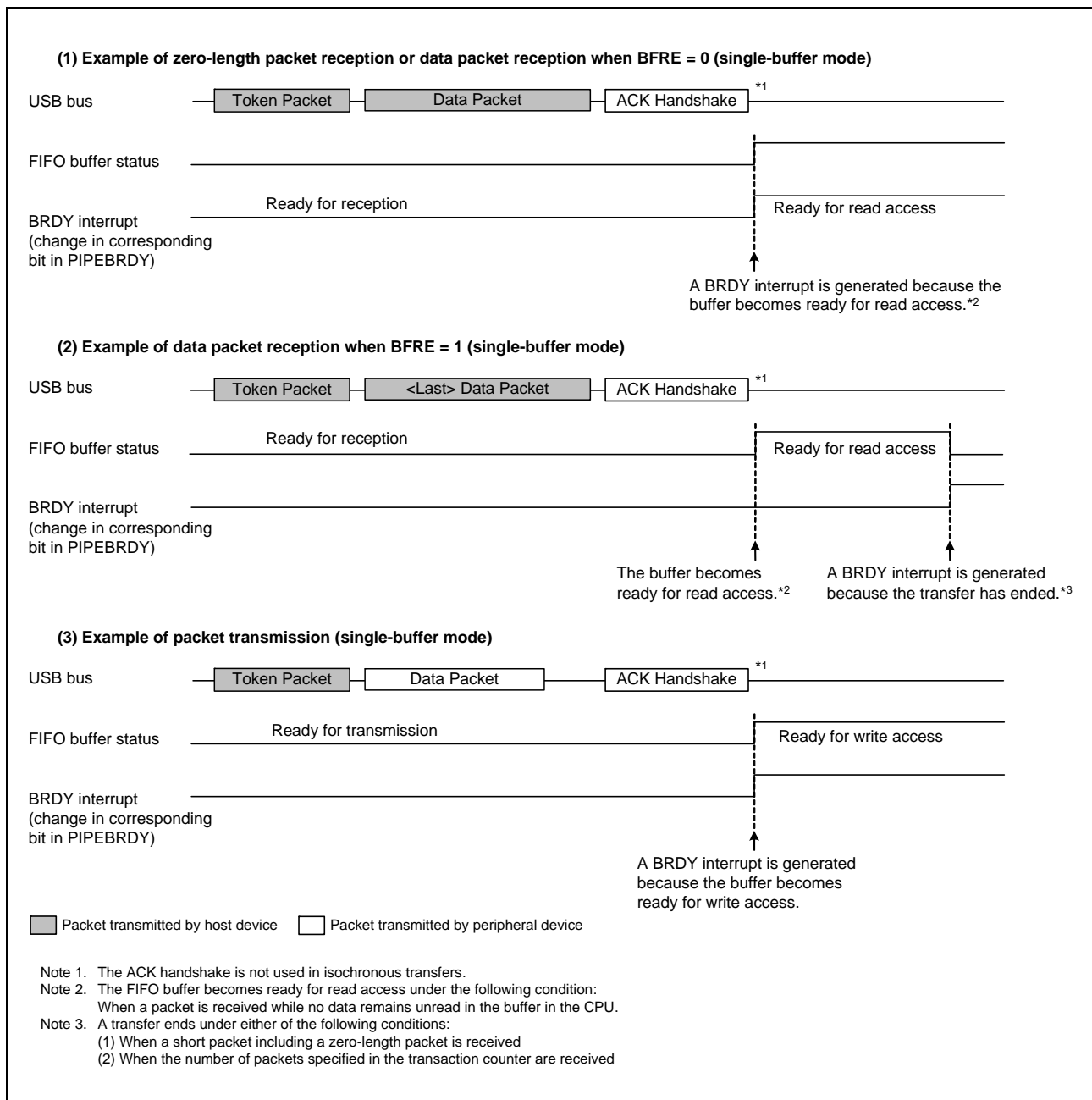


Figure 31.8 Timing of BRDY Interrupt Generation

The condition that USB module clears the INTSTS0.BRDY bit depends on the SOFCFG.BRDYM bit setting. Table 31.13 lists the condition for clearing the BRDY bit.

**Table 31.13 Condition for Clearing BRDY Bit**

BRDYM	Condition for Clearing BRDY Bit
0	The USB module clears the BRDY bit to 0 when software has cleared all bits in BRDYSTS.
1	The USB module clears the BRDY bit to 0 when the BSTS bits for all pipes have become 0.

### 31.3.3.2 NRDY Interrupt

On generating an internal NRDY interrupt request for the pipe whose PID bits are set to BUF by software, the USB module sets the corresponding PIPENRDY bit in NRDYSTS to 1. If the corresponding bit in NRDYENB has been set to 1 by software, the USB module sets the INTSTS0.NRDY bit to 1 and generates a USB interrupt.

The following describes the conditions on which the USB module generates the internal NRDY interrupt request for a given pipe.

The internal NRDY interrupt request is not generated during status stage execution of the control transfer.

#### For the Pipe in the Transmitting Direction:

- When an IN token is received while there is no data to be transmitted in the FIFO buffer.  
In this case, the USB module generates a NRDY interrupt request at the reception of the IN token and sets the NRDYSTS.PIPENRDY bit to 1.  
For the pipe for the isochronous transfers in which an interrupt is generated, the USB module transmits a zero-length packet and sets the FRMNUM.OVRN bit to 1.

#### For the Pipe in the Receiving Direction:

- When an OUT token is received while there is no space available in the FIFO buffer.  
For the pipe for the isochronous transfers in which an interrupt is generated, the USB module generates a NRDY interrupt request at the reception of the OUT token and sets the NRDYSTS.PIPENRDY bit to 1 and the FRMNUM.OVRN bit to 1.  
For the pipe for the transfers other than isochronous transfers in which an interrupt is generated, the USB module generates a NRDY interrupt request when a NAK handshake is transferred after the data following the OUT token is received, and sets the NRDYSTS.PIPENRDY bit to 1.  
However, during re-transmission (due to DATA-PID disagreement), the NRDY interrupt request is not generated. In addition, if an error occurs in the DATA packet, the NRDY interrupt request is not generated.
- For the pipe for isochronous transfers, when a token is not received successfully within an interval frame.  
In this case, the USB module generates a NRDY interrupt request when SOF is received, and sets the NRDYSTS.PIPENRDY bit to 1.

Figure 31.9 shows the timing of NRDY interrupt generation.

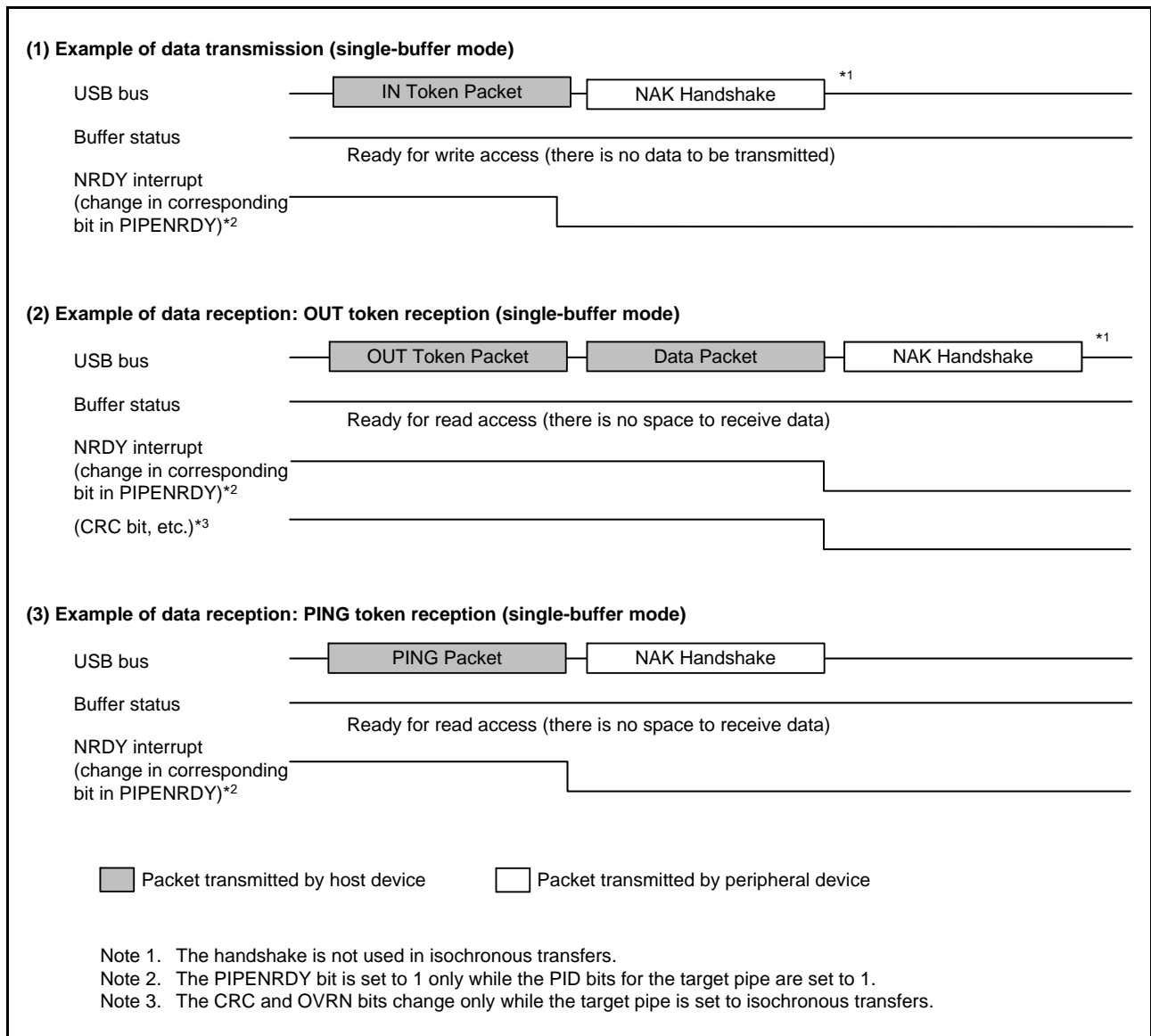


Figure 31.9 Timing of NRDY Interrupt Generation

### 31.3.3.3 BEMP Interrupt

On detecting a BEMP interrupt for the pipe whose PID bits are set to BUF by software, the USB module sets the corresponding BEMPSTS.PIPEnBEMP bit to 1. If the corresponding bit in BEMPENB has been set to 1 by software, the USB module sets the INTSTS0.BEMP bit to 1 and generates a USB interrupt.

The following describes the conditions on which the USB module generates an internal BEMP interrupt request.

#### For the Pipe in the Transmitting Direction:

When the FIFO buffer of the corresponding pipe is empty on completion of transmission (including zero-length packet transmission).

In single buffer mode, an internal BEMP interrupt request is generated simultaneously with the BRDY interrupt for the pipe other than DCP. However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When software (DMAC) has already started writing data to the FIFO buffer of the CPU on completion of transmitting data from one FIFO buffer in double buffer mode.
- When the buffer is cleared (emptied) by setting the PIPEnCTR.ACLRM or the BCLR bit in the port control register to 1.
- When IN transfer (zero-length packet transmission) is performed during the control transfer status stage.

#### For the Pipe in the Receiving Direction:

When the successfully-received data packet size exceeds the specified maximum packet size.

In this case, the USB module generates a BEMP interrupt request, sets the corresponding BEMPSTS.PIPEnBEMP bit to 1, discards the received data, and modifies the setting of the PID[1:0] bits of the corresponding pipe to STALL (11b). Here, the USB module returns STALL response.

However, the internal BEMP interrupt request is not generated on any of the following conditions.

- When a CRC error or a bit stuffing error is detected in the received data.
- When a setup transaction is being performed,  
Writing 0 to the BEMPSTS.PIPEnBEMP bit clears the status.  
Writing 1 to the BEMPSTS.PIPEnBEMP bit has no effect.

Figure 31.10 shows the timing of BEMP interrupt generation.

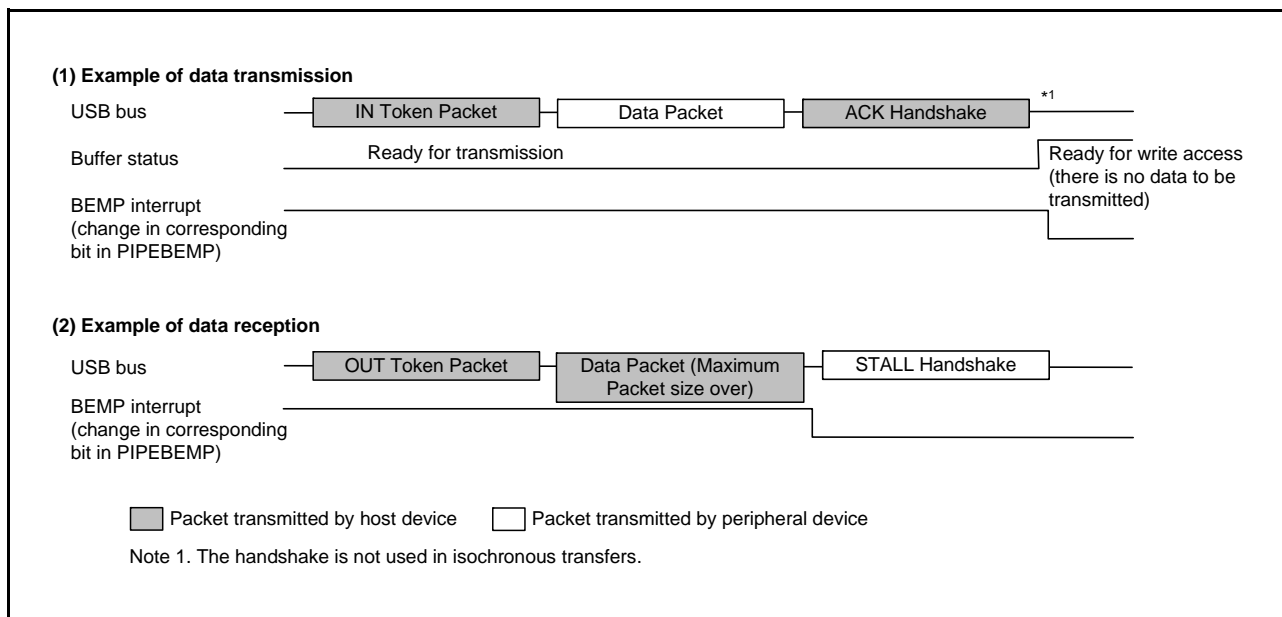


Figure 31.10 Timing of BEMP Interrupt Generation

### 31.3.3.4 Device State Transition Interrupt

Figure 31.11 is a diagram of device state transitions in the USB module. The USB module controls device state and generates device state transition interrupts. However, recovery from the suspended state (resume signal detection) is detected by means of the resume interrupt. The device state transition interrupts can be enabled or disabled individually using INTENB0. The device state to which a transition was made can be confirmed reading the DVSQ[2:0] bits in INTSTS0.

When a transition is made to the default state, a device state transition interrupt is generated after a USB bus reset is detected.

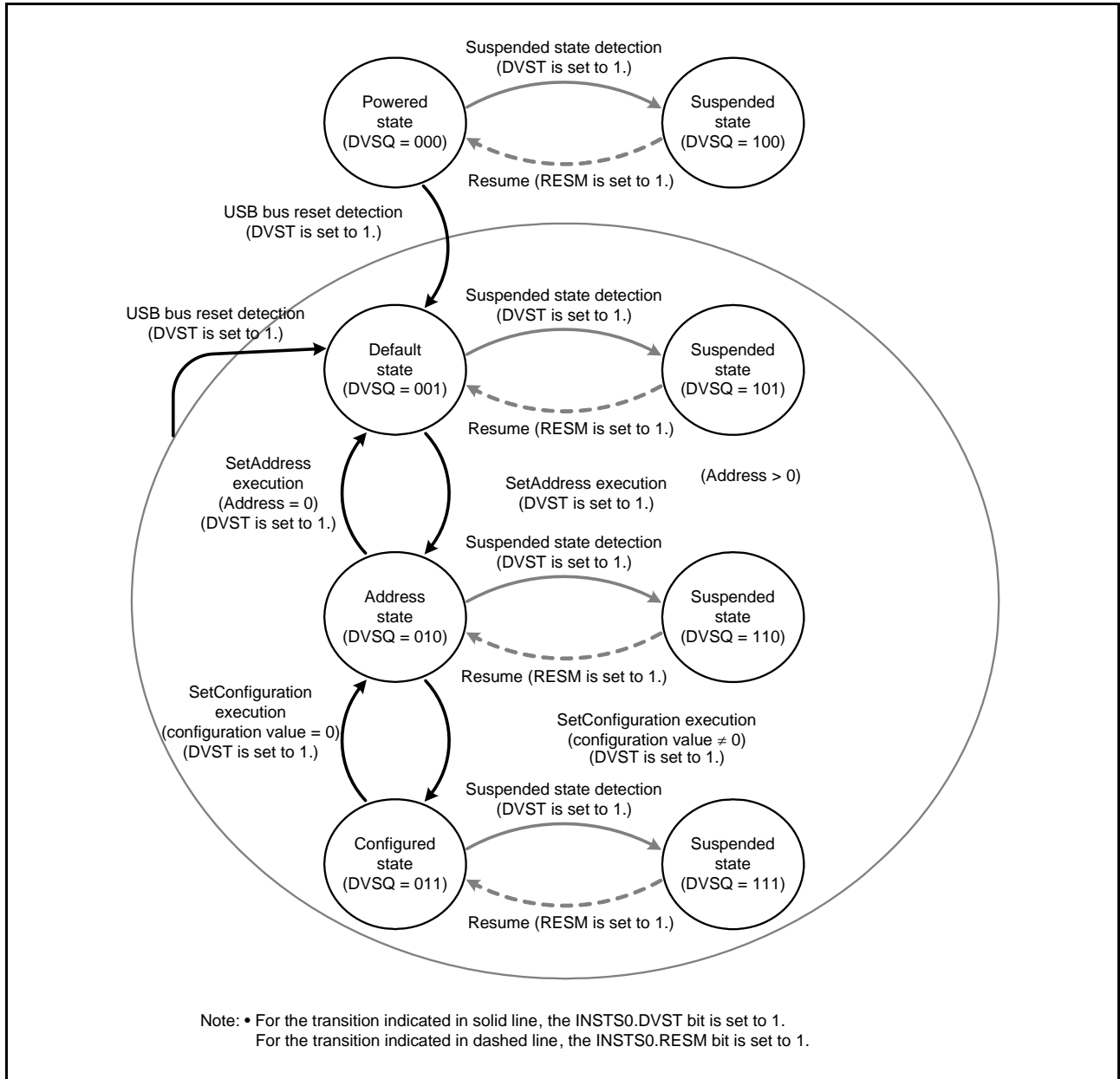


Figure 31.11 Device State Transitions

### 31.3.3.5 Control Transfer Stage Transition Interrupt

Figure 31.12 is a diagram of control transfer stage transitions in the USB module. The USB module controls the control transfer sequence and generates control transfer stage transition interrupts. The control transfer stage transition interrupts can be enabled or disabled individually using INTENB0. The transfer stage to which a transition was made can be confirmed reading the CTSQ[2:0] bits in INTSTS0.

The control transfer sequence errors are listed below. If an error occurs, the PID[1:0] bits in DCPCTR are set to 1xb (STALL response).

During control read transfer:

- An OUT token is received while no data has been transferred for the IN token at the data stage.
- An IN token is received at the status stage.
- A data packet with DATAPID = DATA0 is received at the status stage.

During control write transfer:

- An IN token is received while no ACK response has been returned for the OUT token at the data stage.
- A data packet with DATAPID = DATA0 is received for the first data packet at the data stage.
- An OUT token is received at the status stage.

During no-data control transfers:

- An OUT token is received at the status stage.

At the control write transfer data stage, if the number of receive data exceeds the wLength value of the USB request, it cannot be recognized as a control transfer sequence error. At the control read transfer status stage, packets other than zero-length packets are received by an ACK response and the transfer ends normally.

When a CTRT interrupt occurs in response to a sequence error (the INTSTS0.CTRT bit is 1), the DCPCTR.CTSQ[2:0] = 110 value is retained until CTRT = 0 is written from the system (the interrupt status is cleared). Therefore, while CTSQ [2:0] = 110 is being held, the CTRT interrupt that ends the setup stage will not be generated even if a new USB request is received. (The USB module retains the setup stage end, and after the interrupt status has been cleared by software, a setup stage end interrupt is generated.)

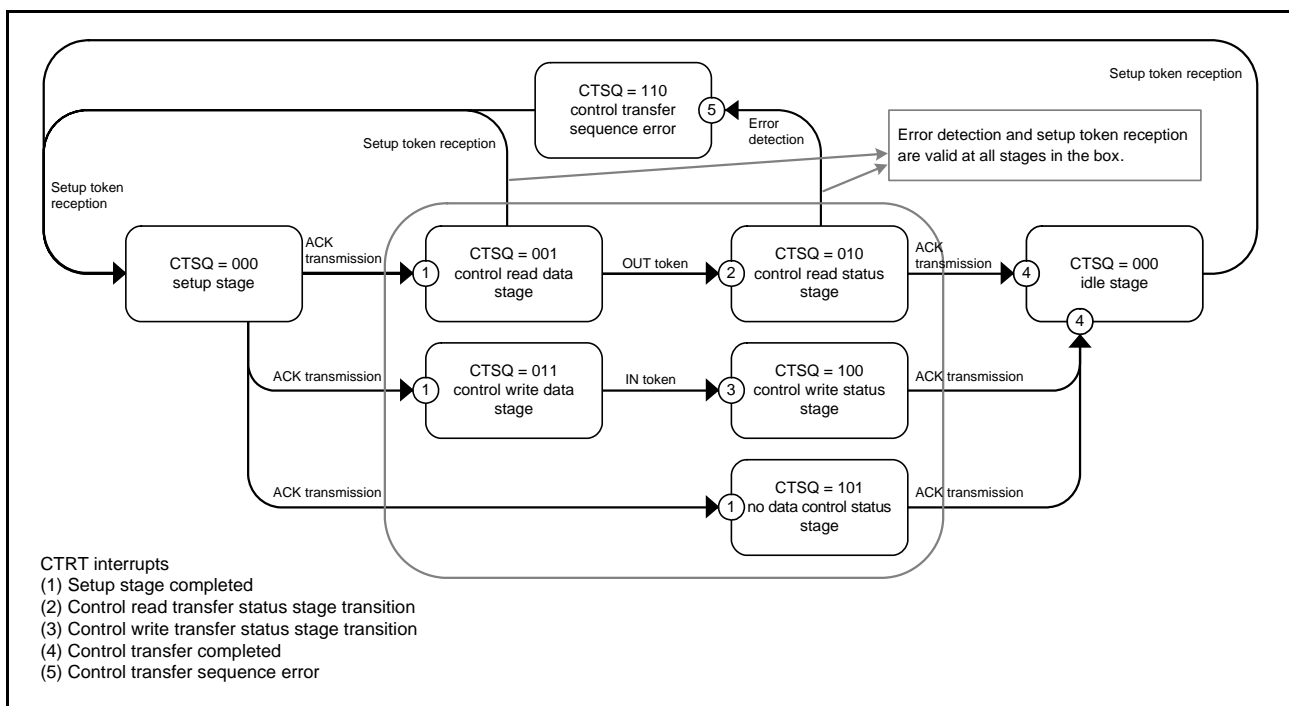


Figure 31.12 Control Transfer Stage Transitions



### 31.3.3.6 Frame Update Interrupt

The USB module updates the frame number and generates an SOFR interrupt if it detects a new SOF packet during full-speed operation.

### 31.3.3.7 VBUS Interrupt

When the USB0\_VBUS pin level changes, a VBUS interrupt is generated. The level of the USB0\_VBUS pin can be checked with the INTSTS0.VBSTS bit.

### 31.3.3.8 Resume Interrupt

A resume interrupt is generated when the device state is the suspended state and the USB bus state has changed (from J-state to K-state, or from J-state to SE0). Recovery from the suspended state is detected by means of the resume interrupt.

## 31.3.4 Pipe Control

Table 31.14 lists the pipe setting items in the USB module. With USB data transfer, data transfer has to be carried out using the logic pipe called the endpoint. The USB module has ten pipes that are used for data transfer.

Appropriate settings should be made for each of the pipes according to the specifications of the system.

**Table 31.14 Pipe Setting Items (1/2)**

Register Name	Bit Name	Setting	Remarks
PIPECFG	TYPE[1:0]	Specifies the transfer type	PIPE1 to PIPE9: Can be set
	BFRE	Selects the BRDY interrupt mode	PIPE1 to PIPE5: Can be set
	DBLB	Selects double buffer mode	PIPE1 to PIPE5: Can be set
	DIR	Selects transfer direction	IN or OUT can be set
	EPNUM[3:0]	Endpoint number	PIPE1 to PIPE9: Can be set A value other than 0000b should be set when the pipe is used.
	SHTNAK	Selects disabled state for pipe when transfer ends	PIPE1 and PIPE2: Can be set (only when bulk transfer has been selected) PIPE3 to PIPE5: Can be set
DCPMAXP PIPEMAXP	MXPS	Maximum packet size	Compliant with the USB standard.
PIPEPERI	IFIS	Buffer flush	PIPE1 and PIPE2: Can be set (only when isochronous transfer has been selected) PIPE3 to PIPE9: Cannot be set
	IITV[2:0]	Interval counter	PIPE1 and PIPE2: Can be set (only when isochronous transfer has been selected) PIPE3 to PIPE9: Cannot be set
DCPCTR PIPEnCTR	BSTS	Buffer status	For the DCP, receive buffer status and transmit buffer status are switched with the ISEL bit.
	INBUFM	IN buffer monitor	Available only for PIPE1 to PIPE5.
	ATREPM	Auto response mode	PIPE1 to PIPE5: Can be set
	ACLRM	Auto buffer clear	PIPE1 to PIPE9: Can be set
	SQCLR	Sequence clear	Clears the data toggle bit.
	SQSET	Sequence set	Sets the data toggle bit.
	SQMON	Sequence monitor	Monitors the data toggle bit.
	PBUSY	Pipe busy status	
PID[1:0]	Response PID	See section 31.3.4.6, Response PID.	
PIPEnTRE	TRENB	Transaction counter enable	PIPE1 to PIPE5: Can be set
	TRCLR	Current transaction counter clear	PIPE1 to PIPE5: Can be set

**Table 31.14 Pipe Setting Items (2/2)**

Register Name	Bit Name	Setting	Remarks
PIPEnTRN	TRNCNT [15:0]	Transaction counter	PIPE1 to PIPE5: Can be set

### 31.3.4.1 Pipe Control Register Switching Procedures

The following bits in the pipe control registers can be modified only when USB communication is disabled (PID = NAK).

Registers that Should Not be Set in the USB Communication Enabled (PID = BUF) State:

- Bits in DCPMAXP
- SQCLR and SQSET bits in DCPCTR
- Bits in PIPECFG, PIPEMAXP, and PIPEPERI
- ATREPM, ACLRM, SQCLR, and SQSET bits in PIPEnCTR
- Bits in PIPEnTRE and PIPEnTRN

In order to modify the above bits in the USB communication enabled (PID = BUF) state, follow the procedure shown below:

1. A request to modify bits in the pipe control register occurs.
2. Modify the PID[1:0] bits corresponding to the pipe to NAK.
3. Wait until the corresponding PIPEnCTR.PBUSY bit is cleared to 0.
4. Modify the bits in the pipe control register.

The following bits in the pipe control registers can be modified only when the pertinent pipe information has not been set by the CURPIPE[3:0] bits in CFIFOSEL, D0FIFOSEL, and D1FIFOSEL.

Registers that Should Not be Set When CURPIPE[3:0] bits are set:

- Bits in DCPMAXP
- Bits in PIPECFG, PIPEMAXP and PIPEPERI

In order to modify pipe information, the CURPIPE[3:0] bits in the port select registers should be set to a pipe other than the pipe to be modified. For the DCP, the buffer should be cleared using the BCLR bit in the port control register after the pipe information is modified.

### 31.3.4.2 Transfer Types

The PIPECFG.TYPE bits are used to specify the transfer type for each pipe. The transfer types that can be set for the pipes are as follows.

- DCP: No setting is necessary (fixed at control transfer).
- PIPE1 and PIPE2: These should be set to bulk transfer or isochronous transfer.
- PIPE3 to PIPE5: These should be set to bulk transfer.
- PIPE6 to PIPE9: These should be set to interrupt transfer.

### 31.3.4.3 Endpoint Number

The PIPECFG.EPNUM[3:0] bits are used to set the endpoint number for each pipe. The DCP is fixed at endpoint 0. The other pipes can be set from endpoint 1 to endpoint 15.

- DCP: No setting is necessary (fixed at end point 0).
- PIPE1 to PIPE9: The endpoint numbers from 1 to 15 should be selected and set. These should be set so that the combination of the PIPECFG.DIR bit and EPNUM[3:0] bits is unique.

### 31.3.4.4 Maximum Packet Size Setting

The DCPMAXP.MAXP[6:0] bits and PIPEMAXP.MXPS[8:0] bits are used to specify the maximum packet size for each pipe. DCP and PIPE1 to PIPE5 can be set to any of the maximum pipe sizes defined by the USB specification. For PIPE6 to PIPE9, 64 bytes are the upper limit of the maximum packet size. The maximum packet size should be set before beginning the transfer (PID = BUF).

- DCP: Set 8, 16, 32, or 64.
- PIPE1 to PIPE5: Set 8, 16, 32, or 64 when using bulk transfer.
- PIPE1 and PIPE2: Set a value between 1 and 256 when using isochronous transfer.
- PIPE6 to PIPE9: Set a value between 1 and 64.

### 31.3.4.5 Transaction Counter (For PIPE1 to PIPE5 in Reading Direction)

When the specified number of transactions has been completed in the data packet receiving direction, the USB module recognizes that the transfer has ended. Two transaction counters are provided: one is the PIPEnTRN register that specifies the number of transactions to be executed and the other is the current counter that internally counts the number of executed transactions. With the PIPECFG.SHTNAK bit set to 1, when the current counter value matches the specified number of transactions, the PID[1:0] bits of the corresponding PIPE is set to NAK and the subsequent transfer is disabled. The transactions can be counted again from the beginning by initializing the current counter of the transaction counter function through the PIPEnTRE.TRCLR bit. The information read from the PIPEnTRN register differs depending on the setting of the PIPEnTRE.TRENB bit.

- PIPEnTRE.TRENB = 0: The specified transaction counter value can be read.
- PIPEnTRE.TRENB = 1: The current counter value indicating the internally counted number of executed transactions can be read.

When operating the PIPEnTRE.TRCLR bit, the following should be noted.

- If the transactions are being counted and PID = BUF, the current counter cannot be cleared.
- If there is any data left in the buffer, the current counter cannot be cleared.

### 31.3.4.6 Response PID

The PID[1:0] bits in DCPCTR and PIPEnCTR are used to set the response PID for each pipe. The following shows the USB module operation with various response PID settings:

- Response PID settings:

The response PID is used to specify the response to transactions from the host.

- NAK setting: The NAK response is returned in response to the generated transaction.
- BUF setting: Responses are made to transactions according to the status of the buffer memory.
- STALL setting: The STALL response is returned in response to the generated transaction.

Note: • For setup transactions, an ACK response is returned regardless of the PID[1:0] setting, and the USB request is stored in the register.

The USB module may write to the PID[1:0] bits, depending on the results of the transaction as described below.

- When the response PID is set by hardware:
- NAK setting: In the following cases, PID = NAK is set and NAK is returned in response to transactions:  
When the SETUP token is received normally (DCP only).  
If the transaction counting ends or a short packet is received when the PIPECFG.SHTNAK bit has been set to 1 for bulk transfer.
- BUF setting: There is no BUF writing by the USB module.
- STALL setting: In the following cases, PID = STALL is set and STALL is returned in response to transactions:  
When a maximum packet size exceeded error is detected in the received data packet.  
When a control transfer sequence error has been detected (DCP only).

#### 31.3.4.7 Data PID Sequence Bit

The USB module automatically toggles the sequence bit in the data PID when data is transferred successfully in the control transfer data stage, bulk transfer, and interrupt transfer. The sequence bit of the next data PID to be transmitted can be confirmed with the SQMON bit in DCPCTR and PIPEnCTR. When data is transmitted, the sequence bit switches at the timing of ACK handshake reception. When data is received, the sequence bit switches at the timing of ACK handshake transmission. The SQCLR bit in DCPCTR and the SQSET bit in PIPEnCTR can be used to change the data PID sequence bit.

When the control transfer is used, the USB module automatically sets the sequence bit when a stage transition is made. DATA0 is returned when the setup stage is ended and DATA1 is returned in a status stage. Therefore, software settings are not required.

For the Clear Feature request transmission or reception, the data PID sequence bit should be set by software.

#### 31.3.4.8 Response PID = NAK Function

The USB module has a function that disables pipe operation (PID response = NAK) at the timing at which the final data packet of a transaction is received (the USB module automatically distinguishes this based on reception of a short packet or the transaction counter) by setting the PIPECFG.SHTNAK bit to 1.

When the double buffer mode is being used for the buffer memory, using this function enables reception of data packets in transfer units. If pipe operation has been disabled, software should set the pipe to the enabled state again (PID response = BUF).

The response PID = NAK function can be used only when bulk transfers are used.

#### 31.3.4.9 Auto Response Mode

With the pipes for bulk transfer (PIPE1 to PIPE5), when the PIPEnCTR.ATREPM bit is set to 1, a transition is made to auto response mode. During an OUT transfer (DIR = 0), OUT-NAK mode is entered, and during an IN transfer (DIR = 1), null auto response mode is entered.

#### 31.3.4.10 OUT-NAK Mode

With the pipes for bulk OUT transfer, NAK is returned in response to an OUT token and an NRDY interrupt is output when the PIPEnCTR.ATREPM bit is set to 1. To make a transition from normal mode to OUT-NAK mode, OUT-NAK mode should be specified in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, OUT-NAK mode becomes valid. However, if an OUT token is received immediately before pipe operation is disabled, the token data is normally received, and an ACK is returned to the host.

To make a transition from OUT-NAK mode to normal mode, OUT-NAK mode should be canceled in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). In normal mode, reception of OUT data is enabled.

### 31.3.4.11 Null Auto Response Mode

With the pipes for bulk IN transfer, zero-length packets are continuously transmitted when the PIPEnCTR.ATREPM bit is set to 1.

To make a transition from normal mode to null auto response mode, null auto response mode should be set in the pipe operation disabled state (response PID = NAK) before enabling pipe operation (response PID = BUF). After pipe operation has been enabled, null auto response mode becomes valid. Before setting null auto response mode, PIPEnCTR.INBUFM = 0 should be confirmed because the mode can be set only when the buffer is empty. If the PIPEnCTR.INBUFM bit is 1, the buffer should be emptied with the PIPEnCTR.ACLRM bit. While a transition to null auto response mode is being made, data should not be written from the FIFO port.

To make a transition from null auto response mode to normal mode, pipe operation disabled state (response PID = NAK) should be retained for the period of zero-length packet transmission (about 10  $\mu$ s) before canceling null auto response mode. In normal mode, data can be written from the FIFO port; therefore, packet transmission to the host is enabled by enabling pipe operation (response PID = BUF).

### 31.3.5 FIFO Buffer Memory

#### 31.3.5.1 FIFO Buffer Memory

The USB module has FIFO buffer memory for data transfer. The memory area used for each pipe is managed by the USB module. The FIFO buffer memory has two states depending on whether the access right is assigned to the system (CPU side) or the USB module (SIE side).

- Buffer Status

Table 31.15 and Table 31.16 list the buffer status in the USB module. The buffer memory status can be confirmed using the DCPCTR.BSTS bit and the PIPEnCTR.INBUFM bit. The access direction for the buffer memory can be specified using either the PIPEnCFG.DIR bit or the CFIFOSEL.ISEL bit (when DCP is selected).

The PIPEnCTR.INBUFM bit is valid for PIPE1 to PIPE5 in the transmitting direction.

When a transmitting pipe uses the double buffer configuration, software can read the DCPCTR.BSTS bit to monitor the buffer memory status on the CPU side and the PIPEnCTR.INBUFM bit to monitor the buffer memory status on the SIE side. When the BEMP interrupt may not show the buffer empty status because the write access to the FIFO port by the CPU (DMAC) is slow, software can use the PIPEnCTR.INBUFM bit to confirm the end of transmission.

**Table 31.15 Buffer Status Indicated by the BSTS Bit**

ISEL or DIR	BSTS	Buffer Memory Status
0 (receiving direction)	0	There is no received data, or data is being received. Reading from the FIFO port is disabled.
0 (receiving direction)	1	There is received data, or a zero-length packet has been received. Reading from the FIFO port is allowed. Note that when a zero-length packet is received, reading is not possible and the buffer must be cleared.
1 (transmitting direction)	0	The transmission has not been completed. Writing to the FIFO port is disabled.
1 (transmitting direction)	1	The transmission has been completed. CPU write is allowed.

**Table 31.16 Buffer Status Indicated by the INBUFM Bit**

DIR	INBUFM	Buffer Memory Status
0 (receiving direction)	Invalid	Invalid
1 (transmitting direction)	0	The transmission has been completed. There is no waiting data to be transmitted.
1 (transmitting direction)	1	The FIFO port has written data to the buffer. There is data to be transmitted.

#### 31.3.5.2 FIFO Buffer Clearing

Table 31.17 lists the clearing of the FIFO buffer memory by the USB module. The buffer memory can be cleared using the BCLR bit, the DnFIFOSEL.DCLRM bit, and the PIPEnCTR.ACLRM bit in the port control register.

**Table 31.17 List of Buffer Clearing Methods**

FIFO Buffer Clearing Mode	Clearing Buffer Memory on CPU Side	Mode for Automatically Clearing Buffer Memory after Reading Specified Pipe Data	Auto Buffer Clear Mode for Discarding All Received Packets
Register used	CFIFOCTR DnFIFOCTR	DnFIFOSEL	PIPEnCTR
Bit used	BCLR	DCLRM	ACLRM
Clearing condition	Cleared by writing 1	1: Mode valid 0: Mode invalid	1: Mode valid 0: Mode invalid

### (1) Auto Buffer Clear Mode Function

With the USB module, all received data packets are discarded if the PIPEnCTR.ACLRm bit is set to 1. If a correct data packet has been received, the ACK response is returned to the host controller. The auto buffer clear mode function can be set only in the buffer memory reading direction.

If the PIPEnCTR.ACLRm bit is set to 1 and then to 0, the buffer memory of the selected pipe can be cleared regardless of the access direction.

An access cycle of at least 100 ns is required for the internal hardware sequence processing time between PIPEnCTR.ACLRm = 1 and PIPEnCTR.ACLRm = 0.

### (2) Buffer Memory Specifications (Single or Double Setting)

Either a single or double buffer configuration can be selected for PIPE1 to PIPE5, using the PIPEnCFG.DBLLB bit.

### 31.3.5.3 FIFO Port Functions

Table 31.18 lists the settings for the FIFO port functions of the USB module. In write access, writing data until the maximum packet size is reached automatically enables transmission of the data. To enable transmission before the maximum packet size is reached, the BVAL bit in the port control register should be set to end writing. To send a zero-length packet, the BCLR bit in the register should be used to clear the buffer and then the BVAL bit set in order to end writing.

In read access, reception of new packets is automatically enabled when all data has been read. Data cannot be read when a zero-length packet has been received (the DTLN[8:0] bits = 0), so the BCLR bit in the register should be used to clear the buffer. The length of the receive data can be confirmed using the DTLN[8:0] bits in the port control register.

**Table 31.18 FIFO Port Function Settings**

Register Name	Bit Name	Function
CFIFOSEL, DnFIFOSEL (n = 0 or 1)	RCNT	Selects DTLN read mode.
	REW	Buffer memory rewind (re-read, rewrite).
	DCLRm	Automatically clears receive data for a specified pipe after the data has been read (only for DnFIFO).
	DREQE	Enables DMA transfers (only for DnFIFO).
	MBW	FIFO port access bit width.
	BIGEND	Selects FIFO port endian.
	ISEL	FIFO port access direction (only for DCP).
CFIFOCTR, DnFIFOCTR (n = 0 or 1)	CURPIPE[3:0]	Selects the current pipe.
	BVAL	Ends writing to the buffer memory.
	BCLR	Clears the buffer memory on the CPU side.
	DTLN[8:0]	Checks the length of receive data.

### (1) FIFO Port Selection

Table 31.19 lists the pipes that can be selected with the various FIFO ports. The pipe to be accessed should be selected using the CURPIPE[3:0] bits in the port select register. After the pipe is selected, whether the written value can be correctly read from the CURPIPE[3:0] bits should be checked. (If the previous pipe number is read, it indicates that the pipe modification is being executed by the USB controller.) Then, the FIFO port can be accessed after the FRDY bit in the port control register = 1 is checked.

In addition, the bus width to be accessed should be selected using the MBW bit in the port select register. The buffer memory access direction conforms to the PIPEnCFG.DIR bit. Only for the DCP, the ISEL bit in the port select register determines the direction.

**Table 31.19 FIFO Port Access Categorized by Pipe**

Pipe	Access Method	Port that can be Used
DCP	CPU access	CFIFO port register
PIPE1 to PIPE9	CPU access	CFIFO port register D0FIFO/D1FIFO port register
	DMA access	D0FIFO/D1FIFO port register

**(2) REW Bit**

It is possible to temporarily stop access to the pipe currently being accessed, access a different pipe, and then continue processing for the current pipe again. The REW bit in the port select register is used for this processing.

If a pipe is selected through the CURPIPE[3:0] bits in the port select register with the REW bit set to 1, the pointer used for reading from and writing to the buffer memory is reset, and reading or writing can be carried out from the first byte. If a pipe is selected with 0 set for the REW bit, data can be read and written in continuation from the previous selection, without the pointer being reset.

To access the FIFO port, the FRDY bit in the port control register = 1 should be checked after selecting a pipe.

**31.3.5.4 DMA Transfers (D0FIFO and D1FIFO Ports)****(1) Overview of DMA Transfers**

For PIPE1 to PIPE9, the FIFO port can be accessed using the DMAC. When accessing the buffer for the pipe targeted for DMA transfer is enabled, a DMA transfer request is issued.

The unit of transfer to the FIFO port should be selected using the DnFIFOSEL.MBW bit and the pipe targeted for the DMA transfer should be selected using the DnFIFOSEL.CURPIPE[3:0] bits. The selected pipe should not be changed during the DMA transfer.

**(2) DnFIFO Auto Clear Mode (D0FIFO and D1FIFO Port Reading Direction)**

If 1 is set in the DnFIFOSEL.DCLRM bit, the USB module automatically clears the buffer memory of the selected pipe when reading of data from the buffer memory has been completed.

Table 31.20 lists the packet reception and buffer memory clearing processing by software for each of the various settings. As listed in Table 31.20, the buffer clearing conditions depend on the value set in the PIPECFG.BFRE bit. Using the DnFIFOSEL.DCLRM bit eliminates the need for the buffer to be cleared by software in any situation that requires buffer clearing. This enables DMA transfers without involving software.

The DnFIFO auto clear mode can be set only in the buffer memory reading direction.

**Table 31.20 Packet Reception and Buffer Memory Clearing Processing by Software**

Buffer Status when Packet is Received	Register Setting			
	DnFIFOSEL.DCLRM = 0		DnFIFOSEL.DCLRM = 1	
	BFRE = 0	BFRE = 1	BFRE = 0	BFRE = 1
Buffer full	Automatically cleared	Automatically cleared	Automatically cleared	Automatically cleared
Zero-length packet reception	Cleared by software	Cleared by software	Automatically cleared	Automatically cleared
Normal short packet reception	Automatically cleared	Cleared by software	Automatically cleared	Automatically cleared
Transaction count end	Automatically cleared	Cleared by software	Automatically cleared	Automatically cleared



### 31.3.6 Control Transfers (DCP)

In the data stage of control transfers, data is transferred using the default control pipe (DCP).

The DCP buffer memory is a 64-byte single buffer and is a fixed area that is shared for both control reading and control writing. The buffer memory can be accessed only through the CFIFO port.

#### 31.3.6.1 Control Transfers

##### (1) Setup Stage

The USB module always sends an ACK response for a correct setup packet targeted to the USB module. The operation of the USB module in the setup stage is described below.

When receiving a new setup packet, the USB module sets the following bits.

- Set the INTSTS0.VALID bit to 1.
- Set the DCPCTR.PID[1:0] bits to NAK.
- Set the DCPCTR.CCPL bit to 0.

When receiving a data packet right after the setup packet, the USB module stores the USB request parameters in USBREQ, USBVAL, USBINDX, and USBLENG.

Response processing with respect to the control transfer should be carried out after setting VALID = 0. In the INTSTS0.VALID = 1 state, PID = BUF cannot be set, and the data stage cannot be terminated.

Using the function of the INTSTS0.VALID bit, the USB module can suspend the current request processing when receiving a new USB request during a control transfer, and can send a response to the newest request.

In addition, the USB module automatically detects the direction bit (bit 8 of bmRequestType) and the request data length (wLength) of the received USB request, distinguishes between control read transfer, control write transfer, and no-data control transfer, and controls stage transitions. For a wrong sequence, the sequence error of the control transfer stage transition interrupt is generated, and the software is notified of occurrence of the error. For the stage control of the USB module, see Figure 31.12.

##### (2) Data Stage

Data transfers corresponding to received USB requests should be done using the DCP. Before accessing the DCP buffer memory, the access direction should be specified using the CFIFOSEL.ISEL bit.

If the transfer data is larger than the size of the DCP buffer memory, the data transfer should be carried out using the BRDY interrupt for control write transfers and the BEMP interrupt for control read transfers.

##### (3) Status Stage

Control transfers are terminated by setting the DCPCTR.CCPL bit to 1 while the DCPCTR.PID[1:0] bits are set to BUF. After the above settings have been made, the USB module automatically executes the status stage in accordance with the data transfer direction determined at the setup stage. The specific procedure is as follows.

- For control read transfers  
The USB module transmits a zero-length packet and receives an ACK response from the USB host.
- For control write transfers and no-data control transfers  
The USB module receives a zero-length packet from the USB host and sends an ACK response.

##### (4) Control Transfer Auto Response Function

The USB module automatically responds to a correct SET\_ADDRESS request. If any of the following errors occurs in the SET\_ADDRESS request, a response from the software is necessary.

- Any transfer other than a control read transfer: bmRequestType ≠ 00h
- Request error : wIndex ≠ 00h
- Any transfer other than a no-data control transfer: wLength ≠ 00h
- Request error: wValue > 7Fh

- Control transfer of a device state error: INTSTS0.DVSQ[2:0] = 011b (Configured)

For all requests other than the SET\_ADDRESS request, a response is required from the corresponding software.

### 31.3.7 Bulk Transfers (PIPE1 to PIPE5)

The buffer memory usage (single/double buffer setting) can be selected for bulk transfers. The USB provides the following functions for bulk transfers.

- BRDY interrupt function (PIPECFG.BFRE bit: see section 31.3.3.1, (2) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 1)
- Transaction count function (PIPEnTRE.TRENB, TRCLR bits, and PIPEnTRN.TRNCNT[15:0] bits: see section 31.3.4.5, Transaction Counter (For PIPE1 to PIPE5 in Reading Direction))
- Response PID = NAK function (PIPECFG.SHTNAK bit: see section 31.3.4.8, Response PID = NAK Function)
- Auto response mode (PIPEnCTR.ATREPM bit: see section 31.3.4.9, Auto Response Mode)

### 31.3.8 Interrupt Transfers (PIPE6 to PIPE9)

The USB module carries out interrupt transfers in accordance with the timing controlled by the host controller.

### 31.3.9 Isochronous Transfers (PIPE1 and PIPE2)

The USB module has the following functions for isochronous transfers.

- Notification of isochronous transfer error information
- Interval counter (specified by the PIPEPERI.IITV[2:0] bits)
- Isochronous IN transfer data setup control (IDL function)
- Isochronous IN transfer buffer flush function (specified by the PIPEPERI.IFIS bit)

#### 31.3.9.1 Error Detection in Isochronous Transfers

The USB module has a function for detecting the error information described below, so that when errors occur in isochronous transfers, software can control them. Table 31.21 and Table 31.22 show the priority in which errors are confirmed and the interrupts generated corresponding to errors.

##### (a) PID errors

- If the PID of the received packet is illegal.

##### (b) CRC errors and bit stuffing errors

- If an error occurs in the CRC of the received packet or the bit stuffing is illegal.

##### (c) Maximum packet size exceeded

- The data of the received packet is larger than the specified maximum packet size.

##### (d) Overrun and underrun errors

- When there is no data to be sent in the buffer memory at the token receiving timing in the IN (transmitting) direction.
- When the buffer memory is full at the token receiving timing in the OUT (receiving) direction.

##### (e) Interval errors

An interval error is generated on any of the following conditions.

- During an isochronous IN transfer, an IN token could not be received in the interval frame.
- During an isochronous OUT transfer, an OUT token was received in frames other than the interval frame.

**Table 31.21 Error Detection when a Token is Received**

Detection Priority	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated (ignored as a corrupted packet).
2	CRC errors and bit stuffing errors	No interrupts are generated (ignored as a corrupted packet).
3	Overrun and underrun errors	An NRDY interrupt is generated to set the FRMNUM.OVRN bit. A zero-length packet is transmitted in response to IN token. However, no data packets are received in response to OUT token.
4	Interval errors	An NRDY interrupt is generated.

**Table 31.22 Error Detection when a Data Packet is Received**

Detection Priority	Error	Generated Interrupt and Status
1	PID errors	No interrupts are generated (ignored as a corrupted packet).
2	CRC errors and bit stuffing errors	An NRDY interrupt is generated to set the FRMNUM.CRCE bit.
3	Maximum packet size exceeded errors	A BEMP interrupt is generated to set the PID[1:0] bits to STALL.

### 31.3.9.2 DATA-PID

The USB module operates as follows in response to the received PID.

IN direction

- DATA0: Sent as data packet PID
- DATA1: Not sent
- DATA2: Not sent
- mData: Not sent

OUT direction

- DATA0: Received normally as data packet PID
- DATA1: Received normally as data packet PID
- DATA2: Packets are ignored
- mData: Packets are ignored

### 31.3.9.3 Interval Counter

The isochronous transfer interval can be selected by setting the PIPEPERI.IITV[2:0] bits in PIPEPERI. The interval counter enables the functions listed in Table 31.23.

**Table 31.23 Interval Counter Function**

Transfer Direction	Function	Conditions for Detection
IN	Flushes transmit buffer	When an IN token cannot be successfully received in the interval frame during an isochronous IN transfer
OUT	Notifies that a token not being received	When an OUT token cannot be successfully received in the interval frame during an isochronous OUT transfer

The interval count is carried out when an SOF is received or for interpolated SOFs, so the isochronism can be maintained even if an SOF is damaged. The frame interval that can be set is the  $2^{IITV}$  frames.

#### (1) Counter Initialization

The USB module initializes the interval counter under the following conditions.

- Power-on Reset  
The PIPEPERI.IITV[2:0] bits are initialized.
- Buffer memory initialization using the ACLRM bit  
The PIPEPERI.IITV[2:0] bits are not initialized but the count value is initialized. Setting the PIPECTR.ACLRM bit to 0 starts counting from the value set in the PIPEPERI.IITV[2:0] bits.

After the interval counter has been initialized, counting is started under either of the following conditions 1 and 2 when a packet has been transferred successfully.

1. An SOF is received after transmission of data in response to an IN token in the PID = BUF state.
2. An SOF is received after reception of data of an OUT token in the PID = BUF state.

Note that the interval counter is not initialized under the following conditions.

1. When the PID[1:0] bits are set to NAK or STALL  
The interval timer does not stop. The USB module attempts transactions at the subsequent interval.
2. When the USB bus is reset or USB is suspended  
The PIPEPERI.IITV[2:0] bits are not initialized. When an SOF has been received, counting is restarted from the value prior to the reception of the SOF.

(2) Interval Counting and Transfer Control

**When the selected pipe is for isochronous OUT transfers:**

The USB module generates an NRDY interrupt when it fails to receive a data packet within the interval set by the PIPEPERI.IITV[2:0] bits.

The USB module also generates an NRDY interrupt when the USB module fails to receive data because of a CRC error or other errors contained in the data packet or because of the FIFO buffer being full.

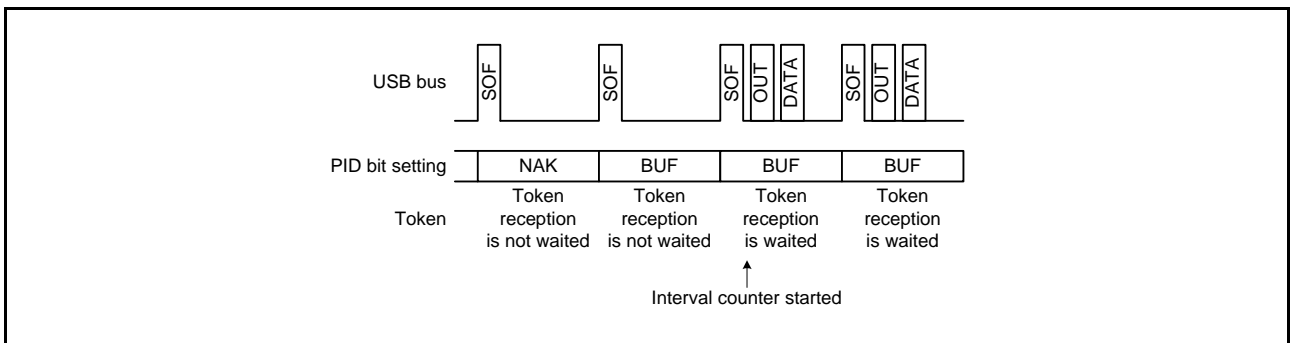
The NRDY interrupt is generated at the timing of SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the interrupt to be generated at the timing to receive the SOF packet.

However, when the IITV bits are set to a value other than 0, the USB module generates an NRDY interrupt on receiving an SOF packet for every interval after starting interval counting operation.

When the PID[1:0] bits are set to NAK by software after starting the interval timer, the USB module does not generate an NRDY interrupt on receiving an SOF packet.

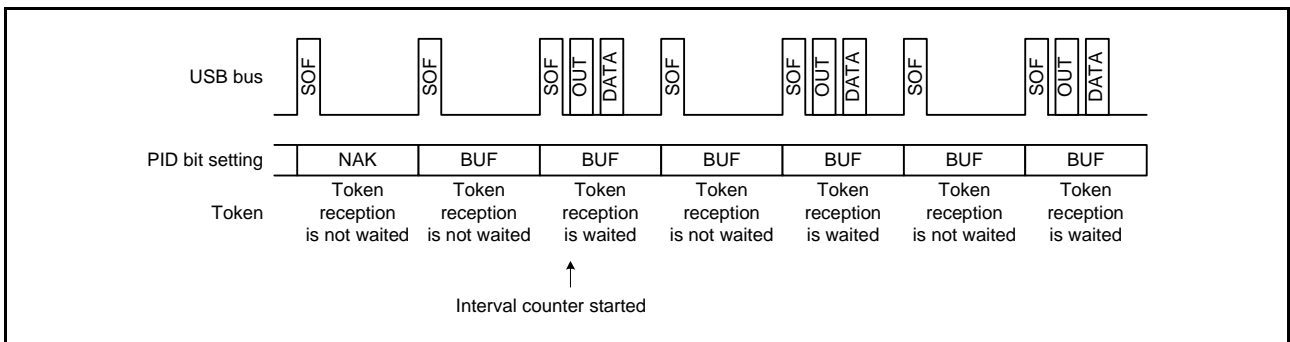
The timing to start interval counting depends on the IITV[2:0] bit setting as shown in the followings.

- When IITV = 0: The interval counting starts at the frame following the frame in which software has set the PID[1:0] bits for the selected pipe to BUF.



**Figure 31.13 Relationship between Frames and Expected Token Reception when IITV = 0**

- When IITV ≠ 0: The interval counting starts on completion of successful reception of the first data packet after the PID[1:0] bits for the selected pipe have been modified to BUF.



**Figure 31.14 Relationship between Frames and Expected Token Reception when IITV ≠ 0**

**When the selected pipe is for isochronous IN transfers:**

The PIPEPERI.IFIS bit should be 1 for this use. When PIPEPERI.IFIS = 0, the USB module transmits a data packet in response to the received IN token irrespective of the PIPEPERI.IITV[2:0] bit setting.

When PIPEPERI.IFIS = 1, the USB module clears the FIFO buffer when the USB module fails to receive an IN token in the frame at the interval set by the IITV[2:0] bits while there is data to be transmitted in the FIFO buffer.

The USB module also clears the FIFO buffer when the USB module fails to receive an IN token successfully because of a bus error such as a CRC error contained in the IN token.

The FIFO buffer is cleared at the timing of SOF packet reception. Even if the SOF packet is corrupted, the internal interpolation allows the FIFO buffer to be cleared at the timing to receive the SOF packet.

The timing to start interval counting depends on the IITV[2:0] bit setting (similar to the timing during OUT transfers). The interval is counted on any of the following conditions.

- When the RES# pin reset is applied to the USB module (here, the IITV[2:0] bits are also cleared to 000b).
- When software sets the PIPEEnCTR.ACLRM bit to 1.
- When the USB module detects a USB bus reset.

**(3) Setup of Data to be Transmitted using Isochronous Transfer**

After data has been written to the buffer memory, a data packet can be transmitted with the next frame after the frame in which an SOF packet is detected. This function is called the isochronous transfer transmission data setup function, and it makes it possible to designate the frame from which transmission began.

In a double buffer configuration, even after the writing of data to both buffers has been completed, transmission will be enabled for only one buffer to which data writing was completed first. Accordingly, even if multiple IN tokens are received in a frame, only one packet of data is transmitted from a single buffer.

When an IN token is received, if the buffer memory is in the transmission enabled state, the USB module transmits data as a normal response. If the buffer memory is not in the transmission enabled state, however, a zero-length packet is sent and an underrun error occurs.

Figure 31.15 shows an example of transmission using the isochronous transfer transmission data setup function with the USB module when the IITV = 0 (every frame) has been set.

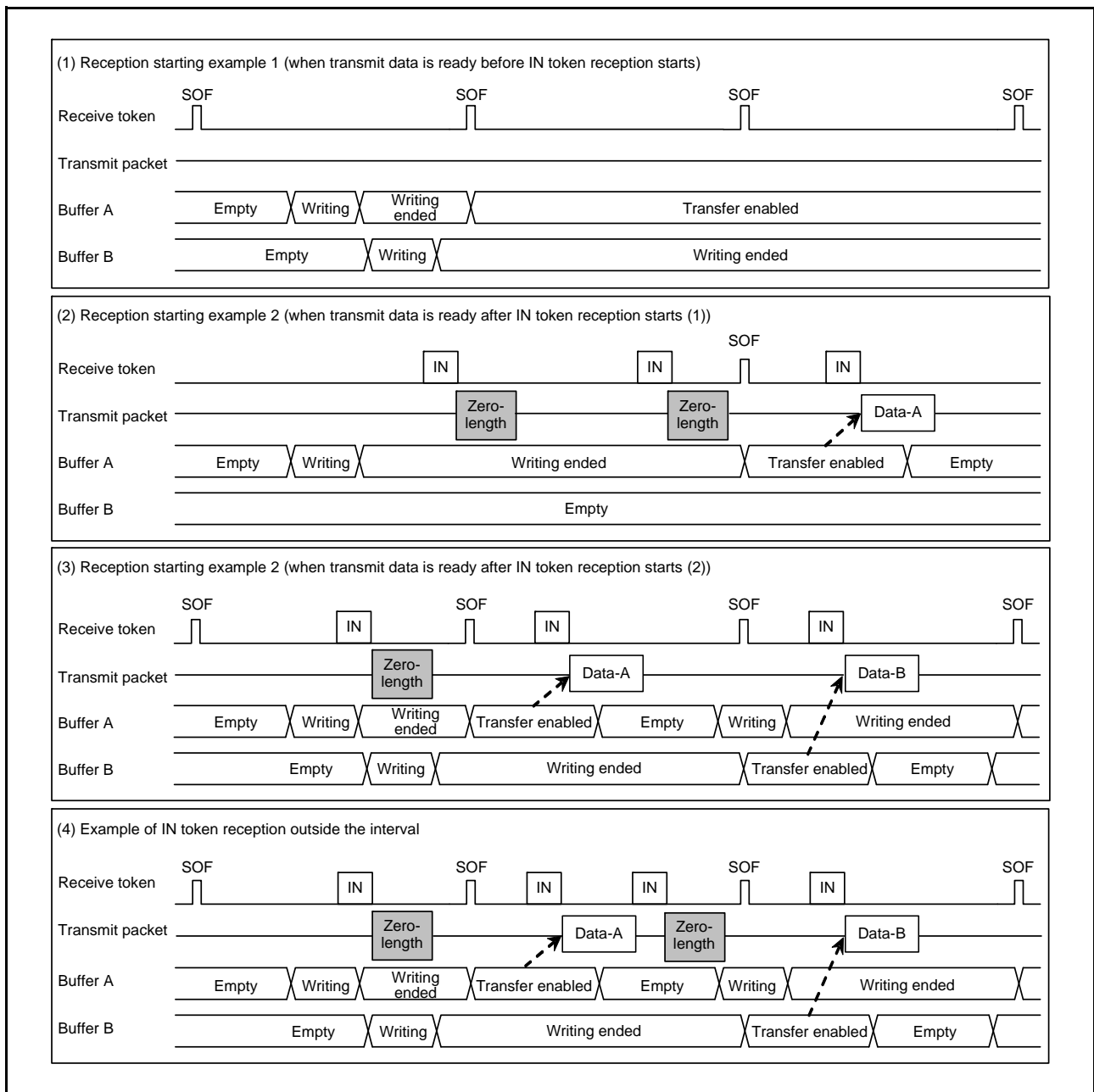


Figure 31.15 Example of Data Setup Function Operation

#### (4) Isochronous Transfer Transmission Buffer Flush

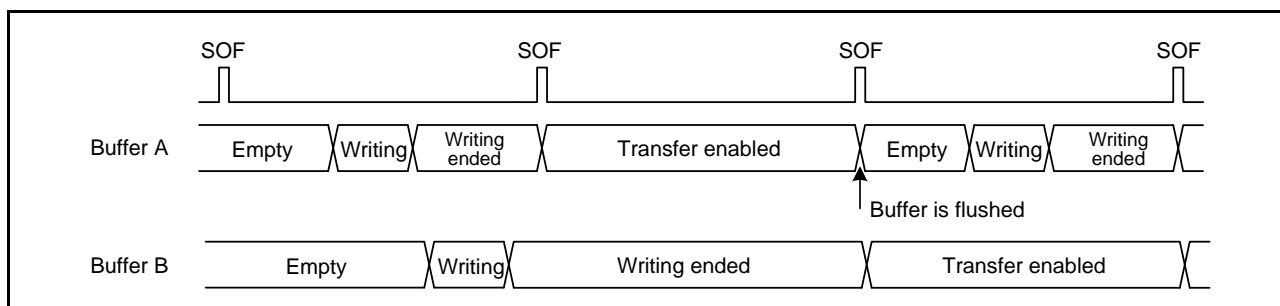
If an SOF packet of the next frame is received without receiving an IN token in an interval frame during isochronous data transmission, the USB module operates as if an IN token had been corrupted, and clears the buffer for which transmission is enabled, putting that buffer in the writing enabled state.

If a double buffer configuration is used and writing to both buffers has been completed, the buffer memory that was cleared is assumed as the data having been sent in the interval frame, and transmission is enabled for the buffer memory that is not cleared with SOF packet reception.

The timing of the buffer flush function depends on the PIPEPERI.IITV[2:0] bit setting.

- When the IITV = 0  
The buffer flush operation starts from the next frame after the pipe becomes valid.
- When the IITV  $\neq$  0  
The buffer flush operation is carried out after the first successful transaction.

Figure 31.16 shows an example of the buffer flush function in the USB module. When an unanticipated token is received before the interval frame, the USB module sends the write data or a zero-length packet as an underrun error according to the data setup state.



**Figure 31.16 Example of Buffer Flush Operation**

Figure 31.17 shows an example of interval error occurrence in the USB module. There are five types of interval errors, as shown below. The interval error is generated at the timing indicated by (1) in the figure, and the buffer flush function is activated.

If an interval error occurs during an IN transfers, the buffer flush function is activated; if it occurs during an OUT transfer, an NRDY interrupt is generated.

The OVRN bit in FRMNUM should be used to distinguish between NRDY interrupts such as received packet errors and overrun errors.

In response to tokens that are shaded in the figure, responses are sent according to the buffer memory status.

IN direction

- If the buffer is in the transmission enabled state, the data is transferred as a normal response.
- If the buffer is in the transmission disabled state, a zero-length packet is sent and an underrun error occurs.

OUT direction

- If the buffer is in the reception enabled state, the data is received as a normal response.
- If the buffer is in the reception disabled state, the data is discarded and an overrun error occurs.



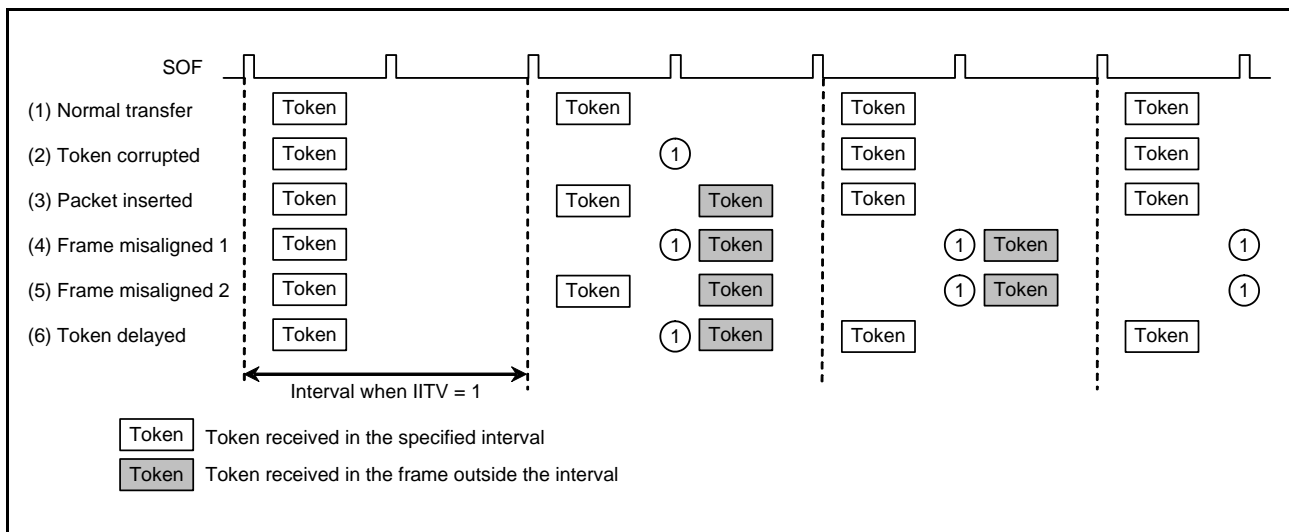


Figure 31.17 Example of Interval Error Occurrence when IITV = 1

### 31.3.10 SOF Interpolation Function

If data could not be received at intervals of 1 ms because an SOF packet was corrupted or missing, the USB module interpolates the SOF. The SOF interpolation operation begins when the USBE and SCKE bits in SYSCFG have been set to 1 and an SOF packet is received. The interpolation function is initialized under the following conditions.

- Power-on reset
- USB bus reset
- Suspended state detected

The SOF interpolation operates as follows.

- The interpolation function is not activated until an SOF packet is received.
- After the first SOF packet is received, interpolation is carried out by counting 1 ms with an internal clock of 48 MHz.
- After the second and subsequent SOF packets are received, interpolation is carried out at the previous reception interval.
- Interpolation is not carried out in the suspended state or while a USB bus reset is being received.

The USB module supports the following functions based on the SOF packet reception. These functions also operate normally with SOF interpolation, if the SOF packet was missing.

- Updating of the frame number
- SOFR interrupt timing
- Isochronous transfer interval count

If an SOF packet is missing when full-speed operation is being used, the FRMNUM.FRNM[10:0] bits are not updated.

## 31.4 Usage Notes

### 31.4.1 Settings for the Module-Stop State

The module-stop control register can be used to stop and start USB operations. The initial setting is for USB operations to be stopped. The registers of the module only become accessible after release from the module-stop state. For details, refer to section 11, Low Power Consumption.

## 32. Serial Communications Interface (SC1c, SC1d)

The RX630 Group has 13 independent serial communications interface (SCI) channels. The SCI is configured as SC1c module (SCI0 to SCI11) and SC1d module (SCI12).

The SC1c (SCI0 to SCI11) 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). Single-master operation as a simple I<sup>2</sup>C bus interface and simple SPI interfaces are also supported.

As well as the functions of the SC1c module, the SC1d module (SCI12) supports an extended serial protocol with a structure formed from Start Frames and Information Frames.

### 32.1 Overview

Table 32.1 lists the specifications of the SC1c module, Table 32.2 lists the specifications of the SC1d module, and Table 32.3 lists the specifications of the individual SCI channels.

Figure 32.1 is a block depicting SCI0 to SC11 (but not SCI5 and SCI6). Figure 32.2 is a block diagram depicting SCI5 and SCI6. Figure 32.3 is a block diagram depicting SCI12 (the SC1d module).

**Table 32.1 Specifications of SC1c (1/2)**

Item	Specifications	
Serial communications mode	<ul style="list-style-type: none"> <li>Asynchronous operation</li> <li>Clock synchronous operation</li> <li>Smart card interface</li> <li>Simple I<sup>2</sup>C bus</li> <li>Simple SPI bus</li> </ul>	
Transfer speed	Bit rate specifiable with on-chip baud rate generator.	
Full-duplex communications	Transmitter: Enables continuous transmission by double-buffering. Receiver: Enables continuous reception by double-buffering.	
I/O pins	See Table 32.4 to Table 32.6.	
Data transfer	Selectable as LSB-first or MSB-first transfer*2	
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 <sup>2</sup> C mode)	
Power consumption reduction function	Module-stop state can be set for each channel.	
Asynchronous mode	Data length	7 or 8 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even, odd, or none
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTS <sub>n</sub> and RTS <sub>n</sub> pins can be used in transfer control.
	Break detection	Break can be detected by reading RXD <sub>n</sub> pin level directly in case of a framing error
	Clock source	Selectable from internal or external clock Enables transfer rate clock input from TMR (SCI5 and SCI6)
	Multi-processor communications function	Serial communication among multiple processors
Clock synchronous mode	Noise cancellation	The signal paths from input on the RXD <sub>n</sub> pins incorporate digital noise filters.
	Data length	8 bits
	Receive error detection	Overrun errors
	Hardware flow control	CTS <sub>n</sub> and RTS <sub>n</sub> pins can be used in transfer control.

**Table 32.1 Specifications of SC1c (2/2)**

Item	Specifications	
Smart card interface mode	Error processing	An error signal can be automatically transmitted on detection of a parity error during reception
		Data can be automatically re-transmitted on receiving an error signal during transmission
Simple I <sup>2</sup> C mode	Data type	Both direct convention and inverse convention are supported.
	Transfer format	I <sup>2</sup> C bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Up to 384 kbps
	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	Eight bits
	Detection of errors	Overrun errors
	SS input pin function*1	Applying the high level to the SS# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock sense are selectable.

Note 1. The SS input pin function in simple SPI mode is not available in products with the on-chip ROM capacity of 1.5 Mbytes or more, or with 176 pins or more.

Note 2. In simple I<sup>2</sup>C mode, only MSB-first is available.

**Table 32.2 Specifications of SC1d (1/2)**

Item	Specifications	
Serial communications mode	<ul style="list-style-type: none"> <li>Asynchronous operation</li> <li>Clock synchronous operation</li> <li>Smart card interface</li> <li>Simple I<sup>2</sup>C bus</li> <li>Simple SPI bus</li> </ul>	
Transfer speed	Bit rate specifiable with on-chip baud rate generator.	
Full-duplex communications	Transmitter: Enables continuous transmission by double-buffering. Receiver: Enables continuous reception by double-buffering.	
Input/output pins	See Table 32.4 to Table 32.7.	
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 <sup>2</sup> C mode)	
Power consumption reduction function	Module-stop state can be set.	
Asynchronous mode	Data length	7 or 8 bits
	Transmission stop bit	1 or 2 bits
	Parity	Even, odd, or none
	Receive error detection	Parity, overrun, and framing errors
	Hardware flow control	CTSn and RTSn pins can be used in transfer control.
	Break detection	Break can be detected by reading RXDn pin level directly in case of a framing error
	Clock source	Selectable from internal or external clock Enables transfer rate clock input from TMR
	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 errors
	Hardware flow control	CTSn and RTSn pins can be used in transfer control.

**Table 32.2 Specifications of SCId (2/2)**

Item		Specifications
Smart card interface mode	Error processing	An error signal can be automatically transmitted on detection of a parity error during reception Data can be automatically re-transmitted on receiving an error signal during transmission
	Data type	Both direct convention and inverse convention are supported.
Simple I <sup>2</sup> C mode	Transfer format	I <sup>2</sup> C bus format
	Operating mode	Master (single-master operation only)
	Transfer rate	Up to 384 kbps
	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 errors
	SS input pin function	Applying the high level to the SS# pin can cause the output pins to enter the high-impedance state.
	Clock settings	Four kinds of settings for clock phase and clock sense are selectable.
Extended serial mode	Start Frame transmission	<ul style="list-style-type: none"> <li>Output of a low level as the Break Field over a specified width and generation of interrupts on completion</li> <li>Detection of bus collisions and the generation of interrupts on detection</li> </ul>
	Start Frame reception	<ul style="list-style-type: none"> <li>Detection of the Break Field low width and generation of an interrupt on detection</li> <li>Comparison of Control Fields 0 and 1 and generation of an interrupt when the two match</li> <li>Two kinds of data for comparison (primary and secondary) can be set in Control Field 1.</li> <li>A priority interrupt bit can be set in Control Field 1.</li> <li>Handling of Start Frames that do not include a Break Field</li> <li>Handling of Start Frames that do not include a Control Field</li> <li>Function for measuring bit rates</li> </ul>
	I/O control function	<ul style="list-style-type: none"> <li>Selectable polarity for TXDX12 and RXDX12 signals</li> <li>Selection of a digital filter for RXDX12</li> <li>Half-duplex operation employing RXDX12 and TXDX12 signals multiplexed on the same pin</li> <li>Selectable timing for the sampling of data received through RXDX12</li> <li>Signals received on RXDX12 can be passed though to SCIc when the extended serial mode control section is off.</li> </ul>
	Timer function	<ul style="list-style-type: none"> <li>Usable as a reloading timer</li> </ul>

Note 1. In simple I<sup>2</sup>C mode, only MSB-first is available.

**Table 32.3 List of Functions of SCI Channels**

Item	SCI0 to SCI4, SCI7 to SCI11	SCI5, SCI6	SCI12
Asynchronous mode	○	○	○
Clock synchronous mode	○	○	○
Smart card interface mode	○	○	○
Simple I <sup>2</sup> C mode	○	○	○
Simple SPI bus	○	○	○
Extended serial mode	—	—	○
TMR clock input	—	○	○

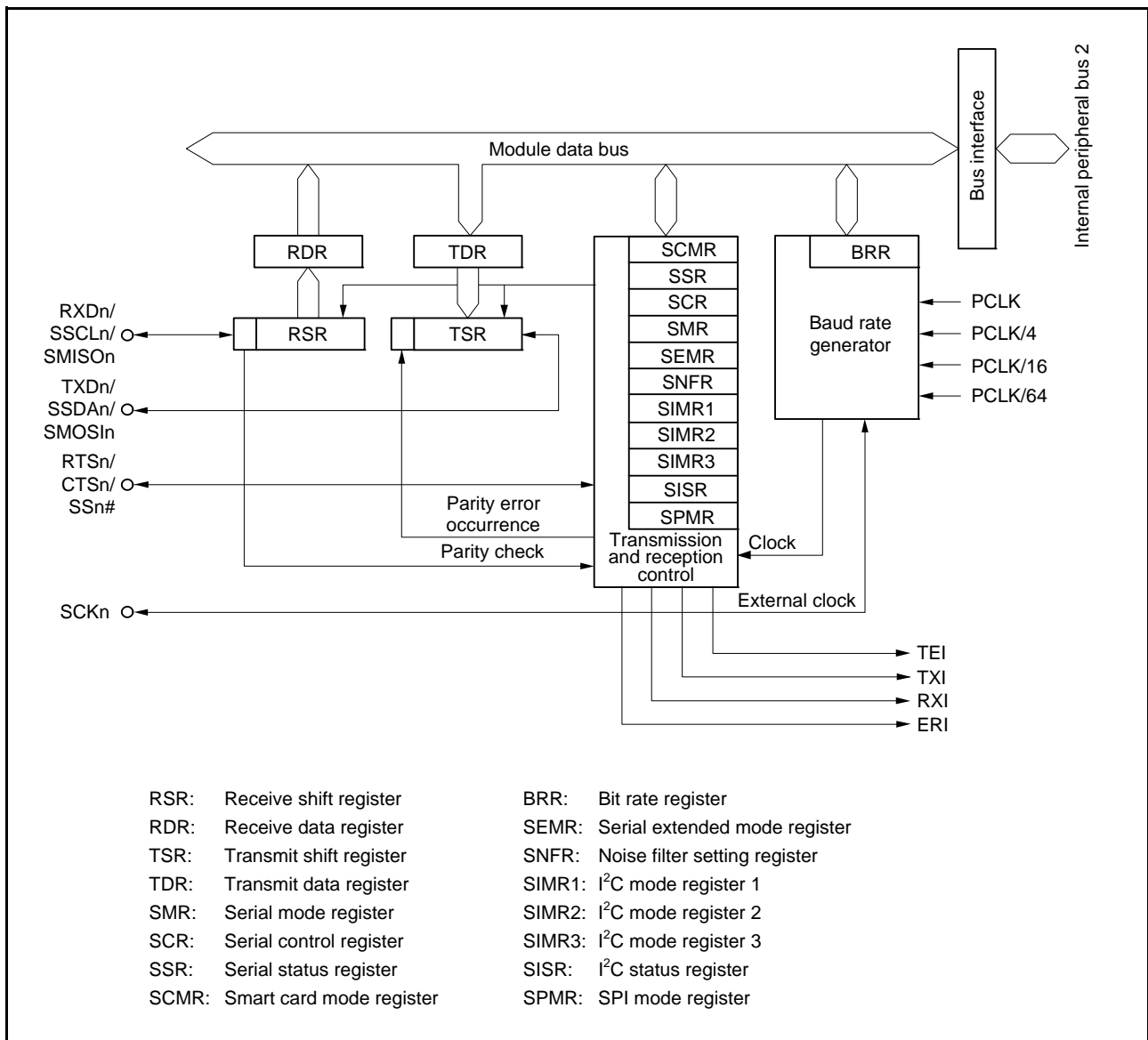


Figure 32.1 Block Diagram of SCI0 to SCI4, SCI7 to SCI11

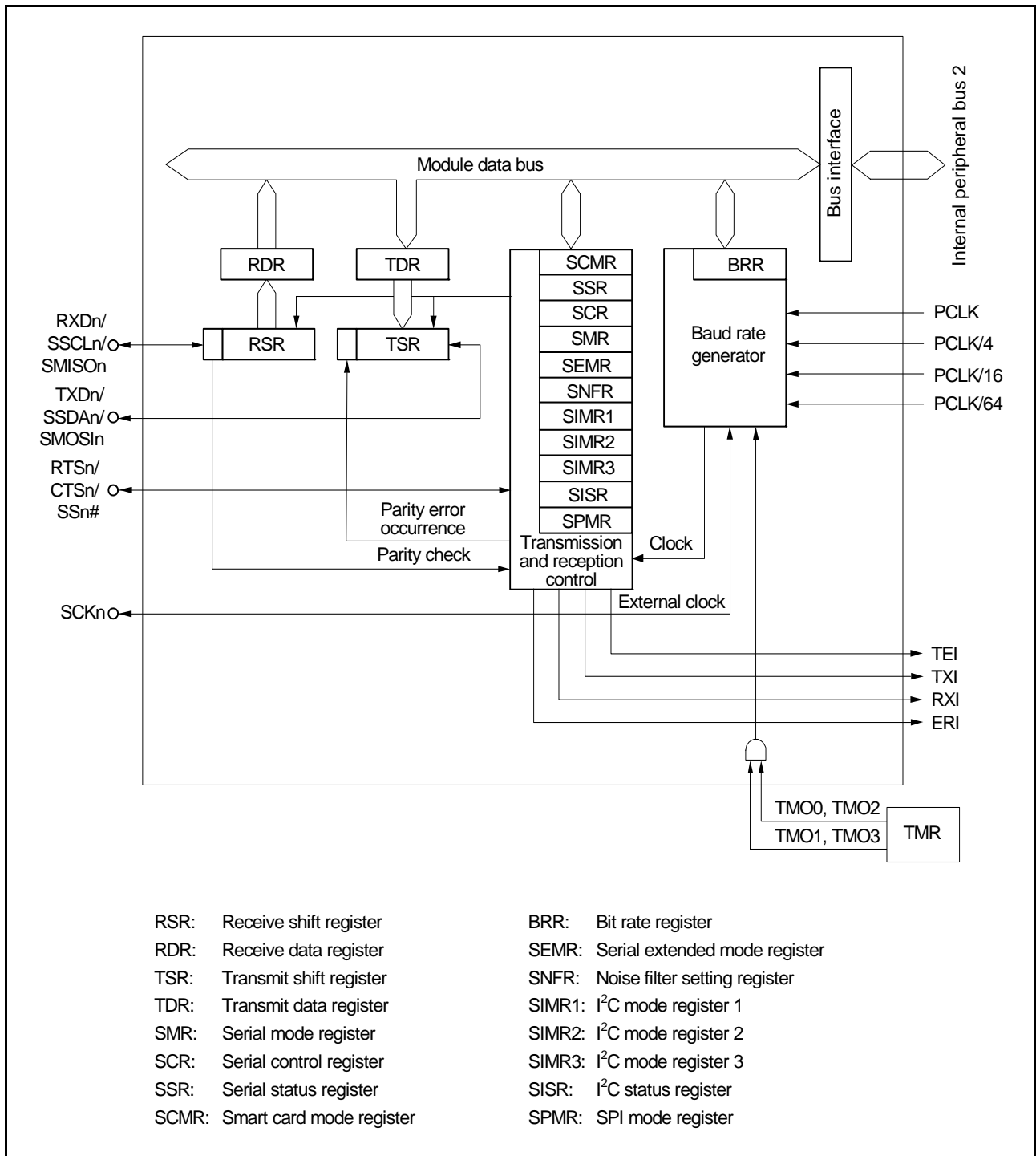


Figure 32.2 Block Diagram of SCI5 and SCI6

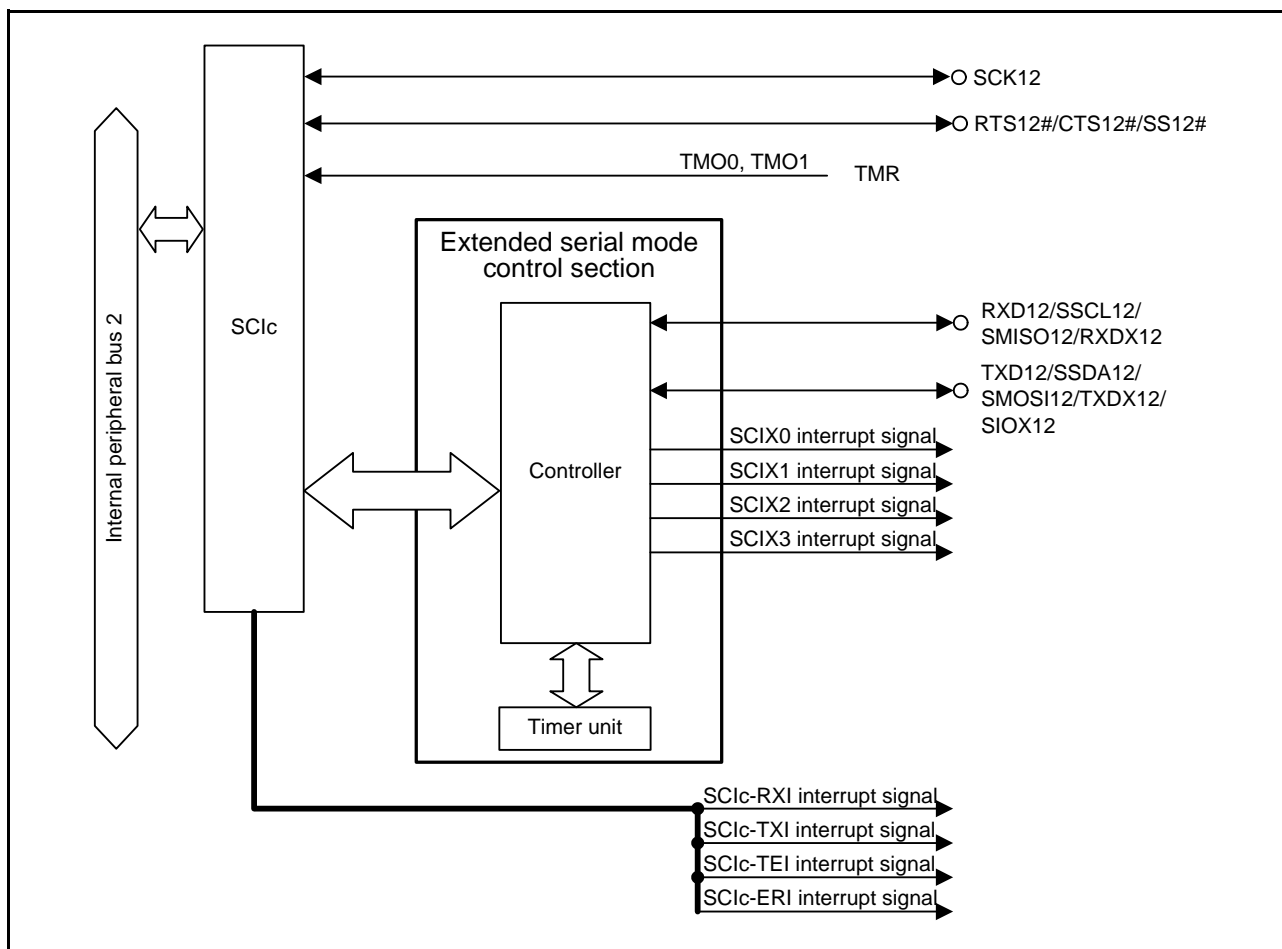


Figure 32.3 Block Diagram of SCI2 (SCId)

Table 32.4 to Table 32.7 list the pin configuration of the SCIs for the individual modes.

Table 32.4 Input and Output Pins of the SCIs (Asynchronous/Clock Synchronous Modes) (1/2)

Channel	Pin Name	I/O	Function
SCI0	SCK0	I/O	SCI0 clock input/output
	RXD0	Input	SCI0 receive data input
	TXD0	Output	SCI0 transmit data output
	CTS0#/RTS0#	I/O	SCI0 transfer start control input/output
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
SCI2	SCK2	I/O	SCI2 clock input/output
	RXD2	Input	SCI2 receive data input
	TXD2	Output	SCI2 transmit data output
	CTS2#/RTS2#	I/O	SCI2 transfer start control input/output
SCI3	SCK3	I/O	SCI3 clock input/output
	RXD3	Input	SCI3 receive data input
	TXD3	Output	SCI3 transmit data output
	CTS3#/RTS3#	I/O	SCI3 transfer start control input/output

**Table 32.4 Input and Output Pins of the SCIs (Asynchronous/Clock Synchronous Modes) (2/2)**

Channel	Pin Name	I/O	Function
SCI4	SCK4	I/O	SCI4 clock input/output
	RXD4	Input	SCI4 receive data input
	TXD4	Output	SCI4 transmit data output
	CTS4#/RTS4#	I/O	SCI4 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
SCI6	SCK6	I/O	SCI6 clock input/output
	RXD6	Input	SCI6 receive data input
	TXD6	Output	SCI6 transmit data output
	CTS6#/RTS6#	I/O	SCI6 transfer start control input/output
SCI7	SCK7	I/O	SCI7 clock input/output
	RXD7	Input	SCI7 receive data input
	TXD7	Output	SCI7 transmit data output
	CTS7#/RTS7#	I/O	SCI7 transfer start control input/output
SCI8	SCK8	I/O	SCI8 clock input/output
	RXD8	Input	SCI8 receive data input
	TXD8	Output	SCI8 transmit data output
	CTS8#/RTS8#	I/O	SCI8 transfer start control input/output
SCI9	SCK9	I/O	SCI9 clock input/output
	RXD9	Input	SCI9 receive data input
	TXD9	Output	SCI9 transmit data output
	CTS9#/RTS9#	I/O	SCI9 transfer start control input/output
SCI10	SCK10	I/O	SCI10 clock input/output
	RXD10	Input	SCI10 receive data input
	TXD10	Output	SCI10 transmit data output
	CTS10#/RTS10#	I/O	SCI10 transfer start control input/output
SCI11	SCK11	I/O	SCI11 clock input/output
	RXD11	Input	SCI11 receive data input
	TXD11	Output	SCI11 transmit data output
	CTS11#/RTS11#	I/O	SCI11 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 32.5 Pin Configuration of SCI (Simple I<sup>2</sup>C Mode) (1/2)**

Channel	Pin Name	I/O	Function
SCI0	SSCL0	I/O	SCI0 I <sup>2</sup> C clock input/output
	SSDA0	I/O	SCI0 I <sup>2</sup> C data input/output
SCI1	SSCL1	I/O	SCI1 I <sup>2</sup> C clock input/output
	SSDA1	I/O	SCI1 I <sup>2</sup> C data input/output
SCI2	SSCL2	I/O	SCI2 I <sup>2</sup> C clock input/output
	SSDA2	I/O	SCI2 I <sup>2</sup> C data input/output



**Table 32.5 Pin Configuration of SCI (Simple I<sup>2</sup>C Mode) (2/2)**

Channel	Pin Name	I/O	Function
SCI3	SSCL3	I/O	SCI3 I <sup>2</sup> C clock input/output
	SSDA3	I/O	SCI3 I <sup>2</sup> C data input/output
SCI4	SSCL4	I/O	SCI4 I <sup>2</sup> C clock input/output
	SSDA4	I/O	SCI4 I <sup>2</sup> C data input/output
SCI5	SSCL5	I/O	SCI5 I <sup>2</sup> C clock input/output
	SSDA5	I/O	SCI5 I <sup>2</sup> C data input/output
SCI6	SSCL6	I/O	SCI6 I <sup>2</sup> C clock input/output
	SSDA6	I/O	SCI6 I <sup>2</sup> C data input/output
SCI7	SSCL7	I/O	SCI7 I <sup>2</sup> C clock input/output
	SSDA7	I/O	SCI7 I <sup>2</sup> C data input/output
SCI8	SSCL8	I/O	SCI8 I <sup>2</sup> C clock input/output
	SSDA8	I/O	SCI8 I <sup>2</sup> C data input/output
SCI9	SSCL9	I/O	SCI9 I <sup>2</sup> C clock input/output
	SSDA9	I/O	SCI9 I <sup>2</sup> C data input/output
SCI10	SSCL10	I/O	SCI10 I <sup>2</sup> C clock input/output
	SSDA10	I/O	SCI10 I <sup>2</sup> C data input/output
SCI11	SSCL11	I/O	SCI11 I <sup>2</sup> C clock input/output
	SSDA11	I/O	SCI11 I <sup>2</sup> C data input/output
SCI12	SSCL12	I/O	SCI12 I <sup>2</sup> C clock input/output
	SSDA12	I/O	SCI12 I <sup>2</sup> C data input/output

**Table 32.6 Pin Configuration of SCI (Simple SPI Mode) (1/2)**

Channel	Pin Name	I/O	Function
SCI0	SCK0	I/O	SCI0 clock input/output
	SMISO0	I/O	SCI0 slave transmit data input/output
	SMOSI0	I/O	SCI0 master transmit data input/output
	SS0#	Input	SCI0 chip select input
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
SCI2	SCK2	I/O	SCI2 clock input/output
	SMISO2	I/O	SCI2 slave transmit data input/output
	SMOSI2	I/O	SCI2 master transmit data input/output
	SS2#	Input	SCI2 chip select input
SCI3	SCK3	I/O	SCI3 clock input/output
	SMISO3	I/O	SCI3 slave transmit data input/output
	SMOSI3	I/O	SCI3 master transmit data input/output
	SS3#	Input	SCI3 chip select input
SCI4	SCK4	I/O	SCI4 clock input/output
	SMISO4	I/O	SCI4 slave transmit data input/output
	SMOSI4	I/O	SCI4 master transmit data input/output
	SS4#	Input	SCI4 chip select input

**Table 32.6 Pin Configuration of SCI (Simple SPI Mode) (2/2)**

Channel	Pin Name	I/O	Function
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
SCI6	SCK6	I/O	SCI6 clock input/output
	SMISO6	I/O	SCI6 slave transmit data input/output
	SMOSI6	I/O	SCI6 master transmit data input/output
	SS6#	Input	SCI6 chip select input
SCI7	SCK7	I/O	SCI7 clock input/output
	SMISO7	I/O	SCI7 slave transmit data input/output
	SMOSI7	I/O	SCI7 master transmit data
	SS7#	Input	SCI7 chip select input
SCI8	SCK8	I/O	SCI8 clock input/output
	SMISO8	I/O	SCI8 slave transmit data input/output
	SMOSI8	I/O	SCI8 master transmit data input/output
	SS8#	Input	SCI8 chip select input
SCI9	SCK9	I/O	SCI9 clock input/output
	SMISO9	I/O	SCI9 slave transmit data input/output
	SMOSI9	I/O	SCI9 master transmit data input/output
	SS9#	Input	SCI9 chip select input
SCI10	SCK10	I/O	SCI10 clock input/output
	SMISO10	I/O	SCI10 slave transmit data input/output
	SMOSI10	I/O	SCI10 master transmit data input/output
	SS10#	Input	SCI10 chip select input
SCI11	SCK11	I/O	SCI11 clock input/output
	SMISO11	I/O	SCI11 slave transmit data input/output
	SMOSI11	I/O	SCI11 master transmit data input/output
	SS11#	Input	SCI11 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 32.7 Pin Configuration of SCI (Extended Serial Mode)**

Channel	Pin Name	I/O	Function
SCI12	RDX12	Input	SCI12 receive data input
	TDX12	Output	SCI12 transmit data output
	SIOX12	I/O	SCI12 transfer data input/output

## 32.2 Register Descriptions

### 32.2.1 Receive Shift Register (RSR)

RSR is a shift register which is used to receive serial data input from the RXDn pin and converts it into parallel data. When one frame of data has been received, it is transferred to RDR automatically.

RSR cannot be directly accessed by the CPU.

### 32.2.2 Receive Data Register (RDR)

Address(es): SC10.RDR 0008 A005h, SC11.RDR 0008 A025h, SC12.RDR 0008 A045h, SC13.RDR 0008 A065h,  
SC14.RDR 0008 A085h, SC15.RDR 0008 A0A5h, SC16.RDR 0008 A0C5h, SC17.RDR 0008 A0E5h,  
SC18.RDR 0008 A105h, SC19.RDR 0008 A125h, SC110.RDR 0008 A145h, SC111.RDR 0008 A165h,  
SC112.RDR 0008 B305h



RDR is an 8-bit register that stores receive data.

When the SCI has received one frame of serial data, it transfers the received serial data from RSR to RDR where it is stored. This allows RSR to receive the next data.

Since RSR and RDR function as a double buffer in this way, continuous receive operations can be performed.

Read RDR 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 RDR, an overrun error occurs.

RDR cannot be written to by the CPU.

### 32.2.3 Transmit Data Register (TDR)

Address(es): SC10.TDR 0008 A003h, SC11.TDR 0008 A023h, SC12.TDR 0008 A043h, SC13.TDR 0008 A063h,  
SC14.TDR 0008 A083h, SC15.TDR 0008 A0A3h, SC16.TDR 0008 A0C3h, SC17.TDR 0008 A0E3h,  
SC18.TDR 0008 A103h, SC19.TDR 0008 A123h, SC110.TDR 0008 A143h, SC111.TDR 0008 A163h,  
SC112.TDR 0008 B303h



TDR is an 8-bit register that stores transmit data.

When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts transmission.

The double-buffered structures of TDR and TSR enable continuous serial transmission. If the next transmit data has already been written to TDR when one frame of data is transmitted, the SCI transfers the written data to TSR to continue transmission.

The CPU is able to read from or write to TDR at any time. Only write data for transmission to TDR once after each instance of the transmit data empty interrupt (TXI).

### 32.2.4 Transmit Shift Register (TSR)

TSR is a shift register that transmits serial data.

To perform serial data transmission, the SCI first automatically transfers transmit data from TDR to TSR, and then sends the data to the TXDn pin.

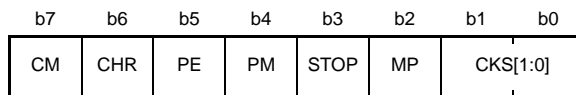
TSR cannot be directly accessed by the CPU.

### 32.2.5 Serial Mode Register (SMR)

Note: • Some bits in SMR have different functions in serial communications interface mode and smart card interface mode.

#### (1) Serial Communications Interface Mode (SMIF in SCMR = 0)

Address(es): SCI0.SMR 0008 A000h, SCI1.SMR 0008 A020h, SCI2.SMR 0008 A040h, SCI3.SMR 0008 A060h, SCI4.SMR 0008 A080h, SCI5.SMR 0008 A0A0h, SCI6.SMR 0008 A0C0h, SCI7.SMR 0008 A0E0h, SCI8.SMR 0008 A100h, SCI9.SMR 0008 A120h, SCI10.SMR 0008 A140h, SCI11.SMR 0008 A160h, SCI12.SMR 0008 B300h



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 clock (n = 0)*1 0 1: PCLK/4 clock (n = 1)*1 1 0: PCLK/16 clock (n = 2)*1 1 1: PCLK/64 clock (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 in asynchronous mode) 0: Selects even parity 1: Selects odd parity	R/W*4
b5	PE	Parity Enable	(Valid only in asynchronous mode) <ul style="list-style-type: none"> <li>When transmitting               <ul style="list-style-type: none"> <li>0: Parity bit addition is not performed</li> <li>1: The parity bit is added</li> </ul> </li> <li>When receiving               <ul style="list-style-type: none"> <li>0: Parity bit checking is not performed</li> <li>1: The parity bit is checked</li> </ul> </li> </ul>	R/W*4
b6	CHR	Character Length	(Valid only in asynchronous mode) 0: Selects 8 bits as the data length*2 1: Selects 7 bits as the data length*3	R/W*4
b7	CM	Communications Mode	0: Asynchronous mode 1: Clock synchronous mode	R/W*4

Note 1. n is the decimal notation of the value of n in BRR (see section 32.2.9, Bit Rate Register (BRR)).

Note 2. In clock synchronous 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 TDR is not transmitted in transmission.

Note 4. Writable only when TE in SCR = 0 and RE in SCR = 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, see section 32.2.9, 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.

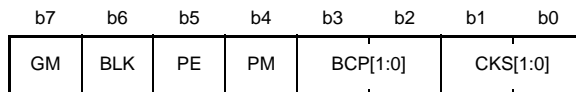
**CHR Bit (Character Length)**

Selects the data length for transmission and reception.

In clock synchronous mode, a fixed data length of 8 bits is used.

**(2) Smart Card Interface Mode (SMIF in SCMR = 1)**

Address(es): SC10.SMR 0008 A000h, SC11.SMR 0008 A020h, SC12.SMR 0008 A040h, SC13.SMR 0008 A060h, SC14.SMR 0008 A080h, SC15.SMR 0008 A0A0h, SC16.SMR 0008 A0C0h, SC17.SMR 0008 A0E0h, SC18.SMR 0008 A100h, SC19.SMR 0008 A120h, SC110.SMR 0008 A140h, SC111.SMR 0008 A160h, SC112.SMR 0008 B300h



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 clock (n = 0)*1 0 1: PCLK/4 clock (n = 1)*1 1 0: PCLK/16 clock (n = 2)*1 1 1: PCLK/64 clock (n = 3)*1	R/W*3
b3, b2	BCP[1:0]	Base Clock Pulse	Selects the number of base clock cycles in combination with the BCP2 bit in SCMR. Setting values in BCP2 bit in SCMR and BCP[1:0] bits in SMR: BCP2 b3 b2 0 0 0: 93 clock cycles (S = 93)*2 0 0 1: 128 clock cycles (S = 128)*2 0 1 0: 186 clock cycles (S = 186)*2 0 1 1: 512 clock cycles (S = 512)*2 1 0 0: 32 clock cycles (S = 32)*2 (Initial value) 1 0 1: 64 clock cycles (S = 64)*2 1 1 0: 372 clock cycles (S = 372)*2 1 1 1: 256 clock cycles (S = 256)*2	R/W*3
b4	PM	Parity Mode	(Valid only when the PE bit is 1 in asynchronous mode) 0: Selects even parity 1: Selects odd parity	R/W*3

Bit	Symbol	Bit Name	Description	R/W
b5	PE	Parity Enable	When this bit is set to 1, a parity bit is added to data for transmission, and the parity of received data is checked. Set this bit to 1 in smart card interface mode.	R/W*3
b6	BLK	Block Transfer Mode	0: Normal mode operation 1: Block transfer mode operation	R/W*3
b7	GM	GSM Mode	0: Normal mode operation 1: GSM mode operation	R/W*3

Note 1. n is the decimal notation of the value of n in BRR (see section 32.2.9, Bit Rate Register (BRR)).

Note 2. S is the value of S in BRR (see section 32.2.9, Bit Rate Register (BRR)).

Note 3. Writable only when TE in SCR = 0 and RE in SCR = 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, see section 32.2.9, 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 BCP2 bit in SCMR.

For details, see section 32.6.4, Receive Data Sampling Timing and Reception Margin.

### 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, see section 32.6.2, 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, see section 32.6.3, Block Transfer Mode.

### GM Bit (GSM Mode)

Setting this bit to 1 allows GSM mode operation.

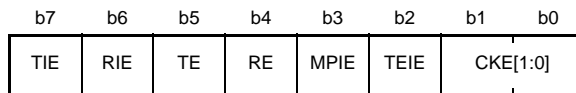
In GSM mode, the SSR.TEND flag set timing is put forward to 11.0 etu (elementary time unit = 1-bit transfer time) from the start and the clock output control function is appended. For details, see section 32.6.6, Serial Data Transmission (Except in Block Transfer Mode) and section 32.6.8, Clock Output Control.

### 32.2.6 Serial Control Register (SCR)

Note: • Some bits in SCR have different functions in serial communications interface mode and smart card interface mode.

#### (1) Serial Communications Interface Mode (SMIF in SCMR = 0)

Address(es): SCI0.SCR 0008 A002h, SCI1.SCR 0008 A022h, SCI2.SCR 0008 A042h, SCI3.SCR 0008 A062h, SCI4.SCR 0008 A082h, SCI5.SCR 0008 A0A2h, SCI6.SCR 0008 A0C2h, SCI7.SCR 0008 A0E2h, SCI8.SCR 0008 A102h, SCI9.SCR 0008 A122h, SCI10.SCR 0008 A142h, SCI11.SCR 0008 A162h, SCI12.SCR 0008 B302h



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	<ul style="list-style-type: none"> <li>• For SCI0 to SCI4, SCI7 to SCI11 (Asynchronous mode)               <ul style="list-style-type: none"> <li>b1 b0</li> <li>0 0: On-chip baud rate generator The SCKn pin functions as I/O port.</li> <li>0 1: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin.</li> <li>1 x: External 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.</li> </ul> </li> <li>(Clock synchronous mode)               <ul style="list-style-type: none"> <li>b1 b0</li> <li>0 x: Internal clock The SCKn pin functions as the clock output pin.</li> <li>1 x: External clock The SCKn pin functions as the clock input pin.</li> </ul> </li> </ul>	R/W*1
b1, b0	CKE[1:0]	Clock Enable	<ul style="list-style-type: none"> <li>• For SCI5, SCI6 and SCI12 (Asynchronous mode)               <ul style="list-style-type: none"> <li>b1 b0</li> <li>0 0: On-chip baud rate generator The SCKn pin functions as I/O port.</li> <li>0 1: On-chip baud rate generator The clock with the same frequency as the bit rate is output from the SCKn pin.</li> <li>1 x: External clock or TMR clock                   <ul style="list-style-type: none"> <li>• When an external clock is used, 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.</li> <li>• The TMR clock can be used.</li> </ul> </li> </ul> </li> <li>(Clock synchronous mode)               <ul style="list-style-type: none"> <li>b1 b0</li> <li>0 x: Internal clock The SCKn pin functions as the clock output pin.</li> <li>1 x: External clock The SCKn pin functions as the clock input pin.</li> </ul> </li> </ul>	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

Bit	Symbol	Bit Name	Description	R/W
b3	MPIE	Multi-Processor Interrupt Enable	(Valid in asynchronous mode when SMR.MP = 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 ORER and FER in SSR 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. A 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 in 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 TMR clock.

### TEIE Bit (Transmit End Interrupt Enable)

Enables or disables a TEI interrupt request.

A TEI interrupt request is disabled by clearing the TEIE bit to 0.

In simple I<sup>2</sup>C mode (when the IICM bit in SIMR is 1), 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 ORER and FER in SSR to 1 is disabled. When the data with the multi-processor bit set to 1 is received, the MPIE is automatically cleared to 0, and normal reception is resumed. For details, see section 32.4, Multi-Processor Communications Function.

When the receive data includes the MPB bit is SSR set to 0, the receive data is not transferred from the RSR to the RDR, a receive error is not detected, and setting the flags ORER and FER to 1 is disabled.

When the receive data includes the MPB bit set to 1, the MPB bit is set to 1, the MPIE bit is automatically cleared to 0, the RXI and ERI interrupt requests are enabled (if the RIE bit in SCR is set to 1), and setting the flags ORER and FER to 1 is enabled.

MPIE should be set 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 SMR should be set prior to setting the RE bit to 1 in order to designate the reception format.

Even if reception is halted by clearing the RE bit to 0, the ORER, FER, and PER flags in SSR 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 TDR. Note that SMR 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 clearing the RIE bit to 0.

An ERI interrupt request can be cancelled by reading 1 from the ORER, FER, or PER flag in SSR and then clearing the flag to 0, or clearing the RIE bit to 0.

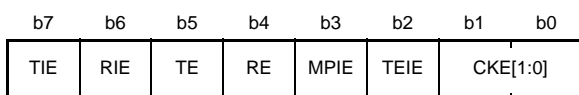
**TIE Bit (Transmit Interrupt Enable)**

Enables or disables notification of a TXI interrupt request.

Notification of a TXI interrupt request is disabled by clearing the TIE bit to 0.

**(2) Smart Card Interface Mode (SMIF in SCMR = 1)**

Address(es): SCI0.SCR 0008 A002h, SCI1.SCR 0008 A022h, SCI2.SCR 0008 A042h, SCI3.SCR 0008 A062h, SCI4.SCR 0008 A082h, SCI5.SCR 0008 A0A2h, SCI6.SCR 0008 A0C2h, SCI7.SCR 0008 A0E2h, SCI8.SCR 0008 A102h, SCI9.SCR 0008 A122h, SCI10.SCR 0008 A142h, SCI11.SCR 0008 A162h, SCI12.SCR 0008 B302h



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	<ul style="list-style-type: none"> <li>• When GM in SMR = 0                             <ul style="list-style-type: none"> <li>b1 b0</li> <li>0 0: Output disabled (The SCKn pin is available for use as an I/O port in accord with the I/O port settings.)</li> <li>0 1: Clock output</li> <li>1 x: (Setting prohibited)</li> </ul> </li> <li>• When GM in SMR = 1                             <ul style="list-style-type: none"> <li>b1 b0</li> <li>0 0: Output fixed low</li> <li>x 1: Clock output</li> <li>1 0: Output fixed high</li> </ul> </li> </ul>	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. A 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 in TE and RE. While the SMR.CM bit is 0, writing is enabled under any condition.

For details on interrupt requests, see section 32.11, 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, see section 32.6.8, 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 SMR should be set prior to setting the RE bit to 1 in order to designate the reception format.

Even if reception is halted by clearing the RE bit to 0, the ORER, FER, and PER flags in SSR 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 TDR. Note that SMR 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 clearing the RIE bit to 0.

An ERI interrupt request can be cancelled by reading 1 from the ORER, FER, or PER flag in SSR and then clearing the flag to 0, or clearing the RIE bit to 0.

**TIE Bit (Transmit Interrupt Enable)**

Enables or disables notification of a TXI interrupt request.

Notification of a TXI interrupt request is disabled by clearing the TIE bit to 0.

### 32.2.7 Serial Status Register (SSR)

Note: • Some bits in SSR have different functions in serial communications interface mode and smart card interface mode.

#### (1) Serial Communications Interface Mode (SMIF in SCMR = 0)

Address(es): SCI0.SSR 0008 A004h, SCI1.SSR 0008 A024h, SCI2.SSR 0008 A044h, SCI3.SSR 0008 A064h, SCI4.SSR 0008 A084h, SCI5.SSR 0008 A0A4h, SCI6.SSR 0008 A0C4h, SCI7.SSR 0008 A0E4h, SCI8.SSR 0008 A104h, SCI9.SSR 0008 A124h, SCI10.SSR 0008 A144h, SCI11.SSR 0008 A164h, SCI12.SSR 0008 B304h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	ORER	FER	PER	TEND	MPB	MPBT
Value after reset:	x	x	0	0	0	1	0	0

x: Undefined

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
b7, b6	—	Reserved	The read value is undefined. The write value should be 1.	R/W

Note 1. Only 0 can be written to this bit, to clear the flag.

#### MPB Bit (Multi-Processor)

Holds the value of the multi-processor bit in the reception frame. This bit does not change when the RE bit in SCR is 0.

#### TEND Flag (Transmission End Flag)

Indicates completion of transmission.

[Setting conditions]

- Clearing of the SCR.TE bit to 0 (disabling serial transmission operations)  
When the SCR.TE bit is changed from 0 to 1, the TEND flag is not affected and retains the value 1.
- The TDR is not updated at the time of transmission of the tail-end bit of a character being transmitted

[Clearing condition]

- When data for transmission are written to the TDR while the SCR.TE is 1  
When the TEND flag is cleared in response to writing of data for transmission to the TDR, dummy read the TEND flag before return from interrupt exception processing for TEI interrupt requests when the latter are enabled.

**PER Bit (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 RDR, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to PER after reading PER = 1 (after writing 0 to it, read the PER bit to check that it has actually been cleared to 0.)  
Even when the RE bit in SCR is cleared to 0 (which indicates that serial reception is disabled), the PER flag is not affected and retains its previous value.

**FER Bit (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 RDR, no RXI interrupt request occurs. In addition, when the FER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to FER after reading FER = 1 (after writing 0 to it, read the FER bit to check that it has actually been cleared to 0.)  
Even when the RE bit in SCR is cleared to 0, the FER flag is not affected and retains its previous value.

**ORER Bit (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 RDR  
In RDR, receive data prior to an overrun error occurrence is retained, but data received after the overrun error occurrence is lost. When the ORER flag is set to 1, subsequent serial reception cannot be performed. Note that, in clock synchronous mode, serial transmission also cannot continue.

[Clearing condition]

- When a 0 is written to ORER after reading ORER = 1 (after writing a 0 to it, read the ORER bit to check that it has actually been cleared to 0.)  
Even when the RE bit in SCR is cleared to 0, the ORER flag is not affected and retains its previous value.

## (2) Smart Card Interface Mode (SMIF in SCMR = 1)

Address(es): SC10.SSR 0008 A004h, SC11.SSR 0008 A024h, SC12.SSR 0008 A044h, SC13.SSR 0008 A064h,  
 SC14.SSR 0008 A084h, SC15.SSR 0008 A0A4h, SC16.SSR 0008 A0C4h, SC17.SSR 0008 A0E4h,  
 SC18.SSR 0008 A104h, SC19.SSR 0008 A124h, SC110.SSR 0008 A144h, SC111.SSR 0008 A164h,  
 SC112.SSR 0008 B304h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	ORER	ERS	PER	TEND	MPB	MPBT

Value after reset: x x 0 0 0 1 0 0

x: Undefined

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
b7, b6	—	Reserved	The read value is undefined. The write value should be 1.	R/W

Note 1. Only 0 can be written to this bit, to clear the flag.

**MPBT Bit (Multi-Processor Bit Transfer)**

This bit should be set to 0 in smart card interface mode.

**MPB Bit (Multi-Processor)**

This bit is not used in smart card interface mode. It should be set to 0.

**TEND Flag (Transmission 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 SCR.TE bit = 0 (disabling serial transmission operations)  
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 one 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 data for transmission are written to the TDR while the SCR.TE is 1

**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 RDR, no RXI interrupt request occurs. Note that when the PER flag is being set to 1, the subsequent receive data is not transferred to RDR.

[Clearing condition]

- When 0 is written to PER after reading PER = 1 (After writing 0 to it, read the PER bit to check that it has actually been cleared to 0.)

Even when the RE bit in SCR is cleared to 0 (which indicates that 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 ERS after reading ERS = 1

**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 RDR

In RDR, 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 ORER after reading ORER = 1 (After writing 0 to it, read the ORER bit to check that it has actually been cleared to 0.)

Even when the RE bit in SCR is cleared to 0, the ORER flag is not affected and retains its previous value.

### 32.2.8 Smart Card Mode Register (SCMR)

Address(es): SC10.SCMR 0008 A006h, SC11.SCMR 0008 A026h, SC12.SCMR 0008 A046h, SC13.SCMR 0008 A066h, SC14.SCMR 0008 A086h, SC15.SCMR 0008 A0A6h, SC16.SCMR 0008 A0C6h, SC17.SCMR 0008 A0E6h, SC18.SCMR 0008 A106h, SC19.SCMR 0008 A126h, SC110.SCMR 0008 A146h, SC111.SCMR 0008 A166h, SC112.SCMR 0008 B306h

b7	b6	b5	b4	b3	b2	b1	b0
BCP2	—	—	—	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: Serial communications interface 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	0: TDR contents are transmitted as they are. Receive data is stored as it is in RDR. 1: TDR contents are inverted before being transmitted. Receive data is stored in inverted form in RDR.	R/W*1																																				
b3	SDIR	Transmitted/Received Data Transfer Direction*2	0: Transfer with LSB-first 1: Transfer with MSB-first	R/W*1																																				
b6 to b4	—	Reserved	This bit is 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. Setting values in the SCMR.BCP2 bit and SMR.BCP[1:0] bits <table border="1"> <thead> <tr> <th>BCP2</th> <th>BCP1</th> <th>BCP0</th> <th></th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> <td>93 clock cycles (S = 93)*3</td> </tr> <tr> <td>0</td> <td>0</td> <td>1</td> <td>128 clock cycles (S = 128)*3</td> </tr> <tr> <td>0</td> <td>1</td> <td>0</td> <td>186 clock cycles (S = 186)*3</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> <td>512 clock cycles (S = 512)*3</td> </tr> <tr> <td>1</td> <td>0</td> <td>0</td> <td>32 clock cycles (S = 32)*3 (Initial Value)</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> <td>64 clock cycles (S = 64)*3</td> </tr> <tr> <td>1</td> <td>1</td> <td>0</td> <td>372 clock cycles (S = 372)*3</td> </tr> <tr> <td>1</td> <td>1</td> <td>1</td> <td>256 clock cycles (S = 256)*3</td> </tr> </tbody> </table>	BCP2	BCP1	BCP0		0	0	0	93 clock cycles (S = 93)*3	0	0	1	128 clock cycles (S = 128)*3	0	1	0	186 clock cycles (S = 186)*3	0	1	1	512 clock cycles (S = 512)*3	1	0	0	32 clock cycles (S = 32)*3 (Initial Value)	1	0	1	64 clock cycles (S = 64)*3	1	1	0	372 clock cycles (S = 372)*3	1	1	1	256 clock cycles (S = 256)*3	R/W*1
BCP2	BCP1	BCP0																																						
0	0	0	93 clock cycles (S = 93)*3																																					
0	0	1	128 clock cycles (S = 128)*3																																					
0	1	0	186 clock cycles (S = 186)*3																																					
0	1	1	512 clock cycles (S = 512)*3																																					
1	0	0	32 clock cycles (S = 32)*3 (Initial Value)																																					
1	0	1	64 clock cycles (S = 64)*3																																					
1	1	0	372 clock cycles (S = 372)*3																																					
1	1	1	256 clock cycles (S = 256)*3																																					

Note 1. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

Note 2. In simple I2C mode, set the SDIR bit to 1 (MSB-first).

Note 3. S is the value of S in BRR (see section 32.2.9, Bit Rate Register (BRR)).

#### 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, asynchronous or clock synchronous mode is selected.

#### SINV Bit (Transmitted/Received Data Invert)

Inverts the transmit/receive data logic level. This bit does not affect the logic level of the parity bit. To invert the parity bit, invert the PM bit in SMR.

#### SDIR Bit (Transmitted/Received Data Transfer Direction)

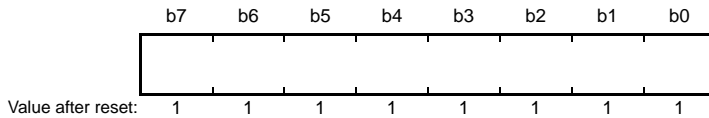
Selects the serial/parallel conversion format.

#### 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.

### 32.2.9 Bit Rate Register (BRR)

Address(es): SCI0.BRR 0008 A001h, SCI1.BRR 0008 A021h, SCI2.BRR 0008 A041h, SCI3.BRR 0008 A061h, SCI4.BRR 0008 A081h, SCI5.BRR 0008 A0A1h, SCI6.BRR 0008 A0C1h, SCI7.BRR 0008 A0E1h, SCI8.BRR 0008 A101h, SCI9.BRR 0008 A121h, SCI10.BRR 0008 A141h, SCI11.BRR 0008 A161h, SCI12.BRR 0008 B301h



BRR 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 32.8 lists the relationships between the setting (N) in the BRR and the bit rate (B) for normal asynchronous mode, multi-processor transfer, clock synchronous mode, smart card interface mode, simple SPI mode, and simple I<sup>2</sup>C mode.

The initial value of BRR is FFh.

BRR can be read from by the CPU at all times, but it can be written to only when the TE and RE bits in SCR are 0.

**Table 32.8 Relationships between N Setting in BRR and Bit Rate B**

Mode	ABCS Bit in SEMR	BRR Setting	Error
Asynchronous, multi-processor transfer	0	$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	$Error (\%) = \left\{ \frac{PCLK \times 10^6}{B \times 64 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
	1	$N = \frac{PCLK \times 10^6}{32 \times 2^{2n-1} \times B} - 1$	$Error (\%) = \left\{ \frac{PCLK \times 10^6}{B \times 32 \times 2^{2n-1} \times (N+1)} - 1 \right\} \times 100$
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 I <sup>2</sup> C*1		$N = \frac{PCLK \times 10^6}{64 \times 2^{2n-1} \times B} - 1$	

B: Bit rate (bps)

N: BRR setting for baud rate generator ( $0 \leq N \leq 255$ )

PCLK: Operating frequency (MHz)

n and S: Determined by the SMR setting listed in the following table.

Note 1. Adjust the bit rate so that the widths at high and low level of the SCL output in simple I<sup>2</sup>C mode satisfy the I<sup>2</sup>C standard.



**Table 32.9 Calculating Widths at High and Low Level for SCL**

Mode	SCL	Formula (Result in Seconds)
I <sup>2</sup> C	Width at high level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 7 \times \frac{1}{\text{PCLK} \times 10^6}$
	Width at low level (minimum value)	$(N+1) \times 4 \times 2^{2n-1} \times 8 \times \frac{1}{\text{PCLK} \times 10^6}$

**Table 32.10 Clock Source Settings**

SMR Setting		
CKS[1:0] Bits	Clock Source	n
0 0	PCLK clock	0
0 1	PCLK/4 clock	1
1 0	PCLK/16 clock	2
1 1	PCLK/64 clock	3

**Table 32.11 Base Clock Settings in Smart Card Interface Mode**

SCMR Setting	SMR Setting	Base Clock Cycles for One-bit Period	
BCP2 Bit	BCP[1:0] Bits		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 32.12, Table 32.13 lists sample N settings in BRR in normal asynchronous mode. Table 32.13 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 32.15. Examples of BRR (N) settings in smart card interface mode are listed in Table 32.15. Examples of BRR (N) settings in simple I<sup>2</sup>C mode are listed in Table 32.19. In smart card interface mode, the number of base clock cycles S in a 1-bit data transfer time can be selected. For details, see section 32.6.4, Receive Data Sampling Timing and Reception Margin. Table 32.14 and Table 32.16 list the maximum bit rates with external clock input.

When the asynchronous mode base clock select bit (ABCS) in the serial extended mode register (SEMR) is set to 1 in asynchronous mode, the bit rate is two times that of listed in Table 32.12.

**Table 32.12 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (1)**

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	8			9.8304			10			12		
	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
150	2	103	0.16	2	127	0.00	2	129	0.16	2	155	0.16
300	1	207	0.16	1	255	0.00	2	64	0.16	2	77	0.16
600	1	103	0.16	1	127	0.00	1	129	0.16	1	155	0.16
1200	0	207	0.16	0	255	0.00	1	64	0.16	1	77	0.16
2400	0	103	0.16	0	127	0.00	0	129	0.16	0	155	0.16
4800	0	51	0.16	0	63	0.00	0	64	0.16	0	77	0.16
9600	0	25	0.16	0	31	0.00	0	32	-1.36	0	38	0.16
19200	0	12	0.16	0	15	0.00	0	15	1.73	0	19	-2.34
31250	0	7	0.00	0	9	-1.70	0	9	0.00	0	11	0.00
38400	—	—	—	0	7	0.00	0	7	1.73	0	9	-2.34

**Table 32.12 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (2)**

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	12.288			14			16			17.2032		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	217	0.08	2	248	-0.17	3	70	0.03	3	75	0.48
150	2	159	0.00	2	181	0.16	2	207	0.16	2	223	0.00
300	2	79	0.00	2	90	0.16	2	103	0.16	2	111	0.00
600	1	159	0.00	1	181	0.16	1	207	0.16	1	223	0.00
1200	1	79	0.00	1	90	0.16	1	103	0.16	1	111	0.00
2400	0	159	0.00	0	181	0.16	0	207	0.16	0	223	0.00
4800	0	79	0.00	0	90	0.16	0	103	0.16	0	111	0.00
9600	0	39	0.00	0	45	-0.93	0	51	0.16	0	55	0.00
19200	0	19	0.00	0	22	-0.93	0	25	0.16	0	27	0.00
31250	0	11	2.40	0	13	0.00	0	15	0.00	0	16	1.20
38400	0	9	0.00	—	—	—	0	12	0.16	0	13	0.00

Note: • This is an example when the ABCS bit in SEMR is 0.  
When the ABCS bit is set to 1, the bit rate is two times.

**Table 32.12 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (3)**

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	18			19.6608			20			25		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	79	-0.12	3	86	0.31	3	88	-0.25	3	110	-0.02
150	2	233	0.16	2	255	0.00	3	64	0.16	3	80	0.47
300	2	116	0.16	2	127	0.00	2	129	0.16	2	162	-0.15
600	1	233	0.16	1	255	0.00	2	64	0.16	2	80	0.47
1200	1	116	0.16	1	127	0.00	1	129	0.16	1	162	-0.15
2400	0	233	0.16	0	255	0.00	1	64	0.16	1	80	0.47
4800	0	116	0.16	0	127	0.00	0	129	0.16	0	162	-0.15
9600	0	58	-0.69	0	63	0.00	0	64	0.16	0	80	0.47
19200	0	28	1.02	0	31	0.00	0	32	-1.36	0	40	-0.76
31250	0	17	0.00	0	19	-1.70	0	19	0.00	0	24	0.00
38400	0	14	-2.34	0	15	0.00	0	15	1.73	0	19	1.73

**Table 32.12 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (4)**

Bit Rate (bps)	Operating Frequency PCLK (MHz)								
	30			33			50		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	3	132	0.13	3	145	0.33	3	221	-0.02
150	3	97	-0.35	3	106	0.39	3	162	-0.15
300	2	194	0.16	2	214	-0.07	3	80	0.47
600	2	97	-0.35	2	106	0.39	2	162	-0.15
1200	1	194	0.16	1	214	-0.07	2	80	0.47
2400	1	97	-0.35	1	106	0.39	1	162	-0.15
4800	0	194	0.16	0	214	-0.07	1	80	0.47
9600	0	97	-0.35	0	106	0.39	1	40	-0.76
19200	0	48	-0.35	0	53	-0.54	0	80	0.47
31250	0	29	0	0	32	0	0	49	0.00
38400	0	23	1.73	0	26	-0.54	0	40	-0.76

Note: • This is an example when the ABCS bit in SEMR is 0.  
 When the ABCS bit is set to 1, the bit rate is two times.

**Table 32.13 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode)**

PCLK (MHz)	Maximum Bit Rate(bps)	n	N
8	250000	0	0
9.8304	307200	0	0
10	312500	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
16	500000	0	0
17.2032	537600	0	0
18	562500	0	0
19.6608	614400	0	0
20	625000	0	0
25	781250	0	0
30	937500	0	0
33	1031250	0	0
50	1562500	0	0

Note: • When the ABCS bit in SEMR is set to 1, the bit rate is two times.

**Table 32.14 Maximum Bit Rate with External Clock Input (Asynchronous Mode)**

PCLK (MHz)	External Input Clock(MHz)	Maximum Bit Rate(bps)	
		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
33	8.2500	515625	1031250
50	12.500	781250	1562500

**Table 32.15 BRR Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode)**

Bit Rate (bps)	Operating Frequency PCLK (MHz)															
	8		10		16		20		25		30		33		50	
	n	N	n	N	n	N	n	N	n	N	n	N	n	N	n	N
110																
250	3	124	—	—	3	249										
500	2	249	—	—	3	124	—	—			3	233				
1 k	2	124	—	—	2	249	—	—	3	97	3	116	3	128	3	194
2.5 k	1	199	1	249	2	99	2	124	2	155	2	187	2	205	3	77
5 k	1	99	1	124	1	199	1	249	2	77	2	93	2	102	2	155
10 k	0	199	0	249	1	99	1	124	1	155	1	187	1	205	2	77
25 k	0	79	0	99	0	159	0	199	0	249	1	74	1	82	1	124
50 k	0	39	0	49	0	79	0	99	0	124	0	149	0	164	1	61
100 k	0	19	0	24	0	39	0	49	0	62	0	74	0	82	0	124
250 k	0	7	0	9	0	15	0	19	0	24	0	29	0	32	0	49
500 k	0	3	0	4	0	7	0	9	—	—	0	14	—	—	0	24
1 M	0	0	—	—	0	3	0	4	—	—	—	—	—	—	—	—
2 M	0	0*1	—	—	0	1	—	—	—	—	—	—	—	—	—	—
2.5 M			0	0*1			0	1	—	—	0	2	—	—	0	4
4 M					0	0*1	—	—	—	—	—	—	—	—	—	—
5 M							0	0*1	—	—	—	—	—	—	—	—
6.25 M									0	0*1	—	—	—	—	0	1
7.5 M											0	0*1	—	—	—	—
8.25 M													0	0*1	—	—

Space: Setting prohibited.

—: Can be set, but there will be error.

Note 1. Continuous transmission or reception is not possible. When the synchronizing clock is being output in transmission, a single bit-period elapses after completion of transmission of the final bit and before transmission of the next frame starts. When the synchronizing clock is being input in transmission, allow at least one bit-period of idle time at the source for output of the synchronizing clock after transmission of the final bit and before the start of transmission of the next frame.

**Table 32.16 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode)**

PCLK (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bps)
8	1.3333	1333333.3
10	1.6667	1666666.7
12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3
25	4.1667	4166666.7
30	5.0000	5000000.0
33	5.5000	5500000.0
50	8.3333	8333333.3

**Table 32.17 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
	10.7136	0	1	25
	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
	33.00	0	4	7.59
	50.00	0	6	0.01

**Table 32.18 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 372)**

PCLK (MHz)	Maximum Bit Rate (bps)	n	N
10.00	13441	0	0
10.7136	14400	0	0
13.00	17473	0	0
16.00	21505	0	0
18.00	24194	0	0
20.00	26882	0	0
25.00	33602	0	0
30.00	40323	0	0
33.00	44355	0	0
50.00	67204	0	0

Table 32.19 BRR Settings for Various Bit Rates (Simple I<sup>2</sup>C Mode) (1)

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	8			10			16			20		
	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
25 k	0	9	0.0	0	12	-3.8	1	4	0.0	1	6	-10.7
50 k	0	4	0.0	0	6	-10.7	1	2	-16.7	1	3	-21.9
100 k	0	2	-16.7	0	3	-21.9	0	4	0.0	0	6	-10.7
250 k	0	0	0.0	0	1	-37.5	0	1	0.0	0	2	-16.7
350 k										0	1	-10.7

Table 32.19 BRR Settings for Various Bit Rates (Simple I<sup>2</sup>C Mode) (2)

Bit Rate (bps)	Operating Frequency PCLK (MHz)											
	25			30			33			50		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
10 k	1	19	-2.3	1	23	-2.3	1	25	-0.8	2	9	-2.3
25 k	1	7	-2.3	1	9	-6.3	1	10	-6.3	2	3	-2.3
50 k	1	3	-2.3	1	4	-6.3	1	5	-14.1	2	1	-2.3
100 k	1	1	-2.3	1	2	-21.9	1	2	-14.1	1	3	-2.3
250 k	0	3	-21.9	0	3	-6.3	0	4	-17.5	0	6	-10.7
350 k	0	2	-25.6	0	2	-10.7	0	2	-1.8	0	4	-10.7

Table 32.20 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple I<sup>2</sup>C Mode) (1)

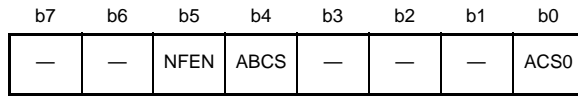
Min. Widths at High/Low Level for SCL (μs)	Operating Frequency PCLK (MHz)											
	8			10			16			20		
	n	N	High/Low Width	n	N	High/Low Width	n	N	High/Low Width	n	N	High/Low Width
10 k	0	24	43.75/50.00	0	31	44.80/51.20	1	12	45.5/52.00	1	15	44.80/51.20
25 k	0	9	17.50/20.00	0	12	18.2/20.80	1	4	17.50/20.00	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

Table 32.20 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple I<sup>2</sup>C Mode) (2)

Min. Widths at High/Low Level for SCL (μs)	Operating Frequency PCLK (MHz)											
	25			30			33			50		
	n	N	High/Low Width	n	N	High/Low Width	n	N	High/Low Width	n	N	High/Low Width
10 k	1	19	44.80/51.20	1	23	44.80/51.20	1	25	44.12/50.42	2	9	44.80/51.20
25 k	1	7	17.92/20.48	1	9	18.66/21.33	1	10	18.66/21.33	2	3	17.92/20.48
50 k	1	3	8.96/10.24	1	4	9.33/10.66	1	5	10.18/11.63	2	1	8.96/10.24
100 k	1	1	4.48/5.12	1	2	5.60/6.40	1	2	5.09/5.81	1	3	4.48/5.12
250 k	0	3	2.24/2.56	0	3	1.86/2.13	0	4	2.12/2.42	0	6	1.96/2.24
350 k	0	2	1.68/1.92	0	2	1.40/1.60	0	2	1.27/1.45	0	4	1.40/1.60

### 32.2.10 Serial Extended Mode Register (SEMR)

Address(es): SCI0.SEMR 0008 A007h, SCI1.SEMR 0008 A027h, SCI2.SEMR 0008 A047h, SCI3.SEMR 0008 A067h, SCI4.SEMR 0008 A087h, SCI5.SEMR 0008 A0A7h, SCI6.SEMR 0008 A0C7h, SCI7.SEMR 0008 A0E7h, SCI8.SEMR 0008 A107h, SCI9.SEMR 0008 A127h, SCI10.SEMR 0008 A147h, SCI11.SEMR 0008 A167h, SCI12.SEMR 0008 B307h



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: TMR clock input (valid only for SCI5, SCI6, and SCI12) The following table lists the correspondence between SCI channels and compare match outputs.	R/W*1													
			<table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left;">SCI</th> <th style="text-align: left;">TMR</th> <th style="text-align: left;">Compare Match Output</th> </tr> </thead> <tbody> <tr> <td>SCI5</td> <td>Unit 0</td> <td>TMO0, TMO1</td> </tr> <tr> <td>SCI6</td> <td>Unit 1</td> <td>TMO2, TMO3</td> </tr> <tr> <td>SCI12</td> <td>Unit 0</td> <td>TMO0, TMO1</td> </tr> </tbody> </table>	SCI	TMR	Compare Match Output	SCI5	Unit 0	TMO0, TMO1	SCI6	Unit 1	TMO2, TMO3	SCI12	Unit 0	TMO0, TMO1		
SCI	TMR	Compare Match Output															
SCI5	Unit 0	TMO0, TMO1															
SCI6	Unit 1	TMO2, TMO3															
SCI12	Unit 0	TMO0, TMO1															
b3 to b1	—	Reserved	These bits are 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 <sup>2</sup> C 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													
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W													

Note 1. Writable only when TE in SCR = 0 and RE in SCR = 0 (both serial transmission and reception are disabled).

SEMR selects the clock source for 1-bit period in asynchronous mode.

For SCI5, SCI6, and SCI12, the TMO<sub>n</sub> output (n = 0 to 3) of TMR units 0 and 1 can be set as the serial transfer base clock.

Figure 32.4 shows a setting example when the TMO<sub>n</sub> output of TMR<sub>n</sub> (n = 0 to 3) is selected.

#### ACS0 Bit (Asynchronous Mode Clock Source Select)

Selects the clock source in the asynchronous mode.

The ACS0 bit is valid in asynchronous mode (CM bit in SMR = 0) and when an external clock input is selected (CKE[1:0] bits in SCR = 10b or 11b). An external clock input or internal TMR clock input can be selected.

Clear the ACS0 bit to 0 in other than asynchronous mode.

These bits for the other SCI channels than SCI5, SCI6, and SCI12 are reserved. The write values to these bits for other than SCI5, SCI6, and SCI12 should always be 0.



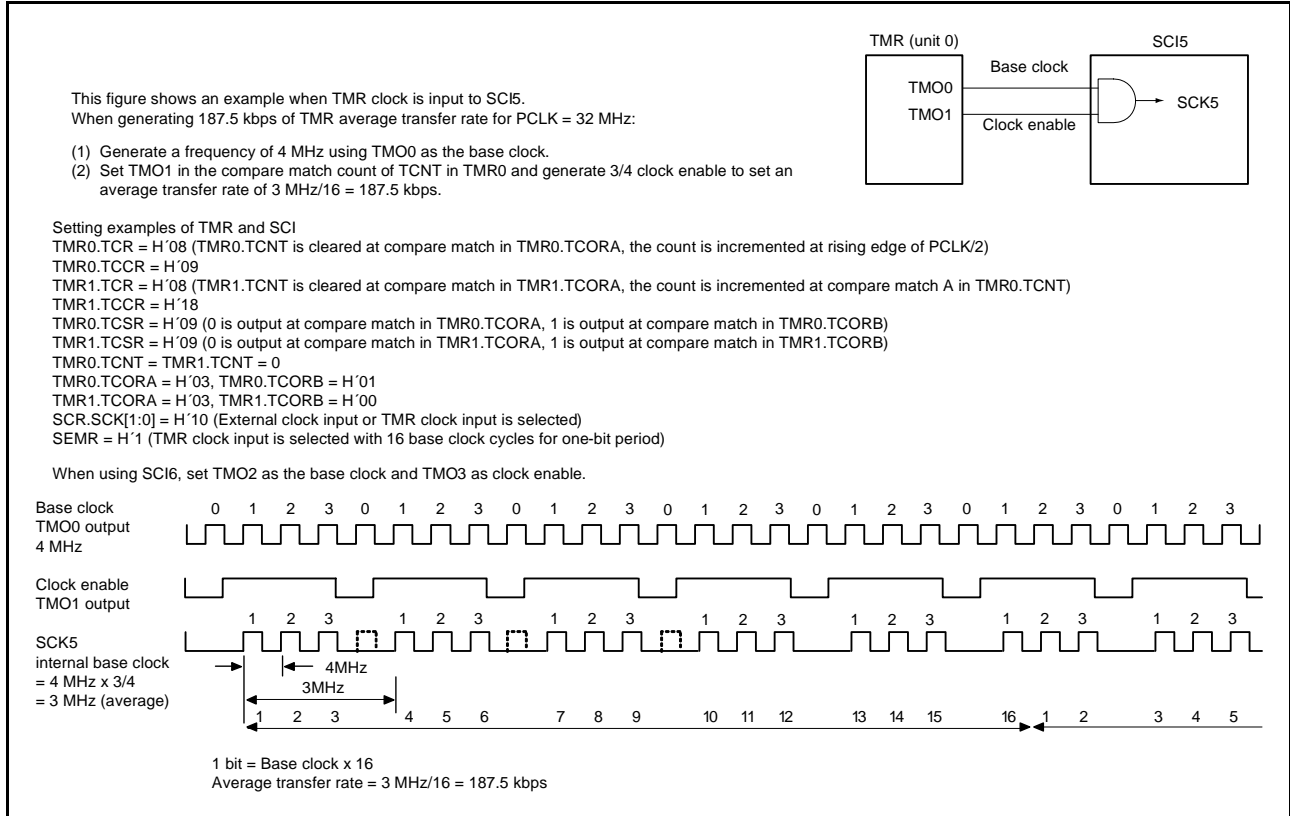
**NFEN Bit (Digital Noise Filter Function Enable)**

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<sup>2</sup>C mode.

In any mode other than above, set the NFEN bit to 0 to disable the digital noise filter function.

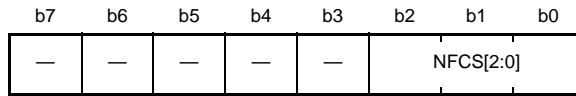
When the function is disabled, input signals are transferred as is, as internal signals.



**Figure 32.4 Example of Average Transfer Rate Setting when TMR Clock is Input**

### 32.2.11 Noise Filter Setting Register (SNFR)

Address(es): SCI0.SNFR 0008 A008h, SCI1.SNFR 0008 A028h, SCI2.SNFR 0008 A048h, SCI3.SNFR 0008 A068h, SCI4.SNFR 0008 A088h, SCI5.SNFR 0008 A0A8h, SCI6.SNFR 0008 A0C8h, SCI7.SNFR 0008 A0E8h, SCI8.SNFR 0008 A108h, SCI9.SNFR 0008 A128h, SCI10.SNFR 0008 A148h, SCI11.SNFR 0008 A168h, SCI12.SNFR 0008 B308h



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 I <sup>2</sup> C mode, the standard settings for the clock source selected for the on-chip baud rate generator 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.  Other values: Do not make settings other than those listed above.	R/W*1
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

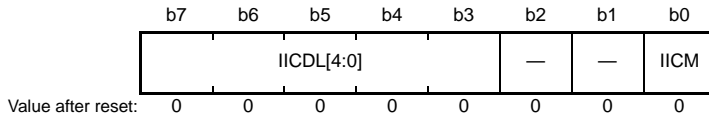
Note 1. Writing to these bits is only possible when the RE and TE bits in the SCR are 0 (disabling reception and transmission).

#### 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 I<sup>2</sup>C mode, set the bits to a value in the range from 001b to 100b.

### 32.2.12 I<sup>2</sup>C Mode Register 1 (SIMR1)

Address(es): SCI0.SIMR1 0008 A009h, SCI1.SIMR1 0008 A029h, SCI2.SIMR1 0008 A049h, SCI3.SIMR1 0008 A069h, SCI4.SIMR1 0008 A089h, SCI5.SIMR1 0008 A0A9h, SCI6.SIMR1 0008 A0C9h, SCI7.SIMR1 0008 A0E9h, SCI8.SIMR1 0008 A109h, SCI9.SIMR1 0008 A129h, SCI10.SIMR1 0008 A149h, SCI11.SIMR1 0008 A169h, SCI12.SIMR1 0008 B309h



Bit	Symbol	Bit Name	Description	R/W
b0	IICM	Simple I <sup>2</sup> C Mode Select	SMIF IICM 0 0: Serial interface mode (in asynchronous, synchronous, or simple SPI mode) 0 1: Simple I <sup>2</sup> C 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 Delay Output 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 are 0 (disabling reception and transmission).

SIMR1 is used to select simple I<sup>2</sup>C mode and the number of delay stages for the SSDA output.

#### IICM Bit (Simple I<sup>2</sup>C Mode Select)

In conjunction with the SMIF bit in SCMR, 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 SMR.CKS[1:0] is supplied as the clock signal from the on-chip baud rate generator. Set these bits to 00000b unless operation is in simple I<sup>2</sup>C mode. In simple I<sup>2</sup>C mode, set the bits to a value in the range from 00001b to 11111b.

### 32.2.13 I<sup>2</sup>C Mode Register 2 (SIMR2)

Address(es): SCI0.SIMR2 0008 A00Ah, SCI1.SIMR2 0008 A02Ah, SCI2.SIMR2 0008 A04Ah, SCI3.SIMR2 0008 A06Ah, SCI4.SIMR2 0008 A08Ah, SCI5.SIMR2 0008 A0AAh, SCI6.SIMR2 0008 A0CAh, SCI7.SIMR2 0008 A0EAh, SCI8.SIMR2 0008 A10Ah, SCI9.SIMR2 0008 A12Ah, SCI10.SIMR2 0008 A14Ah, SCI11.SIMR2 0008 A16Ah, SCI12.SIMR2 0008 B30Ah

b7	b6	b5	b4	b3	b2	b1	b0
—	—	IICACK T	—	—	—	IICCSC	IICINT M

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	IICINTM	I <sup>2</sup> C 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 are 0 (disabling serial reception and transmission).

SIMR2 is used to select how reception and transmission are controlled in simple I<sup>2</sup>C mode.

#### IICINTM Bit (I<sup>2</sup>C Interrupt Mode Select)

This bit selects the sources of interrupt requests in simple I<sup>2</sup>C 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 bit 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.

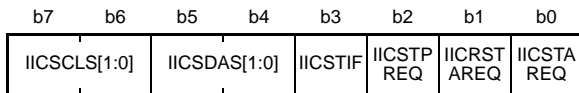
Set the IICCSC bit to 1 except during debugging.

#### IICACKT Bit (ACK Transmission Data)

Transmitted data contains ACK bits. Set this bit to 1 when ACK and NACK bits are received.

### 32.2.14 I<sup>2</sup>C Mode Register 3 (SIMR3)

Address(es): SCI0.SIMR3 0008 A00Bh, SCI1.SIMR3 0008 A02Bh, SCI2.SIMR3 0008 A04Bh, SCI3.SIMR3 0008 A06Bh, SCI4.SIMR3 0008 A08Bh, SCI5.SIMR3 0008 A0ABh, SCI6.SIMR3 0008 A0CBh, SCI7.SIMR3 0008 A0EBh, SCI8.SIMR3 0008 A10Bh, SCI9.SIMR3 0008 A12Bh, SCI10.SIMR3 0008 A14Bh, SCI11.SIMR3 0008 A16Bh, SCI12.SIMR3 0008 B30Bh



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	R/W
b1	IICRSTAREQ	Restart Condition Generation	0: A restart condition is not generated. 1: A restart condition is generated.*2,*3,*4	R/W
b2	IICSTPREQ	Stop Condition Generation	0: A stop condition is not generated. 1: A stop condition is generated.*2,*3,*4	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: All request generation has been completed.	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 SSDA pin. 1 1: Place the SSDA 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 SSCL pin. 1 1: Place the SSCL pin in the high-impedance state.	R/W

- Note 1. Only generate a start condition after checking the bus state and confirming that it is free.
- Note 2. Generate a restart or stop condition after checking the bus state and confirming that it is busy.
- 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.

SIMR3 is used to control the simple I<sup>2</sup>C 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 execution for the generation of a condition, this bit indicates the state of generation being completed. When using the IICSTAREQ, IICRSTAREQ, or IICSTPREQ bit to cause generation of a condition, do so after clearing the IICSTIF flag to 0.

If the TEIE bit in the SCR is enabling interrupt requests, an STI is output on completion of generation of the start, restart, or stop condition when the IICSTIF flag is 1.

[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 zero 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 IICM bit in SIMR1 (when operation is not in simple I<sup>2</sup>C mode)
- Writing 0 to the TE bit in SCR

**IICSDAS[1:0] Bits (SSDA Output Select)**

These bits control output from the SSDAn pin.

Set the IICSDAS and IICSCLS bits to the same value during normal operations.

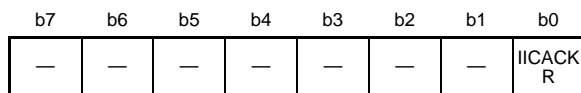
**IICSCLS[1:0] Bits (SSCL Output Select)**

These bits control output from the SSCLn pin.

Set the IICSCLS and IICSDAS bits to the same value during normal operations.

### 32.2.15 I<sup>2</sup>C Status Register (SISR)

Address(es): SCI0.SISR 0008 A00Ch, SCI1.SISR 0008 A02Ch, SCI2.SISR 0008 A04Ch, SCI3.SISR 0008 A06Ch,  
SCI4.SISR 0008 A08Ch, SCI5.SISR 0008 A0ACh, SCI6.SISR 0008 A0CCh, SCI7.SISR 0008 A0ECh,  
SCI8.SISR 0008 A10Ch, SCI9.SISR 0008 A12Ch, SCI10.SISR 0008 A14Ch, SCI11.SISR 0008 A16Ch,  
SCI12.SISR 0008 B30Ch



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 I<sup>2</sup>C mode.

#### IICACKR Flag (ACK Reception Data Flag)

Received ACK and NACK bits can be read from this bit.

The IICACK flag is updated at the rising of SSCLn clock for the ACK/NACK receiving bit.

### 32.2.16 SPI Mode Register (SPMR)

Address(es): SCI0.SPMR 0008 A00Dh, SCI1.SPMR 0008 A02Dh, SCI2.SPMR 0008 A04Dh, SCI3.SPMR 0008 A06Dh, SCI4.SPMR 0008 A08Dh, SCI5.SPMR 0008 A0ADh, SCI6.SPMR 0008 A0CDh, SCI7.SPMR 0008 A0EDh, SCI8.SPMR 0008 A10Dh, SCI9.SPMR 0008 A12Dh, SCI10.SPMR 0008 A14Dh, SCI11.SPMR 0008 A16Dh, 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	SS Pin Function Enable	0: SS pin function is disabled. 1: SS pin function is enabled.	R/W*1
b1	CTSE	CTS Enable	0: CTS pin function is disabled (RTS output function is enabled). 1: CTS pin function is enabled	R/W*1
b2	MSS	Master Slave Select	0: Transmission is through the TXDn pin and reception is through the RXDn pin (master mode). 1: Reception is through the TXDn pin and transmission is through the RXDn 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 are 0 (disabling reception and transmission).

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 (SS 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 SS# 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 CTS bits to “enabled” (even if this setting is made, the functions will be disabled).

#### 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<sup>2</sup>C mode. Do not set both the CTS and SSE bits to “enabled” (even if this setting is made, the functions will be disabled).



**MSS Bit (Master Slave Select)**

This bit selects between master and slave operation in simple SPI mode. The functions of the TXDn and RXDn pins are reversed when the MSS bit is set to 1, so that data are received through the TXDn pin and transmitted through the RXDn 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 SS# 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. See Figure 32.52 for details.

Clear the bit to 0 in other than simple SPI mode and clock synchronous mode.

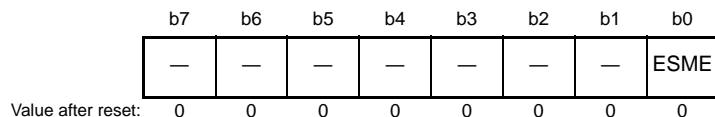
**CKPH Bit (Clock Phase Select)**

This bit selects the phase of the clock signal output through the SCKn pin. See Figure 32.52 for details.

Clear the bit to 0 in other than simple SPI mode and clock synchronous mode.

**32.2.17 Extended Serial Module Enable Register (ESMER)**

Address(es): SCI12.ESMER 0008 B320h



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 enters the following states:

- the extended serial mode control section is initialized

**Table 32.21 Settings of the ESME Bits and Guaranteed Operation by Timer Operation Mode**

ESME bit	Timer Mode	Break Field Low Width Judgment Mode	Break Field Low Width Output Mode
0	○*1	×	×
1	○	○	○

○: Guarantee of operation is necessary. ×: Guarantee of operation is not necessary.

Note 1. Operation is only possible with PCLK selected.

### 32.2.18 Control Register 0 (CR0)

Address(es): SCI12.CR0 0008 B321h

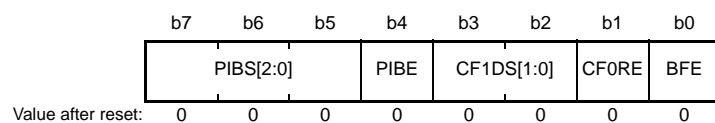
b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	BRME	RXDSF	SFSF	—

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	RXDX Input Status Flag	0: RXDX input is enabled. 1: RXDX 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

## 32.2.19 Control Register 1 (CR1)

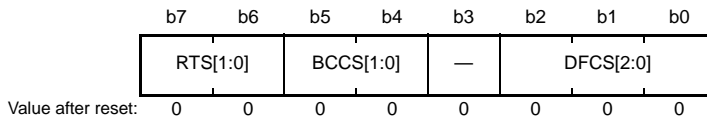
Address(es): SCl12.CR1 0008 B322h



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 PCF1DR. 0 1: Selects comparison with the value in SCF1DR. 1 0: Selects comparison with the values in PCF1DR and SCF1DR. 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

## 32.2.20 Control Register 2 (CR2)

Address(es): SCI12.CR2 0008 B320h



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 is enabled (SCI base clock). 0 1 0: Filter is enabled (SCI base clock/8). 0 1 1: Filter is enabled (SCI base clock/16). 1 0 0: Filter is enabled (SCI base clock/32). 1 0 1: Filter is enabled (SCI base clock/64). 1 1 0: Filter is enabled (SCI base clock/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	b5 b4 0 0: SCI base clock 0 1: SCI base clock frequency divided by 2 1 0: SCI base clock frequency divided by 4 1 1: Setting prohibited	R/W
b7, b6	RTS[1:0]	RXDX12 Reception Sampling Timing Select	<ul style="list-style-type: none"> <li>• When SCI12.SEMR.ABCS = 0               <ul style="list-style-type: none"> <li>b7 b6</li> <li>0 0: Rising edge of the 8th cycle of SCI base clock</li> <li>0 1: Rising edge of the 10th cycle of SCI base clock</li> <li>1 0: Rising edge of the 12th cycle of SCI base clock</li> <li>1 1: Rising edge of the 14th cycle of SCI base clock</li> </ul> </li> <li>• When SCI12.SEMR.ABCS = 1               <ul style="list-style-type: none"> <li>b7 b6</li> <li>0 0: Rising edge of the 4th cycle of SCI base clock</li> <li>0 1: Rising edge of the 5th cycle of SCI base clock</li> <li>1 0: Rising edge of the 6th cycle of SCI base clock</li> <li>1 1: Rising edge of the 7th cycle of SCI base clock</li> </ul> </li> </ul>	R/W

Note 1. The period of the SCI base clock is 1/16 of a single data period when the SCI12.SEMR.ABCS is 0, and 1/8 of a single data period when the SCI12.SEMR.ABCS is 1. To use the SCI base clock, set the SCI12.SCR.TE bit to 1.

### 32.2.21 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 Bits (Start Frame Detection Start)

Detection of a Start Frame begins when this bit is set to 1. The bit is read as 0.

### 32.2.22 Port Control Register (PCR)

Address(es): SCI12.PCR 0008 B325h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	SHARPS	—	—	RXDXP S	TXDXP S

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	TXDXPS	TXDX Signal Polarity Select	0: The polarity of TXDX signal is not inverted for output. 1: The polarity of TXDX signal is inverted for output.	R/W
b1	RXDXP S	RXDX Signal Polarity Select	0: The polarity of RXDX signal is not inverted for input. 1: The polarity of RXDX 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	TXDX/RXDX Pin Multiplexing Select	0: The TXDX and RXDX pins are independent. 1: The TXDX and RXDX 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 (TXDX/RXDX Pin Multiplexing Selection)

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.

### 32.2.23 Interrupt Control Register (ICR)

Address(es): SCI12.ICR 0008 B326h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	AEDIE	BCDIE	PIBDIE	CF1MIE	CF0MIE	BFDIE
0	0	0	0	0	0	0	0

Value after reset:

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

## 32.2.24 Status Register (STR)

Address(es): SCI12.STR 0008 B327h

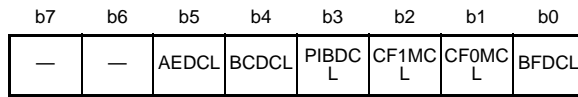
b7	b6	b5	b4	b3	b2	b1	b0
—	—	AEDF	BCDF	PIBDF	CF1MF	CF0MF	BFDF
0	0	0	0	0	0	0	0

Value after reset:

Bit	Symbol	Bit Name	Description	R/W
b0	BFDF	Break Field Low Width Detection Flag	[Setting conditions] <ul style="list-style-type: none"> <li>Detection of the low width for a Break Field</li> <li>Completion of the output of the low width for a Break Field</li> <li>Underflow of the timer</li> </ul> [Clearing condition] <ul style="list-style-type: none"> <li>Writing 1 to the BFDCL bit in STCR</li> </ul>	R
b1	CF0MF	Control Field 0 Match Flag	[Setting condition] <ul style="list-style-type: none"> <li>A match between the value received in Control Field 0 and the set value.</li> </ul> [Clearing condition] <ul style="list-style-type: none"> <li>Writing 1 to the CF0MCL bit in STCR</li> </ul>	R
b2	CF1MF	Control Field 1 Match Flag	[Setting condition] <ul style="list-style-type: none"> <li>A match between the data received in Control Field 1 and the set values.</li> </ul> [Clearing condition] <ul style="list-style-type: none"> <li>Writing 1 to the CF1MCL bit in STCR</li> </ul>	R
b3	PIBDF	Priority Interrupt Bit Detection Flag	[Setting condition] <ul style="list-style-type: none"> <li>Detection of the priority interrupt bit</li> </ul> [Clearing condition] <ul style="list-style-type: none"> <li>Writing 1 to the PIBDCL bit in STCR</li> </ul>	R
b4	BCDF	Bus Collision Detected Flag	[Setting condition] <ul style="list-style-type: none"> <li>Detection of the bus collision</li> </ul> [Clearing condition] <ul style="list-style-type: none"> <li>Writing 1 to the BCDCL bit in STCR</li> </ul>	R
b5	AEDF	Valid Edge Detection Flag	[Setting condition] <ul style="list-style-type: none"> <li>Detection of a valid edge</li> </ul> [Clearing condition] <ul style="list-style-type: none"> <li>Writing 1 to the AEDCL bit in STCR</li> </ul>	R
b7, b6	—	Reserved	These bits are read as 0 and cannot be modified.	R

### 32.2.25 Status Clear Register (STCR)

Address(es): SCI12.STCR 0008 B328h

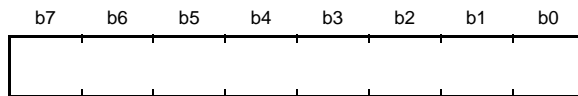


Value after reset: 0 0 0 0 0 0 0 0

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	PIBDC	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

### 32.2.26 Control Field 0 Data Register (CF0DR)

Address(es): SCI12.CF0DR 0008 B329h



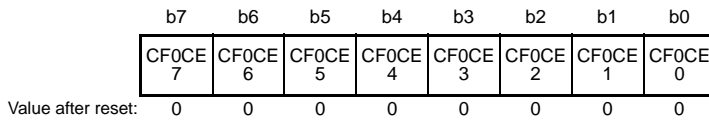
Value after reset: 0 0 0 0 0 0 0 0

CF0DR is an 8-bit readable and writable register that holds a value for comparison with Control Field 0.



### 32.2.27 Control Field 0 Compare Enable Register (CF0CR)

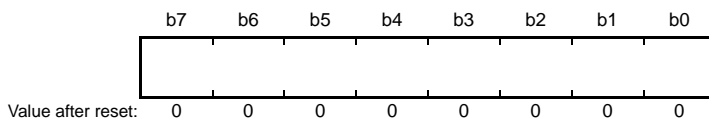
Address(es): SCI12.CF0CR 0008 B32Ah



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

### 32.2.28 Control Field 0 Receive Data Register (CF0RR)

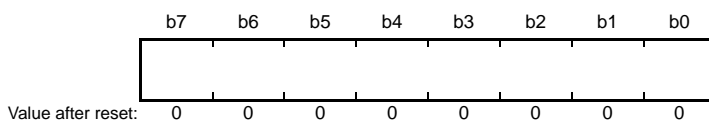
Address(es): SCI12.CF0RR 0008 B32Bh



CF0RR is a read-only register that holds the value received in Control Field 0. Writing to this register from the CPU or DTC is not possible.

### 32.2.29 Primary Control Field 1 Data Register (PCF1DR)

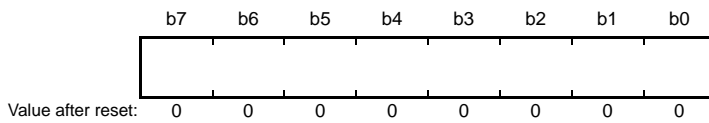
Address(es): SCI12.PCF1DR 0008 B32Ch



PCF1DR is an 8-bit readable and writable register that holds the 8-bit primary value for comparison with Control Field 1.

### 32.2.30 Secondary Control Field 1 Data Register (SCF1DR)

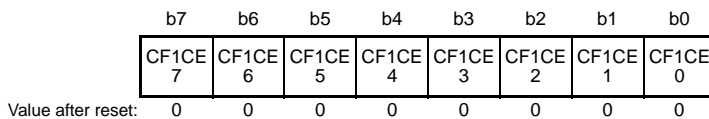
Address(es): SCI12.SCF1DR 0008 B32Dh



PCF1DR is an 8-bit readable and writable register that holds the 8-bit secondary value for comparison with Control Field 1.

### 32.2.31 Control Field 1 Compare Enable Register (CF1CR)

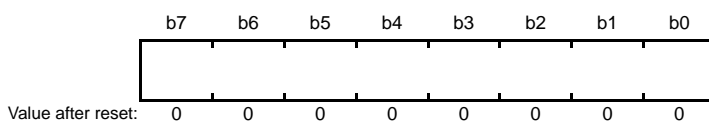
Address(es): SCI12.CF1CR 0008 B32Eh



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
b6	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

### 32.2.32 Control Field 1 Receive Data Register (CF1RR)

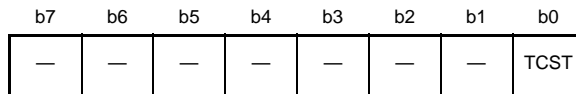
Address(es): SCI12.CF1RR 0008 B32Fh



CF1RR is a read-only register that holds the value received in Control Field 1. Writing to this register from the CPU or DTC is not possible.

### 32.2.33 Timer Control Register (TCR)

Address(es): SCI12.TCR 0008 B330h

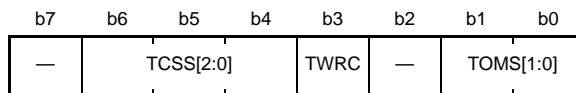


Value after reset: 0 0 0 0 0 0 0 0

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

### 32.2.34 Timer Mode Register (TMR)

Address(es): SCI12.TMR 0008 B31F331h



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: Writing is to the reload register and the counter. 1: Writing is only to the reload register	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	These bits are read as 0. The write value should be 0.	R/W

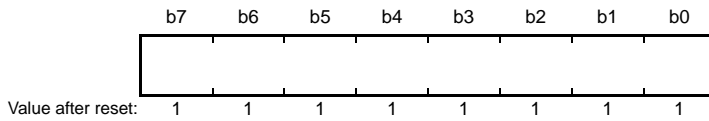
Note 1. Stop the counter (TCST = 0) before writing to the TOMS[1:0] or TCSS[2:0] bits.

#### TWRC Bit (Counter Write Control)

These bits determine whether a value written to TPRES or TCNT is only written to the reload register or is written to both the reload register and the counter.

### 32.2.35 Timer Prescaler Register (TPRE)

Address(es): SCI12.TPRE 0008 B332h

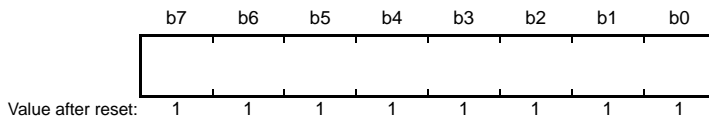


TPRE consists of an eight-bit reload register, a read buffer, and a counter, each of which has FFh as its initial value. The counter counts down in synchronization with the counter clock selected by the TMR.TCSS[2:0] bits, and is reloaded with the value in the reload register when it underflows. Underflows 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, so in writing, transfer is to the reload register and in reading, transfer is of the counter value from the read buffer.

It takes one PCLK cycle to load a value from the reload register to the counter.

### 32.2.36 Timer Count Register (TCNT)

Address(es): SCI12.TCNT0008 B333h



TCNT consists of an eight-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, so in writing, transfer is to the reload register and in reading, transfer is of the counter value from the read buffer.

It takes one PCLK cycle to load a value from the reload register to the counter.

### 32.3 Operation in Asynchronous Mode

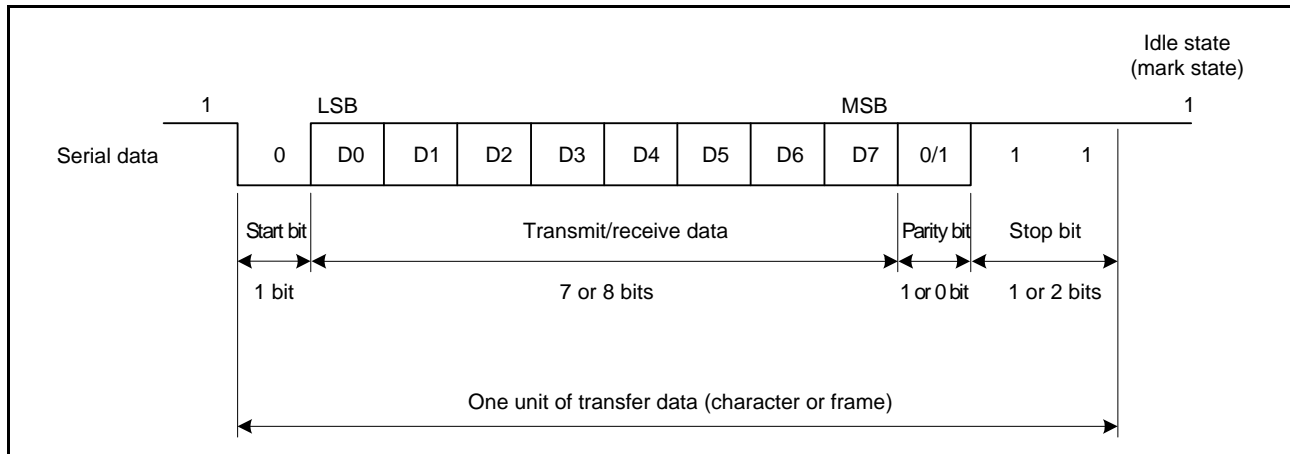
Figure 32.5 shows the general format for asynchronous serial communications.

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, and when it goes to the space state (low level), recognizes a start bit and starts serial communications.

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 32.5 Data Format in Asynchronous Serial Communications**  
(Example with 8-Bit Data, Parity, Two Stop Bits)

#### 32.3.1 Serial Data Transfer Format

Table 32.22 lists the serial data transfer formats that can be used in asynchronous mode.

Any of 12 transfer formats can be selected according to the SMR setting. For details of multi-processor function, see section 32.4, Multi-Processor Communications Function.

**Table 32.22 Serial Transfer Formats (Asynchronous Mode)**

SMR Setting				Serial Transfer Format and Frame Length												
CHR	PE	MP	STOP	1	2	3	4	5	6	7	8	9	10	11	12	
0	0	0	0	S	8-bit data								STOP			
0	0	0	1	S	8-bit data								STOP	STOP		
0	1	0	0	S	8-bit data								P	STOP		
0	1	0	1	S	8-bit data								P	STOP	STOP	
1	0	0	0	S	7-bit data							STOP				
1	0	0	1	S	7-bit data							STOP	STOP			
1	1	0	0	S	7-bit data							P	STOP			
1	1	0	1	S	7-bit data							P	STOP	STOP		
0	—	1	0	S	8-bit data								MPB	STOP		
0	—	1	1	S	8-bit data								MPB	STOP	STOP	
1	—	1	0	S	7-bit data							MPB	STOP			
1	—	1	1	S	7-bit data							MPB	STOP	STOP		

S: Start bit  
 STOP: Stop bit  
 P: Parity bit  
 MPB: Multi-processor bit

### 32.3.2 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 32.6. Thus the reception margin in asynchronous mode is determined by formula (1) below.

$$M = \left| \left( 0.5 - \frac{1}{2N} \right) - (L - 0.5) F - \frac{|D - 0.5|}{N} (1+F) \right| \times 100 [\%] \quad \dots \text{Formula (1)}$$

M: Reception margin

N: Ratio of bit rate to clock (N = 16 when ABCS in SEMR = 0, N = 8 when ABCS in SEMR = 1)

D: Duty cycle of clock (D = 0.5 to 1.0)

L: Frame length (L = 9 to 12)

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.

$$M = \{0.5 - 1/(2 \times 16)\} \times 100 (\%) = 46.875\%$$

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

Note 1. This is an example when the ABCS bit in SEMR 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

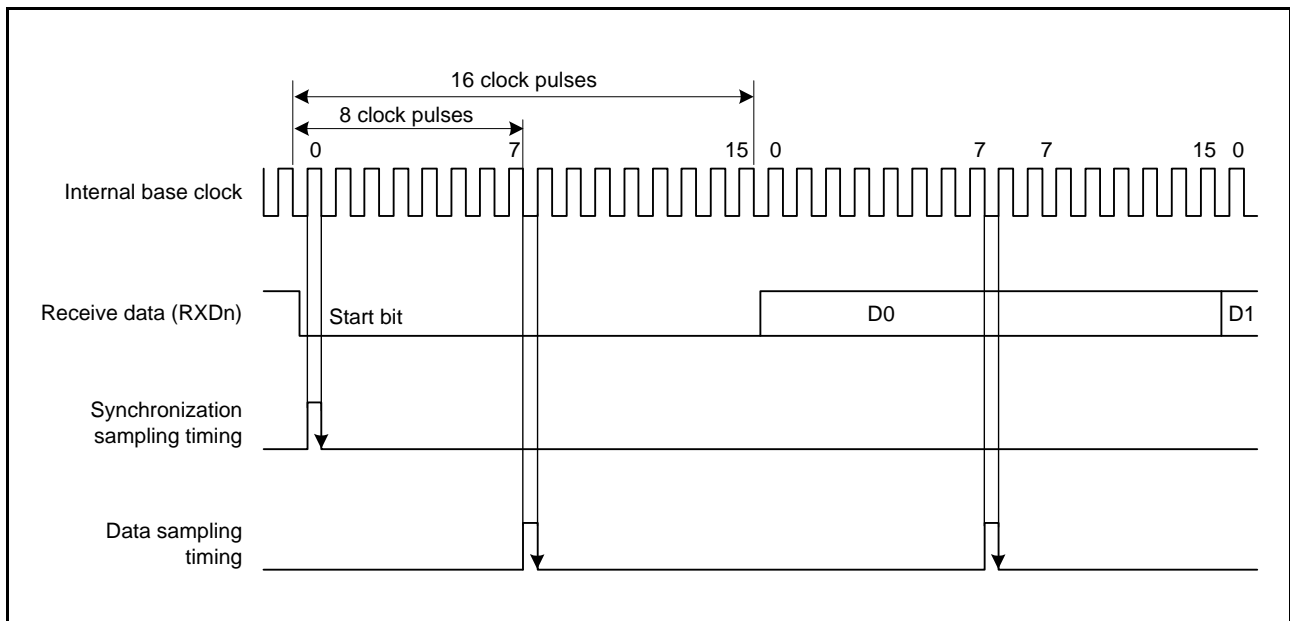


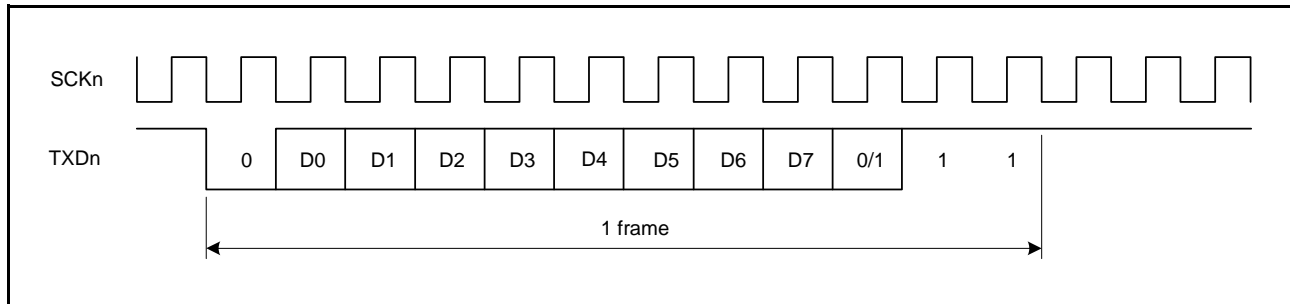
Figure 32.6 Receive Data Sampling Timing in Asynchronous Mode

### 32.3.3 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 CM bit in SMR and the CKE[1:0] bits in SCR.

When an external clock is input to the SCKn pin, the clock frequency should be 16 times the bit rate (when ABCS in SEMR = 0) and 8 times the bit rate (when ABCS in SEMR = 1). In addition, when an external clock is specified, the base clock of TMR0 and TMR1 can be selected by the ACS0 bit in SEMR of SCIn (n = 5, 6, 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 32.7.



**Figure 32.7** Phase Relationship between Output Clock and Transmit Data  
(Asynchronous Mode: SMR.CHR = 0, PE = 1, MP = 0, STOP = 1)

### 32.3.4 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 low level to the CTSn# 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]

Satisfaction of all conditions listed below

- The value of the RE bit in the SCR 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 are all 0

[Condition for high-level output]

- Any of the conditions for the low level not being satisfied



### 32.3.5 SCI Initialization (Asynchronous Mode)

Before transmitting and receiving data, start by writing the initial value “00h” to SCR and then continue through the procedure for SCI given in the sample flowchart (Figure 32.8). Whenever the operating mode or transfer format is changed, SCR 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 clearing the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in SSR nor RDR.

Moreover, note that switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 while the SCR.TIE bit is 1 leads to the generation of a TXI interrupt request.

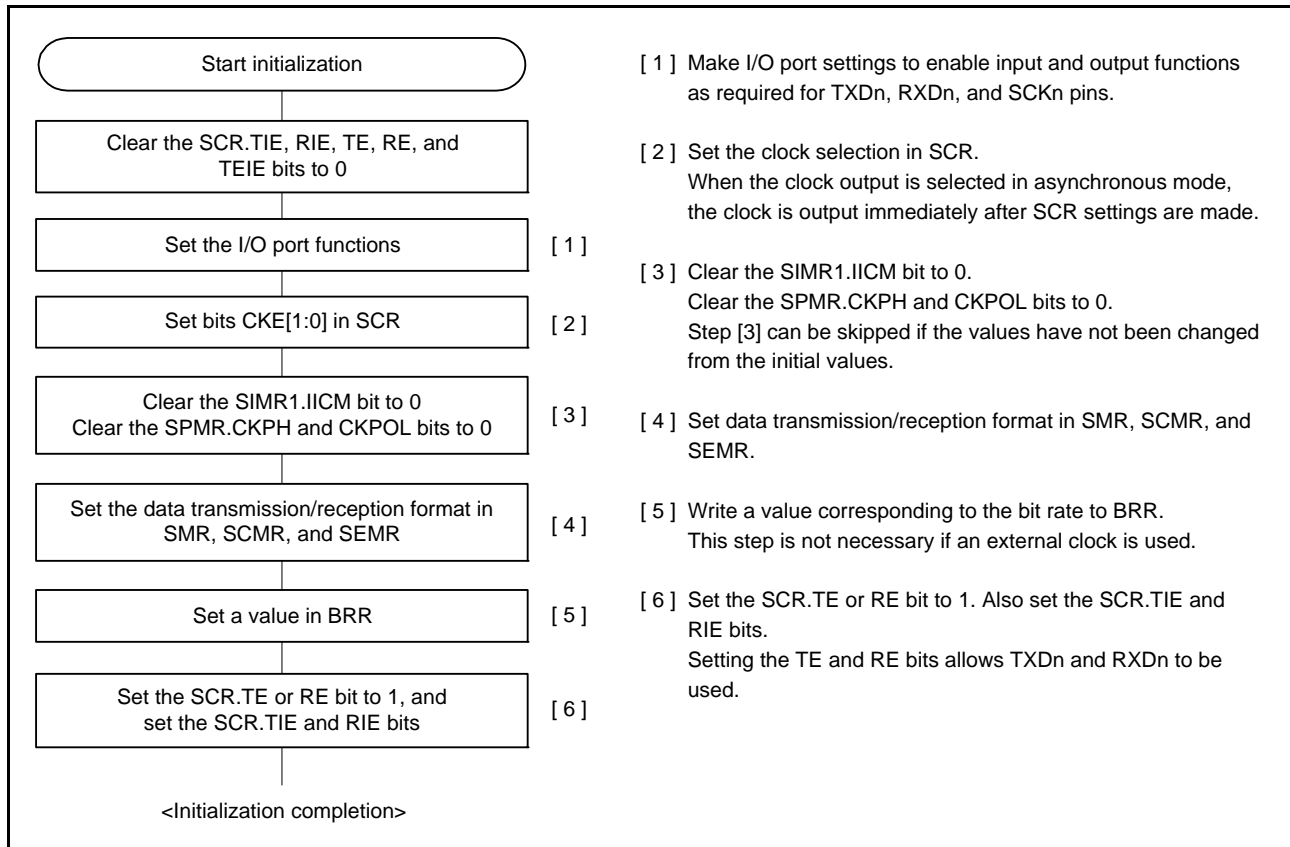


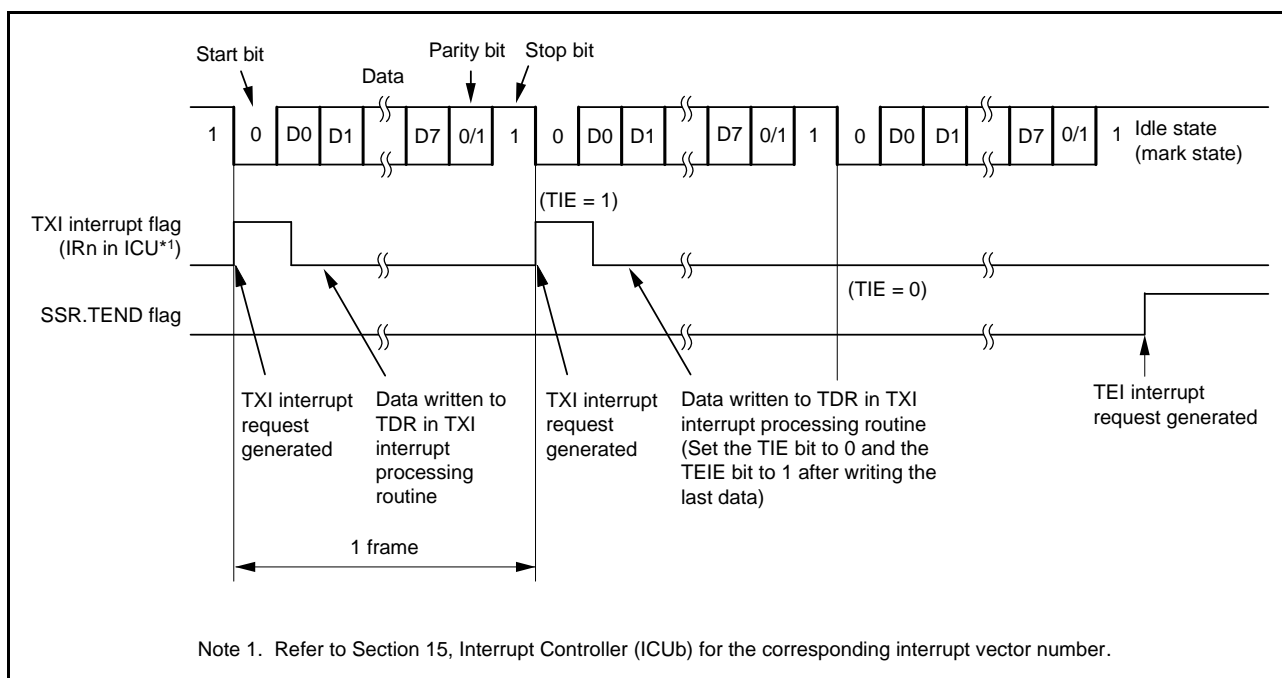
Figure 32.8 Sample SCI Initialization Flowchart (Asynchronous Mode)

### 32.3.6 Serial Data Transmission (Asynchronous Mode)

Figure 32.9 shows an example of the operation for serial transmission in asynchronous mode. In serial transmission, the SCI operates as described below.

1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt processing routine. The TXI interrupt request at the beginning of transmission is generated when the TE bit in SCR is set to 1 after the TIE bit in SCR is set to 1 or when these two bits are set to 1 simultaneously by a single instruction.
2. Transmission starts after the CTSE bit in SPMR is set to 0 (disabling the CTS function) and a low level on the CTS# pin causes data transfer from TDR to TSR. If the TIE bit in SCR is 1 at this time, a TXI interrupt request is generated. Continuous transmission is obtainable by writing the next data for transmission to TDR in the TXI interrupt processing routine before transmission of the current data for transmission is completed. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (disabling TXI requests) and the SCR.TEIE bit to 1 (enabling TEI requests) after the last of the data to be transmitted are written to the TDR from the processing 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) TDR at the time of stop bit output.
5. When TDR is updated, setting of the CTSE bit in SPMR to 0 (CTS function disabled) or a low level input on the CTSn# pin cause the next transfer of the next data for transmission from TDR to TSR and sending of the stop bit, after which serial transmission of the next frame starts.
6. If TDR is not updated, the TEND flag in SSR is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output. If the TEIE flag in SCR is 1 at this time, the TEND flag in SSR is set to 1 and a TEI interrupt request is generated.

Figure 32.10 shows a sample flowchart for serial transmission in asynchronous mode.



**Figure 32.9 Example of Operation for Serial Transmission in Asynchronous Mode (from the Middle of Transmission until Transmission Completion) (Example with 8-Bit Data, Parity, One Stop Bit)**

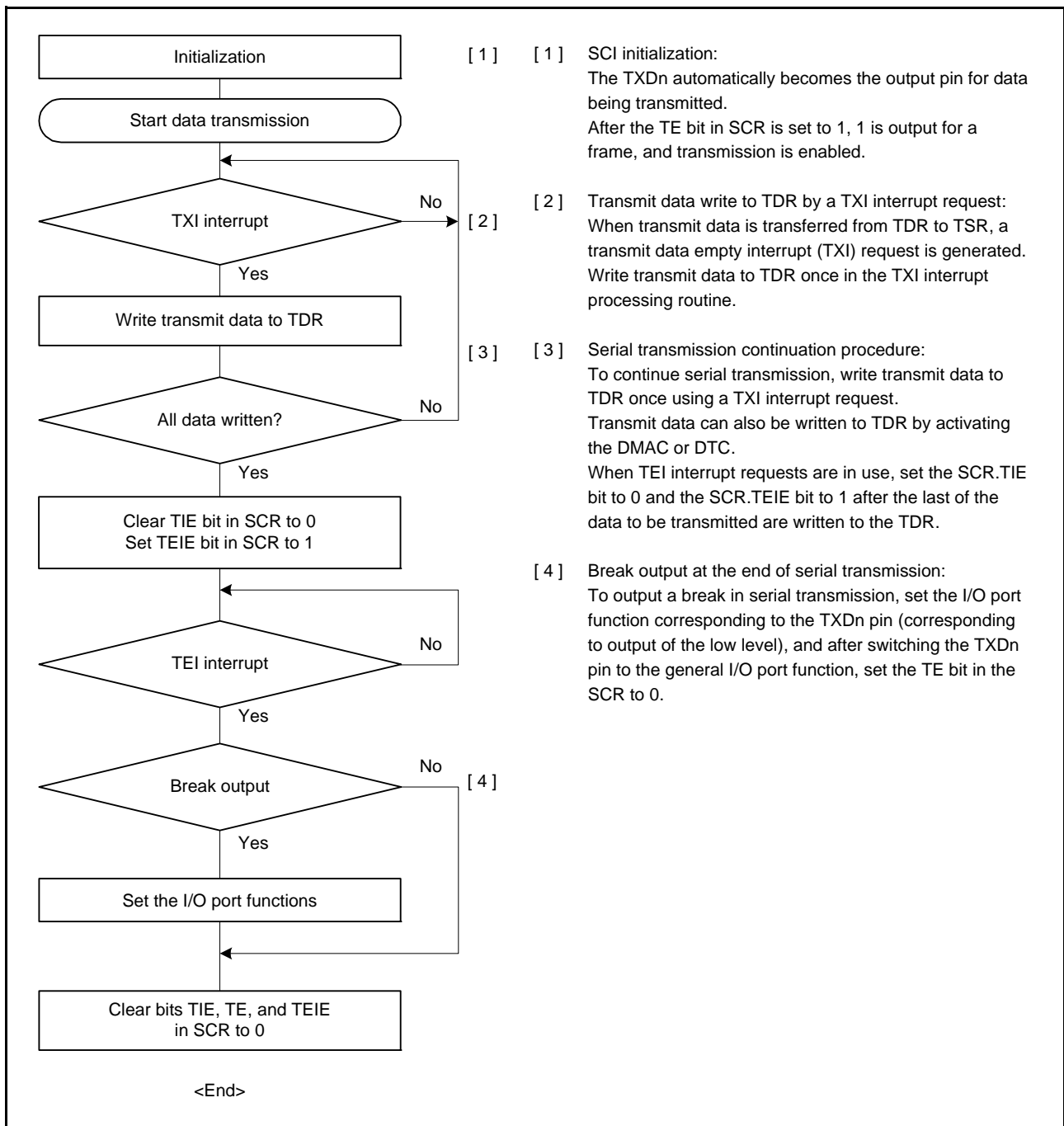
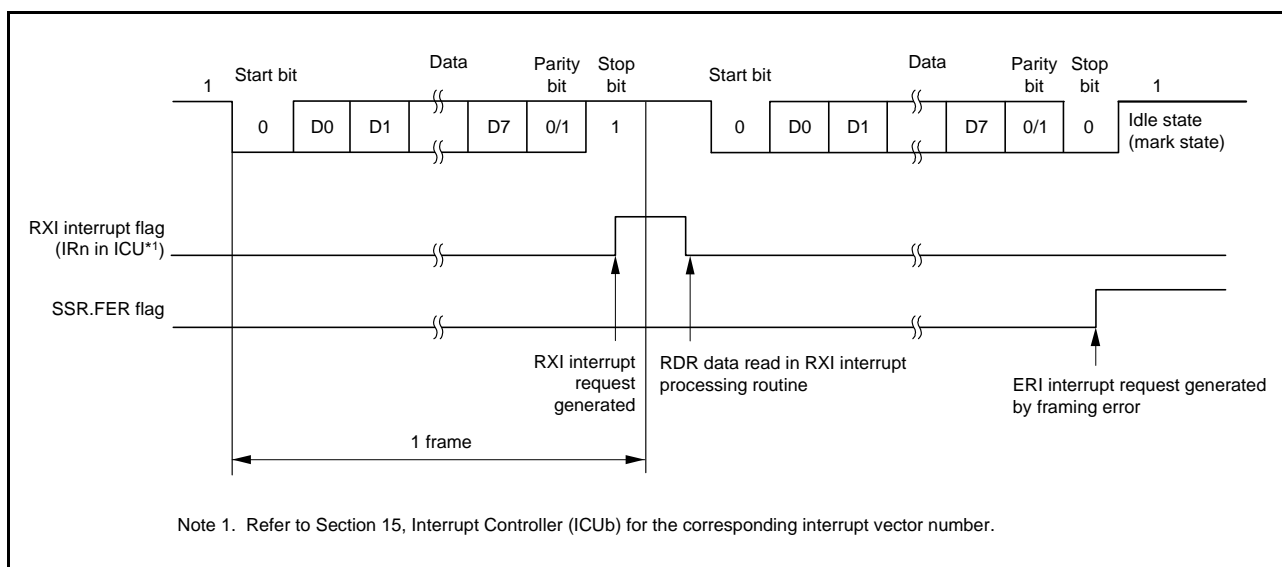


Figure 32.10 Example of Serial Transmission Flowchart in Asynchronous Mode

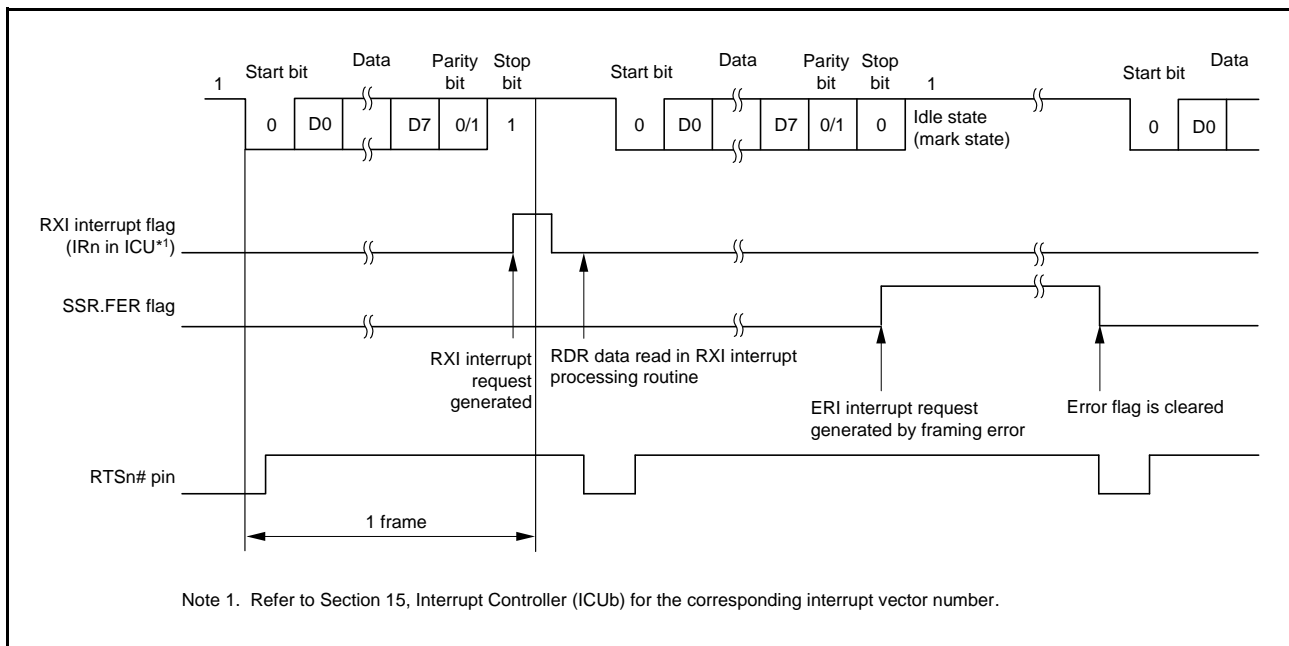
### 32.3.7 Serial Data Reception (Asynchronous Mode)

Figure 32.11 and Figure 32.12 show examples 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 RE bit in SCR 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 RSR, and checks the parity bit and stop bit.
3. If an overrun error occurs, the ORER flag in SSR is set to 1. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR.
4. If a parity error is detected, the PER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR 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 FER bit in SSR is set to 1 and receive data is transferred to RDR. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated.
6. When reception finishes successfully, receive data is transferred to RDR. If the RIE bit in SCR is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to RDR in this RXI interrupt processing routine before reception of the next receive data is completed. Reading out the received data that have been transferred to RDR causes the RTSn# pin to output the low level.



**Figure 32.11 Example of SCI Operation for Serial Reception in Asynchronous Mode (1) (when RTS Function is not Used) (Example with 8-Bit Data, Parity, One Stop Bit)**

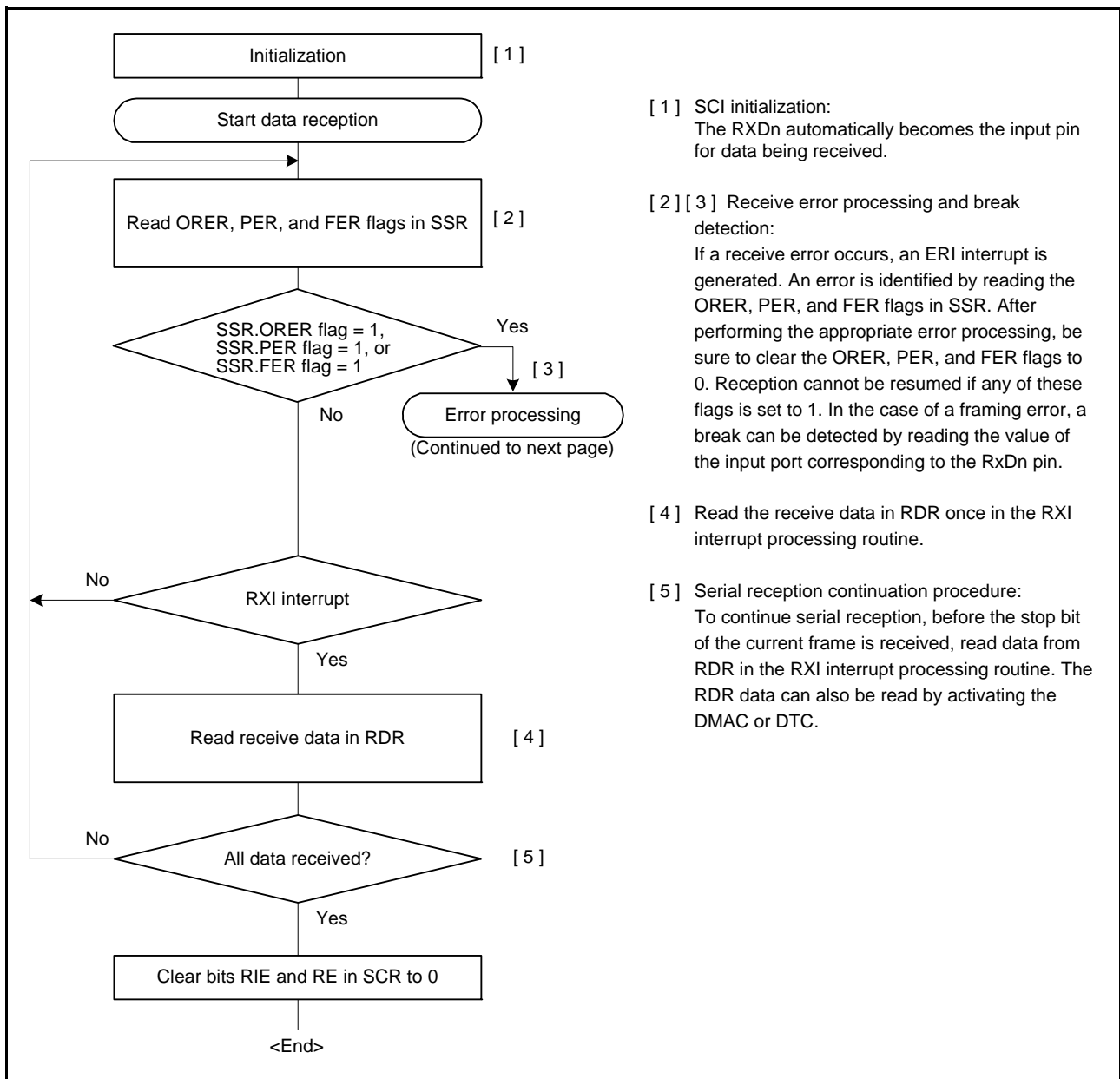


**Figure 32.12 Example of SCI Operation for Serial Reception in Asynchronous Mode (2) (when RTS Function Is Used) (Example with 8-Bit Data, Parity, One Stop Bit)**

Table 32.23 lists the states of the SSR status flags 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, clear the ORER, FER, and PER bits to 0 before resuming reception. Moreover, be sure to read the RDR during overrun error processing. Figure 32.13 and Figure 32.14 show samples of flowcharts for serial data reception.

**Table 32.23 SSR Status Flags and Receive Data Handling**

SSR Status Flag			Receive Data	Receive Error Type
ORER	FER	PER		
1	0	0	Lost	Overrun error
0	1	0	Transferred to RDR	Framing error
0	0	1	Transferred to RDR	Parity error
1	1	0	Lost	Overrun error + framing error
1	0	1	Lost	Overrun error + parity error
0	1	1	Transferred to RDR	Framing error + parity error
1	1	1	Lost	Overrun error + framing error + parity error



[ 1 ] SCI initialization:  
The RXDn automatically becomes the input pin for data being received.

[ 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 SSR. After performing the appropriate error processing, be sure to clear 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 RDR once in the RXI interrupt processing routine.

[ 5 ] Serial reception continuation procedure:  
To continue serial reception, before the stop bit of the current frame is received, read data from RDR in the RXI interrupt processing routine. The RDR data can also be read by activating the DMAC or DTC.

Figure 32.13 Example of Serial Reception Flowchart (1) (Asynchronous Mode)

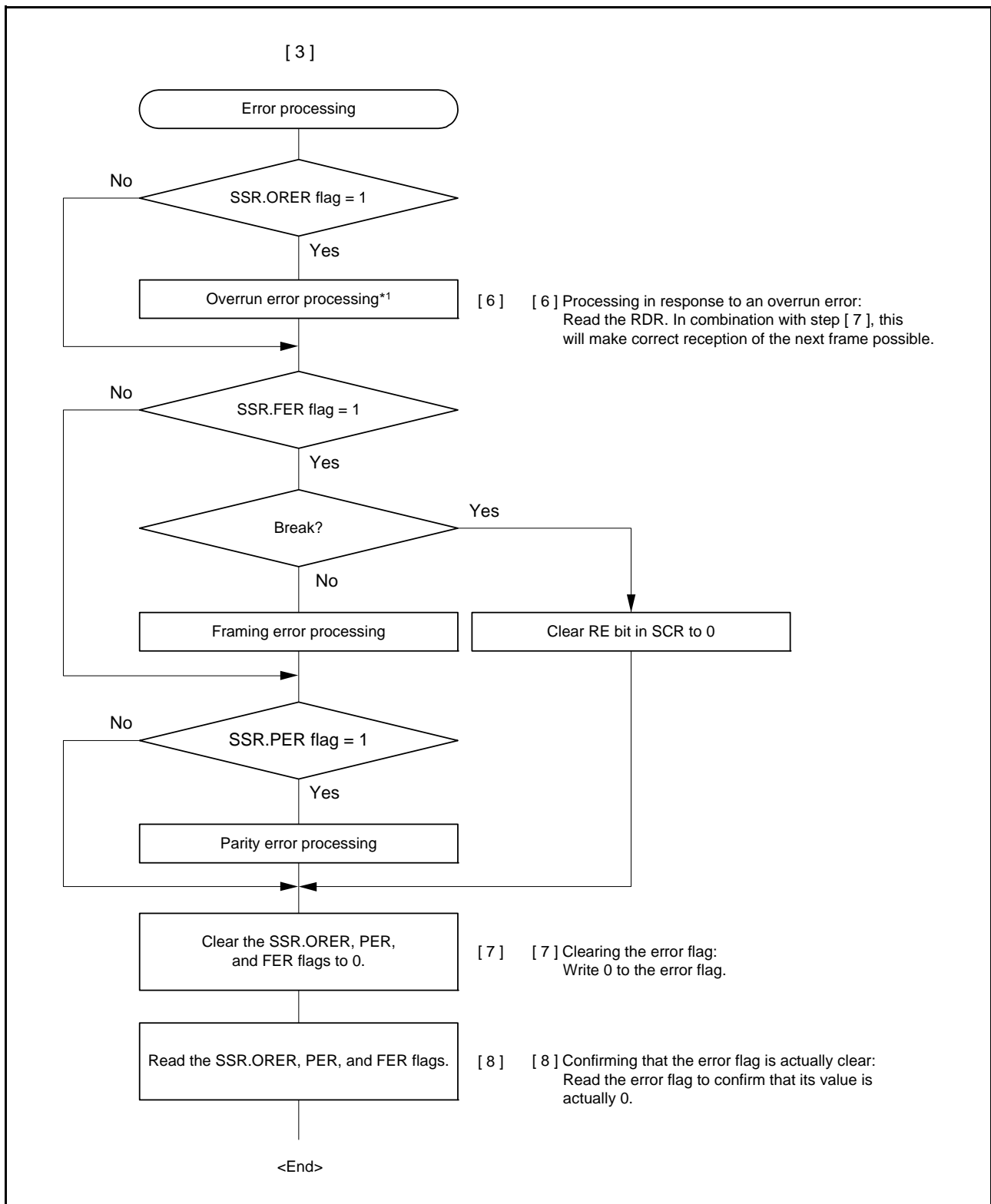
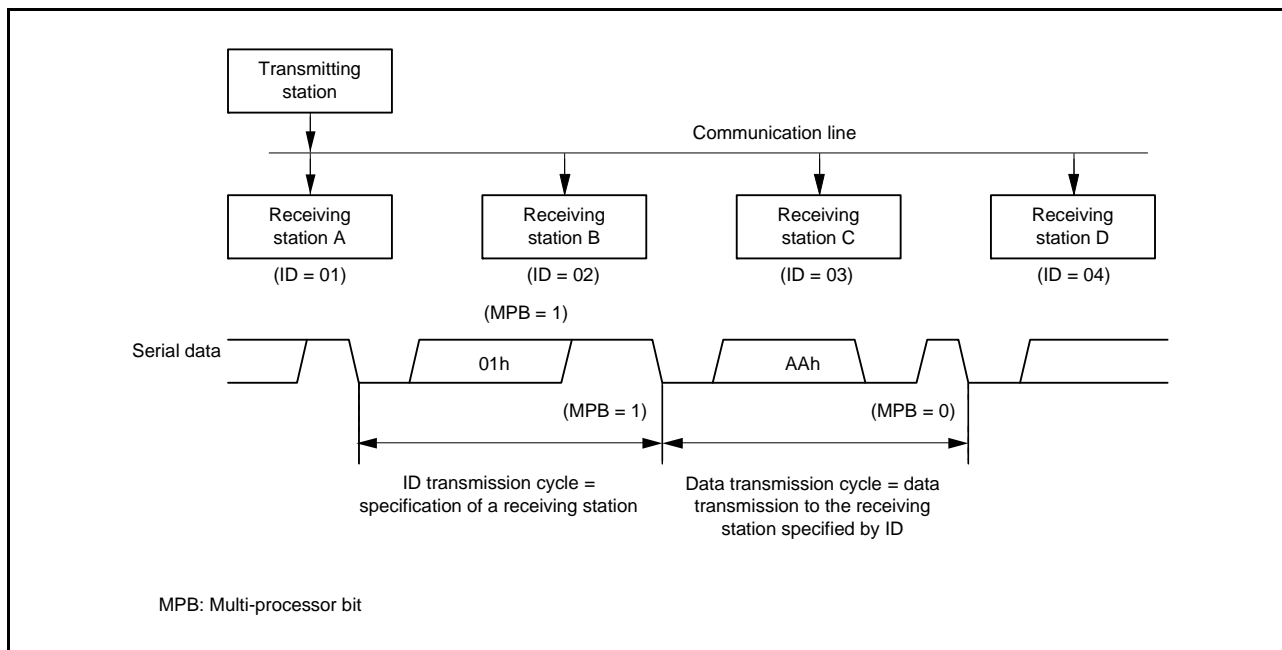


Figure 32.14 Example of Serial Reception Flowchart (2) (Asynchronous Mode)

### 32.4 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 32.15 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 transmission 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 MPIE bit in SCR. When the MPIE bit is set to 1, transfer of receive data from the RSR to the RDR, detection of a reception error, and setting the respective status flags ORER and FER in SSR 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 MPBT bit in SSR is set to 1 and the MPIE bit in SCR is automatically cleared, thus returning to a normal reception operation. During this time, an RXI interrupt is generated if the RIE bit in SCR is set.

When the multi-processor format is specified, specification of the parity bit is disabled. Apart from this, there is no difference from the operation in the 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 32.15 An Example of Communication using the Multi-Processor Format (Example of Transmission of Data AAh to Receiving Station A)**



### 32.4.1 Multi-Processor Serial Data Transmission

Figure 32.16 is a sample flowchart of multi-processor data transmission. In the ID transmission cycle, the ID should be transmitted with the MPBT bit in SSR 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.

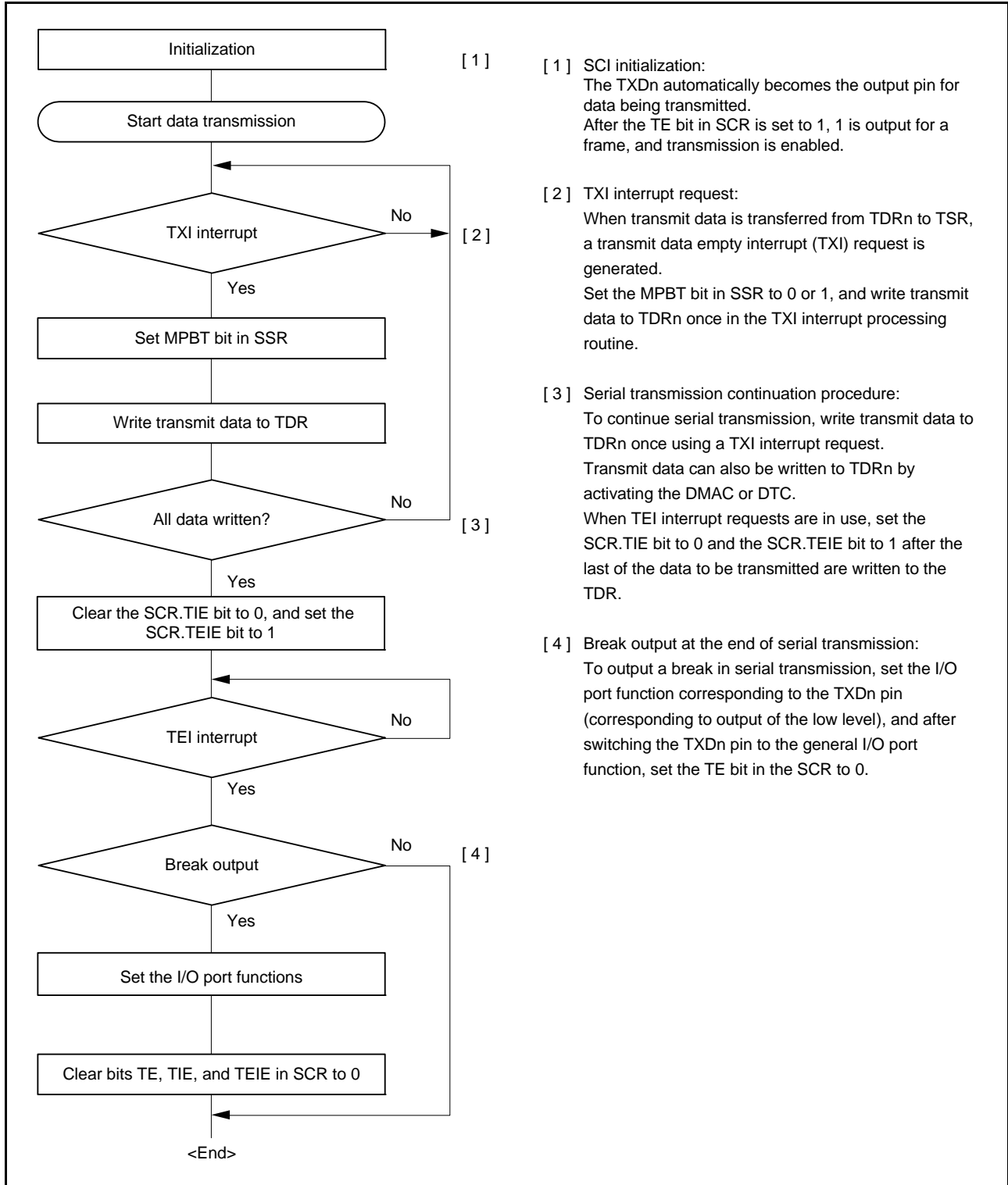


Figure 32.16 Example of Multi-Processor Serial Transmission Flowchart

### 32.4.2 Multi-Processor Serial Data Reception

Figure 32.18 and Figure 32.19 are sample flowcharts of multi-processor data reception. When the MPIO bit in SCR is set to 1, reading the communication data is skipped until reception of the communication data in which the multi-processor bit is set to 1. When the communication data in which the multi-processor bit is set to 1 is received, the received data is transferred to RDR. During this time, the RXI interrupt request is generated. The other operations are the same as the operations in asynchronous mode.

Figure 32.17 is the example of operation for reception.

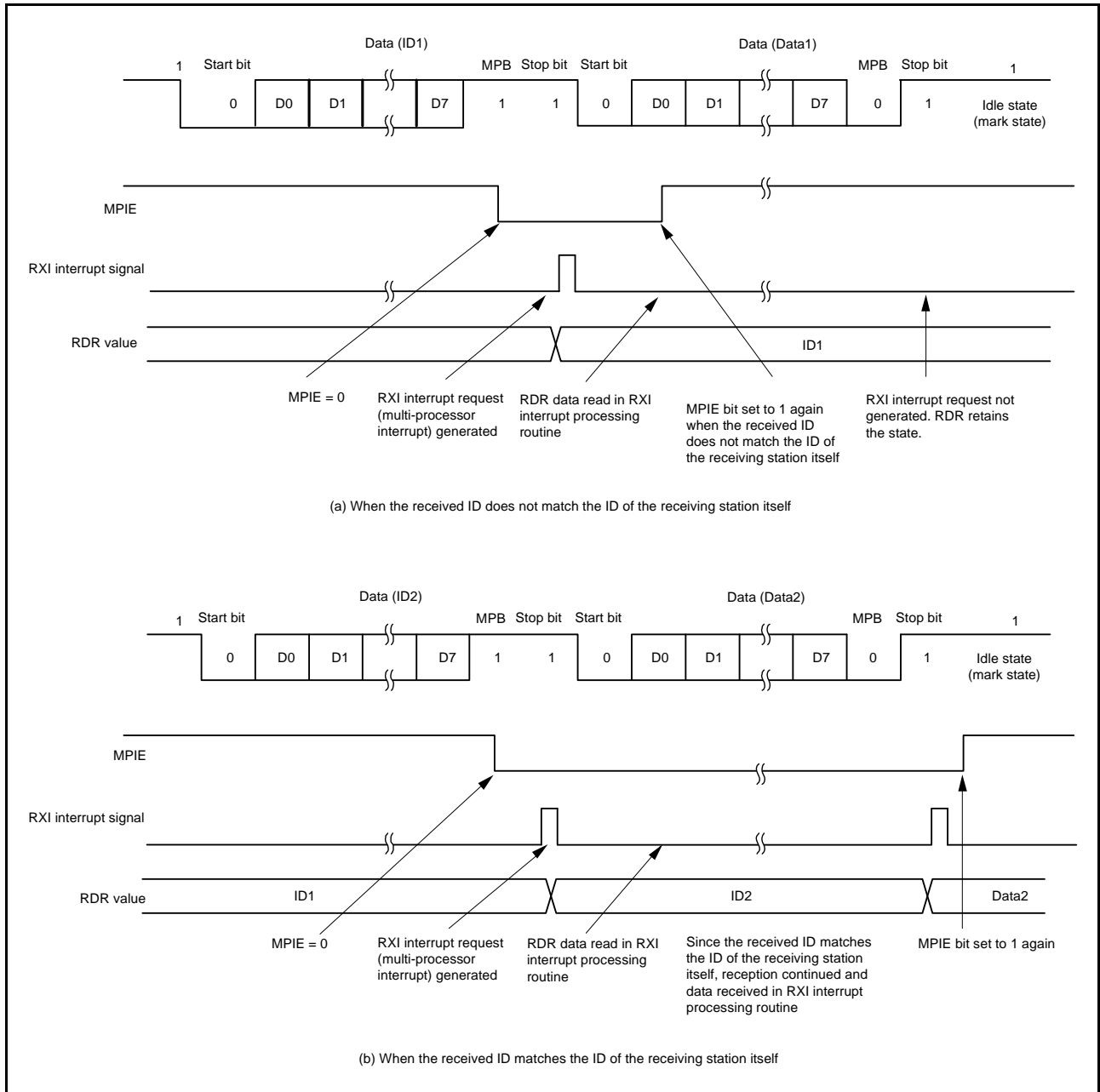


Figure 32.17 Example of SCI Reception (8-Bit Data/Multi-Processor Bit/One Stop Bit)

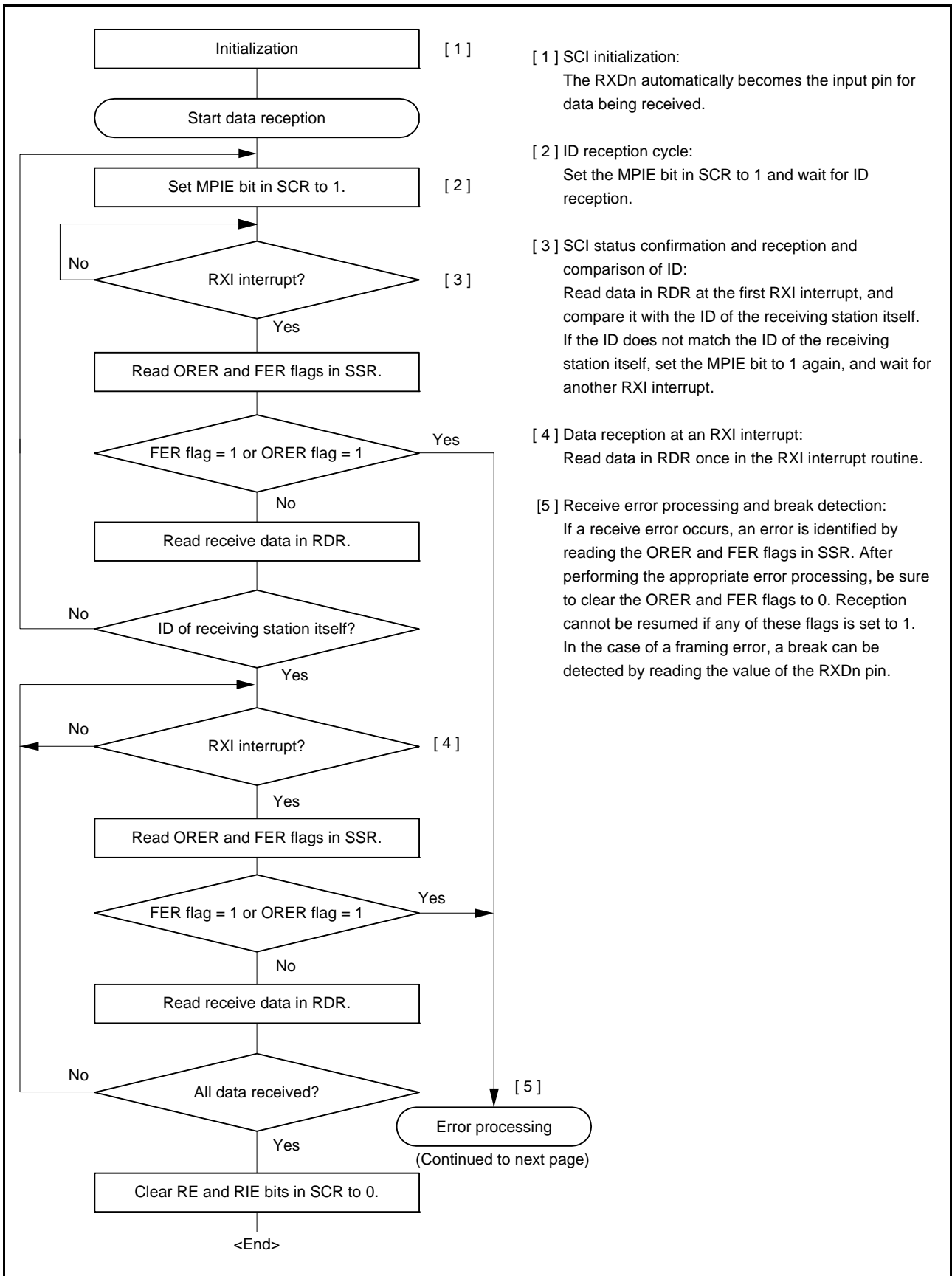


Figure 32.18 Example of Multi-Processor Serial Reception Flowchart (1)

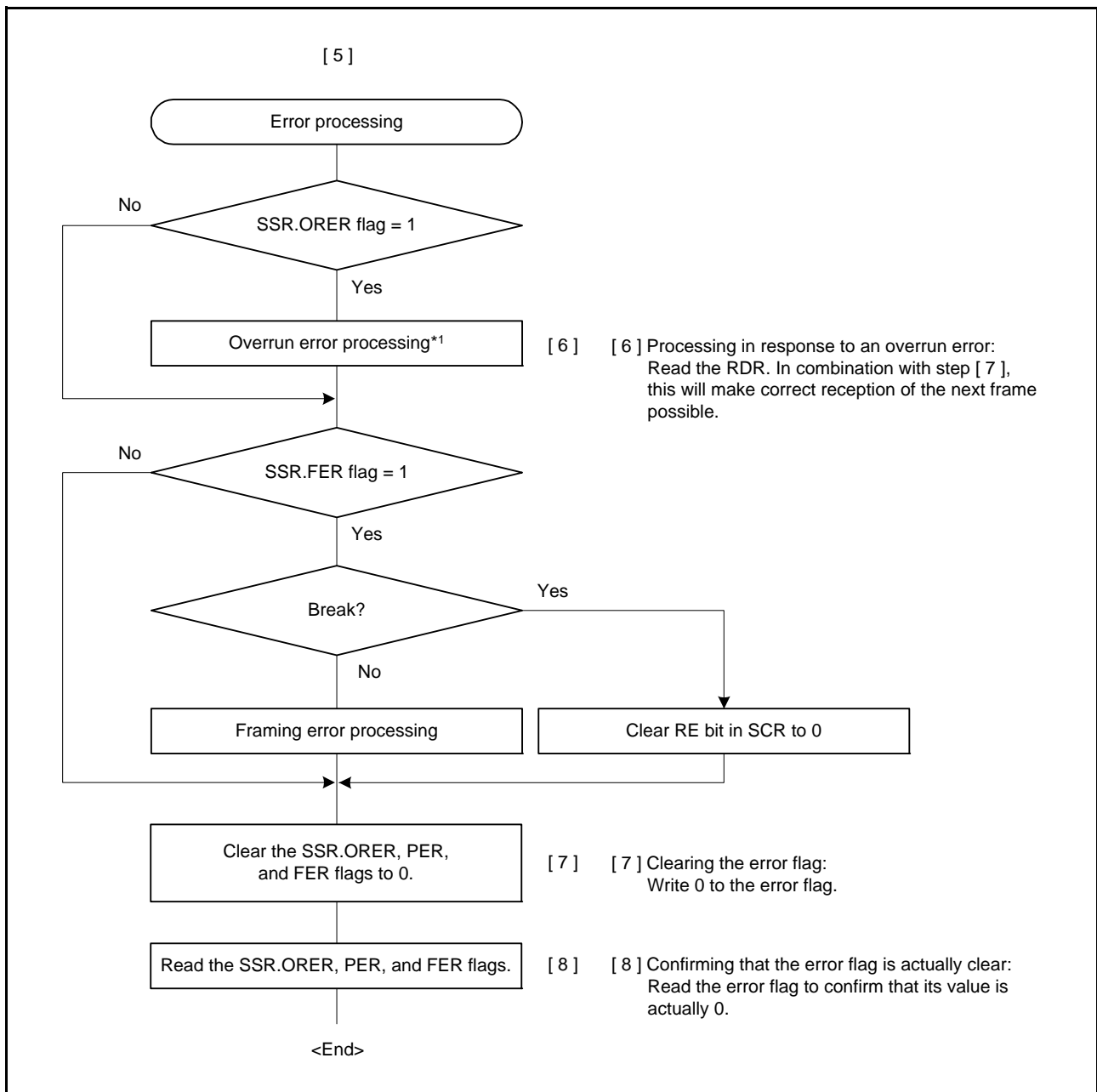


Figure 32.19 Example of Multi-Processor Serial Reception Flowchart (2)

## 32.5 Operation in Clock Synchronous Mode

Figure 32.20 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 transmission 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.

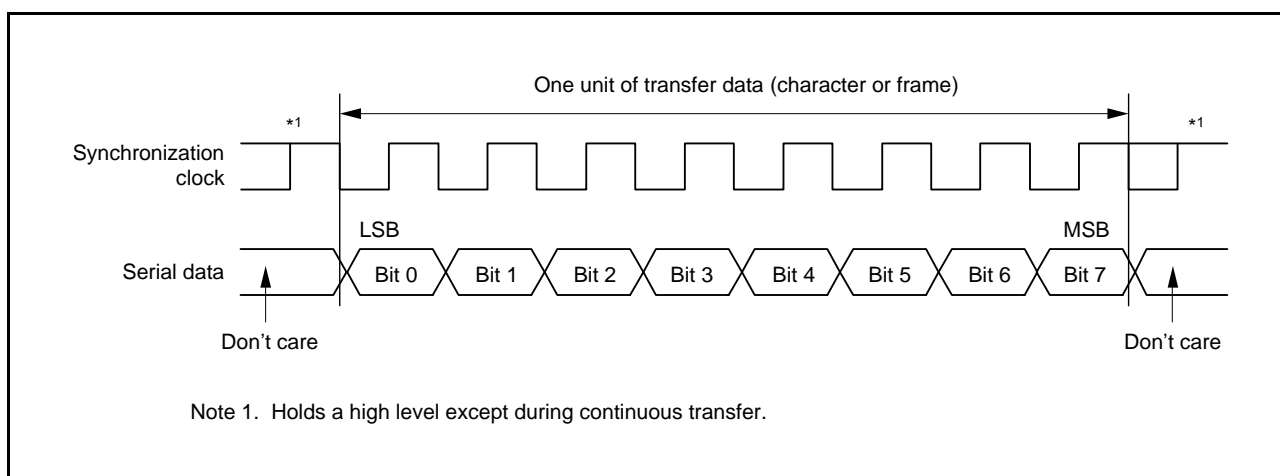


Figure 32.20 Data Format in Clock Synchronous Serial Communications (LSB-First)

### 32.5.1 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 CKEn[1:0] bits in SCR.

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, output of the synchronizing clock signal continues until the CTS function is enabled and the high level is input on the CTSn# pin, an overflow error occurs, or the RE bit in SCR is set to 0. When the CTS function is enabled, the synchronous clock signal output is stopped if the CTSn# pin input is high on completion of the frame reception.

### 32.5.2 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.

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]

Satisfaction of all conditions listed below

- The value of the RE or TE bit in the SCR is 1
- Neither transmission nor reception is in progress
- There are no received data yet to be read (when the SCR.RE bit is 1)
- Transmit data has been written (when the SCR.TE bit is 1)
- ORER flag in SSR is 0

[Condition for high-level output]

Any of the conditions for the low level not being satisfied

### 32.5.3 SCI Initialization (Clock Synchronous Mode)

Before transmitting and receiving data, start by writing the initial value “00h” to the SCR and then continue through the procedure for SCI given in the sample flowchart (Figure 32.21). Whenever the operating mode or transfer format is changed, the SCR must be initialized before the change is made.

Note that clearing the SCR.RE bit to 0 initializes neither the ORER, FER, and PER flags in SSR nor RDR.

Moreover, note that switching the value of the SCR.TE bit from 1 to 0 or 0 to 1 while the SCR.TIE bit is 1 leads to the generation of a TXI interrupt request.

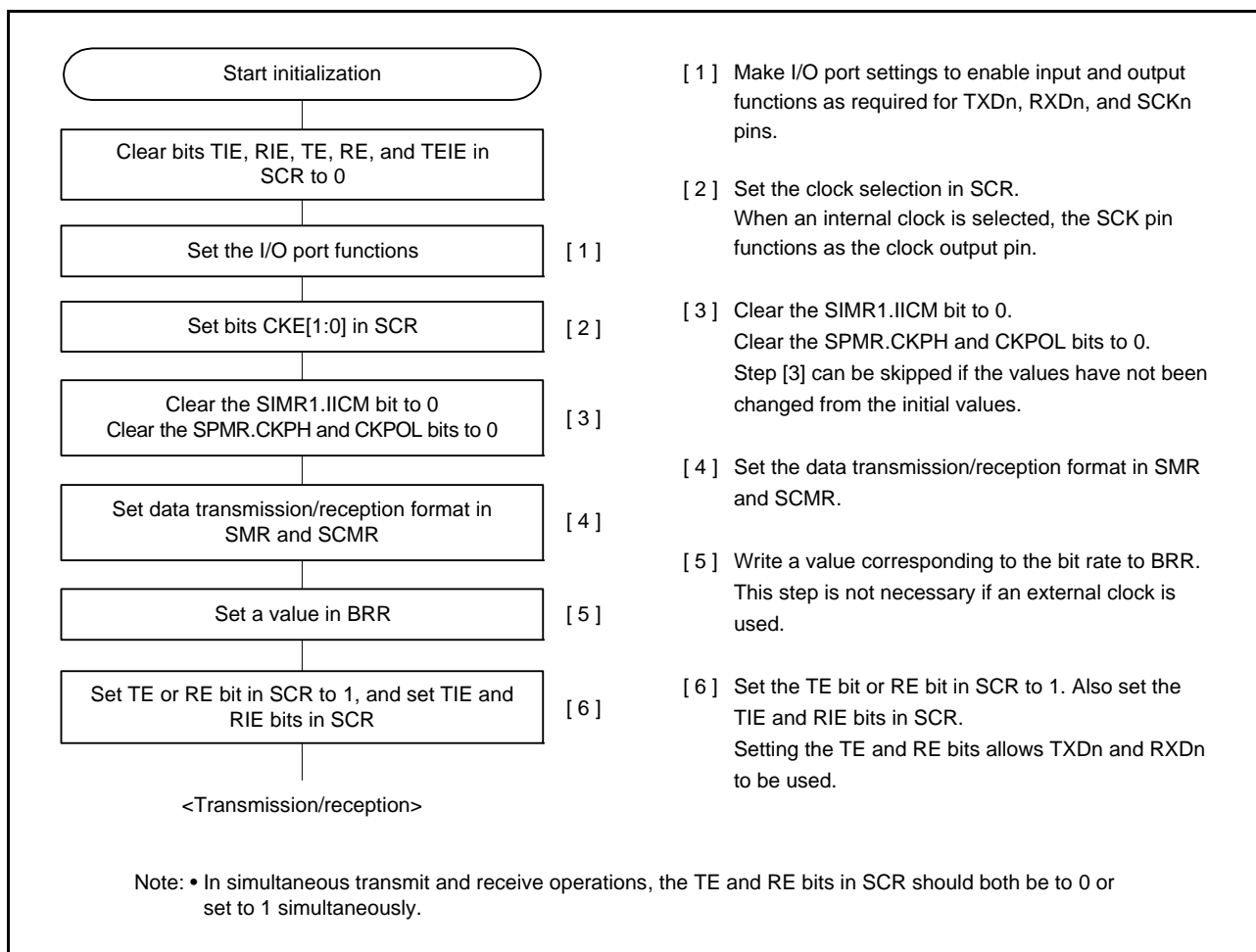


Figure 32.21 Example of SCI Initialization Flowchart (Clock Synchronous Mode)

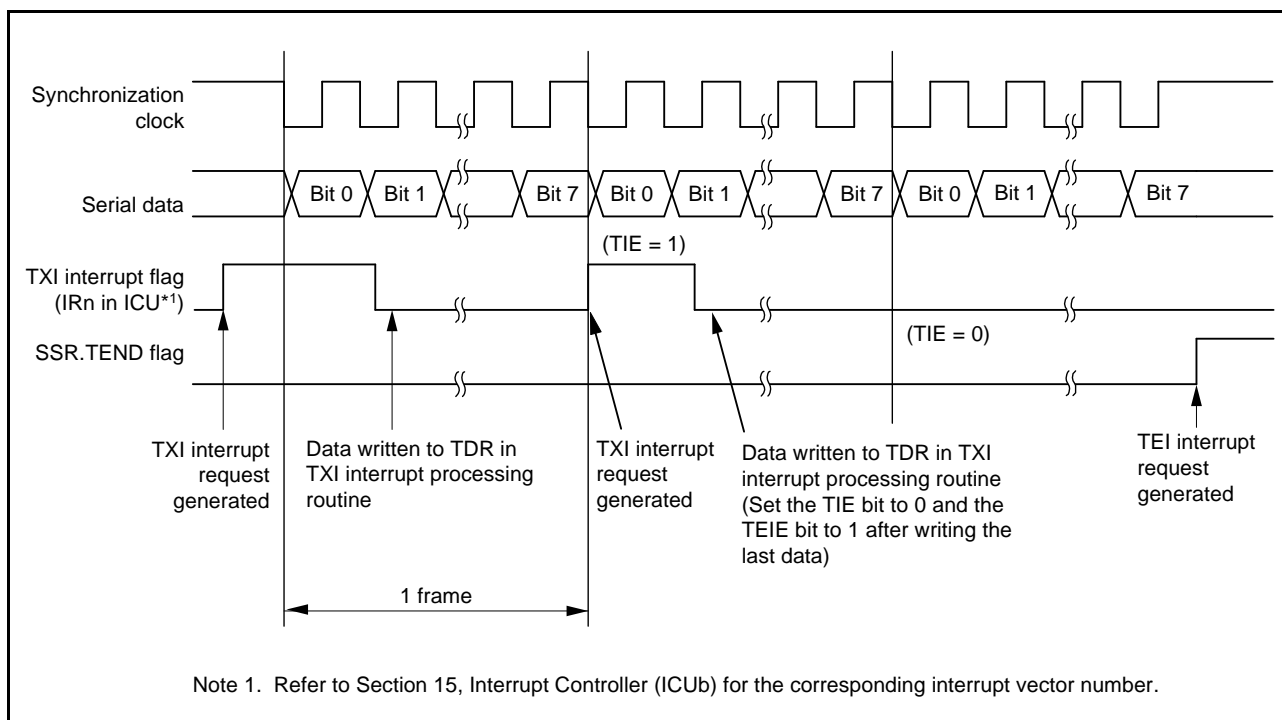
### 32.5.4 Serial Data Transmission (Clock Synchronous Mode)

Figure 32.22 shows an example of the operation for serial transmission in clock synchronous mode. In serial data transmission, the SCI operates as described below.

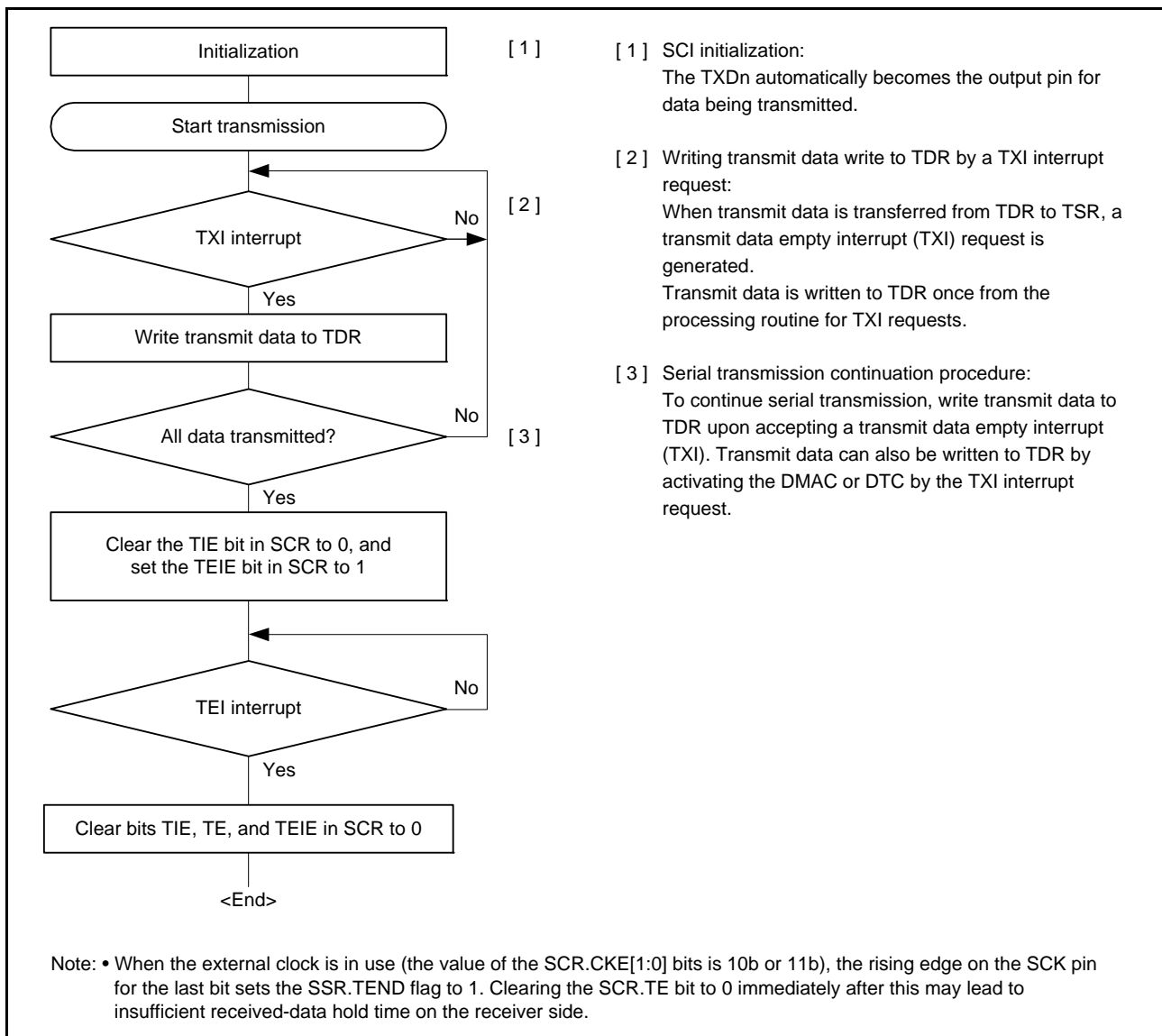
1. The SCI transfers data from TDR to TSR when data is written to TDR in the TXI interrupt processing routine. The TXI interrupt request at the beginning of transmission is generated when the TE bit in SCR is set to 1 after the TIE bit in SCR is set to 1 or when these two bits are set to 1 simultaneously by a single instruction.
2. After transferring data from TDR to TSR, 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 TDR in this TXI interrupt processing routine before transmission of the current transmit data has finished. When TEI interrupt requests are in use, set the SCR.TIE bit to 0 (disabling TXI requests) and the SCR.TEIE bit to 1 (enabling TEI requests) after the last of the data to be transmitted are written to the TDR from the processing 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 CTSE bit in SPMR is 1 (enabling the CTS function).
4. The SCI checks for updating of (writing to) the TDR at the time of the last bit output.
5. When TDR is updated, the next transmit data is transferred from TDR to TSR, and serial transmission of the next frame is started.
6. If TDR is not updated, set the SSR flag in TEND to 1 and the TXDn pin retains the output state of the last bit. If the TEIE bit in SCR is 1 at this time, a TEI interrupt request is generated. The SCKn pin is held high.

Figure 32.23 shows a sample flowchart of serial data transmission.

Transmission will not start while a receive error flag (ORER, FER, or PER in SSR) is set to 1. Be sure to clear the receive error flags to 0 before starting transmission. Note that clearing the RE bit in SCR to 0 does not clear the receive error flags.



**Figure 32.22 Example of Operation for Serial Transmission in Clock Synchronous Mode (from the Middle of Transmission until Transmission Completion)**



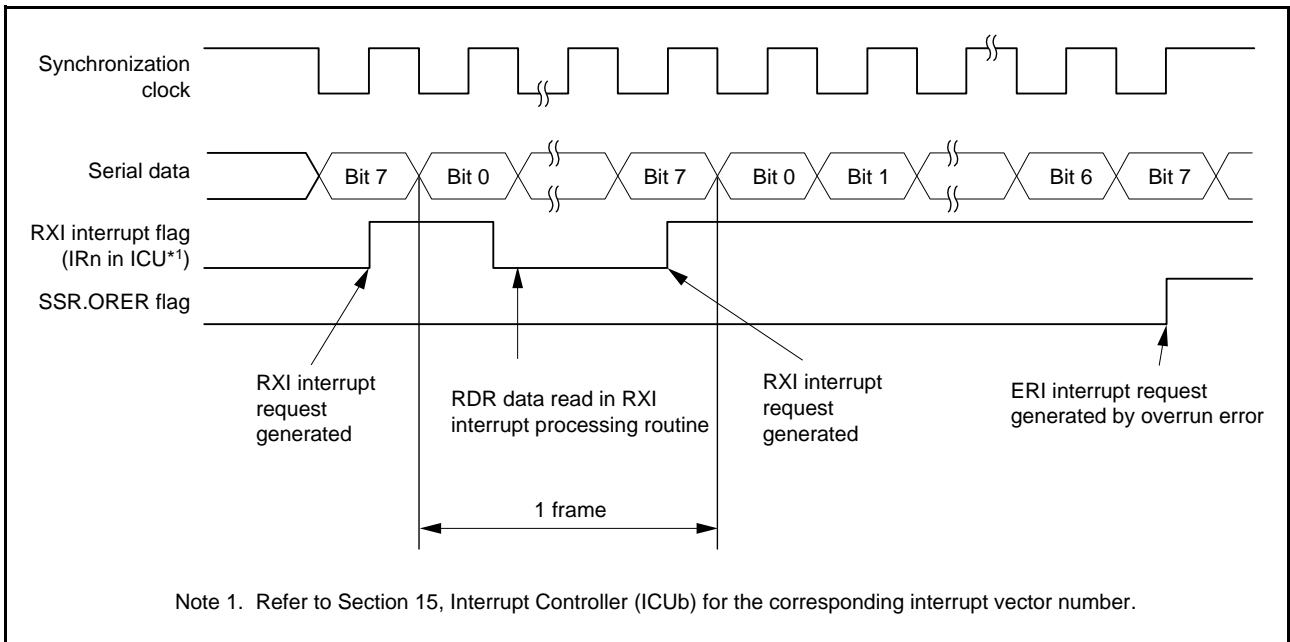
**Figure 32.23 Example of Serial Transmission Flowchart (Clock Synchronous Mode)**

### 32.5.5 Serial Data Reception (Clock Synchronous Mode)

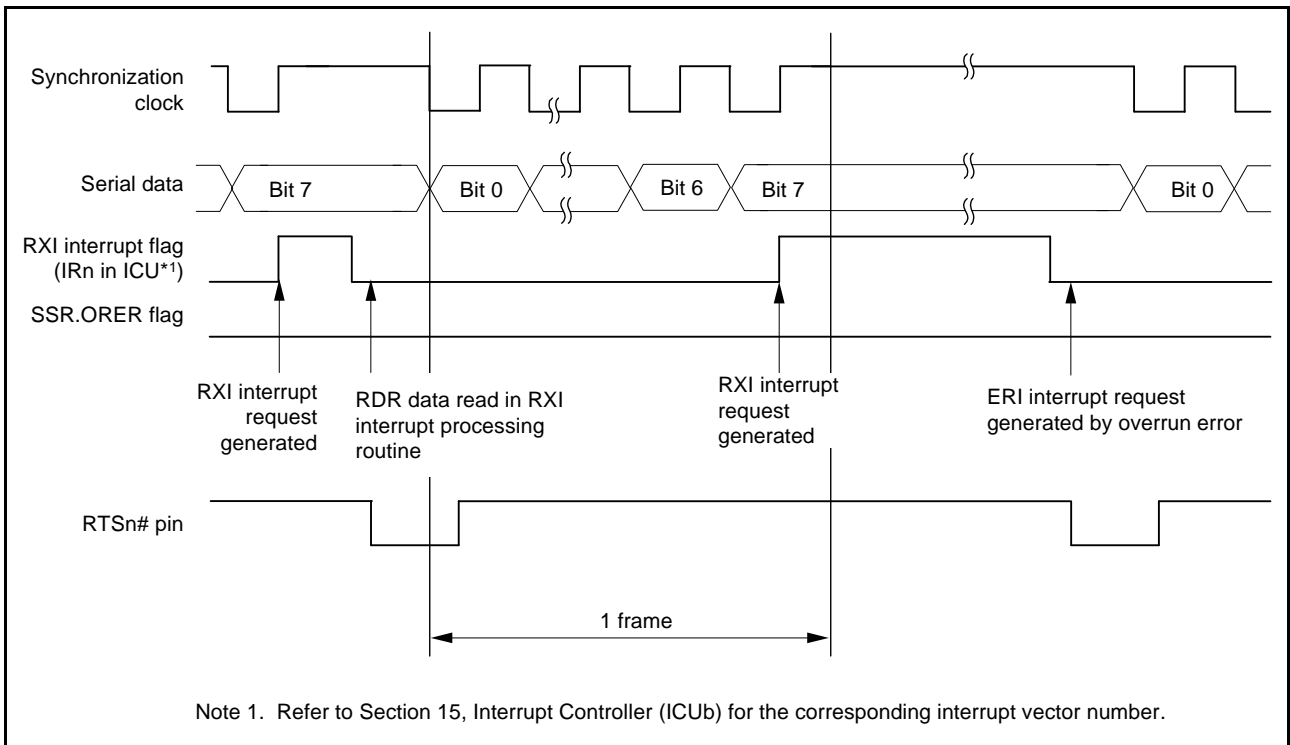
Figure 32.24 and Figure 32.25 show examples of SCI operation for serial reception in clock synchronous mode. In serial data reception, the SCI operates as described below.

1. The value of the RE bit in SCR becoming 1 places the signal output on the RTS 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 RSR.
3. If an overrun error occurs, the ORE bit in SSR is set to 1. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Receive data is not transferred to RDR.
4. When reception finishes successfully, receive data is transferred to RDR. If the RIE bit in SCR is 1 at this time, an RXI interrupt request is generated. Continuous reception is enabled by reading the receive data transferred to RDR in this RXI interrupt processing routine before reception of the next receive data is completed. Reading out the received data that have been transferred to RDR causes the RTSn# pin to output the low level (when the RTS function is in use).





**Figure 32.24 Example of Operation for Serial Reception in Clock Synchronous Mode (1) (when RTS Function is not Used)**



**Figure 32.25 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 bits in SSR to 0 before resuming reception. Moreover, be sure to read the RDR during overrun error processing.

Figure 32.26 shows a sample flowchart for serial data reception.

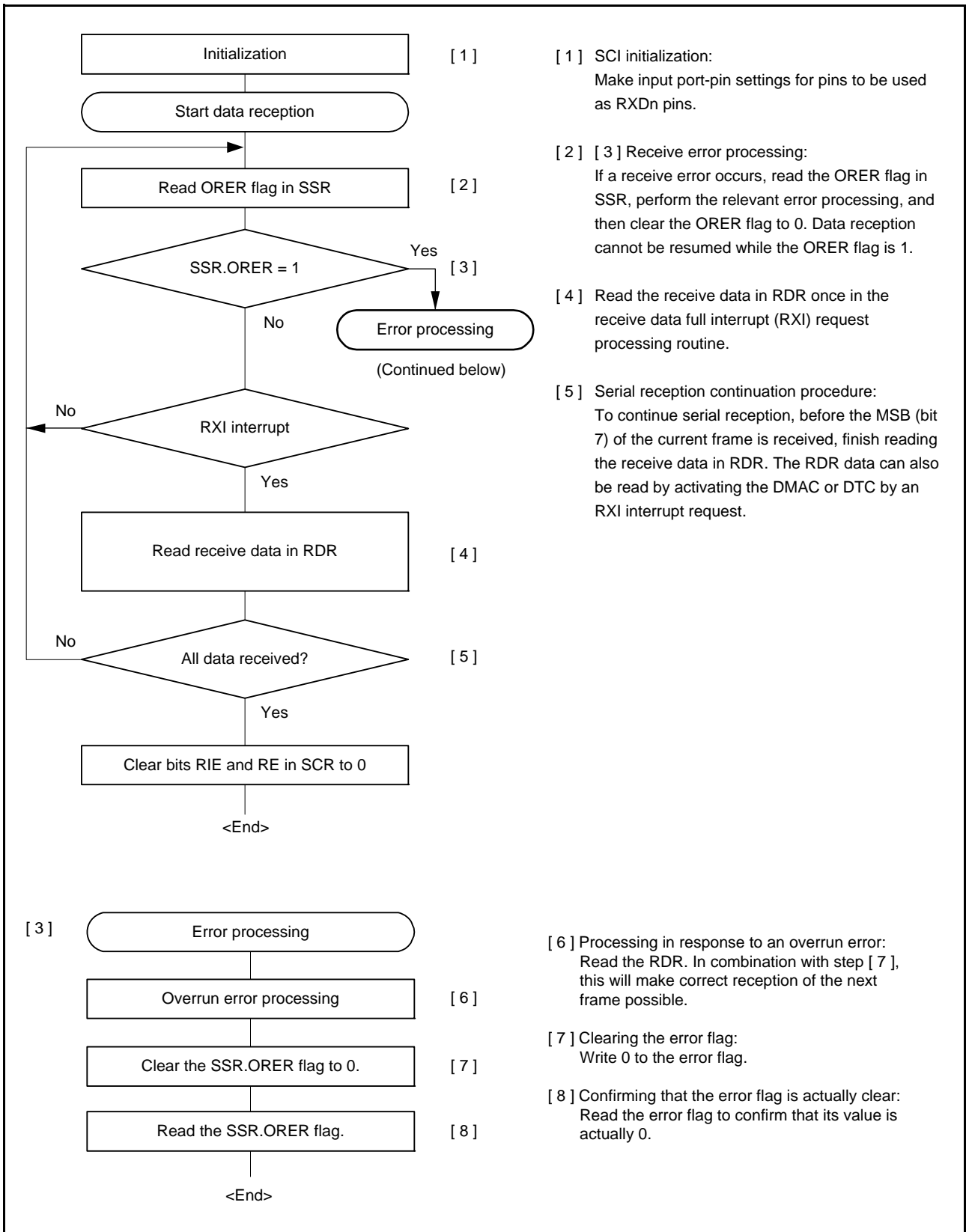


Figure 32.26 Example of Serial Reception Flowchart (Clock Synchronous Mode)

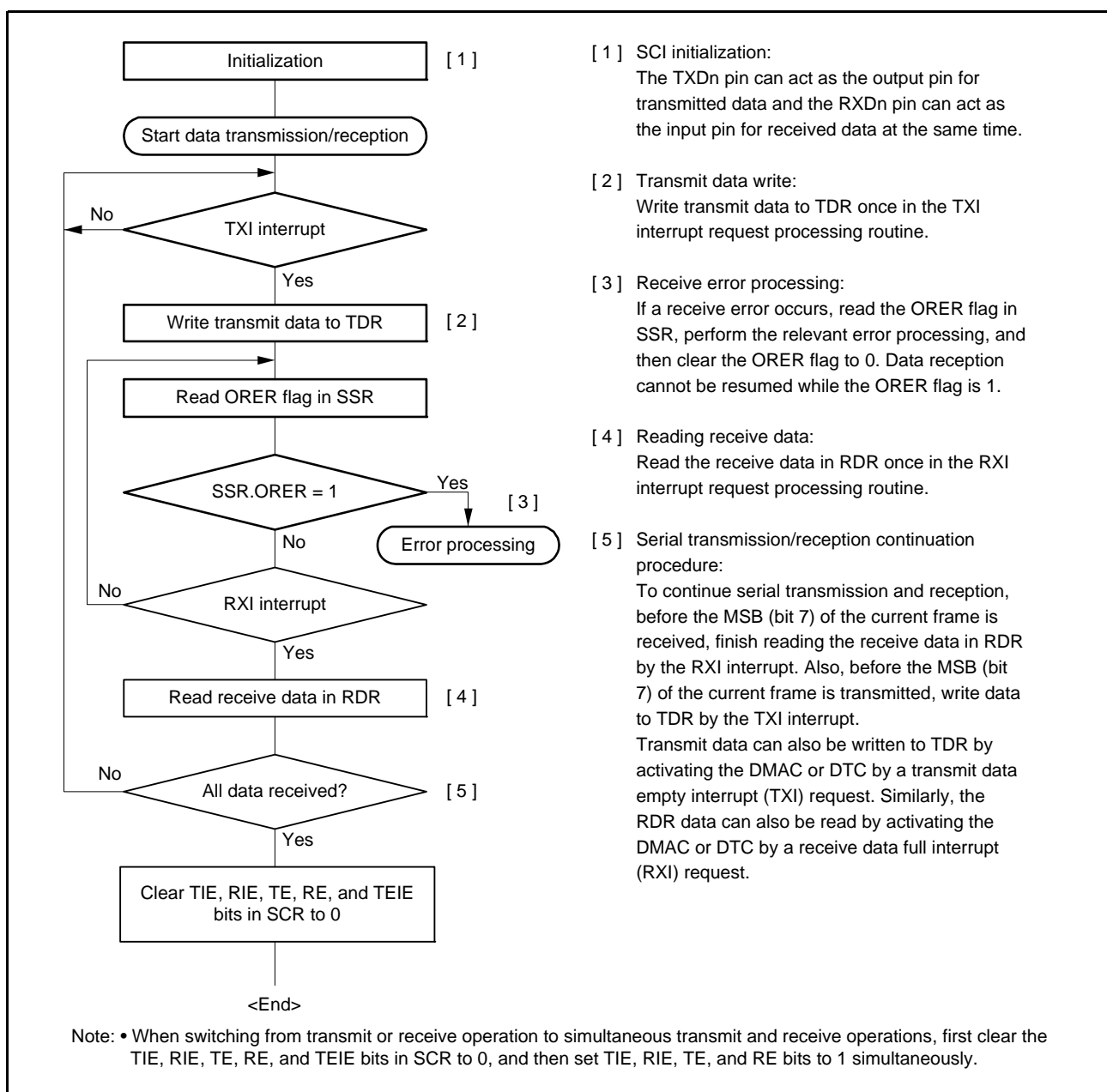
### 32.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode)

Figure 32.27 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 TEND flag in SSR is 1, and then initialize the SCR register. Then set the TIE, RIE, TE, and RE bits in SCR 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 clear the RIE and RE bits to 0. Then check that the receive error flags (ORER, FER, and PER in SSR) are 0, and then set the TIE, RIE, TE, and RE bits in SCR to 1 simultaneously by a single instruction.



**Figure 32.27 Example of Simultaneous Serial Transmission and Reception Flowchart (Clock Synchronous Mode)**

## 32.6 Operation in Smart Card Interface Mode

The SCI supports the smart card (IC card) interface conforming to the ISO/IEC 7816-3 (Identification Card) standard, as an extended function of the SCI.

Smart card interface mode can be selected using the appropriate register.

### 32.6.1 Sample Connection

Figure 32.28 shows a sample connection between a smart card (IC card) and this LSI.

As in the figure, since this LSI communicates with an IC card using a single transmission line, interconnect the TXDn and RXDn pins and pull up the data transmission line to Vcc using a resistor.

Setting the TE and RE bits in SCR 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 RX630 can be used to output a reset signal.

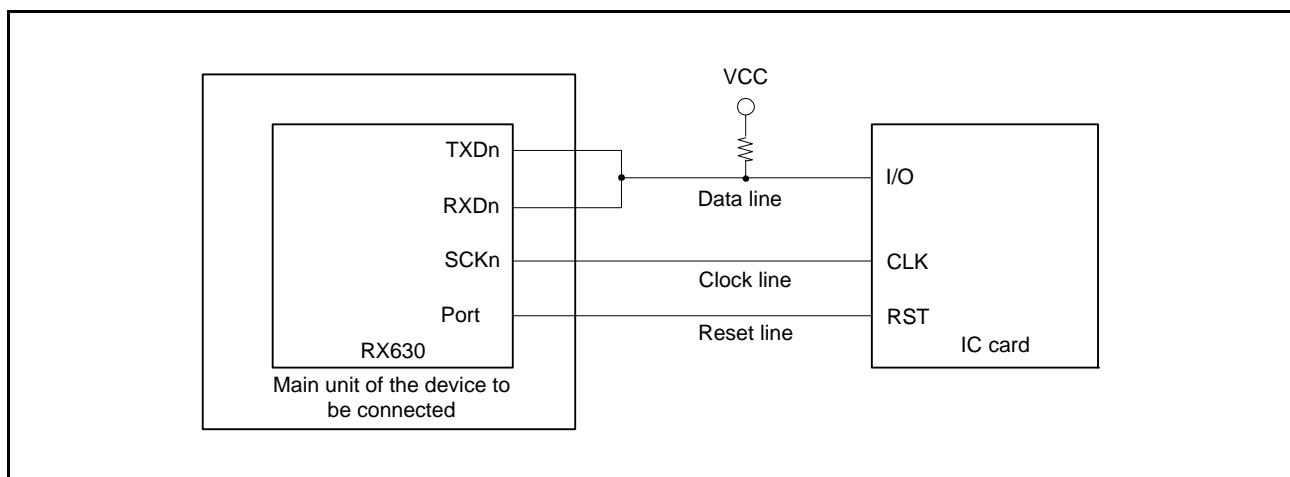
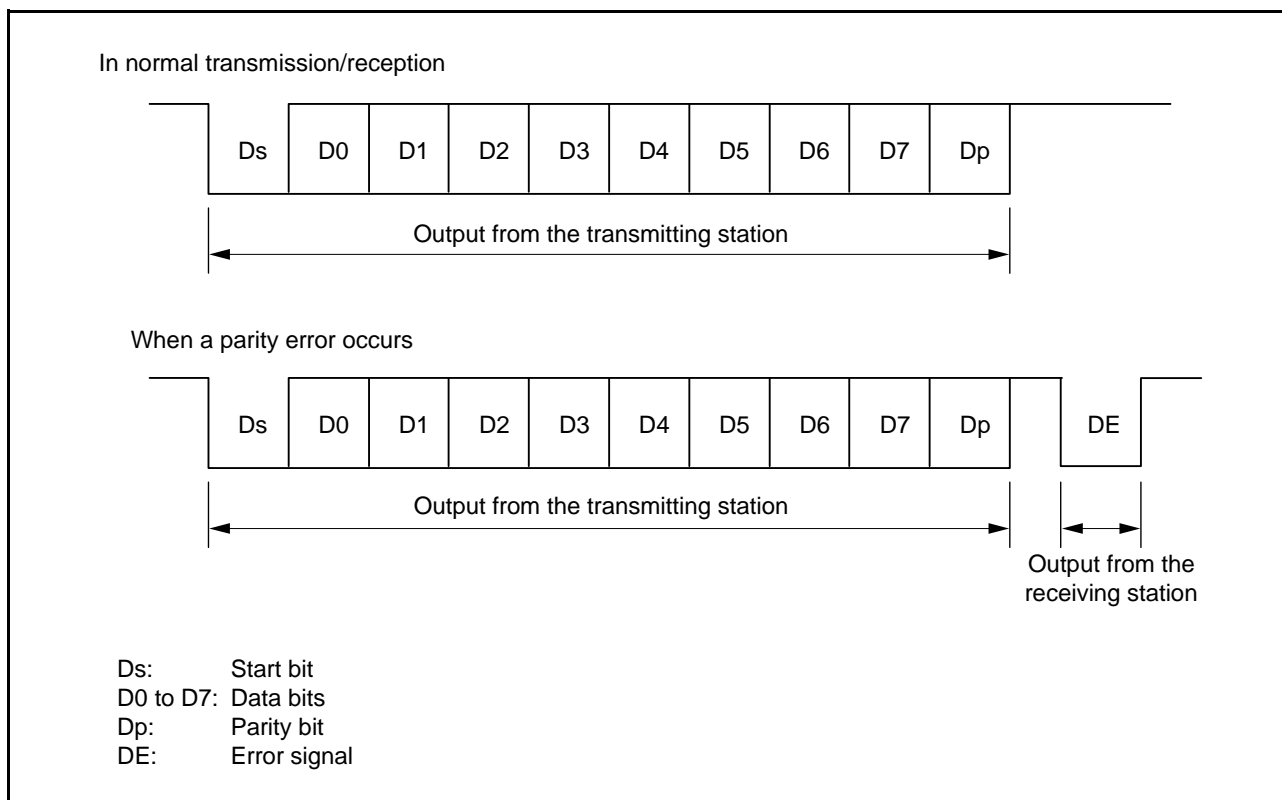


Figure 32.28 Sample Connection with a Smart Card (IC Card)

### 32.6.2 Data Format (Except in Block Transfer Mode)

Figure 32.29 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 one 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 re-transmitted after at least 2 etu.

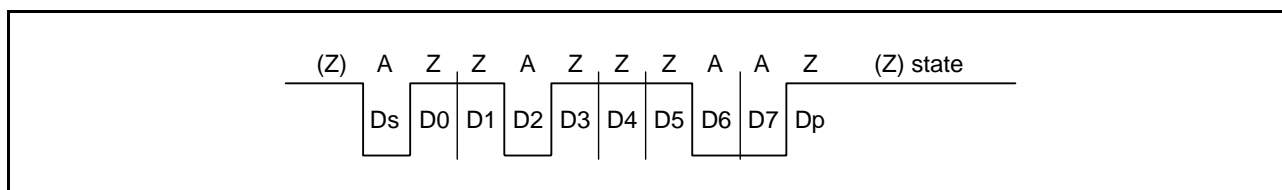


**Figure 32.29 Data Formats in Smart Card Interface Mode**

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 32.30. 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 SCMR. Write 0 to the PM bit in SMR in order to use even parity, which is prescribed by the smart card standard.



**Figure 32.30 Direct Convention (SDIR in SCMR = 0, SINV in SCMR = 0, PM in SMR = 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 32.31. 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 SCMR. 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 RX630 only inverts data bits D7 to D0, write 1 to the PM bit in SMR to invert the parity bit for both transmission and reception.

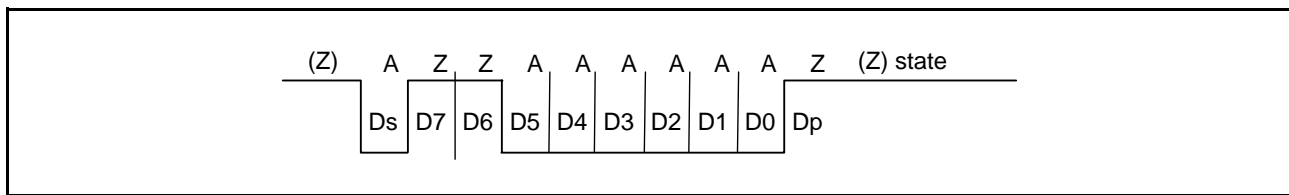


Figure 32.31 Inverse Convention (SDIR in SCMR = 1, SINV in SCMR = 1, PM in SMR = 1)

### 32.6.3 Block Transfer Mode

Block transfer mode is different from normal smart card interface mode in the following respects.

- Even if a parity error is detected during reception, no error signal is output. Since the PER bit in SSR is set by error detection, clear the PER bit before receiving the parity bit of the next frame.
- During transmission, at least 1 etu is secured as a guard time from the end of the parity bit until the start of the next frame.
- Since the same data is not re-transmitted during transmission, the TEND flag in SSR is set 11.5 etu after transmission start.
- In block transfer mode, the ERS flag in SSR indicates the error signal status as in normal smart card interface mode, but the flag is always read as 0 because no error signal is transferred.

### 32.6.4 Receive Data Sampling Timing and Reception Margin

Only the internal clock generated by the on-chip baud rate generator can be used as a transfer clock in smart card interface mode.

In this mode, the SCI can operate on a base clock with a frequency of 32, 64, 372, 256, 93, 128, 186, or 512 times the bit rate according to the settings of the BCP2 bit in SCMR and the BCP[1:0] bits in SMR (the frequency is always 16 times the bit rate in normal asynchronous mode).

For data reception, the falling edge of the start bit is sampled with the base clock to perform internal 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 32.32. 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)

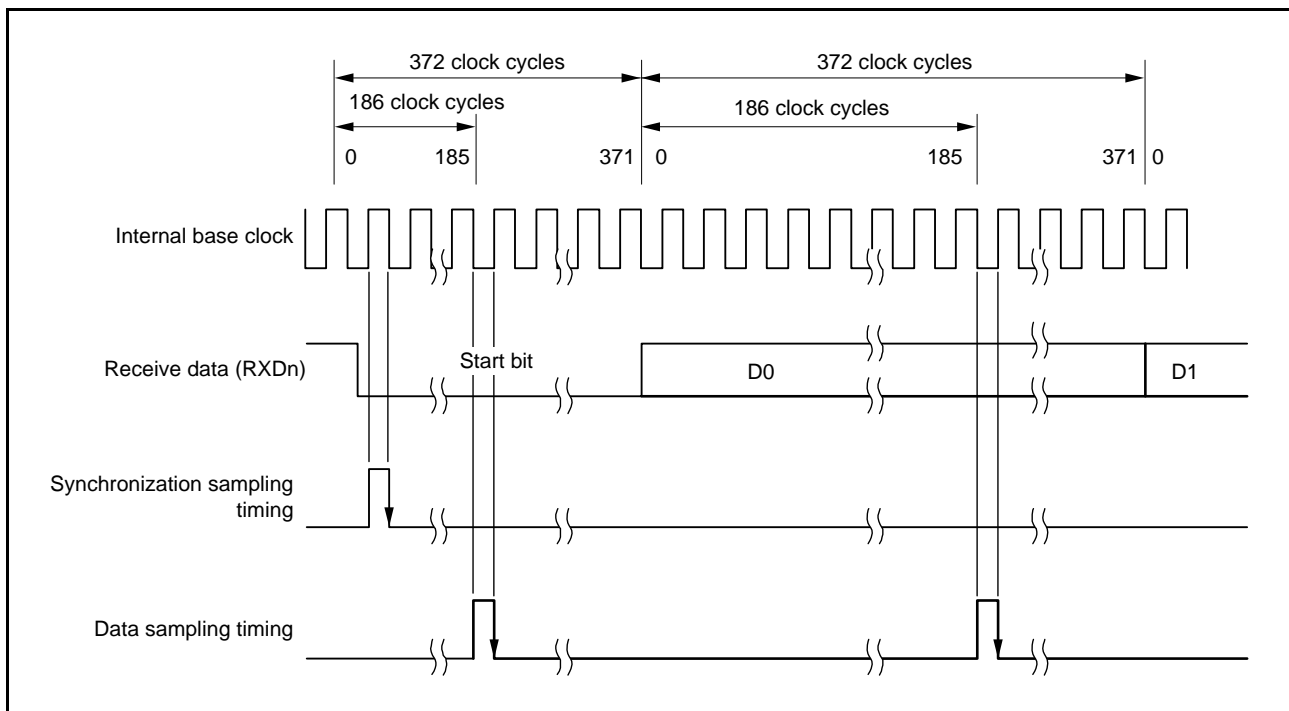
D: Duty cycle of clock (D = 0 to 1.0)

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.

$$M = \{0.5 - 1/(2 \times 372)\} \times 100 [\%] = 49.866\%$$



**Figure 32.32 Receive Data Sampling Timing in Smart Card Interface Mode (When Clock Frequency is 372 Times the Bit Rate)**

### 32.6.5 Initialization of the SCI (Smart Card Interface Mode)

Before transmitting and receiving data, initialize the SCI using the following procedure. Initialization is also necessary before switching from transmission to reception and vice versa.

1. Write the initial value "00h" to the SCR.
2. Make I/O port settings to enable input and output functions as required for TXDn, RXDn, and SCKn pins.
3. Set the error flags ORER, ERS, and PER in SSR to 0.
4. Clear the SIMR1.IICM bit and the SPMR.CKPH and CKPOL bits to 0.  
This step can be skipped if the values have not been changed from the initial values.
5. Set bits GM, BLK, OE, BCP[1:0], and CKS[1:0] in SMR and the BCP2 bit in SCMR appropriately. Also set the PE bit in SMR to 1.
6. Set bits SDIR, SINV, and SMIF in SCMR appropriately. Then, the TXDn and RXDn pins are placed in the high impedance state.
7. Set the value corresponding to the bit rate in BRR.
8. Set the CKE[1:0] bits in SCR appropriately, and set bits TIE, RIE, TE, RE, and TEIE in SCR to 0 at the same time.  
When the CKE[1:0] bit is set to 1, the SCKn pin is allowed to output clock pulses.
9. Set the TIE, RIE, TE, and RE bits in SCR to 1. Setting the TE and RE bits to 1 simultaneously is prohibited except for self-diagnosis.

To change reception mode to transmission mode, first check that reception has completed, and then initialize the SCI. At the end of initialization, set TE = 1 and RE = 0. Reception completion can be verified by reading the RXI request, ORER, or PER flag in SSR.

To change transmission mode to reception mode, first check that transmission has completed, and then initialize the SCI. At the end of initialization, set TE = 0 and RE = 1. Transmission completion can be verified by reading the TEND flag in SSR.

### 32.6.6 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 re-transmitted, is different from that in normal serial communications interface mode. Figure 32.33 shows the data retransfer operation during transmission.

1. When an error signal from the receiver end is sampled after one-frame data has been transmitted, the ERS flag in SSR is set to 1. If the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Clear the ERS flag to 0 before the next parity bit is sampled.
2. For a frame in which an error signal is received, the TEND flag in SSR is not set. Data is retransferred from TDR to TSR 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 retransfer) has been completed, and the TEND flag is set. If the TIE bit in SCR is 1 at this time, a TXI interrupt request is generated. Writing transmit data to TDR starts transmission of the next data.

Figure 32.35 shows a sample flowchart of serial transmission. All the processing steps are automatically performed using a TXI interrupt request to activate the DTC or DMAC.

When the TEND flag in SSR is set to 1 in transmission, if the TIE bit in SCR is 1, a TXI interrupt request is generated. The DTC or DMAC is activated by a TXI interrupt request if the TXI interrupt request is specified as a source of DTC or DMAC activation beforehand, allowing transfer of transmit data. The TEND flag is automatically cleared to 0 when the DTC or DMAC transfers the data.

If an error occurs, the SCI automatically re-transmits the same data. During this retransmission, the TEND flag is kept to 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, since the ERS flag is not automatically cleared, set the RIE bit to 1 beforehand to enable an ERI interrupt request to be generated at error occurrence, and clear the ERS flag to 0.

When transmitting/receiving data using the DTC or DMAC, be sure to make settings to enable the DTC or DMAC before making SCI settings.

For DTC or DMAC settings, see section 18, DMA Controller (DMACA) and section 19, Data Transfer Controller (DTCa).

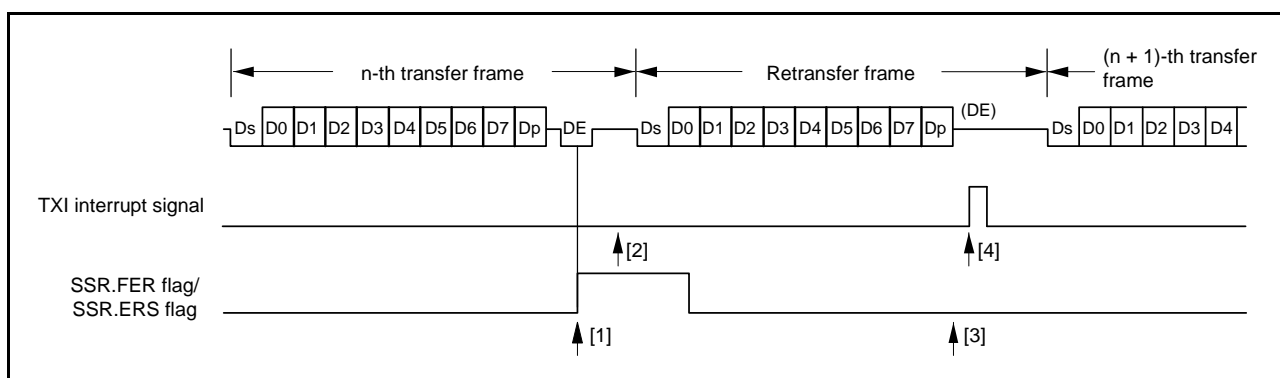
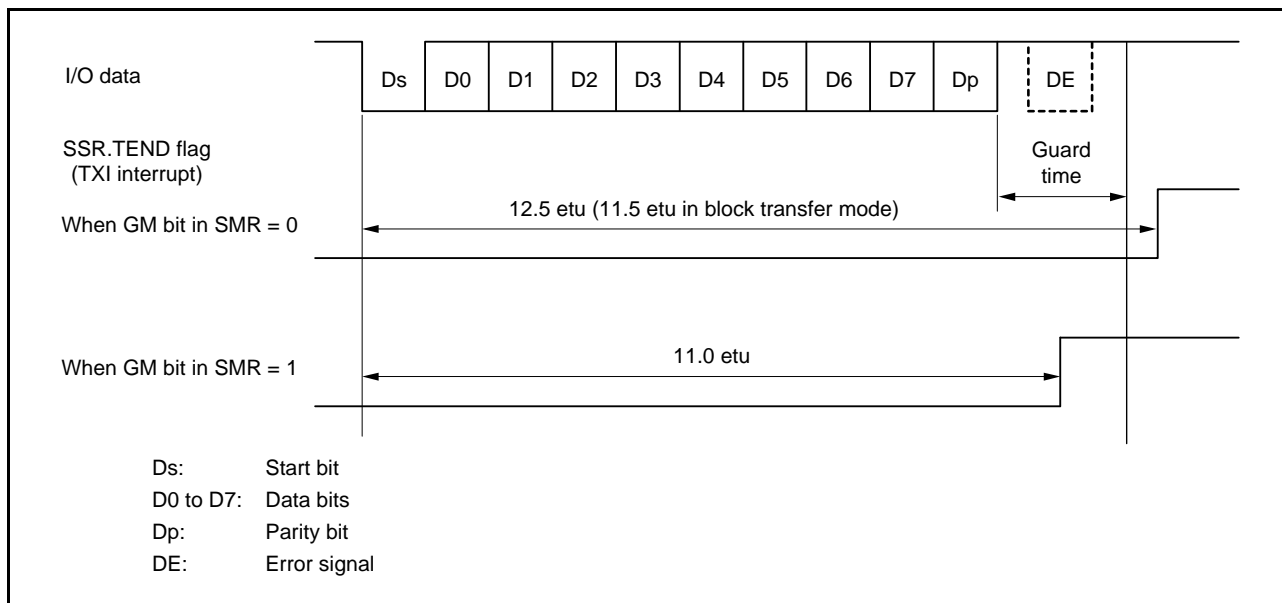


Figure 32.33 Data Retransfer Operation in SCI Transmission Mode



Note that the SSR.TEND flag is set in different timings depending on the GM bit setting in SMR. Figure 32.34 shows the TEND flag generation timing.



**Figure 32.34 SSR.TEND Flag Generation Timing during Transmission**

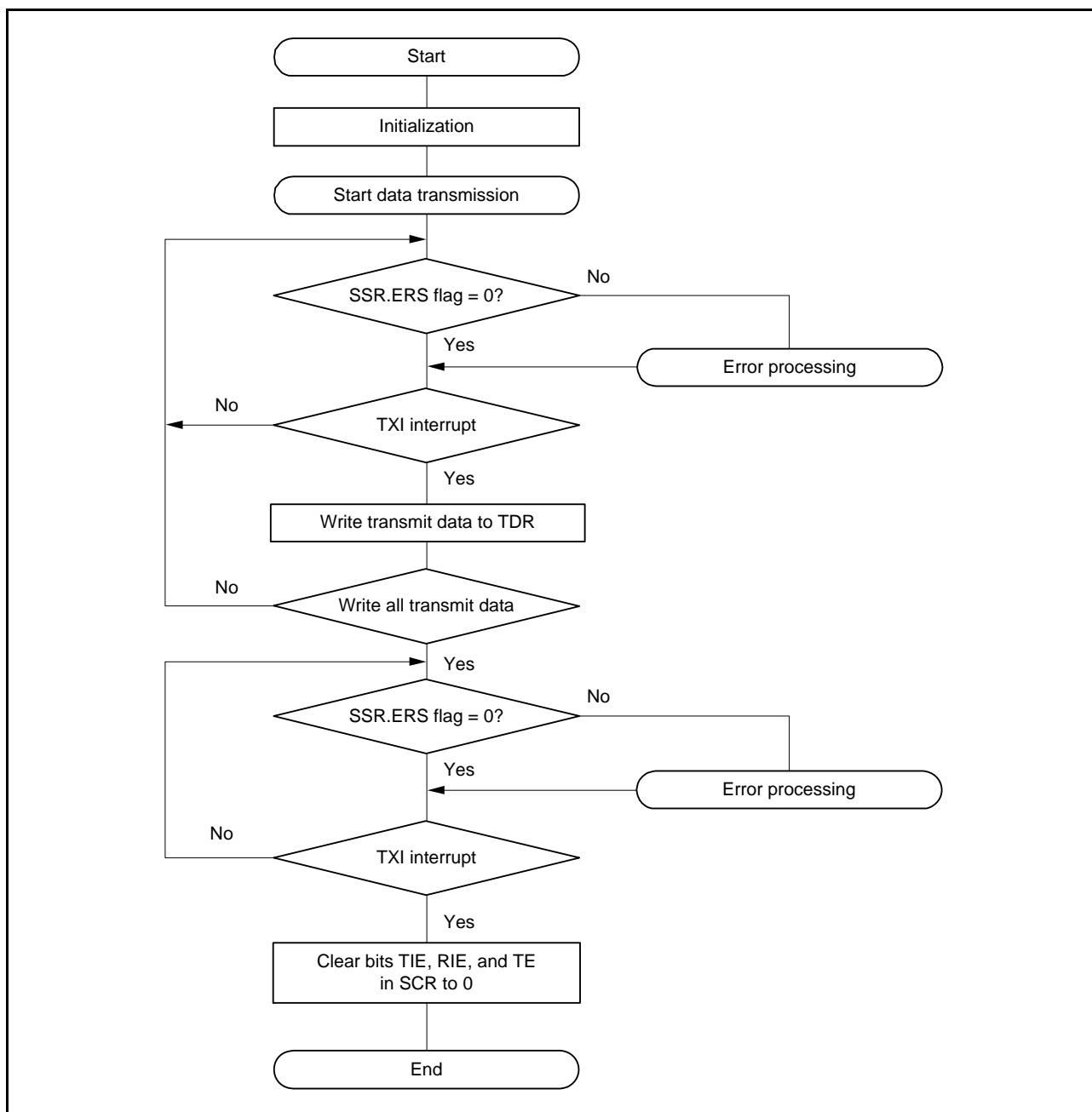


Figure 32.35 Sample Smart Card Interface Transmission Flowchart

### 32.6.7 Serial Data Reception (Except in Block Transfer Mode)

Serial data reception in smart card interface mode is similar to that in serial communications interface mode. Figure 32.36 shows the data retransfer operation in reception mode.

1. If a parity error is detected in receive data, the PER flag in SSR is set to 1. When the RIE bit in SCR is 1 at this time, an ERI interrupt request is generated. Clear the PER flag to 0 before the next parity bit is sampled.
2. For a frame in which a parity error is detected, no RXI interrupt is generated.
3. When no parity error is detected, the PER flag in SSR is not set to 1.
4. In this case, data is determined to have been received successfully. When the RIE bit in SCR is 1, an RXI interrupt request is generated.

Figure 32.37 shows a sample flowchart for serial data reception. All the processing steps are automatically performed using an RXI interrupt request to activate the DTC or DMAC.

In reception, setting the RIE bit to 1 allows an RXI interrupt request to be generated. The DTC or DMAC is activated by an RXI interrupt request if the RXI interrupt request is specified as a source of DTC or DMAC activation beforehand, allowing transfer of receive data.

If an error occurs during reception and either the ORER or PER flag in SSR 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 or DMAC is not activated and receive data is skipped. Therefore, the number of bytes of receive data specified in the DTC or DMAC is transferred. Even if a parity error occurs and the PER flag is set to 1 during reception, receive data is transferred to RDR, thus allowing the data to be read.

Note 1. For operations in block transfer mode, see section 32.3, Operation in Asynchronous Mode.

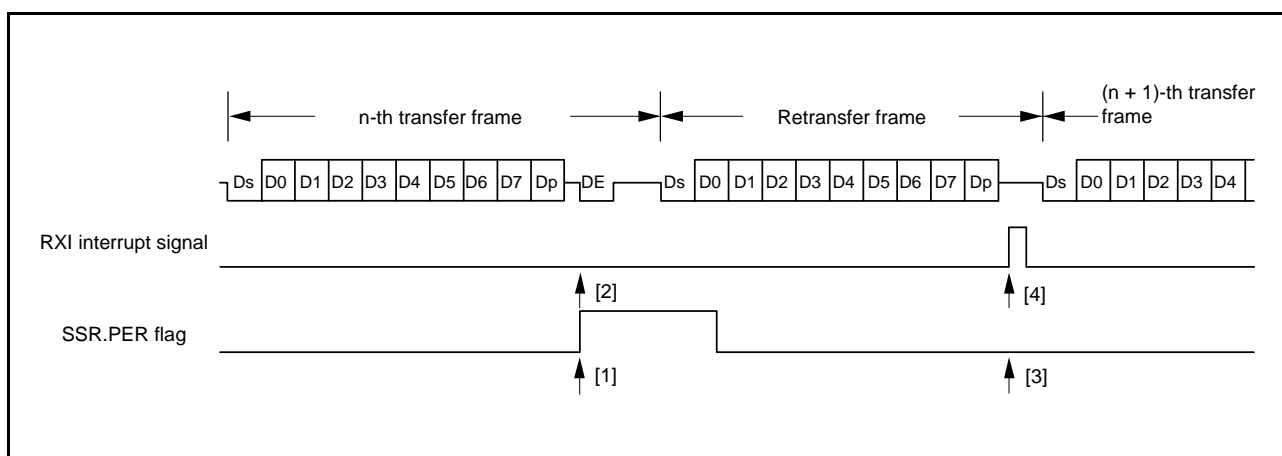


Figure 32.36 Data Retransfer Operation in SCI Reception Mode

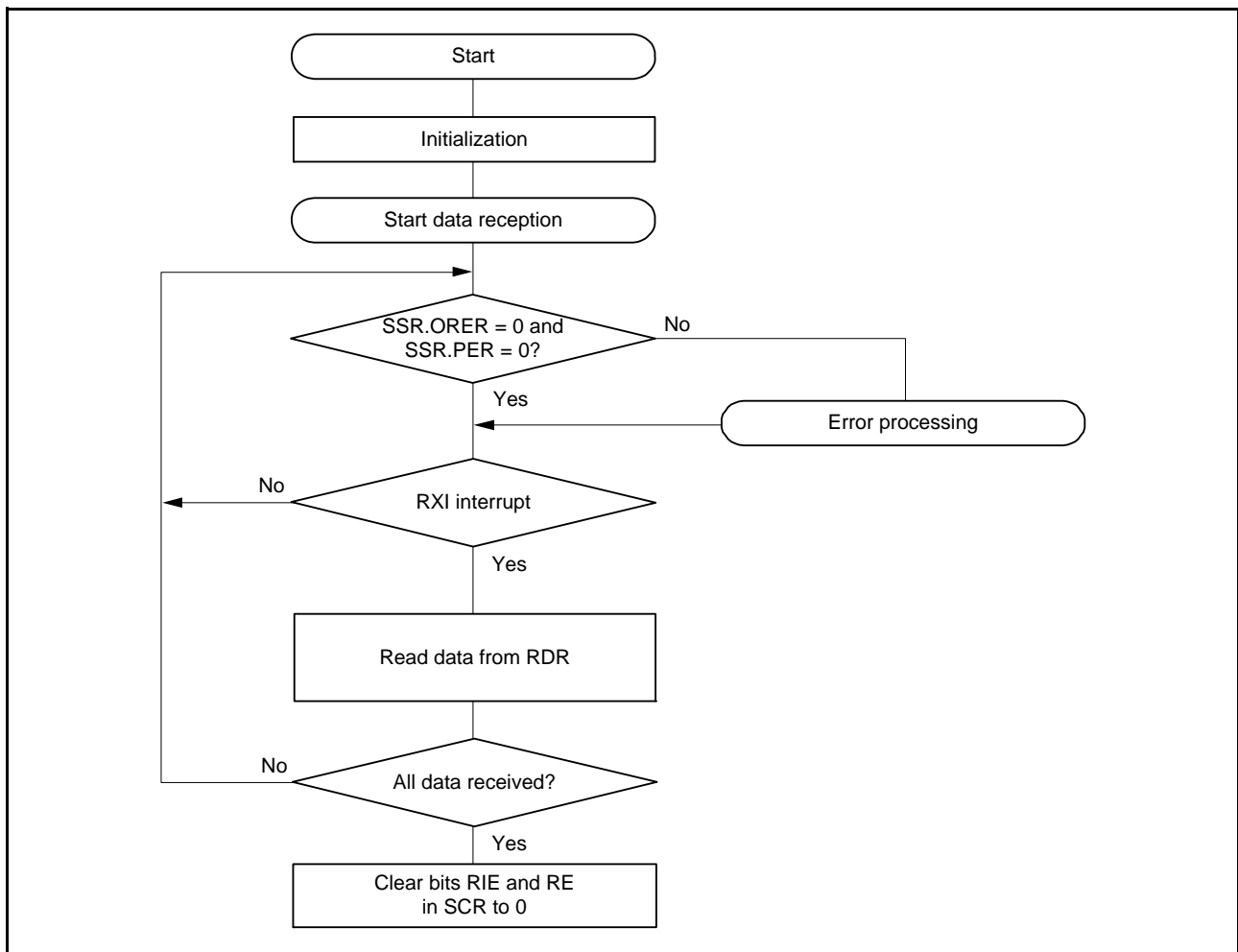


Figure 32.37 Sample Smart Card Interface Reception Flowchart

### 32.6.8 Clock Output Control

Clock output can be fixed using the CKE[1:0] bits in SCR when the GM bit in SMR is 1. Specifically, the minimum width of a clock pulse can be specified.

Figure 32.38 shows an example of clock output fixing timing when the CKE0 bit is controlled with GM = 1 and CKE1 = 0.

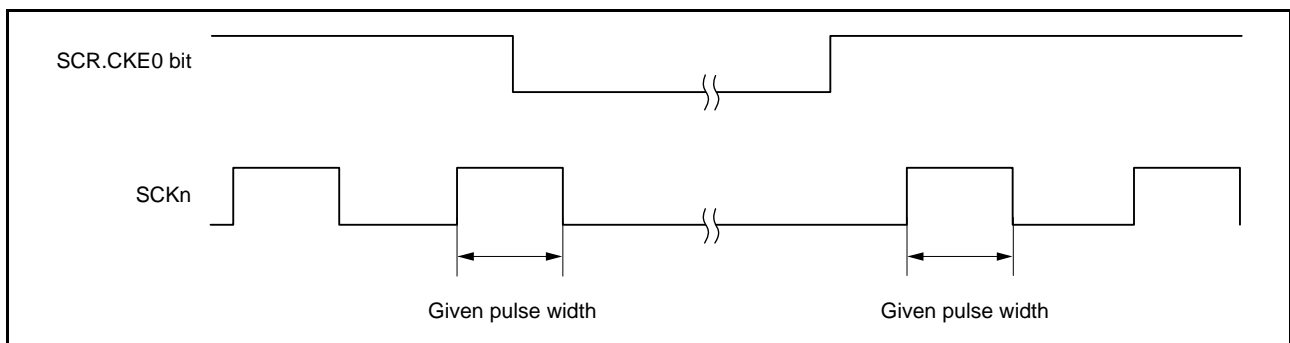


Figure 32.38 Clock Output Fixing Timing

At power-on and transitions to/from software standby mode, use the following procedure to secure the appropriate clock duty cycle.

**(1) At Power-On**

To secure the appropriate clock duty cycle simultaneously with power-on, use the following procedure.

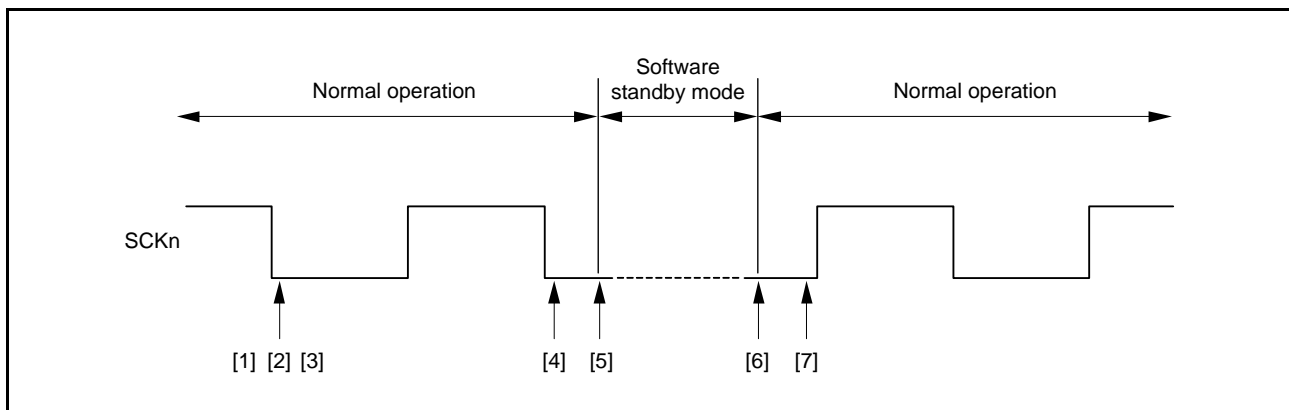
1. Initially, port input is enabled in the high-impedance state. To fix the potential level, use a pull-up or pull-down resistor.
2. Fix the SCKn pin to the specified output by setting the SCR.CKE[1] bit and I/O port functions.
3. Set SMR and SCMR to enable smart card interface mode.
4. Set the SCR.CKE[0] bit to 1 to start clock output.

**(2) At Mode Switching****(a) At transition from smart card interface mode to software standby mode**

1. Set I/O port functions to make the SCKn pin fixed with a desired output value in software standby mode.
2. Write 0 to the TE and RE bits in SCR to stop transmission/reception.  
Simultaneously, set the SCR.CKE[1] bit to the value for the output fixed state in software standby mode.
3. Write 0 to the SCR.CKE[0] bit to stop the clock.
4. Wait for one cycle of the serial clock. In the mean time, the clock output is fixed to the specified level with the duty cycle retained.
5. After switching the SCKn pin to the general I/O port function, make a transition to software standby mode.

**(b) Return from software standby mode to smart card interface mode**

6. Cancel software standby mode.
7. Set the SCR.CKE[0] bit to 1 to start clock output. A clock signal with the appropriate duty cycle is then generated.



**Figure 32.39** Clock Stop and Restart Procedure

### 32.7 Operation in Simple I<sup>2</sup>C Mode

Simple I<sup>2</sup>C bus master format is composed of eight 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 eight data bits in all frames are transmitted in order from the MSB.

The I<sup>2</sup>C format and timing of the I<sup>2</sup>C bus are shown in Figure 32.40 and Figure 32.41.

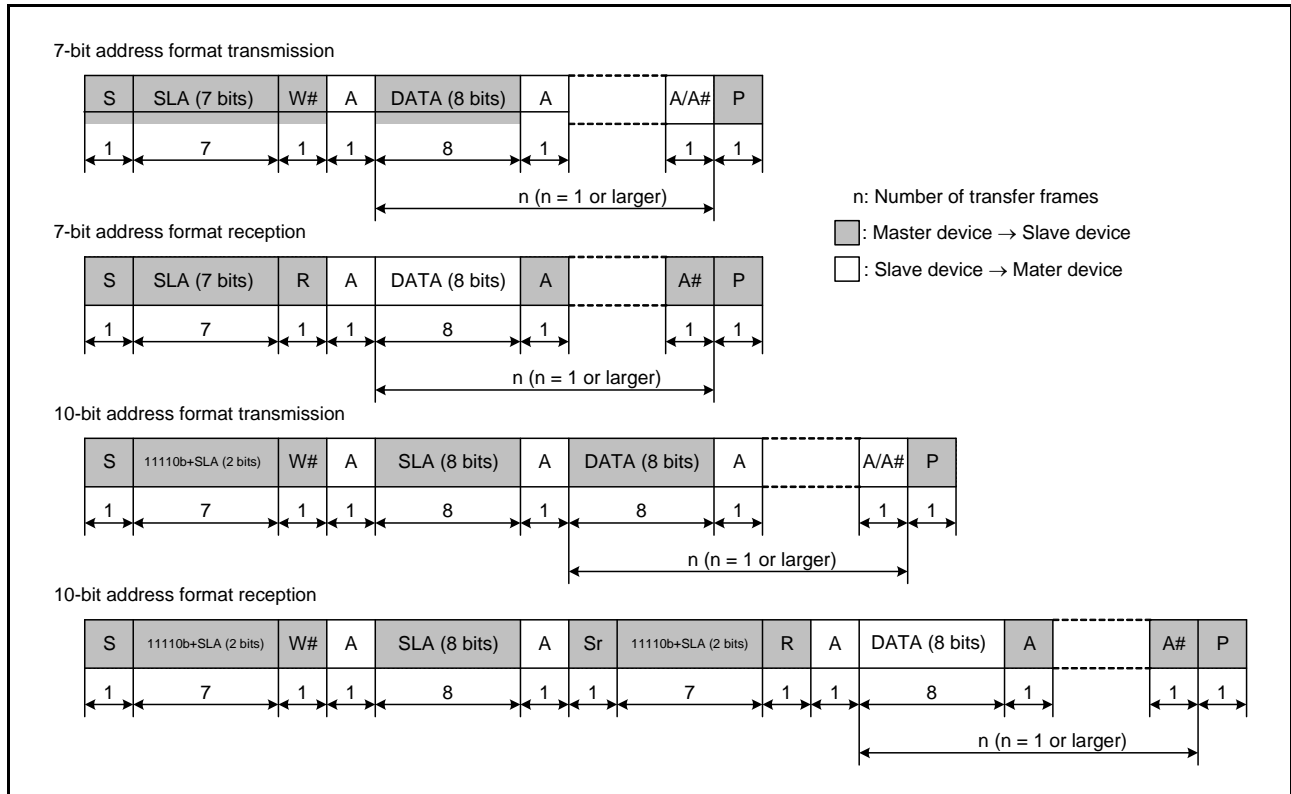


Figure 32.40 I<sup>2</sup>C Bus Master Format

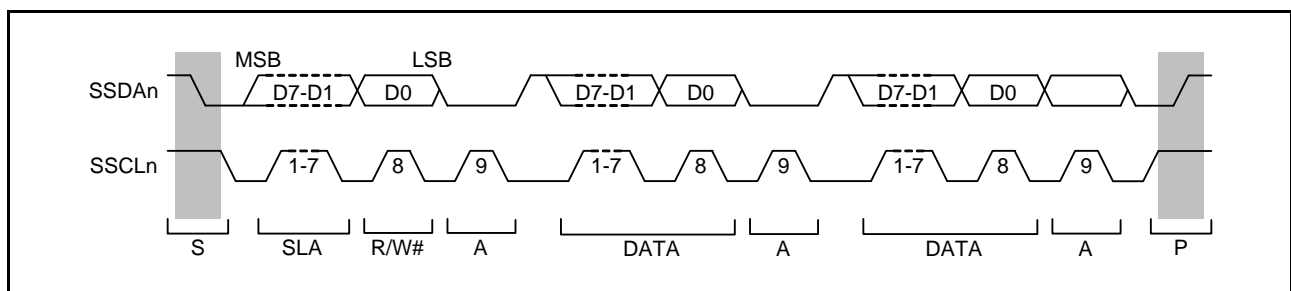


Figure 32.41 I<sup>2</sup>C Bus Timing (When SLA is 7 Bits)

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.

### 32.7.1 Generation of Start, Restart, and Stop Conditions

Writing 1 to the IICSTAREQ bit in SIMR3 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 IICSTAREQ bit in SIMR3 is cleared (to 0), and a start-condition generated interrupt is output.

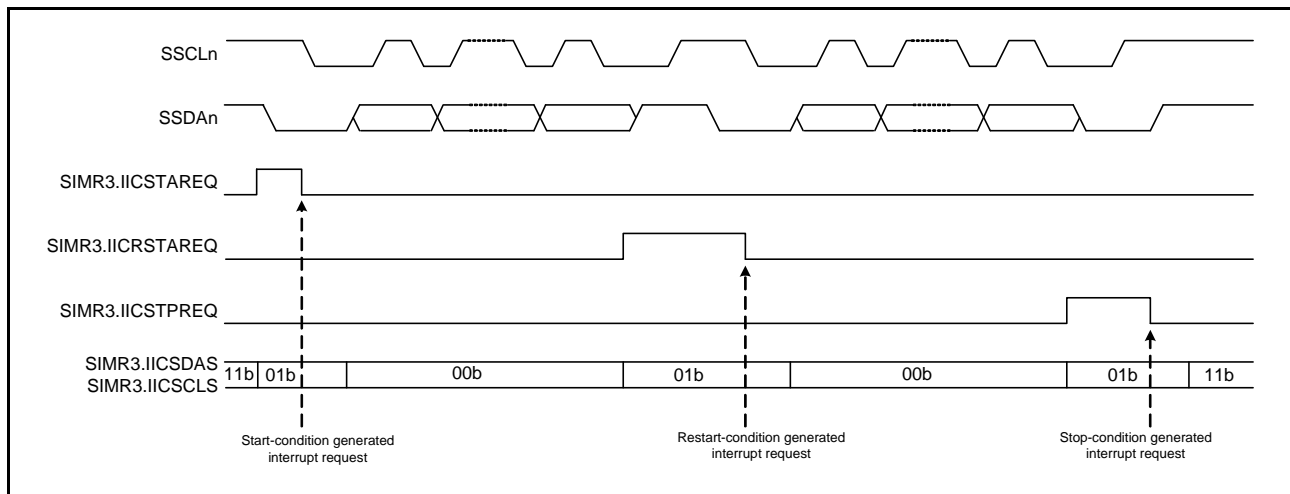
Writing 1 to the IICRSTAREQ bit in SIMR3 causes the generation of a start condition. The generation of a start condition proceeds through the following operations.

- 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 IICRSTAREQ bit in SIMR3 is cleared (to 0), and a restart-condition generated interrupt is output.

Writing 1 to the IICRSTPREQ bit in SIMR3 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 IICRSTPREQ bit in SIMR3 is cleared (to 0), and a stop-condition generated interrupt is output.

Figure 32.42 shows the timing of operations in the generation of start, restart, and stop conditions.



**Figure 32.42** Timing of Operations in the Generation of Start, Restart, and Stop Conditions

### 32.7.2 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 IICCSC bit in SIMR2 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 IICCSC bit in SIMR2 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 input, 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 IICCSC bit in SIMR2 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 IICCSC bit in SIMR2 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 32.43 shows an example of operations to synchronize the clocks.



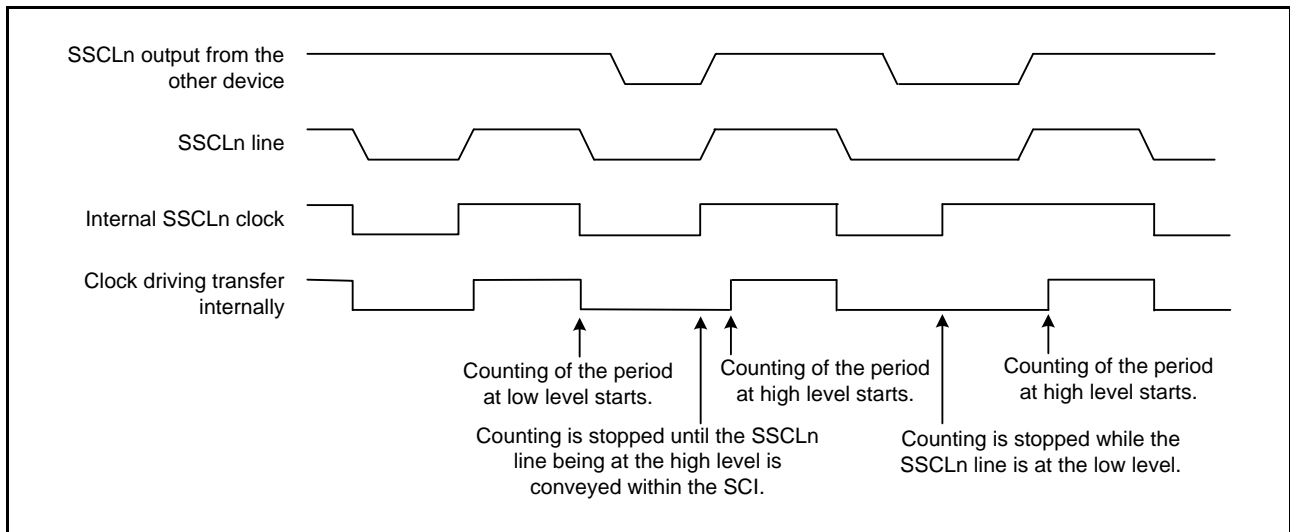


Figure 32.43 Example of Operations for Clock Synchronization

### 32.7.3 SSDA Output Delay

The IICDL[4:0] bits in SIMR1 can be used to set a delay for output on the SSDA 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 CKS[1:0] bits in SMR). 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 SSDA pin are for times greater than the time output on the SSCLn pin takes to fall (300 ns for I<sup>2</sup>C in normal mode and fast mode).

Figure 32.44 shows the timing of delays in output on the SSDA pin.

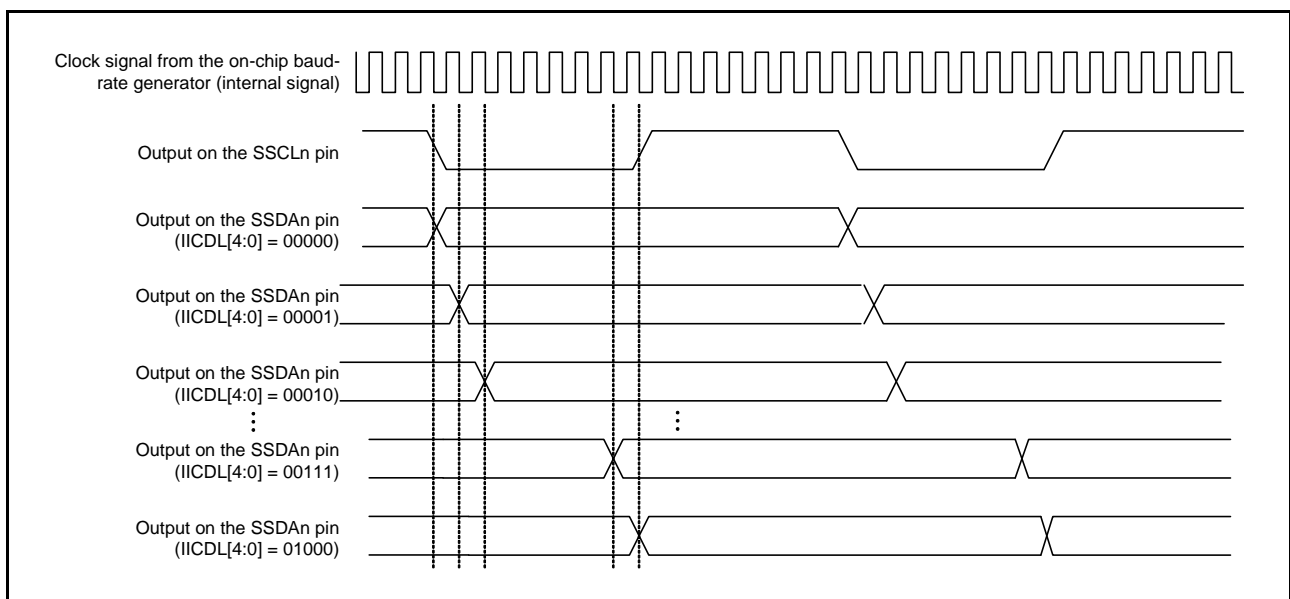


Figure 32.44 Timing of Delays in SSDA Output

### 32.7.4 SCI Initialization (Simple I<sup>2</sup>C Mode)

Before transferring data, write the initial value (00h) to SCR and initialize the interface in accordance with the flowchart shown as Figure 32.45.

When changing the operating mode, transfer format, and so on, be sure to set SCR to its initial value before proceeding with the changes.

In simple I<sup>2</sup>C mode, the open drain setting for the communication ports should be made on the port side.

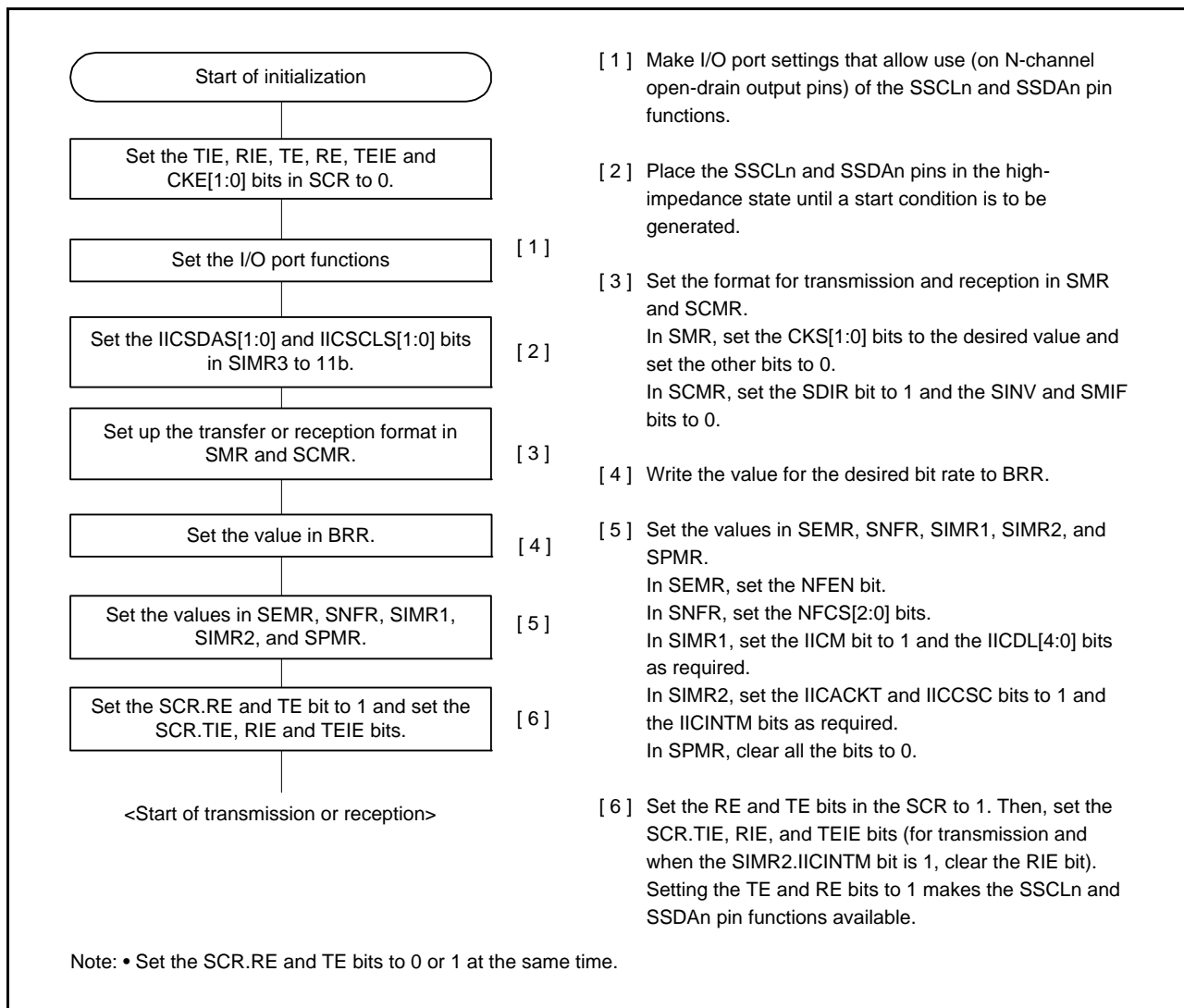


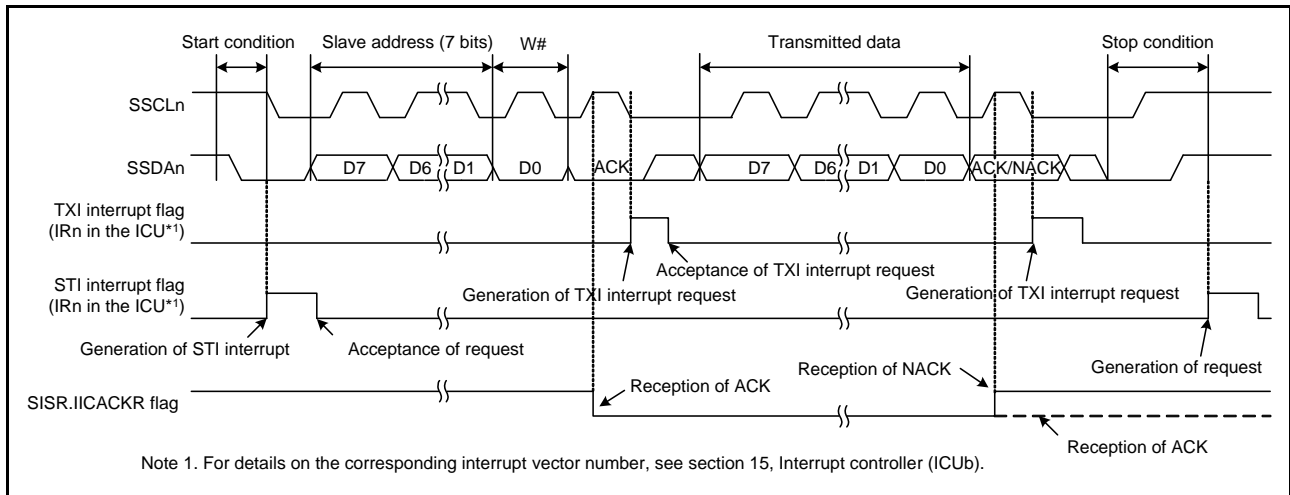
Figure 32.45 Example of the Flow of SCI Initialization (for Simple I<sup>2</sup>C Mode)

### 32.7.5 Operation in Master Transmission (Simple I<sup>2</sup>C Mode)

Figure 32.46 and Figure 32.47 show examples of operations in master transmission and Figure 32.48 is a flowchart showing the procedure for data transmission. The value of the SIMR2.IICINTM bit is assumed to be 1 (transmission and reception interrupts are in use) and the value of the SCR.RIE bit is assumed to be 0 (disabling reception interrupt requests). See Table 32.28 for more information on the STI interrupt.

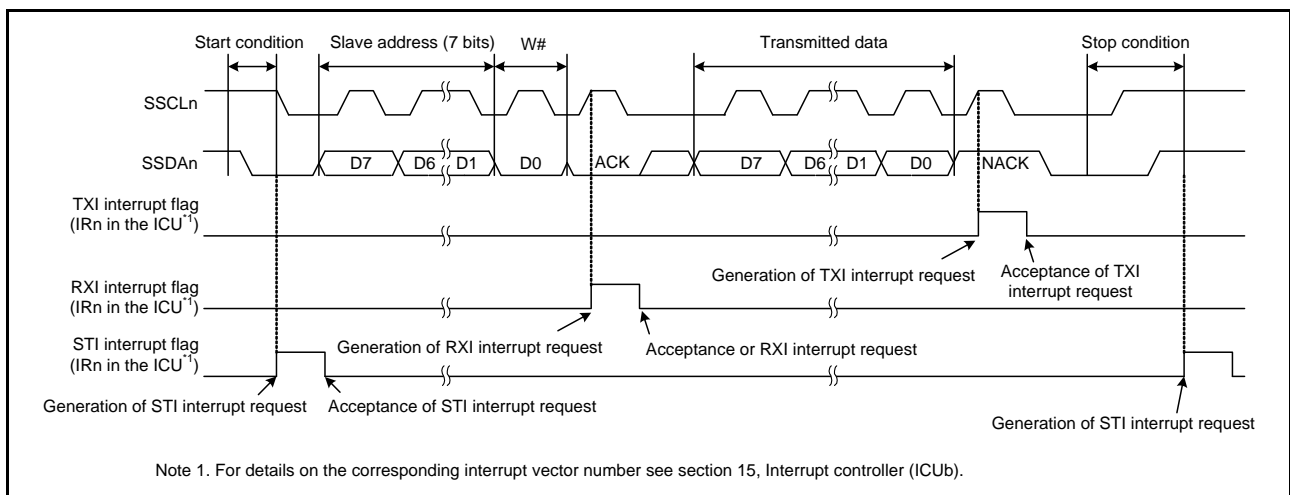
When 10-bit slave addresses are in use, steps [3] and [4] in Figure 32.48 are repeated twice.

In simple I<sup>2</sup>C mode, the transmit end interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock-synchronous transmission.



**Figure 32.46 Example of Operations for Master Transmission in I<sup>2</sup>C Bus Mode (with Seven-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)**

When the SIMR2.IICINTM bit is set to 0 (ACK and NACK interrupts are used) during master transmission, the DTC or DMAC 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 re-transmission, is performed by the NACK interrupt as the trigger.



**Figure 32.47 Example 2 of Operations for Master Transmission in Simple I<sup>2</sup>C Bus Mode (with Seven-Bit Slave Addresses, ACK Interrupts, and NACK Interrupts in Use)**

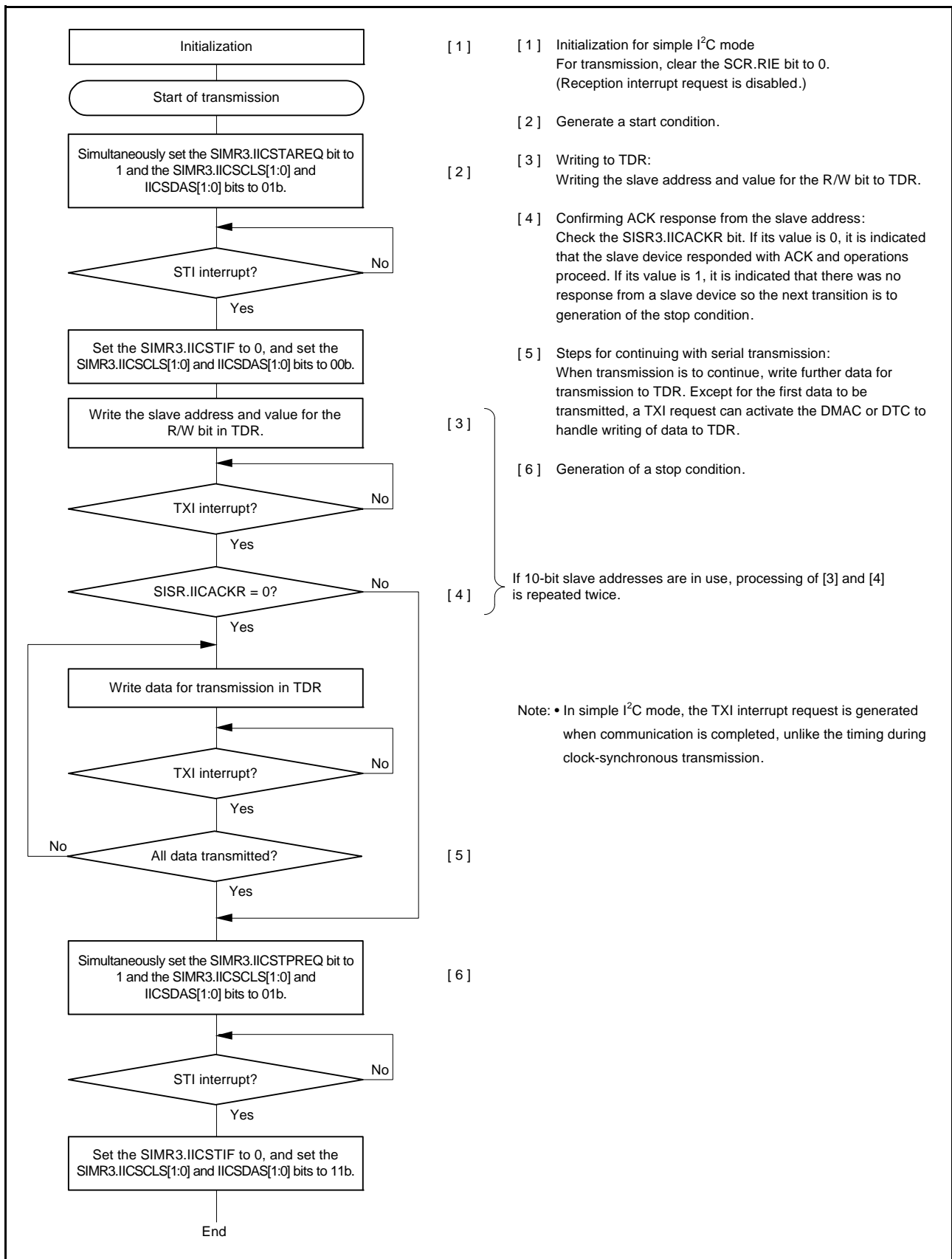


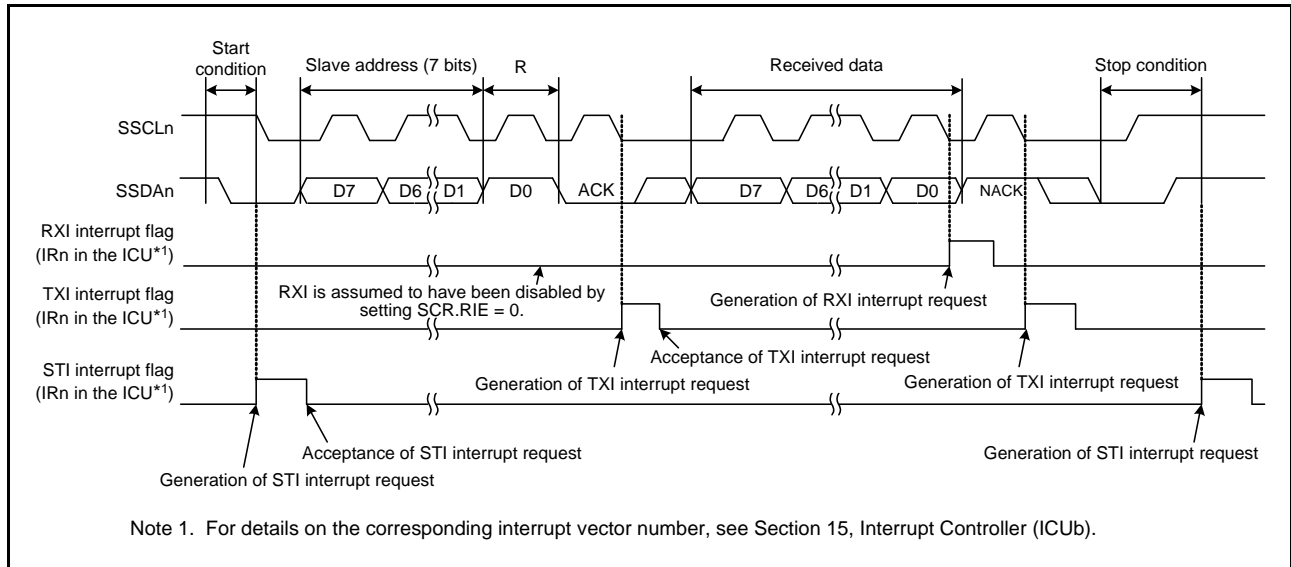
Figure 32.48 Example of the Procedure for Master Transmission Operations in Simple I<sup>2</sup>C Mode (with Transmission Interrupts and Reception Interrupts in Use)

### 32.7.6 Master Reception (Simple I<sup>2</sup>C Mode)

Figure 32.49 shows an example of operations in simple I<sup>2</sup>C mode master reception and Figure 32.50 is a flowchart showing the procedure for master reception.

The value of the SIMR2.IICINTM bit is assumed to be 1 (transmission and reception interrupts are in use).

In simple I<sup>2</sup>C mode, the transmit end interrupt (TXI) is generated when communication of one frame is completed, unlike the TXI interrupt request generation timing during clock-synchronous transmission.



**Figure 32.49 Example of Operations for Master Reception in I<sup>2</sup>C Bus Mode (with Seven-Bit Slave Addresses, Transmission Interrupts, and Reception Interrupts in Use)**

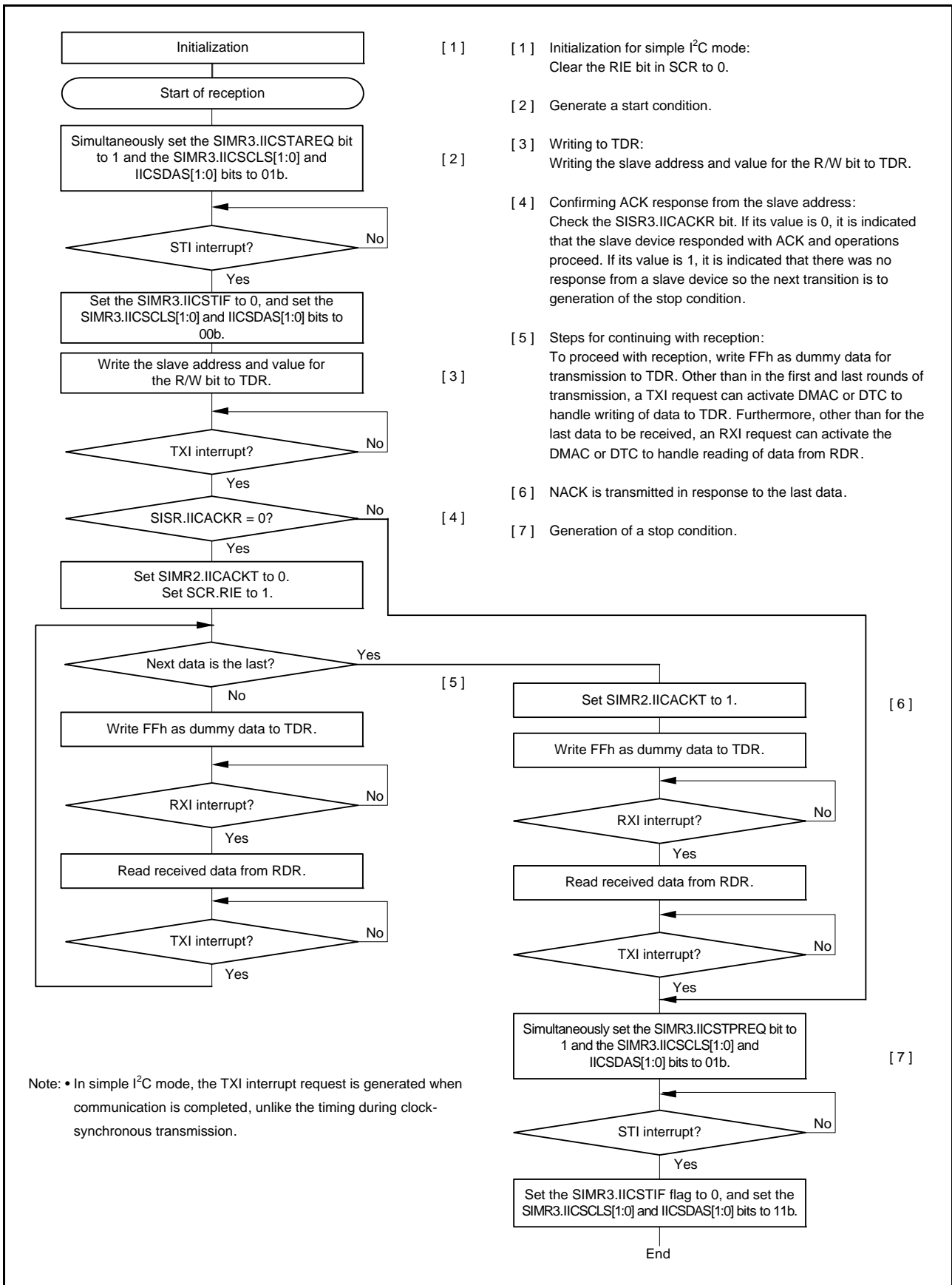


Figure 32.50 Example of the Procedure for Master Reception Operations in Simple I<sup>2</sup>C Mode (with Transmission Interrupts and Reception Interrupts in Use)

### 32.8 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, SIMR.IICM = 0, SMR.CM = 1) plus setting the SSE bit in the SECR 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 SSE bit in the SPMR to 0 in such cases.

Figure 32.51 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 eight 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 reception and transmission sections are independent of each other within the SCI module, full-duplex communications are possible, with a common clock signal. Furthermore, since both sections have a buffered structure, writing of further data for transmission while transmission is in progress and reading of previously received data while reception is in progress are both possible. Continuous transfer is thus possible.

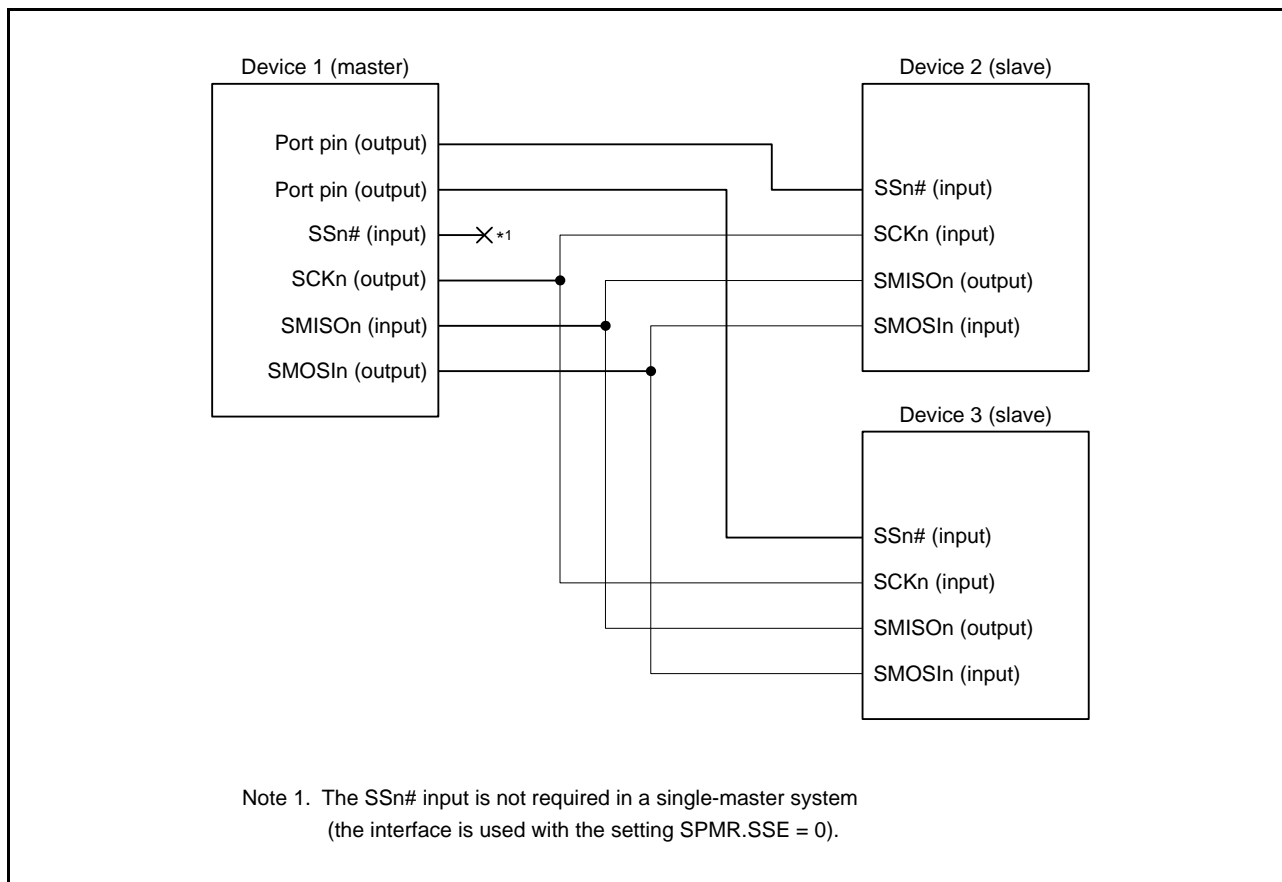


Figure 32.51 Example of Connections via a Simple SPI Mode (In Single Master Mode, SPMR.SSE Bit = 0)

### 32.8.1 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] = 00 or 01 and SPMR.MSS = 0) or slave (SCR.CKE[1:0] = 10 or 11 and SPMR.MSS = 1).

Table 32.24 lists the states of pins according to the mode and the level on the SSn# pin.

**Table 32.24 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 can proceed)	Input for received data (but disabled)	High impedance	Clock input (but disabled)
	Low level (transfer cannot 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 transmission is disabled (SCR.TE bit = 0).

Note 3. The SCKn pin output is in the high-impedance state when transmission is disabled (SCR.TE and RE bits = 00) in a multi-master configuration (SPMR.SSE = 1).

### 32.8.2 SS Function in Master Mode

Setting the CKE[1:0] bits in the SCR to 00 and the MSS bit in the SPMR 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 output pins 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.

### 32.8.3 SS Function in Slave Mode

Setting the CKE[1:0] bits in the SCR to 10 and the MSS bit in the SPMR to 1 selects slave operation. When the level on the SSn# pin is high, the RxDn 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 RxDn 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.



### 32.8.4 Relationship between Clock and Transmit/Receive Data

The SPMR.CKPOL and CKPH bits 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 32.52. The relation is the same for both master and slave operation. This is the same as when the level on the SSn# pin is high. The SSn# pin can be used for another purpose. For details, see section 32.8.2, SS Function in Master Mode.

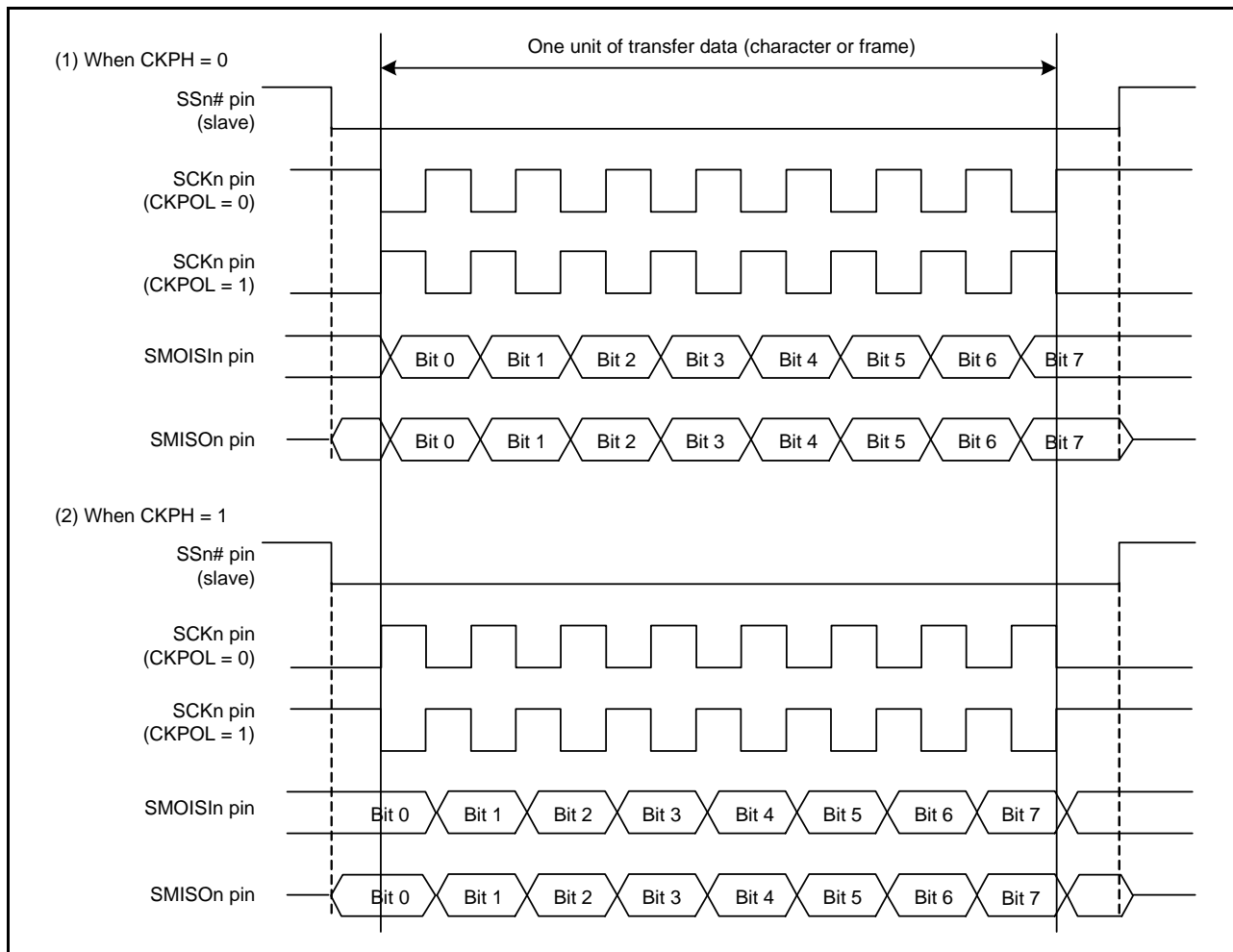


Figure 32.52 Relation between Clock Signal and Transmit/Receive Data in Simple SPI Mode

### 32.8.5 SCI Initialization (Simple SPI Mode)

The procedure is the same as for initialization in clock-synchronous mode (Figure 32.21, Sample SCI Initialization Flowchart). The CKPOL and CKPH bits in the SPMR 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.

Note that the ORER, FER, and PER flags, as well as the RDR, are not initialized even when the RE bit is set to 0.

Note that changing the value of the TE bit from 1 to 0 or from 0 to 1 will lead to the generation of a transmission interrupt (TXI) if the value of the TIE bit in the SCR is 1 at the time.

### 32.8.6 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.

## 32.9 Extended Serial Mode Control Section: Description of Operation

### 32.9.1 Serial Transfer Protocol

In conjunction with an SCIc module, the extended serial mode control section of an SCId module can realize the serial transfer protocol composed of Start Frames and Information Frames that is shown in Figure 32.53.

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.

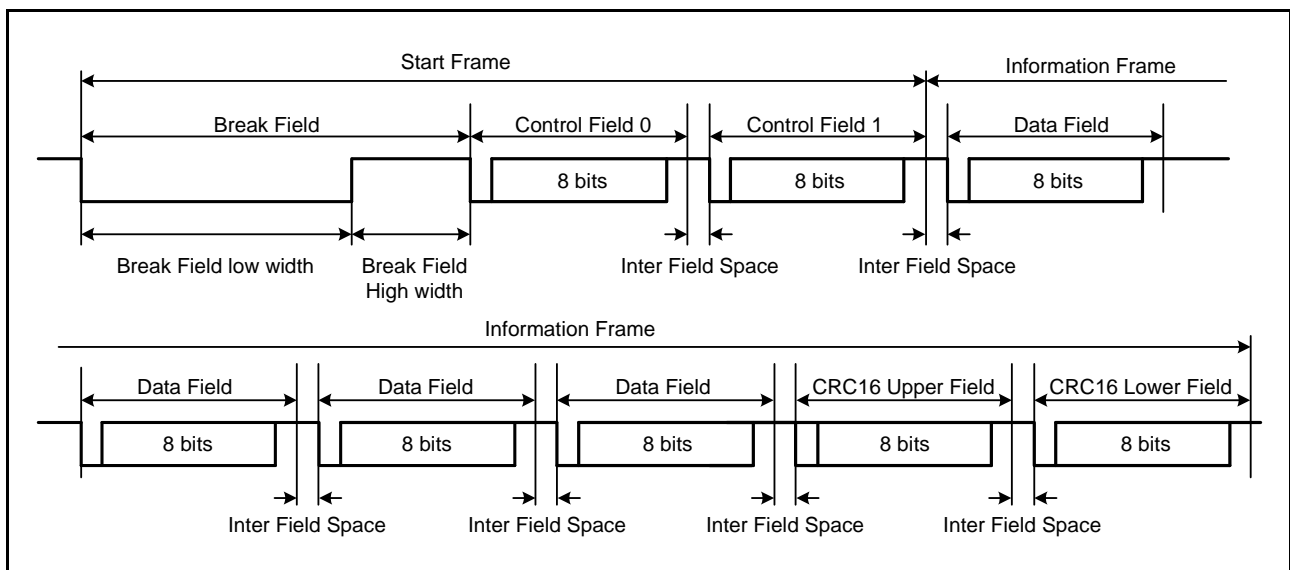


Figure 32.53 Protocol for Serial Transfer by the Extended Serial Mode Control Section

### 32.9.2 Transmitting a Start Frame

Figure 32.54 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 32.55 and Figure 32.56 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 TCST bit in TCR starts counting by the timer, and the low level will be output from the TXDX12 pin over the period corresponding to the TCNT and TPRE settings.
- (2) The output on the TXDX12 pin is inverted when the timer counter underflows, and the BPDF bit in STR is set to 1. An SCIX0 interrupt is also generated if the value of the BFDIE bit in ICR is 1.
- (3) Writing 0 to the TCST bit in TCR stops counting by the timer, and SCI2 is used to send the data for Control Field 0. After the Break Field low width output, stop counting before the next underflow occurs.
- (4) Once the data for Control Field 0 have been transmitted, SCI2 is used to send the data for Control Field 1.
- (5) Once the data for Control Field 1 have been transmitted, SCI2 is used to send an Information Frame.

Omit the Break Field and Control Field 0 to suit the structure of the Start Frame.

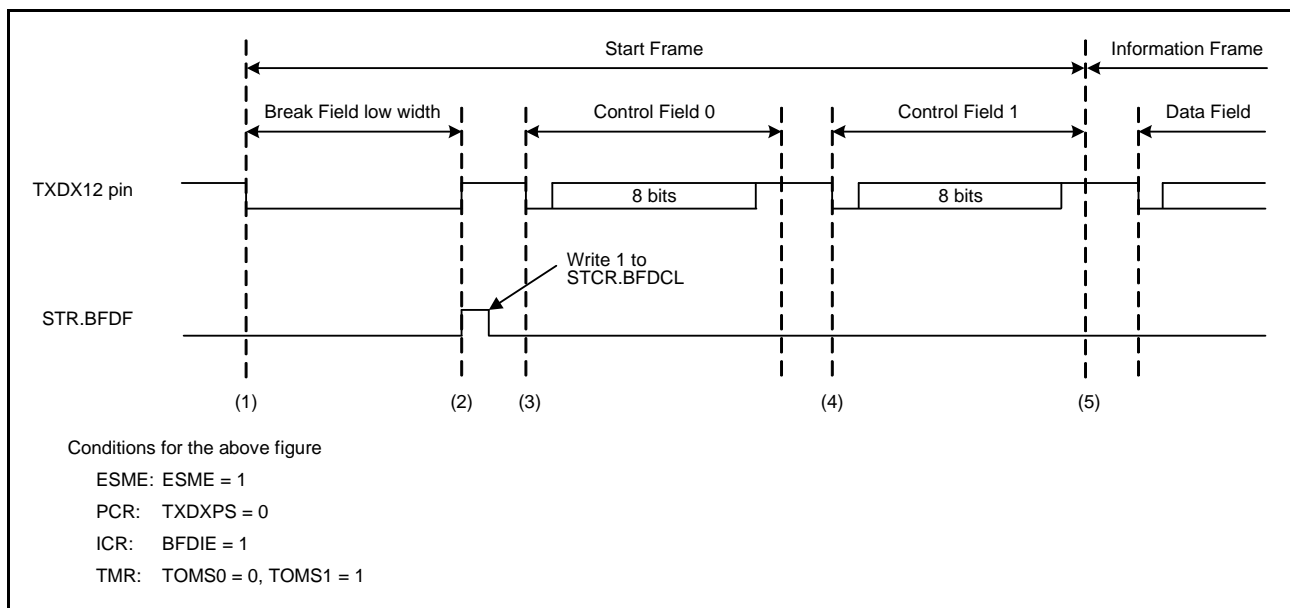


Figure 32.54 Example of Operations at the Time of Start-Frame Transmission

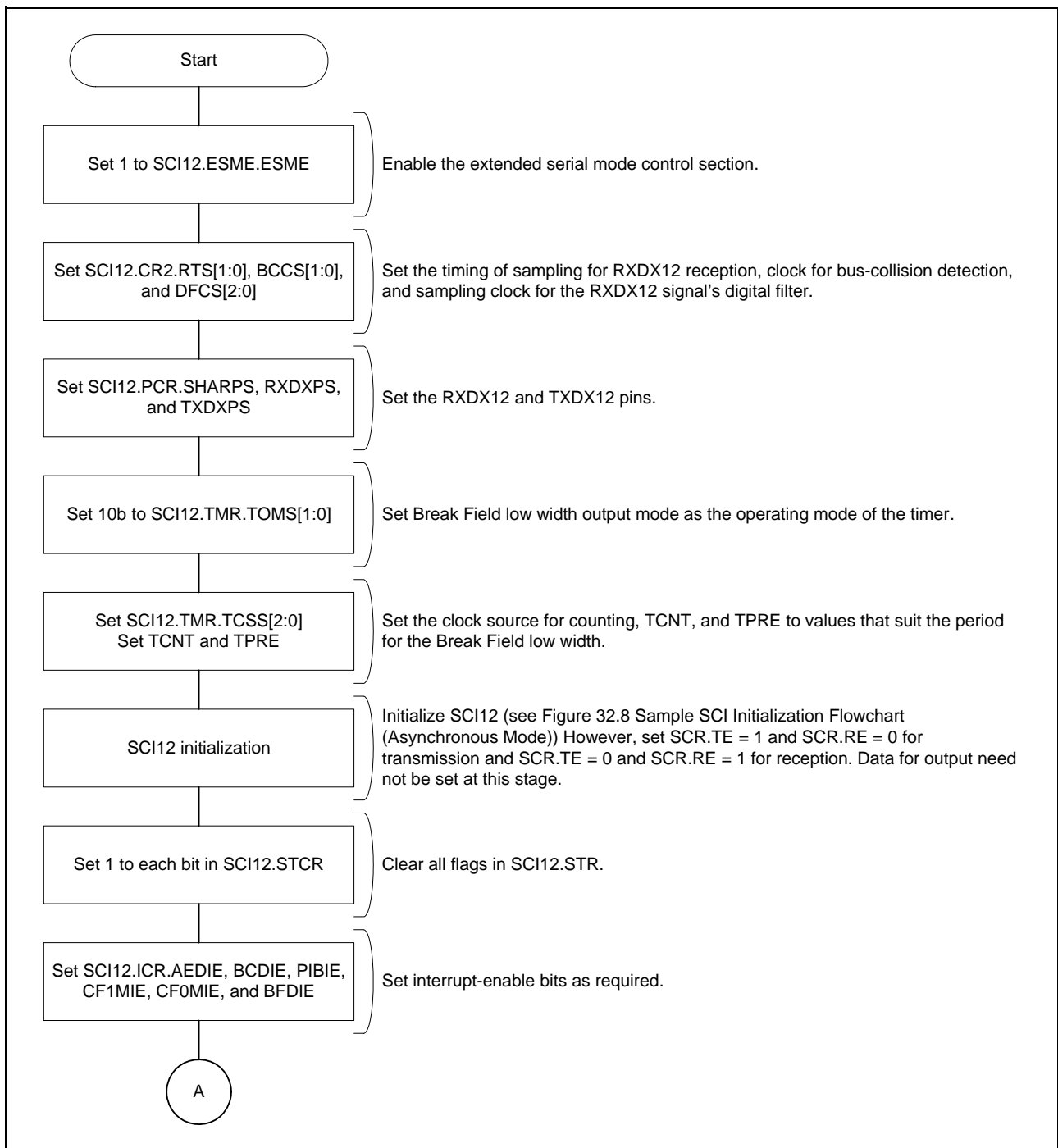


Figure 32.55 Sample Flowchart for Transmission of a Start Frame (1)

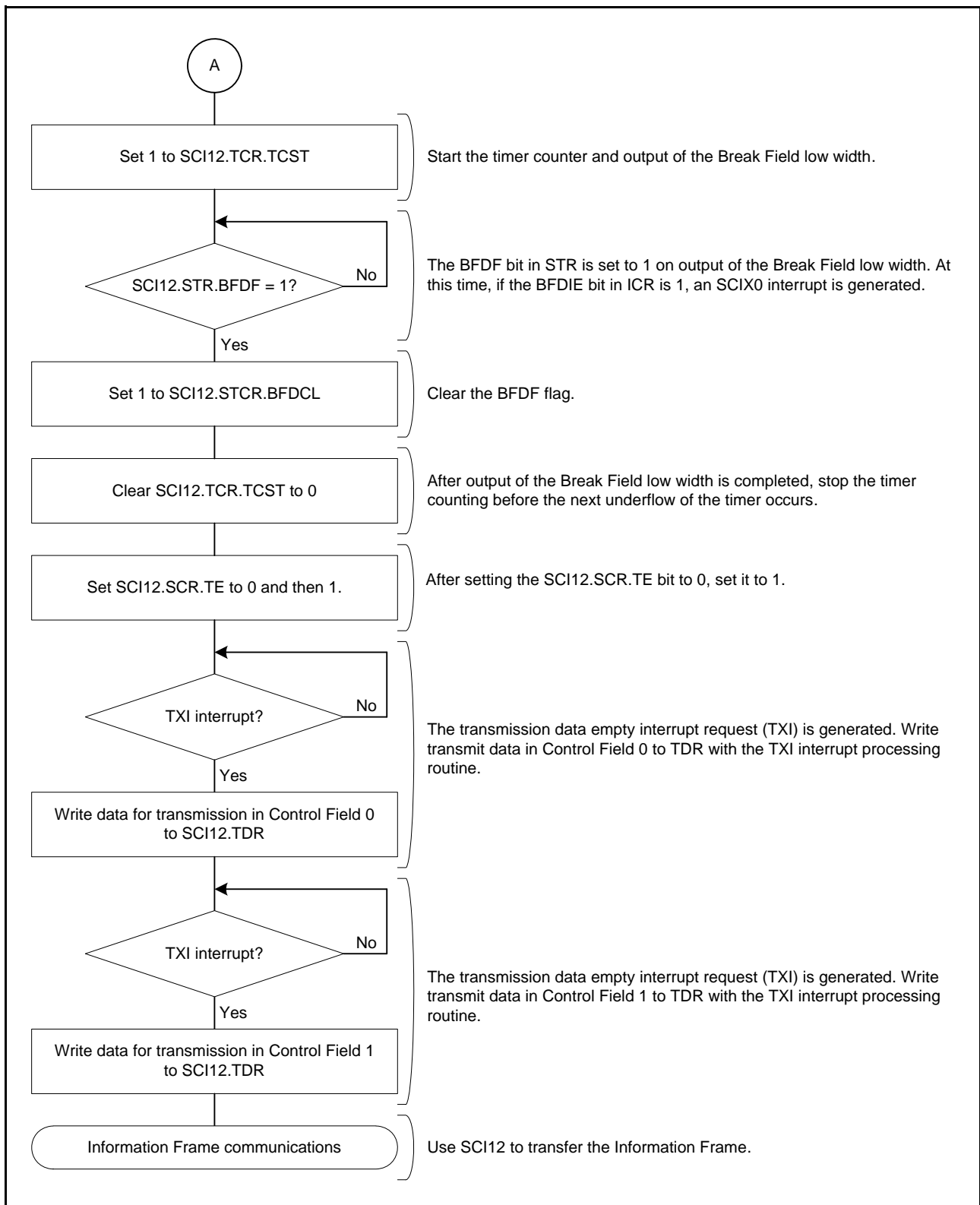


Figure 32.56 Sample Flowchart for Transmission of a Start Frame (2)

### 32.9.3 Receiving a Start Frame

The extended serial mode control section is capable of receiving Start Frames with the structures listed in Table 32.25.

**Table 32.25 Structures of Start Frames**

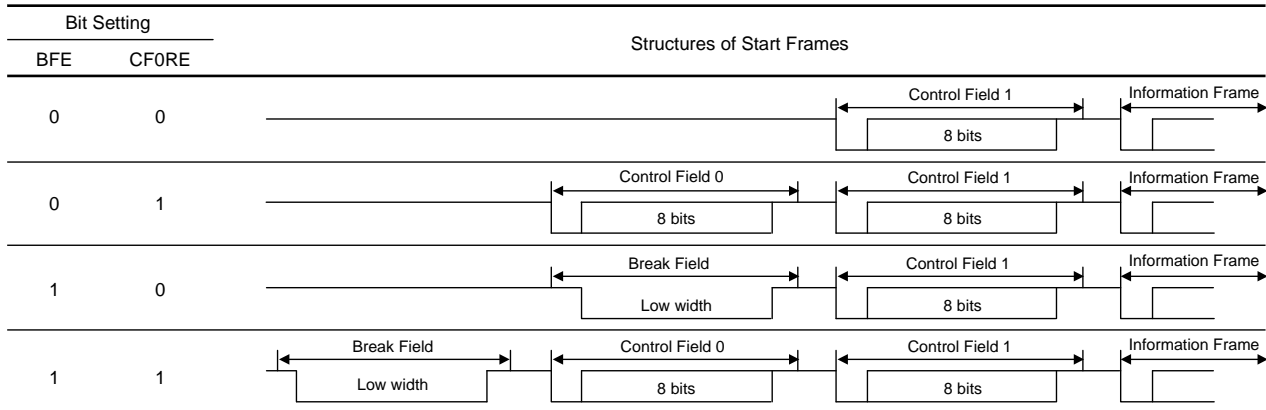


Figure 32.57 shows an example of operations to receive a Start Frame, which is composed of the Break Field low width, Control Field 0, and Control Field 1. Figure 32.58 and Figure 32.59 are flowcharts for the reception of a Start Frame, and Figure 32.60 is a state transition diagram for the extended serial mode control section.

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 SDST bit in CR3 enables detection of the Break Field low width. RXDX12 input to the SCI12 module is disabled at this time.
- (2) Low-level input on the RXDX12 pin continuing over a period longer than that corresponding to the settings of TCNT and TPRE is detected as the Break Field low width. At this time, the BDFD bit in STR is set to 1. An SCIX0 interrupt is also generated if the value of the BFDIE bit in ICR is 1.
- (3) When the input from the RXDX12 pin goes high after the Break Field low width, the RXDSF bit in CR0 becomes zero and reception of Control Field 0 by the SCI12 module starts.
- (4) If the data received in Control Field 0 match the data set in CF0DR, the CF0MF bit in STR is set to 1. An SCIX1 interrupt is also generated if the value of the CF0MIE bit in ICR is 1. Reception of Control Field 1 by the SCI12 module starts after that. If the data received in Control Field 0 do not match the data set in CF0DR, 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 PCF1DR and SCF1DR, the CF1MF bit in STR is set to 1. An SCIX1 interrupt is also generated if the value of the CF1MIE bit in ICR is 1. Transfer of the Information Frame by the SCI12 module starts after that. If the data received in Control Field 1 do not match the data set in either or both of 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.

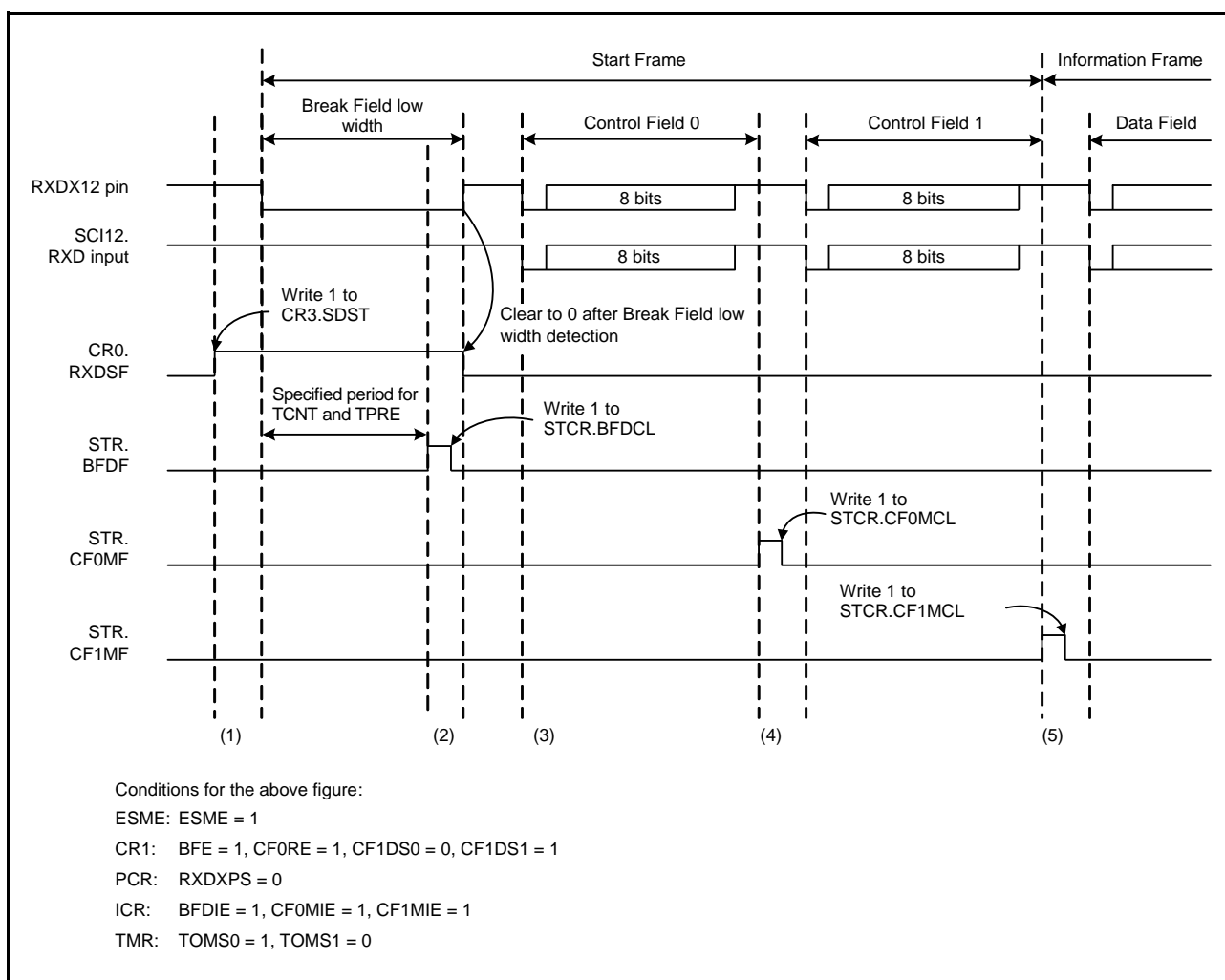


Figure 32.57 Example of Operations at the Time of Start-Frame Reception

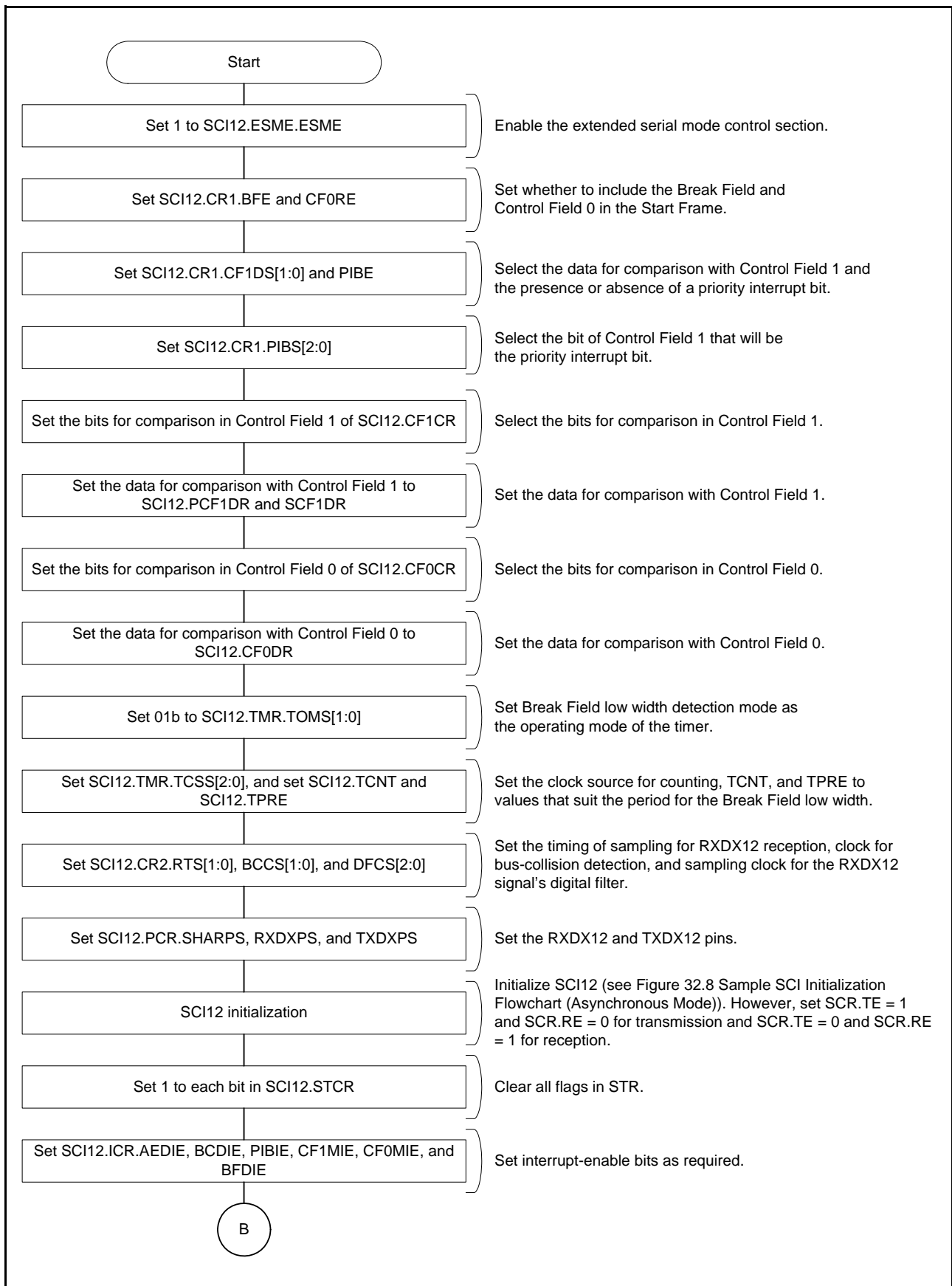


Figure 32.58 Sample Flowchart for Reception of a Start Frame (1)



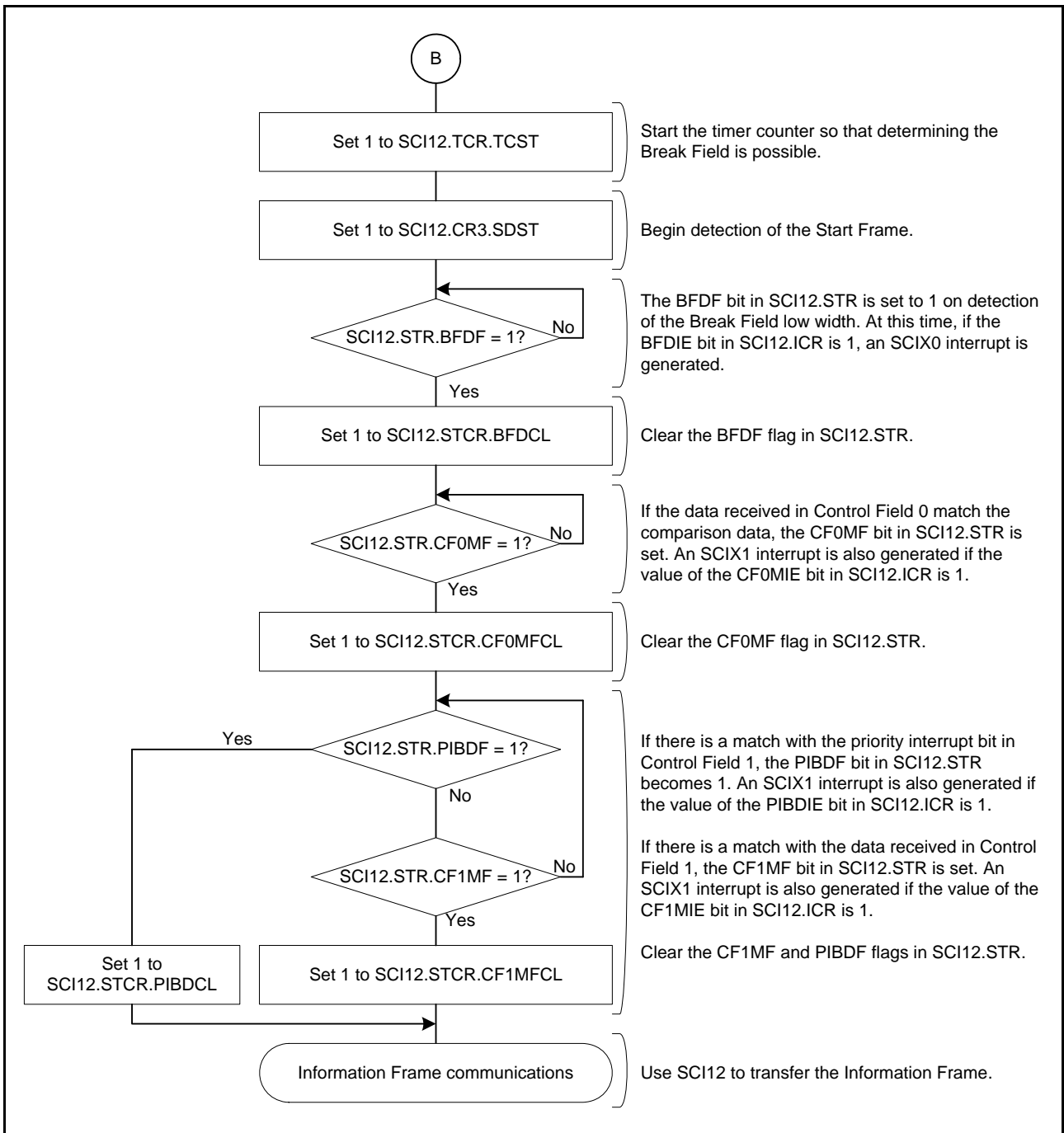


Figure 32.59 Sample Flowchart for Reception of a Start Frame (2)

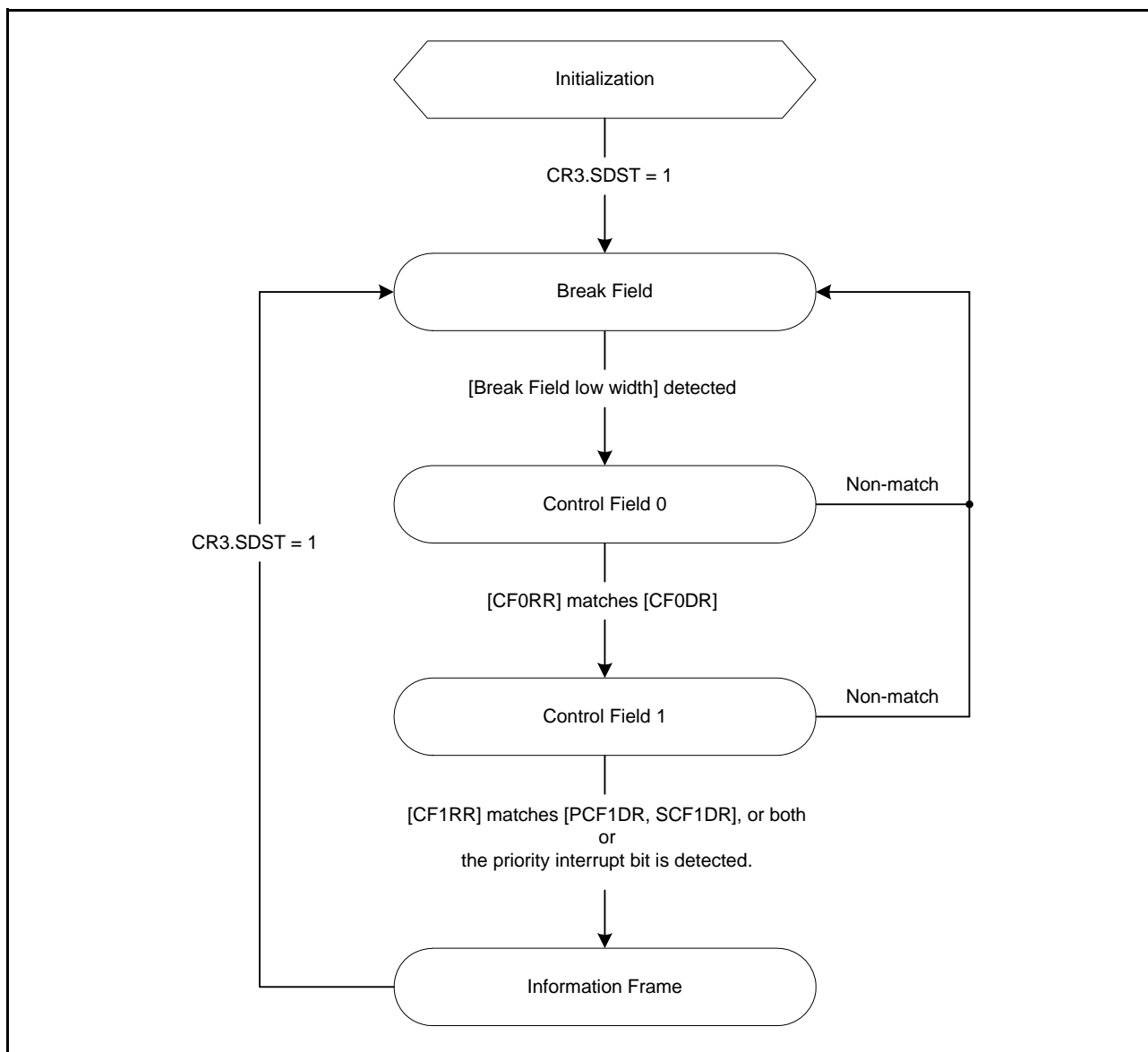


Figure 32.60 Diagram of State Transitions at the Time of Start-Frame Reception

### 32.9.3.1 Priority Interrupt Bit

Figure 32.61 shows an example of operation in Start-Frame reception where a priority interrupt bit is in use. Setting the PIBE bit in CR1 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 32.57, for Start-Frame reception.

- (5) If the value of the bit selected by the PIBS[2:0] bits in CR1 matches the corresponding bit in PCF1DR, the PIBDF bit in STR is set to 1. An SCIX1 interrupt is also generated if the value of the PIBDIE bit in ICR is 1. Transfer of the Information Frame by the SCI2 module starts after that. If the data received in Control Field 1 do not match the data set in either or both of PCF1DR and SCF1DR and the priority interrupt bit is not detected, a transition to the state prior to Break Field low width detection proceeds.

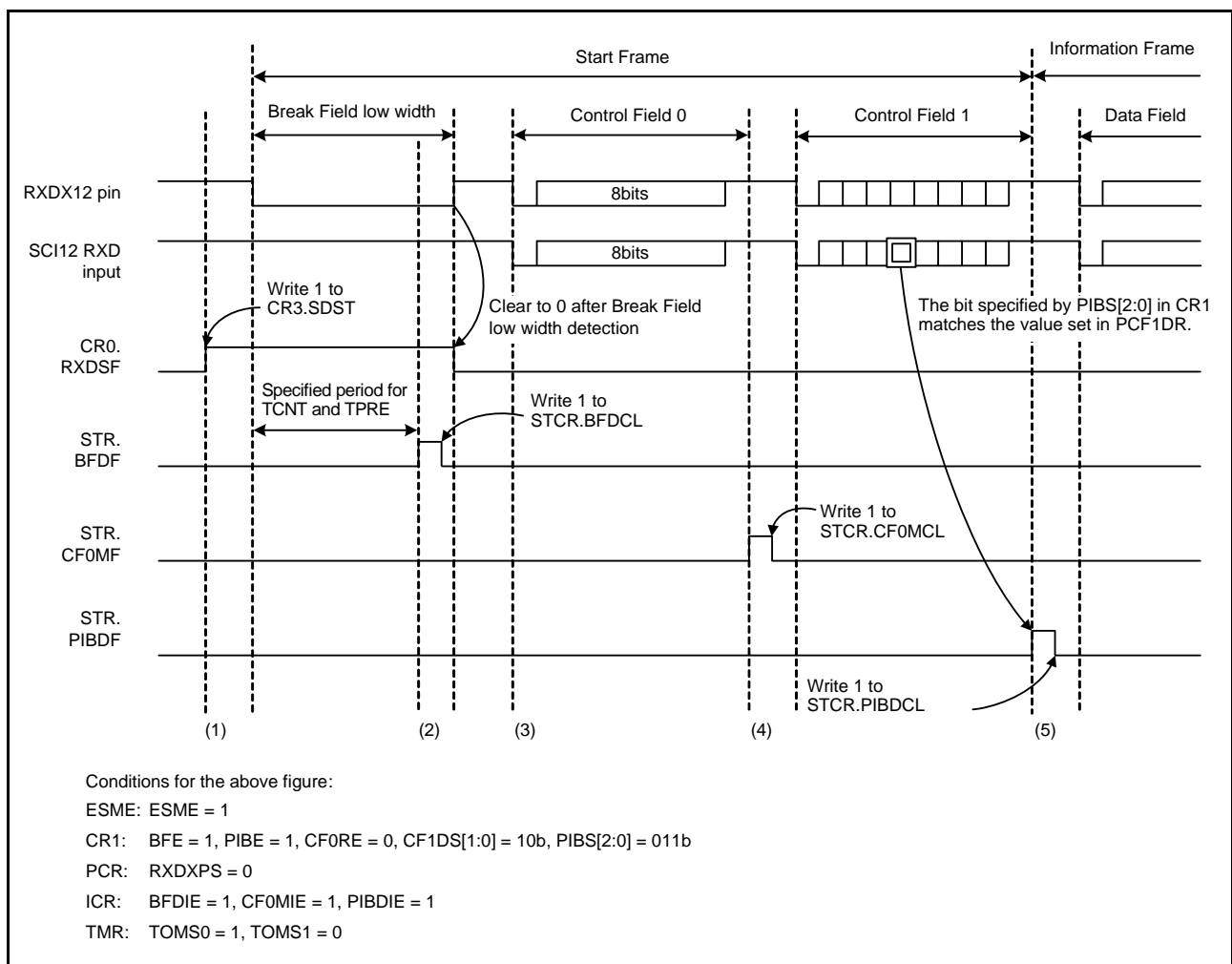


Figure 32.61 Example of Operations at the Time of Start-Frame Reception (with Priority Interrupts in Use)

### 32.9.4 Detection of Bus Collisions

Detection of bus collisions operate for cases where output of the Break Field low width and transmission of data by the SCI12 module are in progress when the `ESMER.ESME` and the `SCI12.SCR.TE` are set to 1.

Figure 32.62 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 `CR2.BCCS[1:0]` as the sampling clock, and the `BCDF` bit in `STR` 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 `BCDIE` bit in `ICR` is 1.

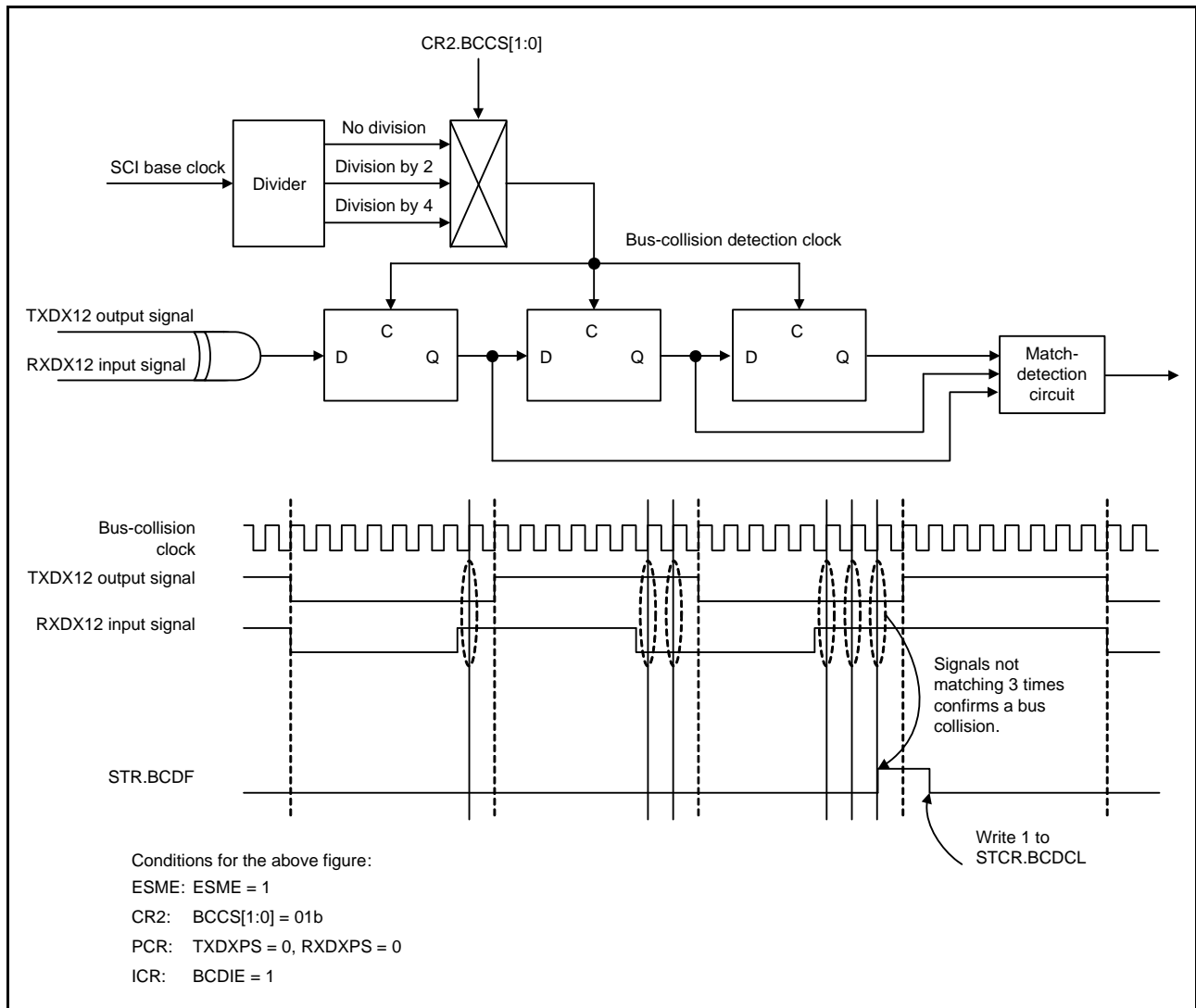


Figure 32.62 Example of Operations with Bus-Collision Detection

### 32.9.5 Digital Filter for Input on the RXDX 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 DFCS[2:0] bits in CR2 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 32.63 shows an example of operations with the digital filter.

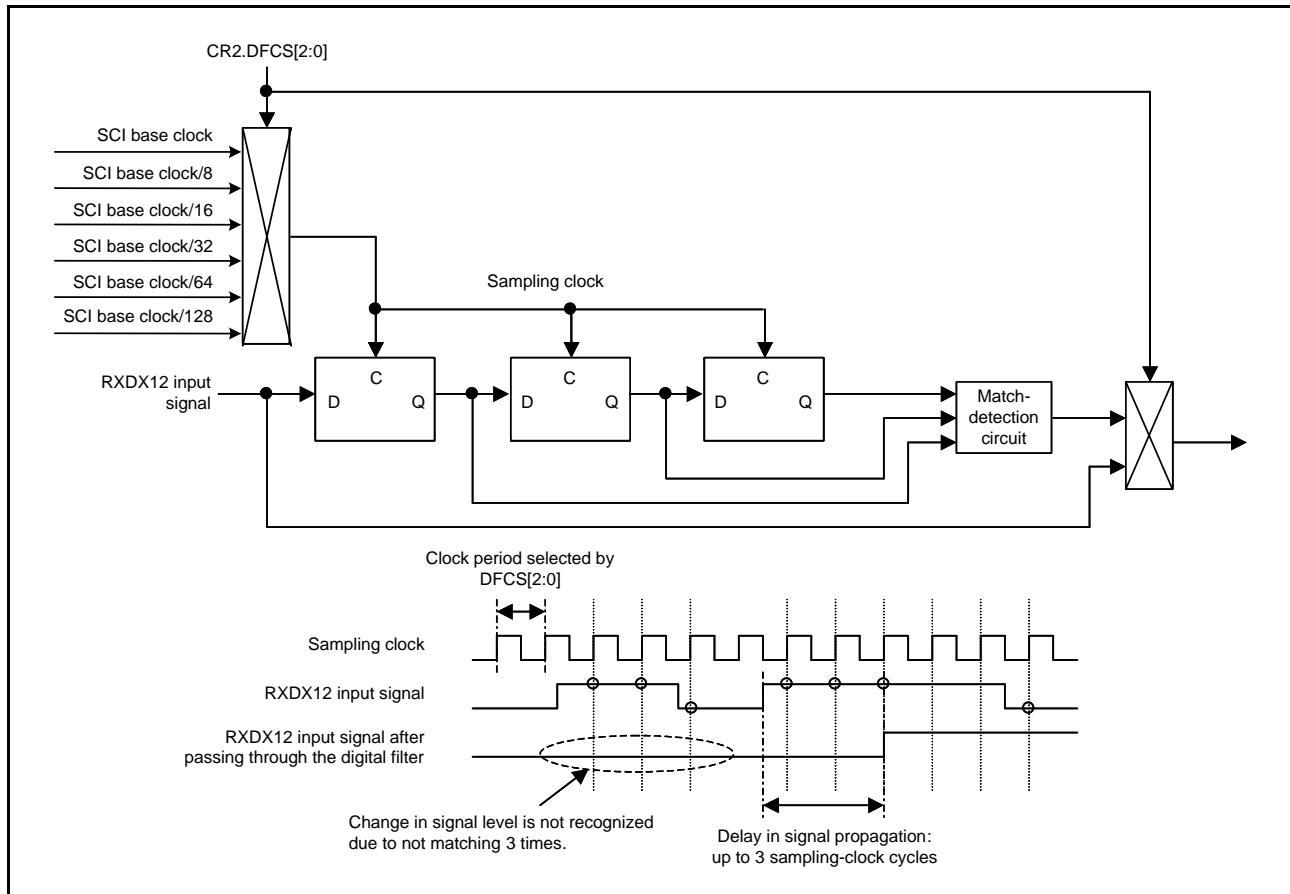


Figure 32.63 Example of Operations with the Digital Filter

### 32.9.6 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 32.64 shows an example of operations for bit-rate measurement.

- (1) Writing 1 to the BRME bit in CR0 enables bit-rate measurement. Only set BRME to 1 when you wish to proceed with bit-rate measurement. Furthermore, bit-rate measurement will not proceed during a Break Field, even if BRME 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 AEDIE bit in ICR is 1. Retention by TCNT and TPRES 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 SCI12 module. To disable the bit-rate measurement after a match with Control Field 1, write 0 to the BRME bit in CR0.

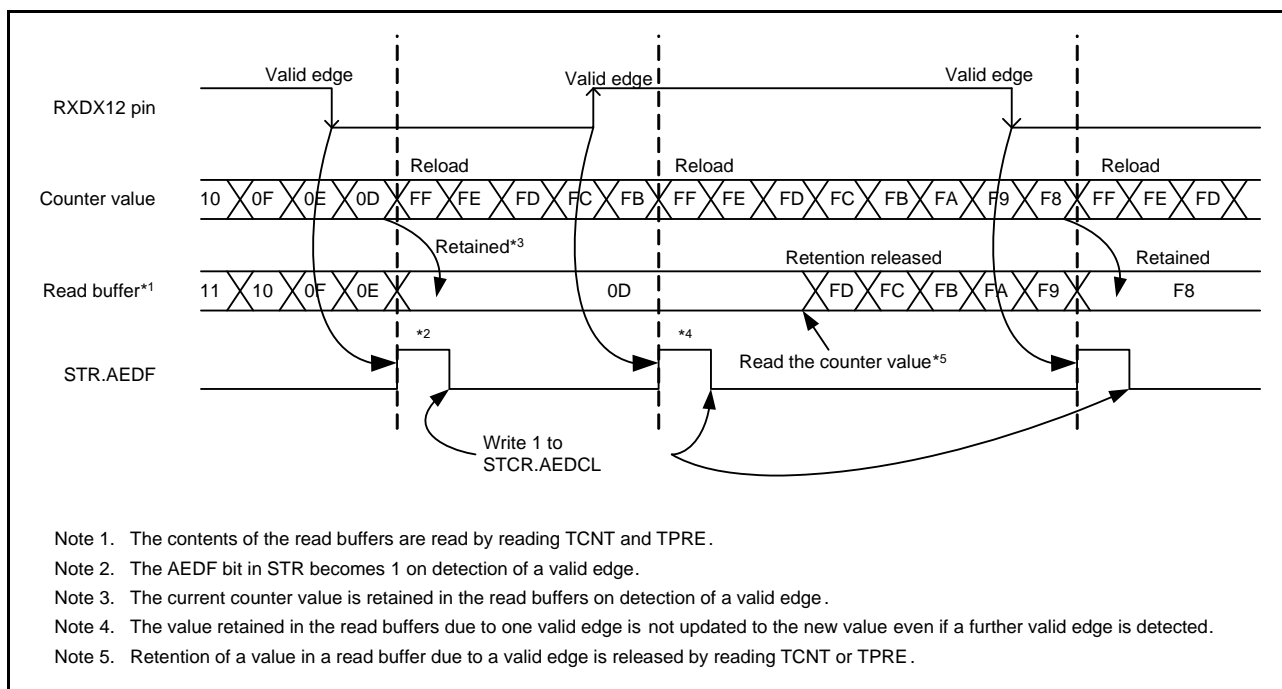


Figure 32.64 Example of Operations for Bit-Rate Measurement

### 32.9.7 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 of an SCI12 module by setting the RS0 and RS1 bits in CR2 to select the rising edges of 8, 10, 12, or 14 cycles of SCI base clock. If the value of the ABCS bit in SEMR is 1, the bits select the rising edges of 4, 5, 6, or 7 cycles of the SCI base clock of the SCI12 module. Figure 32.65 shows timing for the sampling of data received through RXDX12.

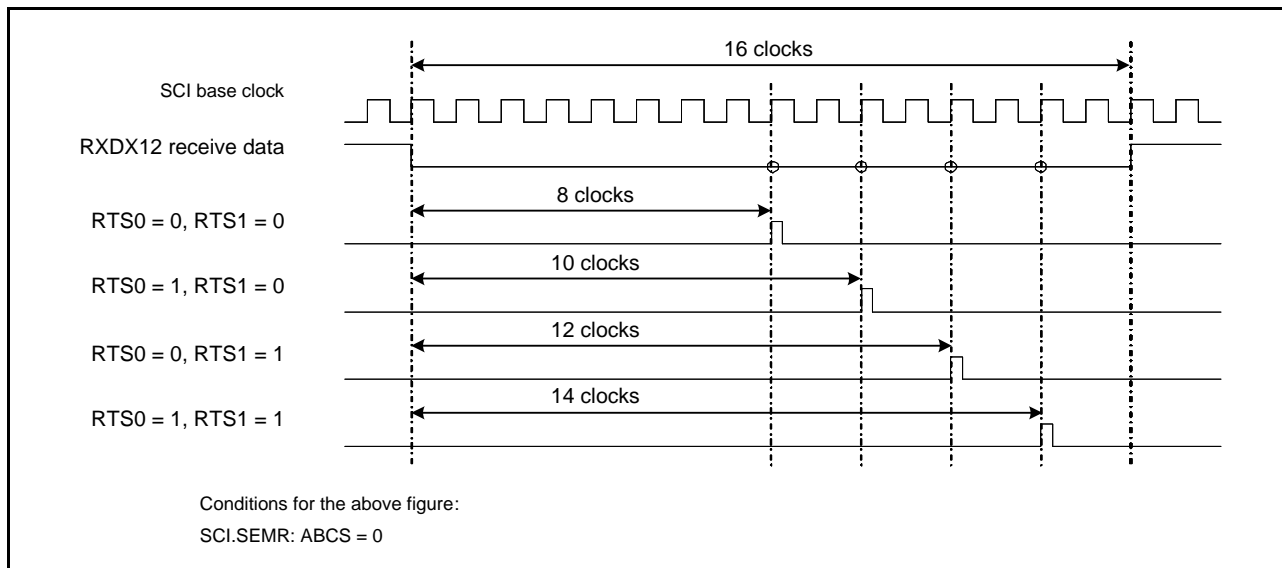


Figure 32.65 Timing for Sampling of Data Received through RXDX12

### 32.9.8 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 time of transmitting a Start Frame. Setting TOMS0 to 0 and TOMS1 to 1 in TMR switches operation to Break Field low width output mode. The TCSS[2:0] bits in TMR select the clock source for the counter. When the TCST bit in TCR 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 BFDL bit in the STR is set to 1. An SCIX0 interrupt is also generated if the value of the BFDIE bit in ICR is 1. When 0 is written to the TCST bit in TCR, counting stops after reloading of TPRE and TCNT. After output of the Break Field low width is completed, stop the timer before it underflows again. Figure 32.66 shows an example of operations in Break Field low width output mode.

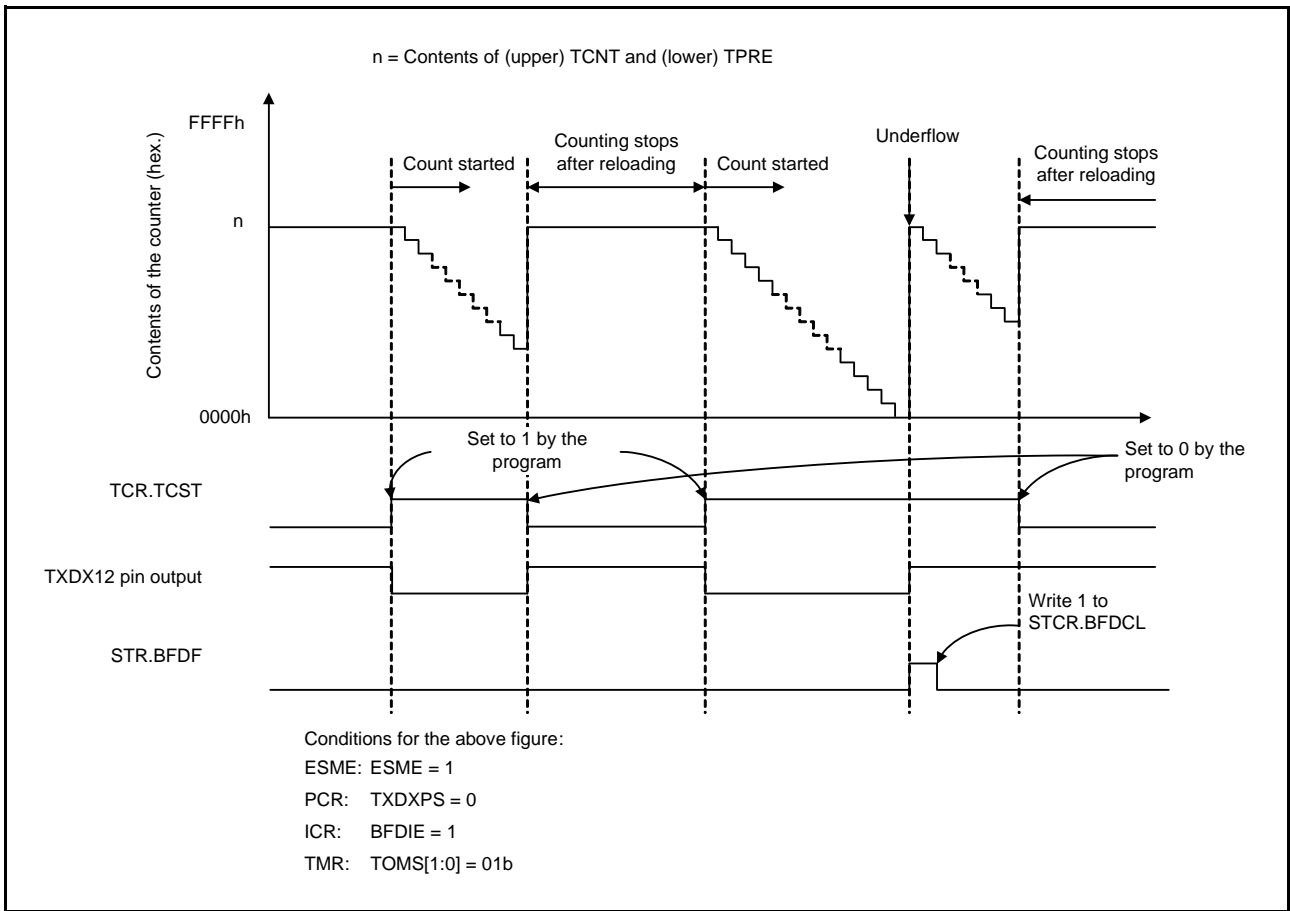


Figure 32.66 Example of Operations in Break Field Low Width Output Mode

### (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 time of receiving a Start Frame. Setting TOMS0 to 1 and TOMS1 to 0 in TMR switches operation to Break Field low width determination mode. The TCSS[2:0] bits in TMR select the clock source for the counter. When the TCST bit in TCR 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, TPRES 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 BFD bit in STR is set to 1. An SCIX0 interrupt is also generated if the value of the BFDIE bit in ICR 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 32.67 shows an example of operations in Break Field low width output mode.



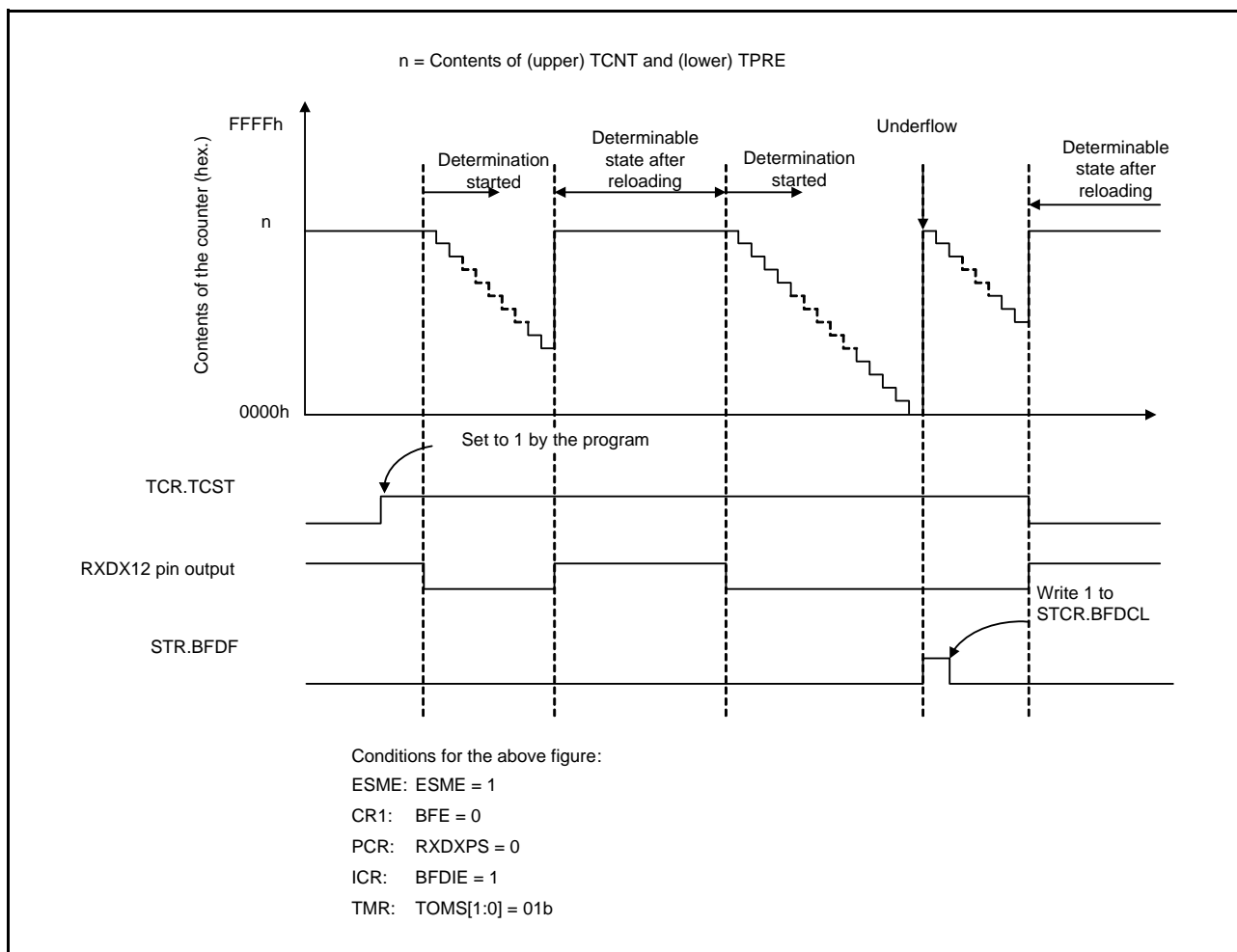


Figure 32.67 Example of Operations in Break Field Low Width Determination Mode

### (3) Timer Mode

This mode is for counting cycles of the internal clock as the clock source. Setting TOMS0 to 0 and TOMS1 to 0 in TMR switches operation to timer mode. The TCSS[2:0] bits in TMR select the clock source for counting. Counting starts when 1 is written to the TCST bit in TCR and stops when 0 is written to TCST. TPRE and TCNT both count down. TPRE counts cycles of the clock source for counting, and underflows of TPRE provide the clock source for counting by TCNT. When the timer underflows, the BFDF bit in STR is set to 1. An SCIX0 interrupt is also generated if the value of the BFDIE bit in ICR is 1.

## 32.10 Noise Cancellation Function

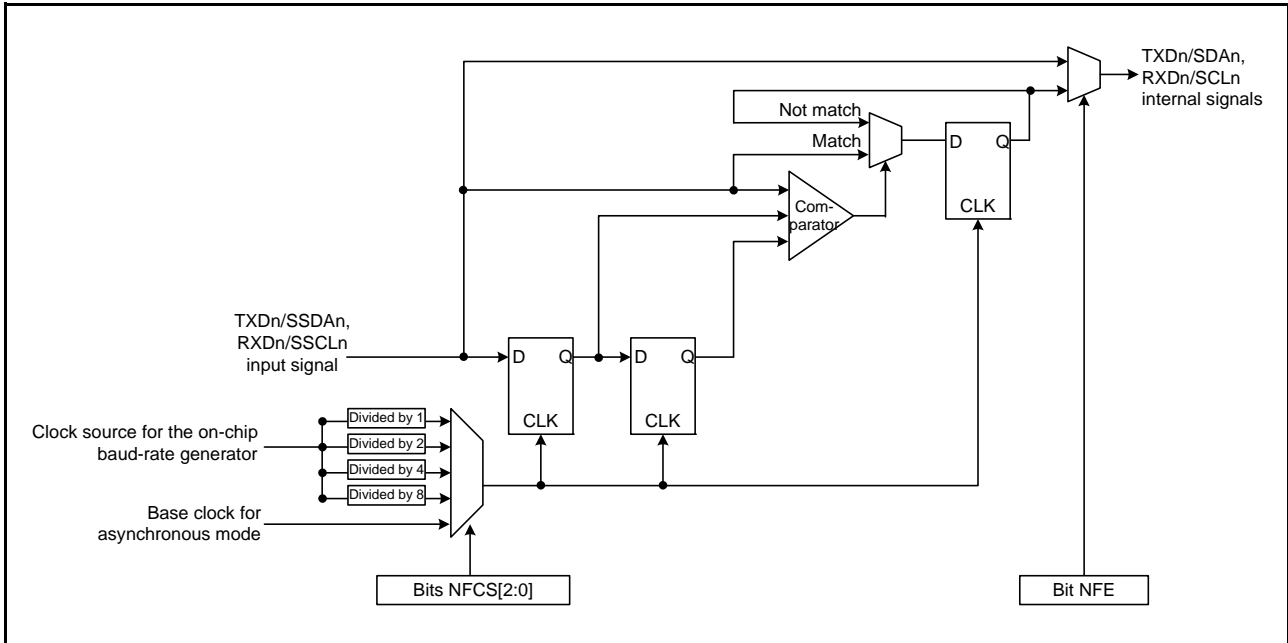
Figure 32.68 shows the configuration of the noise filter used for noise cancellation. The noise filter consists of two stages of flip-flop circuits and a match-detection circuit. When the level on the pin matches in three consecutive samples taken at the set sampling interval, the matching level continues 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<sup>2</sup>C 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 32.68** Block Diagram of Digital Noise Filter Circuit

### 32.11 Interrupt Sources

#### 32.11.1 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 interrupt controller 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 interrupt controller 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) can also be used to discard an internally retained interrupt request.

#### 32.11.2 Interrupts in Serial Communications Interface and Simple SPI Mode

Table 32.26 lists interrupt sources in serial communication interface mode and simple SPI mode. A different interrupt vector is assigned to each interrupt source, and individual interrupt sources can be enabled or disabled with the enable bits in SCR.

If the SCR.TIE bit is 1, a TXI interrupt request is generated when data for transmission are transferred from the TDR to the TSR. 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 or DMAC 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.\*1

When new data are not written by the time of transmission of the last bit of the current data for transmission 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 data for transmission are written to the TDR, and setting the SCR.TEIE bit to 1 leads to the generation of a TEI interrupt request.

Writing data to the TDR 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 are stored in the RDR. An RXI interrupt request can activate the DTC or DMAC to handle data transfer.

Setting of any from among the ORER, FER, and PER flags in the SSR 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. To temporarily prohibit TXI interrupts at the time of transmission of the last of the data and so on when you wish a new round of transmission to start after handling of the transmission-completed interrupt, control prohibiting and permitting 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 32.26 Interrupt Sources**

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation	Priority
ERI	Receive error	ORER, FER, or PER	Not possible	Not possible	High
RXI	Receive data full	—	Possible	Possible	↑
TXI	Transmit data empty	—	Possible	Possible	
TEI	Transmit end	TEND	Not possible	Not possible	Low

### 32.11.3 Interrupts in Smart Card Interface Mode

Table 32.27 lists interrupt sources in smart card interface mode. A transmit end interrupt (TEI) request cannot be used in this mode.

**Table 32.27 SCI Interrupt Sources**

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMACA Activation	Priority
ERI	Receive error or error signal detection	ORER, PER, or ERS	Not possible	Not possible	High
RXI	Receive data full	—	Possible	Possible	↑
TXI	Transmit data empty	TEND	Possible	Possible	

Data transmission/reception using the DTC or DMAC is also possible in smart card interface mode, similar to in the normal SCI mode. In transmission, when the TEND flag in SSR is set to 1, a TXI interrupt request is generated. This TXI interrupt request activates the DTC or DMAC allowing transfer of transmit data if the TXI request is specified beforehand as a source of DTC or DMAC activation. The TEND flag is automatically cleared to 0 when the DTC or DMAC transfers the data.

If an error occurs, the SCI automatically re-transmits the same data. During the retransmission, the TEND flag is kept to 0 and the DTC or DMAC is not activated. Therefore, the SCI and DTC or DMAC automatically transmit the specified number of bytes, including retransmission in the case of error occurrence. However, the ERS flag in SSR is not automatically cleared to 0 at error occurrence. Therefore, the ERS flag must be cleared by previously setting the RIE bit in SCR to 1 to enable an ERI interrupt request to be generated at error occurrence.

When transmitting/receiving data using the DTC or DMAC, be sure to make settings to enable the DTC or DMAC before making SCI settings. For DTC or DMAC settings, see section 18, DMA Controller (DMACA) and section 19, Data Transfer Controller (DTCa).

In reception, an RXI interrupt request is generated when receive data is set to RDR. This RXI interrupt request activates the DTC or DMAC allowing transfer of receive data if the RXI request is specified beforehand as a source of DTC or DMAC activation. If an error occurs, the error flag is set. Therefore, the DTC or DMAC is not activated and an ERI interrupt request is issued to the CPU instead; the error flag must be cleared.

#### 32.11.4 Interrupts in Simple I<sup>2</sup>C Mode

The interrupt sources in simple I<sup>2</sup>C mode are listed in Table 32.28. The STI interrupt is allocated to the transmit end interrupt (TEI) request. The receive error interrupt (ERI) request cannot be used.

The DTC and DMAC can also be used to handle transfer in simple I<sup>2</sup>C mode.

When the value of the IICINTM bit in SIMR2 is 1, an RXI request will be generated on the falling edge of the SSCL signal for the eighth bit. If the RXI has been set up as an activating request for the DTC or DMAC beforehand, the RXI request will activate the DTC or DMAC 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 or DMAC beforehand, the TXI request will activate the DTC or DMAC to handle transfer of the data for transmission.

When the value of the IICINTM bit in SIMR2 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 or DMAC beforehand, the RXI request will activate the DTC or DMAC to handle transfer of the received data.

Also, if the DTC or DMAC is used for data transfer in reception or transmission, be sure to set up and enable the DTC or DMAC before setting up the SCI.

When the IICSTAREQ, IICSTAREQ, and IICSTPREQ bits in SIMR3 are used to generate a start condition, restart condition, or stop condition, the STI request is issued when generation is complete.

**Table 32.28 SCI Interrupt Sources**

Name	Interrupt Source	Interrupt Flag	DTC Activation	DMAC Activation	Priority
RXI	Reception, ACK detection	—	Possible	Possible	High ↑ Low
TXI	Transmission, NACK detection	—	Possible* <sup>1</sup>	Possible* <sup>1</sup>	
STI	Completion of generating a start, restart, or stop condition	IICSTIF	Not possible	Not possible	Low

Note 1. Activation of the DTC or DMAC is only possible when the SIMR2.IICINTM bit is 1 (selecting reception and interrupts).

### 32.11.5 Interrupts 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 32.29.

**Table 32.29 Interrupt Sources of the Extended Serial Mode Control Section**

Interrupt Request	Status Flag	Interrupt Factors
SCIX0 interrupt (Break Field low width detected)	BFDF	<ul style="list-style-type: none"> <li>Detection of a Break Field low width longer than the interval corresponding to the timer setting</li> <li>Completion of the output of a Break Field low width over the interval corresponding to the timer setting</li> <li>Underflow of the timer</li> </ul>
SCIX1 interrupt (Control Field 0 match)	CF0MF	The data received in Control Field 0 matching the value set in CF0DR
SCIX1 interrupt (Control Field 1 match)	CF1MF	The data received in Control Field 1 matching the value set in PCF1DR or SCF1DR
SCIX1 interrupt (priority interrupt bit detected)	PIBDF	The value of the bit specified as the priority interrupt bit matching the value set in PCF1DR
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

## 32.12 Usage Notes

### 32.12.1 Setting the Module-Stop Function

Module-stop control registers B and C (MSTPCRB and MSTPCRC) are used to stop and start SCI operations. With the value after a reset, SCI operations are stopped. The registers of the modules only become accessible after release from the module-stop state. For details, refer to section 11, Low Power Consumption.

### 32.12.2 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 FER flag in SSR is set to 1 (framing error), and the PER flag in SSR may also be set to 1 (parity error). The SCI continues the receive operation even after a break is received. Therefore, note that even if the FER flag is cleared to 0 (no framing error), it will be set to 1 again.

### 32.12.3 The Mark State and Production of Breaks

When the SCR.TE bit is 0 (prohibiting serial transmission), setting the I/O port function makes selection of the level and direction (input or output) of the TXDn pin possible. If this is done, the TXDn pin can be placed in the mark state to send a break at the time of data transmission. Until the SCR.TE bit is set to 1 (permitting serial transmission), the I/O port function is used to set the TXDn pin to output 1 and set the pin mode to a general I/O port pin, and thus place the transfer circuit in the mark state (state of having the value 1). On the other hand, to output a break at the time of data transmission, set the TXDn pin to output 0 and make the pin mode settings for a general I/O port pin. When the SCR.TE bit is set to 0, the transmission section is initialized regardless of the current state of transmission.

### 32.12.4 Receive Error Flags and Transmit Operations (Clock Synchronous Mode Only)

Transmission cannot be started when a receive error flag (ORER) in SSR is set to 1, even if data is written to TDR. Be sure to clear the receive error flags to 0 before starting transmission. Note also that the receive error flags cannot be cleared to 0 even if the RE bit in SCR is cleared to 0 (serial reception disabled).

### 32.12.5 Writing Data to TDR

Data can always be written to TDR. However, if new data is written to TDR when transmit data is remaining in TDR, the previous data in TDR is lost because it has not been transferred to TSR yet. Be sure to write transmit data to TDR in the TXI interrupt request processing routine.

### 32.12.6 Restrictions on Clock Synchronous Transmission

When the external clock source is used as a synchronization clock, update TDR by the DMAC or DTC and wait for at least five clock cycles before allowing the transmit clock to be input. If the transmit clock is input within four clock cycles after TDR is updated, the SCI may malfunction.

### 32.12.7 Restrictions on Using DTC or DMAC

When using the DMAC or DTC to read RDR, be sure to set the receive end interrupt (RXI) as the activation source of the relevant SCI.

### 32.12.8 Points to Note on Starting Transfer

At the point where transfer starts when the interrupt status flag in the interrupt controller is 1, follow the procedure below to clear interrupt requests before permitting operations (by setting the SCR.TE or SCR.RE bit to 1).

- 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 out the corresponding interrupt enable bit (SCR.TIE or SCR.RIE bit) to check that it has actually become 0.
- Set the interrupt status flag in the interrupt controller to 0.

### 32.12.9 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 to 0) after switching the TXDn pin to the general I/O port pin function. Clearing the TE bit to 0 resets the TSR and the TEND bit in the SSR. Depending on the port settings, output pins may output the high level after release from the module-stopped state or software standby. When transitions to these states are made during transmission, the data being transmitted become indeterminate.

To transmit data in the same transmission mode after cancellation of the low power consumption state, set the TE bit to 1, read SSR, and write data to TDR sequentially to start data transmission. To transmit data with a different transmission mode, initialize the SCI first.

Figure 32.69 shows a sample flowchart for transition to software standby mode during transmission. Figure 32.70 and Figure 32.71 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 transmission mode using DTC transfer, stop the transmit operations (TE = 0). To start transmission after cancellation using the DTC, set the TE bit 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 (RE = 0 in SCR). If transition is made during data reception, the data being received will be invalid.

To receive data in the same reception 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 reception mode, initialize the SCI first.

Figure 32.72 shows a sample flowchart for transition to software standby mode during reception.

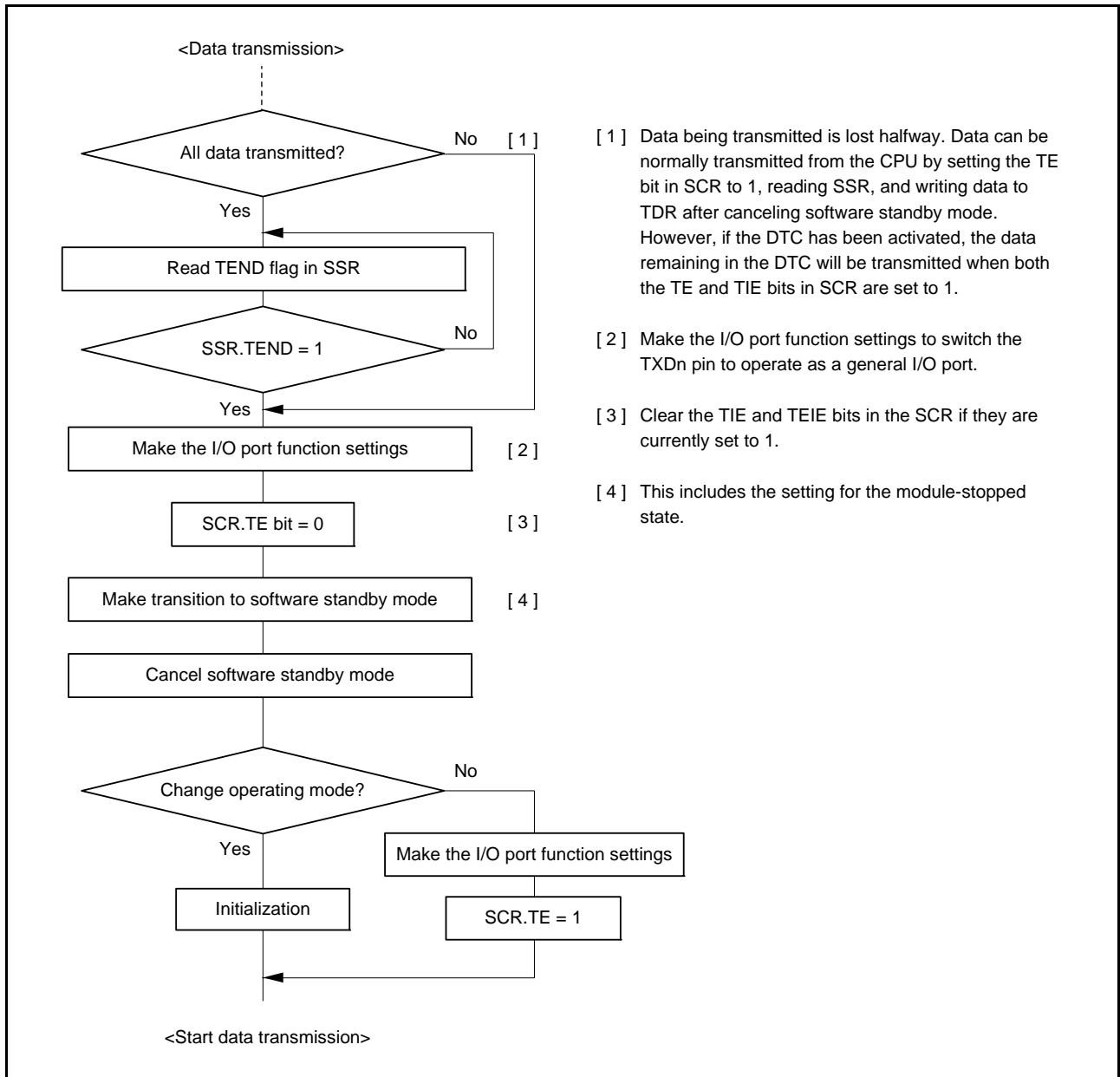


Figure 32.69 Example of Flowchart for Transition to Software Standby Mode during Transmission

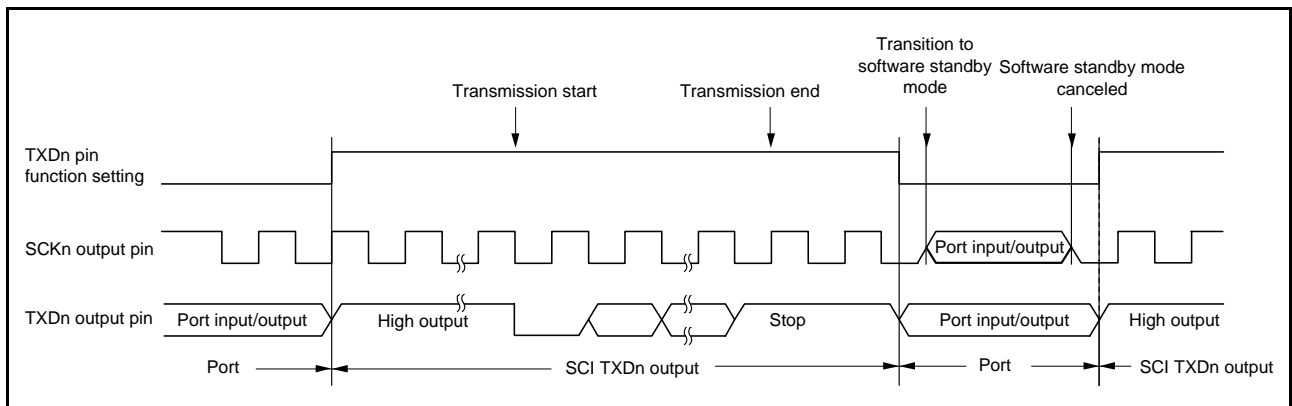
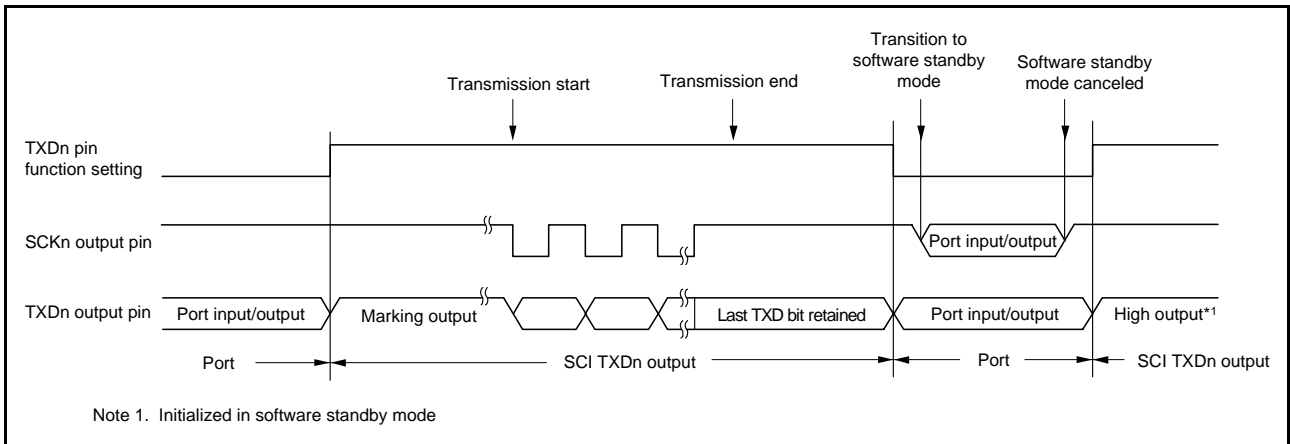
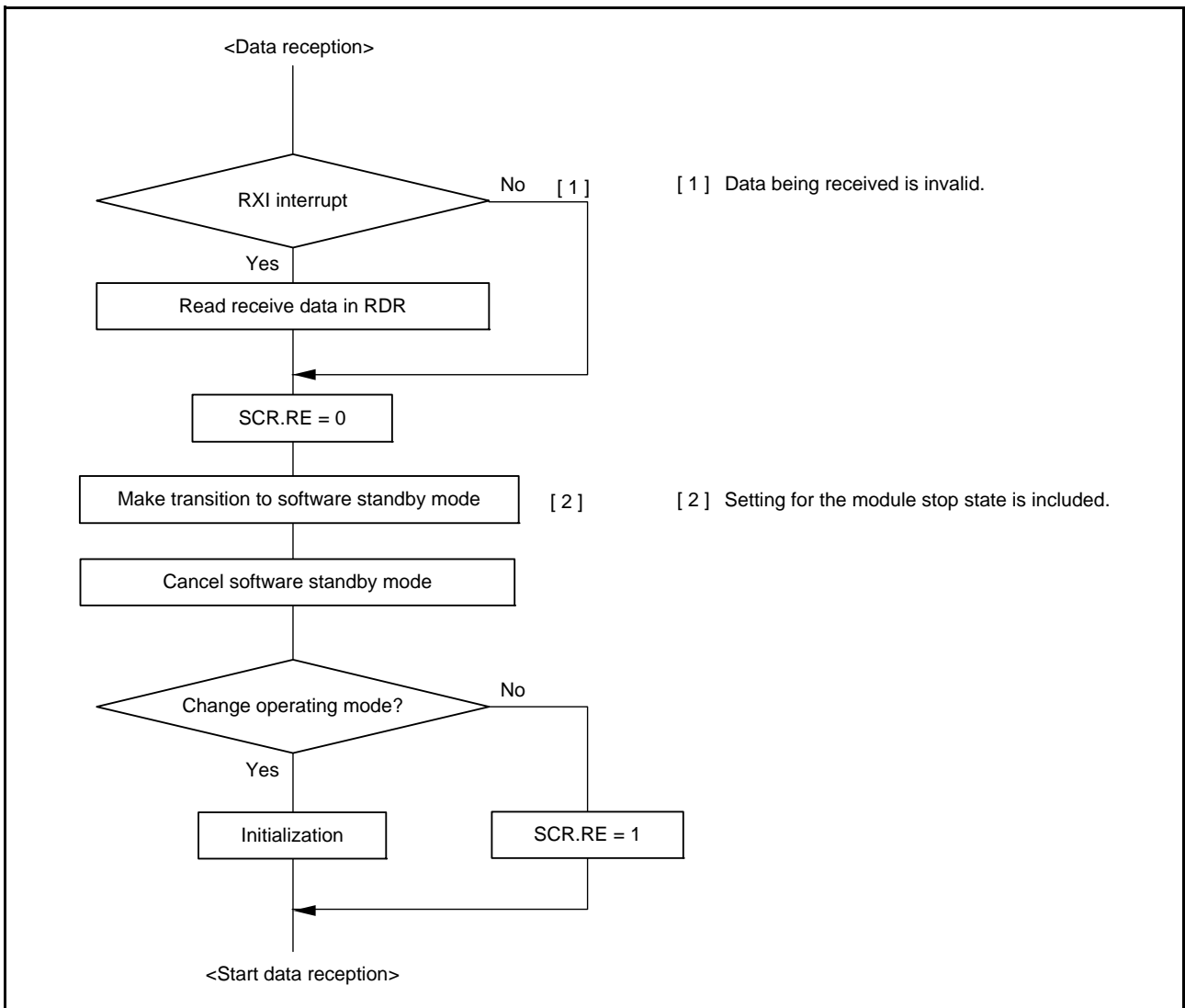


Figure 32.70 Port Pin States during Transition to Software Standby Mode (Internal Clock, Asynchronous Transmission)



**Figure 32.71 Port Pin States during Transition to Software Standby Mode (Internal Clock, Clock Synchronous Transmission)**



**Figure 32.72 Example of Flowchart for Transition to Software Standby Mode during Reception**



### 32.12.10 External Clock Input in Clock Synchronous Mode

In clock synchronous 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

### 32.12.11 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 CKPH and CKPOL bits in SPMR.
- In the case of the setting for clock delay (the SPMR.CKPH bit is 1), the received data full interrupt (RXI interrupt) is generated before the final clock edge on the SCKn pin as indicated in Figure 32.73. If the TE and RE bits in the SCR 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.
- When operation is in multi-master mode, take care because the SCKn pin output becomes high impedance while the input on the SS# 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

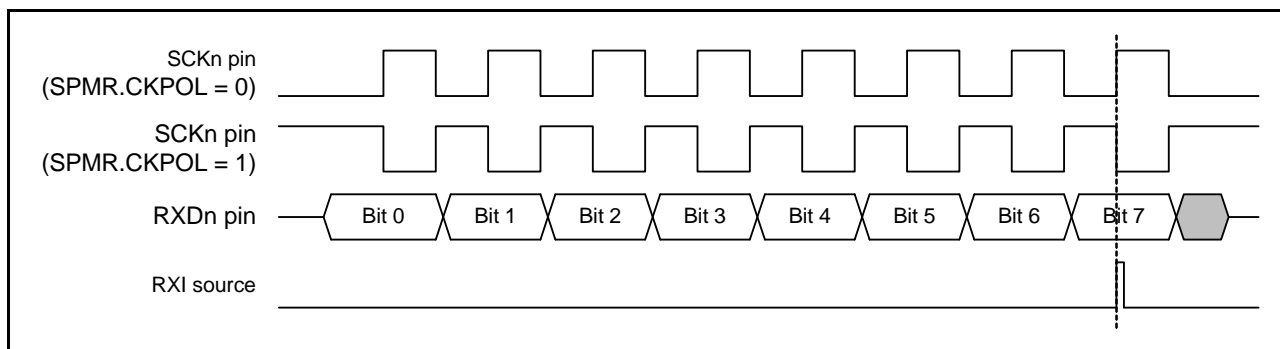


Figure 32.73 Timing of the RXI Interrupt in Simple SPI Mode (with Clock Delay)

#### (2) Slave Mode

- When data for transmission are written to the TDR, secure at least five cycles of the PCLK from input of the low level on the SSn# pin to input of the external clock.
- Provide an external clock signal to the master the same as the data length for transfer.
- 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 to 0 and, after remaking the settings, restart transfer of the first byte.

### 32.12.12 Limitation 1 on Usage of the Extended Serial Mode Control Section

When the SHARPS bit in PCR is set to 1, output on the TXDX12/RXDX12 pin is only possible when the following conditions apply.

- The timer of the SC1d module is in Break Field low width output mode and the value of the TCST bit in TCR 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 TE bit in SC12.SCR is 1.

### 32.12.13 Limitation 2 on Usage of the Extended Serial Mode Control Section

An SC1c interrupt request is generated even if the extended serial mode is enabled. However, the SC1c interrupt should not be used during reception of a start frame because SC1d uses an SC1c interrupt request.

The two ways of dealing with this are described below. When a reception error is detected, clear the error flag of the SC1c and initialize the control section of the SC1d following the example of flowchart shown in Figure 32.74.

- (1) Set the SCR.RIE bit of the SC1c to 0 to disable the output of interrupt requests. Check the error flags in the SSR register for SC1c on completion of the reception of a start frame, because an ERI interrupt is not generated if a reception error occurs. After reception of the start frame is completed, set the SCR.RIE bit of the SC1c to 1 by the time the first byte of the information frame is received.
- (2) Set the SCR.RIE bit of the SC1c 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.

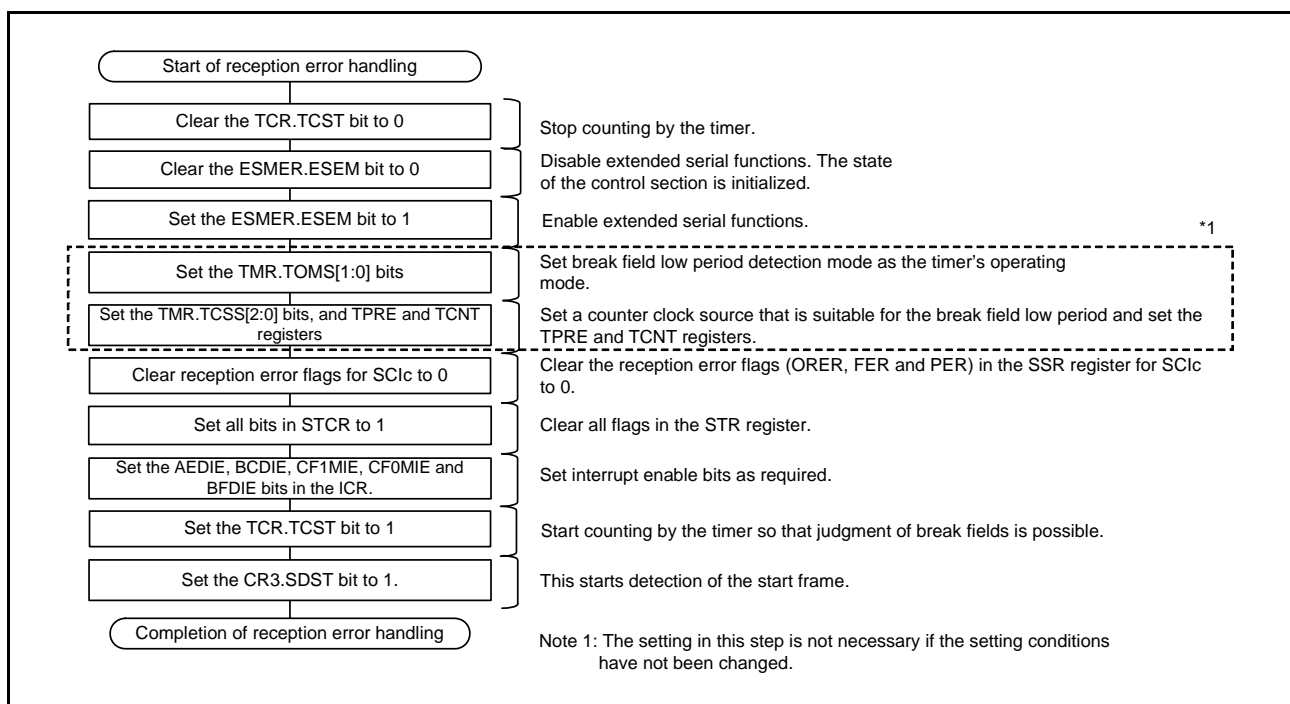


Figure 32.74 Example of Flowchart for Reception Error Handling (During Reception of the Start Frame)

## 33. I<sup>2</sup>C Bus Interface (RIIC)

The RX630 Group has four I<sup>2</sup>C bus interfaces (RIIC modules).

The RIIC module conforms with and provides a subset of the NXP I<sup>2</sup>C bus (Inter-IC-Bus) interface functions.

### 33.1 Overview

Table 33.1 lists the specifications of the RIIC, Figure 33.1 shows a block diagram of the RIIC, and Figure 33.2 shows an example of I/O pin connections to external circuits (I<sup>2</sup>C bus configuration example). Table 33.2 lists the I/O pins of the RIIC.

**Table 33.1 RIIC Specifications**

Item	Specifications
Communications format	<ul style="list-style-type: none"> <li>I<sup>2</sup>C bus format or SMBus format</li> <li>Master mode or slave mode selectable</li> <li>Automatic securing of the various set-up times, hold times, and bus-free times for the transfer rate</li> </ul>
Transfer rate	Up to 1 Mbps
SCL clock	For master operation, the duty cycle of the SCL clock is selectable in the range from 4% to 96%.
Issuing and detecting conditions	Start, restart, and stop conditions are automatically generated. Start conditions (including restart conditions) and stop conditions are detectable.
Slave address	<ul style="list-style-type: none"> <li>Up to three slave-address settings can be made.</li> <li>Seven- and ten-bit address formats are supported (along with the use of both at once).</li> <li>General call addresses, device ID addresses, and SMBus host addresses are detectable.</li> </ul>
Acknowledgement	<ul style="list-style-type: none"> <li>For transmission, the acknowledge bit is automatically loaded. Transfer of the next data for transmission can be automatically suspended on detection of a not-acknowledge bit.</li> <li>For reception, the acknowledge bit is automatically transmitted. If a wait between the eighth and ninth clock cycles has been selected, software control of the value in the acknowledge field in response to the received value is possible.</li> </ul>
Wait function	<ul style="list-style-type: none"> <li>In reception, the following periods of waiting can be obtained by holding the clock signal (SCL) at the low level: <ul style="list-style-type: none"> <li>Waiting between the eighth and ninth clock cycles</li> <li>Waiting between the ninth clock cycle and the first clock cycle of the next transfer (WAIT function)</li> </ul> </li> </ul>
SDA output delay function	Timing of the output of transmitted data, including the acknowledge bit, can be delayed.
Arbitration	<ul style="list-style-type: none"> <li>For multi-master operation <ul style="list-style-type: none"> <li>Operation to synchronize the SCL (clock) signal in cases of conflict with the SCL signal from another master is possible.</li> <li>When issuing the start condition would create conflict on the bus, loss of arbitration is detected by testing for non-matching between the internal signal for the SDA line and the level on the SDA line.</li> <li>In master operation, loss of arbitration is detected by testing for non-matching between the signal on the SDA line and the internal signal for the SDA line.</li> </ul> </li> <li>Loss of arbitration due to detection of the start condition while the bus is busy is detectable (to prevent the issuing of double start conditions).</li> <li>Loss of arbitration in transfer of a not-acknowledge bit due to the internal signal for the SDA line and the level on the SDA line not matching is detectable.</li> <li>Loss of arbitration due to non-matching of internal and line levels for data is detectable in slave transmission.</li> </ul>
Timeout function	The internal time-out function is capable of detecting long-interval stop of the SCL (clock signal).
Noise removal	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	<ul style="list-style-type: none"> <li>Four sources: <ul style="list-style-type: none"> <li>Error in transfer or occurrence of events (detection of AL, NACK, time-out, a start condition including a restart condition, or a stop condition)</li> <li>Receive-data-full (including matching with a slave address)</li> <li>Transmit-data-empty (including matching with a slave address)</li> <li>Transmission complete</li> </ul> </li> </ul>
Low power consumption function	Module-stop state can be set.

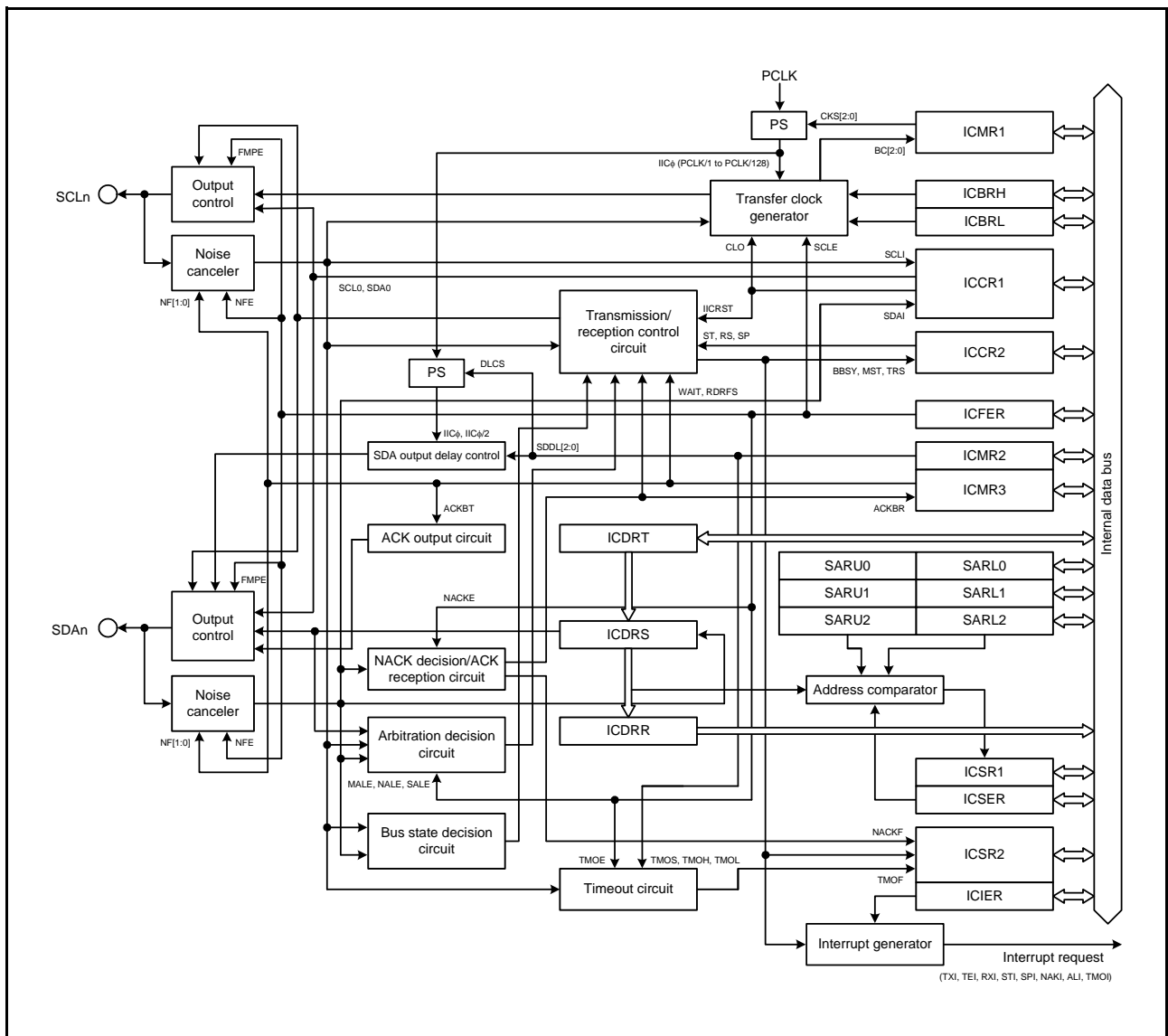
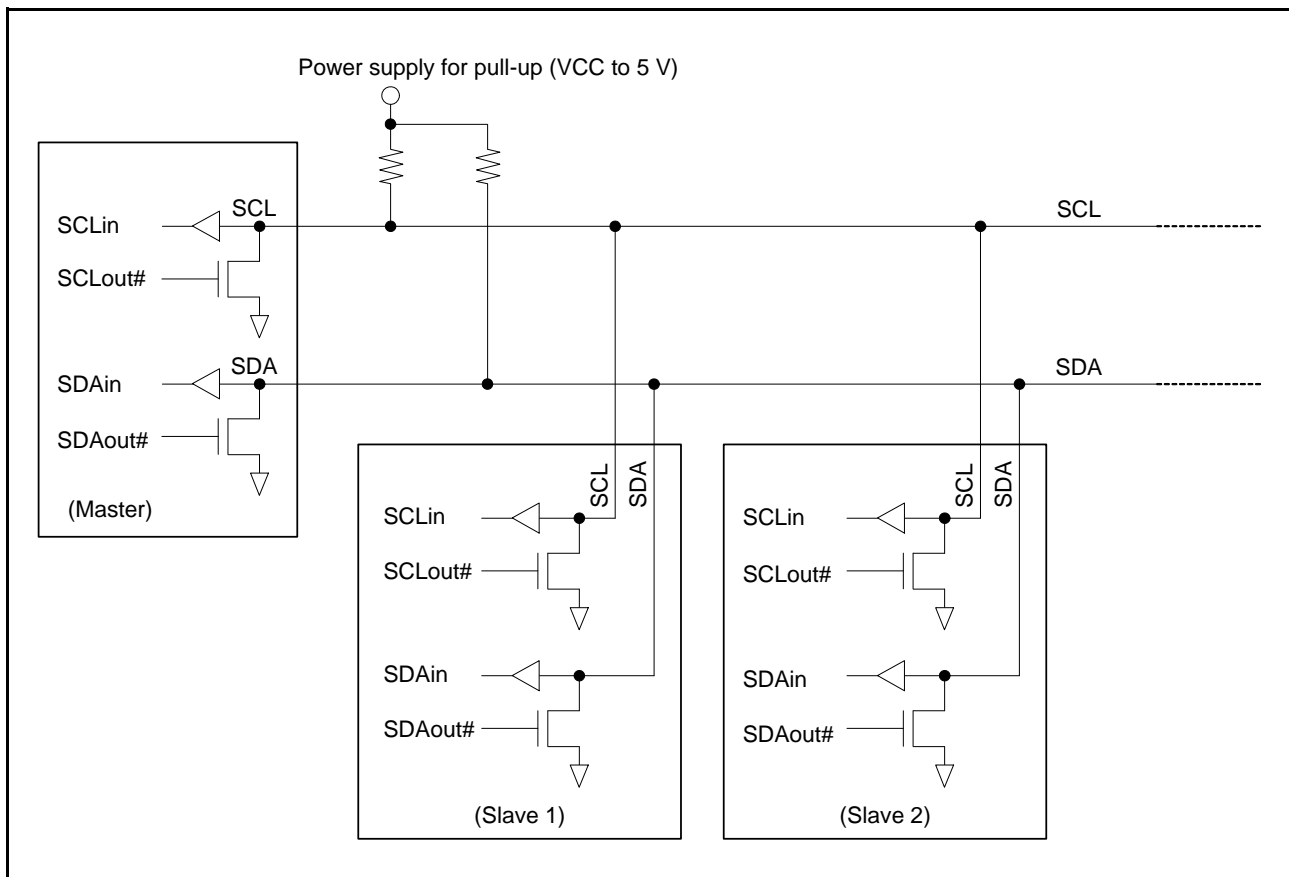


Figure 33.1 Block Diagram of RIIC



**Figure 33.2** Connections to the External Circuit by the I/O Pins (I<sup>2</sup>C Bus Configuration Example)

The input level of the signals for RIIC is CMOS when I<sup>2</sup>C bus is selected (the ICMR3.SMBS bit = 0), or TTL when SMBus is selected (the ICMR3.SMBS bit = 1).

**Table 33.2** 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
RIIC1	SCL1	I/O	RIIC1 serial clock I/O pin
	SDA1	I/O	RIIC1 serial data I/O pin
RIIC2	SCL2	I/O	RIIC2 serial clock I/O pin
	SDA2	I/O	RIIC2 serial data I/O pin
RIIC3	SCL3	I/O	RIIC3 serial clock I/O pin
	SDA3	I/O	RIIC3 serial data I/O pin

## 33.2 Register Descriptions

### 33.2.1 I<sup>2</sup>C Bus Control Register 1 (ICCR1)

Address(es): RIIC0.ICCR1 0008 8300h, RIIC1.ICCR1 0008 8320h, RIIC2.ICCR1 0008 8340h, RIIC3.ICCR1 0008 8360h

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 Bus Input Monitor	0: SDA <sub>n</sub> pin input is at a low level. 1: SDA <sub>n</sub> pin input is at a high level.	R
b1	SCLI	SCL Bus Input Monitor	0: SCL <sub>n</sub> pin input is at a low level. 1: SCL <sub>n</sub> pin input is at a high level.	R
b2	SDAO	SDA Output Control	<ul style="list-style-type: none"> <li>Read:               <ul style="list-style-type: none"> <li>0: SDA<sub>n</sub> pin output is at a low level.</li> <li>1: SDA<sub>n</sub> pin is in a high-impedance state.</li> </ul> </li> <li>Write:               <ul style="list-style-type: none"> <li>0: Changes the SDA<sub>n</sub> pin output to a low level.</li> <li>1: Changes the SDA<sub>n</sub> pin in a high-impedance state. (High level output is achieved through an external pull-up resistor.)</li> </ul> </li> </ul>	R/W *1,*2
b3	SCLO	SCL Output Control	<ul style="list-style-type: none"> <li>Read:               <ul style="list-style-type: none"> <li>0: SCL<sub>n</sub> pin output is at a low level.</li> <li>1: SCL<sub>n</sub> pin is in a high-impedance state.</li> </ul> </li> <li>Write:               <ul style="list-style-type: none"> <li>0: Changes the SCL<sub>n</sub> pin output to a low level.</li> <li>1: Changes the SCL<sub>n</sub> pin in a high-impedance state. (High level output is achieved through an external pull-up resistor.)</li> </ul> </li> </ul>	R/W *1,*2
b4	SOWP	SCLO/SDAO Write Protect	0: Allows the SCLO and SDAO bits to be rewritten. (This bit is read as 1.)	R/W*2
b5	CLO	Extra SCL Clock Cycle Output	0: Does not output an extra SCL <sub>n</sub> clock cycle (default). 1: Outputs an extra SCL <sub>n</sub> clock cycle. (The CLO bit is cleared automatically after one clock cycle is output.)	R/W
b6	IICRST	I <sup>2</sup> C Bus Interface Internal Reset	0: Clears the RIIC reset or internal reset. 1: Initiates the RIIC reset or internal reset. (Clears the bit counter and the SCL <sub>n</sub> /SDA <sub>n</sub> output latch)	R/W
b7	ICE	I <sup>2</sup> C Bus Interface Enable	0: Disabled (SCL <sub>n</sub> , SDA <sub>n</sub> pins not driven). 0: Enabled (SCL <sub>n</sub> , SDA <sub>n</sub> pins driven). This bit also selects an RIIC reset or internal reset in combination with the IICRST bit.	R/W

Note 1. Do not write to these bits during communication. Changing a value during communication may cause a transmission or reception failure or an AL error.

Note 2. To change the SDAO and SCLO bits, set the SOWP bit to 0 at the same timing to set the SDAO and SCLO bits to 0.

#### CLO Bit (Extra SCL Clock Cycle Output)

This bit is used to output an extra SCL clock cycle 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, see section 33.11.2, Extra SCL Clock Cycle Output Function.

**IICRST Bit (I<sup>2</sup>C 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 33.3 lists the resets of the RIIC.

The RIIC reset resets all registers including the BBSY flag in ICCR2 and internal states of the RIIC, and the internal reset resets the bit counter (BC[2:0] bits in ICMR1), the I<sup>2</sup>C bus shift register (ICDRS), and the I<sup>2</sup>C bus status registers (ICSR1 and ICSR2) as well as the internal states of the RIIC. For the reset conditions for each register, see section 33.14, Reset States.

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 SCLn pin and SDAn 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 hangs up with the SCLn line in a low level output state in slave mode, initiate an internal reset and then issue a restart condition from the master device or resume communication from the start condition issuance after issuing a stop condition. If communication is restarted by initiating a reset solely in the slave device without issuing a start condition or restart condition from the master device, synchronization will be lost because the master and slave devices operate asynchronously.

**Table 33.3 RIIC Resets**

IICRST	ICE	State	Specifications
1	0	RIIC reset	Resets all registers and internal states of the RIIC.
	1	Internal reset	Reset the BC[2:0] bits in ICMR1, and the ICSR1, ICSR2, ICDRS registers and the internal states of the RIIC.

**ICE Bit (I<sup>2</sup>C Bus Interface Enable)**

The ICE bit selects driving or non-driving of the SCLn and SDAn pins. Moreover, this bit can perform two types of reset in combination with the IICRST bit. For the types of reset, see Table 33.3, RIIC Resets.

Set the ICE bit to 1 when using RIIC. Setting the ICE bit to 1 selects driving of the SCLn and SDAn pins.

Set the ICE bit to 0 when RIIC is not to be used. Clearing the ICE bit 0 stops driving of the SCLn and SDAn pins.

Moreover, do not allocate the SCLn and SDAn pins to the RIIC when setting the multi-function pin controller (MPC).

Note that if they are allocated to the RIIC, slave address comparison will be made.

### 33.2.2 I<sup>2</sup>C Bus Control Register 2 (ICCR2)

Address(es): RIIC0.ICCR2 0008 8301h, RIIC1.ICCR2 0008 8321h, RIIC2.ICCR2 0008 8341h, RIIC3.ICCR2 0008 8361h

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 Issuance Request	0: Does not request to issue a start condition. 1: Requests to issue a start condition.	R/W
b2	RS	Restart Condition Issuance Request	0: Does not request to issue a restart condition. 1: Requests to issue a restart condition.	R/W
b3	SP	Stop Condition Issuance Request	0: Does not request to issue a stop condition. 1: Requests to issue 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: Transmit mode	R/W*1
b7	BBSY	Bus Busy Detection Flag	0: The I <sup>2</sup> C bus is released (bus free state). 1: The I <sup>2</sup> C bus is occupied (bus busy state or in the bus free state).	R

Note 1. When the MTWP bit in ICMR1 is set to 1, the MST and TRS bits can be written to.

#### ST Bit (Start Condition Issuance Request)

This bit is used to request transition to master mode and issuance of a start condition.

When this bit is set to 1 to request to issue a start condition, a start condition is issued when the BBSY flag is set to 0 (bus free).

For details on the start condition issuance, see section 33.10, Start Condition/Restart Condition/Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the ST bit

[Clearing conditions]

- When 0 is written to the ST bit
- When a start condition has been issued
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Note: • Set the ST bit to 1 (start condition issuance request) when the BBSY flag is set to 0 (bus free).

Note that arbitration may be lost if the ST bit is set to 1 (start condition issuance request) when the BBSY flag is set to 1 (bus busy).



**RS Bit (Restart Condition Issuance Request)**

This bit is used to request that a restart condition be issued in master mode.

When this bit is set to 1 to request to issue a restart condition, a restart condition is issued when the BBSY flag is set to 1 (bus busy) and the MST bit is set to 1 (master mode).

For details on the restart condition issuance, see section 33.10, Start Condition/Restart Condition/Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the RS bit with the BBSY flag in ICCR2 set to 1

[Clearing conditions]

- When 0 is written to the RS bit
- When a restart condition has been issued or a start condition is detected
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Note: • Do not set the RS bit to 1 while issuing a stop condition.

Note: • If the RS bit is set to 1 (restart condition issuance request) in mode other than master mode, the restart condition is not issued in this mode but the restart condition issuance request bit remains set. If the operating mode changes to master mode with the bit not being cleared, the restart condition may be issued.

**SP Bit (Stop Condition Issuance Request)**

This bit is used to request that a stop condition be issued in master mode.

When this bit is set to 1 to request to issue a stop condition, a stop condition is issued when the BBSY flag is set to 1 (bus busy) and the MST bit is set to 1 (master mode).

For details on the stop condition issuance, see section 33.10, Start Condition/Restart Condition/Stop Condition Issuing Function.

[Setting condition]

- When 1 is written to the SP bit with both the BBSY flag and the MST bit in ICCR2 set to 1

[Clearing conditions]

- When 0 is written to the SP bit
- When a stop condition has been issued or a stop condition is detected
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When a start condition and a restart condition are detected
- When 1 is written to the IICRST bit in ICCR1 to apply an 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).

Note: • Do not set the SP bit to 1 while a restart condition is being issued.

**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 the value for transmission mode or reception mode by detection or issuing of a start condition, setting or clearing of the R/W# bit, etc. Although writing to the TRS bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not necessary during normal usage.

[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST 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 ICSEER, with the R/W# bit set to 1
- When 1 is written to the TRS bit with the MTWP bit in ICMR1 set to 1

[Clearing conditions]

- When a stop condition is detected
- The AL (arbitration-lost) flag in ICSR2 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 ICSEER 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 (a start condition is detected with ICCR2.BBSY = 1 and ICCR2.MST = 0)
- When 0 is written to the TRS bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an 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 the value for master mode or slave mode by detection or issuing of a start condition, etc. Although writing to the MST bit is possible when the MTWP bit in ICMR1 is set to 1, writing to this bit is not necessary during normal usage.

[Setting conditions]

- When a start condition is issued normally according to the start condition issuance request (when a start condition is detected with the ST bit set to 1)
- When 1 is written to the MST bit with the MTWP bit in ICMR1 set to 1

[Clearing conditions]

- When a stop condition is detected
- When the AL (arbitration-lost) flag in ICSR2 is set to 1
- When 0 is written to the MST bit with the MTWP bit in ICMR1 set to 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

### BBSY Flag (Bus Busy Detection)

The BBSY flag indicates whether the I<sup>2</sup>C bus is occupied (bus busy) or released (bus free).

This bit is set to 1 when the SDAn line changes from high to low under the condition of SCLn = high, assuming that a start condition has been issued.

When the SDAn line changes from low to high under the condition of SCLn = high, this bit is cleared to 0 after the bus free time (specified in ICBRL) start condition is not detected, assuming that a stop condition has been issued.

[Setting condition]

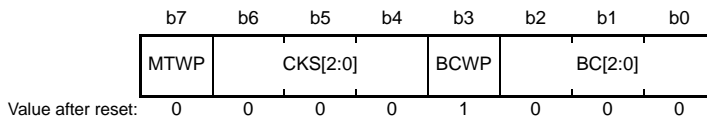
- When a start condition is detected

[Clearing conditions]

- When the bus free time (specified in ICBRL) start condition is not detected after detecting a stop condition
- When 1 is written to the IICRST bit in ICCR1 with the ICE bit in ICCR1 set to 0 (RIIC reset)

### 33.2.3 I<sup>2</sup>C Bus Mode Register 1 (ICMR1)

Address(es): RIIC0.ICMR1 0008 8302h, RIIC1.ICMR1 0008 8322h, RIIC2.ICMR1 0008 8342h, RIIC3.ICMR1 0008 8362h



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 (IIC $\phi$ ) Selection	b6 b4 0 0 0: PCLK/1 clock 0 0 1: PCLK/2 clock 0 1 0: PCLK/4 clock 0 1 1: PCLK/8 clock 1 0 0: PCLK/16 clock 1 0 1: PCLK/32 clock 1 1 0: PCLK/64 clock 1 1 1: PCLK/128 clock	R/W
b7	MTWP	MST/TRS Write Protect	0: Disables writing to the MST and TRS bits in ICCR2. 1: Enables writing to the MST and TRS bits in ICCR2.	R/W

Note 1. Set the BCWP bit to 0 to rewrite the BC[2:0] bits. The BC[2:0] bits must be rewritten by using the MOV instruction.

#### BC[2:0] Bits (Bit Counter)

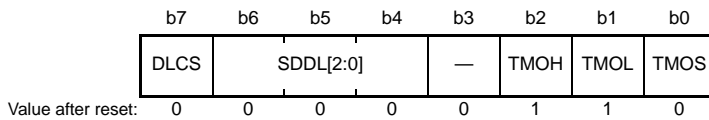
These bits function as a counter that indicates the number of bits remaining to be transferred at the detection of a rising edge on the SCLn line. Although these bits are writable and readable, it is not necessary to access these bits under normal conditions.

To write to these bits, specify the number of bits to be transferred plus one (data is transferred with an additional acknowledge bit) between transferred frames when the SCLn line is at a low level.

The values of the BC[2:0] bits return to 000b at the end of a data transfer including the acknowledge bit or when a start condition including a restart condition is detected.

### 33.2.4 I<sup>2</sup>C Bus Mode Register 2 (ICMR2)

Address(es): RIIC0.ICMR2 0008 8303h, RIIC1.ICMR2 0008 8323h, RIIC2.ICMR2 0008 8343h, RIIC3.ICMR2 0008 8363h



Bit	Symbol	Bit Name	Description	R/W																																																						
b0	TMOS	Timeout Detection Time Selection	0: Long mode is selected. 1: Short mode is selected.	R/W																																																						
b1	TMOL	Timeout L Count Control	0: Count is disabled while the SCLn line is at a low level. 1: Count is enabled while the SCLn line is at a low level.	R/W																																																						
b2	TMOH	Timeout H Count Control	0: Count is disabled while the SCLn line is at a high level. 1: Count is enabled while the SCLn line is at a high level.	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	<ul style="list-style-type: none"> <li>• When ICMR2.DLCS = 0 (IIC<math>\phi</math>)               <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 10px;">b6</td> <td style="padding-right: 10px;">b4</td> <td></td> </tr> <tr> <td>0 0</td> <td>0</td> <td>No output delay</td> </tr> <tr> <td>0 0</td> <td>1</td> <td>1 IIC<math>\phi</math> cycle</td> </tr> <tr> <td>0 1</td> <td>0</td> <td>2 IIC<math>\phi</math> cycles</td> </tr> <tr> <td>0 1</td> <td>1</td> <td>3 IIC<math>\phi</math> cycles</td> </tr> <tr> <td>1 0</td> <td>0</td> <td>4 IIC<math>\phi</math> cycles</td> </tr> <tr> <td>1 0</td> <td>1</td> <td>5 IIC<math>\phi</math> cycles</td> </tr> <tr> <td>1 1</td> <td>0</td> <td>6 IIC<math>\phi</math> cycles</td> </tr> <tr> <td>1 1</td> <td>1</td> <td>7 IIC<math>\phi</math> cycles</td> </tr> </table> </li> <li>• When ICMR2.DLCS = 1 (IIC<math>\phi</math>/2)               <table border="0" style="margin-left: 20px;"> <tr> <td style="padding-right: 10px;">b6</td> <td style="padding-right: 10px;">b4</td> <td></td> </tr> <tr> <td>0 0</td> <td>0</td> <td>No output delay</td> </tr> <tr> <td>0 0</td> <td>1</td> <td>1 or 2 IIC<math>\phi</math> cycles</td> </tr> <tr> <td>0 1</td> <td>0</td> <td>3 or 4 IIC<math>\phi</math> cycles</td> </tr> <tr> <td>0 1</td> <td>1</td> <td>5 or 6 IIC<math>\phi</math> cycles</td> </tr> <tr> <td>1 0</td> <td>0</td> <td>7 or 8 IIC<math>\phi</math> cycles</td> </tr> <tr> <td>1 0</td> <td>1</td> <td>9 or 10 IIC<math>\phi</math> cycles</td> </tr> <tr> <td>1 1</td> <td>0</td> <td>11 or 12 IIC<math>\phi</math> cycles</td> </tr> <tr> <td>1 1</td> <td>1</td> <td>13 or 14 IIC<math>\phi</math> cycles</td> </tr> </table> </li> </ul>	b6	b4		0 0	0	No output delay	0 0	1	1 IIC $\phi$ cycle	0 1	0	2 IIC $\phi$ cycles	0 1	1	3 IIC $\phi$ cycles	1 0	0	4 IIC $\phi$ cycles	1 0	1	5 IIC $\phi$ cycles	1 1	0	6 IIC $\phi$ cycles	1 1	1	7 IIC $\phi$ cycles	b6	b4		0 0	0	No output delay	0 0	1	1 or 2 IIC $\phi$ cycles	0 1	0	3 or 4 IIC $\phi$ cycles	0 1	1	5 or 6 IIC $\phi$ cycles	1 0	0	7 or 8 IIC $\phi$ cycles	1 0	1	9 or 10 IIC $\phi$ cycles	1 1	0	11 or 12 IIC $\phi$ cycles	1 1	1	13 or 14 IIC $\phi$ cycles	R/W
b6	b4																																																									
0 0	0	No output delay																																																								
0 0	1	1 IIC $\phi$ cycle																																																								
0 1	0	2 IIC $\phi$ cycles																																																								
0 1	1	3 IIC $\phi$ cycles																																																								
1 0	0	4 IIC $\phi$ cycles																																																								
1 0	1	5 IIC $\phi$ cycles																																																								
1 1	0	6 IIC $\phi$ cycles																																																								
1 1	1	7 IIC $\phi$ cycles																																																								
b6	b4																																																									
0 0	0	No output delay																																																								
0 0	1	1 or 2 IIC $\phi$ cycles																																																								
0 1	0	3 or 4 IIC $\phi$ cycles																																																								
0 1	1	5 or 6 IIC $\phi$ cycles																																																								
1 0	0	7 or 8 IIC $\phi$ cycles																																																								
1 0	1	9 or 10 IIC $\phi$ cycles																																																								
1 1	0	11 or 12 IIC $\phi$ cycles																																																								
1 1	1	13 or 14 IIC $\phi$ cycles																																																								
b7	DLCS	SDA Output Delay Clock Source Selection	0: The internal reference clock (IIC $\phi$ ) is selected as the clock source of the SDA output delay counter. 1: The internal reference clock divided by 2 (IIC $\phi$ /2) is selected as the clock source of the SDA output delay counter.*1	R/W																																																						

Note 1. The setting DLCS = 1 (IIC $\phi$ /2) only becomes valid when SCL is at the low level. When SCL is at the high level, the setting DLCS = 1 becomes invalid and the clock source becomes the internal reference clock (IIC $\phi$ ).

#### TMOS Bit (Timeout Detection Time Selection)

This bit is used to select long mode or short mode for the timeout detection time when the timeout function is enabled (TMOE bit = 1 in ICFER). 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 SCLn 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 (IIC $\phi$ ) as a count source. For details on the timeout function, see section 33.11.1, 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 SCLn line is held low when the timeout function is enabled (TMOE bit = 1 in ICFER).

**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 SCL<sub>n</sub> line is held high when the timeout function is enabled (TMOE bit = 1 in ICFER).

**SDDL[2:0] Bits (SDA Output Delay Setup 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 acknowledge bit.

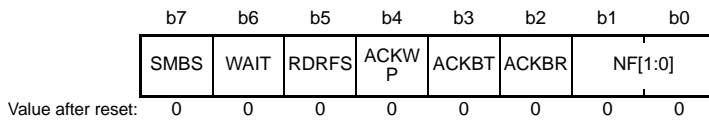
For details on this function, see section 33.5, Facility for Delaying SDA Output.

Note: • Set the SDA output delay time to meet the I<sup>2</sup>C bus standard (within the data enable time/acknowledge enable time\*<sup>1</sup>) or the SMBus standard (within the data hold time: 300 ns or more, and SCL-clock low-level period - the data setup time: 250 ns). Note that, if a value outside the standard is set, communication with communication devices may malfunction or it may seemingly become a start condition or stop condition depending on the bus state.

Note 1. Data enable time/acknowledge enable time  
3,450 ns (up to 100 kbps: standard mode [Sm])  
900 ns (up to 400 kbps: fast mode [fm])  
450 ns (up to 1 Mbps: fast mode plus [fm+])

### 33.2.5 I<sup>2</sup>C Bus Mode Register 3 (ICMR3)

Address(es): RIIC0.ICMR3 0008 8304h, RIIC1.ICMR3 0008 8324h, RIIC2.ICMR3 0008 8344h, RIIC3.ICMR3 0008 8364h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	NF[1:0]	Noise Filter Stage Selection	b1 b0 0 0: Noise of up to one IIC $\phi$ cycle is filtered out (single-stage filter). 0 1: Noise of up to two IIC $\phi$ cycles is filtered out (2-stage filter). 1 0: Noise of up to three IIC $\phi$ cycles is filtered out (3-stage filter). 1 1: Noise of up to four IIC $\phi$ cycles is filtered out (4-stage filter).	R/W
b2	ACKBR	Receive Acknowledge	0: A 0 is received as the acknowledge bit (ACK reception). 1: A 1 is received as the acknowledge bit (NACK reception).	R
b3	ACKBT	Transmit Acknowledge	0: A 0 is sent as the acknowledge bit (ACK transmission). 1: A 1 is sent as the acknowledge 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.	W*1
b5	RDRFS	RDRF Flag Set Timing Selection	0: The RDRF flag is set at the rising edge of the ninth SCL clock cycle. (The SCLn line is not held low at the falling edge of the eighth clock cycle.) 1: The RDRF flag is set at the rising edge of the eighth SCL clock cycle. (The SCLn line is held low at the falling edge of the eighth clock cycle.) 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 cycle and first clock cycle is not held low.) 1: WAIT (The period between ninth clock cycle and first clock cycle is held low.) Low-hold is released by reading ICDRR.	R/W*2
b7	SMBS	SMBus/I <sup>2</sup> C Bus Selection	0: The I <sup>2</sup> C bus is selected. 1: The SMBus is selected.	R/W

Note 1. If it is attempted to write 1 to both ACKWP and ACKBT bits, the ACKBT bit cannot 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 Selection)

These bits are used to select the number of stages of the digital noise filter.

Note: • Set the noise range to be filtered out by the noise filter within a range less than the SCLn line high-level period or low-level period. If the noise range is set to a value of (SCL clock width: high-level period or low-level period, whichever is shorter) - [1.5 internal reference clock (IIC $\phi$ ) cycles + analog noise filter: 120 ns (reference values)] or more, the SCL clock is regarded as noise by the noise filter function of the RIIC, which may prevent the RIIC from operating normally.

#### ACKBR Bit (Receive Acknowledge)

This bit is used to store the acknowledge bit information received from the receive device in transmit mode.

[Setting condition]

- When 1 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1

[Clearing conditions]

- When 0 is received as the acknowledge bit with the TRS bit in ICCR2 set to 1
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (RIIC reset)

**ACKBT Bit (Transmit Acknowledge)**

This bit is used to set the bit to be sent at the acknowledge 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 issuance is detected (when a stop condition is detected with the SP bit in ICCR2 set to 1)
- When 1 is written to the IICRST bit in ICCR1 while the ICE bit in ICCR1 is 0 (RIIC reset)

Note: • The ACKBT bit must be written to while the ACKWP bit is 1. If the ACKBT bit is written to with the ACKWP bit cleared to 0, writing to the ACKBT bit is disabled.

**ACKWP Bit (ACKBT Write Protect)**

This bit is used to control the modification of the ACKBT bit.

**RDRFS Bit (RDRF Flag Set Timing Selection)**

This bit is used to select the RDRF flag set timing in receive mode and also to select whether to hold the SCLn line low at the falling edge of the eighth SCL clock cycle.

When the RDRFS bit is 0, the SCLn line is not held low at the falling edge of the eighth SCL clock cycle, and the RDRF flag is set to 1 at the rising edge of the ninth SCL clock cycle.

When the RDRFS bit is 1, the RDRF flag is set to 1 at the rising edge of the eighth SCL clock cycle and the SCLn line is held low at the falling edge of the eighth SCL clock cycle. The low-hold of the SCLn line is released by writing a value to the ACKBT bit.

After data is received with this setting, the SCLn line is automatically held low before the acknowledge bit is sent. This enables processing to send ACK (ACKBT = 0) or NACK (ACKBT = 1) according to receive data.

**WAIT Bit (WAIT)**

This bit is used to control whether to hold the period between the ninth SCL clock cycle and the first SCL clock cycle low until the receive data buffer (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 clock cycle low. When both the RDRFS and WAIT bits are 0, continuous receive operation is enabled with the double buffer.

When the WAIT bit is 1, the SCLn line is held low from the falling edge of the ninth clock cycle until the ICDRR 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 beforehand.

**SMBS Bit (SMBus/I<sup>2</sup>C Bus Selection)**

Setting this bit to 1 selects the SMBus and enables the HOAE bit in IC SER.

### 33.2.6 I<sup>2</sup>C Bus Function Enable Register (ICFER)

Address(es): RIIC0.ICFER 0008 8305h, RIIC1.ICFER 0008 8325h, RIIC2.ICFER 0008 8345h, RIIC3.ICFER 0008 8365h

b7	b6	b5	b4	b3	b2	b1	b0	
FMPE	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 MST and TRS bits in ICCR2 automatically when arbitration is lost.) 1: Master arbitration-lost detection is enabled. (Enables the arbitration-lost detection function and clears the MST and TRS bits in ICCR2 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 Circuit Enable	0: No digital noise filter circuit is used. 1: A digital noise filter circuit is used.	R/W
b6	SCLE	SCL Synchronous Circuit Enable	0: No SCL synchronous circuit is used. 1: An SCL synchronous circuit is used.	R/W
b7	FMPE*1	Fast-mode Plus Enable	0: No fm+ slope control circuit is used for the SCLn pin and SDAn pin. 1: An fm+ slope control circuit is used for the SCLn pin and SDAn pin.	R/W

Note 1. The fast-mode plus enable bit (FMPE) is only supported by RIIC0. In RIIC1 to RIIC3, bit 7 is reserved.

#### TMOE Bit (Timeout Function Enable)

This bit is used to enable or disable the timeout function.

For details on the timeout function, see section 33.11.1, 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 transfer operation when NACK is received from the slave device 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 received acknowledge content.

For details on the NACK reception transfer suspension function, see section 33.8.2, NACK Reception Transfer Suspension Function.

**SCLE Bit (SCL Synchronous Circuit Enable)**

This bit is used to specify whether to synchronize the SCL clock with the SCL input clock. Normally, set this bit to 1.

When the SCLE bit is cleared to 0 (SCL synchronous circuit not used), the RIIC does not synchronize the SCL clock with the SCL input clock. In this setting, the RIIC outputs the SCL clock with the transfer rate set in ICBRH and ICBRL regardless of the SCLn line state. For this reason, if the bus load of the I<sup>2</sup>C bus line is much larger than the specification value or if the SCL clock output overlaps in multiple masters, the short-cycle SCL clock that does not meet the specification may be output. When no SCL synchronous circuit is used, it also affects the issuance of a start condition, restart condition, and stop condition, and the continuous output of extra SCL clock cycles.

This bit must not be cleared to 0 except for checking the output of the transfer rate.

**FMPE Bit (Fast-mode Plus Enable)**

This bit is used to specify whether to use a slope control circuit for Fast-mode Plus[fm+].

When this bit is set to 1, a slope control circuit conforming to the Fast-mode Plus[fm+] slope control standard (tof) of the I<sup>2</sup>C bus is selected. When this bit is cleared to 0, a slope control circuit conforming to the Standard-mode[Sm] and Fast-mode[fm] slope control standard (tof) of the I<sup>2</sup>C bus is selected.

Set this bit to 1 when using the transmission rate within a range up to 1 Mbps (Fast-mode Plus[fm+]) of the I<sup>2</sup>C bus standard. Clear this bit to 0 when using the transmission rate at other rates (up to 100 kbps[Sm], up to 400 kbps[fm]) or for SMBus (10 to 100 kbps).

### 33.2.7 I<sup>2</sup>C Bus Status Enable Register (ICSER)

Address(es): RIIC0.ICSER 0008 8306h, RIIC1.ICSER 0008 8326h, RIIC2.ICSER 0008 8346h, RIIC3.ICSER 0008 8366h

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 SARL0 and SARU0 is disabled. 1: Slave address in SARL0 and SARU0 is enabled.	R/W
b1	SAR1E	Slave Address Register 1 Enable	0: Slave address in SARL1 and SARU1 is disabled. 1: Slave address in SARL1 and SARU1 is enabled.	R/W
b2	SAR2E	Slave Address Register 2 Enable	0: Slave address in SARL2 and SARU2 is disabled. 1: Slave address in 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 received slave address and the slave address set in SARLy and SARUy.

When this bit is set to 1, the slave address set in SARLy and SARUy is enabled and is compared with the received slave address.

When this bit is cleared to 0, the slave address set in SARLy and SARUy is disabled and is ignored even if it matches the received slave address.

#### GCAE Bit (General Call Address Enable)

This bit is used to specify whether to ignore the general call address (0000 000b + 0 [W]: All 0) when it is received.

When this bit is set to 1, if the received slave address matches the general call address, the RIIC recognizes the received slave address as the general call address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2) and performs data receive operation.

When this bit is cleared 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 frame after a start condition or restart condition is detected.

When this bit is set to 1, if the received first frame matches the device ID, the RIIC recognizes that the Device-ID address has been received. When the following R/W# bit is 0 [W], the RIIC recognizes the second and the following frames as slave addresses and continues the receive operation.

When this bit is cleared to 0, the RIIC ignores the received first frame even if it matches the device ID address and recognizes the first frame as a normal slave address.

For details on the device-ID address detection, see section 33.7.3, 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 SMBS bit in ICMR3 is 1. When this bit is set to 1 while the SMBS bit in ICMR3 is 1, if the received slave address matches the host address, the RIIC recognizes the received slave address as the host address independently of the slave addresses set in SARLy and SARUy (y = 0 to 2) and performs the receive operation.

When the SMBS bit in ICMR3 or the HOAE bit is cleared to 0, the received slave address is ignored even if it matches the host address.

**33.2.8 I<sup>2</sup>C Bus Interrupt Enable Register (ICIER)**

Address(es): RIIC0.ICIER 0008 8307h, RIIC1.ICIER 0008 8327h, RIIC2.ICIER 0008 8347h, RIIC3.ICIER 0008 8367h

	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 Enable	0: Timeout interrupt request (TMOI) is disabled. 1: Timeout interrupt request (TMOI) is enabled.	R/W
b1	ALIE	Arbitration-Lost Interrupt Enable	0: Arbitration-lost interrupt request (ALI) is disabled. 1: Arbitration-lost interrupt request (ALI) is enabled.	R/W
b2	STIE	Start Condition Detection Interrupt Enable	0: Start condition detection interrupt request (STI) is disabled. 1: Start condition detection interrupt request (STI) is enabled.	R/W
b3	SPIE	Stop Condition Detection Interrupt Enable	0: Stop condition detection interrupt request (SPI) is disabled. 1: Stop condition detection interrupt request (SPI) is enabled.	R/W
b4	NAKIE	NACK Reception Interrupt Enable	0: NACK reception interrupt request (NAKI) is disabled. 1: NACK reception interrupt request (NAKI) is enabled.	R/W
b5	RIE	Receive Data Full Interrupt Enable	0: Receive data full interrupt request (ICRXI) is disabled. 1: Receive data full interrupt request (ICRXI) is enabled.	R/W
b6	TEIE	Transmit End Interrupt Enable	0: Transmit end interrupt request (ICTEI) is disabled. 1: Transmit end interrupt request (ICTEI) is enabled.	R/W
b7	TIE	Transmit Data Empty Interrupt Enable	0: Transmit data empty interrupt request (ICTXI) is disabled. 1: Transmit data empty interrupt request (ICTXI) is enabled.	R/W

**TMOIE Bit (Timeout Interrupt Enable)**

This bit is used to enable or disable timeout interrupt requests (TMOI) when the TMOF flag in ICSR2 is set to 1. A TMOI interrupt request is canceled by clearing the TMOF flag or the TMOIE bit to 0.

**ALIE Bit (Arbitration-Lost Interrupt Enable)**

This bit is used to enable or disable arbitration-lost interrupt requests (ALI) when the AL flag in ICSR2 is set to 1. An ALI interrupt request is canceled by clearing the AL flag or the ALIE bit to 0.

**STIE Bit (Start Condition Detection Interrupt Enable)**

This bit is used to enable or disable start condition detection interrupt requests (STI) when the START flag in ICSR2 is set to 1. An STI interrupt request is canceled by clearing the START flag or the STIE bit to 0.

**SPIE Bit (Stop Condition Detection Interrupt Enable)**

This bit is used to enable or disable stop condition detection interrupt requests (SPI) when the STOP flag in ICSR2 is set to 1. An SPI interrupt request is canceled by clearing the STOP flag or the SPIE bit to 0.

**NAKIE Bit (NACK Reception Interrupt Enable)**

This bit is used to enable or disable NACK reception interrupt requests (NAKI) when the NACKF flag in ICSR2 is set to 1. An NAKI interrupt request is canceled by clearing the NACKF flag or the NAKIE bit to 0.

**RIE Bit (Receive Data Full Interrupt Enable)**

This bit is used to enable or disable receive data full interrupt requests (ICRXI) when the RDRF flag in ICSR2 is set to 1.

**TEIE Bit (Transmit End Interrupt Enable)**

This bit is used to enable or disable transmit end interrupts (ICTEI) when the TEND flag in ICSR2 is set to 1. An ICTEI interrupt request is canceled by clearing the TEND flag or the TEIE bit to 0.

**TIE Bit (Transmit Data Empty Interrupt Enable)**

This bit is used to enable or disable transmit data empty interrupts (ICTXI) when the TDRE flag in ICSR2 is set to 1.

**33.2.9 I<sup>2</sup>C Bus Status Register 1 (ICSR1)**

Address(es): RIIC0.ICSR1 0008 8308h, RIIC1.ICSR1 0008 8328h, RIIC2.ICSR1 0008 8348h, RIIC3.ICSR1 0008 8368h

b7	b6	b5	b4	b3	b2	b1	b0
HOA	—	DID	—	GCA	AAS2	AAS1	AAS0
0	0	0	0	0	0	0	0

Value after reset:

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. <ul style="list-style-type: none"> <li>This bit is set to 1 when the received slave address matches the SVA[6:0] value in SARL0 while the FS bit in SARU0 is 0 (7-bit address format selected).</li> <li>This bit is set to 1 when the received slave address matches a value of (11110b + SVA[1:0] in SARU0) and the following address matches the SARL0 value while the FS bit in SARU0 is 1 (10-bit address format selected).</li> </ul> (This bit is set at the rising edge of the ninth SCL clock cycle in the SARL0 match determination frame.)	R(W) *1
b1	AAS1	Slave Address 1 Detection Flag	0: Slave address 1 is not detected. 1: Slave address 1 is detected. <ul style="list-style-type: none"> <li>This bit is set to 1 when the received slave address matches the SVA[6:0] value in SARL1 while the FS bit in SARU1 is 0 (7-bit address format selected).</li> <li>This bit is set to 1 when the received slave address matches a value of (11110b + SVA[1:0] in SARU1) and the following address matches the SARL1 value while the FS bit in SARU1 is 1 (10-bit address format selected).</li> </ul> (This bit is set at the rising edge of the ninth SCL clock cycle in the SARL1 match determination frame.)	R(W) *1
b2	AAS2	Slave Address 2 Detection Flag	0: Slave address 2 is not detected. 1: Slave address 2 is detected. <ul style="list-style-type: none"> <li>This bit is set to 1 when the received slave address matches the SVA[6:0] value in SARL2 while the FS bit in SARU2 is 0 (7-bit address format selected).</li> <li>This bit is set to 1 when the received slave address matches a value of (11110b + SVA[1:0] in SARU2) and the following address matches the SARL2 value while the FS bit in SARU2 is 1 (10-bit address format selected).</li> </ul> (This bit is set at the rising edge of the ninth SCL clock cycle in the SARL2 match determination frame.)	R(W) *1

Bit	Symbol	Bit Name	Description	R/W
b3	GCA	General Call Address Detection Flag	0: General call address is not detected. 1: General call address is detected. • This bit is set to 1 when the received slave address matches the general call address (all 0).	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 frame received immediately after a start condition is detected matches a value of (device ID (1111 100b) + 0[W]).	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) (y = 0 to 2)

[Setting conditions]

<For 7-bit address format: SARUy.FS = 0>

- When the received slave address matches the SVA[6:0] value in SARLy with the SARyE bit in ICSEr set to 1 (slave address y detection enabled)

This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

<For 10-bit address format: SARUy.FS = 1>

- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy) and the following address matches the SARLy value with the SARyE bit in ICSEr set to 1 (slave address y detection enabled)

This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the AASy bit after reading AASy = 1
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

<For 7-bit address format: SARUy.FS = 0>

- When the received slave address does not match the SVA[6:0] value in SARLy with the SARyE bit in ICSEr set to 1 (slave address y detection enabled)

This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

<For 10-bit address format: SARUy.FS = 1>

- When the received slave address does not match a value of (11110b + SVA[1:0] in SARUy) with the SARyE bit in ICSEr set to 1 (slave address y detection enabled)

This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

- When the received slave address matches a value of (11110b + SVA[1:0] in SARUy) and the following address does not match the SARLy value with the SARyE bit in ICSEr set to 1 (slave address y detection enabled)

This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.

### GCA Flag (General Call Address Detection)

[Setting condition]

- When the received slave address matches the general call address (0000 000b + 0 [W]) with the GCAE bit in ICSEr set to 1 (general call address detection enabled)

This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the GCA bit after reading GCA = 1
- When a stop condition is detected
- When the received slave address does not match the general call address (0000 000b + 0 [W]) with the GCAE bit in IC SER set to 1 (general call address detection enabled)  
This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

### DID Flag (Device-ID Address Detection)

[Setting condition]

- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]) with the DIDE bit in IC SER set to 1 (Device-ID address detection enabled)  
This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the DID bit after reading DID = 1
- When a stop condition is detected
- When the first frame received immediately after a start condition or restart condition is detected does not match a value of (device ID (1111 100b)) with the DIDE bit in IC SER set to 1 (Device-ID address detection enabled)  
This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When the first frame received immediately after a start condition or restart condition is detected matches a value of (device ID (1111 100b) + 0 [W]) and the second frame does not match any of slave addresses 0 to 2 with the DIDE bit in IC SER set to 1 (Device-ID address detection enabled)  
This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

### HOA Flag (Host Address Detection)

[Setting condition]

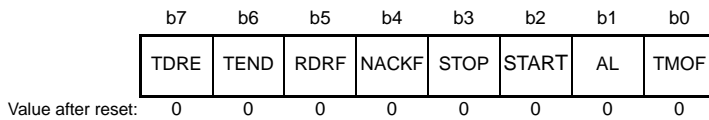
- When the received slave address matches the host address (0001 000b) with the HOAE bit in IC SER set to 1 (host address detection enabled)  
This flag is set to 1 at the rising edge of the ninth SCL clock cycle in the frame.

[Clearing conditions]

- When 0 is written to the HOA bit after reading HOA = 1
- When a stop condition is detected
- When 0 is written to the SMBS bit in ICMR3 or the HOAE bit in IC SER
- When the received slave address does not match the host address (0001 000b) with the HOAE bit in IC SER set to 1 (host address detection enabled)  
This flag is cleared to 0 at the rising edge of the ninth SCL clock cycle in the frame.
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

### 33.2.10 I<sup>2</sup>C Bus Status Register 2 (ICSR2)

Address(es): RIIC0.ICSR2 0008 8309h, RIIC1.ICSR2 0008 8329h, RIIC2.ICSR2 0008 8349h, RIIC3.ICSR2 0008 8369h



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: ICDRR contains no receive data. 1: ICDRR 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: ICDRT contains transmit data. 1: ICDRT contains no transmit data.	R

Note 1. Only 0 can be written to clear the flag.

#### TMOF Flag (Timeout Detection)

This flag is set to 1 when the RIIC recognizes timeout after the SCLn line state remains unchanged for a certain period.  
[Setting condition]

- When the SCLn line state remains unchanged for the period specified by bits TMOH, TMOL, and TMOS in ICMR2 with the TMOE bit in ICFER set to 1 (timeout detection enabled) in master mode or in the slave specification state.

[Clearing conditions]

- When 0 is written to the TMOF bit after reading TMOF = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

#### AL Flag (Arbitration-Lost)

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 issued or an address and data are transmitted. The RIIC monitors the level on the SDAn 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 bit to 1 to indicate that the bus is occupied by another device.

The RIIC can also set the flag to indicate the detection of loss of arbitration during NACK transmission in master mode or during data transmission in slave mode.

[Setting conditions]

<When master arbitration-lost detection is enabled: ICFER.MALE = 1>

- When the internal SDA output state does not match the SDAn line level at the rising edge of SCL clock except for the ACK period during data (including slave address) transmission in master transmit mode (when the SDAn line is driven low while the internal SDA output is at a high level (the SDAn pin is in the high-impedance state))
- When a start condition is detected while the ST bit in ICCR2 is 1 (start condition issuance request) or the internal SDA output state does not match the SDAn line level
- When the ST bit in ICCR2 is set to 1 (start condition issuance request) with the BBSY flag in ICCR2 set to 1.

<When NACK arbitration-lost detection is enabled: ICFER.NALE = 1>

- When the internal SDA output state does not match the SDAn line level at the rising edge of SCL clock in the ACK period during NACK transmission in receive mode

<When slave arbitration-lost detection is enabled: ICFER.SALE = 1>

- When the internal SDA output state does not match the SDAn line level at the rising edge of SCL clock except for the ACK period during data transmission in slave transmit mode

[Clearing conditions]

- When 0 is written to the AL bit after reading AL = 1
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

**Table 33.4 Relationship between Arbitration-Lost Generation Sources and Arbitration-Lost Enable Functions**

ICFER			ICSR2	Error	Arbitration-Lost Generation Source
MALE	NALE	SALE	AL		
1	x	x	1	Start condition issuance error	When internal SDA output state does not match SDAn line level when a start condition is detected while the ST bit in ICCR2 is 1 When ST in ICCR2 is set to 1 with BBSY in ICCR2 set to 1
			1	Transmit data mismatch	When transmit data (including slave address) does not match the bus state in master transmit mode
x	1	x	1	NACK transmission mismatch	When ACK is detected during transmission of NACK in master receive mode or slave receive mode
x	x	1	1	Transmit data mismatch	When transmit data does not match the bus state in slave transmit mode

x: Don't care

### START Flag (Start Condition Detection)

[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 IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

### STOP Flag (Stop Condition Detection)

[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 IICRST bit in ICCR1 to apply an RIIC reset or an internal reset



**NACKF Flag (NACK Detection)**

[Setting condition]

- When acknowledge is not received (NACK is received) from the receive device in transmit mode with the NACKEN bit in ICFER 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 IICRST bit in ICCR1 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 ICDRT in transmit mode or reading from ICDRR in receive mode with the NACKF flag set to 1 does not enable data transmit/receive operation. To restart data transmission/reception, clear the NACKF flag to 0.

**RDRF Flag (Receive Data Full)**

[Setting conditions]

- When receive data has been transferred from ICDRS to ICDRR  
This flag is set to 1 at the rising edge of the eighth or ninth SCL clock cycle (selected by the RDRFS bit in ICMR3)
- When the received slave address matches after a start condition (or a restart condition) is detected with the TRS bit in ICCR2 cleared to 0

[Clearing conditions]

- When 0 is written to the RDRF bit after reading RDRF = 1
- When data is read from ICDRR
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

**TEND Flag (Transmit End)**

[Setting condition]

- At the rising edge of the ninth SCL clock cycle 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 ICDRT
- When a stop condition is detected
- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

**TDRE Flag (Transmit Data Empty)**

[Setting conditions]

- When data has been transferred from ICDRT to ICDRS and ICDRT becomes empty
- When the TRS bit in ICCR2 is set to 1
  - When the MST bit in ICCR2 is set to 1 after a start condition (or a restart condition) is detected
  - When the RIIC enters transmit mode from receive mode
  - When 1 is written to while the ICMR1.MTWP bit is 1
- When the received slave address matches while the TRS bit is 1

[Clearing conditions]

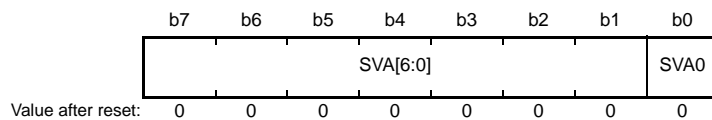
- When data is written to ICDRT
- When the TRS bit in ICCR2 is cleared to 0
  - When a stop condition is detected
  - When the RIIC enters receive mode from transmit mode
  - When 0 is written to while the ICMR1.MTWP bit is 1

- When 1 is written to the IICRST bit in ICCR1 to apply an RIIC reset or an internal reset

Note: • When the NACKF flag is set to 1 while the NACKE bit in ICFER is 1, the RIIC suspends data transmission/reception. Here, if the TDRE flag is 0 (next transmit data has been written), data is transferred to the ICDRS register and the ICDRT register becomes empty at the rising edge of the ninth clock cycle, but the TDRE flag is not set to 1.

### 33.2.11 Slave Address Register Ly (SARLy) (y = 0 to 2)

Address(es): RIIC0.SARL0 0008 830Ah, RIIC1.SARL0 0008 832Ah, RIIC2.SARL0 0008 834Ah, RIIC3.SARL0 0008 836Ah  
 RIIC0.SARL1 0008 830Ch, RIIC1.SARL1 0008 832Ch, RIIC2.SARL1 0008 834Ch, RIIC3.SARL1 0008 836Ch  
 RIIC0.SARL2 0008 830Eh, RIIC1.SARL2 0008 832Eh, RIIC2.SARL2 0008 834Eh, RIIC3.SARL2 0008 836Eh



Bit	Symbol	Bit Name	Description	R/W
b0	SVA0	10-Bit Address LSB	The least significant bit (LSB) of a 10-bit slave address is set. <ul style="list-style-type: none"> <li>• When the FS bit in SARUy is 0 (7-bit address format), this bit is invalid.</li> <li>• When the FS bit in SARUy is 1 (10-bit address format), this bit is the LSB of the lower 8-bit address (combined with the SVA[6:0] bits) of a 10-bit slave address.</li> </ul>	R/W
b7 to b1	SVA[6:0]	7-Bit Address/10-Bit Address Lower Bits	A slave address is set. <ul style="list-style-type: none"> <li>• When the FS bit in SARUy is 0 (7-bit address format), these bits form a 7-bit slave address.</li> <li>• When the FS bit in SARUy is 1 (10-bit address format), these bits form the lower 8-bit address (combined with the SVA0 bit) of a 10-bit slave address.</li> </ul>	R/W

#### SVA0 Bit (10-Bit Address LSB)

When the 10-bit address format is selected (SARUy.FS = 1), this bit functions as the LSB of a 10-bit address and forms the lower eight bits of a 10-bit address in combination with the SVA[6:0] bits.

When the SARyE bit in ICSEr 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.

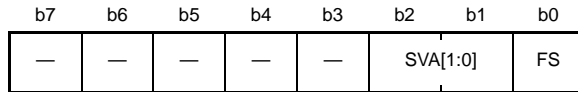
#### SVA[6:0] Bits (7-Bit Address/10-Bit Address Lower Bits)

When the 7-bit address format is selected (SARUy.FS = 0), these bits function as a 7-bit address. When the 10-bit address format is selected (SARUy.FS = 1), these bits function as the lower eight bits of a 10-bit address in combination with the SVA0 bit.

While the SARyE bit in ICSEr is 0, the setting of these bits is ignored.

### 33.2.12 Slave Address Register Uy (SARUy) (y = 0 to 2)

Address(es): RIIC0.SARU0 0008 830Bh, RIIC1.SARU0 0008 832Bh, RIIC2.SARU0 0008 834Bh, RIIC3.SARU0 0008 834Bh  
 RIIC0.SARU1 0008 830Dh, RIIC1.SARU1 0008 832Dh, RIIC2.SARU1 0008 834Dh, RIIC3.SARU1 0008 836Dh  
 RIIC0.SARU2 0008 830Fh, RIIC1.SARU2 0008 832Fh, RIIC2.SARU2 0008 834Fh, RIIC3.SARU2 0008 836Fh



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 Selection	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	A slave address is set. <ul style="list-style-type: none"> <li>• When the SARUy.FS bit is 0 (7-bit address format), these bits are invalid.</li> <li>• When the SARUy.FS bit is 1 (10-bit address format), these bits form the upper two bits of a 10-bit slave address.</li> </ul>	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 Selection)

This bit is used to select 7-bit address or 10-bit address for slave address y (in SARLy and SARUy).

When the SARyE bit in IC SER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 0, the 7-bit address format is selected for slave address y, the SVA[6:0] setting in SARLy is valid, and the settings of the SVA[1:0] bits and the SVA0 bit in SARLy are ignored.

When the SARyE bit in IC SER is set to 1 (SARLy and SARUy enabled) and the SARUy.FS bit is 1, the 10-bit address format is selected for slave address y and the settings of the SVA[1:0] bits and SARLy are valid.

While the SARyE bit in IC SER is 0 (SARLy and SARUy disabled), the setting of the SARUy.FS bit is invalid.

#### SVA[1:0] Bits (10-Bit Address Upper Bits)

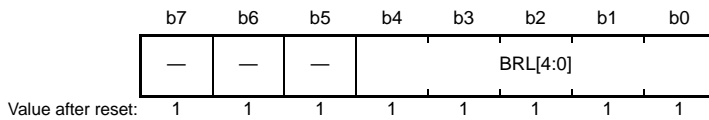
When the 10-bit address format is selected (FS = 1), these bits function as the upper two bits of a 10-bit address.

When the SARyE bit in IC SER 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.

### 33.2.13 I<sup>2</sup>C Bus Bit Rate Low-Level Register (ICBRL)

Address(es): RIIC0.ICBRL 0008 8310h, RIIC1.ICBRL 0008 8330h, RIIC2.ICBRL 0008 8350h, RIIC3.ICBRL 0008 8370h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BRL[4:0]	Bit Rate Low-Level Period	Low-level period of SCL clock	R/W
b7 to b5	—	Reserved	These bits are read as 1. The write value should be 1.	R/W

The ICBRL register is a 5-bit register that is used to set the width at low level for the SCL clock.

It also works to generate the data setup time for automatic SCL low-hold operation (see section 33.8, Automatically 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-level period with the internal reference clock source (IIC $\phi$ ) specified by the CKS[2:0] bits in ICMR1.

- Note 1. Data setup time (t<sub>SU</sub>: DAT)
- 250 ns (up to 100 kbps: standard mode [Sm])
  - 100 ns (up to 400 kbps: fast mode [fm])
  - 50 ns (up to 1 Mbps: fast mode plus [fm+])

### 33.2.14 I<sup>2</sup>C Bus Bit Rate High-Level Register (ICBRH)

Address(es): RIIC0.ICBRH 0008 8311h, RIIC1.ICBRH 0008 8331h, RIIC2.ICBRH 0008 8351h, RIIC3.ICBRH 0008 8371h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	BRH[4:0]	Bit Rate High-Level Period	High-level period of SCL clock	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-level period of SCL clock. ICBRH is valid in master mode. If the RIIC is used only in slave mode, this register need not to set the high-level period.

ICBRH counts the high-level period with the internal reference clock source ( $IIC\phi$ ) specified by the CKS[2:0] bits in ICMR1.

The I<sup>2</sup>C transfer rate and the SCL clock duty are calculated using the following expression.

Transfer rate =  $1 / \{ [(ICBRH + 1) + (ICBRL + 1)] / IIC\phi + SCLn \text{ line rising time } [tr] + SCLn \text{ line falling time } [tf] \}$

Duty cycle =  $\{ SCLn \text{ line rising time } [tr] + (ICBRH + 1) / IIC\phi \} / \{ SCLn \text{ line falling time } [tf] + (ICBRL + 1) / IIC\phi \}$

Note 1.  $IIC\phi = PCLK \times \text{Division ratio}$

Note 2. The SCLn line rising time [tr] and SCLn line falling time [tf] depend on the total bus line capacitance [Cb] and the pull-up resistor [Rp]. For details, see the I<sup>2</sup>C bus standard from NXP Semiconductors.

Table 33.5 lists examples of ICBRH/ICBRL settings.

**Table 33.5 Examples of ICBRH/ICBRL Settings for Transfer Rate (1)**

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)
1000	000b	2 (E2h)	3 (E3h)	000b	2 (E2h)	4 (E4h)	000b	3 (E3h)	6 (E6h)

**Table 33.5 Examples of ICBRH/ICBRL Settings for Transfer Rate (2)**

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)
1000	000b	4 (E4h)	7 (E7h)	000b	5 (E5h)	9 (E9h)	000b	6 (E6h)	12 (ECh)

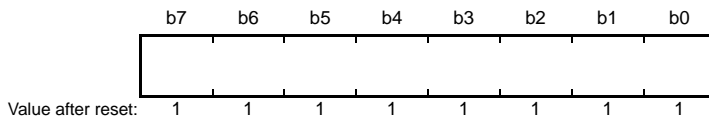
**Table 33.5 Examples of ICBRH/ICBRL Settings for Transfer Rate (3)**

Transfer Rate (kbps)	Operating Frequency PCLK (MHz)								
	30			33			50		
	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL	CKS[2:0]	ICBRH	ICBRL
10	110b	20 (F4h)	24 (F8h)	110b	22 (F6h)	26 (FAh)	111b	16 (F0h)	20 (F4h)
50	100b	15 (EFh)	18 (F2h)	100b	17 (F1h)	20 (F4h)	100b	26 (FAh)	31 (FFh)
100	010b	2 (E2h)	3 (E3h)	011b	16 (F0h)	19 (F3h)	011b	24 (F8h)	29 (FDh)
400	001b	8 (E8h)	19 (F3h)	001b	9 (E9h)	21 (F5h)	010b	7 (E7h)	16 (F0h)
1000	000b	7 (E7h)	14 (EEh)	000b	8 (E8h)	16 (F0h)	000b	12 (ECh)	24 (F8h)

Note: • CBRH/ICBRL settings in these tables are calculated using the following values:  
 SCLn line rising time (tr): 100 kbps or less, [Sm]: 1000 ns, 400 kbps or less, [Fm]: 300 ns, 1 Mbps or less, [Fm+]: 120 ns  
 SCLn line falling time (tf): 400 kbps or less, [Sm/Fm]: 300 ns, 1 Mbps or less, [Fm+]: 120 ns  
 For the specified values of SCLn line rising time (tr) and SCLn line falling time (tf), see the I<sup>2</sup>C bus standard from NXP Semiconductors.

### 33.2.15 I<sup>2</sup>C Bus Transmit Data Register (ICDRT)

Address(es): RIIC0.ICDRT 0008 8312h, RIIC1.ICDRT 0008 8332h, RIIC2.ICDRT 0008 8352h, RIIC3.ICDRT 0008 8372h



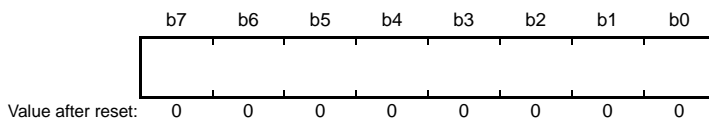
When ICDRT detects a space in the I<sup>2</sup>C bus shift register (ICDRS), it transfers the transmit data that has been written to ICDRT to ICDRS and starts transmitting data in transmit mode.

The double-buffer structure of ICDRT and ICDRS allows continuous transmit operation if the next transmit data has been written to ICDRT while the ICDRS data is being transmitted.

ICDRT can always be read and written. Write transmit data to ICDRT once when a transmit data empty interrupt (ICTXI) request is generated.

### 33.2.16 I<sup>2</sup>C Bus Receive Data Register (ICDRR)

Address(es): RIIC0.ICDRR 0008 8313h, RIIC1.ICDRR 0008 8333h, RIIC2.ICDRR 0008 8353h, RIIC3.ICDRR 0008 8373h



When 1 byte of data has been received, the received data is transferred from the I<sup>2</sup>C bus shift register (ICDRS) to ICDRR to enable the next data to be received.

The double-buffer structure of ICDRS and ICDRR allows continuous receive operation if the received data has been read from ICDRR while ICDRS is receiving data.

ICDRR cannot be written. Read data from ICDRR once when a receive data full interrupt (ICRXI) request is generated. If ICDRR receives the next receive data before the current data is read from ICDRR (while the RDRF flag in ICSR2 is 1), the RIIC automatically holds the SCLn clock low one cycle before the RDRF flag is set to 1 next.

### 33.2.17 I<sup>2</sup>C Bus Shift Register (ICDRS)



ICDRS is an 8-bit shift register to transmit and receive data.

During transmission, transmit data is transferred from ICDRT to ICDRS and is sent from the SDAn pin. During reception, data is transferred from ICDRS to ICDRR after 1 byte of data has been received.

ICDRS cannot be accessed directly.

### 33.3 Operation

#### 33.3.1 Communication Data Format

The I<sup>2</sup>C bus format consists of 8-bit data and 1-bit acknowledge. The frame following a start condition or restart condition is an address frame 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 issued.

Figure 33.3 shows the I<sup>2</sup>C bus format, and Figure 33.4 shows the I<sup>2</sup>C bus timing.

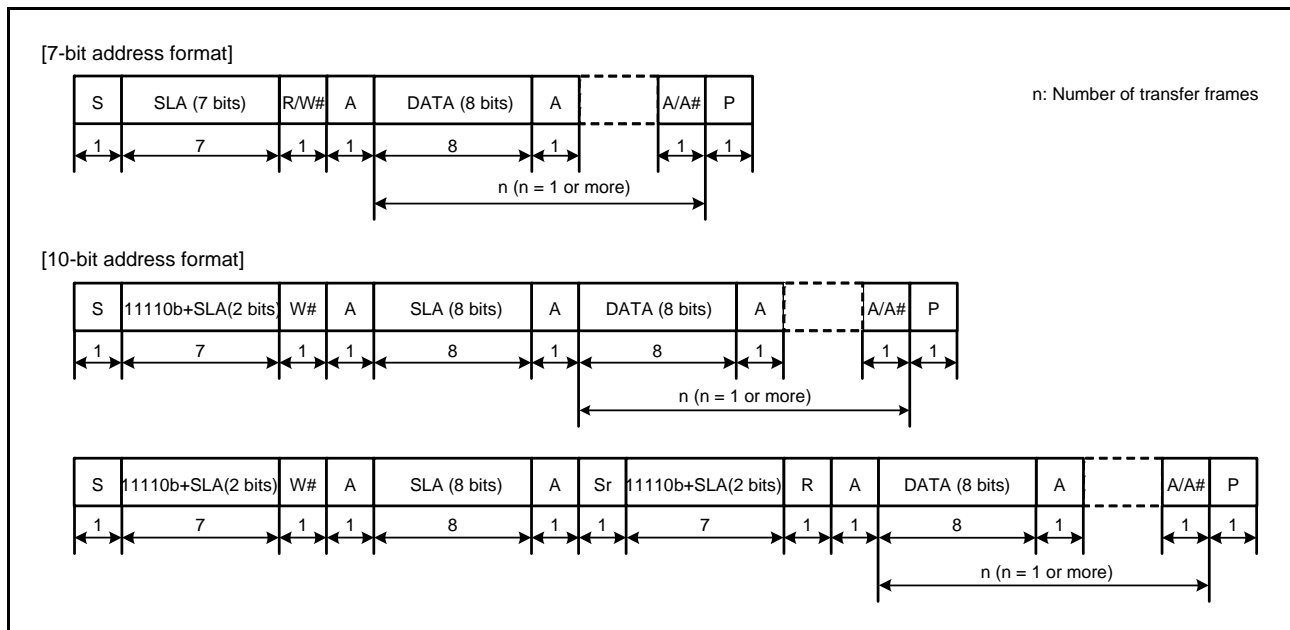


Figure 33.3 I<sup>2</sup>C Bus Format

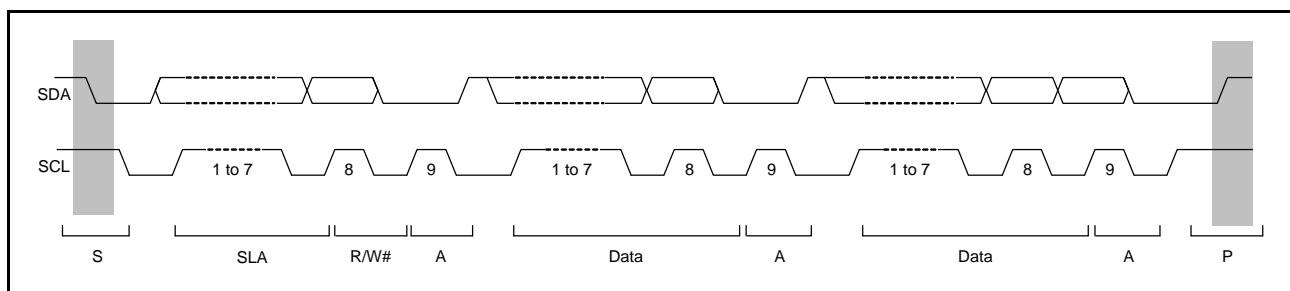


Figure 33.4 I<sup>2</sup>C Bus Timing (SLA = 7 Bits)

- S: Start condition. The master device drives the SDA<sub>n</sub> line low from high level while the SCL<sub>n</sub> line is at a high level.
- SLA: Slave address, by which the master device selects a slave device.
- R/W#: Indicates the direction of data transfer: from the slave device to the master device when R/W is 1, or from the master device to the slave device when R/W is 0.
- A: Acknowledge. The receive device drives the SDA<sub>n</sub> line low. (In master transmit mode, the slave device returns acknowledge. In master receive mode, the master device returns acknowledge.)
- Sr: Restart condition. The master device drives the SDA<sub>n</sub> line low from the high level after the setup time has elapsed with the SCL<sub>n</sub> line at the high level.
- DATA: Transmitted or received data
- P: Stop condition. The master device drives the SDA<sub>n</sub> line high from low level while the SCL<sub>n</sub> line is at a high level.



### 33.3.2 Initial Settings

Before starting data transmission and reception, initialize the RIIC according to the procedure in Figure 33.5.

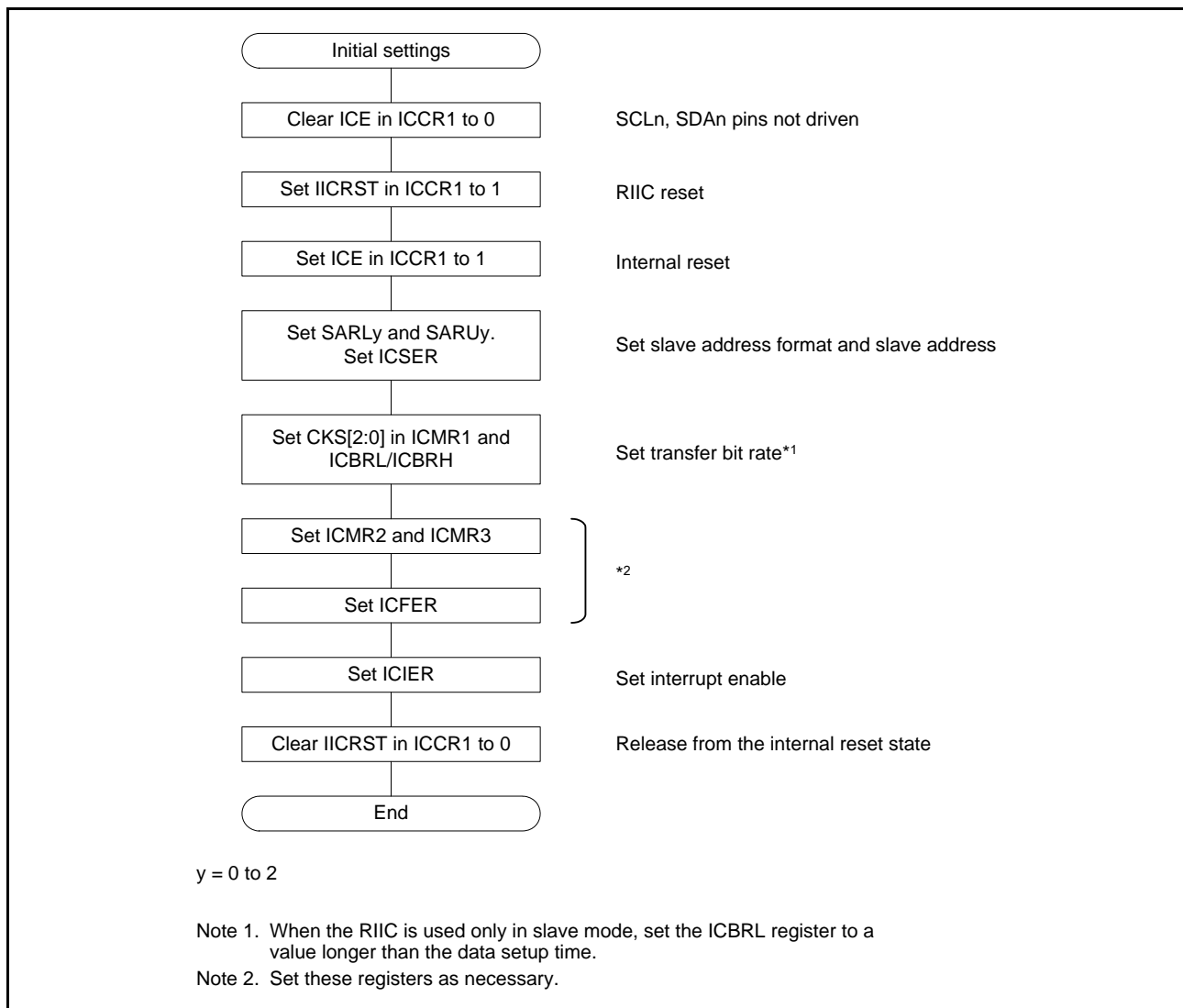


Figure 33.5 Example of RIIC Initialization Flowchart

### 33.3.3 Master Transmit Operation

In master transmit operation, the RIIC outputs the SCL (clock) and transmitted data signals as the master device, and the slave device returns acknowledgements. Figure 33.6 shows an example of usage of master transmission and Figure 33.7 to Figure 33.9 show the timing of operations in master transmission.

The following describes the procedure and operations for master transmission.

- (1) Set the IICRST bit in ICCR1 to 1 (RIIC reset) and then set the ICE bit in ICCR1 to 1 (internal reset) with the ICE bit in ICCR1 cleared to 0 (SCLn and SDAn pins not driven). This initializes the internal state and the various flags of ICSR1. 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 Table 33.5). When the necessary register settings have been completed, set the ICCR1.IICRST bit to 0 (for release from the reset state). This step is not necessary if initialization of the RIIC has already been completed.
- (2) Read the BBSY flag in ICCR2 to check that the bus is open, and then set the ST bit in ICCR2 to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. At the same time, the BBSY flag and the START flag in ICSR2 are automatically set to 1 and the ST bit is automatically cleared 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 SDAn line have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and the MST and TRS bits in ICCR2 are automatically set to 1, placing the RIIC in master transmit mode. The TDRE flag in ICSR2 is also automatically set to 1 in response to setting of the TRS bit to 1.
- (3) Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the slave address and the R/W# bit) to ICDRT. Once the data for transmission are written to ICDRT, the TDRE flag is automatically cleared to 0, the data are transferred from ICDRT to ICDRS, 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.  
 Since 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 issue a stop condition.  
 For data transmission with an address in the 10-bit format, start by writing 1111 0b, the two higher-order bits of the slave address, and W to ICDRT as the first address transmission. Then, as the second address transmission, write the eight lower-order bits of the slave address to ICDRT.
- (4) After confirming that the TDRE flag in ICSR2 is 1, write the data for transmission to the ICDRT register. The RIIC automatically holds the SCLn line low until the data for transmission are ready or a stop condition is issued.
- (5) After all bytes of data for transmission have been written to the ICDRT register, wait until the value of the TEND flag in ICSR2 returns to 1, and then set the SP bit in ICCR2 to 1 (stop condition issuance request). Upon receiving a stop condition issuance request, the RIIC issues the stop condition.
- (6) Upon detecting the stop condition, the RIIC automatically clears the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Furthermore, it automatically clears the TDRE and TEND flags to 0, and sets the STOP flag in ICSR2 to 1.
- (7) After checking that the ICSR2.STOP flag is 1, clear the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

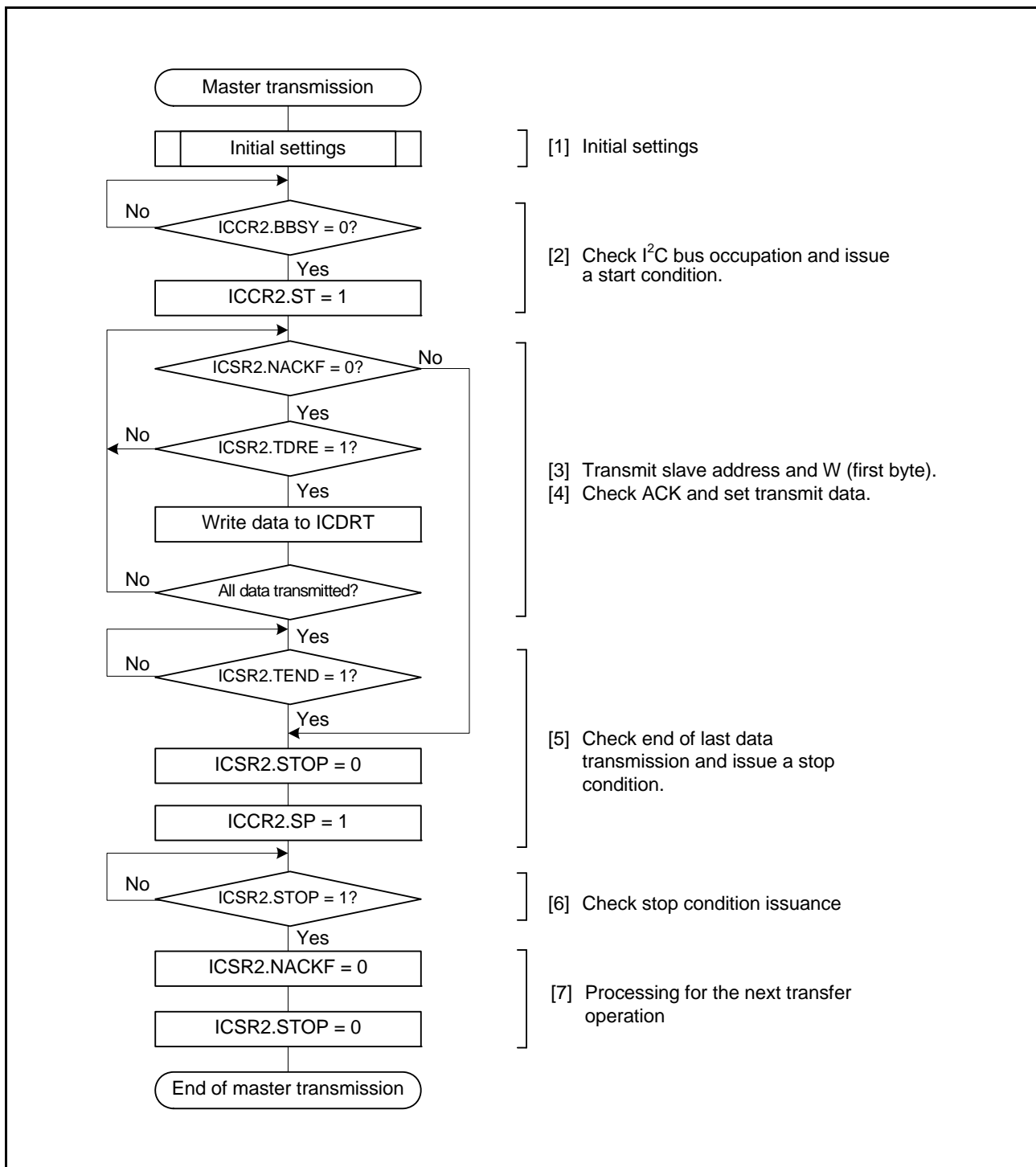


Figure 33.6 Example of Master Transmission Flowchart

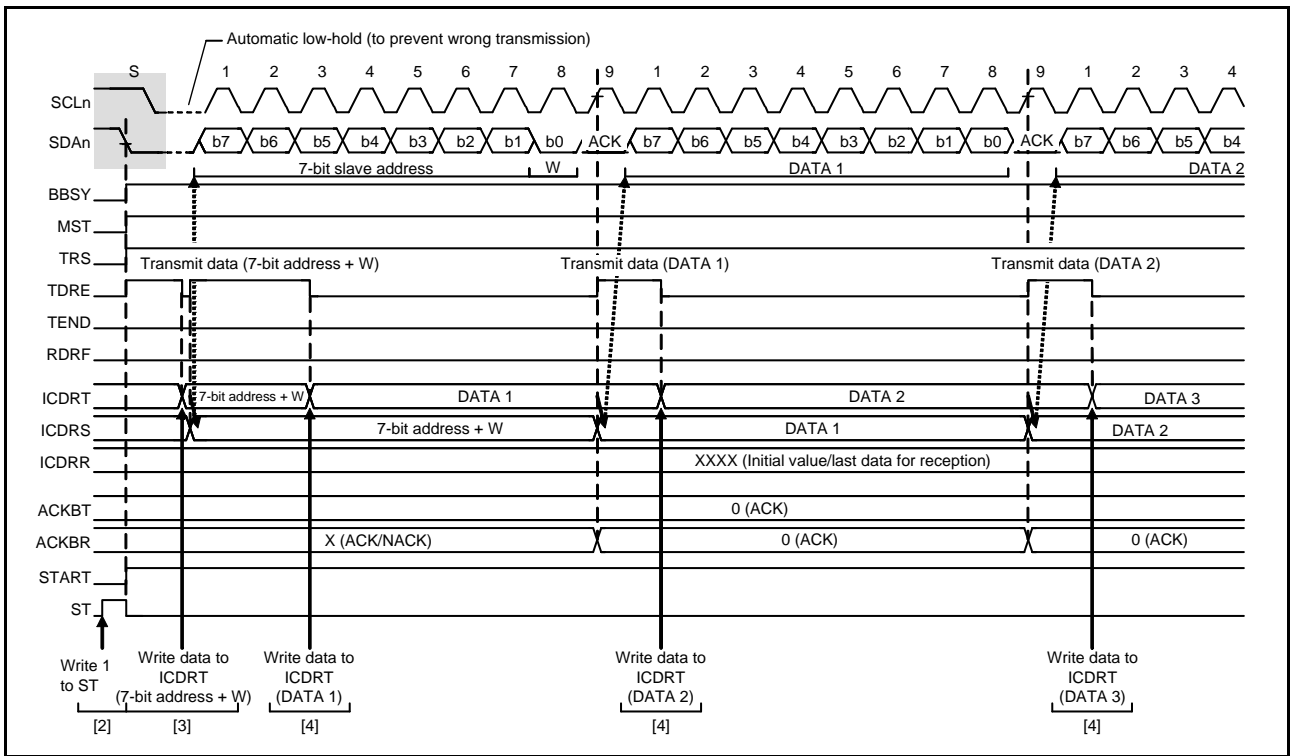


Figure 33.7 Master Transmit Operation Timing (1) (7-Bit Address Format)

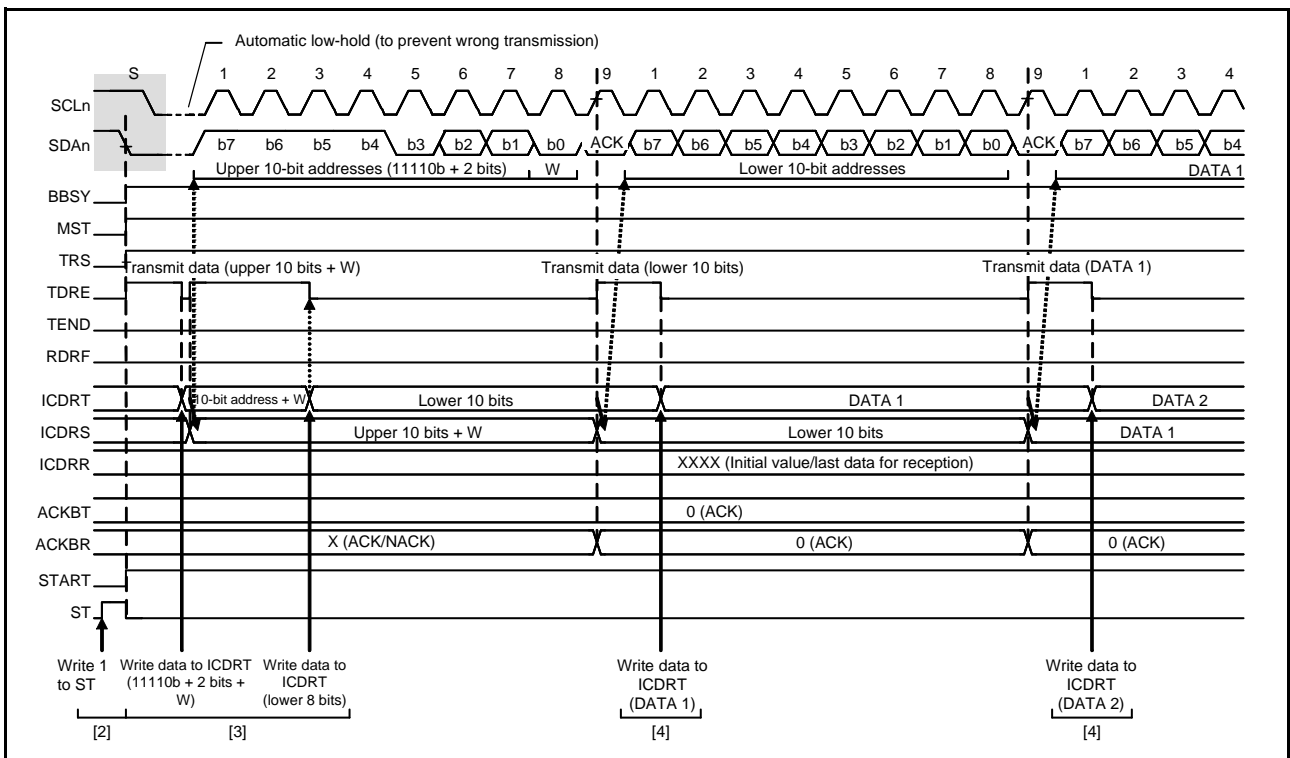


Figure 33.8 Master Transmit Operation Timing (2) (10-Bit Address Format)

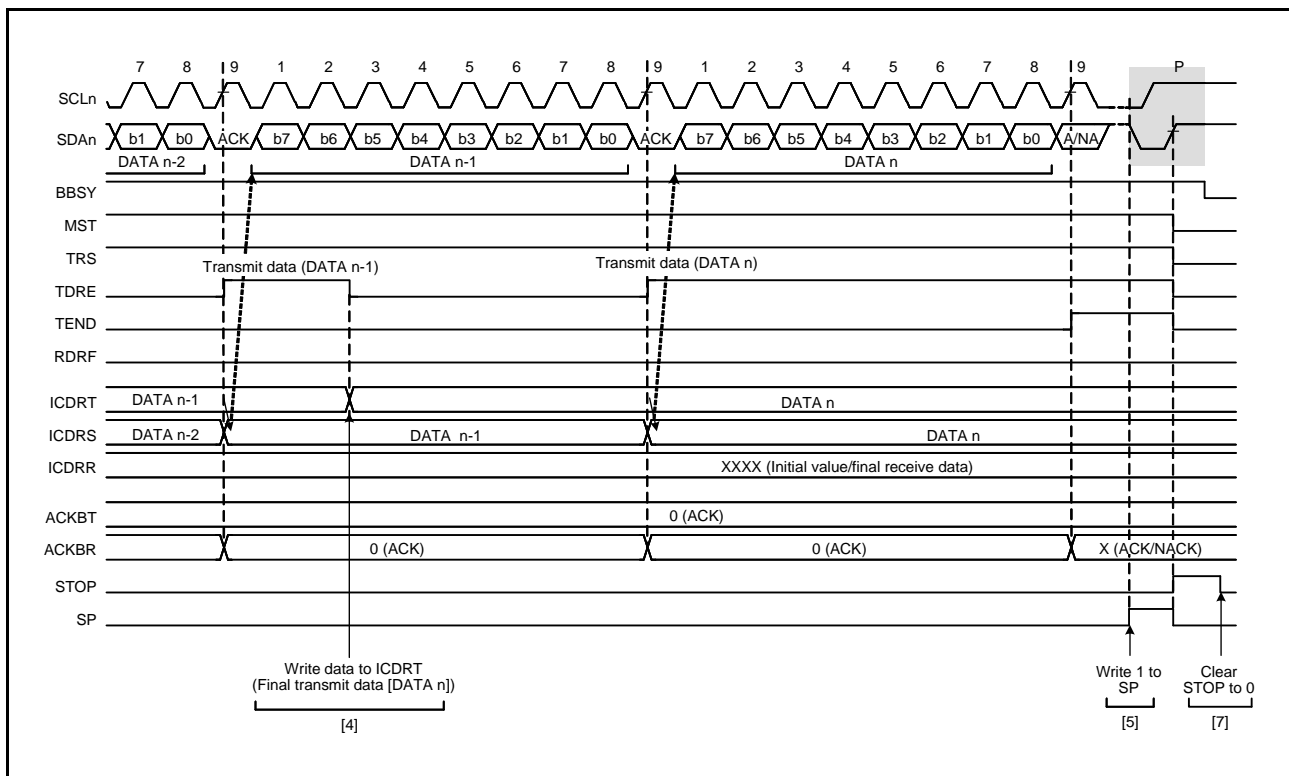


Figure 33.9 Master Transmit Operation Timing (3)

### 33.3.4 Master Receive Operation

In master receive operation, the RIIC as a master device outputs the SCL (clock) signal, receives data from the slave device, and returns acknowledgements. Since 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 33.10 shows an example of usage of master reception and Figure 33.11 and Figure 33.13 show the timing of operations in master reception.

The following describes the procedure and operations for master reception.

- (1) Set the IICRST bit in ICCR1 to 1 (RIIC reset) and then set the ICE bit in ICCR1 to 1 (internal reset) with the ICE bit in ICCR1 cleared to 0 (SCLn and SDA pins not driven). This initializes the internal state and the various flags of ICSR1. 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 Table 33.5). When the necessary register settings have been completed, set the ICCR1.IICRST bit to 0 (for release from the reset state). This step is not necessary if initialization of the RIIC has already been completed.
- (2) Read the BBSY flag in ICCR2 to check that the bus is open, and then set the ST bit in ICCR2 to 1 (start condition issuance request). Upon receiving the request, the RIIC issues a start condition. When the RIIC detects the start condition, the BBSY flag and the START flag in ICSR2 are automatically set to 1 and the ST bit is automatically cleared to 0. At this time, if the start condition is detected and the levels for the SDA output and the levels on the SDA pin have matched while the ST bit is 1, the RIIC recognizes that issuing of the start condition as requested by the ST bit has been successfully completed, and the MST and TRS bits in ICCR2 are automatically set to 1, placing the RIIC in master transmit mode. The TDRE flag in ICSR2 is also automatically set to 1 in response to setting of the TRS bit to 1.

- (3) Check that the TDRE flag in ICSR2 is 1, and then write the value for transmission (the first byte indicates the slave address and value of the R/W# bit) to ICDRT. Once the data for transmission are written to ICDRT, the TDRE flag is automatically cleared to 0, the data are transferred from ICDRT to ICDRS, 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 cleared to 0 on the rising edge of the ninth cycle of SCLn (the clock signal), placing the RIIC in master receive mode. At this time, the TDRE flag is automatically cleared to 0 and the ICSR2.RDRF flag is automatically set to 1.  
Since 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 issue a stop condition.  
For master reception from a device with a 10-bit address, start by using master transmission to issue the 10-bit address, and then issue a restart condition. After that, 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 ICDRR after confirming that the RDRF flag in ICSR2 is 1; this makes the RIIC start output of the SCL (clock) signal and start data reception.
- (5) After 1 byte of data has been received, the RDRF flag in ICSR2 is set to 1 on the rising edge of the eighth or ninth cycle of SCL clock (the clock signal) as selected by the RDRFS bit in ICMR3. Reading out ICDRR at this time will produce the received data, and the RDRF flag is automatically cleared to 0 at the same time. Furthermore, the value of the acknowledgement field received during the ninth cycle of SCL clock is returned as the value set in the ACKBT bit of ICMR3. 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 (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 SCLn line to the low level on the rising edge of the ninth clock cycle in reception of the last byte, so the state is such that issuing 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 ACKBT bit in ICMR3 to 1 (NACK).
- (7) After reading out 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 SP bit in ICCR2 (stop condition issuance request) and then read the last byte from ICDRR. When ICDRR is read, the RIIC is released from the wait state and issues the stop condition after low-level output in the ninth clock cycle is completed or the SCL line is released from the low-hold state.
- (8) Upon detecting the stop condition, the RIIC automatically clears the MST and TRS bits in ICCR2 to 00b and enters slave receive mode. Furthermore, detection of the stop condition leads to setting of the STOP flag in ICSR2 to 1.
- (9) After checking that the ICSR2.STOP flag is 1, clear the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

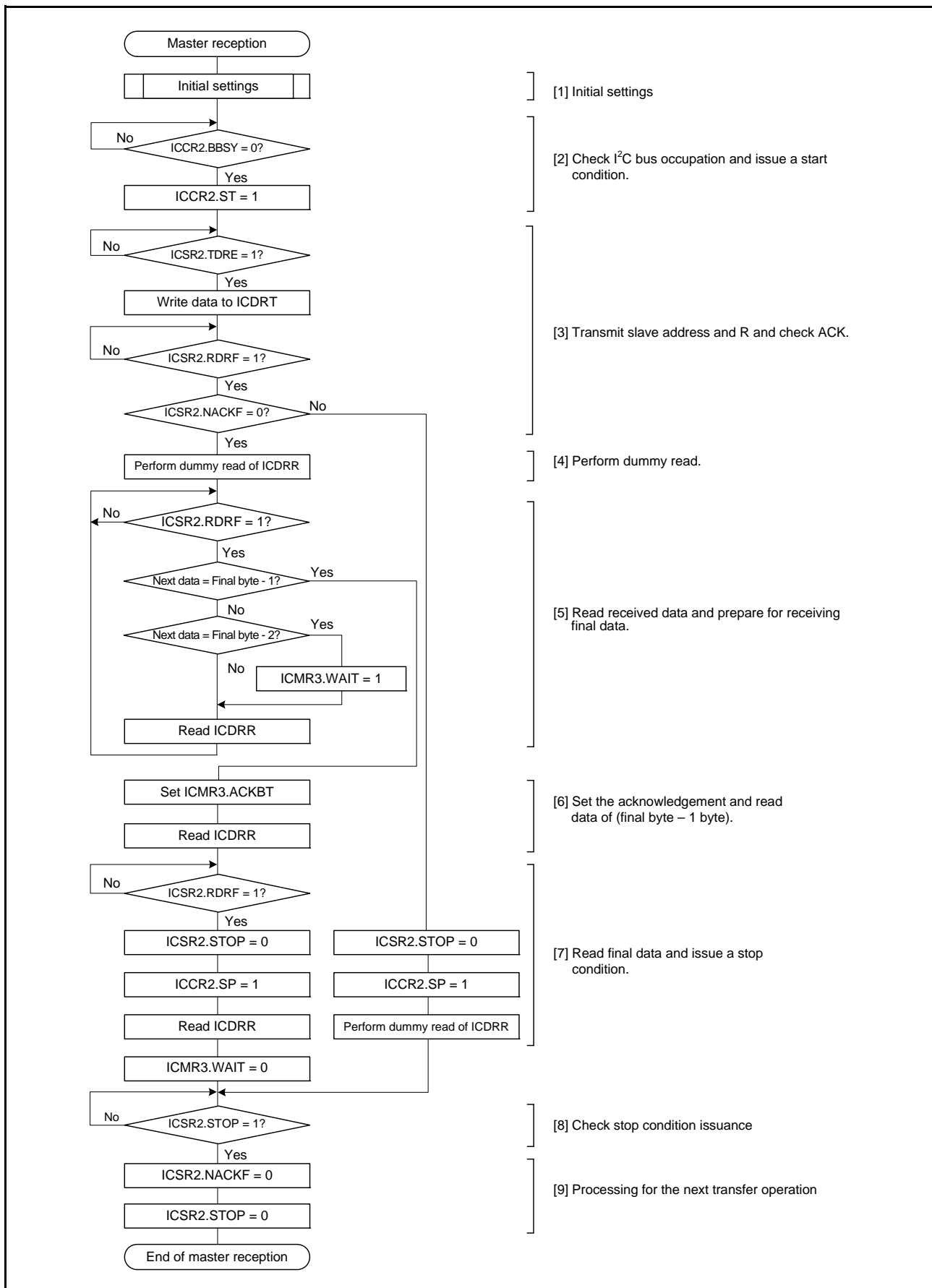


Figure 33.10 Example of Master Reception Flowchart (7-Bit Address Format)

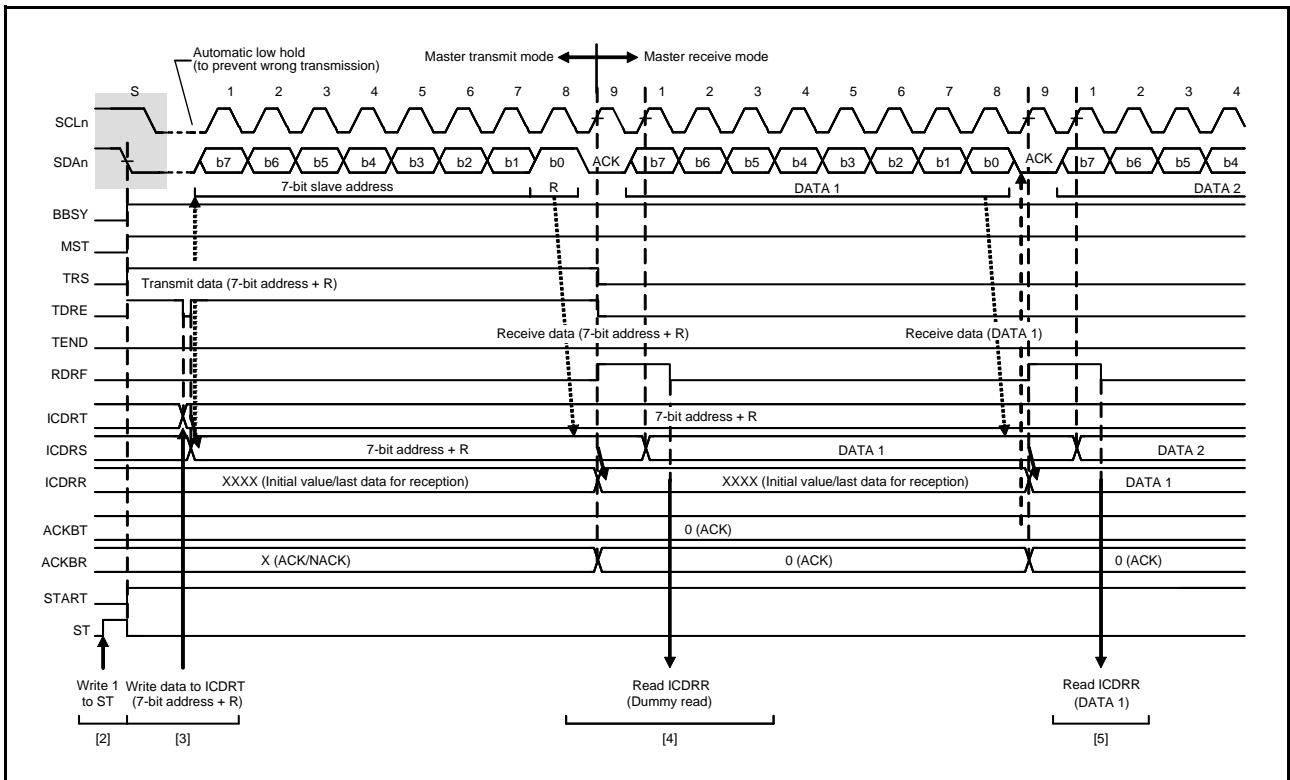


Figure 33.11 Master Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

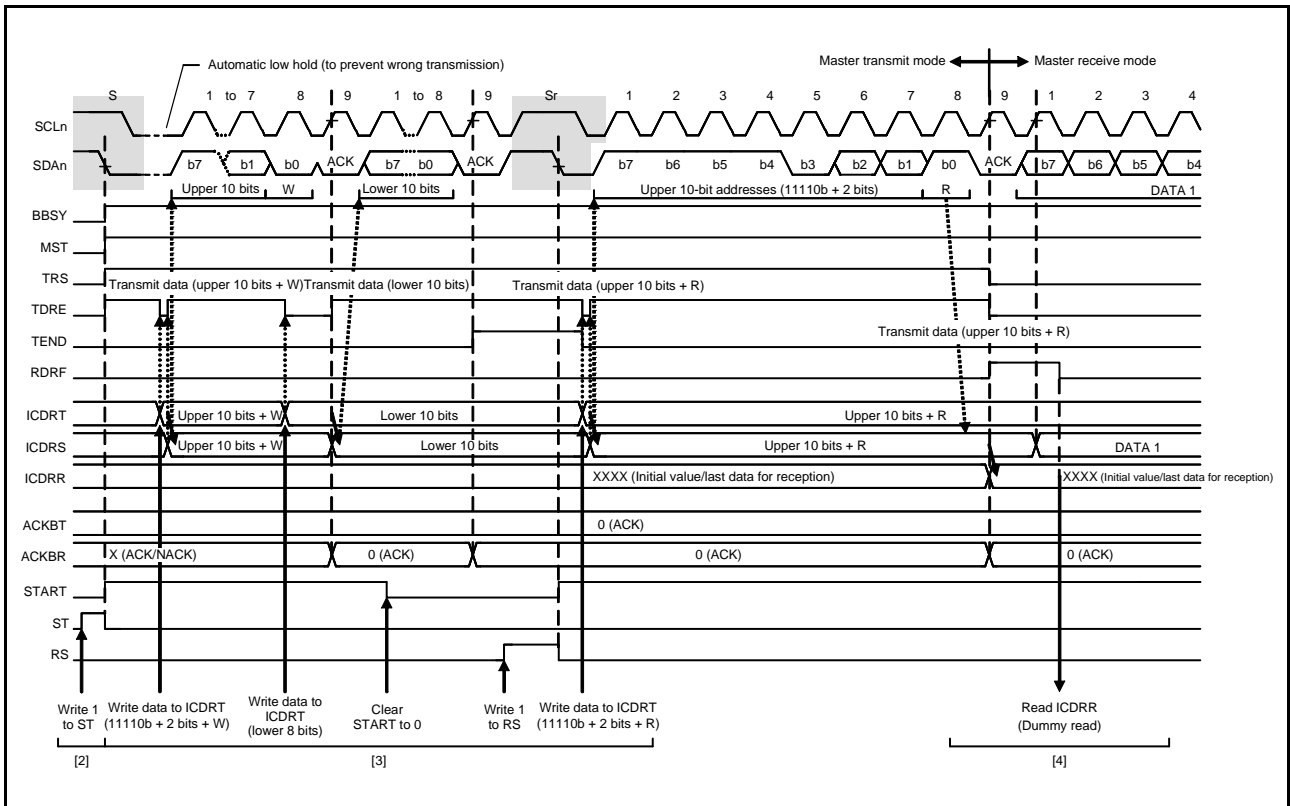


Figure 33.12 Master Receive Operation Timing (2) (10-Bit Address Format, when RDRFS = 0)



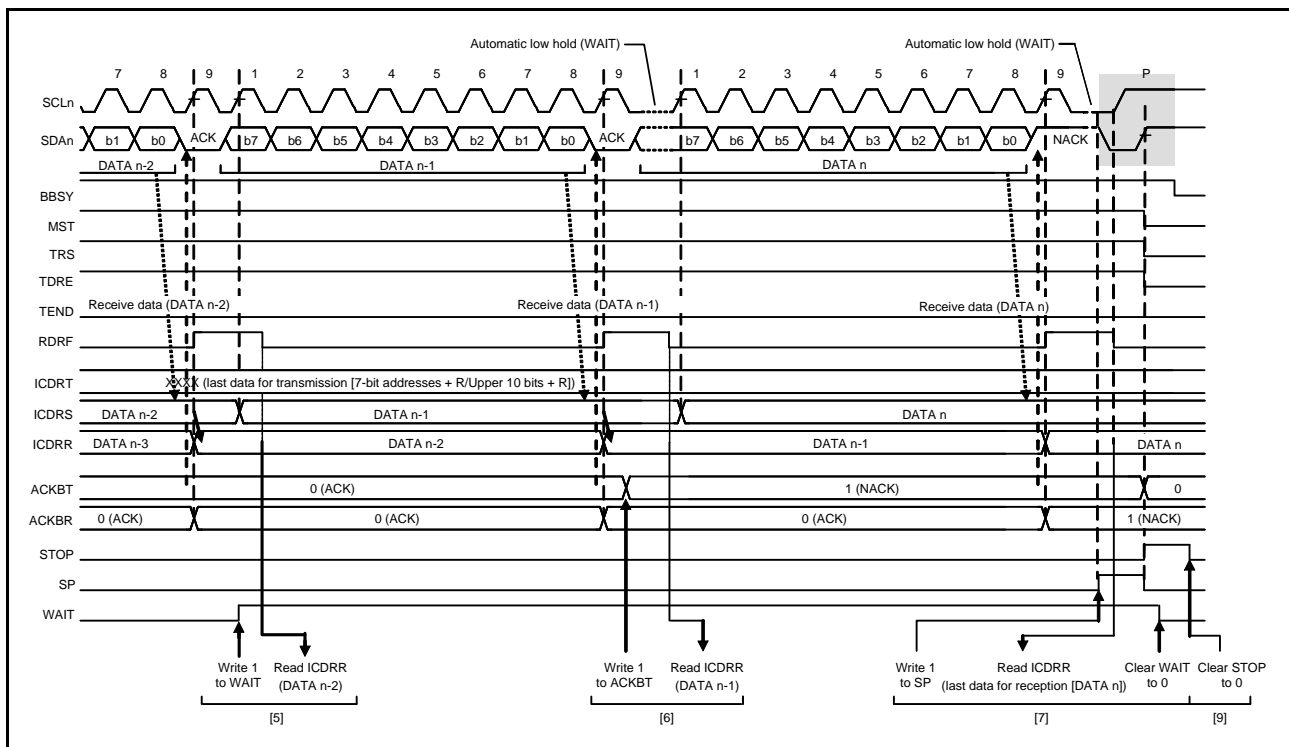


Figure 33.13 Master Receive Operation Timing (3) (when RDRFS = 0)

### 33.3.5 Slave Transmit Operation

In slave transmit operation, the master device outputs the SCL (clock) signal, the RIIC transmits data as a slave device, and the master device returns acknowledgements.

Figure 33.14 shows an example of usage of slave transmission and Figure 33.15 and Figure 33.16 show the timing of operations in slave transmission.

The following describes the procedure and operations for slave transmission.

- (1) Follow the procedure in Figure 33.5 to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. 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 cycle of SCL clock (the clock signal) and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. 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 TRS bit and the TDRE flag in ICSR2 to 1.
- (3) After the ICSR2.TEND flag is confirmed to be 1, write the data for transmission to the ICDRT register. At this time, if the RIIC receives no acknowledge from the master device (receives a NACK signal) while the ICFER.NACKF 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 SCLn line low on the ninth falling edge of SCL clock.
- (5) When the ICSR2.NACKF flag or the ICSR2.TEND flag is 1, dummy read ICDRR to complete the processing. This releases the SCLn line.
- (6) Upon detecting the stop condition, the RIIC automatically clears bits ICSR1.HOA, GCA, and AASy (y = 0 to 2), flags ICSR2.TDRE and TEND, and the ICSR2.TRS bit to 0, and enters slave receive mode.

(7) After checking that the ICSR2.STOP flag is 1, clear the ICSR2.NACKF and STOP flags to 0 for the next transfer operation.

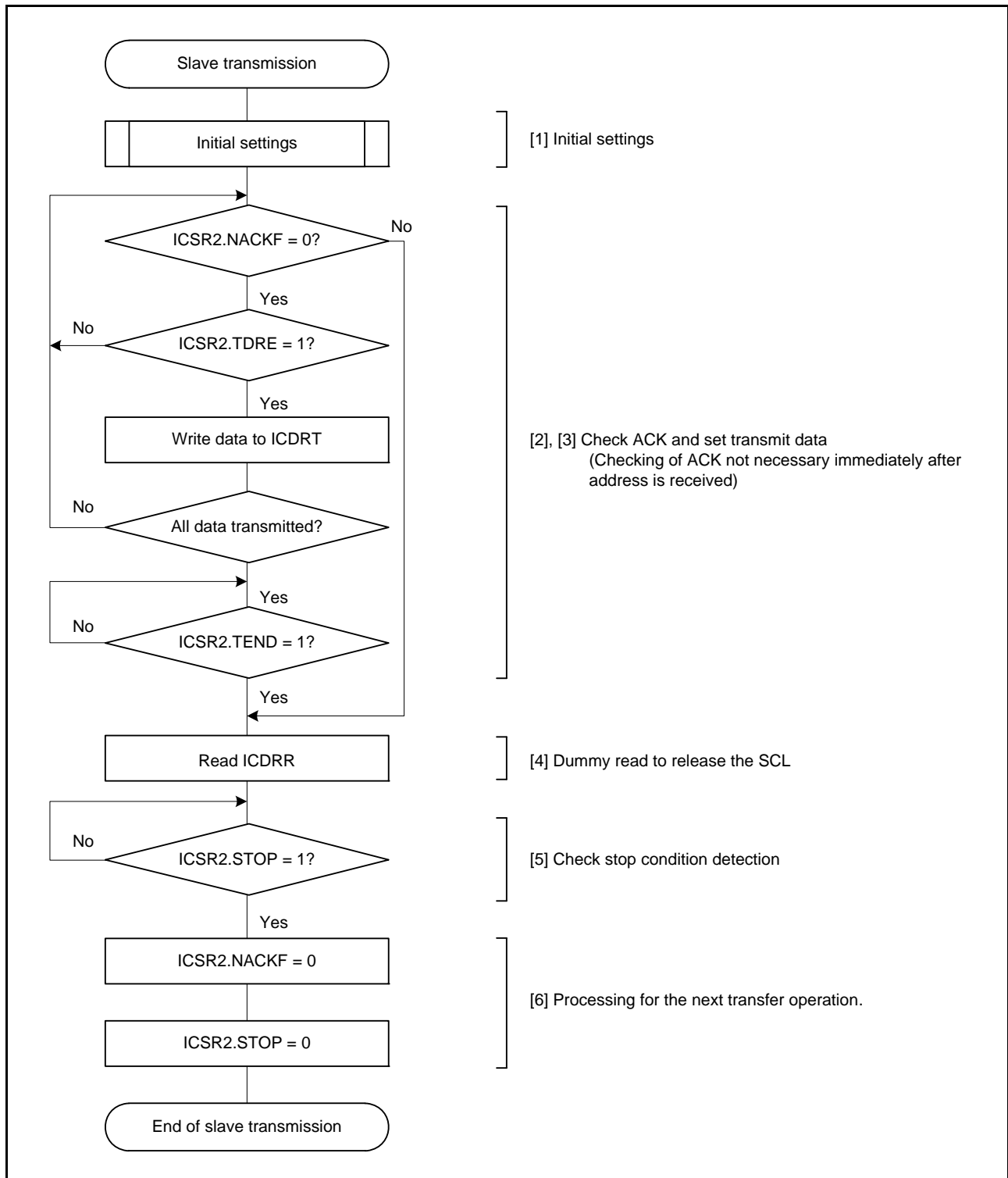


Figure 33.14 Example of Slave Transmission Flowchart

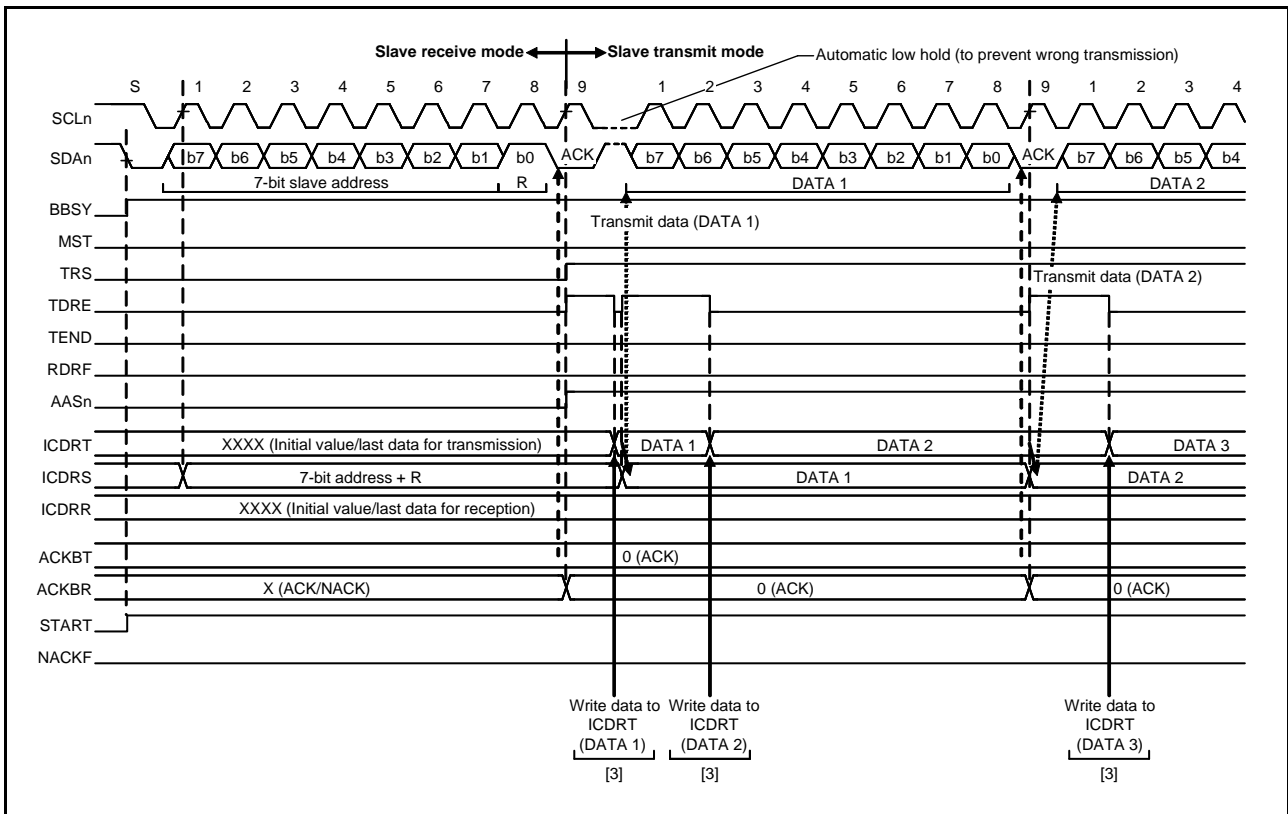


Figure 33.15 Slave Transmit Operation Timing (1) (7-Bit Address Format)

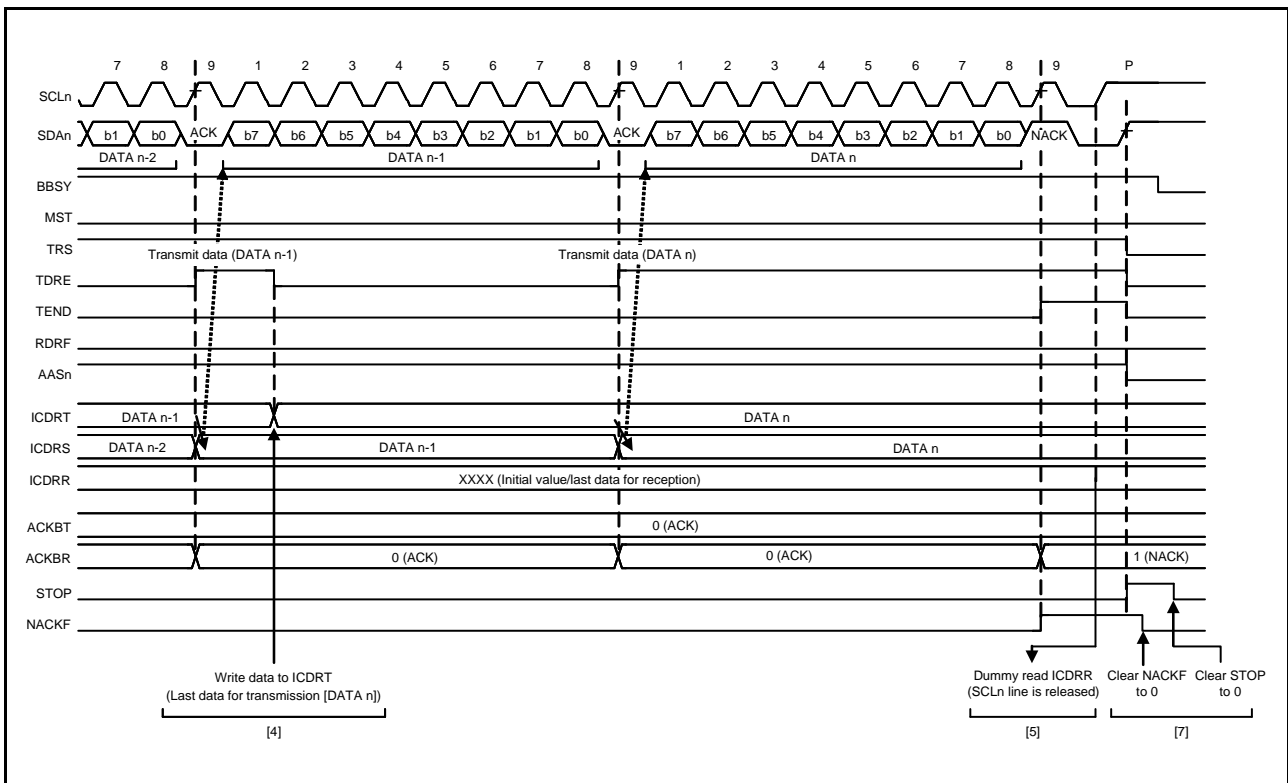


Figure 33.16 Slave Transmit Operation Timing (2)

### 33.3.6 Slave Receive Operation

In slave receive operation, the master device outputs the SCL clock and transmit data, and the RIIC returns acknowledgements as a slave device.

Figure 33.17 shows an example of usage of slave reception and Figure 33.18 and Figure 33.19 show the timing of operations in slave reception.

The following describes the procedure and operations for slave reception.

- (1) Follow the procedure in Figure 33.5 to make initial settings for the RIIC. This step is not necessary if initialization of the RIIC has already been completed. 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 cycle of SCL clock (the clock signal) and outputs the value set in the ICMR3.ACKBT bit to the acknowledge bit on the ninth cycle of SCL clock. 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 RDRF flag in ICSR2 to 1.
  - (3) After the ICSR2.STOP flag is confirmed to be 0 and the ICSR2.RDRF flag to be 1, dummy read ICDRR (the dummy value consists of the slave address and R/W# bit when the 7-bit address format is selected, or the lower eight bits when the 10-bit address format is selected).
  - (4) When ICDRR is read, the RIIC automatically clears the ICSR2.RDRF flag to 0. If reading of ICDRR is delayed and a next byte is received while the RDRF flag is still set to 1, the RIIC holds the SCLn line low from one SCL cycle before the timing with which RDRF should be set. In this case, reading ICDRR releases the SCLn line from being held at the low level.
- When the ICSR2.STOP flag is 1 and the ICSR2.RDRF flag is also 1, read ICDRR until all the data is completely received.
- (5) 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, clear the ICSR2.STOP flag to 0 for the next transfer operation.

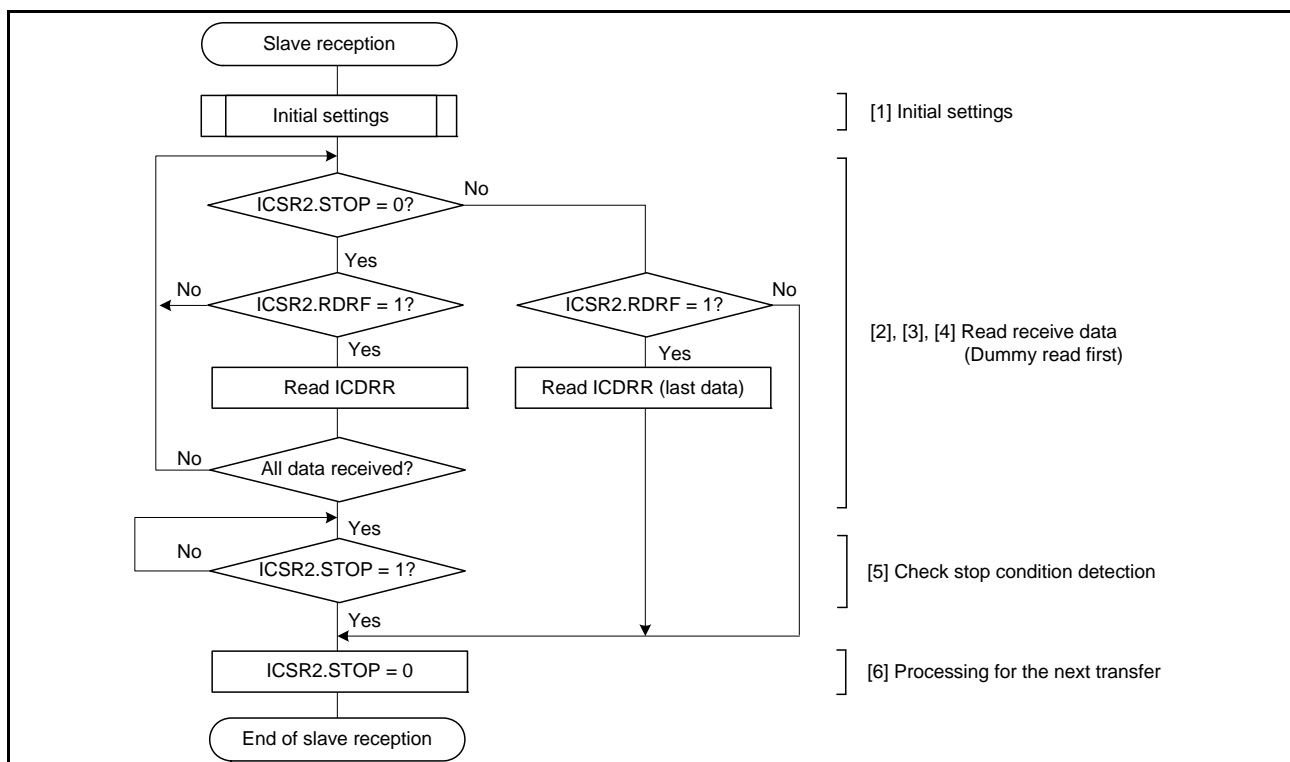


Figure 33.17 Example of Slave Reception Flowchart

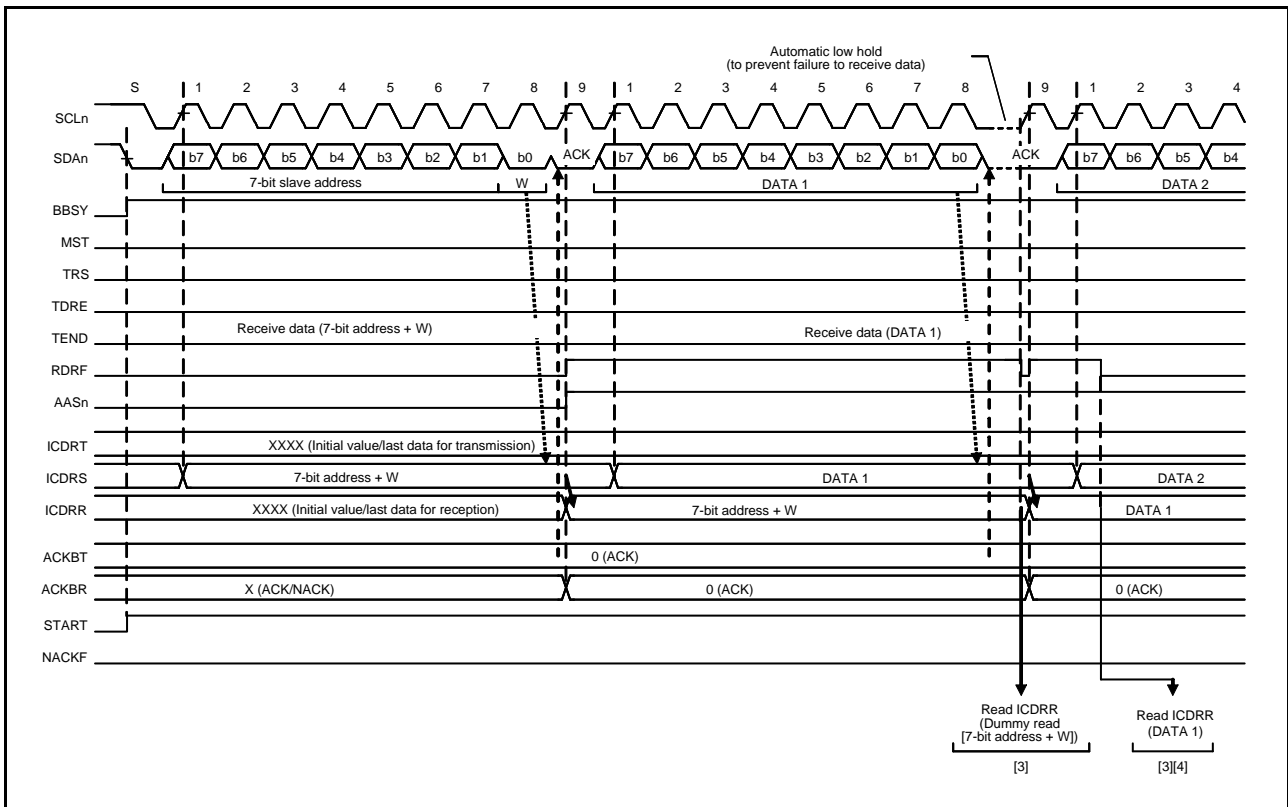


Figure 33.18 Slave Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0)

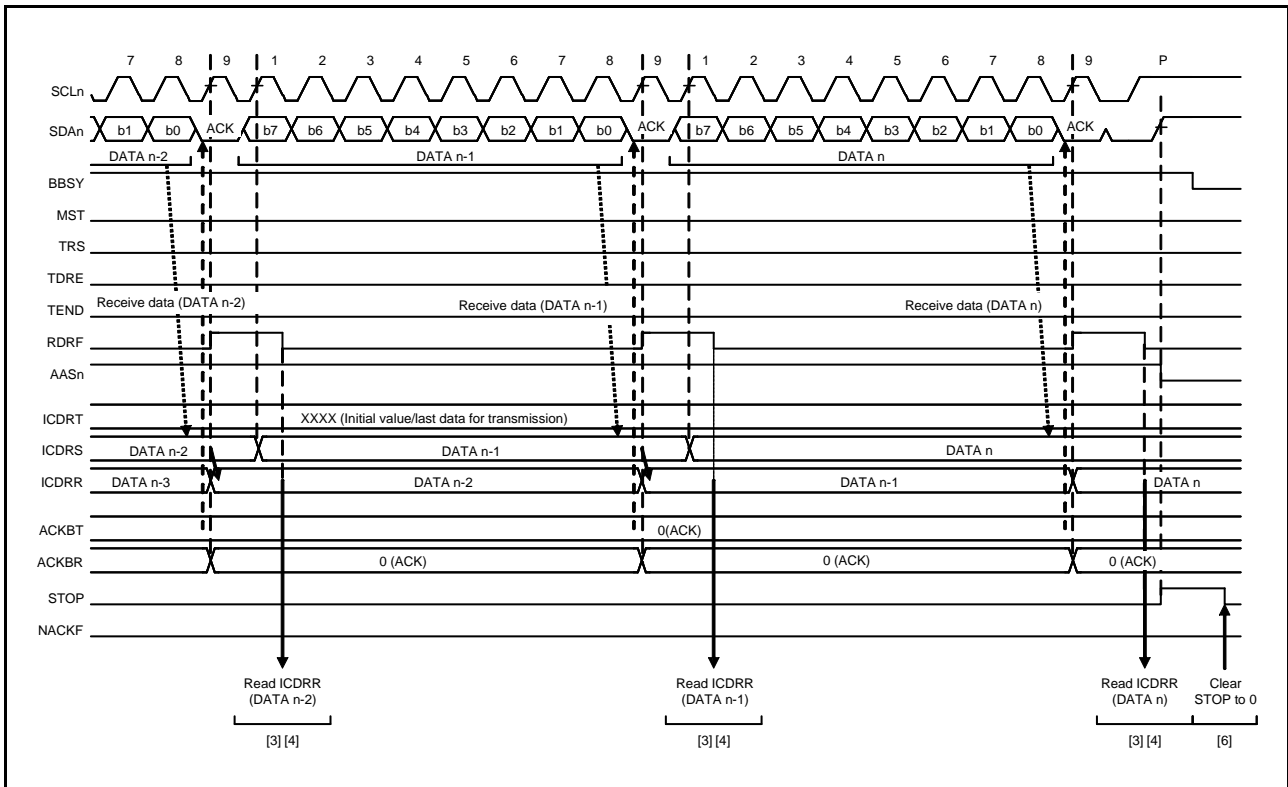


Figure 33.19 Slave Receive Operation Timing (2) (when RDRFS = 0)

### 33.4 SCL Synchronization Circuit

In generation of the SCL (clock) signal, the RIIC starts counting out the value for width at high level specified in ICBRH when it detects a rising edge on the SCLn line and drives the SCLn line low once counting of the width at high level is complete. When the RIIC detects the falling edge of the SCLn line, it starts counting out the width at low level period specified in ICBRL, and then stops driving the SCLn line (releases the line) once counting of the width at low level is complete. The SCL (clock) signal is thus generated.

If multiple master devices are connected to the I<sup>2</sup>C 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 this synchronization of SCL signals must be bit by bit, the RIIC is equipped with a facility (the SCL synchronization circuit) to obtain bit-by-bit synchronization of the SCL clock signals by monitoring the SCLn line while in master mode.

When the RIIC has detected a rising edge on the SCLn line and thus started counting out the width at high level specified in ICBRH, and the level on the SCLn 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 SCLn line low, and starts counting out the width at low level specified in ICBRL. When the RIIC finishes counting out the width at low level, it stops driving the SCLn line to the low level (i.e. releases the line). At this time, if the width at low level of the SCL clock 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 SCL line has been released. When the RIIC finishes outputting the low-level period of the SCL clock, the SCLn line is released and the SCL clock 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 SCLE bit in ICFER is set to 1.

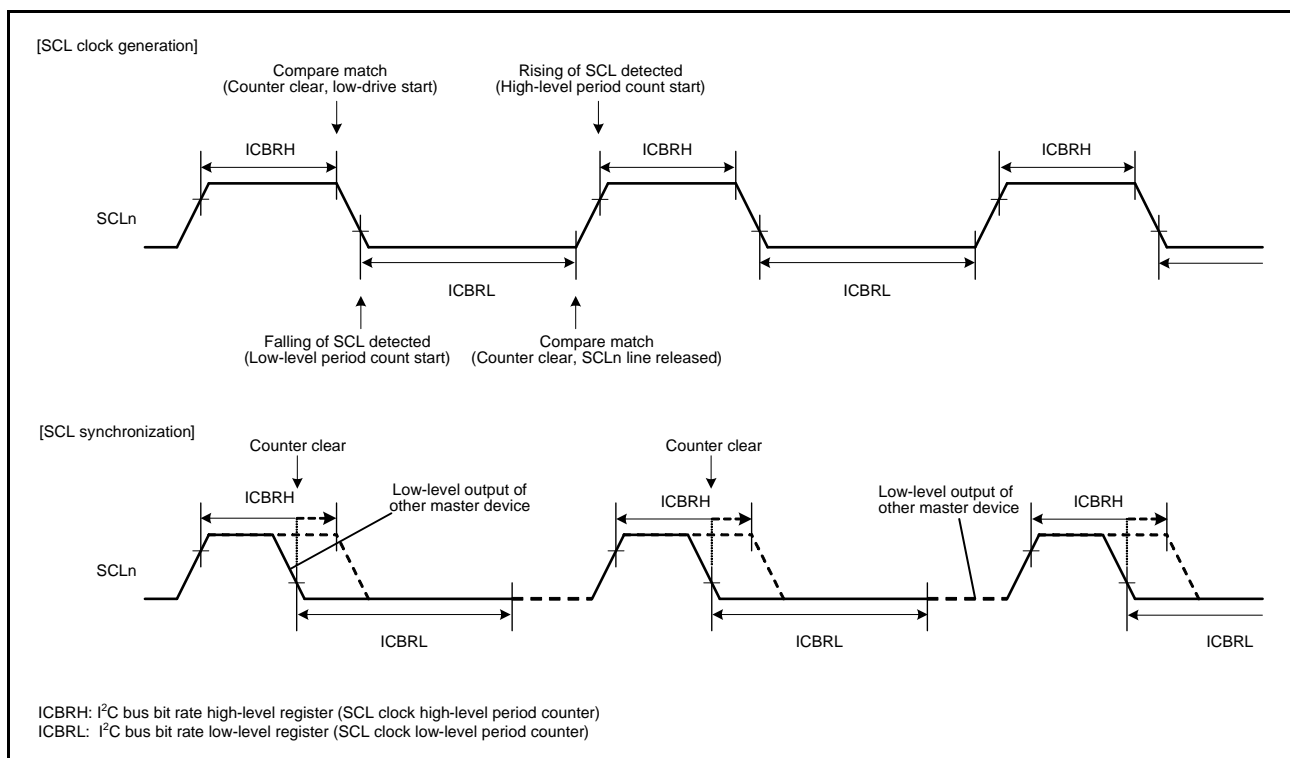


Figure 33.20 Generation and Synchronization of the SCL Signal from the RIIC

### 33.5 Facility for Delaying SDA Output

The RIIC module incorporates a facility for delaying output on the SDA line. The delay can be applied to all output (issuing of the start, restart, and stop conditions, data, and the ACK and NACK signals) on the SDA line.

With the SDA output delay facility, SDA output is delayed from detection of a falling edge of the SCL signal to ensure that the SDA signal is output within the interval over which the SCL (clock) signal is at the low level. Doing this leads to usage with the aim of preventing erroneous operation of communications devices, with the aim of satisfying the 300-ns (min.) data-hold time requirement of the SMBus specification.

The output delay facility is enabled by setting the SDDL[2:0] bits in IMCR2 to any value other than 000b, and disabled by setting the same bits to 000b.

While the SDA output delay facility is enabled (i.e. while the SDDL[2:0] bits in IMCR2 are set to any value other than 000b), the DLCS bit in IMCR2 selects the clock source for counting by the SDA output delay counter as the internal base clock (IIC $\phi$ ) for the RIIC module or as a clock signal derived by dividing the frequency of the internal base clock by two (IIC $\phi$ /2). The counter counts the number of cycles set in the SDDL[2:0] bits in IMCR2. After counting of the set number of cycles of delay is completed, the RIIC module places the required output (start, restart, or stop condition, data, or an ACK or NACK signal) on the SDA line.

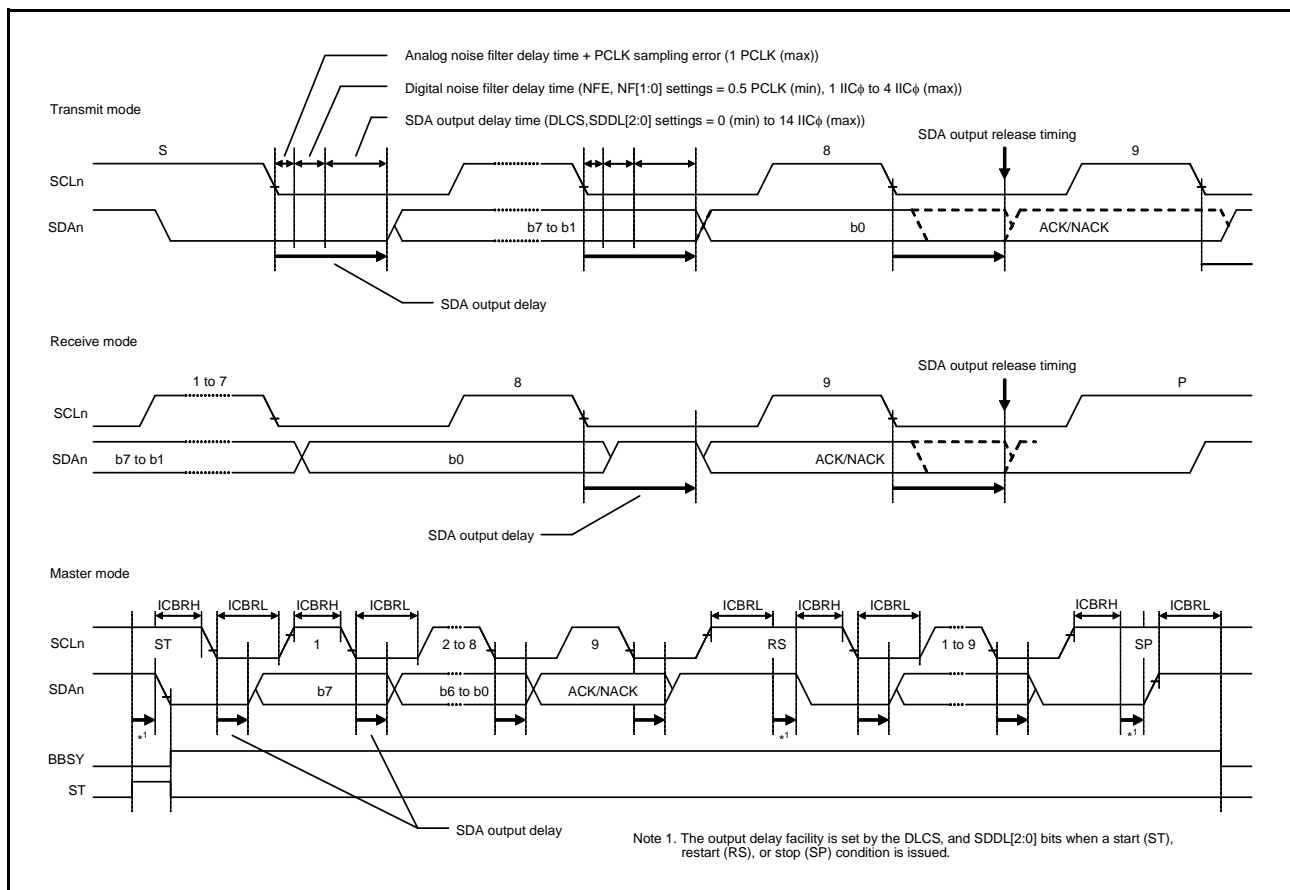


Figure 33.21 SDA Output Delay Facility

### 33.6 Digital Noise-Filter Circuits

The states of the SCLn and SDA<sub>n</sub> pins are conveyed to the internal circuitry through analog noise-filter and digital noise-filter circuits. Figure 33.22 is a block diagram of the digital noise-filter circuit.

The on-chip digital noise-filter circuit of the 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 NF[1:0] bits in ICMR3. The selected number of effective stages determines the noise-filtering capability as a period from one to four IIC $\phi$  cycles.

The input signal to the SCLn pin (or SDA<sub>n</sub> pin) is sampled on falling edges of the IIC $\phi$  signal. When the input signal level matches the output level of the number of effective flip-flop circuit stages as selected by the NF[1:0] bits in ICMR3, the signal level is conveyed to the subsequent stage. If the signal levels do not match, the previous value is retained.

If the ratio between the frequency of the internal operating clock (PCLK) and the transfer rate is 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 circuit (by clearing the NFE bit in ICFER) and use only the analog noise-filter circuit.

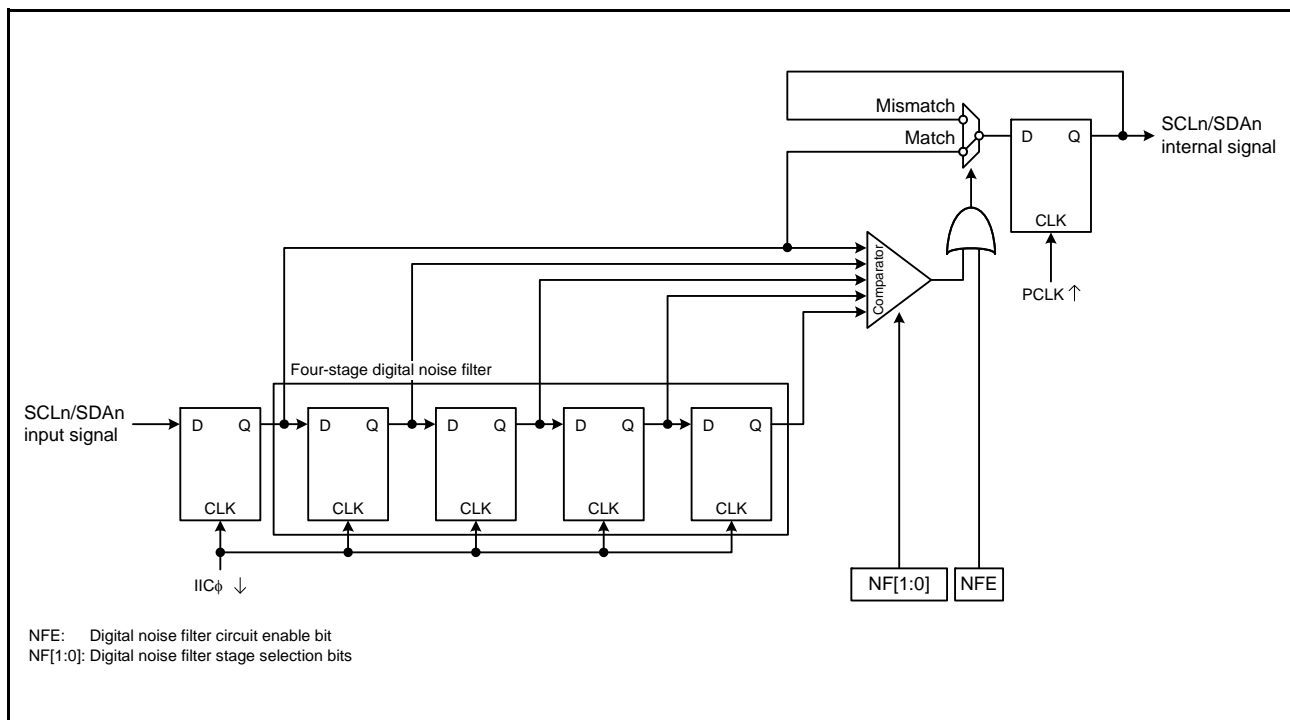


Figure 33.22 Block Diagram of Digital Noise Filter Circuit



### 33.7 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.

#### 33.7.1 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 SARyE bit ( $y = 0$  to 2) in ICSER is set to 1, the slave addresses set in SARUy and SARLy ( $y = 0$  to 2) can be detected.

When the RIIC detects a match of the set slave address, the corresponding AASy flag ( $y = 0$  to 2) in ICSR1 is set to 1 at the falling edge of the ninth SCL clock cycle, and the RDRF flag in ICSR2 or the TDRE flag in ICSR2 is set to 1 by the following R/W# bit. This causes a receive data full interrupt (ICRXI) or transmit data empty interrupt (ICTXI) to be generated. The AASy flag is used to identify which slave address has been specified.

Figure 33.23 to Figure 33.25 show the AASy flag set timing in three cases.

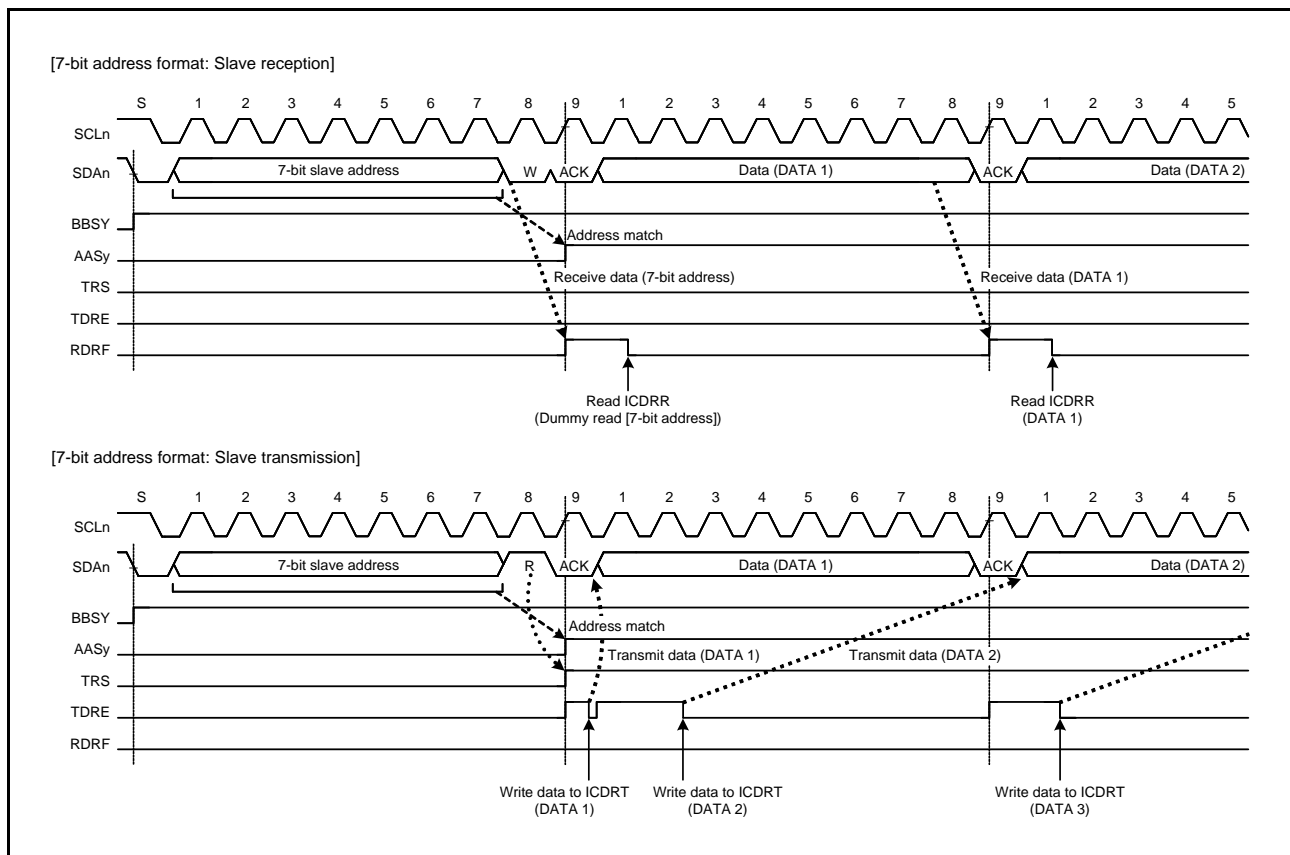


Figure 33.23 AASy Flag Set Timing with 7-Bit Address Format Selected

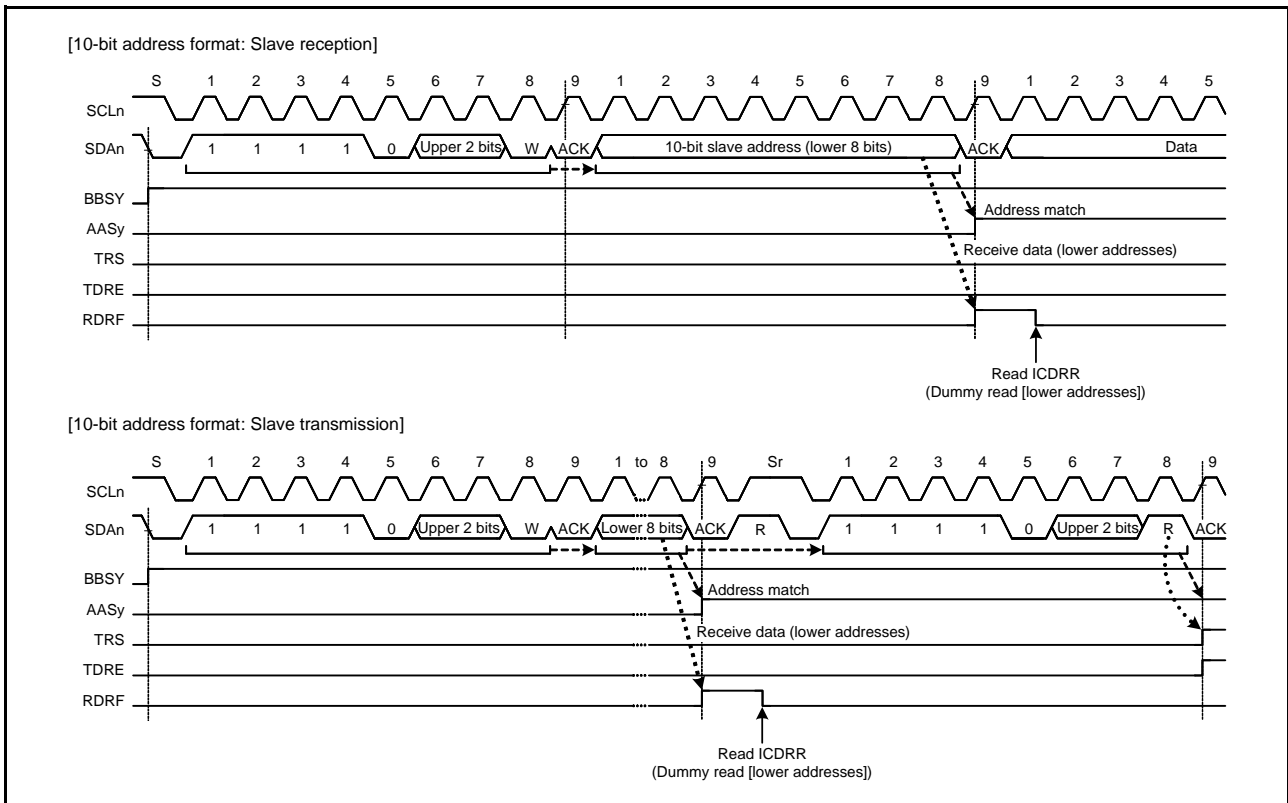


Figure 33.24 AASy Flag Set Timing with 10-Bit Address Format Selected

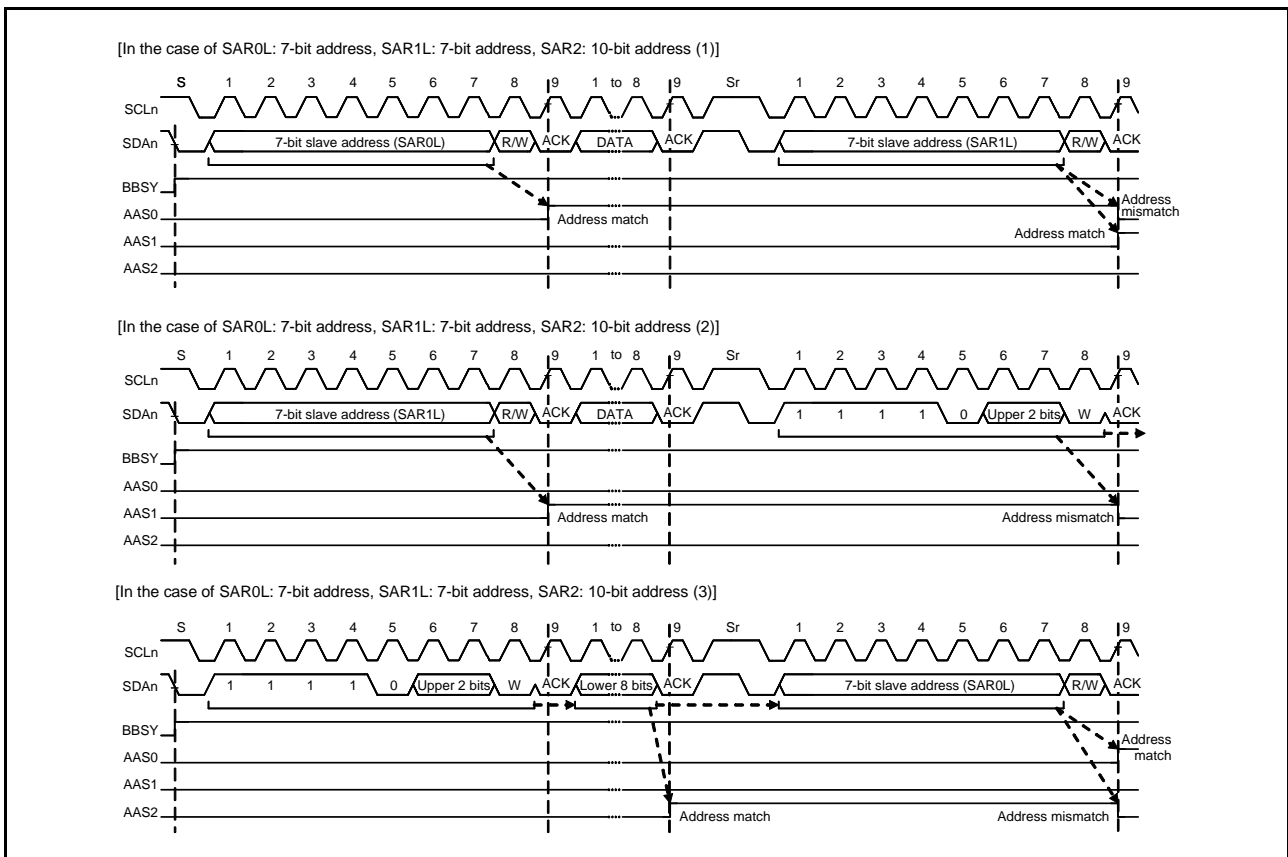


Figure 33.25 AASy Flag Set/Clear Timing with 7-Bit/10-Bit Address Formats Mixed

### 33.7.2 Detection of the General Call Address

The RIIC has a facility for detecting the general call address (0000 000b + 0 [W]). This is enabled by setting the GCAE bit in ICSER to 1.

If the address received after a start or restart condition is issued is 0000 000b + 1[R] (start byte), the 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 GCA flag in ICSR1 and the RDRF flag in ICSR2 are set to 1 on the falling edge of the ninth cycle of SCL clock. This leads to the generation of a receive data full interrupt (ICRXI). 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.

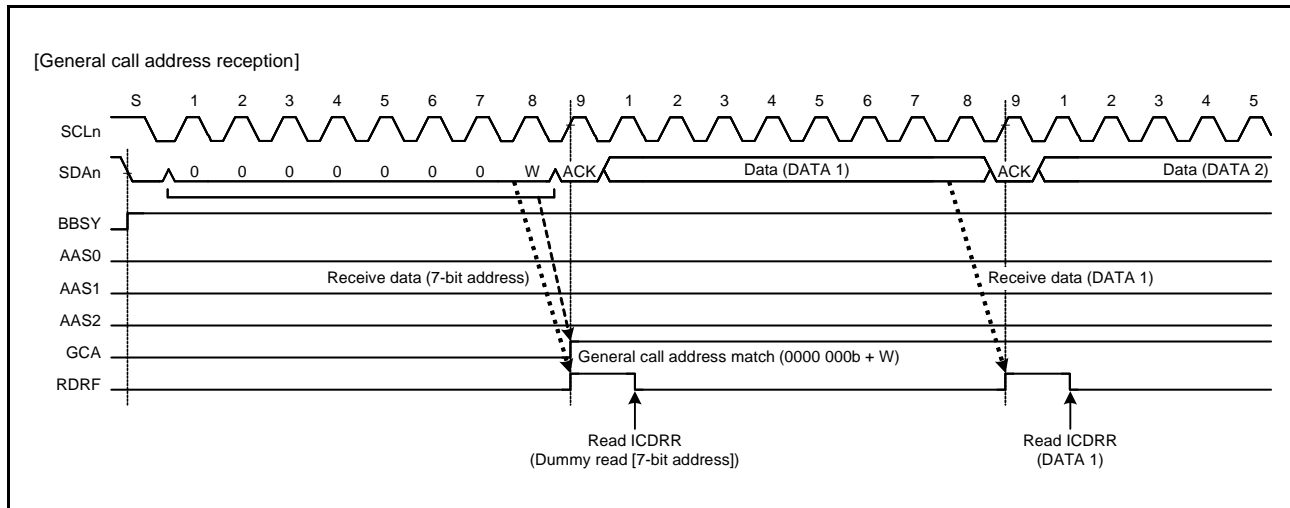


Figure 33.26 Timing of GCA Flag Setting during Reception of General Call Address

### 33.7.3 Device-ID Address Detection

The RIIC module has a facility for detecting device-ID addresses conformant with the I<sup>2</sup>C bus specification (Rev. 03).

When the RIIC receives 1111 100b as the first byte after a start condition or restart condition was issued with the DIDE bit in ICSER set to 1, the RIIC recognizes the address as a device ID, sets the DID flag in ICSR1 to 1 on the rising edge of the ninth SCL clock cycle when the following R/W# bit is 0, and then compares the second and subsequent bytes with its own slave address. If the address matches the value in the slave address register, the RIIC sets the corresponding AASy flag (y = 0 to 2) in ICSR1 to 1.

After that, when the first byte received after a start or restart condition is issued 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 clears 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 = 1.

Furthermore, prepare the device-ID fields (three 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.

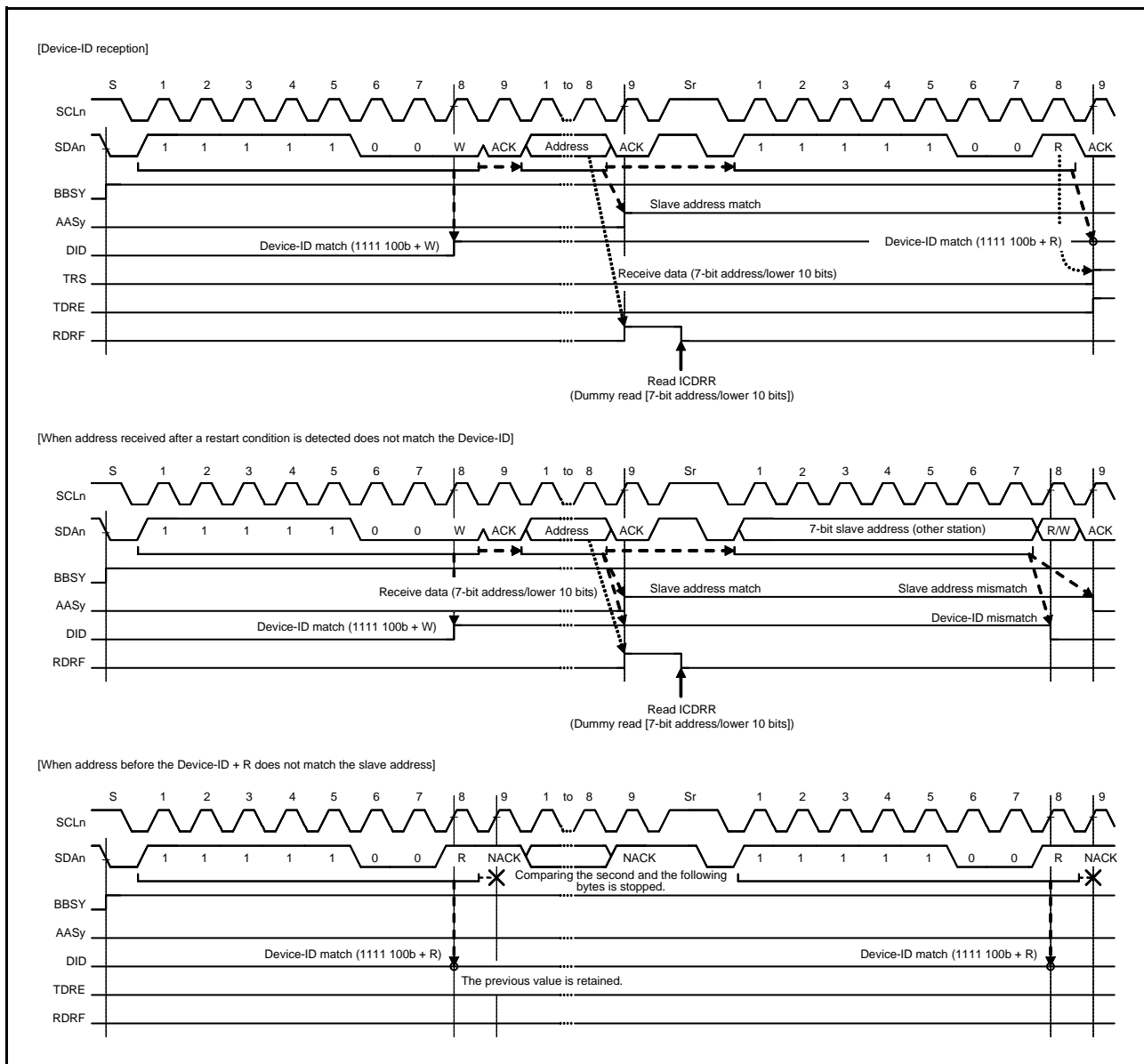


Figure 33.27 AASy/DID Flag Set/Clear Timing during Reception of Device-ID

### 33.7.4 Host Address Detection

The RIIC has a function to detect the host address while the SMBus is operating. When the HOAE bit in ICSER is set to 1 while the SMBS bit in ICMR3 is 1, the RIIC can detect the host address (0001 000b) in slave receive mode (MST and TRS bits = 00b in ICCR2).

When the RIIC detects the host address, the HOA flag in ICSR1 is set to 1 at the falling edge of the ninth SCL clock cycle, and at the same time, the TDRE flag in ICSR2.RDRF is set to 1 when the R/W# bit is 0 (Wr bit). This causes a receive data full interrupt (ICRXI) 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 = 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.

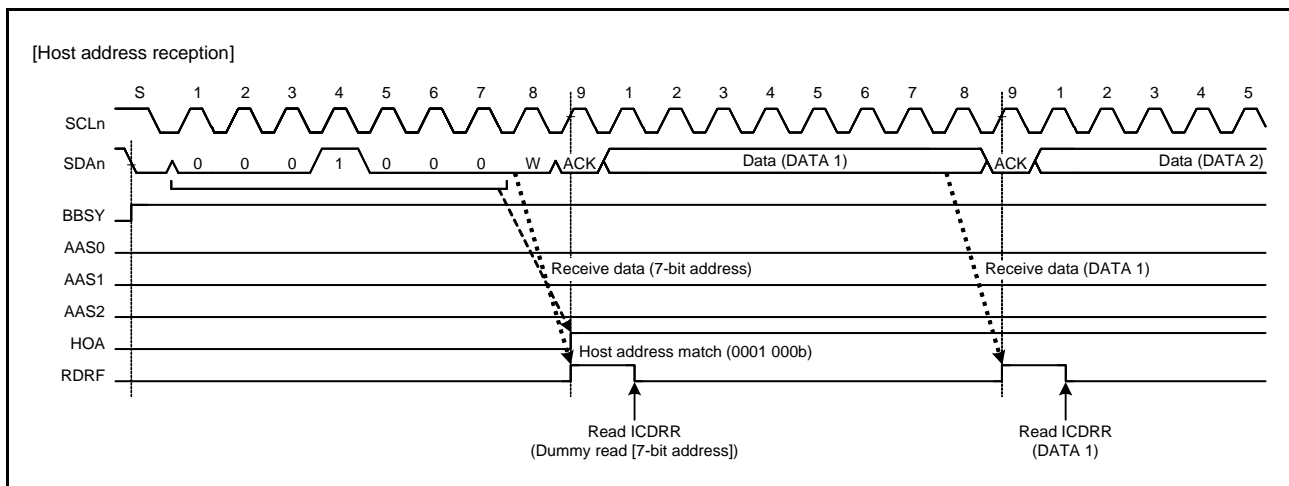


Figure 33.28 HOA Flag Set Timing during Reception of Host Address

### 33.8 Automatically Low-Hold Function for SCL

#### 33.8.1 Function to Prevent Wrong Transmission of Transmit Data

If the shift register (ICDRS) is empty when data have not been written to the transmit data register (ICDRT) with the RIIC in transmission mode (TRS bit = 1 in ICCR2), the SCLn signal is automatically held at the low level over the intervals shown below. This low-hold period is extended until data for transmission have been written, which prevents the unintended transmission of erroneous data.

##### <Master transmit mode>

- Low-level interval after a start condition or restart condition is issued
- Low-level interval of one clock cycle between the ninth clock cycle of one transfer and the first clock cycle of the next

##### <Slave transmit mode>

- Low-level interval between the ninth clock cycle of one transfer and the first clock cycle of the next

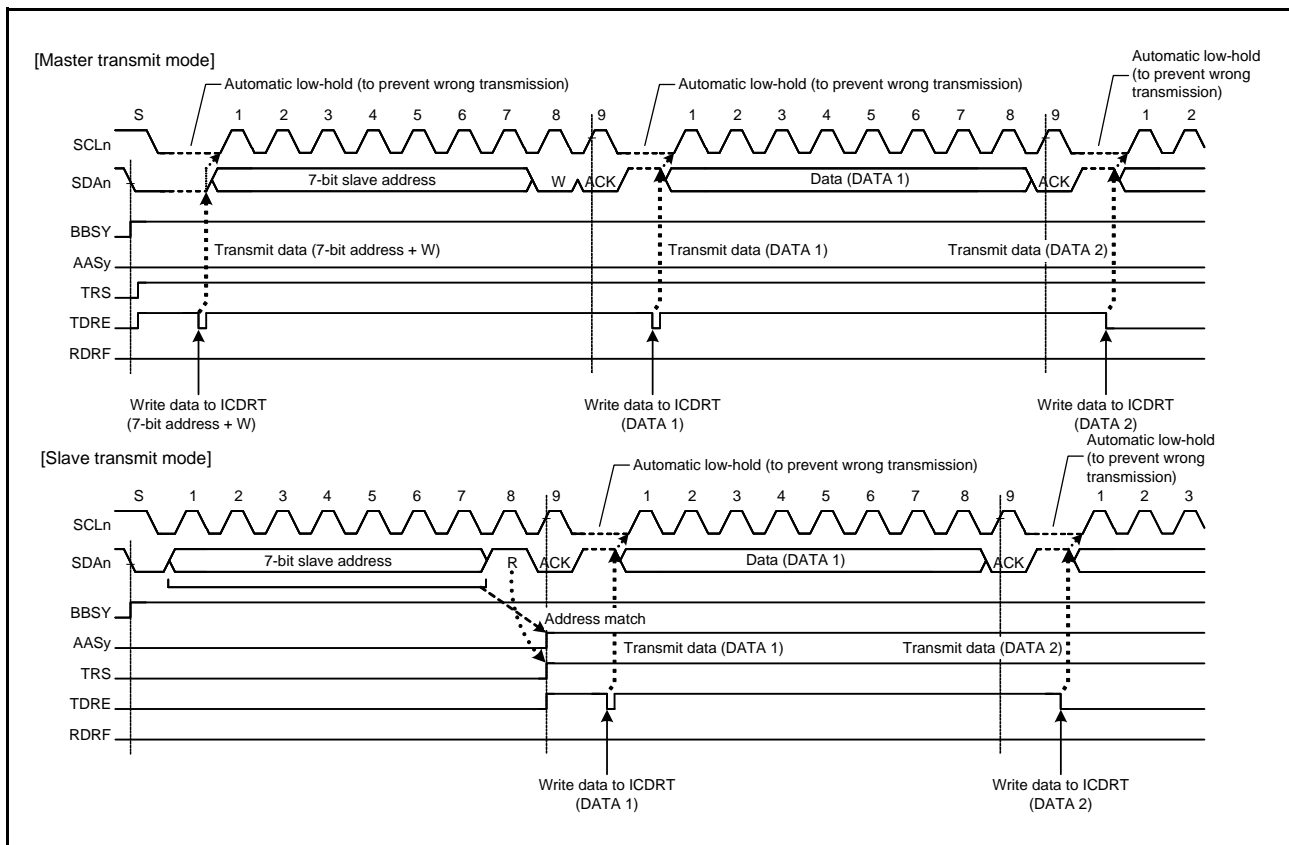


Figure 33.29 Automatic Low-Hold Operation in Transmit Mode

### 33.8.2 NACK Reception Transfer Suspension Function

The RIIC has a function to suspend transfer operation when NACK is received in transmit mode (TRS bit = 1 in ICCR2). This function is enabled when the NACKEN bit in ICFER is set to 1 (transfer suspension enabled). If the next transmit data has already been written (TDRE flag = 0 in ICSR2) when NACK is received, next data transmission at the falling edge of the ninth SCL clock cycle is automatically suspended. This prevents the SDAAn line output level from being held low when the MSB of the next transmit data is 0.

If the transfer operation is suspended by this function (NACKF flag = 1 in ICSR2), transmit operation and receive operation are discontinued. To restore transmit/receive operation, be sure to clear the NACKF flag to 0. In master transmit mode, clear the NACKF flag to 0, issue a restart or stop condition, and then issue a start condition again.

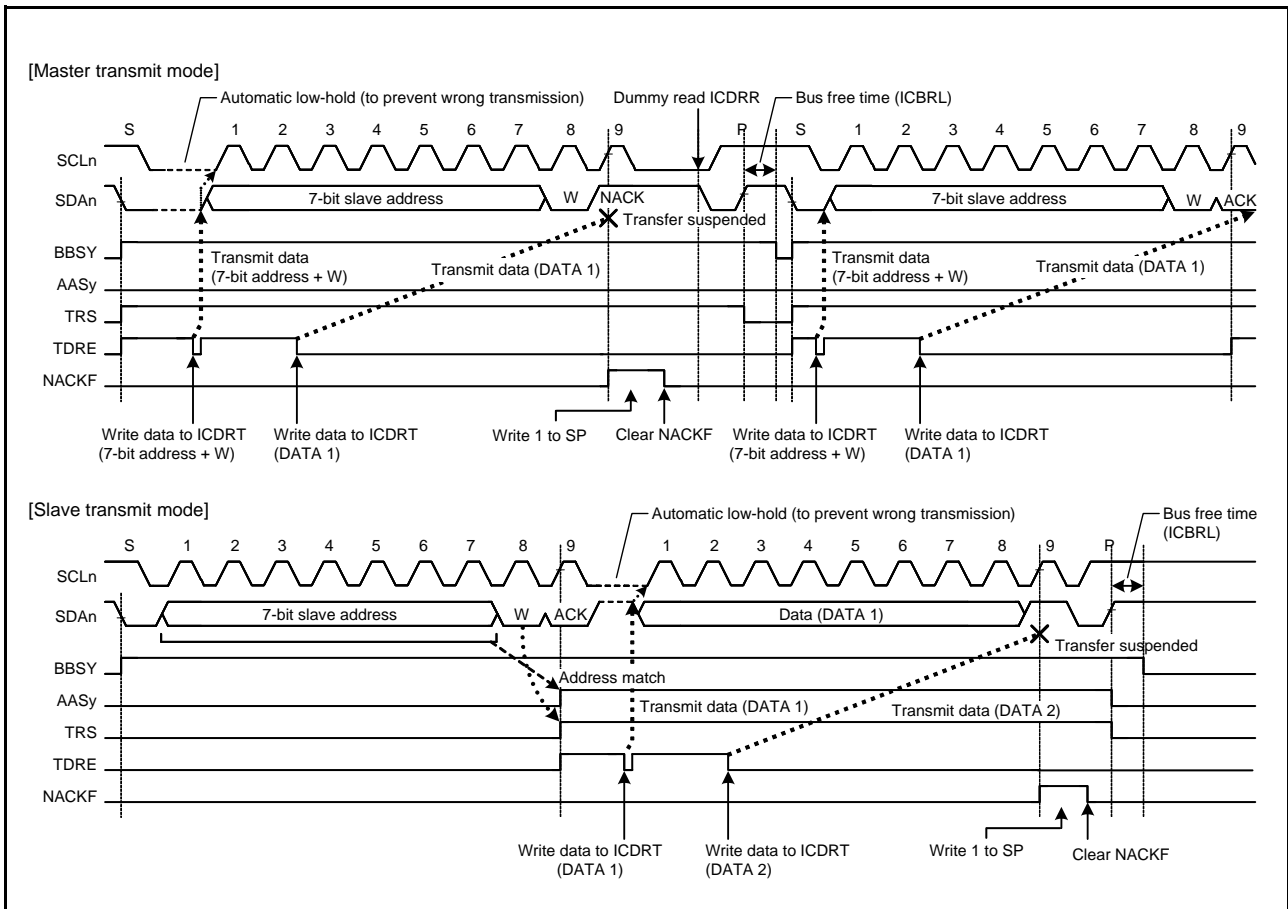


Figure 33.30 Suspension of Data Transfer when NACK is Received (NACK = 1)

### 33.8.3 Function to Prevent Failure to Receive Data

If response processing is delayed when receive data (ICDRR) read is delayed for a period of one transfer frame or more with receive data full (RDRF flag = 1 in ICSR2) in receive mode (TRS = 0 in ICCR2), the RIIC holds the SCLn 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 issued. This function does not disturb other communication because the RIIC does not hold the SCLn line low when a mismatch with its own slave address occurs after a stop condition is issued.

Sections in which the SCLn line is held low can be selected with a combination of the WAIT and RDRFS bits in ICMR3.

#### (1) One-Byte Receive Operation and Automatic Low-Hold Function Using the WAIT Bit

When the WAIT bit in ICMR3 is set to 1, the RIIC performs one-byte receive operation using the WAIT bit function. Furthermore, when the ICMR3.RDRFS bit is 0, the RIIC automatically sends the ACKBT bit value in ICMR3 for the acknowledge bit in the period from the falling edge of the eighth SCL clock cycle to the falling edge of the ninth SCL clock cycle, and automatically holds the SCLn line low at the falling edge of the ninth SCL clock cycle using the WAIT bit function. This low-hold is released by reading data from ICDRR, which enables byte-wise receive operation. The WAIT bit function is enabled for receive frames after a match with the RIIC's own slave address (including the general call address and host address) is obtained in master receive mode or slave receive mode.

(2) One-Byte Receive Operation (ACK/NACK Transmission Control) and Automatic Low-Hold Function Using the RDRFS Bit

When the RDRFS bit in ICMR3 is set to 1, the RIIC performs one-byte receive operation using the RDRFS bit function. When the RDRFS bit is set to 1, the RDRF flag (receive data full) in ICSR2 is set to 1 at the rising edge of the eighth SCLn clock cycle, and the SCLn line is automatically held low at the falling edge of the eighth SCL clock cycle. This low-hold is released by writing a value to the ACKBT bit in ICMR3, but cannot be released by reading data from ICDRR, which enables receive operation by the ACK/NACK transmission control according to the data received in byte units. The RDRFS bit function is enabled for receive frames 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.

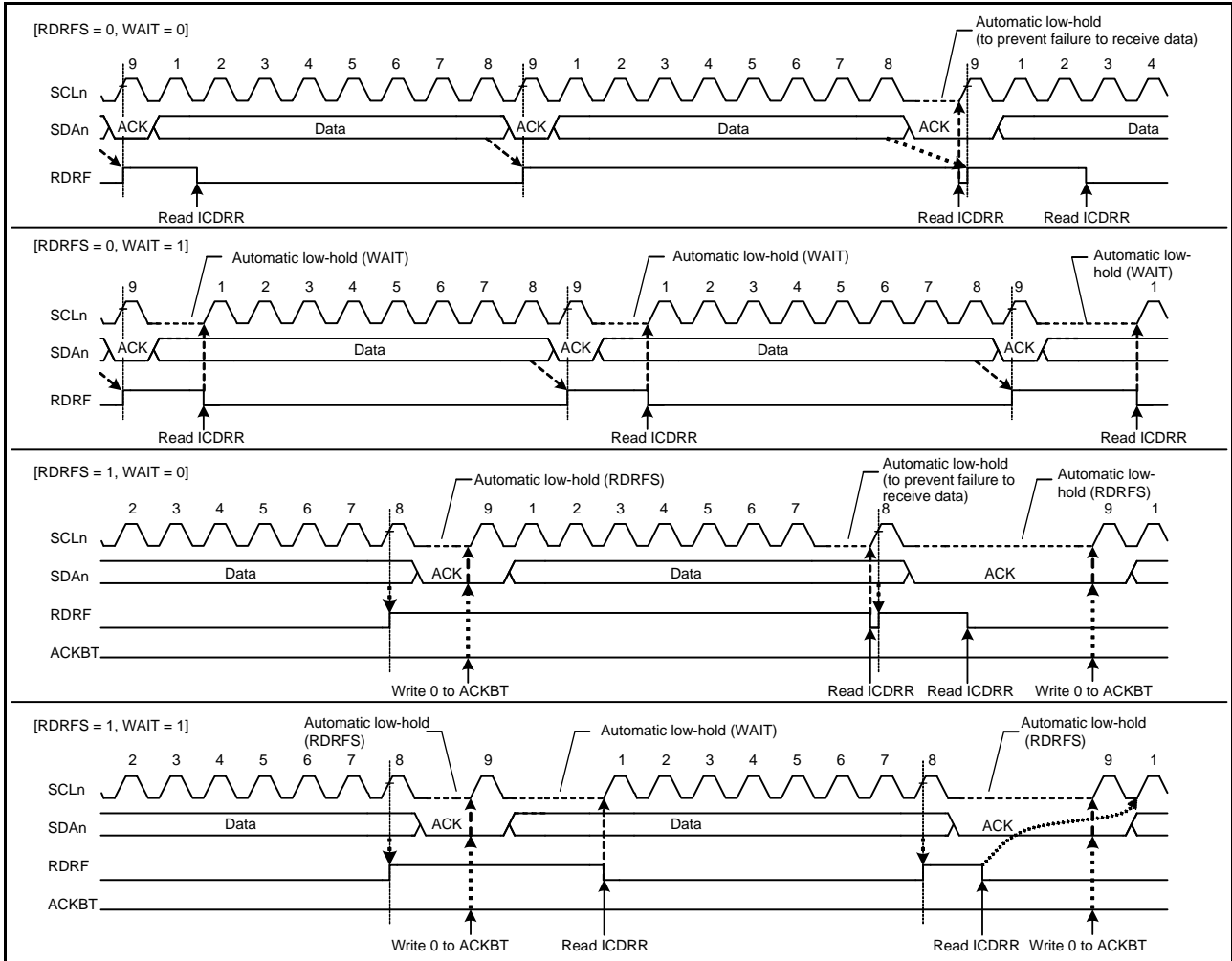


Figure 33.31 Automatic Low-Hold Operation in Receive Mode (Using RDRFS and WAIT Bits)



### 33.9 Arbitration-Lost Detection Functions

In addition to the normal arbitration-lost detection function defined by the I<sup>2</sup>C bus standard, the RIIC has functions to prevent double-issue of a start condition, to detect arbitration-lost during transmission of NACK, and to detect arbitration-lost in slave transmit mode.

#### 33.9.1 Master Arbitration-Lost Detection (MALE Bit)

The RIIC drives the SDA<sub>n</sub> line low to issue a start condition. However, if the SDA<sub>n</sub> line has already been driven low by another master device issuing a start condition, the RIIC regards its own issuing of a start condition as an error and considers this a loss in arbitration, so priority is given to transfer by the other master device. Similarly, if a request to issue a start condition is made by setting the ST bit in ICCR2 to 1 while the bus is busy (BBSY flag = 1 in ICCR2), the RIIC regards this as a double-issuing-of-start-condition error and considers itself to have lost in arbitration, thus preventing a failure of transfer due to issuing of a start condition while transfer is in progress.

When a start condition is issued successfully, if the data for transmission including the address bits (i.e. the internal SDA output level) and the level on the SDA<sub>n</sub> line do not match (the high output as the internal SDA output; i.e. the SDA<sub>n</sub> pin is in the high-impedance state) and the low level is detected on the SDA 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 MALE bit in ICFER is 1 (master arbitration-lost detection enabled).

[Master arbitration-lost conditions]

- Non-matching of the internal level for output on SDA and the level on the SDA<sub>n</sub> line after a start condition was issued by setting the ST bit in ICCR2 to 1 while the BBSY flag in ICCR2 was cleared to 0 (erroneous issuing of a start condition)
- Setting of the ST bit in ICCR2 to 1 (start condition double-issue error) while the BBSY flag is set to 1
- When the transmit data excluding acknowledge (internal SDA output level) does not match the level on the SDA<sub>n</sub> line in master transmit mode (MST and TRS bits = 11b in ICCR2)

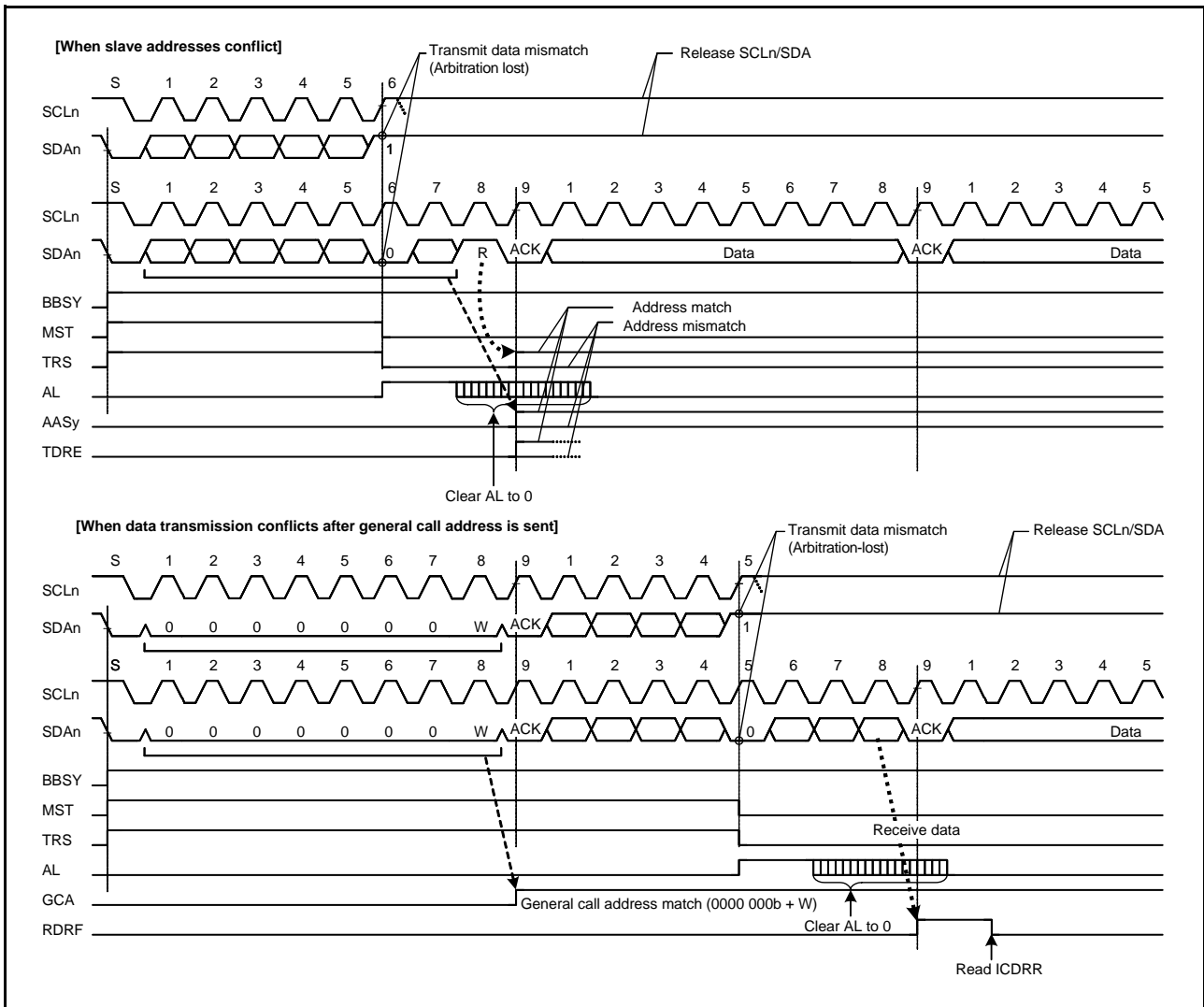


Figure 33.32 Examples of Master Arbitration-Lost Detection (MALE = 1)

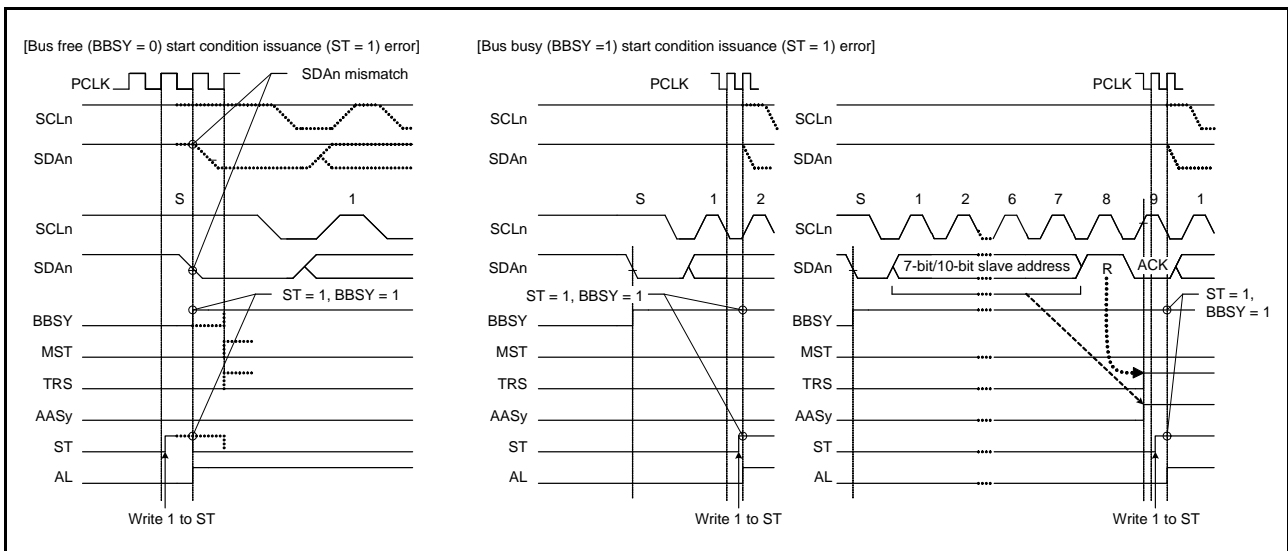
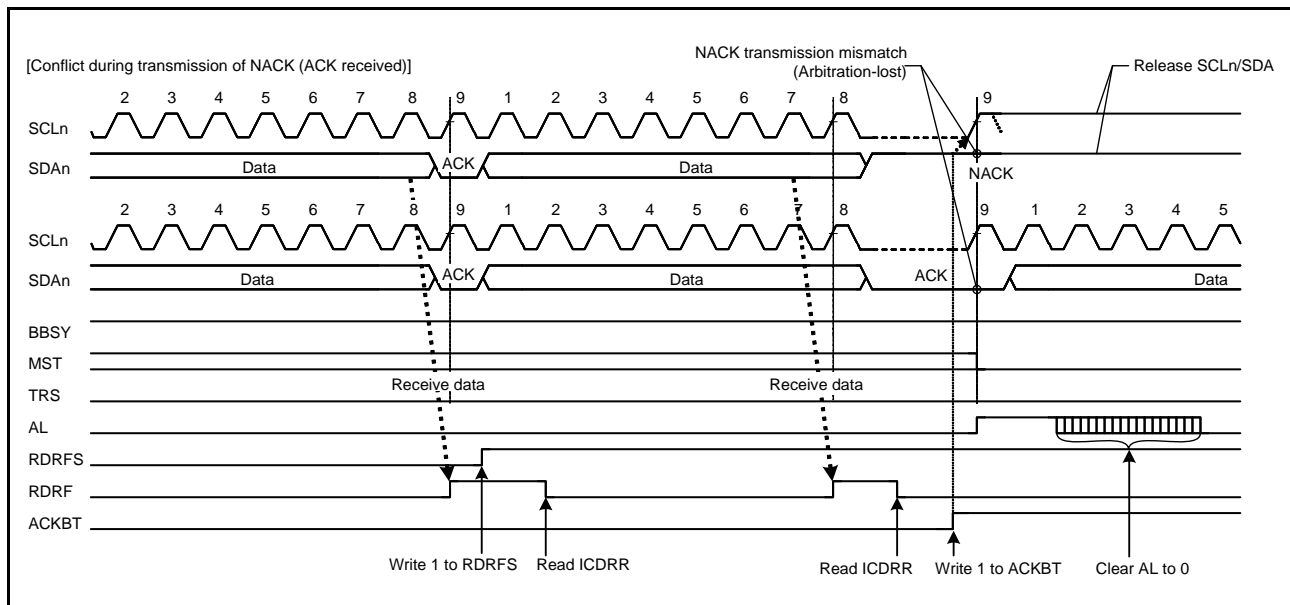


Figure 33.33 Arbitration-Lost when a Start Condition is Issued (MALE = 1)

### 33.9.2 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 SDA<sub>n</sub> line (the high output as the internal SDA output; i.e. the SDA<sub>n</sub> pin is in the high-impedance state) and the low level is detected on the SDA 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 33.34 shows an example of arbitration-lost detection during transmission of NACK.



**Figure 33.34 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 two bytes of data from the slave device, and master B receives four 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 two final bytes of data from the slave device. Meanwhile, master B sends ACK because it has not received necessary four 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 issues a stop condition. Therefore, the issuance of the stop condition conflicts with the SCL clock 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 issued, preventing a communication failure on the bus. Similarly, in the ARP command processing of SMBus, the function to detect loss of arbitration during transmission of NACK is also available for eliminating the extra clock cycle processing (such as 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 NALE bit in ICFER set to 1 (arbitration-lost detection during NACK transmission enabled).

[Condition for arbitration-lost during NACK transmission]

- When the internal SDA output level does not match the SDA<sub>n</sub> line (ACK is received) during transmission of NACK (ACKBT bit = 1 in ICMR3)

### 33.9.3 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 SDA<sub>n</sub> line do not match (the high output as the internal SDA output; i.e. the SDA<sub>n</sub> pin is in the high-impedance state) and the low level is detected on the SDA 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 eliminates subsequent redundant processing (processing for the transmission of FFh).

The RIIC detects slave arbitration-lost when the following condition is met with the SALE bit in ICFER set to 1 (slave arbitration-lost detection enabled).

[Condition for slave arbitration-lost]

- When transmit data excluding acknowledge (internal SDA output level) does not match the SDA<sub>n</sub> line in slave transmit mode (MST and TRS bits = 01b in ICCR2)

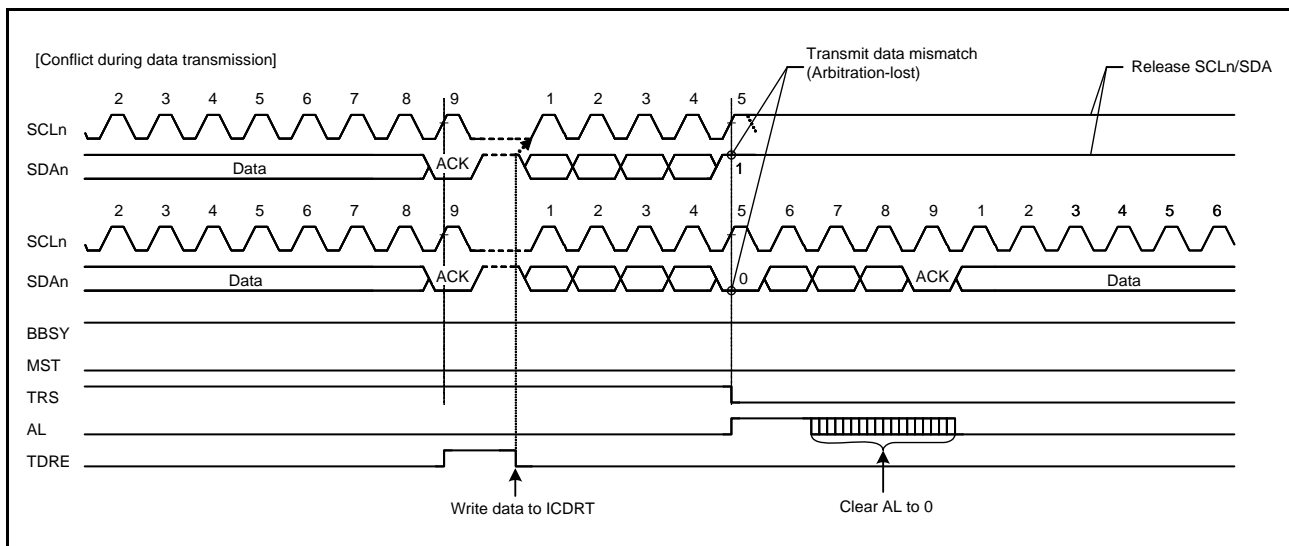


Figure 33.35 Example of Slave Arbitration-Lost Detection (SALE = 1)

### 33.10 Start Condition/Restart Condition/Stop Condition Issuing Function

#### 33.10.1 Issuing a Start Condition

The RIIC issues a start condition when the ST bit in ICCR2 is set to 1.

When the ST bit is set to 1, a start condition issuance request is made and the RIIC issues a start condition when the BBSY flag in ICCR2 is 0 (bus free). When a start condition is issued normally, the RIIC automatically shifts to the master transmit mode.

A start condition is issued in the following sequence.

[Start condition issuance]

- (1) Drive the SDA<sub>n</sub> line low (high level to low level).
- (2) Ensure the time set in ICBRH and the start condition hold time.
- (3) Drive the SCL<sub>n</sub> line low (high level to low level).
- (4) Detect low level of the SCL<sub>n</sub> line and ensure the low-level period of SCL<sub>n</sub> line set in ICBRL.

#### 33.10.2 Issuing a Restart Condition

The RIIC issues a restart condition when the RS bit in ICCR2 is set to 1.

When the RS bit is set to 1, a restart condition issuance request is made and the RIIC issues a restart condition when the BBSY flag in ICCR2 is 1 (bus busy) and the MST bit in ICCR2 is 1 (master mode).

A restart condition is issued in the following sequence.

[Restart condition issuance]

- (1) Release the SDA<sub>n</sub> line.
- (2) Ensure the low-level period of SCL<sub>n</sub> line set in ICBRL.
- (3) Release the SCL<sub>n</sub> line (low level to high level).
- (4) Detect a high level of the SCL<sub>n</sub> line and ensure the time set in ICBRL and the restart condition setup time.
- (5) Drive the SDA<sub>n</sub> line low (high level to low level).
- (6) Ensure the time set in ICBRH and the restart condition hold time.
- (7) Drive the SCL<sub>n</sub> line low (high level to low level).
- (8) Detect a low level of the SCL<sub>n</sub> line and ensure the low-level period of SCL<sub>n</sub> line set in ICBRL.

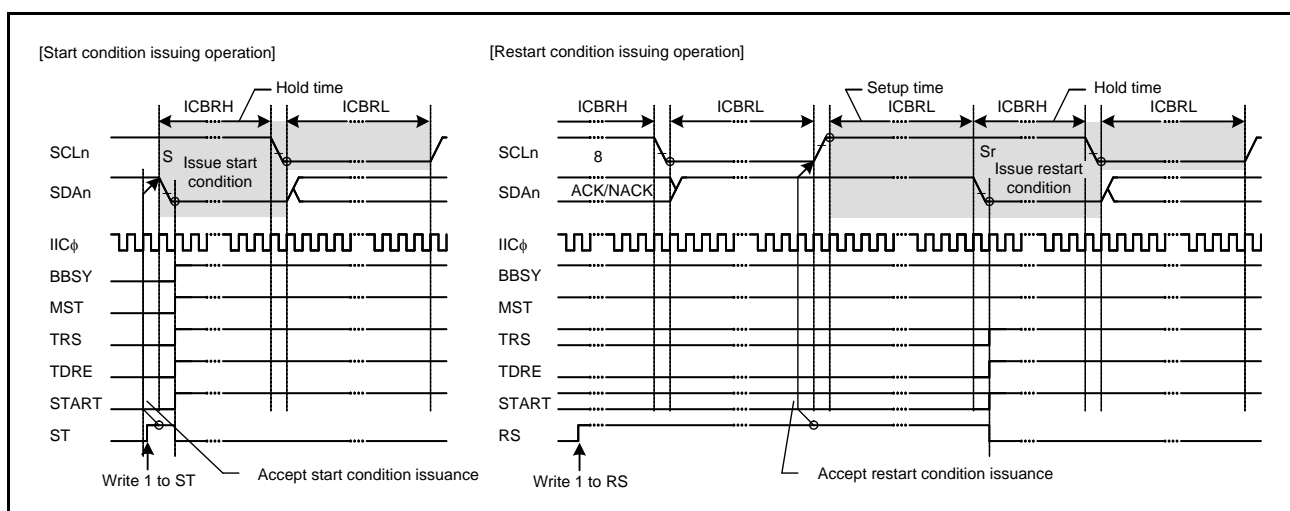


Figure 33.36 Start Condition/Restart Condition Issue Timing (ST and RS Bits)

### 33.10.3 Issuing a Stop Condition

The RIIC issues a stop condition when the SP bit in ICCR2 is set to 1.

When the SP bit is set to 1, a stop condition issuance request is made and the RIIC issues a stop condition when the BBSY flag in ICCR2 is 1 (bus busy) and the MST bit in ICCR2 is 1 (master mode).

A stop condition is issued in the following sequence.

[Stop condition issuance]

- Drive the SDA<sub>n</sub> line low (high level to low level).
- Ensure the low-level period of SCL<sub>n</sub> line set in ICBRL.
- Release the SCL<sub>n</sub> line (low level to high level).
- Detect a high level of the SCL<sub>n</sub> line and ensure the time set in ICBRH and the stop condition setup time.
- Release the SDA<sub>n</sub> line (low level to high level).
- Ensure the time set in ICBRL and the bus free time.
- Clear the BBSY flag to 0 (to release the bus mastership).

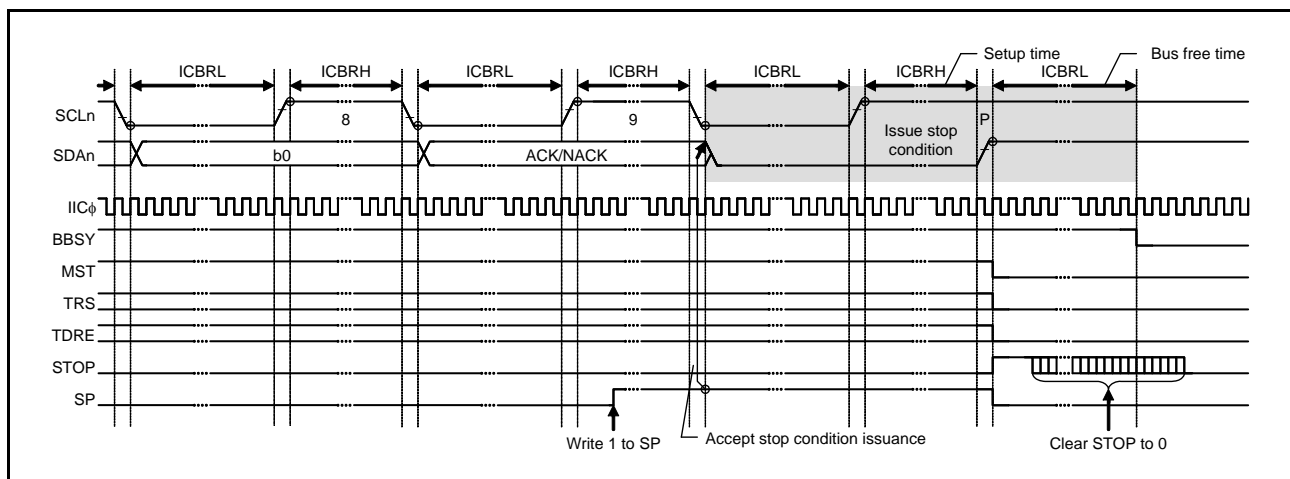


Figure 33.37 Stop Condition Issue Timing (SP Bit)

### 33.11 Bus Hanging

If the clock signals from the master and slave devices go out of synchronization due to noise or other factors, the I<sup>2</sup>C bus might hang with a fixed level on the SCL<sub>n</sub> line and/or SDA<sub>n</sub> line.

As a measure against the bus hanging, the RIIC has a timeout function to detect hanging by monitoring the SCL<sub>n</sub> line, a function for the output of an extra SCL clock cycle to release the bus from a hung state due to clock signals being out of synchronization, and the RIIC/internal reset function.

By checking the SCLO, SDAO, SCLI, and SDAI bits in ICCR1, it is possible to see whether the RIIC or its partner in communications is placing the low level on the SCL<sub>n</sub> or SDA<sub>n</sub> lines.

### 33.11.1 Timeout Function

The RIIC has the timeout function to detect an abnormality that the SCLn line is held for a certain period of time. In the bus busy state, the RIIC can detect an abnormal bus state by monitoring that the SCLn line is held low or high for a predetermined time.

The timeout function monitors the SCLn line state and counts the low-level period or high-level period using the internal counter. The timeout function resets the internal counter each time the SCLn line changes (rising or falling), but continues to count unless the SCLn line changes. If the internal counter overflows due to no SCLn line change, the RIIC can detect the timeout and report the bus abnormality.

This timeout function is enabled when the TMOE bit in ICFER is 1. It detects an abnormal bus state that the SCLn line is held low or high when the bus is busy (BBSY flag = 1 in ICCR2) in master mode or when the BBSY flag is 1 and the RIIC's own slave address matches (ICSR1 is not 00h) in slave mode.

The internal counter of the timeout function works using the internal reference clock (IICφ) set by the CKS[2:0] bits in ICMR1 as a count source. It functions as a 16-bit counter when long mode is selected (TMOS bit = 0 in ICMR2) or a 14-bit counter when short mode is selected (TMOS bit = 1).

The SCLn line level (low/high or both levels) during which this counter is activated can be selected by the setting of the TMOH and TMOL bits in ICMR2. If both TMOL and TMOH bits are cleared to 0, the internal counter does not work.

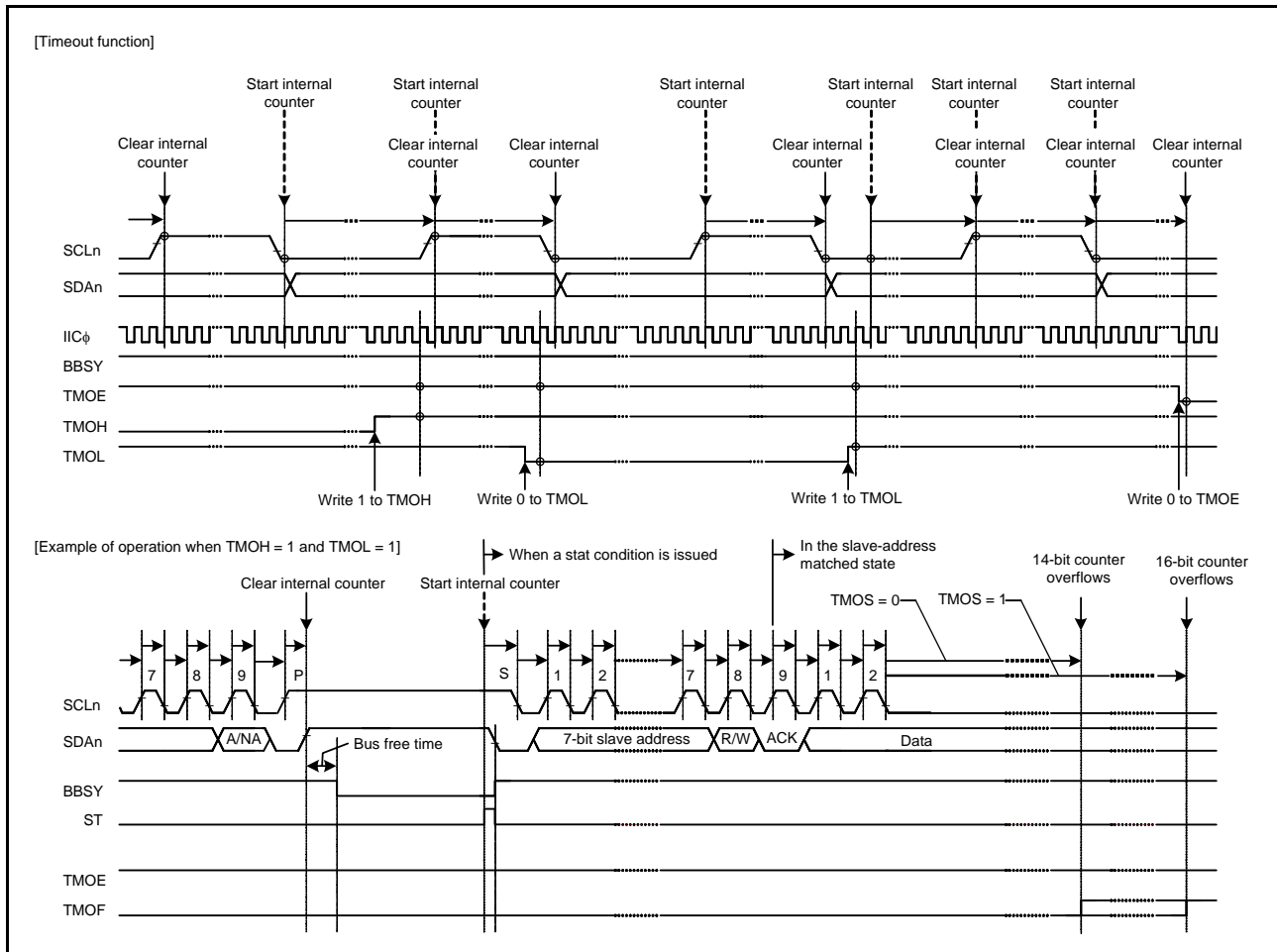


Figure 33.38 Timeout Function (TMOE, TMOS, TMOH, and TMOL Bits)

### 33.11.2 Extra SCL Clock Cycle Output Function

In master mode, the RIIC module has a facility for the output of extra SCL (clock) cycles to release the SDAn line of the slave device from being held at the low level due to the master being out of synchronization with the slave device.

This function is mainly used in master mode to release the SDAn line of the slave device from the state of being fixed to the low level by including extra cycles of SCLn output from the RIIC with single cycles of the SCL (clock) signal as the unit in the case of a bus error where the RIIC cannot issue a stop condition because the slave device is holding the SDAn line at the low level. Do not use this facility in normal situations. Using it when communications are proceeding correctly will lead to malfunctions.

When the CLO bit in ICCR1 is set to 1 in master mode, a single cycle of the SCL clock at the frequency corresponding to the transfer rate settings (settings of the CKS[2:0] bits in ICMR1, and of the ICBRH and ICBRL registers) is output as an extra clock cycle. After output of this single cycle of the SCL clock, the CLO bit is automatically cleared to 0. Therefore, further extra clock cycles can be output consecutively by the software program writing 1 to the CLO bit after having read CLO = 0.

When the RIIC module is in master mode and the slave device is holding the SDAn line at the low level 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 facility for output of an extra cycle of the SCL (clock) signal can be used to output extra cycles of SCL one by one to make the slave device release the SDAn line from being held at the low level, thus recovering the bus from an unusable state. Release of the SDAn line by the slave device can be monitored by reading the SDAI bit in ICCR1. After confirming release of the SDAn line by the slave device, complete communications by reissuing the stop condition. Use this facility with the MALE bit (master arbitration-lost detection disabled) in ICFER cleared to 0. If the MALE bit is set to 1 (master arbitration-lost detection enabled), arbitration is lost when the value of the SDAO bit in ICCR1 does not match the state of the SDAn line, so take care on this point.

[Output conditions for using the CLO bit in ICCR1]

- When the bus is free (BBSY flag in ICCR2 = 0) or in master mode (MST bit = 1 and BBSY flag = 1 in ICCR2)
- When the communication device does not hold the SCLn line low

Figure 33.39 shows the operation timing of the extra SCL clock cycle output function (CLO bit).

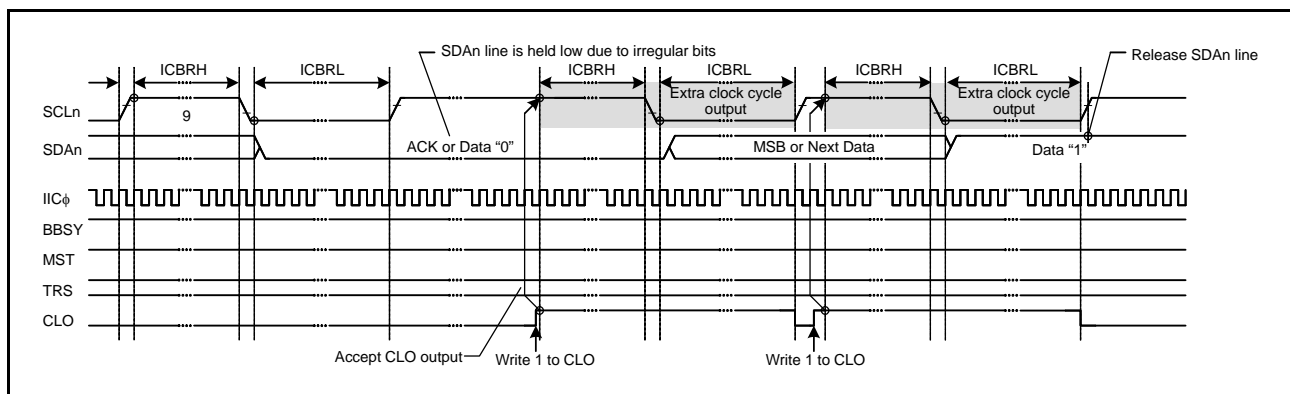


Figure 33.39 Extra SCL Clock Cycle Output Function (CLO Bit)



### 33.11.3 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 BBSY flag in ICCR2. 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 issuing a reset, be sure to clear the IICRST bit in ICCR1 to 0.

Both types of reset are effective for release from bus-hung states since both restore the output state of the SCLn and SDAn pins to the high impedance state.

Issuing 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 (ICE and IICRST bits = 01b in ICCR1).

For a detailed description of the RIIC and internal resets, see section 33.14, Reset States.

## 33.12 SMBus Operation

The RIIC is available for data communication conforming to the SMBus (Version 2.0). To perform SMBus communication, set the SMBS bit in ICMR3 to 1. To use the transfer rate within a range of 10 kbps to 100 kbps of the SMBus standard, set the CKS[2:0] bits in ICMR1, ICBRH, and ICBRL. In addition, determine the values of the DLCS bit in ICMR2 and the SDDL[2:0] bits in ICMR2 to meet the data hold time specification of 300 ns or more. If the RIIC is used only as a slave device, the transfer rate setting is not necessary. When the RIIC is used only as a slave device, the transfer rate setting is not necessary, whereas the ICBRL 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 (SARL0, SARL1, and SARL2), and set the corresponding FS bit (7-bit/10-bit address format select) in SARUy (y = 0 to 2) to 0 (7-bit address format).

When transmitting the UDID (Unique Device Identifier), set the SALE bit in ICFER to 1 to enable the slave arbitration-lost detection function.

### 33.12.1 SMBus Timeout Measurement

#### (1) Measuring timeout of slave device

The following period (timeout interval:  $T_{\text{LOW:SEXT}}$ ) 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 or TMR 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 total clock low-level period [slave device]  $T_{\text{LOW:SEXT}}$ : 25 ms (max.) of the SMBus standard.

If the time measured with the MTU or TMR exceeds the clock low-level detection timeout  $T_{\text{TIMEOUT}}$ : 25 ms (min.) of the SMBus standard, the slave device must release the bus by writing 1 to the IICRST bit in ICCR1 to issue an internal reset of the RIIC. When an internal reset is issued, the RIIC stops driving the bus for the SCLn pin and SDAn pin and make the SCLn/SDAn pin outputs high impedance, which releases the bus.

(2) Measuring timeout of master device

The following periods (timeout interval: T<sub>LOW:MEXT</sub>: MEXT) must be measured for master devices in SMBus communication.

- From start condition to acknowledge bit
- Between acknowledge bits
- From acknowledge bit to stop condition

To measure timeout for master devices, measure these periods with the MTU or TMR timer using a start condition detection interrupt (STI), stop condition detection interrupt (SPI), and transmit end interrupt (ICTEI) or receive data full interrupt (ICRXI) of the RIIC. The measured timeout period must be within the total clock low-level extended period [master device] T<sub>LOW:MEXT</sub>: 10 ms (max.) of the SMBus standard, and the total of all T<sub>LOW:MEXT</sub> from start condition to stop condition must be within T<sub>LOW:SEXT</sub>: 25 ms (max.).

For the ACK receive timing (rising edge of the ninth SMBCLK clock cycle), monitor the TEND flag in ICSR2 in master transmit mode (master transmitter) and the RDRF flag in ICSR2 in master receive mode (master receiver). For this reason, perform bitwise transmit operation in master transmit mode, and hold the RDRFS bit in ICMR3 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 SMBCLK clock cycle.

If the period measured with the MTU or TMR exceeds the total clock low-level extended period [master device] T<sub>LOW:MEXT</sub>: 10 ms (max.) of the SMBus standard or the total of measured periods exceeds the clock low-level detection timeout T<sub>TIMEOUT</sub>: 25 ms (min.) of the SMBus standard, the master device must stop the transaction by issuing a stop condition. In master transmit mode, immediately stop the transmit operation (writing data to ICDRT).

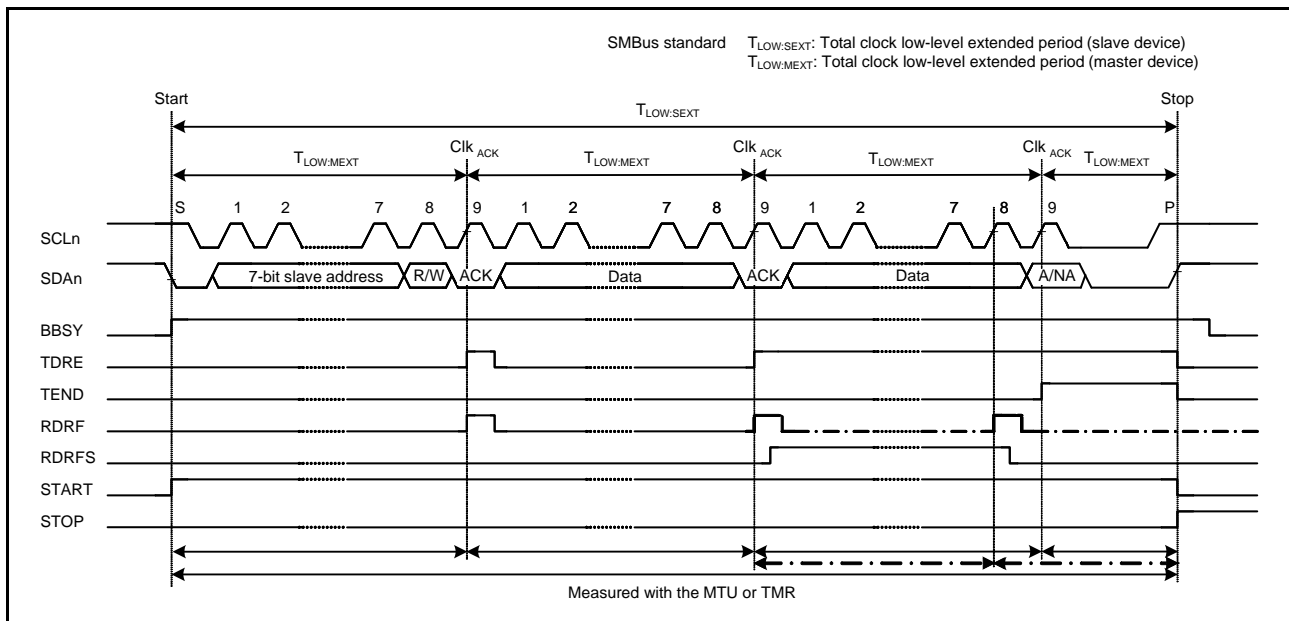


Figure 33.40 SMBus Timeout Measurement

### 33.12.2 Packet Error Code (PEC)

The RX630 Group 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, see section 37, CRC Calculator (CRC).

The PEC data in master transmit mode (master transmitter) 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 (master receiver) 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 RDRFS bit in ICMR3 to 1 before the rising edge of the eighth SMBCLK clock cycle during reception of the final byte, and hold the SCLn line low at the falling edge of the eighth clock cycle.

### 33.12.3 SMBus Host Notification Protocol/Notify ARP Master

In communications over an SMBus, a slave device can temporarily act as a master device to notify the SMBus host (or ARP master) of (or request the SMBus host for) its own slave address or to request its own slave address from the SMBus host.

For a product of the RX630 Group 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 SMBS bit in ICMR3 and the HOAE bit in ICSE to 1. Operation after the host address has been detected is the same as normal slave operation.

### 33.13 Interrupt Request

The RIIC issues four types of interrupt request: transfer error or event generation (arbitration-lost, NACK detection, timeout detection, start condition detection, and stop condition detection), receive end, transmit data empty, and transmit end.

Table 33.6 lists details of the several interrupt requests. The receive data full and transmit data empty are both capable of launching data transfer by the DTC or DMAC.

**Table 33.6 Interrupt Sources**

Symbol	Interrupt Source	Interrupt Flag	DTC Launching	DMACA Launching	Priority	Interrupt Condition
ICEEI	Transfer Error/ Event Generation	AL	Not possible	Not possible	High	AL = 1 • ALIE = 1
		NACKF				NACKF = 1 • NAKIE = 1
		TMOF				TMOF = 1 • TMOIE = 1
		START				START = 1 • STIE = 1
		STOP				STOP = 1 • SPIE = 1
ICRXI	Receive End	—	Possible	Possible	RDRF = 1 • RIE = 1	
ICTXI	Transmit Data Empty	—	Possible	Possible	TDRE = 1 • TIE = 1	
ICTEI	Transmit End	TEND	Not possible	Not possible	Low	TEND = 1 • TEIE = 1

Clear or mask the each flag during interrupt handling.

Notes on interrupt processing:

1. There is a latency (delay) 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 masked, read the relevant flag again to check whether clearing or masking has been completed, and then return from interrupt processing. Returning from interrupt processing without checking that writing to the module has been completed creates a possibility of repeated processing of the same interrupt.
2. Since ICTXI is an edge-detected interrupt, it does not require clearing. Furthermore, the TDRE flag in ICSR2 (a condition for ICTXI) is automatically cleared to 0 when data for transmission are written to ICDRT or a stop condition is detected (STOP flag = 1 in ICSR2).
3. Since ICRXI is an edge-detected interrupt, it does not require clearing. Furthermore, the RDRF flag in ICSR2 (a condition for ICRXI) is automatically cleared to 0 when data are read from ICDRR.
4. When using the ICTEI interrupt, clear the TEND flag in ICSR2 in the ICTEI interrupt processing.  
Note that the TEND flag in ICSR2 is automatically cleared to 0 when data for transmission are written to ICDRT or a stop condition is detected (STOP flag = 1 in ICSR2).

#### 33.13.1 Buffer Operation for ICTXI and ICRXI Interrupts

For the ICTXI and ICRXI interrupts, as well as the ICUIRn.IR flag having the value 1 being a condition for issuing the interrupts, the interrupt request is retained within and not output by the ICU (the capacity for internally retaining the interrupts is one request per source).

An interrupt request that was being retained within the ICU is output when the value of the ICUIRn.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 interrupt enable bit within the given peripheral module.

### 33.14 Reset States

The RIIC has chip reset, RIIC reset, and internal reset functions. Table 33.7 lists the scope of each reset and reset conditions.

**Table 33.7 Reset Conditions**

		Chip Reset	RIIC Reset (ICE = 0, IICRST = 1)	Internal Reset (ICE = 1, IICRST = 1)	Start Condition/ Restart Condition Detection	Stop Condition Detection	
ICCR1	ICE, IICRST	At a reset	Retained	Retained	Operation (retained)	Operation (retained)	
	SCLO, SDAO		At a reset	At a reset			
	Others			Retained			
ICCR2	BBSY	At a reset	At a reset	Operation	Operation	Operation	
	ST			At a reset	At a reset	Operation (retained)	
	Others					At a reset	
ICMR1	BC[2:0]	At a reset	At a reset	At a reset	At a reset	Operation (retained)	
	Others			Retained	Operation (retained)		
ICMR2		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)	
ICMR3		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)	
ICFER		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)	
ICSER		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)	
ICIER		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)	
ICSR1		At a reset	At a reset	At a reset	Operation (retained)	At a reset	
ICSR2	TDRE, TEND	At a reset	At a reset	At a reset	Operation (retained)	At a reset	
	START				Operation		
	STOP				Operation (retained)		Operation
	Others						Operation (retained)
SARL0, 1, 2 SARU0, 1, 2		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)	
ICBRH, ICBRL		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)	
ICDRT		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)	
ICDRR		At a reset	At a reset	Retained	Operation (retained)	Operation (retained)	
ICDRS		At a reset	At a reset	At a reset	Operation (retained)	Operation (retained)	
Timeout function		At a reset	At a reset	Operation	Operation	Operation	
Bus free time measurement		At a reset	At a reset	Operation	Operation	Operation	

## 33.15 Usage Notes

### 33.15.1 Setting Module-Stop Function

Module-stop state can be entered or canceled using module stop control register B (MSTPCRB) or module stop control register C (MSTPCRC). The initial setting is for operation of the RIIC to be stopped. RIIC register access is enabled by clearing module-stop state.

For details of module stop control registers B and C, see section 11, Low Power Consumption.

### 33.15.2 Points to Note on Starting Transfer

If the ICU.IRn.IR flag is 1 at the time transfer is to be started (by setting the ICCR1.ICE bit to 1), follow the procedure below to clear interrupts before enabling operations. Starting transfer with the ICU.IRn.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 ICU.IRn.IR flag.

- (1) Confirm that the ICCR1.ICE bit is 0.
- (2) Clear the relevant interrupt enable bits (ICIER1.TIE, etc.) on the peripheral function side to 0.
- (3) Read the relevant interrupt enable bits (ICIER1.TIE, etc.) on the peripheral function side and confirm that its value is 0.
- (4) Clear the ICU.IRn.IR flag to 0.

## 34. CAN Module (CAN)

### 34.1 Overview

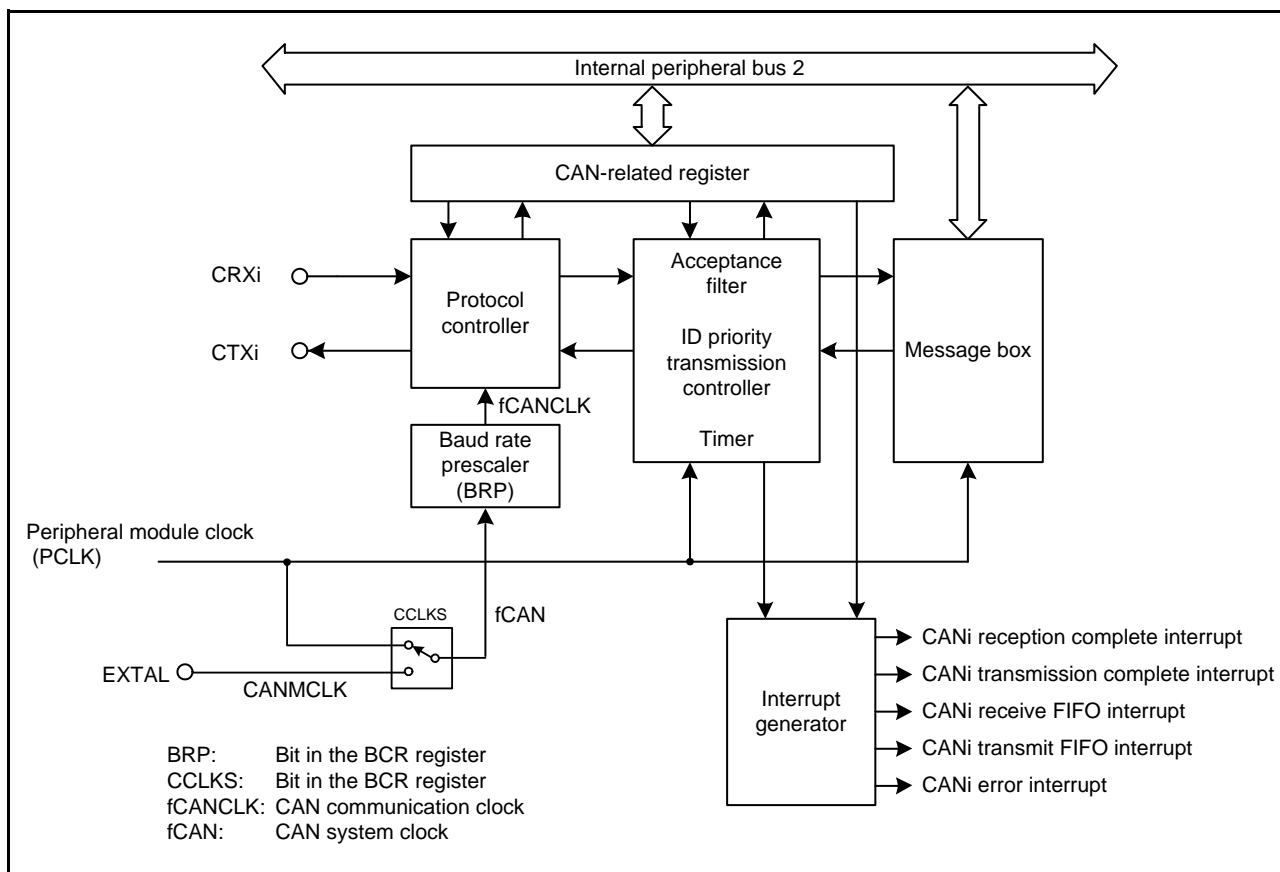
The RX630 Group implements three channels of the CAN (Controller Area Network) module that complies with the ISO11898-1 Specifications. The CAN module transmits and receives both formats of messages, namely the standard identifier (11 bits) (identifier is hereafter referred to as ID) and extended ID (29 bits).

Table 34.1 lists the specifications of the CAN module, and Figure 34.1 shows a block diagram of the CAN module (i = 0 to 2).

Connect the CAN bus transceiver externally.

**Table 34.1 Specifications of CAN Module**

Item	Description
Protocol	<ul style="list-style-type: none"> <li>ISO11898-1 compliant (standard and extended frames)</li> </ul>
Bit rate	<ul style="list-style-type: none"> <li>Programmable bit rate up to 1 Mbps (fCAN ≥ 8 MHz)</li> <li>fCAN: CAN clock source</li> </ul>
Message box	<ul style="list-style-type: none"> <li>32 mailboxes: Two selectable mailbox modes</li> <li>Normal mailbox mode: 32 mailboxes can be configured for either transmission or reception.</li> <li>FIFO mailbox mode: 24 mailboxes can be configured for either transmission or reception.</li> <li>Of the other mailboxes, four FIFO stages can be configured for transmission and four FIFO stages for reception.</li> </ul>
Reception	<ul style="list-style-type: none"> <li>Data frame and remote frame can be received.</li> <li>Selectable receiving ID format (only standard ID, only extended ID, or both IDs)</li> <li>Programmable one-shot reception function</li> <li>Selectable from overwrite mode (message overwritten) and overrun mode (message discarded)</li> <li>The reception complete interrupt can be individually enabled or disabled for each mailbox.</li> </ul>
Acceptance filter	<ul style="list-style-type: none"> <li>Eight acceptance masks (one mask for every four mailboxes)</li> <li>The mask can be individually enabled or disabled for each mailbox.</li> </ul>
Transmission	<ul style="list-style-type: none"> <li>Data frame and remote frame can be transmitted.</li> <li>Selectable transmitting ID format (only standard ID, only extended ID, or both IDs)</li> <li>Programmable one-shot transmission function</li> <li>Selectable from ID priority mode and mailbox number priority mode</li> <li>Transmission request can be aborted (the completion of abort can be confirmed with a flag)</li> <li>The transmission complete interrupt can be individually enabled or disabled for each mailbox.</li> </ul>
Mode transition for bus-off recovery	<ul style="list-style-type: none"> <li>Mode transition for the recovery from the bus-off state can be selected:</li> <li>ISO11898-1 Specifications compliant</li> <li>Automatic entry to CAN halt mode at bus-off entry</li> <li>Automatic entry to CAN halt mode at bus-off end</li> <li>Entry to CAN halt mode by a program</li> <li>Transition into error-active state by a program</li> </ul>
Error status monitoring	<ul style="list-style-type: none"> <li>CAN bus errors (stuff error, form error, ACK error, CRC error, bit error, and ACK delimiter error) can be monitored.</li> <li>Transition to error states can be detected (error-warning, error-passive, bus-off entry, and bus-off recovery).</li> <li>The error counters can be read.</li> </ul>
Time stamp function	<ul style="list-style-type: none"> <li>Time stamp function using a 16-bit counter</li> <li>The reference clock can be selected from 1-, 2-, 4- and 8-bit time periods.</li> </ul>
Interrupt function	<ul style="list-style-type: none"> <li>Five types of interrupt sources (reception complete, transmission complete, receive FIFO, transmit FIFO, and error interrupts)</li> </ul>
CAN sleep mode	<ul style="list-style-type: none"> <li>Current consumption can be reduced by stopping the CAN clock.</li> </ul>
Software support unit	<ul style="list-style-type: none"> <li>Three software support units:</li> <li>Acceptance filter support</li> <li>Mailbox search support (receive mailbox search, transmit mailbox search, and message lost search)</li> <li>Channel search support</li> </ul>
CAN clock source	Peripheral module clock (PCLK) or CANMCLK
Test mode	<ul style="list-style-type: none"> <li>Three test modes available for user evaluation</li> <li>Listen-only mode</li> <li>Self-test mode 0 (external loopback)</li> <li>Self-test mode 1 (internal loopback)</li> </ul>
Power consumption reducing function	Module-stop state can be set.



**Figure 34.1** Block Diagram of CAN Module (i = 0 to 2)

- CRXi and CTXi (i = 0 to 2)  
CAN input and output pins
- Protocol controller  
Handles CAN protocol processing such as bus arbitration, bit timing at transmission and reception, stuffing, and error handling, etc.
- Message box  
Consists of 32 mailboxes which can be configured as either transmit or receive mailboxes. Each mailbox has an individual ID, data length code, a data field (8 bytes), and a time stamp.
- Acceptance filter  
Performs filtering of received messages. MKR0 to MKR7 are used for the filtering process.
- Timer  
Used for the time stamp function. The timer value when a message is stored into the mailbox is written as the time stamp value.
- Interrupt generator:  
Generates the following five types of interrupts:  
CANi reception complete interrupt  
CANi transmission complete interrupt  
CANi receive FIFO interrupt  
CANi transmit FIFO interrupt  
CANi error interrupt



Table 34.2 lists the CAN module pins.

The CAN functions should be selected for the pins multiplexed with other signals. For details, see section 20, I/O Ports.

**Table 34.2 Pin Configuration**

Pin Name	I/O	Function
CRX0	Input	Pin for receiving data
CTX0	Output	Pin for transmitting data
CRX1	Input	Pin for receiving data
CTX1	Output	Pin for transmitting data
CRX2	Input	Pin for receiving data
CTX2	Output	Pin for transmitting data

## 34.2 Register Descriptions

### 34.2.1 Control Register (CTRL)

Address(es): CAN0.CTRL 0009 0840h, CAN1.CTRL 0009 1840h, CAN2.CTRL 0009 2840h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	RBOC	BOM[1:0]	SLPM	CANM[1:0]	TSPS[1:0]	TSRC	TPM	MLM	IDFM[1:0]	MBM				
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	MBM	CAN Mailbox Mode Select*1	0: Normal mailbox mode 1: FIFO mailbox mode	R/W
b2, b1	IDFM[1:0]	ID Format Mode Select*1	b2 b1 0 0: Standard ID mode All mailboxes (including FIFO mailboxes) handle only standard IDs. 0 1: Extended ID mode All mailboxes (including FIFO mailboxes) handle only extended IDs. 1 0: Mixed ID mode All mailboxes (including FIFO mailboxes) handle both standard IDs and extended IDs. Standard IDs or extended IDs are specified by using the IDE bit in the corresponding mailbox in normal mailbox mode. In FIFO mailbox mode, the IDE bit in the corresponding mailbox is used for mailboxes [0] to [23], the IDE bits in FIDCR0 and FIDCR1 are used for the receive FIFO, and the IDE bit in mailbox [24] is used for the transmit FIFO. 1 1: Do not use this combination	R/W
b3	MLM	Message Lost Mode Select*2	0: Overwrite mode 1: Overrun mode	R/W
b4	TPM	Transmission Priority Mode Select*2	0: ID priority transmit mode 1: Mailbox number priority transmit mode	R/W
b5	TSRC	Time Stamp Counter Reset Command*4	0: Nothing occurred 1: Reset*3	R/W
b7, b6	TSPS[1:0]	Time Stamp Prescaler Select*1	b7 b6 0 0: Every bit time 0 1: Every 2-bit time 1 0: Every 4-bit time 1 1: Every 8-bit time	R/W

Bit	Symbol	Bit Name	Description	R/W
b9, b8	CANM[1:0]	CAN Operating Mode Select*5	b9 b8 0 0: CAN operation mode 0 1: CAN reset mode 1 0: CAN halt mode 1 1: CAN reset mode (forcible transition)	R/W
b10	SLPM	CAN Sleep Mode*5,*6	0: Other than CAN sleep mode 1: CAN sleep mode	R/W
b12, b11	BOM[1:0]	Bus-Off Recovery Mode*1	b12 b11 0 0: Normal mode (ISO11898-1 compliant) 0 1: Entry to CAN halt mode automatically at bus-off entry 1 0: Entry to CAN halt mode automatically at bus-off end 1 1: Entry to CAN halt mode (during bus-off recovery period) by a program request	R/W
b13	RBOC	Forcible Return From Bus-Off*2	0: Nothing occurred 1: Forcible return from bus-off*3	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Note 1. Write to the BOM[1:0], TSPS[1:0], TPM, MLM, IDFM[1:0], and MBM bits in CAN reset mode.

Note 2. Set the RBOC bit to 1 in the bus-off state.

Note 3. This bit is automatically set back to 0 after being set to 1. It should be read as 0.

Note 4. Set the TSRC bit to 1 in CAN operation mode.

Note 5. When the CANM[1:0] and SLPM bits are changed, check STR to ensure that the mode has been switched. Do not change the CANM[1:0] bits or SLPM bit until the mode has been switched.

Note 6. Write to the SLPM bit in CAN reset mode or CAN halt mode. When changing the SLPM bit, write 0 or 1 to only the SLPM bit.

### MBM Bit (CAN Mailbox Mode Select)

When the MBM bit is 0 (normal mailbox mode), mailboxes [0] to [31] are configured as transmit or receive mailboxes.

When the MBM bit is 1 (FIFO mailbox mode), mailboxes [0] to [23] are configured as transmit or receive mailboxes.

Mailboxes [24] to [27] are configured as a transmit FIFO and mailboxes [28] to [31] as a receive FIFO.

Transmit data is written into mailbox [24] (mailbox [24] is a window mailbox for the transmit FIFO). Receive data is read from mailbox [28] (mailbox [28] is a window mailbox for the receive FIFO).

Table 34.3 lists the mailbox configuration.

### IDFM[1:0] Bits (ID Format Mode Select)

The IDFM[1:0] bits specify the ID format.

### MLM Bit (Message Lost Mode Select)

The MLM bit specifies the operation when a new message is captured in the unread mailbox. Overwrite mode or overrun mode can be selected. All mailboxes (including the receive FIFO) are set to either overwrite mode or overrun mode.

When the MLM bit is 0, all mailboxes are set to overwrite mode and the new message is overwriting the old message.

When the MLM bit is 1, all mailboxes are set to overrun mode and the new message is discarded.

### TPM Bit (Transmission Priority Mode Select)

The TPM bit specifies the priority of modes when transmitting messages.

ID priority transmit mode or mailbox number transmit mode can be selected. All mailboxes are set for either ID priority transmission or mailbox number priority transmission.

When the TPM bit is 0, ID priority transmit mode is selected and transmission priority complies with the CAN bus arbitration rule, as defined in the ISO11898-1 Specifications. In ID priority transmit mode, mailboxes [0] to [31] (in normal mailbox mode), and mailboxes [0] to [23] (in FIFO mailbox mode), and the transmit FIFO are compared for the IDs of mailboxes configured for transmission. If two or more mailbox IDs are the same, the mailbox with the smaller number has higher priority.

Only the next message to be transmitted from the transmit FIFO is included in the transmission arbitration. If a transmit FIFO message is being transmitted, the next pending message within the transmit FIFO is included in the transmission arbitration.

When the TPM bit is 1, mailbox number transmit mode is selected and the transmit mailbox with the smallest mailbox number has the highest priority. In FIFO mailbox mode, the transmit FIFO has lower priority than normal mailboxes (mailboxes [0] to [23]).

#### **TSRC Bit (Time Stamp Counter Reset Command)**

The TSRC bit is used to reset the time stamp counter. When the TSRC bit is set to 1, TSR is set to 0000h. This bit is automatically set to 0.

#### **TSPS[1:0] Bits (Time Stamp Prescaler Select)**

The TSPS[1:0] bits select the prescaler for the time stamp. The reference clock for the time stamp can be selected to be either 1-, 2-, 4- or 8-bit time periods.

#### **CANM[1:0] Bits (CAN Operating Mode Select)**

The CANM[1:0] bits select one of the following modes for the CAN module: CAN operation mode, CAN reset mode, or CAN halt mode. CAN sleep mode is set by the SLPM bit. For details, refer to section 34.3, Operating Mode.

When the CAN module enters CAN halt mode according to the setting of the BOM[1:0] bits, the CANM[1:0] bits are automatically set to 10b.

#### **SLPM Bit (CAN Sleep Mode)**

When the SLPM bit is set to 1, the CAN module enters CAN sleep mode. When the SLPM bit is set to 0, the CAN module exits CAN sleep mode. For details, refer to section 34.3, Operating Mode.

#### **BOM[1:0] Bits (Bus-Off Recovery Mode)**

The BOM[1:0] bits are used to select bus-off recovery mode for the CAN module.

When the BOM[1:0] bits are 00b, the recovery from bus-off is compliant with the ISO11898-1 Specifications, i.e. the CAN module reenters CAN communication (error-active state) after detecting 11 consecutive recessive bits 128 times. A bus-off recovery interrupt request is generated when recovering from bus-off.

When the BOM[1:0] bits are 01b and the CAN module reaches the bus-off state, the CANM[1:0] bits in CTLR are set to 10b (CAN halt mode) to enter CAN halt mode. No bus-off recovery interrupt request is generated when recovering from bus-off, and TECR and RECR are set to 00h.

When the BOM[1:0] bits are 10b, the CANM[1:0] bits are set to 10b as soon as the CAN module reaches the bus-off state. The CAN module enters CAN halt mode after the recovery from the bus-off state, i.e. after detecting 11 consecutive recessive bits 128 times. A bus-off recovery interrupt request is generated when recovering from bus-off, and TECR and RECR are set to 00h.

When the BOM[1:0] bits are 11b, the CAN module enters CAN halt mode by setting the CANM[1:0] bits to 10b while the CAN module is still in the bus-off state. No bus-off recovery interrupt request is generated when recovering from bus-off and TECR and RECR are set to 00h. However, if the CAN module recovers from bus-off after detecting 11 consecutive recessive bits 128 times before the CANM[1:0] bits are set to 10b, a bus-off recovery interrupt request is generated.

If the CPU requests an entry to CAN reset mode at the same time as the CAN module attempts to enter CAN halt mode (at bus-off entry when the BOM[1:0] bits are 01b, or at bus-off end when the BOM[1:0] bits are 10b), then the CPU request to enter CAN reset mode has higher priority.

**RBOC Bit (Forcible Return From Bus-Off)**

When the RBOC bit is set to 1 (force return from bus-off) in the bus-off state, the CAN module forcibly returns from the bus-off state. This bit is automatically set to 0. The error state changes from bus-off to error-active. When the RBOC bit is set to 1, RECR and TECR are set to 00h and the BOST bit in STR is set to 0 (the CAN module is not in bus-off state). The other registers remain unchanged even when the RBOC bit is set to 1. No bus-off recovery interrupt request is generated by this recovery from the bus-off state. Use the RBOC bit only when the BOM[1:0] bits are 00b (normal mode).

**Table 34.3 Mailbox Configuration**

Mailbox	MBM Bit = 0 (Normal Mailbox Mode)	MBM Bit = 1 (FIFO Mailbox Mode)
Mailboxes [0] to [23]	Normal mailbox	Normal mailbox
Mailboxes [24] to [27]		Transmit FIFO
Mailboxes [28] to [31]		Receive FIFO

Points 1 to 5 below should be considered when the CTRL.MBM bit is set to 1.

Note 1. Transmit FIFO is controlled by TFCR. MCTLj of mailboxes [24] to [27] is disabled. MCTL24 to MCTL27 cannot be used by the transmit FIFO.

Note 2. Receive FIFO is controlled by RFCR. MCTLj of mailboxes [28] to [31] is disabled. MCTL28 to MCTL31 cannot be used by the receive FIFO.

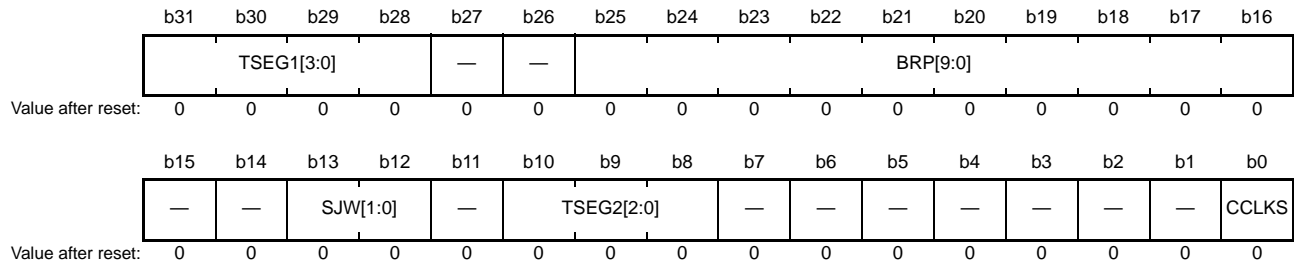
Note 3. Refer to MIER about the FIFO interrupts.

Note 4. The corresponding bits in MKIVLR for mailboxes [24] to [31] are disabled. Set 0 to these bits.

Note 5. Transmit/receive FIFOs can be used for both data frames and remote frames.

### 34.2.2 Bit Configuration Register (BCR)

Address(es): CAN0.BCR 0009 0844h, CAN1.BCR 0009 1844h, CAN2.BCR 0009 2844h



Bit	Symbol	Bit Name	Description	R/W
b0	CCLKS	CAN Clock Source Selection	0: PCLK (generated by the PLL clock) 1: CANMCLK (generated by the main clock)	R/W
b7 to b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b10 to b8	TSEG2[2:0]	Time Segment 2 Control	$b_{10} \ b_8$ 0 0 0: Setting prohibited 0 0 1: 2 Tq 0 1 0: 3 Tq 0 1 1: 4 Tq 1 0 0: 5 Tq 1 0 1: 6 Tq 1 1 0: 7 Tq 1 1 1: 8 Tq	R/W
b11	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b13, b12	SJW[1:0]	Resynchronization Jump Width Control	$b_{13} \ b_{12}$ 0 0: 1 Tq 0 1: 2 Tq 1 0: 3 Tq 1 1: 4 Tq	R/W
b15, b14	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b25 to b16	BRP[9:0]	Prescaler Division Ratio Select *1	These bits set the frequency of the CAN communication clock (fCANCLK).	R/W
b26	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b27	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b31 to 28	TSEG1[3:0]	Time Segment 1 Control	$b_{31} \ b_{28}$ 0 0 0 0: Setting prohibited 0 0 0 1: Setting prohibited 0 0 1 0: Setting prohibited 0 0 1 1: 4 Tq 0 1 0 0: 5 Tq 0 1 0 1: 6 Tq 0 1 1 0: 7 Tq 0 1 1 1: 8 Tq 1 0 0 0: 9 Tq 1 0 0 1: 10 Tq 1 0 1 0: 11 Tq 1 0 1 1: 12 Tq 1 1 0 0: 13 Tq 1 1 0 1: 14 Tq 1 1 1 0: 15 Tq 1 1 1 1: 16 Tq	R/W

Tq: Time Quantum

Note 1. Do not select the value less than 1 while the SCKCR3.CKSEL[2:0] bits are 010b (selecting the main clock oscillator).

For bit timing setting, refer to section 34.4, CAN Communication Speed Setting.

Set BCR before entering CAN halt mode or CAN operation mode from CAN reset mode. After the setting is made once, this register can be written to in CAN reset mode or CAN halt mode.

BCR consists of 24 bits. A 32-bit read/write access should be performed carefully not to rewrite bits b0 to b7.

#### **CCLKS Bit (CAN Clock Source Selection)**

When the CCLKS bit is 0, the peripheral module clock (PCLK) produced by the PLL frequency synthesizer is used as the CAN clock source (fCAN).

When the CCLKS bit is 1, CANMCLK produced externally by the EXTAL pins is used as the CAN clock source (fCAN).

#### **TSEG2[2:0] Bits (Time Segment 2 Control)**

The TSEG2[2:0] bits are used to specify the length of the phase buffer segment 2 (PHASE\_SEG2) with a Tq value. A value from 2 to 8 Tq can be set. Set a value smaller than that of the TSEG1[3:0] bits.

#### **SJW[1:0] Bits (Resynchronization Jump Width Control)**

The SJW[1:0] bits are used to specify the resynchronization jump width with a Tq value. A value from 1 to 4 Tq can be set. Set a value smaller than or equal to that of the TSEG2[2:0] bits.

#### **BRP[9:0] Bits (Prescaler Division Ratio Select)**

The BRP[9:0] bits are used to set the frequency of the CAN communication clock (fCANCLK). The fCANCLK cycle is 1 Tq. If the setting is P (0 to 1023), the baud rate prescaler divides fCAN by P + 1.

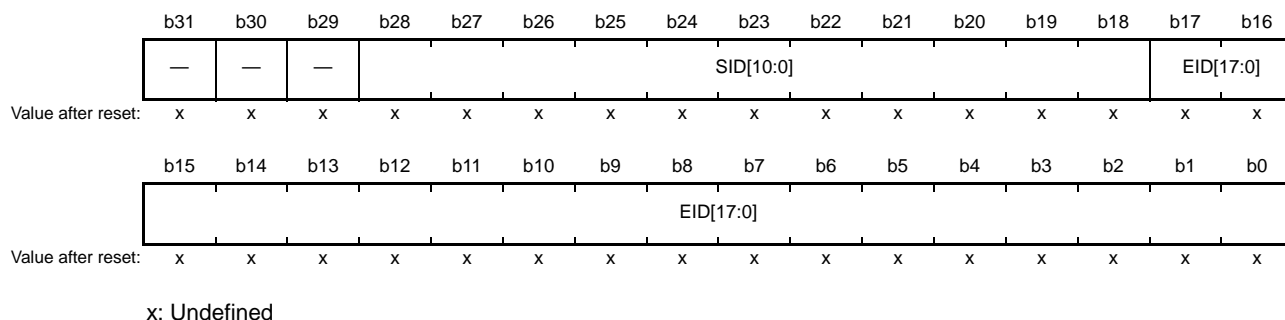
#### **TSEG1[3:0] Bits (Time Segment 1 Control)**

The TSEG1[3:0] bits are used to specify the total length of the propagation time segment (PROP\_SEG) and phase buffer segment 1 (PHASE\_SEG1) with a time quantum (Tq) value.

A value from 4 to 16 Tq can be set.

### 34.2.3 Mask Register k (MKRk) (k = 0 to 7)

Address(es): CAN0.MKR0 0009 0400h, CAN0.MKR1 0009 0404h, CAN0.MKR2 0009 0408h, CAN0.MKR3 0009 040Ch, CAN0.MKR4 0009 0410h, CAN0.MKR5 0009 0414h, CAN0.MKR6 0009 0418h, CAN0.MKR7 0009 041Ch, CAN1.MKR0 0009 1400h, CAN1.MKR1 0009 1404h, CAN1.MKR2 0009 1408h, CAN1.MKR3 0009 140Ch, CAN1.MKR4 0009 1410h, CAN1.MKR5 0009 1414h, CAN1.MKR6 0009 1418h, CAN1.MKR7 0009 141Ch, CAN2.MKR0 0009 2400h, CAN2.MKR1 0009 2404h, CAN2.MKR2 0009 2408h, CAN2.MKR3 0009 240Ch, CAN2.MKR4 0009 2410h, CAN2.MKR5 0009 2414h, CAN2.MKR6 0009 2418h, CAN2.MKR7 0009 241Ch



Bit	Symbol	Bit Name	Description	R/W
b17 to b0	EID[17:0]	Extended ID	0: Corresponding EID[17:0] bit is not compared 1: Corresponding EID[17:0] bit is compared	R/W
b28 to b18	SID[10:0]	Standard ID	0: Corresponding SID[10:0] bit is not compared 1: Corresponding SID[10:0] bit is compared	R/W
b31 to b29	—	Reserved	The read value is undefined. The write value should be 0.	R/W

For the mask function in FIFO mailbox mode, refer to section 34.6, Acceptance Filtering and Masking Functions. Write to MKR0 to MKR7 in CAN reset mode or CAN halt mode.

#### EID[17:0] Bits (Extended ID)

The EID[17:0] bits are the filter mask bits for the CAN extended ID bits.

These bits are used to receive extended ID messages.

When the EID[17:0] bit is set to 0, the received ID is not compared with the mailbox ID for the corresponding EID[17:0] bit.

When the EID[17:0] bit is set to 1, the received ID is compared with the mailbox ID for the corresponding EID[17:0] bit.

#### SID[10:0] Bits (Standard ID)

The SID[10:0] bits are the filter mask bits corresponding to the CAN standard ID bits.

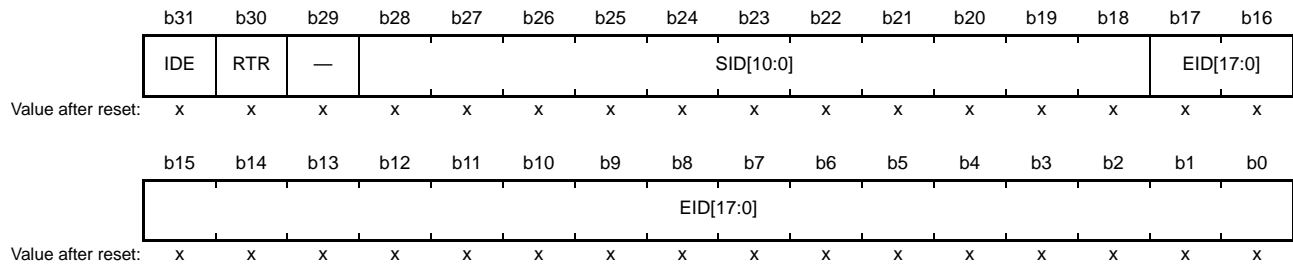
These bits are used to receive both standard ID and extended ID messages.

When the SID[10:0] bit is set to 0, the received ID is not compared with the mailbox ID for the corresponding SID[10:0] bit.

When the SID[10:0] bit is set to 1, the received ID is compared with the mailbox ID for the corresponding SID[10:0] bit.

### 34.2.4 FIFO Received ID Compare Registers 0 and 1 (FIDCR0 and FIDCR1)

Address(es): CAN0.FIDCR0 0009 0420h, CAN0.FIDCR1 0009 0424h, CAN1.FIDCR0 0009 1420h, CAN1.FIDCR1 0009 1424h, CAN2.FIDCR0 0009 2420h, CAN2.FIDCR0 0009 2424h



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b17 to b0	EID[17:0]	Extended ID	0: Corresponding EID[17:0] bits are 0 1: Corresponding EID[17:0] bits are 1	R/W
b28 to b18	SID[10:0]	Standard ID	0: Corresponding SID[10:0] bits are 0 1: Corresponding SID[10:0] bits are 1	R/W
b29	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b30	RTR	Remote Transmission Request	0: Data frame 1: Remote frame	R/W
b31	IDE	ID Extension*1	0: Standard ID 1: Extended ID	R/W

Note 1. When the IDFM[1:0] bits are not 10b, the IDE bit should be written with 0 and read as 0.

FIDCR0 and FIDCR1 are enabled when the MBM bit in CTRLR is set to 1 (FIFO mailbox mode). Bits EID[17:0], SID[10:0], RTR, and IDE in MB28 to MB31 are disabled.

For the usage of FIDCR0 and FIDCR1, refer to section 34.6, Acceptance Filtering and Masking Functions. Write to FIDCR0 and FIDCR1 in CAN reset mode or CAN halt mode.

#### EID[17:0] Bits (Extended ID)

The EID[17:0] bits set the extended ID of data frames and remote frames. These bits are used to receive extended ID messages.

#### SID[10:0] Bits (Standard ID)

The SID[10:0] bits set the standard ID of data frames and remote frames. These bits are used to receive both standard ID and extended ID messages.

#### RTR Bit (Remote Transmission Request)

The RTR bit sets the specified frame format of data frames or remote frames.

- When both RTR bits in FIDCR0 and FIDCR1 are set to 0, only data frames can be received.
- When both RTR bits in FIDCR0 and FIDCR1 are set to 1, only remote frames can be received.
- When the RTR bits in FIDCR0 and FIDCR1 are set to 0 or 1 individually, both data frames and remote frames can be received.



### IDE Bit (ID Extension)

The IDE bit sets the ID format to standard ID or extended ID. The IDE bit is enabled when the IDFM[1:0] bits in CTRL are 10b (mixed ID mode).

- When both IDE bits in FIDCR0 and FIDCR1 are set to 0, only standard ID frames can be received.
- When both IDE bits in FIDCR0 and FIDCR1 are set to 1, only extended ID frames can be received.
- When the IDE bits in FIDCR0 and FIDCR1 are set to 0 or 1 individually, both standard ID and extended ID frames can be received.

### 34.2.5 Mask Invalid Register (MKIVLR)

Address(es): CAN0.MKIVLR 0009 0428h, CAN1.MKIVLR 0009 1428h, CAN2.MKIVLR 0009 2428h

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MB31	MB30	MB29	MB28	MB27	MB26	MB25	MB24	MB23	MB22	MB21	MB20	MB19	MB18	MB17	MB16
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
	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	MB31 to MB0	Mask Invalid	0: Mask valid 1: Mask invalid	R/W

Each bit in MKIVLR corresponds to a mailbox.

The correspondence between the bits and mailboxes is shown below.

Bit 0 in MKIVLR corresponds to mailbox 0 (MB0) and bit 31 corresponds to mailbox 31 (MB31).<sup>\*1</sup>

When a bit is set to 1, the relevant acceptance mask register becomes invalid for the corresponding mailbox. When a mask invalid bit is set to 1, a message is received by the corresponding mailbox only if the receive message ID matches the mailbox ID exactly.

Write to MKIVLR in CAN reset mode or CAN halt mode.

Note 1. Set bits 31 to 24 to 0 in FIFO mailbox mode.

### 34.2.6 Mailbox Register j (MBj) (j = 0 to 31)

Table 34.4 lists the CAN<sub>i</sub> mailbox memory mapping, and Table 34.5 lists the CAN data frame configuration. The value after reset of the CAN<sub>i</sub> mailbox is undefined.

Write to MB<sub>j</sub> only when the related MCTL<sub>j</sub> (j = 0 to 31) is 00h and the corresponding mailbox is not processing an abort request.

See Table 34.4 for detailed register addresses.

**Table 34.4 CAN<sub>i</sub> Mailbox Memory Mapping**

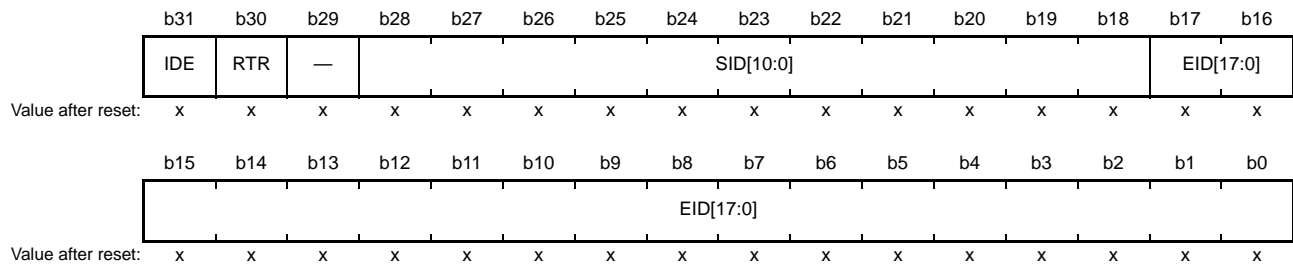
Address			Message Content
CAN0	CAN1	CAN2	Memory Mapping
0009 0200h + 16xj + 0	0009 1200h + 16xj + 0	0009 2200h + 16xj + 0	IDE, RTR, SID10 to SID6
0009 0200h + 16xj + 1	0009 1200h + 16xj + 1	0009 2200h + 16xj + 1	SID5 to SID0, EID17, EID16
0009 0200h + 16xj + 2	0009 1200h + 16xj + 2	0009 2200h + 16xj + 2	EID15 to EID8
0009 0200h + 16xj + 3	0009 1200h + 16xj + 3	0009 2200h + 16xj + 3	EID7 to EID0
0009 0200h + 16xj + 4	0009 1200h + 16xj + 4	0009 2200h + 16xj + 4	—
0009 0200h + 16xj + 5	0009 1200h + 16xj + 5	0009 2200h + 16xj + 5	Data length code (DLC[3:0])
0009 0200h + 16xj + 6	0009 1200h + 16xj + 6	0009 2200h + 16xj + 6	Data byte 0
0009 0200h + 16xj + 7	0009 1200h + 16xj + 7	0009 2200h + 16xj + 7	Data byte 1
0009 0200h + 16xj + 8	0009 1200h + 16xj + 8	0009 2200h + 16xj + 8	Data byte 2
0009 0200h + 16xj + 9	0009 1200h + 16xj + 9	0009 2200h + 16xj + 9	Data byte 3
0009 0200h + 16xj + 10	0009 1200h + 16xj + 10	0009 2200h + 16xj + 10	Data byte 4
0009 0200h + 16xj + 11	0009 1200h + 16xj + 11	0009 2200h + 16xj + 11	Data byte 5
0009 0200h + 16xj + 12	0009 1200h + 16xj + 12	0009 2200h + 16xj + 12	Data byte 6
0009 0200h + 16xj + 13	0009 1200h + 16xj + 13	0009 2200h + 16xj + 13	Data byte 7
0009 0200h + 16xj + 14	0009 1200h + 16xj + 14	0009 2200h + 16xj + 14	Time stamp upper byte
0009 0200h + 16xj + 15	0009 1200h + 16xj + 15	0009 2200h + 16xj + 15	Time stamp lower byte

**Table 34.5 CAN Data Frame Configuration**

SID10 to SID6	SID5 to SID0	EID17 to EID16	EID15 to EID8	EID7 to EID0	DLC3 to DLC1	DATA0	DATA1	...	DATA7
---------------	--------------	----------------	---------------	--------------	--------------	-------	-------	-----	-------

The previous value of each mailbox is retained unless a new message is received.

Address(es): CAN0.MB0 to CAN0.MB63 0009 0200h to 0009 03FFh, CAN1.MB0 to CAN1.MB63 0009 1200h to 0009 13FFh, CAN2.MB0 to CAN2.MB63 0009 2200h to 0009 23FFh

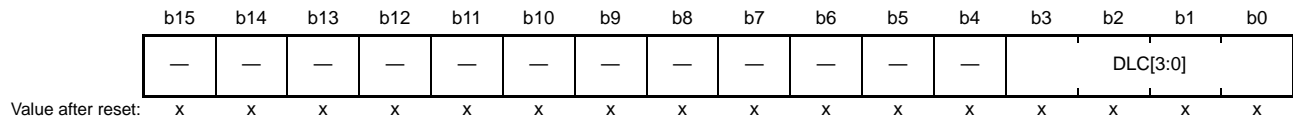


x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b17 to b0	EID[17:0]	Extended ID*1	0: Corresponding EID[17:0] bits are 0 1: Corresponding EID[17:0] bits are 1	R/W
b28 to b18	SID[10:0]	Standard ID	0: Corresponding SID[10:0] bits are 0 1: Corresponding SID[10:0] bits are 1	R/W
b29	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b30	RTR	Remote Frame Request	0: Data frame 1: Remote frame	R/W
b31	IDE	ID Extension*2	0: Standard ID 1: Extended ID	R/W

Note 1. If the mailbox has received a standard ID message, the EID bits in the mailbox are undefined.

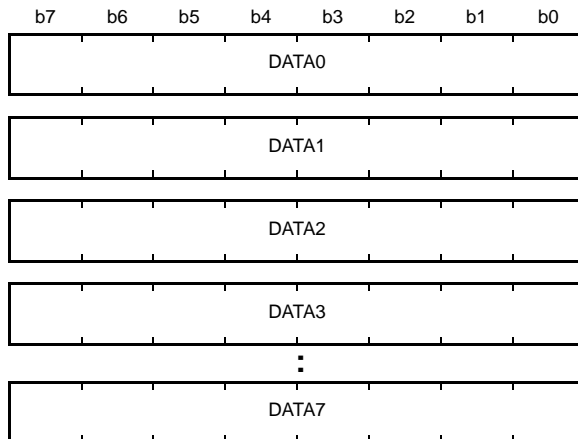
Note 2. The IDE bit is enabled when the IDFM[1:0] bits in CTLR are 10b (mixed ID mode). When the IDFM[1:0] bits are not 10b, it should be written with 0 and read as 0.



x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	DLC[3:0]	Data Length Code*1	b3 b0 0 0 0 0: Data length = 0 byte 0 0 0 1: Data length = 1 byte 0 0 1 0: Data length = 2 bytes 0 0 1 1: Data length = 3 bytes 0 1 0 0: Data length = 4 bytes 0 1 0 1: Data length = 5 bytes 0 1 1 0: Data length = 6 bytes 0 1 1 1: Data length = 7 bytes 1 x x x: Data length = 8 bytes x: Represents any value.	R/W
b15 to b4	—	Reserved	The read value is undefined. The write value should be 0.	R/W

Note 1. If the mailbox has received a message whose data length set by the DLC[3:0] bits is less than 8 bytes, the values of DATA larger than the data length set by the DLC[3:0] bits in the mailbox are undefined.



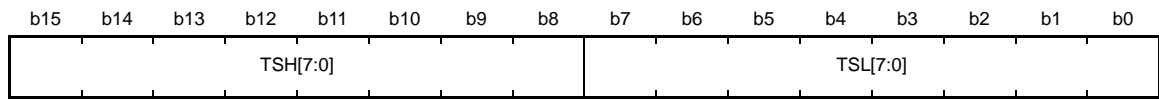
Value after reset: x x x x x x x x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	DATA0 to DATA7	Data Bytes 0 to 7*1,*2	DATA0 to DATA7 store the transmitted or received CAN message data. Transmission or reception starts from DATA0. The bit order on the CAN bus is MSB first, and transmission or reception starts from bit 7.	R/W

Note 1. If the mailbox has received a message with n bytes less than 8 bytes, the values of DATA0 to DATA7 in the mailbox are undefined.

Note 2. If the mailbox has received a remote frame, the previous values of DATA0 to DATA7 in the mailbox are retained.



Value after reset: x x x x x x x x x x x x x x x x

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TSL[7:0]	Time Stamp Lower Byte	Bits TSH[7:0] and TSL[7:0] store the counter value of the time stamp when received messages are stored in the mailbox.	R/W
b15 to b8	TSH[7:0]	Time Stamp Higher Byte		R/W

### EID[17:0] Bits (Extended ID)

The EID[17:0] bits set the extended ID of data frames and remote frames.

These bits are used to transmit or receive extended ID messages.

### SID[10:0] Bits (Standard ID)

The SID[10:0] bits set the standard ID of data frames and remote frames.

These bits are used to transmit or receive both standard ID and extended ID messages.

**RTR Bit (Remote Frame Request)**

The RTR bit sets the frame format of data frames or remote frames.

- Receive mailbox receives only frames with the format specified by the RTR bit.
- Transmit mailbox transmits according to the frame format specified by the RTR bit.
- Receive FIFO mailbox receives the data frame, remote frame, or both frames specified by the RTR bit in FIDCR0 and FIDCR1.
- Transmit FIFO mailbox transmits the data frame or remote frame specified by the RTR bit in the relevant transmit message.

**IDE Bit (ID Extension)**

The IDE bit sets the ID format of standard IDs or extended IDs. The IDE bit is enabled when the IDFM[1:0] bits in CTLR is 10b (mixed ID mode).

- Receive mailbox receives only the ID format specified by the IDE bit.
- Transmit mailbox transmits with the ID format specified by the IDE bit.
- Receive FIFO mailbox receives messages with the standard ID and extended ID specified by the IDE bit in FIDCR0 and FIDCR1.
- Transmit FIFO mailbox transmits messages with the standard ID or extended ID specified by the IDE bit in the relevant transmit message.

**DLC[3:0] Bits (Data Length Code)**

The DLC[3:0] bits specify the number of bytes of data to be transmitted in data frames. When a remote frame is used to request data, this field specifies the requested number of bytes of data.

When a data frame is received, the number of bytes received is stored in this field. When a remote frame is received, this field is used to store the number requested by the frame.

### 34.2.7 Mailbox Interrupt Enable Register (MIER)

Address(es): CAN0.MIER 0009 042Ch, CAN1.MIER 0009 142Ch, CAN2.MIER 0009 242Ch

	b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
	MB31	MB30	MB29	MB28	MB27	MB26	MB25	MB24	MB23	MB22	MB21	MB20	MB19	MB18	MB17	MB16
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
	MB15	MB14	MB13	MB12	MB11	MB10	MB9	MB8	MB7	MB6	MB5	MB4	MB3	MB2	MB1	MB0
Value after reset:	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

x: Undefined

- Normal mailbox mode

Bit	Symbol	Bit Name	Description	R/W
b31 to b0	MB31 to MB0	Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled Bit 31 corresponds to mailbox 31 (MB31), and bit 0 corresponds to mailbox 0 (MB0).	R/W

- FIFO mailbox mode

Bit	Symbol	Bit Name	Description	R/W
b23 to b0	MB23 to MB0	Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled Bit 23 corresponds to mailbox 23 (MB23), and bit 0 corresponds to mailbox 0 (MB0).	R/W
b24	MB24	Transmit FIFO Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/W
b25	MB25	Transmit FIFO Interrupt Generation Timing Control	0: Every time transmission is completed 1: When the transmit FIFO becomes empty due to completion of transmission	R/W
b27, b26	—	Reserved	The read value is undefined. The write value should be 0.	R/W
b28	MB28	Receive FIFO Interrupt Enable	0: Interrupt disabled 1: Interrupt enabled	R/W
b29	MB29	Receive FIFO Interrupt Generation Timing Control*1	0: Every time reception is completed 1: When the receive FIFO becomes buffer warning by completion of reception	R/W
b31, b30	—	Reserved	The read value is undefined. The write value should be 0.	R/W

Note 1. No interrupt request is generated when the receive FIFO becomes buffer warning from full. "Buffer warning" indicates a state in which the third message is stored in the receive FIFO.

MIER can individually enable interrupts for each mailbox.

In normal mailbox mode (all bits) and in FIFO mailbox mode (bits 24 to 0 in MIER), each bit corresponds to the mailbox with the related number. These bits enable or disable transmission/reception complete interrupts for the corresponding mailboxes.

- Bit 0 in MIER corresponds to mailbox 0 (MB0).
- Bit 31 in MIER corresponds to mailbox 31 (MB31).

In FIFO mailbox mode, bits 29, 28, 25, and 24 of MIER specify whether transmit/receive FIFO interrupts are enabled/disabled and the timing when interrupt requests are generated.

Write to MIER only when the related MCTLj (j = 0 to 31) is 00h and the corresponding mailbox is not processing a transmission or reception abort request. In FIFO mailbox mode, change the bits in MIER for the related FIFO only when the TFE bit in TFCR is 0 and the TFEST bit is 1, and the RFE bit in RFCR is 0 and the RFEST bit in RFCR is 1.

### 34.2.8 Message Control Register j (MCTLj) (j = 0 to 31)

Address(es): CAN0.MCTL0 to CAN0.MCTL31 0009 0820h to 0009 083Fh, CAN1.MCTL0 to CAN1.MCTL310009 1820h to 0009 183Fh, CAN2.MCTL0 to CAN2.MCTL31 0009 2820h to 0009 283Fh

- Transmit mode (when the TRMREQ bit is 1 and the RECREQ bit is 0)

b7	b6	b5	b4	b3	b2	b1	b0
TRMREQ	RECREQ	—	ONESHOT	—	TRMABT	TRMACTIVE	SENTDATA
Value after reset:	0	0	0	0	0	0	0

- Receive mode (when the TRMREQ bit is 0 and the RECREQ bit is 1)

b7	b6	b5	b4	b3	b2	b1	b0
TRMREQ	RECREQ	—	ONESHOT	—	MSGLOST	INVALIDATA	NEWDATA
Value after reset:	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SENTDATA	Transmission Complete Flag <sup>*1,*2</sup>	0: Transmission is not completed 1: Transmission is completed	R/W
	NEWDATA	Reception Complete Flag <sup>*1,*2</sup>	0: No data has been received or 0 is written to the NEWDATA bit 1: A new message is being stored or has been stored to the mailbox	R/W
b1	TRMACTIVE	Transmission-in-Progress Status Flag	(Transmit mailbox setting enabled) 0: Transmission is pending or transmission is not requested 1: From acceptance of transmission request to completion of transmission, or error/arbitration-lost	R
	INVALIDATA	Reception-in-Progress Status Flag	(Receive mailbox setting enabled) 0: Message valid 1: Message being updated	R
b2	TRMABT	Transmission Abort Complete Flag <sup>*1,*2</sup>	(Transmit mailbox setting enabled) 0: Transmission has started, transmission abort failed because transmission is completed, or transmission abort is not requested 1: Transmission abort is completed	R/W
	MSGLOST	Message Lost Flag <sup>*1,*2</sup>	(Receive mailbox setting enabled) 0: Message is not overwritten or overrun 1: Message is overwritten or overrun	R/W
b3	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b4	ONESHOT	One-Shot Enable <sup>*3</sup>	0: One-shot reception or one-shot transmission disabled 1: One-shot reception or one-shot transmission enabled	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	RECREQ	Receive Mailbox Request <sup>*2,*3,*4,*5</sup>	0: Not configured for reception 1: Configured for reception	R/W
b7	TRMREQ	Transmit Mailbox Request <sup>*2,*4</sup>	0: Not configured for transmission 1: Configured for transmission	R/W

Note 1. Write 0 only. Writing 1 has no effect.

Note 2. When writing 0 to bits NEWDATA, SENTDATA, MSGLOST, TRMABT, RECREQ, and TRMREQ by a program, do not use the logic operation instruction (AND). Write 0 to only the specified bit and write 1 to the other bits before using the transfer (MOV) instruction.

Note 3. To enter one-shot receive mode, write 1 to the ONESHOT bit at the same time as setting the RECREQ bit to 1. To exit one-shot receive mode, write 0 to the ONESHOT bit after writing 0 to the RECREQ bit and confirming that it has been set to 0.

To enter one-shot transmit mode, write 1 to the ONESHOT bit at the same time as setting the TRMREQ bit to 1.

To exit one-shot transmit mode, write 0 to the ONESHOT bit after the message has been transmitted or aborted.

Note 4. Do not set both the RECREQ and TRMREQ bits to 1.

Note 5. When setting the RECREQ bit to 0, set bits MSGLOST, NEWDATA, and RECREQ to 0 simultaneously.

Write to the MCTLj in CAN operation mode or CAN halt mode.  
Do not use MCTL24 to MCTL31 in FIFO mailbox mode.

#### **SENTDATA Flag (Transmission Complete Flag)**

The SENTDATA flag is set to 1 when data transmission from the corresponding mailbox is completed. The SENTDATA flag is set to 0 by writing 0 by a program.

To set the SENTDATA flag to 0, first set the TRMREQ bit to 0. Bits SENTDATA and TRMREQ cannot be set to 0 simultaneously. To transmit a new message from the corresponding mailbox, set the SENTDATA flag to 0.

#### **NEWDATA Flag (Reception Complete Flag)**

The NEWDATA flag is set to 1 when a new message is being stored or has been stored to the mailbox. The timing for setting this bit to 1 is simultaneous with the INVALIDDATA bit. The NEWDATA flag is set to 0 by writing 0 by a program. The NEWDATA flag cannot be set to 0 by writing 0 by a program while the related INVALIDDATA bit is 1.

#### **TRMACTIVE Bit (Transmission-in-Progress Status Flag)**

The TRMACTIVE bit is set to 1 when the corresponding mailbox of the CAN module begins transmitting a message. The TRMACTIVE is set to 0 when the CAN module has lost CAN bus arbitration, a CAN bus error occurs, or data transmission is completed.

#### **INVALIDDATA Bit (Reception-in-Progress Status Flag)**

After the completion of a message reception, the INVALIDDATA bit is set to 1 while the received message is being updated into the corresponding mailbox. The INVALIDDATA bit is set to 0 immediately after the message has been stored. If the mailbox is read while the INVALIDDATA bit is 1, the data is undefined.

#### **TRMABT Flag (Transmission Abort Complete Flag)**

The TRMABT flag is set to 1 in the following cases:

- Following a transmission abort request, when the transmission abort is completed before starting transmission.
- Following a transmission abort request, when the CAN module detects CAN bus arbitration-lost or a CAN bus error.
- In one-shot transmission mode (RECREQ bit = 0, TRMREQ bit = 1, and ONESHOT bit = 1), when the CAN module detects CAN bus arbitration-lost or a CAN bus error.

The TRMABT flag is not set to 1 when data transmission is completed. In this case, the SENTDATA flag is set to 1. The TRMABT flag is set to 0 by writing 0 by a program.

#### **MSGLOST Flag (Message Lost Flag)**

The MSGLOST flag is set to 1 when the mailbox is overwritten or overrun by a new received message while the NEWDATA flag is 1. The MSGLOST flag is set to 1 at the end of the 6th bit of EOF. The MSGLOST flag is set to 0 by writing 0 by a program.

In both overwrite and overrun modes, the MSGLOST flag cannot be set to 0 by writing 0 by a program during five peripheral module clock (PCLK) cycles following the sixth bit of EOF.



### ONESHOT Bit (One-Shot Enable)

The ONESHOT bit can be used in the following two ways, receive mode and transmit mode.

- One-shot receive mode  
When the ONESHOT bit is set to 1 in receive mode (RECREQ bit = 1 and TRMREQ bit = 0), the mailbox receives a message only one time. (The mailbox does not behave as a receive mailbox after having received a message one time.) The behavior of bits NEWDATA and INVALIDDATA is the same as in normal receive mode. In one-shot receive mode, the MSGLOST flag is not set to 1. To set the ONESHOT bit to 0, first write 0 to the RECREQ bit and ensure that it has been set to 0.
- One-shot transmit mode  
When the ONESHOT bit is set to 1 in transmit mode (RECREQ bit = 0 and TRMREQ bit = 1), the CAN module transmits a message only one time. (The CAN module does not transmit the message again if a CAN bus error or CAN bus arbitration-lost occurs.) When transmission is completed, the SENTDATA flag is set to 1. If transmission is not completed due to a CAN bus error or CAN bus arbitration-lost, the TRMABT flag is set to 1. Set the ONESHOT bit to 0 after the SENTDATA or TRMABT bit is set to 1.

### RECREQ Bit (Receive Mailbox Request)

The RECREQ bit selects receive modes listed in Table 34.10.

When the RECREQ bit is set to 1, the corresponding mailbox is configured for reception of a data frame or a remote frame.

When the RECREQ bit is set to 0, the corresponding mailbox is not configured for reception of a data frame or a remote frame.

Due to hardware protection, the RECREQ bit cannot be set to 0 by writing 0 by a program during the following period.

- Hardware protection is started  
From the acceptance filter processing (the beginning of CRC field)
  - Hardware protection is released
    - For the mailbox that is specified to receive the incoming message, after the received data is stored into the mailbox or a CAN bus error occurs (i.e. maximum period of hardware protection is from the beginning of CRC field to the end of the 7th bit of EOF)
    - For the other mailboxes, after the acceptance filter processing
    - If no mailbox is specified to receive the message, after the acceptance filter processing
- When setting the RECREQ bit to 1, do not set the TRMREQ bit to 1. To change the configuration of a mailbox from transmission to reception, first abort the transmission and then set bits SENTDATA and TRMABT to 0 before changing to reception.

### TRMREQ Bit (Transmit Mailbox Request)

The TRMREQ bit selects transmit modes listed in Table 34.10.

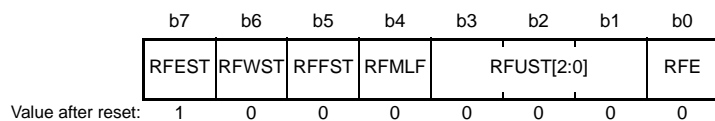
When the TRMREQ bit is set to 1, the corresponding mailbox is configured for transmission of a data frame or a remote frame.

When the TRMREQ bit is set to 0, the corresponding mailbox is not configured for transmission of a data frame or a remote frame.

If the TRMREQ bit is changed from 1 to 0 to cancel the corresponding transmission request, either the TRMABT or SENTDATA flag is set to 1. When setting the TRMREQ bit to 1, do not set the RECREQ bit to 1. To change the configuration of a mailbox from reception to transmission, first abort the reception and then set bits NEWDATA and MSGLOST to 0 before changing to transmission.

### 34.2.9 Receive FIFO Control Register (RFCR)

Address(es): CAN0.RFCR 0009 0848h, CAN1.RFCR 0009 1848h, CAN2.RFCR 0009 2848h



Bit	Symbol	Bit Name	Description	R/W																											
b0	RFE	Receive FIFO Enable	0: Receive FIFO disabled 1: Receive FIFO enabled	R/W																											
b3-b1	RFUST[2:0]	Receive FIFO Unread Message Number Status	<table border="0"> <tr> <td>b3</td> <td>b1</td> <td></td> </tr> <tr> <td>0</td> <td>0</td> <td>0: No unread message</td> </tr> <tr> <td>0</td> <td>0</td> <td>1: 1 unread message</td> </tr> <tr> <td>0</td> <td>1</td> <td>0: 2 unread messages</td> </tr> <tr> <td>0</td> <td>1</td> <td>1: 3 unread messages</td> </tr> <tr> <td>1</td> <td>0</td> <td>0: 4 unread messages</td> </tr> <tr> <td>1</td> <td>0</td> <td>1: Reserved</td> </tr> <tr> <td>1</td> <td>1</td> <td>0: Reserved</td> </tr> <tr> <td>1</td> <td>1</td> <td>1: Reserved</td> </tr> </table>	b3	b1		0	0	0: No unread message	0	0	1: 1 unread message	0	1	0: 2 unread messages	0	1	1: 3 unread messages	1	0	0: 4 unread messages	1	0	1: Reserved	1	1	0: Reserved	1	1	1: Reserved	R
b3	b1																														
0	0	0: No unread message																													
0	0	1: 1 unread message																													
0	1	0: 2 unread messages																													
0	1	1: 3 unread messages																													
1	0	0: 4 unread messages																													
1	0	1: Reserved																													
1	1	0: Reserved																													
1	1	1: Reserved																													
b4	RFMLF	Receive FIFO Message Lost Flag	0: No receive FIFO message lost has occurred 1: Receive FIFO message lost has occurred	R/W																											
b5	RFFST	Receive FIFO Full Status Flag	0: Receive FIFO is not full 1: Receive FIFO is full (4 unread messages)	R																											
b6	RFWST	Receive FIFO Buffer Warning Status Flag	0: Receive FIFO is not buffer warning 1: Receive FIFO is buffer warning (3 unread messages)	R																											
b7	RFEST	Receive FIFO Empty Status Flag	0: Unread message in receive FIFO 1: No unread message in receive FIFO	R																											

Write to RFCR in CAN operation mode or CAN halt mode.

#### RFE Bit (Receive FIFO Enable)

When the RFE bit is set to 1, the receive FIFO is enabled.

When the RFE bit is set to 0, the receive FIFO is disabled for reception and becomes empty (RFEST bit = 1). Write 0 to the RFE bit simultaneously with setting the RFMLF bit.

Do not set this bit to 1 in normal mailbox mode (MBM bit in CTLR = 0). Due to hardware protection, the RFE bit is not set to 0 by writing 0 by a program during the following period.

- Hardware protection is started
  - From the acceptance filter processing (the beginning of CRC field)
- Hardware protection is released
  - If the receive FIFO is specified to receive the incoming message, after the received data is stored into the receive FIFO or a CAN bus error occurs (i.e. maximum period of hardware protection is from the beginning of CRC field to the end of 7th bit of EOF)
  - If the receive FIFO is not specified to receive the message, after the acceptance filter processing

#### RFUST[2:0] Bits (Receive FIFO Unread Message Number Status)

The RFUST[2:0] bits indicate the number of unread messages in the receive FIFO.

The value of the RFUST[2:0] bits is initialized to 000b when the RFE bit is set to 0.

**RFMLF Flag (Receive FIFO Message Lost Flag)**

The RFMLF bit is set to 1 (receive FIFO message lost has occurred) when the receive FIFO receives a new message and the receive FIFO is full. The timing for setting this bit to 1 is at the end of the 6th bit of EOF.

The RFMLF bit is set to 0 by writing 0 by a program (writing 1 has no effect). In both overwrite and overrun modes, the RFMLF bit cannot be set to 0 (receive FIFO message lost has not occurred) by writing 0 by a program due to hardware protection during five peripheral module clock (PCLK) cycles following the sixth bit of EOF, if the receive FIFO is full and determined to receive a message.

**RFFST Flag (Receive FIFO Full Status Flag)**

The RFFST bit is set to 1 (receive FIFO is full) when the number of unread messages in the receive FIFO is 4. The RFFST bit is 0 (receive FIFO is not full) when the number of unread messages in the receive FIFO is less than 4. The RFFST bit is set to 0 when the RFE bit is 0.

**RFWST Flag (Receive FIFO Buffer Warning Status Flag)**

The RFWST bit is set to 1 (receive FIFO is buffer warning) when the number of unread messages in the receive FIFO is 3. The RFWST bit is 0 (receive FIFO is not buffer warning) when the number of unread messages in the receive FIFO is less than 3 or equal to 4. The RFWST bit is set to 0 when the RFE bit is 0.

**RFEST Flag (Receive FIFO Empty Status Flag)**

The RFEST bit is set to 1 (no unread message in receive FIFO) when the number of unread messages in the receive FIFO is 0. The RFEST bit is set to 1 when the RFE bit is set to 0. The RFEST bit is set to 0 (unread message in receive FIFO) when the number of unread messages in the receive FIFO is one or more.

Figure 34.2 shows the receive FIFO mailbox operation.

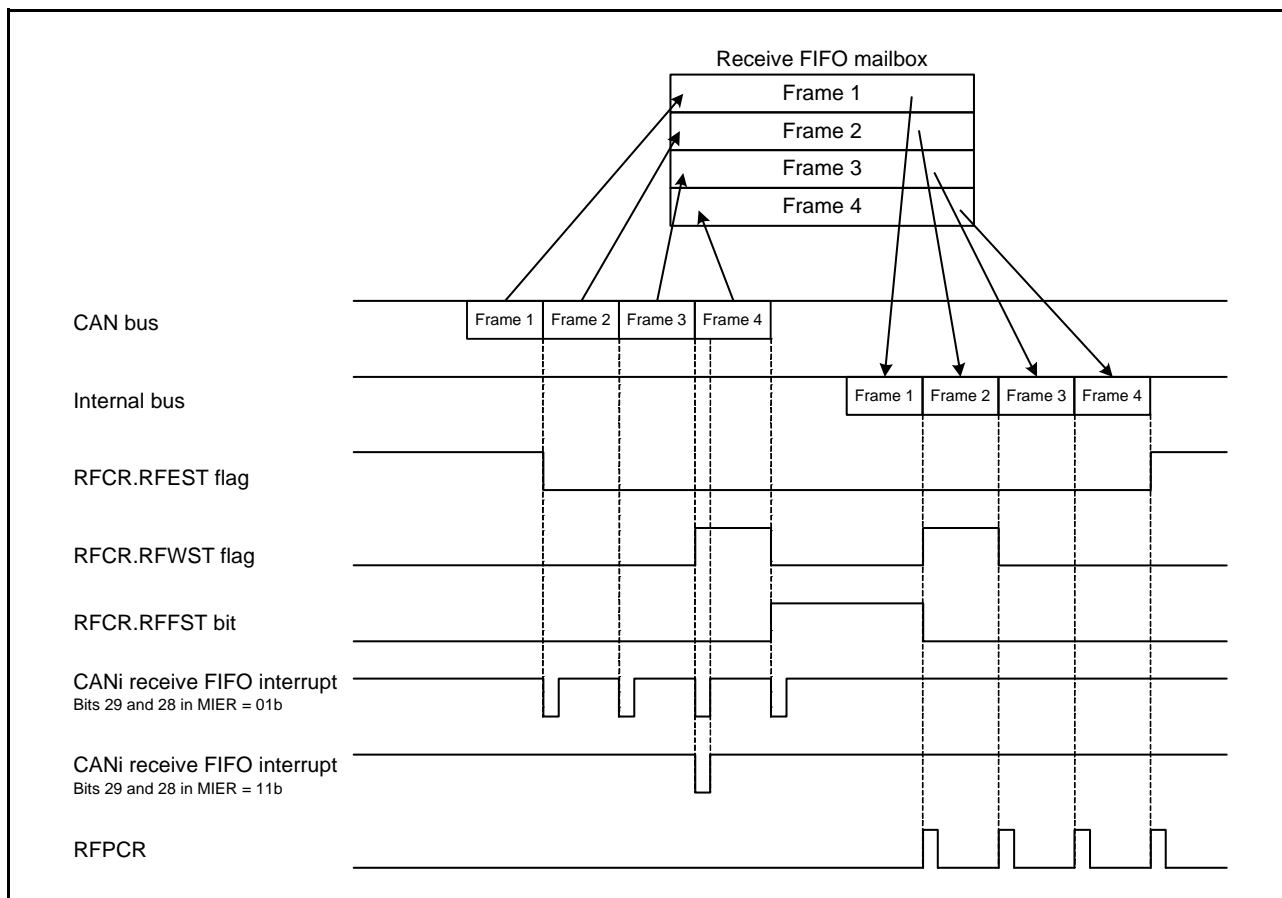
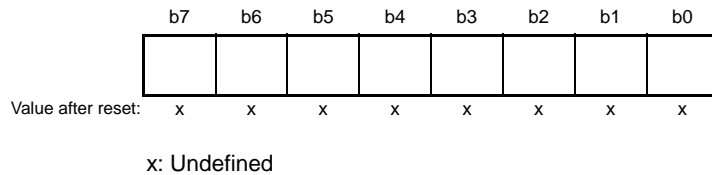


Figure 34.2 Receive FIFO Mailbox Operation (Bits 29 and 28 in MIER = 01b or 11b)

### 34.2.10 Receive FIFO Pointer Control Register (RFPCR)

Address(es): CAN0.RFPCR 0009 0849h, CAN1.RFPCR 009 1849h, CAN2.RFPCR 0009 2849h



Bit	Description	R/W
b7 to b0	The CPU-side pointer for the receive FIFO is incremented by writing FFh to RFPCR.	W

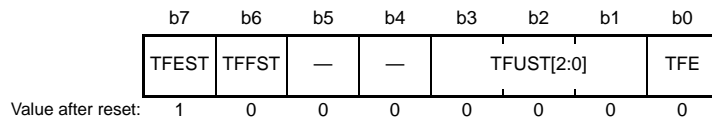
When the receive FIFO is not empty, write FFh to RFPCR by a program to increment the CPU-side pointer for the receive FIFO to the next mailbox location.

Do not write to RFPCR when the RFE bit in RFCR is 0 (receive FIFO disabled).

Both the CAN-side pointer and the CPU-side pointer are incremented when a new message is received and the RFFST bit is 1 (receive FIFO is full) in overwrite mode. When the RFMLF bit is 1 in this condition, the CPU-side pointer cannot be incremented by writing to RFPCR by a program.

### 34.2.11 Transmit FIFO Control Register (TFCR)

Address(es): CAN0.TFCR 0009 084Ah, CAN1.TFCR 0009 184Ah, CAN2.TFCR 0009 284Ah



Bit	Symbol	Bit Name	Description	R/W
b0	TFE	Transmit FIFO Enable	0: Transmit FIFO disabled 1: Transmit FIFO enabled	R/W
b3 to b1	TFUST[2:0]	Transmit FIFO Unsent Message Number Status	b3 b2 b1 0 0 0: No unsent message 0 0 1: 1 unsent message 0 1 0: 2 unsent messages 0 1 1: 3 unsent messages 1 0 0: 4 unsent messages 1 0 1: Reserved 1 1 0: Reserved 1 1 1: Reserved	R
b5, b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b6	TFFST	Transmit FIFO Full Status	0: Transmit FIFO is not full 1: Transmit FIFO is full (4 unsent messages)	R
b7	TFEST	Transmit FIFO Empty Status	0: Unsent message in transmit FIFO 1: No unsent message in transmit FIFO	R

Write to TFCR in CAN operation mode or CAN halt mode.

**TFE Bit (Transmit FIFO Enable)**

When the TFE bit is set to 1, the transmit FIFO is enabled.

When the TFE bit is set to 0, the transmit FIFO becomes empty (TFEST bit = 1) and then unsent messages from the transmit FIFO are lost as described below:

- Immediately if a message from the transmit FIFO is not scheduled for the next transmission or during transmission
- Following the completion of transmission, a CAN bus error, CAN bus arbitration-lost, or entry to CAN halt mode if a message from the transmit FIFO is scheduled for the next transmission or already during transmission

Before setting the TFE bit to 1 again, ensure that the TFEST bit has been set to 1. After setting the TFE bit to 1, write transmit data into MB23.

Do not set the TFE bit to 1 in normal mailbox mode (MBM bit in CTRLR = 0).

**TFUST[2:0] Bits (Transmit FIFO Unsent Message Number Status)**

The TFUST[2:0] bits indicate the number of unsent messages in the transmit FIFO.

The TFUST[2:0] bits are set to 000b after TFE bit is cleared to 0 and transmission is aborted or completed.

**TFFST Bit (Transmit FIFO Full Status)**

The TFFST bit is set to 1 (transmit FIFO is full) when the number of unsent messages in the transmit FIFO is 4. The TFFST bit is set to 0 (transmit FIFO is not full) when the number of unsent messages in the transmit FIFO is less than 4. The TFFST bit is set to 0 when transmission from the transmit FIFO has been aborted.

**TFEST Bit (Transmit FIFO Empty Status)**

The TFEST bit is set to 1 (no message in transmit FIFO) when the number of unsent messages in the transmit FIFO is 0.

The TFEST bit is set to 1 when transmission from the transmit FIFO has been aborted. The TFEST bit is set to 0 (message in transmit FIFO) when the number of unsent messages in the transmit FIFO is not 0.

Figure 34.3 shows the transmit FIFO mailbox operation.

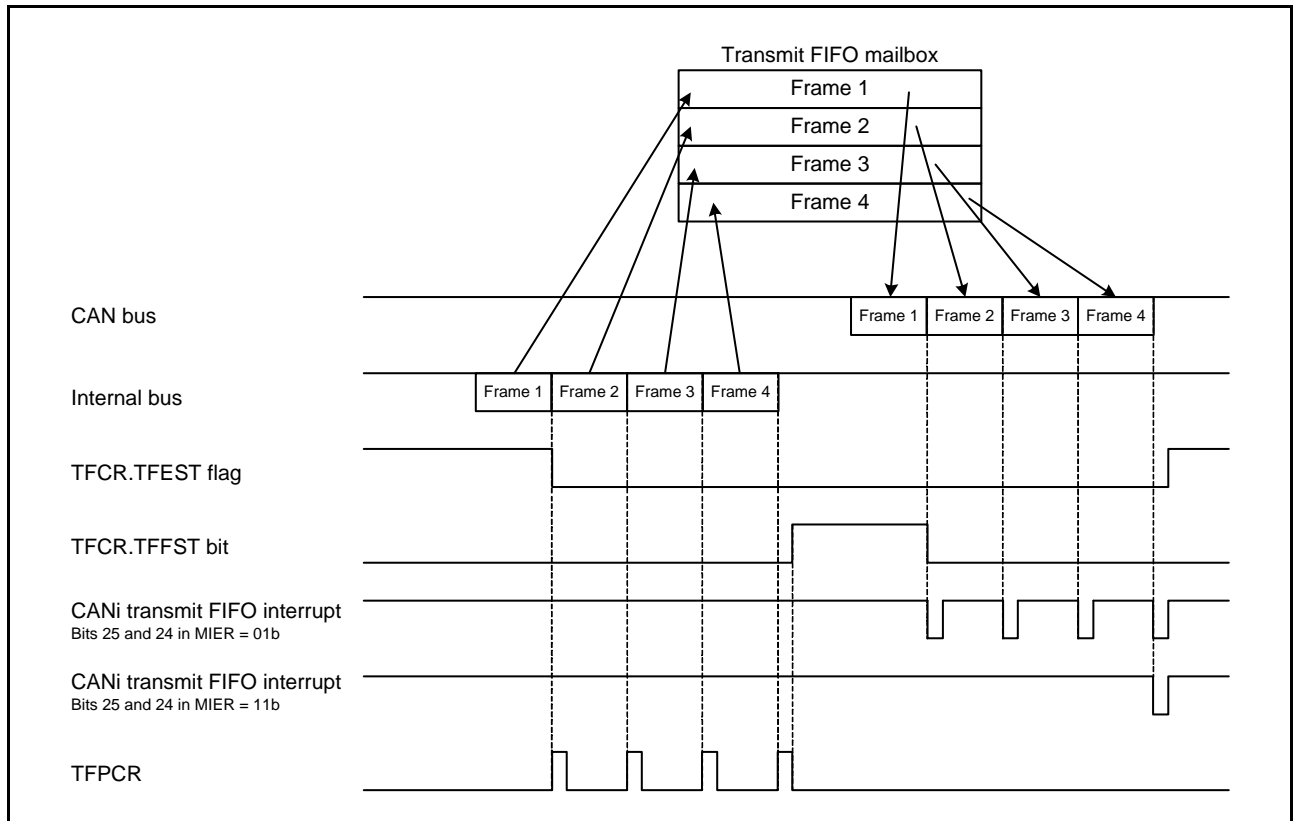
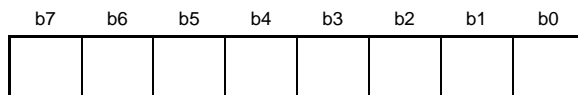


Figure 34.3 Transmit FIFO Mailbox Operation (Bits 25 and 24 in MIER = 01b or 11b)

### 34.2.12 Transmit FIFO Pointer Control Register (TFPCR)

Address(es): CAN0.TFPCR 0009 084Bh, CAN1.TFPCR 0009 184Bh, CAN2.TFPCR 0009 284Bh



Value after reset: x x x x x x x x

x: Undefined

Bit	Description	R/W
b7 to b0	The CPU-side pointer for the transmit FIFO is incremented by writing FFh to TFPCR.	W

When the transmit FIFO is not full, write FFh to TFPCR by a program to increment the CPU-side pointer for the transmit FIFO to the next mailbox location.

Do not write to TFPCR when the TFE bit in TFCR is 0 (transmit FIFO disabled).

### 34.2.13 Status Register (STR)

Address(es): CAN0.STR 0009 0842h, CAN1.STR 0009 1842h, CAN2.STR 0009 2842h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	RECST	TRMST	BOST	EPST	SLPST	HLTST	RSTST	EST	TABST	FMLST	NMLST	TFST	RFST	SDST	NDST
Value after reset:	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	NDST	NEWDATA Status Flag	0: No mailbox with NEWDATA bit = 1 1: Mailbox(es) with NEWDATA bit = 1	R
b1	SDST	SENTDATA Status Flag	0: No mailbox with SENTDATA bit = 1 1: Mailbox(es) with SENTDATA bit = 1	R
b2	RFST	Receive FIFO Status Flag	0: No message in receive FIFO (empty) 1: Message in receive FIFO	R
b3	TFST	Transmit FIFO Status Flag	0: Transmit FIFO is full 1: Transmit FIFO is not full	R
b4	NMLST	Normal Mailbox Message Lost Status Flag	0: No mailbox with MSGLOST bit = 1 1: Mailbox(es) with MSGLOST bit = 1	R
b5	FMLST	FIFO Mailbox Message Lost Status Flag	0: RFMLF bit = 0 1: RFMLF bit = 1	R
b6	TABST	Transmission Abort Status Flag	0: No mailbox with TRMABT bit = 1 1: Mailbox(es) with TRMABT bit = 1	R
b7	EST	Error Status Flag	0: No error occurred 1: Error occurred	R
b8	RSTST	CAN Reset Status Flag	0: Not in CAN reset mode 1: In CAN reset mode	R
b9	HLTST	CAN Halt Status Flag	0: Not in CAN halt mode 1: In CAN halt mode	R
b10	SLPST	CAN Sleep Status Flag	0: Not in CAN sleep mode 1: In CAN sleep mode	R
b11	EPST	Error-Passive Status Flag	0: Not in error-passive state 1: In error-passive state	R
b12	BOST	Bus-Off Status Flag	0: Not in bus-off state 1: In bus-off state	R
b13	TRMST	Transmit Status Flag (transmitter)	0: Bus idle or reception in progress 1: Transmission in progress or in bus-off state	R
b14	RECST	Receive Status Flag (receiver)	0: Bus idle or transmission in progress 1: Reception in progress	R
b15	—	Reserved	The read value is 0.	R

#### NDST Flag (NEWDATA Status Flag)

The NDST bit is set to 1 when at least one NEWDATA bit in MCTLj (j = 0 to 31) is 1 regardless of the value of MIER. The NDST bit is set to 0 when all NEWDATA bits are 0.

#### SDST Flag (SENTDATA Status Flag)

The SDST bit is set to 1 when at least one SENTDATA bit in MCTLj (j = 0 to 31) is 1 regardless of the value of MIER. The SDST bit is set to 0 when all SENTDATA bits are 0.



**RFST Flag (Receive FIFO Status Flag)**

The RFST bit is set to 1 when the receive FIFO is not empty. The RFST bit is set to 0 when the receive FIFO is empty or normal mailbox mode is selected.

**TFST Flag (Transmit FIFO Status Flag)**

The TFST bit is set to 1 when the transmit FIFO is not full. The TFST bit is set to 0 when the transmit FIFO is full or normal mailbox mode is selected.

**NMLST Flag (Normal Mailbox Message Lost Status Flag)**

The NMLST bit is set to 1 when at least one MSGLOST bit in MCTLj ( $i = 0$  to  $2$ ,  $j = 0$  to  $31$ ) is 1 regardless of the value of MIER. The NMLST bit is set to 0 when all MSGLOST bits are 0.

**FMLST Flag (FIFO Mailbox Message Lost Status Flag)**

The FMLST bit is set to 1 when the RFMLF bit in RFCR is 1 regardless of the value of MIER. The FMLST bit is set to 0 when the RFMLF bit is 0.

**TABST Flag (Transmission Abort Status Flag)**

The TABST bit is set to 1 when at least one TRMABT bit in MCTLj ( $i = 0$  to  $2$ ,  $j = 0$  to  $31$ ) is 1 regardless of the value of MIER. The TABST bit is set to 0 when all TRMABT bits are 0.

**EST Flag (Error Status Flag)**

The EST bit is set to 1 when at least one error is detected by EIFR regardless of the value of EIER. The EST bit is set to 0 when no error is detected by EIFR.

**RSTST Flag (CAN Reset Status Flag)**

The RSTST bit is set to 1 when the CAN module is in CAN reset mode. The RSTST bit is 0 when the CAN module is not in CAN reset mode. Even when the state is changed from CAN reset mode to CAN sleep mode, the RSTST bit remains 1.

**HLTST Flag (CAN Halt Status Flag)**

The HLTST bit is set to 1 when the CAN module is in CAN halt mode. The HLTST bit is set to 0 when the CAN module is not in CAN halt mode. Even when the state is changed from CAN halt mode to CAN sleep mode, the HLTST bit remains 1.

**SLPST Flag (CAN Sleep Status Flag)**

The SLPST bit is set to 1 when the CAN module is in CAN sleep mode. The SLPST bit is set to 0 when the CAN module is not in CAN sleep mode.

**EPST Flag (Error-Passive Status Flag)**

The EPST bit is set to 1 when the value of TECR or RECR exceeds 127 and the CAN module is in the error-passive state ( $128 \leq \text{TEC} < 256$  or  $128 \leq \text{REC} < 256$ ). The EPST bit is set to 0 when the CAN module is not in the error-passive state.

**BOST Flag (Bus-Off Status Flag)**

The BOST bit is set to 1 when the value of TECR exceeds 255 and the CAN module is in the bus-off state ( $\text{TEC} \geq 256$ ). The BOST bit is set to 0 when the CAN module is not in the bus-off state.

**TRMST Flag (Transmit Status Flag) (transmitter)**

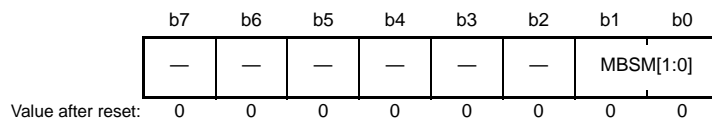
The TRMST bit is set to 1 when the CAN module performs as a transmitter node or is in the bus-off state. The TRMST bit is set to 0 when the CAN module performs as a receiver node or is in the bus-idle state.

**RECST Flag (Receive Status Flag) (receiver)**

The RECST bit is set to 1 when the CAN module performs as a receiver node. The RECST bit is set to 0 when the CAN module performs as a transmitter node or is in the bus-idle state.

**34.2.14 Mailbox Search Mode Register (MSMR)**

Address(es): CAN0.MSMR 0009 0853h, CAN1.MSMR 0009 1853h, CAN2.MSMR 0009 2853h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	MBSM[1:0]	Mailbox Search Mode Select	b1 b0 0 0: Receive mailbox search mode 0 1: Transmit mailbox search mode 1 0: Message lost search mode 1 1: Channel search mode	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Write to MSMR in CAN operation mode or CAN halt mode.

**MBSM[1:0] Bits (Mailbox Search Mode Select)**

The MBSM[1:0] bits select the search mode for the mailbox search function.

When the MBSM[1:0] bits are 00b, receive mailbox search mode is selected. In this mode, the search targets are the NEWDATA bit in MCTLj (j = 0 to 31) for the normal mailbox and the RFEST bit in RFCR.

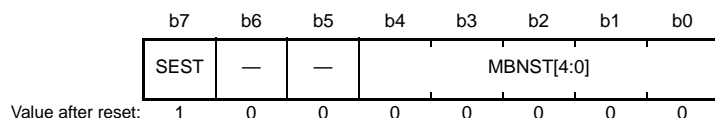
When the MBSM[1:0] bits are 01b, transmit mailbox search mode is selected. In this mode, the search target is the SENTDATA bit in MCTLj.

When the MBSM[1:0] bits are 10b, message lost search mode is selected. In this mode, the search targets are the MSGLOST bit in MCTLj for the normal mailbox and the RFMLF bit in RFCR.

When the MBSM[1:0] bits are 11b, channel search mode is selected. In this mode, the search target is CSSR. Refer to section 34.2.16, Channel Search Support Register (CSSR).

### 34.2.15 Mailbox Search Status Register (MSSR)

Address(es): CAN0.MSSR 0009 0852h, CAN1.MSSR 0009 1852h, CAN2.MSSR 0009 2852h



Bit	Symbol	Bit Name	Description	R/W
b4 to b0	MBNST[4:0]	Search Result Mailbox Number Status	These bits output the smallest mailbox number that is searched in each mode of MSMR.	R
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	SEST	Search Result Status	0: Search result found 1: No search result	R

#### MBNST[4:0] Bits (Search Result Mailbox Number Status)

The MBNST[4:0] bits output the smallest mailbox number that is searched in each mode of MSMR. In receive mailbox search mode, transmit mailbox search mode, and message lost search mode, the value of the mailbox i.e., the search result to be output, is updated as described below:

- When the NEWDATA, SENTDATA or MSGLOST bit for the output mailbox is set to 0
- When the NEWDATA, SENTDATA or MSGLOST bit for a higher-priority mailbox is set to 1

If the MBSM[1:0] bits are set to 00b (receive mailbox search mode) or 10b (message lost search mode), the receive FIFO (mailbox [28]) is output when the receive FIFO is not empty and there are no unread received messages or no lost messages in any of the normal mailboxes (mailboxes [0] to [23]). If the MBSM[1:0] bits are set to 01b (transmit mailbox search mode), the transmit FIFO (mailbox [24]) is not output. Table 34.6 lists the behavior of the MBNST[4:0] bits in FIFO mailbox mode.

In channel search mode, the MBNST[4:0] bits output the corresponding channel number. After MSSR is read by a program, the next target channel number is output.

#### SEST Bit (Search Result Status)

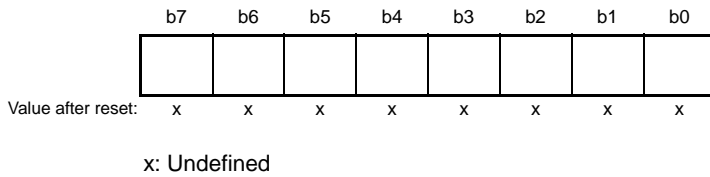
The SEST bit is set to 1 (no search result) when no corresponding mailbox is found after searching all mailboxes. For example, in transmit mailbox search mode, the SEST bit is set to 1 when no SENTDATA bit for mailboxes is 1. The SEST bit is set to 0 when at least one SENTDATA bit is 1. When the SEST bit is 1, the value of the MBNST[4:0] bits is undefined.

**Table 34.6 Behavior of MBNST[4:0] Bits in FIFO Mailbox Mode**

MBSM[1:0] Bits	Mailbox [24] (Transmit FIFO)	Mailbox [28] (Receive FIFO)
00b	Mailbox [24] is not output.	Mailbox [28] is output when no MCTLj.NEWDATA bit for the normal mailboxes is set to 1 (new message is being stored or has been stored to the mailbox) and the receive FIFO is not empty.
01b		Mailbox [28] is not output.
10b		Mailbox [28] is output when no MCTLj.MSGLOST bit for the normal mailboxes is set to 1 (message is overwritten or overrun) and the RFCR.RFMLF bit is set to 1 (receive FIFO message lost has occurred) in the receive FIFO.
11b		Mailbox [28] is not output.

### 34.2.16 Channel Search Support Register (CSSR)

Address(es): CAN0.CSSR 0009 0851h, CAN1.CSSR 0009 1851h, CAN2.CSSR 0009 2851h



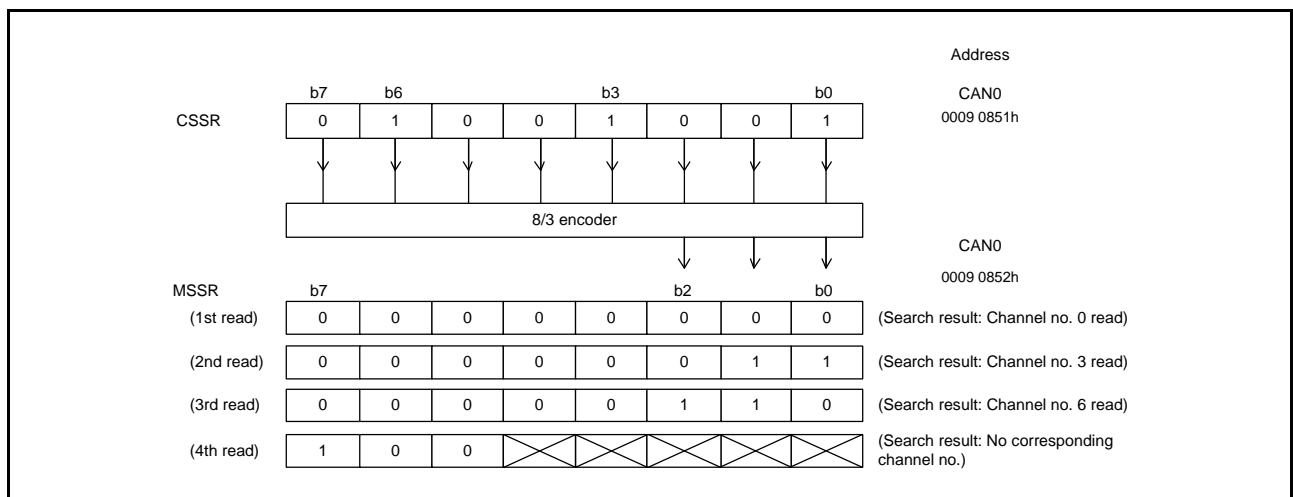
Bit	Description	R/W
b7 to b0	When the value for the channel search is input, the channel number is output to MSSR.	R/W

The bits in CSSR, which are set to 1, are encoded by an 8/3 encoder (the LSB position has the higher priority) and output to the MBNST[4:0] bits in MSSR.

MSSR outputs the updated value whenever MSSR is read by a program.

Write to CSSR only when the MSMR.MBSM[1:0] bits are 11b (channel search mode). Write to CSSR in CAN operation mode or CAN halt mode.

Figure 34.4 shows the write and read of CSSR and MSSR.

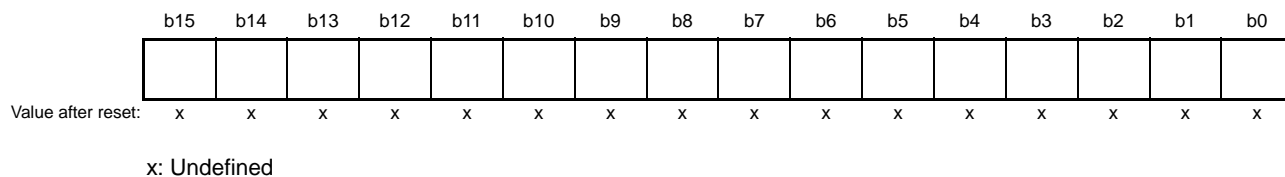


**Figure 34.4 Write and Read of CSSR and MSSR**

The value of CSSR is also updated whenever MSSR is read. When read, the value prior to conversion by the 8/3 encoder can be read.

### 34.2.17 Acceptance Filter Support Register (AFSR)

Address(es): CAN0.AFSR 0009 0856h, CAN1.AFSR 0009 1856h, CAN2.AFSR 0009 2856h



Bit	Description	R/W
b15 to b0	After the standard ID of a received message is written, the value converted for data table search can be read.	R/W

Note: • Write to AFSR in CAN operation mode or CAN halt mode.

The acceptance filter support unit (ASU) can be used for data table (8 bits × 256) search. In the data table, all standard IDs created by the user are set to be valid/invalid in bit units. When AFSR is written with data in 16-bit units including the SID[10:0] bit in MBj (j = 0 to 31), in which a received standard ID is stored, a decoded row (byte offset) position and column (bit) position for data table search can be read. The ASU can be used for standard (11-bit) IDs only.

The ASU is enabled in the following cases:

- When the ID to receive cannot be masked by the acceptance filter  
(Example) IDs to receive: 078h, 087h, and 111h
- When there are too many IDs to receive and software filtering time is expected to be shortened  
It should be noted that AFSR cannot be set in CAN reset mode.

Figure 34.5 shows the write and read of AFSR.

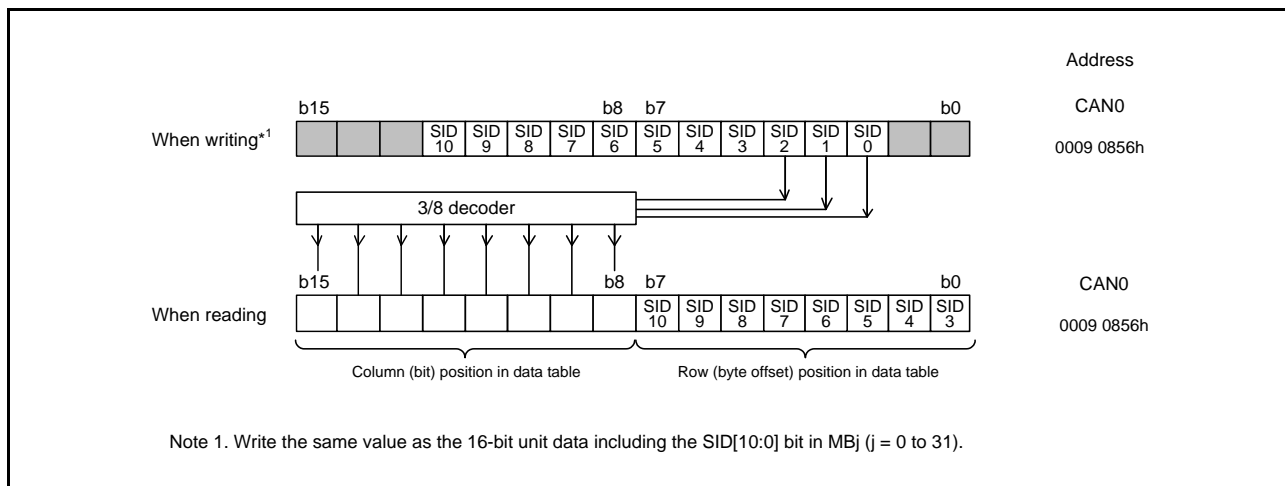


Figure 34.5 Write and Read of AFSR

### 34.2.18 Error Interrupt Enable Register (EIER)

Address(es): CAN0.EIER 0009 084Ch, CAN1.EIER 0009 184Ch, CAN2.EIER 0009 284Ch

b7	b6	b5	b4	b3	b2	b1	b0
BLIE	OLIE	ORIE	BORIE	BOEIE	EPIE	EWIE	BEIE
Value after reset:	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	BEIE	Bus Error Interrupt Enable	0: Bus error interrupt disabled 1: Bus error interrupt enabled	R/W
b1	EWIE	Error-Warning Interrupt Enable	0: Error-warning interrupt disabled 1: Error-warning interrupt enabled	R/W
b2	EPIE	Error-Passive Interrupt Enable	0: Error-passive interrupt disabled 1: Error-passive interrupt enabled	R/W
b3	BOEIE	Bus-Off Entry Interrupt Enable	0: Bus-off entry interrupt disabled 1: Bus-off entry interrupt enabled	R/W
b4	BORIE	Bus-Off Recovery Interrupt Enable	0: Bus-off recovery interrupt disabled 1: Bus-off recovery interrupt enabled	R/W
b5	ORIE	Overrun Interrupt Enable	0: Receive overrun interrupt disabled 1: Receive overrun interrupt enabled	R/W
b6	OLIE	Overload Frame Transmit Interrupt Enable	0: Overload frame transmit interrupt disabled 1: Overload frame transmit interrupt enabled	R/W
b7	BLIE	Bus Lock Interrupt Enable	0: Bus lock interrupt disabled 1: Bus lock interrupt enabled	R/W

EIER is used to enable or disable the error interrupt individually for each error interrupt source in EIFR. Write to EIER in CAN reset mode.

#### BEIE Bit (Bus Error Interrupt Enable)

When the BEIE bit is 0, no error interrupt request is generated even if the BEIF bit in EIFR is set to 1. When the BEIE bit is 1, an error interrupt request is generated if the BEIF bit is set to 1.

#### EWIE Bit (Error-Warning Interrupt Enable)

When the EWIE bit is 0, no error interrupt request is generated even if the EWIF bit in EIFR is set to 1. When the EWIE bit is 1, an error interrupt request is generated if the EWIF bit is set to 1.

#### EPIE Bit (Error-Passive Interrupt Enable)

When the EPIE bit is 0, no error interrupt request is generated even if the EPIF bit in EIFR is set to 1. When the EPIE bit is 1, an error interrupt request is generated if the EPIF bit is set to 1.

#### BOEIE Bit (Bus-Off Entry Interrupt Enable)

When the BOEIE bit is 0, no error interrupt request is generated even if the BOEIF bit in EIFR is set to 1. When the BOEIE bit is 1, an error interrupt request is generated if the BOEIF bit is set to 1.

#### BORIE Bit (Bus-Off Recovery Interrupt Enable)

When the BORIE bit is 0, an error interrupt request is not generated even if the BORIF bit in EIFR is set to 1. When the BORIE bit is set to 1, an error interrupt request is generated if the BORIF bit is set to 1.

**ORIE Bit (Overrun Interrupt Enable)**

When the ORIE bit is 0, an error interrupt request is not generated even if the ORIF bit in EIFR is set to 1. When the ORIE bit is 1, an error interrupt request is generated if the ORIF bit is set to 1.

**OLIE Bit (Overload Frame Transmit Interrupt Enable)**

When the OLIE bit is 0, no error interrupt request is generated even if the OLIF bit in EIFR is set to 1. When the OLIE bit is 1, an error interrupt request is generated if the OLIF bit is set to 1.

**BLIE Bit (Bus Lock Interrupt Enable)**

When the BLIE bit is 0, no error interrupt request is generated even if the BLIF bit in EIFR is set to 1. When the BLIE bit is 1, an error interrupt request is generated if the BLIF bit is set to 1.

**34.2.19 Error Interrupt Factor Judge Register (EIFR)**

Address(es): CAN0.EIFR 0009 084Dh, CAN1.EIFR 0009 184Dh, CAN2.EIFR 0009 284Dh

b7	b6	b5	b4	b3	b2	b1	b0
BLIF	OLIF	ORIF	BORIF	BOEIF	EPIF	EWIF	BEIF

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	BEIF	Bus Error Detect Flag	0: No bus error detected 1: Bus error detected	R/W
b1	EWIF	Error-Warning Detect Flag	0: No error-warning detected 1: Error-warning detected	R/W
b2	EPIF	Error-Passive Detect Flag	0: No error-passive detected 1: Error-passive detected	R/W
b3	BOEIF	Bus-Off Entry Detect Flag	0: No bus-off entry detected 1: Bus-off entry detected	R/W
b4	BORIF	Bus-Off Recovery Detect Flag	0: No bus-off recovery detected 1: Bus-off recovery detected	R/W
b5	ORIF	Receive Overrun Detect Flag	0: No receive overrun detected 1: Receive overrun detected	R/W
b6	OLIF	Overload Frame Transmission Detect Flag	0: No overload frame transmission detected 1: Overload frame transmission detected	R/W
b7	BLIF	Bus Lock Detect Flag	0: No bus lock detected 1: Bus lock detected	R/W

If an event corresponding to each bit occurs, the corresponding bit in EIFR is set to 1 regardless of the setting of EIER. To set each bit to 0, write 0 by a program. If the set timing occurs simultaneously with the clear timing by the program, the bit becomes 1.

When a single bit is set to 0 by a program, do not use the logic operation instruction (AND) – use the transfer instruction (MOV) to ensure that only the specified bit is set to 0 and the other bits are set to 1. Writing 1 has no effect to these bit values.

**BEIF Flag (Bus Error Detect Flag)**

The BEIF bit is set to 1 when a bus error is detected.

**EWIF Flag (Error-Warning Detect Flag)**

The EWIF bit is set to 1 when the value of the receive error counter (REC) or transmit error counter (TEC) exceeds 95. The EWIF bit is set to 1 only when the REC or TEC initially exceeds 95. Thus, if 0 is written to the EWIF bit by a program while the REC or TEC remains greater than 95, the EWIF bit is not set to 1 until the REC and TEC go below 95 and then REC or TEC exceeds 95 again.

**EPIF Flag (Error-Passive Detect Flag)**

The EPIF bit is set to 1 when the CAN error state becomes error-passive (the REC (receive error counter) or TEC (transmit error counter) value exceeds 127).

The EPIF bit is set to 1 only when the REC or TEC initially exceeds 127. Thus, if 0 is written by a program while the REC or TEC remains greater than 127, the EPIF bit is not set to 1 until the REC and TEC go below 127 and then REC or TEC exceeds 127 again.

**BOEIF Flag (Bus-Off Entry Detect Flag)**

The BOEIF bit is set to 1 when the CAN error state becomes bus-off (the TEC (transmit error counter) value exceeds 255). The BOEIF bit is also set to 1 when the BOM[1:0] bits in CTRLR are 01b (entry to CAN halt mode automatically at bus-off entry) and the CAN module becomes the bus-off state.

**BORIF Flag (Bus-Off Recovery Detect Flag)**

The BORIF bit is set to 1 when the CAN module recovers from the bus-off state normally by detecting 11 consecutive bits 128 times in the following conditions:

- When the BOM[1:0] bits in CTRLR are 00b
- When the BOM[1:0] bits in CTRLR are 10b
- When the BOM[1:0] bits in CTRLR are 11b

However, the BORIF bit is not set to 1 if the CAN module recovers from the bus-off state in the following conditions:

- When the CANM[1:0] bits in CTRLR are set to 01b or 11b (CAN reset mode)
- When the RBOC bit in CTRLR is set to 1 (forcible return from bus-off)
- When the BOM[1:0] bits in CTRLR are set to 01b
- When the BOM[1:0] bits in CTRLR are set to 11b and the CANM[1:0] bits in CTRLR are set to 10b (CAN halt mode) before normal recovery occurs

**ORIF Flag (Receive Overrun Detect Flag)**

The ORIF bit is set to 1 when a receive overrun occurs. This bit is not set to 1 in overwrite mode.

In overwrite mode, a reception complete interrupt request is generated if an overwrite condition occurs and the ORIF bit is not set to 1.

In normal mailbox mode, if an overrun occurs in any of mailboxes [0] to [31] in overrun mode, this bit is set to 1. In FIFO mailbox mode, if an overrun occurs in any of mailboxes [0] to [23] or the receive FIFO in overrun mode, this bit is set to 1.

**OLIF Flag (Overload Frame Transmission Detect Flag)**

The OLIF bit is set to 1 if the transmitting condition of an overload frame is detected when the CAN module performs transmission or reception.



**BLIF Flag (Bus Lock Detect Flag)**

The BLIF bit is set to 1 if 32 consecutive dominant bits are detected on the CAN bus while the CAN module is in CAN operation mode.

After the BLIF bit is set to 1, 32 consecutive dominant bits are detected again under either of the following conditions:

- After this bit is set to 0 from 1, recessive bits are detected
- After this bit is set to 0 from 1, the CAN module enters CAN reset mode or CAN halt mode and then enters CAN operation mode again.

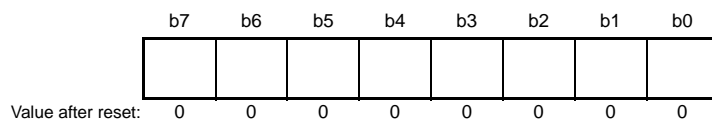
Table 34.7 lists the behavior of bits BOEIF and BORIF according to the CTLR.BOM[1:0] bit setting.

**Table 34.7 Behavior of BOEIF and BORIF Flags according to CTLR.BOM[1:0] Bit Setting**

BOM[1:0] Bits	BOEIF Bit	BORIF Bit
00b	Set to 1 on entry to the bus-off state.	Set to 1 on exit from the bus-off state.
01b		Do not set to 1.
10b		Set to 1 on exit from the bus-off state.
11b		Set to 1 if normal bus-off recovery occurs before the CANM[1:0] bits are set to 10b (CAN halt mode).

**34.2.20 Receive Error Count Register (RECR)**

Address(es): CAN0.RECR 0009 084Eh, CAN1.RECR 0009 184Eh, CAN2.RECR 0009 284Eh



Bit	Description	R/W
b7 to b0	Receive error count function CiRECR increments or decrements the counter value according to the error status of the CAN module during reception.	R

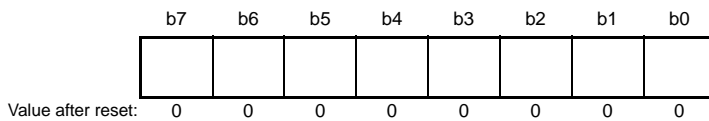
RECR indicates the value of the receive error counter.

Refer to the CAN Specifications (ISO11898-1) about the increment/decrement conditions of the receive error counter.

The value of RECR in the bus-off state is undefined.

### 34.2.21 Transmit Error Count Register (TECR)

Address(es): CAN0.TECR 0009 084Fh, CAN1.TECR 0009 184Fh, CAN2.TECR 0009 284Fh



Bit	Description	R/W
b7 to b0	Transmit error count function CiTECR increments or decrements the counter value according to the error status of the CAN module during transmission.	R

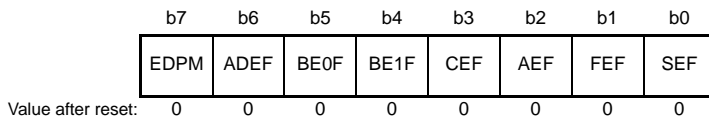
TECR indicates the value of the transmit error counter.

Refer to the CAN Specifications (ISO11898-1) about the increment/decrement conditions of the transmit error counter.

The value of TECR in the bus-off state is undefined.

### 34.2.22 Error Code Store Register (ECSR)

Address(es): CAN0.ECSR 0000 0850h, CAN1.ECSR 0000 1850h, CAN2.ECSR 0000 2850h



Bit	Symbol	Bit Name	Description	R/W
b0	SEF	Stuff Error Flag <sup>*1,*2</sup>	0: No stuff error detected 1: Stuff error detected	R/W
b1	FEF	Form Error Flag <sup>*1,*2</sup>	0: No form error detected 1: Form error detected	R/W
b2	AEF	ACK Error Flag <sup>*1,*2</sup>	0: No ACK error detected 1: ACK error detected	R/W
b3	CEF	CRC Error Flag <sup>*1,*2</sup>	0: No CRC error detected 1: CRC error detected	R/W
b4	BE1F	Bit Error (recessive) Flag <sup>*1,*2</sup>	0: No bit error (recessive) detected 1: Bit error (recessive) detected	R/W
b5	BE0F	Bit Error (dominant) Flag <sup>*1,*2</sup>	0: No bit error (dominant) detected 1: Bit error (dominant) detected	R/W
b6	ADEF	ACK Delimiter Error Flag <sup>*1,*2</sup>	0: No ACK delimiter error detected 1: ACK delimiter error detected	R/W
b7	EDPM	Error Display Mode Select <sup>*3,*4</sup>	0: Output of first detected error code 1: Output of accumulated error code	R/W

Note 1. Writing 1 has no effect to these bit values.

Note 2. To write 0 to bits SEF, FEF, AEF, CEF, BE1F, BE0F, and ADEF, do not use the logic operation instruction (AND). Use the transfer (MOV) instruction to ensure that only the specified bit is set to 0 and the other bits are set to 1.

Note 3. Write to the EDPM bit in CAN reset mode or CAN halt mode.

Note 4. If more than one error condition is detected simultaneously, all related bits are set to 1.

ECSR can be used to monitor whether an error has occurred on the CAN bus.

Refer to the CAN Specifications (ISO11898-1) to check the generation conditions of each error.

To set each bit except for the EDPM bit to 0, write 0 by a program. If the timing at which each bit is set to 1 and the timing at which 0 is written by a program are the same, the relevant bit is set to 1.

#### SEF Flag (Stuff Error Flag)

The SEF bit is set to 1 when a stuff error is detected.

#### FEF Flag (Form Error Flag)

The FEF bit is set to 1 when a form error is detected.

#### AEF Flag (ACK Error Flag)

The AEF bit is set to 1 when an ACK error is detected.

#### CEF Flag (CRC Error Flag)

The CEF bit is set to 1 when a CRC error is detected.

#### BE1F Flag (Bit Error (recessive) Flag)

The BE1F bit is set to 1 when a recessive bit error is detected.

#### BE0F Flag (Bit Error (dominant) Flag)

The BE0F bit is set to 1 when a dominant bit error is detected.

#### ADEF Flag (ACK Delimiter Error Flag)

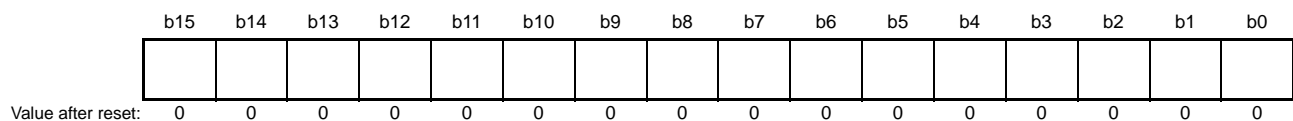
The ADEF bit is set to 1 when a form error is detected with the ACK delimiter during transmission.

#### EDPM Bit (Error Display Mode Select)

The EDPM bit selects the output mode of ECSR. When the EDPM bit is set to 0, ECSR outputs the first error code. When the EDPM bit is set to 1, ECSR outputs the accumulated error code.

### 34.2.23 Time Stamp Register (TSR)

Address(es): CAN0.TSR 0009 0854h, CAN1.TSR 0009 1854h, CAN2.TSR 0009 2854h



Bit	Description	R/W
b15 to b0	Free-running counter value for the time stamp function	R

Note: • Read TSR in 16-bit units.

When TSR is read, the value of the time stamp counter (16-bit free-running counter) at that moment is read.

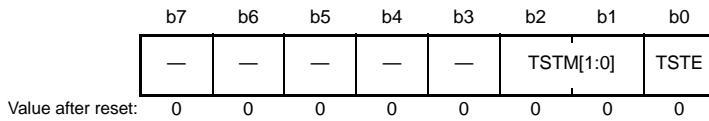
The value of the time stamp counter reference clock is a multiple of 1 bit time, as configured by the TSPS[1:0] bits in CTLR.

The time stamp counter stops in CAN sleep mode and CAN halt mode, and is initialized in CAN reset mode.

The time stamp counter value is stored to bits TSL[7:0] and TSH[7:0] in MBj when a received message is stored in a receive mailbox.

### 34.2.24 Test Control Register (TCR)

Address(es): CAN0.TCR 0009 0858h, CAN1.TCR 0009 1858h, CAN2.TCR 0009 2858h



Bit	Symbol	Bit Name	Description	R/W
b0	TSTE	CAN Test Mode Enable	0: CAN test mode disabled 1: CAN test mode enabled	R/W
b2, b1	TSTM[1:0]	CAN Test Mode Select	b2 b1 0 0: Other than CAN test mode 0 1: Listen-only mode 1 0: Self-test mode 0 (external loopback) 1 1: Self-test mode 1 (internal loopback)	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

TCR controls the CAN test mode. Write to TCR in CAN halt mode only.

#### (1) Listen-Only Mode

The CAN Specifications (ISO11898-1) recommend an optional bus monitoring mode. In listen-only mode, valid data frames and valid remote frames can be received. However, only recessive bits can be sent on the CAN bus, and the ACK bit, overload flag, and active error flag cannot be sent.

Listen-only mode can be used for baud rate detection.

Do not request transmission from any mailboxes in listen-only mode.

Figure 34.6 shows the connection when listen-only mode is selected (i = 0 to 2).

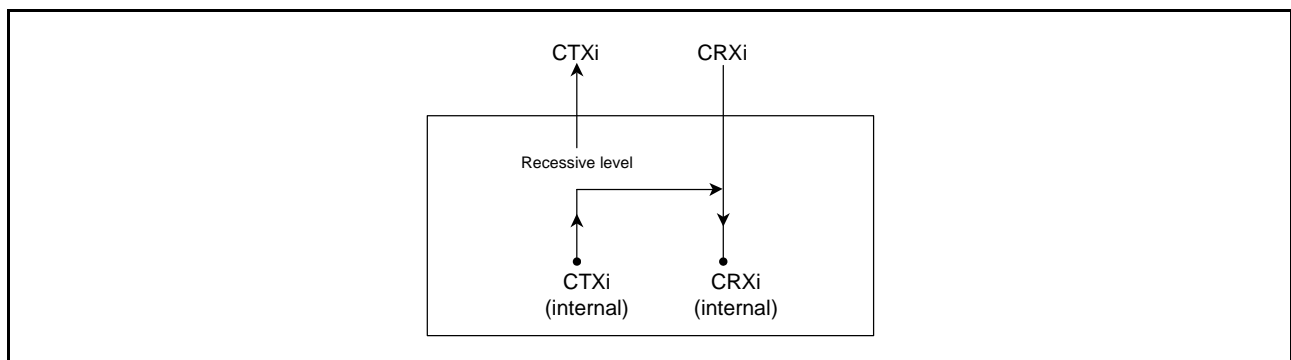


Figure 34.6 Connection when Listen-Only Mode is Selected (i = 0 to 2)

### (2) Self-Test Mode 0 (External Loopback))

Self-test mode 0 is provided for CAN transceiver tests.

In self-test mode 0, the protocol controller treats its own transmitted messages as messages received via the CAN transceiver and stores them into the receive mailbox. To be independent from external stimulation, the protocol controller generates the ACK bit.

Connect the CTXi and CRXi pins to the transceiver.

Figure 34.7 shows the connection when self-test mode 0 is selected ( $i = 0$  to 2).

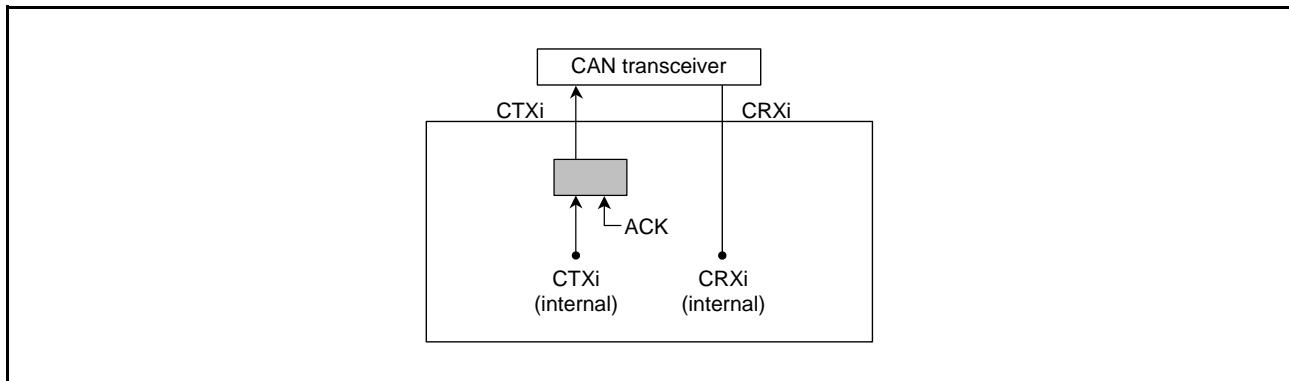


Figure 34.7 Connection when Self-Test Mode 0 is Selected ( $i = 0$  to 2)

### (3) Self-Test Mode 1 (Internal Loopback)

Self-test mode 1 is provided for self-test functions.

In self-test mode 1, the protocol controller treats its transmitted messages as received messages and stores them into the receive mailbox. To be independent from external stimulation, the protocol controller generates the ACK bit.

In self-test mode 1, the protocol controller performs an internal feedback from the internal CTXi pin to the internal CRXi pin. The input value of the external CRXi pin is ignored. The external CTXi pin outputs only recessive bits. The CTXi and CRXi pins do not need to be connected to the CAN bus or any external device.

Figure 34.8 shows the connection when self-test mode 1 is selected ( $i = 0$  to 2).

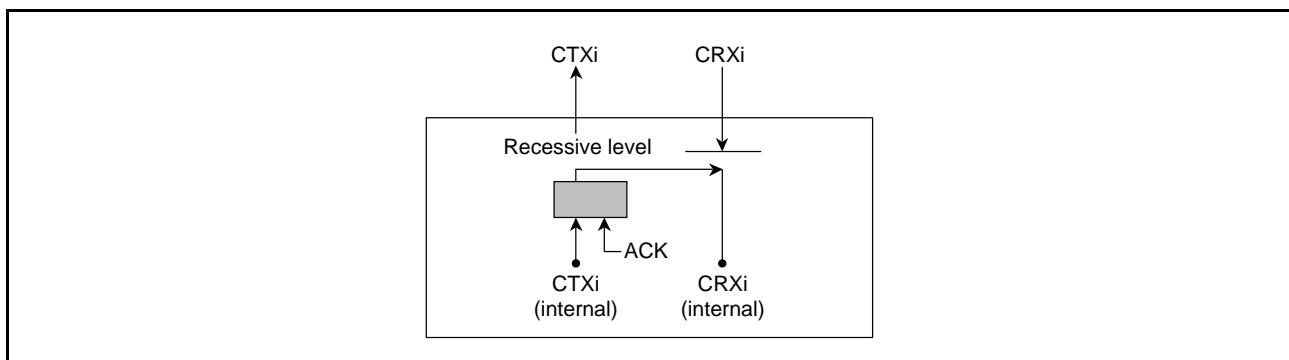


Figure 34.8 Connection when Self-Test Mode 1 is Selected ( $i = 0$  to 2)

### 34.3 Operating Mode

The CAN module has the following four operating modes.

- CAN reset mode
- CAN halt mode
- CAN operation mode
- CAN sleep mode

Figure 34.9 shows the transition between CAN operating modes.

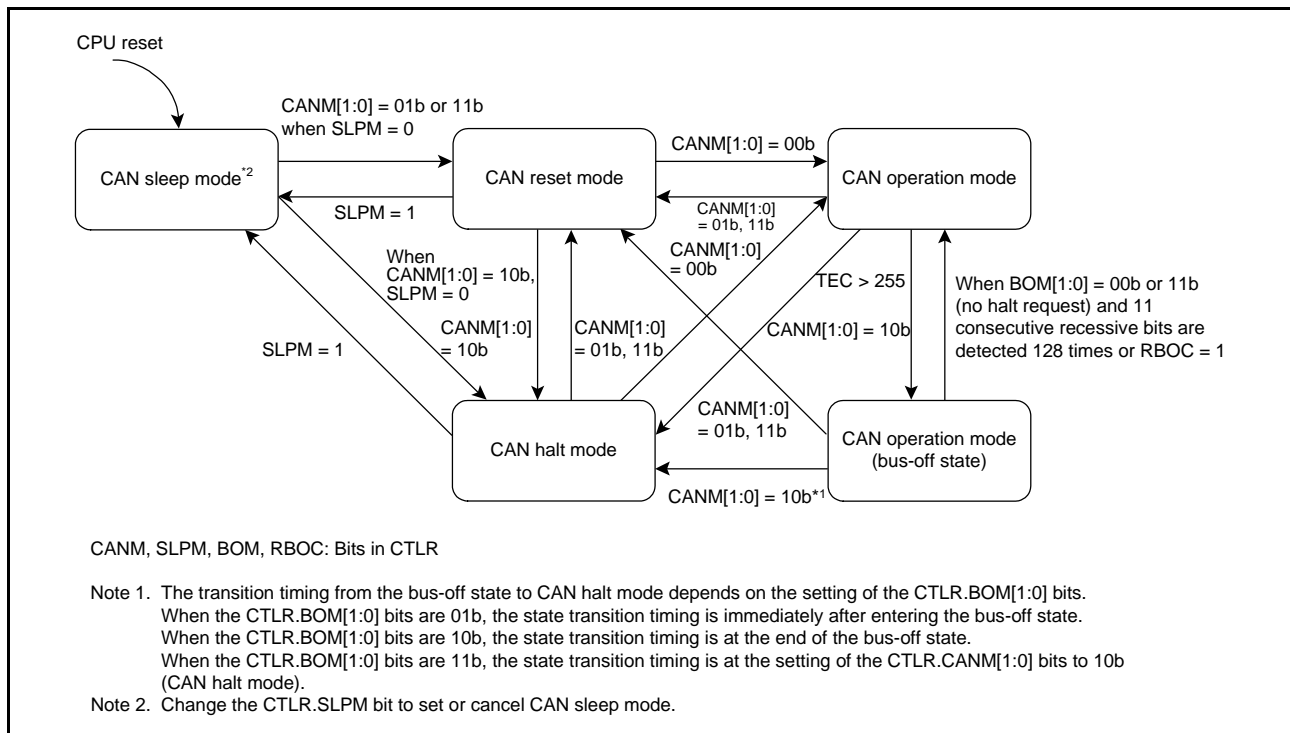


Figure 34.9 Transition between CAN Operating Modes

#### 34.3.1 CAN Reset Mode

CAN reset mode is provided for CAN communication configuration.

When the CTRLR.CANM[1:0] bits are set to 01b or 11b, the CAN module enters CAN reset mode. Then, the STR.RSTST bit is set to 1. Do not change the CTRLR.CANM[1:0] bits until the RSTST bit is set to 1. Set BCR before exiting CAN reset mode to any other modes.

The following registers are initialized to their reset values after entering CAN reset mode, and their initial values are retained during CAN reset mode:

- MCTLj
- STR (except for the SLPST and TFST bits)
- EIFR
- RECR
- TECR
- TSR
- MSSR
- MSMR

- RFCR
- TFCR
- TCR
- ECSR (except for the EDPM bit)

The following registers retain their previous values even after entering CAN reset mode.

- CTLR
- STR (only the SLPST and TFST bits)
- MIER
- EIER
- BCR
- CSSR
- ECSR (only the EDPM bit)
- MBj
- MKR0 to MKR7
- FIDCR0 and FIDCR1
- MKIVLR
- AFSR
- RFPCR
- TFPCR

### 34.3.2 CAN Halt Mode

CAN halt mode is used for mailbox configuration and test mode setting.

When the CTLR.CANM[1:0] bits are set to 10b, CAN halt mode is selected. Then the STR.HLTST bit is set to 1. Do not change the CTLR.CANM[1:0] bits until the HLTST bit is set to 1.

See Table 34.8 for the state transition conditions when transmitting or receiving.

All registers except for bits RSTST, HLTST, and SLPST in STR remain unchanged when the CAN enters CAN halt mode.

Do not change CTLR (except for bits CANM[1:0] and SLPM) and EIER in CAN halt mode. BCR can be changed in CAN halt mode only when listen-only mode is selected for automatic baud rate detection.

**Table 34.8 Operation in CAN Reset Mode and CAN Halt Mode**

Mode	Receiver	Transmitter	Bus-Off
CAN reset mode (forcible transition) CANM[1:0] = 11b	CAN module enters CAN reset mode without waiting for the end of message reception.	CAN module enters CAN reset mode without waiting for the end of message transmission.	CAN module enters CAN reset mode without waiting for the end of bus-off recovery.
CAN reset mode CANM[1:0] = 01b	CAN module enters CAN reset mode without waiting for the end of message reception.	CAN module enters CAN reset mode after waiting for the end of message transmission.*1,*4	CAN module enters CAN reset mode without waiting for the end of bus-off recovery.
CAN halt mode	CAN module enters CAN halt mode after waiting for the end of message reception.*2,*3	CAN module enters CAN halt mode after waiting for the end of message transmission.*1,*4	[When the BOM[1:0] bits are 00b] A halt request from a program will be accepted only after bus-off recovery. [When the BOM[1:0] bits are 01b] CAN module automatically enters CAN halt mode without waiting for the end of bus-off recovery (regardless of a halt request from a program). [When the BOM[1:0] bits are 10b] CAN module automatically enters CAN halt mode after waiting for the end of bus-off recovery (regardless of a halt request from a program). [When the BOM[1:0] bits are 11b] CAN module enters CAN halt mode (without waiting for the end of bus-off recovery) if a halt is requested by a program during bus-off.

BOM[1:0] bits: Bits in CTRL

- Note 1. If several messages are requested to be transmitted, mode transition occurs after the completion of the first transmission. In a case that the CAN reset mode is being requested during suspend transmission, mode transition occurs when the bus is idle, the next transmission ends, or the CAN module becomes a receiver.
- Note 2. If the CAN bus is locked at the dominant level, the program can detect this state by monitoring the BLIF bit in EIFR.
- Note 3. If a CAN bus error occurs during reception after CAN halt mode is requested, the CAN module transits to CAN halt mode.
- Note 4. If a CAN bus error or arbitration-lost occurs during transmission after CAN reset mode or CAN halt mode is requested, the CAN module transits to the requested CAN mode.

### 34.3.3 CAN Sleep Mode

CAN sleep mode is used for reducing current consumption by stopping the clock supply to the CAN module. After a reset from an MCU pin or software reset, the CAN module starts from CAN sleep mode.

When the SLPM bit in CTRL is set to 1, the CAN module enters CAN sleep mode. Then, the SLPST bit in STR is set to 1. Do not change the value of the SLPM bit until the SLPST bit is set to 1. The other registers remain unchanged when the CAN module enters CAN sleep mode.

Write to the SLPM bit in CAN reset mode and CAN halt mode. Do not change any registers (except for the SLPM bit) during CAN sleep mode. Read operation is still allowed.

When the SLPM bit is set to 0, the CAN module is released from CAN sleep mode. When the CAN module exits CAN sleep mode, the other registers remain unchanged.



### 34.3.4 CAN Operation Mode (Excluding Bus-Off State)

CAN operation mode is used for CAN communication.

When the CANM[1:0] bits in CTLR are set to 00b, the CAN module enters CAN operation mode.

Then bits RSTST and HLTST in STR are set to 0. Do not change the value of the CANM[1:0] bits until bits RSTST and HLTST are set to 0.

If 11 consecutive recessive bits are detected after entering CAN operation mode, the CAN module is in the following states:

- The CAN module becomes an active node on the network, thus enabling transmission and reception of CAN messages.
- Error monitoring of the CAN bus, such as receive and transmit error counters, is performed.

During CAN operation mode, the CAN module may be in one of the following three sub-modes, depending on the status of the CAN bus.

- Idle mode: Transmission or reception is not being performed.
- Receive mode: A CAN message sent by another node is being received.
- Transmit mode: A CAN message is being transmitted. The CAN module receives a message transmitted by the local node simultaneously when self-test mode 0 (TSTM[1:0] bits in TCR = 10b) or self-test mode 1 (TSTM[1:0] bits = 11b) is selected.

Figure 34.10 shows the sub-modes of CAN operation mode.

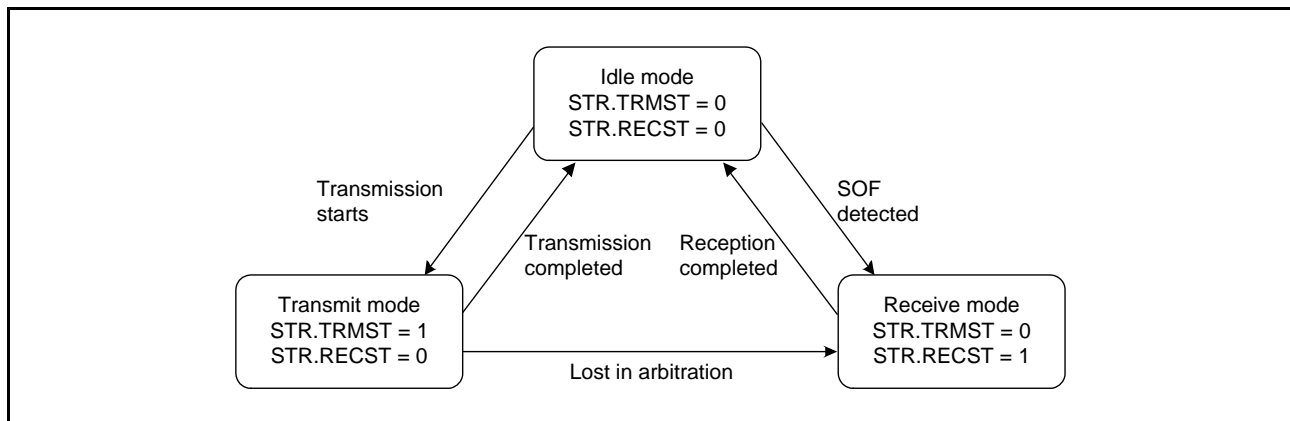


Figure 34.10 Sub-Modes of CAN Operation Mode

### 34.3.5 CAN Operation Mode (Bus-Off State)

The CAN module enters the bus-off state according to the increment/decrement rules for the transmit/error counters in the CAN Specifications.

The following cases apply when the CAN module is recovering from the bus-off state. When the CAN module is in the bus-off state, the values of the CAN-related registers, except for STR, EIFR, RECR, TECR and TSR, remain unchanged.

(1) When bits BOM[1:0] in CTLR are 00b (normal mode)

The CAN module enters the error-active state after it has completed the recovery from the bus-off state and CAN communication is enabled. The BORIF flag in EIFR is set to 1 (bus-off recovery detected) at this time.

(2) When bit RBOC in CTLR is set to 1 (forcible return from bus-off)

The CAN module enters the error-active state when it is in the bus-off state and the RBOC bit is set to 1. CAN communication is enabled again after 11b consecutive recessive bits are detected. The BORIF bit is not set to 1 at this time.

(3) When bits BOM[1:0] are 01b (automatic transition to CAN halt mode at bus-off entry)

The CAN module enters CAN halt mode when it reaches the bus-off state. The BORIF bit is not set to 1 at this time.

(4) When bits BOM[1:0] are 10b (automatic transition to CAN halt mode at bus-off end)

The CAN module enters CAN halt mode when it has completed the recovery from bus-off. The BORIF bit is set to 1 at this time.

(5) When bits BOM[1:0] are 11b (automatic transition to CAN halt mode by a program) and bits CANM[1:0] in CTLR are set to 10b (CAN halt mode) during bus-off state

The CAN module enters CAN halt mode when it is in the bus-off state and the CANM[1:0] bits are set to 10b (CAN halt mode). The BORIF bit is not set to 1 at this time.

If the CANM[1:0] bits are not set to 10b during bus-off, the same behavior as (1) applies.

### 34.4 CAN Communication Speed Setting

The following description explains about CAN communication speed setting.

#### 34.4.1 CAN Clock Setting

The CAN module has a CAN clock selector.

The CAN clock can be set by the CCLKS bit and the BRP[9:0] bits in BCR.

Figure 34.11 shows a block diagram of the CAN clock generator.

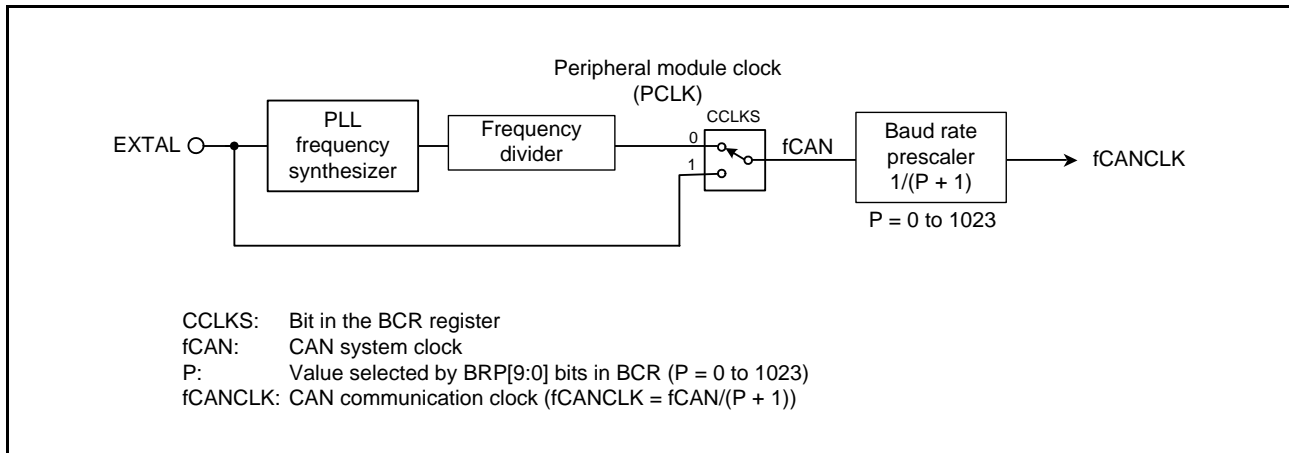


Figure 34.11 Block Diagram of CAN Clock Generator

#### 34.4.2 Bit Timing Setting

The bit time consists of the following three segments.

Figure 34.12 shows the bit timing.

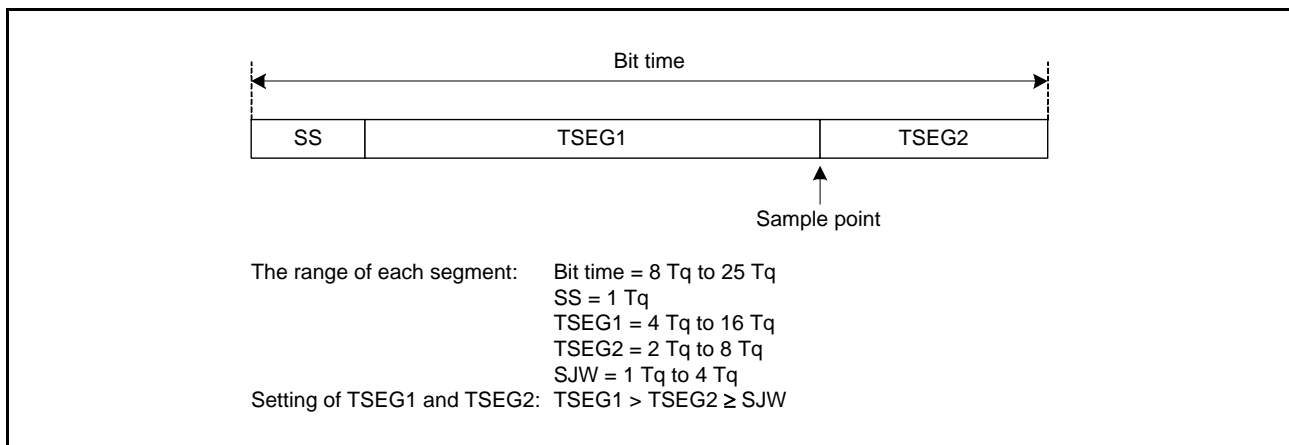


Figure 34.12 Bit Timing

### 34.4.3 Bit Rate

The bit rate depends on the division value of fCAN (CAN clock), the division value of the baud rate prescaler, and the number of Tq of 1 bit time.

$$\text{Bit rate [bps]} = \frac{\text{fCAN}}{\text{Baud rate prescaler division value}^*1 \times \text{number of Tq of 1 bit time}} = \frac{\text{fCANCLK}}{\text{Number of Tq of 1 bit time}}$$

Note 1. Division value of baud rate prescaler = P + 1 (P: 0 to 1023)  
P: Setting of the BRP[9:0] bits in BCR

Table 34.9 lists bit rate examples.

**Table 34.9 Bit Rate Examples**

fCAN	50 MHz		48 MHz		40 MHz		32 MHz	
	Number of Tq	P + 1	Number of Tq	P + 1	Number of Tq	P + 1	Number of Tq	P + 1
1 Mbps	10Tq	5	8Tq	6	10Tq	4	8Tq	4
	25Tq	2	12Tq	4	20Tq	2	16Tq	2
			16Tq	3				
500 kbps	10Tq	10	8Tq	12	10Tq	8	8Tq	8
	25Tq	4	12Tq	8	20Tq	4	16Tq	4
			16Tq	6				
250 kbps	10Tq	20	8Tq	24	10Tq	16	8Tq	16
	25Tq	8	12Tq	16	20Tq	8	16Tq	8
			16Tq	12				
125 kbps	10Tq	40	8Tq	48	10Tq	32	8Tq	32
	25Tq	16	12Tq	32	20Tq	16	16Tq	16
			16Tq	24				
83.3 kbps	10Tq	60	8Tq	72	8Tq	60	8Tq	48
	25Tq	24	12Tq	48	10Tq	48	16Tq	24
			16Tq	36	16Tq	30		
					20Tq	24		
33.3 kbps	10Tq	150	8Tq	180	8Tq	150	8Tq	120
	25Tq	60	12Tq	120	10Tq	120	10Tq	96
			16Tq	90	20Tq	60	16Tq	60
							20Tq	48

### 34.5 Mailbox and Mask Register Structure

Figure 34.13 shows the structure of MBj.

There are 32 mailboxes with the same structure.

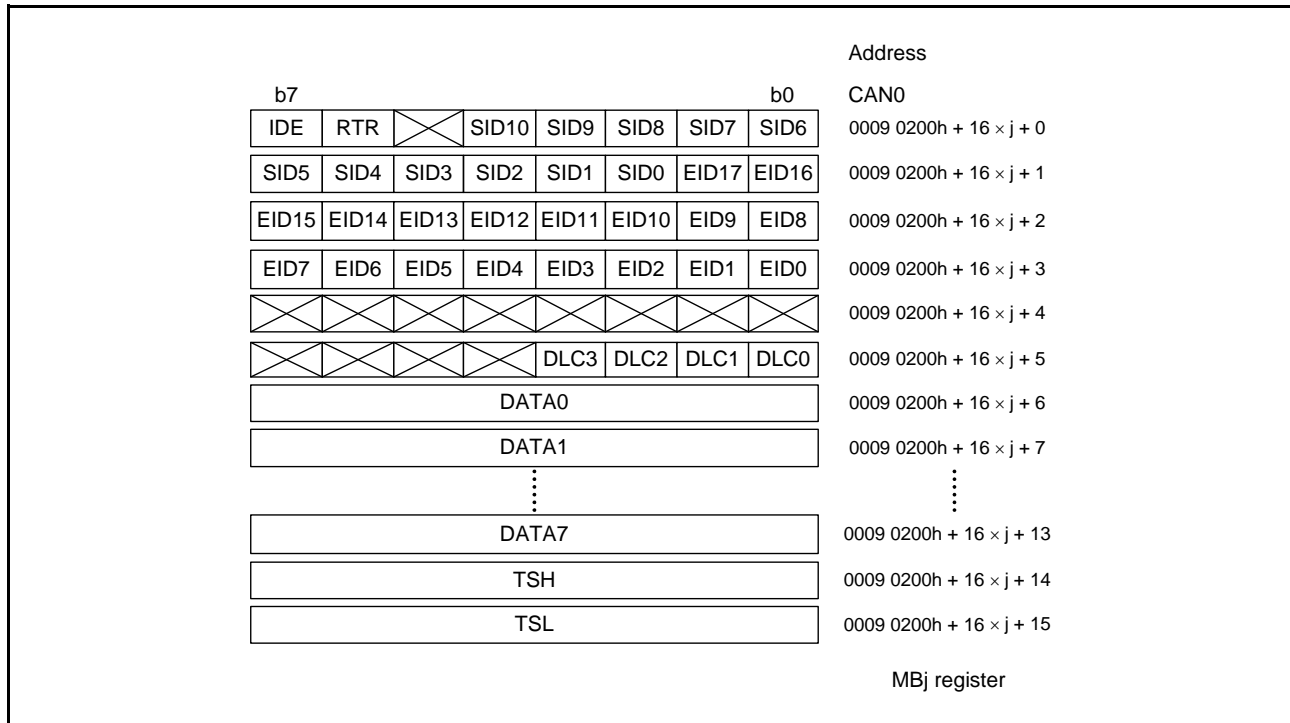


Figure 34.13 Structure of MBj (j = 0 to 31)

Figure 34.14 shows the structure of MKRk.

There are eight mask registers with the same structure.

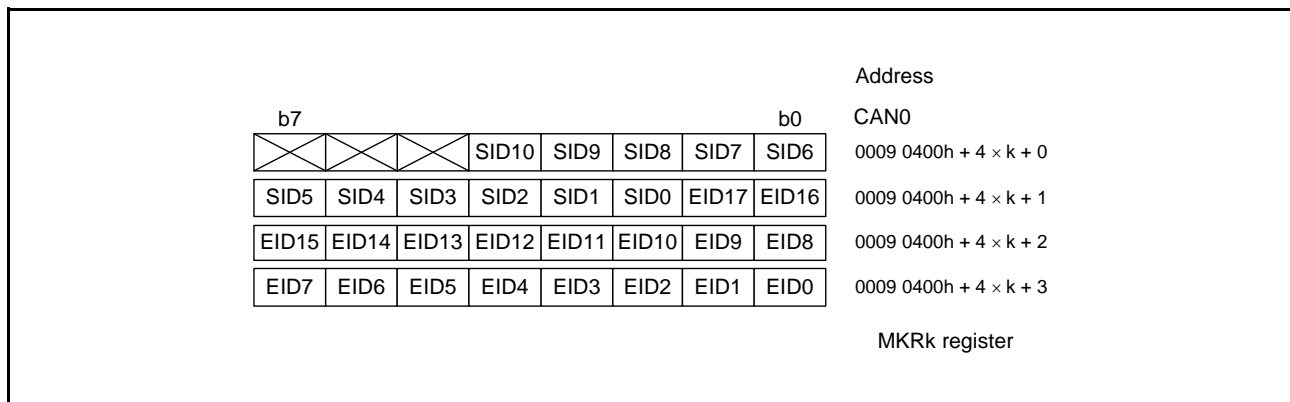


Figure 34.14 Structure of MKRk (k = 0 to 7)

Figure 34.15 shows the structure of FIDCR0 and FIDCR1.

There are two FIFO received ID compare registers with the same structure.

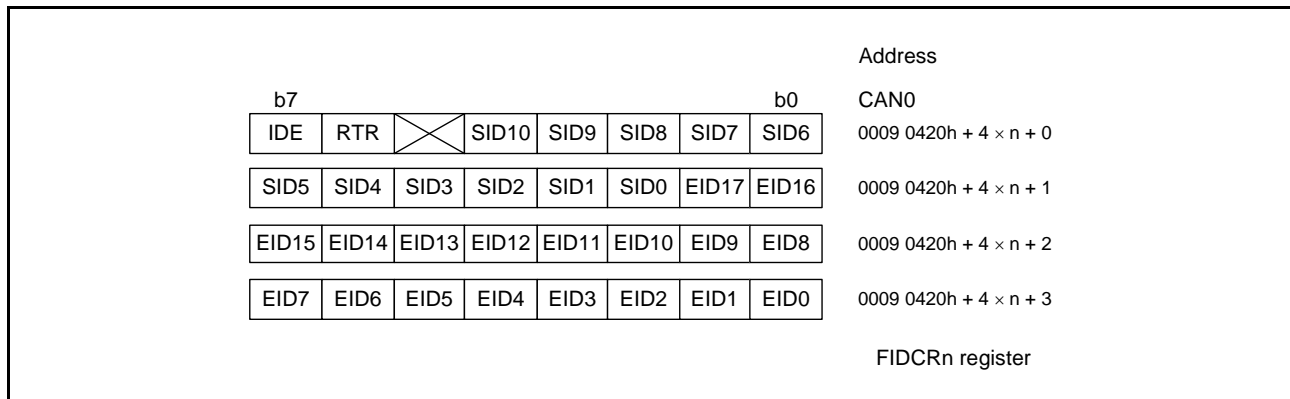


Figure 34.15 Structure of FIDCRn (n = 0, 1)

## 34.6 Acceptance Filtering and Masking Functions

The acceptance filtering function and masking function allows the user to select and receive messages with a specified range of multiple IDs for mailboxes.

Registers MKR0 to MKR7 can perform masking of the standard ID and the extended ID of 29 bits.

- MKR0 corresponds to mailboxes [0] to [3]
- MKR1 corresponds to mailboxes [4] to [7]
- MKR2 corresponds to mailboxes [8] to [11]
- MKR3 corresponds to mailboxes [12] to [15]
- MKR4 corresponds to mailboxes [16] to [19]
- MKR5 corresponds to mailboxes [20] to [23]
- MKR6 corresponds to mailboxes [24] to [27] in normal mailbox mode and the receive FIFO mailboxes [28] to [31] in FIFO mailbox mode.
- MKR7 corresponds to mailboxes [28] to [31] in normal mailbox mode and the receive FIFO mailboxes [28] to [31] in FIFO mailbox mode.

MKIVLR disables acceptance filtering individually for each mailbox.

The IDE bit in MBj is valid when the IDFM[1:0] bits in CTLR are 10b (mixed ID mode).

The RTR bit in MBj selects a data frame or a remote frame.

In FIFO mailbox mode, normal mailboxes (mailboxes [0] to [23]) use the single corresponding register among MKR0 to MKR5 for acceptance filtering. Receive FIFO mailboxes (mailboxes [28] to [31]) use two registers MKR6 and MKR7 for acceptance filtering.

Also, the receive FIFO uses two registers FIDCR0 and FIDCR1 for ID comparison. Bits EID[17:0], SID[10:0], RTR, and IDE in MB28 to MB31 for the receive FIFO are disabled. As acceptance filtering depends on the result of two logic AND operations, two ranges of IDs can be received into the receive FIFO.

MKIVLR is disabled for the receive FIFO.

If both the standard ID and extended ID are set in the IDE bits in FIDCR0 and FIDCR1 individually, both ID formats are received.

If both the data frame and remote frame are set in the RTR bits in FIDCR0 and FIDCR1 individually, both data and remote frames are received.

When combination with two ranges of IDs is not necessary, set the same mask value and the same ID into both the FIFO ID and mask register.

Figure 34.16 shows the correspondence between mask registers and mailboxes. Figure 34.17 shows acceptance filtering.

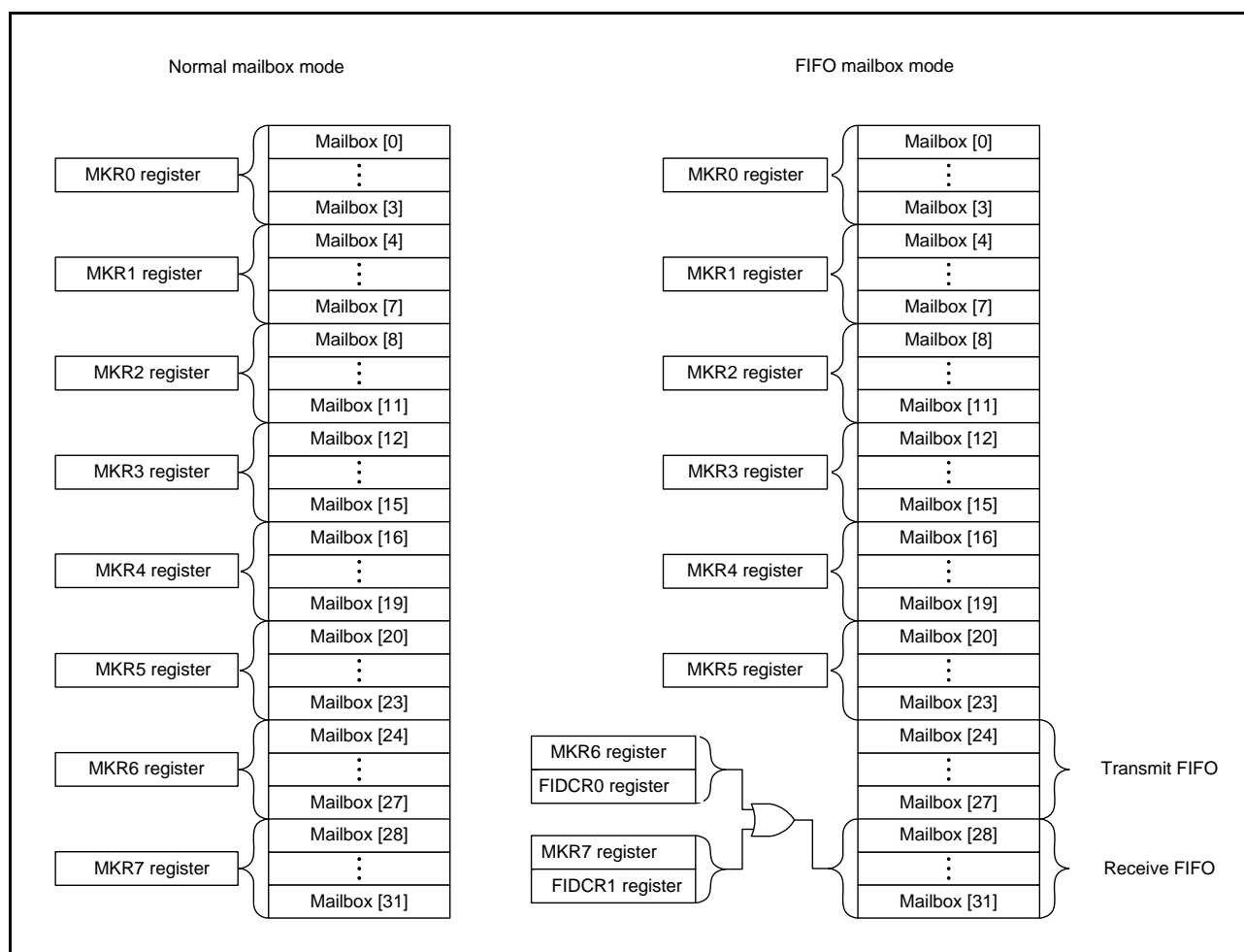


Figure 34.16 Correspondence between Mask Registers and Mailboxes

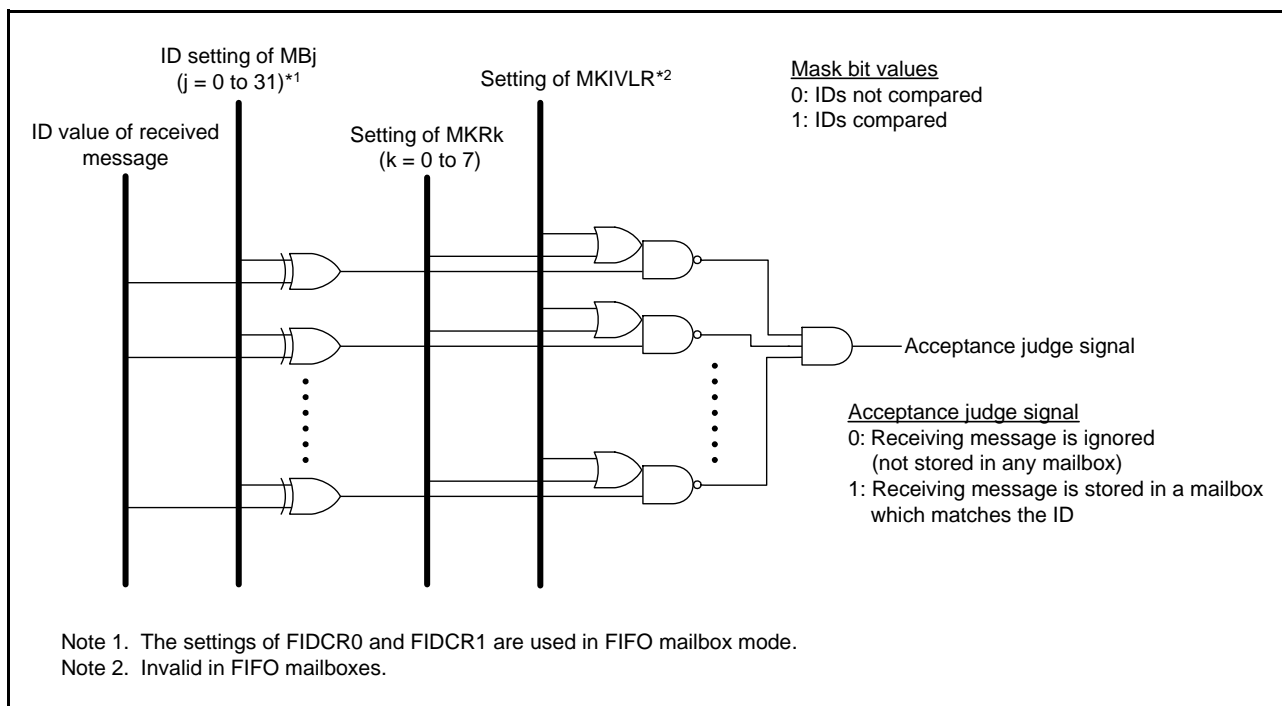


Figure 34.17 Acceptance Filtering

### 34.7 Reception and Transmission

Table 34.10 lists how to make the CAN communication mode settings.

Table 34.10 Setting of CAN Receive Mode and CAN Transmit Mode

MCTL <sub>j</sub> . TRMREQ	MCTL <sub>j</sub> . RECREQ	MCTL <sub>j</sub> . ONESHOT	Communication Mode of Mailbox
0	0	0	Mailbox disabled or transmission being aborted.
0	0	1	Can be configured only when transmission or reception from a mailbox programmed in one-shot mode is aborted.
0	1	0	Configured as a receive mailbox for a data frame or a remote frame.
0	1	1	Configured as a one-shot receive mailbox for a data frame or a remote frame.
1	0	0	Configured as a transmit mailbox for a data frame or a remote frame.
1	0	1	Configured as a one-shot transmit mailbox for a data frame or a remote frame.
1	1	0	Do not set.
1	1	1	Do not set.

j = 0 to 31

When a mailbox is configured as a receive mailbox or a one-shot receive mailbox, note the following:

1. Before a mailbox is configured as a receive mailbox or a one-shot receive mailbox, set MCTL<sub>j</sub> to 00h.
2. A received message is stored into the first mailbox that matches the condition according to the result of receive mode setting and acceptance filtering. Upon deciding the mailbox to store the received message, the mailbox with the smaller number has higher priority.
3. In CAN operation mode, when the CAN module transmits a message whose ID matches with the ID/mask set of a mailbox configured to receive messages, the CAN module never receives the transmitted data. In self-test mode, however, the CAN module will receive its transmitted data. In this case, the CAN module returns ACK.



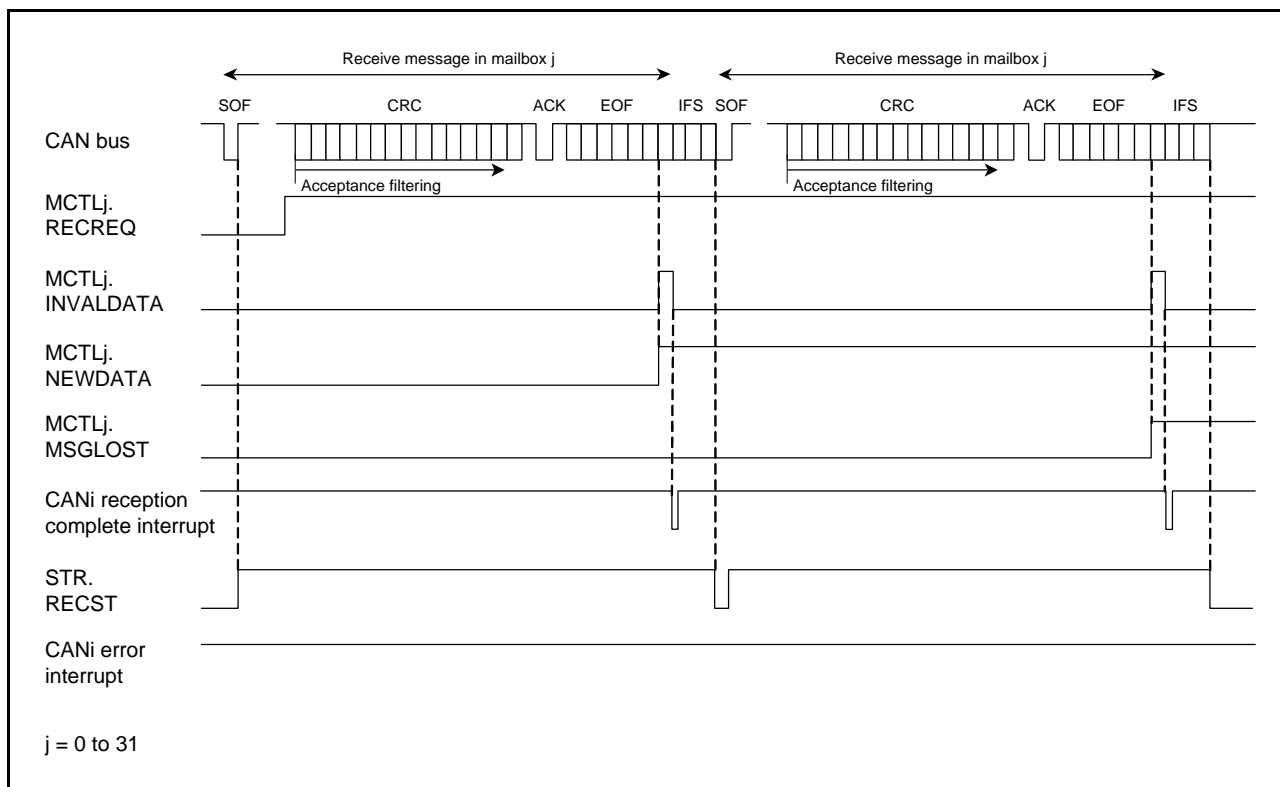
When configuring a mailbox as a transmit mailbox or a one-shot transmit mailbox, note the following:

- Before a mailbox is configured as a transmit mailbox or a one-shot transmit mailbox, ensure that MCTLj is 00h and that there is no pending abort process.

### 34.7.1 Reception

Figure 34.18 shows an operation example of data frame reception in overwrite mode.

This example shows the operation of overwriting the first message when the CAN module receives two consecutive CAN messages which match the receiving conditions of MCTLj ( $j = 0$  to 31).

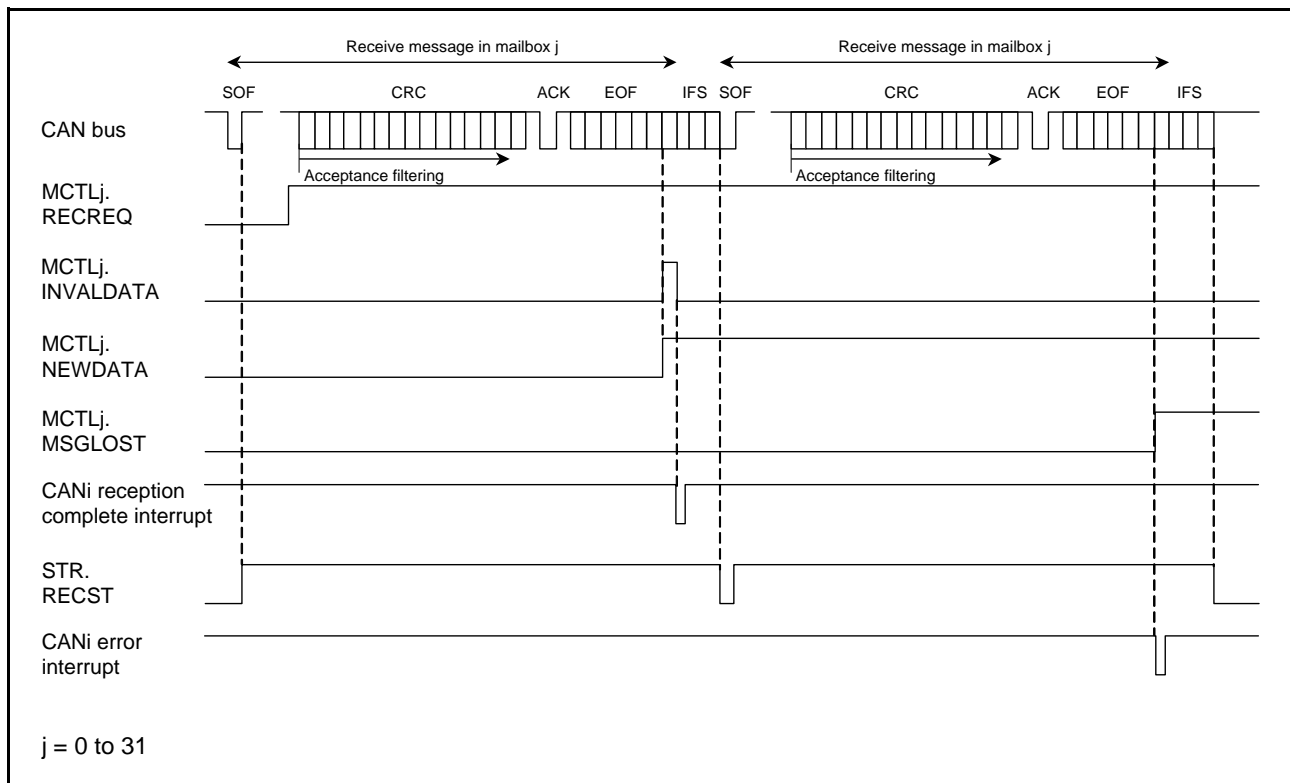


**Figure 34.18 Operation Example of Data Frame Reception in Overwrite Mode**

- When an SOF is detected on the CAN bus, the RECST bit in STR is set to 1 (reception in progress) if the CAN module has no message ready to start transmission.
- The acceptance filter processing starts at the beginning of the CRC field to select the receive mailbox.
- After a message has been received, the NEWDATA bit in MCTLj for the receive mailbox is set to 1 (new message is being stored or has been stored to the mailbox). The INVALIDDATA bit in MCTLj is set to 1 (message is being updated) at the same time, and then the INVALIDDATA bit is set to 0 (message valid) again after the complete message is transferred to the mailbox.
- When the interrupt enable bit in MIER for the receive mailbox is 1 (interrupt enabled), the CANi reception complete interrupt request is generated. This interrupt (CANi reception complete interrupt) is generated when the INVALIDDATA bit is set to 0.
- After reading the message from the mailbox, the NEWDATA bit needs to be set to 0 by a program.
- In overwrite mode, if the next CAN message has been received into a mailbox whose NEWDATA bit is still set to 1, the MSGLOST bit in MCTLj is set to 1 (message has been overwritten). The new received message is transferred to the mailbox. The CANi reception complete interrupt request is generated the same as in 4.

Figure 34.19 shows the operation example of data frame reception in overrun mode.

This example shows the operation of overrunning the second message when the CAN module receives two consecutive CAN messages which match the receiving conditions of MCTLj ( $j = 0$  to 31).



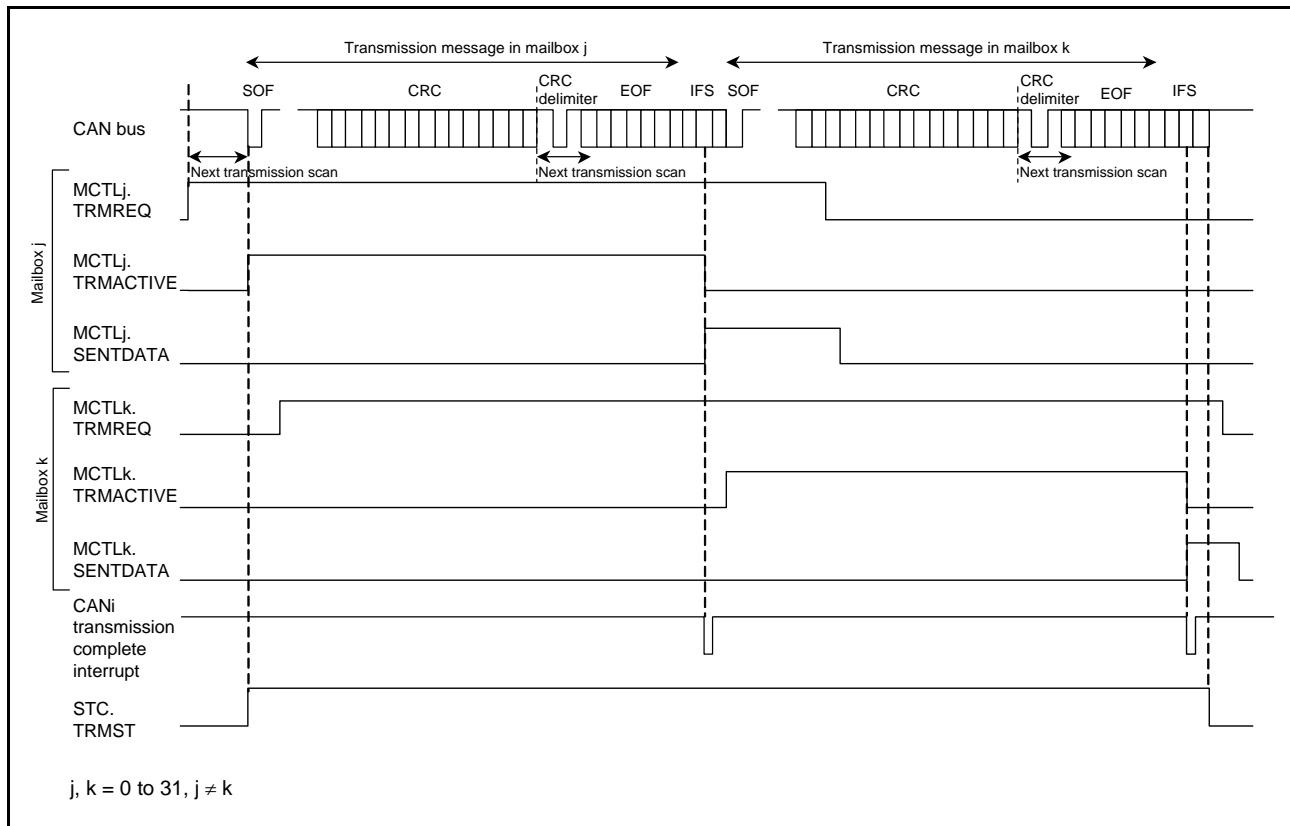
**Figure 34.19 Operation Example of Data Frame Reception in Overrun Mode**

1. to 5. are the same as in overwrite mode.

6. In overrun mode, if the next CAN message has been received before the NEWDATA bit in MCTLj is set to 0, the MSGLOST bit in MCTLj is set to 1 (message has been overrun). The new received message is discarded and a CANi error interrupt request is generated if the corresponding interrupt enable bit in EIER is set to 1 (interrupt enabled).

### 34.7.2 Transmission

Figure 34.20 shows an operation example of data frame transmission.



**Figure 34.20** Operation Example of Data Frame Transmission

1. When a TRMREQ bit in MCTLj ( $j = 0$  to  $31$ ) is set to 1 (transmit mailbox) in the bus-idle state, the mailbox scan processing starts to decide the highest-priority mailbox for transmission. Once the transmit mailbox is decided, the TRMACTIVE bit in MCTLj is set to 1 (from acceptance of transmission request to completion of transmission, or error/arbitration-lost), the TRMST bit in STR is set to 1 (transmission in progress), and the CAN module starts transmission. \*Note: •
2. If other TRMREQ bits are set, the transmission scan processing starts with the CRC delimiter for the next transmission.
3. If transmission is completed without losing arbitration, the SENDDATA bit in MCTLj is set to 1 (transmission completed) and the TRMACTIVE bit is set to 0 (transmission is pending or transmission is not requested). If the interrupt enable bit in MIER is 1 (interrupt enabled), the CANi transmission complete interrupt request is generated.
4. When requesting the next transmission from the same mailbox, set bits SENDTDATA and TRMREQ to 0, then set the TRMREQ bit to 1 after checking that bits SENDTDATA and TRMREQ have been set to 0.

Note: • If arbitration is lost after the CAN module starts transmission, the TRMACTIVE bit is set to 0. The transmission scan processing is performed again to search for the highest-priority transmit mailbox from the beginning of the CRC delimiter. If an error occurs either during transmission or following the arbitration-lost, the transmission scan processing is performed again to search for the highest-priority transmit mailbox from the start of the error delimiter.

## 34.8 CAN Interrupt

The CAN module provides the following CAN interrupts for each channel. Table 34.11 lists CAN interrupts.

- CANi reception complete interrupt (mailboxes 0 to 31) [RXMi]
- CANi transmission complete interrupt (mailboxes 0 to 31) [TXMi]
- CANi receive FIFO interrupt [RXFi]
- CANi transmit FIFO interrupt [TXFi]
- CANi error interrupt [ERSi]

There are eight types of interrupt sources for the CANi error interrupts. These sources can be determined by checking EIFR.

- Bus error
- Error-warning
- Error-passive
- Bus-off entry
- Bus-off recovery
- Receive overrun
- Overload frame transmission
- Bus lock

**Table 34.11 CAN Interrupts**

Module	Interrupt Symbol	Interrupt Source	Source Flag
CANi	ERSi	Bus lock detected	EIFR.BLIF
		Overload frame transmission detected	EIFR.OLIF
		Overrun detected	EIFR.ORIF
		Bus-off recovery detected	EIFR.BORIF
		Bus-off entry detected	EIFR.BOEIF
		Error-passive detected	EIFR.EPIF
		Error-warning detected	EIFR.EWIF
		Bus error detected	EIFR.BEIF
RXFi	RXFi	Receive FIFO message received (MIER[29] = 0)	RFCR.RFUST[2:0]
		Receive FIFO warning (MIER[29] = 1)	
TXFi	TXFi	Transmit FIFO message transmission completed (MIER[25] = 0)	TFCR.TFUST[2:0]
		FIFO last message transmission completed (MIER[25] = 1)	
RXMi		Mailbox 0 to 31 message received	MCTL0.NEWDATA to MCTL31.NEWDATA
TXMi		Mailbox 0 to 31 message transmission completed	MCTL0.SENTDATA to MCTL31.SENTDATA

i = 0 to 2

## 34.9 Usage Notes

### 34.9.1 Setting for the Module-Stop State

Module stop control register B (MSTPCRB) can be used to enable or disable operation of the CAN module. The CAN module is stopped after a reset. The registers become accessible on release from the module-stop state. For details, refer to section 11, Low Power Consumption.

## 35. Serial Peripheral Interface (RSPI)

### 35.1 Overview

The RX630 Group includes three independent channels of Serial Peripheral Interface (RSPI).

The RSPI channels are capable of high-speed, full-duplex synchronous serial communications with multiple processors and peripheral devices.

Table 35.1 lists the specifications of the RSPI, and Figure 35.1 shows a block diagram of the RSPI.

Furthermore, n in this section indicates A, B, or C, and i indicates 0 to 3. Also, m as used with the RSPI command registers (SPCMDm) indicates 0 to 7.

**Table 35.1 Specifications of RSPI**

Item	Description
Number of channels	Three channels
RSPI transfer functions	<ul style="list-style-type: none"> <li>Use of MOSI (master out/slave in), MISO (master in/slave out), SSL (slave select), and RSPCK (RSPI clock) signals allows serial communications through SPI operation (four-wire method) or clock synchronous operation (three-wire method).</li> <li>Transmit-only operation is available.</li> <li>Capable of serial communications in master/slave mode</li> <li>Switching of the polarity of the serial transfer clock</li> <li>Switching of the phase of the serial transfer clock</li> </ul>
Data format	<ul style="list-style-type: none"> <li>MSB-first/LSB-first selectable</li> <li>Transfer bit length is selectable as 8, 9, 10, 11, 12, 13, 14, 15, 16, 20, 24, or 32 bits.</li> <li>128-bit transmit/receive buffers</li> <li>Up to four frames can be transferred in one round of transmission/reception (each frame consisting of up to 32 bits).</li> </ul>
Bit rate	<ul style="list-style-type: none"> <li>In master mode, the on-chip baud rate generator generates RSPCK by frequency-dividing PCLK (the divisor is 2 to 4096).</li> <li>In slave mode, the externally input clock is used as the serial clock (the maximum frequency is that of PCLK divided by 8). Width at high level: 4 cycles of PCLK; width at low level: 4 cycles of PCLK</li> </ul>
Buffer configuration	Double buffer configuration for the transmit/receive buffers
Error detection	<ul style="list-style-type: none"> <li>Mode fault error detection</li> <li>Overrun error detection</li> <li>Parity error detection</li> </ul>
SSL control function	<ul style="list-style-type: none"> <li>Four SSL signals (SSLn0 to SSLn3) for each channel</li> <li>In single-master mode, SSLn0 to SSLn3 signals are output.</li> <li>In multi-master mode: SSLn0 signal for input, and SSLn1 to SSLn3 signals for either output or unused.</li> <li>In slave mode: SSLn0 signal for input, and SSLn1 to SSLn3 signals for unused.</li> <li>Controllable delay from SSL output assertion to RSPCK operation (RSPCK delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)</li> <li>Controllable delay from RSPCK stop to SSL output negation (SSL negation delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)</li> <li>Controllable wait for next-access SSL output assertion (next-access delay) Range: 1 to 8 RSPCK cycles (set in RSPCK-cycle units)</li> <li>Function for changing SSL polarity</li> </ul>
Control in master transfer	<ul style="list-style-type: none"> <li>A transfer of up to eight commands can be executed sequentially in looped execution.</li> <li>For each command, the following can be set: SSL signal value, bit rate, RSPCK polarity/phase, transfer data length, MSB/LSB-first, burst, RSPCK delay, SSL negation delay, and next-access delay</li> <li>A transfer can be initiated by writing to the transmit buffer.</li> <li>MOSI signal value specifiable in SSL negation</li> </ul>
Interrupt sources	<ul style="list-style-type: none"> <li>Maskable interrupt sources RSPI receive interrupt (receive buffer full) RSPI transmit interrupt (transmit buffer empty) RSPI error interrupt (mode fault, overrun, parity error) RSPI idle interrupt (RSPI idle)</li> </ul>

**Table 35.1 Specifications of RSPI**

Item	Description
Others	<ul style="list-style-type: none"><li>• Function for initializing the RSPI</li><li>• Loopback mode</li></ul>
Power consumption reducing function	Module-stop state can be set.

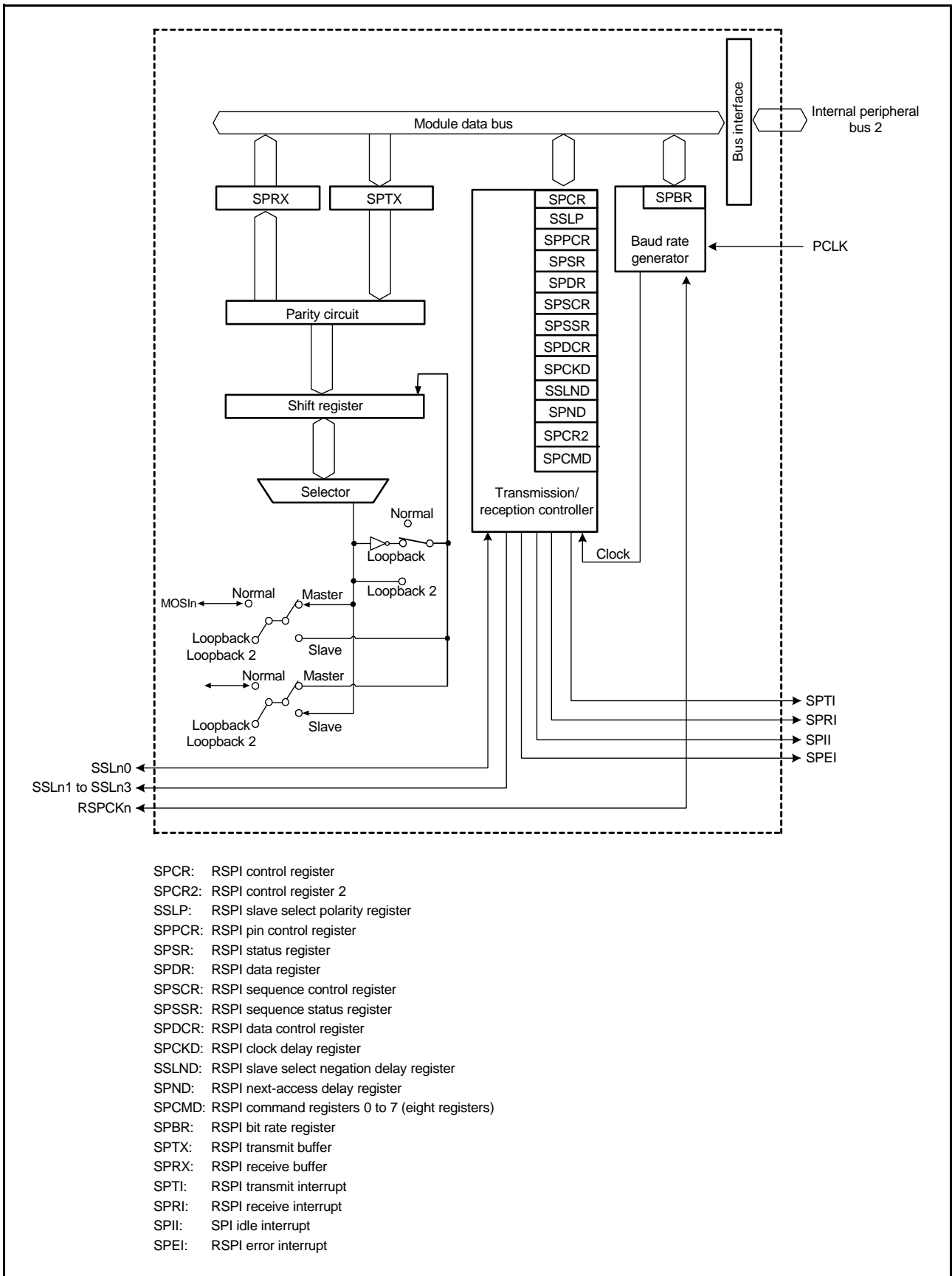


Figure 35.1 Block Diagram of RSPI

Table 35.2 lists the input and output pins used in the RSPI.

The RSPI automatically switches the input/output direction of the SSLn0 pin (n = A, B, or C). SSLn0 is set as an output when the RSPI is a single master and as an input when the RSPI is a multi-master or a slave. Pins RSPCKn, MOSIn, and MISOn (n = A, B, or C) are automatically set as inputs or outputs according to the setting of master or slave and the level input on the SSLn0 pin (see section 35.3.2, Controlling RSPI Pins).

**Table 35.2 RSPI Pin Configuration**

Channel	Pin Name	I/O	Function
RSPI0	RSPCKA	I/O	Clock I/O pin
	MOSIA	I/O	Master transmit data I/O pin
	MISOA	I/O	Slave transmit data I/O pin
	SSLA0	I/O	Slave selection I/O pin
	SSLA1	Output	Slave selection output pin
	SSLA2	Output	Slave selection output pin
	SSLA3	Output	Slave selection output pin
RSPI1	RSPCKB	I/O	Clock I/O pin
	MOSIB	I/O	Master transmit data I/O pin
	MISOB	I/O	Slave transmit data I/O pin
	SSLB0	I/O	Slave selection I/O pin
	SSLB1	Output	Slave selection output pin
	SSLB2	Output	Slave selection output pin
	SSLB3	Output	Slave selection output pin
RSPI2	RSPCKC	I/O	Clock I/O pin
	MOSIC	I/O	Master transmit data I/O pin
	MISOC	I/O	Slave transmit data I/O pin
	SSLC0	I/O	Slave selection I/O pin
	SSLC1	Output	Slave selection output pin
	SSLC2	Output	Slave selection output pin
	SSLC3	Output	Slave selection output pin



## 35.2 Register Descriptions

### 35.2.1 RSPI Control Register (SPCR)

Address(es): RSPI0.SPCR 0008 8380h, RSPI1.SPCR 0008 83A0h, RSPI2.SPCR 0008 83C0h

b7	b6	b5	b4	b3	b2	b1	b0
SPRIE	SPE	SPTIE	SPEIE	MSTR	MODFEN	TXMD	SPMS
Value after reset: 0 0 0 0 0 0 0 0							

Bit	Symbol	Bit Name	Description	R/W
b0	SPMS	RSPI Mode Select	0: SPI operation (four-wire method) 1: Clock synchronous operation (three-wire method)	R/W
b1	TXMD	Communications Operating Mode Select	0: Full-duplex synchronous serial communications 1: Serial communications consisting of only transmit operations	R/W
b2	MODFEN	Mode Fault Error Detection Enable	0: Disables the detection of mode fault error 1: Enables the detection of mode fault error	R/W
b3	MSTR	RSPI Master/Slave Mode Select	0: Slave mode 1: Master mode	R/W
b4	SPEIE	RSPI Error Interrupt Enable	0: Disables the generation of RSPI error interrupt requests 1: Enables the generation of RSPI error interrupt requests	R/W
b5	SPTIE	RSPI Transmit Interrupt Enable	0: Disables the generation of RSPI transmit interrupt requests 1: Enables the generation of RSPI transmit interrupt requests	R/W
b6	SPE	RSPI Function Enable	0: Disables the RSPI function 1: Enables the RSPI function	R/W
b7	SPRIE	RSPI Receive Interrupt Enable	0: Disables the generation of RSPI receive interrupt requests 1: Enables the generation of RSPI receive interrupt requests	R/W

If the SPCR.MSTR, SPCR.MODFEN, and SPCR.TXMD bits are changed while the SPCR.SPE bit is 1, subsequent operations cannot be guaranteed.

#### SPMS Bit (RSPI Mode Select)

The SPMS bit selects SPI operation (four-wire method) or clock synchronous operation (three-wire method). The SSLn0 to SSLn3 pins are not used in clock synchronous operation. The three pins RSPCKn, MOSIn, and MISOIn handle communications. If clock-synchronous operation is to proceed in master mode (SPCR.MSTR = 1), the SPCMDm.CPHA bit can be set to either 0 or 1. Set the CPHA bit to 1 if clock-synchronous operation is to proceed in slave mode (SPCR.MSTR = 0). Operation is not guaranteed if the CPHA bit is set to 0 when clock-synchronous operation is to proceed in slave mode (SPCR.MSTR = 0).

#### TXMD Bit (Communications Operating Mode Select)

The TXMD bit selects full-duplex synchronous serial communications or transmit operations only. When performing communications with the TXMD bit set to 1, the RSPI performs only transmit operations and not receive operations (see section 35.3.6, Communications Operating Mode). When the TXMD bit is set to 1, receive buffer full interrupt requests cannot be used.

#### MODFEN Bit (Mode Fault Error Detection Enable)

The MODFEN bit enables or disables the detection of mode fault error (see section 35.3.8, Error Detection). In addition, the RSPI determines the input/output direction of the SSLn0 to SSLn3 pins based on combinations of the MODFEN and MSTR bits (see section 35.3.2, Controlling RSPI Pins).

**MSTR Bit (RSPI Master/Slave Mode Select)**

The MSTR bit selects master/slave mode of the RSPI. According to MSTR bit settings, the RSPI determines the direction of pins RSPCKn, MOSIn, MISO<sub>n</sub>, and SSL<sub>n</sub>0 to SSL<sub>n</sub>3.

**SPEIE Bit (RSPI Error Interrupt Enable)**

The SPEIE bit enables or disables the generation of RSPI error interrupt requests when the RSPI detects a mode fault error and sets the SPSR.MODF flag to 1, when the RSPI detects an overrun error and sets the SPSR.OVRF flag to 1, or when the RSPI detects a parity error and sets the SPSR.PERF flag to 1 (see section 35.3.8, Error Detection).

**SPTIE Bit (RSPI Transmit Interrupt Enable)**

The SPTIE bit enables or disables the generation of RSPI transmit interrupt requests when the RSPI detects transmit buffer empty.

At the beginning of transmission, the transmit interrupt requests are generated by setting the SPE bit to 1 at the same time or after the SPTIE bit has been set to 1.

Therefore, note that even while the RSPI function is disabled (SPE bit is 0), setting the SPTIE bit to 1 will generate an RSPI transmit interrupt request.

**SPE Bit (RSPI Function Enable)**

The SPE bit enables or disables the RSPI function.

When the SPSR.MODF bit is 1, the SPE bit cannot be set to 1. For details, refer to section 35.3.8, Error Detection. Setting the SPE bit to 0 disables the RSPI function, and initializes a part of the module function. For details, refer to section 35.3.9, Initializing RSPI. Furthermore, an RSPI transmission interrupt request is generated by the state of the SPE bit changing from 0 to 1 or from 1 to 0.

**SPRIE Bit (RSPI Receive Interrupt Enable)**

If the RSPI has detected a receive buffer write after completion of a serial transfer, the SPRIE bit enables or disables the generation of an RSPI receive interrupt request.

**35.2.2 RSPI Slave Select Polarity Register (SSLP)**

Address(es): RSPI0.SSLP 0008 8381h, RSPI1.SSLP 0008 83A1h, RSPI2.SSLP 0008 83C1h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	SSL3P	SSL2P	SSL1P	SSL0P

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	SSL0P	SSL0 Signal Polarity Setting	0: SSL0 signal is active low 1: SSL0 signal is active high	R/W
b1	SSL1P	SSL1 Signal Polarity Setting	0: SSL1 signal is active low 1: SSL1 signal is active high	R/W
b2	SSL2P	SSL2 Signal Polarity Setting	0: SSL2 signal is active low 1: SSL2 signal is active high	R/W
b3	SSL3P	SSL3 Signal Polarity Setting	0: SSL3 signal is active low 1: SSL3 signal is active high	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

If the contents of SSLP are changed while the SPCR.SPE bit is 1, subsequent operations are not guaranteed.

### 35.2.3 RSPI Pin Control Register (SPPCR)

Address(es): RSPI0.SPPCR 0008 8382h, RSPI1.SPPCR 0008 83A2h, RSPI2.SPPCR 0008 83C2h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	MOIFE	MOIFV	—	—	SPLP2	SPLP
Value after reset:	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	SPLP	RSPI Loopback	0: Normal mode 1: Loopback mode (reversed transmit data = receive data)	R/W
b1	SPLP2	RSPI Loopback 2	0: Normal mode 1: Loopback mode (transmit data = receive data)	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	MOIFV	MOSI Idle Fixed Value	0: The level output on the MOSIn pin during MOSI idling corresponds to 0. 1: The level output on the MOSIn pin during MOSI idling corresponds to 1.	R/W
b5	MOIFE	MOSI Idle Value Fixing Enable	0: MOSI output value equals final data from previous transfer 1: MOSI output value equals the value set in the MOIFV bit	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

If the contents of SPPCR are changed while the SPCR.SPE bit is 1, subsequent operations are not guaranteed.

#### SPLP Bit (RSPI Loopback)

The SPLP bit selects the mode of the RSPI pins.

When the SPLP bit is set to 1, the RSPI shuts off the path between the MISO<sub>in</sub> pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSI<sub>in</sub> pin and the shift register if the SPCR.MSTR bit is 0, and connects (reverses) the input path and output path for the shift register (loopback mode).

#### SPLP2 Bit (RSPI Loopback 2)

The SPLP2 bit selects the mode of the RSPI pins.

When the SPLP2 bit is set to 1, the RSPI shuts off the path between the MISO<sub>in</sub> pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSI<sub>in</sub> pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path for the shift register (loopback mode).

#### MOIFV Bit (MOSI Idle Fixed Value)

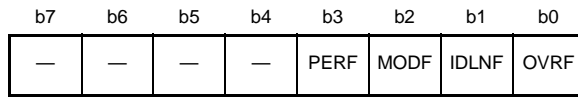
If the MOIFE bit is 1 in master mode, the MOIFV bit determines the MOSI<sub>in</sub> pin output value during the SSL negation period (including the SSL retention period during a burst transfer).

#### MOIFE Bit (MOSI Idle Value Fixing Enable)

The MOIFE bit fixes the MOSI<sub>in</sub> output value when the RSPI in master mode is in an SSL negation period (including the SSL retention period during a burst transfer). When the MOIFE bit is 0, the RSPI outputs the last data from the previous serial transfer during the SSL negation period to the MOSI<sub>in</sub> pin. When the MOIFE bit is 1, the RSPI outputs the fixed value set in the MOIFV bit to the MOSI<sub>in</sub> pin.

### 35.2.4 RSPI Status Register (SPSR)

Address(es): RSPI0.SPSR 0008 8383h, RSPI1.SPSR 0008 83A3h, RSPI2.SPSR 0008 83C3h



Value after reset: x 0 x 0 0 0 0 0

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b0	OVRF	Overrun Error Flag	0: No overrun error occurs 1: An overrun error occurs	R/(W) *1
b1	IDLNF	RSPI Idle Flag	0: RSPI is in the idle state 1: RSPI is in the transfer state	R
b2	MODF	Mode Fault Error Flag	0: No mode fault error occurs 1: A mode fault error occurs	R/(W) *1
b3	PERF	Parity Error Flag	0: No parity error occurs 1: A parity error occurs	R/(W) *1
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	—	Reserved	The read value is undefined. The write value should be 1.	R/W
b6	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7	—	Reserved	The read value is undefined. The write value should be 1.	R/W

Note 1. Only 0 can be written to clear the flag after reading 1.

#### OVRF Flag (Overrun Error Flag)

The OVRF flag indicates the occurrence of an overrun error.

[Setting condition]

- When a serial transfer ends while the SPCR.TXMD bit is 0 and the receive buffer holds data that has not yet been read

[Clearing condition]

- When SPSR is read while the OVRF flag is 1, and then writes the value 0 to the OVRF flag.

**IDLNF Flag (RSPI Idle Flag)**

The IDLNF flag indicates the transfer status of the RSPI.

[Setting condition]

<Master mode>

- When both the Conditions 1 and 2 among the following clearing conditions for master mode are not satisfied

<Slave mode>

- The SPCR.SPE bit is 1 (RSPI function is enabled)

[Clearing condition]

<Master mode>

When the following first clearing condition is satisfied (Condition 1) or all of the following second to fourth clearing conditions are satisfied (Condition 2)

1. The SPCR.SPE bit is 0 (RSPI is initialized)
2. The transmit buffer (SPTX) is empty (data for the next transfer is not set)
3. The SPSSR.SPCP[2:0] bits are 000b (beginning of sequence control)
4. The RSPI internal sequencer has entered the idle state (status in which operations up to the next-access delay have finished)

<Slave mode>

- The SPCR.SPE bit is 0 (RSPI is initialized)

**MODF Flag (Mode Fault Error Flag)**

Indicates the occurrence of a mode fault error.

[Setting condition]

<Multi-master mode>

- When the input level of the SSLni pin changes to the active level while the SPCR.MSTR bit is 1 (master mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPI detects a mode fault error

<Slave mode>

- When the SSLni pin is negated before the RSPCK cycle necessary for data transfer ends while the SPCR.MSTR bit is 0 (slave mode) and the SPCR.MODFEN bit is 1 (mode fault error detection is enabled), the RSPI detects a mode fault error

The active level of the SSLni signal is determined by the SSLP.SSLiP bit (SSL signal polarity setting bit).

[Clearing condition]

- When SPSR is read while the MODF flag is 1, and then writes the value 0 to the MODF flag

**PERF Flag (Parity Error Flag)**

Indicates the occurrence of a parity error.

[Setting condition]

- When a serial transfer ends while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1, the RSPI detects a parity error

[Clearing condition]

- When SPSR is read while the PERF flag is 1, and then writes the value 0 to the PERF flag

### 35.2.5 RSPI Data Register (SPDR)

Address(es): RSPI0.SPDR 0008 8384h, RSPI1.SPDR 0008 83A4h, RSPI2.SPDR 0008 83C4h

b31	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16
SPD31	SPD30	SPD29	SPD28	SPD27	SPD26	SPD25	SPD24	SPD23	SPD22	SPD21	SPD20	SPD19	SPD18	SPD17	SPD16
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
SPD15	SPD14	SPD13	SPD12	SPD11	SPD10	SPD9	SPD8	SPD7	SPD6	SPD5	SPD4	SPD3	SPD2	SPD1	SPD0
Value after reset: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0															

SPDR is the interface with the buffers that hold data for transmission and reception by the RSPI. The transmit buffer (SPTX) and receive buffer (SPRX) are independent but are both mapped to SPDR. Figure 35.2 shows the Configuration of SPDR.

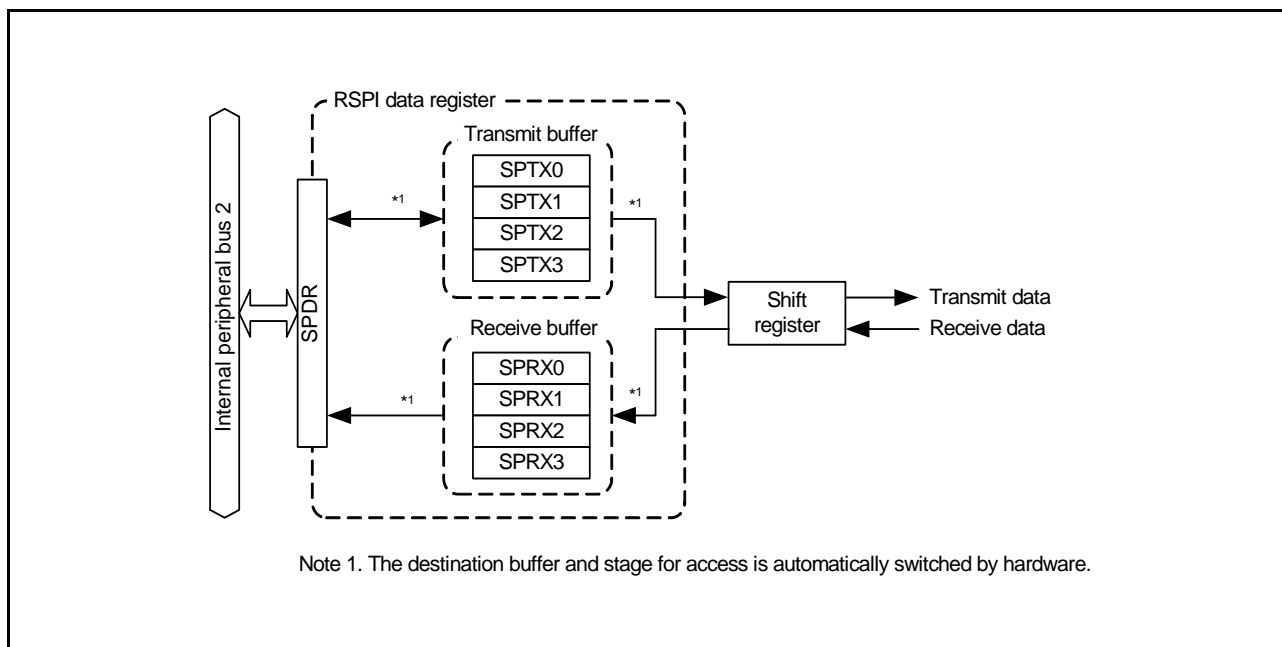


Figure 35.2 Configuration of SPDR

The transmit and receive buffers have four stages each. The number of stages to be used is selectable by the number of frames specification bits in the RSPI data control register (SPDCR.SPFC[1:0]). The eight stages of the buffer are all mapped to the single address of SPDR.

Data written to SPDR are written to a transmit-buffer stage (SPTX0 to SPTX3) and then transmitted from the buffer. The receive buffer holds received data on completion of reception. The receive buffer is not updated if an overrun is generated.

Furthermore, if the data length is other than 32 bits, bits not referred to in SPTXn (n = 0 to 3) are stored in the corresponding bits in SPRXn. For example, if the data length is nine bits, the SPTXn[31:9] bits are stored in SPRXn[31:9] (and received data are stored in the SPTXn[8:0] bits).

## (1) Bus Interface

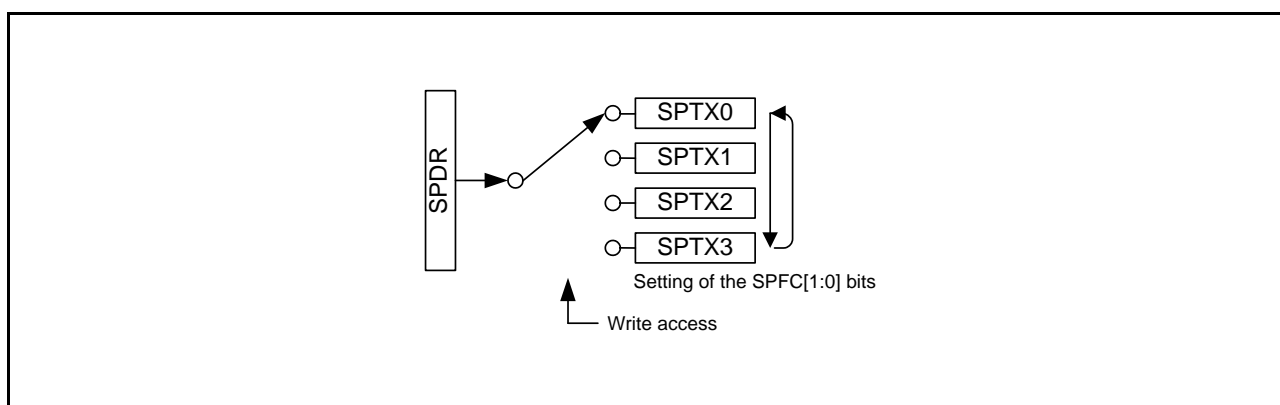
SPDR is the interface with 32-bit wide transmit and receive buffers, each of which has four stages, for a total of 32 bytes. In other words, the 32 bytes are mapped to the four-byte address space for SPDR. Furthermore, the unit of access for SPDR is selected by the RSPI longword access/word access specification bit in the RSPI data control register (SPDCR.SP1W).

Data for transmission should be flush with the LSB end of the register. Received data are stored flush with the LSB end. Operations involved in writing to and reading from SPDR are described below.

### (a) Writing

The transmit buffer includes a transmit buffer write pointer which is automatically updated to indicate the next stage each time data are written to SPDR.

Figure 35.3 shows the configuration of the bus interface with the transmit buffer in the case of writing to SPDR.



**Figure 35.3 Configuration of SPDR (Writing)**

The sequence for switching the transmit buffer write pointer differs with the setting of the number of frames specification bits in the RSPI data control register (SPDCR.SPFC[1:0]).

- Settings of the SPFC[1:0] bits and sequence of switching the pointer among SPTX0 to SPTX03.
  - When the SPFC[1:0] bits are 00b: SPTX0 → SPTX0 → SPTX0 → ...
  - When the SPFC[1:0] bits are 01b: SPTX0 → SPTX1 → SPTX0 → SPTX1 → ...
  - When the SPFC[1:0] bits are 10b: SPTX0 → SPTX1 → SPTX2 → SPTX0 → SPTX1 → ...
  - When the SPFC[1:0] bits are 11b: SPTX0 → SPTX1 → SPTX2 → SPTX3 → SPTX0 → SPTX1 → ...

When 1 is written to the RSPI function enable bit in the RSPI control register (SPCR.SPE) while the bit's current value is 0, SPTX0 will be the destination the next time writing proceeds.

When writing to the transmission buffer (SPTXn) after generation of the RSPI transmission interrupt, write the number of frames set by the number of frames specification bits (SPFC[1:0]) in the RSPI data control register (SPDCR). Writing to the transmission buffer (SPTXn) before the next RSPI transmission interrupt breaks off write access to the SPDR.

## (b) Reading

SPDR can be read to read out the value of a receive buffer (SPRXn; n = 0 to 3) or a transmit buffer (SPTXn; n = 0 to 3). The setting of the RSPI receive/transmit data selection bit in the RSPI data control register (SPDCR.SPRDTD) selects whether reading is of the receive or transmit buffer.

The structure of the SPDR when it is read includes two independent pointers (receive buffer read pointer and transmit buffer read pointer). Reading SPDR causes automatic updating of the pointer so that it indicates the next stage of the buffer.

Figure 35.4 shows the configuration of the bus interface with the receive and transmit buffers in the case of reading from SPDR.

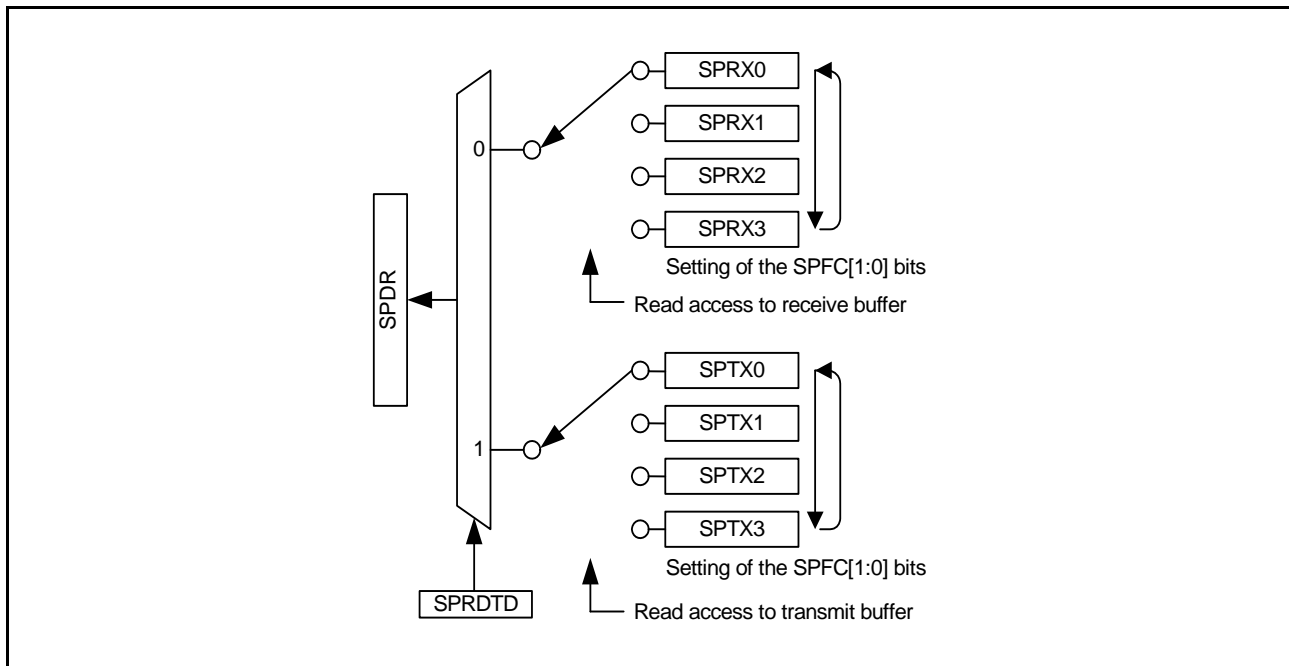


Figure 35.4 Configuration of SPDR (Reading)

The sequence for switching the receive buffer read pointer is the same as that for the transmit buffer write pointer. When reading is from the transmit buffer, the value before that most recently written is read. Furthermore, only the read pointer for the buffer that is currently selected for reading by the setting of the RSPI receive/transmit data selection bit in the RSPI data control register (SPDCR.SPRDTD) is updated, and the state of the read pointer for the other buffer is preserved.

When 1 is written to the RSPI function enable bit in the RSPI control register (SPCR.SPE) while the bit's current value is 0, SPRX0 will be indicated by the buffer read pointer the next time reading proceeds.

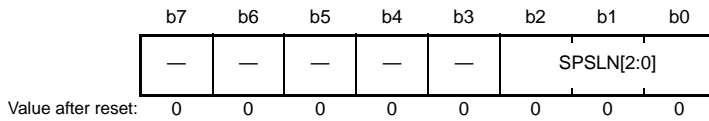
If the transmission buffer is read in the interval after writing of the number of frames of data for transmission specified in the number of frames specification bits (SPFC[1:0]) in the RSPI data control register (SPDCR) and generation of the RSPI transmission interrupt, and before the next RSPI transmission interrupt, all bits of the value read out are 0.

The SPTXn buffer-reading pointer is cleared when the number of frames of data for transmission specified in the number of frames specification bits (SPFC[1:0]) in the RSPI data control register (SPDCR) are written after generation of the RSPI transmission interrupt.



### 35.2.6 RSPI Sequence Control Register (SPSCR)

Address(es): RSPI0.SPSCR 0008 8388h, RSPI1.SPSCR 0008 83A8h, RSPI2.SPSCR 0008 83C8h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPSLN[2:0]	RSPI Sequence Length Specification	b2 b0 Sequence Length Referenced SPCMD0 to SPCMD7 (No.) 0 0 0: 1 0→0→... 0 0 1: 2 0→1→0→... 0 1 0: 3 0→1→2→0→... 0 1 1: 4 0→1→2→3→0→... 1 0 0: 5 0→1→2→3→4→0→... 1 0 1: 6 0→1→2→3→4→5→0→... 1 1 0: 7 0→1→2→3→4→5→6→0→... 1 1 1: 8 0→1→2→3→4→5→6→7→0→... SPCMD0 to SPCMD7 to be referenced and the order in which they are referenced are changed according to the sequence length that is set in these bits. The relationship among the setting of these bits, sequence length, and SPCMD0 to SPCMD7 registers referenced by the RSPI is shown above. However, the RSPI in slave mode always references SPCMD0.	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SPSCR sets the sequence length when the RSPI operates in master mode. When changing the SPSCR.SPSSLN[2:0] bits while both the SPCR.MSTR and SPCR.SPE bits are 1, the bits should be changed while the SPSR.IDLNF flag is 0.

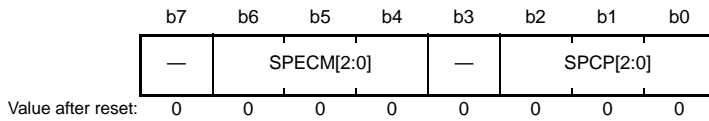
#### SPSLN[2:0] Bits (RSPI Sequence Length Specification)

The SPSLN[2:0] bits specify a sequence length when the RSPI in master mode performs sequential operations. The RSPI in master mode changes SPCMD0 to SPCMD7 registers to be referenced and the order in which they are referenced according to the sequence length that is set in the SPSLN[2:0] bits.

In slave mode, SPCMD0 is always referred to.

### 35.2.7 RSPI Sequence Status Register (SPSSR)

Address(es): RSPI0.SPSSR 0008 8389h, RSPI1.SPSSR 0008 83A9h, RSPI2.SPSSR 0008 83C9h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPCP[2:0]	RSPI Command Pointer	b2 b0 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
b3	—	Reserved	This bit is read as 0.	R
b6 to b4	SPECM[2:0]	RSPI Error Command	b6 b4 0 0 0: SPCMD0 0 0 1: SPCMD1 0 1 0: SPCMD2 0 1 1: SPCMD3 1 0 0: SPCMD4 1 0 1: SPCMD5 1 1 0: SPCMD6 1 1 1: SPCMD7	R
b7	—	Reserved	This bit is read as 0.	R

SPSSR indicates the sequence control status when the RSPI operates in master mode. Any writing to SPSSR is ignored.

#### SPCP[2:0] Bits (RSPI Command Pointer)

The SPCP[2:0] bits indicate SPCMD<sub>m</sub> that is currently pointed to by the pointer during sequence control by the RSPI. For the RSPI's sequence control, see section 35.3.10.1, Master Mode Operation.

#### SPECM[2:0] Bits (RSPI Error Command)

The SPECM[2:0] bits indicate SPCMD<sub>m</sub> that is specified by the SPCP[2:0] bits when an error is detected during sequence control by the RSPI. The RSPI updates the SPECM[2:0] bits only when an error is detected. If both the SPSR.OVRF and SPSR.MODF bits are 0 and there is no error, the values of the SPECM[2:0] bits have no meaning. For the RSPI's error detection function, see section 35.3.8, Error Detection. For the RSPI's sequence control, see section 35.3.10.1, Master Mode Operation.

### 35.2.8 RSPI Bit Rate Register (SPBR)

Address(es): RSPI0.SPBR 0008 838Ah, RSPI1.SPBR 0008 83AAh, RSPI2.SPBR 0008 83CAh

	b7	b6	b5	b4	b3	b2	b1	b0
	SPR7	SPR6	SPR5	SPR4	SPR3	SPR2	SPR1	SPR0
Value after reset:	1	1	1	1	1	1	1	1

SPBR sets the bit rate in master mode. If the contents of SPBR are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations cannot be guaranteed.

When the RSPI is used in slave mode, the bit rate depends on the bit rate of the input clock (bit rate satisfying the electrical characteristics should be used) regardless of the settings of SPBR and the SPCMDm.BRDV[1:0] bits (bit rate division setting bits).

The bit rate is determined by combinations of the SPBR setting and the SPCMDm.BRDV[1:0] bit setting. The equation for calculating the bit rate is given below. In the equation, n denotes an SPBR setting (0, 1, 2, ..., 255), and N denotes a BRDV[1:0] bit setting (0, 1, 2, 3).

$$\text{Bit rate} = \frac{f(\text{PCLK})}{2 \times (n + 1) 2^N}$$

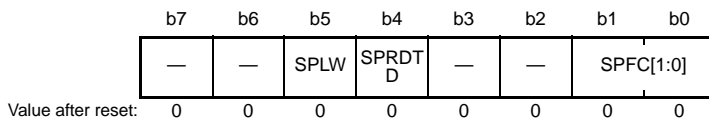
Table 35.3 lists examples of the relationship among the SPBR settings, the BRDV[1:0] settings, and bit rates.

**Table 35.3 Relationship among SPBR Settings, BRDV[1:0] Settings, and Bit Rates**

SPBR (n)	BRDV[1:0] Bits (N)	Division Ratio	Bit Rate			
			PCLK = 32 MHz	PCLK = 36 MHz	PCLK = 40 MHz	PCLK = 50 MHz
0	0	2	16.0 Mbps	18.0 Mbps	20.0 Mbps	25.0 Mbps
1	0	4	8.00 Mbps	9.00 Mbps	10.0 Mbps	12.5 Mbps
2	0	6	5.33 Mbps	6.00 Mbps	6.67 Mbps	8.33 Mbps
3	0	8	4.00 Mbps	4.50 Mbps	5.00 Mbps	6.25 Mbps
4	0	10	3.20 Mbps	3.60 Mbps	4.00 Mbps	5.00 Mbps
5	0	12	2.67 Mbps	3.00 Mbps	3.33 Mbps	4.16 Mbps
5	1	24	1.33 Mbps	1.50 Mbps	1.67 Mbps	2.08 Mbps
5	2	48	667 kbps	750 kbps	833 kbps	1.04 Mbps
5	3	96	333 kbps	375 kbps	417 kbps	521 kbps
255	3	4096	7.81 kbps	8.80 kbps	9.78 kbps	12.2 kbps

### 35.2.9 RSPI Data Control Register (SPDCR)

Address(es): RSPI0.SPDCR 0008 838Bh, RSPI1.SPDCR 0008 83ABh, RSPI2.SPDCR 0008 83CBh



Bit	Symbol	Bit Name	Description	R/W
b1, b0	SPFC[1:0]	Number of Frames Specification	b1 b0 0 0: 1 frame 0 1: 2 frames 1 0: 3 frames 1 1: 4 frames	R/W
b3, b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	SPRDTD	RSPI Receive/Transmit Data Selection	0: SPDR values are read from the receive buffer 1: SPDR values are read from the transmit buffer (but only if the transmit buffer is empty)	R/W
b5	SPLW	RSPI Longword Access/Word Access Specification	0: SPDR is accessed in words 1: SPDR is accessed in longwords	R/W
b7, b6	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Up to four frames can be transmitted or received in one round of transmission or reception activation. The amount of data in each transfer is controlled by the combination of the SPCMDm.SPB[3:0] bits, the SPSCR.SPSSLN[2:0] bits, and the SPDCR.SPFC[1:0] bits.

When changing the SPDCR.SPFC[1:0] bits while the SPSCR.SPE bit is 1, the bits should be changed while the SPSCR.IDLNF flag is 0.

#### SPFC[1:0] Bits (Number of Frames Specification)

The SPFC[1:0] bits specify the number of frames that can be stored in SPDR (per transfer activation). Up to four frames can be transmitted or received in one round of transmission or reception, and the amount of data is determined by the combination of the SPSCR.SPSSLN[2:0] bits, and the SPDCR.SPFC[1:0] bits. Furthermore, the setting of the SPFC[1:0] bits adjusts the number of frames for generation of RSPI reception interrupts, and start of transmission or generation of RSPI transmission interrupts. Table 35.4 show the frame configurations that can be stored in SPDR and examples of combinations of settings for transmission and reception. If combinations of settings other than those shown in the examples are made, subsequent operations are not guaranteed.

#### SPRDTD Bit (RSPI Receive/Transmit Data Selection)

The SPRDTD bit selects whether the SPDR reads values from the receive buffer or from the transmit buffer.

If reading is from the transmit buffer, the value written to SPDR register immediately beforehand is read.

When reading the transmit buffer, do so before writing of the number of frames set in the SPFC[1:0] bits is finished and after generation of the RSPI transmission interrupt.

For details, see section section 35.2.5, RSPI Data Register (SPDR).

#### SPLW Bit (RSPI Longword Access/Word Access Specification)

The SPLW bit specifies the access width for SPDR. Access to SPDR is in words when the SPLW bit is 0 and in longwords when the SPLW bit is 1.

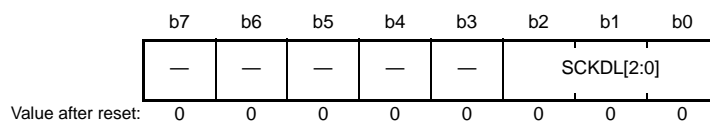
Also, when the SPLW bit is 0, set the SPCMDm.SPB[3:0] bits (RSPI data length setting bits) to 8 to 16 bits. When 20, 24, or 32 bits is specified, operation is not guaranteed.

**Table 35.4 Settable Combinations of SPSLN[2:0] Bits and SPFC[1:0] Bits**

Setting	SPSLN[2:0]	SPFC[1:0]	Number of Frames in a Single Sequence	Number of Frames at which Receive Buffer Full Interrupt Occurs or Transmit Buffer Holding Data is Recognized
1-1	000	00	1	1
1-2	000	01	2	2
1-3	000	10	3	3
1-4	000	11	4	4
2-1	001	01	2	2
2-2	001	11	4	4
3	010	10	3	3
4	011	11	4	4
5	100	00	5	1
6	101	00	6	1
7	110	00	7	1
8	111	00	8	1

### 35.2.10 RSPI Clock Delay Register (SPCKD)

Address(es): RSPI0.SPCKD 0008 838Ch, RSPI1.SPCKD 0008 83ACh, RSPI2.SPCKD 0008 83CCh



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SCKDL[2:0]	RSPCK Delay Setting	b2 b0 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

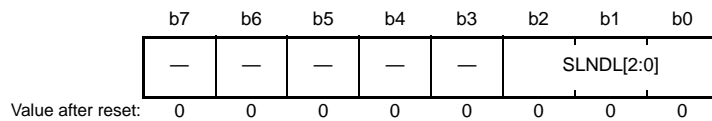
SPCKD sets a period from the beginning of SSLni signal assertion to RSPCK oscillation (RSPCK delay) when the SPCMDm.SCKDEN bit is 1. If the contents of SPCKD are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations cannot be guaranteed.

#### SCKDL[2:0] Bits (RSPCK Delay Setting)

The SCKDL[2:0] bits set an RSPCK delay value when the SPCMDm.SCKDEN bit is 1. When using the RSPI in slave mode, set the SCKDL[2:0] bits to 000b.

### 35.2.11 RSPI Slave Select Negation Delay Register (SSLND)

Address(es): RSPI0.SSLND 0008 838Dh, RSPI1.SSLND 0008 83ADh, RSPI2.SSLND 0008 83CDh



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SLNDL[2:0]	SSL Negation Delay Setting	b2 b0 0 0 0: 1 RSPCK 0 0 1: 2 RSPCK 0 1 0: 3 RSPCK 0 1 1: 4 RSPCK 1 0 0: 5 RSPCK 1 0 1: 6 RSPCK 1 1 0: 7 RSPCK 1 1 1: 8 RSPCK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

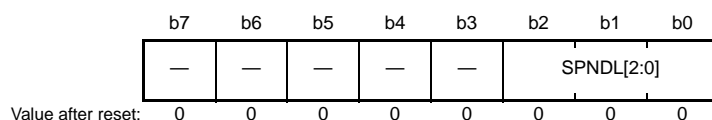
SSLND sets a period (SSL negation delay) from the transmission of a final RSPCK edge to the negation of the SSLni signal during a serial transfer by the RSPI in master mode. If the contents of SSLND are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations cannot be guaranteed.

#### SLNDL[2:0] Bits (SSL Negation Delay Setting)

The SLNDL[2:0] bits set an SSL negation delay value when the RSPI is in master mode. When using the RSPI in slave mode, set the SLNDL[2:0] bits to 000b.

### 35.2.12 RSPI Next-Access Delay Register (SPND)

Address(es): RSPI0.SPND 0008 838Eh, RSPI1.SPND 0008 83AEh, RSPI2.SPND 0008 83CEh



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	SPNDL[2:0]	RSPI Next-Access Delay Setting	b2 b0 0 0 0: 1 RSPCK + 2 PCLK 0 0 1: 2 RSPCK + 2 PCLK 0 1 0: 3 RSPCK + 2 PCLK 0 1 1: 4 RSPCK + 2 PCLK 1 0 0: 5 RSPCK + 2 PCLK 1 0 1: 6 RSPCK + 2 PCLK 1 1 0: 7 RSPCK + 2 PCLK 1 1 1: 8 RSPCK + 2 PCLK	R/W
b7 to b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

SPND sets a non-active period (next-access delay) of the SSLni signal after termination of a serial transfer when the SPCMDm.SPNDEN bit is 1. If the contents of SPND are changed while both the SPCR.MSTR and SPCR.SPE bits are 1, subsequent operations cannot be guaranteed.

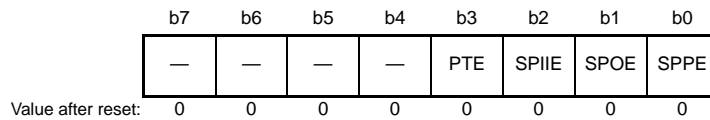
**SPNDL[2:0] Bits (RSPI Next-Access Delay Setting)**

The SPNDL[2:0] bits set a next-access delay when the SPCMDm.SPNDEN bit is 1.

When using the RSPI in slave mode, set the SPNDL[2:0] bits to 000b.

**35.2.13 RSPI Control Register 2 (SPCR2)**

Address(es): RSPI0.SPCR2 0008 838Fh, RSPI1.SPCR2 0008 83AFh, RSPI2.SPCR2 0008 83CFh



Bit	Symbol	Bit Name	Description	R/W
b0	SPPE	Parity Enable	0: Does not add the parity bit to transmit data and does not check the parity bit of receive data 1: Adds the parity bit to transmit data and checks the parity bit of receive data (when SPCR.TXMD = 0) Adds the parity bit to transmit data but does not check the parity bit of receive data (when SPCR.TXMD = 1)	R/W
b1	SPOE	Parity Mode	0: Selects even parity for use in transmission and reception 1: Selects odd parity for use in transmission and reception	R/W
b2	SPIIE	RSPI Idle Interrupt Enable	0: Disables the generation of idle interrupt requests 1: Enables the generation of idle interrupt requests	R/W
b3	PTE	Parity Self-Testing	0: Disables the self-diagnosis function of the parity circuit 1: Enables the self-diagnosis function of the parity circuit	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

If the SPPE bit or SPOE bit in SPCR2 is changed while the SPCR.SPE bit is 1, subsequent operations cannot be guaranteed.

**SPPE Bit (Parity Enable)**

The SPPE bit enables or disables the parity function.

The parity bit is added to transmit data and parity checking is performed for receive data when the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1.

The parity bit is added to transmit data but parity checking is not performed for receive data when the SPCR.TXMD bit is 1 and the SPCR2.SPPE bit is 1.

**SPOE Bit (Parity Mode)**

The SPOE bit specifies odd or even parity.

When even parity is set, parity bit addition is performed so that the total number of 1-bits in the transmit/receive character plus the parity bit is even. Similarly, when odd parity is set, parity bit addition is performed so that the total number of 1-bits in the transmit/receive character plus the parity bit is odd.

The SPOE bit is valid only when the SPPE bit is 1.

**SPIIE Bit (RSPI Idle Interrupt Enable)**

The SPIIE bit enables or disables the generation of RSPI idle interrupt requests when the RSPI being in the idle state is detected and the SPSR.IDLNF flag is cleared to 0.

**PTE Bit (Parity Self-Testing)**

The PTE bit enables the self-diagnosis function of the parity circuit in order to check whether the parity function is operating correctly.

**35.2.14 RSPI Command Registers 0 to 7 (SPCMD0 to SPCMD7)**

Address(es): RSPI0.SPCMD0 0008 8390h, RSPI0.SPCMD1 0008 8392h, RSPI0.SPCMD2 0008 8394h, RSPI0.SPCMD3 0008 8396h, RSPI0.SPCMD4 0008 8398h, RSPI0.SPCMD5 0008 839Ah, RSPI0.SPCMD6 0008 839Ch, RSPI0.SPCMD7 0008 839Eh, RSPI1.SPCMD0 0008 83B0h, RSPI1.SPCMD1 0008 83B2h, RSPI1.SPCMD2 0008 83B4h, RSPI1.SPCMD3 0008 83B6h, RSPI1.SPCMD4 0008 83B8h, RSPI1.SPCMD5 0008 83BAh, RSPI1.SPCMD6 0008 83BCh, RSPI1.SPCMD7 0008 83BEh, RSPI2.SPCMD0 0008 83D0h, RSPI2.SPCMD1 0008 83D2h, RSPI2.SPCMD2 0008 83D4h, RSPI2.SPCMD3 0008 83D6h, RSPI2.SPCMD4 0008 83D8h, RSPI2.SPCMD5 0008 83DAh, RSPI2.SPCMD6 0008 83DCh, RSPI2.SPCMD7 0008 83DEh

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
SCKDEN	SLNDEN	SPNDEN	LSBF		SPB[3:0]		SSLKP		SSLA[2:0]		BRDV[1:0]	CPOL	CPHA		
0	0	0	0	0	1	1	1	0	0	0	0	1	1	0	1

Bit	Symbol	Bit Name	Description	R/W
b0	CPHA	RSPCK Phase Setting	0: Data sampling on odd edge, data variation on even edge 1: Data variation on odd edge, data sampling on even edge	R/W
b1	CPOL	RSPCK Polarity Setting	0: RSPCK is low when idle 1: RSPCK is high when idle	R/W
b3, b2	BRDV[1:0]	Bit Rate Division Setting	b3 b2 0 0: These bits select the base bit rate 0 1: These bits select the base bit rate divided by 2 1 0: These bits select the base bit rate divided by 4 1 1: These bits select the base bit rate divided by 8	R/W
b6 to b4	SSLA[2:0]	SSL Signal Assertion Setting	b6 b4 0 0 0: SSL0 0 0 1: SSL1 0 1 0: SSL2 0 1 1: SSL3 1 x x: — (Setting prohibited) x: Don't care	R/W
b7	SSLKP	SSL Signal Level Keeping	0: Negates all SSL signals upon completion of transfer 1: Keeps the SSL signal level from the end of transfer until the beginning of the next access	R/W
b11 to b8	SPB[3:0]	RSPI Data Length Setting	b11 b8 0100 to 0111: 8 bits 1 0 0 0: 9 bits 1 0 0 1: 10 bits 1 0 1 0: 11 bits 1 0 1 1: 12 bits 1 1 0 0: 13 bits 1 1 0 1: 14 bits 1 1 1 0: 15 bits 1 1 1 1: 16 bits 0 0 0 0: 20 bits 0 0 0 1: 24 bits 0010, 0011: 32 bits	R/W
b12	LSBF	RSPI LSB First	0: MSB first 1: LSB first	R/W
b13	SPNDEN	RSPI Next-Access Delay Enable	0: A next-access delay of 1 RSPCK + 2 PCLK 1: A next-access delay is equal to the setting of the RSPI next-access delay register (SPND)	R/W
b14	SLNDEN	SSL Negation Delay Setting Enable	0: An SSL negation delay of 1 RSPCK 1: An SSL negation delay is equal to the setting of the RSPI slave select negation delay register (SSLND)	R/W



Bit	Symbol	Bit Name	Description	R/W
b15	SCKDEN	RSPCK Delay Setting Enable	0: An RSPCK delay of 1 RSPCK 1: An RSPCK delay is equal to the setting of the RSPI clock delay register (SPCKD)	R/W

SPCMDm register is used to set a transfer format for the RSPI in master mode. Each channel has eight RSPI command registers (SPCMD0 to SPCMD7). Some of the bits in SPCMD0 register is used to set a transfer mode for the RSPI in slave mode. The RSPI in master mode sequentially references SPCMDm register according to the settings in the SPSCR.SPSSLN[2:0] bits, and executes the serial transfer that is set in the referenced SPCMDm register.

SPCMDm register should be set while the transmit buffer is empty (data for the next transfer is not set) and before setting of the data that is to be transmitted when that SPCMDm register is referenced.

SPCMDm that is referenced by the RSPI in master mode can be checked by means of the SPSSR.SPCP[2:0] bits. If the contents of SPCMDm are changed while the SPCR.MSTR bit is 0 and the SPCR.SPE bit is 1, subsequent operations cannot be guaranteed.

### CPHA Bit (RSPCK Phase Setting)

The CPHA bit sets an RSPCK phase of the RSPI in master mode or slave mode. Data communications between RSPI modules require the same RSPCK phase setting between the modules.

### CPOL Bit (RSPCK Polarity Setting)

The CPOL bit sets an RSPCK polarity of the RSPI in master mode or slave mode. Data communications between RSPI modules require the same RSPCK polarity setting between the modules.

### BRDV[1:0] Bits (Bit Rate Division Setting)

The BRDV[1:0] bits are used to determine the bit rate. A bit rate is determined by combinations of the settings in the BRDV[1:0] bits and SPBR (see section 35.2.8, RSPI Bit Rate Register (SPBR)). The settings in SPBR determine the base bit rate. The settings in the BRDV[1:0] bits are used to select a bit rate which is obtained by dividing the base bit rate by 1, 2, 4, or 8. In SPCMDm register, different BRDV[1:0] bit settings can be specified. This permits the execution of serial transfers at a different bit rate for each command.

### SSLA[2:0] Bits (SSL Signal Assertion Setting)

The SSLA[2:0] bits control the SSLni signal assertion when the RSPI performs serial transfers in master mode.

Setting the SSLA[2:0] bits controls the assertion for the SSLni signal. When an SSLni signal is asserted, its polarity is determined by the set value in the corresponding SSLP. When the SSLA[2:0] bits are set to 000b in multi-master mode, serial transfers are performed with all the SLL signals in the negated state (as the SSLn0 pin acts as input).

When using the RSPI in slave mode, set the SSLA[2:0] bits to 000b.

### SSLKP Bit (SSL Signal Level Keeping)

When the RSPI in master mode performs a serial transfer, the SSLKP bit specifies whether the SSLni signal level for the current command is to be kept or negated between the SSL negation timing associated with the current command and the SSL assertion timing associated with the next command.

When using the RSPI in slave mode, the SSLKP bit should be set to 0.

### SPB[3:0] Bits (RSPI Data Length Setting)

The SPB[3:0] bits set a transfer data length for the RSPI in master mode or slave mode.

### LSBF Bit (RSPI LSB First)

The LSBF bit sets the data format of the RSPI in master mode or slave mode to MSB first or LSB first.

**SPNDEN Bit (RSPI Next-Access Delay Enable)**

The SPNDEN bit sets the period from the time the RSPI in master mode terminates a serial transfer and sets the SSLni signal inactive until the RSPI enables the SSLni signal assertion for the next access (next-access delay). If the SPNDEN bit is 0, the RSPI sets the next-access delay to  $1 \text{ RSPCK} + 2 \text{ PCLK}$ . If the SPNDEN bit is 1, the RSPI inserts a next-access delay in compliance with the SPND setting.

When using the RSPI in slave mode, the SPNDEN bit should be set to 0.

**SLNDEN Bit (SSL Negation Delay Setting Enable)**

The SLNDEN bit sets the period from the time the RSPI in master mode stops RSPCK oscillation until the RSPI sets the SSLni signal inactive (SSL negation delay). If the SLNDEN bit is 0, the RSPI sets the SSL negation delay to  $1 \text{ RSPCK}$ . If the SLNDEN bit is 1, the RSPI negates the SSL signal at an SSL negation delay in compliance with the SSLND setting.

When using the RSPI in slave mode, the SLNDEN bit should be set to 0.

**SCKDEN Bit (RSPCK Delay Setting Enable)**

The SCKDEN bit sets the period from the point when the RSPI in master mode activates the SSLni signal until the RSPCK starts oscillation (RSPI clock delay). If the SCKDEN bit is 0, the RSPI sets the RSPCK delay to  $1 \text{ RSPCK}$ . If the SCKDEN bit is 1, the RSPI starts the oscillation of RSPCK at an RSPCK delay in compliance with the SPCKD setting.

When using the RSPI in slave mode, the SCKDEN bit should be set to 0.

## 35.3 Operation

In this section, the serial transfer period means a period from the beginning of driving valid data to the fetching of the final valid data.

### 35.3.1 Overview of RSPI Operations

The RSPI is capable of synchronous serial transfers in slave mode (SPI operation), single-master mode (SPI operation), multi-master mode (SPI operation), slave mode (clock synchronous operation), and master mode (clock synchronous operation). A particular mode of the RSPI can be selected by using the MSTR, MODFEN, and SPMS bits in SPCR.

Table 35.5 lists the relationship between RSPI modes and SPCR settings, and a description of each mode.

**Table 35.5 Relationship between RSPI Modes and SPCR Settings and Description of Each Mode**

Mode	Slave (SPI Operation)	Single-Master (SPI Operation)	Multi-Master (SPI Operation)	Slave (Clock Synchronous Operation)	Master (Clock Synchronous Operation)
MSTR bit setting	0	1	1	0	1
MODFEN bit setting	0 or 1	0	1	0	0
SPMS bit setting	0	0	0	1	1
RSPCKn signal	Input	Output	Output/Hi-Z	Input	Output
MOSIn signal	Input	Output	Output/Hi-Z	Input	Output
MISO <sub>n</sub> signal	Output/Hi-Z	Input	Input	Output	Input
SSL <sub>n0</sub> signal	Input	Output	Input	Hi-Z*1	Hi-Z*1
SSL <sub>n1</sub> to SSL <sub>n3</sub> signals	Hi-Z*1	Output	Output/Hi-Z	Hi-Z*1	Hi-Z*1
SSL polarity modification function	Supported	Supported	Supported	—	—
Transfer rate	Up to PCLK/8	Up to PCLK/2	Up to PCLK/2	Up to PCLK/8	Up to PCLK/2
Clock source	RSPCK input	On-chip baud rate generator	On-chip baud rate generator	RSPCK input	On-chip baud rate generator
Clock polarity	Two	Two	Two	Two	Two
Clock phase	Two	Two	Two	One (CPHA = 1)	Two
First transfer bit	MSB/LSB	MSB/LSB	MSB/LSB	MSB/LSB	MSB/LSB
Transfer data length	8 to 32 bits	8 to 32 bits	8 to 32 bits	8 to 32 bits	8 to 32 bits
Burst transfer	Possible (CPHA = 1)	Possible (CPHA = 0,1)	Possible (CPHA = 0,1)	—	—
RSPCK delay control	Not supported	Supported	Supported	Not supported	Supported
SSL negation delay control	Not supported	Supported	Supported	Not supported	Supported
Next-access delay control	Not supported	Supported	Supported	Not supported	Supported
Transfer activation method	SSL input active or RSPCK oscillation	Transmit buffer is written to at generation of a transmit buffer empty interrupt request	Transmit buffer is written to at generation of a transmit buffer empty interrupt request	RSPCK oscillation	Transmit buffer is written to at generation of a transmit buffer empty interrupt request
Sequence control	Not supported	Supported	Supported	Not supported	Supported
Transmit buffer empty detection	Supported	Supported	Supported	Supported	Supported
Receive buffer full detection	Supported*2	Supported*2	Supported*2	Supported*2	Supported*2
Overrun error detection	Supported*2	Supported*2	Supported*2	Supported*2	Supported*2
Parity error detection	Supported*2,*3	Supported*2,*3	Supported*2,*3	Supported*2,*3	Supported*2,*3
Mode fault error detection	Supported (MODFEN = 1)	Not supported	Supported	Not supported	Not supported

Note 1. This function is not supported in this mode.

Note 2. When the SPCR.TXMD bit is 1, receiver buffer full detection, overrun error detection, and parity error detection are not performed.

Note 3. When the SPCR2.SPPE bit is 0, parity error detection is not performed.

### 35.3.2 Controlling RSPI Pins

The RSPI can switch pin states in response to the settings of the MSTR, MODFEN, and SPMS bits in SPCR and the ODRn.Bi bit for the I/O port. Setting the ODRn.Bi bit for an I/O port to 0 selects CMOS output; setting it to 1 selects open-drain output. Table 35.6 lists the relationship between pin states and bit settings. The I/O port settings should follow this relationship.

**Table 35.6 Relationship between Pin States and Bit Settings**

Mode	Pin	Pin State*2	
		ODRn.Bi = 0	ODRn.Bi = 1
Single-master mode (SPI operation) (MSTR = 1, MODFEN = 0, SPMS = 0)	RSPCKn	CMOS output	Open-drain output
	SSLn0 to SSLn3	CMOS output	Open-drain output
	MOSIn	CMOS output	Open-drain output
	MISO <sub>n</sub>	Input	Input
Multi-master mode (SPI operation) (MSTR = 1, MODFEN = 1, SPMS = 0)	RSPCKn*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	SSLn0	Input	Input
	SSLn1 to SSLn3*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
	MOSIn*3	CMOS output/Hi-Z	Open-drain output/Hi-Z
Slave mode (SPI operation) (MSTR = 0, SPMS = 0)	RSPCKn	Input	Input
	SSLn0	Input	Input
	SSLn1 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	Input	Input
Master mode (Clock synchronous operation) (MSTR = 1, MODFEN = 0, SPMS = 1)	RSPCKn	CMOS output	Open-drain output
	SSLn0 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	CMOS output	Open-drain output
	MISO <sub>n</sub>	Input	Input
Slave mode (Clock synchronous operation) (MSTR = 0, SPMS = 1)	RSPCKn	Input	Input
	SSLn0 to SSLn3*5	Hi-Z*1	Hi-Z*1
	MOSIn	Input	Input
	MISO <sub>n</sub>	CMOS output	Open-drain output

Note 1. This function is not supported in this mode.

Note 2. RSPI settings are not reflected in the multiplex pins for which the RSPI function is not selected.

Note 3. When SSLn0 is at the active level, the pin state is Hi-Z.

Note 4. When SSLn0 is at the non-active level or the SPCR.SPE bit is cleared (= 0), the pin state is Hi-Z.

Note 5. These pins are available for use as I/O port pins.

The RSPI in single-master mode (SPI operation) or multi-master mode (SPI operation) determines MOSI signal values during the SSL negation period (including the SSL retention period during a burst transfer) according to MOIFE and MOIFV bit settings in SPPCR, as listed in Table 35.7.

**Table 35.7 MOSI Signal Value Determination during SSL Negation Period**

MOIFE Bit	MOIFV Bit	MOSIn Signal Value during SSL Negation Period
0	0, 1	Final data from previous transfer
1	0	Always low
1	1	Always high

### 35.3.3 RSPi System Configuration Examples

#### 35.3.3.1 Single Master/Single Slave (with This LSI Acting as Master)

Figure 35.5 shows a single-master/single-slave RSPi system configuration example when this LSI is used as a master. In the single-master/single-slave configuration, the SSLn0 to SSLn3 output of this LSI (master) are not used. The SSL input of the RSPi slave is fixed to the low level, and the RSPi slave is always maintained in a select state.\*1 This LSI (master) always drives the RSPCKn and MOSIn. The RSPi slave always drives the MISO.

Note 1. In the transfer format corresponding to the case where the SPCMDm.CPHA bit is 0, there are slave devices for which the SSL signal cannot be fixed to the active level. In situations where the SSL signal cannot be fixed, the SSLni output of this LSI should be connected to the SSL input of the slave device.

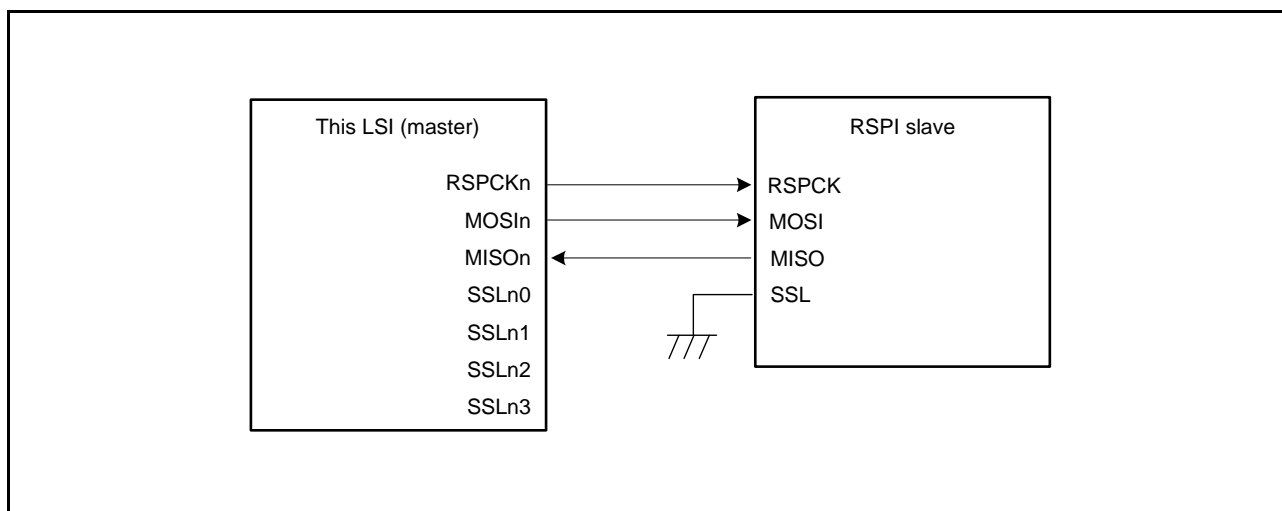


Figure 35.5 Single-Master/Single-Slave Configuration Example (This LSI = Master)

### 35.3.3.2 Single Master/Single Slave (with This LSI Acting as Slave)

Figure 35.6 shows a single-master/single-slave RSPi system configuration example when this LSI is used as a slave. When this LSI is to operate as a slave, the SSLn0 pin is used as SSL input. The RSPi master always drives the RSPCK and MOSI. This LSI (slave) always drives the MISO.<sup>\*1</sup>

In the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, the SSLn0 input of this LSI (slave) is fixed to the low level, this LSI (slave) is always maintained in a select state, and in this manner it is possible to execute serial transfer (Figure 35.7).

Note 1. When SSLn0 is at the non-active level, the pin state is Hi-Z.

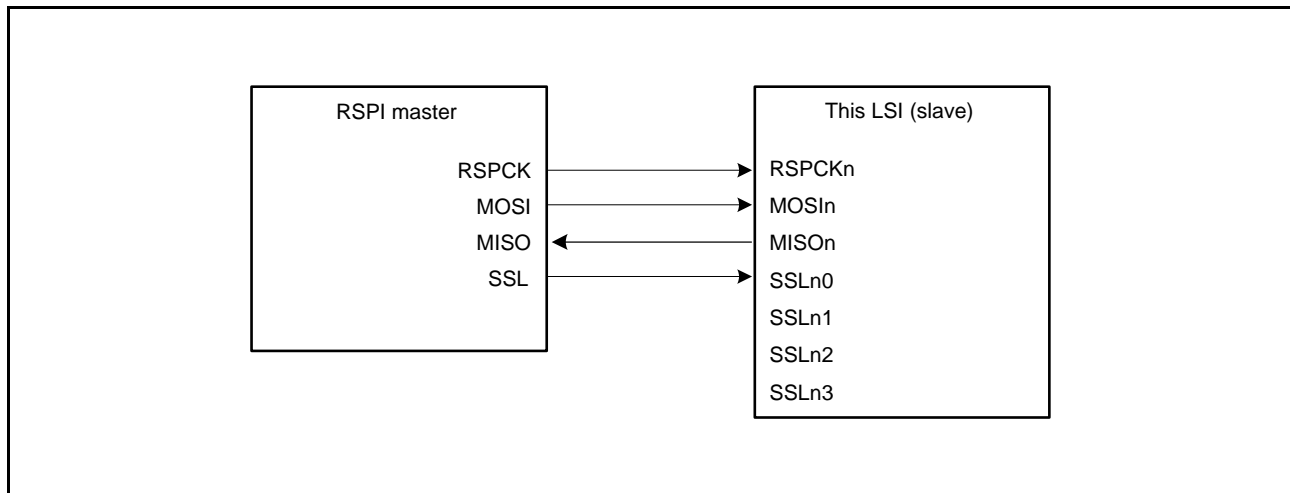


Figure 35.6 Single-Master/Single-Slave Configuration Example (This LSI = Slave, CPHA = 0)

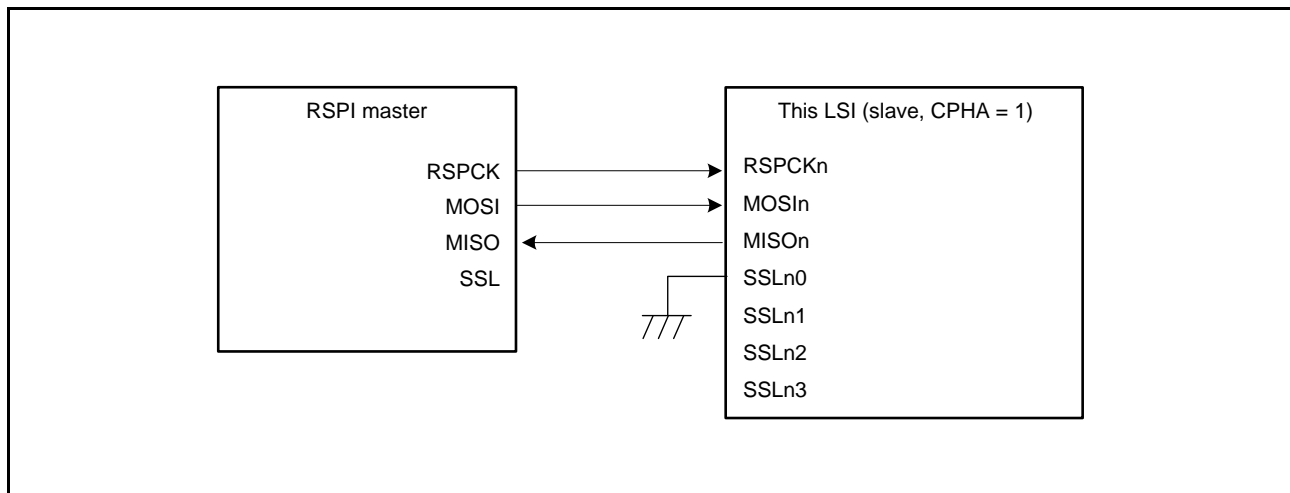


Figure 35.7 Single-Master/Single-Slave Configuration Example (This LSI = Slave, CPHA = 1)

### 35.3.3.3 Single Master/Multi-Slave (with This LSI Acting as Master)

Figure 35.8 shows a single-master/multi-slave RSPI system configuration example when this LSI is used as a master. In the example of Figure 35.8, the RSPI system is comprised of this LSI (master) and four slaves (RSPI slave 0 to RSPI slave 3).

The RSPCKn and MOSIn outputs of this LSI (master) are connected to the RSPCK and MOSI inputs of RSPI slave 0 to RSPI slave 3. The MISO outputs of RSPI slave 0 to RSPI slave 3 are all connected to the MISO<sub>n</sub> input of this LSI (master). SSLn0 to SSLn3 outputs of this LSI (master) are connected to the SSL inputs of RSPI slave 0 to RSPI slave 3, respectively.

This LSI (master) always drives RSPCK, MOSI, and SSLn0 to SSLn3. Of the RSPI slave 0 to RSPI slave 3, the slave that receives low-level input into the SSL input drives MISO.

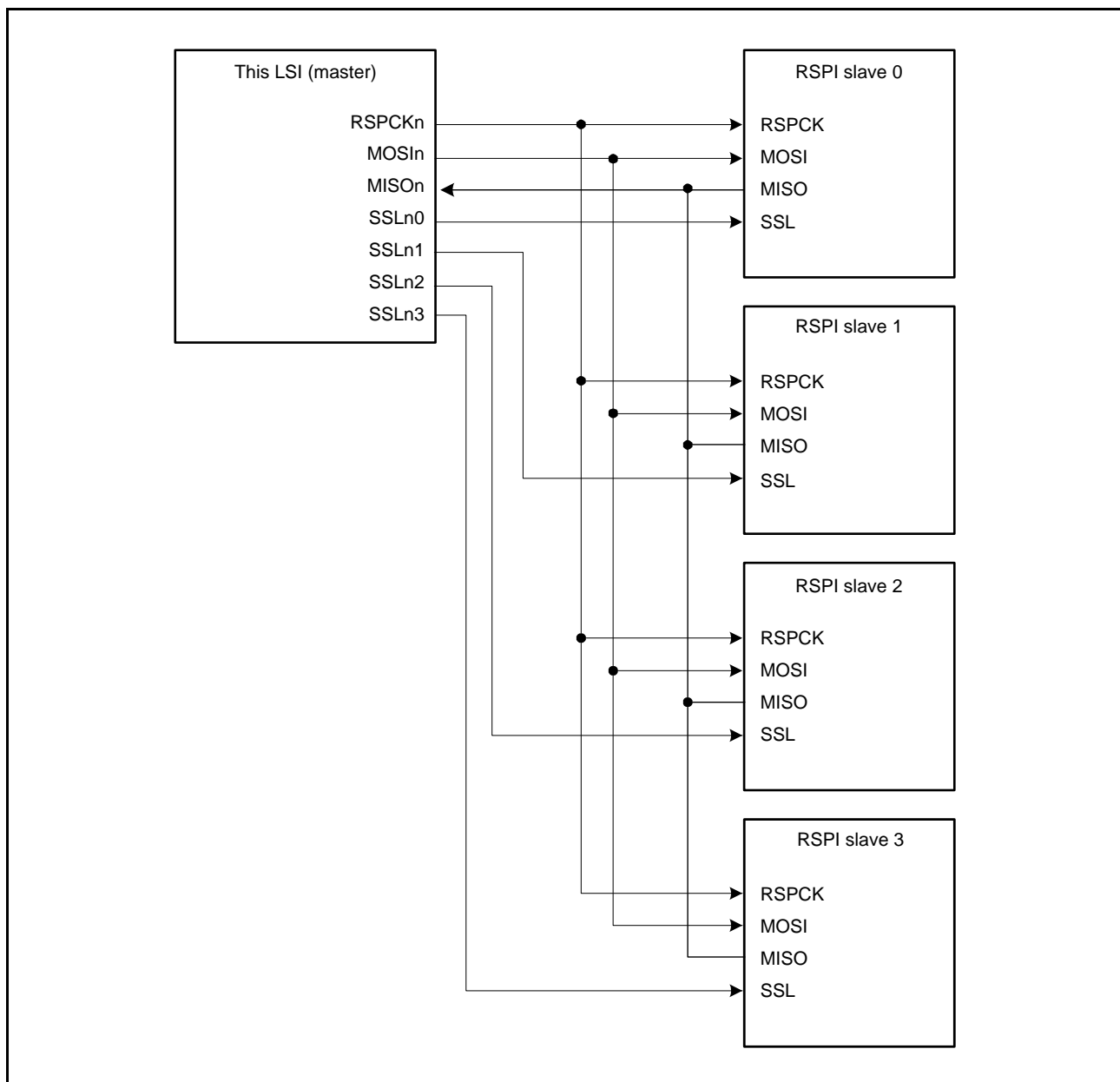


Figure 35.8 Single-Master/Multi-Slave Configuration Example (This LSI = Master)

### 35.3.3.4 Single Master/Multi-Slave (with This LSI Acting as Slave)

Figure 35.9 shows a single-master/multi-slave RSPI system configuration example when this LSI is used as a slave. In the example of Figure 35.9, the RSPI system is comprised of an RSPI master and two LSIs (slave X and slave Y).

The RSPCK and MOSI outputs of the RSPI master are connected to the RSPCKn and MOSIn inputs of the LSIs (slave X and slave Y). The MISO outputs of the LSIs (slave X and slave Y) are all connected to the MISO input of the RSPI master. SSLX and SSLY outputs of the RSPI master are connected to the SSLn0 inputs of the LSIs (slave X and slave Y), respectively.

The RSPI master always drives RSPCK, MOSI, SSLX, and SSLY. Of the LSIs (slave X and slave Y), the slave that receives low-level input into the SSLn0 input drives MISO.

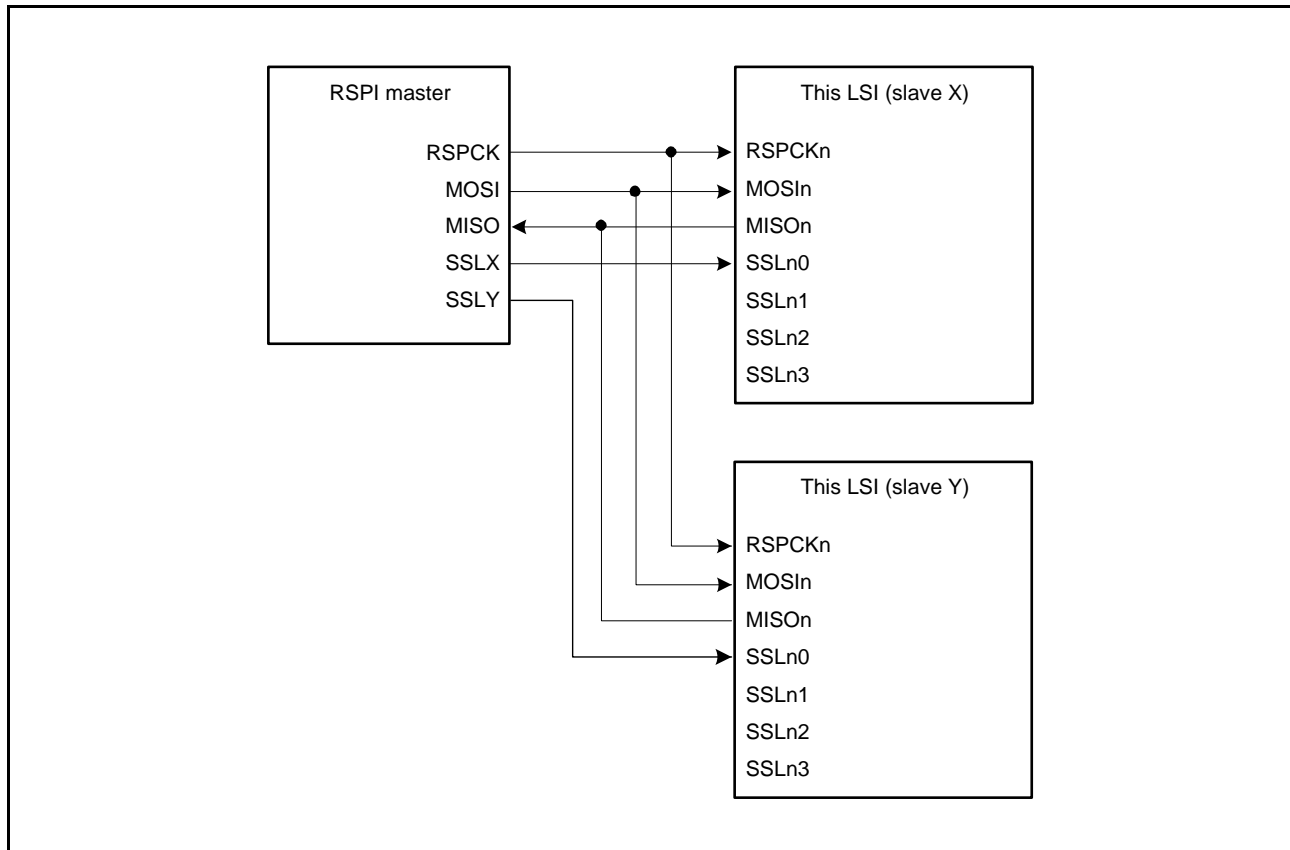


Figure 35.9 Single-Master/Multi-Slave Configuration Example (This LSI = Slave)



### 35.3.3.5 Multi-Master/Multi-Slave (with This LSI Acting as Master)

Figure 35.10 shows a multi-master/multi-slave RSPI system configuration example when this LSI is used as a master. In the example of Figure 35.10, the RSPI system is comprised of two LSIs (master X and master Y) and two RSPI slaves (RSPI slave 1 and RSPI slave 2).

The RSPCKn and MOSIn outputs of the LSIs (master X and master Y) are connected to the RSPCK and MOSI inputs of RSPI slaves 1 and 2. The MISO outputs of RSPI slaves 1 and 2 are connected to the MISO inputs of the LSIs (master X and master Y). Any generic port Y output from this LSI (master X) is connected to the SSLn0 input of this LSI (master Y). Any generic port X output of this LSI (master Y) is connected to the SSLn0 input of this LSI (master X). The SSLn1 and SSLn2 outputs of the LSIs (master X and master Y) are connected to the SSL inputs of the RSPI slaves 1 and 2. In this configuration example, since the system can be comprised solely of SSLn0 input, and SSLn1 and SSLn2 outputs for slave connections, the SSLn3 output of this LSI is not required.

This LSI drives RSPCKn, MOSIn, SSLn1, and SSLn2 when the SSLn0 input level is high. When the SSLn0 input level is low, this LSI detects a mode fault error, sets RSPCKn, MOSIn, SSLn1, and SSLn2 to Hi-Z, and releases the RSPI bus right to the other master. Of the RSPI slaves 1 and 2, the slave that receives low-level input into the SSL input drives MISO.

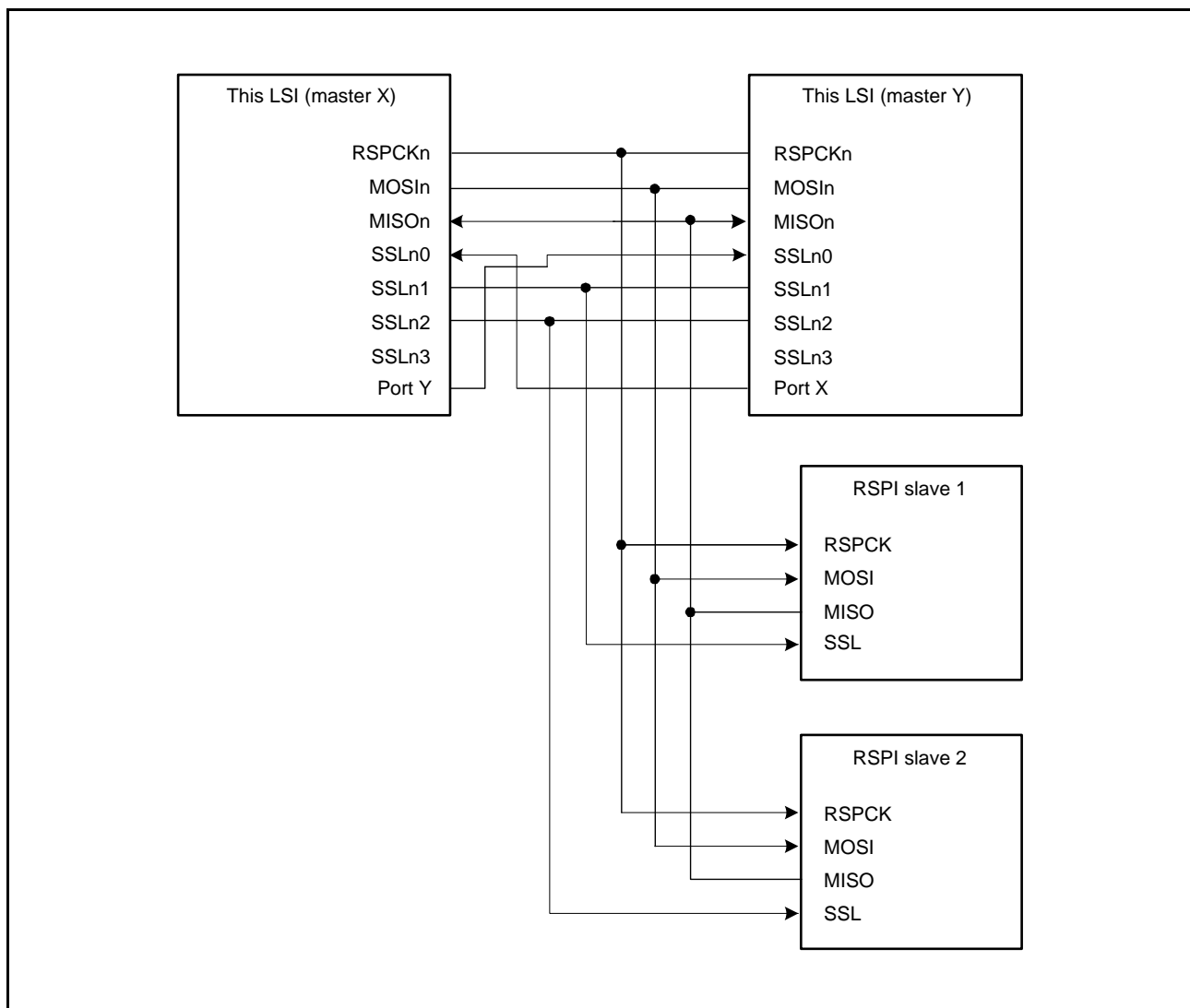
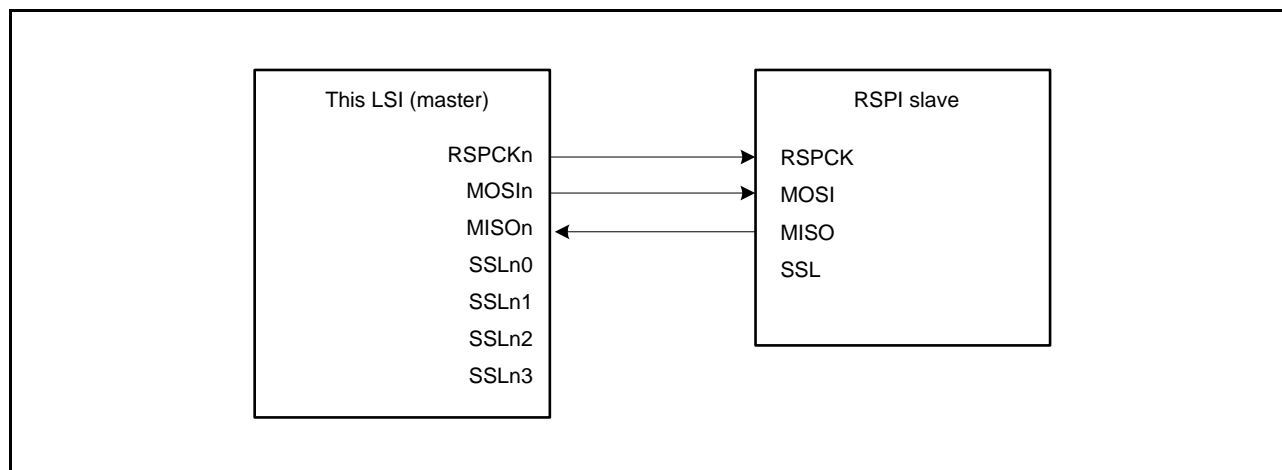


Figure 35.10 Multi-Master/Multi-Slave Configuration Example (This LSI = Master)

### 35.3.3.6 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This LSI Acting as Master)

Figure 35.11 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPi system configuration example when this LSI is used as a master. In the master (clock synchronous operation)/slave (clock synchronous operation) configuration, SSLn0 to SSLn3 of this LSI (master) are not used.

This LSI (master) always drives the RSPCKn and MOSIn. The RSPi slave always drives the MISO.

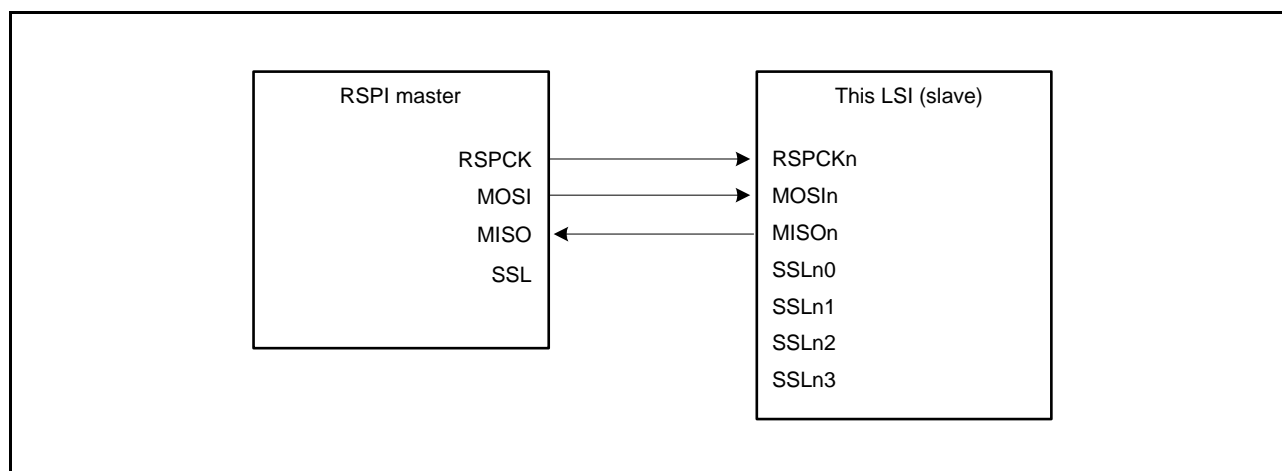


**Figure 35.11 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This LSI = Master)**

### 35.3.3.7 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) (with This LSI Acting as Slave)

Figure 35.12 shows a master (clock synchronous operation)/slave (clock synchronous operation) RSPi system configuration example when this LSI is used as a slave. When this LSI is to operate as a slave (clock synchronous operation), this LSI (slave) always drives the MISO and the RSPi master always drives the RSPCK and MOSI. In addition, SSLn0 to SSLn3 of this LSI (slave) are not used.

Only in the single-slave configuration in which the SPCMDm.CPHA bit is set to 1, this LSI (slave) can execute serial transfer.



**Figure 35.12 Master (Clock Synchronous Operation)/Slave (Clock Synchronous Operation) Configuration Example (This LSI = Slave, CPHA = 1)**

### 35.3.4 Data Format

The RSPI's data format depends on the settings in the RSPI command register  $m$  (SPCMD $m$ ) ( $m = 0$  to  $7$ ) and the parity enable bit of RSPI control register 2 (SPCR2.SPPE). Regardless of whether the MSB or LSB is first, the RSPI treats the range from the LSB bit of the RSPI data register (SPDR) to the selected data length as transfer data.

The format of one frame of data before or after transfer is shown below.

#### (a) With Parity Disabled

When parity is disabled, transmission or reception of data proceeds with the length in bits selected in the RSPI data length setting bits in the RSPI command register  $m$  (SPCMD $m$ .SPB[3:0]).

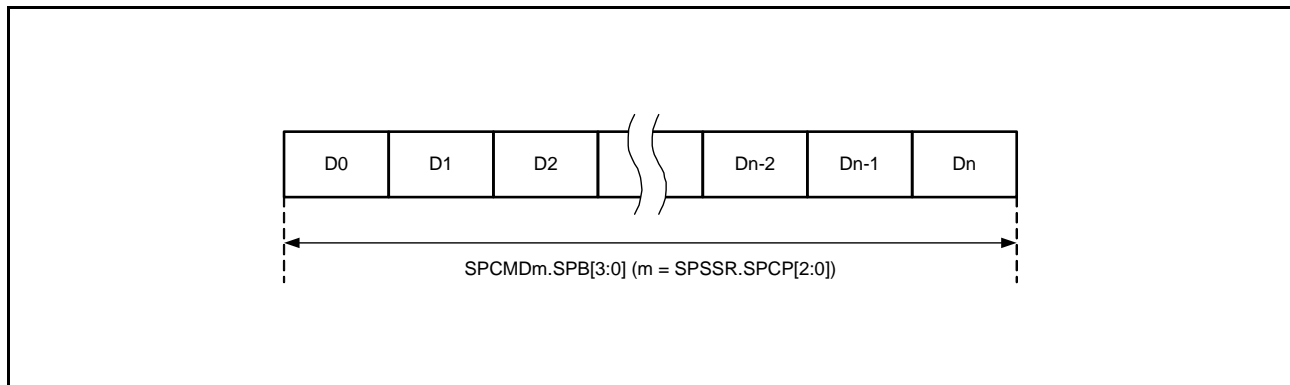


Figure 35.13 Outline of the Data Format (with Parity Disabled)

#### (b) With Parity Enabled

When parity is enabled, transmission or reception of data also proceeds with the length in bits selected in the SPCMD $m$ .SPB[3:0] bits. In this case, however, the last bit is a parity bit.

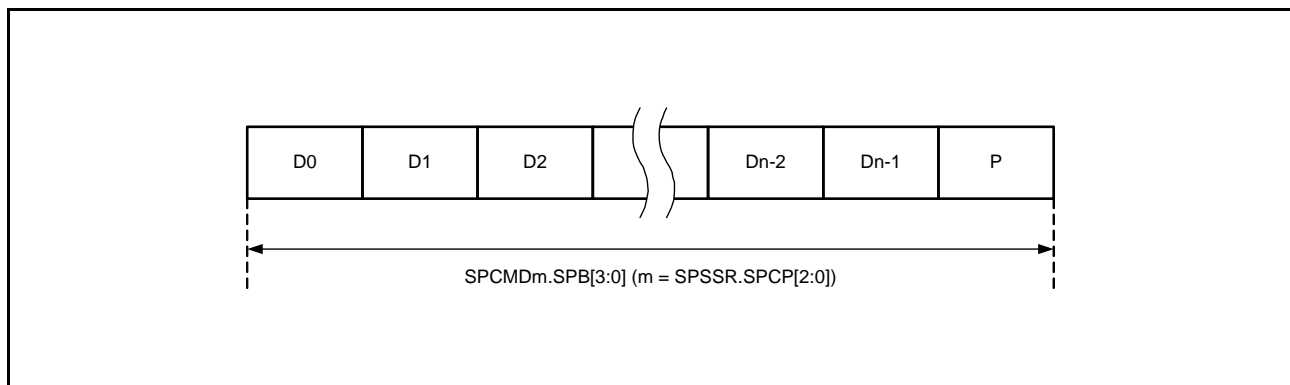


Figure 35.14 Outline of the Data Format (with Parity Enabled)

### 35.3.4.1 When Parity is Disabled (SPCR2.SPPE = 0)

When parity is disabled, data for transmission are copied to the shift register with no prior processing. A description of the connection between the RSPI data register (SPDR) and the shift register in terms of the combination of MSB- or LSB-first and data length is given below.

#### (1) MSB-First Transfer (32-Bit Data)

Figure 35.15 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and MSB-first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T31, through T30, and so on to T00.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

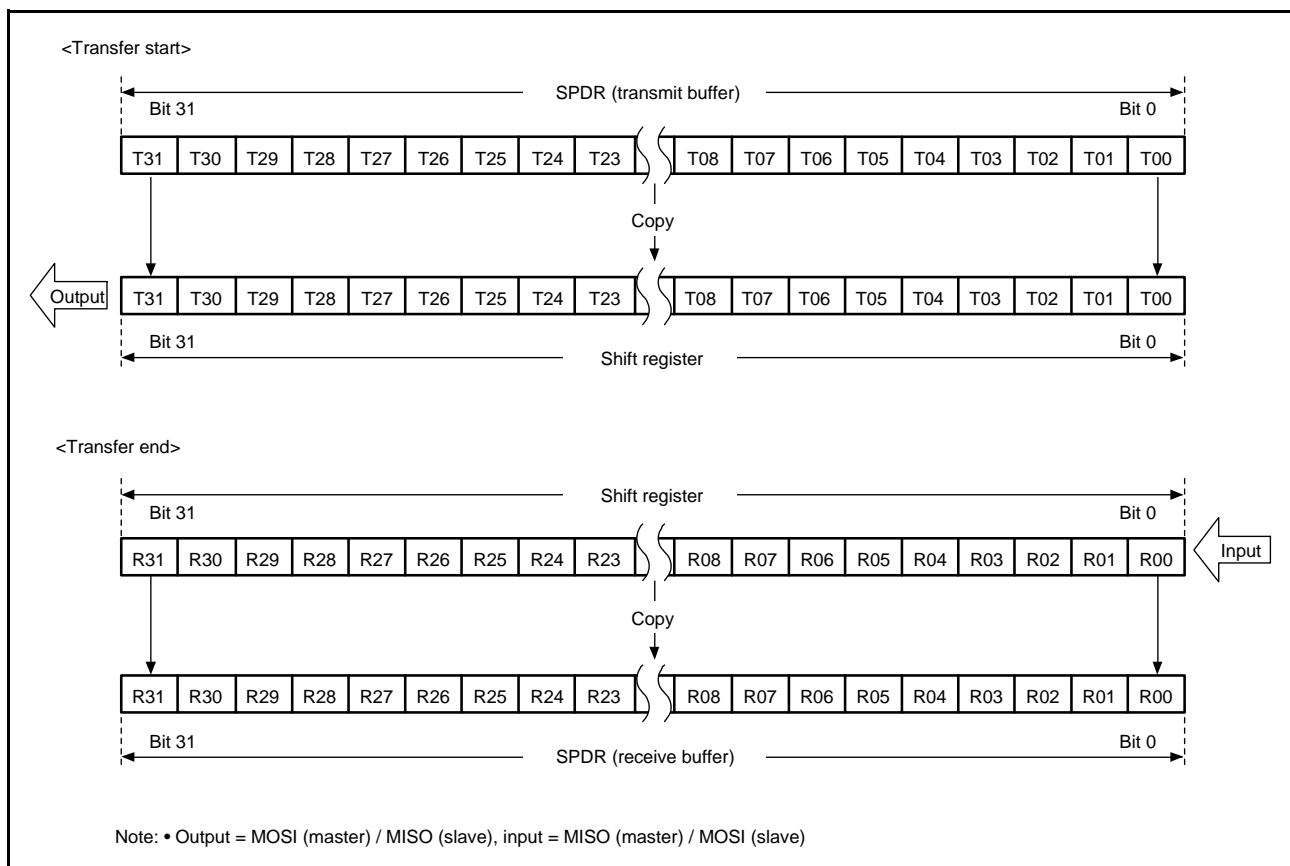


Figure 35.15 MSB-First Transfer (32-Bit Data, Parity Disabled)

(2) MSB-First Transfer (24-Bit Data)

Figure 35.16 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB-first selected.

In transmission, the lower-order 24 bits (T23 to T00) from the current stage of the transmit buffer are copied to the shift register. Data for transmission are shifted out from the shift register in order from T23, through T22, and so on to T00. In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to R00 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. At this time, the higher-order eight bits of the transmit buffer are stored in the higher-order eight bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order eight bits of the receive buffer.

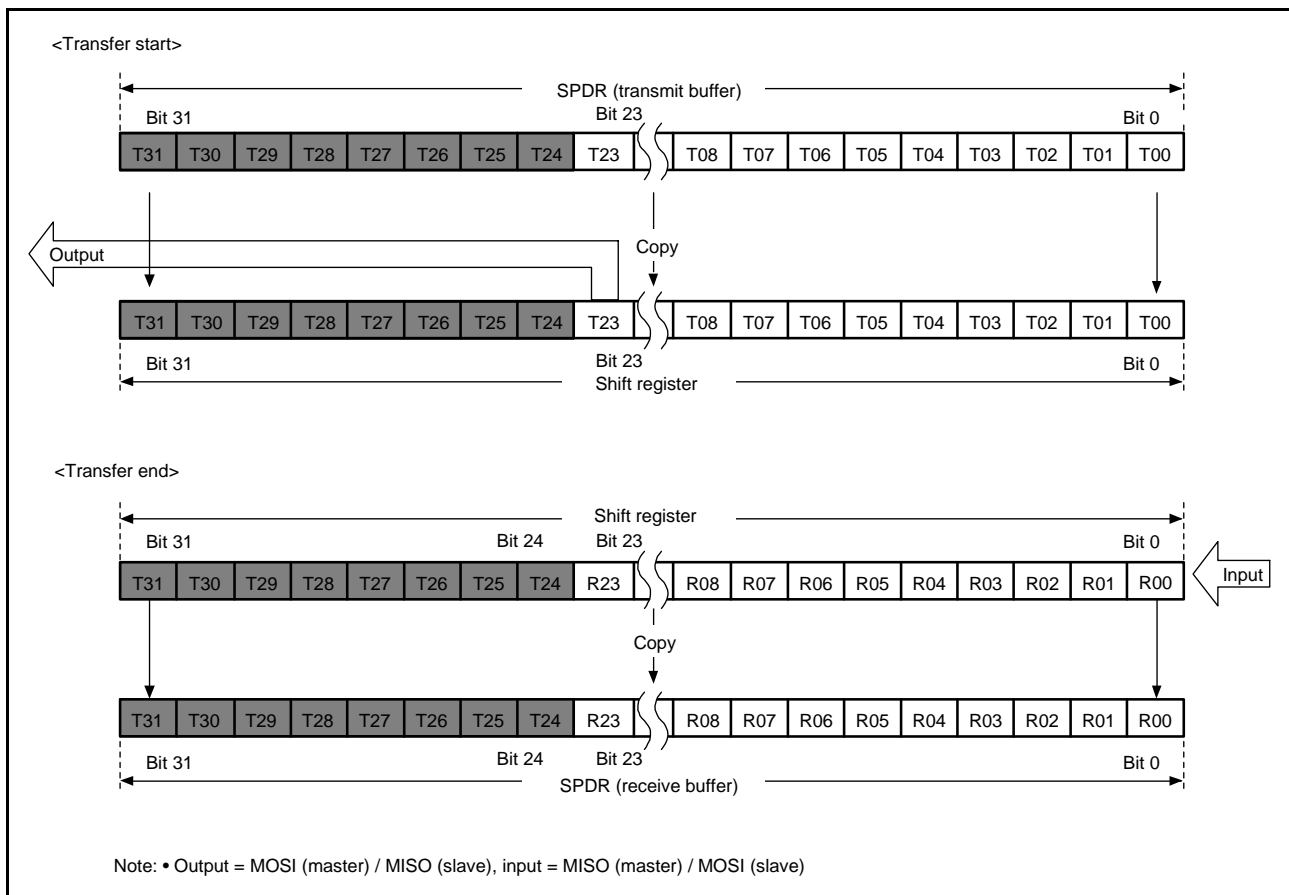


Figure 35.16 MSB-First Transfer (24-Bit Data, Parity Disabled)

(3) LSB-First Transfer (32-Bit Data)

Figure 35.17 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, an RSPI data length of 32 bits, and LSB-first selected.

In transmission, bits T31 to T00 from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T31 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T31.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to R31 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

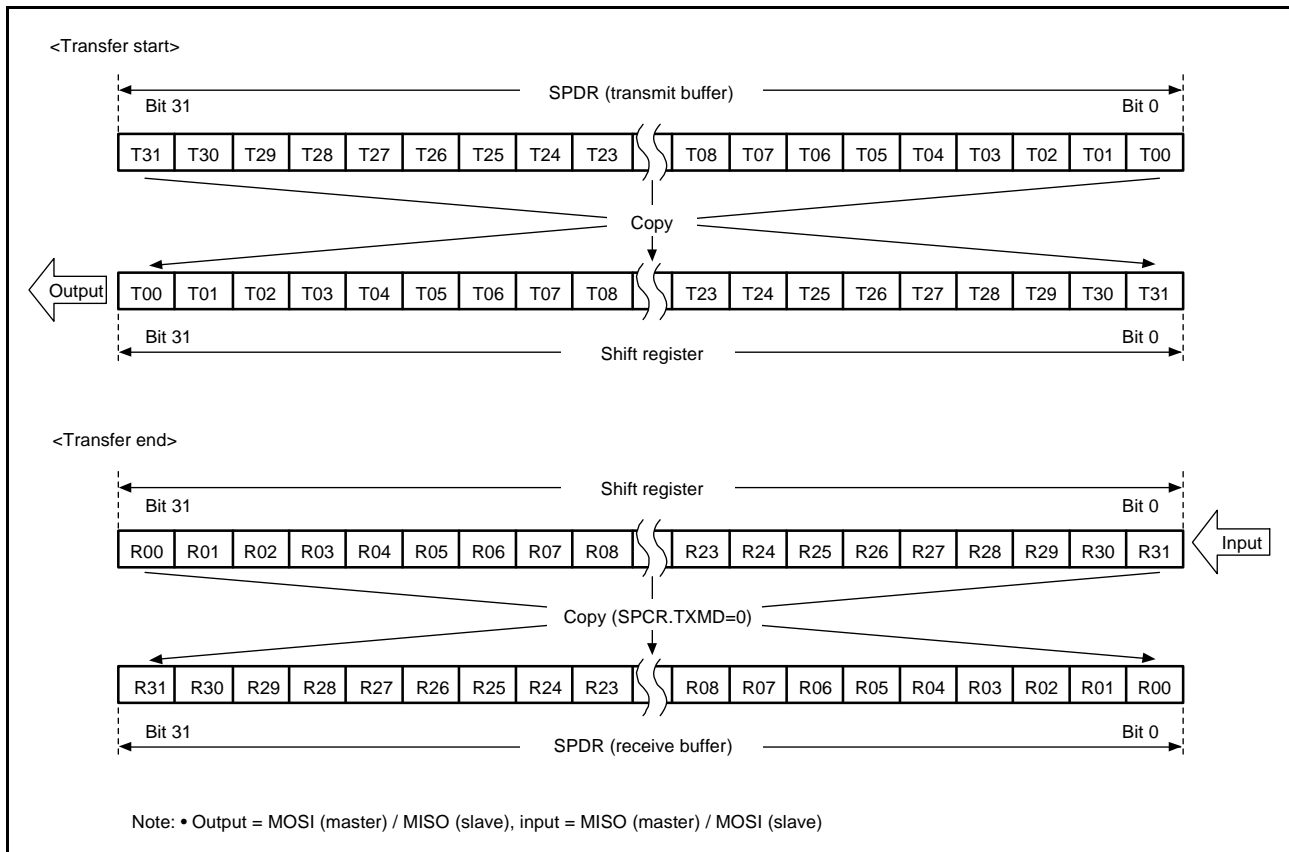


Figure 35.17 LSB-First Transfer (32-Bit Data, Parity Disabled)

(4) LSB-First Transfer (24-Bit Data)

Figure 35.18 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity disabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB-first selected.

In transmission, the lower-order 24 bits (T23 to T00) from the current stage of the transmit buffer are reordered bit by bit to obtain the order T00 to T23 for copying to the shift register. Data for transmission are shifted out from the shift register in order from T00, through T01, and so on to T23.

In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R00 to R23 have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer.

At this time, the higher-order eight bits of the transmit buffer are stored in the higher-order eight bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order eight bits of the receive buffer.

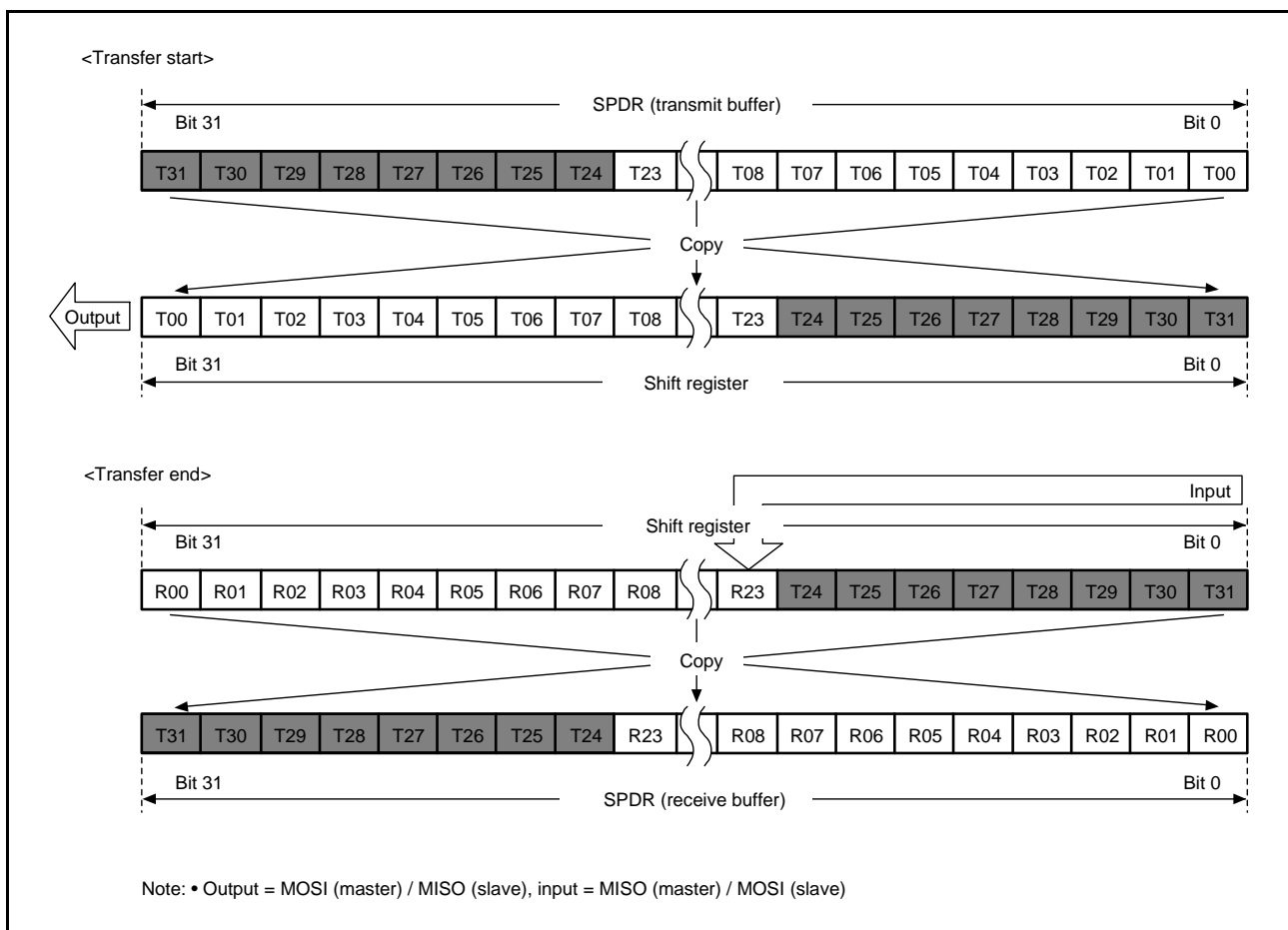


Figure 35.18 LSB-First Transfer (24-Bit Data, Parity Disabled)

### 35.3.4.2 When Parity is Enabled (SPCR2.SPPE = 1)

When parity is enabled, the lowest-order bit of the data for transmission becomes a parity bit. Hardware calculates the value of the parity bit.

#### (1) MSB-First Transfer (32-Bit Data)

Figure 35.19 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and MSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T31 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T31, T30, ..., T01, and P.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R31 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R31 to P are checked by judging the parity.

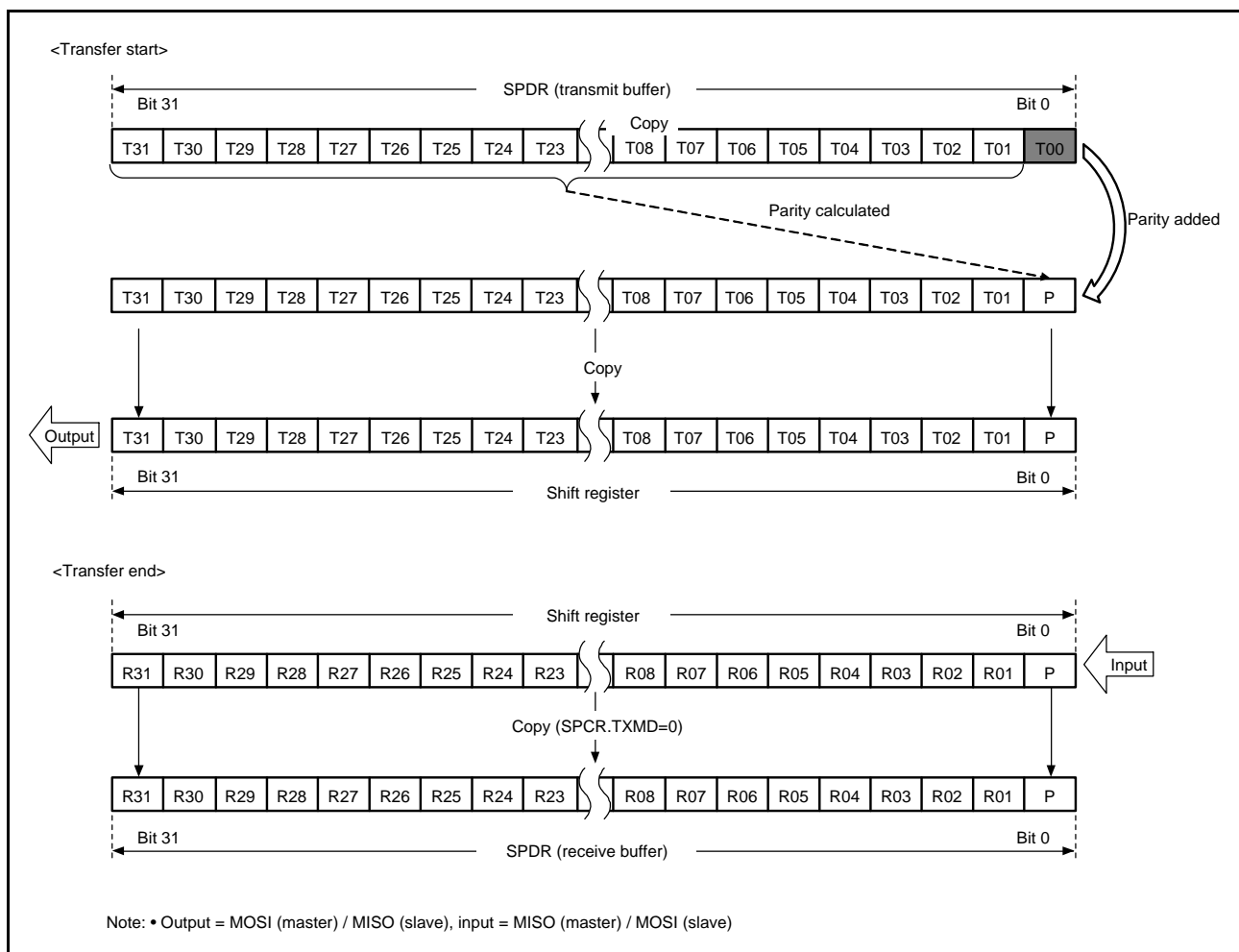


Figure 35.19 MSB-First Transfer (32-Bit Data, Parity Enabled)



(2) MSB-First Transfer (24-Bit Data)

Figure 35.20 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and MSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T23 to T01. This replaces the final bit, T00, and the whole is copied to the shift register. Data are transmitted in the order T23, T22, ..., T01, and P.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R23 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R23 to P are checked by judging the parity. At this time, the higher-order eight bits of the transmit buffer are stored in the higher-order eight bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order eight bits of the receive buffer.

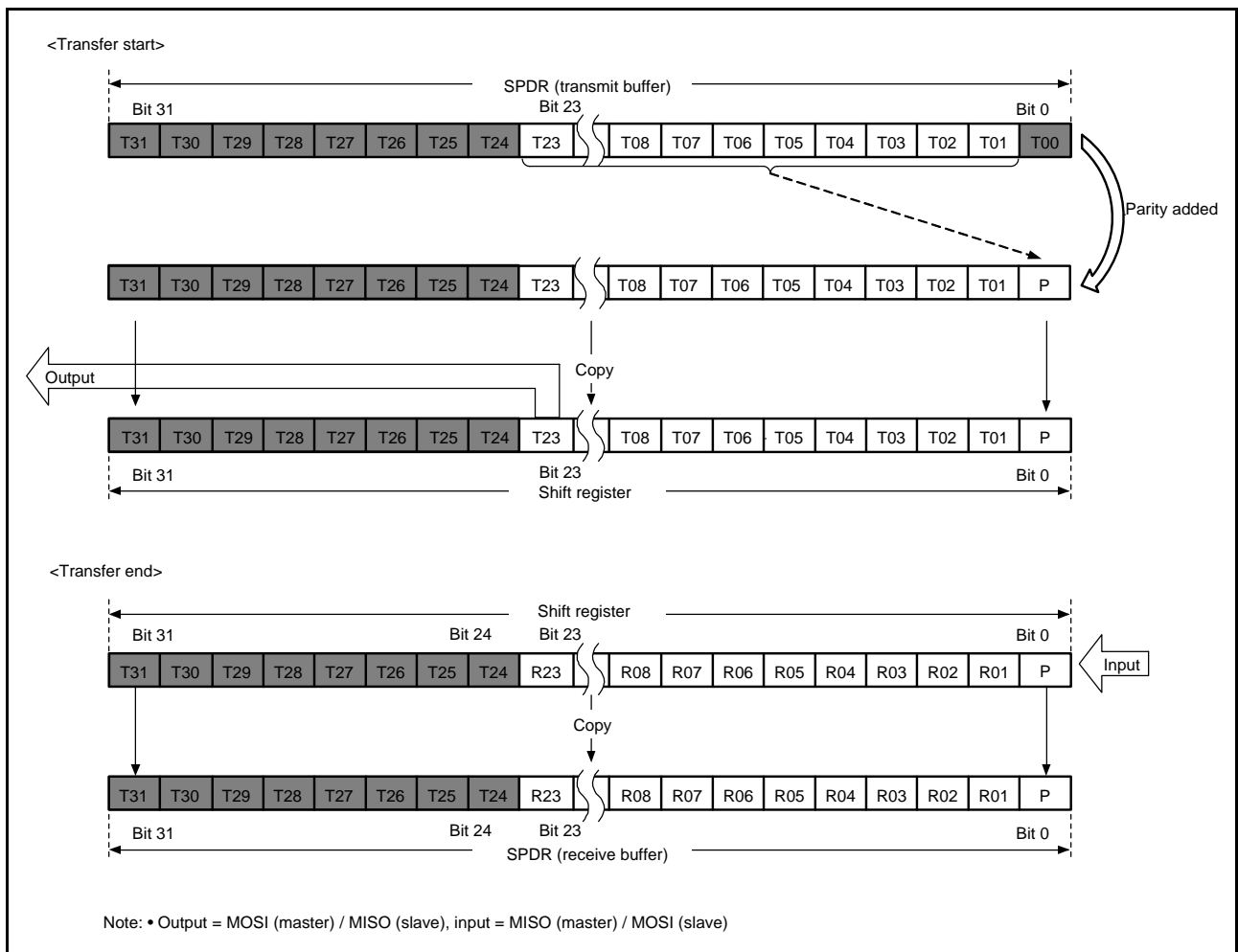


Figure 35.20 MSB-First Transfer (24-Bit Data, Parity Enabled)

(3) LSB-First Transfer (32-Bit Data)

Figure 35.21 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, an RSPI data length of 32 bits, and LSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T30 to T00. This replaces the final bit, T31, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, ..., T30, and P.

In reception, received data are shifted in bit by bit through bit 0 of the shift register. When bits R00 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P are checked by judging the parity.

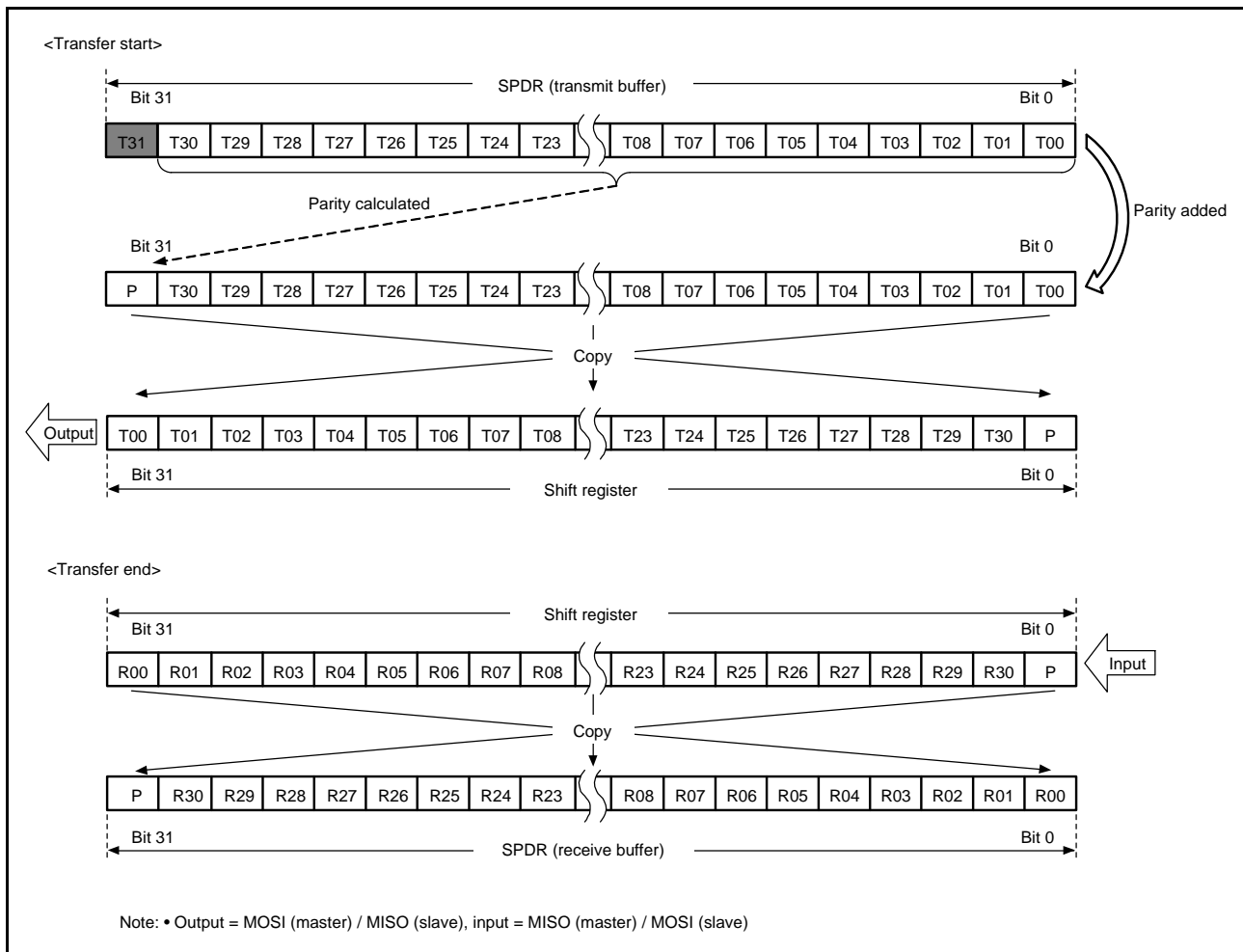


Figure 35.21 LSB-First Transfer (32-Bit Data, Parity Enabled)

(4) LSB First Transfer (24-Bit Data)

Figure 35.22 shows details of operations by the RSPI data register (SPDR) and the shift register in transfer with parity enabled, 24 bits as the RSPI data length for an example that is not 32 bits, and LSB-first selected.

In transmission, the value of the parity bit (P) is calculated from bits T22 to T00. This replaces the final bit, T23, and the whole is copied to the shift register. Data are transmitted in the order T00, T01, ..., T22, and P.

In reception, received data are shifted in bit by bit through bit 8 of the shift register. When bits R23 to P have been collected after input of the required number of cycles of RSPCK, the value in the shift register is copied to the receive buffer. On copying of data to the shift register, the data from R00 to P are checked by judging the parity. At this time, the higher-order eight bits of the transmit buffer are stored in the higher-order eight bits of the receive buffer. Writing 0 to bits T31 to T24 at the time of transmission leads to 0 being inserted in the higher-order eight bits of the receive buffer.

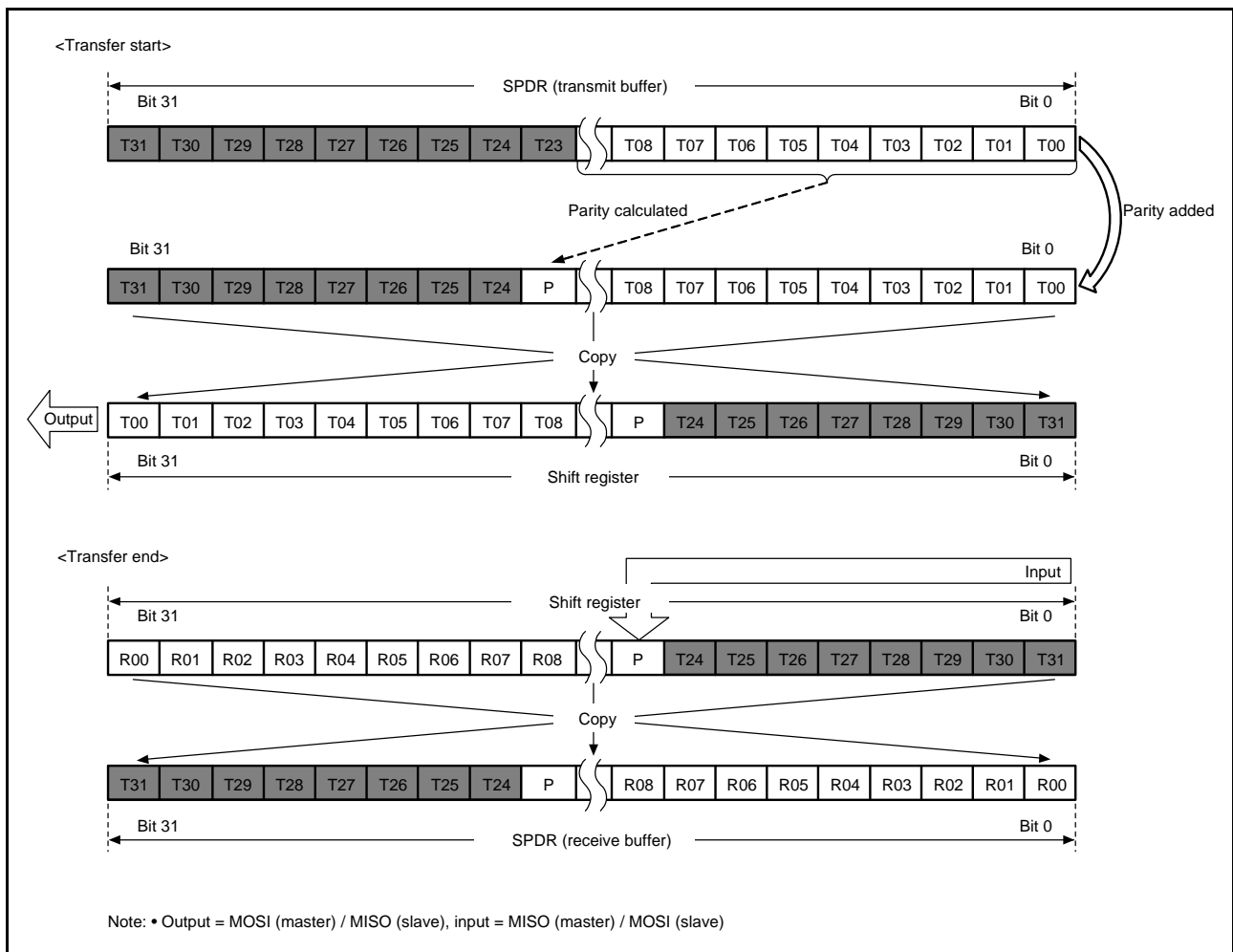


Figure 35.22 LSB-First Transfer (24-Bit Data, Parity Enabled)

### 35.3.5 Transfer Format

#### 35.3.5.1 CPHA = 0

Figure 35.23 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 0. Note that clock synchronous operation (the SPCR.SPMS bit is 1) is not guaranteed when the RSPI operates in slave mode (SPCR.MSTR = 0) and the CPHA bit is 0. In Figure 35.23, RSPCKn (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0; RSPCKn (CPOL = 1) indicates the RSPCKn signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The input/output directions of the signals depend on the RSPI settings. For details, see section 35.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 0, the driving of valid data to the MOSIn and MISO<sub>n</sub> signals commences at an SSL<sub>ni</sub> signal assertion timing. The first RSPCKn signal change timing that occurs after the SSL<sub>ni</sub> signal assertion becomes the first transfer data fetch timing. After this timing, data is sampled at every 1 RSPCK cycle. The change timing for MOSIn and MISO<sub>n</sub> signals is always 1/2 RSPCK cycles after the transfer data fetch timing. The CPOL bit setting does not affect the RSPCK signal operation timing; it only affects the signal polarity.

t1 denotes a period from an SSL<sub>ni</sub> signal assertion to RSPCKn oscillation (RSPCK delay). t2 denotes a period from the termination of RSPCKn oscillation to an SSL<sub>ni</sub> signal negation (SSL negation delay). t3 denotes a period in which SSL<sub>ni</sub> signal assertion is suppressed for the next transfer after the end of serial transfer (next-access delay). t1, t2, and t3 are controlled by a master device running on the RSPI system. For a description of t1, t2, and t3 when the RSPI of this LSI is in master mode, see section 35.3.10.1, Master Mode Operation.

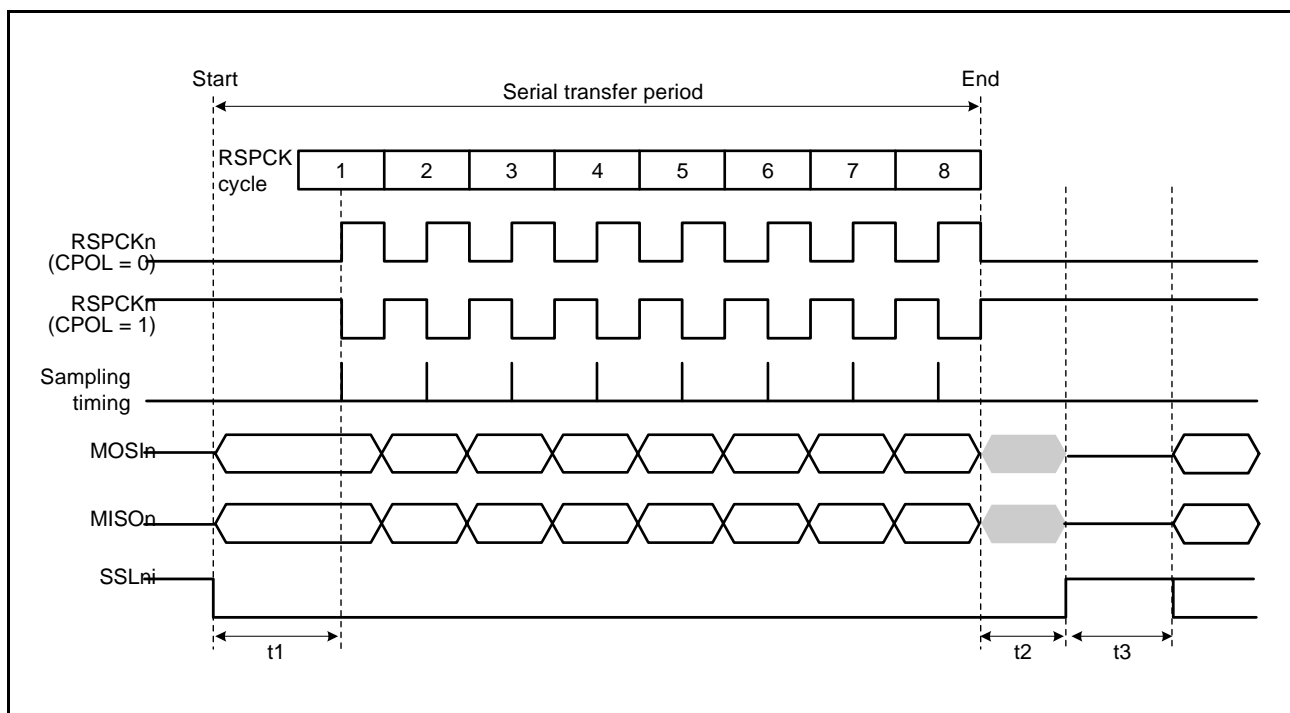


Figure 35.23 RSPI Transfer Format (CPHA = 0)

### 35.3.5.2 CPHA = 1

Figure 35.24 shows a sample transfer format for the serial transfer of 8-bit data when the SPCMDm.CPHA bit is 1. However, when the SPCR.SPMS bit is 1, the SSLni signals are not used, and only the three signals RSPCKn, MOSIn, and MISO<sub>n</sub> handle communications. In Figure 35.24, RSPCK (CPOL = 0) indicates the RSPCKn signal waveform when the SPCMDm.CPOL bit is 0; RSPCK (CPOL = 1) indicates the RSPCKn signal waveform when the CPOL bit is 1. The sampling timing represents the timing at which the RSPI fetches serial transfer data into the shift register. The input/output directions of the signals depend on the RSPI mode (master or slave). For details, see section 35.3.2, Controlling RSPI Pins.

When the SPCMDm.CPHA bit is 1, the driving of invalid data to the MISO<sub>n</sub> signal commences at an SSLni signal assertion timing. The output of valid data to the MOSIn and MISO<sub>n</sub> signals commences at the first RSPCKn signal change timing that occurs after the SSLni signal assertion. After this timing, data is updated at every 1 RSPCK cycle. The transfer data fetch timing is always 1/2 RSPCK cycles after the data update timing. The SPCMDm.CPOL bit setting does not affect the RSPCKn signal operation timing; it only affects the signal polarity.

t1, t2, and t3 are the same as those in the case of CPHA = 0. For a description of t1, t2, and t3 when the RSPI of this LSI is in master mode, see section 35.3.10.1, Master Mode Operation.

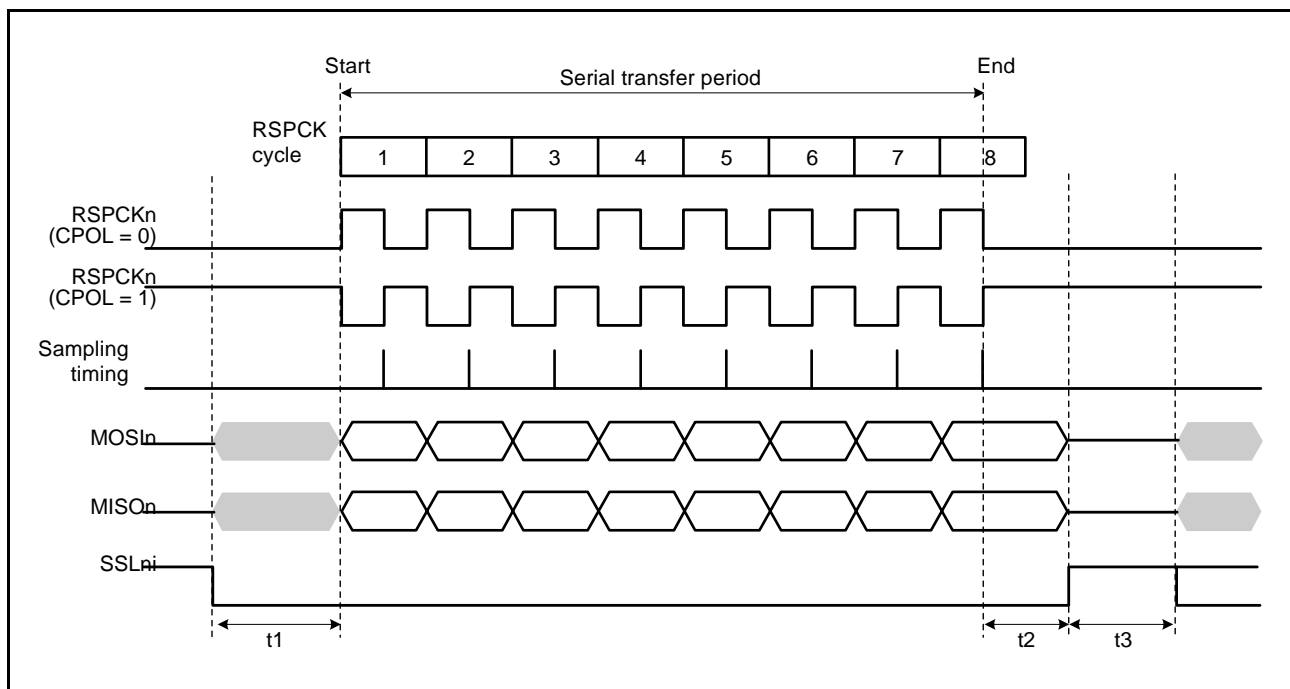


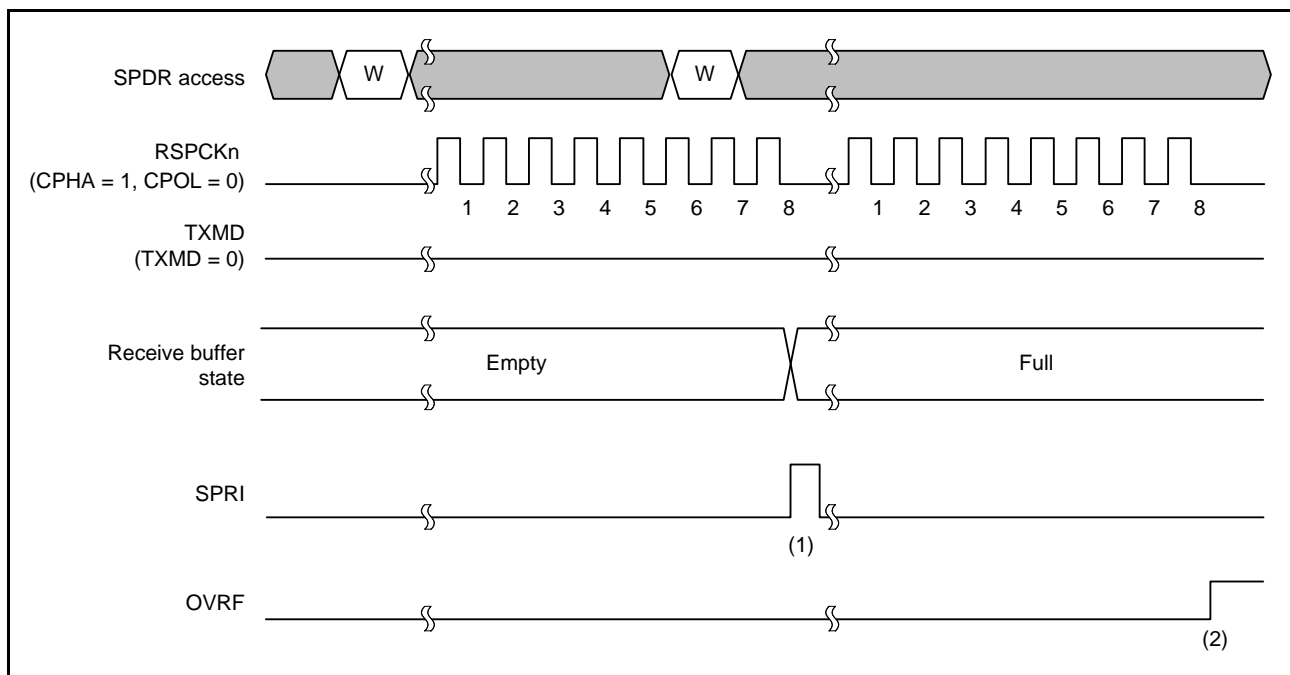
Figure 35.24 RSPI Transfer Format (CPHA = 1)

### 35.3.6 Communications Operating Mode

Full-duplex synchronous serial communications or transmit operations only can be selected by the communications operating mode select bit (SPCR.TXMD). The SPDR access shown in Figure 35.25 and Figure 35.26 indicates the condition of access to the SPDR register, where W denotes a write cycle.

#### 35.3.6.1 Full-Duplex Synchronous Serial Communications (SPCR.TXMD = 0)

Figure 35.25 shows an example of operation when the communications operating mode select bit (SPCR.TXMD) is set to 0. In the example in Figure 35.25, the RSPCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKn waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).



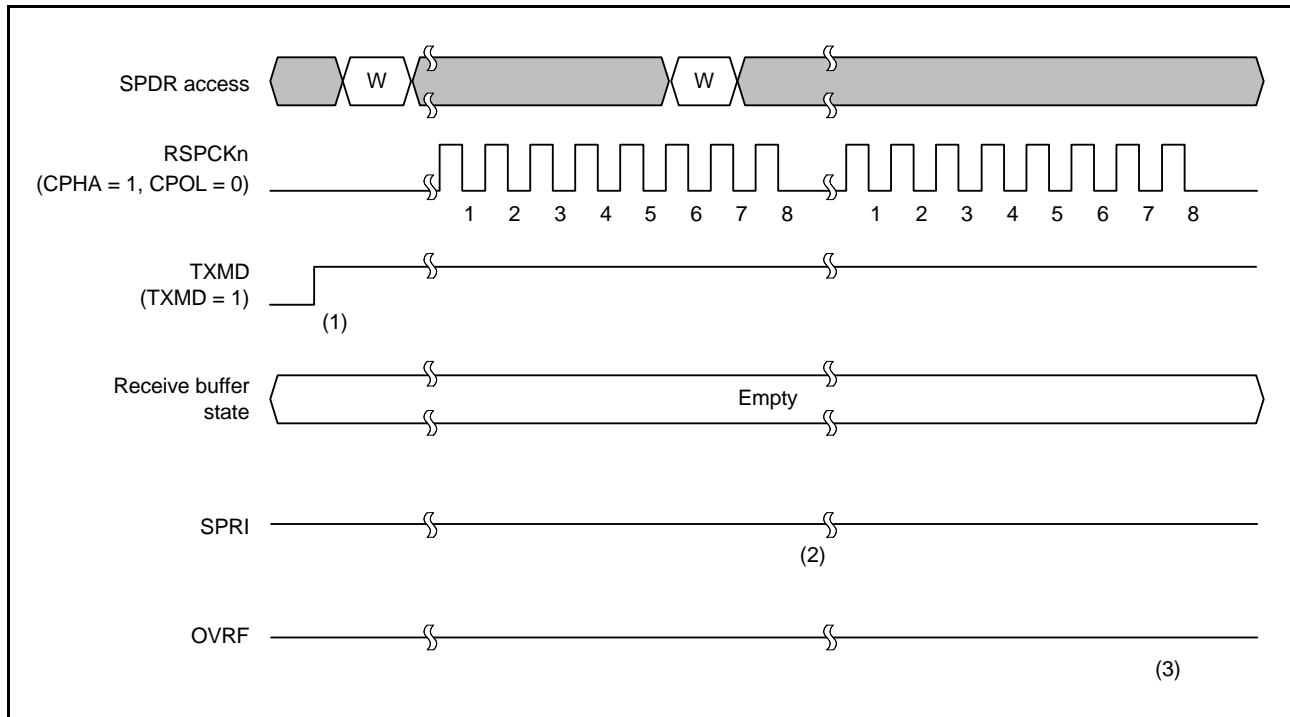
**Figure 35.25 Operation Example of SPCR.TXMD = 0**

The operation of the flags at timings shown in steps (1) and (2) in the figure is described below.

- (1) When a serial transfer ends with the receive buffer of SPDR empty, the RSPI generates a receive buffer full interrupt request (SPRI) and copies the received data in the shift register to the receive buffer.
- (2) When a serial transfer ends with the receive buffer of SPDR holding data that was received in the previous serial transfer, the RSPI sets the SPSR.OVRF flag to 1 and discards the received data in the shift register.

### 35.3.6.2 Transmit Operations Only (SPCR.TXMD = 1)

Figure 35.26 shows an example of operation when the communications operating mode select bit (SPCR.TXMD) is set to 1. In the example in Figure 35.26, the RSPI performs an 8-bit serial transfer in which the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKn waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).



**Figure 35.26** Operation Example of SPCR.TXMD = 1

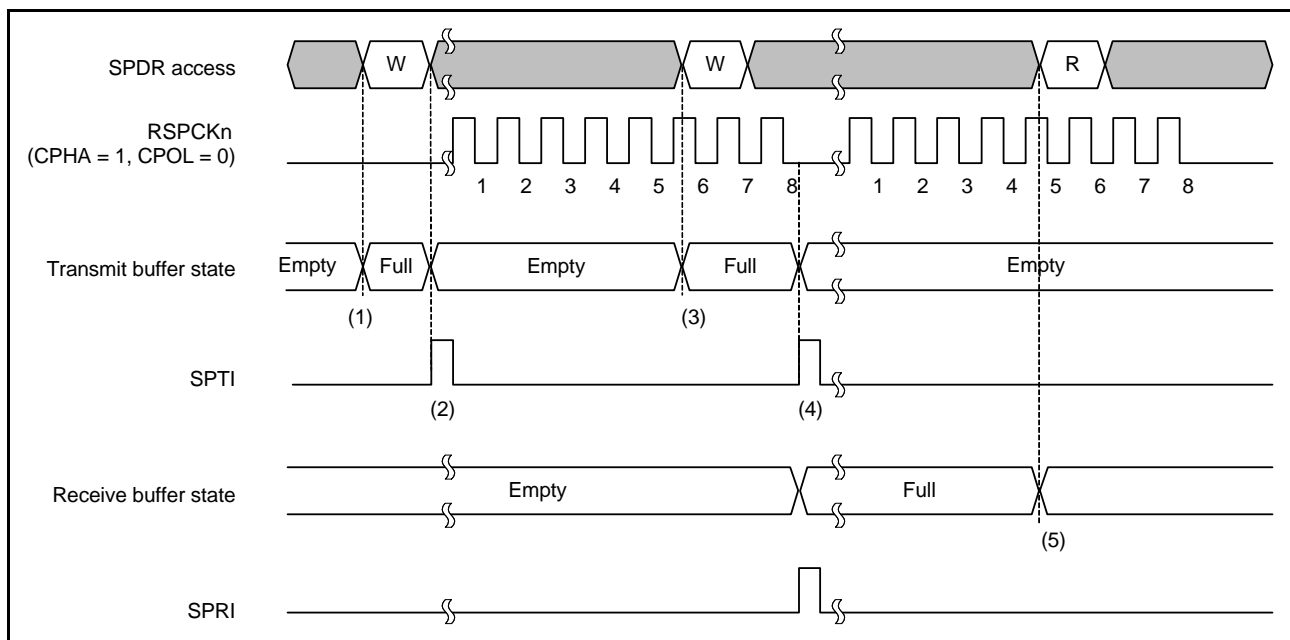
The operation of the flags at timings shown in steps (1) to (3) in the figure is described below.

- (1) Make sure there is no data left in the receive buffer and the SPSR.OVRF flag is 0 before entering the mode of transmit operations only (SPCR.TXMD = 1).
- (2) When a serial transfer ends with the receive buffer of SPDR empty, if the mode of transmit operations only is selected (SPCR.TXMD = 1), the RSPI does not copy the data in the shift register to the receive buffer.
- (3) Since the receive buffer of SPDR does not hold data that was received in the previous serial transfer, even when a serial transfer ends, the SPSR.OVRF flag retains the value of 0, and the data in the shift register is not copied to the receive buffer.

When performing transmit operations only (SPCR.TXMD = 1), the RSPI transmits transmit data but does not receive received data. Therefore, the SPSR.OVRF flag remains cleared to 0 at the timings of (1) to (3).

### 35.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts

Figure 35.27 shows an example of operation of the RSPI transmit buffer empty interrupt (SPTI) and the RSPI receive buffer full interrupt (SPRI). The SPDR register access shown in Figure 35.27 indicates the condition of access to the SPDR register, where W denotes a write cycle, and R a read cycle. In the example in Figure 35.27, the RSPI performs an 8-bit serial transfer in which the SPCR.TXMD bit is 0, the SPDCR.SPFC[1:0] bits are 00b, the SPCMDm.CPHA bit is 1, and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKn waveform represent the number of RSPCKn cycles (i.e., the number of transferred bits).



**Figure 35.27 Operation Example of SPTI and SPRI Interrupts**

The operation of the interrupts at timings shown in steps (1) to (5) in the figure is described below.

1. When transmit data is written to SPDR when the transmit buffer of SPDR is empty (data for the next transfer is not set), the RSPI writes data to the transmit buffer.
2. If the shift register is empty, the RSPI copies the data in the transmit buffer to the shift register and generates a transmit buffer empty interrupt request (SPTI). How a serial transfer is started depends on the mode of the RSPI. For details, see section 35.3.10, SPI Operation, and section 35.3.11, Clock Synchronous Operation.
3. When transmit data is written to SPDR by the transmit buffer empty interrupt routine, the data is transferred to the transmit buffer. Because the data being transferred serially is stored in the shift register, the RSPI does not copy the data in the transmit buffer to the shift register.
4. When the serial transfer ends with the receive buffer of SPDR being empty, the RSPI copies the receive data in the shift register to the receive buffer and generates a receive buffer full interrupt request (SPRI). Since the shift register becomes empty upon completion of serial transfer, when the transmit buffer had been full before the serial transfer ended, the RSPI copies the data in the transmit buffer to the shift register. Even when received data is not copied from the shift register to the receive buffer in an overrun error status, upon completion of the serial transfer, the RSPI determines that the shift register is empty, thus data transfer from the transmit buffer to the shift register is enabled.
5. When SPDR is read by the receive buffer full interrupt routine, the receive data can be read.



If SPDR is written to when the transmit buffer holds data that has not yet been transmitted, the RSPI does not update the data in the transmit buffer. When writing to SPDR, make sure to use a transmit buffer empty interrupt request. To use an RSPI transmit interrupt, set the SPTIE bit in SPCR to 1.

If the RSPI function is disabled (the SPE bit in SPCR being 0), set the SPTIE bit to 0.

When serial transfer ends with the receive buffer being full, the RSPI does not copy data from the shift register to the receive buffer, and detects an overrun error (see section 35.3.8, Error Detection). To prevent a receive data overrun error, read the received data using a receive buffer full interrupt request before the next serial transfer ends. To use an RSPI receive interrupt, set the SPCR.SPRIE bit to 1.

Transmission and reception interrupts or the corresponding IRn.IR flags (where n is the vector number) in ICU can be used to confirm the states of the transmission and reception buffers. See section 15, Interrupt controller (ICUb), for the vector numbers.

### 35.3.8 Error Detection

In the normal RSPI serial transfer, the data written to the transmit buffer of SPDR is serially transmitted, and the serially received data can be read from the receive buffer of SPDR. If access is made to SPDR, depending on the status of the transmit/receive buffer or the status of the RSPI at the beginning or end of serial transfer, in some cases non-normal transfers can be executed.

If a non-normal transfer operation occurs, the RSPI detects the event as an overrun error, parity error, or mode fault error. Table 35.8 lists the relationship between non-normal transfer operations and the RSPI's error detection function.

**Table 35.8 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function**

	Occurrence Condition	RSPI Operation	Error Detection
A	SPDR is written when the transmit buffer is full.	<ul style="list-style-type: none"> <li>The contents of the transmit buffer are kept.</li> <li>Missing write data.</li> </ul>	None
B	Serial transfer is started in slave mode when transmit data is still not loaded on the shift register.	Data received in previous serial transfer is serially transmitted.	None
C	SPDR is read when the receive buffer is empty.	Previously received serial data is output.	None
D	Serial transfer terminates when the receive buffer is full.	<ul style="list-style-type: none"> <li>The contents of the receive buffer are kept.</li> <li>Missing serial receive data.</li> </ul>	Overrun error
E	An incorrect parity bit is received when performing full-duplex synchronous serial communications with the parity function enabled.	The parity error flag is asserted.	Parity error
F	The SSLn0 input signal is asserted when the serial transfer is idle in multi-master mode.	<ul style="list-style-type: none"> <li>Driving of the RSPCKn, MOSIn, SSLn1 to SSLn3 output signals is stopped.</li> <li>RSPI function is disabled.</li> </ul>	Mode fault error
G	The SSLn0 input signal is asserted during serial transfer in multi-master mode.	<ul style="list-style-type: none"> <li>Serial transfer is suspended.</li> <li>Missing transmit/receive data.</li> <li>Driving of the RSPCKn, MOSIn, SSLn1 to SSLn3 output signals is stopped.</li> <li>RSPI function is disabled.</li> </ul>	Mode fault error
H	The SSLn0 input signal is negated during serial transfer in slave mode.	<ul style="list-style-type: none"> <li>Serial transfer is suspended.</li> <li>Missing transmit/receive data.</li> <li>Driving of the MISO output signal is stopped.</li> <li>RSPI function is disabled.</li> </ul>	Mode fault error

On operation A described in Table 35.8, the RSPI does not detect an error. To prevent data omission during the writing to SPDR, write operations to SPDR should be executed using a transmit interrupt request.

Likewise, the RSPI does not detect an error on operation B. In a serial transfer that was started before the shift register was updated, the RSPI sends the data that was received in the previous serial transfer, and does not treat the operation indicated in B as an error. Note that the received data from the previous serial transfer is retained in the receive buffer of SPDR, thus it can be correctly read (if SPDR is not read before the end of the serial transfer, an overrun error may occur). Similarly, the RSPI does not detect an error on operation C. To prevent extraneous data from being read, SPDR read operation should be executed using a receive interrupt request.

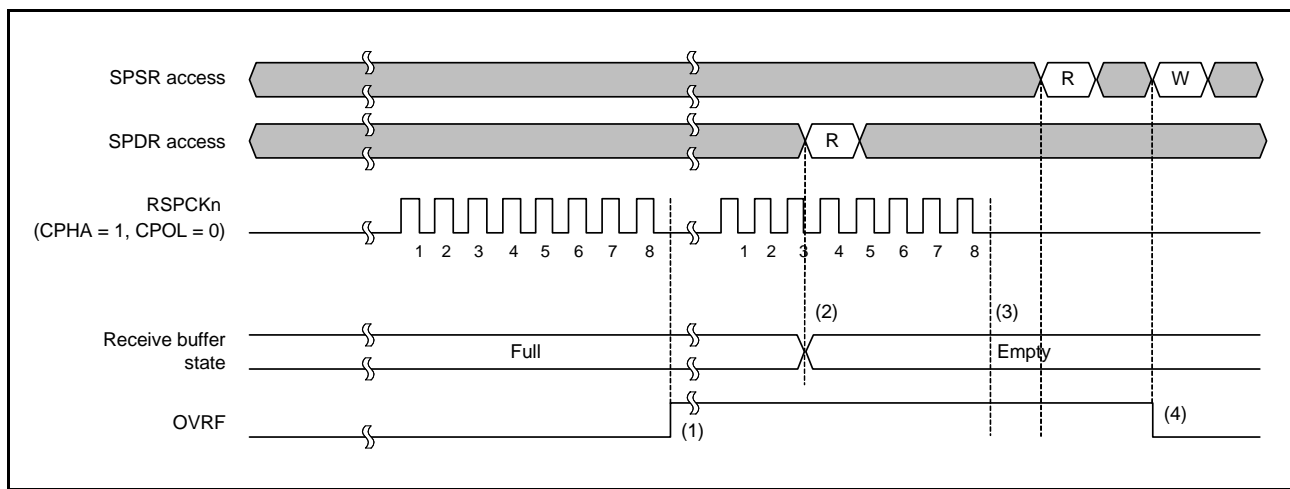
An overrun error shown in D is described in section 35.3.8.1, **Overrun Error**. A parity error shown in E is described in section 35.3.8.2, **Parity Error**. A mode fault error shown in F to H is described in section 35.3.8.3, **Mode Fault Error**.

For the transmit and receive interrupts, refer to section 35.3.7, **Transmit Buffer Empty/Receive Buffer Full Interrupts**.

### 35.3.8.1 Overrun Error

If a serial transfer ends when the receive buffer of SPDR is full, the RSPI detects an overrun error, and sets the OVRF flag in SPSR to 1. When the OVRF flag is 1, the RSPI does not copy data from the shift register to the receive buffer so that the data prior to the occurrence of the error is retained in the receive buffer. To set the OVRF flag to 0, write 0 to the OVRF flag after the CPU has read SPSR with the OVRF flag set to 1.

Figure 35.28 shows an example of operation of the OVRF flag. The SPSR and SPDR accesses shown in Figure 35.28 indicate the condition of accesses to SPSR and SPDR, respectively, where W denotes a write cycle, and R a read cycle. In the example in Figure 35.28, the RSPI performs an 8-bit serial transfer in which the SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKn waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).



**Figure 35.28** Operation Example of OVRF Flag

The operation of the flags at the timing shown in steps (1) to (4) in the figure is described below.

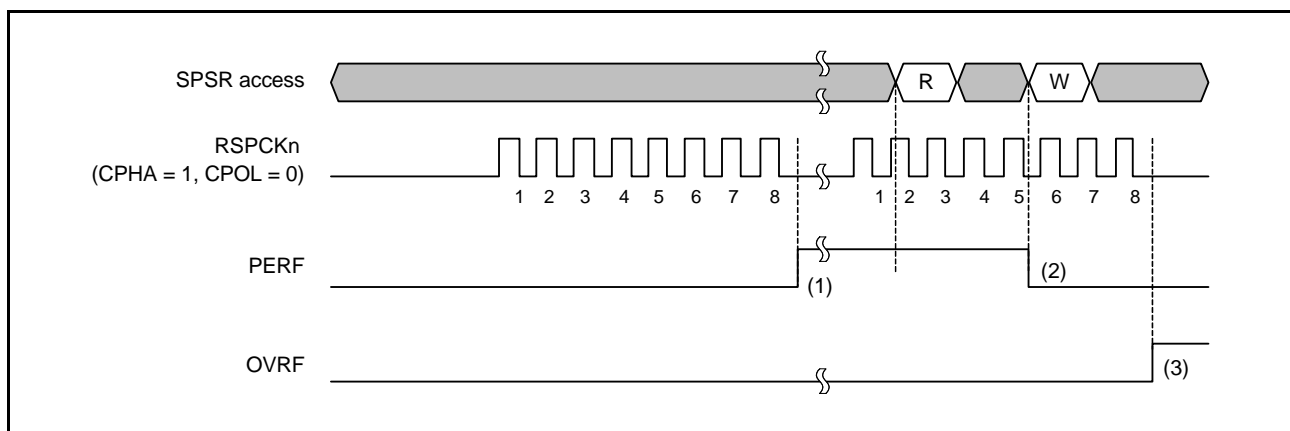
1. If a serial transfer terminates with the receive buffer full, the RSPI detects an overrun error, and sets the OVRF flag to 1. The RSPI does not copy the data in the shift register to the receive buffer. Even if the SPPE bit is 1, parity errors are not detected. In master mode, the RSPI copies the pointer value to SPCMDm register to the SPSSR.SPECM[2:0] bits.
2. When SPDR is read, the RSPI outputs the data in the receive buffer can be read. The receive buffer becoming empty does not clear the OVRF flag.
3. If the serial transfer ends with the OVRF flag being 1 (an overrun error), the RSPI does not copy the data in the shift register to the receive buffer. A reception-buffer interrupt is not generated. Even if the SPPE bit is 1, parity errors are not detected. When in master mode, the RSPI does not update the SPSSR.SPECM[2:0] bits. When in an overrun error state and the RSPI does not copy the received data from the shift register to the receive buffer, upon termination of the serial transfer, the RSPI determines that the shift register is empty; in this manner, data transfer from the transmit buffer to the shift register is enabled.
4. If the value 0 is written to the OVRF flag after SPSR is read when the OVRF flag is 1, OVRF flag is cleared to 0.

The occurrence of an overrun can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. When executing a serial transfer, measures should be taken to ensure the early detection of overrun errors, such as reading SPSR immediately after SPDR is read. When the RSPI is used in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

If an overrun error occurs and the OVRF flag is set to 1, normal reception operations cannot be performed until the OVRF flag is cleared.

### 35.3.8.2 Parity Error

If full-duplex synchronous serial communications is performed with the SPCR.TXMD bit cleared to 0 and the SPCR2.SPPE bit set to 1, when serial transfer ends, the RSPI checks whether there are parity errors. Upon detecting a parity error in the received data, the RSPI sets the SPSR.PERF flag to 1. Since the RSPI does not copy the data in the shift register to the receive buffer when the SPSR.OVRF flag is set to 1, parity error detection is not performed for the received data. To set the PERF flag to 0, write 0 to the PERF flag after SPSR register is read with the PERF flag set to 1. Figure 35.29 shows an example of operation of the OVRF and PERF flags. The SPSR access shown in Figure 35.29 indicates the condition of access to SPSR register, where W denotes a write cycle, and R a read cycle. In the example of Figure 35.29, full-duplex synchronous serial communications is performed while the SPCR.TXMD bit is 0 and the SPCR2.SPPE bit is 1. The RSPI performs an 8-bit serial transfer in which the SPCMDm.CPHA bit is 1 and the SPCMDm.CPOL bit is 0. The numbers given under the RSPCKn waveform represent the number of RSPCK cycles (i.e., the number of transferred bits).



**Figure 35.29** Operation Example of PERF Flag

The operation of the flags at the timing shown in steps (1) to (3) in the figure is described below.

1. If a serial transfer terminates with the RSPI not detecting an overrun error, the RSPI copies the data in the shift register to the receive buffer. The RSPI judges the received data at this timing, and sets the PERF flag to 1 if a parity error is detected. In master mode, the RSPI copies the pointer value to SPCMDm register to the SPSSR.SPECM[2:0] bits.
2. If the value 0 is written to the PERF flag after SPSR register is read when the PERF flag is 1, the PERF flag is cleared to 0.
3. When the RSPI detects an overrun error and serial transfer is terminated, the data in the shift register is not copied to the receive buffer. The RSPI does not perform parity error detection at this timing.

The occurrence of a parity error can be checked either by reading SPSR register or by using an RSPI error interrupt and reading SPSR register. When executing a serial transfer, measures should be taken to ensure the early detection of parity errors, such as reading SPSR. When the RSPI is used in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

### 35.3.8.3 Mode Fault Error

The RSPI operates in multi-master mode when the SPCR.MSTR bit is 1, the SPCR.SPMS bit is 0, and the SPCR.MODFEN bit is 1. If the active level is input with respect to the SSLn0 input signal of the RSPI in multi-master mode, the RSPI detects a mode fault error irrespective of the status of the serial transfer, and sets the MODF bit in the RSPI status register (SPSR) to 1. Upon detecting the mode fault error, the RSPI copies the value of the pointer to SPCMDm to the SPSSR.SPECM[2:0] bits. The active level of the SSLn0 signal is determined by the SSL0P bit in the RSPI slave select polarity register (SSLP).

When the MSTR bit is 0, the RSPI operates in slave mode. The RSPI detects a mode fault error if the MODFEN bit in the RSPI in slave mode is 1, and the SPMS bit is 0, and if the SSLn0 input signal is negated during the serial transfer period (from the time the driving of valid data is started to the time the final valid data is fetched).

Upon detecting a mode fault error, the RSPI stops driving of the output signals and clears the SPCR.SPE bit to 0 (see section 35.3.9, Initializing RSPI). In the case of multi-master configuration, detection of a mode fault error is used to stop driving of the output signals and the RSPI function, which allows the master right to be released.

The occurrence of a mode fault error can be checked either by reading SPSR or by using an RSPI error interrupt and reading SPSR. Detecting mode-fault errors without utilizing the RSPI error interrupt requires polling of SPSR. When using the RSPI in master mode, the pointer value to SPCMDm register at the occurrence of the error can be checked by reading the SPSSR.SPECM[2:0] bits.

When the MODF bit is 1, writing of the value 1 to the SPE bit is ignored by the RSPI. To enable the RSPI function after the detection of a mode fault error, the MODF bit must be set to 0.

### 35.3.9 Initializing RSPI

If the value 0 is written to the SPE bit in the RSPI control register (SPCR) or the RSPI clears the SPE bit to 0 because of the detection of a mode fault error, the RSPI disables the RSPI function, and initializes some of the module functions. When a system reset is generated, the RSPI initializes all of the module functions. The following describes initialization by the clearing of the SPE bit in SPCR and initialization by a system reset.

#### 35.3.9.1 Initialization by Clearing the SPE Bit

When the SPE bit in SPCR is cleared, the RSPI performs the following initialization:

- Suspending any serial transfer that is being executed
- Stopping the driving of output signals (Hi-Z) in slave mode
- Initializing the internal state of the RSPI
- Initializing the transmit buffer of the RSPI

Initialization by the clearing of the SPE bit does not initialize the control bits of the RSPI. For this reason, the RSPI can be started in the same transfer mode as prior to the initialization if the SPE bit is set to 1 again.

The OVRF and MODF flags in SPSR are not initialized, nor is the value of the RSPI sequence status register (SPSSR) initialized. For this reason, even after the RSPI is initialized, data from the receive buffer can be read in order to check the status of error occurrence during an RSPI transfer.

The transmit buffer is initialized to an empty state. Therefore, if the SPTIE bit in SPCR is set to 1 after RSPI initialization, an RSPI transmit interrupt is generated. When the RSPI is initialized by the CPU, in order to disable any RSPI transmit interrupt, the value 0 should be written to the SPTIE bit simultaneously with the writing of the value 0 to the SPE bit. To disable any RSPI transmit interrupt after a mode fault error is detected, use an error handling routine to write the value 0 to the SPTIE bit.

#### 35.3.9.2 System Reset

The initialization by a system reset completely initializes the RSPI through the initialization of all bits for controlling the RSPI, initialization of the status bits, and initialization of data registers, in addition to the requirements described in section 35.3.9.1, Initialization by Clearing the SPE Bit.

## 35.3.10 SPI Operation

### 35.3.10.1 Master Mode Operation

The only difference between single-master mode operation and multi-master mode operation lies in mode fault error detection (see section 35.3.8, Error Detection). When operating in single-master mode, the RSPI does not detect mode fault errors whereas the RSPI running in multi-master mode does detect mode fault errors. This section explains operations that are common to single-master mode and multi-master mode.

#### (1) Starting a Serial Transfer

The RSPI updates the data in the transmit buffer (SPTX) when data is written to the RSPI data register (SPDR) with the RSPI transmit buffer being empty (data for the next transfer is not set). When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR, the RSPI copies data from the transmission buffer to the shift register and starts serial transfer. Upon copying transmit data to the shift register, the RSPI changes the status of the shift register to “full”, and upon termination of serial transfer, it changes the status of the shift register to “empty”. The status of the shift register cannot be referenced.

For details on the RSPI transfer format, see section 35.3.5, Transfer Format. The polarity of the SSLn<sub>i</sub> output pins depends on the SSLP register settings.

#### (2) Terminating a Serial Transfer

Irrespective of the CPHA bit in the RSPI command register (SPCMD<sub>m</sub>), the RSPI terminates the serial transfer after transmitting an RSPCK<sub>n</sub> edge corresponding to the final sampling timing. If free space is available in the receive buffer (SPRX), upon termination of serial transfer, the RSPI copies data from the shift register to the receive buffer of the RSPI data register (SPDR).

It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the SPCMD<sub>m</sub>.SPB[3:0] bit setting. The polarity of the SSLn<sub>i</sub> output pin depends on the SSLP register settings.

For details on the RSPI transfer format, see section 35.3.5, Transfer Format.

#### (3) Sequence Control

The transfer format that is employed in master mode is determined by SPSCR, SPCMD<sub>m</sub>, SPBR, SPCKD, SSLND, and SPND registers.

SPSCR is a register used to determine the sequence configuration for serial transfers that are executed by the RSPI in master mode. The following items are set in SPCMD<sub>m</sub> register: SSLn<sub>i</sub> pin output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCK polarity/phase, whether SPCKD is to be referenced, whether SSLND is to be referenced, and whether SPND is to be referenced. SPBR holds some of the bit rate settings; SPCKD, an RSPI clock delay value; SSLND, an SSL negation delay; and SPND, a next-access delay value.

According to the sequence length that is assigned to SPSCR, the RSPI makes up a sequence comprised of a part or all of SPCMD<sub>m</sub> register. The RSPI contains a pointer to the SPCMD<sub>m</sub> register that makes up the sequence. The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPCR.SPE bit is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in SPCMD<sub>0</sub>, and incorporates the SPCMD<sub>0</sub> settings into the transfer format at the beginning of serial transfer. The RSPI increments the pointer each time the next-access delay period for a data transfer ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, the RSPI sets the pointer in SPCMD<sub>0</sub>, and in this manner the sequence is executed repeatedly.

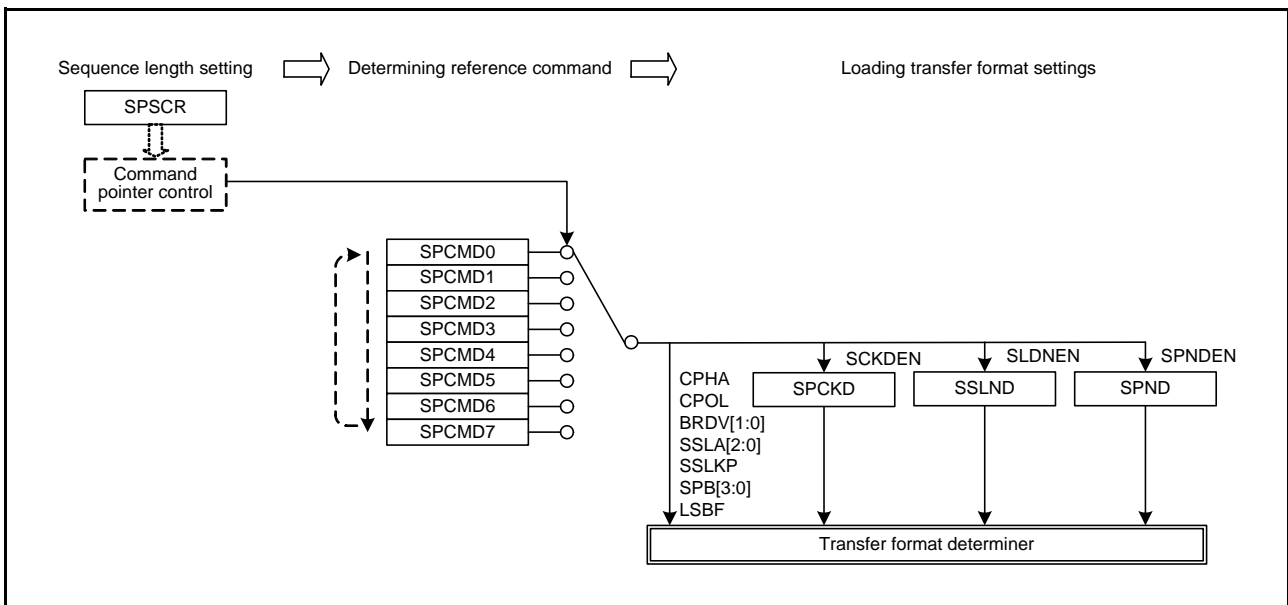


Figure 35.30 Procedure for Determining the Form of Serial Transmission in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

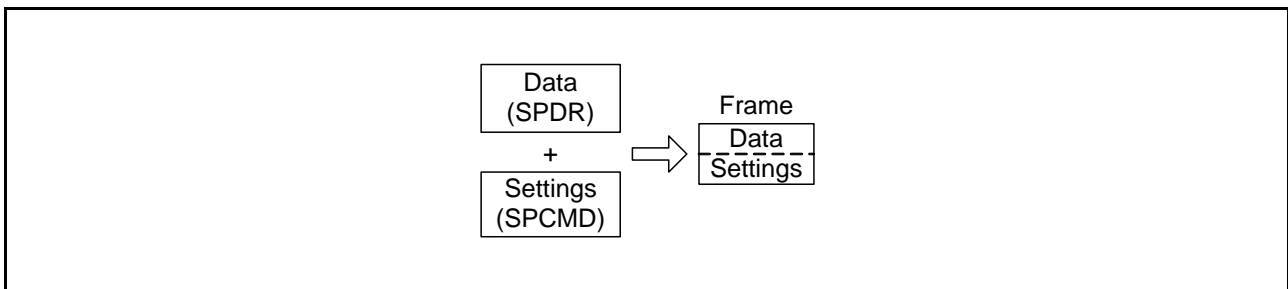
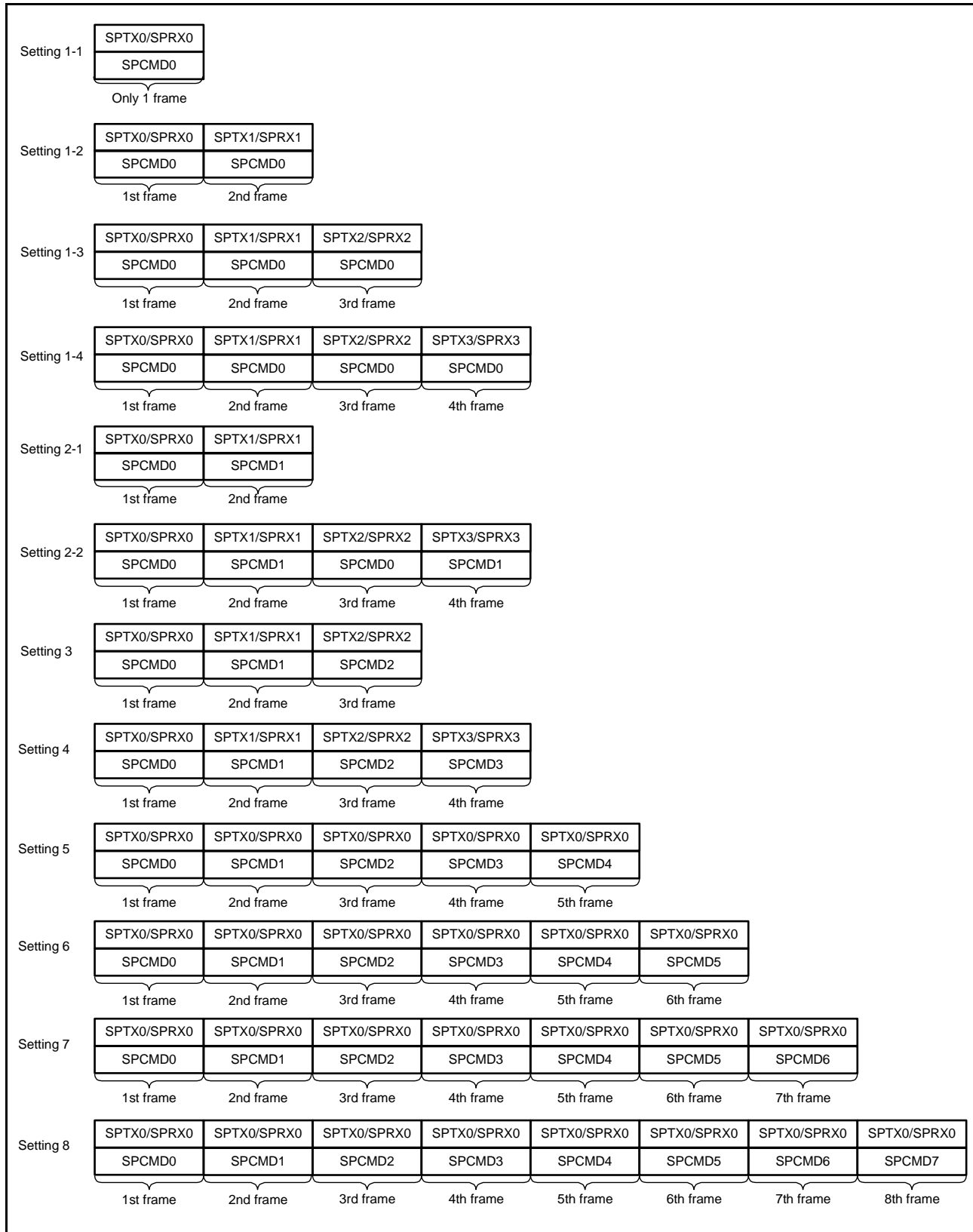


Figure 35.31 Concept of a Frame



Figure 35.32 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 35.4.

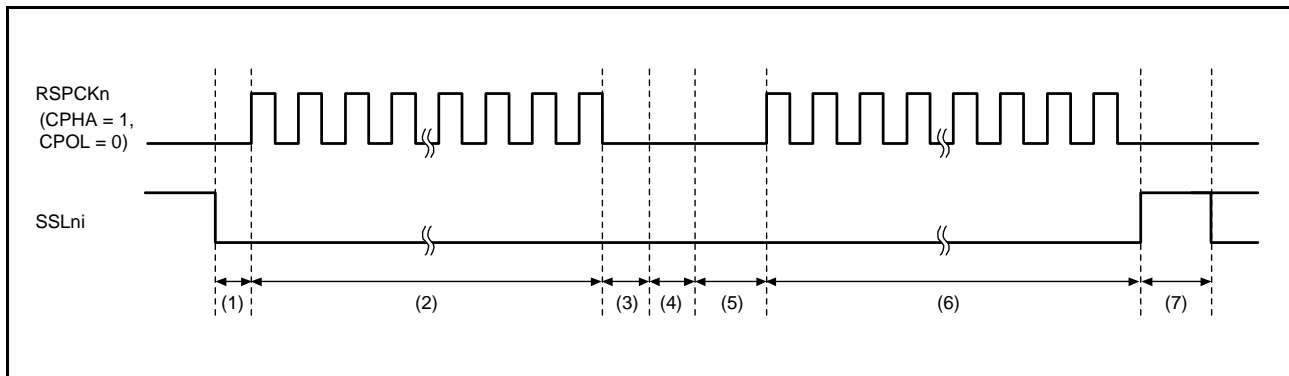


**Figure 35.32 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations**

#### (4) Burst Transfer

If the SSLKP bit in the RSPI command register (SPCMDm) that the RSPI references during the current serial transfer is 1, the RSPI keeps the SSLni signal level during the serial transfer until the beginning of the SSLni signal assertion for the next serial transfer. If the SSLni signal level for the next serial transfer is the same as the SSLni signal level for the current serial transfer, the RSPI can execute continuous serial transfers while keeping the SSLni signal assertion status (burst transfer).

Figure 35.33 shows an example of an SSLni signal operation for the case where a burst transfer is implemented using SPCMD0 and SPCMD1 register settings. The text below explains the RSPI operations (1) to (7) as shown in Figure 35.33. It should be noted that the polarity of the SSLni output signal depends on the SSLP register settings.



**Figure 35.33 Example of Burst Transfer Operation using SSLKP Bit**

- (1) Based on SPCMD0, the RSPI asserts the SSLni signal and inserts RSPCK delays.
- (2) The RSPI executes serial transfers according to SPCMD0.
- (3) The RSPI inserts SSL negation delays.
- (4) Since the SPCMD0.SSLKP bit is 1, the RSPI keeps the SSLni signal value on SPCMD0. This period is sustained, at the shortest, for a period equal to the next-access delay of SPCMD0. If the shift register is empty after the passage of a minimum period, this period is sustained until the transmit data is stored in the shift register for the next transfer.
- (5) Based on SPCMD1, the RSPI asserts the SSLni signal and inserts RSPCK delays.
- (6) The RSPI executes serial transfers according to SPCMD1.
- (7) Because the SPCMD1.SSLKP bit is 0, the RSPI negates the SSLni signal. In addition, a next-access delay is inserted according to SPCMD1.

If the SSLni signal output settings in the SPCMDm register in which 1 is assigned to the SSLKP bit are different from the SSLni signal output settings in the SPCMDm register to be used in the next transfer, the RSPI switches the SSLni signal status to SSLni signal assertion ((5) in Figure 35.33) corresponding to the command for the next transfer. Note that if such an SSLni signal switching occurs, the slaves that drive the MISOn signal compete, and collision of signal levels may occur.

The RSPI in master mode references the SSLni signal operation within the module for the case where the SSLKP bit is not used. Even when the SPCMDm.CPHA bit is 0, the RSPI can accurately start serial transfers by using the SSLni signal assertion for the next transfer that is detected internally. For this reason, burst transfers in master mode can be executed irrespective of CPHA bit settings (see section 35.3.10, SPI Operation).

**(5) RSPCK Delay (t1)**

The RSPCK delay value of the RSPI in master mode depends on the SPCMDm.SCKDEN bit setting and the SPCKD register setting. The RSPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines an RSPCK delay value during serial transfer by using the SPCMDm.SCKDEN bit and SPCKD, as listed in Table 35.9. For a definition of RSPCK delay, see section 35.3.5, Transfer Format.

**Table 35.9 Relationship among SCKDEN Bit, SPCKD, and RSPCK Delay Value**

SPCMDm.SCKDEN Bit	SPCKD.SCKDL[2:0] Bits	RSPCK Delay Value
0	000 to 111	1 RSPCK
1	000	1 RSPCK
	001	2 RSPCK
	010	3 RSPCK
	011	4 RSPCK
	100	5 RSPCK
	101	6 RSPCK
	110	7 RSPCK
	111	8 RSPCK

**(6) SSL Negation Delay (t2)**

The SSL negation delay value of the RSPI in master mode depends on the SPCMDm.SLN DEN bit setting and the SSLND register setting. The RSPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines an SSL negation delay value during serial transfer by using the SPCMDm.SLN DEN bit and SSLND, as listed in Table 35.10. For a definition of SSL negation delay, see section 35.3.5, Transfer Format.

**Table 35.10 Relationship among SLNDEN Bit, SSLND, and SSL Negation Delay Value**

SPCMDm.SLN DEN Bit	SSLND.SLNDL[2:0] Bits	SSL Negation Delay Value
0	000 to 111	1 RSPCK
1	000	1 RSPCK
	001	2 RSPCK
	010	3 RSPCK
	011	4 RSPCK
	100	5 RSPCK
	101	6 RSPCK
	110	7 RSPCK
	111	8 RSPCK

**(7) Next-Access Delay (t3)**

The next-access delay value of the RSPI in master mode depends on the SPCMDm.SPNDEN bit setting and the SPND setting. The RSPI determines the SPCMDm register to be referenced during serial transfer by pointer control, and determines a next-access delay value during serial transfer by using the SPCMDm.SPNDEN bit and SPND, as listed in Table 35.11. For a definition of next-access delay, see section 35.3.5, Transfer Format.

**Table 35.11 Relationship among SPNDEN Bit, SPND, and Next-Access Delay Value**

SPCMDm.SPNDEN Bit	SPND.SPNDL[2:0] Bits	Next-Access Delay Value
0	000 to 111	1 RSPCK + 2 PCLK
1	000	1 RSPCK + 2 PCLK
	001	2 RSPCK + 2 PCLK
	010	3 RSPCK + 2 PCLK
	011	4 RSPCK + 2 PCLK
	100	5 RSPCK + 2 PCLK
	101	6 RSPCK + 2 PCLK
	110	7 RSPCK + 2 PCLK
	111	8 RSPCK + 2 PCLK

## (8) Initialization Flowchart

Figure 35.34 is a flowchart illustrating an example of initialization in SPI operation when the RSPI is used in master mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, see the descriptions given in the individual blocks.

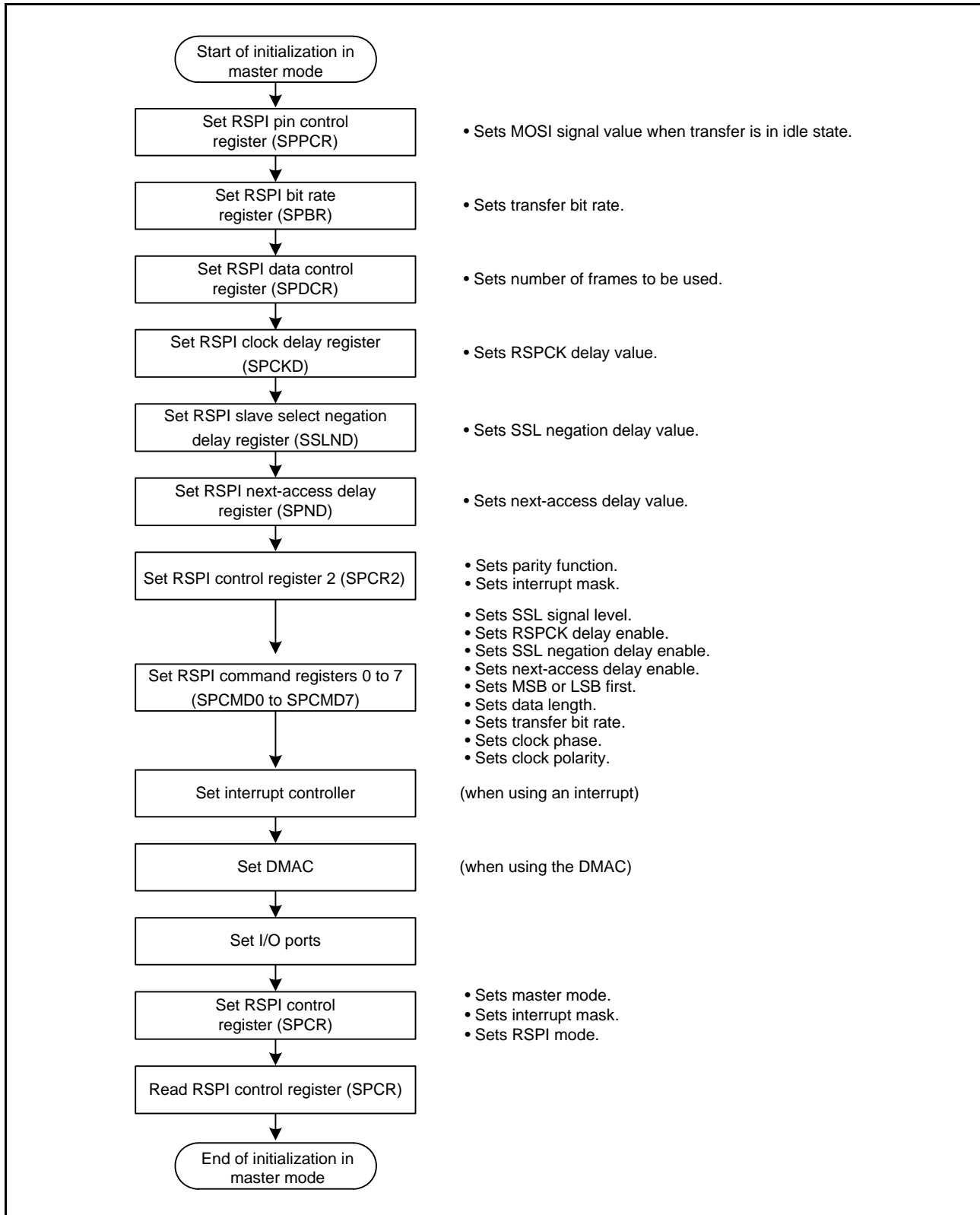


Figure 35.34 Example of Initialization Flowchart in Master Mode (SPI Operation)

(9) Software Processing Flow

Figure 35.35 to Figure 35.37 show examples of the flow of software processing.

(a) Transmit Processing Flow

When transmitting data, the CPU will be notified of the completion of data transmission after the last writing of data for transmission if the SPII interrupt is enabled.

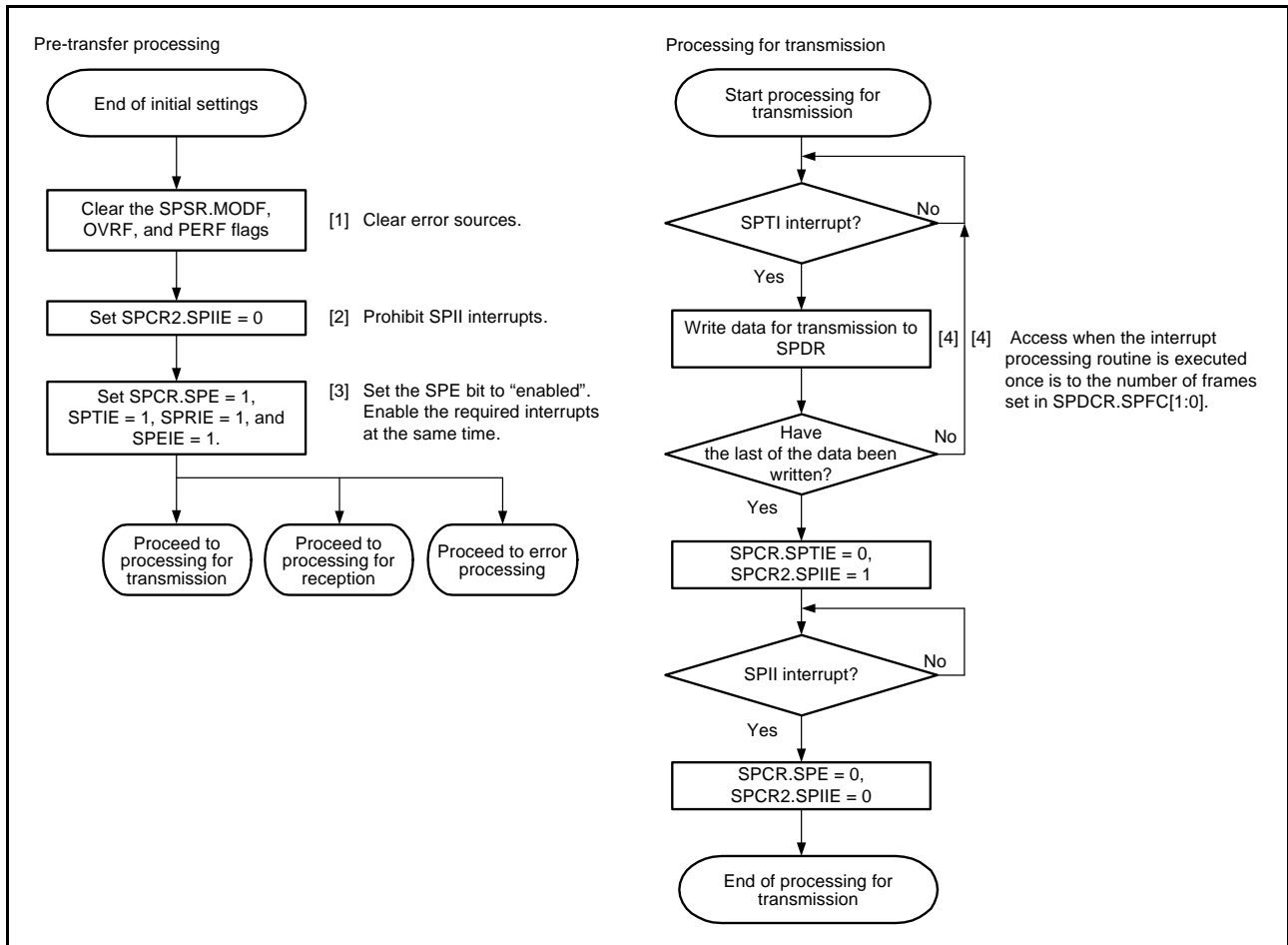


Figure 35.35 Flowchart in Master Mode (Transmission)

(b) Receive Processing Flow

The RSPI does not handle receive-only operation, so processing for transmission is required if reception is to proceed.

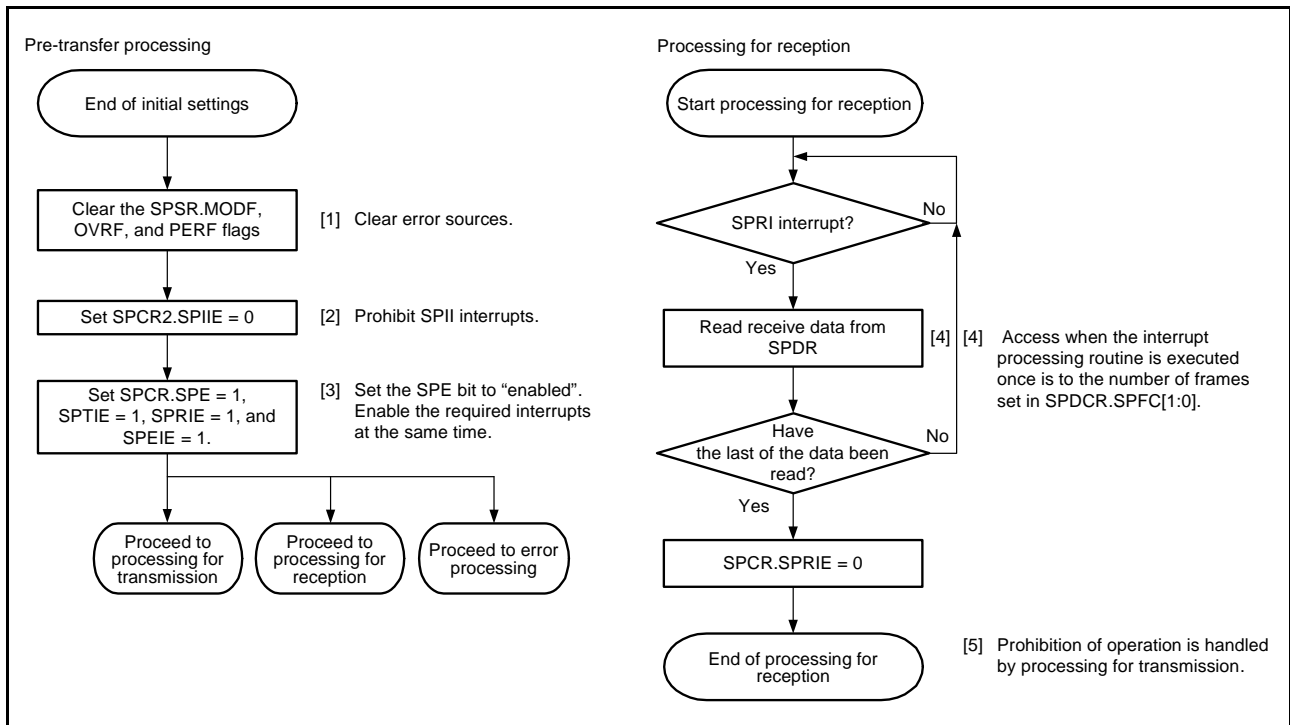


Figure 35.36 Flowchart in Master Mode (Reception)

(c) Flow of error processing

The RSPI has three types of error. When a mode-fault error is generated, the SPCR.SPE bit is automatically cleared, stopping operations for transmission and reception. For errors from other sources, however, the SPCR.SPE bit is not cleared and operations for transmission and reception continue; accordingly, we recommend clearing of the SPCR.SPE bit to stop operations in the case of errors other than mode-fault errors. Not doing so will lead to updating of the SPSSR.SPECM[2:0] bits.

When an error occurs, clear the ICU.IRn.IR flag from within the error processing routine. If this is not done, the ICU.IRn.IR flag may continue to indicate the SPTI interrupt or SPRI interrupt request.

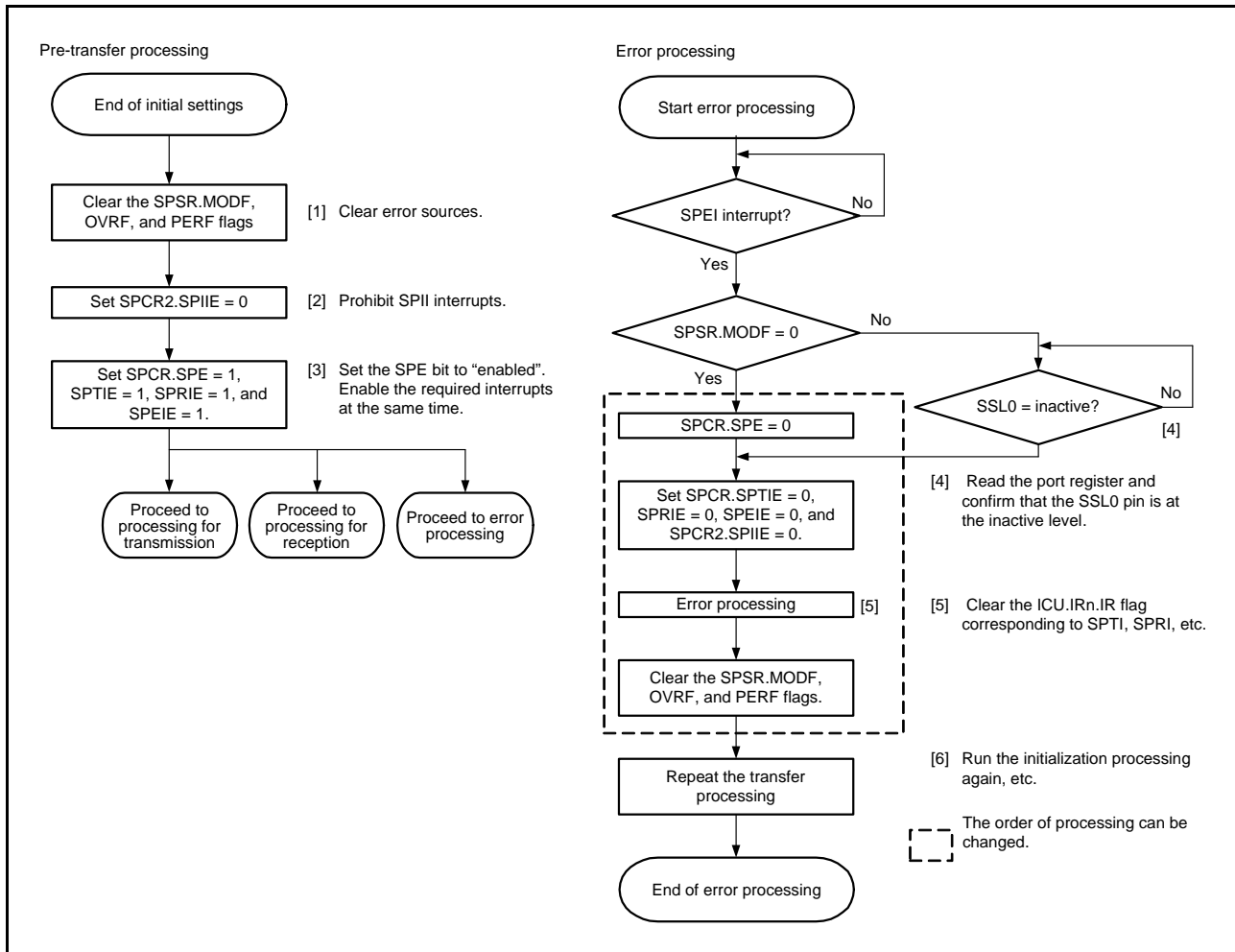


Figure 35.37 Flowchart for Master Mode (Error Processing)



### 35.3.10.2 Slave Mode Operation

#### (1) Starting a Serial Transfer

If the SPCMD0.CPHA bit is 0, when detecting an SSLn0 input signal assertion, the RSPI needs to start driving valid data to the MISO<sub>n</sub> output signal. For this reason, when the CPHA bit is 0, the assertion of the SSLn0 input signal triggers the start of a serial transfer.

If the CPHA bit is 1, when detecting the first RSPCK<sub>n</sub> edge in an SSLn0 signal asserted condition, the RSPI needs to start driving valid data to the MISO<sub>n</sub> output signal. For this reason, when the CPHA bit is 1, the first RSPCK<sub>n</sub> edge in an SSLn0 signal asserted condition triggers the start of a serial transfer.

When detecting the start of a serial transfer in a condition in which the shift register is empty, the RSPI changes the status of the shift register to “full”, so that data cannot be copied from the transmit buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, the RSPI leaves the status of the shift register unchanged, in the full state.

Irrespective of CPHA bit setting, the timing at which the RSPI starts driving of the MISO<sub>n</sub> output signal is the SSLn0 signal assertion timing. The data which is output by the RSPI is either valid or invalid, depending on the CPHA bit setting.

For details on the RSPI transfer format, see section 35.3.5, Transfer Format. The polarity of the SSLn0 input signal depends on the setting of the SSL<sub>OP</sub> bit in the RSPI slave select polarity register (SSLP).

#### (2) Terminating a Serial Transfer

Irrespective of the SPCMD0.CPHA bit, the RSPI terminates the serial transfer after detecting an RSPCK<sub>n</sub> edge corresponding to the final sampling timing. When free space is available in the receive buffer, upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the RSPI data register (SPDR).

Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty”, regardless of the receive buffer state. A mode fault error occurs if the RSPI detects an SSLn0 input signal negation from the beginning of serial transfer to the end of serial transfer (see section 35.3.8, Error Detection).

The final sampling timing changes depending on the bit length of transfer data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[3:0] bit setting. The polarity of the SSLn0 input signal depends on the SSLP.SSL<sub>OP</sub> bit setting.

For details on the RSPI transfer format, see section 35.3.5, Transfer Format.

#### (3) Notes on Single-Slave Operations

If the SPCMD0.CPHA bit is 0, the RSPI starts serial transfers when it detects the assertion edge for an SSLn0 input signal. In the type of configuration shown in Figure 35.7 as an example, if the RSPI is used in single-slave mode, the SSLn0 signal is always fixed at the active state. Therefore, when the CPHA bit is set to 0, the RSPI cannot correctly start a serial transfer. To correctly execute transmit/receive operations by the RSPI in slave mode in a configuration in which the SSLn0 input signal is fixed at the active state, the CPHA bit should be set to 1. If there is a need for setting the CPHA bit to 0, the SSLn0 input signal should not be fixed.

#### (4) Burst Transfer

If the SPCMD0.CPHA bit is 1, continuous serial transfer (burst transfer) can be executed while retaining the assertion state for the SSLn0 input signal. If the CPHA bit is 1, the period from the first RSPCK<sub>n</sub> edge to the sampling timing for the reception of the final bit in an SSLn0 signal active state corresponds to a serial transfer period. Even when the SSLn0 input signal remains at the active level, the RSPI can accommodate burst transfers because it can detect the start of an access.

If the CPHA bit is 0, the second and subsequent serial transfers during burst transfer cannot be executed correctly.

## (5) Initialization Flowchart

Figure 35.38 is a flowchart illustrating an example of initialization in SPI operation when the RSPI is used in slave mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, see the descriptions given in the individual blocks.

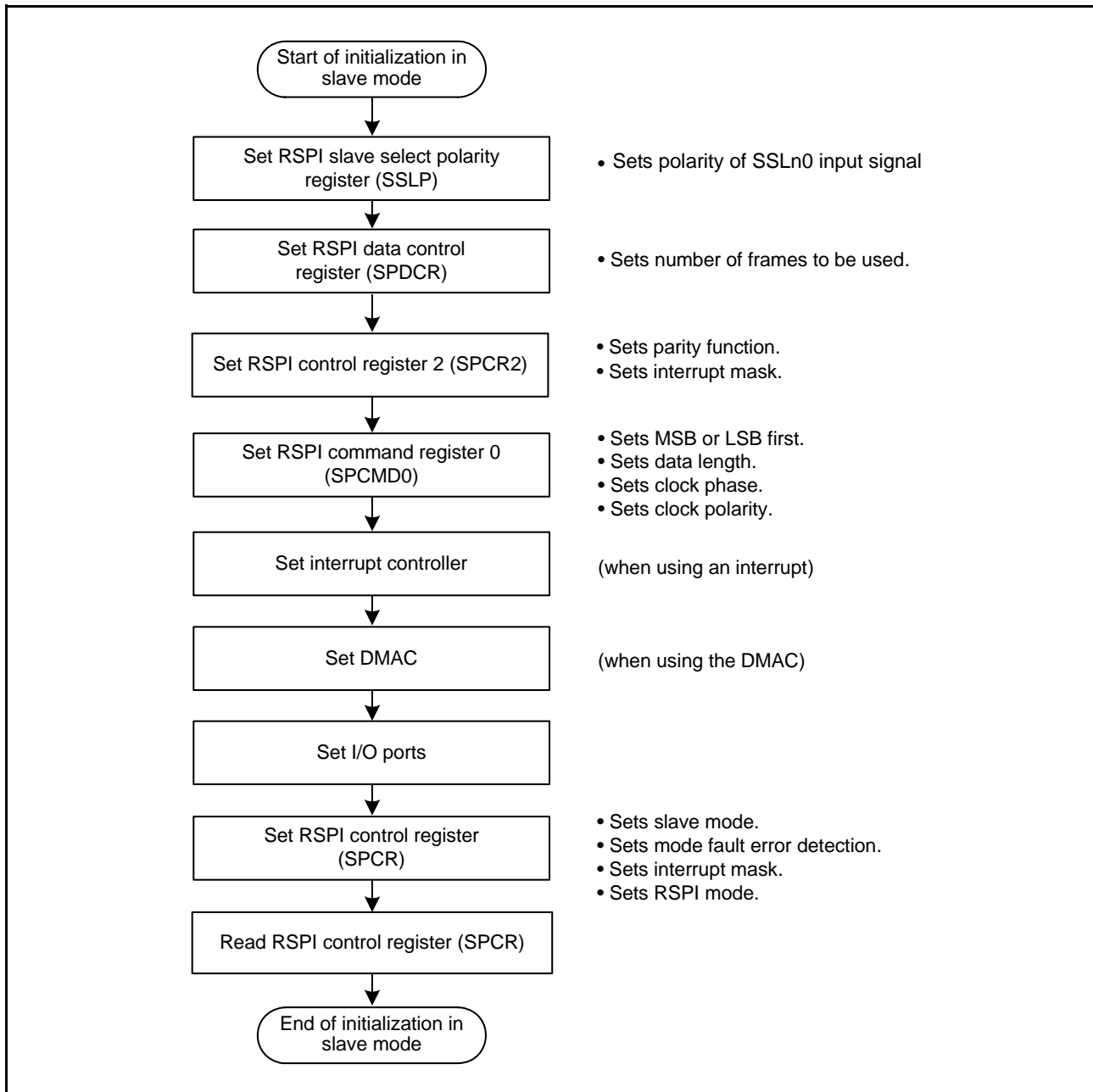


Figure 35.38 Example of Initialization Flowchart in Slave Mode (SPI Operation)

(6) Flow of Software Processing

Figure 35.39 to Figure 35.41 show examples of the flow of software processing.

(a) Flow of processing for transmission

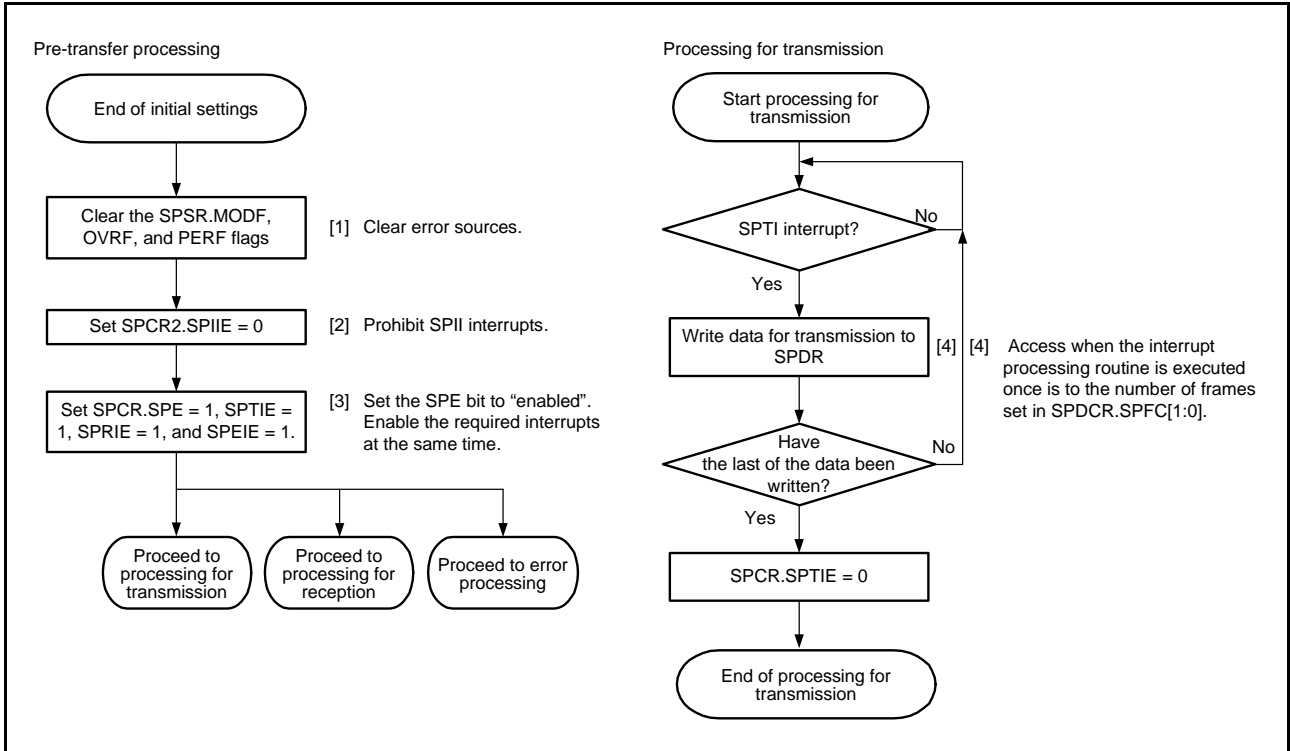


Figure 35.39 Flowchart for Slave Mode (Transmission)

(b) Flow of processing for reception

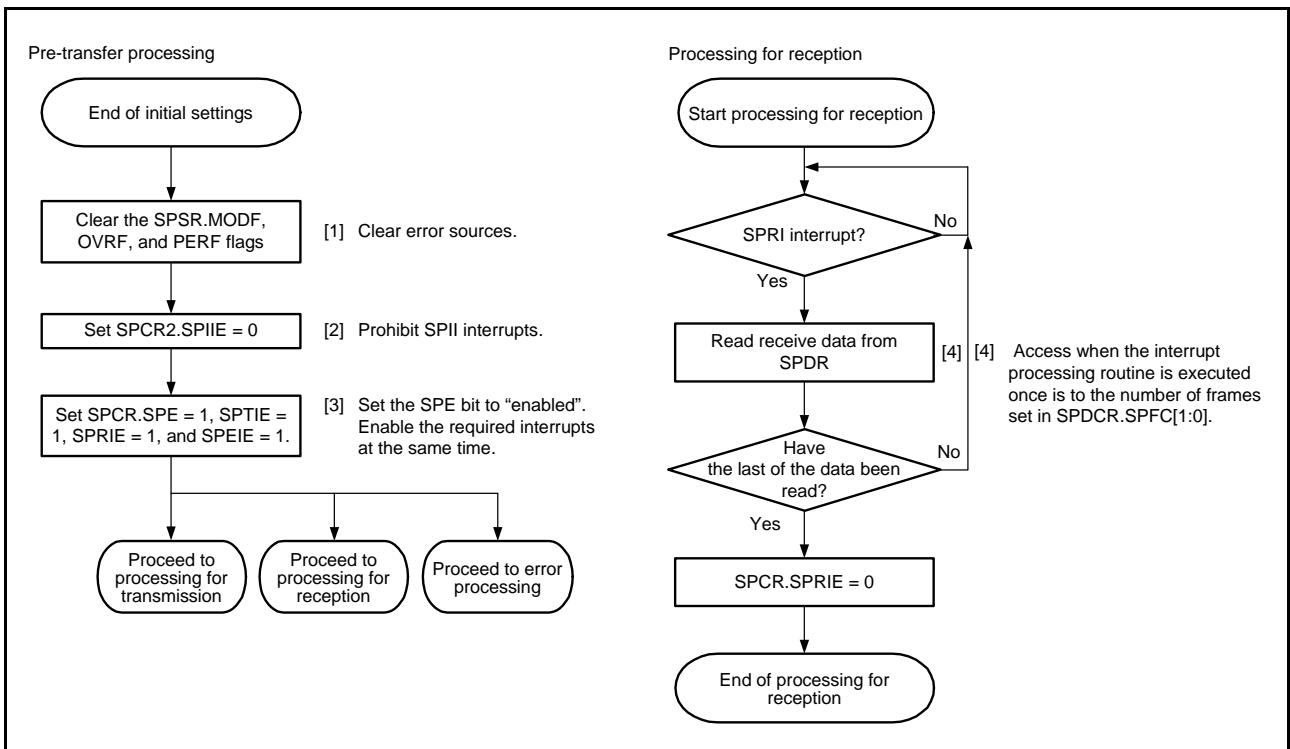


Figure 35.40 Flowchart for Slave Mode (Reception)

(c) Flow of error processing

In slave operation, even when a mode-fault error is generated, the SPSR.MODF flag can be cleared without de-asserting the pin.

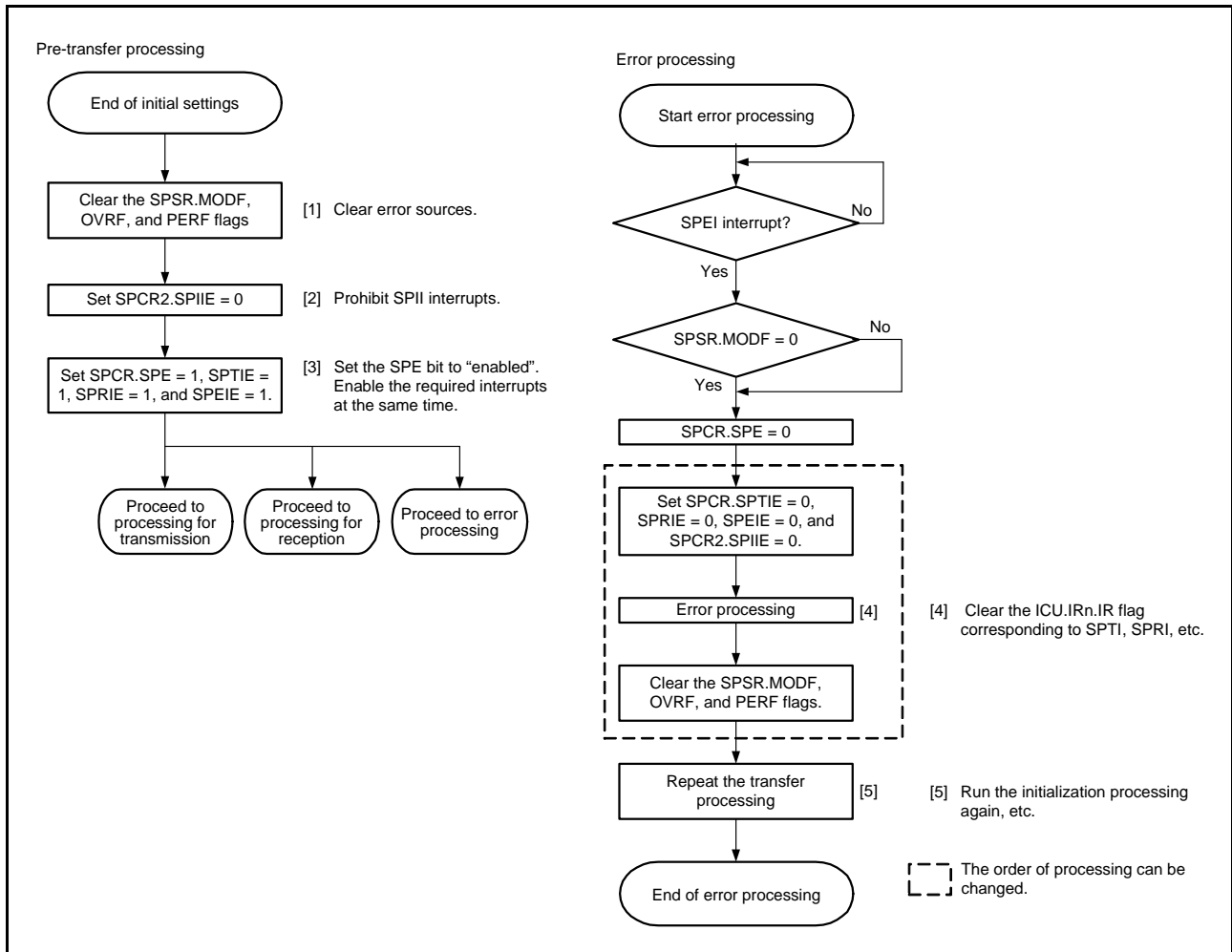


Figure 35.41 Flowchart for Slave Mode (Error Processing)

### 35.3.11 Clock Synchronous Operation

Setting the SPMS bit in the RSPI control register (SPCR) to 1 selects clock synchronous operation of the RSPI. In clock synchronous operation, the SSLni pin is not used, and the three pins of RSPCKn, MOSIn, and MISO<sub>n</sub> handle communications. The SSLni pin is available as I/O port pins.

Although clock synchronous operation does not require use of the SSLni pin, operation of the module is the same as in SPI operation. That is, in both master and slave operations, communications can be performed with the same flow as in SPI operation. However, mode fault errors are not detected because the SSLni pin is not used.

Furthermore, operation is not guaranteed if clock synchronous operation proceeds when the SPCMDm.CPHA bit is set to 0 in slave mode (SPCR.MSTR = 0).

### 35.3.12 Master Mode Operation

#### (1) Starting a Serial Transfer

The RSPI updates the data in the transmit buffer (SPTX) of SPDR when data is written to the RSPI data register (SPDR) with the transmit buffer being empty (data for the next transfer is not set). When the shift register is empty after the number of frames set in the SPDCR.SPFC[1:0] bits are written to the SPDR, the RSPI copies data from the transmission buffer to the shift register and starts serial transmission. Upon copying transmit data to the shift register, the RSPI changes the status of the shift register to “full”, and upon termination of serial transfer, it changes the status of the shift register to “empty”. The status of the shift register cannot be referenced.

For details on the RSPI transfer format, see section 35.3.5, Transfer Format.

#### (2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after transmitting an RSPCKn edge corresponding to the sampling timing. If free space is available in the receive buffer, upon termination of serial transfer, the RSPI copies data from the shift register to the receive buffer of the RSPI data register (SPDR).

It should be noted that the final sampling timing varies depending on the bit length of transfer data. In master mode, the RSPI data length depends on the SPCMDm.SP[3:0] bit setting.

For details on the RSPI transfer format, see section 35.3.5, Transfer Format.

However, transfer in clock-synchronous operation is conducted without the SSLn0 output signal.

#### (3) Sequence Control

The transfer format employed in master mode is determined by SPSCR, SPCMDm, SPBR, SPCKD, SSLND, and SPND registers. Although the SSLni signals are not output in clock synchronous operation, these settings are valid.

SPSCR is a register used to determine the sequence configuration for serial transfers that are executed by the RSPI in master mode. The following items are set in SPCMDm register: SSLni output signal value, MSB/LSB first, data length, some of the bit rate settings, RSPCKn polarity/phase, whether SPCKD is to be referenced, whether SSLND is to be referenced, and whether SPND is to be referenced. SPBR holds some of the bit rate settings; SPCKD, an RSPI clock delay value; SSLND, an SSL negation delay; and SPND, a next-access delay value.

According to the sequence length that is assigned to SPSCR, the RSPI makes up a sequence comprised of a part or all of SPCMDm register. The RSPI contains a pointer to the SPCMDm register that makes up the sequence. The value of this pointer can be checked by reading the SPSSR.SPCP[2:0] bits. When the SPE bit in the RSPI control register (SPCR) is set to 1 and the RSPI function is enabled, the RSPI loads the pointer to the commands in SPCMD0 register, and incorporates the SPCMD0 register setting into the transfer format at the beginning of serial transfer.

The RSPI increments the pointer each time the next-access delay period for a data transfer ends. Upon completion of the serial transfer that corresponds to the final command comprising the sequence, the RSPI sets the pointer in SPCMD0 register, and in this manner the sequence is executed repeatedly.

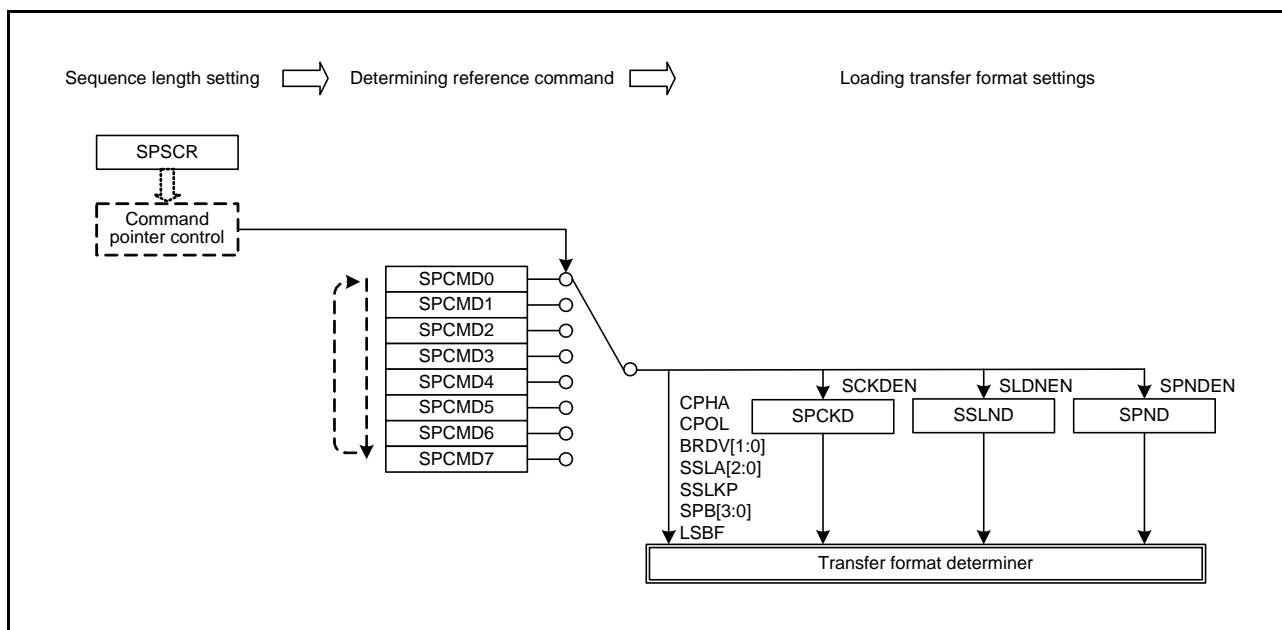


Figure 35.42 Procedure for Determining the Form of Serial Transmission in Master Mode

In this section, a frame is the combination of the data (SPDR) and the settings (SPCMDm).

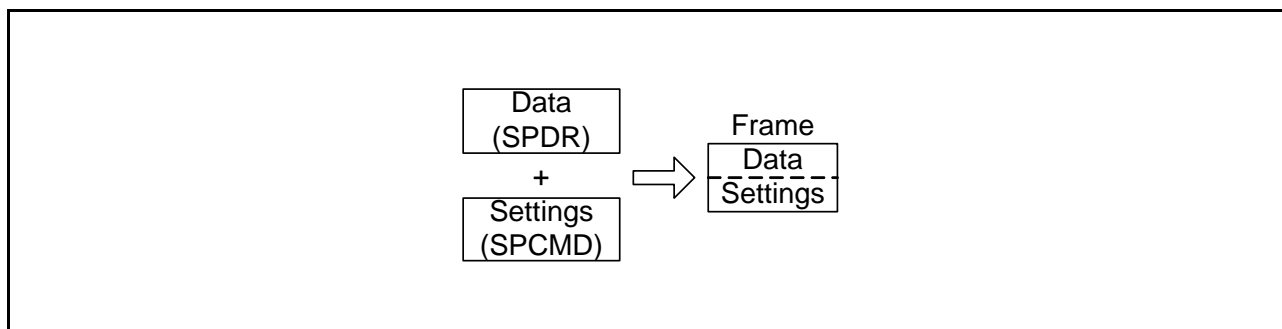


Figure 35.43 Concept of a Frame

Figure 35.44 shows the relationship between the command and the transmit and receive buffers in the sequence of operations specified by the settings in Table 35.4.

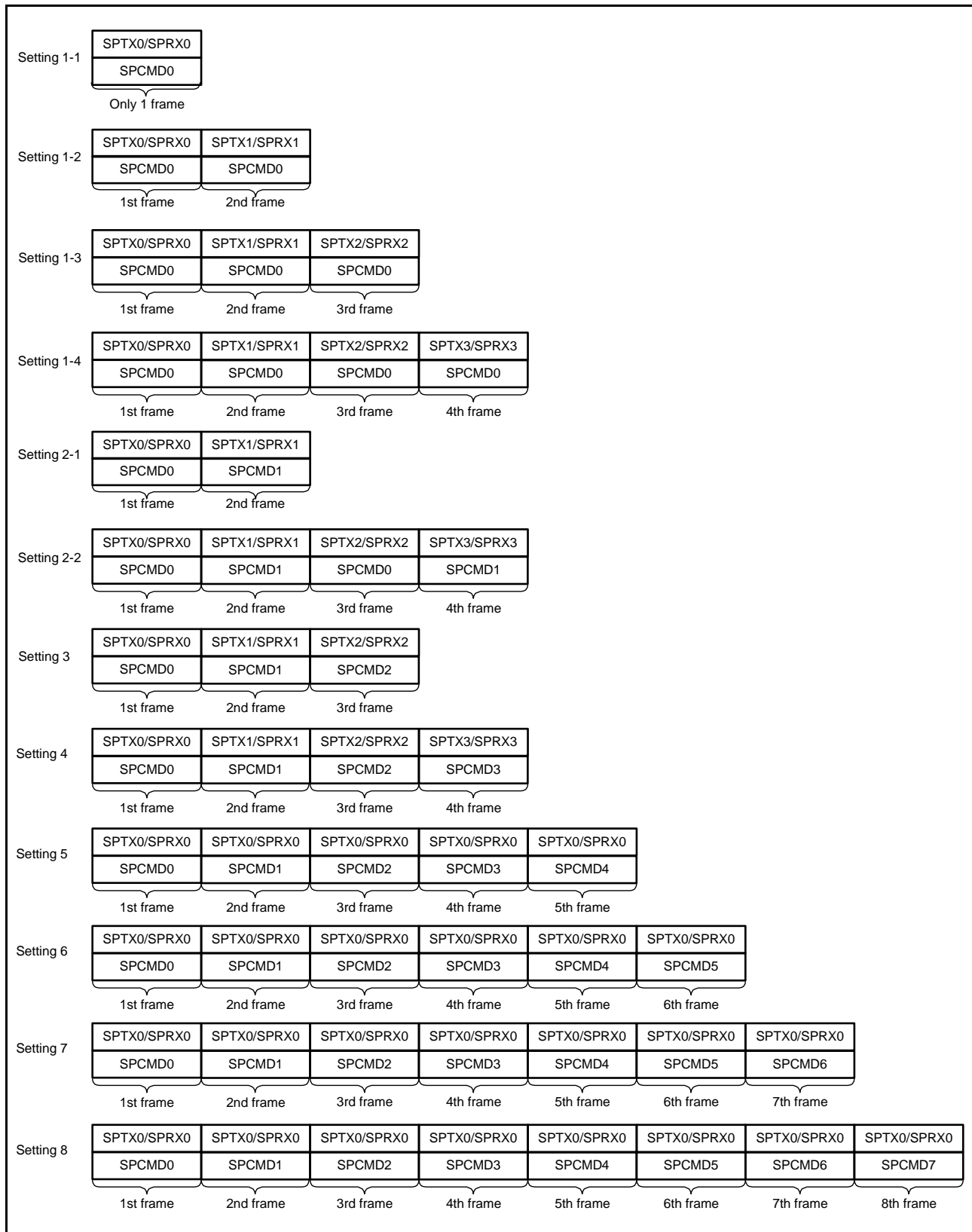


Figure 35.44 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations

(4) Initialization Flowchart

Figure 35.45 is a flowchart illustrating an example of initialization in clock synchronous operation when the RSPi is used in master mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, see the descriptions given in the individual blocks.

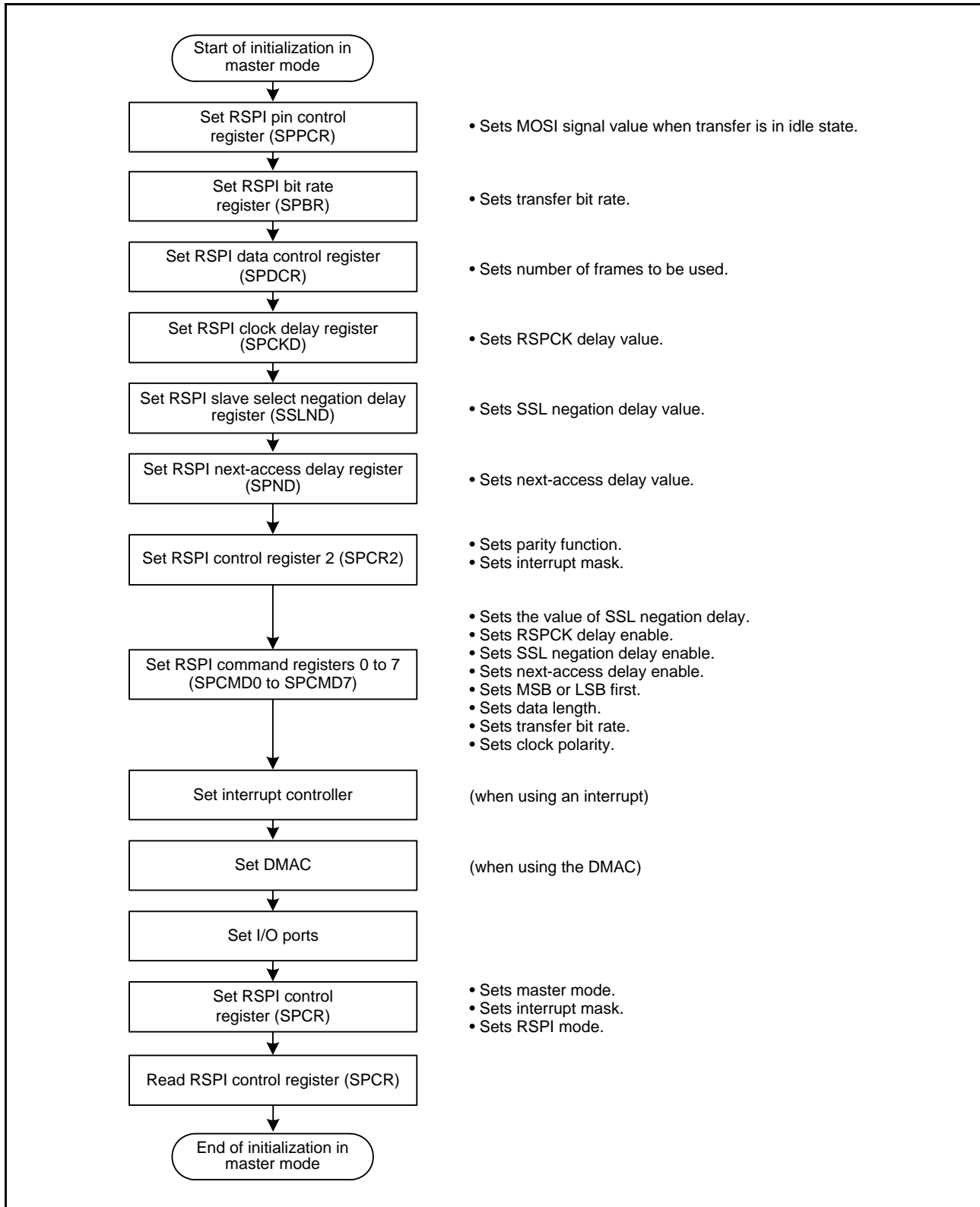


Figure 35.45 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation)



### (5) Flow of Software Processing

Software processing during clock-synchronous master operation is the same as that for SPI master operation. For details, see 35.3.10.1, (9) Software Processing Flow. Note that mode-fault errors will not occur.

## 35.3.13 Slave Mode Operation

### (1) Starting a Serial Transfer

When the SPCR.SPMS bit is 1, the first RSPCKn edge triggers the start of a serial transfer in the RSPI. When detecting the start of a serial transfer in a condition in which the shift register is empty, the RSPI changes the status of the shift register to “full”, so that data cannot be copied from the transmit buffer to the shift register when serial transfer is in progress. If the shift register was full before the serial transfer started, the RSPI keeps the status of the shift register unchanged, in the full state. When the SPMS bit is 1, the RSPI always drives the MISO<sub>n</sub> output signal.

For details on the RSPI transfer format, see section 35.3.5, Transfer Format. It should be noted that the SSL0 input signal is not used in clock synchronous operation.

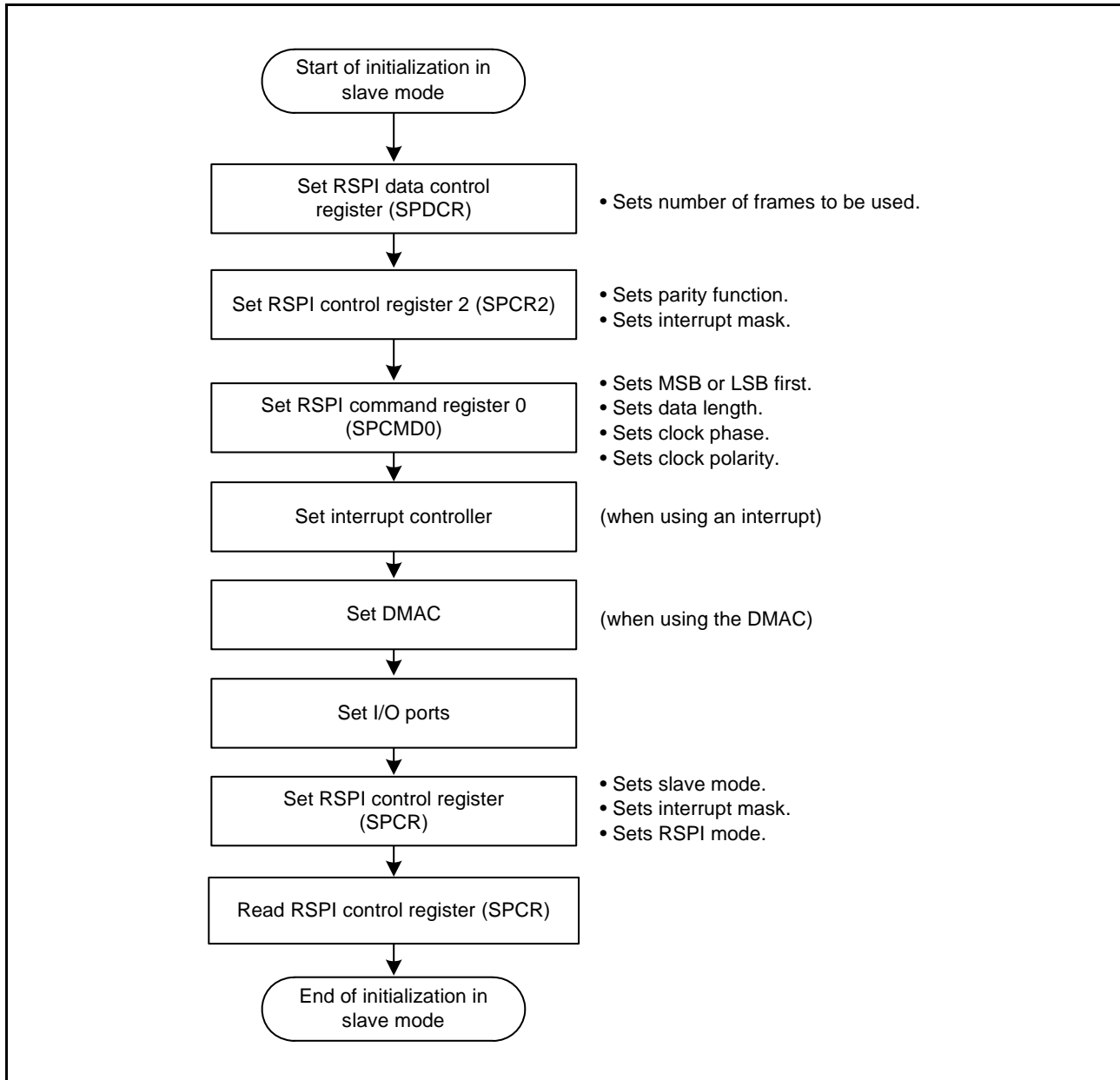
### (2) Terminating a Serial Transfer

The RSPI terminates the serial transfer after detecting an RSPCKn edge corresponding to the final sampling timing. When free space is available in the receive buffer, upon termination of serial transfer the RSPI copies received data from the shift register to the receive buffer of the RSPI data register (SPDR).

Upon termination of a serial transfer the RSPI changes the status of the shift register to “empty”. The final sampling timing changes depending on the bit length of transfer data. In slave mode, the RSPI data length depends on the SPCMD0.SPB[3:0] bit setting. For details on the RSPI transfer format, see section 35.3.5, Transfer Format.

### (3) Initialization Flowchart

Figure 35.46 is a flowchart illustrating an example of initialization in clock synchronous operation when the RSPI is used in slave mode. For a description of how to set up the interrupt controller, DMAC, and I/O ports, see the descriptions given in the individual blocks.



**Figure 35.46 Example of Initialization Flowchart in Slave Mode (Clock Synchronous Operation)**

### (4) Flow of Software Processing

Software processing during clock-synchronous slave operation is the same as that for SPI slave operation. For details, see 35.3.10.2, (6) Flow of Software Processing. Note that mode-fault errors will not occur.

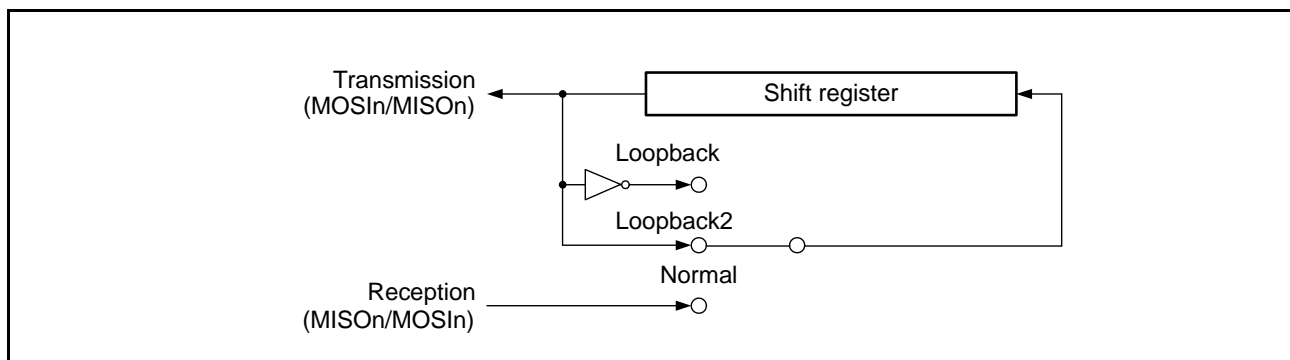
### 35.3.14 Loopback Mode

When 1 is written to the SPPCR.SPLP2 bit or SPPCR.SPLP bit, the RSPI shuts off the path between the MISO pin and the shift register if the SPCR.MSTR bit is 1, and between the MOSI pin and the shift register if the SPCR.MSTR bit is 0, and connects the input path and output path of the shift register. The RSPI does not shut off the path between the MOSI pin and the shift register if the SPCR.MSTR bit is 1, and between the MISO pin and the shift register if the SPCR.MSTR bit is 0. This is called loopback mode. When a serial transfer is executed in loopback mode, the transmit data for the RSPI or the reversed transmit data becomes the received data for the RSPI.

Table 35.12 lists the relationship among the SPLP2 and SPLP bits and the received data. Figure 35.47 shows the configuration of the shift register I/O paths for the case where the RSPI in master mode is set in loopback mode (SPPCR.SPLP2 = 0, SPPCR.SPLP = 1).

**Table 35.12 SPLP2 and SPLP Bit Settings and Received Data**

SPPCR.SPLP2 Bit	SPPCR.SPLP Bit	Received Data
0	0	Input data from the MOSI pin or MISO pin
0	1	Reversed transmit data
1	0	Transmit data
1	1	Transmit data



**Figure 35.47 Configuration of Shift Register I/O Paths in Loopback Mode (Master Mode)**

### 35.3.15 Self-Diagnosis of Parity Bit Function

The parity circuit consists of a parity bit adding unit used for transmit data and an error detecting unit used for received data. In order to detect defects in the parity bit adding unit and error detecting unit of the parity circuit, self-diagnosis is executed for the parity circuit following the flowchart shown in Figure 35.48.

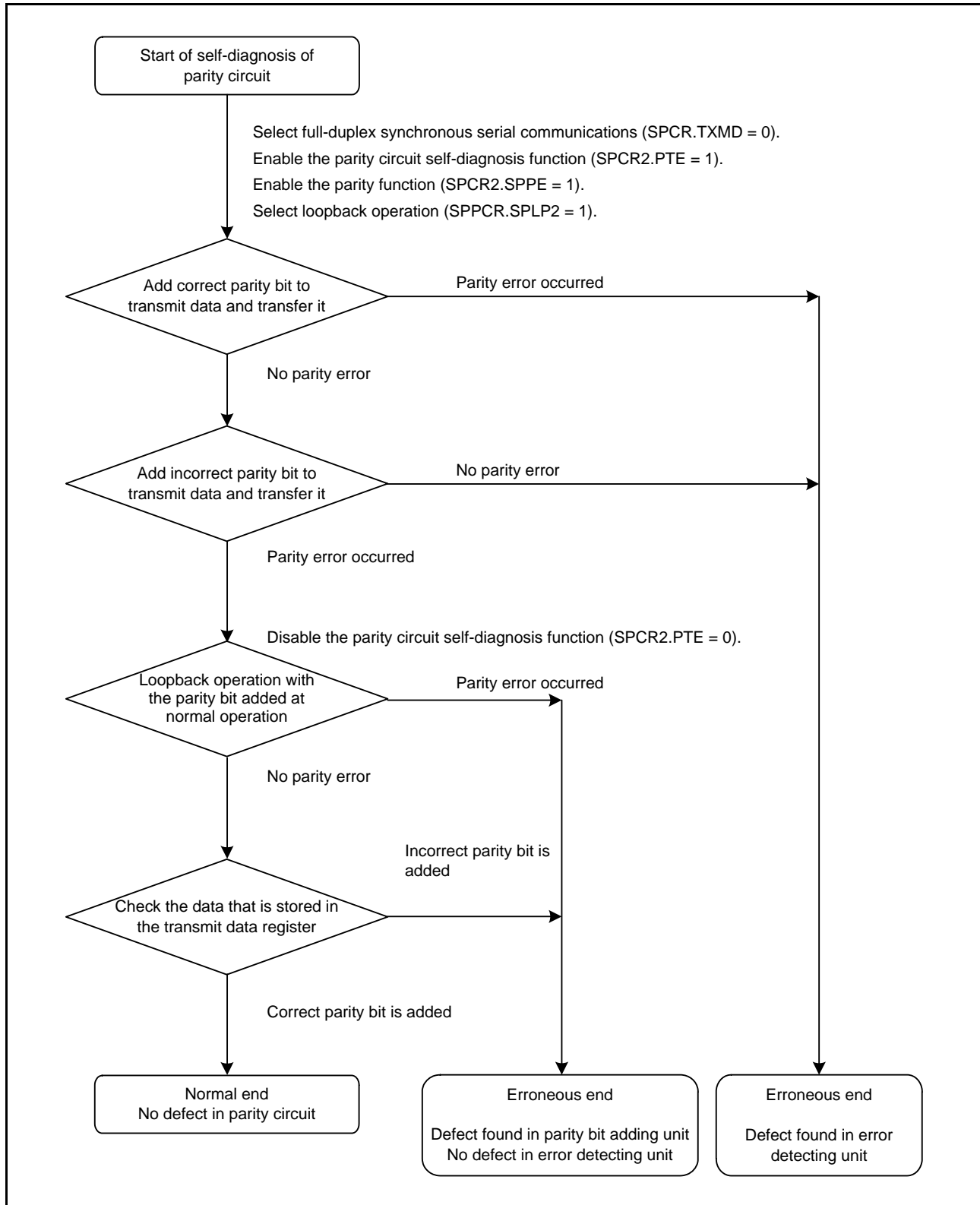


Figure 35.48 Flowchart for Self-Diagnosis of Parity Circuit

### 35.3.16 Interrupt Sources

The RSPI has interrupt sources of receive buffer full, transmit buffer empty, mode fault, overrun, parity error, and RSPI idle. In addition, the DTC or DMAC can be activated by the receive buffer full or transmit buffer empty interrupt to perform data transfer.

Since the vector address for SPEI is allocated to interrupt requests due to mode-fault, overrun, and parity errors, the actual interrupt source must be determined from the flags. Interrupt sources for the RSPI are listed in Table 35.13. An interrupt is generated on satisfaction of an interrupt condition in Table 35.13. Clear the receive buffer full and transmit buffer empty sources through data transfer.

When using the DTC or DMAC to perform data transmission/reception, the DTC or DMAC must be set up first to be in a status in which transfer is enabled before making the RSPI settings. For the method for setting the DTC or DMAC, refer to section 18, DMA Controller (DMACA), or section 19, Data Transfer Controller (DTCa).

If the conditions for generating a transmit buffer empty or receive buffer full interrupt are generated while the ICU.IRn.IR flag is 1, the interrupt is not output as a request for ICU but is retained internally (the capacity for retention is one request per source). A retained interrupt request to ICU is output when the ICU.IRn.IR flag becomes 0. A retained interrupt request is automatically discarded once it is output as an actual interrupt request. The interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) for an internally retained interrupt request can also be cleared to 0.

**Table 35.13 Interrupt Sources of RSPI**

Interrupt Source	Symbol	Interrupt Condition	DMAC/DTC Activation
Receive buffer full	SPRI	The receive buffer becomes full while the SPCR.SPRIE bit is 1.	Possible
Transmit buffer empty	SPTI	The transmit buffer becomes empty while the SPCR.SPTIE bit is 1.	Possible
RSPI errors (mode fault, overrun, and parity error)	SPEI	The SPSR.MODF, OVRF, or PERF flag is set to 1 while the SPCR.SPEIE bit is 1.	Impossible
RSPI idle	SPII	The SPSR.IDLNF flag is set to 0 while the SPCR.SPIIE bit is 1.	Impossible

## 35.4 Usage Note

### 35.4.1 Setting Module-Stop Function

Module stop control register B (MSTPCRB) or module stop control register C (MSTPCRC) can be used to enable or disable operation of the RSPI. The RSPI is stopped after a reset. The registers become accessible on release from the module-stop state. For details, refer to section 11, Low Power Consumption.

### 35.4.2 Cautionary Note on the Low Power Consumption Functions

When a low power consumption function is to be used to lower power consumption by the RSPI, use the low power consumption function after the SPCR.SPE bit is set to 0 and transfer ends.

### 35.4.3 Points to Note on Starting Transfer

If the ICU.IRn.IR flag is 1 at the time transfer is to be started, an interrupt request is internally retained after transfer starts, and this can lead to unanticipated behavior of the ICU.IRn.IR flag.

When the ICU.IRn.IR flag is 1 at the time transfer is to start, follow the procedure below to clear interrupt requests before enabling operations (by setting the SPCR.SPE bit to 1).

1. Confirm that transfer has stopped (i.e. that the SPCR.SPE bit is 0).
2. Clear the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) to 0.
3. Read the relevant interrupt enable bit (the SPCR.SPTIE or SPCR.SPRIE bit) and confirm that its value is 0.
4. Clear the ICU.IRn.IR flag to 0.

## 36. IEBus Controller (IEB)

### 36.1 Overview

The RX630 Group has an on-chip one-channel IEBus controller (IEB).

The Inter Equipment Bus™ (IEBus™)\*1 implements a small-scale digital data transfer system for inter-equipment data transfer.

The RX630 Group does not have an on-chip IEBus driver/receiver, so it is necessary to mount a dedicated driver/receiver externally.

Table 36.1 lists the specifications of the IEB, Figure 36.3 shows a block diagram of the IEB, and Table 36.5 lists the input/output pins of the IEB.

Note 1. IEBus™ (Inter Equipment Bus™) is a trademark of Renesas Electronics Corporation.

**Table 36.1 Specifications of IEB**

Item	Specifications
Communications protocol	<ul style="list-style-type: none"> <li>IEBus protocol control (layer 2) supported</li> <li>Half-duplex asynchronous communications</li> <li>Multi-master system</li> <li>Broadcast communications function</li> <li>Selectable two communications modes (modes 0, 1) with different transfer speeds</li> </ul>
Data transmission and reception	<ul style="list-style-type: none"> <li>On-chip buffers for data transmission and reception</li> <li>Transmission and reception buffers: 32 bytes each</li> <li>Up to 32 bytes of consecutive transmit/reception (maximum number of transfer bytes in mode 1)</li> </ul>
Operating frequency	12 MHz, 12.58 MHz (IEB uses IECLK divided by 2) 18 MHz, 18.87 MHz (IEB uses IECLK divided by 3) 24 MHz, 25.16 MHz (IEB uses IECLK divided by 4) 30 MHz, 31.45 MHz (IEB uses IECLK divided by 5) 36 MHz, 37.74 MHz (IEB uses IECLK divided by 6) 42 MHz, 44.03 MHz (IEB uses IECLK divided by 7)
Power consumption reducing function	Module-stop state can be set.

### 36.1.1 Transfer Data (Data Field Contents)

The data field contents are specified by the control bits.

**Table 36.2 Control Bit Contents**

Set Value	Bit 3*1	Bit 2	Bit 1	Bit 0	Function*2
0h	0	0	0	0	Reads slave status (SSR)
1h	0	0	0	1	Undefined
2h	0	0	1	0	Undefined
3h	0	0	1	1	Reads data and locks
4h	0	1	0	0	Reads locked address (lower 8 bits)
5h	0	1	0	1	Reads locked address (upper 4 bits)
6h	0	1	1	0	Reads slave status (SSR) and unlocks
7h	0	1	1	1	Reads data
8h	1	0	0	0	Undefined
9h	1	0	0	1	Undefined
Ah	1	0	1	0	Writes command and locks
Bh	1	0	1	1	Writes data and locks
Ch	1	1	0	0	Undefined
Dh	1	1	0	1	Undefined
Eh	1	1	1	0	Writes command
Fh	1	1	1	1	Writes data

Note 1. Depending on the value of bit 3 (MSB), the transfer directions of the message length bits in the following message length field and data in the data field vary.

When bit 3 is 1: Data is transferred from the master unit to the slave unit.

When bit 3 is 0: Data is transferred from the slave unit to the master unit.

Note 2. 3h, 6h, Ah, and Bh are control bits to specify lock setting and cancellation.

When the undefined values of 1h, 2h, 8h, 9h, Ch, and Dh are transmitted, the acknowledge signal is not returned.

When the control bits received from a unit that is not the master unit that has requested a lock are not listed in Table 36.3, the slave unit locked by the master unit does not accept the control bits and does not return the acknowledge bit.

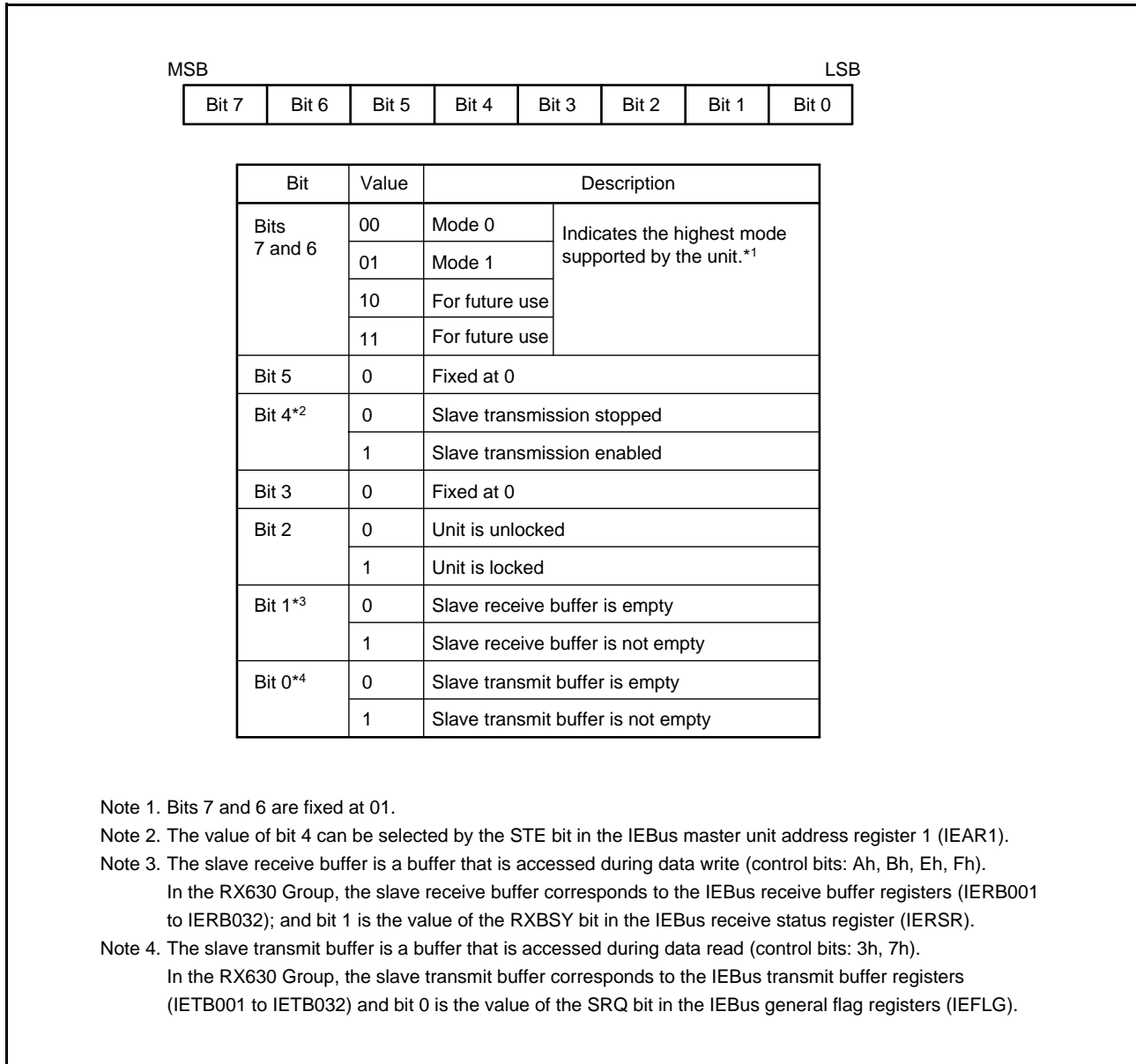
**Table 36.3 Control Field for Locked Slave Unit**

Set Value	Bit 3	Bit 2	Bit 1	Bit 0	Function
0h	0	0	0	0	Reads slave status
4h	0	1	0	0	Reads locked address (lower 8 bits)
5h	0	1	0	1	Reads locked address (upper 4 bits)



## (1) Slave Status Read (Control Bits: 0h, 6h)

The master unit can check the reason the slave unit does not return the acknowledgement (ACK) by reading the slave status (0h, 6h). The slave status indicates the result of the last communications that the slave unit performed. All slave units can provide slave status information. Figure 36.1 shows the bit configuration of the slave status.



**Figure 36.1 Bit Configuration of Slave Status (SSR)**

**(2) Data Command Transfer (Control Bits: Read (3h, 7h), Write (Ah, Bh, Eh, Fh))**

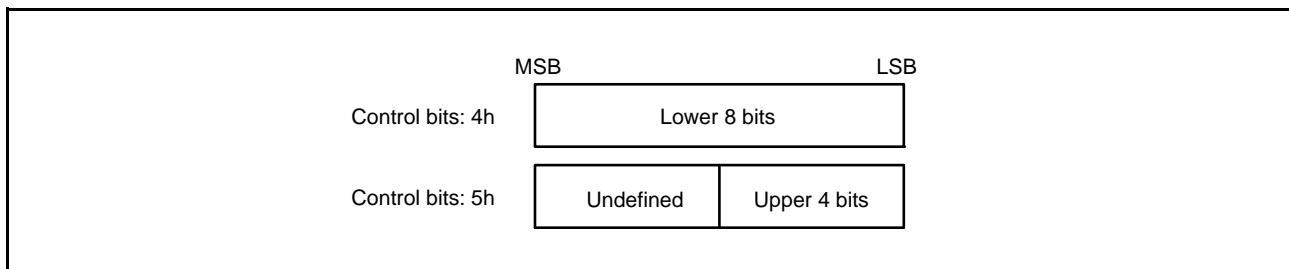
In the case of data read (3h, 7h), data in the data buffer of the slave unit is read in the master unit. In the case of data write (Bh, Fh) or command write (Ah, Eh), data received in the slave unit is processed in accordance with the operation specification of the slave unit.

Note: • The user can select data and commands freely in accordance with the system.

Note: • 3h, Ah, or Bh may lock depending on the communications condition and status.

**(3) Locked Address Read (Control Bits: 4h, 5h)**

In the case of the locked address read (4h, 5h), the address (12 bits) of the master unit that issues the lock instruction is configured in bytes as shown below and read.



**Figure 36.2 Locked Address Configuration**

**(4) Locking/Unlocking (Control Bits: Setting (3h, Ah, Bh), Cancellation: (6h))**

The lock function is used for message transfer over multiple communications frames. A locked unit receives data only from the unit that locked it.

Locking and unlocking are described below.

**(a) Locking**

When an acknowledge bit of 0 in the message length field is transmitted/received with the control bits (3h, Ah, Bh) indicating the lock operation, and then the communications frame is completed before completion of data transmission/reception for the number of bytes specified by the message length bits, the slave unit is locked by the master unit. In this case, the bit (bit 2) relevant to locking in the byte data indicating the slave status is set to 1.

Lock is set only when a transmit/receive-frame maximum transfer byte overflow is generated. Lock is not set by other error terminations.

**(b) Unlocking**

When the control bits indicate the lock (3h, Ah, Bh) or unlock (6h) operation and the byte data for the number of bytes specified by the message length bits are transmitted/received in a single communications frame, the slave unit is unlocked by the master unit. In this case, the bit (bit 2) relevant to locking in the byte indicating the slave status is cleared to 0.

Note that locking and unlocking are not done in broadcast communications.

Note: • There are three ways to cause a locked unit to unlock itself.

- Perform the RES# pin reset
- Perform a software reset
- Issue an unlock command through the IEBus command register (IECMR)

Note that the LCK flag in IEFLG can be used to check whether the unit is locked or unlocked.

Figure 36.3 shows the block diagram of the IEB and Table 36.4 lists the functions of each block.

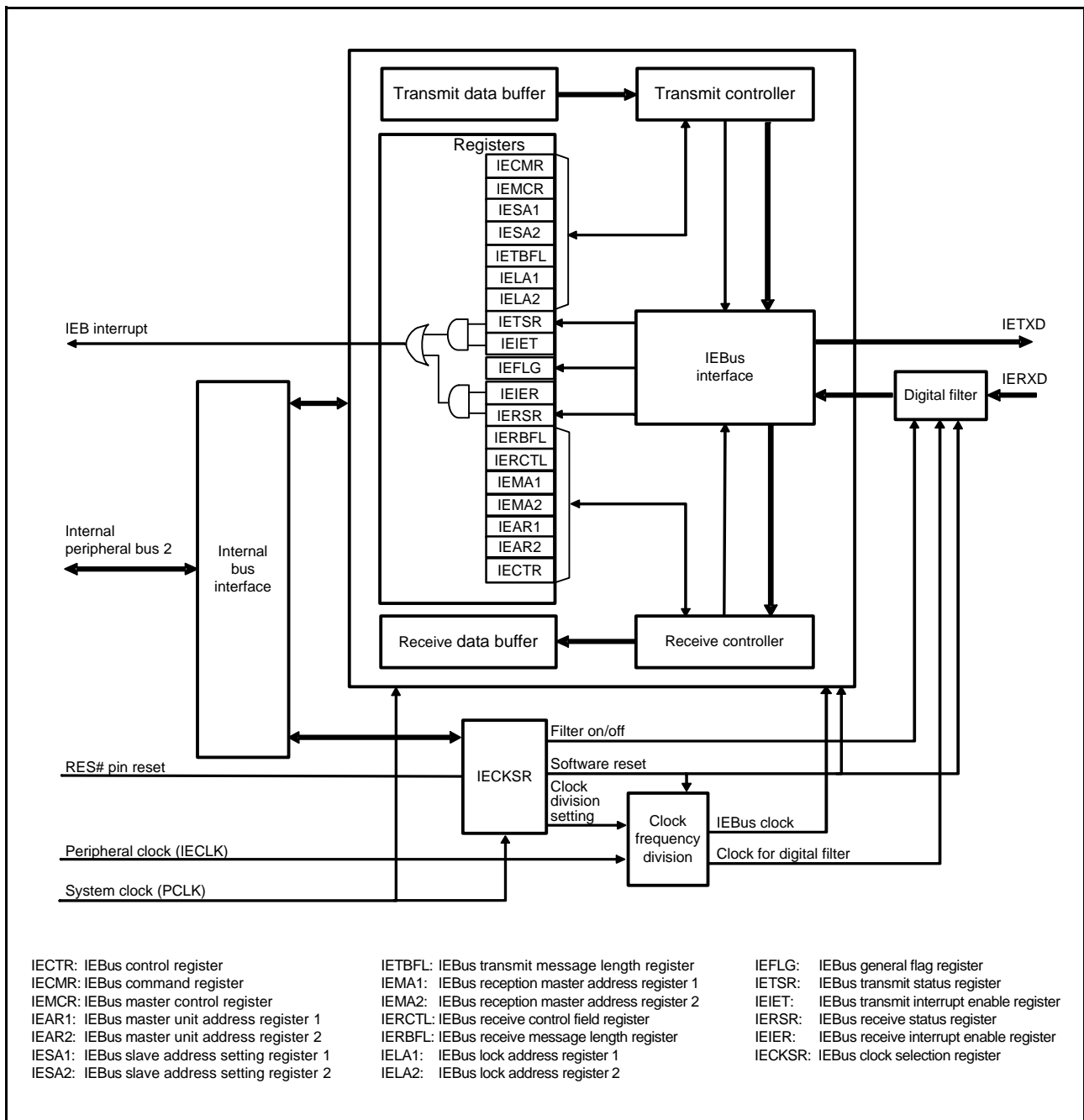


Figure 36.3 Block Diagram of IEB

**Table 36.4 Functions of Each Block**

Block	Function
Bus interface	Internal bus interface <ul style="list-style-type: none"> <li>• Data width: 8 bits</li> <li>• IEB register access</li> </ul>
IEBus interface	Interface conforms to IEBus specifications <ul style="list-style-type: none"> <li>• Outputs data from transmit controller to IEBus in IEBus specification bit format</li> <li>• Picks out frame data in IEBus specification bit format to transfer to receive controller</li> </ul>
IECKSR	Division of the clock for IEBus/digital filter, selected by the IEBus clock selection register Software reset setting function <ul style="list-style-type: none"> <li>• Assertion/negation of the asynchronous software reset for all FFs except the IECKSR register</li> <li>• Digital filter on/off setting</li> <li>• Clock frequency division setting of the input clock to the IEBus clock</li> <li>• Clock frequency division setting of the input clock to the digital filter clock</li> </ul>
Registers	IEB control registers <ul style="list-style-type: none"> <li>• Registers to control IEB</li> <li>• Readable/writable from internal bus</li> </ul>
Clock division	Division output function of the input clock <ul style="list-style-type: none"> <li>• Divides the input clock to be used as the IEBus clock, and outputs it to the IEBus interface</li> <li>• Divides the input clock to be used as the digital filter clock, and outputs it to the digital filter</li> </ul>
Digital filter	Filtering function of the input to the IERXD pin of the IEBus <ul style="list-style-type: none"> <li>• Digital filter on/off setting</li> <li>• Three-strand match filtering</li> </ul>
Transmit controller	Transmits data in transmit buffer to IEBus <ul style="list-style-type: none"> <li>• Generates transmit frame combining header information in register and data in transmit buffer to transmits</li> <li>• Detects transmit error</li> </ul>
Receive controller	Stores data from IEBus in receive buffer <ul style="list-style-type: none"> <li>• Stores header information and data in received frame in register and receive buffer, respectively</li> <li>• Detects receive error</li> </ul>
Transmit data buffer	Buffer for data transmission <ul style="list-style-type: none"> <li>• Buffer that stores data to be transmitted to IEBus</li> <li>• Buffer size: 32 bytes</li> </ul>
Receive data buffer	Buffer for data reception <ul style="list-style-type: none"> <li>• Buffer that stores data received from IEBus</li> <li>• Buffer size: 32 bytes</li> </ul>

**Table 36.5 Input/Output Pins of IEB**

Pin Name	I/O	Function
IERXD	Input	Receive data input pin
IETXD	Output	Transmit data output pin

## 36.2 Register Descriptions

### 36.2.1 IEBus Control Register (IECTR)

Address(es): 0008 A800h

b7	b6	b5	b4	b3	b2	b1	b0
—	IOL	DEE	—	RE	—	—	—

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	RE	Receive Enable	0: Reception is disabled. 1: Reception is enabled.	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	DEE	Broadcast Receive Error Interrupt Enable	0: A broadcast receive error is not generated up to the control field. 1: A broadcast receive error is generated up to the control field.	R/W
b6	IOL	Input/Output Level	0: Pin input/output is set to active low. (Logic 1 is low level and logic 0 is high level.) 1: Pin input/output is set to active high. (Logic 1 is high level and logic 0 is low level.)	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

#### RE Bit (Receive Enable)

This bit enables or disables IEB reception. The RE bit must be set at the initial setting before frame reception.

#### DEE Bit (Broadcast Receive Error Interrupt Enable)

If the DEE bit is set to 1, a broadcast reception error interrupt occurs when the receive buffer is not in the receive enabled state during broadcast reception (when the RE bit is not set to 1 or the IERSR.RXBSY flag is set). At this time, the master address is stored in IEBus reception master address registers 1 and 2.

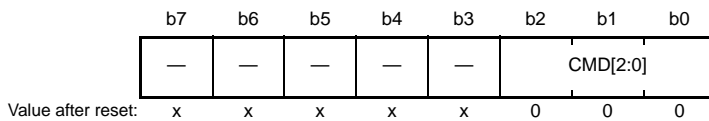
While the DEE bit is 0, a broadcast reception error interrupt does not occur when the receive buffer is not in the receive enabled state, and the reception stops and enters the wait state. The master address is not saved.

#### IOL Bit (Input/Output Level)

This bit selects the input/output pin level (polarity) for the IERXD and IETXD pins.

### 36.2.2 IEBus Command Register (IECMR)

Address(es): 0008 A801h



Value after reset:

x: Undefined

Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CMD[2:0]	Command	b2 b1 b0 0 0 0: No operation. Operation is not affected. 0 0 1: Cancels the lock required from other units*1 0 1 0: Requires communications as the master 0 1 1: Stops master communications*2 1 0 0: Undefined*3 1 0 1: Requires data transfer from the slave 1 1 0: Stops data transfer from the slave*4 1 1 1: Undefined*3	W
b7 to b3	—	Reserved	The read value is undefined. The write value should be 0.	W

Note 1. Do not execute this command in slave communications.

Note 2. This command is valid only during master communications (IEFLG.MRQ = 1). In other states, this command issuance is ignored. If this command is issued in master communications, the communications controller immediately enters the wait state. At this time, the issued master transmission request ends (IEFLG.MRQ = 0).

Note 3. Undefined bits. Issuing this command does not affect operation.

Note 4. This command is valid only during slave communications (IEFLG.SRQ = 1). In other states, this command issuance is ignored. Once this command is issued in slave transmission, the IEFLG.SRQ flag is 0 before slave transmission. Therefore, a transmit request from the master is not responded to. If a transmit request is issued during slave transmission, the transmission stops and the wait state is entered (IEFLG.SRQ = 0).

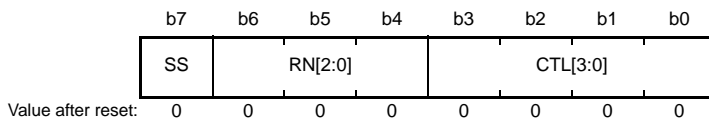
Since this register is a write-only register, the read value is undefined.

#### CMD[2:0] Bits (Command)

These bits issue a command to control IEB communications. While the IEFLG.CMX flag is set after the command issuance, the command is in execution. When the IEFLG.CMX flag becomes 0, the operation state is entered.

### 36.2.3 IEBus Master Control Register (IEMCR)

Address(es): 0008 A802h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	CTL[3:0]*1	Control	b3 b2 b1 b0 0 0 0 0: Reads slave status 0 0 0 1: Setting prohibited 0 0 1 0: Setting prohibited 0 0 1 1: Reads data and locks*2 0 1 0 0: Reads locked address (lower 8 bits) 0 1 0 1: Reads locked address (upper 4 bits) 0 1 1 0: Reads slave status and unlocks*2 0 1 1 1: Reads data 1 0 0 0: Setting prohibited 1 0 0 1: Setting prohibited 1 0 1 0: Writes command and locks*2 1 0 1 1: Writes data and locks*2 1 1 0 0: Setting prohibited 1 1 0 1: Setting prohibited 1 1 1 0: Writes command 1 1 1 1: Writes data	R/W
b6 to b4	RN[2:0]	Retransmission Count Setting	b6 b5 b4 0 0 0: 0 0 0 1: 1 0 1 0: 2 0 1 1: 3 1 0 0: 4 1 0 1: 5 1 1 0: 6 1 1 1: 7	R/W
b7	SS	Broadcast/Normal Communications Select	0: Broadcast communications for master communications 1: Normal communications for master communications	R/W

Note 1. CTL[3] determines the data transfer direction of the message length bits in the message length field and data bits in the data field:

CTL[3] = 1: Transfer is from master unit to slave unit

CTL[3] = 0: Transfer is from slave unit to master unit

Note 2. Control bits for lock and unlock

IEMCR sets the communication conditions for master communications.

#### CTL[3:0] Bits (Control)

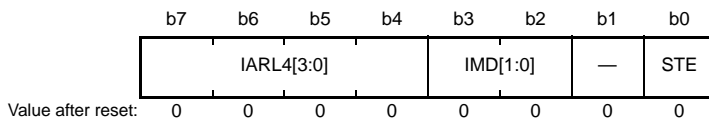
These bits set the control bits in the control field for master transmission.

#### RN[2:0] Bit (Retransmission Count Setting)

These bits set the number of times retransmission is done when arbitration is lost in master communications. If arbitration is lost, the TXEAL flag in IETSR is set and transmission ends.

### 36.2.4 IEBus Master Unit Address Register 1 (IEAR1)

Address(es): 0008 A803h



Bit	Symbol	Bit Name	Description	R/W
b0	STE	Slave Transmission Setting	0: Bit 4 in the slave status register is 0 (slave transmission stop state) 1: Bit 4 in the slave status register is 1 (slave transmission enabled state)	R/W
b1	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3, b2	IMD[1:0]	IEBus Communications Mode	b3 b2 0 0: Communications mode 0 0 1: Communications mode 1 1 0: Setting prohibited 1 1: Setting prohibited	R/W
b7 to b4	IARL4[3:0]	Lower 4 Bits of IEBus Master Unit Address	These bits set the lower four bits of the IEBus master unit address.	R/W

In master communications, the master unit address becomes the master address field value. In slave communications, the master unit address is compared with the received slave address field.

#### STE Bit (Slave Transmission Setting)

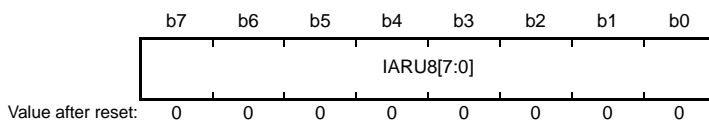
This bit sets bit 4 in the slave status register. Transmitting the slave status register informs the master unit that the slave transmission enabled state is entered by setting the STE bit to 1. Note that STE bit only sets the slave status register value and does not directly affect slave transmission.

#### IARL4[3:0] Bits (Lower 4 Bits of IEBus Master Unit Address)

These bits set the lower four bits of the master unit address. This value becomes the master address field value. In slave communications, the master unit address is compared with the received slave address field.

### 36.2.5 IEBus Master Unit Address Register 2 (IEAR2)

Address(es): 0008 A804h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	IARU8[7:0]	Upper 8 Bits of IEBus Master Unit Address	These bits set the upper eight bits of the IEBus master unit address.	R/W

In master communications, this value becomes the master address field value. In slave communications, this value is compared with the received slave address field.

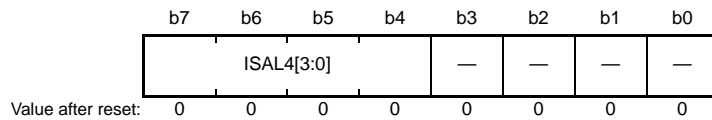


**IARU8[7:0] Bits (Upper 8 Bits of IEBus Master Unit Address)**

These bits set the upper eight bits of the master unit address. This value becomes the master address field value. In slave communications, the master unit address is compared with the received slave address field.

**36.2.6 IEBus Slave Address Setting Register 1 (IESA1)**

Address(es): 0008 A805h



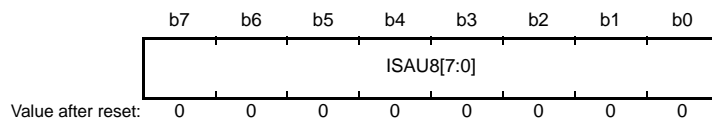
Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7 to b4	ISAL4[3:0]	Lower 4 Bits of IEBus Slave Address	These bits set the lower four bits of IEBus slave address.	R/W

**ISAL4[3:0] Bits (Lower 4 Bits of IEBus Slave Address)**

These bits set the lower four bits of the communication destination slave unit address.

**36.2.7 IEBus Slave Address Setting Register 2 (IESA2)**

Address(es): 0008 A806h



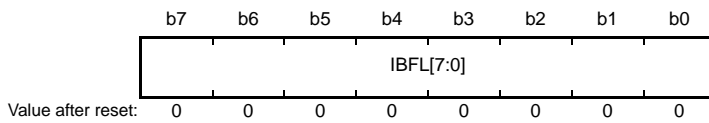
Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ISAU8[7:0]	Upper 8 Bits of IEBus Slave Address	These bits set the upper eight bits of the IEBus slave address.	R/W

**ISAU8[7:0] Bits (Upper 8 Bits of IEBus Slave Address)**

These bits set the upper eight bits of the communications destination slave unit address.

### 36.2.8 IEBus Transmit Message Length Register (IETBFL)

Address(es): 0008 A807h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	IBFL[7:0]	Transmit Message Length	01h: 1 byte 02h: 2 bytes : : 1Fh: 31 bytes 20h: 32 bytes 21h: Setting prohibited : : FFh: Setting prohibited 00h: Setting prohibited	R/W

IETBFL sets the message length for master or slave transmission.

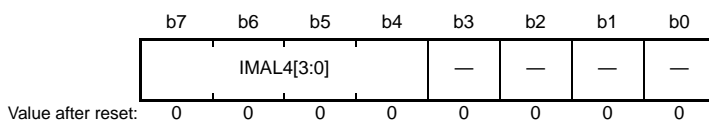
#### IBFL[7:0] Bits (Transmit Message Length)

These bits set the message length for master transmission.

Set the message length that does not exceed the maximum transmit bytes in communications mode.

### 36.2.9 IEBus Reception Master Address Register 1 (IEMA1)

Address(es): 0008 A809h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0 and cannot be modified.	R
b7 to b4	IMAL4[3:0]	Lower 4 Bits of IEBus Reception Master Address	These bits indicate the lower four bits of the master unit address in IEBus slave/broadcast reception.	R

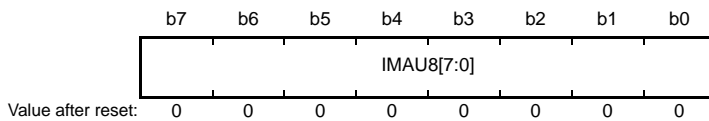
#### IMAL4[3:0] Bits (Lower 4 Bits of IEBus Reception Master Address)

These bits indicates the lower four bits of the communication destination master unit address in slave/broadcast reception. IEMA1 is enabled when slave/broadcast reception starts, and the contents are changed at the time of setting the IERSR.RXS flag.

If a broadcast receive error interrupt is selected by the DEE bit in IECTR and the receive buffer is not in the receive enabled state at control field reception, a receive error interrupt is generated and the lower four bits of the master address are stored in IEMA1.

### 36.2.10 IEBus Reception Master Address Register 2 (IEMA2)

Address(es): 0008 A80Ah



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	IMAU8[7:0]	Upper 8 Bits of IEBus Reception Master Address	These bits indicate the upper eight bits of the master unit address in slave/broadcast reception.	R

IEMA2 is enabled when slave/broadcast reception starts, and the contents are changed at the time of setting the IERSR.RXS flag.

If a broadcast receive error interrupt is selected with the DEE bit in IECTR and the receive buffer is not in the receive enabled state at control field reception, a receive error interrupt is generated and the upper eight bits of the master address are stored in IEMA2.

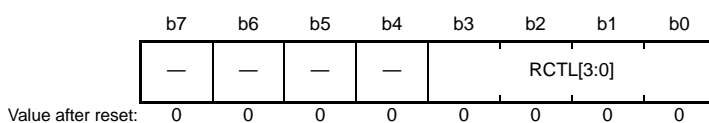
#### IMAU8[7:0] Bits (Upper 8 Bits of IEBus Reception Master Address)

These bits indicate the upper eight bits of the communications destination master unit address in slave/broadcast reception. IEMA2 is enabled when slave/broadcast reception starts, and the contents are changed at the time of setting the IERSR.RXS flag.

If a broadcast receive error interrupt is selected by the DEE bit in IECTR and the receive buffer is not in the receive enabled state at control field reception, a receive error interrupt is generated and the upper eight bits of the master address are stored in IEMA2.

### 36.2.11 IEBus Receive Control Field Register (IERCTL)

Address(es): 0008 A80Bh



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	RCTL[3:0]	IEBus Receive Control Field	These bits indicate the control field value in slave/broadcast reception.	R
b7 to b4	—	Reserved	These bits are read as 0 and cannot be modified.	R

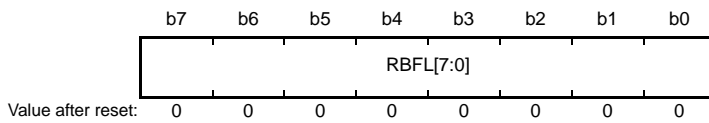
IERCTL is enabled when slave/broadcast receive starts, and the contents are changed at the time of setting the RXS flag in IERSR.

#### RCTL[3:0] Bits (IEBus Receive Control Field)

These bits indicate the control field value in slave/broadcast reception. IERCTL is enabled when slave/broadcast reception starts, and the contents are changed at the time of setting the IERSR.RXS flag.

### 36.2.12 IEBus Receive Message Length Register (IERBFL)

Address(es): 0008 A80Ch

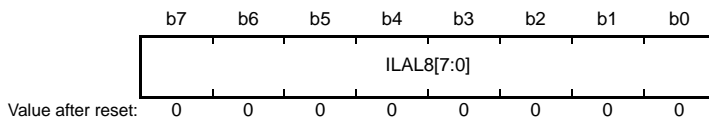


Bit	Symbol	Bit Name	Description	R/W
b7 to b0	RBFL[7:0]	IEBus Receive Message Length	These bits store the contents of the message length field in slave/broadcast reception.	R

IERBFL is enabled when slave/broadcast receive starts, and the contents are changed at the time of setting the RXS flag in IERSR.

### 36.2.13 IEBus Lock Address Register 1 (IELA1)

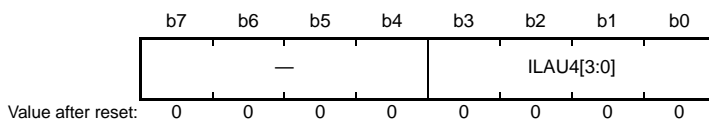
Address(es): 0008 A80Eh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	ILAL8[7:0]	Lower 8 Bits of IEBus Locked Address	These bits indicate the lower eight bits of the master unit address when a unit is locked. These bits are valid only when the LCK flag in IEFLG is set.	R

### 36.2.14 IEBus Lock Address Register 2 (IELA2)

Address(es): 0008 A80Fh



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ILAU4[3:0]	Upper 4 Bits of IEBus Locked Address	These bits store the upper four bits of the master unit address when a unit is locked. These bits are valid only when the LCK flag in IEFLG is set.	R
b7 to b4	—	Reserved	These bits are read as 0 and cannot be modified.	R

### 36.2.15 IEBus General Flag Register (IEFLG)

Address(es): 0008 A810h

b7	b6	b5	b4	b3	b2	b1	b0
CMX	MRQ	SRQ	SRE	LCK	—	RSS	GG
Value after reset:	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	GG	General Broadcast Reception Acknowledgement Flag	0: A unit is in slave reception When FFFh is not acknowledged in the slave address field in broadcast reception 1: When FFFh is acknowledged in the slave address field in broadcast reception	R
b1	RSS	Receive Broadcast Flag	0: Received broadcast bit is 0 1: Received broadcast bit is 1	R
b2	—	Reserved	This bit is read as 0 and cannot be modified.	R
b3	LCK	Lock Status Indication Flag	0: A unit is unlocked 1: A unit is locked	R
b4	SRE	Slave Receive Status Flag	0: Slave/broadcast reception is not being executed 1: Slave/broadcast reception is being executed	R
b5	SRQ	Slave Transmission Request Flag	0: The unit is not in the transmit request state as a slave unit 1: The unit is in the transmit request state as a slave unit	R
b6	MRQ	Master Communications Request Flag	0: The unit is not in the communications request state as a master unit 1: The unit is in the communications request state as a master unit	R
b7	CMX	Command Execution Status Flag	0: Command execution is completed 1: A command is being executed	R

#### GG Flag (General Broadcast Reception Acknowledgement)

This flag is set to 1 when the slave address is acknowledged as FFFh in broadcast reception. Like the receive broadcast bit, this flag is valid when the slave/broadcast reception is started. (This flag is changed at the time of setting the RXS flag in IERSR.)

The previous value remains unchanged until the next slave/broadcast reception is started. This flag is 0 in slave normal reception.

#### RSS Flag (Receive Broadcast)

This flag indicates the received broadcast bit value. This flag is valid when the slave/broadcast reception is started. (This flag is changed at the time of setting the RXS flag in IERSR.)

The previous value remains unchanged until the next slave/broadcast reception is started.

#### LCK Flag (Lock Status Indication)

This flag is set to 1 when a unit is locked by a lock request from the master unit. The IELA1 and IELA2 values are valid only when the LCK flag is set to 1.

[Setting condition]

- When data for the number of bytes specified by the message length is not received/transmitted after the control bits (3h, Ah, Bh) that make the unit locked are received from the master unit. (The LCK flag is set to 1 only when a transmit/receive-frame maximum transfer byte overflow is generated. This flag is not set by completion with other errors.)

[Clearing condition]

- When an unlock condition is satisfied or when an unlock command is issued.

### **SRE Flag (Slave Receive Status)**

This flag indicates the execution status in slave/broadcast reception.

[Setting condition]

- When the slave/broadcast reception is started while the RE bit in IECTR is set to 1.

[Clearing condition]

- When the slave/broadcast reception has been completed.

### **SRQ Flag (Slave Transmission Request)**

This flag indicates whether the unit is in the transmit request state as a slave unit.

[Setting condition]

- When the CMX flag is cleared to 0 after the slave transmit request command is issued.

[Clearing condition]

- When a slave transmission has been completed.

### **MRQ Flag (Master Communications Request)**

This flag indicates whether the unit is in the communications request state as a master unit.

[Setting condition]

- When the CMX flag is cleared to 0 after the master communications request command is issued.

[Clearing condition]

- When the master communications have been completed.

### **CMX Flag (Command Execution Status)**

This flag indicates the command execution status.

[Setting condition]

- When a master communications request or slave transmit request command is issued while the MRQ, SRQ, or SRE flag is set.

[Clearing condition]

- When a command execution has been completed.

### 36.2.16 IEBus Transmit Status Register (IETSR)

Address(es): 0008 A811h

b7	b6	b5	b4	b3	b2	b1	b0
—	TXS	TXF	—	TXEAL	TXETT ME	TXERO	TXEAC K
0	0	0	0	0	0	0	0

Value after reset:

Bit	Symbol	Bit Name	Description	R/W
b0	TXEACK	Acknowledge Flag	0: 1 (NAK) has not been detected in the acknowledge bit. 1: 1 (NAK) has been detected in the acknowledge bit.	R/(W) *1
b1	TXERO	Transmit-Frame Maximum Transfer Byte Overflow Flag	0: Transmit has been completed after the maximum number of bytes defined by the communications mode have been transmitted. 1: Transmit has not been completed although the maximum number of bytes defined by the communications mode have been transmitted.	R/(W) *1
b2	TXETTME	Transmit Timing Error Flag	0: No timing error has occurred during data transmission. 1: A timing error has occurred during data transmission.	R/(W) *1
b3	TXEAL	Arbitration Loss Flag	0: Normal transmission 1: Arbitration has been lost during data transmission and the transmission has been terminated.	R/(W) *1
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	TXF	Transmit Normal Completion Flag	0: Data has not been transmitted for the number of bytes specified by the message length bits. 1: Data has been transmitted for the number of bytes specified by the message length bits.	R/(W) *1
b6	TXS	Transmit Start Flag	0: The master address field transmission has not been completed in master transmission. 1: Arbitration has been won and the master address field transmission has been completed in master transmission.	R/(W) *1
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

Note 1. Only 1 can be written to clear the flag.

Each status flag in IETSR corresponds to a bit in the IEBus transmit interrupt enable register (IEIET) that enables or disables each interrupt. IETSR is cleared by writing 1 to each bit.

#### TXEACK Flag (Acknowledge)

This flag indicates the data received in the acknowledge bit of the data field.

- Acknowledge bit other than in the data field  
The IEB terminates the transmission and enters the wait state if a NAK is received. In this case, the TXEACK flag is set to 1.
- Acknowledge bit in the data field  
The IEB retransmits data up to the maximum number of bytes defined by the communications mode until an ACK is received from the receive unit if a NAK is received from the receive unit during data field transmission. In this case, when an ACK is received from the receive unit during retransmission, the TXEACK flag is not set and transmission will be continued. When transmission is terminated without receiving an ACK within the maximum number of bytes defined by the communications mode, the TXEACK flag is set to 1.

Note: • The TXEACK flag is invalid in broadcast communications.

[Setting condition]

- When the acknowledge bit of 1 (NAK) is detected.

[Clearing condition]

- When 1 is written.

### **TXERO Flag (Transmit-Frame Maximum Transfer Byte Overflow)**

This flag indicates that the maximum number of bytes defined by the communications mode have been transmitted because a NAK has been received from the receive unit and retransmit has been performed, or that transmission has not been completed because the message length value exceeds\*1 the maximum number of transfer bytes in one frame. The IEB sets the TXERO flag and enters the wait state.

Note 1. When the value specified as the message length is greater than the maximum number of transfer bytes, the IEB terminates communications after transmitting the first byte data.

[Setting condition]

- When the transmit has not been completed although the maximum number of bytes defined by the communications mode have been transmitted.

[Clearing condition]

- When 1 is written.

### **TXETTME Flag (Transmit Timing Error)**

This flag is set to 1 if data is not transmitted with the timing specified by the IEB protocol during data transmission. The IEB sets the TXETTME flag and enters the wait state.

[Setting condition]

- When a timing error occurs during data transmission up to the maximum number of bytes defined by the communications mode.

[Clearing condition]

- When 1 is written.

### **TXEAL Flag (Arbitration Loss)**

The IEB retransmits from the start bit for the number of times specified by setting the RN bits in IEMCR if the arbitration has been lost in master communications. If the arbitration has been lost for the specified number of times, the TXEAL flag is set to enter the wait state. If the arbitration has been won within retransmit for the specified number of times, the TXEAL flag is not set to 1. The TXEAL flag is set only when the arbitration has been lost and the wait state is entered.

[Setting condition]

- When the arbitration has been lost during data transmission and the transmission has been terminated.

[Clearing condition]

- When 1 is written.

### **TXF Flag (Transmit Normal Completion)**

This flag indicates that data for the number of bytes specified by the message length bits has been transmitted with no error.

[Setting condition]

- When data for the number of bytes specified by the message length bits has been transmitted during transfer up to the maximum transfer bytes per frame.



[Clearing condition]

- When 1 is written.

### TXS Flag (Transmit Start)

This flag indicates that the IEB starts transmission.

[Setting condition]

- When the arbitration is won and the master address field transmission is completed in master transmission.

[Clearing condition]

- When 1 is written.

## 36.2.17 IEBus Transmit Interrupt Enable Register (IEIET)

Address(es): 0008 A812h

	b7	b6	b5	b4	b3	b2	b1	b0
	—	TXSE	TXFE	—	TXEAL E	TXETT MEE	TXERO E	TXEAC KE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	TXEACKE	Acknowledge Interrupt Enable	0: Disables an acknowledge bit (TXEACK) interrupt 1: Enables an acknowledge bit (TXEACK) interrupt	R/W
b1	TXEROE	Transmit-Frame Maximum Transfer Byte Overflow Interrupt Enable	0: Disables a transmit-frame maximum transfer byte overflow (TXERO) interrupt 1: Enables a transmit-frame maximum transfer byte overflow (TXERO) interrupt	R/W
b2	TXETTMEE	Transmit Timing Error Interrupt Enable	0: Disables a transmit timing error (TXETTME) interrupt 1: Enables a transmit timing error (TXETTME) interrupt	R/W
b3	TXEALE	Arbitration Loss Interrupt Enable	0: Disables an arbitration loss (TXEAL) interrupt 1: Enables an arbitration loss (TXEAL) interrupt	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b5	TXFE	Transmit Normal Completion Interrupt Enable	0: Disables a transmit normal completion (TXF) interrupt 1: Enables a transmit normal completion (TXF) interrupt	R/W
b6	TXSE	Transmit Start Interrupt Enable	0: Disables a transmit start (TXS) interrupt 1: Enables a transmit start (TXS) interrupt	R/W
b7	—	Reserved	This bit is read as 0. The write value should be 0.	R/W

## 36.2.18 IEBus Receive Status Register (IERSR)

Address(es): 0008 A814h

b7	b6	b5	b4	b3	b2	b1	b0
RXBSY	RXS	RXF	RXEDE	RXEOVE	RXERTME	RXEDLE	RXEPE

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	RXEPE	Parity Error Flag	0: No parity error has occurred 1: A parity error has occurred	R/(W) *1
b1	RXEDLE	Receive-Frame Maximum Transfer Byte Overflow Flag	0: No overrun error has occurred 1: An overrun error has occurred	R/(W) *1
b2	RXERTME	Receive Timing Error Flag	0: No receive timing error has occurred 1: A receive timing error has occurred	R/(W) *1
b3	RXEOVE	Receive Overrun Flag	0: No receive overrun error has occurred 1: A receive overrun error has occurred	R/(W) *1
b4	RXEDE	Broadcast Receive Error Flag	0: No broadcast receive error has occurred 1: A broadcast receive error has occurred	R/(W) *1
b5	RXF	Receive Normal Completion Flag	0: Reception has not been completed successfully 1: Reception has been completed successfully	R/(W) *1
b6	RXS	Receive Start Flag	0: Reception has not started 1: Reception has started	R/(W) *1
b7	RXBSY	Receive Busy Flag	0: Receive data buffer is ready for reception 1: Receive data buffer is busy	R/(W) *1

Note 1. Only 1 can be written to clear the flag.

Each status flag in IERSR corresponds to a bit in the IEIER that enables or disables each interrupt. IERSR is cleared by writing 1 to each bit.

**RXEPE Flag (Parity Error)**

This flag indicates that a parity error has occurred during data field reception. If a parity error occurs before data field reception, the IEB immediately enters the wait state and the RXEPE flag is not set. If a parity error occurs during data field reception when the maximum number of transfer bytes in one frame have not been received, the RXEPE flag is not set yet. When a parity error occurs, the IEB returns a NAK to the communications destination unit via the acknowledge bit. In this case, the communications destination unit continues retransfer up to the maximum number of transfer bytes in one frame and if the reception has been completed normally with no parity error, the RXEPE flag is not set. If the parity error is still found when the reception is terminated and wait state is entered before receiving data for the number of bytes specified by the message length, the RXEPE flag is set. In broadcast reception, if a parity error occurs during data field reception, the IEB enters the wait state immediately after setting the RXEPE flag. This flag is enabled only after the receive start flag (RXS) is set. If this error occurs before the receive start flag is set, the IEB stops communication and enters the wait state. This flag is not set in this case.

[Setting condition]

- When the parity bit of the last receive data of the data field is not correct after the maximum number of transfer bytes have been received.

[Clearing condition]

- When 1 is written.

**RXEDLE Flag (Receive-Frame Maximum Transfer Byte Overflow)**

This flag indicates that the data reception has not finished within the maximum number of bytes defined by the communications mode because of a parity error or overrun error causing the retransfer of data, or that reception has not been completed because the message length value exceeds\*1 the maximum number of transfer bytes in one frame. The IEB sets the RXEDLE flag and enters the wait state. This flag is enabled only after the receive start flag (RXS) is set. If this error occurs before the receive start flag is set, the IEB stops communication and enters the wait state. This flag is not set in this case.

Note 1. When the value specified as the message length is greater than the maximum number of transfer bytes, the IEB terminates communications after receiving the first byte data.

[Setting condition]

- When the reception of the value specified as the message length has not been completed within the maximum number of bytes defined by communications mode.

[Clearing condition]

- When 1 is written.

**RXERTME Flag (Receive Timing Error)**

This flag is set to 1 if data is not received with the timing specified by the IEB protocol during data reception. The IEB sets the RXERTME flag and enters the wait state. This flag is enabled only after the receive start flag (RXS) is set. If this error occurs before the receive start flag (RXS) is set, the IEB stops communication and enters the wait state. The RXERTME flag is not set in this case.

[Setting condition]

- When a timing error occurs during data reception of the maximum number of bytes defined by communications mode.

[Clearing condition]

- When 1 is written.

**RXEOVF Flag (Receive Overrun)**

This flag indicates the overrun during data reception.

The IEB sets the RXEOVF flag when the IEB receives the next byte data while the receive data has not been read (the RXBSY flag is not cleared). In this case, the IEB assumes that an overrun error has occurred and returns a NAK to the communications destination unit. The communications destination unit retransmits data up to the maximum number of transfer bytes. The IEB, however, returns a NAK when the RXBSY flag remains set. If the RXBSY flag is cleared to 0, the IEB returns an ACK and receives the next data. In broadcast reception, if the RXBSY flag is set when data reception starts, the IEB immediately enters the wait state. This flag becomes enabled only after the receive start flag (RXS) is set.

[Setting condition]

- When the next byte data is received while the RXBSY flag is not cleared.

[Clearing condition]

- When 1 is written.

**RXEDE Flag (Broadcast Receive Error)**

This flag indicates that data could not be received because the receive buffer is not in the receive enabled state (when the IECTR.RE bit is not set to 1 or the IERSR.RXBSY flag is set.) during control field reception in broadcast reception. The RXEDE flag functions when the DEE bit in IECTR is set to 1.

[Setting condition]

- When data could not be received during broadcast reception.

[Clearing condition]

- When 1 is written.

**RXF Flag (Receive Normal Completion)**

This flag indicates that data for the number of bytes specified by the message length bits has been received successfully.

[Setting condition]

- When data for the number of bytes specified by the message length bits has been received successfully.

[Clearing condition]

- When 1 is written.

**RXS Flag (Receive Start)**

This flag indicates that the IEB has started reception.

[Setting condition]

- When the data up to message length field has been received correctly from the master unit in slave reception.

[Clearing condition]

- When 1 is written.

**RXBSY Flag (Receive Busy)**

This flag indicates that the receive data is stored in the receive data buffer (IERB001 to IERB032). Clear this flag after reading out all data. The next receive data cannot be received while the RXBSY flag is set.

[Setting condition]

- When all receive data has been written to the receive data buffer.

[Clearing condition]

- When 1 is written.

## 36.2.19 IEBus Receive Interrupt Enable Register (IEIER)

Address(es): 0008 A815h

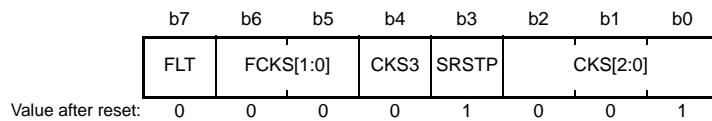
b7	b6	b5	b4	b3	b2	b1	b0
RXBSY E	RXSE	RXFE	RXEDE E	RXEOV EE	RXERT MEE	RXEDL EE	RXEPE E

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b0	RXEPEE	Parity Error Interrupt Enable	0: Disables a parity error (RXEPE) interrupt 1: Enables a parity error (RXEPE) interrupt	R/W
b1	RXEDLEE	Receive-Frame Maximum Transfer Byte Overflow Interrupt Enable	0: Disables a receive-frame maximum transfer byte overflow (RXEDLE) interrupt 1: Enables a receive-frame maximum transfer byte overflow (RXEDLE) interrupt	R/W
b2	RXERTMEE	Receive Timing Error Interrupt Enable	0: Disables a receive timing error (RXERTME) interrupt 1: Enables a receive timing error (RXERTME) interrupt	R/W
b3	RXEOVEE	Receive Overrun Interrupt Enable	0: Disables an overrun control flag (RXEOVE) interrupt 1: Enables an overrun control flag (RXEOVE) interrupt	R/W
b4	RXEDEE	Broadcast Receive Error Interrupt Enable	0: Disables a broadcast receive error (RXEDE) interrupt 1: Enables a broadcast receive error (RXEDE) interrupt	R/W
b5	RXFE	Receive Normal Completion Interrupt Enable	0: Disables a receive normal completion (RXF) interrupt 1: Enables a receive normal completion (RXF) interrupt	R/W
b6	RXSE	Receive Start Interrupt Enable	0: Disables a receive start (RXS) interrupt 1: Enables a receive start (RXS) interrupt	R/W
b7	RXBSYE	Receive Busy Interrupt Enable	0: Disables a receive busy (RXBSY) interrupt 1: Enables a receive busy (RXBSY) interrupt	R/W

### 36.2.20 IEBus Clock Selection Register (IECKSR)

Address(es): 0008 A818h



Bit	Symbol	Bit Name	Description	R/W
b2 to b0	CKS[2:0]	Input Clock Selection*1	b2 b1 b0 0 0 0: Setting prohibited 0 0 1: The peripheral clock (IECLK) divided by two is used (IECLK = 12 MHz, 12.58 MHz) 0 1 0: The peripheral clock (IECLK) divided by three is used (IECLK = 18 MHz, 18.87 MHz) 0 1 1: The peripheral clock (IECLK) divided by four is used (IECLK = 24 MHz, 25.16 MHz) 1 0 0: The peripheral clock (IECLK) divided by five is used (IECLK = 30 MHz, 31.45 MHz) 1 0 1: The peripheral clock (IECLK) divided by six is used (IECLK = 36 MHz, 37.74 MHz) 1 1 0: The peripheral clock (IECLK) divided by seven is used (IECLK = 42 MHz, 44.03 MHz) 1 1 1: Setting prohibited	R/W
b3	SRSTP	Software Reset	0: Software reset is negated 1: Software reset is asserted	R/W
b4	CKS3	Input Clock Selection 3*1,*2	0: Peripheral clock (IECLK) is used 1: Setting prohibited	R/W
b6, b5	FCKS[1:0]	Digital Filter Clock Selection*1	b6 b5 0 0: 1/1 divided peripheral clock (IECLK) is used 0 1: 1/2 divided peripheral clock (IECLK) is used 1 0: 1/3 divided peripheral clock (IECLK) is used 1 1: 1/4 divided peripheral clock (IECLK) is used	R/W
b7	FLT	Digital Filter Enable*1	0: Disables digital filter function 1: Enables digital filter function	R/W

Note 1. Do not change the settings of the FLT, FCKS[1:0], CKS3, and CKS[2:0] bits while IEBus is in transmit/receive operation. IECKSR should be set according to the flowchart for initial setting shown in Figure 36.6. Otherwise, the transmit/receive operation is not guaranteed.

Note 2. The clock supply is stopped except for IECKSR when the MSTPCRC.MSTPC18 bit is set to 1. To use the IEB without specifying low power consumption mode, be sure to clear the MSTPCRC.MSTPC18 bit to 0 to cancel module stop mode.

#### CKS[2:0] Bits (Input Clock Selection)

These bits select the division ratio for the clock used in the IEB.

When using the digital filter function, the digital filter clock should be set as follows so that the digital filter clock becomes an integral multiple of the clock used in the IEB.

- When the CKS[2:0] bits are set to 001b, the FCKS[1:0] bits should be set to 00b or 01b.
- When the CKS[2:0] bits are set to 010b, the FCKS[1:0] bits should be set to 00b or 10b.
- When the CKS[2:0] bits are set to 011b, the FCKS[1:0] bits should be set to 00b, 01b, or 11b.
- When the CKS[2:0] bits are set to 100b, the FCKS[1:0] bits should be set to 00b.
- When the CKS[2:0] bits are set to 101b, the FCKS[1:0] bits should be set to 00b, 01b, or 10b.
- When the CKS[2:0] bits are set to 110b, the FCKS[1:0] bits should be set to 00b.

**SRSTP Bit (Software Reset)**

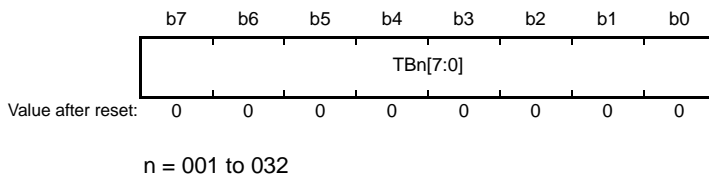
When a hardware reset is executed, it sets the software reset and then all registers except the IECKSR register are reset. After the hardware reset, negate the software reset.

Since registers except the IECKSR register have been reset while the software reset is being asserted, accessing to the registers except the IECKSR register is prohibited.

In addition, if bits FLT, FCKS[1:0], CKS3, and CKS[2:0] are modified without following the flowchart for initial setting shown in Figure 36.6, assert or negate the software reset to prevent a malfunction.

### 36.2.21 IEBus Transmit Data Buffer Registers 001 to 032 (IETB001 to IETB032)

Address(es): 0008 A900h to 0008 A91Fh



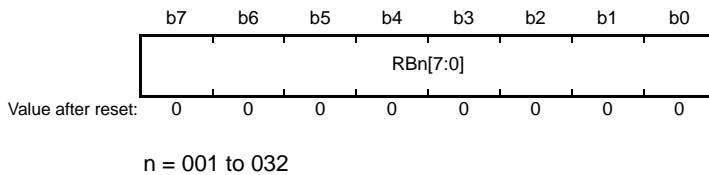
Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TBn[7:0]	IEBus Transmit Data Buffer	Data to be transmitted in the data field during master transmission is written to TB001 to TB032. Data is written starting with TB001 for the start 1-byte data, followed by TB002 and TB003 and so on according to the transmission order, and TB032 stores the last data of 32-byte transmission.	W*1

Note 1. Since writing to these bits during master or slave transmission (IEFLG.MRQ bit is 1, or IEFLG.SRQ is 1) is disabled, accessing to these bits is prohibited.

IETB001 to IETB032 works as a 32-byte ( $8 \times 32$ ) buffer to which data to be transmitted during master transmission is written.

### 36.2.22 IEBus Receive Data Buffer Registers 001 to 032 (IERB001 to IERB032)

Address(es): 0008 AA00h to 0008 AA1Fh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	RBn[7:0]	IEBus Receive Data Buffer	Data in RB001 to RB032 can be read when the RXBSY bit in the IEBus receive status register (IERSR) is set to 1. Data read from RB001 to RB032 is the field data during slave reception. Receive data is written starting with RB001 for the start 1-byte data, followed by RB002 and RB003 and so on, and RB032 stores the last data of 32-byte reception.	R*1

Note 1. Since reading these bits during master reception (after master communications request is issued while IEMCR.CTL[3] is 0, IEFLG.MRQ is 1 and IERSR.RXBSY is 0) or slave reception (IEFLG.SRE is 1 and IERSR.RXBSY is 0) is disabled, accessing to these bits is prohibited. (Read value is undefined.)

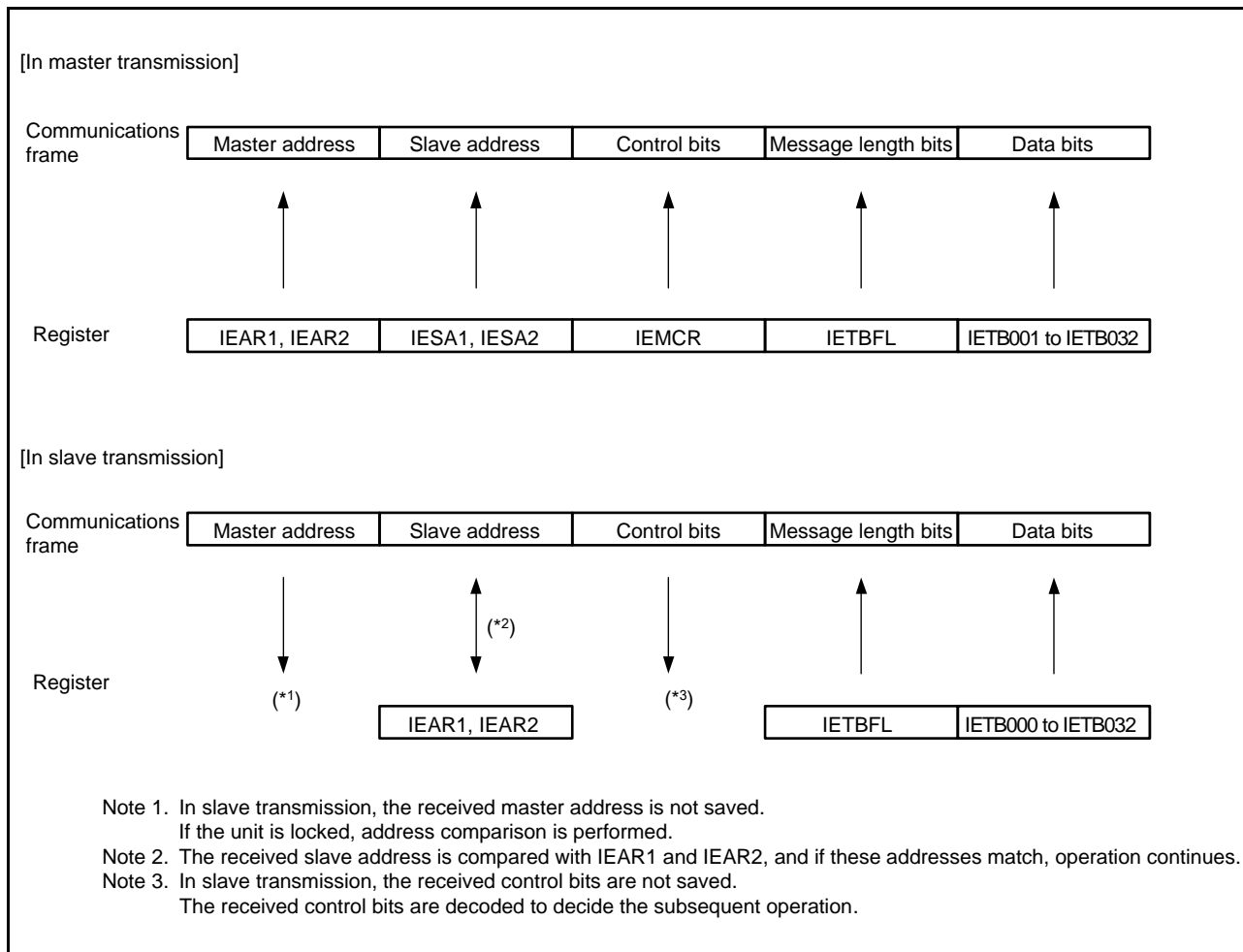
IERB001 to IERB032 work as a 32-byte ( $8 \times 32$ ) buffer in which data received during slave reception is stored.



### 36.3 Data Format

#### 36.3.1 Transmission Format

Figure 36.4 shows the relationship between the transfer format and each register during the IEBus data transmission.



**Figure 36.4 Relationship between Transfer Format and Each Register during IEBus Data Transmission**

### 36.3.2 Reception Format

Figure 36.5 shows the relationship between the transfer format and each register during the IEBus data reception.

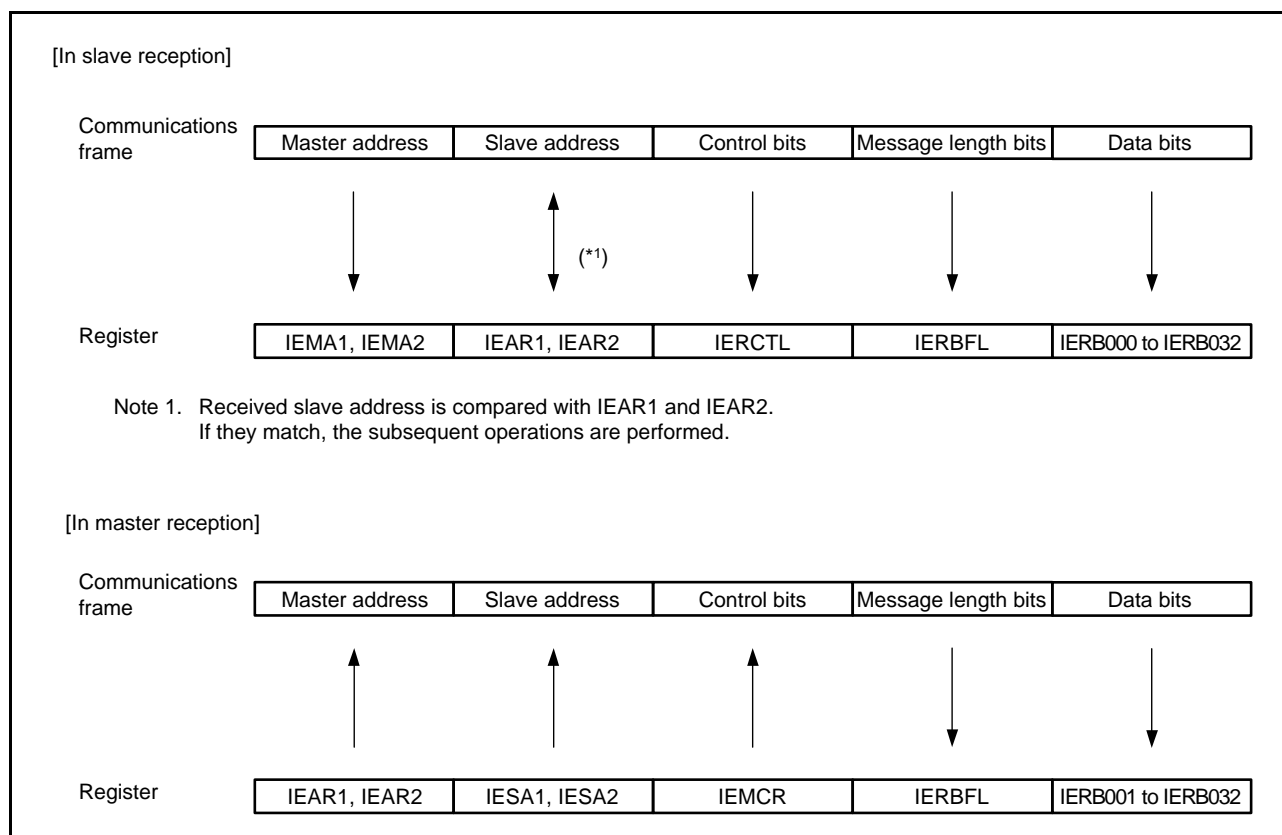


Figure 36.5 Relationship between Transfer Format and Each Register during IEBus Data Reception

## 36.4 Control Flows

### 36.4.1 Initial Setting

Figure 36.6 shows the flowchart for the initial setting.

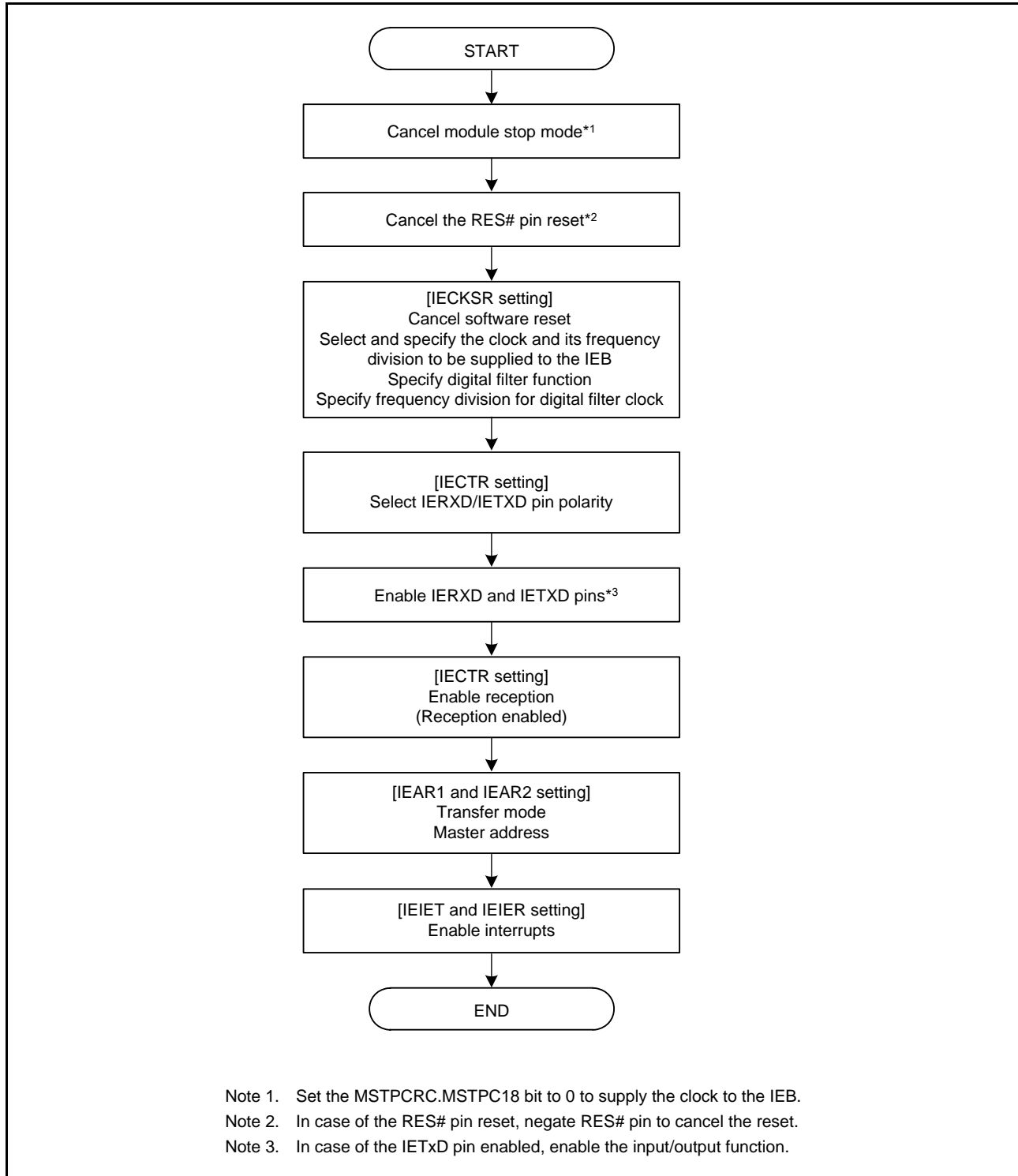


Figure 36.6 Flowchart for Initial Setting

### 36.4.2 Master Transmission

Figure 36.7 shows the flowchart for master transmission.

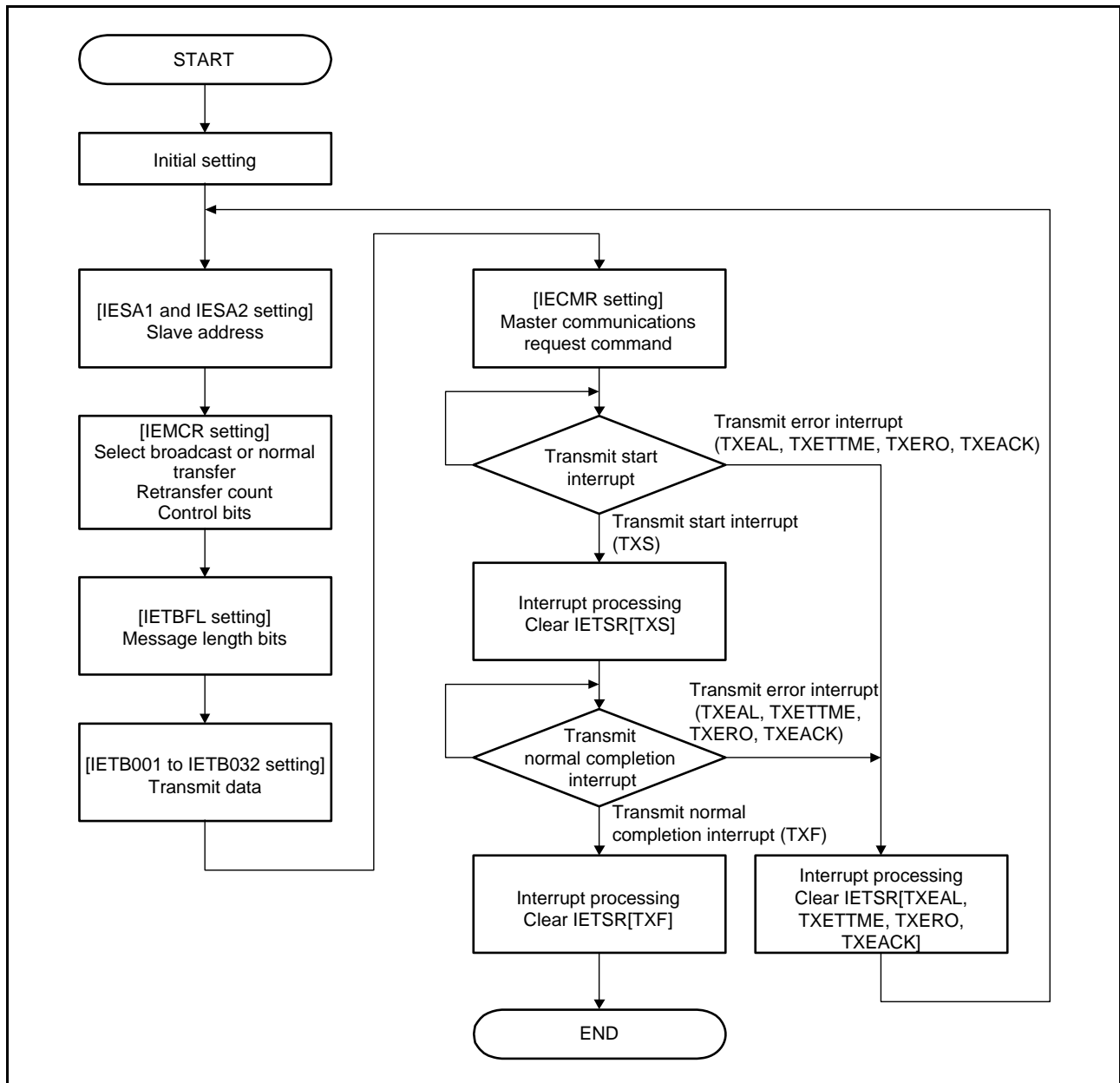


Figure 36.7 Flowchart for Master Transmission

### 36.4.3 Slave Reception

Figure 36.8 shows the flowchart for slave reception.

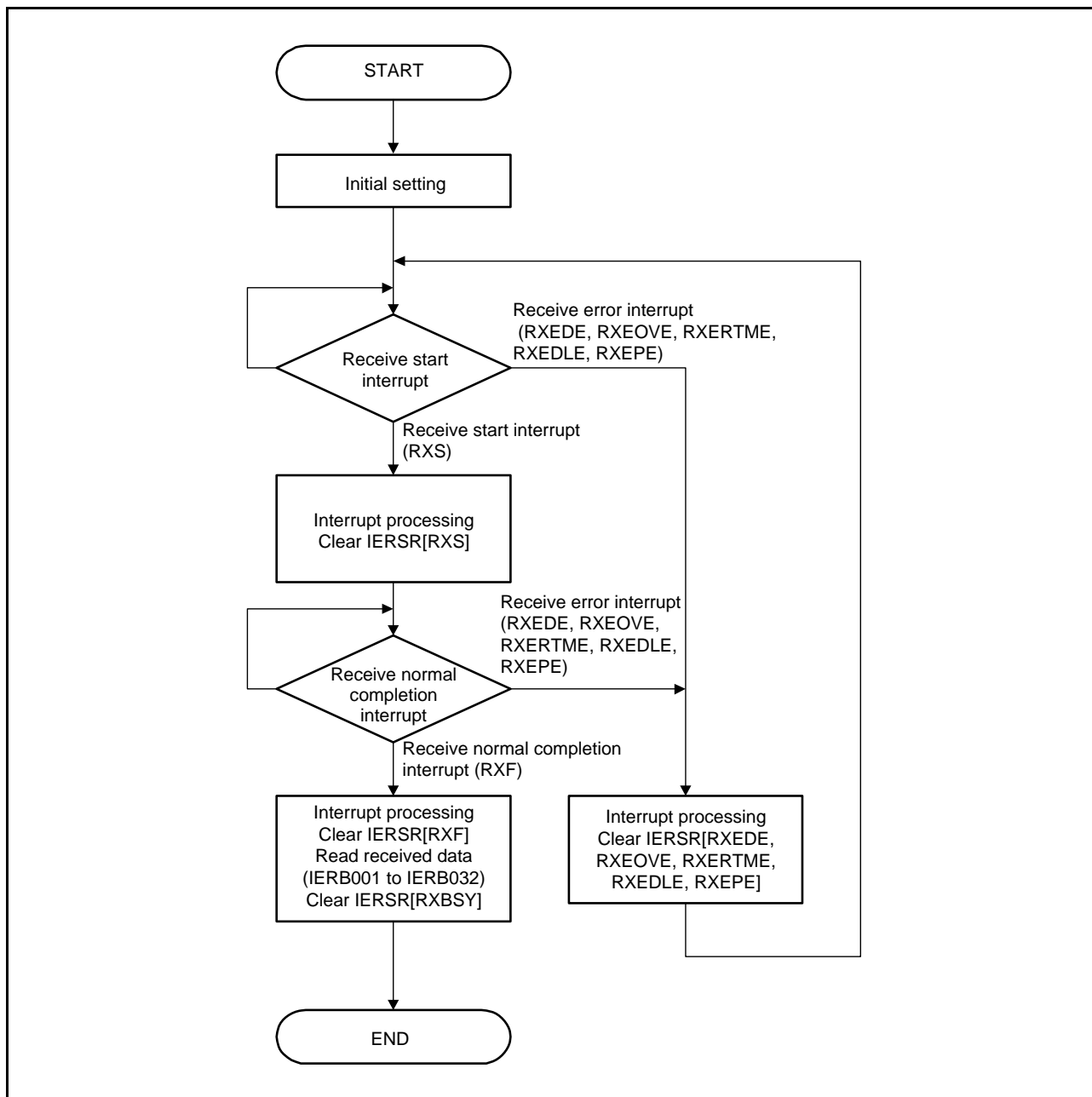


Figure 36.8 Flowchart for Slave Reception

### 36.4.4 Master Reception

Figure 36.9 shows the flowchart for master reception.

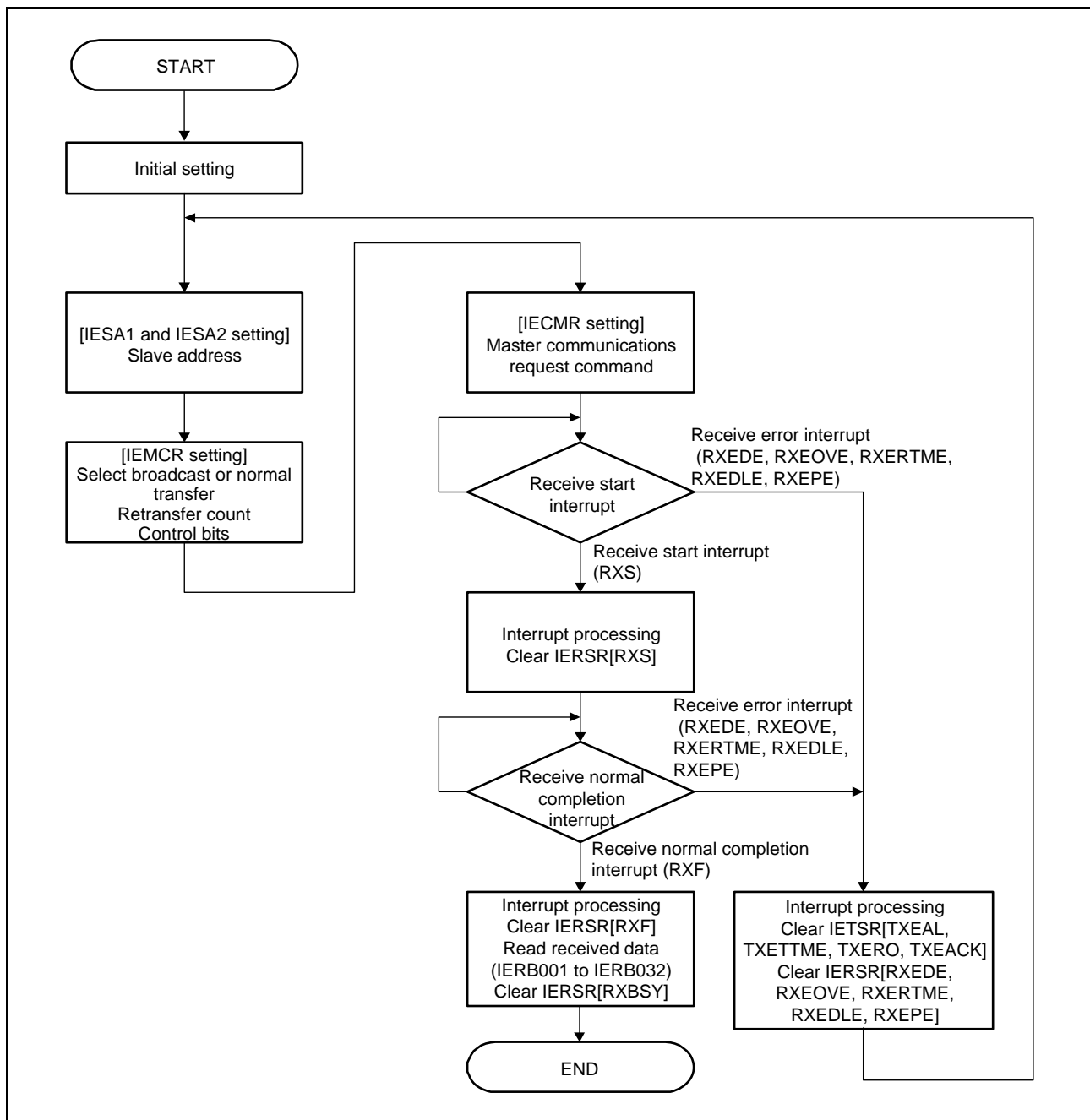


Figure 36.9 Flowchart for Master Reception

### 36.4.5 Slave Transmission

Figure 36.10 shows the flowchart for slave transmission.

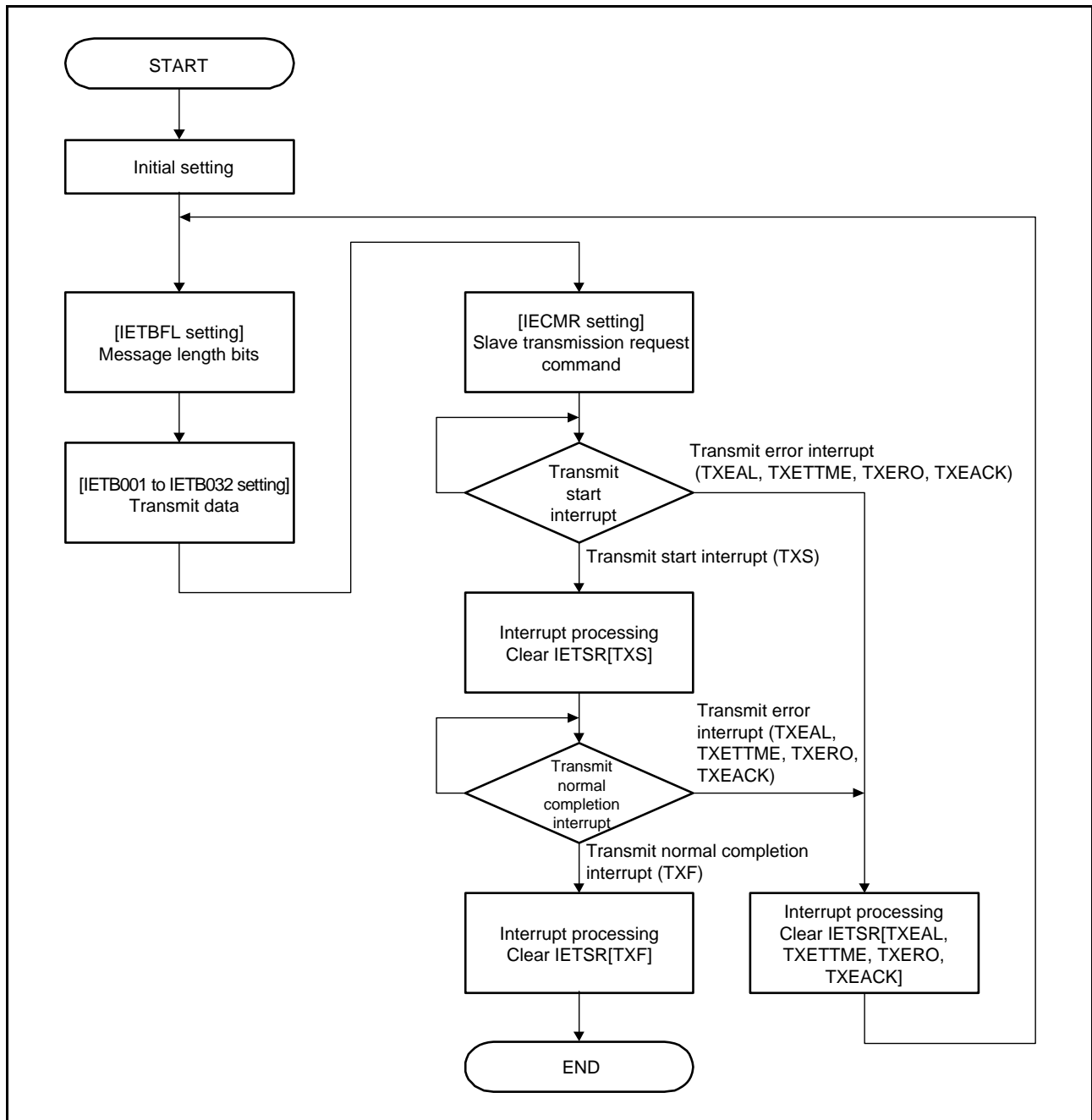


Figure 36.10 Flowchart for Slave Transmission

### 36.5 Operation Timing

#### 36.5.1 Master Transmission

Figure 36.11 shows the timing for master transmission.

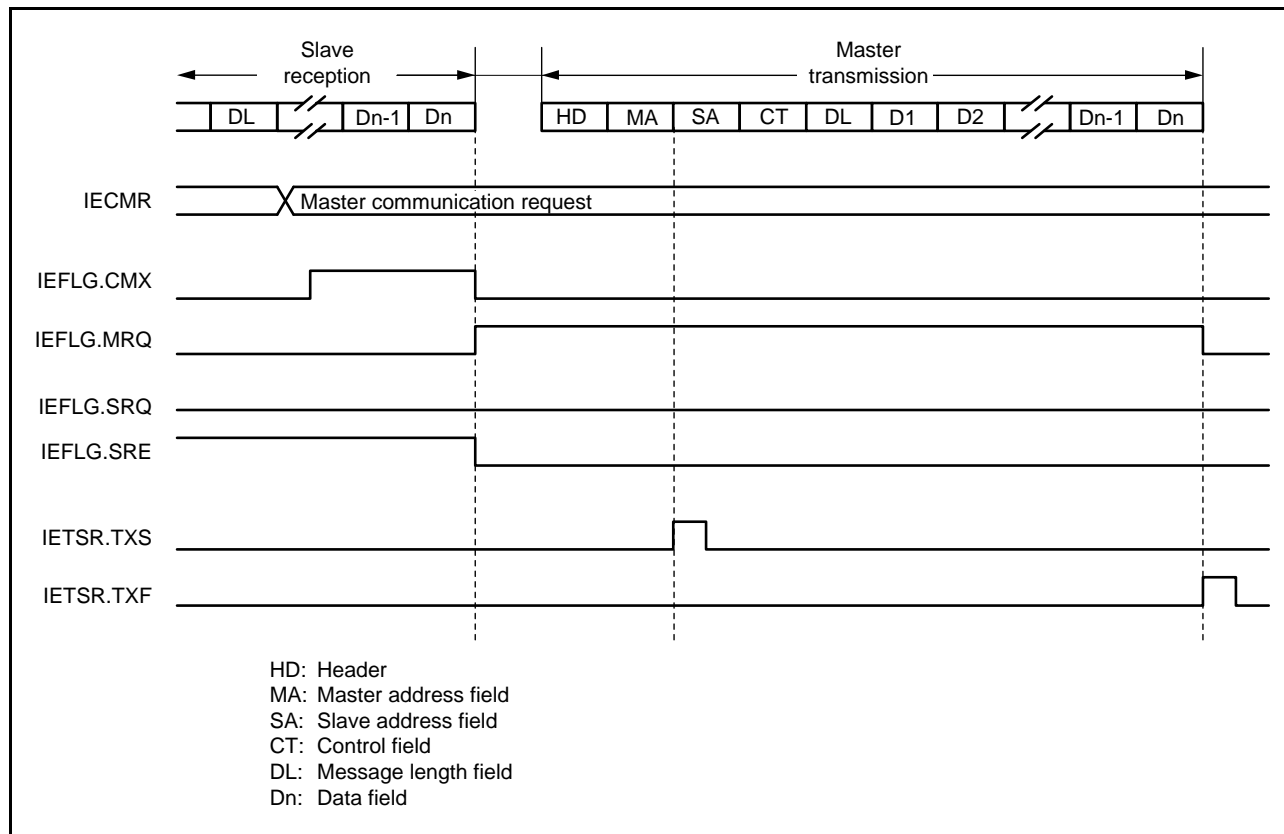


Figure 36.11 Master Transmission Timing



### 36.5.2 Slave Reception

Figure 36.12 shows the timing for slave reception.

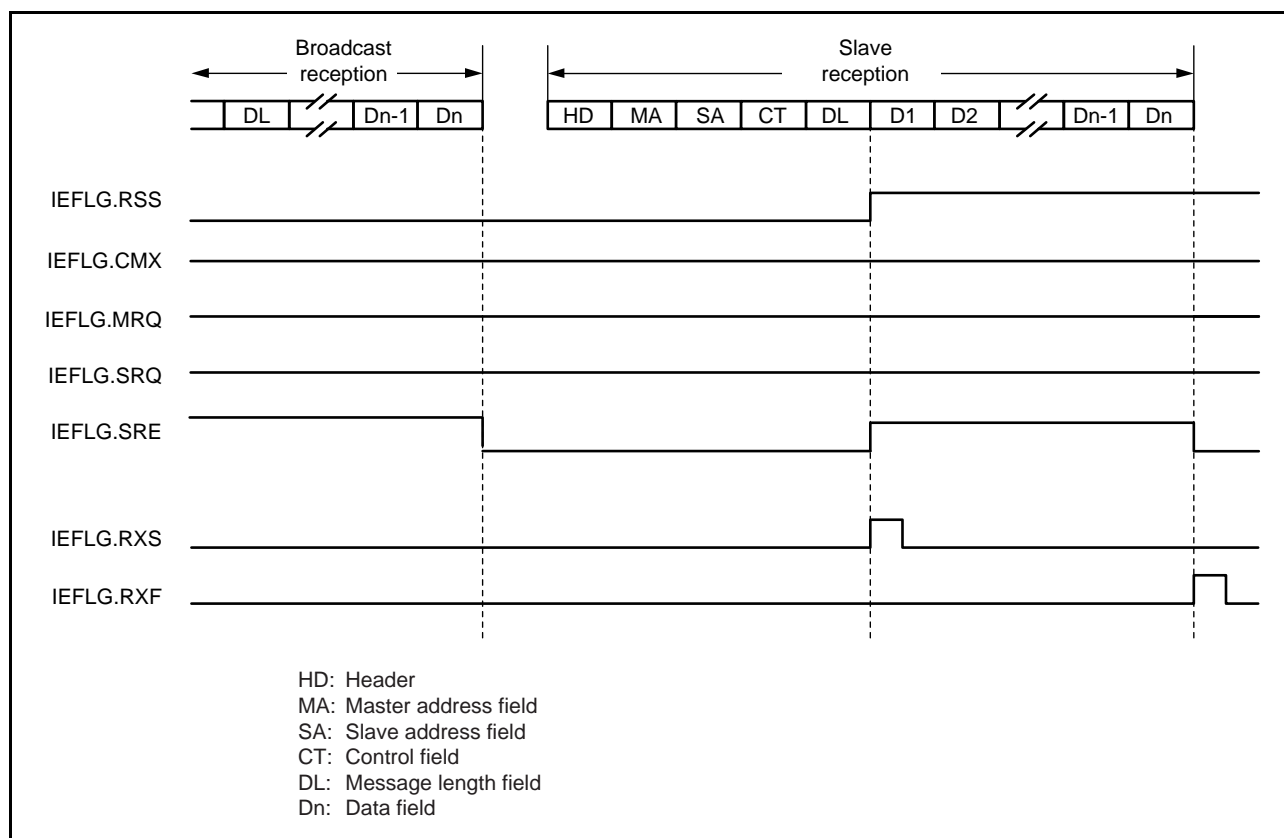


Figure 36.12 Slave Reception Timing

### 36.5.3 Master Reception

Figure 36.13 shows the timing for master reception.

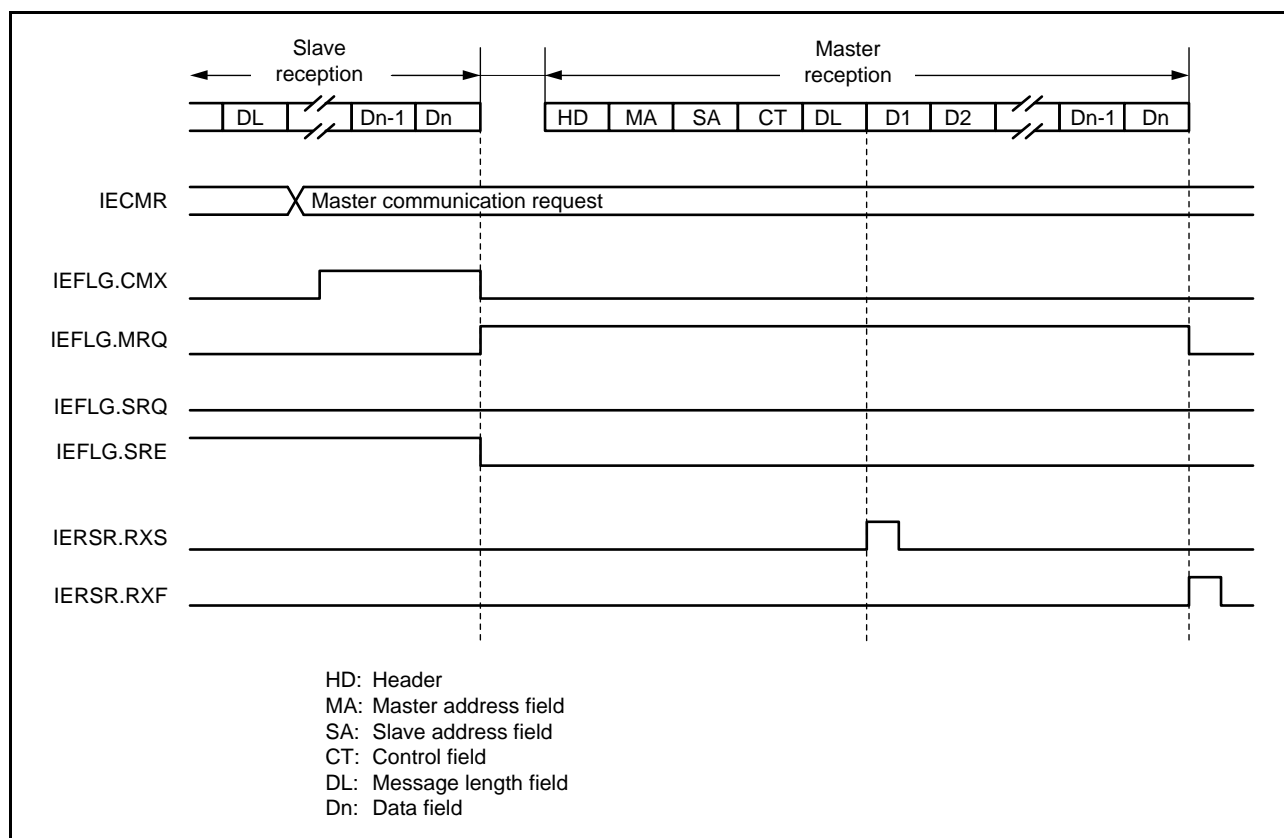


Figure 36.13 Master Reception Timing

### 36.5.4 Slave Transmission

Figure 36.14 shows the timing for slave transmission.

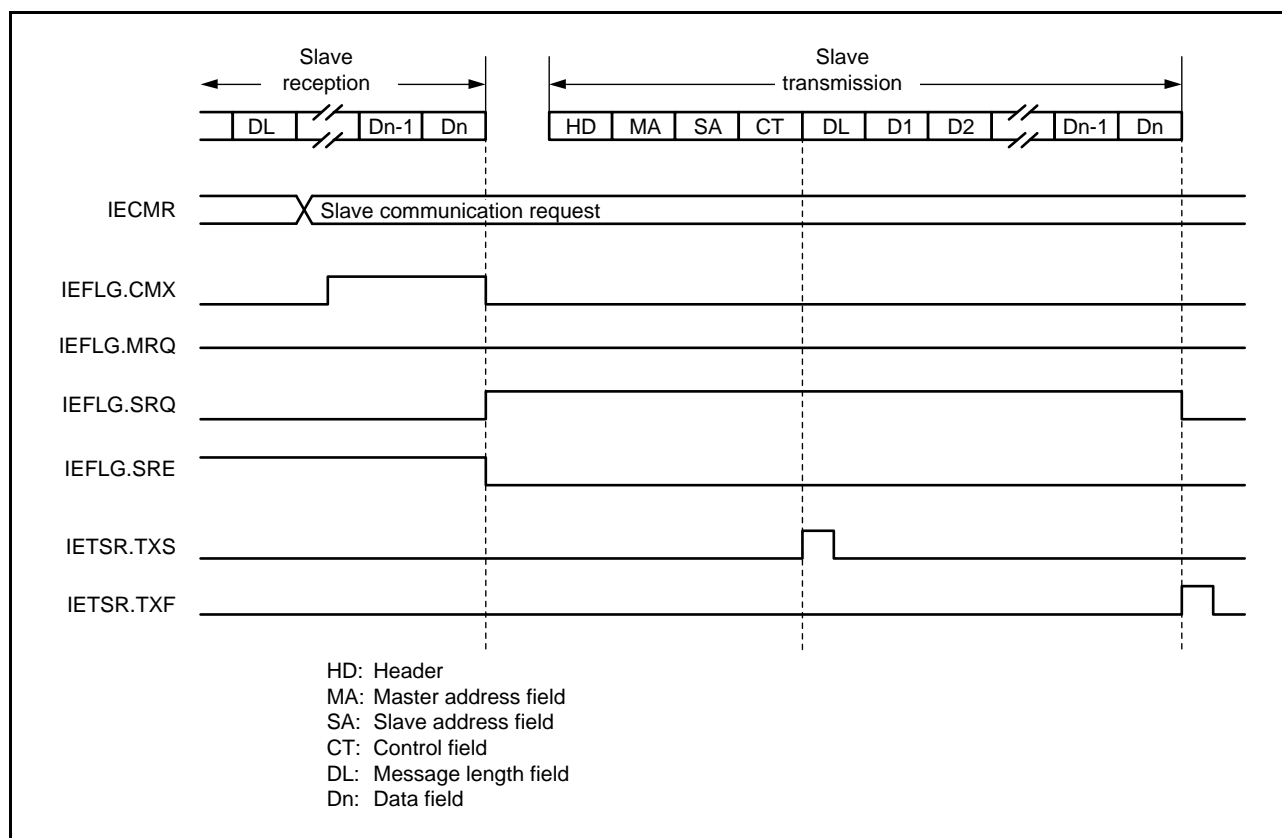


Figure 36.14 Slave Transmission Timing

### 36.6 Interrupt Sources

The IEB interrupt sources include the following: transmit start (TXS), transmit normal completion (TXF), arbitration loss (TXEAL), transmit timing error (TXETTME), transmit-frame maximum transfer byte overflow (TXERO), acknowledge bit (TXEACK), receive busy (RXBSY), receive start (RXS), receive normal completion (RXF), broadcast receive error (RXEDE), receive overrun flag (RXEOVE), receive timing error (RXERTME), receive-frame maximum transfer byte overflow (RXEDLE), and parity error (RXEPE).

Each source has a corresponding bit in the IEBus transmit interrupt enable register (IEIET) or the IEBus receive interrupt enable register (IEIER), which enables or disables the interrupt. Each source also has a corresponding status flag in the IEBus transmit status register (IETSR) or IEBus receive status register (IERSR). Reading the status flags allows determination of the interrupt sources.

Figure 36.15 shows the relations between the IEB interrupt sources.

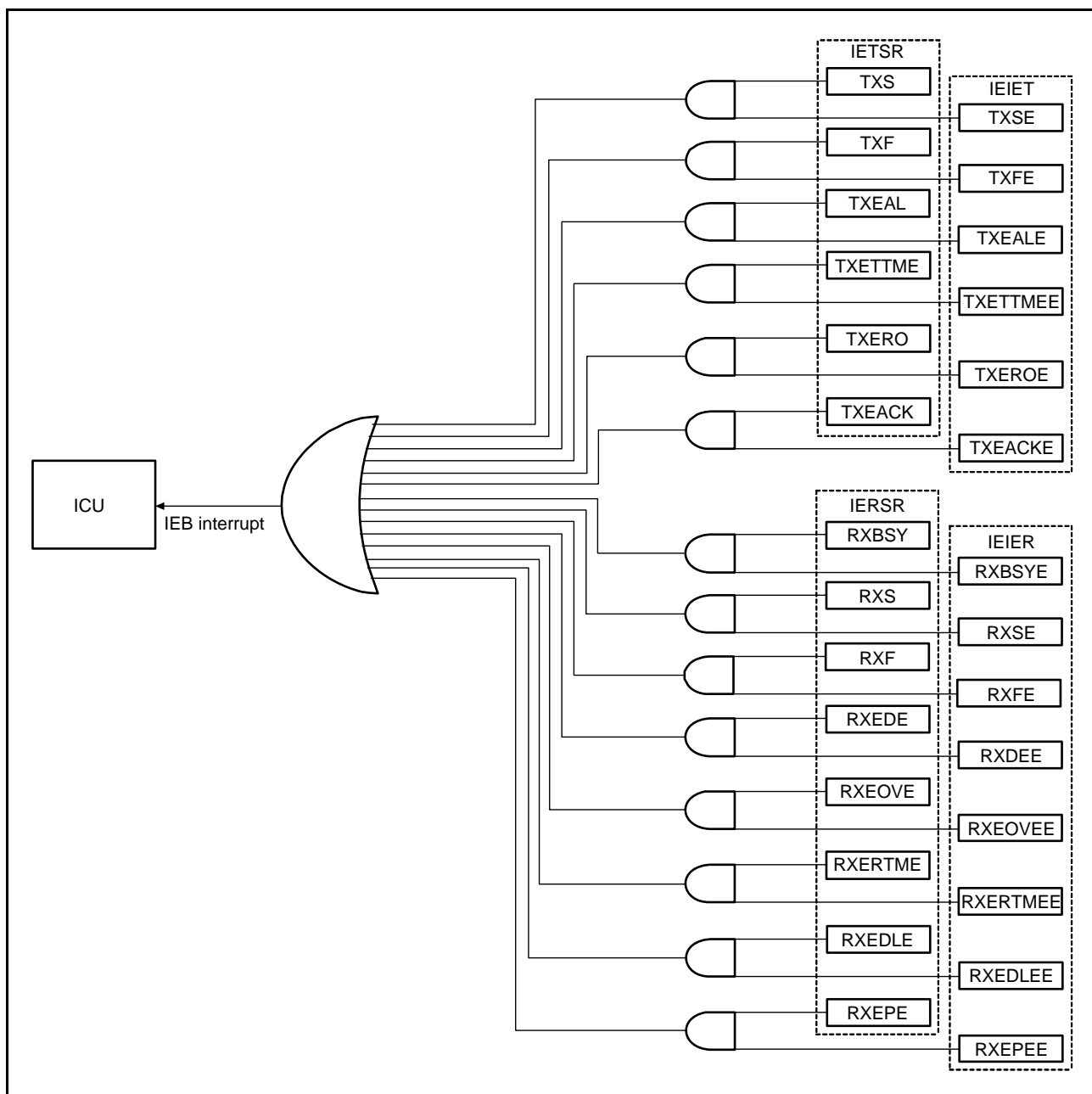


Figure 36.15 Relations between IEB Interrupt Sources

### 36.7 Usage Notes

#### 36.7.1 Notes on Operation when Transfer Is Not Completed within Maximum Number of Transfer Bytes

##### (1) Data Transmission

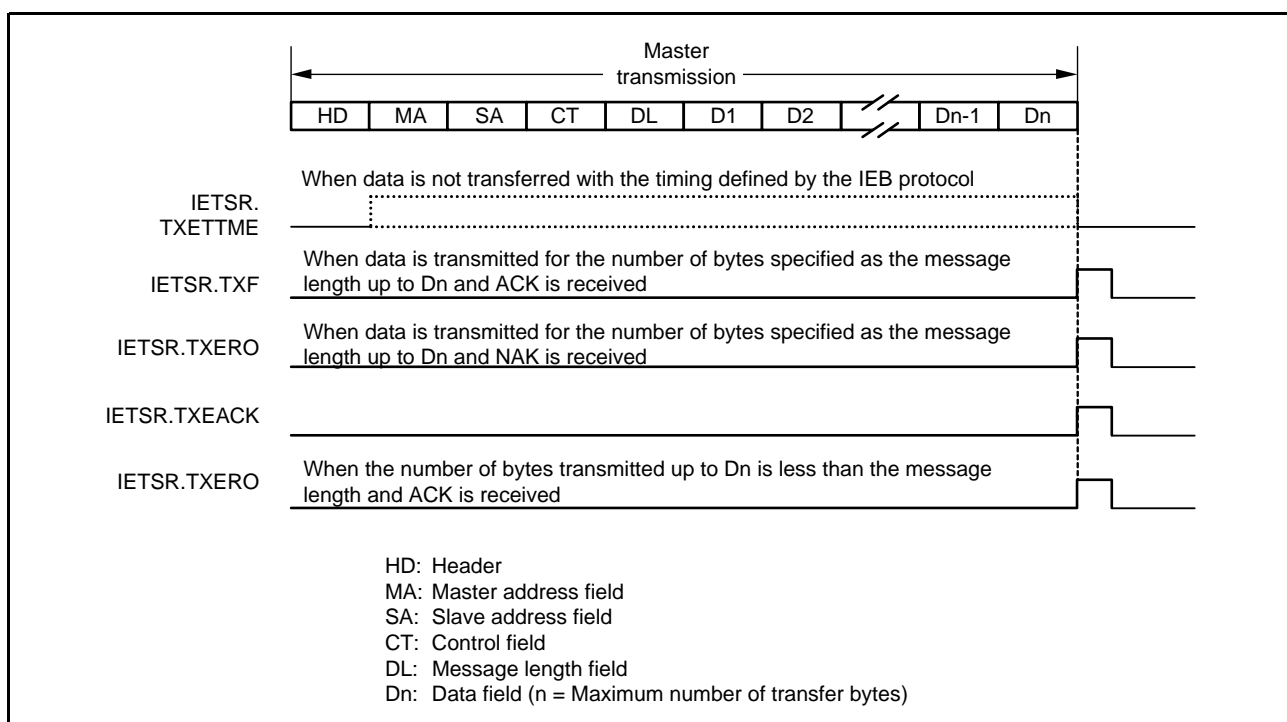
When data has been transmitted up to the maximum number of bytes defined by the communications mode but transmission has not been completed because a NAK has been received, from the receive unit during data transmission the IEB sets the error flag in IESTR and enters a wait state.

When a timing error has occurred during transmission up to the number of bytes specified as the message length, the IEB sets the IETSR.TXETTME flag and enters a wait state.

When data has been transferred for the number of bytes specified as the message length within the maximum number of bytes defined by the communications mode, if an ACK has been received, the IEB does not set the IETSR.TXERO or IETSR.TXEACK flag but sets the IETSR.TXF flag. If a NAK has been received, the IEB sets the IETSR.TXERO and IETSR.TXEACK flags.

When data has been transferred up to the maximum number of bytes defined by the communications mode but the transferred bytes are less than the message length, if an ACK has been received, the IEB sets the IETSR.TXERO flag. If a NAK has been received, the IEB sets the IETSR.TXERO and IETSR.TXEACK flags.

Figure 36.16 shows the timing of operations when transmission has not been completed within the maximum number of bytes.



**Figure 36.16** Timing of Operations when Transmission is Not Completed within Maximum Number of Transfer Bytes

## (2) Data Reception

When data reception has not been completed within the maximum number of bytes defined by the communications mode because of a parity error or overrun error causing the retransfer of data, the IEB sets the error flag in IERSR and enters a wait state.

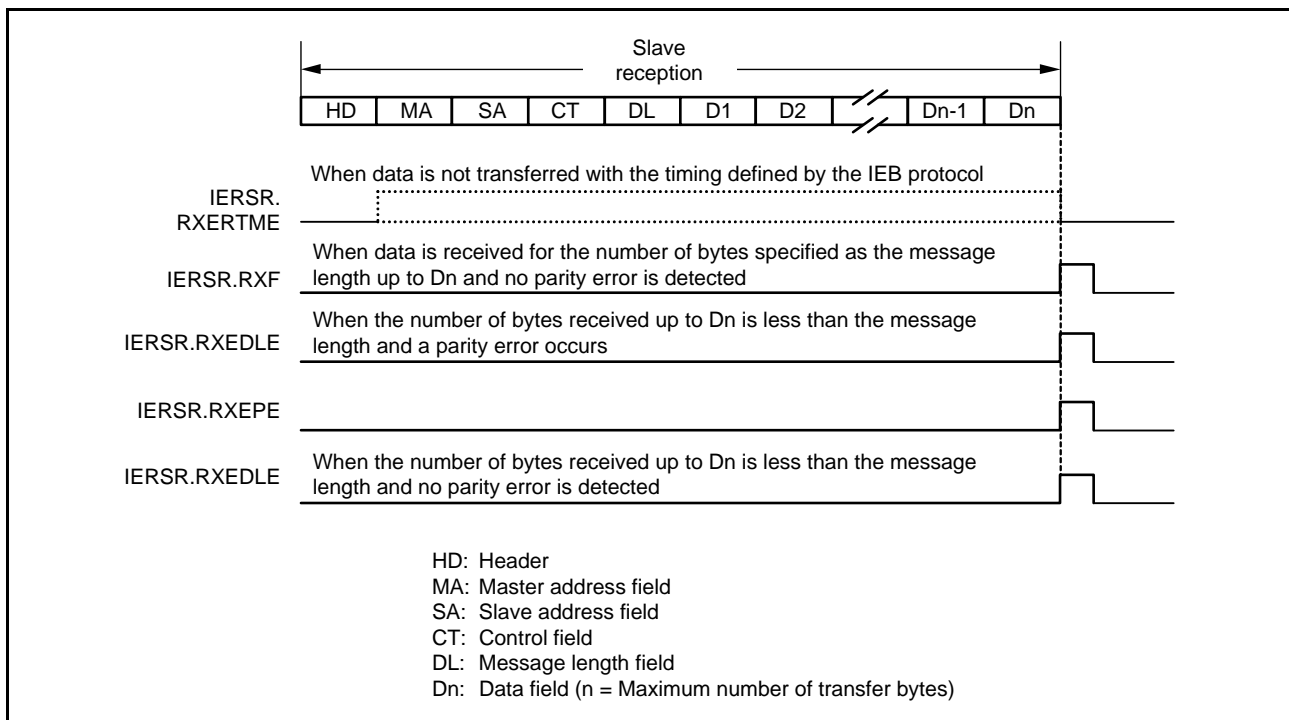
When a timing error has occurred during reception up to the number of bytes specified as the message length, the IEB sets the IERSR.RXERTME flag and enters a wait state.

When data has been received for the number of bytes specified as the message length within the maximum number of bytes defined by the communications mode, if no parity error has been detected, the IEB does not set the IERSR.RXEDLE or IERSR.RXEPE flag but sets the IERSR.RXF flag. If a parity error has occurred, the IEB sets the IERSR.RXEDLE and IERSR.RXEPE flags.

When data has been received up to the maximum number of bytes defined by the communications mode but the transferred bytes are less than the message length, if no parity error has been detected, the IEB sets the IERSR.RXEDLE flag. If a parity error has occurred, the IEB sets the IERSR.RXEDLE and IERSR.RXEPE flags.

When the IERSR.RXEDLE flag is set, discard the reception data because data has not been normally written to the receive buffer.

Figure 36.17 shows the timing of operations when the maximum number of transfer bytes is reached but reception has not been completed.



**Figure 36.17** Timing of Operations when Reception is Not Completed within Maximum Number of Transfer Bytes

### 36.7.2 Notes on Operation when Message Length Is Greater than Maximum Number of Transfer Bytes

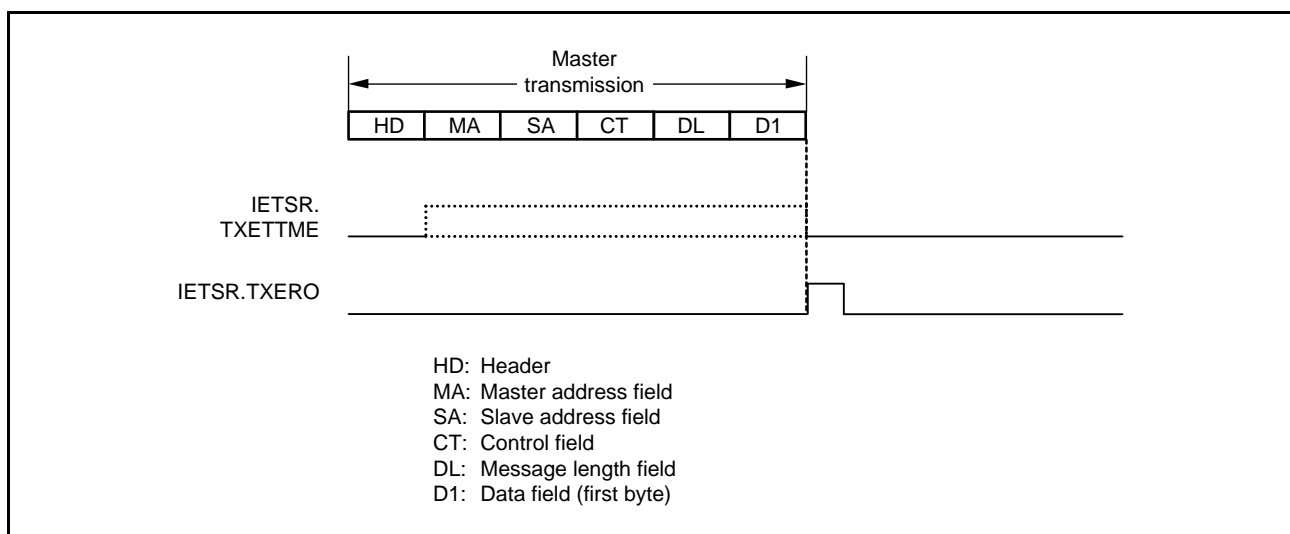
In the RX630 Group, the value specified in the message length bits should not be greater than the maximum number of transfer bytes in one frame.

The following describes operations when a greater value is specified by mistake.

#### (1) Data Transmission

When the value specified in the IEBus transmit message length register (IETBFL) is greater than the maximum number of transfer bytes in data transmission, the IEB terminates transfer after transmission of the first byte data and sets the transmit-frame maximum transfer byte overflow (TXERO) flag in the IEBus transmit status register (IETSR).

Figure 36.18 shows the timing of operations when the transmit message length is greater than the maximum number of transfer bytes.



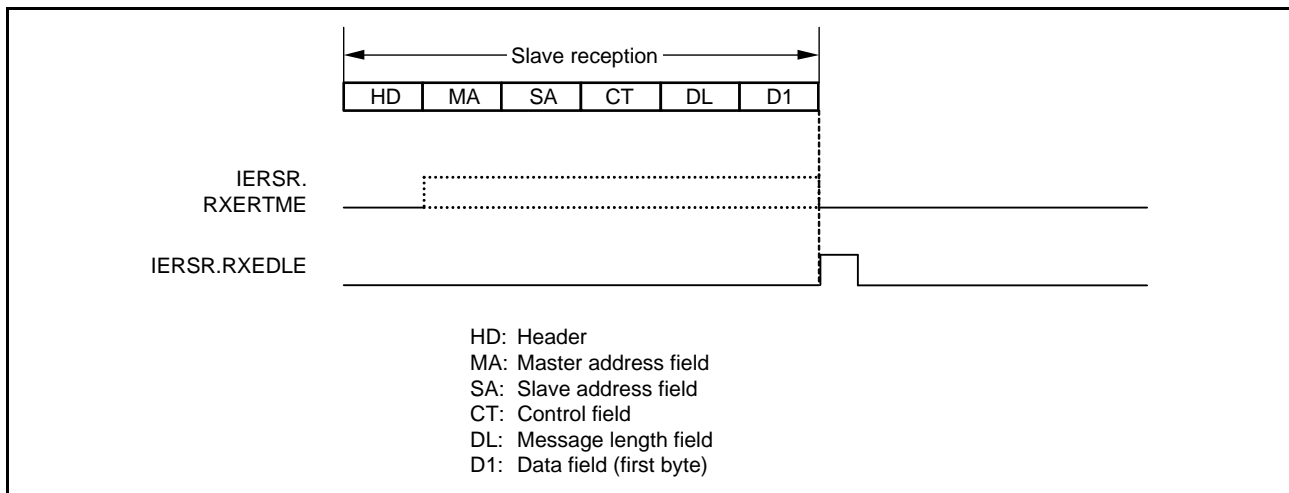
**Figure 36.18** Timing of Operations when Transmit Message Length is Greater than Maximum Number of Transfer Bytes

## (2) Data Reception

When the value specified in the message length field to be received is greater than the maximum number of transfer bytes in data reception, the IEB terminates transfer after reception of the first byte data and sets the receive-frame maximum transfer byte overflow (RXEDLE) flag in the IEBus receive status register (IERSR).

When the IERSR.RXEDLE flag is set, discard the reception data because data has not been normally written to the receive buffer.

Figure 36.19 shows the timing of operations when the value in the receive message length field is greater than the maximum number of transfer bytes.



**Figure 36.19** Timing of Operations when Value in Receive Message Length Field is Greater than Maximum Number of Transfer Bytes

### 36.7.3 Setting for the Module-Stop State

Module stop control register C (MSTPCRC) can be used to enable or disable operation of the IEB. The IEB is stopped after a reset. The registers become accessible on release from the module-stop state. For details, refer to section 11, Low Power Consumption.



## 37. CRC Calculator (CRC)

The CRC (Cyclic Redundancy Check) calculator generates CRC codes of data blocks.

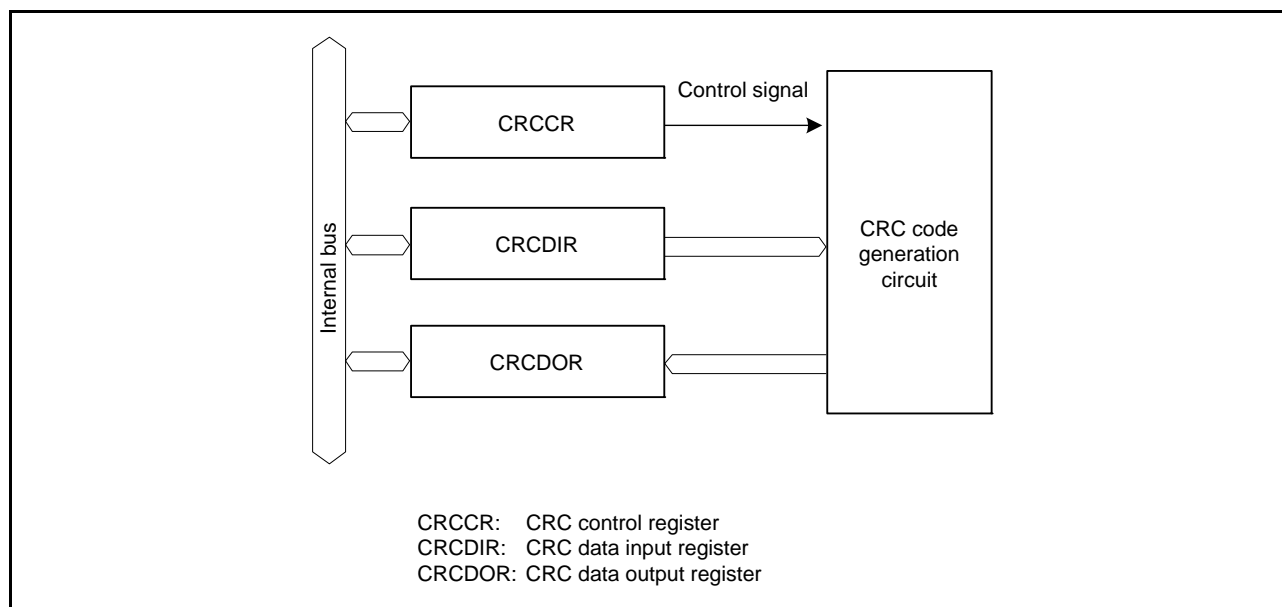
### 37.1 Overview

Table 37.1 lists the specifications of the CRC calculator, and Figure 37.1 shows a block diagram of the CRC calculator.

**Table 37.1 Specifications of CRC**

Item	Description
Data for CRC calculation*1	CRC code generated for any desired data in 8n-bit units (where n is a whole number)
Data block size	8 bits
CRC processor unit	Operation executed on eight bits in parallel
CRC generating polynomial	One of three generating polynomials selectable <ul style="list-style-type: none"> <li>• 8-bit CRC <math>X^8 + X^2 + X + 1</math></li> <li>• 16-bit CRC <math>X^{16} + X^{15} + X^2 + 1</math> <math>X^{16} + X^{12} + X^5 + 1</math></li> </ul>
CRC calculation switching	CRC code generation for LSB-first or MSB-first communication selectable
Low-power consumption function	Module-stop state can be set

Note 1. The circuit does not have functionality to divide data for calculation into a data-block size. Write data in 8-bit units.

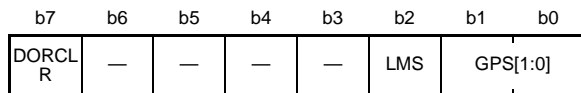


**Figure 37.1 Block Diagram of CRC Calculator**

## 37.2 Register Descriptions

### 37.2.1 CRC Control Register (CRCCR)

Address(es): 0008 8280h



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: $X^8 + X^2 + X + 1$ 1 0: $X^{16} + X^{15} + X^2 + 1$ 1 1: $X^{16} + X^{12} + X^5 + 1$	R/W
b2	LMS	CRC Calculation Switching	0: Performs CRC operation for LSB-first communication. The lower-order byte (bits 7 to 0) is the first to be transmitted when the contents of the CRCDOR (CRC code) are divided into bytes. 1: Performs CRC operation for MSB-first communication. The higher-order byte (bits 15 to 8) is first to be transmitted when the contents of the CRCDOR (CRC code) are divided into bytes.	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: Clear the CRCDOR register*1 This bit is read as 0.	W

Note 1. Only 1 can be written.

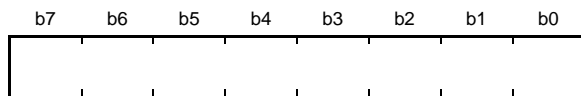
#### DORCLR Bit (CRCDOR Register Clear)

Write 1 to this bit so that the CRCDOR register is cleared to 0000h.

This bit is read as 0. Only 1 can be written.

### 37.2.2 CRC Data Input Register (CRCDIR)

Address(es): 0008 8281h

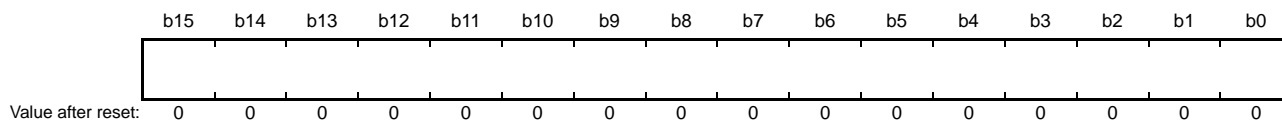


Value after reset: 0 0 0 0 0 0 0 0

CRCDIR is an 8-bit readable/writable register, to which the bytes to be CRC-operated are written.

### 37.2.3 CRC Data Output Register (CRCDOR)

Address(es): 0008 8282h



CRCDOR is a 16-bit readable/writable register that contains the result of CRC calculation.

In general, the value will be 0 if there is no CRC error when the calculated CRC code matches the CRC code that continues on, for verification, from the transferred data.

When an 8-bit CRC ( $X^8 + X^2 + X + 1$  polynomial) is in use, the valid CRC code is obtained in the lower-order byte (b7 to b0). The higher-order byte (b15 to b8) is not updated.

### 37.3 Operation

The CRC calculator generates CRC codes for use in LSB-first or MSB-first transfer.

The following figures show examples in which the CRCCR.GPS[1:0] bits are set to 11b so the CRC code is calculated by using a 16-bit CRC (with the polynomial  $X^{16} + X^{12} + X^5 + 1$ ), and the CRC code is calculated for the value "F0h".

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 CRCDOR.

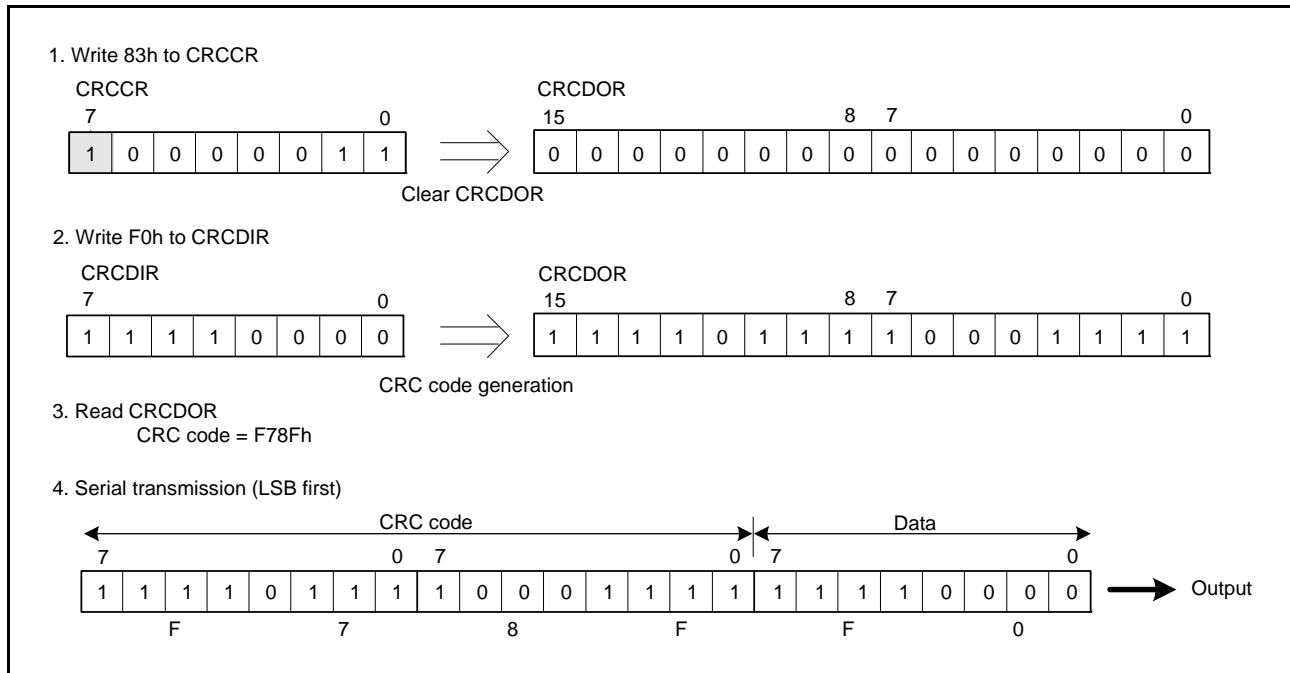


Figure 37.2 LSB-First Data Transmission

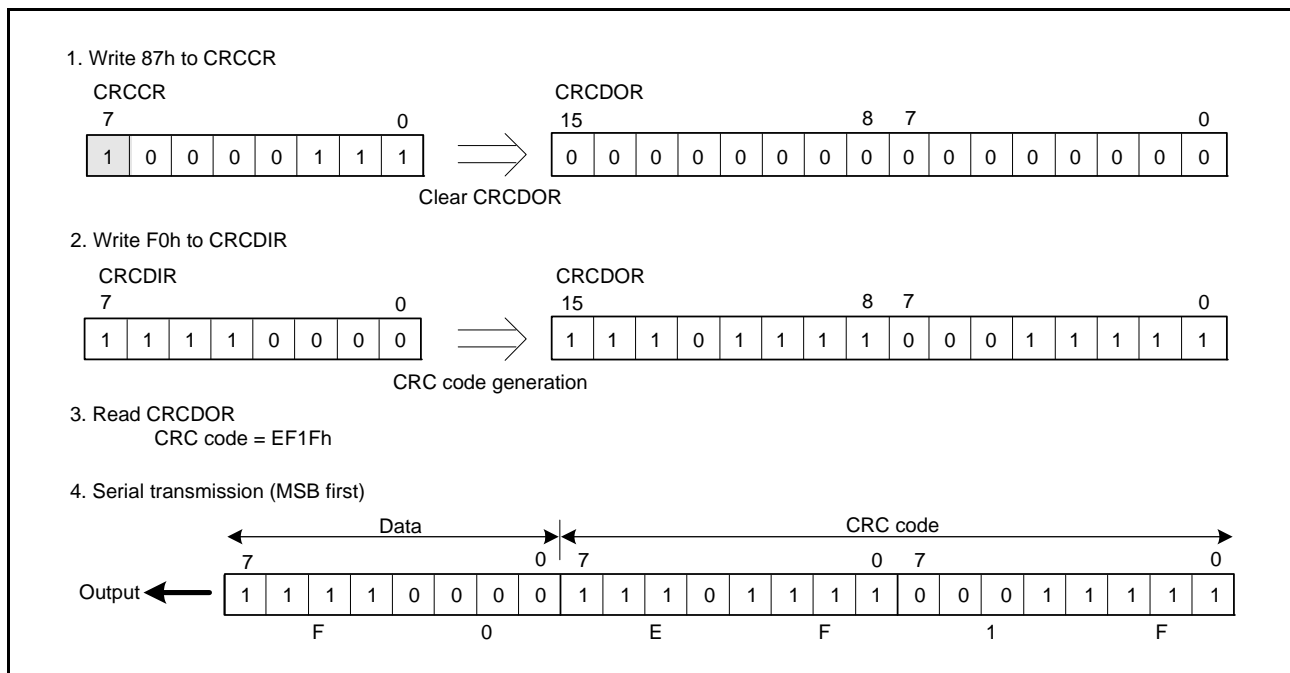


Figure 37.3 MSB-First Data Transmission

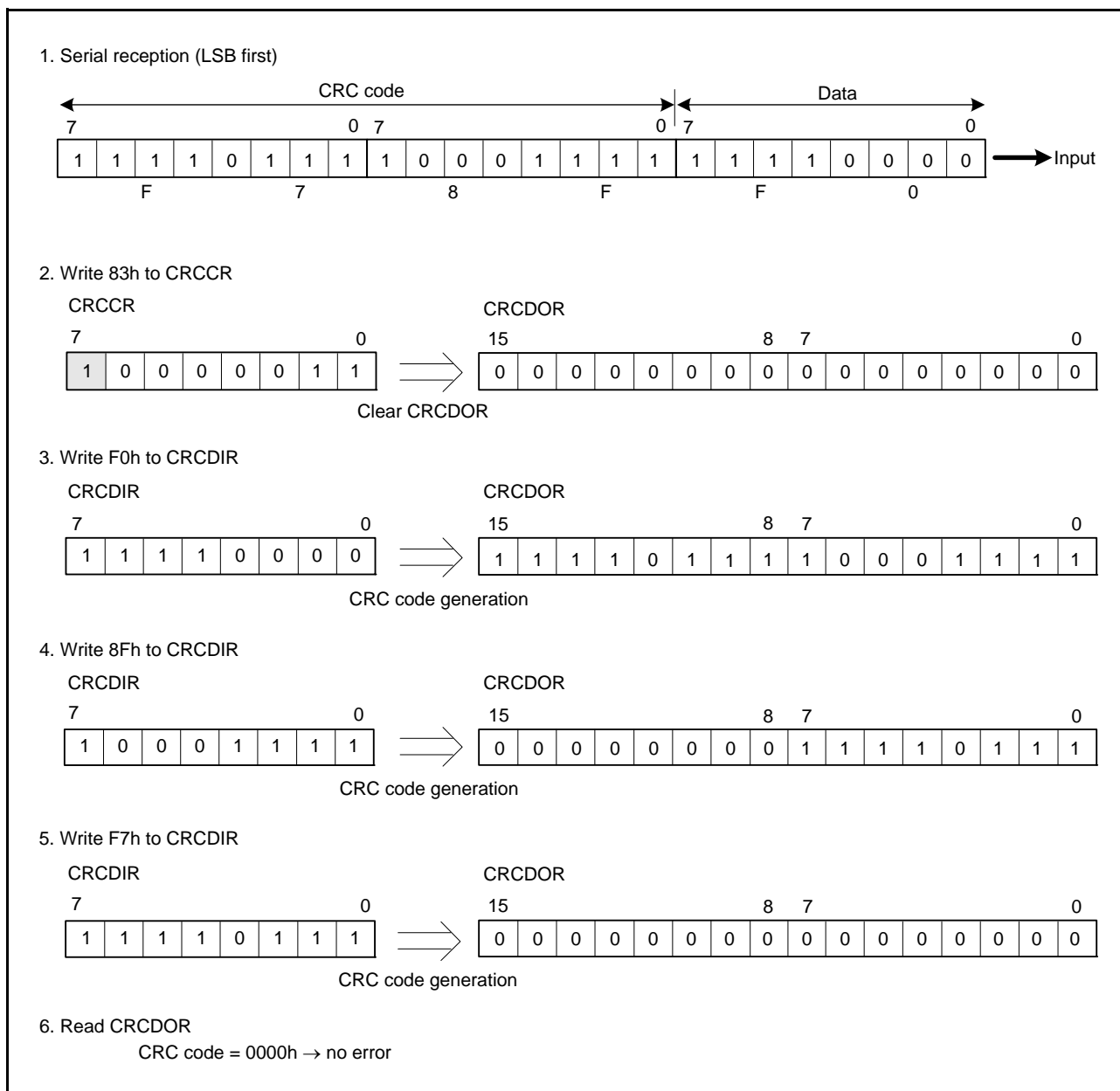


Figure 37.4 LSB-First Data Reception



### 37.4 Usage Notes

#### 37.4.1 Module-Stop Function Setting

Operation of the CRC calculator can be disabled or enabled using the module stop control register B (MSTPCRB). The initial setting is for operation of the CRC calculator to be stopped. Register access is enabled by clearing the module-stop state. For details, see section 11, Low Power Consumption.

#### 37.5 Note on Transmission

Note that the sequence of transmission for the CRC code differs according to whether transmission is LSB-first or MSB-first.

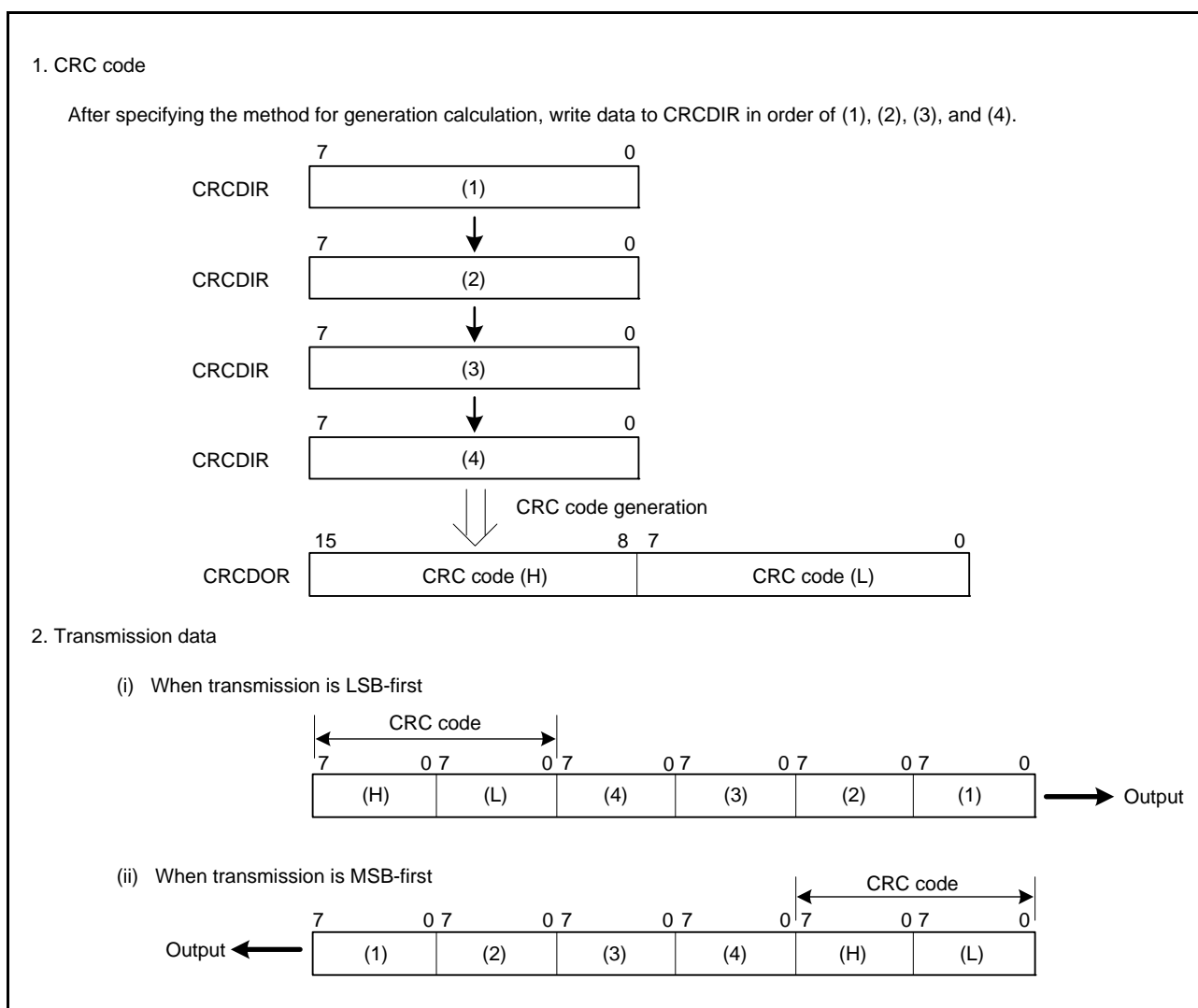


Figure 37.6 LSB-First and MSB-First Data Transmission

## 38. 12-Bit A/D Converter (S12ADa)

### 38.1 Overview

The RX630 Group includes a 12-bit successive approximation A/D converter. Up to 21 channel analog inputs, temperature sensor outputs, or A/D internal reference voltages can be selected.

The A/D converter has two operating modes: single scan mode in which the analog input of up to 21 channels is converted for only once in ascending channel order and continuous scan mode in which the analog inputs of up to 21 arbitrarily selected channels are continuously converted in ascending channel order.

A/D conversion of the temperature sensor output or the A/D internal reference voltage is accomplished independently. Table 38.1 lists the specifications of the 12-bit A/D converter and Table 38.2 indicates the functions of the 12-bit A/D converter. Figure 38.1 shows a block diagram of the 12-bit A/D converter.

**Table 38.1 Specifications of 12-Bit A/D Converter**

Item	Specifications
Number of units	One unit
Input channels	Up to 21 channels
Extended analog inputs	Temperature sensor output, A/D internal reference voltage*3
A/D conversion method	Successive approximation method
Resolution	12 bits
Conversion time	1.0 $\mu$ s per 1 channel (when operating with A/D conversion clock ADCLK = 50 MHz)
A/D conversion clock (ADCLK)	4 types: PCLK, PCLK/2, PCLK/4, PCLK/8
Data registers	<ul style="list-style-type: none"> <li>For analog input: 21 data registers</li> <li>For temperature sensor: One data register</li> <li>For internal reference voltage: One data register</li> <li>The A/D conversion result is held in a 12-bit A/D data register.</li> <li>In A/D-converted value addition mode, A/D conversion results are stored in a 14-bit A/D data register.</li> </ul>
Operating modes	<ul style="list-style-type: none"> <li>Single-cycle scan mode: A/D conversion is to be performed for only once on the analog inputs of up to 21 arbitrarily selected channels. A/D conversion is to be performed only once on the temperature sensor output. A/D conversion is to be performed only once on the internal reference voltage.</li> <li>Continuous scan mode: A/D conversion is to be performed sequentially on the analog inputs of up to 21 arbitrarily selected channels.*1</li> </ul>
Conditions of A/D conversion start	<ul style="list-style-type: none"> <li>Software trigger</li> <li>Synchronous trigger Trigger by MTU, TPU, or TMR</li> <li>Asynchronous trigger A/D conversion can be started by the ADTRG0# pin.</li> </ul>
Functions	<ul style="list-style-type: none"> <li>Sample-and-hold function</li> <li>Number of sampling states is adjustable.</li> <li>A/D-converted value addition mode</li> </ul>
Interrupt sources	<ul style="list-style-type: none"> <li>A scan end interrupt (S12ADI0) request can be generated on completion of A/D conversion.</li> <li>An S12ADI0 interrupt can activate the DMAC or DTC.</li> </ul>
Low-power consumption function	Module-stop state can be specified.*2

Note 1. Do not use continuous scan mode when temperature sensor output or A/D internal reference voltage is selected.

Note 2. When the module-stop state is canceled, A/D conversion can be started after 10 ms has elapsed.

Note 3. Refer to section 45, Electrical Characteristics for the A/D internal reference voltage.



**Table 38.2 Function Overview of the 12-Bit A/D Converter**

Item	Function/Internal Trigger Source			
Analog input channel	AN000 to AN020			
A/D conversion start conditions	Software	Software trigger	Enabled	
	Asynchronous trigger	ADTRG0#	Enabled	
	Synchronous trigger (MTU, TMR, TPU)*3	TRG0AN_0	MTU0.TGRA and MTU0.TCNT	Input-capture/compare-match
			TRG0BN_0	
		TRGAN_0	MTU0.TGRA and MTU0.TCNT	Input-capture/compare-match
			MTU1.TGRA and MTU1.TCNT	
			MTU2.TGRA and MTU2.TCNT	
			MTU3.TGRA and MTU3.TCNT	
			MTU4.TGRA and MTU4.TCNT	
			MTU4.TCNT	
		TRGAN_1	TPU0.TGRA	Input-capture/compare-match
			TPU1.TGRA	
			TPU2.TGRA	
			TPU3.TGRA	
TPU4.TGRA				
TRG0EN_0	MTU0.TGRE and MTU0.TCNT	Compare-match		
TRG0FN_0	MTU0.TGRF and MTU0.TCNT			
TRG4ABN_0	MTU4.TADCORA and MTU4.TCNT or MTU4.TADCORB and MTU4.TCNT	Compare-match using delayed A/D conversion start request function		
TRG4ABN_1	TPU0.TGRA	Input-capture/compare-match		
TMTRG0AN_0	TMR0.TCORA and TMR0.TCNT	Compare-match		
TMTRG0AN_1	TMR2.TCORA and TMR2.TCNT			
Interrupt	S12ADI0 interrupt			
Module-stop function setting*1,*2	MSTPCRA.MSTPA17 bit			

Note 1. When the module-stop state is canceled, A/D conversion can be started after 10 ms has elapsed.

Note 2. For details, see section 11, Low Power Consumption.

Note 3. The suffix “\_0” or “\_1” appended to the name of a synchronous trigger signal indicates the number of the unit. For synchronous trigger output settings, see section 22.4.3, A/D Converter Activation, and section 26.6.2, A/D Converter Activation.

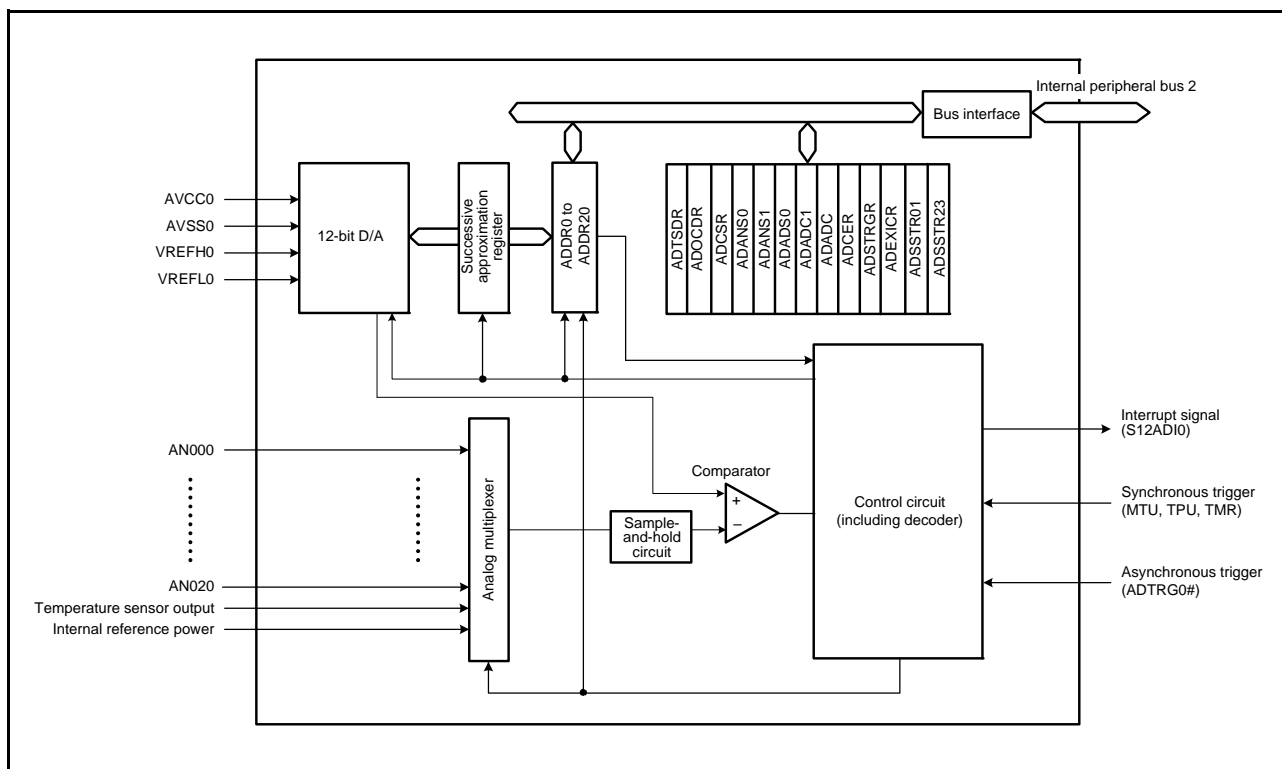


Figure 38.1 Block Diagram of 12-Bit A/D Converter

Table 38.3 indicates the input pins of the 12-bit A/D converter.

Table 38.3 Input Pins of 12-Bit A/D Converter

Pin Name	Input	Function
AVCC0	Input	Analog block power supply pin for 12-bit A/D converter
AVSS0	Input	Ground pin for 12-bit A/D converter
VREFH0	Input	Reference power supply pin for 12-bit A/D converter
VREFL0	Input	Reference power supply ground pin for 12-bit A/D converter
AN000 to AN020	Input	Analog input pins
ADTRG0#	Input	External trigger input pin for starting A/D conversion

## 38.2 Register Descriptions

### 38.2.1 A/D Control Register (ADCSR)

Address(es): 0008 9000h

b7	b6	b5	b4	b3	b2	b1	b0
ADST	ADCS	—	ADIE	CKS[1:0]	TRGE	EXTRG	
Value after reset:	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	EXTRG	Trigger Select* <sup>1</sup>	0: A/D conversion is started by the synchronous trigger (MTU, TPU, or TMR). 1: A/D conversion is started by the asynchronous trigger (ADTRG0#).	R/W
b1	TRGE	Trigger Start Enable	0: Disables A/D conversion to be started by the synchronous or asynchronous trigger 1: Enables A/D conversion to be started by the synchronous or asynchronous trigger	R/W
b3, b2	CKS[1:0]	A/D Conversion Clock Select	b3 b2 0 0: PCLK/8 0 1: PCLK/4 1 0: PCLK/2 1 1: PCLK	R/W
b4	ADIE	Scan End Interrupt Enable	0: Disables S12ADI0 interrupt generation upon scan completion. 1: Enables S12ADI0 interrupt generation upon scan completion.	R/W
b5	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b6	ADCS	Scan Mode Select	0: Single scan mode 1: Continuous scan mode	R/W
b7	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)  
If 1 is written to both the TRGE and EXTRG bits in ADCSR when a high-level signal is input to the external pin (ADTRG0#), and then if the ADTRG0# signal is driven low, the falling edge of ADTRG0# is detected and the scan conversion process is started. In this case, the pulse width of the low-level input must be at least 1.5 PCLK cycles.

#### CKS[1:0] Bits (A/D Conversion Clock Select)

The CKS[1:0] bits select the clock to be used in A/D conversion.

The CKS[1:0] bits should be set with the ADST bit being 0. (In addition, the CKS[1:0] bits should not be set simultaneously with the setting of the ADST bit to 1.)

#### ADIE Bit (Scan End Interrupt Enable)

The ADIE bit enables or disables the scan end interrupt (S12ADI0).

When A/D conversion of all the selected channels is completed while the ADIE bit is set to 1, the scan end interrupt is generated. When A/D conversion is completed even if the temperature sensor output or the A/D internal reference voltage is selected and the ADIE bit is set to 1, the scan end interrupt is generated.

### ADCS Bit (Scan Mode Select)

The ADCS bit selects the scan mode.

In single scan mode, A/D conversion is performed for the analog inputs of a maximum of 21 channels selected with the ADANS0 and ADANS1 registers in the ascending order of the channel number, and when one cycle of A/D conversion is completed for all the selected channels, scan conversion is stopped.

In continuous scan mode, while the ADST bit in ADCSR is 1, A/D conversion is performed for the analog inputs of a maximum of 21 channels selected with the ADANS0 and ADANS1 registers 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 beginning at the first channel. A/D conversion is repeated until the ADST bit in ADCSR is set to 0.

When the temperature sensor output or the A/D internal reference voltage is selected, single scan mode should be selected and all the channels selected by the ADANS0 and ADANS1 registers should be deselected, after which A/D conversion is to be started. The A/D conversion stops after completion of A/D conversion of the temperature sensor output or the A/D internal reference voltage selected.

The ADCS bit should be set while the ADST bit is 0 (it should not be set simultaneously when 1 is written to the ADST bit.)

Table 38.4 shows the scope of selectability of items for A/D conversion by scan mode.

**Table 38.4 Scan Mode and Scope of Selectability for A/D Conversion**

Scan Mode Setting	Scope of A/D Conversion		
	Analog Input	Temperature Sensor Output	Internal Reference Voltage
Single scan mode	○	○	○
Continuous scan mode	○	×	×

○: Selectable, ×: Not selectable

Note 1. Ensure that the analog input is not selected at the same time as the temperature sensor output or A/D internal reference voltage.

Note 2. Ensure that neither the temperature sensor output nor the A/D internal reference voltage is selected at the same time as the analog input.

### ADST Bit (A/D Conversion Start)

This 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]

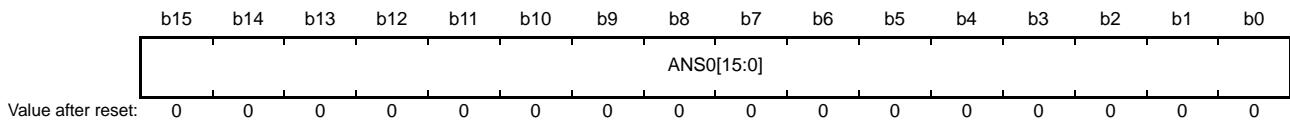
- When 1 is written by software
- When the ADCSR.EXTRG bit is set to 0, the ADCSR.TRGE bit is set to 1, and the synchronous trigger (MTU, TPU, or TMR) selected by the ADSTRGR.ADSTRS[3:0] bits is detected
- When the ADCSR.EXTRG and ADCSR.TRGE bits are set to 1, the ADSTRGR.ADSTRS[3:0] bits are set to 0000b, and the asynchronous trigger is detected

[Clearing conditions]

- When 0 is written by software
- When the A/D conversion of all channels selected in single scan mode is completed
- When the A/D conversion of the temperature sensor output or the A/D internal reference voltage selected in single scan mode is completed

### 38.2.2 A/D Channel Select Register 0 (ADANS0)

Address(es): 0008 9004h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	ANS0[15:0]	A/D Conversion Channel Select	0: AN000 to AN015 are not subjected to conversion. 1: AN000 to AN015 are subjected to scan conversion.	R/W

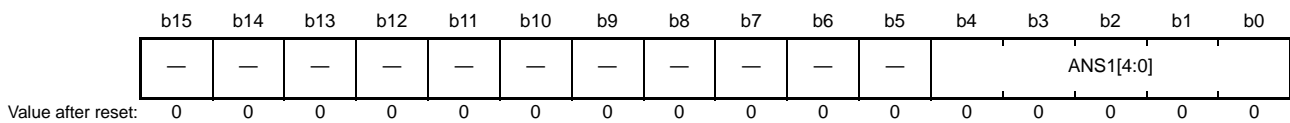
#### ANS0[15:0] Bits (A/D Conversion Channel Select)

These bits select analog inputs of the channels AN000 to AN015 that are subjected to A/D conversion. The channel that is to be selected and the number of channels can be arbitrarily set. ANS0[0] corresponds to AN000 and ANS0[15] corresponds to AN015. When A/D conversion of analog inputs of the channels is to be performed, A/D conversion of the temperature sensor output and the A/D internal reference voltage should not be performed.

The ANS0[15:0] bits should be set while the ADST bit in ADCSR is set to 0.

### 38.2.3 A/D Channel Select Register 1 (ADANS1)

Address(es): 0008 9006h



Bit	Symbol	Bit Name	Description	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4 to b0	ANS1[4:0]	A/D Conversion Channel Select	0: AN016 to AN020 are not subjected to conversion. 1: AN016 to AN020 are subjected to scan conversion.	R/W

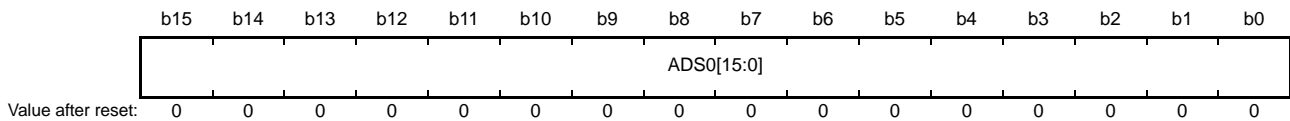
#### ANS1[4:0] Bits (A/D Conversion Channel Select)

These bits select analog inputs of the channels AN016 to AN020 that are subjected to A/D conversion. The channel that is to be selected and the number of channels can be arbitrarily set. ANS1[0] corresponds to AN016 and ANS1[4] corresponds to AN020. When A/D conversion of analog inputs of the channels is to be performed, A/D conversion of the temperature sensor output and the A/D internal reference voltage should not be performed.

The ANS1[4:0] bits should be set while the ADST bit in ADCSR is set to 0.

### 38.2.4 A/D-Converted Value Addition Mode Select Register 0 (ADADS0)

Address(es): 0008 9008h



Bit	Symbol	Bit Name	Description	R/W
b15 to b0	ADS0[15:0]	A/D-Converted Value Addition Channel Select	0: A/D-converted value addition mode for AN000 to AN015 is not selected. 1: A/D-converted value addition mode for AN000 to AN015 is selected.	R/W

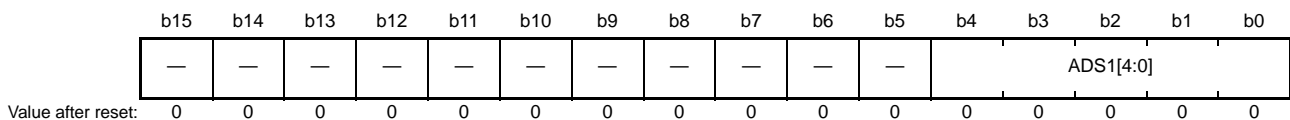
#### ADS0[15:0] Bits (A/D-Converted Value Addition Channel Select)

When the ADS0[n] bit of the number that is the same as that of A/D converted channel selected by ANS0[n] (n = 0 to 15) in ADANS is set to 1, these bits perform A/D conversion of analog input of the selected channels successively 2 to 4 times that is set with the ADC[1:0] bits in ADADC and returns the added (integrated) conversion results to the A/D data register. For the channel for which the A/D conversion is performed and addition mode is not selected, a normal one-time conversion is performed and the conversion result is returned to the A/D data register.

The ADS0[15:0] bits should be set while the ADST bit in ADCSR is set to 0.

### 38.2.5 A/D-Converted Value Addition Mode Select Register 1 (ADADS1)

Address(es): 0008 900Ah



Bit	Symbol	Bit Name	Description	R/W
b15 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4 to b0	ADS1[4:0]	A/D-Converted Value Addition Channel Select	0: A/D-converted value addition mode for AN016 to AN020 is not selected. 1: A/D-converted value addition mode for AN016 to AN020 is selected.	R/W

#### ADS1[4:0] Bits (A/D-Converted Value Addition Channel Select)

When the ADS1[n] bit of the number that is the same as that of A/D converted channel selected by ANS1[n] (n = 0 to 4) in ADANS is set to 1, these bits perform A/D conversion of analog input of the selected channels successively 2 to 4 times that is set with the ADC[1:0] bits in ADADC and returns the added (integrated) conversion results to the A/D data register. For the channel for which the A/D conversion is performed and addition mode is not selected, a normal one-time conversion is performed and the conversion result is returned to the A/D data register.

The ADS1[4:0] bits should be set while the ADST bit in ADCSR is set to 0.

Figure 38.2 shows a scanning operation sequence in which both the ADS0[2] and ADS0[6] bits are set to 1. In continuous scan mode (ADCSR.ADCS = 1), it is assumed that the addition count is set to 3 (ADADC.ADC[1:0] = 11b) and the channels AN000 to AN007 are selected (ADANS0.ANS0[15:0] = 00FFh). The conversion process begins with AN000. The AN002 conversion is performed successively 4 times, and the added (integrated) value is returned to the A/D data register 2. After that the AN003 conversion process is started. The AN006 conversion is performed successively 4 times and the added (integrated) value is returned to A/D data register 6. After conversion of AN007, the conversion operation is once again performed in the same sequence from AN000. For the channel for which the addition mode is not selected, the A/D data register format is determined by the ADRFMT bit in ADCER (flush-right or flush-left).

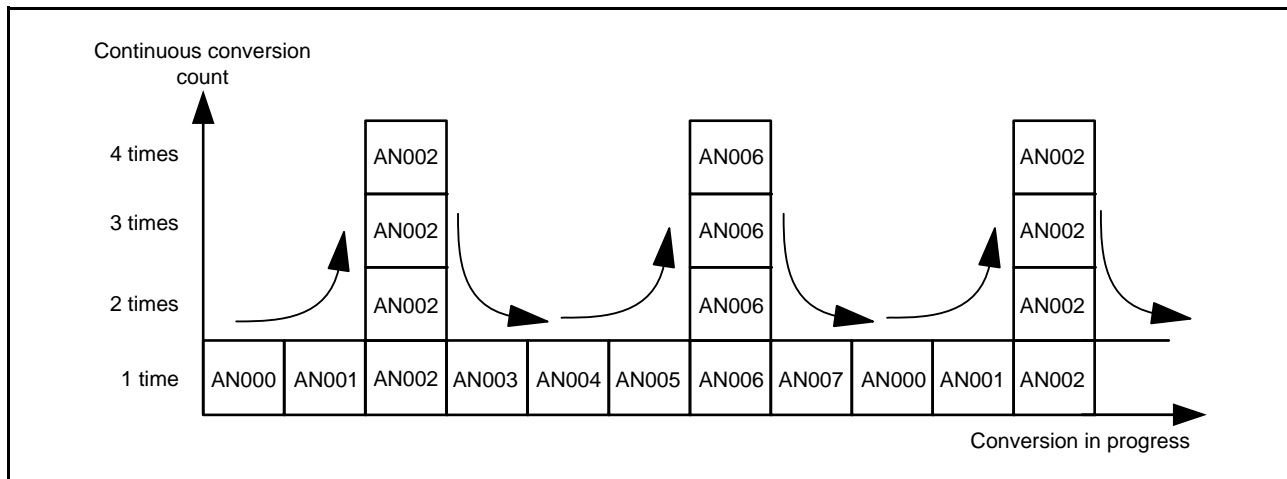
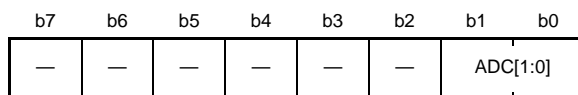


Figure 38.2 Scan Conversion Sequence with ADADC.ADC[1:0] = 11b, ADS0[2] = 1, and ADS0[6] = 1

### 38.2.6 A/D-Converted Value Addition Count Select Register (ADADC)

Address(es): 0008 900Ch



Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b1, b0	ADC[1:0]	Addition Count Select	b1 b0 0 0: 1-time conversion (no addition; same as normal conversion) 0 1: 2-time conversion (addition once) 1 0: 3-time conversion (addition twice) 1 1: 4-time conversion (addition three times)	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

#### ADC[1:0] Bits (Addition Count Select)

The ADC[1:0] bits set the addition count common for the channels for which A/D conversion is performed and the addition mode is selected, and for A/D conversion of the temperature sensor output and the A/D internal reference voltage. The ADC[1:0] bits should be set while the ADST bit in ADCSR is set to 0.

### 38.2.7 A/D Control Extended Register (ADCER)

Address(es): 0008 900Eh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	ADRFMT	—	—	—	—	—	—	—	—	—	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	Automatic Clearing Enable	0: Disables automatic clearing. 1: Enables automatic clearing	R/W
b14 to b6	—	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

#### ACE Bit (Automatic Clearing Enable)

The ACE bit enables or disables automatic clearing (all 0) of ADDR<sub>y</sub> or after the given register has been read by the CPU, DTC, or DMAC. This function also enables detection of failures to update ADDR<sub>y</sub>.

However, neither the A/D temperature sensor data register (ADTSDR) nor the A/D internal reference voltage data register (ADOCDR) is automatically cleared.

#### ADRFMT Bit (A/D Data Register Format Select)

The ADRFMT bit specifies flush-right or flush-left for the data to be stored in ADDR<sub>y</sub>, the A/D temperature sensor data register (ADTSDR), and the A/D internal reference voltage data register (ADOCDR).

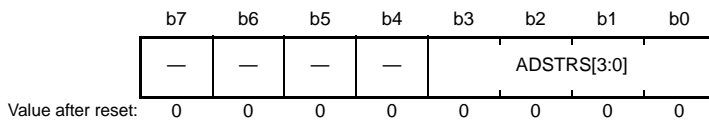
When the A/D converted value addition mode is selected, the format of each data register is fixed to flush-left, irrespective of the ADRFMT bit value.

For details on the format of the data registers, see section 38.2.12, A/D Data Register *y* (ADDR<sub>y</sub>) (*y* = 0 to 20), section 38.2.10, A/D Temperature Sensor Data Register (ADTSDR), and section 38.2.11, A/D Internal Reference Voltage Data Register (ADOCDR).



### 38.2.8 A/D Start Trigger Select Register (ADSTRGR)

Address(es): 0008 9010h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	ADSTRS[3:0]	A/D Conversion Start Trigger Select	These bits select the A/D conversion startup source from the on-chip peripheral modules.	R/W
b7 to b4	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

#### ADSTRS[3:0] Bits (A/D Conversion Start Trigger Select)

The ADSTRS[3:0] bits select A/D conversion start trigger. Table 38.5 lists a list of A/D conversion startup sources.

- When using the A/D conversion startup source of the synchronous trigger (MTU, TPU, or TMR), set the TRGE bit in ADCSR to 1 and set the EXTRG bit in ADCSR to 0.
- When using the asynchronous trigger (ADTRG0#), set the TRGE bit in ADCSR to 1 and set the EXTRG bit in ADCSR to 1.
- Software trigger (ADST bit in ADCSR) is always enabled regardless of the set values of the TRGE, EXTRG in ADCSR, and the ADSTRS[3:0] bits.

**Table 38.5 List of Sources to Start A/D Conversion**

Selectable Source	ADSTRS[3]	ADSTRS[2]	ADSTRS[1]	ADSTRS[0]	Remarks
Software trigger	—	—	—	—	
Asynchronous trigger	0	0	0	0	
TRG0AN_0	0	0	0	1	MTU0.TGRA and MTU0.TCNT
TRG0BN_0	0	0	1	0	MTU0.TGRB and MTU0.TCNT
TRGAN_0	0	0	1	1	MTUn.TGRA and MTUn.TCNT (n = 0 to 4)
TRGAN_1	0	1	0	0	TPUn.TGRA (n = 0 to 4)
TRG0EN_0	0	1	0	1	MTU0.TGRE and MTU0.TCNT
TRG0FN_0	0	1	1	0	MTU0.TGRF and MTU0.TCNT
TRG04ABN_0	0	1	1	1	MTU4.TADCORA and MTU4.TCNT or MTU4.TADCORB and MTU4.TCNT
TRG04ABN_1	1	0	0	0	TPU0.TGRA
TMTRG0AN_0	1	0	0	1	TMR0.TCORA and TMR0.TCNT
TMTRG0AN_1	1	0	1	0	TMR2.TCORA and TMR2.TCNT

### 38.2.9 A/D-Converted Extended Input Control Register (ADEXICR)

Address(es): 0008 9012h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	OCS	TSS	—	—	—	—	—	—	OCSAD	TSSAD
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	TSSAD	Temperature Sensor Output A/D Converted Value Addition Mode Select	0: Temperature sensor output A/D converted value addition mode is not selected 1: Temperature sensor output A/D converted value addition mode is selected	R/W
b1	OCSAD	A/D Internal Reference Voltage A/D Converted Value Addition Mode Select	0: A/D internal reference voltage A/D converted value addition mode is not selected 1: A/D internal reference voltage A/D converted value addition mode is selected	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b8	TSS	Temperature Sensor Output A/D Conversion Select	0: A/D conversion of temperature sensor output is not performed 1: A/D conversion of temperature sensor output is performed	R/W
b9	OCS	A/D Internal Reference Voltage A/D Conversion Select	0: A/D conversion of A/D internal reference voltage is not performed 1: A/D conversion of A/D 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

#### TSSAD Bit (Temperature Sensor Output A/D Converted Value Addition Mode Select)

The TSSAD bit selects A/D conversion for the temperature sensor output. Setting the TSSAD bit to 1 performs A/D conversion of the temperature sensor output successively 2 to 4 times that is set with the ADC[1:0] bits in ADADC and returns the integrated value to the A/D temperature sensor data register (ADTSDR).

The TSSAD bit should be set while the ADST bit in ADCSR is set to 0.

#### OCSAD Bit (A/D Internal Reference Voltage A/D Converted Value Addition Mode Select)

The OCSAD bit selects A/D conversion for the A/D internal reference voltage. Setting the OCSAD bit to 1 performs A/D conversion of the A/D internal reference voltage successively 2 to 4 times that is set with the ADC[1:0] bits in ADADC and returns the integrated value to the A/D internal reference voltage data register (ADOCADR). The OCSAD bit should be set while the ADST bit in ADCSR is set to 0.

#### TSS Bit (Temperature Sensor Output A/D Conversion Select)

The TSS bit selects A/D conversion for the temperature sensor output. When A/D conversion of the temperature sensor output is to be performed, all the bits in the ADANS0 and ADANS1 registers, and the OCS bit should be set to 0. The continuous scan mode (the ADCS bit in ADCSR = 1) should not be selected. The TSS bit should be set while the ADST bit in ADCSR is set to 0.

#### OCS Bit (A/D Internal Reference Voltage A/D Conversion Select)

The OCS bit selects A/D conversion for the A/D internal reference voltage. When A/D conversion of the internal reference voltage is to be performed, all the bits in the ADANS0 and ADANS1 registers, and the TSS bit should be set to 0. The continuous scan mode (the ADCS bit in ADCSR = 1) should not be selected. The OCS bit should be set while the ADST bit in ADCSR is set to 0.

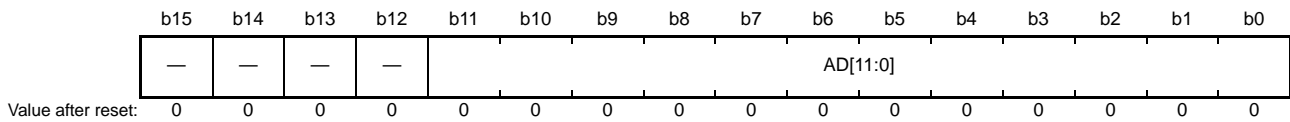
### 38.2.10 A/D Temperature Sensor Data Register (ADTSDR)

ADTSDR is a 16-bit read-only register that stores the A/D conversion results of the temperature sensor output.

The following different formats are used depending on the setting of the A/D data register format select bit (ADRFMT) in ADCER or A/D-converted value addition mode.

- ADCER.ADRFMT = 0 (Setting for flush-right)

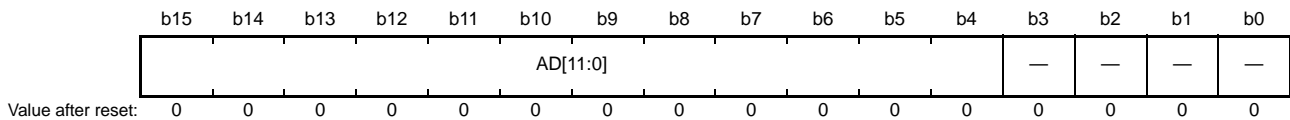
Address(es): 0008 901Ah



Bit	Symbol	Bit Name	Description	R/W
b11 to b0	AD[11:0]	—	12-bit A/D-converted value	R
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- ADCER.ADRFMT = 1 (Setting for flush-left)

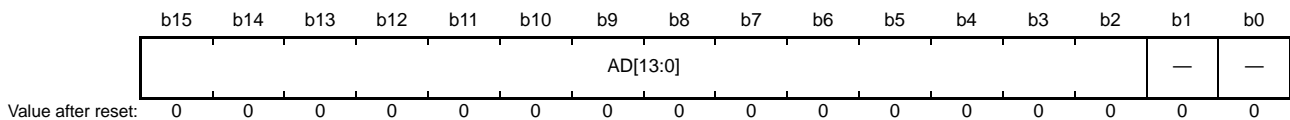
Address(es): 0008 901Ah



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b4	AD[11:0]	—	12-bit A/D-converted value	R

- When A/D-converted value addition mode is selected

Address(es): 0008 901Ah



Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b2	AD[13:0]	—	14-bit A/D-converted value addition result	R

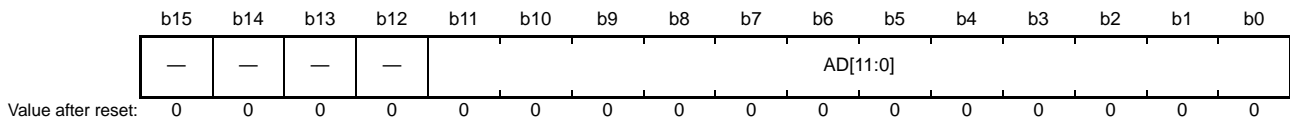
When A/D-converted value addition mode is selected, the AD[13:0] bits in ADTSDR show the value added by the A/D-converted value of the temperature sensor output. In A/D-converted value addition mode, the setting of ADRFMT bit in ADCER becomes invalid and the format of the register becomes flush-left.

### 38.2.11 A/D Internal Reference Voltage Data Register (ADOCDR)

ADOCDR is a 16-bit read-only register that stores the A/D conversion results of the A/D internal reference voltage. The following different formats are used depending on the setting of the A/D data register format select bit (ADRFMT) in ADCER or A/D-converted value addition mode.

- ADCER.ADRFMT = 0 (Setting for flush-right)

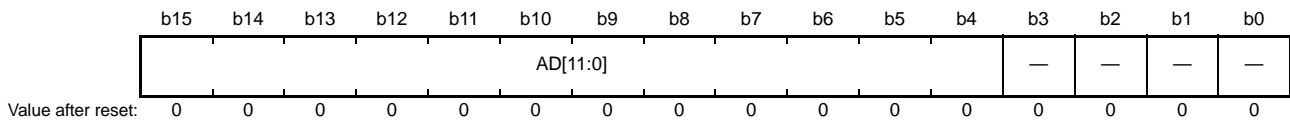
Address(es): 0008 901Ch



Bit	Symbol	Bit Name	Description	R/W
b11 to b0	AD[11:0]	—	12-bit A/D-converted value	R
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- ADCER.ADRFMT = 1 (Setting for flush-left)

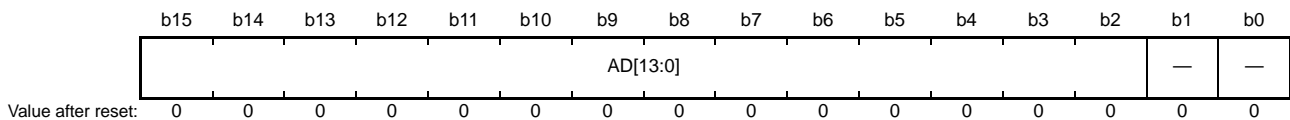
Address(es): 0008 901Ch



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b4	AD[11:0]	—	12-bit A/D-converted value	R

- When A/D-converted value addition mode is selected

Address(es): 0008 901Ch



Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b2	AD[13:0]	—	14-bit A/D-converted value addition result	R

When A/D-converted value addition mode is selected, the AD[13:0] bits in ADOCADR show the value added by the A/D-converted value of the A/D internal reference voltage. In A/D-converted value addition mode, the setting of the ADRFMT bit in ADCER becomes invalid and the format of the register becomes flush-left.

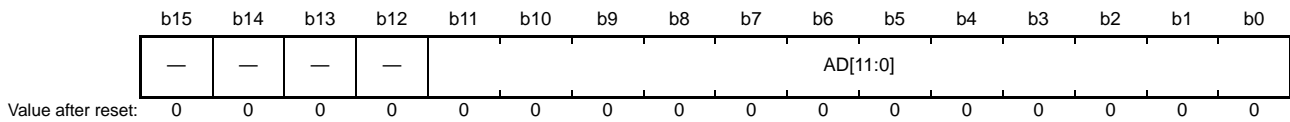
### 38.2.12 A/D Data Register y (ADDRy) (y = 0 to 20)

ADDRy are 16-bit read-only registers which store the A/D conversion results of channels AN000 to AN020.

The A/D data registers use the following different formats depending on the setting of the A/D data register format select bit (ADRFMT) in ADCER or A/D-converted value addition mode.

- ADCER.ADRFMT = 0 (Setting for flush-right)

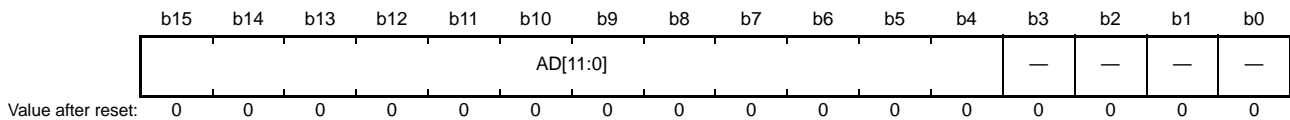
Address(es): 0008 9020h to 0008 9048h



Bit	Symbol	Bit Name	Description	R/W
b11 to b0	AD[11:0]	—	12-bit A/D-converted value	R
b15 to b12	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

- ADCER.ADRFMT = 1 (Setting for flush-left)

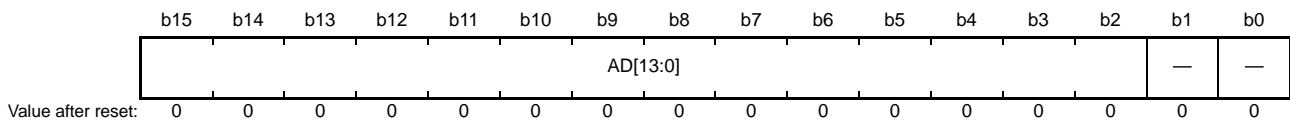
Address(es): 0008 9020h to 0008 9048h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b4	AD[11:0]	—	12-bit A/D-converted value	R

- When A/D-converted value addition mode is selected

Address(es): 0008 9020h to 0008 9048h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b2	AD[13:0]	—	14-bit A/D-converted value addition result	R

When A/D-converted value addition mode is selected, the AD[13:0] bits in ADDR<sub>y</sub> show the value added by the A/D-converted value of the respective channels. In A/D-converted value addition mode, the setting of the ADRFMT bit in ADCER becomes invalid and the format of the register becomes flush-left.

The following minimum and maximum values apply to channels on which A/D-converted value addition mode is selected.

First conversion:  $0000h \leq ADDR_y (y = 0 \text{ to } 20) \leq 3FFCh$

(ADDR<sub>y</sub> (y = 0 to 20): Bits 15 and 14 = 00b, bits 13 to 2 = AD11 to AD0, bits 1 and 0 = 00b)

Second conversion:  $0000h \leq ADDR_y (y = 0 \text{ to } 20) \leq 7FF8h$

(ADDR<sub>y</sub> (y = 0 to 20): Bit 15 = 0, bits 14 to 2 = AD12 to AD0, bits 1 and 0 = 00b)

Third conversion:  $0000h \leq ADDR_y (y = 0 \text{ to } 20) \leq BFF4h$

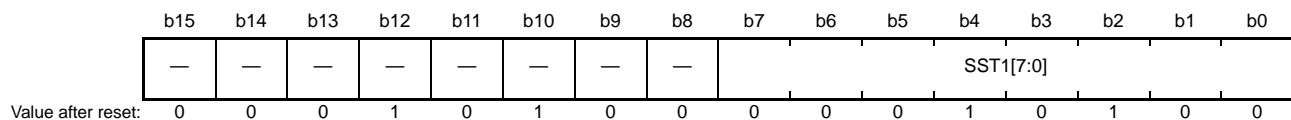
(ADDR<sub>y</sub> (y = 0 to 20): Bits 15 to 2 = AD13 to AD0, bits 1 and 0 = 00b)

Fourth conversion:  $0000h \leq ADDR_y (y = 0 \text{ to } 20) \leq FFF0h$

(ADDR<sub>y</sub> (y = 0 to 20): Bits 15 to 2 = AD13 to AD0, bits 1 and 0 = 00b)

### 38.2.13 A/D Sampling State Register 01 (ADSSTR01)

Address(es): 0008 9060h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	SST1[7:0]	Sampling Time 1 Setting	These bits set the sampling time as a number of states (cycles of the A/D conversion clock) from 10 to 255.	R/W
b15 to b8	—	Reserved	The write value should be 00010100b. These bits are read as the same value.	R/W

#### SST1[7:0] Bits (Sampling Time 1 Setting)

These bits are used to set the sampling time for the analog input of the selected channel.

One state refers to one cycle of the A/D conversion clock (ADCLK), so when the frequency of the clock is 50 MHz, one state becomes 20 ns. The default setting is 20 states. The sampling time can be adjusted if the source impedance of the analog input signal is high enough that the sampling time is too short or the ADCLK frequency is too low for the current setting. Only set the SST1[7:0] bits while the ADSCR.ADST bit is 0.

The sampling time setting should be in the range from 10 to 255 states, so that the sampling time is longer than 0.4 μs.

### 38.2.14 A/D Sampling State Register 23 (ADSSTR23)

Address(es): 0008 9070h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	—	Reserved	These write value should be 00010100b. These bits are read as the same value.	R/W
b15 to b8	SST2[7:0]	Sampling Time 2 Setting	These bits set the sampling time in 10 to 255 states.	R/W

#### SST2[7:0] Bits (Sampling Time 2 Setting)

These bits set the sampling time for the temperature sensor output. One state is one cycle of the A/D conversion clock (ADCLK). When the ADCLK is running at 50 MHz, one state is 20 ns. The initial value is 20 states. If the impedance of the analog input signal source is too high for this to secure sufficient sampling time or if the ADCLK is at a low frequency, the sampling time can be adjusted. The SST2[7:0] bits should be set while the ADCSR.ADST bit is 0. The sampling time setting for the temperature sensor output should be in the range from 10 to 255 states, so that the sampling time is longer than 0.4 μs.

### 38.3 Operation

#### 38.3.1 Scanning Operation

In Scanning, A/D conversion is performed sequentially on the analog inputs of the specified channels.

A scan conversion is performed in two operating modes: single scan mode and continuous scan mode. In the single scan mode, one or more specified channels are scanned once. In the continuous scan mode, one or more specified channels are scanned repeatedly until the ADST bit in ADCSR is set to 0 by software.

In either mode, A/D conversion is performed for ANn channels selected by the ADANS0 and ADANS1 registers, starting from the channel with the lowest number n.

A/D conversion is performed in single scan mode, when the temperature sensor output or A/D internal reference voltage is selected. This operation is similar to the scanning operation in which only one channel is selected in single scan mode.

#### 38.3.2 Single Scan Mode

##### 38.3.2.1 Basic Operation

In basic operation of single scan mode, A/D conversion is to be performed for only once on the analog input of the specified single channel as below. In selected channel scanning, temperature sensor output A/D conversion select bit (TSS) and A/D internal reference voltage A/D conversion select bit (OCS) in ADEXICR should both be set to 0 (non-selection).

- (1) A/D conversion for the selected channel is started when the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, synchronous trigger (MTU, TPU, or TMR), or asynchronous trigger input. A/D conversion is performed for ANn channels selected by the ADANS0 and ADANS1 registers, starting from the channel with the lowest number n.
- (2) When A/D conversion for single channel is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy) of the channel.
- (3) When A/D conversion of all selected channels is completed, if the ADCSR.ADIE bit is set to 1 (enables S12ADI0 interrupt generation upon scan conversion completion), an S12ADI0 interrupt request is generated.
- (4) The ADCSR.ADST bit remains at 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion of all selected channels ends. Then the 12-bit A/D converter enters a wait state.

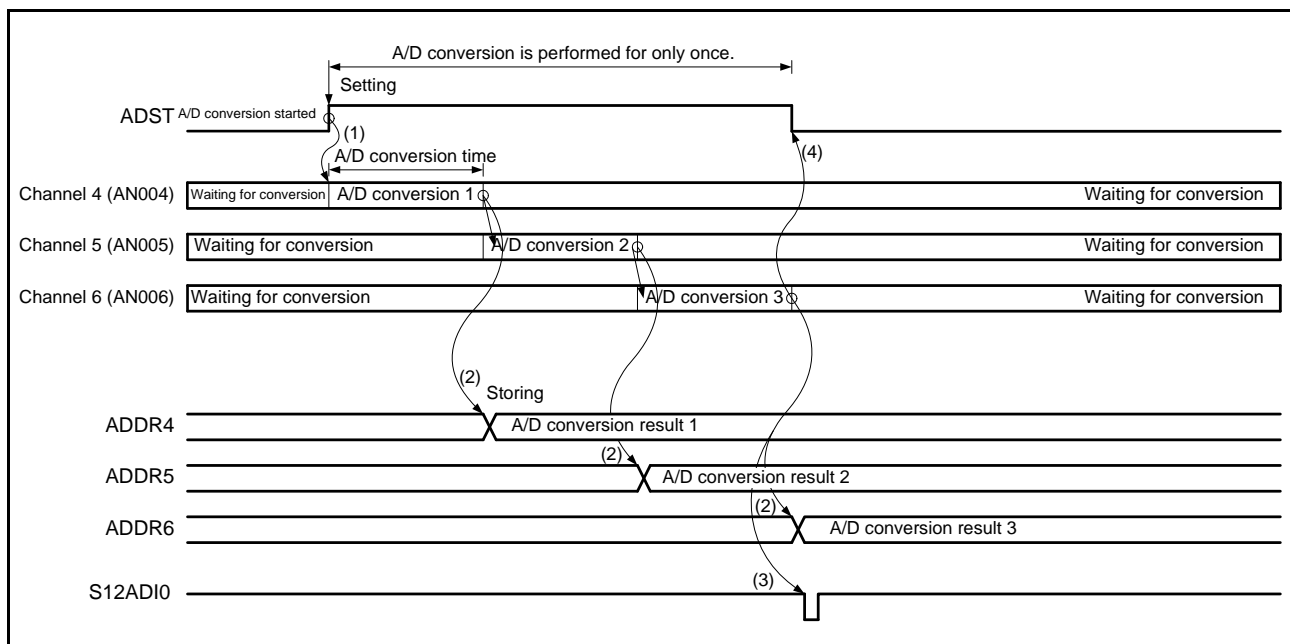


Figure 38.3 Example of Operation of Single Scan Mode (Basic Operation: AN004 to AN006 Selection)



### 38.3.2.2 A/D Conversion when selecting Temperature Sensor Output/Internal Reference Voltage

When A/D conversion for the temperature sensor output or A/D internal reference voltage is selected, it is performed in single scan mode. The operation is as follows.

Non-selection must be set for all channels (set all bits in ADANS0.ANS0[15:0] and ADANS1.ANS1[4:0] to 0).

Furthermore, when A/D conversion for the temperature sensor output is selected, set A/D internal reference voltage A/D conversion select bit (OCS) in ADEXICR to 0 (non-selection). When A/D conversion of A/D internal reference voltage is selected, set temperature sensor output A/D conversion select bit (TSS) in ADEXICR to 0 (non-selection).

- (1) When the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, the synchronous trigger (MTU, TPU, TMR), or the asynchronous trigger input, A/D conversion is started for the temperature sensor output or A/D internal reference voltage.
- (2) When A/D conversion is completed, the A/D conversion results are stored into the corresponding A/D temperature sensor data register (ADTSDR) or the A/D internal reference voltage data register (ADOCDR). If the ADIE bit in ADCSR is set to 1 (enables S12ADI0 interrupt generation upon scan conversion completion), an S12ADI0 interrupt request is generated.
- (3) The ADST bit in ADCSR remains at 1 (A/D conversion start) during A/D conversion, and is automatically cleared to 0 when A/D conversion ends. Then the 12-bit A/D converter enters a wait state.

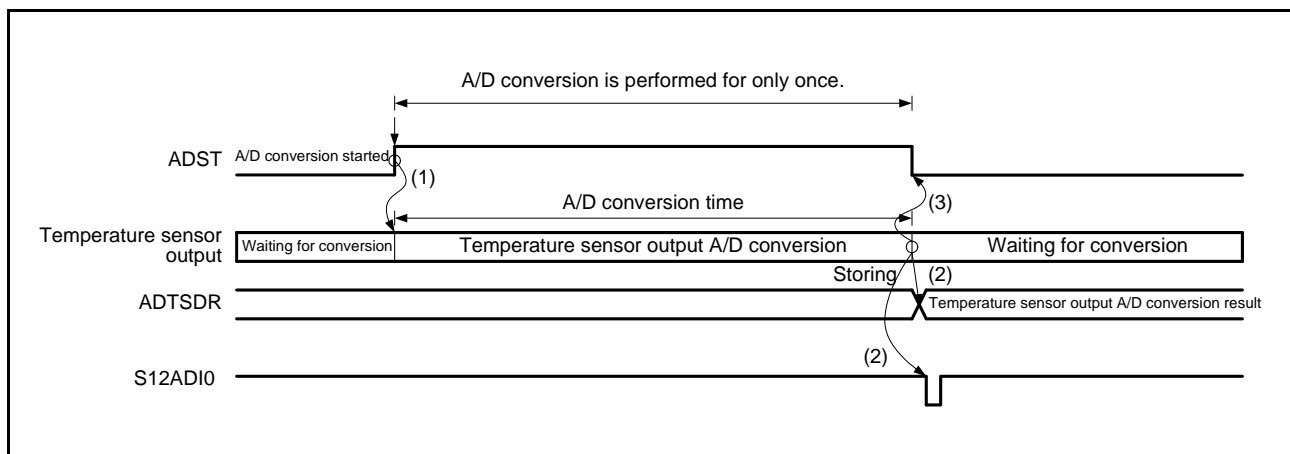


Figure 38.4 Example of Single Scan Mode (Temperature Sensor Output Selection)

### 38.3.3 Continuous Scan Mode

#### 38.3.3.1 Basic Operation

In basic operation of continuous scan mode, A/D conversion is to be performed sequentially on the analog inputs of the specified channels as below.

In continuous scan mode, temperature sensor output A/D conversion select bit (TSS) and A/D internal reference voltage A/D conversion select bit (OCS) in ADEXICR should both be set to 0 (non-selection).

- (1) A/D conversion for the selected channel is started when the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, synchronous trigger (MTU, TPU, or TMR), or asynchronous trigger input. A/D conversion is performed for ANn channels selected by the ADANS0 and ADANS1 registers, starting from the channel with the lowest number n.
- (2) When A/D conversion for the 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 selected channels is completed, if the ADIE bit in ADCSR is set to 1 (enables S12ADI0 interrupt generation upon scan conversion completion), an S12ADI0 interrupt request is generated. Furthermore, the 12-bit A/D converter continuously starts A/D conversion for ANn channels selected by the ADANS0 and ADANS1 registers, starting from the channel with the lowest number n.
- (4) The ADST bit in ADCSR is not cleared automatically, and steps 2 and 3 are repeated as long as the ADST bit remains set to 1 (A/D conversion start). When the ADST in ADCSR 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) If the ADST bit in ADCSR is later set to 1 (A/D conversion start), A/D conversion starts again for ANn channels selected by the ADANS0 and ADANS1 registers, starting from the channel with the lowest number n.

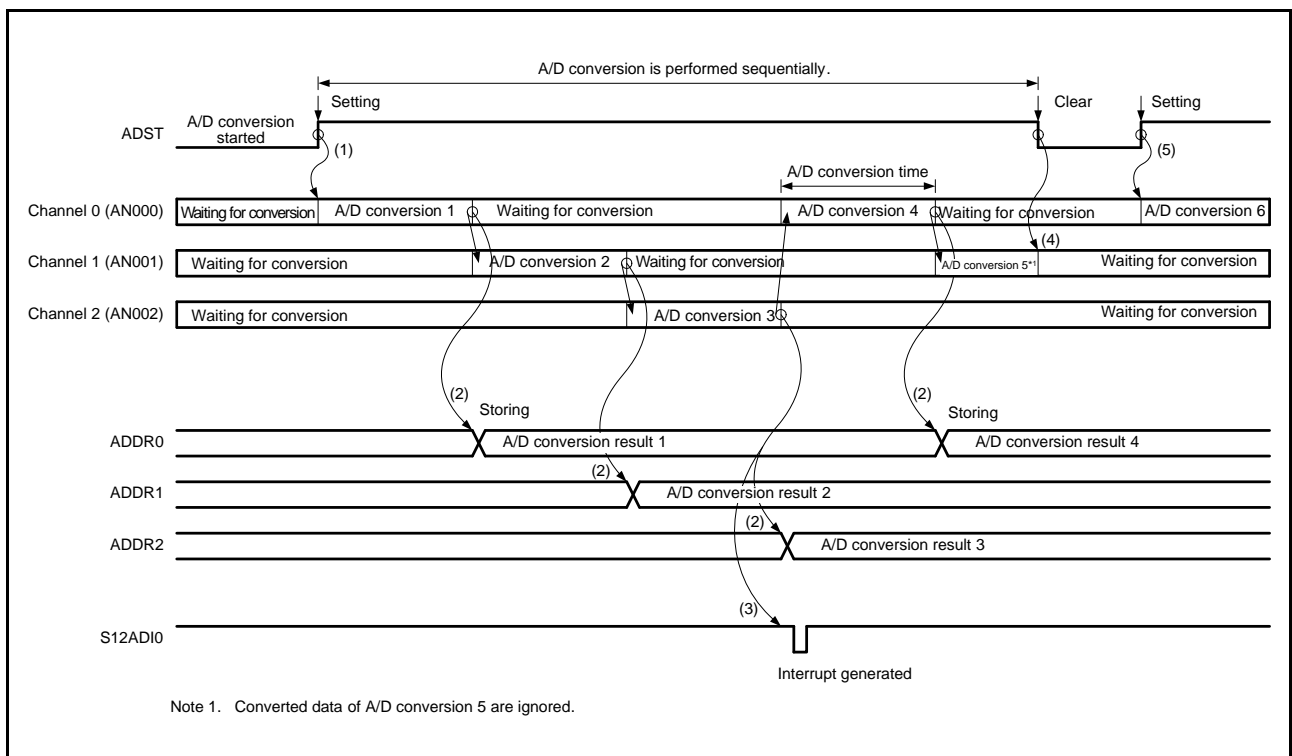


Figure 38.5 Example of Operation of Continuous Scan Mode (Basic Operation: AN000 to AN002 Selection)

### 38.3.4 Analog Input Scan Time

The times required for the processing listed in Table 38.6 (Table 38.7) determine the scan time.

**Table 38.6 Processing During a Scan Operation**

Processing	Description
Scan start processing	Processing to start scanning in response to a software trigger, synchronous trigger, or asynchronous trigger
Analog input A/D conversion processing* <sup>1</sup>	Sample-and-hold and successive-approximation processing by the 12-bit A/D converter
Scan end processing	Processing to store A/D-converted data and output the S12ADI0 interrupt

Note 1. Time for this processing varies with the settings for numbers of sampling states in the SST1[7:0] to SST2[7:0] bit fields in A/D sampling state registers 01 and 23 (ADSSTR01 and ADSSTR23).

**Table 38.7 Times Required in Scanning Operations (as Numbers of ADCLK and PCLK Cycles)**

Item	Symbol	Type or Condition	Cycles
Scan start processing time	$t_D$	Synchronous trigger (MTU, TPU, TMR) or software trigger	2 PCLK + 3 ADCLK* <sup>1</sup>
		Asynchronous trigger (ADTRG0#)	4 PCLK + 3 ADCLK* <sup>1</sup>
A/D conversion processing time	$t_{CONV}$	ADSSTRxx.SSTy[7:0] Initial value 14h* <sup>2</sup>	50 ADCLK
Scan end processing time	$t_{ED}$	—	1 PCLK + 2 ADCLK* <sup>3</sup>

Note 1. Two and four cycles of the PCLK are the times taken to write to the register and for trigger processing. Three cycles of the ADCLK is the maximum time from setting of the ADCSR.ADST bit to the start of 12-bit A/D converter operations.

Note 2. ADSSTRxx.SSTy[7:0] indicates the ADSSTR01.SST1[7:0], and ADSSTR23.SST2[7:0] bit fields.

Note 3. Two cycles of the ADCLK is the maximum time, and an S12ADI0 interrupt is output one cycle of the PCLK later. For processing time when scanning is forcibly stopped, refer to section 38.5.3, Notes on Restarting A/D Conversion after a Forcible Stop.

The scan conversion time ( $t_{SCAN}$ ) in single-cycle scan mode for which the number of selected channels is  $n$  can be determined as follows:

$$t_{SCAN} = t_D + (t_{CONV} \times n) + t_{ED}$$

- The scan conversion time for the first cycle in continuous scan mode is  $t_{SCAN}$  for single-cycle scan minus  $t_{ED}$ .
- The scan conversion time for the second and subsequent cycles in continuous scan mode is a fixed period equal to  $(t_{CONV} \times n)$ .
- A/D conversion processing time ( $t_{CONV}$ ) in states is given below.  
A/D conversion processing time ( $t_{CONV}$ ) = 30 (fixed) + setting of the ADSSTRxx.SSTy[7:0] bits

### 38.3.5 Usage Example of ADDRy Register Automatic Clearing Function

Setting the ADCER.ACE bit to 1 selects automatic clearing of each A/D data register (ADDRy) to 0000h when the given register is read by the CPU, DTC, or DMAC.

There is no function for automatically clearing register ADTSDR or ADOCDR.

When the analog input voltage is not in the region of 0 V, the automatic clearing function can be used to detect failure to update the A/D data register y (ADDRy). Since 0000h can be a normal result of A/D conversion when the analog input voltage is in the region of 0 V, the automatic clearing function is not usable.

Examples in which the function to automatically clear the ADDRy registers are enabled and disabled are shown below. In a case where the ACE bit in ADCER is 0 (automatic clearing is disabled), if the A/D conversion result (0222h) is not written to the ADDRy registers for some reason, the old data (0111h) will become the ADDRy value. Furthermore, if this ADDRy value is read into a general register using an A/D scan conversion end interrupt, the old data (0111h) can be saved in the general register. When checking data that has not been updated, it is necessary to frequently save the old data in the RAM or a general register.

In a case where the ACE bit in ADCER is 1 (automatic clearing is enabled), when ADDRy = 0111h is read by the CPU, DTC, or DMAC, ADDRy is automatically cleared to 0000h. After that, if the A/D conversion result of 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 conversion 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 whether the read data value is 0000h.

### 38.3.6 A/D-Converted Value Addition Function

The same channel is A/D converted continuously two to four times and the sum 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.

A/D converted value addition function can be used at the time of selecting the channel selection analog input A/D conversion, temperature sensor output A/D conversion, or A/D internal reference voltage A/D conversion.

### 38.3.7 Starting A/D Conversion with Asynchronous Trigger

The A/D conversion can be started by the input of asynchronous trigger. To start up the A/D conversion by an asynchronous trigger after setting the A/D conversion start trigger select bits (ADSTRS[3:0]) in ADSTRGR to 00h and applying a high-level signal to the asynchronous trigger (ADTRG0# pin), both the TRGE and EXTRG bits in ADCSR should be set to 1. Then if the asynchronous trigger (ADTRG0# pin) is driven low, the ADST bit is set to 1 and the A/D conversion is started. In this case, the low-level pulse width must be at least 1.5 PCLK clock cycles. For the time required for the A/D conversion to start after the ADCSR.ADST bit is set, refer to section 38.5.3, Notes on Restarting A/D Conversion after a Forcible Stop.

Figure 38.6 shows a timing for the ADCSR.ADST bit setting by the asynchronous trigger input.

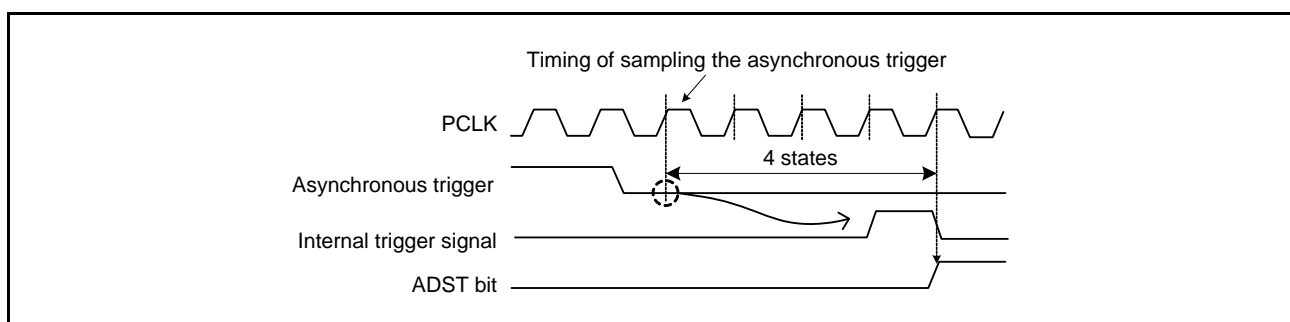


Figure 38.6 Asynchronous Trigger Input Timing

### 38.3.8 Starting A/D Conversion with Synchronous Trigger from Peripheral Modules

A/D conversion can be started by synchronous trigger of the MTU, TPU, or TMR. To start up the A/D conversion by a synchronous trigger, the TRGE bit in ADCSR should be set to 1, the EXTRG bit in ADCSR should be cleared to 0, and the A/D conversion start source should be selected by ADSTRS[3:0] bits in ADSTRGR.

Table 38.8 lists A/D conversion start sources to be selected.

**Table 38.8 A/D Conversion Start Sources to Be Selected**

Module	Source	Remarks	ADSTRS[3]	ADSTRS[2]	ADSTRS[1]	ADSTRS[0]
S12AD	ADST	Software trigger	—	—	—	—
External input	ADTRG0#	Asynchronous trigger	0	0	0	0
MTU	TRG1N	TRG0AN_0	0	0	0	1
	TRG2N	TRG0BN_0	0	0	1	0
	TRG3N	TRGAN_0	0	0	1	1
TPU	TRG4N	TRGAN_1	0	1	0	0
MTU	TRG5N	TRG0EN_0	0	1	0	1
	TRG6N	TRG0FN_0	0	1	1	0
	TRG7N	TRG4ABN_0	0	1	1	1
TPU	TRG8N	TRG4ABN_1	1	0	0	0
TMR	TRG9N	TMTRG0AN_0	1	0	0	1
	TRG10N	TMTRG0AN_1	1	0	1	0

Note: • If the ADTRG0# signal is to be set as the trigger for the start of A/D conversion, set the corresponding P0.PDR.B7, P1.PDR.B6, or P2.PDR.B5 bit to 0 (input port operation) and the corresponding P0.PMR.B7, P1.PMR.B6, or P2.PMR.B5 pin to 1 (enabling the input buffer for the given pin). For details, see section 20, I/O Ports and section 21, Multi-Function Pin Controller (MPC).

### 38.3.8.1 Starting A/D Conversion Using MTU TRG0AN\_0 and TRG0BN\_0

Upon the TGRA input capture/compare match of the MTU0, the trigger signal TRG0AN\_0 can be generated and A/D conversion can be started with TRG0AN\_0.

Similarly, upon the TGRB input capture/compare match of the MTU0, the trigger signal TRG0BN\_0 can be generated and A/D conversion can be started.

Figure 38.7 shows the connection of MTU TRG0AN\_0 output and TRG0BN\_0 output to the A/D converter.

To start A/D conversion upon the TRGA input capture/compare match of the MTU0, set ADCSR.TRGE to 1, set ADCSR.EXTRG to 0, and set ADSTRGR.ADSTRS[3:0] to 0001b (source: TRG1N; corresponding trigger: TRG0AN\_0).

To start A/D conversion upon the TRGB input capture/compare match of the MTU0, set ADCSR.TRGE to 1, set ADCSR.EXTRG to 0, and set ADSTRGR.ADSTRS[3:0] to 0010b (source: TRG2N; corresponding trigger: TRG0BN\_0).

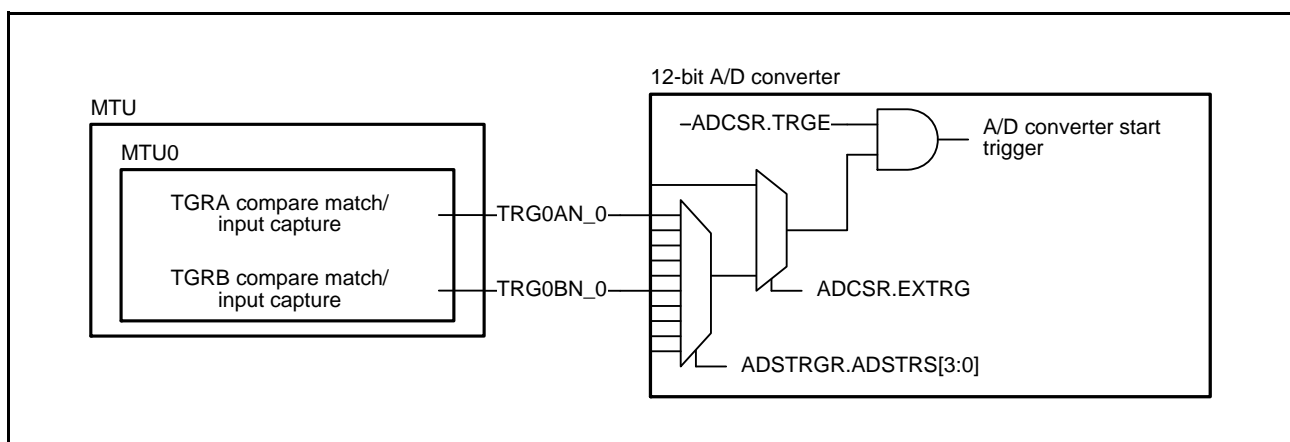


Figure 38.7 Connection of MTU TRG0AN\_0 Output and TRG0BN\_0 Output to A/D Converter

### 38.3.8.2 Starting A/D Conversion Using MTU TRGAN\_0 and TPU TRGAN\_1

Upon the TRGA input capture/compare match of the MTU0 to MTU4 and the TCNT underflow (trough) of the MTU4 in complementary PWM mode, the trigger signal TRGAN\_0 can be generated and A/D conversion can be started.

Similarly, upon the TRGA input capture/compare match of the TPU0 to TPU4 in the TPU (unit 0), the trigger signal TRGAN\_1 can be generated and A/D conversion can be started.

Figure 38.8 shows the connection of MTU TRGAN\_0 output and TPU (unit 0) TRGAN\_1 output to the A/D converter.

When A/D conversion is to be started by input capture to or compare match with TGRA in MTU0 to MTU4 and underflows (troughs) of TCNT of MTU4 in complementary PWM mode, set the ADCSR.TRGE bit to 1, the ADCSR.EXTRG bit to 0, the ADSTRGR.ADSTRS[3:0] bits to 0011b (selecting TRG3N as the source and TRGAN\_0 as the corresponding trigger), and the MTUn.TIER.TTGE bits (n = 0 to 4) and MTU4.TIER.TTGE2 bit to 1.

Also, when A/D conversion is to be started by input capture to or compare match with TGRA in TPU0 to TPU4, set the ADCSR.TRGE bit to 1, the ADCSR.EXTRG bit to 0, the ADSTRGR.ADSTRS[3:0] bits to 0100b (selecting TRG4N as the source and TRGAN\_1 as the corresponding trigger), and the TPU<sub>n</sub>TIER.TTGE bits (n = 0 to 4) to 1.

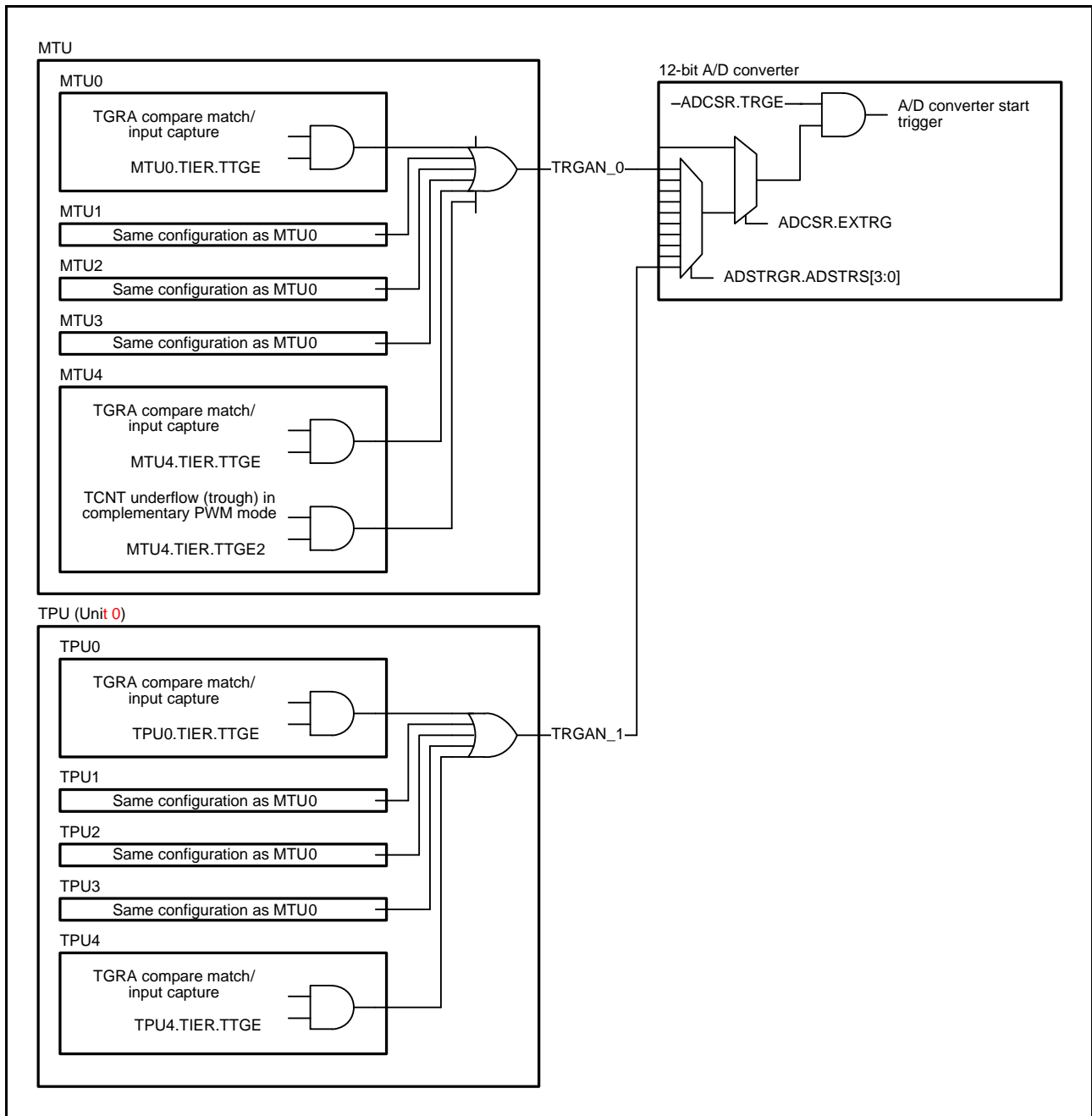


Figure 38.8 Connection of MTU TRGAN\_0 Output and TPU (unit 0) TRGAN\_1 Output to A/D Converter

### 38.3.8.3 Starting A/D Conversion Using MTU TRG0EN\_0 and TRG0FN\_0

Upon the TGRE compare match of the MTU0, the trigger signal TRG0EN\_0 can be generated and A/D conversion can be started with TRG0EN\_0. Similarly, upon the TGRF compare match of the MTU0, the trigger signal TRG0FN\_0 can be generated and A/D conversion can be started.

Figure 38.9 shows the connection of MTU TRG0EN\_0 output and TRG0FN\_0 output to the A/D converter.

To start A/D conversion upon the TGRE compare match of the MTU0, set ADCSR.TRGE to 1, set ADCSR.EXTRG to 0, and set ADSTRGR.ADSTRS[3:0] to 0101b (source: TRG5N; corresponding trigger: TRG0EN\_0).

To start A/D conversion upon the TGRF compare match of the MTU0, set ADCSR.TRGE to 1, set ADCSR.EXTRG to 0, and set ADSTRGR.ADSTRS[3:0] to 0110b (source: TRG6N; corresponding trigger: TRG0FN\_0).

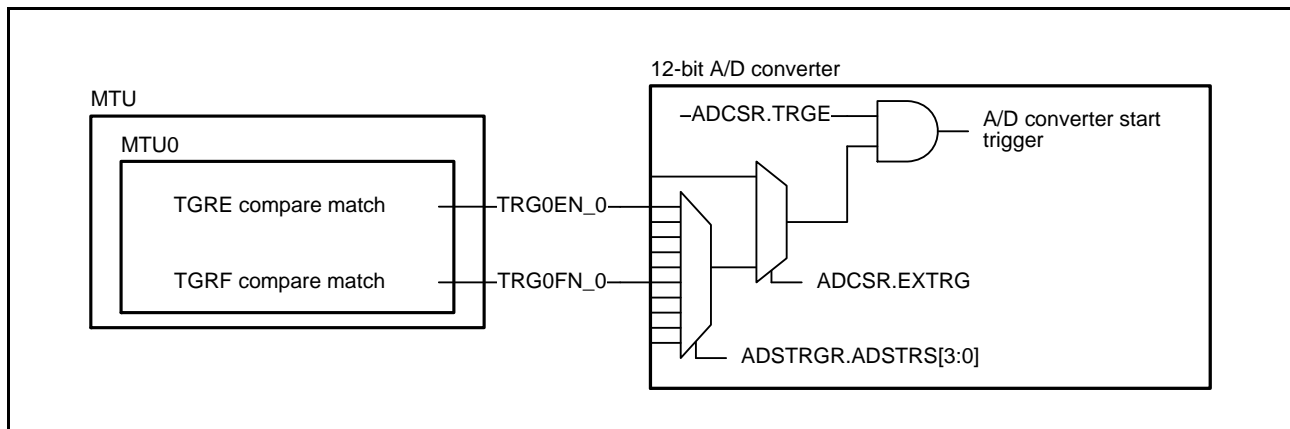


Figure 38.9 Connection of MTU TRG0EN\_0 Output and TRG0FN\_0 Output to A/D Converter



### 38.3.8.4 Starting A/D Conversion Using MTU TRG4ABN\_0 and TPU TRG4ABN\_1

Upon the compare match of the MTU4 generated using the delayed A/D conversion start request function, the trigger signal TRG4ABN\_0 can be generated and A/D conversion can be started. Similarly, upon the TRGA input capture/compare match of the TPU0 in the TPU (unit 0), the trigger signal TRG4ABN\_1 can be generated and A/D conversion can be started.

Figure 38.10 shows the connection of MTU TRG4ABN\_0 output and TPU (unit 0) TRG4ABN\_1 output to the A/D converter. To use the delayed A/D conversion start request function of the MTU4 to start A/D conversion upon the compare match between the TADCORA and TCNT countup, set ADCSR.TRGE to 1, set ADCSR.EXTRG to 0, set ADSTRGR.ADSTRS[3:0] to 0111b (source: TRG7N; corresponding trigger: TRG4ABN\_0), set the cycle to MTU4.TADCOBRA/MTU4.TADCOBRB and MTU4.TADCORA/MTU4.TADCORB, and set MTU4.TADCR.UT4AE to 1.

Also, when A/D conversion is to be started by input capture to or compare match with TGRA in TPU0, set the ADCSR.TRGE bit to 1, the ADCSR.EXTRG bit to 0, the ADSTRGR.ADSTRS[3:0] bits to 1000b (selecting TRG8N as the source and TRG4ABN\_1 as the corresponding trigger), and the TPU0.TIER.TTGE bit to 1.

For details of the delayed A/D conversion start request function, see section 22.3.9, A/D Converter Start Request Delaying Function.

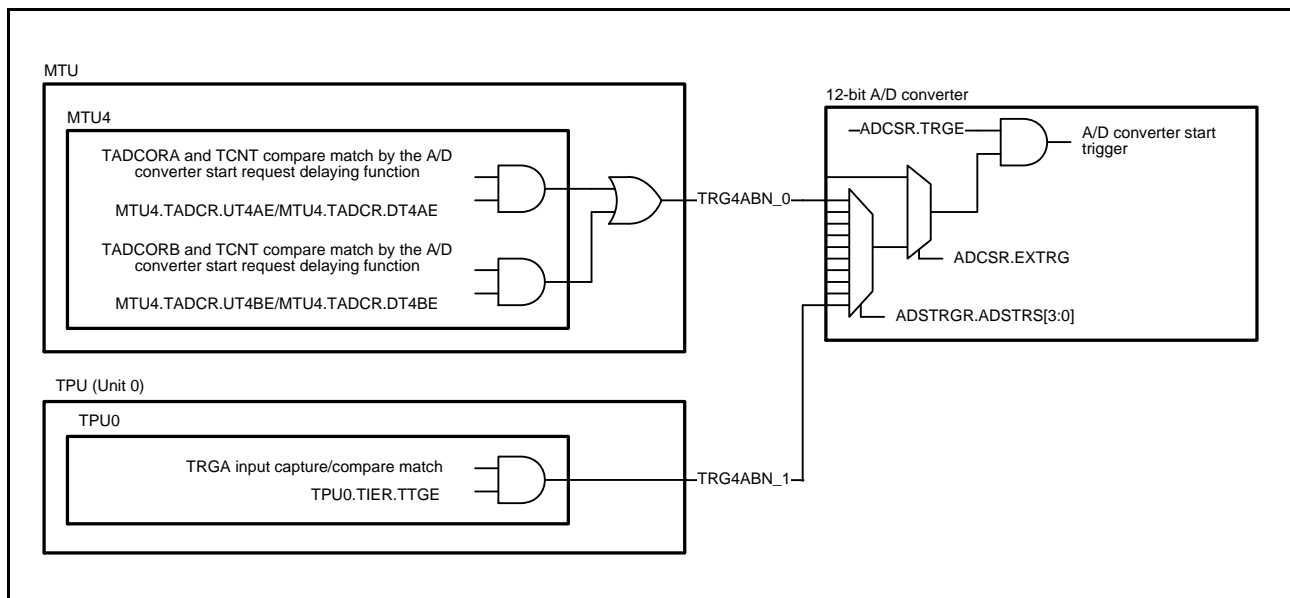


Figure 38.10 Connection of MTU TRG4ABN\_0 Output and TPU (Unit 0) TRG4ABN\_1 Output to A/D Converter

### 38.3.8.5 Starting A/D Conversion Using TMR TMTRG0AN\_0 and TMTRG0AN\_1

Upon the TCORA compare match (compare match A) of the TMR0 in the TMR (unit 0), A/D conversion can be started. Similarly, upon the TCORA compare match (compare match A) of the TMR2 in the TMR (unit 1), A/D conversion can be started.

Figure 38.11 shows the connection of TMR (unit 0 and unit 1) TMTRG0AN\_0 output and TMTRG0AN\_1 output to the A/D converter.

To start A/D conversion upon the TCORA compare match (compare match A) of the TMR0 in the TMR (unit 0), set ADCSR.TRGE to 1, set ADCSR.EXTRG to 0, set ADSTRGR.ADSTRS[3:0] to 1001b (source: TRG9N; corresponding trigger: TMTRG0AN\_0), and set TMR0.TCSR.ADTE to 1.

To start A/D conversion upon the TCORA compare match (compare match A) of the TMR2 in the TMR (unit 1), set ADCSR.TRGE to 1, set ADCSR.EXTRG to 0, set ADSTRGR.ADSTRS[3:0] to 1010b (source: TRG10N; corresponding trigger: TMTRG0AN\_1), and set TMR2.TCSR.ADTE to 1.

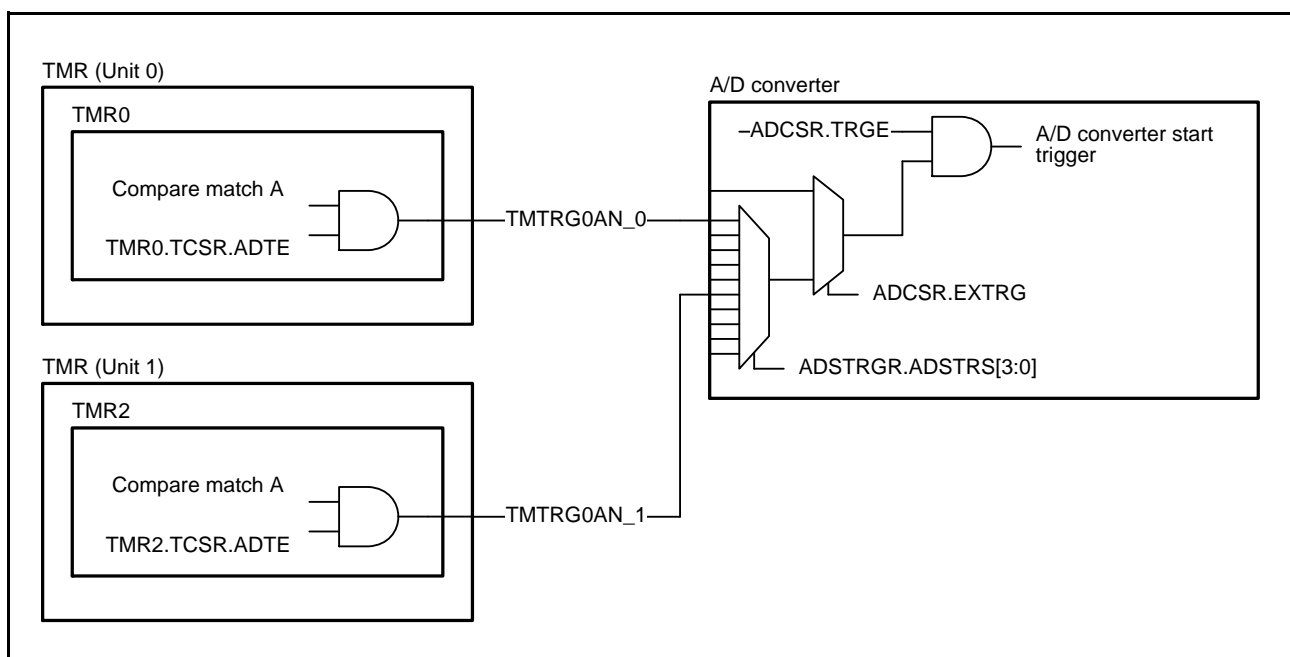


Figure 38.11 Connection of TMR (Unit 0 and Unit 1) TMTRG0AN\_0 Output and TMTRG0AN\_1 Output to A/D Converter

## 38.4 Interrupt Sources and DMA Transfer Requests

### 38.4.1 Interrupt Request on Completion of Each Scanning Conversion

The 12-bit A/D converter can send a scan end interrupt request S12ADI0 interrupt to the CPU.

By setting the ADIE bit in ADCSR to 1, an S12ADI0 interrupt is enabled; by clearing the ADIE bit to 0, an S12ADI0 interrupt is disabled.

In addition, the DTC or DMAC can be started up when an S12ADI0 interrupt is generated. Using an S12ADI0 interrupt to allow the DMAC to read the converted data enables continuous conversion without burden on software.

For details on DTC settings, see section 19, Data Transfer Controller (DTCa), and for details on DMAC settings, see section 18, DMA Controller (DMACA).

## 38.5 Usage Notes

### 38.5.1 Notes on Reading Data Registers

The A/D data registers, A/D temperature sensor data register, and A/D internal reference voltage data register should be read in the units of word. If read in the units of byte that is classified into upper byte and the lower byte, the A/D converted values read in the units of upper byte are likely to vary from the A/D converted values read in the units of lower byte. Thus, the reading should not be carried out in the units of byte.

### 38.5.2 Notes on Forcibly Stopping A/D Conversion

When an asynchronous trigger or a synchronous trigger has been selected as the condition for starting A/D conversion, an unintended A/D conversion may start by just setting ADCSR.ADST bit to 0 to stop A/D conversion forcibly. Therefore, to make the 12-bit A/D convertor stop for sure, set the ADCSR.TRGE bit to 0 to select the software trigger as the trigger to start A/D conversion, and then set the ADCSR.ADST bit to 0 (to stop A/D conversion).

### 38.5.3 Notes on Restarting A/D Conversion after a Forcible Stop

Due to the analog section of the converter, although setting the ADCSR.ADST bit to 1 restarts operation of the analog section of the A/D converter, this requires three cycles of the ADCLK. Although the analog section of the 12-bit A/D converter stops when the ADCSR.ADST bit is set to 0 to forcibly stop A/D conversion, this requires time equivalent to two cycles of the ADCLK.

### 38.5.4 Module-Stop Function Setting

Operation of the 12-bit A/D converter can be disabled or enabled using the module stop control register. The initial setting is for operation of the 12-bit A/D converter to be stopped. Cancellation of the module-stop state makes it possible to access registers. After the cancellation, wait for 10 ms to start A/D conversion. For details, see section 11, Low Power Consumption.

### 38.5.5 Notes on Entering Low Power Consumption States

When entering module-stop mode or software standby mode, make sure to stop the A/D conversion. To do so, set the ADST bit in ADCSR to 0, and allow time for disabling of the analog input to the 12-bit A/D converter.

Follow the procedure given below to ensure that this time is secured.

1. Set the ADCSR.TRGE bit to 0 (software trigger).
2. Clear the ADCSR.ADST bit to 0.
3. Set the ADCSR.CKS[1:0] bits to 11b (PCLK).
4. After confirming that the A/D converter has been disabled (at least six cycles of the PCLK must be secured as waiting time until A/D conversion is stopped), place the LSI in the module-stop state mode or software standby mode.

### 38.5.6 Notes on Canceling Software Standby Mode

When the software standby mode is canceled, wait until the crystal oscillation stabilization time or PLL-circuit stabilization time has elapsed. Then, wait for another 10 ms to start the A/D conversion. For details, see section 11, Low Power Consumption.

### 38.5.7 Influences on Absolute Precision

Adding capacitors produces coupling with the ground, allowing noise from the ground to adversely affect absolute precision. Be sure to make the connection to an electrically stable ground such as AVSS0.

Care is also required to ensure that filter circuits do not interfere with digital signals or act as antennas on the mounting board.

### 38.5.8 Analog Power Supply Pin Setting Range

If this LSI is used outside the range of the following voltage settings, the reliability of the LSI may be adversely affected.

- Analog Input Voltage Range

The voltage applied to analog input pin ANn should be in the range  $VREFL0 \leq V_{AN} \leq VREFH0$ .

- Relationship between power supply pins (AVCC0 – AVSS0, VREFH0 – VREFL0, VCC – VSS)

Establish the relationship of  $AVCC0 = VCC$  and  $AVSS0 = VSS$ . As shown in Figure 38.12, connect a 0.1- $\mu$ F capacitor between each of the power supplies to form a closed loop in the shortest way so that  $AVCC0 = VCC$  and  $VREFL0 = AVSS0 = VSS$  in the source side. When the A/D converter is not used, set  $VREFH0 = AVCC0 = VCC$  and  $VREFL0 = AVSS0 = VSS$ .

- Range of VREFH0 Pin Settings

The setting of the reference voltage using the VREFH0 pin should be in the range of  $VREFH0 \leq AVCC0$ .

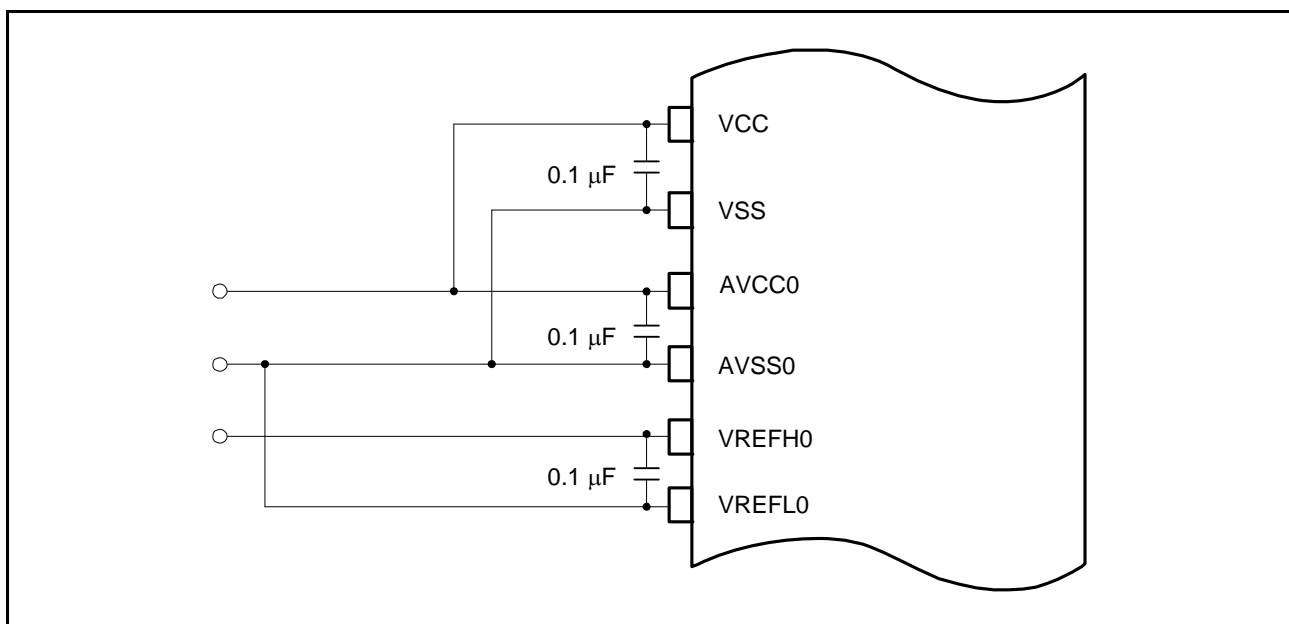


Figure 38.12 Connection Example of Power Supply Pins

### 38.5.9 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and the layout in which the digital circuit signal lines and analog circuit signal lines cross or are in close proximity to each other should be avoided as much as possible. Failure to do so may result in the incorrect operation of the analog circuitry due to inductance, adversely affecting the A/D conversion values.

In addition, digital circuitry must be isolated from the analog input signals (AN000 to AN020), analog reference voltages (VREFH0, VREFL0), analog power supply (AVCC0), and analog ground (AVSS0). AVSS0 should be connected at one point to a stable digital ground (VSS) on the board.

### 38.5.10 Notes on Noise Countermeasures

To prevent damage due to an abnormal voltage, such as an excessive surge at the analog input pins (AN000 to AN020), connect capacitors between the AVCC0 and AVSS0, and VREFH0 and VREFL0, and a protection circuit relative to the analog input pins (AN000 to AN020), as shown in Figure 38.13.

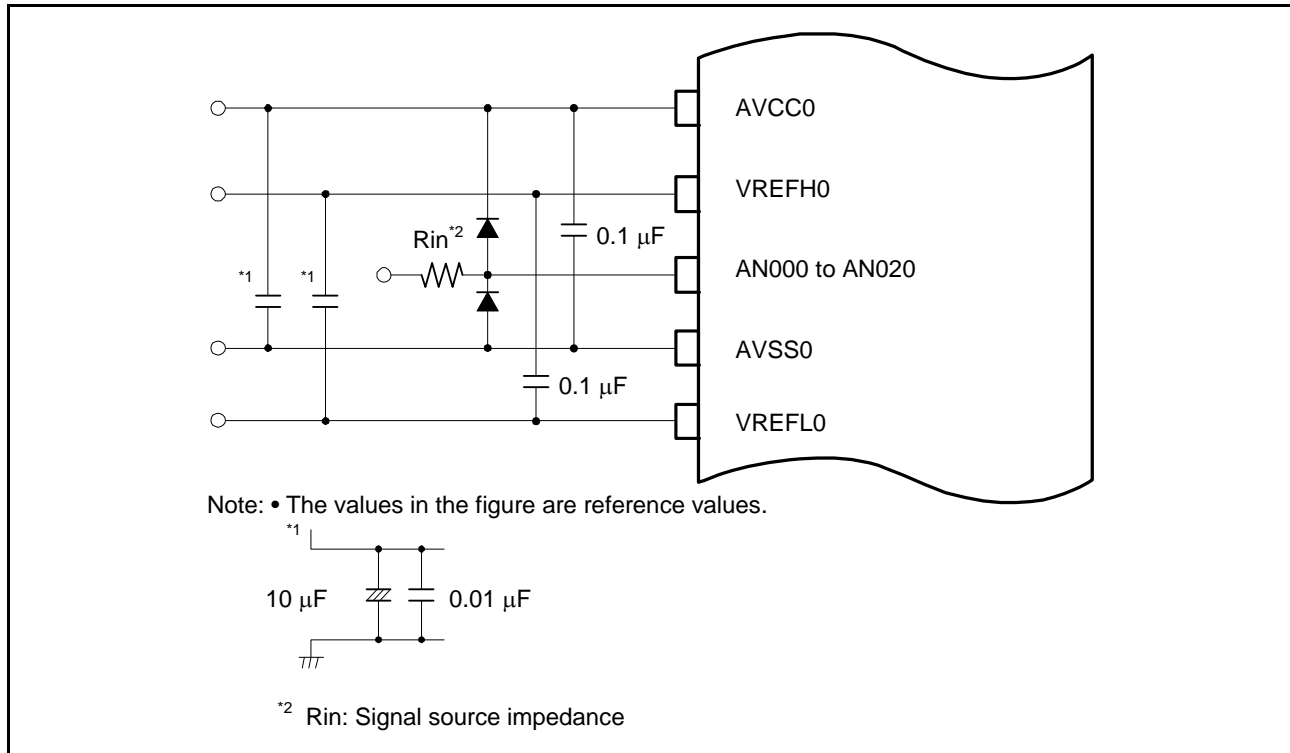


Figure 38.13 Example of Analog Input Pin Protection Circuit

### 38.5.11 Port Setting for Using 12-bit A/D Converter Inputs

When the 12-bit A/D converter is in use, use port 4 as a general input port. Moreover, ports 0, 9, and D are all recommended for use as general input ports. When using pins of ports 0, 9, and D as output pins, take a countermeasure such as repeating conversion several times and calculating the average of the converted values, excluding the maximum and minimum values.

### 38.5.12 Notes on Using the External Bus

When A/D conversion is done during access to the external bus, the accuracy may degrade.

In this case, repeat conversion several times and calculate by software the average of the converted values, excluding the maximum and minimum values.

## 39. 10-Bit A/D Converter (ADb)

### 39.1 Overview

The RX630 Group device includes a single unit of successive approximation type 10-bit A/D converter. The unit allows up to 8 analog input channels (AN0 to AN7) and one extended analog input channel (ANEX1) to be selected.

The 10-bit A/D converter has two operating modes: single channel mode which converts the analog input of the specified single channel or a single extended analog input for only once and scan mode which continuously converts the analog inputs of the specified channels up to eight or the extended analog input 1 (ANEX1).

Table 39.1 lists the specifications of the 10-bit A/D converter and the Table 39.2 lists the function overview of the 10-bit A/D converter. Figure 39.1 shows the block diagram of the 10-bit A/D converter.

**Table 39.1 Specifications of 10-Bit A/D Converter**

Item	Specifications
Number of units	One unit
Input channels	8 channels + one extended analog input
A/D conversion method	Successive approximation method
Resolution	10 bits
Conversion time	1.0 $\mu$ s per 1 channel (when operating peripheral module clock PCLK = 50 MHz)
A/D conversion clock	4 types: PCLK, PCLK/2, PCLK/4, PCLK/8
Operating modes	<ul style="list-style-type: none"> <li>• Single channel mode: A/D conversion is to be performed for only once on the analog input of the specified single channel or a single extended analog input.</li> <li>• Scan mode               <ul style="list-style-type: none"> <li>Continuous scan mode: A/D conversion is to be performed sequentially on the analog inputs of the specified channels up to eight or the extended analog input 1 (ANEX1).</li> <li>Single scan mode: A/D conversion is to be performed for one cycle on the analog inputs of the specified channels up to eight or the extended analog input 1 (ANEX1).</li> </ul> </li> </ul>
Conditions of A/D conversion start	<ul style="list-style-type: none"> <li>• Software trigger</li> <li>• 6 synchronous A/D conversion triggers from MTU, TPU, or TMR</li> <li>• 1 asynchronous A/D conversion trigger</li> </ul> A/D conversion can be started by the ADTRG# pin.
Function	<ul style="list-style-type: none"> <li>• Sample-and-hold function</li> <li>• Number of sampling states is adjustable.</li> <li>• Self-diagnostic functions of the 10-bit A/D converter</li> </ul>
Interrupt source	<ul style="list-style-type: none"> <li>• After the end of A/D conversion, an interrupt request (ADI0) is generated.</li> <li>• DMAC and DTC can be started by an ADI0 interrupt.</li> </ul>
Low-power consumption function	Module-stop state can be set.

**Table 39.2 Function Overview of the 10-Bit A/D Converter**

Item		Function/Internal Trigger Source		
Analog input channel		AN0 to AN7, ANEX1		
A/D conversion start conditions	Software	Software trigger	Enabled	
	Asynchronous trigger	ADTRG#	Enabled	
		Synchronous trigger (MTU, TMR, TPU) *1	TRG0AN_0	MTU0.TGRA and MTU0.TCNT
	TRGAN_0		MTU0.TGRA and MTU0.TCNT	Input-capture/compare-match
			MTU1.TGRA and MTU1.TCNT	
			MTU2.TGRA and MTU2.TCNT	
			MTU3.TGRA and MTU3.TCNT	
			MTU4.TGRA and MTU4.TCNT	
	MTU4.TCNT			TCNT underflow (trough) in complementary PWM mode
	TRGAN_1		TPU0.TGRA	
TPU1.TGRA				
TPU2.TGRA				
TPU3.TGRA				
TPU4.TGRA				
TRG4ABN_0	MTU4.TADCORA and MTU4.TCNT or MTU4.TADCORB and MTU4.TCNT	Compare-match using delayed A/D conversion start request function		
TRG4ABN_1	TPU0.TGRA	Input-capture/compare-match		
TMTRG0AN_0	TMR0.TCORA and TMR0.TCNT	Compare-match		
Interrupt		ADI0 interrupt		
Module-stop function setting*2		MSTPCRA.MSTPA23 bit		

Note 1. The suffix “\_0” or “\_1” appended to the name of a synchronous trigger signal indicates the number of the unit. For synchronous trigger output setting, see section 22.4.3, A/D Converter Activation, and section 26.6.2, A/D Converter Activation.

Note 2. For details, see section 11, Low Power Consumption.

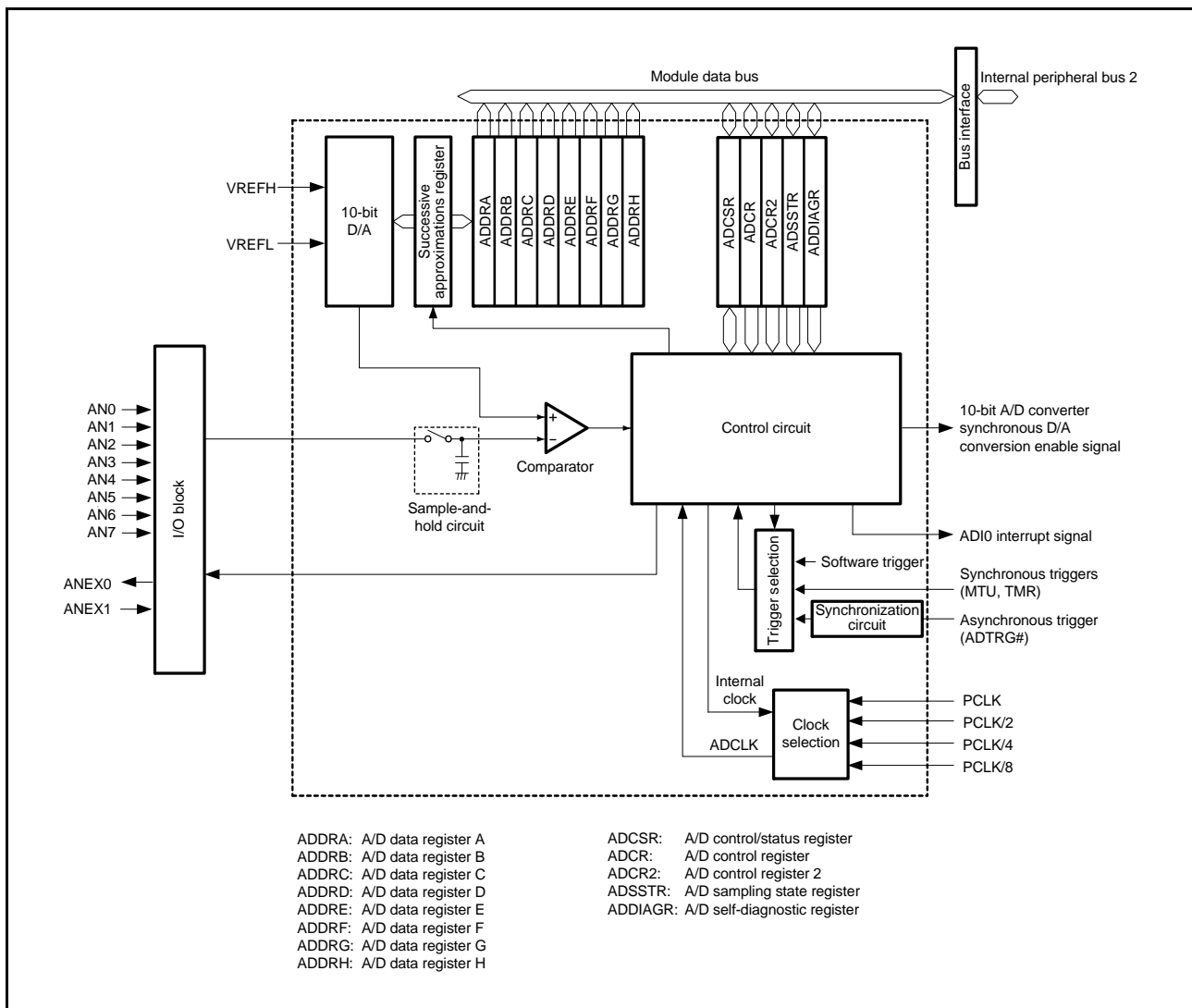


Figure 39.1 Block Diagram of 10-Bit A/D Converter

Table 39.3 indicates the input pins of the 10-bit A/D converter.

Table 39.3 Input Pins of 10-Bit A/D Converter

Pin Name	Input/Output	Function
AN0 to AN7	Input	Analog input pins
ANEX1	Input	Extended analog input pin
ANEX0	Output	Extended analog output pin
ADTRG#	Input	Asynchronous trigger input pin for starting A/D conversion
VREFH	Input	Reference voltage input pin for the 10-bit A/D converter and D/A converter. This is used as the analog power supply for each of the modules. Connect this pin to VCC if neither 10-bit A/D converter nor D/A converter are used.
VREFL	Input	Reference voltage input pin for the 10-bit A/D converter and D/A converter. This is used as the analog ground for each of the modules. Set this pin to the same potential as the VSS pin.

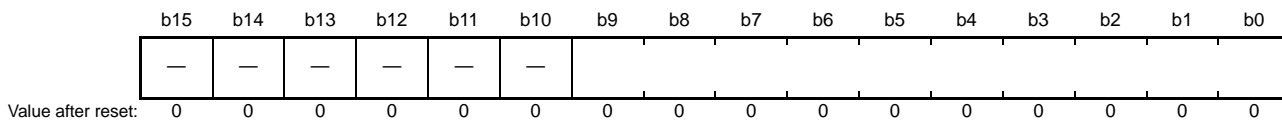


## 39.2 Register Descriptions

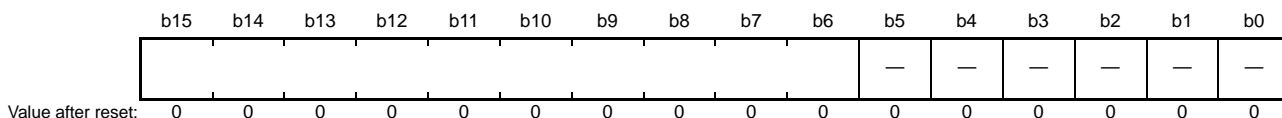
### 39.2.1 A/D Data Register y (ADDRy) (y = A to H)

Address(es): ADDRA 0008 9800h, ADDRb 0008 9802h, ADDRc 0008 9804h, ADDRd 0008 9806h  
 ADDRE 0008 9808h, ADDRf 0008 980Ah, ADDRg 0008 980Ch, ADDRh 0008 980Eh

- ADCSR2.DPSEL bit = 0 (Flush-right)



- ADCSR2.DPSEL bit = 1 (Flush-left)



ADDRy are 16-bit read-only registers, which store an A/D conversion result.

Table 39.4 lists the analog input channels and corresponding ADDRy registers, and Table 39.5 lists the extended analog input 1 (ANEX1) and corresponding ADDRy registers.

10-bit data can be relocated by setting the ADCR2.DPSEL bit.

Bits “—” are read as 0. The write value should be 0.

**Table 39.4 Analog Input Channels and Corresponding ADDRy Registers**

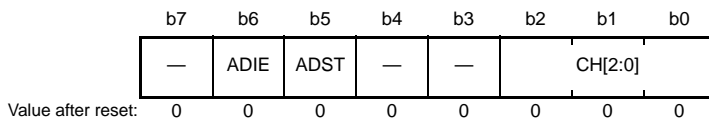
Analog Input Channel	ADDRy Register
AN0	ADDRA
AN1	ADDRB
AN2	ADDRc
AN3	ADDRd
AN4	ADDRE
AN5	ADDRf
AN6	ADDRg
AN7	ADDRh

**Table 39.5 Extended Analog Input 1 (ANEX1) and Corresponding ADDRy Registers**

Source Analog Input Channel	A/D Conversion Target	ADDRy Register
AN0	ANEX1	ADDRA
AN1		ADDRB
AN2		ADDRc
AN3		ADDRd
AN4		ADDRE
AN5		ADDRf
AN6		ADDRg
AN7		ADDRh

### 39.2.2 A/D Control/Status Register (ADCSR)

Address(es): 0008 9810h



Bit	Symbol	Bit Name	Description	R/W																				
b2 to b0	CH[2:0]	Channel Select*1	When ADCR.MODE[1:0] = 00b <table style="display: inline-table; vertical-align: top; margin-left: 20px;"> <tr><td style="padding-right: 5px;">b2</td><td style="padding-right: 5px;">b0</td><td>0 0 0: AN0</td></tr> <tr><td>0 0 1: AN1</td></tr> <tr><td>0 1 0: AN2</td></tr> <tr><td>0 1 1: AN3</td></tr> <tr><td>1 0 0: AN4</td></tr> <tr><td>1 0 1: AN5</td></tr> <tr><td>1 1 0: AN6</td></tr> <tr><td>1 1 1: AN7</td></tr> </table> When ADCR.MODE[1:0] = 10b or 11b <table style="display: inline-table; vertical-align: top; margin-left: 20px;"> <tr><td style="padding-right: 5px;">b2</td><td style="padding-right: 5px;">b0</td><td>0 0 0: AN0</td></tr> <tr><td>0 0 1: AN0, AN1</td></tr> <tr><td>0 1 0: AN0 to AN2</td></tr> <tr><td>0 1 1: AN0 to AN3</td></tr> <tr><td>1 0 0: AN0 to AN4</td></tr> <tr><td>1 0 1: AN0 to AN5</td></tr> <tr><td>1 1 0: AN0 to AN6</td></tr> <tr><td>1 1 1: AN0 to AN7</td></tr> </table>	b2	b0	0 0 0: AN0	0 0 1: AN1	0 1 0: AN2	0 1 1: AN3	1 0 0: AN4	1 0 1: AN5	1 1 0: AN6	1 1 1: AN7	b2	b0	0 0 0: AN0	0 0 1: AN0, AN1	0 1 0: AN0 to AN2	0 1 1: AN0 to AN3	1 0 0: AN0 to AN4	1 0 1: AN0 to AN5	1 1 0: AN0 to AN6	1 1 1: AN0 to AN7	R/W
b2	b0	0 0 0: AN0																						
0 0 1: AN1																								
0 1 0: AN2																								
0 1 1: AN3																								
1 0 0: AN4																								
1 0 1: AN5																								
1 1 0: AN6																								
1 1 1: AN7																								
b2	b0	0 0 0: AN0																						
0 0 1: AN0, AN1																								
0 1 0: AN0 to AN2																								
0 1 1: AN0 to AN3																								
1 0 0: AN0 to AN4																								
1 0 1: AN0 to AN5																								
1 1 0: AN0 to AN6																								
1 1 1: AN0 to AN7																								
b4, b3	—	Reserved	These bits are read as 0. The write value should be 0.	R/W																				
b5	ADST	A/D Start	0: Stops A/D conversion 1: Starts A/D conversion	R/W																				
b6	ADIE	A/D Interrupt Enable	0: ADI0 interrupt is disabled by completing A/D conversion 1: ADI0 interrupt is enabled by completing A/D conversion	R/W																				
b7	—	Reserved	The read value is undefined. The write value should be 0.	R/W																				

Note 1. The Pn.PDR.Bi bit for the analog input should be set to 0 (input port) and the Pn.PMR.Bi bit should be set to 0 (disabling the input buffer and fixing the input signal to the high level) (n = D, E, i = 0 to 7). For details, see section 20, I/O Ports and section 21, Multi-Function Pin Controller (MPC).

Set the CH[2:0] and ADIE bits while the ADST bit is 0 (A/D conversion stopped).

#### CH[2:0] Bits (Channel Select)

These bits select analog input channels that allow A/D conversion.

- Single channel mode (ADCR.MODE[1:0] bits = 00b)  
Select the single analog input channel that allows A/D conversion.
- Scan mode (ADCR.MODE[1:0] bits = 10b or 11b)  
Select up to 8 analog input channels that allow A/D conversion.

#### ADST Bit (A/D Start)

The ADST bit starts/stops A/D conversion.

Before setting the ADST bit to 1, complete the setting for A/D conversion clock and the operation mode.

[Setting conditions]

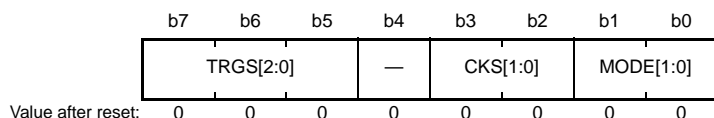
- When 1 is written by software
- Detection of the trigger selected by the TRGS[2:0] bits in ADCR

[Clearing conditions]

- When 0 is written by software
- The A/D conversion is completed in single channel mode
- The A/D conversion is completed on every selected channel in single scan mode

### 39.2.3 A/D Control Register (ADCR)

Address(es): 0008 9811h



Bit	Symbol	Bit Name	Description	R/W
b1, b0	MODE[1:0]	Operation Mode Select	b1 b0 0 0: Single channel mode 0 1: Setting prohibited 1 0: Continuous scan mode 1 1: Single scan mode	R/W
b3, b2	CKS[1:0]	Clock Select	b3 b2 0 0: PCLK/8 0 1: PCLK/4 1 0: PCLK/2 1 1: PCLK	R/W
b4	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b7 to b5	TRGS[2:0]	Trigger Select	b7 b5 0 0 0: Software trigger 0 0 1: Compare-match/input-capture A from MTU0 to MTU4 0 1 0: Compare-match from TMR0 0 1 1: Trigger from ADTRG# 1 0 0: Compare-match/input-capture A from MTU0 1 0 1: Compare-match/input-capture A from TPU0 to TPU4 1 1 0: Compare-match from MTU4 1 1 1: Compare-match input-capture A from TPU0	R/W

Set the MODE[1:0] and CKS[1:0] bits while the ADST bit in ADCSR is 0 (A/D conversion stopped).

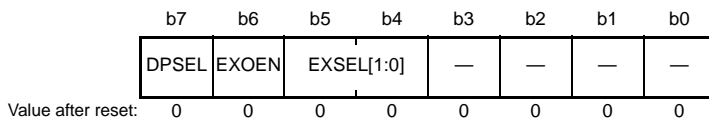
#### CKS[1:0] Bits (Clock Select)

These bits set the frequency of the A/D conversion clock (ADCLK) and thus select the A/D conversion time. Set the frequency of ADCLK above 4 MHz.

For details, see section 39.3.4, Input Sampling and A/D Conversion Time.

### 39.2.4 A/D Control Register 2 (ADCR2)

Address(es): 0008 9812h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b5, b4	EXSEL[1:0]	Extended Analog Input Select	b5 b4 0 0: Analog input channel (ANxx) 0 1: ANEX1 1 0: Setting prohibited 1 1: Setting prohibited	R/W
b6	EXOEN	Extended Analog Output Control	0: Output is disabled. 1: Output is enabled.	R/W
b7	DPSEL	ADDRy Format Select	0: A/D data is flush-right. 1: A/D data is flush-left.	R/W

Set the EXSEL[1:0], and EXOEN bits while the ADST bit in ADCSR is 0 (A/D conversion stopped).

#### EXSEL[1:0] Bits (Extended Analog Input Select)

These bits can select extended analog inputs ANEX1 other than the analog input channel (ANxx).

ANEX1: ANEX0 should be input through the external operational amplifier.

For details, see section 39.3.3, Extended Analog Input.

#### EXOEN Bit (Extended Analog Output Control)

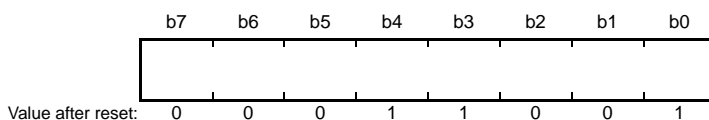
This bit can control extended analog output (ANEX0).

When output is enabled, an ANxx-multiplexed value is output to ANEX0.

When the EXSEL[1:0] bits are 00b, output should be disabled.

### 39.2.5 A/D Sampling State Register (ADSSTR)

Address(es): 0008 9813h



ADSSTR is an 8-bit readable/writable register that is used to set the sampling time for analog inputs.

Sampling time is adjustable when the signal source impedance of analog input is high and the sampling time is insufficient or the speed of peripheral module clock (PCLK) is low.

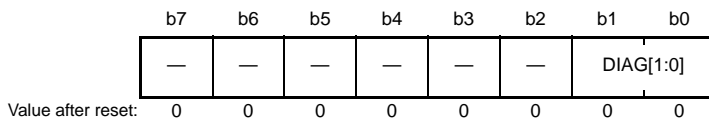
Set the value of 02h or larger.

Ensure to set this register while the A/D conversion is stopped (the ADST bit in ADCSR is 0).

For details, see section 39.3.4, Input Sampling and A/D Conversion Time.

### 39.2.6 A/D Self-Diagnostic Register (ADDIAGR)

Address(es): 0008 981Fh



Bit	Symbol	Bit Name	Description	R/W
b1, b0	DIAG[1:0]	Self-Diagnostic	b1 b0 0 0: Self-diagnostic function is off. 0 1: A/D conversion of $V_{ref} \times 0$ voltage value is enabled. 1 0: A/D conversion of $V_{ref} \times 1/2$ voltage value is enabled. 1 1: A/D conversion of $V_{ref} \times 1$ voltage value is enabled.	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Specify ADDIAGR while the ADST bit in ADCSR is 0 (A/D conversion stopped).

#### DIAG[1:0] Bits (Self-Diagnostic)

The self-diagnostic function is used to detect faults in the 10-bit A/D converter. A voltage value is selected to convert from three internally generated voltages:  $V_{ref} \times 0$ ,  $V_{ref} \times 1/2$ , and  $V_{ref} \times 1$ .

To perform self-diagnostic, select the voltage value with the DIAG[1:0] bits and start the A/D conversion with the setting below.

- Single channel mode (the ADCR.MODE[1:0] bits = 00b)
- Analog input channel AN0 is only enabled (the ADCSR.CH[2:0] bits = 000b, and ADCR2.EXSEL[1:0] bits = 00b)\*1
- A/D conversion is started by software. (the ADCR.TRGS[2:0] bits = 000b)

After A/D conversion ends, the conversion result is stored in the A/D data register A. Then, the value in ADDRA is read by software to determine whether the conversion result is within the normal range (normal) or not (error). The execution time of self-diagnosis is the same as that for A/D conversion of one channel.

Note 1. Set AN0 to perform self-diagnosis of 10-bit A/D converter. Though this setting is required to select the data register to store the conversion result, all-analog input (AN0 to AN7) has no effect.

### 39.3 Operation

The RX630 Group device includes a single unit of 10-bit A/D converter.

A/D conversion targets can be selected from among 8 analog input channels (AN0 to AN7) and one extended analog input (ANEX1), and then can be input.

The analog input channels and the extended analog inputs are selected using the EXSEL[1:0] bits in ADCR2. Each of the analog input channels (AN0 to AN7), the extended analog input 1 (ANEX1) are exclusively operated.

When the analog input channel (00b) is selected with the EXSEL[1:0] bits in ADCR2, the extended analog output (ANEX0) must be disabled by clearing the EXOEN bit in ADCR2 to 0. If output is not disabled, the time-shared analog value will be output from ANEX0. Further, due to internal wiring load of the LSI, the A/D conversion will be delayed.

The 10-bit A/D converter has two operating modes: single channel mode and scan mode.

In single channel mode, A/D conversion is performed only once on the analog input of the specified single channel or an extended analog input.

In scan mode, A/D conversion is performed sequentially on the analog inputs of the specified channels or the extended analog input 1 (ANEX1).

Two types of scan mode are provided: continuous scan mode where A/D conversion is repeatedly performed and one-cycle scan mode where one cycle of A/D conversion is performed on the specified channels.

### 39.3.1 Single Channel Mode

In single channel mode, A/D conversion is performed only once on the analog input of the specified single channel or the single extended analog channel as below.

- (1) A/D conversion for the selected channel is started when the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, a synchronous trigger input, or an asynchronous trigger input.
- (2) When A/D conversion is completed, the A/D conversion result is stored into the corresponding A/D data register (ADDRy) of the channel.
- (3) When A/D conversion is completed, if the ADIE bit in ADCSR is set to 1 (ADI0 interrupt enable by completing A/D conversion), an ADI0 interrupt request is generated.
- (4) The ADST bit is held at 1 during A/D conversion, and is automatically cleared to 0 when A/D conversion ends. Then the 10-bit A/D converter enters a wait state.
- (5) If the ADST bit is cleared to 0 (A/D conversion stop) during A/D conversion, A/D conversion stops and the 10-bit A/D converter enters a wait state.

Figure 39.2 shows an example of operation when AN1 is selected as an analog input.

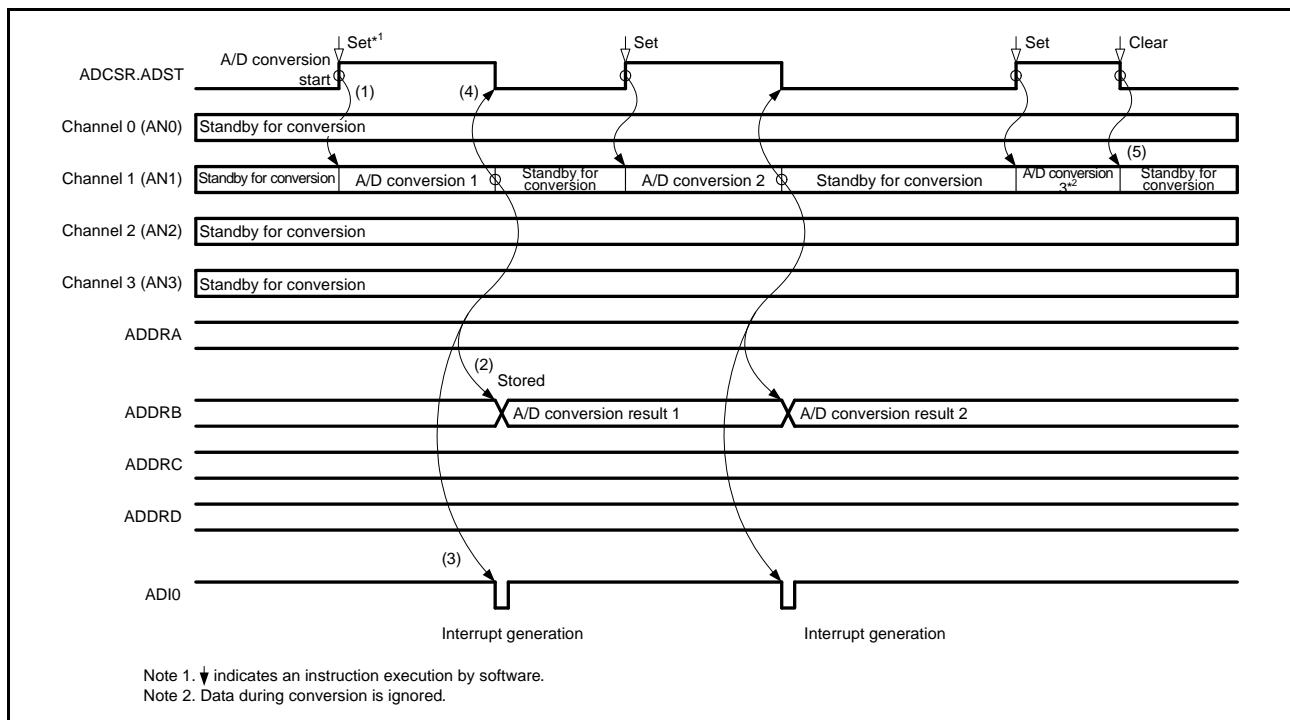


Figure 39.2 Example of 10-Bit A/D Converter Operation (Single Channel Mode)

### 39.3.2 Scan Mode

A case when the analog input channel is selected (the EXSEL[1:0] bits in ADCR2 are set to 00b) is described below. For the case with the extended analog input selected, see section 39.3.3, Extended Analog Input.

In scan mode, A/D conversion is performed sequentially on the analog inputs of the channels up to eight that are specified with the CH[2:0] bits in ADCSR. There are two scan modes: continuous scan mode where A/D conversion is repeatedly performed and one-cycle scan mode where A/D conversion is performed on the specified channels for one cycle.

#### 39.3.2.1 Continuous Scan Mode

In continuous scan mode, A/D conversion is performed sequentially on the analog inputs of the specified channels as below.

- (1) When the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, a synchronous trigger input, or an asynchronous trigger input, A/D conversion starts from the first channel (AN0) in the specified channel group.
- (2) When A/D conversion for each channel is completed, the A/D conversion result is stored into the corresponding A/D data registers (ADDRy).
- (3) When A/D conversion of all the selected channels is completed, if the ADIE bit in ADCSR is set to 1 (ADIO interrupt enabled by completion of A/D conversion), an ADIO interrupt request is generated. The 10-bit A/D converter starts A/D conversion from the first channel (AN0).
- (4) The ADST bit is not cleared automatically, and steps 2 and 3 are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0 (A/D conversion stop), A/D conversion stops and the 10-bit A/D converter enters a wait state.
- (5) After the ADST bit in ADCSR is set to 1, A/D converter starts A/D conversion from the first channel (AN0) again.

Figure 39.3 shows an example of A/D conversion when three channels (AN0 to AN2) are selected for analog input.

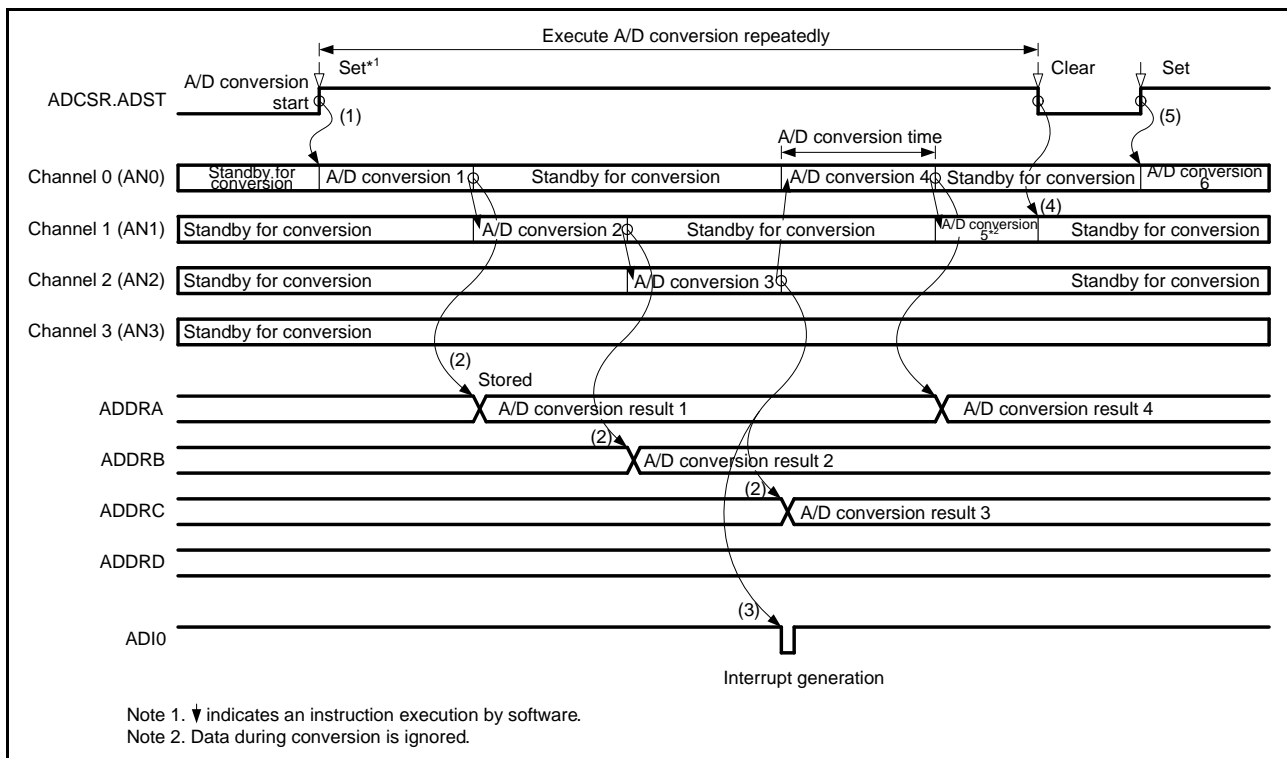


Figure 39.3 Example of 10-Bit A/D Converter Operation (Continuous Scan Mode)



### 39.3.2.2 Single Scan Mode

In single scan mode, A/D conversion is performed for one cycle on the analog inputs of the specified channels as below.

- (1) When the ADST bit in ADCSR is set to 1 (A/D conversion start) by software, a synchronous trigger input, or an asynchronous trigger input, A/D conversion starts from the first channel (AN0) in the specified channel group.
- (2) When A/D conversion for each channel is completed, the A/D conversion result is stored into the corresponding A/D data registers (ADDRy).
- (3) When A/D conversion is completed for all the selected channels, an ADI0 interrupt request is generated if the ADIE bit in ADCSR is set to 1 (ADI0 interrupt enabled by completion of A/D conversion).
- (4) The ADST bit is held at 1 during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed, which enters the 10-bit A/D converter into a wait state.

Figure 39.4 shows an example of A/D conversion when three channels (AN0 to AN2) are selected for analog input.

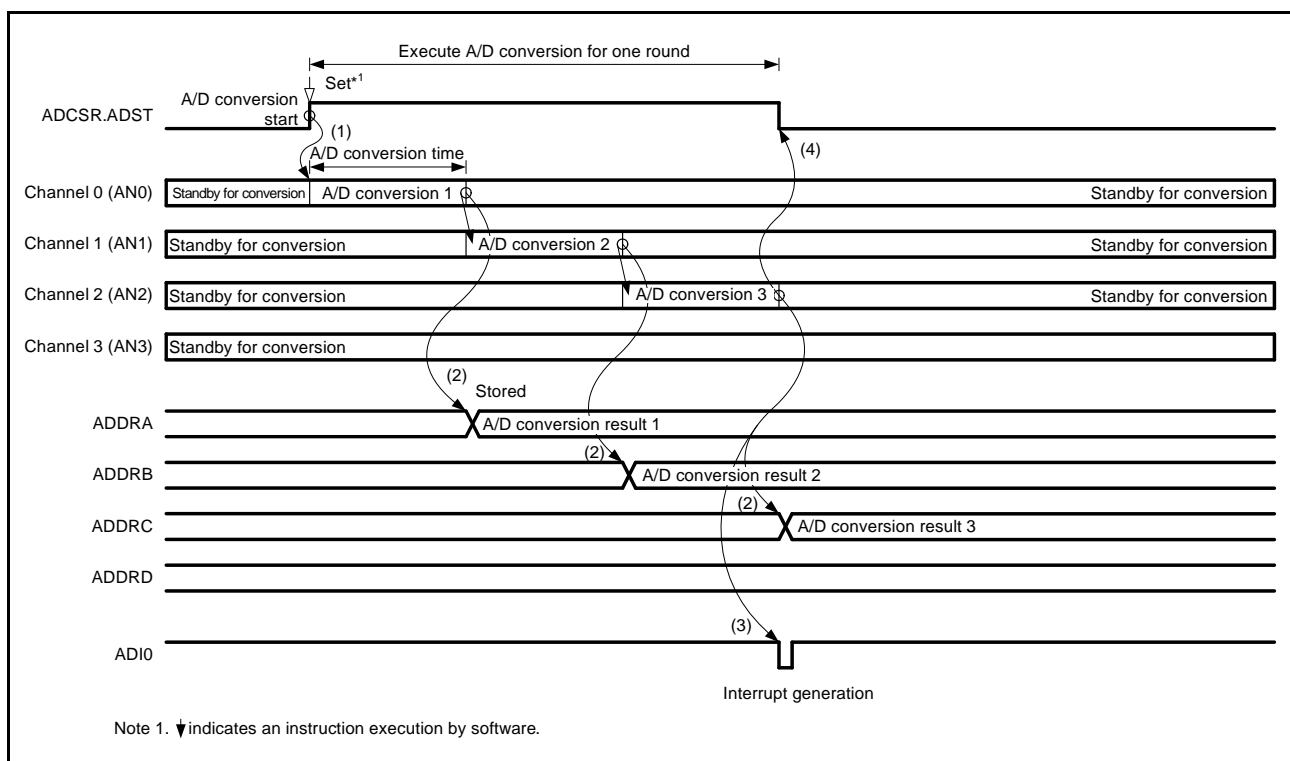


Figure 39.4 Example of 10-Bit A/D Converter Operation (Single Scan Mode)

### 39.3.3 Extended Analog Input

There is one extended analog input ANEX1. The usage of the extended analog input is determined as given below. Only the following usage method should be used.

The ANEX1 is used when an external operational amplifier is connected to the LSI to perform A/D conversion for the multiple analog values.

Figure 39.5 shows a configuration example of the extended analog input circuit.

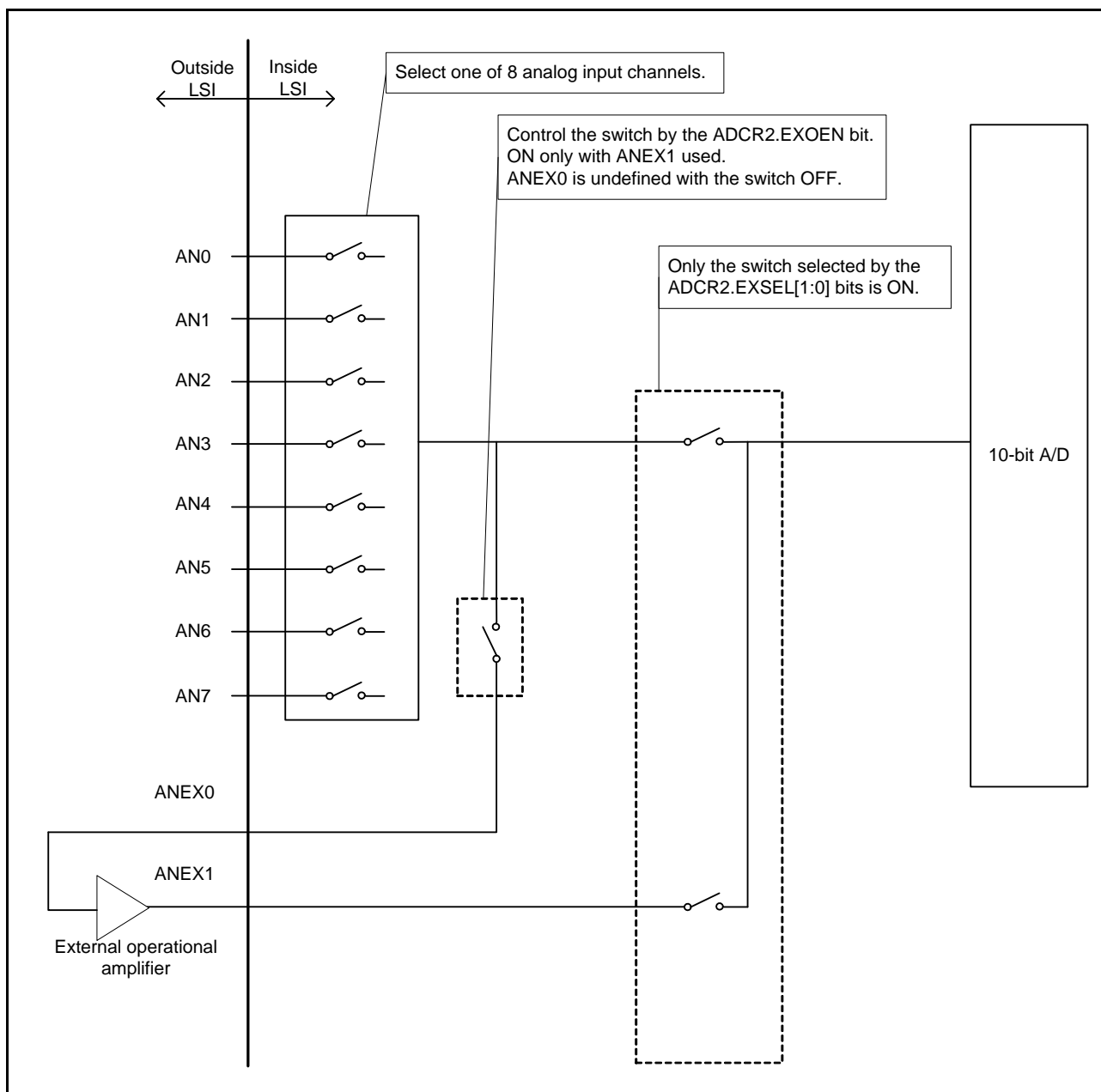


Figure 39.5 Configuration Example of Extended Analog Input Circuit

### 39.3.3.1 Usage of ANEX1

To perform A/D conversion of the multiple analog values via the operational amplifier, the analog values are input using the analog input channels of the LSI (AN0 to AN7). Then the time-shared analog values are taken from the extended analog output (ANEX0), and the operational amplifier is connected between ANEX0 and ANEX1.

To select ANEX1, set the ADCR2.EXSEL[1:0] bits to 01b. To enable ANEX0 output, set the ADCR2.EXOEN bit to 1. In addition, select single channel mode or scan mode.

Figure 39.6 shows the operation when three channels (AN0, AN1, and AN2) and single scan mode are selected.

- (1) When the ADCSR.ADST bit is set to 1 (A/D conversion start) by software, a synchronous trigger input, or an asynchronous trigger input, A/D conversion starts from the first channel (AN0) in the specified channel group.
- (2) When A/D conversion is completed, the A/D conversion result is stored in an A/D data register (ADDRy).
- (3) When A/D conversion of all the selected channels is completed, if the ADCSR.ADIE bit is 1 (ADI0 interrupt enabled by completion of A/D conversion), an ADI0 interrupt request is generated.
- (4) The ADCSR.ADST bit is held at 1 during A/D conversion, and is automatically cleared to 0 when A/D conversion of all the selected channels is completed, and the 10-bit A/D converter enters a wait state.

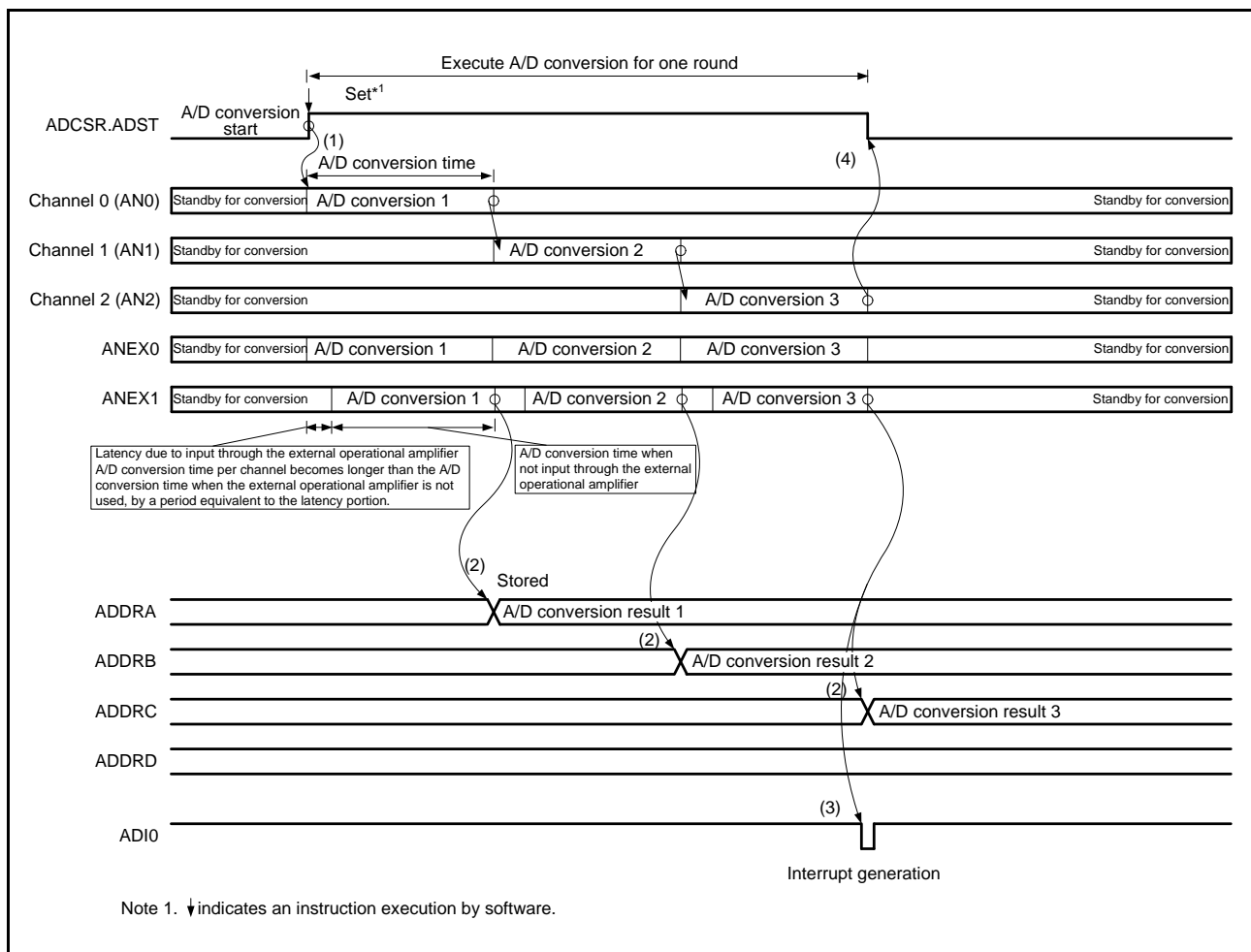


Figure 39.6 ANEX1 Input Operation Example (Single Scan Mode)

When the extended analog input is selected, the individual A/D conversion time becomes longer due to the latency of the operational amplifier, as compared to the A/D conversion that is directly performed for the analog input channels.

### 39.3.4 Input Sampling and A/D Conversion Time

The 10-bit A/D converter performs sampling of analog input when the A/D conversion start delay time (tD) has elapsed after A/D conversion start conditions are generated by software, a synchronous trigger, or an asynchronous trigger. After sampling, A/D conversion is started.

Figure 39.7 shows the A/D conversion timing.

As shown in the figure, the A/D conversion time (tCONV) following the generation of A/D conversion start conditions is a total of the A/D conversion start delay time (tD), the input sampling time (tSPL), and the successive conversion time (tSAM). The subsequent A/D conversion time (tCONV) is addition of tSPL and tSAM.

The sampling time (tSPL) is the time for charging the input capacitance of the 10-bit A/D converter's sample-and-hold circuit. If the impedance of the analog input signal source is too high to ensure sufficient sampling time, or the peripheral module clock (PCLK) is running at low speed, the sampling time can be adjusted by using the ADSSTR.

The successive conversion time (tSAM) is fixed at 25 cycles of ADCLK.

The samples of ADSSTR settings are listed in Table 39.6 and the A/D conversion times in Table 39.7.

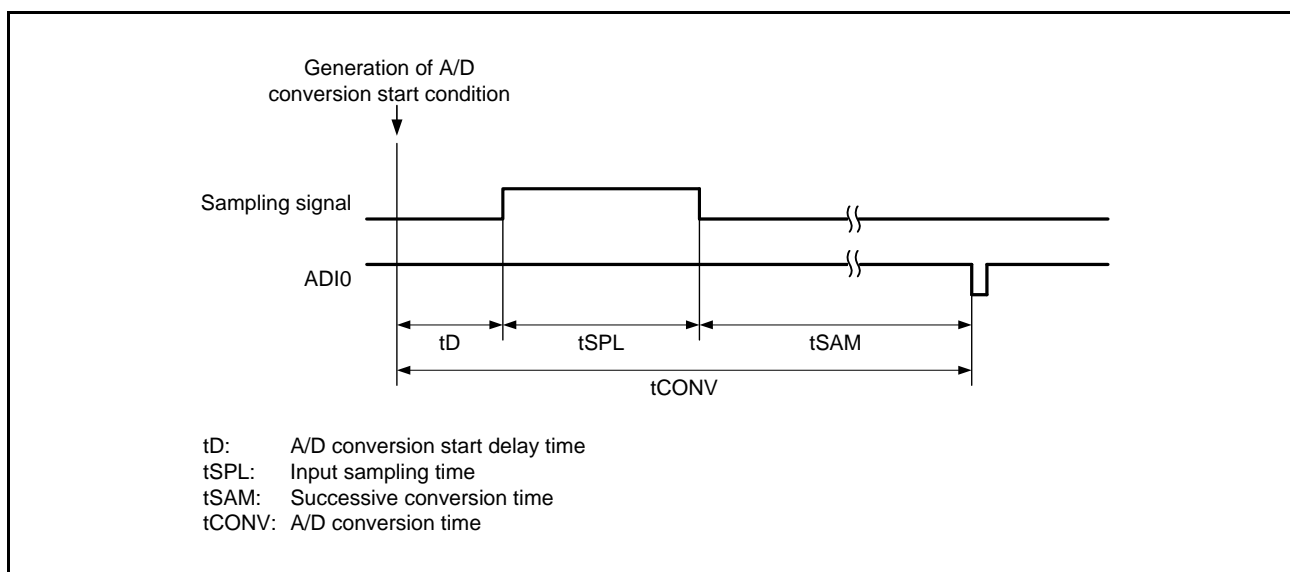


Figure 39.7 A/D Conversion Timing

Table 39.6 Example of ADSSTR Setting

Setting Conditions	Setting Range	Sampling Time*1
Standard (initial value)	19h	0.5 μs (When PCLK = ADCLK = 50 MHz)
Analog input signal source impedance is higher, resulting in insufficient sampling time	1Ah to FFh	Ex: FFh 5.1 μs (When PCLK = ADCLK = 50 MHz)
With ADCLK below 50 MHz, sampling time needs to be less than the initial value.	02h to 18h	Ex: 14h 0.5 μs (When PCLK = ADCLK = 40 MHz)

Note 1. Set the sampling time ≥ 0.5 μs. Sampling time is shown by:

$$\text{Sampling time } (\mu\text{s}) = \frac{\text{Setting value of ADSSTR}}{\text{ADCLK (MHz)}}$$

**Table 39.7 A/D Conversion Time**

Item	Symbol	Formula	
		Min.	Max.
A/D conversion start delay time (1)	tD	$\frac{3}{\text{PCLK (MHz)}}$	$\frac{1}{\text{ADCLK (MHz)}} + \frac{4}{\text{PCLK (MHz)}}$
Input sampling time (2)	tSPL	$\frac{\text{Setting value of ADSSTR}}{\text{ADCLK (MHz)}}$	
Successive conversion time (3)	tSAM	$\frac{25}{\text{ADCLK (MHz)}}$	
A/D conversion time*1	tCONV	(1) + (2) + (3)	
A/D conversion time*2	tCONV	(2) + (3)	

Note 1. A/D conversion time in single channel mode and scan mode (first round)

Note 2. A/D conversion time in scan mode (after the second round)

The examples of the calculation of A/D conversion times are listed below.

When PCLK = ADCLK = 50 MHz, ADSSTR = 19h, and the conversion is the second round in scan mode,

$$\begin{aligned}
 \text{A/D conversion time (tCONV)} &= \text{ADSSTR/ADCLK} + 25/\text{ADCLK} \\
 &= 25/50 \text{ MHz} + 25/50 \text{ MHz} \\
 &= 0.5 \mu\text{s} + 0.5 \mu\text{s} \\
 &= 1.0 \mu\text{s}
 \end{aligned}$$

When PCLK = ADCLK = 40 MHz, ADSSTR = 14h, and conversion is the first (min.) round in scan mode,

$$\begin{aligned}
 \text{A/D conversion time (tCONV)} &= 3/\text{PCLK} + \text{ADSSTR/ADCLK} + 25/\text{ADCLK} \\
 &= 3/40 \text{ MHz} + 20/40 \text{ MHz} + 25/40 \text{ MHz} \\
 &= 0.075 \mu\text{s} + 0.5 \mu\text{s} + 0.625 \mu\text{s} \\
 &= 1.2 \mu\text{s}
 \end{aligned}$$

### 39.3.5 Starting 10-Bit A/D Conversion with Asynchronous Trigger

A/D conversion can be started by the asynchronous trigger (ADTRG# pin) input.

When the ADCR.TRGS[2:0] bits are set to 011b (asynchronous trigger (ADTRG# pin)), the ADCSR.ADST bit is set to 1 (A/D conversion start) on a falling edge of the asynchronous trigger (ADTRG# pin) to start A/D conversion. This timing is shown in Figure 39.8.

Note that when the asynchronous trigger input is already low in using the asynchronous trigger, generation of a falling edge of the internal signal may cause A/D conversion to be activated.

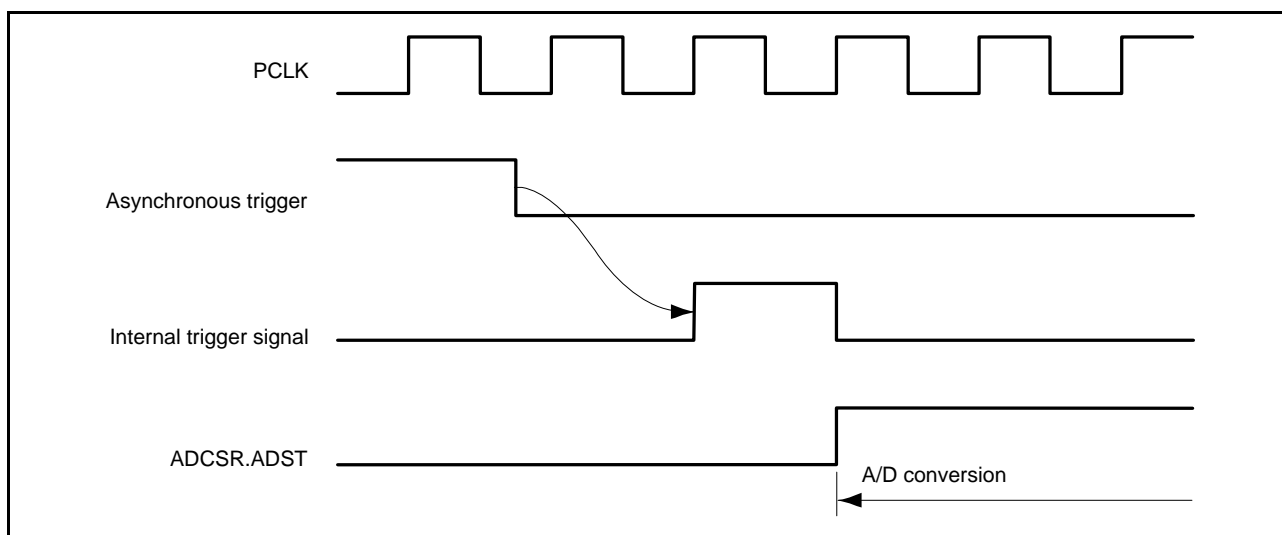


Figure 39.8 Asynchronous Trigger Input Timing

### 39.3.6 Using Synchronous Triggers from Peripheral Modules to Start A/D Conversion

A/D conversion can be started by synchronous triggers from the MTU, TPU, and TMR modules. When using a synchronous trigger to start A/D conversion, set the ADCR.TRGS[2:0] bits to select the required start source.

#### 39.3.6.1 Starting A/D Conversion Using MTU TRG0AN\_0

Upon the TRGA input capture/compare match of the MTU0, the trigger signal TRG0AN\_0 can be generated and A/D conversion can be started.

Figure 39.9 shows the connection of MTU TRG0AN\_0 output to the A/D converter.

To start A/D conversion upon the TRGA input capture/compare match of the MTU0, set ADCR.TRGS[2:0] in the A/D converter to 100b (trigger signal: TRG0AN\_0).

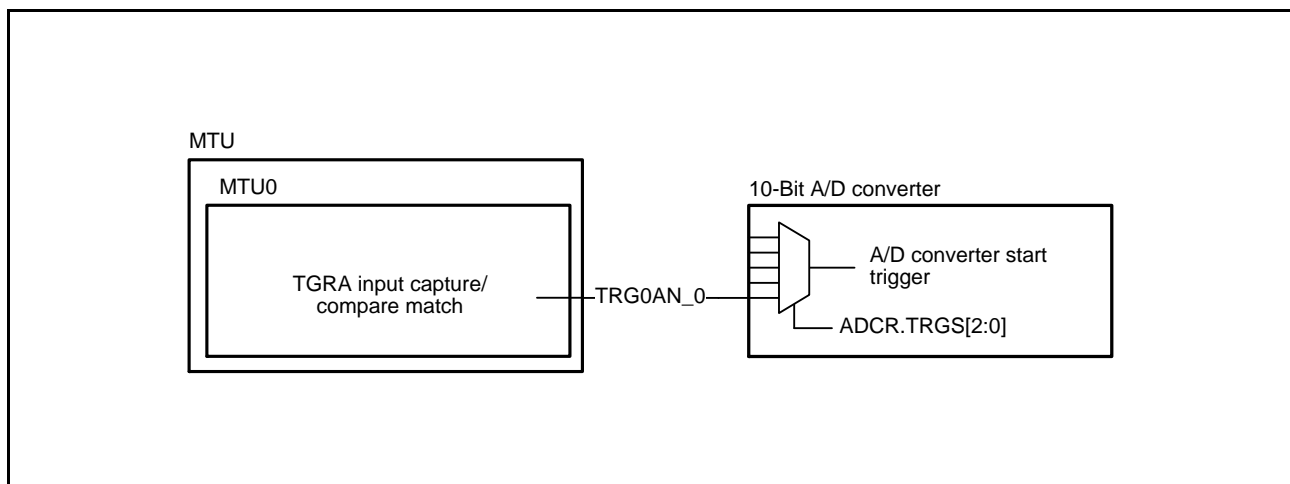


Figure 39.9 Connection of MTU TRG0AN\_0 Output to A/D Converter

#### 39.3.6.2 Starting A/D Conversion Using MTU TRGAN\_0 and TPU TRGAN\_1

Upon the TRGA input capture/compare match of the MTU0 to MTU4 and the TCNT underflow (trough) of the MTU4 in complementary PWM mode, the trigger signal TRGAN\_0 can be generated and A/D conversion can be started.

Similarly, upon the TRGA input capture/compare match of the TPU0 to TPU4 in the TPU (unit 0), the trigger signal TRGAN\_1 can be generated and A/D conversion can be started.

Figure 39.10 shows the connection of MTU TRGAN\_0 output and TPU (unit 0) TRGAN\_1 output to the A/D converter.

To start A/D conversion upon the TRGA input capture/compare match of the MTU0 to MTU4 and the TCNT underflow (though) of the MTU4 in complementary PWM mode, set the ADCR.TRGS[2:0] bits to 001b (selecting: TRGAN\_0 as the trigger signal), set the MTUn.TIER.TTGE bit ( $n = 0$  to 4), and the MTU4.TIER.TTGE2 bit to 1.

To start A/D conversion upon input capture to or compare match with TGRA in TPU0 to TPU4, set the ADCR.TRGS[2:0] bits to 101b (selecting TRGAN\_1 as the trigger signal), and the corresponding TPUn.TIER.TTGE bit ( $n = 0$  to 4) to 1.

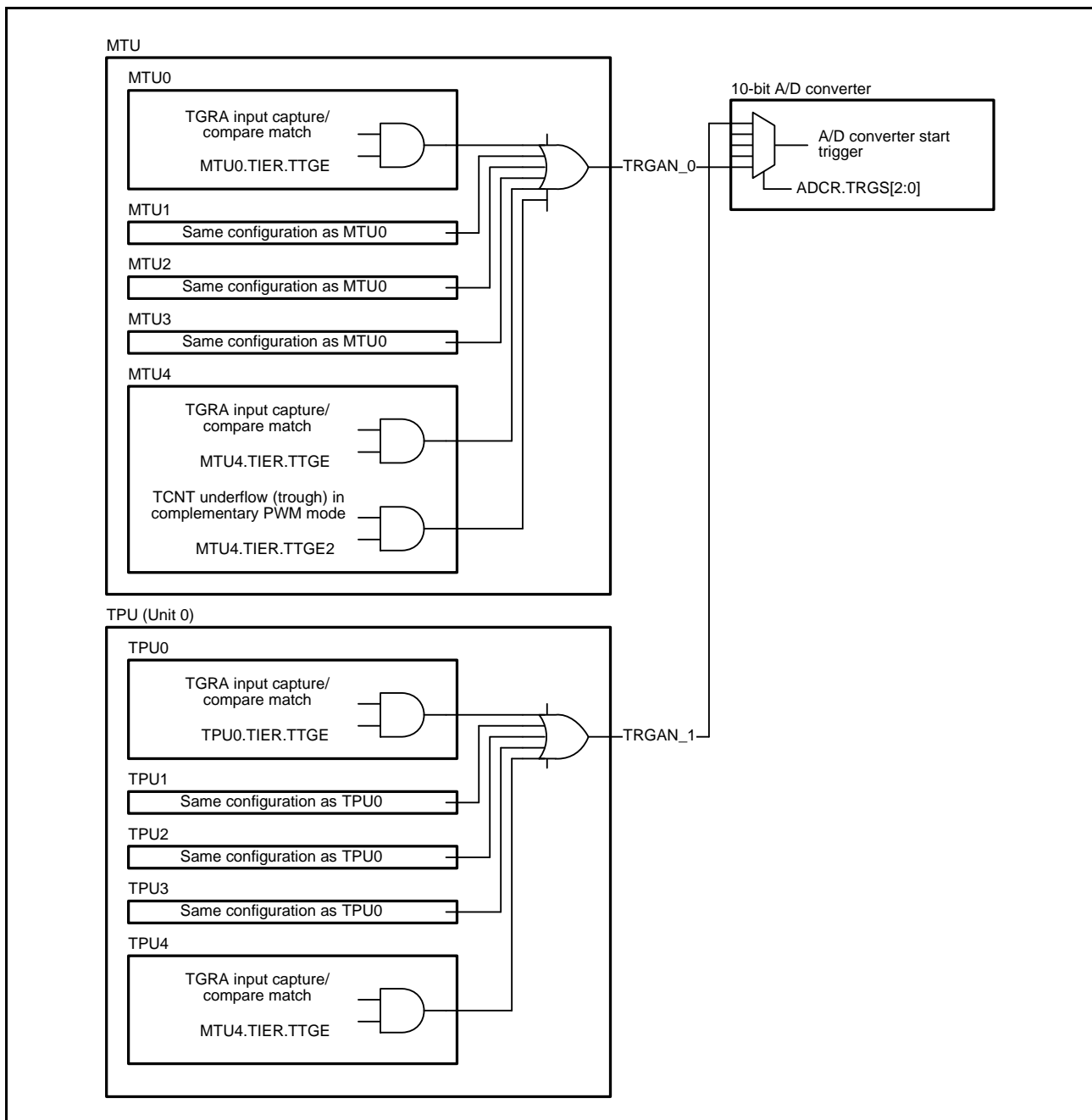


Figure 39.10 Connection of MTU TRGAN\_0 Output and TPU (unit 0) TRGAN\_1 Output to A/D Converter



### 39.3.7 Starting A/D Conversion Using MTU TRG4ABN\_0 and TPU TRG4ABN\_1

Upon the compare match of the MTU4 generated using the delayed A/D conversion start request function, the trigger signal TRG4ABN\_0 can be generated and A/D conversion can be started. Similarly, upon the TRGA input capture/compare match of the TPU0 in the TPU (unit 0), the trigger signal TRG4ABN\_1 can be generated and A/D conversion can be started with TRG4ABN\_1.

Figure 39.11 shows the connection of MTU TRG4ABN\_0 output and TPU (unit 0) TRG4ABN\_1 output to the A/D converter.

To use the delayed A/D conversion start request function of the MTU4 to start A/D conversion upon the compare match between the TADCORA and TCNT countup, set ADCR.TRGS[2:0] in the A/D converter to 110b (trigger signal: TRG4ABN\_0), set the cycle to MTU4.TADCOBRA/MTU4.TADCOBRB and MTU4.TADCORA/MTU4.TADCORB, and set MTU4.TADCR.UT4AE to 1.

To start A/D conversion upon input capture to or compare match with TGRA in TPU0, set the ADCR.TRGS[2:0] bit to 111b (selecting TRG8N as source and TRG4ABN\_1 as the trigger signal) and the TPU0.TIER.TTGE bit to 1.

For details on the delayed A/D conversion start request function, see section 22.3.9, A/D Converter Start Request Delaying Function.

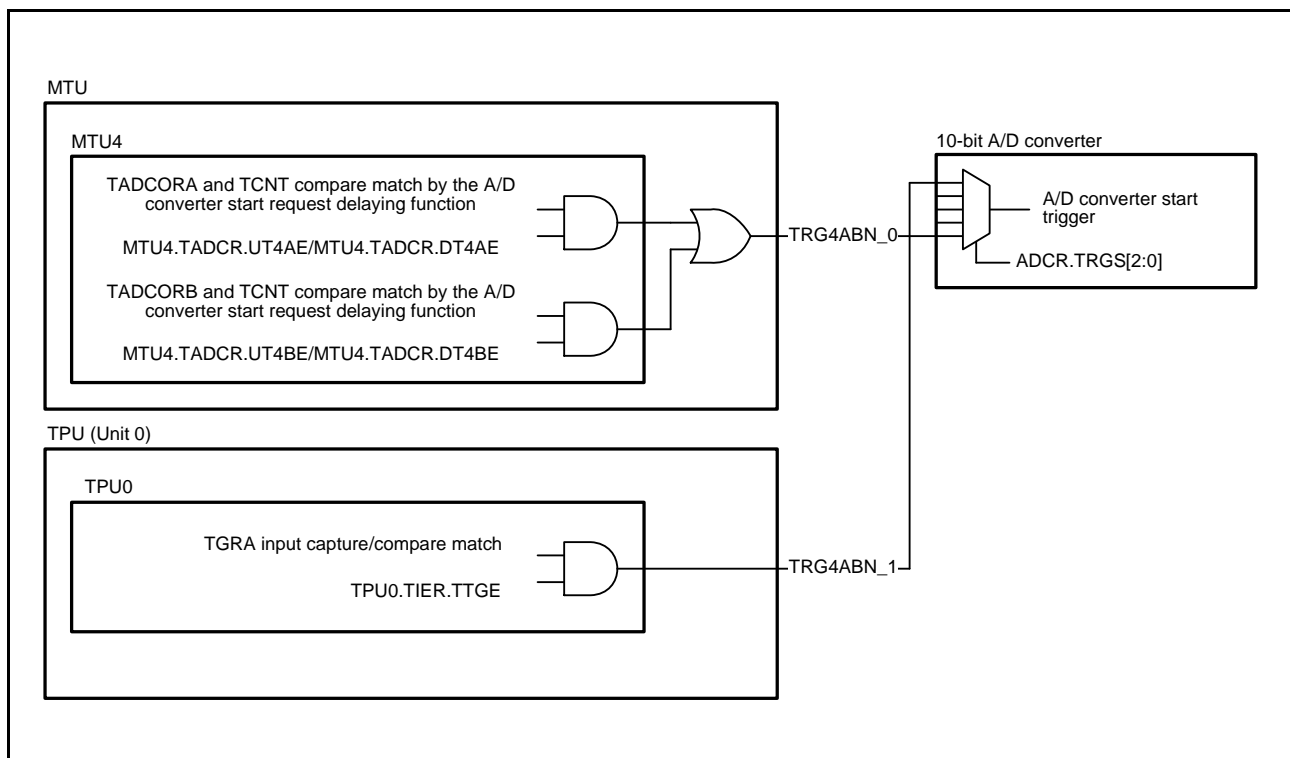


Figure 39.11 Connection of MTU TRG4ABN\_0 Output and TPU (unit 0) TRG4ABN\_1 Output to A/D Converter

### 39.3.8 Starting A/D Conversion Using TMR TMTRG0AN\_0

Upon the TCORA compare match (compare match A) of the TMR0 in the TMR (unit 0), A/D conversion can be started. Figure 39.12 shows the connection of TMR (unit 0) TMTRG0AN\_0 output to the A/D converter.

To start A/D conversion upon the TCORA compare match (compare match A) of the TMR0 in the TMR (unit 0), set ADCR.TRGS[2:0] in the A/D converter to 010b (trigger signal: TMTRG0AN\_0), and set TMR0.TCSR.ADTE to 1.

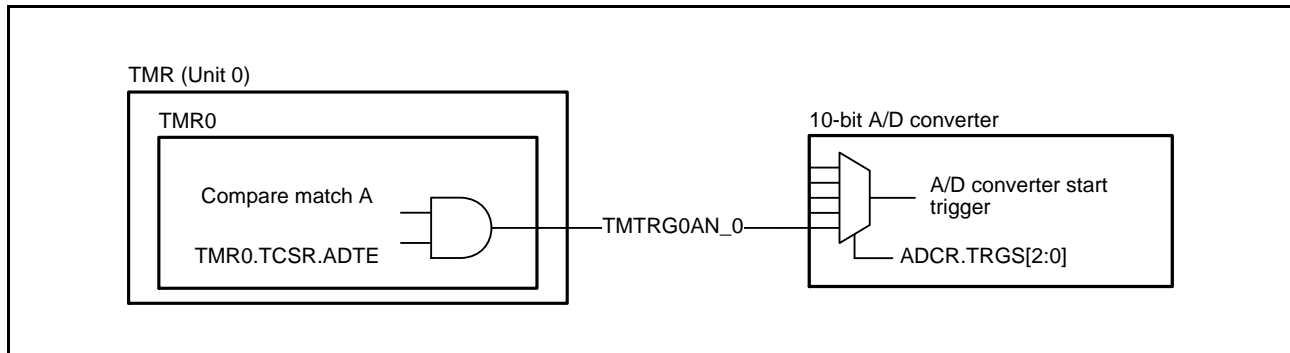


Figure 39.12 Connection of TMR (Unit 0) TMTRG0AN\_0 Output to A/D Converter

### 39.3.9 10-Bit A/D Converter Synchronous D/A Conversion Enable Signal

The 10-bit A/D converter synchronous D/A conversion enable signal output is high when the ADCSR.ADST bit in the 10-bit A/D converter is 0 (waiting for conversion). When the ADCSR.ADST bit is 1 (A/D conversion in progress), a high pulse of one PCLK cycle is output upon completion of A/D conversion. Figure 39.13 shows the output timing of the 10-bit A/D converter synchronous D/A conversion enable signal.

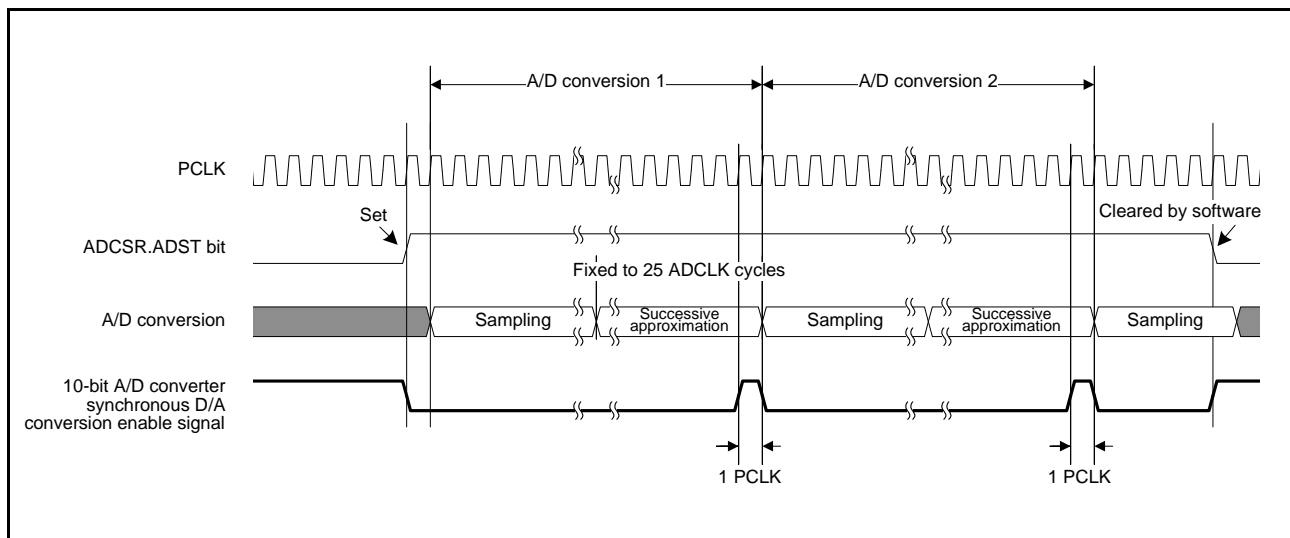


Figure 39.13 Output Timing of the 10-Bit A/D Converter Synchronous D/A Conversion Enable Signal

### 39.4 Interrupt Sources

The 10-bit A/D converter generates an A/D conversion end interrupt (ADIO) when A/D conversion is terminated with the ADIE bit in ADCSR set to 1 (enabled ADIO interrupt after A/D conversion completion).

In addition, the DTC or DMAC can be started up when an ADIO interrupt is generated. Using an ADIO interrupt to allow the DTC or DMAC to read the converted data enables continuous conversion without burden on software.

**Table 39.8 10-Bit A/D Converter Interrupt Source**

Name	Interrupt Source	DTC Activation	DMAC Activation
ADIO	A/D conversion end	Possible	Possible

### 39.5 A/D Conversion Accuracy Definitions

The RX630 Group's A/D conversion accuracy is defined as below:

- Resolution  
The number of A/D converter digital output codes
- Quantization error  
The deviation inherent in the A/D converter, given by 1/2 LSB (see Figure 39.14)
- Offset error  
The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value 000000000b (000h) to 000000001b (001h) (see Figure 39.15)
- Full-scale error  
The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from 111111110b (3FEh) to 111111111b (3FFh) (see Figure 39.15)
- Nonlinearity error  
The error with respect to the ideal A/D conversion characteristic between zero voltage and the full-scale voltage, excluding offset error, full-scale error, and quantization error (see Figure 39.15).
- Absolute accuracy  
The deviation between the digital value and analog input value, including offset error, full-scale error, quantization error, and nonlinearity error.

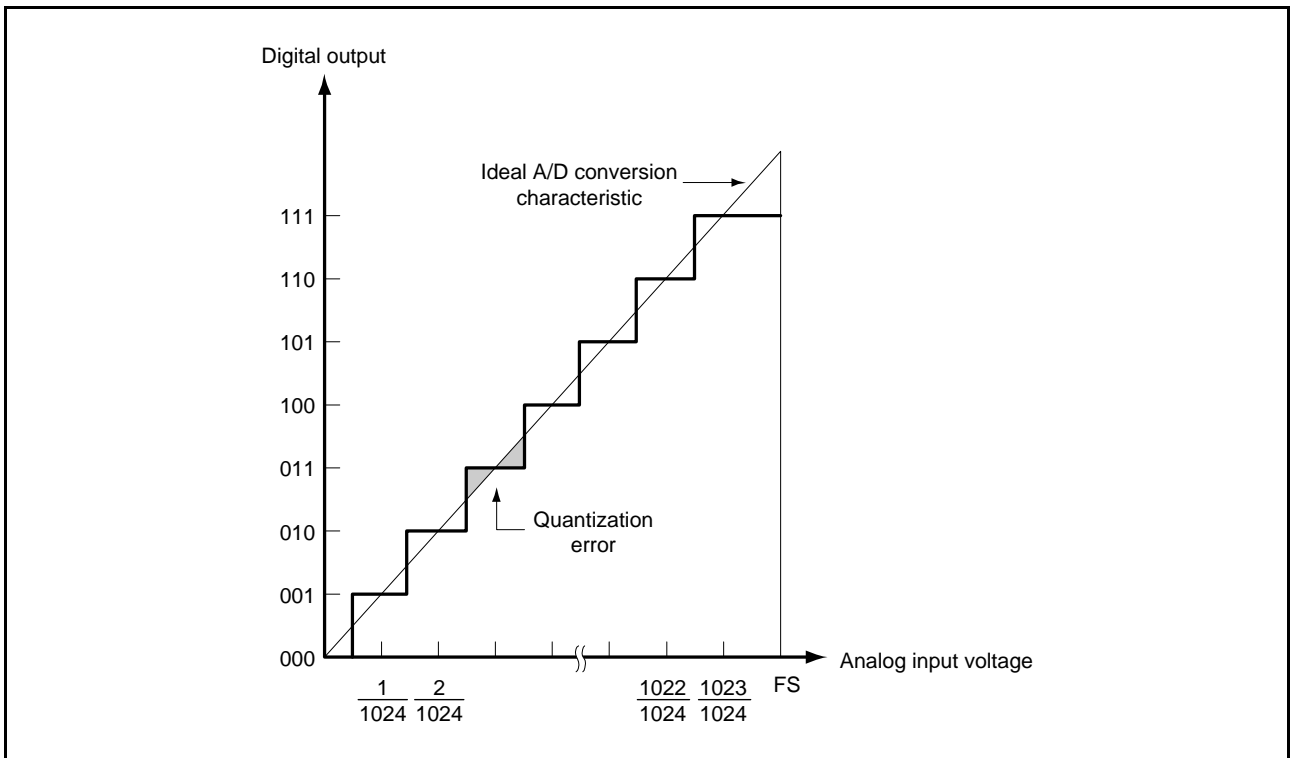


Figure 39.14 A/D Conversion Accuracy Definitions (1)

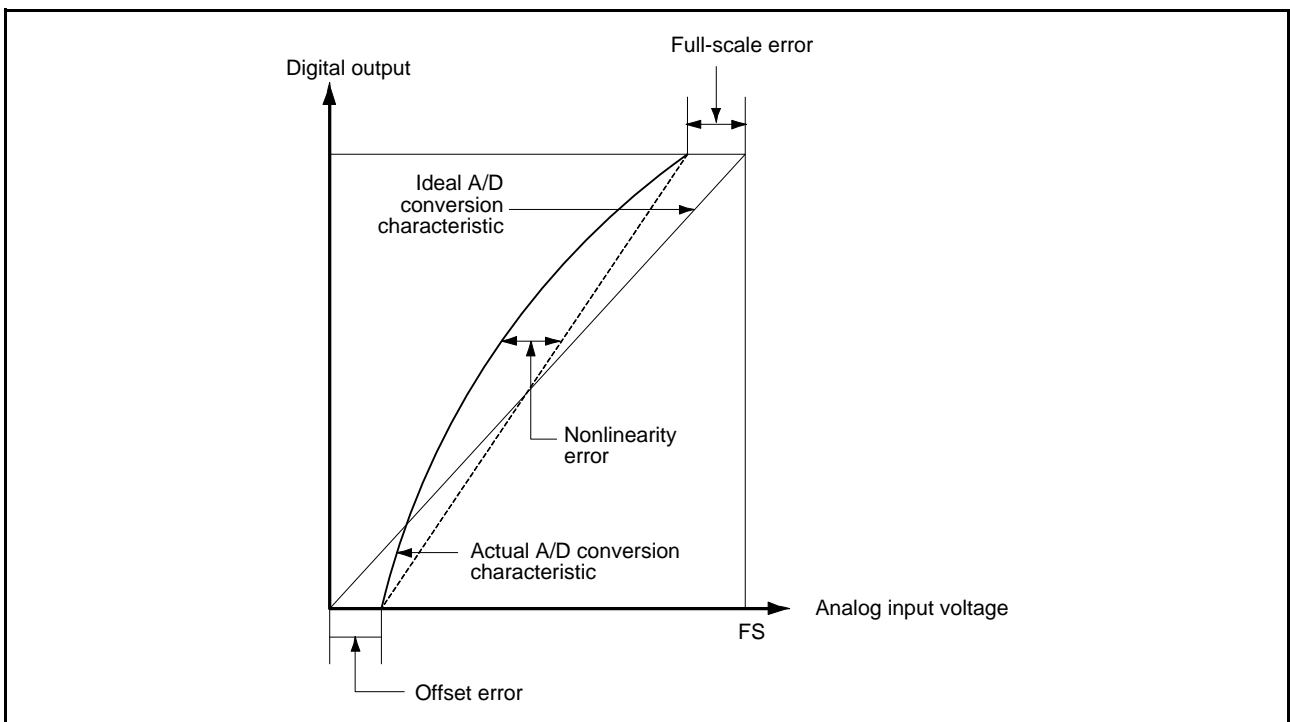


Figure 39.15 A/D Conversion Accuracy Definitions (2)

## 39.6 Usage Notes

### 39.6.1 Setting for the Module-Stop State

Module-stop control registers can be used to enable or disable operation of the 10-bit A/D converter. The 10-bit A/D converter is stopped in the initial state. The registers become accessible on release from the module-stop state. For details, refer to section 11, Low Power Consumption.

### 39.6.2 Notes on Forcibly Stopping A/D Conversion

If an asynchronous trigger (ADTRG# pin) or a synchronous trigger (from the MTU, TPU, or TMR modules) is selected as the condition for the start of A/D conversion, unintended A/D conversion may still start even if the ADCSR.ADST bit has been set to 0 to forcibly stop A/D conversion.

Therefore, to make sure that the A/D converter is stopped, set the ADCR.TRGS[2:0] bits to 000b to select the software trigger as the condition for the start of A/D conversion and then clear the ADCSR.ADST bit to 0 (to stop A/D conversion).

### 39.6.3 A/D Conversion Restart Timing after Forced Stop

When A/D conversion is forcibly stopped by clearing the ADCSR.ADST bit to 0, one ADCLK cycle of time is required to stop the analog section of the 10-bit A/D converter.

When A/D conversion is started by setting the ADCSR.ADST bit to 1 while the analog section of the 10-bit A/D converter has been stopped, A/D conversion restarts after one ADCLK cycle has elapsed.

### 39.6.4 Notes on Entering Power-Down States

When the RX630 Group device enters the module-stop state or software standby mode with A/D conversion enabled, the analog power supply current value becomes the same as that used during A/D conversion. To reduce the analog power supply current in the module-stop state or software standby mode, disable A/D conversion by clearing the ADCSR.ADST bit to 0. Note that to halt A/D conversion, sufficient time is required after the ADST bit is cleared to the stop of 10-bit A/D converter analog cycle. To ensure this, follow the procedure below.

1. Set the TRGS[2:0] bits in ADCR to 000b (software trigger).
2. Clear the ADST bit in ADCSR to 0.
3. Set the CKS[1:0] bits in ADCR to 11b (PCLK).
4. Ensure A/D conversion is stopped, then enter the device in the module-stop state or software standby mode.

### 39.6.5 Notes on Performing 10-Bit A/D Converter Synchronous D/A Conversion

To perform 10-bit A/D converter synchronous D/A conversion to avoid DA to AD interference by setting the DAADST bit in the D/A A/D synchronous start control register (DAADSCR) of the D/A converter to 0, set the ADCR.TRGS[2:0] bits to 000b (software trigger), and then set the D/A control register (DACR) of the D/A converter while the ADCSR.ADST bit is 0.

During A/D conversion, the clock should not be halted due to module-stop or other reasons. This may result in that D/A conversion as well as A/D conversion is halted. When A/D conversion is halted by setting the ADCSR.ADST bit to 0, one ADCLK cycle of time is required to halt the analog section of the 10-bit A/D converter.

When A/D conversion is restarted by setting the ADCSR.ADST bit to 1 immediately after setting the ADCSR.ADST bit to 0, A/D conversion is restarted after one ADCLK cycle period of time has elapsed.

### 39.6.6 Permissible Signal Source Impedance

This LSI's analog input is designed such that conversion precision is guaranteed for an input signal for which the signal source impedance is 1.0 k $\Omega$  or less to implement high conversion speed of 1.0  $\mu$ s. If a capacitor with large capacitance is connected for only one pin conversion in single channel mode, the input load will essentially comprise only the internal input resistance  $R_s$ , and the signal source impedance is ignored. However, as a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/ $\mu$ s or greater) (see Figure 39.16). When converting a high-speed analog signal or converting multiple pins in scan mode, a low impedance buffer should be inserted.

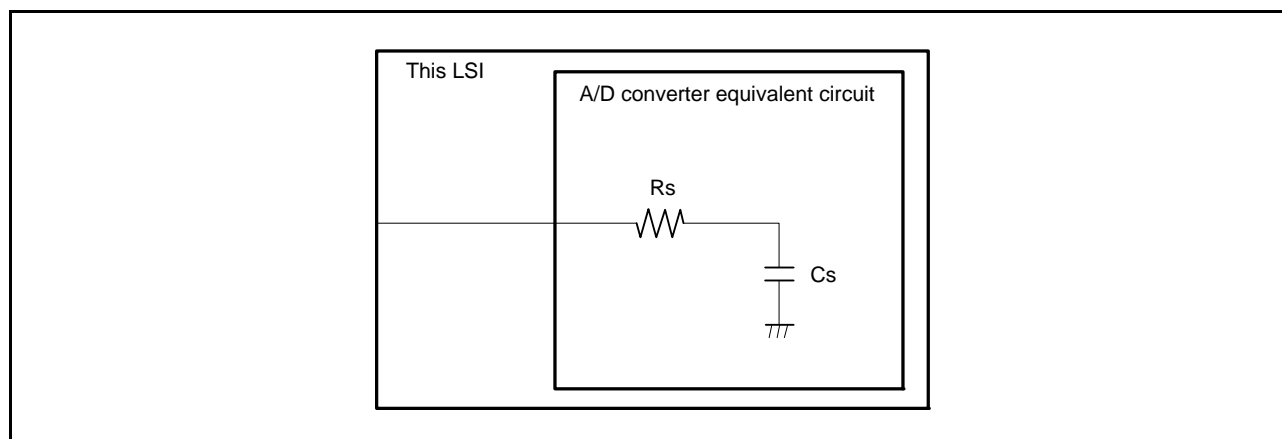


Figure 39.16 Internal Equivalent Circuit for Analog Input Pins

Table 39.9 Analog Pin Specifications (Reference Values)

Item		Min.	Max.	Unit
Permissible signal source impedance		—	1.0	k $\Omega$
Internal equivalent circuit for pins	$R_s$	—	6.5	k $\Omega$
	$C_s$	—	6	pF

### 39.6.7 Influences on Absolute Precision

Adding capacitor results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. Be sure to make the connection to an electrically stable GND.

Care is also required to ensure that filter circuits do not interfere with digital signals or act as antennas on the mounting board.

### 39.6.8 Analog Power Supply Pin Setting Range

If this LSI is used outside the range of the following voltage settings, the reliability of the LSI may be adversely affected.

- Analog input voltage range

The voltage applied to analog input pin (AN0 to AN7) should be in the range  $VREFL \leq V_{AN} \leq VREFH$ .

- Relationship between power supply pins (VREFH - VREFL, VCC - VSS)

Establish the relationship between the reference voltage input pins (VREFH, VREFL) and the digital power supply pins (VCC, VSS) so that  $VREFH \leq VCC$  and  $VREFL = VSS$ . As shown in Figure 39.17, connect a 0.1- $\mu$ F capacitor between each of the power supplies to form a closed loop in the shortest way so that  $VREFL = VSS$  in the source side. When the A/D converter is not used, set  $VREFH = VCC$  and  $VREFL = VSS$ .

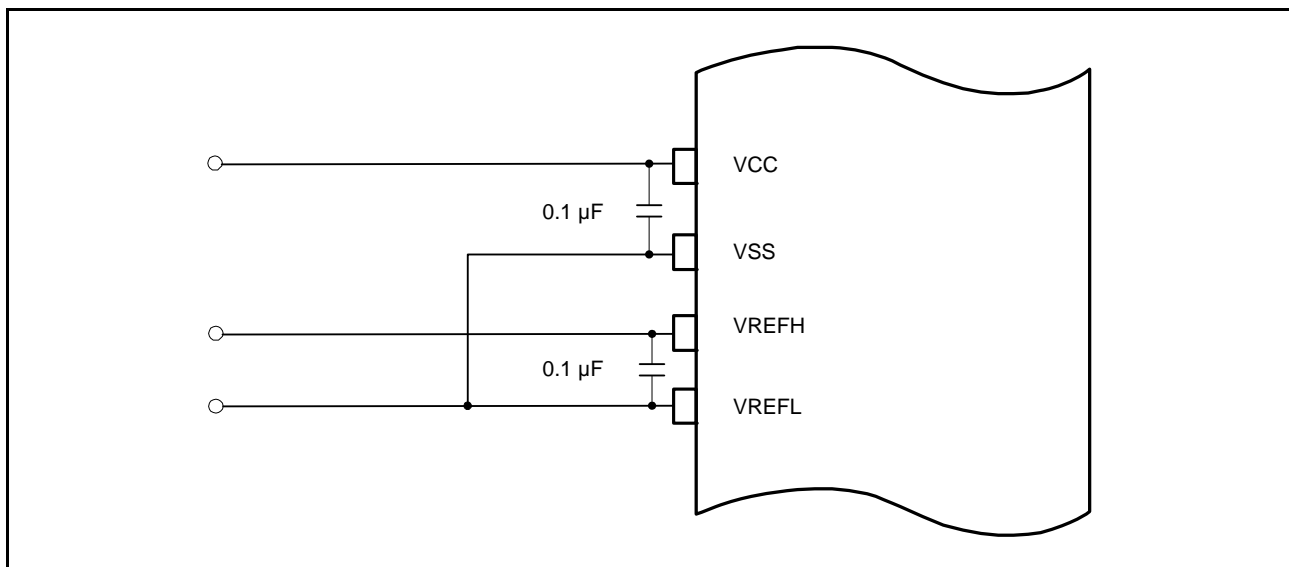


Figure 39.17 Connection Example of Power Supply Pins

### 39.6.9 Notes on Board Design

In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and the layout in which the digital circuit signal lines and analog circuit signal lines cross or are in close proximity to each other should be avoided as much as possible. Failure to do so may result in the incorrect operation of the analog circuitry due to inductance, adversely affecting the A/D conversion values.

In addition, digital circuitry must be isolated from the analog input signals (AN0 to AN7), analog power supply (VREFH), and analog ground (VREFL). The analog ground (VREFL) should be connected at one point to a stable digital ground (VSS) on the board.

### 39.6.10 Notes on Noise Countermeasures

To prevent damage due to an abnormal voltage, such as an excessive surge at the analog input pins (AN0 to AN7), connect capacitors between the VREFH and VREFL, and a protection circuit relative to the analog input pins (AN0 to AN7), as shown in Figure 39.18.

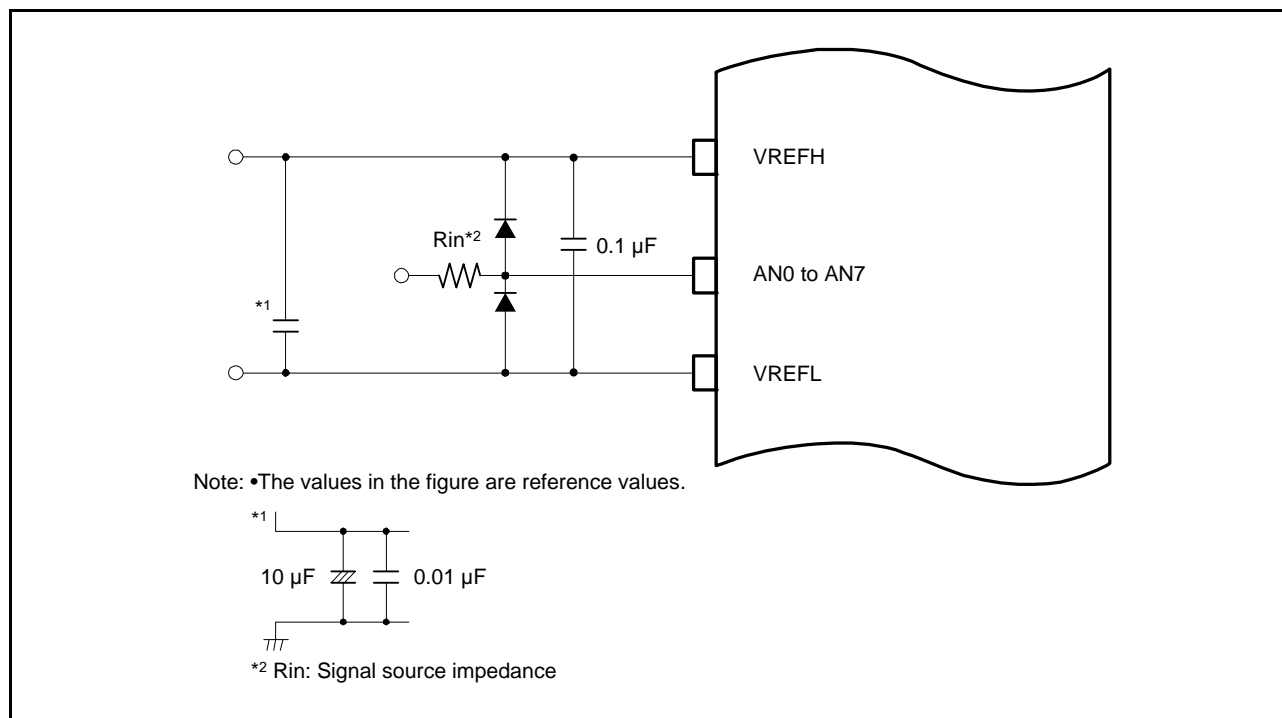


Figure 39.18 Example of Analog Input Pin Protection Circuit

### 39.6.11 Notes on Using the External Bus

When A/D conversion is done during access to the external bus, the accuracy may degrade.

In this case, repeat conversion several times and calculate by software the average of the converted values, excluding the maximum and minimum values.



### 39.6.12 Realizing High-Speed Conversion

To realize high-speed conversion, connect external 0.1- $\mu\text{F}$  capacitors between the analog input pins (AN0 to AN7) and VREFL. This is shown in Figure 39.19. However, to hide the impedance of the signal source due to the input capacitance of the sample-and-hold circuit of the 10-bit A/D converter, the externally connected capacitors must be fully charged before the start of conversion.

Furthermore, when the voltages on the analog input pins fluctuate due to scanning and so on, so describing renewal of the charge of the externally connected capacitors, such full charge is beyond the scope of this description.

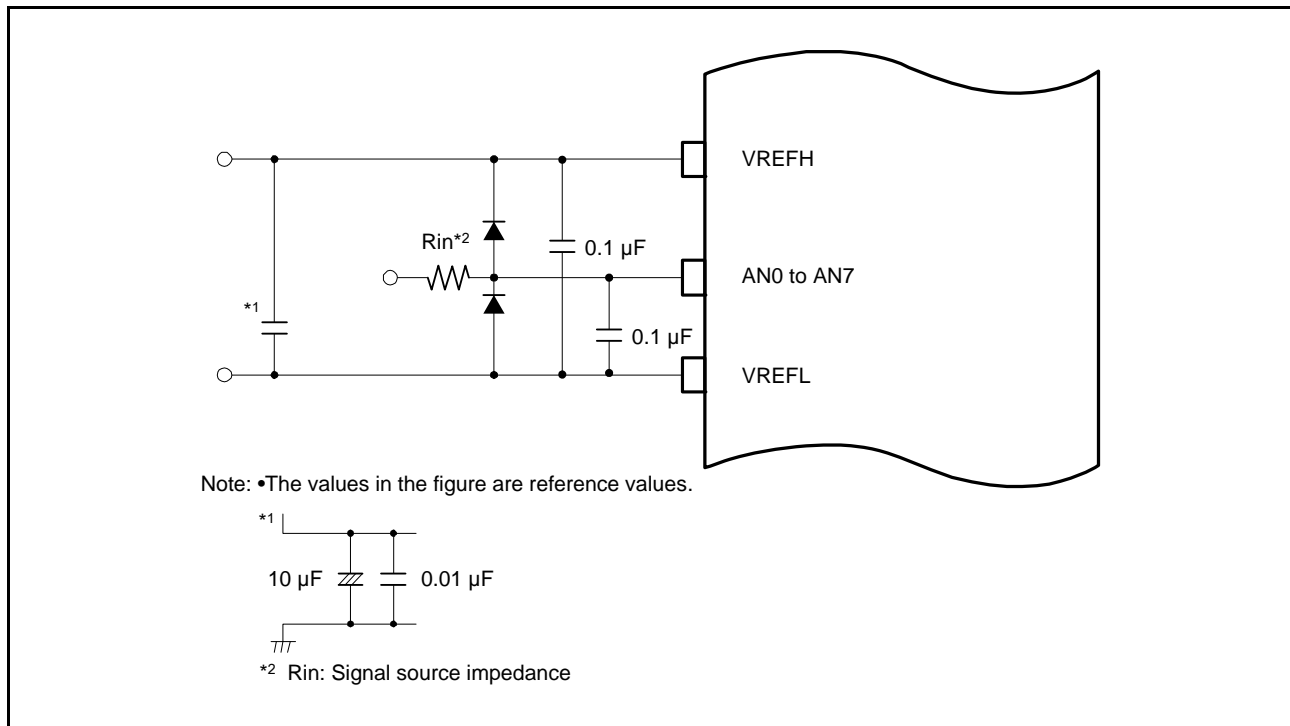


Figure 39.19 Example of an Externally Connected Capacitor for High-Speed Conversion

## 40. D/A Converter (DAa)

### 40.1 Overview

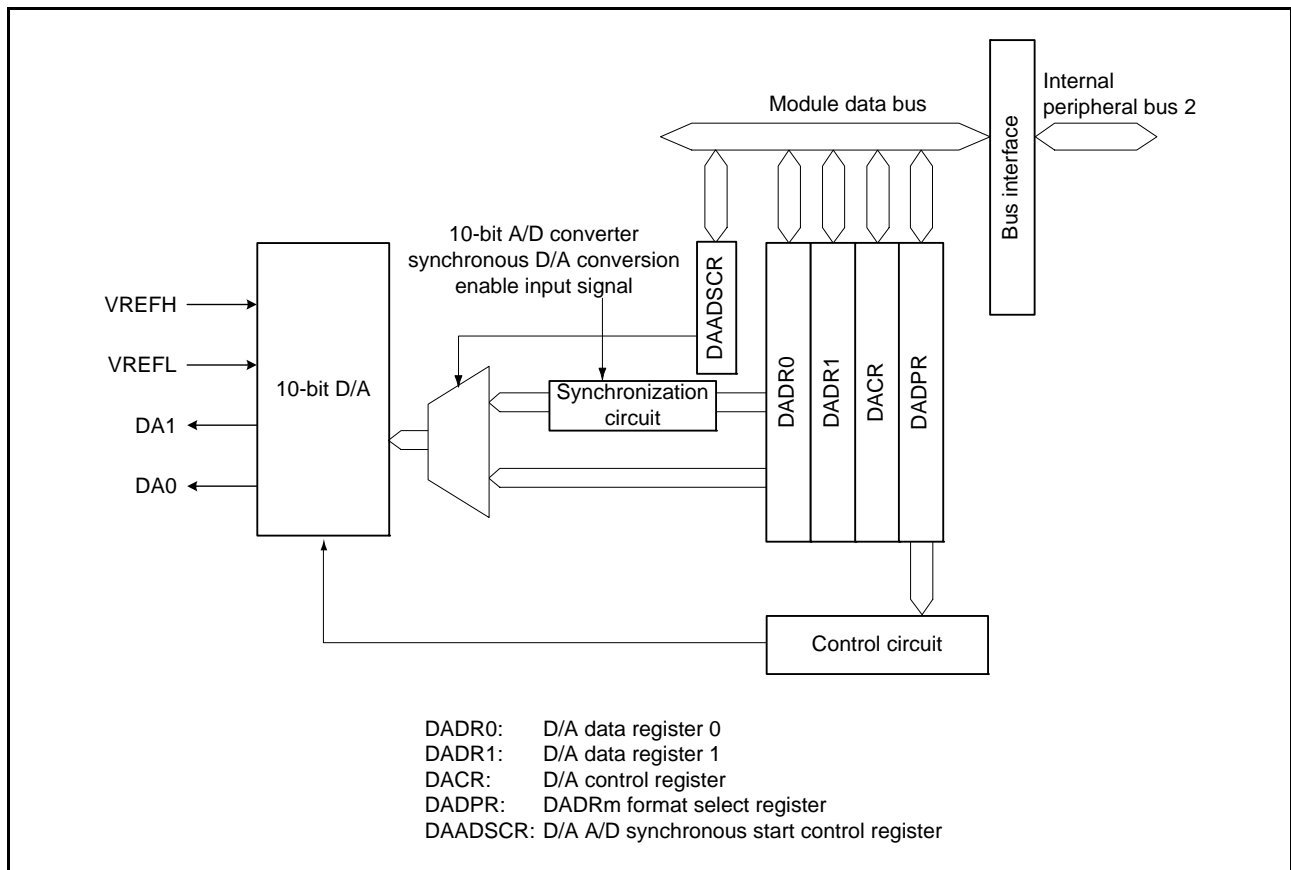
The RX630 Group includes two-channels of 10-bit D/A converter.

Table 40.1 lists the specifications of the D/A converter and Figure 40.1 shows a block diagram of the D/A converter.

**Table 40.1 Specifications of D/A Converter**

Item	Specifications
Resolution	10 bits
Output channels	Two channels
Countermeasure against mutual interference between analog modules*1	Measure against interference between D/A and A/D conversion D/A converted data update timing is controlled by the 10-bit A/D converter synchronous D/A conversion enable input signal from the 10-bit A/D converter (degradation of A/D conversion accuracy caused by interference is reduced by controlling the D/A converter inrush current generation timing with the enable signal).
Low-power consumption function	Module-stop state can be set for each unit.

Note 1. This function is not included in products with on-chip ROM capacity of 1.5 Mbytes or more or with 176 pins or more.



**Figure 40.1 Block Diagram of D/A Converter**

Table 40.2 lists the pin configuration of the D/A converter.

**Table 40.2 Pin Configuration of D/A Converter**

Pin Name	I/O	Function
VREFH	Input	Reference voltage input pin for the 10-bit A/D converter and D/A converter. This pin is also used as an analog power supply pin for each module. Connect to VCC when neither of these modules are used.
VREFL	Input	Reference voltage input pin for the 10-bit A/D converter and D/A converter. This pin is also used as an analog ground pin for each module. Set VREFL to the same potential as VSS.
DA0	Output	Channel 0 analog output pin
DA1	Output	Channel 1 analog output pin

## 40.2 Register Descriptions

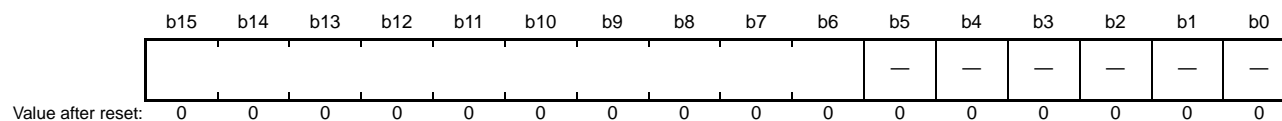
### 40.2.1 D/A Data Register m (DADRm) (m = 0, 1)

Address(es): DADR0 0008 80C0h, DADR1 0008 80C2h

- DADPR.DPSEL bit = 0 (data are flush with the right end of the register)



- DADPR.DPSEL bit = 1 (data are flush with the left end of the register)



DADRm registers are 16-bit readable/writable registers, which store data to which D/A conversion is to be performed. Whenever an analog output is enabled, the values in DADRm are converted and output to the analog output pins. 10-bit data can be relocated by setting the DPSEL bit in DADPR. Bits “—” are read as 0. The write value should be 0.

## 40.2.2 D/A Control Register (DACR)

Address(es): 0008 80C4h

b7	b6	b5	b4	b3	b2	b1	b0
DAOE1	DAOE0	DAE	—	—	—	—	—

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	DAE*1	D/A Enable	0: D/A conversion is independently controlled on channels 0 and 1. 1: D/A conversion on channels 0 and 1 is enabled as a single whole.	R/W
b6	DAOE0	D/A Output Enable 0	0: Analog output of channel 0 (DA0) is disabled. 1: D/A conversion of channel 0 is enabled. Analog output of channel 0 (DA0) is enabled.*2	R/W
b7	DAOE1	D/A Output Enable 1	0: Analog output of channel 1 (DA1) is disabled. 1: D/A conversion of channel 1 is enabled. Analog output of channel 1 (DA1) is enabled.*2	R/W

Note 1. This bit controls D/A conversion in combination with the DAOEi bit (i = 0, 1). The DAOEi bit controls output of the results of conversion. For details, see Table 40.3.

Note 2. Set the P0.PDR.Bm bit (m = 3, 5) for pins used as analog outputs and the corresponding P0.PMR.Bm bit (m = 3, 5) to 0. In addition, set them to analog pins by the P03PFS and P05PFS registers. For details, see section 20, I/O Ports, and section 21, Multi-Function Pin Controller (MPC).

Table 40.3 Controlling of D/A Conversion

b5	b7	b6	Description
DAE	DAOE1	DAOE0	
0	0	0	D/A conversion and analog output pins (DA0, DA1) are disabled.*1
		1	D/A conversion of channel 0 is enabled. D/A conversion of channel 1 is disabled. Analog output of channel 0 (DA0) is enabled. Analog output of channel 1 (DA1) is disabled.*1
	1	0	D/A conversion of channel 0 is disabled. D/A conversion of channel 1 is enabled. Analog output of channel 0 (DA0) is disabled.*1 Analog output of channel 1 (DA1) is enabled.
		1	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0, DA1) is enabled.
1	0	0	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0, DA1) is disabled.*1
		1	D/A conversion of channels 0 and 1 is enabled.sss Analog output of channel 0 (DA0) is enabled. Analog output of channel 1 (DA1) is disabled.*1
	1	0	D/A conversion of channels 0 and 1 is enabled. Analog output of channel 0 (DA0) is disabled.*1 Analog output of channel 1 (DA1) is enabled.
		1	D/A conversion of channels 0 and 1 is enabled. Analog output of channels 0 and 1 (DA0, DA1) is enabled.

Note 1. When analog output is disabled, the analog output signal is placed in the Hi-Z state.

This register should be set while the 10-bit A/D converter is halted when the DAADSCR.DAADST bit is 1 (enabling the measure against interference between D/A and A/D conversion) (The register should be set while the ADCSR.ADST bit is 0 after the software trigger is selected as the trigger source for the 10-bit A/D converter).

**DAE Bit (D/A Enable)**

The DAE bit controls D/A conversion in combination with the DAOEi (i = 0 or 1) bit.

When the DAE bit is 0, D/A conversion is independently controlled on channels 0 and 1. When the DAE bit is 1, D/A conversion on channels 0 and 1 is controlled as a single whole. The DAOEi bit controls output of the results of conversion.

When the measure against interference for D/A and A/D conversion is enabled (the DAADSCR.DAADST bit = 1), set this bit while the ADCSR.ADST bit is set to 0. To ensure that the 10-bit A/D converter remains stopped while the setting is made, select the software trigger as the trigger source in advance.

**DAOE0 Bit (D/A Output Enable 0)**

The DAOE0 bit controls the D/A conversion and analog output.

When the measure against interference between D/A and A/D conversion is enabled (the DAADSCR.DAADST bit = 1), set this bit while the ADCSR.ADST bit is set to 0. To ensure that the 10-bit A/D converter remains stopped while the setting is made, select the software trigger as the trigger source in advance.

**DAOE1 Bit (D/A Output Enable 1)**

The DAOE1 bit controls the D/A conversion and analog output.

When the measure against interference between D/A and A/D conversion is enabled (the DAADSCR.DAADST bit = 1), set this bit while the ADCSR.ADST bit is set to 0. To ensure that the 10-bit A/D converter remains stopped while the setting is made, select the software trigger as the trigger source in advance.

**40.2.3 DADRm Format Select Register (DADPR)**

Address(es): 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	DADRm Format Select	0: Data is flush with the right end of the D/A data register. 1: Data is flush with the left end of the D/A data register.	R/W

### 40.2.4 D/A A/D Synchronous Start Control Register (DAADSCR)

Address(es): 0008 80C6h

b7	b6	b5	b4	b3	b2	b1	b0
DAADST	—	—	—	—	—	—	—

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	DAADST	D/A A/D Synchronous Conversion	0: D/A converter operation does not synchronize with 10-bit A/D converter operation. (measure against interference between D/A and A/D conversion is disabled) 1: D/A converter operation synchronizes with 10-bit A/D converter operation. (measure against interference between D/A and A/D conversion is enabled)	R/W

As a measure against interference between D/A and A/D conversion, DAADSCR switches synchronization of the timing of the start of D/A conversion with the 10-bit A/D converter synchronous D/A conversion enable input signal off or on. This register should be set while the 10-bit A/D converter is halted (while the ADCSR.ADST bit is 0 after selecting software trigger as the 10-bit A/D converter trigger).

This register is not included in products with on-chip ROM capacity of 1.5-Mbytes or more or with 176 pins or more.

#### DAADST Bit (D/A A/D Synchronous Conversion)

Setting the DAADST bit to 0 allows the DADRm register value to be converted into analog data at any time. Setting the DAADST bit to 1 allows synchronous D/A conversion with the synchronous D/A conversion enable signal from the A/D converter. Therefore, even if the DADRm register value is modified, D/A conversion does not start until the A/D converter completes A/D conversion.

Set this bit while the ADCSR.ADST bit is set to 0. To ensure that the 10-bit A/D converter remains stopped while the setting is made, select the software trigger as the trigger source in advance.

### 40.3 Operation

The D/A converter includes D/A conversion circuits for two channels, each of which can operate independently. When the DAOEi bit (i = 0, 1) in DACR is set to 1, D/A converter is enabled and the conversion result is output.

An operation example of D/A conversion on channel 0 is shown below. Figure 40.2 shows the timing of this operation.

1. Write the data for conversion to DADR0.
2. Set the DAOE0 bit in DACR to 1 to start D/A conversion. The conversion result is output from the analog output pin DA0 after the conversion time  $t_{DCONV}$  has elapsed. The conversion result continues to be output until DADR0 is written to again or the DAOE0 bit is cleared to 0. The output value is expressed by the following formula:

$$\frac{\text{Setting value of DADR0}}{1024} \times V_{REFH}$$

3. If DADR0 is written to again, the conversion is immediately started. The conversion result is output after the conversion time  $t_{DCONV}$  has elapsed.

When the DAADSCR.DAADST bit is 1 (measure against interference between D/A and A/D conversion is enabled), it takes a maximum of one A/D conversion time for D/A conversion to start.

4. If the DAOE0 bit is cleared to 0, analog output is disabled.

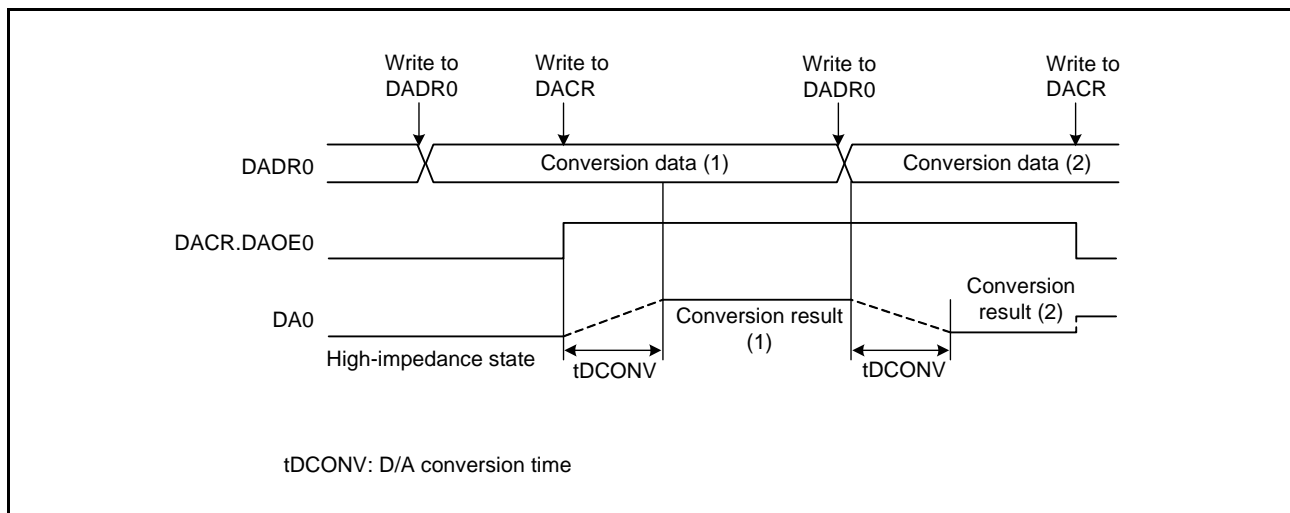


Figure 40.2 Example of D/A Converter Operation

### 40.3.1 Measure against Interference between D/A and A/D Conversion

When D/A conversion starts, the D/A converter generates inrush current. Since the same analog power supply is shared by the D/A converter and 10-bit A/D converter, the generated inrush current may interfere with 10-bit A/D converter operation.

To prevent such interference, the D/A converter start timing can be synchronized with the 10-bit A/D converter synchronous D/A conversion enable signal.

With the DAADSCR.DAADST bit being 1, even if the DADRm register data is modified during 10-bit A/D converter operation, D/A conversion does not start immediately but starts synchronously with A/D conversion completion. It takes a maximum of one A/D conversion time for the DADRm register data update to be reflected as the D/A conversion circuit input. Before reflection, the DADRm register value does not correspond to the analog output value.

When this function is enabled, it is impossible to check by any software means whether the DADRm register value has been D/A converted or not.

Even with DAADSCR.DAADST being 1, when the DADRm register data is modified while the 10-bit A/D converter is halted, D/A conversion starts in one PCLK cycle.

The following describes an example of channel 0 D/A conversion, in which the D/A converter operates synchronously with the 10-bit A/D converter.

- (1) Confirm that the 10-bit A/D converter is halted. Set the DAADSCR.DAADST bit to 1.
  - (2) Confirm that the 10-bit A/D converter is halted. Set the DACR.DAOE0 bit to 1.
  - (3) Set the DADR0 register.
- If the 10-bit A/D conversion is halted when the DADR0 register is modified, D/A conversion starts in one PCLK cycle.
  - If the 10-bit A/D conversion is in progress when the DADR0 register is modified, D/A conversion starts upon A/D conversion completion. If the DADR0 register is modified twice during A/D conversion, the first update may not be converted.

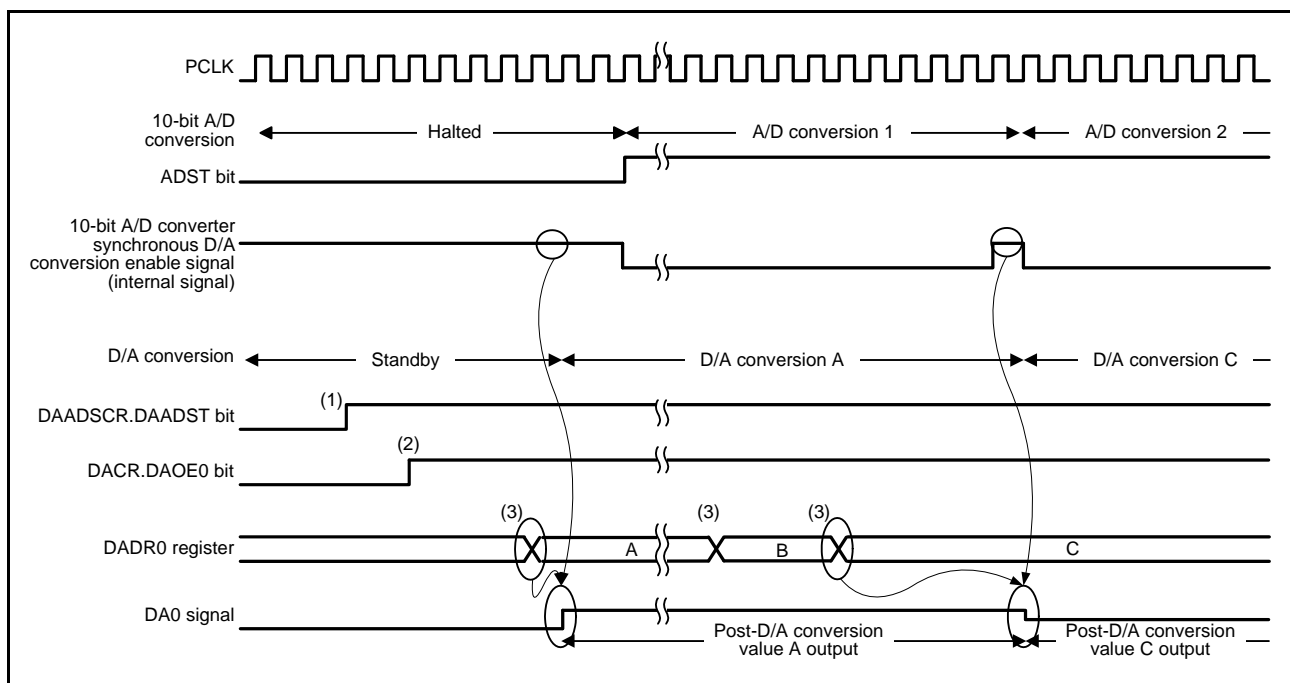


Figure 40.3 Example of Conversion when the D/A Converter is Synchronized with the 10-Bit A/D Converter



## 40.4 Usage Notes

### 40.4.1 Module-Stop Function Setting

Operation of the D/A converter can be disabled or enabled by using the module stop control register. The initial setting is for operation of the D/A converter to be halted. Register access is enabled by clearing the module-stop state. For details, refer to section 11, Low Power Consumption.

### 40.4.2 Operation of the D/A Converter in Module-Stop State

When the RX630 Group enters the module-stop state with D/A conversion enabled, the D/A 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 clearing the DAOE1, DAOE0, and DAE bits in DACR to 0.

### 40.4.3 Operation of the D/A Converter in Software Standby Mode

When the RX630 Group enters software standby mode with D/A conversion enabled, the D/A 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 clearing the DAOE1, DAOE0, and DAE bits in DACR to 0.

### 40.4.4 Note on Entering Deep Software Standby Mode

When the RX630 Group enters deep software standby mode with D/A conversion enabled, the outputs of the D/A converter are placed in a high impedance state.

### 40.4.5 Note on Usage when Measure against Interference between D/A and A/D Conversion is Enabled

When the DAADSCR.DAADST bit is 1 (measure against interference between D/A and A/D conversion is enabled), do not place the 10-bit A/D converter into the module-stop state. It may halt D/A conversion in addition to A/D conversion.

## 41. Temperature Sensor

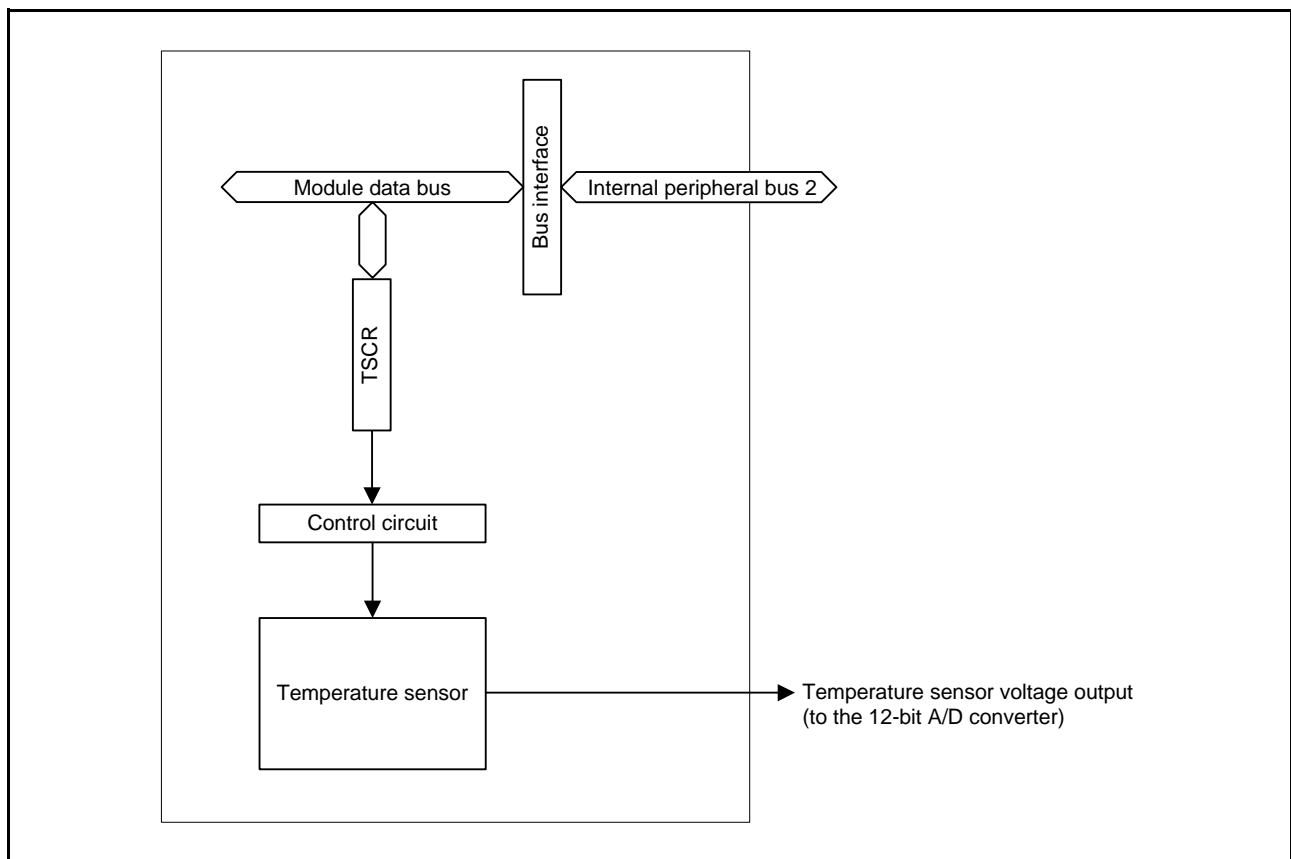
### 41.1 Overview

The RX630/RX630 Group includes a temperature sensor. The temperature sensor outputs a voltage which varies with the temperature. The 12-bit A/D converter can convert the voltage from the sensor into a digital value. The user can then obtain the temperature around the LSI by converting the value into the temperature.

Table 41.1 lists the specifications of the temperature sensor, and Figure 41.1 shows a block diagram of the temperature sensor.

**Table 41.1 Specifications of Temperature Sensor**

Item	Description
Temperature sensor voltage output	Temperature sensor outputs a voltage to the 12-bit A/D converter.
Low-power consumption function	The module-stop state is selectable.



**Figure 41.1 Block Diagram of Temperature Sensor**

## 41.2 Register Descriptions

### 41.2.1 Temperature Sensor Control Register (TSCR)

Address: 0008 C500h

b7	b6	b5	b4	b3	b2	b1	b0
TSEN	—	—	TSOE	—	—	—	—

Value after reset: 0 0 0 0 0 0 0 0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	TSOE	Temperature Sensor Output Enable	0: Disables output from the temperature sensor to the 12-bit A/D converter. 1: Enables output from the temperature sensor to the 12-bit A/D converter.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	TSEN	Temperature Sensor Enable	0: Stops the temperature sensor. 1: Starts the temperature sensor.	R/W

### 41.3 Using the Temperature Sensor

The temperature sensor outputs a voltage which varies with the temperature.

This voltage is converted to a digital value by the 12-bit A/D converter. The user can then obtain the temperature around the LSI by converting the value into the temperature.

#### 41.3.1 Preparation for Using the Temperature Sensor

The temperature characteristics of the temperature sensor are shown below. The voltage output by the temperature sensor is proportional to temperature, which can be calculated according to the following formula.

Formula for the temperature characteristic:

$$T = (V_s - V_1)/\text{Slope} + T_1$$

T: Measured temperature (°C)

V<sub>s</sub>: Voltage output by the temperature sensor at the time of temperature measurement (V)

T<sub>1</sub>: Temperature experimentally measured at one point (°C)

V<sub>1</sub>: Voltage output by the temperature sensor at the time of measurement of T<sub>1</sub> (V)

T<sub>2</sub>: Temperature at the experimental measurement of another point (°C)

V<sub>2</sub>: Voltage output by the temperature sensor at the time of measurement of T<sub>2</sub> (V)

Slope: Temperature gradient by the temperature sensor (V/°C); slope = (V<sub>2</sub> - V<sub>1</sub>)/(T<sub>2</sub> - T<sub>1</sub>)

Characteristics vary from sensor to sensor. Therefore, the following experimental measurement at two different temperatures is recommended.

Use the 12-bit A/D converter to measure the voltage V<sub>1</sub> output by the temperature sensor at temperature T<sub>1</sub>.

Again, using the 12-bit A/D converter, measure the voltage V<sub>2</sub> output by the temperature sensor at a different temperature T<sub>2</sub>. Obtain the temperature gradient (slope = (V<sub>2</sub> - V<sub>1</sub>)/(T<sub>2</sub> - T<sub>1</sub>)) from these results.

Subsequently, obtain temperatures by substituting the slope into the formula for the temperature characteristic (T = (V<sub>s</sub> - V<sub>1</sub>)/slope + T<sub>1</sub>).

If you are using the temperature gradient given in Table 45.25 of section 45, Electrical Characteristics, use the 12-bit A/D converter to measure the voltage V<sub>1</sub> output by the temperature sensor at temperature T<sub>1</sub>, and then calculate the temperature characteristic by using the formula below.

However, this method gives less accurate temperatures than measurement at two points.

$$T = (V_s - V_1)/\text{Slope} + T_1$$

T: Measured temperature (°C)

V<sub>s</sub>: Voltage output by the temperature sensor at the time of temperature measurement (V)

T<sub>1</sub>: Temperature experimentally measured at one point (°C)

V<sub>1</sub>: Voltage output by the temperature sensor at the time of measurement of T<sub>1</sub> (V)

Slope: Temperature gradient (V/°C) given in Table 45.25.

### 41.3.2 Setting of 12-Bit A/D Converter

For A/D conversion of temperature sensor output voltages, 12-bit A/D converter registers should be set as follows.

- **Selecting the Temperature Sensor Voltage as an A/D Conversion Target**  
Select A/D conversion of the voltage from the temperature sensor by setting the temperature sensor output A/D conversion select bit in the A/D conversion extended input control register (ADEXICR.TSS) to 1. Additionally, remove other sources from the scope of conversion by setting all of the bits in the A/D channel select registers 0 and 1 (ADANS0 and ADANS1), and the internal reference voltage A/D conversion select bit in the A/D conversion extended input control register (ADEXICR.OCS), to 0.
- **Setting Single Scan Mode**  
Select single scan mode by setting the scan mode select bit in the A/D control register (ADCSR.ADCS) to 0.
- **Setting Addition Mode**  
For A/D conversion of the temperature sensor output, addition mode is selectable. To enable addition mode, set the temperature sensor output A/D converted value addition mode select bit in the A/D conversion extended input control register (ADEXICR.TSSAD) to 1, and the addition count select bits in the A/D-converted value addition count select register (ADADC.ADC[1:0]) to the desired number for addition.
- **Setting the Number of Sampling States of the 12-Bit A/D Converter**  
The number of states for sampling of the output of the temperature sensor for A/D conversion is selectable. The initial setting is for 20 states. To change the number of states for sampling from 20 states, set the sampling time 2 setting bits in A/D sampling state register 23 (ADSSTR23.SST2[7:0]).

Setting the A/D conversion start bit in the A/D control register (ADCSR.ADST) to 1 starts A/D conversion, and the result is stored in the A/D temperature sensor data register (ADTSDR). If you will be using A/D conversion of the output from the temperature sensor, do so in accord with section 41.3.3, Procedure for Using the Temperature Sensor.

### 41.3.3 Procedure for Using the Temperature Sensor

Figure 41.2 shows the procedure for using the temperature sensor.

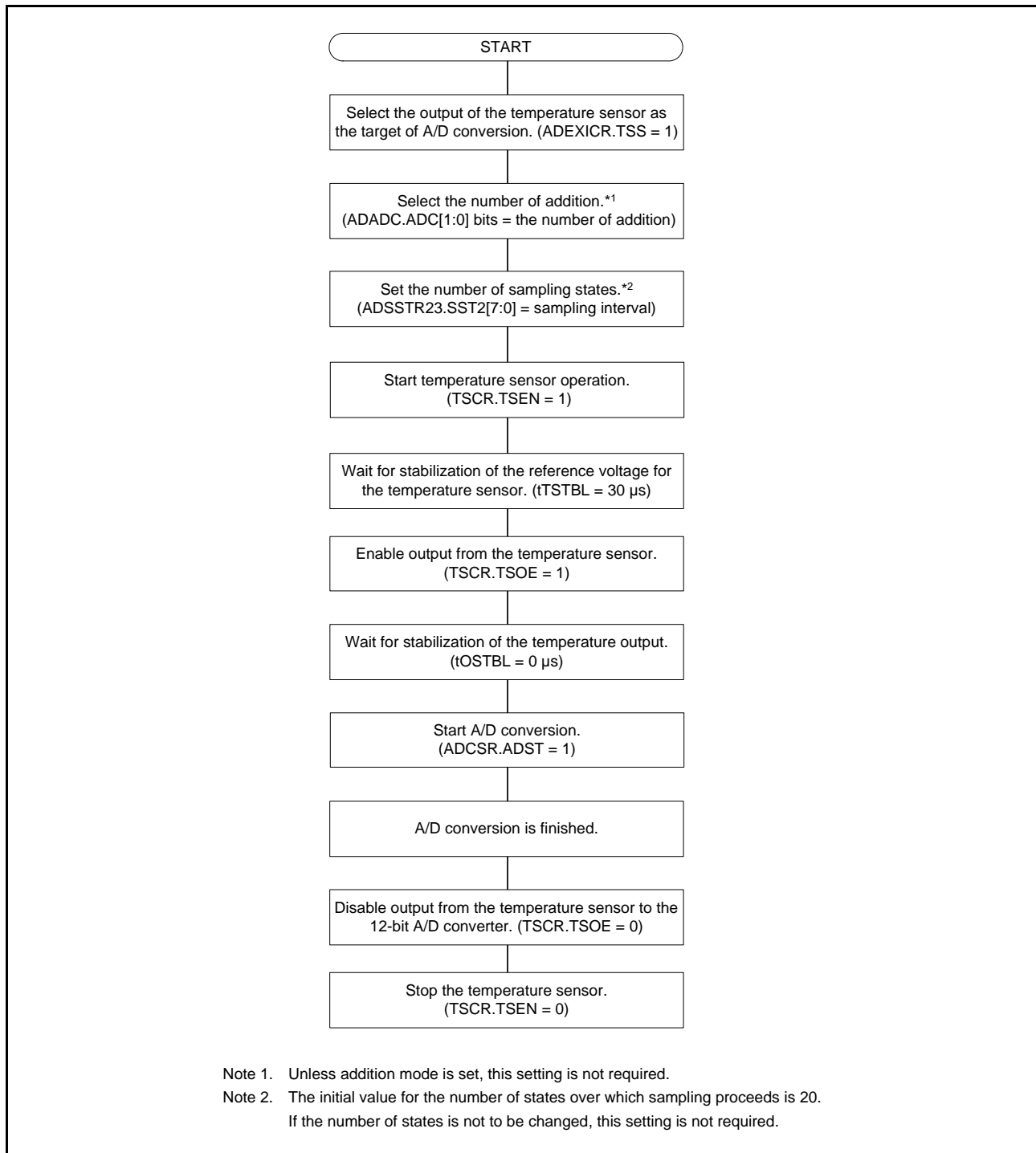
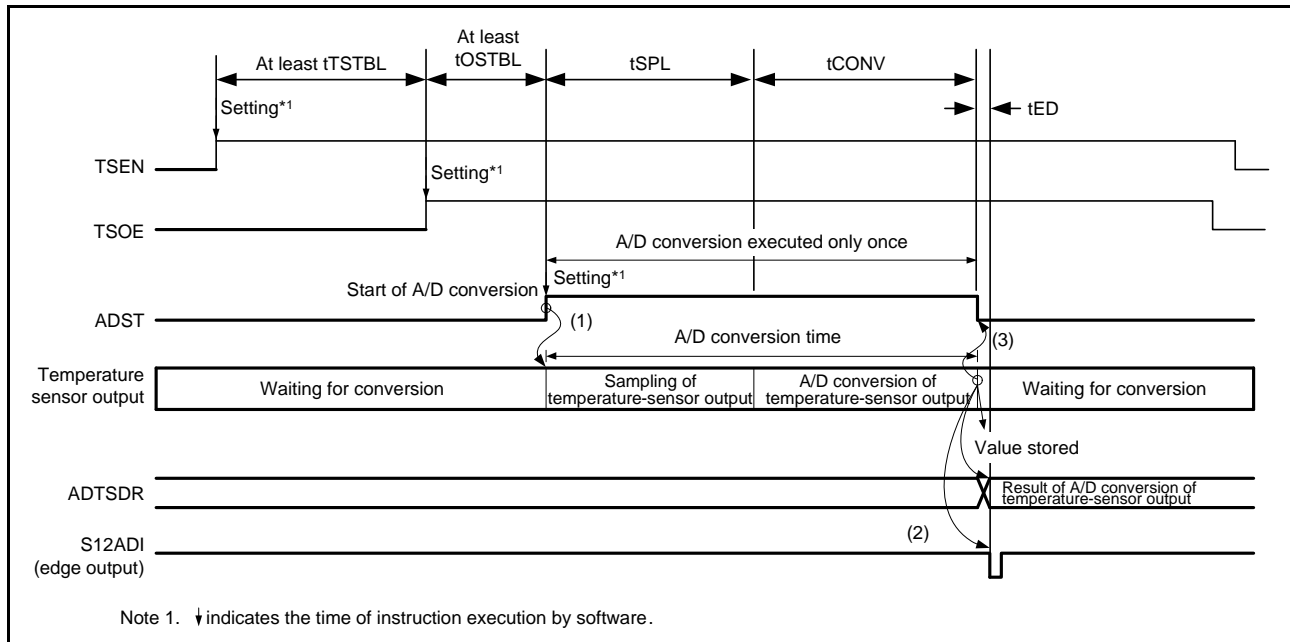


Figure 41.2 Procedure for Using the Temperature Sensor

### 41.3.4 Timing of A/D Conversion of Temperature Sensor Output

Figure 41.3 shows the timing from the start of temperature-sensor operation until the completion of A/D conversion. Times are listed in Table 41.2.



**Figure 41.3** Timing from the Start of Temperature-Sensor Operation until Completion of A/D Conversion

**Table 41.2** Time until Completion of A/D Conversion after the Start of Temperature-Sensor Operation

Item	Symbol	Time
Waiting time for temperature-sensor reference-voltage stabilization	tTSTBL	30 $\mu$ s (max)
Waiting time for temperature-sensor output stabilization	tOSTBL	0 $\mu$ s (max)
12-bit A/D converter input sampling time	tSPL	ADSSTR23 setting $\times$ PCLK cycles
A/D conversion time	tCONV	Refer to Table 38.7, Times Required in Scanning Operations (as Numbers of ADCLK and PCLK Cycles) in section 38.3.4, Analog Input Scan Time
Scan conversion end delay time	tED	Refer to Table 38.7, Times Required in Scanning Operations (as Numbers of ADCLK and PCLK Cycles) in section 38.3.4, Analog Input Scan Time

## 41.4 Usage Note

### 41.4.1 Module-Stop Function Setting

The corresponding bit in module stop control register B (MSTPCRB) can be used to enable and disable the temperature sensor. The initial setting is for the temperature sensor to be stopped. The register becomes accessible on release from the module-stop state. For details, see section 11, Low Power Consumption.

## 42. RAM

The RX630 Group has an on-chip high-speed static RAM.

### 42.1 Overview

Table 42.1 lists the specifications of the RAM.

**Table 42.1 Specifications of RAM**

Item	Description
RAM capacity	Max. 128 Kbytes (RAM0: 64 Kbytes, RAM1: 64 Kbytes)*2
Access	<ul style="list-style-type: none"> <li>• Single-cycle access is possible for both reading and writing.</li> <li>• RAM can be enabled or disabled.*1</li> </ul>
Data retention function	Data in RAM0 can be retained in deep software standby mode.
Low-power consumption function	The module-stop state is independently selectable for RAM0 and RAM1.

Note 1. Selectable by the RAME bit in SYSCR1. For details on SYSCR1, see section 3.2.4, System Control Register 1 (SYSCR1).

Note 2. The capacity of RAM differs depending on the products.

RAM Capacity	RAM Address
64 Kbytes	RAM0: 0000 0000h to 0000 FFFFh (64 Kbytes)
96 Kbytes	RAM0: 0000 0000h to 0000 FFFFh (64 Kbytes) RAM1: 0001 0000h to 0001 7FFFh (32 Kbytes)
128 Kbytes	RAM0: 0000 0000h to 0000 FFFFh (64 Kbytes) RAM1: 0001 0000h to 0001 FFFFh (64 Kbytes)

## 42.2 Operation

### 42.2.1 Data Retention

The address space for RAM is divided into the RAM0 and RAM1 areas. The difference between the two is whether internal power can be supplied in deep software standby mode.

Whether or not the supply of internal power to RAM0 continues in deep software standby mode is selectable by the DPSBYCR.DEEPCUT[1:0] bits.

If continuation of the supply of internal power is selected, data in RAM0 are retained in deep software standby mode.

The supply of internal power supply to RAM1 is stopped at this time, so data are not retained in RAM1.

See section 11, Low Power Consumption, for details on the DPSBYCR.DEEPCUT[1:0] bits.

### 42.2.2 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 RAM.

If the MSTPC0 bit in MSTPCRC is set to 1, supply of the clock signal to RAM0 is stopped. If the MSTPC1 bit in MSTPCRC is set to 1, supply of the clock signal to RAM1 is stopped.

The respective modules (RAM0 and RAM1) are thus placed in the module-stop state by stopping supply of the clock signals. The RAM operates after a reset.

RAM is not accessible if it is in the module-stop state. A transition to the module-stop state should not be made while access to RAM is in progress.

For details on the MSTPCRC register, see section 11, Low Power Consumption.



## 43. Flash Memory

The RX630 Group has a maximum 2-Mbyte flash memory for storing code (ROM) and a 32-Kbyte flash memory for storing data (E2 DataFlash).

### 43.1 Overview

Table 43.1 lists the specifications of the ROM and E2 DataFlash memory, and Figure 43.1 is a block diagram of the ROM, E2 DataFlash memory, and related modules.

Regarding the configuration of the ROM memory area, refer to section 43.1.1, Configuration of the ROM Area, and for the configuration of the memory area for the E2 DataFlash, refer to section 43.1.3, Configuration of the E2 DataFlash Area.

**Table 43.1 Specifications of ROM/E2 DataFlash**

Item	ROM	E2 DataFlash
Memory space	User area: 2 Mbytes max. User boot area: 16 Kbytes	Data area: 32 Kbytes
Read cycle	A read operation takes one cycle of ICLK	A read operation takes six cycles of FCLK in words or bytes
Programming/erasing method	<ul style="list-style-type: none"> <li>The chip incorporates a dedicated sequencer (FCU) for programming of the ROM/E2 DataFlash.</li> <li>Programming and erasing the ROM/E2 DataFlash are handled by issuing commands to the FCU.</li> </ul>	
Value after erasure	FFh	Undefined
BGO (background operation)	<ul style="list-style-type: none"> <li>The CPU is able to execute program code from the ROM while the E2 DataFlash memory is being programmed or erased.</li> </ul>	
Suspension and resumption	<ul style="list-style-type: none"> <li>The CPU is able to execute program code from the ROM during suspension of programming or erasure.</li> <li>The CPU is able to execute program code from the E2 DataFlash during suspension of programming or erasure.</li> <li>Programming and erasure of the ROM/E2 DataFlash can be restarted (resumed) after suspension.</li> </ul>	
Units of programming and erasure	<ul style="list-style-type: none"> <li>Units of programming for the user area or user boot area: 128 bytes</li> <li>Units of erasure for the user area: In block units</li> <li>Units of erasure for the user boot area: 16 Kbytes</li> </ul>	<ul style="list-style-type: none"> <li>Unit of programming for the data area: 2 bytes</li> <li>Unit of erasure for the data area: 32 bytes</li> </ul>
On-board programming (four types)	Programming in boot mode <ul style="list-style-type: none"> <li>The asynchronous serial interface (SCI1) is used.</li> <li>The transfer rate is adjusted automatically.</li> <li>The user boot area can also be programmed.</li> </ul> Programming in USB boot mode <ul style="list-style-type: none"> <li>USB0 is used.</li> <li>Dedicated hardware is not required, so direct connection to a PC is possible.</li> </ul> Programming in the user boot mode <ul style="list-style-type: none"> <li>Able to create original boot programs of the user's making.</li> </ul> Programming by a routine for ROM/E2 DataFlash programming within the user program <ul style="list-style-type: none"> <li>This allows ROM/E2 DataFlash programming without resetting the system.</li> </ul>	
Off-board programming	A Flash programmer can be used to program the user area and user boot area.	A Flash programmer cannot be used to program the data area.
Protection	Software-controlled protection	<ul style="list-style-type: none"> <li>The registers can be used to prevent unintentional programming or reading.</li> <li>Protection with the registers is performed on a 2-Kbyte basis.</li> </ul>
	FCU command-lock	When abnormal operations are detected during programming/erasure, this function disables any further programming/erasure.

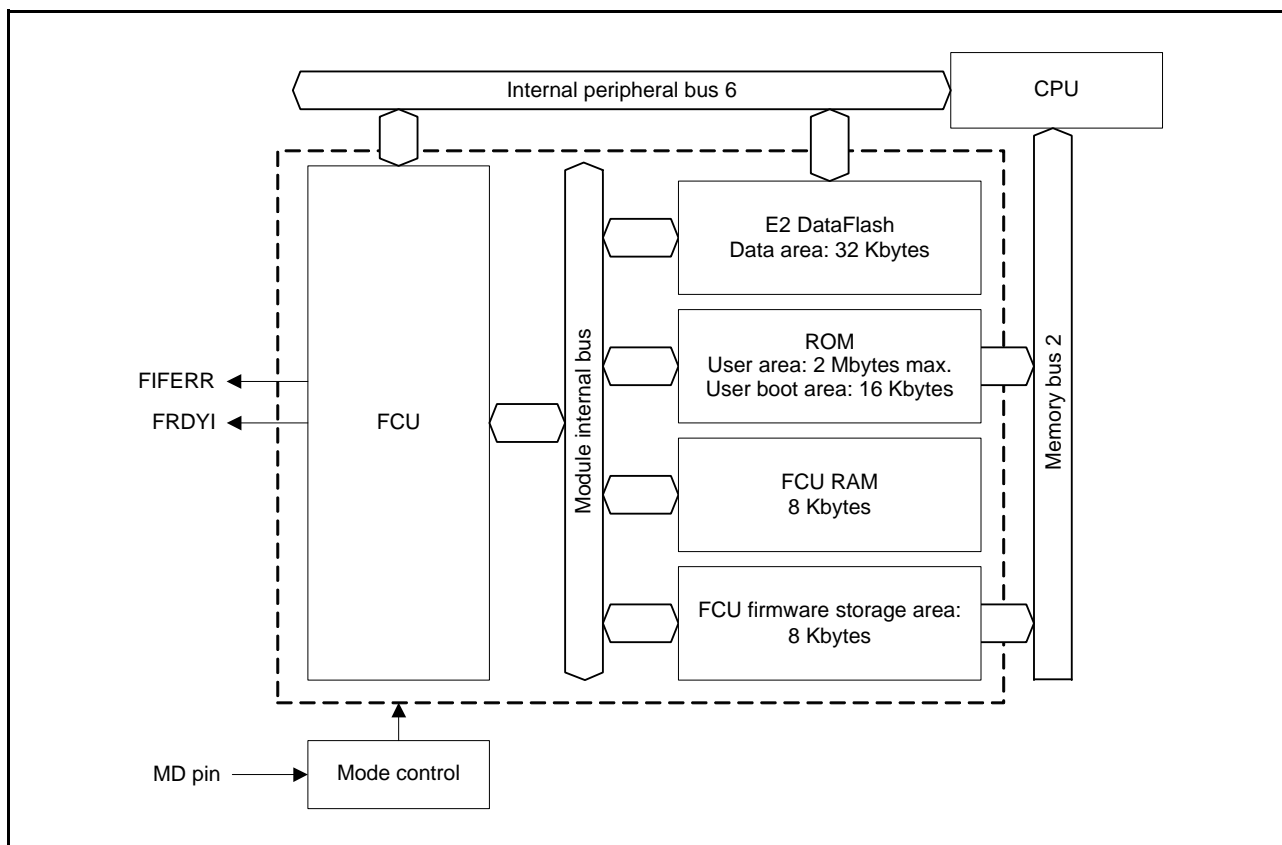


Figure 43.1 Block Diagram of ROM/E2 DataFlash

### 43.1.1 Configuration of the ROM Area

The ROM of products in the RX630 Group is configured of a maximum 2-Mbyte user area and a 16-Kbyte user boot area. Figure 43.2 shows the address ranges of these areas.

Note that for the user area, the address range for reading differs from the address range for programming and erasure.

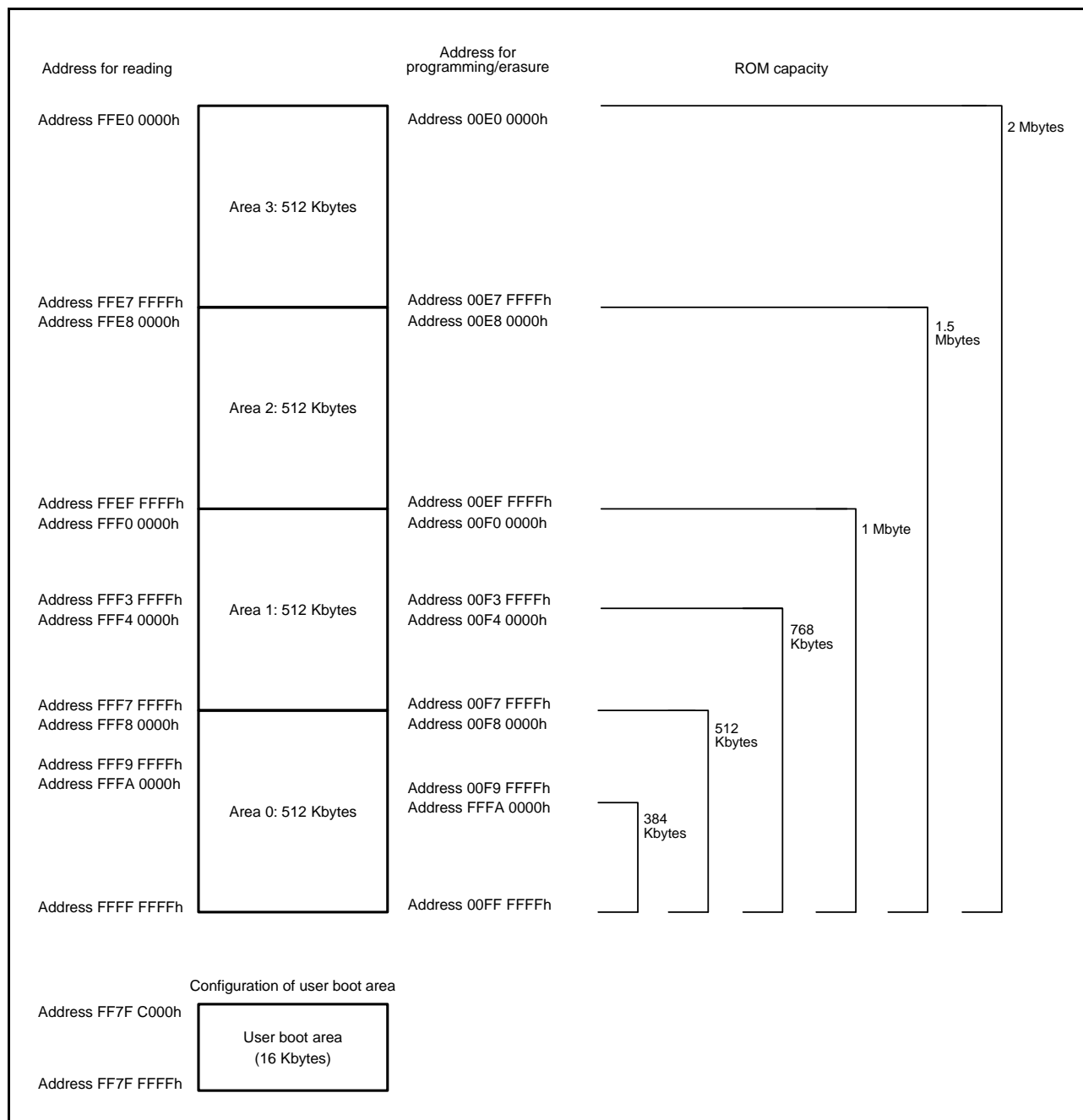


Figure 43.2 Memory Area Configuration of ROM

### 43.1.2 Block Configuration of the ROM

The user area (area 0 to area 3) is divided into blocks with different sizes, and erasure proceeds in block units. The configuration of the blocks of the user area is shown in Figure 43.3, and the relations between blocks and addresses of the user area are listed in Table 43.2.

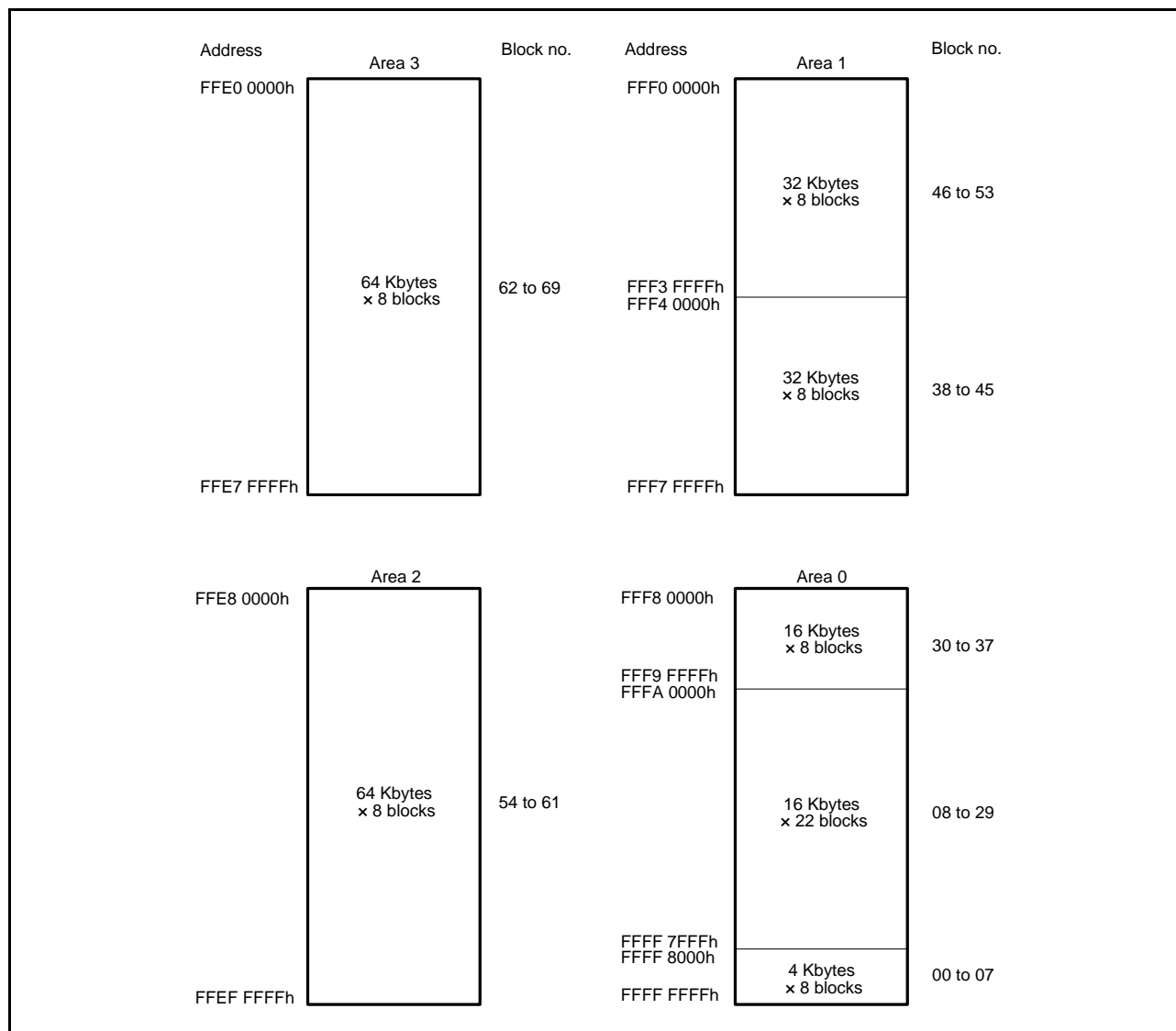


Figure 43.3 Block Configuration of the User Area

Table 43.2 Correspondence between Blocks and Addresses of the User Area

Block No.	Start Address	Block Configuration	Area	Block No.	Start Address	Block Configuration	Area		
69	FFE0 0000h	64 Kbyte x 8 blocks	Area 3	37	FFF8 0000h	16 Kbyte x 8 blocks	Area 0		
68	FFE1 0000h			36	FFF8 4000h				
67	FFE2 0000h			35	FFF8 8000h				
66	FFE3 0000h			34	FFF8 C000h				
65	FFE4 0000h			33	FFF9 0000h				
64	FFE5 0000h			32	FFF9 4000h				
63	FFE6 0000h			31	FFF9 8000h				
62	FFE7 0000h			30	FFF9 C000h				
61	FFE8 0000h	64 Kbyte x 8 blocks	Area 2	29	FFFA 0000h	16 Kbyte x 22 blocks			
60	FFE9 0000h			28	FFFA 4000h				
59	FFEA 0000h			27	FFFA 8000h				
58	FFEB 0000h			26	FFFA C000h				
57	FFEC 0000h			25	FFFB 0000h				
56	FFED 0000h			24	FFFB 4000h				
55	FFEE 0000h			23	FFFB 8000h				
54	FFEF 0000h			22	FFFB C000h				
53	FFF0 0000h	32 Kbyte x 8 blocks	Area 1	21	FFFC 0000h				
52	FFF0 8000h			20	FFFC 4000h				
51	FFF1 0000h			19	FFFC 8000h				
50	FFF1 8000h			18	FFFC C000h				
49	FFF2 0000h			17	FFFD 0000h				
48	FFF2 8000h			16	FFFD 4000h				
47	FFF3 0000h			15	FFFD 8000h				
46	FFF3 8000h			14	FFFD C000h				
45	FFF4 0000h	32 Kbyte x 8 blocks		13	FFFE 0000h	4 Kbyte x 8 blocks			
44	FFF4 8000h			12	FFFE 4000h				
43	FFF5 0000h			11	FFFE 8000h				
42	FFF5 8000h			10	FFFE C000h				
41	FFF6 0000h			9	FFFF 0000h				
40	FFF6 8000h			8	FFFF 4000h				
39	FFF7 0000h			7	FFFF 8000h				
38	FFF7 8000h			6	FFFF 9000h				
				5	FFFF A000h				
				4	FFFF B000h				
				3	FFFF C000h				
				2	FFFF D000h				
				1	FFFF E000h				
				0	FFFF F000h				

### 43.1.3 Configuration of the E2 DataFlash Area

The E2 DataFlash memory of products in the RX630 Group is configured as a 32-Kbyte data area whose addresses range from 0010 0000h to 0010 7FFFh.

Unlike the user area, the address ranges for reading from and programming and erasure of the data area are the same.

### 43.1.4 Block Configuration of the E2 DataFlash

The data area is divided into 1024 blocks, and erasure is executed in block units. The relations between blocks and addresses of the data area and the corresponding bits to support permission of reading and of programming and erasure are listed in Table 43.3.

The address of block N (address where block N starts) is calculated from the following formula.

Address of block N = (N x 32) + address where the data area starts (0010 0000h).

**Table 43.3 Block Configuration of the Data Area**

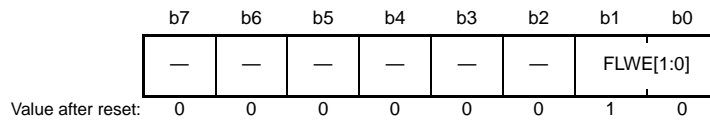
Block No.	Start Address	Reading and Programming/ Erasure Enable Bit	Block No.	Start Address	Reading and Programming/ Erasure Enable Bit
0000	0010 0000h	DFLRE0.DBRE00 DFLWE0.DBWE00	0512	0010 4000h	DFLRE1.DBRE08 DFLWE1.DBWE08
∴	∴		∴	∴	
0063	0010 07E0h	DFLRE0.DBRE01 DFLWE0.DBWE01	0575	0010 47E0h	DFLRE1.DBRE09 DFLWE1.DBWE09
0064	0010 0800h		∴	∴	
∴	∴	DFLRE0.DBRE02 DFLWE0.DBWE02	0639	0010 4FE0h	DFLRE1.DBRE10 DFLWE1.DBWE10
0127	0010 0FE0h		∴	∴	
0128	0010 1000h	DFLRE0.DBRE03 DFLWE0.DBWE03	0640	0010 5000h	DFLRE1.DBRE11 DFLWE1.DBWE11
∴	∴		∴	∴	
0191	0010 17E0h	DFLRE0.DBRE04 DFLWE0.DBWE04	0703	0010 57E0h	DFLRE1.DBRE12 DFLWE1.DBWE12
0192	0010 1800h		∴	∴	
∴	∴	DFLRE0.DBRE05 DFLWE0.DBWE05	0767	0010 5FE0h	DFLRE1.DBRE13 DFLWE1.DBWE13
0255	0010 1FE0h		∴	∴	
0256	0010 2000h	DFLRE0.DBRE06 DFLWE0.DBWE06	0768	0010 6000h	DFLRE1.DBRE14 DFLWE1.DBWE14
∴	∴		∴	∴	
0319	0010 27E0h	DFLRE0.DBRE07 DFLWE0.DBWE07	0831	0010 67E0h	DFLRE1.DBRE15 DFLWE1.DBWE15
0320	0010 2800h		∴	∴	
∴	∴	DFLRE0.DBRE08 DFLWE0.DBWE08	0832	0010 6800h	DFLRE1.DBRE15 DFLWE1.DBWE15
0383	0010 2FE0h		∴	∴	
0384	0010 3000h	DFLRE0.DBRE09 DFLWE0.DBWE09	0895	0010 6FE0h	DFLRE1.DBRE15 DFLWE1.DBWE15
∴	∴		∴	∴	
0447	0010 37E0h	DFLRE0.DBRE10 DFLWE0.DBWE10	0896	0010 7000h	DFLRE1.DBRE15 DFLWE1.DBWE15
0448	0010 3800h		∴	∴	
∴	∴	DFLRE0.DBRE11 DFLWE0.DBWE11	0959	0010 77E0h	DFLRE1.DBRE15 DFLWE1.DBWE15
0511	0010 3FE0h		∴	∴	
			1023	0010 7FE0h	

## 43.2 Register Descriptions

Some registers are common to the ROM and the E2 DataFlash, while others are dedicated to one or the other.

### 43.2.1 Flash Write Erase Protection Register (FWEPROR)

Address(es): 0008 C296h



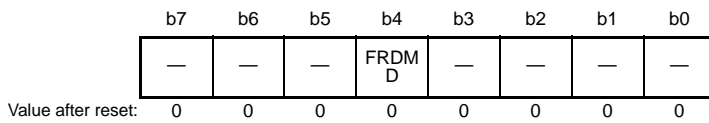
Bit	Symbol	Bit Name	Description	R/W
b1 to b0	FLWE[1:0]	Flash Programming/ Erasure	b1 b0 0 0: Disables programming and erasure of the ROM/E2 DataFlash, programming and erasure of lock bits, reading of lock bits, and blank checking 0 1: Enables programming and erasure of the ROM/E2 DataFlash, programming and erasure of lock bits, reading of lock bits, and blank checking 1 0: Disables programming and erasure of the ROM/E2 DataFlash, programming and erasure of lock bits, reading of lock bits, and blank checking 1 1: Disables programming and erasure of the ROM/E2 DataFlash, programming and erasure of lock bits, reading of lock bits, and blank checking	R/W
b7 to b2	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Programming and erasure of the ROM or E2 DataFlash memory, programming and erasure of lock bits, reading of lock bits, and blank checking by software are prohibited.

FWEPROR is initialized by a reset due to the signal on the RES# pin, by transitions to software standby and deep software standby, and by the power supply voltage falling below the threshold for detection.

### 43.2.2 Flash Mode Register (FMODR)

Address(es): 007F C402h



Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b4	FRDMD	FCU Read Method Select	This bit selects internal processing by the FCU when a 0x71 command is issued. For details, see “FRDMD Bit (FCU Read Method Select)”.	R/W
b7 to b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

Regarding the FMODR, when a 0x71 command is issued in relation to the FCU, this bit selects internal processing by the FCU. Internal processing by the FCU differs according to the address where the 0x71 command was issued. This register is common to the ROM and the E2 DataFlash. When the on-chip ROM is disabled, the data read from this register is 00h and writing is disabled.

#### FRDMD Bit (FCU Read Method Select)

Table 43.4 shows the relations between the status of the FRDMD bit and internal processing by the FCU when a 0x71 command is issued.

**Table 43.4 Correspondence between the FRDMD Bit and 0x71 Command**

Where the 0x71 Command is Issued	FRDMD Bit Status	Function	Internal Processing by the FCU
ROM	0	Lock bits are read using the memory area read method (lock bit read 1)	Transition to ROM lock bit read mode
	1	Lock bits are read using the register read method (lock bit read 2)	By going on to issue a 0xD0 command, the lock bit read 2 command is executed.
E2 DataFlash	0	Transition to E2 DataFlash lock bit read mode	The E2 DataFlash memory does not have lock bits. Accordingly, when the E2 DataFlash memory makes a transition to E2 DataFlash lock bit read mode, the values read from the E2 DataFlash area are undefined.
	1	Blank check command	By going on to issue a 0xD0 command, the blank check command is executed.



### 43.2.3 Flash Access Status Register (FASTAT)

Address(es): 007F C410h

	b7	b6	b5	b4	b3	b2	b1	b0
	ROMAE	—	—	CMDLK	DFLAE	—	DFLRPE	DFLWPE
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	DFLWPE	E2 DataFlash Programming/Erase Protection Violation Flag	0: Programming/erasure protection violation 1: No programming/erasure protection violation	R/(W) *1
b1	DFLRPE	E2 DataFlash Read Protection Violation Flag	0: Read protection violation 1: No read protection violation	R/(W) *1
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	DFLAE	E2 DataFlash Access Violation Flag	0: No E2 DataFlash access violation 1: E2 DataFlash access violation	R/(W) *1
b4	CMDLK	FCU Command Lock Flag	0: FCU accepts the command 1: FCU does not accept the command	R
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ROMAE	ROM Access Violation Flag	0: No ROM access violation 1: ROM access violation	R/(W) *1

Note 1. Writing to these bits is only possible when clearing the flag. In this case, write 0 after reading 1.

This register is common to the ROM and the E2 DataFlash. When the on-chip ROM is disabled, the data read from this register is 00h and writing is disabled.

When any bit from among the DFLWPE, DFLRPE, DFLAE, and ROMAE bits in FASTAT is set to 1, the FSTAT0.ILGLERR bit is set to 1, which places the FCU in the command-locked state.

#### DFLWPE Bit (E2 DataFlash Programming/Erase Protection Violation)

This bit is used to indicate whether or not the programming/erasure protection set by DFLWE<sub>y</sub> (y = 0, 1) is violated. When the DFLWPE bit is set to 1, the FSTAT0.ILGLERR bit is set to 1, and the CMDLK bit become 1 (command-locked state).

[Setting condition]

- A programming/erasure command is issued for an E2 DataFlash area for which programming or erasure is disabled by DFLWE<sub>y</sub> (y = 0, 1).

[Clearing condition]

- When 0 is written after reading 1

**DFLRPE Bit (E2 DataFlash Read Protection Violation Flag)**

This bit is used to indicate whether or not the reading protection set by DFLRE<sub>y</sub> (y = 0, 1) is violated.

When the DFLRPE bit is set to 1, the FSTATR0.ILGLERR bit is set to 1, and the CMDLK bit becomes 1 (command-locked state).

[Setting condition]

- An E2 DataFlash area for which reading is disabled by DFLRE<sub>y</sub> (y=0, 1) is read out.

[Clearing condition]

- When 0 is written after reading 1

**DFLAE Bit (E2 DataFlash Access Violation Flag)**

This bit indicates whether an E2 DataFlash access violation occurred.

When the DFLAE bit is set to 1, the ILGLERR bit in FSTATR0 is set to 1, and the CMDLK bit becomes 1 (command-locked state).

[Setting conditions]

- An E2 DataFlash area is read in E2 DataFlash P/E normal mode and when the FENTRYD bit in FENTRYR is set to 1.
- An FCU command is issued for an E2 DataFlash area when the FENTRYD bit is set to 0.
- An FCU command is issued for an E2 DataFlash area when any of bits FENTRYR.FENTRY<sub>n</sub> (n = 0 to 3) is set to 1, or the area is read out.

[Clearing condition]

- When 0 is written after reading 1

**CMDLK Bit (FCU Command Lock Flag)**

This bit is used to indicate whether the FCU can receive commands.

When any bit of the FFASTAT register is set to 1, the CMDLK bit is set to 1, and the FCU receives no commands (see section 43.6.2, Command-Locked State). To enable the FCU to receive commands, a status register clear command must be issued to the FCU after setting FFASTAT to 10h.

[Setting condition]

- The FCU detects an error, and the CMDLK bit is set to 1 (command-locked state).

[Clearing condition]

- A status register clear command is issued while the FFASTAT register is 10h.

**ROMAE Bit (ROM Access Violation Flag)**

This bit indicates whether a ROM access violation occurred.

When the ROMAE bit is set to 1, the FSTATR0.ILGLERR bit is set to 1, and the CMDLK bit becomes 1 (command-locked state).

[Setting conditions]

- A ROM programming/erasure address is read out when the FCU is in ROM P/E normal mode\*<sup>1</sup>

ROM Capacity	ROM programming/erasure address ranges			
	The FENTRY0 bit is 1	The FENTRY1 bit is 1	The FENTRY2 bit is 1	The FENTRY3 bit is 1
384 Kbytes	00FA 0000h to 00FF FFFFh	—	—	—
512 Kbytes	00F8 0000h to 00FF FFFFh	—	—	—
768 Kbytes	00F8 0000h to 00FF FFFFh	00F4 0000h to 00F7 FFFFh	—	—
1 Mbyte	00F8 0000h to 00FF FFFFh	00F0 0000h to 00F7 FFFFh	—	—
1.5 Mbytes	00F8 0000h to 00FF FFFFh	00F0 0000h to 00F7 FFFFh	00E8 0000h to 00EF FFFFh	—
2 Mbytes	00F8 0000h to 00FF FFFFh	00F0 0000h to 00F7 FFFFh	00E8 0000h to 00EF FFFFh	00E0 0000h to 00E7 FFFFh

- An FCU command is issued for a ROM programming/erasure address or the address is read out\*<sup>1</sup>

ROM Capacity	ROM programming/erasure address ranges			
	The FENTRY0 bit is 0	The FENTRY1 bit is 0	The FENTRY2 bit is 0	The FENTRY3 bit is 0
384 Kbytes	00FA 0000h to 00FF FFFFh	—	—	—
512 Kbytes	00F8 0000h to 00FF FFFFh	—	—	—
768 Kbytes	00F8 0000h to 00FF FFFFh	00F4 0000h to 00F7 FFFFh	—	—
1 Mbyte	00F8 0000h to 00FF FFFFh	00F0 0000h to 00F7 FFFFh	—	—
1.5 Mbytes	00F8 0000h to 00FF FFFFh	00F0 0000h to 00F7 FFFFh	00E8 0000h to 00EF FFFFh	—
2 Mbytes	00F8 0000h to 00FF FFFFh	00F0 0000h to 00F7 FFFFh	00E8 0000h to 00EF FFFFh	00E0 0000h to 00E7 FFFFh

Note 1. The FENTRY1, FENTRY2, and FENTRY3 bits are provided when the user area capacity exceeds 512 Kbytes, 1 Mbyte, and 1.5 Mbytes, respectively.

- A ROM-reading address is read out while FENTRYR has placed the ROM in ROM P/E mode

ROM Capacity	ROM-reading address ranges
384 Kbytes	FFFA 0000h to FFFF FFFFh
512 Kbytes	FFF8 0000h to FFFF FFFFh
768 Kbytes	FFF4 0000h to FFFF FFFFh
1 Mbyte	FFF0 0000h to FFFF FFFFh
1.5 Mbytes	FFE8 0000h to FFFF FFFFh
2 Mbytes	FFE0 0000h to FFFF FFFFh

[Clearing condition]

- When 0 is written after reading 1

### 43.2.4 Flash Access Error Interrupt Enable Register (FAEINT)

Address(es): 007F C411h

	b7	b6	b5	b4	b3	b2	b1	b0
	ROMAEIE	—	—	CMDLKIE	DFLAEIE	—	DFLRPEIE	DFLWPEIE
Value after reset:	1	0	0	1	1	0	1	1

Bit	Symbol	Bit Name	Description	R/W
b0	DFLWPEIE	E2 DataFlash Programming/Erase Protection Violation Interrupt Enable	This bit selects the generation or non-generation of an FIFERR interrupt in response to violations of protection against programming and erasure. 0: The interrupt is not generated. 1: The interrupt is generated.	R/W
b1	DFLRPEIE	E2 DataFlash Read Protection Violation Interrupt Enable	This bit selects the generation or non-generation of an FIFERR interrupt in response to violations of protection against reading. 0: The interrupt is not generated. 1: The interrupt is generated.	R/W
b2	—	Reserved	This bit is read as 0. The write value should be 0.	R/W
b3	DFLAEIE	E2 DataFlash Access Violation Interrupt Enable	This bit selects the generation or non-generation of an FIFERR interrupt in response to violations of access to the E2 DataFlash. 0: The interrupt is not generated. 1: The interrupt is generated.	R/W
b4	CMDLKIE	FCU Command Lock Interrupt Enable	This bit selects the generation or non-generation of an FIFERR interrupt when the FASTAT.CMDLK bit is set to 1 (command-locked state). 0: The interrupt is not generated. 1: The interrupt is generated.	R/W
b6, b5	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b7	ROMAEIE	ROM Access Violation Interrupt Enable	This bit selects the generation or non-generation of an FIFERR interrupt in response to violations of access to the ROM. 0: The interrupt is not generated. 1: The interrupt is generated.	R/W

This register is common to the ROM and the E2 DataFlash. When the on-chip ROM is disabled, the data read from this register is 00h and writing is disabled.

#### DFLWPEIE Bit (E2 DataFlash Programming/Erase Protection Violation Interrupt Enable)

This bit is used to enable or disable FIFERR interrupt requests when an E2 DataFlash programming/erase protection violation occurs and the DFLWPE bit in FASTAT is set to 1.

#### DFLRPEIE Bit (E2 DataFlash Read Protection Violation Interrupt Enable)

This bit is used to enable or disable FIFERR interrupt requests when an E2 DataFlash read protection violation occurs and the DFLRPE bit in FASTAT is set to 1.

#### DFLAEIE Bit (E2 DataFlash Access Violation Interrupt Enable)

This bit is used to enable or disable FIFERR interrupt requests when an E2 DataFlash access violation occurs and the DFLAE bit in FASTAT is set to 1.

**CMDLKIE Bit (FCU Command Lock Interrupt Enable)**

This bit is used to enable or disable FIFERR interrupt requests when the FASTAT.CMDLK bit is set to 1 (command-locked state).

**ROMAEIE Bit (ROM Access Violation Interrupt Enable)**

This bit is used to enable or disable FIFERR interrupt requests when a ROM access violation occurs and the FASTAT.ROMAE bit is set to 1.

**43.2.5 Flash Ready Interrupt Enable Register (FRDYIE)**

Address(es): 007F C412h

b7	b6	b5	b4	b3	b2	b1	b0
—	—	—	—	—	—	—	FRDYI E
0	0	0	0	0	0	0	0

Value after reset:

Bit	Symbol	Bit Name	Description	R/W
b0	FRDYIE	Flash Ready Interrupt Enable	0: FRDYI interrupt requests disabled 1: FRDYI interrupt requests enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

This register is common to the ROM and the E2 DataFlash. When the on-chip ROM is disabled, the data read from this register is 00h and writing is disabled.

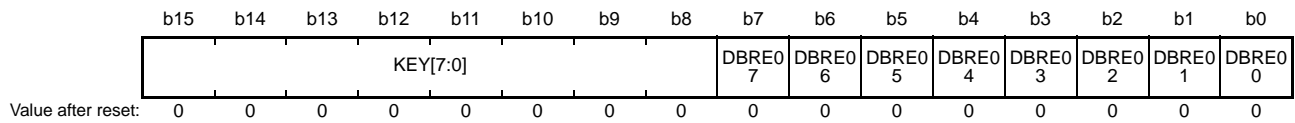
**FRDYIE Bit (Flash Ready Interrupt Enable)**

This bit is to enable or disable a flash ready interrupt request when programming/erasure is completed.

If the FRDYIE bit is set to 1, a flash ready interrupt request (FRDYI) is generated when execution of the FCU command has completed (FSTATR0.FRDY bit changes from 0 to 1).

### 43.2.6 E2 DataFlash Read Enable Register 0 (DFLRE0)

Address(es): 007F C440h



Bit	Symbol	Bit Name	Description	R/W
b0	DBRE00	0000-0063 Block Read Enable	0: Read disabled 1: Read enabled	R/W
b1	DBRE01	0064-0127 Block Read Enable		R/W
b2	DBRE02	0128-0191 Block Read Enable		R/W
b3	DBRE03	0192-0255 Block Read Enable		R/W
b4	DBRE04	0256-0319 Block Read Enable		R/W
b5	DBRE05	0320-0383 Block Read Enable		R/W
b6	DBRE06	0384-0447 Block Read Enable		R/W
b7	DBRE07	0448-0511 Block Read Enable		R/W
b15 to b8	KEY[7:0]	Key Code	These bits control permission and prohibition of writing to the DFLRE0 register. To modify the DFLRE0 register, write 2Dh to the eight higher-order bits and the desired value to the eight lower-order bits as a 16-bit unit.	R/(W) *1

Note 1. Write data is not retained.

DFLRE0 is a register to enable or disable the 0000 to 0511 blocks of the data area (see Table 43.3) to be read. Reading is enabled or disabled in 2 Kbytes (64 blocks).

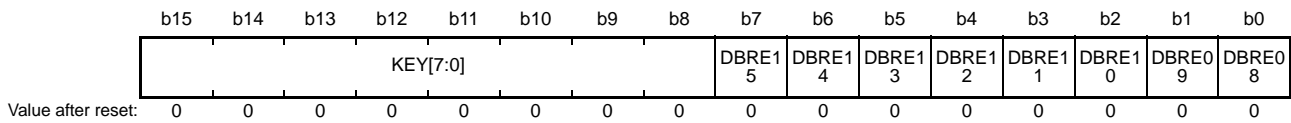
This register is dedicated to the E2 DataFlash. When the on-chip ROM is disabled, the data read from this register is 0000h and writing is disabled.

#### DBREj Bit (DBj Block Read Enable) (j = 00 to 07)

This bit is used to enable or disable the DB0000 to DB0511 blocks of the data area to be read.

### 43.2.7 E2 DataFlash Read Enable Register 1 (DFLRE1)

Address(es): 007F C442h



Bit	Symbol	Bit Name	Description	R/W
b0	DBRE08	0512-0575 Block Read Enable	0: Read disabled	R/W
b1	DBRE09	0576-0639 Block Read Enable	1: Read enabled	R/W
b2	DBRE10	0640-0703 Block Read Enable		R/W
b3	DBRE11	0704-0767 Block Read Enable		R/W
b4	DBRE12	0768-0831 Block Read Enable		R/W
b5	DBRE13	0832-0895 Block Read Enable		R/W
b6	DBRE14	0896-0959 Block Read Enable		R/W
b7	DBRE15	0960-1023 Block Read Enable		R/W
b15 to b8	KEY[7:0]	Key Code	These bits control permission and prohibition of writing to the DFLRE1 register. To modify the DFLRE1 register, write D2h to the eight higher-order bits and the desired value to the eight lower-order bits as a 16-bit unit.	R/(W) *1

Note 1. Write data is not retained.

DFLRE1 is a register to enable or disable the 0512 to 1023 blocks of the data area (see Table 43.3) to be read. Reading is enabled or disabled in 2 Kbytes (64 blocks).

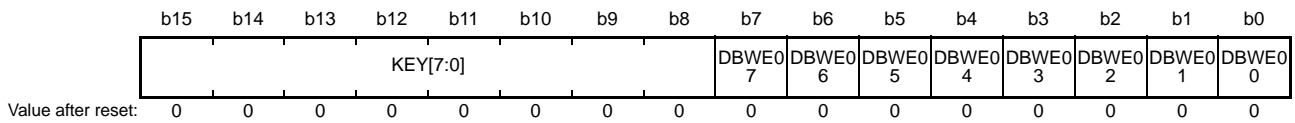
This register is dedicated to the E2 DataFlash. When the on-chip ROM is disabled, the data read from this register is 0000h and writing is disabled.

#### DBREj Bit (Block Read Enable) (j = 08 to 15)

This bit is used to enable or disable the 0512 to 1023 blocks of the data area to be read.

### 43.2.8 E2 DataFlash P/E Enable Register 0 (DFLWE0)

Address(es): 007F C450h



Bit	Symbol	Bit Name	Description	R/W
b0	DBWE00	0000-0063 Block Programming/ Erasure Enable	0: Programming/erasure disabled 1: Programming/erasure enabled	R/W
b1	DBWE01	0064-0127 Block Programming/ Erasure Enable		R/W
b2	DBWE02	0128-0191 Block Programming/ Erasure Enable		R/W
b3	DBWE03	0192-0255 Block Programming/ Erasure Enable		R/W
b4	DBWE04	0256-0319 Block Programming/ Erasure Enable		R/W
b5	DBWE05	0320-0383 Block Programming/ Erasure Enable		R/W
b6	DBWE06	0384-0447 Block Programming/ Erasure Enable		R/W
b7	DBWE07	0448-0511 Block Programming/ Erasure Enable		R/W
b15 to b8	KEY[7:0]	Key Code	These bits control permission and prohibition of writing to the DFLWE0 register. To modify the DFLWE0 register, write 1Eh to the eight higher-order bits and the desired value to the eight lower-order bits as a 16-bit unit.	R/(W) *1

Note 1. Write data is not retained.

DFLWE0 is a register to enable or disable the 0000 to 0511 blocks of the data area (see Table 43.3) to be programmed or erased. Programming or erasing is enabled or disabled in 2 Kbytes (64 blocks).

This register is dedicated to the E2 DataFlash. When the on-chip ROM is disabled, the data read from this register is 0000h and writing is disabled.

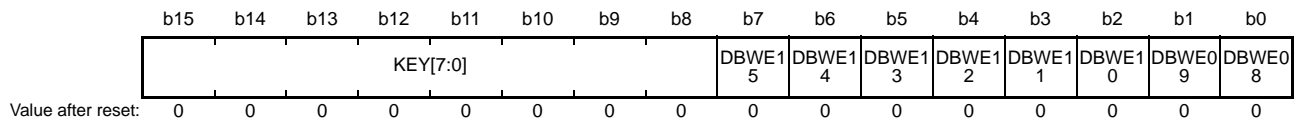
#### DBWE<sub>j</sub> Bit (Block Programming/Erasure Enable) (j = 00 to 07)

This bit is used to enable or disable the 0000 to 0511 blocks of the data area to be programmed or erased.



### 43.2.9 E2 DataFlash P/E Enable Register 1 (DFLWE1)

Address(es): 007F C452h



Bit	Symbol	Bit Name	Description	R/W
b0	DBWE08	0512-0575 Block Programming/ Erasure Enable	0: Programming/erasure disabled 1: Programming/erasure enabled	R/W
b1	DBWE09	0576-0639 Block Programming/ Erasure Enable		R/W
b2	DBWE10	0640-0703 Block Programming/ Erasure Enable		R/W
b3	DBWE11	0704-0767 Block Programming/ Erasure Enable		R/W
b4	DBWE12	0768-0831 Block Programming/ Erasure Enable		R/W
b5	DBWE13	0832-0895 Block Programming/ Erasure Enable		R/W
b6	DBWE14	0896-0959 Block Programming/ Erasure Enable		R/W
b7	DBWE15	0960-1023 Block Programming/ Erasure Enable		R/W
b15 to b8	KEY[7:0]	Key Code	These bits control permission and prohibition of writing to the DFLWE1 register. To modify the DFLWE1 register, write E1h to the eight higher-order bits and the desired value to the eight lower-order bits as a 16-bit unit.	R/(W) *1

Note 1. Write data is not retained.

DFLWE1 is a register to enable or disable the 0512 to 1023 blocks of the data area (see Table 43.3) to be programmed or erased. Programming or erasing is enabled or disabled in 2 Kbytes (64 blocks).

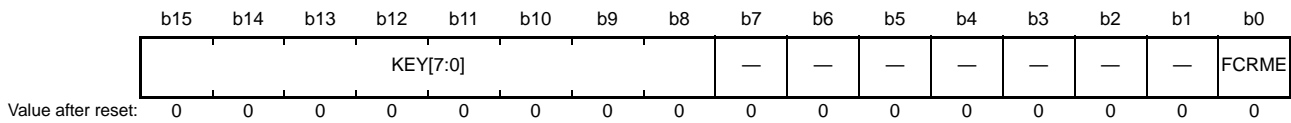
This register is dedicated to the E2 DataFlash. When the on-chip ROM is disabled, the data read from this register is 0000h and writing is disabled.

#### DBWE<sub>j</sub> Bit (Block Programming/Erasure Enable) (j = 08 to 15)

This bit is used to enable or disable the 0512 to 1023 blocks of the data area to be programmed or erased.

### 43.2.10 FCU RAM Enable Register (FCURAME)

Address(es): 007F C454h



Bit	Symbol	Bit Name	Description	R/W
b0	FCRME	FCU RAM Enable	0: Access to the FCU RAM disabled 1: Access to the FCU RAM enabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	KEY[7:0]	Key Code	These bits control permission and prohibition of writing to the FCURAME register. To modify the FCURAME register, write C4h to the eight higher-order bits and the desired value to the eight lower-order bits as a 16-bit unit.	R/(W) *1

Note 1. Write data is not retained.

This register is common to the ROM and the E2 DataFlash. When the on-chip ROM is disabled, the data read from this register is 0000h and writing is disabled.

#### FCRME Bit (FCU RAM Enable)

This bit is used to enable or disable access to the FCU RAM.

When writing to the FCU RAM, set FENTRYR to 0000h and stop the FCU.

Furthermore, data in FCU RAM cannot be read regardless of whether access to FCU RAM is enabled or disabled. Values read when reading is attempted are undefined.

### 43.2.11 Flash Status Register 0 (FSTATR0)

Address(es): 007F FFB0h

	b7	b6	b5	b4	b3	b2	b1	b0
	FRDY	ILGLERR	ERSERR	PRGERR	SUSRDY	—	ERSSPD	PRGSPD
Value after reset:	1	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b0	PRGSPD	Programming Suspend Status Flag	0: Other than the status described below 1: During programming suspend processing or programming suspended	R
b1	ERSSPD	Erase Suspend Status Flag	0: Other than the status described below 1: When erasure suspend processing or erasure suspended	R
b2	—	Reserved	This bit is read as 0 and cannot be modified.	R
b3	SUSRDY	Suspend Ready Flag	0: P/E suspend commands cannot be received 1: P/E suspend commands can be received	R
b4	PRGERR	Programming Error Flag	0: Programming terminates normally 1: An error occurs during programming	R
b5	ERSERR	Erase Error	0: Erasure terminates normally 1: An error occurs during erasure	R
b6	ILGLERR	Illegal Command Error Flag	0: FCU detects no illegal command or illegal ROM/E2 DataFlash access 1: FCU detects an illegal command or illegal ROM/E2 DataFlash access	R
b7	FRDY	Flash Ready Flag	0: During programming/erasure, During suspending programming/erasure, During the lock bit read 2 command processing, During the peripheral clock notification command processing, During the blank check processing of E2 DataFlash. 1: Processing described above is not performed	R

FSTATR0 is also reset by setting the FRESETR.FRESET bit to 1.

When the on-chip ROM is disabled, the data read from this register is 00h and writing is disabled. This register is common to the ROM and the E2 DataFlash.

#### PRGSPD Bit (Programming Suspend Status Flag)

This bit is used to indicate that the FCU enters the programming suspend processing state or programming suspended state. For details, see section 43.5, Suspending Operation.

[Setting condition]

- The FCU has initiated a write suspend command.

[Clearing condition]

- The FCU has accepted a resume command.

**ERSSPD Bit (Erasure Suspend Status Flag)**

This bit is used to indicate that the FCU enters the erasure suspend processing state or erasure suspended state. For details, see section 43.5, Suspending Operation.

[Setting condition]

- The FCU has initiated an erasure suspend command.

[Clearing condition]

- The FCU has accepted a resume command.

**SUSRDY Bit (Suspend Ready Flag)**

This bit is used to indicate whether the FCU can receive a P/E suspend command.

[Setting condition]

- After starting programming/erasure process, the FCU enters a state in which P/E suspend commands can be received.

[Clearing conditions]

- The FCU has accepted a P/E suspend command.
- During programming/erasure process, the FCU enters the command-locked state.

**PRGERR Bit (Programming Error Flag)**

This bit is used to indicate the result of the ROM/E2 DataFlash programming process by the FCU. When the PRGERR bit is set to 1, the FASTAT.CMDLK bit becomes 1 (command-locked state).

[Setting conditions]

- An error occurs during programming.
- A programming command is issued to areas protected by a lock bit.

[Clearing condition]

- After the FCU processes a status register clear command

**ERSERR Bit (Erasure Error Flag)**

This bit is used to indicate the result of the ROM/E2 DataFlash erasure process by the FCU. When the ERSERR bit is set to 1, the FASTAT.CMDLK bit becomes 1 (command-locked state).

[Setting conditions]

- An error occurs during erasure.
- A block erase command is issued to areas protected by a lock bit.

[Clearing condition]

- After the FCU processes a status register clear command

**ILGLERR Bit (Illegal Command Error Flag)**

This bit is used to indicate that the FCU detects any illegal command or ROM/E2 DataFlash access. When the ILGLERR bit is set to 1, the FASTAT.CMDLK bit becomes 1 (command-locked state).

[Setting conditions]

- The FCU detects an illegal command.
- The FCU detects an illegal ROM/E2 DataFlash access (one of the ROMAЕ, DFLAЕ, DFLRPE, and DFLWPE bits in FASTAT is 1).
- The setting of FENTRYR is invalid.

[Clearing condition]

- After a status register clear command has been issued under conditions where FASTAT is set to 10h

**43.2.12 Flash Status Register 1 (FSTATR1)**

Address(es): 007F FFB1h

	b7	b6	b5	b4	b3	b2	b1	b0
	FCUER R	—	—	FLOCK ST	—	—	—	—
Value after reset:	0	0	0	0	0	0	0	0

Bit	Symbol	Bit Name	Description	R/W
b3 to b0	—	Reserved	These bits are read as 0 and cannot be modified.	R
b4	FLOCKST	Lock Bit Status	0: Protected 1: Not protected	R
b6, b5	—	Reserved	These bits are read as 0 and cannot be modified.	R
b7	FCUERR	FCU Error Flag	0: No error occurs in the FCU processing 1: An error occurs in the FCU processing	R

FSTATR1 is also reset by setting the FRESETR.FRESET bit to 1. When the on-chip ROM is disabled, the data read from this register is 00h and writing is disabled. This register is common to the ROM and the E2 DataFlash.

**FLOCKST Bit (Lock Bit Status)**

This bit is to reflect the read data of a lock bit when using the lock bit read 2 command.

When the FSTATR0.FRДY bit is set to 1 after a lock bit read 2 command is issued, the value of the lock bit status is stored in the FLOCKST bit. The value of the FLOCKST bit is retained until the completion of the next lock bit read 2 command.

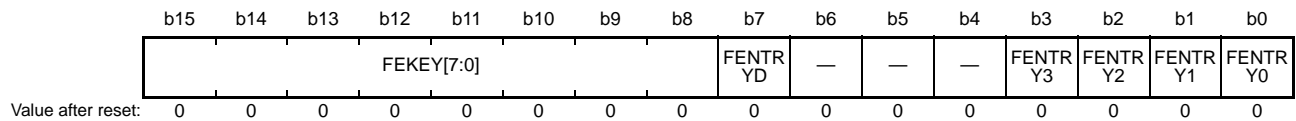
**FCUERR Bit (FCU Error Flag)**

This bit is used to indicate that an error occurs in the FCU internal processing.

When the FCUERR bit is set to 1, set the FRESETR.FRESET bit to 1 to initialize the FCU. Additionally, recopy the FCU firmware from the FCU firmware area to the FCU RAM area.

## 43.2.13 Flash P/E Mode Entry Register (FENTRYR)

Address(es): 007F FFB2h



Bit	Symbol	Bit Name	Description	R/W
b0	FENTRY0	ROM P/E Mode Entry 0	0: Area 0 is in ROM read mode. 1: Area 0 is in ROM P/E mode.	R/W
b1	FENTRY1	ROM P/E Mode Entry 1*1	0: Area 1 is in ROM read mode. 1: Area 1 is in ROM P/E mode.	R/W
b2	FENTRY2	ROM P/E Mode Entry 2*1	0: Area 2 is in ROM read mode. 1: Area 2 is in ROM P/E mode.	R/W
b3	FENTRY3	ROM P/E Mode Entry 3*1	0: Area 3 is in ROM read mode. 1: Area 3 is in ROM P/E mode.	R/W
b6 to b4	—	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 is in P/E mode.	R/W
b15 to b8	FEKEY[7:0]	Key Code	These bits control permission and prohibition of writing to the FENTRYR register. To modify the FENTRYR register, write AAh to the eight higher-order bits and the desired value to the eight lower-order bits as a 16-bit unit.	R/(W) *1

Note 1. Write data is not retained.

To place the ROM/E2 DataFlash in ROM P/E mode so that the FCU can accept commands, either the FENTRYD or FENTRY<sub>n</sub> (n = 0 to 3) bits must be set to 1. Note that if a value is set other than AA01h, AA02h, AA04h, AA08h, and AA80h in FENTRYR, the FSTATR0.ILGLERR bit is set to 1 and the FSATAT.CMDLK bit is set to 1 (command-locked state).

The ROM exists from area 0 up to area 3 at the maximum, and the FENTRY0 to FENTRY3 bits correspond to the respective areas. The FENTRY<sub>n</sub> (n = 0 to 3) bits for areas that are not present cannot be set to 1. See Figure 43.2, Memory Area Configuration of ROM, for the relation between the ROM capacity and areas.

FENTRYR is also reset when the FRESETR.FRESET bit is set to 1. When on-chip ROM is disabled, the data read from FENTRYR is 0000h and writing is disabled. This register is common to the ROM and the E2 DataFlash.

**FENTRYn Bit (ROM P/E Mode Entry n: n = 0 to 3)**

This bit is used to place area n (n = 0 to 3) in P/E mode.

[Writing-enable conditions (when all of the following conditions are met)]

- On-chip ROM is enabled.
- The FSTATR0.FRDY bit is set to 1.
- AAh is written to the FEKEY[7:0] bits in word access.

[Setting condition]

- The writing-enable conditions are met, FENTRYR is set to 0000h, and 1 is written to the FENTRYn (n = 0 to 3) bit.

[Clearing conditions]

- Data is written in byte access.
- Data is written in word access when the FEKEY[7:0] bits are other than AAh.
- When the writing-enable conditions are met, 0 is written to the FENTRYn (n = 0 to 3) bit.
- When the writing-enable conditions are met and FENTRYR is other than 0000h, data is written to FENTRYR.

**FENTRYD Bit (E2 DataFlash P/E Mode Entry)**

The FENTRYD bit is used to place the E2 DataFlash in P/E mode.

[Writing-enable conditions (when all of the following conditions are met)]

- On-chip ROM is enabled.
- The FRDY bit in FSTATR0 is set to 1
- AAh is written to the FEKEY[7:0] bits in word access.

[Setting condition]

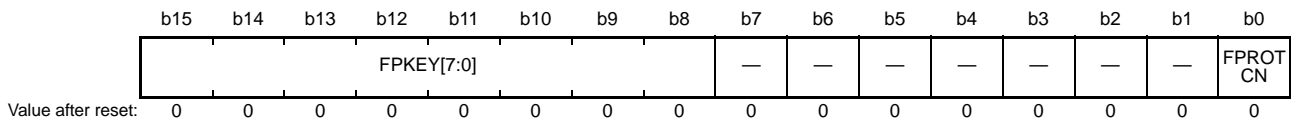
- When the writing-enable conditions are met, FENTRYR is set to 0000h, and 1 is written to the FENTRYD bit

[Clearing conditions]

- Data is written in byte access.
- Data is written in word access when the FEKEY[7:0] bits are other than AAh.
- When the writing-enable conditions are met, 0 is written to the FENTRYD bit.
- When the writing-enable conditions are met and FENTRYR is other than 0000h, data is written to FENTRYR.

### 43.2.14 Flash Protection Register (FPROTR)

Address(es): 007F FFB4h



Bit	Symbol	Bit Name	Description	R/W
b0	FPROTCN	Lock Bit Protection Cancel	0: Protection with a lock bit enabled 1: Protection with a lock bit disabled	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	FPKEY[7:0]	Key Code	These bits control permission and prohibition of writing to the FPROTR register. To modify the FPROTR register, write 55h to the eight higher-order bits and the desired value to the eight lower-order bits as a 16-bit unit.	R/(W) *1

Note 1. Write data is not retained.

FPROTR is also reset when the FRESETR.FRESET bit is set to 1.

When on-chip ROM is disabled, the data read from FPROTR is 0000h and writing is disabled. This register is dedicated to the ROM.

#### FPROTCN Bit (Lock Bit Protection Cancel)

This bit is used to enable/disable the programming/erasure protection with a lock bit.

[Setting condition]

- 55h is written to the FPKEY[7:0] bits and 1 is written to the FPROTCN bit in word access when the value of FENTRYR is other than 0000h.

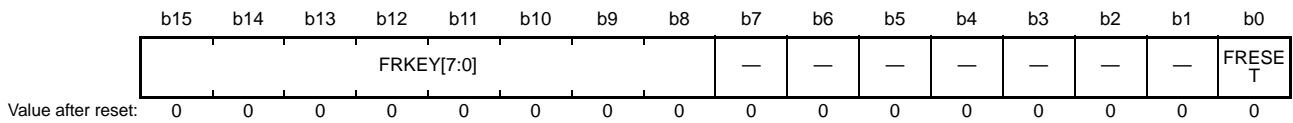
[Clearing conditions]

- Data is written in byte access.
- Data is written in word access when the FPKEY[7:0] bits are other than 55h.
- 55h is written to the FPKEY[7:0] bits and 0 is written to the FPROTCN bit in word access.
- The value of FENTRYR is 0000h.



### 43.2.15 Flash Reset Register (FRESETR)

Address(es): 007F FFB6h



Bit	Symbol	Bit Name	Description	R/W
b0	FRESET	Flash Reset	0: FCU is not reset 1: FCU is reset	R/W
b7 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15 to b8	FRKEY[7:0]	Key Code	These bits control permission and prohibition of writing to the FRESETR register. To modify the FRESETR register, write CCh to the eight higher-order bits and the desired value to the eight lower-order bits as a 16-bit unit.	R/(W) *1

Note 1. Write data is not retained.

When on-chip ROM is disabled, the data read from FRESETR is 0000h and writing is disabled. This register is common to the ROM and the E2 DataFlash.

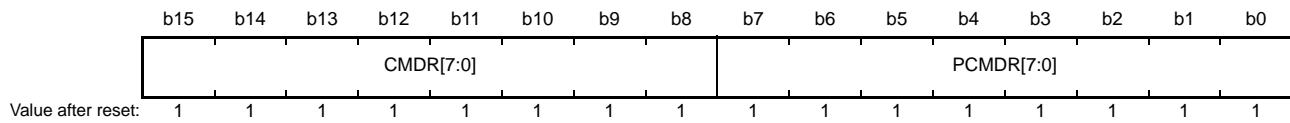
#### FRESET Bit (Flash Reset)

When the FRESET bit is set to 1, programming/erasure operations for the ROM/E2 DataFlash are forcibly terminated, and the FCU is initialized.

High voltage is applied to the memory of the ROM/E2 DataFlash during programming/erasure. To ensure the time required for dropping the voltage applied to the memory, keep the FRESET bit set to 1 for tFCUR (FCU reset time, see section 45, Electrical Characteristics) when initializing the FCU. While the FRESET bit is kept as 1, prohibit the ROM/E2 DataFlash from being read. Additionally, when the FRESET bit is set to 1, the FCU commands cannot be used because FENTRYR is initialized.

### 43.2.16 FCU Command Register (FCMDR)

Address(es): 007F FFBAh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	PCMDR[7:0]	Precommand	Store the command immediately before the last command received by the FCU.	R
b15 to b8	CMDR[7:0]	Command	Store the last command received by the FCU.	R

FCMDR is also initialized when the FRESETR.FRESET bit is set to 1.

Table 43.5 lists the states of FCMDR after receiving each command.

When on-chip ROM is disabled, data read from FCMDR is 0000h and writing is disabled. This register is common to the ROM and the E2 DataFlash.

**Table 43.5 States of FCMDR after Receiving Each Command**

Command	CMDR[7:0]	PCMDR[7:0]
Normal mode transition	FFh	Previous command
Status read mode transition	70h	Previous command
Lock bit read mode transition (lock bit read 1)	71h	Previous command
Peripheral clock notification	E9h	Previous command
Programming	E8h	Previous command
Block erase	D0h	20h
P/E suspend	B0h	Previous command
P/E resume	D0h	Previous command
Status register clear	50h	Previous command
Lock bit read 2/blank check	D0h	71h
Lock bit programming	D0h	77h

### 43.2.17 FCU Processing Switching Register (FCPSR)

Address(es): 007F FFC8h

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	ESUSPMD
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	ESUSPMD	Erase Suspend Mode	0: Suspension priority mode 1: Erasure priority mode	R/W
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

FCPSR is also reset when the FRESETR.FRESET bit is set to 1.

When on-chip ROM is disabled, the data read from FCPSR is 0000h and writing is disabled. This register is common to the ROM and the E2 DataFlash.

#### ESUSPMD Bit (Erasure Suspend Mode)

This bit is to select the erasure suspend mode for when a P/E suspend command is issued while the FCU executes the erasure processing for the ROM/E2 DataFlash. For details, see section 43.5, Suspending Operation.

### 43.2.18 E2 DataFlash Blank Check Control Register (DFLBCCNT)

Address(es): 007F FFCAh

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
	BCSIZE	—	—	—	—	BCADR[10:0]										—
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
b10 to b0	BCADR[10:0]	Blank Check Address Setting	Set the address of the area to be checked	R/W
b14 to b11	—	Reserved	These bits are read as 0. The write value should be 0.	R/W
b15	BCSIZE	Blank Check Size Setting	0: The size of the area to be blank checked is 2 bytes. 1: The size of the area to be blank checked is 2 Kbytes.	R/W

DFLBCCNT is also reset when the FRESETR.FRESET bit is set to 1.

When on-chip ROM is disabled, the data read from DFLBCCNT is 0000h and writing is disabled. This register is dedicated to the E2 DataFlash.

#### BCADR[10:0] Bits (Blank Check Address Setting)

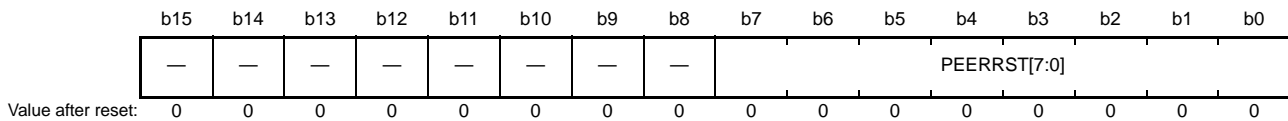
These bits are used to set the address of the area to be checked when the size of the area to be checked by a blank check command is 2 bytes (the BCSIZE bit is 0). Set the BCADR[0] bit to 0.

When the BCSIZE bit is 0, the start address of the area to be checked is obtained by adding the DFLBCCNT setting value to the block start address (in 2-Kbyte units) specified at issuance of a blank check command.

When the BCSIZE bit is 1, the setting of the BCADR[10:0] bits will be ignored.

### 43.2.19 Flash P/E Status Register (FPESTAT)

Address(es): 007F FCCh



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	PEERRST[7:0]	P/E Error Status	00h: No error 01h: Programming error against areas protected by a lock bit 02h: Programming error due to sources other than the lock bit protection 11h: Erasure error against areas protected by a lock bit 12h: Erasure error due to sources other than the lock bit protection (Values other than above are reserved)	R
b15 to b8	—	Reserved	These bits are read as 0 and cannot be modified.	R

FPESTAT is also reset when the FRESETR.FRESET bit is set to 1.

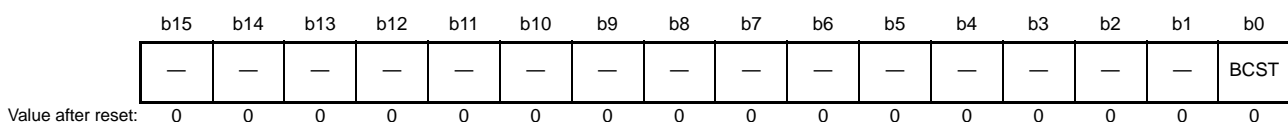
When on-chip ROM is disabled, the data read from FPESTAT is 0000h and writing is disabled. This register is dedicated to the ROM.

#### PEERRST[7:0] Bits (P/E Error Status)

These bits are used to indicate the reason of an error that occurs during the programming/erasure processing for the ROM. The value of the PEERRST[7:0] bits is valid only when the FSTATR0.FR DY bit is set to 1 while the FSTATR0.ERSERR bit or FSTATR0.PRGERR bit is 1. The value of the reason of the past error is retained in the PEERRST[7:0] bits when the ERSERR bit and PRGERR bit is 0.

### 43.2.20 E2 DataFlash Blank Check Status Register (DFLBCSTAT)

Address(es): 007F FCCh



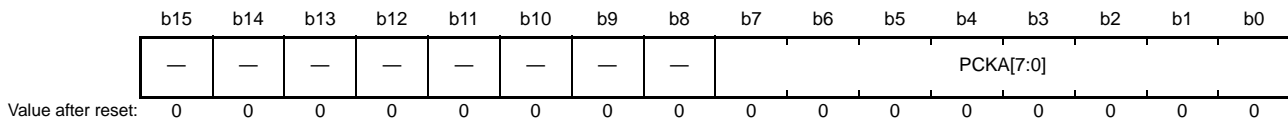
Bit	Symbol	Bit Name	Description	R/W
b0	BCST	Blank Check Status	0: The area to be blank-checked is erased (blank) 1: 0 or 1 is written in the area to be blank-checked	R
b15 to b1	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

DFLBCSTAT is also reset when the FRESETR.FRESET bit is set to 1.

When on-chip ROM is disabled, the data read from DFLBCSTAT is 0000h and writing is disabled. This register is dedicated to the E2 DataFlash.

### 43.2.21 Peripheral Clock Notification Register (PCKAR)

Address(es): 007F FFE8h



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	PCKA[7:0]	Peripheral Clock Notification	These bits are used to set the FlashIF clock (FCLK) at the programming/erasure for the ROM/E2 DataFlash.	R/W
b15 to b8	—	Reserved	These bits are read as 0. The write value should be 0.	R/W

PCKAR is also reset when the FRESETR.FRESET bit is set to 1.

When on-chip ROM is disabled, the data read from PCKAR is 0000h and writing is disabled. This register is common to the ROM and the E2 DataFlash.

#### PCKA[7:0] Bits (Peripheral Clock Notification)

These bits are used to set the FlashIF clock (FCLK) at the programming/erasure for the ROM/E2 DataFlash.

Set the FCLK frequency in the PCKA[7:0] bits and issue a peripheral clock notification command before programming/erasure. Do not change the frequency during programming/erasure for the ROM/E2 DataFlash.

Write a setting to the PCKA[7:0] bits as a binary value that selects the operating frequency in MHz units.

For example, when the operating frequency of the FlashIF clock is 35.9 MHz, the setting value is calculated as follows:

- Round 35.9 off to a whole number.
- Convert 36 to binary and set the upper bits and lower bits of the PCKA[7:0] bits to 00h and 24h (0010 0100b).

Note 1. When the PCKA[7:0] bits are set to values outside the range from 4 to 50 MHz, do not issue a programming command to the ROM/E2 DataFlash.

Note 2. When the PCKA[7:0] bits are set to a frequency that is different from the FCLK, the data of the ROM/E2 DataFlash may be damaged.

Note 3. Note that the programming time depends on the frequency to some extent even if the PCKA[7:0] bits are used.

### 43.3 Operating Modes Associated with Flash Memory

For information on the relationship between the setting of the level on the MD pin and the operating mode for the RX630 Group, refer to section 3, Operating Modes.

The ROM and E2 DataFlash can be read, programmed, and erased on board in boot mode, USB boot mode, user boot mode, single-chip mode (with on-chip ROM enabled), or on-chip ROM enabled extended mode.

The area where programming and erasure are permitted, the area from which booting up proceeds, and areas erased at the time of booting up differ with the mode. The differences between modes are indicated in Table 43.6.

**Table 43.6 Differences between Modes**

Item	Boot Mode	USB Boot Mode	User Boot Mode	Single-Chip Mode (with On-Chip ROM Enabled) or On-chip ROM Enabled Extended Mode
Environment for programming and erasure	On-board programming			
Programmable and erasable area	User area/user boot area/data area	User area/data area	User area/data area	User area/data area
Division into erasure blocks	Possible*1	Possible*1	Possible	Possible
Boot program at a reset	Boot program	USB boot program	User boot program	User program

Note 1. The entire ROM may be erased at the time of booting up. Specified blocks can subsequently be erased. For details, refer to section 43.8.4, ID Code Protection (Boot Mode), section 43.8.2, State Transitions in Boot Mode, and section 43.9.2, State Transitions.

- Programming and erasure of the user boot area are only possible in boot mode.
- In boot mode, a host is able to program, erase, or read the user area, user boot area, or data area via an SCI.
- In boot mode, on-chip RAM is employed for the boot program. Therefore, the data on the on-chip RAM is not retained.
- Booting-up in USB boot mode and user boot mode is from the user boot area. The user boot area of the product as-shipped holds the USB boot program, which is capable of reading from or writing to the user area and data area. Furthermore, rewriting of the user boot area in boot mode can enable reading from or writing to the user area and data area via any desired interface.

#### 43.3.1 Erasure of Areas that are Subject to ID Code Protection

The areas to be erased depend on the operating mode at the time of booting up and the result of reference to the ID code. For a description of the ID code protection, refer to section 43.8.4, ID Code Protection (Boot Mode).

**Table 43.7 Connection between areas for erasure, the operating mode, and ID code protection**

Operating Mode at the Time of Booting Up	Control Code of ID Code Protection	Matching or non-matching ID codes	User Area	User Boot Area	Data Area
Boot mode	45h	Matching	—	—	—
		Non-matching three times consecutively	Erase	Erase*1	Erase
	52h	Matching or non-matching	—	—	—
	Other than 45h and 52h (ID code protection is disabled)	—	Erase	Erase*1	Erase
USB boot mode	ID code protection is not present in USB boot mode.		Erase		Erase
User boot mode	ID code protection depends on the user boot mode specification.				
Single-chip mode	Erasure does not proceed when booting up is in single-chip mode.				

Note 1. The user boot area is not erased if the user boot program is stored in the user boot area.

## 43.4 FCU

The ROM and E2 DataFlash operations are performed by issuing commands to a dedicated sequencer (FCU).

The mode transitions of the FCU and the system of commands are described below. The descriptions apply in common to boot mode, USB boot mode, user boot mode, single-chip mode (with on-chip ROM enabled), and on-chip ROM enabled expansion mode.

### 43.4.1 FCU Modes

The FCU has eight modes. Transitions between modes are caused by modifying FENTRYR or issuing FCU commands. Since the E2 DataFlash P/E mode is included in ROM read mode, high-speed reading from the ROM is possible in E2 DataFlash P/E mode. Figure 43.4 is a diagram of the FCU mode transitions.

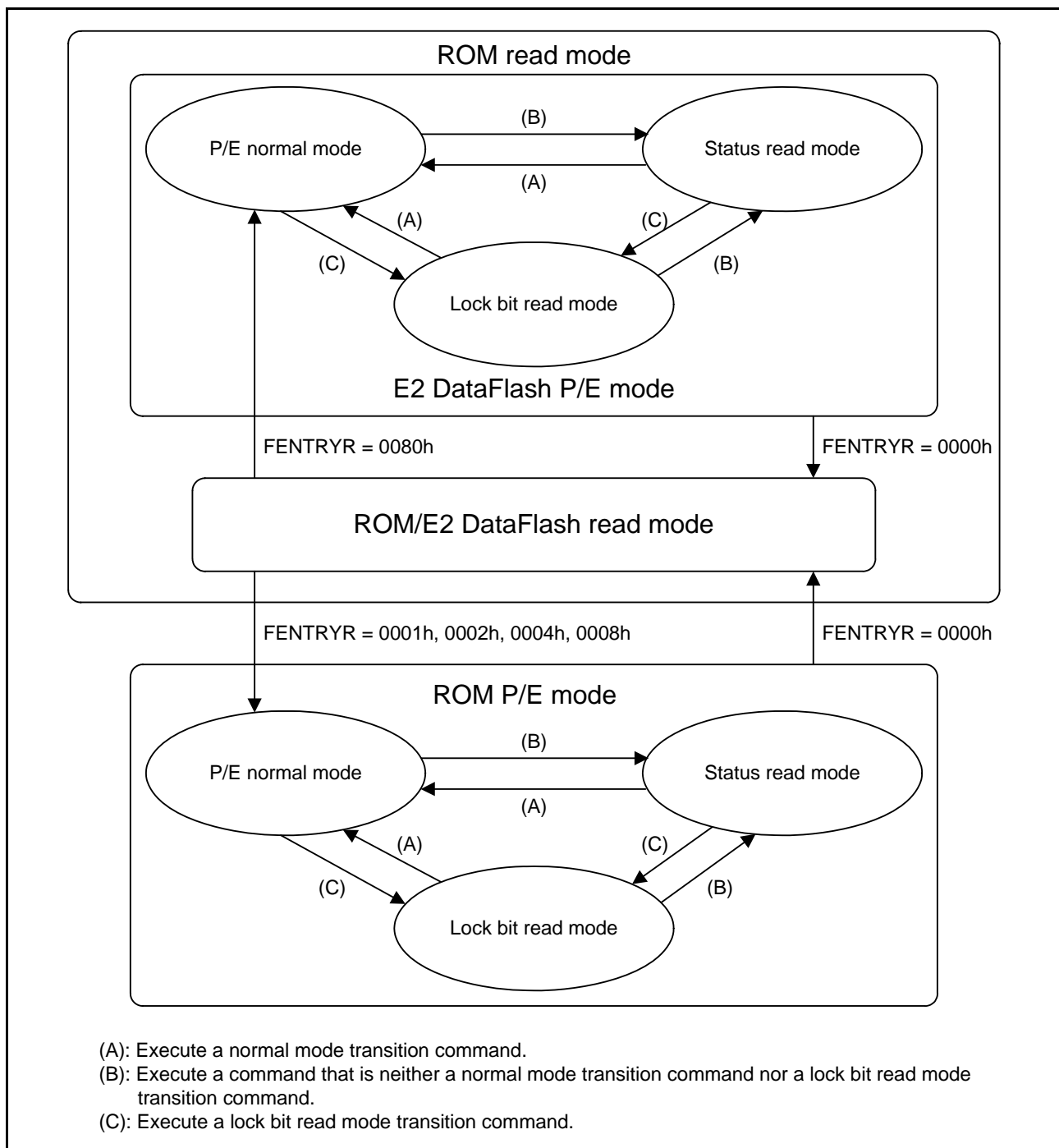


Figure 43.4 Mode Transitions of the FCU

### 43.4.1.1 ROM Read Modes

The ROM read modes are for high-speed reading of the ROM. Reading out from an address for reading can be accomplished in one cycle of ICLK.

ROM/E2 DataFlash read mode and E2 DataFlash P/E mode are the two kinds of ROM read modes.

### 43.4.1.2 ROM/E2 DataFlash Read Mode

This mode is for reading the ROM and E2 DataFlash memory. The FCU does not accept FCU commands. The FCU enters this mode when the FENTRYR.FENTRY<sub>n</sub> (n = 0 to 3) bits are set to 0 with the FENTRYR.FENTRYD bit set to 0.

### 43.4.1.3 ROM P/E Modes

The ROM P/E modes are for programming and erasure of the ROM. High-speed reading of the ROM is not possible in these modes. Reading out from an address within the range for reading causes a ROM-access violation, and the FASTAT.CMDLK bit is set to 1 (command-locked state) (see section 43.6.2, Command-Locked State).

There are three ROM P/E modes.

#### (1) ROM P/E Normal Mode

The transition to ROM P/E normal mode is the first transition in the process of programming or erasing the ROM. The FCU enters this mode when the FENTRYR.FENTRYD bit is set to 0, with any of the FENTRYR.FENTRY<sub>n</sub> (n = 0 to 3) bits set to 1 in ROM read mode, or when the normal mode transition command is received in ROM P/E modes. Table 43.11 lists the acceptable commands in this mode.

Reading out from an address within the range for programming and erasure while any of the FENTRYR.FENTRY<sub>n</sub> (n = 0 to 3) bits are set to 1 causes a ROM-access violation, and the FASTAT.CMDLK bit is set to 1 (command-locked state) (see section 43.6.2, Command-Locked State).

#### (2) ROM Status Read Mode

In the ROM status read mode, the state of the ROM can be read. The FCU enters this mode when a status read mode transition command is received, or when a command other than the normal mode transition or lock bit read mode transition command is received in ROM P/E modes.

ROM status read mode encompasses the states where the FSTATR0.FRDY bit is 0 and the FASTAT.CMDLK bit is set to 1 (command-locked state) after an error has occurred. Table 43.11 lists the acceptable commands in this mode.

Reading out from an address within the range for programming and erasure while any of the FENTRYR.FENTRY<sub>n</sub> (n = 0 to 3) bits are 1 allows the value of FSTATR0 to be read.

#### (3) ROM Lock-Bit Read Mode

In the ROM lock-bit read mode, reading the ROM allows the lock bits to be read. The FCU enters this mode when a lock-bit read mode transition command is received in ROM P/E modes. Table 43.11 lists the acceptable commands in this mode.

Reading out from an address within the range for programming and erasure while any of the FENTRYR.FENTRY<sub>n</sub> (n = 0 to 3) bits are 1 allows the value of the lock bit of the block including the accessed address to be read from all the read bits.



#### 43.4.1.4 E2 DataFlash P/E Modes

These modes are for programming and erasure of the E2 DataFlash memory. Although high-speed reading from the ROM is possible, reading from the E2 DataFlash is not executable. Although FCU commands for the E2 flash memory are accepted in this mode, FCU commands for the ROM are not. The FCU enters this mode when the FENTRYR.FENTRY<sub>n</sub> (n = 0 to 3) bits are all set to 0 and the FENTRYR.FENTRYD bit is set to 1.

There are three E2 DataFlash P/E modes.

##### (1) E2 DataFlash P/E Normal Mode

The transition to E2 DataFlash P/E normal mode is the first transition in the process of programming or erasing the E2 DataFlash.

The FCU enters this mode when the FENTRYR.FENTRYD bit is set to 1 and the FENTRYR.FENTRY<sub>n</sub> (n = 0 to 3) bits are all set to 0 in ROM/E2 DataFlash read mode, or when the normal mode transition command is received in E2 DataFlash P/E modes. Table 43.11 lists the acceptable commands in this mode.

##### (2) E2 DataFlash Status Read Mode

The E2 DataFlash status read mode is for reading information on the state of the E2 DataFlash.

The FCU enters this mode when a status read mode transition command is received, or when a command other than the normal mode transition and lock bit read mode transition command is received in E2 DataFlash P/E modes. E2 DataFlash status read mode encompasses the states where the FRDY bit in FSTATR0 is 0 and the FASTAT.CMDLK bit is set to 1 (command-locked state) after an error has occurred. Table 43.11 lists the acceptable commands in this mode. Reading out from an address within the E2 DataFlash area will actually read the value of the FSTATR0 register. High-speed reading of the ROM is possible.

##### (3) E2 DataFlash Lock-Bit Read Mode

Since the E2 DataFlash memory does not have lock bits, the lock bits are not read even if a transition to this mode is made. If the E2 DataFlash memory area is read after a transition to this mode, an E2 DataFlash access violation is not generated, but the values read are undefined. High-speed reading of the ROM is possible.

The FCU enters this mode when a lock-bit read mode transition command is received in E2 DataFlash P/E modes. Table 43.11 lists the acceptable commands in this mode.

### 43.4.2 FCU Commands

FCU commands consist of commands for mode transitions of the FCU and commands for programming and erasure. Table 43.8 lists the FCU commands for use with the ROM and E2 DataFlash.

**Table 43.8 FCU Commands**

Command	ROM	E2 DataFlash
P/E normal mode transition	Shifts to normal mode (see section 43.4.3, Connections between FCU Modes and Commands)	
Status read mode transition	Shifts to status read mode (see section 43.4.3, Connections between FCU Modes and Commands)	
Lock bit read mode transition (lock bit read 1)	Shifts to lock bit read mode (see section 43.4.3, Connections between FCU Modes and Commands)	
Peripheral clock notification	Sets the FlashIF clock (FCLK)	
Programming	ROM programming (in 128-byte units)	E2 DataFlash programming (in 2-byte units)
Block erase	ROM erasure (in block units, with the lock bit being erased simultaneously)	E2 DataFlash erasure (in block units)
P/E suspend	Suspends programming/erasure	
P/E resume	Resumes programming/erasure	
Status register clear	Clears the ILGLERR, ERSERR and PRGERR bits in FSTATR0 and the FASTAT.CMDLK bit (FCU command lock bit)	
Lock bit read 2	Reads the lock bit of a specified block (the value of the lock bit is reflected in the FSTATR1.FLOCKST bit)	—
Lock bit programming	Programs the lock bit of a specified block	—
Blank checking	—	Checks whether the E2 DataFlash memory is blank

The lock bit read 2 command is for the ROM also used as the blank check command for the E2 DataFlash memory. That is, when a lock bit read 2 command is issued for the E2 DataFlash, blank checking is executed for the E2 DataFlash memory.

Commands for the FCU are issued by writing an FCU command to addresses within the range for ROM programming and erasure or an address in the E2 DataFlash. Table 43.9 lists the formats of the FCU commands. Writing data to addresses listed in Table 43.9 in accordance with certain conditions causes the FCU to execute processing for the corresponding command. An explanation in summary of the FCU command format is given in Table 43.10.

Furthermore, although almost all of the FCU commands are in byte units, some of the commands have to be issued in word units.

For details on the conditions for the acceptance of the individual FCU commands, see section 43.4.3, Connections between FCU Modes and Commands. For how to use the FCU commands, see section 43.4.4, FCU Command Usage.

**Table 43.9 FCU Command Formats**

Command	Number of Bus Cycles	Address	1st Cycle	2nd Cycle	3rd Cycle	4th Cycle	5th Cycle	6th Cycle	7th to 66th Cycles	67th Cycle
		Data								
P/E normal mode transition	1	Address	RA	—	—	—	—	—	—	—
		Data	FFh	—	—	—	—	—	—	—
Status read mode transition	1	Address	RA	—	—	—	—	—	—	—
		Data	70h	—	—	—	—	—	—	—
Lock bit read mode transition (lock bit read 1)	1	Address	RA	—	—	—	—	—	—	—
		Data	71h	—	—	—	—	—	—	—
Peripheral clock notification	6	Address	RA	RA	RA	RA	RA	RA	—	—
		Data	E9h	03h	0F0Fh <sub>*1</sub>	0F0Fh <sub>*1</sub>	0F0Fh <sub>*1</sub>	D0h	—	—
Programming (ROM)	67	Address	RA	RA	WA	RA	RA	RA	RA	RA
		Data	E8h	40h	WDn <sub>*1</sub>	WDn <sub>*1</sub>	WDn <sub>*1</sub>	WDn <sub>*1</sub>	WDn <sub>*1</sub>	D0h
Programming (E2 DataFlash)	4	Address	RA	RA	WA	RA	—	—	—	—
		Data	E8h	01h	WDn <sub>*1</sub>	D0h	—	—	—	—
Block erase	2	Address	RA	BA	—	—	—	—	—	—
		Data	20h	D0h	—	—	—	—	—	—
P/E suspend	1	Address	RA	—	—	—	—	—	—	—
		Data	B0h	—	—	—	—	—	—	—
P/E resume	1	Address	RA	—	—	—	—	—	—	—
		Data	D0h	—	—	—	—	—	—	—
Status register clear	1	Address	RA	—	—	—	—	—	—	—
		Data	50h	—	—	—	—	—	—	—
Lock bit read 2 (ROM)	2	Address	RA	BA	—	—	—	—	—	—
		Data	71h	D0h	—	—	—	—	—	—
Blank checking (E2 DataFlash)	2	Address	RA	BA	—	—	—	—	—	—
		Data	71h	D0h	—	—	—	—	—	—
Lock bit programming (ROM)	2	Address	RA	BA	—	—	—	—	—	—
		Data	77h	D0h	—	—	—	—	—	—

Note 1. Write data in word units.

**Table 43.10 FCU Commands**

Item		ROM	E2 DataFlash
Address	RA	Any address for programming or erasure within the target ROM area*1	Any address within the E2 DataFlash
	WA	Programming-destination address in the range for programming or erasure (128-byte alignment)	Programming-destination address (2-byte alignment)
	BA	Any address for programming or erasure within the target erasure block	Any address within the target erasure block
Data	WDn	nth word of data for programming (n = 1 to 64)	nth word of data for programming (n = 1)
	Others	Command issued to the target address	

Note 1. Addresses that can be used for programming or erasure differ according to the ROM capacity and the settings of the FENTRYR register. For the ROM capacity, see section 43.1.1, Configuration of the ROM Area, and for FENTRYR, see section 43.2.13, Flash P/E Mode Entry Register (FENTRYR).

### 43.4.3 Connections between FCU Modes and Commands

The sets of FCU commands that can be accepted in each of the FCU modes are fixed. Furthermore, which commands are acceptable in a given FCU mode also depends on the state of the FCU.

Issuing of an FCU command must follow checking of the FCU's state after transitions of the FCU mode.

Commands that are acceptable in the various FCU modes and states are listed in Table 43.11. Issuing a command that is not currently acceptable leads to the FASTAT.CMDLK bit being set to 1 (command-locked state) (see section 43.6.2, Command-Locked State).

Issuing of an FCU command must follow checking of the values of the FRDY, ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and of the FSTATR1.FCUERR bit after transitions of the FCU mode. Furthermore, the FASTAT.CMDLK bit can be checked to see if an error has occurred. The value of the FASTAT.CMDLK bit is the logical OR of the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the FSTATR1.FCUERR bit.

**Table 43.11 Acceptable Commands and the State and Mode (ROM P/E Mode and E2 DataFlash P/E Mode) of the FCU**

	P/E Normal Mode			Status Read Mode										Lock-Bit Read Mode		
	Programming Suspended	Erase Suspended	Other State	Programming or Erasure	Programming while erasure is suspended	Processing to Suspend Programming or Erasure	Lock Bit Read 2 Processing (ROM)	Blank checking (E2 DataFlash Memory)	Programming Suspended	Erase Suspended	Command-Locked State (FRDY = 0)	Command-Locked State (FRDY = 1)	Other State	Programming Suspended	Erase Suspended	Other State
FSTATR0.FRDY bit	1	1	1	0	0	0	0	0	1	1	0	1	1	1	1	1
FSTATR0.SUSRDY bit	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
FSTATR0.ERSSPD bit	0	1	0	0	1	0/1	0/1	0/1	0	1	0/1	0/1	0	0	1	0
FSTATR0.PRGSPD bit	1	0	0	0	0	0/1	0/1	0/1	1	0	0/1	0/1	0	1	0	0
FASTAT.CMDLK bit	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0
Normal mode transition	A	A	A	x	x	x	x	x	A	A	x	x	A	A	A	A
Status read transition	A	A	A	x	x	x	x	x	A	A	x	x	A	A	A	A
Lock-bit read transition (lock bit read 1)	A	A	A	x	x	x	x	x	A	A	x	x	A	A	A	A
Peripheral clock notification	x	x	A	x	x	x	x	x	x	x	x	x	A	x	x	A
Programming	x	*	A	x	x	x	x	x	x	*	x	x	A	x	*	A
Block erase	x	x	A	x	x	x	x	x	x	x	x	x	A	x	x	A
P/E suspend	x	x	x	A	x	x	x	x	x	x	x	x	x	x	x	x
P/E resume	A	A	x	x	x	x	x	x	A	A	x	x	x	A	A	x
Status register clear	A	A	A	x	x	x	x	x	A	A	A	A	A	A	A	A
Lock bit read 2 (ROM)	A	A	A	x	x	x	x	x	A	A	x	x	A	A	A	A
Lock bit programming (ROM)	x	*	A	x	x	x	x	x	x	*	x	x	A	x	*	A
Blank checking (E2 DataFlash)	A	A	A	x	x	x	x	x	A	A	x	x	A	A	A	A

A: Acceptable

\*: Only programming is acceptable for blocks other than the block where erasure was suspended

x: Not acceptable

### 43.4.4 FCU Command Usage

The set of FCU commands consists of commands for FCU mode transitions, actually programming or erasing the ROM, error processing, and suspension and resumption. The following passages describe the various commands. For a description of the modes and states where the respective commands are acceptable, see section 43.4.3, Connections between FCU Modes and Commands.

#### 43.4.4.1 Mode Transitions

This subsection covers the commands for mode transitions. For an illustration of the various transitions between modes, see Figure 43.4.

##### (1) Switching to ROM Read Mode or ROM/E2 DataFlash Read Mode

High-speed reading of the ROM requires clearing of the FENTRYR.FENTRY<sub>n</sub> (n = 0 to 3) bits to 0000b, which places the FCU in ROM read mode. Writing of 02h as a byte to FWEPROR is also required to disable programming and erasure (see section 43.2.1, Flash Write Erase Protection Register (FWEPROR)).

Before switching the FCU from ROM P/E mode to read mode, ensure that all processing of FCU commands has been completed and that the FCU has not detected an error.

For a transition to ROM/E2 DataFlash read mode, the FENTRYR.FENRTY<sub>n</sub> (n = 0 to 3) bits and the FENTRYD bit must be set to 0.

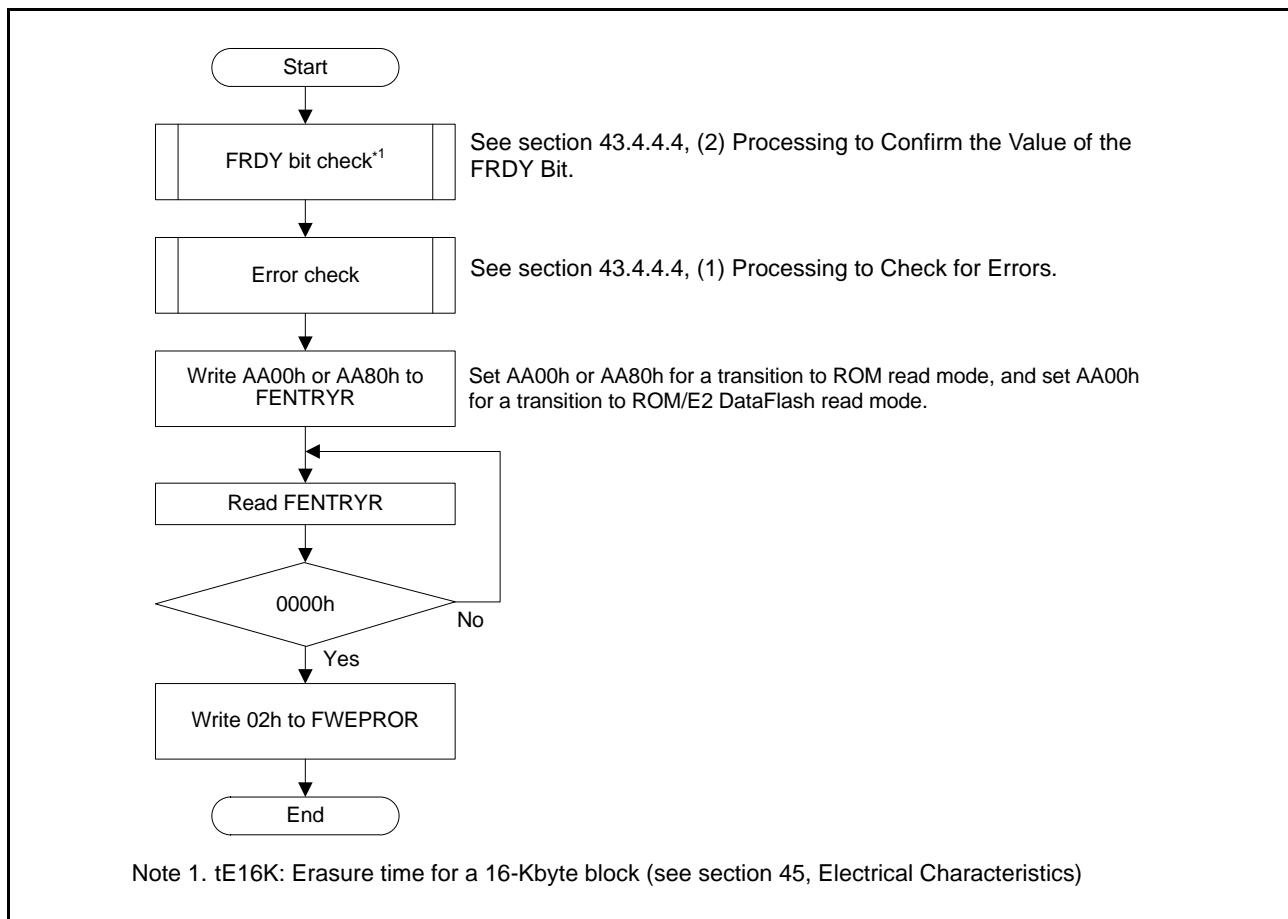


Figure 43.5 Procedure for Transition to ROM Read Mode or ROM/E2 DataFlash Read Mode

## (2) Switching to P/E Mode

A transition to ROM P/E mode is required before executing an FCU command for programming or erasure of the ROM. Setting any of the FENTRYR.FENTRYn (n = 0 to 3) bits to 1 causes a transition to ROM

P/E mode for programming and erasure of the corresponding address range.

A transition to E2 DataFlash P/E mode is required before executing an FCU command for programming or erasure of the E2 DataFlash. For a transition to E2 DataFlash P/E mode, set the FENTRYR.FENTRYD bit to 1.

Before actually proceeding to program or erase the ROM, enable programming and erasure by writing 01h as a byte to FWEPROR (see section 43.2.1, Flash Write Erase Protection Register (FWEPROR)).

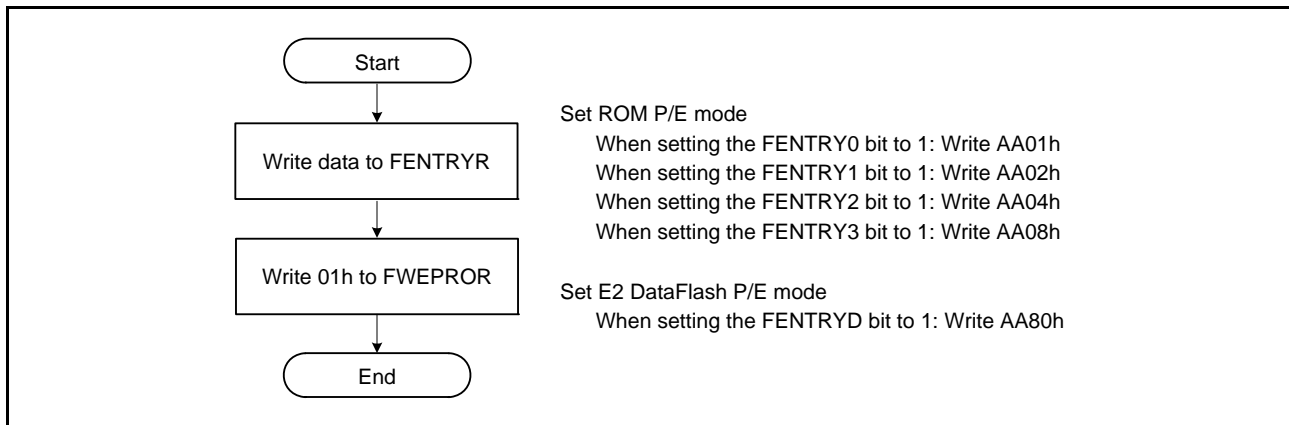


Figure 43.6 Procedure for Transition to ROM P/E Mode or E2 DataFlash P/E Mode

## (3) Switching to P/E Normal Mode

Two methods are available for the transition to P/E normal mode: setting FENTRYR while the FCU is in ROM/E2 DataFlash read mode (see section 43.4.1, FCU Modes), or issuing the normal mode transition command while the FCU is in P/E mode (see Figure 43.7). The normal mode transition command is issued by writing FFh to a ROM programming/erasure address or to an E2 DataFlash address.

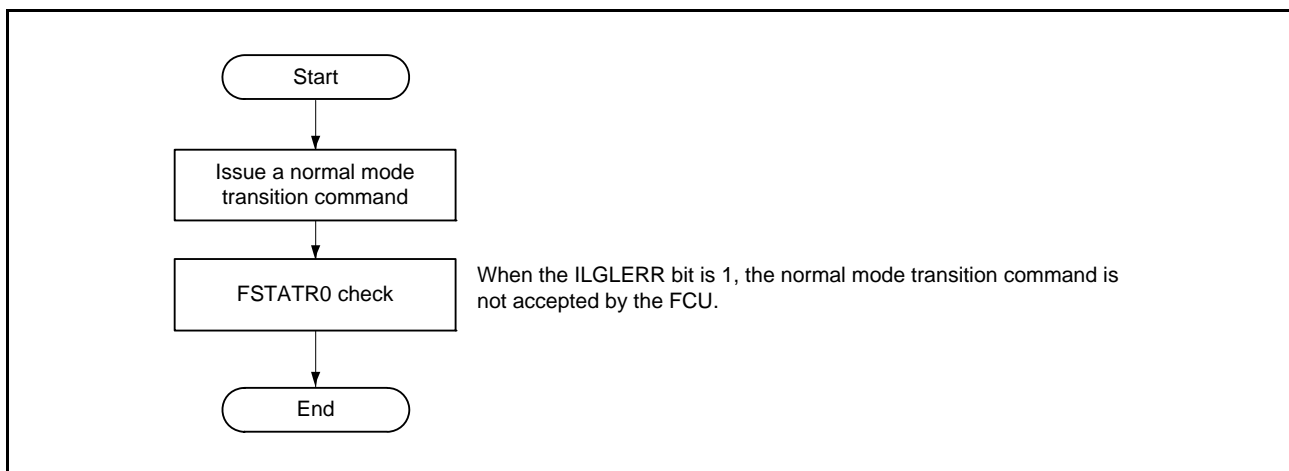


Figure 43.7 Procedure for Transition to ROM P/E Normal Mode

#### (4) Switching to Status Read Mode

Issuing a status read mode transition command or an FCU command other than a normal mode transition or lock bit read mode transition command places the FCU in status read mode. Figure 43.8 shows the procedure for checking FSTATR0 as an example. In the example, the status read mode transition command is issued to place the FCU in ROM status read mode, and the value of FSTATR0 is obtained by reading out from a ROM programming/erasure address or an E2 DataFlash address and then checked.

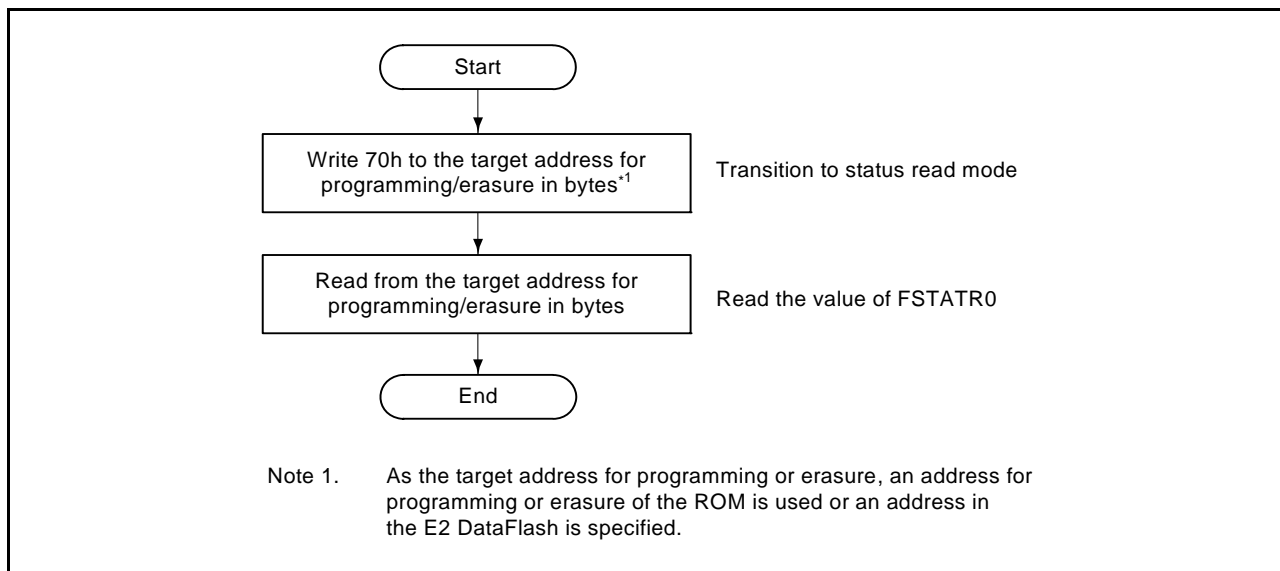
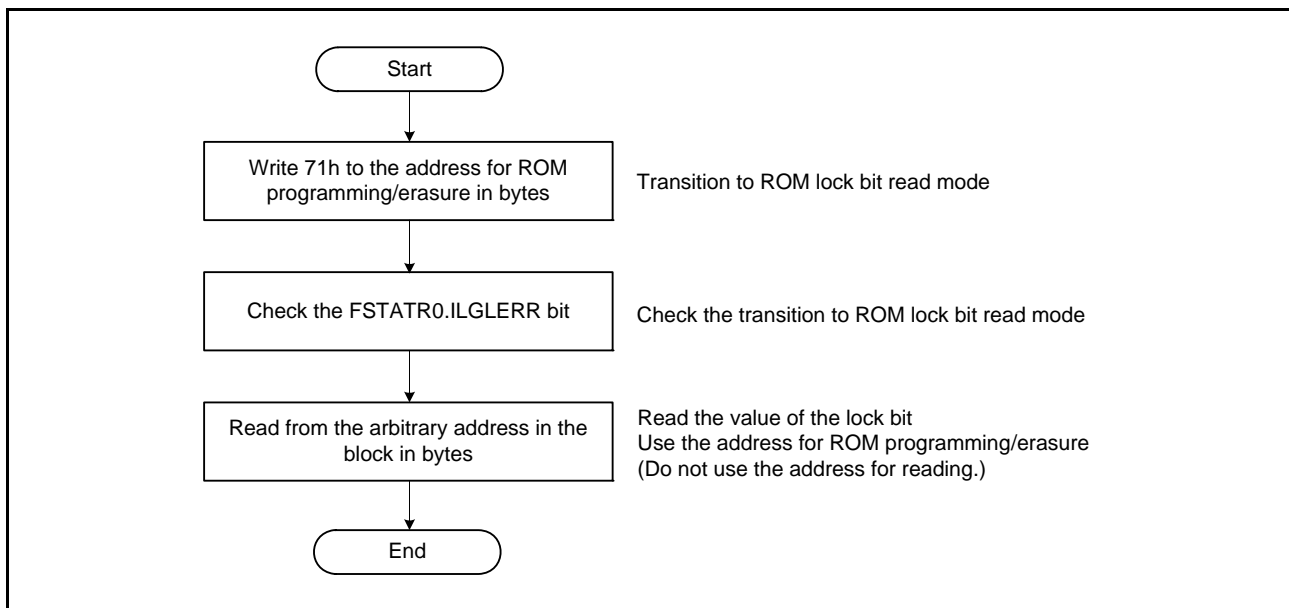


Figure 43.8 Procedure for Transition to ROM Status Read Mode and the Status Checking

### (5) Switching to ROM Lock-Bit Read Mode

Clearing the FMODR.FRDM bit (memory area reading method) issues a lock bit read mode transition (lock bit read 1) command. After the transition to lock bit read mode, the lock bit value is obtained by reading out from a ROM programming/erasure address. All bits of a value thus read have the value of the lock bit of the block that contains the accessed address (Figure 43.9).

Since there are no lock bits for the E2 DataFlash, undefined data are read from the E2 DataFlash area after a transition of the lock bit read mode is made.



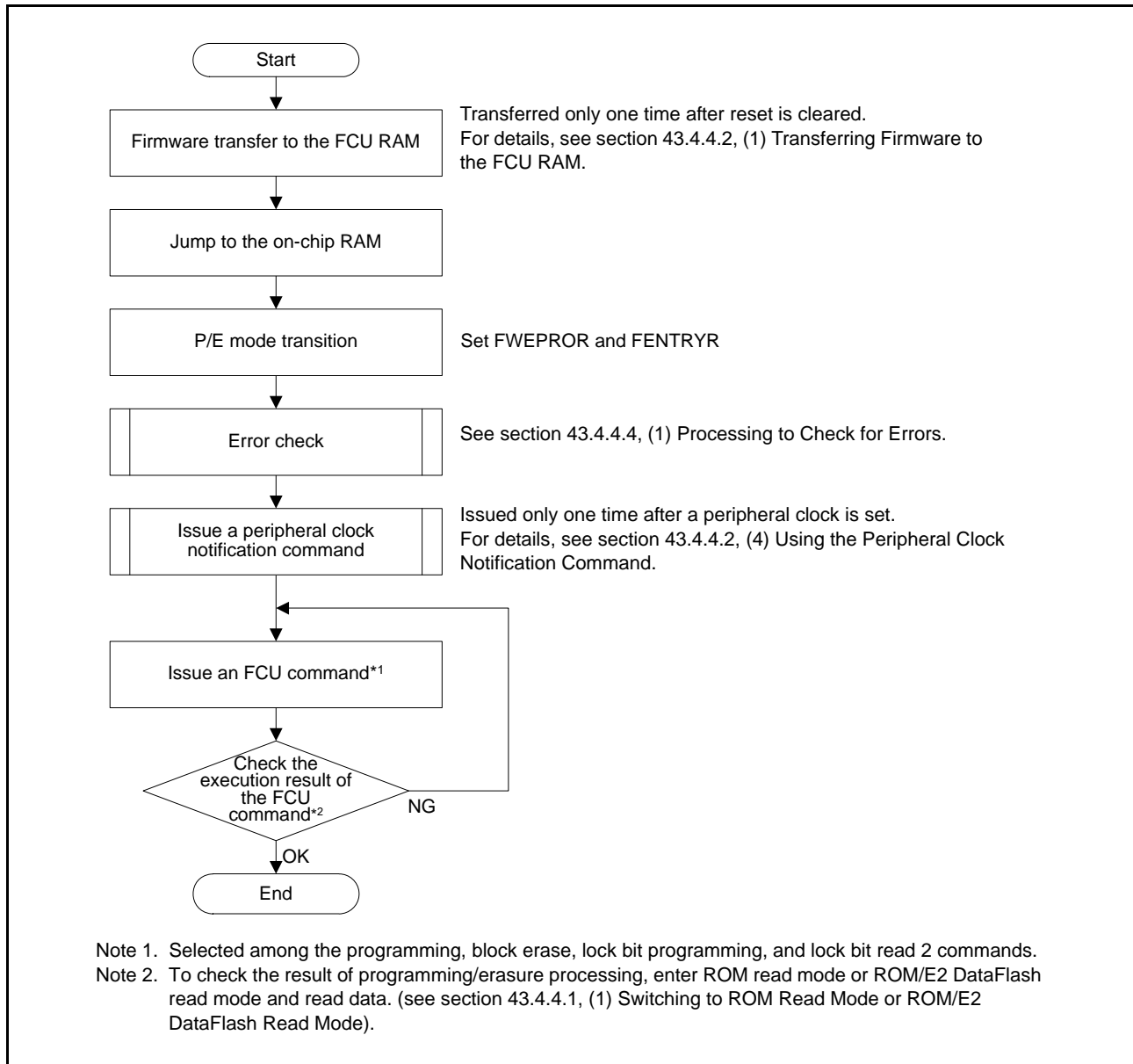
**Figure 43.9 Procedure for Transition to ROM Lock-Bit Read Mode and Lock-Bit Reading**



### 43.4.4.2 Programming and Erasure Procedures

The following passages describe the flow of procedures for programming or erasing the ROM or E2 DataFlash. For details on the acceptance of commands by the FCU, see section 43.4.3, Connections between FCU Modes and Commands.

Figure 43.10 is a simple flowchart of the procedure for executing FCU commands.

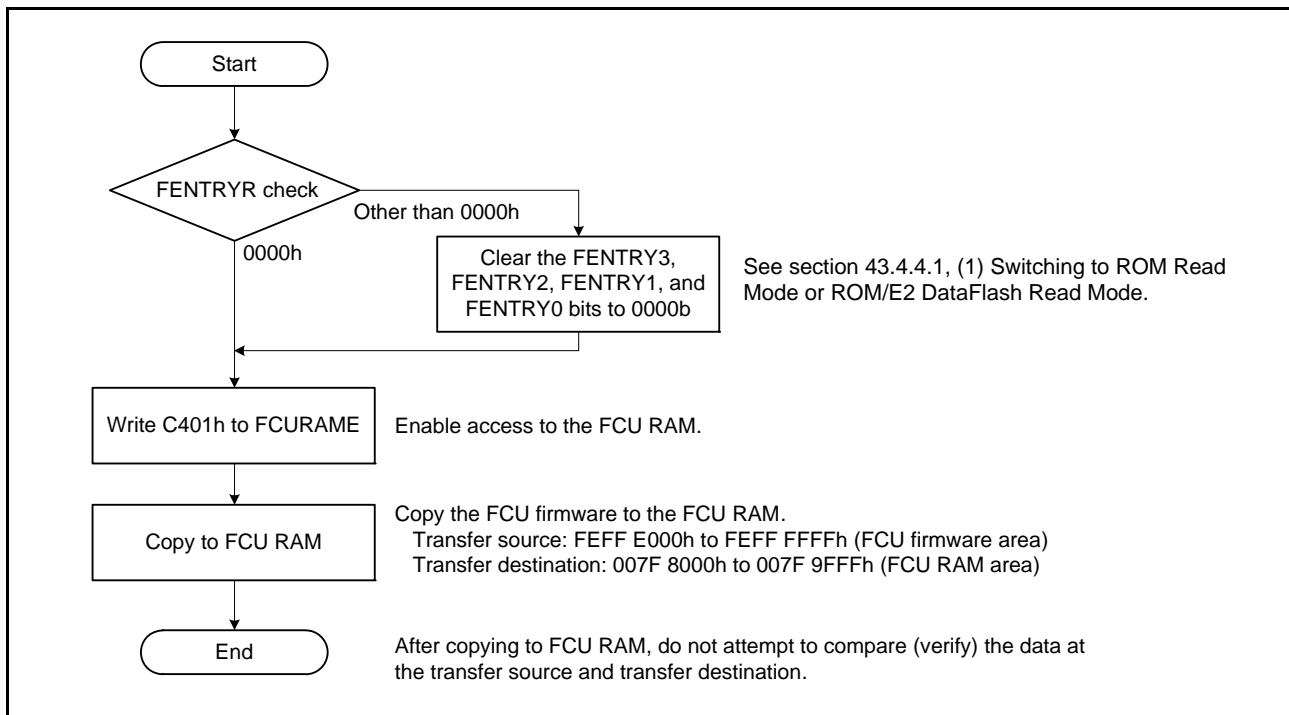


**Figure 43.10 Simple Flowchart of the Procedure for Programming and Erasure**

### (1) Transferring Firmware to the FCU RAM

FCU commands can only be used if the FCU RAM holds the firmware for the FCU. The FCU RAM does not hold the FCU firmware immediately after the chip has been booted up, so the firmware must be copied from the FCU firmware area to the FCU RAM. Furthermore, when the FSTATR1.FCUERR bit is set to 1, the FCU must be reset and the firmware recopied because the firmware stored in the FCU RAM may have been corrupted.

Figure 43.11 shows the flow of the procedure for transferring firmware to the FCU RAM. Before writing data to the FCU RAM, set FENTRYR to 0000h and stop the FCU.



**Figure 43.11 Procedure for Firmware Transfer to FCU RAM**

### (2) Jumping to Locations in On-chip RAM

Since fetching instructions from the ROM is not possible while the ROM is being programmed or erased, instructions have to be fetched from an area other than the ROM. Copy the required program code to on-chip RAM and then make a jump to the address where the code starts in on-chip RAM.

### (3) Transition to P/E Mode

The FCU is placed in P/E mode by setting the FENTRYR.FENTRY<sub>n</sub> (n = 0 to 3) bits and FWEPROR register. For details, see section 43.4.4.1, (2) Switching to P/E Mode.

#### (4) Using the Peripheral Clock Notification Command

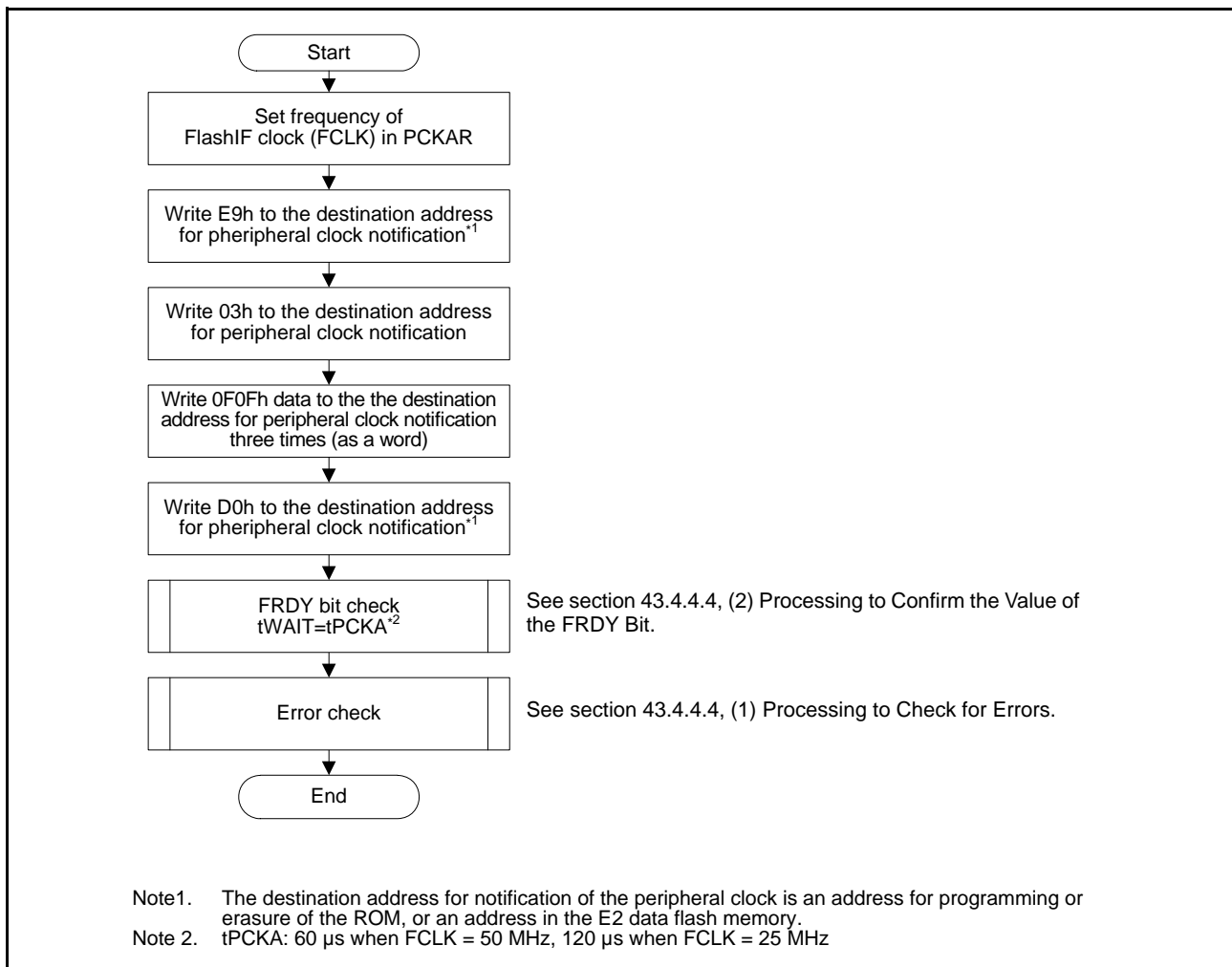
The FlashIF clock (FCLK) is used in programming and erasing the ROM or E2 DataFlash, so the frequency of this clock has to be set in PCKAR. Frequencies in the range from 1 to 100 MHz are selectable. If a frequency within this range has not been set, the FCU will detect the error leading the FASTAT.CMDLK bit being set to 1 (command-locked state) (see section 43.6.2, Command-Locked State).

Note that if the PCKA[7:0] bits in the PCKAR register are set to values outside the range from 4 MHz to 50 MHz, do not issue a programming command to the ROM/E2 DataFlash.

The peripheral clock notification command is used after the PCKAR setting has been made. In the first and second cycles for the peripheral clock notification command, respectively, the values E9h and 03h are written to the address range for programming and erasure of the ROM or the address range in the E2 DataFlash. Write 0F0Fh to the address range for programming and erasure of the ROM or the address range in the E2 DataFlash three times in the third to fifth cycles of the command. After 0F0Fh has been written three times (as a word) to the address range for programming and erasure of the ROM or the address range in the E2 DataFlash, the process of the FCU setting the frequency of the peripheral clock starts once the value D0h has been written in the sixth cycle. The FSTATR0.FRDY bit can be used to check whether or not the settings have been completed.

In the case of the ROM, addresses that can be used in the first to sixth cycles differ according to the settings of the FENTRYR.FENTRYn (n = 0 to 3) bits. Ensure that the addresses suit the settings of the FENTRYR.FENTRYRn bits. If issuing of the command is attempted for an address in the area for which P/E mode is disabled by the FENTRYR register, the FCU will detect the error and the FASTAT.CMDLK bit is set to 1 (command-locked state) (see section 43.6.2, Command-Locked State).

Furthermore, if the setting for the peripheral clock in use will not be changed from this setting after release from the reset state, the setting by the peripheral clock notification command is also valid for the next FCU command.



**Figure 43.12 Using the Peripheral Clock Notification Command**

## (5) Programming

The programming command is used to write data to the ROM or the E2 DataFlash.

- ROM Programming

In the first and second cycles for the programming command, respectively, the values E8h and 40h are written to the address range for programming and erasure of the ROM. In the third cycle, write the actual data to be programmed, as a word unit, to the start address of the target area for programming. For this start address, always use an address that is aligned on a 128-byte boundary. In the fourth to the 66th cycles, write the data for programming in 63 word-unit rounds to the address range for programming and erasure of the ROM. Once the value D0h has been written to the address range for programming and erasure of the ROM in the 67th cycle, the FCU begins the actual process of programming the ROM. The FSTAT0.FRDY bit can be used to check whether or not the programming has been completed.

Addresses that can be used in the first to 67th cycles differ according to the setting of the FENTRYR.FENTRYn (n = 0 to 3) bits. Ensure that the addresses suit the setting of FENTRYR.FENTRYn bits. If issuing of the command is attempted for an address in the area for which P/E mode is disabled by the FENTRYR register, the FCU will detect the error leading the FASTAT.CMDLK bit being set to 1 (command-locked state) (see section 43.6.2, Command-Locked State).

In cases where the target range in the third to 66th cycles includes addresses that do not require programming, use FFFFh as the data for programming to those addresses. Furthermore, when a lock is programmed so that protection by the lock bit becomes effective, the FPROTR.FPROTCN bit must be set to 1.

- E2 DataFlash Programming

Write E8h to an address within the E2 DataFlash area in the first cycle of the programming command, and 01h in the second cycle. In the third cycle, write the first word of data for programming to the address where the target area for programming starts. This address must be on a 2-byte boundary.

After writing words to addresses in the E2 DataFlash area one time, write byte D0h to an address within the E2 DataFlash area in the fourth cycle; the FCU will then start actual programming of the E2 DataFlash. Read the FRDY bit in FSTATR0 to confirm the completion of E2 DataFlash programming.

When programming for locking to prohibit programming and erasure according to the setting of the DFLWEy (y = 0, 1) register proceeds, the relevant bit of the DFLWEy (y = 0, 1) register must be set to 1.

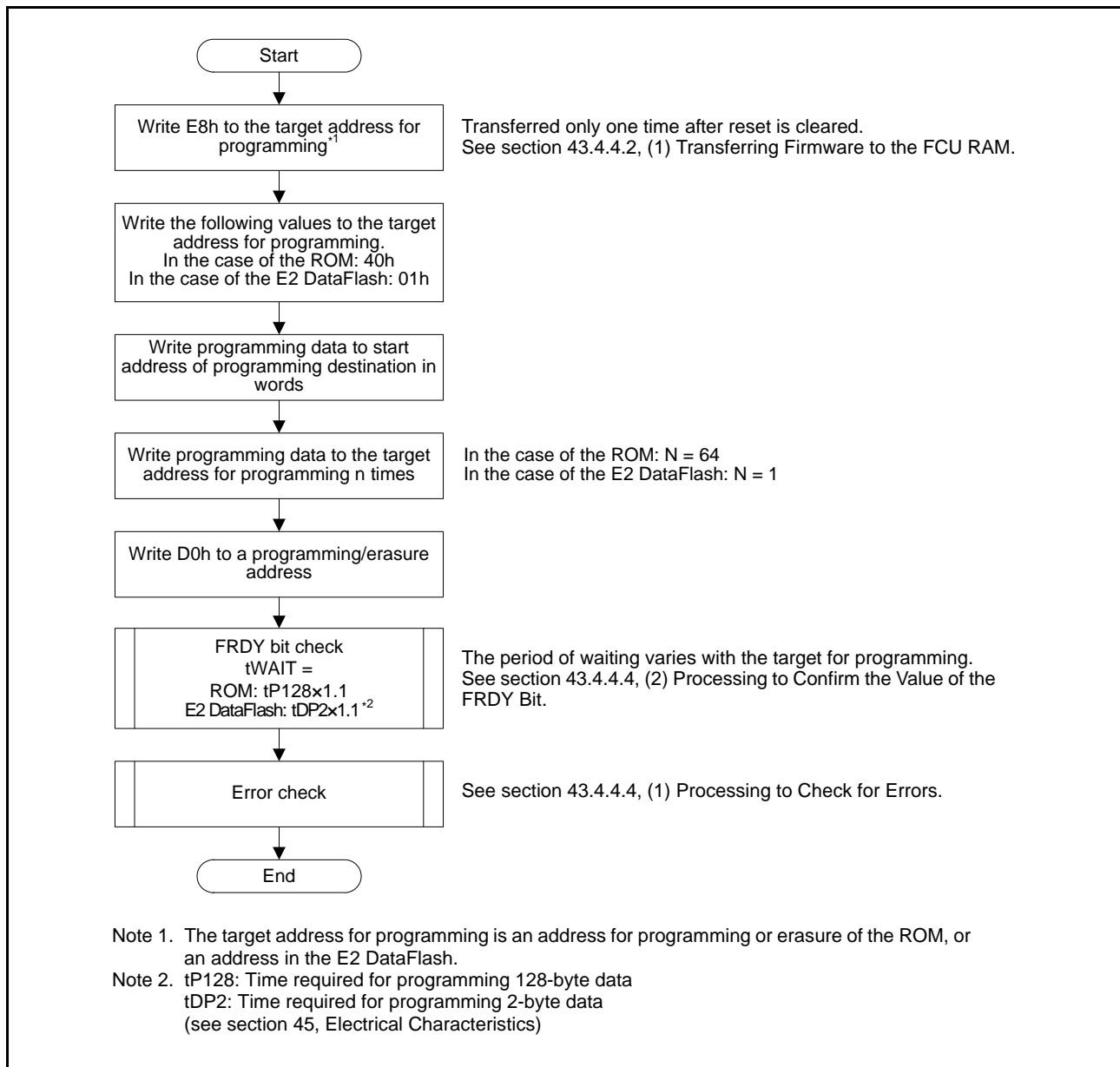


Figure 43.13 Procedure for ROM Programming

## (6) Erasure

To erase the ROM/E2 DataFlash, use the block erasure command.

In the first cycle of a block-erasure command, 20h is written to an address for programming or erasure of the ROM or an address in the E2 DataFlash. In the second cycle, when D0h is written to any address within the target block for erasure, the FCU starts processing to erase the ROM or E2 DataFlash. The FSTATR0.FR DY bit can be checked to confirm the completion of erasure. When the CPU reads ROM that has been erased, the value read is FFFF FFFFh. In the case of the E2 DataFlash, values read are undefined. In the case of the ROM, the FPROTR.FPROTCN bit must be set to 1 if protection by the lock bit is in effect for a block to be erased.

Note that the E2 DataFlash has a programming and erasure protection function that is controlled by DFLWE<sub>y</sub> (y = 0, 1). When erasure for locking to prohibit programming and erasure according to the setting of the DFLWE<sub>y</sub> (y = 0, 1) register proceeds, the relevant bit of the DFLWE<sub>y</sub> (y = 0, 1) register must be set to 1.

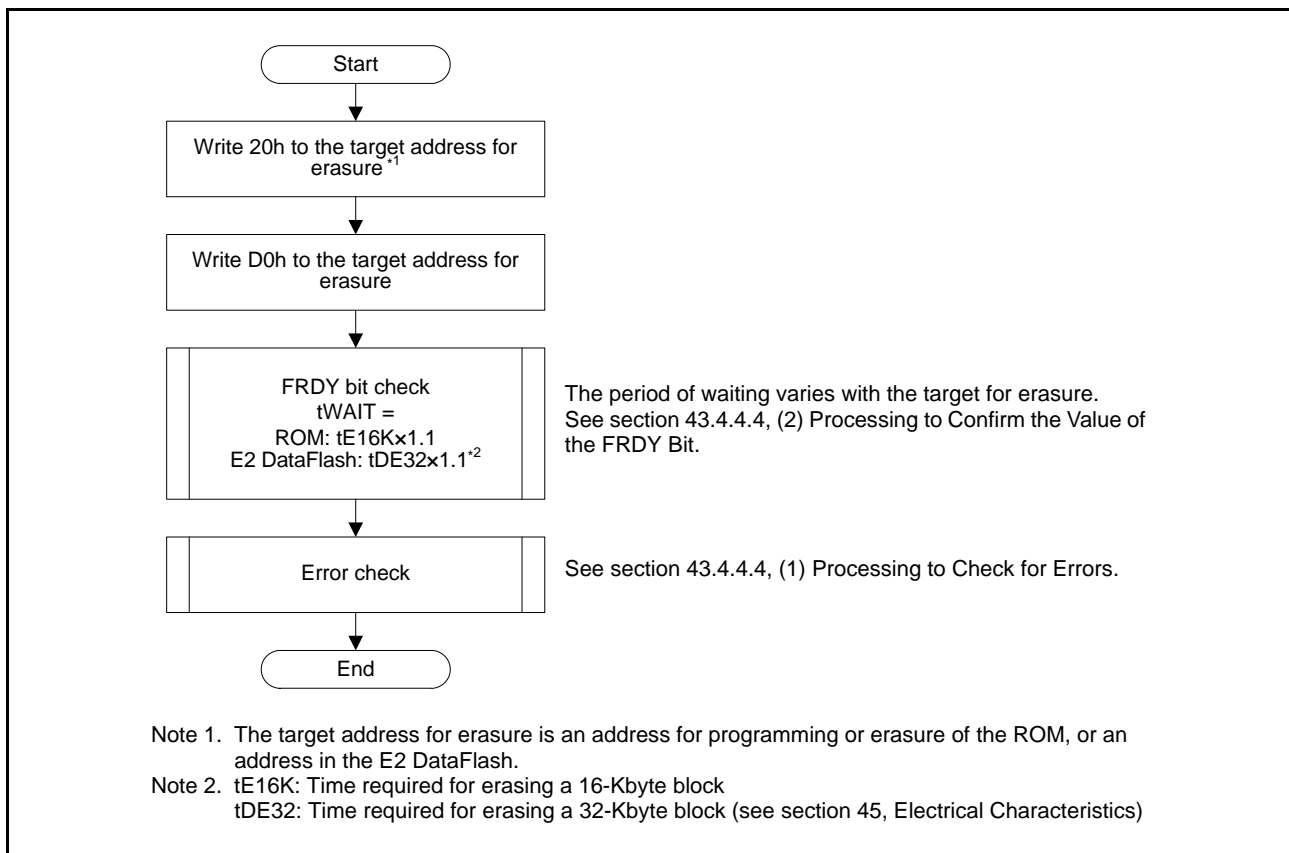
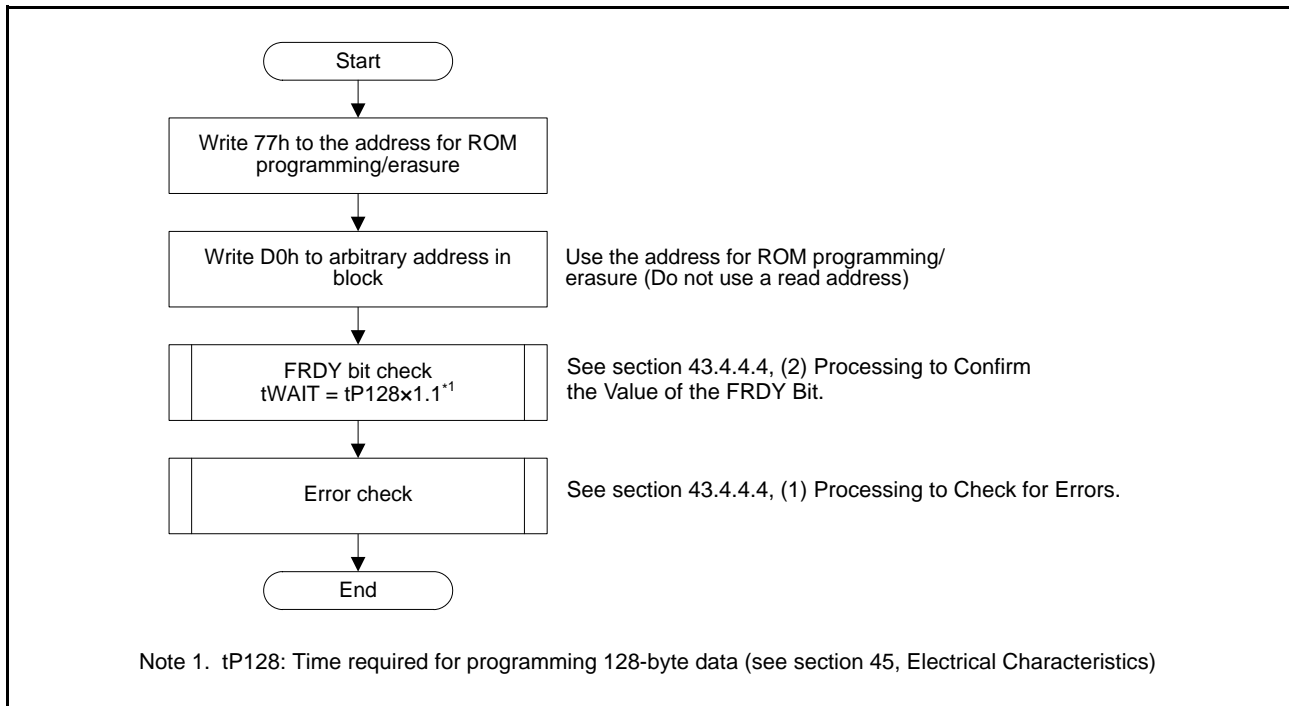


Figure 43.14 Procedure for ROM/E2 DataFlash Erasure

### (7) Programming or Erasing Lock Bits

Lock-bit programming can only be executed on the ROM. Each block in the user area includes a lock bit. To write to a lock bit, use the lock bit programming command. In the first cycle of the lock bit programming command, 77h is written to the ROM programming/erasure address. When D0h is written to an arbitrary address in a block whose lock bit is to be written to in the second cycle, the FCU starts the processing to write to the lock bit. Whether writing is completed can be checked with the FSTAT0.FRDY bit.



**Figure 43.15 Procedure for Programming the Lock Bit**

The block erase command is used to erase lock bits.

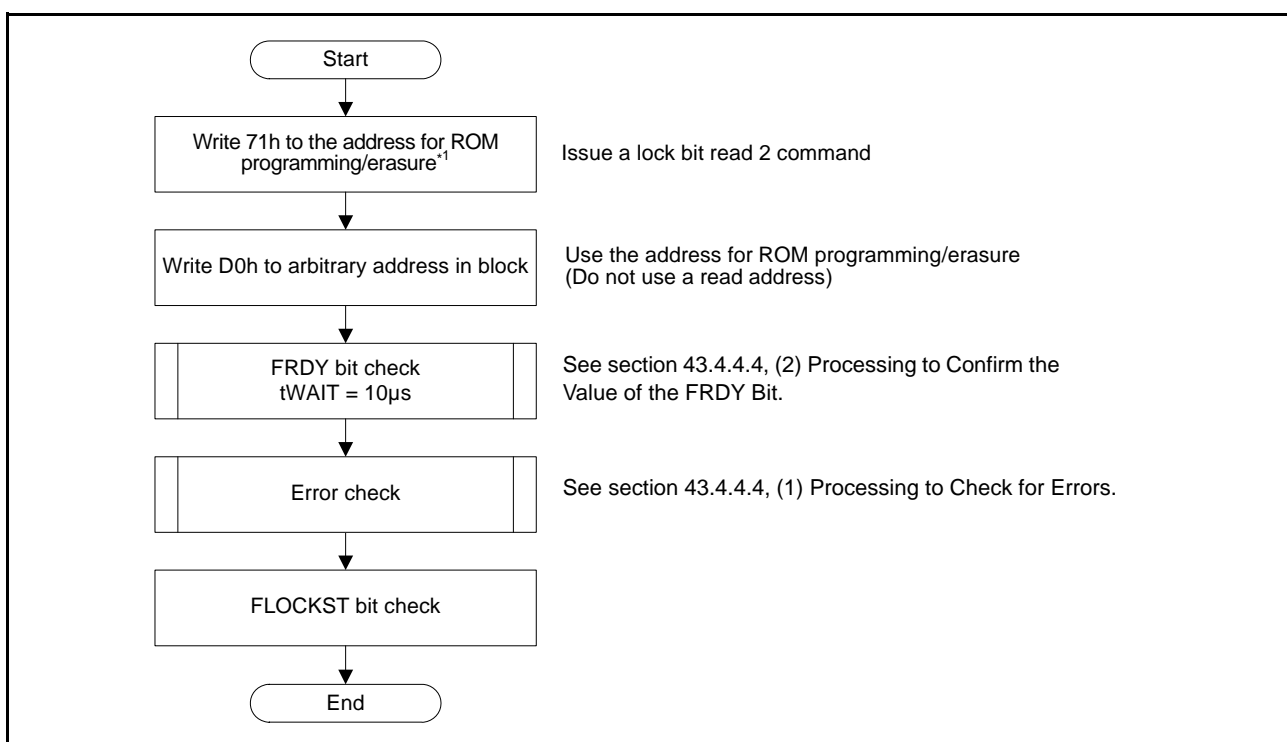
When the FPROTR.FPROTCN bit is 0, erasure blocks whose lock bit is set to 0 cannot be erased. When erasing a lock bit, issue a block erase command with the FPROTCN bit set to 1. Using the block erase command erases all data in the block. It is impossible to erase only a lock bit.

### (8) Reading Lock Bits

Lock-bit reading can only be executed on the ROM. Lock bits can be read by either reading from a memory area or reading from a register.

The lock bit read 2 command is issued in the case of the register reading method (i.e. when the FMODR.FRDM bit is set to 1). This command is issued to an address within the block for which the lock bit is to be read; the address range is that for programming and erasing the ROM. In the first and second cycles of the lock bit read 2 command, the values 71h and D0h are written; once these values have been written, the value of the lock bit for the specified block is copied to the FSTATR1.FLOCKST bit.

In the case of the memory area reading method (i.e. when the FMODR.FRDM bit is 0), the FCU is placed in ROM lock-bit read mode, and the lock bit is obtained by reading from an address within the address range for programming and erasure of the ROM. For details, see Figure 43.9, Procedure for Transition to ROM Lock-Bit Read Mode and Lock-Bit Reading.



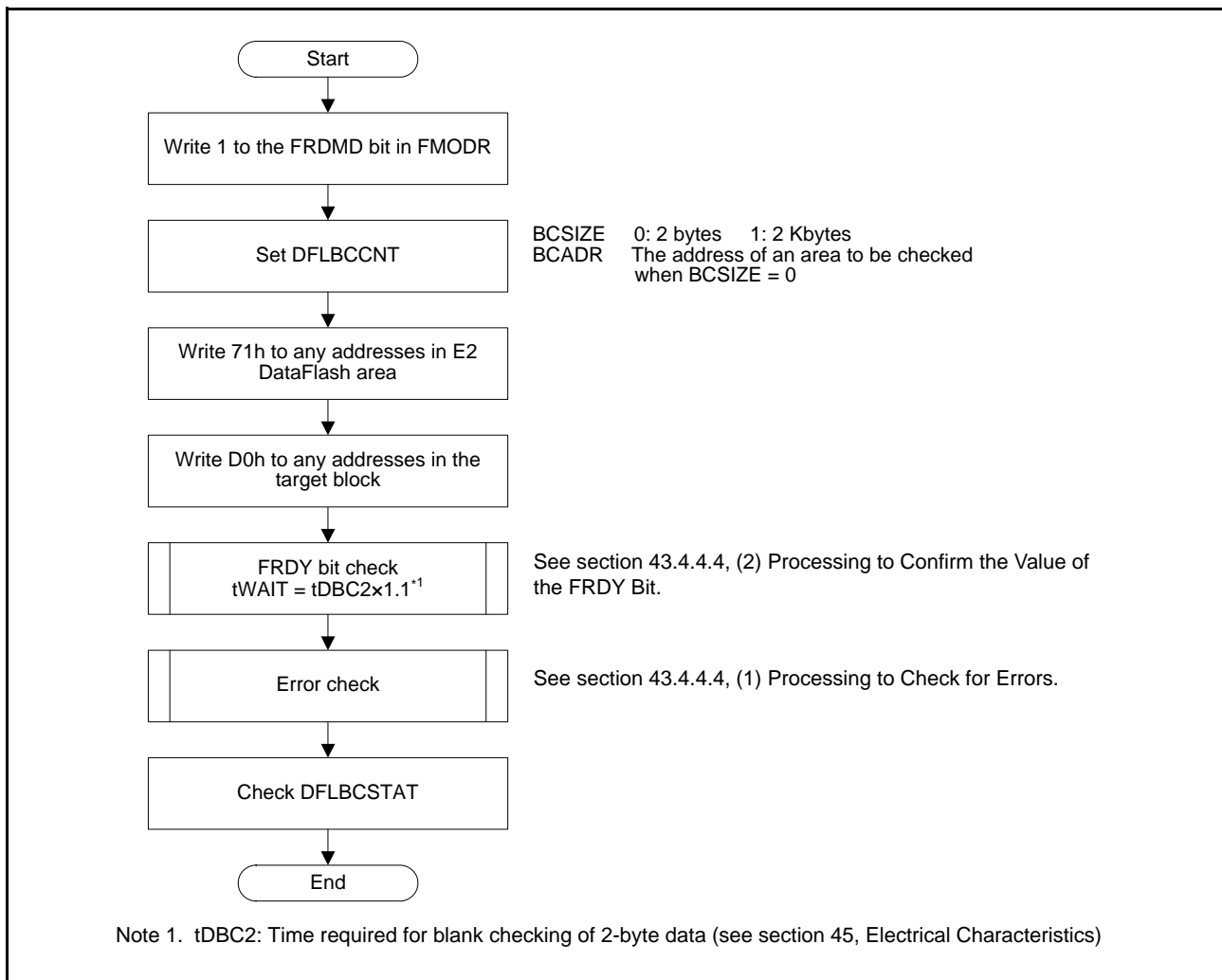
**Figure 43.16 Procedure for Reading Lock Bit in Register Read Mode**

### (9) Blank Checking

Blank checking can only be executed on the E2 DataFlash. Since using the CPU to read erased areas of the E2 DataFlash produces undefined values, the blank checking command should be used to check whether the E2 DataFlash has actually been erased. To make the blank checking command available for use, start by setting the FRDM bit in FMODR to 1 to enable the command, and then specify the size and start address of the target area in DFLBCCNT. When the DFLBCCNT.BCSIZE bit is 1, checking can be performed on the entire area (2 Kbytes) as specified in the second cycle of the blank check command. When the BCSIZE bit is 0, checking can be performed on the 2-byte range starting from the address obtained by adding the start address of the area as specified in the second cycle of the command and the value held by DFLBCCNT. In the first cycle of the command sequence, the value 71h is written to an address in the E2 DataFlash. In the second cycle, when the value D0h is written to an address in the erasure block within the target area, the FCU starts blank checking of the E2 DataFlash. Test the FRDY bit in the FSTATR0 register to check whether or not the check is complete. On completion of blank checking, check the BCST bit of DFLBCSTAT to see whether the target area has been erased or is filled with 0s and/or 1s.

Figure 43.17 shows the procedure for blank checking of the E2 DataFlash.





**Figure 43.17 Procedure for Blank Checking of the E2 DataFlash**

### 43.4.4.3 Suspension and Resumption

#### (1) Suspending Programming or Erasure

To suspend programming/erasure for the ROM/E2 DataFlash, use the P/E suspend command.

When issuing a P/E suspend command, check that the ILGLERR, ERSERR, and PRGERR bits in FSTATR0 and the FSTATR1.FCUERR bit are 0, and the execution of programming/erasure is normally performed. To confirm that the suspend command can be received, also check that the FSTATR0.SUSRDY bit is 1. After issuing a P/E suspend command, read FSTATR0 and FSTATR1 to confirm that no error occurs.

If an error occurs during programming/erasure, at least one of the ILGLERR, PRGERR, ERSERR, and FCUERR bits is set to 1. When programming/erasure processing has finished during the interval from when the SUSRDY bit is checked as 1 to when a P/E suspend command is received, the ILGLERR bit is set to 1 because the issued P/E suspend command is detected as an illegal command.

When programming/erasure processing has finished simultaneously with the reception of a P/E suspend command, no error occurs and the suspended state is not entered (FSTATR0.FRDY bit is 1 and ERSSPD and PRGSPD bits in FSTATR0 are 0). When a P/E suspend command is received and then the programming/erasure suspend processing finishes normally, the FCU enters the suspended state, the FRDY bit is set to 1, and the ERSSPD or PRGSPD bit is set to 1. After issuing a P/E suspend command, check that the FRDY bit is set to 1, and the ERSSPD or PRGSPD bit is 1 and the FCU enters the suspended state, and then decide the subsequent flow. When issuing a P/E resume command in the

subsequent flow although the FCU does not enter the suspended state, an illegal command error occurs and the FASTAT.CMDLK bit is set to 1 (command-locked state) (see section 43.6.2, Command-Locked State). If the erasure suspended state is entered, programming to blocks other than an erasure target can be performed. Additionally, the programming and erasure suspended states can shift to ROM read mode by clearing FENTRYR. For details on FCU operations at the reception of a P/E suspend command, see section 43.5, Suspending Operation.

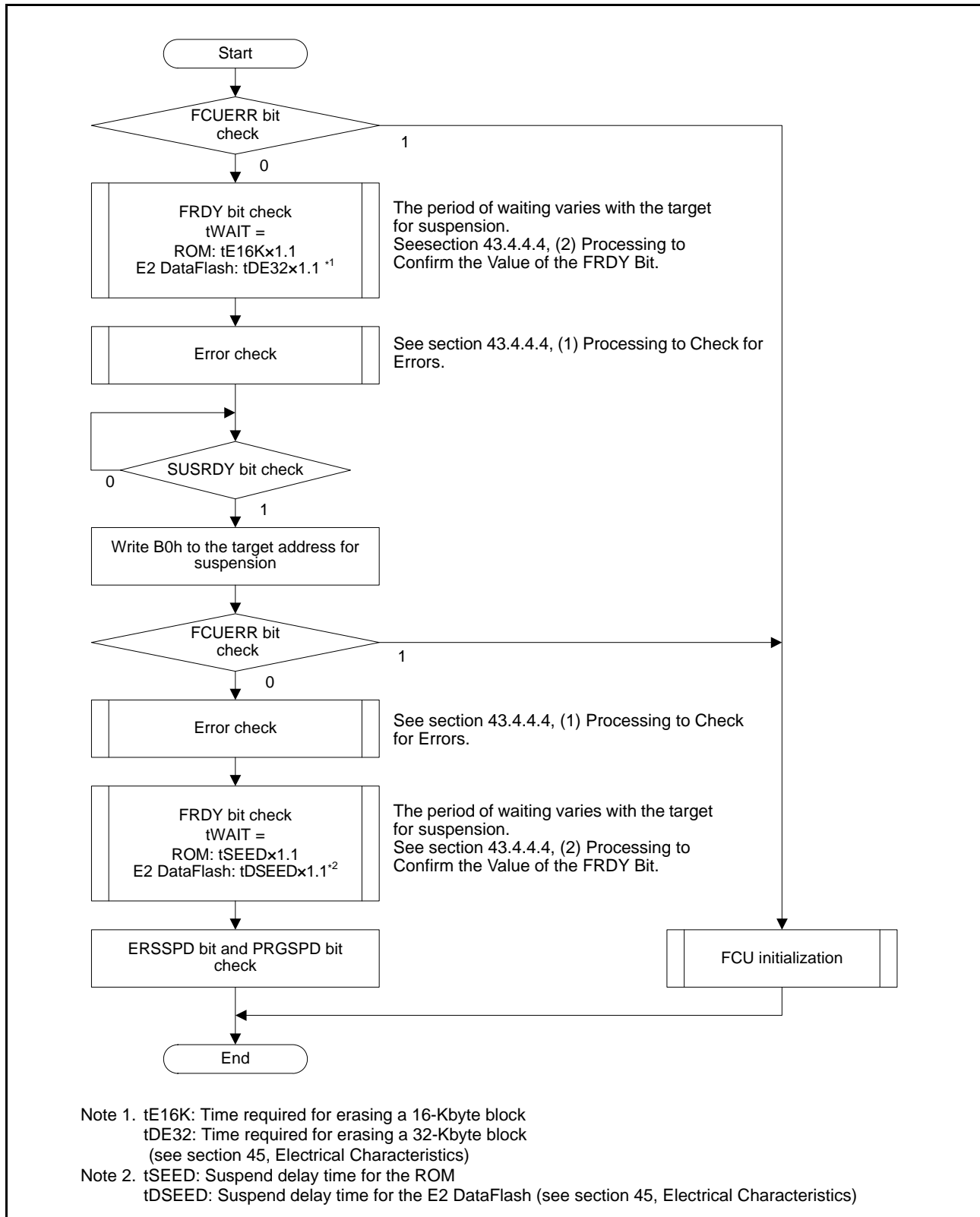
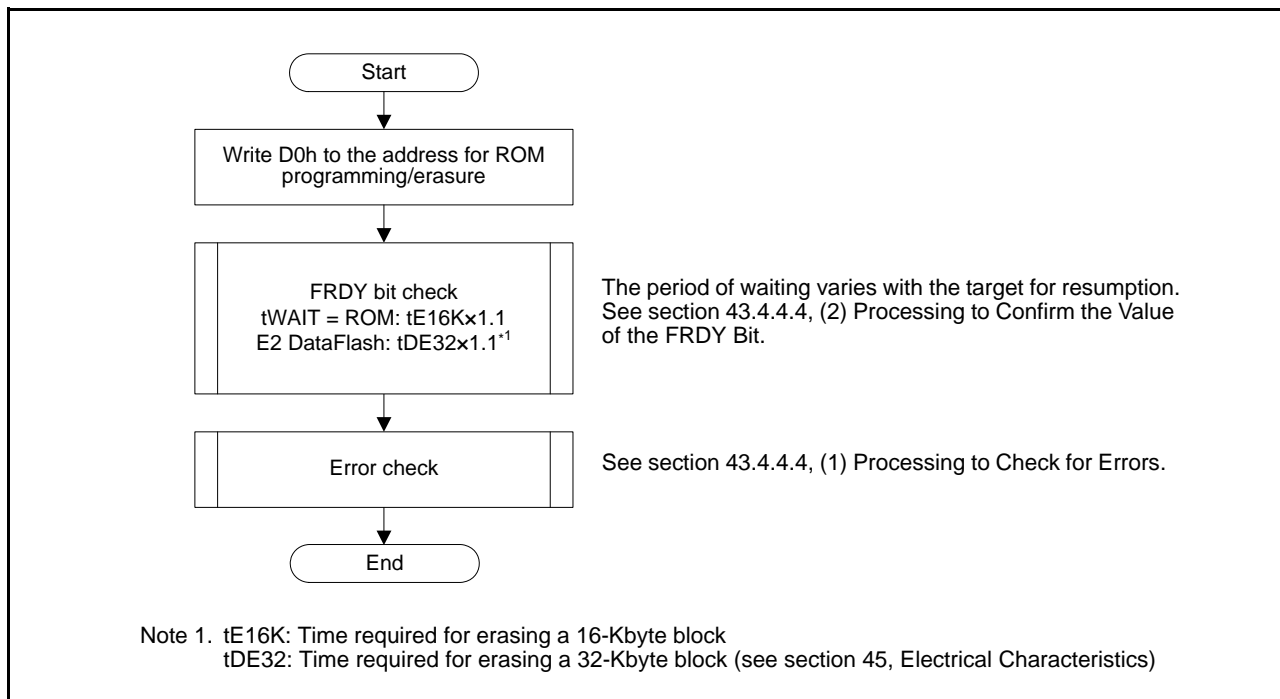


Figure 43.18 Procedure for Programming/Erasure Suspension

## (2) Resuming Programming or Erasure

To resume a suspended programming/erasure processing, use the P/E resume command. When the settings of FENTRYR are changed during suspension, reset FENTRYR to the value immediately before the P/E suspend command was issued, and then issue a P/E resume command.



**Figure 43.19 Procedure for Resuming Programming or Erasure**

#### 43.4.4.4 Processing to Check for Errors and to Confirm the Value of the FRDY Bit

The following passages describe processing to check for errors and processing to confirm the value of the FRDY bit. For details on errors, see section 43.6, Protection.

##### (1) Processing to Check for Errors

The values of the FSTATR0 and FSTATR1 registers can be read to check whether or not an error has occurred.

Setting the FRESETR.FRESET bit to 1 while the value of the FSTATR1.FCUERR bit is also 1 leads to initialization of the FCU. In such cases, re-copy the firmware from the FCU firmware area to the FCU RAM.

- Checking Flash Status Register 0 (FSTATR0)

To check FSTATR0, read FSTATR0 directly or read the ROM programming/erasure address in ROM status read mode. For the reading in ROM status read mode, see section 43.4.4.1, (4) Switching to Status Read Mode.

- Clearing Flash Status Register 0 (FSTATR0)

To clear the ILGLERR, ERSERR and PRGERR bits in FSTATR0 to 0, use the status register clear command.

When one of the ILGLERR, ERSERR and PRGERR bits in FSTATR0 is 1, the FASTAT.CMDLK bit is set to 1 (command-locked state) and the FCU receives no FCU commands other than the status register clear command. If the ILGLERR bit is 1, also check the values of the ROMAE, DFLAE, DFLRPE, and DFLWPE bits in FASTAT. Even if issuing a status register clear command without clearing these bits, the ILGLERR bit is not cleared to 0.

Figure 43.20 shows the flow of processing to check for errors and of the subsequent processing.

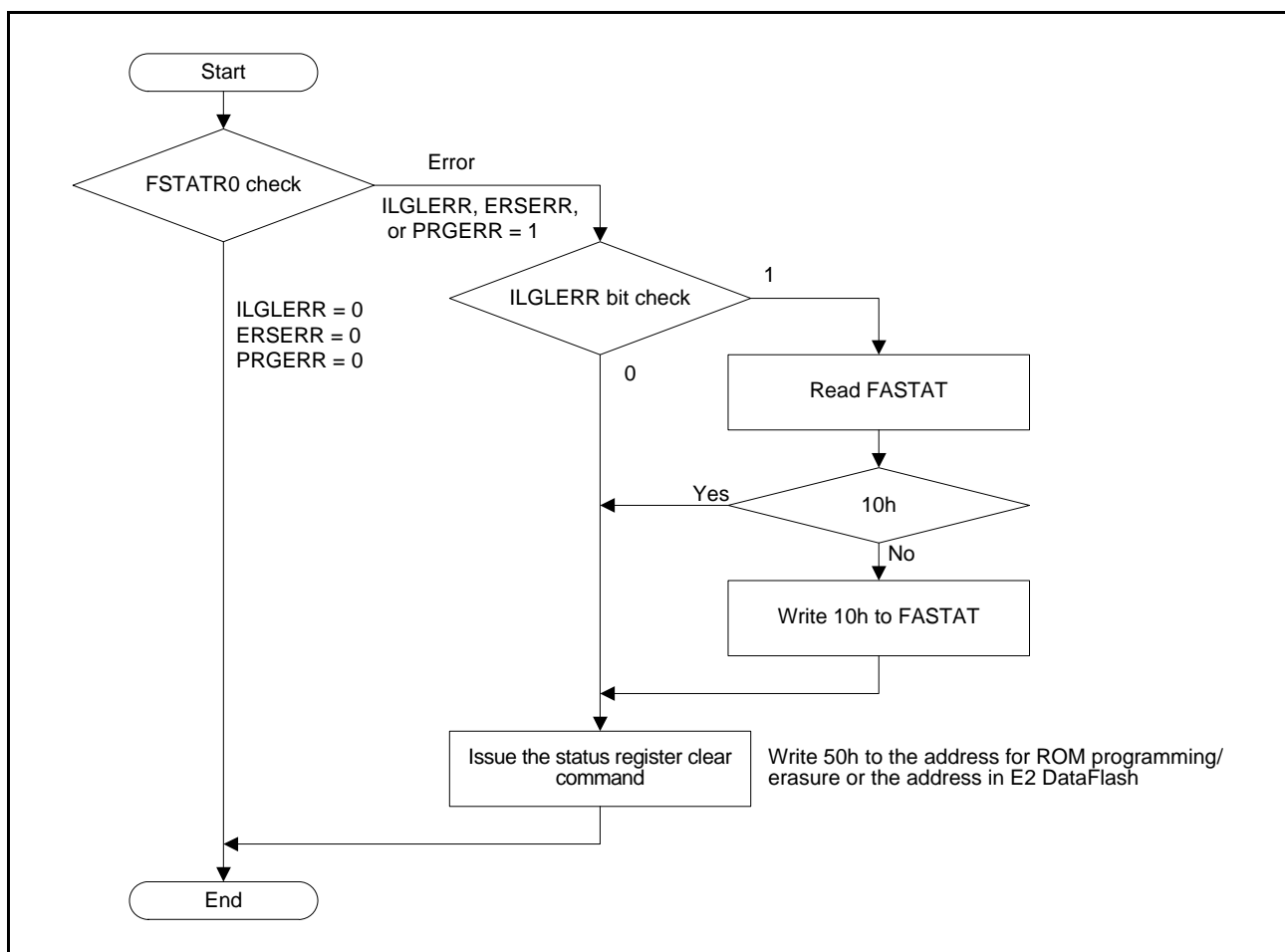


Figure 43.20 Processing to Check for Errors

## (2) Processing to Confirm the Value of the FRDY Bit

A period of waiting for processing is required after an FCU command is issued. For details on the period of waiting, see section 45, Electrical Characteristics. When a timeout leads to the FSTAT0.FRDY bit not being set to 1, FRESETR must be used to initialize the FCU. For the initialization of the FCU, see section (3), Initializing the FCU.

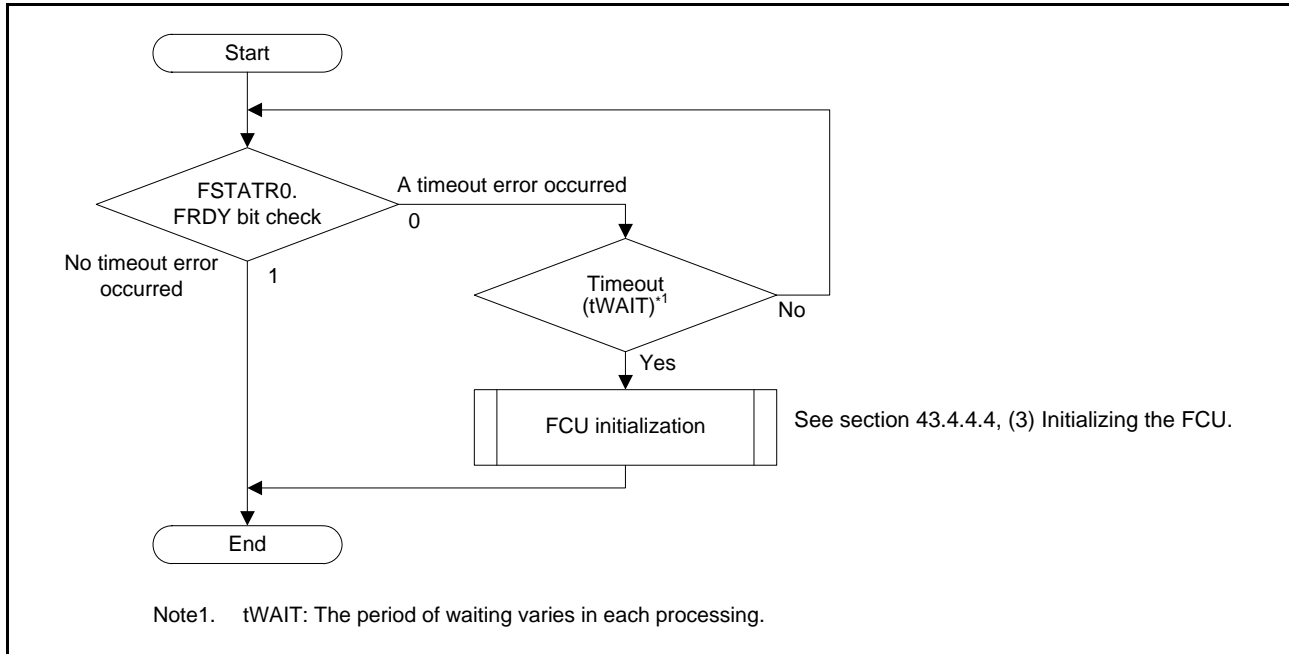


Figure 43.21 Processing to Confirm the Value of the FRDY Bit

## (3) Initializing the FCU

If a timeout occurs after an FCU command has been issued, or the FSTAT0.FRDY bit is 1, FRESETR must be used to initialize the FCU. In either case, maintain the FRESETR.FRESET bit set to 1 for a period of  $t_{FCR}$  (FCU reset time, see section 45, Electrical Characteristics). Disable reading from the ROM and E2 DataFlash memory during this period of keeping the FRESET bit set to 1. In addition, while the FRESET bit is 1, FCU commands are disabled because FENTRYR is initialized. Restart the processing from the start, as shown in Figure 43.10.

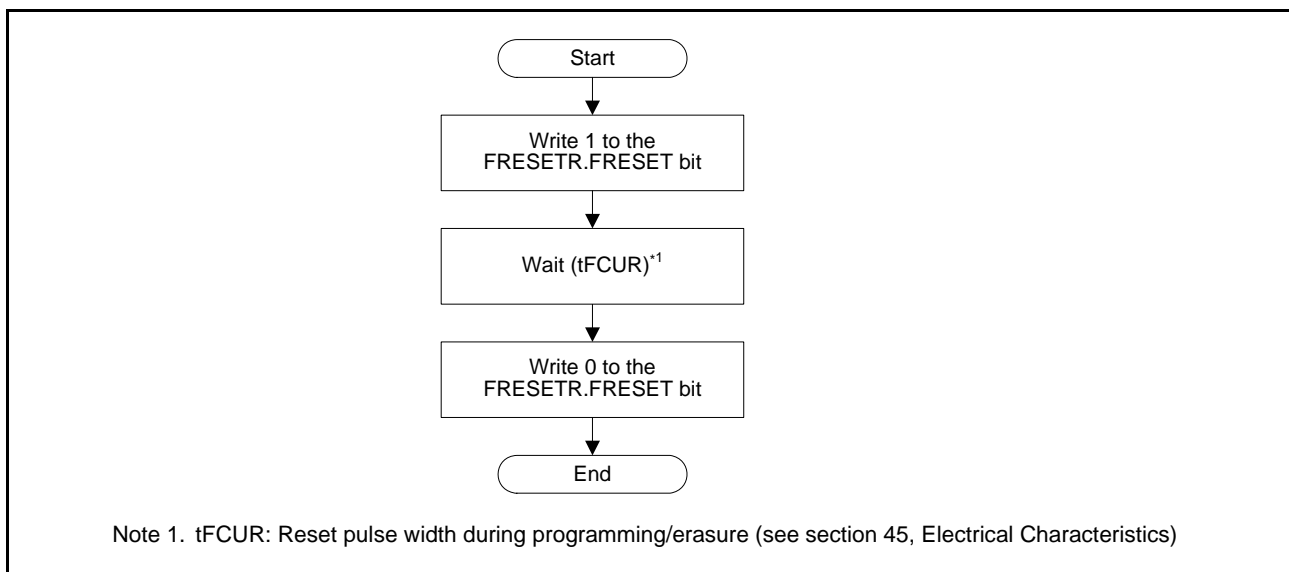


Figure 43.22 FCU Initialization Flow

## 43.5 Suspending Operation

The ROM/E2 DataFlash cannot be read during programming/erasure. The ROM/E2 DataFlash can be read by suspending the ROM/E2 DataFlash programming/erasure with the P/E suspend command. The P/E suspend command includes one programming mode and two erasure modes (suspension priority mode and erasure priority mode). The P/E resume command that resumes suspended programming/erasure processing is also provided.

### 43.5.1 Suspension during Programming

When issuing a P/E suspend command during the ROM/E2 DataFlash programming, the FCU suspends programming processing. Figure 43.23 shows the suspend operation of programming.

When receiving a programming-related command, the FCU clears the FSTATR0.FRDY bit to 0 to start programming. If the FCU enters the state in which the P/E suspend command can be received after starting programming, it sets the FSTATR0.SUSRDY bit to 1. When a P/E suspend command is issued, the FCU receives the command and clears the SUSRDY bit to 0. If the FCU receives a P/E suspend command while a programming pulse is being applied, the FCU continues applying the pulse. After the specified pulse application time, the FCU finishes pulse application, and starts the programming suspend processing and sets the FSTATR0.PRGSPD bit to 1. When the suspend processing finishes, the FCU sets the FRDY bit to 1 to enter the programming suspended state. When receiving a P/E resume command in the programming suspended state, the FCU clears the FRDY and PRGSPD bits to 0 and resumes programming.

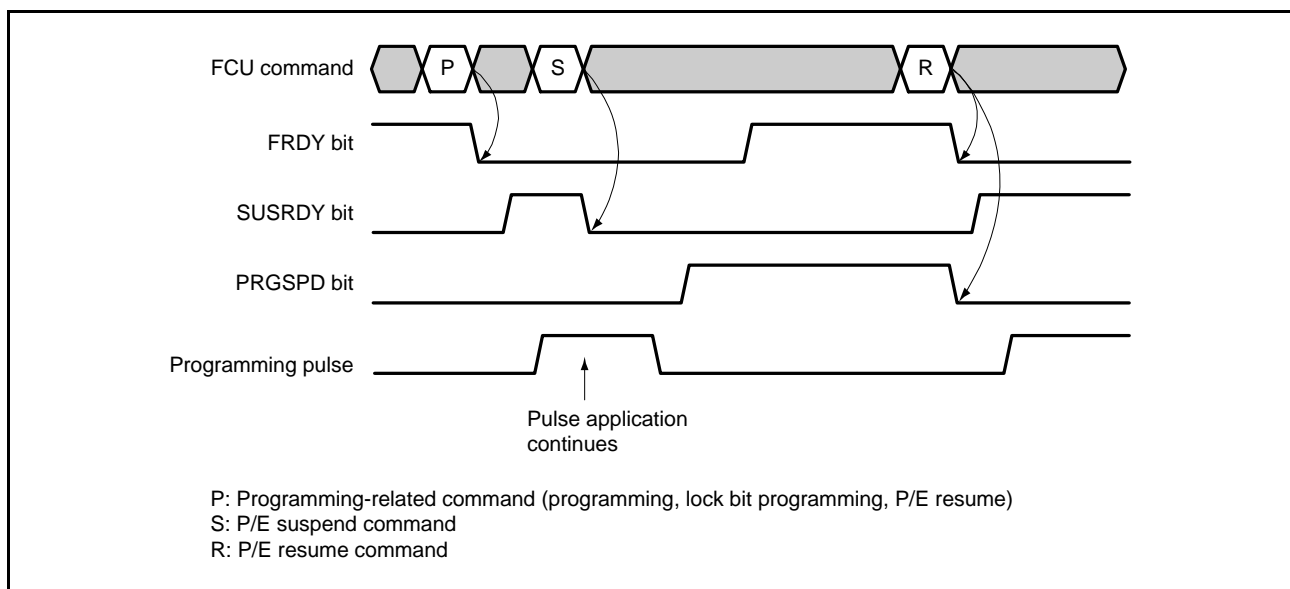


Figure 43.23 Suspension during Programming

### 43.5.2 Suspension during Erasure (Suspension Priority Mode)

The RX630 has a suspension priority mode for the suspension of erasure.

Figure 43.24 shows the suspend operation of erasure when the erasure suspend mode is set to the suspension priority mode (FCPSR.ESUSPMD bit is 0).

When receiving an erasure-related command, the FCU clears the FSTATR0.FRDY bit to 0 to start erasure. If the FCU enters the state in which the P/E suspend command can be received after starting erasure, it sets the FSTATR0.SUSRDY bit to 1. When a P/E suspend command is issued, the FCU receives the command and clears the SUSRDY bit to 0. When receiving a suspend command during erasure, the FCU starts the suspend processing and sets the FSTATR0.ERSSPD bit to 1 even if it is applying an erasure pulse. When the suspend processing finishes, the FCU sets the FRDY bit to 1 to enter the erasure suspended state. When receiving a P/E resume command in the erasure suspended state, the FCU clears the FRDY and ERSSPD bits to 0 and resumes erasure. Operations of the FRDY, SUSRDY, and ERSSPD bits at the suspension and resumption of erasure are the same, regardless of the erasure suspend mode.

The setting of the FCPSR.ESUSPMD bit affects the control method of erasure pulses. In suspension priority mode, when receiving a P/E suspend command while erasure pulse A that has never been suspended in the past is being applied, the FCU suspends the application of erasure pulse A and enters the erasure suspended state. When receiving a P/E suspend command while reapplying erasure pulse A after erasure is resumed by a P/E resume command, the FCU continues applying erasure pulse A. After the specified pulse application time, the FCU finishes erasure pulse application and enters the erasure suspended state. When the FCU receives a P/E resume command next and erasure pulse B starts to be newly applied, and then the FCU receives a P/E suspend command again, the application of erasure pulse B is suspended. In suspension priority mode, delay due to suspension can be minimized because the application of an erasure pulse is suspended one time per pulse and priority is given to the suspend processing.

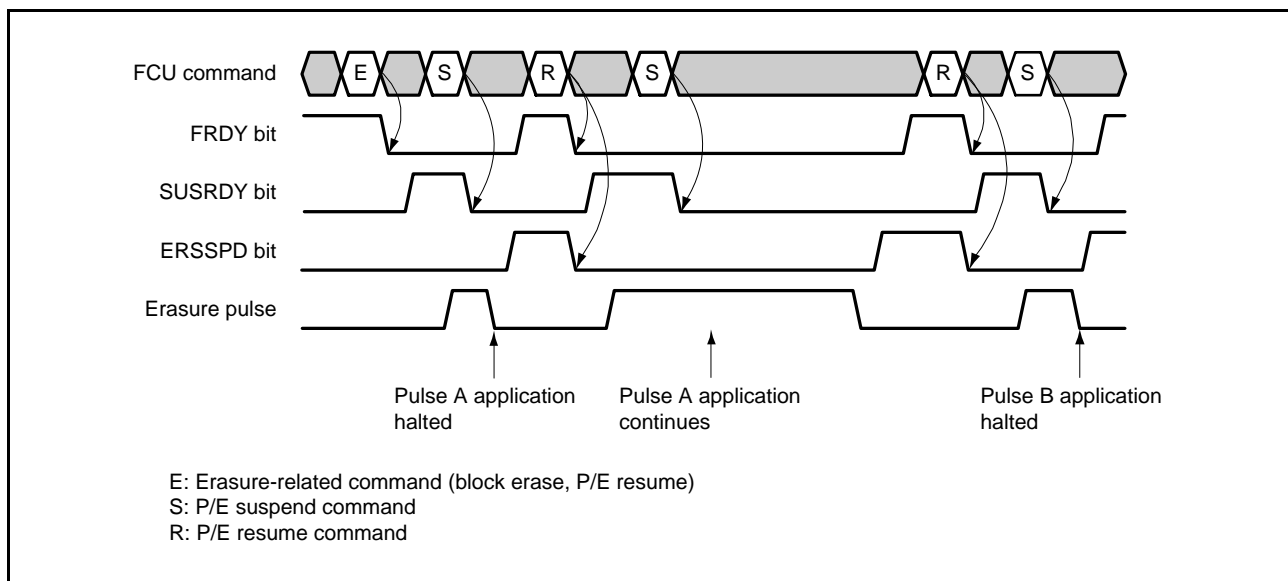


Figure 43.24 Suspension during Erasure (Suspension Priority Mode)

### 43.5.3 Suspension during Erasure (Erasure Priority Mode)

The RX630 has a suspension priority mode for the suspension of erasure.

Figure 43.25 shows the suspend operation of erasure when the erasure suspend mode is set to the erasure priority mode (FCPSR.ESUSPMD bit is 1). The control method of erasure pulses in erasure priority mode is the same as that of programming pulses for the programming suspend processing.

If the FCU receives a P/E suspend command while an erasure pulse is being applied, the FCU definitely continues applying the pulse. In this mode, the required time for the whole erasure processing can be reduced as compared with the suspension priority mode because the reapplication of erasure pulses does not occur when a P/E resume command is issued.

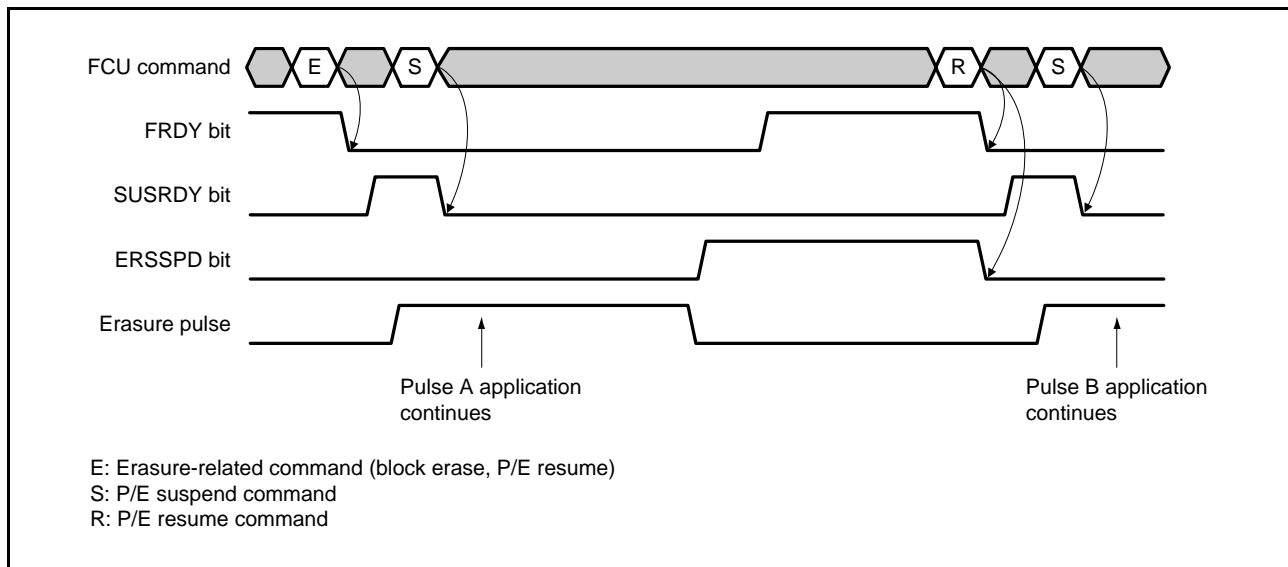


Figure 43.25 Suspension during Erasure (Erasure Priority Mode)

## 43.6 Protection

Protection against programming/erasure for the ROM/E2 DataFlash includes software protection and the command-locked state.

### 43.6.1 Software Protection

With the software protection, the ROM/E2 DataFlash programming/erasure is prohibited by the settings of the control registers or user area lock bit. When the software protection is violated and a ROM/E2 DataFlash programming/erasure-related command is issued, the FCU detects an error and the FASTAT.CMDLK bit is set to 1 (command-locked state).

#### (1) Protection through FWEPROR

If the FWEPROR.FLWE[1:0] bits are not set to 01b, programming cannot be performed in any of the modes.

#### (2) Protection through FENTRYR

When the FENTRYR.FENTRY3, FENTRY2, FENTRY1, FENTRY0 bits are 0, ROM/E2 DataFlash read mode is selected. Because the FCU command cannot be received in ROM/E2 DataFlash read mode, ROM/E2 DataFlash programming/erasure is prohibited. When an FCU command is issued in ROM/E2 DataFlash read mode, the FCU detects an illegal command error and the FASTAT.CMDLK bit is set to 1 (command-locked state) (see section 43.6.2, Command-Locked State).



### (3) Protection through Lock Bit

Each block in the user area includes a lock bit. When the FPROTR.FPROTCN bit is 0, blocks whose lock bit is set to 0 are prohibited from being programmed/erased. To program or erase blocks whose lock bit is set to 0, set the FPROTCN bit to 1. When the lock bit protection is violated and a ROM programming/erasure-related command is issued, the FCU detects a programming/erasure error and the FASTAT.CMDLK bit is set to 1 (command-locked state) (see section 43.6.2, Command-Locked State).

### (4) Protection through DFLWE<sub>y</sub>

When the DBWE<sub>j</sub> ( $j = 00$  to 15) bit in DFLWE<sub>y</sub> ( $y = 0, 1$ ) is 0, programming and erasure of block DB<sub>j</sub> in the data area is disabled. If an attempt is made to program or erase block DB<sub>j</sub> while the DBWE<sub>j</sub> bit is 0, the FCU detects a programming/erasure protection error and the FASTAT.CMDLK bit is set to 1 (command-locked state) (see section 43.6.2, Command-Locked State).

### (5) Protection through DFLRE<sub>y</sub>

When the DBRE<sub>j</sub> ( $j = 00$  to 15) bit in DFLRE<sub>y</sub> ( $y = 0, 1$ ) is 0, reading of block DB<sub>j</sub> in the data area is disabled. If an attempt is made to read block DB<sub>j</sub> while the DBRE<sub>j</sub> bit is 0, the FCU detects a read protection error and the FASTAT.CMDLK bit is set to 1 (command-locked state) (see section 43.6.2, Command-Locked State).

## 43.6.2 Command-Locked State

With the command-locked state, the FCU detects malfunctions caused by FCU command issuance errors and prohibited access occurrences, and an FCU command is prohibited from being received.

When any bit from among the status bits (the ILGLERR, ERSERR, and PRGERR bits in FSTATR0, the FSTATR1.FCUERR bit, and the ROMAE, DFLAE, DFLRPE and DFLWPE bits in FASTAT), the FCU will be in the command-locked state (FASTAT.CMDLK bit is set to 1), so programming and erasure of the ROM/E2 DataFlash are prohibited. To clear the command-locked state, a status register clear command must be issued with FASTAT set to 10h. While the interrupt enable bit in FAEINT is 1, if the corresponding bit in FASTAT is set to 1, a flash interface error (FIFERR) interrupt occurs.

Table 43.12 lists the relationship between the error contents and status bit values (FSTATR0.ILGLERR, ERSERR, and PRGERR bits, FSTATR1.FCUERR bit, and FASTAT.ROMAE, DFLAE, DFLRPE, and DFLWPE bits) at error detection. If a command other than the suspend command is issued during programming/erasure and the FCU enters the command-locked state, it continues programming/erasure. In this state, it is impossible to issue a P/E suspend command and suspend programming/erasure. When a command is issued in the command-locked state, the ILGLERR bit is set to 1.

Table 43.12 Errors and Status Bits

Type	Description	ILGLERR	ERSERR	PRGERR	FCUERR	ROMAE	DFLAE	DFLRPE	DFLWPE
FENTRYR setting error	FENTRYR setting is other than 0001h, 0002h*1, 0004h*1, and 0080h*1.	1	0	0	0	0	0	0	0
	The FENTRYR setting at suspension disagrees with that at resumption	1	0	0	0	0	0	0	0
Illegal command error (Common to ROM/ E2 DataFlash)	Undefined code is specified in the first cycle of an FCU command	1	0	0	0	0	0	0	0
	Other than D0h is specified in the last cycle of a multi-cycle FCU command	1	0	0	0	0	0	0	0
	The peripheral clock is set to other than 1 to 100 MHz in PCKAR (an error is not detected if the setting is from 1 to 4 MHz or from 50 to 100 MHz)	1	0	0	0	0	0	0	0
	A command other than the suspend command is issued during programming/erasure	1	0	0	0	0	0	0	0
	A suspend command is issued during processing other than programming/ erasure	1	0	0	0	0	0	0	0
	A suspend command is issued in the suspended state	1	0	0	0	0	0	0	0
	A resume command is issued in other than the suspended state	1	0	0	0	0	0	0	0
	A programming/erasure-related command (programming/lock bit programming/block erase) is issued in the programming suspended state	1	0	0	0	0	0	0	0
	A block erase command is issued in the erasure suspended state	1	0	0	0	0	0	0	0
	A programming or lock bit programming command is issued to an erasure suspend target area in the erasure suspended state	1	0	0	0	0	0	0	0
	A command is issued when the FASTAT.CMDLK bit is 1 (in the command- locked state)	1	0/1	0/1	0/1	0/1	0	0	0
Illegal command error (ROM)	Other than 40h is specified in the second cycle of a programming command	1	0	0	0	0	0	0	0
Erasure error	An error occurs during erasure	0	1	0	0	0	0	0	0
	When the FPROTR.FPROTCN bit is 0, a block erase command is issued to an erasure block whose lock bit is set to 0	0	1	0	0	0	0	0	0
Illegal command error (E2 DataFlash)	Other than 01h is specified in the second cycle of a programming command	1	0	0	0	0	0	0	0
	When the FENTRYR.FENTRYD bit is 1, lock bit programming command is issued to the E2 DataFlash area when the FENTRYR.FENTRYD bit is 1	1	0	0	0	0	0	0	0
Programming error	An error occurs during programming	0	0	1	0	0	0	0	0
	When the FPROTR.FPROTCN bit is 0, a programming or lock bit programming command is issued to a block whose lock bit is set to 0	0	0	1	0	0	0	0	0
FCU error	An error occurs during FCU internal processing	0	0	0	1	0	0	0	0
ROM access violation	When the area n (n = 0 to 3) is in the ROM P/E mode (FENTRYR.FENTRYn bit = 1), an attempt is made to read a read address in the area n*1	1	0	0	0	1	0	0	0
	An FCU command was issued for the address for programming/erasure in the area n (n = 0 to 3) while the area n is in read mode (FENTRYR.FENTRYn bit = 0)	1	0	0	0	1	0	0	0
	An attempt is made to read the area n (n = 0 to 3) while in the ROM P/E mode (the FENTRYR register is set)	1	0	0	0	1	0	0	0
E2 DataFlash access error	An attempt is made to read the E2 DataFlash area while in the E2 DataFlash P/E normal mode (FENTRYR.FENTRYD bit = 1)	1	0	0	0	0	1	0	0
	An FCU command is issued for the E2 DataFlash area while FENTRYR.FENTRYD is 0	1	0	0	0	0	1	0	0
	An FCU command is issued for the E2 DataFlash area while any of the FENTRYR.FENTRYn bits (n = 0 to 3) is 1	1	0	0	0	0	1	0	0
E2 DataFlash read protect error	An attempt is made to read the E2 DataFlash area while it is protected against reading by the DFLREy (y = 0, 1) setting	1	0	0	0	0	0	1	0
E2 DataFlash programming protect error	A programming/erasure command is issued for the E2 DataFlash area while it is protected against programming and erasure by the DFLWEy (y = 0, 1) setting	1	0	0	0	0	0	0	1

Note 1. For the correspondence between ROM capacity and areas, see section 43.1.1, Configuration of the ROM Area.

### 43.7 User Boot Mode

If the low level is on the MD pin and the high level is on the PC7 pin at the time of release from the reset state, the chip starts in user boot mode. The reset vector at this time points to the address FF7F FFFCh of the user boot area. For other vector tables, refer to normal vector table (see section 15, Interrupt controller (ICUb)).

In user boot mode, it is possible to perform programming using a given interface; user area or data area can be programmed or erased by issuing the FCU command. Note that programming to the user boot area should be performed in boot mode.

### 43.8 Boot Mode

#### 43.8.1 System Configuration

In boot mode, the host sends control commands and data for programming, and the user area, data area, and user boot area are programmed or erased accordingly. An on-chip SCI handles transfer between the host and RX630 in asynchronous mode. Tools for transmission of control commands and the data for programming must be prepared in the host.

When the RX630 is activated in boot mode, the program on the boot area is executed. This program automatically adjusts the bit rate of the SCI and controls programming/erasure by receiving control commands from the host.

Figure 43.26 shows the system configuration for operations in boot mode.

Input and output pins associated with the ROM/E2 DataFlash are listed in Table 43.13.

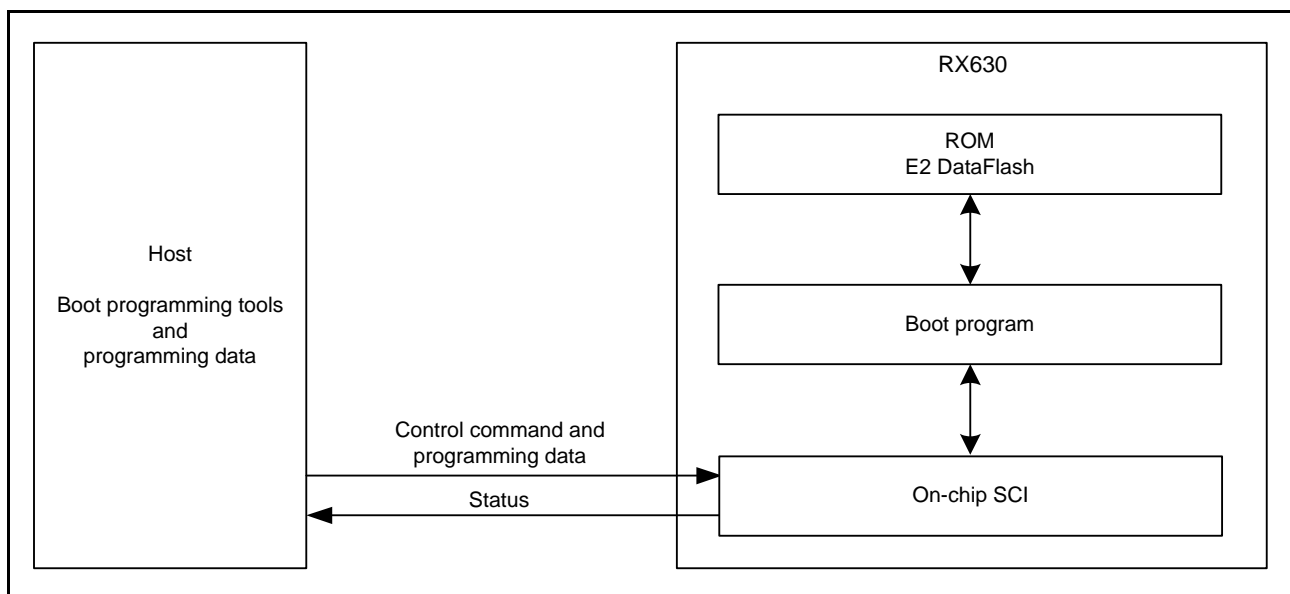


Figure 43.26 System Configuration for Operations in Boot Mode

Table 43.13 Input and Output Pins Associated with the ROM/E2 DataFlash

Pin Name	I/O	Mode to be Used	Use
MD	Input	Boot mode	Selection of operation mode
PC7	Input	User boot mode USB boot mode	Selection of boot mode (SCI boot), user boot mode, or USB boot mode
PF2/RXD1 (176-pin package) P30/RXD1 (144/100/80-pin packages)	Input	Boot mode	For host communication (to receive data through SCI)
PF0/TXD1 (176-pin package) P26/TXD1 (144/100/80-pin packages)	Output		For host communication (to transmit data through SCI)
USB0_DP, USB0_DM	I/O	USB boot mode	Data input/output of USB
P14/USB0_DPUPE	I/O		Control of pull-up for USB
P16/USB0_VBUS	Input		Detection of connection and disconnection of USB cables
P35	Input		Selection of USB bus-power mode or self-power mode

### 43.8.2 State Transitions in Boot Mode

Figure 43.27 is a diagram of the state transitions in boot mode.

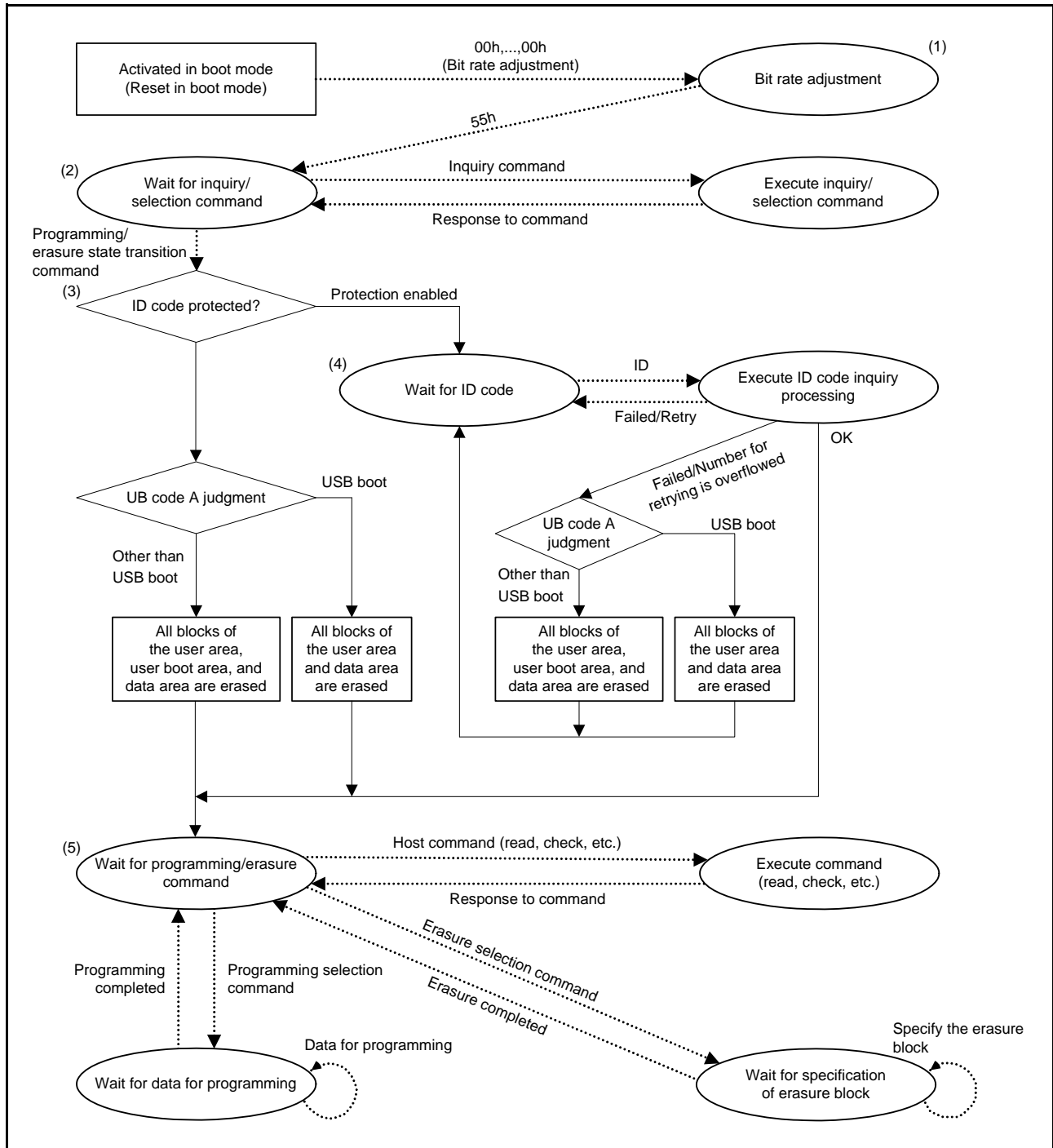


Figure 43.27 State Transitions in Boot Mode

### (1) Matching the Bit Rates

When the RX630 is activated in boot mode, the bit rate of the SCI is automatically adjusted to match that of the host. On completion of this automatic bit rate adjustment, the RX630 transmits the value 00h to the host. On subsequent correct reception of the value 55h sent from the host, the RX630 enters the wait for a command for inquiry or selection. For details on matching of the bit rates, see section 43.8.3, Automatic Adjustment of the Bit Rate.

### (2) Waiting for a Command for Inquiry or Selection

This state is for inquiries on the area size, the area configuration, the addresses where the areas start, the state of support, etc., and for selection of the device, clock mode, and bit rate. The RX630 receives a programming/erasure state transition command issued by the host and then enters the state to determine whether ID code protection is enabled or disabled. For the inquiry/selection commands, see section 43.8.7, Inquiry/Selection Command Wait.

### (3) Judging ID Code Protection

This state is for determining whether ID code protection is enabled or disabled. The control code and ID code written in ROM are used to determine whether ID code protection is enabled or disabled. When enabled, the state of waiting for the ID code is entered. When disabled, the user area and data area are completely erased, and the wait for programming and erasure commands is entered. For details on the control code and ID code, see section 43.8.4, ID Code Protection (Boot Mode).

### (4) Waiting for an ID Code

This state is for waiting for the control code and ID code to be sent from the host. The control code and ID code sent by the host are compared with the code stored in ROM, and the state of waiting for programming and erasure commands is entered if the two match. If they do not match, the next transition is back to the state of waiting for an ID code. However, if the ID codes fail to match three times in a row and also the state of protection is authentication method 1, the ROM is completely erased, and the state of waiting for an ID code is entered again. Turn the power off and start all over. For details on the control code and ID code, see section 43.8.4, ID Code Protection (Boot Mode).

### (5) Waiting for a Command for Programming or Erasure

In this state, programming and erasure proceed in accordance with commands from the host. In response to the reception of a command, the RX630 enters the wait for the data to use in programming, the wait for specification of the erasure block to be erased, or the state of executing the processing of commands, such as read and check.

When the RX630 receives a programming selection command, it enters the state of waiting for the data to use in programming. After the host has issued the programming selection command, the process continues with the address where programming is to start and then the data for programming. Setting of FFFF FFFFh as the address where programming is to start indicates the completion of programming, and the next transition is from the wait for the data to use in programming to the wait for programming and erasure commands.

When the RX630 receives an erasure selection command, it enters the state of waiting for specification of the erasure block to be erased. After the host has issued the erasure selection command, the process continues with the number of the erasure block to be erased. Setting of FFh as the number of the erasure block indicates the completion of erasure, and the next transition is from the wait for specification of the erasure block to the wait for programming and erasure commands. Since the user area, user boot area and data area are all completely erased during the interval between booting up in boot mode and transition to the wait for programming and erasure commands, execution of erasure is not necessary unless data newly programmed in boot mode is to be erased without a further reset.

Other than the programming and erasure commands, commands for execution in this state include those for checksum of the user area and user boot area, blank checking, reading from memory, and acquiring status information.

### 43.8.3 Automatic Adjustment of the Bit Rate

When the RX630 is booted up in boot mode, asynchronous transfer by the SCI is used to measure the periods at low level of consecutive bytes with value 00h that are sent from the host. While the period at low level is being measured, set the host's SCI transfer format to 8-bit data, one stop bit, no parity, and a transfer rate of 9,600 bps or 19,200 bps. The RX630 calculates the host's SCI bit rate from the measured periods at low level, adjusts its own bit rate accordingly, and then sends the value 00h to the host.

If reception of the value 00h by the host is successful, the host responds by sending the value 55h to the RX630. If successful reception of 00h by the host is not possible, reboot the RX630 in boot mode, and then repeat the process of automatically adjusting the bit rate. If reception of the value 55h by the RX630 is successful, it responds by sending E6h to the host, and if successful reception of 55h by the RX630 is not possible, it responds by sending FFh to the host.

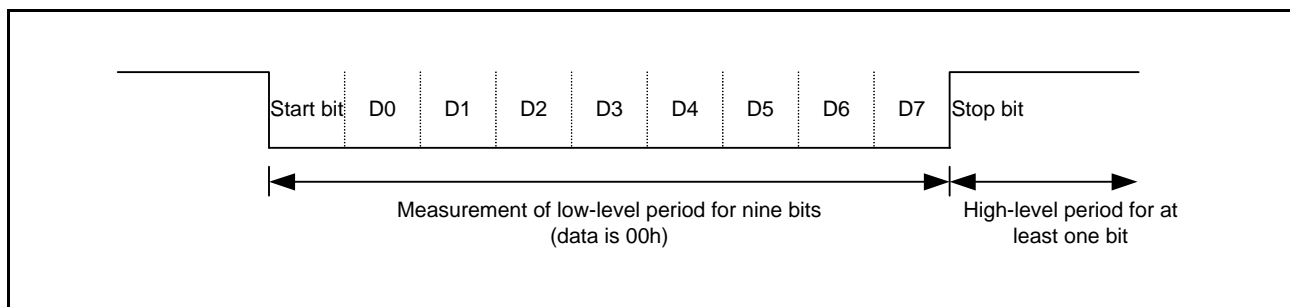


Figure 43.28 Transfer Format Used by SCI in Automatic Adjustment of Bit Rate

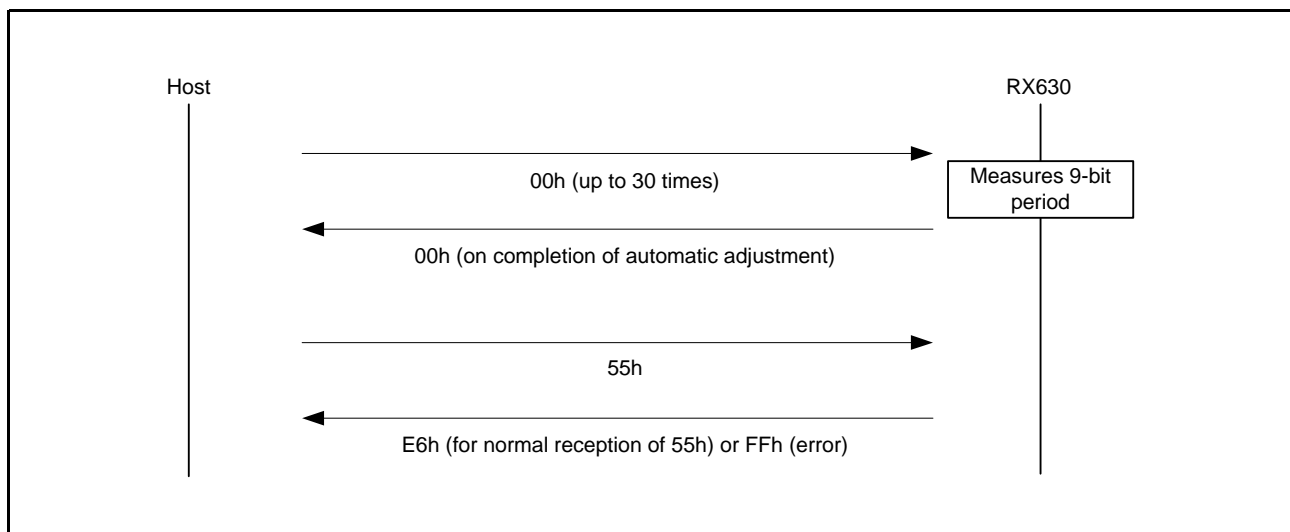


Figure 43.29 Sequence of Transfer between Host and RX630

Since the bit rate of the RX630 depends on the bit rate of SCI communications and the frequency of the RX630's peripheral clock, adjustment to match the bit rate will not be possible under some conditions. Accordingly, ensure that SCI communication is under the conditions given in Table 43.14.

Table 43.14 Conditions for Automatic Bit-Rate Adjustment

Bit Rate of SCI in Host	Frequency Range for EXTAL Signal
9,600 bps	4 to 20 MHz*1
19,200 bps	8 to 20 MHz*1

Note 1. The maximum frequency of the resonator is 16 MHz.

### 43.8.4 ID Code Protection (Boot Mode)

This function is used to prohibit reading/programming/erasure from the host such as the PC.

After automatic adjustment of the bit rate when booting up in boot mode, the ID code transmitted from the host and the control and ID codes written to the ROM are used to determine disabling or enabling of ID code protection. When ID code protection is enabled, the code sent from the host is compared with the control code and ID code in the ROM to determine whether they match, and reading/programming/erasure will be enabled only when the two match.

The control code and ID code in the ROM consists of four 32-bit words. Figure 43.30 shows the configuration of the control code and ID code. The ID code should be set in 32-bit units.

	31	24	23	16	15	8	7	0
FFFF FFA0h	Control code			ID code 1	ID code 2	ID code 3		
FFFF FFA4h	ID code 4			ID code 5	ID code 6	ID code 7		
FFFF FFA8h	ID code 8			ID code 9	ID code 10	ID code 11		
FFFF FFACh	ID code 12			ID code 13	ID code 14	ID code 15		

Figure 43.30 Configuration of Control Code and ID Code in ROM

#### (1) Control Code

The control code determines whether ID code protection is enabled or disabled and the method of authentication to use with the host. Table 43.15 lists how the control code determines the method of authentication.

Table 43.15 Specifications for ID Code Protection

Control Code	ID Code	State of Protection	Operations at the Time of SCI Connection
45h	As desired	Protection enabled (authentication method 1)	Matching ID code: The command wait is entered. Non-matching ID code: The ID code protection wait is entered again. However, if a non-matching ID code is received three times in a row, all blocks are erased.
52h	Sequences other than 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh	Protection enabled (authentication method 2)	Matching ID code: The command wait is entered. Non-matching ID code: The ID code protection wait is entered again.
	50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh	Protection enabled (authentication method 3)	Always judged to be a non-matching ID code.
Other than above	—	Protection disabled	All blocks are erased.

#### (2) ID Code

The ID code can be set to any desired value. However, if the control code is 52h and the ID code is 50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh (from the ID code 1 field), there is no determination of matching and the ID code is always considered to be non-matching. Accordingly, reading, programming, and erasure from the host are prohibited.

### (3) Program Example for ID Code Setting

The following assembler directives set up a control code of 45h and an ID code of 01h, 02h, 03h, 04h, 05h, 06h, 07h, 08h, 0Ah, 0Bh, 0Ch, 0Dh, 0Eh, 0Fh (from the ID code 1 field).

```
.SECTION ID_CODE,CODE
.ORG 0FFFFFFA0h
.LWORD 45010203h
.LWORD 04050607h
.LWORD 08090A0Bh
.LWORD 0C0D0E0Fh
```

#### 43.8.5 UB Code A

For the UB Code A, see section 7.3, UB Codes.

#### 43.8.6 Configuration of Commands and Responses

Communications between the host and the RX630 in boot mode consist of commands sent by the host and responses sent as replies from the RX630.

The notation "SUM" in descriptions of commands means the checksum, in the case of the total of the bytes transmitted by the RX630, and byte data for which the total becomes 00h is indicated. "Size" indicates the size of the command from the byte where it starts, and is the number of bytes transmitted or received other than the SUM byte.

Moreover, if the host has sent an undefined command, the RX630 returns a response indicating a command error in the format shown below. The command field holds the first byte of the undefined command sent from the host.

Error response

80h	Command
-----	---------



### 43.8.7 Inquiry/Selection Command Wait

Table 43.16 lists the commands available in the inquiry/selection command wait. The boot program status inquiry command can also be used in the programming/erasure command wait. The other commands can only be used in the inquiry/selection command wait.

**Table 43.16 Inquiry/Selection Commands**

Host Command Name	Function
Supported device inquiry	Inquires regarding the device codes and the series name
Device selection	Selects a device code
Clock mode inquiry	Inquires regarding the number of clock modes and their values
Clock mode selection	Notifies the selected clock mode
Multiplication ratio inquiry	Inquires regarding the number of clock types, the number of multiplication/division ratios, and the multiplication /division ratios
Operating frequency inquiry	Inquires regarding the number of clock types and the maximum and minimum operating frequencies
User boot area information inquiry	Inquires regarding the number of user boot area and the start and end addresses
User area information inquiry	Inquires regarding the number of user area and the start and end addresses
Block information inquiry	Inquires regarding the number of blocks and the start and end addresses
Programming size inquiry	Inquires regarding the size of programming data
Data area inquiry	Inquires regarding the availability of data area
Data area information inquiry	Inquires regarding the number of data areas and the start and end addresses
New bit rate selection	Modifies the bit rate of SCI communications between the host and RX630
Programming/erasure state transition	Enters the state for determining ID code protection
Boot program status inquiry	Inquires regarding the processing state

In the inquiry/selection command wait, send selection commands from the host in the order of device selection, clock mode selection, and new bit rate selection to set up the RX630 according to the responses to inquiry commands. Note that the supported device inquiry and clock mode inquiry commands are the only inquiry commands that can be sent before the clock mode selection command; other inquiry commands must not be issued before the clock mode selection command. If commands are issued in an incorrect order, the RX630 returns a response indicating a command error. Figure 43.31 shows an example of the procedure to use commands in the inquiry/selection command wait.

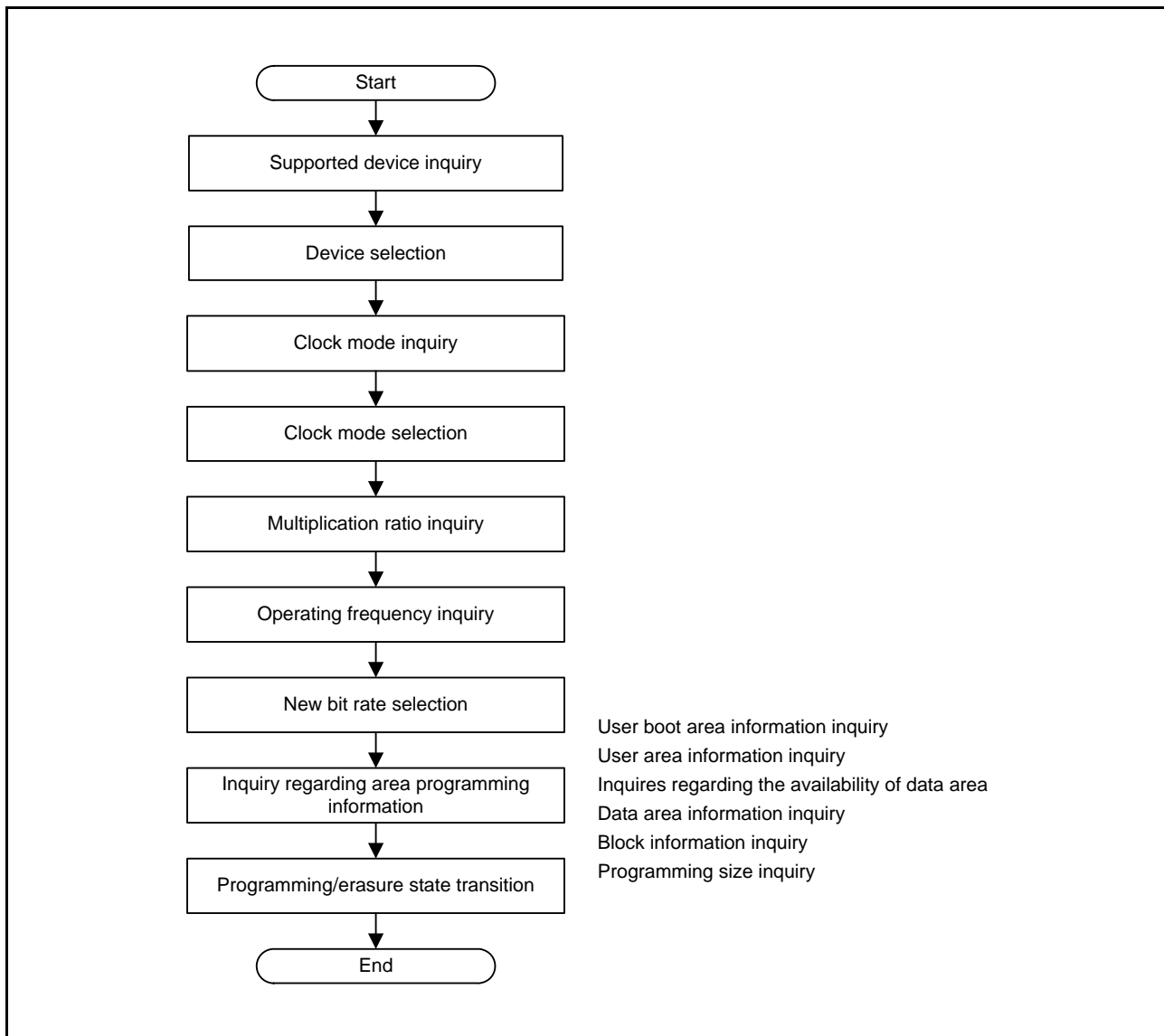


Figure 43.31 Example of Procedure to Use Inquiry/Selection Commands

## (1) Supported Device Inquiry

In response to a supported device inquiry command sent from the host, the RX630 returns the information concerning the devices supported by the boot program for boot mode. If the supported device inquiry command comes after the host has selected a device, the RX630 only returns the information concerning the selected device. In response to supported device inquiry commands, the RX630 transmits two sets of device information in turn, one in little endian and the other in big endian.

Command 

20h
-----

Response	30h	Size	Device count	
	Character count	Device code (little endian is specified)		Series name
	Character count	Device code (big endian is specified)		Series name
	SUM			

Size (1 byte): Total number of bytes in the device count, character count, device code, and series name fields  
 Device count (1 byte): Number of device types supported by the boot program for boot mode  
 Character count (1 byte): Number of characters included in the device code and series name fields  
 Device code (4 bytes): Chip recognition code  
 Series name (n bytes): ASCII code for the supported device  
 SUM (1 byte): Checksum

## (2) Device Selection

In response to a device selection command sent from the host, the RX630 checks if the selected device is supported. When the selected device is supported, the RX630 and returns a response (06h). If the selected device is not supported or the sent command is illegal, the RX630 returns an error response (90h).

Select the device code with the endian specification from the two sets of device information transmitted in response to a supported device inquiry command in accord with the written data.

Command 

10h	Size	Device code	SUM
-----	------	-------------	-----

Response 

06h
-----

Error response 

90h	Error
-----	-------

Size (1 byte): Number of characters in the device code field (fixed at 4)  
 Device code (4 bytes): ASCII code for the series name of the chip (same code as the response to the supported device inquiry command)  
 SUM (1 byte): Checksum  
 Error (1 byte): Error code  
 11h: Checksum error (illegal command)  
 21h: Device code error

### (3) Clock Mode Inquiry

In response to a clock mode inquiry command sent from the host, the RX630 returns the supported clock modes. If the clock mode inquiry command comes after the host has selected a clock mode, the RX630 only returns the information concerning the selected clock mode.

Command	21h	
Response	31h	Size
	Mode	
	SUM	

Size (1 byte): Total number of bytes in the mode count and mode fields  
 Mode (1 byte): Supported clock mode (for example, 01h indicates clock mode 1)  
 SUM (1 byte): Checksum

### (4) Clock Mode Selection

In response to a clock mode selection command sent from the host, the RX630 checks if the selected clock mode is supported. When the selected mode is supported, the RX630 specifies this clock mode for use and returns a response (06h). If the selected mode is not supported or the sent command is illegal, the RX630 returns an error response (91h). Be sure to issue a clock mode selection command only after issuing a device selection command. Even when 00h or 01h has been returned as the number of supported clock modes in response to a clock mode inquiry command, issue a clock mode selection command to specify the clock mode that has been returned as the result of the inquiry.

Command	11h	Size	Mode	SUM
Response	06h			
Error response	91h	Error		

Size (1 byte): Number of characters in the mode field (fixed at 1)  
 Mode (1 byte): Clock mode (same mode as the response to the clock mode inquiry command)  
 SUM (1 byte): Checksum  
 Error (1 byte): Error code  
 11h: Checksum error (illegal command)  
 22h: Clock mode error

**(5) Multiplication Ratio Inquiry**

In response to a multiplication ratio inquiry command sent from the host, the RX630 returns the clock types, the number of multiplication/division ratios, and the multiplication division ratios supported.

Command 

22h
-----

Response	32h	Size	Clock type count		
	Multiplication ratio count	Multiplication ratio	Multiplication ratio	...	Multiplication ratio
	Multiplication ratio count	Multiplication ratio	Multiplication ratio	...	Multiplication ratio
	SUM				

- Size (1 byte): Total number of bytes in the clock type count, multiplication ratio count, and multiplication ratio fields
- Clock type count (1 byte): Number of clock types (for example, 02h indicates two clock types; that is, a system clock and a peripheral clock)
- Multiplication ratio count (1 byte): Number of supported multiplication/division ratios (for example, 04h indicates that four multiplication ratios are supported for the system clock (multiplied by 1, multiplied by 2, multiplied by 4, and multiplied by 8))
- Multiplication ratio (1 byte): A positive value indicates a multiplication ratio (for example, 04h = 4 = multiplied by 4)  
A negative value indicates a division ratio (for example, FEh = -2 = divided by 2)
- SUM (1 byte): Checksum

**(6) Operating Frequency Inquiry**

In response to an operating frequency inquiry command sent from the host, the RX630 returns the minimum and maximum operating frequencies for each clock.

Command 

23h
-----

Response	33h	Size	Clock type count
	Minimum frequency		Maximum frequency
	Minimum frequency		Maximum frequency
	SUM		

- Size (1 byte): Total number of bytes in the clock type count, minimum frequency, and maximum frequency fields
- Clock type count (1 byte): Number of clock types (for example, 02h indicates two clock types; that is, a system clock and a peripheral clock)
- Minimum frequency (2 bytes): Minimum value of the operating frequency (for example, 07D0h indicates 20.00 MHz).  
This value should be calculated by multiplying the frequency value (MHz) to two decimal places by 100.
- Maximum frequency (2 bytes): Maximum value of the operating frequency  
This value is represented in the same format as the minimum frequency
- SUM (1 byte): Checksum

## (7) User Boot Area Information Inquiry

In response to a user boot area information inquiry command sent from the host, the RX630 returns the number of user boot areas and their addresses.

Command	24h		
Response	34h	Size	Area count
	Area start address		
	Area end address		
	SUM		

Size (1 byte): Total number of bytes in the area count, area start address, and area end address fields

Area count (1 byte): Number of user boot areas (consecutive areas are counted as one area)

Area start address (4 bytes): Start address of a user boot area

Area end address (4 bytes): End address of a user boot area

SUM (1 byte): Checksum

## (8) User Area Information Inquiry

In response to a user area information inquiry command sent from the host, the RX630 returns the number of user areas and their addresses.

Command	25h		
Response	35h	Size	Area count
	Area start address		
	Area end address		
	SUM		

Size (1 byte): Total number of bytes in the area count, area start address, and area end address fields

Area count (1 byte): Number of user areas (consecutive areas are counted as one area)

Area start address (4 bytes): Start address of a user area

Area end address (4 bytes): End address of a user area

SUM (1 byte): Checksum

## (9) Block Information Inquiry

In response to a block information inquiry command sent from the host, the RX630 returns the number of total erasure blocks in the user area and data area, and their addresses.

Command	26h		
Response	36h	Size	Block count
	Block start address		
	Block end address		
	Block start address		
	Block end address		
	...		
	Block start address		
	Block end address		
	SUM		

Size (2 bytes):	Total number of bytes in the block count, block start address, and block end address fields
Block count (1 byte):	Number of erasure blocks in the user area
Block start address (4 bytes):	Start address of an erasure block
Block end address (4 bytes):	End address of an erasure block
SUM (1 byte):	Checksum

## (10) Programming Size Inquiry

In response to a programming size inquiry command sent from the host, the RX630 returns the programming size.

Command	27h			
Response	37h	Size	Programming size	SUM

Size (1 byte):	Number of characters included in the programming size field (fixed at 2)
Programming size (2 bytes):	Programming unit (bytes)
SUM (1 byte):	Checksum

## (11) Data Area Inquiry

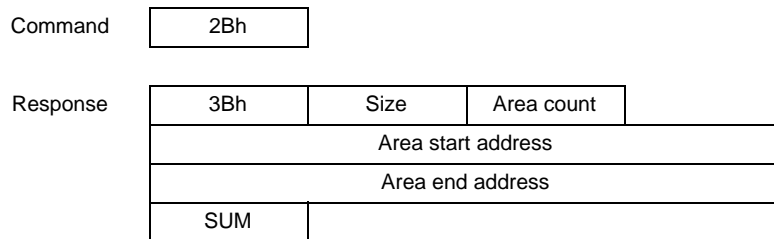
In response to a data area inquiry command sent from the host, the RX630 returns the information concerning the availability of data areas.

Command	2Ah			
Response	3Ah	Size	Area availability	SUM

Size (1 byte):	Number of characters in the area availability field (fixed at 1)
Area availability (1 byte):	Availability of data areas (fixed at 21h) 21h: Data area is available
SUM (1 byte):	Checksum

## (12) Data Area Information Inquiry

In response to a data area information inquiry command sent from the host, the RX630 returns the number of data area counts and their addresses.



Size (1 byte):	Total number of bytes in the area count, area start address, and area end address fields
Area count (1 byte):	Number of data area counts (consecutive areas are counted as one area)
Area start address (4 bytes):	Start address of a data area
Area end address (4 bytes):	End address of a data area
SUM (1 byte):	Checksum

The information concerning the block configuration in the data area is included in the response to the block information inquiry command (see section 43.8.7, Inquiry/Selection Command Wait).

## (13) New Bit Rate Selection

In response to a new bit rate selection command sent from the host, the RX630 checks if the on-chip SCI can be set to the selected new bit rate. When the SCI can be set to the new bit rate, the RX630 returns a response (06h) and sets the SCI to the new bit rate. If the SCI cannot be set to the new bit rate or the sent command is illegal, the RX630 returns an error response (BFh). Upon reception of response 06h, the host waits for a one-bit period in the previous bit rate with which the new bit rate selection command has been sent, and then sets the host's bit rate to the new one. After that, the host sends confirmation data (06h) in the new bit rate, and the RX630 returns a response (06h) for the confirmation data. Be sure to issue a new bit rate selection command only after a clock mode selection command.

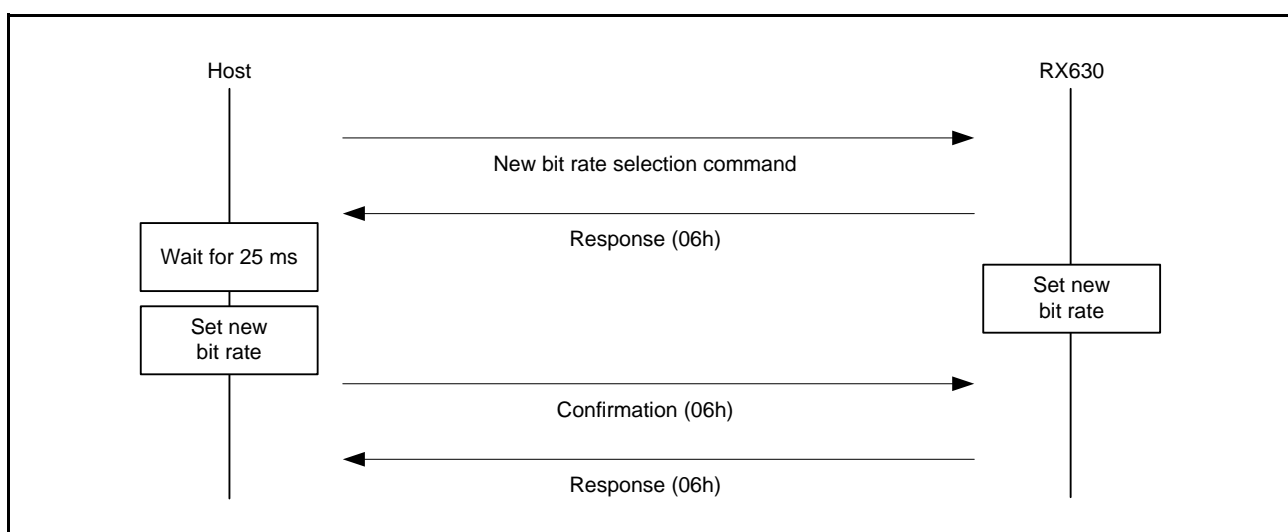


Figure 43.32 New Bit Rate Selection Sequence



Command	3Fh	Size	Bit rate	Input frequency
	Clock type count	Multiplication ratio 1	Multiplication ratio 2	
	SUM			
Response	06h			
Error response	BFh	Error		
Confirmation	06h			
Response	06h			

Size (1 byte):	Total number of bytes in the bit rate, input frequency, clock type count, and multiplication ratio fields
Bit rate (2 bytes):	New bit rate (for example, 00C0h indicates 19200 bps) 1/100 of the new bit rate value should be specified.
Input frequency (2 bytes):	Frequency input to the RX630 (for example, 04E2h indicates 12.50 MHz) This value should be calculated by multiplying the input frequency value to two decimal places by 100.
Clock type count (1 byte):	Number of clock types (fixed: 02h indicates two clock types; that is, a system clock and a peripheral clock)
Multiplication ratio 1 (1 byte):	Multiplication/division ratio of the input frequency to obtain the system clock (ICLK) A positive value indicates a multiplication ratio (for example, 04h = 4 = multiplied by 4) A negative value indicates a division ratio (for example, FEh = -2 = divided by 2)
Multiplication ratio 2 (1 byte):	Multiplication/division ratio of the input frequency to obtain the peripheral clock (PCLK) This value is represented in the same format as multiplication ratio 1
SUM (1 byte):	Checksum
Error:	Error code 11h: Checksum error 24h: Bit rate selection error 25h: Input frequency error 26h: Multiplication ratio error 27h: Operating frequency error

- Bit rate selection error

A bit rate selection error occurs when the bit rate selected through a new bit rate selection command cannot be set for the SCI of the RX630 within an error of 4%. The bit rate error can be obtained by the following equation from the bit rate (B) selected through a new bit rate selection command, the input frequency ( $f_{EX}$ ), multiplication ratio 2 ( $M_{P\phi}$ ), the bit rate register (BRR) setting (N) in the SCI, and the CKS[1:0] bit value (n) in the serial mode register (SMR).

$$\text{Error (\%)} = \left\{ \frac{f_{EX} \times M_{P\phi} \times 10^6}{B \times 64 \times 2^{2n-1} \times (N + 1)} - 1 \right\} \times 100$$

- Input frequency error

An input frequency error occurs when the input frequency specified through a new bit rate selection command is outside the range from the minimum to maximum input frequencies for the clock mode selected through a clock mode selection command.

- Multiplication ratio error

A multiplication ratio error occurs when the multiplication ratio specified through a new bit rate selection command does not match the clock mode selected through a clock mode selection command. To check the selectable multiplication ratios, issue a multiplication ratio inquiry command.

- Operating frequency error

An operating frequency error occurs when the RX630 cannot operate at the operating frequencies selected through a new bit rate selection command. The RX630 calculates the operating frequencies from the input frequency and multiplication ratios specified through a new bit rate selection command and checks if each calculated frequency is within the range from the minimum to maximum frequencies for the respective clock. To check the minimum and maximum operating frequencies for each clock, issue an operating frequency inquiry command.

#### (14) Programming/Erase State Transition

In response to a programming/erase state transition command sent from the host, the RX630 determines whether ID code protection is enabled or disabled using the control code and ID code written in the ROM. When ID code protection is enabled, the RX630 returns a response (16h) and waits for the ID code. When ID code protection is disabled, the RX630 erases the entire area of each of the user area, user boot area and data area.\*1 After completing entire erasure, the RX630 returns a response (26h) and waits for a programming/erase command. If the RX630 has failed to complete erasure due to an error, it returns an error response (C0h, and 51h).

Do not issue a programming/erase state transition command before the device selection, clock mode selection, and new bit rate selection commands.

Note 1. The user-boot area is not erased if it holds a USB boot program.

Command	40h	
Response	ACK	
Error response	C0h	51h

ACK (1 byte): ACK code  
 26h: ID code protection is disabled  
 16h: ID code protection is enabled

## (15) Boot Program Status Inquiry

In response to a boot program status inquiry command sent from the host, the RX630 returns its current status. The boot program status inquiry command can be issued in both the inquiry/selection command wait and programming/erasure command wait.

Command	4Fh			
Response	5Fh	Size	Status	Error

Size (1 byte): Total number of bytes in the status and error fields (fixed at 2)

Status (1 byte): Current status of RX630 (see Table 43.17)

Error (1 byte): Error status of RX630 (see Table 43.18)

**Table 43.17 Status Code**

Code	Description
11h	Waiting for device selection
12h	Waiting for clock mode selection
13h	Waiting for bit rate selection
1Fh	Waiting for transition to programming/erasure command wait (bit rate has been selected)
31h	Erasing the user area/user boot area
3Fh	Waiting for a programming/erasure command
4Fh	Waiting for reception of programming data
5Fh	Waiting for erasure block specification

**Table 43.18 Error Code**

Code	Description
00h	No error
11h	Checksum error
21h	Device code error
22h	Clock mode error
24h	Bit rate selection error
25h	Input frequency error
26h	Multiplication ratio error
27h	Operating frequency error
29h	Block number error
2Ah	Address error
2Bh	Data size error
51h	Erasure error
52h	Incomplete erasure error
53h	Programming error
54h	Selection error
80h	Command error
FFh	Bit rate adjustment confirmation error

### 43.8.8 ID Code Wait State

Table 43.19 shows the command available in the ID code wait state.

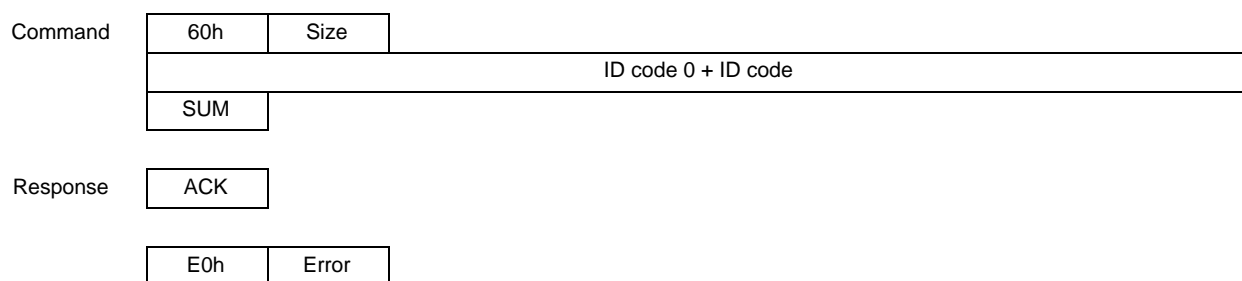
**Table 43.19 ID Code Check Command**

Command Name	Function
ID code check	Performs the ID code check

If the host has sent an undefined command, the RX630 returns a response indicating a command error. For the contents of a command error, see section 43.8.7, Inquiry/Selection Command Wait.

#### (1) ID Code Check

In response to an ID code check command sent from the host, the RX630 compares the code sent from the host with the control code and ID code in the ROM and returns the result.



Size (1 byte): Number of bytes in the ID code field (fixed at 16)

ID code (16 bytes): ID code 0 (1 byte) + ID code (15 bytes)

SUM (1 byte): Checksum

ACK (1 byte): ACK code

26h: Returns the response for a programming/erasure state transition command

Error (1 byte): Error code

11h: Checksum error

61h: ID code mismatch

63h: ID code mismatch (erasure error)

An error has occurred during erasure triggered by an ID code mismatch.

### 43.8.9 Programming/Erasure Command Wait

Table 43.20 lists the commands available in the programming/erasure command wait.

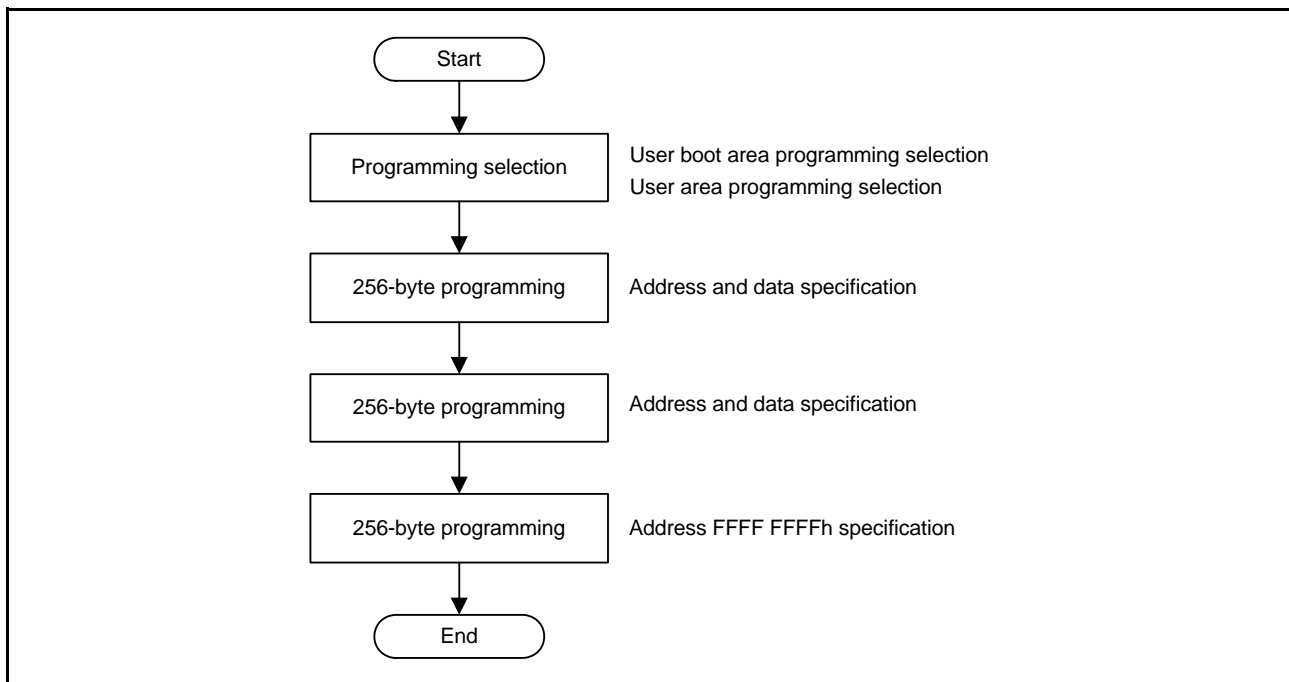
**Table 43.20 Programming/Erasure Commands**

Command Name	Function
User boot area programming selection	Selects the program for user boot area programming
User/data area programming selection	Selects the program for user/data area programming
256-byte programming	Programs 256 bytes of data
Erasure selection	Selects the erasure program
Block erase	Erases block data
Memory read	Reads data from memory
User boot area checksum	Performs checksum verification for the user boot area
User area checksum	Performs checksum verification for the user area
Data area checksum	Performs checksum verification for the data area
User boot area blank check	Checks whether the user boot area is blank
User area blank check	Checks whether the user area is blank
Data area blank check	Checks whether the data area is blank
Read lock bit status	Reads from the lock bit
Lock bit program	Writes to the lock bit
Lock bit enable	Enables the lock bit protection
Lock bit disable	Disables the lock bit protection
Boot program status inquiry	Inquires regarding the state of the RX630

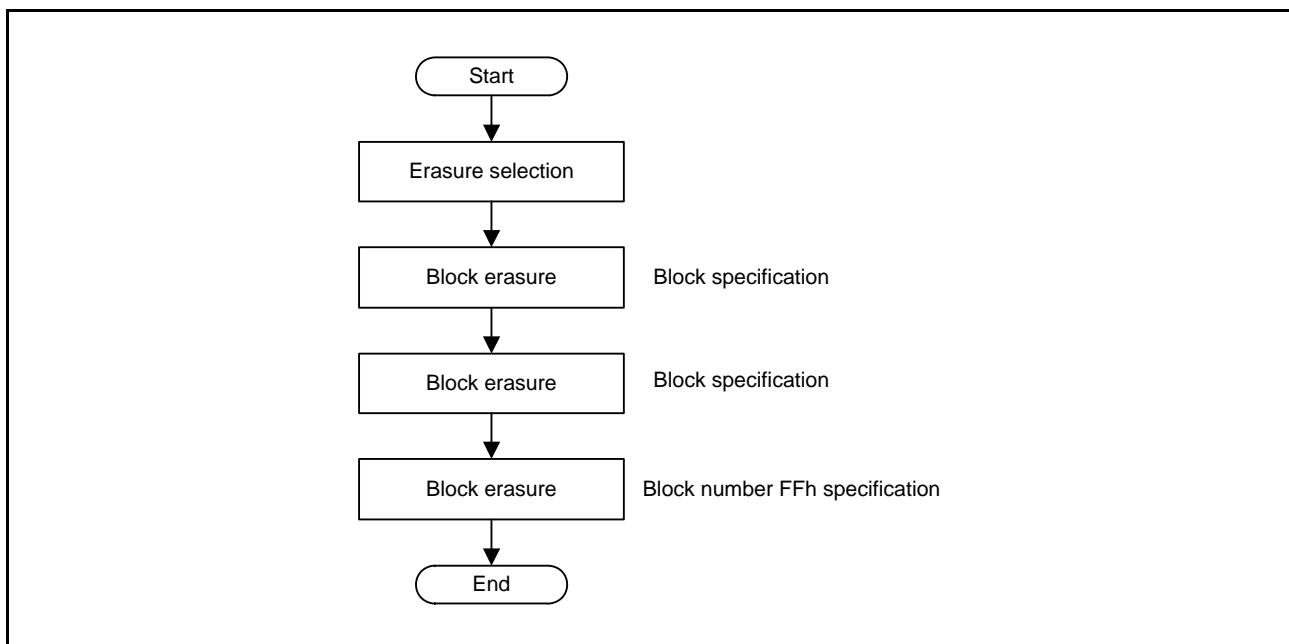
If the host has sent an undefined command, the RX630 returns a response indicating a command error. For the contents of a command error, see section 43.8.7, Inquiry/Selection Command Wait.

To program the ROM, issue a programming selection command (user area/data programming selection, user boot area programming selection) and then a 256-byte programming command from the host. To program the DataFlash, issue a user/data area programming selection command and then a 256-byte programming command specifying a data area address as the programming address. Upon reception of a programming selection command, the RX630 enters the programming data wait state (see section 43.8.2, State Transitions in Boot Mode). In response to a 256-byte programming command sent from the host in this state, the RX630 starts programming the ROM/E2 DataFlash. When the host sends a 256-byte programming command specifying FFFF FFFFh as the programming start address, the RX630 detects it as the end of programming and enters the programming/erasure command wait.

To erase the ROM/E2 DataFlash, issue an erasure selection command and then a block erase command from the host. Upon reception of an erasure selection command, the RX630 enters the erasure block selection wait state (see section 43.8.2, State Transitions in Boot Mode). In response to a block erase command sent from the host in this state, the RX630 erases the specified block in the ROM/E2 DataFlash. When the host sends a block erase command specifying FFh as the block number, the RX630 detects it as the end of erasure and enters the programming/erasure command wait. To read data from the data area, select the user area through a memory read command specifying a data area address as the read address.



**Figure 43.33 Procedure for ROM/E2 DataFlash Programming in Boot Mode**



**Figure 43.34 Procedure for ROM/E2 DataFlash Erasure in Boot Mode**

Each command is described in detail below. The “command” in the description indicates a command sent from the host to the RX630 and the “response” indicates a response sent from the RX630 to the host. The “checksum” is byte-size data calculated so that the sum of all bytes to be sent by the RX630 becomes 00h.

## (1) User Boot Area Programming Selection

In response to a user boot area programming selection command sent from the host, the RX630 selects the program for user boot area programming and waits for programming data.

Command	42h
Response	06h

## (2) User/Data Area Programming Selection

In response to a user/data area programming selection command sent from the host, the RX630 selects the program for user area programming and waits for programming data. To program the E2 DataFlash, issue a user/data area programming selection command.

Command	43h
Response	06h

## (3) 256-Byte Programming

In response to a 256-byte programming command sent from the host, the RX630 programs the ROM/E2 DataFlash. After completing ROM/E2 DataFlash programming successfully, the RX630 returns a response (06h). If an error has occurred during ROM programming, the RX630 returns an error response (D0h).

Command	50h	Programming address		
	Data	Data	...	Data
	SUM			
Response	06h			
Error response	D0h	Error		

Programming address (4 bytes): Target address of programming  
To execute programming, a 256-byte boundary address should be specified.  
To terminate programming, FFFF FFFFh should be specified.

Data (256 bytes): Programming data  
FFh should be specified for the bytes that do not need to be programmed.  
When terminating programming, no data needs to be sent (only the programming address and SUM should be sent in that order).

SUM (1 byte): Checksum

Error (1 byte): Error code  
11h: Checksum error  
2Ah: Address error (the specified address is not in the target area)  
53h: Programming cannot be done due to a programming error

## (4) Erasure Selection

In response to an erasure selection command sent from the host, the RX630 selects the erasure program and waits for erasure block specification.

Command 

48h
-----

Response 

06h
-----

## (5) Block Erasure

In response to a block erase command sent from the host, the RX630 erases the ROM/E2 DataFlash. After completing ROM/E2 DataFlash erasure successfully, the RX630 returns a response (06h). If an error has occurred during ROM/E2 DataFlash erasure, the RX630 returns an error response (D8h).

Command 

58h	Size	Block	SUM
-----	------	-------	-----

Response 

06h
-----

Error response 

D8h	Error
-----	-------

Size (1 byte): Number of bytes in the block specification field (fixed at 1)

Block (1 byte): Block number whose data is to be erased  
To terminate erasure, FFh should be specified.

SUM (1 byte): Checksum

Error (1 byte): Error code  
11h: Checksum error  
29h: Block number error (an incorrect block number is specified)  
51h: Erasure cannot be done due to an erasure error



**(6) Memory Read**

In response to a memory read command sent from the host, the RX630 reads data from the ROM/E2 DataFlash. After completing ROM/E2 DataFlash reading successfully, the RX630 returns the data stored in the address specified by the memory read command. If the RX630 has failed to read the ROM/E2 DataFlash, the RX630 returns an error response (D2h).

Command	52h	Size	Area	Read start address	
	Reading size			SUM	
Response	52h	Reading size			
	Data	Data	...	Data	
	SUM				
Error response	D2h	Error			

Size (1 byte): Total number of bytes in the area, read start address, and reading size fields

Area (1 byte): Target area to be read  
 00h: User boot area  
 01h: User area, data area

Read start address (4 bytes): Start address of the area to be read

Reading size (4 bytes): Size of data to be read (bytes)

SUM (1 byte): Checksum

Data (1 byte): Data read from the ROM/E2 DataFlash

Error (1 byte): Error code  
 11h: Checksum error  
 2Ah: Address error
 

- The value specified for area selection is neither 00h nor 01h.
- The specified read start address is outside the selected area.

 2Bh: Data size error
 

- 00h is specified for the reading size.
- The reading size is larger than the area.
- The end address calculated from the read start address and the reading size is outside the selected area.

**(7) User Boot Area Checksum**

In response to a user boot area checksum command sent from the host, the RX630 sums the user boot area data in byte units and returns the result (checksum).

Command	4Ah			
Response	5Ah	Size	Area checksum	SUM

Size (1 byte): Number of bytes in the area checksum field (fixed at 4)

Area checksum (4 bytes): Checksum of the user boot area data

SUM (1 byte): Checksum (for the response data)

## (8) User Area Checksum

In response to a user area checksum command sent from the host, the RX630 sums the user area data in byte units and returns the result (checksum).

Command 

4Bh
-----

Response 

5Bh	Size	Area checksum	SUM
-----	------	---------------	-----

Size (1 byte): Number of bytes in the area checksum field (fixed at 4)

Area checksum (4 bytes): Checksum of the user area data  
The user area also stores the key code for debugging function authentication. Note that the checksum includes this key code value.

SUM (1 byte): Checksum (for the response data)

## (9) Data Area Checksum

In response to a data area checksum command sent from the host, the RX630 sums the data area data in byte units and returns the result (checksum).

Command 

61h
-----

Response 

71h	Size	Area checksum	SUM
-----	------	---------------	-----

Size (1 byte): Number of bytes in the area checksum field (fixed at 4)

Area checksum (4 bytes): Checksum of the data area

SUM (1 byte): Checksum (for the response data)

## (10) User Boot Area Blank Check

In response to a user boot area blank check command sent from the host, the RX630 checks whether the user boot area is completely erased. When the user boot area is completely erased, the RX630 returns a response (06h). If the user boot area has an unerased area, the RX630 returns an error response (CCh, 52h).

Command 

4Ch
-----

Response 

06h
-----

Error response 

CCh	52h
-----	-----

## (11) User Area Blank Check

In response to a user area blank check command sent from the host, the RX630 checks whether the user area is completely erased. When the user area is completely erased, the RX630 returns a response (06h). If the user area has an unerased area, the RX630 returns an error response (CDh, 52h).

Command	4Dh
Response	06h
Error response	CDh 52h

## (12) Data Area Blank Check

In response to a data area blank check command sent from the host, the RX630 checks whether the data area is completely erased. When the data area is completely erased, the RX630 returns a response (06h). If the data area has an unerased area, the RX630 returns an error response (E2h, 52h).

Command	62h
Response	06h
Error response	E2h 52h

## (13) Read Lock Bit Status

In response to a read lock bit status command sent from the host, the RX630 reads data from the lock bit. After completing the lock bit reading successfully, the RX630 returns the data stored in the address specified by the read lock bit status command. If the RX630 has failed to read the lock bit, the RX630 returns an error response (F1h).

Command	71h	Size	Area	A15 to A8	A23 to A16	A31 to A24	SUM
---------	-----	------	------	-----------	------------	------------	-----

A15 to A8 (1 byte): The last address in the specified block (bits 15 to 8)  
 A23 to A16 (1 byte): The last address in the specified block (bits 23 to 16)  
 A31 to A24 (1 byte): The last address in the specified block (bits 31 to 24)

Response	Status
----------	--------

Error response	F1h	Error
----------------	-----	-------

Size (1 byte):	Total number of bytes in the area, A15 to A8, A23 to A16, and A31 to A24 (fixed at 4 in the RX630)
Area (1 byte):	Target area to be read 01h: User area
A15 to A8 (1 byte):	A15 to A8 of the last address in the specified block (bits 8 to 15)
A23 to A16 (1 byte):	A23 to A16 of the last address in the specified block (bits 16 to 23)
A31 to A24 (1 byte):	A31 to A24 of the last address in the specified block (bits 24 to 31)
SUM (1 byte):	Checksum
Status (1 byte):	Bit 6 locked at 0 Bit 6 unlocked at 1
Error (1 byte):	Error code 11h: Checksum error 2Ah: Address error (the specified address is not in the target area)

## (14) Lock Bit Program

In response to a lock bit program command sent from the host, the RX630 writes to a lock bit and locks the specified block. After completing the lock bit blocking successfully, the RX630 returns a response (06h). If the RX630 has failed to lock, the RX630 returns an error response (F7h).

Command	77h	Size	Area	Third highest order address	Second highest order address	Highest order address	SUM
Response	06h						
Error response	F7h	Error					

Size (1 byte):	Total number of bytes in the area, third highest order address, second highest order address, and highest order address fields (fixed at 4 in the RX630)
Area (1 byte):	Target area to be locked 01h: User area
Third highest order address (1 byte):	Third highest order address at the specified block's end address (8 to 15 bits)
Second highest order address (1 byte):	Second highest order address at the specified block's end address (16 to 23 bits)
Highest order address (1 byte):	Highest order address at the specified block's end address (24 to 31 bits)
SUM (1 byte):	Checksum
Error (1 byte):	Error code 11h: Checksum error 2Ah: Address error (the specified address is not in the target area) 53h: Lock cannot be done due to a programming error

## (15) Lock Bit Enable

In response to a lock bit enable command sent from the host, the RX630 enables a lock bit.

Command	7Ah
Response	06h

## (16) Lock Bit Disable

In response to a lock bit disable command sent from the host, the RX630 disables a lock bit.

Command	75h
Response	06h

## (17) Boot Program Status Inquiry

For details, refer to section 43.8.7, Inquiry/Selection Command Wait.

### 43.9 USB Boot Mode

In USB boot mode, the user area is programmed or erased by control commands and data for programming transmitted from an externally connected host via the USB.

Using USB boot mode requires preparation on the host side of tools for transmitting the control commands and data for programming, and of the data. Figure 43.35 shows the configuration of a system for use in USB boot mode. Interrupt requests generated in USB boot mode are ignored. Ensure that interrupt requests are not generated on the system side.

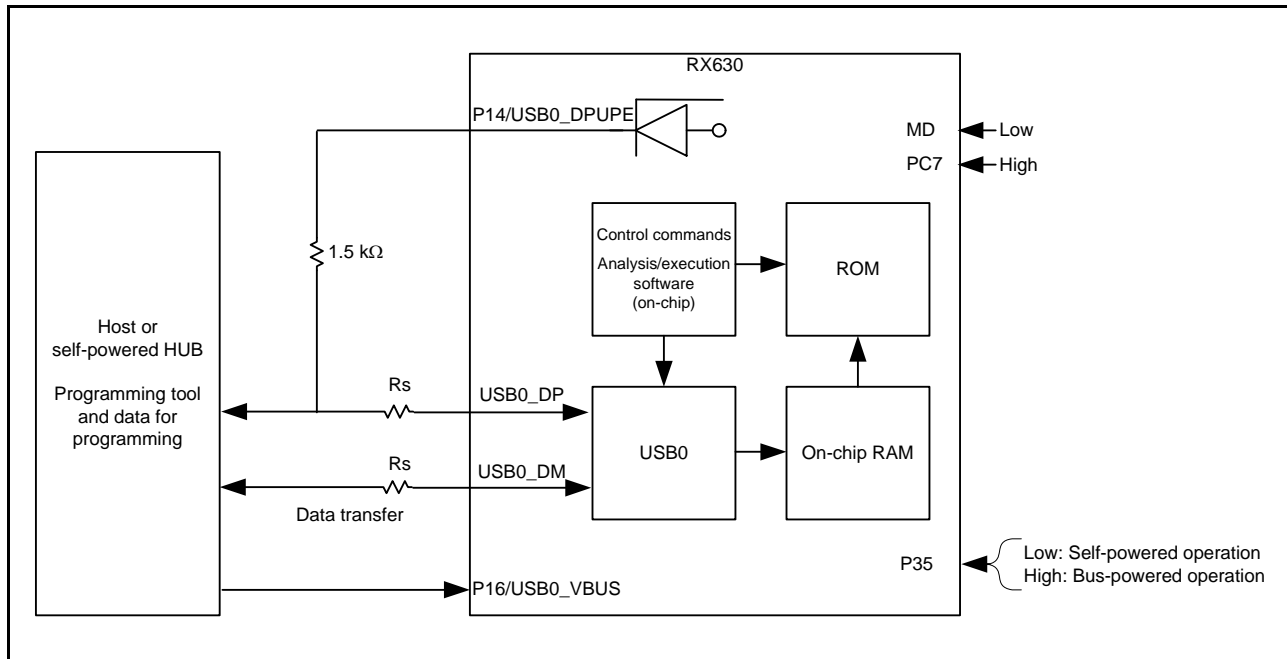


Figure 43.35 System Configuration in USB Boot Mode

#### 43.9.1 Features

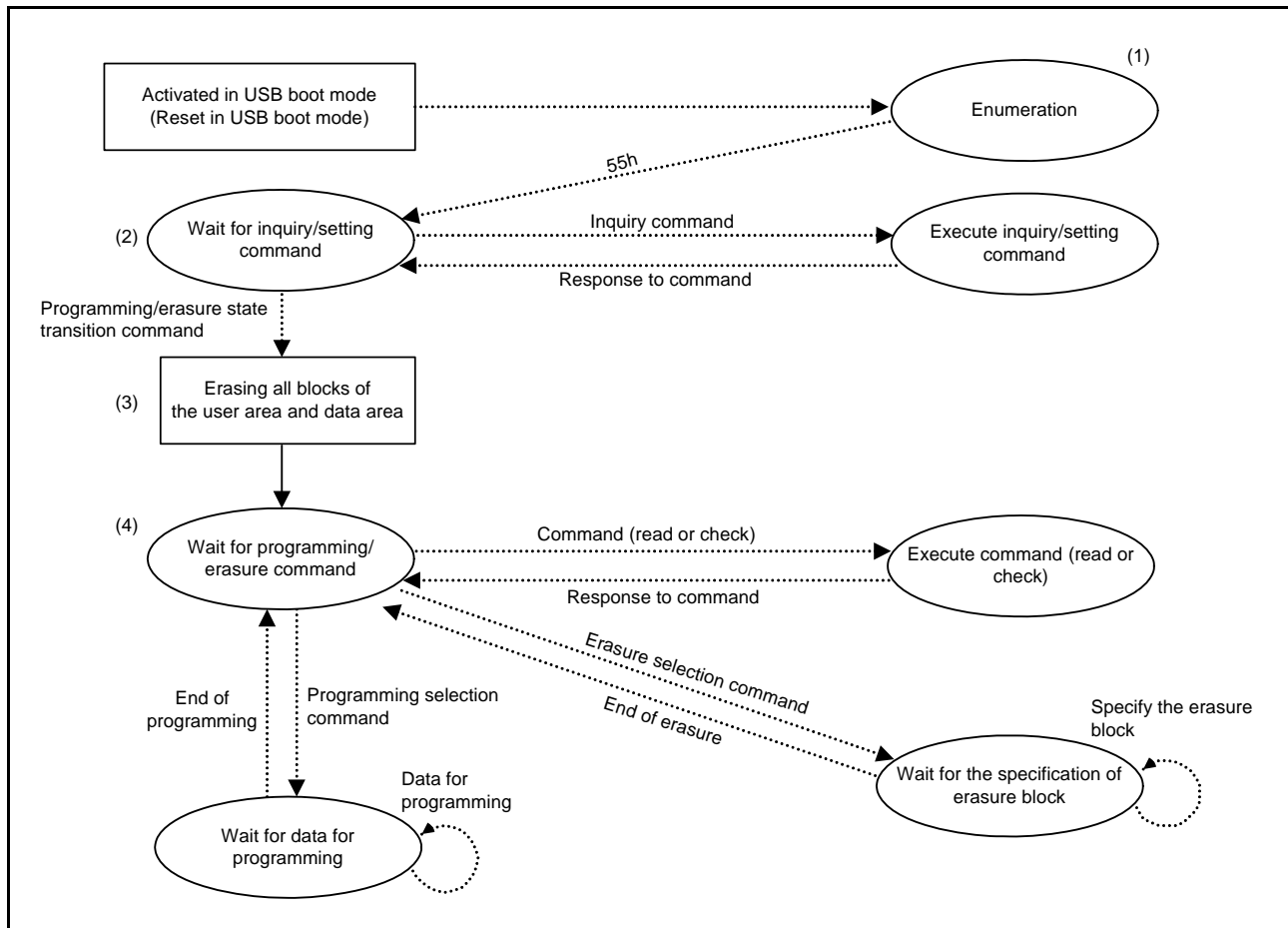
- Selection of bus-powered or self-powered mode
- Only the USB0\_DPUPE pin is used for control of D+ pull-up connection
- See Table 43.21 for the enumeration information

Table 43.21 Enumeration Information

USB specification	Ver. 2.0 (Full-speed)	
Transfer modes	Control (in/out), Bulk (in/out)	
Maximum current	Self-powered mode (pin P35 = 0)	100 mA
	Bus-powered mode (pin P35 = 1)	500 mA
Endpoint configuration	EP0 Control (in out) 8 bytes Configuration1 └─ InterfaceNumber0 └─ AlternateSetting0 └─ EP1 Bulk (out) 64 bytes └─ EP2 Bulk (in) 64 bytes	

### 43.9.2 State Transitions

State transitions after activation in USB boot mode are shown in Figure 43.36.



**Figure 43.36 State Transitions in USB Boot Mode**

- (1) The USB boot program the device as shipped contains in the user boot area runs in the case of booting up in USB boot mode. The host sends a single byte with the value 55h to the RX630 after enumeration between the host and the RX630. Reset the RX630 if it does not send E6h in return.
- (2) Inquiry data on the size, structure, first address, state of support, etc. of the user area are sent to the host.
- (3) On completion of the inquiries, the whole user area is automatically erased.
- (4) After automatic erasure of the user area, the state shifts to waiting for programming and erasure commands. When a programming selection command is received, the state shifts to waiting for data to be programmed. When an erasure-selection command is received, the state shifts to waiting for specification of blocks to be erased. Other than the programming and erasure commands, commands for the following processes on the user area are also possible: checksum, blank checking, reading memory, and acquiring current status information.

### 43.9.3 Notes on Program Execution in USB Boot Mode

- (1) In USB boot mode, a 12-MHz clock oscillator must be used and a 48-MHz clock signal must be provided to the USB module. Set the frequency of the external clock and clock oscillator so as the USB dedicated clock (UCLK) to be 48 MHz. Clock oscillators running at rates other than 12 MHz cannot be used in USB boot mode. For details, see section 9, Clock Generation Circuit.
- (2) Use the USB0\_DPUPE pin for controlling D+ pull-up connection.
- (3) To ensure that the power supply is stable during programming and erasure of flash memory, do not connect a cable via a bus-powered HUB.
- (4) Do not disconnect the USB cable during programming and erasure of flash memory. This may permanently damage the LSI circuit.
- (5) Even if the USB bus enters the suspended mode in bus-powered mode, this LSI will not enter software standby mode (low-power consumption mode).

### 43.10 ID Code Protection on Connection of the On-Chip Debugger

This function is used to prohibit connection with the on-chip debugger. When connecting an on-chip debugger, the control code and ID code that have been written to the ROM are used to determine whether ID code protection on connection of the on-chip debugger is enabled or disabled and to judge ID code protection on connection of the on-chip debugger. When the ID code protection is enabled, the code sent from the on-chip debugger is compared with the control code and ID code in the ROM to determine whether they match. If they match, connection with the on-chip debugger is allowed. If they do not match, the on-chip debugger cannot be connected. However, if the control code is 52h and the ID code is 50h, 72h, 6Fh, 74h, 65h, 63h, and 74h (from the first to the seventh field of the ID code), this is always considered to be a non-match, judgment of the ID code does not proceed, and connection of the on-chip debugger is prohibited.

Furthermore, if all bytes of the control code and ID code have the value FFh, there is no determination of matching, the ID code is always considered to match, and connection of the on-chip debugger is allowed. See Figure 43.30 for the configuration of ID codes in flash memory.

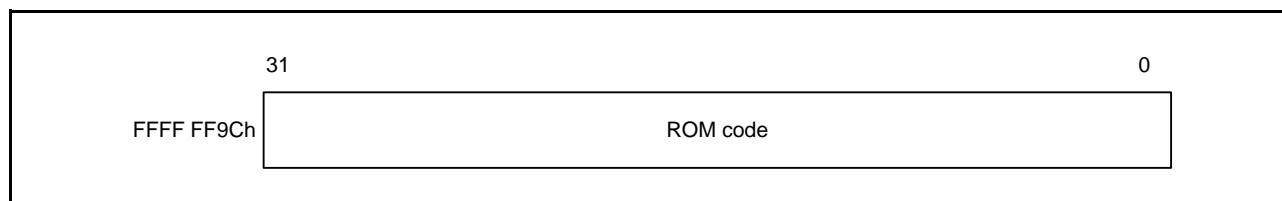
**Table 43.22 Specifications for ID Code Protection on Connection of the On-Chip Debugger**

Control Code	ID Code	State of Protection	Operations at On-Chip Debugger Connection
FFh	FFh, ..., FFh (all bytes FFh)	Protection disabled	The ID code always matches, and connection to the on-chip debugger is permitted.
52h	50h, 72h, 6Fh, 74h, 65h, 63h, 74h, FFh, ..., FFh	Protection enabled	The ID code is always non-matching, and connection to the on-chip debugger is prohibited.
Other than above	Other than above	Protection enabled	Matching ID code: Authentication of the on-chip debugger is ended and connection with the on-chip debugger is permitted. Non-matching ID code: The ID code protection waiting state is entered again.

### 43.11 ROM Code Protection

ROM code protection is a facility for prohibiting a PROM programmer from reading from or programming flash memory. The ROM code in flash memory is a 32-bit code. Figure 43.37 shows the configuration of the ROM code. Set the ROM code in 32-bit units.

For release from ROM code protection, erase the EB00 block of the user area that contains the ROM code in boot mode or by user programming.



**Figure 43.37 Configuration of ROM Code**

**Table 43.23 Specifications for ROM Code Protection**

ROM Code	State of Protection	Operations at the Time of Connection with the PROM Programmer
0000 0000h	Protection enabled (ROM code protection 1)	Access (both reading and writing) to the user area and the user boot area are prohibited.
0000 0001h	Protection enabled (ROM code protection 2)	Reading from the user area and the user boot area are prohibited.
Other than above	Protection disabled	Access (both reading and writing) to the user area and the user boot area are permitted.



## 43.12 Usage Notes (Common to the ROM/E2 DataFlash Memory)

### (1) Areas where Programming or Erasure is Suspended

Data in areas where programming or erasure is suspended are undefined. To avoid malfunctions due to the reading of undefined data, prevent the reading of data and execution of code from areas where programming or erasure is currently suspended.

### (2) Suspending Programming or Erasure

If you use the programming/erasure suspension command to suspend the processing of programming or erasure, be sure to use the resume command so that the processing is completed.

### (3) Prohibition of Reprogramming

Two or more programming operations cannot be performed for the same address range. If an address range that has already been programmed is to be programmed again, be sure to erase the area in advance of the programming.

### (4) Reset during Programming, Erasure, or Blank Check

In case of the generation of a reset by the signal on the RES# pin during programming and erasure or blank checking, only de-assert the reset signal after it has remained within the range of operating voltage stipulated in the electrical characteristics over the period tRESWF (see section 45, Electrical Characteristics).

In case of resetting of the FCU by the FRESETR.FRESET bit during programming and erasure or blank checking, maintain the reset state over the period tFCUR (see section 45, Electrical Characteristics).

During resetting of the FCU, do not read ROM that is the target of programming and erasure or blank checking.

Resets by the watchdog timer and watchdog timer and software resets during programming and erasure or blank checking can be used without securing the above periods.

### (5) Prohibition of Non-Maskable Interrupts during Programming or Erasure

If a non-maskable interrupt (NMI pin interrupt, oscillation stop detection interrupt, WDT underflow, refresh error, IWDT underflow/refresh error, voltage-monitoring 1 interrupt or voltage-monitoring 2 interrupt) occurs during programming or erasure, as this will lead to fetching of the vector from the ROM, and the data read out will be undefined. Therefore, avoid a non-maskable interrupt being generated during programming or erasing in the ROM (this prohibition only applies to the ROM).

### (6) Interrupt Vector Assignment During Programming, Erasure, or Blank Check

The generation of interrupts during programming, erasure, or blank check may lead to the fetching of vectors from the ROM. To prevent access to the ROM area due to the generation of interrupts, set the interrupt table register (INTB) of the CPU so that the destination for the fetching of interrupt vector is an area outside the ROM.

### (7) Programming and Erasure in Low-Speed Operating Modes 1 and 2

Do not program or erase the flash memory after low-speed operating mode 1 or 2 has been selected in the operating voltage control register (OPCCR).

### (8) Abnormal End during Programming, Erasure or Blank Check

In cases where programming and erasure or blank checking is not completed due to the operating voltage rising above the range stipulated in the electrical characteristics, a reset, the FCU being reset by using the FRESETR.FRESET bit, error detection leading to the flash memory being placed in the command-locked state, or one of the prohibitions under item (9), the lock bit may become 0 (indicating the protected state).

In such cases, erase the lock bit by issuing a block-erase command while the FPROTR.FPROTCN is 1. Redo programming that was not completed normally after that.

### (9) Actions Prohibited during Programming, Erasure, or Blank Check

During programming, erasure, or blank check, high voltage is applied to the flash memory. In order to prevent damaging the flash memory, the following must be observed.

- Do not allow the input voltage to be below the operating voltage of the RX630.
- Do not change the value of the FWEPROR.FLWE[1:0] bits.
- Do not change the operating mode by the setting of the SYSCR0.ROME bit.
- Do not change the value of the OPCCR.OPCM[2:0] bits.
- Do not change the clock-source by setting of the SCKCR3.CKSEL[2:0] bits.
- Do not enable the switching of the clock source for use on recovery from sleep mode by the setting of the RSTCKCR.RSTCKEN bit.
- Do not change the frequency dividing ratio for the FlashIF clock (FCLK).
- Do not make transition to all-module clock-stop mode, software standby mode, or deep software standby mode.

### (10) Notes on Flash Programming in Boot Mode or USB Boot Mode

The main clock must be being input for programming of the flash memory in boot mode or USB boot mode. In the case of boot mode, a crystal oscillator within the range indicated in section 45.3.2, Clock Timing, must be connected between the XTAL and EXTAL pins. Connect a 12-MHz oscillator if operation is in USB boot mode.

### (11) Handling of the EXTAL Pin in Boot Mode

When operation is in boot mode, provide a clock signal through connection of an external input or crystal oscillator to the EXTAL pin.

### (12) Handling of the EXTAL Pin in USB Boot Mode

When operation is in USB boot mode, provide a 12-MHz clock signal through connection of a crystal oscillator to the EXTAL pin.

## 43.13 Usage Notes (for E2 DataFlash)

### (1) Protection of Data Area Immediately after a Reset

As the initial values of DFLRE<sub>y</sub> and DFLWE<sub>y</sub> ( $y = 0, 1$ ) are 0000h, programming, erasure, and reading of the data area are disabled after a reset. To read data from the data area, set DFLRE<sub>y</sub> appropriately before accessing the data area. To program or erase the data area, set DFLWE<sub>y</sub> appropriately before issuing an FCU command for programming or erasure. If an attempt is made to read, program, or erase the data area without setting the registers, the FCU detects the error and the FASTAT.CMDLK bit becomes 1 (the command-locked state).

### (2) Reading in Low-Speed Operating Mode 2

The E2 DataFlash memory is not readable in low-speed operating mode 2. Place the chip in another mode if reading is to proceed.

## 44. Boundary Scan

The RX630 Group has boundary scan function.

The boundary scan is a serial I/O interface based on the JTAG (Joint Test Action Group, IEEE Std.1149.1 and IEEE Standard Test Access Port and Boundary-Scan Architecture).

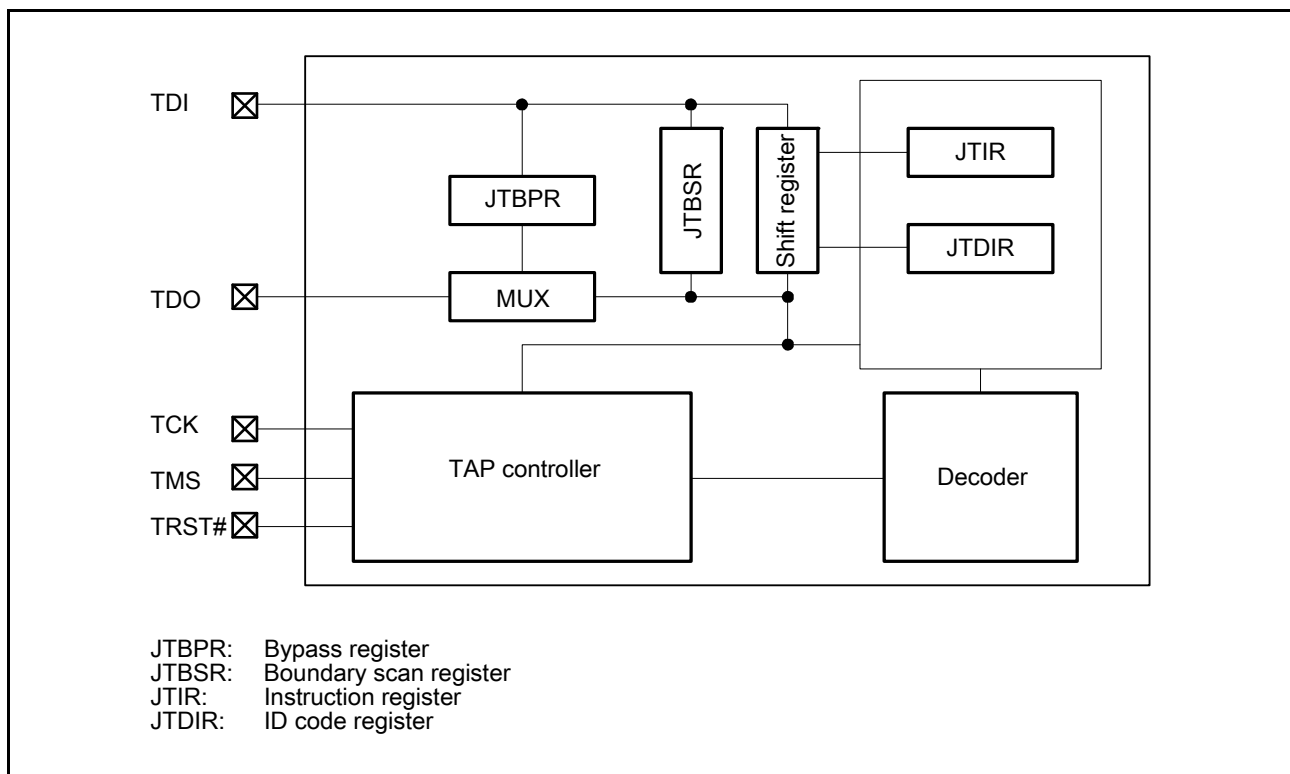
### 44.1 Features

Table 44.1 lists the specifications of boundary scan.

Figure 44.1 shows a block diagram of the boundary scan function.

**Table 44.1 Specifications of Boundary Scan**

Item	Description
Boundary scan enabled/disabled	Boundary scan is enabled when the EMLE pin is driven low and the BSCANP pin is driven high.
Dedicated boundary scan pins	The following pins are dedicated for the JTAG, when boundary scan function is enabled (TDO/TCK/TDI/TMS/TRST#). 177-pin TFLGA/176-pin LFBGA: PF0/PF1/PF2/PF3/PF4 145-pin TFLGA: P26/P27/P30/P31/P34
Six test modes	<ul style="list-style-type: none"> <li>• BYPASS mode</li> <li>• EXTEST mode</li> <li>• SAMPLE/PRELOAD mode</li> <li>• CLAMP mode</li> <li>• HIGHZ mode</li> <li>• IDCODE mode</li> </ul>



**Figure 44.1 Block Diagram of JTAG**

Table 44.2 shows the I/O pins used in the boundary scan function.

**Table 44.2 Pin Configuration**

Pin Name	I/O	Description
TCK	Input	Test clock input pin Clock signal for boundary scan. Input the clock the duty cycle of which is 50 percent when boundary scan function is used.
TMS	Input	Test mode select pin
TDI	Input	Test data input pin
TDO	Output	Test data output pin
TRST#	Input	Test reset input pin

## 44.2 Register Descriptions

Table 44.3 lists the boundary scan registers.

**Table 44.3 List of Boundary Scan Registers**

Register Name	Symbol	Value after Reset	Address	Access Size
Instruction register	JTIR	55h	—	—
ID code register	JTIDR	080B B447h	—	—
Bypass register	JTBPR	Undefined	—	—
Boundary scan register	JTBSR	Undefined	—	—

Instructions can be input to the JTIR register via the TDI pin by serial transfer.

The JTBPR register, which is a 1-bit register, is connected between the TDI and TDO pins in BYPASS mode.

The JTBSR register, which is configured according to Table 44.6, is connected between the TDI and TDO pins when test data are being shifted in.

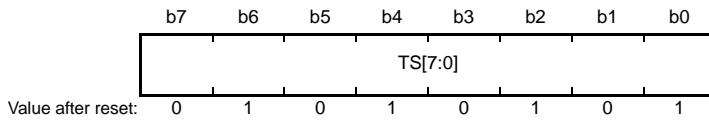
None of the registers is accessible from the CPU.

Table 44.4 shows the availability of serial transfer for the registers.

**Table 44.4 Serial Transfer for the Registers**

Register Name	Serial Input	Serial Output
Instruction register (JTIR)	Available	Not available
ID code register (JTIDR)	Not available	Available
Bypass register (JTBPR)	Available	Available
Boundary scan register (JTBSR)	Available	Available

### 44.2.1 Instruction Register (JTIR)



Bit	Symbol	Bit Name	Description	R/W
b7 to b0	TS[7:0]	Test Bit Set	The command configuration is as shown in Table 44.5.	—

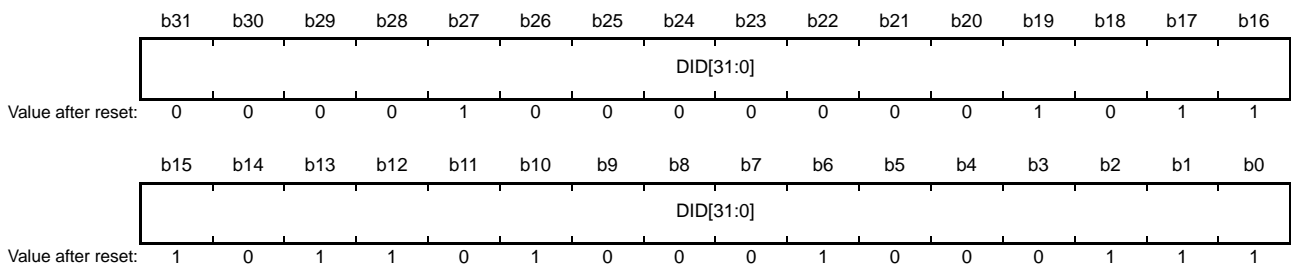
**Table 44.5 Command Configuration**

TS7	TS6	TS5	TS4	TS3	TS2	TS1	TS0	Instruction
0	0	0	0	0	0	0	0	EXTEST
0	1	0	0	0	0	0	0	SAMPLE/PRELOAD
0	1	0	1	0	1	0	1	IDCODE (initial value)
1	1	0	1	0	0	0	0	CLAMP
1	0	0	0	0	0	0	0	HIGHZ
1	1	1	1	1	1	1	1	BYPASS
Other than above								Reserved

JTAG instructions can be transferred to the JTIR register by serial input from the TDI pin.

The JTIR register is initialized when the TRST# pin is driven low, or when the TAP controller is in the Test-Logic-Reset state.

### 44.2.2 ID Code Register (JTIDR)



Bit	Symbol	Bit Name	Description	R/W
b31 to b0	DID[31:0]	Reserved	JTIDR is a register with the fixed value that indicates the device IDCODE.	—

JTIDR data is output from the TDO pin when the IDCODE instruction has been executed.

### 44.2.3 Bypass Register (JTBPR)

The JTBPR register is a 1-bit register and is connected between the TDI and TDO pins when the JTIR register is set to BYPASS mode.

The JTBPR register cannot be read from or written to by the CPU.

### 44.2.4 Boundary Scan Register (JTBSR)

The JTBSR register is a shift register to control the external input and output pins of this LSI and is distributed across the pads.

The EXTEST, SAMPLE/PRELOAD, CLAMP, and HIGHZ instructions are issued to apply the JTBSR register in boundary-scan testing.

Table 44.6 and Table 44.7 show the correspondence between the JTBSR bits and the pins of this LSI.

The value after reset is undefined.

**Table 44.6** Boundary Scan Register  
177-Pin TFLGA/176-Pin LFBGA (1/10)

From TDI			
Pin No.	Pin Name	Input/Output	Bit Name
B1	P05	Output	417
		Output enable	416
		Input	415
D3	P03	Output	414
		Output enable	413
		Input	412
D2	P02	Output	411
		Output enable	410
		Input	409
D1	P01	Output	408
		Output enable	407
		Input	406
D4	P00	Output	405
		Output enable	404
		Input	403
E3	PF5	Output	402
		Output enable	401
		Input	400
E1	PJ5	Output	399
		Output enable	398
		Input	397
F3	PJ3	Output	396
		Output enable	395
		Input	394
H4	P35	Input	393
K1	P33	Output	392
		Output enable	391
		Input	390
K2	P32	Output	389
		Output enable	388
		Input	387
L4	P25	Output	386
		Output enable	385
		Input	384
N1	PH5	Output	383
		Output enable	382
		Input	381
M3	P24	Output	380
		Output enable	379
		Input	378
P1	PH4	Output	377
		Output enable	376
		Input	375

**Table 44.6** Boundary Scan Register  
177-Pin TFLGA/176-Pin LFBGA (2/10)

Pin No.	Pin Name	Input/Output	Bit Name
N2	P23	Output	374
		Output enable	373
		Input	372
N3	P22	Output	371
		Output enable	370
		Input	369
R1	P21	Output	368
		Output enable	367
		Input	366
R2	P20	Output	359
		Output enable	358
		Input	357
P2	P17	Output	356
		Output enable	355
		Input	354
P3	P87	Output	353
		Output enable	352
		Input	351
R3	P16	Output	350
		Output enable	349
		Input	348
M4	P86	Output	347
		Output enable	346
		Input	345
N4	P15	Output	344
		Output enable	343
		Input	342
P4	P14	Output	341
		Output enable	340
		Input	339
R4	P85	Output	338
		Output enable	337
		Input	336
M5	P13	Output	335
		Output enable	334
		Input	333
N5	P12	Output	332
		Output enable	331
		Input	330
R5	P11	Output	329
		Output enable	328
		Input	327

**Table 44.6** Boundary Scan Register  
177-Pin TFLGA/176-Pin LFBGA (3/10)

Pin No.	Pin Name	Input/Output	Bit Name
P5	P10	Output	326
		Output enable	325
		Input	324
N6	P57	Output	323
		Output enable	322
		Input	321
M6	P56	Output	320
		Output enable	319
		Input	318
R8	PL4	Output	317
		Output enable	316
		Input	315
P8	PL3	Output	314
		Output enable	313
		Input	312
N8	PL2	Output	311
		Output enable	310
		Input	309
M6	P55	Output	308
		Output enable	307
		Input	306
M7	P54	Output	305
		Output enable	304
		Input	303
M8	P53	Output	302
		Output enable	301
		Input	300
R9	P84	Output	299
		Output enable	298
		Input	297
P9	P52	Output	296
		Output enable	295
		Input	294
N9	P51	Output	293
		Output enable	292
		Input	291
M9	P50	Output	290
		Output enable	289
		Input	288
P10	P83	Output	287
		Output enable	286
		Input	285
N10	PC7	Output	284
		Output enable	283
		Input	282

**Table 44.6** Boundary Scan Register  
177-Pin TFLGA/176-Pin LFBGA (4/10)

Pin No.	Pin Name	Input/Output	Bit Name
P11	PC6	Output	281
		Output enable	280
		Input	279
M10	PC5	Output	278
		Output enable	277
		Input	276
N11	P82	Output	275
		Output enable	274
		Input	273
M11	P81	Output	272
		Output enable	271
		Input	270
R12	P80	Output	269
		Output enable	268
		Input	267
P12	PC4	Output	266
		Output enable	265
		Input	264
N12	PC3	Output	263
		Output enable	262
		Input	261
M12	P77	Output	260
		Output enable	259
		Input	258
R13	P76	Output	257
		Output enable	256
		Input	255
P13	PC2	Output	254
		Output enable	253
		Input	252
P14	P75	Output	251
		Output enable	250
		Input	249
R14	P74	Output	248
		Output enable	247
		Input	246
R15	PC1	Output	245
		Output enable	244
		Input	243
P15	PL1	Output	242
		Output enable	241
		Input	240
N13	PC0	Output	239
		Output enable	238
		Input	237



**Table 44.6** Boundary Scan Register  
177-Pin TFLGA/176-Pin LFBGA (5/10)

Pin No.	Pin Name	Input/Output	Bit Name
N15	PL0	Output	236
		Output enable	235
		Input	234
N14	P73	Output	233
		Output enable	232
		Input	231
M13	PB7	Output	230
		Output enable	229
		Input	228
L12	PB6	Output	227
		Output enable	226
		Input	225
M14	PB5	Output	224
		Output enable	223
		Input	222
M15	PB4	Output	221
		Output enable	220
		Input	219
L13	PB3	Output	218
		Output enable	217
		Input	216
K12	PB2	Output	215
		Output enable	214
		Input	213
L14	PB1	Output	212
		Output enable	211
		Input	210
L15	P72	Output	209
		Output enable	208
		Input	207
K13	P71	Output	206
		Output enable	205
		Input	204
K14	PK7	Output	203
		Output enable	202
		Input	201
K15	PB0	Output	200
		Output enable	199
		Input	198
J13	PK6	Output	197
		Output enable	196
		Input	195
J14	PA7	Output	194
		Output enable	193
		Input	192

**Table 44.6** Boundary Scan Register  
177-Pin TFLGA/176-Pin LFBGA (6/10)

Pin No.	Pin Name	Input/Output	Bit Name
J15	PA6	Output	191
		Output enable	190
		Input	189
J12	PA5	Output	188
		Output enable	187
		Input	186
H12	PA4	Output	185
		Output enable	184
		Input	183
H13	PA3	Output	182
		Output enable	181
		Input	180
H15	PG7	Output	179
		Output enable	178
		Input	177
H14	PA2	Output	176
		Output enable	175
		Input	174
G13	PG6	Output	173
		Output enable	172
		Input	171
G14	PA1	Output	170
		Output enable	169
		Input	168
G12	PG5	Output	167
		Output enable	166
		Input	165
F14	PA0	Output	164
		Output enable	163
		Input	162
F13	PG4	Output	161
		Output enable	160
		Input	159
E15	P67	Output	158
		Output enable	157
		Input	156
E14	PG3	Output	155
		Output enable	154
		Input	153
F12	P66	Output	152
		Output enable	151
		Input	150
E13	PG2	Output	149
		Output enable	148
		Input	147

**Table 44.6** Boundary Scan Register  
177-Pin TFLGA/176-Pin LFBGA (7/10)

Pin No.	Pin Name	Input/Output	Bit Name
D15	P65	Output	146
		Output enable	145
		Input	144
D14	PE7	Output	143
		Output enable	142
		Input	141
E12	PE6	Output	140
		Output enable	139
		Input	138
D13	PK5	Output	137
		Output enable	136
		Input	135
C15	P70	Output	134
		Output enable	133
		Input	132
C14	PK4	Output	131
		Output enable	130
		Input	129
D12	PE5	Output	128
		Output enable	127
		Input	126
C13	PE4	Output	125
		Output enable	124
		Input	123
B15	PE3	Output	122
		Output enable	121
		Input	120
A15	PE2	Output	119
		Output enable	118
		Input	117
A14	PE1	Output	116
		Output enable	115
		Input	114
B14	PE0	Output	113
		Output enable	112
		Input	111
B13	P64	Output	110
		Output enable	109
		Input	108
A13	P63	Output	107
		Output enable	106
		Input	105
C12	P62	Output	104
		Output enable	103
		Input	102

**Table 44.6** Boundary Scan Register  
177-Pin TFLGA/176-Pin LFBGA (8/10)

Pin No.	Pin Name	Input/Output	Bit Name
D11	P61	Output	101
		Output enable	100
		Input	99
B12	PK3	Output	98
		Output enable	97
		Input	96
A12	P60	Output	95
		Output enable	94
		Input	93
C11	PK2	Output	92
		Output enable	91
		Input	90
D10	PD7	Output	89
		Output enable	88
		Input	87
B11	PG1	Output	86
		Output enable	85
		Input	84
A11	PD6	Output	83
		Output enable	82
		Input	81
C10	PG0	Output	80
		Output enable	79
		Input	78
D9	PD5	Output	77
		Output enable	76
		Input	75
B10	PD4	Output	74
		Output enable	73
		Input	72
A10	P97	Output	71
		Output enable	70
		Input	69
C9	PD3	Output	68
		Output enable	67
		Input	66
D8	PK1	Output	65
		Output enable	64
		Input	63
B9	P96	Output	62
		Output enable	61
		Input	60
A9	PK0	Output	59
		Output enable	58
		Input	57

**Table 44.6** Boundary Scan Register  
177-Pin TFLGA/176-Pin LFBGA (9/10)

Pin No.	Pin Name	Input/Output	Bit Name
C8	PD2	Output	56
		Output enable	55
		Input	54
D7	P95	Output	53
		Output enable	52
		Input	51
B8	PD1	Output	50
		Output enable	49
		Input	48
A8	P94	Output	47
		Output enable	46
		Input	45
C7	PD0	Output	44
		Output enable	43
		Input	42
D6	P93	Output	41
		Output enable	40
		Input	39
B7	P92	Output	38
		Output enable	37
		Input	36
B6	P91	Output	35
		Output enable	34
		Input	33
C6	P90	Output	32
		Output enable	31
		Input	30
B5	P47	Output	29
		Output enable	28
		Input	27
A5	P46	Output	26
		Output enable	25
		Input	24
C5	P45	Output	23
		Output enable	22
		Input	21
D5	P44	Output	20
		Output enable	19
		Input	18
C4	P43	Output	17
		Output enable	16
		Input	15
A4	P42	Output	14
		Output enable	13
		Input	12

**Table 44.6** Boundary Scan Register  
177-Pin TFLGA/176-Pin LFBGA (10/10)

Pin No.	Pin Name	Input/Output	Bit Name
B4	P41	Output	11
		Output enable	10
		Input	9
B3	P40	Output	8
		Output enable	7
		Input	6
B2	P07	Output	5
		Output enable	4
		Input	3
To TDO			

**Table 44.7 Boundary Scan Register  
(145-Pin TFLGA) (1/8)**

From TDI			
Pin No.	Pin Name	Input/Output	Bit Name
B3	P05	Output	417
		Output enable	416
		Input	415
D3	P03	Output	414
		Output enable	413
		Input	412
C2	P02	Output	411
		Output enable	410
		Input	409
D4	P01	Output	408
		Output enable	407
		Input	406
D1	P00	Output	405
		Output enable	404
		Input	403
D2	PF5	Output	402
		Output enable	401
		Input	400
E3	PJ5	Output	399
		Output enable	398
		Input	397
F3	PJ3	Output	396
		Output enable	395
		Input	394
H4	P35	Output	393
J2	P33	Output	392
		Output enable	391
		Input	390
J3	P32	Output	389
		Output enable	388
		Input	387
L1	P25	Output	386
		Output enable	385
		Input	384
L4	P24	Output	380
		Output enable	379
		Input	378
L2	P23	Output	374
		Output enable	373
		Input	372
M1	P22	Output	371
		Output enable	370
		Input	369

**Table 44.7 Boundary Scan Register  
(145-Pin TFLGA) (2/8)**

Pin No.	Pin Name	Input/Output	Bit Name
N1	P21	Output	368
		Output enable	367
		Input	366
N2	P20	Output	359
		Output enable	358
		Input	357
M2	P17	Output	356
		Output enable	355
		Input	354
N3	P87	Output	353
		Output enable	352
		Input	351
L3	P16	Output	350
		Output enable	349
		Input	348
M3	P86	Output	347
		Output enable	346
		Input	345
K4	P15	Output	344
		Output enable	343
		Input	342
N4	P14	Output	341
		Output enable	340
		Input	339
L5	P13	Output	335
		Output enable	334
		Input	333
M4	P12	Output	332
		Output enable	331
		Input	330
L6	P56	Output	320
		Output enable	319
		Input	318
N7	P55	Output	308
		Output enable	307
		Input	306
K5	P54	Output	305
		Output enable	304
		Input	303
K6	P53	Output	302
		Output enable	301
		Input	300

**Table 44.7 Boundary Scan Register  
(145-Pin TFLGA) (3/8)**

Pin No.	Pin Name	Input/Output	Bit Name
L7	P52	Output	296
		Output enable	295
		Input	294
K7	P51	Output	293
		Output enable	292
		Input	291
M7	P50	Output	290
		Output enable	289
		Input	288
L8	P83	Output	287
		Output enable	286
		Input	285
N9	PC7	Output	284
		Output enable	283
		Input	282
M8	PC6	Output	281
		Output enable	280
		Input	279
L9	PC5	Output	278
		Output enable	277
		Input	276
N10	P82	Output	275
		Output enable	274
		Input	273
M9	P81	Output	272
		Output enable	271
		Input	270
K9	P80	Output	269
		Output enable	268
		Input	267
L10	PC4	Output	266
		Output enable	265
		Input	264
N11	PC3	Output	263
		Output enable	262
		Input	261
M10	P77	Output	260
		Output enable	259
		Input	258
K10	P76	Output	257
		Output enable	256
		Input	255
L11	PC2	Output	254
		Output enable	253
		Input	252

**Table 44.7 Boundary Scan Register  
(145-Pin TFLGA) (4/8)**

Pin No.	Pin Name	Input/Output	Bit Name
N12	P75	Output	251
		Output enable	250
		Input	249
N13	P74	Output	248
		Output enable	247
		Input	246
M12	PC1	Output	245
		Output enable	244
		Input	243
M13	PL1	Output	242
		Output enable	241
		Input	240
M11	PC0	Output	239
		Output enable	238
		Input	237
L13	PL0	Output	236
		Output enable	235
		Input	234
L12	P73	Output	233
		Output enable	232
		Input	231
K11	PB7	Output	230
		Output enable	229
		Input	228
K12	PB6	Output	227
		Output enable	226
		Input	225
K13	PB5	Output	224
		Output enable	223
		Input	222
J11	PB4	Output	221
		Output enable	220
		Input	219
J10	PB3	Output	218
		Output enable	217
		Input	216
J12	PB2	Output	215
		Output enable	214
		Input	213
J13	PB1	Output	212
		Output enable	211
		Input	210
H10	P72	Output	209
		Output enable	208
		Input	207

**Table 44.7 Boundary Scan Register  
(145-Pin TFLGA) (5/8)**

Pin No.	Pin Name	Input/Output	Bit Name
H11	P71	Output	206
		Output enable	205
		Input	204
H12	PB0	Output	200
		Output enable	199
		Input	198
H13	PA7	Output	194
		Output enable	193
		Input	192
G11	PA6	Output	191
		Output enable	190
		Input	189
G10	PA5	Output	188
		Output enable	187
		Input	186
G13	PA4	Output	185
		Output enable	184
		Input	183
F10	PA3	Output	182
		Output enable	181
		Input	180
F13	PA2	Output	176
		Output enable	175
		Input	174
F12	PA1	Output	170
		Output enable	169
		Input	168
E10	PA0	Output	164
		Output enable	163
		Input	162
E13	P67	Output	158
		Output enable	157
		Input	156
E11	P66	Output	152
		Output enable	151
		Input	150
E12	P65	Output	146
		Output enable	145
		Input	144
D10	PE7	Output	143
		Output enable	142
		Input	141
D13	PE6	Output	140
		Output enable	139
		Input	138

**Table 44.7 Boundary Scan Register  
(145-Pin TFLGA) (6/8)**

Pin No.	Pin Name	Input/Output	Bit Name
D11	PK5	Output	137
		Output enable	136
		Input	135
C12	P70	Output	134
		Output enable	133
		Input	132
C13	PK4	Output	131
		Output enable	130
		Input	129
D12	PE5	Output	128
		Output enable	127
		Input	126
B13	PE4	Output	125
		Output enable	124
		Input	123
A13	PE3	Output	122
		Output enable	121
		Input	120
B12	PE2	Output	119
		Output enable	118
		Input	117
A12	PE1	Output	116
		Output enable	115
		Input	114
C11	PE0	Output	113
		Output enable	112
		Input	111
D9	P64	Output	110
		Output enable	109
		Input	108
C10	P63	Output	107
		Output enable	106
		Input	105
A11	P62	Output	104
		Output enable	103
		Input	102
B11	P61	Output	101
		Output enable	100
		Input	99
A10	PK3	Output	98
		Output enable	97
		Input	96
D8	P60	Output	95
		Output enable	94
		Input	93

**Table 44.7 Boundary Scan Register  
(145-Pin TFLGA) (7/8)**

Pin No.	Pin Name	Input/Output	Bit Name
B10	PK2	Output	92
		Output enable	91
		Input	90
C9	PD7	Output	89
		Output enable	88
		Input	87
A9	PD6	Output	83
		Output enable	82
		Input	81
D7	PD5	Output	77
		Output enable	76
		Input	75
B9	PD4	Output	74
		Output enable	73
		Input	72
C8	PD3	Output	68
		Output enable	67
		Input	66
A8	PD2	Output	56
		Output enable	55
		Input	54
C7	PD1	Output	50
		Output enable	49
		Input	48
B8	PD0	Output	44
		Output enable	43
		Input	42
D6	P93	Output	41
		Output enable	40
		Input	39
A7	P92	Output	38
		Output enable	37
		Input	36
B7	P91	Output	35
		Output enable	34
		Input	33
A6	P90	Output	32
		Output enable	31
		Input	30
B6	P47	Output	29
		Output enable	28
		Input	27
C5	P46	Output	26
		Output enable	25
		Input	24

**Table 44.7 Boundary Scan Register  
(145-Pin TFLGA) (8/8)**

Pin No.	Pin Name	Input/Output	Bit Name
A5	P45	Output	23
		Output enable	22
		Input	21
E5	P44	Output	20
		Output enable	19
		Input	18
B5	P43	Output	17
		Output enable	16
		Input	15
A4	P42	Output	14
		Output enable	13
		Input	12
C4	P41	Output	11
		Output enable	10
		Input	9
A3	P40	Output	8
		Output enable	7
		Input	6
A2	P07	Output	5
		Output enable	4
		Input	3
		To TDO	

### 44.3 Operations

The boundary scan functionality is valid when the RES# pin is driven high, the EMLE pin is driven low, and the BSCANP pin is driven high.

#### 44.3.1 TAP Controller

Figure 44.2 shows the state transition diagram of the TAP controller.

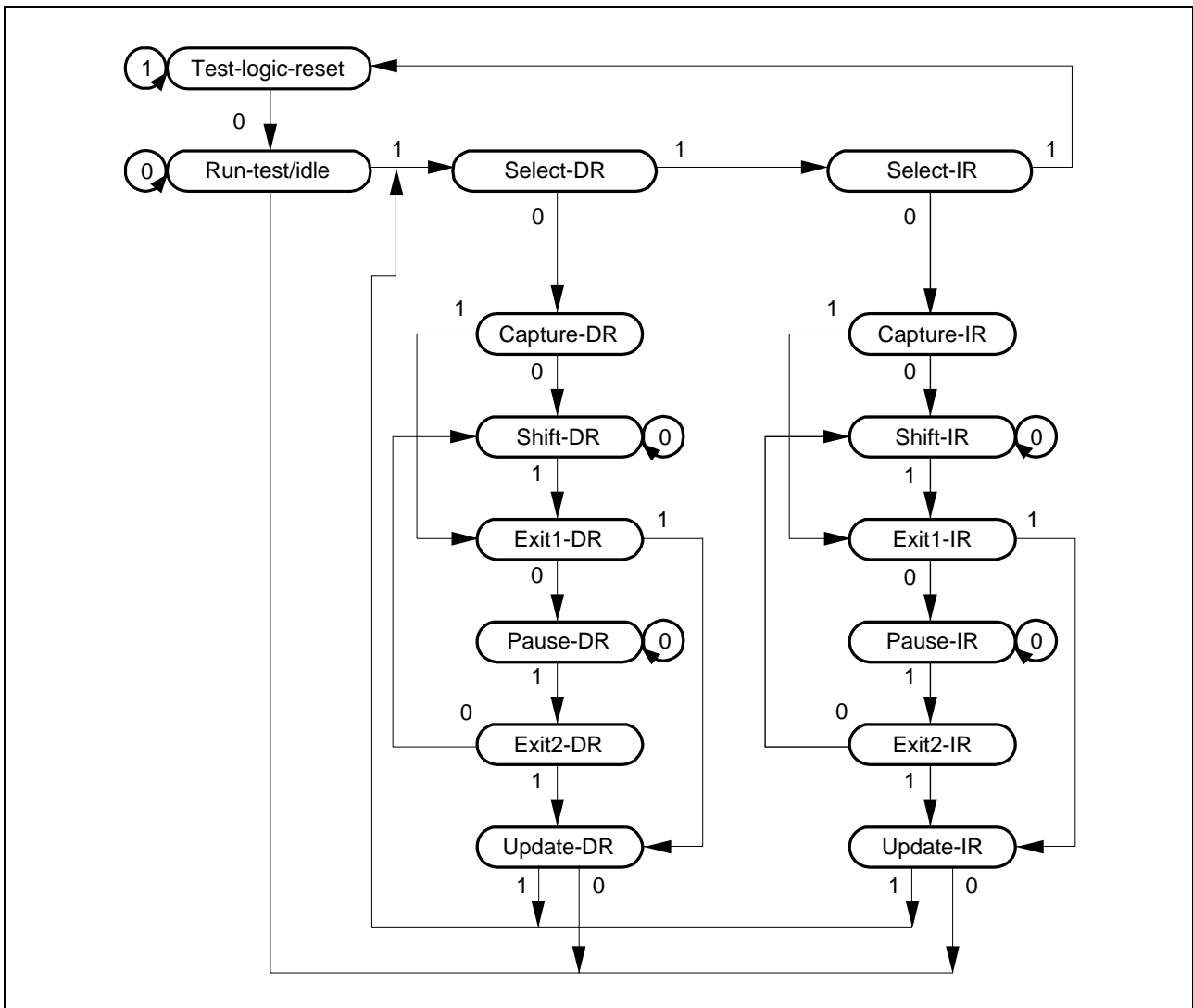


Figure 44.2 State Transition of TAP Controller



### 44.3.2 List of Commands

#### (1) BYPASS [Instruction Code: 1111 1111b]

The BYPASS instruction is an instruction that drives the bypass register (JTBPR). This instruction shortens the shift path, facilitating the transfer of serial data to other LSIs on a printed-circuit board at higher speeds. While this instruction is being executed, the test circuit has no effect on the system circuits.

The bypass register (JTBPR) is connected between the TDI and TDO pins. Bypass operation is initiated from shift-DR operation. The TDO is low in the first clock cycle in the shift-DR state; in the subsequent clock cycles, the TDI signal is output on the TDO pin.

#### (2) EXTEST [Instruction Code: 0000 0000b]

The EXTEST instruction is used to test external circuits when this LSI is installed on the printed circuit board. If this instruction is executed, output pins are used to output test data (specified by the SAMPLE/PRELOAD instruction) from the boundary scan register to the print circuit board, and input pins are used to input test result.

#### (3) SAMPLE/PRELOAD [Instruction Code: 0100 0000b]

The SAMPLE/PRELOAD instruction is used to input data from the LSI internal circuits to the boundary scan register, output data from scan path, and reload the data to the scan path. While this instruction is executed, input signals are directly input to the LSI and output signals are also directly output to the external circuits. The LSI system circuit is not affected by this instruction.

In SAMPLE operation, the boundary scan register latches the snap shot of data transferred from input pins to internal circuit or data transferred from internal circuit to output pins. The latched data is read from the scan path. The scan register latches the snap data at the rising edge of the TCK in Capture-DR state. The scan register latches snap shot without affecting the LSI normal operation.

In PRELOAD operation, initial value is written from the scan path to the parallel output latch of the boundary scan register prior to the EXTEST instruction execution. If the EXTEST is executed without executing this PRELOAD operation, undefined values are output from the beginning to the end (transfer to the output latch) of the EXTEST sequence. (In EXTEST instruction, output parallel latches are always output to the output pins.)

#### (4) IDCODE [Instruction Code: 0101 0101b]

When the IDCODE instruction is selected, IDCODE register value is output to the TDO in Shift-DR state of the TAP controller. In this case, IDCODE register value is output from the LSB. During this instruction execution, test circuit does not affect the system circuit. INSTR is initialized by the IDCODE instruction in Test-Logic-Reset state of the TAP controller.

#### (5) CLAMP [Instruction Code: 1101 0000b]

When the CLAMP instruction is selected, output pins output the boundary scan register value which was specified by the SAMPLE/PRELOAD instruction in advance. While the CLAMP instruction is selected, the status of boundary scan register is maintained regardless of the TAP controller state.

BYPASS is connected between TDI and TDO, the same operation as BYPASS instruction can be achieved.

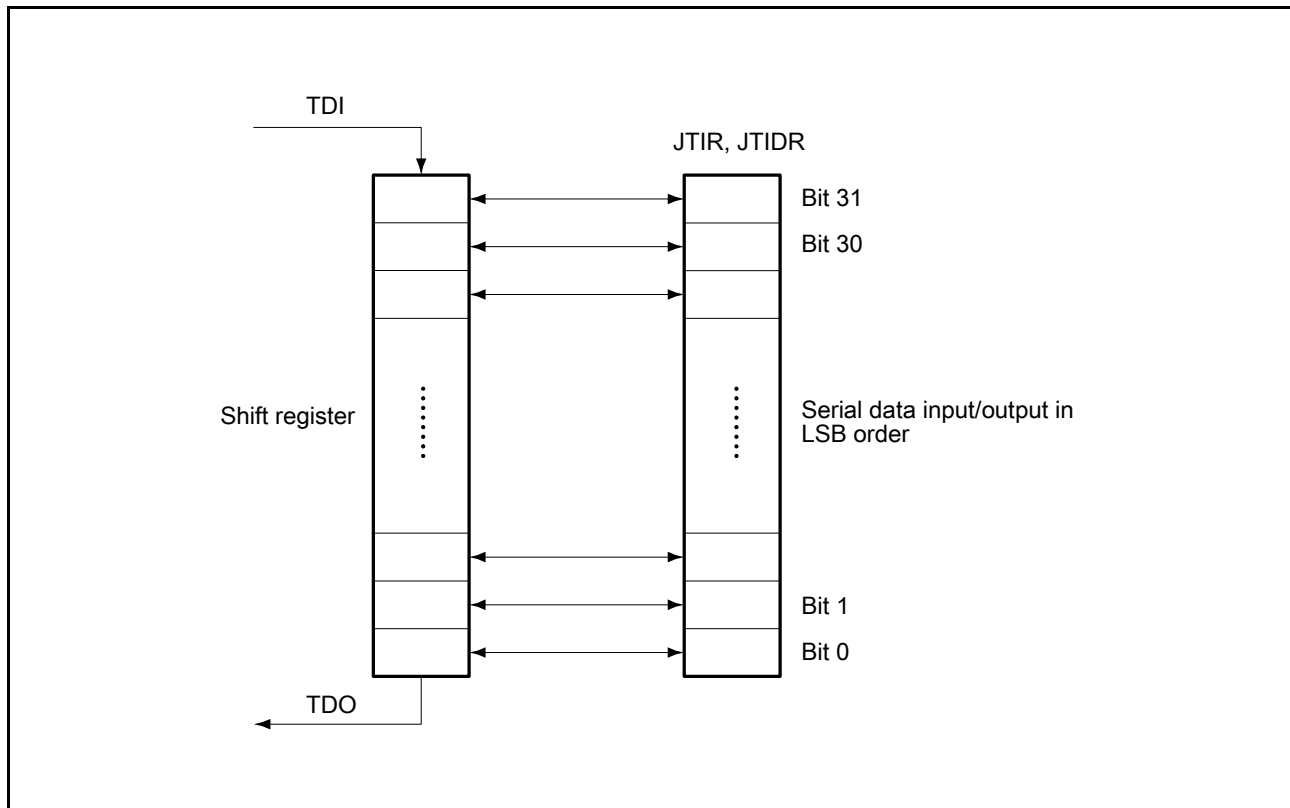
#### (6) HIGHZ [Instruction Code: 1000 0000b]

When the HIGHZ instruction is selected, all output pins enter high-impedance state. While the HIGHZ instruction is selected, the status of boundary scan register is maintained regardless of the state of the TAP controller.

BYPASS is connected between TDI and TDO pins, leading to the same operation as when the BYPASS instruction has been selected.

## 44.4 Usage Notes

- (1) Pin serial transfer, data are input or output in LSB order (see Figure 44.3).



**Figure 44.3** Serial Data Input/Output

- (2) Pins of the boundary scan (TCK, TDI, TMS, and TRST#) have to be pulled up by pull-up resistors. However, handle the #TRST pin in the way described in the manual for the given on-chip emulator if an on-chip emulator is in use. If the #TRST pin is pulled down but a boundary scan is to proceed, ensure that the #TRST pin is also controllable.
- (3) Power supply pins (VCC, VCL, VSS, AVCC0, AVSS0, VREFH0, VREFL0, VREFH, VREFL, VCC\_USB, and VSS\_USB) cannot be boundary-scanned.
- (4) Clock pins (EXTAL, XTAL, XCIN, and XCOUT) cannot be boundary-scanned.
- (5) Reset signal (RES#) cannot be boundary-scanned.
- (6) USB dedicated pins (USB0\_DP, USB1\_DP, USB0\_DM and USB1\_DM) cannot be boundary-scanned.
- (7) The on-chip emulator enable pin (EMLE) cannot be boundary-scanned.
- (8) The boundary-scan pin (BSCANP) cannot be boundary-scanned.
- (9) The boundary-scan pins (TCK, TMS, TRST#, TDI, and TDO) cannot be boundary-scanned.
- (10) The boundary-scan facility is not available when the chip is in the states below.
- Reset state
  - Software standby or deep software standby
- (11) For a pin that incorporates open-drain functionality and for which the open-drain function is enabled, if the boundary-scan function sets the corresponding bit in the output scan register and output enable register to 1, executing an EXTTEST, CLAMP, or SAMPLE/PRELOAD instruction makes the pin output the high level rather than placing it in the high-impedance state.

- (12) Figure 44.4 (1) shows the pin configuration of pins P12, P13, P20, and P21. When the boundary scan function is used with pins P12, P13, P16, P17, P20, P21, PC0, and PC1 to be used as RIIC pins (SCL0[FM+], SCL1 to SCL3, SDA0[FM+], and SDA1 to SDA3), the conflict with open-drain output or sneak current might be generated.
- (13) Figure 44.4 (2) shows the pin configuration of pins P00 to P02, P40 to P47, P90 to P93, PD0 to PD7, and PE2 to PE7. When the boundary scan function is used with pins P00 to P02, P40 to P47, P90 to P93, PD0 to PD7, and PE2 to PE7 to be used as AD input pins (AN000 to AN020, and AN0 to AN7), the conflict with the AD input or sneak current might be generated.
- (14) Figure 44.4 (3) shows the pin configuration of pins P03 and P05. When the boundary scan function is used with pins P03 and P05 to be used as DA output pins (DA0 and DA1), the conflict with the DA output or sneak current might be generated.

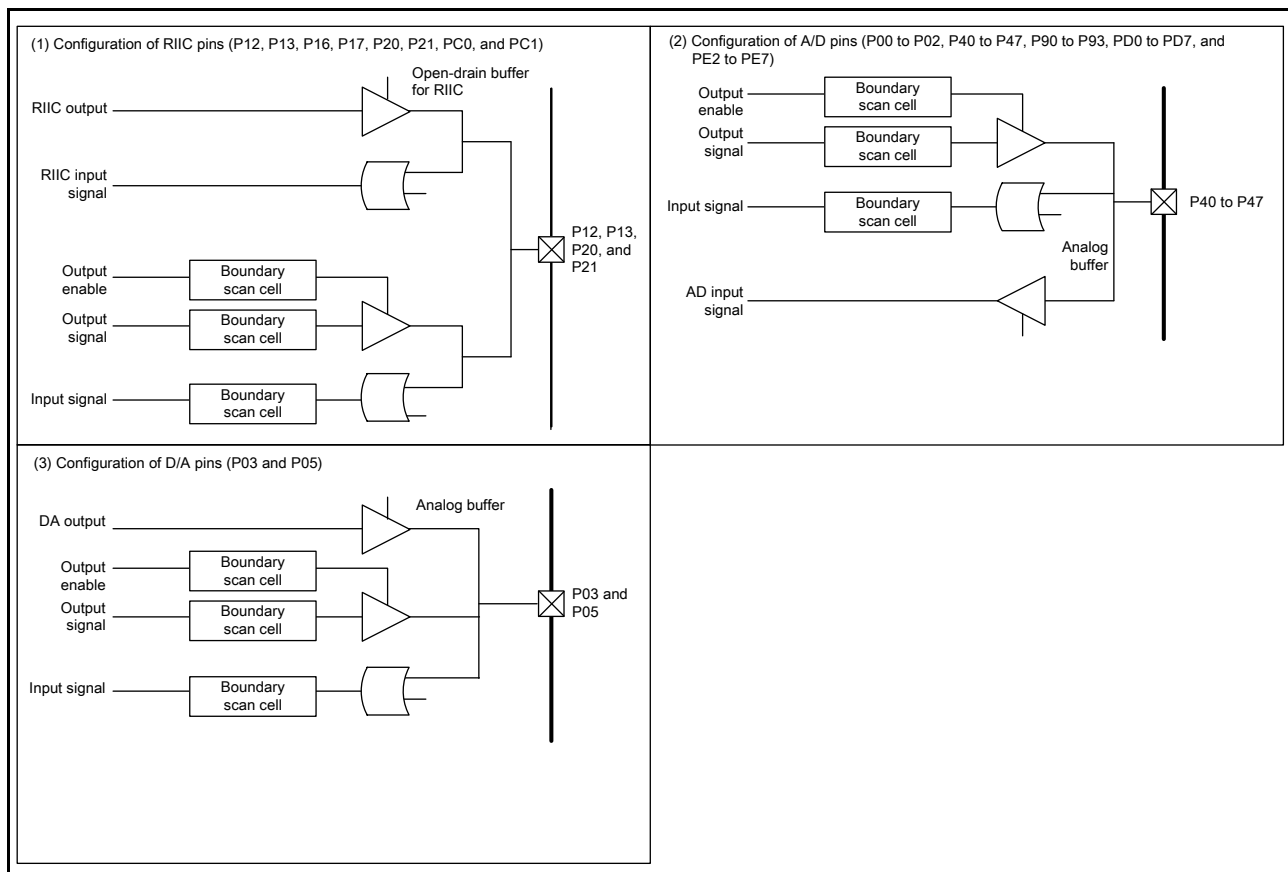


Figure 44.4 Pin Configuration

## 45. Electrical Characteristics

### 45.1 Absolute Maximum Ratings

**Table 45.1 Absolute Maximum Ratings**

Conditions: VSS = AVSS0 = VREFL/VREFL0 = VSS\_USB = 0 V

Item	Symbol	Value	Unit
Power supply voltage	VCC, VCC_USB	-0.3 to +4.6	V
V <sub>BATT</sub> power supply voltage	V <sub>BATT</sub>	-0.3 to +4.6	V
Input voltage (except for ports for 5 V tolerant*1)	V <sub>in</sub>	-0.3 to VCC +0.3	V
Input voltage (ports for 5 V tolerant*1)	V <sub>in</sub>	-0.3 to +5.8	V
Reference power supply voltage	VREFH	-0.3 to VCC +0.3	V
Analog power supply voltage	AVCC*2	-0.3 to +4.6	V
Analog input voltage	V <sub>AN</sub>	-0.3 to VCC +0.3	V
Operating temperature	T <sub>opr</sub>	-40 to +85	°C
Storage temperature	T <sub>stg</sub>	-55 to +125	°C

Caution: Permanent damage to the LSI may result if absolute maximum ratings are exceeded.

Note 1. Ports 07, 12 to 17, 20 to 25, 30 to 34, 50 to 52, 54 to 57, 67, 74 to 77, 80 to 82, A1 to A4, A6, B, and C are 5 V tolerant.

Note 2. Connect AVCC0 to VCC. When neither the A/D converter nor the D/A converter is in use, do not leave the AVCC0, VREFH/VREFH0, AVSS0, and VREFL/VREFL0 pins open. Connect the AVCC0 and VREFH/VREFH0 pins to VCC, and the AVSS0 and VREFL/VREFL0 pins to VSS, respectively.

## 45.2 DC Characteristics

**Table 45.2 DC Characteristics (1)**

Conditions:  $V_{CC} = AV_{CC0} = V_{REFH} = V_{CC\_USB} = 2.7$  to  $3.6$  V,  $V_{REFH0} = 2.7$  V to  $AV_{CC0}$ ,  
 $V_{SS} = AV_{SS0} = V_{REFL}/V_{REFL0} = V_{SS\_USB} = 0$  V,  $T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Schmitt trigger input voltage	IRQ input pin*1	$V_{IH}$	$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$	V
	MTU input pin*1	$V_{IL}$	-0.3	—	$V_{CC} \times 0.2$	
	TMR input pin*1	$\Delta V_T$	$V_{CC} \times 0.06$	—	—	
	SCI input pin*1					
	ADTRG# input pin*1					
	RES#, NMI					
	RIIC input pin (except for SMBus)	$V_{IH}$	$V_{CC} \times 0.7$	—	5.8	
		$V_{IL}$	-0.3	—	$V_{CC} \times 0.3$	
		$\Delta V_T$	$V_{CC} \times 0.05$	—	—	
	Ports for 5 V tolerant*2	$V_{IH}$	$V_{CC} \times 0.8$	—	5.8	
		$V_{IL}$	-0.3	—	$V_{CC} \times 0.2$	
	Other input pins excluding ports for 5 V tolerant*3	$V_{IH}$	$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$	
$V_{IL}$		-0.3	—	$V_{CC} \times 0.2$		
Input high voltage (except for Schmitt trigger input pin)	MD pin, EMLE	$V_{IH}$	$V_{CC} \times 0.9$	—	$V_{CC} + 0.3$	V
	EXTAL, RSPI, WAIT#, TCK		$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$	
	XCIN*3		$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$	
	D0 to D31		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$	
	RIIC (SMBus)		2.1	—	$V_{CC} + 0.3$	
Input low voltage (except for Schmitt trigger input pin)	MD pin, EMLE	$V_{IL}$	-0.3	—	$V_{CC} \times 0.1$	V
	EXTAL, RSPI, WAIT#, TCK		-0.3	—	$V_{CC} \times 0.2$	
	XCIN*3		-0.3	—	$V_{CC} \times 0.2$	
	D0 to D31		-0.3	—	$V_{CC} \times 0.3$	
	RIIC (SMBus)		-0.3	—	0.8	

Note 1. This does not include the pins, which are multiplexed as ports for 5 V tolerant.

Note 2. Ports 07, 12 to 17, 20 to 25, 30 to 34, 50 to 52, 54 to 57, 67, 74 to 77, 80 to 82, A1 to A4, A6, B, and C are 5 V tolerant.

Note 3. For P32, P31, P30, and XCIN, input as follows when the  $V_{BATT}$  power supply is selected.

$V_{IH}$  Min. =  $V_{BATT} \times 0.8$ ,  $V_{IH}$  Max. =  $V_{BATT} + 0.3$ ,  $V_{IL}$  Min. = -0.3,  $V_{IL}$  Max. =  $V_{BATT} \times 0.2$

**Table 45.3 DC Characteristics (2)**

Conditions:  $V_{CC} = AVCC0 = VREFH = VCC\_USB = 2.7$  to  $3.6$  V,  $VREFH0 = 2.7$  V to  $AVCC0$ ,  
 $VSS = AVSS0 = VREFL/VREFL0 = VSS\_USB = 0$  V,  $T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Output high voltage	All output pins	$V_{OH}$	$V_{CC} - 0.5$	—	—	V $I_{OH} = -1$ mA
Output low voltage	All output pins (except for RIIC pins)	$V_{OL}$	—	—	0.5	V $I_{OL} = 1.0$ mA
	RIIC pins		—	—	0.4	
	RIIC pins (only P12 and P13 in channel 0)	$V_{OL}$	—	—	0.4	V $I_{OL} = 15.0$ mA (ICFER.FMPE = 1)
			—	0.4	—	
Input leakage current	RES#, MD pin, EMLE*1, NMI	$ I_{in} $	—	—	1.0	$\mu$ A $V_{in} = 0$ V $V_{in} = V_{CC}$
Three-state leakage current (off state)	Other than ports for 5 V tolerant	$ I_{TSI} $	—	—	1.0	$\mu$ A $V_{in} = 0$ V $V_{in} = V_{CC}$
	Ports for 5 V tolerant		—	—	5.0	
Input pull-up MOS current	Ports 0 to 2, 30 to 34, 36, 37, 4 to G, H4, H5, J3, J5, K, L	$I_p$	-300	—	-10	$\mu$ A $V_{CC} = 2.7$ to $3.6$ V $V_{in} = 0$ V
Input capacitance	All input pins (except for ports 12, 13, 16, 17, 20, 21, 4, C0, C1, and EMLE)	$C_{in}$	—	—	15	pF $V_{in} = 0$ V $f = 1$ MHz $T_a = 25^\circ$ C
	Ports 12, 13, 16, 17, 20, 21, 4, C0, C1, EMLE		—	—	30	
Input pull-down MOS current	EMLE BSCANP*2	$I_p$	10	—	300	$\mu$ A $V_{in} = V_{CC}$

Note 1. The input leakage current value at the EMLE pin is only when  $V_{in} = 0$  V.

Note 2. The BSCANP pin is present in 177-, 176-, and 145-pin versions.

**Table 45.4 DC Characteristics (3)**

Conditions: VCC = AVCC0 = VREFH = VCC\_USB = V<sub>BATT</sub> = 2.7 to 3.6 V, VREFH0 = 2.7 V to AVCC0,  
VSS = AVSS0 = VREFL/VREFL0 = VSS\_USB = 0 V, T<sub>a</sub> = T<sub>opr</sub>

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions		
Supply current*1	High-speed operating mode	Max.*2	I <sub>CC</sub> *3	—	—	100	mA	ICLK = 100 MHz PCLKB = 50 MHz FCLK = 50 MHz BCLK = 50 MHz	
		Normal		Peripheral function: clock signal supplied*4	—	52			—
				Peripheral function: clock signal stopped*4	—	40			—
		Sleep mode		—	25	60			
		All-module-clock-stop mode (reference value)		—	20	30			
		Increased by BGO operation*5		—	15	—			
	Low-speed operating mode 1*6		—	4	—	ICLK = 1 MHz			
	Low-speed operating mode 2		—	1	—	ICLK = 32.768 kHz			
	Software standby mode		—	0.2	6				
	Deep software standby mode	Power supplied to RAM and USB resume detecting unit	—	22	200	μA			
			Power not supplied to RAM and USB resume detecting unit	—	21			60	
		Power-on reset circuit and low-power consumption function disabled		—	6.2			28	
		Power-on reset circuit and low-power		—	3			—	
		Increased by RTC operation		—	1.7			—	V <sub>BATT</sub> = 2.3 V
RTC operation when VCC is off		—	3.3	—	V <sub>BATT</sub> = 3.3 V				
Analog power supply current*7	During 12-bit A/D conversion (including temperature sensor)		A <sub>I</sub> CC	—	2.3	3.2	mA		
	During 10-bit A/D conversion			—	1.0	1.65			
	During D/A conversion (per unit)			—	0.7	1.0			
	Waiting for A/D, D/A conversion (all units)			—	25	35			μA
	A/D, D/A converter in standby mode (all units)			—	0.1	4.0			μA
Reference power supply current	During 12-bit A/D conversion		I <sub>REFH</sub>	—	0.6	0.7	mA		
	Waiting for 12-bit A/D conversion (per unit)			—	0.5	0.6			
	12-bit A/D converter in standby mode (per unit)			—	0.1	2.0			μA
RAM standby voltage		V <sub>RAM</sub>	2.7	—	—	V			
VCC rising gradient		SrVCC	8.4	—	20000	μs/V			
VCC falling gradient*8		SfVCC	8.4	—	—	μs/V			

Note 1. Supply current values are with all output pins unloaded and all input pull-up MOSs in the off state.

Note 2. Measured with clocks supplied to the peripheral functions. This does not include the BGO operation.

Note 3. I<sub>CC</sub> depends on f (ICLK) as follows. (ICLK:PCLK:BCLK:BCLK pin = 8:4:4:2)

I<sub>CC</sub> Max. = 0.87 × f + 13 (max. operation in high-speed operating mode)

I<sub>CC</sub> Typ. = 0.35 × f + 5 (normal operation in high-speed operating mode)

I<sub>CC</sub> Typ. = 1.0 × f + 3 (low-speed operating mode 1)

I<sub>CC</sub> Max. = 0.48 × f + 12 (sleep mode)

Note 4. This does not include the BGO operation.

Note 5. This is the increase for programming or erasure of the ROM or flash memory for data storage during program execution.

Note 6. Supply of the clock signal to peripherals is stopped in this state. This does not include the BGO operation.

Note 7. The reference power supply current is included in the power supply current value for 10-bit A/D conversion and D/A conversion.

Note 8. When VBATT is used

**Table 45.5 Permissible Output Currents**

Conditions:  $V_{CC} = AVCC0 = VREFH = VCC\_USB = 2.7$  to  $3.6$  V,  $VREFH0 = 2.7$  V to  $AVCC0$ ,  
 $VSS = AVSS0 = VREFL/VREFLO = VSS\_USB = 0$  V,  $T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	
Permissible output low current (average value per pin)	All output pins*1	Normal drive	$I_{OL}$	—	—	2.0	mA
	All output pins*2	High drive	$I_{OL}$			3.8	mA
Permissible output low current (max. value per pin)	All output pins*1	Normal drive	$I_{OL}$	—	—	4.0	mA
	All output pins*2	High drive	$I_{OL}$			7.6	mA
Permissible output low current (total)	Total of all output pins		$\Sigma I_{OL}$	—	—	80	mA
Permissible output high current (average value per pin)	All output pins (except for USB_DPUPE pin)*1	Normal drive	$-I_{OH}$	—	—	-2.0	mA
	USB_DPUPE pin*2	High drive	$-I_{OH}$	—	—	-3.8	mA
Permissible output high current (max. value per pin)	All output pins*1	Normal drive	$-I_{OH}$	—	—	-4.0	mA
	All output pins*2	High drive	$-I_{OH}$	—	—	-7.6	mA
Permissible output high current (total)	Total of all output pins		$\Sigma -I_{OH}$	—	—	-80	mA

Caution: To protect the LSI's reliability, the output current values should not exceed the values in this table.

Note 1. This is the value when normal driving ability is set with a pin for which normal driving ability is selectable.

Note 2. This is the value when high driving ability is set with a pin for which normal driving ability is selectable or the value of the pin to which high driving ability is fixed.



## 45.3 AC Characteristics

**Table 45.6 Operation Frequency Value (High-Speed Operating Mode)**

Conditions:  $V_{CC} = AVCC0 = VREFH = VCC\_USB = V_{BATT} = 2.7$  to  $3.6$  V,  $VREFH0 = 2.7$  V to  $AVCC0$ ,  
 $VSS = AVSS0 = VREFL/VREFLO = VSS\_USB = 0$  V,  $T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit
Operation frequency	System clock (ICLK)	f	—	—	100	MHz
	Peripheral module clock (PCLKB)		—*1	—	50	
	FlashIF clock (FCLK)		—*2	—	50	
	External bus clock (BCLK)		—	—	50	
	BCLK pin output		—	—	25	
	USB clock (UCLK)		—	—	48	
	IEBUS clock (IECLK)		—	—	44.03	

Note 1. The PCLKB must run at a frequency of at least 24 MHz if the USB is in use.

Note 2. The FCLK must run at a frequency of at least 4 MHz when changing the ROM or E2 DataFlash memory contents.

**Table 45.7 Operation Frequency Value (Low-Speed Operating Mode 1)**

Conditions:  $V_{CC} = AVCC0 = VREFH = VCC\_USB = V_{BATT} = 2.7$  to  $3.6$  V,  $VREFH0 = 2.7$  V to  $AVCC0$ ,  
 $VSS = AVSS0 = VREFL/VREFLO = VSS\_USB = 0$  V,  $T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit
Operation frequency	System clock (ICLK)	f	—	—	1	MHz
	Peripheral module clock (PCLKB)		—	—	1	
	FlashIF clock (FCLK)		—	—	1	
	External bus clock (BCLK)		—	—	1	
	BCLK pin output		—	—	1	
	USB clock (UCLK)		—	—	1	
	IEBUS clock (IECLK)		—	—	1	

**Table 45.8 Operation Frequency Value (Low-Speed Operating Mode 2)**

Conditions:  $V_{CC} = AVCC0 = VREFH = VCC\_USB = V_{BATT} = 2.7$  to  $3.6$  V,  $VREFH0 = 2.7$  V to  $AVCC0$ ,  
 $VSS = AVSS0 = VREFL/VREFLO = VSS\_USB = 0$  V,  $T_a = T_{opr}$

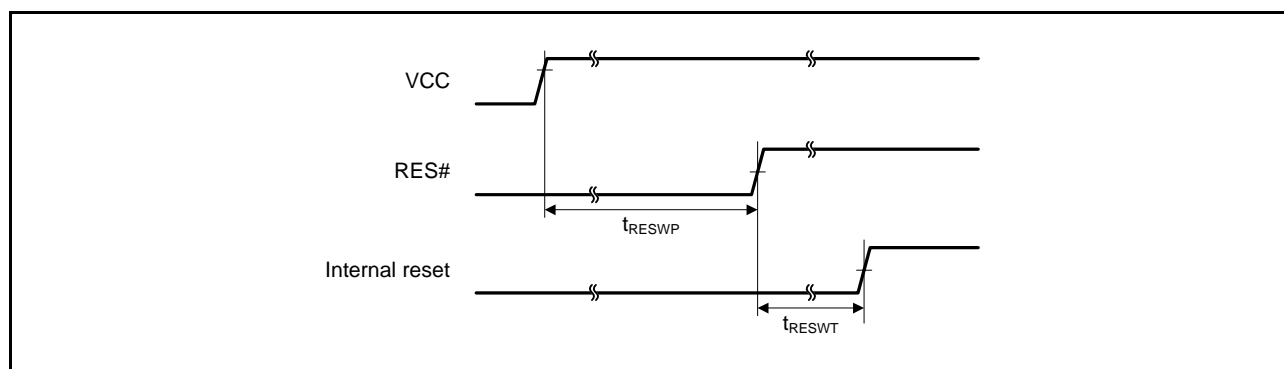
Item		Symbol	Min.	Typ.	Max.	Unit
Operation frequency	System clock (ICLK)	f	32	—	143.75	kHz
	Peripheral module clock (PCLKB)		—	—	143.75	
	FlashIF clock (FCLK)		32	—	143.75	
	External bus clock (BCLK)		—	—	143.75	
	BCLK pin output		—	—	143.75	
	USB clock (UCLK)		—	—	143.75	
	IEBUS clock (IECLK)		—	—	143.75	

### 45.3.1 Reset Timing

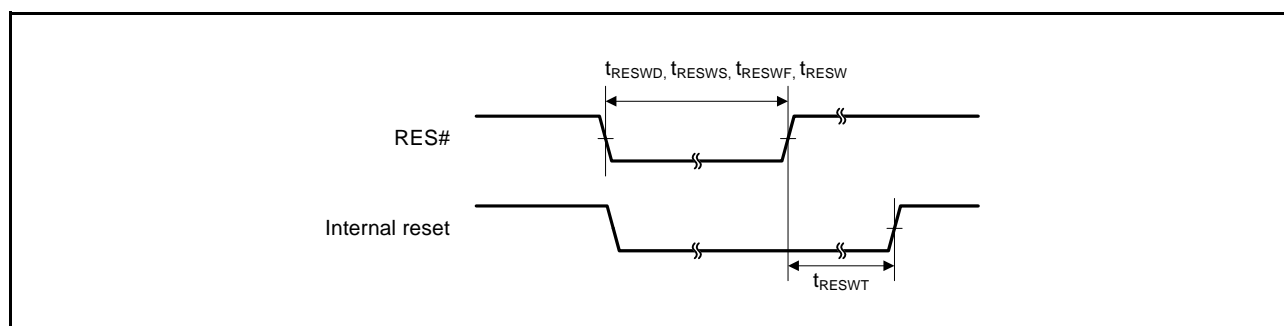
**Table 45.9 Reset Timing**

Conditions:  $V_{CC} = AVCC0 = VREFH = VCC\_USB = V_{BATT} = 2.7$  to  $3.6$  V,  $VREFH0 = 2.7$  V to  $AVCC0$ ,  
 $VSS = AVSS0 = VREFL/VREFLO = VSS\_USB = 0$  V,  $T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions
RES# pulse width	Power-on	$t_{RESWP}$	2	—	—	ms	Figure 45.1
	Deep software standby mode	$t_{RESWD}$	1	—	—	ms	Figure 45.2
	Software standby mode, low-speed operating mode 2	$t_{RESWS}$	1	—	—	ms	
	Programming or erasure of the ROM or E2 data-flash memory or blank checking of the E2 DataFlash memory	$t_{RESW}$	200	—	—	$\mu$ s	
	Other than above	$t_{RESW}$	200	—	—	$\mu$ s	
Wait time after RES# cancellation		$t_{RESWT}$	59	—	60	$t_{cyc}$	Figure 45.1
Internal reset time (independent watchdog timer reset, watchdog timer reset, software reset)		$t_{RESW2}$	112	—	120	$t_{cyc}$	



**Figure 45.1 Reset Input Timing at Power-On**



**Figure 45.2 Reset Input Timing**

## 45.3.2 Clock Timing

**Table 45.10 Clock Timing (Except for Sub-Clock Related)**

Conditions: VCC = AVCC0 = VREFH = VCC\_USB = V<sub>BATT</sub> = 2.7 to 3.6 V, VREFH0 = 2.7 V to AVCC0,  
VSS = AVSS0 = VREFL/VREFLO = VSS\_USB = 0 V, T<sub>a</sub> = T<sub>opr</sub>

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
BCLK pin output cycle time	t <sub>Bcyc</sub>	40	—	—	ns	Figure 45.3	
BCLK pin output high pulse width	t <sub>CH</sub>	15	—	—	ns		
BCLK pin output low pulse width	t <sub>CL</sub>	15	—	—	ns		
BCLK pin output rising time	t <sub>Cr</sub>	—	—	5	ns		
BCLK pin output falling time	t <sub>Cf</sub>	—	—	5	ns		
EXTAL external clock input cycle time	t <sub>EXcyc</sub>	50	—	—	ns	Figure 45.4	
EXTAL external clock input high pulse width	t <sub>EXH</sub>	20	—	—	ns		
EXTAL external clock input low pulse width	t <sub>EXL</sub>	20	—	—	ns		
EXTAL external clock rising time	t <sub>EXr</sub>	—	—	5	ns		
EXTAL external clock falling time	t <sub>EXf</sub>	—	—	5	ns		
EXTAL external clock input wait time*1	t <sub>EXWT</sub>	1	—	—	ms		
Main clock frequency	f <sub>MAIN</sub>	4	—	16	MHz		
Main clock oscillator start-up time	t <sub>MAINOSC</sub>	—	—	—*3	ms	Figure 45.5	
Main clock oscillation stabilization wait time	t <sub>MAINOSCWT</sub>	—	—	—*4	ms		
LOCO clock cycle time	t <sub>cyc</sub>	6.96	8	9.4	μs		
LOCO clock oscillation frequency	f <sub>LOCO</sub>	106.25	125	143.75	kHz		
LOCO clock oscillation stabilization wait time	t <sub>LOCOWT</sub>	—	—	20	μs	Figure 45.6	
HOCO clock oscillator oscillation frequency	f <sub>HOCO</sub>	45	50	55	MHz		
HOCO clock oscillation stabilization wait time 1*2	t <sub>HOCOWT1</sub>	—	—	1.8	ms	Figure 45.7	
HOCO clock oscillation stabilization wait time 2	t <sub>HOCOWT2</sub>	—	—	2.0	ms	Figure 45.8	
HOCO clock power supply settling time	t <sub>HOCOP</sub>	—	—	1	ms	Figure 45.9	
PLL clock frequency	f <sub>PLL</sub>	104	—	200	MHz		
PLL clock time	PLL operation started after main clock oscillation has settled	t <sub>PLL1</sub>	—	—	500	μs	Figure 45.10
PLL clock oscillation stabilization wait time		t <sub>PLLWT1</sub>	—	—	—*5	ms	
PLL clock time	PLL operation started before main clock oscillation has settled	t <sub>PLL2</sub>	—	—	t <sub>MAINOSC</sub> + t <sub>PLL1</sub>	ms	Figure 45.11
PLL clock oscillation stabilization wait time		t <sub>PLLWT2</sub>	—	—	—*5	ms	

Note 1. This is the time until the clock is used after setting P36 and P37 as inputs, and then clearing the main clock oscillator stop bit (MOSCCR.MOSTP) to 0 (selecting operation).

Note 2. This is the time until the frequency of oscillation by the HOCO (fHOCO) reaches the range for guaranteed operation. after release from the reset state.

Note 3. When using a main clock, ask the manufacturer of the oscillator to evaluate its oscillation. Refer to the results of evaluation provided by the manufacturer for the oscillation stabilization time.

Note 4. The number of cycles n selected by the value of the MOSCWTCR.MSTS[4:0] bits determines the main-clock oscillation stabilization waiting time in accord with the formula below.

$$t_{\text{MAINOSCWT}} = t_{\text{MAINOSC}} + \frac{n + 16384}{f_{\text{MAIN}}}$$

Note 5. The number of cycles n selected by the value of the PLLWTCR.PSTS[4:0] bits determines the PLL-clock oscillation stabilization waiting time in accord with the formula below.

$$t_{\text{PLLWT1}} = t_{\text{PLL1}} + \frac{n + 131072}{f_{\text{PLL}}}$$

$$t_{\text{PLLWT2}} = t_{\text{PLL2}} + \frac{n + 131072}{f_{\text{PLL}}} = t_{\text{MAINOSC}} + t_{\text{PLL1}} + \frac{n + 131072}{f_{\text{PLL}}}$$

**Table 45.11 Clock Timing (Sub-Clock Related)**

Conditions:  $V_{CC} = AVCC0 = VREFH = VCC\_USB = 2.7$  to  $3.6$  V,  $VREFH0 = 2.7$  V to  $AVCC0$ ,  $V_{BATT} = 2.3$  to  $3.6$  V,  $VSS = AVSS0 = VREFL/VREFL0 = VSS\_USB = 0$  V,  $T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Sub-clock frequency	$f_{SUB}$	—	32.768	—	kHz	
Sub-clock oscillator start-up time	$t_{SUBOSC}$	—	—	*1	s	Figure 45.12
Sub-clock oscillation stabilization wait offset time*2	$t_{SUBOSCWT0}$	1.8	—	2.6	s	
Sub-clock oscillation stabilization wait time	$t_{SUBOSCWT}$	—	—	*2	s	

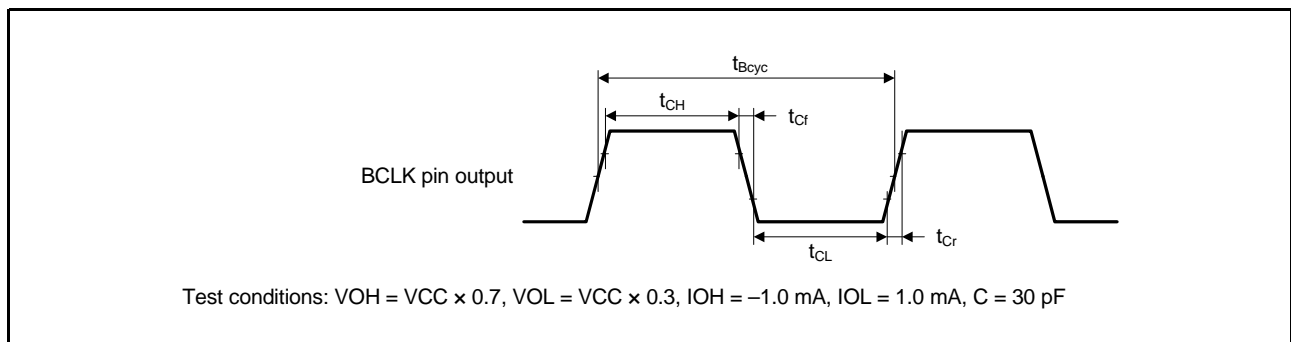
Note 1. When using a sub-clock, ask the manufacturer of the oscillator to evaluate its oscillation. Refer to the results of evaluation provided by the manufacturer for the oscillation stabilization time.

Note 2. The minimum and maximum values for sub-clock oscillation stabilization waiting offset time ( $t_{SUBOSCWT0}$ ) only apply to products tagged with “\*2” in Figure 1.3, List of Products. For other products, take the value of ( $t_{SUBOSCWT0}$ ) to be 0.

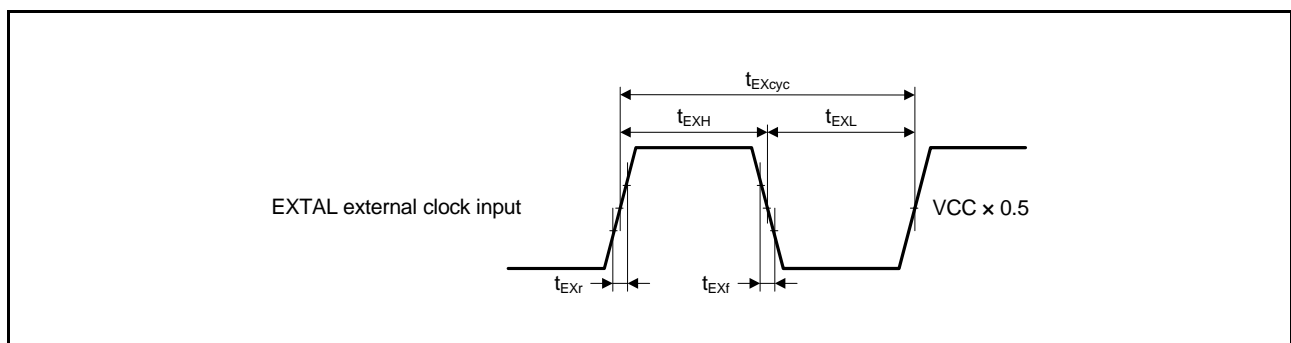
Note 3. The number of cycles n selected by the value of the SOSCWTCR.SSTS[4:0] bits determines the sub-clock oscillation stabilization waiting time in accord with the formula below.

$$t_{SUBOSCWT} = \max(t_{SUBOSC}, t_{SUBOSCWT0}) + \frac{n}{f_{SUB}}$$

The notation “ $\max(t_{SUBOSC}, t_{SUBOSCWT0})$ ” indicates whichever is higher of  $t_{SUBOSC}$  and  $t_{SUBOSCWT0}$ .



**Figure 45.3 BCLK Pin Output Timing**



**Figure 45.4 EXTAL External Clock Input Timing**

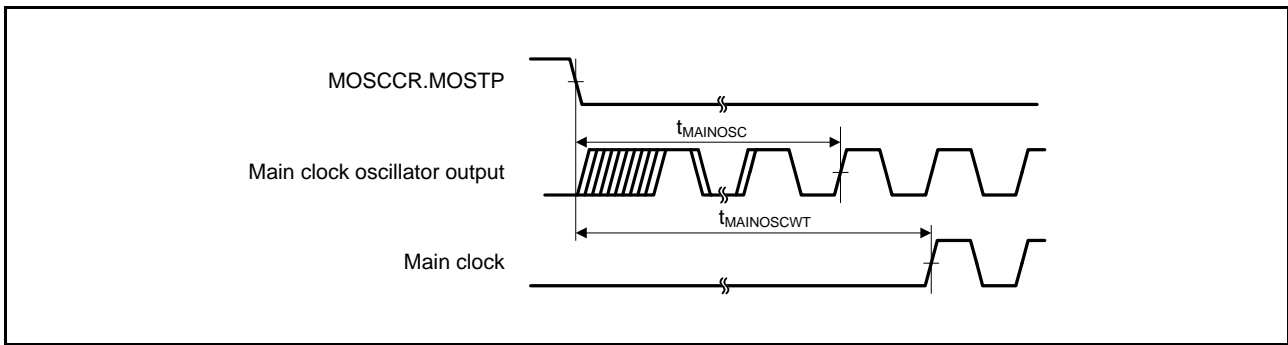


Figure 45.5 Main Clock Oscillation Start Timing

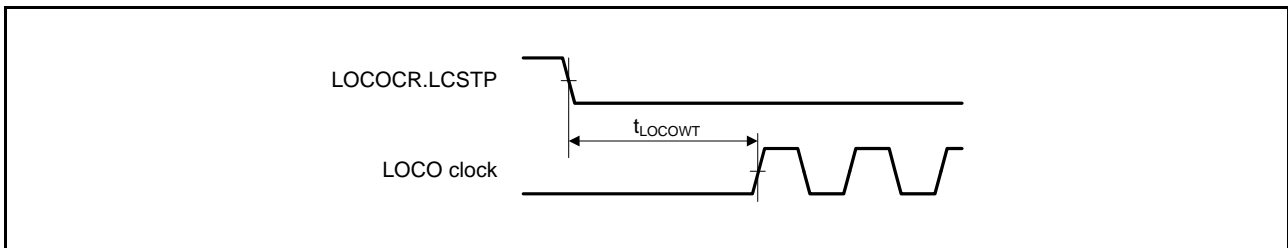


Figure 45.6 LOCO Clock Oscillation Start Timing

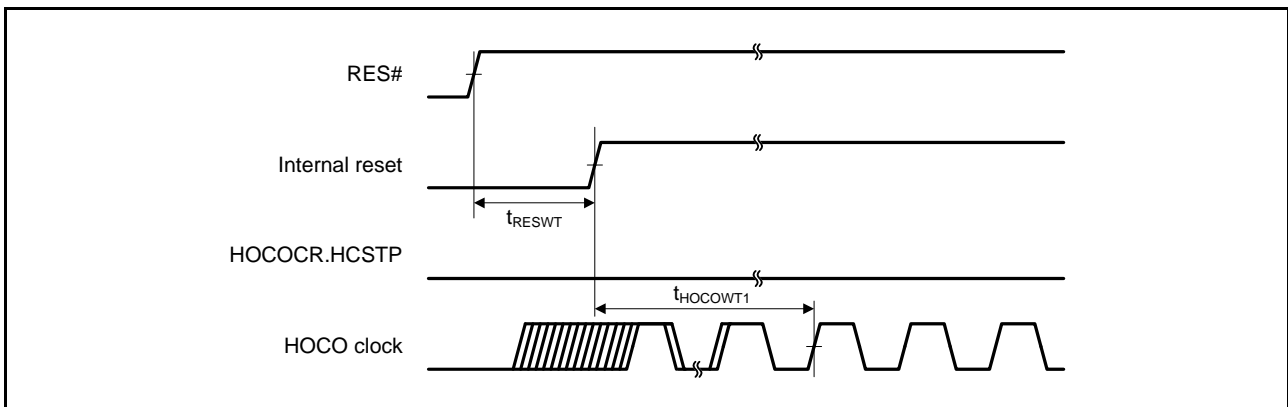


Figure 45.7 HOCO Oscillation Start Timing (After Reset is Canceled by Setting the OFS1.HOCOEN Bit to 0)

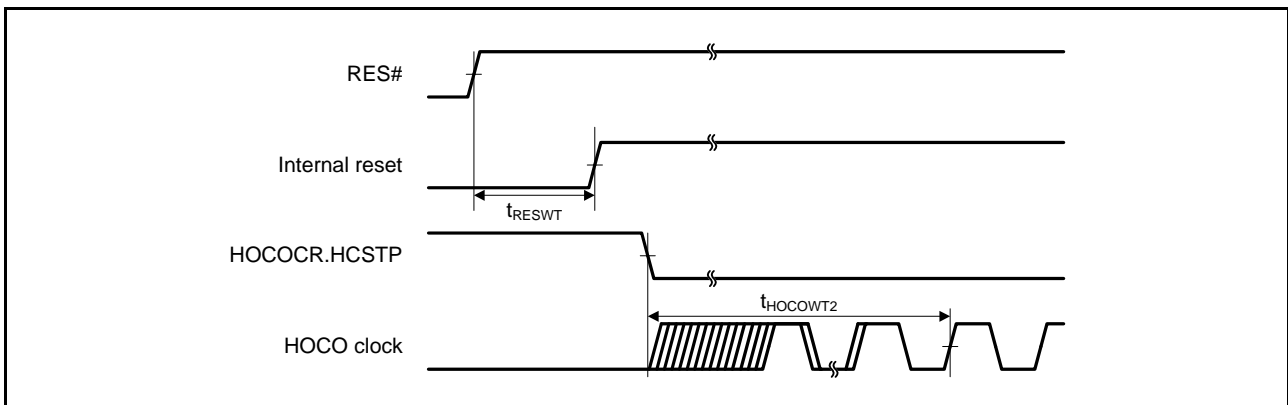


Figure 45.8 HOCO Clock Oscillation Start Timing (Oscillation is Started by Setting the HOCOCR.HCSTP Bit)

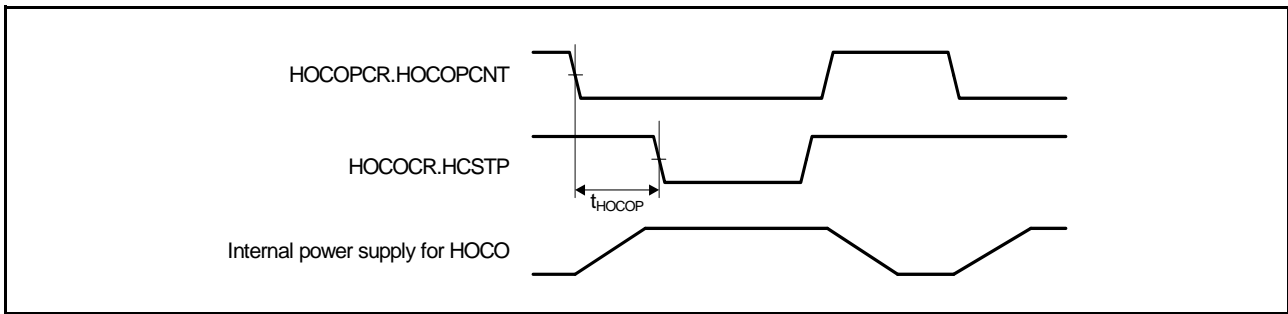


Figure 45.9 HOCO Power Supply Control Timing

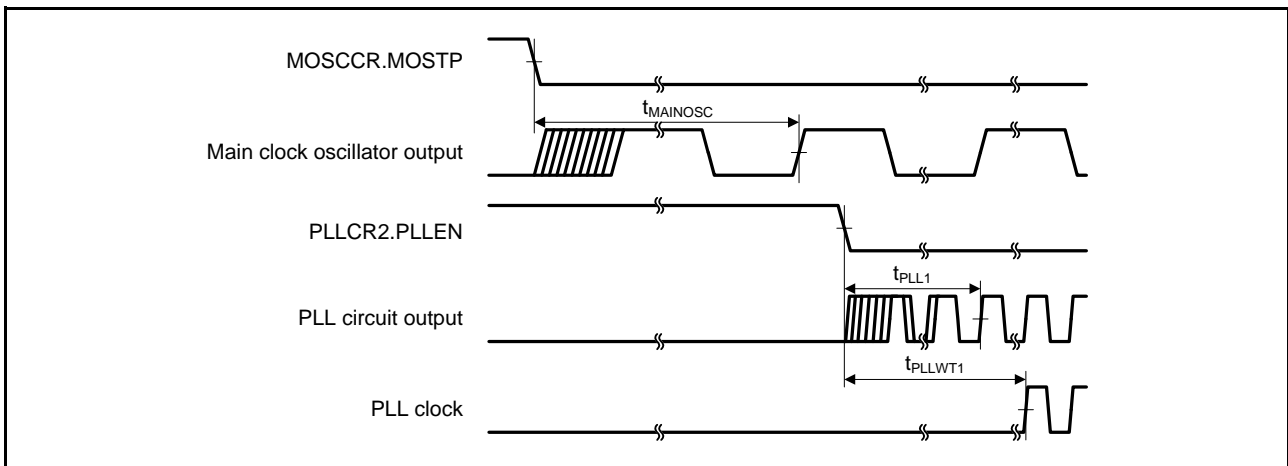


Figure 45.10 PLL Clock Oscillation Start Timing (PLL is Operated after Main Clock Oscillation Has Settled)

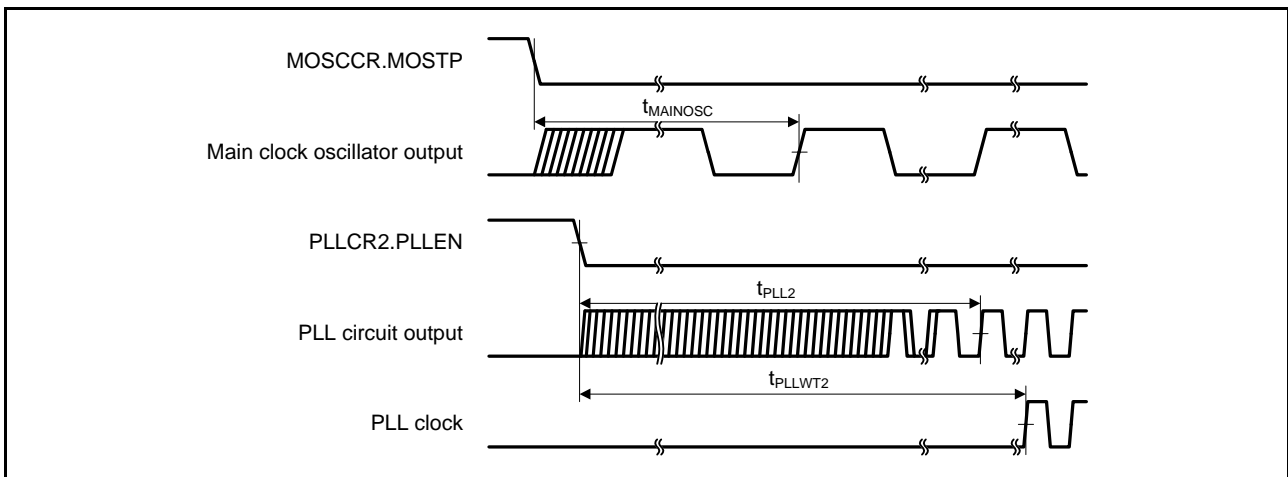


Figure 45.11 PLL Clock Oscillation Start Timing (PLL is Operated before Main Clock Oscillation Has Settled)

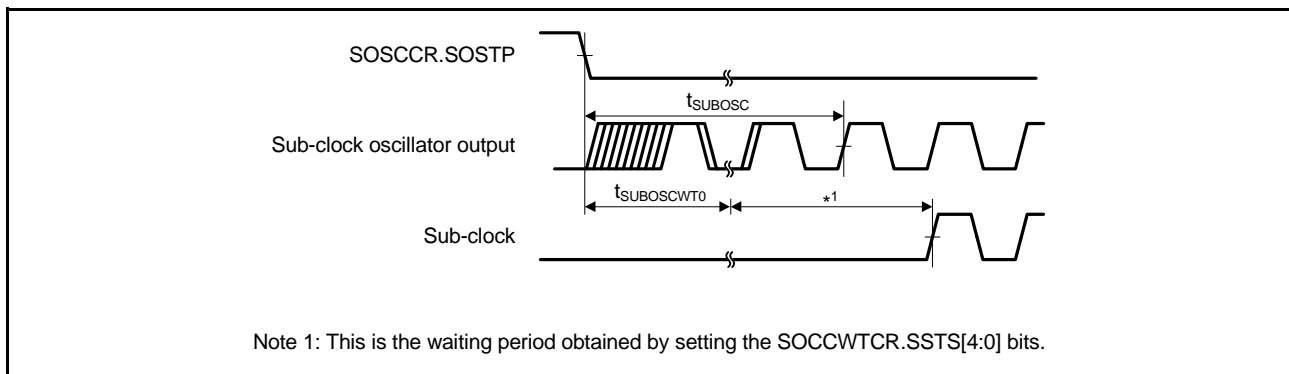


Figure 45.12 Sub-Clock Oscillation Start Timing

### 45.3.3 Timing of Recovery from Low Power Consumption Modes

Table 45.12 Timing of Recovery from Low Power Consumption Modes

Conditions: VCC = AVCC0 = VREFH = VCC\_USB = V<sub>BATT</sub> = 2.7 to 3.6 V, VREFH0 = 2.7 V to AVCC0, VSS = AVSS0 = VREFL/VREFLO = VSS\_USB = 0 V, T<sub>a</sub> = T<sub>opr</sub>

Item			Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Recovery time after cancellation of software standby mode	Crystal resonator connected to main clock oscillator	Main clock oscillator operating	t <sub>SBYMC</sub>	10	—	—	ms	Figure 45.13
		Main clock oscillator and PLL circuit operating	t <sub>SBYPC</sub>	10	—	—	ms	
	External clock input to main clock oscillator	Main clock oscillator operating	t <sub>SBYEX</sub>	1	—	—	ms	
		Main clock oscillator and PLL circuit operating	t <sub>SBYPE</sub>	1	—	—	ms	
	Sub-clock oscillator operating		t <sub>SBYSC</sub>	2	—	—	s	
	High-speed on-chip oscillator operating		t <sub>SBYHO</sub>	—	—	2	ms	
	Low-speed on-chip oscillator or IWDG-dedicated on-chip oscillator operating		t <sub>SBYLO</sub>	—	—	800	μs	
Recovery time after cancellation of deep software standby mode			t <sub>DSBY</sub>	—	—	1.0	ms	Figure 45.14
Wait time after cancellation of deep software standby mode			t <sub>DSBYWT</sub>	45	—	46	t <sub>cyc</sub>	

Note: • The wait time varies depending on the state in which each oscillator was when the WAIT instruction was executed. The recovery time when multiple oscillators are operating is the same period as that when the oscillator which requires the longest time of all operating oscillators to recover is operating alone.

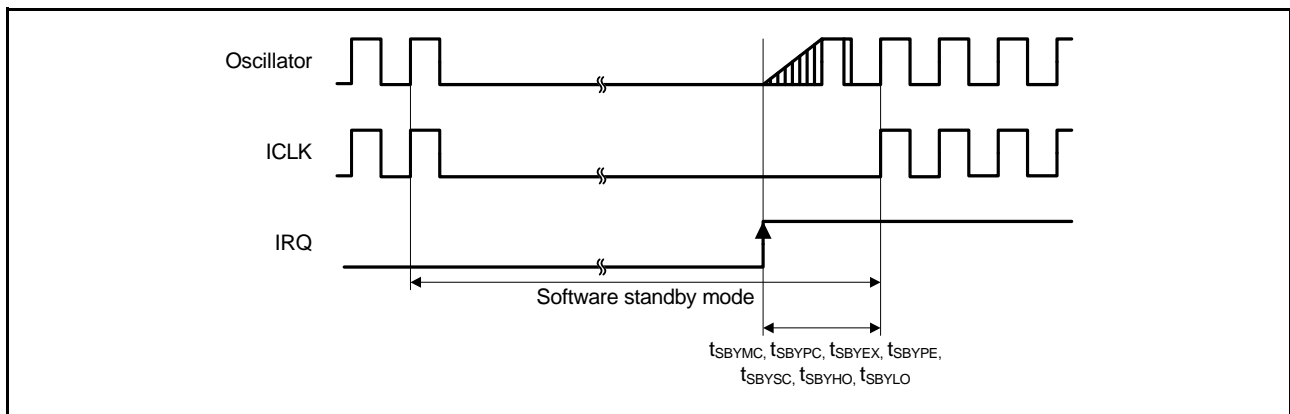


Figure 45.13 Software Standby Mode Cancellation Timing

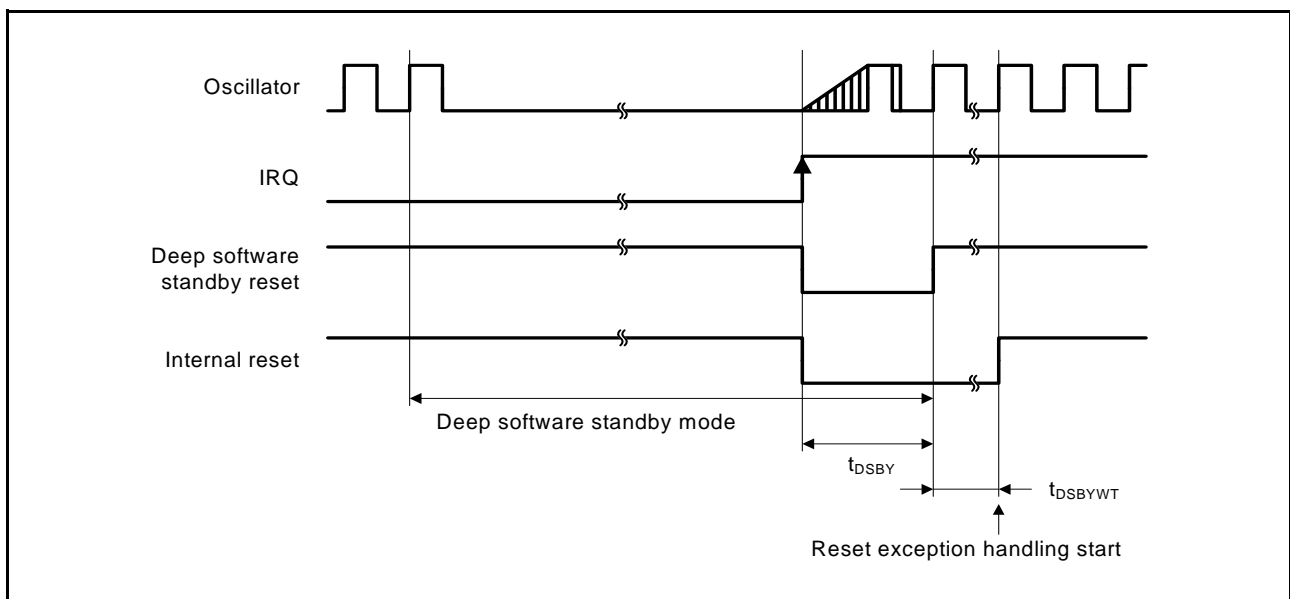


Figure 45.14 Deep Software Standby Mode Cancellation Timing

### 45.3.4 Control Signal Timing

Table 45.13 Control Signal Timing

Conditions:  $V_{CC} = AV_{CC0} = V_{REFH} = V_{CC\_USB} = V_{BATT} = 2.7$  to  $3.6$  V,  $V_{REFH0} = 2.7$  V to  $AV_{CC0}$ ,  
 $V_{SS} = AV_{SS0} = V_{REFL}/V_{REFL0} = V_{SS\_USB} = 0$  V,  $T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
NMI pulse width	$t_{NMIW}$	200	—	—	ns	$t_c (PCLK) \times 2 \leq 200$ ns Figure 45.15
		$t_c (PCLK) \times 2$				$t_c (PCLK) \times 2 > 200$ ns Figure 45.15
IRQ pulse width	$t_{IRQW}$	200	—	—	ns	$t_c (PCLK) \times 2 \leq 200$ ns Figure 45.16
		$t_c (PCLK) \times 2$				$t_c (PCLK) \times 2 > 200$ ns Figure 45.16



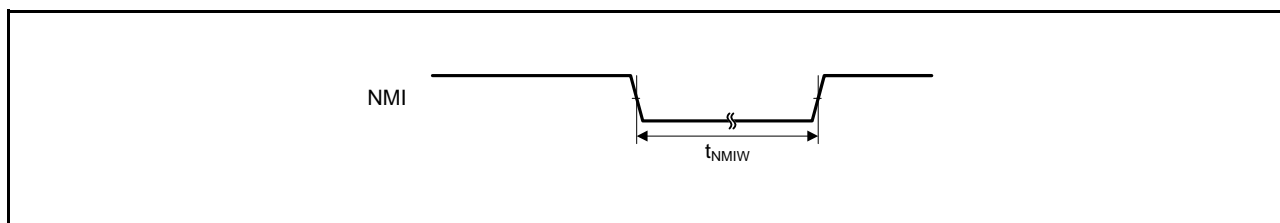


Figure 45.15 NMI Interrupt Input Timing

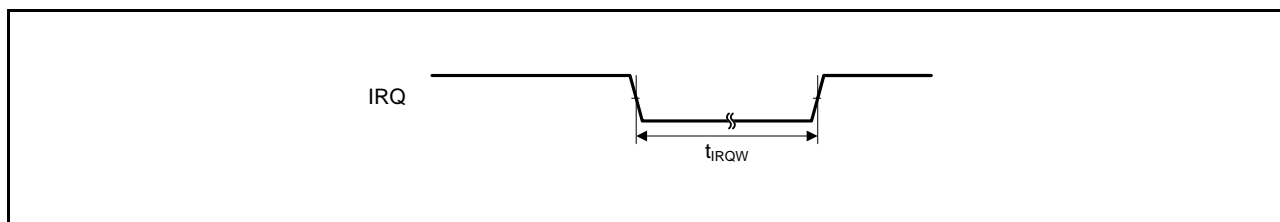


Figure 45.16 IRQ Interrupt Input Timing

### 45.3.5 Bus Timing

**Table 45.14 Bus Timing**

Conditions:  $V_{CC} = AVCC0 = VREFH = VCC\_USB = 2.7$  to  $3.6$  V,  $VREFH0 = 2.7$  V to  $AVCC0$ ,  
 $VSS = AVSS0 = VREFL/VREFL0 = VSS\_USB = 0$  V,  
 $ICLK = 8$  to  $100$  MHz,  $BCLK = 8$  to  $50$  MHz,  $T_a = T_{opr}$   
 Output load conditions:  $V_{OH} = VCC \times 0.5$ ,  $V_{OL} = VCC \times 0.5$ ,  $I_{OH} = -1.0$  mA,  $I_{OL} = 1.0$  mA,  $C = 30$  pF  
 High drive output is selected by the drive capacity control register.

Item	Symbol	Min.	Max.	Unit	Test Conditions
Address delay time	$t_{AD}$	—	20	ns	Figure 45.17 to Figure 45.22
Byte control delay time	$t_{BCD}$	—	20	ns	
CS# delay time	$t_{CSD}$	—	20	ns	
ALE delay time	$t_{ALED}$	—	20	ns	
RD# delay time	$t_{RSD}$	—	20	ns	
Read data setup time	$t_{RDS}$	15	—	ns	
Read data hold time	$t_{RDH}$	0	—	ns	
WR# delay time	$t_{WRD}$	—	20	ns	
Write data delay time	$t_{WDD}$	—	20	ns	
Write data hold time	$t_{WDH}$	0	—	ns	
WAIT# setup time	$t_{WTS}$	15	—	ns	
WAIT# hold time	$t_{WTH}$	0	—	ns	

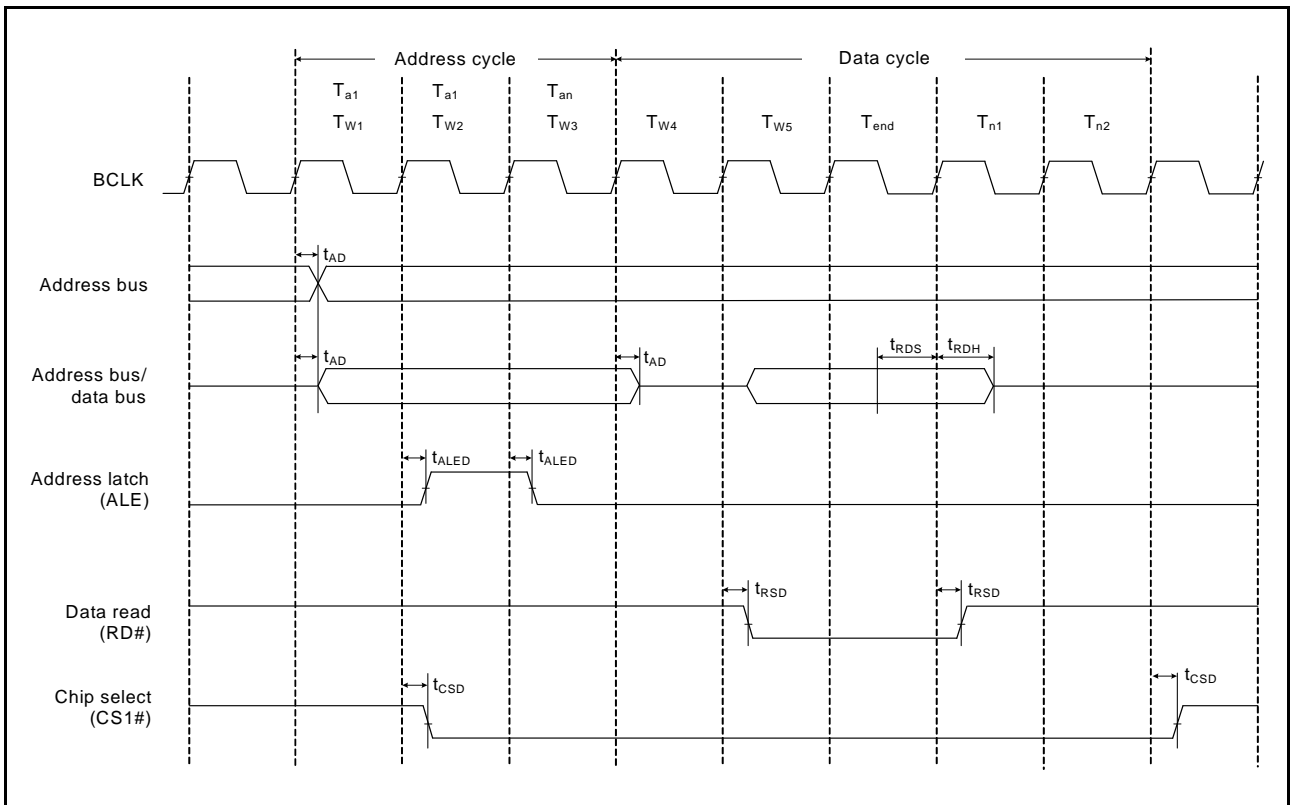


Figure 45.17 Address/Data Multiplexed Bus Read Access Timing

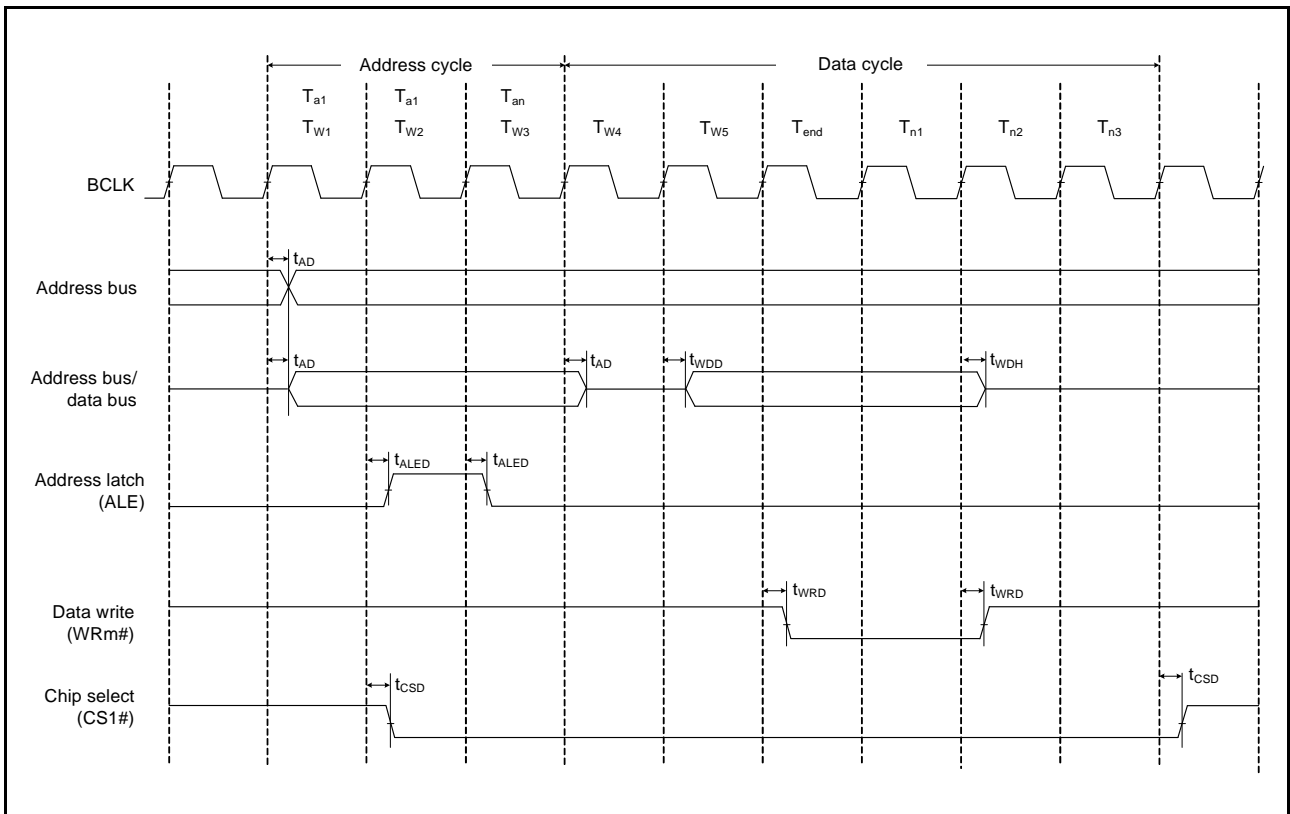


Figure 45.18 Address/Data Multiplexed Bus Write Access Timing

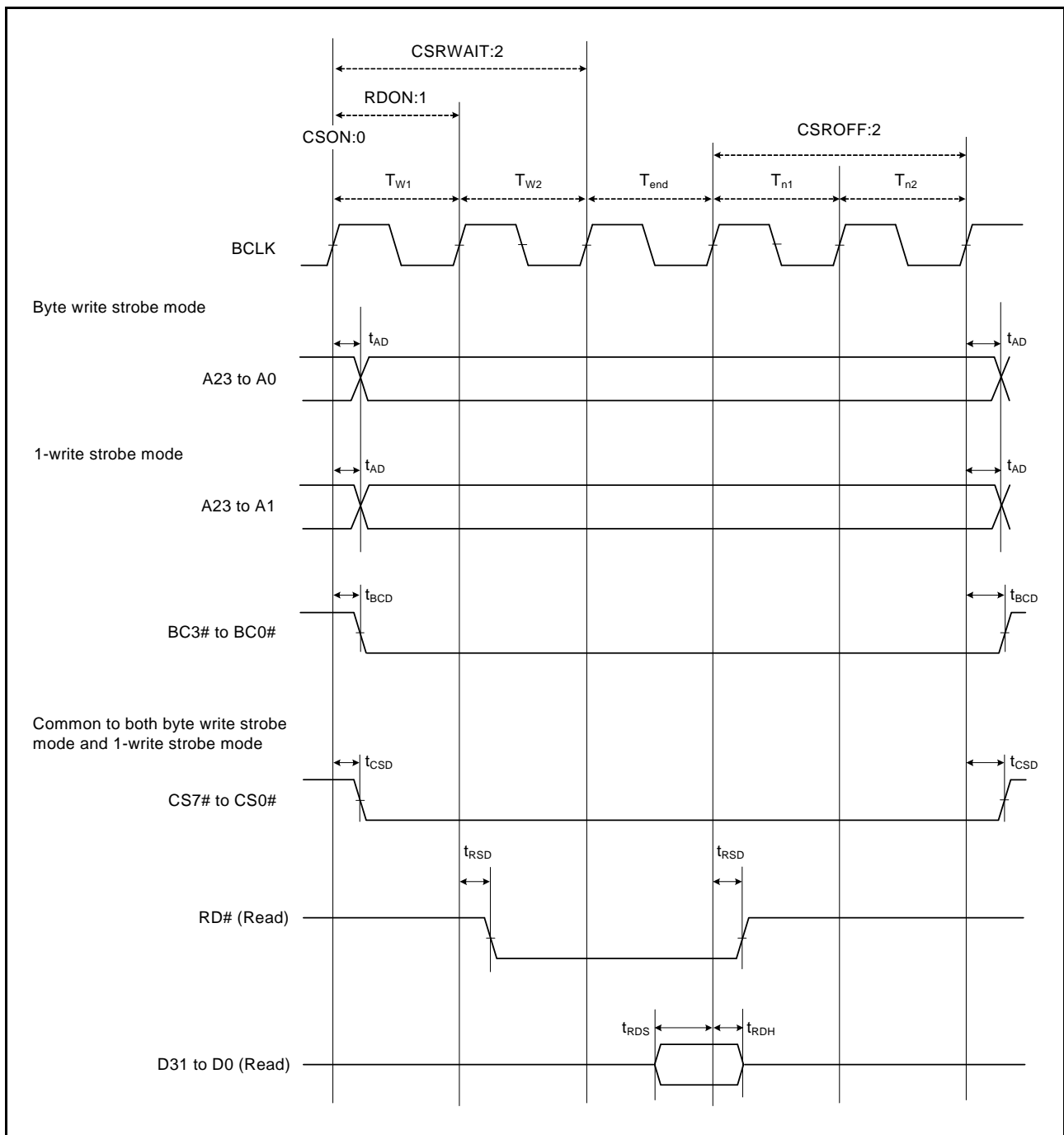


Figure 45.19 External Bus Timing/Normal Read Cycle (Bus Clock Synchronized)

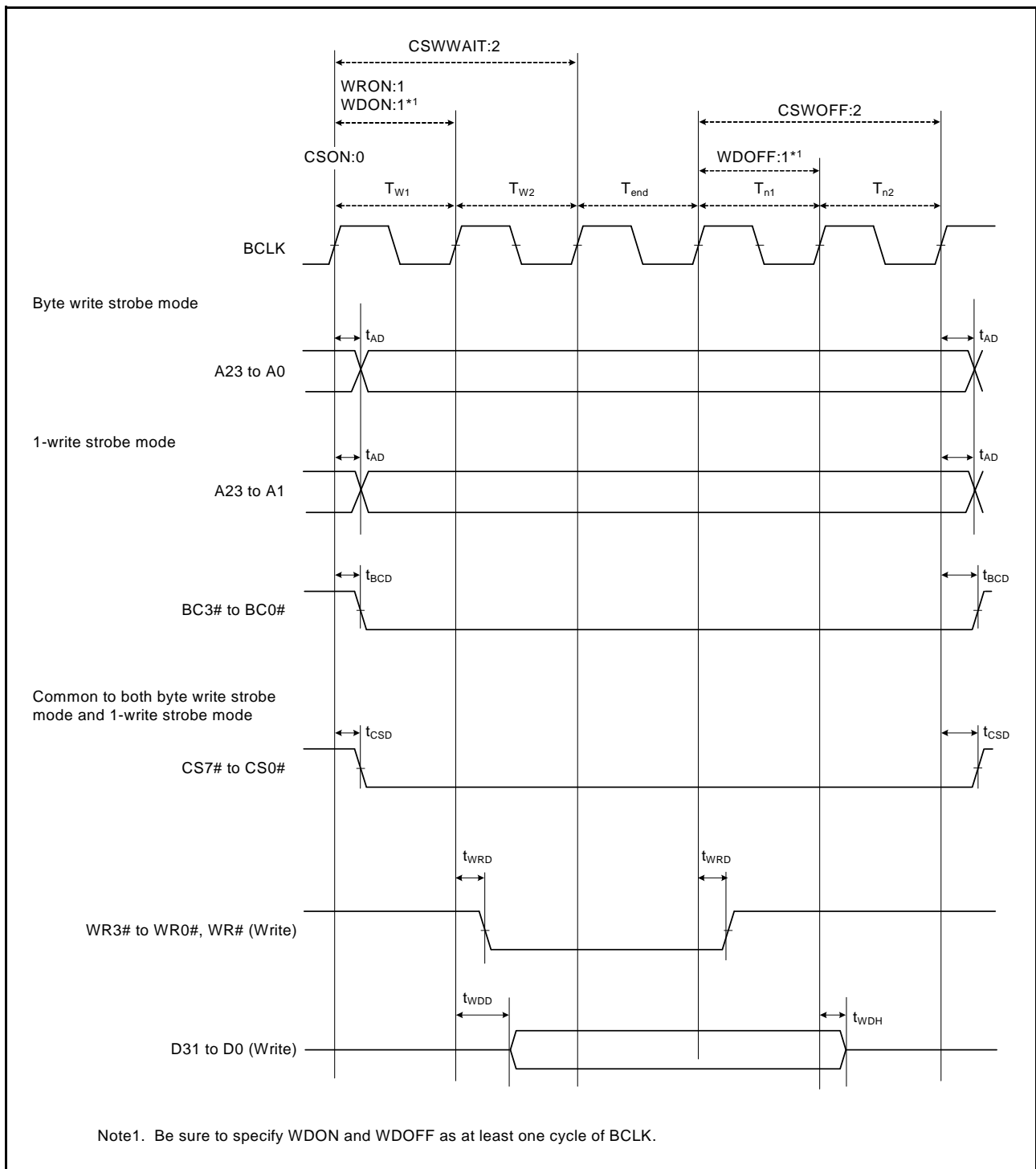


Figure 45.20 External Bus Timing/Normal Write Cycle (Bus Clock Synchronized)

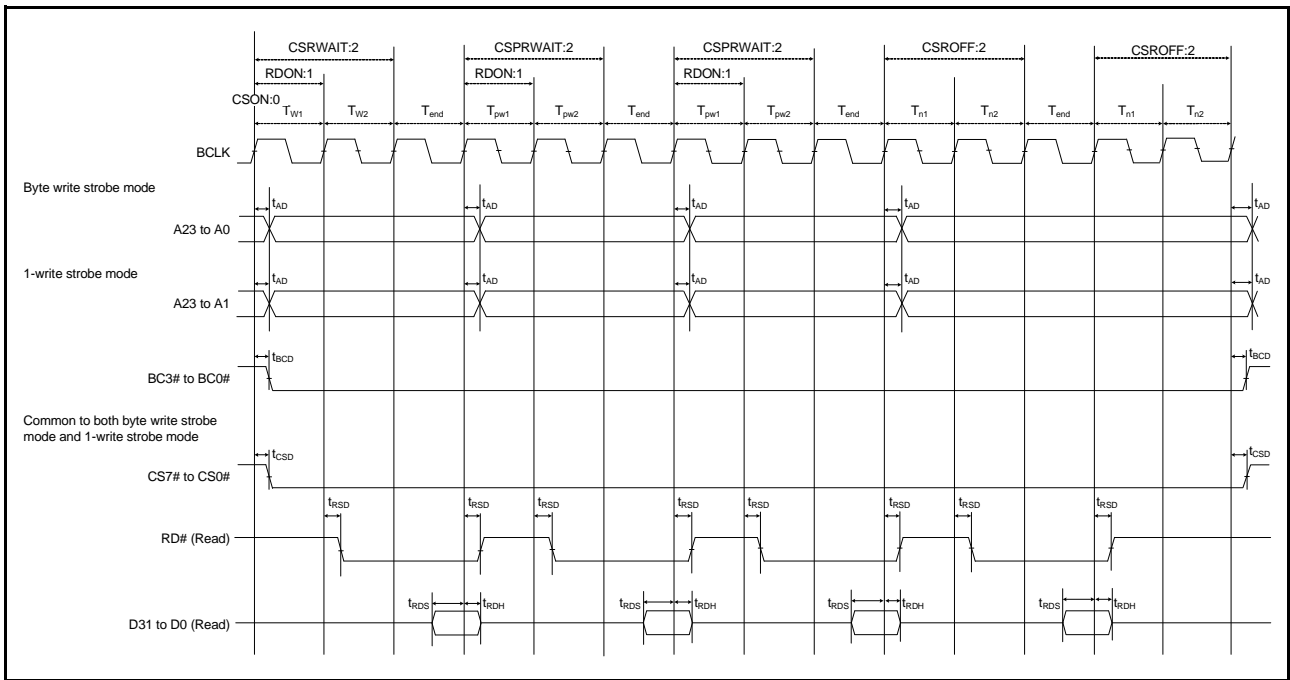


Figure 45.21 External Bus Timing/Page Read Cycle (Bus Clock Synchronized)

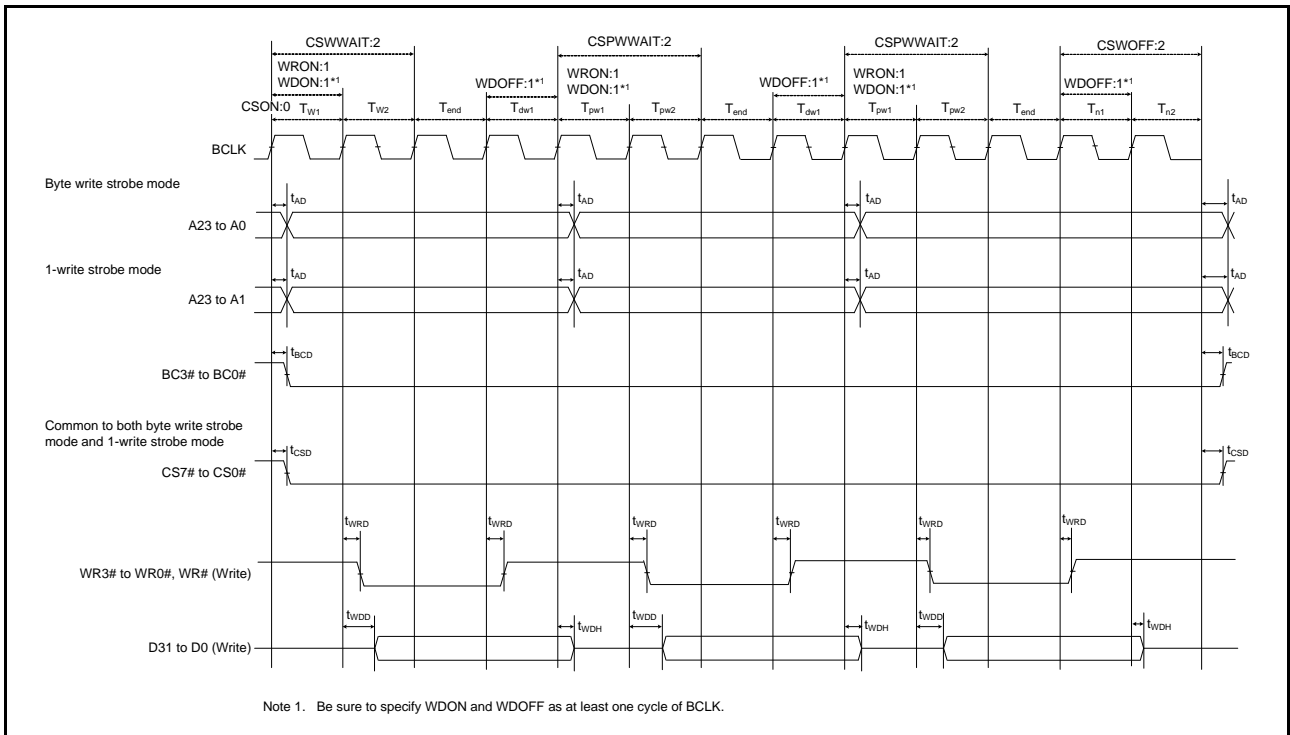


Figure 45.22 External Bus Timing/Page Write Cycle (Bus Clock Synchronized)

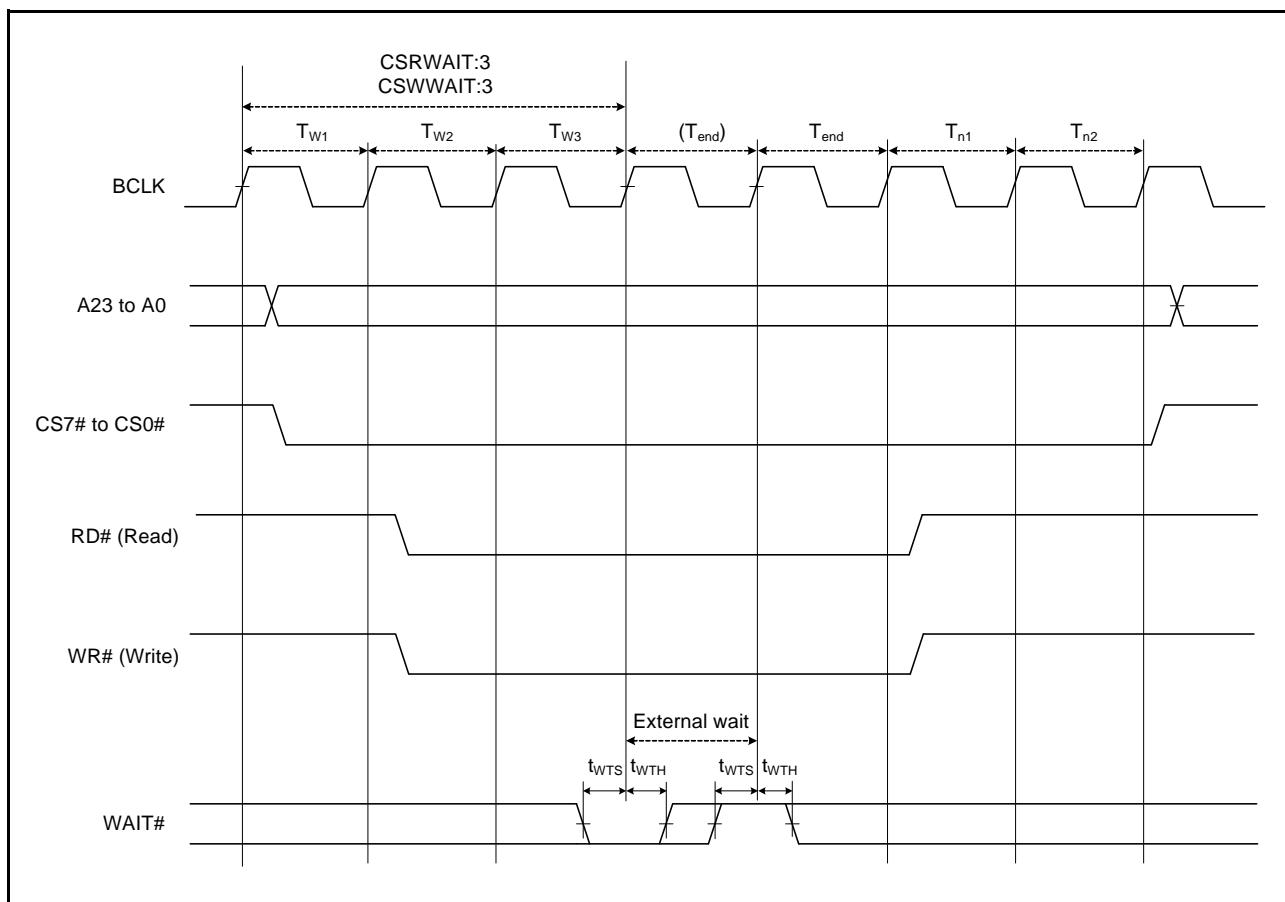


Figure 45.23 External Bus Timing/External Wait Control

## 45.3.6 Timing of On-Chip Peripheral Modules

**Table 45.15 Timing of On-Chip Peripheral Modules (1)**

Conditions: VCC = AVCC0 = VREFH = VCC\_USB = 2.7 to 3.6 V, VREFH0 = 2.7 V to AVCC0,  
VSS = AVSS0 = VREFL/VREFL0 = VSS\_USB = 0 V  
PCLK = 8 to 50 MHz  
T<sub>a</sub> = T<sub>opr</sub>  
High drive output is selected by the drive capacity control register.

Item		Symbol	Min.	Max.	Unit*1	Test Conditions	
I/O ports	Input data pulse width	t <sub>PRW</sub>	1.5	—	t <sub>Pcyc</sub>	Figure 45.24	
MTU/TPU	Input capture input pulse width	Single-edge setting	t <sub>TICW</sub>	1.5	—	t <sub>Pcyc</sub>	Figure 45.25
		Both-edge setting		2.5	—		
	Timer clock pulse width	Single-edge setting	t <sub>TCKWH</sub> , t <sub>TCKWL</sub>	1.5	—	t <sub>Pcyc</sub>	Figure 45.26
Both-edge setting		2.5		—			
Phase counting mode		2.5		—			
POE	POE# input pulse width	t <sub>POEW</sub>	1.5	—	t <sub>Pcyc</sub>	Figure 45.27	
8-bit timer	Timer clock pulse width	Single-edge setting	t <sub>TMCWH</sub> , t <sub>TMCWL</sub>	1.5	—	t <sub>Pcyc</sub>	Figure 45.28
		Both-edge setting		2.5	—		
SCI	Input clock cycle	Asynchronous	t <sub>Scyc</sub>	4	—	t <sub>Pcyc</sub>	Figure 45.29
		Clock synchronous		6	—		
	Input clock pulse width		t <sub>SCKW</sub>	0.4	0.6	t <sub>Scyc</sub>	
	Input clock rise time		t <sub>SCKr</sub>	—	20	ns	
	Input clock fall time		t <sub>SCKf</sub>	—	20	ns	
	Output clock cycle	Asynchronous	t <sub>Scyc</sub>	16	—	t <sub>Pcyc</sub>	
		Clock synchronous		4	—		
	Output clock pulse width		t <sub>SCKW</sub>	0.4	0.6	t <sub>Scyc</sub>	
	Output clock rise time		t <sub>SCKr</sub>	—	20	ns	
	Output clock fall time		t <sub>SCKf</sub>	—	20	ns	
	Transmit data delay time	Clock synchronous	t <sub>TXD</sub>	—	40	ns	Figure 45.30
Receive data setup time	Clock synchronous	t <sub>RXS</sub>	40	—	ns		
Receive data hold time	Clock synchronous	t <sub>RXH</sub>	40	—	ns		
A/D converter	10-bit A/D converter trigger input pulse width	t <sub>TRGW</sub>	1.5	—	t <sub>Pcyc</sub>	Figure 45.31	
	12-bit A/D converter trigger input pulse width		1.5	—			

Note 1. t<sub>Pcyc</sub>: PCLK cycle

**Table 45.16 Timing of On-Chip Peripheral Modules (2)**

Conditions:  $V_{CC} = AV_{CC0} = V_{REFH} = V_{CC\_USB} = 2.7$  to  $3.6$  V\*1,  $V_{REFH0} = 3.0$  V to  $AV_{CC0}$ \*1,  
 $V_{SS} = AV_{SS0} = V_{REFL}/V_{REFL0} = V_{SS\_USB} = 0$  V,  
 $PCLK = 8$  to  $50$  MHz,  
 $T_a = T_{opr}$   
 High drive output is selected by the drive capacity control register.

Item			Symbol	Min.	Max.	Unit*2	Test Conditions	
RSPI	RSPCK clock cycle	Master	$t_{SPcyc}$	2	4096	$t_{Pcyc}$	C = 30pF, Figure 45.32	
		Slave		8	4096			
RSPCK clock high pulse width	Master	$t_{SPCKWH}$	$(t_{SPcyc} - t_{SPCKR} - t_{SPCKF}) / 2 - 3$	—	ns			
	Slave		$(t_{SPcyc} - t_{SPCKR} - t_{SPCKF}) / 2$	—				
RSPCK clock low pulse width	Master	$t_{SPCKWL}$	$(t_{SPcyc} - t_{SPCKR} - t_{SPCKF}) / 2 - 3$	—	ns			
	Slave		$(t_{SPcyc} - t_{SPCKR} - t_{SPCKF}) / 2$	—				
RSPCK clock rise/fall time	Output	$t_{SPCKr},$ $t_{SPCKf}$	—	5	ns			
	Input		—	1	$\mu$ s			
Data input setup time	Master	$V_{CC} \geq 3.0$ V $V_{CC} < 3.0$ V	$t_{SU}$	15	—	ns		C = 30pF, Figure 45.33 to Figure 45.36
				20	—			
	Slave	$20 - t_{Pcyc}$		—				
Data input hold time	Master	$t_H$	0	—	ns			
	Slave		$20 + 2 \times t_{Pcyc}$	—				
SSL setup time	Master	$t_{LEAD}$	1	8	$t_{SPcyc}$			
	Slave		4	—	$t_{Pcyc}$			
SSL hold time	Master	$t_{LAG}$	1	8	$t_{SPcyc}$			
	Slave		4	—	$t_{Pcyc}$			
Data output delay time	Master	$t_{OD}$	—	18	ns			
	Slave		—	$3 \times t_{Pcyc} + 40$				
Data output hold time	Master	$t_{OH}$	0	—	ns			
	Slave		0	—				
Successive transmission delay time	Master	$t_{TD}$	$t_{SPcyc} + 2 \times t_{Pcyc}$	$8 \times t_{SPcyc} + 2 \times t_{Pcyc}$	ns			
	Slave		$4 \times t_{Pcyc}$	—				
MOSI and MISO rise/fall time	Output	$t_{Dr}, t_{Df}$	—	5	ns			
	Input		—	1		$\mu$ s		
SSL rise/fall time	Output	$t_{SSLr},$ $t_{SSLf}$	—	5	ns			
	Input		—	1		$\mu$ s		
Slave access time			$t_{SA}$	—	4	$t_{Pcyc}$	C = 30pF, Figure 45.35 and Figure 45.36	
Slave output release time			$t_{REL}$	—	3	$t_{Pcyc}$		

Note 1. When operation at 3.0 V or a lower voltage is needed, please contact a Renesas sales office.

Note 2.  $t_{Pcyc}$ : PCLK cycle



**Table 45.17 Timing of On-Chip Peripheral Modules (3)**

Conditions:  $V_{CC} = AV_{CC0} = V_{REFH} = V_{CC\_USB} = 2.7$  to  $3.6$  V,  $V_{REFH0} = 2.7$  V to  $AV_{CC0}$   
 $V_{SS} = AV_{SS0} = V_{REFL}/V_{REFL0} = V_{SS\_USB} = 0$  V  
 $PCLK = 8$  to  $50$  MHz  
 $T_a = T_{opr}$   
 High drive output is selected by the drive capacity control register.

Item		Symbol	Min.	Max.	Unit*1	Test Conditions
Simple SPI	SCK clock cycle output (master)	$t_{SPcyc}$	4	65536	$t_{Pcyc}$	Figure 45.32
	SCK clock cycle input (slave)		8	65536		
	SCK clock high pulse width	$t_{SPCKWH}$	0.4	0.6	$t_{SPcyc}$	
	SCK clock low pulse width	$t_{SPCKWL}$	0.4	0.6	$t_{SPcyc}$	
	SCK clock rise/fall time	$t_{SPCKr}, t_{SPCKf}$	—	20	ns	
	Data input setup time	$t_{SU}$	40	—	ns	Figure 45.33 to Figure 45.36
	Data input hold time	$t_{H}$	40	—	ns	
	SS input setup time	$t_{LEAD}$	1	—	$t_{SPcyc}$	
	SS input hold time	$t_{LAG}$	1	—	$t_{SPcyc}$	
	Data output delay time	$t_{OD}$	—	40	ns	
	Data output hold time	$t_{OH}$	-10	—	ns	
	Data rise/fall time	$t_{Dr}, t_{Df}$	—	20	ns	
	SS input rise/fall time	$t_{SSLr}, t_{SSLf}$	—	20	ns	
	Slave access time	$t_{SA}$	—	5	$t_{Pcyc}$	Figure 45.35 and Figure 45.36
	Slave output release time	$t_{REL}$	—	5	$t_{Pcyc}$	

Note 1.  $t_{Pcyc}$ : PCLK cycle

**Table 45.18 Timing of On-Chip Peripheral Modules (4)**

Conditions:  $V_{CC} = AV_{CC0} = V_{REFH} = V_{CC\_USB} = 2.7$  to  $3.6$  V,  $V_{REFH0} = 2.7$  V to  $AV_{CC0}$   
 $V_{SS} = AV_{SS0} = V_{REFL}/V_{REFLO} = V_{SS\_USB} = 0$  V  
 $PCLK = 8$  to  $50$  MHz  
 $T_a = T_{opr}$   
 High drive output is selected by the drive capacity control register.

Item		Symbol	Min.*1,*2	Max.	Unit	Test Conditions
IIC (Standard-mode, SMBus) $ICFER.FMPE = 0$	SCL input cycle time	$t_{SCL}$	$6(12) \times t_{IICcyc} + 1300$	—	ns	Figure 45.37
	SCL input high pulse width	$t_{SCLH}$	$3(6) \times t_{IICcyc} + 300$	—	ns	
	SCL input low pulse width	$t_{SCLL}$	$3(6) \times t_{IICcyc} + 300$	—	ns	
	SCL, SDA input rise time	$t_{Sr}$	—	1000	ns	
	SCL, SDA input fall time	$t_{Sf}$	—	300	ns	
	SCL, SDA input spike pulse removal time	$t_{SP}$	0	$1(4) \times t_{IICcyc}$	ns	
	SDA input bus free time	$t_{BUF}$	$3(6) \times t_{IICcyc} + 300$	—	ns	
	Start condition input hold time	$t_{STAH}$	$t_{IICcyc} + 300$	—	ns	
	Restart condition input setup time	$t_{STAS}$	1000	—	ns	
	Stop condition input setup time	$t_{STOS}$	1000	—	ns	
	Data input setup time	$t_{SDAS}$	$t_{IICcyc} + 50$	—	ns	
	Data input hold time	$t_{SDAH}$	0	—	ns	
	SCL, SDA capacitive load	$C_b$	—	400	pF	
IIC (Fast-mode)	SCL input cycle time	$t_{SCL}$	$6(12) \times t_{IICcyc} + 600$	—	ns	
	SCL input high pulse width	$t_{SCLH}$	$3(6) \times t_{IICcyc} + 300$	—	ns	
	SCL input low pulse width	$t_{SCLL}$	$3(6) \times t_{IICcyc} + 300$	—	ns	
	SCL, SDA input rise time	$t_{Sr}$	$20 + 0.1C_b$	300	ns	
	SCL, SDA input fall time	$t_{Sf}$	$20 + 0.1C_b$	300	ns	
	SCL, SDA input spike pulse removal time	$t_{SP}$	0	$1(4) \times t_{IICcyc}$	ns	
	SDA input bus free time	$t_{BUF}$	$3(6) \times t_{IICcyc} + 300$	—	ns	
	Start condition input hold time	$t_{STAH}$	$t_{IICcyc} + 300$	—	ns	
	Restart condition input setup time	$t_{STAS}$	300	—	ns	
	Stop condition input setup time	$t_{STOS}$	300	—	ns	
	Data input setup time	$t_{SDAS}$	$t_{IICcyc} + 50$	—	ns	
	Data input hold time	$t_{SDAH}$	0	—	ns	
	SCL, SDA capacitive load	$C_b$	—	400	pF	

Note: •  $t_{IICcyc}$ : IIC internal reference clock (IIC $\phi$ ) Cycle

Note 1. The value within parentheses is applicable when the value of the ICMR3.NF[1:0] bits is 11b while the digital filter is enabled by the setting ICFER.NFE = 1.

Note 2.  $C_b$  is the total capacitance of the bus lines.

**Table 45.19 Timing of On-Chip Peripheral Modules (5)**

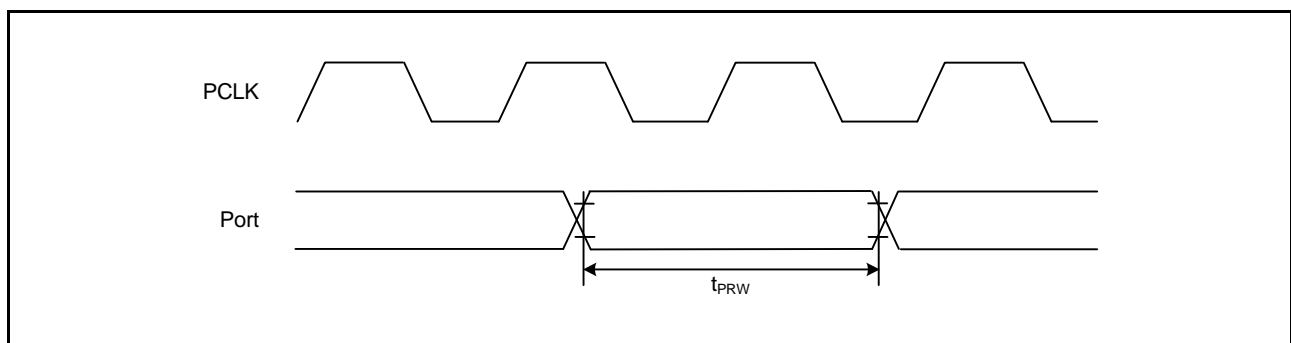
Conditions: VCC = AVCC0 = VREFH = VCC\_USB = 2.7 to 3.6 V, VREFH0 = 2.7 V to AVCC0  
VSS = AVSS0 = VREFL/VREFLO = VSS\_USB = 0 V  
PCLK = 8 to 50 MHz  
T<sub>a</sub> = T<sub>opr</sub>  
High drive output is selected by the drive capacity control register.

Item		Symbol	Min.* *2	Max.*	Unit	Test Conditions
RIIC (Fast-mode+) ICFER.FMPE = 1	SCL input cycle time	t <sub>SCL</sub>	6(12) × t <sub>IICcyc</sub> + 240	—	ns	Figure 45.37
	SCL input high pulse width	t <sub>SCLH</sub>	3(6) × t <sub>IICcyc</sub> + 120	—	ns	
	SCL input low pulse width	t <sub>SCLL</sub>	3(6) × t <sub>IICcyc</sub> + 120	—	ns	
	SCL, SDA input rise time	t <sub>Sr</sub>	—	120	ns	
	SCL, SDA input fall time	t <sub>Sf</sub>	—	120	ns	
	SCL, SDA input spike pulse removal time	t <sub>SP</sub>	0	1(4) × t <sub>IICcyc</sub>	ns	
	SDA input bus free time	t <sub>BUF</sub>	3(6) × t <sub>IICcyc</sub> + 120	—	ns	
	Start condition input hold time	t <sub>STAH</sub>	t <sub>IICcyc</sub> + 120	—	ns	
	Restart condition input setup time	t <sub>STAS</sub>	120	—	ns	
	Stop condition input setup time	t <sub>STOS</sub>	120	—	ns	
	Data input setup time	t <sub>SDAS</sub>	t <sub>IICcyc</sub> + 120	—	ns	
	Data input hold time	t <sub>SDAH</sub>	0	—	ns	
	SCL, SDA capacitive load	C <sub>b</sub>	—	550	pF	
Simple IIC (Standard-mode)	SDA input rise time	t <sub>Sr</sub>	—	1000	ns	
	SDA input fall time	t <sub>Sf</sub>	—	300	ns	
	SDA input spike pulse removal time	t <sub>SP</sub>	0	4 × t <sub>IICcyc</sub>	ns	
	Data input setup time	t <sub>SDAS</sub>	250	—	ns	
	Data input hold time	t <sub>SDAH</sub>	0	—	ns	
	SCL, SDA capacitive load	C <sub>b</sub>	—	400	pF	
Simple IIC (Fast-mode)	SCL, SDA input rise time	t <sub>Sr</sub>	20 + 0.1C <sub>b</sub>	300	ns	
	SCL, SDA input fall time	t <sub>Sf</sub>	20 + 0.1C <sub>b</sub>	300	ns	
	SCL, SDA input spike pulse removal time	t <sub>SP</sub>	0	4 × t <sub>IICcyc</sub>	ns	
	Data input setup time	t <sub>SDAS</sub>	100	—	ns	
	Data input hold time	t <sub>SDAH</sub>	0	—	ns	
	SCL, SDA capacitive load	C <sub>b</sub>	—	400	pF	

Note: • t<sub>IICcyc</sub>: RIIC internal reference clock (IICφ) Cycle, t<sub>Pcyc</sub>: PCLK cycle

Note 1. The value in parentheses is used when ICMR3.NF[1:0] are set to 11b while a digital filter is enabled with ICFER.NFE = 1.

Note 2. C<sub>b</sub> indicates the total capacity of the bus line.

**Figure 45.24 I/O Port Input Timing**

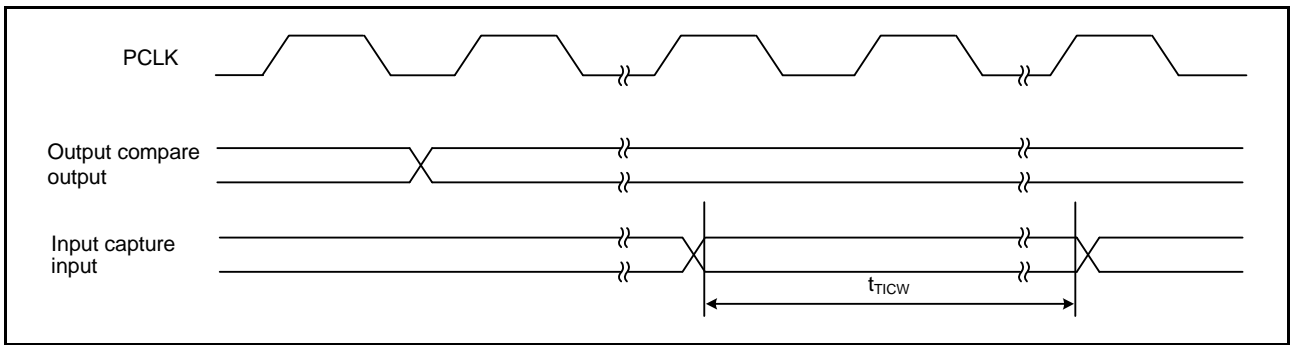


Figure 45.25 MTU Input/Output Timing

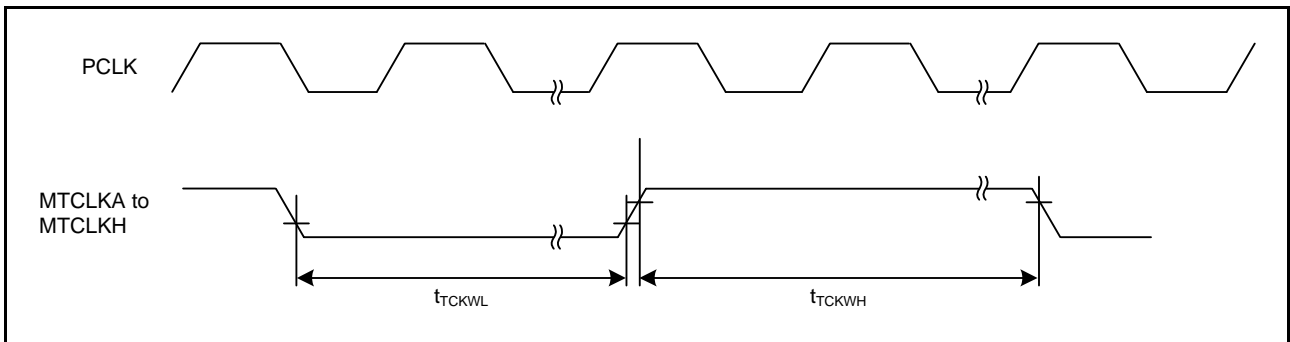


Figure 45.26 MTU Clock Input Timing

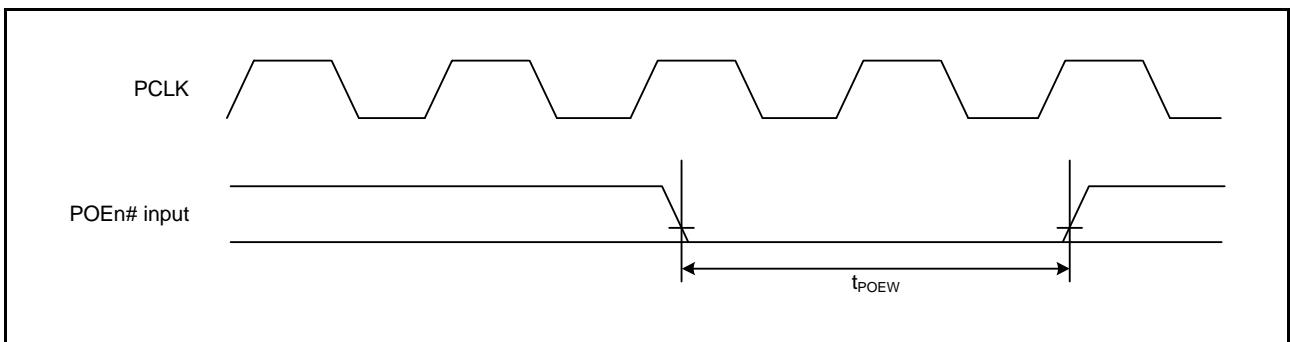


Figure 45.27 POE# Input Timing

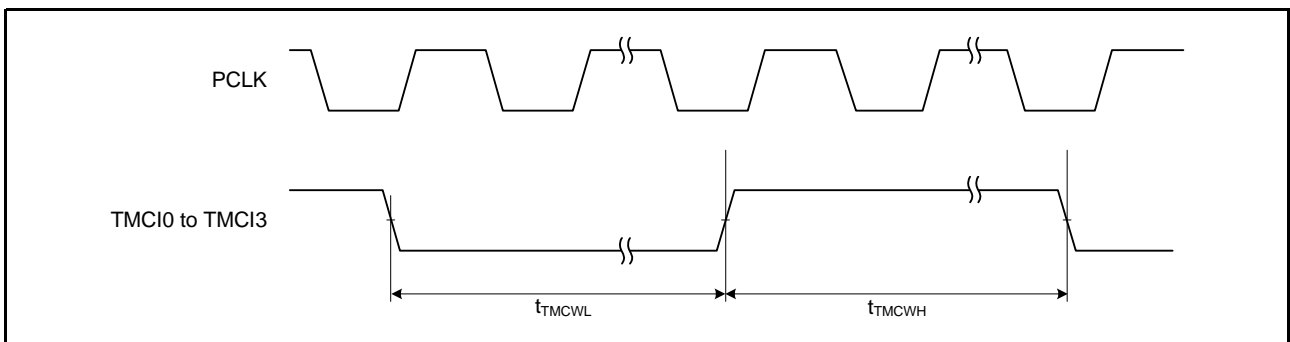


Figure 45.28 8-Bit Timer Clock Input Timing

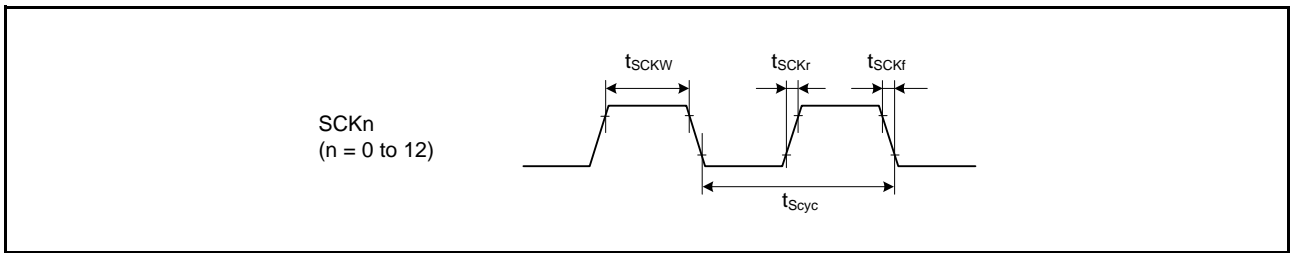


Figure 45.29 SCK Clock Input Timing

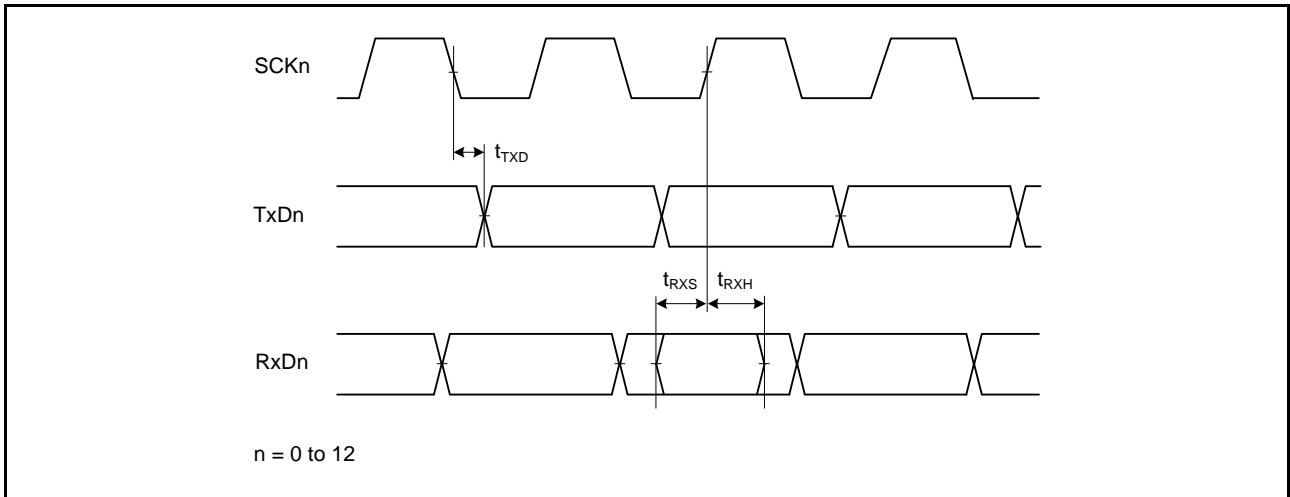


Figure 45.30 SCI Input/Output Timing: Clock Synchronous Mode

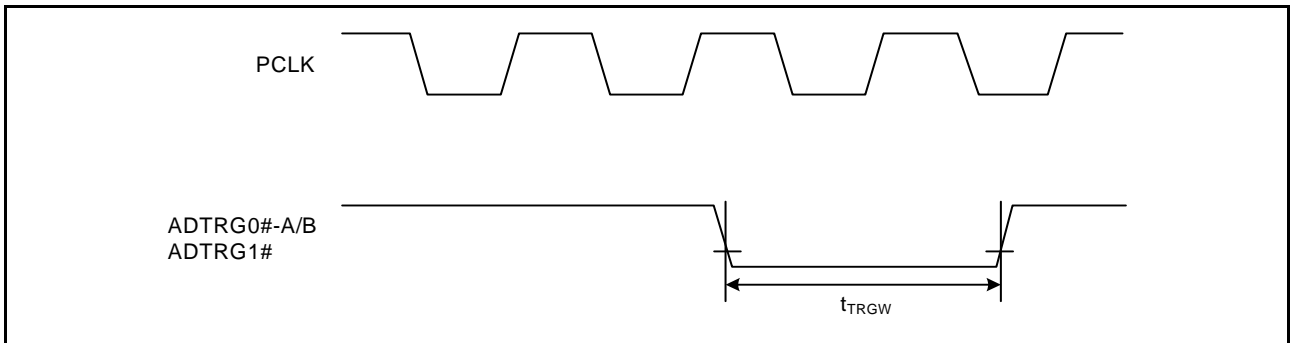


Figure 45.31 A/D Converter External Trigger Input Timing

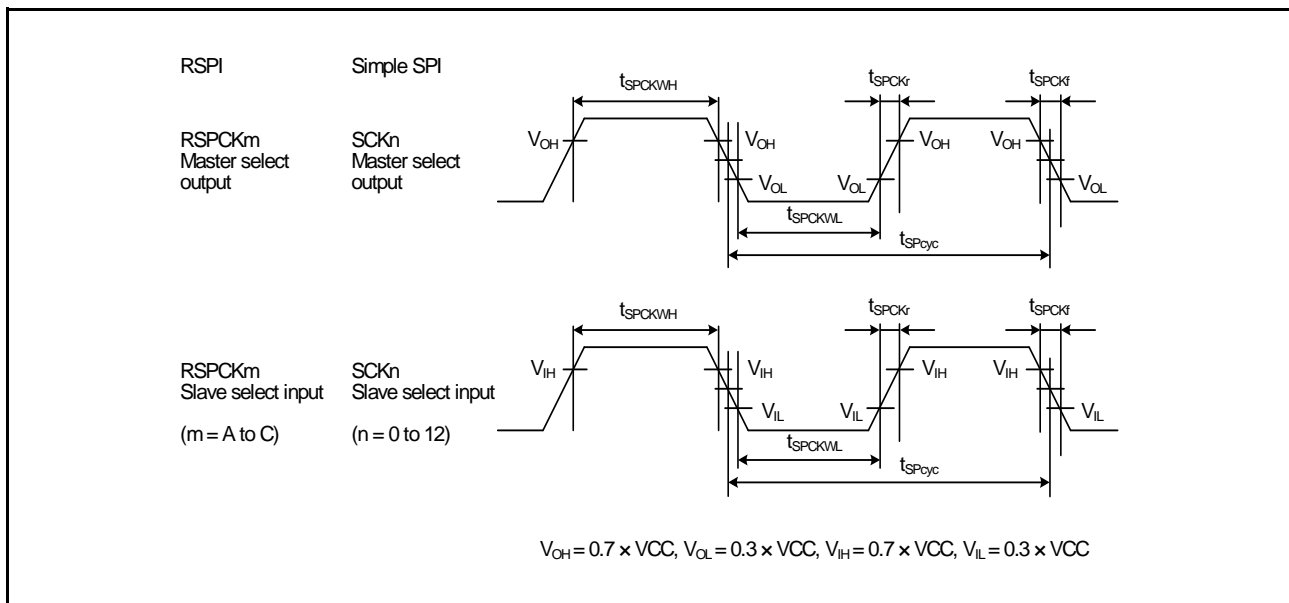


Figure 45.32 RSPI Clock Timing and Simple SPI Clock Timing

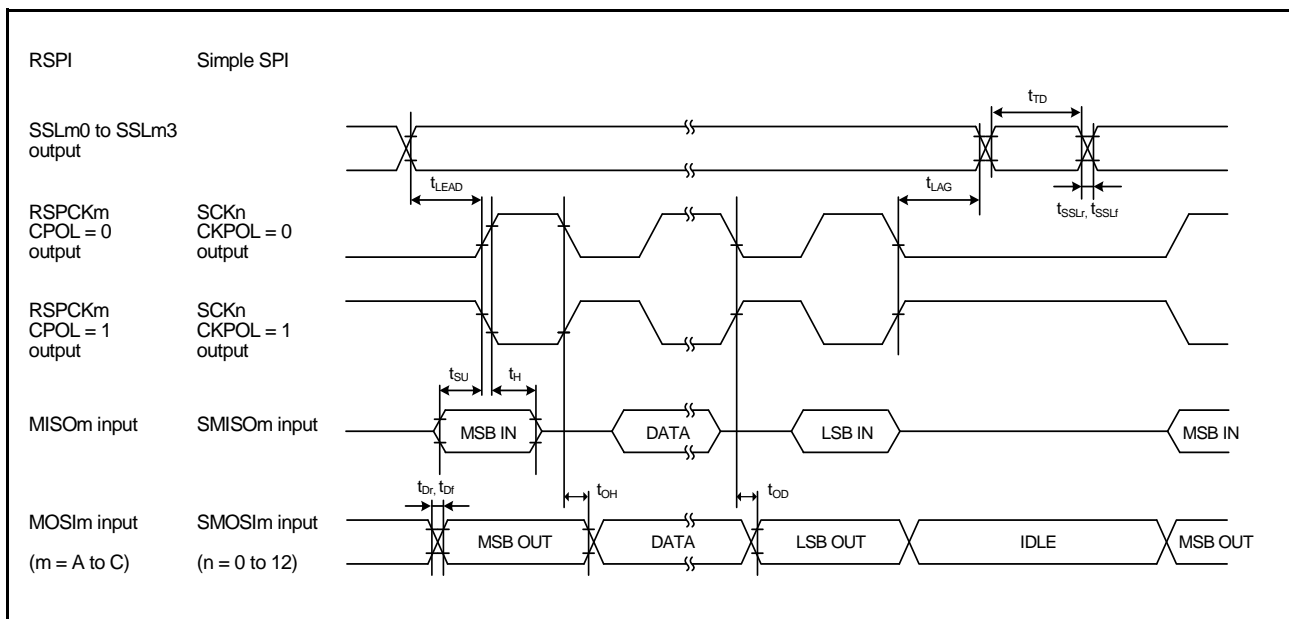


Figure 45.33 RSPI Timing (Master, CPHA = 0) and Simple SPI Timing (Master, CKPH = 1)

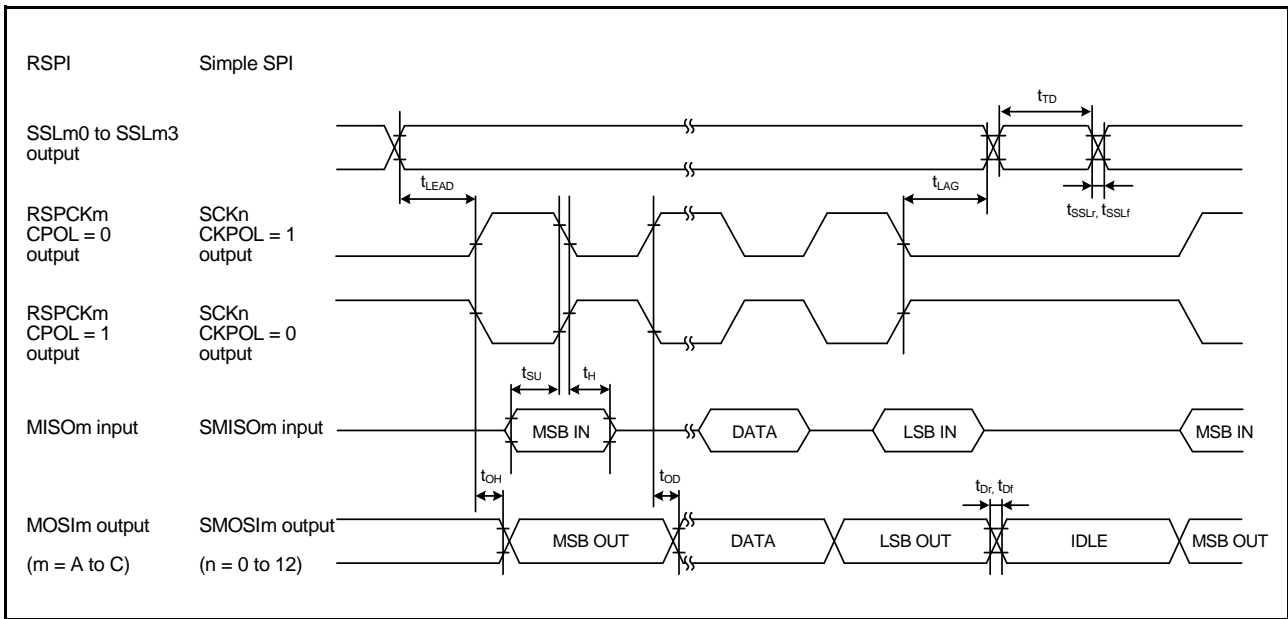


Figure 45.34 RSPI Timing (Master, CPHA = 1) and Simple SPI Timing (Master, CKPH = 0)

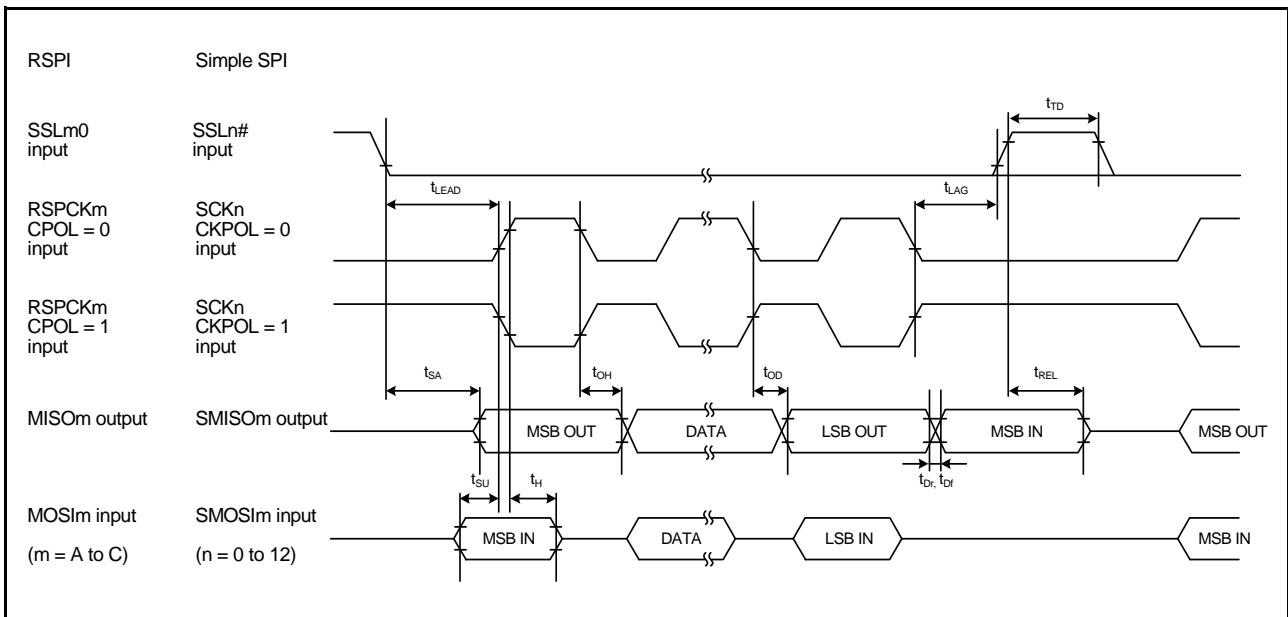


Figure 45.35 RSPI Timing (Slave, CPHA = 0) and Simple SPI Timing (Slave, CKPH = 1)

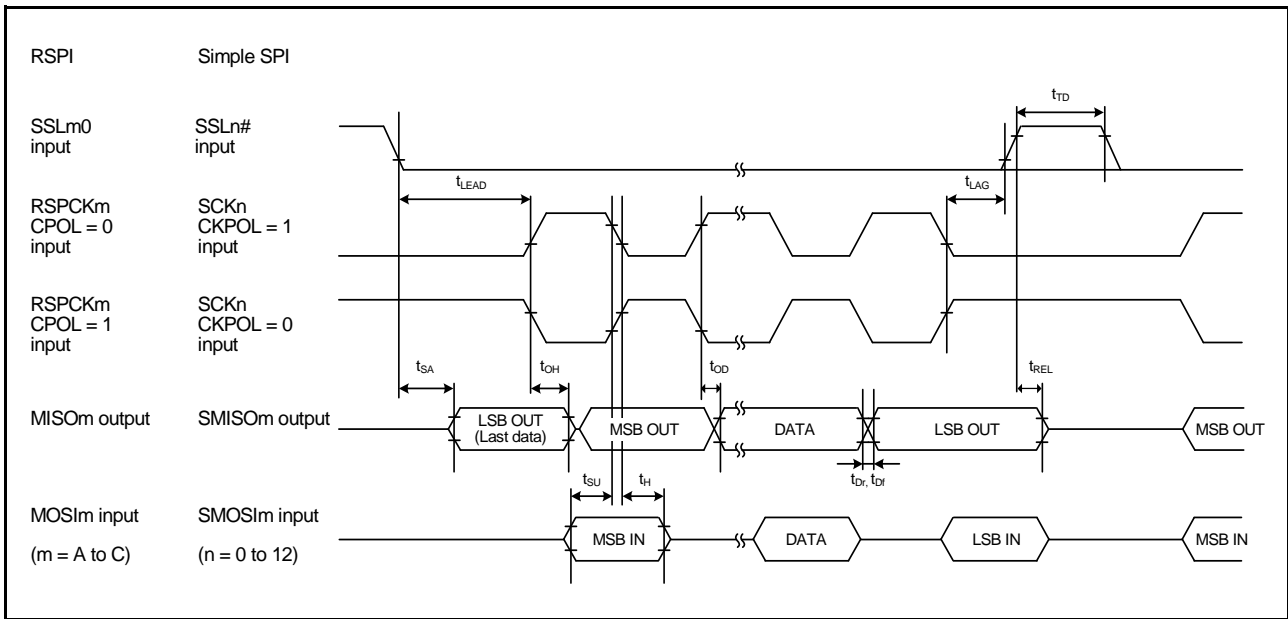


Figure 45.36 RSPI Timing (Slave, CPHA = 1) and Simple SPI Timing (Slave, CKPH = 0)

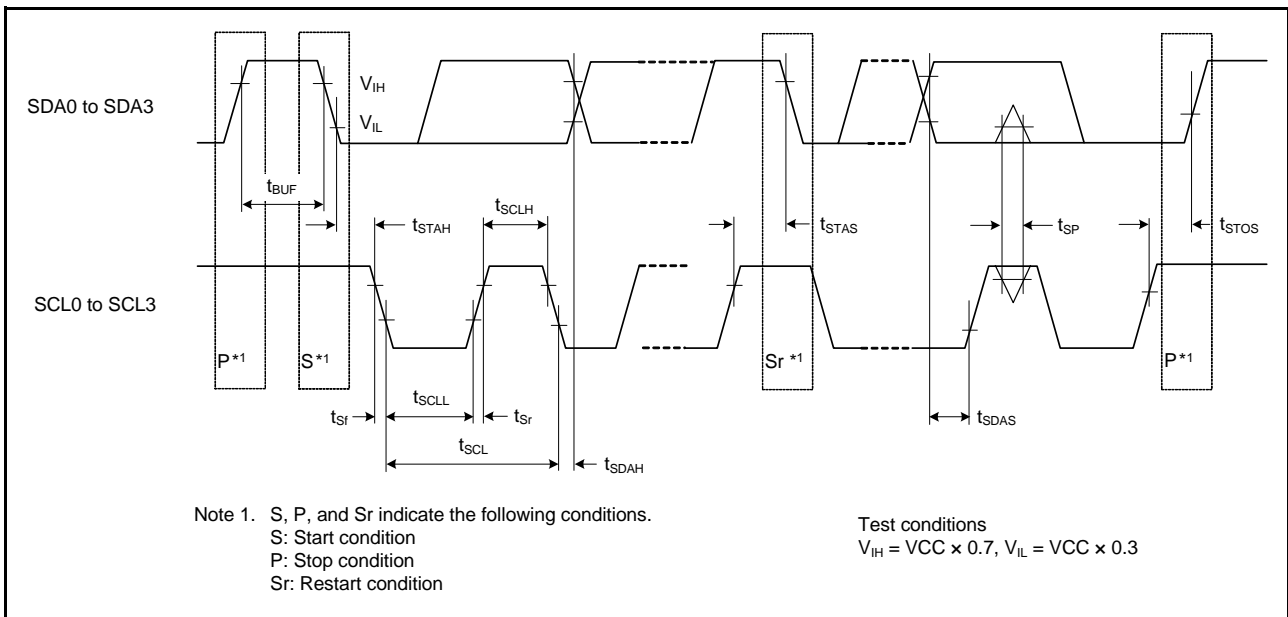


Figure 45.37 IIC Bus Interface Input/Output Timing and Simple IIC Bus Interface Input/Output Timing

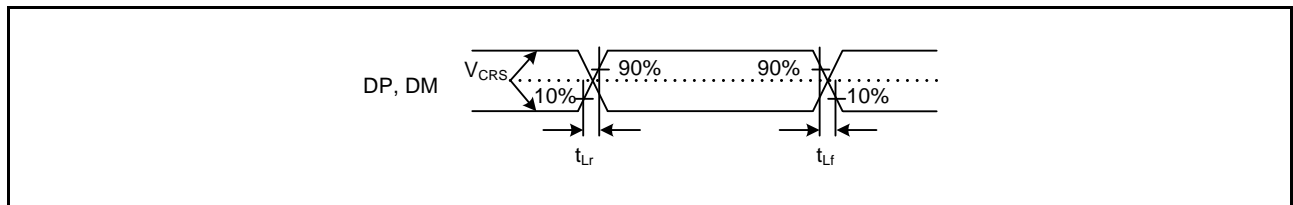


45.4 USB Characteristics

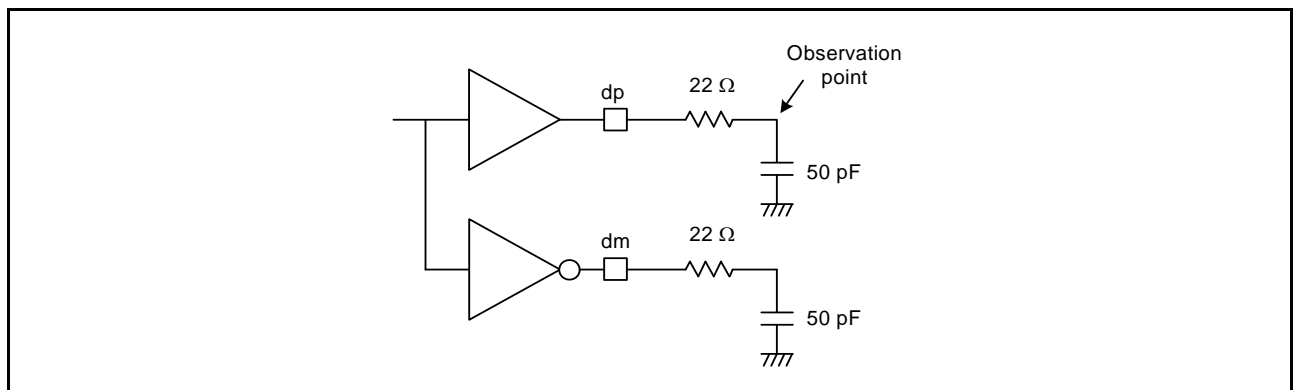
**Table 45.20 On-Chip USB Full-Speed Characteristics (DP and DM Pin Characteristics)**

Conditions:  $V_{CC} = AV_{CC0} = V_{REFH} = V_{CC\_USB} = 3.0$  to  $3.6$  V,  $V_{REFH0} = 3.0$  V to  $AV_{CC0}$   
 $V_{SS} = AV_{SS0} = V_{REFL}/V_{REFL0} = V_{SS\_USB} = 0$  V  
 $PCLK = 24$  to  $50$  MHz  
 $T_a = T_{opr}$   
 High drive output is selected by the drive capacity control register.

Item		Symbol	Min.	Max.	Unit	Test Conditions	
Input characteristics	Input high level voltage	$V_{IH}$	2.0	—	V		
	Input low level voltage	$V_{IL}$	—	0.8	V		
	Differential input sensitivity	$V_{DI}$	0.2	—	V	DP – DM	
	Differential common mode range	$V_{CM}$	0.8	2.5	V		
Output characteristics	Output high level voltage	$V_{OH}$	2.8	3.6	V	$I_{OH} = -200 \mu A$	
	Output low level voltage	$V_{OL}$	0.0	0.3	V	$I_{OL} = 2$ mA	
	Cross-over voltage	$V_{CRS}$	1.3	2.0	V		Figure 45.38
	Rise time	$t_{Lr}$	4	20	ns		
	Fall time	$t_{Lf}$	4	20	ns		
	Rise/fall time ratio	$t_{Lr} / t_{Lf}$	90	111.11	%	$t_{Lr} / t_{Lf}$	
	Output resistance	$Z_{DRV}$	28	44	$\Omega$	$R_s = 22 \Omega$ included	



**Figure 45.38 DP and DM Output Timing (Full-Speed)**



**Figure 45.39 Test Circuit (Full-Speed)**

## 45.5 A/D Conversion Characteristics

**Table 45.21 10-Bit A/D Conversion Characteristics**

Conditions:  $V_{CC} = AV_{CC0} = V_{REFH} = V_{CC\_USB} = 2.7$  to  $3.6$  V,  $V_{REFH0} = 2.7$  V to  $AV_{CC0}$   
 $V_{SS} = AV_{SS0} = V_{REFL}/V_{REFL0} = V_{SS\_USB} = 0$  V  
 $PCLK = 8$  to  $50$  MHz  
 $T_a = T_{opr}$

Item		Min.	Typ.	Max.	Unit	Test Conditions	
Resolution		10	10	10	Bit		
Conversion time*1 (Operation at $PCLK = 50$ MHz)	With 0.1- $\mu$ F external capacitor	When the capacitor is charged enough*2	3.0 (2.5)*3	—	—	$\mu$ s	Sampling in 125 states
	Without 0.1- $\mu$ F external capacitor	Permissible signal source impedance (max.) = 1.0 k $\Omega$ , $V_{CC} \geq 3.0$ V	1.5 (1.0)*3	—	—		Sampling in 50 states
		Permissible signal source impedance (max.) = 1.0 k $\Omega$ , $V_{CC} \geq 2.7$ V	3.5 (3.0)*3	—	—		Sampling in 150 states
		Permissible signal source impedance (max.) = 5.0 k $\Omega$ , $V_{CC} \geq 3.0$ V	2.0 (1.5)*3	—	—		Sampling in 75 states
		Permissible signal source impedance (max.) = 5.0 k $\Omega$ , $V_{CC} \geq 2.7$ V	4.0 (3.5)*3	—	—		Sampling in 175 states
Analog input capacitance		—	—	6.0	pF		
Offset error		—	$\pm 1.5$	$\pm 3.0$	LSB		
Full-scale error		—	$\pm 1.5$	$\pm 3.0$	LSB		
Quantization error		—	$\pm 0.5$	—	LSB		
Absolute accuracy		—	$\pm 1.5$	$\pm 3.0$	LSB		
DNL differential nonlinearity error		—	$\pm 0.5$	$\pm 1.0$	LSB		
INL integral nonlinearity error		—	$\pm 1.5$	$\pm 3.0$	LSB		

Note: • The above specification values apply when there is no access to the external bus during A/D conversion. If access proceeds during A/D conversion, values may not fall within the above ranges.

Note 1. The conversion time includes the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Note 2. The scanning is not supported.

Note 3. The value in parentheses indicates the sampling time.

**Table 45.22 12-Bit A/D Conversion Characteristics**

Conditions:  $V_{CC} = AVCC0 = VREFH = VCC\_USB = 2.7$  to  $3.6$  V,  $VREFH0 = 2.7$  V to  $AVCC0$   
 $VSS = AVSS0 = VREFL/VREFLO = VSS\_USB = 0$  V  
 $PCLK = 8$  to  $50$  MHz  
 $T_a = T_{opr}$

Item			Min.	Typ.	Max.	Unit	Test Conditions
Resolution			12	12	12	Bit	
Conversion time*1 (Operation at $PCLK = 50$ MHz)	AN0 to AN7	Permissible signal source impedance (max.) = $1.0$ k $\Omega$	$1.0 (0.4)^{*2}$	—	—	$\mu$ s	Sampling in 20 states
	Other channels	Permissible signal source impedance (max.) = $1.0$ k $\Omega$ , $AVCC \geq 3.0$ V	$2.0 (1.4)^{*2}$	—	—	$\mu$ s	Sampling in 70 states
		Permissible signal source impedance (max.) = $1.0$ k $\Omega$ , $AVCC \geq 2.7$ V	$5.6 (5.0)^{*2}$	—	—	$\mu$ s	Sampling in 250 states
Analog input capacitance			—	—	30	pF	
Offset error			—	$\pm 2.0$	$\pm 7.5$	LSB	
Full-scale error			—	$\pm 2.0$	$\pm 7.5$	LSB	
Quantization error			—	$\pm 0.5$	—	LSB	
Absolute accuracy			—	$\pm 2.5$	$\pm 8.0$	LSB	
DNL differential nonlinearity error			—	$\pm 2.0$	$\pm 4.0$	LSB	
INL integral nonlinearity error			—	$\pm 2.0$	$\pm 4.0$	LSB	

Note: • The above specification values apply when there is no access to the external bus during A/D conversion. If access proceeds during A/D conversion, values may not fall within the above ranges.

Note 1. The conversion time includes the sampling time and the comparison time. As the test conditions, the number of sampling states is indicated.

Note 2. The value in parentheses indicates the sampling time.

**Table 45.23 A/D Internal Reference Voltage Characteristics**

Conditions:  $V_{CC} = AVCC0 = VREFH = VCC\_USB = 2.7$  to  $3.6$  V,  $VREFH0 = 2.7$  V to  $AVCC0$   
 $VSS = AVSS0 = VREFL/VREFLO = VSS\_USB = 0$  V  
 $PCLK = 8$  to  $50$  MHz  
 $T_a = T_{opr}$

Item	Min.	Typ.	Max.	Unit	Test Conditions
A/D Internal reference voltage	1.45	1.50	1.55	V	

## 45.6 D/A Conversion Characteristics

**Table 45.24 D/A Conversion Characteristics**

Conditions:  $VCC = AVCC0 = VREFH = VCC\_USB = 2.7$  to  $3.6$  V,  $VREFH0 = 2.7$  V to  $VCC$   
 $VSS = AVSS0 = VREFL/VREFL0 = VSS\_USB = 0$  V  
 $T_a = T_{opr}$

Item	Min.	Typ.	Max.	Unit	Test Conditions
Resolution	10	10	10	Bit	
Conversion time	—	—	3.0	$\mu$ s	20-pF capacitive load
Absolute accuracy	—	$\pm 2.0$	$\pm 4.0$	LSB	2-M $\Omega$ resistive load
	—	—	$\pm 3.0$	LSB	4-M $\Omega$ resistive load
	—	—	$\pm 2.0$	LSB	10-M $\Omega$ resistive load
RO output resistance	—	3.6	—	k $\Omega$	

## 45.7 Temperature Sensor Characteristics

**Table 45.25 Temperature Sensor Characteristics**

Conditions:  $VCC = AVCC0 = VREFH = VCC\_USB = 2.7$  to  $3.6$  V,  $VREFH0 = 2.7$  V to  $VCC$   
 $VSS = AVSS0 = VREFL/VREFL0 = VSS\_USB = 0$  V  
 $T_a = T_{opr}$

Item	Min.	Typ.	Max.	Unit	Test Conditions
Relative accuracy	—	$\pm 1$	—	$^{\circ}$ C	
Temperature slope	—	4.1	—	mV/ $^{\circ}$ C	
Output voltage (@25 $^{\circ}$ C)	—	1.26	—	V	
Temperature sensor start time	—	—	30	$\mu$ s	
Sampling time	—	—	5	$\mu$ s	

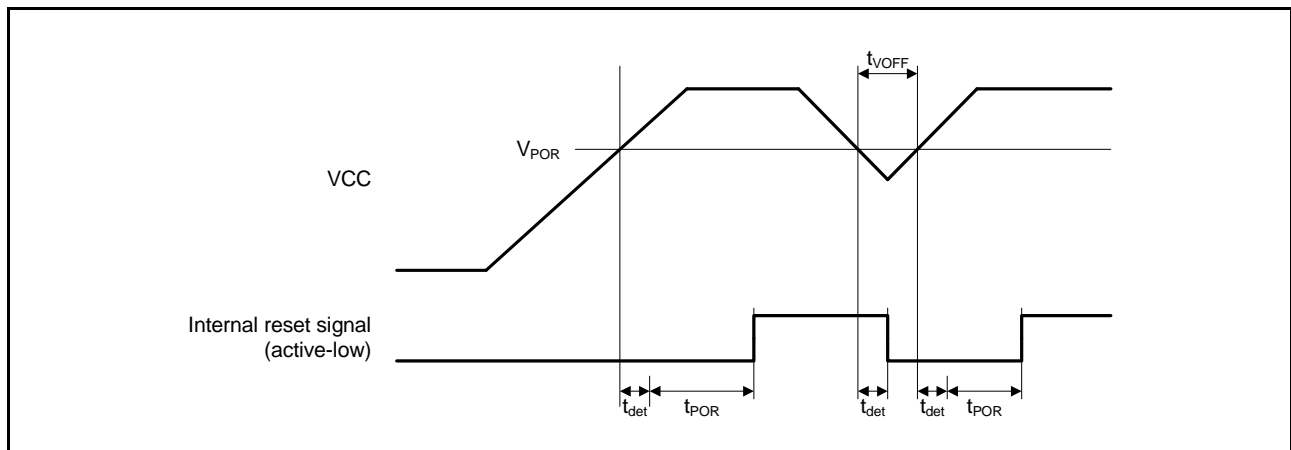
### 45.8 Power-on Reset Circuit and Voltage Detection Circuit Characteristics

**Table 45.26 Power-on Reset Circuit and Voltage Detection Circuit Characteristics**

Conditions:  $V_{CC} = AV_{CC0} = V_{REFH} = V_{CC\_USB} = V_{BATT} = 2.7$  to  $3.6$  V,  $V_{REFH0} = 2.7$  V to  $AV_{CC0}$   
 $V_{SS} = AV_{SS0} = V_{REFL}/V_{REFL0} = V_{SS\_USB} = 0$  V  
 $T_a = T_{opr}$

Item		Symbol	Min.	Typ.	Max.	Unit	Test Conditions	
Voltage detection level	Power-on reset (POR)	Low power consumption function disabled	$V_{POR}$	2.5	2.6	2.7	V	Figure 45.40
		Low power consumption function enabled		2.0	2.35	2.7		
	Voltage detection circuit (LVD0)		$V_{det0}$	2.7	2.80	2.9		Figure 45.41
	Voltage detection circuit (LVD1)		$V_{det1\_A}$	2.75	2.95	3.15		
	Voltage detection circuit (LVD2)		$V_{det2\_A}$	2.75	2.95	3.15		
Internal reset time	Power-on reset time		$t_{POR}$	—	4.6	—	ms	Figure 45.40
	LVD0 reset time		$t_{LVD0}$	—	4.6	—		Figure 45.41
	LVD1 reset time		$t_{LVD1}$	—	0.9	—		Figure 45.42
	LVD2 reset time		$t_{LVD2}$	—	0.9	—		Figure 45.43
Minimum VCC down time			$t_{V_{OFF}}$	200	—	—	$\mu$ s	Figure 45.40 and Figure 45.41
Response delay time			$t_{det}$	—	—	200	$\mu$ s	Figure 45.40 to Figure 45.43
LVD operation stabilization time (after LVD is enabled)			$T_{d(E-A)}$	—	—	3	$\mu$ s	Figure 45.42 and Figure 45.43
Hysteresis width (LVD1 and LVD2)			$V_{LVH}$	—	80	—	mV	

Note: • The minimum VCC down time indicates the time when VCC is below the minimum value of voltage detection levels  $V_{POR}$ ,  $V_{det1}$ , and  $V_{det2}$  for the POR/ LVD.



**Figure 45.40 Power-on Reset Timing**

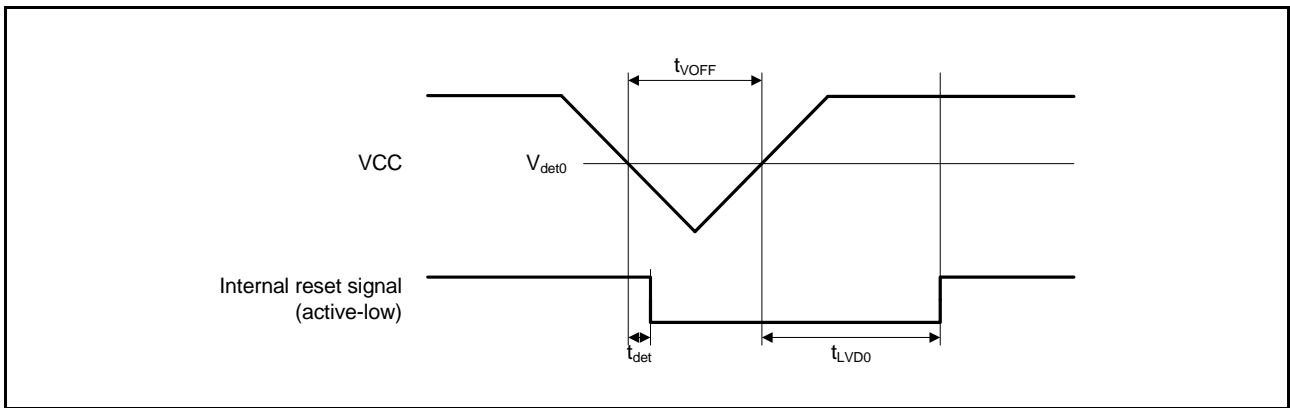


Figure 45.41 Voltage Detection Circuit Timing ( $V_{det0}$ )

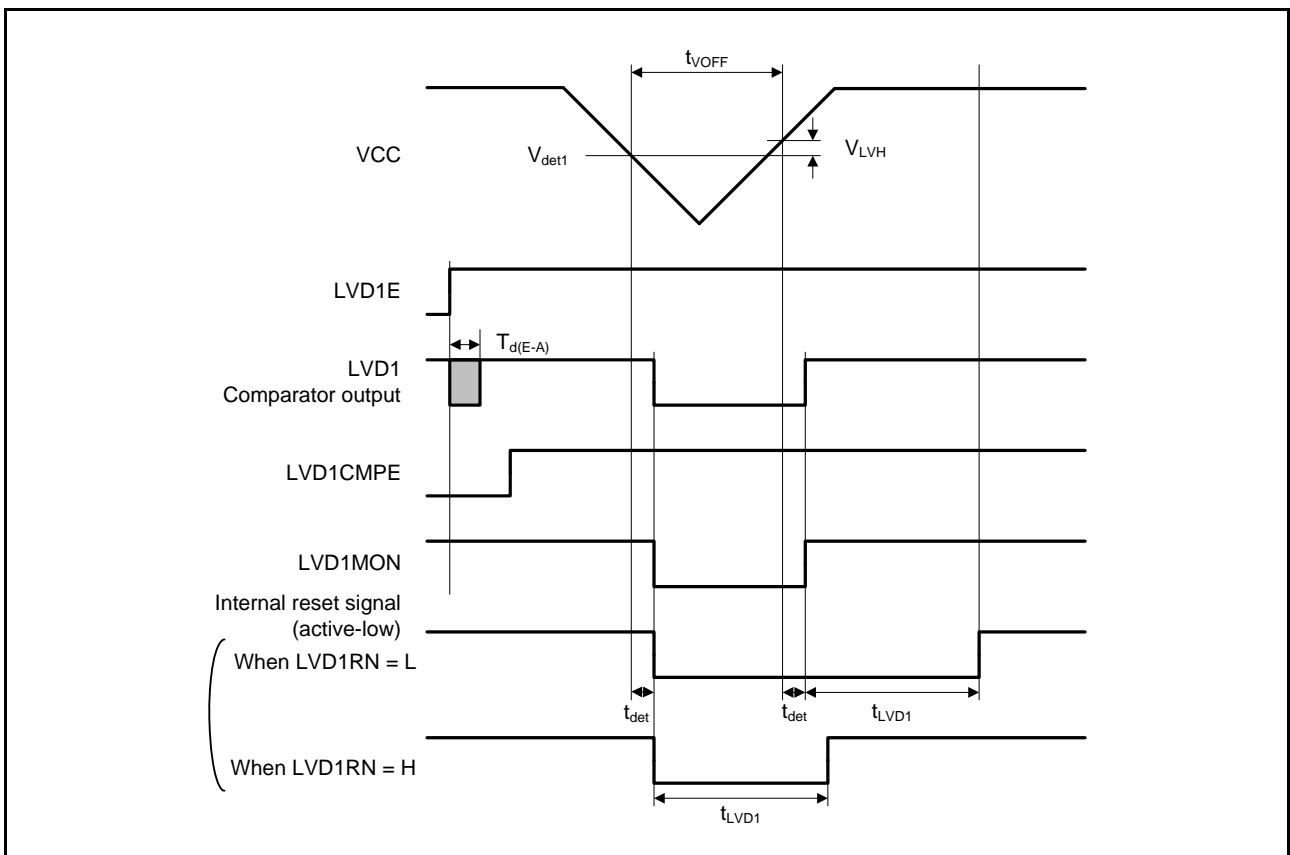


Figure 45.42 Voltage Detection Circuit Timing ( $V_{det1}$ )

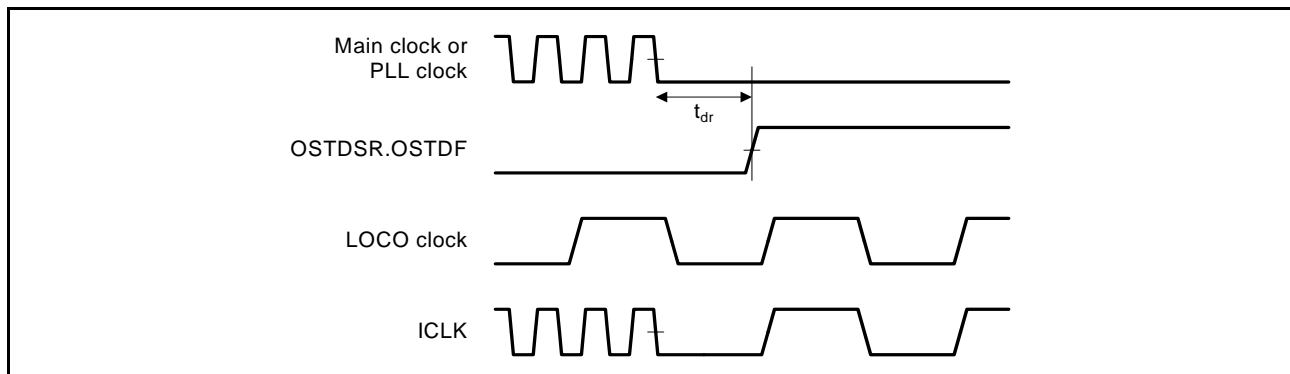


### 45.9 Oscillation Stop Detection Timing

**Table 45.27 Oscillation Stop Detection Circuit Characteristics**

Conditions:  $V_{CC} = AVCC0 = VREFH = VCC\_USB = V_{BATT} = 2.7$  to  $3.6$  V,  $VREFH0 = 2.7$  V to  $AVCC0$   
 $VSS = AVSS0 = VREFL/VREFLO = VSS\_USB = 0$  V  
 $T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Detection time	$t_{dr}$	—	—	1	ms	Figure 45.44



**Figure 45.44 Oscillation Stop Detection Timing**

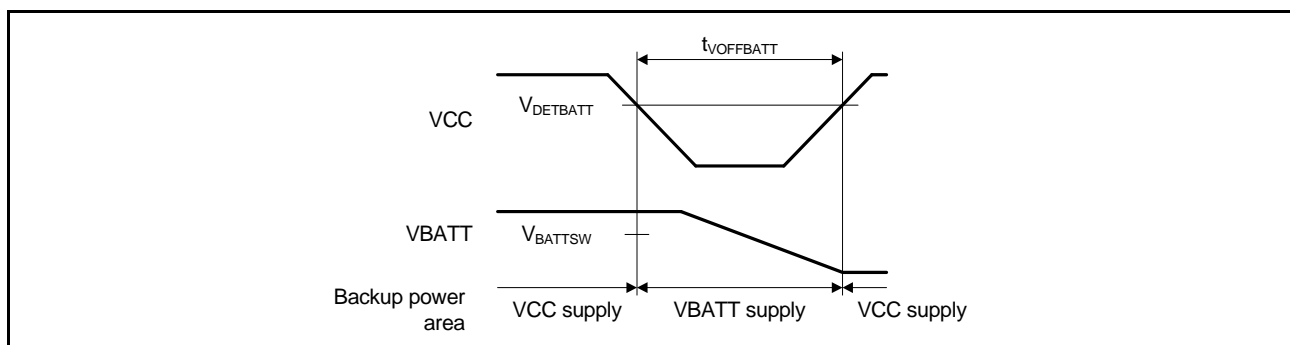
### 45.10 Battery Backup Function Characteristics

**Table 45.28 Battery Backup Function Characteristics**

Conditions:  $V_{CC} = AVCC0 = VREFH = VCC\_USB = 2.7$  to  $3.6$  V,  $VREFH0 = 2.7$  V to  $AVCC0$ ,  $V_{BATT} = 2.3$  to  $3.6$  V  
 $VSS = AVSS0 = VREFL/VREFLO = VSS\_USB = 0$  V  
 $T_a = T_{opr}$

Item	Symbol	Min.	Typ.	Max.	Unit	Test Conditions
Voltage level for switching to battery backup	$V_{DETBATT}$	2.50	2.60	2.70	V	Figure 45.45
Lower-limit VBATT voltage for power supply switching due to VCC voltage drop	$V_{BATTSW}$	2.70	—	—		
VCC-off period for starting power supply switching	$t_{VOFFBATT}$	200	—	—	$\mu s$	

Note: • The VCC-off period for starting power supply switching indicates the period in which VCC is below the minimum value of the voltage level for switching to battery backup ( $V_{DETBATT}$ ).



**Figure 45.45 Battery Backup Function Characteristics**



## 45.11 ROM (Flash Memory for Code Storage) Characteristics

**Table 45.29 ROM (Flash Memory for Code Storage) Characteristics**

Conditions: VCC = AVCC0 = VREFH = VCC\_USB = 2.7 to 3.6 V, VREFH0 = 2.7 V to AVCC0  
VSS = AVSS0 = VREFL/VREFL0 = VSS\_USB = 0 V  
Temperature range for the programming/erasure operation: T<sub>a</sub> = T<sub>opr</sub>

Item		Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 50 MHz			Unit
			Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time N <sub>PEC</sub> ≤ 100 hours	128 bytes	t <sub>P128</sub>	—	2.8	28	—	1	10	ms
	4 Kbytes	t <sub>P4K</sub>	—	63	140	—	23	50	ms
	16 Kbytes	t <sub>P16K</sub>	—	252	560	—	90	200	ms
Programming time N <sub>PEC</sub> > 100 hours	128 bytes	t <sub>P128</sub>	—	3.4	33.6	—	1.2	12	ms
	4 Kbytes	t <sub>P4K</sub>	—	75.6	168	—	27.6	60	ms
	16 Kbytes	t <sub>P16K</sub>	—	302.4	672	—	108	240	ms
Erasure time N <sub>PEC</sub> ≤ 100 hours	4 Kbytes	t <sub>E4K</sub>	—	50	120	—	25	60	ms
	16 Kbytes	t <sub>E16K</sub>	—	200	480	—	100	240	ms
Erasure time N <sub>PEC</sub> > 100 hours	4 Kbytes	t <sub>E4K</sub>	—	60	144	—	30	72	ms
	16 Kbytes	t <sub>E16K</sub>	—	240	576	—	120	288	ms
Reprogram/erase cycle*1	N <sub>PEC</sub>	1000*2	—	—	1000*2	—	—	Times	
Suspend delay time during programming	t <sub>SPD</sub>	—	—	400	—	—	120	μs	
First suspend delay time during erasing (in suspend priority mode)	t <sub>SESD1</sub>	—	—	300	—	—	120	μs	
Second suspend delay time during erasing (in suspend priority mode)	t <sub>SESD2</sub>	—	—	1.7	—	—	1.7	ms	
Suspend delay time during erasing (in erasure priority mode)	t <sub>SEED</sub>	—	—	1.7	—	—	1.7	ms	
Data hold time*3	t <sub>DRP</sub>	10	—	—	10	—	—	Year	
FCU reset time	t <sub>FCUR</sub>	35	—	—	35	—	—	μs	

Note 1. Definition of reprogram/erase cycle:

The reprogram/erase cycle is the number of erasing for each block. When the reprogram/erase cycle is n times (n = 1000), erasing can be performed n times for each block. For instance, when 256-byte programming is performed 16 times for different addresses in 4-Kbyte block and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address for several times as one erasing is not enabled (overwriting is prohibited).

Note 2. This indicates the minimum number that guarantees the characteristics after reprogramming. (The guaranteed value is in the range from one to the minimum number.)

Note 3. This indicates the characteristic when reprogram is performed within the specification range including the minimum number.

45.12 E<sup>2</sup> Flash Characteristics**Table 45.30 E<sup>2</sup> Flash Characteristics**

Conditions: VCC = AVCC0 = VREFH = VCC\_USB = 2.7 to 3.6 V, VREFH0 = 2.7 V to AVCC0  
 VSS = AVSS0 = VREFL/VREFL0 = VSS\_USB = 0 V  
 Temperature range for the programming/erasure operation: T<sub>a</sub> = T<sub>opr</sub>

Item	Symbol	FCLK = 4 MHz			20 MHz ≤ FCLK ≤ 50 MHz			Unit
		Min.	Typ.	Max.	Min.	Typ.	Max.	
Programming time N <sub>PEC</sub> ≤ 100 hours	2 bytes t <sub>DP2</sub>	—	0.7	6	—	0.25	2	ms
Programming time N <sub>PEC</sub> > 100 hours	2 bytes t <sub>DP2</sub>	—	0.7	6	—	0.25	2	ms
Erasure time N <sub>PEC</sub> ≤ 100 hours	32 bytes t <sub>DE32</sub>	—	4	40	—	2	20	ms
Erasure time N <sub>PEC</sub> > 100 hours	32 bytes t <sub>DE32</sub>	—	7	40	—	4	20	ms
Blank check time	2 bytes t <sub>DBC2</sub>	—	—	100	—	—	30	μs
Reprogram/erase cycle* <sup>1</sup>	N <sub>DPEC</sub>	100000* <sup>2</sup>	—	—	100000* <sup>2</sup>	—	—	Times
Suspend delay time during programming	t <sub>DSPD</sub>	—	—	250	—	—	120	μs
First suspend delay time during erasing (in suspend priority mode)	t <sub>DSESD1</sub>	—	—	250	—	—	120	μs
Second suspend delay time during erasing (in suspend priority mode)	t <sub>DSESD2</sub>	—	—	500	—	—	300	μs
Suspend delay time during erasing (in erasure priority mode)	t <sub>DSEED</sub>	—	—	500	—	—	300	μs
Data hold time* <sup>3</sup>	t <sub>DDRP</sub>	10	—	—	10	—	—	Year

Note 1. Definition of reprogram/erase cycle:

The reprogram/erase cycle is the number of erasing for each block. When the reprogram/erase cycle is n times (n = 100000), erasing can be performed n times for each block. For instance, when 128-byte programming is performed 16 times for different addresses in 2-Kbyte block and then the entire block is erased, the reprogram/erase cycle is counted as one. However, programming the same address for several times as one erasing is not enabled (overwriting is prohibited).

Note 2. This indicates the minimum number that guarantees the characteristics after reprogramming. (The guaranteed value is in the range from one to the minimum number.)

Note 3. This indicates the characteristic when reprogram is performed within the specification range including the minimum number.

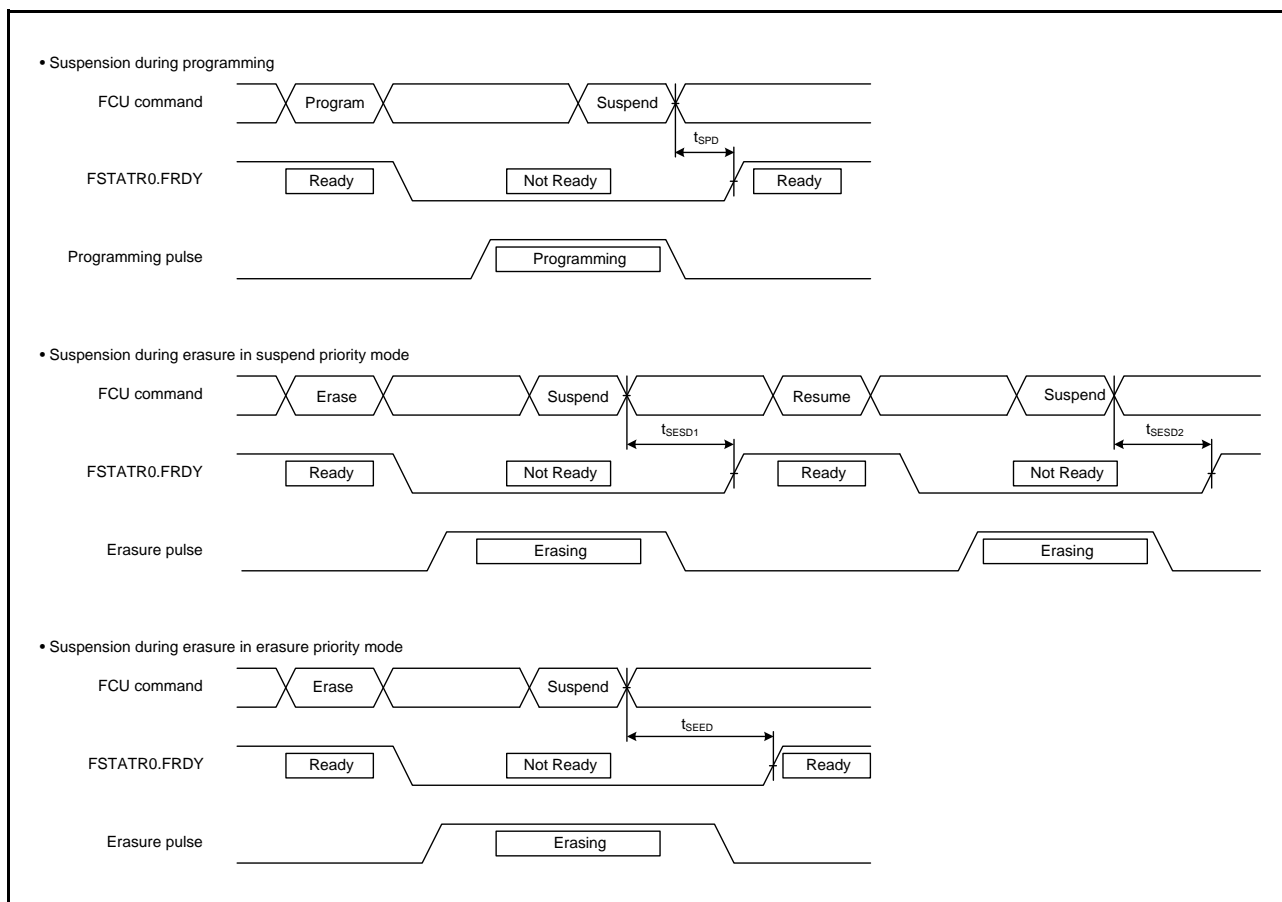


Figure 45.46 Flash Memory Program/Erase Suspend Timing

## Appendix 1. Port States in Each Processing Mode

Table 1.1 Port States in Each Processing State (1/3)

Port Name Pin Name	Operating Mode According to Registers Setting	Reset	Software Standby Mode		Deep Software Standby Mode IOKEEP = 1/0	After Deep Software Standby Mode is Canceled (Return of Start-up Mode)	
			OPE = 1	OPE = 0		IOKEEP = 1*1	IOKEEP = 0
P00/IRQ8, P01/IRQ9, P02/IRQ10	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P03/DA0/IRQ11	All	Hi-Z	[DAOE0 = 1] DA output retained [DAOE0 = 0] Keep-O*2		[DAOE0 = 1] Hi-Z [DAOE0 = 0] Keep	[DAOE0 = 1] Hi-Z [DAOE0 = 0] Keep	Hi-Z
P05/DA1/IRQ13	All	Hi-Z	[DAOE1 = 1] DA output retained [DAOE1 = 0] Keep-O*2		[DAOE1 = 1] Hi-Z [DAOE1 = 0] Keep	[DAOE1 = 1] Hi-Z [DAOE1 = 0] Keep	Hi-Z
P07/IRQ15	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P10/IRQ0, P11/IRQ1, P12/IRQ2, P13/IRQ3, P14/IRQ4, P15/IRQ5/CRX1, P16/IRQ6/SCL2, P17/IRQ7/ SDA2	All	Hi-Z	Keep-O*2		Keep-O*3	Keep	Hi-Z
P20/IRQ8, P21/IRQ9, P22, P23	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P24/CS4#, P25/CS5#, P26/CS6#, P27/CS7#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[CS output] H [Other than the above] Keep-O	[CS output] Hi-Z [Other than the above] Keep-O			
P30/IRQ0, P31/IRQ1, P32/IRQ2, P33/IRQ3, P34/IRQ4, P35/NMI	All	Hi-Z	Keep-O*2		Keep-O*3	Keep	Hi-Z
P36, P37	All	Hi-Z	Keep-O		Keep-O	Keep	Hi-Z
P40/IRQ8, P41/IRQ9, P42/IRQ10, P43/IRQ11, P44/IRQ12, P45/IRQ13, P46/IRQ14, P47/IRQ15	All	Hi-Z	Keep-O*2		Keep-O*3	Keep	Hi-Z
P50/WR0#/WR#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[WR0#/WR# output] H	[WR0#/WR# output] Hi-Z			
P51/WR1#/BC1#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[WR1#/BC1# output] H [Other than the above] Keep-O	[WR1#/BC1# output] Hi-Z [Other than the above] Keep-O			
P52/RD#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[RD# output] H	[RD# output] Hi-Z			
P53/BCLK	All	Hi-Z	[Clock output] H [Other than the above] Keep-O		Keep	Keep	Hi-Z
P54/ALE/TRDATA2	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[ALE output] L [Other than the above] Keep-O	[ALE output] Hi-Z [Other than the above] Keep-O			
P55/TRDATA3/IRQ10	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
P56/WR2#/BC2#	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[WR2#/BC2# output] H [Other than the above] Keep-O	[WR2#/BC2# output] Hi-Z [Other than the above] Keep-O			

**Table 1.1 Port States in Each Processing State (2/3)**

Port Name Pin Name	Operating Mode According to Registers Setting		Reset	Software Standby Mode		Deep Software Standby Mode IOKEEP = 1/0	After Deep Software Standby Mode is Canceled (Return of Start-up Mode)	
				OPE = 1	OPE = 0		IOKEEP = 1*1	IOKEEP = 0
P57/WR3#/BC3#	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)			[WR3#/BC3# output] H [Other than the above] Keep-O	[WR3#/BC3# output] Hi-Z [Other than the above] Keep-O			
P60/CS0#, P61/CS1#, P62/CS2#, P63/CS3#, P64/CS4#, P65/CS5#, P66/CS6#	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)			[CS output] H [Other than the above] Keep-O	[CS output] Hi-Z [Other than the above] Keep-O			
P67/CS7#/IRQ15	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)			[CS output] H [Other than the above] Keep-O*2	[CS output] Hi-Z [Other than the above] Keep-O*2			
P70	All		Hi-Z	Keep-O		Keep	Keep	Hi-Z
P71/CS1#, P72/CS2#, P73/CS3#, P74/CS4#, P75/CS5#, P76/CS6#, P77/CS7#	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)			[CS output] H [Other than the above] Keep-O	[CS output] Hi-Z [Other than the above] Keep-O			
P80/TRDATA0, P81/TRDATA1, P82 to P87	All		Hi-Z	Keep-O		Keep	Keep	Hi-Z
Port 9	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)			[Address output] Address output retained [Data output] HiZ [Other than the above] Keep-O	[Address output] Hi-Z [Data output] Hi-Z [Other than the above] Keep-O			
PA0, PA1/IRQ11, PA2, PA3/IRQ6, PA4/IRQ5, PA5, PA6, PA7	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O*2		Keep-O*3	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)			[Address output] Address output retained [Other than the above] Keep-O*2	[Address output] Hi-Z [Other than the above] Keep-O*2			
PB0/IRQ12, PB1/IRQ4, PB2 to PB7	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O*2		Keep-O*3	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)			[Address output] Address output retained [Other than the above] Keep-O*2	[Address output] Hi-Z [Other than the above] Keep-O*2			
PC0/IRQ14, PC1/IRQ12, PC2, PC3, PC4/CS3#, PC5/CS2#, PC6/CS1#/IRQ13, PC7/CS0#/IRQ14	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)			[Address output] Address output retained [CS output] H [Other than the above] Keep-O*2	[Address output] Hi-Z [CS output] Hi-Z [Other than the above] Keep-O*2			
PD0/IRQ0, PD1/IRQ1, PD2/IRQ2, PD3/IRQ3, PD4/IRQ4, PD5/IRQ5, PD6/IRQ6, PD7/IRQ7	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)			[Data output] HiZ				
PE0, PE1, PE2/IRQ7, PE3, PE4, PE5/IRQ5, PE6/IRQ6, PE7/IRQ7	Single-chip mode (EXBE = 0)		Hi-Z	Keep-O*2		Keep-O*3	Keep	Hi-Z
	On-chip ROM enabled/ disabled extended (EXBE = 1)	8 bits in width of bus		Keep-O*2				
		16 bits in width of bus		[Data output] Hi-Z				

**Table 1.1 Port States in Each Processing State (3/3)**

Port Name Pin Name	Operating Mode According to Registers Setting	Reset	Software Standby Mode		Deep Software Standby Mode IOKEEP = 1/0	After Deep Software Standby Mode is Canceled (Return of Start-up Mode)	
			OPE = 1	OPE = 0		IOKEEP = 1*1	IOKEEP = 0
PF0, PF1, PF2, PF3, PF4, PF5/IRQ4	All	Hi-Z	Keep-O*2		Keep	Keep	Hi-Z
PG0, PG1, PG2/TRDATA0, PG3/TRDATA1, PG4, PG5, PG6/TRDATA2, PG7/TRDATA3	Single-chip mode (EXBE = 0)	Hi-Z	Keep-O		Keep	Keep	Hi-Z
	On-chip ROM enabled/disabled extended (EXBE = 1)		[Data output] Hi-Z [Other than the above] Keep-O				
Port H	All	Hi-Z	Keep-O		Keep	Keep	Hi-Z
Port J	All	Hi-Z	Keep-O		Keep	Keep	Hi-Z
Port K	All	Hi-Z	Keep-O		Keep	Keep	Hi-Z
Port L	All	Hi-Z	Keep-O		Keep	Keep	Hi-Z
USB0_DM	All	Hi-Z	Keep-O*4		Hi-Z*3	Hi-Z	
USB0_DP	All	Hi-Z	Keep-O*4		Hi-Z*3	Hi-Z	

H: High-level  
 L: Low-level  
 Keep-O: Output pins retain their previous values, and input pins become high-impedance.  
 Keep: Pin states are retained during periods in software standby mode.  
 Hi-Z: High-impedance

- Note 1. Retains the I/O port state until the DPSBYCR.IOKEEP bit is cleared to 0.
- Note 2. Input is enabled if the pin is specified as the software standby cancelling source while it is used as an external interrupt pin.
- Note 3. Input is enabled if the pin is specified as the deep software standby cancelling source.
- Note 4. Input is enabled while the pin is used as an input 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.

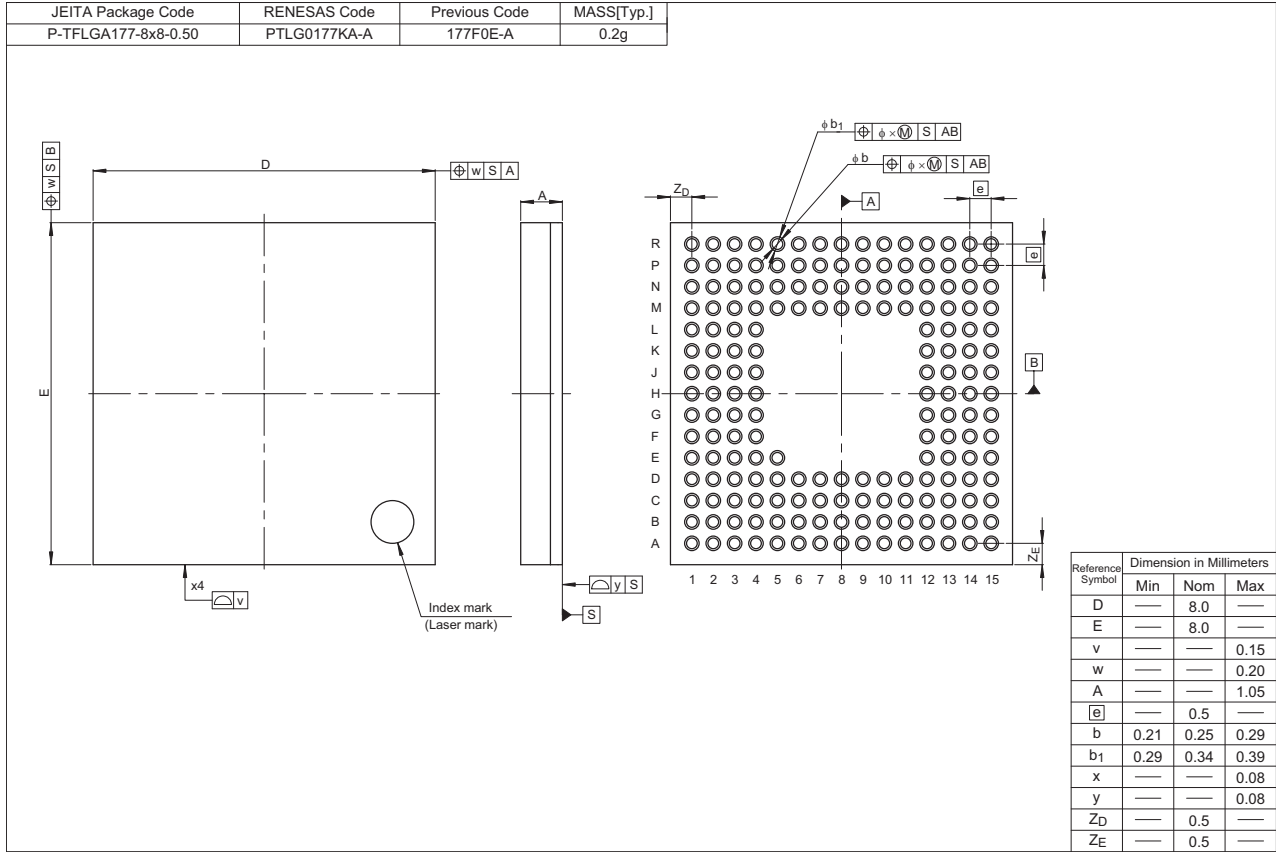


Figure A 177-Pin TFLGA (PTLG0177KA-A)

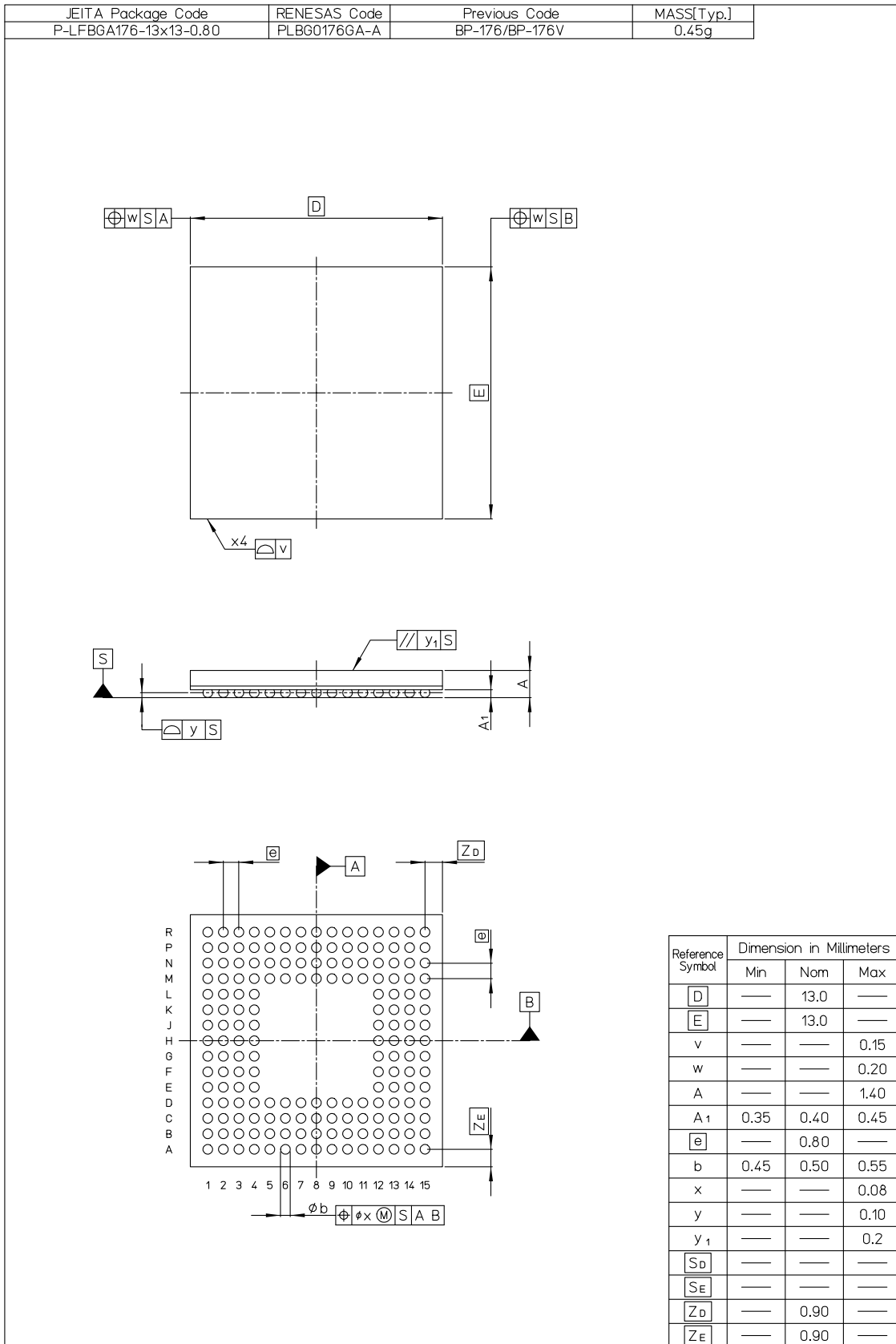


Figure B 176-Pin LFBGA (PLBG0176GA-A)



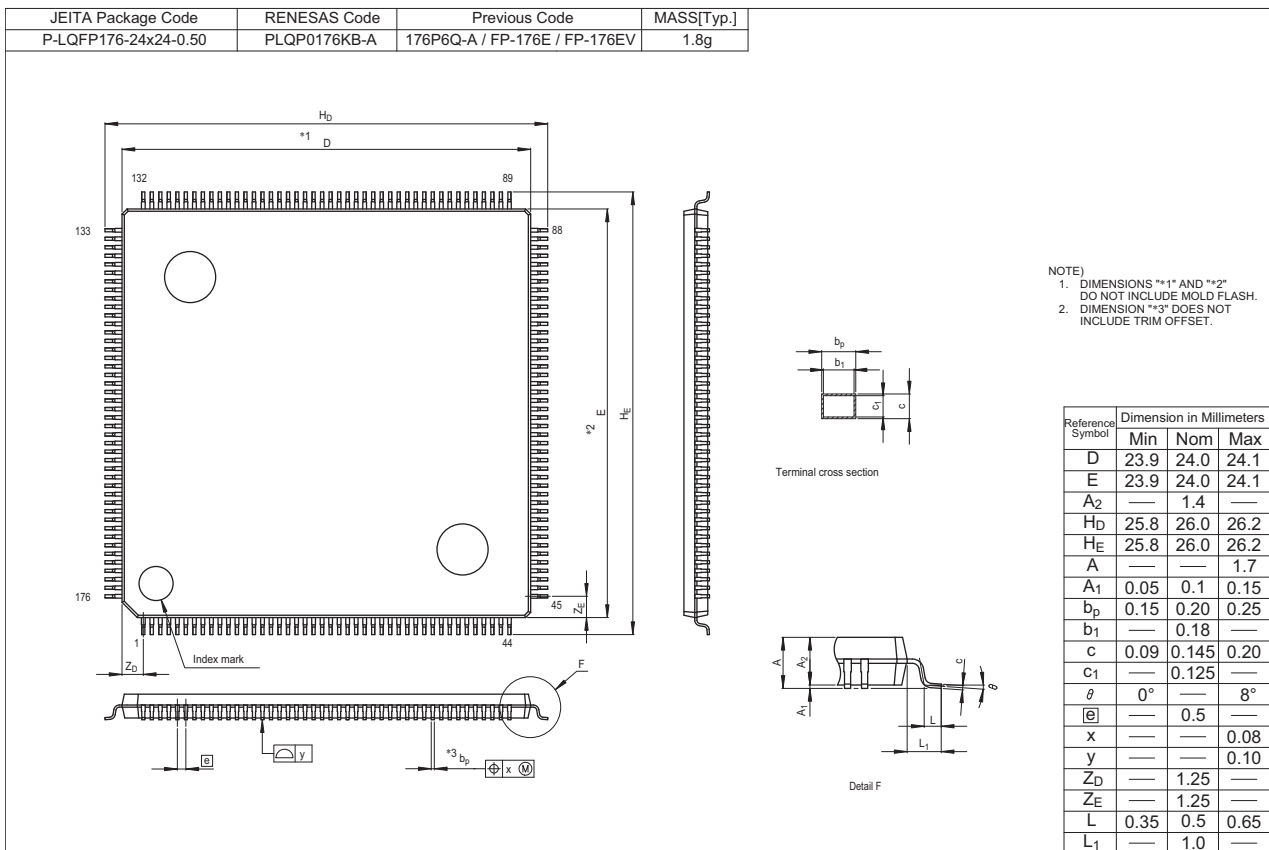


Figure C 176-Pin LQFP (PLQP0176KB-A)

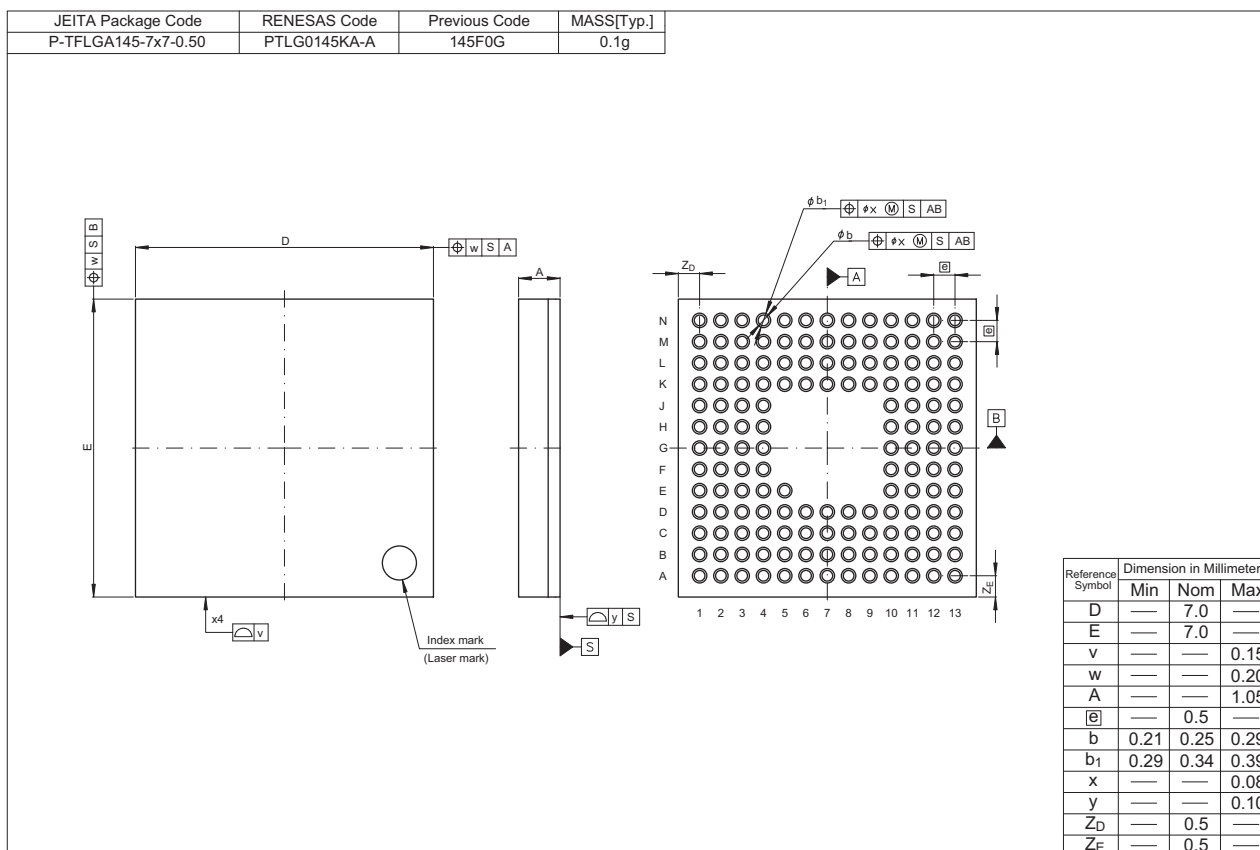


Figure D 145-Pin TFLGA (PTLG0145KA-A)

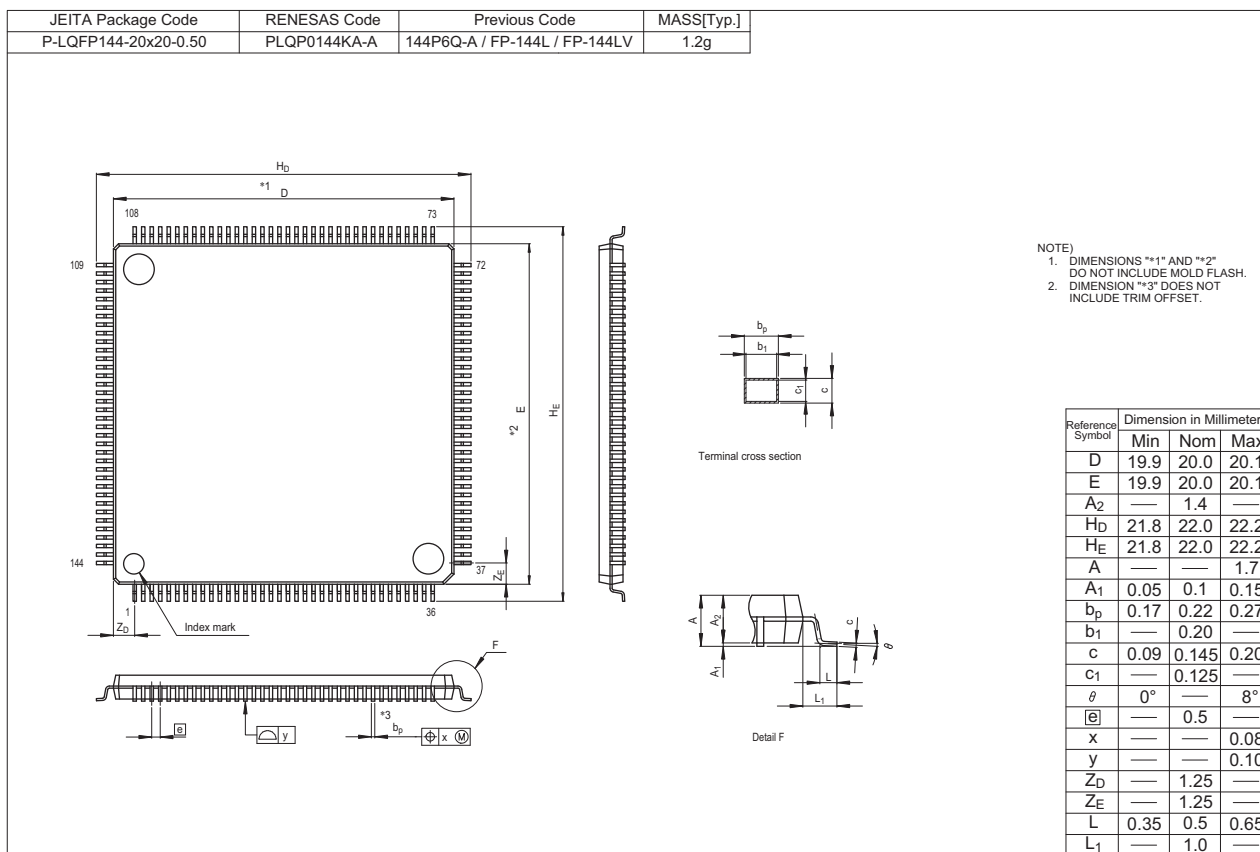


Figure E 144-Pin LQFP (PLQP0144KA-A)

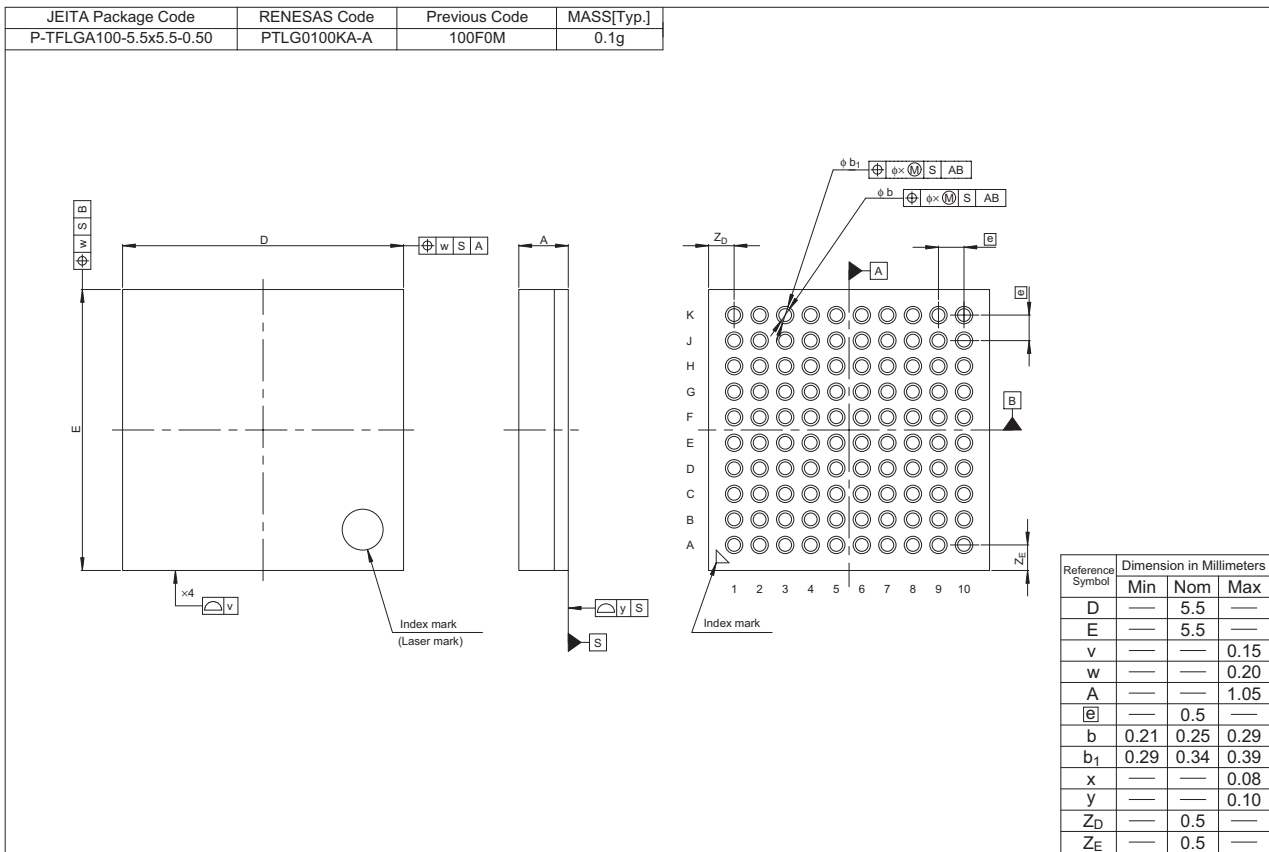


Figure F 100-Pin TFLGA (PTLG0100KA-A)

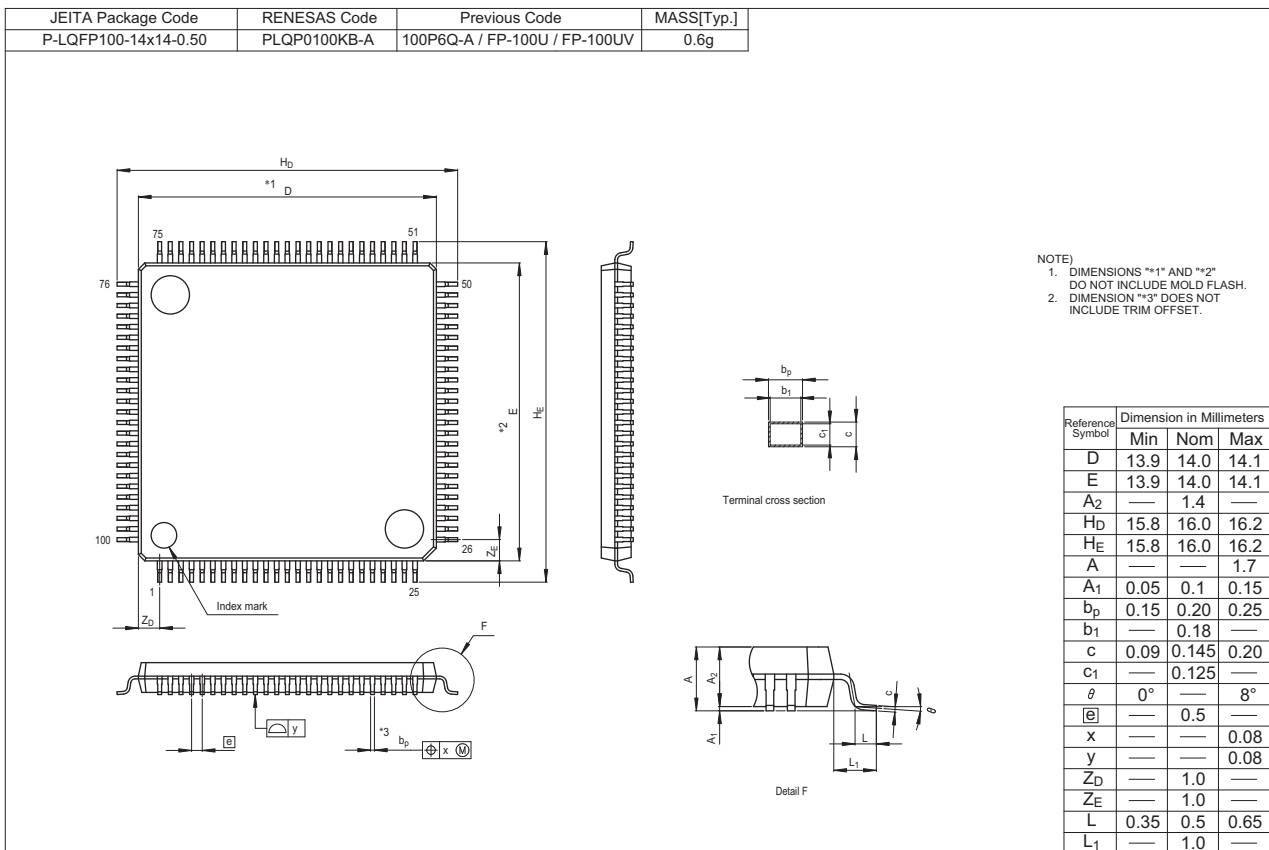


Figure G 100-Pin LQFP (PLQP0100KB-A)

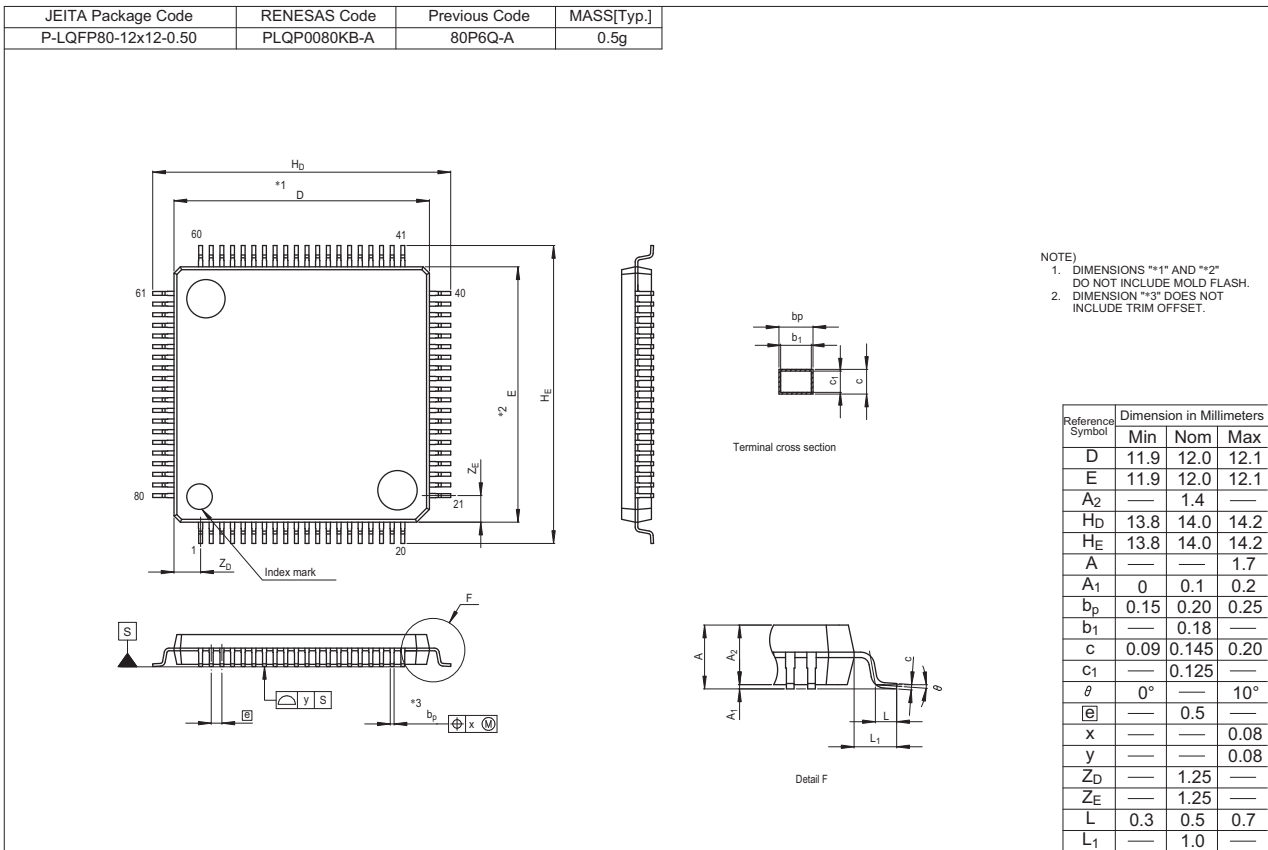


Figure H 80-Pin LQFP (PLQP0080KB-A)

REVISION HISTORY	RX630 Group User's Manual: Hardware
------------------	-------------------------------------

Rev.	Date	Description	
		Page	Summary
0.50	Mar 31, 2011	—	First Edition issued
1.00	Sep 8, 2011	All	
		—	Package added (177-pin TFLGA, 176-pin LFBGA, 145-pin TFLGA, 100-pin TFLGA) Module name changed
		<b>1. Overview</b>	
		45 to 49	Table 1.1 Outline of Specifications: Reset, real time clock, package, changed
		51, 52	Table 1.3 List of Products Table, changed
		55 to 59	Table 1.4 List of Pin Functions: BSCANP pin, added
		60	Figure 1.3 Pin Assignments (177-Pin TFLGA), added
		61	Figure 1.4 Pin Assignments (176-Pin LFBGA), added
		62	Figure 1.5 Pin Assignments (176-Pin LQFP): 16-pin and 18-pin, changed
		63	Figure 1.6 Pin Assignments (145-Pin TFLGA), added
		64	Figure 1.7 Pin Assignments (144-Pin LQFP): 16-pin, changed
		65	Figure 1.8 Pin Assignments (100-Pin TFLGA), added
		66	Figure 1.9 Pin Assignments (100-Pin LQFP): 7-pin, changed
		68 to 75	Table 1.5 List of Pins and Pin Functions (177-Pin TFLGA, 176-Pin LFBGA), added
		84 to 90	Table 1.7 List of Pins and Pin Functions (145-Pin TFLGA), added
		98 to 102	Table 1.9 List of Pins and Pin Functions (100-Pin TFLGA), added
		<b>2. CPU</b>	
		132	2.8.2 Instructions and Pipeline Processing, changed
		132	Table 2.13 Instructions that are Converted into a Single Micro-Operation, changed
		134, 135	Table 2.14 Instructions that are Converted into Multiple Micro-Operations, changed
		<b>3. Operating Modes</b>	
		140	Table 3.1 Selection of Operating Modes by the Mode Pin: Note 1 and Note 2, changed
		140	Table 3.2 Selection of Operating Modes by Register Setting: Note 1, changed
		140	Table 3.3 Endian Setting Method, changed
		140	Table 3.4 Selection of Endian, changed
		142	3.2.3 System Control Register 0 (SYSCR0) Bit function table (KEY[7:0]): Function, changed
		144	3.3.5 User Boot Mode, changed
		<b>5. I/O Registers</b>	
		150	(1) I/O Register Addresses (Address Order), changed
		151	(3) Number of I/O Registers to Access Cycles, changed
		152 to 191	Table 5.1 List of I/O Registers, changed
		<b>6. Resets</b>	
		193	Table 6.2 Targets to be Initialized by Each Reset Source, changed
		198	6.3.2 Power-On Reset and Voltage Monitoring 0 Reset, changed
		199	6.3.3 Voltage Monitoring 1 Reset and Voltage Monitoring 2 Reset, changed
		200	Figure 6.2 Example of Cold/Warm Start Determination Operation, changed
		201	6.3.4 Deep Software Standby Reset, changed
		201	6.3.5 Independent Watchdog Timer Reset, changed
		201	6.3.6 Watchdog Timer Reset, changed
		201	6.3.7 Software Reset, changed
		<b>7. Option-Setting Memory</b>	
		209	7.2.2 Option Function Select Register 1 (OFS1) Bit function table: Note 1, deleted, register description changed
		211	7.3 UB Code, changed
		<b>8. Voltage Detection Circuit (LVDA)</b>	
		212	8.1 Overview changed
		212	Table 8.1 Specifications of Voltage Detection Circuit, changed
		214	Figure 8.2 Block Diagram of Voltage Monitoring 1 Interrupt/Reset Circuit, changed
		214	Figure 8.3 Block Diagram of Voltage Monitoring 2 Interrupt/Reset Circuit, changed
		216	8.2.2 Voltage Monitoring 1 Circuit Status Register (LVD1SR) Bit function table (LVD1MON): Function changed, LVD1DET flag: Bit description changed

Rev.	Date	Description			
		Page	Summary		
1.00	Sep 8, 2011	218	8.2.4 Voltage Monitoring 2 Circuit Status Register (LVD2SR) Bit function table (LVD2MON): Function changed, LVD2DET flag: Bit description changed		
		220	8.2.6 Voltage Detection Level Select Register (LVDLVL) Bit allocation: Value after reset changed, Bit function table: Function changed, register description changed		
		221	8.2.7 Voltage Monitoring 1 Circuit Control Register 0 (LVD1CR0), LVD1RIE, LVD1DFDIS, LVD1FSAMP [1:0] bit: Bit description changed		
		223	8.2.8 Voltage Monitoring 2 Circuit Control Register 0 (LVD2CR0), LVD2RIE, LVD2DFDIS, LVD2FSAMP [1:0] bit: Bit description changed		
		225	8.3.2 Monitoring Vdet1, changed		
		225	8.3.3 Monitoring Vdet2, changed		
		227	8.5 Interrupt and Reset from Voltage Monitor 1, changed		
		227	Table 8.2 Procedures for Setting Bits Related to the Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset so that Voltage Monitoring Operates, changed		
		227	Table 8.3 Procedures for Setting Bits Related to the Voltage Monitor 1 Interrupt and Voltage Monitor 1 Reset to Stop Voltage Monitoring, added		
		228	Figure 8.5 Example of Voltage Monitoring 1 Interrupt Operation, changed		
		229	8.6 Interrupt and Reset from Voltage Monitor 2, changed		
		229	Table 8.4 Procedures for Setting Bits Related to the Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset so that Voltage Monitoring Operates, changed		
		229	Table 8.5 Procedures for Setting Bits Related to the Voltage Monitor 2 Interrupt and Voltage Monitor 2 Reset to Stop Voltage Monitoring, added		
		230	Figure 8.6 Example of Voltage Monitoring 2 Interrupt Operation, changed		
		<b>9. Clock Generation Circuit</b>			
		All	High-speed clock oscillator → High-speed on-chip oscillator, Low-speed clock oscillator → Low-speed on-chip oscillator, IWDT-dedicated low-speed clock oscillator → IWDT-dedicated on-chip oscillator, High-speed clock → HOCO clock, Low-speed clock → LOCO clock		
		231	Table 9.1 Specifications of Clock Generation Circuit: Operating frequency, sub-clock oscillator changed		
		234	9.2.1 System Clock Control Register (SCKCR) Bit function table (b3 to b0, b7 to b4, b15 to b12): Function changed, Notes 1 to 4 changed		
		235	9.2.2 System Clock Control Register 2 (SCKCR2), Bit function table (UCK[3:0]): Function changed		
		237	9.2.4 PLL Control Register (PLLCR), Bit function table (PLIDIV[1:0]): Function changed, register description changed		
		238	9.2.5 PLL Control Register 2 (PLLCR2), PLEN bit: Bit description changed		
		240	9.2.7 Main Clock Oscillator Control Register (MOSCCR), MOSTP bit: Bit description changed		
		241	9.2.8 Sub-Clock Oscillator Control Register (SOSCCR), SOSTP bit: Bit description changed		
		242	9.2.9 Low-Speed On-chip Oscillator Control Register (LOCOCR), LCSTP bit: Bit description changed		
		243	9.2.10 IWDT-Dedicated Low-Speed Clock Oscillator Control Register (ILOCOCR), ILCSTP bit: Bit description changed		
		244	9.2.11 High-Speed On-Chip Oscillator Control Register (HOCOOCR), HCSTP bit: Bit description changed		
		245	9.2.12 Oscillation Stop Detection Control Register (OSTDCR), OSTDIE bit: Bit description changed		
		247	9.2.14 Main Clock Oscillator Forced Oscillation Control Register (MOFCR), MOFXIN bit: Bit description changed		
		248	9.2.15 High-Speed On-Chip Oscillator Power Supply Control Register (HOCOPCR), HOCOPCNT bit: Bit description changed		
		249	Figure 9.2 Example of Crystal Resonator Connection, changed		
		249	Table 9.3 Damping Resistance (Reference Values), changed		
		249	Figure 9.3 Equivalent Circuit of Crystal Resonator, changed		
		249	Table 9.4 Crystal Resonator Characteristics (Reference Values), changed		
		250	9.3.2 External Clock Input, changed		
		250	9.3.3 Notes on the External Clock Input, added		
		251	Figure 9.5 Connection Example of 32.768-kHz Crystal Resonator, changed		
		251	Table 9.5 Crystal Resonator Characteristics (Reference Values), changed		
		251	9.4.2 Handling of Pins when Sub-Clock is Not Used, changed		
		253	Figure 9.8 Flow of Recovery from Detection of Oscillator Stop, added		
		253	9.5.2 Oscillation Stop Detection Interrupts, changed		
		254	9.7 Internal Clock, changed		
		256	9.9 Pin Settings When an External Clock is Connected: (2) Sub-clock, deleted		



Rev.	Date	Description	
		Page	Summary
1.00	Sep 8, 2011	256	9.10.1 Notes on Clock Generation Circuit: (2) and (3) changed
		257	9.10.4 Notes on Resonator Connect Pin, changed
		257	9.10.5 Notes on Sub-Clock, changed
		10. Frequency Measurement Circuit (MCK)	
		258	Table 10.1 Specifications of Frequency Measurement Circuit, changed
		259	Figure 10.1 Block Diagram of Frequency Measurement Circuit with MTU (System 1), changed
		259	Figure 10.2 Block Diagram of Frequency Measurement Circuit with TPU Unit 0 (System 2), changed
		260	Figure 10.3 Software Configuration, changed
		261	10.2.1 Counter-Clock Extension Register n (SCKn), Bit function table (SCK[1:0]): Function changed, register description changed
		—	10.2.2 Counter-Clock Extension Register 2 (SCK2), deleted
		262	Figure 10.4 Operation Flow of Frequency Measurement Circuit, changed
		263	Figure 10.5 Example of MTU/TPU Operation, changed
		11. Low Power Consumption	
		265	11.1 Overview, changed
		265	Table 11.1 Specifications of Low Power Consumption Functions, changed
		265, 266	Table 11.2 Entering and Exiting Low Power Consumption Modes and Operating States in Each Mode: Note 9 and Note 13 changed
		267	Figure 11.1 Mode Transitions, changed
		268	11.2.1 Standby Control Register (SBYCR), SSBY bit: Bit description changed
		273	11.2.5 Operating Power Control Register (OPCCR): Register description changed, OPCM[2:0] bits: Bit description changed
		276	11.2.7 Main Clock Oscillator Wait Control Register (MOSCWTCR): Register description changed
		277	11.2.8 Sub-Clock Oscillator Wait Control Register (SOSCWTCR): Register description changed
		278	11.2.9 PLL Wait Control Register (PLLWTCR): Register description changed
		279	11.2.10 Deep Standby Control Register (DPSBYCR), DEEPCUT[1:0], DPSBY bits: Bit description changed
		283	11.2.13 Deep Standby Interrupt Enable Register 2 (DPSIER2), Bit function table (DLVD1IE, DLVD2IE): Function changed
		294	11.3 Reducing Power Consumption by Switching Clock Signals, changed
		295	11.6.1.1 Transition to Sleep Mode, changed
		296	11.6.2.1 Transition to All-Module Clock Stop Mode, changed
		297	11.6.2.2 Canceling All-Module Clock Stop Mode, changed
		297	11.6.3.1 Transition to Software Standby Mode: Changed, Note 1 changed
		298	11.6.3.2 Canceling Software Standby Mode, changed
		301	11.6.4.2 Canceling Deep Software Standby Mode: changed
		303	Figure 11.4 Example of Flowchart to Use Deep Software Standby Mode, changed
		304	11.7.7 Rewrite the Register by DMACA and DTC in Sleep Mode, added
		304	11.7.8 Transition to Software Standby Mode for Products with 1.5-Mbyte ROM or More or Products with 176 Pins or More, changed
		305	11.7.10 Point for Caution when Using the Sub-Clock as the Source of the System Clock, added
		12. Battery Backup Function	
		—	12.2 Register Descriptions, deleted
		—	12.2.1 Battery Backup Voltage Monitor Function Select Register (VBATTMNSLR), deleted
		—	12.2.2 Battery Backup Voltage Monitor Register (VBATTMONR), deleted
		307	12.2.1 Battery Backup Function, changed
		307	Figure 12.2 Operation for Switching to Battery Backup Function, changed
		—	12.3.2 VBATT Voltage Monitor, deleted
		—	Figure 12.3 Operation Example of VBATT Voltage Monitor, deleted
		308	12.3 Usage Notes, changed
		13. Register Write Protection Function	
		309	Table 13.1 Correspondences between PRCR Bits and Registers to be Protected: PRC1, PRC3 bits changed
		310	13.1.1 Protect Register (PRCR) Bit function table (PRKEY[7:0]): Function changed
		15. Interrupt controller (ICUb)	
		319	Table 15.1 Specifications of Interrupt controller, changed
		320	Figure 15.1 Block Diagram of Interrupt controller, changed

Rev.	Date	Description			
		Page	Summary		
1.00	Sep 8, 2011	332	15.2.13 Non-Maskable Interrupt Status Register (NMISR), OSTST, LVD1ST, LVD2ST bits: Bit name changed		
		334	15.2.14 Non-Maskable Interrupt Enable Register (NMIER), OSTEN, LVD1EN, LVD2EN bits: Bit name changed		
		362	Figure 15.15 Operation Example in Which Multiple Level Detection Interrupt Requests of a Group are Generated, changed		
		368	15.6 Non-maskable Interrupt Operation, changed		
		370	15.8.1 Point for Caution Regarding WAIT Instructions While the Non-Maskable Interrupt is in Use, added		
		16. Buses			
		373	Table 16.2 Addresses Assigned for Each Bus, changed		
		376	Table 16.5 Specifications of the External Bus: Note 1, deleted		
		377	Table 16.6 Pin Configuration of the External Bus WR1#, WR2#, WR3#: Function changed, Note 1, changed		
		380	16.3.1 CSn Control Register (CSnCR) Bit function table: Note 3, deleted		
		388	16.3.5 CSn Wait Control Register 1 (CSnWCR1) CSPWAIT[2:0], CSWWAIT[4:0] bits: Note changed		
		391	16.3.6 CSn Wait Control Register 2 (CSnWCR2) Bit function table: Note 1, deleted WDOFF[2:0] bit: Bit description changed RDON[2:0], WRON[2:0], WDON[2:0] bits: Note changed		
		404	16.5.1 Separate Bus, changed		
		413	Figure 16.23 Normal Read Access, changed		
		427	Figure 16.40 Example of Recovery Cycle Insertion When a Bus Access is Split, changed		
		428	Figure 16.42 Example of Operation for Recovery Cycles when the BCLK Pin Output Selection Bits Are Set for Frequency-Division of BCLK by 2 (For the Case of Normal Access Through a Separate Bus Interface), added		
		429	Figure 16.43 Example of Operation for Recovery Cycles when the BCLK Pin Output Selection Bits Are Set for Frequency-Division of BCLK by 2 (For the Case where Bus Access is Divided up), added		
		429	Figure 16.44 Example of Recovery Cycle Insertion (with Address/Data Multiplexed I/O Interface), changed		
		430	Figure 16.45 Example of Recovery Cycle Insertion when a Bus Access is Split (with Address/Data Multiplexed I/O Interface), changed		
		431	Table 16.9 Limitations at the Time of Normal and Page Access, changed		
		434	Table 16.11 Types of Bus Errors, changed		
		17. DMA Controller (DMACA)			
		439	17.2.3 DMA Transfer Count Register (DMCRA), (3) Block Transfer Mode, changed		
		439	17.2.4 DMA Block Transfer Count Register (DMCRB): Register description changed		
		455	17.3.2 Extended Repeat Area Function, changed		
		463	Table 17.7 DMAC Execution Cycles: Note 1 added		
		18. Data Transfer Controller (DTCa)			
		472	Table 18.1 DTC Specifications, changed		
		475	18.2.2 DTC Mode Register B (MRB) Bit function table (CHNS): Function changed		
		478	18.2.6 DTC Transfer Count Register B (CRB) Register description changed		
		479	18.2.8 DTC Vector Base Register (DTCVBR) Bit allocation changed, register description changed		
		482	18.3 Sources of Activation, changed		
		492	Figure 18.9 Example (1) of DTC Operation Timing, changed		
		492	Figure 18.10 Example (2) of DTC Operation Timing, changed		
		493	Figure 18.11 Example (3) of DTC Operation Timing, changed		
		493	Figure 18.12 Example (4) of DTC Operation Timing, changed		
		494	Figure 18.13 Example of Operation when Transfer Information Skip is Executed, changed		
		495	Table 18.8 Execution Cycles of the DTC: Note 7, added		
		19. I/O Ports			
		—	All in this section, NMOS open drain output → N-channel open drain, changed		
		503	Table 19.1 Specifications of I/O Ports, changed		
		504, 505	Table 19.2 Port Functions, added		

Rev.	Date	Description	
		Page	Summary
1.00	Sep 8, 2011	506 to 509	19.2 I/O Port Configuration, added
		510	19.3.1 Port Direction Register (PDR): Register description changed
		511	19.3.2 Port Output Data Register (PODR): Register description changed
		512	19.3.3 Port Input Data Register (PIDR): Register name changed
		513	19.3.4 Port Mode Register (PMR): Register description changed
		514	19.3.5 Open Drain Control Register 0 (ODR0): Register description changed
		515	19.3.6 Open Drain Control Register 1 (ODR1): Register description changed
		516	19.3.7 Pull-Up Control Register (PCR): Register name changed
		517	19.3.8 Driving Ability Control Register (DSCR): Register description changed
		518	19.4 Handling of Unused Pins, added
		20. Multi-Function Pin Controller (MPC)	
		520 to 533	Table 20.1 Functions Assigned to Each Multiplexed Pin: Note 3 changed, Note 4 and Note 5, deleted
		534	20.2.1 Write-Protect Register (PWPR), PFSWE bit: Bit description changed
		556	Table 20.30 Register Settings for Input/Output Pin Function in 100-Pin TFLGA, 100-Pin LQFP, changed
		556	Table 20.31 Register Settings for Input/Output Pin Function in 80-Pin LQFP, changed
		558	Table 20.33 Register Settings for Input/Output Pin Function in 177-Pin TFLGA, 176-Pin LFBGA, 176-Pin LQFP, changed
		563	20.2.26 External Bus Control Register 1 (PFBCR1), Bit function table: Note 1, deleted, WAITS[1:0] bits: Bit description added
		565 to 567	20.3 How to Set the External Bus Interface, added
		568	20.4.2 Notes on MPC Register Setting: (6) changed
		569	Table 20.36 Register Settings, changed
		569	20.4.3 Notes on the Use of Analog Functions, changed
		21. Multi-Function Timer Pulse Unit 2 (MTU2a)	
		571, 572	Table 21.2 MTU Functions, changed
		573	Figure 21.1 Block Diagram of MTU, changed
		603	21.2.17 Timer Output Master Enable Registers (TOER) Bit function table: Note 1, changed
		649	Figure 21.39 Counter Operation in Complementary PWM Mode, changed
		651	Figure 21.40 Example of Operation in Complementary PWM Mode, changed
		653	Figure 21.41 Example of Operation without Dead Time, changed
		654	Figure 21.42 Example of PWM Cycle Updating, changed
		655	Figure 21.43 Example of Data Updating in Complementary PWM Mode, changed
		656	Figure 21.44 Example of Initial Output in Complementary PWM Mode (1), changed
		657	Figure 21.45 Example of Initial Output in Complementary PWM Mode (2), changed
		658	Figure 21.46 Example of Waveform Output in Complementary PWM Mode (1), changed
		658	Figure 21.47 Example of Waveform Output in Complementary PWM Mode (2), changed
		659	Figure 21.48 Example of Waveform Output in Complementary PWM Mode (3), changed
		659	Figure 21.49 Example of 0% and 100% Waveform Output in Complementary PWM Mode (1), changed
		660	Figure 21.50 Example of 0% and 100% Waveform Output in Complementary PWM Mode (2), changed
		660	Figure 21.51 Example of 0% and 100% Waveform Output in Complementary PWM Mode (3), changed
		661	Figure 21.52 Example of 0% and 100% Waveform Output in Complementary PWM Mode (4), changed
		661	Figure 21.53 Example of 0% and 100% Waveform Output in Complementary PWM Mode (5), changed
		662	Figure 21.55 Counter Clearing Synchronized with Another Channel, changed
		663	Figure 21.56 Timing for Synchronous Counter Clearing, changed
		664	Figure 21.58 Example of Synchronous Clearing in Dead Time during Up-Counting, changed
		665	Figure 21.59 Example of Synchronous Clearing in Interval Tb at Crest, changed
		665	Figure 21.60 Example of Synchronous Clearing in Dead Time during Down-Counting, changed
		666	Figure 21.61 Example of Synchronous Clearing in Interval Tb at Trough, changed
		669	Figure 21.68 Periods during which Interrupt Skipping Count can be Changed, changed
		671	Figure 21.70 Example of Operation when Buffer Transfer is Disabled (BTE1 = 0 and BTE0 = 1), changed

Rev.	Date	Description			
		Page	Summary		
1.00	Sep 8, 2011	671	Figure 21.71 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE1 = 1 and BTE0 = 0), changed		
		672	Figure 21.72 Relationship between Bits T3AEN and T4VEN in TITCR and Buffer Transfer-Enabled Period, changed		
		682	21.4.3 A/D Converter Activation: (4) changed		
		682	Table 21.58 Interrupt Sources and A/D Converter Start Request Signals, changed		
		700	Figure 21.121 Contention between Overflow and Counter Clearing, changed		
		700	21.6.18 Contention between TCNT Write Operation and Overflow/Underflow, changed		
		700	Figure 21.122 Contention between TCNT Write Operation and Overflow, changed		
		701	21.6.23 Notes when Complementary PWM Mode Output Protection Functions are not Used, changed		
		704	21.6.25 Continuous Output of Compare-Match Pulse Interrupt Signal, added		
		<b>22. Port Output Enable 2 (POE2a)</b>			
		731	Figure 22.1 POE Block Diagram, changed		
		733	Table 22.2 LSI Input/Output Pins to be Used by POE, changed		
		740	22.2.7 Input Level Control/Status Register 3 (ICSR3), OSTSTF flag: Bit description changed		
		743	Figure 22.3 Low-Level Detection Operation, changed		
		745	22.5.1 Transitions to Software Standby Mode or Deep Standby Mode, changed		
		745	22.5.2 When POE is not Used, added		
		745	22.5.3 Specifying Pins corresponding to the MTU, added		
		<b>23. 16-Bit Timer Pulse Unit (TPUa)</b>			
		749	Figure 23.1 Block Diagram of TPU (Unit 0), changed		
		750	Figure 23.2 Block Diagram of TPU (Unit 1), changed		
		803	23.8 PPG Trigger, added		
		<b>24. Programmable Pulse Generator (PPG)</b>			
		819	Figure 24.1 Block Diagram of PPG0, changed		
		819	Figure 24.2 Block Diagram of PPG1, changed		
		824 to 826	24.2.3 Output Data Registers H (PODRH), Output Data Register L (PODRL) <ul style="list-style-type: none"> <li>• PPG0.PODRH Bit function table: Function changed, PODi bit: Bit description added</li> <li>• PPG0.PODRL Bit function table: Function changed, PODi bit: Bit description added</li> <li>• PPG1.PODRH Bit function table: Function changed, PODi bit: Bit description added</li> <li>• PPG1.PODRL Bit function table: Function changed, PODi bit: Bit description added</li> </ul>		
		836	Figure 24.4 Schematic Diagram of PPG, changed		
		841	Figure 24.9 Non-Overlapping Pulse Output, changed		
		<b>25. 8-Bit Timer (TMR)</b>			
		—	All in this section, internal clock → frequency dividing clock, state → PCLK, changed		
		849	Table 25.2 Pin Configuration of TMR: Cascaded connection, changed		
		850	Figure 25.1 Block Diagram of TMR (Unit 0), changed		
		851	Figure 25.2 Block Diagram of TMR (Unit 1), changed		
		857, 858	25.2.6 Timer Control/Status Register (TCSR) <ul style="list-style-type: none"> <li>• TMR0.TCSR, TMR2.TCSR Register Bit function table: Note 1, changed</li> <li>• TMR1.TCSR, TMR3.TCSR Register Bit function table: Note 1, changed</li> </ul>		
		859	25.3.1 Pulse Output, changed		
		861	Figure 25.7 Timing of Interrupt Flag Setting to 1 at Compare Match, changed		
		863	Figure 25.10 Timing of Clearance by External Reset (Rising Edge), changed		
		863	Figure 25.11 Timing of Clearance by External Reset (High Level), changed		
		864	Figure 25.12 Timing of Overflow Interrupt Flag Setting to 1, changed		
		865	Table 25.7 A/D Converter Activation, added		
		866	Figure 25.13 Conflict between TCNT Write and Counter Clear, changed		
		870	Figure 25.16 Continuous Output of Compare Match Interrupt Signal, changed		
		<b>26. Compare Match Timer (CMT)</b>			
		—	All in this section, internal clock → frequency dividing clock, changed		
		871	Figure 26.1 Block Diagram of CMT (Unit 0), changed		
		—	26.5.4 Point for Caution When Changing the Value of CMCR, deleted		
		—	26.5.5 Point for Caution Regarding the Counter (CMCNT) and CMCOR, deleted		
		<b>27. Realtime Clock (RTCa)</b>			
		—	All in this section, software reset → RTC software reset, changed		
		878	Table 27.1 Specifications of RTC, Interrupt changed		
		879	Figure 27.1 Block Diagram of RTC, changed		

Rev.	Date	Description	
		Page	Summary
1.00	Sep 8, 2011	880	27.2 Register Descriptions, changed
		881	27.2.2 Second Counter (RSECCNT) Bit allocation, Values after reset, Bit function table (b7), changed
		881	27.2.3 Minute Counter (RMINCNT) Bit allocation, Values after reset, Bit function table (b7), changed
		882	27.2.4 Hour Counter (RHRCNT) Bit allocation, Values after reset, Bit function table (b7), changed
		883	27.2.5 Day-of-Week Counter (RWKCNT) Bit allocation, Values after reset, Bit function table (b7 to b3), changed
		888	27.2.12 Day-of-Week Alarm Register (RWKAR) Bit allocation, Values after reset, Bit function table (b6 to b3), changed
		889	27.2.13 Date Alarm register (RDAYAR) Bit allocation, Values after reset, Bit function table (b6), changed
		890	27.2.14 Month Alarm Register (RMONAR) Bit allocation, Values after reset, Bit function table (b6, b5), changed
		891	27.2.16 Year Alarm Enable Register (RYRAREN) Bit allocation, Values after reset, Bit function table (b6 to b0), changed
		892	27.2.17 RTC Control Register 1 (RCR1) Bit function table (AIE, CIE, PIE), CIE bit description, changed
		893, 894	27.2.18 RTC Control Register 2 (RCR2) Bit allocation, Values after reset, Bit function table (b7), changed
		895	27.2.19 RTC Control Register 3 (RCR3) Bit function table (b3 to b1), Register description, changed
		897	27.2.22 Time Error Adjustment Register (RADJ) ADJ[5:0] bit name, PMADJ[1:0] bit description, changed
		898, 899	27.2.23 Time Capture Control Register y (RTCCRy) TCNF[1:0] bit description, changed
		899	27.2.24 Second Capture Control Register y (RSECCPy) Bit allocation, Values after reset, Bit function table (b7), changed
		900	27.2.25 Minute Capture Control Register y (RMINCPy) Bit allocation, Values after reset, Bit function table (b7), changed
		900	27.2.26 Hour Capture Register y (RHRCPy) Bit allocation, Values after reset, Bit function table (b7), changed
		901	27.2.27 Date Capture Register y (RDAYCPy) Bit allocation, Values after reset, Bit function table (b7, b6), changed
		902	Figure 27.2 Outline of Initial Settings after Power-On, changed
		904	Figure 27.4 Setting the Time, changed
		908	27.3.8.3 Procedure for Changing the Mode of Adjustment, changed
		910	27.4 Interrupt Sources (1), (3), changed
		912	Figure 27.13 Using Periodic Interrupt Function, changed
		912	27.5.4 Transitions to Low Power Consumption Modes after Setting Registers, changed
		912	27.5.5 Points for Caution When Writing to and Reading from Registers, changed
			28. Watchdog Timer (WDTA)
		913	Overview, changed
		913, 914	Table 28.1 Auto-start mode in Specifications of WDT, changed
		914	Figure 28.1 Block Diagram of WDT, changed
		922	28.3.1.2 Auto-Start Mode, changed
		924	Figure 28.5 Control Waveforms Produced in Response to Writing to the WDTCCR, changed
		924, 925	28.3.3 Refresh Operation [Sample sequences of writing that are not valid for refreshing the counter], changed
		925	28.3.4 Status Flags, changed
		926	28.3.5 Reset Output, changed
		926	28.3.7 Reading the Down-Counter Value, changed
			29. Independent Watchdog Timer (IWDTa)
		928	Overview, changed
		928, 929	Table 29.1 Auto-start mode in Specifications of IWDT, changed
		929	Figure 29.1 Block Diagram of IWDT, changed
		937	29.3.1.2 Auto-Start Mode, changed
		939	Figure 29.5 Control Waveforms Produced in Response to Writing to the IWDTCCR, changed
		939, 940	29.3.3 Refresh Operation [Sample sequences of writing that are not valid for refreshing the counter], changed

Rev.	Date	Description			
		Page	Summary		
1.00	Sep 8, 2011	941	29.3.4 Status Flags, changed		
		941	29.3.5 Reset Output, changed		
		942	29.3.7 Reading the Down-Counter Value, changed		
		943	29.4 Usage Notes, 29.4.1 Refresh Operations, added		
		30. USB 2.0 Function Module (USBa)			
		947	Table 30.3 Registers Initialized by Writing SYSCFG.USBE = 0, changed		
		947	30.2.2 System Configuration Status Register 0 (SYSSTS0) Bit function table (LNST[1:0]), changed		
		955, 956	30.2.6 CFIFO Port Control Register (CFIFOCTR), D0FIFO Port Control Register (D0FIFOCTR), D1FIFO Port Control Register (D1FIFOCTR) BVAL bit description, changed		
		—	30.2.23 DCP Configuration Register (DCPCFG), deleted		
		977	30.2.27 Pipe Maximum Packet Size Register (PIPEMAXP) Bit function table (MXPS[8:0]), changed		
		979 to 981	30.2.29 Pipe Cycle Control Register (PIPEPERI) Bit function table (IITV[2:0]), changed IITV[2:0], IFIS bit description, added		
		982	Table 30.7 Information Cleared by USB Module by Setting ACLRM = 1, changed		
		984	Table 30.9 Information Cleared by USB Module by Setting ACLRM = 1, changed		
		994	Table 30.11 Interrupt Sources, changed		
		—	30.3.3.9 BCHG Interrupt, deleted		
		1005, 1006	Table 30.14 Pipe Setting Items, changed		
		1006	30.3.4.1 Pipe Control Register Switching Procedures, changed		
		31. Serial Communications Interface (SCIC, SCID)			
		1021, 1022	Table 31.1 Specifications of SCIC, Note 1 changed		
		1022, 1023	Table 31.2 Specifications of SCID, Note 1 changed		
		1024	Figure 31.1 Block Diagram of SCIO to SCI4, SCI7 to SCI11, changed		
		1025	Figure 31.2 Block Diagram of SCI5 and SCI6, changed		
		1026	Figure 31.3 Block Diagram of SCI12 (SCID), changed		
		1032, 1033	31.2.5 Serial Mode Register (SMR) (2) Smart Card Interface Mode Bit function table (PE), changed		
		1034 to 1037	31.2.6 Serial Control Register (SCR) (1) Serial Communications Interface Mode Bit function table (CKE[1:0]), CKE[1:0] bit description, changed (2) Smart Card Interface Mode Bit function table (CKE[1:0]), changed		
		1038 to 1041	31.2.7 Serial Status Register (SSR) (1) Serial Communications Interface Mode TEND flag description, changed (2) Smart Card Interface Mode TEND flag description, changed		
		1042	31.2.8 Smart Card Mode Register (SCMR) Note 2 in the Bit function table, added SINV, SDIR, bit name, changed SDIR bit description, changed		
		1048	Table 31.16 Maximum Bit Rate with External Clock Input (Asynchronous Mode) (1), changed		
		1049	Table 31.17 BRR Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode), changed		
		1051	31.2.10 Serial Extended Mode Register (SEMR) Bit function table (NFEN), changed ACS0, NFEN bit description, changed		
		1053	31.2.11 Noise Filter Setting Register (SNFR) Bit function table (NFCS[2:0]), changed NFCS[2:0] bit description, changed		
		1059, 1060	31.2.16 SPI Mode Register (SPMR) CKPOL, CKPH bit description, changed		
		1060	31.2.17 Extended Serial Module Enable Register (ESMER) ESME bit description, changed		
		1061	Table 31.21 Settings of the ESME Bits and Guaranteed Operation by Timer Operation Mode, added		
		1063	31.2.20 Control Register 2 (CR2) Bit function table (DFCS[2:0], BCCS[1:0], RTS[1:0]) changed, Note added		

Rev.	Date	Description	
		Page	Summary
1.00	Sep 8, 2011	1075	31.3.4 CTS and RTS Functions, changed
		1076	Figure 31.8 Sample SCI Initialization Flowchart (Asynchronous Mode), changed
		1077	31.3.6 Serial Data Transmission (Asynchronous Mode), changed
		1077	Figure 31.9 Example of Operation for Serial Transmission in Asynchronous Mode (from the Middle of Transmission until Transmission Completion), changed
		1078	Figure 31.10 Example of Serial Transmission Flowchart in Asynchronous Mode, changed
		1079	Figure 31.11 Example of SCI Operation for Serial Reception in Asynchronous Mode (1) (when RTS Function is not Used), changed
		1080	Figure 31.12 Example of SCI Operation for Serial Reception in Asynchronous Mode (2) (when RTS Function is Used), added
		1081	Figure 31.13 Example of Serial Reception Flowchart (1) (Asynchronous Mode), changed
		1082	Figure 31.14 Example of Serial Reception Flowchart (2) (Asynchronous Mode), changed
		1084	Figure 31.16 Example of Multi-Processor Serial Transmission Flowchart, changed
		1086	Figure 31.18 Example of Multi-Processor Serial Reception Flowchart (1), changed
		1087	Figure 31.19 Example of Multi-Processor Serial Reception Flowchart (2), changed
		1088	31.5.2 CTS and RTS Functions, changed
		1089	Figure 31.21 Example of SCI Initialization Flowchart (Clock Synchronous Mode), changed
		1090	31.5.4 Serial Data Transmission (Clock Synchronous Mode), changed
		1090	Figure 31.22 Example of Operation for Serial Transmission in Clock Synchronous Mode (from the Middle of Transmission until Transmission Completion), changed
		1091	Figure 31.23 Example of Serial Transmission Flowchart (Clock Synchronous Mode), changed
		1092	Figure 31.24 Example of Operation for Serial Reception in Clock Synchronous Mode (1) (when RTS Is not Used), changed
		1092	Figure 31.25 Example of Operation for Serial Reception in Clock Synchronous Mode (2) (when RTS Is Used), added
		1093	Figure 31.26 Example of Serial Reception Flowchart (Clock Synchronous Mode), changed
		1094	31.5.6 Simultaneous Serial Data Transmission and Reception (Clock Synchronous Mode), changed
		1094	Figure 31.27 Example of Simultaneous Serial Transmission and Reception Flowchart (Clock Synchronous Mode), changed
		1098	31.6.5 Initialization of the SCI (Smart Card Interface Mode), changed
		1101	Figure 31.35 Sample Smart Card Interface Transmission Flowchart, changed
		1103	Figure 31.37 Sample Smart Card Interface Reception Flowchart, changed
		1104	31.6.8 Clock Output Control (1), (2), changed
		1107	31.7.2 Clock Synchronization, changed
		1109	Figure 31.45 Example of the Flow of SCI Initialization (for Simple I2C Mode), changed
		1111	Figure 31.48 Example of the Procedure for Master Transmission Operations in Simple I2C Mode (with Transmission Interrupts and Reception Interrupts in Use), changed
		1113	Figure 31.50 Example of the Procedure for Master Reception Operations in Simple I2C Mode (with Transmission Interrupts and Reception Interrupts in Use), changed
		1114	31.8 Operation in Simple SPI Mode, changed
		1115	Table 31.24 States of Pins by Mode and Input Level on the SSn# Pin Note 1 changed, Note 2, Note 3, added
		1116	31.8.5 SCI Initialization (Simple SPI Mode), changed
		1119	Figure 31.55 Sample Flowchart for Transmission of a Start Frame (1), changed
		1123	Figure 31.58 Sample Flowchart for Reception of a Start Frame (1), changed
		1127	31.9.4 Detection of Bus Collisions, changed
		1127	Figure 31.62 Example of Operations with Bus-Collision Detection, changed
		1128	Figure 31.63 Example of Operations with the Digital Filter, changed
		1130	31.9.7 Selectable Timing for Sampling Data Received through RXDX12, changed
		1130	Figure 31.65 Timing for Sampling of Data Received through RXDX12, changed
		1132, 1133	31.10 Noise Cancellation Function, changed
		1133, 1134	31.11.2 Interrupts in Serial Communications Interface and Simple SPI Mode, changed
		1136	31.12.3 The Mark State and Production of Breaks, changed
		1140	31.12.11 Limitations on Simple SPI Mode, added
		1141	31.12.13 Limitation 2 on Usage of the Extended Serial Mode Control Section, changed
		1141	Figure 31.74 Example of Flowchart for Reception Error Handling (During Reception of the Start Frame), changed
		32. I <sup>2</sup> C Bus Interface (RIIC)	
		1142	32.1 Overview Description of RIIC input signal level, added

Rev.	Date	Description	
		Page	Summary
1.00	Sep 8, 2011	—	32.15.2 Setting Input Buffer Control Register, deleted
			<b>33. CAN Module (CAN)</b>
		1211	Figure 33.1 Block Diagram of CAN Module, changed
		1212 to 1215	33.2.1 Control Register (CTLR) BOM[1:0] bit description, changed
		1226 to 1228	33.2.8 Message Control Register j (MCTLj) TRMREQ bit description, changed
		1232, 1233	33.2.11 FIFO Control Register (TFCR) TFUST[2:0] bit description, changed
		1239	33.2.16 Channel Search Support Register (CSSR) Register description, changed
		1239	Figure 33.4 Write and Read of CSSR and MSSR, changed
		1240	33.2.17 Acceptance Filter Support Register (AFSR) Note of the Bit function table, added
		1242 to 1244	33.2.19 Error Interrupt Factor Judge Register (EIFR) EWIF flag description, changed
		1249	Figure 33.9 Transition between CAN Operating Modes, added
		1251	Table 33.8 Operation in CAN Reset Mode and CAN Halt Mode, Note 4, changed
		1252	Figure 33.10 Sub-Modes of CAN Operation Mode, changed
			<b>34. Serial Peripheral Interface (RSPI)</b>
		1264	34.1 Overview, Description of input/output pin for RSPI, changed
		1264	Table 34.1 Specifications of RSPI, changed
		1265	Figure 34.1 Block Diagram of RSPI, Others, changed
		1267, 1268	34.2.1 RSPI Control Register (SPCR) SPTIE, SPE, SPRIE bit description, changed
		1269	34.2.3 RSPI Pin Control Register (SPPCR) SPOM bit, deleted (changed to a reserved bit)
		1271 to 1274	34.2.5 RSPI Data Register (SPDR) Register description, changed
		1272	Figure 34.2 Configuration of SPDR, added
		1273	Figure 34.3 Configuration of SPDR (Writing), added
		1274	Figure 34.4 Configuration of SPDR (Reading), added
		1276	34.2.7 RSPI Sequence Status Register (SPSSR) Bit function table (b3, b7), R/W, changed
		1278	34.2.9 RSPI Data Control Register (SPDCR) SLSEL[1:0] bits deleted (changed to reserved bits), Bit function table (SPFC[1:0]) changed, SPFC[1:0], SPRDTD bit description changed
		—	Figure 34.5 Frame Configurations and Examples of Combinations of Transmission and Reception Settings, deleted
		1279	34.2.10 RSPI Clock Delay Register (SPCKD) Register description, changed
		1280	34.2.11 RSPI Slave Select Negation Delay Register (SSLND) Register description, changed
		1280, 1281	34.2.12 RSPI Next-Access Delay Register (SPND) Register description, changed
		1285	Table 34.5 Relationship between RSPI Modes and SPCR Settings and Description of Each Mode, changed
		1286	34.3.2 Controlling RSPI Pins, changed
		1286	Table 34.6 Relationship between Pin States and Bit Settings, changed
		1293	34.3.4 Data Format, changed
		1293	Figure 34.13 Outline of the Data Format (with Parity Disabled), added
		1293	Figure 34.14 Outline of the Data Format (with Parity Enabled), added
		1294 to 1297	34.3.4.1 When Parity is Disabled (SPCR2.SPPE = 0), changed
		1298 to 1301	34.3.4.2 When Parity is Enabled (SPCR2.SPPE = 1), changed
		1304	Figure 34.25 Operation Example of SPCR.TXMD = 0, changed
		1305	Figure 34.26 Operation Example of SPCR.TXMD = 1, changed
		1306	Figure 34.27 Operation Example of SPTI and SPRI Interrupts
		1308	Table 34.8 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function, changed
		1309	Figure 34.28 Operation Example of OVRF Flag, changed



Rev.	Date	Description			
		Page	Summary		
1.00	Sep 8, 2011	1310	Figure 34.29 Operation Example of PERF Flag, changed		
		1312 to 1315, 1319	34.3.10.1 Master Mode Operation (1) Starting a Serial Transfer, changed (2) Terminating a Serial Transfer, changed (3) Sequence Control, changed Figure 34.30 Procedure for Determining the Form of Serial Transmission in Master Mode, changed Figure 34.31 Concept of a Frame, changed Figure 34.32 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations, changed (4) Burst Transfer, changed (9) Flowchart of Operations, changed Figure 34.35 Transfer Operation Flowchart in Master Mode (SPI Operation), changed		
		1320, 1322	34.3.10.2 Slave Mode Operation (2) Terminating a Serial Transfer, changed (6) Flowchart of Operations, changed Figure 34.37 Example Flowchart for Transfer Operations in Slave Mode (SPI Operation), changed		
		1323 to 1327	34.3.12 Master Mode Operation (1) Starting a Serial Transfer, changed (2) Terminating a Serial Transfer, changed (3) Sequence Control, changed Figure 34.38 Procedure for Determining the Form of Serial Transmission in Master Mode, changed Figure 34.39 Concept of a Frame, changed Figure 34.40 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations, changed (4) Initialization Flowchart, changed Figure 34.41 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation), changed (5) Flowchart of Operations, changed Figure 34.42 Example Flowchart for Transfer Operations in Master Mode (Clock Synchronous Operation), changed		
		1330	34.3.13 Slave Mode Operation (4) Transfer Operation Flowcharts, changed Figure 34.44 Example Flowchart for Transfer Operations in Slave Mode (CPHA = 1) (Clock Synchronous Operation), changed		
		1333	Figure 34.48 Configuration of Shift Register I/O Paths in Loopback Mode (Master Mode), changed		
		1335	34.3.17 Interrupt Sources, changed		
		1335	34.4.2 Cautionary Note on the Low Power Consumption Functions, added		
		35. IEBus Controller (IEB)			
		1343	Figure 35.5 Block Diagram of IEB, changed		
		36. CRC Calculator (CRC)			
		1386	36.2.1 CRC Control Register (CRCCR) Bit function table (DORCLR) changed, Note 1 added		
		1387	36.2.3 CRC Data Output Register (CRCDOR) Register description, changed		
		37. 12-Bit A/D Converter (S12ADa)			
		—	All in this section, internal reference voltage → A/D internal reference voltage, activation → start, one-cycle scan mode → single scan mode, changed		
		1392	Table 37.1 Specifications of 12-Bit A/D Converter, changed, Note 3 added		
		1393	Table 37.2 Function Overview of the 12-Bit A/D Converter, changed, Note 3 added		
		1394	Figure 37.1 Block Diagram of 12-Bit A/D Converter, changed		
		1395, 1396	37.2.1 A/D Control Register (ADCSR) ADCS bit description changed, Table 37.4 Scan Mode and Scope of Selectability for A/D Conversion, changed, ADST bit description changed		
		1400	37.2.7 A/D Control Extended Register (ADCER) ACE bit description, changed		
		1401	37.2.8 A/D Start Trigger Select Register (ADSTRGR) Table 37.5 List of Sources to Start A/D Conversion, changed		
		1406	37.2.13 A/D Sampling State Register 01 (ADSSTR01), added		
		1407	37.2.14 A/D Sampling State Register 23 (ADSSTR23) Bit function table (b7 to b0), changed		
		1411	37.3.4 Analog Input Sampling and Scan Conversion Time, changed		
		1411	Table 37.6 Processing During a Scan Operation, changed		
		1411	Table 37.7 Times Required in Scan Operations, changed		

Rev.	Date	Description	
		Page	Summary
1.00	Sep 8, 2011	1412	37.3.5 Usage Example of ADDRy Register Automatic Clearing Function, changed
		1413	37.3.8 Starting A/D Converter with Synchronous Trigger from Peripheral Modules, changed
		1413	Table 37.8 Selecting the Trigger for A/D Conversion, changed
		1414	37.3.8.2 Starting A/D Conversion using MTU TRGAN_0 and TPUA TRGAN_1, changed
		1417	37.3.8.4 Starting A/D Conversion using MTU TRG4ABN_0 and TPUA TRG4ABN_1, changed
		1419	37.5.2 Notes on Forcibly Stopping A/D Conversion, changed
		1419	37.5.3 Notes on Restarting A/D Conversion after a Forcible Stop, changed
		1419	37.5.5 Notes on Entering Low Power Consumption States, changed
		—	37.5.7 Permissible Signal Source Impedance, deleted
		38. 10-Bit A/D Converter (ADb)	
		1423	Table 38.2 Function Overview of the 10-Bit A/D Converter, Note 1 changed
		1424	Figure 38.1 Block Diagram of 10-Bit A/D Converter, changed
		1426	38.2.2 A/D Control/Status Register (ADCSR) Bit function table Note 1, changed
		1439	38.3.6 Using Synchronous Triggers from Peripheral Modules to Start A/D Conversion, added
		1439	38.3.6.2 Starting A/D Conversion using MTU TRGAN_0 and TPUA TRGAN_1, changed
		1445	38.6.2 Notes on Forcibly Stopping A/D Conversion, changed
		1445	38.6.3 A/D Conversion Restart Timing after Forced Stop, changed
		39. D/A Converter (DAa)	
		1450	Table 39.1 Specifications of D/A Converter, Note 1 changed
		1450	Figure 39.1 Block Diagram of D/A Converter, changed
		1452, 1453	39.2.2 D/A Control Register (DACR) Register description, changed DAE, AOE0, DAOE1 bit description, changed
		1454	39.2.4 D/A A/D Synchronous Start Control Register (DAADSCR) Bit function table (DAADST) changed, register description changed, DAADST bit description changed
		1455	39.3 Operation, 3, changed
		1457	39.4.5 Note on Usage when Measure against Interference between D/A and A/D Conversion is Enabled, changed
		1457	39.4.6 Notes on Usage when Measure against Interference between D/A and A/D Conversion is Disabled, changed
		40. Temperature Sensor	
		1458	Figure 40.1 Block Diagram of Temperature Sensor, changed
		1460	40.3.1 Preparation for Using the Temperature Sensor, changed
		1461	40.3.2 Setting of 12-Bit A/D Converter, changed
		1462	Figure 40.2 Procedure for Using the Temperature Sensor, changed
		1463	Figure 40.3 Timing from the Start of Temperature-Sensor Operation until Completion of A/D Conversion, changed
		1463	Table 40.2 Time until Completion of A/D Conversion after the Start of Temperature-Sensor Operation, changed
		42. ROM (Flash Memory for Code Storage)	
		—	All in this section, mat → area, changed
		1465	Table 42.2 Correspondence between ROM Capacity and ROM Addresses, changed
		1466	Figure 42.1 Block Diagram of ROM, changed
		1468	42.2.2 Flash Mode Register (FMODR), register description changed
		1469	42.2.3 Flash Access Status Register (FASTAT) Register description changed, ROMA bit description changed
		1471	42.2.4 Flash Access Error Interrupt Enable Register (FAEINT) Register description changed
		1472	42.2.5 FCU RAM Enable Register (FCURAME) Register description changed
		1476	42.2.8 Flash Ready Interrupt Enable Register (FRDYIE) Register description changed
		1477	42.2.9 Flash P/E Mode Entry Register (FENTRYR) Bit function table, Note 2 changed, register description changed, Note 2 changed
		1480	42.2.11 Flash Reset Register (FRESETR) Register description changed
		1485	42.4 Block Configuration, changed

Rev.	Date	Description			
		Page	Summary		
1.00	Sep 8, 2011	—	Figure 42.4 Transitions between Operating Modes in Terms of the ROM, deleted		
		1491	Table 42.8 Acceptable Commands and the State and Mode (ROM P/E Mode) of the FCU, changed		
		1493	Figure 42.6 Procedure for Transition to ROM Read Mode, changed (tFCUR)		
		1494	Figure 42.8 Procedure for Transition to ROM Status Read Mode and the Status Checking, changed		
		1497	Figure 42.11 Procedure for Firmware Transfer to FCU RAM, changed		
		1499	Figure 42.12 Using the Peripheral Clock Notification Command, changed (tFCUR)		
		1501	Figure 42.13 Procedure for ROM Programming, changed (tFCUR)		
		1502	Figure 42.14 Procedure for ROM Erasure, changed (tFCUR)		
		1503	Figure 42.15 Procedure for Programming the Lock Bit, changed (tFCUR)		
		1504	Figure 42.16 Procedure for Reading Lock Bit in Register Read Mode, changed (tFCUR)		
		1507	Figure 42.18 Procedure for Programming/Erasure Suspension, changed (tFCUR)		
		1508	Figure 42.19 Procedure for Resuming Programming or Erasure, changed (tFCUR)		
		1509	Figure 42.20 Suspension during Programming, changed		
		1513	Table 42.9 Error Protection Types (Types Dedicated to ROM and Types Common to ROM and E2 DataFlash), illegal command error, changed		
		1519 to 1521, 1525	42.10.6 Inquiry/Selection Host Command Wait State (1) Supported Device Inquiry, changed (2) Device Selection, changed (9) Erasure Block Information Inquiry, changed (11) New Bit Rate Selection, changed		
		1530, 1531	42.10.8 Programming/Erasure Host Command Wait State, changed (128 bytes → 256 bytes) Table 42.16 Programming/Erasure Host Commands, changed (128 bytes → 256 bytes) Figure 42.30 Procedure for ROM Programming in Boot Mode, changed		
		1532	(3) 256-Byte Programming, changed (128 bytes → 256 bytes)		
		1542, 1543	42.14 Usage Notes (2) Suspending Programming or Erasure, changed (4) Reset during Programming or Erasure, changed (5) Prohibition of Non-Maskable Interrupts during Programming or Erasure, changed (7) Programming and Erasure in Low-Speed Operating Modes 1 and 2, added (8) Programming/Erasure Abnormal End, changed (9) Actions Prohibited during Programming and Erasure, changed (10) Notes on Flash Programming in Boot Mode or USB Boot Mode, added (11) Handling of the EXTAL Pin in Boot Mode, added (12) Handling of the EXTAL Pin in USB Boot Mode, added		
		43. E2 DataFlash Memory (Flash Memory for Data Storage)			
		1544	Table 43.1 Specifications of E2 DataFlash Memory, changed (Reading via the peripheral bus)		
		1545	Figure 43.1 Block Diagram of E2 DataFlash Memory, changed		
		1546	Table 43.2 Input and Output Pins Associated with the E2 DataFlash, changed (PC7 added)		
		1553	43.2.6 E2 DataFlash Programming/Erasure Enable Register 0 (DFLWE0), register description, changed		
		1554	43.2.7 E2 DataFlash Programming/Erasure Enable Register 1 (DFLWE1), register description, changed		
		1555	43.2.8 Flash P/E Mode Entry Register (FENTRYR) Register description changed, Note 2 changed		
		1562	Table 43.5 FCU Command Formats (Commands Dedicated to the E2 DataFlash Memory), WA, BA changed		
		1563	Table 43.6 Acceptable Commands and the State and Mode (E2 DataFlash P/E Mode) of the FCU, changed		
		1573	43.9 Usage Notes, (2) Reading in Low-Speed Operating Mode 2, added		
		44. Boundary Scan			
		1574 to 1590	Added		
		45. Electrical Characteristics			
		1591 to 1630	Added		
		Appendix 1. Port States in Each Processing Mode			
		1631 to 1633	Table 1.1 Port States in Each Processing State, changed		
		Appendix 2. Package Dimensions			
		1634	Figure A. 177-Pin TFLGA (PTLG0177KA-A), added		
		1635	Figure B. 176-Pin LFBGA (PLBG0176GA-A), added		
		1637	Figure D. 145-Pin TFLGA (PTLG0145KA-A), added		
		1639	Figure F. 100-Pin TFLGA (PTLG0100KA-A), added		
		1.10	Oct 7, 2011	41. RAM	
				1464	Description on RAM, changed

Rev.	Date	Description	
		Page	Summary
1.11	Oct 24, 2011	1. Overview	
		53	Figure 1.1 How to Read the Product Part Number, changed
		7. Option-Setting Memory	
		211	7.3 UB Codes, changed
		28. Watchdog Timer (WDTA)	
		919	28.2.3 WDT Status Register (WDTSR), UNDF, and REFEF flags: Flags description changed
		920	28.2.4 WDT Reset Control Register (WDTRCR), RSTIRQS bit: Bit description changed
		29. Independent Watchdog Timer (IWDtA)	
		934	29.2.3 IWDt Status Register (IWDTSR), UNDF, and REFEF flags: Flags description changed
		935	29.2.4 IWDt Reset Control Register (IWDTRCR), RSTIRQS bit: Bit description changed
1.12	Nov 21, 2011	42. ROM (Flash Memory for Code Storage)	
		1466	Figure 42.1 Block Diagram of ROM, changed
1.12	Nov 21, 2011	1. Overview	
		98 to 102	Table 1.9 List of Pins and Pin Functions (100-Pin TFLGA), changed (pinsTPU6 to TPU11, and RSPI2 have been deleted)
1.20	Mar 26, 2012	103 to 107	Table 1.10 List of Pins and Pin Functions (100-Pin LQFP), changed (pinsTPU6 to TPU11, and RSPI2 have been deleted)
		—	Memory-Protection Unit (MPU) section and related information, added
1.20	Mar 26, 2012	—	"80 pin" changed to "80-pin", 80-pin part number changed to "80-pin (in planning)"
		—	Corrected fluctuation of terms (on-chip oscillator → HOCO/LOCO, description of on-chip ROM and ROM/RAM, host command → command, low/high output → 0/1 output)
1.20	Mar 26, 2012	4	Precautions when Porting Software among RX630 Group Products, added
		Features	
1.20	Mar 26, 2012	46	On-chip main flash memory, no wait states, changed
		1. Overview	
1.20	Mar 26, 2012	47 to 51	Table 1.1 Outline of Specifications: CPU, ROM, RAM, E2 DataFlash, clock generation circuit, temperature sensor, power supply voltage, changed. Low power consumption, deleted
		53 to 54	Table 1.3 List of Products, changed, notes added
1.20	Mar 26, 2012	56	Figure 1.2 Block Diagram, changed
		57 to 61	Table 1.4 Pin Functions, description of VCC, changed
1.20	Mar 26, 2012	2. CPU	
		97	2.1 Features, changed
1.20	Mar 26, 2012	114	Figure 2.8 Fixed Vector Table, changed
		3. Operating Modes	
1.20	Mar 26, 2012	125	3.1 Operating Mode Types and Selection, changed
		125	Table 3.1 Selection of Operating Modes by the Mode Pin to Table 3.4 Selection of Endian, changed
1.20	Mar 26, 2012	126	3.2.1 Mode Monitor Register (MDMONR), changed
		126	3.2.2 Mode Status Register (MDSR), changed
1.20	Mar 26, 2012	127	3.2.3 System Control Register 0 (SYSCR0), changed
		128	3.2.4 System Control Register 1 (SYSCR1), changed
1.20	Mar 26, 2012	129	3.3.1 Single-Chip Mode, changed
		129	3.3.2 On-Chip ROM Enabled Extended Mode, changed
1.20	Mar 26, 2012	129	3.3.3 On-Chip ROM Disabled Extended Mode, changed
		129	3.3.4 Boot Mode, changed
1.20	Mar 26, 2012	129	3.3.5 USB Boot Mode, added
		129	3.3.6 User Boot Mode, changed
1.20	Mar 26, 2012	129	Figure 3.1 Mode-Setting Pin Level and Operating Mode, changed
		131	Figure 3.2 Setting of SYSCR0.ROME and EXBE Bits and Operating Modes, changed
1.20	Mar 26, 2012	5. I/O Registers	
		137 to 178	Table 5.1, module name, added
1.20	Mar 26, 2012	6. Resets	
		180 to 181	Table 6.2 Targets to be Initialized by Each Reset Source, changed

Rev.	Date	Description	
		Page	Summary
1.20	Mar 26. 2012	182	6.2.1 Reset Status Register 0 (RSTSR0), notes changed
		184 to 185	6.2.3 Reset Status Register 2 (RSTSR2), notes changed
		187	Figure 6.1 Operation Examples During a Power-On Reset and Voltage Monitoring 0 Reset, changed
		188	Figure 6.2 Operation Examples During Voltage Monitoring 1 Reset and Voltage Monitoring 2 Reset, changed
		7. Option-Setting Memory	
		192	Figure 7.1 Option-Setting Memory Area, changed
		193	7.2.1 Option Function Select Register 0 (OFS0), note deleted
		198	7.2.3 Endian Select Register B (MDEB), Endian Select Register S (MDES), changed
		8. Voltage Detection Circuit (LVD)	
		200	Table 8.1 Voltage Detection Circuit Specifications, changed
		204	8.2.2 Voltage Monitoring 1 Circuit Status Register (LVD1SR), description changed
		213	8.3.2 Monitoring Vdet1, changed
		213	Table 8.2 Procedures for Setting up Monitoring against Vdet1, changed
		213	8.3.3 Monitoring Vdet2, changed
		213	Table 8.3 Procedures for Setting up Monitoring against Vdet2, changed
		214	Figure 8.4 Example of Voltage Monitoring 0 Reset Operation, changed
		216	Figure 8.5 Example of Voltage Monitoring 1 Interrupt Operation, changed
		218	Figure 8.6 Example of Voltage Monitoring 2 Interrupt Operation, changed
		9. Clock Generation Circuit	
		219	Table 9.1 Specifications of Clock Generation Circuit: Operating frequency, changed, notes added
		226	9.2.5 PLL Control Register 2 (PLLCR2), description changed
		228	9.2.7 Main Clock Oscillator Control Register (MOSCCR), description changed
		229	9.2.8 Sub-Clock Oscillator Control Register (SOSCCR), description changed
		237	Figure 9.2 Example of Crystal Resonator Connection, changed
		239	Figure 9.5 Connection Example of 32.768-kHz Crystal Resonator, changed
		239	9.4.2 Handling of Pins when Sub-Clock is Not Used, changed
		239	Figure 9.7 Pin Handling when Sub-Clock is not Used, changed
		241	Figure 9.8 Flow of Recovery from Detection of Oscillator Stop, changed
		242	9.7.3 FlashIF Clock, changed
		244	9.8 Pin Settings When an Oscillator is Connected, description changed
		245	9.10.5 Notes on Sub-Clock, changed
		11. Low Power Consumption	
		255	Table 11.2 Entering and Exiting Low Power Consumption Modes and Operating States in Each Mode, notes changed
		256	Figure 11.1 Mode Transitions, changed
		262	11.2.5 Operating Power Control Register (OPCCR), description changed
		263	Table 11.3 Relationship between Operating Power Control Mode, Operating Range, and Power Consumption, changed
		266	11.2.8 Sub-Clock Oscillator Wait Control Register (SOSCWTCR), changed
		285	11.6.2.1 Transition to All-Module Clock Stop Mode, description changed, notes deleted
		287	11.6.3.2 Canceling Software Standby Mode, description changed
		291	Figure 11.3 Example of Deep Software Standby Mode Application, changed
		292	Figure 11.4 Example of Flowchart to Use Deep Software Standby Mode, changed
		294	11.7.9 Canceling All-Module Clock Stop Mode, changed
		294	11.7.11 Points for Caution on Return from Software Standby, added
		294	11.7.12 Point for Caution when Shifting from Low-Speed Mode to Software Standby Mode, added
		13. Register Write Protection Function	

Rev.	Date	Description	
		Page	Summary
1.20	Mar 26, 2012	298	Table 13.1 Association between PRCR Bits and Registers to be Protected, changed
		14. Exception Handling	
		300	Figure 14.1 Types of Exception Events, changed
		300	14.1.3 Access Exceptions, added
		302	Figure 14.2 Outline of Exception Handling Procedure, changed
		303	Table 14.1 Acceptance Timing and Saved PC Value, changed
		304	Table 14.2 Vector and Site for Saving the Values in the PC and PSW, changed
		305	14.5.3 Access Exceptions, added
		307	Table 14.3 Return from Exception Handling Routine, changed
		307	Table 14.4 Priority of Exception Events, content changed
		15. Interrupt Controller (ICUb)	
		308	Table 15.1 Specifications of Interrupt Controller, changed
		324	15.2.15 Non-Maskable Interrupt Status Clear Register (NMICLR), description changed
		—	15.2.21 Group m Interrupt Clear Register (GCRm) (m = group number): GCR12 register, deleted
		334 to 340	Table 15.3 Interrupt Vector Table: Changed, columns sorted by pin number deleted, notes deleted
		342 to 343	Table 15.4 Group m Interrupt Requests, changed
		16. Buses	
		360	Figure 16.1 Bus Configuration, changed
		361	Table 16.2 Addresses Assigned for Each Bus, changed
		362	Table 16.4 Connection of Peripheral Modules to the Internal Peripheral Buses, changed
		363	16.2.5 Write Buffer Function (Internal Peripheral Bus), title added
		368	16.3.1 CSn Control Register (CSnCR) (n = 0 to 7), notes deleted, register description changed
		369	16.3.2 CSn Recovery Cycle Register (CSnREC) (n = 0 to 7), register description changed
		371	16.3.3 CS Recovery Cycle Insertion Enable Register (CSRECEN), register description changed
		374	16.3.4 CSn Mode Register (CSnMOD) (n = 0 to 7), register description changed
		376	16.3.5 CSn Wait Control Register 1 (CSnWCR1) (n = 0 to 7), register description changed
		379	16.3.6 CSn Wait Control Register 2 (CSnWCR2) (n = 0 to 7), register description changed
		383	16.3.9 Bus Error Status Register 1 (BERSR1), changed
		419	16.5.6 Write Buffer Function (External Bus), title changed
		17. Memory-Protection Unit (MPU) section, added	
		19. Data Transfer Controller (DTCa)	
		488	19.2.8 DTC Vector Base Register (DTCVBR), changed
		505	19.5 DTC Setting Procedure, changed
		511	19.9.3 Setting the DTC Activation Enable Register (ICU.DTCERn) of the Interrupt Controller, changed
		20. I/O Ports	
		515 to 518	Figure 20.1 to Figure 20.4 I/O Port Configuration, changed
		527	Table 20.3 Handling of Unused Pins, changed
		21. Multi-Function Pin Controller (MPC)	
		543	21.2.1 Write-Protect Register (PWPR), changed
		549	Table 21.10 Register Settings for Input/Output Pin Function in 177-/145-/100-Pin TFLGA, 176-Pin LFBGA, 176-/144-/100-Pin LQFP, note added
		553	Table 21.15 Register Settings for Input/Output Pin Function in 177-/145-Pin TFLGA, 176-Pin LFBGA, 176-/144-Pin, notes added
		563	Table 21.27 Register Settings for Input/Output Pin Function in 100-Pin TFLGA, 100-Pin LQFP, notes added
		568	21.2.19 PKn Pin Function Control Register (PKnPFS) (n = 2 to 5), addresses changed
		569	21.2.21 CS Output Pin Select Register 0 (PFCSS0), notes added
		570	21.2.22 CS Output Pin Select Register 1 (PFCSS1), notes added

Rev.	Date	Description			
		Page	Summary		
1.20	Mar 26. 2012	572	21.2.26 External Bus Control Register 1 (PFBCR1), changed		
		578 to 579	Table 21.36 How to Set the External Bus Interface (145-Pin, 144-Pin), added		
		580 to 581	Table 21.37 How to Set the External Bus Interface (100-Pin), added		
		581	21.4.1 Procedure for Specifying Input/Output Pin Function, changed		
		583	Table 21.38 Register Settings: Content changed, notes changed / deleted		
		22. Multi-Function Timer Pulse Unit 2 (MTU2a)			
		585 to 586	Table 22.2 MTU Functions (2/2), changed		
		592 to 593	22.2.2 Timer Mode Register (TMDR), changed		
		617	22.2.17 Timer Output Master Enable Registers (TOER), changed		
		618 to 619	22.2.18 Timer Output Control Registers 1 (TOCR1), notes changed		
		620 to 621	22.2.19 Timer Output Control Registers 2 (TOCR2), notes changed		
		629	22.2.29 Timer Dead Time Enable Registers (TDER), notes deleted		
		630	22.2.30 Timer Waveform Control Registers (TWCR), notes changed		
		643	22.3.4 Cascaded Operation, description changed		
		646	Figure 22.23 Cascaded Operation Example (c), notes added		
		662	Figure 22.38 Example of Complementary PWM Mode Setting Procedure, changed		
		662	Figure 22.40 Example of Operation in Complementary PWM Mode, changed		
		667	Figure 22.41 Example of Operation without Dead Time, changed		
		672	Figure 22.46 Example of Waveform Output in Complementary PWM Mode (1), legend added		
		672	Figure 22.47 Example of Waveform Output in Complementary PWM Mode (2), legend added		
		673	Figure 22.48 Example of Waveform Output in Complementary PWM Mode (3), legend added		
		677	Figure 22.56 Timing for Synchronous Counter Clearing, changed		
		679	Figure 22.59 Example of Synchronous Clearing in Interval Tb at Crest (Timing (6) in Figure 22.56; Bit WRE of TWCR is 1 in the MTU), changed		
		679	Figure 22.60 Example of Synchronous Clearing in Dead Time during Down-Counting (Timing (8) in Figure 22.56; Bit WRE of TWCR is 1), changed		
		680	Figure 22.61 Example of Synchronous Clearing in Interval Tb at Trough (Timing (11) in Figure 22.56; Bit WRE of TWCR is 1), changed		
		680	Figure 22.62 Example of Counter Clearing Operation by MTU3.TGRA Compare Match, changed		
		684	Figure 22.69 Example of Interrupt Skipping Operation, changed		
		685	Figure 22.70 Example of Operation when Buffer Transfer is Disabled (BTE1 = 0 and BTE0 = 1), changed		
		686	Figure 22.71 Example of Operation when Buffer Transfer is Linked with Interrupt Skipping (BTE1 = 1 and BTE0 = 0), title changed		
		688	Figure 22.73 Example of Procedure for Specifying A/D Converter Start Request Delaying Function, changed		
		689	Figure 22.74 Basic Example of A/D Converter Start Request Signal (TRG4AN) Operation, changed		
		688	22.3.9 A/D Converter Start Request Delaying Function, (3) Buffer Transfer, changed		
		690	Figure 22.75 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping (when the output of TRG4AN in counting up and down by TCNT is enabled), changed		
		690	Figure 22.76 Example of A/D Converter Start Request Signal (TRG4AN) Operation Linked with Interrupt Skipping (when the output of TRG4AN in counting up by TCNT is enabled), changed		
693	Figure 22.80 Example of Dead Time Compensation Setting Procedure, changed				
694	22.3.11 Dead Time Compensation, (2) TCNT Capture at Crest and/or Trough in Complementary PWM Mode Operation, changed				
694	Figure 22.82 MTU5.TCNT Capture at Crest and/or Trough in Complementary PWM Operation, changed				
694	22.3.12 Noise Filter, changed				
694	Figure 22.83 Timing of Noise Filtering, changed				
695	22.4.1 Interrupt Sources and Priorities, changed				
704	Figure 22.101 TGI Interrupt Timing (Compare Match) (MTU5), changed				
706	22.6.2 Input Clock Restrictions, changed				

Rev.	Date	Description	
		Page	Summary
1.20	Mar 26, 2012	706	Figure 22.106 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode, changed
		706	22.6.3 Note on Cycle Setting, changed
		713	22.6.14 Buffer Operation Setting in Complementary PWM Mode, changed
		714	Figure 22.119 Buffer Operation and Compare Match Flags in Reset-Synchronized PWM Mode, changed
		714	Figure 22.120 Overflow Flags in Reset-Synchronized PWM Mode, changed
		717	Figure 22.123 Example of Synchronous Clearing (When Condition 1 Applies), changed
		718	Figure 22.124 Example of Synchronous Clearing (When Condition 2 Applies), changed
		719	22.6.25 Continuous Output of Interrupt Signal in Response to a Compare Match, title and content changed
		719	Figure 22.125 Continuous Output of Interrupt Signal in Response to a Compare Match, changed
		721	22.7.3 Overview of Pin Initialization Procedures and Mode Transitions in Case of Error during Operation Figure 22.126 Error Occurrence in Normal Mode, Recovery in Normal Mode to Figure 22.154 Error Occurrence in Reset-Synchronized PWM Mode, Recovery in Reset-Synchronized PWM Mode: Changes made due to the addition of MPC setting
		721 to 749	22.7.3 Overview of Pin Initialization Procedures and Mode Transitions in Case of Error during Operation (1) Operation when Error Occurs in Normal Mode and Operation is Restarted in Normal Mode to (29) Operation when Error Occurs in Reset-Synchronized PWM Mode and Operation is Restarted in Reset-Synchronized PWM Mode: Description changed due to the addition of MPC setting
		23. Port Output Enable 2 (POE2a)	
		750	Table 23.1 POE Specifications, changed
		24. 16-Bit Timer Pulse Unit (TPUa)	
		794	24.2.10 Noise Filter Control Register (NFCR), notes changed
		827	Figure 24.40 TGI <sub>my</sub> Interrupt Timing (Compare Match), changed
		827	Figure 24.41 TGI <sub>my</sub> Interrupt Timing (Input Capture), changed
		828	Figure 24.42 TCImV Interrupt Setting Timing, changed
		828	Figure 24.43 TCImU Interrupt Setting Timing, changed
		829	24.10.2 Input Clock Restrictions, changed
		829	Figure 24.44 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode, changed
		830	24.10.3 Caution on Cycle Setting, changed
		26. 8-Bit Timer (TMR)	
		872	26.2.1 Timer Counter (TCNT), description changed
		872	26.2.2 Time Constant Register A (TCORA), description changed
		873	26.2.3 Time Constant Register B (TCORB), description changed
		880	26.4.2 Timing of Interrupt Flag Setting to 1 at Compare Match, changed
		880	Figure 26.7 Timing of Interrupt Flag Setting to 1 at Compare Match, changed
		882	Figure 26.11 Timing of Clearance by External Reset (High Level), changed
		883	26.4.6 Timing of Overflow Interrupt Flag Setting to 1, changed
		883	Figure 26.12 Timing of Overflow Interrupt Flag Setting to 1, changed
		886	26.7.5 Conflict between TCORA or TCORB Write and Compare Match, changed
		886	Figure 26.15 Conflict between TCORA or TCORB Write and Compare Match, changed
		27. Compare Match Timer	
		895	27.4.2 Timing of Compare Match Interrupt Generation, changed
		895	Figure 27.4 Timing of Interrupt Flag Setting to 1 at Compare Match, changed
		28. Realtime Clock (RTCa)	
		898	Table 28.2 Pin Configuration of RTC, changed
		904	28.2.9 Second Alarm Register (RSECAR) to 28.2.16 Year Alarm Enable Register (RYRAREN): Description of values after reset, changed
		912	28.2.18 RTC Control Register 2 (RCR2), changed
		914	28.2.19 RTC Control Register 3 (RCR3), changed



Rev.	Date	Description			
		Page	Summary		
1.20	Mar 26, 2012	923	Figure 28.3 Clock Setting Procedure, changed		
		924	Figure 28.4 Setting the Time, changed		
		931	Table 28.4 RTC Interrupt Sources, changed		
		934	28.5.6 Initialization Procedure when the Realtime Clock is not to be Used: Text and figure, added		
		29. Watchdog Timer (WDTA)			
		935	29.1 Overview, description changed		
		31. USB 2.0 Function Module (USBa)			
		968	31.2.1 System Configuration Control Register (SYSCFG): USBE bit, bit description changed		
		971	31.2.4 CFIFO Port Register (CFIFO) D0FIFO Port Register (D0FIFO) D1FIFO Port Register (D1FIFO): Register description changed, FIFOPORT[15:0] bits, bit description changed		
		977	31.2.6 CFIFO Port Control Register (CFIFOCTR) D0FIFO Port Control Register (D0FIFOCTR) D1FIFO Port Control Register (D1FIFOCTR): BCLR bit, bit description changed		
		980	31.2.8 BRDY Interrupt Enable Register (BRDYENB), register description changed		
		981	31.2.9 NRDY Interrupt Enable Register (NRDYENB), register description changed		
		982	31.2.10 BEMP Interrupt Enable Register (BEMPENB), register description changed		
		983	31.2.11 SOF Output Configuration Register (SOFCFG): Register description changed; EDGESTS bit, bit description changed		
		983	31.2.12 Interrupt Status Register 0 (INTSTS0): Bit function table changed; BRDY bit, NRDY bit, BEMP bit and SOFR bit, bit description changed		
		986	31.2.13 BRDY Interrupt Status Register (BRDYSTS), notes changed		
		989	31.2.16 Frame Number Register (FRMNUM), title changed		
		990	31.2.17 Device State Changing Register (DVCHGR): Bit function table, DVCHG changed		
		990	31.2.18 USB Address Register (USBADDR): Bit function table, STSRECOV[3:0] changed		
		995	31.2.24 DCP Control Register (DCPCTR): BSTS bit, bit description changed		
		997	31.2.25 Pipe Window Select Register (PIPESEL), register description changed		
		998	31.2.26 Pipe Configuration Register (PIPECFG): Bit function table changed; Note 1 to Note 3, changed; SHTNAK bit, BFRE bit, and TYPE[1:0] bits, bit description changed		
		1001	31.2.28 Pipe Cycle Control Register (PIPEPERI): IFIS bit, bit description changed		
		1002	31.2.29 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9): • PIPEnCTR (n = 1 to 5), bit function table changed; Notes changed; PID[1:0] bits, ATREPM bit, INBUFM bit, and BSTS bit, bit description changed		
		1004	Table 31.6 USB Module Operation as Determined by the Setting of the PID[1:0] Bits, title changed		
		1005	Table 31.7 Information Cleared by USB Module by Setting ACLRM = 1: 3 and 4, changed		
		1005	Table 31.8 Operation of BSTS Bit: BSTS Bit Function, changed		
		1006 to 1007	31.2.29 PIPEn Control Registers (PIPEnCTR) (n = 1 to 9): • PIPEnCTR (n = 6 to 9), bit function table changed; Notes changed, PID[1:0] bits and BSTS bit, bit description changed		
		1007	Table 31.9 Information Cleared by USB Module by Setting ACLRM = 1: No. 2, changed		
		1008	31.2.30 PIPEn Transaction Counter Enable Registers (PIPEnTRE) (n = 1 to 5): Bit function table changed; Notes changed		
		1009	31.2.31 PIPEn Transaction Counter Registers (PIPEnTRN) (n = 1 to 5): TRNCNT[15:0] bit, bit description changed		
		1012	Figure 31.2 Sample Peripheral Connection of USB Connector (USB0) in Self-Powered State, changed		
		1019 to 1021	31.3.3.1 BRDY Interrupt, changed (1) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 0, changed (2) When SOFCFG.BRDYM = 0 and PIPECFG.BFRE = 1, changed (3) When BRDYM = 1 and BFRE = 0, changed		
		1021	Table 31.13 Condition for Clearing BRDY Bit, changed		
		1022	31.3.3.2 NRDY Interrupt, changed		
		1024	31.3.3.3 BEMP Interrupt, changed		
		1028	31.3.3.7 VBUS Interrupt, changed		
		1029	31.3.4.1 Pipe Control Register Switching Procedures, changed		
		1029	31.3.4.2 Transfer Types, changed		
		1030	31.3.4.3 Endpoint Number, changed		

Rev.	Date	Description	
		Page	Summary
1.20	Mar 26, 2012	1030	31.3.4.4 Maximum Packet Size Setting, changed
		1030	31.3.4.6 Response PID, changed
		1031	31.3.4.8 Response PID = NAK Function, changed
		1031	31.3.4.9 Auto Response Mode, changed
		1033	31.3.5.1 FIFO Buffer Memory: Buffer Status, changed
		1033 to 1034	31.3.5.2 FIFO Buffer Clearing, changed (1) Auto Buffer Clear Mode Function, changed (2) Buffer Memory Specifications (Single or Double Setting), changed
		1034 to 1035	31.3.5.3 FIFO Port Functions, changed (1) FIFO Port Selection, changed (2) REW Bit, changed
		1035	31.3.5.4 DMA Transfers (D0FIFO and D1FIFO Ports) (1) Overview of DMA Transfers, changed (2) DnFIFO Auto Clear Mode (D0FIFO and D1FIFO Port Reading Direction), changed
		1036 to 1037	31.3.6.1 Control Transfers (1) Setup Stage, changed (2) Data Stage, changed (3) Status Stage, changed (4) Control Transfer Auto Response Function, changed
		1037	31.3.7 Bulk Transfers (PIPE1 to PIPE5), changed
		1039 to 1044	31.3.9.3 Interval Counter, changed (2) Interval Counting and Transfer Control When the selected pipe is for isochronous OUT transfers, changed When the selected pipe is for isochronous IN transfers, changed (3) Setup of Data to be Transmitted using Isochronous Transfer, changed (4) Isochronous Transfer Transmission Buffer Flush, changed
		1040	Figure 31.13 Relationship between Frames and Expected Token Reception when IITV = 0, changed
		1040	Figure 31.14 Relationship between Frames and Expected Token Reception when IITV ≠ 0, changed
		1044	Figure 31.17 Example of Interval Error Occurrence when IITV = 1, changed
		1044	31.3.10 SOF Interpolation Function, changed
		32. Serial Communications Interface (SCIc, SCId)	
		1045 to 1046	Table 32.1 Specifications of SCIc, notes added
		1046 to 1047	Table 32.2 Specifications of SCId, notes added
		1050	Figure 32.3 Block Diagram of SCI12 (SCId), changed
		1058	32.2.6 Serial Control Register (SCR) (1) Serial Communications Interface Mode: CKE[1:0] in bit function table, changed
		1066	32.2.8 Smart Card Mode Register (SCMR): Bit function table, notes added; SDIR bit, bit description changed
		1069	Table 32.12 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (1), title changed
		1069	Table 32.12 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (2), title and columns, added
		1070	Table 32.12 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (3) and (4), title added
		1070	Table 32.13 Maximum Bit Rate for Each Operating Frequency (Asynchronous Mode), changed
		1071	Table 32.14 Maximum Bit Rate with External Clock Input (Asynchronous Mode), changed
		1072	Table 32.16 Maximum Bit Rate with External Clock Input (Clock Synchronous Mode, Simple SPI Mode), changed
		1073	Table 32.17 BRR Settings for Various Bit Rates (Smart Card Interface Mode, n = 0, S = 372), changed
		1073	Table 32.18 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 372), changed
		1074	Table 32.19 BRR Settings for Various Bit Rates (Simple I <sup>2</sup> C Mode): (1) and (2), title changed, added
		1074	Table 32.20 Minimum Widths at High and Low Level for SCL at Various Bit Rates (Simple I <sup>2</sup> C Mode): (1) and (2), title changed, added
		1100	Figure 32.8 Sample SCI Initialization Flowchart (Asynchronous Mode), changed

Rev.	Date	Description	
		Page	Summary
1.20	Mar 26. 2012	1106	Figure 32.14 Example of Serial Reception Flowchart (2) (Asynchronous Mode), changed
		1111	Figure 32.19 Example of Multi-Processor Serial Reception Flowchart (2), changed
		1117	Figure 32.26 Example of Serial Reception Flowchart (Clock Synchronous Mode), changed
		1127 to 1128	32.6.8 Clock Output Control, (2) At Mode Switching (a) At transition from smart card interface mode to software standby mode: 2 and 5, changed (b) Return from software standby mode to smart card interface mode: 6, changed
		1139	Table 32.24 States of Pins by Mode and Input Level on the SSn# Pin: State of SMISOn pin, changed
		1139	32.8.3 SS Function in Slave Mode, changed
		1142	Figure 32.54 Example of Operations at the Time of Start-Frame Transmission, changed
		1145	Table 32.25 Structures of Start Frames, changed
		1146	Figure 32.57 Example of Operations at the Time of Start-Frame Reception, changed
		1148	Figure 32.59 Sample Flowchart for Reception of a Start Frame (2), changed (BDST → SDST)
		1150	Figure 32.61 Sample Flow Chart for Reception of a Start Frame (with Priority Interrupts in Use), changed
		1151	Figure 32.62 Example of Operations with Bus-Collision Detection, changed
		1152	Figure 32.63 Example of Operations with the Digital Filter, changed
		1053	Figure 32.64 Example of Operations for Bit-Rate Measurement, changed
		1154	Figure 32.65 Timing for Sampling of Data Received through RXDX12, changed
		1055	Figure 32.66 Example of Operations in Break Field Low Width Output Mode, changed
		1156	Figure 32.67 Example of Operations in Break Field Low Width Determination Mode, changed
		33. I <sup>2</sup> C Bus Interface (RIIC)	
		1169	33.2.1 I <sup>2</sup> C Bus Control Register 1 (ICCR1): Bit function table, changed; ICE bit, bit description changed
		1175	33.2.4 I <sup>2</sup> C Bus Mode Register 2 (ICMR2): Bit function table, notes added
		1191	33.2.13 I <sup>2</sup> C Bus Bit Rate Low-Level Register (ICBRL), register description changed
		1193	Table 33.5 Examples of ICBRH/ICBRL Settings for Transfer Rate: (1) to (3), title changed, added
		1196	Figure 33.5 Example of RIIC Initialization Flow, changed
		1197	33.3.3 Master Transmit Operation: (1), changed
		1200	33.3.4 Master Receive Operation: (1), changed
		1203	Figure 33.11 Master Receive Operation Timing (1) (7-Bit Address Format, when RDRFS = 0), changed
		1203	Figure 33.12 Master Receive Operation Timing (2) (10-Bit Address Format, when RDRFS = 0), changed
		1231	33.13 Interrupt Request, changed
		1231	Table 33.6 Interrupt Sources: ICRXI, changed
		35. Serial Peripheral Interface (RSPI)	
		1288	Table 35.1 Specifications of RSPI: Bit rate, changed
		1293	35.2.2 RSPI Slave Select Polarity Register (SSLP): Bit function table, Description, changed
		1294	35.2.3 RSPI Pin Control Register (SPPCR): Bit function table, MOIFV, changed
		1297 to 1298	35.2.5 RSPI Data Register (SPDR): Register description, (1) Bus Interface, (a) Writing, changed (SPRX0 to 3) → (SPRXn)
		1303	35.2.9 RSPI Data Control Register (SPDCR): SPRDTD bit, bit description changed
		1305	35.2.11 RSPI Slave Select Negation Delay Register (SSLND), register description changed
		1310	35.3.1 Overview of RSPI Operations, changed
		1311	35.3.2 Controlling RSPI Pins, changed
		1312	35.3.3 RSPI System Configuration Examples, changed
		1321 to 1322	35.3.4.1 When Parity is Disabled (SPCR2.SPPE = 0): (3) LSB-First Transfer (32-Bit Data), changed; (4) LSB-First Transfer (24-Bit Data), changed
		1329	35.3.6.1 Full-Duplex Synchronous Serial Communications (SPCR.TXMD = 0), changed
		1329	Figure 35.25 Operation Example of SPCR.TXMD = 0, changed

Rev.	Date	Description	
		Page	Summary
1.20	Mar 26, 2012	1331	35.3.7 Transmit Buffer Empty/Receive Buffer Full Interrupts, changed
		1333	Table 35.8 Relationship between Non-Normal Transfer Operations and RSPI Error Detection Function, changed
		1334	35.3.8.1 Overrun Error: 4., changed
		1335	35.3.8.2 Parity Error: 3., changed
		1336	35.3.8.3 Mode Fault Error, changed
		1337	35.3.9.1 Initialization by Clearing the SPE Bit, changed
		1340	Figure 35.32 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations, changed
		1343	Table 35.11 Relationship among SPNDEN Bit, SPND, and Next-Access Delay Value: SPC-MDm.SPNDEN bit, changed
		1344	Figure 35.34 Example of Initialization Flowchart in Master Mode (SPI Operation), changed
		—	Figure 34.35 Example of Transfer Operation Flowchart in Master Mode (SPI Operation), deleted
		1345 to 1347	35.3.10.1 Master Mode Operation: (9) Software Processing Flow, changed; (a) Transmit Processing Flow, (b) Receive Processing Flow, (c) Flow of error processing, added
		1345	Figure 35.35 Flowchart in Master Mode (Transmission), added
		1346	Figure 35.36 Flowchart in Master Mode (Reception), added
		1347	Figure 35.37 Flowchart for Master Mode (Error Processing), added
		1349	Figure 35.38 Example of Initialization Flowchart in Slave Mode (SPI Operation), changed
		1350	35.3.10.2 Slave Mode Operation: (6) Flow of Software Processing, changed: (a) Flow of processing for transmission, (b) Flow of processing for reception, (c) Flow of error processing, added
		—	Figure 34.37 Example Flowchart for Transfer Operations in Slave Mode (SPI Operation), deleted
		1350	Figure 35.39 Flowchart for Slave Mode (Transmission), added
		1350	Figure 35.40 Flowchart for Slave Mode (Reception), added
		1351	Figure 35.41 Flowchart for Slave Mode (Error Processing), added
		1352	35.3.12 Master Mode Operation: (3) Sequence Control, changed
		1354	Figure 35.44 Correspondence between the RSPI Command Register and Transmit/Receive Buffers in Sequence Operations, changed
		1355	Figure 35.45 Example of Initialization Flowchart in Master Mode (Clock Synchronous Operation), changed
		1356	35.3.12 Master Mode Operation: (5) Flow of Software Processing, changed
		—	Figure 34.42 Example Flowchart for Transfer Operations in Master Mode (Clock Synchronous Operation), deleted
		1357	Figure 35.46 Example of Initialization Flowchart in Slave Mode (Clock Synchronous Operation), changed
		1357	35.3.13 Slave Mode Operation: (4) Flow of Software Processing, changed
		—	Figure 34.44 Example Flowchart for Transfer Operations in Slave Mode (CPHA = 1) (Clock Synchronous Operation), deleted
		—	35.3.15 Error Handling, deleted
		—	Figure 35.49 Error Handling (Overrun Error), deleted
		—	Figure 35.50 Error Handling (Parity Error), deleted
		—	Figure 35.51 Error Handling (Mode Fault Error), deleted
		36. IEBus Controller (IEB)	
		1362	Table 36.1 Specifications of IEB: Operating frequency, changed
		1364	Figure 36.5 Block Diagram of IEB, changed
		1392	36.2.20 IEBus Clock Selection Register (IECKSR): Bit function table, CKS[2:0], changed
		38. 12-Bit A/D Converter (S12ADa)	
		1418	Table 38.1 Specifications of 12-Bit A/D Converter, changed
		1428	38.2.9 A/D Conversion Extended Input Control Register (ADEXICR)A/D: Bit function table, changed
		1447	38.5.11 12 Port Setting for Using 12-bit A/D Converter Inputs, changed
		1447	38.5.12 Notes on Using the External Bus, changed
		39. 10-Bit A/D Converter (ADb)	

Rev.	Date	Description	
		Page	Summary
1.20	Mar 26, 2012	1460	Figure 39.5 Configuration Example of Extended Analog Input Circuit, changed
		1472	Table 39.9 Analog Pin Specifications (Reference Values), title changed
		1475	39.6.12 Notes on Using the External Bus, added
		40. D/A Converter (DAa)	
		1480	40.2.4 D/A A/D Synchronous Start Control Register (DAADSCR), register description changed
		—	40.4.6 Notes on Usage when Measure against Interference between D/A and A/D Conversion is Disabled, deleted
		41. Temperature Sensor	
		1487	41.3.2 Setting of 12-Bit A/D Converter, changed
		42. RAM	
		1490	Table 42.1 Specifications of RAM: Data retention function, changed
		43. ROM (Flash Memory for Code Storage)	
		1491	Table 43.1 Specifications of ROM: On-board programming (four types), Command-locked state, changed
		1492	Figure 43.1 Block Diagram of ROM, changed
		1492	43.1 Overview, changed
		—	Table 43.3 Input and Output Pins Associated with the ROM, deleted
		1493	43.2.1 Flash Write Erase Protection Register (FWEPROR), register description changed
		1494	43.2.2 Flash Mode Register (FMODR), register description changed
		1495	43.2.3 Flash Access Status Register (FASTAT): Register description changed; ROMAE bit, [Setting conditions], changed
		1498	43.2.5 FCU RAM Enable Register (FCURAME): Bit function table, register description, bit description changed
		1499	43.2.6 Flash Status Register 0 (FSTATR0), register description changed
		1501	43.2.7 Flash Status Register 1 (FSTATR1): Register description changed; FCUERR bit, bit description changed
		1502	43.2.8 Flash Ready Interrupt Enable Register (FRDYIE), register description changed
		1503	43.2.9 Flash P/E Mode Entry Register (FENTRYR): Bit function table changed; register description changed; FENTRY0 bit, FENTRY1 bit, FENTRY2 bit, and FENTRY3 bit, bit description changed
		1505	43.2.10 Flash Protection Register (FPROTR), register description changed
		1506	43.2.11 Flash Reset Register (FRESETR): Bit function table, register description, bit description changed
		1507	43.2.12 FCU Command Register (FCMDR), register description changed
		1508	43.2.13 FCU Processing Switching Register (FCPSR), register description changed
		1508	43.2.14 Flash P/E Status Register (FPESTAT), register description changed
		1509	43.2.15 Peripheral Clock Notification Register (PCKAR): Register description changed; PCKA[7:0] bits, bit description changed
		1510	43.3 Configuration of Memory Areas for the ROM, changed
		1512	Table 43.4 Differences between Modes: USB Boot Mode, User Boot Mode, changed
		1512	43.5 Operating Modes Associated with the ROM, changed
		1515	Table 43.5 FCU Command Formats: Address column, changed
		1519	Figure 43.6 Procedure for Transition to ROM Read Mode, changed
		1522	Figure 43.10 Simple Flowchart of the Procedure for Programming and Erasure, changed
		1323	Figure 43.11 Procedure for Firmware Transfer to FCU RAM, changed
		1524	43.6.4.2 Programming and Erasure Procedures: (4) Using the Peripheral Clock Notification Command, changed
		1534	Figure 43.19 Procedure for Resuming Programming or Erasure, changed (tFCUR)
		1535	43.7.1 Suspension during Programming, changed
		1536	43.7.2 Suspension during Erasure (Suspension Priority Mode): Title, description, changed
1537	43.7.3 Suspension during Erasure (Erasure Priority Mode): Title, changed		
1537	43.8 Protection, changed		
1538	43.8.2 Command-Locked State, changed		

Rev.	Date	Description	
		Page	Summary
1.20	Mar 26, 2012	1539	Table 43.8 Errors that Lead to the Command-Locked State (Types Dedicated to ROM and Types Common to ROM and E2 DataFlash): Title, FENTRYR setting error, Illegal command error, ROM access violation, Notes, changed
		1541	43.9 User Boot Mode, changed
		1541	43.10.1 System Configuration, changed
		1541	Figure 43.23 System Configuration for Operations in Boot Mode, changed
		1542	Table 43.9 Input and Output Pins Associated with the ROM, changed
		1543	Figure 43.24 State Transitions in Boot Mode, changed
		1546	Table 43.11 Specifications for ID Code Protection: Operations at the Time of SCI Connection, changed
		1547	Table 43.12 Inquiry/Selection Commands: Title, function, changed
		1547 to 1555	43.10.6 Inquiry/Selection Command Wait (1) Supported Device Inquiry, changed (2) Device Selection, changed (3) Clock Mode Inquiry, changed (4) Clock Mode Selection, changed (5) Multiplication Ratio Inquiry, changed (6) Operating Frequency Inquiry, changed (7) User Boot Area Information Inquiry, changed (8) User Area Information Inquiry, changed (10) Programming Size Inquiry, changed (11) New Bit Rate Selection: Changed; Bit rate selection error, equation changed (12) Programming/Erasure State Transition, changed
		1554	Figure 43.29 New Bit Rate Selection Sequence, changed
		1557	43.10.7 ID Code Wait State: (1) ID Code Check, changed
		1558	Table 43.16 Programming/Erasure Commands, changed
		1558 to 1564	43.10.8 Programming/Erasure Command Wait State (2) User/Data Area Programming Selection, changed (3) 256-Byte Programming, changed (9) User Boot Area Blank Check, changed (10) User Area Blank Check, changed (11) Read Lock Bit Status, changed
		1566	Table 43.17 Enumeration Information: Transfer modes, changed
		1567	43.11.2 State Transitions:(1), changed
		1567	Figure 43.33 State Transitions in USB Boot Mode, changed
		1568	43.11.3 Notes on Program Execution in USB Boot Mode: (1), changed
		1568	43.12 ID Code Protection on Connection of the On-Chip Debugger, changed
		1568	Table 43.18 Specifications for ID Code Protection on Connection of the On-Chip Debugger: ID Code, Operations at On-Chip Debugger Connection, changed
		1570 to 1571	43.14 Usage Notes (4) Reset during Programming or Erasure, changed (5) Prohibition of Non-Maskable Interrupts during Programming or Erasure, changed (8) Programming/Erasure Abnormal End, changed (9) Actions Prohibited during Programming and Erasure, changed
		44. E2 DataFlash Memory (Flash Memory for Data Storage)	
		1572	Table 44.1 Specifications of E2 DataFlash Memory: On-board programming (four types), changed
		1573	Figure 44.1 Block Diagram of E2 DataFlash Memory, changed
		1573	44.1 Overview, changed
		—	Table 44.2 Input and Output Pins Associated with the E2 DataFlash, deleted
		1574	44.2.1 Flash Mode Register (FMODR): Register description changed
		1575	44.2.2 Flash Access Status Register (FASTAT): Register description changed
		1578	44.2.4 E2 DataFlash Read Enable Register 0 (DFLRE0): Bit function table, register description, changed; DBREj bit, bit description changed
		1579	44.2.5 E2 DataFlash Read Enable Register 1 (DFLRE1): Bit function table, register description, changed; DBREj bit, bit description changed
		1580	44.2.6 E2 DataFlash Programming/Erasure Enable Register 0 (DFLWE0): Bit function table, register description changed; DBWEj bit, bit description changed

Rev.	Date	Description			
		Page	Summary		
1.20	Mar 26, 2012	1581	44.2.7 E2 DataFlash Programming/Erase Enable Register 1 (DFLWE1): Bit function table, register description, changed; DBWEj bit, bit description changed		
		1582	44.2.8 Flash P/E Mode Entry Register (FENTRYR): Bit function table, register description, changed; FENTRYD bit, bit description changed		
		1583	44.2.9 E2 DataFlash Blank Check Control Register (DFLBCCNT): Register description, changed; BCADR[10:0] bits, bit description changed		
		1584	44.2.10 E2 DataFlash Blank Check Status Register (DFLBCSTAT): Register description changed		
		1585	44.3 Configuration of Memory Area for the E2 DataFlash Memory, changed		
		1586	44.5 Operating Modes Associated with the E2 DataFlash, changed		
		1586	Table 44.3 Differences between Modes: USB Boot Mode, User Boot Mode, changed		
		1591	44.6.4 FCU Command Usage: (2) Programming, changed		
		1592	Figure 44.5 Procedure for E2 DataFlash Programming, changed		
		1593	44.6.4 FCU Command Usage: (3) Erasure, changed		
		1594	Figure 44.6 Procedure for Blank Checking of the E2 DataFlash, changed		
		1596	44.7.2 Error Protection, changed		
		1597	44.8.1 Inquiry/Selection Commands, changed		
		1598	44.8.2 Programming/Erasing Commands, changed		
		1599	44.8.2 Programming/Erasing Commands: (1) Data Area Checksum, changed		
		1600	44.9 Usage Notes: (3) Other Points to Note, changed		
		45. Boundary Scan			
		1616	45.4 Usage Notes: (6), changed		
		1617	Figure 45.4 Pin Configuration, changed		
		46. Electrical Characteristics			
		1620	Table 46.3 DC Characteristics (2): Three-state leakage current (off state), Test conditions, changed; Input pull-up MOS current, notes added		
		1621	Table 46.4 DC Characteristics (3): RAM standby voltage, changed		
		1623	Table 46.6 Operation Frequency Value (High-Speed Operating Mode), notes changed		
		1624	Table 46.9 Clock Timing (Except for Sub-Clock Related): BCLK pin output high pulse width, BCLK pin output low pulse width, LOCO clock cycle time, PLL clock oscillation stabilization wait, changed		
		1625	Table 46.10 Clock Timing (Sub-Clock Related): Sub-clock oscillation stabilization wait offset time, changed; notes added		
		1625	Figure 46.1 BCLK Pin Output Timing, changed (SDCLK)		
		1626	Figure 46.5 HOCO Oscillation Start Timing (After Reset is Canceled by Setting the OFS1.HOCOEN Bit to 0), changed		
		1626	Figure 46.6 HOCO Clock Oscillation Start Timing (Oscillation is Started by Setting the HOCOEN.HCSTP Bit), changed		
		1627	Figure 46.10 Sub-Clock Oscillation Start Timing, changed		
		1628	Table 46.11 Reset Timing: Programming or erasure of the ROM or E2 data-flash memory or blank checking of the E2 DataFlash memory; changed; Wait time after RES# cancellation, changed; Note, deleted		
		1628	Figure 46.11 Reset Input Timing at Power-On, changed		
		1628	Figure 46.12 Reset Input Timing, changed		
		1630	Figure 46.14 Deep Software Standby Mode Cancellation Timing, changed		
		1631	Table 46.14 Bus Timing: Read data setup time, changed		
		1633	Figure 46.19 External Bus Timing/Normal Read Cycle (Bus Clock Synchronized), changed		
		1634	Figure 46.20 External Bus Timing/Normal Write Cycle (Bus Clock Synchronized), changed		
		1635	Figure 46.21 External Bus Timing/Page Read Cycle (Bus Clock Synchronized), changed		
		1635	Figure 46.22 External Bus Timing/Page Write Cycle (Bus Clock Synchronized), changed		
		1636	Figure 46.23 External Bus Timing/External Wait Control, changed		
		1638	Table 46.16 Timing of On-Chip Peripheral Modules (2): Conditions, Test Conditions, Data input setup time (Slave), changed		
		1639	Table 46.17 Timing of On-Chip Peripheral Modules (3): Data input hold time, changed		

Rev.	Date	Description	
		Page	Summary
1.20	Mar 26, 2012	1640	Table 46.18 Timing of On-Chip Peripheral Modules (4): min, max, changed; Note, added
		1641	Table 46.19 Timing of On-Chip Peripheral Modules (5): min, max, changed; Note, added
		1644	Figure 46.32 RSPI Clock Timing and Simple SPI Clock Timing, changed
		1645	Figure 46.33 RSPI Timing (Master, CPHA = 0) and Simple SPI Timing (Master, CKPH = 1), changed
		1645	Figure 46.34 RSPI Timing (Master, CPHA = 1) and Simple SPI Timing (Master, CKPH = 0), changed
		1646	Figure 46.35 RSPI Timing (Slave, CPHA = 0) and Simple SPI Timing (Slave, CKPH = 1), changed
		1646	Figure 46.36 RSPI Timing (Slave, CPHA = 1) and Simple SPI Timing (Slave, CKPH = 0), changed
		1647	Figure 46.37 IIC Bus Interface Input/Output Timing and Simple IIC Bus Interface Input/Output Timing, changed
		1648	Table 46.20 On-Chip USB Full-Speed Characteristics (DP and DM Pin Characteristics): Conditions, changed
		1649	Table 46.21 10-Bit A/D Conversion Characteristics: Note, changed
		1650	Table 46.22 12-Bit A/D Conversion Characteristics: Note, changed
1.50	Sep 28, 2012	All	
		—	Corrected fluctuation of terms (on-chip RAM → RAM; distinguished on-chip ROM from ROM; pin reset, reset from pin → RES# pin reset; IWDT LOCO clock, IWDT-dedicated low speed clock, → IWDT-dedicated clock (IWDTCCLK); high speed clock → HOCO clock; low speed clock → LOCO clock; 0 output → low output; 1 output → high output; CANCLK → CANMCLK)
		1. Overview	
		47 to 96	TRSYNC# pin, changed to TRSYNC pin
		2. CPU	
		106	2.4 Data Types: Description, changed
		—	2.4.1 Integer to 2.4.4 Strings, deleted
		117	Table 2.14 Instructions that are Converted into Multiple Micro-Operations, changed
		6. Reset	
		187	Figure 6.2 Operation Examples During Voltage Monitoring 1 Reset and Voltage Monitoring 2 Reset, changed
		7. Option-Setting Memory	
		198	Figure 7.2 UB Code A Structure, changed
		198	Figure 7.3 UB Code B Structure, changed
		9. Clock Generation Circuit	
		—	IWDT-dedicated low speed clock, changed to IWDT-dedicated clock In the text, WAIT instruction issued, changed to WAIT instruction executed
		215	Figure 9.1 Block Diagram of Clock Generation Circuit, changed
		218	9.2.2 System Clock Control Register 2 (SCKCR2): Note 1, added
		11. Low Power Consumption	
		—	In the text, WAIT instruction issued, changed to WAIT instruction executed
		260	11.2.7 Main Clock Oscillator Wait Control Register (MOSCWTCR): Description, changed
		261, 262	11.2.8 Sub-Clock Oscillator Wait Control Register (SOSCWTCR): Description, changed
		263, 264	11.2.9 PLL Wait Control Register (PLLWTCR): Description, changed
		282	11.6.2.1 Transition to All-Module Clock Stop Mode: Description, changed, Note 5, added
		285	Figure 11.2 Example of Software Standby Mode Application, changed
		15. Interrupt Controller (ICUb)	
		331	15.3.1 Interrupt Vector Table: Text, added
		332	Table 15.3 Interrupt Vector Table, changed
16. Buses			
393	Figure 16.18 Example of Normal-Write Operation (when 32-Bit Bus Space is Accessed in 16 Bits, in Byte Strobe Mode) (n = 0 to 7), changed		
394	Figure 16.19 Example of Normal-Write Operation (when 32-Bit Bus Space is Accessed in 32 Bits, in Single Write Strobe Mode) (n = 0 to 7), deleted		



Rev.	Date	Description	
		Page	Summary
1.50	Sep 28, 2012	406	Figure 16.34 Example of Operation When a Read Access is Split (with Address/Data Multiplexed I/O Interface; m = 0, 1), Figure 16.35 Example of Operation When a Write Access is Split (with Address/Data Multiplexed I/O Interface; m = 0, 1), deleted
		—	Figure 16.43 Example of Operation for Recovery Cycles when the BCLK Pin Output Selection Bits Are Set for Frequency-Division of BCLK by 2 (For the Case where Bus Access is Divided up; m = 0 to 3), deleted
		414	Table 16.11 Types of Bus Errors, changed
		18. DMA Controller (DMACA)	
		435	Figure 18.1 Block Diagram of DMAC, changed
		19. Data Transfer Controller (DTCa)	
		472	Figure 19.1 Block Diagram of DTC, changed
		484	Figure 19.4 Operation Flowchart of the DTC, changed
		20. I/O Ports	
		517	Table 20.3 Handling of Unused Pins, changed
		517	20.5 Usage Notes, added
		21. Multi-Function Pin Controller (MPC)	
		558	21.2.21 CS Output Pin Select Register 0 (PFCSS0): Note, deleted
		559	21.2.22 CS Output Pin Select Register 1 (PFCSS1): Note, deleted
		561	21.2.26 External Bus Control Register 1 (PFBCR1): Note, deleted from WAIT Select Bits
		563	21.3 How to Set the External Bus Interface, changed
		563 to 565	Table 21.35 How to Set the External Bus Interface, changed
		566	21.4.2 Notes on MPC Register Setting: Text, added to (6)
		567	Table 21.36 Register Settings: Note, changed
		—	Table 21.36 How to Set the External Bus Interface (145-Pin, 144-Pin): deleted
		—	Table 21.37 How to Set the External Bus Interface (100-Pin): deleted
		22. Multi-Function Timer Pulse Unit 2 (MTU2a)	
		571	Figure 22.1 Block Diagram of MTU, changed
		595	22.2.9 Timer A/D Converter Start Request Control Register (TADCR): Note, changed
		656	22.3.8 Complementary PWM Mode, (j) Method for Generating PWM Output in Complementary PWM Mode: Data, changed to compare
		658 to 660	Figure 22.49 Example of 0% and 100% Waveform Output in Complementary PWM Mode (1) to Figure 22.53 Example of 0% and 100% Waveform Output in Complementary PWM Mode (5), moved
		682	22.4.3 A/D Converter Activation, (4) A/D Converter Activation by Input Capture or Compare Match with MTU0.TGRA or MTU0.TGRB, changed
		682	Table 22.58 Interrupt Sources and A/D Converter Start Request Signals, changed
		694	Figure 22.112 Contention between TGR Read Operation and Input Capture (MTU0 to MTU5): changed
		—	Figure 22.113 Contention between TGR Read Operation and Input Capture (MTU5), deleted
		701	22.6.24 Point for Caution Regarding MTU5.TCNT and MTU5.TGR Registers, added
		705	22.7.2 Operation in Case of Re-Setting Due to Error during Operation: Description, changed
		708	22.7.3 Overview of Pin Initialization Procedures and Mode Transitions in Case of Error during Operation, (2) Operation when Error Occurs in Normal Mode and Operation is Restarted in PWM Mode 1: Text, added
		716	22.7.3 Overview of Pin Initialization Procedures and Mode Transitions in Case of Error during Operation, (9) Operation when Error Occurs in PWM Mode 1 and Operation is Restarted in PWM Mode 2: Text (12), deleted
		23. Port Output Enable 2 (POE2a)	
		732	Table 23.1 POE Specifications, changed
		733	Figure 23.1 POE Block Diagram, changed
		735	23.2.1 Input Level Control/Status Register 1 (ICSR1): Bit table, changed
		738	23.2.3 Input Level Control/Status Register 2 (ICSR2): Bit table, changed

Rev.	Date	Description	
		Page	Summary
1.50	Sep 28, 2012	24. 16-Bit Timer Pulse Unit (TPUa)	
		748	Table 24.2 TPU (Unit 0) Functions, changed
		749	Table 24.3 TPU (Unit 1) Functions, changed
		750	Figure 24.1 Block Diagram of TPU (Unit 0), changed
		751	Figure 24.2 Block Diagram of TPU (Unit 1)TPU, changed
		769	24.2.4 Timer Interrupt Enable Register (TIER): Note 3, added
		804	24.7 A/D Converter Activation: Description, changed
		815	Figure 24.52 Conflict between Overflow and Counter Clearing, changed
		816	Figure 24.53 Conflict between TPUM.TCNT Write and Overflow, changed
		817	Figure 24.54 Continuous Output of Compare-Match Pulse Interrupt Signal, changed
		817	Figure 24.55 Continuous Output of Input-Capture Pulse Interrupt Signal, changed
		818	Figure 24.56 Continuous Output of Underflow Pulse Interrupt Signal, changed
		26. 8-Bit Timer (TMR)	
		850	Table 26.2 Pin Configuration of TMR, changed
		28. Realtime Clock (RTCa)	
		—	Standby, changed to standby mode
		879	Table 28.1 Specifications of RTC: Note 1, added
		905	Figure 28.3 Clock Setting Procedure, changed
		908	Figure 28.6 Reading Time, changed
		912	28.3.8.5 Capturing the Time: Title, changed
		916	Figure 28.14 Initialization Procedure, changed
		30. Independent Watchdog Timer (IWDTa)	
		932	Table 30.1 Specifications of IWDT, changed
		31. USB 2.0 Function Module (USBa)	
		—	Standby, changed to standby mode
		32. Serial Communications Interface (SCIc, SCId)	
		1027, 1028	Table 32.1 Specifications of SCIc, Table 32.2 Specifications of SCId, changed
		1040	32.2.6 Serial Control Register (SCR): Note 2, changed
		1043	32.2.7 Serial Status Register (SSR), changed
		1048	Table 32.8 Relationships between N Setting in BRR and Bit Rate B: Note, changed
		1049	Table 32.11 Base Clock Settings in Smart Card Interface Mode, changed
		1048	32.2.9 Bit Rate Register (BRR): Description, changed
		1051	Table 32.12 Examples of BRR Settings for Various Bit Rates (Asynchronous Mode) (4), changed
		1053	Table 32.15 BRR Settings for Various Bit Rates (Clock Synchronous Mode, Simple SPI Mode): Note, changed
		1054	Table 32.18 Maximum Bit Rate for Each Operating Frequency (Smart Card Interface Mode, S = 372), changed
		1057	Figure 32.4 Example of Average Transfer Rate Setting when TMR Clock is Input, changed
		1083	Figure 32.10 Example of Serial Transmission Flowchart in Asynchronous Mode, changed
		1087	Figure 32.14 Example of Serial Reception Flowchart (2) (Asynchronous Mode), changed
		1089	Figure 32.16 Example of Multi-Processor Serial Transmission Flowchart, changed
		1092	Figure 32.19 Example of Multi-Processor Serial Reception Flowchart (2), changed
		1094	Figure 32.21 Example of SCI Initialization Flowchart (Clock Synchronous Mode), changed
		1098	Figure 32.26 Example of Serial Reception Flowchart (Clock Synchronous Mode), changed
		1109	32.6.8 Clock Output Control, (2) At Mode Switching: Text, added to 5
		1120	32.8.3 SS Function in Slave Mode: Description, changed
		1124	Figure 32.55 Sample Flowchart for Transmission of a Start Frame (1), changed
		1141	32.12.3 The Mark State and Production of Breaks: Description, changed

Rev.	Date	Description	
		Page	Summary
1.50	Sep 28, 2012	1142	32.12.9 SCI Operations during Low Power Consumption State, (1) Transmission: Description, changed
		1143	Figure 32.69 Example of Flowchart for Transition to Software Standby Mode during Transmission, changed
		1145	32.12.11 Limitations on Simple SPI Mode, (1) Master Mode: Description, changed
		33. I <sup>2</sup> C Bus Interface (IIC)	
		1173	33.2.14 I <sup>2</sup> C Bus Bit Rate High-Level Register (ICBRH): Note 1, changed
		1196	33.7.4 Host Address Detection: Description, changed
		34. CAN Module (CAN)	
		1221	34.2.2 Bit Configuration Register (BCR), changed
		35. Serial Peripheral Interface (RSPI)	
		1269	Table 35.1 Specifications of RSPI, changed
		1277	35.2.4 RSPI Status Register (SPSR): Description, changed
		36. IEBus Controller (IEB)	
		—	36.1.1 IEBus Communications Protocol, 36.1.2 Communications Protocol, 36.1.4 Bit Format, deleted
		43. Flash Memory	
		1465 to 1554	43. Flash Memory, added (Created by integrating the previous version's 43. ROM (Flash Memory for Code Storage) and 44. E2 DataFlash (Flash Memory for Data Storage))
		45. Electrical Characteristics	
		All	Characteristics and timing conditions in the tables, changed
		1577	45.3 AC Characteristics, changed
		1579	Table 45.10, Clock Timing (Except for Sub-Clock Related): Condition and the table, changed, note, added
		1580	Table 45.11 Clock Timing (Sub-Clock Related): Condition and the table, changed, note, added
		Appendix 1. Port States in Each Processing Mode	
1614	Table 1.1 Port States in Each Processing State, changed		

---

RX630 Group User's Manual: Hardware

Publication Date: Rev.0.50 Mar 13, 2011  
Rev.1.50 Sep 28, 2012

Published by: Renesas Electronics Corporation

---

**SALES OFFICES**

Renesas Electronics Corporation

<http://www.renesas.com>Refer to "<http://www.renesas.com/>" for the latest and detailed information.

**Renesas Electronics America Inc.**  
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.  
Tel: +1-408-588-6000, Fax: +1-408-588-6130

**Renesas Electronics Canada Limited**  
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada  
Tel: +1-905-898-5441, Fax: +1-905-898-3220

**Renesas Electronics Europe Limited**  
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K  
Tel: +44-1628-651-700, Fax: +44-1628-651-804

**Renesas Electronics Europe GmbH**  
Arcadiastrasse 10, 40472 Düsseldorf, Germany  
Tel: +49-211-65030, Fax: +49-211-6503-1327

**Renesas Electronics (China) Co., Ltd.**  
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China  
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

**Renesas Electronics (Shanghai) Co., Ltd.**  
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China  
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

**Renesas Electronics Hong Kong Limited**  
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong  
Tel: +852-2886-9318, Fax: +852-2886-9022/9044

**Renesas Electronics Taiwan Co., Ltd.**  
13F, No. 363, Fu Shing North Road, Taipei, Taiwan  
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

**Renesas Electronics Singapore Pte. Ltd.**  
80 Bendemeer Road, Unit #06-02 Hyflux Innovation Centre Singapore 339949  
Tel: +65-6213-0200, Fax: +65-6213-0300

**Renesas Electronics Malaysia Sdn.Bhd.**  
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia  
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

**Renesas Electronics Korea Co., Ltd.**  
11F., Samik Lavied'or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea  
Tel: +82-2-558-3737, Fax: +82-2-558-5141

RX630 Group

## X-ON Electronics

Largest Supplier of Electrical and Electronic Components

*Click to view similar products for [Microprocessors - MPU category](#):*

*Click to view products by [Renesas manufacturer](#):*

Other Similar products are found below :

[MC68302EH20C](#) [MC7457RX1000LC](#) [MC7457RX1267LC](#) [A2C00010998 A](#) [A2C52004004](#) [R5F117BCGNA#20](#) [R5F52106BDLA#U0](#)  
[R5S72690W266BG#U0](#) [ADJ3400IAA5DOE](#) [MPC8245TVV266D](#) [MPC8245TZU300D](#) [MPC8260ACVVMHBB](#) [MPC8323ECVRAFDCA](#)  
[MPC8536ECVJAVLA](#) [BOXNUC5PGYH0AJ](#) [20-668-0024](#) [P1010NSN5DFB](#) [P2020NXE2HHC](#) [P5020NSE7TNB](#) [P5020NSE7VNB](#)  
[LS1020ASN7KQB](#) [LS1020AXN7HNB](#) [LS1020AXN7KQB](#) [A2C00010729 A](#) [A2C00039344](#) [T1022NSE7MQB](#) [T1022NXN7PQB](#)  
[T1023NSE7MQA](#) [T1024NXE7PQA](#) [T1042NSN7MQB](#) [T1042NXN7WQB](#) [T2080NSN8PTB](#) [T2080NXE8TTB](#) [T2081NXN8TTB](#)  
[R5F101AFASP#V0](#) [MC68302CEH20C](#) [TS68040MF33A](#) [MPC8260ACVVMIBB](#) [MPC8280CZUUPEA](#) [MPC8313ECVRAFFC](#)  
[MPC8313ECVRAGDC](#) [MPC8313EVRADDC](#) [MPC8313EVRAFFC](#) [MPC8313VRADDC](#) [MPC8323EVRAFDCA](#) [BOXSTCK1A8LFCL](#)  
[UPD78F0503AMCA-CAB-G](#) [UPD78F0513AGA-8EU-AT](#) [UPD78F0730MC-CAB-AX](#) [DF2134BFA20V](#)